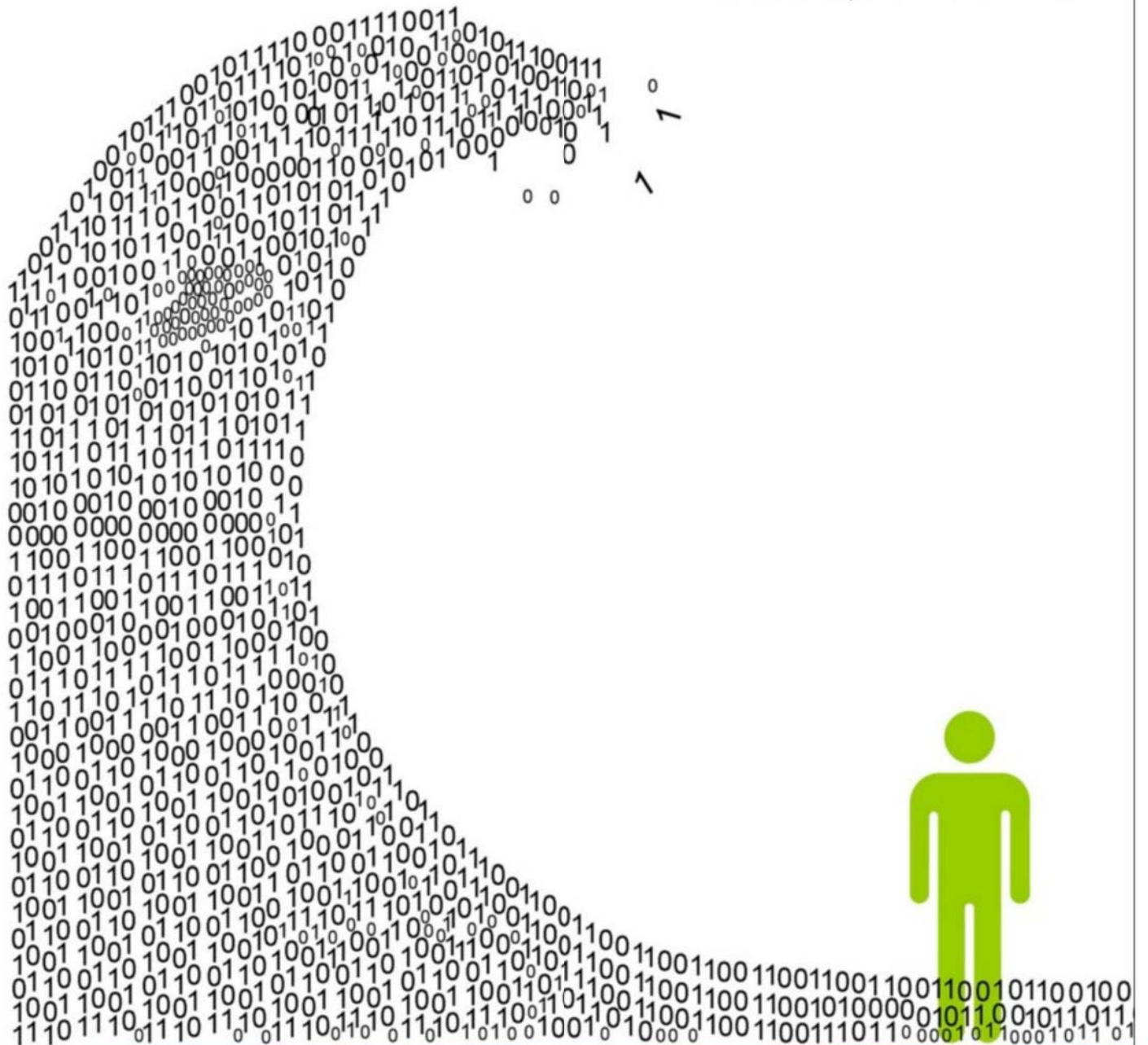




داده کاوی با R

به همراه تحلیل شبکه های اجتماعی و متن کاوی

تالیف: دکتر بابک تیمورپور، حیدر نبفی



إِنَّ اللَّهَ الرَّحْمَنُ الرَّحِيمُ

٢٥



داده‌کاوی با R

به همراه متن‌کاوی و تحلیل شبکه‌های اجتماعی

نویسندگان :

بابک تیمورپور

حیدر نجفی

آبان ۹۴

سرشناسه	: تیمورپور، بابک، ۱۳۵۰
شناسه افزوده	: نجفی، حیدر، ۱۳۶۹
عنوان و نام پدیدآور	: داده کاوی با R به همراه متن کاوی و تحلیل شبکه‌های اجتماعی / نویسنده بابک تیمورپور، حیدر نجفی.
مشخصات ظاهری	: ۳۰۱ ص.: مصور، جدول، نمودار.
مشخصات نشر	: تهران: مرکز تحقیقات و توسعه سازمان اتکا، ۱۳۹۴.
شابک	: ۱۵۰۰۰۰ ریال: ۹۷۸-۶۰۰-۷۶۲۴-۶۸-۵
وضعیت فهرست‌نویسی	: فیپا
موضوع	: داده کاوی
موضوع	: رسانه های اجتماعی
شناسه افزوده	: اتکا. مرکز تحقیقات و توسعه
رده‌بندی کنگره	: ۱۳۹۴ ت۹ د۲/۹ QAV۶
رده‌بندی دیویی	: ۰۰۶/۳۱۲
شماره کتابشناسی ملی	: ۳۹۳۳۳۸۳



انتشارات مرکز تحقیقات و توسعه سازمان اتکا

عنوان:	داده کاوی با R به همراه متن کاوی و تحلیل شبکه‌های اجتماعی
مؤلفان:	بابک تیمورپور، حیدر نجفی
ویراستار:	-
زمان و نوبت چاپ:	آبان ۱۳۹۴، اول
شمارگان:	۲۰۰ نسخه
طراح جلد:	حیدر نجفی
شابک:	۹۷۸-۶۰۰-۷۶۲۴-۶۸-۵
بهاء:	۱۵۰,۰۰۰ ریال

کلیه حقوق چاپ این کتاب محفوظ و متعلق به مرکز تحقیقات، توسعه و کیفیت سازمان اتکا است.

هر نوع تکثیر یا نشر تمام یا بخشی از این کتاب منوط به اجازه کتبی سازمان اتکا خواهد بود.

نشانی: تهران، خ امام خمینی (ره)، خ شهید مرادی، ساختمان شهدای مرکزی اتکا، ط ۲

تلفن: ۰۲۱-۶۱۹۱۵۱۲۴ دورنگار: ۰۲۱-۶۱۹۱۵۳۵۵ وبسایت: www.etkaec.ir

ایمیل: tec@ERDCenter.com

بسمه تعالی

«کار علمی و تحقیقاتی تان را جدی بگیرید... دانش خیلی از نیازهای شما در دانشگاه‌ها وجود دارد.»

مقام معظم رهبری

حضرت امام خامنه‌ای (مدظله‌العالی)

داده‌کاوی^۱ یکی از مفاهیم نوین میان‌رشته‌ای است که داده‌های حوزه‌های مختلفی همچون ریاضیات، آمار، فناوری اطلاعات، مدیریت و سایر زمینه‌های مرتبط را باهم ترکیب می‌کند تا اطلاعات و دانش مفید و موثر پنهان در حجم بزرگی از داده‌ها را استخراج نماید. امروزه با گسترش فزاینده فناوری اطلاعات، تقریباً همه سازمان‌ها حجم عظیمی از داده‌ها را در پایگاه داده خود ذخیره می‌نمایند. داده‌کاوی؛ این پایگاه‌ها را، متعقب کشف و استخراج دانش، با استفاده از روشهای یادگیری ماشینی^۲ و نیمه ماشینی تحلیل می‌نماید.

در این راستا و به منظور بهره‌برداری حداکثری در حوزه داده‌کاوی، نرم‌افزارهای مختلفی برای سازمان‌ها مهیا شده که یکی از مهمترین و پرکاربردترین آنها نرم‌افزار R می‌باشد. این نرم‌افزار یک زبان برنامه‌نویسی و محیط نرم‌افزاری برای محاسبات آماری و تحلیل داده‌ها است و به‌صورت رایگان در دسترس عموم قرار گرفته است. نرم‌افزار R حاوی محدوده‌ی گسترده‌ای از تکنیک‌های آماری از جمله، مدل‌سازی خطی و غیرخطی، آزمون‌های کلاسیک آماری، تحلیل سری‌های زمانی، رده‌بندی، خوشه‌بندی و قابلیت‌های گرافیکی است.

کتاب "داده‌کاوی با نرم‌افزار R به همراه متن‌کاوی و تحلیل شبکه‌های اجتماعی" توسط پژوهشگر محترم مرکز تحقیقات و نوآوری سازمان اتکا آقای حیدر نجفی گردآوری و با همکاری دکتر تیمورپور تالیف گردیده است. ضمن سپاسگزاری از مولفین محترم، امیدوارم این اثر ارزشمند دریچه‌ای نو به فهم مسائل مرتبط به تجزیه و تحلیل داده‌ها و اطلاعات به روی مخاطبان ارجمند بگشاید و برای ارتقا سطح دانش مدیران و کارکنان این سازمان و کلیه کاربران در سطح میهن عزیز اسلامی ایران مفید واقع گردد. انشاء الله ...

محمد مهدی کربلایی

مدیرعامل سازمان اتکا

¹ Data Mining

² Machine Learning Method

فهرست مطالب

فصل ۱_ مقدمه‌ای بر داده‌کاوی	۱۰
۱-۱_ داده‌کاوی و کشف دانش در پایگاه داده‌ها	۱۱
۲-۱_ فرآیند کشف دانش	۱۳
۳-۱_ عملکردهای داده‌کاوی	۱۷
۴-۱_ کاربردهای داده‌کاوی	۱۸
۵-۱_ انواع حوزه‌های اطلاعات	۱۹
۶-۱_ انواع داده	۲۳
۷-۱_ داده، اطلاعات و دانش	۲۸
۸-۱_ انواع ابزارها و زبان‌های مورد استفاده در داده‌کاوی	۳۳
۹-۱_ چالش‌های تحقیقات و کاربردهای داده‌کاوی	۳۴
۱۰-۱_ انبار داده و پایگاه داده تراکنشی	۳۸
فصل ۲_ مقدمه‌ای بر زبان R	۴۲
۱-۲_ مقدمات و توابع ابتدایی	۴۳
۲-۲_ مزایای منحصربه‌فرد نرم‌افزار R	۴۴
۳-۲_ قراردادهای ابتدایی	۴۸
۴-۲_ ساختار داده‌ها در R	۵۲
۵-۲_ محاسبات ریاضی در R	۶۱
۶-۲_ نوشتن توابع در R	۷۰
۷-۲_ آمار توصیفی و نمودارها	۷۳
فصل ۳_ تحلیل اکتشافی داده‌ها	۸۲
۱-۳_ آماده‌سازی داده‌ها برای داده‌کاوی	۸۳
۲-۳_ پیش‌پردازش داده‌ها	۸۳
۳-۳_ تحلیل اکتشافی داده‌ها در زبان R	۹۳

فصل ۴ خوشه‌بندی ۱۲۲

- ۱-۴ خوشه‌بندی ۱۲۳
- ۲-۴ تفاوت خوشه‌بندی و دسته‌بندی ۱۲۳
- ۳-۴ فرآیند خوشه‌بندی ۱۲۴
- ۴-۴ کاربردهای خوشه‌بندی ۱۲۵
- ۵-۴ اعتبارسنجی خوشه‌ها ۱۲۶
- ۶-۴ شاخصهای اعتبارسنجی ۱۲۷
- ۷-۴ روش‌های خوشه‌بندی ۱۲۸
- ۸-۴ روش افزایش ۱۳۳
- ۹-۴ روش خوشه‌بندی سلسله‌مراتبی ۱۳۶
- ۱۰-۴ روش‌های مبتنی بر چگالی ۱۴۰
- ۱۱-۴ ارزیابی خوشه‌بندی با معیار سیلوئت ۱۴۱
- ۱۲-۴ خوشه‌بندی در نرم‌افزار R ۱۴۲

فصل ۵ دسته‌بندی و پیش‌بینی ۱۵۲

- ۱-۵ دسته‌بندی ۱۵۳
- ۲-۵ تفاوت دسته‌بندی و خوشه‌بندی ۱۵۵
- ۳-۵ بیز ساده ۱۵۵
- ۴-۵ دسته‌بندی بر مبنای نزدیک‌ترین همسایگی ۱۵۶
- ۵-۵ شبکه‌های عصبی ۱۵۷
- ۶-۵ درخت تصمیم ۱۵۸
- ۷-۵ پیش‌بینی ۱۶۰
- ۸-۵ قواعد انجمنی ۱۶۱
- ۹-۵ دسته‌بندی در نرم‌افزار R ۱۶۳

فصل ۶ سری‌های زمانی ۱۸۸

- ۱-۶ تعریف سری زمانی ۱۸۹
- ۲-۶ سری‌های زمانی در داده‌کاوی ۱۹۲

۱۹۴	۳-۶_ اجزای سری‌های زمانی
۱۹۶	۴-۶_ شناسایی ، تجزیه و حذف اجزای سری‌های زمانی
۱۹۷	۵-۶_ سری زمانی در نرم‌افزار R
۲۱۲	فصل ۷_ تحلیل شبکه‌های اجتماعی
۲۱۳	۱-۷_ تحلیل شبکه‌های اجتماعی
۲۱۷	۲-۷_ انواع مرکزیت و شاخص‌های اصلی در تحلیل شبکه
۲۲۳	۳-۷_ اجتماع‌یابی در شبکه
۲۲۴	۴-۷_ هم‌ارزی ساختاری
۲۲۶	۵-۷_ تحلیل شبکه‌ای در نرم‌افزار R
۲۶۰	فصل ۸_ متن‌کاوی
۲۶۱	۱-۸_ متن‌کاوی
۲۶۲	۲-۸_ پیش‌پردازش متون
۲۷۱	۳-۸_ خوشه‌بندی متون
۲۸۳	۴-۸_ کدهای متن‌کاوی
۲۹۴	منابع

فصل اول

مقدمه ای بر داده کاوی

۱-۱- داده کاوی و کشف دانش در پایگاه داده‌ها

«کشف دانش و داده کاوی»^۱ یک حوزه جدید میان‌رشته‌ای^۲ و در حال رشد است که حوزه‌های مختلفی همچون پایگاه داده، آمار، یادگیری ماشین^۳ و سایر زمینه‌های مرتبط را باهم تلفیق کرده تا اطلاعات و دانش ارزشمند نهفته در حجم بزرگی از داده‌ها را استخراج نماید. با رشد سریع کامپیوتر و استفاده از آن در دو دهه اخیر تقریباً همه سازمان‌ها حجم عظیمی داده در پایگاه داده خود ذخیره کرده‌اند. این سازمان‌ها به فهم این داده‌ها و یا کشف دانش مفید از آن‌ها نیاز دارند. به عبارت دیگر، هدف کشف دانش و داده کاوی یافتن الگوها و یا مدل‌های جالب موجود در پایگاه داده‌ها است که در میان حجم عظیمی از داده‌ها مخفی هستند.

برخی از تعاریف متداول کشف دانش و داده کاوی به شرح زیر می‌باشند:

۱. تحلیل داده‌های توصیفی کامپیوتری، در مجموعه‌های بزرگ و پیچیده داده‌ها (Friedman, ۱۹۹۷).
۲. تحلیل ثانوی^۴ مجموعه‌های بزرگ داده (Hand et al., ۲۰۰۰).
۳. پرس و جوی الگو در پایگاه داده‌ها (Imielinski and Virmani, ۱۹۹۹). این دیدگاه بر مشابهت جستجوی الگوها با پرس‌وجوهای انجام‌شده توسط سیستم‌های مدیریت پایگاه داده‌ها تأکید می‌کند.
۴. کشف دانش، فرایند تشخیص الگوهای معتبر، نو، مفید و نهایتاً قابل درک در داده‌ها است (Friedman, ۱۹۹۷).
۵. داده کاوی، آمار در مقیاس و سرعت است (Pregibon, ۱۹۹۹).

^۱ - Knowledge Discovery and Data Mining (KDD)

^۲ - Interdisciplinary

^۳ - Machine Learning

^۴ - ثانوی به این معنا است که منظور اصلی کسب و کار از جمع‌آوری پایگاه داده‌ها، کشف دانش نبوده است.

۶. داده کاوی یک حوزه میان رشته‌ای و با رشد سریع است که حوزه‌های مختلفی همچون پایگاه داده، آمار، یادگیری ماشین و سایر زمینه‌های مرتبط را باهم تلفیق کرده است تا اطلاعات و دانش ارزشمند نهفته در حجم بزرگی از داده‌ها را استخراج نماید (ACM, ۲۰۰۶).

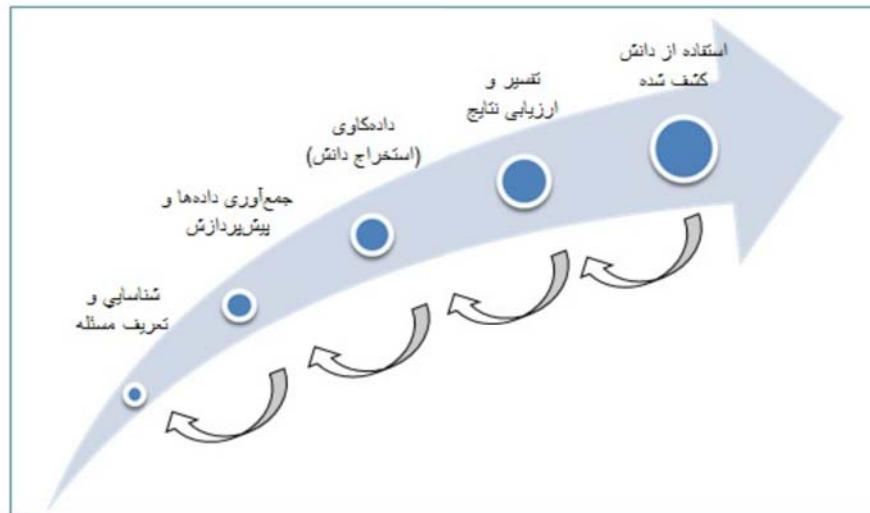
۷. داده کاوی، اکتشاف و تحلیل حجم زیادی از داده‌ها برای کشف الگوها و قواعد معنادار است. فرایند داده کاوی گاهی کشف دانش نیز نامیده می‌شود (Han and Kamber, ۲۰۰۶).

شاپیرو (Shapiro, ۲۰۰۰) که در سال ۱۹۸۹ واژه KDD را ابداع کرده است می‌گوید: «واژه KDD در جامعه هوش مصنوعی و یادگیری ماشین متداول شد. ولیکن محققان پایگاه داده‌ها در ارتباط بیشتری برای گفتمان با اهل کسب و کار و رسانه‌ها بودند و واژه داده کاوی در اخبار کسب و کار متداول شد.» داده کاوی واژه‌ای قدیمی‌تر از KDD است که در جامعه تحلیل داده‌های آمارمحور، ابداع شده است.

باید توجه داشت که در زبان فارسی فعل «کاویدن» هم برای داده‌ها و هم برای دانشی که از داده‌ها استخراج می‌شود، قابل استفاده است. یعنی اصطلاح کاویدن بر روی داده‌ها و کاویدن بر روی دانش، هر دو درست است. در این تحقیق از منظور اول استفاده می‌کنیم. یعنی فعل «کاویدن» را برای داده‌ها و برای دانش از فعل «کشف» استفاده کرده و واژگان داده کاوی و کشف دانش را به کار می‌بریم.

بر اساس دیدگاهی که داده کاوی را بخشی از فرایند کشف دانش می‌دانند، کشف دانش شامل مراحل متعددی مطابق با شکل ۱-۱ است (غضنفری و همکاران، ۱۳۸۷).

۱-۲- فرایند کشف دانش



شکل ۱-۱: فرایند کشف دانش

اولین قدم: درک حوزه کاربرد موردنظر و نحوه رابطه بندی مسئله است. این قدم به‌وضوح پیش‌نیاز استخراج دانش مفید و انتخاب روش‌های داده‌کاوی مناسب در قدم سوم، با توجه به هدف کاربرد و طبیعت داده‌ها است.

قدم دوم: جمع‌آوری و پیش‌پردازش داده‌ها^۱ شامل انتخاب منابع داده، حذف نقاط پرت^۲ یا مغشوش^۳، طرز برخورد با داده‌های مفقوده^۴ و تبدیل^۵ و یا گسسته سازی^۶ و کاهش^۷ داده‌ها است. این مرحله معمولاً در کل فرایند KDD بیشترین زمان را می‌برد.

^۱ - Preprocess

^۲ - Outliers

^۳ - Noise

^۴ - Missing Data

^۵ - Transformation

^۶ - Discrimination

^۷ - Reduction

قدم سوم: داده کاوی است که هدف آن استخراج الگوها و یا مدل های مخفی در داده ها است. مدل را می توان به شکل زیر بیان نمود: «مدل یک تصویر کلی^۱ از ساختاری است که روابط سامانمند میان داده ها را بیان می کند» در مقابل، «یک الگو، ساختاری محلی است که فقط به چند متغیر محدود و تعدادی مشاهده مرتبط است.»

روش های اصلی داده کاوی به دو دسته توصیفی^۲ و پیشبینانه تقسیم می شوند. دسته بندی از مهم ترین روش های توصیفی می باشد.

قدم چهارم: تفسیر (یا پس پردازش^۳) دانش کشف شده است. این تفسیر، عملاً توصیفی یا پیشبینانه است که دو هدف اصلی سیستم های اکتشافی می باشند. تجربه نشان داده است که همیشه الگوها یا مدل های کشف شده از داده ها، مفید و جالب نیستند بنابراین فرایند KDD فرایندی تکراری^۴ می باشد. یک راه استاندارد ارزیابی، تقسیم داده ها به دو مجموعه برای آموزش و آزمون است. می توان این فرایند را بارها با تقسیمات مختلف تکرار کرد و میانگین نتایج را برای تخمین عملکرد قواعد در نظر گرفت.

قدم پنجم: استفاده عملی از دانش کشف شده است. برخی اوقات می توان از دانش کشف شده بدون کامپیوتری کردن آن استفاده کرد. در مواقع دیگر کاربر انتظار دارد دانش کشف شده از طریق یک برنامه کامپیوتری به کار گرفته شود. بی شک به کارگیری عملی نتایج فرایند کشف دانش هدف نهایی این فرایند است.

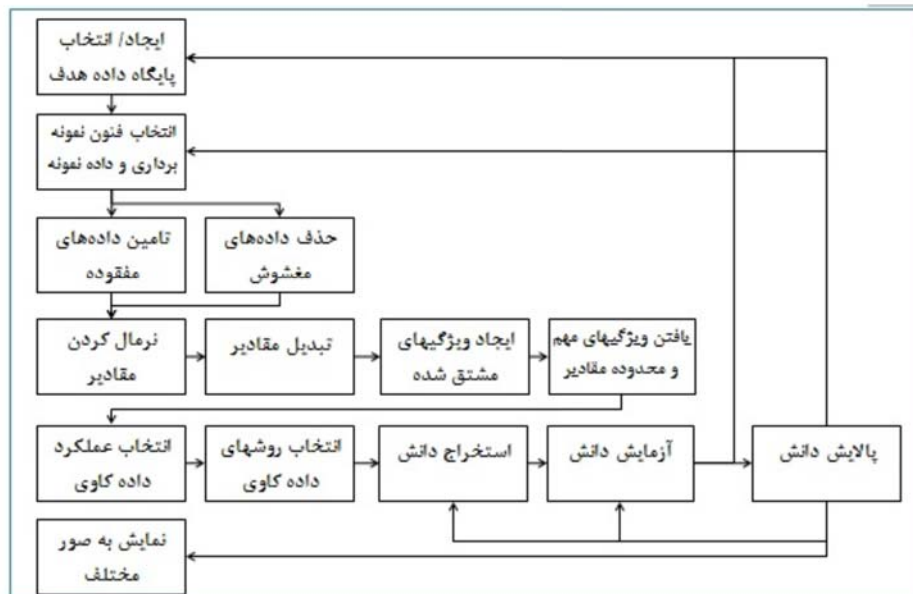
^۱ - Global Representation

^۲ - Descriptive

^۳ - Post-Process

^۴ - Iterative

جزئیات وظایف مربوط به فرایند KDD که در شکل زیر آمده، به شرح ذیل است (Ho and Dam, ۲۰۰۵):



شکل ۱-۲: وظایف فرایند کشف دانش (Ho and Dam, ۲۰۰۵)

جزئیات وظایف مربوط به فرایند KDD شامل مراحل ذیل است:

۱. ایجاد مجموعه داده های هدف: انتخاب مجموعه داده ها یا تمرکز روی زیرمجموعه ای از متغیرها یا نمونه های داده که قرار است روی آن ها اکتشاف انجام شود، ایجاد مجموعه داده های هدف نامیده می شود.
۲. پیش پردازش یا پاک سازی داده^۱: عملیات مقدماتی مثل حذف نویز یا نقاط پرت، جمع کردن اطلاعات لازم برای مدل کردن یا مقابله با نویز، تصمیم گیری در مورد چگونگی رفتار با داده های مفقوده، در نظر گرفتن توالی زمانی و تغییرات شناخته شده در اطلاعات، پاک سازی داده ها نامیده می شود.

^۱ - Data Cleaning Preprocessing

۳. کاهش داده‌ها و تصویر کردن آن‌ها: یافتن مشخصه‌های مفید برای نمایش داده بسته به هدف وظیفه و استفاده از روش‌های کاهش بُعد یا تبدیل برای کاهش تعداد مؤثر متغیرهای موردنظر یا پیدا کردن نمود مناسب و معادل داده‌ها، کاهش داده‌ها نامیده می‌شود.
۴. انتخاب عملکرد داده کاوی: تصمیم‌گیری در مورد هدف فرایند KDD که می‌تواند دسته‌بندی، رگرسیون، خوشه‌بندی یا غیره باشد. عملکردهای مختلف الگوریتم داده کاوی به‌طور مفصل در بخش‌های بعدی تشریح می‌شوند.
۵. انتخاب روش‌های داده کاوی: این گام شامل انتخاب روش‌های جستجوی الگوها در داده‌ها بوده و شامل انتخاب مدل‌ها و پارامترهای مناسب تطابق یک روش داده کاوی خاص با معیارهای کلی فرایند KDD است. برای مثال مدل مورداستفاده برای داده‌های طبقه‌ای با مدل‌های مورداستفاده برای داده‌های عددی متفاوت می‌باشد. به‌علاوه ممکن است کاربر نهایی علاقه‌مند به درک مدل بوده و به قابلیت‌های پیش‌بینی آن علاقه‌ای نداشته باشد.
۶. داده کاوی برای استخراج الگوها/مدل‌ها: در این گام به جستجوی الگوهای موردنظر به یک یا چند شکل خاص (قواعد یا درختان دسته‌بندی، رگرسیون، خوشه‌بندی و مانند آن) پرداخته می‌شود. کاربر با انجام درست مراحل قبل می‌تواند کمک بسیاری به روش داده کاوی کند.
۷. تفسیر و ارزیابی الگوها/مدل‌ها: لازم است الگوها و مدل‌های مختلف به‌منظور استفاده بعدی مورد ارزیابی و تفسیر قرار گیرند.
۸. پالایش یا تثبیت^۱ دانش کشف شده: ترکیب این دانش با سیستم اجرایی یا حداقل مستندسازی و گزارش آن به گروه‌های علاقه‌مند، تثبیت دانش نامیده می‌شود. این کار شامل بررسی و حل تضادهای^۲ بالقوه این دانش با

^۱ - Consolidation

^۲ - Conflicts

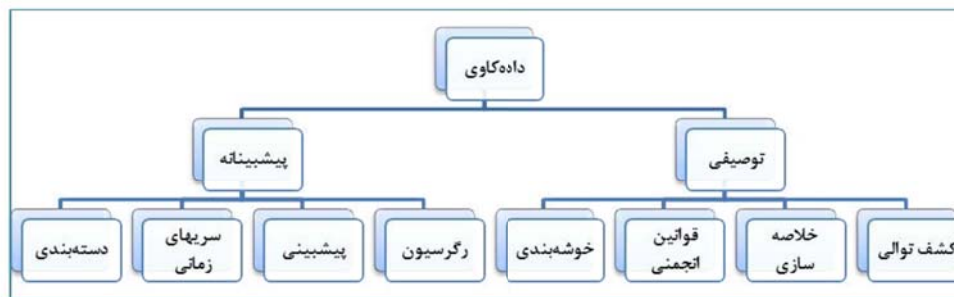
دانش‌های مورد قبول (یا کشف شده) پیشین می‌باشد. ممکن است میان هر قدم و قدم قبلی آن عملاً نوعی تکرار رخ دهد.

۱-۳- عملکردهای داده کاوی

همان‌طور که در تعریف داده کاوی گفته شد، داده کاوی یک حوزه میان‌رشته‌ای است که حوزه‌های مختلفی همچون پایگاه داده، آمار، یادگیری ماشین و سایر زمینه‌های مرتبط را باهم تلفیق می‌کند (Han and Kamber, ۲۰۰۶).

روش‌های اصلی داده کاوی دو دسته می‌باشند: توصیفی^۱ و پیش‌بینانه^۲. وظایف توصیفی خواص عمومی داده‌ها را مشخص می‌کنند. هدف از توصیف، یافتن الگوهایی در مورد داده‌هاست که برای انسان قابل تفسیر باشد. وظایف پیش‌بینانه به منظور پیش‌بینی رفتارهای آینده آن‌ها استفاده می‌شوند. منظور از پیش‌بینی به کارگیری چند متغیر یا فیلد در پایگاه داده برای پیش‌بینی مقادیر آینده یا نا شناخته دیگر متغیرهای مورد علاقه است. عملکردهای داده کاوی در شکل زیر نشان داده شده‌اند (Dunham, ۲۰۰۳).

متداول‌ترین روش توصیفی، خوشه‌بندی^۳ و متداول‌ترین روش پیش‌بینانه، دسته‌بندی^۴ می‌باشد.



(Dunham, ۲۰۰۳)

شکل ۱-۳: عملکردهای داده کاوی

^۱ - Descriptive

^۲ Predictive

^۳ Clustreing

^۴ Classification

۴-۱- کاربردهای داده کاوی

ما در عصری به سر می‌بریم که از آن به‌عنوان عصر اطلاعات نام برده می‌شود. در این عصر اطلاعات، ما معتقدیم که اطلاعات به قدرت و موفقیت منجر می‌شود و در همین راستا در زمینه فناوری‌هایی مانند رایانه‌ها، ماهواره‌ها و ...، بخش عظیمی از اطلاعات را جمع‌آوری نموده‌ایم. در ابتدای ظهور رایانه‌ها و وسایل دیجیتال، جمع‌آوری و مرتب‌سازی انواع داده‌ها و محاسبات با کمک رایانه‌ها و مرتب‌سازی حجم زیادی از اطلاعات مطرح بوده ولی محاسبه حجم بزرگ از داده‌ها که در ساختارهای متفاوتی ذخیره شده‌اند بسیار دشوار بود. این بی‌نظمی داده‌ها منجر به ایجاد پایگاه داده‌های ساختارمند و سیستم‌های مدیریت پایگاه داده شده است. وجود سیستم‌های کارای مدیریت پایگاه‌های داده از جمله مهم‌ترین ابزارهای مدیریتی در مواجهه با حجم زیادی از داده‌ها می‌باشد که در برهه خاص زمانی خودش که نیاز به استخراج بخش مشخصی از داده‌ها به‌صورت کارا و بهینه هستیم.



شکل ۴-۱: حوزه‌های مفهومی داده کاوی

ازدیاد این سیستم‌های مدیریت پایگاه‌های داده کمک به جمع‌آوری حجم زیادی از انواع اطلاعات نموده است. امروزه ما اطلاعات زیادی داریم که فراتر از توان مدیریت ما بر آن‌هاست. از تعاملات تجاری تا داده‌های علمی، تا هر چیز دیگری که برای تصمیم‌گیری به ما کمک می‌نماید. در مواجهه با این حجم بزرگ از داده‌ها، ما نیازهای جدیدی را ایجاد نموده‌ایم که به ما در تصمیمات مدیریتی

بهبتر کمک می‌رساند. این نیازها شامل خلاصه کردن داده‌ها، استخراج جوهر اطلاعات ذخیره شده و کشف الگوهای پنهان در سطرهای داده‌ها می‌باشد. در دهه‌های پیشین مثال‌های کمی از اکتشاف داده‌های واقعی وجود داشت. از جمله موارد قابل توجه می‌توان به کشف قوانین طیف‌سنج جرمی (Buchanan and Mitchell ۱۹۷۸)، قوانین تشخیصی جدید برای بیماری‌های سویا (Michalski ۱۹۸۰) and Chilausky و عوارض جانبی مواد مخدر در یک پایگاه داده مربوط به یک بیمار روماتیسم (Blum ۱۹۸۲) اشاره نمود.

از آن زمان تاکنون، روش‌های اکتشاف داده در زمینه‌های بسیار بیشتری اعمال شده است که در ادامه به برخی از این حیطه‌ها اشاره خواهیم نمود. هرچند مواردی که اشاره خواهد شد کامل نیست، اما مثال‌هایی است که نشان‌دهنده انواع حیطه‌هایی است که اکتشاف داده در آن امکان‌پذیر می‌باشد. در شکل ۱-۴ یک تقسیم‌بندی از حوزه‌های مفهومی داده کاوی را مشاهده می‌نمایید.

۱-۵- انواع حوزه‌های اطلاعات

ما تعداد زیادی از داده‌ها را جمع‌آوری نمودیم از اندازه‌گیری‌های ساده عددی و سندهای متنی، تا اطلاعات پیچیده‌تری همچون داده‌های فضایی، شبکه‌های چندرسانه‌ای و اسناد ابرمتنی. اینجا یک لیست غیر منحصراً شامل تعداد متنوعی از انواع اطلاعات جمع‌آوری شده که به‌صورت دیجیتال در پایگاه‌های داده و فایل‌های ساده ذخیره گردآوری شده است.

- معاملات تجاری^۱: هر معامله‌ای در صنایع تجاری معمولاً در حافظه‌ها تا ابد باقی می‌ماند. معاملات معمولاً به زمان‌ها مرتبط هستند و می‌تواند معاملات بین کسب و کار مانند خرید، مبادلات، بانکداری، سهام، و غیره باشد. فروشگاه‌های بزرگ که دارای استفاده گسترده از بارکدها می‌باشند، میلیون‌ها تعامل روزانه در خود دارند که نشان‌دهنده داده‌هایی با حجم ترابایت هستند. حجم ذخیره مشکل عمده نیست، به‌عنوان مثال قیمت هارددیسک به‌طور مداوم پایین می‌آید، اما استفاده کار از داده‌ها

^۱ Business transactions

در یک زمان معقول و موجه برای تصمیم‌گیری رقابتی، قطعاً مهم‌ترین مسئله برای حل شدن در تجارت‌هاست که موجب تلاش برای باقی ماندن در عرصه رقابت گسترده جهانی است.

- داده‌های علمی^۱ : چه در یک آزمایشگاه شتاب‌دهنده ذرات هسته‌ای ذرات در سوئیس، یا در یک مرکز مطالعاتی در جنگل کانادا، یا در جمع‌آوری داده‌ها در مورد فعالیت‌های اقیانوسی کوه یخ قطب جنوب، و یا در یک دانشگاه آمریکایی که تحقیقاتی پیرامون روانشناسی انسان انجام می‌دهد، جامعه ما در حال گردآوری مقادیر عظیم داده‌های علمی ست که نیاز به تجزیه و تحلیل دارد. متأسفانه، ما می‌توانیم داده‌های جدید بیشتری را با سرعت بیشتری ضبط و ذخیره نماییم تا اینکه بخواهیم داده‌های قبلی که ذخیره نموده‌ایم تجزیه و تحلیل نماییم.

- داده‌های شخصی و پزشکی^۲ : زمانی که دولت اشخاص را سرشماری می‌کند و فایل‌های مشتریان را دریافت می‌کند، میزان بسیار زیادی از اطلاعات درباره افراد و گروه‌ها به صورت مداوم جمع‌آوری می‌گردد. دولت‌ها، سازمان‌ها و شرکت‌ها همچون بیمارستان‌ها، مقدار بسیار مهمی از داده‌های اشخاص را برای کمک به مدیریت منابع انسانی، فهم بهتر بازار یا کمک ساده‌تر به مشتریان جمع‌آوری نموده‌اند. صرف‌نظر از مسائل مربوط به حریم شخصی این نوع از داده‌ها معمولاً آشکار می‌گردد، جمع‌آوری می‌گردد، استفاده می‌شود و حتی به اشتراک گذاشته می‌شود. زمانی که این داده‌ها با اطلاعات دیگر مورد انطباق قرار بگیرد موجب آشکار شدن رفتار و علایق مشتریان می‌گردد.

- ویدئوهای مراقبتی و تصاویر^۳ : با کاهش عجیب قیمت دوربین‌های ویدئویی، این دوربین‌ها در همه‌جا یافت می‌گردند. نوارهای ویدئویی از

^۱ Scientific data

^۲ Medical and personal data

^۳ Surveillance video and pictures

دوربین‌های مراقبتی معمولاً بازیافت می‌گردند و بنابراین محتوای آن از بین می‌رود. معمولاً تمایل وجود دارد که نوارها ذخیره گردد و برای استفاده و تحلیل در آینده به صورت دیجیتال تبدیل گردد.

- حسگرهای ماهواره‌ای^۱: تعداد قابل توجهی از ماهواره‌ها پیرامون جهان وجود دارند، برخی از این ماهواره‌ها به صورت خاص در یک منطقه متمرکز هستند و برخی دیگر دور زمین در حال چرخش‌اند. اما همه آن‌ها جریان بدون توقفی از داده‌ها را از سطح زمین ارسال می‌نمایند. ناسا که تعداد زیادی از این ماهواره‌ها را کنترل می‌نماید داده‌های بیشتری را هر ثانیه دریافت می‌کند تا همه پژوهشگران و مهندسان ناسا بتوانند از روی آن کپی نمایند. تعداد زیادی از تصاویر و داده‌ها به محض این‌که دریافت می‌گردند به صورت عمومی قرار داده می‌شوند تا دیگر پژوهشگران نیز بتوانند این داده‌ها را تجزیه و تحلیل نمایند.
- ورزش‌ها^۲: جامعه ما مقدار بسیار عظیمی از داده و آمارهای مربوط به ورزش‌ها، بازیکنان و ورزشکاران را جمع‌آوری می‌نماید. از امتیازات هاکی، پاس‌های بسکتبال و امتیازات اتومبیلرانی، تا رکوردهای شنا، ضربات مشت‌زنان و حرکات‌های شطرنج‌بازان، همه این داده‌ها ذخیره می‌گردند. روزنامه‌نگاران و گزارشگران از این اطلاعات برای گزارش دادن استفاده می‌نمایند اما ورزشکاران و تمرین‌کنندگان می‌خواهند از این اطلاعات برای بهبود عملکرد خود و شناخت بهتر رقیبان استفاده نمایند.
- رسانه‌های دیجیتال^۳: افزایش تعداد اسکنرهای ارزان قیمت، دوربین‌های ویدئویی دیجیتال و رومیزی یکی از علت‌های بهره‌برداری بیشتر از مخازن رسانه‌های دیجیتال است. به علاوه تعداد زیادی ایستگاه‌های رادیویی، کانال‌های تلویزیونی و استودیوهای فیلم مجموعه‌های صوتی و تصویری خود را به صورت دیجیتال درمی‌آورند تا بتوانند دارایی‌های چندرسانه‌ای خود را بهتر مدیریت نمایند. شبکه‌های رسانه‌ای مختلف

^۱ Satellite sensing

^۲ Games

^۳ Digital media

قبلاً تبدیل مجموعه‌های بزرگ بازی خود را به صورت دیجیتال آغاز نموده‌اند.

- داده‌های مهندسی نرم‌افزار^۱: چندین سیستم رایانه‌ای طراح وجود دارد تا طراحی ساختمان‌ها و مهندسان برای مصورسازی اجزا و مدارهای سیستمی‌شان را مصورسازی نمایند. این سیستم‌ها حجم عظیمی از داده‌ها را ایجاد می‌نمایند. به علاوه مهندسی نرم‌افزار یک منبع قابل توجه داده‌های مشابه همچون کد، توابع کتابخانه‌ای، اشیا و ... که ابزارهای قدرتمندی را برای پشتیبانی و مدیریت نیازمند است را تشکیل می‌دهد.
- جهان‌های مجازی^۲: تعداد زیادی از نرم‌افزارها وجود دارد که امکان استفاده از فضاهای مجازی چندبعدی را مهیا می‌سازد. این فضاها و اشیایی که دربرمی‌گیرند به وسیله زبان خاصی همچون VRML شرح داده می‌شوند. این فضاهای مجازی به صورتی شرح داده می‌شود که قابلیت به اشتراک‌گذاری نیز داشته باشد. تعداد قابل توجهی از مخازن مربوط به اشیا و فضای واقعیت‌های مجازی در دسترس می‌باشد. مدیریت این مخازن از آنجایی که این مخازن دارای امکان جستجوی محتوای محور می‌باشد و بازیافت از این مخازن هنوز به صورت جستجوی موضوعی می‌باشد، زمانی که اندازه این مجموعه‌ها به سمت افزایش گزارش‌های متنی و پیام‌های ایمیلی می‌رود، بسیاری از ارتباطات درون و بین شرکت‌ها و موسسه‌های تحقیقاتی و حتی بین افراد شخصی مبتنی بر گزارش‌های و ایمیل‌هایی با فرمت متنی می‌باشد. این پیام‌ها به صورت منظم به فرمت دیجیتال برای استفاده‌های آینده و مرجع دهی کتابخانه‌های دیجیتال نیرومند ذخیره‌سازی می‌گردد.
- مخازن شبکه جهانی وب^۳: از زمان ایجاد شبکه جهانی وب در سال ۱۹۹۳ ایجاد گردیده است، اسنادی از انواع فرمت‌ها، محتوا و شرح جمع‌آوری گردیده است و به وسیله ابرمتن‌های به هم پیوسته آن را

^۱ CAD and Software engineering data

^۲ Virtual Worlds

^۳ The World Wide Web repositories

به صورت بزرگ‌ترین مخزن داده که تاکنون ساخته شده است تبدیل می‌نماید. با وجود این طبیعت غیرساختاریافته و پویا، این ویژگی‌های ناهمگونش و با وجود افزونگی و ناسازگاری‌اش، شبکه جهانی وب مهم‌ترین مجموعه داده است که به صورت مرتب به عنوان یک مرجع مورد استفاده قرار می‌گیرد، به خاطر تنوع وسیعی از عنوان‌ها و حیطه‌هایی که پوشش می‌دهد و توسعه‌دهندگان و سهم نامحدودی که دارد، بسیاری معتقدند که شبکه جهانی وب تلفیقی از دانش‌های متنوع انسانی می‌باشد.

۱-۶- انواع داده

درواقع، داده کاوی تنها محدود به نوع خاصی از داده‌ها نمی‌شود. داده کاوی این امکان را دارد که بر روی هر نوع از مخازن اطلاعاتی اعمال شود. هرچند که الگوریتم‌ها و روش‌ها ممکن است زمانی که بر روی انواع مختلف داده‌ها اعمال می‌شوند، متفاوت از یکدیگر باشند. داده کاوی می‌تواند بر روی انواع پایگاه‌های داده شامل پایگاه‌های داده رابطه‌ای، پایگاه‌های داده‌ای شیء‌گرا، انبار داده‌ها، پایگاه‌های داده تراکنشی، مخازن بدون ساختار و نیمه ساخت یافته مانند شبکه جهانی وب، پایگاه‌های داده فضایی، پایگاه‌های داده چندرسانه‌ای، پایگاه‌های داده‌ای سری زمانی و پایگاه‌های داده متنی اعمال گردد. در ادامه به برخی مثال‌های همراه با جزئیات اشاره می‌نماییم. (Osmar, ۱۹۹۹)

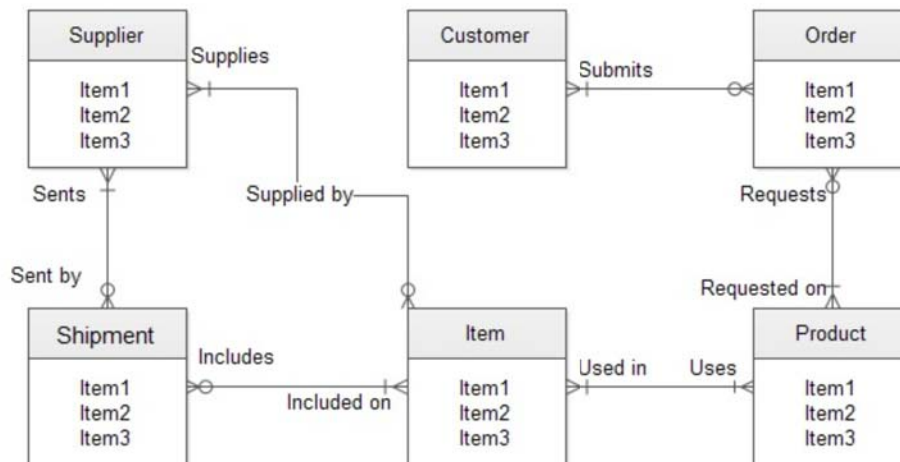
- **فایل‌های تخت**^۱: فایل‌های تخت رایج‌ترین منابع داده‌ای برای الگوریتم‌های داده کاوی به خصوص در سطح تحقیقاتی می‌باشند. فایل‌های تخت داده‌های ساده‌ای در قالب متن یا فرمت دودویی هستند که به وسیله الگوریتم‌های داده کاوی با ساختار مشخصی شناخته

^۱ Flat files

می‌شوند. داده‌ها در این نوع فایل‌ها می‌توانند تراکنش‌ها، داده‌های مربوط به سری‌های زمانی، اندازه‌گیری‌های علمی و موارد مشابه باشند.

- **پایگاه‌های داده‌ای رابطه‌ای**^۱: به‌صورت خلاصه، یک پایگاه داده رابطه‌ای متشکل از مجموعه‌ای از جداول است که دربرگیرنده مقادیر ویژگی‌های مشخصه‌ها یا مقادیر ویژگی‌های مربوط به روابط مشخصه‌ها می‌باشد. جداول سطرها و ستون‌هایی دارند، ستون‌ها نشان‌دهنده صفات و سطرها هرکدام از چندتایی‌ها را شامل می‌گردد. یک چندتایی در جدول ارتباطی با یک شیء یا رابطه بین اشیاء مطابقت دارد و با مجموعه‌ای از مقادیر صفات با یک کلید مشخص نمایش داده می‌شود. این ارتباطات زیرمجموعه یک نمونه پایگاه داده ارتباطی می‌باشد. متداول‌ترین زبان پرس‌وجو برای پایگاه‌های داده ارتباطی زبان sql می‌باشد که اجازه فراخوانی و دست‌کاری داده‌های ذخیره‌شده در جداول را می‌دهد که شامل برخی توابع محاسباتی همچون محاسبه میانگین، جمع، حداقل، حداکثر و ... می‌باشد. الگوریتم‌های داده‌کاوی که از پایگاه‌های داده ارتباطی استفاده می‌نمایند قابلیت‌های بیشتری نسبت به الگوریتم‌هایی دارند که برای فایل‌های تخت نوشته می‌شوند. از آنجایی که داده‌کاوی می‌تواند مزایایی از sql برای انتخاب داده، تبدیل داده‌ها و تثبیت داده‌ها داشته باشد با استفاده از این زبان می‌توان برای پیش‌بینی، مقایسه و تشخیص انحراف استفاده نمود.

^۱ Relational Databases

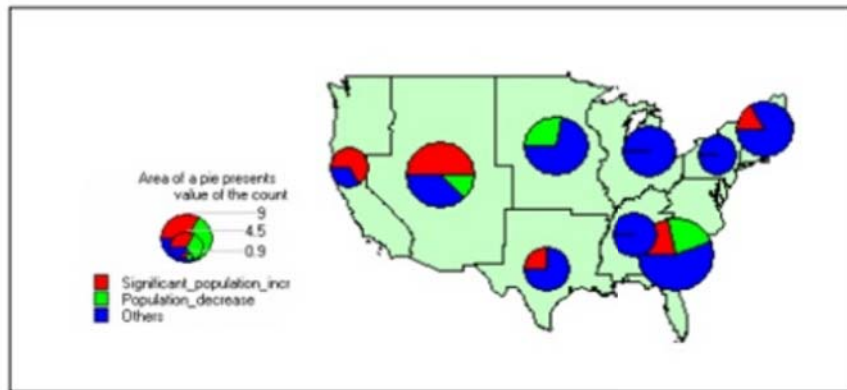


شکل ۱-۵: پایگاه داده ارتباطی

- پایگاه‌های داده‌ای چندرسانه‌ای^۱:** این پایگاه‌های داده چندرسانه‌ای شامل ویدئو، تصاویر، صوت و متن می‌باشند. این داده‌ها می‌توانند به صورت مدل‌های شیء-رابطه گسترش یافته یا پایگاه‌های داده شیء-گرا ذخیره گردند یا به صورت یک سیستم سندی ساده گردند. چندرسانه‌ای‌ها معمولاً به واسطه خاصیت چندبعدی بودن شناخته می‌شوند که داده کاوی را با چالش روبرو می‌نمایند. داده کاوی از مخازن چندرسانه‌ای ممکن است نیازمند دید محاسباتی، گرافیک رایانه‌ای و تفسیر تصویری و متدهای پردازش زبان‌های طبیعی باشند.
- پایگاه‌های داده فضایی^۲:** پایگاه‌های داده فضایی پایگاه‌های داده‌ای هستند که علاوه بر داده‌های معمول، اطلاعات جغرافیایی همانند نقشه‌ها یا موقعیت‌های مکانی را نیز ذخیره می‌نمایند. این نوع داده‌های فضایی چالش‌های جدیدی را برای الگوریتم‌های داده کاوی ایجاد نموده‌اند.

^۱ Multimedia Databases

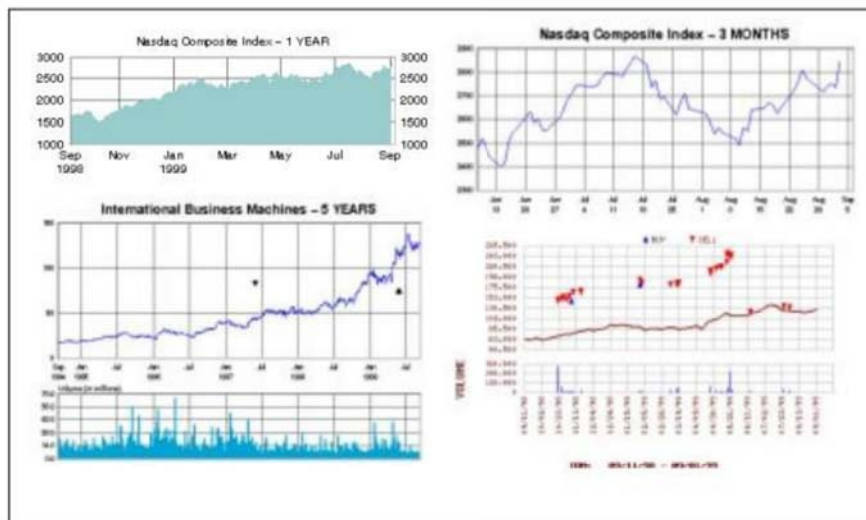
^۲ Spatial Databases



شکل ۱-۶: پایگاه داده فضایی

- **پایگاه‌های داده سری‌های زمانی^۱:** پایگاه‌های داده سری‌های زمانی شامل داده‌های مرتبط با زمان مانند داده‌های مرتبط با بازار سهام یا فعالیت‌های انجام‌شده می‌باشد. این پایگاه‌های داده معمولاً حاوی یک جریان مستمر داده‌های ورودی می‌باشد که در برخی مواقع نیازمند تحلیل‌های آنی می‌باشد. داده‌کاوی در چنین پایگاه‌های داده‌ای معمولاً شامل شناخت روندها و همبستگی بین تغییرات مربوط به متغیرهای مختلف همانند پیش‌بینی روندها و حرکت متغیرها در زمان می‌باشد. شکل زیر برخی مثال‌های مربوط به سری‌های زمانی را شامل می‌شود.

^۱ Time-Series Databases



شکل ۱-۷: داده‌های سری زمانی

- **پیکره‌های متنی و وبی**^۱: شبکه جهانی اینترنت بزرگ‌ترین مخزن ناهمگون و پویای داده‌هاست که در دسترس همگان می‌باشد. تعداد بسیار زیادی از نویسندگان و منتشرکنندگان به صورت مستمر در حال توسعه و دگر دیسی این شبکه می‌باشند و تعداد قابل ملاحظه‌ای از کاربران روزانه به این مخزن دسترسی دارند. داده‌ها در شبکه جهانی وب به صورت اسناد به هم متصل شده سازمان‌دهی شده‌اند. این اسناد شامل متن، صدا، ویدئو و حتی نرم‌افزارها می‌باشند. به صورت مفهومی شبکه جهانی وب، از سه جز اصلی تشکیل گردیده است: محتوای وب که شامل دربرگیرنده اسناد در دسترس می‌باشد، ساختار وب که ابرمتن‌ها و ارتباط بین اسناد را دربرمی‌گیرد و استفاده از وب که زمان و چگونگی دسترسی به منابع را توصیف می‌کند. یک بعد چهارم نیز می‌تواند اضافه گردد که شامل خصلت پویایی یا متغیر بودن اسناد را بیان می‌نماید. داده‌کاوی در شبکه جهانی اینترنت یا وب کاوی تلاش می‌کند که به همه این موارد پردازد که معمولاً به کاوش محتوای وب، کاوش ساختار وب و کاوش ساختار وب تقسیم می‌شود.

^۱ World Wide Web

• **داده شبکه‌های اجتماعی :** شبکه‌های اجتماعی، عبارت است از شبکه‌ای که شامل افراد و گروه‌ها و ارتباطات بین آن‌ها می‌باشد. افراد و گروه‌های عضو در این شبکه نود و گره‌ها را تشکیل می‌دهند و ارتباطات و وابستگی‌های بین این مؤلفه‌ها نیز مانند دوستی، خویشاوندی، تجارت، علایق مشترک و غیره یال، پیوند، پیکان، ربط یا بند بین نودها یا گره‌ها را تشکیل می‌دهند. با افزایش تعداد گره‌ها و ارتباطات بین آن‌ها شبکه دارای پیچیدگی بیشتری می‌شود و به واسطه تحلیل ریاضیاتی شبکه می‌توان آن‌ها را مورد تجزیه و تحلیل قرار داد. شبکه به صورت مجموعه‌ای از گره‌ها و روابط بین آن‌ها تعریف می‌شود. گره‌ها می‌توانند فرد، گروه، سازمان، کشور و غیره باشند. در واقع، در تحلیل شبکه، مطالعه روابط بین گره‌ها مورد نظر است. این روابط ممکن است جهت‌دار، بدون جهت، وزن‌دار یا دوتایی (صفر و یکی) باشد. ساده‌ترین نوع شبکه، شبکه روابط دوتایی بدون جهت است که فقط وجود یا عدم رابطه بین گره‌ها را نشان می‌دهد. با استفاده از وزن رابطه می‌توان آن را بیشتر توصیف کرد. وزن می‌تواند نشان‌دهنده میزان، تکرار یا شدت رابطه باشد. برای مثال، در رابطه بین سازمان‌ها وزن رابطه‌ها ممکن است نشان‌دهنده میزان تماس‌های آن‌ها باهم باشد (وسرمن، ۱۹۹۴؛ اسکات، ۱۹۹۱، گارتون و دیگران، ۱۹۹۹). اگر در شبکه رابطه‌ای جهت‌دار باشد به آن کمان و اگر بدون جهت باشد به آن یال می‌گوییم. (هوگان، ۲۰۰۷)

۷-۱- داده ، اطلاعات و دانش

داده‌ها رشته واقعیت‌های عینی و مجرد در مورد رویدادها هستند. از دیدگاه سازمانی، داده‌ها - به درستی یک سلسله معاملات ثبت‌شده منظم تلقی شده‌اند. داده‌ها تنها بخشی از واقعیت‌ها را نشان می‌دهند و از هر نوع قضاوت، تفسیر و مبنای قابل‌اتکا برای اقدام مناسب تهی هستند. داده‌ها را می‌توان مواد خام عناصر

موردنیاز برای تصمیم‌گیری به شمار آورد، چراکه نمی‌توانند عمل لازم را تجویز کنند. داده‌ها نشانگر ربط، بی‌ربطی و اهمیت خود نیستند، اما به‌هرحال برای سازمان‌های بزرگ اهمیت زیادی دارند، چراکه مواد اولیه ضروری برای خلق دانش به شمار می‌روند.

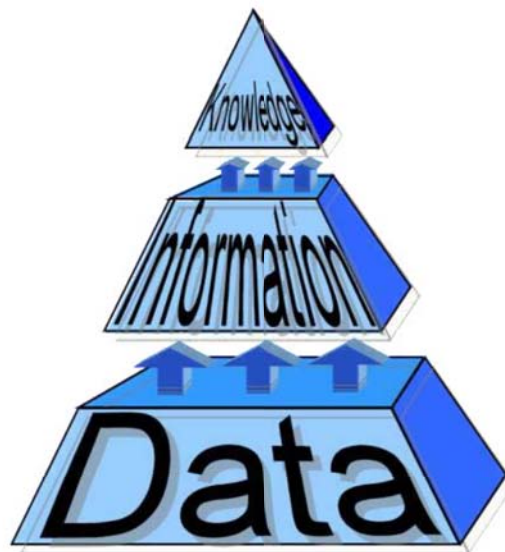
اطلاعات را نوعی پیام به شمار می‌آورند. پیام موردنظر ما معمولاً به شکل مدرکی مکتوب یا به‌صورت ارتباطی شنیداری یا دیداری نمود می‌یابد. اطلاعات باید متضمن آگاهی و حاوی داده‌هایی تغییردهنده باشد. واژه inform در انگلیسی به معنای «شکل دادن» و information نیز به معنی «شکل دادن» بینش و دید دریافت‌کننده اطلاعات است. اگر بخواهیم معنی واژه موردبحث را دقیق‌تر و سخت‌گیرانه‌تر روشن کنیم، باید بگوییم: «تنها گیرنده می‌تواند مشخص کند که دریافتی‌های او واقعاً اطلاعات بوده و او را تحت تأثیر قرار داده است». اطلاعات برخلاف داده‌ها، معنی‌دار هستند. به قول پیتر دراکر: «داشتن ارتباط و هدف، ویژگی اطلاعات است». اطلاعات، نه‌تنها دارای قابلیت تأثیرگذاری بر گیرنده هستند، بلکه خود نیز شکل خاصی دارند و برای هدف خاصی سازمان می‌یابند. داده‌ها زمانی به اطلاعات تبدیل می‌شوند که ارائه‌دهنده آن‌ها، معنی و مفهوم خاصی به آن‌ها ببخشد. با افزودن ارزش به داده‌ها، درواقع آن‌ها را به اطلاعات تبدیل می‌کنیم.

دانش مخلوط سیالی از تجربیات، ارزش‌ها، اطلاعات موجود و نگرش‌های کارشناسی نظام‌یافته است که چارچوبی برای ارزشیابی و بهره‌گیری از تجربیات و اطلاعات جدید به دست می‌دهد. دانش، در ذهن دانشور به وجود آمده و به کار می‌رود. دانش در سازمان‌ها نه‌تنها در مدارک و ذخایر دانش، بلکه در رویه‌های کاری، فرایندهای سازمانی، اعمال و هنجارها مجسم می‌شود. این تعریف، از اول مشخص می‌کند که دانش ساده و روشن نیست، مخلوطی از چند عامل متفاوت است؛ سیالی است که درعین حال ساختارهای مشخصی دارد و نهایت اینکه، ابهامی و شهودی است و به همین علت، به راحتی نمی‌توان آن را در قالب کلمات گنجاند و به‌صورت تعریفی منطقی عرضه کرد. دانش در خود مردم وجود دارد و بخشی از پیچیدگی ندانسته‌های انسانی است. ما گرچه به‌طور سنتی، سرمایه‌ها را مشخص

و ملموس می‌دانیم، اما سرمایه دانش را نمی‌توان به راحتی تعریف کرد. درست مشابه ذره اتمی که می‌تواند موج یا ذره باشد، بسته به اینکه دانشمندان چگونه وجود آن را دنبال کنند. دانش به شکل‌های پویا و نیز انباشته و ایستا قابل تصور است (رحمان سرشت، ۱۳۷۹).

دانش از اطلاعات و اطلاعات از داده‌ها ریشه می‌گیرند. تبدیل اطلاعات به دانش در عمل بر عهده خود بشر است. با نگرش فراتری به این موضوع، آشکار می‌شود که معمولاً «دانش پایه» عامل تمایز بین داده، اطلاعات و دانش است. این یکی از دلایلی است که در محیط و فضای متکی به دانش، برخی مؤسسات یا شرکت‌ها می‌توانند همچنان برتری‌های اقتصادی و رقابتی خود را حفظ کنند. «کوهن» و «لونیتال» در مباحث خود، این حقیقت را تشریح می‌کنند که گسترش دانش منوط به شور و هیجان یادگیری و دانش پیشین است. به عبارت دیگر، دانش اندوخته شده عامل مؤثری در افزایش واکنش و فراگیری سهل‌تر مفاهیم است. بنابراین، دانش ترکیب سازمان‌یافته‌ای است از «داده‌ها» که از طریق قوانین، فرایندها و عملکردها و تجربه حاصل آمده است. به عبارت دیگر، «دانش» معنا و مفهومی است که از فکر پدید آمده است و بدون آن اطلاعات و داده تلقی می‌شود. تنها از طریق این مفهوم است که «اطلاعات» حیات یافته و به دانش تبدیل می‌شوند (Leviathan, ۱۹۹۰ & Cohen).

همچنین در تعریفی دیگر یکی از پیشرفت‌های اصلی در مسیر تعریف دانش، شناخت تفاوت میان دانش، اطلاعات و داده ذکر شده است. «داده»، مجموعه‌ای از حقایق و امور مسلم درباره یک پدیده است. اطلاعات شامل سازمان‌دهی، گروه‌بندی و مقوله‌بندی داده‌ها در الگوهایی معنادار است؛ و دانش، اطلاعاتی است که با تجربه، زمینه، تعبیر و تأمل ترکیب شده و اقدام صحیح را ممکن می‌سازد. (DAVEPORT AND PRUSAK, ۱۹۹۸).



شکل ۱-۸: هرم داده، اطلاعات و دانش

انواع داده‌های مورد استفاده در داده کاوی

یک مجموعه داده از اشیا داده تشکیل شده است. نام‌های دیگر شیء داده عبارت‌اند از رکورد، نقطه، بردار، الگو، واقعه، مورد، نمونه، مشاهده و یا موجودیت. هر شیء داده نیز با تعدادی ویژگی توصیف می‌شود که خصوصیات اصلی آن شیء را بیان می‌کنند. نام‌های دیگر ویژگی عبارت‌اند از متغیر، خصیصه، فیلد، مشخصه و یا بعد.

ویژگی‌های کمی و کیفی

در توصیف ویژگی‌های یک شیء می‌توان چهار نوع اسمی، رتبه‌ای یا ترتیبی، فاصله‌ای یا بازه‌ای و نسبی یا نسبی را تعریف نمود. ویژگی‌های اسمی و رتبه‌ای باهم به عنوان ویژگی طبقه‌ای یا اسمی شناخته می‌شوند. این ویژگی‌ها واجد خواص عددی محدودی هستند. حتی اگر این ویژگی‌ها با عدد مثلاً عدد صحیح بیان شوند با آن‌ها باید به شکل نماد رفتار شود. دو نوع دیگر ویژگی شامل فاصله‌ای

و نسبتی به عنوان ویژگی کمی یا عددی شناخته می‌شوند. ویژگی‌های کمی با اعداد بیان شده و اکثر خواص اعداد را دارند. این ویژگی‌ها می‌توانند مقدار صحیح یا پیوسته داشته باشند. تفاوت این دو مقیاس فاصله‌ای با نسبتی در چگونگی قرارگیری نقطه صفر در مقیاس است. نقطه صفر در مقیاس فاصله‌ای به طور قراردادی و اختیاری تعریف شده است. در جدول زیر هر کدام از این ویژگی‌ها را در قالب یک جدول مشاهده می‌نمایید.

جدول ۱-۱: ویژگی‌های کمی و کیفی

تبدیل	عملیات	مثالها	توضیح	نوع ویژگی
جابجایی	مد، آنتروپی همبستگی توافقی	رنگ، جنسیت	$(=, \neq)$. تمایز	اسمی
تغییر مقادیر با حفظ ترتیب	میانه دهک رتبه همبستگی	سختی {خوب، بهتر، بهترین} رتبه گنگور شماره خیابان	ترتیب $(<, >)$	ترتیبی
جدید=قدیم+ضرب=ثابت	میانگین واریانس همبستگی	تاریخ درجه حرارت سلسیوس	$(+, -)$. تفاوت	بازه ای
f جدید=قدیم+ضرب	میانگین هندسی تفاوت نسبی	درجه حرارت کلوین پول سن، وزن، قد	$(*, /)$ تفاوت و نسبت	نسبتی

طبقه ای (کیفی)

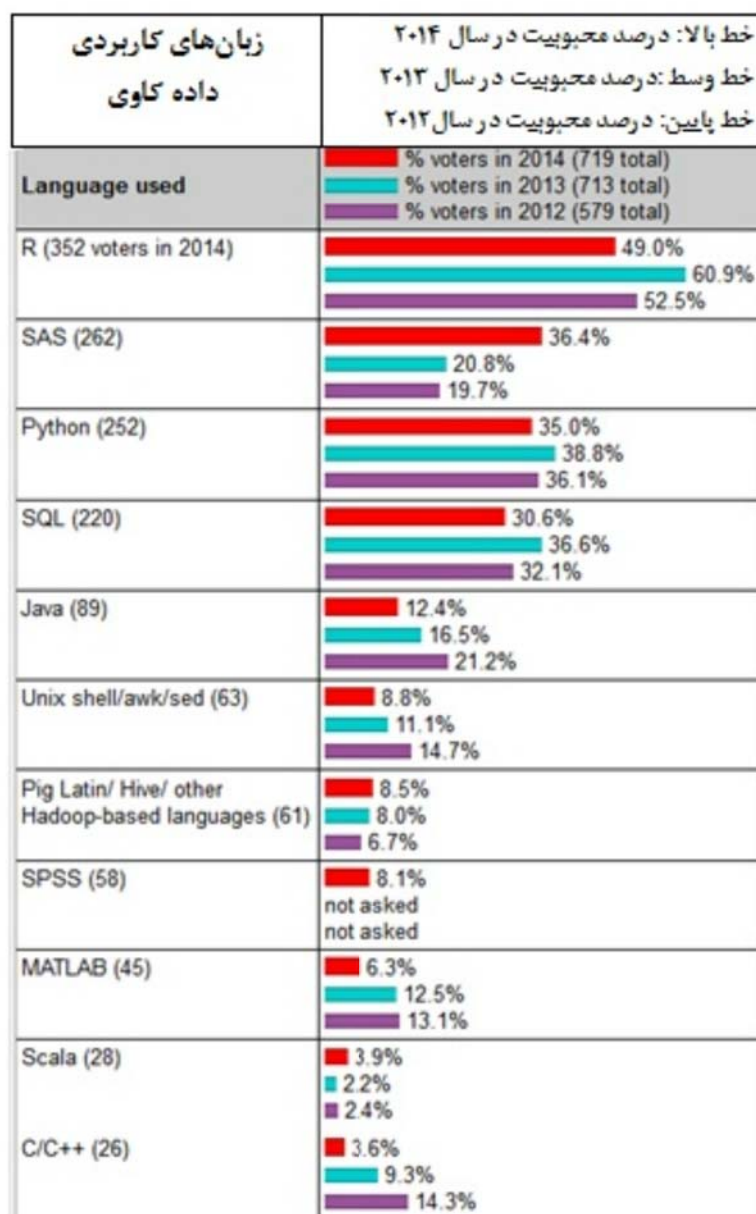
عددی (کمی)

ویژگی‌های گسسته و پیوسته

راه دیگر تفکیک ویژگی‌ها برحسب تعداد مقادیری است که می‌توانند بگیرند. ویژگی گسسته، مجموعه مقادیر محدود و یا نامحدود قابل شمارش دارد. این ویژگی‌ها می‌توانند مانند کد پستی یا شماره پرسنلی از نوع طبقه‌ای باشند و یا مثل شمارش از نوع عددی باشند. ویژگی پیوسته دارای مقادیری از نوع حقیقی است. برای مثال ویژگی‌هایی مانند دما، قد یا وزن پیوسته هستند. ویژگی‌های پیوسته نوعاً با متغیرهای اعشاری با دقت محدود بیان می‌شوند.

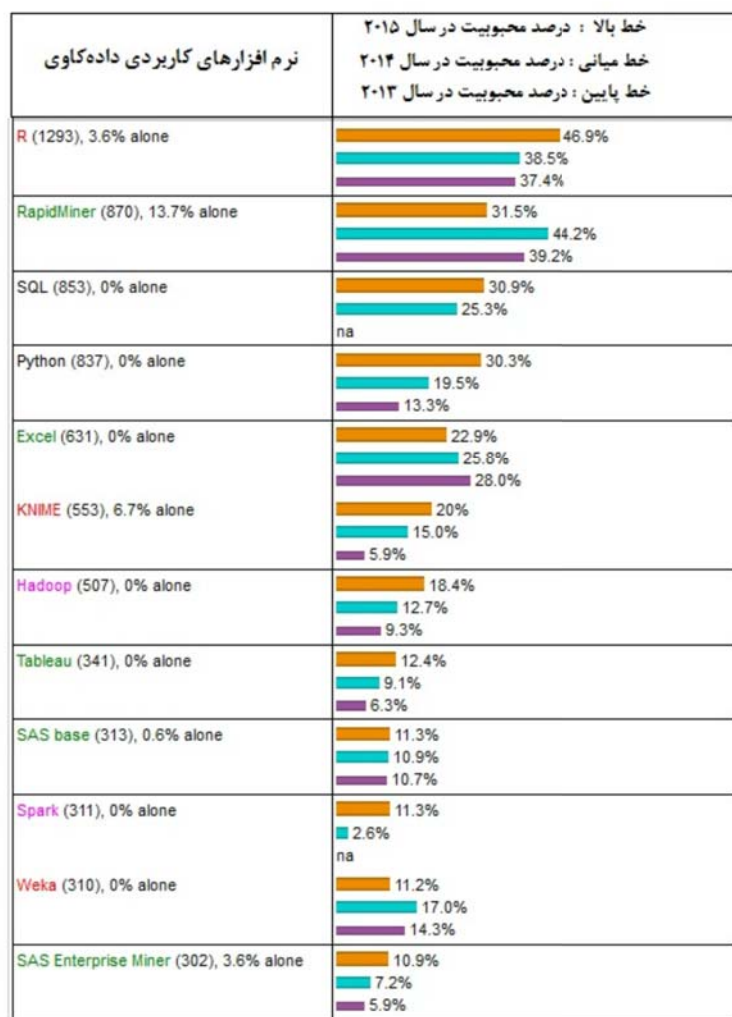
۸-۱- انواع ابزارها و زبان‌های مورد استفاده در داده کاوی

در شکل زیر مهم‌ترین زبان‌های برنامه‌نویسی و آماری مرتبط با داده کاوی را مشاهده می‌نمایید. همچنین در شکل بعدی مهم‌ترین ابزارهای داده کاوی در دنیا در سالهای ۲۰۱۳، ۲۰۱۴ و ۲۰۱۵ با یکدیگر مقایسه شده است.



شکل ۹-۱: مهم‌ترین زبان‌های داده کاوی

(<http://www.kdnuggets.com/polls/۲۰۱۴/languages-analytics-data-mining-data-science.html>)



شکل ۱- ۱۰ : مهم ترین ابزارهای داده کاوی

(<http://www.kdnuggets.com/۲۰۱۵/۰۵/poll-r-rapidminer-python-big-data-spark.html>)

۹-۱ چالش های تحقیقات و کاربردهای داده کاوی

در ادامه به برخی از چالش های مربوط به پژوهش ها و متدهای داده کاوی در حال حاضر می پردازیم. موارد زیر هرچند همه این چالش ها را در بر نمی گیرد اما قصد داریم که به برخی از انواع مشکلات و چالش های داده کاوی را بررسی نماییم.

- پایگاه‌های داده بزرگ‌تر^۱ : پایگاه‌های داده با هزاران فیلد و جدول و میلیون‌ها رکورد با چندین گیگابایت حجم امروزه پیش‌پاافتاده هستند و پایگاه‌های داده در ابعاد ترابایت در حال ظهور می‌باشند. برای مواجهه مناسب با این حجم بزرگ از داده‌ها نیازمند استفاده از الگوریتم‌های کارا^۲ (Agrawal et al. ۱۹۹۶)، نمونه‌گیری^۳، تقریب^۴ و پردازش موازی^۵ انبوه می‌باشیم. (Holsheimer et al. ۱۹۹۶)
- ابعاد بزرگ^۶: در برخی موارد ممکن است پایگاه‌های داده دارای رکوردهای زیاد نباشند اما دارای فیلدهای زیادی (خصوصیت^۷، متغیرها^۸) باشند که در این حالت تعداد ابعاد مسئله بالا می‌رود. در مجموعه داده‌های با ابعاد بالا مشکلاتی را در افزایش حجم جستجو خواهیم داشت. برای افزایش موفقیت در داده‌کاوی در این موارد می‌توان از روش‌های غیررسمی استفاده نمود که به‌صورت کلی معتبر نیستند. روش‌های این مسائل شامل متدهایی است که موجب کاهش ابعاد مسئله به‌صورت کارا و استفاده از دانش سابق برای شناخت متغیرهای نامربوط می‌باشد.
- عدم تطابق^۹: زمانی که الگوریتمی برای بهترین پارامترها برای یک مدل مشخص در یک مجموعه داده محدود اعمال می‌شود، دارای الگوی عمومی برای داده‌ها نمی‌باشند. وجود هر داده مغشوشی در داده‌ها، موجب عملکرد ضعیف مدل در تست داده‌ها می‌شود. راه‌حل‌های ممکن

^۱ Larger databases

^۲ efficient algorithms

^۳ sampling

^۴ approximation

^۵ parallel processing

^۶ High dimensionality

^۷ attributes

^۸ variables

^۹ Overfitting

شامل روش‌های اعتبارسنجی متقاطع^۱، تنظیم کردن^۲ و راهبردهای آماری پیشرفته^۳ می‌باشد.

- ارزیابی آماری^۴: این مشکل زمانی رخ می‌دهد که سیستم روی مدل‌های ممکن بسیاری مورد جستجو قرار بگیرد. برای مثال اگر یک سیستم مدل‌ها را در یک هزارم سطح اهمیت تست نماید، سپس به طور متوسط، با داده‌های خالص تصادفی، $N/1000$ این مدل‌ها به صورت معنادار مورد قبول واقع خواهد شد. این نکته‌ای است که به صورت مکرر در تلاش‌های اولیه در کشف دانش و داده کاوی فراموش می‌شود. یک راه برای مواجهه با این مشکل استفاده از متدهایی است که با تستهای آماری به عنوان یک تابع جستجو تنظیم شده است. برای مثال تنظیمات بنفرونی^۵ برای تستهای مستقل یا تستهای تصادفی می‌باشد.
- تغییر داده‌ها و دانش^۶: تغییر سریع داده‌ها می‌تواند الگوهای کشف شده قبلی را نامعتبر نماید. به علاوه متغیرهایی که اندازه‌گیری می‌شود، در یک پایگاه داده نرم‌افزاری می‌تواند اصلاح گردد، پاک شود و یا در طی زمان با اندازه‌گیری‌های جدید تقویت گردد. راه‌حل‌های ممکن شامل افزایش روش‌هایی برای به روزرسانی الگوها و بهبود تغییر به عنوان یک فرصت برای کشف با جستجو برای الگوهای تغییر می‌باشد. (Matheus, Piatetsky-Shapiro, and McNeill ۱۹۹۶)
- داده‌های گم‌شده و نویز^۷: این مشکل به ویژه در پایگاه داده‌های تجاری رخ می‌دهد. طبق گزارش داده‌های سرشماری ایالات متحده آمریکا نرخ خطا در برخی فیلدها حدود ۲۰ درصد می‌باشد. خصوصیت‌های مهم می‌توانند از دست بروند اگر پایگاه‌های داده به صورت اکتشافی و صحیح

^۱ cross-validation

^۲ regularization

^۳ sophisticated statistical strategies

^۴ Assessing of statistical significance

^۵ Bonferroni

^۶ Changing data and knowledge

^۷ Missing and noisy data

در ذهن طراحی نشده باشند. راه حل های ممکن شامل راهبردهای آماری پیشرفته برای شناخت متغیرها و وابستگی های پنهان می باشد.

(Heckerman ۱۹۹۶; Smyth et al. ۱۹۹۶)

- ارتباط پیچیده بین فیلدها^۱ : خصوصیت ها و مقادیر با ساختار سلسله مراتبی^۲، ارتباطات بین ویژگی ها و متدهای پیچیده بیشتر برای نمایش دانش درباره محتوای یک پایگاه داده نیازمند الگوریتم هایی است که به صورت کارا از برخی اطلاعات استفاده نماید. به طور معمول الگوریتم های داده کاوی برای رکوردهای مقادیر و ویژگی های ساده توسعه یافته است، اگرچه تکنیک های جدید برای استخراج ارتباط بین متغیرها ایجاد گردیده است

(Dzeroski ۱۹۹۶; Djoko, Cook, and Holder ۱۹۹۵).

- فهم الگوها^۳: در بسیاری از نرم افزارها، این مهم است که اکتشافات برای انسان قابل فهم باشد. راه حل های ممکن شامل نمایش گرافیکی (Heckerman ۱۹۹۶; Buntine ۱۹۹۶)، ساختار قوانین^۴، تولید زبان طبیعی^۵ و تکنیک هایی برای مصورسازی داده ها و دانش می باشد. راهبردهای پالایش قانون (Major and Mangano ۱۹۹۵) می تواند برای آدرس دهی به مسائل مرتبط مورد استفاده قرار بگیرد. دانش کشف شده ممکن است به صورت شفاف یا صریح، زائد باشد.
- تعامل کاربری و دانش قبلی^۶: بسیاری از متدها و ابزارهای کشف دانش و داده کاوی جاری به صورت صحیح متعامل^۷ نیستند و نمی تواند به راحتی با دانش قبلی پیرامون یک مسئله ترکیب گردد مگر در یک مدل ساده. استفاده از دامنه دانش در همه مراحل فرآیند کشف دانش

^۱ Complex relationships between fields

^۲ Hierarchically

^۳ Understandability of patterns

^۴ rule structuring

^۵ natural language

^۶ User interaction and prior knowledge

^۷ interactive

و داده کاوی مهم می باشد. مدل های بیزی^۱ (Cheeseman [۱۹۹۰]) از احتمالات سابق در داده ها و توزیعات به عنوان یک نوع رمزگذاری دانش سابق استفاده می نماید. دیگران قابلیت های استنتاجی پایگاه داده را برای کشف به منظور هدایت جستجوهای پایگاه داده به خدمت می گیرند.

(for example, Simoudis, Livezey, and Kerber [۱۹۹۵])

- ادغام با سیستم های دیگر^۲: یک سیستم اکتشافی مستقل ممکن است خیلی مفید نباشد. موضوعات ادغام طبیعی شامل ادغام با یک سیستم مدیریت پایگاه داده (به عنوان مثال اینترفیس پرس و جو، ادغام با صفحات گسترده و ابزارهای مصورسازی و انطباق خواندن حسگرهای بلادرنگ می باشد.

Simoudis, Livezey, and Kerber (۱۹۹۵) and Stolorz, Nakamura, Mesrobiam, Muntz, Shek, Santos, Yi, Ng, Chien, Mechoso, and Farrara (۱۹۹۵).

۱۰-۱- انبار داده و پایگاه داده تراکنشی

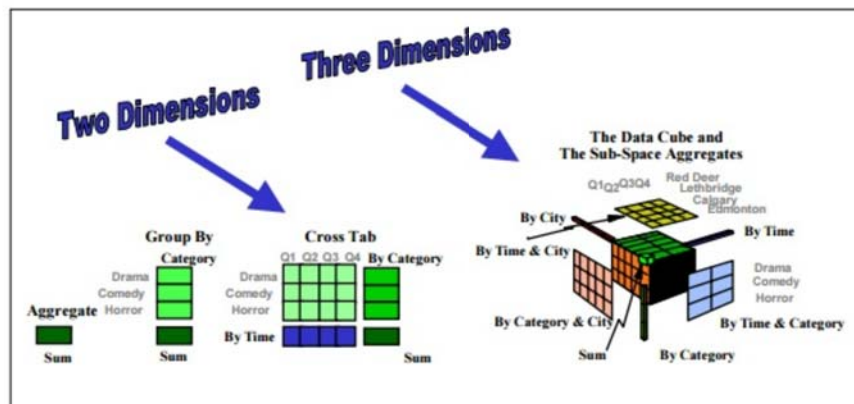
- **انبار داده**^۳: یک انبار داده همانند یک انبار، مخزنی از داده های جمع آوری شده از چند منبع داده ای (معمولاً نامتجانس) می باشد که به صورت جامع در یک شمای یکپارچه مورد استفاده قرار می گیرد. یک انبار داده این امکان را می دهد که داده ها را از منابع مختلف در یک بستر مشترک مورد تجزیه و تحلیل قرار دهیم. تصور کنید که یک شرکت دارای چند شعبه و دارای پایگاه داده های متفاوت با ساختار متفاوتی می باشد. در صورتی که مدیر شرکت بخواهد به همه داده ها برای یک تصمیم گیری استراتژیک، جهت گیری برای آینده، بازاریابی یا موارد مشابه اقدام نماید بهتر است که همه داده ها در یک مکان با یک ساختار

^۱ Bayesian approaches

^۲ Integration with other systems

^۳ Data Warehouses

متجانس ذخیره گردد که اجازه تحلیل تعاملی را بدهد. به عبارت دیگر، داده‌ها از مخازن مختلف گردآوری، پاک‌سازی، منتقل و با یکدیگر یکپارچه می‌گردند. برای تسهیل تصمیم‌گیری و یک دیدگاه چندبعدی، انبارهای داده معمولاً با یک ساختار چندبعدی مدل‌سازی می‌گردند. شکل زیر نشان‌دهنده یک ساختار داده‌ای چندبعدی به شکل مکعب می‌باشد که برای نمایش داده‌های انبار داده یک کمپانی استفاده می‌گردد. این شکل هر بعد این مکعب داده‌ای شامل سلسله مراتبی از مقادیر برای یک ویژگی می‌باشد. به خاطر این ساختار داده‌های خلاصه‌شده و پیش‌پردازش شده و به خاطر ویژگی سلسله‌مراتبی در ابعاد مختلف، مکعب‌های داده‌ای، برای پرس‌وجوهای تعاملی سریع و تحلیل داده‌ها در سطوح مختلف معنایی مناسب می‌باشد که با عنوان پردازش تحلیل برخط شناخته می‌شود.



شکل ۱-۱۱: مکعب چندبعدی داده‌ها در انبار داده

- پایگاه‌های داده تراکنشی^۱: یک پایگاه داده تراکنشی مجموعه‌ای از رکوردهاست که نشان‌دهنده تراکنش‌ها می‌باشد که هر کدام دارای زمان خاص خود، یک مشخصه و مجموعه‌ای از اعلام می‌باشد. موارد مرتبط

^۱ Transaction Databases

با فایل تراکنش‌ها می‌تواند همچنین توصیف‌کننده داده‌ها برای هر کدام از اعلام باشد. برای مثال در مورد نمونه ذخیره ویدئوها، یک جدول مربوط به پرداخت اجاره همان‌طور که در شکل زیر نشان داده شده یک پایگاه داده تراکنشی را نشان می‌دهد. هر رکورد یک قرارداد با یک مشتری دارای شناسه، یک تاریخ و لیست مواردی که اجاره شده است را شامل می‌گردد. از آنجایی که پایگاه‌های داده‌ای رابطه‌ای اجازه جداول تودرتو را نمی‌دهد، تراکنش‌ها معمولاً در یک فایل تخت یا دو جدول استاندارد شده تراکنش‌ها، یکی برای خود تراکنش‌ها و دیگری موارد تراکنشی، ذخیره می‌شوند. یک تحلیل مرسوم داده‌کاوی بر روی چنین داده‌هایی به صورت تجزیه و تحلیل سبد بازار یا قواعد انجمنی بررسی می‌گردد.

Rentals				
transactionID	date	time	customerID	itemList
T12345	99/09/06	19:38	C1234	{I2, I6, I10, I45 ... }
...				

شکل ۱-۱۲: پایگاه داده تراکنشی

فصل دوم

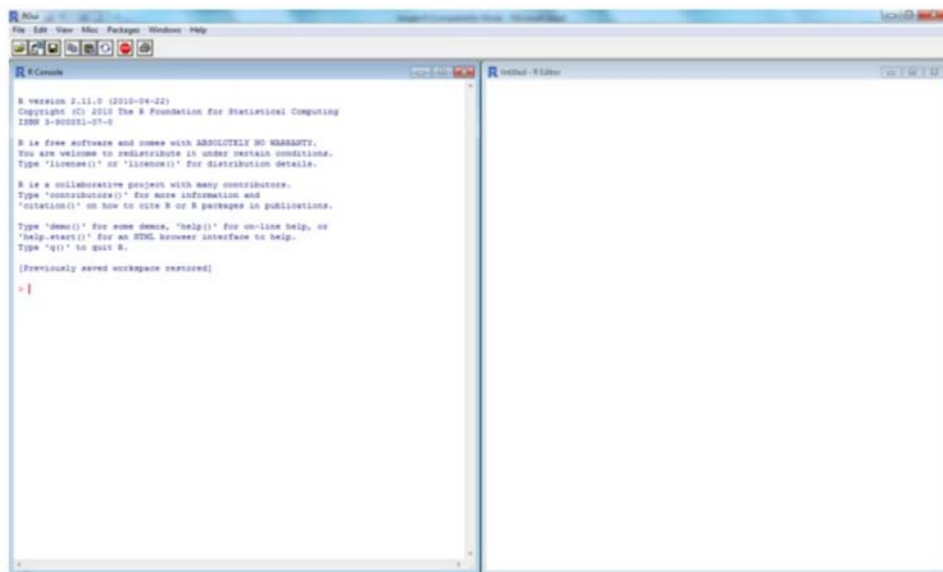
مقدمه ای بر زبان R

۲-۱- مقدمات و توابع ابتدایی

R، یک زبان برنامه‌نویسی و محیط نرم‌افزاری برای محاسبات آماری و تحلیل داده است، که بر اساس زبان اس و اسکیم پیاده‌سازی شده است. این نرم‌افزار متن‌باز تحت اجازه‌نامه عمومی همگانی گنو (GPL ۳) عرضه شده و به‌صورت رایگان در دسترس عموم می‌باشد.

گرچه نرم‌افزار R اغلب به‌منظور انجام محاسبات آماری به کار می‌رود، این نرم‌افزار قابل به‌کارگیری در محاسبات ماتریسی است و در این زمینه، همپای نرم‌افزارهایی چون اکتاو و نسخه تجاری آن متلب است. R، همچنین نرم‌افزار قدرتمندی برای ایجاد اشکال گرافیکی و نمودارهاست.

نمایی از محیط نرم‌افزار R :



شکل ۲-۱ : نمایی از نرم‌افزار R

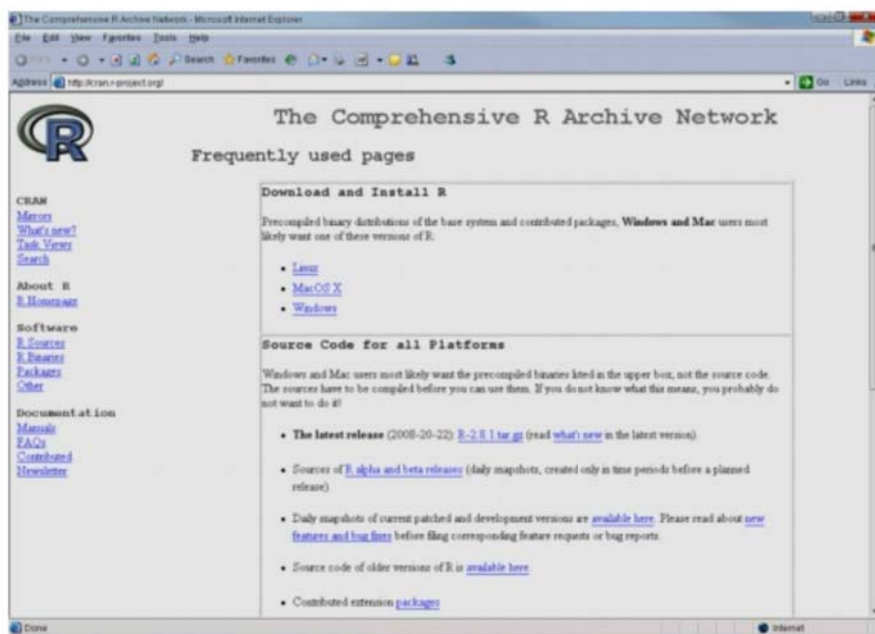
۲-۲- مزایای منحصربه‌فرد نرم‌افزار R

یک نرم‌افزار متن‌باز (open source) و رایگان است. یک بسته آموزشی رایگان در مراکز آموزشی قابل استفاده است. توسعه‌پذیری و انعطاف R باعث دسترسی همگان در کوتاه‌ترین زمان به روش‌های آماری نوین شده است.

قابلیت‌های فنی :

- کاربری و نگهداری داده‌ها به صورت مفید و مؤثر
- دارا بودن بسیاری از عملگرهای لازم برای محاسبات ماتریسی و آرایه‌ای
- دارا بودن مجموعه کاملی از ابزار تجزیه و تحلیل داده‌ها
- امکانات گرافیکی منحصربه‌فرد برای تحلیل داده‌ها و نمایش آن‌ها
- زبان برنامه‌نویسی ساده با قابلیت‌های بروز رسانی بالا

برای دانلود نرم‌افزار R می‌توان به سایت آن مراجعه نمود . (www.r-project.org)



شکل ۲-۲: وب‌سایت نرم‌افزار R

وبسایت <http://cran.r-project.org> نیز به عنوان راهنمایی انجام پروژه‌های R پاسخگوی محققین می‌باشد :



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages. **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2014-07-10: R 3.1.1) is [available here](#). Read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [package](#).

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

شکل ۲-۳ : دانلود نرم‌افزار R با توجه به سیستم عامل

از طریق این سایت و از قسمت سمت چپ می‌توان به پکیج‌های مختلف (حدود ۶۰۰۰ پکیج) دسترسی پیدا نمود. این پکیج‌ها هم به صورت الفبایی و هم به صورت تاریخ اضافه شدن قابل دسترسی هستند.

Available CRAN Packages By Date of Publication

Date	Package	Title
2014-09-16	herryFunctions	function collection related to hydrology, zooming and shapefiles
2014-09-16	Distance	A simple way to fit detection functions to distance sampling data and calculate abundance density for biological populations
2014-09-16	dum	Density surface modelling of distance sampling data
2014-09-16	glmnet	Gamma Lasso Regression
2014-09-16	GlobalOptions	Generate functions to get or set global options
2014-09-16	hypergeom	Hypergeometric tests
2014-09-16	kselection	Selection of k in k-means clustering
2014-09-16	LiblineaR	Linear Predictive Models Based On The liblinear C/C++ Library
2014-09-16	mtfe	Models for Financial Economics

شکل ۲-۴ : دسترسی به بسته‌های نرم‌افزار R از طریق وبسایت این نرم‌افزار با توجه به تاریخ انتشار

Available CRAN Packages By Name

ABCDEFGHIJKLMNOPQRSTUVWXYZ

A3	A3: Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
abc	Tools for Approximate Bayesian Computation (ABC)
abcdeFBA	ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package
ABCExtremes	ABC Extremes
ABCOptim	Implementation of Artificial Bee Colony (ABC) Optimization
ABCP2	Approximate Bayesian Computational model for estimating P2
abctools	Tools for ABC analyses
abd	The Analysis of Biological Data
abf2	Load Axon ABF2 files (currently only in gap-free recording mode)
abind	Combine multi-dimensional arrays
abn	Data Modelling with Additive Bayesian Networks
abundant	Abundant regression and high-dimensional principal fitted components
accelerometry	Functions for processing minute-to-minute accelerometer data
AcceptanceSampling	Creation and evaluation of Acceptance Sampling Plans

شکل ۲-۵: دسترسی به بسته‌های نرم‌افزار R از طریق وبسایت این نرم‌افزار با توجه به ترتیب الفبایی با انتخاب هر پکیج نیز می‌توان فایل مربوط به آن را دانلود و در R نصب نمود. راهنمای استفاده هر پکیج نیز در همان صفحه پکیج قرار داده شده است.

agridat: Agricultural datasets

Datasets from books and papers related to agriculture. Example analyses included. Functions for plotting field designs and GGE biplots.

Version: 1.9
 Depends: grid, lattice, reshape2
 Suggests: agricolae (≥ 1.2), car, coin, corrgram, equivalence, FrF2, gam, gstat, HH, knitr, latticeExtra, lme4 (≥ 1.1-5), mapproj.
 Published: 2014-07-02
 Author: Kevin Wright
 Maintainer: Kevin Wright <kw.stat at gmail.com>
 BugReports: <https://github.com/kwstat/agridat/issues>
 License: [GPL-2](#)
 URL: <https://github.com/kwstat/agridat>
 NeedsCompilation: no
 Materials: [README NEWS](#)
 CRAN checks: [agridat results](#)

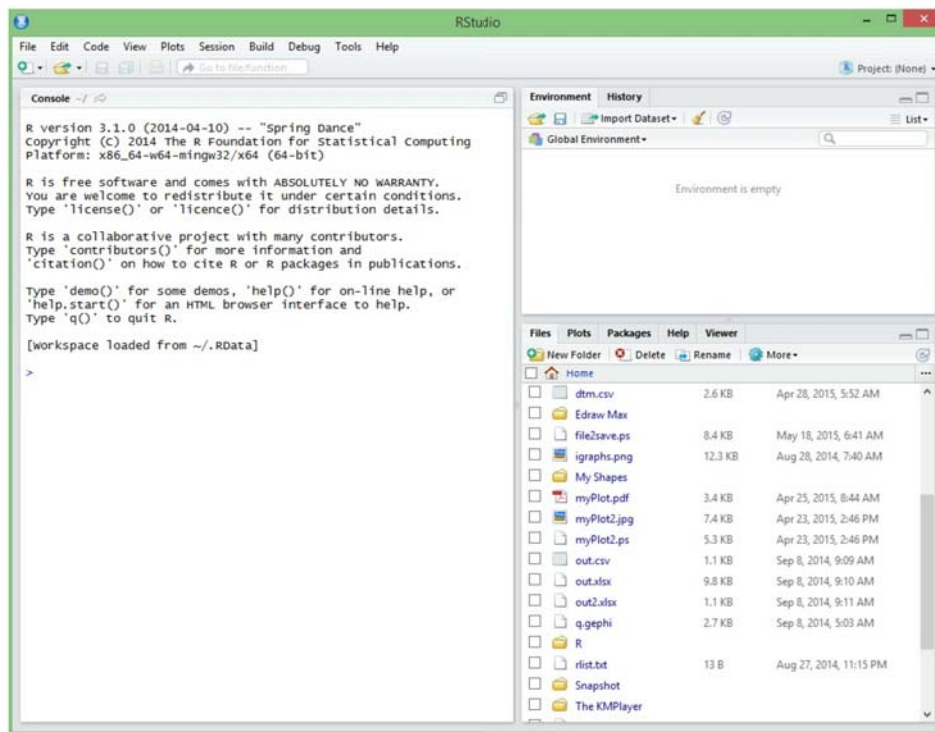
Downloads:

Reference manual: [agridat.pdf](#)
 Vignettes: [Some examples from 'agridat'](#)
 Package source: [agridat_1.9.tar.gz](#)
 Windows binaries: r-devel: [agridat_1.9.zip](#), r-release: [agridat_1.9.zip](#), r-oldrel: [agridat_1.9.zip](#)
 OS X Snow Leopard binaries: r-release: [agridat_1.9.tgz](#), r-oldrel: [agridat_1.9.tgz](#)
 OS X Mavericks binaries: r-release: not available

شکل ۲-۶: راهنمای استفاده از بسته‌های نرم‌افزار R در وبسایت این نرم‌افزار

نگاهی به نرم افزار R-studio

بعد از اینکه نرم افزار R را دانلود نموده اید می توانید نسخه R studio را نیز از داندلود نمود و پس از نصب استفاده نمایید. در این نرم افزار کدنویسی به مراتب ساده تر بوده و رابط کاربری راحت تری را برای کاربران مهیا می نماید. از جمله مزایای این نرم افزار که می توان اشاره نمود قابلیت مدیریت همزمان کدنویسی و خروجی های برنامه ، فایل ها ، بسته ها ، تصاویر و ... در یک صفحه می باشد و همچنین امکان کدنویسی به مراتب راحت تر گردیده است. امکان عیب یابی تسهیل گردیده و همچنین مدیریت ورود و خروج و مشاهده مجموعه های داده های آسان شده است. در کل توصیه می شود کاربرانی که علاقه مند به رابط کاربری قوی تری هستند از این نسخه استفاده نمایند. در شکل زیر نمایی از این نرم افزار را مشاهده می نمایید. برای نصب بسته های جدید نیز از منوی Tools گزینه افزودن بسته را انتخاب می نمایم.



شکل ۲-۷: نمایی از نرم افزار R Studio

۲-۳- قراردادهای ابتدایی

برای ذخیره یک ساختار داده از عملگر تخصیص "=", ">" یا "<" استفاده می‌شود. با تایپ یک ساختار داده می‌توانیم محتویات آن را مشاهده نماییم.

```
> x=5
> x
[1] 5
```

تابع `print(x)` نیز کار دستور فوق را انجام می‌دهد. از این دستور بیشتر برای نوشتن توابع یا حلقه‌ها استفاده می‌شود.

نام یک ساختار (متغیر) باید با یک حرف شروع شود. (A تا Z و a تا z) و می‌تواند شامل حروف، اعداد و نقطه باشد. برای مثال تمام اسامی زیر در R قابل قبول هستند.

`Var, var, var.10, var.new`

R نسبت به بزرگ و کوچک بودن حروف حساس است. بنابراین در R، 'X' و 'x' باهم تفاوت دارند.

R معمولاً فاصله‌ها را در نظر نمی‌گیرد.

```
> 10 +4
[1] 14
```

اکثر دستورات در R به صورت تابع هستند. شکل کلی توابع به صورت زیر است.

Function(شناسه‌ها)

شناسه‌های تابع ، اعداد یا عباراتی هستند که چگونگی عملکرد تابع را کنترل می‌کنند. هر تابع تعدادی شناسه دارد که برخی لازم و برخی اختیاری هستند. در صورتی که نام تابع را بنویسید، متن تابع روی صفحه نمایش ظاهر می‌شود.

آشنایی با برخی توابع مقدماتی در R

attach()	ایجاد متغیر مستقل در ساختارهای داده. مثلاً اگر متغیر x ترکیب دو متغیر y و z باشد، با اجرای دستور attach(x) متغیرهای y و z به عنوان دو متغیر آزاد قابل دسترسی هستند.
detach()	این تابع عکس عملیات تابع attach() را انجام می دهد.
help()	برای استفاده از راهنمایی های R از تابع help() یا عملگر "?" استفاده می شود. شناسه این تابع اغلب نام تابعی است که می خواهیم درباره آن اطلاعاتی کسب کنیم. Help(نام تابع) نام تابع ?
apropos()	برای جستجوی یک عبارت در اسامی توابع از تابع apropos() یا عملگر "???" استفاده می شود. مثلاً برای مشاهده همه توابعی که دارای عبارت mean هستند از دستور زیر استفاده می شود : ??mean
ls()	لیست متغیرهایی را که تاکنون ساخته ایم نمایش می دهد. برای مشاهده اسامی همه متغیرها از این عبارت استفاده می کنیم. برای مشاهده اسامی متغیرهایی که حرف m دارند از عبارت زیر استفاده می کنیم.
rm()	برای حذف کردن متغیرهای ایجاد شده از این دستور استفاده می شود. برای حذف کردن همه متغیرها : rm(list=ls()) برای حذف کردن متغیر x : rm(x)
sink()	این تابع خروجی برنامه یا عبارت را در یک فایل متن ذخیره می کند. برای استفاده از این تابع باید قبل از اجرای دستوراتی که مایل به ذخیره کردن خروجی آن ها هستیم. یک نام برای فایل متن داخل

<p>''' به عنوان شناسه در تابع <code>sink()</code> قرار دهیم، بعد از اجرای دستورات دوباره تابع <code>sink()</code> را بدون شناسه اجرا می کنیم.</p> <pre>sink("list.txt") ls() sink()</pre> <p>در این مثال خروجی تابع <code>ls()</code> در فایل <code>list.txt</code> ذخیره می شود.</p>	
<p>برای فراخوانی لیستی از دستورات از یک فایل خارجی ، فایل اسکریپت خود را در <code>notepad</code> نوشته و آن را ذخیره می کنیم. در نرم افزار R از منوی <code>File</code> گزینه <code>Source R code</code> را انتخاب می نماییم. با این کار همه دستورات داخل فایل اسکریپت خوانده و اجرا می شود. همچنین می توانیم از دستور زیر برای انجام این کار استفاده نماییم.</p> <pre>source("h:/analysis.txt")</pre> <p>اگر مایل به انتخاب تنها بخشی از دستورات باشیم از منوی <code>file</code> گزینه <code>Display file</code> را انتخاب نموده و با این کار فایل اسکریپت در پنجره ای جدید باز می شود و می توان دستورات مورد نظر را کپی نمود و سپس <code>paste</code> کرد. در محیط R امکان ویرایش فایل اسکریپت نمی باشد.</p>	<p><code>source()</code></p>
<p>در R تعداد زیادی ساختار داده از قبل وجود دارد که برای اهداف آموزشی و مثال ها استفاده می شود. برای مشاهده لیست این داده ها از تابع <code>data()</code> استفاده می شود. مثلاً یکی از این ساختارها <code>orange</code> است که به داده های مربوط به رشد درختان پرتقال مربوط می باشد.</p>	<p><code>data()</code></p>
<p>لیست تمام پکیج های نصب شده در R را نمایش می دهد. برای بارگذاری یک پکیج خاص نیز از این دستور استفاده می شود. مثلاً برای بارگذاری پکیج <code>foreign</code> از دستور زیر استفاده می نماییم :</p> <pre>library(foreign)</pre>	<p><code>library()</code></p>

۲-۴- ساختار داده‌ها در R

توجه داشته باشیم که نرم‌افزار R به بزرگی و کوچکی حروف حساس می‌باشد. در کد زیر مشاهده می‌فرمایید که در مقدار دادن به متغیرها کوچک یا بزرگ بودن حروف دارای اهمیت می‌باشد.

```
> a=10
> A=20
> a
[1] 10
> A
[1] 20
```

بردار

ساده‌ترین نوع ساختار در R بردار می‌باشد که مجموعه‌ای مرتب از مقادیر عددی، کاراکتری و یا منطقی است. رایج‌ترین روش ساختن بردارها استفاده از `c()` می‌باشد. با استفاده از این تابع مقادیر دلخواه را می‌توان باهم ترکیب نمود و یک بردار ساخت.

```
> x<-c(1,2,3,4)
> x
[1] 1 2 3 4
> y<-c(5,6,7,8)
> z<-c(x,y,9,10)
> z
[1] 1 2 3 4 5 6 7 8 9 10
```

یک بردار همچنین می‌تواند شامل مجموعه‌ای از کاراکترها باشد.

```
> c("first","second name","22")
[1] "first" "second name" "22"
```

ایجاد بردارهای خاص از طریق کد زیر امکان‌پذیر است.

```

> x=10:20
> x
[1] 10 11 12 13 14 15 16 17 18 19 20
> x=seq(10,30,by=2)
> x
[1] 10 12 14 16 18 20 22 24 26 28 30
> x=seq(10,30,length=5)
> x
[1] 10 15 20 25 30
> rep(2,4)
[1] 2 2 2 2
> x<-c(2,3,4)
> rep(x,3)
[1] 2 3 4 2 3 4 2 3 4

```

نمایش عنصر n ام از یک بردار، نمایش عنصر n ام و m ام از یک بردار، نمایش همه عناصر به جز، عناصر n ام و m ام از یک بردار و تغییر عنصر n ام را در ادامه می بینید.

```

> x<-c(10,11,12,13,14)
> x[3]
[1] 12
> x[c(1,3)]
[1] 10 12
> x[-c(1,3)]
[1] 11 13 14
> x[3]<-0
> x
[1] 10 11 0 13 14

```

دسترسی به عناصر دلخواه یک بردار از طریق کد زیر امکان پذیر می باشد.

```
> x=c(10,7,9,10,10,12,9,10)
> which(x==10)
[1] 1 4 5 8
> x[which(x<10)]
[1] 7 9 9
```

هر بردار دارای سه ویژگی طول ، حالت و نام است که در ادامه مشاهده می نمایید.

```
> x=c(1,3,6,7,11)
> length(x)
[1] 5
> mode(x)
[1] "numeric"
```

نام گذاری عناصر بردار (دو روش) به همراه بازیابی نام عناصر در کد زیر قابل مشاهده است.

```
> a<-c(x=1,y=2.4,z=-22)
> a
      x      y      z
1.0  2.4 -22.0
> a<-c(1,2.4,-22)
> names(a)<-c("x","y","z")
>
> names(a)
[1] "x" "y" "z"
```

ماتریس

ماتریس به آرایشی مستطیلی شکل از اعداد یا عبارات ریاضی که به صورت سطر و ستون شکل یافته گفته می شود. به طوری که می توان گفت که هر ستون یا هر سطر یک ماتریس، یک بردار را تشکیل می دهد و همه عناصر ماتریس از جنس عدد می باشد.

```

> x<-1:12
> dim(x)<-c(3,4)
> x
      [,1] [,2] [,3] [,4]
[1,]     1     4     7    10
[2,]     2     5     8    11
[3,]     3     6     9    12

> matrix(1:12,nrow=3,byrow=T)
      [,1] [,2] [,3] [,4]
[1,]     1     2     3     4
[2,]     5     6     7     8
[3,]     9    10    11    12
> x<-matrix(1:9,nrow=3)
> rownames(x)<-LETTERS[1:3]
> x
      [,1] [,2] [,3]
A         1     4     7
B         2     5     8
C         3     6     9

```

ترکیب ماتریس‌ها به صورت سطری و ستونی از طریق کد زیر امکان پذیر است.

```

> cbind(A=1:4,B=5:8,C=9:12)
      A B  C
[1,] 1 5  9
[2,] 2 6 10
[3,] 3 7 11
[4,] 4 8 12
> rbind(A=1:4,B=5:8,C=9:12)
      [,1] [,2] [,3] [,4]
A         1     2     3     4
B         5     6     7     8
C         9    10    11    12

```

انتخاب زیرمجموعه‌ای از یک ماتریس را در این قطعه کد مشاهده می‌نمایید.

```
> x<-matrix(1:9,ncol=3)
> x
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> x[3,2]
[1] 6
> x[1,]
[1] 1 4 7
> x[,-c(2,3)]
[1] 1 2 3
> x[x>4]
[1] 5 6 7 8 9
```

مشاهده بعد ماتریس ، حالت ماتریس و طول ماتریس (تعداد درایه‌های ماتریس) در ادامه مشاهده می‌گردد.

```
> dimnames(x)<-list(paste("row",letters[1:3]),
+ paste("col",LETTERS[1:3]))
> x
      col A col B col C
row a      1      4      7
row b      2      5      8
row c      3      6      9
```

نام‌گذاری سطرها و ستون‌های ماتریس را مشاهده می‌نمایید.

```
> x=array(1:24,c(3,4,2))
> x
, , 1
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

, , 2
      [,1] [,2] [,3] [,4]
[1,]   13   16   19   22
[2,]   14   17   20   23
[3,]   15   18   21   24
```

```

> x=c(1:8,11:18,111:118)
> dim(x)=c(2,3,4)
> x
, , 1
      [,1] [,2] [,3]
[1,]     1     3     5
[2,]     2     4     6

, , 2
      [,1] [,2] [,3]
[1,]     7    11    13
[2,]     8    12    14

, , 3
      [,1] [,2] [,3]
[1,]    15    17   111
[2,]    16    18   112

, , 4
      [,1] [,2] [,3]
[1,]   113   115   117
[2,]   114   116   118

```

دیتافریم

نوع دیگری از ساختار داده‌ها که بسیار پرکاربرد می‌باشد و شامل حالت‌های ترکیبی و مختلفی از داده‌ها می‌باشد (شامل داده‌های عددی، کاراکترها و ...) دیتافریم می‌باشد. این قالب داده‌ای یک قالب مرسوم در نرم‌افزارهای آماری همچون SAS، SPSS و Stats می‌باشد. جدول زیر یک مثال از دیتافریم می‌باشد که دارای ستونهای عددی و کاراکتری می‌باشد.

جدول ۱-۲ دیتافریم بیماران

PatientID	AdmDate	Age	Diabetes	Status
1	10/15/2009	25	Type1	Poor
2	11/1/2009	34	Type2	Improved
3	10/21/2009	28	Type1	Excellent
4	10/28/2009	52	Type1	Poor

برای تعریف این جدول در نرم افزار R از کد زیر استفاده می نماییم.

```
> patientID <- c(1, 2, 3, 4)
> age <- c(25, 34, 28, 52)
> diabetes <- c("Type1", "Type2", "Type1", "Type1")
> status <- c("Poor", "Improved", "Excellent", "Poor")
> patientdata <- data.frame(patientID, age, diabetes, status)
> patientdata
  patientID age diabetes  status
1         1  25   Type1    Poor
2         2  34   Type2 Improved
3         3  28   Type1 Excellent
4         4  52   Type1     Poor
```

با استفاده از کدهای زیر نیز می توان بخش های مختلف این مجموعه دیتا را فراخوانی نمود.

```
> patientdata[1:2]
  patientID age
1         1  25
2         2  34
3         3  28
4         4  52
> patientdata[c("diabetes", "status")]
  diabetes  status
1   Type1    Poor
2   Type2 Improved
3   Type1 Excellent
4   Type1     Poor
> patientdata$age
[1] 25 34 28 52
> table(patientdata$diabetes, patientdata$status)

      Excellent Improved Poor
Type1         1         0   2
Type2         0         1   0
```


لیست

یک لیست شامل مجموعه‌ای از اشیا مختلف مانند بردار و ماتریس و جدول اطلاعات می‌باشد.

```
> x1=c(12,14,13,14,16)
> x2=c(12,13,10,11,15)
> mylist=list(before=x1,after=x2,drug.name="xyz")
> mylist
$before
[1] 12 14 13 14 16

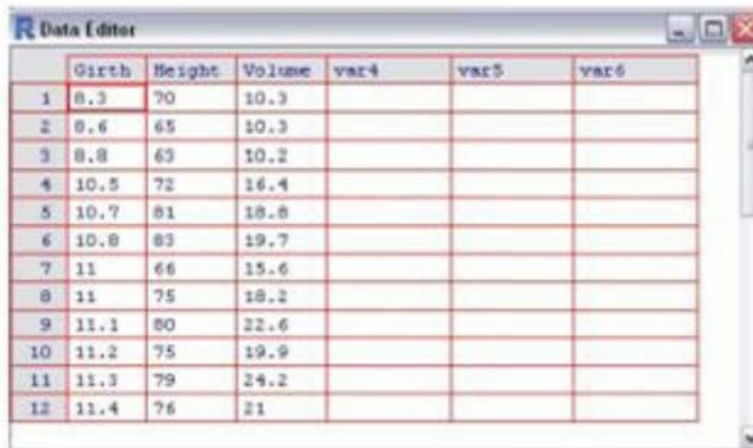
$after
[1] 12 13 10 11 15

$drug.name
[1] "xyz"

> mylist$after
[1] 12 13 10 11 15
> length(mylist)
[1] 3
> names(mylist)
[1] "before"      "after"       "drug.name"
```

تصحیح داده‌ها را در کد و شکل زیر مشاهده می‌فرمایید. داده trees یک مجموعه داده آماده می‌باشد که بجای آن می‌توان هر سری داده خاص را که توسط توابع ورودی خوانده می‌شود قرار داد.

```
> data(trees)
> data.entry(trees)
```



	Girth	Height	Volume	var4	var5	var6
1	8.3	70	10.3			
2	8.6	65	10.3			
3	8.8	63	10.2			
4	10.5	72	16.4			
5	10.7	81	18.8			
6	10.8	83	19.7			
7	11	66	15.6			
8	11	75	18.2			
9	11.1	80	22.6			
10	11.2	75	19.9			
11	11.3	79	24.2			
12	11.4	76	21			

عامل و طبقه را در قطعه کد زیر مشاهده می‌نمایید.

```
> active=c("Yes", "NO", "NO", "YES", "YES")
> factor(active)
[1] Yes NO NO YES YES
Levels: NO Yes YES

> factor(c(1,2,2,1,3,2,1), levels=c(1,2), label=c("YES", "NO"))
[1] YES NO NO YES <NA> NO YES
Levels: YES NO
```

وارد کردن داده‌ها در کد زیر آورده شده است، سطر اول شامل اسامی متغیرهاست و داده‌ها با علامت "،" از یکدیگر جدا شده‌اند و نحوه ذخیره داده‌ها نیز مشاهده می‌گردد.

```
> data=read.table("c:/data.txt")
> mydata=read.table(h:/data.dat", H=T, sep=", ")
> write.table(data, file="d:/data.dat", col.names=T, sep=", ")
```

۲-۵- محاسبات ریاضی در R

عملگرهای ریاضی و منطقی مورد استفاده در نرم افزار R را در شکل زیر مشاهده می نمایید.

توضیح	عملگر
کوچک تر	<
بزرگ تر	>
کوچک تر یا مساوی	<=
بزرگ تر یا مساوی	=>
مساوی	==
نامساوی	!=
نقیض	!
"و" منطقی	&
"یا" منطقی	
تفریق	-
جمع	+
تقسیم	/
ضرب	*
توان	^
خارج قسمت تقسیم	%/%
باقیمانده تقسیم	%%

شکل ۲-۸ : عملگرهای ریاضی و منطقی در نرم افزار R

برخی کدهای این عملگرها در ادامه آمده است.

```
> 3+4
[1] 7
> y=2
> x=4
> x*y^2
[1] 16
> x^4
[1] 256
> z=3
> (x+y)*z/x
[1] 4.5
> 25%%4
[1] 1
> 25%/%4
[1] 6

> x=2
> y=5
> if(x<y) z=x+1
> z
[1] 3
> if(x==4|x==2) w=x+y
> w
[1] 7
>
> a=c(2,4,7,11)
> a[a>=6]
[1] 7 11
```

توابع مقدماتی ریاضی

تابع	شرح
sum()	مجموع
prod()	حاصل ضرب
abs()	قدر مطلق
sign()	نشانه
exp()	تابع نمایی
cumprod()	ضرب تجمعی اعداد یک بردار
cumsum()	جمع تجمعی اعداد یک بردار
gamma()	تابع گاما
factorial()	فاکتوریل
log()	لگاریتم
log ₁₀ ()	لگاریتم در مبنای ۱۰
round()	گرد کردن اعداد
sqrt()	ریشه دوم عدد
trunc()	نزدیک‌ترین عدد صحیح به عدد
sin(),tan(),cos()	توابع مثلثاتی

شکل ۲-۹: توابع مقدماتی ریاضی در نرم‌افزار R

توابع اصلی و مقدماتی ریاضی در جدول بالا قابل دسترس می‌باشد و کدهای مربوط به آن را در ادامه آورده شده است.

```
> abs(2-4)
[1] 2
> sum(1:10)
[1] 55
> floor(3.7)
[1] 3
> factorial(7)
[1] 5040
> log(12)
[1] 2.484907
> round(sqrt(2), 4)
[1] 1.4142
> cos(2*pi)
[1] 1
> pmin(c(1,4,9), c(2,3,10))
[1] 1 3 9
> choose(8,3) #8!/3!*5!
[1] 56
> y=c(1,3,5)
> y=c(2,4,6,8,10,12)
> x+y
[1] 4 6 8 10 12 14
```

محاسبات ماتریسی نیز از طریق جدول زیر امکان پذیر می باشد.

تابع	شرح
diag()	محاسبه قطر ماتریس ایجاد یک ماتریس قطری
t()	ترانهاده ماتریس
det()	دترمینان ماتریس
ginv()	معکوس ماتریس
eigen()	محاسبه مقادیر و بردارهای ویژه یک ماتریس

شکل ۲-۱۰: محاسبات ماتریسی در نرم افزار R

در مثال زیر تعریف یک ماتریس را با دو روش متفاوت را مشاهده می نمایید.

```
> A=matrix(c(1,0,4,2,-1,1,0,3,1),nrow=3)
> A
      [,1] [,2] [,3]
[1,]    1    2    0
[2,]    0   -1    3
[3,]    4    1    1
> r=c(1,3,5)
> s=c(2,4,6)
> t=c(-2,0,1)
> B=cbind(r,s,t)
```

```

> B
      r s t
[1,] 1 2 -2
[2,] 3 4  0
[3,] 5 6  1
> A+B
      r s t
[1,] 2 4 -2
[2,] 3 3  3
[3,] 9 7  2
> A*B
      r s t
[1,]  1  4  0
[2,]  0 -4  0
[3,] 20  6  1

```

ضرب برداری و ماتریسی را در کد زیر مشاهده می‌نمایید.

```

> A%%B
      r s t
[1,]  7 10 -2
[2,] 12 14  3
[3,] 12 18 -7
> B%%A
      [,1] [,2] [,3]
[1,]   -7   -2    4
[2,]    3    2   12
[3,]    9    5   19
> c(0,2,3)%%c(1,4,1)
      [,1]
[1,]   11

```

کاربرد توابع در ماتریس شامل ترانزپوز ماتریس، دترمینان ماتریس، عناصر روی قطر اصلی، ایجاد ماتریس قطری، ایجاد ماتریس واحد را در کدهای زیر می‌بینید.


```

> F=matrix(c(11,10,15,7,8,11,15,14,120),nrow=3)
> sqrt(F)
      [,1] [,2] [,3]
[1,] 3.316625 2.645751 3.872983
[2,] 3.162278 2.828427 3.741657
[3,] 3.872983 3.316625 10.954451
> trunc(log(F))
      [,1] [,2] [,3]
[1,] 2 1 2
[2,] 2 2 2
[3,] 2 2 4
> t(F)
      [,1] [,2] [,3]
[1,] 11 10 15
[2,] 7 8 11
[3,] 15 14 120
> det(F)
[1] 1786
> diag(c(5,10,15))
      [,1] [,2] [,3]
[1,] 5 0 0
[2,] 0 10 0
[3,] 0 0 15
> diag(4)
      [,1] [,2] [,3] [,4]
[1,] 1 0 0 0
[2,] 0 1 0 0
[3,] 0 0 1 0
[4,] 0 0 0 1

```

حل دستگاه معادلات خطی

برای حل دستگاه معادلات خطی از تابع `solve` استفاده می‌کنیم. شناسه اول تابع ماتریس ضرایب و شناسه دوم بردار ستونی مقادیر معلوم است. به‌عنوان مثال دستگاه معادلات زیر را در نظر بگیرید.

$$x + 2y + 3z = 14$$

$$2x + y = 4$$

$$3x - y + 2z = 7$$

```
> A=matrix(c(1,2,3,2,1,0,3,-1,2),nrow=3,byrow=T)
> b=c(14,4,7)
> solve(A,b)
[1] 1 2 3
```

مشتق، انتگرال، پیدا کردن ریشه چندجمله‌ای

همچنین در مثال‌های زیر نمونه‌ای از مشتق‌گیری، انتگرال‌گیری و پیدا کردن ریشه چندجمله‌ای را مشاهده می‌نمایید.

$$x^3 + 2x - 3 = 0 \quad \ln(y) \quad \text{مشتق عبارت } ae^{-bx} \quad \int_0^2 2x \, dx \quad 2x^3 \quad \text{مشتق عبارت}$$

```
> D(expression(3*x^2), "x")
3 * (2 * x)
> D(expression(log(y)), "y")
1/y
> D(expression(a*exp(-b*x)), "x")
-(a * (exp(-b * x) * b))
> fx=function(x) 2*x
> integrate(fx, 0, 2)
4 with absolute error < 4.4e-14
> integrate(dnorm, -Inf, Inf)
1 with absolute error < 9.4e-05
> polyroot(c(-3,2,0,1))
[1] 1.0+0.000000i -0.5+1.658312i -0.5-1.658312i
```

احتمال و اعداد تصادفی

در جدول زیر توابع اصلی مربوط به احتمال و اعداد تصادفی را ملاحظه می‌نمایید.

توزیع	کد
دوجمله‌ای	binom
دوجمله‌ای منفی	nbinom
هندسی	geom
فوق هندسی	hyper
پواسون	pois
یکنواخت	unif
نرمال	norm
تی استودنت	t
خی دو	chisp
نمایی	exp
گاما	gamma
کوشی	cauchy
بتا	beta
لجستیک	logis

شکل ۲-۱۱ : احتمال و اعداد تصادفی در نرم‌افزار R

پیدا کردن نسبت افراد دارای وزن کمتر از ۱۰۰ کیلوگرم در جامعه‌ای که توزیع وزن نرمال با میانگین ۶۵ و انحراف معیار ۱۰ است از دستور زیر استفاده می‌شود :

```
> pnorm(100,65,10)
[1] 0.9997674
```

تولید ۵ عدد تصادفی از توزیع یکنواخت در بازه صفر و یک :

```
> runif(6)
[1] 0.4785608 0.2077610 0.5731409
     0.9903042 0.6192675 0.7808516
```

۲-۶- نوشتن توابع در R

ساختار عمومی یک برنامه در R به شکل زیر می باشد :

```
> function( شناسه ها ) {
+   متن برنامه
+ }
```

برنامه مربع میانگین یک بردار از طریق کد زیر امکان پذیر است.

```
> ms=function(x) {
+   mean(x)^2
+ }
> a=1:10
> ms(a)
[1] 30.25
```

خوش آمد گویی ساده را مشاهده می نمایید.

```
> wel=function() print("welcome")
> wel()
[1] "welcome"
```

مقادیر پیش فرض و شناسه ها در تعریف توابع در کد زیر قابل مشاهده است.

```
> myprog=function(x="datamining"){
+ print(paste("my research area is",x,"and i like it."))
+ }
>
> myprog()
[1] "my research area is datamining and i like it."
> myprog("economy")
[1] "my research area is economy and i like it."
```

مثالی از جملات شرطی در کد زیر بررسی می گردد. تشخیص زوج و فرد بودن عدد با استفاده از if() در قطعه کد زیر آورده شده است.

```
> find=function(x){
+ if(x%%2==0)print("x is couple")
+ else print("x is odd")
+ }
> find(4)
[1] "x is couple"
> find(7)
[1] "x is odd"
```

نمونه ای از عبارت ifelse() را در کد زیر مشاهده می نمایید.

```
> x=c(6:-4)
> sqrt(x)
[1] 2.449490 2.236068 2.000000 1.732051 1.414214
      1.000000 0.000000  NaN  NaN  NaN  NaN
Warning message:
In sqrt(x) : NaNs produced

> sqrt(ifelse(x>=0,x,NA))
[1] 2.449490 2.236068 2.000000 1.732051 1.414214
      1.000000 0.000000  NA  NA  NA  NA
```

نمونه ساده ای از حلقه for در کد زیر آورده شده است.

```
> for(i in 1:6) print(i^2+1)
[1] 2
[1] 5
[1] 10
[1] 17
[1] 26
[1] 37
```

محاسبه فاکتوریل یک عدد به صورت یک تابع را در کد زیر می بینید.

```
> fac1=function(x){
+ f=1
+ if(x<2) return(1)
+ for(i in 2:x){
+ f=f*i}
+ f}
> sapply(0:5,fac1)
[1] 1 1 2 6 24 120
```

۷-۲- آمار توصیفی و نمودارها

تابع	شرح
min(x)	کوچک‌ترین مقدار
max(x)	بزرگ‌ترین مقدار
range(x)	فاصله کوچک‌ترین و بزرگ‌ترین مقدار
mean(x)	میانگین
median(x)	میانه
IQR(x)	برد میان چارکی
var(x)	واریانس
sd(x)	انحراف معیار
cor(x,y)	همبستگی
cov(x,y)	کوواریانس
quantile(x,n)	n امین چندک

شکل ۷-۲: آمار تصادفی و نمودارها در نرم‌افزار R

جدول بالا مهم‌ترین متغیرهای آماری را ارائه نموده است. در قطعه کدهای بعدی چند مثال از محاسبه میانگین، انحراف معیار، واریانس، میانه، چندک‌های مهم، چند موردنظر و تفاوت چارک اول و سوم برای ۵۰ عدد تصادفی از توزیع نرمال استاندارد را مشاهده می‌نمایید.

```

> x=rnorm(50)
> mean(x)
[1] -0.1805553
> sd(x)
[1] 0.9191144
> var(x)
[1] 0.8447713
> median(x)
[1] -0.2597522
> quantile(x)
      0%      25%
-2.3591730 -0.7498358

      50%      75%      100%
-0.2597522  0.4884890  1.9933705

> pvec=seq(0,1,0.1)
> pvec
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> quantile(x,pvec)
      0%      10%      20%      30%
-2.35917302 -1.32517806 -0.84326437 -0.62388029

      40%      50%      60%      70%
-0.46789559 -0.25975220 -0.05491817  0.23089665

      80%      90%      100%
 0.65318528  0.96468057  1.99337047

> IQR(x)
[1] 1.238325

```


نمایش خلاصه از متغیرهای عددی در کد زیر آورده شده است.

```
> attach(Loblolly)
> names(Loblolly)
[1] "height" "age"      "Seed"
> summary(height)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   3.46  10.47   34.00   32.36  51.36   64.10
> summary(Loblolly)
      height          age          Seed
Min.   : 3.46  Min.   : 3.0   329   : 6
1st Qu.:10.47  1st Qu.: 5.0   327   : 6
Median :34.00  Median :12.5   325   : 6
Mean   :32.36  Mean   :13.0   307   : 6
3rd Qu.:51.36  3rd Qu.:20.0   331   : 6
Max.   :64.10  Max.   :25.0   311   : 6
                        (Other):48
```

نمودار شاخه و برگ از طریق کد زیر در دسترس می‌باشد.

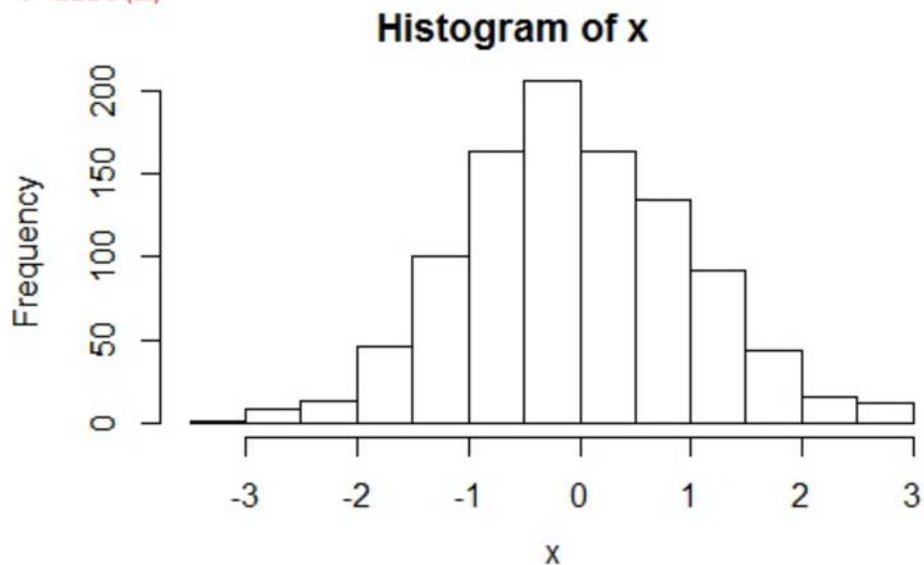
```
> scores=scan()
1: 2 3 16 23 14 12 13 2 16 17 0
6 28 33 32 7 15 3 37 3
Read 20 items
> stem(scores)

The decimal point is 1 digit(s)
to the right of the |

0 | 02233367
1 | 2345667
2 | 38
3 | 237
```

در کد و شکل زیر ترسیم هیستوگرام را مشاهده می‌نمایید.

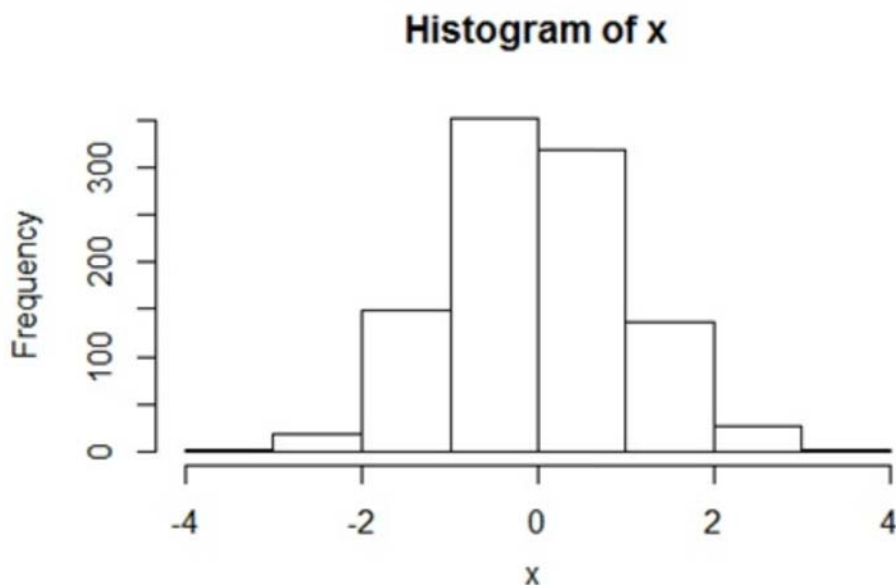
```
> x=rnorm(1000)
> hist(x)
```



شکل ۲-۱۳ : نمودار هیستوگرام

در این شکل داده‌ها در تعداد کلاس مشخص قرار داده شده‌اند.

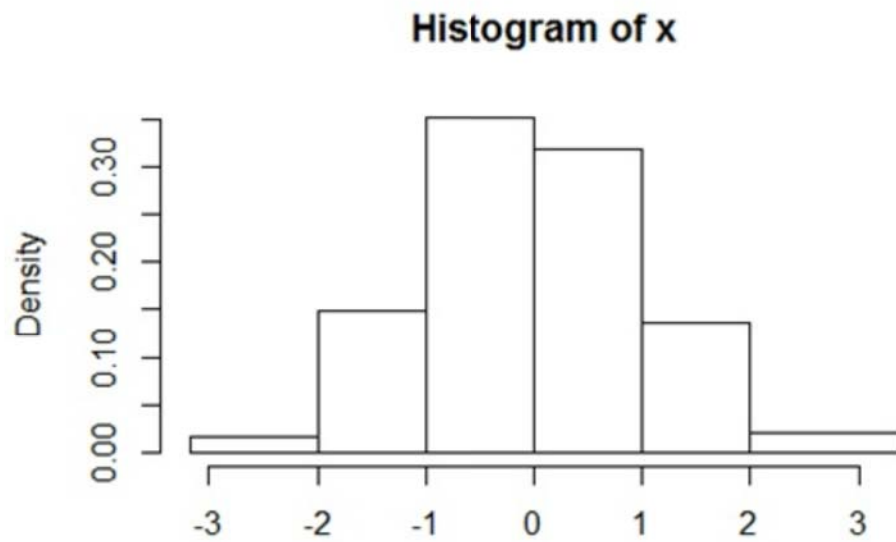
```
> hist(x, nclass=7)
```



شکل ۲-۱۴ : نمودار هیستوگرام در تعداد دسته مشخص

در این شکل نمودار هیستوگرام با نقاط قطع مشخص ترسیم گردیده است.

```
> hist(x,breaks=c(min(x),-2,-1,0,1,2,max(x)))
```



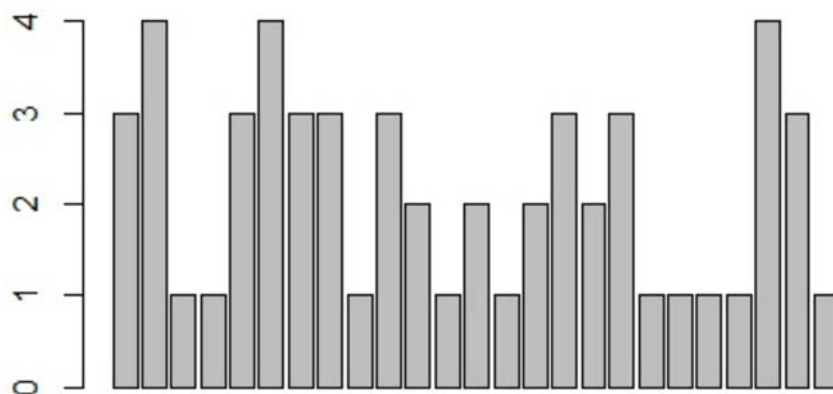
شکل ۲-۱۵: نمودار هیستوگرام با نقاط قطع مشخص

در کد زیر نحوه استفاده از جدول آورده شده است.

```
> x=c("Yes","NO","NO","YES","YES")
> table(x)
x
NO Yes YES
 2   1   2
```

شکل و کدهای مربوط به ترسیم نمودار میله‌ای در ادامه آورده شده است.

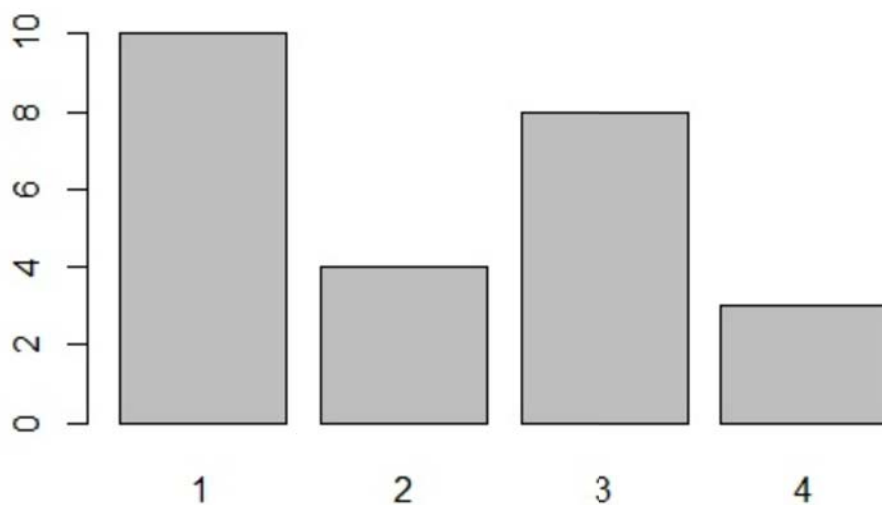
```
> uni=scan()
1: 3 4 1 1 3 4 3 3 1 3 2 1 2 1 2 3 2 3 1 1 1 1 4 3 1
26:
Read 25 items
> barplot(uni)
```



شکل ۲-۱۶: نمودار میله‌ای

کد زیر نمودار میله‌ای با محورهای متفاوت را نشان می‌دهد.

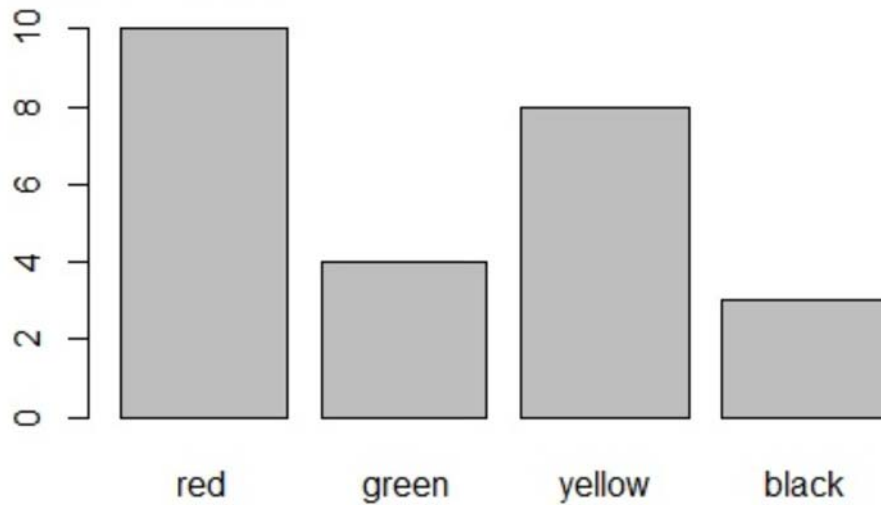
```
> tuni=table(uni)
> barplot(tuni)
```



شکل ۲-۱۷: نمودار میله‌ای با محورهای متفاوت

نام‌گذاری ستون‌ها در نمودار میله‌ای از طریق کد زیر امکان‌پذیر می‌باشد.

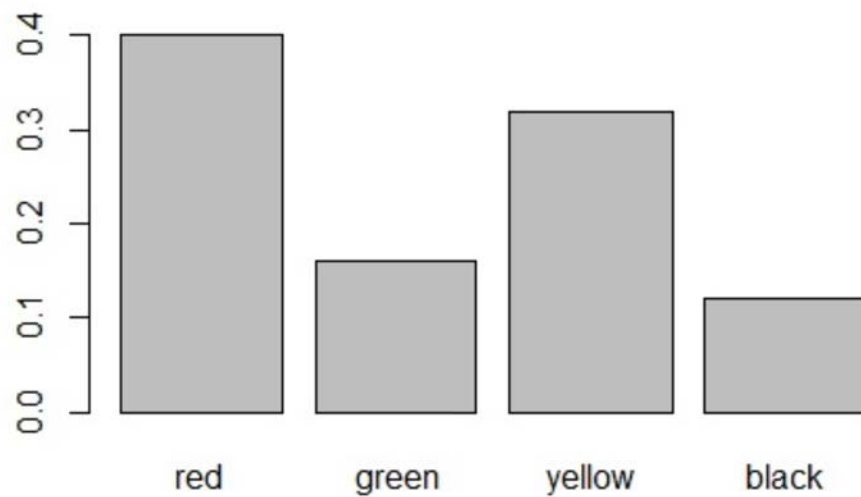
```
> names(tuni)=c("red","green","yellow","black")
> barplot(tuni)
```



شکل ۲-۱۸: نام‌گذاری ستون‌ها در نمودار میله‌ای

در کدها و شکل زیر نام‌گذاری ستون‌ها در نمودار میله‌ای به همراه محاسبه درصدی در ستون عمودی را مشاهده می‌نمایید.

```
> barplot(tuni/length(tuni))
```

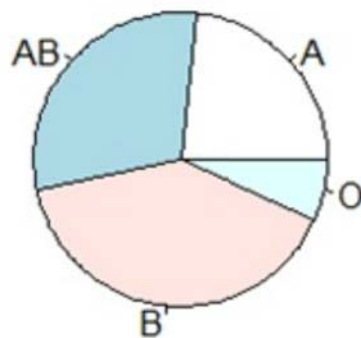


شکل ۲-۱۹: نام‌گذاری ستون‌ها در نمودار میله‌ای به همراه محاسبه درصدی

```
> tuni/length(uni)
   red green yellow black
0.40  0.16  0.32  0.12
```

نمودار و کدهای مربوط به نمودار دایره‌ای در قطعه کد زیر آورده شده است.

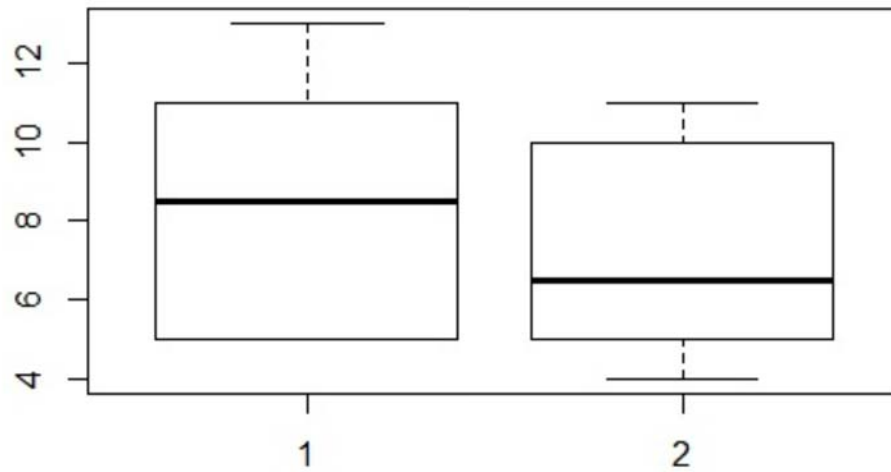
```
> blood=sample(c("A","B","AB","O"),30,replace=T)
> blood
 [1] "AB" "A"  "AB" "AB" "B"  "B"  "O"  "A"  "AB" "A"
[11] "B"  "B"  "AB" "B"  "A"  "B"  "AB" "O"  "B"  "AB"
[21] "B"  "B"  "B"  "A"  "B"  "A"  "A"  "AB" "AB" "B"
> tblood=table(blood)
> tblood
blood
 A AB B O
 7  9 12 2
> pie(tblood)
```



شکل ۲-۲۰: نمودار دایره‌ای

ترسیم نمودار جعبه‌ای را در قطعه کد زیر مشاهده می‌نمایید.

```
> x=c(5,5,5,13,7,11,11,9,8,9)
> y=c(11,8,4,5,9,5,10,5,4,10)
> boxplot(x,y)
```



شکل ۲-۲۱: نمودار جعبه‌ای

فصل سوم

تخلیص اکتشافی داده ها

۳-۱- آماده سازی داده ها برای داده کاوی

در یک تعریف، داده کاوی مرحله ای از فرایند کشف دانش، الگوها و یا مدل ها را در میان انبوهی از داده ها تعریف می شود. داده کاوی یک حیطه علمی بین علوم متفاوتی همچون؛ آمار، یادگیری ماشینی، پایگاه های اطلاعاتی و مانند آن می باشد و ماده اولیه به کاررفته در آن، داده می باشد. از این رو اولین مرحله از عملیات داده کاوی خوب، استفاده و دسترسی به داده های اولیه خوب و مناسب است؛ که به آن آماده سازی یا پیش پردازش داده ها می گویند. درواقع برای کشف دانش به کمک داده کاوی نیاز است که فعالیت های مقدماتی انجام گیرد؛ که مجموعه این اقدامات را آماده سازی داده ها گویند.

آماده سازی داده ها دارای اهمیت بالایی می باشد به دلیل این واقعیت که فقدان داده موجب با فقدان کیفیت در نتایج کاوش است و ورودی ناقص و اشتباه خروجی اشتباه را به دنبال دارد. متأسفانه بسیاری اهمیت آماده سازی داده ها را فراموش کرده و یا آن را کم اهمیت به حساب می آورند. از این رو تلاش های بسیاری برای بسط و توسعه آماده سازی داده ها در داده کاوی انجام گرفته است. وظیفه اصلی پیش پردازش داده ها؛ سازمان دهی داده ها در ساختار مناسب برای داده کاوی می باشد که در ادامه به برخی از آن ها اشاره می نماییم.

۳-۲- پیش پردازش داده ها

فرآیند پیش پردازش داده ها شامل موارد زیر می باشد:

۱- درک و فهم داده ها: به واسطه این موضوع، می توان مراحل بعدی داده کاوی را بهبود داد. به این معنی که می توان جامع و مانع بودن داده ها، هدف و کاربرد داده ها و مواردی از این دست را درک کرد تا ضمن افزایش قابلیت اطمینان به عملیات داده کاوی، سرعت انجام کار نیز افزایش یابد.

۲- پاک سازی داده ها: این مرحله دربرگیرنده پر کردن داده های گم شده، هموار کردن نویزها، شناخت و حذف داده های پرت و برطرف کردن ناسازگاری ها می باشد.

۳- یکپارچه نمودن داده‌ها: این موضوع، معمولاً به هنگام تلفیق چندین پایگاه داده یا فایل پراهمیت شمرده می‌شود. مسائلی همانند افزونگی داده‌ها در این دسته جای می‌گیرد.

۴- تبدیل داده‌ها: در این مرحله از پیش‌پردازش داده‌ها، با عملیاتی همچون نرمال‌سازی، تغییر و تجمیع داده‌ها مواجه می‌شویم.

۵- کاهش داده و کاهش بعد: در این مرحله قصد داریم که به حجم کوچک‌تری از داده‌ها دست‌یابیم. نکته مهم در این مرحله از آماده‌سازی داده‌ها، این مسئله است که دست‌یابی به نتایج تحلیلی مشابه با اصل و تمام داده‌ها تضمین گردد؛ چرا که در غیر این صورت این کاهش اثر مثبتی برای ما در پی نخواهد داشت.

پاک‌سازی داده‌ها

در پاک‌سازی داده‌ها با نوعی تمیز نمودن داده‌ها روبرو می‌باشیم. پاک‌سازی داده‌ها، عبارت است از فرآیند تشخیص و حذف یا تصحیح اطلاعات در یک پایگاه داده که دارای برخی خطاهاست. اهمیت این فرایند تا آن جا می‌باشد که قیمت برخی ابزارها و نرم‌افزارهای مربوط به تمیز کردن داده‌ها بسیار گران می‌باشد.

ابعاد پاک‌سازی داده‌ها

(۱) **اکتساب داده‌ها:** در این گام، مباحثی همچون شناخت نقش، نوع و جزئیات کاربردی داده مورد تحلیل و بررسی قرار می‌گیرد. همچنین در مواردی که نیاز است تا انبارهای داده و بازارهای داده‌ای ساخته شوند ممکن است نیازمند ساخت فراداده برای داده‌هایمان باشیم.

(۲) **پر کردن داده‌های مفقوده:** گاهی با مشکل فقدان داده‌ها روبرو هستیم. به دلایل مختلفی داده‌ها ممکن است مفقوده گردند. از جمله می‌توان به موارد روبرو

اشاره نمود: داده‌ها هنگام ورود حائز اهمیت نبوده‌اند، در تجهیزات ثبت داده‌ها ایراد وجود دارد، به خاطر دشواری فهم، داده وارد نشده است. داده موردنظر، با داده دیگر ناسازگار بوده و به ناچار حذف شده است. باید بررسی نمود که چگونه باید با این مشکل برخورد کرد. انتخاب روش برخورد با داده‌ها که وجود ندارد؛ بستگی به شرایط مسئله دارد. یکی از شرایط مؤثر در این تصمیم‌گیری‌ها آن است که درک نماییم چه عامل یا عواملی دلیل فقدان داده‌ها بوده است.

برخی داده‌های مفقوده کاملاً از نظر آماری غیر وابسته به داده‌هایی است که تاکنون مشاهده شده‌اند؛ این داده‌ها را مفقودشده‌ی کاملاً تصادفی می‌گویند. در برخی موارد نیز مقادیر مفقوده، تصادفی هستند و به تعدادی از متغیرها یا طبقه داده‌های پیش‌بینی کننده مشروط می‌باشند. دسته‌ای دیگر از داده‌های مفقوده نیز، غیرقابل چشم‌پوشی هستند؛ به این معنا که این نوع داده‌های مفقوده به کمک داده‌های مشاهده‌شده قبل از خود قابل نقل هستند. این قبیل تفاوت‌ها سبب می‌شود که روش‌های متفاوتی برای برخورد با مقادیر مفقوده مورد استفاده قرار گیرد:

- **حذف نمودن رکورد:** روش حذف نمودن برای عملیات دسته‌بندی و بر روی داده‌های طبقه‌ای صورت می‌گیرد. نکته‌ای که باید مدنظر باشد آن است که اگر تعداد داده‌های مفقوده فراوان باشد استفاده از این روش سبب می‌شود که از حجم نمونه به شدت کاسته شود. این مشکل به شکل ویژه هنگامی اثرات خود را بر نتایج نشان می‌دهد که برخی از نمونه داده‌ها بسیار نادر و کم بوده و حذف رکورد مربوط به آن‌ها، سبب از دست دادن نمونه‌ای با ارزش شود. از این رو حذف رکورد بایستی در موارد خاص انجام گیرد.
- **حذف نمودن مشاهده:** این انتخاب زمانی روی می‌دهد که رکورد دارای مقدار مفقوده، مورد نیاز باشد؛ چراکه در غیر این صورت بود یا نبود مقدار برای ما مهم نیست. البته در صورت نیاز به استفاده از این روش باید به یاد داشته باشیم که محاسبات انجام‌شده برای مقادیر آمار

توصیفی؛ از قبیل میانگین، واریانس و کوواریانس به اندازه‌های متفاوت نمونه مربوط خواهد شد که تأثیر آن باید مدنظر باشد.

- **پرنمودن دستی:** استفاده از این روش چندان کاربردی نمی‌باشد؛ زیرا پیدا کردن و اصطلاحات لازم زمان‌بر است. البته در برخی مواقعی تنها راه حل ممکن است. مثلاً، دو نام و آدرس فرضی علی محمدی ساکن اصفهان و علیرضا محمدی ساکن اصفهان را در نظر بگیرید. اگر این دو نفر دقیقاً یکی بوده و تمامی سایر مشخصات آن‌ها نیز یکی باشند؛ تشخیص و رفع این مشکل ممکن است به کمک کامپیوتر مقدور نباشد. البته این موارد بسیار محدود است.

- **پرنمودن با استفاده از مقدار ثابت سراسری:** در این موارد مقادیر مفقوده با مقداری هم چون Unknown، پر می‌شوند. مسئله‌ای که در این صورت با آن مواجه خواهیم بود آن است که، ممکن است در حجم بالای داده‌ها ویژگی مقداردهی شده با این مورد، جزء داده‌های محاسباتی محسوب شده و در محاسبات منظور گردد؛ و به این شکل ایجاد خطا نماید. به‌علاوه هنگامی که عملیات پاک‌سازی داده‌ها برای ساخت انبار داده استفاده می‌شود، این روش انتخاب مناسبی نخواهد بود.

- **پرنمودن با استفاده از میانگین ویژگی:** این روش یکی از بهترین روش‌ها در این زمینه می‌باشد. استفاده از این روش ممکن است سبب شود تا به دلیل تأثیر مقادیر نسبت داده شده به این ویژگی، نتایج به دست آمده به نفع این میانگین بایاس شود؛ حتی ممکن است اتخاذ این روش سبب حذف یا انتقال رکوردهای مربوط به یک دسته خاص از داده‌ها به سمت دسته نتایج دیگری شده و یک دسته مهم و واقعی از نتایج را نادیده بگیریم.

- **پرنمودن با استفاده از مقادیر با احتمال بیشتر:** این روش که یکی از پرکاربردترین روش‌ها می‌باشد، شامل روش‌های استنتاجی و به‌کارگیری فرمول‌های بیزین، رگرسیون و درخت تصمیم است. به نوعی در این روش‌ها بر اساس استنتاج منطقی که مبتنی بر نوع اطلاعات

موجود است؛ عمل پیش‌بینی صورت می‌گیرد. علاوه بر این موارد؛ روش‌های دیگری همانند پر کردن مقادیر با میانگین ویژگی برای دسته‌های مشابه، نیز وجود دارد که چندان متداول نمی‌باشند. البته باید اشاره نماییم که، نوع داده‌ها و شناخت آن‌ها قبل از پر کردن مقادیر مفقوده ضروری است. مثلاً نمی‌توان داده طبقه‌ای را با روش میانگین ویژگی پرکرد، چراکه میانگین برای این نوع داده‌ها قطعاً بی‌معنا خواهد بود. درک این موارد در مواجهه با این قبیل مشکلات بسیار حائز اهمیت می‌باشد.

- **پرنمودن با استفاده از روش میانگین همسایه‌ها:** پرکردن با میانگین همسایه‌ها یکی از روش‌هایی است که در مورد داده‌های از دست رفته مورد استفاده قرار می‌گیرد. ابتدا با محاسبه فاصله چند تا از همسایگان که دارای رکوردهای مشابه با رکورد فعلی می‌باشند یافت شده و سپس مقادیر مفقوده با میانگین آن‌ها در این چند رکورد عوض می‌شود این روش از کارآمدترین روش‌ها می‌باشد که از همه اطلاعات رکورد استفاده می‌کند.

۳) حل نمودن مشکل افزونگی

در بسیاری از موارد در هنگام کار کردن با داده‌ها، آن‌ها را از منابع و پایگاه داده‌های مختلف در کنار یکدیگر جمع می‌کنیم. در داده کاوی این موضوعات در قالب ساخت بازارهای داده و انبارهای داده مورد بررسی قرار می‌گیرد که نیازمند بحثی گسترده می‌باشد. به هر حال پایگاه‌های مختلف داده هنگامی که گسسته از یکدیگر طراحی می‌شوند؛ به ناچار دارای فیلدهای و داده‌های یکسانی هستند که اتفاقاً داده‌های حیاتی پایگاه داده‌ها و سیستم‌ها می‌باشد. برای این گونه مسائل روش‌های متعددی وجود دارد که برخی از آن‌ها همچون افزونگی معمول در پایگاه داده‌ها را با آزمون‌های مختلف آماری می‌توان حل کرد.

۴) یکسان نمودن قالب‌ها

بحث یکسان نمودن قالب‌ها نیز یکی از مسائل مهم به هنگام جمع‌آوری داده‌ها می‌باشد. برای درک پنهان و مشکل بودن تشخیص این موارد بهتر است به مثالی اشاره نماییم. به عنوان مثال در مورد فیلد تاریخ، فرمت‌های مختلفی برای ذخیره داده‌ها استفاده می‌شود؛ که در صورت عدم دقت به این مسئله، داده‌کاو، اثربخشی لازم را به دنبال نداشته و بازسازی انبارهای داده ساخته شده نیز، هزینه بالایی به دنبال خواهد داشت. راه حل این مشکل عموماً درگرو درک داده‌های موجود در پایگاه‌های مختلف، از قبل جمع‌آوری آن‌ها می‌باشد.

۵) تصحیح داده‌های ناسازگار

تصحیح داده‌های ناسازگار مرتبط با تناقض در داده‌ها می‌باشد و از جمله مواردی است که نیازمند تجربه و صرف وقت بسیار است. به عنوان مثال در فیلد تاریخ تولد و سن مربوط به یک مشتری خاص، در صورتی که همخوانی لازم وجود نداشته باشد، ناسازگاری محسوب می‌شود. این شکل از خطاها ممکن است به دلیل استفاده از منابع مختلف داده و در زمان ترکیب دو منبع مختلف از داده‌ها روی دهد. اما مشکل عمده‌ای که با آن مواجه می‌شویم و تشخیص آن بسیار مشکل است؛ تعیین ناسازگاری‌های نهفته است. به عنوان مثال اگر به دنبال کشف الگو در مورد مسائل مربوط به بیمارستان داری باشید و قیمت مربوط به بیمارستان‌های دنیا را از منابع مختلف جمع‌آوری کنید، جدای از بحث تبدیل نرخ‌ها و رفع ناسازگاری مربوط به مسائل خاص ارزی هر کشور، باز هم قیمت بیمارستان‌ها نمی‌تواند ملاک مناسبی باشد؛ چراکه لازم است تا خدماتی همچون، سرویس تغذیه، امکانات رفاهی و سایر خدماتی را که در جاهای مختلف به شیوه‌های مختلف ارائه می‌شود، مدنظر داشت. به عبارتی قیمت هر شب اقامت در بیمارستان در کنار نوع، شیوه و مقدار ارائه خدمات جانبی آن معنا پیدا می‌کند. روش عمده و اصلی در حل ناسازگاری‌ها درک ماهیت داده‌ها است. اما در مواردی نیز ناسازگاری‌ها را که حاصل جمع‌آوری چند منبع مختلف بوده و بیانگر افزونگی داده‌هاست؛ می‌توان با کمک روش‌های آماری برطرف کرد.

۶) برخورد با داده‌های نویز و پرت

قبل از هر چیز دیگر در این جا لازم است تا تفاوت بین داده‌های نویز و داده‌های پرت را درک کنیم. این تفاوت در این نکته است که داده‌های نویز در اثر خطاهای تصادفی بروز می‌کنند. از جمله عواملی که سبب بروز داده نویز می‌شود؛ می‌توان به موارد زیر اشاره کرد:

- به کاربردن ابزارها و متدهای معیوب در جمع‌آوری داده
- مشکلات حین ورود داده
- محدودیت فناوری

این نکته بسیار مهم می‌باشد که تشخیص نویز یا پرت بودن مهم‌تر از حل این مشکل است. تشخیص اشتباه همواره درمان اشتباه به همراه دارد. از این رو بایستی مطمئن شد که اولاً آنچه گمان می‌کنیم مثلاً داده نویز است؛ واقعاً داده نویز باشد تا مبادا به عنوان انجام اصلاح در داده، داده‌ای با ارزش را تغییر دهیم. برای مواجهه با داده نویز و هموار کردن داده‌ها، روش‌های مختلفی وجود دارد، که از جمله می‌توان به گسسته سازی داده‌ها، رگرسیون، خوشه‌بندی و روش‌های ترکیبی بازرسی ماشین و انسان اشاره کرد. البته برخی از این روش‌ها، هم چون استفاده از رگرسیون و خوشه‌بندی در داده‌های پرت نیز کاربرد دارد.

تلخیص توصیفی داده‌ها

نتایج حاصل از تلخیص توصیفی داده‌ها می‌تواند به شکل گرافیکی درآمده و درک و توصیف داده‌ها را میسر سازد. از جمله گراف‌هایی که برای نمایش گرافیکی تلخیص توصیفی داده‌ها استفاده می‌شود می‌توان به هیستوگرام، چنک، نمودار پراکندگی، نمودار لویس، نمودار جعبه‌ای، نمودار میله‌ای و ... اشاره کرد.

نرم افزارهای مختلف آماری بسیاری از نرم افزارهای کاربردی داده کاوی با فراهم کردن امکان نمایش گرافیکی داده های توصیفی تلخیص شده، در عملیات آماده سازی داده ها سهیم شده اند.

گسسته سازی

هدف از این روش آن است که داده ها را برحسب قواعدی در دسته بندی هایی قرار دهیم؛ و دسته ای را که تعداد داده های موجود در آن بسیار کم باشد، کنار می گذاریم. توجیه آن است که این داده ها با دیگر داده ها تفاوت داشته و بنا به اشتباهاتی به وجود آمده اند. فراموش نکنیم که این روش نبایستی حذف نمونه های ارزشمند را در تشخیص الگوها به همراه داشته باشد. از این رو تأکید می کنیم که تشخیص نویز یا پرت بودن داده؛ از حل مشکل آن مهم تر است.

رگرسیون

رگرسیون تنها روشی است که در صورت مهیا بودن شرایط استفاده، علاوه بر مشخص نمودن داده مغشوش برای آن مقدار هم پیشنهاد می دهد. رگرسیون بر روی تعداد مختلف ویژگی قابل اجراست. در صورتی که بر روی دو محور متعامد تنها دو ویژگی را در نظر داشته باشیم خروجی رگرسیون برازش خطی برای تطبیق نقاط این دو ویژگی است که به آن رگرسیون خطی می گویند. در صورتی که تعداد بیشتری متغیر و با انواع ارتباط خطی و غیرخطی داشته باشیم رگرسیون ما یک رگرسیون چند متغیره و یا غیرخطی خواهد بود. قبل از استفاده از روش رگرسیون بهتر است تا ویژگی هایی را که پیش بینی کننده خوبی برای متغیر وابسته هستند؛ انتخاب کنیم. این کار یا بر اساس نظر خبره و یا به کمک تست های مختلف آماری از قبیل تست های جهت و میزان همبستگی صورت می گیرد. مسئله مهم برای استفاده از رگرسیون آن است که این روش به داده های پرت حساس است. از این رو می توان با تعیین اولیه برخی نقاط پرت توسط این روش یا هر روش دیگر و حذف آن ها دوباره رگرسیون را تکرار کرد تا مرحله ای که تعدادی داده مغشوش مشخص و مقادیر پیش بینی شده آن با نظر خبره تأیید گردد. نکته مهم دیگر آن که، دامنه

استفاده از رگرسیون محدود به داده‌های عددی نیست و با انجام مقدماتی می‌توان برای داده‌های گسسته طبقه‌ای و ترتیبی نیز مورد استفاده قرار گیرد. از این قبیل موارد می‌توان به رگرسیون لجستیک و پواسون اشاره کرد که بیان جزییات مربوط به آن‌ها در این مقوله نمی‌گنجد.

خوشه‌بندی

از خوشه‌بندی نیز می‌توان برای تعیین داده‌ها و خوشه‌هایی که می‌تواند پرت بوده و یا برای مسئله مورد بررسی ما کاربرد نداشته باشد استفاده کرد. به عبارتی یکی از کاربردهای خوشه‌بندی تعیین داده‌های فضای مسئله مورد بررسی است. همان‌گونه که می‌دانید در خوشه‌بندی، مجموعه‌ای از داده‌ها که بر اساس ویژگی‌های مختلف بیشترین شباهت دارند در کنار یکدیگر قرار می‌گیرند. همان‌گونه که در شکل زیر نیز می‌بینید؛ برخی داده‌ها بیرون خوشه‌ها قرار گرفته و می‌توان آن‌ها را کنار گذارد. البته همواره بایستی احتیاط‌های لازم را مدنظر داشت.

پیش‌پردازش و آماده‌سازی داده‌ها

مرحله آماده‌سازی داده‌ها مهم‌ترین و زمان‌برترین مرحله در پروژه‌های داده‌کاوی است. از آنجا که داده‌ها در این پروژه‌ها ورودی پروژه هستند هر قدر این ورودی دقیق‌تر باشد، خروجی کار دقیق‌تر خواهد بود. یعنی ما از پدیده «ورودی نامناسب، خروجی نامناسب» دور می‌شویم. هرچند به هر حال می‌توان یک روش را بر روی داده‌ها اعمال کرد و سپس بر اساس عملکرد پیش‌بینی تخمینی آن نتایج را ارزیابی نمود، ولیکن این کار به هیچ‌وجه موجب کاهش اهمیت وظیفه اولیه ما یعنی توجه دقیق به آماده‌سازی داده‌ها نمی‌شود. باینکه روش‌های پیش‌بینی ممکن است توانایی‌های نظری قوی داشته باشند ولی توان همه آن‌ها در عمل با توجه به وضعیت داده‌ها در مقایسه با فضای نامحدود جستجو، محدود می‌شود.

فرق داده پرت با نویز

یکی از چالش‌ها در تشخیص داده‌های پرت، وجود نویز می‌باشد. نویز با داده پرت متفاوت است. نویز، خطا (یا واریانس) تصادفی در داده‌هاست که باید قبل از

تشخیص داده پرت حذف شود. اما داده‌های پرت، داده‌هایی هستند که آن‌قدر اختلاف زیادی با داده‌های ما دارند که به نظر می‌رسد با روش دیگری ساخته شده‌اند، برای مثال کشف جرم یا کشف خریدهای مکرر یا گران را می‌توان با استفاده از روش‌های تشخیص داده‌های پرت تشخیص داد. بنابراین تشخیص داده‌های پرت از داده‌های نویز مهم است. نحوه مواجهه با این دو از نظر الگوریتمی مشابه است ولی بستگی به صورت مسئله دارد.

مقایسه روش‌های کاهش بعد برای بصری سازی

در شکل زیر انواع روش‌های کاهش بعد برای بصری سازی را به همراه کاربردها، مزایا و معایب هر کدام مشاهده می‌فرمایید.

روش	نوع	محاسبات	کاربرد دیگر	مزایا	معایب
PCA	خطی آمار چند متغیره	حداکثر واریانس بردار ویژه	پیش‌پردازش (کاهش بعد- متغیرهای مستقل)	ساده و سرراست شهود هندسی	تصویری خطی با ازدست رفتن زیاد اطلاعات
MDS	غیرخطی آمار چند متغیره	حداقل مربع تفاضل گرادیان		بهترین تناسب فاصله (ذاتی)	ماتریس فواصل بسیار بزرگ و محاسبه کند
SOM	غیر خطی شبکه عصبی	شبکه عصبی بدون ناظر- شروع تصادفی	خوشه‌بندی پیش‌پردازش	محاسبه سریع در بعد و داده زیاد	برای تعداد مشاهده کم مناسب نیست

شکل ۳-۱: مقایسه روش‌های کاهش بعد برای بصری سازی

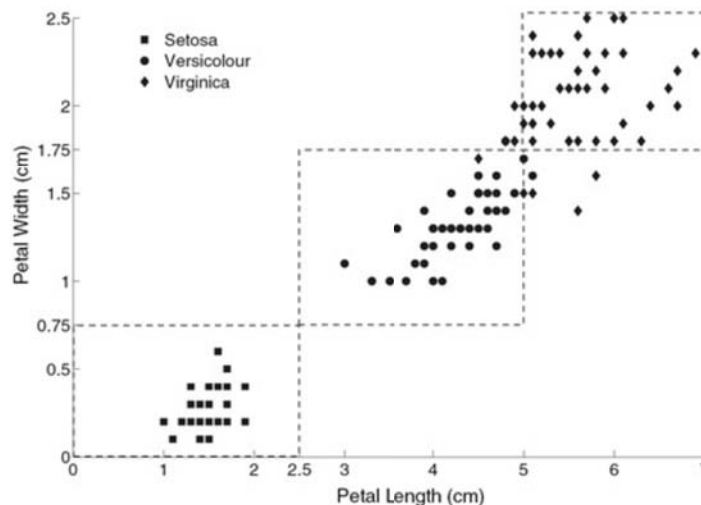
۳-۳- تحلیل اکتشافی داده‌ها در زبان R

معرفی مجموعه داده مربوط به گل‌های زنبق

مجموعه داده مربوط به گل زنبق در بسیاری از پژوهش‌ها برای دسته‌بندی مورد استفاده قرار می‌گیرد. این مجموعه داده دارای ۵۰ نمونه از سه دسته گل زنبق است. این مجموعه داده دارای ۵ خصوصیت به شرح زیر می‌باشد:

- طول کاسبرگ در سانتی‌متر
- عرض کاسبرگ در سانتی‌متر
- طول گلبرگ در سانتی‌متر
- عرض گلبرگ در سانتی‌متر
- دسته: Setosa, Versicolour, و Virginica.

```
> str(iris)
'data.frame': 150 obs. of 5 variables:
 $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species : Factor w/ 3 levels "setosa","versicolor",...
 : 1 1 1 1 1 1 1 1 1 1 ...
```



شکل ۳-۲: پراکندگی داده‌های Iris با استفاده از دو متغیر

ذخیره و فراخوانی داده‌ها در زبان برنامه‌نویسی R

داده‌ها در R را می‌توان در فایلی با فرمت Rdata. با استفاده از تابع `save()` ذخیره نمود و سپس مجدداً با تابع `load` فراخوانی نمود. تابع `rm()` اشیا به وجود آمده را حذف می‌نماید.

```
> data <- 1:20
> save(data, file="d:/r/file.Rdata")
> rm(data)
> load("d:/r/file.Rdata")
> print(data)
[1] 1 2 3 4 5 6 7 8 9 10
    11 12 13 14 15 16 17 18 19 20
```

ورود و ذخیره نمودن فایل‌های CSV.

در مثال زیر دیتافریم را در قالب یک فایل CSV. ذخیره نموده و مجدداً در دیتافریم جدیدی به نام فراخوانی می‌نماییم.

```
> input1 <- 1:5
> input2 <- (1:5) * 5
> input3 <- c("knowledge", "information",
              "data", "wisdom", "think")

> box1 <- data.frame(input1, input2, input3)
> names(box1) <- c("col1", "col2", "col3")
> write.csv(box1, "d:/r/databox.csv", row.names = FALSE)
> box2 <- read.csv("d:/r/databox.csv")
> print(box2)
  col1 col2      col3
1    1    5 knowledge
2    2   10 information
3    3   15      data
4    4   20    wisdom
5    5   25     think
```

اکتشاف دانش

در این بخش مثال‌هایی از کشف دانش به‌وسیله زبان R را بررسی می‌نماییم که شامل برخی مفاهیم ابتدایی آماری و نمودارهای متنوعی از چارت‌ها و هیستوگرام می‌باشد. همچنین در این بخش از مجموعه داده Iris برای نمایش اکتشاف دانش استفاده می‌کنیم. ابتدا اندازه و ساختار داده را به‌وسیله توابع `dim()` و `name()` بررسی نموده و سپس به‌وسیله توابع `str()` و `attributes()` ساختار و خصوصیت‌های داده‌ها را نمایش می‌دهیم.

```
> str(iris)
'data.frame': 150 obs. of 5 variables:
 $ Sepal.Length: num 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species : Factor w/ 3 levels "setosa","versicolor",...
 : 1 1 1 1 1 1 1 1 1

> attributes(iris)
$names
[1] "Sepal.Length" "Sepal.Width" "Petal.Length"
    "Petal.Width" "Species"

$row.names
 [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
[19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
[37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
[55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
[73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87
[91] 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105
[109] 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123
[127] 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141
[145] 145 146 147 148 149 150

$class
[1] "data.frame"

> names(iris)
[1] "Sepal.Length" "Sepal.Width" "Petal.Length"
    "Petal.Width" "Species"

> dim(iris)
[1] 150 5
```


همچنین با استفاده از توابع `head()` یا `tail()` می‌توانید ۵ سطر ابتدایی و یا انتهایی مجموعه داده‌ها را مشاهده نماییم.

```
> iris[1:7,]
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5         1.4         0.2  setosa
2          4.9         3.0         1.4         0.2  setosa
3          4.7         3.2         1.3         0.2  setosa
4          4.6         3.1         1.5         0.2  setosa
5          5.0         3.6         1.4         0.2  setosa
6          5.4         3.9         1.7         0.4  setosa
7          4.6         3.4         1.4         0.3  setosa

> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5         1.4         0.2  setosa
2          4.9         3.0         1.4         0.2  setosa
3          4.7         3.2         1.3         0.2  setosa
4          4.6         3.1         1.5         0.2  setosa
5          5.0         3.6         1.4         0.2  setosa
6          5.4         3.9         1.7         0.4  setosa

> tail(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
145          6.7         3.3         5.7         2.5 virginica
146          6.7         3.0         5.2         2.3 virginica
147          6.3         2.5         5.0         1.9 virginica
148          6.5         3.0         5.2         2.0 virginica
149          6.2         3.4         5.4         2.3 virginica
150          5.9         3.0         5.1         1.8 virginica
```

همچنین می‌توانیم مقادیر مربوط به یک ستون خاص را استخراج نماییم. برای مثال ۱۰ مقدار اولیه از `Sepal.Length` را می‌توانیم از طریق کد زیر محاسبه نماییم.

```
> iris$ Petal.Length[1:5]
[1] 1.4 1.4 1.3 1.5 1.4
> iris[1:5, "Petal.Length"]
[1] 1.4 1.4 1.3 1.5 1.4
```

اکتشاف متغیرهای خاص

توزیع هر متغیر عددی می‌تواند به وسیله تابع `summary()` بررسی گردد که میزان حداقل، میانه، میانگین و چارک اول و سوم را نشان می‌دهد. همچنین برای هر متغیر مربوط به دسته‌بندی نیز تعداد هر سطح را بیان می‌نماید.

```
> summary(iris)
```

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100	setosa :50
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300	versicolor:50
Median :5.800	Median :3.000	Median :4.350	Median :1.300	virginica :50
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199	
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800	
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500	

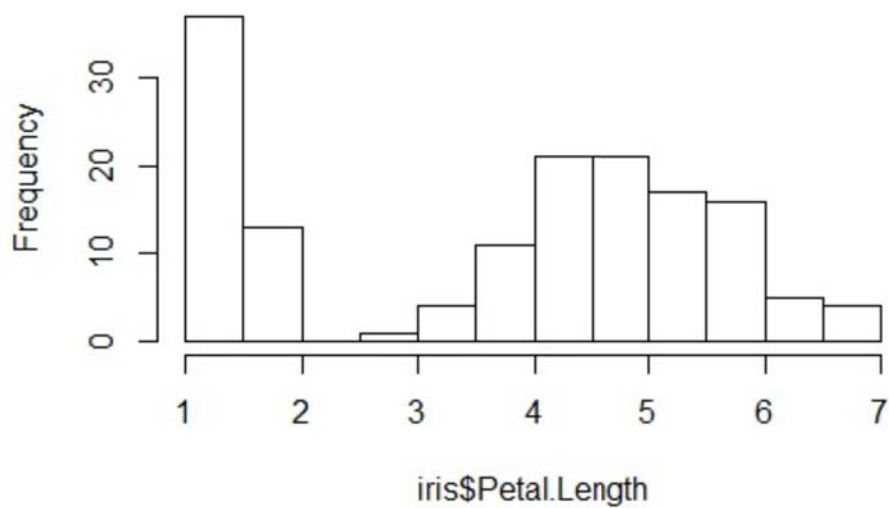
همچنین مقادیر میانگین، میانه و طول را می‌توان به وسیله توابع `mean()`، `median()` و `range()` محاسبه نمود. چارک‌ها و صدک‌ها را نیز می‌توان به وسیله تابع `quantile()` محاسبه کرد.

```
> quantile(iris$Petal.Length)
 0%  25%  50%  75% 100%
1.00 1.60 4.35 5.10 6.90
> quantile(iris$Petal.Length, c(.2, .4, .55, .94))
 20%  40%  55%  94%
1.500 3.900 4.500 6.006
```

همچنین به وسیله تابع `var()` می‌توان میزان واریانس را به دست آورد و به وسیله تابع `hist()` و `density()` هیستوگرام و میزان فشردگی را برای هر متغیر استخراج نمود.

```
> var(iris$Petal.Length)
[1] 3.116278
> hist(iris$Petal.Length)
```

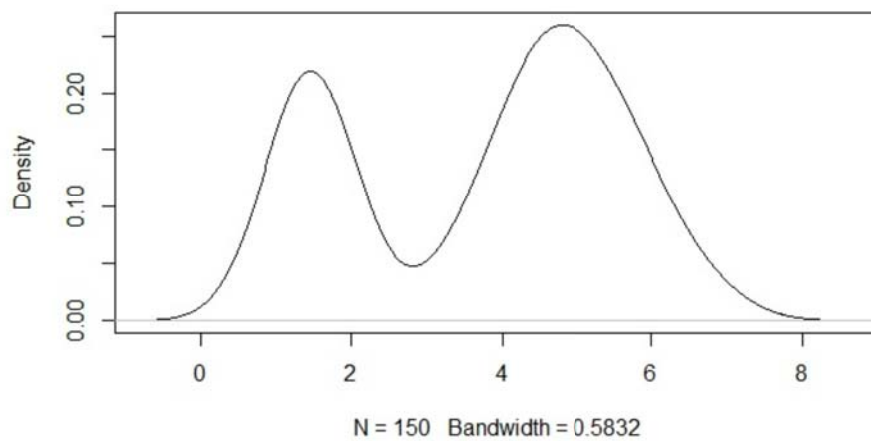
Histogram of iris\$Petal.Length



شکل ۳-۳: نمودار مربوط به هیستوگرام طول

```
> plot(density(iris$Petal.Length))
```

density.default(x = iris\$Petal.Length)

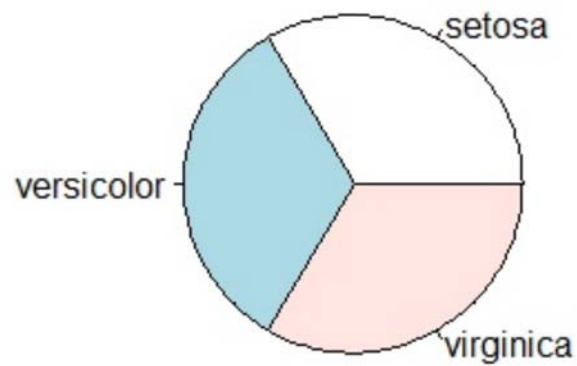


شکل ۳-۴: نمودار مربوط به چگالی

ترسیم نمودار جدولی به وسیله تابع `table()` و ترسیم نمودار برشی توسط تابع `pie()` و ترسیم نمودار ستونی با استفاده از تابع `barplot()` امکان پذیر می باشد.

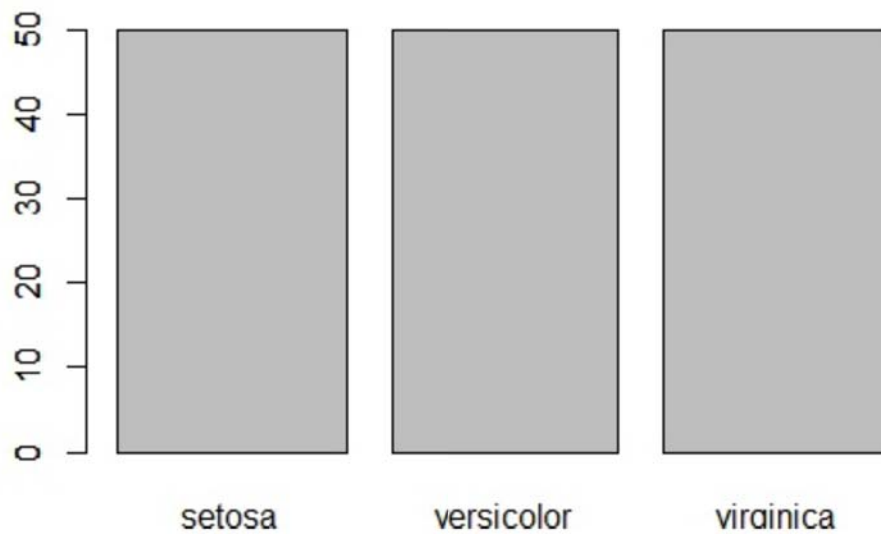
```
> table(iris$Species)

setosa versicolor virginica
    50      50         50
> pie(table(iris$Species))
```



شکل ۳-۵: نمودار برشی داده های Iris

```
> barplot(table(iris$Species))
```



شکل ۳-۶: نمودار ستونی داده های Iris

کاوش متغیرهای چندگانه

پس از بررسی توزیع مربوط به متغیرهای خاص می توان ارتباطات بین دو متغیر را نیز محاسبه نمود. به وسیله دو تابع `cov()` و `cor()` می توان مقدار کوواریانس و همبستگی دو متغیر را بدست آورد.

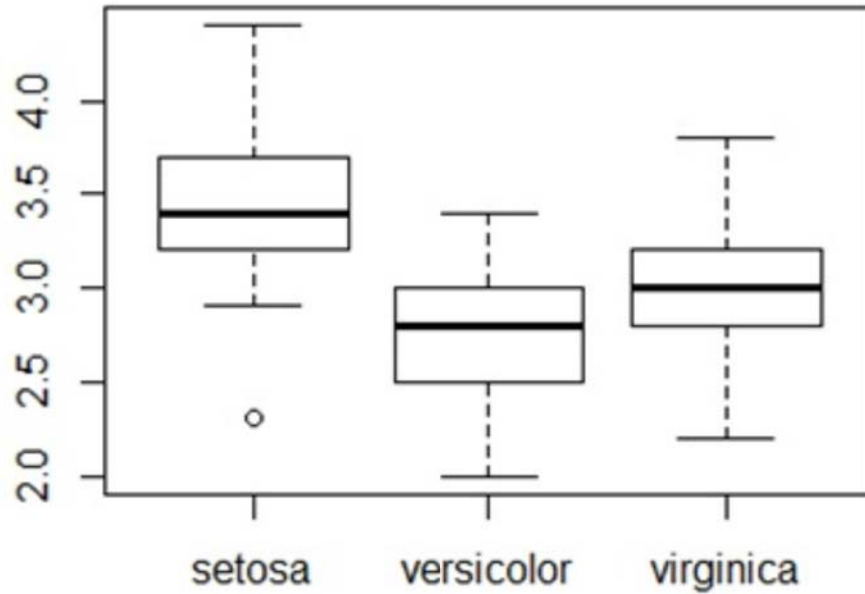
```
> cor(iris$Sepal.Width, iris$Petal.Width)
[1] -0.3661259
> cor(iris[,1:4])
      Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length      1.0000000  -0.1175698    0.8717538    0.8179411
Sepal.Width       -0.1175698   1.0000000   -0.4284401   -0.3661259
Petal.Length      0.8717538  -0.4284401   1.0000000    0.9628654
Petal.Width       0.8179411  -0.3661259    0.9628654    1.0000000
> cov(iris$Sepal.Width, iris$Petal.Width)
[1] -0.1216394
> cov(iris[,1:4])
      Sepal.Length Sepal.Width Petal.Length Petal.Width
Sepal.Length      0.6856935  -0.0424340    1.2743154    0.5162707
Sepal.Width       -0.0424340   0.1899794   -0.3296564   -0.1216394
Petal.Length      1.2743154  -0.3296564    3.1162779    1.2956094
Petal.Width       0.5162707  -0.1216394    1.2956094    0.5810063
\ |
```

همچنین می توان میزان متغیرهای آماری یک خصوصیت را برای هر دسته به وسیله تابع `aggregate()` بررسی نمود.

```
> aggregate(Sepal.Length ~ Species, summary, data=iris)
      Species Sepal.Length.Min. Sepal.Length.1st Qu. Sepal.Length.Median
1   setosa      4.300           4.800                5.000
2 versicolor  4.900           5.600                5.900
3  virginica   4.900           6.225                6.500
      Sepal.Length.Mean Sepal.Length.3rd Qu. Sepal.Length.Max.
1           5.006           5.200                5.800
2           5.936           6.300                7.000
3           6.588           6.900                7.900
```

به وسیله تابع `boxplot()` می‌توان نمودار جعبه‌ای را ترسیم نمود. چارک اول و سوم در این نمودار مشخص است. خط وسط نشانگر میانه می‌باشد. کل جعبه رنج چارکی بین ۲۵ درصد و ۷۵ درصد را نمایش می‌دهد.

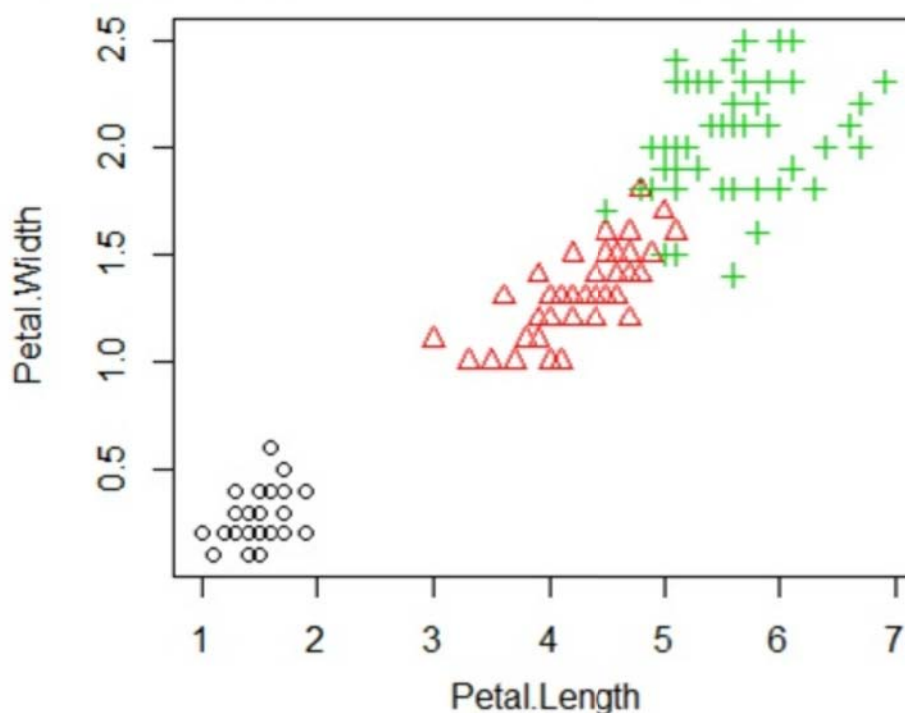
```
> boxplot(Sepal.Width~Species, data=iris)
```



شکل ۳-۷: نمودار جعبه‌ای داده‌های Iris

نمودار پراکندگی نیز بین دو متغیر با استفاده از تابع `plot()` قابل ترسیم می‌باشد. با استفاده از تابع `with()` نیازی به اضافه کردن نام دیتاست به ابتدای متغیرها نمی‌باشد. با استفاده از متغیر `col` و `pch` می‌توان رنگ و سمبل خاص نشانگر را برای هر دسته مشخص نمود.

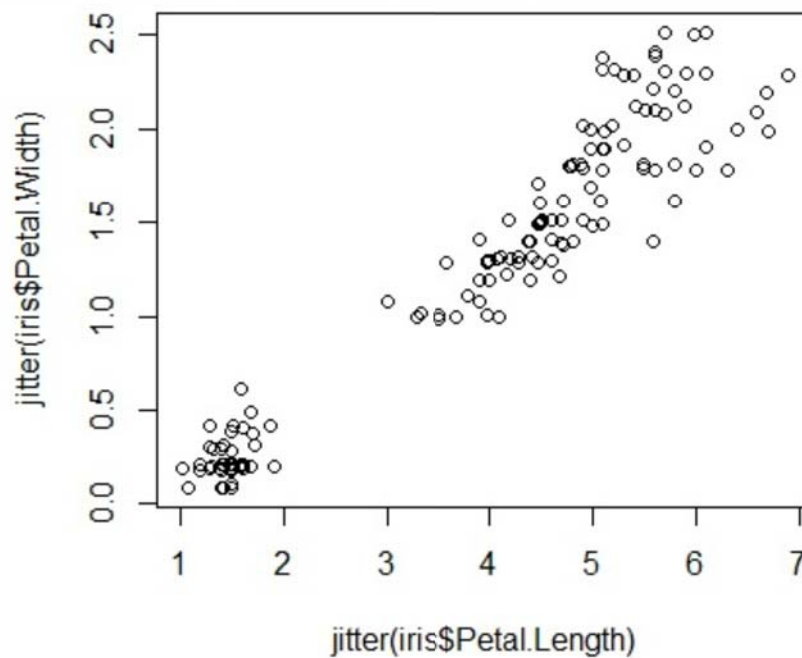
```
> with(iris, plot(Petal.Length, Petal.Width,  
+ col=Species, pch=as.numeric(Species)))
```



شکل ۳-۸: نمودار پراکندگی داده‌های Iris

زمانی که نشانگرهای فراوانی داریم ممکن است برخی با یکدیگر همپوشانی داشته باشند. می‌توانیم از تابع jitter برای اضافه نمودن مقداری نویز پیش از ترسیم داده‌ها استفاده نماییم.

```
> plot(jitter(iris$Petal.Length), jitter(iris$Petal.Width))
```

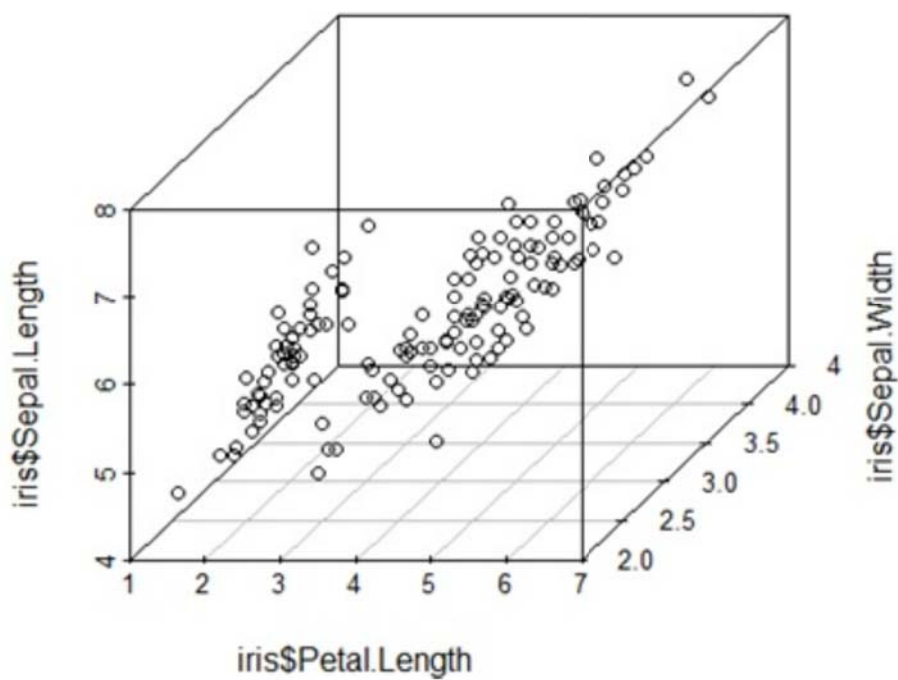


شکل ۳-۹: نمودار پراکندگی داده‌های Iris

اکتشافات بیشتر

این بخش برخی از نمودارها شامل، ترسیم سه‌بعدی، ترسیم چندسطحی، ترسیم شمارشی، ترسیم تعاملی و مختصات موازی را شامل می‌گردد. یک نمودار پراکندگی سه‌بعدی از پکیج scatterplot3d استفاده می‌نماید.

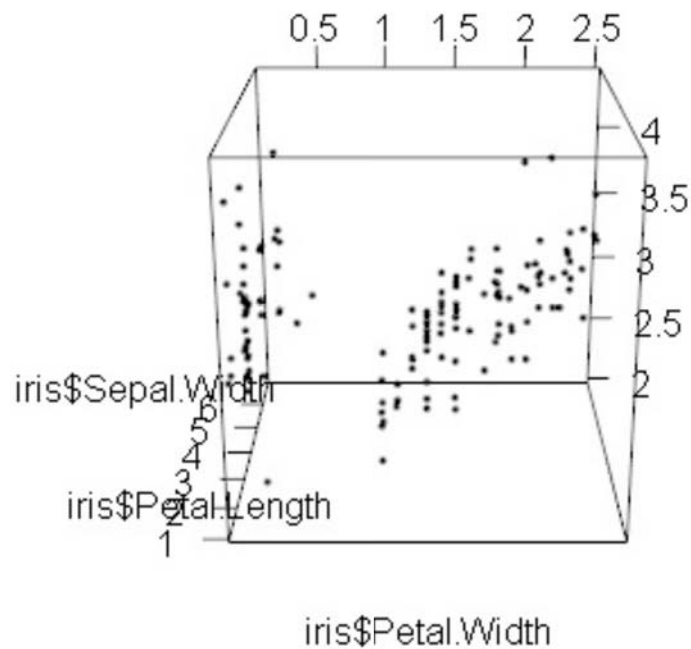
```
> library(scatterplot3d)
> scatterplot3d(iris$Petal.Length, iris$Sepal.Width,
  iris$Sepal.Length)
```



شکل ۳-۱۰: نمودار سه‌بعدی داده‌های Iris با استفاده از بسته scatterplot3d

همچنین بسته rgl نمودار پراکندگی تعاملی را با استفاده از تابع plot3d() امکان‌پذیر می‌نماید.

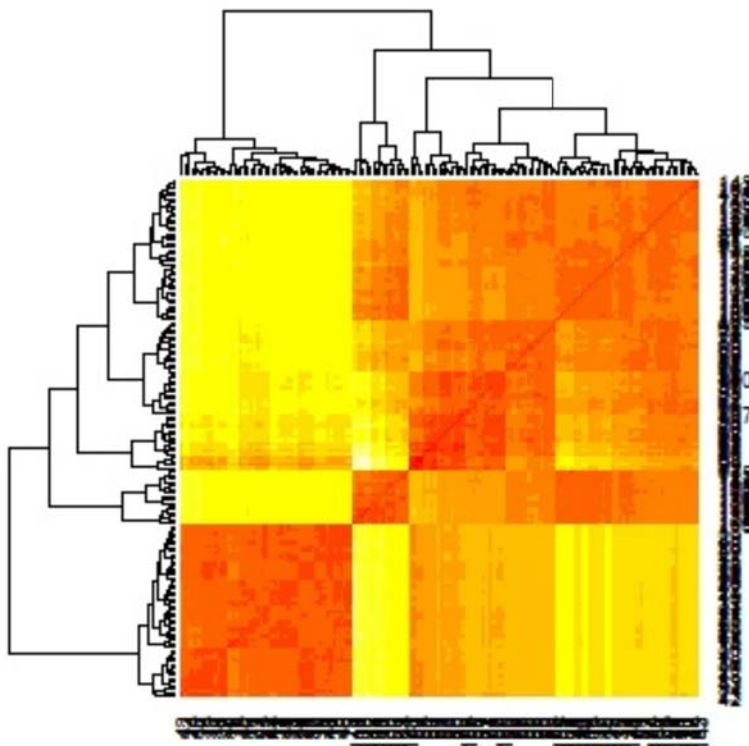
```
> library(rgl)
> plot3d(iris$Petal.Width, iris$Petal.Length,
iris$Sepal.Width)
```



شکل ۳-۱۱: نمودار سه بعدی داده‌های Iris با استفاده از بسته rgl

نقشه حرارتی نمایش‌دهنده یک ماتریس داده‌ای دوبعدی ست که با استفاده از تابع `heatmap()` ترسیم می‌گردد. با استفاده از کد زیر محاسبه تشابهات بین گل‌های متفاوت در داده‌های Iris را با استفاده از تابع `dist()` می‌توان محاسبه نمود و توسط نقشه حرارتی ترسیم نمود.

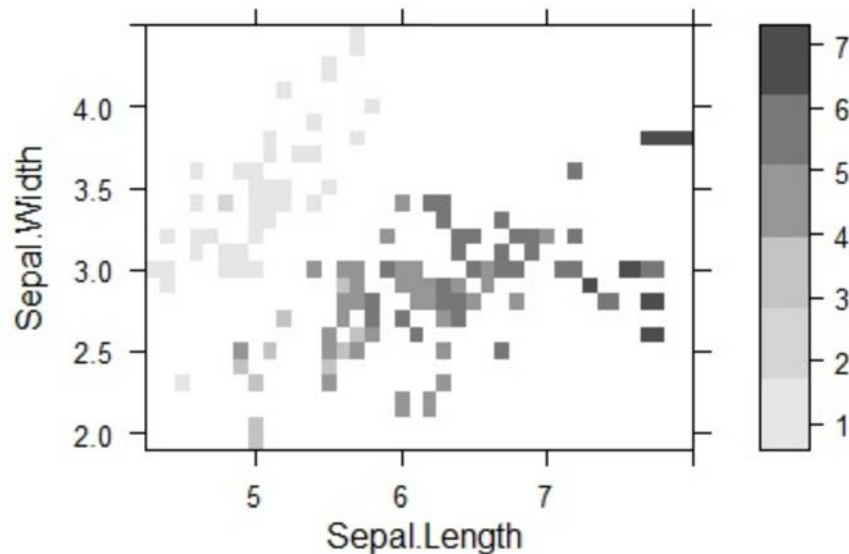
```
> d <- as.matrix(dist(iris[,1:4]))  
> heatmap(d)
```



شکل ۳-۱۲: نمودار heatmap داده‌های Iris

یک نمودار چندسطحی توسط تابع `levelplot()` در پکیج `lattice` قابل دسترسی می‌باشد. توابع `rainbow()` و `grey.colors()` یک بردار با رنگ خاکستری تولید می‌نمایند.

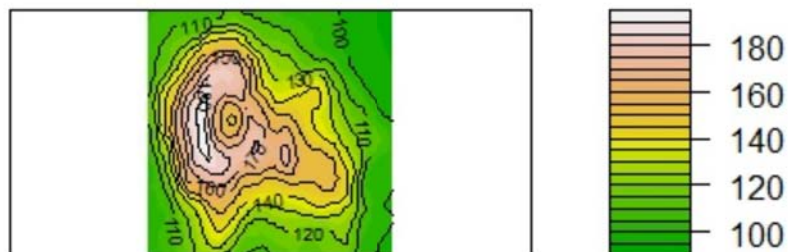
```
> library(lattice)
> levelplot(Petal.Length~Sepal.Length*Sepal.Width,
+ iris, cuts=5, col.regions=grey.colors(7)[7:1])
```



شکل ۳-۱۳: نمودار چندسطحی داده‌های Iris

نمودار شمارشی نیز توسط تابع `counter()` و `filled.counter()` در بسته `graphic` و تابع `contourplot()` در بسته `lattice` قابل دسترس می‌باشد.

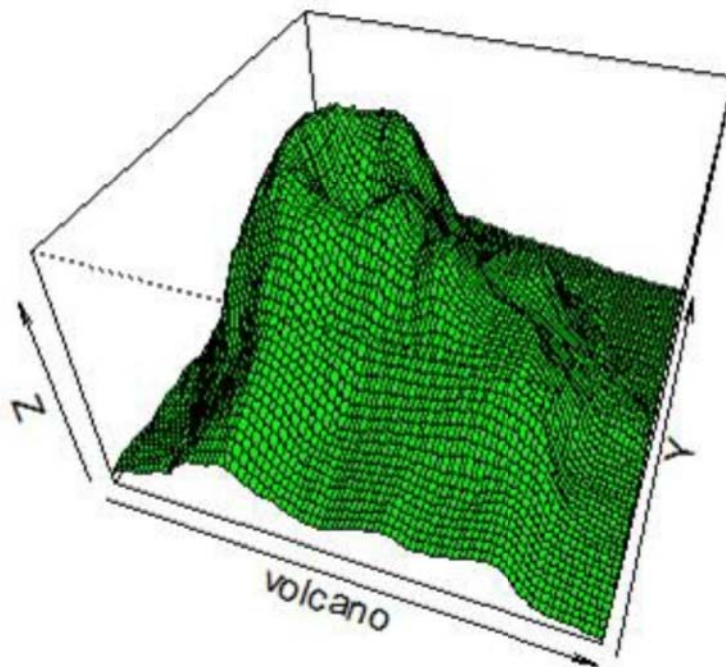
```
> filled.contour(volcano, asp=1, color=terrain.colors,
+ plot.axes=contour(volcano, add=T))
```



شکل ۳-۱۴: نمودار شمارشی داده‌های Iris

یک راه دیگر برای نمایش ماتریس عددی در یک سطح سه‌بعدی با استفاده از تابع `persp()` امکان‌پذیر می‌باشد.

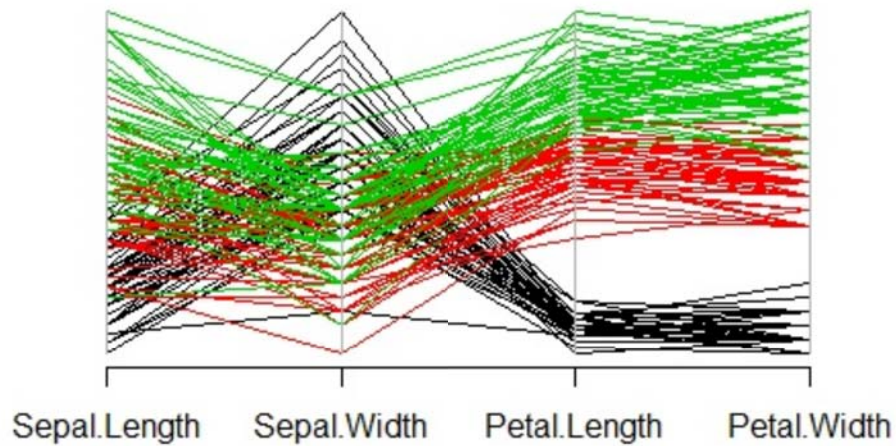
```
> persp(volcano, theta=20, phi=40, expand=0.6,  
col="green")
```



شکل ۳-۱۵ : نمودار سه‌بعدی داده‌های Iris با استفاده از تابع `persp()`

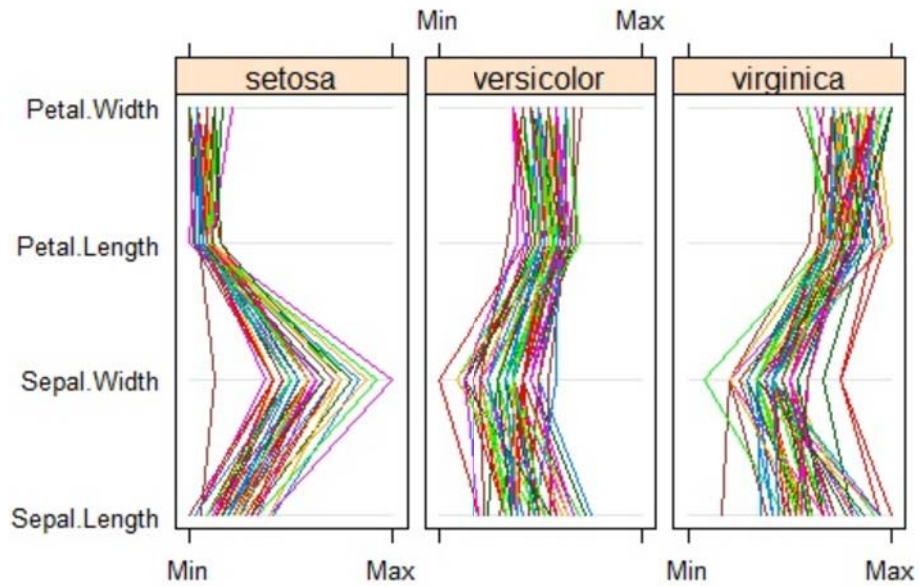
مختصات موازی یک مصورسازی زیبا را با استفاده از داده‌های چندبعدی میسر می‌نماید. یک نمودار مختصات موازی با استفاده از تابع `parcoord()` در بسته MASS و تابع `parallelplot()` در بسته `lattice` در دسترس می‌باشد.

```
> library(MASS)
> parcoord(iris[1:4], col=iris$Species)
```



شکل ۳-۱۶: نمودار مختصات موازی داده‌های Iris با استفاده از بسته Mass

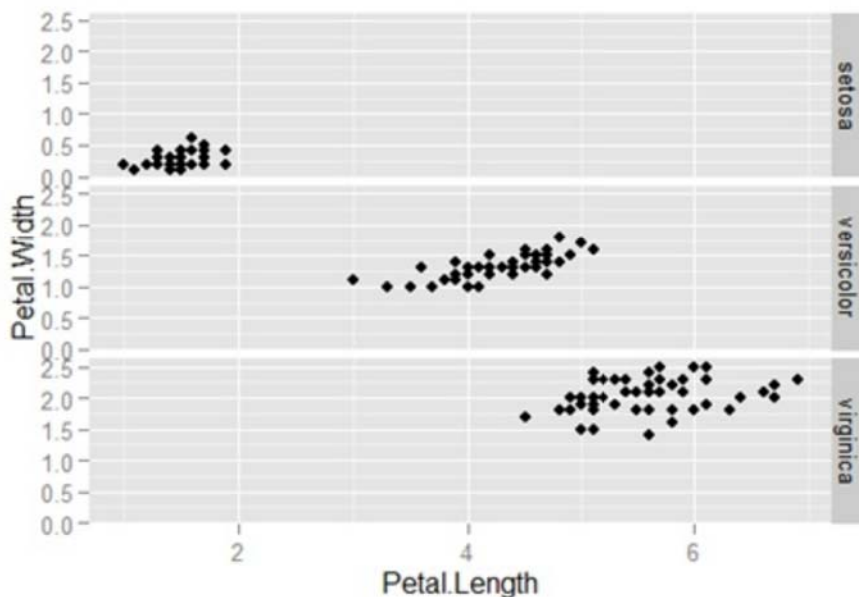
```
> library(lattice)
> parallelplot(~iris[1:4] | Species, data=iris)
```



شکل ۳-۱۷: نمودار سه‌بعدی داده‌های Iris با استفاده از بسته lattice

بسته ggplot۲ دارای یک گرافیک پیشرفته است که مناسب برای اکتشاف داده می باشد. یک مثال را در زیر مشاهده می فرمایید.

```
> library(ggplot2)
> qplot(Petal.Length, Petal.Width, data=iris, facets=Species ~.)
```



شکل ۳-۱۸ : ترسیم نمودار داده های Iris با استفاده از بسته ggplot۲

ذخیره نمودارها در فایل های مختلف

برای ذخیره نمودارها می توان از تابع pdf() و postscript() استفاده نمود. bmp(), jpeg(), png(), tiff() انواع فرمت های تصویری هستند که می توانند مورد استفاده قرار بگیرند. هر فایلی پس از ترسیم توسط توابع graphics.off() یا dev.off() نیاز به بسته شدن دارد.

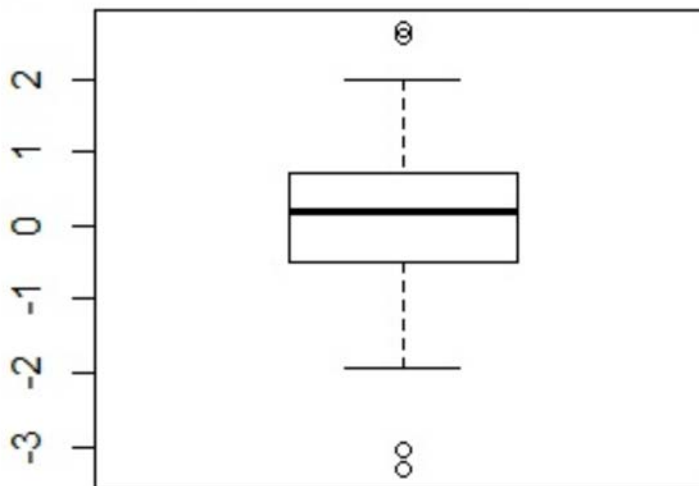
```
> pdf("d:/r/filesave.pdf")
> x <- 1:100
> plot(x, log(x))
> graphics.off()

> postscript("file2save.ps")
> x <- -40:40
> plot(x, -2*x^3)
> graphics.off()
```

تشخیص داده پرت تک متغیره

در این بخش مثالی از تشخیص داده پرت تک متغیره را بررسی خواهیم نمود و نشان خواهیم داد که چگونه می‌توان در داده‌های چندمتغیره نیز همین مدل را اعمال نمود. در این مثال با استفاده از تابع `boxplot.stats()` داده پرت تک متغیره را تشخیص خواهیم داد و با استفاده از مفاهیم آماری به‌صورت مصور نمایش خواهیم داد. در نتیجه این تابع لیستی از داده‌های پرت را نشان خواهد داد. برای اطلاعات بیشتر راجع به کاربردهای تابع مذکور می‌توان از کد `boxplot.stats?` استفاده نمود. شکل زیر نمودار جعبه‌ای را نمایش می‌دهد که در آن چند نقطه جزو نقاط پرت به حساب می‌آیند.


```
> set.seed(3147)
> Var <- rnorm(100)
> summary(Var)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-3.3150 -0.4837  0.1867  0.1098  0.7120  2.6860
> boxplot.stats(Var)$out
[1] -3.315391  2.685922 -3.055717  2.571203
> boxplot(Var)
```



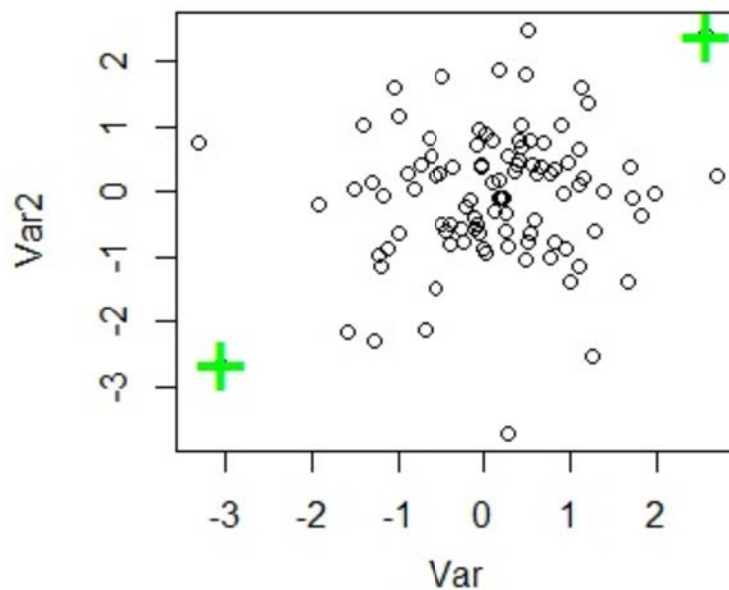
شکل ۳-۱۹: نمودار جعبه‌ای تشخیص داده پرت در مجموعه‌های داده‌ای تک متغیره

تشخیص نقاط پرت تک متغیره می‌تواند در داده‌های چندمتغیره نیز مورد استفاده قرار بگیرد. در مثال زیر ما ابتدا یک قالب داده‌ای با نام `df` ایجاد نموده که دارای دو ستون `X` و `Y` می‌باشد. سپس نقاط پرت مربوط به این مجموعه داده دوستونی را پیدا می‌نماییم. بعد از آن داده‌های پرت به صورت مجزا در هر ستون تشخیص داده می‌شوند. در شکل زیر نقاط پرت با علامت بعلاوه نشان داده شده‌اند.

```

> Var2 <- rnorm(100)
> datafr <- data.frame(Var, Var2)
> rm(Var, Var2)
> head(datafr)
      Var      Var2
1 -3.31539150  0.7619774
2 -0.04765067 -0.6404403
3  0.69720806  0.7645655
4  0.35979073  0.3131930
5  0.18644193  0.1709528
6  0.27493834 -0.8441813
> attach(datafr)
> (m1 <- which(Var %in% boxplot.stats(Var)$out))
[1]  1 33 64 74
> (m2 <- which(Var2 %in% boxplot.stats(Var2)$out))
[1] 24 25 49 64 74
> detach(datafr)
> (outlier.list1 <- intersect(m1,m2))
[1] 64 74
> plot(datafr)
> points(datafr[outlier.list1,], col="green",
        pch="+", cex=2.8)

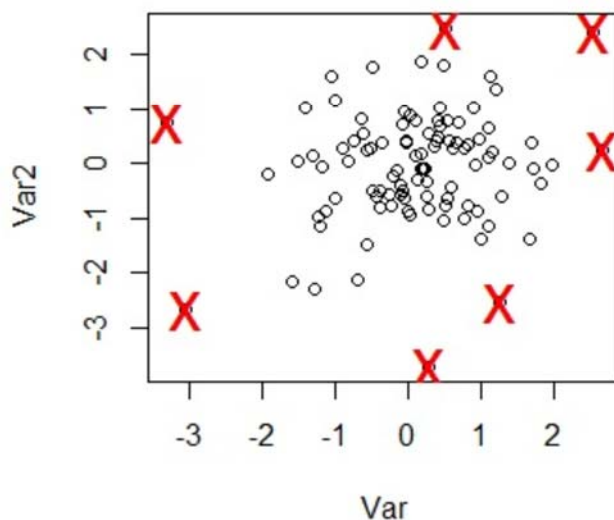
```



شکل ۳-۲۰: تشخیص نقاط داده پرت در مجموعه‌های داده‌ای دومتغیره

به طور مشابه ما می‌توانیم نقاط پرت را در داده‌هایی که در هر دو ستون X و Y دارای خطا هستند پیدا نماییم. نقاط پرت با استفاده از علامت ضربدر در شکل نشان داده شده است.

```
> (outlier.list2 <- union(m1,m2))
[1] 1 33 64 74 24 25 49
> plot(datafr)
> points(datafr[outlier.list2,], col="red", pch="x", cex=2.5)
```



شکل ۳-۲۱: تشخیص نقاط داده پرت در مجموعه‌های داده‌ای دومتغیره دارای خطا در هر دو متغیر

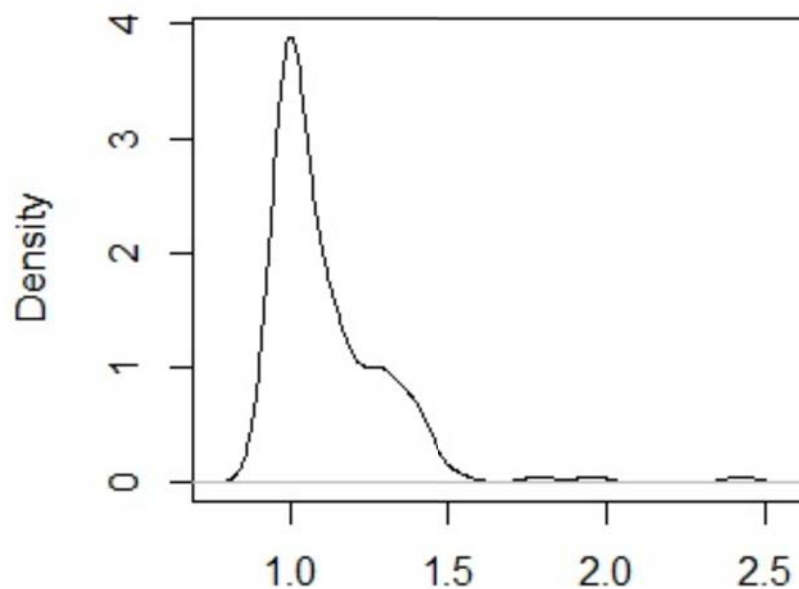
تشخیص داده پرت با استفاده از LOF

روش LOF یک الگوریتم برای شناسایی نقاط پرت بر اساس روش‌های مبتنی بر چگالی می‌باشد. با استفاده از این روش چگالی محلی در یک نقطه با همسایه‌هایش مقایسه می‌گردد. اگر چگالی در یک نقطه به صورت مشخص کمتر از بعدی‌ها باشد این نقطه در مناطق خالی مستقر است که به نظر می‌رسد یک نقطه پرت باشد. تابع `lofactor()` برای محاسبه الگوریتم ذکر شده استفاده می‌گردد که در بسته `DMwR` و `dprep` قابل دسترسی می‌باشد. یک مثال از تشخیص نقاط پرت با

استفاده از روش LOF در ادامه آمده است که k عددی است از همسایه‌ها که برای محاسبه نقاط پرت محلی مورد استفاده قرار می‌گیرد.

```
> library(DMwR)
> irisdata2 <- iris[,1:4]
> outlier.scores <- lofactor(irisdata2, k=6)
> plot(density(outlier.scores))
```

density.default(x = outlier.scores)



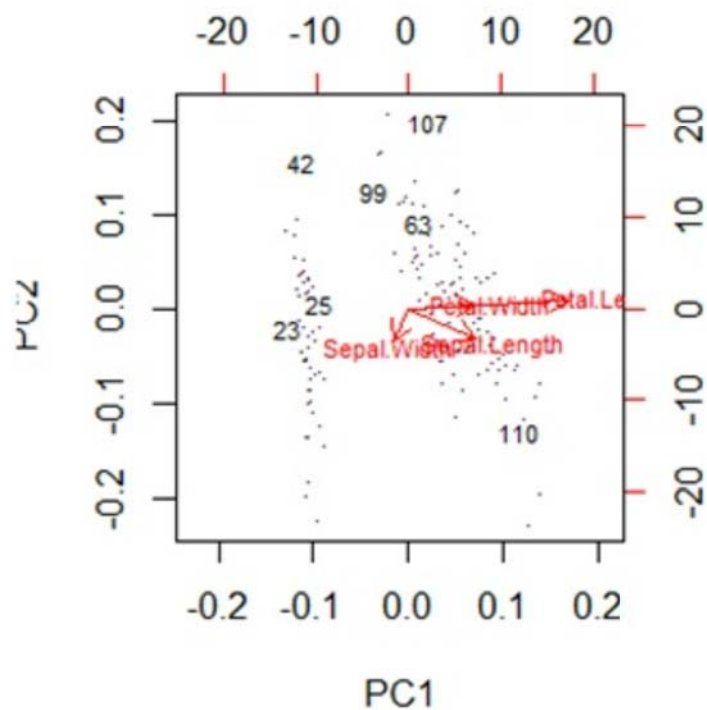
N = 150 Bandwidth = 0.04897

شکل ۳-۲۲: تشخیص داده‌های پرت با استفاده از LOF

```
> outliersdata <- order(outlier.scores, decreasing=T)[1:7]
> print(outliersdata)
[1] 42 107 23 110 99 63 25
> print(irisdata2[outliersdata,])
      Sepal.Length Sepal.Width Petal.Length Petal.Width
42           4.5         2.3         1.3         0.3
107          4.9         2.5         4.5         1.7
23           4.6         3.6         1.0         0.2
110          7.2         3.6         6.1         2.5
99           5.1         2.5         3.0         1.1
63           6.0         2.2         4.0         1.0
25           4.8         3.4         1.9         0.2
```

در ادامه نقاط پرت را با استفاده از تابع biplot نمایش می دهیم.

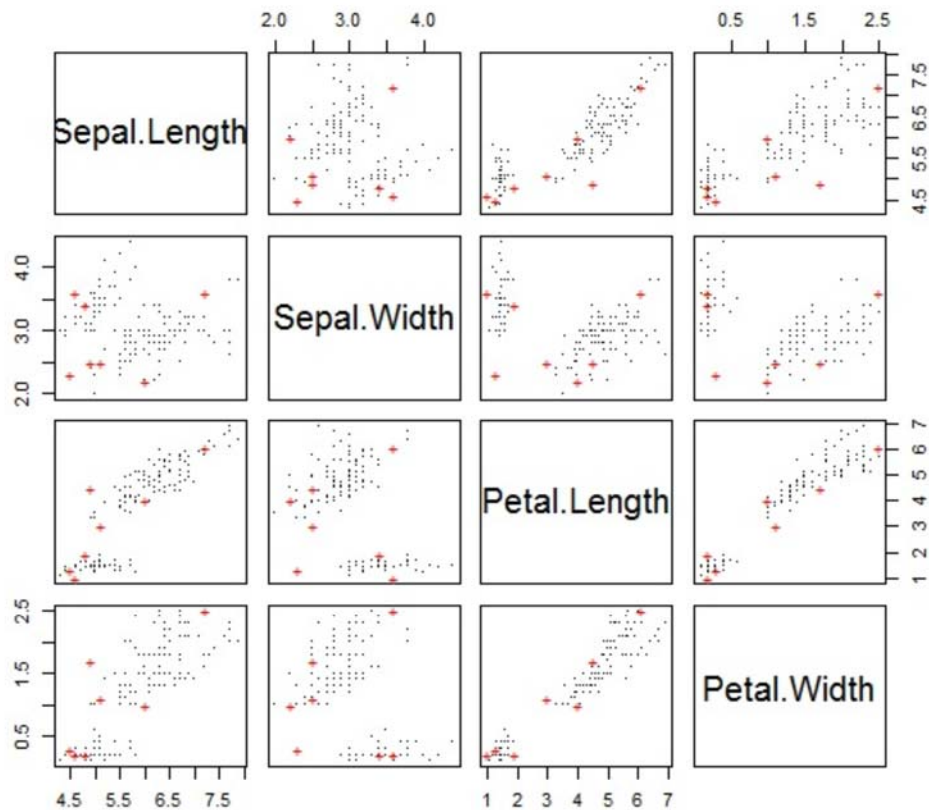
```
> number <- nrow(irisdata2)
> labels <- 1:number
> labels[-outliersdata] <- "."
> biplot(prcomp(irisdata2), cex=.7,
  xlabs=labels)
```



شکل ۳-۲۳: تشخیص داده های پرت با استفاده از تابع biplot

در کد بالا تابع `prcomp()` یک تحلیل برای اجزای اصلی موردنظر انجام می‌دهد و تابع `biplot()` داده‌ها را با استفاده از دو جزء اصلی نمایش می‌دهد. در شکل زیر محورهای X و Y هر کدام به ترتیب برای هر کدام از ستون‌های داده‌ای موردبررسی قرار می‌گیرد. برای هر کدام از تقاطع ستون‌های داده‌ای چند نقطه از نقاط پرت به صورت علامت بعلاوه نمایش داده شده است.

```
> pchdata <- rep(".", number)
> pchdata[outliersdata] <- "+"
> col <- rep("black", number)
> col[outliersdata] <- "red"
> pairs(irisdata2, pch=pchdata, col=col)
```



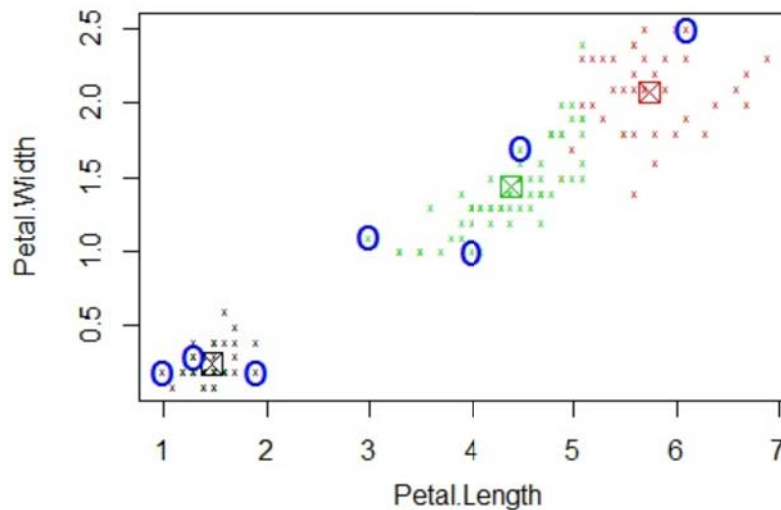
شکل ۳-۲۴: تشخیص داده‌های پرت در همه متغیرهای موردبررسی

تشخیص داده پرت با استفاده از خوشه‌بندی

راه دیگری برای تشخیص نقاط پرت استفاده از متدهای خوشه‌بندی می‌باشد. به‌وسیله خوشه‌بندی داده‌ها، آن داده‌هایی که در هیچ خوشه‌ای جای نگرفته‌اند به‌عنوان نقاط پرت در نظر گرفته می‌شوند. برای مثال در خوشه‌بندی مبتنی بر چگالی که با عنوان DBSCAN شناخته می‌شود داده‌هایی که در یک گروه خوشه‌بندی می‌شوند در یک منطقه پرجمعیت به‌صورت فشرده به هم متصل‌اند. بنابراین داده‌هایی که به هیچ خوشه‌ای تعلق ندارند و به‌صورت ایزوله قرار می‌گیرند جزو نقاط پرت به حساب می‌آیند. همچنین از روش خوشه‌بندی k-mens نیز می‌توان نقاط پرت را شناسایی نمود. با استفاده از این روش داده‌ها در k خوشه با استفاده از تخصیص آن‌ها به مراکز خوشه‌ها تقسیم‌بندی می‌شوند. بعد از این مرحله ما می‌توانیم فاصله بین هر داده با مرکز خوشه را محاسبه نماییم و داده‌های دارای بیشترین فاصله را به‌عنوان نقاط پرت در نظر بگیریم. در ادامه مثالی از روش تشخیص نقاط پرت با استفاده از روش k-means را مشاهده می‌فرمایید.

```
> datacenters <- kmeans.result$datacenters[kmeans.result$cluster, ]
> datadistances <- sqrt(rowSums((irisdata2 - centers)^2))
> outliers <- order(datadistances, decreasing=T)[1:7]
> print(outliersdata)
[1] 42 107 23 110 99 63 25
> print(irisdata2[outliersdata,])
      Sepal.Length Sepal.Width Petal.Length Petal.Width
42           4.5         2.3         1.3         0.3
107          4.9         2.5         4.5         1.7
23           4.6         3.6         1.0         0.2
110          7.2         3.6         6.1         2.5
99           5.1         2.5         3.0         1.1
63           6.0         2.2         4.0         1.0
25           4.8         3.4         1.9         0.2
```

```
> plot(irisdata2[,c("Petal.Length", "Petal.Width")], pch="x",
+ col=kmeans.result$cluster, cex=0.4)
> points(kmeans.result$centers[,c("Petal.Length", "Petal.Width")],
+ col=1:3, pch=7, cex=1.7)
> points(irisdata2[outliersdata, c("Petal.Length", "Petal.Width")],
+ pch="o", col=4, cex=1.7)
```

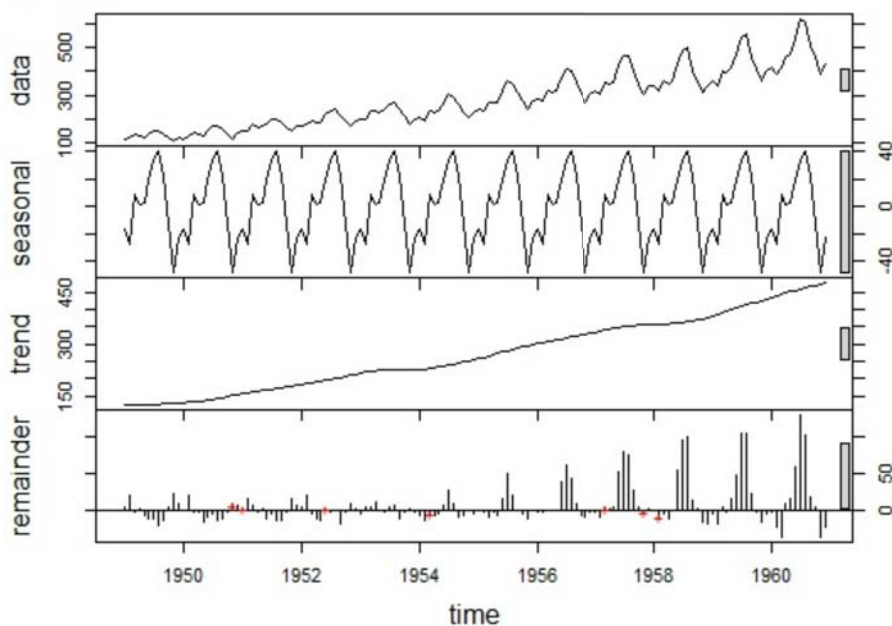


شکل ۳-۲۵: تشخیص نقاط پرت با استفاده از روش k-means

تشخیص داده پرت با استفاده از سری زمانی

در این بخش مثالی از تشخیص نقاط پرت با استفاده از داده‌های سری زمانی را بررسی می‌نماییم. در این مثال داده‌های سری زمانی را با استفاده از رگرسیون و با استفاده از تابع `stl()` از یکدیگر مجزا نموده و سپس نقاط پرت را شناسایی می‌نماییم.

```
> fitt <- stl(AirPassengers, "periodic", robust=TRUE)
> (outliers <- which(fitt$weights<1e-8))
[1] 79 91 92 102 103 104 114 115 116 126 127 128 138 139 140
> op <- par(mar=c(0, 4, 0, 3), oma=c(5, 0, 4, 0), mfcol=c(4, 1))
> plot(fitt, set.pars=NULL)
> s <- fitt$time.series
> points(time(s)[outliersdata], 0.7*s[, "remainder"][outliersdata],
+ pch="+", col="red")
> par(op)
```



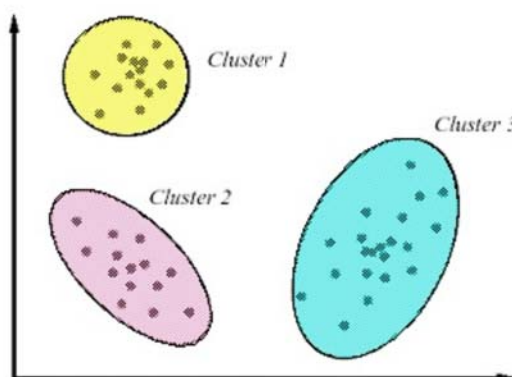
شکل ۳-۲۶: تشخیص نقاط پرت با استفاده از سری زمانی

فصل چهارم

خوشه بندی

۴-۱- خوشه‌بندی

خوشه‌بندی از ده‌ها سال پیش برای تشخیص تعداد محدودی گروه یا خوشه استفاده شده است (Kaufman and Rousseeuw, ۱۹۹۰). خوشه‌بندی یک روش متداول توصیفی است که در آن گروه‌ها، ممکن است افزاز شده یا همپوشان باشند. یک مثال ساده از خوشه‌بندی می‌تواند عبارت از کشف زیرگروه‌های همگن مصرف‌کنندگان در یک پایگاه داده مربوط به داده‌های بازاریابی باشد. داده‌ها بر اساس اصل حداکثر کردن شباهت داخل گروه‌ها و حداقل کردن شباهت بین گروه‌ها، خوشه‌بندی یا گروه‌بندی می‌شوند.



شکل ۴-۱: مثالی از خوشه‌بندی داده‌ها

۴-۲- تفاوت دسته‌بندی و خوشه‌بندی

خوشه‌بندی و دسته‌بندی جزو متدهای اساسی داده‌کاوی می‌باشند. دسته‌بندی بیشتر یک روش یادگیری نظارت‌شده است در صورتی که خوشه‌بندی یک روش یادگیری غیرنظارت‌شده می‌باشد. (البته برخی مدل‌ها شامل هر دو می‌شوند) هدف نهایی خوشه‌بندی توصیفی است درحالی که دسته‌بندی جزو روش‌های پیش‌بینی‌کننده می‌باشد. هدف از خوشه‌بندی کشف دسته‌های جدید، ایجاد گروه‌های جدید می‌باشد و نوع ارزیابی در خوشه‌بندی درونی می‌باشد. اما دسته‌بندی اگرچه یک بخش بسیار مهم از ارزیابی بیرونی می‌باشد، گروه‌ها باید

منعکس کننده مجموعه‌های ارجاعی از دسته‌ها باشند. درواقع فهم جهان ما نیازمند مفهوم‌سازی مشابهت‌ها و تمایزات بین مؤلفه‌های تشکیل دهنده آن می‌باشد.

۴-۳- فرآیند خوشه‌بندی

فرآیند گروه‌بندی مجموعه‌ای از اشیای فیزیکی یا مجرد در کلاسهای دارای اشیای مشابه را خوشه‌بندی می‌نامند. یک خوشه مجموعه‌ای از اشیای داده‌ای است که مشابه با دیگر داده‌های خوشه خود و غیرمشابه با داده‌های موجود در خوشه‌های دیگر می‌باشند. خوشه‌ای از اشیای داده‌ای می‌توانند در مجموع به عنوان یک گروه در نظر گرفته شوند و یا ممکن است به عنوان یک شکل فشرده از داده‌ها در نظر گرفته شوند. اگرچه دسته‌بندی یکی از روش‌های مؤثر برای تمایز گروه‌ها یا کلاسهای اشیا می‌باشد اما این روش نیازمند جمع‌آوری و برچسب‌گذاری پرهزینه‌ای برای الگوها و مجموعه‌های چندتایی با مقدار زیاد می‌باشد که کسی که نقش دسته‌بندی را دارد باید هر گروه را مدل‌سازی نماید. این روش برای دسترسی در جهت معکوس معمولاً مطلوب است. اولین بخش قرار دادن مجموعه داده‌ها بر اساس شباهت داده‌ها و سپس تخصیص برچسب به تعداد کوچکی از گروه‌ها که دارای مشابهت هستند می‌باشد. مزایای دیگر فرآیند خوشه‌بندی این است که با تغییرات سازگار است و زمانی که یک خصوصیت مجرد مفید و بدون ارتباط با گروه‌های دیگر داریم این متد بهتر ما را یاری می‌رساند. خوشه‌بندی اشیا یک نیاز قدیمی انسانی برای توصیف ویژگی‌های بارز انسان‌ها و اشیا و تشخیص آن‌ها با یک نوع خاص می‌باشد. بنابراین این متد با حیطه‌های علمی گسترده‌ای هم اشتراکی دارد. از ریاضیات و آمار تا بیولوژی و ژنتیک هرکدام از اصطلاحات متفاوتی برای توصیف ساختارهایی که با این تحلیل‌ها شکل می‌گیرند استفاده می‌نمایند. از اصطلاح بیولوژیکی رده‌بندی تا اصطلاح پزشکی سندرم و اصطلاح ژنتیکی نوع جنس برای تولید مفهوم گروه استفاده می‌شوند. اما مشکل یکسان است: تشکیل دسته‌ای از مشخصه‌ها و تخصیص اشیا به گروه‌های متناسب با آن‌ها.

۴-۴- کاربردهای خوشه‌بندی

از آنجاکه خوشه‌بندی یک روش یادگیری بدون نظارت محسوب می‌گردد، در موارد بسیاری می‌تواند کاربرد داشته باشد.

- بازاریابی^۱: دسته‌بندی مشتری‌ها به دسته‌هایی برحسب رفتارها و نیازهای آن‌ها از طریق مجموعه زیادی از ویژگی‌ها و آخرین خریدهای آن‌ها.
- زیست‌شناسی^۲: دسته‌بندی حیوانات و گیاهان از روی ویژگی‌های آن‌ها
- کتابداری: دسته‌بندی کتاب‌ها
- نقشه‌برداری شهری^۳: دسته‌بندی خانه‌ها بر اساس نوع و موقعیت جغرافیایی آن‌ها.
- مطالعات زلزله‌نگاری^۴: تشخیص مناطق حادثه‌خیز بر اساس مشاهدات قبلی.
- وب: دسته‌بندی اسناد و یا دسته‌بندی مشتریان به سایت‌ها و
- داده‌کاوی: کشف اطلاعات و ساختار جدید از داده‌های موجود
- در تشخیص گفتار^۵: در تقسیم کردن گفتار برحسب گویندگان آن و یا فشرده‌سازی گفتار
- در تقسیم‌بندی تصاویر^۶: تقسیم‌بندی تصاویر پزشکی و یا ماهواره‌ای (Wiley & Sons, ۲۰۰۰)

متأسفانه چندین مسئله در خصوص روش‌های خوشه‌بندی مطرح است که هنوز به شکل کامل پا سخ داده نشده‌اند. و همچنان تلاش‌های بسیاری به‌منظور حل آن‌ها انجام می‌گیرد.

^۱ Marketing

^۲ Biology

^۳ City Planning

^۴ Earthquake studies

^۵ Speech Recognition

^۶ Image Segmentation

- روش‌های خوشه‌بندی قادر نیستند تمامی نیازهای مسائل را به‌طور هم‌زمان برآورده کنند.
- به دلیل پیچیدگی محاسباتی زیاد در برخورد با مجموعه داده‌های بزرگ با تعداد داده زیاد و تعداد ویژگی‌های زیاد برای هر داده عملی نیستند.
- به دلیل وابستگی شدید به تعریف معیار شباهت بین داده‌ها در مسائلی که تعریف معیار شباهت مشکل باشد نتایج مطلوبی تولید نمی‌کنند. (در داده‌ها با تعداد ویژگی زیاد)
- برای نتایج آن‌ها می‌توان تفسیرهای مختلفی بیان کرد. (Wiley & Sons, ۲۰۰۰)

۴-۵- اعتبارسنجی خوشه‌ها

نتایج حاصل از اعمال الگوریتم‌های خوشه‌بندی روی یک مجموعه داده با توجه به انتخاب‌های پارامترهای الگوریتم‌ها می‌تواند بسیار متفاوت از یکدیگر باشد. هدف از اعتبارسنجی خوشه‌ها یافتن خوشه‌هایی است که بهترین تناسب را با داده‌های موردنظر داشته باشند. دو معیار پایه اندازه‌گیری پیشنهادشده برای ارزیابی و انتخاب خوشه‌های بهینه عبارت‌اند از:

- تراکم^۱: داده‌های متعلق به یک خوشه بایستی تا حد ممکن به یکدیگر نزدیک باشند. معیار رایج برای تعیین میزان تراکم داده‌ها واریانس داده‌ها است.
- جدایی^۲: خوشه‌ها خود بایستی به اندازه کافی از یکدیگر جدا باشند. سه راه برای سنجش میزان جدایی خوشه‌ها مورد استفاده قرار می‌گیرد که عبارت‌اند از:

○ فاصله بین نزدیک‌ترین داده‌ها از دو خوشه

^۱ Compactness

^۲ Separation

- فاصله بین دورترین داده‌ها از دو خوشه
- فاصله بین مراکز خوشه‌ها

همچنین روش‌های ارزیابی خوشه‌های حاصل از خوشه‌بندی را به صورت سه دسته تقسیم می‌کنند که عبارت‌اند از:

- ☐ معیارهای بیرونی^۱
- ☐ معیارهای درونی^۲
- ☐ معیارهای نسبی^۳

هم معیارهای خروجی و هم معیارهای درونی بر مبنای روش‌های آماری عمل می‌کنند و پیچیدگی محاسباتی بالایی را نیز دارا هستند. معیارهای خروجی عمل ارزیابی خوشه‌ها را با استفاده از بینش خاص کاربران انجام می‌دهند. معیارهای درونی عمل ارزیابی خوشه‌ها را با استفاده از مقداری که از خوشه‌ها و نمای آن‌ها محاسبه می‌شود، انجام می‌دهند ولی در معیارهای بیرونی خوشه‌های به‌دست‌آمده با دسته‌های از پیش موجود مقایسه می‌شوند.

پایه معیارهای نسبی، مقایسه بین شماهای خوشه‌بندی (الگوریتم به علاوه پارامترهای آن) مختلف است. یک و یا چندین روش مختلف خوشه‌بندی چندین بار با پارامترهای مختلف روی یک مجموعه داده اجرا می‌شوند و بهترین شمای خوشه‌بندی از بین تمام شماها انتخاب می‌شود. در این روش مبنای مقایسه، شاخص‌های اعتبارسنجی^۴ هستند. شاخص‌های ارزیابی بسیار متنوعی پیشنهاد شده‌اند که در این قسمت سعی می‌شوند تعدادی از رایج‌ترین آن‌ها معرفی شوند. (Kovacs, Legány, Babos, ۲۰۰۳)

^۱ External Criteria

^۲ Internal Criteria

^۳ Relative Criteria

^۴ Validity-Index

۴-۶- شاخص‌های اعتبارسنجی

شاخص‌های اعتبارسنجی برای سنجش میزان خوبی^۱ نتایج خوشه‌بندی به‌منظور مقایسه بین روش‌های خوشه‌بندی مختلف یا مقایسه نتایج حاصل از یک روش با پارامترهای مختلف مورد استفاده قرار می‌گیرد.

۴-۷- روش‌های خوشه‌بندی

در این بخش برخی از مهم‌ترین الگوریتم‌های خوشه‌بندی را شرح خواهیم داد. با توجه به وجود تعداد الگوریتم‌های فراوان برای خوشه‌بندی می‌توان فهمید که معنای خوشه به‌صورت دقیق تعریف نشده است. (Estivill-Castro, ۲۰۰۰)

رویکردهای اصلی خوشه‌بندی عبارت‌اند از :

- روش‌های افرازی^۲
- روش‌های سلسله مراتبی^۳
- روش‌های مبتنی بر چگالی^۴
- روش‌های مبتنی بر مشبک کردن فضا^۵
- نقشه‌های خودسازمان‌ده^۶

در ادامه تعاریف مربوط به هر کدام از این رویکردهای اصلی خوشه‌بندی را بررسی می‌نماییم و در برخی موارد مراحل الگوریتمی این رویکردها را بیان می‌نماییم. (Han and Kamber, ۲۰۰۶)

^۱ Goodness

^۲ Partitioning

^۳ Hierarchical Clustering

^۴ Density Based methods

^۵ Grid Based Method

^۶ Self-Organizing Feature Maps

۴-۷-۱- روش‌های افرازی

- **K-میانگین:** ایده اصلی معرفی هسته‌ها یا مراکزی است که مشاهدات جذب آن‌ها می‌شوند تا یک خوشه را شکل دهند. لازم است تعداد خوشه‌ها را از قبل مشخص کنیم. زمان محاسبات از درجه $O(n)$ است. n تعداد مشاهدات است. در خوشه‌بندی K-میانگین تعداد G مشاهده اول به عنوان هسته انتخاب می‌شوند. سپس یک فرایند تکراری اجرا می‌گردد که در هر گام آن خوشه‌های موقت شکل گرفته و هر مشاهده به خوشه دارای نزدیک‌ترین هسته تخصیص می‌یابد. هر بار که مشاهده‌ای به یک خوشه تخصیص می‌یابد، هسته با میانگین آن خوشه جایگزین می‌شود. فرایند تا همگرایی یعنی تا زمانی که تغییر عمده‌ای در هسته‌ها رخ ندهد تکرار می‌شود. در انتهای فرایند در مجموع G خوشه با مراکز متناظر خواهیم داشت.
- **K-میانه:** همان K-میانگین ولی به جای میانگین خوشه‌ها، یکی از مشاهدات خوشه‌ها به عنوان مرکز خوشه انتخاب می‌شود. حساسیت کمتری نسبت به نقاط پرت دارد. زمان محاسبات از درجه $O(n^2)$ می‌باشد.
- خوشه‌بندی به روش نقشه‌های خود سازمان: نقشه‌های خود سازمان^۱ روشی برای خوشه‌بندی به کمک شبکه‌های عصبی هستند (Kohonen, ۱۹۸۲). مزیت اصلی این روش، ایجاد شبکه‌ای برای ذخیره اطلاعات می‌باشد به نحوی که ارتباط مکانی (توپولوژیک) بین مجموعه آموزشی حفظ می‌شود. با گذشت بیش از ۲۰ سال هنوز کاربردهای جدیدی برای این شبکه پیدا شده و فنون مربوط به آن توسعه می‌یابند (Oja, ۲۰۰۲) فرایندهای اصلی این روش عبارتند از:
- رقابت: برای تعیین نورون برنده

^۱ Self-organizing maps (SOM)

- همکاری: کمک به همسایگان مجاور
 - تطبیق: اصلاح وزن‌ها برای نزدیکی بیشتر به بردار ورودی
- فرض کنید یک پایگاه داده با n شیء داریم. یک روش افرازی، k افراز از این داده‌های اشیا درست می‌کند به‌طوری‌که هر افراز یک خوشه را نشان می‌دهد و $k < n$. پس داده‌های اشیا در k گره خوشه‌بندی شده و دارای دو شرط می‌باشند:
- هر گروه حداقل یک شیء دارد.
 - هر شیء تنها به یک گروه تعلق دارد. (این شرط در روش‌های افرازی فازی می‌تواند قابل انعطاف باشد).
- در روش افرازی برای k معلوم، یک افراز ابتدایی ایجاد می‌شود. سپس یک روش جابجایی تکراری را به کار برده که تلاش به بهبود افرازبندی دارد. به این صورت که اشیا را از یک گروه به دیگر گروه‌ها می‌برد. یک معیار عمومی برای یک افرازبندی خوب این است که اشیا در یک خوشه به هم نزدیک یا به یکدیگر وابسته باشند و در مقابل اشیا در خوشه‌های مختلف، از یکدیگر دور یا تا حد امکان متفاوت باشند.
- برای دستیابی به خوشه‌بندی بهینه در روش افرازی، به دو شمارش کامل همه افرازهای ممکن نیاز خواهد بود یعنی تمام حالات ممکن باید بررسی شوند که این روش برای پایگاه داده‌های بزرگ ناممکن است. لذا الگوریتم‌های هیوریستیک زیر برای بررسی این گونه موارد استفاده می‌شوند.
- الگوریتم k -means که هر خوشه با میانگین اشیا آن خوشه یا مرکز خوشه، نمایش داده می‌شود
 - الگوریتم k -medoids که هر خوشه با یکی از اشیا که در نزدیکی مرکز خوشه جای گرفته است، نمایش داده می‌شود.

این روش‌ها برای یافتن خوشه‌هایی به شکل کره در پایگاه داده‌های کوچک تا متوسط به خوبی کار می‌کنند، اما برای یافتن خوشه‌هایی با اشکال پیچیده و یا دارای مجموعه داده‌های بزرگ باید توسعه داده شوند.

۴-۷-۲- روش‌های سلسله مراتبی

این روش ساختاری سلسله‌مراتبی از اشیا یک مجموعه معلوم ایجاد می‌کند. روش سلسله‌مراتبی می‌تواند خوشه‌بندی را به صورت تجمیعی و یا به صورت تقسیمی انجام دهد. به رویکرد تجمیعی، رویکرد پایین به بالا نیز گفته می‌شود. این روش با شکل‌دهی گروه‌های مجزا که هر یک شامل حداقل یک شیء می‌باشند شروع می‌شود. سپس اشیا یا گروه‌های نزدیک به هم را یکی می‌کند تا اینکه در نهایت یک گروه کلی در بالاترین سطح ایجاد شود. در روش تقسیمی کل اشیا در یک خوشه در نظر گرفته شده و در هر تکرار یک خوشه به دو خوشه کوچک‌تر تقسیم می‌شود. این روش نیاز به دانستن خوشه‌ها از قبل ندارد ولی ممکن است به قدرت محاسباتی بسیار زیادی نیاز داشته باشد. یک نمودار دندان‌ه‌ای برای سطوح مختلف خوشه‌بندی تشکیل می‌شود. روش‌های سلسله‌مراتبی به دو دسته تقسیم می‌شوند.

- تجمیعی^۱.

- تقسیمی^۲.

۴-۷-۳- روش مبتنی بر چگالی

بسیاری از روش‌های افرازی، اشیا را بر اساس فاصله آن‌ها نسبت به یکدیگر خوشه‌بندی می‌کنند. برخی روش‌ها تنها خوشه‌بندی کروی شکل را پیدا می‌کنند و در برابر خوشه‌هایی به شکل‌های دلخواه با مشکل مواجه می‌شوند. در مقابل برخی روش‌های دیگر خوشه‌بندی بر پایه چگالی توسعه یافته‌اند. ایده عمومی این

^۱ Agglomerative

^۲ Divisive

روش‌ها رشد دادن خوشه‌ها بر پایه چگالی در همسایگی با شعاع مشخص آن‌هاست. به این معنی که برای هر نقطه داده در یک خوشه معلوم، همسایه‌ای با شعاع مشخص در نظر گرفته می‌شود. این نوع خوشه‌بندی برای هموارسازی نویزها و کشف خوشه‌هایی با اشکال دلخواه به کار می‌رود. برای کشف خوشه‌های دارای شکل دلخواه، اخیراً روش‌های خوشه‌بندی مبتنی بر چگالی توسعه یافته‌اند. این روش‌ها خوشه‌ها را نواحی پرچگالی از اشیاء در فضای داده‌ها در نظر می‌گیرد که با نواحی کم‌چگالی از هم جدا شده‌اند. نواحی کم‌چگالی نشانگر نویز هستند. مزیت مهم دیگر این روش‌ها عدم نیاز به دانستن تعداد خوشه‌ها از قبل است. سه روش عمده وجود دارد:

- خوشه‌بندی مکانی مبتنی بر چگالی کاربردهای دارای نویز^۱
(Ester et al., ۱۹۹۶)
- مرتب کردن نقاط برای تشخیص ساختار خوشه^۲
(Ankerst et al., ۱۹۹۹)
- خوشه‌بندی بر مبنای بر توابع توزیع چگالی^۳
(Hinneburg and Keim, ۱۹۹۸)

برای استفاده از مزایای خوشه‌بندی‌های سلسله‌مراتبی و افزایی می‌توان آن‌ها را ترکیب کرد. برای این کار ابتدا یک خوشه‌بندی غیرسلسله‌مراتبی مانند K- میانگین روی کل داده‌ها با تعداد زیاد خوشه G انجام می‌شود. این خوشه‌ها ورودی مرحله دوم هستند. در مرحله دوم روش خوشه‌بندی سلسله‌مراتبی روی نمونه‌ای از داده‌ها برای یافتن تعداد بهینه خوشه‌ها اجرا می‌شود. از آنجا که تعداد خوشه‌ها نمی‌تواند بزرگ‌تر از G باشد، فرایند تجمعی است یعنی با G خوشه شروع شده و به سمت کمتر شدن خوشه‌ها پیش می‌رود. در مرحله سوم، همین که تعداد بهینه خوشه انتخاب شد، الگوریتم با یک خوشه‌بندی غیرسلسله‌مراتبی مانند SOM ادامه می‌دهد تا مشاهدات را به G گروه منتخب

^۱ Density-Based Spatial Clustering of Applications with Noise: DBSCAN

^۲ Ordering Points To Identify the Clustering Structure: OPTICS

^۳ DENsity-based CLUstEring: DENCLUE

تخصیص دهد. هسته‌های اولیه این گروه‌ها مراکز به‌دست‌آمده در مرحله دوم می‌باشند.

۴-۸- روش افرازی

فرض کنیم یک پایگاه داده با n شیء داریم. علاوه بر آن تعداد خوشه‌هایی که باید تشکیل شوند نیز معلوم است. یک الگوریتم افرازی، اشیا را در k افراز سازمان‌دهی کرده به طوری که هر افراز یک خوشه را نمایش می‌دهد. خوشه‌ها معمولاً با معیاری که تابع شباهت نیز نام دارد، شکل می‌گیرند. بنابراین اشیا داخل یک خوشه به هم شبیه‌اند و در مقابل اشیا در خوشه‌های مختلف به هم شبیه نیستند. این شباهت و عدم شباهت اشیا بر مبنای پایگاه‌های داده تعیین می‌شود. دو الگوریتم مهم این روش عبارت‌اند از k -means و k -medoids.

۴-۸-۱- الگوریتم k -means

این الگوریتم پارامتر k را به‌عنوان ورودی گرفته و مجموعه n شیء را به k خوشه افراز می‌کند. به طوری که سطح شباهت داخلی خوشه‌ها بالا بوده و سطح شباهت اشیا بیرون خوشه‌ها پایین باشد. شباهت هر خوشه نسبت به متوسط اشیا آن خوشه سنجیده شده که این متوسط، مرکز خوشه نیز نامیده می‌شود. این الگوریتم به صورت زیر کار می‌کند:

ورودی: k تعداد خوشه‌ها و یک پایگاه داده شامل n شیء

خروجی: یک مجموعه از k خوشه که معیار مربع خطا را حداقل می‌کند.

این الگوریتم را در گام‌های زیر مشاهده می‌فرمایید:

قدم ۱) به صورت تصادفی k نقطه دلخواه را به عنوان مراکز خوشه‌های ابتدایی انتخاب کن. (بهتر است k نقطه از n نقطه موجود انتخاب شود)

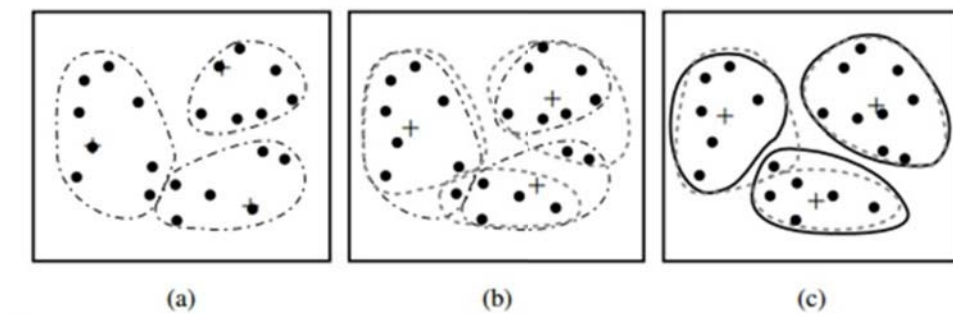
قدم ۲) هر شیء را با توجه به بیشترین شباهت آن به مراکز خوشه‌ها، به خوشه‌ها تخصیص بده.

قدم ۳) مراکز خوشه‌ها را به روز کن به این معنی که برای هر خوشه میانگین اشیا آن خوشه را محاسبه کن.

قدم ۴) با توجه به مراکز جدید خوشه‌ها به قدم دوم برگرد تا هنگامی که هیچ تغییری در خوشه‌ها رخ ندهد. (در این حالت الگوریتم پایان یافته است.)

روش k -means تنها هنگامی کاربرد دارد که بتوان مراکز خوشه‌ها را تعریف نمود. مثلاً برای داده‌هایی با ویژگی‌های طبقه‌ای این روش کارا نیست. از معایب این روش تعیین k است که می‌بایست کاربر ابتدا آن را معین کند و راه خاصی برای تعیین آن مشخص نشده است. یک راه امتحان k های مختلف و بررسی معیار مربع خطا برای هر k می‌باشد. این روش برای کشف خوشه‌هایی با شکل‌های پیچیده مناسب نیست. یکی از مهم‌ترین نقاط ضعف این روش این است که در برابر نویزها و نقاط پرت حساس است زیرا این داده‌ها به راحتی مراکز را تغییر می‌دهند و ممکن است نتایج مطلوبی حاصل نشود.

یکی دیگر از روش‌های مشابه با k -means روش k -modes می‌باشد. در اینجا روش k -means را به منظور استفاده از داده‌های طبقه‌ای توسعه می‌دهد و به جای استفاده از مراکز خوشه‌ها از مدهای خوشه‌ها استفاده می‌کند. لذا از یک رابطه اندازه‌گیری عدم شباهت جدید برای داده‌های اسمی یا طبقه‌ای استفاده می‌کند.



شکل ۴-۲: خوشه‌بندی k-means

۴-۸-۲- الگوریتم k-medoids

در این الگوریتم به جای استفاده از مرکز یک خوشه به عنوان مرجع، می‌توان از medoid ها (اشیایی که در مرکزی‌ترین محل یک خوشه می‌باشند) استفاده کرد. این روش بر اساس اصل حداقل سازی مجموع عدم شباهت‌ها میان هر شیء و شیء مرجع عمل می‌کند. استراتژی اساسی الگوریتم خوشه‌بندی k-medoids پیدا کردن k شیء نماینده آغازین (medoid) به طور دلخواه از n شیء پایگاه داده می‌باشد. هر شیء باقیمانده با medoid ای هم خوشه می‌شد که بیشترین شباهت را به آن داشته باشد. سپس این استراتژی مکرراً یکی از اشیا medoid را با یکی از اشیا غیر medoid جایگزین می‌کند به طوری که کیفیت نتیجه خوشه‌بندی بهبود یابد. این کیفیت با به کارگیری تابع هزینه تخمین زده می‌شود که با میانگین عدم تشابه بین یک شیء و medoid آن خوشه را اندازه‌گیری می‌کند. در اینجا الگوریتم بیان می‌شود.

ورودی : k تعداد خوشه‌ها و پایگاه داده‌ها شامل n شیء

خروجی : یک مجموعه از خوشه‌ها که مجموع عدم تشابه بین تمام اشیا و نزدیک‌ترین medoid آن‌ها را حداقل می‌کند.

این الگوریتم را در قدم‌های زیر مشاهده می‌نمایید:

قدم (۱) k شیء تصادفی به‌عنوان medoid های اولیه اختیار کن.

قدم (۲) هر کدام از اشیا باقیمانده را به خوشه‌ای با نزدیک‌ترین medoid تخصیص بده.

قدم (۳) به طور تصادفی یک شیء غیر medoid را انتخاب کن، $O(\text{random})$

قدم (۴) هزینه نهایی S را از عوض کردن O_j (medoid آن خوشه) و O_{random} محاسبه کن. اگر $S < 0$ آنگاه جای O_j و O_{random} را عوض کن تا مجموعه k تا medoid جدید شکل بگیرد، در غیر این صورت مراکز را عوض نکرده و به قدم ۳ برو

قدم (۵) این الگوریتم را تا زمانی که همه نقاط به‌عنوان medoid انتخاب شده و تغییری در خوشه‌ها ایجاد نشود ادامه بده.

۴-۹- روش خوشه‌بندی سلسله مراتبی

این روش با گروه‌بندی اشیا به صورت یک درخت کار می‌کند و معمولاً به دو صورت پایین به بالا (تجمیعی) یا بالا به پایین (تقسیمی) انجام می‌شود. این دو روش را می‌توان به صورت‌های زیر بیان کرد:

تجمیعی: در این روش خوشه‌ها مکرراً باهم ترکیب می‌شوند. به این صورت که ابتدا هر یک از اشیا را به‌عنوان یک خوشه در نظر می‌گیرد و سپس با ترکیب کردن این خوشه‌ها، آن‌ها را به خوشه‌های بزرگ‌تر تبدیل می‌کند تا اینکه همه اشیا در یک خوشه قرار گیرند و یا به شرط پایان برسد.

تجزیه‌ای یا تقسیمی: در این روش خوشه‌ها مکرراً تقسیم می‌شوند. این روش دقیقاً برعکس روش تجمیعی عمل می‌کند به این صورت که ابتدا یک خوشه شامل همه اشیا ایجاد می‌شود و سپس الگوریتم این خوشه‌ها را به خوشه‌های کوچک و کوچک‌تر تجزیه می‌کند تا اینکه هر شیء در یک خوشه قرار گیرد. این روش

معمولاً مناسب نیست و خیلی کم مورد استفاده قرار می‌گیرد زیرا پیچیدگی محاسباتش بالاست. توجه کنید که هر خوشه را به چندین حالت متفاوت می‌توان به خوشه‌های کوچک‌تر تقسیم کرد که باید بهترین حالت آن انتخاب شود.

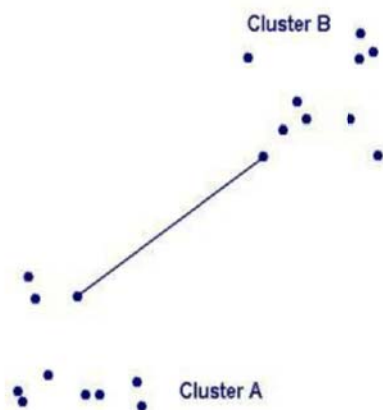
معیارهای گوناگونی که در روش‌های سلسله مراتبی برای فاصله بین خوشه‌ها بکار می‌روند عبارت‌اند از:

پیوند تکی: فاصله بین خوشه‌ها برحسب حداقل فاصله ممکنه بین عناصر آن‌ها محاسبه می‌شود. این روش یکی از قدیمی‌ترین و ساده‌ترین روش‌های خوشه‌بندی است و جزء روش‌های خوشه‌بندی سلسله مراتبی و انحصاری محسوب می‌شود. به این روش خوشه‌بندی، تکنیک نزدیک‌ترین هم‌سایه^۱ نیز گفته می‌شود. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \min_{i \in A, j \in B} d_{ij}$$

که i یک نمونه داده متعلق به خوشه A و j یک نمونه داده متعلق به خوشه B می‌باشد. در واقع در این روش شباهت بین دو خوشه، کمترین فاصله بین یک عضو از یکی با یک عضو از دیگری است. در شکل زیر این مفهوم بهتر نشان داده شده است. (Wiley & Sons, ۲۰۰۲)

^۱ Nearest Neighbour



شکل ۳-۴: شباهت بین دو خوشه در روش Single-Link برابر است با کمترین فاصله بین داده‌های دو خوشه

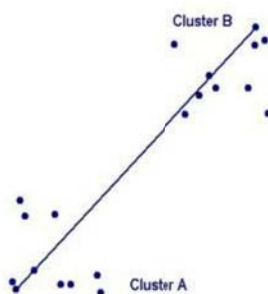
پیوند کامل: فاصله بین خوشه‌ها بر حسب دورترین فاصله ممکنه بین عناصر آنها محاسبه می‌شود.

این روش همانند Single-Link جزء روش‌های خوشه‌بندی سلسله‌مراتبی و انحصاری محسوب می‌شود. به این روش خوشه‌بندی، تکنیک دورترین همسایه^۱ نیز گفته می‌شود. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \max_{i \in A, j \in B} d_{ij}$$

که i یک نمونه داده متعلق به خوشه A و j یک نمونه داده متعلق به خوشه B می‌باشد. در واقع در این روش شباهت بین دو خوشه بیشترین فاصله بین یک عضو از یکی با یک عضو از دیگری است. در شکل زیر این مفهوم بهتر نشان داده شده است. (Wiley & Sons, ۲۰۰۲)

^۱ furthest Neighbour

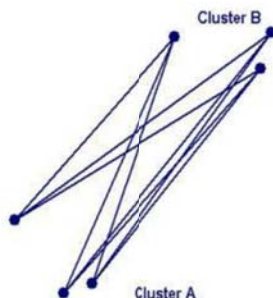


شکل ۴-۴: شباهت بین دو خوشه در روش Complete-Link برابر است با بیشترین فاصله بین داده‌های دو خوشه.

پیوند متوسط: فاصله دو خوشه مساوی مقادیر متوسط کلیه فاصله‌های ممکنه بین عناصر دو خوشه است. این روش همانند Single-Link جزء روش‌های خوشه‌بندی سلسله‌مراتبی و انحصاری محسوب می‌شود. از آنجاکه هر دو روش خوشه‌بندی Single-link و Complete-link به شدت به نویز حساس می‌باشد، این روش که محاسبات بیشتری دارد، پیشنهاد شد. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \frac{\sum_{i \in A, j \in B} d_{ij}}{N_A N_B}$$

که یک نمونه داده متعلق به خوشه A و یک نمونه داده متعلق به خوشه B می‌باشد. و N_A تعداد اعضاء خوشه A و N_B تعداد اعضاء خوشه B است. درواقع در این روش، شباهت بین دو خوشه میانگین فاصله بین تمام اعضاء یکی با تمام اعضاء دیگری است. در شکل زیر این مفهوم بهتر نشان داده شده است. (Wiley & Sons, ۲۰۰۲)

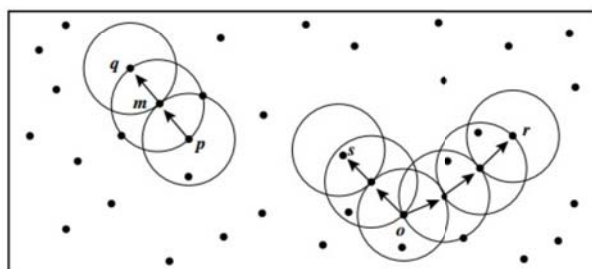


شکل ۴-۵: شباهت بین دو خوشه در روش Average-Link برابر است با میانگین فاصله بین داده‌های دو خوشه

پیوند مرکزی: فاصله بین دو خوشه بر اساس فاصله بین مراکز آن دو خوشه محاسبه می‌شود. برای محاسبه مراکز خوشه‌ها می‌توان از روش‌های مختلفی از جمله روش میانگین استفاده نمود.

۴-۱۰- روش‌های مبتنی بر چگالی

همان‌طور که در روش‌های قبل به‌خصوص روش‌های افرازی مشاهده شد، خوشه‌های حاصل از این روش‌ها اغلب دارای شکل‌های متقارن در فضای مسئله بودند. بدین‌صورت که اغلب حول یک مرکزیت (مثلاً میانگین متغیرهای درون خوشه و یا عنصری که به‌عنوان مرکزیت آن خوشه انتخاب شده بود یعنی medoid) شکل دایره‌ای یا کروی و ... را تشکیل می‌دادند. گاه ممکن است بنا به ماهیت مسئله به دنبال خوشه‌هایی با الگوهای پیچیده‌تر باشیم و یا اینکه رابطه‌ای خاص بین ابعاد مختلف داده‌ها و متغیرها وجود داشته باشد و به دنبال یافتن عناصری باشیم که چنین خصوصیتی را دارند. در این حالت از روش‌های مبتنی بر چگالی استفاده می‌کنیم. ایده اصلی این روش‌ها بر این اساس است که ابتدا به دنبال نقاطی می‌گردیم که چگالی حول آن‌ها زیاد باشد سپس سعی می‌کنیم به گونه‌ای نقاطی را که با این مراکز تجمع در ارتباط هستند پیدا کنیم. گاه پس از طی چند مرحله دو یا چند مرکز تجمع به یکدیگر متصل شده و یک خوشه را شکل می‌دهند این روش‌ها همچنین در حذف داده‌های پرت و مغشوش بسیار مفید هستند.



شکل ۴-۶: خوشه‌بندی مبتنی بر چگالی

یکی از مهم‌ترین الگوریتم‌های مبتنی بر چگالی الگوریتم DBSCAN (یا خوشه‌بندی فضایی بر پایه چگالی برای داده‌های مغشوش) نام دارد. در این

الگوریتم ابتدا برای تمامی نقاط یک شعاع فرضی در نظر می‌گیریم و تعداد نقاطی که اطراف این شعاع فرضی قرار دارند را مشخص می‌کنیم. سپس کاربر باید تعداد نقاط حداقل را برای شروع کار الگوریتم تعریف کند. در پیاده‌سازی این الگوریتم ابتدا نقاط مرکزی را مشخص کرده و هر کدام به‌عنوان یک خوشه در نظر گرفته می‌شوند. سپس نقاط قابل دسترس به آن اضافه می‌شوند و گاه خوشه‌ها را نیز با یکدیگر ادغام می‌کنند. این کار آنقدر تکرار می‌شود تا دیگر تغییری در خوشه‌ها ایجاد نشود، یعنی هیچ عنصری به خوشه‌ها اضافه نشود. ترتیب انتخاب اشیا باید طوری باشد که از عنصری که برای عضویت خوشه به کمترین میزان فاصله نیاز دارد اول از همه موردبررسی قرار گیرد. OPTICS، روشی است که این ترتیب را مشخص می‌کند و برای این کار به محاسبه دو متغیر فاصله مرکزی و فاصله دسترسی نیاز دارد. روش دیگری نیز وجود دارد که بر اساس تابع توزیع چگالی در فضا عمل می‌کند این روش خوشه‌بندی بر پایه چگالی یا به اختصار DENCLUE نام دارد. این روش بر اساس سه ایده استوار است :

- تأثیر هر داده‌ای بر فضا را می‌توان به طور رسمی با یک تابع ریاضی به نام تابع تأثیر مدل کرد. این تابع می‌تواند توصیفی از اثر داده موردبحث بر همسایگی خودش باشد.
- تأثیر کل داده‌ها بر فضا را می‌توان به صورت مدلی متأثر از تمام داده‌های آن فضا بیان نمود.
- خوشه‌ها را می‌توان به طور خودکار با شناسایی عوامل جاذب چگالی در جاهایی که افزایش چگالی وجود دارد مشخص نمود.

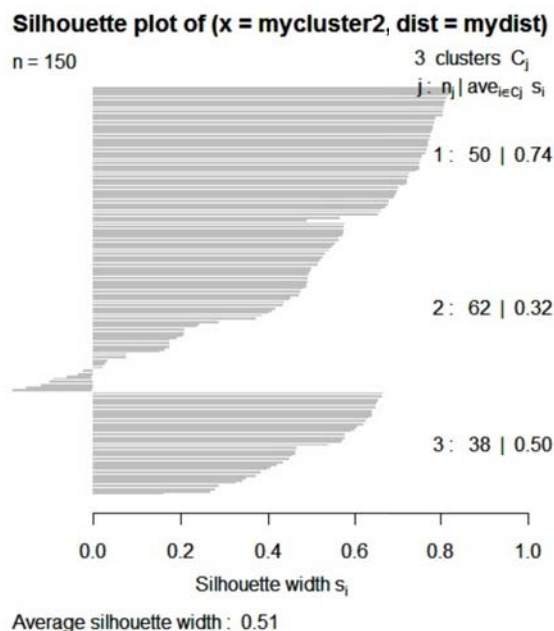
۴-۱۱- ارزیابی خوشه‌بندی با معیار سیلوئت

برای اینکه بفهمیم خوشه‌های خوب گروه‌بندی شده است، باید از یک معیار ارزیابی استفاده نماییم. در این زمینه معیارهای ارزیابی مختلفی همچون dunn، سیلوئت^۱ و ... وجود دارد. معیار سیلوئت برای هر مشاهده حساب می‌شود و متوسط فاصله بین آن مشاهده و اعضای خوشه خودش را تا نزدیک‌ترین مشاهده خوشه

^۱ silhouette

کناری اندازه‌گیری می‌کند. این معیار را می‌توان برای اعضای یک خوشه میانگین‌گیری کرد تا سیلوئت یک خوشه بدست آید و متوسط آن برای کل مشاهدات سیلوئت کل را می‌دهد که مقداری کوچکتر از یک می‌باشد و در صورتی که این مقدار بالاتر از نیم باشد خوشه بندی مناسب می‌باشد. اگر برای هر نقطه متوسط فواصل از بقیه نقاط خوشه خودش را برابر a و حداقل فاصله از نزدیکترین نقطه در خوشه دیگر را برابر b بگیریم آنگاه معیار سیلوئت از فرمول زیر محاسبه می‌گردد :

$$s = (b - a) / \max(a, b)$$



شکل ۴-۷: ارزیابی خوشه‌بندی بر اساس معیار سیلوئت

۱۲-۴- خوشه‌بندی در نرم‌افزار R

خوشه‌بندی k-means

در این بخش خوشه‌بندی k-means را بر روی مجموعه داده Iris اعمال خواهیم نمود. ابتدا ستون مربوط به نوع (گونه) را از مجموعه داده حذف می‌نماییم. سپس

تابع `kmeans()` را بر روی مجموعه داده جدید اعمال نموده و نتایج خوشه‌بندی را در `kmeans.result` ذخیره می‌نماییم.

```
> data <- iris
> data$Species <- NULL
> (kmeans.result <- kmeans(iris$Species, 3))
K-means clustering with 3 clusters of sizes 62, 50, 38

Cluster means:
      Sepal.Length Sepal.Width Petal.Length Petal.Width
1      5.901613      2.748387      4.393548      1.433871
2      5.006000      3.428000      1.462000      0.246000
3      6.850000      3.073684      5.742105      2.071053

Clustering vector:
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[40] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[79] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[118] 3 3 1 3 1 3 1 3 3 1 1 3 3 3 3 3 3 1 3 3 3 3 1 3 3 3 3 1 3 3 3 1 3

Within cluster sum of squares by cluster:
[1] 39.82097 15.15100 23.87947
      (between_SS / total_SS =  88.4 %)

Available components:

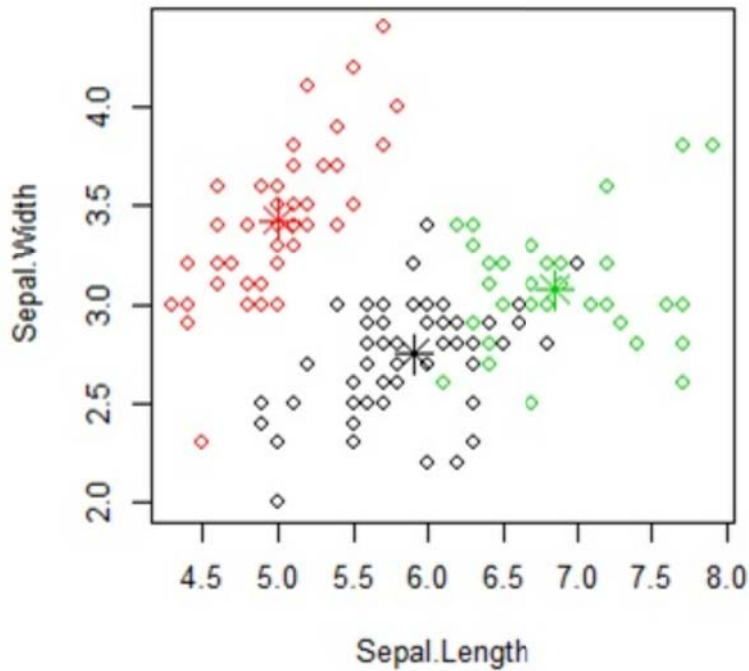
[1] "cluster" "centers" "totss" "withinss" "tot.withinss"
[6] "betweenss" "size" "iter" "ifault"

> table(iris$Species, kmeans.result$cluster)

      1  2  3
setosa   0 50  0
versicolor 48  0  2
virginica 14  0 36
```

نتایج خوشه‌بندی با دسته‌بندی اولیه مجموعه داده Iris مقایسه می‌گردد تا شباهت این نتایج با گروه‌بندی اولیه مقایسه گردد. نتایج بالا نشان‌دهنده این است که گونه *setosa* می‌تواند کاملاً مجزا از دو گونه دیگر باشد. اما دو گونه دیگر دارای مقداری هم‌پوشانی هستند. در شکل زیر شکل مربوط به این خوشه‌بندی (البته با دو ویژگی از گونه‌ها) را مشاهده می‌فرمایید.


```
> plot(data[c("Sepal.Length", "Sepal.Width")],
+ col = kmeans.result$cluster)
> points(kmeans.result$centers[,c("Sepal.Length",
+ "Sepal.Width")], col = 1:3, pch = 8, cex=2)
```



شکل ۴-۷: خوشه‌بندی k-means بر روی مجموعه داده Iris

خوشه‌بندی k-medoids

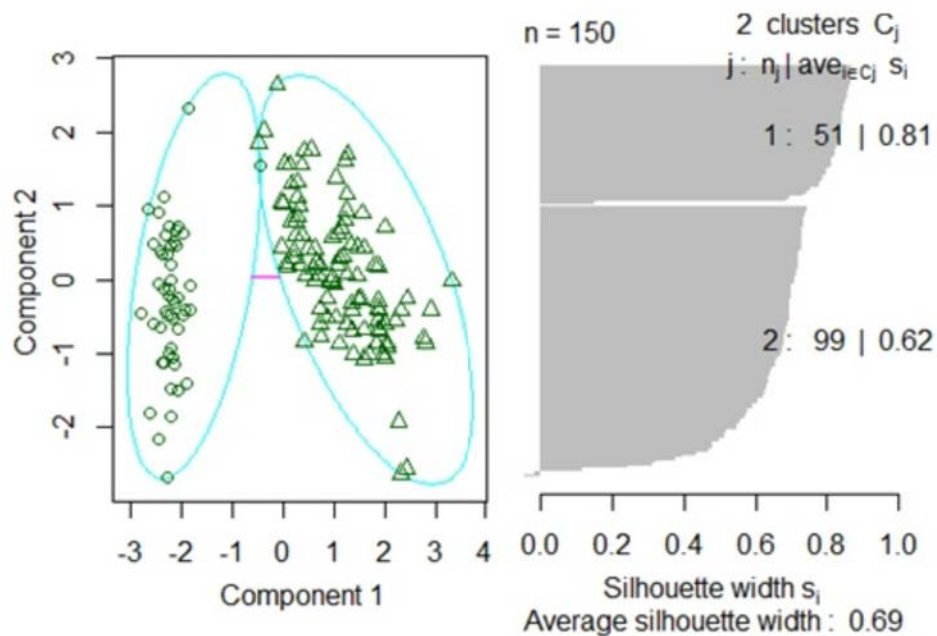
در این بخش می‌خواهیم خوشه‌بندی k-medoids را با استفاده از تابع `pam()` و `pamk()` بررسی نماییم. استفاده از تابع `clara()` نیز زمانی می‌باشد که تعداد داده بسیار زیادی داشته باشیم. البته لازم به ذکر است که روش سیلوئت اشاره به متدی برای تفسیر و اعتبارسنجی خوشه‌های داده‌ها دارد. این تکنیک یک شکل خلاصه گرافیکی از قرار گرفتن داده‌ها در خوشه‌ها را نمایش می‌دهد.

```
> library(fpc)
> pamk.result <- pamk(data)
> pamk.result$nc
[1] 2
> table(pamk.result$pamobject$clustering,
       iris$Species)
```

	setosa	versicolor	virginica
1	50	1	0
2	0	49	50

```
> layout(matrix(c(1,2),1,2))
```

```
> plot(pamk.result$pamobject)
> layout(matrix(1))
```



شکل ۴-۸: مصورسازی خوشه‌بندی k-medoids بر روی مجموعه داده Iris با استفاده از

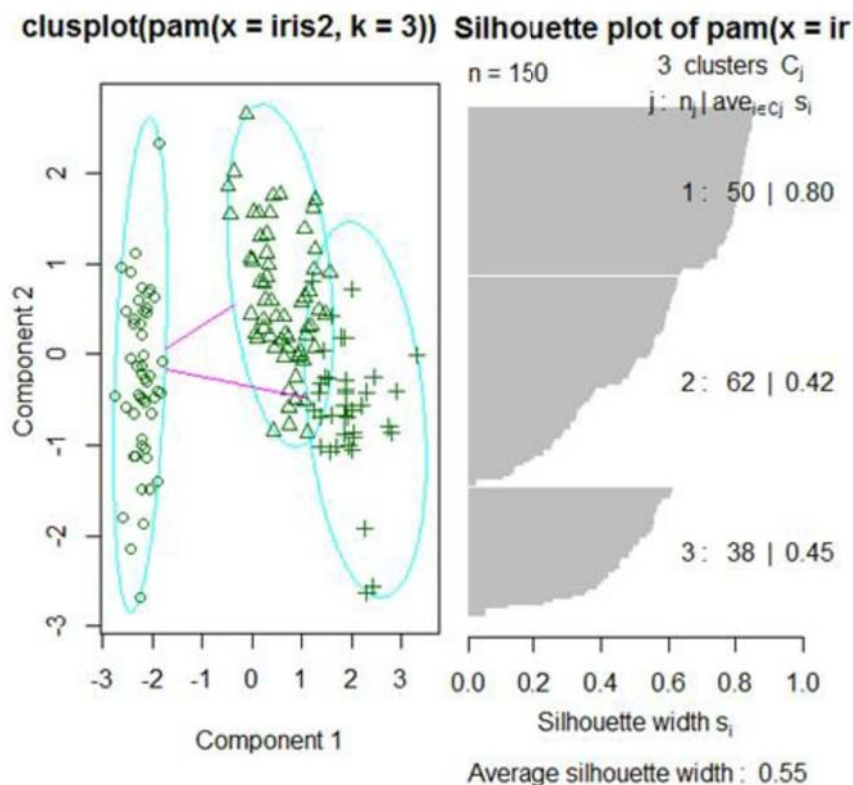
تابع pamk()

```
> pam.result <- pam(iris2, 3)
> table(pam.result$clustering, iris$Species)
```

	setosa	versicolor	virginica
1	50	0	0
2	0	48	14
3	0	2	36

در مثال قبل تابع `pam()` دو خوشه ایجاد می‌نماید، یک خوشه برای گونه `setosa` و دیگری برای دو گونه دیگر مجموعه داده `Iris`. در شکل زیر نمودار سمت چپ یک نمودار دوبعدی برای دو خوشه است که خط نمایش داده شده نمایشگر فاصله بین خوشه‌هاست. نمودار سمت چپ نیز نشان‌دهنده شکل `silhouettes` این خوشه‌ها می‌باشد.

```
> layout(matrix(c(1,2),1,2))
> plot(pam.result)
> layout(matrix(1))
```



شکل ۴-۹ : خوشه‌بندی `k-medoids` بر روی مجموعه داده `Iris` با استفاده از تابع `pam()`

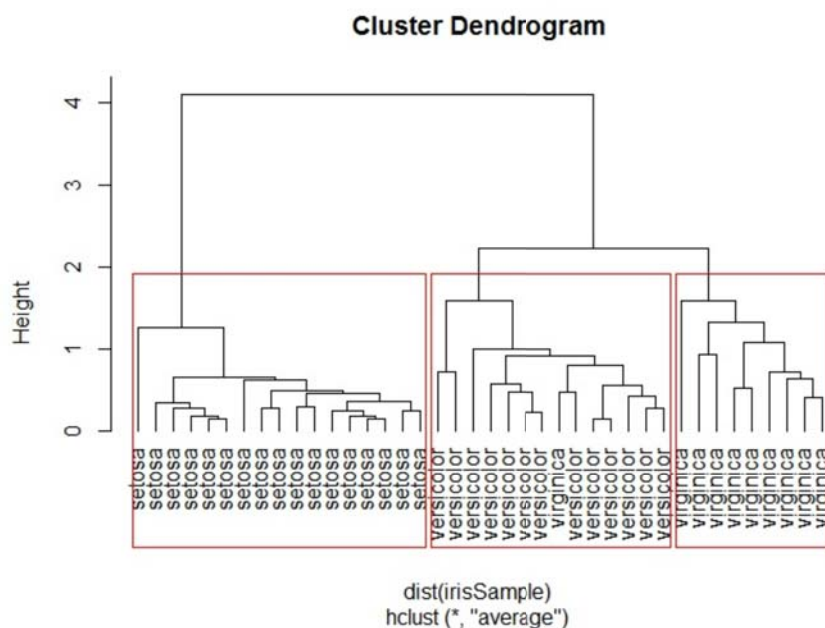
در شکل قبلی نتایج با استفاده از تابع `pam()` نشان داده شده است که از سه خوشه استفاده شده است. خوشه اول نشان‌دهنده گونه `setosa`، خوشه دوم نشان‌دهنده داده `versicolor` و خوشه سوم نشان‌دهنده گونه `virginica` از مجموعه داده `Iris` می‌باشد. درواقع نمی‌توان گفت کدام یک از روش‌های `pam()`

و یا `pamk()` بهتر می‌باشد. استفاده از این روش‌ها بستگی به نوع مسئله موردنظر و دامنه دانش و تجربه می‌باشد. در این مسئله نتایج تابع `pam()` به نظر بهتر می‌رسد به خاطر اینکه سه خوشه را بهتر توانسته است تشخیص دهد. در شکل چیدمان نقاط با توجه به الگوریتم MDS (مقیاس‌گذاری چندبعدی) ایجاد شده است که بر اساس حفظ فاصله نقاط در فضای دوبعدی با توجه به فاصله‌ی اصلیشان در فضای چندبعدی کار می‌کند.

خوشه‌بندی سلسله مراتبی

در این بخش خوشه‌بندی سلسله مراتبی را با استفاده از تابع `hclust()` در مجموعه داده Iris بررسی خواهیم نمود. ابتدا ۴۰ رکورد از مجموعه داده Iris را برای ترسیم بهتر نمودارها و جلوگیری از شلوغی استفاده می‌نماییم. متغیر گونه گل نیز از مجموعه داده حذف می‌گردد. البته ابتدا باید کتابخانه `cluster` با استفاده از دستور `library(cluster)` در ابتدای کدها قرار داده شود.

```
> idx <- sample(1:dim(iris)[1], 40)
> irisSample <- iris[idx,]
> irisSample$Species <- NULL
> hc <- hclust(dist(irisSample), method="ave")
> plot(hc, hang = -1, labels=iris$Species[idx])
> rect.hclust(hc, k=3)
> groups <- cutree(hc, k=3)
```



شکل ۴-۱۰: خوشه‌بندی سلسله مراتبی با استفاده از تابع `hclust()` در مجموعه داده Iris

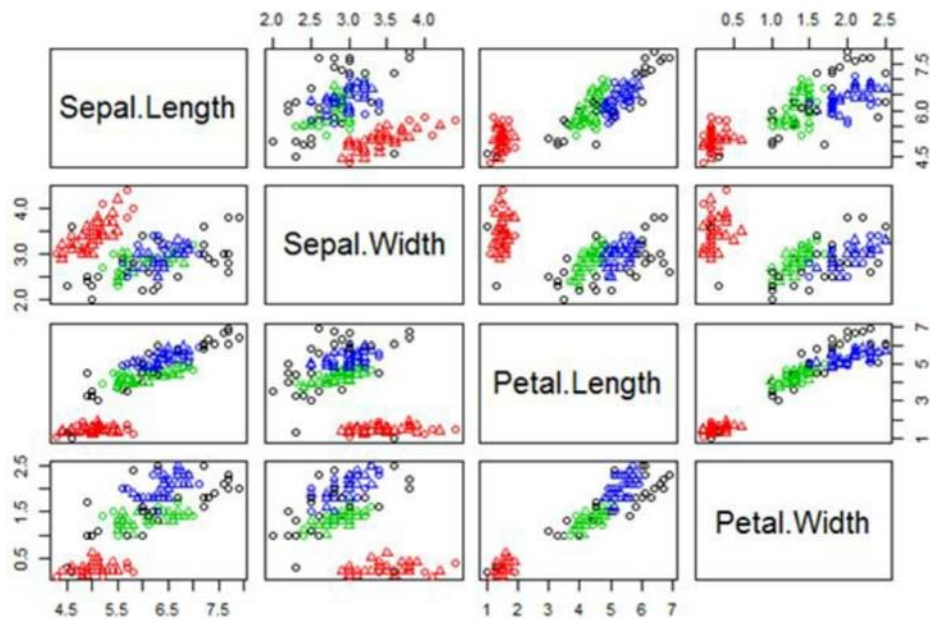
خوشه‌بندی مبتنی بر چگالی

الگوریتم DBSCAN با استفاده از بسته `fpc` یک روش خوشه‌بندی مبتنی بر چگالی را داده‌های عددی مهیا می‌کند. این روش بر اساس ایجاد خوشه‌ها و داده‌های فشرده در یک ناحیه متراکم که به هم متصل شده‌اند عمل می‌کند. دو پارامتر کلیدی اساسی در این الگوریتم مورد توجه می‌باشد. پارامتر اول با نام `eps` به معنای فاصله دسترسی‌پذیری که انداز همسایگی را تعریف می‌کند و پارامتر دوم با عنوان `MinPts` که حداقل تعداد نقطه‌ها را نشان می‌دهد. اگر تعداد نقاط در همسایگی یک نقطه کمتر از میزان `MinPts` نباشد آن نقطه یک نقطه فشرده است. در شکل زیر خوشه‌بندی با استفاده از الگوریتم مبتنی بر چگالی را مشاهده می‌نمایید.

```
> library(fpc)
> data <- iris[-5]
> v <- dbscan(data, eps=0.42, MinPts=5)
> table(v$cluster, iris$Species)
```

	setosa	versicolor	virginica
0	2	10	17
1	48	0	0
2	0	37	0
3	0	3	33

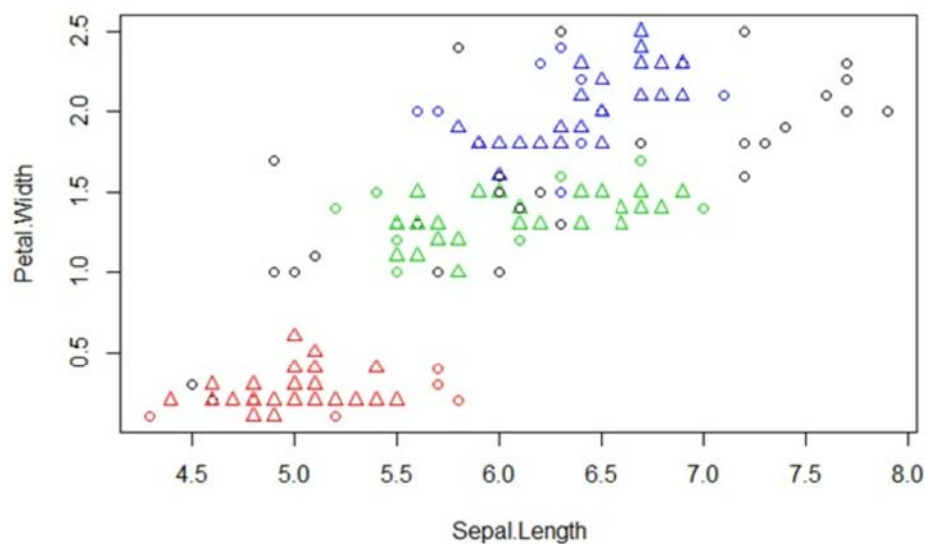
```
> plot(v,data)
```



شکل ۴-۱۱: خوشه‌بندی مبتنی بر چگالی در مجموعه داده Iris

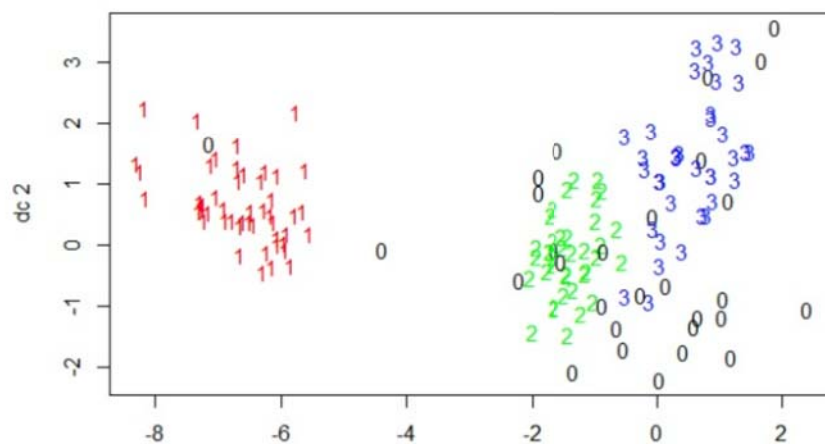
در جدول بالا در میان کدها اعداد ۱ الی ۳ در ستون اول سه خوشه شناخته هستند. عدد صفر برای داده‌های پرت و نویز است که نشان‌دهنده داده‌هایی است که متعلق به هیچ خوشه‌ای نیستند که در نمودار به صورت دایره مشکی رنگ دیده می‌شود. خوشه‌ها هم به صورت نقاط پراکنده با استفاده از ستون‌های اول و چهارم داده‌ها در شکل زیر نشان داده شده‌است.

```
> plotcluster(data, v$cluster)
```



شکل ۴-۱۲: مصورسازی خوشه‌بندی مبتنی بر چگالی و نقاط پرت
 راه دیگر برای نمایش خوشه‌ها استفاده از تابع `plotcluster()` در بسته `fpc` می‌باشد که در شکل زیر مشاهده می‌فرمایید. توجه داشته باشید که داده‌ها در کلاس‌های متفاوت قرار گرفته‌اند.

```
> plot(v, data)
```

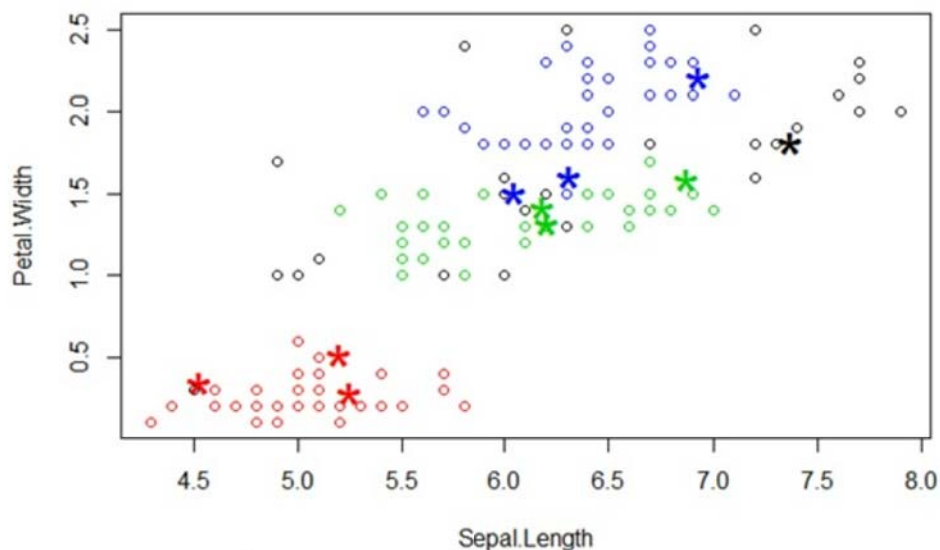


شکل ۴-۱۳: مصورسازی خوشه‌بندی مبتنی بر چگالی و نقاط پرت با استفاده از تابع `plotcluster()`

مدل خوشه‌بندی می‌تواند از داده‌های برچسب‌گذاری شده استفاده نماید که بر اساس مشابهت بین داده‌های جدید و خوشه‌ها می‌باشد. مثال زیر نمونه‌ای از داده سطر از داده‌های Iris را نمایش می‌دهد که تعدادی نویز به آن‌ها اضافه شده و مجموعه داده جدیدی تشکیل گردیده است. نویزهای تصادفی با استفاده از تابع `runif()` تولید شده است

```
> set.seed(435)
> idx <- sample(1:nrow(iris), 10)
> nd <- iris[idx,-5]
> nd <- nd + matrix(runif(10*4, min=0, max=0.2), nrow=10, ncol=4)
> myPred <- predict(v, data, nd)
> plot(data[c(1,4)], col=1+v$cluster)
> points(nd[c(1,4)], pch="*", col=1+myPred, cex=3)
> table(myPred, iris$Species[idx])
```

myPred	setosa	versicolor	virginica
0	0	0	1
1	3	0	0
2	0	3	0
3	0	1	2



شکل ۴-۱۴ : مصورسازی خوشه‌بندی مبتنی بر چگالی و نقاط پرت با استفاده از تابع `runif()` همان‌طور که در نتایج بالا می‌بینیم داده‌های جدید به‌صورت ستاره نمایش داده می‌شود.

فصل پنجم

دست‌بندی و پیش‌بینی

۵-۱- دسته‌بندی

دسته‌بندی و پیش‌بینی دو نوع عملیات برای تحلیل داده‌ها و استخراج مدل به‌منظور توصیف دسته‌های مهم داده‌ها، فهم و پیش‌بینی رفتار آینده آن‌ها می‌باشد. مدل‌های دسته‌بندی در پیش‌بینی متغیرهای گسسته و طبقه‌ای بکار رفته و مدل‌های پیش‌بینی یا رگرسیون بیشتر بر روی داده‌های پیوسته بکار می‌رود.

به‌عنوان مثال یک مدل دسته‌بندی ممکن است برای دسته‌بندی کردن وام‌های بانک به دو طبقه وام‌های بی‌خطر و پرخطر به کار رود درحالی‌که مدل‌های پیش‌بینی به کار گرفته شده در این کسب‌وکار خاص سعی در پیش‌بینی میزان مخارج و هزینه‌های مشتریان بر اساس ویژگی‌های درآمدی و شغلی آن‌ها دارند. دسته‌بندی، هر جزء از داده‌ها را بر مبنای اختلاف بین داده‌ها به مجموعه‌های از پیش تعریف شده دسته‌ها تصویر می‌کند. باید بگوییم دسته‌بندی یادگیری با نظارت می‌باشد که دسته‌ها از قبل مشخص می‌باشند.

دسته‌بندی داده‌ها، فرایند دومرحله‌ای می‌باشد. اولین مرحله ساخت مدل و دومین مرحله استفاده از مدل و پیش‌بینی از طریق داده‌های قبلی می‌باشد.

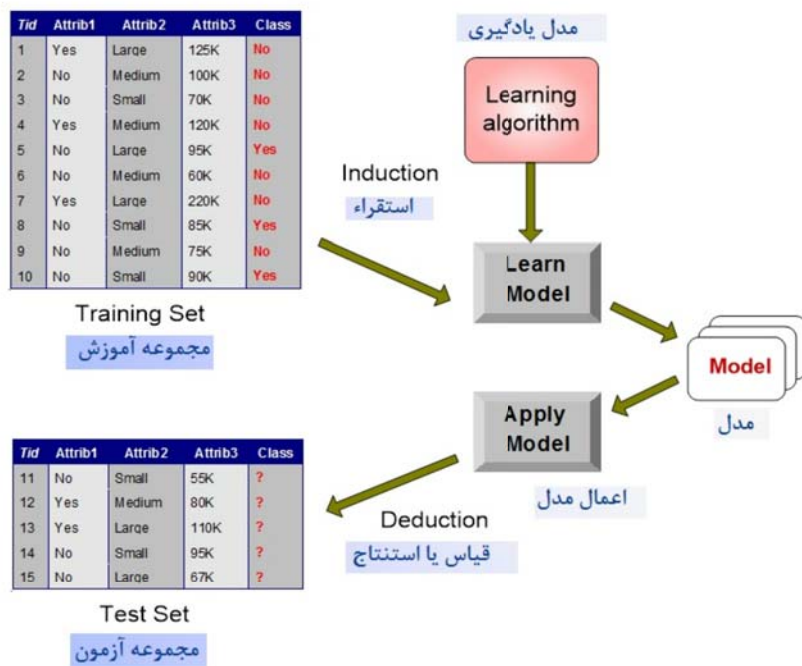
مرحله اول ساخت مدل: عبارت است از توصیف یک سری از دسته‌های از پیش تعیین شده بر مبنای مجموعه داده‌های آموزش مدل که البته این فرآیند، یادگیری نیز نامیده می‌شود در این فرآیند سعی می‌شود با توجه به نمونه‌های موجود مدلی ساخته شود که بر اساس آن بتوان داده‌ای فاقد برچسب دسته را در دسته‌های مربوط به خودشان قرار داد.

مرحله دوم استفاده از مدل: این مرحله دارای دو بخش می‌باشد. در بخش نخست مدل ساخته‌شده مورد آزمون واقع می‌شود تا دقت پیش‌بینی آن بررسی شود. در بخش دوم نیز مدلی که دارای دقت مناسبی است، برای دسته‌بندی داده‌ها به کار گرفته می‌شود. به‌منظور تخمین دقت پیش‌بینی مدل مجموعه‌ای از داده‌های آزمایشی به طور اتفاقی از میان داده‌ها انتخاب می‌شوند و مدل روی آن‌ها اجرا

می‌شود. هدف اصلی در اینجا بالاتر بودن تخمین دقت مدل می‌باشد تا به هنگام استفاده هر داده را به دسته مناسب آن تخصیص دهد. (غضنفری، علیزاده، تیمورپور، ۱۳۸۷)

راه متداول ساخت مدل دسته‌بندی

راه متداول ساخت مدل دسته‌بندی را در شکل زیر مشاهده می‌نمایید. در شکل زیر استقراء یعنی رسیدن به نتیجه کلی از طریق مشاهدات جزئی و مکرر و استنتاج نیز یعنی می‌توان از مقدمات کلی به نتایج جزئی رسید. در مورد پیش‌بینی نیز این مدل صادق می‌باشد.



شکل ۵-۱: راه متداول ساخت دسته‌بندی

۵-۲- تفاوت دسته‌بندی و خوشه‌بندی

دسته‌بندی، هر جز از داده‌ها را بر مبنای اختلاف بین داده‌ها به مجموعه‌های از پیش تعریف شده دسته‌ها تصویر می‌کند. درحالی‌که خوشه‌بندی، داده‌ها را به گروه‌های مختلف (خوشه‌ها) که از قبل معین نیستند، (بر اساس مشابهت درون خوشه و تفاوت بیرون خوشه) تقسیم می‌کند. لذا اگر بخواهیم با استفاده از مفهوم یادگیری، دسته‌بندی و خوشه‌بندی را متمایز کنیم، باید بگوییم دسته‌بندی یادگیری با نظارت و خوشه‌بندی یادگیری بدون نظارت است. یادگیری با نظارت یا دسته‌بندی عبارت است از یادگیری به‌وسیله نمونه‌ها. به عبارت دیگر در این روش دسته‌ها از قبل مشخص هستند. ولی در یادگیری بدون نظارت یا خوشه‌بندی، خوشه‌ها مشخص نیستند و هدف خوشه‌بندی تعیین خوشه‌های داده‌ها است. (غضنفری، علیزاده، تیمورپور، ۱۳۸۷)

روش‌های زیادی برای دسته‌بندی وجود دارد که از آن جمله می‌توان به موارد ذیل اشاره کرد :

- شبکه‌های بیزی
- نزدیک‌ترین همسایگی
- شبکه‌های عصبی
- درخت تصمیم
- رگرسیون

۵-۳- بیز^۱ ساده

بیز روشی برای دسته‌بندی پدیده‌ها، بر پایه احتمال وقوع یا عدم وقوع یک پدیده است و در نظریه احتمالات با اهمیت و پرکاربرد است. اگر برای فضای

^۱ Bayes' theorem

نمونه‌ای مفروضی بتوانیم چنان افرازی انتخاب کنیم که با دانستن اینکه کدامیک از پیشامدهای افراز شده رخ داده‌است، بخش مهمی از عدم‌اطمینان تقلیل یابد.

این قضیه از آن جهت مفید است که می‌توان از طریق آن احتمال یک پیشامد را با مشروط کردن نسبت به وقوع و یا عدم وقوع یک پیشامد دیگر محاسبه کرد. در بسیاری از حالت‌ها، محاسبه احتمال یک پیشامد به صورت مستقیم کاری دشوار است. با استفاده از این قضیه و مشروط کردن پیشامد مورد نظر نسبت به پیشامد دیگر، می‌توان احتمال مورد نظر را محاسبه کرد.

۴-۵- دسته‌بندی بر مبنای نزدیک‌ترین همسایگی

در تقسیم‌بندی‌های مربوط به دسته‌بندی‌ها آن‌ها را به دو دسته کلی مشتاق و کاهل تقسیم‌بندی می‌کنند. دسته‌بند مشتاق، نوعی از مدل داده‌هاست که در مرحله آموزش ایجاد می‌گردد. یکی از نمونه‌های دسته‌بندی‌های مشتاق درخت‌های تصمیم می‌باشند که با دریافت نمونه‌های آموزشی، مدلی به شکل درخت ایجاد می‌کند. حالت دیگری از دسته‌بندی‌ها به کاهل شناخته می‌شوند. در این حالت از دسته‌بندی‌ها نمونه‌ای آموزشی دریافت و ذخیره شده و تنها در موقع دسته‌بندی از آن‌ها استفاده می‌گردد. به عبارت دیگر تا اینجا مدلی از داده‌ها ساخته نشده و یادگیری تا زمان دسته‌بندی به تعویق می‌افتد. این نوع دسته‌بندی‌ها، یادگیری مبتنی بر نمونه نیز نامیده می‌شوند.

تمایز اصلی بین این دو متد این است که مدل مشتاق زمان زیادی را در مرحله آموزش، صرف ساخت مدل نموده و در زمان دسته‌بندی بسیار سریع عمل می‌کند، ولی در طرف مقابل، نوع کاهل آن، در هنگام ورود داده‌ها در مرحله آموزش، فقط آن‌ها را ذخیره نموده و زمان بیشتری را صرف دسته‌بندی می‌نماید. هر کدام از این متدها دارای کارایی مربوط به خود می‌باشند که در ادامه نگاهی خواهیم داشت. به عنوان مثال یکی از نمونه‌های دسته‌بندی کاهل مدل نزدیک‌ترین همسایگی می‌باشد.

در مدل‌های دسته‌بندی الگوریتم نزدیک‌ترین همسایگی دارای سه مرحله زیر می‌باشد :

- در گام اول فاصله نمونه‌های ورودی را با تمام نمونه‌های آموزشی محاسبه می‌نماییم.
- سپس در گام بعدی نمونه‌های آموزشی را مبتنی بر فاصله و انتخاب نمودن k همسایه نزدیک‌تر مرتب می‌نماییم.
- گام سوم عبارت است از استفاده از دسته‌ای که اکثریت را در همسایه‌های نزدیک، به‌عنوان تخمینی برای دسته نمونه ورودی دارد.

۵-۵- شبکه‌های عصبی^۱

شبکه‌های عصبی روشی است که قصد دارد با استفاده از مدل‌های ریاضی و توان کامپیوتر، برخی از جنبه‌های ساده مغز انسان را شبیه‌سازی کند. شبکه‌های عصبی به‌صورت یکی از بخش‌های پیچیده مغز انسان، به‌عنوان یک ساختار یادگیری غیرقابل درک مشهور شده است. این ساختار پیچیده از مجموعه‌ای از نرونها به وجود آمده است که خود نرونها ساختار ساده‌ای داشته ولی شبکه اتصال این نرونها وظایف یادگیری بسیار پیچیده‌ای را به انجام می‌رساند. لذا شناخت و درک ساختار بیولوژی مغز انسان می‌تواند ما را در ایجاد شبکه‌های عصبی مصنوعی به‌عنوان یک ابزار کارا در حل مسائل و کاربردهای علمی و فنی یاری رساند.

یکی از کاربردهای بارز شبکه‌های عصبی مصنوعی در داده‌کاوی می‌باشد. تا آنجایی که حوزه‌ای تحت عنوان داده‌کاوی بر مبنای شبکه‌های عصبی به وجود آمده است. شبکه‌های عصبی مصنوعی در برخی از عملیات مانند پیش‌بینی و دسته‌بندی در مقایسه با سایر روش‌ها دارای مزایای نسبی بوده و معمولاً در کارهای اجرایی ترجیح داده می‌شوند. در این بخش ضمن آشنایی با مفاهیم و اصل مورد نیاز شبکه‌های عصبی برای به‌کارگیری در مسائل داده‌کاوی، سعی می‌شود کاربرد شبکه‌های عصبی در دسته‌بندی تشریح گردد.

^۱ Neural Networks

از جمله مزایای شبکه‌های عصبی می‌توان به قابلیت مواجهه با داده‌های مغشوش، قابلیت استفاده در زمانی که دانش بسیار کمی در مورد مسئله وجود دارد، مناسب برای هر دو نوع داده کمی و کیفی، کاربرد در مسائل متفاوتی از پردازش تصویر گرفته تا تشخیص درمان و سرعت بالا نسبت به سایر روش‌ها به دلیل کارکرد موازی اشاره نمود.

این شبکه‌ها می‌توانند به عنوان روش مناسب در ایجاد مدل‌های تحلیلی تخمینی و برخورد با داده‌های متفاوت سازمانی در حوزه‌ها و پروژه‌های متفاوت داده‌کاوی به کار گرفته شوند. به طور مثال در عملیات پیش‌بینی و سری‌های زمانی، داده‌های پیچیده مالی، داده‌های بورس و شبکه‌های عصبی کاربردهای فراوانی دارند. (غضنفری، علیزاده، تیمورپور، ۱۳۸۷)

۵-۶- درخت تصمیم^۱

ساختار درخت تصمیم یک ساختار درختی، شبیه فلوچارت است. بالاترین گره در درخت، گره ریشه است و گره‌های برگ، دسته‌ها یا توزیع دسته‌ها را نشان می‌دهند. تصویر یک درخت تصمیم نمونه را در شکل زیر می‌بینید.

درخت تصمیم یکی از ابزارهای قوی و متداول برای دسته‌بندی و پیش‌بینی می‌باشد. درخت تصمیم برخلاف شبکه‌های عصبی به تولید قاعده می‌پردازند. ساختار درخت تصمیم، پیش‌بینی به دست آمده از درخت در قالب یکسری قواعد توضیح داده می‌شود. درحالی‌که در شبکه‌های عصبی تنها نتیجه پیش‌بینی بیان می‌شود و چگونگی به دست آمدن آن‌ها در خود شبکه پنهان می‌ماند. همچنین در درخت تصمیم برخلاف شبکه‌های عصبی ضرورتی وجود ندارد که داده‌ها لزوماً به صورت عددی باشند. در برخی موارد تنها صحت دسته‌بندی و پیش‌بینی مهم است و لزوماً ارائه توضیحی برای پیش‌بینی انجام شده نیاز نیست.

^۱ Decision tree

در مورد خصوصیات درخت تصمیم به موارد زیر می‌توان اشاره نمود :

- روش درخت تصمیم در تقسیم‌بندی داده‌ها به گروه‌های مختلف به گونه‌ای است که هیچ داده‌ای حذف نمی‌شود (تعداد داده‌ها در گروه مادر با مجموع داده‌ها در شاخه‌های درخت ایجادشده برابر هستند)
- استفاده از درخت تصمیم آسان می‌باشد.
- درک مدل ایجاد شده توسط درخت تصمیم آسان می‌باشد. به عبارت دیگر با وجود اینکه فهمیدن روش کار الگوریتم‌های سازنده درخت، چندان ساده نیست ولی فهمیدن نتایج به دست آمده از آن‌ها آسان است.
- دسته‌بندی‌هایی که توسط درخت تصمیم ایجاد می‌شوند از روی شباهت داده‌های ذخیره شده در پارامترهای پیش‌بینی کننده قابل انجام می‌باشد.

در ایجاد درخت تصمیم نیز یکسری سؤال وجود دارد و با مشخص شدن پاسخ هر سؤال یک سؤال دیگر پرسیده می‌شود. اگر سؤال‌ها درست و مناسب با ویژگی‌ها پرسیده شوند، یک مجموعه کوتاه از سؤالات برای پیش‌بینی کردن دسته مربوطه به هر شیء جدید کافی می‌باشد.

در مسائل مرتبط با درخت‌های تصمیم با دو نوع از متغیرها مواجه هستیم :

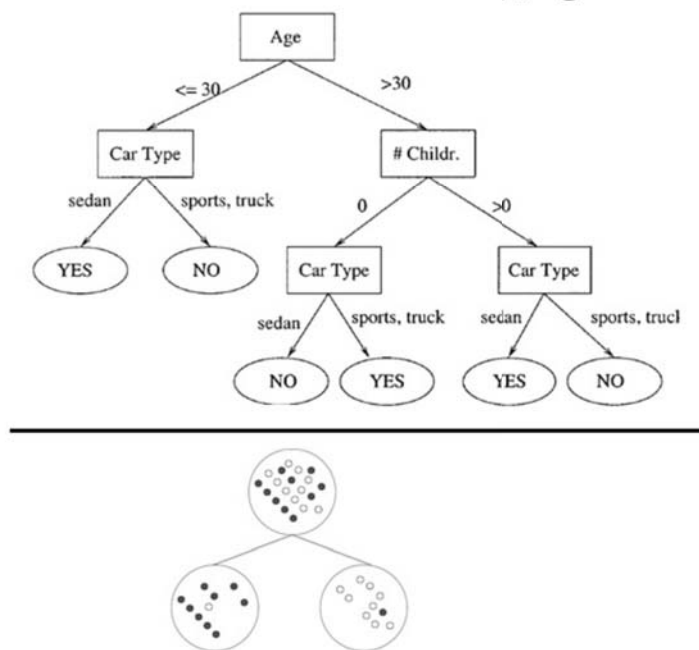
- متغیرهای عددی مثل مشخصه سن که مقادیر آن عددی است.
- متغیرهای طبقه‌ای مثل مشخصه نوع ماشین که مقادیر آن متنی و گروهی است.

پیدایش درخت تصمیم از دو مرحله رشد و ایجاد درخت و مرحله هرس درخت با هدف حداقل کردن خطای پیش‌بینی تشکیل می‌شود.

(غضنفری، علیزاده، تیمورپور، ۱۳۸۷)

یک مدل توصیفی به اندازه کافی واقعی می‌باشد که امکان پیش‌بینی و استدلال می‌دهد و در ابعاد زیر تشریح می‌گردد :

- آموزش^۱: به کمک برخی از مشاهدات تعدادی مدل می‌سازیم.
- اعتبار^۲: با برخی مشاهدات دیگر یکی از این مدل‌ها را انتخاب می‌کنیم.
- آزمون^۳: مدل انتخاب‌شده را روی مشاهدات جدیدتر امتحان می‌کنیم.



شکل ۵-۲: نمونه‌ای از درخت تصمیم

۵-۷- پیش‌بینی

پیش‌بینی عبارت است از تعیین مقدار یک متغیر پاسخ پیوسته (متغیر وابسته) بر حسب مقادیر متغیرهای مستقل، پیش‌بینی مشابه دسته‌بندی است با این تفاوت که متغیر وابسته در دسته‌بندی، گسسته می‌باشد. برآورد حقوق فارغ‌التحصیلان با ۱۰ سال تجربه کاری یا فروش بالقوه یک محصول جدید بر حسب قیمت آن،

^۱ Train

^۲ Validation

^۳ Test

مواردی از پیش‌بینی می‌باشند. مهم‌ترین روش مورد استفاده در پیش‌بینی عددی، رگرسیون است.

البته برخی دیگر از روش‌های دسته‌بندی نظیر الگوریتم پس انتشار و ماشین‌های بردار پشتیبان نیز می‌توانند به عنوان روش‌های پیش‌بینی مورد استفاده قرار گیرند. در داده کاوی، متغیرهای مستقل و متغیر وابسته همان ویژگی‌های تشریح شده برای هر نمونه یا مشاهده می‌باشند. معمولاً مقادیر متغیرهای مستقل معلوم است. هر چند با استفاده از روش‌های خاصی، می‌توان مواردی که در آن‌ها بعضی از مقادیر مفقوده را نیز پیش‌بینی کرد. در بسیاری از موارد با بکار بردن روش‌های تبدیل و تغییر متغیر، می‌توان یک مسئله غیرخطی را با استفاده از رگرسیون خطی حل کرد. (غضنفری، علیزاده، تیمورپور، ۱۳۸۷)

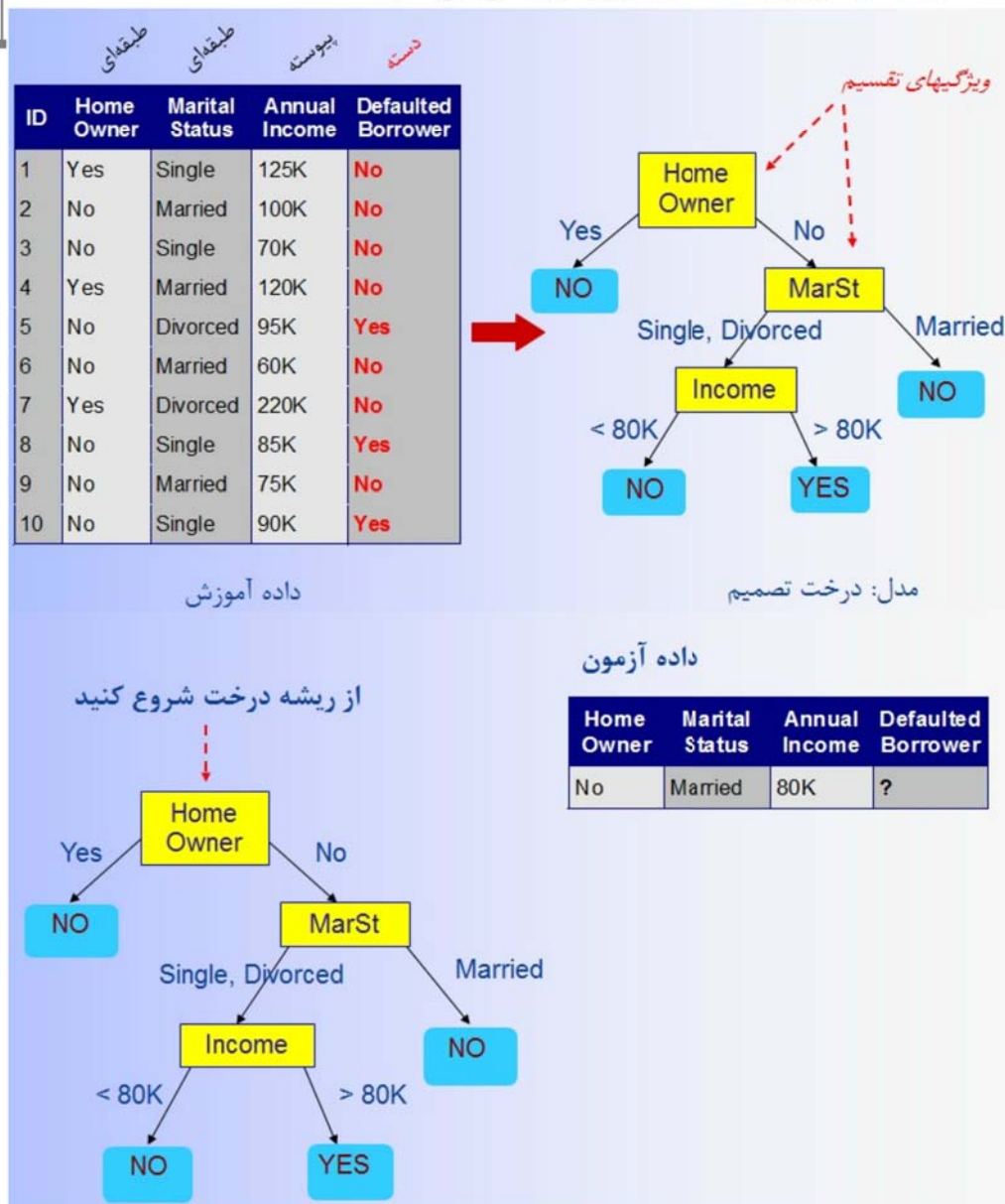
۵-۸- قواعد انجمنی^۱

همزمان با پیدایش علم داده کاوی در اوایل دهه ۹۰ الگوریتم‌های استخراج قوانین وابستگی از پایگاه داده‌ها نیز پا به عرصه گذاشت. نویسندگان زیادی در زمینه استخراج قوانین وابستگی در پایگاه داده‌ها بحث کرده‌اند رابرت. اس (۲۰۰۳) در مقاله خود اقدام به مقایسه الگوریتم‌های مهم استخراج قوانین وابستگی پرداخته است. در این مطالعه به مزیت‌ها و معایب سه الگوریتم مهم مورد استفاده در استخراج قوانین وابستگی یعنی *Apriori*, *Sampling* و *Partitoning* پرداخته است.

اساساً ارتباط میان مجموعه اشیاء (چیزها) وابستگی‌های جالب توجهی هستند که منجر به امکان آشکارسازی الگوهای مفید و قوانین وابستگی برای پشتیبانی تصمیم، پیش‌بینی‌های مالی، سیاست‌های بازاریابی، وقایع پزشکی و خیلی کاربردهای دیگر می‌شود. در حقیقت توجهات زیادی را در تحقیقات اخیر به خود جلب کرده است. تحلیل وابستگی‌ها یک حالت غیرنظارتی داده کاوی می‌باشد که به جستجو برای یافتن ارتباط در مجموعه داده‌ها می‌پردازد. یکی از

^۱ association rule

کاربردی‌ترین حالات تحلیل وابستگی‌ها «تجزیه تحلیل سبد بازار» می‌باشد که در آن هدف یافتن کالاهایی است که معمولاً به طور همزمان خریداری می‌شوند. این کار کمک می‌کند که خرده‌فروشان بهتر بتوانند کالاهای خود را سازمان‌دهی کرده و چیدمان بهتری از محصولات خود داشته باشند. در شکل زیر مثالی از درخت تصمیم را مشاهده می‌نمایید که توانایی برگرداندن وام گرفته شده را برای داده‌های جدید از روی داده‌های موجود بررسی می‌نماید.



شکل ۵-۳: مثالی از درخت تصمیم برگرداندن وام

۵-۹- دسته‌بندی در نرم‌افزار R

ترسیم درخت تصمیم با استفاده از بسته party

این بخش پیرامون ساخت یک مدل پیش‌بینی کننده با استفاده از بسته‌های `party`، `rpart` و `randomforest` می‌باشد. ابتدا با ساخت یک درخت تصمیم با استفاده از بسته `party` و استفاده از ساخت درخت برای دسته‌بندی و درخت تصمیم با بسته `rpart` شروع نموده و در نهایت با مثالی از یک جنگل تصادفی با بسته `randomForest` این بخش را به پایان می‌رسانیم.

این بخش نشان می‌دهد که چطور یک درخت تصمیم با استفاده از تابع `ctree()` در بسته `party` بر روی مجموعه داده `Iris` بسازیم. جزئیات این مجموعه داده را در فصل‌های پیشین اشاره نمودیم که دارای چهار ویژگی می‌باشد که از روی آن‌ها می‌توان گونه مجموعه داده را پیش‌بینی نمود. در این بسته تابع `ctree()` یک درخت تصمیم می‌سازد و تابع `predict()` یک پیش‌بینی بر روی داده‌های جدید انجام می‌دهد. قبل از مدل‌سازی داده‌های `Iris` به دو زیرمجموعه تقسیم‌بندی شده‌است. بخش آموزشی شامل ۷۰ درصد و بخش تست شامل ۳۰ درصد دیگر داده‌ها.

```
> set.seed(1234)
> var <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))
> trd <- iris[var==1,]
> tsd <- iris[var==2,]
> library(party)
> frml <- Species ~ Sepal.Length + Sepal.Width + Petal.Length +
  Petal.Width
> irisct <- ctree(frml, data=trd)
> table(predict(irisct), trd$Species)
```

	setosa	versicolor	virginica
setosa	40	0	0
versicolor	0	37	3
virginica	0	1	31

سپس بسته `party` را فراخوانی نموده و یک درخت تصمیم می‌سازیم و سپس نتایج پیش‌بینی را بررسی می‌نماییم. تابع `ctree()` برخی پارامترها مانند

MaxDepth، MaxSurrogate، MinBucket، MinSplit را برای کنترل درخت تصمیم استفاده می‌نماید. در ادامه می‌توانیم نگاهی به ساخت یک درخت با نمایش قواعد انجمنی و ترسیم درخت داشته باشیم.

```
> print(irisct)
```

```
Conditional inference tree with 4 terminal nodes
```

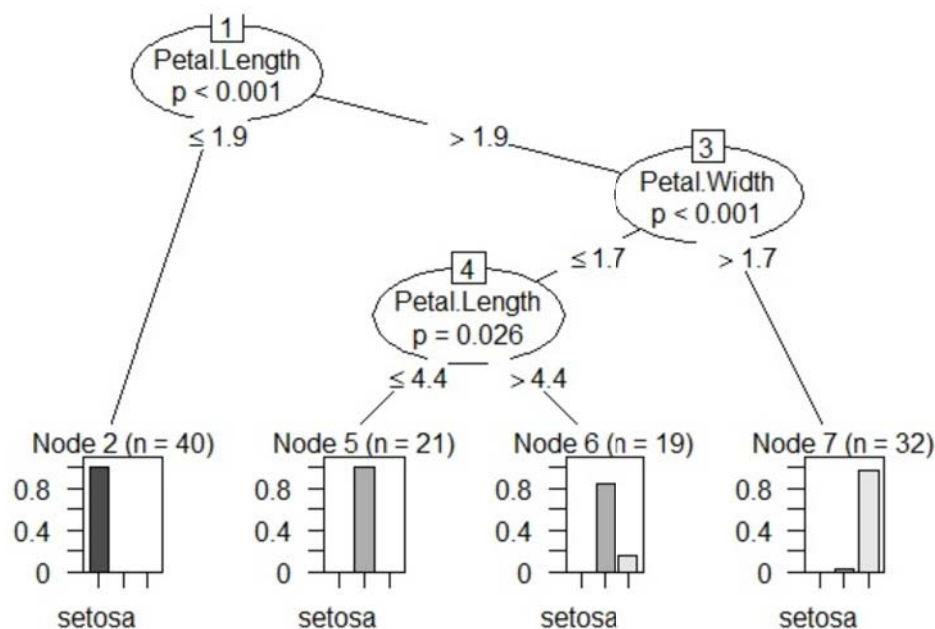
```
Response: Species
```

```
Inputs: Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
```

```
Number of observations: 112
```

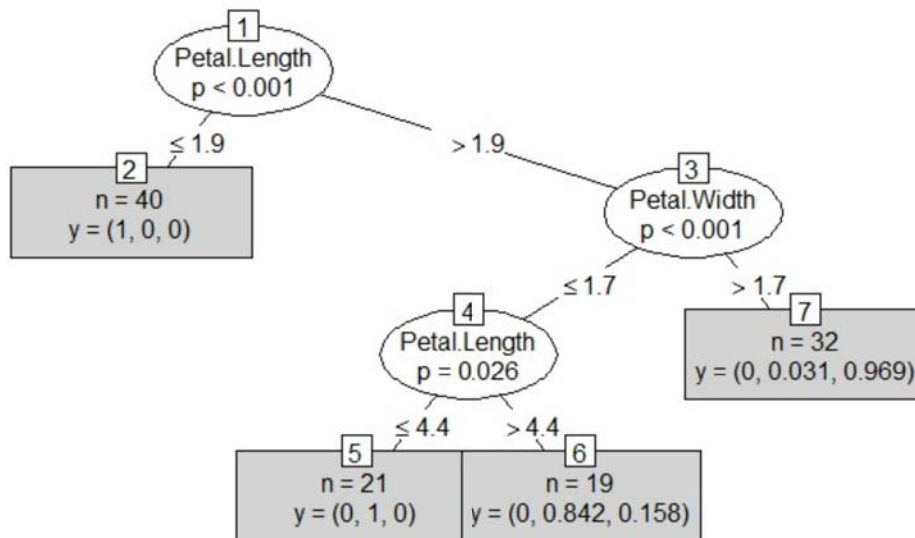
```
1) Petal.Length <= 1.9; criterion = 1, statistic = 104.643
2)* weights = 40
1) Petal.Length > 1.9
3) Petal.Width <= 1.7; criterion = 1, statistic = 48.939
4) Petal.Length <= 4.4; criterion = 0.974, statistic = 7.397
5)* weights = 21
4) Petal.Length > 4.4
6)* weights = 19
3) Petal.Width > 1.7
7)* weights = 32
```

```
plot(irisct)
```



شکل ۴-۵: ترسیم درخت تصمیم در مجموعه داده Iris

```
> plot(irisct, type="simple")
```



شکل ۵-۵: ترسیم درخت تصمیم در مجموعه داده Iris

در شکل بالا نمودار جعبه‌ای برای هر گره برگ نمایش‌دهنده احتمال رخ دادن آن مؤلفه در فضای احتمالی درخت است که در شکل بعدی به صورت علامت y در گره‌های برگ نشان داده شده است. برای مثال گره ۲ با عدد $n=40$ و $y=(1, 0, 0)$ برچسب‌گذاری شده است که به معنای دربرداشتن ۴۰ رخداد است که همه آن‌ها متعلق به کلاس اول یعنی *setosa* می‌باشد. بعد از این مرحله این درخت تصمیم نیازمند داده‌های تست می‌باشد.

```
> tsp <- predict(irisct, newdata = tsd)
> table(tsp, tsd$Species)
```

tsp	setosa	versicolor	virginica
setosa	10	0	0
versicolor	0	12	2
virginica	0	0	14

نسخه فعلی تابع `ctree()` مقادیر مفقوده را نمی‌تواند تحلیل نماید، مواردی که دارای مقادیر مفقوده هستند در حالاتی ممکن است به زیرشاخه چپ و در مواردی به زیرشاخه سمت راست بروند.

جنگل تصادفی^۱

بسته `random-Forest` در ادامه به منظور ساخت یک مدل پیش‌بینی کننده برای مجموعه داده `Iris` مورد استفاده قرار گرفته است. دو محدودیت برای تابع `randomForest()` وجود دارد. اول اینکه این تابع نمی‌تواند داده‌های با مقادیر مفقوده را بررسی نماید و کاربران باید داده‌ها را قبل از دادن به تابع تکمیل نمایند. دوم اینکه حداکثر تعداد سطحی که می‌توان دسته‌بندی کرد ۳۲ سطح می‌باشد. خصوصیت‌هایی با بیش از ۳۲ سطح ابتدا باید تبدیل و بعد مورد استفاده قرار بگیرند. جایگزینی برای استفاده از جنگل تصادفی استفاده از تابع `cforest()` بسته `party` می‌باشد که دارای محدودیت در سطوح نمی‌باشد. در ادامه مجموعه داده `Iris` ابتدا به دو زیرمجموعه داده‌های آموزشی شامل ۷۰ درصد و داده‌های تست شامل ۳۰ درصد تقسیم می‌گردد.

```
> var <- sample(2, nrow(iris), replace=TRUE, prob=c(0.7, 0.3))
> trd <- iris[var==1,]
> tsd <- iris[var==2,]
> library(randomForest)
> x <- randomForest(Species ~ ., data=trd, ntree=100, proximity=TRUE)
> table(predict(x), trd$Species)
```

	setosa	versicolor	virginica
setosa	33	0	0
versicolor	0	32	3
virginica	0	2	35

^۱ Random Forest

```
> print(x)
```

```
Call:
```

```
randomForest(formula = Species ~ ., data = trd, ntree = 100,  
proximity $
```

```
      Type of random forest: classification
```

```
      Number of trees: 100
```

```
No. of variables tried at each split: 2
```

```
      OOB estimate of  error rate: 4.76%
```

```
Confusion matrix:
```

	setosa	versicolor	virginica	class.error
setosa	33	0	0	0.00000000
versicolor	0	32	2	0.05882353
virginica	0	3	35	0.07894737

```
> attributes(x)
```

```
$names
```

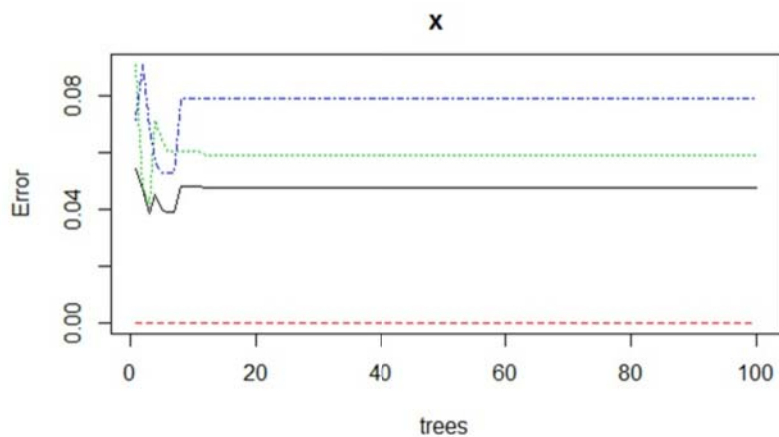
[1]	"call"	"type"	"predicted"	"err.rate"
[5]	"confusion"	"votes"	"oob.times"	"classes"
[9]	"importance"	"importanceSD"	"localImportance"	"proximity"
[13]	"ntree"	"mtry"	"forest"	"y"
[17]	"test"	"inbag"	"terms"	

```
$class
```

```
[1] "randomForest.formula" "randomForest"
```

بعد از این درصد خطا را با تعداد مختلفی از درختان می‌توانیم ترسیم نماییم.

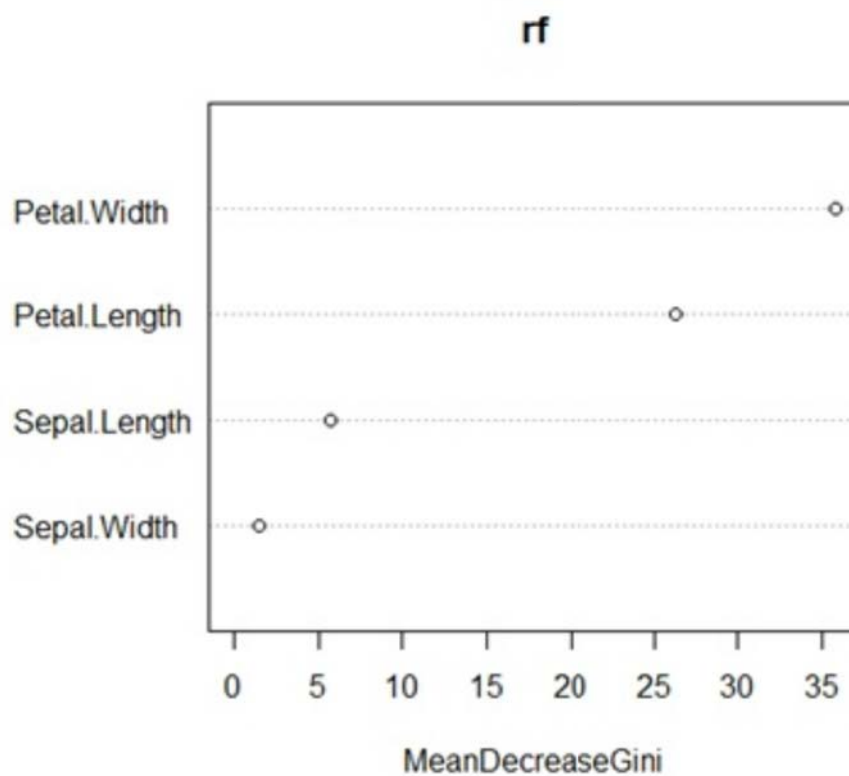
```
> plot(x)
```



شکل ۵-۶: میزان خطا در جنگل تصادفی

اهمیت متغیرها می‌تواند از طریق توابع `importance()` و `varImpPlot()` بدست آید.

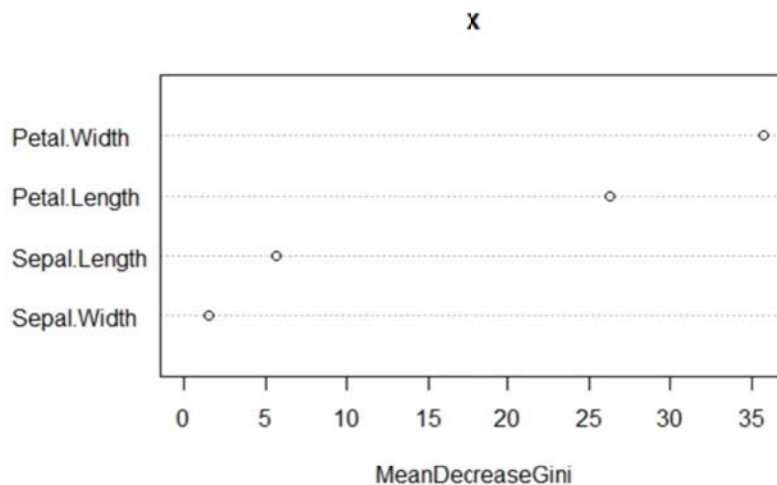
```
> importance(rf)
              MeanDecreaseGini
Sepal.Length      5.672861
Sepal.Width       1.528480
Petal.Length     26.291115
Petal.Width      35.764210
> varImpPlot(rf)
```



شکل ۵-۷: میزان اهمیت متغیرهای مجموعه داده Iris

درنهایت ساخت جنگل تصادفی بر روی داده‌های تست آزمایش گردید و نتایج با استفاده از تابع `table()` و تابع `margin()` بررسی گردیده است. میزان حاشیه برای یک داده عبارت است از نسبت رأی برای هر کلاس منهای نسبت حداکثر رأی برای کلاسهای دیگر است. به عبارت عام، حاشیه مثبت به معنای دسته‌بندی صحیح است.

```
> importance(x)
               MeanDecreaseGini
Sepal.Length      5.672861
Sepal.Width       1.528480
Petal.Length     26.291115
Petal.Width      35.764210
> varImpPlot(x)
```



شکل ۵-۸: جنگل تصادفی و تست مجموعه داده Iris

رگرسیون^۱

رگرسیون به معنای ساختن تابعی از متغیرهای مستقل (که معمولاً با عنوان پیش‌بینی کننده‌ها شناخته می‌شوند) برای پیش‌بینی متغیرهای وابسته است.

رگرسیون خطی

رگرسیون خطی عبارت است از پیش‌بینی پاسخ با یک تابع خطی از پیش‌بینی کننده‌ها. در شکل زیر هرکدام از x_i ها یک پیش‌بینی کننده و y پاسخ این پیش‌بینی کننده‌ها می‌باشد.

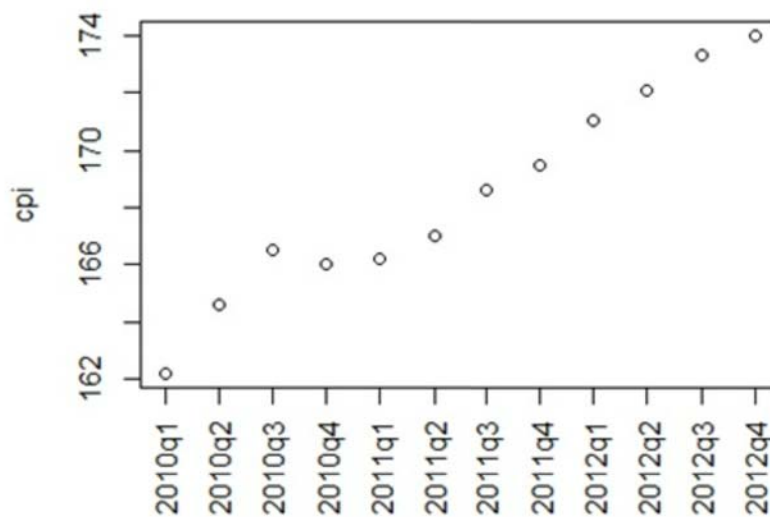
$$y = c_0 + c_1x_1 + c_2x_2 + \dots + c_kx_k$$

در مثال زیر رگرسیون خطی از تابع $lm()$ بر روی داده‌ای از مشتریان که شامل میزان CPI^۲ از سال ۲۰۰۸ الی ۲۰۱۲ می‌باشد استفاده شده است. در ابتدا داده‌ها ایجاد و ترسیم شده است. تابع $axis()$ متن را افقی می‌نماید.

```
> y <- rep(2010:2012, each=4)
> q <- rep(1:4, 3)
> cpi <- c(162.2, 164.6, 166.5, 166.0,
+ 166.2, 167.0, 168.6, 169.5,
+ 171.0, 172.1, 173.3, 174.0)
> plot(cpi, xaxt="n", ylab="cpi", xlab="")
> axis(1, labels=paste(y,q,sep="q"), at=1:12, las=3)
```

^۱ Regression

^۲ Consumer Price Index



شکل ۵-۹: داده‌های مربوط به CPI بین سالهای ۲۰۰۸ تا ۲۰۱۰

سپس همبستگی بین CPI و متغیرهای دیگر همانند سال و بخش را بررسی می‌نماییم.

یک مدل رگرسیون خطی با استفاده از تابع $\text{lm}()$ بر روی داده‌های بالا ساخته می‌شود که از متغیرهای سال و بخش برای پیش‌بینی میزان CPI به‌عنوان پاسخ استفاده می‌شود.

$$\text{CPI} = c_0 + c_1 * \text{year} + c_2 * \text{quarter}$$

با استفاده از مدل خطی بالا، میزان CPI با استفاده از فرمول زیر محاسبه می‌گردد. که c_0 ، c_1 و c_2 ضرایب مدل می‌باشند. تفاوت بین مقادیر مشاهده شده و مقادیر منصوب با استفاده از تابع $\text{residuals}()$ محاسبه می‌گردد.

```
> cor(y, cpi)
[1] 0.9096316
>
> cor(q, cpi)
[1] 0.3738028
>
> f <- lm(cpi ~ y + q)
> f
```

```
Call:
lm(formula = cpi ~ y + q)
```

Coefficients:

(Intercept)	y	q
-7652.262	3.887	1.167

```

> (cpi2011 <- fit$coefficients[[1]]
+ + fit$coefficients[[2]]*2011 + fit$coefficients[[3]]*(1:4))
numeric(0)

> attributes(f)
$names
[1] "coefficients" "residuals" "effects" "rank"
[5] "fitted.values" "assign" "qr" "df.residual"
[9] "xlevels" "call" "terms" "model"

$class
[1] "lm"

> f$coefficients
(Intercept)          y          q
-7652.262500    3.887500    1.166667

> residuals(f)
      1      2      3      4
-0.5791667  0.6541667  1.3875000 -0.2791667
      5      6      7      8
-0.4666667 -0.8333333 -0.4000000 -0.6666667
      9     10     11     12
 0.4458333  0.3791667  0.4125000 -0.0541667

> summary(f)

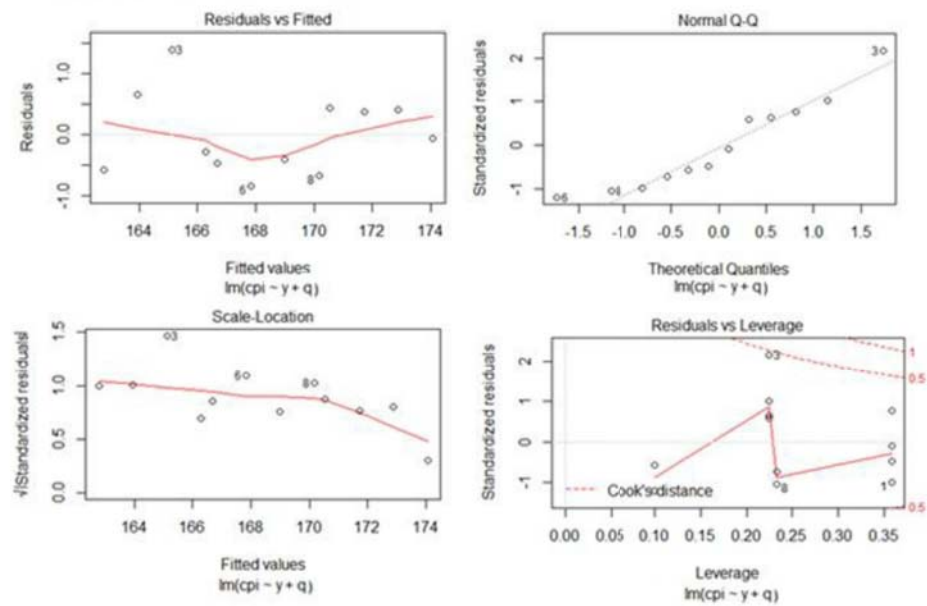
Call:
lm(formula = cpi ~ y + q)

Residuals:
    Min       1Q   Median       3Q      Max
-0.8333 -0.4948 -0.1667  0.4208  1.3875

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -7652.2625    519.1706  -14.739 1.31e-07 ***

```

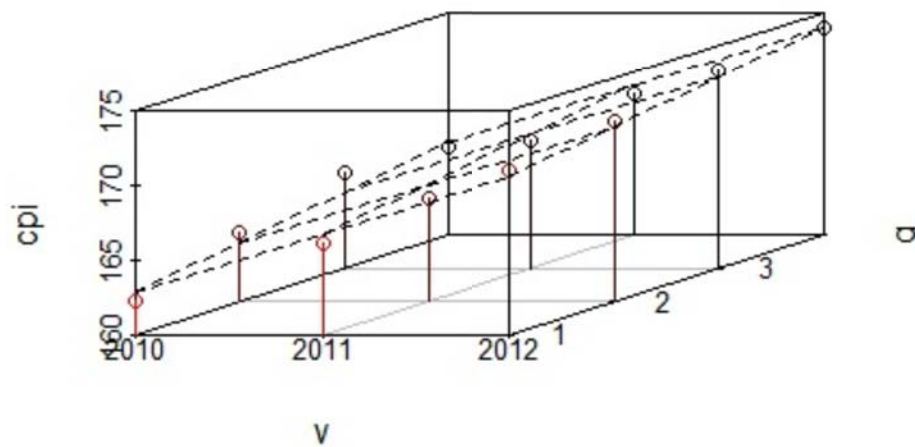
`> plot(f)`



شکل ۵-۱۰: محاسبه میزان CPI با استفاده رگرسیون خطی

همچنین می‌توانیم این مدل را در یک شکل سه‌بعدی ترسیم نماییم. تابع `scatterplot3d()` نمودار سه‌بعدی پراکنده ایجاد نموده و تابع `plane3d()` یک نمودار صفحه‌ای ترسیم می‌نماید.

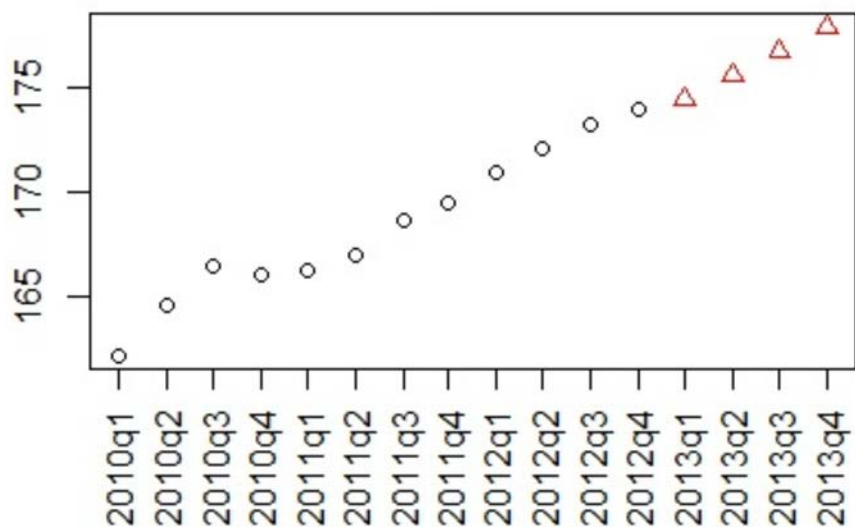
```
> library(scatterplot3d)
> s3d <- scatterplot3d(y, q, cpi,
  highlight.3d=T, type="h", lab=c(2,3))
> s3d$plane3d(f)
```



شکل ۵-۱۱: نمودار سه‌بعدی پراکنده با استفاده از تابع `scatterplot3d()`

با استفاده از این مدل مقدار CPI در سال ۲۰۱۱ می‌تواند پیش‌بینی گردد و مقادیر پیش‌بینی شده در مثلث‌های رنگی در شکل زیر قابل مشاهده می‌باشند.

```
> d13 <- data.frame(y=2013, q=1:4)
> cpi13 <- predict(f, newdata=d13)
> st <- c(rep(1,12), rep(2,4))
> plot(c(cpi, cpi13), xaxt="n", ylab="cpi",
       xlab="", pch=style, col=style)
> axis(1, at=1:16, las=3,
+ labels=c(paste(y,q,sep="q"), "2013q1",
+ "2013q2", "2013q3", "2013q4"))
```



شکل ۵-۱۲: پیش‌بینی مقدار CPI در سال ۲۰۱۱ با استفاده از رگرسیون خطی

قواعد انجمنی

در این بخش مثال‌هایی از کاوش قواعد انجمنی را با استفاده از نرم‌افزار R بررسی می‌نماییم. در ابتدا مفاهیمی از قواعد انجمنی را بررسی نموده و سپس نمایش این قواعد را با استفاده از این نرم‌افزار پی می‌گیریم. سپس با ذکر مثال‌هایی از هرس قواعد اضافی و تفسیر و مصورسازی این قواعد انجمنی می‌پردازیم.

قواعد انجمنی شامل قواعدی ست که نمایش‌دهنده ارتباط و همبستگی بین مشخصه‌هایی می‌باشد. در یک قاعده انجمنی $A \Rightarrow B$ ، A و B هر کدام مشخصه‌های جدا از هم می‌باشند. سه معیار پرکاربرد برای انتخاب قوانین جذاب شامل $support$ ، $confidence$ و $lift$ می‌باشد.

Support: درصدی از موارد که شامل هر دو رویداد A و B باشد.

Confidence: درصد مواردی از رویداد A که شامل رویداد B نیز می‌باشد.

Lift: نسبت $confidence$ به $percentage$ در مواردی که شامل B می‌باشد.

در ادامه شکل فرمولی هر کدام از موارد بالا را مشاهده می‌نمایید:

$$support(A \Rightarrow B) = P(A \cup B)$$

$$P(A) / confidence(A \Rightarrow B) = P(B|A) = P(A \cup B)$$

$$P(A)P(B) / P(B) = P(A \cup B) / lift(A \Rightarrow B) = confidence(A \Rightarrow B)$$

مجموعه داده Titanic

مجموعه داده Titanic در بسته dataset شامل جدولی چهاربندی با اطلاعات خلاصه شده از مسافران شامل طبقه اجتماعی، جنسیت، سن و زنده ماندن می‌باشد. برای تسهیل کاوش قواعد انجمنی داده‌ها را به شکل سطری در متغیری به نام titanic.raw تبدیل نموده‌ایم که هر سطر نشان‌دهنده یک فرد می‌باشد.

```
> str(Titanic)
table [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...
- attr(*, "dimnames")=List of 4
..$ Class : chr [1:4] "1st" "2nd" "3rd" "Crew"
..$ Sex : chr [1:2] "Male" "Female"
..$ Age : chr [1:2] "Child" "Adult"
..$ Survived: chr [1:2] "No" "Yes"
>
> dataframe <- as.data.frame(Titanic)
> head(dataframe)
  Class Sex Age Survived Freq
1  1st Male Child      No    0
2  2nd Male Child      No    0
3  3rd Male Child      No   35
4  Crew Male Child      No    0
5  1st Female Child      No    0
6  2nd Female Child      No    0
> tr <- NULL
> for(i in 1:4) {tr <- cbind(tr, rep(as.character(dataframe[,i]), dataframe$F$
> tr <- as.data.frame(tr)
> names(tr) <- names(dataframe)[1:4]
> dim(tr)
[1] 2201 4
> str(tr)
'data.frame': 2201 obs. of 4 variables:
 $ Class : Factor w/ 4 levels "1st","2nd","3rd",...: 3 3 3 3 3 3 3 3 3 ...
 $ Sex : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 ...
 $ Age : Factor w/ 2 levels "Adult","Child": 2 2 2 2 2 2 2 2 2 ...
 $ Survived: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 ...
```



```
> head(tr)
  Class Sex   Age Survived
1  3rd Male Child       No
2  3rd Male Child       No
3  3rd Male Child       No
4  3rd Male Child       No
5  3rd Male Child       No
6  3rd Male Child       No
>
> summary(tr)
  Class      Sex      Age      Survived
1st :325   Female: 470   Adult:2092   No :1490
2nd :285   Male  :1731   Child: 109   Yes: 711
3rd :706
Crew:885
```

حالا ما مجموعه داده‌ای داریم که هر سطر از آن نشان‌دهنده یک فرد می‌باشد و برای کاوش قواعد انجمنی مناسب می‌باشد.

کاوش قواعد انجمنی^۱

یک الگوریتم کلاسیک برای کاوش قواعد انجمنی الگوریتم APRIORI می‌باشد. این الگوریتم تعاملات را برای موارد تکراری محاسبه کرده و سپس قواعد انجمنی را از آن‌ها استخراج می‌کند. تابع `apriori()` در بسته `arules` در همین مورد می‌باشد. الگوریتم دیگری برای کاوش قواعد انجمنی الگوریتم ECLAT می‌باشد که موارد با دسته هم‌ارز را پیدا نموده و بخش‌بندی می‌نماید. این الگوریتم با استفاده از تابع `eclat()` در همان بسته قابل دسترس می‌باشد.

در مثال زیر کاوش قواعد انجمنی را با استفاده از تابع `apriori()` بررسی می‌نماییم. تنظیمات اولیه عبارت است از:

۱) `supp=۰.۱`, ۲) `conf=۰.۸`, and ۳) `maxlen=۱۰`.

^۱ Association rule mining

```
> library(arules)
> ra <- apriori(tr)
```

Parameter specification:

```
confidence minval smax arem aval originalSupport support minlen maxlen
0.8 0.1 1 none FALSE TRUE 0.1 1 10
target ext
rules FALSE
```

Algorithmic control:

```
filter tree heap memopt load sort verbose
0.1 TRUE TRUE FALSE TRUE 2 TRUE
```

```
apriori - find association rules with the apriori algorithm
version 4.21 (2004.05.09) (c) 1996-2004 Christian Borgelt
set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[10 item(s), 2201 transaction(s)] done [0.00s].
sorting and recoding items ... [9 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 done [0.00s].
writing ... [27 rule(s)] done [0.00s].
creating S4 object ... done [0.00s].
```

در بخش زیر قواعد انجمنی استخراج شده را مشاهده می‌فرمایید.

```
> inspect(rules.all)
```

	lhs	rhs	support	confidence	lift
1	{}	=> {Age=Adult}	0.9504771	0.9504771	1.0000000
2	{Class=2nd}	=> {Age=Adult}	0.1185825	0.9157895	0.9635051
3	{Class=1st}	=> {Age=Adult}	0.1449341	0.9815385	1.0326798
4	{Sex=Female}	=> {Age=Adult}	0.1930940	0.9042553	0.9513700
5	{Class=3rd}	=> {Age=Adult}	0.2848705	0.8881020	0.9343750
6	{Survived=Yes}	=> {Age=Adult}	0.2971377	0.9198312	0.9677574
7	{Class=Crew}	=> {Sex=Male}	0.3916402	0.9740113	1.2384742
8	{Class=Crew}	=> {Age=Adult}	0.4020900	1.0000000	1.0521033
9	{Survived=No}	=> {Sex=Male}	0.6197183	0.9154362	1.1639949
10	{Survived=No}	=> {Age=Adult}	0.6533394	0.9651007	1.0153856
11	{Sex=Male}	=> {Age=Adult}	0.7573830	0.9630272	1.0132040
12	{Sex=Female, Survived=Yes}	=> {Age=Adult}	0.1435711	0.9186047	0.9664669
13	{Class=3rd, Sex=Male}	=> {Survived=No}	0.1917310	0.8274510	1.2222950
14	{Class=3rd, Survived=No}	=> {Age=Adult}	0.2162653	0.9015152	0.9484870
15	{Class=3rd, Sex=Male}	=> {Age=Adult}	0.2099046	0.9058824	0.9530818
16	{Sex=Male, Survived=Yes}	=> {Age=Adult}	0.1535666	0.9209809	0.9689670

```

17 {Class=Crew,
    Survived=No} => {Sex=Male}    0.3044071  0.9955423  1.2658514
18 {Class=Crew,
    Survived=No} => {Age=Adult}    0.3057701  1.0000000  1.0521033
19 {Class=Crew,
    Sex=Male}    => {Age=Adult}    0.3916402  1.0000000  1.0521033
20 {Class=Crew,
    Age=Adult}   => {Sex=Male}    0.3916402  0.9740113  1.2384742
21 {Sex=Male,
    Survived=No} => {Age=Adult}    0.6038164  0.9743402  1.0251065
22 {Age=Adult,
    Survived=No} => {Sex=Male}    0.6038164  0.9242003  1.1751385
23 {Class=3rd,
    Sex=Male,
    Survived=No} => {Age=Adult}    0.1758292  0.9170616  0.9648435
24 {Class=3rd,
    Age=Adult,
    Survived=No} => {Sex=Male}    0.1758292  0.8130252  1.0337773
25 {Class=3rd,
    Sex=Male,
    Age=Adult}   => {Survived=No} 0.1758292  0.8376623  1.2373791
26 {Class=Crew,
    Sex=Male,
    Survived=No} => {Age=Adult}    0.3044071  1.0000000  1.0521033
27 {Class=Crew,
    Age=Adult,
    Survived=No} => {Sex=Male}    0.3044071  0.9955423  1.2658514

```

به عنوان یک پدیده طبیعی برای کاوش قواعد انجمنی ، بسیاری از قواعد ایجاد شده غیردلخواه هستند. ما به دنبال قواعدی هستیم که در سمت راست زنده ماندن یا زنده نماندن را نمایش دهد. بقیه موارد که دلخواه ما نیستند را حذف می‌نماییم. با قرار دادن متغیر `minlen` به حداقل ۲ قواعدی که دارای بخش خالی هستند حذف می‌گردند.

```

> r <- apriori(tr, control = list(verbose=F),
+ parameter = list(minlen=2, supp=0.005, conf=0.8),
+ appearance = list(rhs=c("Survived=No", "Survived=Yes")
+ , default="lhs"))
> quality(r) <- round(quality(r), digits=3)
> rs <- sort(r, by="lift")
> inspect(rs)

```

	lhs	rhs	support	confidence	lift
1	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.011	1.000	3.096
2	{Class=2nd, Sex=Female, Age=Child}	=> {Survived=Yes}	0.006	1.000	3.096
3	{Class=1st, Sex=Female}	=> {Survived=Yes}	0.064	0.972	3.010
4	{Class=1st, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.064	0.972	3.010
5	{Class=2nd, Sex=Female}	=> {Survived=Yes}	0.042	0.877	2.716
6	{Class=Crew, Sex=Female}	=> {Survived=Yes}	0.009	0.870	2.692
7	{Class=Crew, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.009	0.870	2.692
8	{Class=2nd, Sex=Female, Age=Adult}	=> {Survived=Yes}	0.036	0.860	2.663
9	{Class=2nd, Sex=Male, Age=Adult}	=> {Survived=No}	0.070	0.917	1.354
10	{Class=2nd, Sex=Male}	=> {Survived=No}	0.070	0.860	1.271
11	{Class=3rd, Sex=Male, Age=Adult}	=> {Survived=No}	0.176	0.838	1.237
12	{Class=3rd,				

در کد بالا حداقل میزان support عدد ۰,۰۰۵ می‌باشد که هر قاعده به‌وسیله حداقل ۱۲ مورد پشتیبانی می‌گردد که برای جمعیت ۲۲۰۱ مرد قابل‌پذیرش می‌باشد. علاوه بر معیارهایی همچون Support، confidence و lift معیارهای دیگری همچون chi-square، conviction، gini و leverage در این تحلیل‌ها قابل‌استفاده می‌باشند. برای دسترسی به معیارهای دیگر میتوان از تابع interestMeasure() در بسته arules استفاده نمود.

حذف افزونگی^۱

برخی قواعد که تولید شده اطلاعات اضافی به ما نمی‌دهد وقتی قواعد دیگری وجود دارد که همان نتایج را می‌رساند. برای مثال قاعده ۲ اطلاعات اضافه‌ای به ما نمی‌دهد وقتی قاعده ۱ موجود می‌باشد. به عبارت دیگر وقتی یک قاعده یک قاعده مافوق بر قاعده دیگر می‌باشد، قاعده پایین‌تر به عنوان یک قاعده اضافی و دارای افزونگی در نظر گرفته می‌شود. در شکل بالا قواعد ۴، ۷ و ۸ نیز از جمله قواعد اضافی می‌باشند که قابل مقایسه با به ترتیب قواعد ۳، ۶ و ۵ هستند. در ادامه این قواعد اضافی را هرس می‌نماییم. توجه داشته باشید که قواعد به وسیله متغیر lift قبلاً نزولی شده‌اند.

```
> subset.matrix <- is.subset(rs, rs)
> subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA
> rdnt <- colSums(subset.matrix, na.rm=T) >= 1
> which(rdnt)
[1] 2 4 7 8
> rp <- rs[!rdnt]
> inspect(rp)
```

	lhs	rhs	support	confidence	lift
1	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.011	1.000	3.096
2	{Class=1st, Sex=Female}	=> {Survived=Yes}	0.064	0.972	3.010
3	{Class=2nd, Sex=Female}	=> {Survived=Yes}	0.042	0.877	2.716
4	{Class=Crew, Sex=Female}	=> {Survived=Yes}	0.009	0.870	2.692
5	{Class=2nd, Sex=Male, Age=Adult}	=> {Survived=No}	0.070	0.917	1.354
6	{Class=2nd, Sex=Male}	=> {Survived=No}	0.070	0.860	1.271
7	{Class=3rd, Sex=Male, Age=Adult}	=> {Survived=No}	0.176	0.838	1.237
8	{Class=3rd, Sex=Male}	=> {Survived=No}	0.192	0.827	1.222

^۱ Redundancy

در کد بالا تابع `is.subset(r1, r2)` بررسی می‌نماید که آیا ۲۱ زیرمجموعه‌ای از ۲۲ می‌باشد یا خیر؟ (و بالعکس). تابع `lower.tri()` یک ماتریس منطقی با مقادیر صحیح پایین مثلثی را برمی‌گرداند. در نتایج بالا مشاهده می‌شود که قواعد ۲ و ۴ و ۷ و ۸ که در بخش قبل دارای افزونگی بودند با موفقیت حذف گردیده‌اند.

تفسیر قواعد

در شکل بالا مشاهده می‌شود که در قاعده نخست کسانی که در کلاس ۲ و با سن کودک هستند با درجه اطمینان بالایی زنده مانده‌اند درحالی که قاعده مربوط به زنده ماندن کودکان در کلاسهای دیگر اصلاً وجود ندارد. برای حل این مسئله قواعدی را پیدا می‌نماییم که در سمت راست زنده ماندن مثبت است (`Survived=Yes`) و در سمت چپ شامل کلاس ۱ و کلاس ۲ و کلاس ۳ و سن کودک و سن بزرگسال باشد. می‌توانیم از عدد آستانه کمتری برای `support` و `confidence` استفاده کنیم تا قواعد بیشتری یافت گردد.

```
> r <- apriori(tr, parameter = list(minlen=3, supp=0.002, conf=0.2),
+ appearance = list(rhs=c("Survived=Yes"), lhs=c("Class=1st", "Class=2nd"
+ , "Class=3rd", "Age=Child", "Age=Adult"), default="none"),
+ control = list(verbose=F))
> rs <- sort(r, by="confidence")
> inspect(rs)
```

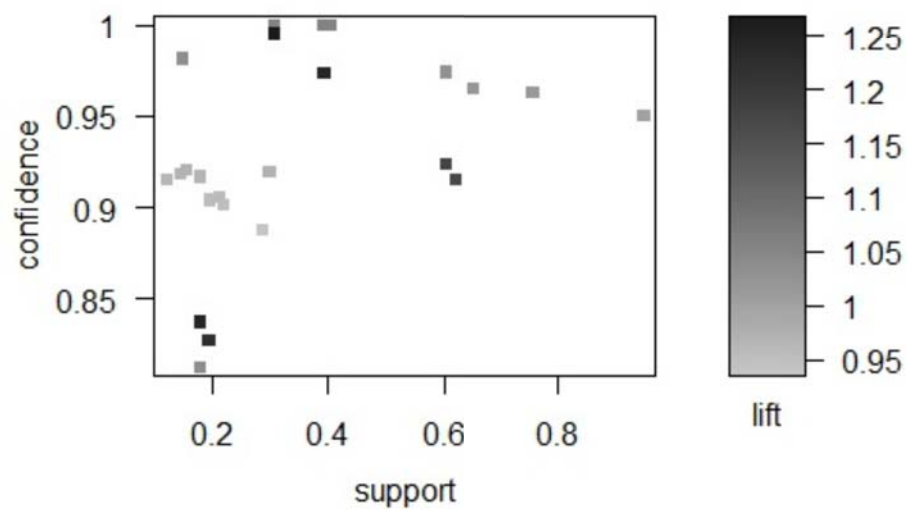
	lhs	rhs	support	confidence	lift
1	{Class=2nd, Age=Child}	=> {Survived=Yes}	0.010904134	1.0000000	3.0956399
2	{Class=1st, Age=Child}	=> {Survived=Yes}	0.002726034	1.0000000	3.0956399
3	{Class=1st, Age=Adult}	=> {Survived=Yes}	0.089504771	0.6175549	1.9117275
4	{Class=2nd, Age=Adult}	=> {Survived=Yes}	0.042707860	0.3601533	1.1149048
5	{Class=3rd, Age=Child}	=> {Survived=Yes}	0.012267151	0.3417722	1.0580035
6	{Class=3rd, Age=Adult}	=> {Survived=Yes}	0.068605179	0.2408293	0.7455209

مصورسازی قواعد انجمنی

در ادامه برخی روش‌ها برای مصورسازی قواعد انجمنی از جمله نمودار پراکندگی ، نمودار بالونی، گراف و نمودار مختصات موازی را می‌توان نام برد.

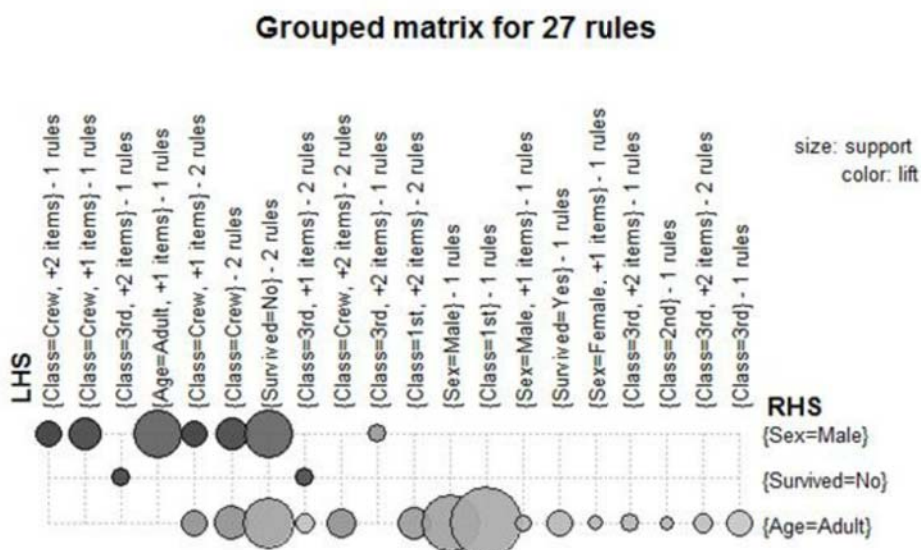
```
> library(arulesViz)
> plot(ra)
```

Scatter plot for 27 rules



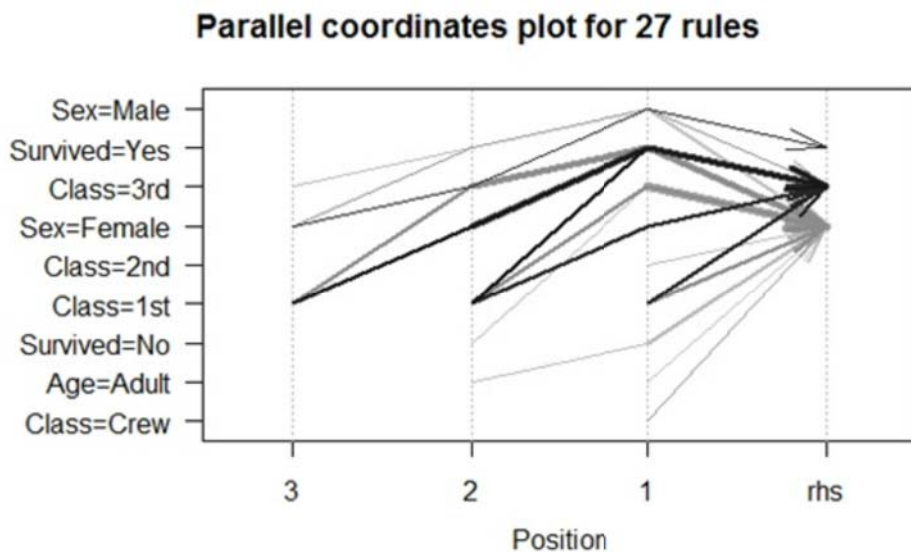
شکل ۵-۱۳ : نمودار پراکندگی قواعد انجمنی برای مجموعه داده titanic

```
> plot(ra, method="grouped")
```



شکل ۵-۱۴: نمودار ماتریس گروهی قواعد انجمنی برای مجموعه داده titanic

```
> plot(ra, method="paracoord", control=list(reorder=TRUE))
```



شکل ۵-۱۵: نمودار مختصات موازی قواعد انجمنی برای مجموعه داده titanic

فصل ششم

سری های زمانی

۶-۱- تعریف سری زمانی^۱

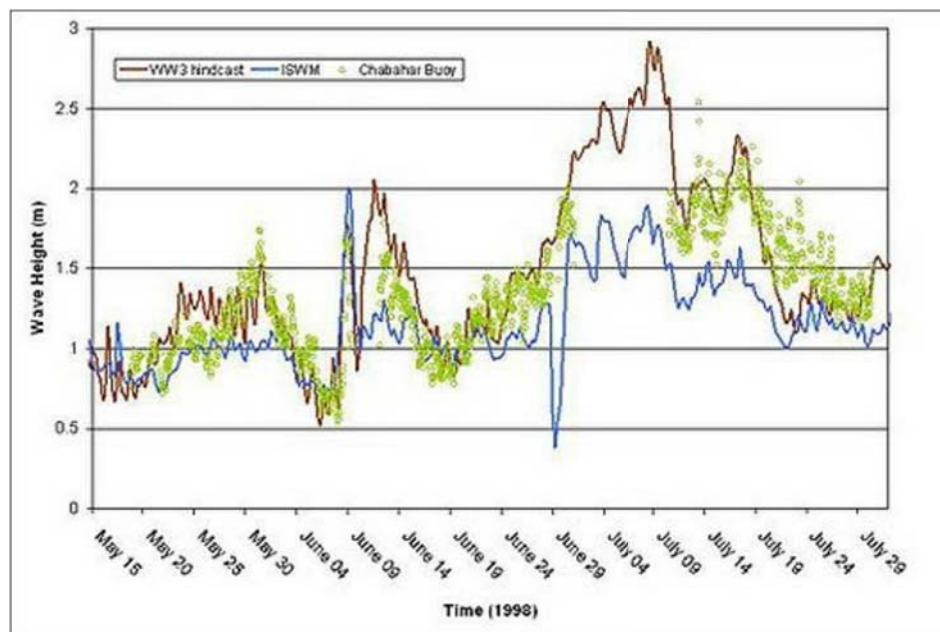
در هر علم، به آمار جمع‌آوری شده مربوط به متغیری که قرار است پیش‌بینی شود و در دوره‌های زمانی گذشته موجود است، اصطلاحاً سری زمانی می‌گویند. منظور از یک سری زمانی مجموعه‌ای از داده‌های آماری است که در فواصل زمانی مساوی و منظمی جمع‌آوری شده باشند. روش‌های آماری که این گونه داده‌های آماری را مورد استفاده قرار می‌دهد روش‌های تحلیل سری‌های زمانی نامیده می‌شود. مانند فروش فصلی یک شرکت طی سه سال گذشته. یک سری زمانی مجموعه مشاهداتی است که بر اساس زمان مرتب شده باشند. مثال‌های آن از اقتصاد و حتی رشته‌های مهندسی دیده می‌شود. بخصوص روش‌های تجزیه و تحلیل سری‌های زمانی قسمت مهمی از آمار را تشکیل می‌دهد. به عنوان مثال می‌توان به سری‌های زمانی زیر اشاره نمود:

- سری زمانی در اقتصاد، مانند قیمت سهام در روزهای متوالی، صادرات در ماه‌های متوالی، متوسط درآمد در ماه‌های متوالی...
- سری زمانی فیزیک، به‌ویژه در علوم مربوط به آثار جوی، علوم دریایی، فیزیک زمین (ژئوفیزیک)
- سری‌های زمانی بازاریابی، تجزیه و تحلیل ارقام فروش در هفته یا ماه‌ها متوالی یک مسئله مهم در تجارت است.
- سری‌های زمانی جمعیت‌نگاری، اندازه‌گیری سالانه جمعیت با هدف پیش‌بینی تغییرات جمعیت در مدت زمان ده تا بیست سال آینده.
- فرایندهای دوتایی، سری‌هایی که مشاهدات یکی از دو مقدار که معمولاً با ۰ و ۱ نشان می‌دهند را اختیار کند، که بخصوص در نظریه ارتباطات اتفاق می‌افتد را فرایند دوتایی می‌نامند.
- فرایندهای نقطه‌ای، نوعی سری زمانی که پیشامدهای رخ داده به طور تصادفی در زمان رخ داده، زمان‌های رخ دادن تصادفات قطارها.

^۱ Time Series

سری پیوسته، سری هایی که مشاهدات به طور پیوسته در زمان ایجاد می شوند (حتی اگر مقادیر گسسته ای اختیار کنند)

سری گسسته، سری که مشاهدات در زمان های معین و معمولاً در فاصله های مساوی رخ می دهند.



شکل ۶-۱: نمونه ای از سری زمانی

سری های زمانی یکی از شاخه های آمار و احتمال است که در سایر رشته های علوم مانند ژئوفیزیک، اقتصاد، مهندسی ارتباطات، هواشناسی و ... کاربرد فراوانی دارد؛ دامنه کاربردهای سری های زمانی روز به روز گسترده تر می شود و نیاز دانش پژوهان در این زمینه افزون تر می گردد. درواقع یک سری زمانی مجموعه مشاهداتی است که برحسب زمان مرتب شده باشند.

داده‌هایی که از مشاهدات یک پدیده در طول زمان بدست می‌آیند بسیار متداول هستند، در کسب‌وکار و اقتصاد، در هواشناسی، در کشاورزی، در علوم بیولوژیکی فهرست زمینه‌هایی که در آن سری زمانی مشاهده شده و تجزیه و تحلیل می‌شود بی‌پایان است. هدف تجزیه و تحلیل سری‌های زمانی معمولاً دو مورد می‌باشد:

- درک یا به مدل درآوردن مکانیسم تصادفی که منجر به مشاهده، سری می‌شود.

- پیش‌بینی مقادیر آینده سری، بر مبنای گذشته آن

در تجزیه و تحلیل یک سری زمانی چندین هدف ممکن وجود دارد. این اهداف را می‌توانیم به صورت توصیف، تشریح، پیش‌بینی و کنترل رده‌بندی کنیم. هر چند توصیف رفتار یک سری زمانی از لحاظ تغییرات موضعی و درازمدت در آن یا مطالعه وابستگی‌های موجود بین عناصر سری از بررسی‌های متداولی است که روی سری‌های زمانی انجام می‌شود اما می‌توان گفت مهم‌ترین هدف از تحلیل سری زمانی پیش‌بینی مقادیر آینده آن است. برای یک تحلیل سری زمانی و پیش‌بینی آینده آن چه باید کرد؟ بدیهی است لازمه اتخاذ هر تصمیمی در این مورد آشنایی با رفتار سری به عنوان تابعی از زمان است. ساده‌ترین راه برای این منظور رسم نمودار سری زمانی است. پیدا کردن الگوهای مناسب برای سری‌های زمانی کاری است مهم؛ یک استراتژی چندمرحله‌ای را برای ساختن یک الگو توسعه می‌دهیم. در این روش سه مرحله عمده وجود دارد که از هر یک از آن‌ها ممکن است چندین بار استفاده کنیم ۱- تشخیص یا شناسایی الگو ۲- برازش الگو ۳- تشخیص درستی الگو در یک تحلیل سری زمانی اولین مرحله رسم نمودار داده‌هاست. با امتحان و بررسی دقیق نمودار سری زمانی می‌توانیم ایده‌ی خوبی در مورد این که روند، نوسانات فصلی، نقاط پرت و واریانس غیرثابت و ... وجود دارند یا خیر، به دست آوریم. (نیرومند، بزرگ‌نیا، ۱۳۷۲)

یکی از خاصیت‌ها در سری‌های زمانی خاصیت روش میانگین متحرک می‌باشد. خاصیت روش میانگین متحرک این است که تغییرات موجود در یک مجموعه را کاهش می‌دهد. در سری‌های زمانی از این خاصیت برای حذف نوسانات غیرضروری استفاده می‌شود.

عیب روش میانگین متحرک حذف شدن بعضی از مشاهدات از ابتدا و انتهای سری زمانی است. یک عیب دیگر این است که ممکن است باعث تغییرات دوره‌ای یا سایر تغییرات شود که در داده‌های اولیه وجود نداشته‌اند. عیب سوم میانگین متحرک این است که به شدت تحت تأثیر ماکسیمم و مینیمم مشاهدات قرار دارد. برای رفع این عیب از میانگین متحرک موزون می‌توان استفاده کرد. در این حالت به مشاهدات مرکزی بیشترین وزن و به مشاهدات انتهایی کمترین وزن را می‌دهند. (نیرومند، بزرگ‌نیا، ۱۳۷۲)

۶-۲- سری‌های زمانی در داده‌کاوی

یک سری زمانی دنباله‌ای از مشاهدات بر روی یک متغیر مورد توجه است که در نقاط گسسته‌ای از زمان که معمولاً فاصله‌های مساوی دارند (روزانه - هفتگی - ماهانه - فصلی - سالانه) رخ می‌دهد. تجزیه تحلیل سری‌های زمانی، متضمن توصیف فرآیند یا پدیده‌ای است که تولید دنبال می‌کند. جهت پیش‌بینی سری‌های زمانی، لازم است که رفتار فرآیند را با یک مدل ریاضی که قابل تعمیم به آینده باشد، توصیف کرد. معمولاً لازم نیست مدل نماینده مشاهدات خیلی قدیمی یا فراتر از زمان مورد انتظار پیش‌بینی باشد. مهم‌ترین نکته در داده‌های سری زمانی، آن است که این داده‌ها دارای همبستگی هستند. از ضریب همبستگی برای تعیین همبستگی بین مقادیر X و Y استفاده می‌شود، اما وقتی خود متغیرهای مستقل، مقادیرشان به هم مرتبط باشد، به آن خودهمبستگی گویند. با توجه به تعریف داده‌های سری زمانی، روش‌های آماری مبتنی بر فرض مستقل بودن مشاهدات، مناسب نبوده به جای آن می‌توان از معادلات خودهمبستگی در تحلیل سری‌های زمانی استفاده کرد.

یک سری زمانی ساده‌ترین شکل داده‌های زمانی است. سری زمانی دنباله‌ای از اعداد حقیقی است که به صورت منظم در طول زمان گردآوری شده است. هر عدد، نشان‌دهنده مقدار یک متغیر مشاهده شده می‌باشد. همان طور که اشاره شد، داده‌های سری زمانی در حوزه‌های مختلفی مثل تحلیل بازار سهام، علوم ارتباطات، پزشکی، داده‌های مالی و غیره مطرح می‌شوند. همچنین داده‌های وب که میزان

استفاده از وب سایتهای مختلف را ثبت می کنند (برای مقاله تعداد کلیک ها) را می توان با سری های زمانی مدل کرد. در حقیقت، سری های زمانی برای نمایش بخش بزرگی از داده های ذخیره شده در بانک های اطلاعاتی تجاری به کار می رود که به تدریج به عنوان یک نوع داده متفاوت، اهمیت بیشتری یافته است. اهمیت داده های سری های زمانی موجب تحقیقات زیادی در زمینه تحلیل این نوع داده ها شده است. ادبیات آماری در مورد سری های زمانی بسیار وسیع است و به طور عمده به مسائلی مانند شناسایی الگوها و تحلیل روند (مانند رشد خطی فروش شرکت در طول یک سال)، تحلیل های فصلی (مثلاً فروش زمستانی یک محصول تقریباً دو برابر فروش تابستانی است) و پیش بینی (مانند پیش بینی فروش فصل آینده) می پردازند. این موضوعات کلاسیک در تعدادی از مراجع آماری بررسی شده است. (Ye, 2003) در سری زمانی دو مبحث پیش بینی نقطه آتی با سری زمانی و نیز داده کاوی روی خودسری های زمانی وجود دارد که شامل دسته بندی و خوشه بندی سری های زمانی می باشد. به عنوان مثال در زمینه بورس در مورد پیش بینی با سری زمانی می توان گفت با توجه به روند قیمت روزهای گذشته یک سهم، قیمت محتمل آن سهم در روز آتی چقدر خواهد بود. ولی در مورد داده کاوی سری زمانی برای مثال گروه های سهامی که باهم نوسان می نمایند را می توانیم پیدا کنیم. هدف اصلی داده کاوی سری های زمانی کشف الگوهای موجود در وقایع و داده های یک سری زمانی است. کشف این الگوهای ناشناخته، همگام با استفاده از دیگر روش های مختلف داده کاوی مانند سیستم های پایگاه داده، آمار، یادگیری ماشین، شبکه های عصبی، تئوری مجموعه ها، منطق فازی و غیره در داده کاوی اتفاق می افتد. از مهم ترین کاربردهای داده کاوی سری های زمانی، می توان به دسته بندی، خوشه بندی و کشف قواعد از داده ها اشاره کرد. در داده کاوی سری های زمانی دو سؤال اساسی زیر مطرح می شود.

- چگونه می توان روابط همبستگی در درون سری های زمانی را پیدا کرد؟
- چگونه می توان سری های زمانی با حجم انبوهی از داده ها را تحلیل کرده و الگوهای منظم، روند تغییرات تصادفی، داده های مغشوش و غیره را از آنها استخراج کرد؟ (Han, Kamber, 2006)

سری های زمانی را به طور کلی در دو قسمت تقسیم بندی می کنند. بخش اول پیش بینی نقطه بعدی با استفاده از نقاط قبلی که به تحلیل سری های زمانی معروف است و بخش دوم انجام داده کاوی روی خود سری های زمانی که خود هر سری زمانی به عنوان یک مشاهده در نظر گرفته می شود که شامل خوشه بندی و دسته بندی سری های زمانی می باشد.

۶-۳- اجزای سری های زمانی

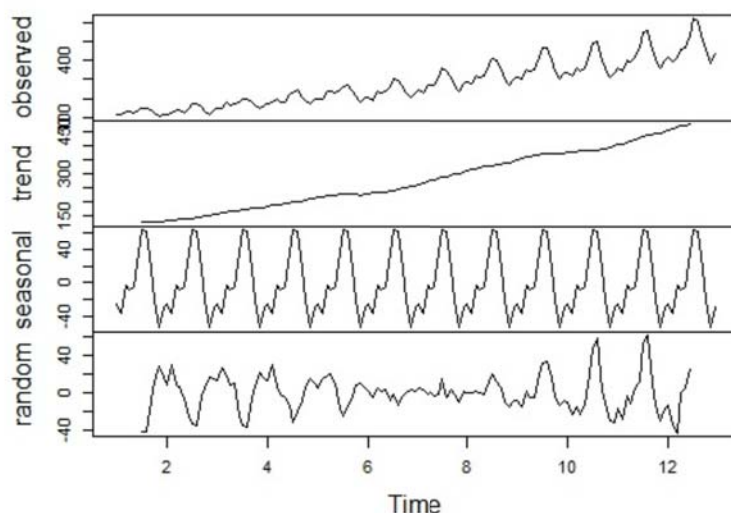
یک سری زمانی شامل یک متغیر وابسته (Y) می باشد که تابعی از زمان است. چنین تابعی به شکل یک نمودار سری زمانی نمایش داده می شود. در مطالعه داده های سری های زمانی ، همواره دو هدف اساسی زیر دنبال می شود:

(Han, Kamber, ۲۰۰۶)

- مدل سازی سری های زمانی با تأکید بر فرآیند ایجاد سری های زمانی
 - پیش بینی سری های زمانی با تأکید بر پیش بینی مقادیر متغیرهای سری زمانی
- تجزیه و تحلیل روند شامل شناسایی چهار جز یا مشخصه اساسی هر سری زمانی می باشد. (نیرومند، بزرگ نیا، ۱۳۷۲)

- **روند خطی و غیر خطی** : تغییر درازمدت در میانگین یا به عبارت دیگر حرکت درازمدت تدریجی افزایشی یا کاهشی داده ها در طول زمان
- **تغییر سیکلی** : تغییرات دوره ای موجود در داده های یک سری زمانی است. معمولاً این نوع افزایش ها و کاهش های لحظه ای در داده ها ، در دوره های بیشتر از یک سال اتفاق می افتد.
- **تغییرات فصلی** : تغییراتی است که به صورت فصلی در داده های سری زمانی اتفاق می افتد. این نوع الگوی تغییر در طول یک سال مشاهده می شود.

- تغییرات باقیمانده‌ها یا تصادفی: اگر سه جز قبلی از یک سری زمانی حذف شود، سری باقیمانده حاصل می‌شود که ممکن است تصادفی باشد. (Han, Kamber, ۲۰۰۶)



شکل ۶-۱: انواع سری زمانی

یکی از مهم‌ترین جنبه‌های کاربرد، سری‌های زمانی پیش‌بینی می‌باشد پیش‌بینی سری‌های زمانی با استفاده از یک معادله ریاضی، الگویی تاریخی در داده‌های سری زمانی ایجاد می‌کند. این روش برای پیش‌بینی کوتاه‌مدت یا بلندمدت مقادیر آینده مورد استفاده قرار می‌گیرد. روش‌های مختلفی برای پیش‌بینی سری‌های زمانی، مورد استفاده است که از بین آن‌ها روش میانگین متحرک تلفیق شده با اتورگرسیون که به مدل «باکس- جنکینز» نیز موسوم است از اهمیت ویژه‌ای برخوردار است. (Han, Kamber, ۲۰۰۶)

معمولاً مطلوب است که سیستم پیش‌بینی بتواند تغییرات پایدار را مشخص و با تعدیل مدل پیش‌بینی، فرآیند جدید را تعقیب کند. درعین حال سیستم پیش‌بینی، تغییرات تصادفی و موقتی را تشخیص داده و در مقابل واکنش نشان ندهد. در هنگام کار با مدل‌های پیش‌بینی با توجه به مراحل مختلف سیکل پیش‌بینی (مثلاً عمر محصول) لازم است که مدل‌های پیش‌بینی مختلفی به کار

گرفته شوند. مثلاً ممکن است گاهی اوقات فقط روند را حفظ کرده و بقیه علل تغییرات سری زمانی را حذف کنیم. (ابریشمی، ۱۳۷۴)

به طور کلی یک سری زمانی را ایستا گویند، هرگاه تغییر منظمی در میانگین و واریانس آن وجود نداشته و تغییرات دوره‌ای اکید حذف شده باشد. نظریه احتمال سری های زمانی بیشتر با سری های زمانی ایستا سروکار دارد و به این دلیل است که در تجزیه و تحلیل سری های زمانی، برای استفاده از نظریه ایستایی لازم است که سری نایستا را به ایستا تبدیل کنیم. مثلاً می توانیم روند و تغییرات فصلی را از مجموعه داده ها حذف کرده و سپس به وسیله یک فرآیند تصادفی ایستا، تغییر در باقی مانده ها را الگوسازی کنیم. (نیرومند، بزرگنیا، ۱۳۷۲)

۶-۴- شناسایی ، تجزیه و حذف اجزای سری های زمانی

برای تجزیه و تحلیل مجموعه ای داده ها ، در اولین مرحله لازم است که نمودار مشاهدات را نسبت به زمان رسم کنیم. این کار غالباً مهم ترین خواص یک سری زمانی مانند روند، فصلی بودن و مشاهدات دورافتاده را آشکار می کند.

روش های متعددی برای شناسایی ، تعیین و یا حذف برخی از اجزای سری های زمانی وجود دارد. با توجه به اینکه مهم ترین اجزای یک سری زمانی جزو روند و فصلی می باشد، در ادامه به بررسی روش های شناسایی، هموارسازی، تجزیه و یا حذف این اجزا پرداخته می شود. (نیرومند، بزرگنیا، ۱۳۷۲)

۶-۵- سری زمانی در نرم افزار R

داده سری زمانی در R

در این بخش به مثال هایی از سری زمانی که شامل تجزیه، پیش بینی، خوشه بندی و دسته بندی می باشد اشاره خواهیم نمود. در بخش اول به صورت مختصر داده های سری های زمانی در R را بیان می کنیم. در بخش دوم مثال هایی از تجزیه سری های زمانی به قطعات روندی، فصلی و تصادفی را نمایش می دهیم. بخش سوم نمایش می دهد که چگونه یک مدل متحرک میانگین یکپارچه در R می توان ساخت و از آن به عنوان پیش بینی آینده استفاده نمود. در بخش چهارم زمان های پویای تابیده و خوشه بندی سلسله مراتبی سری های زمانی با استفاده از فاصله اقلیدسی و فاصله DTW ارائه خواهیم نمود. در بخش پنجم چند مثال از دسته بندی زمانی شامل استفاده از داده های اصلی، استفاده از تبدیل موج گسسته و استفاده از دسته بندی k-NN را ارائه خواهیم نمود. کلاس ts نمایش دهنده داده های نمونه از زمان می باشد. در مثال زیر داده سری زمانی با ۳۰ مقدار را بررسی می نماییم.

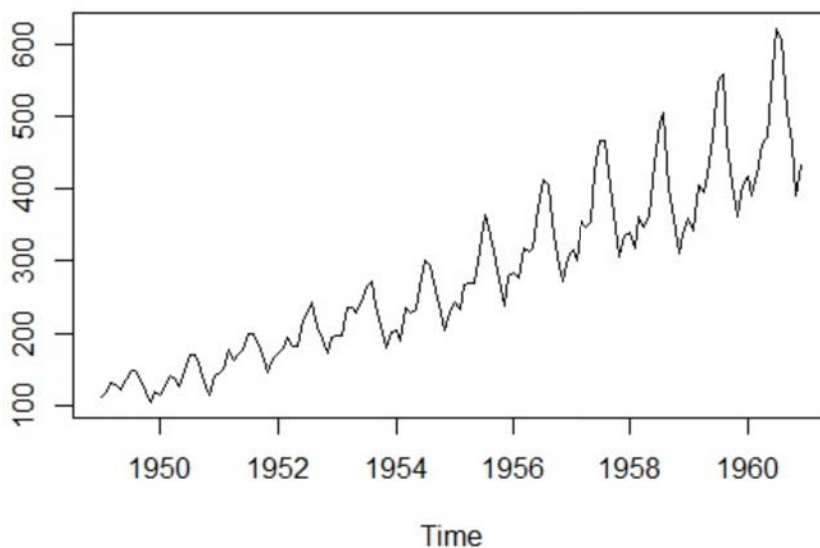
```
> tsvar <- ts(1:30, frequency=12, start=c(2012,5))
> print(tsvar)
      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
2012          1  2  3  4  5  6  7  8
2013   9  10  11  12  13  14  15  16  17  18  19  20
2014  21  22  23  24  25  26  27  28  29  30
> str(tsvar)
Time-Series [1:30] from 2012 to 2015: 1 2 3 4 5 6 7
8 9 10 ...
> attributes(tsvar)
$tsr
[1] 2012.333 2014.750 12.000

$class
[1] "ts"
```

تجزیه سری زمانی در R

تجزیه سری زمانی عبارت است از تجزیه سری زمانی به روند، فصل و اجزای دوره‌ای و نامنظم. بخش روندی اشاره به یک روند بلندمدت دارد. بخش فصلی اشاره به تغییرات فصلی دارد. بخش دوره‌ای شامل موارد تکراری و نوسانات دوره‌ای و اجزای نامنظم می‌باشد. سری زمانی مسافران هوایی که در ادامه بررسی می‌نماییم نمایش‌دهنده یک سری زمانی می‌باشد که در مثال در تجزیه سری زمانی مورد استفاده قرار گرفته است. این داده زمانی شامل اطلاعات مربوط به پروازهای بین سالهای ۱۹۴۹ تا ۱۹۶۰ یک شرکت هواپیمایی می‌باشد که ۱۴۴ مقدار دارد.

```
> plot(AirPassengers)
```



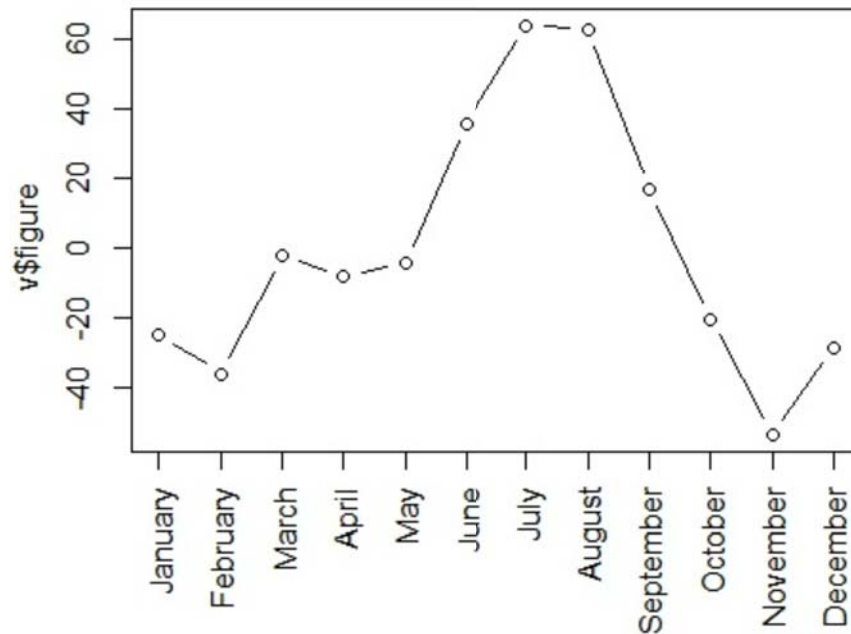
شکل ۶-۲: سری زمانی مجموعه داده مسافران هوایی

تابع `decompose()` در زیر بر روی داده مسافران هوایی به منظور شکستن این داده‌ها به اجزای مختلف استفاده می‌شود.

```

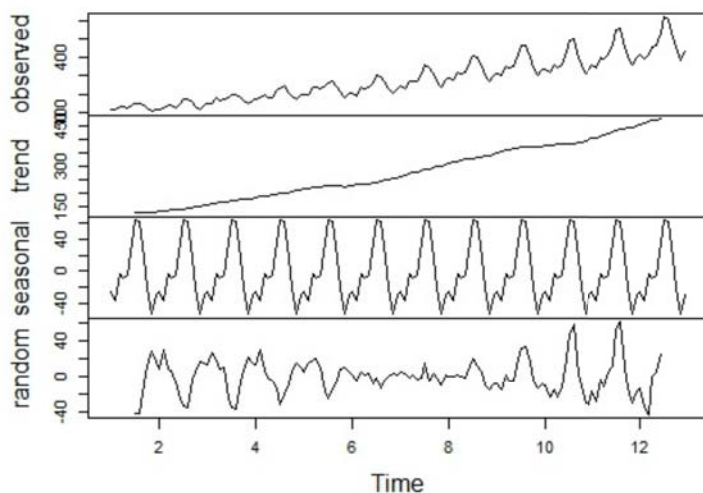
> Var <- ts(AirPassengers, frequency=12)
> v <- decompose(Var)
> v$figure
[1] -24.748737 -36.188131 -2.241162 -8.036616 -4.506313
[6] 35.402778 63.830808 62.823232 16.520202 -20.642677
[11] -53.593434 -28.619949
> plot(v$figure, type="b", xaxt="n", xlab="")
> monthNames <- months(ISOdate(2011,1:12,1))
> axis(1, at=1:12, labels=monthNames, las=2)

```



شکل ۶-۳: شکستن سری زمانی مجموعه داده مسافران هوایی

```
> plot(v)
```



شکل ۶-۴: تجزیه سری زمانی مجموعه داده مسافران هوایی

در این شکل نمودار اول سری زمانی طبیعی می‌باشد. نمودار دوم روندی از داده‌هاست و نمودار سوم مؤلفه‌های فصل را نمایش می‌دهد و نمودار آخر اجزای باقیمانده پس از حذف روندها و مؤلفه‌های فصلی می‌باشد. برخی توابع دیگر برای تجزیه سری‌های زمانی توسط تابع `stl()` در بسته `stat` و تابع `decomo()` در بسته `timsac` و تابع `tsr()` در بسته `ast` قابل دسترسی می‌باشند.

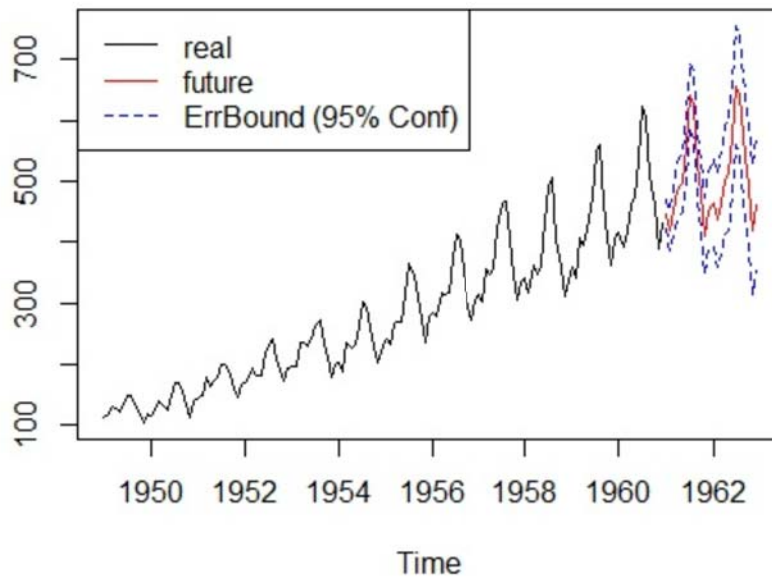
پیش‌بینی سری‌های زمانی

سری‌های زمانی پیش‌بینی کننده، رویدادهای آینده مبتنی بر داده‌های تاریخی را پیش‌بینی می‌نماید. در ادامه یک مثال برای پیش‌بینی قیمت سهام بر اساس عملکرد گذشته را بررسی می‌نماییم. دو مدل معروف سری‌های زمانی پیش‌بینی

ARMA^۱ و ARIMA^۲ می‌باشد. این مثال بر روی سری زمانی تک متغیره به‌منظور پیش‌بینی با استفاده از مدل ARIMA می‌باشد.

```
> f <- arima(AirPassengers, order=c(1,0,0),
+ list(order=c(2,1,0), period=12))
> fore <- predict(f, n.ahead=24)
> # error bounds at 95% confidence level
> U <- fore$pred + 2*fore$se
> L <- fore$pred - 2*fore$se
> ts.plot(AirPassengers, fore$pred, U, L, col=c(1,2,4,4),
+ lty = c(1,1,2,2))

> legend("topleft", c("real", "future", "ErrBound (95% Conf)"),
+ col=c(1,2,4), lty=c(1,1,2))
```



شکل ۵-۶: پیش‌بینی سری زمانی

در این شکل خطوط قرمز نمایش‌دهنده مقادیر پیش‌بینی شده و نقاط آبی محدوده خطا با درصد اطمینان حدود ۹۵ درصد می‌باشد.

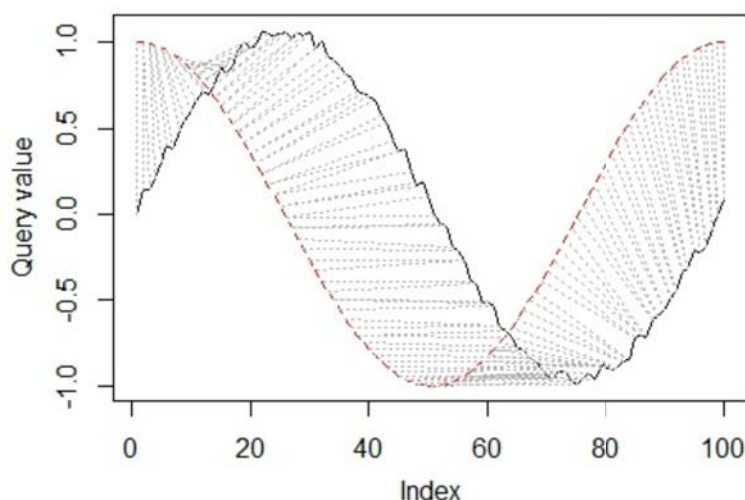
^۱ autoregressive moving average

^۲ autoregressive integrated moving average

خوشه‌بندی سری های زمانی - پیچیدگی زمانی پویا

خوشه‌بندی سری های زمانی عبارت است از تقسیم‌بندی داده‌های سری زمانی به گروه‌هایی که مبتنی بر مشابهت یا فاصله می‌باشند بنابراین سری های زمانی در یک خوشه یکسان دارای مشابهت با یکدیگر می‌باشند. معیارهای متفاوتی از فاصله یا عدم تشابه از جمله فاصله اقلیدسی^۱، فاصله منهتنی^۲، نرم‌های حداکثری^۳، فاصله همینگ^۴، فاصله بین دو بردار و فاصله DTW^۵ در این بخش بسیار معروف می‌باشند. روش DTW هم‌ترازی بهینه بین دو سری زمانی را پیدا می‌نماید و آن را با استفاده از بسته dtw پیاده‌سازی می‌نماید.

```
> library(dtw)
> idx <- seq(0, 2*pi, len=100)
> v1 <- sin(idx) + runif(100)/10
> v2 <- cos(idx)
> align <- dtw(v1, v2, step=asymmetricP1, keep=T)
> dtwPlotTwoWay(align)
```



شکل ۶-۶: خوشه‌بندی سری زمانی با استفاده از بسته dtw

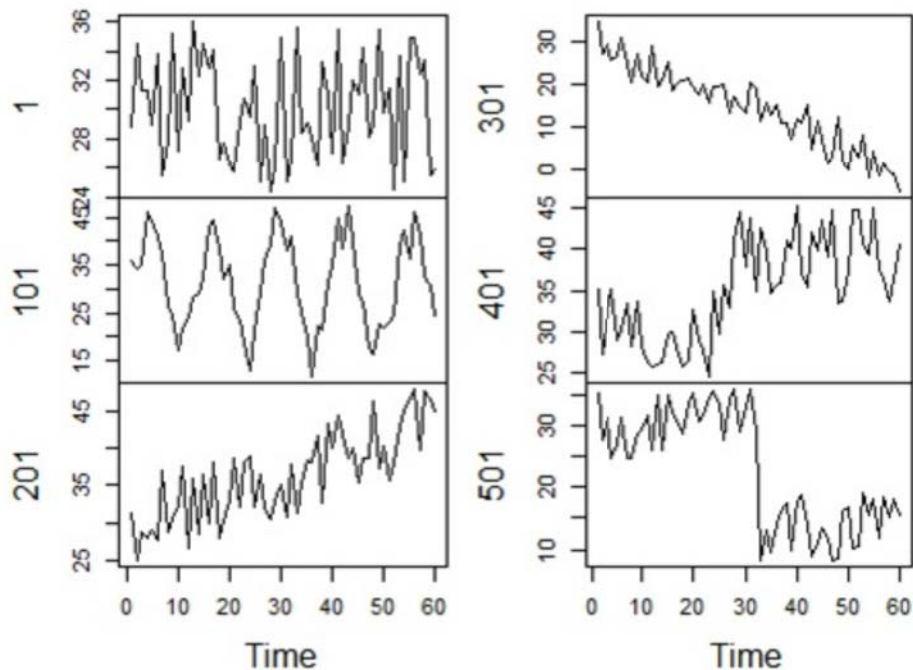
-
- ^۱ Euclidean distance
 - ^۲ Manhattan distance
 - ^۳ Maximum norm
 - ^۴ Hamming distance
 - ^۵ Dynamic Time Warping

سری زمانی نمودار کنترل ترکیبی برای مثال در بخش بعدی استفاده می‌شود. این مجموعه داده حاوی ۶۰۰ مثال نمودارهای کنترل ترکیبی می‌باشد که هر کدام از این نمودارهای کنترلی یک سری زمانی با ۶۰ مقدار می‌باشد و همچنین ۶ کلاس وجود دارد :

- ۱ الی ۱۰۰ : طبیعی
- ۱۰۱ الی ۲۰۰ : چرخه‌ای
- ۲۰۱ الی ۳۰۰ : روند افزایشی
- ۳۰۱ الی ۴۰۰ : روند کاهشی
- ۴۰۱ الی ۵۰۰ : شیفت به بالا
- ۵۰۱ الی ۶۰۰ : شیفت به پایین

خوشه‌بندی سری‌های زمانی – نمودار کنترل ترکیبی

```
> library(dtw)
> idx <- seq(0, 2*pi, len=100)
> v1 <- sin(idx) + runif(100)/10
> v2 <- cos(idx)
> align <- dtw(v1, v2, step=asymmetricP1, keep=T)
> dtwPlotTwoWay(align)
```

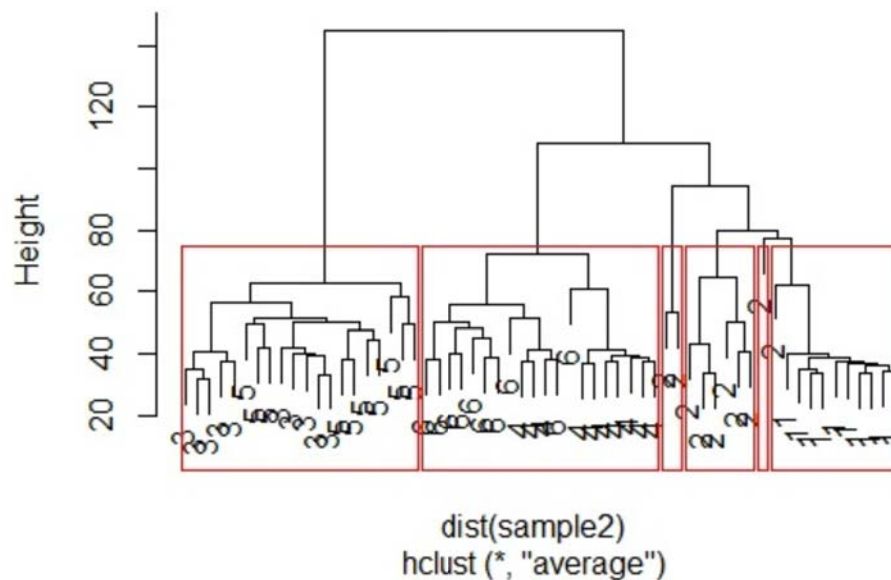


شکل ۶-۷: خوشه‌بندی سری زمانی با استفاده از نمودار ترکیبی

در ابتدا ۱۰ مورد را به صورت تصادفی از هر کلاس انتخاب می‌نماییم. در غیر این صورت نمونه‌های فراوانی خواهیم داشت و نقاط خوشه‌بندی سلسله مراتبی بسیار شلوغ و غیرقابل فهم خواهد بود.

خوشه‌بندی سری‌های زمانی - خوشه‌بندی سلسله‌مراتبی با استفاده از فاصله اقلیدسی

```
> set.seed(6218)
> number <- 10
> samp <- sample(1:100, number)
> idx <- c(samp, 100+samp, 200+samp, 300+samp,
  400+samp, 500+samp)
> sample2 <- sc[idx,]
> observedLabels <- rep(1:6, each=number)
> hcvar <- hclust(dist(sample2), method="average")
> plot(hcvar, labels=observedLabels, main="")
> rect.hclust(hcvar, k=6)
```



شکل ۶-۸: خوشه‌بندی سلسله‌مراتبی سری زمانی با استفاده از فاصله اقلیدسی

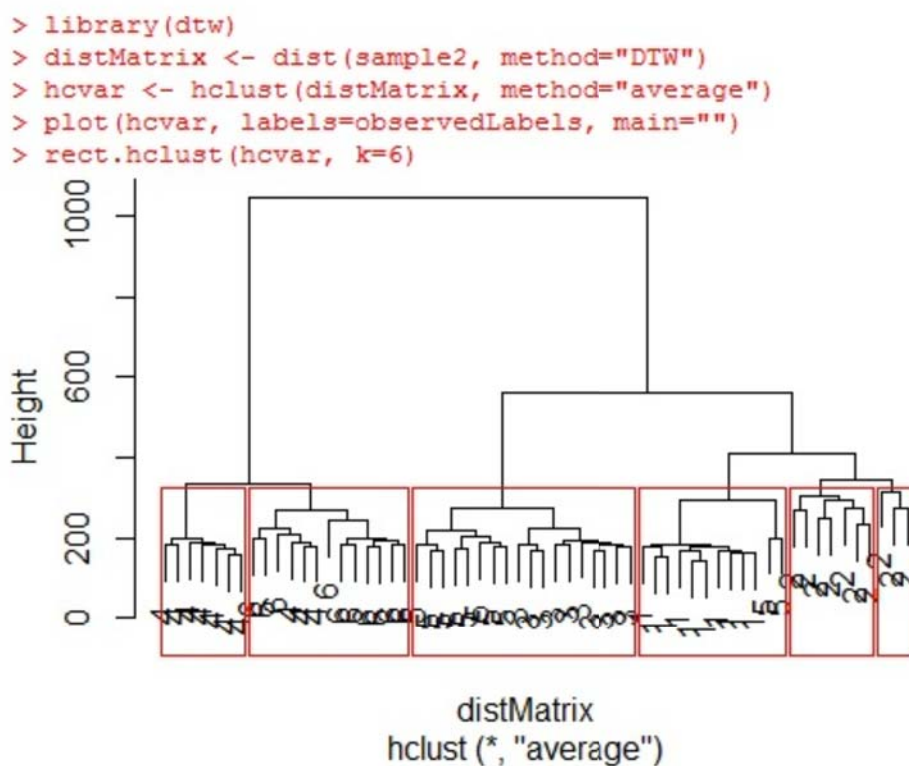
```
> mvar <- cutree(hcvar, k=6)
> table(observedLabels, mvar)
      mvar
observedLabels  1  2  3  4  5  6
      1 10  0  0  0  0  0
      2  1  6  2  1  0  0
      3  0  0  0  0 10  0
      4  0  0  0  0  0 10
      5  0  0  0  0 10  0
      6  0  0  0  0  0 10
```

نتایج خوشه‌بندی نشان می‌دهد که افزایش روندها (کلاس ۳) و تغییر به سمت بالا (کلاس ۵) به خوبی جدا نشده است و کاهش روندها (کلاس ۴) و تغییر به سمت پایین (کلاس ۶) با یکدیگر مخلوط شده است.

خوشه‌بندی سری‌های زمانی - خوشه‌بندی سلسله مراتبی با استفاده از

فاصله DTW

در ادامه خوشه‌بندی سلسله مراتبی را با استفاده از فاصله DTW بررسی می‌نماییم.



شکل ۶-۹: خوشه‌بندی سلسله مراتبی سری زمانی با استفاده از فاصله DTW

```
> mvar <- cutree(hcvar, k=6)
> table(observedLabels, mvar)
```

	mvar					
observedLabels	1	2	3	4	5	6
1	10	0	0	0	0	0
2	0	7	3	0	0	0
3	0	0	0	10	0	0
4	0	0	0	0	7	3
5	2	0	0	8	0	0
6	0	0	0	0	0	10

با مقایسه دو تصویر قبلی می‌توانیم ببینیم که فاصله DTW از فاصله اقلیدسی برای اندازه‌گیری مشابهت بین سری‌های زمانی بهتر می‌باشد.

دسته‌بندی سری‌های زمانی

دسته‌بندی سری زمانی برای ساختن مدل‌های دسته‌بندی بر اساس برجسب‌گذاری سری‌های زمانی و استفاده از این مدل به‌منظور پیش‌بینی سری‌های زمانی بدون برجسب می‌باشد. ویژگی‌های جدید استخراج‌شده از سری‌های زمانی ممکن است برای بهبود عملکرد مدل‌های دسته‌بندی کمک‌کننده باشد. تکنیک‌های استخراج ویژگی‌های شامل SVD^۱، DFT^۲، DWT^۳، PAA^۴، PIP^۵ و ... می‌باشد.

دسته‌بندی سری‌های زمانی با داده‌های اصلی

از تابع `ctree()` در بسته `party` برای نمایش دسته‌بندی سری‌های زمانی با داده‌های اصلی استفاده می‌نماییم. درخت تصمیم ساخته شده در شکل قابل‌مشاهده می‌باشد. قبل از دادن داده‌ها به تابع، برجسب‌های کلاس به مقادیر

^۱ Singular Value Decomposition

^۲ Discrete Fourier Transform

^۳ Discrete Wavelet Transform

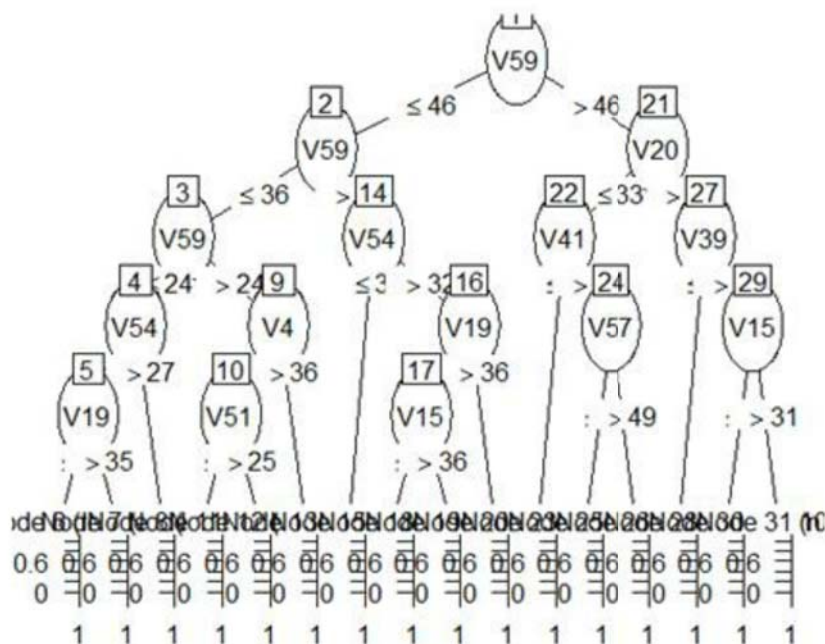
^۴ Piecewise Aggregate Approximation

^۵ Perpetually Important Points

طبقه‌بندی‌شده تغییر داده شده است بنابراین نمی‌توانیم برچسب کلاس‌ها را بر اساس اعداد حقیقی بدست آوریم.

```
> cid <- rep(as.character(1:6), each=100)
> nv <- data.frame(cbind(cid, sc))
> library(party)
> ct <- ctree(cid ~ ., data=nv, controls =
+ ctree_control(minsplit=30, minbucket=10, maxdepth=5))
> pcid <- predict(ct)
> table(cid, pcid)
      pcid
cid    1    2    3    4    5    6
  1   97    0    0    0    0    3
  2    1   93    2    0    0    4
  3    0    0   96    0    4    0
  4    0    0    0  100    0    0
  5    4    0   10    0   86    0
  6    0    0    0   87    0   13
> (sum(cid==pcid)) / nrow(sc)
[1] 0.8083333

> plot(ct, ip_args=list(pval=FALSE), ep_args=list(digits=0))
```



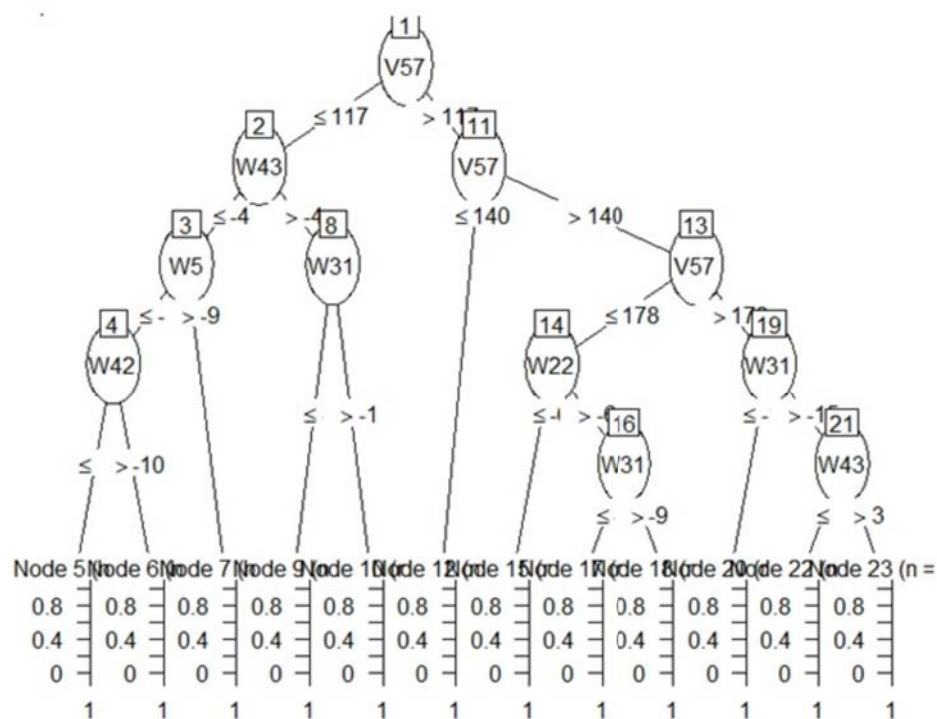
شکل ۶-۱۰: دسته‌بندی سری‌های زمانی داده‌های اصلی با تابع ctree()

دسته‌بندی سری‌های زمانی با استفاده از خصوصیت‌های استخراج‌شده

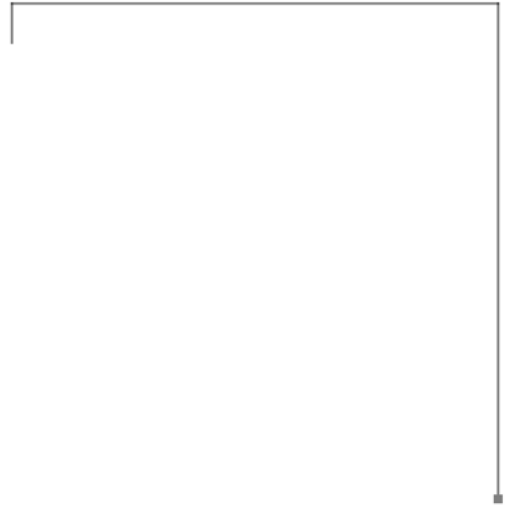
در ادامه از روش DWT برای استخراج ویژگی‌های سری زمانی و ساختن مدل دسته‌بندی استفاده می‌نماییم. این روش یک نمایش با دقت از امواج را شامل می‌شود. یک مثال از تبدیل موجی Haar که ساده‌تر DWT است یک تکنیک تبدیل فوریه‌ای معروف برای استخراج ویژگی‌ها می‌باشد.

```
> library(wavelets)
> wtv <- NULL
> for (i in 1:nrow(sc)) {a <- t(sc[i,])
+ wt <- dwt(a, filter="haar", boundary="periodic")
+ wtv <- rbind(wtv, unlist(c(wt@W, wt@V[[wt@level]])))}
> wtv <- as.data.frame(wtv)
> wtSc <- data.frame(cbind(cid, wtv))
> ct <- ctree(cid ~ ., data=wtSc, controls = ctree_control
+ (minsplit=30, minbucket=10, maxdepth=5))
> pcid <- predict(ct)
> table(cid, pcid)
      pcid
cid  1  2  3  4  5  6
  1 97  3  0  0  0  0
  2  1 99  0  0  0  0
  3  0  0 81  0 19  0
  4  0  0  0 63  0 37
  5  0  0 16  0 84  0
  6  0  0  0  1  0 99
>
> (sum(cid==pcid)) / nrow(wtSc)
[1] 0.8716667
```

```
> plot(ct, ip args=list(pval=FALSE), ep args=list(digits=0))
```



شکل ۶-۱۱: دسته‌بندی سری‌های زمانی با استفاده از خصوصیت‌های استخراج‌شده

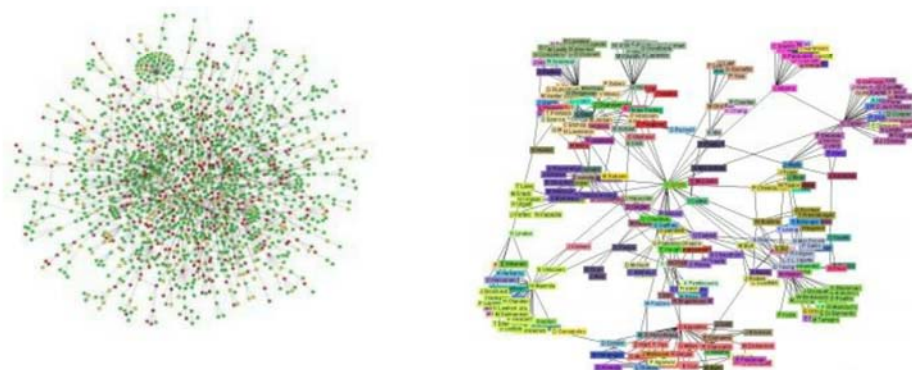


فصل پنجم

تحلیل شبکه های اجتماعی

۷-۱- تحلیل شبکه‌های اجتماعی

شبکه‌های اجتماعی، عبارت است از شبکه‌ای که شامل افراد و گروه‌ها و ارتباطات بین آن‌ها می‌باشد. افراد و گروه‌های عضو در این شبکه نود و گره‌ها را تشکیل می‌دهند و ارتباطات و وابستگی‌های بین این مؤلفه‌ها نیز مانند دوستی، خویشاوندی، تجارت، علایق مشترک و غیره یال، پیوند، پیکان، ربط یا بند بین نودها یا گره‌ها را تشکیل می‌دهند. با افزایش تعداد گره‌ها و ارتباطات بین آن‌ها شبکه دارای پیچیدگی بیشتری می‌شود و به واسطه تحلیل ریاضیاتی شبکه می‌توان آن‌ها را مورد تجزیه و تحلیل قرار داد. شبکه به صورت مجموعه‌ای از گره‌ها و روابط بین آن‌ها تعریف می‌شود. گره‌ها می‌توانند فرد، گروه، سازمان، کشور و غیره باشند. در واقع، در تحلیل شبکه، مطالعه روابط بین گره‌ها مورد نظر است. این روابط ممکن است جهت‌دار، بدون جهت، وزن‌دار یا دوتایی (صفر و یکی) باشد. ساده‌ترین نوع شبکه، شبکه روابط دوتایی بدون جهت است که فقط وجود یا عدم رابطه بین گره‌ها را نشان می‌دهد. با استفاده از وزن رابطه می‌توان آن را بیشتر توصیف کرد. وزن می‌تواند نشان‌دهنده میزان، تکرار یا شدت رابطه باشد. برای مثال، در رابطه بین سازمان‌ها وزن رابطه‌ها ممکن است نشان‌دهنده میزان تماس‌های آن‌ها باهم باشد (وسرمن، ۱۹۹۴؛ اسکات، ۱۹۹۱، گارتون و دیگران، ۱۹۹۹). اگر در شبکه رابطه‌ای جهت‌دار باشد به آن کمان و اگر بدون جهت باشد به آن یال می‌گوییم. (هوگان، ۲۰۰۷)



شکل ۷-۱: دو نمونه از شبکه‌های اجتماعی

تحلیل شبکه های اجتماعی یک روش و متد مهم در علوم اجتماعی مدرن می باشد. تحلیل شبکه های اجتماعی به دلیل رشد و توسعه چشمگیر شبکه های اجتماعی آنلاین بیشتر در بین افراد مطرح است اما تحلیل هر نوع شبکه ای اجتماعی را شامل می شود. هر شبکه اجتماعی شامل تعدادی گره است که می تواند مبین و معرف افراد، گروه ها، سازمان ها یا حتی کشورها باشد. روابط میان این گره های هم ماهیت را با یال هایی نشان می دهند که به آن لینک یا پیوند گفته می شود. بر اساس ارتباطاتی که بین گره های مختلف به واسطه ی مسیرهای ارتباطی مختلف وجود دارد، می توان نقل و انتقال سرمایه، پول، دوستی، صمیمیت یا هر متغیر دیگر مرتبط با یک شبکه اجتماعی را مورد تجزیه و تحلیل قرار داد. تحلیل گر شبکه می کوشد با ایجاد مدلی برای این ارتباطات، ساختار گروه را به مصورسازی نماید. در آینده پژوهشگری دیگر می تواند تأثیر این ساختار بر عملکرد گروه و یا تأثیر این ساختار بر افراد درون یک گروه را مورد بررسی قرار دهد. محققین حوزه شبکه های اجتماعی معتقدند که موفقیت یا شکست یک فرد یا سازمان بیش از آن که به فرد و قابلیت های فردی او وابسته باشد، به نوع ارتباطات او در یک گروه، دادوستدهای اجتماعی و ارتباطات میان فردی بستگی دارد.

به منظور بررسی روابط اجتماعی به مدلی برای نشان دادن افراد، سازمان ها یا مؤلفه های درگیر در رابطه و ارتباطات میان آنها احتیاج داریم. همچنین به ابزار و نظریه ای برای تجزیه و تحلیل این ارتباطات نیاز داریم که بدین منظور معمولاً از نظریه های ریاضیاتی چون نظریه گراف، نظریه ماتریس ها یا روش های انجام اعمال ریاضیاتی بر روی ماتریس ها استفاده می شود. در هر شبکه اجتماعی مؤلفه های کنشگر را، به عبارت دیگر یعنی مصداق های نمایش دهنده گروهی از انسان ها یا سازمان ها را به عنوان یک نقطه، دایره یا شیء مشابه به حساب می آورند و ارتباطات میان آنها را با خطوط نشان می دهند. به نگاره ای که بر این اساس به دست می آید، شبکه اجتماعی گفته می شود. نگاره ها یا گراف های به دست آمده امکان بررسی دیداری روابطی را مهیا می کنند که شاید در زندگی واقعی به راحتی امکان پذیر نباشد. یکی از مسائل مهمی که همواره در جامعه شناسی مورد توجه بوده بررسی الگوهای روابط عناصر در سطوح مختلف جامعه است: روابط بین مردم،

نهاده‌ها، سازمان‌ها، دولت‌ها و غیره (برکوویتز، ۱۹۸۲؛ و سرمن، ۱۹۹۴؛ ولمن، ۱۹۹۸).

تحلیل شبکه اجتماعی با هدف تحلیل این روابط از دهه ۱۹۴۰ در انسان‌شناسی باب شد و مورد استفاده قرار گرفت و بعدها در جامعه‌شناسی آمریکایی و کانادایی تکامل و توسعه یافت (ولمن، ۱۹۸۸).

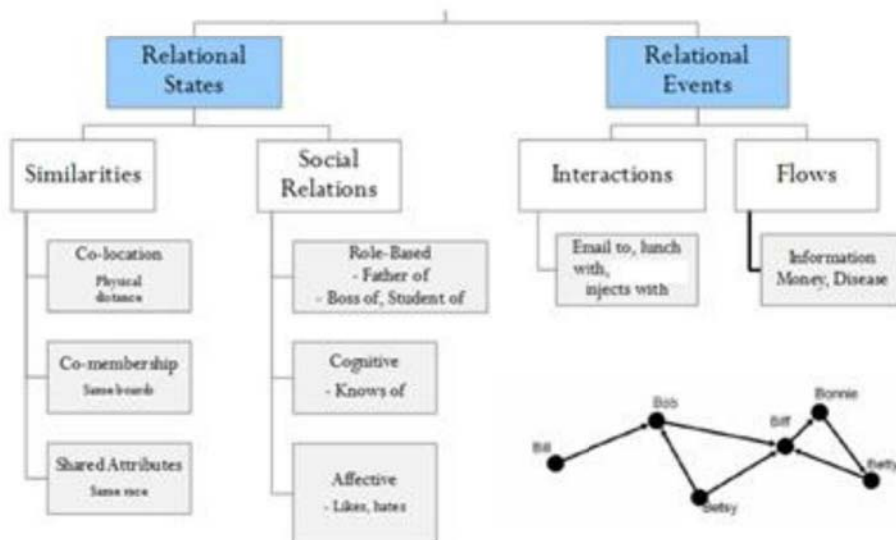
اساس تحلیل شبکه این عقیده است که «برای توضیح سازمان‌دهی اجتماعی نباید از محرک‌های درونی یا نیروهای خارجی انتزاعی استفاده کرد، بلکه می‌توان ساختار روابطی را که محدودکننده یا تواناکننده هستند مورد بررسی قرار داد» (همان). در واقع، هدف تحلیل شبکه، مطالعه ساخت است و به همین منظور مجموعه‌ای از نظریه‌ها، مفاهیم، اصول روشی، تکنیک و ابزار را ایجاد کرده است.

تحلیل شبکه بر بررسی تجربی ساختار اجتماعی به‌عنوان روابط و پیوندها بین کنشگران تأکید خاصی دارد و برای رسیدن به این هدف اصول، روش‌ها، تکنیک‌ها، ابزارها و حتی موارد تحلیلی خاصی را پیشنهاد می‌کند. این رویکرد به مطالعه جریان منابع و نحوه دسترسی افراد به این منابع نهفته در شبکه‌ها، که اطلاعات یکی از مهم‌ترین آن‌هاست، علاقه خاصی دارد. مواردی مانند بررسی شبکه‌های غیررسمی و تأثیرات آن‌ها در سازمان‌ها، روابط بین‌الملل، بیماری‌های واگیردار، شبکه‌های تروریستی یا بزهکاران، شبکه‌های بانفوذ در جریان‌های اقتصادی یا سیاسی، شبکه‌های افراد، انواع حمایت‌هایی که برای آن‌ها فراهم می‌کنند و تأثیر حمایت‌ها در زندگی آن‌ها و... از موارد مورد مطالعه این رویکرد هستند.

داده‌های شبکه اجتماعی شامل دست‌کم یک متغیر ساختاری است که روی مجموعه‌ای از کنشگران اندازه‌گیری شده است. معمولاً مسئله تحقیق و نظریه‌ها تعیین می‌کنند که این متغیرها چه هستند و چه تکنیک‌هایی برای اندازه‌گیری آن‌ها مناسب‌تر است (وسرمن، ۲۸: ۱۹۹۴).

در شکل زیر که برگرفته از کتاب بورگتی است، انواع ارتباطات افراد مورد بررسی قرار می‌گیرد. روابط در این شکل به دو قسمت عمده روابط موقعیتی و روابط

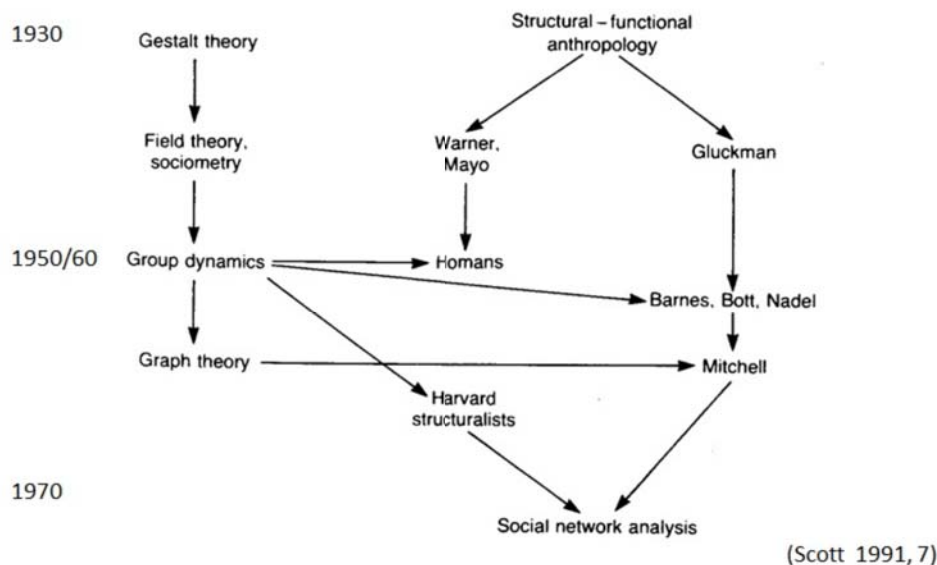
رویدادی تقسیم‌بندی شده است. روابط موقعیتی به دو قسمت عمده مشابهت‌ها و روابط اجتماعی بخش‌بندی می‌شود. از جمله روابط زیرمجموعه مشابهت می‌توان به هم موقعیت بودن، هم عضو بودن و دارای صفات یکسان بودن اشاره نمود. همچنین نقش‌های اجتماعی، شناخت، عقیده نسبت به دیگران از جمله موارد روابط اجتماعی می‌باشد. در حوزه رویدادهای ارتباطی دو نوع رویداد تعامل شامل ارتباطات ایمیلی و جریان کار شامل انتقال اطلاعات، شیوع بیماری و انتقال پول از جمله موارد زیرمجموعه می‌باشد.



Borgatti, Brass, Mehra & Labianca, 2009 and Borgatti & Halgin 2011

شکل ۷-۲: انواع رابطه‌های موجود

همان‌طور که در شکل زیر مشاهده می‌شود روند توسعه و شکل‌گیری دانش تحلیل شبکه‌های اجتماعی به صورت نموداری مشاهده می‌گردد.



شکل ۷-۳: نگاهی به توسعه تحلیل شبکه‌های اجتماعی

۷-۲- انواع مرکزیت و شاخص‌های اصلی در تحلیل شبکه

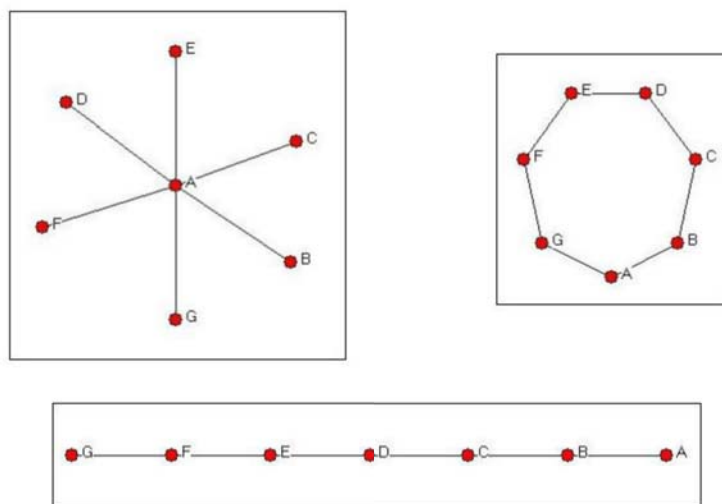
ویژگی‌های شبکه‌های اجتماعی به سه دسته ساختی، تعاملی و کارکردی تقسیم می‌شوند که هر کدام شاخص‌هایی دارند. این شاخص‌ها بنابر مسئله و هدف تحقیق انتخاب می‌شوند. «منظور از ویژگی‌های ساختی شبکه ویژگی‌هایی است که بیشتر با ساخت و نه محتوای شبکه ارتباط دارند؛ مانند اندازه، تراکم و ترکیب. ویژگی‌های تعاملی بیشتر به خصوصیات مربوط به روابط بین اعضا مانند فراوانی تماس‌ها، قوت، چندگانگی، نزدیکی، مدت رابطه و ... می‌پردازد. در ویژگی‌های کارکردی به کارکردهایی که شبکه برای اعضا دارد، مانند انواع حمایت‌های اجتماعی شبکه، توجه می‌شود» (باستانی، ۱۳۸۵).

- اندازه شبکه : تعداد کل پیوندهای موجود در شبکه را نشان می‌دهد.

- تراکم شبکه : یکی از شاخص‌هایی است که از آن زیاد استفاده می‌شود. این شاخص به صورت نسبت تعداد همه پیوندهای موجود به همه پیوندهای ممکن تعریف می‌شود. این شاخص معرف میزان همبستگی شبکه است. (باستانی، ۲۰۰۷). در یک شبکه به هم پیوسته با تراکم بالا، روابط مستقیم زیادی بین اعضا وجود دارد؛ این همان وضعیت روستای کوچک قدیمی یا گروه‌های کاری است. (ولمن، ۱۹۹۹). امروزه در شبکه‌هایی که با واسطه کامپیوتر شکل می‌گیرد، امکان وجود چنین شبکه‌های با تراکم بالا خیلی کم شده است.

- مرکزیت و قدرت: همه جامعه‌شناسان معتقدند قدرت خصوصیت اساسی ساختارهای اجتماعی است. در تحلیل شبکه تحلیل قدرت با مفهوم مرکزیت ارتباط زیادی دارد. شاید مهم‌ترین فرض در رویکرد شبکه این است که قدرت اساساً رابطه‌ای است. یک فرد به تنهایی نمی‌تواند قدرت داشته باشد، چرا که نمی‌تواند بر دیگران مسلط باشد. قدرت یک فرد، وابسته به دیگران است. چون قدرت به ساختار وابسته است؛ بنابراین، می‌تواند خیلی متغیر باشد. اگر سیستمی خیلی کم همبسته باشد (تراکم کم)، قدرت زیادی نمی‌تواند در آن اعمال شود. قدرت هم در سطح کلان و هم در سطح خرد مطرح است. در تحلیل شبکه قدرت در هر دوی این سطوح قابل مطالعه و بررسی است، چرا که رویکرد شبکه این دو سطح را به هم پیوند می‌دهد (هنمن و رایدل، ۱۴۵: ۲۰۰۵).

برای درک بهتر شیوه‌هایی که تحلیل شبکه برای مطالعه قدرت به کار می‌برد، ابتدا توجه شما را به چند سیستم خیلی ساده جلب می‌کنیم. به گراف‌های سه شبکه ساده در شکل‌های زیر توجه کنید:



شکل ۷-۴: انواع ساختارهای ارتباطی

از مشاهده این شکل‌ها برمی‌آید که موقعیت کنشگر A در شبکه ستاره‌ای از همه کنشگران بهتر است. اما چرا این کنشگر موقعیتی بهتر از دیگران در شبکه ستاره‌ای دارد؟ آیا واقعاً همه کنشگران در شبکه دایره‌ای موقعیت ساختاری یکسانی دارند؟ (هنمن و رایدل، ۱۴۶: ۲۰۰۵).

این سؤالات را می‌توان با توجه به مفهوم مرکزیت در تحلیل شبکه پاسخ داد. این مفهوم شامل چند شاخص است که در ادامه آن‌ها را بیان می‌کنیم.

درجه: ساده‌ترین تعریف از مرکزیت کنشگر این است که کنشگران مرکزی باید فعال‌ترین کنشگران باشند و بیشترین پیوندها را با کنشگران دیگر داشته باشند (وسرمن، ۱۷۸: ۱۹۹۴). در گراف‌های جهت‌دار دو درجه ورودی و خروجی برای یک گره محاسبه می‌شود که اولی نشان‌دهنده پیوندهای خروجی است و دومی پیوندهای ورودی گره را نشان می‌دهد. تعبیر جامعه‌شناختی این دو شاخص به این صورت است که پیوندهای خروجی به معنای ارائه منابعی به شبکه است (که بیشتر برای اطلاعات مورد استفاده قرار می‌گیرد) و پیوندهای ورودی به معنای دریافت منابع است. «میزان بالای درجه خروجی نشان‌دهنده اقتدار است. به این معنا که این نوع گره‌ها خیلی سریع می‌توانند اطلاعاتی را انتشار دهند. میزان

بالای درجه ورودی نیز نشان دهنده شهرت فرد است. این به معنای آن است که افراد زیادی به این گره ها توجه و مراجعه می کنند.» (هوگان، ۲۰۰۷)

حال کنشگر A در شبکه ستاره ای دارای بالاترین درجه است (البته در اینجا روابط بدون جهت فرض شده اند). این کنشگر فرصت های بیشتری نسبت به دیگران در شبکه دارد. اگر این کنشگر به منبعی در شبکه نیاز داشته باشد، می تواند از طریق روابط زیادی که دارد آن را به دست آورد. چنانچه کنشگری از ارائه منبعی به A خودداری کرد، این کنشگر می تواند از طریق پیوندهای دیگرش آن منبع را به دست آورد. اما این مسئله در مورد کنشگران دیگر صادق نیست. آن ها فقط یک پیوند و یک راه برای دسترسی به منابع شبکه دارند و بنابراین با محدودیت های بیشتری مواجه اند. این موقعیت کنشگر A باعث می شود که کمتر به کنشگر خاص دیگری وابسته باشد و بنابراین قدرت بیشتری داشته باشد.

در شبکه دایره ای همه کنشگران درجات برابری دارند و بنابراین همه موقعیت یکسانی به لحاظ سود و زیان در شبکه دارند. در شبکه خطی، کنشگران انتهایی، A و G، موقعیتی متفاوت با بقیه دارند. در این شبکه این دو کنشگر فرصت ها و در نتیجه قدرت کمتری دارند (همان). با استفاده از این شاخص می توان گره های مرکزی (از نظر درجه) را مشخص کرد. همچنین می توان تغییرات این شاخص را در زمان مشاهده کرد که نشان دهنده میزان پویایی شبکه در زمان است. به صورت خلاصه تر می توان مفهوم درجه را در تعریف مرکزیت درجه ای مشاهده کرد.

مرکزیت درجه^۱: تعداد ارتباطات مستقیمی که هر نقطه داراست و نشان‌دهنده میزان اطلاعاتی است که یک نقطه احتمالاً از اطلاعاتی که در شبکه در جریان است دریافت می‌کند.

مرکزیت بینابینی^۲: نقطه‌ای است که بینابین بسیاری از جفت نقاط دیگر باشد؛ درواقع نقاطی واسطه‌ای هستند که راه‌های ارتباطی نقاط دیگر از آن‌ها می‌گذرد. این نقاط دارای قدرت ایزوله کردن یا افزایش ارتباطات می‌باشند. مرکزیت بینابینی به‌طور خلاصه عبارت است از تعداد افرادی در شبکه که یک شخص به‌طور غیرمستقیم از طریق خطوط مستقیم آن‌ها متصل شده است.

مرکزیت نزدیکی^۳: فاصله یک فرد با کلیه افراد دیگر در شبکه را می‌سنجد، هر چه یک فرد به دیگران نزدیک‌تر باشد، آن فرد برگزیده‌تر و مشهورتر است. افرادی با نمرات نزدیکی بالا، احتمالاً اطلاعات را خیلی سریع‌تر از دیگران دریافت می‌کنند، به خاطر اینکه میانجی‌های کمتری بین آن‌ها وجود دارد. سنجه مرکزیت نزدیکی بر اساس فاصله ژئودیسک محاسبه می‌شود. این سنجه مقدار فاصله یک گره از سایر گره‌ها را اندازه‌گیری می‌کند. این سنجه نشان‌دهنده‌ی دسترس‌پذیری، سلامت و امنیت عامل‌ها می‌باشد. (Frank, ۲۰۰۲).

مرکزیت بردار ویژه^۴: یکی دیگر از سنجه‌های مرکزیت می‌باشد و بر اساس این ایده پیشنهاد شده است که مرکزیت یک گره خاص نمی‌تواند مجزا از مرکزیت دیگر گره‌هایی که با آن متصل شده است، تخمین زده شود. نمرات مرکزیت، به گره‌ها بر اساس این اصل که ارتباط به گره‌های با نمره بالا در نمرات یک گره خاص نسبت به ارتباط (اتصال) به گره‌های با نمره پایین مشارکت بیشتری دارد، اختصاص داده می‌شود (Bonacich, ۱۹۷۲).

^۱ Degree centrality

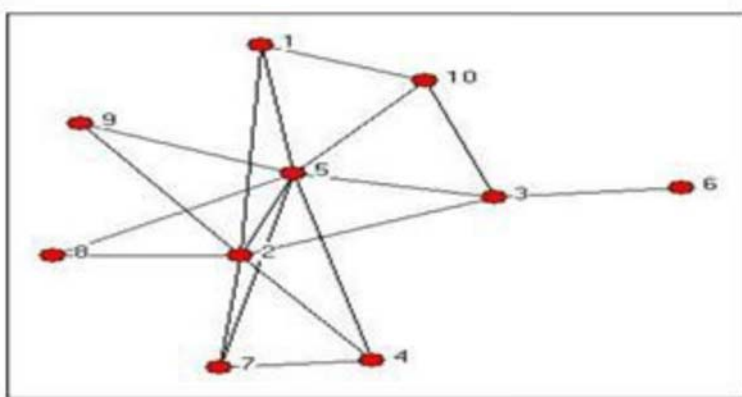
^۲ Betweenness centrality

^۳ Closeness centrality

^۴ Eigenvector centrality

ایزوله‌ها: گره‌هایی هستند که در هیچ مؤلفه‌ای قرار نمی‌گیرند. ایزوله‌ها در واقع هیچ ارتباطی با دیگران در شبکه ندارند. میزان گره‌های ایزوله در شبکه نشان‌دهنده میزان کنشگران غیرفعالی است که در هیچ رابطه‌ای مشارکت نداشته‌اند.

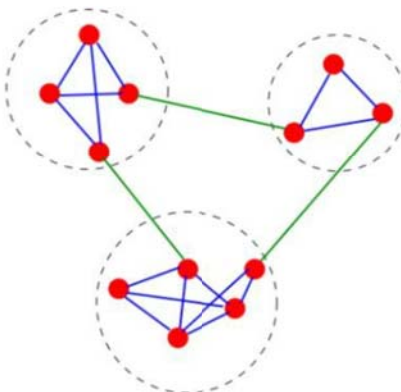
گراف زیربیشینه کامل: زیر گراف‌های حداکثری که همه گره‌های آن به هم متصل باشد.



شکل ۷-۵: گراف زیربیشینه کامل

۷-۳-۱ اجتماع یابی در شبکه

اجتماع در یک شبکه به مجموعه‌ای از گره‌ها گفته می‌شود که یال‌هایی که این گره‌ها را به هم وصل کرده است بسیار بیشتر از یال‌هایی است که این گره‌ها را به سایر گره‌های شبکه وصل می‌کنند. (Fortunato, ۲۰۱۰)



شکل ۷-۶: اجتماع یابی در شبکه‌های اجتماعی

شبکه $G=(V,E)$ را در نظر بگیرید که V ، مجموعه گره‌ها و E مجموعه یال‌های آن باشد و A ، ماتریس مجاورت (یا وزن) باشد. الگوریتم‌های اجتماع یابی الگوریتم‌هایی هستند که G و A را به عنوان ورودی گرفته و افرازی از گره‌ها را به عنوان خروجی در اختیار ما قرار دهد (Sheldon, ۲۰۱۰).

در ادامه به برخی مثال‌ها در زمینه اجتماع یابی اشاره می‌کنیم.

- اجتماع یابی در شبکه‌ی وب: در شبکه وب، اجتماعات، وبسایتهای دارای موضوعات وابسته یا مشترک هستند. گروه‌بندی مشتری‌های خرید اینترنتی یک سایت مخصوص که دارای علائق یکسانی هستند، این امکان را می‌دهند که سیستم‌های توصیه‌ای قوی برای این سایت‌ها در نظر بگیریم.

(Dourisboure et al., ۲۰۰۷; Flake et al, ۲۰۰۲).

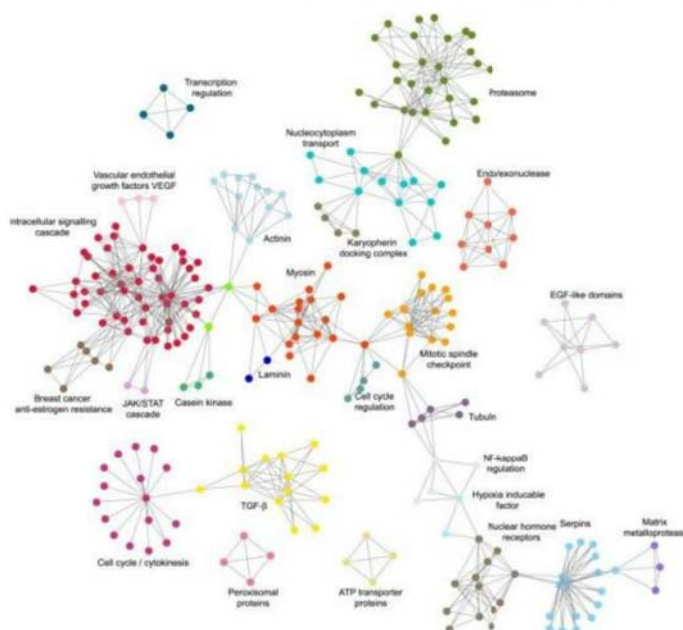
- اجتماع یابی در شبکه‌ی ارتباطات پروتئینی: زمانی که دو یا بیش از دو پروتئین به یکدیگر متصل شوند تعامل پروتئین-پروتئین رخ می‌دهد،

که غالباً عملکردهای بیولوژیکی را انجام می دهند. در شبکه های پروتئینی، پروتئین ها بیشتر تمایل دارند تا با پروتئین اصلی که در اجتماع خود قرار دارند ارتباط تعامل داشته باشند و یال های بین گروه ها هم عامل اصلی ارتباط هستند.

(Rives and Galitski, ۲۰۰۳)

- اجتماع یابی در شبکه های اجتماعی: در شبکه های اجتماعی، اجتماعات گروه های مختلف موجود در جامعه هستند. گروه های کاری و دوستی، روستاها، شهرستان ها و ... با شناسایی این اجتماعات بهتر می توان شبکه های اجتماعی را تحلیل کرد. (Fortunato, ۲۰۱۰)
- اجتماع یابی در شبکه های علمی (مرجع دهی): در شبکه های علمی اجتماعات نشان دهنده زمینه های مختلف علمی هستند.

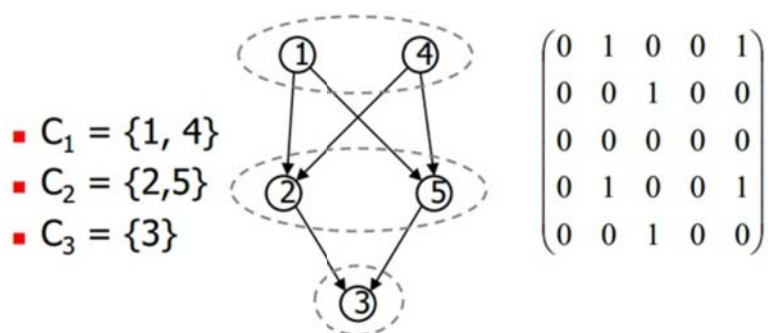
(Rosvall and Bergstrom, ۲۰۰۸)



شکل ۷-۷: اجتماع یابی در تحلیل شبکه ای

۷-۴- هم‌ارزی ساختاری^۱:

هم‌ارزی ساختاری به مشخصه‌های دارای روابط مشابه در یک شبکه می‌پردازد و رویه‌های تحلیل شباهت‌های ساختاری کنش‌گرا و الگوهای روابط در شبکه‌های چند رابطه‌ای را بررسی می‌نماید و هدف آن بازنمایی الگوهای شبکه اجتماعی پیچیده به شکلی ساده برای آشکارسازی زیرمجموعه‌های کنشگرهایی که به طور مشابهی در روابط شبکه نهفته‌اند، می‌باشد. کنشگر i و j در صورتی هم‌ارز ساختاری هستند که برای همه کنشگرها و روابط، کنشگر i بند به k است اگر و فقط اگر j بند به k باشد و i بندی از k دارد اگر و فقط اگر j نیز بندی از k داشته باشد. اگر i و j هم‌ارز ساختاری باشند، آنگاه به موقعیت یکسانی دست می‌یابند. دو گره هم‌ارز ساختاری هستند اگر از گره‌ها مشابهی به گره‌های مشابهی ارتباط داشته باشند. (Lorrain and white)



شکل ۷-۸: هم‌ارزی ساختاری

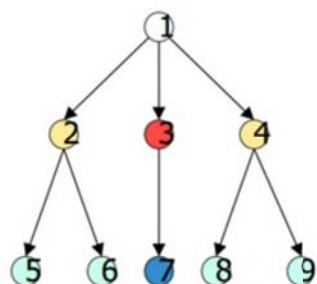
نگاهی به انواع هم‌ارزی‌ها:

هم‌ارزی ساختاری^۲: در این هم‌ارزی به همسایه مشابه خود متصل است. در شکل $\{۵, ۶\}$ و $\{۸, ۹\}$ دارای هم‌ارزی ساختاری هستند.

^۱ Structural equivalence^۲ Structural

هم‌ارزی خودسانی^۱: این هم‌ارزی مرتبط با توزیع مشابه از نظر رنگ‌بندی می‌باشد. در شکل $\{۴،۲\}$ و $\{۵،۶،۸،۹\}$ دارای هم‌ارزی خودسانی می‌باشند.

هم‌ارزی منظم^۲: این هم‌ارزی مرتبط با موقعیت مشابه می‌باشد. $\{۱\}$ ، $\{۲،۳،۴\}$ و $\{۵،۶،۷،۸،۹\}$ دارای این هم‌ارزی می‌باشند.



شکل ۷-۹: هم‌ارزی ساختاری

۷-۵- تحلیل شبکه‌ای در نرم‌افزار R

چند بسته اصلی در زبان برنامه‌نویسی R وجود دارد که بیش از دیگر بسته‌ها در زمینه تحلیل شبکه‌های اجتماعی کاراست: بسته igraph، بسته statnet و بسته sna. البته بسته tent نیز در برخی موارد مورد استفاده قرار می‌گیرد. در این بخش ابتدا به نحوه ورود داده‌ها به این نرم‌افزار می‌پردازیم. داده‌ها را در برنامه R در فرمت‌های مختلفی می‌توان وارد نمود. از جمله این فرمت‌ها می‌توان به ماتریس مجاورت^۳، لیست یال‌ها^۴، لیست مجاورت^۵، ماتریس وابستگی^۶ و ...

^۱ Automorphic

^۲ Regular

^۳ adjacency matrix

^۴ edge lists

^۵ adjacency lists

^۶ affiliation matrix

ورود داده‌ها

ماتریس مجاورت ماتریسی شامل سطرها و ستون‌هاست که نمایش‌دهنده گره‌ها و یال‌ها می‌باشد. یک ماتریس مجاورت غیر وزن‌دار دارای مقادیر ۰ و ۱ می‌باشد در صورتی که ماتریس مجاورت وزن‌دار می‌تواند شامل دیگر اعداد نیز باشد که نشان‌دهنده وزن یال ارتباطی می‌باشند.

شکل زیر مثالی از ماتریس مجاورت بدون وزن می‌باشد که به صورت فرمت CSV ذخیره شده است.

	A	B	C	D	E	F	G	H
1		a	b	c	d	e	f	g
2	a		0	1	0	1	0	1
3	b		1	0	1	1	0	1
4	c		0	1	0	0	0	0
5	d		1	1	0	0	1	1
6	e		0	0	0	1	0	1
7	f		1	1	0	1	1	0
8	g		0	0	0	0	0	1

شکل ۷-۱۰: مثالی از مجموعه داده‌ها در قالب ماتریس مجاورت

با استفاده از کدهای زیر از بسته `igraph` می‌توانیم این داده‌ها را وارد نموده و به صورت ماتریس و گراف ذخیره نماییم.

```
> library(igraph)
> file=read.csv(file.choose(),header=TRUE,row.names=1,check.names=FALSE)
> matr=as.matrix(file)

> graph=graph.adjacency(matr,mode="undirected",weighted=NULL)
> graph
IGRAPH UN-- 7 10 --
+ attr: name (v/c)
> matr
      23732 23778 23824 23871 58009 58098 58256
23732      0      1      0      1      0      1      0
23778      1      0      1      1      0      1      0
23824      0      1      0      0      0      0      0
23871      1      1      0      0      1      1      0
58009      0      0      0      1      0      1      0
58098      1      1      0      1      1      0      1
58256      0      0      0      0      0      1      0
```

همچنین نحوه ورود با استفاده از بسته statnet به صورت زیر می باشد :

```
> library(statnet)
> evar=read.csv(file.choose(),header=TRUE,
  row.names=1,check.names=FALSE)
> matr2=as.matrix(evar)
> net=network(matr,matrix.type="adjacency",directed=FALSE)
> net
Network attributes:
  vertices = 7
  directed = FALSE
  hyper = FALSE
  loops = FALSE
  multiple = FALSE
  bipartite = FALSE
  total edges= 10
    missing edges= 0
    non-missing edges= 10

Vertex attribute names:
  vertex.names

No edge attributes

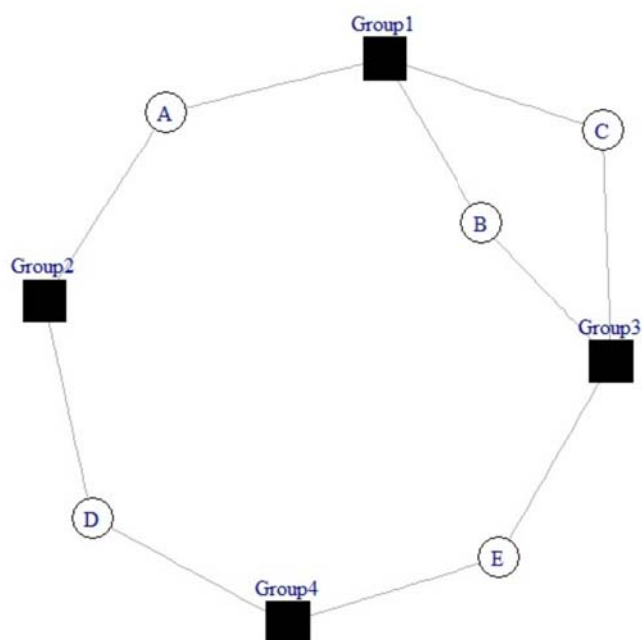
> matr2
  a b c d e f g
a 0 1 0 1 0 1 0
b 1 0 1 1 0 1 0
c 0 1 0 0 0 0 0
d 1 1 0 0 1 1 0
e 0 0 0 1 0 1 0
f 1 1 0 1 1 0 1
g 0 0 0 0 0 1 0
```

همچنین داده را در قالبی به‌غیر از ورود به‌صورت فایل می‌توان ایجاد نمود. در قطعه کد زیر ایجاد این مجموعه داده را مشاهده می‌نمایید.

```
> A=c(1,1,0,0)
> B=c(1,0,1,0)
> C=c(1,0,1,0)
> D=c(0,1,0,1)
> E=c(0,0,1,1)
> bm=matrix(c(A,B,C,D,E),nrow=5,byrow=TRUE)
> dimnames(bm)=list(c("A","B","C","D","E"),
  c("Group1","Group2","Group3","Group4"))
> bm
  Group1 Group2 Group3 Group4
A      1      1      0      0
B      1      0      1      0
C      1      0      1      0
D      0      1      0      1
E      0      0      1      1
> bg=graph.incidence(bm)
> V(bg)$type
[1] FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE
> V(bg)$name
[1] "A"      "B"      "C"      "D"      "E"
   "Group1" "Group2" "Group3" "Group4"

> shapes=c("circle","circle","circle","circle","circle",
+ "square","square","square","square")
> labeldistances=c(0,0,0,0,0,0.6,0.6,0.6,0.6)
> plot(bg,vertex.shape=shapes,vertex.label.degree=-pi/2,
+ vertex.label.dist=labeldistances,vertex.color=V(bg)$type)
```

با استفاده از قطعه کد بالا این مجموعه داده به صورت یک مجموعه داده در دو دسته ایجاد گردید و نمودار گراف آن ترسیم گردیده است.



شکل ۷-۱۱: مثالی از گراف دوبخشی و ارتباطات مربوط به آن

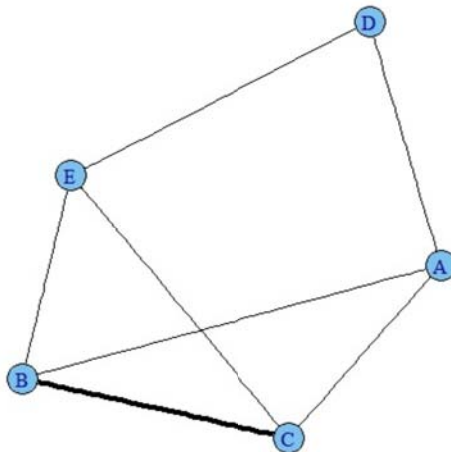
برای تحلیل وضعیت هم‌گروهی هر کدام از مؤلفه‌ها می‌توان از تابع `bipartite` استفاده نمود.

```
> pr=bipartite.projection(bg)
> pr
$proj1
IGRAPH UNW- 5 7 --
+ attr: name (v/c), weight (e/n)

$proj2
IGRAPH UNW- 4 4 --
+ attr: name (v/c), weight (e/n)

> get.adjacency(pr$proj1,sparse=FALSE,attr="weight")
  A B C D E
A 0 1 1 1 0
B 1 0 2 0 1
C 1 2 0 0 1
D 1 0 0 0 1
E 0 1 1 1 0
> plot(pr$proj1,edge.width=E(pr$proj1)$weight^2,
+ edge.color="black",vertex.label=V(pr$proj1)$name)
```

در قطعه کد بالا ماتریس هم‌گروهی اعضا ایجاد گردیده که گراف آن را در شکل زیر مشاهده می‌نمایید.



شکل ۷-۱۲: گراف مربوط به ماتریس تشکیل شده

محاسبات روی گره‌ها و تحلیل شبکه‌ای :

برخی از توابع مربوط به تحلیل شبکه‌های اجتماعی را در جدول زیر مشاهده می‌نمایید. این توابع به صورت مجزا در بسته‌های igraph و statnet آورده شده است.

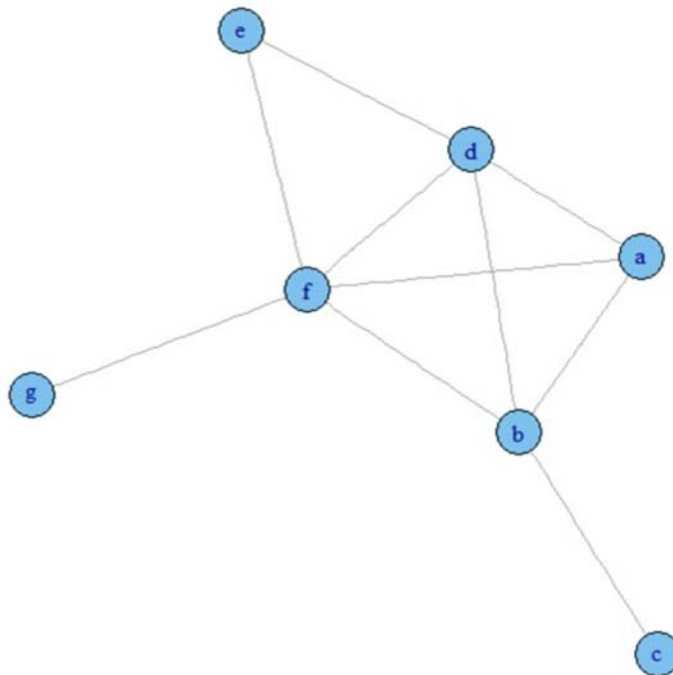
Node-level index	statnet	igraph
degree	degree()	degree()
betweenness	betweenness()	betweenness()
power (Bonacich 1987)	bonpow()	bonpcw()
closeness	closeness()	closeness()
eigenvector centrality	evcent()	evcent()
flow betweenness	flowbet()	N/A
graph centrality (Harary)	graphcent()	N/A
information centrality	infocent()	N/A
load centrality	loadcent()	N/A
prestige	prestige()	N/A
stress centrality	stresscent()	N/A
reach	see codes below	see codes below
distance-weighted reach	see codes below	see codes below
alpha centrality	N/A	alpha centrality()
Kleinberg's authority score	N/A	authority.score()
Kleinberg's hub score	N/A	hub.score()

شکل ۷-۱۳: برخی از توابع مربوط به تحلیل شبکه‌های اجتماعی

مصورسازی شبکه‌ها

در این بخش به چگونگی مصورسازی داده‌های مربوط به تحلیل شبکه‌ای در R می‌پردازیم. در قطعه کد زیر نحوه ترسیم گراف مربوط به ماتریس مجاورت زیر را در بسته igraph مشاهده می‌نمایید.

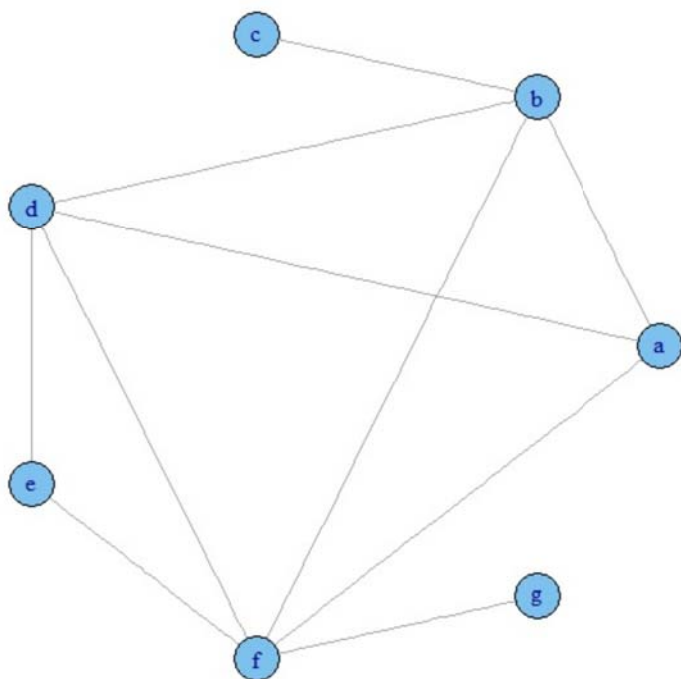
```
> file=read.csv(file.choose(),header=TRUE,row.names=1,
,check.names=FALSE)
> matr=as.matrix(file)
> gr=graph.adjacency(matr,mode="undirected",
weighted=NULL,diag=FALSE)
> plot.igraph(gr)
```



شکل ۷-۱۴: گراف ماتریس مجاورت در بسته igraph

در قطعه کد زیر مصورسازی داده‌ها به صورت گراف به صورت دایره‌ای را مشاهده می‌نمایید.

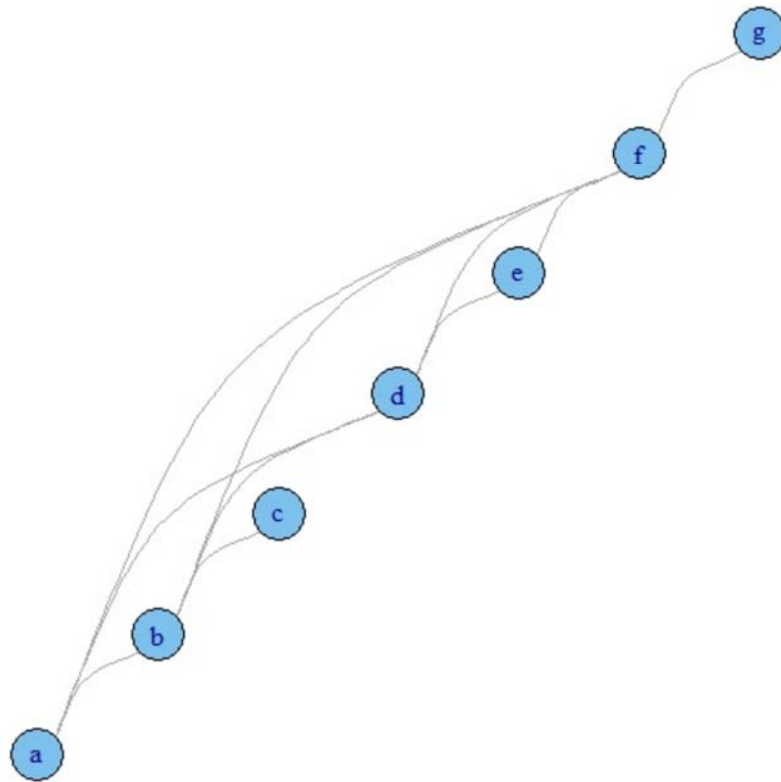
```
> plot.igraph(gr, layout=layout.circle)
```



شکل ۷-۱۵: گراف ماتریس مجاورت در بسته igraph به صورت دایره‌ای

قطعه کد زیر داده‌های مذکور را به صورت منحنی ترسیم می‌نماید.

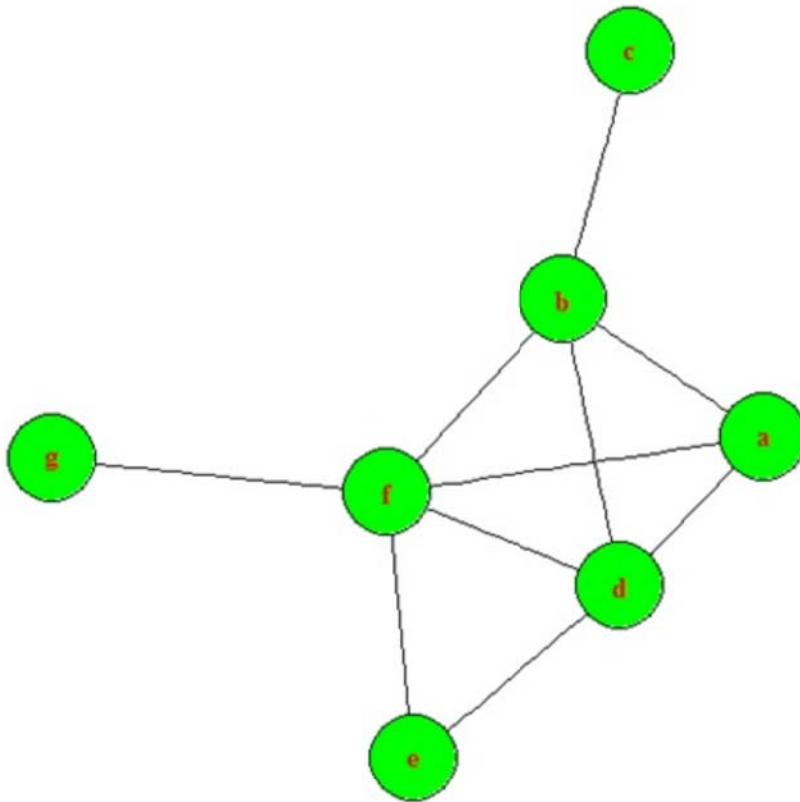
```
> lvar=matrix(c(1,2,3,4,5,6,7, 1,2,3,4,5,6,7),ncol=2)  
> plot.igraph(gr,layout=lvar,edge.curved=TRUE)
```



شکل ۷-۱۶: گراف ماتریس مجاورت در بسته igraph به صورت منحنی

برخی از مشخصات مربوط به گره‌ها، رنگ‌بندی، فونت و دیگر موارد را می‌توان در گراف تصویر شده تغییر داد که برخی از این تغییرات را در شکل زیر مشاهده می‌نمایید.

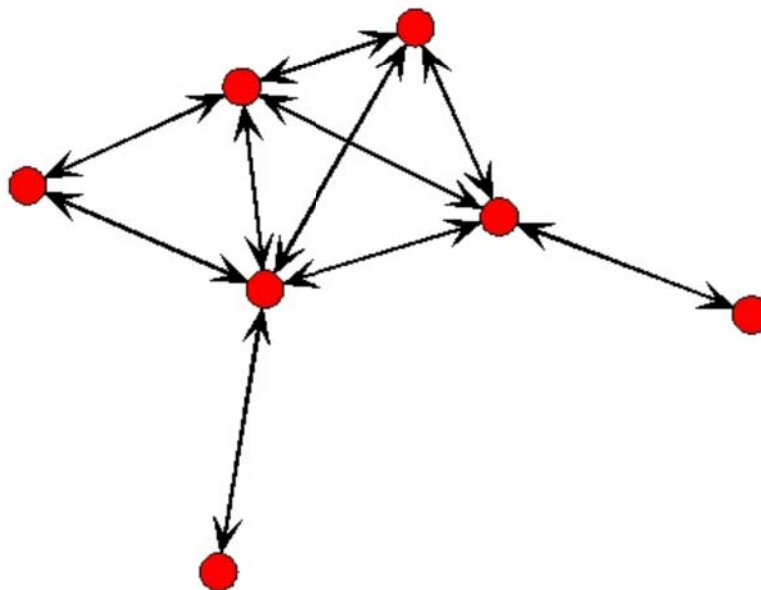
```
> plot.igraph(gr, vertex.label=V(gr)$name, vertex.size=25  
+ , vertex.label.color="red", vertex.label.font=2  
+ , vertex.color="green", edge.color="black")
```



شکل ۷-۱۷: گراف ماتریس مجاورت در بسته igraph

علاوه بر بسته igraph از بسته statnet نیز می‌توان استفاده نمود. در شکل زیر ترسیم گراف با استفاده از ماتریس مجاورت و در قالب بسته statnet مشاهده می‌کنید. شکل زیر گراف مصور شده را با یال‌های دوطرفه نشان می‌دهد.

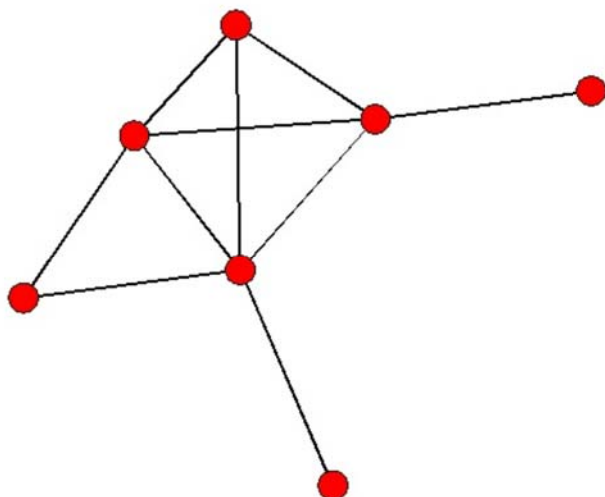
```
> library(statnet)
> file=read.csv(file.choose(),header=TRUE,row.names=1,check.names=FALSE)
> matr=as.matrix(file)
> netw=network(matr,matrix.type="adjacency",directed=FALSE)
> gplot(netw)
```



شکل ۷-۱۸ : گراف ماتریس مجاورت در بسته statnet

قطعه کد زیر جهت یال ها را حذف می نماید.

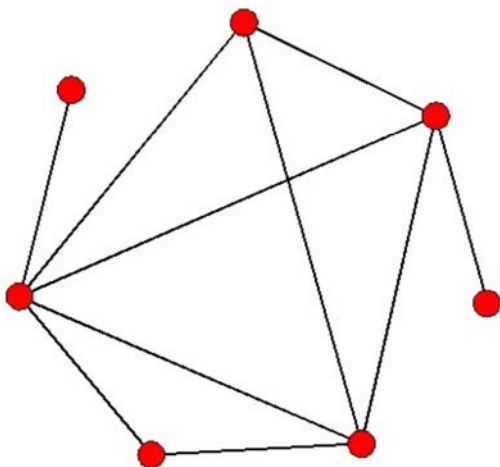
```
> gplot (netw, gmode="graph")
```



شکل ۷-۱۹: حذف جهت یال ها در گراف ماتریس مجاورت در بسته statnet

قطعه کد زیر گراف مصور شده را به قالب دایره ای ترسیم می نماید.

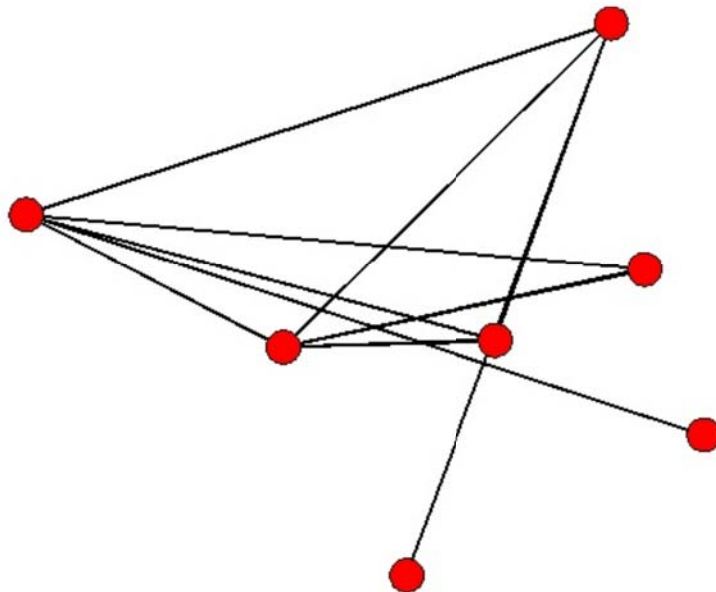
```
> gplot (netw, gmode="graph", mode="circle")
```



شکل ۷-۲۰: گراف ماتریس مجاورت در بسته statnet در قالب دایره ای

در شکل زیر ترسیم گراف مذکور را به صورت چندبعدی می بینید.

```
> gplot(netw, gmode="graph", mode="mds")
```



شکل ۷-۲۱: گراف ماتریس مجاورت در بسته statnet در قالب چندبعدی

مصورسازی شبکه ها (با استفاده از مشخصه های گره)

در برخی موارد از ترسیم گره نیازمند لیستی از خصوصیات هستیم که با هر گره ارتباط دارد. به عنوان مثال در مجموعه داده زیر که به صورت یک لیست از خصوصیات می باشد نمونه ای از این داده ها را مشاهده می نمایید. این مجموعه داده که در قالب CSV ذخیره شده است دارای دو ستون مشخصه و جنسیت می باشد.

	A	B
1	ID	type
2	a	F
3	b	F
4	c	F
5	d	M
6	e	M
7	f	M
8	g	M

شکل ۷-۲۲: قالب داده ای لیستی از خصوصیات

هر کدام از این مشخصه ها که دارای صفت جنسیت می باشند را ابتدا باید به صورت یک ماتریس مجاورت وارد نمود. در قطعه کد زیر ورود این ماتریس مجاورت به نرم افزار را مشاهده می نمایید.

	A	B	C	D	E	F	G	H
1		a	b	c	d	e	f	g
2	a		0	1	0	1	0	1
3	b		1	0	1	1	0	1
4	c		0	1	0	0	0	0
5	d		1	1	0	0	1	1
6	e		0	0	0	1	0	1
7	f		1	1	0	1	1	0
8	g		0	0	0	0	1	1

شکل ۷-۲۳: قالب داده ای ماتریس مجاورت

```

> library(igraph)
> file=read.csv(file.choose(),header=TRUE,
  row.names=1,check.names=FALSE)

> matr=as.matrix(file)
> netw=graph.adjacency(matr,mode="undirected",
  weighted=NULL,diag=FALSE)

> V(netw)$name
[1] "a" "b" "c" "d" "e" "f" "g"

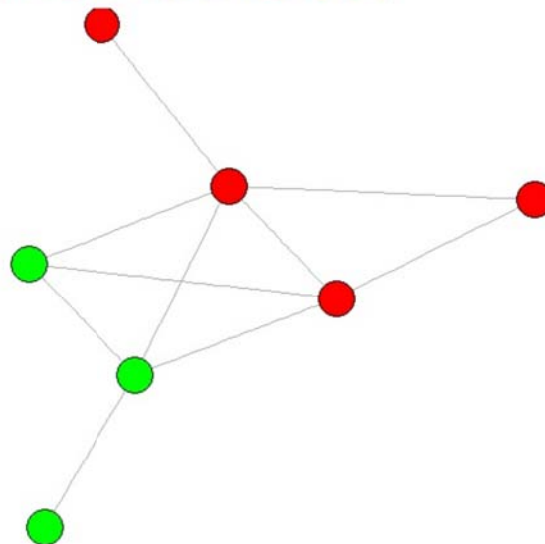
```

با استفاده از کد زیر می‌توان لیست مربوط به خصوصیات را وارد نموده و سپس گره‌هایی که دارای جنسیت مشخص می‌باشند با رنگ‌بندی از یکدیگر جدا نمود.

```

> var=read.csv(file.choose())
> V(netw)$type=as.character(var$type
  [match(V(netw)$name,var$ID)])
> V(netw)$type
[1] "F" "F" "F" "M" "M" "M" "M"
> V(netw)$color=V(netw)$type
> V(netw)$color=gsub("F","green",V(netw)$color)
> V(netw)$color=gsub("M","red",V(netw)$color)
> plot.igraph(netw,vertex.label=NA,
  layout=layout.fruchterman.reingold)

```



شکل ۷-۲۴: ترسیم گراف ماتریس مجاورت با خصوصیات مشترک

مصورسازی شبکه ها (یال های وزن دار)

در بسیاری از موارد در گراف ها شامل یال های دارای وزن هستیم. به عنوان مثال همان طور که در شکل زیر می بینید در ماتریس مجاورت یال ها دارای اعدادی به غیر از صفر و یک می باشند.

	A	B	C	D	E	F	G	H
1		a	b	c	d	e	f	g
2	a		0	4	0	1	0	9
3	b		4	0	2	12	0	1
4	c		0	2	0	0	0	0
5	d		1	12	0	0	1	4
6	e		0	0	0	1	0	8
7	f		9	1	0	4	8	0
8	g		1	0	0	0	0	6

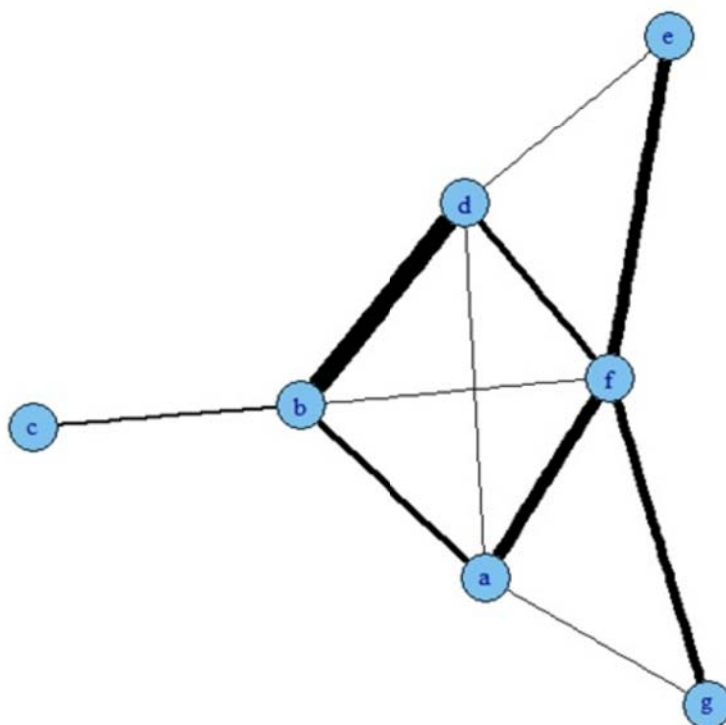
شکل ۷-۲۵ : مجموعه داده ماتریس مجاورت با یال های وزن دار

در قطعه کد زیر نحوه ورود این ماتریس مجاورت به نرم افزار را مشاهده می نمایید.

```
> library(igraph)
> file=read.csv(file.choose(),header=TRUE,
  row.names=1,check.names=FALSE)
> matr=as.matrix(file)
> netw=graph.adjacency(matr,mode="undirected",
  weighted=TRUE,diag=FALSE)
> summary(netw)
IGRAPH UNW- 7 11 --
attr: name (v/c), weight (e/n)
> E(netw)$weight
[1] 4 1 9 1 2 12 1 1 4 8 6
```


ترسیم گراف مصورسازی شده این ماتریس مجاورت با قطعه کد زیر امکان پذیر می باشد.

```
> plot.igraph(netw, vertex.label=V(netw)$name,
+ layout=layout.fruchterman.reingold,
+ edge.color="black", edge.width=E(netw)$weight)
```



شکل ۷-۲۶: گراف ماتریس مجاورت با یال های وزن دار

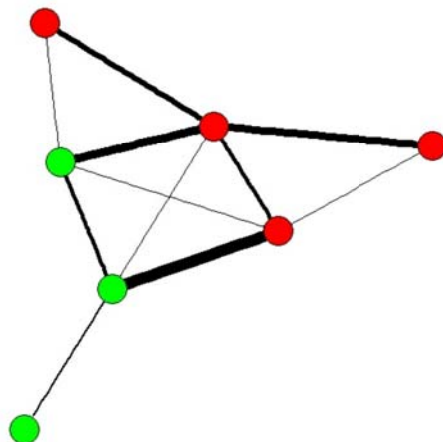
در این قسمت می توان ماتریس مجاورت داده ها را با لیست خصوصیت های هر کدام از آیتم ها ترکیب نمود

	A	B
1	ID	type
2	a	F
3	b	F
4	c	F
5	d	M
6	e	M
7	f	M
8	g	M

شکل ۷-۲۷: لیست خصوصیت مربوط به ماتریس مجاورت با یال های وزن دار

در قطعه کد زیر ابتدا ماتریس مجاورت و در چند خط بعدی خصوصیت ها شامل ID و جنسیت به نرم افزار وارد شده است. گره های دارای رنگ مشابه دارای جنسیت مشابه می باشند.

```
> library(igraph)
> file=read.csv(file.choose(),header=TRUE,row.names=1,check.names=FALSE)
> matr=as.matrix(file)
> netw=graph.adjacency(matr,mode="undirected",weighted=TRUE,diag=FALSE)
> var=read.csv(file.choose())
> V(netw)$type=as.character(var$type[match(V(netw)$name,var$ID)])
> V(netw)$color=V(netw)$type
> V(netw)$color=gsub("F","green",V(netw)$color)
> V(netw)$color=gsub("M","red",V(netw)$color)
> plot.igraph(netw,vertex.label=NA,layout=layout.fruchterman.reingold,
+ edge.color="black",edge.width=E(netw)$weight)
```



شکل ۷-۲۸: گراف ماتریس مجاورت با یال های وزن دار به همراه خصوصیت مشترک

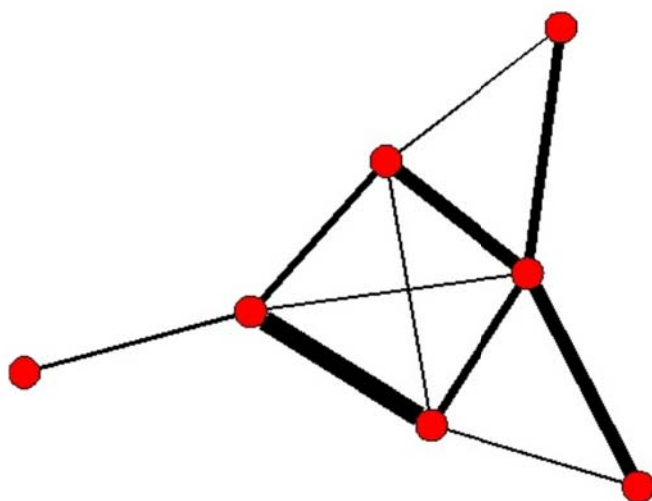
نحوه ورود ماتریس مجاورت وزن دار و ترسیم گراف مصورسازی شده این ماتریس با استفاده از بسته statnet در قطعه کد زیر قابل مشاهده می باشد.

```
> library(statnet)
> file=read.csv(file.choose(),header=TRUE,
+ row.names=1,check.names=FALSE)

> matr=as.matrix(file)
> netw=network(matr,matrix.type="adjacency",directed=FALSE,
+ ignore.eval=FALSE,names.eval="value")

> summary(netw)
Network attributes:
  vertices = 7
  directed = FALSE
  hyper = FALSE
  loops = FALSE
  multiple = FALSE
  bipartite = FALSE
total edges = 11
  missing edges = 0
  non-missing edges = 11
density = 0.5238095
Vertex attributes:
vertex.names:
  character valued attribute
  7 valid vertex names
Edge attributes:
value:
  numeric valued attribute
  attribute summary:
    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 1.000   1.000   4.000   4.455   7.000  12.000
Network adjacency matrix:
  a b c d e f g
a 0 1 0 1 0 1 1
b 1 0 1 1 0 1 0
c 0 1 0 0 0 0 0
d 1 1 0 0 1 1 0
e 0 0 0 1 0 1 0
f 1 1 0 1 1 0 1
g 1 0 0 0 0 1 0
```

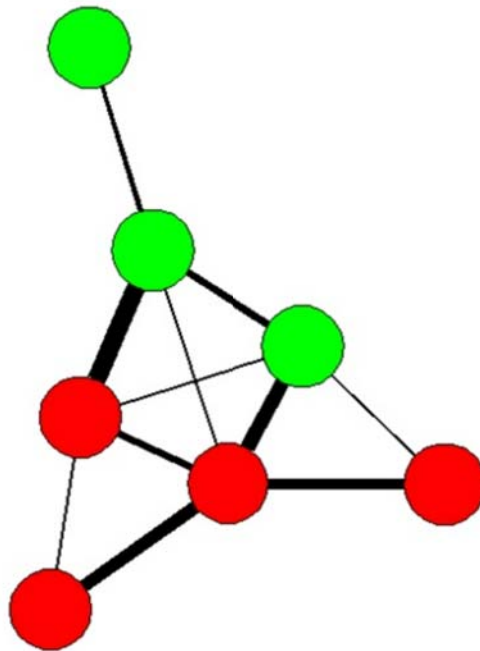
```
> w=as.sociomatrix(netw,"value")
> w
  a b c d e f g
a 0 4 0 1 0 9 1
b 4 0 2 12 0 1 0
c 0 2 0 0 0 0 0
d 1 12 0 0 1 4 0
e 0 0 0 1 0 8 0
f 9 1 0 4 8 0 6
g 1 0 0 0 0 6 0
> gplot(netw,gmode="graph",edge.lwd=netw$e$'value')
```



شکل ۷-۲۹: گراف ماتریس مجاورت با یال های وزن دار با استفاده از بسته statnet

ترسیم گراف مربوط به ماتریس مجاورت به انضمام خصوصیات هر کدام از گره‌ها (جنسیت) با استفاده از بسته statnet را در ادامه مشاهده می‌نمایید.

```
> library(statnet)
> file=read.csv(file.choose(),header=TRUE,row.names=1,check.names=FALSE)
> matr=as.matrix(file)
> netw=network(matr,matrix.type="adjacency",
+ directed=FALSE,ignore.eval=FALSE, names.eval="value")
> var=read.csv(file.choose())
> netw$v$type=as.character(var$type[match(netw$v$vertex.names',var$ID)])
> netw$v$color=netw$v$type
> netw$v$color=gsub("F","green",netw$v$color')
> netw$v$color=gsub("M","red",netw$v$color')
> gplot(netw,gmode="graph",mode="fruchtermanreingold", edge.lwd=netw$e$value',
+ vertex.col=netw$v$color',vertex.cex=2.8,vertex.sides=18)
```

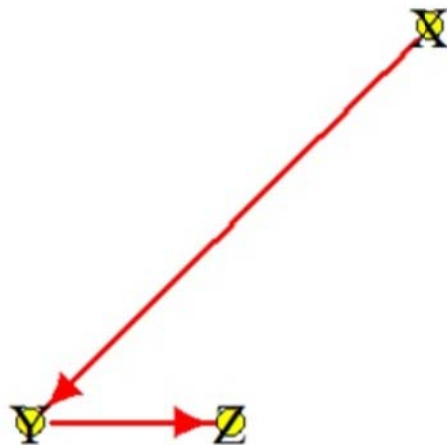


شکل ۷-۳۰: گراف ماتریس مجاورت با یال‌های وزن‌دار با استفاده از بسته statnet به همراه خصوصیت مشترک

مصورسازی شبکه ها (شبکه های جهت دار)

در این بخش به گراف هایی می پردازیم که دارای یال های جهت دار می باشند. می توان گره ها و یال های جهت دار را به صورت دستی وارد نمود. در کد زیر علامت $+$ به معنای ورود یال از یک گره به گره دیگر می باشد.

```
> library(igraph)
> graph=graph.formula(X--+Y,Y--+Z)
> layout=matrix(c(3,2, 1,0, 2,0),byrow=TRUE,nrow=3)
> plot.igraph(graph,layout=layout,edge.color="red",
+ vertex.color="yellow", vertex.label=c("X","Y","Z"),vertex.size=15,
+ vertex.label.cex=2, vertex.label.color="black",edge.width=3,
+ edge.arrow.size=1.2,edge.arrow.width=1.2)
```



شکل ۷-۳۱: گراف دارای یال های جهت دار

این شکل گراف معادل ماتریس مجاورت زیر می‌باشد.

	X	Y	Z
X	۰	۱	۰
Y	۰	۰	۱
Z	۰	۰	۰

شکل ۷-۳۲: ماتریس مجاورت معادل با گراف

یا به صورت جدول گره به شکل زیر نشان داده می‌گردد.

Node ۱	Node ۲
X	Y
Y	Z

شکل ۷-۳۳: جدول گره‌ها معادل با گراف

روشی ساده برای ترسیم گراف مستقیم و وزن دار:

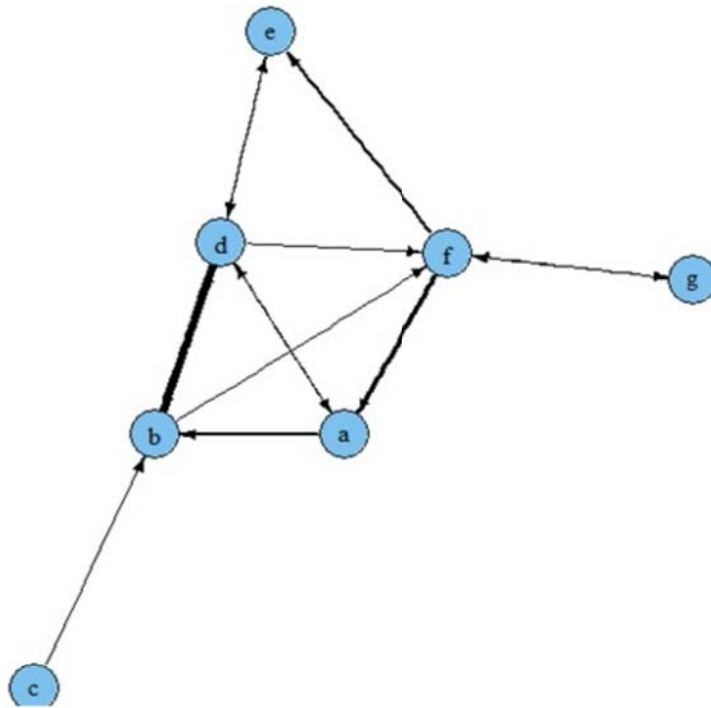
به عنوان مثال ماتریس مجاورت دارای وزن در شکل زیر را به صورت مجموعه داده‌ای با فرمت CSV ذخیره گردیده است را مشاهده نمایید.

	A	B	C	D	E	F	G	H
1		a	b	c	d	e	f	g
2	a		0	4	0	1	0	9
3	b		4	0	2	12	0	1
4	c		0	2	0	0	0	0
5	d		1	12	0	0	1	4
6	e		0	0	0	1	0	8
7	f		9	1	0	4	8	0
8	g		1	0	0	0	0	6

شکل ۷-۳۴: مجموعه داده‌ای ماتریس مجاورت دارای یال‌های وزن دار

برای ترسیم گراف این ماتریس وزن دار از قطعه کد زیر استفاده می‌نماییم.

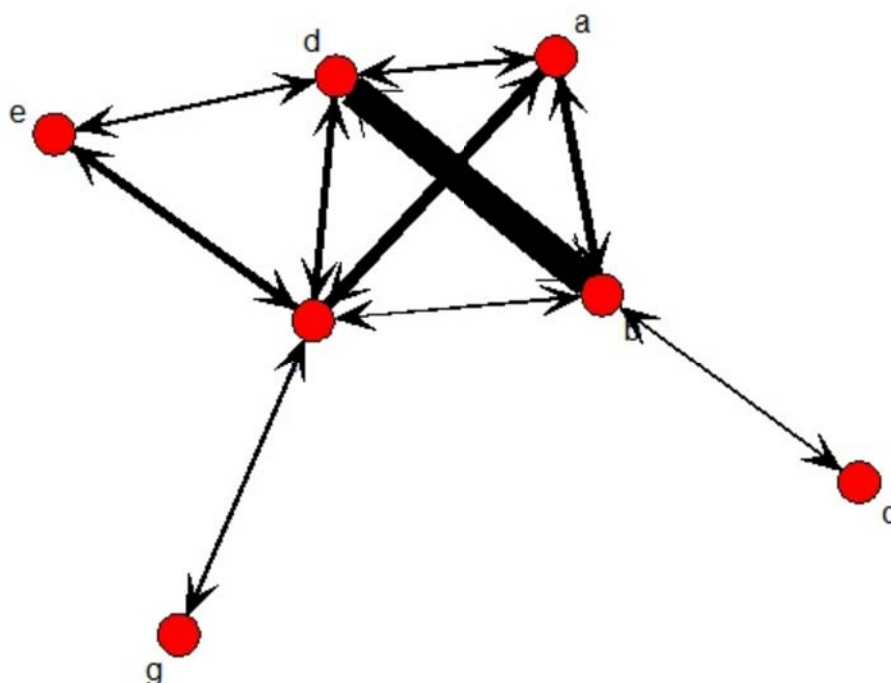
```
> library(igraph)
> file=read.csv(file.choose(),header=TRUE,row.names=1,
+ check.names=FALSE)
> matr=as.matrix(file)
> netw=graph.adjacency(matr,mode="directed",
+ weighted=TRUE,diag=FALSE)
> plot.igraph(netw,vertex.label=V(netw)$name,
+ layout=layout.fruchterman.reingold,vertex.label.color="black",
+ edge.color="black",edge.width=E(netw)$weight/3,
+ edge.arrow.size=0.5)
```



شکل ۷-۳۵: گراف ماتریس مجاورت دارای یال‌های وزن دار

معادل این کد در بسته statnet کد زیر می‌باشد.

```
> library(statnet)
> file=read.csv(file.choose(),header=TRUE,row.names=1,
+ check.names=FALSE)
> matr=as.matrix(file)
> netw=network(matr,matrix.type="adjacency",
+ directed=FALSE,ignore.eval=FALSE, names.eval="value")
> w=as.sociomatrix(netw,"value")
> w
      a      b      c      d      e      f      g
a 0  4.0 0.0  1.0 0.0 5.0 0
b 4  0.0 0.5 15.0 0.0 0.5 0
c 0  0.5 0.0  0.0 0.0 0.0 0
d 1 15.0 0.0  0.0 1.0 2.5 0
e 0  0.0 0.0  1.0 0.0 3.5 0
f 5  0.5 0.0  2.5 3.5 0.0 1
g 0  0.0 0.0  0.0 0.0 1.0 0
> gplot(netw,gmode="digraph",edge.lwd=netw$e$'value',
+ label=netw$v$'vertex.names', arrowhead.cex=1)
```

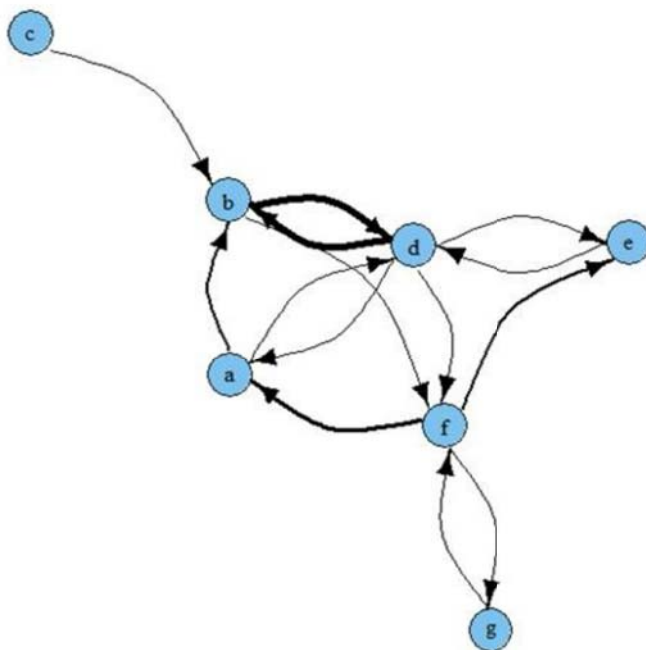


شکل ۷-۳۶: گراف ماتریس مجاورت دارای یال‌های وزن‌دار در بسته statnet

ترسیم شبکه ها با یال های چندگانه و وزن دار :

در صورتی که دارای یال های وزن دار و چند طرفه در گراف باشیم از کد زیر استفاده می نماییم.

```
> library(igraph)
> files=read.csv(file.choose(),header=TRUE,
+ row.names=1,check.names=FALSE)
> matr=as.matrix(files)
> netw=graph.adjacency(matr,mode="directed",
+ weighted=TRUE,diag=FALSE)
> plot.igraph(netw,vertex.label=V(netw)$name,
+ layout=layout.fruchterman.reingold,
+ vertex.label.color="black",edge.color="black",
+ edge.width=E(netw)$weight/3,
+ edge.arrow.size=1,edge.curved=TRUE)
```



شکل ۷-۳۷ : گراف با یال های چندگانه و وزن دار

اجتماع یابی

اجتماع یابی به معنی تشخیص اجتماعاتی از داده‌هاست که بیشترین مشابهت با یکدیگر را دارند و ترسیم آن‌ها توسط گراف که به صورت رنگ‌بندی ساختار این اجتماعات را نمایش می‌دهد. مجموعه داده زیر که با فرمت CSV ذخیره شده است را در نظر بگیرید. ستون سمت چپ هر کدام نوع خاصی محصول می‌باشد و هر کدام از ستون‌های دیگر کشوری که نیازمند آن محصول می‌باشد را ذکر می‌کند. می‌خواهیم گونه‌هایی که تشکیل اجتماع می‌دهند را در این مجموعه داده پیدا نماییم.

	A	B	C	D	E	F
1	type	asia	africa	europa	America	Australia
2	a	0	1	0	0	0
3	b	0	1	0	0	0
4	c	0	0	0	0	0
5	d	0	0	0	0	0
6	e	1	0	0	0	0
7	f	0	0	0	1	0
8	g	0	1	1	0	0
9	h	0	0	0	0	1
10	i	0	0	0	0	0
11	j	1	1	1	1	0
12	k	0	1	1	1	1
13	l	1	0	0	0	0
14	m	0	0	1	0	0
15	n	0	0	0	0	1
16	o	1	0	0	0	0
17	p	0	1	1	1	0
18	q	0	1	0	0	0

شکل ۷-۳۸: مجموعه داده‌ای گونه‌های پرندگان به همراه کشورهای مرتبط با این گونه‌ها

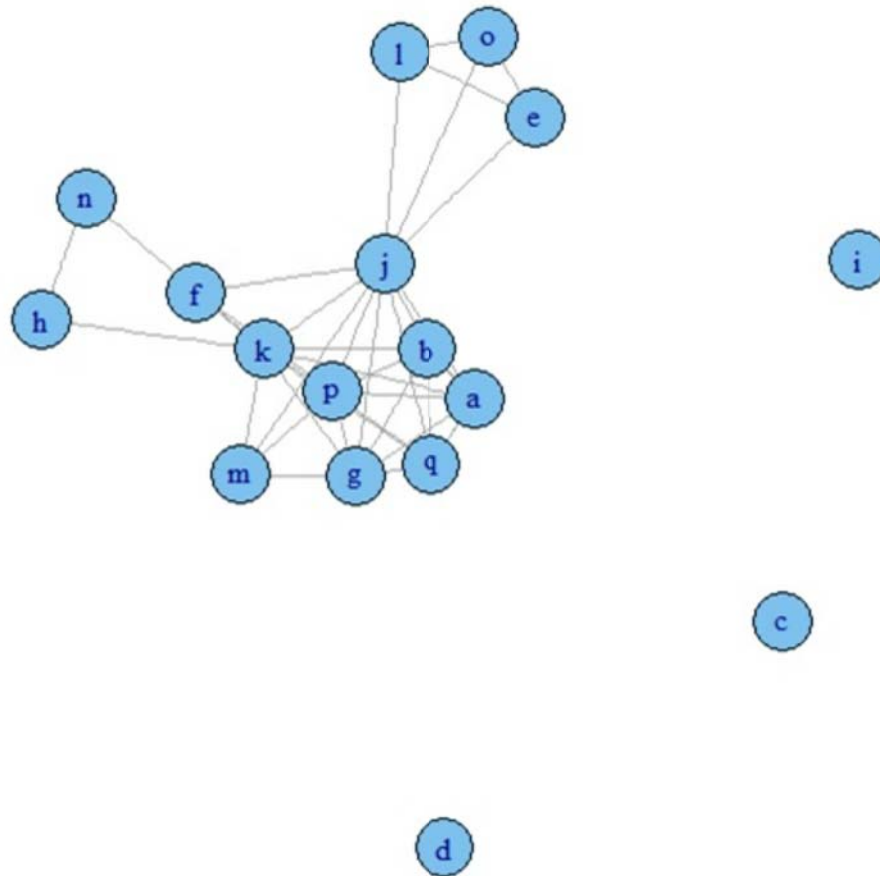
ابتدا این مجموعه داده را با دستور زیر وارد می‌نماییم.

```
> library(igraph)
> var=read.csv(file.choose(),header=TRUE,
  row.names=1,check.names=FALSE)
> var1=as.matrix(var)
> var1
```

	asia	africa	europa	America	Australia
a	0	1	0	0	0
b	0	1	0	0	0
c	0	0	0	0	0
d	0	0	0	0	0
e	1	0	0	0	0
f	0	0	0	1	0
g	0	1	1	0	0
h	0	0	0	0	1
i	0	0	0	0	0
j	1	1	1	1	0
k	0	1	1	1	1
l	1	0	0	0	0
m	0	0	1	0	0
n	0	0	0	0	1
o	1	0	0	0	0
p	0	1	1	1	0
q	0	1	0	0	0

با دستور زیر ساختار اجتماعات این گونه‌ها را ترسیم می‌نماییم.

```
> var2=graph.incidence(var1)
> var3=bipartite.projection(var2)[[1]]
> plot(var3)
```



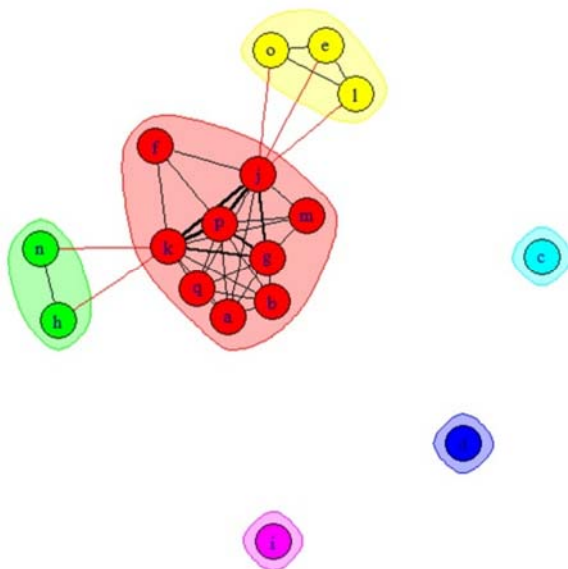
شکل ۷-۳۹: اجتماع یابی مجموعه داده‌ای محصولات

دستور زیر ساختار اجتماعات را با استفاده از الگوریتم walktrap محاسبه نموده و تعداد اجتماعات تشکیل شده و Modularity را به دست می آورد. در ادامه هر کدام از گونه ها را در یک دسته قرار می دهد.

```
> f=walktrap.community(var3)
> f
Graph community structure calculated
with the walktrap algorithm
Number of communities (best split): 6
Modularity (best split): 0.1424858
Membership vector:
a b c d e f g h i j k l m n o p q
1 1 4 5 2 1 1 3 6 1 1 2 1 3 2 1 1
```

ترسیم این اجتماعات به صورت رنگ بندی شده. گره هایی که هم رنگ هستند در یک اجتماع قرار می گیرند.

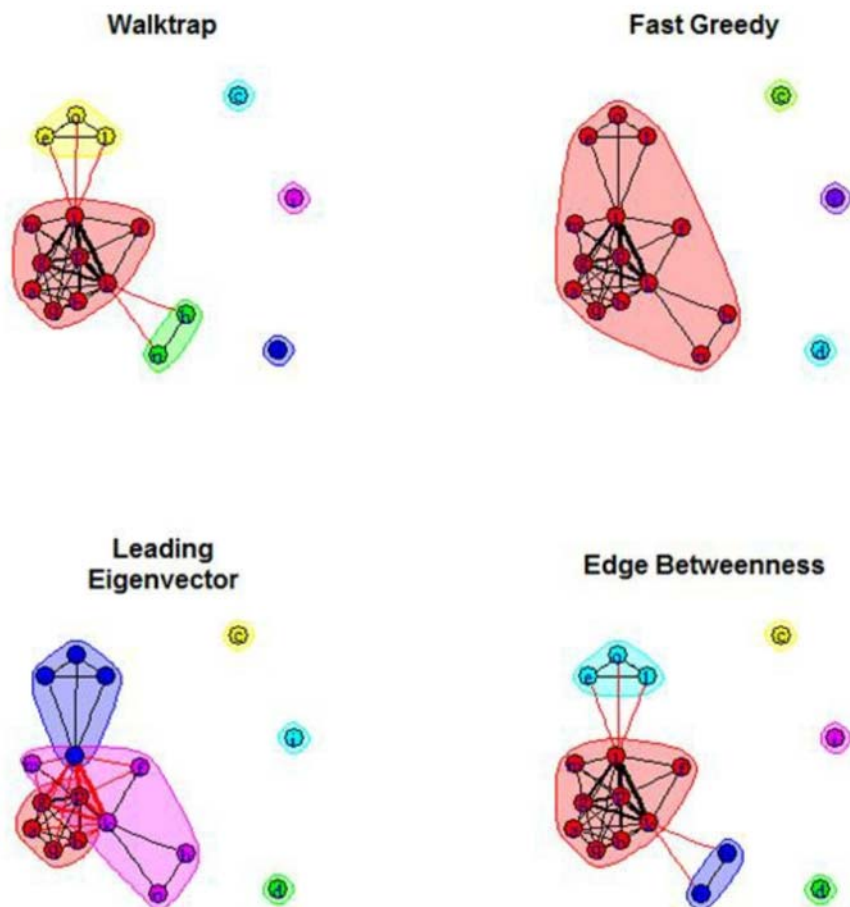
```
> plot(f,var3,edge.width=E(var3)$weight)
```



شکل ۷-۴۰: اجتماع یابی رنگ بندی شده مجموعه داده ای محصولات

در ادامه فرآیند اجتماع یابی را علاوه بر الگوریتم Walktrap با استفاده از الگوریتم‌های دیگری همچون Fast Greedy، Leading Eigenvector، Edge Betweenness نمایش می‌دهد.

```
> f1=fastgreedy.community(var3,weights=E(spp)$weight)
> f2=leading.eigenvector.community(var3)
> f3=edge.betweenness.community(var3,directed=FALSE,weights=E(var3)$weight)
> l=layout.fruchterman.reingold(var3)
> par(mfrow=c(2,2))
> plot(f,var3,layout=l,edge.width=E(var3)$weight,main="Walktrap")
> plot(f1,var3,layout=l,edge.width=E(var3)$weight,main="Fast Greedy")
> plot(f2,var3,layout=l,edge.width=E(var3)$weight,main="Leading
+ Eigenvector")
> plot(f3,var3,layout=l,edge.width=E(var3)$weight,main="Edge Betweenness")
```



شکل ۷-۴۱: اجتماع یابی رنگ‌بندی شده مجموعه داده‌ای محصولات با الگوریتم‌های مختلف

میزان ماجولاریتی را نیز از طریق کد زیر می توان برای هر کدام از الگوریتم های بالا این گونه محاسبه نمود.

```
> modularity(f)
[1] 0.1424858
> modularity(f1)
[1] 0.1255268
> modularity(f2)
[1] 0.1259452
> modularity(f3)
[1] 0.1424858
```

محاسبه مرکزیت ها

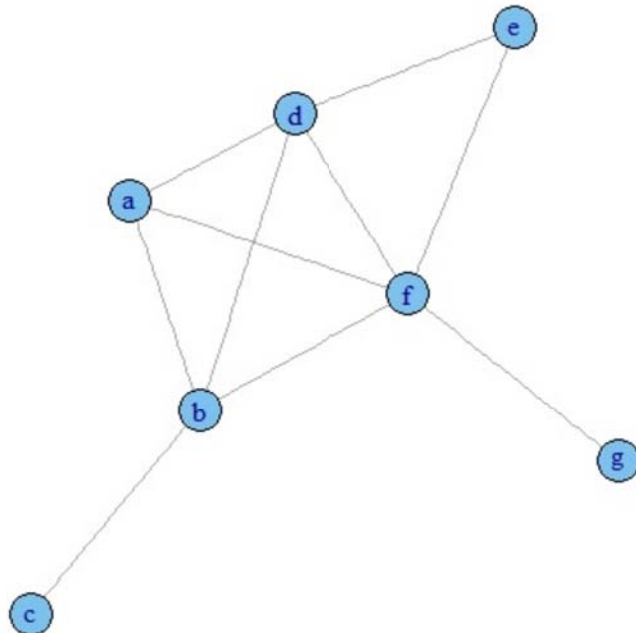
در این قسمت به محاسبه مرکزیت ها در شبکه شامل مرکزیت درجه ای، مرکزیت بینابینی و مرکزیت نزدیکی می پردازیم که توضیحات مربوط به هر کدام در بخش های قبلی ارائه گردیده است. مجموعه داده زیر را در نظر بگیرید و هر کدام از مؤلفه های موجود در سطر و ستون ها را یک موجودیت خاص تصور نمایید.

	a	b	c	d	e	f	g
a	0	1	0	1	0	1	0
b	1	0	1	1	0	1	0
c	0	1	0	0	0	0	0
d	1	1	0	0	1	1	0
e	0	0	0	1	0	1	0
f	1	1	0	1	1	0	1
g	0	0	0	0	0	1	0

شکل ۷-۴۲: نمونه مجموعه داده ورودی برای محاسبه مرکزیت ها

در کد زیر ابتدا مجموعه داده را وارد نرم افزار نموده و سپس تبدیل به ماتریس می نماییم. بعد از آن ماتریس را به گراف تبدیل نموده و در نهایت با دستوری که مشاهده می نمایید مرکزیت های مربوط به هر کدام از مؤلفه ها را استخراج می نماییم.

```
> library(igraph)
> dat=read.csv(file.choose(),header=TRUE,row.names=1,check.names=FALSE)
> m=as.matrix(dat)
> g=graph.adjacency(m,mode="undirected",weighted=NULL,diag=FALSE)
> out <- data.frame(V(g)$name, degree(g),closeness(g),betweenness(g))
> out
  V.g..name degree.g. closeness.g. betweenness.g.
a         a         3  0.11111111          0.0
b         b         4  0.12500000          5.0
c         c         1  0.07692308          0.0
d         d         4  0.12500000          1.5
e         e         2  0.09090909          0.0
f         f         5  0.14285714          6.5
g         g         1  0.08333333          0.0
> plot.igraph(g)
```



شکل ۷-۴۳: گراف مربوط به داده های ورودی

فصل هشتم

متن کاوی

۸-۱- متن کاوی

متن کاوی شاخه‌ای از داده کاوی است. داده کاوی روی پایگاه‌های داده ساخت‌یافته کاوش می‌کند در حالی که در متن کاوی الگوها از متن زبان طبیعی استخراج می‌شوند. پایگاه‌های داده برای پردازش خودکار توسط برنامه‌ها طراحی شده‌اند در حالی که متن برای خواننده شدن توسط مردم نوشته شده است. ما هنوز برنامه‌هایی نداریم که بتوانند به معنای واقعی متن را بخوانند و در آینده نزدیک هم نخواهیم داشت (Hearst, ۲۰۰۳). تحقیق در مورد اعمال روش‌های داده کاوی به متن ساخت‌نیافته به عنوان کشف دانش در متون یا متن کاوی شناخته می‌شود. با این تعریف می‌توان متن کاوی را مصداقی از محتوا کاوی وب در نظر گرفت (Kosala and Blockeel, ۲۰۰۰).



(Dörre et al., ۱۹۹۹)

شکل ۸-۱: انواع داده‌های متنی در یک شرکت

متن کاوی کاربردهای متعددی از جمله بررسی روندهای علمی در علم بیولوژی و هوش تجاری دارد. از آنجا که ۹۰٪ از داده‌های شرکت‌ها با روش‌های معمولی داده کاوی قابل کشف دانش نیستند، انگیزه زیادی برای استفاده از متن کاوی در

صنعت وجود دارد. انواع داده‌های متنی شرکت‌ها در بالا نشان داده شده‌اند (Dörre et al., ۱۹۹۹). روش‌های متن کاوی اغلب همان روش‌های داده کاوی هستند که با تغییراتی برای متون استفاده می‌شوند.

۸-۲- پیش پردازش متون

یکی از مهم‌ترین تفاوت‌های متن کاوی با داده کاوی تراکنشی، نحوه پیش‌پردازش متون می‌باشد. پیش‌پردازش متون از مرا حل زیر تشکیل می‌شود (شکل ۸-۲):



شکل ۸-۲: مراحل پیش‌پردازش متون

۱. **تفکیک هر متن به عنوان، کلمات کلیدی و چکیده:** برای دادن ضریب متفاوت اهمیت هر متن به عنوان، کلمات کلیدی و چکیده تفکیک می‌شود.
۲. **تفکیک به کلمات یا عبارات:** هر کلمه، توالی غیر خالی از کاراکترها با کنار گذاشتن فاصله و نقطه‌گذاری‌ها (کلمات و اعداد) می‌باشد. لازم است برچسب‌های متن مثل دستورات html که برای تعیین فرمت کلمات استفاده می‌شوند حذف شوند. هر کلمه را با یک عدد صحیح ۴ بایتی می‌توان بیان نمود. بنا به انتخاب می‌توان با استفاده از پردازش زبان طبیعی عبارات اسمی مرکب را به دست آورد و به جای عبارات ساده آن‌ها را شمارش نمود.
۳. **حذف کلمات توقف:** افعال عمومی، حروف اضافه و ربط همگی کلمات توقف هستند. این کلمات در تعداد زیادی سند ظاهر شده و چندان برای

جستجو مفید نیستند. بنابراین در نمایه ذخیره نمی شوند تا حجم کاهش و سرعت افزایش یابد.

۴. **مترادفها**: برای پیش گیری از بروز خطا در محاسبه تعداد واژگان، لغات هم معنی و عبارات اختصار یافته^۱ استخراج می گردند.

۵. **ریشه یابی**: ریشه یابی می تواند به روش تحلیل شکل کلمه انجام شود. ریشه یابی موجب افزایش یادآوری ولی کاهش دقت می شود. مثالی از ریشه یابی تبدیل "کلمه+نیم فاصله+یشان" به "کلمه" می باشد. کلمات با الگوریتم پورتر^۲ ریشه یابی می شوند.

۶. **انتخاب مشخصه**: عبارات با توجه به یک آستانه بالا و پایین برای فراوانی فیلتر می شوند.

۷. **ضریب تأثیر برای عبارات به تفکیک بخشها**: طبق نظر خبره ضرایب اهمیت متفاوتی برای عنوان، کلمات کلیدی و چکیده در نظر گرفته می شود.

۸. **وزن دهی به کلمات**: شمارش و وزن دهی به کلمات با مدل فضای برداری انجام می شود.

۹. **نرمال کردن کسینوسی**: طول بردار یکه می شود.

۱۰. **کاهش بعد**: از روش LSI و یا نگاشت تصادفی^۳ استفاده می شود (J. Lin and Gunopulos, ۲۰۰۳). نگاشت تصادفی بسیار سریع تر از LSI بوده و درعین حال دقت آن قابل مقایسه با LSI می باشد (Fradkin and Madigan, ۲۰۰۳). نگاشت تصادفی با ضرب کردن بردارهای اسناد در یک ماتریس تصادفی انجام می شود که در آن بعد خروجی از ورودی کوچکتر است. این فن بین کلمات کمی خطای شباهت تصادفی ایجاد می کند.

^۱ Abbreviation

^۲ Porter

^۳ Random mapping

به طور نظری و عملی نشان داده شده که اگر بعد خروجی به اندازه کافی بزرگ باشد، اثرات تصادفی اثر کمی روی محاسبه شباهت بین اسناد دارند.

بازیابی اطلاعات و مدل فضای برداری

بازیابی اطلاعات^۱، بازیابی خود کار همه اسناد مرتبط و درعین حال بازیابی کمترین تعداد مقدور از اسناد نامرتبط بر اساس پرس و جوی داده شده است. روش های کلاسیک بازیابی اطلاعات از اوایل دهه ۱۹۸۰ برای نشان دادن نتایج جستجو و دسته بندی اسناد متنی، به کار می رفته اند. به دلیل ابعاد زیاد اطلاعات متنی (به تعداد کلمات متن) و لزوم تشخیص ابعاد بارزش، روش های بازیابی متن (بازیابی اطلاعات) با روش های کلاسیک تفاوت هایی دارند. این تفاوت ها در تعریف معیارهای شباهت به جای فاصله و نیز رده بندی نتایج بارز باشند. از روش های متداول بازیابی اطلاعات، مدل فضای برداری^۲ است.

مدل فضای برداری

در مدل فضای برداری، متون به شکل بردارهایی در فضای چندبعدی اقلیدسی نمایش داده می شود (Salton et al., ۱۹۷۵). هر محور این فضا متناظر با یک کلمه یا عبارت است.

فراوانی کلمه (TF)^۳: دفعات رخ دادن کلمه t در متن d یعنی $n(d,t)$ است. در این مقاله فراوانی کلمه نرمال نشده است زیرا نرمال سازی کسینوسی طول بردار کلمات که در ادامه توضیح داده می شود به طور طبیعی این منظور را نیز برآورده می کند.

فراوانی معکوس متن (IDF)^۴: همه محورهای فضای برداری به اندازه هم مهم نیستند. ضریب IDF موجب کاهش اهمیت کلماتی می گردد که در تعدادی

^۱ Information Retrieval (IR)

^۲ Vector Space Model

^۳ Term Frequency

^۴ Inverse document frequency

زیادی از متون وجود داشته و بنابراین ارزش کمی دارند. اگر N تعداد کل متون و DF_t تعداد متون دارای کلمه t باشد، آنگاه یک شکل متداول وزن دهی IDF بر اساس زیر است:

$$IDF(t) = -\log_2 \left(\frac{DF_t}{N} \right)$$

اگر $DF_t < N$ آنگاه کلمه t دارای ضریب مقیاس IDF بزرگی خواهد بود و بالعکس.

TF و IDF باهم به طوری طبیعی ترکیب می‌شوند تا مدل فضای برداری را تشکیل دهند که در آن مختصات متن d در محور t طبق رابطه زیر داده می‌شود:

$$w_t = TF(d, t) IDF(t)$$

برای اندازه‌گیری تشابه بین دو سند متنی \vec{d}_1 و \vec{d}_2 ، کسینوس زاویه بین \vec{d}_1 و \vec{d}_2 طبق رابطه زیر محاسبه می‌شود.

$$\text{sim}(d_1, d_2) = \frac{\vec{d}_1 \bullet \vec{d}_2}{|\vec{d}_1| \times |\vec{d}_2|} = \frac{\sum_{i=1}^t w_{i,1} \times w_{i,2}}{\sqrt{\sum_{i=1}^t w_{i,1}^2} \times \sqrt{\sum_{i=1}^t w_{i,2}^2}}$$

این نحوه اندازه‌گیری تشابه معادل نرمال کردن کسینوسی طول بردار متون است که در آن وزن کلمات هر متن بر طول بردار آن کلمات آن متن تقسیم می‌شوند.

به‌طور خلاصه، یک سیستم IR مبتنی بر TFIDF ابتدا ماتریس معکوسی با اطلاعات TF و IDF می‌سازد و با دادن یک پرس‌وجو (بردار) به آن، تعدادی از بردارهای متن را که شبیه‌تر به پرس‌وجو هستند، فهرست می‌کند. پرس‌وجوهای کلمه‌ای، دقیق نیستند یعنی نمی‌توان برای شمول یا عدم شمول یک پاسخ

تصمیم قطعی گرفت. راه ایمن‌تر، رتبه‌بندی^۱ هر متن با توجه به ارضای نیاز اطلاعاتی کاربر و ربط^۲، مرتب کردن امتیاز به‌طور نزولی و نمایش نتایج به شکل یک لیست رتبه‌بندی شده است.

برای انجام متن کاوی ابتدا با استفاده از مدل فضای برداری ماتریس TF.IDF تشکیل شده و سپس با روش‌های متداول داده کاوی، از روی داده‌ها مدل ساخته می‌شود.

نمایه‌سازی معنایی پنهان

نمایه‌سازی معنایی پنهان^۳ (LSI) روشی برای یافتن پیوندهای غیرمستقیم در بین اسناد و کلمات است. این روش از فن تجزیه مقدار منفرد^۴ (SVD) استفاده می‌کند (Bellegarda, ۲۰۰۷; Berry et al., ۱۹۹۵; Dumais et al., ۱۹۸۸).

فن LSI از روش تجزیه مقدار مفرد ماتریس عبارت-سند استفاده می‌کند تا تلازمات^۵ معنایی پنهان را به دست آورد. فن LSI به علت توانایی آن برای کار با عبارات عموماً مشکل‌زای پرس و جوی حاوی کلمات مترادف^۶ و دارای معانی متعدد^۷ معروف می‌باشد. فن SVD روش‌های LSI را قادر می‌سازد تا به طور ذاتی اسناد و عبارات را به مفاهیم خوشه‌بندی کند (Berry et al., ۱۹۹۹). به‌طور مثال کلمات ماشین، خودرو و وسیله نقلیه ممکن است در یک خوشه گروه‌بندی شوند درحالی‌که کلمه چند معنی ماشین ممکن است با توجه به معانی مختلف آن (خودرو، دستگاه) تقسیم شود. البته قدرت تفکیک LSI محدود به مجموعه کوچک اسناد است زیرا محاسبه SVD (که معادل یافتن بردارهای ویژه می‌باشد) و ذخیره‌سازی آن برای ماتریس بزرگ عبارت-سند پرهزینه است.

^۱ Ranking

^۲ Relevance

^۳ Latent Semantic Indexing (LSI)

^۴ Singular Value Decomposition

^۵ Associations

^۶ Synonyms

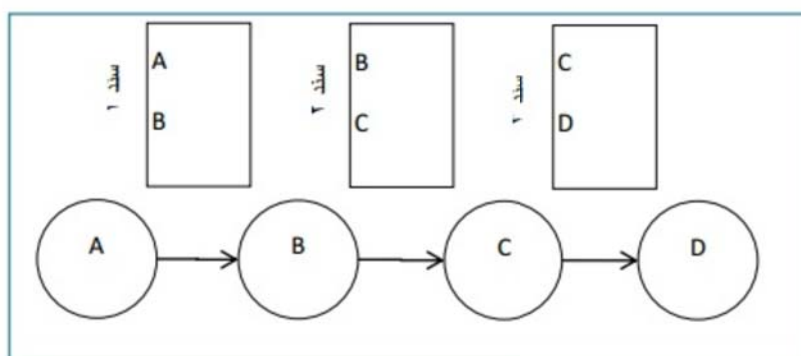
^۷ Polysems

مجموعه‌های بزرگ سند مانند صفحات وب WWW به کلی خارج از حوزه LSI است.

هم‌رخدادی

می‌توان از مقادیر ویژه استفاده کرده و ابعادی را که بیشترین تأثیر را از ضرب یک ماتریس در یک بردار می‌پذیرند برجسته نمود. در LSI ساختار پنهانی که در آن اصطلاحات، اسناد و پرس‌وجوها نهفته‌اند آشکار می‌شود. این روش تفاوت مهمی با مدل‌های فضای برداری دارد. در مدل فضای برداری هر اصطلاح، بعدی از فضای اصطلاح محسوب شده و اسناد و پرس‌وجوها به شکل بردارهایی در این فضای اصطلاح بیان می‌شوند. در این فضا اصطلاحات مستقل از هم فرض می‌شوند. این وضعیت با دانش عمومی مغایر است زیرا در زبان شفاهی و کتبی اصطلاحات از نظرهای مختلف به هم وابسته‌اند. متداول‌ترین وابستگی کلمات مترادف و کلمات چندشکلی هستند.

روش LSI با بیان اسناد، اصطلاحات و پرس‌وجوها در فضای یکسان بهتر از مدل فضای برداری عمل می‌کند. این روش مشکل هم‌رخدادی درجات بالا (بیشتر از یک) مانند هم‌رخدادی انتقالی را حل می‌کند. هم‌رویدادی انتقالی بیان می‌کند که لازم نیست اصطلاحات برای مربوط بودن به یک سند واقعاً در آن سند باشند. مثالی از هم‌رخدادی انتقالی، کلمات مترادف هستند. کلمات مترادف معمولاً باهم روی نمی‌دهند ولی در زمینه یکسانی روی می‌دهند که با اندازه‌گیری درجات بالاتر هم‌رخدادی قابل بیان است (J. Allan et al., ۲۰۰۲; Holzman et al., ۲۰۰۴; Kontostathis and Pottenger, ۲۰۰۳, ۲۰۰۶; Kontostathis et al., ۲۰۰۴; Kontostathis, ۲۰۰۳, ۲۰۰۷; Newo, ۲۰۰۵). این موضوع در ادامه نشان داده شده است.



شکل ۸-۳: ردگیری درجه هم‌رخدادی عبارات در اسناد

عبارت A و C در یک سند نیستند ولی با یک واسطه مشترک یعنی عبارت B به هم پیوند دارند. پس دارای هم‌رخدادی درجه ۲ هستند. هم‌رخدادی درجه ۱ شامل $\{A, B\}, \{B, C\}, \{C, D\}$ است. هم‌رخدادی درجه ۲ شامل $\{A, C\}, \{B, D\}$ و هم‌رخدادی درجه ۳ شامل $\{A, D\}$ می‌باشد.

مدل‌سازی زبانی

اخیراً مدل‌های آماری دیگری علاوه بر LSI متداول شده‌اند که بر اساس مدل‌سازی زبانی کار می‌کنند. مدل pLSI و LDA از مدل‌های معروف این حوزه می‌باشند. مدل LDA، یک شبکه بیزی است که یک سند را به کمک ترکیبی از عناوین تولید می‌کند. در فرایند مولد^۱ آن، برای هر سند d ، یک توزیع چندجمله‌ای θ روی عناوین به طور تصادفی از یک دیریکله با پارامتر α نمونه‌گیری شده و سپس برای تولید هر کلمه، عنوان z از این توزیع عنوان انتخاب شده و کلمه w با نمونه‌گیری تصادفی از توزیع چندجمله‌ای ϕ_z مخصوص عنوان تولید می‌شود.

این مدل‌های زبانی توسعه یافته‌اند تا ارتباط مباحث و افراد را نشان دهند. مدل نویسنده-گیرنده-عنوان یکی از این مدل‌های توسعه‌یافته می‌باشد (McCallum et al., ۲۰۰۵, ۲۰۰۴).

^۱ Generative

کشف و ردیابی عنوان

کشف و ردیابی عنوان^۱ (TDT) به سازمان‌دهی مبتنی بر واقعه اخبار رسانه‌ای می‌پردازد. انگیزه تحقیق در TDT فراهم کردن فناوری کلیدی برای نظارت بر اخبار رسانه‌های جمعی و اعلام اخبار جدید و جالب به تحلیلگر می‌باشد. بدین طریق، تحلیلگر می‌تواند از پس حجم عظیم اطلاعات دریافتی روزانه برآید (J. Allan, ۲۰۰۲). هرچند TDT در باطن مشابه کارهای قبلی انجام شده در زمینه فیلتر کردن و بازیابی اطلاعات می‌باشد، ولی دارای تعریف مشخصی برای “درباره بودن” می‌باشد.

در TDT یک مبحث یا عنوان^۲، مجموعه‌ای از داستان‌های خبری می‌باشد که توسط واقعه‌ای قابل توجه در دنیای واقعی به هم به طور قوی مرتبط شده‌اند. هر واقعه^۳ را می‌توان یک موضوع^۴ دانست ولی هر موضوعی لزوماً یک واقعه نیست. مثلاً، گل‌هایی که در سایه رشد می‌کنند یک موضوع است ولی واقعه نیست. ادراک عنوان مبتنی بر واقعه، محدودتر از عنوان مبتنی بر موضوع است زیرا بر اساس واقعه راه‌انداز آن بنا می‌شود. تفاوت دیگر TDT با سازمان‌دهی مبتنی بر موضوع، طبیعت زمانی عناوین آن می‌باشد. این عناوین زمانی، حتی می‌توانند در طول زمان متحول شده و شامل وقایع مرتبطی شوند که به‌ظاهر روی همان موضوع نیستند. داستان‌های خبری موردعلاقه TDT هم از روزنامه‌ها و هم از منابع صوتی می‌آیند. معمولاً ابتدا منابع صوتی به متن تبدیل می‌شوند.

وظایف کشف و ردیابی عنوان

وظایف TDT عبارت‌اند از:

^۱ Topic Detection and Tracking: TDT

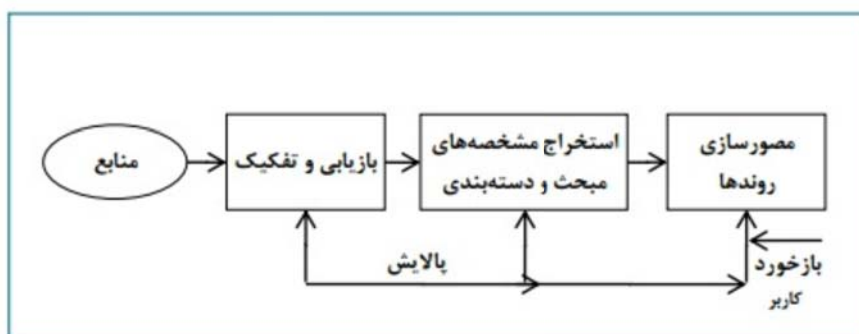
^۲ Topic

^۳ Event

^۴ Subject

۱. بخش‌بندی خبر: مسئله تقسیم رونوشت پیوسته یک برنامه خبری به داستان‌های منفرد است. در تبدیل صوت به متن، داستان‌ها متوالی بوده و لازم است تفکیک شوند.
 ۲. کشف اولین داستان: مسئله شناخت آغاز یک عنوان جدید در جریانی از داستان‌های خبری است.
 ۳. کشف خوشه: مسئله گروه‌بندی همه داستان‌ها در زمان رسیدن آن‌ها بر اساس عناوینی است که در مورد آن بحث می‌کنند. هرچند می‌دانیم که برخی داستان‌ها در مورد چند عنوان بحث می‌کنند، طبق ارزیابی مورد استفاده تا ۲۰۰۱ TDT، هیچ داستانی نمی‌تواند در بیش از یک گروه قرار گیرد.
 ۴. ردیابی: احتیاج به نظارت بر جریان داستان‌های خبری برای یافتن داستان‌های دیگر مربوط به یک موضوع را دارد. این موضوع قبلاً با تعدادی خبر نمونه مشخص شده است. با توجه به ردیابی مستقل عناوین، برخلاف وظیفه خوشه‌بندی، یک داستان اجازه دارد در چند عنوان ردیابی شود.
 ۵. کشف پیوند داستان: مسئله تصمیم‌گیری در این مورد است که آیا دو داستان به تصادف انتخاب شده، درباره عنوان خبری واحدی بحث می‌کنند یا خیر. با وجود کلیدی بودن برای همه وظایف دیگر، این وظیفه چندان مورد توجه جامعه تحقیق قرار نگرفته است.
- یک سیستم شناسایی روند طبق شکل ۸- دارای چهار جزء ضروری است: مجموعه منبع یا ورودی، باز یابی و تجزیه^۱، استخراج و طبقه‌بندی مشخصه‌های عنوان، مصورسازی روند (Khoo Khyou Bun, ۲۰۰۴).

^۱ Parsing



(Khoo Khyou Bun, ۲۰۰۴)

شکل ۸-۴: معماری عمومی سیستم‌های شناسایی ردیابی و عنوان

۸-۳- خوشه‌بندی متون

خوشه‌بندی متون، اعمال روش‌های متداول خوشه‌بندی با اصلاحاتی به‌منظور کار بر روی ماتریس‌های تنک بزرگ می‌باشد. روش‌های مختلفی برای انجام این کار روی مجموعه بزرگ متون وجود دارد که در ادامه توضیح داده می‌شوند.

در یکی از اولین کارهای انجام شده خوشه‌بندی متون بزرگ، کاتینگ و همکاران خوشه‌بندی را به‌عنوان بازیابی اکتشافی اطلاعات معرفی می‌کند (Cutting et al., ۱۹۹۲). کاتینگ و همکاران، الگوریتم‌های سریعی را معرفی می‌کند که در دو مرحله تکراری کار را انجام می‌دهد. در مرحله اول پراکندن^۱، خوشه‌ها به کاربر ارائه می‌شود. کاربر خوشه‌های موردنظر را انتخاب می‌کند. در مرحله دوم جمع‌آوری^۲، خوشه‌های موردنظر ترکیب شده و به مرحله اول برای خوشه‌بندی مجدد و این بار روی حوزه محدود شده‌تر برگشت داده می‌شوند. این کار تا رسیدن به اسناد منفرد ادامه می‌یابد.

^۱ Scatter

^۲ Gather

زمیر و اتزینی (Zamir and Etzioni, ۱۹۹۸) مدل خوشه‌بندی درخت دنباله^۱، را ارائه کردند. این مدل، الگوریتمی با تابع پیچیدگی (زمان) خطی است که بر اساس تشخیص عبارات مشترک بین گروه‌های متون عمل می‌کند. این الگوریتم در موتور جستجوی metacrawler که یک ابرموتور جستجو روی دیگر موتورهای جستجو است، استفاده شده است. این الگوریتم به جنبه‌ای مهم از اطلاعات متنی یعنی توالی کلمات در عبارات مختلف توجه و از آن حداکثر بهره‌برداری را می‌نماید. عبارت، ترتیب مشخصی از یک یا چند کلمه است. خوشه پایه، مجموعه‌ای از متون بوده که دارای عبارت مشترک هستند. این الگوریتم، افزایشی و مستقل از ترتیب است یعنی با رسیدن هر متن از وب، آن متن را پاک‌سازی و به درخت اضافه می‌کند.

استنباخ و همکاران (Steinbach et al., ۲۰۰۰) مروری بر برخی از روش‌های خوشه‌بندی متون می‌کنند. در کار چوانگ و همکاران (Chuang et al., ۲۰۰۰) یک سلسله‌مراتب مفهومی^۲ توسط انسان ساخته می‌شود که هر گره آن دارای برچسبی از چند کلمه است. به هر گره تعدادی سند به طور دستی تخصیص داده می‌شود تا برای یادگیری بانظر آموزش و آزمون استفاده شود. مفروض است که هر گره والد شامل همه گره‌های فرزند می‌باشد. برای هر گره TFIDF که $TF_i \cdot \log(|D|/DF_i)$ می‌باشد حساب می‌گردد. مشخصه TFIDF با تجمع TF و DF از پایین تا بالای سلسله‌مراتب محاسبه می‌شود. هر گره با داشتن یک آستانه به عنوان یک طبقه‌بند آموزش داده می‌شود. پس از آموزش می‌توان هر سند جدید را با توجه به مقایسه شباهت کسینوسی TFIDF آن به گره با آستانه، از بالا به پایین در سلسله‌مراتب طبقه‌بندی کرد. ارزیابی با خطاهای اشتباه مثبت و اشتباه منفی انجام می‌شود. می‌توان دانش زمینه را با ساختن دستی برداری از کلمات مثبت و منفی در نظر گرفت. کلمات منفی در حین آموزش از TFIDF ها کسر می‌شوند. کلمات مثبت در حین آزمون در یک تابع اطمینان استفاده می‌شوند. اگر اطمینان به سطح مشخصی برسد، کفایت کرده و

^۱ Suffix Tree Clustering: STC

^۲ Concept hierarchy

دیگر معیار کسینوس برر سی نمی شود. امکان تخصیص سند به رده‌های میانی سلسله‌مراتب وجود دارد.

یکی از کاربردهای خوشه‌بندی سلسله‌مراتبی متون، ایجاد سلسله‌مراتب برای دسته‌بندی سلسله‌مراتبی می‌باشد. در کار سان و لیم (Sun and Lim, ۲۰۰۱) در سلسله‌مراتب می‌توان سند را به گره‌های رده‌های میانی نیز تخصیص داد. در محاسبه خطای اشتباه مثبت یا اشتباه منفی، درجه دوری (یا شباهت) دسته اشتباه از دسته واقعی در سلسله‌مراتب نیز در ارزیابی در نظر گرفته می‌شود. در هر گره دو طبقه‌بند SVM ساخته می‌شود. طبقه‌بند اول برای تشخیص تخصیص سند به گره بوده و طبقه‌بند دوم برای بررسی ادامه یا عدم ادامه طبقه‌بندی در بچه‌ها به کار می‌رود. در محاسبه خطا دو راه وجود دارد. راه اول محاسبه شباهت کسینوسی مرکز خوشه اشتباه با خوشه درست و دیگری محاسبه فاصله تعداد پیوندهای بین دو خوشه در سلسله‌مراتب می‌باشد. IDF در سطح هر گره حساب می‌شود. می‌توان یک سند را به گره‌ای تخصیص نداد ولی به بچه آن تخصیص داد. در پژوهش دیگری، IDF به جای هر سطح از سلسله‌مراتب در کل اسناد محاسبه شده است که جای تأمل دارد (Pulijala and Gauch, ۲۰۰۴).

از آنجا که بسیاری از الفاظ چند معنی دارند بسیاری از اسناد غیرمرتبط ممکن است به خاطر شباهت لغوی در مجموعه جواب لحاظ شوند. از طرفی چون یک مفهوم را با الفاظ متعددی می‌توان توصیف کرد، اسناد مرتبطی که دارای الفاظ پرس‌وجو نیستند بازایی نخواهند شد. فرض می‌شود که هر سند یا پرس‌وجو می‌تواند به فضای بعد کمتر نگاشت شود تا وابستگی‌های بالقوه بین الفاظ در نظر گرفته شود. روش LSI یکی از این روش‌ها است. در پژوهش کاریپیس و هَن (G. Karypis and E. Han, ۲۰۰۰) ابتدا اسناد بدون دسته خوشه‌بندی شده و برای اسناد دارای دسته، خوشه معلوم در نظر گرفته می‌شود. سپس مرکز هر خوشه حساب می‌شود. نمایش اسناد با TFIDF انجام شده و هر بردار به طول یک نرمال می‌شود. برای کاهش بعد، تصویر هر سند جدید روی بردار مرکز هر خوشه محاسبه می‌شود. بنابراین ابعاد کاهش یافته به تعداد خوشه‌ها می‌باشند.

در واقع هر سند جديد در فضای غير متعامد مراکز خوشه‌ها تصوير می‌شود. با توجه به یک بودن طول هر بردار، تصوير هر سند روی مرکز هر خوشه که همان ضرب داخلی است، نشانگر شباهت کسینوسی سند به خوشه است. چون هر خوشه خلاصه‌ای از عنوان اسناد آن فرض می‌شود، بنابراین در واقع هر سند جديد را در فضای عناوین بیان کرده‌ایم. نتایج آزمایش طبقه‌بندی در روش kNN یا SVM با کاهش بعد به این شیوه بهتر از کاهش بعد LSI هستند. علاوه بر این LSI در حالت بانظر که طبقه اسناد از پیش معلوم هستند نمی‌تواند کار کند.

برخی از مهم‌ترین پژوهش‌های خوشه‌بندی متون توسط گروه پژوهشی کاریپیس^۱ انجام شده است. از دستاوردهای مهم این گروه نرم‌افزار Cluto است که به رایگان در دسترس می‌باشد. جامع‌ترین گزارش از این پژوهش‌ها، گزارش فنی ژائو و کاریپیس می‌باشد (Y. Zhao and G. Karypis, ۲۰۰۲). در این گزارش، اطلاعات هر خوشه در بردار مرکز هر خوشه C_r (یا بردار جمع آن D_r) خلاصه می‌شود. هدف اصلی مقایسه روش‌های تجمیعی یا افزایی (تفکیکی) در خوشه‌بندی سلسله‌مراتبی متون می‌باشد. نتیجه می‌گیرد که روش‌های افزایی هم سریع‌تر و هم بهتر خوشه‌بندی می‌کنند. شباهت برحسب کسینوس است. برای ارزیابی بهینگی خوشه‌ها، شش معیار تابع هدف از ادبیات موضوع بیان می‌شود:

معیار ۱: جمع شباهت زوج میانگین‌های بین اسناد تخصیص داده شده به هر خوشه را با وزن اندازه خوشه، حداکثر می‌کند.

معیار ۲: بر مبنای K-میانگین بوده و شباهت بین هر سند و مرکز خوشه آن حداکثر می‌شود.

معیار ۳: شباهت بردار مرکز هر خوشه با بردار مرکز کل مجموعه را حداقل می‌کند (حداکثر کردن تفاوت).

^۱ Karypis

معیار ۴ و ۵: ترکیب معیارهای حداکثر کردن شباهت داخلی با تفاوت بیرونی هستند.

معیار ۶: بر اساس نظریه گراف، اسناد را به گروه‌هایی افراز می‌کند که برش لبه^۱ هر بخش افراز حداقل باشد.

الگوریتم خوشه‌بندی افرازی از K-میانگین الهام گرفته و در هر بار خوشه‌بندی دو سند تصادفی را به عنوان هسته خوشه انتخاب می‌کند (هر بار شکستن به دو خوشه). سپس همه اسناد آن سطح را به یکی از دو خوشه تخصیص می‌دهد. در فاز بهبود، اسناد به طور تصادفی انتخاب شده و در صورت بهبود معیار بهینه‌سازی بین دو خوشه جابه‌جا می‌شوند. در مرحله بعدی خوشه‌بندی، با بررسی تمام حالات (خوشه‌ها) خوشه‌ای برای افراز مجدد انتخاب می‌شود که بهترین بهبود را در تابع هدف بدهد. خوشه‌بندی تجمعی نیز به سه روش پیوند تکی، پیوند کامل و میانگین گروهی انجام شدند. برای استفاده از مزایای توأم خوشه‌بندی افرازی (سرعت، دید کلی) و خوشه‌بندی تجمعی (بررسی محلی و گروه‌های چسبیده کوچک)، روش تجمعی با محدودیت نیز ابداع شد. ابتدا به روش افرازی، تعدادی خوشه ایجاد می‌شود. سپس در هر کدام از این خوشه‌ها، روش تجمعی اجرا می‌شده و روی خوشه‌های به دست آمده، روش تجمعی تا رسیدن به خوشه کل مجموعه ادامه می‌یابد. ارزیابی با معیار FScore انجام می‌شود که میانگین موزون^۲ دقت و یادآوری است. طبقات اسناد از پیش تعیین شده‌اند. برای هر طبقه، مناسب‌ترین خوشه طبق معیار FScore در نظر گرفته شده و میانگین وزنی FScore طبقات حساب می‌شود. در انتها بحث می‌شود که چرا معیار ۱ افرازی، جوابی متفاوت از تجمعی میانگین گروهی به دست می‌دهد (باوجود تشابه ذاتی). اگر دو طبقه نزدیک بوده و دارای سفتی^۳ متفاوت و یا سفتی کم باشند، معیار ۱ ممکن است زیرخوشه‌های آن‌ها را به اشتباه در مراحل اولیه تلفیق کند.

^۱ Edge-cut

^۲ Balanced

^۳ Tightnes

ژائو و کاریپیس در پژوهش بعدی (Y. Zhao and G. Karypis, ۲۰۰۵a) اصلاحات زیر را اعمال نمودند:

۱. ارزیابی: حالت ایده آل این است که به ازای هر طبقه دستی فقط یک گره خوشه وجود داشته باشد. در معیار FScore خوشه‌های رها شده و بی طبقه در نظر گرفته نمی‌شوند. برای رفع مشکل، معیار آنتروپی در نظر گرفته می‌شود که اندازه‌گیری توزیع اسناد در همه گره‌های درخت می‌باشد.

۲. با نمونه‌گیری مکرر آماری از اسناد و سپس خوشه‌بندی، امکان بررسی تفاوت معنی‌دار روش‌ها از طریق آزمون t فراهم شده است.

۳. روش ابداعی تجمعی با محدودیت تحلیل و با روش بدون محدودیت مقایسه شده است. این کار از طریق مقایسه کیفیت شبیه‌ترین اسناد به هر سند و اثر آن در درخت‌های سلسله‌مراتبی حاصله انجام می‌شود. برای هر سند توزیع طبقه t تا از شبیه‌ترین اسناد، یک بار در کل مجموعه اسناد و بار دیگر در خوشه محدود شده آن، تحلیل می‌شود. تحلیل به کمک آنتروپی و متوسط زوج شباهت‌ها انجام می‌شود. ایجاد محدودیت موجب بهبود می‌شود به خصوص وقتی که سند موردنظر شباهت کمی به بقیه داشته باشد و یا تعداد زیادی سند با تشابه زیاد وجود داشته باشند.

در سال ۲۰۰۵ گروه کاریپیس خوشه‌بندی نرم^۱ را بررسی کردند (Conrad et al., ۲۰۰۵). در این تحقیق بحث مدیریت دانش در شرکت‌های حقوقی و ارتباط آن با طبقه‌بندی و خوشه‌بندی متون مطرح می‌شود. در خوشه‌بندی، دو روش جدید نرم که هر سند را به بیش از یک خوشه تخصیص می‌دهد بررسی می‌گردد. در روش اول در صورت وجود تقطیع^۲ حوزه‌های عنوانی، خوشه‌بندی سخت به هر قطعه اعمال می‌شود. در روش دوم ابتدا خوشه‌بندی سخت انجام‌شده، آنگاه برای هر سند، خوشه‌های نزدیک بر اساس امتیاز متوسط شباهت سند با اسناد هر خوشه محاسبه می‌شود. این امتیاز میانگین و انحراف

^۱ Soft clustering

^۲ Segmentation

معیار همین نوع شباهت با کلیه اسناد خارج از آن خوشه خود سند نرمال شده است. سپس سفت‌ترین خوشه‌ها (شباهت درون خوشه‌ای بالا) انتخاب می‌گردند. معیار ارزیابی خوشه‌یابی نرم، FScore و معیارهای ارزیابی خوشه‌یابی سخت، آنتروپی و خلوص می‌باشند. معیار خلوص به شکل تعداد اسناد بزرگترین طبقه در یک خوشه تقسیم بر اندازه خوشه محاسبه می‌شود.

ژائو و کارپیس در تحقیق دیگری، دانش زمینه را در خوشه‌بندی در نظر گرفتند (Y. Zhao and G. Karypis, ۲۰۰۵b). یکی از روش‌های قبلی این کار روش شبه‌باناظر می‌باشد که در آن برخی از اسناد برچسب طبقه دارند. زدن برچسب طبقه به تعداد زیاد اسناد، بسیار پرهزینه است. نوآوری این مقاله معرفی خوشه‌بندی بر اساس عنوان است. یعنی لیستی از عناوین موجود است که هر عنوان شامل تیترو چند کلمه می‌باشد. در خوشه‌بندی به طور هم‌زمان، دو هدف روابط بین اسناد و عناوین و رابطه بین خود اسناد در نظر گرفته می‌شود. در روش اول، توابع معیار ترکیبی برای بهینه‌سازی معرفی می‌شوند که بهینه‌سازی دو هدفه است. اهداف با الهام از افراز گراف، به طور خطی با ضریبی متناسب با عکس مقدار بهینه هر هدف تنها (برای نرمال‌سازی) ترکیب شده‌اند. روش دوم، تابع معیار دوگانه‌ای است که اهداف را در هم ادغام می‌کند. در فرایند لازم است همیشه بردار عنوان با خوشه‌اش ملازم باشد. در صورت نداشتن اطلاعات کافی از عناوین، سند میانه هر طبقه به عنوان نماینده نمونه عنوان در نظر گرفته می‌شود. نویسندگان، خوشه‌بندی بر اساس عنوان را یک نوآوری مطرح نموده‌اند در صورتی که این موضوع قبلاً در مباحث خوشه‌بندی بر اساس عنوانهای مدل LDA مطرح شده است.

روش‌های خوشه‌بندی متون دارای اشتراکات زیادی با طبقه‌بندی متون به خصوص در پیش‌پردازش و ارزیابی می‌باشد. سباستیان (Sebastiani, ۲۰۰۲) (۲۰۰۵) ادبیات طبقه‌بندی را به طور جامع مرور کرده است.

از روش‌های خوشه‌بندی مجموعه بزرگ اسناد، روش WEBSOM می‌باشد. خوشه‌بندی WEBSOM یک روش شبکه عصبی مبتنی بر نقشه خود-سازمان می‌باشد (Lagus, ۲۰۰۰). این روش پیش از جستجو یا پویش، مجموعه‌ای از اقلام

متنی مثل اسناد را با توجه به محتوای آن‌ها مرتب کرده و آن‌ها را به یک ارائه معمولی دوبعدی از نقاط نقشه نگاشت می‌کند (Lagus et al., ۲۰۰۴). اسنادی که از نظر محتوا مشابه هستند به نقاط یکسان یا مجاور نگاشت شده و از هر نقطه واحد پیوندهایی به پایگاه داده اسناد وجود دارد. بنابراین در حالی که می‌توان جستجو را با مکان‌یابی اسنادی که بهترین تطابق را با عبارت جستجو دارند شروع کرد، نتایج فراتر مرتبط می‌توانند بر اساس اشاره‌گرهای ذخیره شده در همان واحد نقشه یا نقاط مجاور یافته شوند، حتی اگر این نقاط دقیقاً با معیار جستجو تطابق نداشته باشند.

نگاشت دامنه‌های دانش

نگاشت دامنه‌های دانش^۱، واژه‌ای برای توصیف حوزه جدید، در حال تکامل و میان‌رشته‌ای از علم است که فرایند نقشه‌کشی^۲، کاوش، تحلیل، مرتب کردن، قابل‌ناوبری نمودن و نمایش دانش را هدف می‌گیرد. هدف این رشته، سهولت دسترسی به اطلاعات، آشکار کردن ساختار دانش و موفق کردن جستجوگران دانش در تلاش‌هایشان است (Shiffrin and Börner, ۲۰۰۴).

فنون جدید که توانایی پردازش حجم فوق‌العاده‌ای از داده‌ها را دارند، نوید آشکار کردن دانش ضمنی را می‌دهد که تاکنون فقط برای آگاهان دامنه و آن‌هم به طور ناقص شناخته شده است. این فنون توانایی تشخیص و سازمان‌دهی حوزه‌های تحقیق را با توجه به خبرگان، مؤسسه‌ها، مقالات، نشریات، استنادات و متون می‌دهند. هم‌چنین امکان بررسی تغییرات پویا مانند سرعت رشد و چند شاخه شدن، یافتن و نگاشت شبکه‌های علمی و اجتماعی را فراهم می‌کنند.

این فنون جدید، پشتیبان و مکمل قضاوت انسانی هستند. ارزش نگاشت دامنه‌های دانش از مرزهای علوم اطلاعاتی فراتر رفته و دانشمندان، متخصصین، مؤسسات دولتی، صنعت و اعضای عمومی جامعه را در بر می‌گیرد. به دلیل وجود هزاران بُعد در دانش، مصورسازی دامنه، قابلیت تعامل با دانش و دیدن آن از

^۱ Mapping Knowledge Domains

^۲ Charting

زوایای مختلف نقشی کلیدی دارد. نقشه‌ها می‌توانند پژوهشگران برجسته، پراستنادترین مقالات و کتب، مقالات جدیدی که هنوز استناد چندانی به آن‌ها نشده ولی محتوایشان به روندهای نوظهور اشاره دارد، مقالات سازمان‌دهی شده در درخت‌های موضوعی (بر حسب محتوا، استنادات و نویسندگان) و کمک‌های مالی اعطایی بر حسب موضوع را نشان دهند. این فنون نه تنها کاربر را در جنگل دانش قادر به تجسم چند درخت نزدیک می‌کنند بلکه به درک کل چشم‌انداز کمک می‌کنند.

با توجه به پیش‌پردازش‌های قابل توجه لازم، انتظار می‌رود در آینده نزدیک ابزارهایی توسعه یابد که اطلاعات را در فرم‌های اصلی مختلف و مغشوش گرفته و آن‌ها را به شکلی یکسان تبدیل کند و یا داده‌های مغشوش و ناسازگار را مستقیماً تحلیل نماید. از آن جا روش‌های تحلیل فعلی نیاز به داده‌های تمیز دارند، اغلب لازم است از پایگاه‌های داده انحصاری مانند ISI استفاده شود.

مصورسازی دامنه‌های دانش دارای اهمیت کلیدی است (Börner et al., ۲۰۰۳).

بورنر در مصاحبه‌ای اهمیت این حوزه علمی را به خوبی بیان کرده است (Börner, ۲۰۰۵):

« دانش بشری و وسایل ما برای به اشتراک‌گذاری آن با نرخی تصاعدی رشد می‌کند ولی توانایی‌های ادراکی و شناختی ما تقریباً ثابت هستند. از ما انتظار می‌رود از کارهایی خبر داشته باشیم که با صدها بار زندگی کردن هم نمی‌توانیم بخوانیم و بفهمیم. در نتیجه، آگاهان، بسیار متخصص شده و در جزیره تنهایی خود مشغول تحقیق می‌باشند. علم به چندپارگی، دوباره‌کاری و دوباره اختراع کردن خود ادامه می‌دهد. برای بقای گونه بشر، لازم است از سیاره خود محافظت کنیم و یا وسایلی برای حفظ حیات به گونه‌ای که می‌شناسیم بیابیم. علاوه بر بقاء، باید همه ابنای بشر را قادر به زندگی سالم، بهره‌ور و ارضاکنده نماییم. امروز برای دسترسی به همه دانش و آگاهی بشری از موتورهای جستجو استفاده می‌کنیم. موتورهای جستجو، حقایق را از حوزه در حال رشد اطلاعات بازیابی

می‌کنند. ولی این حوزه چه قدر بزرگ است؟ چگونه راه خود را به طور مؤثر به جزایر مفید دانش بیابیم؟ دانش چگونه در مقیاس جهانی پیوند داخلی شده است؟ کدام حوزه‌ها ارزش سرمایه‌گذاری منابع ما را دارد؟ نمی‌دانیم. نقشه‌های جغرافیایی اماکن فیزیکی، برای قرن‌ها راهنمای جستجوهای انسان بوده است. این نقشه‌ها کشف جهان‌های جدید و علامت‌گذاری قلمروهای حیات وحش ناشناخته را ممکن کرده است. بدون نقشه گم می‌شدیم. نقشه‌های دامنه تهیه شده از فضاهای معنایی مجرد در خدمت کاشفان امروزی برای راه‌یابی و مدیریت دنیای علم هستند. این نقشه‌ها از طریق تحلیل علمی مجموعه داده‌های دانشگاهی تهیه می‌شوند تا به اتصال و استفاده از پاره‌های دانش موجود در متون علمی کمک کنند. آن‌ها برای تشخیص عینی حوزه‌های عمده پژوهشی، آگاهان، مؤسسات، جمع‌ها، کمک‌های مالی، مقالات، مجلات، ایده‌ها و غیره به کار می‌روند. در دامنه مورد علاقه، نقشه‌های محلی، دیدی کلی از یک حوزه خاص، همگنی آن، عوامل ورود-صدور و سرعت نسبی را فراهم می‌کنند. آن‌ها اجازه ردیابی ظهور، تکامل و ناپدید شدن مباحث را داده و به تشخیص نویدبخش‌ترین حوزه‌های تحقیقاتی کمک می‌کنند. برای پذیرش عمومی، نیاز بزرگی به م‌صور سازی خوانا و مؤثر برای حل نیازهای حقیقی کاربر وجود دارد. باید دانش اصول درک بصری و پردازش شناختی انسان برای طراحی رابطه‌ای بصری به کار گرفته شود که کار را به‌طور بهینه بین انسان و ماشین توزیع کرده و خوانا و قابل فهم باشند.»

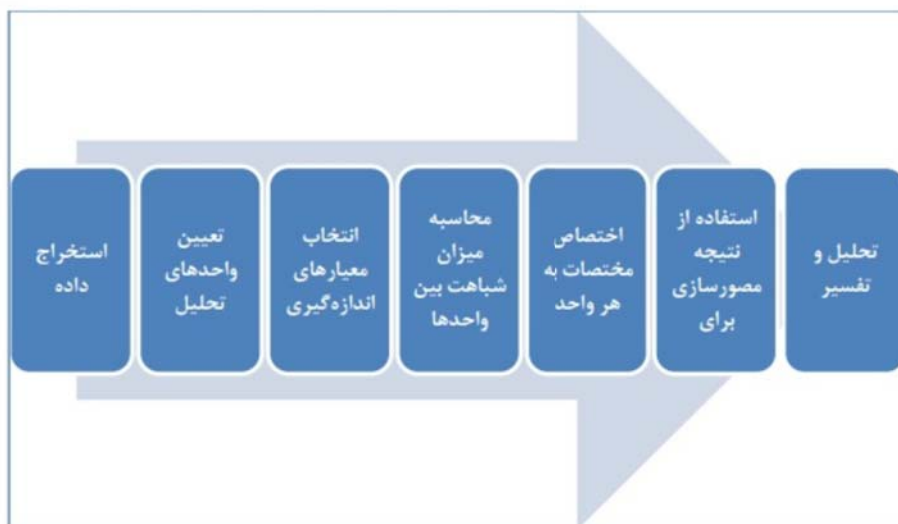
یکی از کاربردی‌ترین روش‌های ارزیابی توسعه علم و فناوری تجزیه و تحلیل متون منتشره در این حوزه، روش‌های علم‌سنجی^۱ است. روش‌های متعارف علم‌سنجی شامل تحلیل استنادی^۲ و متن کاوی می‌باشند (Glenisson et al., ۲۰۰۵; Janssens et al., ۲۰۰۸).

^۱ Scientometrics

^۲ Citation Analysis

فرآیند نگاشت دانش

فرآیند نگاشت دانش شامل مراحل مختلفی است. در ادبیات مختلف این مراحل تقریباً مشابه هستند و از یک الگوی کلی تبعیت می‌کنند. مراحل مختلف فرآیند نگاشت دانش در شکل ۸- نشان داده شده است (Börner et al., ۲۰۰۳).



(Börner et al., ۲۰۰۳)

شکل ۸-۵: قدم‌های مختلف نگاشت حوزه دانش

فنون نگاشت دامنه دانش و کاربرد آن‌ها

واژه‌نگاشت، علاوه بر بخش مصورسازی، به فنون تحلیلی زیربنایی و داده‌کاوی نیز دلالت دارد (Boyack, ۲۰۰۴). ورودی نگاشت دامنه‌های دانش، شامل موضوعات متنوعی مانند تحلیل شبکه (وب، شبکه‌های اجتماعی، شبکه‌های مقیاس آزاد، مسیرهای متابولیک)، زبان‌شناسی، استخراج مفهوم یا مبحث، تحلیل استناد، شاخص‌های علم و فناوری و فنون مصورسازی می‌باشد. دامنه دانش وسیع‌تر از "رشته فنی"^۱ است. برای ناآگاهان، نگاشت فراهم‌کننده مدخلی به یک دامنه بوده و کمکی برای کسب دانش در سطوح خرد و کلان می‌باشد.

^۱ Technical field

برای آگاهان، نگاشت فراهم‌کننده اعتبارسنجی ادراکات^۱ و کمکی برای مطالعه روندها و اطلاعات جدید است. نگاشت و اکتشاف تعاملی، موجب اعجاب خبره از توسعه‌های حول برداشت خود است. کاربرد فنون نگاشت دامنه دانش در جدول ۸-۱ نشان داده شده است.

جدول ۸-۱: خلاصه فنون نگاشت دامنه دانش و کاربرد آن‌ها

سؤالات مرتبط به				
واحد تحلیل	رشته‌ها و پارادایمها	جوامع و شبکه‌ها	عملکرد تحقیق یا مزیت رقابتی	الگوریتمهای متداول
نویسندگان			عملکرد تحقیق یا مزیت رقابتی	برنامه‌های شبکه‌های اجتماعی، مقیاس‌بندی چندبعدی، تحلیل عامل، شبکه‌های مسیریاب
اسناد	ساختار رشته، پویاییها، توسعه پارادایم	استفاده از ویژگیهای شبکه به عنوان شاخص	هم‌استنادی، هم‌واژگی، فضای برداری، تحلیل معنایی پنهان، تحلیل مؤلفه‌های اصلی، روشهای خوشه‌بندی	
مجلات	ساختار علم، پویاییها، دسته‌بندی، نشت بین رشته‌ها	استفاده از نگاشت رشته با شاخصها	هم‌استنادی، بین‌استنادی	
کلمات		ساختار شـناختی، پویاییها	فضای برداری، تحلیل معنایی پنهان، تخصیص دیریکله پنهان (LDA)	
شاخصها و معیارها		مقایسه رشته‌ها، مؤسسات، کشورها و غیره، ورودی-خروجی		

(Boyack, ۲۰۰۴)

تحلیل هم‌نویسندگی مشابه تحلیل شبکه‌های اجتماعی می‌باشد. درحالی‌که تحلیل شبکه‌های اجتماعی با خصوصیات سراسری پایگاه‌های بزرگ داده نویسندگان کار می‌کند، مطالعات هم‌نویسندگی به دنبال پاسخ به سؤال‌های

^۱ Perception

خاص در مورد گروه‌های همکاری می‌باشد. تحلیل هم‌استنادی نویسندگان برای بررسی ساختار و سابقه معنوی^۱ مناسب بوده و اغلب با تحلیل عاملی و مقیاس‌بندی چندبعدی انجام می‌شود. مقیاس‌بندی شبکه مسیریاب^۲ برای آماده‌سازی داده‌ها برای چیدمان در مصورسازی مناسب هستند.

۸-۴- کدهای متن کاوی

در ادامه برخی کدهای مربوط به متن کاوی در نرم‌افزار R را به شما ارائه خواهیم نمود. در ابتدا چند بسته مربوط به متن کاوی را مشاهده می‌فرمایید. بسته `tm` یک فریم ورک برای متن کاوی است که جزو مهم‌ترین بسته‌های متن کاوی زبان R به حساب می‌آید. در ادامه دیگر بسته‌های مرتبط با متن کاوی را مشاهده می‌نمایید.

```
> library(tm)
> library(SnowballC)
> library(qdap)
> library(qdapDictionaries)
> library(dplyr)
> library(RColorBrewer)
> library(ggplot2)
> library(scales)
```

انواع داده‌ای مختلفی هستند که توسط بسته `tm` پشتیبانی می‌شوند. با استفاده از قطعه کد `getsources()` می‌توانیم انواع این منابع داده و همچنین با استفاده از تابع `getReaders()` انواع فرمت مربوط به این منابع را مشاهده نماییم.

^۱ Intellectual

^۲ Pathfinder

```

> library(tm)
> getSources()
[1] "DataframeSource" "DirSource"          "URISource"
[4] "VectorSource"    "XMLSource"
> getReaders()
[1] "readDOC"          "readPDF"
[3] "readPlain"        "readRCV1"
[5] "readRCV1asPlain"  "readReut21578XML"
[7] "readReut21578XMLasPlain" "readTabular"
[9] "readXML"

```

با استفاده از دستور `file.path` محل قرار گرفتن فایل‌های موردنظر برای متن کاوی می‌باشد. همان‌طور که مشاهده می‌شود ۵ فایل در این پوشه برای متن کاوی وجود دارد. بهترین نوع داده در متن کاوی با پسوند `txt` می‌باشد که هر نوع داده‌ای را به وسیله نرم‌افزارهای مختلف می‌توان به این فرمت تبدیل نمود و سپس الگوریتم‌های متن کاوی را بر روی آن‌ها اعمال نمود.

```

> path <- file.path("d:/r", "root", "textfiles")
> path
[1] "d:/r/root/textfiles"
> length(dir(path))
[1] 5
> dir(path)
[1] "Data Mining.txt" "Network Data.txt" "pdftest.txt"
[4] "pg100.txt"       "R_mining_2.txt"

```

توسط بسته tm همه اسناد را در متغیری به نام files قرار می‌دهیم.

```
> files <- Corpus(DirSource(path))
> files
<<VCorpus (documents: 5, metadata (corpus/indexed): 0/0)>>
> class(files)
[1] "VCorpus" "Corpus"
> class(files[[1]])
[1] "PlainTextDocument" "TextDocument"
> summary(files)
      Length Class      Mode
Data Mining.txt      2 PlainTextDocument list
Network Data.txt      2 PlainTextDocument list
pdftest.txt           2 PlainTextDocument list
pg100.txt             2 PlainTextDocument list
R_mining_2.txt        2 PlainTextDocument list
```

برای آماده‌سازی داده‌ها برای متن‌کاوی نیاز به یک سری پیش‌پردازش بر روی اسناد داریم. تابع `getTransformations()` برخی از این توابع را معرفی می‌نماید.

```
> getTransformations()
[1] "removeNumbers"      "removePunctuation" "removeWords"
[4] "stemDocument"       "stripWhitespace"
```

برای دسترسی مستقیم به محتوای هرکدام از اسناد از طریق کد زیر می‌توان اقدام نمود.

```
> files <- Corpus(DirSource(path))
> inspect(files[3])
```

از تابع `tm_map()` به منظور اعمال تبدیلات مختلف بر روی اسناد موردنظر استفاده می‌گردد. در قطعه کدهای زیر با استفاده از این تابع به ترتیب مواردی همچون علامت‌های خاص، اعداد، کلمات خاص، فاصله‌ها و ... اضافی حذف گردیده‌اند و به اسنادی نسبتاً استاندارد تبدیل شده‌اند.

```

> toSpace <- content_transformer(function
  (x, pattern) gsub(pattern, " ", x))
> files <- tm_map(files, toSpace, "/")
> files <- tm_map(files, toSpace, "@")
> files <- tm_map(files, toSpace, "\\|")
> files <- tm_map(files, toSpace, "/|@|\\|")
> files <- tm_map(files, content_transformer(tolower))
> files <- tm_map(files, removeNumbers)
> files <- tm_map(files, removePunctuation)
> files <- tm_map(files, removeWords, stopwords("english"))
> files <- tm_map(files, removeWords, c("department", "email"))
> files <- tm_map(files, stripWhitespace)
> library(SnowballC)
> files <- tm_map(files, stemDocument)

```

تابع `content_transformer(tolower)` به منظور تبدیل به حروف کوچک ، `tm_map(docs, removeNumbers)` به منظور حذف اعداد ، `tm_map(docs, removePunctuation)` به منظور حذف نقاط و نشانه‌ها ، `tm_map(docs, removeWords, stopwords("english"))` ، به منظور حذف برخی کلمات رایج و پراستفاده که در ادامه این کلمات را مشاهده می‌فرمایید،

`tm_map(docs, removeWords, c("department", "email"))` به منظور حذف کلماتی خاص، `tm_map(docs, stripWhitespace)` به منظور حذف فاصله‌های اضافی استفاده می‌گردند. همچنین `stemDocument` به منظور حذف پسوندهای اضافی در کلمات و استخراج ریشه کلمات مورد استفاده قرار می‌گیرد.

```

> length(stopwords("english"))
[1] 174
> stopwords("english")
[1] "i" "me" "my" "myself" "we"
[6] "our" "ours" "ourselves" "you" "your"
[11] "yours" "yourself" "yourselves" "he" "him"
[16] "his" "himself" "she" "her" "hers"
[21] "herself" "it" "its" "itself" "they"
[26] "them" "their" "theirs" "themselves" "what"
[31] "which" "who" "whom" "this" "that"
[36] "these" "those" "am" "is" "are"
[41] "was" "were" "be" "been" "being"
[46] "have" "has" "had" "having" "do"
[51] "does" "did" "doing" "would" "should"
[56] "could" "ought" "i'm" "you're" "he's"
[61] "she's" "it's" "we're" "they're" "i've"
[66] "you've" "we've" "they've" "i'd" "you'd"
[71] "he'd" "she'd" "we'd" "they'd" "i'll"
[76] "you'll" "he'll" "she'll" "we'll" "they'll"
[81] "isn't" "aren't" "wasn't" "weren't" "hasn't"
[86] "haven't" "hadn't" "doesn't" "don't" "didn't"
[91] "won't" "wouldn't" "shan't" "shouldn't" "can't"
[96] "cannot" "couldn't" "mustn't" "let's" "that's"
[101] "who's" "what's" "here's" "there's" "when's"
[106] "where's" "why's" "how's" "a" "an"
[111] "the" "and" "but" "if" "or"
[116] "because" "as" "until" "while" "of"
[121] "at" "by" "for" "with" "about"
[126] "against" "between" "into" "through" "during"
[131] "before" "after" "above" "below" "to"
[136] "from" "up" "down" "in" "out"
[141] "on" "off" "over" "under" "again"
[146] "further" "then" "once" "here" "there"
[151] "when" "where" "why" "how" "all"
[156] "any" "both" "each" "few" "more"
[161] "most" "other" "some" "such" "no"
[166] "nor" "not" "only" "own" "same"
[171] "so" "than" "too" "very"

```


با استفاده از تابع `DocumentTermMatrix()` کلمات را در قالب ماتریس تبدیل نموده که تعداد تکرار هر کدام از این کلمات نیز قابل بررسی می‌باشد. در کدهای زیر مشاهده می‌شود از ۵ سندی که به نرم‌افزار داده شده است حدود ۲۶۴۴۷ کلمه استخراج گردیده است.

```
> dtm <- DocumentTermMatrix(files)
> dtm
<<DocumentTermMatrix (documents: 5, terms: 26447)>>
Non-/sparse entries: 32064/100171
Sparsity           : 76%
Maximal term length: 73
Weighting           : term frequency (tf)
> inspect(dtm[1:5, 1000:1005])
<<DocumentTermMatrix (documents: 5, terms: 6)>>
Non-/sparse entries: 10/20
Sparsity           : 67%
Maximal term length: 16
Weighting           : term frequency (tf)

> class(dtm)
[1] "DocumentTermMatrix"      "simple_triplet_matrix"
> dim(dtm)
[1]      5 26447

> tdm <- TermDocumentMatrix(files)
> tdm
<<TermDocumentMatrix (terms: 26447, documents: 5)>>
Non-/sparse entries: 32064/100171
Sparsity           : 76%
Maximal term length: 73
Weighting           : term frequency (tf)
```

با استفاده از کدهای زیر می‌توان کم‌تکرارترین و پرتکرارترین کلمات را مشاهده نمود.

```
> freq <- colSums(as.matrix(dtm))
> ord <- order(freq)
> freq[head(ord)]
      pdftest.txt  R_mining_2.txt Network Data.txt
           1987           15105           36291

      Data Mining.txt  pg100.txt
           46677           480431

> freq[tail(ord)]
      pdftest.txt  R_mining_2.txt Network Data.txt
           1987           15105           36291

      Data Mining.txt  pg100.txt
           46677           480431

> head(table(freq), 15)
freq
 1987 15105 36291 46677 480431
    1     1     1     1     1
> tail(table(freq), 15)
freq
 1987 15105 36291 46677 480431
    1     1     1     1     1
```

با استفاده از تابع `removeSparseTerms()` می‌توان کلماتی که جزو کلمات کم‌تکرار هستند را حذف نمود.

```
> m <- as.matrix(dtm)
> dim(m)
[1] 5 26447
> dim(dtm)
[1] 5 26447
> dtms <- removeSparseTerms(dtm, 0.1)
> dim(dtms)
[1] 5 284
> inspect(dtms)
<<DocumentTermMatrix (documents: 5, terms: 284)>>
Non-/sparse entries: 1420/0
Sparsity : 0%
Maximal term length: 10
Weighting : term frequency (tf)
```

با استفاده از کد `findFreqTerms()` کلماتی با حداقل تعداد تکرار را می‌توان مشاهده نمود و با استفاده از تابع `findAssocs` کلمات با میزان همبستگی بالا قابل دسترسی می‌باشند.

```
> findFreqTerms(dtm, lowfreq=1000)
[1] "can"      "come"      "data"      "day"      "duke"      "enter"
[8] "eye"      "father"    "first"     "function"  "give"      "god"
[15] "hand"     "hath"      "heart"     "ill"      "king"      "know"
[22] "let"      "like"      "look"      "lord"     "love"      "make"
[29] "may"      "mine"      "model"     "much"     "must"      "name"
[36] "never"    "now"       "one"       "say"      "see"       "set"
[43] "sir"      "speak"     "take"      "tell"     "thee"      "thi"
[50] "thou"     "time"      "tis"       "two"      "upon"      "use"
[57] "will"     "word"      "yet"

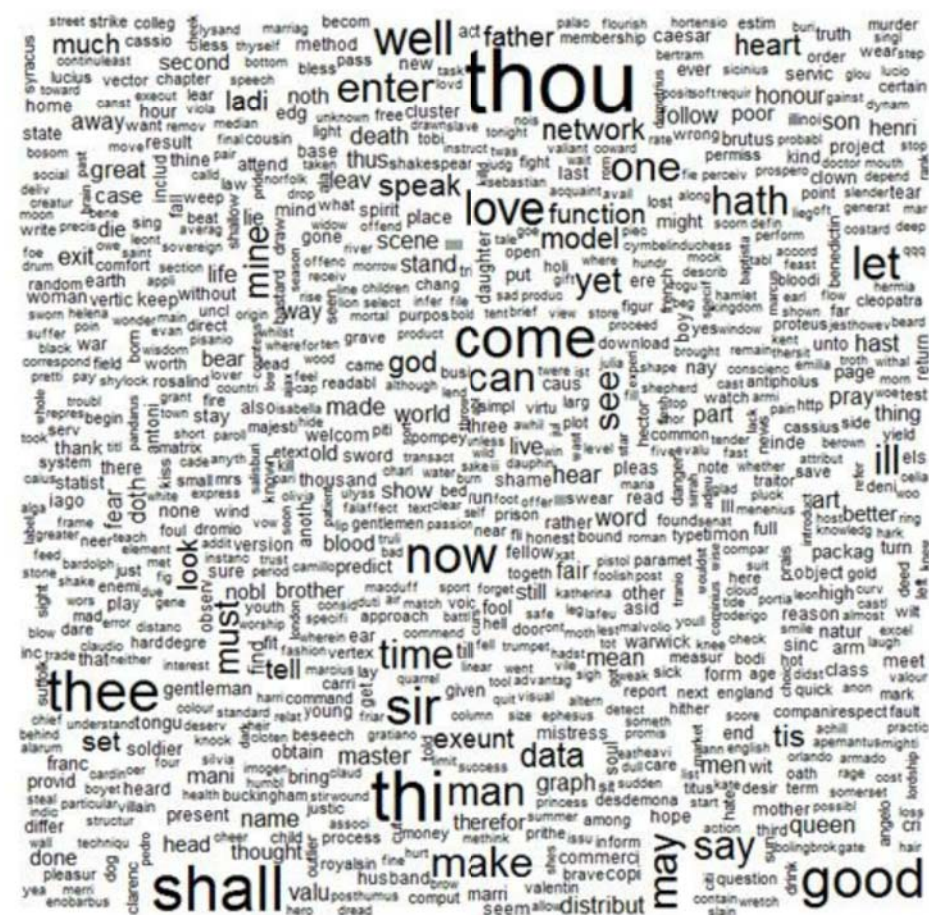
> findAssocs(dtm, "data", corlimit=0.99)
      data
queri   1.00
reduct  1.00
regress 1.00
applic  0.99
cleveland 0.99
techniqu 0.99
```

بررسی کلماتی که بیشترین تکرار را دارند با استفاده از کد زیر قابل دسترسی می‌باشد.

```
> freq <- sort(colSums(as.matrix(dtm)), decreasing=TRUE)
> head(freq, 14)
will thou thi shall lord come thee king good now si:
5712 5485 4032 3602 3566 3322 3178 3171 3027 2841 2801
> wf <- data.frame(word=names(freq), freq=freq)
> head(wf)
      word freq
will  will 5712
thou   thou 5485
thi     thi 4032
shall  shall 3602
lord   lord 3566
come   come 3322
```


برای مصورسازی بهتر کلمات می‌توان از توده ابری استفاده نمود که در شکل زیر مشاهده می‌فرمایید.

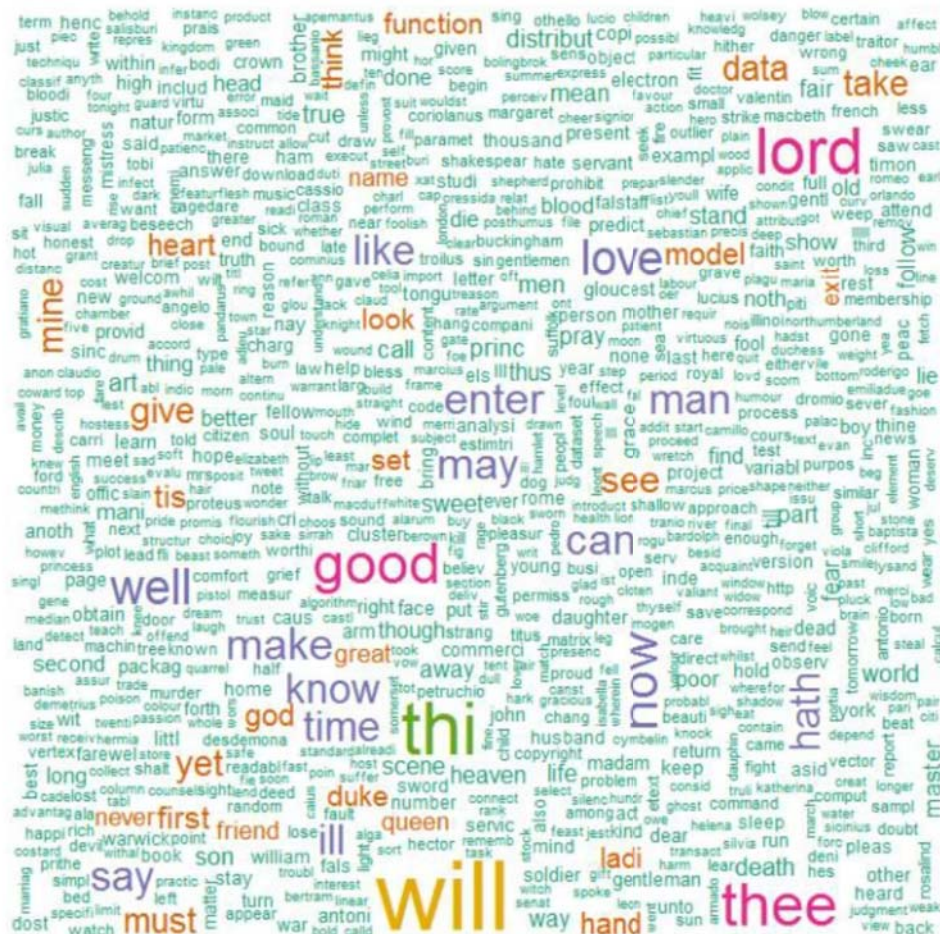
```
> library(wordcloud)
> set.seed(123)
> wordcloud(names(freq), freq, min.freq=100)
```



شکل ۸-۷: مصورسازی کلمات به صورت ابری

در کد زیر نیز شکل توده ابر کلمات پر کاربرد را به صورت رنگی مشاهده می‌نمایید.

```
> set.seed(142)
> wordcloud(names(freq), freq, min.freq=100, colors=brewer.pal(6, "Dark2"))
```



شکل ۸-۸: مصورسازی رنگی کلمات به صورت ابری

منابع

منابع

- ابریشمی ، حمید، (۱۳۷۳) ، اقتصادسنجی کاربردی
- باستانی ، سوسن ، رئیسی ، مهین ، (۱۳۹۰) ، روش تحلیل شبکه: استفاده از رویکرد شبکه‌های کل در مطالعه اجتماعات متن‌باز ، مجله مطالعات اجتماعی ایران، دوره پنجم، شماره ۲
- تیمورپور، بابک، (۱۳۸۸)، کشف روندهای نوظهور در حوزه‌های علمی بر پایه خوشه‌بندی پویا با رویکرد متن‌کاوی و تحلیل پیوند، پایان نامه دکتری دانشگاه تربیت مدرس.
- عصاره، فریده؛ سهیلی، فرامرز؛ فرج پهلوی، عبدالحسین ، معرف زاده، عبدالحمید (۱۳۹۱). بررسی سنجه مرکزیت در شبکه هم نویسندگی مقالات مجلات علم اطلاعات.
- عصاره، فریده؛ سهیلی، فرامرز؛ مفاهیم مرکزیت و تراکم در شبکه‌های علمی و اجتماعی ، فصلنامه مطالعات ملی کتابداری و سازمان‌دهی اطلاعات ، شماره ۹۵
- غضنفری_م، علیزاده_س و تیمورپور_ب. (۱۳۸۷)، داده‌کاوی و کشف دانش، انتشارات دانشگاه علم و صنعت ایران.
- فاطمی قمی م.ت. (۱۳۷۵) ، "پیش‌بینی و تجزیه و تحلیل سری‌های زمانی" ، نشر دانش امروز
- نیرومند، ح. ، بزرگ‌نیا ا. (۱۳۷۲) " مقدمه‌ای بر تحلیل سری‌های زمانی " ، نشر دانشگاه فردوسی مشهد
- نیرومند، حسینعلی، (۱۳۷۱) " تجزیه و تحلیل سری‌های زمانی، نشر دانشگاه فردوسی مشهد
- ماهنامه‌ی رایانه شماره (۱۸۸)
- ACM. (۲۰۰۶), "Data Mining Curriculum: A Proposal (Version ۱.۰)." Retrieved from http://www-sal.cs.uiuc.edu/~hanj/kdd_curriculum.pdf
- Allan, J., Kontostathis, A., Galitsky, L., M.Pottenger, W., Roy, S. and Phelps, D.J. (۲۰۰۲), "A survey of emerging trend detection in textual data mining," Springer-Verlag.
- Alpaydin E., "Introduction to Machine Learning", The MIT Press, ۲۰۰۴
- Ankerst, M., Breunig, M.M., Kriegel, H.P. and Sander, J. (۱۹۹۹), "OPTICS: ordering points to identify the clustering structure," Vol. ۲۸, pp. ۴۹-۶۰. Retrieved from <http://portal.acm.org/citation.cfm?id=۳۰۴۱۸۷>

- Barnes, J. A. "Class and Committees in a Norwegian Island Parish", *Human Relations* ۷:۳۹-۵۸
- Banerjee A., Ghosh J. (۲۰۰۰) "clickstream clustering using weighted longest common subsequences"
- Bellegarda, J.R. (۲۰۰۷), "Latent Semantic Mapping: Principles & Applications," in Juang, B.H. (Ed.), *Synthesis Lectures on Speech and Audio Processing*, Morgan & Claypool.
- Berkowitz, S. D. (۱۹۸۲). *An Introduction to Structural Analysis: The Network Approach to Social Research*. Toronto: Butterworth.
- Berry, M.W., Drmac, Z. and Jessup, E.R. (۱۹۹۹), "Matrices, vector spaces, and information retrieval," *SIAM review*, pp. ۳۳۵-۳۶۲. Retrieved from <http://www.jstor.org/stable/۲۶۵۳۰۷۷>
- Breiger, Ronald L. ۲۰۰۴. "The Analysis of Social Networks." Pp. ۵۰۵-۵۲۶ in *Handbook of Data Analysis*, edited by Melissa Hardy and Alan Bryman. London: Sage Publications. Excerpts in pdf format (<http://www.u.arizona.edu/~breiger/NetworkAnalysis.pdf>)
- Bonacich, P. (۱۹۸۷) Power and Centrality: A Family of Measures, *The American Journal of Sociology*, ۹۲ (۵), pp ۱۱۷۰-۱۱۸۲
- Börner, K. (۲۰۰۵), "Interviewing Katy Börner." Retrieved July ۲۳, (۲۰۱۱), from <http://www.infovis.net/printMag.php?num=۱۷۰&lang=۲>
- Börner, K., Chen, C. and Boyack, K.W. (۲۰۰۳), "Visualizing Knowledge Domains," *Annual Review of Information Science & Technology*, NJ: Information Today, Inc./American Society for Information Science and Technology., Vol. ۳۷, pp. ۱۷۹-۲۵۵.
- Boyack, K.W. (۲۰۰۴), "Mapping knowledge domains: characterizing PNAS," *Proceedings of the National Academy of Sciences of the United States of America*, Vol. ۱۰۱ Suppl No. suppl_۱, pp. ۵۱۹۲-۹. doi: ۱۰.۱۰۷۳/pnas.۰۳.۷۵۰.۹۱۰۰
- Bun, Khoo Khyou. (۲۰۰۴), *Topic Trend Detection and Mining in World Wide Web*.
- Burt, Ronald S. (۱۹۹۲). *Structural Holes: The Structure of Competition*. Cambridge, MA: Harvard University Press. Carrington, Peter J., John Scott and Stanley Wasserman (Eds.). ۲۰۰۵. *Models and Methods in Social Network Analysis*. New York: Cambridge University Press.
- Carrington, J. Scott, & S. Wasserman (Editors), *Models and Methods in Social Network Analysis* (pp. ۲۷۰-۳۱۶). New York: Cambridge University Press.
- Conrad, J.G., Al-Kofahi, K., Zhao, Y. and Karypis, G. (۲۰۰۵), "Effective document clustering for large heterogeneous law firm collections," *bolonga*.
- Chen, C. (۲۰۰۶), "CiteSpace II: Detecting and Visualizing Emerging Trends and Transient Patterns in Scientific Literature," *Journal of the American Society for Information Science and Technology*, Vol. ۵۷, pp. ۳۵۹-۳۷۷.

- Christakis, Nicholas and James H. Fowler "The Spread of Obesity in a Large Social Network Over ٣٢ Years," New England Journal of Medicine ٣٥٧ (٤): ٣٧٠-٣٧٩ (٢٦ July ٢٠٠٧)
- c.f. Katzmeier (٢٠٠٧): Social Network Analysis. The Science of Measuring, Visualizing and Simulating Data on Social Relationships, Working Paper Series, Vienna
- Chuang, W.T., Tiyyagura, A., Yang, J. and Giuffrida, G. (٢٠٠٠), "A fast algorithm for hierarchical text classification," p. ٤٠٩-
- Cutting, D.R., Pedersen, J.O., Karger, D.R. and Tukey, J.W. (١٩٩٢), "Scatter/gather: A cluster-based approach to browsing large document collections," Copenhagen, pp. ٣١٨-٣٢٩.
- Dennis, S. (٢٠٠٤), "An unsupervised method for the extraction of propositional information from text," PNAS, Vol. ١٠١, pp. ٥٢٠٦-٥٢١٣.
- Dörre, J., Gerstl, P. and Seiffert, R. (١٩٩٩), "Text mining: finding nuggets in mountains of textual data," San Diego, California, United States, ACM.
- Dunham, M.H. (٢٠٠٣), Data mining introductory and advanced topics, Prentice Hall/Pearson Education, p. ٣١٥. Retrieved from <http://books.google.com/books?id=٣٠LBQgAACAAJ>
- Ester, M., Kriegel, H.P., Sander, J. and Xu, X. (١٩٩٦), "A density-based algorithm for discovering clusters in large spatial databases with noise," Vol. (١٩٩٦), pp. ٢٢٦-٢٣١. Retrieved from <https://www.aaai.org/Papers/KDD/١٩٩٦/KDD٩٦-٠٣٧.pdf>
- Faloutsos C., Lin K. (١٩٩٥) "Fast map"
- Friedman, J.H. (١٩٩٧), "Data Mining and Statistics. What's the Connection?," Houston, Texas, Computing Science and Statistics.
- Fradkin, D. and Madigan, D. (٢٠٠٣), "Experiments with random projections for machine learning," pp. ٥١٧-٥٢٢. Retrieved from <http://portal.acm.org/citation.cfm?id=٩٥٦٨١٢>
- Feng, A. and Allan, J. (٢٠٠٥), Hierarchical topic detection in TDT-٢٠٠٤', CIIR technical report, University of Massachusetts Amherest.
- Garfield, E. (١٩٥٥), "Citation indexes for science: a new dimension in documentation through association of ideas," Science, Vol. ١٢٢, pp. ١٠٨-١١١.
- Garfield, E. (٢٠٠٤), "The HistCite system for mapping and bibliometric analysis of the output of searches using the ISI web of knowledge," Vol. Presented .
- Graham W. (٢٠١٤) , Hands-On Data Science with R , Text Mining chapter
- Ginsparg, P., Houle, P., oachims, T.. and Sul, J. (٢٠٠٤), "Mapping subsets of scholarly information," PNAS, Vol. ١٠١, pp. ٥٢٣٦-٥٢٤٠.
- Glenisson, P., Glanzel, W., Janssens, F. and Moor, B. De. (٢٠٠٥), "Combining full-text and bibliometric information in mapping scientific disciplines," Information Processing and Management, Special Issue on Informetrics, Vol. ٤١, pp. ١٥٤٨-١٥٧٢.
- Han. J. Kamber. M. (٢٠٠٦) "data mining concepts and techniques"

- Han, J. and Kamber, M. (۲۰۰۶), Introduction to Data mining concepts and techniques, Morgan Kaufmann.
- Hand, D.J., Mannila, H. and Smyth, P. (۲۰۰۰), Principles of Data Mining, MIT Press.
- Han J. & Kamber M., ۲۰۰۶. "Chapter : Mining Frequent Patterns, Associations, and Correlations", Data mining: concepts and techniques, ۲nd Edition, Morgan Kaufmann Publishers.
- Han J. & Kamber M., ۲۰۰۶., Data mining: concepts and techniques, ۲nd Edition, Morgan Kaufmann Publishers.
- Hearst, M. (۲۰۰۳), "What is text mining," Retrieved October, Vol. ۱۸, p.(۲۰۰۵). Retrieved from <http://www.jaist.ac.jp/~bao/MOT-Ishikawa/FurtherReadingNo۱.pdf>
- Hinneburg, A. and Keim, D.A. (۱۹۹۸), An efficient approach to clustering in large multimedia databases with noise, Bibliothek der Universität Konstanz. Retrieved from <http://www.aai.org/Papers/KDD/۱۹۹۸/KDD۹۸-۰۰۹.pdf>
- Ho, T.B. and Dam, H.C. (۲۰۰۵), "Introduction to Knowledge Discovery and Data Mining." Retrieved from <http://www.jaist.ac.jp/~bao/MOT-Ishikawa/MOT-Ishikawa.pdf>
- Hill, R. and Dunbar, R. (۲۰۰۲). "Social Network Size in Humans." Human Nature, Vol. ۱۴, No. ۱, pp. ۵۳-۷۲. Google (http://www.google.com/search?q=cache:sZ_e۹TbhRboJ:www.liv.ac.uk/evolpsych/)
- Hill_Dunbar_networks.pdf+social+network+size&hl=en&gl=ca&ct=clnk&cd=۱) Social network analysis software ۴۱
- Hopcroft, J., Khan, O., Kulis, B. and Selman, B. (۲۰۰۴), "Tracking evolving communities in large linked networks," PNAS, Vol. ۱۰۱, pp. ۵۲۴۹-۵۲۵۳.
- Huang X., Acero A., Hon H. W., "Spoken Language Processing", Printice Hall, ۲۰۰۰
- Kantardzic M. (۲۰۰۳) "Chapter ۲: Preparation the Data", Data Mining: Concepts, Models, Methods, and Algorithms, John Wiley & Sons.
- Janssens, F. (۲۰۰۷), Clustering of scientific fields by integrating text mining and bibliometrics, Leuven.
- Jivawei Han, Micheline Kamber, ۲۰۰۶, Data Mining concepts & Techniques, Elsevier Inc
- Karypis, G. and Han, E. (۲۰۰۰), "Fast dimensionality reduction algorithm with applications to document retrieval & categorization," p. ۱۲.
- Keller F., "Clustering", Computer University Saarlandes, Tutorial Slides
- Keogh. E, Pazzani M. (۲۰۰۰) "Scaling up dynamic time warping for datamining application", proceeding of the sixth acm sigkdd conference on knowledge discovery and datamining
- Keogh E., Pazzani .M. (۲۰۰۱) "derivative dynamic time warping"
- Keogh E. et al (۲۰۰۴) "Extract indexing of dynamic time warping"

- Kolaczyk E., Csardi G. (٢٠١٤), Statistical Analysis of Network Data with R.
- Kosala, R. and Blockeel, H. (٢٠٠٠), "Web mining research: A survey," ACM SIGKDD Explorations Newsletter, Vol. ٢ No.
- Koschützki, D.; Lehmann, K. A.; Peeters, L.; Richter, S.; Tenfelde-Podehl, D. and Zlotowski, O. (٢٠٠٥)
- Kovács F., Legány C., Babos A., "Cluster Validity Measurement Techniques", Department of Automation and Applied Informatics, Budapest University of Technology and Economics, (٢٠٠٣)
- Krebs, Valdis (٢٠٠٨) A Brief Introduction to Social Network Analysis (Common metrics in most SNA software Web Reference (<http://www.orgnet.com/sna.html>).)
- Krebs, Valdis (٢٠٠٨) Various Case Studies & Projects using Social Network Analysis software Web Reference (<http://www.orgnet.com/cases.html>).)
- Kumar V. (٢٠١١), Data Mining and Knowledge Discovery Series, University of Minnesota, Department of Computer Science and Engineering
- Kaufman, L. and Rousseeuw, P. (١٩٩٠), Finding Groups in Data: An Introduction to Cluster Analysis, New York, John Wiley and Sons. Retrieved from http://www.garfield.library.upenn.edu/papers/science_v١٢٢v٣١٥٩p١٠٨y١٩٥٥.html
- Kohonen, T. (١٩٨٢), "Self-organized formation of topologically correct feature maps," Biological Cybernetics, Vol. ٤٣ No. ١, pp. ٥٩-٦٩. doi:١٠.١٠٠٧/BF٠٠٣٣٧٢٨٨
- Landauer, T.K., Laham, D. and Derr, M. (٢٠٠٤), "From paragraph to graph: Latent semantic analysis for information visualization," PNAS, Vol. ١٠١, pp. ٥٢١٤-٥٢١٩.
- Lagus, K. (٢٠٠٠), "Text Mining with the WEBSOM," Acta Polytechnica Scandinavica, Mathematics and Computing Series, p. ٥٤. Retrieved from <http://lib.tkk.fi/Diss/٢٠٠٠/isbn٩٥١٢٢٥٢٦٠٠/>
- Lagus, K., Kaski, S. and Kohonen, T. (٢٠٠٤), "Mining massive document collections by the WEBSOM method," Information Sciences, Vol. ١٦٣ No. ١-٣, pp. ١٣٥-١٥٦. Retrieved from <http://www.sciencedirect.com/science/article/pii/S٠٠٢٠٠٢٥٥٠٣٠٠٤١٩٥>
- Larose, D. T. (٢٠٠٥) Discovering Knowledge in Data :an introduction to datamining . Wiley Interscience. Daniel Larose, "Datamining Methods and Models, Hoboken, NJ, ٢٠٠٥.
- Lin, Nan, Ronald S. Burt and Karen Cook, eds. (٢٠٠١). Social Capital: Theory and Research. New York: Aldine de Gruyter.
- Lin, J. and Gunopulos, D. (٢٠٠٣), "Dimensionality reduction by random projection and latent semantic indexing." Retrieved from [http://cchen١.csie.ntust.edu.tw:٨٠٨٠/students/\(٢٠٠٩\)/Dimensionality reduction by random projection and latent semantic indexing.pdf](http://cchen١.csie.ntust.edu.tw:٨٠٨٠/students/(٢٠٠٩)/Dimensionality%20reduction%20by%20random%20projection%20and%20latent%20semantic%20indexing.pdf)
- Liu, X., Yu, S., Moreau, Y., Janssens, F. and Moor, B.D. (٢٠٠٩),

“Hybrid clustering by integrating text and citation based graphs in journal database analysis,” IEEE International Conference on Data Mining Workshops.

- Maimon O., Rokach L., (۲۰۱۰), Data Mining and Knowledge Discovery Handbook, Tel Aviv University.
- Martin, B. (۱۹۹۵): Instance-Based Learning: Nearest Neighbour with Generalisation. University of Waikato.
- McCallum, A., Corrada-Emmanuel, A. and Wang, X. (۲۰۰۵), “Topic And Role Discovery in Social Networks,” pp. ۷۸۶-۷۹۱.
- MOHAMMED ZAKI M., MEIRA W. (۲۰۱۴) , DATA MINING AND ANALYSIS Fundamental Concepts and Algorithms, Cambridge University
- Morris, S.A. and G.Yen, G. (۲۰۰۴), “Crossmaps: visualization of overlapping relationships in collections of journal papers,” PNAS, Vol. ۱۰۱, pp. ۵۲۹۱-۵۲۹۶.
- Mullins, Nicholas. ۱۹۷۳. Theories and Theory Groups in Contemporary American Sociology. New York: Harper and Row.
- Müller-Prothmann, Tobias (۲۰۰۶): Leveraging Knowledge Communication for Innovation. Framework, Methods and Applications of Social Network Analysis in Research and Development, Frankfurt a. M. et al.: Peter Lang, ISBN ۰-۸۲۰-۴۹۸۸۹-۰.
- Nallapati, R., Feng, A., Peng, F. and Allan, J. (۲۰۰۴), “Event threading within newstoppers’.”
- Newman, Mark (۲۰۰۳). "The Structure and Function of Complex Networks". SIAM Review ۵۶: ۱۶۷-۲۵۶. doi:۱۰.۱۱۳۷/S۰۰۳۶۱۴۰۳۴۲۴۸۰.Pdf(<http://www.santafe.edu/files/gems/paleofoodwebs/Newman۲۰۰۳SIAM.pdf>)
- Newman, M.E.J. (۲۰۰۴), “Coauthorship networks and patterns of scientific collaboration,” PNAS, Vol. ۱۰۱, pp. ۵۲۰۰-۵۲۰۵.
- Oja, E. (۲۰۰۲), “Unsupervised learning in neural computation,” Theoretical Computer Science, Vol. ۲۸۷ No. ۱, pp. ۱۸۷-۲۰۷. doi:۱۶/S۰۳۰۴-۳۹۷۵(۰۲)۰۰۱۶۰-۳
- Pregibon, D. (۱۹۹۹), “۲۰۰۱: a statistical odyssey.”
- Pulijala, A. and Gauch, S. (۲۰۰۴), “Hierarchical text classification’۲۰۰۴.”
- Salton, G., Wong, A. and Yang, C.S. (۱۹۷۵), “A vector space model for automatic indexing,” Communications of the ACM, Vol. ۱۸, pp. ۶۱۳-۶۲۰.
- Sander J., "Principles of Knowledge Discovery in Data: Clustering I", Department of Computing Science University of Alberta, Tutorial Slides, (۲۰۰۳)
- Sebastiani, F. (۲۰۰۲), “Machine learning in automated text categorisation,” ACM Computing Surveys, Vol. ۳۴, pp. ۱-۴۷.
- Shapiro, G.P. (۲۰۰۰), “Knowledge discovery in database: ۱۰ years after,” ACM SIGKDD exploration, Vol. ۱.

- Shiffrin, R.M. and Börner, K. (٢٠٠٤), "Mapping knowledge domains.," Proceedings of the National Academy of Sciences of the United States of America, Vol. ١٠١ Suppl No. suppl_١, pp. ٥١٨٣-٥. doi:١٠.١٠٧٣/pnas.٠٣٠٧٨٥٢١٠٠
- sImielinski, T. and Virmani, A. (١٩٩٩), "Data Mining and Knowledge Discovery Journal."
- Skupin, A. (٢٠٠٤), "The world of geography: Visualizing a knowledge domain with cartographic means," PNAS, Vol. ١٠١, pp. ٥٢٧٤-٥٢٧٨.
- Smith L. I. (٢٠٠٢) "A tutorial on principal components analysis"
- Steinbach, M., Karypis, G. and Kumar, V. (٢٠٠٠), "A comparison of document clustering techniques." Retrieved from rakaposhi.eas.asu.edu/cse٤٩٤/notes/clustering-doccluster.pdf
- Sun, A. and Lim, E. (٢٠٠١), "Hierarchical text classification and evaluation."
- Shiffrin, R.M. and Börner, K. (٢٠٠٤), "Mapping knowledge domains.," Proceedings of the National Academy of Sciences of the United States of America, Vol. ١٠١ Suppl No. suppl_١, pp. ٥١٨٣-٥. doi:١٠.١٠٧٣/pnas.٠٣٠٧٨٥٢١٠٠
- Tan P.N , Steinbach M.Kumar V. (٢٠٠٥) "Chapter ٢:Data", Introduction to datamining,Addison-Welsey.
- Wasserman, Stanley, & Faust, Katherine. (١٩٩٤). Social Networks Analysis: Methods and Applications.Cambridge: Cambridge University Press.
- Wasserman, Stanley / Faust, Katherine (٢٠٠٨): Social Network Analysis. Methods and Applications, Cambridge, University Press
- White, H.D., Lin, X., Buzydlowski, J.W. and Chen, C. (٢٠٠٤), "User-controlled mapping of significant literatures," PNAS, Vol. ١٠١, pp. ٥٢٩٧-٥٣٠٢.
- Watts, Duncan. (٢٠٠٣). Small Worlds: The Dynamics of Networks between Order and Randomness. Princeton :Princeton University Press.
- Watts, Duncan. (٢٠٠٤). Six Degrees: The Science of a Connected Age. W. W. Norton & Company.
- Wellman, Barry (١٩٩٩). Networks in the Global Village. Boulder, CO: Westview Press.
- Wellman, Barry. (٢٠٠١). "Physical Place and Cyber-Place: Changing Portals and the Rise of Networked
- Wellman, Barry and Berkowitz, S.D. (١٩٨٨). Social Structures: A Network Approach. Cambridge: Cambridge University Press.
- Ye N.(٢٠٠٣) "The handbook of Datamining"
- Zamir, O. and Etzioni, O. (١٩٩٨), "Web document clustering: a feasibility demonstration," pp. ٤٦-٥٤. Retrieved from <http://portal.acm.org/citation.cfm?id=٢٩٠٩٥٦>

- Zhao, y. and Karyapis, G. (۲۰۰۴), “Empirical and theoretical comparisons of selected criterion functions for document clustering,” Machine Learning, Vol. ۵۵, pp. ۳۱۱-۳۳۱.
- Zhao, Y. and Karypis, G. (۲۰۰۲), “Evaluation of hierarchical clustering algorithms for document datasets: technical report,”McLean.Retrievedfrom <http://www.cs.umn.edu/~karypis>
- Zhao, Y. and Karypis, G. (۲۰۰۵), “Hierarchical Clustering Algorithms for Document Datasets,” Data Mining and Knowledge Discovery, Vol. ۱۰, pp. ۲۵۸-۳۶۹. Retrievedfrom www.cs.umn.edu/~karypis
- Zhao, Y. and Karypis, G. (۲۰۰۵), “Topic- driven clustering for document databasets.”
- Zhao Y. (۲۰۱۳), R and Data Mining: Examples and Case Studies