LSE-PKU Summer School

Beijing, China

5-16 August 2019

LPS-MY201

Big Data: Data Analytics for Business and Beyond

Reading Pack

Contents Page

LSE-PKU Summer School 2019

LPS-MY201 | Big Data: Data Analytics for Business and Beyond

Instructor

Qiwei Yao, Professor of Statistics at London School of Economics and Political Science, Distinguished Visiting Professor at Guanghua School of Management of Peking University.

Qiwei Yao is a leading expert in high-dimensional time series analysis and nonlinear time series analysis. He is Fellow of the Institute of Mathematics Statistics, Fellow of the American Statistical Association, and Elected Member of the International Statistical Institute. His current research focuses on modelling and forecasting with vast time series data.

Qiwei Yao has undertaken extensive data analytics consultancy projects from major industry companies including Barclays Bank, Electricite de France (EDF), and Winton Capital Management Ltd.

Course Summary

In this modern information age, the broad availability of Big Data (i.e. data of unprecedented sizes and complexities) brings opportunities with challenges to business and beyond. Companies are focused on exploiting data for competitive advantages. Cyberspace communication reveals complex social interactions. Big Data surveillance is an effective way to detect actionable security threats. Data analytics is a subject of learning from data, of measuring, controlling, and communicating uncertainty, and of data-driven decision-makings (DDD). It will become ever more critical as businesses, governments and also academia rely increasingly on DDD, expanding the demand for data analytics expertise.

The primary goal of this course is to help you view various problems from business, science and social domains from a data perspective and understand the principles of extracting useful information and knowledge from data. You will also gain the hands-on experience using R– a programming language and software environment for data analysis and graphics. (R is free and available from http://www.r- project.org/.) The focus is on and

basic principles and concepts of data analytic methodology. We will also point out the limitation of data analysis: one should not be carried away by the findings from data and models. Common sense, intuition, domain knowledge and creativity often play roles in good data analytics.

To achieve this primary goal, inevitably we will introduce some basic data analytic methods and illustrate them with real-life examples (some from China). This is the second goal of the course. Data analytics has multiple facets and approaches, encompassing diverse statistical techniques under a variety of names such as data mining, machine learning. The methods to be covered include:

- Classification. Among all customers of EDF, who are likely to switch to another energy supplier?
- Regression (i.e. value estimation.) How much will a given customer use the service?
- Similarity matching. Identify individuals who are similar to your most royal customer group.
- Clustering. How should our customer care teams be structured?
- Market-basket analysis. Should beers be placed next to baby napkins in a supermarket.
- Link prediction. As you and John share 10 friends, maybe you would like to be John's friend?
- Causal modelling. Is the increase of sales caused by a particular advertisement?
  This is not a course on algorithms and IT technologies required for handling massive data, which deserve separate courses. The focus is on the fundamental principles and concepts of data analytics or data science. It becomes ever-increasingly important in this information age to gain adequate understanding of data science even if you never intend to apply it yourself.

Course Overview

1. Introduction: data-analytic thinking, data mining for knowledge discovery, data science solution for business problems.

2. Predictive modelling: correlation and supervised learning, regression and classification, support- vector machines, overfitting and its avoidance.

3. Clustering data: similarity, nearest neighbours, unsupervised learning methods.

4. Decision analytic thinking: what is a good model, visualizing model performance, evidence and probabilities.

5. Additional topics: text mining and network data

Prerequisites

Knowledge of calculus and statistics at the undergraduate freshman level. Participants should also bring a laptop and a calculator (calculators will be needed in the final examination).

Assessment

Coursework (30%) and final exam (70%).

Recommended Preparatory Readings

[1] Provost, F. and Fawcett, T. (2013). Data Science for Business. O'Reilly. [2] Runkler, T.A. (2012). Data Analytics. Springer.

[3] James, G., Witten, D., Hastie, T. and Tibshirani, R. (2013). An Introduction to Statistical Learning with Applications in R. Springer.

Students may choose to read any one of the above three books. They are listed in the ascending order in terms of the technical level, as [3] is technically the most advanced. [3] also illustrates how to implement data analytic methods in R.

References for R:

[4] Venables N. et. al. (2015). An Introduction to R. is available online at http://cran.r-project.org/doc/manuals/R-intro.pdf.

[5] Zuur, A., Ieno, E. and Meesters, E. (2009). A Beginners Guide to R. Springer. [1]

BigData: Data Analytics for Business and Beyond

Qiwei Yao
Department of Statistics
London School of Economics

q.yao@lse.ac.uk

- Data analytic thinking
- Principles and concepts
- Illustration with R – a versatile statistical package

Chapter 1. Introduction: Data Analytics

Data Analytics: the science of examining data in order to draw conclusions. It enables companies and organization to make Data-Driven Decisions (DDD). It can be used to verify or disprove existing hypotheses and theories.

Data Science: extract information from data such as undiscovered patterns, hidden relationships etc.

Difference: not really, though Data Analytics focuses on application and conclusions while Data Science on info extracting and knowledge discovery. But both involve making inference from data.

Ability to view various problems from a data perspective, understand the principles for extracting info from data:

intelligence quotient emotional quotient data quotient

Additional reading: Chapters 1 &2 of Provos and Fawcett (2013).

It's all about data:

- Business data: on customers, portfolio, sales, marketing, pricing, financial, risk, and fraud. For example,
  - shopping basket analysis to increase sales and cross selling
  - customer segmentation for tailored advertising and sales promotions
  - consumer data across multiple service channels (branch, web, mobile)

- Industrial process data: automate and control industrial production, manufacturing, distribution, logistics and supply chain processes.
  - sensors and actuators at the field level
  - control signals at the control level
  - operation and monitoring data at the execution level
  - schedules and indicators at the planning level

- Financial market data: historical records over long time periods and with increasing granularity (e.g. high-frequency data, limit order book, continuous market data)

- Personal data: demographics/geographics (factual) information, personality (behaviour) information, psychographics (attitudinal) information etc

- Text and unstructured data: text documents, company reports, news, messages, emails, web based data bases (the so-called deep web), in order to filter, search, extract, and structure information.

- Image data: image sensors, smartphone, satellite cameras, to find and recognize objects, analyze and classify scenes, and relate image data with other information sources

- Biomedical data: lab experiment data, DNA sequences, to understand and annotate genome functions...

- ......

New York Times story from 2004: Hurricane Frances

Hurricane Frances was on its way, threatening a direct hit on Florida's Atlantic coast.

Executives at Wal-Mart stores see the situation offered an opportunity to use DDD to predict unusual sales pattern.

Water, flashlights — common sense, no DDD required.

By mining through the trillions bytes of sales history when Hurricane Charley struck earlier, it was revealed that the sales for strawberry Pop-Tars increase 7 times, also a particular DVD film sold out (a coincident?). But the pre-hurricane top-selling item was beer.

`http://www.nytimes.com/2004/11/14/business/yourmoney/14wal.html?_r=0`

Predicting Customer Churn

Churn: a customer switches from one company to another.

It is expensive all around: one company must spend on incentives to attract a customer while another company loses revenue when the customer departs.

A marketing budget is allocated to prevent churn, as attracting new customers is more expensive than retaining existing ones.

Suppose you work for a credit card company which lose about 20% of customers per annum. Your task is to devise a detailed plan (DDD) on how and which customers be offered a retention deal.

Home equity loans of Bank of America: Failing to attract enough good customers in spite of several mail campaigns

Bank could lower interest rates to bring in more customers at the expense of lowered earnings. Existing customers may switch to the lower rates, further depressing earnings. Assuming the original rates were reasonably competitive, lower rates might bring in the disloyal customers.

Business consultants based on their marketing expertise provide the insights:

- people with college-age children want loans to pay tuition fees
- people with high but variable incomes want loans to smooth out the income fluctuations

BoA store the data of its millions of customers in a large relational database on a powerful parallel computer from `Teradata`. The data were recorded from 1914. More recent records has about 250 attributes, including income, number of children, type of home, etc.

Decision trees derived rules to classify existing customers into two categories: likely or unlikely to respond to a home equity loan offer. This adds a new attribute to each individual in the database: `likely responder` or `unlikely responder` to a home equity loan.

Then cluster analysis is performed to automatically segment the customers into groups with similar attributes. The 14 clusters were found, and many of them did not seem particularly interesting. Nevertheless one cluster has two intriguing properties:

- 39% of the people in the cluster has both business of personal accounts
- the cluster contains more than 25% of the customers with the attribute `likely responder` to a home equity loan.

People might be using home equity loans to start businesses

Change the campaign message from 'use the value of your home to send your kids to college' to 'now the house is empty, use your equity to do what you've always wanted to do'

The response rate for home equity campaigns increased from 0.7% to 7%.

Gmail is capable of classifying users into 'millions of buckets'

A new business model: free services in exchange for personal information.

Google is the world's largest advertising company: After only 15 years in business Google makes more money from ads than all the world's newspapers combined.

750 million Gmail users in October 2014, 900 million in May 2015

In late 2010 two obscure trial lawyers in Texas made what was to them a momentous discovery: ads in Gmail are correlated with keywords contained in the emails. This triggers a long lawsuit, to seek for billions of compensation from Google. The case 'ends' in July 2014 with one out-of-court payment for a single plaintiff.

- Google holds adequate user consent
- Gmail does not make much money from ads: 70+% Gmail users never click on ads.

From its earliest days Gmail was intended to be a money-making product. Instead of relying on demographic information users provided about themselves at sign up, Gmail would attempt to grasp the actual meaning of user messages and target ads accordingly.

Gmail's limitless data mining ambitions: One patent filed in June 2003 described a lengthy series of message attributes that could be used in any combination to extract the meaning of an email and select the best ads to match it.

Gmail profiles all its users: an old idea on a new (gigantic) scale – tailoring messages to specific audiences increases the advertiser's return.

Google's advertising business can be viewed as a giant but *sparse* spreadsheet with hundreds of thousands of advertisers aligned across the top and hundreds of millions of users down the left side: more than 99.99% of the cells are empty.

Nielsen PRIZM, a system developed in 1970s, use sophisticated clustering algorithms to divide Americans into such marketer-relevant buckets as Upper Crust, Blue Blood, Young Digerati, Beltway Boomers, Rustic Elders, Back Country Folks and Hard Scrabble, among dozens of others.

Google: clusters extracted by the PHIL algorithm from documents the user has viewed (web pages, inbound emails) or created (outbound emails, social media posts).

Inbound emails to Gmail users are of particular value: the emails received obviously include messages from family and friends, social media notifications, newsletters subscribed and whatever commercial offers have made it through your spam filter settings. They also typically include a large amount of data-rich correspondence from institutions: banks, utilities, schools, tax authorities, internet providers, TV companies and, last but not least, online merchants such as Amazon, eBay or travel reservation sites where you have made purchases. Taken together, these inbound messages discriminate an individual from other users with a high degree of granularity.

What is Big Data?

From a *Google* search in 2012:

> data which are too extensive to permit iterative analysis: one-pass analysis is necessary
>
> data sets which standard database tools cannot handle
>
> data which exceeds 20% of the RAM of a given machine

From *Siri* in 2017:

> Information assets characterized by such a high volume, velocity and variety to require specific technology and analytic methods for its transformation into value

From *Wikipedia* in 2018:

> A term for data sets that are so large or complex that traditional data processing application software is inadequate to deal with them. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating and information privacy.

4V or 5V: Volume, Variety, Velocity, Value and Vanity.

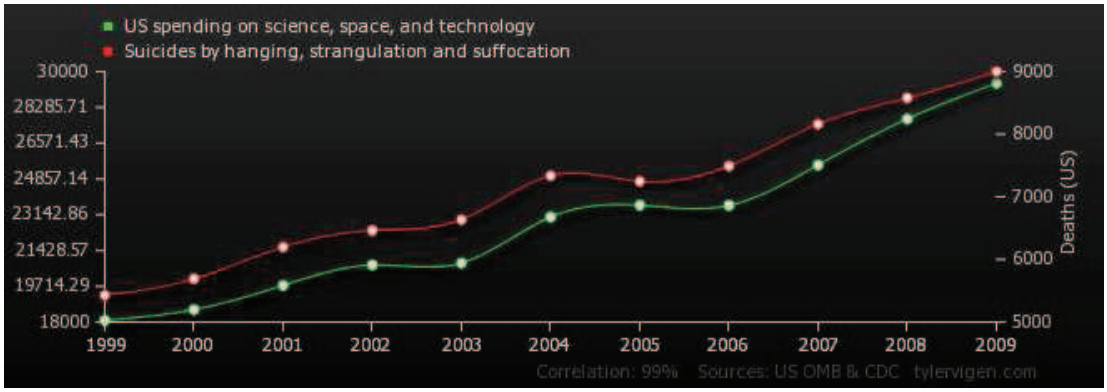
Volume: large size of data sets, or a large number of small data sets

Variety: lack of homogeneity in format, structure, quality

Velocity: high speed of data generation and data processing

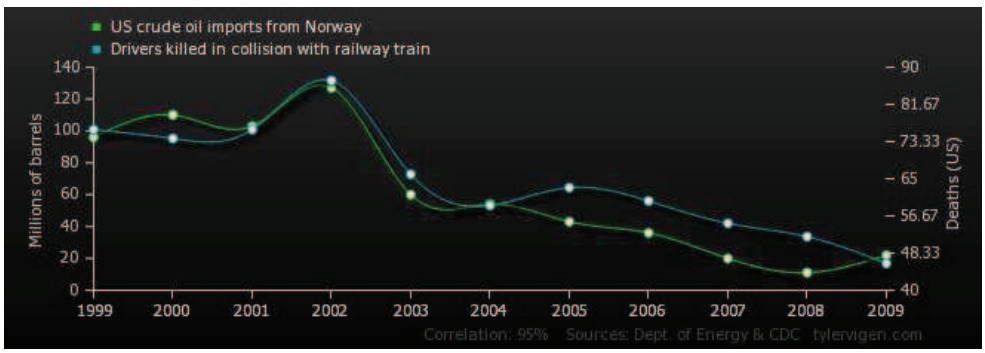
Value: valuable information buried in data sea

Vanity: 'everything' significant, spurious correlations

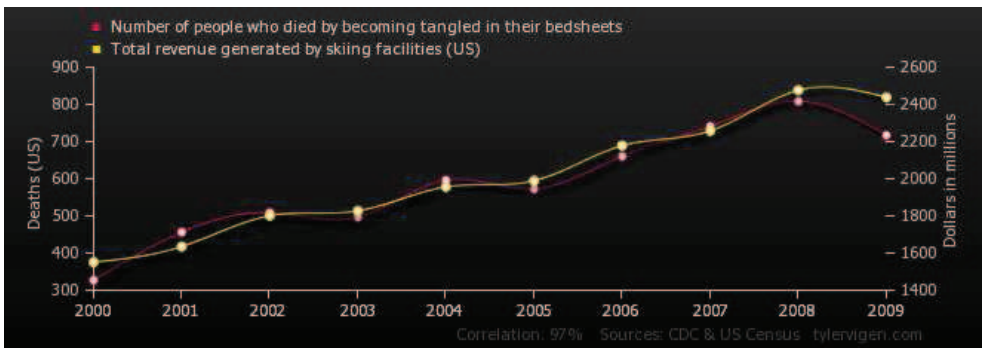


Correlation=0.992082

http://www.tylervigen.com/



Correlation=0.954509



Correlation=0.969724

Some big data stories

- Astronomy – Digital Sky Survey's archive in 2010: 140 terabytes
- Sequencing the Human Genome: 3.3 billion base pairs
- US equity markets: 7 billion shares change hands everyday
- Social network: quintillion (10^{18}) bytes per day
- Climate modelling: Coupled model intercomparison project 5th phase: more than 2 petabytes
- Google Translate: statistical machine translation; 200 billion words

Digit data expands quickly: doubling almost every 3 years

Big Data: not new!

- 1994: Wal-Mart, with over 7 billion transactions
- 1997: AT&T, with over 70 billion long distant phone call records
- 1990s: Mobil Oil, over 100 terabytes of data
- 2000: in just a few months the Sloan Digital Sky Survey collected more data than had previously been collected in the entire history of astronomy

Why now?

- automatic data capture in large scale (often secondary)
- exponential growth in computer memory and speed: making storage and computing cheap
- Data Analytics: Data → Information → Value → Profits

Is Require Big data Tech.

Value shifting from Physical infrastructure (land, factories) to intangibles such as brands, intellectual property and now data.

Why is it exciting?

A new world, according to many!

McKinsey & Company: *'we are on the cusp of a tremendous wave of innovation, productivity, and growth, as well as new modes of competition and value capture, all driven by big data as consumers, companies, and economic sectors exploit its potential'*

Chris Anderson: *'Out with every theory of human behavior, from linguistics to sociology. Forget taxonomy, ontology, and psychology. Who knows why people do what they do? The point is they do it, and we can track and measure it with unprecedented fidelity. With enough data, the numbers speak for themselves'.*

(From 'The end of theory: the data deluge makes scientific method obsolete' in Wired.)

But numbers do not speak for themselves!

= a half of 

per person & day

Data do not speak for themselves: positive or negative framing in data presentation can change the emotional impact.

A London Underground Poster (in 2011): '99% of young Londoners do not commit serious youth violence'

As 1% of young Londoners (aged between 15 and 25) $\approx$ 10,000 people, the message of the above poster is equivalent to: 'There are 10,000 seriously violent young people in London'.

☆ How to live with clever analytics.

What's the cancer risk from bacon scanwiches?

In November 2015, the International Agency for Research in Cancer (IARC) announced: processed meat is a 'Group I carcinogen', putting bacon in the same category as cigarettes

Daily Record (a Scottish newpaper) published a headline: Bacon, Ham and Sausages Have the Same Cancer Risk as Cigarettes Warn Experts

To quell the public panic caused, IARC stated: Group I classification is about being confident that an increased risk of cancer exists, $\cdots$ 50g processed meat a day is associated with an increased relative risk of bowel cancer of 18%.

肠

The increased absolute risk of bowel cancer is merely 1%!

The risks for non-bacon-eaters and bacon-eaters are, respectively, 6% and 7%. The relative risk is defined as relative odds-ratio: $\frac{7/93}{6/94} = 1.18$, therefore an increase of relative risk $0.18 = 18\%$.

More, Messy, Good enough: 3 shifts in extracting information

- More: More data with increasing granularity not only require more advanced IT techniques, but also change our way of thinking and ambition in analysing data
- Messy: data are complex in formats and structure, varies in quality
- Good enough: often satisfied with discovering correlations, patterns instead of causality — looking for *What* instead of *Why*

Some hedge funds parse Twitter to predict stock markets

Amazon and Netflix base product recommendations on a myriad of user interactions on their sites.

Twitter, LinkedIn, Facebook map users' social graph of relationships to learn their preferences.

Big Data: more data = more info? or more complexity and more noisy?

Joint force from computer science, statistics and applied mathematics is required to tackle the challenges with Big Data

Modern Data Analytics is typically teamwork!

Computer Science: manipulating data including capturing, storing, sorting, searching, selecting, aggregating, concatenating, etc

Statistics: extracting information from data, making inference

Applied Mathematics: complexity leading to many new mathematical challenges, and requiring new models and new algorithms

Big data does not mean end of 'small data'

David Hand's Power law for data set size: The probability of observing a data set of size n is inversely related to a power of n.

There are vastly more small data sets than very large ones, which are also important assets for data analytics.

'Big' is not necessarily more, useful, valuable, or interesting: it is possible to be data rich but information poor!

The future is not big data, but what we learn from it.

Your big data problem might be a small data problem in disguise, or be a large number of small data problems.

McKinsey & Company stated in 2011:

> 'there will be a shortage of talent necessary for organizations to take advantage of big data. By 2018, the United States alone could face a shortage of 140,000 to 190,000 people with deep analytical skills as well as 1.5 million managers and analysts with the know-how to use the analysis of big data to make effective decisions.'

Each data scientist needs 10 managers?

It is important to understand data science even if you never intend to apply it yourself.

Most of the problems we want to solve are inferential.

The ultimate goal of Big Data is to forecast future.

With the advancement of big data analytics, we would be able to predict accurately the likelihood of

– a malfunction of machine (service in advance)

– a heart attack (pay more for health insurance)

– default on a mortgage (be denied a loan)

– committing a crime (perhaps be arrested in advance???)

Some Ethical issues/legal challenges with Big Data

- individual privacy versus data collection, sharing and usage: require new rules to safeguard the sanctity of the individual
- personalised service versus exploring/manipulating
- incomprehensible nature of data-driven-decisions: the data dictatorship shifts the world from causation to correlation
- self-regulation versus global law enforcement
- .....

GDPR: The General Data Protection Regulation (EU) 2016/679 (`eugdpr.org`) became effective 25 May 2018.

Cambridge Analytica – A UK political consulting firm combining data mining, data brokerage, and data analysis with strategic communication during electoral processes

Started in 2013, and closed down in May 2018 due to the Facebook-Cambridge Analytica data scandal

> Today in the United States we have somewhere close to four or five thousand data points on every individual ... So we model the personality of every adult across the United States, some 230 million people.
>
> – Alexander Nix, CEO Cambridge Analytica, October 2016.

Nix claimed that CA provided service to

- 44 US political races in 2014
- Ted Cruz' presidential campaign in 2015
- Donald Trump's presidential campaign in 2016
- `Leave.EU` in 2016

CA's role in those campaigns is controversial, and is subject to criminal investigations in USA and UK.

Political scientists and the clients (including Trump's aides) also question CA's claims about the effectiveness of its methods of targeting voters.

CA's method: 'psychographic analysis' based on data enhancement and audience segmentation techniques such as *Big Five model* of personality, to 'fine your votes', and move them to action by personalized political adverts.

Big Five personality traits, also known as five-factor model or *OCEAN* or *CANOE* model:

- Openness to experience (inventive/curious vs. consistent/cautious)
- Conscientiousness (efficient/organized vs. easy-going/careless)
- Extraversion (outgoing/energetic vs. solitary/reserved)
- Agreeableness (friendly/compassionate vs. challenging/detached)
- Neuroticism (sensitive/nervous vs. secure/confident)

Channel 4 News investigation: lasted 4 months started in Nov 2017

An undercover reporter posed as a potential customer, hoping to help Sri Lankan candidates get elected. Video footage of the investigation was published on 19 March 2018. Within 7 weeks, CA collapsed as the "siege of media coverage has driven away virtually all of the Company's customers and suppliers".

From the footage, Nix said that his company uses honey traps, bribery stings, and prostitutes, for opposition research. He offered to discredit political opponents in Sri Lanka with suggestive videos using 'beautiful Ukrainian girls' and offers of bribes...

Cambridge Analytica immediately released a statement that the video footage was 'edited and scripted to grossly misrepresent' the recorded conversations and company's ethics and business practices.

On 17 March 2018, Christopher Wylie, LLB from LSE and former employee of CA, alleged in an interview with *Observer* that CA 'exploited Facebook to harvest millions of people's profiles' and used the data to target voters with personalized political adverts.

Personal data were collected via an app called 'This Is Your Digital Life' created by a Cambridge academic Dr Aleksandr Kogan. At the time, Facebook allowed app to collect data not only about app users but also their Facebook friends. As the result the personal data from 87 million people were acquired via the 0.27 million app users.

UK Information Commissioner's Office (ICO) confirmed:

> Dr Aleksandr Kogan and his company GSR, harvested the Facebook data of up to 87 million people worldwide, without their knowledge. A subset of this data was later shared with other organisations, including SCL Group, the parent company of Cambridge Analytica who were involved in political campaigning in the US.

In announcing its winding down in May 2018, Cambridge Analytica said it has 'unwavering confidence that its employees have acted ethically and lawfully', while the ICO said it would 'continue its civil and criminal investigation'.

On July 2018, several former Cambridge Analytica staff launched a new company 'Auspex International' in the field of data analytics and work in Africa and the Middle East initially.

`https://www.theguardian.com/news/series/cambridge-analytica-files`

One serious issue has raised from the story: almost irreconcilable tension between

- legal consent, i.e. ticking box for long and often incomprehensible consent and privacy statement, and
- moral and informed consent, i.e. what users actually feel comfortable with

In modern networked societies, privacy is a shared responsibility by individuals, friends, companies and governments across the global.

A critical skill in data science: decompose a data analytics problem into pieces such that each piece can be solved by a known or *newly invented* data-mining method. A data-mining task can be viewed as a process of learning from data by a computer, called machine learning, or by a statistical method, termed as statistical learning.

`Data mining, Information extraction, Knowledge Discovery: A craft`

Most frequently used data analysis methods

- Classification. *Among all customers of EDF, who are likely to switch to another energy supplier?*

- Regression (i.e. value estimation.) *How much will a given customer use the service?*

- Similarity matching. *Identify individuals who are similar to your most loyal customer group.*

→ unsupervised learning → search in the dark.

- Clustering. *How should our customer care teams be structured?*

- Market-basket analysis. *Should beers be placed next to baby napkins in a supermarket?*

- Link prediction. *As you and John share 10 friends, maybe you would like to be John's friend?*

- Causal modelling. *Is the increase of sales caused by a particular advertisement?*

- Network analysis. *How do social network structures affect, disease spreading, information dissemination, human behaviour?*

Most data-mining tasks can be divided into two categories:

- *Supervised learning*: learning with a set of "training data" with known labels (such as *Classification*) or relationships (such as *regression*).

- *Unsupervised learning*: learning without training data (such as *Clustering*, *Co-occurrence grouping*, *Profiling*).

{ only have x, don't have y
search in the dark.

In terms of purposes, data-mining tasks may be viewed as two types:

- Discovering/Inference: understand patterns and/or relationships within data.

  The Wal-Mart/Hurricane example: understand the sales pattern before hurricane

  Simple, transparent with easy interpretations

  Difficult with big and/or complex data (e.g. financial market).

- Predictive: learning results are used to predict unknown values.

  Which customers will respond to a particular ad?

  Prediction accuracy is of primary concern, 'black-box' methods are often used.

Artificial Intelligence (AI) and Deep Learning: the reincarnation since 2010 of neural networks (which was sidelined in the mid 1990s) due to

- massive improvements in computer resources
- some methodological innovations
- ideal niche learning tasks (image/video classification, speech and text processing)

AI support, or AI takeover?

Narrow AI: the tasks of which rules and sets of possibilities are unchanging

General AI: interact with and respond to multiple changing environments, requiring empathy, creativity, critical thinking, and dreaming – a distant prospect, with an unclear timeline for development

The AI arms race

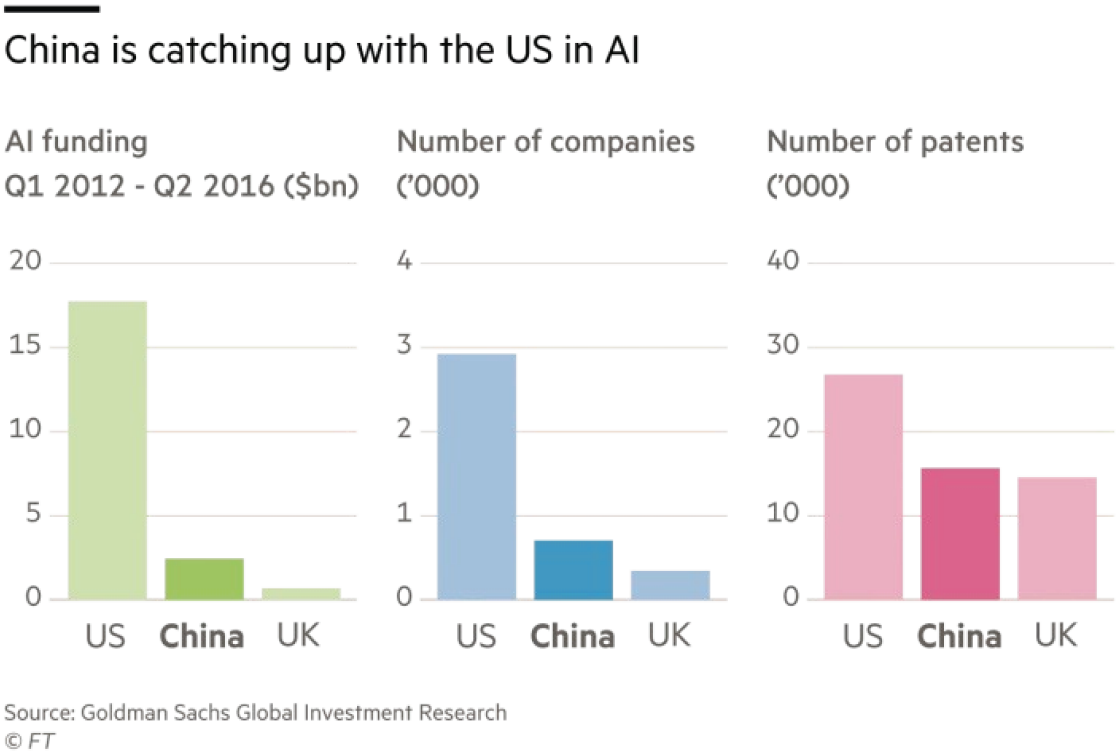
Vladimir Putin: "Whoever becomes the leader in this sphere (i.e. AI) will become the ruler of the world."

In April 2018, 25 EU counties signed an agreement to collaborate on AI with the investment target of 20 billion euros by the end of 2020 (`bit.ly/2HGJ3p9`).

The UK House of Lords report "AI in the UK: ready, willing and able?" (April 2018). It was convinced that the UK can lead in AI, building on a historically strong research programme (`bit.ly/2HGHhEv`).

The 3 key factors for AI development:

Data Algorithms/Scientists High-Tech



Five principles (UK House of Lords report):

1. AI should be developed for the common good and benefit of humanity

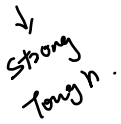
2. AI should operate on principles of intelligibility and fairness

3. AI should not be used to diminish the data rights or privacy of individuals, families or communities

4. All citizens have the right to be educated to enable them to flourish mentally, emotionally and economically alongside AI

5. The autonomous power to hurt, destroy or deceive human beings should never be vested in AI.

Robust AI and Interpretable AI



IT aspect of Data analytics – important but not covered in this course

It is hard to imagine a working data scientist who is not proficient with some software tools!

- Computer programming: packages (R, SAS, Matlab) and programming languages (Python, JAVA, C++, C#)

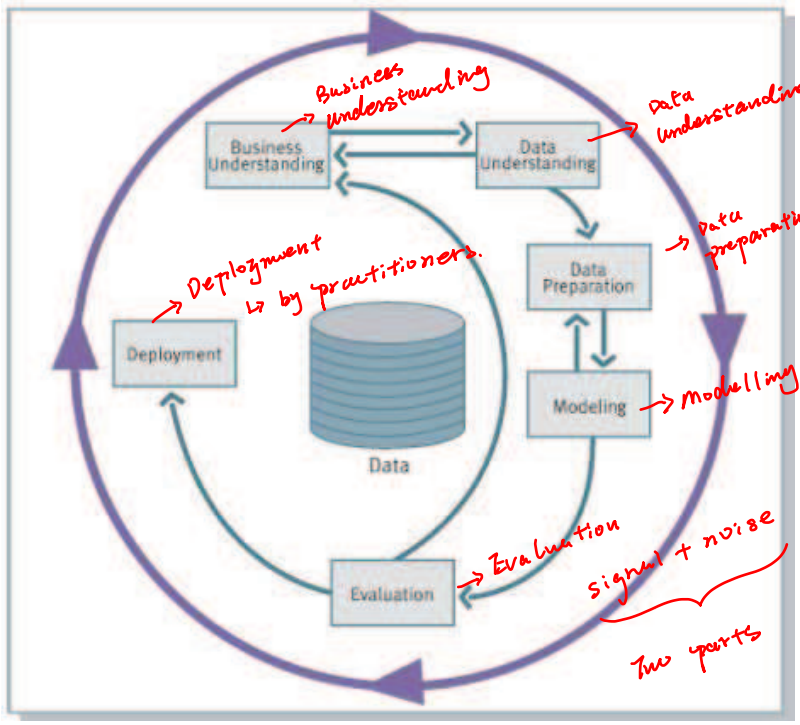
- Managing data: data cleansing, data structures, databases, data querying.

- Data visualisation (such as `Tableau`, `ggplot2`)

- Parallel computing and distributed processing (such as Hadoop/Spark)

Data analytical thinking: Extracting useful info from data can be treated systematically by following a process with reasonably well-defined stages. For example,

> CRISP-DM: the Cross Industry Standard Process for Data Mining.
>
> `https://the-modeling-agency.com/crisp-dm.pdf`



CRISP-DM Process diagram: iteration is the rule

CRISP data mining process:

1. *Business understanding*

   Understand the problem to be solved: business projects seldom come pre-packaged as clear and unambiguous. Recasting the problem and designing a data analytic procedure is typically an iterative process (see the diagram); requiring business knowledge, data analytic creativity/experience and common sense.

   - What exactly want to do?
   - How exactly do it?
   - Which statistical techniques/methods are relevant?

   This aspect will be further elaborated late in the course.

2. *Data understanding*

   Data – the raw material from which the solution will be built.

   Strengths and limitation of data?

   Data are typically collected without explicit purposes, are often with varying degrees of reliability.

   Costs and benefits of each data source: both collecting and analysing costs.

   For example, Data mining has been used extensively for *fraud detection*

   Catching credit card fraud: supervised data mining

   Catching medicare fraud: unsupervised data mining

3. *Data Preparation* — often proceeding along with data understanding

   Data cleaning, removing outliers, inferring missing values, converting data format, data transformation etc

4. *Modelling*: apply relevant machine/statistical learning methods

5. *Evaluation*: gain confidence by assessing modelling results

   If one looks hard enough at any dataset, one will find patterns – not survive careful scrutiny.

   Qualitative evaluation: model may be accurate in lab, but much less so in actual business context.

   Fraud detection, Spam detection etc typically produce too many false alarms. (How much would it cost to deal with the false alarms? What would be the cost in customer dissatisfaction?)

   Leading to a better business understanding – iterations

   Qualitative evaluation: make models comprehensible to stakeholders who need to 'sign off' before any deployment

6. *Deployment*

   Deployment the results and, increasingly, the data mining techniques themselves in real use, in order to adapt to the changing world.

   - A model used for real-life predicting the likelihood of churn to help churn management
   - A fraud detection model is used to alarm possible fraud cases
   - Online advertising and recommendation etc

Identifying likely buyers for a sport-utility vehicle: a case study
by Wei-Xiong Ho & Joseph Harder, Southern Illinois University

In 1992, one of the big three U.S. auto makers entrusted SIU to develop an *expert system* to identify likely buyers of a particular sport-utility vehicle.

The initial challenge was to improve response to a direct mail campaign for a particular model. The campaign involved sending an invitation to a prospect to come test-drive the new model. Anyone accepting the invitation would be offered a free pair of sunglasses.

Very few people were returning the response card or calling the toll-free number for more information, and few of those that did ended up buying the vehicle.

A lot of money could be saved by not sending the offer to people unlikely to respond!

Merging data sets

As is often the case when the data to be mined is from several different sources:

mail file : a mailing list containing names and addresses of about a million people who had been sent the promotional mailing

Appendix of *mail file* contains zip codes with demographic and *psychographic* characterizations of the neighborhoods associated with the zip codes.

response file : a list of people who sent back the response card

call file : a list of people who called the toll-free number from more information

sales file : a list of people who bought cars within the 3-months following the mailing, containing the info on names, addresses, model purchased

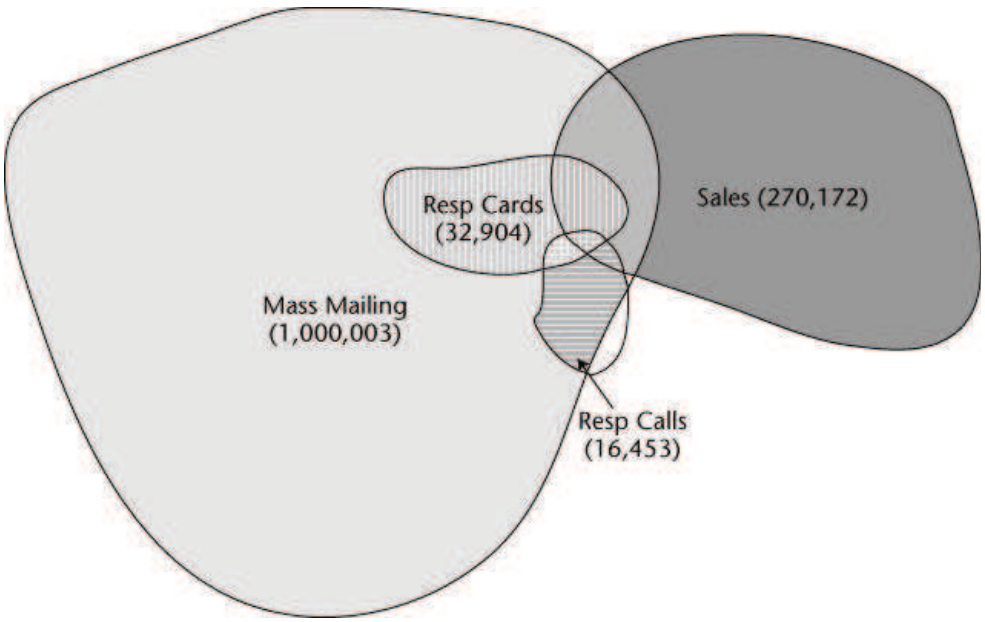
Linking the response cards back to the original mailing file was simple because the mail file contained a nine-character key printed on the response cards.

Telephone responders presented more of a problem since their reported name and address might not exactly match their address in the database, and there is no guarantee that the call even came from someone on the mailing list since the recipient may have passed the offer on to someone else.

The initial response rate 5%: Of 1,000,003 people who were sent the mailing, 32,904 responded by sending back a card and 16,453 responded by calling the toll-free number

The total sales in the period is 270,172. An automated name-matching program with loosely set matching standards discovered around 22,000

apparent matches between people who bought the car of the advertised model and people who had received the mailing. Hand editing reduced the intersection to 4,764 people.



Simple classification

success was defined as *received a mailing and bought the car* and failure was defined as *received the mailing, but did not buy the car.*

决策树 神经网络

A series of trials was run using decision trees and neural networks. The tools were tested on various kinds of training sets. Some of the training sets reflected the true proportion of successes in the database, while others were enriched to have up to 10 percent successes and higher concentrations might have produced better results.

☆ they have different training sets with differ probability of success.

The neural network did better on the sparse training sets, while the decision tree tool appeared to do better on the enriched sets.

稀疏的训练集

丰富训练集.

An improved two-stage approach: First, a neural network determined who was likely to buy a car, any car, from the company. Then, the decision tree was used to predict which of the likely car buyers would choose the advertised model.

This two step process proved quite successful. The hybrid data mining model combining decision trees and neural networks missed very few buyers of the targeted model while at the same time screening out many more nonbuyers than either the neural net or the decision tree was able to do.

Too simplistic! For example, people who test-drive one model, but end up buying another should be in a different class than nonresponders, or people who respond, but buy nothing. People who did not received ad but bought the car are in an even more interesting group.

Resulting actions

Armed with a model that could effectively reach responders the company decided to take the money saved by mailing fewer pieces and put it into improving the lure offered to get likely buyers into the showroom. Instead of sunglasses for the masses, they offered a nice pair of leather boots to the far smaller group of likely buyers. The new approach proved much more effective than the first.

Iterating the Cycle

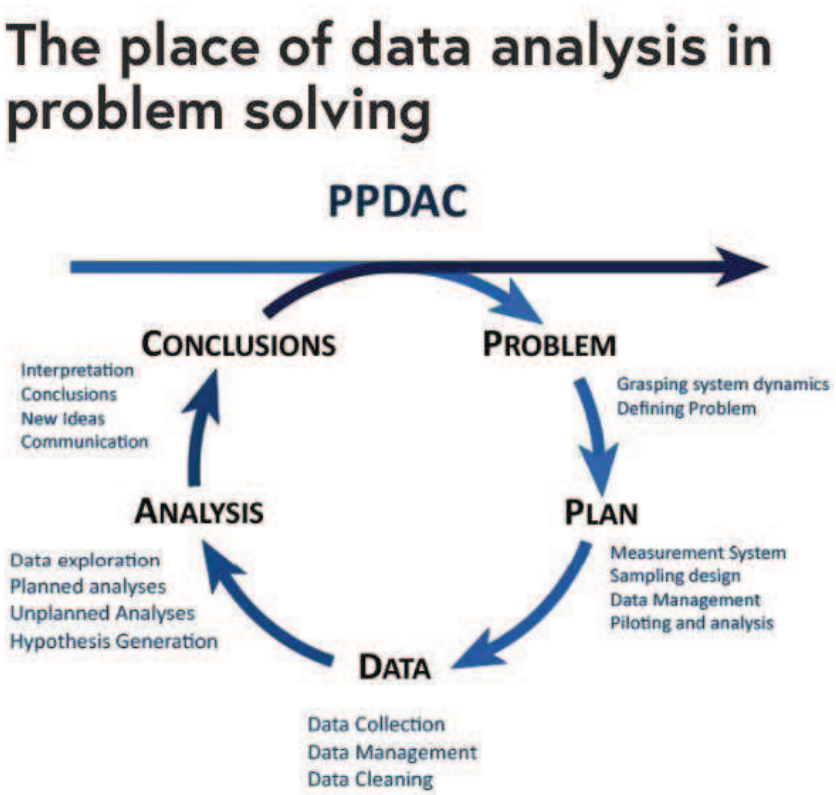
The approach described above used only a limited number of broad-brush variables and was crude and too simplistic by today's standard, in spite of its success in improving the effectiveness of a direct marketing campaign for a big-ticket item like an automobile.

The next step is to gather more data, build better models, and try again!

A cautionary remark. If you look too hard at data, you may find something which might not generalize beyond this particular data set

Overfitting, spurious correlation, incidental causality

Formulating data analytical solutions and evaluating the results involves thinking carefully about the context in which they will be used. Intuition, creativity, common sense, and domain knowledge can be brought to bear.



Descriptive Data Analysis in R (I)

What is *R*: an environment for data analysis and graphics based on *S* language

- a full-featured programming language
- freely available to everyone (with complete source code)
- easier access to the means of handling BigData such as parallel computation, Hadoop, distributed computation.
- official homepage: http://www.R-project.org

1.1 Installation

Installing *R*: R consists of two major parts: the base system and a collection of (over 10K) user contributed add-on packages, all available from the above website.

To install the base system, click on `download R` at http://www.R-project.org, then choose a mirror site close to you. Follow the link that describes your operating system: *Windows, Mac, or Linux*.

Note. The base distribution comes with many high-priority add-on packages such as graphic systems, linear models etc.

After the installation, one may start R by doubt-clicking the logo 'R' on your desktop in Windows or Mac. An R-console will pops up with a prompt character like '>'. You can input commands in the R language at the prompt.

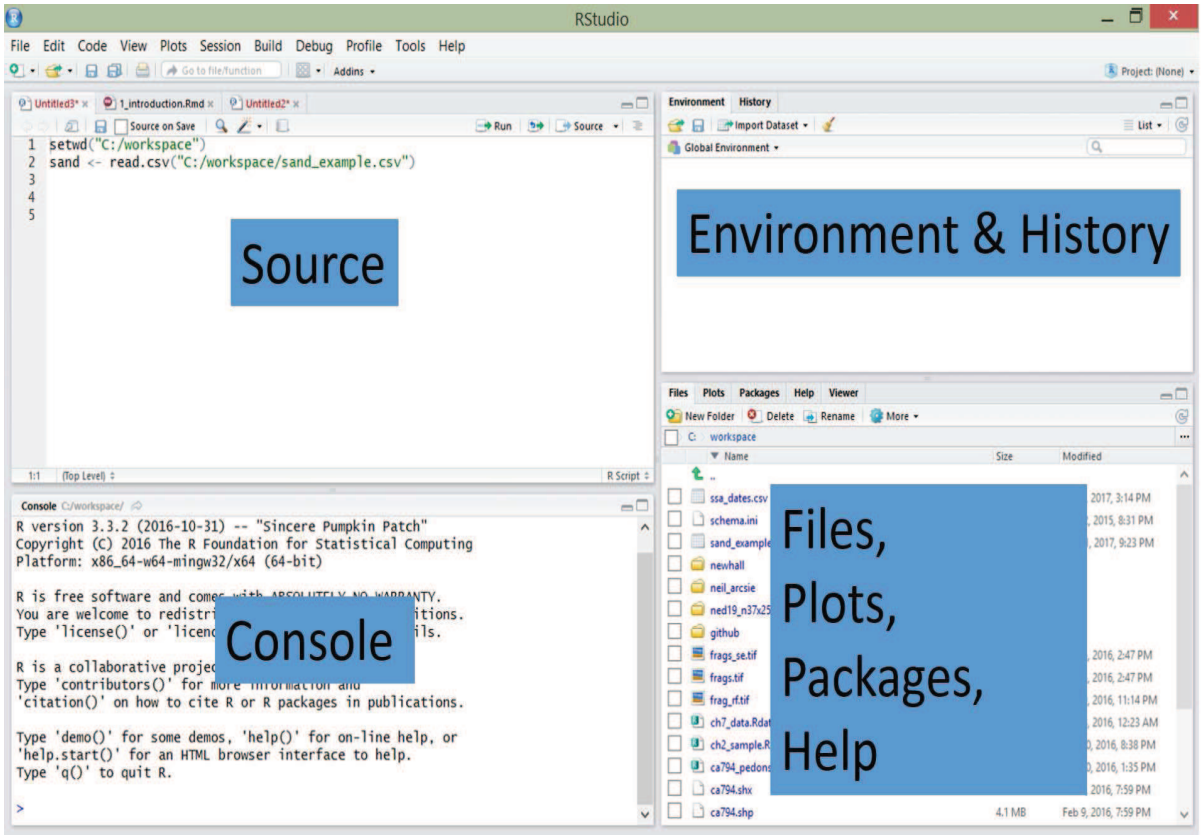
Nowadays most people use `RStudio` to use R, which is an application providing more user-friendly interface with R.

To install `RStudio`, click on `download` at `https://www.rstudio.com`. Once installed, you can open `RStudio` like any other program on your computer, usually by clicking an icon on your desktop.

When you open RStudio, a window appears with three panes in it. The largest pane is a console window. This is where you run R codes and see results.

The console window is almost exactly what you see if you ran R directly. The real work happens here.

Hidden in the other panes are a text editor, a graphics window, a debugger, a file manager, and much more.



R may be used as a calculators. Of course it can do much more. Try out in the console window:

```
> sqrt(9)/3 -1
```

To quit R or RStudio, type at the prompt '`q()`'.

To define a vector x consisting of integers $1, 2, \cdots, 100$

```
> x <- 1:100
> x
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
 [19]  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
 [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
 [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
 [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
 [91]  91  92  93  94  95  96  97  98  99 100
> sum(x)
> [1] 5050
```

Or we may also try

```
> y <- (1:100)^2
> y
  [1]     1     4     9    16    25    36    49    64    81   100   121   144
 [13]   169   196   225   256   289   324   361   400   441   484   529   576
 [25]   625   676   729   784   841   900   961  1024  1089  1156  1225  1296
 [37]  1369  1444  1521  1600  1681  1764  1849  1936  2025  2116  2209  2304
 [49]  2401  2500  2601  2704  2809  2916  3025  3136  3249  3364  3481  3600
 [61]  3721  3844  3969  4096  4225  4356  4489  4624  4761  4900  5041  5184
 [73]  5329  5476  5625  5776  5929  6084  6241  6400  6561  6724  6889  7056
 [85]  7225  7396  7569  7744  7921  8100  8281  8464  8649  8836  9025  9216
 [97]  9409  9604  9801 10000
> y[14]     # print out the 14-th element of vector y
```

```
[1] 196
```

One may also try `x+y`, `(x+y)/(x+y)`, `help(log)`, `log(x)` etc.

Additional packages can be installed directly from the R prompt. Information on the available packages is available at

http://cran.r-project.org/web/views/
http://cran.r-project.org/web/packages/

For example, one may install ggplot2 – a package for elegant graphics for data analysis.

```
> install.packages("ggplot2")
> library("ggplot2")  # To load all the objects in the package
                      #  into the current session
```

Note. In the bottom right window of RStudio, you may click on `Packages` $\rightarrow$ `Install` $\cdots$ to install the package to the same effect.

1.2 Help and documentation

To start help manual, click onn `help` alsp in the bottom right window. Then click on `Packages` to access the manuals for installed packages.

Alternatively online manual: `http://cran.r-project.org/manuals.html`

To access the info on an added-on package: `help(package="ggplot2")`

Quick access to the manual for 'mean': `help(mean)`, or `?mean`

Also try `?plot`, `?qplot`, `?sd`, `?summary`

R Newsletter: http://cran.r-project.org/doc/Rnews/

R FAQ: http://cran.r-project.org/faqs.html

Last but not least, `google` whatever questions often leads to most helpful answers

1.3 Data Import/Export

Working directory: a directory/folder from/to which R imports and outports files. You may change your working directory by clicking on

```
Session -> Setting Working Directory
```

with RStudio. For a direct R session, click on `File -> Change dir...`. For example, I create on my laptop `D:\bigData` as my working directory for this course.

The easiest form of data to import into R is a simple text file. The primary function to import from a text file is `scan`. You may check out what 'scan' can do: `> ?scan`

Create a plain text file 'simpleData', in your working directory, as follow:

```
This is a simple data file, created for illustration
of importing data in text files into R
1 2 3 4
5 6 7 8
9 10 11 12
```

It has two lines of explanation and 3 lines numbers. The R session below imports it into R as a vector `x` and 3×4 matrix `y`, perform some simple operations. Note the flag `skip=2` instructs R to ignore the first two lines in the file.

Note. R ignores anything after '#' in a command line.

```
> x <- scan("simpleData.txt", skip=2)
> x # print out vector x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
> length(x)
[1] 12
```

```
> mean(x); range(x) # write 2 commands in one line to save space
[1] 6.5
[1]  1 12
> summary(x)    # a very useful command!
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.75    6.50    6.50    9.25   12.00

> y <- matrix(scan("simpleData.txt", skip=2), byrow=T,
    ncol=4)
> y   # print out matrix y
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
> dim(y)    # size of matrix y
[1] 3 4
> y[1,]    # 1st row of y
[1] 1 2 3 4
```

```
> y[,2]    # 2nd column of y
[1]  2  6 10
> y[2,4]   # the (2,4)-th element of matrix y
[1] 8
```

A business school sent a questionnaire to its graduates in past 5 years and received 253 returns. The data are stored in a plain text file 'Jobs' which has 6 columns:

C1: ID number

C2: Job type, 1 - accounting, 2 - finance, 3 - management, 4 - marketing and sales, 5 -others

C3: Sex, 1 - male, 2 - female

C4: Job satisfaction, 1 - very satisfied, 2 - satisfied, 3 - not satisfied

C5: Salary (in thousand pounds)

C6: No. of jobs after graduation

```
IDNo.  JobType  Sex    Satisfaction  Salary     Search
1        1       1       3             51      1
2        4       1       3             38      2
3        5       1       3             51      4
4        1       2       2             52      5
... ...
```

We import data into R using command `read.table`

```
> jobs <- read.table("Jobs.txt"); jobs
          V1      V2  V3           V4     V5     V6
1        IDNo. JobType Sex Satisfaction Salary Search
2          1       1   1            3     51      1
3          2       4   1            3     38      2
4          3       5   1            3     51      4
   ... ...
> View(jobs)    # display data properly in the top left window
> dim(jobs)
[1] 254   6
> jobs[1,]
     V1      V2  V3           V4     V5     V6
1 IDNo. JobType Sex Satisfaction Salary Search
```

We repeat the above again by taking the 1st row as the names of variables (`header=T`) and the entries in 1st column as the names of the rows (`row.names =1`).

```
> jobs <- read.table("Jobs.txt", header=T, row.names=1)
> dim(jobs)
[1] 253   5
> names(jobs)
[1] "JobType"  "Sex"   "Satisfaction"   "Salary"  "Search"
> class(jobs)
[1] "data.frame"
> class(jobs[,1]); class(jobs[,2]); class(jobs[,3]);
   class(jobs[,4]); class(jobs[,5])
[1] "integer"
[1] "integer"
[1] "integer"
[1] "integer"
[1] "integer"
```

Since the first three variables are nominal, we may specify them as 'factor', while "Salary" can be specified as 'numeric':

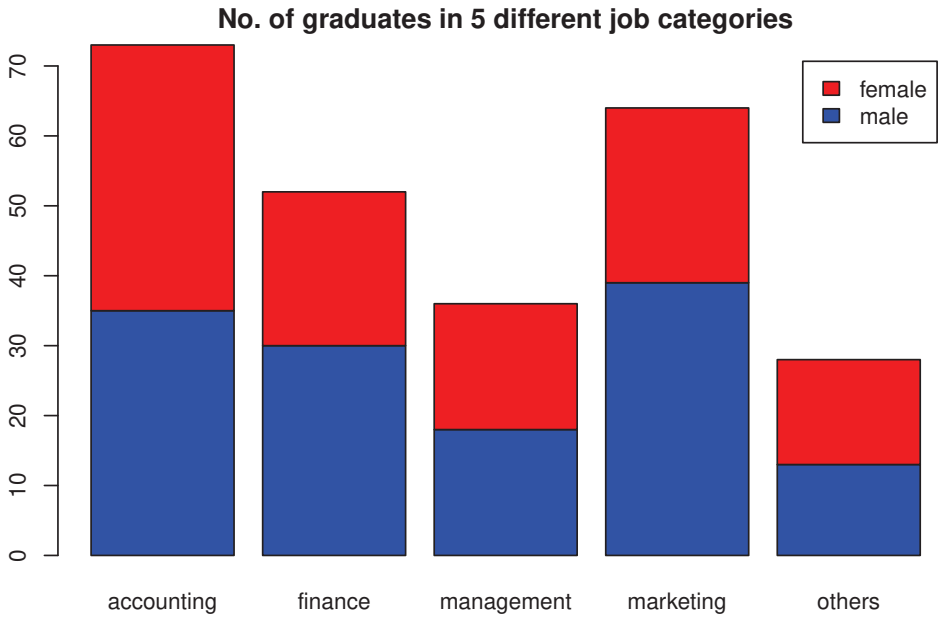
```
>  jobs <- read.table("Jobs.txt", header=T, row.names=1,
    colClasses = c("integer", "factor", "factor", "factor",
    "numeric", "integer"))
> class(jobs[,1]); class(jobs[,2]); class(jobs[,3]);
      class(jobs[,4]); class(jobs[,5])
[1] "factor"
[1] "factor"
[1] "factor"
[1] "numeric"
[1] "integer"
```

Note. we need to specify the class for the row name variable (i.e. 1st column) as well.

Now we do some simple descriptive statistical analysis for this data.

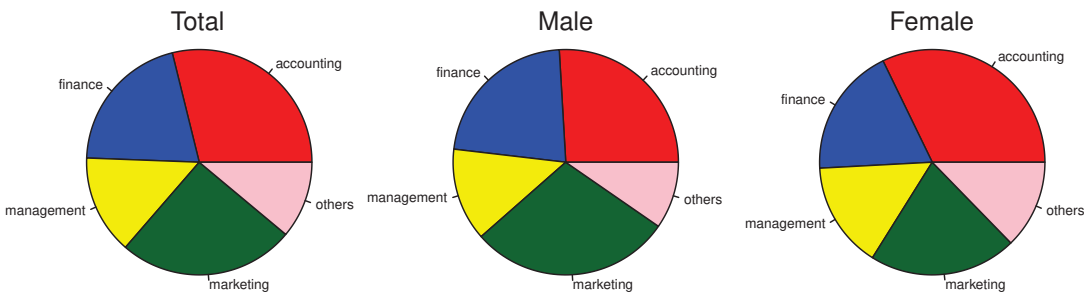
```
> table(jobs[,1])

 1  2  3  4  5
73 52 36 64 28   # No. of graduates with 5 different JobTypes
> t <-table(jobs[,2], jobs[,1], deparse.level=2)  # store table in t
> t
         jobs[, 1]
jobs[, 2]  1  2  3  4  5
        1 35 30 18 39 13    # No. of males with 5 different JobTypes
        2 38 22 18 25 15    # No. of females with 5 different JobTypes
> 100*t[1,]/sum(t[1,])
       1        2        3        4        5
25.92593   22.22222   13.33333   28.88889   9.62963
         # Percentages of males with 5 different JobTypes
> 100*t[2,]/sum(t[2,])
       1        2        3        4        5
32.20339   18.64407   15.25424   21.18644   12.71186
         # Percentages of females with 5 different JobTypes
> barplot(t, main="No. of graduates in 5 different job categories",
          legend.text=c("male", "female"), names.arg=c("accounting",
```

```
"finance", "management", "marketing", "others"))
# draw a bar-plot
```



The barplot shows the difference in job distribution due to gender. We may also draw pie-plots, which are regarded as less effective.

```
> pie(t[1,]+t[2,],label=c("accounting","finance","management",
      "marketing","others")); text(0,1, "Total", cex=2)
> pie(t[1,],label=c("accounting","finance","management",
       "marketing","others")); text(0,1, "Male", cex=2)
> pie(t[2,],label=c("accounting","finance","management",
       "marketing","others")); text(0,1, "Female", cex=2)
```



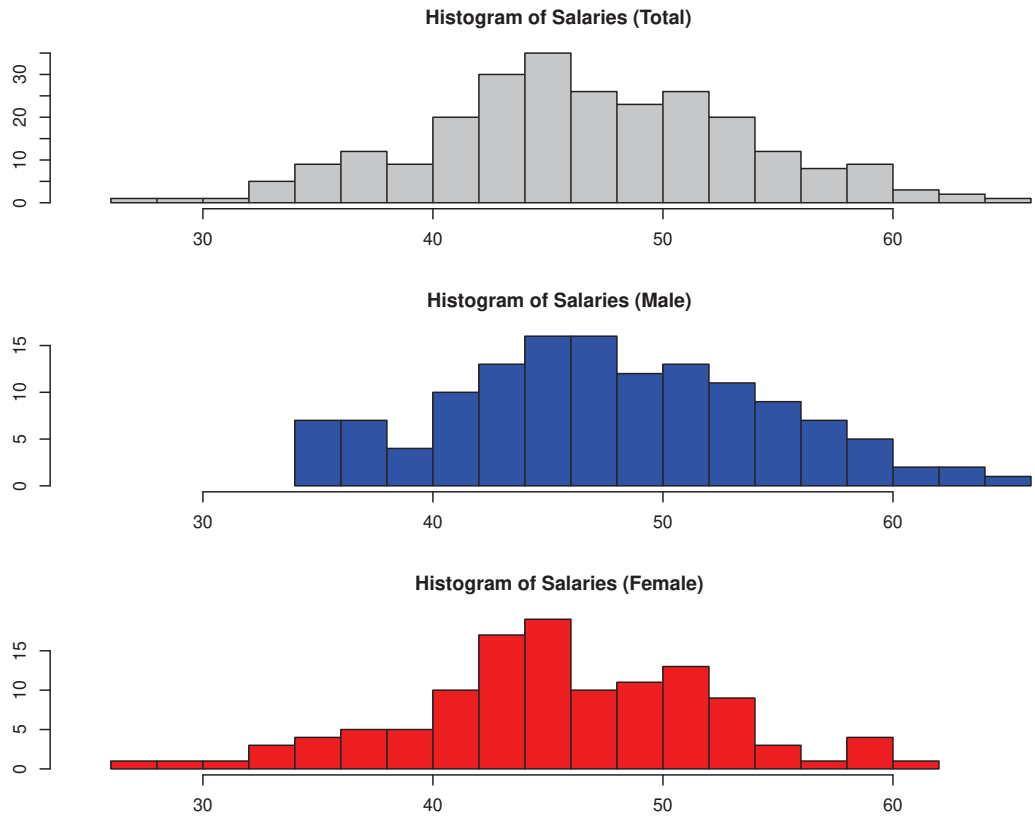
Now let look at the salary (`jobs[,4]`) distribution, and the impact due to gender.

```
> mSalary <- jobs[,4][jobs[,2]==1]
                # extract the salary data from male
> fSalary <- jobs[,4][jobs[,2]==2]
                # extract the salary data from female
> summary(jobs[,4]); summary(mSalary); summary(fSalary)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  26.00   43.00   47.00   47.13   52.00   65.00
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  34.00   44.00   48.00   48.11   53.00   65.00
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  26.00   42.25   46.00   46.00   51.00   61.00
> hist(jobs[,4], col="gray", nclass=15, xlim=c(25,66),
                main="Histogram of Salaries (Total)")
      # plot the histogram of salary data
> hist(mSalary, col="blue", nclass=15, xlim=c(25,66),
```



```
                main="Histogram of Salaries (Male)")
> hist(fSalary, col="red", nclass=15, xlim=c(25,66),
                main="Histogram of Salaries (Female)")
```

You may also try stem-and-leaf plot: `stem(jobs[,4])`



To export data from R, use `write.table` or `write`.

To write `jobs` into a plain text file 'Jobs1.txt':

```
> write.table(jobs, "Jobs1.txt")
```

which retains both the row and column names. Note the different entries in the file are separated by spaces.

We may also use

```
> write.table(jobs, "Jobs2.txt", row.names=F, col.names=F),
> write.table(jobs, "Jobs3.txt", sep=",")
```

Compare the three output files.

Note that the values of factor variables are recorded with " ". To record all the levels of factor variables as numerical values, we need to define a pure numerical data.frame first:

```
> t <- data.frame(as.numeric(jobs[,1]), as.numeric(jobs[,2]),
        as.numeric(jobs[,3]), jobs[,4], jobs[,5])
> write.table(t, "Jobs4.txt")
```

The file "Jobs4.txt" contains purely numerical values.

Note. (i) (i) Working directory — a directory/folder from/to which R imports and outports files. You may change your working directory by clicking on

```
Session -> Setting Working Directory
```

(ii) Saving a session — when you quit an R session `q()`, you will be offered an option to 'save workspace image'. By clicking on "yes",

you will save all the objects (including data sets, loaded functions from added-on packages etc) in your R session. You may continue to work on this session by directly double-clicking on the image file (with the last name `RData`) in your working directory.

Alternatively you may save work done in an R session including all objects, use "save.image" in console window:

```
> save.image("filename.RData")
```

the file must have "RData" as its last name.

To save all the commands used in an R session only, type in console

```
> savehistory("filename.Rhistory")
```

the file must have "Rhistory" as its last name.

A useful tip: Create a separate working directory for each of your R projects.

1.4 Organising an Analysis

An R analysis typically consists of executing several commands. Instead of typing each of those commands on the R prompt, we may collect them into a plain text file – an R-script. It can be resulted from editting an `Rhistory`-file saved from `savehistory`. For example, the file "jobsAnalysis.r" in my working directory reads like:

```
jobs <- read.table("Jobs.txt", header=T, row.names=1)
        # File "Jobs.txt" is in the working directory now
mSalary <- jobs[,4][jobs[,2]==1]
fSalary <- jobs[,4][jobs[,2]==2]
summary(jobs[,4])
summary(mSalary)
summary(fSalary)
par(mfrow=c(3,1))   # display 3 figures in one column
hist(jobs[,4], col="gray", nclass=15, xlim=c(25,66),
         main="Histogram of Salaries (Total)")
hist(mSalary, col="blue", nclass=15, xlim=c(25,66),
        main="Histogram of Salaries (Male)")
hist(fSalary, col="red", nclass=15, xlim=c(25,66),
        main="Histogram of Salaries (Female)")
```

You may carry out the project by sourcing the file into an R session:

```
> source("jobAnalysis.r", echo=T)
```

Also try `source("jobAnalysis.r")`.

Editing, including cleaning up, and saving the commands from the current session can be easily done by clicking on `History` in the top right window (i.e. Environment window of RStudio). Then highlight those you would like to save, and click on the buttom of `To Source`. All the highlighted contents will be displayed in the top left window. You can then further edit and save them into a file (i.e. an R-script) which can used again later.

To execute the commands in "jobAnalysis.r", open the file by clicking on `File -> Open File`. The contents of the file will be displayed in the top left window. Highlight the commands you like to execute, click on `Run`-buttom.

Chapter 2. Data Visualization and Preparation

- Data visualization
- Tidy data: preparation and cleaning (`tidyverse`)
- Data manipulation and extraction with `dplyr`
- Error Handling
- Data transformation

To learn more about R techniques in handling data, read and practise along with Parts I & II of Wichham and Grolemund (2017).

Introduction to Data

Data lies at the root of modern technological and methodological advancements.

The goal of data collection is to answer scientific, social or business questions.

Though the question of interest should be specified prior to data collection, the preliminary phase of data analysis typically involves solidifying the question to be as precise as possible, often taking into account the limitations of the available data.

This process can be repeated as many times as desired, as articulated by George Box: data analysis itself is an iterative and sequential process.

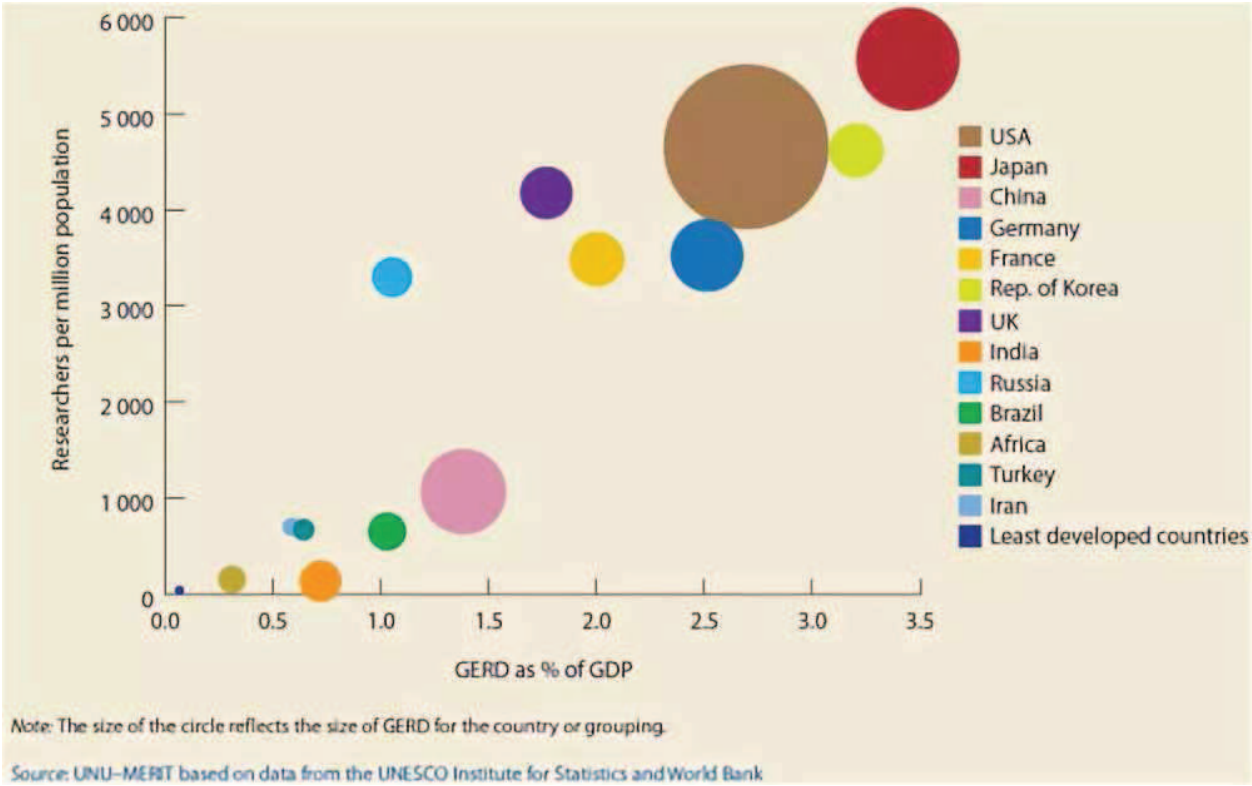
Data visualization: to help people understand the significance of data by placing it in a visual context: Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier when displayed visually.

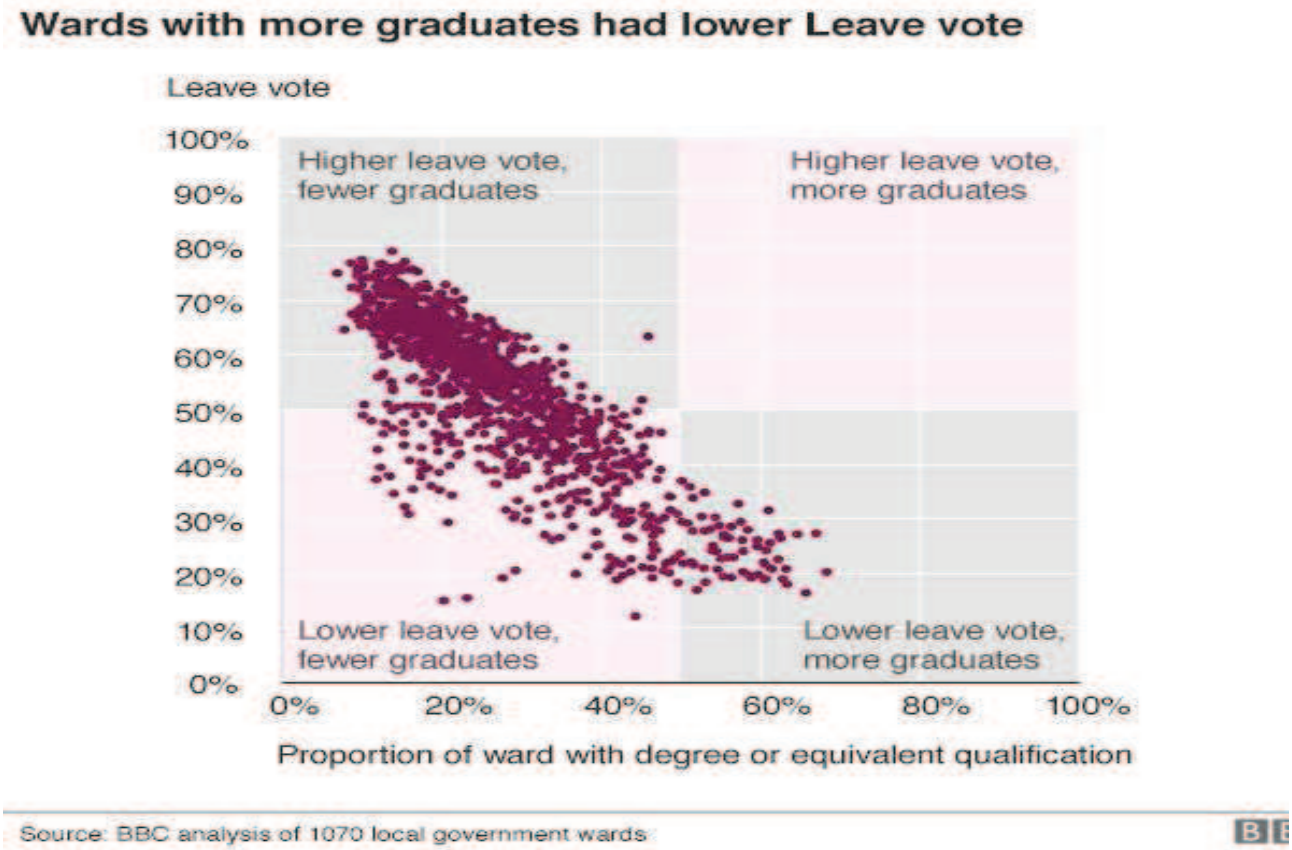
An effective visual display is often a powerful tool for data analysis.

Hans Rosling (Gapminder inventor): find catchy ways to illustrate statistics, and

> Having the data is not enough, I have to show it in the ways people both enjoy and understand.

Watch "The Joy of Statistics" on YouTUbe


Researchers per million population
6 000
5 000
4 000
3 000
2 000
1 000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
GERD as % of GDP
USA
Japan
China
Germany
France
Rep. of Korea
UK
India
Russia
Brazil
Africa
Turkey
Iran
Least developed countries
Note: The size of the circle reflects the size of GERD for the country or grouping.
Source: UNU-MERIT based on data from the UNESCO Institute for Statistics and World Bank


Wards with more graduates had lower Leave vote
Leave vote
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Higher leave vote, fewer graduates
Higher leave vote, more graduates
Lower leave vote, fewer graduates
Lower leave vote, more graduates
0%
20%
40%
60%
80%
100%
Proportion of ward with degree or equivalent qualification
Source: BBC analysis of 1070 local government wards

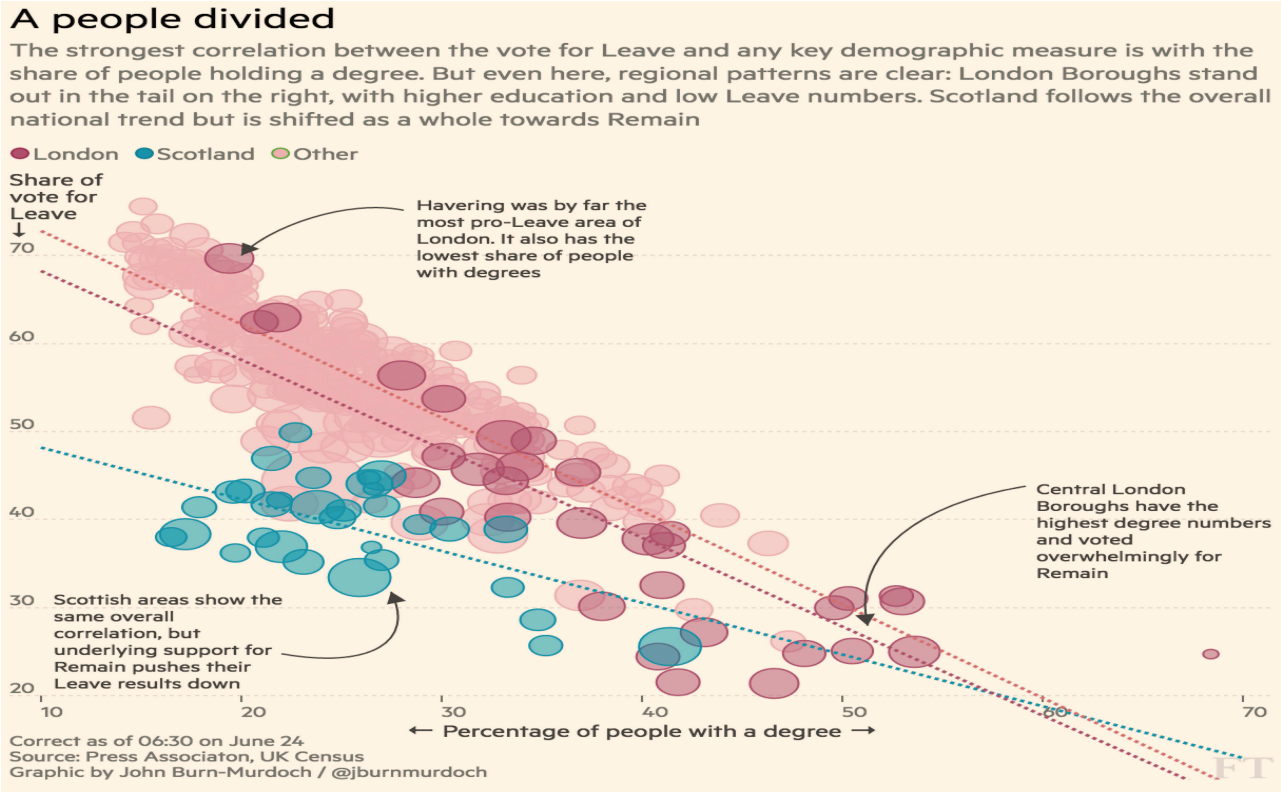
Some comments by public:

- It seemed strange to split the x-axis at 50%. Surely this should have been split at the median of the data?

- The only real criticism that comes immediately to attention is the overplotting of the points in the center: that makes it difficult to assess the numbers of points there, which makes the plot a little less useful than it could be

- the graph is misleading in a sense that it purports to show that there are no data points in the quadrant categorically described as high leave vote %, high % of graduates. What is high and low becomes relative to the axis limits, not the actual data. A more objective way to visualise this data would be to set the scatter plot axis limits at the maxmin of the data and then divide the chart into quadrants of an equal area.

- I view this as a clear informative plot. The message is immediately apparent and is not misleading. The BBC has plotted the actual data points. They have not manipulated the x or y axes. The annotation on the plot is correct and not over-stated. They have not added spurious trend lines or any other unnecessary interpretation. Compared to most data figures presented in the media, this plot is excellent – it is quite a good example of letting the data speak for itself. You can find some ways to improve the plot, but simple is usually best.

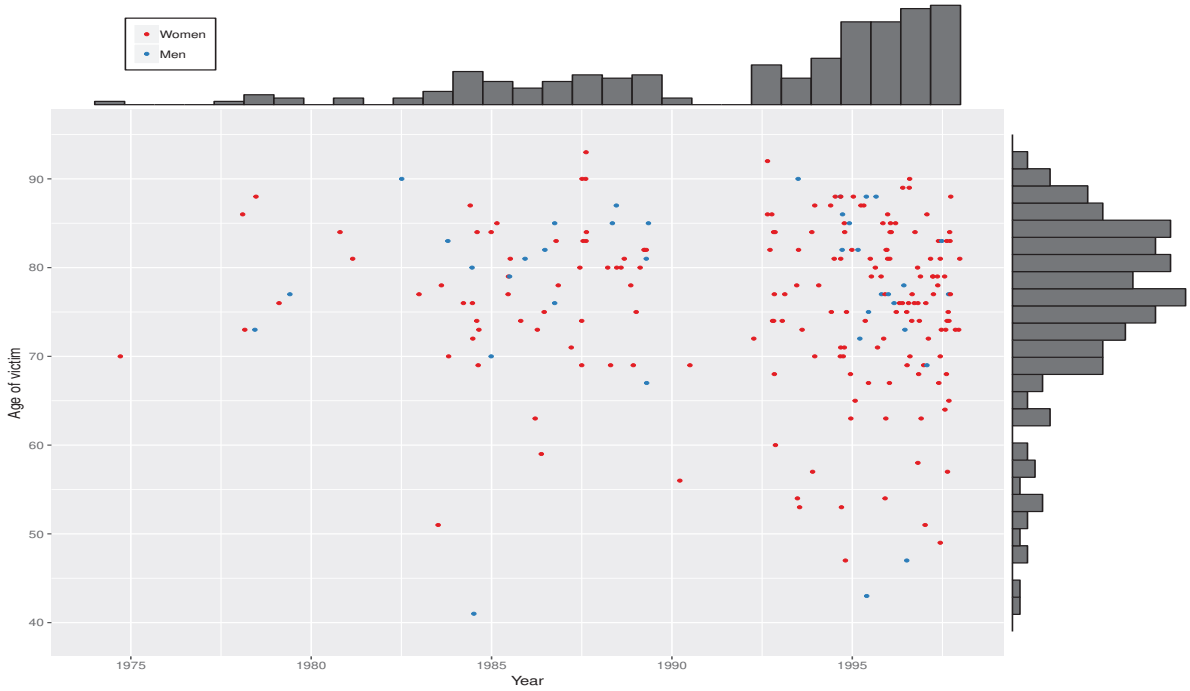


Harold Shipman was Britain's most prolific convicted murderer: between 1975 and 1998 he injected at least 215 of his mostly elderly patients with a massive opiate overdose.

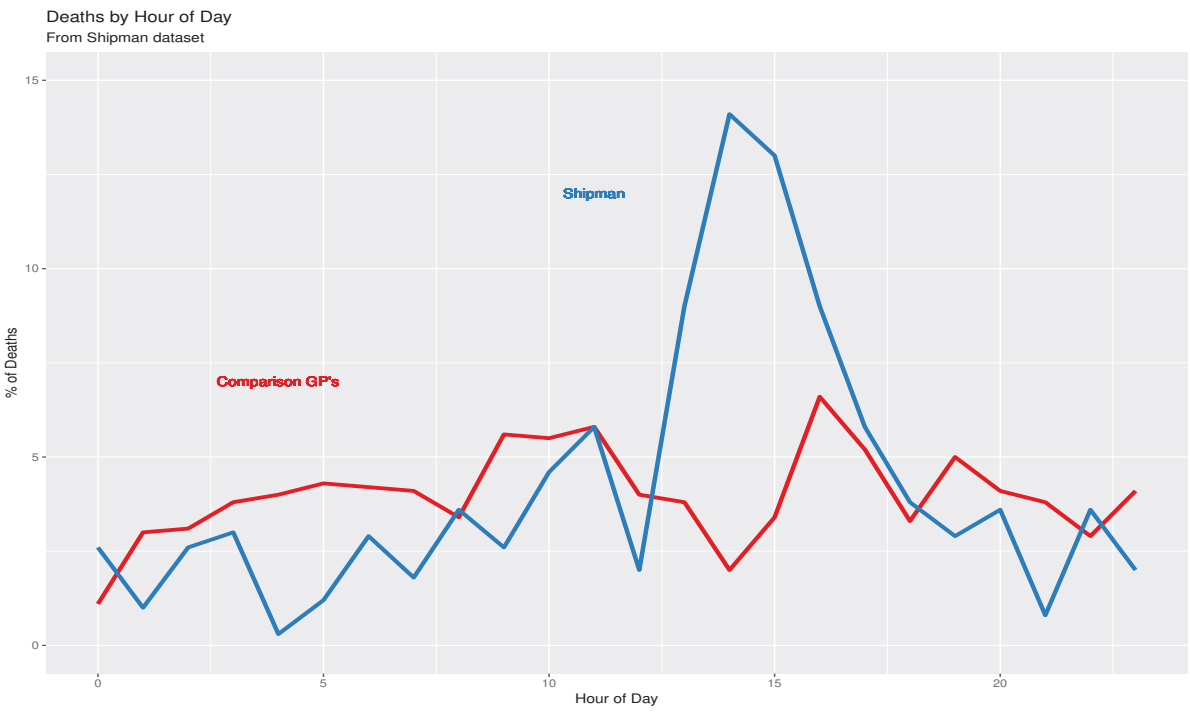
He finally made the mistake of forging the will of one of his victims so as to leave him some money: her daughter was a solicitor, suspicions were aroused, and forensic analysis of his computer showed he had been retrospectively changing patient records to make his victims appear sicker than they really were.

He was well known as an enthusiastic early adopter of technology, but he was not tech-savvy enough to realize that every change he made was time-stamped.

Of his patients who had not been cremated, 15 were exhumed and lethal levels of diamorphine, the medical form of heroin, were found in their bodies. Shipman was subsequently tried for 15 murders in 1999, but chose not to offer any defence and never uttered a word at his trial. He was found guilty and jailed for life.



A scatter-plot showing the age and the year of death of Shipman's 215 confirmed victims. Bar-charts at the top and on the right to reveal the pattern of victims' ages and the pattern of years in which he committed murders



The time at which Harold Shipman's patients died, compared to the times at which patients of other local general practitioners died. The pattern does not require sophisticated statistical analysis.

A public inquiry was set up to determine what crimes he might have committed apart from those for which he had been tried, and whether he could have been caught earlier. A number of statistician were called to give evidence at the public inquiry, which concluded that he had definitely murdered 215 of his patients, and possibly 45 more.

His reasons for committing these murders have never been explained: he gave no evidence at his trial, never spoke about his misdeeds to anyone, including his family, and committed suicide in prison, conveniently just in time for his wife to collect his pension.

It is common in this modern age that collect data first, and ask questions later, as data collection becomes easy, cheap and automated especially for big companies such as Google, Facebook, and also for supermarkets, banks, hospitals etc.

Nevertheless collecting relevant and high quality data which are readily usable for answering the questions of interest is not easy, or difficult, as data comes in many shapes and sizes, some data are messy, confusing with many missing values.

Data are often not in a 'tidy' format which can be readily employed for analysis.

Tidy data format is in the form of matrix/spreadsheet: (i) each variable is a column, (ii) each observation is a row, and (iii) each value is a cell.

To extract the relevant information from data is challenging, hence *Data Science* becomes one of the most important scientific disciplines in this information era.

Data Preparation and Cleaning – necessary and time-consuming!

Important but how?

`tidyverse` is a collection of R packages for data science, including core packages:

- `tidyr`: provides functions `gather, spread, separate, unite` for tidying data
- `dplyr`: provides functions `filter, arrange, select, mutate, summarise` etc for data manipulation/transformation
- `ggplot2`: for creating graphics and data visualization
- `readr`: contains functions `read_csv, read_csv2, read_tsv, read_delim` for importing data fast and friendly, and `parse_*` for parsing various types of data
- `tibble`: a modern re-imagining of the data frame which is lazy and surly: do less and complain more; leading to cleaner and more expressive codes

Installation: `install.packages(tidyverse)`

References: Grolemund and Wickham (2017), and many online sources such as

`https://www.tidyverse.org/`

1. **Tidying the data structure**

Put data into a tidy format (e.g. a dataframe/tibble in R):

Each variable/attribute forms a column

Each observation forms a row

Each value has its own cell

As an illustration, consider the *Communities and Crime* data set downloaded from UC-Irvine Machine Learning Repository (`archive.ics.uci.edu/`

Source: socio-economic data for communities in the US obtained from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR. The data consist of 2,215 communities, and each has 147 attributes.

Question to ask: what features of a community affect the violent crime rates

Open an RStudio session, specify a working directory, and put in the working directory 3 files: `communityCrimeData.csv`, `communityCrimeInfo.pdf` `communityCrimeAttributes.txt`.

There are two data files:

- `communityCrimeData.txt` – a table of 2215×147 entries
- `communityCrimeAttributes.txt` – names and definitions of 147 attributes

More detailed information can be found at

`archive.ics.uci.edu/ml/datasets/Communities+and+Crime+Unnormalized#`

or in the file `communityCrimeInfo.pdf`.

To import data,

```
> CC=read.csv("communityCrimeData.csv", header=F)
> dim(CC)
[1] 2215  147
```

`read.csv` is a special case of `read.table` for reading 'comma separated value' files. Note `read_csv` in `readr` could be about 10 times faster in

importing big data files. However data are imported as `tibble` instead of standard `data.frame`.

To view the whole data set, `View(CC)`.

We also need to attach the attribute names to each columns. The information on the attributes is in another file

`communityCrimeAttributes.txt`

which has more than 2215 lines, and each line has different number of entries (separated by space). The names of attributes are the 1st entry in each line. The file looks like

```
Attribute Information
(125 predictive, 4 non-predictive, 18 potential goal)
communityname  Community name - not predictive - for information only (string)
state  US state (by 2 letter postal abbreviation)(nominal)
countyCode  numeric code for county - not predictive, and many missing values (numeric)
communityCode  numeric code for community - not predictive and many missing values (numeric)
fold  fold number for non-random 10 fold cross validation, potentially useful for debugging, paired t
population  population for community (numeric - expected to be integer)
householdsize  mean people per household (numeric - decimal)
... ...
```

We need to input this unstructured file into R

```
> ccAttr=read.delim("communityCrimeAttributes.txt", header=F,
           sep=" ", skip=2)
> dim(ccAttr) # show size 147 x 23
> names(CC)=ccAttr[,1] # assign column names to file CC
> View(CC)
```

2. Data manipulation and extraction

Now we manipulate the dataset using the package `dplyr`

```
> library(dplyr)  # upload the package to the current session
> tbl_df(CC)
```

`tbl_df` displays only the first 10 rows and the number of columns fit to the display.

To extract the communities with population $\geq$ 800K

```
> bigPopul=filter(CC, population>=800000)
                      # slect rows with population >= 800K
```

```
> dim(bigPopul)
[1]  10 147
> bigPopul[,1]
 [1] NewYorkcity      Philadelphiacity LosAngelescity   Dallascity
 [5] Detroitcity      Houstoncity      SanDiegocity     SanAntoniocity
 [9] Chicagocity      Phoenixcity
```

murdPerPop records the number of murders per 100K population

```
> attach(CC) # make the columns recognizable in R
> summary(murdPerPop)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   2.170   5.859   8.365  91.090
> tt=filter(CC, population>=800000, murdPerPop>8.365)
> tt[,1]
 [1] NewYorkcity      Philadelphiacity LosAngelescity   Dallascity
 [5] Detroitcity      Houstoncity      SanDiegocity     SanAntoniocity
 [9] Chicagocity      Phoenixcity
```

```
> tt=filter(CC, population>=800000, murdPerPop>20)
> tt[,1]
[1] NewYorkcity      Philadelphiacity LosAngelescity   Dallascity
[5] Detroitcity      Houstoncity      SanAntoniocity   Chicagocity
> tt=filter(CC, population>=800000, murdPerPop>30)
> tt[,1]
[1] LosAngelescity Dallascity     Detroitcity    Chicagocity
```

To extract columns of interest for those 10 cities with big population:

```
> big10C = select(bigPopul, communityname, PctKidsBornNeverMar, racepctblack, pctWPubAsst,
            TotalPctDiv, PctUnemployed, ViolentCrimesPerPop)
> big10C
   communityname PctBornNeverMar racepctblack pctWPubAsst TotalPctDiv PctUnemployed VioCrimePerPop
     NewYorkcity          10.50        28.71       13.12       11.77          8.98        2097.71
Philadelphiacity          11.53        39.86       13.98       12.53          9.62         1279.6
  LosAngelescity           9.32        13.99       10.68       12.26          8.34        2414.77
      Dallascity           7.28        29.50        5.77       15.58          7.43        1744.19
     Detroitcity          16.59        75.67       26.14       17.88         19.67              ?
     Houstoncity           6.91        28.09        7.06       15.07          8.18              ?
    SanDiegocity           4.50         9.39        8.81       12.80          5.72        1162.84
  SanAntoniocity           3.48         7.04        9.58       14.01          8.92         672.57
     Chicagocity          12.08        39.07       14.36       12.99         11.32              ?
     Phoenixcity           4.43         5.19        5.79       14.74          6.64        1097.07
```

Note. `names(CC)` prints out all column names.

5 attributes/variables: percentage of children born to unmarried parents, black population, population receiving public assistance income, divorced, and unemployed, respectively

Response: no. of violent crimes (i.e. murder, rape, robbery, and assault) per 100K population

Detroit has the highest proportions for all the 5 variables

Chicago is the 2nd highest in children with unmarried parents, public assistance in income, unemployment, the 3rd in the other two variables

Violent crime rates for Detroit and Chicago are missing: Controversy concerning the reporting of rapes resulted in missing values for the number of rapes, and subsequently for violent crime rate.

Further investigation: among all cities whose population is at least 500,000 Chicago and Detroit had the second and third highest assault rates respectively, and Detroit had the second highest murder rate, and Chicago had the seventh highest

Not entirely implausible to believe that Chicago and Detroit might have higher violent crime rates than most other communities

These missing values are informative in the sense that the values that are missing may correspond to the communities with higher crime rates.

Therefore, subsequent estimates of overall violent crime rates (e.g. over states or even country-wide) may be biased downwards (underestimated).

`dplyr` can also perform more sophisticated data manipulations and cleaning. For example, to rearrange in the rows according to the ascending order of `population`:

```
> tt=arrange(CC, population)
> tt[2215,1]
[1] NewYorkcity
```

See also

`ran.rstudio.com/web/packages/dplyr/vignettes/introduction.html`

3. **Identifying inconsistencies**

After the data is put in a tidy format, the next step is to ensure that the data makes sense: do the range of values for each variable match what you expect? are there any outliers? how many missing values?

A useful function in R: `summary` – list a summary for each column:

For numerical variables: min, 1st & 3rd quartiles, median, mean, max

For categorical variables: list the frequency at each level

Unmarr. parents	Black	Public assist.	Divorced	Unemp.	Violent crimes
Min 0.000	Min 0.000	Min 0.180	Min 2.830	Min 1.320	Min 0.0
1st-Q 1.070	1st-Q 0.860	1st-Q 3.270	1st-Q 8.575	1st-Q 4.045	1st-Q 161.7
Med 2.040	Med 2.870	Med 5.610	Med 10.900	Med 5.450	Med 374.1
Mean 3.115	Mean 9.335	Mean 6.801	Mean 10.813	Mean 6.045	Mean 589.1
3rd-Q 3.910	3rd-Q 11.145	3rd-Q 9.105	3rd-Q 12.985	3rd-Q 7.440	3rd-Q 794.4
Max 27.350	Max 96.670	Max 44.820	Max 22.230	Max 31.230	Max 4877.1
					NA's 221

All communities with population greater than 800K have much higher violent crime rates than the median 374.1

The distribution for violent crime rates is skewed towards the right as mean >> median; indicating some excessive high crime rates.

4. **Normalization**: convert different quantities to a common scale for the purpose of comparability, numerical stabilization, or outlier detection

A source of confusion: Normalize or not? – the decision will be a judgment call.

? ? ?

- *Standardization*: $x_{new} = (x - \bar{x})/\text{STD}(x)$ Then $\bar{x}_{new} = 0$ and $\text{STD}(x_{new}) = 1$
- *Making data between 0 and 1*: $x_{new} = (x - x_{\text{min}})/(x_{\text{max}} - x_{\text{min}})$

Normalization is usually applied to each variables (columns) not to individuals (rows).

To standardize only if it is necessary: Standardization tends to remove a lot of the natural dependence structures that exist within the data. Further, if the variables had meaningful units such that interpretations after scaling become less transparent, we might be enticed to leave the data alone.

Normalization by transformation: logarithmic or square-root transformations are the most frequently used in practice. Both 'squeeze' data towards the mean, to make data look more normally distributed.

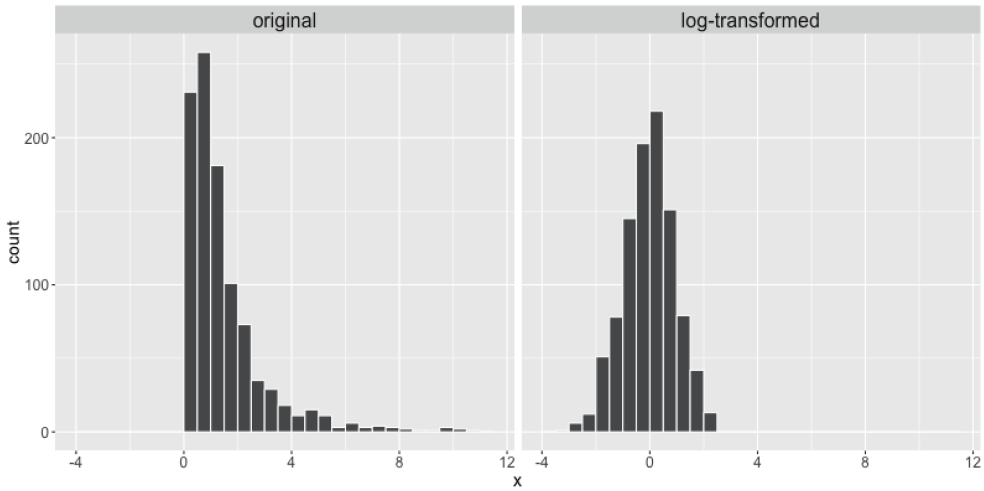
Many statistical models and methods are designed for normal data, or work better for normal data.

Note. (i) Linear regression models do not rely on the normality assumption. But when errors are normally distributed, OLS is MLE and the t-tests are exact instead of being asymptotically approximate

(ii) if $y = x_1 x_2$, $\log y = \log x_1 + \log x_2$. Be mindful on the interpretation of the models for $\log y$!

(iii) Box-Cox transformation

(iv) log-transformation may transform a skewed distribution into a normal-like one



5. **Errors and Outliers**

Most real data are subject to errors.

Errors can be divided into two types: Random and Deterministic.

Deterministic errors can be corrected, if the source of the errors can be correctly identified.

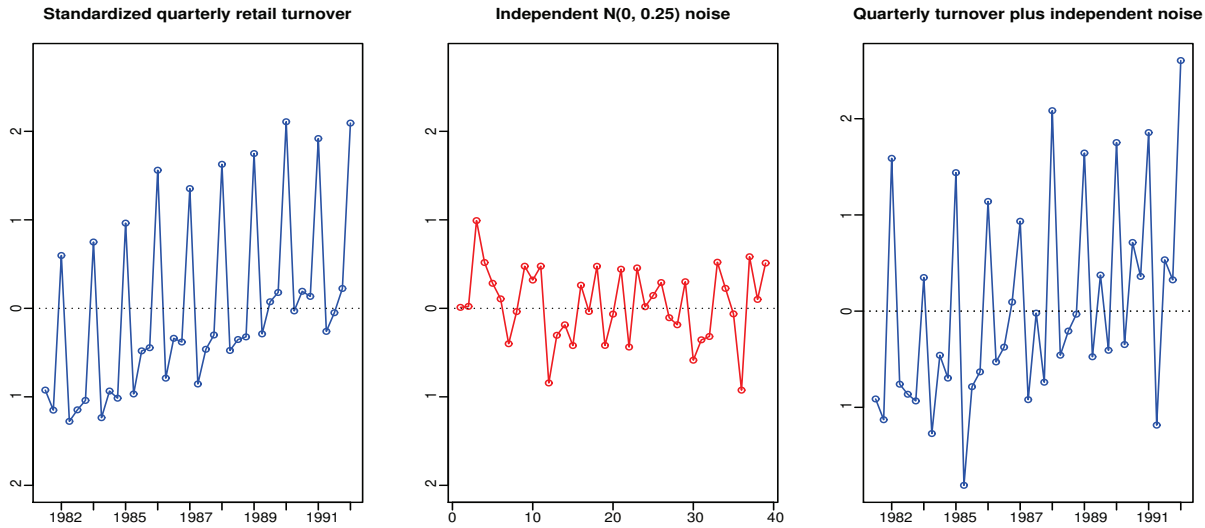
Random errors cannot be corrected from data, though their impact on data mining may be controlled or eliminated sometimes.

Examples of random (stochastic) errors: measurement or transmission errors – typically additive

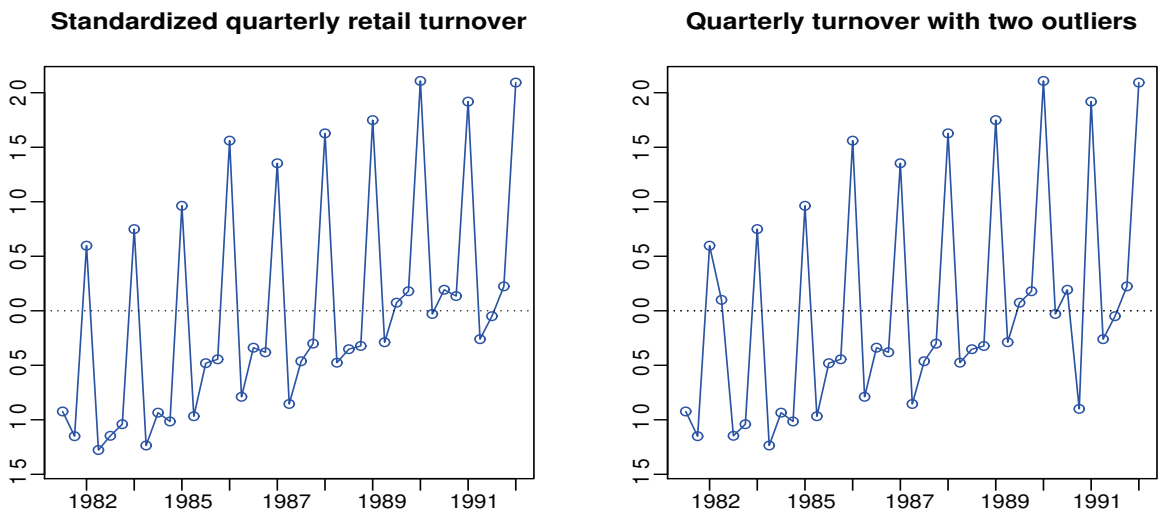
Examples of deterministic errors: wrong formulas for computing data, wrong calibration/scaling, sensor drift, different meanings of '.' or ',' on different systems (i.e. 1.234 and 1,234 may denote the same number)

Standardization: $x \rightarrow (x - \text{mean})/\text{std}$

mean = 14623.17, std=1298.926



Outliers: individual data points showing-off odd behaviour (e.g. too large or too small, or off phase) in comparison with the majority of data, may be either random or deterministic.



Out-range outliers detection

Rule of thumb: If $|x - \text{mean}|/\text{std} > 2$, x is regarded as outlier.

Caution: unusual but correct values should not be removed/modified. Such as financial crisis, earthquake.

Distinguish exotic but valuable data from erroneous data: using domain knowledge!

For the *Communities and Crime Data*, the population (i.e. Column 6 in the dataset) in New York City is 7323000, which is much greater than

$$\text{mean} + 2\,\text{std} = 53120 + 2 * 204620.3 = 462360.6$$

Of course this is not an outlier, even the 2nd largest population is Los Angeles 3485398. The summary of this variable is

```
Min.   1st Qu.  Median   Mean   3rd Qu.    Max.
10000   14370   22790   53120   43020     7323000
```

Error Handling: The identified outliers and missing values can be handled in various ways as follows.

1. *Invalidity list*: the indices of the invalid data are stored in a separate list that is checked in each data processing step.

2. *Invalid values*: replace invalid values directly by `NA` (not available).

3. Remove outliers.

4. *Correction or estimation*: invalid values can be estimated by one of the following methods.
   - replacing by the mean, median, minimum, or maximum of the valid data

- nearest neighbour correction – applicable for correcting one outlier component of vector data: replacing invalid x_k^i by x_j^i, where $\mathbf{x}_k = (x_k^1, \cdots, x_k^p)'$, and

$$\|\mathbf{x}_j - \mathbf{x}_k\|_{-i} = \min_{\ell} \|\mathbf{x}_\ell - \mathbf{x}_k\|_{-i}.$$

- *linear interpolation for time series*:

$$x_t = (x_{t-1} + x_{t+1})/2, \qquad \text{(regularly sampled data)}$$
$$x_k = \frac{x_{k-1}(t_{k+1} - t_k) + x_{k+1}(t_k - t_{k-1})}{t_{k+1} - t_{k-1}}, \qquad \text{(irregularly sampled data)}$$

- *Nonlinear interpolation* using, eg. splines.

- *Filtering*: remove outliers, more often used to remove noise from time series data

- *Model-based estimation* by, eg. regression.

6. **Data Merging**: merge together relevant data from different data sets, files, database or systems.

- identify a clearly defined rule for merging

- identify relevant IT tools to merge data

Data cleaning can be a complex and challenging process; requiring open-minded, critical thinking and relevant IT skills to manipulate complex and large datasets. The goal is first to identify all suspicious data points, and then to treat (i.e. correct, remove or down-weight) them. The former is relatively easy than the latter for which *common sense, subject knowledge, and the knowledge on data collection and recording* are brought to bear.

Data cleaning $\neq$ Oiling data!

1.5 Writing functions in R

For some repeated task, it is convenient to define a function in R. We illustrate this idea by an example.

Consider the famous 'Birthday Coincidences' problem: *In a class of k students, what is the probability that at least two students have the same birthday*?

Let us make some assumptions to simplify the problems:
(i) only 365 days in every year,
(ii) every day is equally likely to be a birthday,
(iii) students' birthdays are independent with each other.

With k people, the total possibilities is $(365)^k$.

Consider the complementary event: all k birthdays are different. The total such possibility is

$$365 \times 364 \times 363 \times \cdots \times (365 - k + 1) = \frac{365!}{(365-k)!}$$

So the probability that at least two students have the same birthday is

$$p(k) = 1 - \frac{365!}{(365-k)!(365)^k}.$$

We may use R to compute $p(k)$. Unfortunately factorials are often too large, e.g. $52! = 8.065525e + 67$, and often cause overflow in computer. We adopt the alternative formula

$$p(k) = 1 - \exp\{\log(365!) - \log((365-k)!) - k\log(365)\}.$$

We define a R-function pBirthday to perform this calculation for different k.

```
> pBirthday <- function(k)
```

```
+ 1 - exp(lfactorial(365) - lfactorial(365-k) - k*log(365))
          # lfactorial(n) returns log(n!)
> pBirthday(100)
[1] 0.9999997   # probability with a class of 100 students
> x <- c(20, 30, 40, 50, 60)
> pBirthday(x)
[1] 0.4114384 0.7063162 0.8912318 0.9703736 0.9941227
```

With 20 students in class, the probability of having overlapping birthdays is about 0.41. But with 60 students, the probability is almost 1, i.e. *it is almost always true that at least 2 out of 60 students have the same birthday.*

Note. The expression in a function may have several lines. In this case the expression is enclosed in curly braces { }, and the final line determines the return value.

Instead of writing R functions directly in a console, RStudio offers the top left window for scripts (or data) editing. To create a new file, use the `File -> New File` menu. To open an existing file, use `File -> Open File`.

RStudio's script editor includes a variety of productivity enhancing features including syntax highlighting, code completion, multiple-file editing, and find/replace. To execute commands in the editing window, highlight them and click on `run`-button.

Another Example — **The capture and recapture problem**

To estimate the number of whitefish in a lake, 50 whitefish are caught, tagged and returned to the lake. Some time later another 50 are caught and only 3 are tagged ones. Find a reasonable estimate for the number of whitefish in the lake.

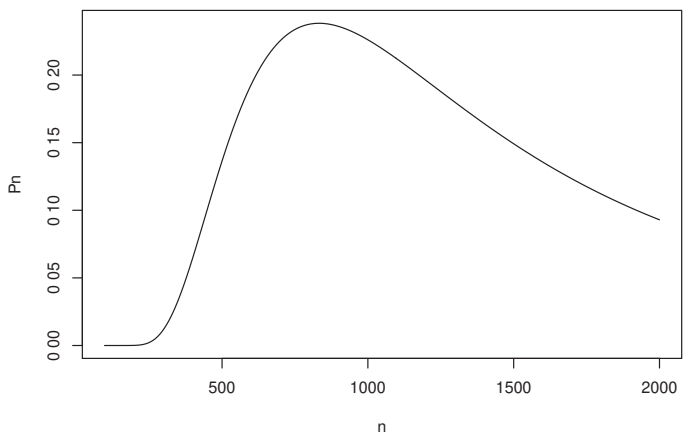
Suppose there are n whitefish in the lake. Catching 50 fish can be done in $\begin{pmatrix} n \\ 50 \end{pmatrix} = \frac{n!}{50!(n-50)!}$ ways, while catching 3 tagged ones and 47 untagged can be done in $\begin{pmatrix} 50 \\ 3 \end{pmatrix}\begin{pmatrix} n-50 \\ 47 \end{pmatrix}$ ways. Therefore the probability for the latter event to occur is

$$P_n = \begin{pmatrix} 50 \\ 3 \end{pmatrix}\begin{pmatrix} n-50 \\ 47 \end{pmatrix} \Big/ \begin{pmatrix} n \\ 50 \end{pmatrix}.$$

Therefore, a reasonable estimate for n should be the value at which P_n obtains its maximum. We use R to compute P_n and to find the estimate.

```
> Pn <- function(n) {
+      tmp <- choose(50,3)*choose(n-50,47)
+      tmp/choose(n,50)
+ }           # Definition for function Pn ends here
> n <- 97:2000    # as there are at least 97 fish in the lake
> plot(n, Pn(n), type='l')
```

It produces the plot of P_n against n:



To find the maximum:

```
> m <- max(Pn(n)); m
[1] 0.2382917
> n[Pn(n)==m]
[1] 833
```

Hence the estimated number of fish in the lake is 833.

1.6 Control structure: loops and conditionals

An `if` statement has the form

```
if (condition) expression1 else expression2
```

It executes 'expression1' if 'condition' is true, and 'expression2' otherwise.
When 'condition' contains several lines, they should be enclosed in curly braces { }. The same applies to expressions.

The above statement can be compactly written in the form

```
ifelse(condition, expression1, expression2)
```

When the else-part is not present:

```
if (condition) expression
```

It executes 'expression' if 'condition' is true, and does nothing otherwise.

A `for` loop allows a statement to be iterated as a variables assumes values in a specified sequence. It has the form:

```
for(variable in sequence) statement
```

A `while` loop does not use an explicit loop variable:

```
while (condition) expression
```

It repeats 'expression' as long as 'condition' holds. This makes it differently from the "if-statement" above.

We illustrate those control commands by examining a simple 'doubling' strategy in gambling.

You go to a casino to play a simple 0-1 game: you bet x dollars and flip a coin. You win $2x$ dollars and keep your bet if 'Head', and lose x dollars if 'Tail'. You start 1 dollar in first game, and double your bet in each new games, i.e. you bet 2^{i-1} dollars in the i-th game, $i = 1, 2, \cdots$.

With this strategy, once you win, say, at the $(k+1)$-th game, you will recover all your losses in your previous games plus a profit of $2^k + 1$ dollars, as

$$2 \times 2^k > \sum_{i=1}^{k} 2^{i-1} = 2^k - 1.$$

Hence as long as (i) the probability p of the occurrence of 'Head' is positive (no matter how small), and (ii) you have enough capital to keep you in the games, you may win handsomely at the end — is it really true?

Condition (ii) is not trivial, as the maximum loss in 20 games is $2^{20} - 1 = 1,048,575$ dollars!

Plan A: Suppose you could afford to lose maximum n games and, therefore, decide to play n games. We define the R-function `nGames` below to simulate your final earning/loss (after n games).

```
nGames <- function(n,p) {
      # n is the No. of games to play
      # p is the prob of winning each game
x <- 0 # earning after each game
for(i in 1:n)  ifelse(runif(1)<p, x <- x+2^i, x <- x-2^(i-1))
      # runif(1) returns a random number from uniform dist on (0, 1)
x     # print out your final earning/loss
}
```

To play $n = 20$ games with $p = 0.1$:

```
> nGames(20, 0.1)
[1] -999411
> nGames(20, 0.1)
[1] -1048575
> nGames(20, 0.1)
[1] 524289
```

```
> nGames(20, 0.1)
[1] -655263
> nGames(20, 0.1)
[1] -1016895
```

We repeated the experience 5 times above, with 5 different results.

One way to assess this gameplan is to repeat a large number of times and look at the average earning/loss:

```
> x = vector(length=5000)
> for(i in 1:5000) x[i] <- nGames(20, 0.1)
> mean(x)
[1] -733915
```

In fact, this mean -733915 is stable measure reflecting the average loss of this gameplan.

Plan B: Play the maximum n, but quit as soon as winning one game. The R-function `winStop` simulates the earning/loss.

```
winStop <- function(n,p) {
      # n -- maximum No. of games, p -- prob of winning each game
i <- 1
ifelse((runif(1)<p), x<- 2, x<- -1)  # play 1st game
while((x<0)&(i<n)){ i <- i+1      # i records the no. of games played
                  ifelse(runif(1)<p, x <- x+2^i, x <- x-2^(i-1))
                }
x
}
```

Set $n = 20$, $p = 0.1$, we repeat the experience a few times:

```
>winStop(20, 0.1)
[1] 2
```

```
> winStop(20, 0.1)
[1] 17
> winStop(20, 0.1)
[1] 129
> winStop(20, 0.1)
[1] -1048575
> winStop(20, 0.1)
[1] 16385
```

To assess the gameplan:

```
> x<- 1:5000
> for(i in 1:5000) x[i] <- winStop(20, 0.1)
> mean(x)
[1] -112672.9   # This indicates "Plan B" is better than "Plan A"
> for(i in 1:5000) x[i] <- winStop(80, 0.1)
          # the maximum no. of games is 80 now
```

```
> mean(x)
[1] -7.22886e+20
> for(i in 1:5000) x[i] <- winStop(90, 0.1)
          # the maximum no. of games is 90 now
> mean(x)
3.790896e+18
```

With p as small as 0.1, you need a huge capital in order to play about 90 games to generate the positive returns in average.

The best and the most effective way to learn R: use it!

Hands-on experience is the most illuminating.

R Markdown: A powerful authoring framwork for

- creating, executing and saving R codes, and
- generating a quality report, in HTML, pdf or word, containing codes, R-outputs (graphics) and text. It could be a normal document, or a set of slides for presentation

To install, `install.packages("rmarkdown")`

R Markdown Cheatsheet:
`www.rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf`

R Markdown Reference Guide:
`www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf`

An R Markdown file is a plain text file with the extension `.Rmd`. It contains three types of content:

- An optional header sandwiched by two sets of ---
- R code chunks sandwiched by ```{r } and ```
- text chunks with simple text formatting (see R Markdown Reference Guide)

To start a session with R Markdown, on the top left panel of RStudio, click on

`File` → `New File` → `R Markdown ...`

This creates a template markdown file which one can edit. For example, the file below illustrates some elementary and useful functions.

````
---
title: "R Markdown Basic"
author: "Qiwei Yao"
date: "May 31, 2019"
output: html_document
---

## On this document
This is an R Markdown document illustrating its rudimentary function
```{r }
setwd("~/teaching/bigData/data")
jobs <- read.table("Jobs.txt", header=T, row.names=1)
t <-table(jobs[,2], jobs[,1], deparse.level=2)  # store table in t
t
100*t[1,]/sum(t[1,])
100*t[2,]/sum(t[2,])
```

### Include a plot
```{r }
barplot(t, main="No. of graduates in 5 different job categories",
    legend.text=c('male', 'female'), names.arg=c('accounting',
    'finance', 'management', 'marketing', 'others'),
    col=c("blue", "red"))
```

R Markdown can include texts and formulas such as $x^2 + y^2 = z^2$,
or a display

$$y^2_{t+1}= y_t^2 + 2\cdot z^{1/3} - \sqrt{r_2}.$$
````

Then remember to save the file first with the last name `.Rmd`, then click on `Knit`-button to produce the html-document below.

arkdown Basic

Qiwei Yao

May 31, 2019

On this document

This is an R Markdown document illustrating its rudimentary function

```
setwd("~/teaching/bigData/data")
jobs <- read.table("Jobs.txt", header=T, row.names=1)
t <-table(jobs[,2], jobs[,1], deparse.level=2)  # store table in t
t
```

```
##          jobs[, 1]
## jobs[, 2]  1  2  3  4  5
##         1 35 30 18 39 13
##         2 38 22 18 25 15
```

```
100*t[1,]/sum(t[1,])
```

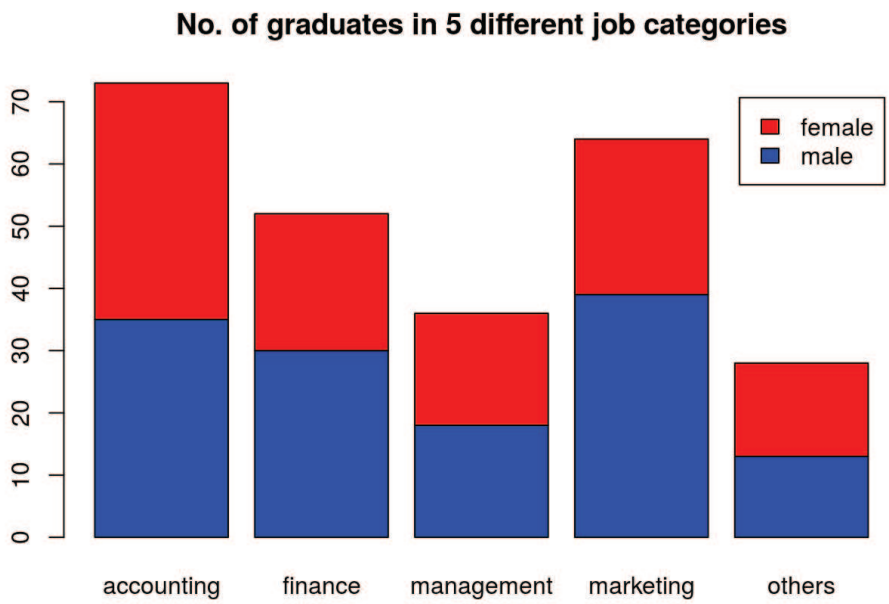
```
##        1        2        3        4        5
## 25.92593 22.22222 13.33333 28.88889  9.62963
```

```
100*t[2,]/sum(t[2,])
```

```
##        1        2        3        4        5
## 32.20339 18.64407 15.25424 21.18644 12.71186
```

Include a plot

```
barplot(t, main="No. of graduates in 5 different job categories", legend.text=c('mal
e', 'female'), names.arg=c('accounting', 'finance', 'management', 'marketing', 'other
s'), col=c("blue", "red"))
```

R Markdown can include texts and formulas such as $x^2 + y^2 = z^2$, or a display

$$y_{t+1}^2 = y_t^2 + 2 \cdot z^{1/3} - \sqrt{r_2}.$$

Chapter 3. Classification

- Measuring uncertainties and information gains
- Decision trees
- Logistic regression
- Classification in R: `tree, glm`

Further readings:

James et al. (2013) Sections 4.1-4.3 & 8.1,

Provost and Fawcett (2013) Chapter 3.

Classification problems occur often, perhaps even more so than regression problems. Examples include:

- An online banking service must be able to determine whether or not a transaction being performed is fraudulent, on the basis of the user's transaction history, IP address, $\cdots$
- Customer Churn: decide if a customer should be offered a special retention deal prior to the expiration of his/her contract
- A patient arrives at a hospital with a set of symptoms which could possibly be attributed to one of three medical conditions. Which of the three conditions does the individual have?
- Spam detection: is a newly arrived email a spam or not?
- Advertising: who among the customers database would be interested a new model of BMW?

* classification

Classification is typically a supervised learning problem: use historical data (i.e. training data) to identify the characteristics of different classes, and predict the class of new data based on the identified characteristics.

Key: identify the relevant characteristic variables which carry the information on the classes.

Information is a quantity that reduces uncertainty

Variable selection: choose characteristics/variables which carry most information, or, equivalently, minimize uncertainty.

* more information = less uncertainty

Tree methods

Particularly useful for analyzing large data sets or the data sets with the number of input variables, denoted by p, greater than the number of individuals (i.e. the number of observations), denoted by n.

Training data: data with both input variables, denoted by $\mathbf{X}$, and outcome (i.e. label of classes) known, denoted by Y.

Basic idea: partition the feature space (i.e. $\mathbf{X}$-space) into a set of rectangles, then assign each rectangle to a particular class, (or fit a simple model, such as a constant, in each rectangle for a regression tree)

Conceptually simple yet powerful, and computationally demanding when the feature variable $\mathbf{X}$ is high dimensional

When p is large, the partition is done by a subset of components of $\mathbf{X}$.

$$I(A) = \begin{cases} 1 & A \text{ occurs} \\ 0 & \text{otherwise} \end{cases}$$

An illustration by example: $Y = f(X_1, X_2) + \varepsilon$.

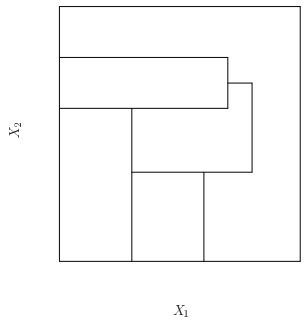
Fitting via *recursive binary splitting*:

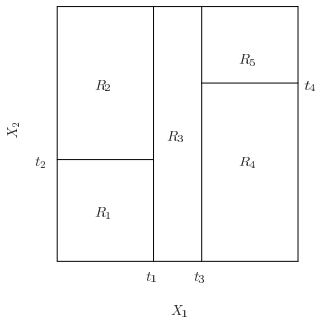
$$\hat{f}(X_1, X_2) = \sum_{m=1}^{5} c_m I\{(X_1, X_2) \in R_m\}$$

where the partition $\{R_1, \cdots, R_5\}$ is obtained by several simple splits along either X_1 or X_2 as illustrated in the figure below.

no overlap

The unit of five is whole Space



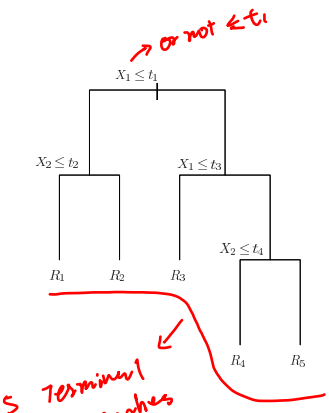


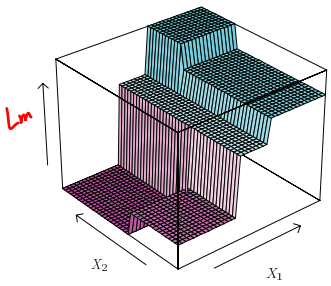
Top Right: Output of recursive binary splitting

Bottom Left: Tree corresponding to the partition

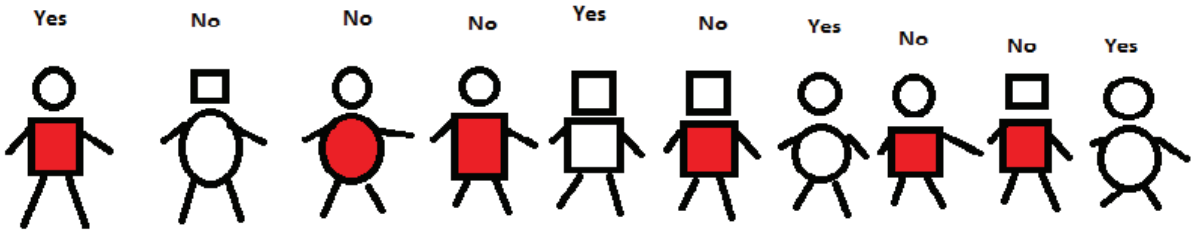
Bottom Right: Perspective plot of the predictive surface corresponding to the tree

Top Left: A partition cannot be resulted from recursive binary splitting.





Churn or not?



$$p_1 \equiv P(\text{Churn}) = 0.4, \qquad p_0 \equiv P(\text{non-Churn}) = 0.6.$$

Entropy — a measure for uncertainty: $-\sum_i p_i \log p_i$

For this example, Entropy$= -0.4\log(0.4) - 0.6\log(0.6) = 0.6730$

Note. Entropy is 0, when one of p_i is 1 — No uncertainty!

Let $Y = I(\text{churn})$. Then $Y = 1$ indicates a churn, and $Y = 0$ indicates non-churn, and the entropy of Y is

$$H(Y) = -\sum_i P(Y = i) \log P(Y = i) = 0.6730.$$

Now we try to reduce the uncertainly of Y by predicting Y based on the three characteristics:

- $X_1 = 1$ (round head), or 0 (square head)
- $X_2 = 1$ (solid body), or 0 (light body)
- $X_3 = 1$ (round body), or 0 (square body)

Always try to max that.

Rule to grow tree: choose X_i to maximize information gain (IG)

The conditional entropy of Y given X is $H(Y|X)$ defined as (weighted Average.)

$$H(Y|X) = \sum_k P(X = k) h(Y|X = k)$$

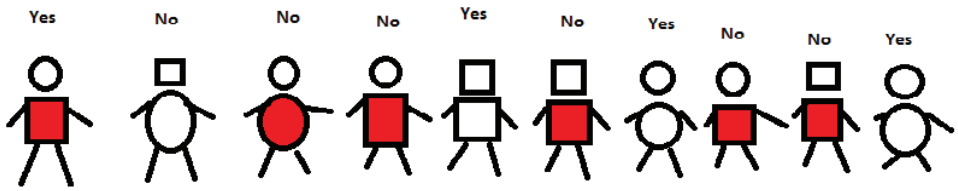
where $h(Y|X = k) = -\sum_i P(Y = i|X = k) \log\{P(Y = i|X = k)\}$.

$$IG(Y|X) = H(Y) - H(Y|X)$$

$= -0.5 \log 0.5 - 0.5 \log 0.5 = 0.6931$

$P(Y|X=1) = -P(1|X_1=1) \log(1|X_1=1) - P(0|X_1=1) \log P(0|X_1=1)$

Churn or not?



Let $p(j|X_i = k) = P(Y = j|X_i = k)$

i	$P(X_i = 1)$	$P(1\|X_i = 1)$	$P(0\|X_i = 1)$	$P(X_i = 0)$	$P(1\|X_i = 0)$	$P(0\|X_i = 0)$
1	0.6	0.5	0.5	0.4	0.25	0.75
2	0.6	1/6	5/6	0.4	0.75	0.25
3	0.4	0.5	0.5	0.6	1/3	2/3

$$h(Y|X_1 = 1) = .6931, \quad h(Y|X_1 = 0) = .5623, \quad H(Y|X_1) = .6408$$
$$h(Y|X_2 = 1) = .4506, \quad h(Y|X_2 = 0) = .5623, \quad H(Y|X_2) = .4953$$
$$h(Y|X_3 = 1) = .6931, \quad h(Y|X_3 = 0) = .6365, \quad H(Y|X_3) = .6591$$

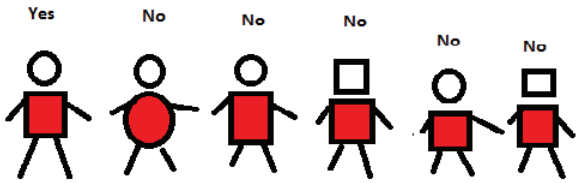
Thus, $IG(Y|X_2)$ is the biggest: 1st branch is generated by X_2.

Note. $H(Y) \geq H(Y|X)$ for any Y and X.

$H(Y|X_1) = P(X_1=1) h(Y|X_1=1) + P(X_1=0) h(Y|X_1=0)$

$= 0.6 \times 0.6931 + 0.4 \times 0.5623 = 0.6408$

$X_2 = 1$ (solid body)



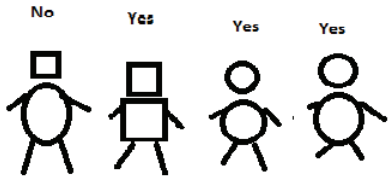
$H(Y) = -(\frac{1}{6}\log\frac{1}{6} + \frac{5}{6}\log\frac{5}{6}) = 0.4505$

$H(Y|X_1) = \frac{4}{6}h(Y|X_1 = 1) + \frac{2}{6}h(Y|X_1 = 0)$
$= \frac{4}{6} \times 0.5623 = 0.3748$

$H(Y|X_3) = \frac{1}{6}h(Y|X_3 = 1) + \frac{5}{6}h(Y|X_3 = 0)$
$= \frac{5}{6} \times 0.5004 = 0.4170$

A further branch grows out of X_1.

$X_2 = 0$ (light body)



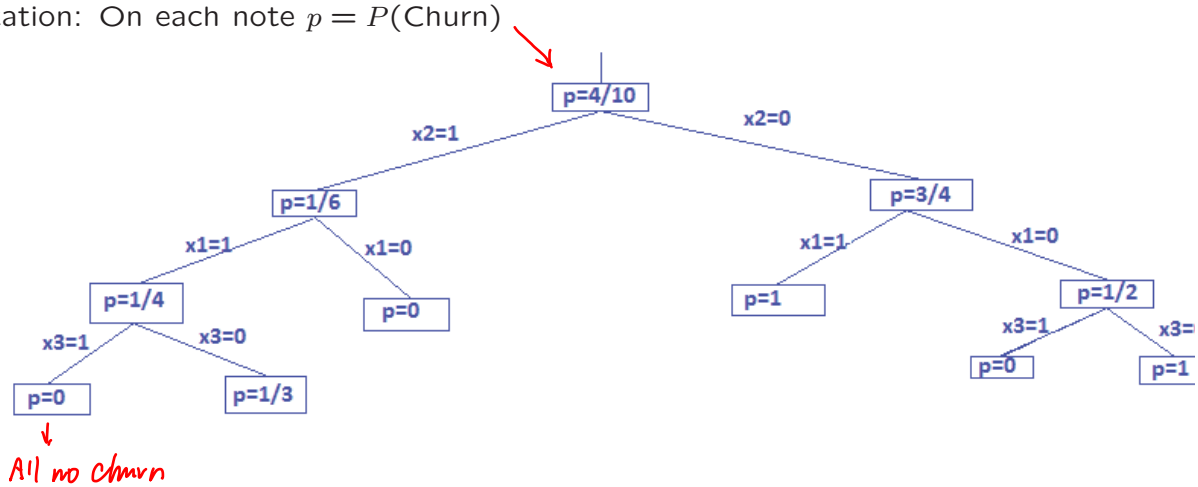
$H(Y) = -(\frac{1}{4}\log\frac{1}{4} + \frac{3}{4}\log\frac{3}{4}) = 0.5623$

$H(Y|X_1) = \frac{2}{4}h(Y|X_1 = 1) + \frac{2}{4}h(Y|X_1 = 0)$
$\frac{2}{4} \times 0.6931 = 0.3466$

$H(Y|X_3) = \frac{3}{4}h(Y|X_3 = 1) + \frac{1}{4}h(Y|X_3 = 0)$
$\frac{3}{4} \times 0.6365 = 0.4774$

A further branch grows out of X_1.

Notation: On each note $p = P(\text{Churn})$



- Computing intensity: more so with more X's or/and X's take multiple/continuous values.

- Stepwise searching: exhausting searching only at each step

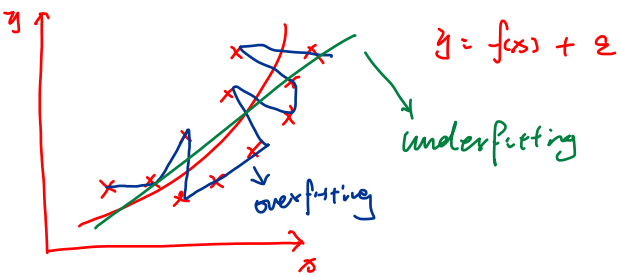
- Uncertainty may still exist at some terminal nodes (e.g. along branches $X_2 = 1, X_1 = 1, X_3 = 0$). We then classify the node into the class according to the majority of the individuals in this node (i.e. non-Churn, as $p = 1/3$).

- Overfitting – should be avoided!

  Common wisdom: Data consist of signal (i.e. relevant information) plus noise. A good modelling should extract relevant information only.

  There are sound statistical procedures for preventing or detecting overfitting.

  A good data-mining/statistical modelling should always make judicious use of subject knowledge, creativity and common sense.



In reality, variables affecting mobile-customer churns include:

- *College*: is the customer college-educated?
- *Income*: annual income
- *Overage*: average overcharges per month
- *Leftover*: average number of leftover minutes per month
- *House*: estimated value of house
- *Handset price*: cost of phone
- *Average call duration*: average duration of calls
- *Long call per month*: average number of long calls ($\geq$ 15mins) per month
- *Reported satisfaction*: reported level of satisfaction
- *Reported usage level*: self-reported usage level

For a historical data set of 20,000 customers who either has stayed the company or has churned, we fit a tree model for predicting future churns.

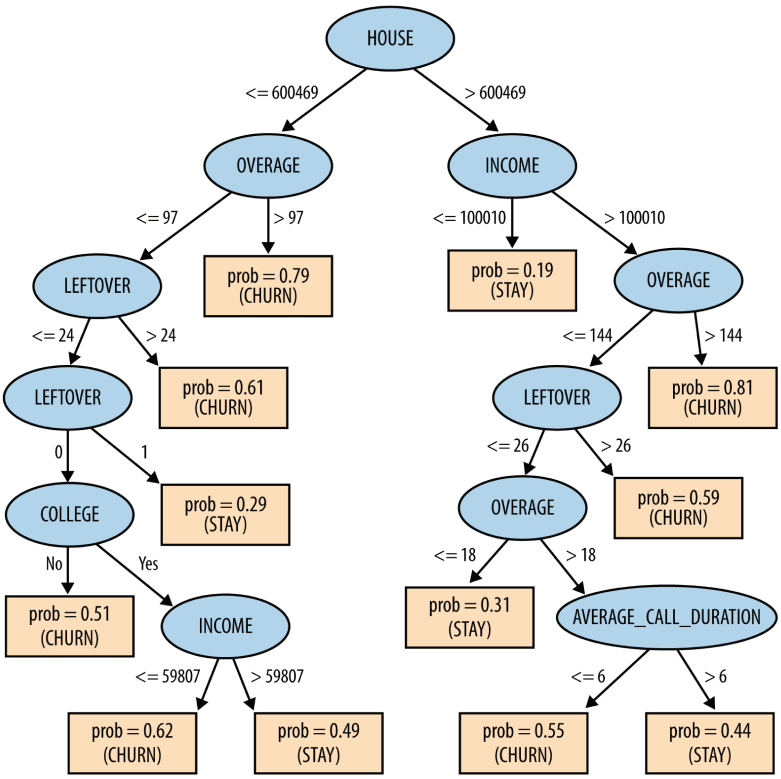
Information gains for each individual variables are: *House* .0461, *Overage* 0.0436, *Long call per month* .0356, *leftover* .0101 and etc

In the tree, terminal nodes are in square with the churn-probability and the classified class printed in square.

The tree achieved 73% accuracy of its decision for the training data.

1. Do we trust this number in the sense that the tree will produce 73% accuracy for a different data set?

2. If we do trust this number, is the model with 73% accuracy worth using?



A general strategy: grow up a big tree according to, e.g. the entropy criterion, then prune the tree by removing some internal nodes according to some criteria.

It is not a good idea to use a threshold to control the growth of the initial big tree, as an uninteresting branch may lead to some interesting branches later on.

A pruning criterion: for a given penalty constant $\lambda > 0$, search for the tree which minimizes

$$(\text{misclassification rate}) + \lambda \times (\text{number of terminal nodes})$$

Note. The 1st term measures the goodness of fit to the training data, and the 2nd term penalizes the complexity of model (i.e. the size of tree).

↓

make Trade-off between them

The role of the tuning parameter λ:

- $\lambda = 0$ leads to a big tree with the minimum misclassification on training data. But the resulting model typically performs badly in predicting non-training data

- too large λ leads to a tree with few branches, which underfits the data

For computational efficiency, the pruning is often carried out in stepwise fashion: at each step collapse one internal node which leads to the minimum increase in the misclassification rate.

Choose λ: 5-fold or 10-fold cross-validation.

↳ leave 1 of do fitting using 4 left.

Predicting Email Spam

Binary indicator Y: 1 – spam, 0 – (genuine) email

57 predictive variables:

- 48 quantitative variables — percentages of the given 48 words including *business, address, internet, free, george*
- 6 quantitative variables — percentages of the 6 characters ; ([! $ # (among all characters)
- The average length of uninterrupted sequences of capital letters: CAPAVE
- The length of the longest uninterrupted sequence of capital letters: CAPMAN
- The sum of the length of uninterrupted sequences of capital letters: CAPTOT

Use 3065 data points (i.e. emails) for estimation, and randomly selected 1536 points for testing.

☆ Other as fitting data point

The entropy measure was used to grow the tree, then the misspecification rate was used to prune the tree.

Figure 9.4 shows that the misclassification rate flattens out at around 17 terminal nodes, giving the pruned tree in Figure 9.5.

		Predicted Class	
		email	spam
True Class	email	57.3%	4.0%
	spam	5.3%	33.4%

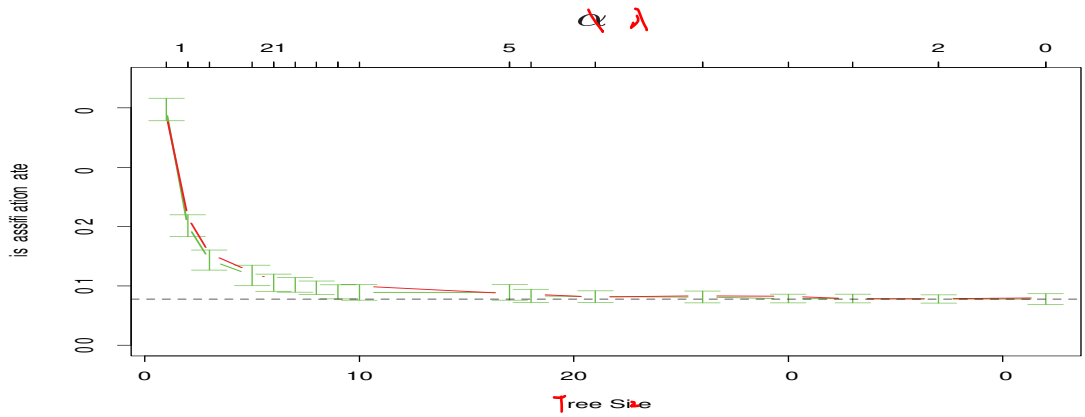
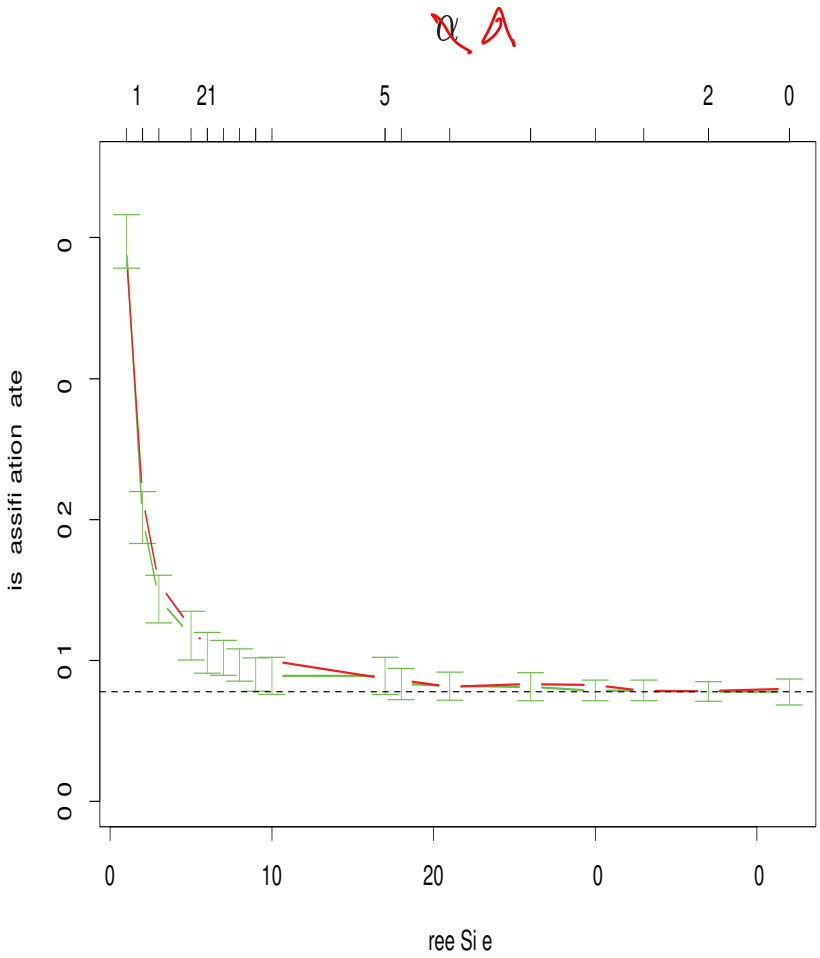


Figure 9.4: *Results for* **spam** *example. The green curve is the tenfold cross-validation estimate of misclassification rate as a function of tree size, with ± two standard error bars. The minimum occurs at a tree size with about 17 terminal nodes. The red curve is the test error, which tracks the CV error quite closely. The cross-validation was indexed by values of α, shown above. The tree sizes shown below refer to $|T_\alpha|$, the size of the original tree indexed by α.*



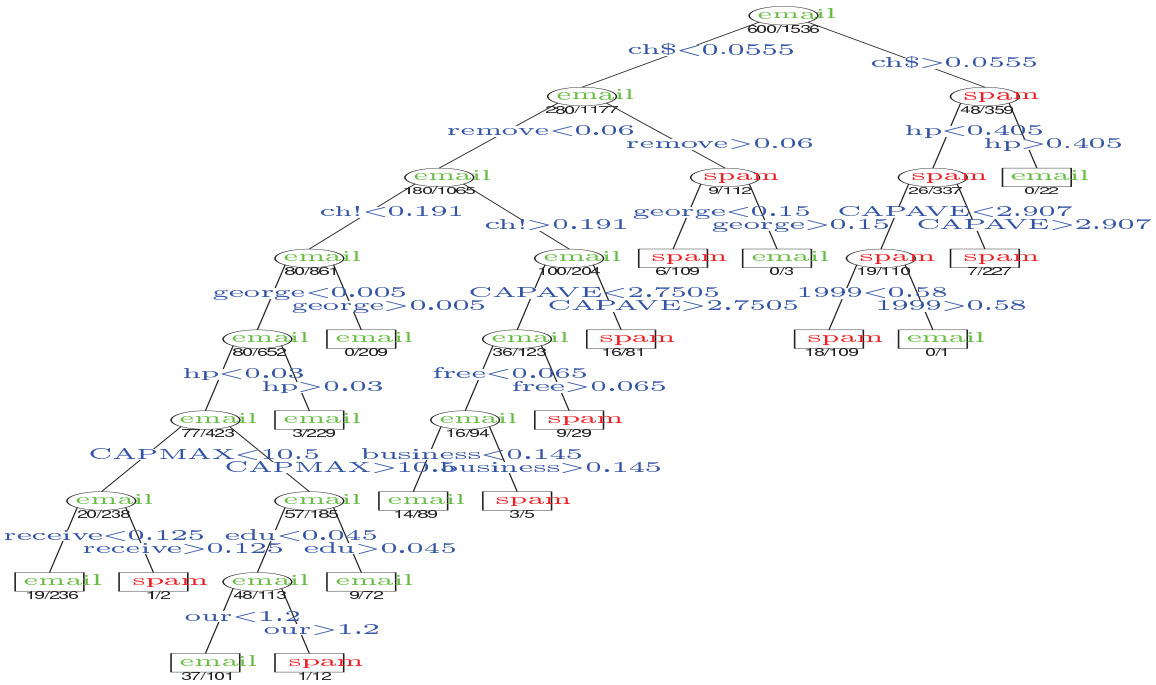


Figure 9.5: *The pruned tree for the* **spam** *example. The split variables are shown in blue on the branches, and the classification is shown in every node. The numbers under the terminal nodes indicate misclassification rates on the test data.*

On the rightmost branches of the tree, we have a spam warning if more than 5.5% of the characters are '$'. However if in addition the phrase hp occurs frequently, it is likely to be company business and we classify as email. All of the 22 cases in the test set satisfying these criteria were correctly classified. If the second condition is not met, and in addition the average length of capital letters CAPAVE is larger than 2.9, we classify as spam. Of the 227 test cases, only 7 were misspecified.

Note. A classifier may take the same value on the two sub-regions from one splitting.

Advantage: easy to explain and interpret graphically, easy to handle qualitative predictors

Disadvantage: less accurate than other classification methods.

However improvements are possible by aggregating many decision trees using *bagging*, *random forests* and *boosting*.

Fitting a decision tree with package `tree`

We use a data set `Carseats` from the package `ISLR` for the illustration.

```
> install.packages("tree"); install.packages("ISLR")
> library(tree); library(ISLR)
> Carseats
    Sales CompPrice Income Advertising Population Price ShelveLoc Age Education Urban  US
1    9.50       138     73          11        276   120       Bad  42        17   Yes Yes
2   11.22       111     48          16        260    83      Good  65        10   Yes Yes
3   10.06       113     35          10        269    80    Medium  59        12   Yes Yes
4    7.40       117    100           4        466    97    Medium  55        14   Yes Yes
... ...
```

The data set contains info on car seat sales in 400 stores with 11 variables:

Sales: Unit sales (in thousands) at each location

CompPrice: Price charged by competitor at each location

Income: Community income level (in thousands of dollars)

Advertising: Local advertising budget for company at each location (in thousands of dollars)

Population: Population size in region (in thousands)

Price: Price company charges for car seats at each site

ShelveLoc: A factor with levels Bad, Good and Medium indicating the quality of the shelving location for the car seats at each site

Age: Average age of the local population

Education: Education level at each location

Urban: A factor with levels No and Yes to indicate whether the store is in an urban or rural location

US: A factor with levels No and Yes to indicate whether the store is in the US or not

```
> attach(Carseats)
> High=ifelse(Sales<=8, "No", "Yes")    # Define the label High iff Sales >8
> Carseats2=data.frame(Carseats, High)   # combine the label into the data set
> tree.carseats=tree(High~.-Sales, Carseats2) # . indicates using all the predictors
                                              # -Sales: exclude Sales
> summary(tree.carseats)

Classification tree:
tree(formula = High ~ . - Sales, data = Carseats2)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Income"      "CompPrice"   "Population"
[6] "Advertising" "Age" "US"
Number of terminal nodes:  27
Residual mean deviance:  0.4575 = 170.7 / 373
Misclassification error rate: 0.09 = 36 / 400
```

The error rate for the training data is 9%.

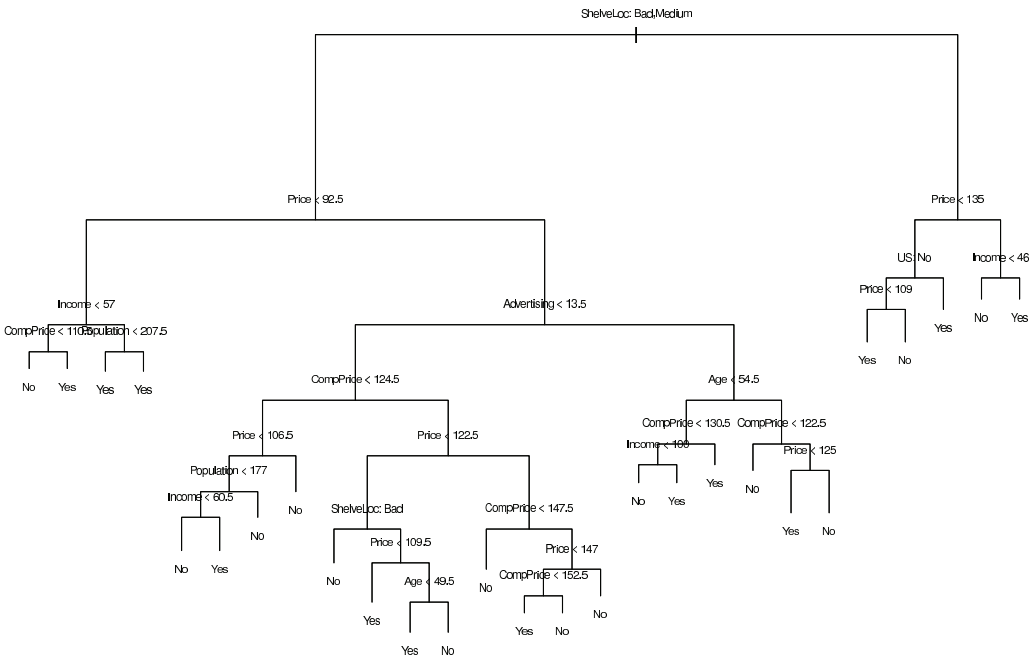
The deviance is defined as

$$-2\sum_m \sum_k n_{mk} \log \widehat{p}_{mk}$$

where n_{mk} is the number of observations in the m-th terminal node that belong to the k-th class, $\widehat{p}_{mk} = n_{mk}/n_m$, and n_m is the total number of observations in the m-th terminal node. The smaller the deviance is, the better fit for the (training) data.

```
> plot(tree.carseats)
> text(tree.carseats, pretty=0, cex=0.6)
```

The most important indicator is shelving location!



To assess the performance of the fitted tree to new data, we split the observations into two parts: training set and testing set.

```
> train=sample(1:nrow(Carseats2), 200) # randomly select 200 numbers between 1
                                       # and nrow(Carseats2)
> testData=Carseats2[-train,]     # test data for checking performance
> High.test=High[-train]
> tree2=tree(High~.-Sales, Carseats2, subset=train)
> tree.pred=predict(tree2, testData, type="class") # type="vector" returns
                           # predictive probabilities, check ?predict.tree
> table(tree.pred, High.test)
         High.test
tree.pred No Yes
      No  81  23
      Yes 35  61
> (23+35)/(23+35+81+61)
[1] 0.29   # Misclassfication rate for the testing data
```

We expect that the accuracy for classifying new data from this fitted tree would be about 1-0.29=71%.

→ resampling data

$x_1, \dots x_n$ i.i.d (μ, σ^2)

$\bar{x} = \frac{1}{n}(x_1 + \dots + x_n) \sim (\mu, \frac{\sigma^2}{n})$

Bagging: a bootstrap aggregation method

聚集

Decision tree suffers from *high variance*, i.e. the tree depends on training data sensitively.

Basic idea to reduce variance: average a set of observations (such as a sample mean)

Create B sets of training data by bootstrapping from the original data set. For each bootstrap sample, create a decision tree.

Those trees are grown deep, and are not pruned. Hence each individual tree has high variance but low bias.

'Averaging' those B trees reduces the variance.

For decision trees, 'averaging' means taking the majority votes of the B trees.

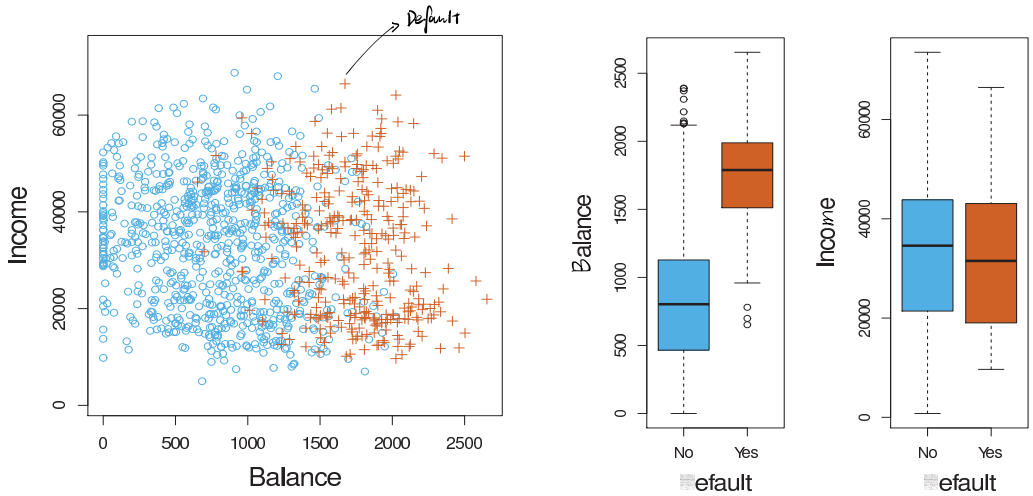
→ only use subset of variable

Random forests: similar to Bagging, but at each split only choose from randomly selected $\sqrt{p}$ input variables

Boosting: create a strong learner from a set of weak learners.

☆ a very frequently use procedure nowadays

Logistic Regression: model the probability that Y belongs to a particular class.



Goal: predict if an individual will default on his/her credit card payment, using the credit card *balance* and the annual *income*.

Balance is a more effective predictor!

Let

$$P(\text{ default } = \text{ Yes } | \text{ balance }) \equiv P(Y = 1|X) \equiv p(X).$$

Then $p(X) \in [0, 1]$, it varies wrt *balance*.

Logistic Regression: Bound to be a number

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

Consequently,

$$1 - p(X) = P(Y = 0|X) = \frac{1}{1 + e^{\beta_0+\beta_1 X}},$$

and log-odds or logit is

$$\log \frac{p(X)}{1 - p(X)} = \beta_0 + \beta_1 X. \quad (\text{A linear function})$$

Estimation for logistic regression: Suppose we have training data $(Y_i, X_i), i = 1, \cdots n$.

They are not same !

Likelihood function: probability function

$$L(\beta_0, \beta_1) = \prod_{i:Y_i=1} p(X_i) \prod_{j:Y_j=0} (1 - p(X_j))$$

Log-likelihood function: easy to work; info. about; unknown parameters

$$l(\beta_0, \beta_1) = \sum_{i:Y_i=1} \log p(X_i) + \sum_{j:Y_j=0} \log(1 - p(X_j)).$$

Additivity of info.; plotted by observations

The maximum likelihood estimators (MLE):

$$(\hat{\beta}_0, \hat{\beta}_1) = \arg\max_{\beta_0,\beta_1} l(\beta_0, \beta_1) = \arg\max_{\beta_0,\beta_1} L(\beta_0, \beta_1).$$

	Coefficient	Std error	Z-statistic	P-value
Intercept β_0	-10.6513	0.3612	-29.5	<0.0001
balance β_1	0.0055	0.0002	24.9	<0.0001

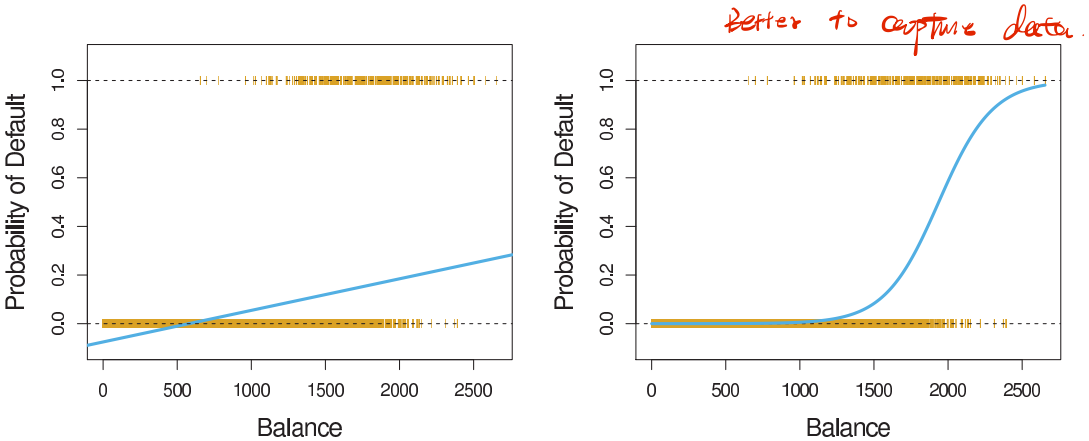
Risk to reject H_0. $H_0: \beta_1 = 0$

A unit increase in balance leads to an increase of 0.0055 in the log odds of default

Prediction: For an individual with balance \$1000, the predicted default probability is

$$\widehat{p}(1000) = \frac{e^{-10.6513+0.0055\times 1000}}{1+e^{-10.6513+0.0055\times 1000}} = 0.00576,$$

i.e. the probability that this individual will default is less than 1%. However, for individual with balance \$2000, the predicted default probability is 58.6%.



Right panel: plot of the estimated $p(X)$ against X. Orange dots are the training data (X_i, Y_i) used in estimation.

Left panel: direct linear regression estimation $Y = \widehat{\beta}_0 + \widehat{\beta}_1 X$

Logistic regression with a discrete predictor

Predict credit card default using student status indicator: $X = 1$ – students, $X = 0$ – non-student:

	Coefficient	Std error	Z-statistic	P-value
Intercept	-3.5041	0.0707	-49.55	$<$0.0001
student	0.4049	0.1150	3.52	0.0004

This leads to the predictive probabilities:

$$P(\text{default}|\text{student}) = \frac{e^{-3.5041+0.4049\times 1}}{1+e^{-3.5041+0.4049\times 1}} = 0.0431,$$

$$P(\text{default}|\text{non-student}) = \frac{e^{-3.5041}}{1+e^{-3.5041}} = 0.0292.$$

☆ Model too simple, Balance matters as well

Thus students tend to have higher default probabilities than non-students — useful info for credit card companies.

Do they need to know more?

Multiple logistic regression: Let

$$p(X_1,\cdots,X_p) \equiv P(Y=1|X_1,\cdots,X_p) = 1 - P(Y=0|X_1,\cdots X_p)$$

Then the model is of the form

$$p(\mathbf{X}) = \frac{e^{\beta_0+\beta_1X_1+\cdots+\beta_pX_p}}{1+e^{\beta_0+\beta_1X_1+\cdots+\beta_pX_p}},$$

or equivalently, the log odds is linear in $X_1,\cdots,X_p$

$$\log\frac{p(\mathbf{X})}{1-p(\mathbf{X})} = \beta_0+\beta_1X_1+\cdots+\beta_pX_p,$$

Fitting a logistic model for predicting default using *balance*, *income* and *student status* (i.e. 1 for student, and 0 for non-student):

	Coefficient	Std error	Z-statistic	P-value
Intercept	-10.8690	0.4932	-22.08	<0.0001
balance	0.0057	0.0002	24.74	<0.0001
income	0.0030	0.0082	0.37	0.7115
student	-0.6468	0.2362	-2.74	0.0062

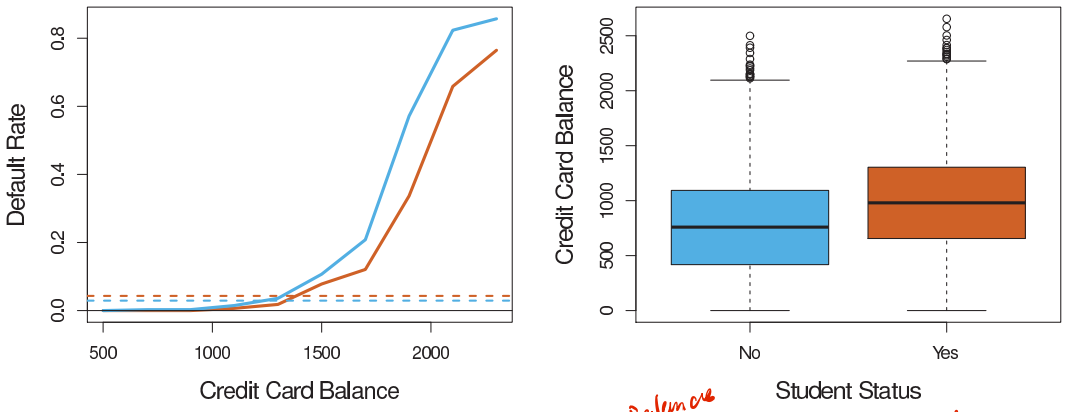
correlation should be 0 (Removed) ← income

Too large → Not significant

P-value for *Income* is large, indicating that it has no predictive power to the possibility of default. Thus it can and should be removed from the model

Coefficient for student is negative, indicating that students are less likely to default than non-students — Contradictory?

Confounding between *Balance* X_1 **and** *Student status* X_2



balance ↑ → status of students

Left panel: plots of default probability $\hat{p}(X_1, X_2)$ against X_1 with $X_2 = 1$ (student) in orange, and $X_2 = 0$ (non-student) in blue.

With the same balance, students are less likely to default than non-students

Right panel: Boxplots of *Balance* for students in orange, and non-students in blue.

Summary — information for credit card companies

- Individual income has no predictive power on default
- A student is riskier than a non-student in general
- With the same credit card balance, a student is less riskier than a non-student

Logistic regression in R

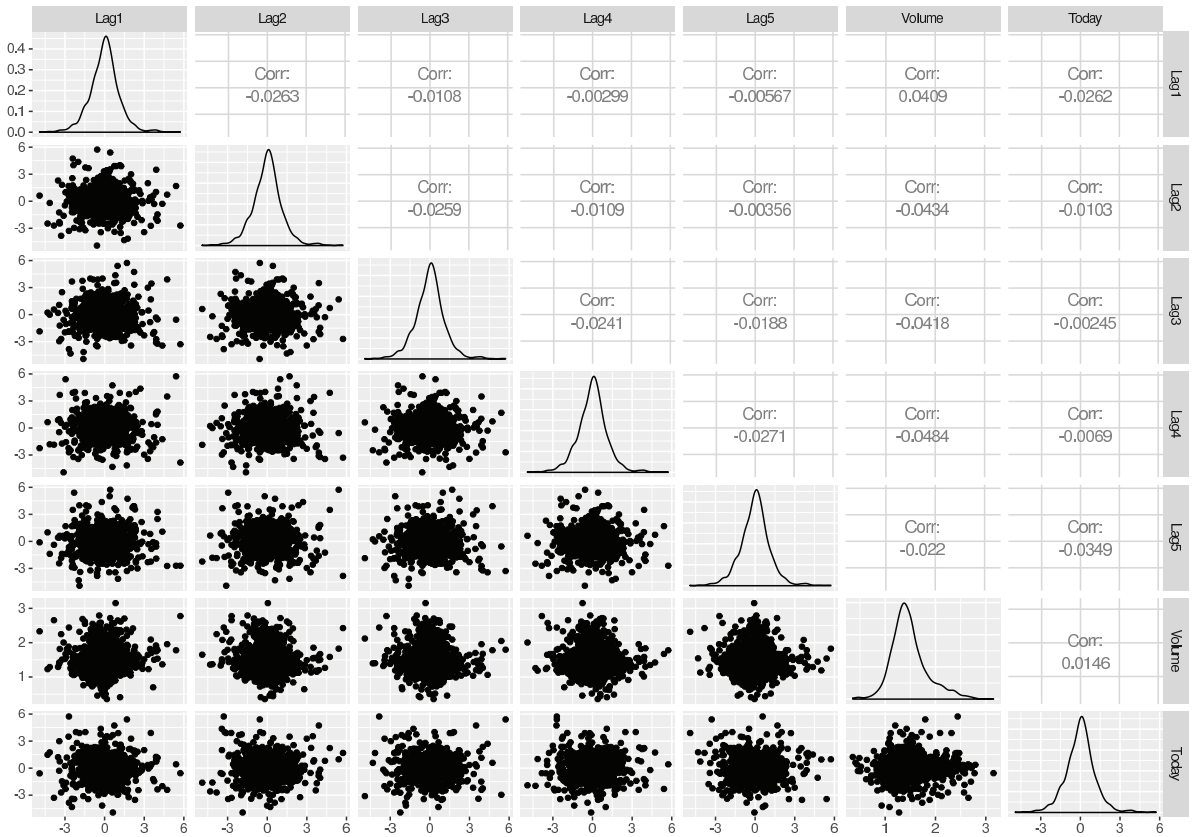
Consider dataset `Smarket` from package `ISLR`, which contains the daily percentage returns of S&P 500 index over 1,250 days in 2001 – 2005.

```
> library(ISLR)
> names(Smarket)
[1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"      "Volume"
[8] "Today"     "Direction"
> dim(Smarket)
[1] 1250    9
> summary(Smarket)
      Year           Lag1                Lag2                Lag3                Lag
 Min.   :2001   Min.   :-4.9220   Min.   :-4.9220   Min.   :-4.9220   Min.   :-4.922
 1st Qu.:2002   1st Qu.:-0.6395   1st Qu.:-0.6395   1st Qu.:-0.6400   1st Qu.:-0.640
 Median :2003   Median : 0.0390   Median : 0.0390   Median : 0.0385   Median : 0.038
 Mean   :2003   Mean   : 0.0038   Mean   : 0.0039   Mean   : 0.0017   Mean   : 0.001
 3rd Qu.:2004   3rd Qu.: 0.5967   3rd Qu.: 0.5967   3rd Qu.: 0.5967   3rd Qu.: 0.596
 Max.   :2005   Max.   : 5.7330   Max.   : 5.7330   Max.   : 5.7330   Max.   : 5.733
      Lag5              Volume           Today          Direction
 Min.   :-4.9220   Min.   :0.3561   Min.   :-4.9220   Down:602
 1st Qu.:-0.6400   1st Qu.:1.2574   1st Qu.:-0.6395   Up  :648
 Median : 0.0385   Median :1.4229   Median : 0.0385
 Mean   : 0.0056   Mean   :1.4783   Mean   : 0.0031
 3rd Qu.: 0.5970   3rd Qu.:1.6417   3rd Qu.: 0.5967
```

```
 Max.   : 5.7330   Max.   :3.1525   Max.   : 5.7330

> round(cor(Smarket[-c(1,9)]), digits=4)
          Lag1    Lag2    Lag3    Lag4    Lag5  Volume   Today
Lag1    1.0000 -0.0263 -0.0108 -0.0030 -0.0057  0.0409 -0.0262
Lag2   -0.0263  1.0000 -0.0259 -0.0109 -0.0036 -0.0434 -0.0103
Lag3   -0.0108 -0.0259  1.0000 -0.0241 -0.0188 -0.0418 -0.0024
Lag4   -0.0030 -0.0109 -0.0241  1.0000 -0.0271 -0.0484 -0.0069
Lag5   -0.0057 -0.0036 -0.0188 -0.0271  1.0000 -0.0220 -0.0349
Volume  0.0409 -0.0434 -0.0418 -0.0484 -0.0220  1.0000  0.0146
Today  -0.0262 -0.0103 -0.0024 -0.0069 -0.0349  0.0146  1.0000

> install.packages("GGally")
> library(GGalley)
> ggpairs(Smarket[,-c(1,9)])
```

There are hardly any correlations among today's return and its lagged values – the efficient market hypothesis!

Note. `GGalley` is an added-on package to `ggplot2`. `ggpairs` presents more information than `pairs` – a standard plot in R.



Now we fit a logistic model for predicting the direction of market:

```
> logistic.Smarket=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Smarket,
                       family=binomial)
> summary(logistic.Smarket)

Call:
glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
    Volume, family = binomial, data = Smarket)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.446  -1.203   1.065   1.145   1.326

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.126000   0.240736  -0.523    0.601
Lag1        -0.073074   0.050167  -1.457    0.145
Lag2        -0.042301   0.050086  -0.845    0.398
Lag3         0.011085   0.049939   0.222    0.824
Lag4         0.009359   0.049974   0.187    0.851
Lag5         0.010313   0.049511   0.208    0.835
Volume       0.135441   0.158360   0.855    0.392

AIC: 1741.6

Number of Fisher Scoring iterations: 3
```

The most significant predictor is Lag1 with a negative coefficient; indicating mean regression. Note the p-value is 0.145 — not very significant.

Function `predict` can be used to predict the probability that the market will go up, given values of the predictors. The type="response" option tells R to output probabilities of the form $P(Y = 1|X)$, as opposed to other information such as the logit. If no data set is supplied, the probabilities are computed for the training data that was used to fit the logistic regression model. Note $Y = 1$ stands for 'up', as

```
> contrasts(Direction)
Down  0
Up    1

> pred.Smarket=predict(logistic.Smarket, type="response")
> pred.SmarketDiction=rep("Down", 1250) # a sequence of "Down" repeated 1250 times
> pred.SmarketDiction[pred.Smarket>.5]="Up"
```

```
> table(pred.SmarketDiction, Direction)
                   Direction
pred.SmarketDiction Down  Up
               Down  145 141
               Up    457 507
> (145+507)/1250
[1] 0.5216  # accuracy rate on the training data
```

Given most the predictors are insignificant, we re-run the fitting using only Lag1 and Lag2:

```
>logistic.Smarket=glm(Direction~Lag1+Lag2, data=Smarket, family=binomial)
> pred.Smarket=predict(logistic.Smarket, type="response")
> pred.SmarketDiction[pred.Smarket>.5]="Up"
> table(pred.SmarketDiction, Direction)
                   Direction
pred.SmarketDiction Down  Up
               Down   93  84
               Up    509 564
> (93+564)/1250
[1] 0.5256
```

Those accuracy rates are expected to be greater than the real rates when we use the fitting on new data.

An alternative is to split the sample into training and testing subsets:

```
> Smarket.2005=Smarket[Year>=2005,]
> dim(Smarket.2005)
[1] 252   9
> Direction.2005=Direction[Year==2005]
> logistic.Smarket=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Smarket,
                 family=binomial, subset=(Year <2005))
> Prob.2005=predict(logistic.Smarket, Smarket.2005, type="response")
> PredDirection.2005=rep("Down", 252)
> PredDirection.2005[Prob.2005>0.5]="Up"
> table(PredDirection.2005, Direction.2005)
                  Direction.2005
PredDirection.2005 Down Up
              Down   77 97
              Up     34 44
> (77+44)/252
[1] 0.4801587  # Accuracy rate on testing sample
```

Now we use only two lagged variables:

```
> logistic.Smarket=glm(Direction~Lag1+Lag2, data=Smarket, family=binomial,
              subset=(Year <2005))
> Prob.2005=predict(logistic.Smarket, Smarket.2005, type="response")
> PredDirection.2005=rep("Down", 252)
```

```
> PredDirection.2005[Prob.2005>0.5]="Up"
> table(PredDirection.2005, Direction.2005)
                  Direction.2005
PredDirection.2005 Down  Up
              Down   35  35
              Up     76 106
> (35+106)/252
[1] 0.5595238
```

The overfitted model (with 6 predictors) perform about the same on the training data as the model with the 2 predictors. However it performs much worse on the testing data!

Chapter 4. Regression Analysis

- Simple linear regression
- Multiple linear regression
- Understanding regression results
- Nonlinear effects in linear regression
- Regression trees
- Bagging, random forests and boosting
- From global modelling to local fitting
- Regression analysis in R

Linear regression is one of the oldest and also the most frequently used statistical or data-mining methods

A useful tool for predicting a quantitative response based on some observable features/predictors/variables

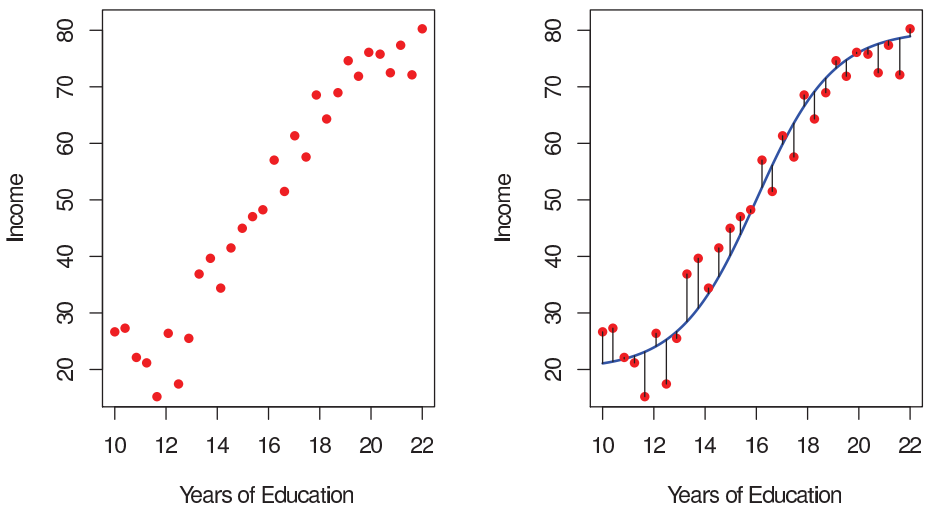
Many fancy data-mining methods can be viewed as the extensions of linear regression

Further reading:

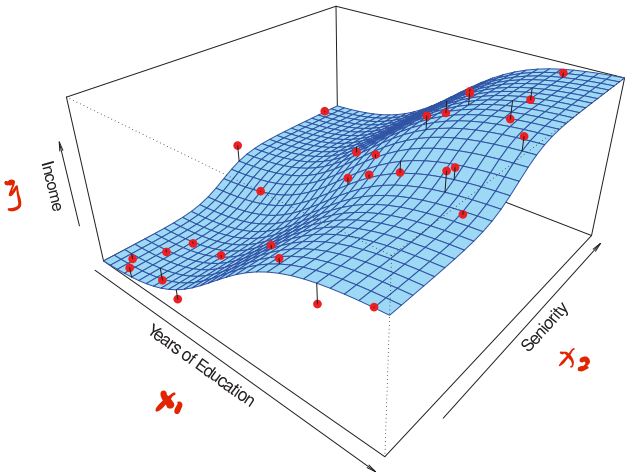
James et al. (2013) Chapter 3 & Section 4.6,

Provost and Fawcett (2013) Chapter 4.

Task of regression analysis: estimate the true curve from data points.



Left panel: Incomes of 30 individuals are plotted against their years of education.
Right panel: The curve (or regression curve) represents the true underlying relationship between income and years of education



Plot of income as a function of years of education and seniority. The blue surface represents the true relationship.

Task: to estimate the surface from the data.

Much harder! Data points are sparse: curse-of-dimensionality

Way-out: impose some parametric forms for the unknown surface, such as linear regression models.

Mincer Equation: How is one's earning related to human capital?

$$\log(Y) = \beta_0 + \beta_1 X + \beta_2 U + \beta_3 U^2 + \varepsilon,$$

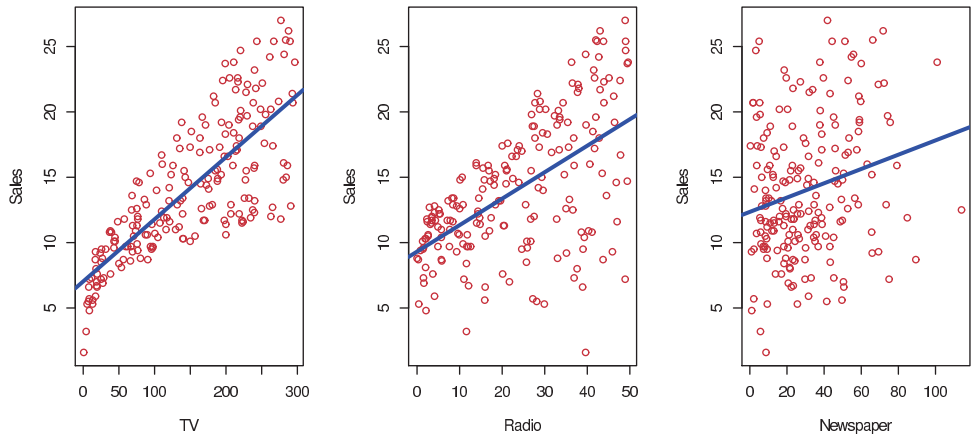
Y — earning
X — education capital: No. of years in school/university
U — experience capital: No. of years in employment

Rate of return to education: β_1

β_1, β_2 are positive, and β_3 tends to negative and small

This is a simple linear regression model.

We may even add an interaction term: $X \cdot U$



Sales are plotted against the ad budget in, respectively, TV, radio and newspaper. In each plot, the blue straight line is the regression estimator $Y = \widehat{\beta}_0 + \widehat{\beta}_1 X$.

Advertising in TV is most effective. Is there any added value to advertise in addition in radio and newspaper?

What can we learn from regression?

- is there a relationship between ad budget and sales?
- how strong is the relationship if there is?
- is the relationship linear?
- which media contribute to sales?
- how accurately can we estimate the effect of each medium on sales?
- how accurately can we predict the future sales?
- how should we distribute ad budget over different media? (synergy effect or interaction effect)

Simple line regression: $Y = \beta_0 + \beta_1 X + \varepsilon$

With n data points $(x_1, y_1), \cdots, (x_n, y_n)$, we calculate the LSE:

$$(\widehat{\beta}_0, \widehat{\beta}_1) = \arg\min_{\beta_0, \beta_1} \sum_{i=1}^{n} (y_i - \beta_0 - x_i \beta_1)^2.$$

It can be shown that

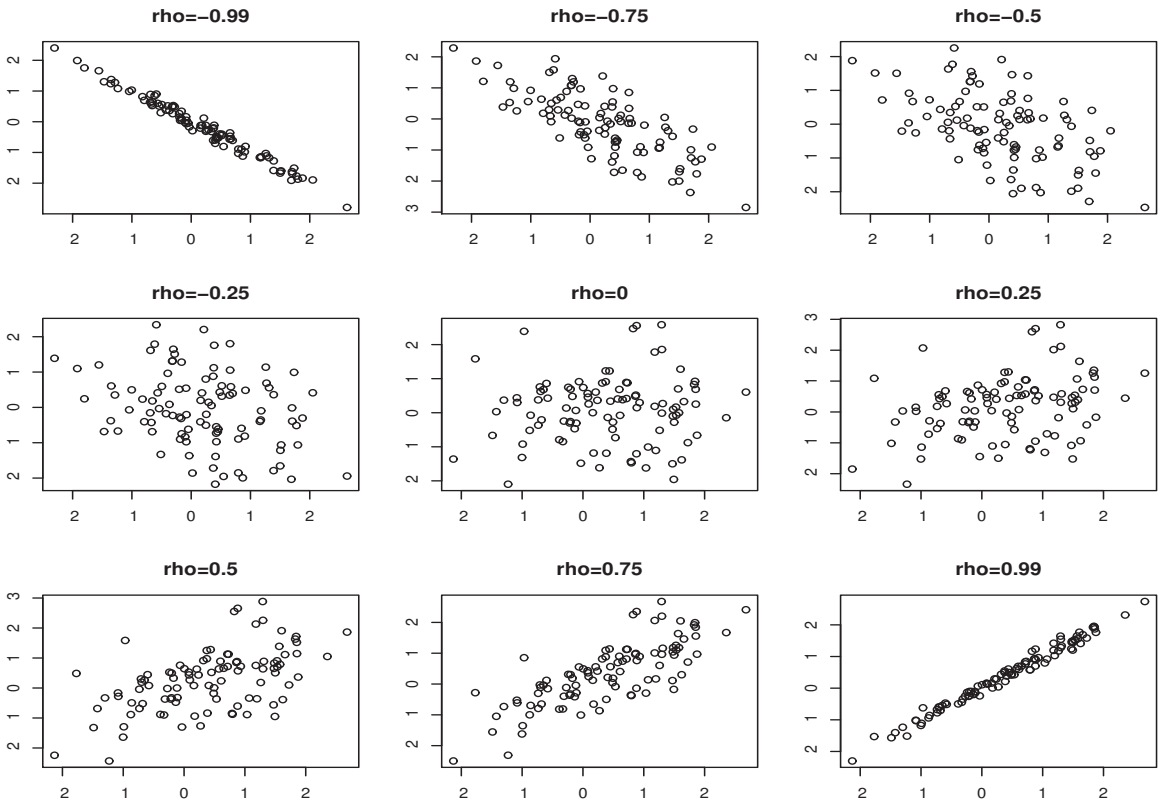
$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \qquad \widehat{\beta}_0 = \bar{y} - \bar{x}\widehat{\beta}_1,$$

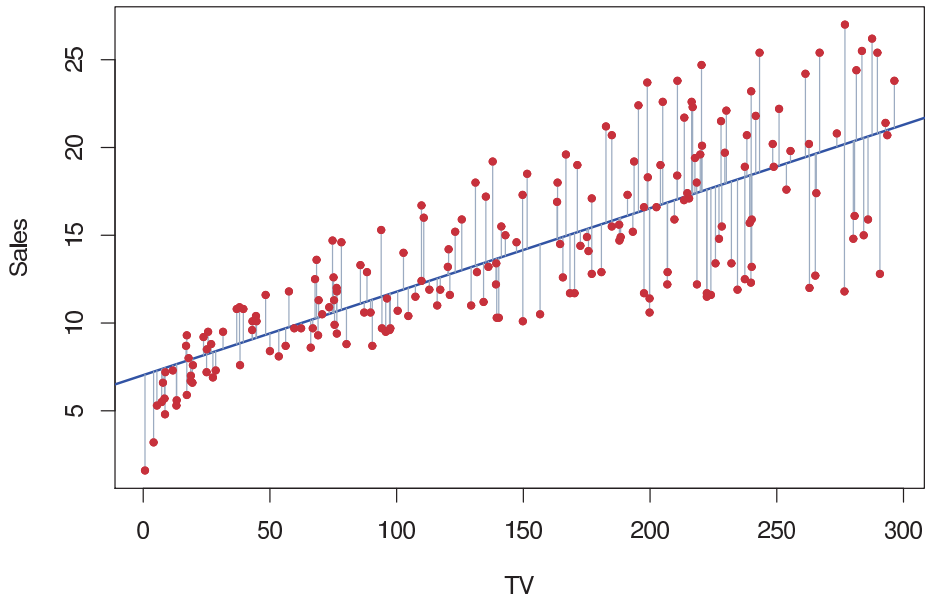
where $\bar{y} = n^{-1} \sum_i y_i$, $\bar{x} = n^{-1} \sum_i x_i$.

Interpretation: make RSS $= \sum_i \varepsilon_i^2$ as small as possible, where

$$\varepsilon_i = y_i - \beta_0 - x_i \beta_1, \qquad i = 1, \cdots, n.$$

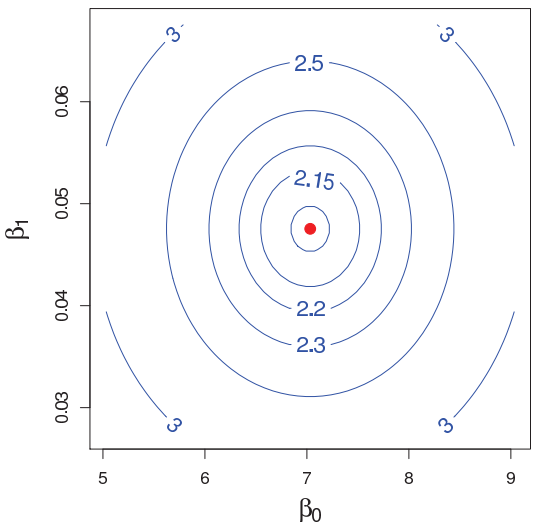
Sample correlation: $\widehat{\rho}_{x,y} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$. Note $\widehat{\beta}_1 = \widehat{\rho}_{x,y} \sqrt{\frac{\sum_i (y_i - \bar{y})^2}{\sum_i (x_i - \bar{x})^2}}$.

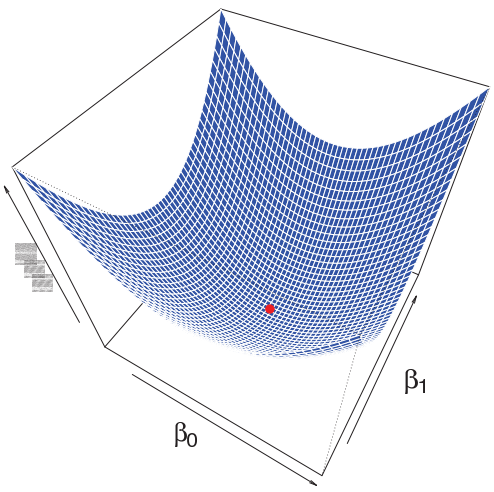




Fitting: Sales $= \beta_0 + \beta_1 \text{TVad} + \varepsilon$. LSE: $\widehat{\beta}_0 = 7.03$, $\widehat{\beta}_1 = 0.0475$

Is it right to conclude that increasing one unit of `TV` budget leads to an increase in `sales` by 0.0475 unit?





How accurate are $\hat{\beta}_0$, $\hat{\beta}_1$?

Assumption: $E(\varepsilon) = 0$, $\text{Var}(\varepsilon) = \sigma^2$. Then

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2.$$

Standard errors:

$$\text{SE}(\hat{\beta}_0) = \hat{\sigma}\left(\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^{1/2}, \qquad \text{SE}(\hat{\beta}_1) = \left(\frac{\hat{\sigma}}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right)^{1/2}$$

Note. $\text{SE}(\hat{\beta}_1)$ is an approximation for $\{\text{Var}(\hat{\beta}_1)\}^{1/2}$.

95% Confidence intervals (error bars): $\hat{\beta}_j \pm 1.96 \cdot \text{SE}(\hat{\beta}_j)$, or simply,

$$\hat{\beta}_j \pm 2 \cdot \text{SE}(\hat{\beta}_j), \qquad j = 1, 2$$

$$\frac{\hat{\beta} - \beta}{SE(\hat{\beta})} \overset{A}{\sim} N(0, 1)$$

For the model $\text{Sales} = \beta_0 + \beta_1 \text{TVad} + \varepsilon$, the 95% Confidence interval is $[6.130, 7.935]$ for β_0, and $[0.042, 0.053]$ for β_1.

Interpretation: In the absence of any advertising, sales will on average fall between 6.130 and 7.935. Furthermore, an increase of 1000 units in TV advertising is likely to increase sales between 42 and 53 units.

Since $\hat{\beta}_1 = 0.0475$ is so small, is it possible that $\beta_1 = 0$ in the sense that there is no relationship between sales and TV advertising.

Hypothesis tests. To test the null hypothesis

H_0 : there is no relationship between sales and TV advertising

which is equivalently to $H_0 : \beta_1 = 0$

Since the 95% confidence interval for β_1 does not contain 0, we reject H_0.

How much is the sales fluctuation due to TV advertising?

Total SS: $\sum_{i=1}^{n}(y_i - \bar{y})^2$

Regression SS: $\sum_{i=1}^{n}(\widehat{y}_i - \bar{y})^2$, where $\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$

Residual SS: $\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2$

It can be shown that

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\widehat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \widehat{y}_i)^2.$$

Regression correlation coefficient:

$$R = \Big(\frac{\text{Regression SS}}{\text{Total SS}}\Big)^{1/2} = \Big(1 - \frac{\text{Residual SS}}{\text{Total SS}}\Big)^{1/2}.$$

Then $R \in [0, 1]$.

Interpretation: $100R^2$ is the percentage of the total variation of Y explained by the regressor X.

Adjusted regression correlation coefficient:

$$R_{adj} = \Big(1 - \frac{(\text{Residual SS})/(n-2)}{(\text{Total SS})/(n-1)}\Big)^{1/2}.$$

For the model $\text{Sales} = \beta_0 + \beta_1 \text{TVad} + \varepsilon$,

$$\widehat{\sigma} = 3.26, \qquad R^2 = 61.2\%$$

Multiple Linear Regression: $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$.

Available data: $(y_i, x_{i1}, \cdots, x_{ip})$, $i = 1, \cdots, n$.

LSE: $\widehat{\beta}_0, \widehat{\beta}_1, \cdots, \widehat{\beta}_p$ are obtained by minimizing

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \cdots - \beta_p x_{ip})^2.$$

Sum of squared residuals:

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \widehat{y}_i)^2, \quad \text{where } \widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_{i1} + \cdots + \widehat{\beta}_p x_{ip}.$$

Square of regression correlation coefficient:

$$R^2 = 1 - \frac{\text{RSS}}{\sum_i (y_i - \bar{y})^2}, \qquad R^2_{adj} = 1 - \frac{\text{RSS}/(n-p-1)}{\sum_i (y_i - \bar{y})^2/(n-1)}$$

Fitting: $\text{sales} = \beta_0 + \beta_1 \cdot \text{TV} + \beta_2 \cdot \text{radio} + \beta_3 \cdot \text{newspaper} + \varepsilon$

	Coefficient	Std error	t-statistic	P-value
Intercept	2.939	0.3119	9.24	<0.0001
TV	0.046	0.0014	32.81	<0.0001
ratio	0.189	0.0086	21.89	<0.0001
newspaper	-0.001	0.0059	-0.18	0.8599 → so large ↳ Removed

1. Given the budget for the two other media, an increase of one unit in TV budget will bring in an increase of 0.046 unit in sales, an increase of one unit in radio budget will bring in an increase of 0.189 unit in sales.

   Is it more effective to advertise in ratio than in TV?

   Not necessarily, the fitted model is valid only within the range of the learning data (i.e. observations).

   Caution should be exercised when extrapolating a fitted model outside of observed range

2. $\widehat{\beta}_3 = -0.001$ with the 95% confidence interval $-0.001 \pm 2 \times 0.0059 = [-0.0128, 0.0108]$. The interval contains the value 0. Hence we cannot reject the hypothesis $H_0 : \beta_3 = 0$.

   Having advertising on TV and radio, the effect of advertising on newspaper is not significant.

   However the cross correlations are

	TV	radio	newspaper	sales
TV	1.0000	0.0548	0.0567	0.7822
radio		1.0000	0.3541	0.5762
newspaper			1.0000	0.2283
sales				1.0000

   Note the correlation between ratio and newspaper budgets is 0.3541!

   The fitted univariate models are:

	Coefficient	Std error	t-statistic	P-value
Intercept	7.0325	0.4758	15.36	<0.0001
TV	0.0475	0.0027	17.67	<0.0001
Intercept	9.312	0.563	16.54	<0.0001
radio	0.203	0.020	9.92	<0.0001
Intercept	12.351	0.621	19.88	<0.0001
newspaper	0.055	0.017	3.30	0.00115

   ↳ make some impacts

   Thus, the estimated coefficients for TV and ratio are about the same in both the multiple regression and the simple univariate regression.

   The coefficients for newspaper in the multiple regression and the univariate regression are significantly different.

Two possible interpretations:

(a) The effect from advertising in newspaper is encapsulated in that from TV or radio, i.e. the sales will increase by advertising in newspaper even if no advertising in both TV and radio

(b) Advertising on newspaper has no effect on sales. The significance in the univariate regression is due to the significant correlation between newspaper budget and radio budget, i.e. newspaper advertising acts as a surrogate for ratio advertising.
↓substitutes

Common sense would suggest that (a) is more likely the case for this example, or not? (as only fewer people read news papers those days)

3. Since newspaper is not significant, we can refine the model using TV and radio only.

Variable selection: How many variables should be selected in the model?

A general principle: choose the model which minimizes a certain criterion defined as, for example,
→ Trade-off

(Goodness of fit of model) + (Penalty for model complexity)

↓ Why RSS should be min; ↓ → How many of Xs can be removed. How Penalty complex

- *Forward selection*. Starting with the model with a intercept only, add one variable each time such that the added variable leads to the maximum reduction in residual sum of squares (RSS). Stop according to a certain criterion.

- *Backward selection*. Start with the full model, delete each time one variable such that the deleted variable causes the minimum increase in RSS. Stop according to a certain criterion.

- *Stepwise selection*. After adding each variable, delete all *redundant* variables according to an appropriate criterion before

↳ check multicollinearity ☆.

adding a new variable. Stop when no variables can be added or deleted.

Note. None of the above procedures will give you the overall optimal model. They are tradeoff between searching for a good model and computation efficiency. The stepwise selection procedure proves to be more effective.

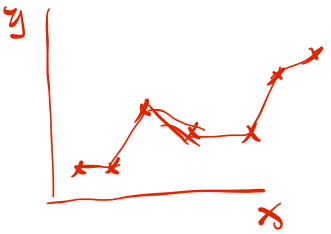
4. How well the various models fit the data?

   Two simple measures: R^2 and $\hat{\sigma}$

   Note. Neither R^2 nor $\hat{\sigma}$ can be used as a sole measure for variable selection!

   For regression models for `sales`,

regressors	(T, r, n)	(T, r)	T	r	n
R^2	0.8972	0.89719	0.6119	0.332	0.05212
$\hat{\sigma}$	1.686	1.681	3.259	4.275	5.092



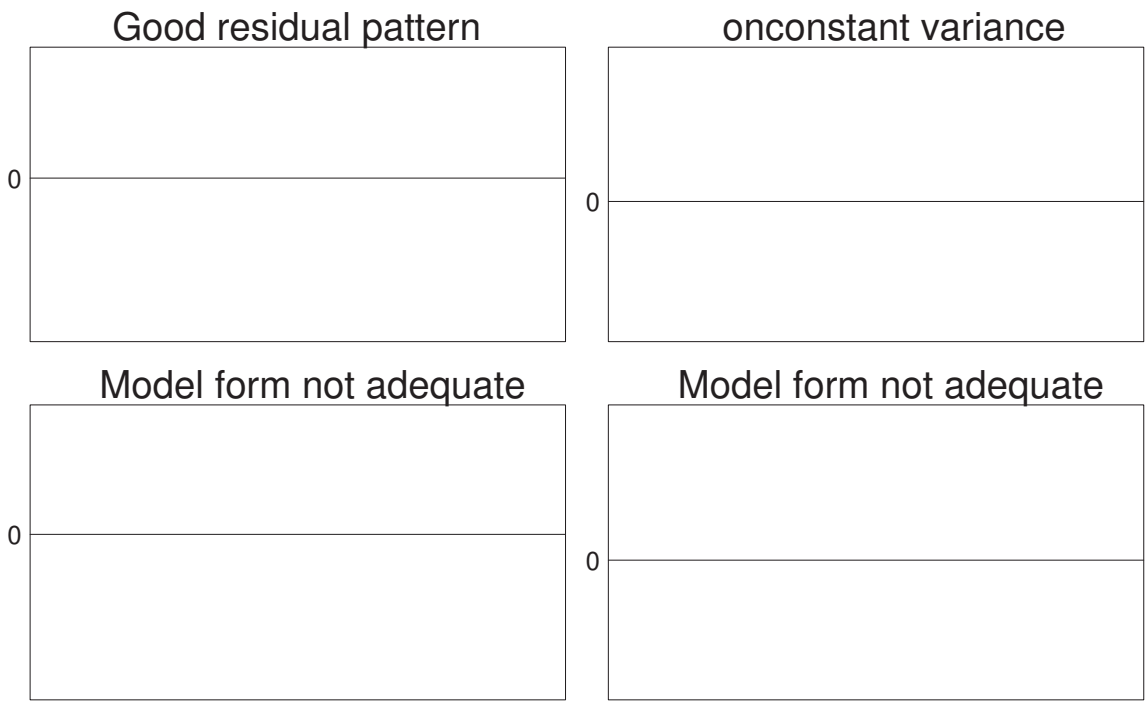
The model with two regressors (`TV, ratio`) fits the data better.

5. Goodness-of-fit: checking residuals $\hat{\varepsilon}_i = y_i - \hat{y}_i,\ i = 1, \cdots, n.$

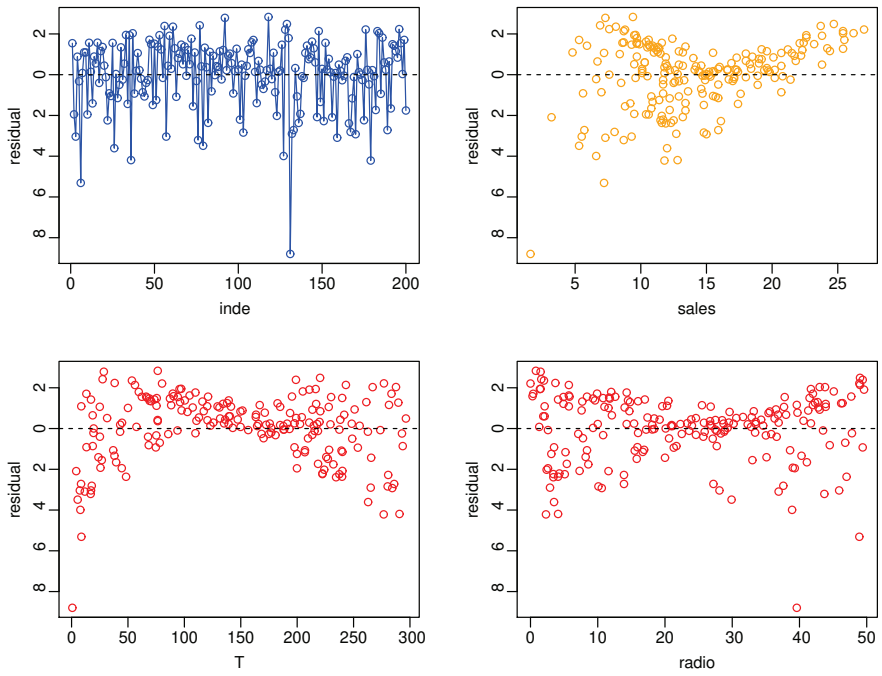
   A good fitting leads to patternless residuals.

   Plot residuals against index, y_i or $x_{i1}, \cdots, x_{ip}$.

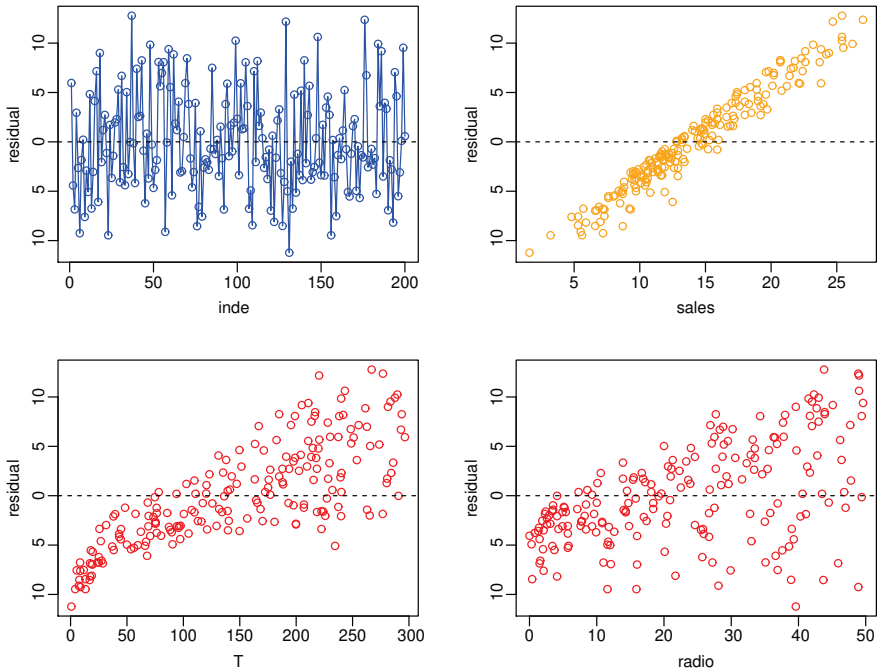
   Powerless in detecting overfitting!



Residuals: `sales` $-$ (2.9211 $+$ 0.0458`TV` $+$ 0.1880`radio`)



Residuals: `sales` $-$ (12.3514 + 0.0547`newspapge`)



6. Prediction

Given new values $x_1, \cdots, x_p$, we can predict the corresponding y by

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \cdots + \widehat{\beta}_p x_p.$$

The error in predicting y by $\widehat{y}$ may be caused by 3 sources:

(a) Estimation errors in $\widehat{\beta}_0, \widehat{\beta}_1, \cdots, \widehat{\beta}_p$.

This type of errors can be quantified by constructing a confidence interval for

$$E(y) = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p.$$

(b) Unobserved error $\varepsilon = y - E(y)$.

By estimating $\sigma^2 = \text{Var}(\varepsilon)$, we can enlarge the confidence interval above to a predictive interval.

(c) Model bias, i.e. $E(y)$ may not be linear in $x_1, \cdots, x_p$.

This is more difficult to quantify. Typically a linear model is regarded as an approximation.

With the model $\texttt{sales} = 2.9211 + 0.0458\texttt{TV} + 0.1880\texttt{radio}$, the predicted sales for spending 100 on TV and 20 on radio is 11.26, with the 95% confidence interval $[10.99, 11.53]$ and the 95% predictive interval $[7.93, 14.58]$.

Note. Confidence interval is for $E(y)$, i.e. spending 100 on TV and 20 on radio over many cities, the average sales over those cities will fall between 10.99 and 11.53 (with the probability 95%).

Predictive interval is for y, i.e. spending 100 on TV and 20 on radio in one city, the sales will fall between 7.93 and 14.58 (with the probability 95%).

7. Nonlinear regressors.

Linear regression models are linear in coefficients $\beta_0, \beta_1, \cdots, \beta_p$, while regressors $X_1, \cdots, X_p$ can be replaced by any known functions of them.

The capacity is far beyond linear relationship, as the regressors $x_1, \cdots, x_p$ may be

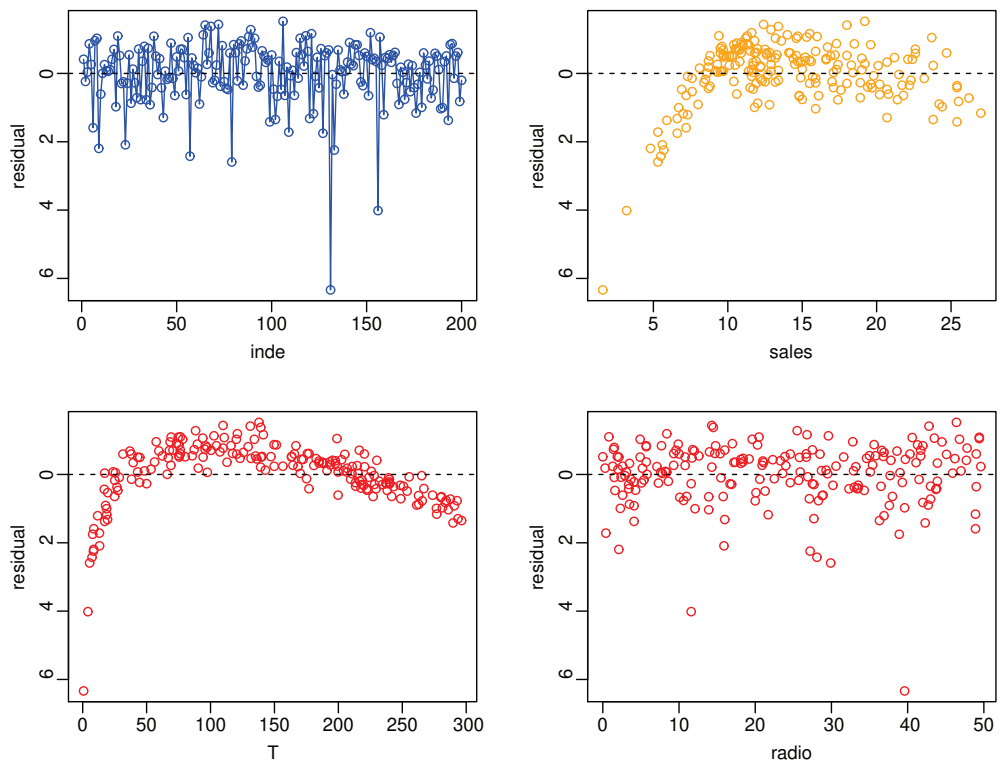
- quantitative inputs
- transformations of quantitative inputs, such as log, square-root etc
- interactions between variables, e.g. $x_3 = x_1 x_2$
- basis expansions, such as $x_2 = x_1^2, x_3 = x_1^3, \cdots$
- numeric or "dummy" coding of the levels if qualitative variables.

Fitting: $\texttt{sales} = \beta_0 + \beta_1 \texttt{TV} + \beta_2 \texttt{radio} + \beta_3 \texttt{TV} \cdot \texttt{radio} + \varepsilon$

	Coefficient	Std error	t-statistic	P-value
Intercept	6.7502	0.248	27.23	$<$0.0001
TV	0.0191	0.002	12.70	$<$0.0001
radio	0.0289	0.009	3.24	0.0014
TV·radio	0.0011	0.000	20.73	$<$0.0001

$R^2 = 0.9678, \qquad \widehat{\sigma} = 0.9435.$

A better fitting???



A market plan for sales

1. There is a clear relationship between sales and advertising budget, as the hypotheses $\beta_1 = 0$ (TV) and $\beta_2 = 0$ (radio) is overwhelmingly rejected. But there is little impact on sales by advertising on newspaper.

2. The strength of the relationships can be measured by either R^2 and $\hat{\sigma}$ from the judiciously selected models. The about 90% of variation in sales is due to the advertising on TV and radio. The recommended model for this data set is

$$\texttt{sales} = 2.9211 + 0.0458\texttt{TV} + 0.1880\texttt{radio}.$$

3. The effect on sales from the advertising can be reflected by the estimates for β_1 (TV) and β_2 (radio), or more precisely their confidence intervals (0.043, 0.049) and (0.172, 0.206). For example, an increase of 1 unit budget in TV advertising would lead to an increase of sales between 0.043 and 0.049 unit. But one should be cautious in extrapolating the results out of the observed range.

4. We can predict the future sales based on the above model. There are two types intervals for gauging the prediction errors: confidence interval for predicting sales over many cities, and predictive interval for predicting sales for one city.

→ only one x, but x^2, x^5 ....

Polynomial regression: an illustration by example

The data set `Auto.txt` contains various indices for 387 cars. Let us consider the relationship between `mpg` (gas mileage in miles per gallon) versus `horsepower`.

Looking at scatter plots, the relationship does not appear to be linear. We fit polynomial regression:

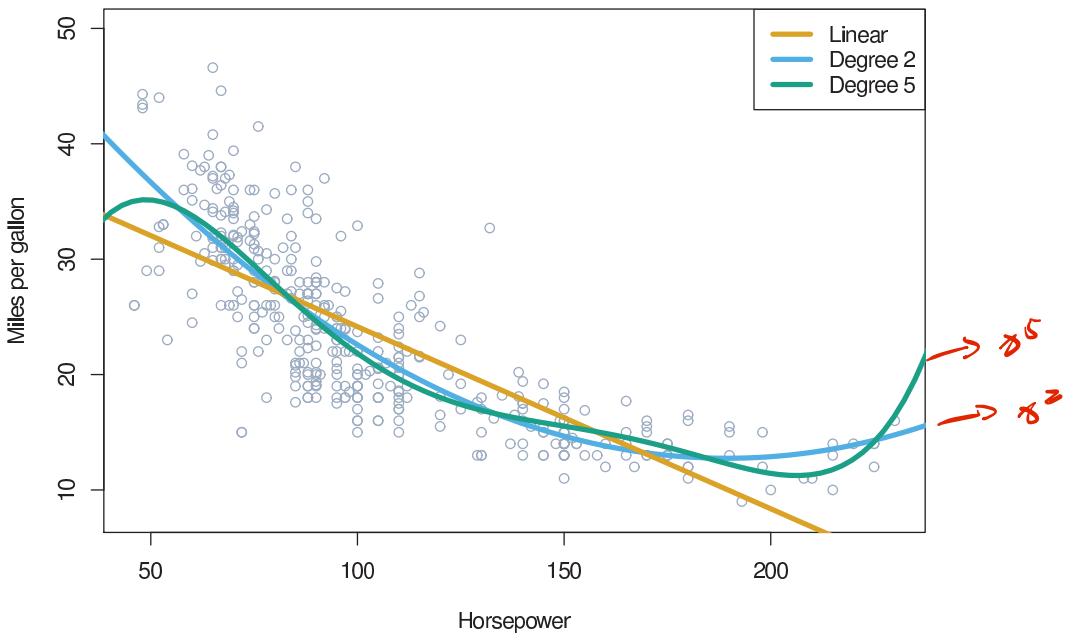
$$\texttt{mpg} = \beta_0 + \beta_1 \texttt{hpower} + \cdots + \beta_p \texttt{hpower}^p + \varepsilon$$

for $p = 1, 2, \cdots$.

The results for $p = 2$ are listed below

	Coefficient	Std error	t-statistic	P-value
Intercept	56.9001	1.8004	31.6	<0.0001
horsepower	-0.4662	0.0311	-15.0	<0.0001
horsepower^2	0.0012	0.0001	10.1	<0.0001

poly regression not good idea.



→ x^5

→ x^2

Extrapolation of polynomial regression can be explosive!

↳ put small boundaries.

Linear Regression in R

We use dataset `Boston` in package `MASS` as an illustration.

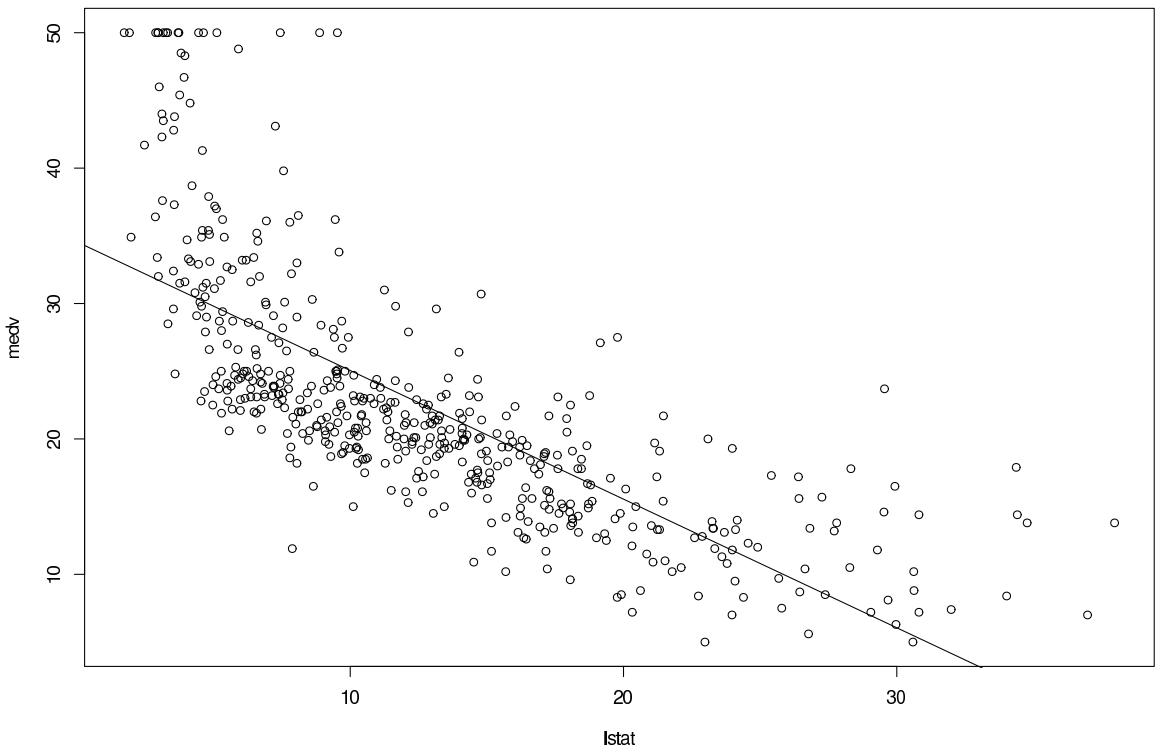
```
> install.packages("MASS")
> library(MASS)
> names(Boston)
 [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"     "dis"
 [9] "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
> View(Boston)
```

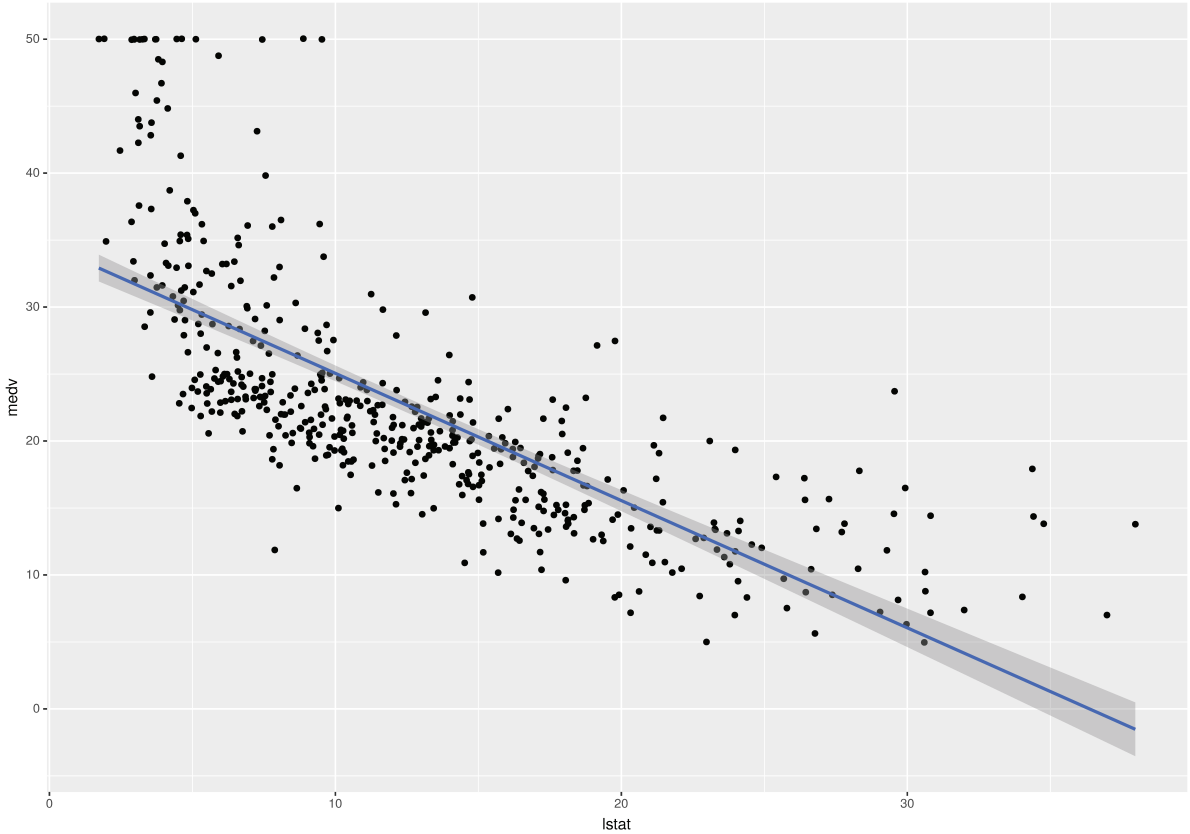
The data record `medv` (median house value) for 506 neighbourhoods around Boston, together with other 13 variables including `rm` (average number of rooms), `age` (average age of houses), `lstat` (percent of households with low socioeconomic status). More info is available from `?Boston`.

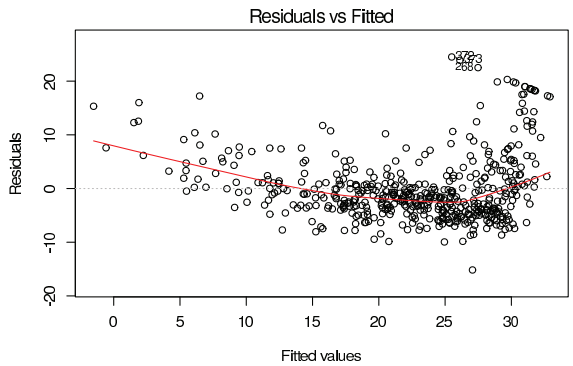
```
> attach(Boston)
> lm1.Boston=lm(medv~lstat)
> summary(lm1.Boston)

Call:
lm(formula = medv ~ lstat)
Residuals:
    Min      1Q  Median      3Q     Max
-15.168  -3.990  -1.318   2.034  24.500
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.55384    0.56263   61.41   <2e-16 ***
lstat       -0.95005    0.03873  -24.53   <2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 6.216 on 504 degrees of freedom
Multiple R-squared:  0.5441,Adjusted R-squared:  0.5432
F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16

> names(lm1.Boston)  # additional info/components in output
 [1] "coefficients"  "residuals"     "effects"    "rank"  "fitted.values" "assign"
 [7] "qr"            "df.residual"   "xlevels"    "call"  "terms"         "model"
> lm1.Boston$rank   # to call for components in output
  # rank of X-matrix in matrix representation of regression model
[1] 2

> confint(lm1.Boston)  # Confidence intervals for coefficients
                2.5 %     97.5 %
(Intercept) 33.448457 35.6592247
lstat       -1.026148 -0.8739505
```

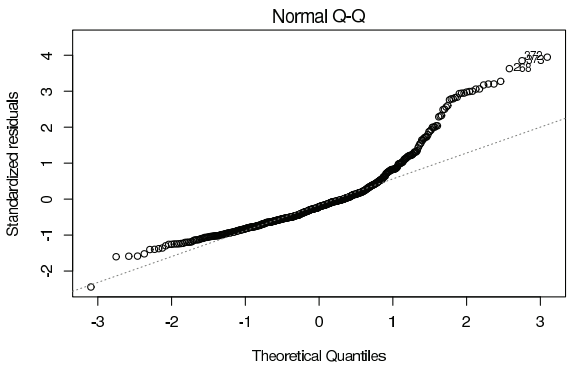
```
> predict(lm1.Boston, data.frame(lstat=c(5,10,15)), interval="confidence")
       fit      lwr      upr
1 29.80359 29.00741 30.59978
2 25.05335 24.47413 25.63256
3 20.30310 19.73159 20.87461

> predict(lm1.Boston, data.frame(lstat=c(5,10,15)), interval="prediction")
       fit       lwr      upr
1 29.80359 17.565675 42.04151
2 25.05335 12.827626 37.27907
3 20.30310  8.077742 32.52846

> plot(lstat, medv); abline(lm1.Boston)
> library(ggplot2) # re-do the plot using ggplot2
> ggplot(Boston, aes(lstat, medv))+geom_jitter()+geom_smooth(method=lm)

> par(mfrow=c(2,2)) # put 4 plots in one panel of 2x2 format
> plot(lm1.Boston)  # 4 plots for model diagnostic checking
```

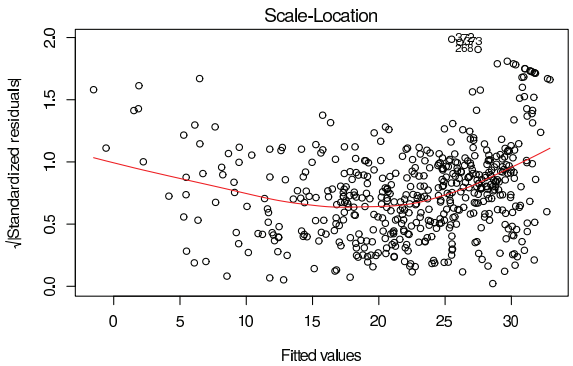
Residual plots are not patternless! The Q-Q plot indicates that residuals are not normal-like.

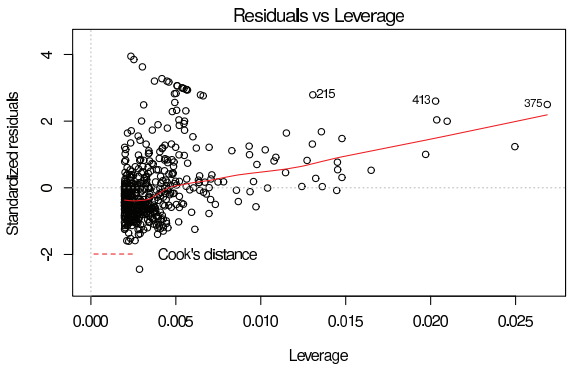



medv
lstat
0
10
20
30
40
50


Residuals vs Fitted
Residuals
Fitted values


Normal Q-Q
Standardized residuals
Theoretical Quantiles


Scale-Location
√|Standardized residuals|
Fitted values


Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

`plot(lm1.Boston)` produces 4 diagnostic plots for the fitted model.

Top-left: Residuals vs Fitted — a plot of $\widehat{\varepsilon}_i$ against $\widehat{y}_i$. If the model is adequate, this plot should be patternless, as $\widehat{\varepsilon}_i$ should behave like random noise. The plot is helpful in detecting outliers, i.e. those y_i far away from $\widehat{y}_i$.

Top-right: Normal Q-Q — a plot of the quantiles of standardized residuals against the $N(0,1)$ quantiles. If residuals are normally distributed, the points should be on the straight line. It is particularly effective in highlighting *heavy tails: the points near the left-end are below the straight line, and the points close to the right-end are above the straight line*.

Bottom-left: Scale-Location — a plot of $\sqrt{|\widetilde{\varepsilon}_i|}$ against $\widehat{y}_i$, where $\widetilde{\varepsilon}_i$ denotes the standardized residual. This plot should be patternless too if the fitting is adequate. It is powerful in detecting inhomogeneous variances among different observations. Note that for $Z \sim N(0, \sigma^2)$, $|Z|$ is heavily skewed to the left, and $\sqrt{|Z|}$ is much less skewed.

Bottom-right: Residuals vs Leverage – a plot of $\widetilde{\varepsilon}_i$ against leverage h_{ii}, where h_{ii} is the (i,i)-th element of the hat matrix $\mathbf{P}_x$ which defines the fitted value $\widehat{\mathbf{y}} = \mathbf{P}_x\mathbf{y}$, $\mathbf{y} = (y_1, \cdots, y_n)'$.

A **leverage point** is an observation which has a great influence on the analysis. The amount of the leverage of the i-th observation is reflected by h_{ii}. It can be shown that the total leverage is

$$\sum_{i=1}^{n} h_{ii} = \text{trace}(\mathbf{P}_x) = p + 1.$$

Therefore the average leverage for each observations is $\frac{p+1}{n}$.

A rule of thumb: if $h_{ii} > \frac{2(p+1)}{n}$, the i-th observation is a leverage point.

Note that as $\mathbf{P}_x$ only depends on $\mathbf{x}_1, \cdots, \mathbf{x}_n$, so is the leverage.

A leverage point is called a **good leverage point** if the corresponding y is close to $\widehat{y}$. It is called a **bad leverage point** if the corresponding y is an outlier.

For the food data set, $\frac{2(p+1)}{n} = \frac{4}{506} = 0.0079$. The figure shows the 215th, 413th and 375th observations are bad leverage points. We may consider to remove them from the analysis, since they have great influence on the fitted model.

We may try

```
lm(medv ~ lstat + age) # Use 2 regressors: lstat and age
lm(medv ~ lstat*age) # Use 3 regressors: lstat, age, and their product
lm(medv ~ lstat + I(lstat^2)) # I(lstat^2) represent lstat^2
lm(medv ~ poly(lstat, 5)) # using polynomial function of order 5
lm(medv ~ ., data=Boston)   # Using all variables in Boston as regressors
lm(medv ~.-age, data=Boston) # Using all but age
```

Let us try a few.

```
> lm2.Boston=lm(medv~., Boston)
> summary(lm2.Boston)
Call:
lm(formula = medv ~ ., data = Boston)
Residuals:
    Min      1Q  Median      3Q     Max
-15.595  -2.730  -0.518   1.777  26.199
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
zn           4.642e-02  1.373e-02   3.382 0.000778 ***
indus        2.056e-02  6.150e-02   0.334 0.738288
chas         2.687e+00  8.616e-01   3.118 0.001925 **
nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
```

```
rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
age          6.922e-04  1.321e-02   0.052 0.958229
dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
black        9.312e-03  2.686e-03   3.467 0.000573 ***
lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,Adjusted R-squared:  0.7338
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Since age has the largest P-value (i.e. 0.958), we remove it from the model

```
> lm3.Boston=lm(medv~.-age, Boston)
> summary(lm3.Boston)
Call:
lm(formula = medv ~ . - age, data = Boston)
Residuals:
     Min       1Q   Median       3Q      Max
-15.6054  -2.7313  -0.5188   1.7601  26.2243
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.436927   5.080119   7.172 2.72e-12 ***
crim         -0.108006   0.032832  -3.290 0.001075 **
```

```
zn            0.046334   0.013613   3.404 0.000719 ***
indus         0.020562   0.061433   0.335 0.737989
chas          2.689026   0.859598   3.128 0.001863 **
nox         -17.713540   3.679308  -4.814 1.97e-06 ***
rm            3.814394   0.408480   9.338  < 2e-16 ***
dis          -1.478612   0.190611  -7.757 5.03e-14 ***
rad           0.305786   0.066089   4.627 4.75e-06 ***
tax          -0.012329   0.003755  -3.283 0.001099 **
ptratio      -0.952211   0.130294  -7.308 1.10e-12 ***
black         0.009321   0.002678   3.481 0.000544 ***
lstat        -0.523852   0.047625 -10.999  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4.74 on 493 degrees of freedom
Multiple R-squared:  0.7406,Adjusted R-squared:  0.7343
F-statistic: 117.3 on 12 and 493 DF,  p-value: < 2.2e-16
```

As indus is not significant, we remove it now

```
> lm4.Boston = update(lm3.Boston, ~.-indus)
> summary(lm4.Boston)
Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
    tax + ptratio + black + lstat, data = Boston)
Residuals:
```

```
     Min       1Q   Median       3Q      Max
-15.5984  -2.7386  -0.5046   1.7273  26.2373
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
crim         -0.108413   0.032779  -3.307 0.001010 **
zn            0.045845   0.013523   3.390 0.000754 ***
chas          2.718716   0.854240   3.183 0.001551 **
nox         -17.376023   3.535243  -4.915 1.21e-06 ***
rm            3.801579   0.406316   9.356  < 2e-16 ***
dis          -1.492711   0.185731  -8.037 6.84e-15 ***
rad           0.299608   0.063402   4.726 3.00e-06 ***
tax          -0.011778   0.003372  -3.493 0.000521 ***
ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
black         0.009291   0.002674   3.475 0.000557 ***
lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 4.736 on 494 degrees of freedom
Multiple R-squared:  0.7406,Adjusted R-squared:  0.7348
F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

Although we have removed two variables `age, indus`, the regression correlation coefficient is unchanged, as $R^2 = 0.7406$ always. But the adjusted correlation coefficient increases slightly.

We can also fit the data using the stepwise procedure to select variables. R function `step` implements this selection method using the AIC criterion. To use `step`, we need to specify a maximum model, which is `ls2.Boston` including all the variables, and a minimum model which may include intercept term only:

```
> lm0.Boston=lm(medv~1)
> lm0.Boston
Call:
lm(formula = medv ~ 1)

Coefficients:
(Intercept)
      22.53
```

The selected model will be between `ls0.Boston` and `ls2.Boston`.

Now we are ready to call for stepwise selection:

```
> step.Boston=step(lm0.Boston, scope=list(upper=lm2.Boston))
> summary(step.Boston)
```

```
Call:
lm(formula = medv ~ lstat + rm + ptratio + dis + nox + chas +
    black + zn + crim + rad + tax)

Residuals:
     Min       1Q   Median       3Q      Max
-15.5984  -2.7386  -0.5046   1.7273  26.2373

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
rm            3.801579   0.406316   9.356  < 2e-16 ***
ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
dis          -1.492711   0.185731  -8.037 6.84e-15 ***
nox         -17.376023   3.535243  -4.915 1.21e-06 ***
chas          2.718716   0.854240   3.183 0.001551 **
black         0.009291   0.002674   3.475 0.000557 ***
zn            0.045845   0.013523   3.390 0.000754 ***
crim         -0.108413   0.032779  -3.307 0.001010 **
rad           0.299608   0.063402   4.726 3.00e-06 ***
tax          -0.011778   0.003372  -3.493 0.000521 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.736 on 494 degrees of freedom
Multiple R-squared:  0.7406,Adjusted R-squared:  0.7348
F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

The final selected model is actually `ls4.Boston`.

Note. R prints out on the screen the whole process of how this model was derived from the initial `ls0.Boston` by adding and deleting variables step by step.

Regression with qualitative Predictors

We use the data `Carseats` in ISLR as an illustration.

```
> summary(Carseats)
     Sales          CompPrice       Income        Advertising       Population
 Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000   Min.   : 10.0
 1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000   1st Qu.:139.0
 Median : 7.490   Median :125   Median : 69.00   Median : 5.000   Median :272.0
 Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635   Mean   :264.8
 3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000   3rd Qu.:398.5
 Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000   Max.   :509.0
    Price       ShelveLoc        Age          Education    Urban       US
Min.   : 24.0  Bad   : 96   Min.   :25.00   Min.   :10.0   No :118   No :142
1st Qu.:100.0  Good  : 85   1st Qu.:39.75   1st Qu.:12.0   Yes:282   Yes:258
Median :117.0  Medium:219   Median :54.50   Median :14.0
Mean   :115.8               Mean   :53.32   Mean   :13.9
3rd Qu.:131.0               3rd Qu.:66.00   3rd Qu.:16.0
Max.   :191.0               Max.   :80.00   Max.   :18.0
```

The 3 qualitative variables: `Urban` and `US` are binary, and `ShelveLoc` takes 3 values.

We add interaction terms `Income:Advertising` and `Price:Age` in the model.

```
> lm.Sales=lm(Sales~.+Income:Advertising+Price:Age, data=Carseats)
> summary(lm.Sales)

Call:
lm(formula = Sales ~ . + Income:Advertising + Price:Age, data = Carseats)

Residuals:
    Min      1Q  Median      3Q     Max
-2.9208 -0.7503  0.0177  0.6754  3.3413

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.5755654  1.0087470   6.519 2.22e-10 ***
CompPrice           0.0929371  0.0041183  22.567  < 2e-16 ***
Income              0.0108940  0.0026044   4.183 3.57e-05 ***
Advertising         0.0702462  0.0226091   3.107 0.002030 **
Population          0.0001592  0.0003679   0.433 0.665330
Price              -0.1008064  0.0074399 -13.549  < 2e-16 ***
ShelveLocGood       4.8486762  0.1528378  31.724  < 2e-16 ***
ShelveLocMedium     1.9532620  0.1257682  15.531  < 2e-16 ***
Age                -0.0579466  0.0159506  -3.633 0.000318 ***
Education          -0.0208525  0.0196131  -1.063 0.288361
```

```
UrbanYes            0.1401597  0.1124019   1.247 0.213171
USYes              -0.1575571  0.1489234  -1.058 0.290729
Income:Advertising  0.0007510  0.0002784   2.698 0.007290 **
Price:Age           0.0001068  0.0001333   0.801 0.423812
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.011 on 386 degrees of freedom
Multiple R-squared:  0.8761,Adjusted R-squared:  0.8719
F-statistic:   210 on 13 and 386 DF,  p-value: < 2.2e-16
```

For binary variables `Urban` and `US`, R creates dummy variables `UrbanYes`, `USYes`.

For `ShelveLoc` with 3 values, two dummies `ShelveLocGood` and `ShelveLocMedi` are created. To check their definition,

```
> attach(Carseats}
> contrasts(ShelveLoc)
       Good Medium
Bad       0      0
Good      1      0
Medium    0      1
```

i.e. `ShelveLocGood` takes value 1 if `ShelveLoc=Good`, and 0 otherwise, and `ShelveLocMedium` takes value 1 if `ShelveLoc=Medium`, and 0 otherwise.

Both `ShelveLocGood` and `ShelveLocMedium` are significant in the fitted model with coefficient 4.849 and 1.953 respectively, indicating that a medium shelving location leads to higher sales than a bad shelving location but lower sales than a good shelving location.

Regression trees

Linear regression specifies an explicit model, which is *linear* in coefficients, for the regression function. It works well when the model is about correct.

When the true function is highly nonlinear, a tree model may provide a valid alternative.

A regression free:

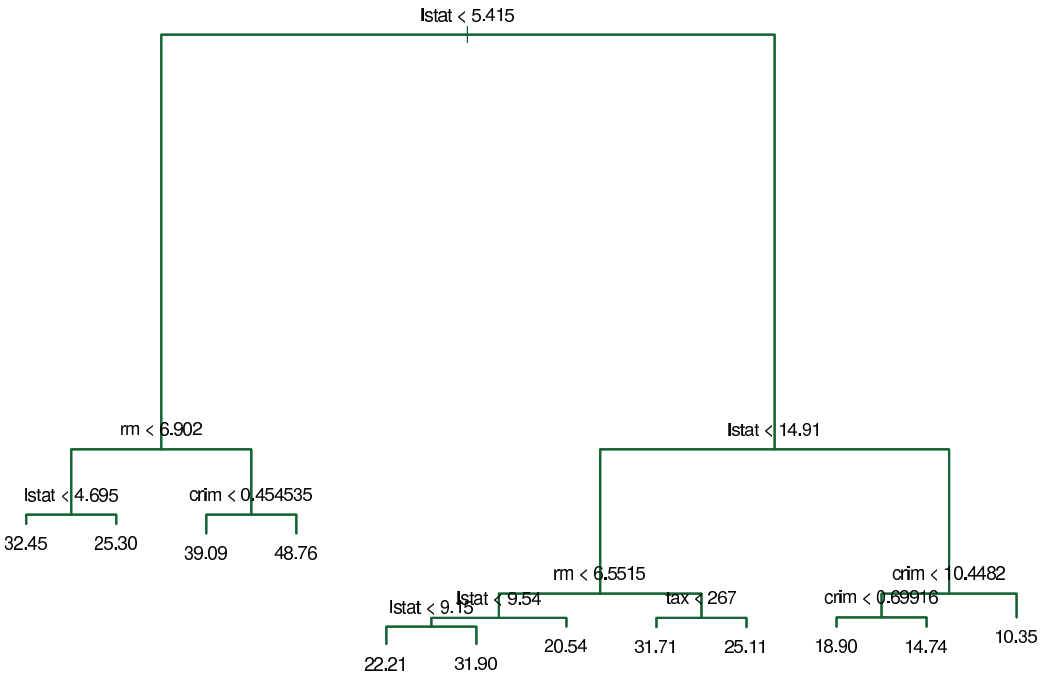
$$\widehat{Y} = \sum_{i=1}^{M} c_i I(\mathbf{X} \in R_i),$$

where $R_1, \cdots, R_M$ form a partition of the feature space (i.e. the $\mathbf{X}$-space) and

$$c_i = \sum_{j=1}^{M} y_j I(\mathbf{x}_j \in R_i) \Big/ \sum_{j=1}^{M} I(\mathbf{x}_j \in R_i).$$

Similar to growing a decision tree, we use recursive binary splitting to grow a regression tree. But each time we search for the splitting which maximises the reduction of RSS$= \sum_i (y_i - \widehat{y}_i)^2$. We stop when, for example, each terminal node has fewer than k_0 observations, where k_0 is a prespecified integer.

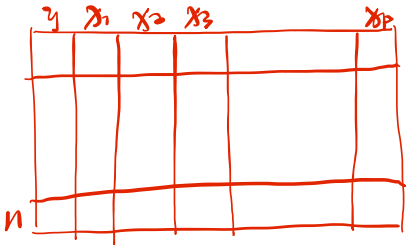
```
> library(MASS); library(tree)
> train=sample(1:nrow(Boston), nrow(Boston)/2)
> tree.Boston=tree(medv~., data=Boston, subset=train)
> summary(tree.Boston)

Regression tree:
tree(formula = medv ~ ., data = Boston, subset = train)
Variables actually used in tree construction:
[1] "lstat" "rm"    "crim"  "tax"
Number of terminal nodes:  12
Residual mean deviance:  12.79 = 3082 / 241 # Mean RSS for regression
Distribution of residuals:
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-11.0000  -1.8060  -0.3394   0.0000   1.8860  18.1000
> plot(tree.Boston, col="blue")
> text(tree.Boston, pretty=0)
```



The fitting entails the mean RSS 12.79 for the training data. Let us check how it performs on testing data

```
> medv.test=Boston[-train, "medv"]
> medv.predict=predict(tree.Boston, newdata=Boston[-train,])
> mean((medv.predict-medv.test)^2)
[1] 30.14707
```

The mean squares of predictive errors is 30.15 for the testing sample, which is greater than 12.79 for the training sample.



Bagging: a bootstrap aggregation method

A fitted tree suffers from *high variance*, i.e. the tree depends on training data sensitively.

Basic idea to reduce variance: average a set of observations.

For example, the variance of a sample mean, from a sample of size n, is reduced from σ^2 to σ^2/n.

If we had many fitted trees, the "mean" of those trees would have a smaller variance. But in practice we only have one sample.

Create B sets of training data by bootstrapping from the original data set. For each bootstrap sample, fitting a tree. Those trees are grown deep, and are not pruned. Hence each individual tree has high variance but low bias.

'Averaging' those B trees reduces the variance.

For decision trees, 'averaging' means taking the majority votes of the B trees.

Bootstrap in R: Let `X` be a vector containing n observation. Then a bootstrap sample is obtained as follows

```
> Xstar = sample(X, n, replace=T)
```

Note. Bagging improves prediction accuracy at the expense of interpretability, as the final result is an average of many different trees.

Importance: The importance of each predictor can be measured by the average reduction of RSS (for regression trees), or average information gain (for decision trees) over different trees – The larger the better!

→ Better than Bagging

Random forests: an improvement of Bagging by decorrelating the trees.

☆ X boosting : more popular

Similar to Bagging, we build a tree for each bootstrap sample. Differently from Bagging, at each step we split the feature space by searching the split from m, instead of all p, predictors, where $m \leq p$. Furthermore, we use a different and randomly selected subsets of m predictors for each split.

Typical choice: $m = \sqrt{p}$.

When $m = p$, it reduces to Bagging.

Why is better: the trees in the forest are less correlated than those in the bagging.

For example, suppose there is a dominate predictor which is likely to enter all or most trees if $m = p$. This would make the fitted tree highly correlated.

```
> install.packages("randomForest")
> library(randomForest)
> bag.Boston=randomForest(medv~., data=Boston, subset=train,
    mtry=13, importance=T)  # mtry=13 sets m=p
> bag.Boston
Call:
 randomForest(formula = medv ~ ., data = Boston, mtry = 13,
              importance = T, subset = train)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 13
          Mean of squared residuals: 14.48475
                    % Var explained: 82.31
```

The option `mtry=13` demands to search over all $p = 13$ variables in each split, and therefore, it is a Bagging fitting. The mean RSS is 14.48, and $R^2 = 82.31\%$.

```
> medv.predict=predict(bag.Boston, newdata=Boston[-train,])
> mean((medv.predict-medv.test)^2)
[1] 12.83074
```

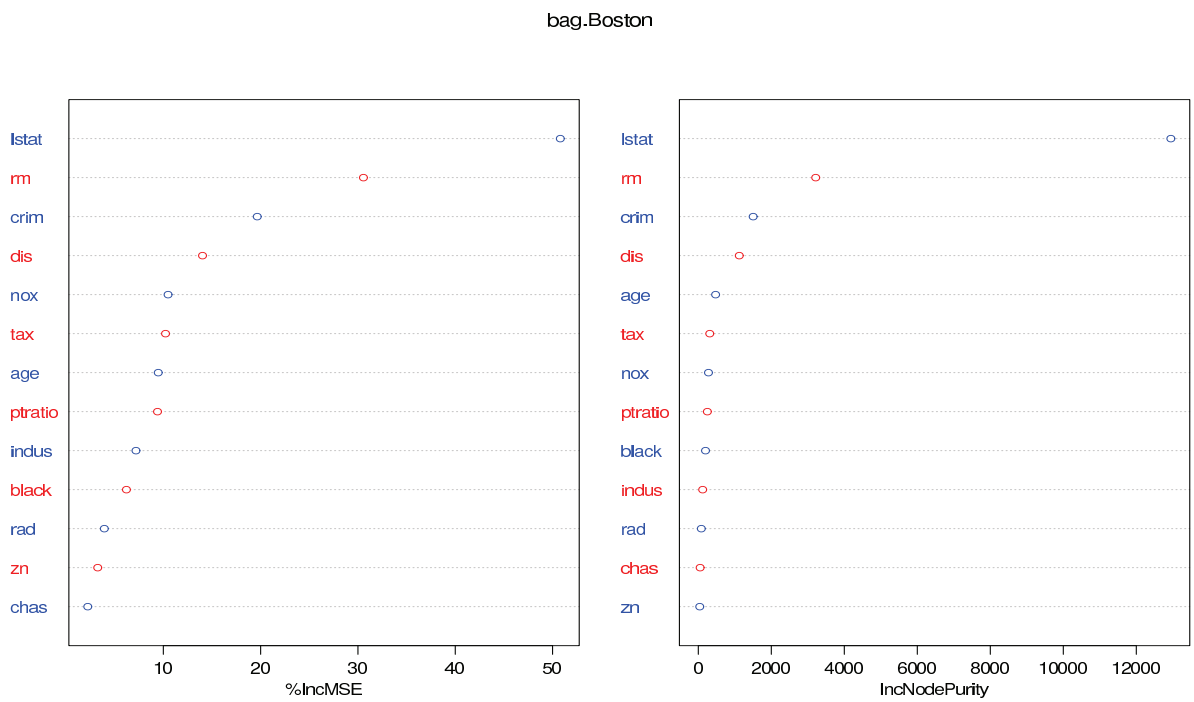
The mean squares of predictive errors for the test sample from this Bagging fitting is 12.83.

```
> importance(bag.Boston)
          %IncMSE IncNodePurity
crim    19.651982    1503.30977
zn       3.258542      39.40467
indus    7.189510     119.63707
chas     2.234998      50.37624
nox     10.487697     278.67558
rm      30.564156    3216.50360
age      9.478481     473.88776
dis     14.028146    1122.75188
rad      3.936966      82.59776
tax     10.225864     313.85307
ptratio  9.402331     248.13892
black    6.207921     197.53684
lstat   50.795822   12946.76696
```

Two measures of importance are reported: The former is based on the mean decrease of accuracy in prediction on the 'out of bag' sample when the predictor is excluded from the model. The latter is measure of the total decrease in node impurity that results from splits over the variable, averaged over all trees.

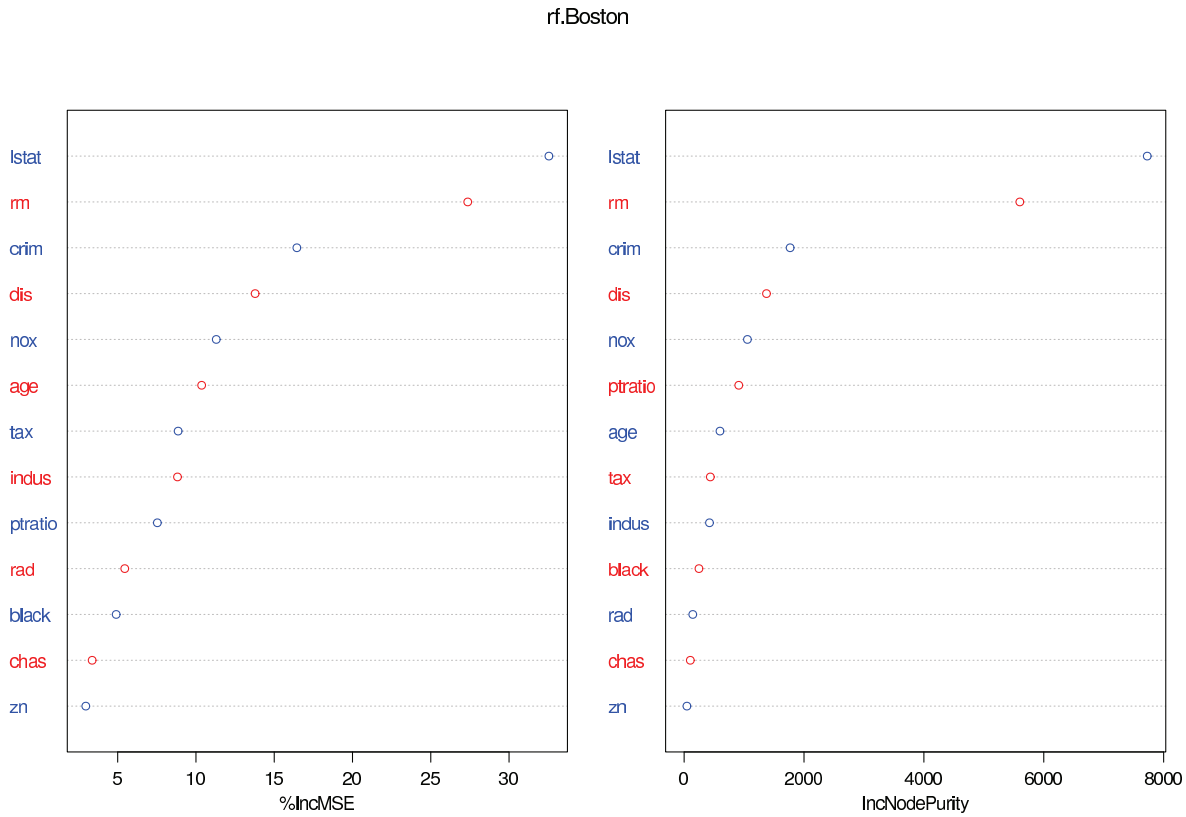
For this example, `lstat` (the wealth level of the community) is the most important predictor, followed by rm (house size).

```
> varImpPlot(bag.Boston, col=c("blue","red")) # Importance measure plot
```



```
> rf.Boston=randomForest(medv~., data=Boston, subset=train,
                  mtry=6, importance=T)  # 'mtry=6' sets m=6<p
> rf.Boston
Call:
 randomForest(formula = medv ~ ., data = Boston, mtry = 6,
               importance = T, subset = train)
              Type of random forest: regression
                    Number of trees: 500
No. of variables tried at each split: 6

          Mean of squared residuals: 13.69494
                    % Var explained: 83.27
> medv.predict=predict(rf.Boston, newdata=Boston[-train,])
> mean((medv.predict-medv.test)^2)
[1] 12.18501
> varImpPlot(rf.Boston, col=c("blue","red"))
```

Mean squares of predictive errors for the testing sample from this random forest model is 12.18501, smaller than that from the Bagging model.



Boosting → Always update y, avoid some specific variables

Like Bagging, boosting can be applied to many learning methods for regression and classification. We use regression trees as an illustration.

Boosting algorithm for regression trees

1. Set $\widehat{f}(x) = 0$ and $r_i = y_i$ for all i.
2. For $b = 1, \cdots, B$, repeat:
   (a) Fit a tree $\widehat{f}^b$ with d split ($d+1$ terminal nodes) to the training data $\{x_i, r_i\}$.
   (b) Update $\widehat{f} = \widehat{f} + \lambda \widehat{f}^b$.
   (c) Update residuals $r_i = r_i - \lambda \widehat{f}^b(x_i)$.
3. Output the boosted model $\widehat{f}(x) = \sum_{b=1}^{B} \widehat{f}^b(x)$.

3 tuning parameters:

B is a large integer, too large B leads to overfitting. It can be selected by cross-validation.

d is a small integer, typically taking values 1 or 2.

$\lambda \in (0, 1)$, is typically small such as 0.01 or 0.001. Smaller λ requires larger B.

It is known that *fitting the data hard* may lead to overfitting. Boosting is in the spirit of *learning slowly*: Given the current model, we fit a small tree (as d is small) to the residuals from the model. By fitting small trees to the residuals, we slowly improve $\widehat{f}$ in areas where it does not perform well. The shrinkage parameter λ slows the process down even further, allowing more and different shaped trees to attack the residuals.

Note that in boosting, unlike in Bagging, the construction of each tree depends strongly on the trees that have already been grown.

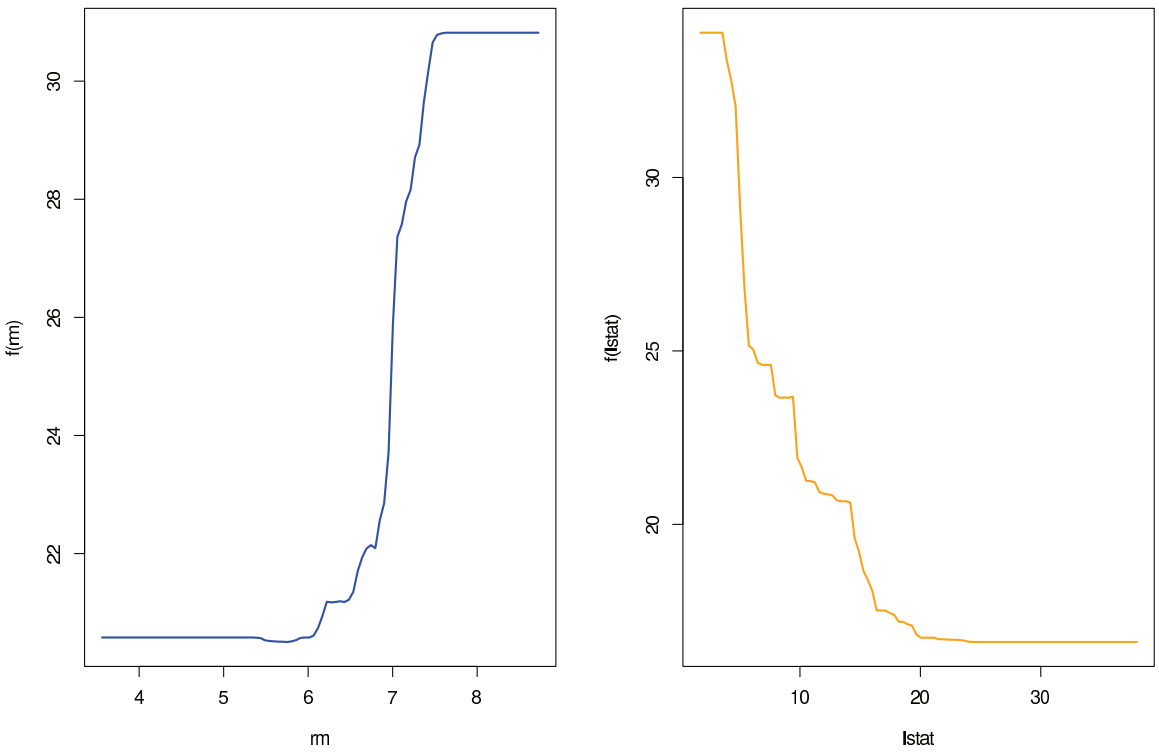
```
> install.packages("gbm") # gbm: Generalized boosted model
> library(gbm)
> boost.Boston=gbm(medv~., data=Boston[train,], n.trees=5000,
      interaction.depth = 4) # B=n.trees, default value lambda=0.001
     # d=interaction.depth
> summary(boost.Boston)
            var    rel.inf
lstat     lstat 51.6849642
rm           rm 23.1720615
crim       crim  7.9381089
```

```
dis          dis  6.4563409
nox          nox  2.4229547
age          age  2.2533567
ptratio ptratio  2.2168609
black      black  1.0393828
tax          tax  0.8387594
indus      indus  0.6450746
chas        chas  0.6439905
rad          rad  0.5473836
zn            zn  0.1407613
```

The summary provides the relative inference statistics: `lstat` and `rm` are by far the most important variables.

We can also produce partial dependence plots for these two variables. These plots illustrate the marginal effect of the selected variables on the response after integrating out the other variables. In this case, as we might expect, median house prices are increasing with `rm` and decreasing with `lstat`.

```
> par(mfrow=c(1,2))
> plot(boost.Boston, i="rm", lwd=2, col="blue")
> plot(boost.Boston, i="lstat", lwd=2, col="orange")
```



To compute the mean squares of predictive errors for the test sample:

```
> medv.predict=predict(boost.Boston, newdata=Boston[-train,],
          n.trees=5000)
> mean((medv.predict-medv.test)^2)
[1] 13.896
```

One can, for example, set $\lambda = 0.2$:

```
> boost.Boston=gbm(medv~., data=Boston[train,], n.trees=5000,
                interaction.depth =4, shrinkage =0.2)
```

From global fitting to local fitting — an illustration by example

Consider linear regression model

$$Y = X_1\beta_1 + \cdots + X_d\beta_d + \varepsilon, \tag{1}$$

where $\varepsilon \sim (0, \sigma^2)$.

Put $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_d)^\tau$

With observations $\{(y_i, \mathbf{x}_i),\ 1 \le i \le n\}$, where $\mathbf{x}_i = (x_{i1}, \cdots, x_{id})^\tau$, the LSE minimizes

$$\sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^\tau \boldsymbol{\beta}\right)^2, \tag{2}$$

resulting to

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}^\tau \mathbf{X})^{-1} \mathbf{X}^\tau \mathbf{y},$$

where $\mathbf{y} = (y_1, \cdots, y_n)^\tau$, and $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^\tau$ is an $n \times d$ matrix.

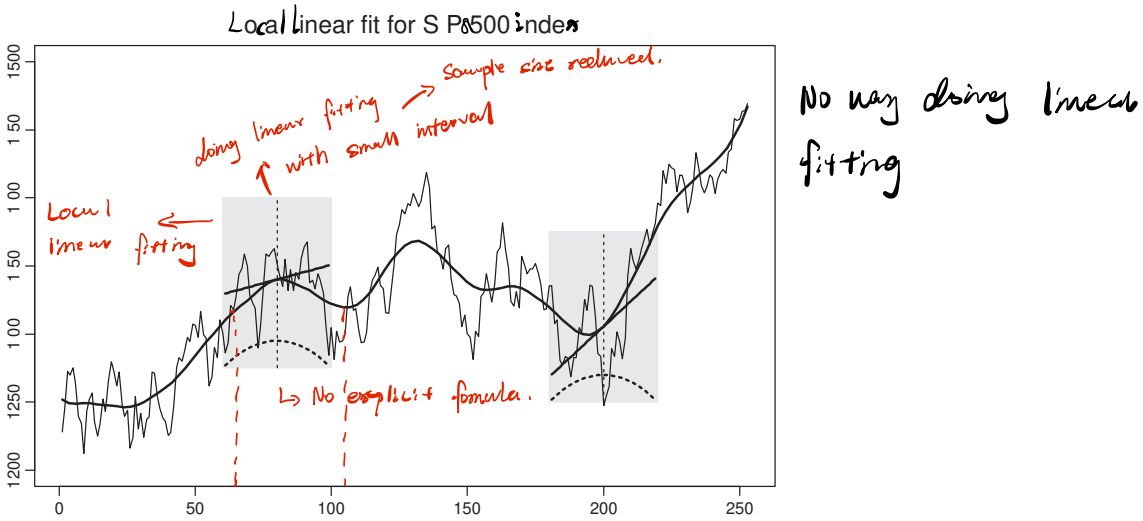
The fitted model is

$$\widehat{\mathbf{y}} = \mathbf{X}\widehat{\boldsymbol{\beta}}.$$

This is a **global** fitting, since the model is assumed to be true everywhere in the sample space and the estimator $\widehat{\boldsymbol{\beta}}$ is obtained using all the available data.

Such a global fitting is efficient **if** the assumed form of the regression function (1) is correct.

In general (1) may be incorrect globally. But it may provide a reasonable approximation at any small area in the sample space. We fit for each given small area a different linear model — This is the basic idea of **local** fitting.



Local linear fit for the S&P 500 Index from January 4, 1999 to December 31, 1999, using the Epanechnikov kernel and bandwidth $h = 20$. The dashed parabola in each window indicates the weight that each local data point receives.

Local fitting – Non-parametric models: little assumption on the underlying model, but estimation is less efficient.

Serious drawback for multivariate model: curse of dimensionality

Semi-parametric models: mitigate the curse of dimensionality

- partial linear models
- index models
- additive models (R package `gam`)
- varying-coefficient linear models

Does a tree model provide a local fitting?

Mincer Equation: How is one's earning related to human capital?

$$\log(Y) = \beta_0 + \beta_1 X + \beta_2 U + \beta_3 U^2 + \varepsilon,$$

- Y — earning
- X — education capital: No. of years in school/university
- U — experience capital: No. of years in employment

Rate of return to education: β_1

This is a simple linear regression model.

Drawbacks: no interaction between X and U

Extended Mincer Equation:

$$\begin{aligned}\log(Y) &= \beta_0 + \beta_{11}X + \beta_{12}UX + \beta_{13}U^2X + \beta_2 U + \beta_2 U^2 + \varepsilon \\ &= (\beta_0 + \beta_2 U + \beta_2 U^2) + (\beta_{11} + \beta_{12}U + \beta_{13}U^2)X + \varepsilon \\ &= g_0(U) + g_1(U)X + \varepsilon.\end{aligned}$$

If we see $g_0(\cdot)$ and $g_1(\cdot)$ as coefficient functions, this is varying-coefficient linear model.

In general, we do not restrict the coefficients as quadratic functions: local fitting.

Rate function of return to education: $g_1(\cdot)$

Example. (Wang and Yue 2008). Survey data on annual incomes and human capitals of Chinese citizen in 1991, 1995, 2000 and 2004.

Fitting a linear model:

$$\log(Y) = \beta_0 + \beta_1 X + \beta_2 U + \beta_3 U^2 + \eta_1 Z_1 + \eta_2 Z_2 + \varepsilon$$

where

Y: total annual income
X: no. of years in school/university
U: no. of years in employment
$Z_1 = 1$ – female, $Z_1 = 0$ – male
$Z_2 = 1$ – eastern China, $Z_2 = 0$ – central/western China

Hence

η_1: difference in log(income) between female and male
η_2: difference in log(income) between the Eastern and the rest of China

	1991	1995	2000	2004
$\widehat{\beta}_0$	6.823	7.647	7.410	7.747
$\widehat{\beta}_1$	0.028	0.045	0.086	0.105
$\widehat{\beta}_2$	0.054	0.026	0.034	0.024
$\widehat{\beta}_3$	-0.0001	-0.0002	-0.0004	-0.0002
$\widehat{\gamma}_1$	-0.095		-0.171	-0.272
$\widehat{\gamma}_2$	0.195	0.293	0.260	0.362

Note. The quality of the data for 1995 is poor: no gender, censored to the minimum 1003 yuans etc.

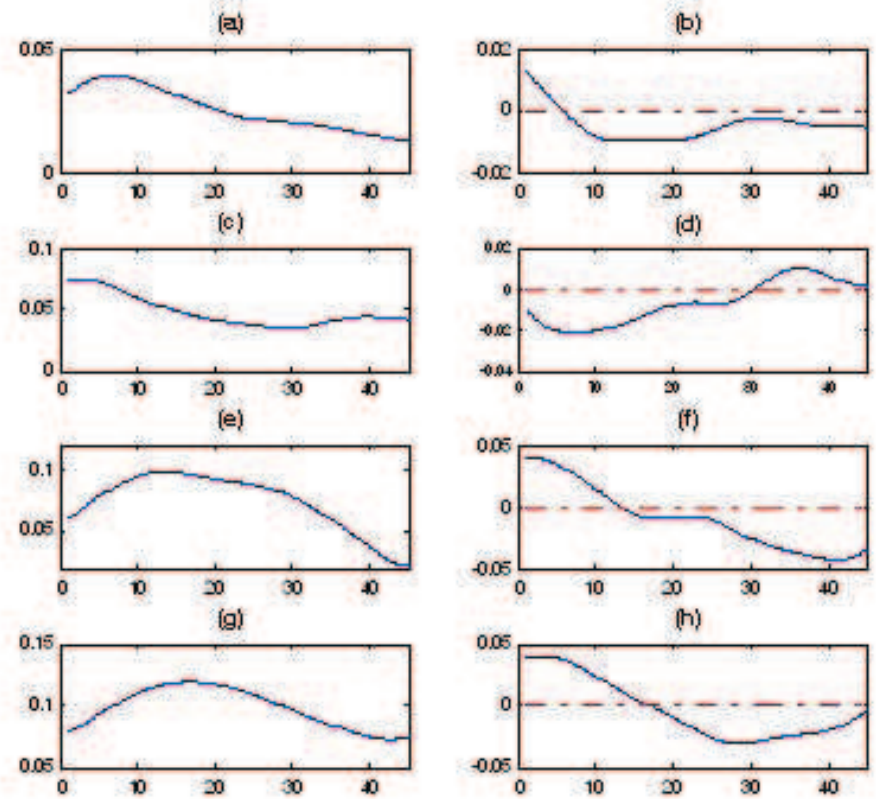
Fitting a varying-coefficient linear model:

$$\log(Y) = g_0(U) + g_1(U)X + \eta_1 Z_1 + \eta_2 Z_2 + \varepsilon$$

where

Y: total annual income
X: no. of years in school/university
U: no. of years in employment
$Z_1 = 1$ – female, $Z_1 = 0$ – male
$Z_2 = 1$ – eastern China, $Z_2 = 0$ – central/western China

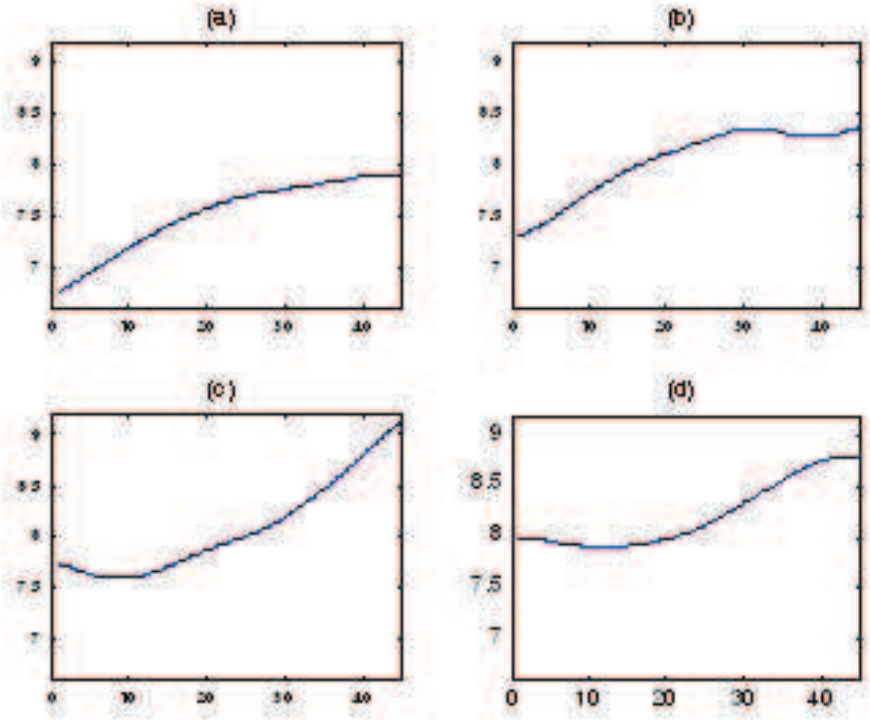
The estimated values for η_1 and η_2 hardly changed.



Estimated $g_1(U)$ for (a) 1991, (c) 1995, (e) 2000 & (g) 2004, and their derivatives.

In 1991, g_1 = max when $U = 5.8$
In 2000, g_1 = max when $U = 13.5$
In 2004, g_1 = max when $U = 16.9$

Those started to work around 1985 – 1987 always have the maximum returns out of education.



Estimated $g_0(U)$ for (a) 1991, (b) 1995, (c) 2000 & (d) 2004.

$g_0(U)$ increases initially, then gradually flat out after about 20-30 years when experience saturated.

Chapter 5. Overfitting and Its Avoidance

"*If you torture the data long enough, it will confess*" – Nobel Laureate Ronald Coase

- *Basic concepts*: Fitting and overfitting, complexity control, generalization
- *Exemplary techniques*: Cross-validation, variable selection, tree pruning, regularization

Further readings:

James et al. (2013) Sections 5.1, 5.3.1-5.3.3, 6.1-6.2 & 6.5-6.6,

Provost and Fawcett (2013) Chapter 5.

If we allows ourselves enough flexibility in searching for patterns in a data set, we will find pattern. Unfortunately those 'patterns' may be just chance occurrences in the data. They do not represent systematic characteristics of the underlying system.

Overfitting: a model is tailored too much to the training data at the expense of generalization to previously unseen data point.

Overfitting is associated with model complexity: more complex a model is, more likely it is overfitting.

A extreme case of overfitting is the so-called *table model* which records all the training data exactly. A table model performs typically poorly in *generalization*, i.e. it often predict a new value badly.

A tree table model: Grow tree by keeping splitting on variables until there is a single point at each leaf node.

A regression table model: Fitting $y = \beta_0 + \beta_1 x + \cdots + \beta_{n-1} x^{n-1}$ with to the data $(y_i, x_i), i = 1, \cdots, n$ results zero residuals.

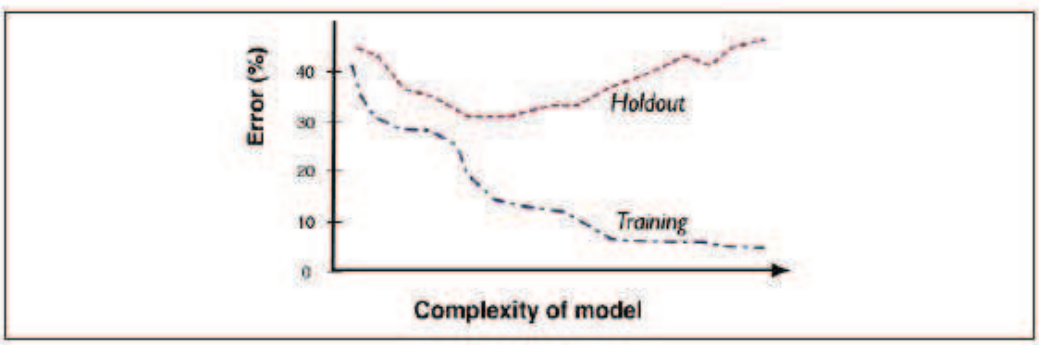
In general data are regarded containing signal with noise. A good model should catch signal but leave out the noise. An overfitting takes also noise as a part of signal.

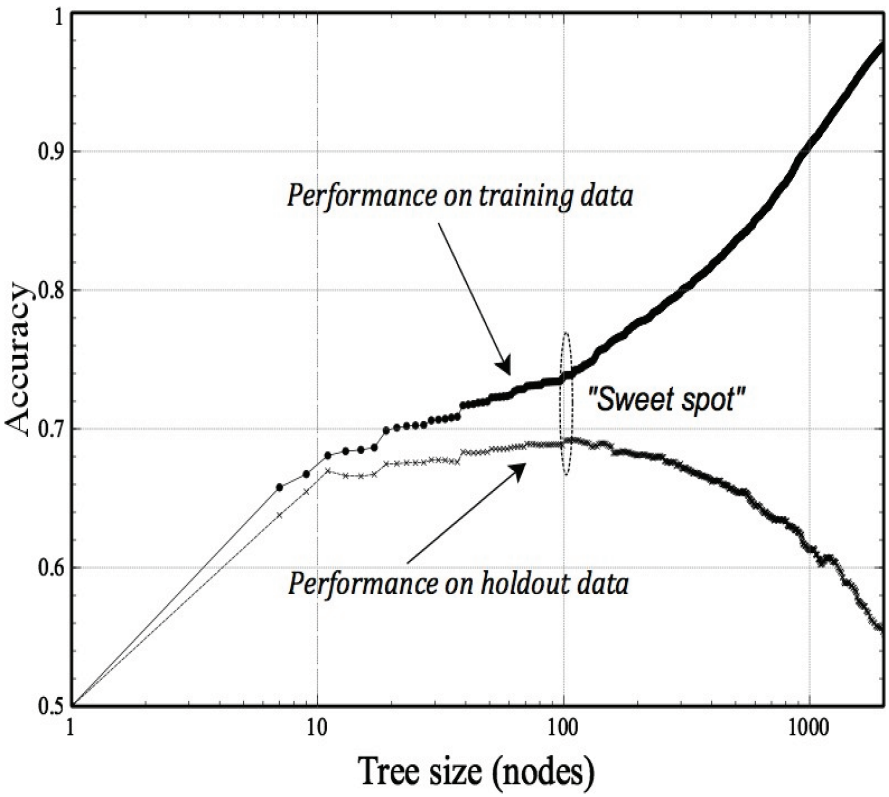
Difficulty: how much noise in data is unknown.

Check for overfitting: check the performance of a fitted model on the data not used in fitting.

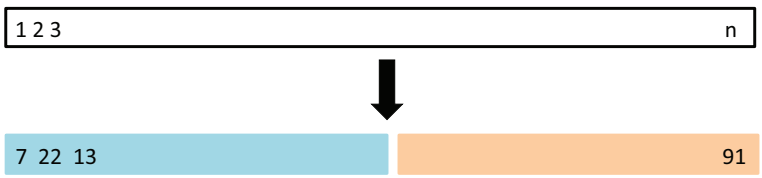
- Learning (training) data: data used in fitting

- Validation (holdout) data: data not used in fitting, held for testing the performance of a fitted model





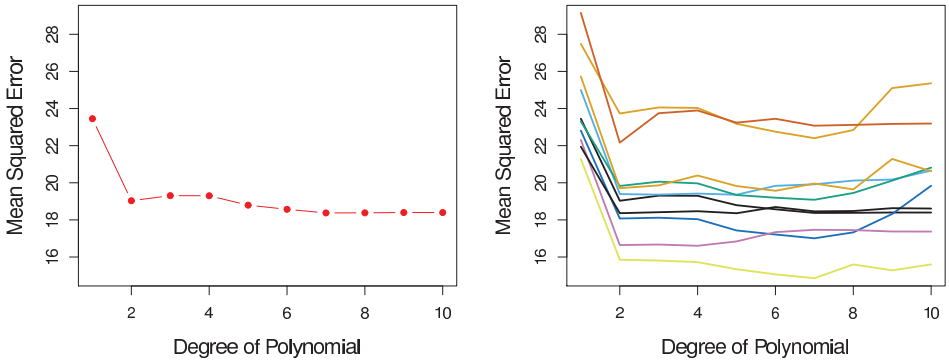
Holdout Evaluation. Divide the available data into two parts: training data used to fit a model, and holdout data for validation.



Two drawbacks:

- The usage of the data is not efficient: both training and validation data sets are smaller than the available data.

- The validation error depends on the particular split of validation set and training set.



The holdout evaluation is applied to `Auto.txt` with models

$$\texttt{mpg} = \beta_0 + \beta_1 \texttt{hpower} + \cdots + \beta_p \texttt{hpower}^p + \varepsilon$$

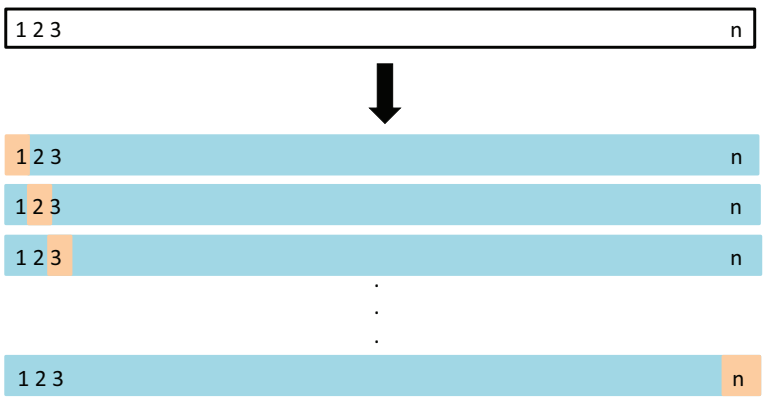
for $p = 1, \cdots, 10$. MSE is the average $(\widehat{y} - y)^2$ for y over a holdout set, and $\widehat{y}$ is the predicted value of y based on the model estimated using the training set.

Left panel: Plot of MSE vs p for one split between training and holdout sets.

Right panel: for 10 different splits.

Cross-validation (CV) – choosing model complexity by minimizing validation errors

Leave-one-out approach: each time leave one data point out, fit the model using all the other data, calculate the error on the point left out. Repeat this process for each data point. The best model should minimize the accumulative errors.



Computationally intensive, but more efficient use of the data. The MSE is calculated as

$$CV(w) = \frac{1}{n}\sum_{j=1}^{n}(y_j - \widehat{y}_j(w))^2$$

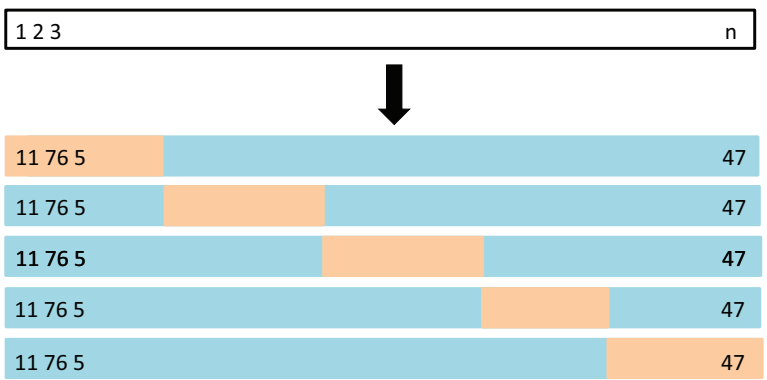
where $\widehat{y}_j$ is the predicted value for y_j based on the estimated model with the complexity indexed by w using the other $(n-1)$ observations.

The CV selected model should have the complexity $\widehat{w}$ which minimizes $CV(w)$.

Complexity w: the number of regressors in a regression model, or the number of terminal nodes of a tree.

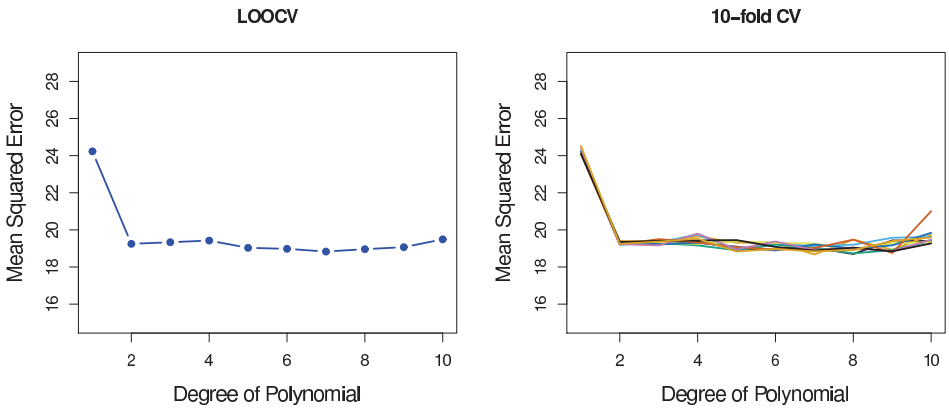
How to measure the complexity of a fitted model using the K nearest neighbours?

k-fold approach: randomly divide data into k groups (or folds) of equal size. Leave out one group each as a testing sample. Repeat this process k times.



A version of leave-$\frac{n}{k}$-out approach

The leave-one-out CV is unique while k-fold CV is not, as the way of dividing the whole data into k groups is not unique.



Cross validation is applied to `Auto.txt` with models

$$\texttt{mpg} = \beta_0 + \beta_1 \texttt{hpower} + \cdots + \beta_p \texttt{hpower}^p + \varepsilon$$

for $p = 1, \cdots, 10$.

Left panel: CV error curve: plot of CV-MSE vs p for leave-one-out approach

Right panel: 10-fold CV was run 9 times, each with a different random split of the data into 10 parts.

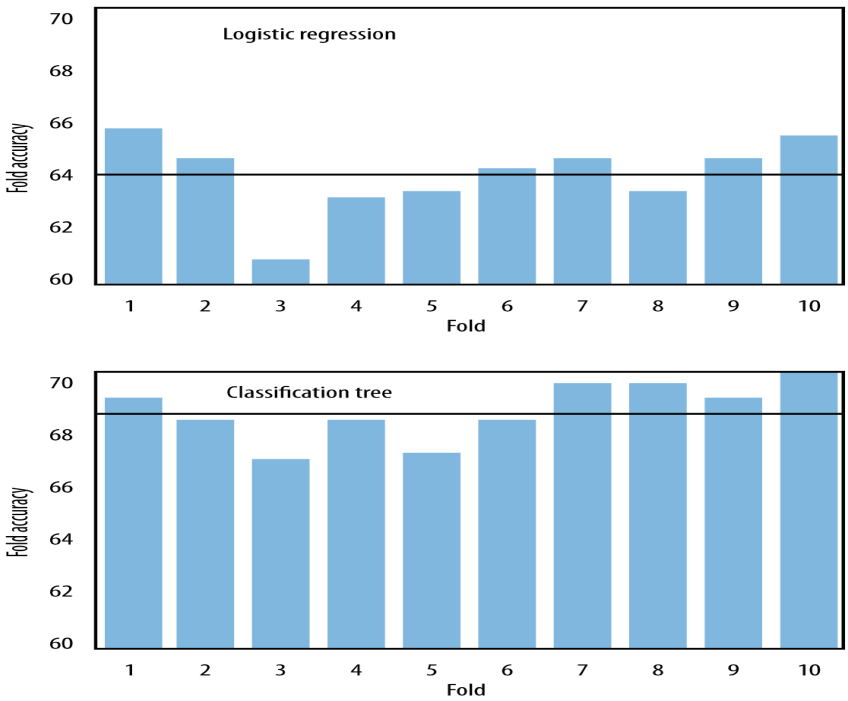
Churn example revisited

Section 4 derived a classification tree used the entire dataset for both training and testing with the reported 73% accuracy.

Now we run 10-fold cross-validation to select decision trees and logistic regression models: the whole data set is randomly divided into 10 folds. Each fold in turn served as a single holdout set while the other nine were collectively used for training.

The resulting classification: the majority votes of the 10 models.

The horizontal line is the average of accuracies of the 10 models in each panel.



- The average accuracy of the 10 folds with classification trees is 68.6%; significantly lower than 73%, 68.6% is a more realistic indicator for the accuracy when the method applies in the real world

- For this particular dataset, tree method performs better than logistic regression (with the average accuracy 64.1%). Also the accuracy variation of the tree method over the 10 folds is smaller or much smaller: the standard deviations are 1.1% (for trees) and 1.3% (for logistic regression).

- The performance fluctuations of the two methods show the similar pattern: worst for Fold 3 and best for Fold 10.

Minimize expected MSE: An alternative approach to the (computationally intensive) validation methods

Consider linear regression:

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

There are p candidate regressors $\mathbf{X} = (X_1, \cdots, X_p)$.

The model complexity is measured by the number of regressors used in a fitted model, say, d, $0 \leq d \leq p$

For given d, let $\widehat{f}_d$ denote the optimal subset regression with d regressors.

Ideally we should choose d to minimize the theoretical mean squared error:

$$\text{MSE}(d) = E[\{Y - \widehat{f}_d(\mathbf{X})\}^2]$$

Since it is unknown, we use an unbiased estimator

$$C_p(d) = \frac{1}{n}\{\text{RSS}(d) + 2d\widehat{\sigma}^2\},$$

where $\widehat{\sigma}^2$ is the estimated variance for ε with the full model (i.e. p regressors).

C_p-criterion: choose the optimal subset regression with $\widehat{d}$ regressors which minimizes $C_p(d)$.

Note. C_p-criterion is not applicable when p is large in relation to n!

Rule of thumb: $p \leq n/3$ or $n/4$.

Akaike information criterion (AIC)

Let $f_d(\mathbf{z}, \boldsymbol{\theta}_d)$ be a family of likelihood with the complexity indexed by d. Suppose there are available both training data and validation data, both with the sample size n.

- estimate $\boldsymbol{\theta}_d$ by maximising $\sum_{\mathbf{z}} \log f_d(\mathbf{z}, \boldsymbol{\theta}_d)$, leading to MLE $\widehat{\boldsymbol{\theta}}_d$, where the sum is taken over all $\mathbf{z}$ in the training data set
- choosing d to maximize $\sum_{\mathbf{z}} \log f_d(\mathbf{z}, \widehat{\boldsymbol{\theta}}_d)$, where the sum is taken over all $\mathbf{z}$ in the validation data set.

In practice we only have one data set, $-\frac{2}{n}\sum_{\mathbf{z}} \log f_d(\mathbf{z}, \widehat{\boldsymbol{\theta}}_d)$ has the same asymptotic mean as

$$\text{AIC} = -\frac{2}{n}(\text{maximized log likelihood}) + \frac{2}{n}(\text{No. of estimated parameters})$$

For regression model with normal errors,

$$\text{AIC}(d) = \log(\hat{\sigma}_d^2) + 2d/n,$$

where $\hat{\sigma}_d^2$ is the MLE for σ^2 in the optimal subset regression with d regressors.

AIC: choose d which minimizes AIC(d).

Bayesian information criterion (BIC)

$$\text{BIC} = -\frac{2}{n}(\text{maximized log likelihood}) + \frac{\log n}{n}(\text{No. of estimated parameters})$$

For regression model with normal errors,

$$\text{BIC}(d) = \log(\hat{\sigma}_d^2) + (\log n)d/n,$$

where $\hat{\sigma}_d^2$ is the MLE for σ^2 in the optimal subset regression with d regressors.

BIC: choose d which minimizes BIC(d).

Note. AIC tends to overestimate the number of regressors, while BIC gives a consistent estimator for the true d (i.e. the estimator converges to the true d when $n \to \infty$ but d is fixed).

Control complexity

A general principle: A good statistical model should provide an adequate fit to the available data, and should be as simple as possible.

Therefore an optimum model can be defined as the trade-off between *the goodness-of-fit* and *complexity of model*, i.e. it minimizes

(Goodness of fit of the model) + (Penalty for model complexity)

Two terms move to opposite directions when model complexity increases: GOF $\searrow$, Penalty $\nearrow$.

C_p-criterion, AIC and BIC are all of this form.

A criterion to pruning decision trees: for a given penalty constant $\lambda > 0$, search for the tree which minimizes

$$F_{\text{obj}} \equiv (\text{misclassification rate}) + \lambda \times (\text{number of nodes})$$

Remark. (i) λ controls the size of the tree. It can be selected by, for example, cross-validation or multi-fold cross validation as follows:

> Choose a grid of λ values, and compute the cross-validation version of F_{obj} for each value of λ. We then select the value of λ for which the cross-validation version of F_{obj} is smallest. Finally, the model is re-fit using all of the available observations and the selected value of λ.

(ii) An alternative approach is to require that each (terminal) node contains at least k data points, where $k \geq 1$ is an integer which controls the size of the tree.

Why as prediction?

Ridge Regression: an L_2 shrinkage method

For a regression model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{pi} + \varepsilon_i, \quad i = 1, \cdots, p,$$

The ridge regression estimators for $\beta_0, \cdots, \beta_p$ are obtained by minimizing

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2,$$

where $\lambda > 0$ is a tuning parameter controlling the degree of shrinkage: the larger λ is the smaller $|\beta_j|$ are.

The advantage of ridge regression is measured by MSE(= Var + bias^2): increasing λ leads to decrease of variance and increase of bias. With appropriate values of λ, the MSE of ridge regression estimator can be smaller than that of the OLS.

Choosing λ: cross validation or multi-fold CV.

Why as model selection?

LASSO: L_1 shrinkage.

Ridge regression makes $|\beta_j|$ smaller while the resulting model typically contain all the p regressors.

LASSO changes the L_2 norm in the penalty to the L_1 norm:

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

LASSO shrinks small β_j to 0, hence it also serves a variable/feature selection procedure.

LARS algorithm: compute LASSO solution path for all possible values of λ.

Final remark. Even proper validation is performed, a fitted model can still perform poorly in practice. They are many possible reasons:

- Data used in fitting the model do not match well the actual use scenario.
- Non-stationarity: the world has changed since the data used in fitting the model were collected.
- All candidate models are inadequate, running into the problem due to *multiple comparisons*

No silver bullet or magic recipe to truly get 'the optimal' model unfortunately! Also look into the 2nd or 3rd 'best' model.

We illustrate the R-implementation of some methods discussed in this chapter using the dataset `Hitter` in *ISLR* library: records on 20 variables from 322 major league baseball players in 1986/7 season.

```
> library(ISLR)
> names(Hitters)
 [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"       "Walks"     "Years"
 [8] "CAtBat"    "CHits"     "CHmRun"    "CRuns"     "CRBI"      "CWalks"    "League
[15] "Division"  "PutOuts"   "Assists"   "Errors"    "Salary"    "NewLeague"
> dim(Hitters)
[1] 322  20
> sum(is.na(Hitters))
[1] 59  # there are 59 missing values. Also try '> is.na(Hitters)' directly
> Hitters1=na.omit(Hitters) # remove the rows with missing values
> dim(Hitters1)
[1] 263  20
```

Function `regsubsets` function (part of the `leaps` library) performs best subset selection by identifying the best model that contains a given number of predictors, where best is in the sense of minimum RSS.

```
> install.packages("leaps")
> library(leaps)
> subset.Hitter = regsubsets(Salary~., data=Hitters1)
> summary(subset.Hitter)
```

```
Subset selection object
Call: regsubsets.formula(Salary ~ ., data = Hitters1)
19 Variables  (and intercept)
1 subsets of each size up to 8
Selection Algorithm: exhaustive
         AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI CWalks
1  ( 1 ) " "   " "  " "   " "  " " " " " "   " "    " "   " "    " "   "*"  " "
2  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "    " "   " "    " "   "*"  " "
3  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "    " "   " "    " "   "*"  " "
4  ( 1 ) " "   "*"  " "   " "  " " " " " "   " "    " "   " "    " "   "*"  " "
5  ( 1 ) "*"   "*"  " "   " "  " " " " " "   " "    " "   " "    " "   "*"  " "
6  ( 1 ) "*"   "*"  " "   " "  " " " " "*"   " "    " "   " "    " "   "*"  " "
7  ( 1 ) " "   "*"  " "   " "  " " " " "*"   " "    "*"   "*"    "*"   " "  " "
8  ( 1 ) "*"   "*"  " "   " "  " " " " "*"   " "    " "   " "    "*"   "*"  " "
         DivisionW PutOuts Assists Errors NewLeagueN
1  ( 1 ) " "       " "     " "     " "    " "
2  ( 1 ) " "       " "     " "     " "    " "
3  ( 1 ) " "       "*"     " "     " "    " "
4  ( 1 ) "*"       "*"     " "     " "    " "
5  ( 1 ) "*"       "*"     " "     " "    " "
6  ( 1 ) "*"       "*"     " "     " "    " "
7  ( 1 ) "*"       "*"     " "     " "    " "
8  ( 1 ) "*"       "*"     " "     " "    " "
```

For example, the best model with 3 regressors selected `Hits`, `CRBI`, `PutOuts`.

```
> subset3.Hitters=lm(Salary~Hits+CRBI+PutOuts, Hitters1)
> summary(subset3.Hitters)

Call:
lm(formula = Salary ~ Hits + CRBI + PutOuts, data = Hitters1)

Residuals:
    Min      1Q  Median      3Q     Max
-856.54 -171.37  -24.87  103.22 2169.87

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -71.45922   55.20273  -1.294 0.196650
Hits          2.80382    0.49229   5.695 3.33e-08 ***
CRBI          0.68253    0.06584  10.366  < 2e-16 ***
PutOuts       0.27358    0.07778   3.517 0.000514 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 336.1 on 259 degrees of freedom
Multiple R-squared:  0.4514,Adjusted R-squared:  0.4451
F-statistic: 71.05 on 3 and 259 DF,  p-value: < 2.2e-16
```

To see the info in the output

```
> names(summary(subset.Hitter))
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
> summary(subset.Hitter)$rsq # regression coefficient R^2
[1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146 0.5141227 0.5285569
> summary(subset.Hitter)$bic  # BIC values for 8 selected models
[1]  -90.84637 -128.92622 -135.62693 -141.80892 -144.07143 -147.91690
[7]  -145.25594 -147.61525
> summary(subset.Hitter)$cp  # C_p values of 8 selected models
[1] 104.281319  50.723090  38.693127  27.856220  21.613011  14.023870
[7] 13.128474   7.400719
```

The squared regression correlation coefficient R^2 increases when the number of regressors increases.

BIC selects the best subset regression with 6 regressors

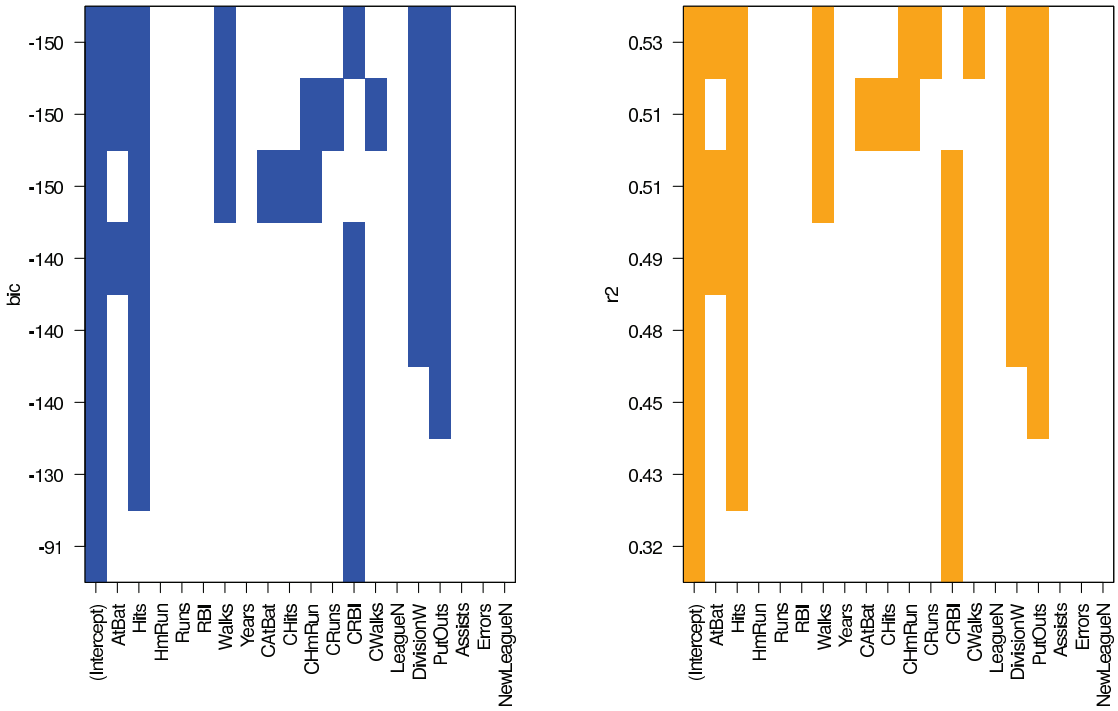
```
> coef(subset.Hitter, 6)
(Intercept)       AtBat        Hits       Walks        CRBI    DivisionW     PutOuts
 91.5117981  -1.8685892   7.6043976   3.6976468   0.6430169 -122.9515338   0.2643076
```

C_p selects the best subset regression with 8 regressors???

One can fit the models with the maximum 19 variables for this dataset:
$>$ `regsubsets(Salary ., data=Hitters1, nvmax=19)`

`regsubsets` has a built-in plots which display the selected variables for the best model with a given number of predictors, ranked according to the BIC, Cp, R^2 and etc.

```
> par(mfrow=c(1,2))
> plot(subset.Hitter, scale="bic", col="blue")
> plot(subset.Hitter, scale="r2", col="orange")
```



Now we apply 10-fold CV to select variables. First we divide randomly the original 253 observations into 10 folds of (about) equal size

```
> folds=rep(1:10, 26) # repeat the sequence {1, ..., 10} 26 times
> length(folds)
> [1] 260
> folds=folds[1:253]  # cut it at length 253
> folds=sample(folds, 253, replace=F)  # change to random order
> folds
  [1] 6 1 7 4 8 9 2 6 2 2 6 2 7 3 1 4 10 2 7 1 8 6 2 7 9 5 5 5 5 5
 ... ...
 #  Observation 1 is in 6th fold, 2 in 1st fold, 3 in 7th fold ..
```

We fit the models using the data in 9 folds, and calculate the CV-RSS using the data in the fold which was left out.

Since `regsubsets` does not have a build-in prediction function, please download the file 'predict.regsubsets.r' from the course moodle page, and put it in your working directory.

```
> source("predict.regsubsets.r")
> cv.errors=matrix(nrow=10, ncol=19)
> for(j in 1:10) {
+      best.fit=regsubsets(Salary~., data=Hitters1[folds!=j,], nvmax=19)
```

```
+      for(i in 1:19) {
+           pred=predict(best.fit, Hitters1[folds==j,], id=i)
+           cv.errors[j,i]=mean((Hitters1$Salary[folds==j]-pred)^2)
+      }
+ }
```

To find the number of regressors which entails the minimum CV-errors across 10 folds

```
> cvErrors=apply(cv.errors, 2, mean) # apply mean to each column
> cvErrors
       1        2        3        4        5        6        7        8        9
151805.0 130218.6 140482.6 132581.5 132990.3 129238.9 124624.2 122341.4 124876.7
      10       11       12       13       14       15       16       17       18
123263.0 124850.5 124516.8 124608.5 125259.7 126883.1 127252.0 127290.5 126773.5
      19
126910.8

> n=1:19    # n is a vector with elements 1,...,19
> n[cvErrors[n] == min(cvErrors)]
> [1] 8     # the value n such that cvErrors[n]=min
```

Thus the 10-fold CV selects the best subset model with 8 predictors. The final selected model should be the one with 8 predictors/regressors but fitted using the whole data set.

Note. Repeating the above computation may lead to a different model (i.e. with a different number of predictors), as the division of the 10 folds is random.

If one set random seed fixed in the beginning of computation, say `set.seed(3)`, the same results will be repeated on the same computer.

To illustrate the application of CV for selecting tree models, we recall some of our treatment of dataset `Carseats` in Chapter 3:

```
> attach(Carseats)
> High=ifelse(Sales<=8, "No", "Yes")    # Define the label High iff Sales >8
> library(tree)
> Carseats2=data.frame(Carseats, High)   # combine the label into the data set
> tree.carseats=tree(High~.-Sales, Carseats2) # . indicates using all the predictors
                                              # -Sales: exclude Sales
> summary(tree.carseats)
Classification tree:
tree(formula = High ~ . - Sales, data = Carseats2)
Variables actually used in tree construction:
[1] "ShelveLoc"   "Price"       "Income"      "CompPrice"   "Population"
[6] "Advertising" "Age" "US"
Number of terminal nodes:  27
Residual mean deviance:  0.4575 = 170.7 / 373
Misclassification error rate: 0.09 = 36 / 400
```

Now we apply CV to determine the tree size.

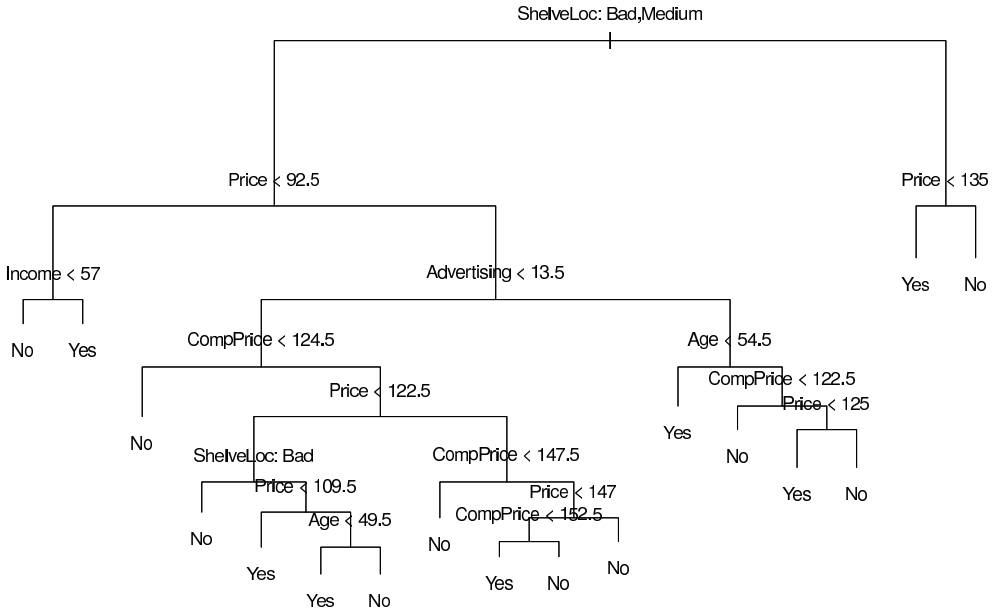
```
> cv.carseats=cv.tree(tree.carseats, FUN=prune.misclass)
> cv.carseats$size   # Number of terminal nodes
 [1] 27 26 24 22 19 17 14 12  7  6  5  3  2  1
> cv.carseats$dev    # Number of CV-misclassfications
```

```
 [1] 104 105 103 102 102 101 102 102 108 107 105 108 117 165
```

The minimum CV-value is 101, the corresponding tree has 17 terminal nodes. Now we apply function `prune.misclass` to prune the tree to obtain the nine node tree

```
> prune.carseats=prune.misclass(tree.carseats, best=17)
> plot(prune.carseats)
> par(mfrow=c(1,1))
> plot(prune.carseats)
> text(prune.carseats, pretty=0)
```

The resulting tree is much simpler and more interpretable than that in Chapter 3.



Chapter 6. Similarity and Nearest Neighbours

- *Basic concepts*: Similarity measures for objects described by data, using similarity for prediction
- *Exemplary techniques*: searching for similar entities, nearest neighbour methods.

Basic idea: If two things (people, products, companies) are similar in some ways, they often share other characteristics as well

Further readings: Provost and Fawcett (2013) Chapter 6 (1st half), James et al. (2013) Section 4.6.5.

Different business tasks involve reasoning from similar examples:

- Retrieve similar things directly
  IBM wants to find companies which are similar to their best business customers
  HP maintains many high-performance severs for clients, aided by a tool that, given a server configuration, retrieves information on other similarly configured servers
  Advertisers serve online ads to consumers who are similar to their current good customers

- Use similarity for classification and regression

- Use similarity for clustering
  Divide the entire customer base into several clusters such that the customers within each cluster are similar in some aspects

- Recommend similar products
  'Customers who bought X have also bought Y'
  'Customers with your browsing history have also looked at $\cdots$'

- Beyond business
  A doctor may reason about a new difficult case by recalling a similar case
  A lawyer often argues cases by citing legal precedents
  Chinese Medicine
  Case-based reasoning systems are built based on Artificial Intelligence

Similarity and Distance

Objects are represented by data. Typically each object has multiple attributes, i.e. represented by a vector. Therefore we may calculate

the distance between two vectors: the larger the distance is, the less similar the two objects are.

There are many distance measures for vectors $\mathbf{x} = (x_1, \ldots, x_p)^T, \mathbf{y} = (y_1, \cdots, y_p)^T \in R^p$.

- Euclidean (or L_2) distance: $\left((x_1 - y_1)^2 + \cdots + (x_p - y_p)^2\right)^{1/2}$
- L_1 distance: $|x_1 - y_1| + \cdots + |x_p - y_p|$
- L_∞ distance: $\max_j |x_j - y_j|$
- Weighted L_2 distance: $\left(w_1(x_1 - y_1)^2 + \cdots + w_p(x_p - y_p)^2\right)^{1/2}$, where $w_j \geq 0$ are a set of weights
- Correlation based distance: $1 - \rho(\mathbf{x}, \mathbf{y})$, where

$$\rho(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{p} (x_i - \bar{x})(y_i - \bar{y}) \Big/ \Big\{ \sum_{i=1}^{p} (x_i - \bar{x})^2 \sum_{j=1}^{p} (y_j - \bar{y})^2 \Big\}^{1/2}.$$

- Mahalanobis distance: Let $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})$‘, $i = 1, \cdots, n$ be n vectors, the Mahalanobis distance between any pair $\mathbf{x}_i$ and $\mathbf{x}_j$ is

$$d_{ij} \equiv \Big\{ \sum_{k,l=1}^{p} (x_{ik} - x_{jk})(x_{il} - x_{jl}) a_{kl} \Big\}^{1/2},$$

where $\mathbf{A} = (a_{ij})$ is the inverse of the sample covariance matrix

$$\mathbf{A}^{-1} = \frac{1}{n-1} \sum_{j=1}^{n} (\mathbf{x}_j - \bar{\mathbf{x}})(\mathbf{x}_j - \bar{\mathbf{x}})^T, \quad \bar{\mathbf{x}} = \frac{1}{n} \sum_{j=1} \mathbf{x}_j.$$

Note. The Mahalanobis distance can be viewed as a distance of the normalized and decorrelated data: $\mathbf{y}_i = \mathbf{A}^{1/2} \mathbf{x}_i$, and

$$d_{ij} = \Big\{ \sum_{k=1}^{p} (y_{ik} - y_{jk})^2 \Big\}^{1/2}.$$

A credit card marketing problem: predict if a new customer will respond to a credit card offer based on how other, similar customers have responded.

Customer	Age	Income	Cards	Response	Distance to David
David	37	50	2	?	0
John	35	35	3	Yes	15.16
Rachael	22	50	2	No	15
Ruth	63	200	1	No	152.23
Jefferson	59	170	1	No	122
Norah	25	40	4	Yes	15.74

Note. Euclidean distance was used. For example, the distance between John and David is

$$15.16 = \sqrt{(35-37)^2 + (35-50)^2 + (3-2)^2}.$$

Rachael is most similar to David, and Ruth is most dissimilar to David.

Distance, as a (dis)similarity measure, is just a number: it has no units, no meaningful interpretation in general.

Distance is useful for comparing the similarity among different pairs: the relative sizes matter.

Distance measures should be used with care, especially the attributes (i.e. components of vectors) are inhomogeneous, contain some nominal variables, or on different scales.

Further consideration: using Mahalanobis distance or a weighted Euclidean distance (with heavier weight on, for example, number of cards)?

Example: Whiskey Analysis

Source: `http://adn.biol.umontreal.ca/~numericalecology/data/scotch.ht`

For someone who loves *single malt Scotch whiskey*, it is important to be able to identify those similar to a particular single malt he really likes, among hundreds of different single malts.

Single malt Scotch whisky is one of the most revered spirits in the world. It has such scope for variation, it can offer complexity or simplicity, unbridled power or a subtle whisper. To legally be called a single malt Scotch, the whisky must be distilled at a single distillery in Scotland, in a copper pot still from nothing other than malted barley, yeast and water. It must then be aged in an oak cask for at least three years and a day, and be bottled at no less than 40% abv.

Suppose Foster likes *Bunnahabhain*, he likes to find other ones like that among all the many single malts.

We need to define a feature vector for each single malt such that the similar vectors (i.e. with small distances among them) represent whiskeys with similar taste.

Lapointe and Legendre (1994). A classification of pure malt Scotch whiskies. *Applied Statistics*, **43**, 237-257.

Jackson (1989) *Michael Jackson's Malt Whisky Companion: A Connoisseur's Guide to the Malt Whiskies of Scotland*: tasting notes for 109 different single malt Scotches.

The tasting notes are in the form of literary descriptions on 5 whiskey attributes: Color, Nose, Body, Palate, Finish.

One example: 'Appetizing aroma of peat smoke, almost incense-like, heather honey with a fruity softness'.

Question: How to turn those literary descriptions into numerical feature vectors?

Collecting different values for 5 attributes:

Color: yellow, very pale, pale, gold, pale gold, old gold, full gold, amber, etc (14 values)

Nose: aromatic, peaty, sweet, light, fresh, sea, dry, grassy, etc (12 values)

Body: soft, full, round, smooth, firm, medium, light, oily (8 values)

Palate: full, dry, sherry, big, sweet, fruity, clean, grassy, smoky, salty, etc (15 values)

Finish: full, dry, warm, light, smooth, clean, fruity, grassy, smoky, etc (19 values)

The values in blue are characteristics for Bunnahabhain. Multiple values are possible for each whiskey!

68 values in total: a 68×1 vector with components equal to 1 (present) or 0 (absent).

Calculate the Euclidean distances between Bunnahabhain and each of the other 108 single malts, the most similar ones are:

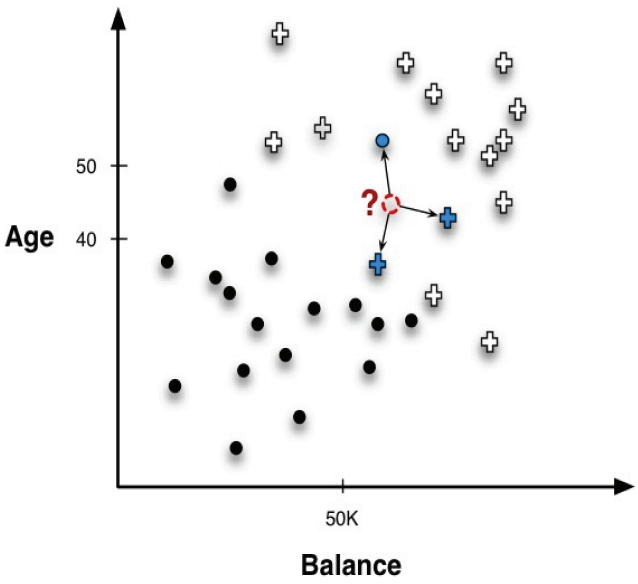
Whiskey	Distance	Description
Bunnahabhain	0	gold; firm,ned,light; sweet,fruit,clean; fresh,sea; full
Glenglassaugh	.643	gold; firm,light,smooth; sweet,grass; fresh,gr
Tullibardine	.647	gold; firm,med,smooth; sweet,fruit,full,grass, clean; sweet; big,aroma,sweet
Ardbeg	.667	sherry; firm,med,full,light; sweet; dry,peat, sea; salt
Bruichladdich	.667	pale; firm,light,smooth; dry,sweet,smoke,clea light; full
Glenmorangie	.667	p.gold; med,oily,light; sweet,grass,spice; sweet,spicy,grass,sea,fresh; full, long

Nearest Neighbours (NN)

Similarity defines nearest neighbours for a individual: the individual with the smallest distance is the nearest neighbour. For any $k \geq 1$, the k-nearest neighbours are the k individuals. with the k smallest distances.

Using NN for prediction: predict the target value for the new individual by using the average (for regression) or the majority votes (for classification) of the known target values of its k-nearest neighbours among the training data.

Question: How to choose k?



NN-classification: The poin
labeled with ? is classified a
+ by 3-NN method, as th
majority (i.e. 2 out of 3) c
its neighbours are +.

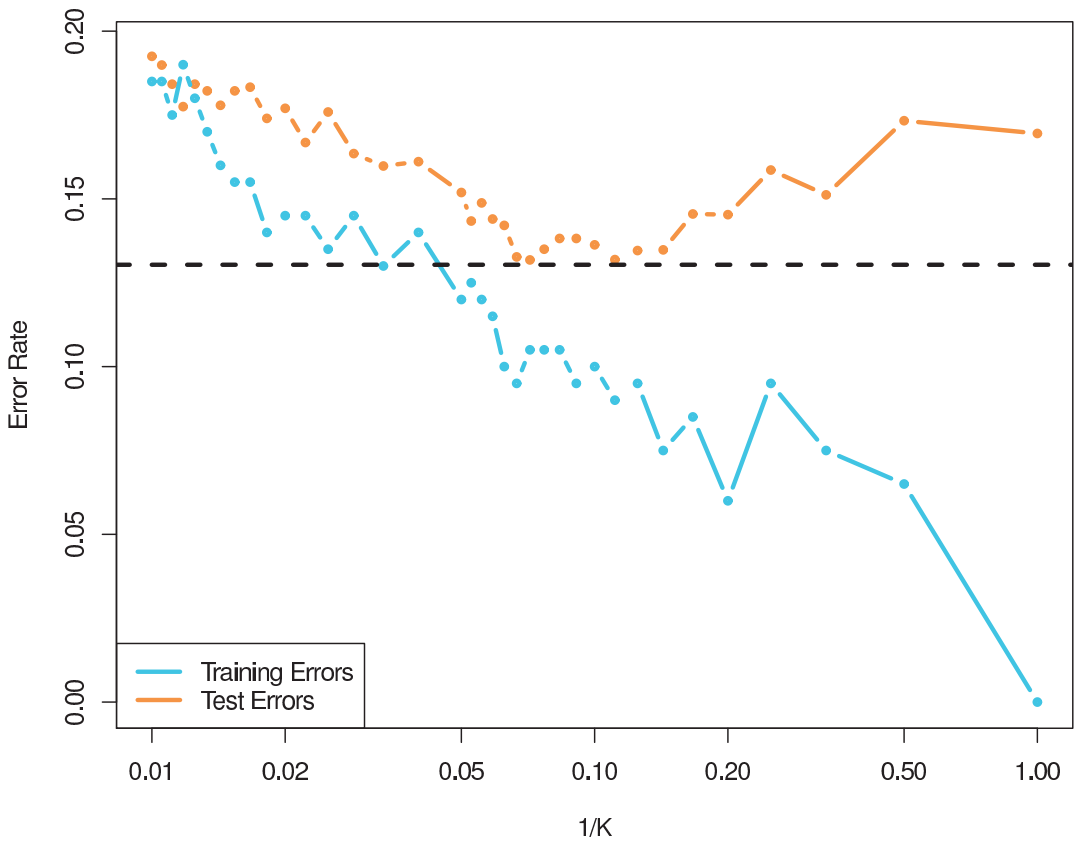
How many neighbours?

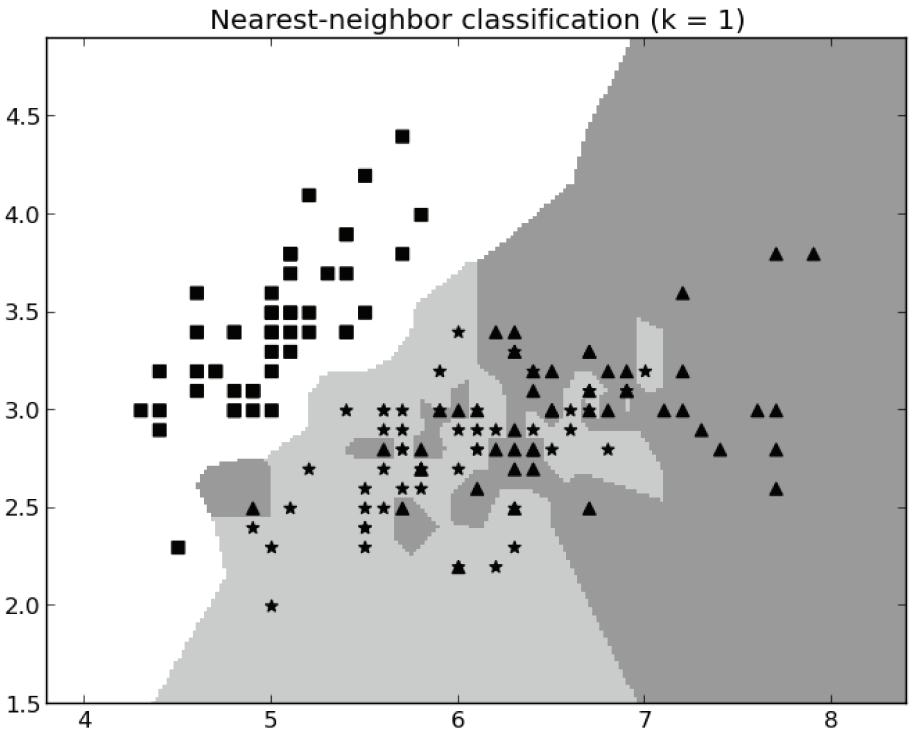
The k in k-NN method serves as a smooth parameter: the larger k is, the more smooth the method is.

When $k = n$, the implied regression model is a constant (i.e. sample mean), and the implied classification method puts all objects into one class.

$k = 1$ leads to overfitting, the complexity of the method is at its maximum,

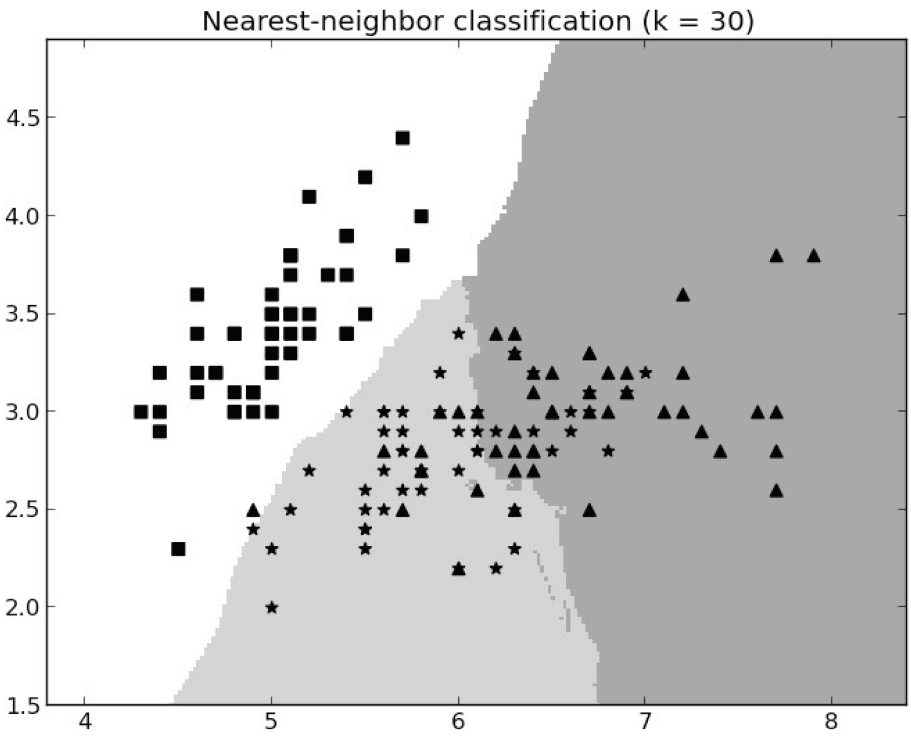
Choosing k by cross-validation or the other holdout methods.





3 class,

classification boundaries created by 1-N



3 class,

classification boundaries created by 30-NN

A NN approach to making recommendations

Preparing recommendation for a new customer in an automatic manner may consist of three steps:

- Building a customer profile by getting the new customer to rate a selection of items such as movies, songs, or restaurants
- Comparing the new customer profile with the profiles of other customers using some measure of similarity
- Using some combination of the ratings of customers with similar profiles to predict the rating that the new customer would give to items he or she has not yet rated.

Building profiles

Sparseness of profiles: There are often far more items to be rated than any one person is likely to have experienced or be willing to rate. A user profile is a numeric vector consisting of, e.g. the digits between -5 (most negative) to 5 (most positive), while 0 stands for neutrality or no opinion.

On the other hand, forcing customers to rate a particular subset may miss interesting information because ratings of more obscure items may say more about the customer than ratings of common ones. A fondness for the Beatles is less revealing than a fondness for Mose Allison.

A reasonable approach is to have new customers rate a list of the twenty or so most frequently rated items (a list that might change over time) and then free them to rate as many additional items as they please.

Comparing profiles

Once a customer profile has been built, the next step is to measure its distance from other profiles. The most obvious approach would be to treat the profile vectors as geometric points and calculate the Euclidean distance between them, but many other distance measures have been tried. Some give higher weight to agreement when users give a positive rating especially when most users give negative ratings to most items. Still others apply statistical correlation tests to the ratings vectors.

Making Predictions

The final step is to use some combination of nearby profiles in order to come up with estimated ratings for the items that the customer has not rated. One approach is to take a weighted average where the weight is inversely proportional to the distance.

Issues with Nearest-Neighbour Methods

- *Justification*

  In some fields such as medicine or law, reasoning about similar cases is a natural way of making decision about a new case, a NN method may be a good fit.

  However a mortgage applicant may not be satisfied with the explanation: 'We decline your application because you remind us of the Smiths and the Mitchells, who both defaulted'.

  In contrast with a regression model, one may be able to say: 'all else being equal, if your income has been $ 20,000 higher, you would have been granted this mortgage.'

  Some careful and judicious presentation of NN based decision is useful.

Netflix uses a NN classification for their recommendations, explaining the recommendations with sentences like: 'The movie *Billy Elliot* was recommended based on your interest in *Amadeus, The Constant Gardener* and *Little Miss Sunshine*'.

- *Interpretation*

  It is difficult to explain what 'knowledge' has been used from the data in a NN method.

- *Dimensionality*

  Most practical problems have too many seemingly relevant attributes, i.e. vectors are very long. For a particular problem, many those attributes are irrelevant. In practice, either feature/variable selection method should be used (multi-fold CV) to select those to be used for calculating distance measures, or domain knowledge should be used to define an appropriate distance.

We consider a wholesale customers data set available at UCI Machine Learning Repository
`https://archive.ics.uci.edu/ml/datasets/wholesale+customers`
It is also available from the course Moodle as 'wholesaleCustomers.csv'. Download the data set and put it in your working directory.

```
> customers=read.csv("wholesaleCustomers.csv")
> dim(customers)
[1] 440   8
> View(customers)
```

It contains the info on 440 clients of a wholesale distributor: the annual spending on monetary units (m.u.) on 6 product categories.

Since the dataset is openly available, it has been analysed for quite a few times already. You can easily find them via google search.

Note. Google Dataset Search `https://toolbox.google.com/datasetsearch` can help you to find many datasets. Try searching for, eg., customers, churn, recommendation.

There are 8 variables/columns:

`Channel`: 1 - Horeca (Hotel/Restaurant/Cafe), 2 - Retail

`Region`: 1 - Lisbon, 2 - Oporto, 3 - Other regions

`Fresh`: annual spending (m.u.) on fresh products

`Milk`: annual spending on milk products

`Grocery`: annual spending on grocery products

`Detergents_paper`: annual spending on detergents and paper products

`Delicatessen`: annual spending on delicatessen products.

```
> summary(customers)
    Channel          Region          Fresh             Milk          Grocery
 Min.   :1.000   Min.   :1.000   Min.   :     3   Min.   :   55   Min.   :    3
 1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3128   1st Qu.: 1533   1st Qu.: 2153
 Median :1.000   Median :3.000   Median :  8504   Median : 3627   Median : 4756
 Mean   :1.323   Mean   :2.543   Mean   : 12000   Mean   : 5796   Mean   : 7951
 3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16934   3rd Qu.: 7190   3rd Qu.:10656
 Max.   :2.000   Max.   :3.000   Max.   :112151   Max.   :73498   Max.   :92780
     Frozen        Detergents_Paper    Delicassen
 Min.   :   25.0   Min.   :    3.0   Min.   :    3.0
 1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2
```

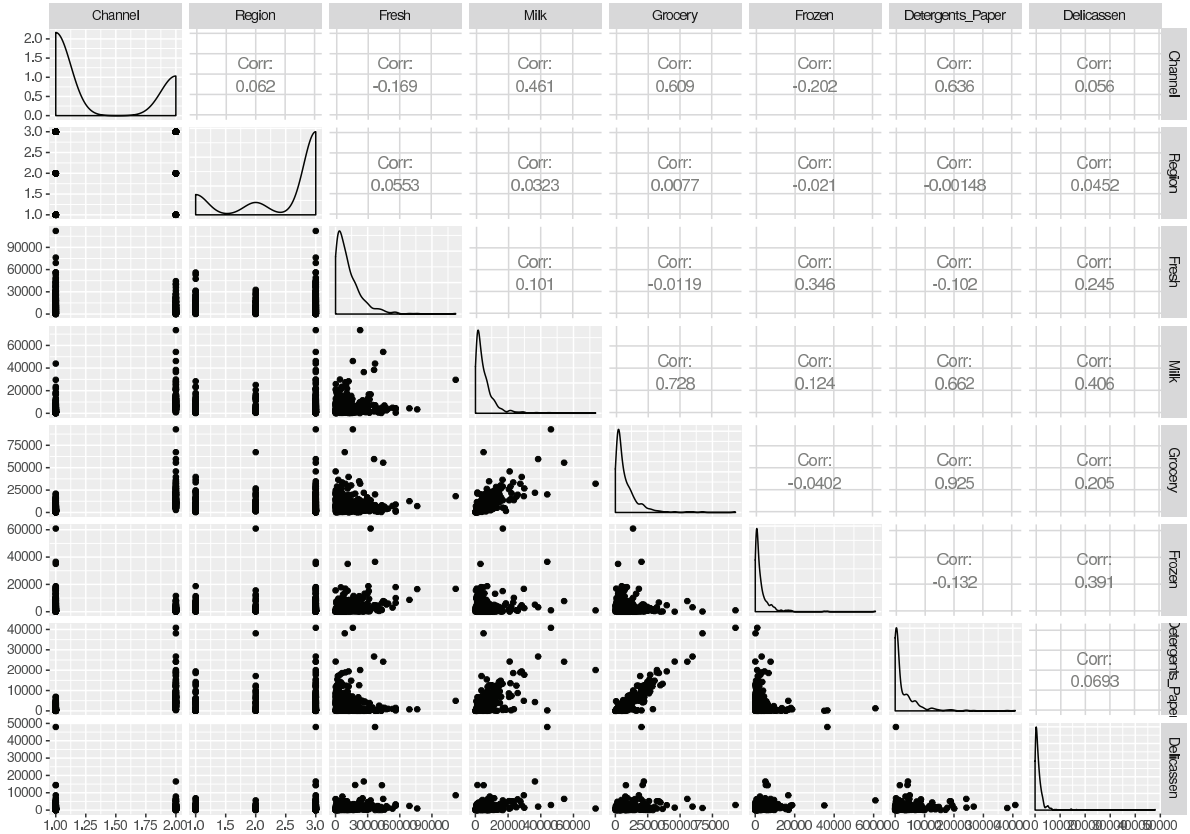
```
 Median : 1526.0   Median :  816.5   Median :  965.5
 Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9
 3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2
 Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
> library(GGally)
> ggpairs(customers)
```

No missing values, substantial variations in spending.

`Region` shows little correlation with all other variables

Only retail clients spend heavily on `Grocery` and `Detergents_paper`

Strong correlations between `Grocery` and `Detergents_paper`, `Milk` and `Grocery`



Now how we define the distances between pair clients? Here are some possibilities:

1. Euclidean, L_1 or maximum component distance:

```
> dist_euc=dist(customers, method="euclidean")
 # Change euclidan to maximum for the maximum component, to manhattan
 # for $L_1$ distance, and to canberra for
 # $\sum_i |x_i - y_i|/|x_i + y_i|$
> summary(dist_euc)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
   278.9  10083.6  16302.6  20848.7  25741.6 128968.4
> length(dist_euc)
[1] 96580  # = 440*439/2
> dist_euc=as.matrix(dist(customers, method="euclidean")) #  distance matrix
> dim(dist_euc)
[1] 440 440
> sort(dist_euc[2,])[1:4] #  rearrange the componemts  in ascending order
       2      245      397      165
   0.000 2612.974 3499.196 3509.257
# The 3 nearest neighbours of Row 2 are Rows 245, 397, 165
```

The contribution of `channel, Region` in the above distance is negligible!

2. absolute difference in `Channel` + Mahalanobis distance of 6 spending variables

```
> attach(customers)
> D1=outer(Channel, Channel, "-")  # D1[i,j] = Channel[i]-Channel[j]
> dim(D1)
[1] 440 440
> customer6=customers[,3:8]
> S=var(customer6)
> S
                     Fresh     Milk  Grocery   Frozen Detergents_Paper Delicasse
Fresh            159954927  9381789 -1424713 21236655       -6147825.7  8727310.
Milk               9381789 54469967 51083186  4442612       23288343.5  8457924.
Grocery           -1424713 51083186 90310104 -1854282       41895189.7  5507291.
Frozen            21236655  4442612 -1854282 23567853       -3044324.9  5352341.
Detergents_Paper  -6147826 23288343 41895190 -3044325       22732436.0   931680.
Delicassen         8727310  8457925  5507291  5352342         931680.7  7952997.
> tt=eigen(S, symmetric=T) # perform eigen-analysis for matrix S
> d=tt$values  # eigenvalues
> G=tt$vectors # 6x6 matrix with eigenvectors as columns
> S2=G%*%diag(1/sqrt(d))%*%t(G)  # S2 = S^{-1/2}
> customer6N=as.matrix(customer6)%*%S2  # Normalize the columns of customer6
> var(customer6N)
              [,1]          [,2]          [,3]          [,4]          [,5]          [,6]
[1,]  1.0000e+00 -3.0375e-17 -2.7856e-16  1.4898e-16 -2.8542e-16  6.8510e-17
[2,] -3.0375e-17  1.0000e+00  2.8102e-16  2.2774e-16  4.3874e-16  9.1557e-16
[3,] -2.7856e-16  2.8102e-16  1.0000e+00 -2.3778e-16  2.8542e-16 -4.6251e-16
[4,]  1.4898e-16  2.2774e-16 -2.3778e-16  1.0000e+00  2.4173e-16  5.4908e-16
[5,] -2.8542e-16  4.3874e-16  2.8542e-16  2.4173e-16  1.0000e+00  2.4843e-16
[6,]  6.8510e-17  9.1557e-16 -4.6251e-16  5.4908e-16  2.4843e-16  1.0000e+00
> dist_maha = abs(D1) + as.matrix(dist(customer6N, method="euclidean"))
> sort(dist_maha[2,])[1:4]
        2       109       397        83
0.0000000 0.5688088 0.5721256 0.6339615
```

Now the 3 nearest neighbours of Row 2 are Rows 109, 397 and 83.

The above calculation is based on the fact that a Mahalanobis distance is the Euclidean distance and normalized vectors.

We can use `k*abs(D1) + as.matrix(dist(customer6N, method="euclidean"))` instead, where constant $k > 0$ balances the relative importance of the two terms. We may even consider to choose k according to some appropriate criterion.

3. absolute difference in `Channel` + (1 - correlation based on 6 spending variables)

```
> dist_cor=abs(D1)+1-cor(t(customer6))
# cor calculates the correlations btw columns, hence tranpose t(customer6)
> sort(dist_cor[2,])[1:4]
[1] 0.00000000 0.01309468 0.04160087 0.04768788
> sort.int(dist_cor[2,], index.return=T)$ix[1:4]  # check ?sort and ?sort.int
[1]   2  48  95 165
> sort.int(dist_cor[2,], index.return=T)$x[1:4]
[1] 0.00000000 0.01309468 0.04160087 0.04768788
```

Now the 3 nearest neighbours of Row 2 are Rows 48, 95, 165.

Chapter 7. Clustering

An unsupervised learning technique: more challenging, more subjective, often more difficult to perform evaluation of the results obtained: no cross-validation

Two techniques: Hierarchical clustering, and K-means clustering

Further readings:

James et al. (2013) Sections 10.3 & 10.5,

Provost and Fawcett (2013) Chapter 6.

Classification: finding groups of objects that differ with respect to some target characteristics of interest. So the groups are known in prior, defined as the target characteristics.

Clustering: finding groups of objects such that the objects within each group are similar, but the objects in different groups are not so similar. So the number of groups and the groups themselves are unknown in prior.

- Do our customers naturally fall into different groups, so we can develop better products, better marketing campaigns, better sales methods, or better customer service by understanding the natural subgroups?

- A search engine might choose what search results to display to a particular individual based on the click histories of other individuals with similar search patterns.

- Clustering single-malt-scotch whiskeys: natural groupings by taste – run a small shop as 'a place to go for single-malt scotch' in a well-to-do neighbourhood

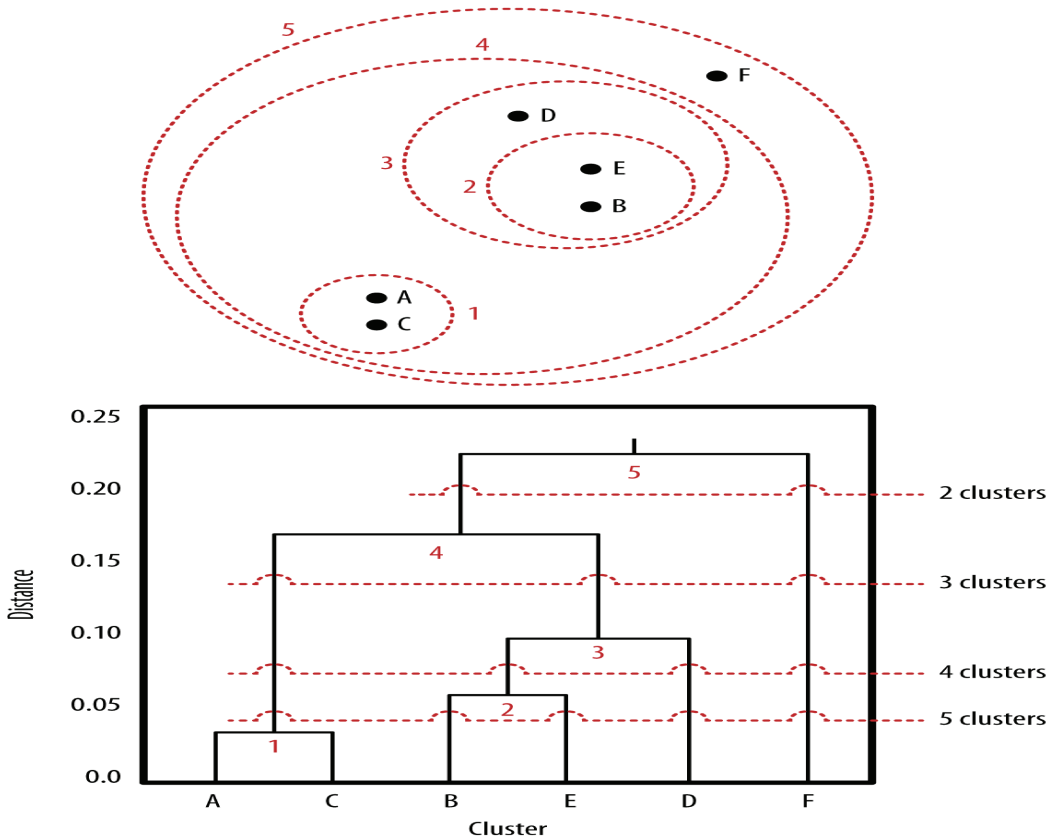
Hierarchical Clustering: clustering by similarity, dendrogram

1. Calculate an appropriate distance for any two objects. At distance 0, put each object into a separate group.

2. Put two groups with the smallest distance together. Update the distances between any two groups using one of the four linkages:

- *Complete*: the maximum pairwise distance between the objects from two groups
- *Single*: the minimum pairwise distance between the objects from two groups
- *Average*: the average pairwise distance between the objects from two groups
- *Centroid*: the distance between the centroids (i.e. the mean vectors) of the two groups.

3. Repeat Step 2 above until all the objects are in one group.



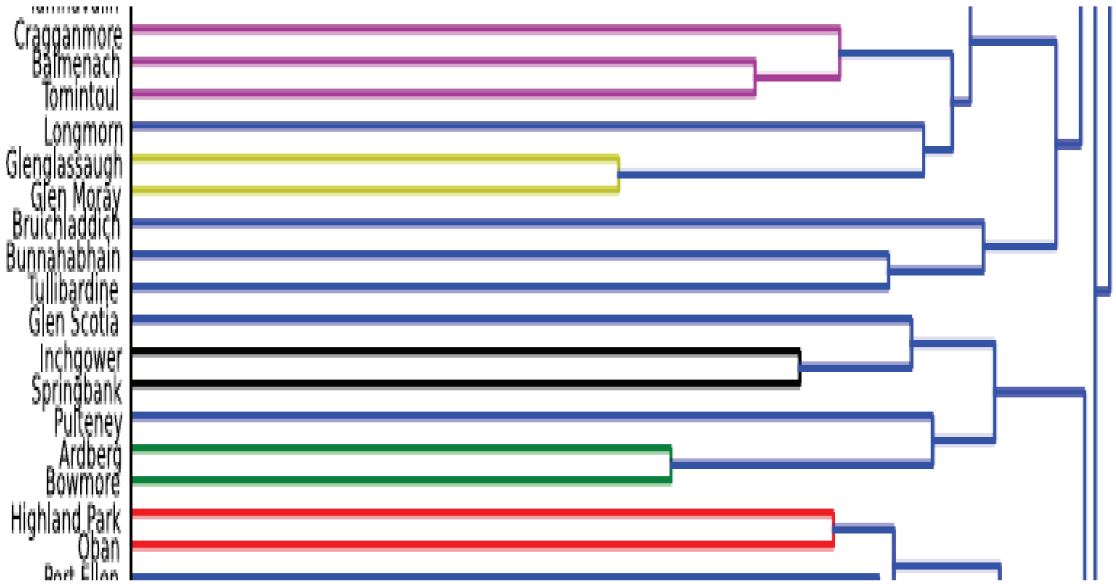
Dendrogram: a landscape of data similarity

Any number of clusters can be obtained by cutting dendrogram at an appropriate height (i.e. distance)

In practice, the number of clusters is often chosen by looking at dendrogram.

For the example on previous page, 3 clusters could be selected as there is a relatively long distance between Group 3 (0.10) and Group 4 (0.17).

An Excerpt of hierarchical clustering of single malt Scotch whiskeys.



Choice of Dissimilarity Measure and Linkage Function: very important, strong effect on the resulting dendrogram

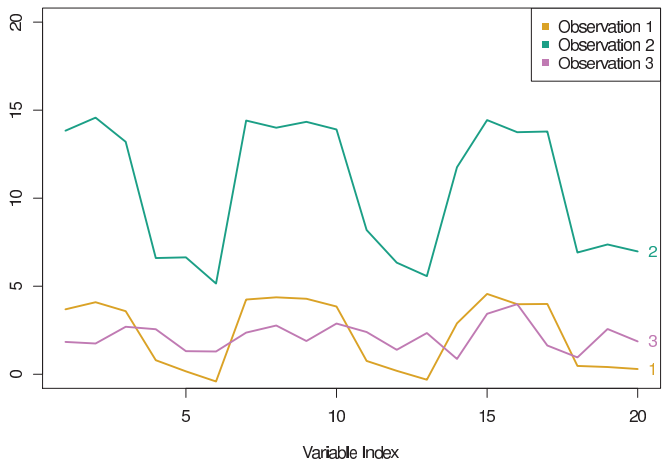
Type of data, business question in hand should be taken into account.

Clustering online shoppers: Data form a matrix with shoppers being rows and items available for purchase being columns, the elements of the matrix indicate the number of the times a given shopper has purchased a given item

Using Euclidean distance clusters together the shoppers who have bought very few items, which is not desirable.

Correlation-based distance clusters together shoppers with similar preferences (e.g. bought A and B but not C and D etc).

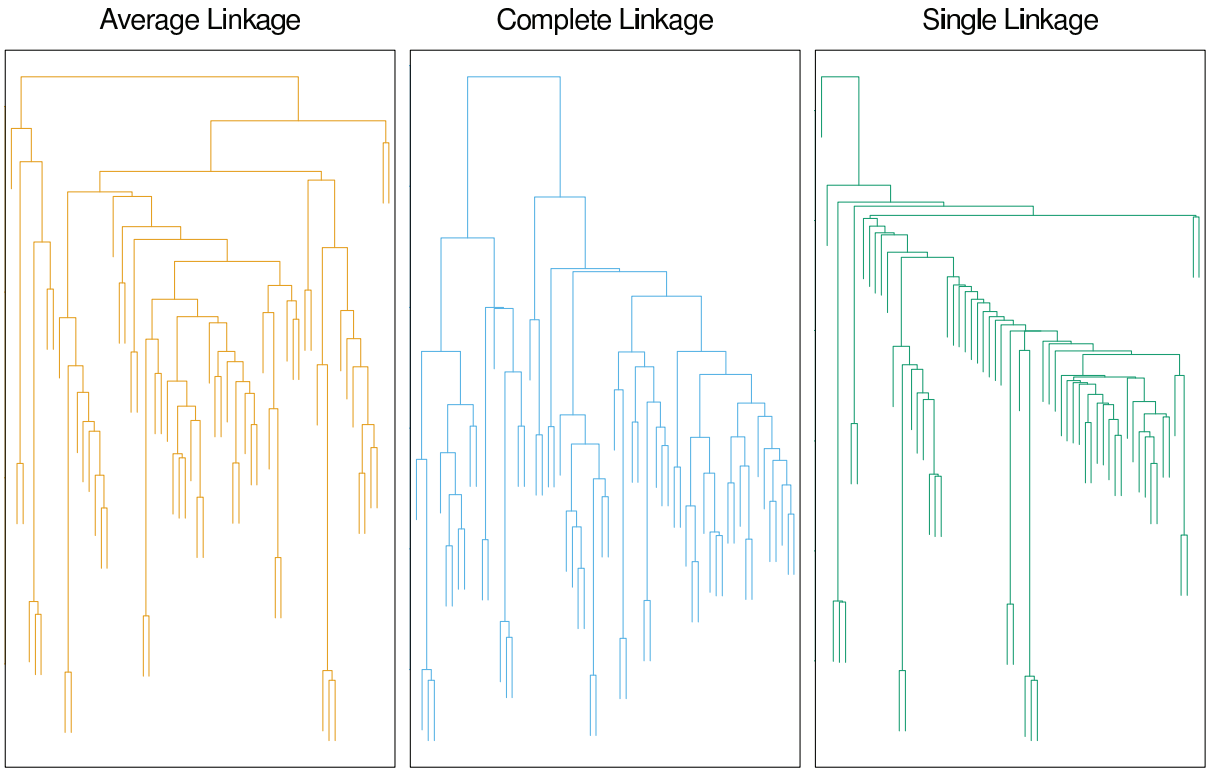
Complete linkage would make the shoppers in the same cluster more homogeneous, such that same ads can be shown to each clusters.



Each observation is measured on 20 attributes.

Euclidean distance between Observations 1 and 3 is small.

Correlation between Observations 1 and 2 is large, so the correlation based distance is small.



K-**means clustering**: partition the whole objects into K distinct, non-overlapping clusters.

An nearest neighbour approach: each object is represented by a p-feature vector.

K needs to be pre-specified.

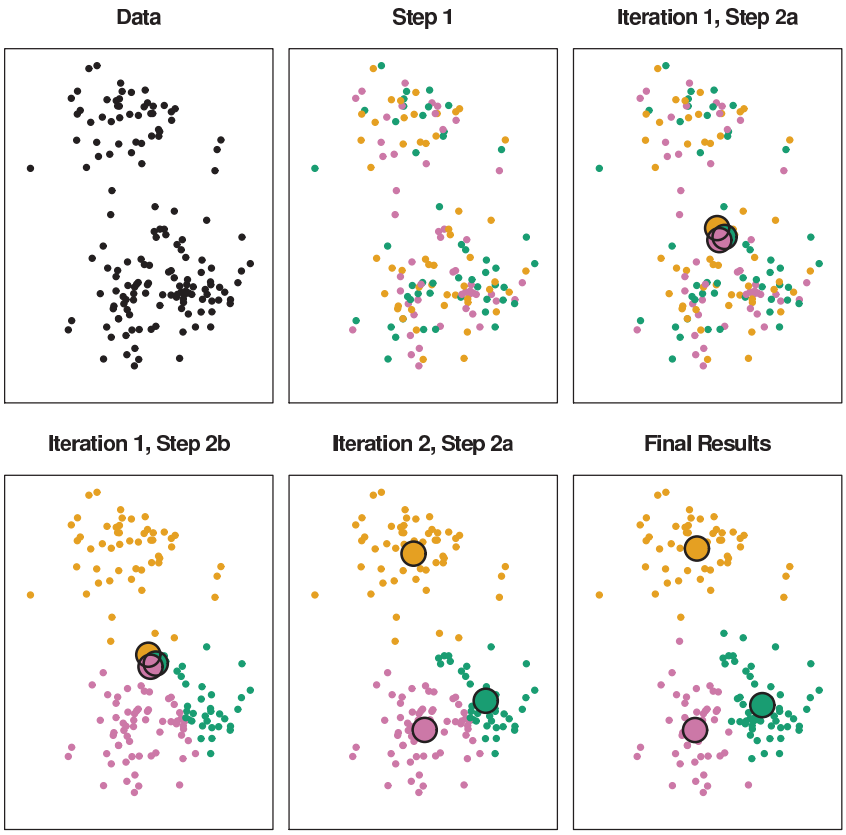
1. Randomly assign all the objects into one of K clusters

2. Iterate until the cluster assignments stop changing:

   (a) Calculate the centroid (i.e. a p-feature mean vector) for each of the K clusters

   (b) Assign each object to the cluster whose centroid is closest, measured using, e.g. Euclidean distance.

Remark. Step 1 assigns an initial value. When p is large, using a good initial value is important.



K-means clustering with K
3 for randomly generated
points.

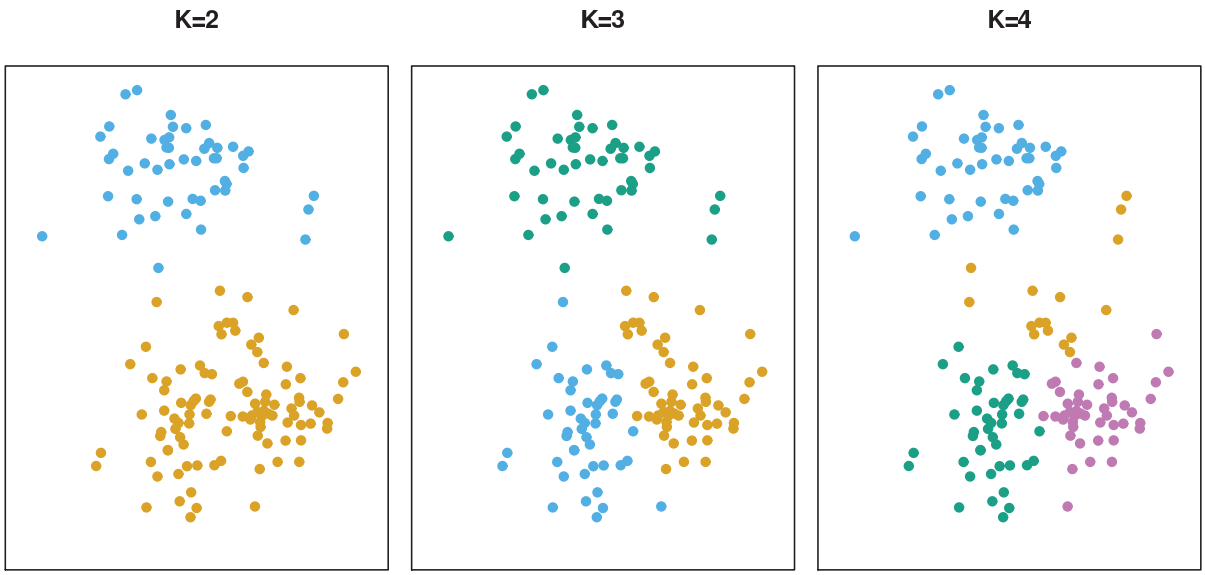
Step 1. Randomly assign
points to 3 groups

Step 2(a) Calculate centro
for each of 3 groups

Step 2(b) Assign each po
to the cluster with the clos
centroid

Performing 6 times the
means clustering with $K =$

3 different local optima w
obtained



Example: Clustering Business News Stories

Objective: identify different groupings of news stories released on a particular company.

Purposes: gain a quick understanding of the news on a company without having to read every news story; categorize forthcoming news stories for a news prioritization process; or simply to understand data better before undertaking deeper learning (such as to relate business news stories to stock performance).

Data: Thomson Reuters Text Research Collection (TRC2)

`http://trec/nist/gov/data/reuters/reuters.html`

1,800,370 news stories from January 2008 to February 2009 (14 months)

312 news stories whose headlines specifically mentioned company Apple or its stock symbol APPL from the above corpus

Data Preparation:

remove HTML, URL and other stop words

text case-normalized

eliminate words which occurred rarely (fewer than 2 documents)

eliminate words which occurred too commonly (more than 50% documents)

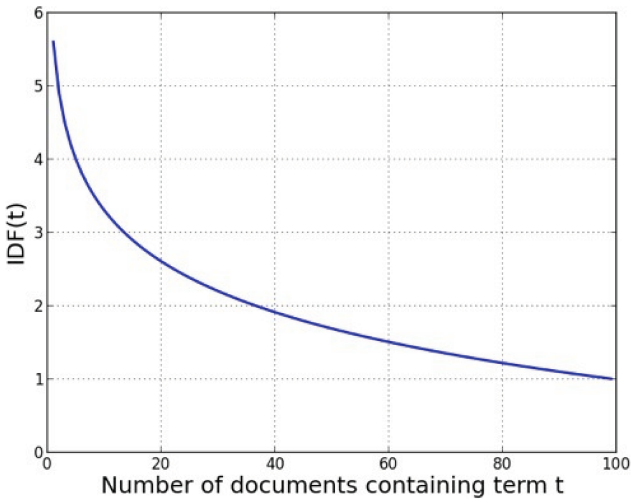
The remaining words form the *vocabulary*.

Each document is represented by a long numerical vector consisting of "TFIDF score" for each word in the vocabulary.

$$\text{TFIDF} = \text{TF (term frequency)} \times \text{IDF (inverse document frequency)}$$

$$\text{TF(t, d)} = \text{No. of times of word t occurring in document d}$$

$$\text{IDF(t)} = 1 + \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing word t}}\right)$$



IDF of a term t within a corpus of 100 documents

9 clusters for the Apple news stories, derived by K-means clustering method.

Below we present a description of the 9 clusters, along with some headlines of the stories contained in each clusters. Note that entire news story was used in the clustering, not just the headline.

Cluster 1. These stories are analyst's announcements concerning ratings changes and price target adjustments:

```
* RBC RAISES APPLE <AAPL.O> PRICE TARGET TO $200 FROM $190; KEEPS OUT
PERFORM RATING
* THINKPANMURE ASSUMES APPLE <AAPL.O> WITH BUY RATING; $225 PRICE TARG
* AMERICAN TECHNOLOGY RAISES APPLE <AAPL.O> TO BUY FROM NEUTRAL
* CARIS RAISES APPLE <AAPL.O> PRICE TARGET TO $200 FROM $170; RATING
ABOVE AVERAGE
* CARIS CUTS APPLE <AAPL.O> PRICE TARGET TO $155 FROM $165; KEEPS ABOV
```

```
AVERAGE RATING
```

Cluster 2. This cluster contains stories about Apple's stock price movements, during and after each day of trading:

```
* Apple shares pare losses, still down 5 pct
* Apple rises 5 pct following strong results
* Apple shares rise on optimism over iPhone demand
* Apple shares decline ahead of Tuesday event
* Apple shares surge, investors like valuation
```

Cluster 3. In 2008, there were many stories about Steve Jobs, Apple's charismatic CEO, and his struggle with pancreatic cancer. Jobs' declining health was a topic of frequent discussion, and many business stories speculated on how well Apple would continue without him. Such stories clustered here:

```
* ANALYSIS-Apple success linked to more than just Steve Jobs
```

* NEWSMAKER-Jobs used bravado, charisma as public face of Apple
* COLUMN-What Apple loses without Steve: Eric Auchard
* Apple could face lawsuits over Jobs' health
* INSTANT VIEW 1-Apple CEO Jobs to take medical leave
* ANALYSIS-Investors fear Jobs-less Apple

Cluster 4. This cluster contains various Apple announcements and releases. Superficially, these stories were similar, though the specific topics varied:

* Apple introduces iPhone "push" e-mail software
* Apple CFO sees 2nd-qtr margin of about 32 pct
* Apple says confident in 2008 iPhone sales goal
* Apple CFO expects flat gross margin in 3rd-quarter
* Apple to talk iPhone software plans on March 6

Cluster 5. This cluster's stories were about the iPhone and deals to sell iPhones in other countries:

* MegaFon says to sell Apple iPhone in Russia
* Thai True Move in deal with Apple to sell 3G iPhone
* Russian retailers to start Apple iPhone sales Oct 3
* Thai AIS in talks with Apple on iPhone launch
* Softbank says to sell Apple's iPhone in Japan

Cluster 6. One class of stories reports on stock price movements outside of normal trading hours (known as Before and After the Bell):

* Before the Bell-Apple inches up on broker action
* Before the Bell-Apple shares up 1.6 pct before the bell
* BEFORE THE BELL-Apple slides on broker downgrades
* After the Bell-Apple shares slip
* After the Bell-Apple shares extend decline

Centroid 7. This cluster contained little thematic consistency:

* ANALYSIS-Less cheer as Apple confronts an uncertain 2009

* TAKE A LOOK - Apple Macworld Convention
* TAKE A LOOK-Apple Macworld Convention
* Apple eyed for slim laptop, online film rentals
* Apple's Jobs finishes speech announcing movie plan

Cluster 8. Stories on iTunes and Apple's position in digital music sales formed this cluster:

* PluggedIn-Nokia enters digital music battle with Apple
* Apple's iTunes grows to No. 2 U.S. music retailer
* Apple may be chilling iTunes competition
* Nokia to take on Apple in music, touch-screen phones
* Apple talking to labels about unlimited music

Cluster 9. A particular kind of Reuters news story is a News Brief, which is usually just a few itemized lines of very terse text (e.g. 'Says purchase new movies on iTunes same day as dvd release'). The contents of these New Briefs varied, but because of their very similar form they clustered together:

* BRIEF-Apple releases Safari 3.1
* BRIEF-Apple introduces ilife 2009
* BRIEF-Apple announces iPhone 2.0 software beta
* BRIEF-Apple to offer movies on iTunes same day as DVD release
* BRIEF-Apple says sold one million iPhone 3G's in first weekend

Some of these clusters are interesting and thematically consistent while others are not. Some are just collections of superficially similar text.

Syntactic similarity is not semantic similarity.

We should not expect every cluster to be meaningful and interesting. Nevertheless, clustering is often a useful tool to uncover structure in our data that we did not foresee. Clusters can suggest new and interesting data mining opportunities.

Small Decision with Big Consequences

- Should the observations or features first be standardized in some way? For example, make the mean of each feature 0, the STD 1.

- For hierarchical clustering,

  What distance measure should be used?
  What linkage should be used?
  Where should one cut the dendrogram in order to obtain clusters?

- For K-means clustering,

  How many clusters should we set for?
  What distance measure should be used?

In practice, try different choices, and look for the one with most useful or interpretable solution.

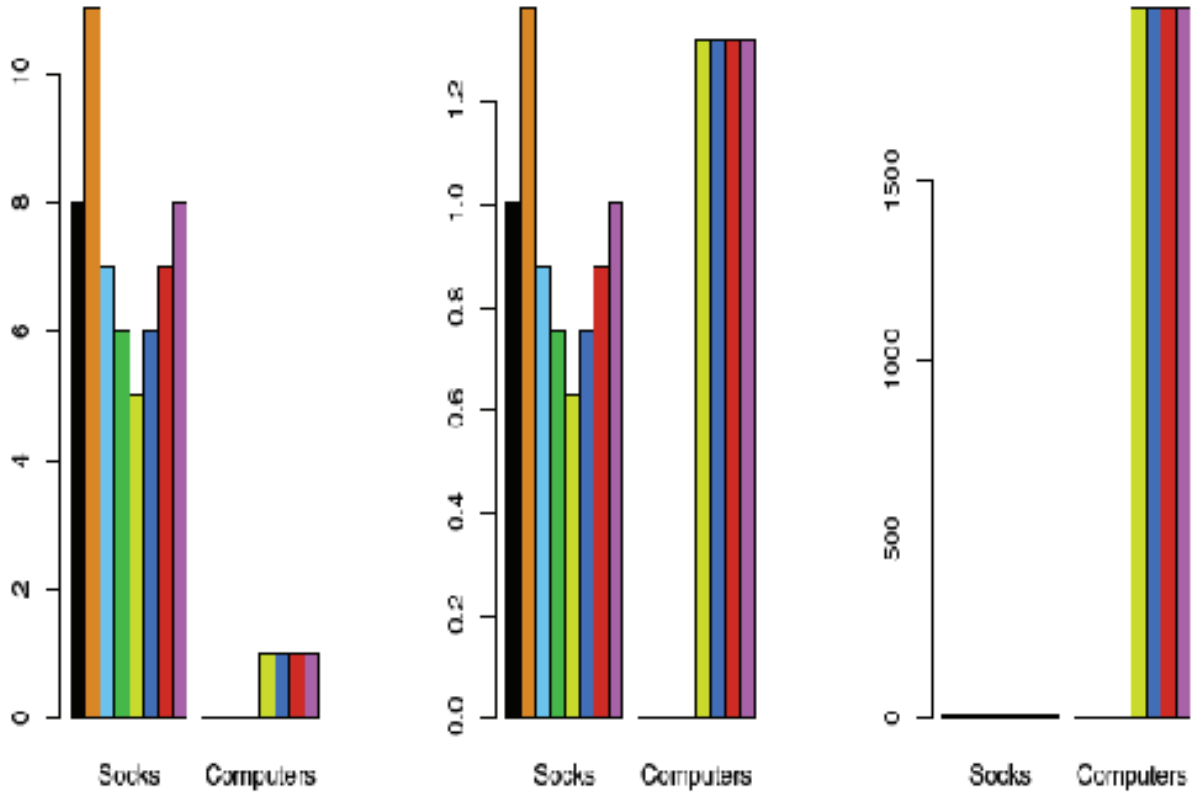
There is no single right answer for Clustering. Any solution which exposes some interesting aspects of the data should be considered.

An eclectic online retailer sells two items: socks and computers

Left the number of pairs of socks, and computers, purchased by eight online shoppers is displayed. Each shopper is shown in a different color. If inter-observation dissimilarities are computed using Euclidean distance on the raw variables, then the number of socks purchased by an individual will drive the dissimilarities obtained, and the number of computers purchased will have little effect. This might be undesirable, since (1) computers are more expensive than socks and so the online retailer may be more interested in encouraging shoppers to buy computers than socks, and (2) a large difference in the number of socks purchased by two shoppers may be less informative about the shoppers' overall shopping preferences than a small difference in the number of computers purchased.

Center the same data is shown, after scaling each variable by its standard deviation. Now the number of computers purchased will have a much greater effect on the inter-observation dissimilarities obtained.

Right the same data are displayed, but now the y-axis represents the number of dollars spent by each online shopper on socks and on computers. Since computers are much more expensive than socks, now computer purchase history will drive the inter-observation dissimilarities obtained.



Understanding the Results of Clustering

Clustering is often used as a data-exploratory analysis.

Understanding and Interpretation of the Results: domain knowledge, intuition and creativity

For the whiskey example, 12 clusters are resulted by cutting dendrogram at a certain height, here are two of them:

- Group A: `Aberfeldy, Glenugie, Laphroaig, Scapa`
- Group H: `Bruichladdich, Deanston, Fettercairn, Glenfiddich, Glen Mhor, Glen Spey, Glentauchers, Ladyburn, Tobermory`

We can simply look into the names of whiskeys in each clusters, which may make sense for whiskey-lovers/experts.

However such an approach does not make sense for, e.g., the customer clusters of a large retailer.

One can go further:

- Group A
  * Scotches: Aberfeldy, Glenugie, Laphroaig, Scapa
  * The best of its class: Laphroaig (Islay), 10 years, 86 points
  * Average characteristics: full gold; fruity, salty; medium; oily, salty, sherry; dry
- Group H
  * Scotches: Bruichladdich, Deanston, Fettercairn, Glenfiddich, Glen Mhor, Glen Spey, Glentauchers, Ladyburn, Tobermory
  * The best of its class: Bruichladdich (Islay), 10 years, 76 point
  * Average characteristics: white wyne, pale; sweet; smooth, light; sweet, dry, fruity, smoky; dry, light

Now those information is useful for whiskey retailers/shops and not-experts.

Two additional pieces of info:

Best whiskey in the class: from Jackson (1989) – unused in clustering

Average characteristics: extract from the centroid of each cluster (i.e. those with averages close to 1).

Using Supervised Learning to Generate Cluster Descriptions

A cluster centroid, in effect, reflects the average characteristics of the members in the cluster.

Those characteristics may be descriptive, represent the commonalities of the cluster members. But they do not tell how the clusters differ.

Basic Idea: Introduce a label (responsive) variable; each individual is given a label according to the cluster it belongs to. We then derive, for example, a decision tree, for 'classifying the clusters'.

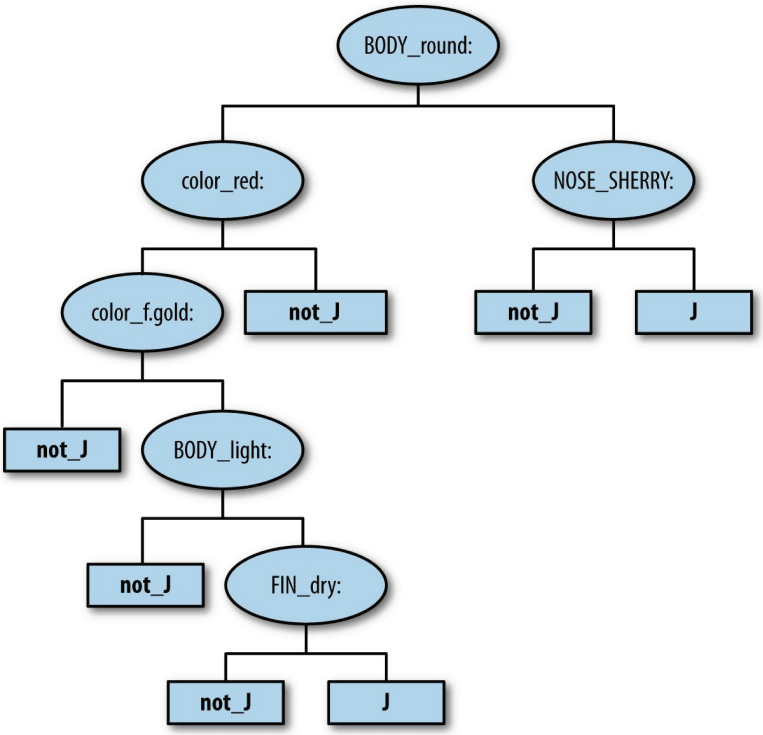
Two approaches: classify k-classes (i.e. one class per cluster), or k separate classification problems: each classify one cluster from the others.

We adopt the 2nd approach to whiskey example: to classify cluster J from the others:

- Group J
  * Scotches: Glen Albyn, Glengoyne, Glen Grant, Glenlossie, Linkwoo
    North Port, Saint Magdalene, Tamdhu
  * The best of its class: Linkwood (Speyside), 12 years, 83
  * Average characteristics: full gold; dry, peaty, sherry;
    light to medium, round; sweet; dry

An excerpt of the dataset looks like this:

```
0,0,0,...,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,J % Glen Grant
0,0,0,...,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,not_J % Glen K
0,0,0,...,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,not_J % Glen M
```



A decision tree for Cluste on Scotches data.

'No' for all branches on left,

'Yes' for all branches on right

Only two leaves labelled J

Two leaves labelled J:

```
1. (ROUND_BODY = 1) AND (NOSE_SHERRY = 1)

2. (ROUND_BODY = 0) AND (color_red = 0) AND (color_f.gold = 1) AND
(BODY_light = 1) AND (FIN_dry = 1)
```

Therefore, we may conclude: J cluster is distinguished by Scotches having either:

1. A round body and a sherry nose, or

2. A full gold (but not red) color with a light (but not round) body and a dry finish.

Two sets of descriptions for Cluster J:

- *Characteristic Description*: represented by the cluster centroid, describing typical characteristics of the cluster, ignoring whether the other clusters may share some of those characteristics

  Intragroup Commonalities

- *Differential Description*: derived from a decision tree, describing what differentiates this cluster from the others, ignoring some commonalities of the members within the cluster

  Intergroup Differences

Neither is inherently better — it depends on what you are using it for.

One more example: Credit line optimization

Cluster the existing customers based on similarity in their use of their cards, payment of their bill, and profitability of the company, leading to 5 clusters representing very different consumer behaviour (e.g. spending a lot but paying off the cards in full each month, spending a lot but keeping their balance near their credit limit).

Those different clusters can tolerate different credit lines.

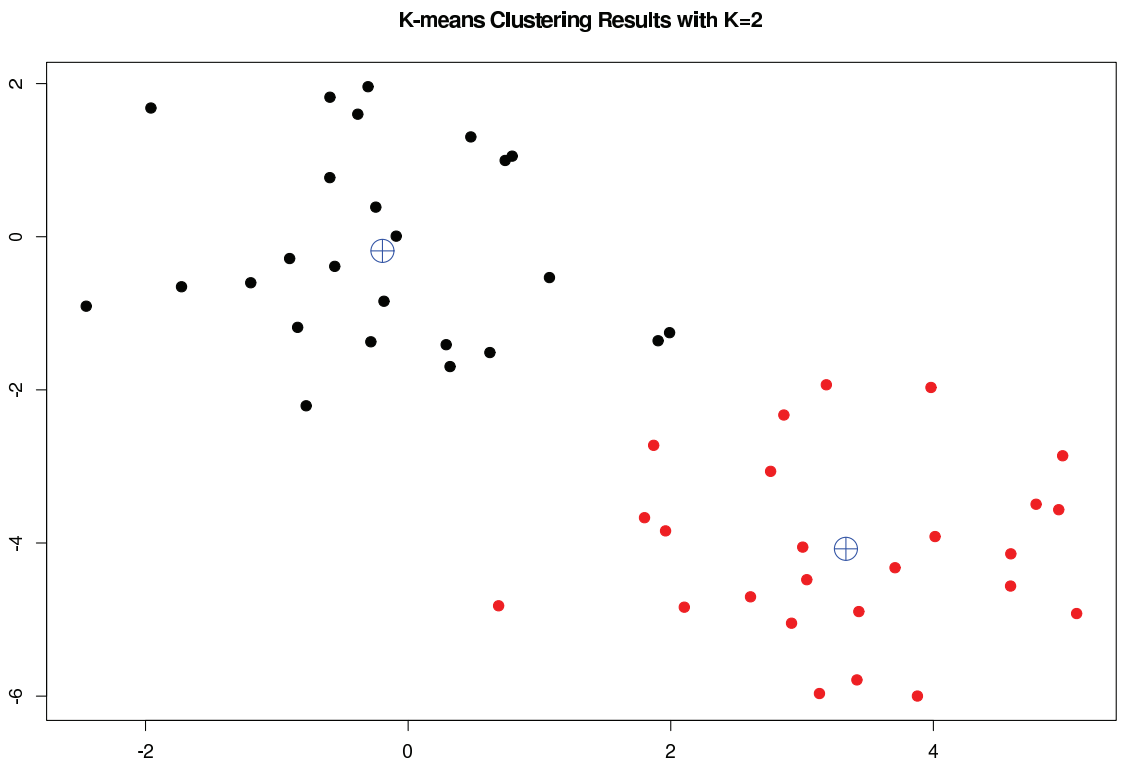
However the data used for clustering are not available for new customers to whom credit lines need to be assigned.

Use the data available at the time of credit approval to build a supervised learning model to classify customers into the 5 different clusters. This model then can be used to improve initial credit line decision.

Reference: Haimowitz, I., & Schwartz, H. (1997). Clustering and prediction for credit line optimization. In Fawcett, Haimowitz, Provost,& Stolfo (Eds.), AI Approaches to Fraud Detection and Risk Management, pp. 29-33. AAAI Press. Available as Technical Report WS-97-07.

In R function `kmeans` perform K-means clustering analysis. We start with a simple simulation example.

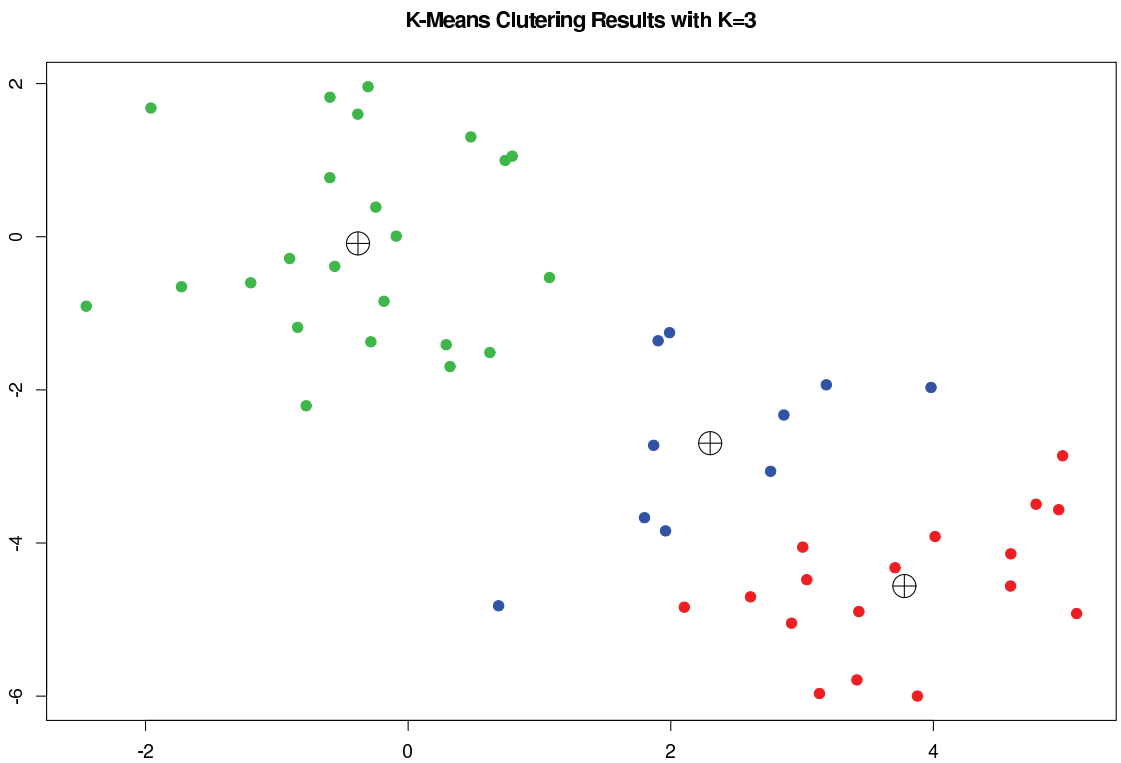
```
> x=matrix(rnorm(100), ncol=2) # 50x2 matrix containing indep N(0, 1) r.v.s.
> x[1:25,1]=x[1:25,1]+3
> x[1:25,2]=x[1:25,2]-4  # change the 1st 25 points to mean (3,-4)
> km.out=kmeans(x,2,nstart=20) # perform K-means with K=2, and 20 initial values
> km.out$cluster
 [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1
 [31] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> km.out$centers
        [,1]       [,2]
1 -0.1956978 -0.1848774
2  3.3339737 -4.0761910
> plot(x, col=(km.out$cluster), main="K-means Clustering Results with K=2", xlab="",
   ylab="", pch=20, cex=2)
> points(km.out$centers, col='blue', pch=10, cex=3) # adding two centroid points
```



What will happen if we set $K = 3$:

```
> km.out2=kmeans(x, 3, nstart = 20)
> km.out2
K-means clustering with 3 clusters of sizes 17, 23, 10
Cluster means:
        [,1]        [,2]
1  3.7789567 -4.56200798
2 -0.3820397 -0.08740753
3  2.3001545 -2.69622023

Clustering vector:
 [1] 1 3 1 3 1 1 1 3 1 3 1 3 1 3 1 3 1 1 1 1 1 3 1 1 1 2 2 2 2 2
 [31] 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 2 2 2 2

Within cluster sum of squares by cluster:
[1] 25.74089 52.67700 19.56137
 (between_SS / total_SS =  79.3 %)

> plot(x, col=(km.out2$cluster+1), main="K-Means Clustering Results with K=3",
     xlab="", ylab="", pch=20, cex=2)
> points(km.out2$centers, pch=10, cex=3)
```

Note. It is always a good idea to use multiple initial values, i.e. set `nstart`> 1.



We continue with dataset `customers` analysed in Chapter 6. It contains 8 attributes of 440 clients of a wholesale distributor: the first two attributes are `Channel` (i.e. Horeca or Retail) and `Region`. The other 6 variables are annual spendings on 6 categories of products. Now we use only the last 6 variables for clustering the 440 customers into, say, 5 clusters.

```
> > customers=read.csv("wholesaleCustomers.csv")
> dim(customers)
[1] 440   8
> customer6=customers[,3:8]
> km.custermers=kmeans(customer6, 5, nstart=20)
> table(Channel, km.customer$cluster, deparse.level = 2)
       km.customer$cluster
Channel   1   2   3   4   5
      1 195  13  83   0   7
      2  38   0  23   7  74
```

This is interesting, as Clusters 1, 2, 3 are dominated by Horeca customers while Clusters 4 and 5 are dominated by Retail customers. The percentages of Horeca customers in each clusters are:

```
> tab=table(Channel, km.customer$cluster, deparse.level = 2)
> for(i in 1:5) print(tab[1,i]/sum(tab[,i]))
[1] 0.8369099
```

```
[1] 1
[1] 0.7830189
[1] 0
[1] 0.08641975
```

This can be done more efficiently (i.e. avoiding the for-loop) as follow.

```
> tab.csum=apply(tab, 2, sum) # calculate sum for all columns of tab
> tab[1,]/tab.csum
        1         2         3         4         5
0.83690987 1.00000000 0.78301887 0.00000000 0.08641975
```

The wholesales distributor may provide different services to the 5 different clusters, according to their different spending profiles.

Also note those clusters tell little about customers' regions

```
> table(Region, km.customer$cluster, deparse.level = 2)
      km.customer$cluster
Region   1   2   3   4   5
     1  43   3  16   1  14
     2  23   1  11   1  11
     3 167   9  79   5  56
```

```
> table(Region)
Region
  1   2   3
 77  47 316
```

There are 316 customers in Region 3, and Clusters 1, 3, 5 have more customers from each of the three regions than the two small clusters.

Note. One can run K-means method based on other distance measures, check out `?kmeans`.

One also can use Mahalanobis distance in K-means clustering: this requires to standard the data first. For the above example, we use `customer6N` instead of `customer6`. See Chapter 6 for how the normalised data `customer6N` is defined.

Hierarchical clustering can be performed using function `hclust` together with function `cutree`. The input should be a distance matrix such as an output of function `dist`.

In the illustration below, we use rescaled 7 variables to cluster 440 customers.

```
> customersU=scale(customers)
    # make each columns of customers have variance 1
> hcC.customers=hclust(dist(customersU[,-2]), method="complete")
    # exclude Region in analysis. One may change complete to use
    # other linkage methods, check ?hclust
> hcC.index20=cutree(hcC.customers, 20) # cut tree with 20 terminal nc
            # can also cut tree at certain height, check ?cutree
> table(Channel, hcC.index20, deparse.level = 2)
       hcC.index5
```

```
Channel   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
      1   0 208  43   0   0  15   0   0   0  20   2   0   0   1   2
      2  89   0  10   1   6   0  23   2   5   3   0   1   1   0   0
Channel  16  17  18  19  20
      1   1   1   4   1   0
      2   0   0   0   0   1
> hcC.index10=cutree(hcC.customers, 10)
> table(Channel, hcC.index10, deparse.level = 2)
       hcC.index10
Channel   1   2   3   4   5   6   7   8   9  10
      1 275   2  17   0   0   0   2   1   1   0
      2 108   1   0  28   3   1   0   0   0   1
```

One can plot the tree via `plot(hcC.customers)`. However with 440 individuals, the tree is too big to be plotted on a piece of paper.

Chapter 8. Data Analytic Thinking: What Is a Good Model?

Fundamental concepts: What is desired from data analysis from business consideration; Expected value as a key evaluation framework; Comparative baseline models/methods.

Exemplary techniques: Evaluation metrics; Estimation costs and benefits; Calculating expected profit; Creating baseline methods for comparison.

Further Reading:

Provost and Fawcett (2013): Chapters 7 & 8.

Evaluation Classifiers

As discussed before, evaluation is a key step in data mining, which can be carried out using a 'holdout' data set or cross-validation, to detect/avoid overfitting.

However the accuracy, defined below, is simplistic and has some well-known problem

$$\text{accuracy} = \frac{\text{No. of correct decision made}}{\text{Total No. of decision made}}$$

Consider binary classification: two classes only labelled as '*Positive*' and '*Negative*'

Confusion Matrix

	Positive	Negative
Yes	No. true positive	No. false positive
No	No. false negative	No. true negative

True classes: Positive and Negative

Predicted Classes: Yes, it's positive, or, No, it's negative.

Note. *Negative* denotes the normal and often uninteresting status, and *Positive* denotes unusual ones.

The goal of classification is to sift through a large number of normal and uninteresting (i.e. negative) individuals in order to single out a relative small number of unusual (i.e. positive) ones.

Examples: looking for defrauded customers, checking for defective parts, targeting consumers who actually would respond to an ad.

When the classes are skewed with the ratio 999:1, a simple rule – always choose the majority class gives 99.9% accuracy. This is unlikely to be satisfactory!

Consider two classifiers for customer churn:

Classifier A: 80% accuracy

Classifier B: 64% accuracy

Is A better than B? — Not necessary!

They may use different test data.

In reality, the churn rate is only about 10%, now both A and B look bad, or?

We need to look into the data sets used by A and B carefully, as they may be quite different from each other, and also different from real data with different churn rates.

There are two types of errors: false positive, and false negative

Now test the two classifiers on 1000 customers with 500 positive (churn) and 500 negative (not churn). The results are as follows:

Classifier A

	churn	not churn
Y	500	200
N	0	300

Classifier B

	churn	not churn
Y	300	0
N	200	500

Classifier A: 200 false positive

Classifier B: 200 false negative

For application to real data with 10% churn rate, B achieves overall 96% accuracy while A achieves merely 64%.

Unequal costs should be associated with the two types of errors

Unequal Costs and Benefits

The costs for false positive and false negative are often different from each other.

The benefits for predicting true positive and true negative correctly are also different.

For customer churn example, the cost of giving a customer a retention incentive *unnecessarily* (a false positive) and that of losing a customer because no incentive was offered (a false negative) are different!

Note. This also applies to other data analysis problems such as regression: positive residuals and negative residuals may lead to very different business costs.

For example, a movie recommendation model predicts how many stars each customer would give to an unseen movie. For such a problem, the MSE or regression R^2 are no longer appropriate.

Another example is to use variance or STD as a measure for risk.

Questions to ask:

What is important for business?

What is the goal of analysis?

Are we assessing the results of data analysis appropriately given the actual goal?

Is the metric used meaningful? Is there a better one?

Expected Value: A Key Analytical Framework

Expected value computation: provides a framework in organizing the thinking about data-analytic problems into three stages:

1. the structure of the problem

2. the elements of the analysis that can be extracted from the data

3. the elements of the analysis that need to be acquired from other sources (e.g. business knowledge of subject matter experts)

An expected value may represent expected profits (to be maximized), expected losses (to be minimized).

General form of an expected value: Let $O_1, O_2, O_3, \cdots$ denote all the possible outcomes, $p_i = P(O_i)$ denote the probability of the occurrence of O_i, and V_i be the value when O_i occurs. Then the expected value is

$$EV = p_1 V_1 + p_2 V_2 + p_3 V_3 + \cdots .$$

1. Possible outcomes $O_1, O_2, O_3, \cdots$ are identified from a proper understanding the structure of the problem

2. Probabilities $p_1, p_2, p_3, \cdots$ are evaluated from data analysis

3. Values $V_1, V_2, V_3, \cdots$ are obtained from other sources.

Using Expected Value to Frame Classifier Use

Consider a targeted marketing problem, we assign each consumer a class of *likely responder* versus a class of *not likely responder*. (Then the resource should be spent on the individuals in the likely responder class.)

This is typically an unbalanced classification problem, as, for example, the response rate for an advertisement is small, or very small, say 1%.

Hence the predicting every one to NP yields accuracy 99%, suggesting no advertising at all!

Let $\mathbf{x}$ denote the feature variable of an individual, $p_r(\mathbf{x})$ be the probability to response. Suppose the profit from buying the product is $£100$, and the ad cost is $£1$. Then the expected value for this individual is

$$EV(\mathbf{x}) = (100 - 1)p_r(\mathbf{x}) - 1 \cdot (1 - p_r(\mathbf{x})) = 100p_r(\mathbf{x}) - 1.$$

Hence we should send the ad to this individual if $EV(\mathbf{x}) > 0$, which is

$$p_r(\mathbf{x}) > 0.01.$$

Using Expected Value to Evaluate Classifiers

Since each model will make some decisions better than the others, we need compare them collectively.

Models to be compared for a classification problems may include:

- Some baseline models (such as completely random classifiers)
- Bayes or Naive Bayes classifiers
- K-NN classifiers
- Decision trees
- Logistic regression
- Hand-crafted model suggested by subject matter experts

Aggregated expected value over the whole population

For a two-class classification problem with the total 110 individual, suppose the confusion matrix is

	p	n
Y	56	7
N	5	42

The rates for one individual falling in each cell are estimated by

$$p(Y,p) = 56/110 = 0.51, \quad p(Y,n) = 7/110 = 0.06,$$

$$p(N,p) = 5/110 = 0.05, \quad p(N,n) = 42/110 = 0.38.$$

To calculated the expected value or expected benefit, we need to know the benefit (or cost) for each of the above 4 cells.

Let us continue with the targeted AD example,

- A *false positive* occurs when we classify a consumer as a likely responder and therefore target her, but she does not respond. So the benefit is $b(Y,n) = -1$, or the cost is $c(Y,n) = 1$.
- A *false negative* is a consumer who was predicted not to be a likely responder (so was not offered the product), but would have bought it if offered. In this case, no money was spent and nothing was gained, so $b(N,p) = 0$.
- A *true positive* is a consumer who is offered the product and buys it. The benefit in $b(Y,p) = 100 - 1 = 99$
- A *true negative* is a consumer who was not offered a deal and who would not have bought it even if it had been offered. The benefit in this case is zero (no profit but no cost), so $b(N,n) = 0$.

The cost-benefit matrix is

	p	n
Y	99	-1
N	0	0

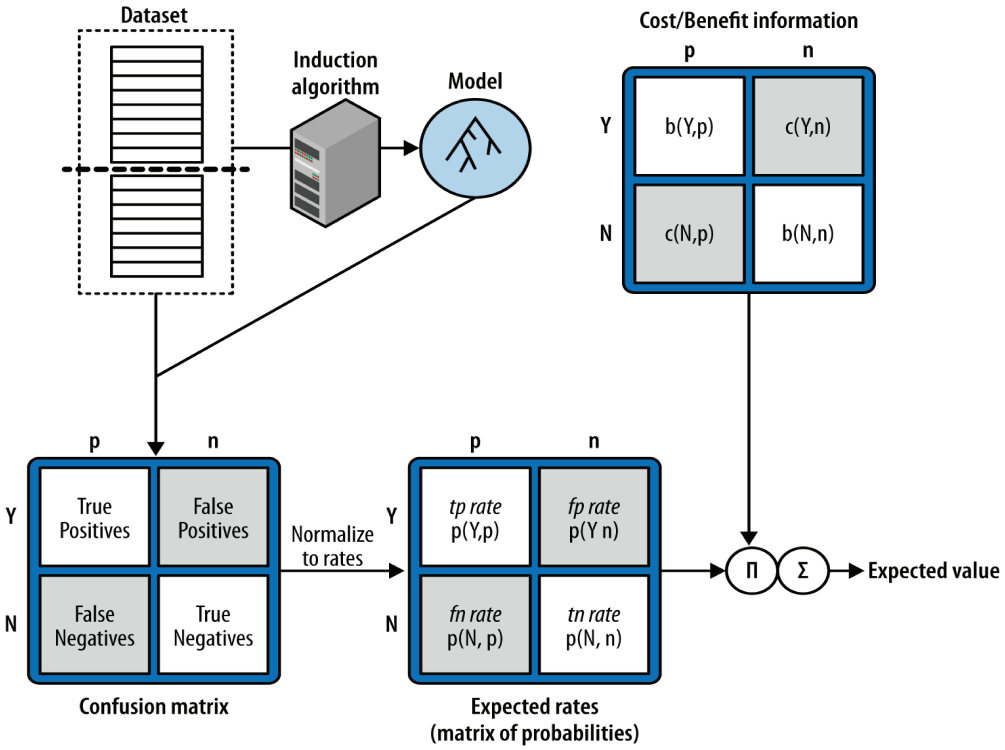
Thus the expected benefit is

$$p(Y,p)\cdot b(Y,p)+p(Y,n)\cdot b(Y,n)+p(N,p)\cdot b(N,p)+p(N,n)\cdot b(N,n)$$
$$=0.51\times 99-0.06\times 1=50.43.$$

Remark. (i) Avoid 'double count' by, for example, using $b(N,p)=-99$. (This can be detected by calculating the increased benefit from moving one individual from (N,p) to (Y,p).)

(ii) *R computing*: Let `B` be a cost-benefit matrix, `P` be a probability matrix. The expected benefit is `sum(B*P)`.

(iii) Extension to the cases with more than two classes is obvious.



Other often used metrics for 2-class classification problems

Let *TP, FP, TN and FN* denote, respectively, the number of true positive, false positive, true negative and false negative. Then $n = TP + FP + TN + FN$, $TP + FN$ is the number of true positive individuals, and $TN + FP$ is the number of true negative individuals, and

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

- True positive rate: $TP/(TP + FN)$ – rate being correct when the truth is positive
- False negative rate: $FN/(TP + FN)$ – rate being incorrect when the truth is positive

True negative rate and false positive rate are defined in the similar manner

In text classification,

- Precision: $TP/(TP + FP)$
- Recall: $TP/(TP + FN)$ i.e. true positive rate
- F-measure: $F_1 = 2\frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$, i.e. the harmonic mean of Precision and Recall

Ideally choose a model to maximize both Precision and Recall, which is often impossible. A compromise is to maximize F_1.

ROC Graphs (Receiver Operating Characteristics Graphs)

For each classifier, plot *True Positive Rate* (also called hit rate)

$$TPR = TP/(TP + FN) \qquad \text{(the bigger the better)}$$

against *False Positive Rate* (also called false alarm rate)

$$FPR = FP/(FP + TN) \qquad \text{(the smaller the better)}$$

Some characteristics of an ROC graph (see the graph on next page):

- Main diagonal corresponds to random classifiers. For example, E at (0.6, 0.6) classifies each individual to *Positive* with probability 0.6.

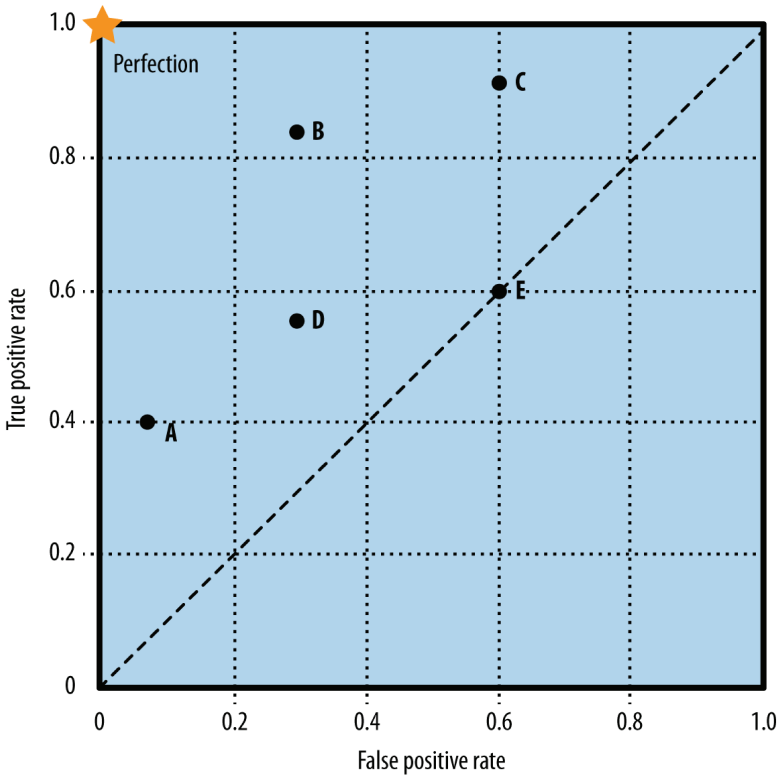
- Any admissible classifier should be above the main diagonal.

- The (0, 1) indicates the perfect classifier which identify every individuals correctly.

- In the graph on next page, A is more conservative than B, which in turn is more conservative than C.

- B is a better classifier than D, as it has a higher TPR and the equal FNR.

Note. ROC graphs do not take into account of costs/benefits. Nevertheless it is a more appropriate measure that the (overall) misclassification rate. For example, it rules out the decision of taking no action in an advertising campaign with a population with responsive rate 1%, as such a decision entails point (0, 0) on an ROC graph.



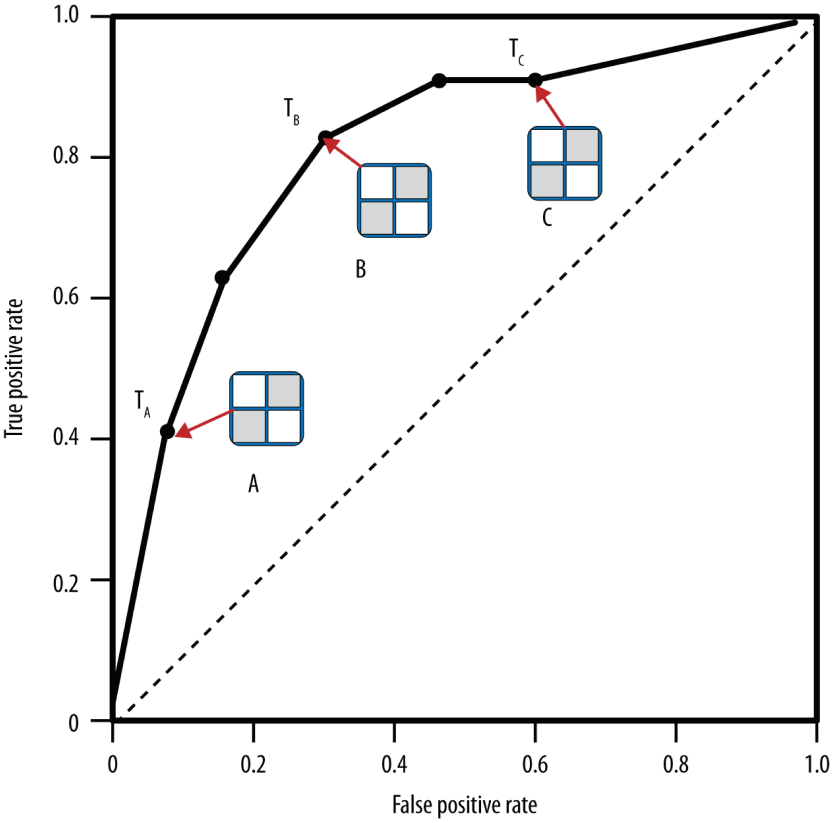
ROC curves

For any classifier for two-class problem, let $P(Y|\mathbf{X})$ be the predictive probability for *positive* given feature $\mathbf{X}$, and $P(N|\mathbf{X}) = 1 - P(Y|\mathbf{X})$ be the probability to predict *negative*.

Classification decision: predict *positive* if $P(Y|\mathbf{X}) > t$, where $t \in [0, 1)$ is a threshold.

The threshold value t is determined by, e.g. by using the expected value. The conventional practice: $t = 0.5$

For each $t \in [0, 1)$, the decision rule $P(Y|\mathbf{X}) > t$ leads to a different confusion matrix, therefore a point in the ROC graph. Let t vary from 0 to 1, it generates an **ROC curve**.



ROC curve for one classifier: each point on the curve corresponds to a different threshold value t and a different confusion matrix.

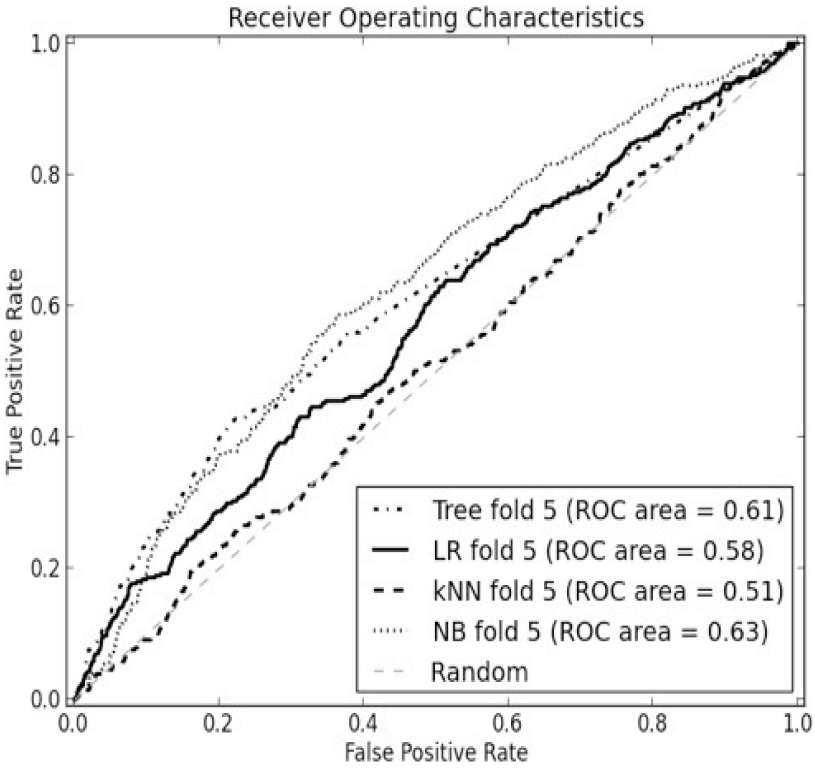
We can compare different classifiers by comparing their ROC curves.

Drawing a vertical line (all the points on the line have the same FPR), the classifier at the top achieves the maximum TPR

Drawing a horizontal line (all the points on the line have the same TPR), the classifier at the most left achieves the minimum FPR

If we prefer one index to compare different classifier, use the **Area Under the ROC curve (AUC)**.

For random classifier, AUC=0.5. So any reasonable classifiers should have AUC>0.5 at least.



ROC curves of the classifiers on one fold of cross-validation for the churn problem

LR: logistic regression
kNN: K-nearest neighbours
NB: Naive Bayesian

Compare Performance with Some Baseline Models

It is important to consider carefully what would be a reasonable baseline for a given problem. The selected model should perform better than the baseline model according to a appropriate performance metric.

Below are some possible baseline models.

For classification problems,

- completely random model (i.e. assign to all classes with the equal probability)
- majority classifier

For regression problems,

mean $\bar{y}$

the nearest neighbour → NN

For time series,

the previous value

average of today over many years

For predicting how many stars a customer would give to a particular film: a convex combination of the average ranking from other customer and the average ranking from this customer over different films.

→ Supervised learning.

The (naive) Bayesian methods are other candidates which should be included in the comparison

For simplification, consider a classification problem for which Y is the label of classes and $\mathbf{X} = (X_1, \cdots, X_m)$ are predictors/features taking discrete values.

It follows the Bayesian formula that

$$p(Y = j|\mathbf{X}) = \frac{p(Y = j, \mathbf{X})}{p(\mathbf{X})}.$$

Thus for given $\mathbf{X}$, and the Bayes classifier sets $Y = \ell$, where ℓ satisfies the inequality $p(Y = \ell|\mathbf{X}) \geq \max_j p(Y = j|\mathbf{X})$, or equivalently

$$p(Y = \ell, \mathbf{X}) \geq \max_j p(Y = j, \mathbf{X}).$$

Lift: $p(Y = j|\mathbf{X}) \big/ p(Y = j)$

Note both $p(Y = j, \mathbf{X})$ (also $p(Y = j)$ and $p(\mathbf{X})$) can be estimated using the relatively frequencies from the training data.

The naive Bayes assumes the conditional independence:

$$p(\mathbf{X}|Y) = \prod_{i=1}^{m} p(X_i|Y).$$

Then

$$p(Y = j|\mathbf{X}) = \frac{p(Y = j, \mathbf{X})}{p(\mathbf{X})} = \frac{p(Y = j)}{p(\mathbf{X})} p(\mathbf{X}|Y = j) = \frac{p(Y = j)}{p(\mathbf{X})} \prod_{i=1}^{m} p(X_i|Y =$$

Thus for given $\mathbf{X} = (X_1, \cdots, X_m)$, the naive Bayes classifier sets $Y = \ell$ with ℓ satisfying the following inequality

$$p(Y = \ell) \prod_{i=1}^{m} p(X_i|Y = \ell) \geq \max_j p(Y = j) \prod_{i=1}^{m} p(X_i|Y = j)$$

- R packages `ROCR` and `pROC` produce ROC curves

- How to draw ROC curves for the classification problems with more than two classes?

  1. Multiple two-class classification: one class versus all other classes.

  2. For more general approaches, see

     Krzanowski, W.K. and Hand, D.J. (2009). *ROC Curves for Continuous Data*. CRC Press.

R package: ROCR

Two important functions: `prediction` and `performance`

- First apply `prediction` to data:
  `prediction.output = prediction(pred, labels, label.ordering = NULL`

  `pred` consists of predicted measures (eg. estimated probabilities, odds), can be the outputs of a decision tree, a logistic regression and etc. `labels` are true binary (i.e. 0 or 1) labels of classes. Both `pred` and `labels` can be a vector, matrix or data.frame, but they must be of the same size.

- To produce ROC graph,
  ```
  roc1=performance(prediction.output, measure="tpr", x.measure="fpr"
  plot(roc1, col=as.list(1:10))
  abline(a=0,b=1) # adding a straight line y=a+bx
  ```

- To calculate AUC (area under the curve):
  ```
  auc1=performance(prediction.output, measure="auc")
  ```

  ```
  auc1@y.values
  ```

- To produce the overall accuracy curve against cut-off probability:
  ```
  acc1=performance(prediction.output, measure="acc")
  plot(acc1, col=as.list(1:10)) # p is No. of ACC curves
  ```

More information on `ROCR`:

`https://www.r-bloggers.com/a-small-introduction-to-the-rocr-package/`

Example: re-visit the spam email example in Chapter 3.

4601 emails: 2788 true mails, and 1812 spam mails

57 quantitative predictors

```
> spamData=read.table("spam.txt")
> dim(spamData)
[1] 4601   58    # last colunm is labels: 1 for spam, 0 for true mail
> spamNames=read.table("spamNames.txt")  # names of 58 variables
> dim(spamNames)
[1] 58  1
> names(spamData)=spamNames[,1]
> names(spamData)
 [1] "make"       "address" "all"       "3d"       "our"      "over"       "remove"
 [8] "internet"   "order"   "mail"      "receive"  "will"     "people"     "report"
[15] "addresses"  "free"    "business"  "email"    "you"      "credit"     "your"
[22] "font"       "000"     "money"     "hp"       "hpl"      "george"     "650"
[29] "lab"        "labs"    "telnet"    "857"      "data"     "415"        "85"
[36] "technology" "1999"    "parts"     "pm"       "direct"   "cs"         "meeting"
[43] "original"   "project" "re"        "edu"      "table"    "conference" ";"
[50] "("          "["       "!"         "$"        "#"        "CAPAVE"     "CAPMAN"
[57] "CAPTOT"     "LABEL"
```

In the 54-th line of file "spamNames.txt", # is written as "#" !!!

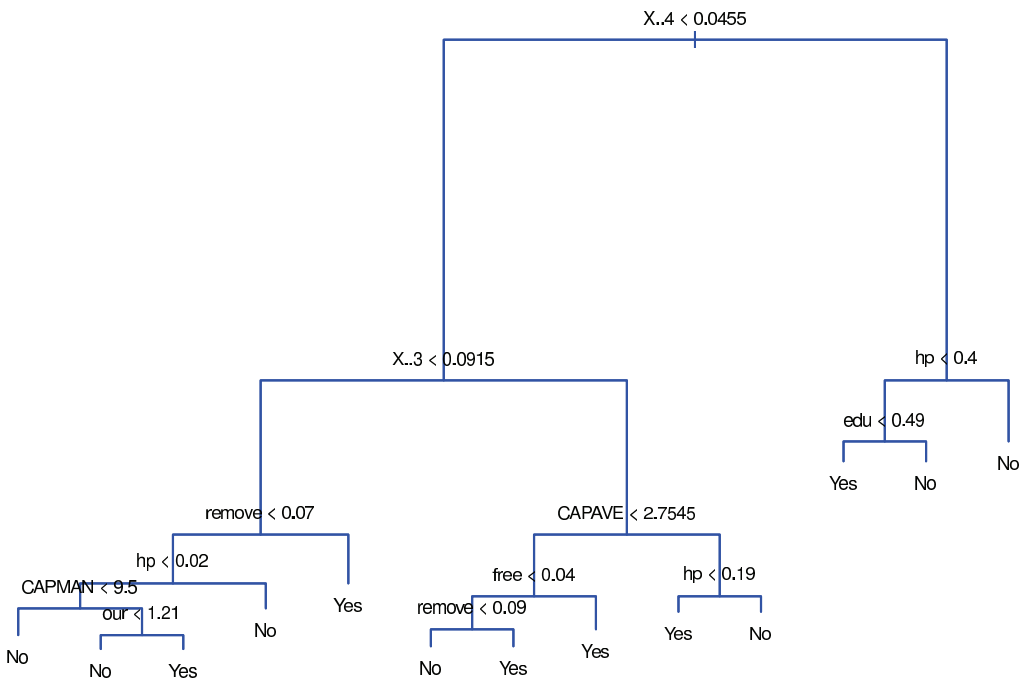
```
> attach(spamData)
> spam=rep("No", 4601);  spam[LABEL==1]="Yes"
> spamData1=data.frame(spamData, spam)
> train=sample(1:4601, 3065, replace=F) # 3065 emails as training sample
> spamTest=spamData1[-train,]
> dim(spamTest)
[1] 1536   59     # 1536 emails for testing
> library(tree)
> tree1=tree(spam~.-LABEL, data=spamData1, subset=train)
Error: unexpected ',' in "tree1=tree(label~,"
```

The error message is not clear. Checking it with google, we find out that we have used some illegal names such as '415', '[' in R!

```
> make.names(names(spamData1), unique=T, allow_=T)
 [1] "make"       "address" "all"      "X3d"      "our"      "over"       "remove"
 [8] "internet"   "order"   "mail"     "receive"  "will"     "people"     "report"
[15] "addresses"  "free"    "business" "email"    "you"      "credit"     "your"
[22] "font"       "X000"    "money"    "hp"       "hpl"      "george"     "X650"
[29] "lab"        "labs"    "telnet"   "X857"     "data"     "X415"       "X85"
[36] "technology" "X1999"   "parts"    "pm"       "direct"   "cs"         "meeting"
[43] "original"   "project" "re"       "edu"      "table"    "conference" "X."
[50] "X..1"       "X..2"    "X..3"     "X..4"     "X..5"     "CAPAVE"     "CAPMAN"
```

```
[57] "CAPTOT"     "LABEL"    "spam"
> names(spamData1)=make.names(names(spamData1), unique=T, allow_=T)
> detach(spamData)  # NECESSARY, otherwise R will be confused with same names
                    # from the two data sets
> attach(spamData1)
> tree1=tree(spam~.-LABEL, data=spamData1, subset=train) # exclude LABEL!
> summary(tree1)

Classification tree:
tree(formula = spam ~ . - LABEL, data = spamData1, subset = train)
Variables actually used in tree construction:
 [1] "X..4" "remove" "X..3" "hp"  "CAPMAN" "our" "CAPAVE" "free" "george" "edu"
Number of terminal nodes:  13
Residual mean deviance:  0.4804 = 1466 / 3052
Misclassification error rate: 0.08418 = 258 / 3065
> tree2=cv.tree(tree1, FUN=prune.misclass)
> tree2$size
 [1] 13 11  8  7  6  5  3  2  1
> tree2$dev
 [1]  311  323  323  325  337  368  430  624 1178
> plot(tree1, col="blue", lwd=2)
> text(tree1, pretty=0)
```

Note `tree1` with 13 terminal nodes is the CV-selected model, no need to prune further. **Note `X..4`=$, `X..3`=!**

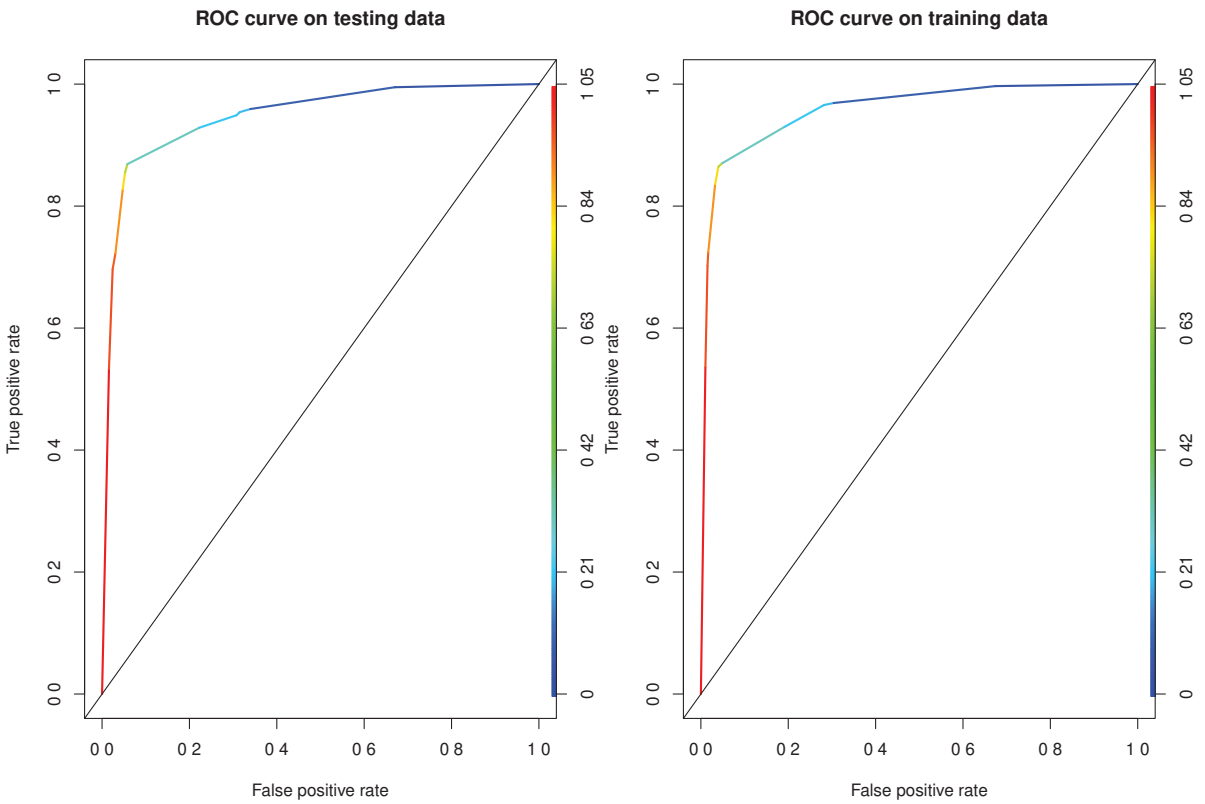


Now we calculate its predicted probabilities for 'Yes' (spam) on both training and testing data.

```
> predT=predict(tree1, spamTest, type="vector") # check ?predict.tree
> predTest.tree=predT[,2] # predicted probabilities for 'Yes' (spam)
> length(predTest.tree)
[1] 1536
> summary(predTest.tree)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.00000 0.04288 0.14237 0.40017 0.88931 0.94986
> predT=predict(tree1, spamData1[train,], type="vector")
> predTrain.tree=predT[,2]
> length(predTrain.tree)
[1] 3065
```

To produce ROC curves,

```
> library(ROCR)
> prediction.treeTest=prediction(predTest.tree, LABEL[-train])
> prediction.treeTrain=prediction(predTrain.tree, LABEL[train])
> rocTest=performance(prediction.treeTest, measure="tpr", x.measure="fpr")
> rocTrain=performance(prediction.treeTrain, measure="tpr", x.measure="fpr")
> par(mfrow=c(1,2))
> plot(rocTest, lwd=2, colorkey=T, colorize=T, main="ROC curve on testing data")
> abline(0,1)    # You may like to check ?plot.performance
> plot(rocTrain, lwd=2, colorkey=T, colorize=T,, main="ROC curve on training data")
> abline(0,1)
```



Option `colorize=T` adds the color code to the ROC curve according to the cut-off probability for 'Yes' (spam), `colorkey=T` adds the color key for the cut-off probability vertically on the right. Note that when the cut-off probability increases, both TPR and FPR decrease. This makes an ROC graph much more imformative.

In practice we need to choose the cut-off probability. A convenient choice is 0.5. However a more meaningful approach is to consider costs/benefits for different scenarios and choose the one which maximize (or minimize) the expected benefit (or cost).

To count for the importance of not filtering out genuine emails, we define the cost/benefit matrix as follows:

	Email	Spam
No	0	-1
Yes	-4	0

```
> predLab=ifelse((predTrain.tree>=0.5), "Yes","No")
> confusion=table(predLab, spam[train], deparse.level=2)
> confusion
```

```
       spam[train]
predLab    0    1
    No  1799  170
    Yes   88 1008
> CB=matrix(c(0,-1,-4,0), nrow=2, byrow=T)
> CB
     [,1] [,2]
[1,]    0   -1
[2,]   -4    0
> sum(CB*confusion)/sum(confusion) # compute expected benifit
[1] -0.17031
```

The expected benefit for using this filter is -0.170. To reduce the number of false positives, we should increase the cut-off probability. Below we find the value of the cut-off probability, which maximizes the expected benefit.

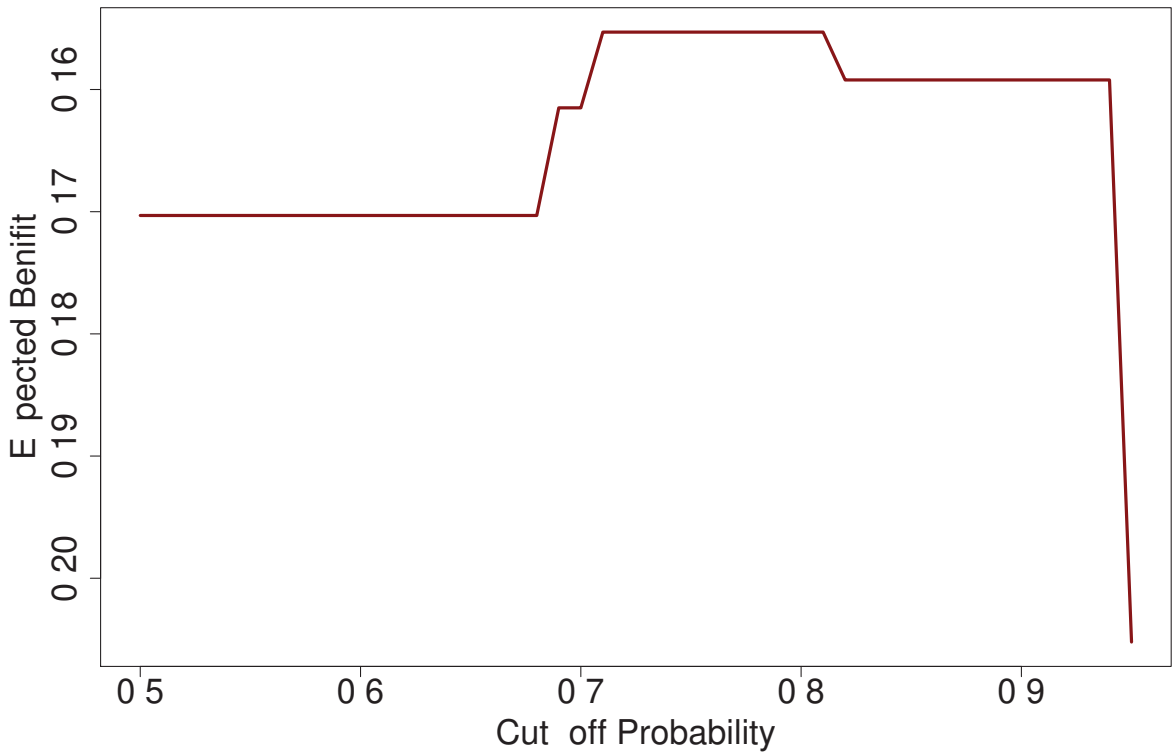
```
> alpha=seq(0.5, 0.95, 0.01)
> eBenifit=vector(length=length(alpha))
> for(i in 1:length(alpha)) {
        predLab=ifelse((predTrain.tree>=alpha[i]), "Yes","No")
        confusion=table(predLab, LABEL[train],deparse.level=2)
        eBenifit[i]=sum(CB*confusion)/sum(confusion)
 }
> plot(alpha, eBenifit, type="l", lwd=3, col="darkred", xlab="Cut-off Probability",
```

```
     ylab="Expected benifit")
> alpha[eBenifit==max(eBenifit)]
    # find value(s) of alpha which maximize expected benifit
 [1] 0.71 0.72 0.73 0.74 0.75 0.76 0.77 0.78 0.79 0.80 0.81
```

Based on this, we can use the cut-off probability between 0.71 and 0.81. Now we test the performace with 0.5 and 0.75:

```
> predLab=ifelse((predTest.tree>=0.5), "Yes", "No")
> confusion=table(predLab, spam[-train])
> sum(CB*confusion)/sum(confusion)
[1] -0.2363281
> predLab=ifelse((predTest.tree>=0.75), "Yes", "No")
> confusion=table(predLab, spam[-train])
> sum(CB*confusion)/sum(confusion)
[1] -0.2063802
```

With the cut-off probability at 0.75, the expected cost is lower than that at 0.5.



To calculate the areas under curves,

```
> performance(prediction.treeTrain, measure="auc")@y.values
[[1]]
[1] 0.9523803  # area under curve for traning sample
> performance(prediction.treeTest, measure="auc")@y.values
[[1]]
[1] 0.9374082 # area under curve for testing sample
```

Now we construct the K-NN classifiers, with $K = 3$ or 5, based on the training data, and then check the performance on the testing data. We also compare them with the tree classifier obtained above.

We only use the variables selected in the tree model to define the distances.

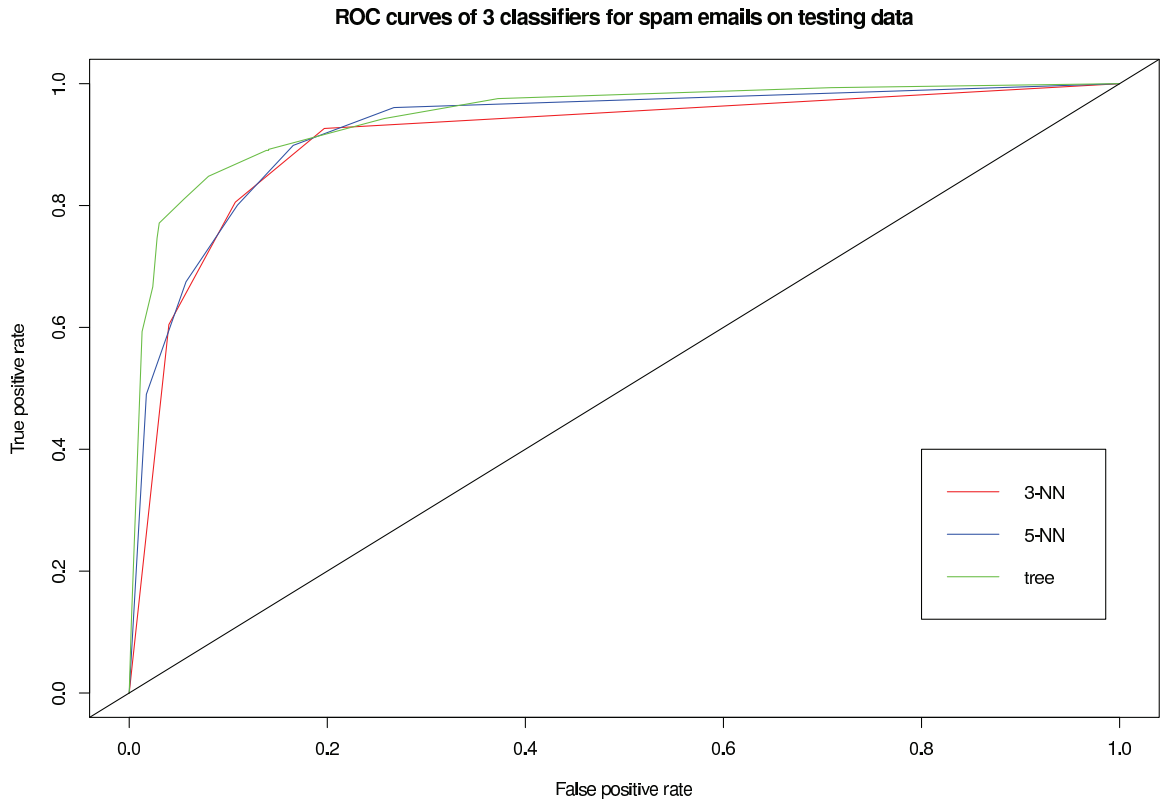
```
> library(dplyr)
> X=select(spamData1, X..4, X..3, remove, hp, CAPMAN, our, CAPAVE, free, edu)
> dim(X)
[1] 4601    9
> Xtrain=X[train,]
> Xtest=X[-train,]
> dim(Xtrain)
```

```
[1] 3065    9
> dim(Xtest)
[1] 1536    9
> summary(Xtrain)
      X..4              X..3             remove             hp
 Min.   :0.00000   Min.   : 0.0000   Min.   :0.0000   Min.   : 0.0000
 1st Qu.:0.00000   1st Qu.: 0.0000   1st Qu.:0.0000   1st Qu.: 0.0000
 Median :0.00000   Median : 0.0000   Median :0.0000   Median : 0.0000
 Mean   :0.07362   Mean   : 0.2652   Mean   :0.1099   Mean   : 0.5281
 3rd Qu.:0.05000   3rd Qu.: 0.3060   3rd Qu.:0.0000   3rd Qu.: 0.0000
 Max.   :5.30000   Max.   :19.1310   Max.   :7.2700   Max.   :20.8300
     CAPMAN             our             CAPAVE             free
 Min.   :   1.00   Min.   :0.0000   Min.   :   1.000   Min.   : 0.0000
 1st Qu.:   6.00   1st Qu.:0.0000   1st Qu.:   1.571   1st Qu.: 0.0000
 Median :  15.00   Median :0.0000   Median :   2.250   Median : 0.0000
 Mean   :  53.02   Mean   :0.3156   Mean   :   5.081   Mean   : 0.2504
 3rd Qu.:  43.00   3rd Qu.:0.3800   3rd Qu.:   3.657   3rd Qu.: 0.0900
 Max.   :9989.00   Max.   :9.0900   Max.   :1021.500   Max.   :20.0000
      edu
 Min.   : 0.0000
 1st Qu.: 0.0000
 Median : 0.0000
 Mean   : 0.1737
 3rd Qu.: 0.0000
 Max.   :10.0000
```

Since data are sparse (i.e. many zeros), we use the correlation based distance measure $1 - \text{Corr}$.

```
> D=-cor(t(Xtest), t(Xtrain))+1 # correlation based distances between rows of Xtest
         # and rows of Xtrain, cor(X1, X2) returns correlations between
         # columns of two matrices X1, X2. Hence transpose
> dim(D)
> [1] 1536 3065
> inDex=matrix(nrow=1536, ncol=5)
> for(i in 1:1536) inDex[i,]=sort.int(D[i,], index.return = T)$ix[1:5]
 # inDex[i, ] contains the row indices of the 5 NN of Xtest[i,] among
 # all rows in Xtrain.   Check ?sort.int
> predKNN=matrix(nrow=1536, ncol=2)
> Y=LABEL[train]
> for(i in 1:1536) predKNN[i,]=c(mean(Y[inDex[i,1:3]]), mean(Y[inDex[i,]]))
> summary(predKNN)
       V1               V2
 Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.3333   1st Qu.:0.2000
 Median :0.3333   Median :0.2000
 Mean   :0.2582   Mean   :0.2939
 3rd Qu.:0.3333   3rd Qu.:0.4000
 Max.   :1.0000   Max.   :1.0000
> pred3=prediction(data.frame(predKNN, predTest.tree), data.frame(LABEL[-train],
    LABEL[-train],LABEL[-train]))
```

```
> roc3=performance(pred3, measure ="tpr", x.measure ="fpr")
> dev.off()
> plot(roc3, col=as.list(c("red","blue","green")), main="ROC curves of 3 classifiers
         for spam emails on testing data")
> legend(0.8, 0.4, c("3-NN","5-NN", "tree"), col=c("red","blue","green"), lty=c(1,1,
> abline(0,1)
```



Based on the ROC graph, the tree model seems to be the best though three classifiers do not differ that much.

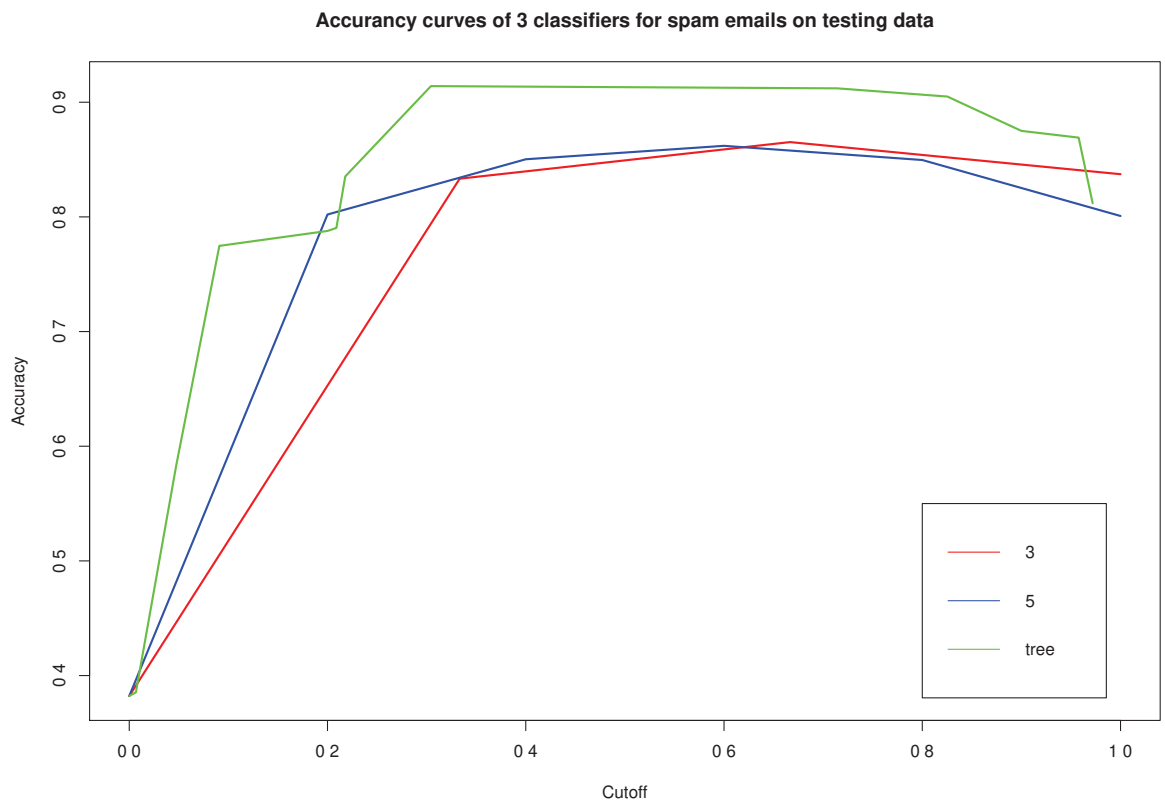
```
> performance(pred3, measure ="auc")@y.values
[[1]]
[1] 0.9107125
[[2]]
[1] 0.9266129
[[3]]
[1] 0.9473464
```

According to AUC, the tree model is the best classifier among the three, while the K-NN classifier with $K = 5$ performs better than that with $K = 3$.

To produce over all accurancy rate (i.e. 1 - misclassification rate) curves

```
> acc3=performance(pred3, measure="acc")
> plot(acc3, lwd=2, col=as.list(c("red","blue","green")), main="Accurancy curves
      of 3 classifiers for spam emails on testing data")
> legend(0.8, 0.55, c("3-NN","5-NN", "tree"), col=c("red","blue","green"),
   lty=c(1,1,1))
> detach(spamData1) # do this upon the completion of a project: a good practice
```

The clean R scripts for this example are collected in the file 'spamEmail.r' available in Moodle.



Chapter 9. Market-Basket Analysis

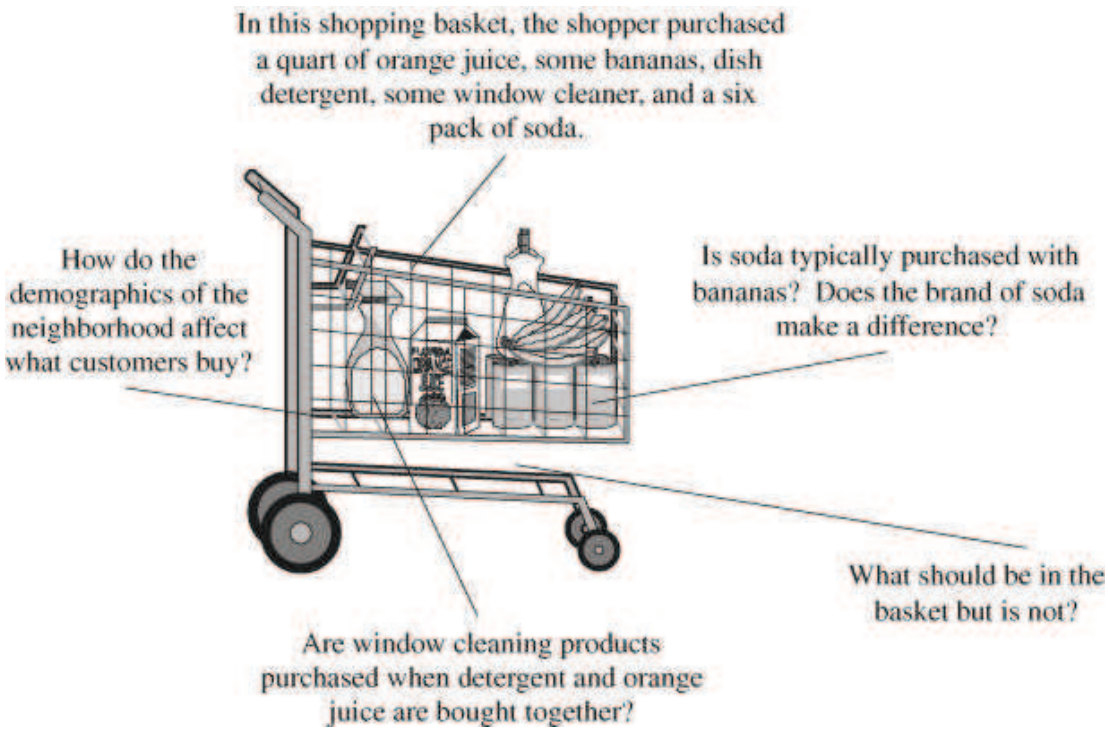
Goal: identify co-occurring items.

Techniques: quantify so-called support, confidence and lift etc important concepts in basket analysis, the Apriori Algorithm

Potential applications: stocking shelves, cross-marketing in sales promotions, catalog design, and consumer segmentation based on buying patterns.

Further Reading:

Provost and Fawcett (2013): Chapter 9.



Each customer purchases a different set of products, in different quantities, at different times.

Market basket analysis uses the information about what customers purchase to provide insight into who they are and why they make certain purchases. Market basket analysis provides insight into the merchandise by telling us which products tend to be purchased together and which are most amenable to promotion.

This information is actionable: it can suggest new store layouts; it can determine which products to put on special sales; it can indicate when to issue coupons, and so on.

When this data can be tied to individual customers through a loyalty card or Web site registration, it becomes even more valuable.

Other applications: Items purchased on a credit card, such as rental cars and hotel rooms, provide insight into the next product that customers are likely to purchase.

Optional services purchased by telecommunications customers (call waiting, call forwarding, DSL, speed call, and so on) help determine how to bundle these services together to maximize revenue.

Banking services used by retail customers (money market accounts, investment services, car loans, and so on) identify customers likely to want other services.

Unusual combinations of insurance claims can be a sign of fraud and can spark further investigation.

Medical patient histories can give indications of likely complications based on certain combinations of treatments.

Suppose a supermarket sells p items in total, has recorded n transactions.

Denote each transaction with a p-vector with components 0 or 1, i.e.

$$x_{ij} = 1 \text{ if } i\text{-th transaction contains a purchase of } j\text{-th item,}$$
$$0 \text{ otherwise}$$

Thus i-th transaction is represented by $\mathbf{x}_i = (x_{i1}, \cdots, x_{ip})'$, a vector with most components equal to 0.

Let $\mathcal{J} \subset \{1, \cdots, p\}$, indicating a subset of the items sold in the supermarket.

We call $\mathcal{J}$ an 'item set', the number of the elements in $\mathcal{J}$ is called the 'size' of $\mathcal{J}$.

There are in total $2^p - 1$ item sets.

For $p = 10000$, 2^p can be regarded as infinity!

Support for index set $\mathcal{J}$:

$$T(\mathcal{J}) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j \in \mathcal{J}} x_{ij}$$

i.e. $T(\mathcal{J})$ is the proportion of the transactions which contain all the items in set $\mathcal{J}$.

Note. $T(\mathcal{J})$ is the (estimated) probability for the event that the items in the set $\mathcal{J}$ are purchased together.

Basket Analysis: to identify all the set $\mathcal{J}$ with $T(\mathcal{J}) \geq t$, where $t \in (0, 1)$ is a constant.

Typically t is taken as a small constant such as 0.05 for big supermarket data.

The problem looks simple, at least conceptually. However it is computationally infeasible to search over all possible item sets, as typically $p \approx 10^4$ and $n \approx 10^8$ for big supermarkets.

The Apriori Algorithm. It makes the search feasible if the number of the item sets satisfying the condition $T(\mathcal{J}) > t$ is a small fraction of 2^p (i.e. t cannot be too small).

A simple fact: If item set $\mathcal{K}$ is a subset of item set $\mathcal{J}$, $T(\mathcal{K}) \geq T(\mathcal{J})$.

Key Idea of the Apriori Algorithm:

- The 1st pass over the data computes the support of all single-item sets, and discards those with support smaller than t.

- The 2nd pass computes the support of all item sets of size 2 that are formed from pairs of the single item sets surviving the 1st pass, and discards those with support smaller than t.

- For $m \geq 3$, the m-th pass computes the support of all item sets of size m that are formed from a surviving item set of size $m-1$ and a surviving single item set, and discards those with support smaller than t.

There are many additional tricks to increase the speed and convergence in the Apriori Algorithm; see Agrawal *et al.*(1995). *Fast discovery of association rules*, Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, Cambridge, MA.

The Apriori algorithm represents one of the major advances in data mining technology.

Association rules: defined for each high support item set $\mathcal{J}$ with size at least 2.

Partition $\mathcal{J}$ into two non-overlapping subsets $\mathcal{A}$ and $\mathcal{B}$, i.e. $\mathcal{J} = \mathcal{A} \cup \mathcal{B}$ and $\mathcal{A} \cap \mathcal{B}$ is an empty set.

Confidence (or **Predictability**): The confidence of $\mathcal{B}$ from $\mathcal{A}$ is defined as

$$C(\mathcal{A} \Rightarrow \mathcal{B}) = T(\mathcal{A} \cup \mathcal{B}) \Big/ T(\mathcal{A}).$$

Note. $C(\mathcal{A} \Rightarrow \mathcal{B})$ is the (estimated) conditional probability of $\mathcal{B}$ given $\mathcal{A}$. $T(\mathcal{A} \cup \mathcal{B})$ is an estimate for $P(\mathcal{A}\mathcal{B}) \neq P(\mathcal{A} \cup \mathcal{B})$.

Lift: The lift of of $\mathcal{B}$ from $\mathcal{A}$ is defined as

$$L(\mathcal{A} \Rightarrow \mathcal{B}) = C(\mathcal{A} \Rightarrow \mathcal{B}) \Big/ T(\mathcal{B}).$$

Note. (i) $L(\mathcal{A} \Rightarrow \mathcal{B})$ is the ratio of probability of the event that all items in $\mathcal{A}$ and $\mathcal{B}$ are purchased together to the product of the probabilities of two events: (i) all items in $\mathcal{A}$ are purchased together, (ii) all items in $\mathcal{B}$ are purchased together

(ii) $T(\mathcal{B})$ is also called 'expected confidence'. Thus the lift is the confidence divided by the expected confidence.

(iii) $L(\mathcal{A} \Rightarrow \mathcal{B}) = L(\mathcal{B} \Rightarrow \mathcal{A})$

For example, let $\mathcal{J} = \{\texttt{peanutbutter}, \texttt{jelly}, \texttt{bread}\}$ with support 0.03. Let $\mathcal{A} = \{\texttt{peanutbutter}, \texttt{jelly}\}$ and $\mathcal{B} = \{\texttt{bread}\}$. Suppose $T(\mathcal{A}) = 0.04$. Then the confidence is

$$C(\mathcal{A} \Rightarrow \mathcal{B}) = 0.03/0.04 = 75\%.$$

Hence when peanut butter and jelly were purchased, 75% of the time bread was also purchased.

If $T(\mathcal{B}) = 0.4$, i.e. the 40% purchases include bread, the lift for $\{\texttt{bread}\}$ by $\{\texttt{peanut butter, jelly}\}$ is

$$L(\mathcal{A} \Rightarrow \mathcal{B}) = 0.75/0.4 = 1.875.$$

Example. Consider $n = 9404$ questionnaires filled out by shopping mall customers in the San Francisco Bay Area (Impact Resources, Inc., Columbus OH, 1987). Here we only use answers to the first 14 questions, relating to demographics. These questions are listed below.

The data consist of a mixture of ordinal and (unordered) categorical variables, many of the latter having more than a few values. There are many missing values.

After removing observations with missing values, each ordinal predictor was cut at its median and coded by two dummy variables; each categorical predictor with k categories was coded by k dummy variables. This resulted in a 6876$\times$50 matrix of 6876 observations on 50 dummy variables (i.e. each of 6876 questionnaires is represented by a vector with 50 components being either 0 or 1).

Feature	Demographic	# Values	Type
1	Sex	2	Categorical
2	Marital status	5	Categorical
3	Age	7	Ordinal
4	Education	6	Ordinal
5	Occupation	9	Categorical
6	Income	9	Ordinal
7	Years in Bay Area	5	Ordinal
8	Dual incomes	3	Categorical
9	Number in household	9	Ordinal
10	Number of children	9	Ordinal
11	Householder status	3	Categorical
12	Type of home	5	Categorical
13	Ethnic classification	8	Categorical
14	Language in home	3	Categorical

With $t = 0.1$, the 6288 item sets with size not greater than 5 were found. Understanding this large set of rules is itself a challenging data analysis task.

Figure 14.2 the relative frequency of each dummy variable in the data (top) and the association rules (bottom). Prevalent categories tend to appear more often in the rules, for example, the first category in language (English). However, others such as occupation are under-represented, with the exception of the first and fifth level.

FIGURE 14.2. *Market basket analysis: relative frequency of each dummy variable (coding an input category) in the data (top), and the association rules found by the Apriori algorithm (bottom).*

Three examples of Association Rule found by the Apriori algorithm

- Support 25%, confidence 99.7% and lift 1.03.

$\mathcal{A}$	$\mathcal{B}$
number in household =1 number of children = 0	language in home = English

- Support 13.4%, confidence 80.8%, and lift 2.13

$\mathcal{A}$	$\mathcal{B}$
language in home = English householder status = own occupation = professional/managerial	income $\geq$ \$ 40,000

- Support 26.5%, confidence 82.8% and lift 2.15.

$\mathcal{A}$	$\mathcal{B}$
language in home = English income < $ 40,000 number of children = 0	education $\notin$ {college graduate, graduate study}

Association Rules

There is an intuitive appeal to an association rule because it expresses how tangible products and services group together.

While association rules are easy to understand, they are not always useful.

- Actionable Rules: contains high-quality, actionable information, such as

  Wal-Mart customers who purchase Barbie dolls have a 60 percent likelihood of also purchasing one of three types of candy bars.

- Trivial Rules: already known by anyone at all familiar with the business, may simply the reflection of previous marketing campaigns.

Customers who purchase paint buy paint brushes; oil and oil filters are purchased together, as are hamburgers and hamburger buns, and charcoal and lighter fluid.

Customers who purchase maintenance agreements are very likely to purchase large appliances.

- Inexplicable Rules: have no explanation and do not suggest a course of action.

  When a new hardware store opens, one of the most commonly sold items is toilet bowl cleaners – Discovered for new store openings by a large hardware company, intriguing, giving little insight into consumer behavior or the merchandise or suggest further actions

FAMOUS RULES: BEER AND DIAPERS – A famous story in late 1980s when computers were just getting powerful enough to analyze large volumes of transaction data.

Somewhere in the midwest of America, the fact that beer and diapers are selling together was discovered in the transaction data.

This immediately sets marketing minds in motion to figure out what is happening. A flash of insight provides the explanation: beer drinkers do not want to interrupt their enjoyment of televised sports, so they buy diapers to reduce trips to the bathroom – No, that is not it!

The more likely story: families with young children are preparing for the weekend, diapers for the kids and beer for Dad. Dad probably knows that after he has a couple of beers, Mom will change the diapers.

This is a powerful story. Setting aside the analytics, what can a retailer do with this information? There are two competing views. One says

to put the beer and diapers close together, so when one is purchased, customers remember to buy the other one. The other says to put them as far apart as possible, so the customer must walk by as many stocked shelves as possible, having the opportunity to buy yet more items. The store could also put higher-margin diapers a bit closer to the beer, although mixing baby products and alcohol would probably be unseemly.

The story was debunked on 16 April 1998 in an article in *Forbes* Magazine called Beer-Diaper Syndrome

Note. Basket analysis may apply to the data collected at different levels, instead of supermarket transaction data only.

Case Study: Spanish or English

A chain of supermarket in Texas use the summarized data to investigate the differences in shopping patterns between Hispanics and non-Hispanics communities

Business problem: should this chain of supermarkets advertise the same products in Spanish as in English?

Data: The accumulated weekly sales of all products in different supermarket stores. In addition, the percentages of different ethnic groups in the catchment area for each store are also available.

Initial analysis found the association rule: the higher the percentage of African-American population, the lower the Hispanic population, and vice versa — Not interesting!

Rephrased business question: What are the differences in products sold in stores with high Hispanic catchment area versus in a low Hispanic catchment area?

This leads to the division of the stores into three groups: *High Hispanic*, *Mixed* and *Not very Hispanic*. Only use the data from High Hispanic and Not very Hispanic stores

Let $\mathcal{A} = \{\texttt{High Hispanic store}\}$ and $\mathcal{B} = \{\texttt{Not very Hispanic store}\}$.

Define *Hispanicity Preference* for each product $\mathcal{I}$ as

$$C(\mathcal{A} \Rightarrow \mathcal{I}) - C(\mathcal{B} \Rightarrow \mathcal{I}),$$

then sort different products according this Hispanicity Preference score.

Different purchase patterns were discovered. For example, non-Hispanics tend to prefer beef which Hispanics prefer pork; non-Hispanics prefer potato chips and French fries as snacks whereas Hispanics prefer corn chips as snacks.

Ads for the 4th July picnics: hamburgers and potato chips in English, and perhaps sausages and Doritos corn chips in Spanish.

The Apriori algorithm is implemented in R package `arules`. Data set `AdultCUI` within the package contains 48,842 responses for a questionnaire with 15 questions. Since each question will be represented by several items (i.e. binary variables), it is necessary to have a structure which can deal large amounts of sparse binary data in an efficient manner. See the example at the bottom of the help menu under `?AdultCUI`.

More detailed information on `arules`, with a data example illustration, can be found at
`https://cran.r-project.org/web/packages/arules/vignettes/arules.pdf`

Chapter 10. Representing and Mining Text

Fundamental concepts: preparation and representation of text data for mining

Exemplary techniques: Bag of words, TFIDF scores, n-grams, stemming, named entity extraction, topic models

Text data are extremely common nowadays, largely due to Internet which has become a ubiquitous channel of communication.

One important challenge is to represent each text data point (i.e. a document) as a numerical vector such that the data mining tools are become directly applicable.

The basic idea is also helpful in dealing with other types of non-numerical data.

Further Reading:

Provost and Fawcett (2013): Chapter 10

Why Text is Important? – It is everywhere!

Medical records, consumer complaint logs, product inquiries, and repair records are all in the form of text, for communication between people.

Internet is the new media: most of it still in the form of text – personal web pages, Twitter feeds, email, Facebook status updates, product descriptions, blog postings etc

Google and Bing are based on massive amounts of text-oriented data science.

Exploiting this vast amount of data requires converting text to the format which is meaningful to computers, i.e. a vector consisting of numerical attributes.

Why Text is Difficult?

- *Unstructured*: no uniform structure across different texts. Each text has its own free-form sequence of words, length, number of paragraphs, symbols, tables and figures.

- *Dirty*: some documents may be written ungrammatically, with misspell words, or words together, abbreviate unpredictably, and punctuate randomly.

- *Ambiguity*: different words share the same meaning, or the same words mean differently in different contexts.

  Texts are intended for human consumption, *context* is important. The same words or statements may mean different things in different context. It can be difficult to evaluate any particular word

or phrase here without taking into account the entire context.

> "The first part of this movie is far better than the second. The acting is poor and it gets out-of-control by the end, with the violence overdone and an incredible ending, but it's still fun to watch."

In this movie review excerpt, it is not clear if the overall sentiment is positive or negative, or if the word *incredible* is used positively or negatively?

Text must undergo serious preprocessing before it can be used for data mining

Document: one piece of text (regardless its length or contents)

Corpus: a collection of documents concerned.

Term or *Token*: a word, a phrase, or several connected words.

Bag of Words – a basic tools for text data representation

Treat every document as just a collection of individual words, ignoring grammar, word order, sentence structure, and punctuation.

This is a very simple approach, inexpensive to generate, and tends to work well for many tasks.

However some preprocessing is necessary:

- *Case-normalization*: make every word in lower-case

  `iPhone`, `iphone` and `IPHONE` are treated as one word

- *Stemming*: remove suffixes

  verbs like `announces`, `announced` and `announcing` are all reduced to `announc`

change noun plurals to singular, e.g. `directors` is recorded as `director`

- *Stopwords*: such as `and, a, an, of, on, at`. Those words are very common and tend to occur in all documents.

  For some applications (but not the information retrieval!), one may also exclude words which occur too rare, say, under 3% of the documents in the corpus

  On the other hand, the words occurring in most documents are not useful either for, e.g. classification and clustering, should be removed for those applications.

After the above preprocessing, every remaining word is a possible feature. There are several ways to present the value for each feature.

1. *Binary*: each word is a token with value 1 if token occurs in the document, and 0 otherwise.

   Each document is represented by the set of words contained in it, represented by a long vector consisting of 1 and 0. The length of the vector is the total number of words contained in all the documents in the corpus.

2. *Term Frequency*: using the word count (frequency) in the document instead of just 1 or 0.

   An obvious drawback: longer documents tend to produce larger TF scores.

   The TF may be divided by the total number of words in the document

3. *TFIDF*: The TFIDF value of a term t in a given document d is defined as

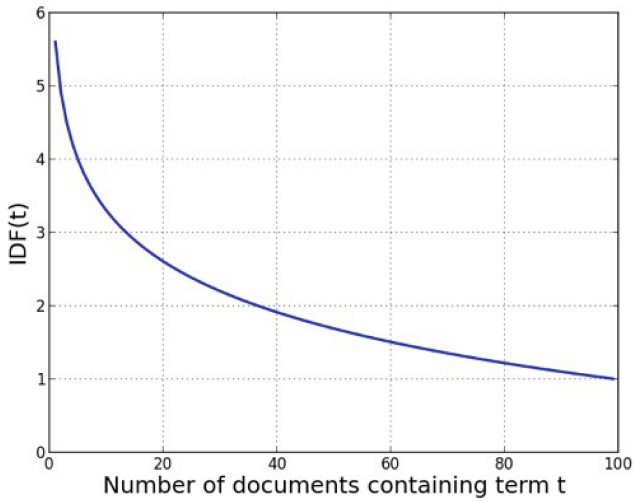
$$\text{TFIDF} = \text{TF (term frequency)} \times \text{IDF (inverse document frequency)}$$

$$\text{TF(t, d)} = \text{No. of times of word t occurring in document d}$$

$$\text{IDF(t)} = 1 + \log\left(\frac{\text{Total number of documents}}{\text{Number of documents containing word t}}\right)$$

Term counts within the documents form the TF values for each term, and the document counts across the corpus form the IDF values.

IDF boosts rare terms.



IDF of a term t within a corpus of 100 documents

Since the number of features is often excessively large. Feature selection is often necessary, which can be carried out by imposing minimum and maximum thresholds of term counts, and/or using a measure such as information gain to rank the terms by importance so that low-gain terms can be culled.

The bag-of-words text representation approach treats words in a document as independent terms of the document by assigning values to each term. TFIDF is a very commonly used, based on frequency and rarity. But it could be binary, term frequency, with normalization or without.

Experiment with different representations to see which produces the best results.

Example: Jazz Musicians

Data: Excerpts of the biographies from Wikipedia for 16 jazz musicians.

- `Charlie Parker`

  `Charles "Charlie" Parker, Jr., was an American jazz saxophonist an composer. Miles Davis once said, "You can tell the history of jaz in four words: Louis Armstrong. Charlie Parker." Parker acquire the nickname "Yardbird" early in his career and the shortened form "Bird", which continued to be used for the rest of his life, inspi the titles of a number of Parker compositions, ···`

- `Duke Ellington`

  `Edward Kennedy "Duke" Ellington was an American composer, pianist, and bigband leader. Ellington wrote over 1,000 compositions. In the opinion of Bob Blumenthal of The Boston Globe, "in the century since his birth, there has been no greater composer, American or`

otherwise, than Edward Kennedy Ellington.'' A major figure in the history of jazz, Ellington's music stretched into various other ge including blues, gospel, film scores, popular, and classical. ···

- Miles Davis

  Miles Dewey Davis III was an American jazz musician, trumpeter, ba and composer. Widely considered one of the most influential music of the 20th century, Miles Davis was, with his musical groups, at the forefront of several major developments in jazz music, includi bebop, cool jazz, hard bop, modal jazz, and jazz fusion. ···

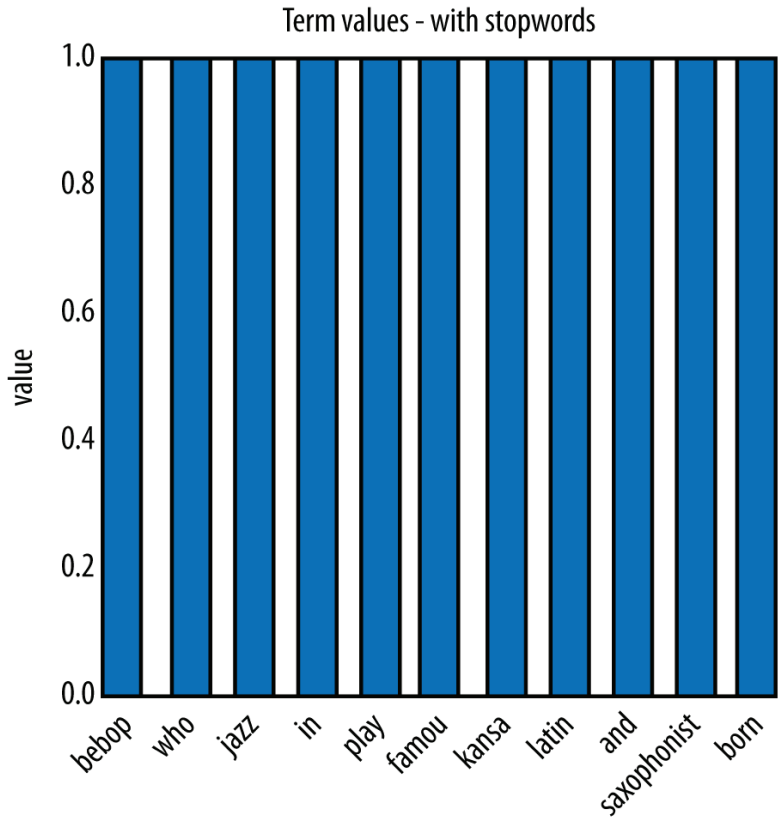
  ......

For this small corpus with $n = 16$, its vocabulary are large with $p \approx 2000$ after stemming and stopword removal. Applying the Bags of Words technical above with TFIDF scores, we translate each biography into a p-vector.

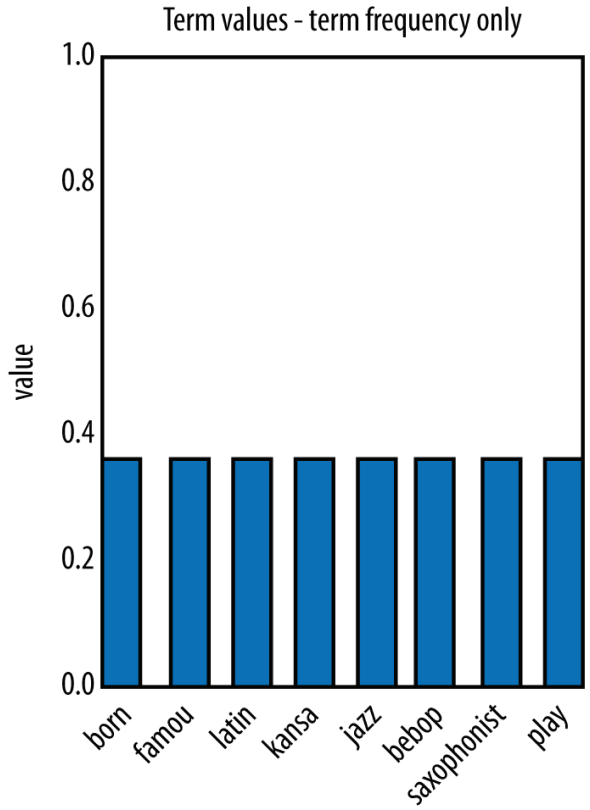
To illustrate its usefulness, suppose a search engine received a query:

Famous jazz saxophonist born in Kansas who played bebop and latin.

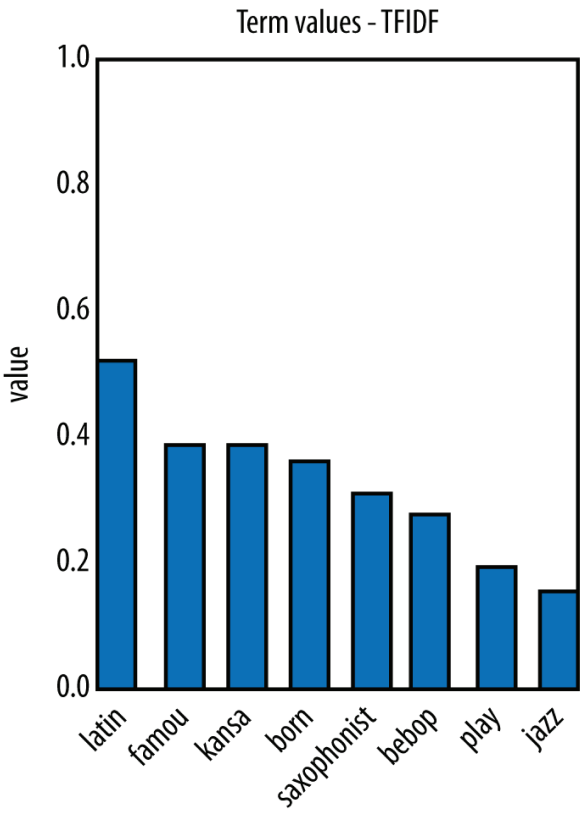
It treats the query exactly as a document to process it using the Bag-of-Words techniques.



Representation of the query 'Famous jazz saxophonist born in Kansas who played bebop and latin' after stemming



Representation of the query 'Famous jazz saxophonist born in Kansas who played bebop and latin' after stop-word removal and term frequency normalization



Final TFIDF representation of the query 'Famous jazz saxophonist born in Kansas who played bebop and latin.'

The IDF scores were calculated based on 16 biographies in the corpus.

Using the correlation-based measure,

$$\rho(\mathbf{x},\mathbf{y}) = \sum_i x_i y_i \Big/ \sqrt{\sum_i x_i^2 \sum_j y_j^2}$$

the similarity between the query and each of the 16 Jazz musicians' biography was calculated.

Musician	Similarity	Musician	Similarity
Charlie Parker	0.135	Count Basie	0.119
Dizzie Gillespie	0.086	John Coltrane	0.079
Art Tatum	0.050	Miles Davis	0.050
Clark Terry	0.047	Sun Ra	0.030
Dave Brubeck	0.027	Nina Simone	0.026
Thelonius Monk	0.025	Fats Waller	0.020
Charles Mingus	0.019	Duke Ellington	0.017
Benny Goodman	0.016	Louis Armstrong	0.012

Charlie Parker is the closest match. He in fact is a saxophonist born in Kansas and who played the bebop style of jazz. He sometimes combined other genres, including Latin, a fact that is mentioned in his biography.

Beyond Bag of Words

The basic bag of words approach is relatively simple, requires no linguistic analysis. It performs surprisingly well on a variety of tasks.

But some further improvements are required for many applications.

k-gram Sequences

To take into account of the order of words, take k adjacent words as a term.

For example, the sentence '`The quick brown fox jumps`' will generate terms {`quick, brown, fox, jump, quick-brown, brown-fox, fox-jump`} in a bag-of-words with 2-gram sequences

The advantage of k-gram is obvious when particular phrases are significant but their component words may not be. For example, the tri-gram `exceed-analyst-expectation` is more meaningful than the 3 individual words.

However it increases the number of attributes substantially, demanding more storage and computing/searching power.

Named Entity Extraction

Many text-processing toolkits include a named entity extractor, to extract phrases annotated with terms like *person* or *organization*.

This knowledge has to be learned from a large corpus, or coded by hand.

Some extractors may have particular areas of expertise, such as industry, government, or popular culture.

Example: Mining News Stories to Predict Stock Price Movement

Background: Companies make and announce decisions of mergers, new products, earnings projections, and so forth. Investors read these news stories, possibly change their beliefs about the prospects of the companies involved, and trade stock accordingly. Then stock prices change.

Ideally we would like predict in advance and with precision the change in a company's stock price based on the stream of news. In reality there are many complex factors involved in stock price changes, many of which are not conveyed in news stories.

Goal: to mine the news to recommend interesting news stories.

A piece of news is interesting if it will likely result in a significant change in a stock's price.

Assumptions:

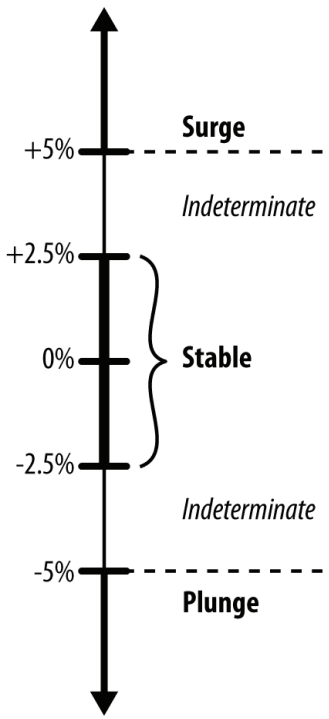
1. Only the changes in price on the same day are considered, as too difficult to predict the impact in the future.

2. Simplify stock price movements into two categories: `change` and `no change`, as predicting the exact changes are too difficult. (Here the direction of change is ignored.)

3. Only count for relatively large changes, ignoring the subtlety of small fluctuations.

4. Only news stories mentioning a specific stock will affect that stock's price.

This is inaccurate of course, and is a simplification to make the analysis easier.



BPercentage change in price, and corresponding label.

No change = stable

change = {surge, plunge}

Data. Two separate time series: the stream of news stories (text documents), and a corresponding stream of daily stock prices in 1999, for the stocks listed on the New York Stock Exchange and NASDAQ.

About 36,000 news stories.

For example, to see what news stories are available about Apple Computer, Inc., see the corresponding Yahoo!Finance page. Yahoo! aggregates news stories from a variety of sources such as Reuters, PR Web, and Forbes.

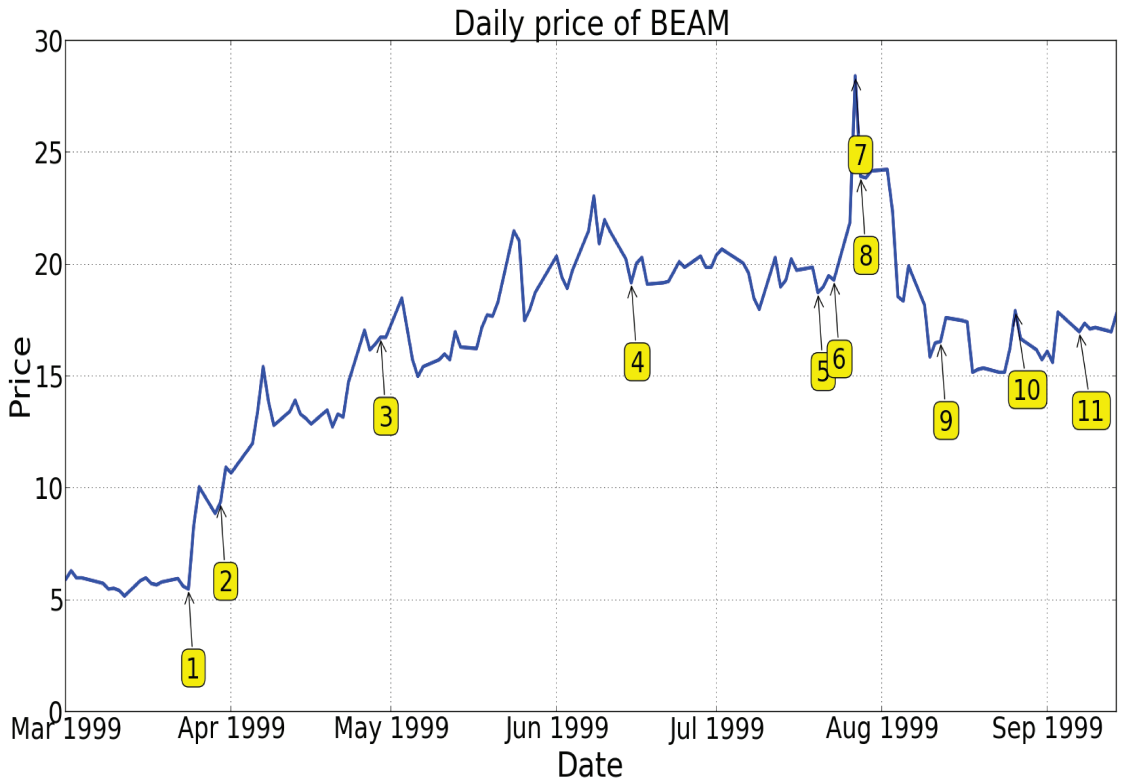
The new stories contain many miscellaneous materials: date and time, the news source, stock symbols, links to other sites, as well as background material not strictly germane to the news.

Here is one news story:

```
1999-03-30 14:45:00
WALTHAM, Mass.--(BUSINESS WIRE)--March 30, 1999--Summit Technology,
Inc. (NASDAQ:BEAM) and Autonomous Technologies Corporation
(NASDAQ:ATCI) announced today that the Joint Proxy/Prospectus for
Summit's acquisition of Autonomous has been declared effective by the
Securities and Exchange Commission. Copies of the document have been
mailed to stockholders of both companies. "We are pleased that these
proxy materials have been declared effective and look forward to the
shareholder meetings scheduled for April 29," said Robert Palmisano,
Summit's Chief Executive Officer.
```

Each such story is tagged with the stock mentioned.

Graph of stock price of Summit Technologies, Inc., (NASDAQ:BEAM) annotated with news story summaries.



1 Summit Tech announces revenues for the three months ended Dec 31, 1998 were $22.4 million, an increase of 13%.
2 Summit Tech and Autonomous Technologies Corporation announce that the Joint Proxy/Prospectus for Summit‘s acquisition of Autonomous has been declared effective by the SEC.
3 Summit Tech said that its procedure volume reached new levels in the first quarter and that it had concluded its acquisition of Autonomous Technologies Corporation.
4 Announcement of annual shareholders meeting.
5 Summit Tech announces it has filed a registration statement with the SEC to sell 4,000,000 shares of its common stock.
6 A US FDA panel backs the use of a Summit Tech laser in LASIK procedures to correct nearsightedness with or without astigmatism.
7 Summit up 1-1/8 at 27-3/8.
8 Summit Tech said today that its revenues for the three months ended June 30, 1999 increased 14% ...
9 Summit Tech announces the public offering of 3,500,000 shares of its common stock priced at $16/share.
10 Summit announces an agreement with Sterling Vision, Inc. for the purchase of up to six of Summit‘s state of the art, Apex Plus Laser Systems.
11 Preferred Capital Markets, Inc. initiates coverage of Summit Technology Inc. with a Strong Buy rating and a 12-16 month price target of $22.50.

News is Messy

- News comprises a wide variety of stories, including earnings announcements, analysts' assessments, market commentary, SEC filings, financial balance sheets, and so on. Companies are mentioned for many different reasons and a single document may actually comprise multiple unrelated news blurbs of the day.
- Stories come in different formats, some with tabular data, some in multiparagraphs. Much of the meaning is imparted by context.
- Stock tagging is not perfect, tends to be overly permissive, such that stories are included in the news feed of stocks that were not actually referenced in the story.

Data Preprocessing

Each stock has an opening (at 9:30am EST) and closing (at 4pm EST) price for each day.

To classify each day into `'change'`, `'no change'` or not classified, let

$$\text{PercenC} = 100 \times \frac{\text{(Price at 4pm) - (Price at 10am)}}{\text{(Price at 10am)}}$$

If $|\text{PercenC}| \geq 5$, `'change'`

If $|\text{PercenC}| < 2.5$, `'no change'`

Why use prices at 10am?

News also occurs off trading hours, and fluctuations near the opening hours can be erratic. Therefore in addition, define the change between days

$$\text{PercenC} = 100 \times \frac{\text{(Price at 10am) - (Price at 4pm yesterday)}}{\text{(Price at 4am yesterday)}}$$

The news stories require more care!

Stories without timestamps are discarded.

Stories mentioning two stocks or more are discarded.

Each story is aligned with the correct stock at correct trading day/window.

Stories corresponding to unclassified trading days/windows are discarded.

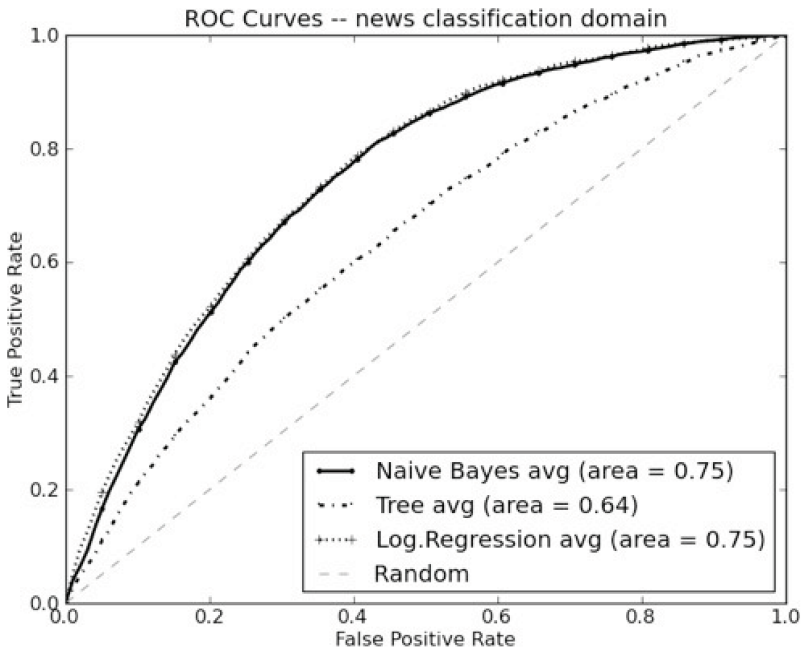
Remaining stories are gained a label `'change'` or `'no change'`.

Features are extracted from each of those stories using *Bag-of-Words* with 2-grams, after case-normalization, stemming, and stopword-removing

Finally, there are about 16,000 usable tagged stories, with about 75% with label `no change` and 25% with label `change` (actually 13% for `surge` and 12% for `plunge`).

Results

As the goal is modest, i.e. to identify the news stories which are associated to substantial stock price changes, no cost and benefit analysis here and no expected value calculation either.



Average from ten-fold cross-validation, using change as the positive class and no change as the negative class.

1. Predictive signal of news stories is indicated by the 'bowing out' of the curves above the diagonal line (random classifiers). The AUCs are greater 0.5 substantially.

2. Logistic regression and Naive Bayes perform similarly, whereas the classification tree (Tree) is considerably worse.

3. No classifiers (with any threshold values) are close to the perfection point (0, 1).

Below are the words (or stems) in the 'Bag of Words' which are most informative (i.e. with smallest conditional entropies):

```
alert(s,ed), architecture, auction(s,ed,ing,eers), average(s,d),
award(s,ed), bond(s), brokerage, climb(ed,s,ing), close(d,s),
comment(ator,ed,ing,s), commerce(s), corporate, crack(s,ed,ing),
cumulative, deal(s), dealing(s), deflect(ed,ing), delays, depart(s,ed)
department(s), design(ers,ing), economy, econtent, edesign, eoperate,
esource, event(s), exchange(s), extens(ion,ive), facilit(y,ies),
gain(ed,s,ing), higher, hit(s), imbalance(s), index, issue(s,d),
late(ly), law(s,ful), lead(s,ing), legal(ity,ly), lose, majority,
merg(ing,ed,es), move(s,d), online, outperform(s,ance,ed),
partner(s), payments, percent, pharmaceutical(s), price(d), primary,
recover(ed,s), redirect(ed,ion), stakeholder(s), stock(s),
violat(ing,ion,ors)
```

Many are suggestive of of good or bad news for a company or its stock price.

Some of them (econtent, edesign, eoperate) are also suggestive of the 'Dotcom Boom' of the late 1990s

This is perhaps the most complex example encountered so far. However it still represents an excessively simplistic approach to a real and complex project.

- No particular effort on the extraction of the names of companies and people involved. Furthermore it is not clear from individual words who are the subjects and objects of the events.
- Important modifiers like not, despite, and expect may not be adjacent to the phrases they modify.
- Markets react to news quickly. Hourly or instantaneous price changes should be used in order to trade on the information.
- Consider 3-class classification: `no change, surge, plunge`
- Time series nature of the data is almost completely ignored.

In addition to Chapter 10 of the textbook by Provost and Fawcett, here are a few references on this 'News-Stock Price' example:

Mittermayer, M., and Knolmayer, G. (2006). Text mining systems for market response to news: A survey. Working Paper No.184, Institute of Information Systems, University of Bern.

Zhang, J., Haerdle, W.K., Cheng, C.Y. and Bommes, E. (2015). Distillation of news flow into analysis of stock reactions.
`http://edoc.hu-berlin.de/series/sfb-649-papers/2015-5/PDF/5.pdf`

Chapter 11. Text Mining with R

Reference: Silge, J. and Robinson, D. (2017). Text Minging With R. O'Reilly. Available online at `https://www.tidytextmining.com/`

1. Tidy text format: a table with one token per row.

Token (or *Term*): a word, a phrase, or several connected words.

Other text data structures:

String: text data is often imported into R as strings (i.e. character vectors).

Corpus: a collection of raw strings annotated with additional meta data and details.

Document-term matrix: a sparse matrix representing a collection (i.e. a corpus) of documents, in which each row stands for a document, each column stands for a term/token, and each entry is, e.g. TFIDF.

unnest_tokens: a function in R-package tidytext which transform text strings to tidy text format via data_frame

```
text1 = c("Long ago, big data was a thick screen, I was here, mainframe computing wa
        there", "And now, big data is a thin smart-phone, I am here, cloud computing
        is there", "In future, big data will be tiny particles, I will be here,
        quantum computing will be there")
               # Poem "Big Data" by Professor Yazhen Wang
> class(text1)
[1] "character"
 length(text1)
[1] 3
> text1
[1] "Long ago, big data was a thick screen, I was here, mainframe computing was ther
[2] "And now, big data is a thin smart-phone, I am here, cloud computing is there"
[3] "In future, big data will be tiny particles, I will be here, quantum computing
       will be there"
```

text1 is a typical text vector to be analysed. In order to turn it into a tidy text dataset, we need to put it into a data frame using data_frame (NOT data.frame!!!).

```
> library(dplyr)
> text1_df = data_frame(text1)
> text1_df
```

```
# A tibble: 3 x 1
   text1
1 Long ago, big data was a thick screen, I am here, mainframe computing was there
2 And now, big data is a thin smart-phone, I am here, cloud computing is there
3 In future, big data will be tiny particles, I will be here, quantum computing ...
```

A tibble is a modern version of data frame. read_csv imports data into the tibble format.

```
> install.packages("tidytext")
> library(tidytext)
> unnest_tokens(text1_df, word1, text1)
# A tibble: 48 x 1
   word1
   <chr>
 1 long
 2 ago
 3 big
 4 data
 5 was
 6 a
 7 thick
 8 screen
 9 i
10 was
# ... with 38 more rows
```

```
> text1_tidy=unnest_tokens(text1_df, word1, text1)
> text1_tidy$word1
 [1] "long"      "ago"       "big"       "data"      "was"       "a"          "thick"
 [8] "screen"    "i"         "was"        "here"      "mainframe" "computing" "was"
[15] "there"     "and"       "now"       "big"       "data"      "is"         "a"
[22] "thin"      "smart"     "phone"     "i"         "am"        "here"       "cloud"
[29] "computing" "is"        "there"     "in"        "future"    "big"        "data"
[36] "will"      "be"        "tiny"      "particles"  "i"         "will"       "be"
[43] "here"      "quantum"   "computing" "will"      "be"        "there"
```

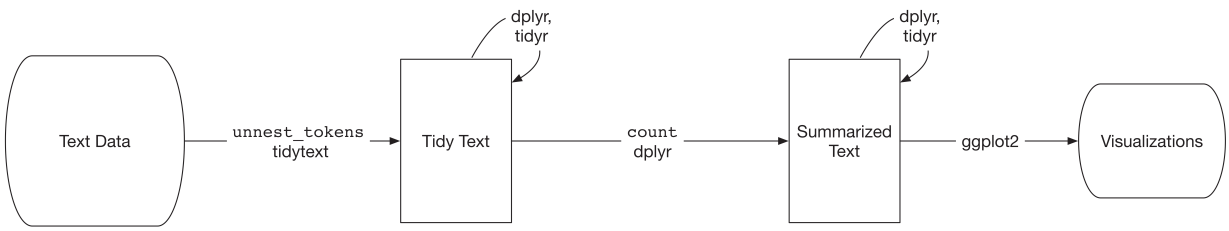
Output vector `word1` discards puctuation, converts the tockens (i.e. words) to lowercase, change `smart-phone` to `smart` and `phone`.

Using pipe: %>%

Command `text1_tidy=unnest_tokens(text1_df, word1, text1)` can be equivalently written as

```
> text1_tidy = text1_df %>% unnest_tokens(word1, text1)
```

Data in tidy-text format allow further analysis as illustrated below



For example

```
> count(text1_tidy, word1, sort=T)
# A tibble: 30 x 2
   word1         n
   <chr>     <int>
 1 be            3
 2 big           3
 3 data          3
 4 here          3
 5 i             3
 6 there         3
 7 will          3
 8 a             2
```

```
 9 was           2
10 computing     2
# ... with 20 more rows
```

The above steps can be combined together using pipes:

```
> text1_df %>% unnest_tokens(word1, text1) %>% count(word1, sort=T)
```

Now let us look at the novels by Jane Austen.

```
> install.packages("janeaustenr")
> library(janeaustenr); library(dplyr); library(tidytext)
> prideprejudice[1:11]
 [1] "PRIDE AND PREJUDICE"
 [2] ""
 [3] "By Jane Austen"
 [4] ""
 [5] ""
 [6] ""
 [7] "Chapter 1"
 [8] ""
 [9] ""
[10] "It is a truth universally acknowledged, that a single man
            in possession"
[11] "of a good fortune, must be in want of a wife."
> PP_df <- data_frame(prideprejudice)
> PP_tidy <- PP_df %>% unnest_tokens(word, prideprejudice)
```

Now all the words in *Pride & Prejudice* are in tidy text file `PP_tidy`.

To load the database of stop words, `data(stop_words)`. Note vector `stop_words` contains all the stop words from 3 lexicons `SMART, snowball, onix`. To use only the stop words from one lexicon, `stopwords1 = filter(stop_words, lexicon=="SMART")`.

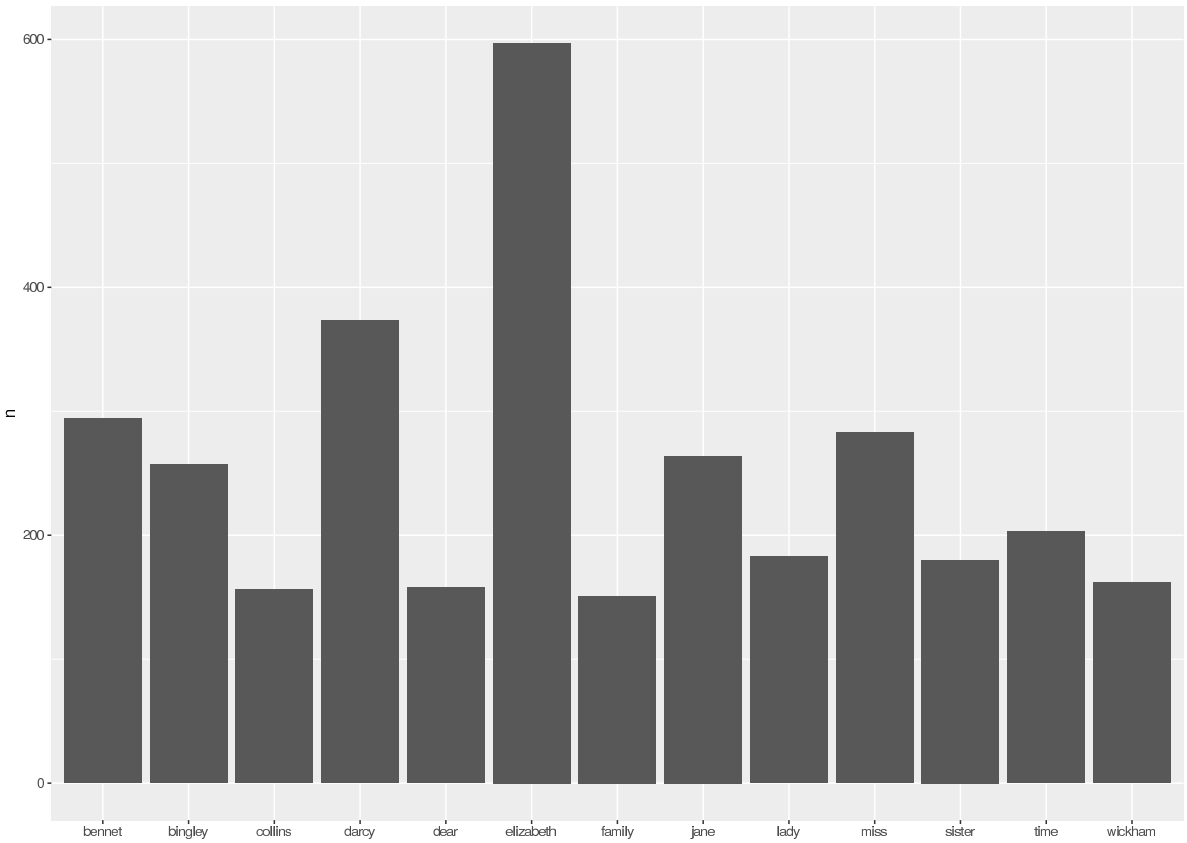
To separate stop words from the others in `PP_tidy`:

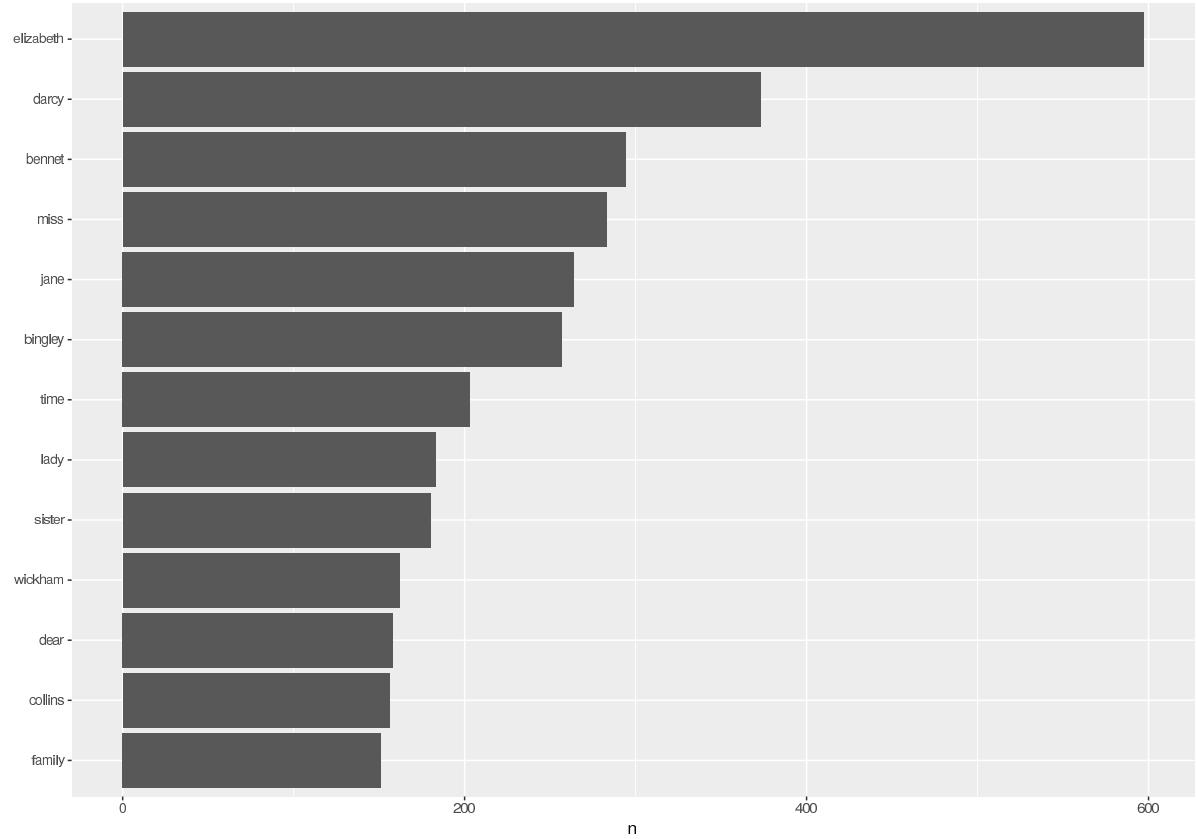
```
> PP_noS <- anti_join(PP_tidy, stop_words)
          # extract non-stop words from PP_tidy
> PP_stop <- semi_join(PP_tidy, stop_words)
          # extract stop words from PP_tidy
```

Now we produce a word-frequency bar-chart using `ggplot2`. It also illustrates the usefulness of piping %>%.

```
> library(ggplot2)
> PP_noS %>% count(word, sort=T)
# A tibble: 6,009 x 2
   word          n
   <chr>     <int>
 1 elizabeth   597
 2 darcy       373
 3 bennet      294
 4 miss        283
 5 jane        264
 6 bingley     257
 7 time        203
 8 lady        183
 9 sister      180
10 wickham     162
# ... with 5,999 more rows
> PP_noS %>% count(word, sort=T) %>% filter(n>150)
# A tibble: 13 x 2
   word          n
   <chr>     <int>
 1 elizabeth   597
 2 darcy       373
 3 bennet      294
 4 miss        283
 5 jane        264
```

```
 6 bingley     257
 7 time        203
 8 lady        183
 9 sister      180
10 wickham     162
11 dear        158
12 collins     156
13 family      151
> PP_noS %>% count(word, sort=T) %>% filter(n>150) %>% ggplot(aes(word,n)) +
+ geom_col() +
+ xlab(NULL)
           # Produce 1st figure
> PP_noS %>% count(word, sort=T) %>% filter(n>150) %>% mutate(word=reorder(word,n))
+ ggplot(aes(word,n)) +
+ geom_col() +
+ xlab(NULL) +
+ coord_flip()
           # Produce 2nd figure
>
```





To produce a word cloud plot:

```
> install.packages("wordcloud2")
> library(wordcloud2)
> PP_noS %>% count(word, sort=T) %>% wordcloud2()
```

You may also try

```
> PP_noS %>% count(word, sort=T) %>% filter(n>60) %>% wordcloud2()
```

Suppose we try to identify the authorship of a novel. One effective approach is to compare the relative frequencies of stop words in novels.

```
> emma_df <-  data_frame(emma)
> emma_tidy <- emma_df %>% unnest_tokens(word, emma)
> emma_stop <- emma_tidy %>% semi_join(stop_words)
> emma_noS <- emma_tidy %>% anti_join(stop_words)
> dim(emma_noS); dim(emma_stop); dim(PP_noS); dim(PP_stop)
[1] 46775       1
[1] 114221      1
[1] 37246       1
[1] 84958       1
```

Surprisingly both novels contain far more stop words than non-stop words.

```
> bind_rows(mutate(PP_stop, book="Pride & Prejudice"), mutate(emma_stop, book="Emma"
        # mutate adds a new column to data.frame
        # bind_rows binds data.frames with the same number columns together
# A tibble: 199,179 x 2
   word  book
   <chr> <chr>
 1 and   Pride & Prejudice
 2 by    Pride & Prejudice
```

```
 3 it    Pride & Prejudice
 4 is    Pride & Prejudice
 5 a     Pride & Prejudice
 6 that  Pride & Prejudice
 7 a     Pride & Prejudice
 8 man   Pride & Prejudice
 9 in    Pride & Prejudice
10 of    Pride & Prejudice
# ... with 199,169 more rows

> bind_rows(mutate(PP_stop,book="Pride & Prejudice"),mutate(emma_stop,book="Emma"))
+ count(book, word)
# A tibble: 1,056 x 3
   book  word            n
   <chr> <chr>       <int>
 1 Emma  a            3129
 2 Emma  able           72
 3 Emma  about         249
 4 Emma  above          12
 5 Emma  according       5
 6 Emma  accordingly     4
 7 Emma  across          7
 8 Emma  actually       29
 9 Emma  after         161
10 Emma  afterwards     41
```

```
# ... with 1,046 more rows

> bind_rows(mutate(PP_stop,book="Pride & Prejudice"),mutate(emma_stop,book="Emma"))
+ count(book, word) %>% mutate(proportion=n/sum(n))
# A tibble: 1,056 x 4
   book  word            n proportion
   <chr> <chr>       <int>      <dbl>
 1 Emma  a            3129  0.0157
 2 Emma  able           72  0.000361
 3 Emma  about         249  0.00125
 4 Emma  above          12  0.0000602
 5 Emma  according       5  0.0000251
 6 Emma  accordingly     4  0.0000201
 7 Emma  across          7  0.0000351
 8 Emma  actually       29  0.000146
 9 Emma  after         161  0.000808
10 Emma  afterwards     41  0.000206

> bind_rows(mutate(PP_stop,book="Pride & Prejudice"),mutate(emma_stop,book="Emma"))
+ count(book, word) %>% mutate(proportion=n/sum(n)) %>% select(-n)
# A tibble: 1,056 x 3
   book  word        proportion
   <chr> <chr>            <dbl>
 1 Emma  a           0.0157
 2 Emma  able        0.000361
```

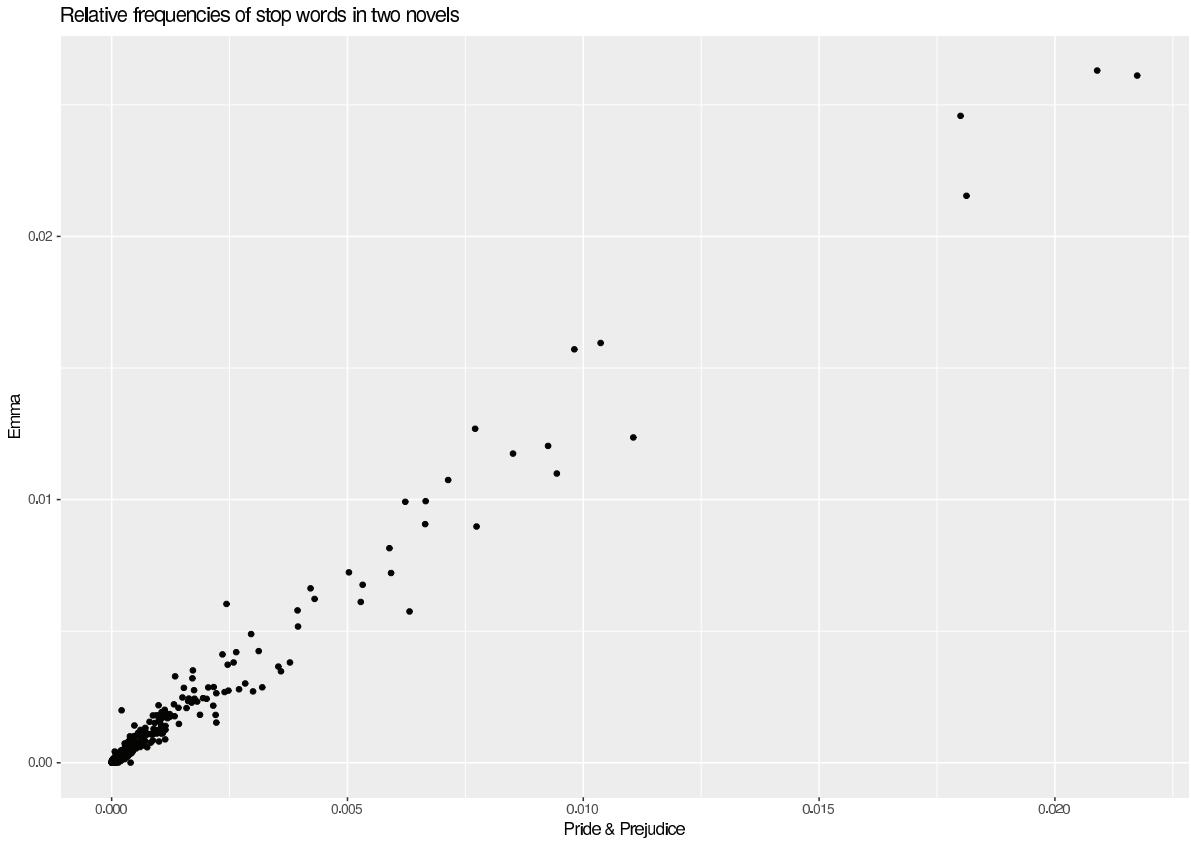
```
 3 Emma  about        0.00125
 4 Emma  above        0.0000602
 5 Emma  according    0.0000251
 6 Emma  accordingly  0.0000201
 7 Emma  across       0.0000351
 8 Emma  actually     0.000146
 9 Emma  after        0.000808
10 Emma  afterwards   0.000206
# ... with 1,046 more rows
```

Now we use `spread` (& `gather`) in `tidyr` to put the data in the shape for comparison:

```
> library(tidyr)
> bind_rows(mutate(PP_stop,book="Pride & Prejudice"),mutate(emma_stop,book="Emma"))
+ count(book, word) %>% mutate(proportion=n/sum(n)) %>% select(-n) %>%
+ spread(book, proportion)
# A tibble: 564 x 3
   word        Emma                ‘Pride & Prejudice‘
   <chr>           <dbl>               <dbl>
 1 a           0.0157              0.00981
 2 able        0.000361            0.000271
 3 about       0.00125             0.000613
 4 above       0.0000602           0.000105
 5 according   0.0000251           0.0000402
```

```
 6 accordingly 0.0000201           0.0000301
 7 across      0.0000351           0.0000251
 8 actually    0.000146            0.0000602
 9 after       0.000808            0.00100
10 afterwards  0.000206            0.000161
# ... with 554 more rows
> rF = bind_rows(mutate(PP_stop, book="Pride & Prejudice"),
                 mutate(emma_stop, book="Emma")) %>%
+ count(book, word) %>% mutate(proportion=n/sum(n)) %>% select(-n) %>%
+ spread(book, proportion)
> qplot(rF[,3], rF[,2], ylab="Emma", xlab="Pride & Prejudice",
       main="Relative frequencies of stop words in two novels")
```

The figure shows that the relative frequencies of the occurrency of stop words in the two Austen's novels are similar.



To compare Austen's writings with others, we download 2 Dickens' books from `http://www.gutenberg.org/ebooks/`.

First, Dickens' Great Expectation in html format.

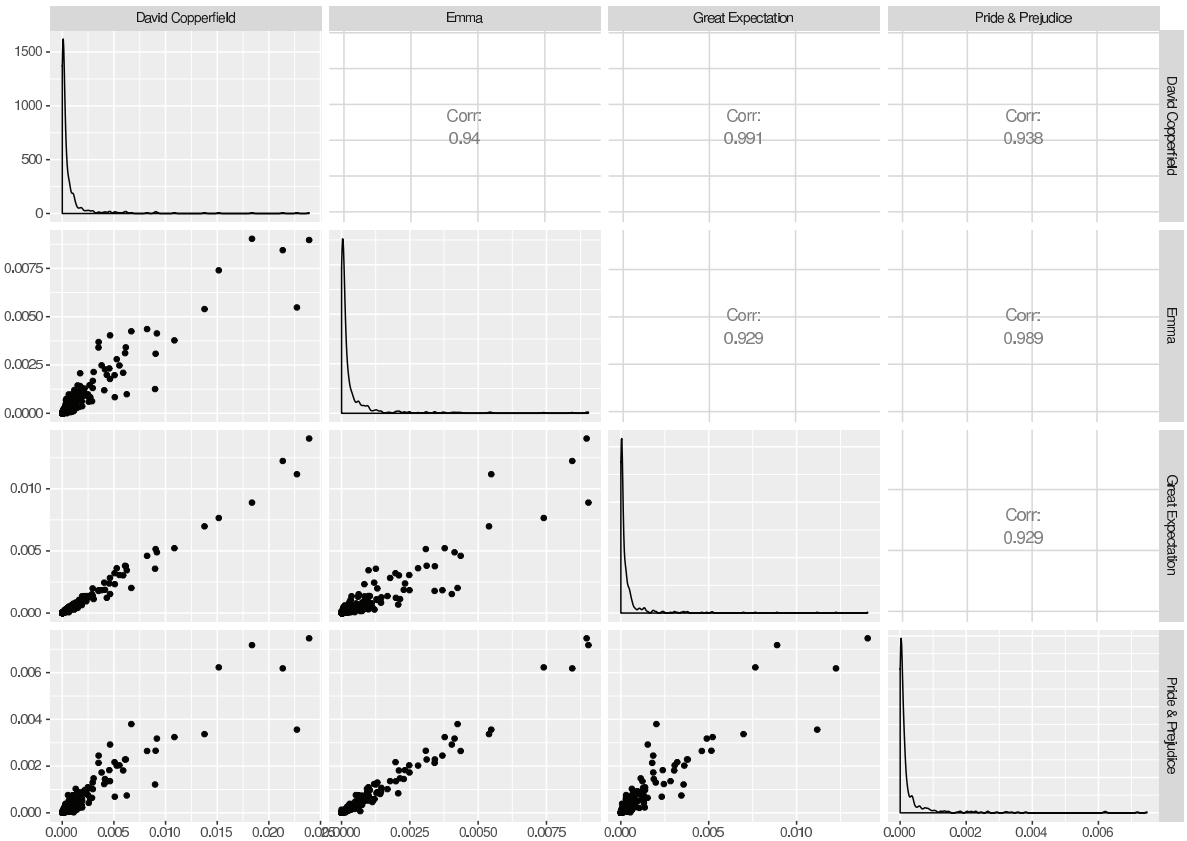
```
> install.packages("rvest") # Package for easy scrape of web pages
> library(rvest)
> GE <- read_html("http://www.gutenberg.org/files/1400/1400-h/1400-h.htm")
> GE_text=GE %>% html_nodes("p") %>% html_text()  # Extract text from html file
> GE_df = data_frame(GE_text)
> GE_tidy = GE_df %>% unnest_tokens(word, GE_text)
> GE_stop = GE_tidy %>% semi_join(stop_words)
```

To get Dickens' David Copperfield,

```
> DC = read_html("http://www.gutenberg.org/files/9744/9744-index.htm")
> DC_text = DC %>% html_nodes("p") %>% html_text()
> DC_df = data_frame(DC_text)
> DC_tidy = DC_df %>% unnest_tokens(word, DC_text)
> DC_stop = DC_tidy %>% semi_join(stop_words)
```

Now we combine the relative frequencies of stop words in 4 books together to produce a plot for comparison.

```
> rF4 = bind_rows(mutate(PP_stop, book="Pride & Prejudice"),
     mutate(emma_stop, book="Emma"),
     mutate(GE_stop, book="Great Expectation"),
     mutate(DC_stop, book="David Copperfield")) %>%
+ count(book, word) %>% mutate(proportion=n/sum(n)) %>% select(-n) %>%
+ spread(book, proportion)
> rF4
# A tibble: 659 x 5
   word        `David Copperfield`   Emma     `Great Expectation`   `Pride & Prejudi
   <chr>                     <dbl>   <dbl>                   <dbl>              <dbl>
 1 a                        0.0138  0.00540                 0.00698            0.00337
 2 able                     0.0000707 0.000124              0.0000552          0.0000931
 3 about                    0.00114 0.000429                0.000552           0.000210
 4 above                    0.0000966 0.0000207             0.0000552          0.0000362
 5 according                0.0000310 0.00000862            0.0000310          0.0000138
 6 accordingly              0.0000362 0.00000690            0.00000345         0.0000103
 7 across                   0.0000948 0.0000121             0.0000759          0.00000862
 8 actually                 0.0000276 0.0000500             0.0000172          0.0000207
 9 after                    0.000769 0.000278               0.000504           0.000345
10 afterwards               0.000203 0.0000707              0.0000724          0.0000552
# ... with 649 more rows

> rF4c = rF4 %>% drop_na()   # Drop the rows with "na"
> library("GGally", lib.loc="~/R/x86_64-pc-linux-gnu-library/3.2")
> ggpairs(rF4c[,2:5])
```



Chapter 12. Data Visualization

Goal: to help people understand the significance of data by placing it in a visual context: Patterns, trends and correlations that might go undetected in text-based data can be exposed and recognized easier when displayed visually.

Hans Rosling (Gapminder inventor): find catchy ways to illustrate statistics, and

> Having the data is not enough, I have to show it in the ways people both enjoy and understand.

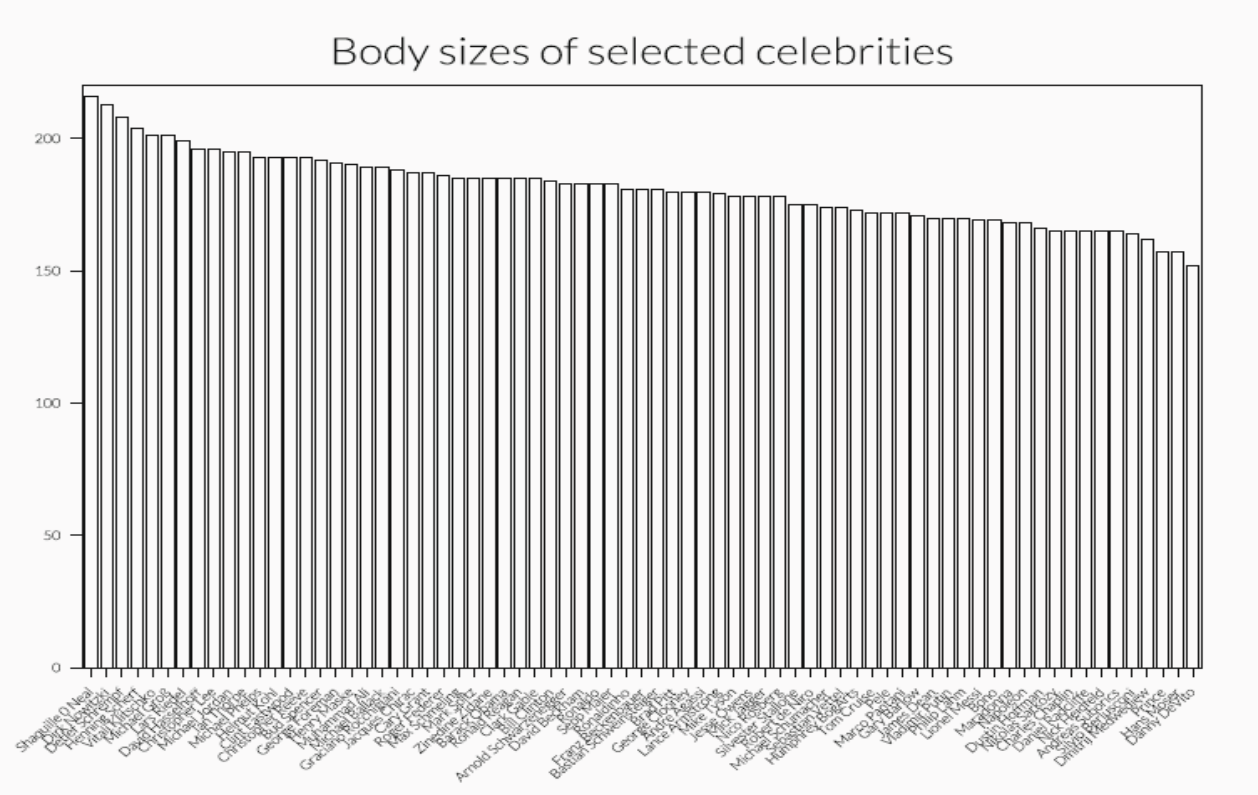
Rahlf, T. (2017). Data Visualisation with R. Springer.
(Data and R-scripts are available at `extras.springer.com/2017/978-3-319-49750-1`)

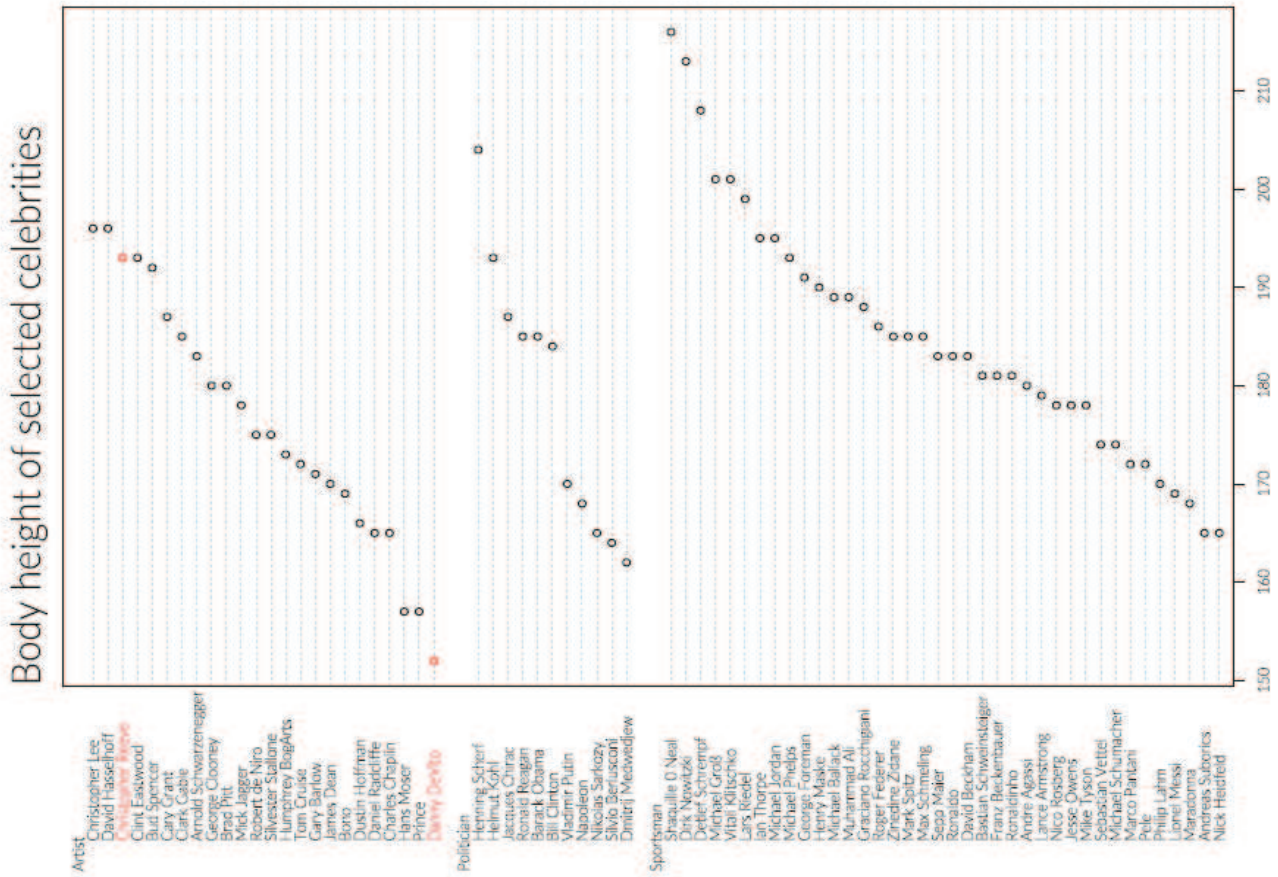
Unwin, A. (2015). Graphical Data Analysis with R. CRC Press.

Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer. (Available online)

An important tip in designing figures: Catch accurate perception of the data.

An unfortunate choice of presentation format even in simple plots can severely impair the information in data





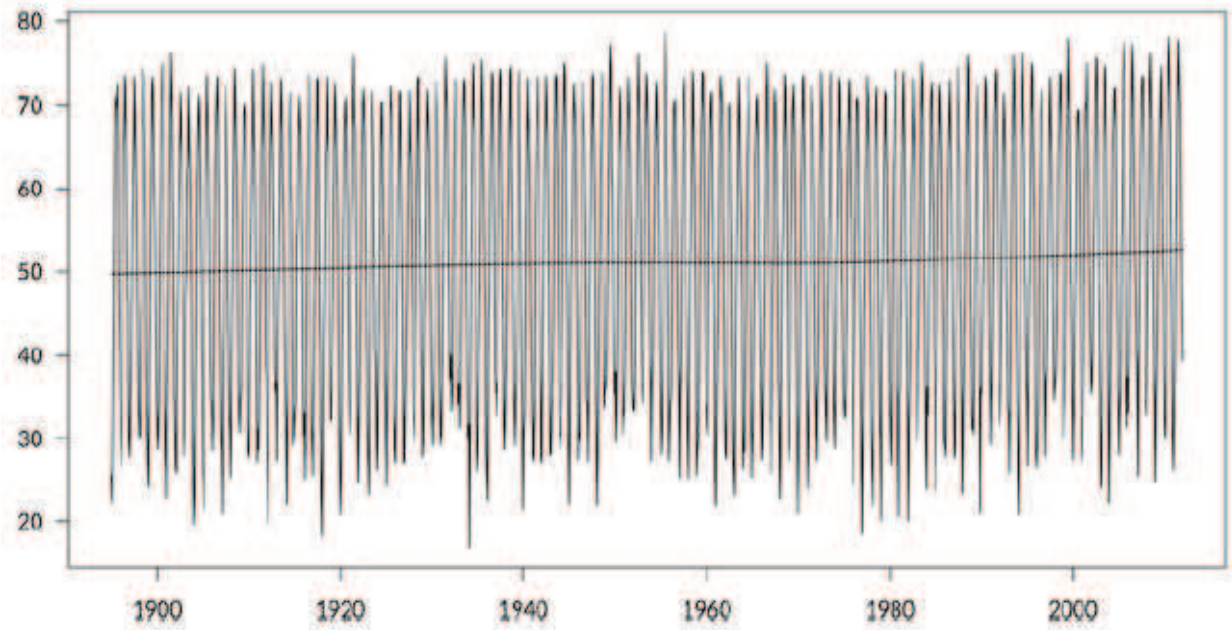
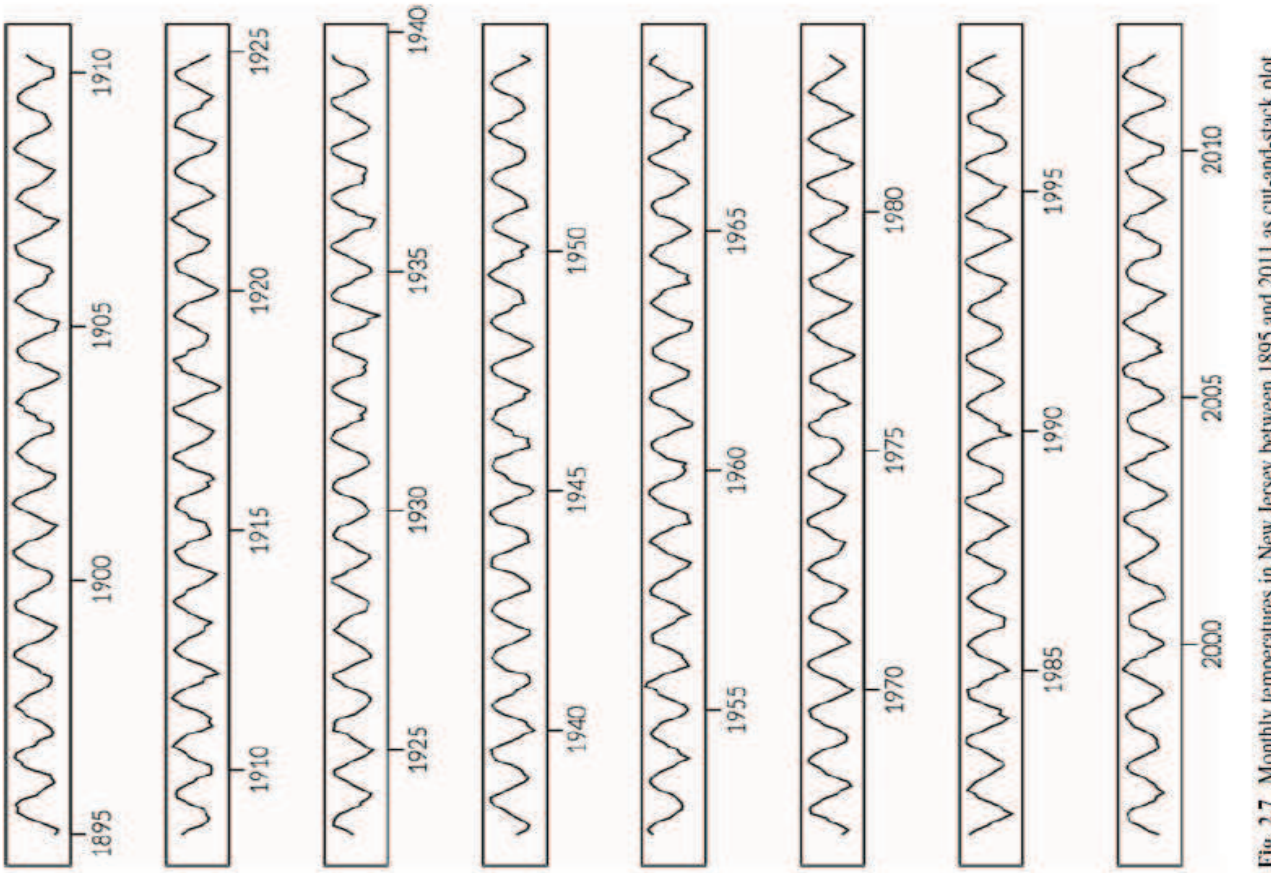


Fig. 2.6 Monthly temperatures in New Jersey between 1895 and 2011 with trend line



More good examples of data visualization are available at:

http://www.r-graph-gallery.com/portfolio/ggplot2-package/

http://flowingdata.com

http://driven-by-data.net

http://marijerooze.nl/thesis

http://visualizingeconomics.com

http://rgraphgallery.blogspot.de

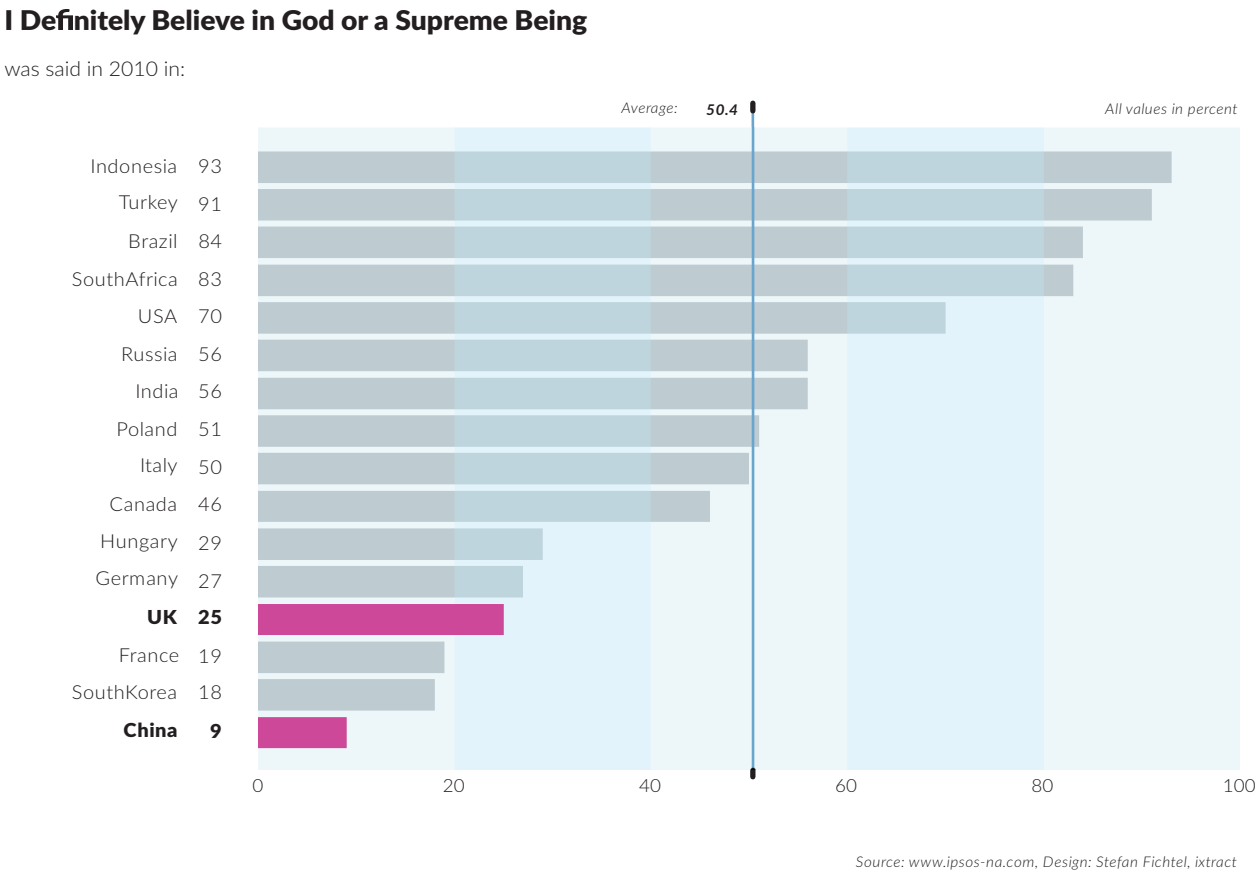
A bar chart: a simple illustration

The figure shows the results of a 2010 survey carried out in different countries: How many percent of the respondents agreed with the statement 'I Definitely Believe in God or a Supreme Being'?

Data are derived from an Ipsos survey that was ordered by the Thompson Reuters News Service and performed between 7 and 23 September 2010 in 24 countries.

The survey included approximately 500 to 1000 people aged between 16 and 64 and from each country.

For good presentation, only the data from 16 countries are used in plot.



Source: www.ipsos-na.com, Design: Stefan Fichtel, ixtract

Here is the script to produce the bar-chart (‘believeGod.r’):

```
# cairo_ps("believeGod.eps", width=8.3, height=6, pointsize=10)
# cairo_pdf("believeGod.pdf", width=8.3, height=6, pointsize=10)
par(omi=c(0.65,0.25,0.75,0.75),mai=c(0.3,2,0.35,0),mgp=c(3,3,0),
family="Lato Light", las=1)

# Import data and prepare chart
ipsos=read.table("believeGod.txt",header=T)
sort.ipsos=ipsos[order(ipsos$Percent) ,]
attach(sort.ipsos)

# Create chart
x=barplot(Percent,names.arg=F,horiz=T,border=NA,xlim=c(0,100),
col="grey", cex.names=0.85,axes=F)

# Label chart
for (i in 1:length(Country))
{
if (Country[i] %in% c("UK","China"))
  {myFont="Lato Black"} else {myFont="Lato Light"}
text(-8,x[i],Country[i],xpd=T,adj=1,cex=0.85,family=myFont)
text(-3.5,x[i],Percent[i],xpd=T,adj=1,cex=0.85,family=myFont)
}
```

```
# Other elements
rect(0,-0.5,20,28,col=rgb(191,239,255,80,maxColorValue=255), border=NA)
rect(20,-0.5,40,28,col=rgb(191,239,255,120,maxColorValue=255), border=NA)
rect(40,-0.5,60,28,col=rgb(191,239,255,80,maxColorValue=255), border=NA)
rect(60,-0.5,80,28,col=rgb(191,239,255,120,maxColorValue=255), border=NA)
rect(80,-0.5,100,28,col=rgb(191,239,255,80,maxColorValue=255), border=NA)
myValue2=c(9,0,0,25,0,0,0,0,0,0,0,0,0,0,0,0)
myColour2=rgb(255,0,210,maxColorValue=255)
x2=barplot(myValue2,names.arg=F,horiz=T,border=NA,xlim=c(0,100),
col=myColour2,cex.names=0.85,axes=F,add=T)
arrows(50.4,-0.5,50.4,20.5,lwd=1.5,length=0,xpd=T,col="skyblue3")
arrows(50.4,-0.5,50.4,-0.75,lwd=3,length=0,xpd=T)
arrows(50.4,20.5,50.4,20.75,lwd=3,length=0,xpd=T)
text(41,20.5,"Average",adj=1,xpd=T,cex=0.65,font=3)
text(49,20.5,"50.4",adj=1,xpd=T,cex=0.65,family="Lato",font=4)
text(100,20.5,"All values in percent",adj=1,xpd=T,cex=0.65,font=3)
mtext(c(0,20,40,60,80,100),at=c(0,20,40,60,80,100),1,line=0,cex=0.80)

# Titling
mtext("I Definitely Believe in God or a Supreme Being",3,line=1.3,adj=0,
cex=1.2,family="Lato Black",outer=T)
mtext("was said in 2010 in:",3,line=-0.4,adj=0,cex=0.9,outer=T)
mtext("Source: www.ipsos-na.com, Design: Stefan Fichtel, ixtract",1,line=1,
adj=1.0,cex=0.65,outer=T,font=3)
# dev.off()
```

ggplot2: to produce elegant and sophisticated (but static) 2-dim graphics economically

- based on `grammar of graphics` (Wilkinson, 2005)
- plot specification at a high level of abstraction
- flexible and iterative processing
- theme system for polishing plot appearance
- mature and complete graphics system
- many users, active mailing list: `groups.google.com/group/ggplot2`

Iteratively: an initial layer showing raw data only, adding layers of annotations and statistical summaries.

3 key components of every plot: data, aesthetics and geoms.

To install: `install.packages("ggplot2")`

Illustration with the fuel economy data set `mpg` included in `ggplot2`:

```
> library(ggplot2)
> mpg
# A tibble: 234 x 11
   manufacturer      model displ  year   cyl      trans   drv   cty   hwy    fl
          <chr>      <chr> <dbl> <int> <int>      <chr> <chr> <int> <int> <chr>
 1         audi         a4   1.8  1999     4   auto(l5)     f    18    29     p
 2         audi         a4   1.8  1999     4 manual(m5)     f    21    29     p
 3         audi         a4   2.0  2008     4 manual(m6)     f    20    31     p
 4         audi         a4   2.0  2008     4   auto(av)     f    21    30     p
 5         audi         a4   2.8  1999     6   auto(l5)     f    16    26     p
 6         audi         a4   2.8  1999     6 manual(m5)     f    18    26     p
 7         audi         a4   3.1  2008     6   auto(av)     f    18    27     p
 8         audi a4 quattro   1.8  1999     4 manual(m5)     4    18    26     p
 9         audi a4 quattro   1.8  1999     4   auto(l5)     4    16    25     p
10         audi a4 quattro   2.0  2008     4 manual(m6)     4    20    28     p
# ... with 224 more rows, and 1 more variables: class <chr>
```

`cty` and `hwy` record miles per gallon (mpg) for city and highway driving

`displ` is the engine displacement in liters

`drv` is the drive train: front wheel (f), rear wheel (r) or four wheel (4)

`model` is the model of car, in total 38 models between 1999 and 2008

`class` (not shown), describes the type of car: two seater, SUV, compact, etc.

Questions of interest:

How are engine size and fuel economy related?

Do certain manufacturers care more about economy than others?

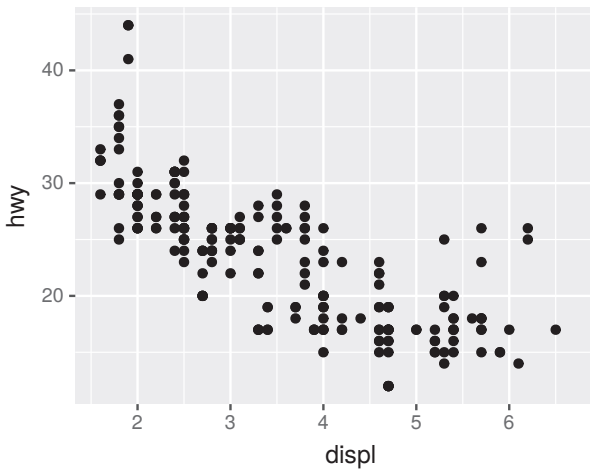
Has fuel economy improved in the last ten years?

Every ggplot2 plot has three key components:

1. data in the form of `data.frame` or `tipple`, containing all the variables to be used in the plot
2. a set of aesthetic mappings between variables in the data and visual properties, and
3. at least one layer which describes how to render each observation. Layers are usually created with a geom function.

We start with a simple plot:

```
ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point()
```



This scatterplot is defined by:

1. Data: `mpg`
2. Aesthetic mapping: x - engine size, y - fuel economy
3. Layer: points

Note. First 2 arguments are always mapped to x and y.

Equivalently,

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point()
```

Also try:

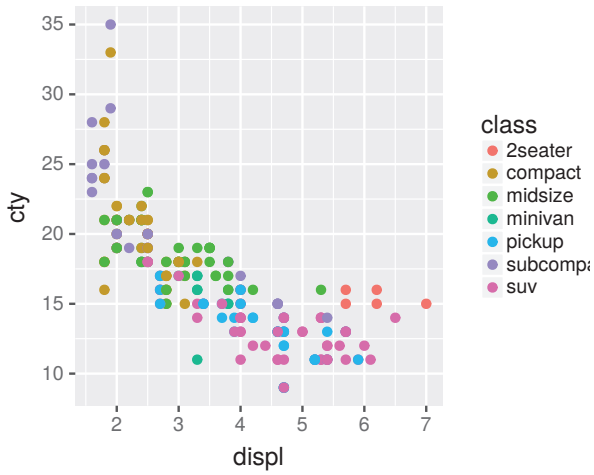
```
1. ggplot(mpg, aes(cty, hwy)) + geom_point()
2. ggplot(diamonds, aes(carat, price)) + geom_point()
3. ggplot(economics, aes(date, unemploy)) + geom_line()
4. ggplot(mpg, aes(cty)) + geom_histogram()
```

To add additional variables to a plot, use other aesthetics like colour, shape, and size, such as

```
aes(displ, hwy, colour = class)
aes(displ, hwy, shape = drv)
aes(displ, hwy, size = cyl)
```

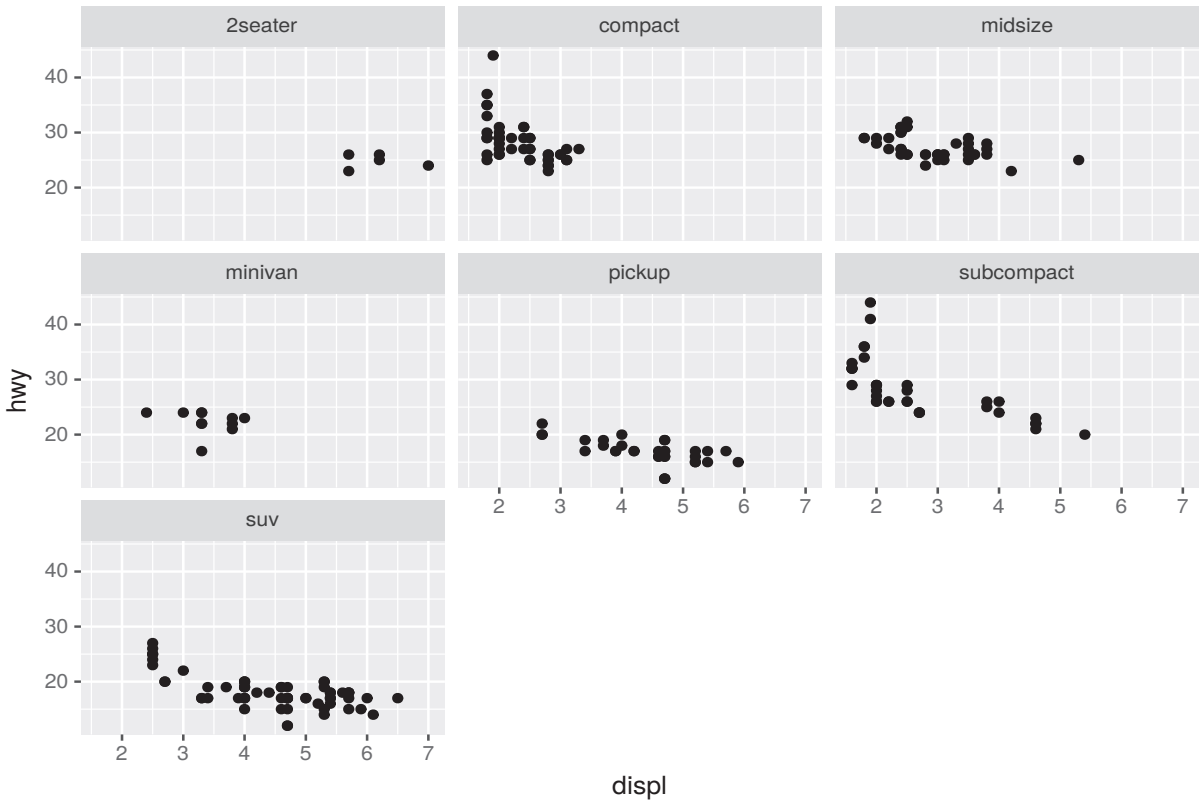
The plot on the right is produced by

```
ggplot(mpg,aes(displ,cty,colour=class))+
geom_point()
```



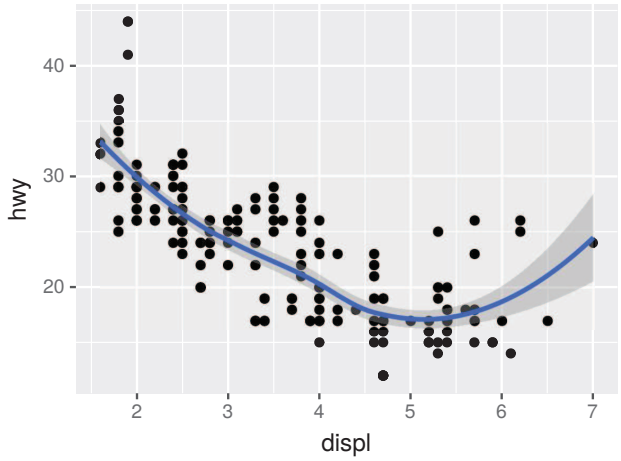
Faceting: creates tables of graphics by splitting the data into subsets and displaying the same graph for each subset

```
ggplot(mpg, aes(displ, hwy)) + geom_point() + facet_wrap(~class)
```

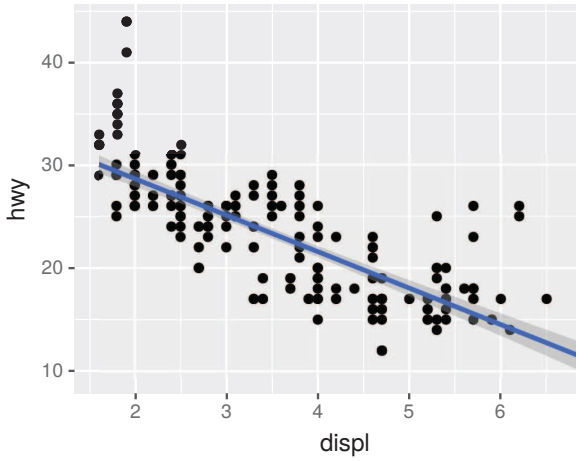


For a noisy scatterplot, add smoothed or straight line to reveal the dominant pattern.

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  geom_smooth()
```

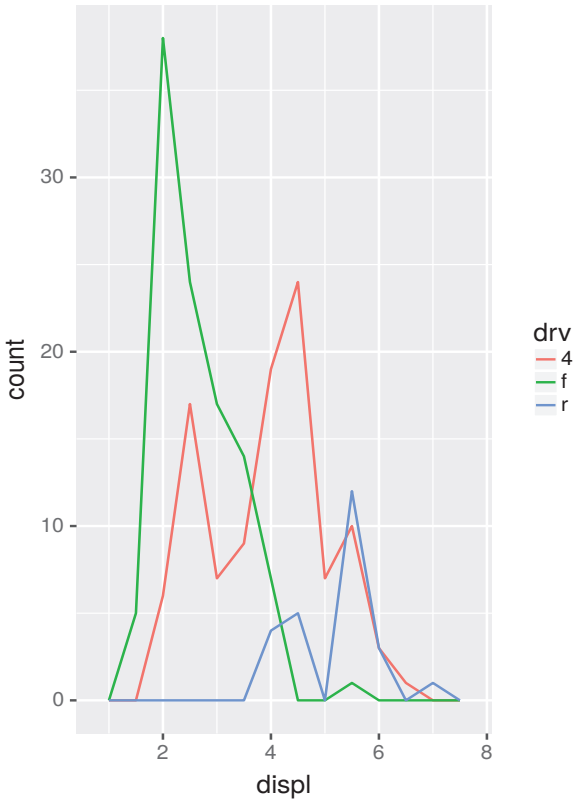


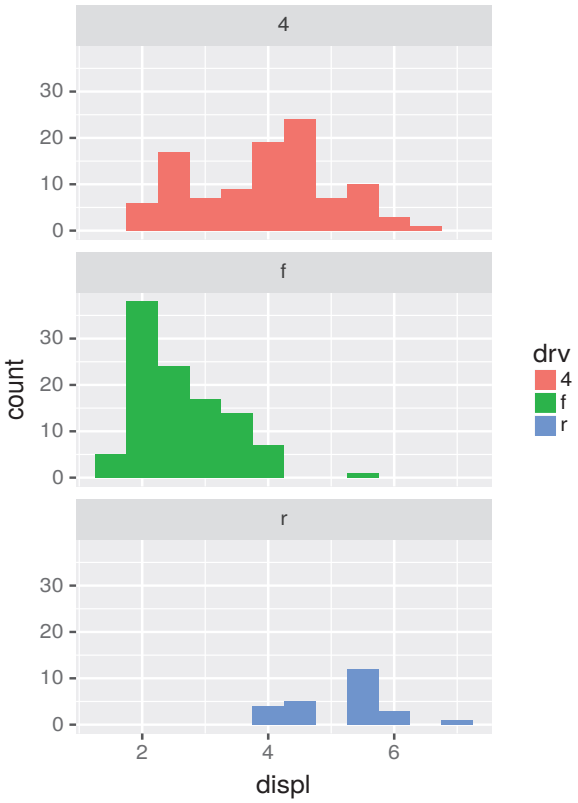

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  geom_smooth(method="lm")
```



Histograms and frequency polygons show the distribution of a single numeric variable. They provide more information about the distribution of a single group than boxplots do.

```
install.packages("gridExtra") # Put multiple plots together within ggp
require(gridExtra)
plot1=ggplot(mpg, aes(displ, colour=drv)) + geom_freqpoly(binwidth=0.5
plot2=ggplot(mpg, aes(displ, fill=drv)) + geom_histogram(binwidth=0.5)
  facet_wrap(~drv, ncol = 1)
grid.arrange(plot1, plot2, ncol=2)
```


count
30
20
10
0
2
4
6
8
displ
drv
4
f
r


4
f
r
count
30
20
10
0
2
4
6
displ
drv
4
f
r

Chapter 13. Principal Component Analysis

A generic unsupervised technique for dimensional reduction in the sense that most (if not all) variation/information of the data is surpressed into a few directions which is called principal component directions.

Principal Component Analysis (PCA) refers to the process of computing principal components and subsequent using those components in understanding data.

PCA has been used in regression, classification, clustering and other data mining tasks. It has also been used as data exploratory tools such as visulizing high-dimensional data.

Given a $n \times p$ data matrix $\mathbf{X} = (x_{ij})$, i.e. x_{ij} is the (i, j)-th element, denoting the j-th attribute of the i-th individual.

Basic idea: seek for a linear combination (i.e. a new attribute)

$$z_i = \phi_1 x_{i1} + \phi_2 x_{i2} + \cdots + \phi_p x_{ip}$$

such that the (sample) variance of z_i is maximized.

Constraint: $\phi_1^2 + \phi_2^2 + \cdots + \phi_p^2 = 1$.

Note. Without loss of generality, we may assume that the mean of each of the p attributes is 0. This means that we replace x_{ij} by $x_{ij} - \bar{x}_j$, where

$$\bar{x}_j = \frac{1}{n}(x_{1j} + x_{2j} + \cdots + x_{nj})$$

Find the first principal component:

$$\max_{\{\phi_j\}} \frac{1}{n} \sum_{i=1}^{n} (\phi_1 x_{i1} + \phi_2 x_{i2} + \cdots + \phi_p x_{ip})^2 \quad \text{subject to} \quad \phi_1^2 + \phi_2^2 + \cdots + \phi_p^2 = 1.$$

Once we have found the 1st PC, we can repeat the above idea to seek for the 2nd PC.

But now we should require the 2nd PC to be linear independent of (i.e uncorrelated to) the 1st PC.

Similarly the 3rd PC should be uncorrelated to the 1st and 2nd PCs.

Algorithm. Eigenanalysis for sample covariance matrix $\mathbf{\Sigma} \equiv \mathbf{X}^T\mathbf{X}$, then it holds that

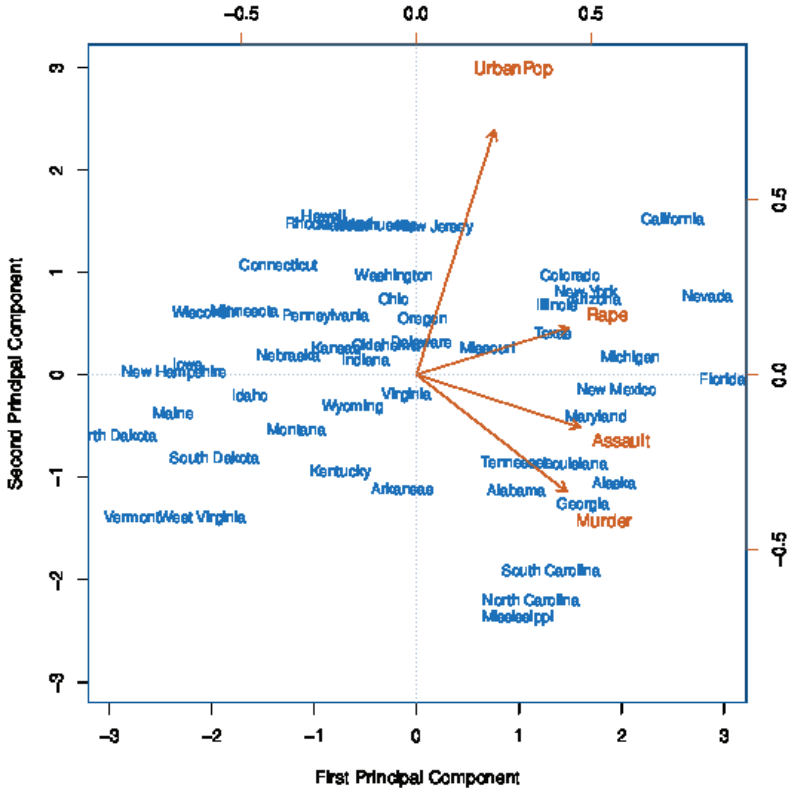
$$\mathbf{\Sigma} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}^T,$$

where $\mathbf{\Gamma} = (\boldsymbol{\gamma}_1, \cdots, \boldsymbol{\gamma}_p)$ be $p \times p$ orthogonal matrix, and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \cdots, \lambda_p)$ be a diagonal matrix with $\lambda_1 \geq \cdots \geq \lambda_p \geq 0$.

Then the components of $\boldsymbol{\gamma}_k$ are the coefficients, also called loadings, of the k-th PC, and the k-th PC has the sample variance λ_k, $k = 1, \cdots, p$.

Illustration with a real data: For each of $n = 50$ states in USA, we collect $p = 4$ attributes — `Assault`, `Murder`, `Rape` are the numbers of of arrests per 100, 000 residents for each of those three crimes.

Also record `UrbanPop` – the percent of the population in each state living in urban areas.



Two principal component biplots

The blue state names represent the scores for the first two principal components.

The orange arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for Rape on the first component is 0.54, and its loading on the second principal component 0.17 (the word Rape is centered at the point (0.54, 0.17)).

The 1st PC places approximately equal weight on `Assault`, `Murder`, and `Rape`, with much less weight on `UrbanPop`. Hence this component roughly corresponds to a measure of overall rates of serious crimes.

The 2nd places most of its weight on `UrbanPop` and much less weight on the other three features. Hence, this component roughly corresponds to the level of urbanization of the state.

	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

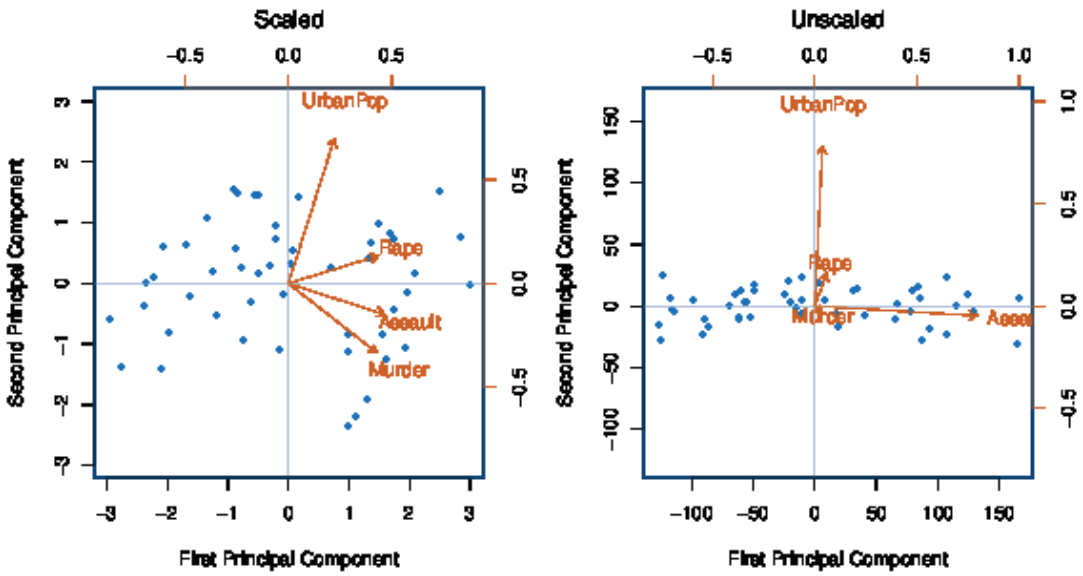
Interpretation. States with large positive scores on the first component, such as California, Nevada and Florida, have high crime rates, while states like North Dakota, with negative scores on the first component, have low crime rates. California also has a high score on the second component, indicating a high level of urbanization, while the opposite is true for states like Mississippi. States close to zero on both components, such as Indiana, have approximately average levels of both crime and urbanization.

Scaling the Variables

We should center each attribute first before applying PCA.

The results obtained when we perform PCA will also depend on whether the variables have been individually scaled (each multiplied by a different constant).

One often used scaling: make each attribute a unit vector.



Left: the first two PCs with the data scaled such that each attribute have unit variance. Right: the first two PCs using unscaled data.

The sample variances for `Murder, Rape, Assault, UrbanPop` are 18.97, 87.73, 6945.16, and 209.5 respectively.

If we perform PCA on the unscaled variables, the 1st PC has a large loading on `Assault`

Uniqueness of the Principal Components

Each PC is unique, up to a sign flip.

Deciding How Many Principal Components to Use

In general a $n \times p$ data matrix X has p distinct principal components. Usually not all of them are interesting.

We would like to use the smallest number of principal components required to get a good understanding of the data. How many principal components are needed? Unfortunately, there is no single (or simple!) answer to this question.

Eyeballing: looking for a point at which the proportion of variance explained by each subsequent principal component drops off.