Exploratory data analysis

Cleaning and exploring the data

Vladimir Batagelj

IMFM Ljubljana, IAM UP Koper, NRU HSE Moscow

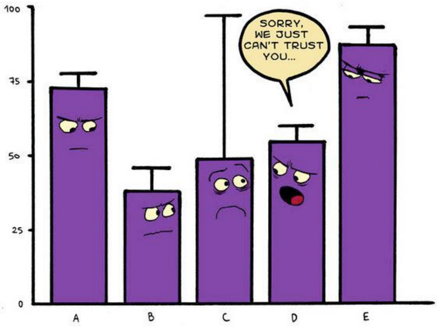
Master's programme

Applied Statistics with Social Network Analysis

International Laboratory for Applied Network Research

NRU HSE, Moscow 2021

Outline



Vladimir Batagelj: vladimir.batagelj@fmf.uni-lj.si

Current version of slides (November 21, 2021 at 22 : 31): slides PDF

Cleaning the data

We collected the data in a CSV file. We can inspect them using a text editor or a spreadsheet program. We can also import them into R

```
> wdir <- "D:/vlado/EDA/data"
> setwd(wdir)
> booksF <- paste("https://raw.githubusercontent.com/bavla/",
   "HSE/master/EDA/newBooks.csv",sep="")
> T <- read.csv2(url(booksF),stringsAsFactors=FALSE)
> dim(T)
[1] 970  15
> nrow(T)
[1] 970
> ncol(T)
[1] 15
> head(T)
> tail(T)
> T[c(5,9,333),1:8]
    bID     Amazon      bind npag          pub year    lang wid
5     5 0199206651 Hardcover  720   Oxford UP 2010 English 9.8
9     9 1473952123 Paperback  248         SAGE 2017 English 6.7
333 332 1546640010 Paperback   74 CreateSpace 2017 English   6
```

Cleaning and exploring the data

An informative view of a data frame is provided by the function `str`

```
> str(T)
'data.frame':   970 obs. of  15 variables:
 $ bID    : chr  "1" "2" "3" "4" ...
 $ Amazon: chr  "0521840856" "0521387078" "1446247414" "0195379470" ...
 $ bind   : chr  "Hardcover" "Paperback" "Paperback" "Paperback" ...
 $ npag   : int  402 857 304 264 720 207 344 744 248 272 ...
 $ pub    : chr  "Cambridge University Press" "Cambridge University Press" "SAGE Publi
 $ year   : int  2004 1994 2013 2011 2010 2014 2005 2010 2017 2011 ...
 $ lang   : chr  "English" "English" "English" "English" ...
 $ wid    : chr  "6" "6" "7.3" "9.2" ...
 $ thi    : chr  "1.1" "1.5" "0.7" "0.7" ...
 $ hei    : chr  "9" "9" "9.1" "6.1" ...
 $ duni   : chr  "inches" "inches" "inches" "inches" ...
 $ weig   : chr  "1.4" "2.6" "1.4" "12.8" ...
 $ wuni   : chr  "pounds" "pounds" "pounds" "ounces" ...
 $ pric   : chr  "121.52" "52.41" "37.38" "20.75" ...
 $ titl   : chr  "Amazon.com: Generalized Blockmodeling (Structural Analysis in the Sc
```

The data obtained from our scraping program are “messy” – we need to *clean* them to be ready for analysis. This is true for most data obtained from different sources. After cleaning we *explore* the data to “get feeling” and ideas for analyses. Sometimes, if possible, we need to correct our scraping program and repeat the data collection. For larger data collections a test collection of a small sample is adwised.
It is useful to preserve a copy of original raw data. Many problems can be resolved by correcting the original data in its copy. From the corrected data we construct a data frame (or some other structure) for analyses.

Cleaning the data

Typical tasks in data cleaning

- correcting for unexpected values; consider extreme and influential units.
- normalization of values (dates in different formats; weights, money, lengths in different units; recategorization; unification: lower/upper case, nonASCII chars, `’`; names (first, last)).
- factorization of ordinal and categorical variables.
- splitting variables (date $\rightarrow$ year, month, day; name $\rightarrow$ first, last).
- combining variables (year, month, day $\rightarrow$ date).
- transforming variables (date $\rightarrow$ day of week; Box-Cox (1, 2, 3)).
- combining, adding data from other sources (geographical coordinates).
- aggregating data.
- dealing with missing data.

Missing data

There are different options to deal with missing data:

- do nothing, mark with NA.
- find the value and insert it.
- remove the unit (in creating clean data frame).
- impute a value (guess, mean value, random, nearest neighbor, interpolation)

Identity (entity resolution) problem

In dealing with data extracted from text sources we often encounter the *identity problem*. It has two parts:

- *equivalence* (different words/phrases representing the same term – synonyms); and
- *ambiguity* (same word/phrase representing different terms – homonyms).

When dealing with names of people that include Chinese the "three Zhang, four Li" effect can make it to the surface.

The problem can be partially solved using dictionaries, considering context, using tools like stemming and lemmatization, etc.

For cleaning of Amazon data see the wiki page.

Amazon: old books – May 2012

A larger Amazon data set (more than 150000 books) is available at Github.

```
> help(read.csv)
> getwd()
[1] "C:/Users/Batagelj/test/python/2012/amazon"
> setwd("C:/Users/Batagelj/test/python/2012/amazon")
> dat <- read.csv2("booksT.csv",header=FALSE,stringsAsFactors=FALSE)
> dim(dat)
[1] 16804    23
> names(dat)
 [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12" "V13" "V
[16] "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23"
> dat[c(3,7),]
  V1 V2 V3         V4                                          V5
3  3 30 33 1451648537                             Walter Isaacson               Steve J
7  7 53 60 140123206X Scott Snyder, Jock, Francesco Francavilla Batman: The Black Mir
                                                     V7   V8             V9  V10   V11   V1
3 Simon & Schuster; First Edition ~1st Printing edition 2011 Hardcover 656  35.0 16.8
7                                               DC Comics 2011 Hardcover 304 29.99  16.
                             V13
3 Biography/Autobiography§1955-2011§Biography§Businessmen§Computer engineers§Jobs, St
7 Comic books, strips, etc§Graphic novels§Comics & Graphic Novels§Comics & Graphic No
  V14 V15 V16 V17 V18 V19 V20 V21 V22 V23
3  26  27  28  29  30  31  27  32  26  33
7  54  55  56  57  58  59  54  55  56  60
>
------------------------
V1   index    V4   AmazonID    V7   publisher     V10   pages            V13   subject
V2   lenQ     V5   authors     V8   year          V11   listPrice    V14-V23   neighbo
V3   lenK     V6   title       V9   binding       V12   price
------------------------
> year <- dat$V8
> summary(year)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
      0    2002    2008    1970    2011    2013      17
```

Amazon: data cleaning and exploration

```
> year <- dat$V8; pages <- dat$V10; binding <- dat$V9; price <- dat$V12
> isNA <- which(is.na(year)|is.na(pages)|is.na(binding)|is.na(price))
> year <- year[-isNA]; pages <- pages[-isNA]; binding <- binding[-isNA]
> typeof(price)
[1] "character"
> price <- as.numeric(price[-isNA])
> OK <- (0<pages)&(pages<2050) & (1900<year)&(year<2013) & (0<price)&(price<2000)
> table(OK)
OK
FALSE  TRUE
 1759 15028
> pages <- pages[OK]; binding <- binding[OK]; year <- year[OK]; price <- price[OK]
> bind <- rep(3,length(binding))
> B1 <- c("Paperback", "Perfect Paperback", "Mass Market Paperback")
> B2 <-  c("Hardcover", "Bonded Leather", "Leather Bound", "Hardcover-spiral")
> bind[binding %in% B1] <- 1
> bind[binding %in% B2] <- 2
> table(bind)
> plot(density(pages))
> plot(density(year))
> plot(density(price[(0<price)&(price<60)]))
> plot(pages,price,col=c("red","blue","green")[bind],pch=16,cex=0.1)
```

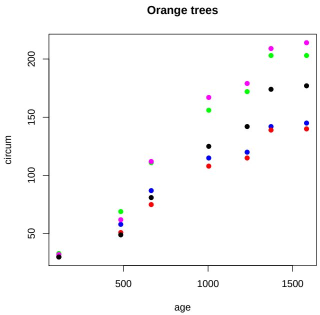
Exploring the data

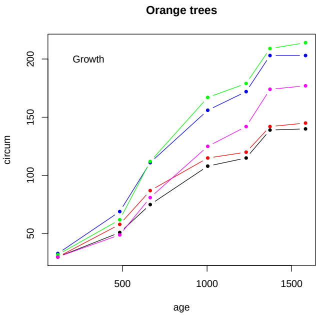
Exploration phase of data analysis gives us an initial insight in the data – we get feeling about variables and their relations. It also provides hypotheses for further analyses.

We usually start the exploration by looking at each variable separately (univariate). Besides numerical characteristics we use also visualizations according to the type of variable.

Later we look to relations among variables (multivariate). The two main types of relations are association (regression) and grouping (clustering).

Basic data visualization in R

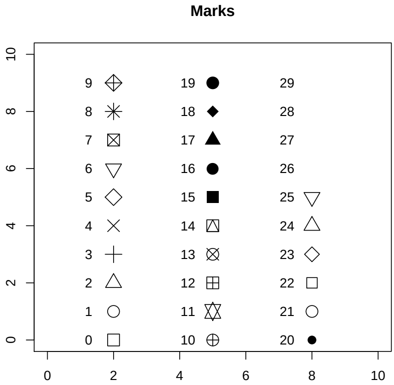





```
> help(plot)
> (c <- Orange[29:35,2])
[1]  118  484  664 1004 1231 1372 1582
> b <- c("red","blue","black","green","magenta")
> plot(Orange[,2],Orange[,3],col=b[Orange[,1]],xlab="age",ylab="circum",
+  pch=20,cex=1.5,main="Orange trees")
> plot(Orange[,2],Orange[,3],xlab="age",ylab="circum",main="Orange trees",type="n")
> for(k in 1:5){points(c,Orange[(7*k-6):(7*k),3],col=b[k],pch=20,type="b")}
> text(300,200,"Growth")
```

Orange: 5 orange trees in 7 time points (tree, age, circumference).

Marks




```
> plot(0:10,0:10,type="n",main="Marks",xlab="",ylab="")
> k <- -1
> for(i in c(2,5,8)){for(j in 0:9){
  k <- k+1;text(i-0.75,j,k);points(i,j,pch=k,cex=2)}}
```

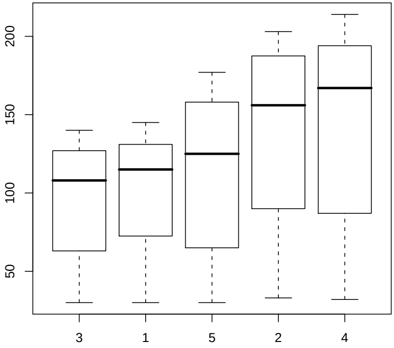
Colors

Spectral (divergent)

```
> colors()
  [1] "white"                   "aliceblue"               "antiquewhite"
 .....
[655] "yellow3"                 "yellow4"                 "yellowgreen"
> library(RColorBrewer)
> display.brewer.pal(11,'Spectral')
> help(rgb); help(palette); help(RColorBrewer)
```

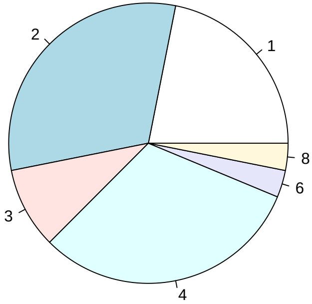
Escaping RGBland

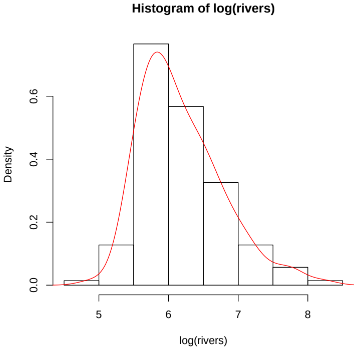
Categorical : numerical




```
> plot(Orange$Tree,Orange$circumference)
```

Categorical : numerical






```
> table(mtcars$carb)
 1  2  3  4  6  8
 7 10  3 10  1  1
> barplot(table(mtcars$carb))
> pie(table(mtcars$carb))
```

```
> dotchart(table(mtcars$carb))
> stripchart(mtcars$carb,method="stack",pch=16)
> hist(log(rivers),prob=TRUE)
> lines(density(log(rivers)),col="red")
```

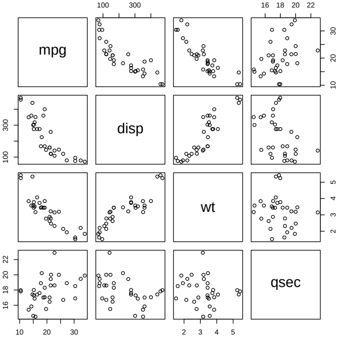
mtcars: 32 automobiles from the 1974 Motor Trend US magazine; carb – number of carburetors
rivers: the lengths (in miles) of 141 “major” rivers in North America

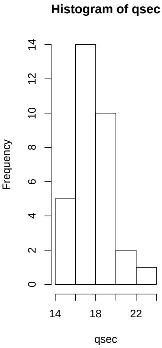
Different displays

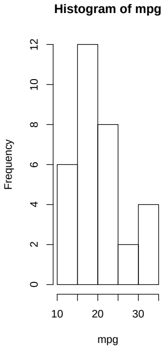
```
> attach(faithful)
> hist(waiting)
> summary(waiting)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   43.0    58.0    76.0    70.9    82.0    96.0
> bins <- seq(42,109,by=10)
> bins
[1]  42  52  62  72  82  92 102
> freqs <- table(cut(waiting,bins))
> y <- c(0,freqs,0)
> x <- seq(37,107,by=10)
> plot(x,y,type="l")
> rug(waiting)
> hist(waiting,breaks="Scott",prob=TRUE,ylab="",main="Faithful")
> lines(density(waiting),col="blue",lwd=2)
> boxplot(rivers)
> plot(rev(rivers[order(rivers)]))
> boxplot(rivers)
> f <- fivenum(rivers)
> f
[1]  135  310  425  680 3710
> text(rep(1.3,5),f,labels=c("min","1/4","1/2","3/4","max"))
```

faithful: waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone: (eruptions, witing) .

Relations among variables



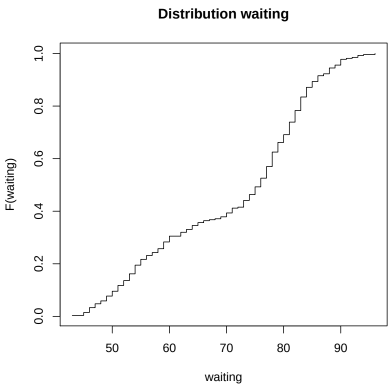





```
> attach(mtcars)
> pairs(mtcars[,c(1,3,6,7)])
> par(mfrow=c(1,2))
> hist(qsec,breaks="scott")
> hist(mpg,breaks="scott")
> par(mfrow=c(1,1))
```

mtcars: mpg – miles/(US) gallon; disp – displacement (cu.in.); wt – weight (1000 lbs); qsec – 1/4 mile time

Distribution using step function



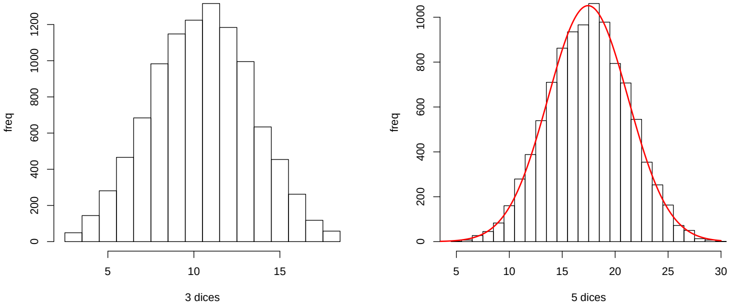

```
> attach(faithful)
> n <- length(waiting)
> plot(sort(waiting),(1:n)/n,type="s",xlab="waiting",
+ ylab="F(waiting)",main="Distribution waiting")
> plot(ecdf(waiting)) # empirical cumulative distribution func.
```

Distributions in R

Most of the standard distributions is available in R as functions. For a distribution *dist* are: `d`*dist* – density $g(x)$, `p`*dist* – cumulative $F(x) = \int_{-\infty}^{x} g(t)dt$, `q`*dist* – inverse – quantille function $q = F^{-1}(p)$, `r`*dist* – random numbers distributed according to *dist*.

Examples of *dist* (use `help`): `unif`, `beta`, `binom`, `cauchy`, `exp`, `chisq`, `f`, `gamma`, `geom`, `hyper`, `lnorm`, `logis`, `nbinom`, `norm`, `pois`, `signrank`, `t`, `weibull`, `wilcox`. The function `sample` supports random sampling (`replace=TRUE`) from a given set.

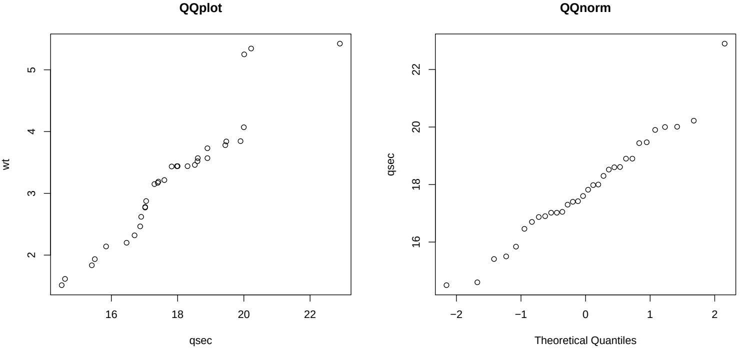
Central limit theorem




```
> a <- sample(1:6,replace=TRUE,10000); b <- sample(1:6,replace=TRUE,10000)
> c <- sample(1:6,replace=TRUE,10000); s <- a+b+c
> hist(s,breaks=2.5:18.5,xlab="3 dices",ylab="freq",main="")
> d <- sample(1:6,replace=TRUE,10000); e <- sample(1:6,replace=TRUE,10000)
> s <- s+d+e; x <- seq(1,30,0.1)
> hist(s,breaks=4.5:30.5,xlab="5 dices",ylab="freq",main="")
> lines(x,dnorm(x,mean(s),sd(s))*10000,lwd=2,col="red")
```

Comparing distributions

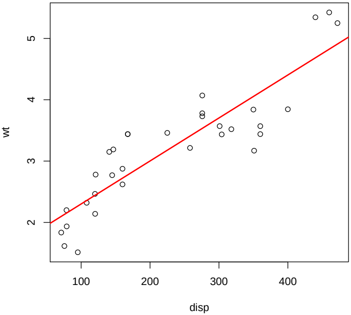
`QQplot` consists of points (x, y) over the domains of distributions F_1 and F_2, such that $F_1(x) = F_2(y)$. For equal distributions they lie on the diagonal. In function `QQnorm` the distribution F_1 is normal.




```
> attach(mtcars)
> qqplot(qsec,wt,main="QQplot")
> qqnorm(qsec,ylab="qsec",main="QQnorm")
```

Models

With an expression $y \sim f(x_1, x_2, \ldots, x_k)$ we describe a *model* – relation between dependent variable and independent variables. There exist some functions that on the basis of data determine (parameters of) the function f optimizing some fit criterion: `lm`, `gam`, `loess`, `lowess`, ... The values of the model function in selected points are obtained using the function `predict`. The simplest model is the *regression* line:




```
> attach(mtcars)
> res <- lm(wt ~ disp)
> res[[1]]
(Intercept)        disp
1.599814597 0.007010325
> plot(wt ~ disp)
> abline(res,col="red",lwd=2)
> predict(res,list(disp=c(410,200)))
        1        2
 4.474048 3.001880
```

Fitting the data

From the selected class of functions $\mathcal{F}$ we would like to select one that fits the best our data (x_k, y_k), $k \in I$. Let's denote it with $f(x, a)$. a are parameters. The error in a point (x_k, y_k) is equal to

$$y_k = f(x_k, a) + \varepsilon_k$$

These errors can be combined into a *total error* $E(f)$ in different ways

$$\begin{aligned}
E_1(f) &= \sum_k |\varepsilon_k| \\
E_2(f) &= \sum_k \varepsilon_k^2 \\
E_3(f) &= \max_k |\varepsilon_k| \\
E_4(f) &= \mathrm{lik}(f) = \prod_k f(x_k, a), \quad f \text{ is a distribution}
\end{aligned}$$

First three $\min$; E_4 $\max$.

Fitting

Instead with ε_k we can measure the point error also using some other quantities – *ortogonal* error ϱ_k.

For fitting distributions the *maximum likelihood* (E_4) is usually used..

For general functions the *least squares method* (E_2) is used. In many cases it allows to get the solution analitically. Its main weakness is that it is very sensitive to outliers. Using computers also other, more robust methods became an option.

Weighted fitting

$$E(a) = \sum_i w_i \varepsilon_i^2 = \sum_{i=1}^{n} w_i (f(x_i, a) - y_i)^2$$

Measurements with precision $y_i \pm \sigma_i$; then $\varepsilon'_i = \frac{\varepsilon_i}{\sigma_i}$

$$E'(a) = \sum_i (\varepsilon'_i)^2 = \sum_i (\frac{\varepsilon_i}{\sigma_i})^2 = \sum_i \frac{1}{\sigma_i^2} \varepsilon_i^2$$

Therefore $w_i = \frac{1}{\sigma_i^2}$.
Relative error: $y_i = f(x_i)(1 + \delta_i)$

$$\delta_i = \frac{y_i - f(x_i)}{f(x_i)} \approx \frac{y_i - f(x_i)}{y_i} \Rightarrow w_i = \frac{1}{y_i^2}$$

Is there a functional relation between given variables?

Let $p(X) = (p(x_i))_{i=1}^n$ be a discrete pobability distribution. Its *entropy* is defined as

$$H(X) = -\sum_{i=1}^{n} p(x_i) \lg p(x_i)$$

where $\lg \equiv \log_2$ and $p = 0 \Rightarrow p \lg p = 0$.

It holds $0 \leq H(X) \leq \lg n$. For $p(x_k) = 1;\ p(x_i) = 0,\ i \neq k$ we have $H = 0$; and for $p(x_i) = \frac{1}{n},\ i = 1, \ldots, n$ we get $H = \lg n$. The *normalized entropy* $h(X) = \frac{H(X)}{\lg n}$ has values in $[0, 1]$.

For discrete variables X and Y with distributions $p(X)$ and $p(Y)$ and joint probability distribution $p(XY)$ their *information* is

$$I(X, Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i, y_j) \lg \frac{p(x_i, y_j)}{p(x_i)p(y_j)}$$

Considering $\sum_{j=1}^m p(x_i, y_j) = p(x_i)$ and $\sum_{i=1}^n p(x_i, y_j) = p(y_j)$ we get

$$I(X, Y) = H(X) + H(Y) - H(XY)$$

Raiski's coefficient

Information $I(X, Y)$ has value 0 iff we have for all pairs $p(x_i, y_j) = p(x_i)p(y_j)$ – X and Y are independent.

The other extreme is attained iff X and Y are functionally related – in each row and each column of the distribution there is at most one nonempty cell, $H(X) = H(Y) = H(XY) = I(X, Y)$.

In 1964 Raiski introduced a coefficient

$$R(X \leftrightarrow Y) = \frac{I(X, Y)}{H(XY)} \quad \text{or in directed version} \quad R(X \to Y) = \frac{I(X, Y)}{H(Y)}$$

Both take values in $[0, 1]$ and have value 0 when X and Y are independent

$R(X \to Y) = 1$, when Y is a function of X; $R(X \leftrightarrow Y) = 1$, when the variables are linked one-to-one.

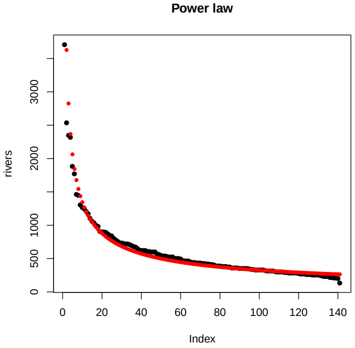
The Raiski's coefficient is defined for **all types of scales**.

Power law (Zipf, Lotka, Pareto)

The model function is selected in different ways: availability of a tool, simplification, guess – similarity to a curve on the picture, on theoretical basis (laws in the field), etc.

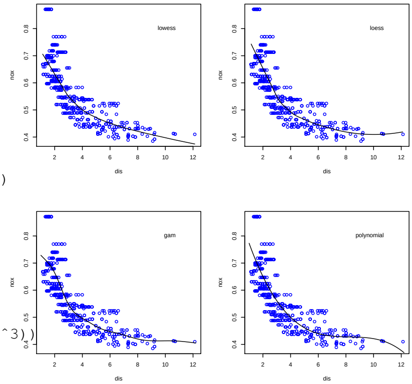
In double-logarithmic scale a *power law* curve is a line. Therefore we can determine its coefficients (little cheating) using the regression line:

```
> plot(rev(sort(rivers)))
> plot(rev(sort(rivers)),log="xy")
> x <- log(1:length(rivers))
> y <- log(rev(sort(rivers)))
> plot(y ~ x)
> rp <- lm(y ~ x)
> (a <- rp[[1]])
(Intercept)           x
  8.6233680  -0.6160568
> abline(rp,col="red",lwd=2)
> plot(rev(sort(rivers)),ylab="rivers",
+ pch=16,main="Power law")
> pow <- function(x){exp(a[1])*x^a[2]}
> x <- 1:length(rivers)
> y <- pow(x)
> points(x,y,pch=20,col="red")
```



Nonparametric smoothing / Boston


```
> library(MASS); attach(Boston)
> pairs(Boston)
> plot(dis,nox); s <- order(dis)
> plot(dis,nox,col="blue")
> lines(dis[s],nox[s])
> par(mfrow=c(2,2),cex=0.5)
> plot(dis,nox,col="blue")
> text(11,0.8,"lowess",pos=2)
> lines(lowess(dis,nox))
> plot(dis,nox,col="blue")
> text(11,0.8,"loess",pos=2)
> model <- loess(nox ~ dis)
> x <- seq(1,12.2,0.05)
> y <- predict(model,data.frame(dis=x))
> lines(x,y)
> plot(dis,nox,col="blue")
> text(11,0.8,"gam",pos=2)
> library(mgcv)
> model <- gam(nox ~ s(dis))
> y <- predict(model,list(dis=x))
> lines(x,y)
> plot(dis,nox,col="blue")
> text(11,0.8,"polynomial",pos=2)
> model <- lm(nox ~ dis+I(dis^2)+I(dis^3))
> y <- predict(model,list(dis=x))
> lines(x,y)
> par(mfrow=c(1,1),cex=1)
```



Boston: data frame has 506 rows (suburbs) and 14 columns; dis – weighted mean of distances to five Boston employment centres; nox – nitrogen oxides concentration (parts per 10 million).

Fitting OECD data pcinc ~ agr

EDA, clean and explore

V. Batagelj

Cleaning

Exploring

Regression

Clustering

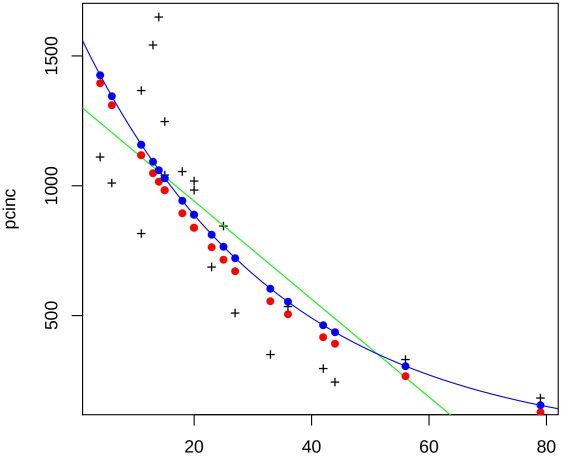
Solving the clustering problem

OECD data

```
> oecd <- read.table("OECD.dat",header=TRUE)
> pairs(oecd); attach(oecd)
> plot(agr,pcinc,pch="+")
> # linear regression
> lin <- lm(pcinc ~ agr)
> abline(lin,col="green")
> lp <- lin$coef[2]*agr + lin$coef[1]
> sum((lp - pcinc)^2)
> # exponential with linear regression
> pi <- log(pcinc); m <- lm(pi ~ agr )
> b <- exp(m$coef[1]); a <- exp(m$coef[2])
> pl <- function(x){b*a^x}
> points(agr,pl(agr),col="red",pch=16)
> # least squares
> f <- function(t,p){a <- p[1]; b <- p[2]; b*a^t}
> E <- function(p){d <- f(agr,p) - pcinc; sum(d^2)}
> p0 <- c(a,b); best <- optim(p0,E)
> E(p0)
> best
> pr <- function(x){f(x,best$par)}
> points(agr,pr(agr),col="blue",pch=16)
> d <- seq(0,84,2); lines(spline(d,pr(d)),col="blue")
```

OECD: 20 countries: pcinc – per capita income; agr – percentage of agrarian

Fitting OECD



Clustering

Given a set of units $\mathcal{U}$ the clustering is a process of organizing units into groups – clusters of similar units. In real life clustering problems we have to deal with different theirs characteristics:

- description of units: vectors (types of measurement scales, number of variables, missing values, . . .) or structured units;
- size of the set of units;
- structure of units ”space” (density, shapes of clusters).

A recent book on clustering in R is the ”Practical Guide to Cluster Analysis in R” by Alboukadel Kassambara (2017).

Clustering and optimization

We approach the clustering problem as an optimization problem over the set of *feasible* clusterings Φ_k – partitions of units into k clusters. A cluster is a nonempty subset of the set of unit $\mathcal{U}$. The *criterion function* has the following form

$$P(\mathbf{C}) = \sum_{C \in \mathbf{C}} p(C)$$

The *total error* $P(\mathbf{C})$ of the clustering $\mathbf{C} = \{C_1, C_2, \ldots, C_k\}$ is a sum of *cluster errors* $p(C)$.

There are many possibilities how to express the cluster error $p(C)$. Here we shall assume a model in which the error of a cluster is a sum of differences of its units from the cluster's *representative* T

$$p(C, T) = \sum_{X \in C} d(X, T)$$

Note that in general the representative needs not to be from the same "space" (set) as units.

Representatives, dissimilarities

The best representative is called a *leader*

$$T_C = \operatorname*{argmin}_T p(C, T)$$

Then we define

$$p(C) = p(C, T_C) = \min_T \sum_{X \in C} d(X, T)$$

In most cases we express the cluster error in terms of a *dissimilarity* between units $d(X, Y)$; $d(X, X) = 0$ and $d(X, Y) = d(Y, X)$.

Another example of cluster error is a diameter

$$p(C) = \text{diam}(C) = \max_{X, Y \in C} d(X, Y)$$

Dissimilarities on $\mathbb{R}^m$ / examples 1

n	measure	definition	range	note
1	Euclidean	$\sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}$	$[0, \infty)$	$M(2)$
2	Sq. Euclidean	$\sum_{i=1}^{m}(x_i - y_i)^2$	$[0, \infty)$	$M(2)^2$
3	Manhattan	$\sum_{i=1}^{m}\lvert x_i - y_i\rvert$	$[0, \infty)$	$M(1)$
4	rook	$\max_{i=1}^{m}\lvert x_i - y_i\rvert$	$[0, \infty)$	$M(\infty)$
5	Minkowski	$\sqrt[p]{\sum_{i=1}^{m}(x_i - y_i)^p}$	$[0, \infty)$	$M(p)$

Dissimilarities on $\mathbb{R}^m$ / examples 2

n	measure	definition	range	note
6	Canberra	$\sum_{i=1}^{m} \frac{\|x_i - y_i\|}{\|x_i + y_i\|}$	$[0, \infty)$	
7	Heincke	$\sqrt{\sum_{i=1}^{m} (\frac{\|x_i - y_i\|}{\|x_i + y_i\|})^2}$	$[0, \infty)$	
8	Self-balanced	$\sum_{i=1}^{m} \frac{\|x_i - y_i\|}{\max(x_i, y_i)}$	$[0, \infty)$	
9	Lance-Williams	$\frac{\sum_{i=1}^{m} \|x_i - y_i\|}{\sum_{i=1}^{m} x_i + y_i}$	$[0, \infty)$	
10	Correlation c.	$\frac{\mathrm{cov}(X, Y)}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}}$	$[1, -1]$	

(Dis)similarities on $\mathbb{B}^m$ / examples

Let $\mathbb{B} = \{0, 1\}$. For $X, Y \in \mathbb{B}^m$ we define $a = XY$, $b = X\overline{Y}$, $c = \overline{X}Y$, $d = \overline{X}\overline{Y}$. It holds $a + b + c + d = m$. The counters a, b, c, d are used to define several (dis)similarity measures on binary vectors.

In some cases the definition can yield an indefinite expression $\frac{0}{0}$. In such cases we can restrict the use of the measure, or define the values also for indefinite cases. For example, we extend the values of Jaccard coefficient such that $s_4(X, X) = 1$. And for Kulczynski coefficient, we preserve the relation $T = \frac{1}{s_4} - 1$ by

$$s_4 = \begin{cases} 1 & d = m \\ \frac{a}{a+b+c} & \text{otherwise} \end{cases} \qquad s_3^{-1} = T = \begin{cases} 0 & a = 0,\ d = m \\ \infty & a = 0,\ d < m \\ \frac{b+c}{a} & \text{otherwise} \end{cases}$$

We transform a similarity s from $[1, 0]$ into dissimilarity d on $[0, 1]$ by $d = 1 - s$.

For details see Batagelj, Bren (1995).

(Dis)similarities on $\mathbb{B}^m$ / examples 1

n	measure	definition	range
1	Russel and Rao (1940)	$\frac{a}{m}$	$[1, 0]$
2	Kendall, Sokal-Michener (1958)	$\frac{a+d}{m}$	$[1, 0]$
3	Kulczynski (1927), T^{-1}	$\frac{a}{b+c}$	$[\infty, 0]$
4	Jaccard (1908)	$\frac{a}{a+b+c}$	$[1, 0]$
5	Kulczynski	$\frac{1}{2}(\frac{a}{a+b} + \frac{a}{a+c})$	$[1, 0]$
6	Sokal & Sneath (1963), un_4	$\frac{1}{4}(\frac{a}{a+b} + \frac{a}{a+c} + \frac{d}{d+b} + \frac{d}{d+c})$	$[1, 0]$
7	Driver & Kroeber (1932)	$\frac{a}{\sqrt{(a+b)(a+c)}}$	$[1, 0]$
8	Sokal & Sneath (1963), un_5	$\frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	$[1, 0]$

(Dis)similarities on $\mathbb{B}^m$ / examples 2

n	measure	definition	range
9	Q_0	$\frac{bc}{ad}$	$[0, \infty]$
10	Yule (1927), Q	$\frac{ad-bc}{ad+bc}$	$[1, -1]$
11	Pearson, ϕ	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}$	$[1, -1]$
12	– bc –	$\frac{4bc}{m^2}$	$[0, 1]$
13	Baroni-Urbani, Buser (1976), S^{**}	$\frac{a+\sqrt{ad}}{a+b+c+\sqrt{ad}}$	$[1, 0]$
14	Braun-Blanquet (1932)	$\frac{a}{\max(a+b,a+c)}$	$[1, 0]$
15	Simpson (1943)	$\frac{a}{\min(a+b,a+c)}$	$[1, 0]$
16	Michael (1920)	$\frac{4(ad-bc)}{(a+d)^2+(b+c)^2}$	$[1, -1]$

Dissimilarities between sets

Let $\mathcal{F}$ be a finite family of subsets of the finite set U; $A, B \in \mathcal{F}$ and let $A \oplus B = (A \setminus B) \cup (B \setminus A)$ denotes the symmetric difference between A and B.

The 'standard' dissimilarity between sets is the *Hamming distance*:

$$d_H(A, B) := \mathrm{card}(A \oplus B)$$

Usually we normalize it $d_h(A, B) = \frac{1}{M}\mathrm{card}(A \oplus B)$. One normalization is $M = \mathrm{card}(U)$; the other $M = m_1 + m_2$, where m_1 and m_2 are the first and the second largest value in $\{\mathrm{card}(X) : X \in \mathcal{F}\}$.

Other dissimilarities

$$d_s(A, B) = \frac{\mathrm{card}(A \oplus B)}{\mathrm{card}(A) + \mathrm{card}(B)} \qquad d_u(A, B) = \frac{\mathrm{card}(A \oplus B)}{\mathrm{card}(A \cup B)}$$

$$d_m(A, B) = \frac{\max(\mathrm{card}(A \setminus B), \mathrm{card}(B \setminus A))}{\max(\mathrm{card}(A), \mathrm{card}(B))}$$

For all these dissimilarities $d(A, B) = 0$ if $A = B = \emptyset$.

Problems with dissimilarities

Functions in R: `dist`, `cluster/daisy`

What to do in the case of *mixed units* (with variables measured in different types of scales)?

- conversion to a common scale
- compute the dissimilarities on homogeneous parts and combine them (Gower's dissimilarity)

Fairness of dissimilarity – all variables contribute equally. Approaches: use of normalized variables, analysis of dependencies among variables.

Gower's dissimilarity

the Gower dissimilarity coefficient for a mix of variables

$d_{ij} = \sum_{v=1}^{m} \frac{\delta_{ijv} d_{ijv}}{\sum_{i=1}^{m} \delta_{ijv}}$

where δ_{ijv} is a binary indicator equal to one whenever both observations i and j are nonmissing for variable v, and zero otherwise. Observations with missing values are not included.

For binary and nominal variables v, $d_{ijv} = 0$ if $x_{iv} = x_{jv}$; and $d_{ijv} = 1$ otherwise.

Ordinal variables v are considered as categorical ordinal variables and the values are substituted with the corresponding position index, r_{iv} in the factor levels. These position indexes are transformed in the following manner $z_{iv} = \frac{r_{iv}-1}{\max_k r_{kv}-1}$ These new values, z_{iv}, are treated as observations of an interval scaled variable.

For continuous variables v,

$d_{ijv} = \frac{|x_{iv} - x_{jv}|}{\max_k(x_{kv}) - \min_k(x_{kv})}$

d_{ijv} is set to 0 if $\max_k(x_{kv}) = \min_k(x_{kv})$.

Functions `cluster/daisy` and `StatMatch/gower.dist`.

Solving the clustering problem

Finite - solution always exists, but in most cases algorithmically hard problem $\rightarrow$ heuristics.

- hierarchical
  - agglomerative methods (`hclust`, `cluster/agnes`, `amap/hcluster`, `amap/hclusterpar`)
  - divisive methods (`cluster/diana`, `cluster/mona`)
  - adding methods
- local optimization (leaders method) (`kmeans`, `cluster/pam`, `cluster/clara`, `cluster/fanny`)
- linear algebra methods
- graph theory methods
- other methods (`mclust/Mclust`, `fpc/dbscan`, `dbscan/dbscan`, `factoextra/hkmeans`)

Acronyms

Agnes - Agglomerative Nesting
Diana - Divisive Analysis
PAM - Partitioning around medoids
CLARA - Clustering Large Applications
hkmeans - Hierarchical K-means
FANNY - Fuzzy analysis clustering
Mclust - Model based clustering
DBSCAN - Density-Based Clustering

Leaders method

Leaders method is a generalization of a popular nonhierarchical clustering k-means method.
The idea is to get ”optimal” clustering into a pre-specified number of clusters with the following iterative procedure:

determine an initial clustering
repeat
  determine leaders of the clusters in the current clustering;
  assign each unit to the nearest new leader – producing a new clustering
until the leaders stabilize.

Hierarchical agglomerative clustering

The *hierarchical agglomerative clustering* procedure is based on a step-by-step merging of the two closest clusters.

each unit forms a cluster: $\mathbf{C}_n = \{\{X\} : X \in \mathcal{U}\}$;
they are at level 0: $h(\{X\}) = 0, \; X \in \mathcal{U}$;
for $k = n - 1$ **to** 1 **do**
  determine the closest pair of clusters
    $(u, v) = \operatorname{argmin}_{i,j:\, i \neq j}\{D(C_i, C_j) : C_i, C_j \in \mathbf{C}_{k+1}\}$;
  join the closest pair of clusters $C_{(uv)} = C_u \cup C_v$
    $\mathbf{C}_k = (\mathbf{C}_{k+1} \setminus \{C_u, C_v\}) \cup \{C_{(uv)}\}$;
    $h(C_{(uv)}) = D(C_u, C_v)$
    determine the dissimilarities $D(C_{(uv)}, C_s), C_s \in \mathbf{C}_k$
endfor

$\mathbf{C}_k$ is a partition of the finite set of units $\mathcal{U}$ into k clusters.
The level $h(C)$ of the cluster $C_{(uv)} = C_u \cup C_v$.

Methods

Hierarchical methods differ in selection of a between cluster dissimilarity D:

- **single linkage**: $D(C_p, C_q) = \min_{X \in C_p, Y \in C_q} d(X, Y)$
- **complete linkage**: $D(C_p, C_q) = \max_{X \in C_p, Y \in C_q} d(X, Y)$
- **Ward**: $D(C_p, C_q) = \frac{n_p \cdot n_q}{n_p + n_q} d(T_p, T_q)$
- see `help` and paper