107.258 Computerstatistik

Exercise sheet

Laura Vana

For each submission, upload Rmarkdown file (.Rmd) and a PDF in TUWEL. Each task and subtask should have a conclusion using full sentences!

Task 1

Use the different help systems available for R (i.e., R homepage, built-in R functions) to find information about:

1. the command `[`
2. the `ordered` argument for `factor`
3. a function to create normally distributed random variables
4. a function that does conditional logistic regression
5. what a vignette is
6. which vignettes are on your R installation available, and access one of them

Task 2

Calculate the sum $s_n = \sum_{i=1}^{n} r^i$, for $r = 1.08$ and compare it to $(r^{n+1} - 1)/(r - 1) - 1$ for $n = 10, 20, 30, 40$. Use the formula to compute the sum for all values of n between 1 and 100, and store the results in a vector.

Task 3

Create the following vectors.
(Hint: the following functions will be useful here: `seq`, `rep`, `rev`, `cumsum`, `cumprod`, `cut`)

```
x1:  1  3  3  6  6  6 10 10 10 10 15 15 15 15 15
x2:  0.00000000  0.07142857  0.14285714  0.21428571 0.28571429 0.35714286
     0.42857143  0.50000000  0.57142857  0.64285714 0.71428571 0.78571429
     0.85714286  0.92857143  1.00000000
x3: 0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60
    0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00
x4: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
x5: 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1
x6: 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
x7: 15  29  42  54  65  75  84  92  99 105 110 114 117 119 120
x8: 1.000000e+00  2.000000e+00  6.000000e+00  2.400000e+01 1.200000e+02
```

```
    7.200000e+02  5.040000e+03  4.032000e+04  3.628800e+05 3.628800e+06
    3.991680e+07  4.790016e+08  6.227021e+09  8.717829e+10 1.307674e+12
x9: 120 119 117 114 110 105  99  92  84  75  65  54  42  29  15
x10: "a" "b" "b" "b" "b" "c" "c" "c" "c" "c" "c" "c" "c" "d" "d"
x11: Define x11 as a factor which is based on x5
X12: Define x12 as a vector of factors based on x10, but include that there could
     be also a level "e"
X13: Create a vector x13 which is a factor vector based on x2:
    - values less then 0.37 belong to level "less"
    - between 0.37 and 0.55 to level "sufficient"
    - between 0.55 and 0.96 to the level "good"
    - above 0.96 to the level "plenty"
```

Task 4

a.

Explain the following:

- Why is `1L == "1"` TRUE
- Why is `-2 < FALSE` TRUE
- Why is `"one" < 2` FALSE

b.

What does the following code return? Why?

```
a <- c(TRUE, TRUE, TRUE)
b <- c(1, 2, 3)
a & (b - 2)
```

c.

What is the difference between the following chunks of code? Explain. *Hint:* look up the difference between `&` and `&&`.

```
x <- -7
x > 0 & sqrt(x) < 2
```

```
x <- -7
x > 0 && sqrt(x) < 2
```

Task 5

Explain what the following code does

```r
set.seed(1)
DAT <- sample(LETTERS[1:7], 15, replace = TRUE)
F1 <- factor(DAT, levels = (LETTERS[1:7]))
F2 <- F1
levels(F2) <- rev(levels(F2))
F3 <- rev(factor(DAT, levels = (LETTERS[1:7])))
F4 <- factor(DAT, levels = rev(LETTERS[1:7]))
```

Is there a difference between `F2`, `F3` and `F4`? If yes, what is the difference?

Task 6

a.

What does the function `dim` return when applied to an atomic vector? What happens if the dim attributed of a matrix or an array is assigned the value `NULL`?

b.

Read carefully the help for the function `scale`.

Create the a matrix `X` with 100 rows and three columns such that $x_i \sim N(\mu, \Sigma)$ where $\mu = (1, 2, 3)^\top$ is the mean vector and $\Sigma = diag(1, 2, 3)$ is the variance-covariance matrix. You cannot use the function `matrix` here but only the functions `c`, `rnorm` and `dim` to create X. Set a random seed for reproducibility using `set.seed(1234)`.

- give X row names r1,...,r100 and column names c1,c2,c3. *Hint:* use the function `paste`.
- compute:
  - `X1 <- scale(X, TRUE, TRUE)`
  - `X2 <- scale(X, TRUE, FALSE)`
  - `X3 <- scale(X, FALSE, TRUE)`
  - `X4 <- scale(X, FALSE, apply(X, 2, sd))`. Here `apply(X, 2, sd)` will compute the standard deviation for each column of X.
- compare the attributes of X, X1, X2, X3, X4.
- explain why X3 and X4 differ.

c.

What does `as.matrix` do to a data frame which has columns of different type?

Task 7

Unlike some other programming languages, base R does not have a dedicated data type for sets. Instead, R treats a vector like a set by taking only its distinct elements. The set operators `setdiff`, `intersect`, `union`, `setequal` and `%in%` are available in base R (see `?union`). (*Note*: For `v %in% S`, only `S` is treated as a set, however, not the vector `v`). There are however packages such as package `sets` on CRAN which specialize on set operations.

a.

Consider the following vector objects in R:

```
set.seed(1234)
x <- c(1, 3, 4)
y <- sample(1:100, 10)
```

Compute their union, intersection and the set of elements in `y` but not in `x`. Check for each element of `x` whether it can be found in `y`.

b.

For `s <- 1:200`, use logical operators to find the subvector of `s` that is divisible by 7, but not divisible by 2. *Hint:* use `%%` as the modulo operator; `which()` can be used to check which positions in a logical vector are equal to TRUE.

c.

Create a vector `s7` which contains all the numbers divisible by 7 in the vector of integers between 1 and 200. Create a vector `s2` which contains all the numbers divisible by 2 in the vector of integers between 1 and 200 (*Hint*: you can use `seq`). Use the set operators mentioned above to find the subvector of `1:200` that is divisible by 7, but not divisible by 2.

Task 8

a.

Find the smallest normalized positive IEEE 754 64-bit floating-point number ϵ for which $1 + \epsilon \neq 1$ ("pen and paper", not in R). Compare the number you get with the machine epsilon `.Machine$double.eps`. Also, try checking `1 + epsilon == 1` in R.

b.

Find the smallest normalized positive IEEE 754 64-bit floating-point number ϵ for which $1 - \epsilon \neq 1$ ("pen and paper", not in R). Compare the number you get with the machine epsilon `.Machine$double.neg.eps`. Also, try checking `1 - epsilon == 1` in R.

c.

The machine precision or machine epsilon is an upper bound on the relative approximation error due to rounding in floating point arithmetic: $fl(x) = x(1+\xi), |\xi| \leq \epsilon$. Explain why the following commands deliver different results, based on the subtasks a and b above. Which basic law of arithmetic is not satisfied by floating point arithmetic in the example below?

```
.Machine$double.eps/2 + 1 - 1
```

```
## [1] 0
```

```
.Machine$double.eps/2 + (1 - 1)
```

```
## [1] 1.110223e-16
```

Task 9

a.

Write a function `sum_for_sort` which takes a vector of length greater than one and first sorts the vector in ascending order and then, by using a `for` loop, computes the sum of the sorted vector.

Call the function for the sequence:

```
set.seed(1)
x <- rnorm(1e6)
```

Compare the result with the sum computed using the built-in function `sum(x)` and with the sum computed using 80 bit precision by employing tools from the multi-precision arithmetic R package **Rmpfr**.

```
# install.packages("Rmpfr")
library("Rmpfr")
s80 <- sum(mpfr(x, 80))
```

Briefly discuss the results.

b.

Write two functions to compute the binomial coefficient $\binom{n}{k}$, once using `prod()` for computing the needed factorials $n! = \prod_{i=1}^{n} i$. and the second using `log()`, `exp()` and `sum()`. Evaluate the functions at the value $n = 1000$ and $k = 500$. Discuss any differences, if any occur, and explain what you think is happening.

c.

Write a function with approximates the exponential function e^x at a value `x0` using the first `n` terms of the Taylor expansion.

- Compare the results of the function at $x_0 = 1$ and $n = 25$ with the results returned by `exp(x0)`.
- Compare the results of the function at $x_0 = -25$ and $n = 25$ with the results returned by `exp(x0)`.

Discuss any differences, if any occur, and explain what you think is happening.

Task 10

a.

Install the package `fueleconomy` using `install.packages("fueleconomy")` and make yourself familiar with the dataset `vehicles`. Are there observations with missing values? *Hint:* use `complete.cases()`.

b.

Obtain for each variable the type.

c.

How many unique values do the variables `class`, `trans`, `drive` and `fuel` have? Hint: check what the function `unique` does.

d.

Compute the frequency table of `drive` vs `fuel`. Compute the total, column and row percentages.

e.

Make a data frame named `FED` containing the variables `cyl`, `displ`, `hwy` and `cty`. Compute the median of all variables in `FED`, missing values should be removed for the computations.

f.

Subtract from each variable in `FED` the median.

g.

Compute the trimmed means for the variable `displ` by the variables `drive` and `fuel` (use `trim = 0.1` and recall the missing values).

h.

Make a factor `DRIVE` which takes the value `2WD` when the variable `drive` has the values `2-Wheel Drive`, `Front-Wheel Drive` or `Rear-Wheel Drive`. Otherwise the the factor should get the value `4WD`. *Hint*: check `%in%`.

i.

Make a data frame `REGULAR` which contains the factor created above and the variables `hwy` and `cty` - but only when the fuel type is `Regular`.

Task 11

Install the package `fivethirtyeight` using `install.packages("fivethirtyeight")` and make yourself familiar with the dataset `candy_rankings` using `?candy_rankings`.

a.

Explore the univariate distribution of some of the categorical variables in the data set i.e., compute the percentage of candies in each level of the categorical variable. Repeat for categorical variables `chocolate`, `caramel`, `bar`. You can use the `prop.table()` function for this purpose.

b.

Make barplots for each variable showing the percentages computed in a. Put the name of the variable as the plot title. Put the three plots side by side in one row.

c.

Generate a table which contains the number of candies for all all factor level combinations for the variables `chocolate`, `caramel`, `bar`, `fruity`, `peanutyalmondy`, `crispedricewafer`. Order the resulting table in decreasing order by the number of candies in each group. What is the most common "taste profile" among the candies in the sample? *Hint: look at aggregate() for generating the table.*

d.

Take the table in c. and for each row, generate a string containing the taste profile: e.g., for rows

```
   chocolate caramel   bar fruity peanutyalmondy crispedricewafer
       FALSE   FALSE FALSE   TRUE          FALSE            FALSE
        TRUE    TRUE FALSE  FALSE          FALSE            FALSE
```

the strings will be `"fruity"` and `"chocolate,caramel"`. For the row containing only FALSE values, use `""`. *Hint: you can use apply() and paste0(, collapse = ",").*

Make a new data frame which contains the taste profile variable and the number of observations in the sample.

e.

Inspect the `winpercent` variable (which proxies preference of customers).

- What are the top 5 candies?
- What are the bottom 5 candies?
- What is the mean, median and standard deviation of this variable
- Make a histogram and boxplot of this variable.

f.

Compute the correlation between the different quantitative variable. Discuss how the correlation is between `winpercent` and the other two.

g.

For the different taste profiles in c., compute the average `winpercent`. Which taste profile is the "best" and "worst", respectively?

Task 12

a.

Install the package `ISwR` and read the help for the data set `hellung`.

b.

Add to the data frame the variable GLUCOSE which is the corresponding factor of `glucose` which gives labels `Yes` and `No`. Make a reasonable plot that lets you compare the values of diameter for the two glucose groups.

c.

Plot `conc` against `diameter`, give different colors and plotting symbols depending on the variable glucose.

d.

Make same plot as in 3, but now use a logarithm scale for the x axis.

e.

Make same plot as in 4, but now use a logarithm scale for both axes. Add then also a legend to the plot, the text should say *glucose* and *no glucose*.

f.

Make a "publication" ready figure from 5. It should have at least a nice legend, proper axis labels and unnecessary space around the figure should be cut off. Write the code so that the figure will be saved as a pdf with size 7×7 and that it is saved in the working directory as `YourName.pdf`. Upload this .pdf together with the .Rmd and .PDF of the solution in TUWEL.

Task 13

Write a function which takes two vectors `x` and `y` of equal length and returns a scalar N.

For a collection of emails,

- The vector `y` records whether a certain email has been "spam" or "ham" (i.e., non-spam).
- The vector `x` records how many times capital letters appeared in each email.

We want to use `x` in order to predict whether a new email is spam, by selecting a cutoff N for the number of capital letters and classify new emails which have more than N capital letters as spam and all others as ham. Among an extensive list of possible cutoffs, the optimal cutoff is selected such that the number of wrongly classified `y`'s is minimal.

Find the optimal N for the following data:

```
y <- c("ham", "ham", "ham", "ham", "ham", "spam",
       "ham", "spam", "ham", "spam", "ham", "ham",
       "spam", "spam", "spam", "ham", "spam", "spam")
x <- c(0, 0, 0, 1, 1, 1,
       2, 2, 2, 2, 2, 4,
       5, 5, 6, 6, 8, 8)
```

In writing the function, assume that the possible cutoff values are only the ones appearing in `x` (this is restrictive of course in general), namely 0, 1, 2, 4, 5, 6, 8. Also, if there are ties in the misclassification rate, choose the smallest cutoff as the optimal one.

Task 14

a.

Download the data set `sdi.csv` from TUWEL and read it into R in an object called `sdi`. The original data set is available from gapminder and it contains the sustainable development index (SDI) for a collection of countries.

b.

For each year, compute the percentage of missing values in the SDI and plot them as a line plot with the years on the x-axis and the percentage on the y-axis. Label the axes of this plot sensibly.

c.

What is the percetage of countries with complete observations in the SDI?

d.

Compute the number of missings per country and store the results in `missings_per_country`. (*Hint:* you can use `rowSums`.) Assign the names of the countries as the names of this vector. Return from `missings_per_country` only the elements with at least one missing. sorted in decreasing order (i.e., first element of the vector corresponds to the country with most missings, etc.).

e.

Convert the data set, which is in wide format, to a long format. Name this new object `sdi_long`. Convert the year variable in long format data set to a factor.

f.

Eliminate from the data set `sdi_long` the rows with missing values.

g.

Generate a graph which displays parallel boxplots of the SDI for each year. Briefly describe the results.

h.

For each year, compute the average and the median SDI from the `sdi_long`. Generate a line plot with the years on the x-axis and the average SDI on the y-axis. Add to this plot a dashed line of the medians (*Hint:* check the `lty` argument of the `lines()` or `plot()` function). Give appropriate labels to the plot. Add a legend for the line types.

Task 15

In the following we will consider a sample of the latest movie lens data sets. Original data is available at https://grouplens.org/datasets/movielens/.

- `movies1.csv` in TUWEL contains the movie id, the title and the genre. In this data set there is one genre for each movie.
- `movies2.csv` in TUWEL contains the movie id, the title and the genre. In this data set there are multiple genres for each movie separated by a `|`.
- `ratings.csv` in TUWEL contains the movie id, a userid, the rating for movie and a timestamp.

a.

Combine the `movies1.csv` and the `ratings.csv` files. How many users didn't rate any of the movies in `movies1.csv`? What is the dimension of the combined data set if you keep all user ratings?

What is the dimension of the combined data set if you only keep the user ratings for the movies in `movies1.csv`?

b.

Combine the two movie data sets into one. Name this data `movies`.

c.

Combine the data set `movies` and the data in file `ratings.csv` such that all movies in `movies` are contained in the combined data set. How many movies did not receive any review?

d.

Write out the combined data set which only contains user ratings for the movies in `movies1.csv` as a text file `.txt` to your working directory, in the file missing values should appear as `999999`. The text file should not include a column for the row names.

Task 16

Use the `decathlon` data from package **FactoMineR**. Check the documentation of the data set.

a.

Make a correlation plot of the first 10 numeric variables in the data set. Briefly comment on the output.

b.

Plot a histogram of points obtained and overlay it with a kernel density estimate and a theoretical normal distribution using the sample mean and sample standard deviation as parameters.

Give the lines different colors, the axes reasonable labels and a reasonable plot title. Add also a legend which tells which curve is the kernel and which one the theoretic curve.

Draw also qqplots for points obtained against the quantiles of the normal distribution using the `qqnorm()` function. Add a `qqline`.

c.

Andrews curves can be used to visualize high-dimensional data. Each observation $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ in the sample gets mapped to:

$$\begin{aligned} f_i(t) &= \frac{x_{i1}}{\sqrt{2}} + x_{i2}\sin t + x_{i3}\cos t + x_{i4}\sin 2t + x_{i5}\cos 2t + \ldots \\ &= \frac{x_{i1}}{\sqrt{2}} + \sum_{1\le k\le \lfloor p/2\rfloor} x_{i,2k}\sin kt + \sum_{1\le k< \lfloor p/2\rfloor} x_{i,2k+1}\cos kt, \qquad -\pi \le t \le \pi \end{aligned}$$

One implementation in R for this function is:

```
f_t <- function(t, x) {
  p <- length(x)
  x[1]/sqrt(2) +
    sum(x[2 * (1:floor(p/2))] *
          sin((1:floor(p/2)) * t)) +
    sum(x[2 * (1:(floor(p/2)-1)) + 1] *
          cos((1:(floor(p/2) - 1)) * t))
}
```

In R, compute the curves for a sequence `tseq <- seq(-pi, pi, length = 100)` for each observation based on the first $p = 10$ columns. Using a `for` loop, plot the curves in a graph using `lines()`, where the type of curve and the color differs for 2004 Olympic Game and 2004 Decastar.

For better visualization, normalize the data before doing the computations, by subtracting the minimum value and dividing by the range (max - min) for each column.

Task 17

The `linelist_messy_dates.rds` file from TUWEL contains data simulated for the Ebola epidemic.

a.

Import the data set in R. Familiarize yourself with the data set and try to make sense of the variables in the data set. Briefly discuss which information is contained in this data set. What are the observational units contained in the rows? Are there missing values?

b.

Convert the all date columns into an object of class `Date`.

c.

Make a bar plot for the number of infections for each month of the year. Each bar must be labelled with the abbreviated name of the month (e.g., Jan, Feb, ...). Pay attention to the ordering of the bars! Briefly comment on the plot. *Hint: to extract the month from a date you can use function format().*

d.

For each row, compute the number of days between infection and onset and save this info in a new variable `date_infection_onset`. Make parallel boxplots of `date_infection_onset` for each month of the year. Make sure that the axis are readable. Briefly comment on the plot.

e.

Compute the daily number of hospitalizations. *Hint: you can use aggregate()*

Make parallel boxplots for the number of daily hospitalization split in two groups: work days (Mo-Fr) vs. weekend (Sa-Su). Are there any notable differences? *Hint: you can use weekdays()*

Task 18

Download and load the data `multivariate_data_outliers.rds` from TUWEL in R.

a.

Make a pairwise scatterplot of the data using `pairs` as well as a 3D-scatterplot using function `scatterplot3d()` of package **scatterplot3d**.

b.

For each column, identify the univariate outliers using the 1.5 * IQR rule used by the boxplot. Create a new column `univ_out` in the data set which contains 0 if the observation has not been identified as an outlier, 1 if observation is univariate outlier for column 1, 2 if observation is univariate outlier for column 2 and 3 if observation is univariate outlier for column 3.

c.

For each column, check normality using qqplots.

d.

Multivariate outliers can be identified by using the Mahalanobis distance (MD) between each point $x_i \in \mathbb{R}^p$ (i.e., row) and the mean vector $\mu \in \mathbb{R}^p$:

$$MD(x_i) = d(x_i, \mu) = \sqrt{(x_i - \mu)^\top \Sigma^{-1} (x_i - \mu)}$$

where Σ is a positive-definite $p \times p$ covariance matrix.

Note that the squared Mahalanobis distance is a sum of squares. The χ^2 distribution for MD^2 is justified if we assume that the x's are normally distributed. Moreover, it is assumed that μ and Σ are population level parameters. However, in practice we estimate them from the data. The estimated MH distance is:

$$\widehat{MD}(x_i) = d(x_i, \hat{\mu}) = \sqrt{(x_i - \hat{\mu})^\top \hat{\Sigma}^{-1} (x_i - \hat{\mu})}$$

An observation is declared a multivariate outlier if $MD^2(x_i) > Q_\alpha$, where Q_α is the α quantile of the χ^2 distribution with p degrees of freedom (typical α values are 0.95 or 0.975).

- Estimate the squared MD for each observation in the data set using the sample mean and the sample covariance matrix of the data. Add it as a new column `MD2` to the data set. *Hint: In R, you can use mahalanobis()*
- For $\alpha = 0.975$, make a column `MD_out` which contains a 1 if the observation is declared as outlier and 0 otherwise.

e.

The mean and covariance estimators used in estimating MD are sensitive to outliers, and can therefore be contaminated easily. Repeat the exercise in d. but now use robust estimators for the mean and the covariance parameter. A robust estimator of location is the median. A robust estimator for the covariance is the MCD (Minimum Covariance Determinant) estimator which can be computed using the function `cov.mcd()` from package **MASS**.

```
# install.packages("MASS")
library("MASS")
robust_cov <- cov.mcd(x[,1:p])$cov
```

Name the new columns `MD2rob` and `MDrob_out`.

f.

Create 3 pairwise scatterplots and 3 scatterplots in 3D where you visualize the different outliers:

- one pairwise and one 3D scatterplot where you color the univariate outliers with 3 different colors (for 1,2,3).
- one pairwise and one 3D scatterplot where you color the multivariate outliers based on MD with one different color.
- one pairwise and one 3D scatterplot where you color the multivariate outliers based on robust MD with one different color.

Discuss the results. How many outliers are identified by each of the methods?

g.

Another distance which can be used for outlier detection is the Euclidean distance between the point and the center parameter.

$$\sqrt{\left(\frac{x_{i1}-\mu_1}{\sigma_1}\right)^2+\ldots+\left(\frac{x_{ip}-\mu_p}{\sigma_p}\right)^2}=\sqrt{(x_i-\mu)^\top \text{diag}(\sigma_1^2,\ldots,\sigma_p^2)^{-1}(x_i-\mu)}$$

where $\mu=(\mu_1,\ldots\mu_p)^\top$ and the standard deviations $\sigma=(\sigma_1,\ldots\sigma_p)^\top$ have to again be estimated from the data. Note that, given that the variables can have different scales, one should normalize each centered column vector by dividing by the standard deviation.

Explain what is the difference between the Euclidean distance and the Mahalanobis distance when used for outlier detection for multivariate normally distributed data and state which one would you prefer.

Task 19

Consider the linear model:

$$\boldsymbol{y}=\boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\epsilon}$$

with $E(\boldsymbol{\epsilon})=0$ and $cov(\boldsymbol{\epsilon})=\sigma^2 I_n$ for a sample of n observations. Assume that the first column on $\boldsymbol{X}$ is a vector of ones (i.e., model contains an intercept). Assume we estimate the coefficients using OLS.

a.

Show that $\boldsymbol{X}^\top \boldsymbol{r} = 0$. What does this property mean in terms of the correlation between the covariates and the residuals?

b.

Show that $\sum_{i=1}^{n} r_i = 0$, where r_i is the i-th residual.

c.

Run the following code to obtain the vector `r` of residuals. The model being estimated is a linear regression with 2 independent variables X_1 and X_2 and an intercept.

```
data("mtcars", package = "datasets")
y <- mtcars$mpg
X1 <- mtcars$disp
X2 <- mtcars$hp
m <- lm(y ~ X1 + X2)
r <- residuals(m)
head(r)
```

```
##         1         2         3         4         5         6
## -2.148091 -2.148091 -2.348379  1.225844  3.235770 -3.199783
```

Write a function named `standardized_resids` "by hand" which takes as arguments the residuals and the design matrix and computes the standardized residuals (i.e., do not use any built-in functions in R). Apply the function to the vector `r` and the proper design matrix for this example. *Hint: Do not forget the intercept when building the design matrix.*

d.

Compute the standardized residuals $\tilde{r}_i$ using the following built in function applied to the regression object. Check whether the results are equal to your computation above.

e.

The studentized residuals are computed by:

$$\check{r}_i = \frac{y_i - \hat{y}_{(i)}}{\sqrt{\frac{\hat{\sigma}^2_{(i)}}{(1-h_i)}}},$$

where y_i is the omitted observation and $\hat{y}_{(i)}$ the prediction of y_i based on a model that was fitted after excluding the ith observation. The term $\hat{\sigma}_{(i)}$ denotes the standard deviation of the $n-1$ residuals obtained from the regression without i.

Using the fact that $y_i - \hat{y}_{(i)} = \frac{r_i}{1-h_i}$ (see e.g., here) one can show that

$$\check{r}_i = \tilde{r}_i \left(\frac{n-p-2}{n-p-1-\tilde{r}_i^2} \right)^{1/2}$$

where p is the number of independent variables (excluding the intercept).

Write a function named `studentized_resids` "by hand" which takes as arguments the original residuals and the design matrix and computes the studentized residuals (i.e., do not use any built-in functions in R). Apply the function to the vector `r` and the proper design matrix for this example. *Hint: Do not forget the intercept when building the design matrix. Note that p represents the number of independent variables without intercept.*

f.

Compute the studentized residuals $\tilde{r}_i$ using the following built in function applied to the regression object. Check whether the results are equal to your computation above.

Task 20

Which residual vs. fitted plot in Figure 1 indicates the best fit? Also inspect the qqplots in Figure 2 and comment on them. For each of the four data sets, explain which assumptions of the linear regression model seem to be violated (if any).

Task 21

Assume that different models are fit to a data set which contains information on the sales of a product in different countries (in hundred units) and on the advertising budgets which have been spent for promoting the product: `TV`, `youtube`, `social` media (all measured in dollars). Different regression models are fit to different subsamples of the data:

$$\text{Model 1:} \quad \widehat{sales} = 7 + 0.03 \cdot TV + 0.2 \cdot youtube + 0.5 \cdot social$$

$$\text{Model 2:} \quad \widehat{sales} = 6 + 0.2 \cdot youtube + 0.5 \cdot social + 0.25 \cdot social \cdot youtube$$

$$\text{Model 3:} \quad \widehat{sales} = 5.5 + 0.2 \cdot youtube + 0.3 \cdot youtube^2 + 0.1 \cdot social$$

$$\text{Model 4:} \quad \widehat{sales} = 200 + 0.05 \cdot (TV - \bar{TV}) + 0.2 \cdot (youtube - \bar{youtube}) + 0.5 \cdot social$$

Interpret the following quantities: *Hint: pay attention the ceteris paribus interpretation. Example interpretation: If youtube ad budget is increased by x we would expect an increase in sales by 2000 dollars, while keeping all other variables unchanged.*

a. In Model 1, the coefficient of TV.

b. In Model 1, the intercept.

c. In Model 2, what is the effect of youtube advertising on sales for a given value of social media advertising? What happens when the social media ad budget increases? How does it impact the effect of youtube budget on sales?

d. In Model 3, how does the relationship between *sales* and *youtube* look like? Which values of youtube advertising would be preferable for the sales? How would this relationship change if the coefficient of $youtube^2$ were negative? How would the interpretation change?

e. In Model 4, the intercept.

f. In Model 4, the coefficient of $(TV - \bar{TV})$.

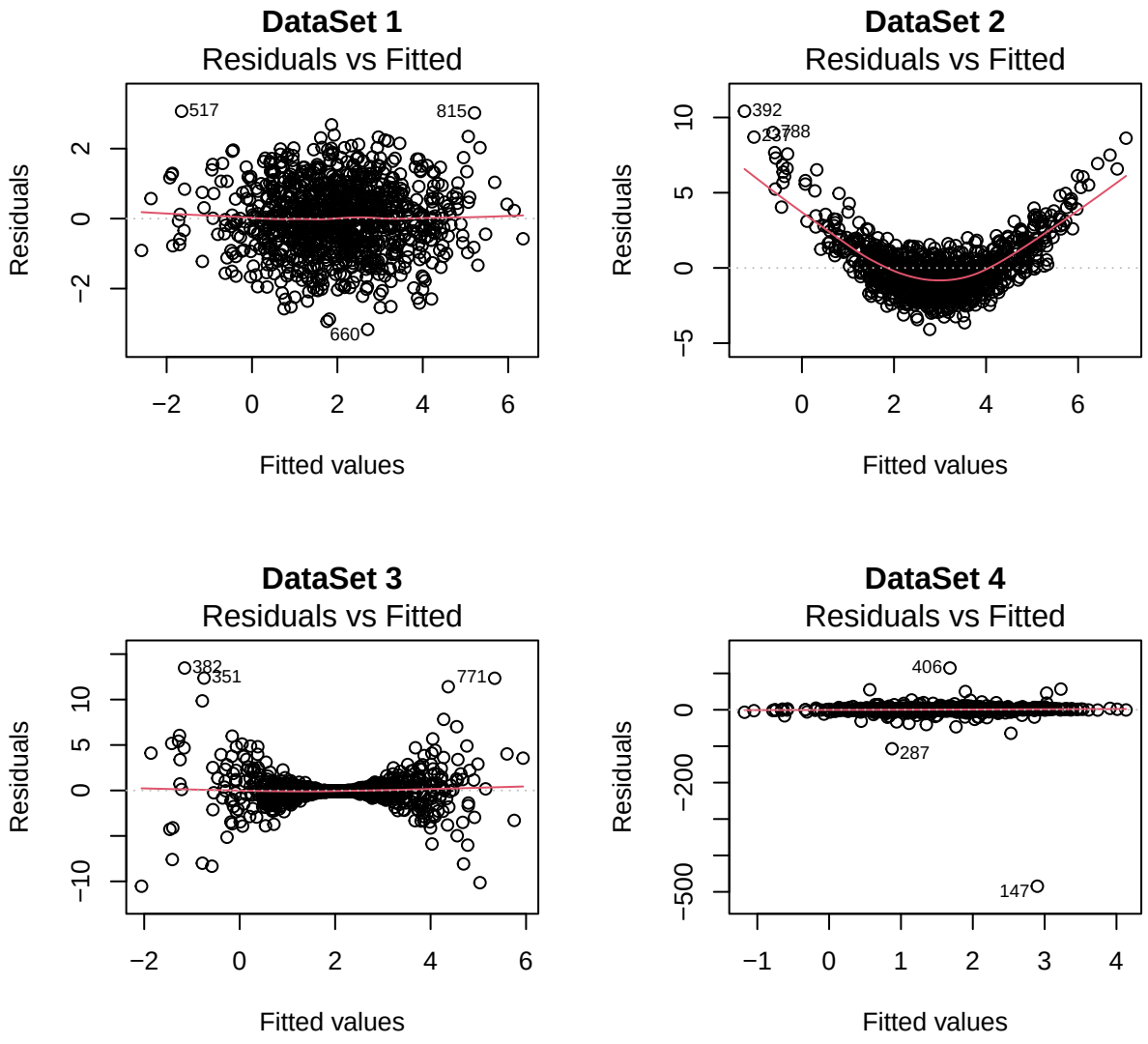

Figure 1: Residuals vs. fitted plot

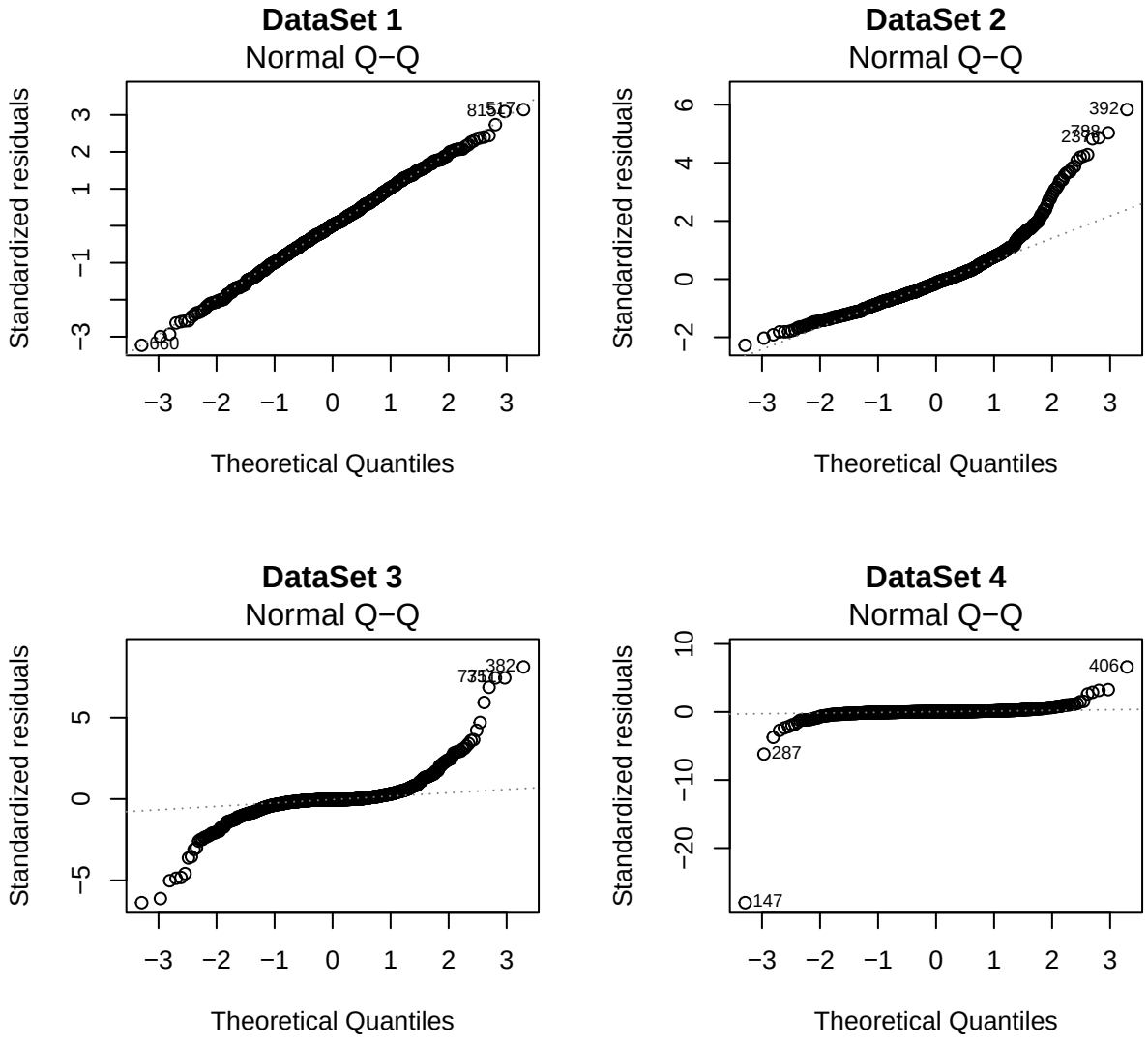

Figure 2: QQplots for the residuals

Task 22

The `ToothGrowth` data in R contains info on the length of odontoblasts (cells responsible for tooth growth) in 60 guinea pigs. Each animal received one of three dose levels of vitamin C (0.5, 1, and 2 mg/day) by one of two delivery methods, orange juice or ascorbic acid (a form of vitamin C and coded as VC).

```
data("ToothGrowth", package = "datasets")
```

a. Descriptives

Transform `dose` into an (unordered) factor variable with the levels `low`, `medium`, `high`.

Inspect the data using descriptives and graphs: i.e.,

- look at the summary,
- plot the boxplots of length for the 6 different combinations of dose and supp.
- make a table using pipes `|>` in order to build a table with the mean, standard deviation and number of observations for each level of `supp`.
- make a table using pipes `|>` in order to build a table with the mean, standard deviation and number of observations for each level of `dose`.
- make a table using pipes `|>` in order to build a table with the mean, standard deviation and number of observations for each of the 6 combinations for `supp` and `dose`.

b. Linear model

Run a linear regression model for `len` where you use `supp` and `dose` as independent variables. What is the interpretation of the coefficients for `Intercept`, `dosemedium` and `suppVC`?

c. Sum contrasts

Fit the same model as in b. but using the sum contrasts for both `dose` and `supp`. Interpret the coefficients for `Intercept`, `dose1` and `supp1`. *Hint: pay attention to whether this is a balanced or unbalanced design*

d. Interactions

Fit a model with an interaction between `dose` and `supp` (using treatment contrasts). From the coefficients compute the main effects (i.e., average length) for each of the 6 factor combinations. Compare this with the averages in a. *Hint: you can look at the model.matrix() for help.*

Task 23

Load the data set `child.iq.dta` into R (you can download it from TUWEL). Note that this is a Stata file. *Hint: You can use the Import data set button.*

The data set contains children's test scores at age 3, mother education and mother age at the time she gave birth for a sample for 400 children.

a.

Fit a regression of child test scores and mother's age. Display the data in a scatterplot together with the fitted regression line. Check assumptions of the model. Interpret the slope coefficient. If you were to recommend a birth age based on this model what would this be? *(Consider scores below 90 to be problematic).*

b.

Fit the regression also including education (as a numeric variable, not a factor). Interpret both slope coefficients. What is your recommendation for age now? Did it change?

c.

Fit the regression including mom age and education (as a factor variable). Plot the different regression lines for the different education levels in the scatterplot of test scores and mom age. Color points according to the different education levels. Also show the projection of the points on the corresponding regression line. (*See slides for an example*).

Task 24

Data were collected on the volume of users on the Northampton Rail Trail in Florence, Massachusetts for ninety days from April 5, 2005 to November 15, 2005. The data set is available in the **mosaicData** package in R:

Variables in the data set include:

- the number of crossings on a particular day (measured by a sensor near the intersection with Chestnut Street, `volume`),
- the average of the min and max temperature in degrees Fahrenheit for that day (`avgtemp`),
- dichotomous indicators for spring, summer and fall,
- dichotomous factor of whether the day was a weekday or a weekend/holiday (`dayType`),
- measure of cloud cover (`cloudcover` in oktas),
- measure of precipitation (`precip` in inches).

```
data("RailTrail", package = "mosaicData")
```

a. Processing the data

Remove duplicated observations if any, remove observations with missing values if there are any present in the data. Remove the columns which contain duplicated information. Make a factor variable `season` instead of the dummies for spring, summer and fall.

b. Descriptives

Inspect the data using descriptives and graphs: i.e., summary, pairwise scatterplots, parallel boxplots where relevant. Comment on the relation between volume and all the remaining variables.

c. Linear model

Try to estimate the model with `volume` as a dependent variable and all other variables as independent variables. What do you see? Are there any problems with the estimation? Comment on the results. If needed, adjust the variables in the sample and re-estimate the model. Briefly comment on the relationship between the variables and volume (focus on sign of coefficients rather than magnitude.)

d. Model comparison I

Test whether `cloudcover` can be dropped from the regression model given that `precipitation`, `hightemp`, and `lowtemp` are retained. Use the F statistic and level of significance 0.01. *Hint: you can use* `anova()`. State the hypotheses, the corresponding p-value, and conclusion in terms of the problem.

e. Model comparison II

Test whether in the model containing `precip`, `hightemp` and `lowtemp` variables `hightemp` and `lowtemp` should have equal coefficients or rather different coefficients. Use the F statistic and level of significance 0.01. State the hypotheses, the corresponding p-value, and conclusion in terms of the problem.

f. Model selection

Use the `step` function from the base **stats** package in R to perform backward model selection starting from a linear model containing all possible variables (i.e., `precip`, `hightemp`, `lowtemp`, `cloudcover`, `dayType`, `season`). Check the documentation of the `step()` function and make sure you understand how the function works. What is the final model selected by this procedure?

Task 25

a.

Write a function which for argument n constructs the $n \times n$ Hilbert matrix where the entry $(i, j) = 1/(i+j-1)$.

b.

For $n = 3$ and $n = 10$, using the **microbenchmark** library, compare how computing the inverse using the Cholesky decomposition compares to the inverse using `solve()`. Comment on the results.

Task 26

a.

Compute $Y = 1.5 \cdot X$ where X is the matrix.

```
X <- matrix(c(1, 2, 3, 1, 4, 9), ncol = 2)
```

Compute $Y^\top Y$ and $YY^\top$ by matrix multiplication and by the built-in functions. Benchmark the results for 100 repetitions. Discuss results.

b.

Consider the following matrix A and vector v:

```
A <- matrix(rep(1, 1000000), nrow = 1000)
v <- rep(1, 1000)
```

We are interested in computing A^2v.

In R implement the following matrix multiplications and compare their runtimes using benchmarking for 10 times. Why does the third calculation take less time than the other two? Discuss the results.

$$A^2v, \qquad (AA)v, \qquad A(Av)$$

c.

Assume $B = I_{1000}$ is the identity matrix. Similar to b., microbenchmark the operations Av, ABv, $(AB)v$ and $A(Bv)$. Discuss results.

Task 27

a.

For X being the Hilbert matrix for $n = 6$, calculate $H = X(X^\top X)^{-1}X^\top$ (try to do this in an efficient way).

b.

Compute the eigenvectors and eigenvalues of H in R (you can use the built-in function). Look at the object obtained and check its' structure using `str()`. Extract the vector of eigenvalues.

c.

Compute the trace of H and compare to the sum of the eigenvalues.

d.

Compute the determinant of H and compare to product of eigenvalues.

e.

Compute the inverse of X and its eigenvalues and eigenvectors. Is there any relationship between the eigenvalues of the inverse and of the original matrix?

Task 28

a.

Try to compute in R the Cholesky of the Hilbert matrix for $n = 20$. Comment on the results and explain what you think is happening in the background.

b.

The function `kappa()` can be used to find the condition number of a given matrix (the ratio of largest to smallest non-zero singular values). This gives an idea of how bad certain matrix calculations will be when applied to the matrix. Large values indicate poor numerical properties. Calculate the condition number for the 3×3, 5×5 and 7×7, 20×20 Hilbert matrices. Interpret the results.

Task 29

Assume y is a numeric n-vector (the response) and X the $n \times p$ ($n > p$) model matrix having full column rank. Let β be the coefficient vector of interest having length p.

The ordinary least squares problem is then formulated as

$$\hat{\beta} = \arg\min_{\beta} ||y - X\beta||^2.$$

The textbook solution is then usually the closed form expression

$$\hat{\beta} = (X^\top X)^{-1} X^\top y.$$

The question is how to compute that in R?

a.

Write a function `LmNaive` which takes as input `y` and `X` and returns $\hat{\beta}$. The function should be a naive implementation of the above equation.

b.

Using `solve` with two arguments is more efficient than with one argument and similar `crossprod` and `tcrossprod` have advantages over the standard matrix multiplication function `%*%`.

Implement a function using these functions and name it `LmCP`.

c.

Another way to implement it is using a Cholesky decomposition of $X^\top X$. Implement a function called `LmChol`. *Hint: check how the Cholesky can be used for solving linear systems of equations. Use the two step approach here.*

d.

Implement a function called `LmSvd` which uses singular value decomposition of X in computing the OLS solution.

e.

Another way to compute $\hat{\beta}$ is by solving the system $X\hat{\beta} = y$ using a QR decomposition of the non-quadratic matrix X. There is a built-in function `qr.solve(X, y)` for this purpose. However, here you should implement a function called `LmQR` by hand. *Hint: check how the QR can be used for solving linear systems of equations. Use the two step approach here.*

f.

Familiarize yourself with the function `lm.fit()` and describe in words what it does.

g.

Consider this small data set:

```
set.seed(1)
n <- 50
x <- rt(n, 2)
eps <- rnorm(n,0, 0.1)
y <- 3 - 2 * x + eps
X <- cbind(1, x)
colnames(X) <- c("Intercept","x")
```

Check for each of the functions in subtasks a-f the OLS solutions. Then use the `microbenchmark` package for timing comparisons (repeat 10 times) of all versions above and for the `qr.solve(X, y)` and `lm.fit(X, y)` functions. Discuss the results.

Task 30

A matrix is called *sparse* if most of its entries are zero and *dense* otherwise. Typically, a *sparse* matrix has a degree of sparsity (i.e., percentage of zero entries) of at least 90%. Building special functionality for sparse matrices can lead to memory savings and improved computation time.

a.

Assume matrix X of dimension $(n \times n)$ is diagonal:

$$X_{i,j} = \begin{cases} x_i, & i = j \\ 0, & i \neq j \end{cases}$$

and β is an $n \times 1$ vector.

How many flops (one flop is a unit of computation which could denote one addition, subtraction, multiplication or division of floating point numbers) are required to compute $y = X\beta$ if we do not take the sparsity of X into account?

How can we calculate y by taking the sparsity structure of X into account? How many flops would be required for this operation?

b.

One way to represent sparse matrices is through a coordinate list with the following elements:

- i - row index of the non-zero entries
- j - column index of the non-zero entries
- x - value of the non-zero entries

In R write a function `make_sparse_cl` which takes a (sparse) matrix `M` as an argument and converts in to a data frame with 3 columns, where the columns contain the 3 elements i, j and x. *Hint: you can use which(..., arr.ind = TRUE).* Call the function on the following matrix:

```
set.seed(1234)
n <- 30
M <- matrix(0, nrow = n, ncol = n)
M[sample(1:(n ^ 2), floor(n^2 * 0.1))] <- rnorm(floor(n^2 * 0.1))
```

c.

Using the **Matrix** package, one can cast a matrix `M` into a sparse format by `as(M, "sparseMatrix")`. Using `str(as(M, "sparseMatrix"))` inspect the resulting object. Explain the "slots" or elements of the resulting object.

d.

Write a function `sparse_add` for sparse matrix addition which takes two data frames as in b. as arguments. The function should return another data frame with columns `i`, `j`, `x` containing information on the non-zero entries of the sum matrix. *Hint: you can make use of function merge(..., all = TRUE).*

e.

Many simple operations such as standardization (more specifically centering) can destroy the sparsity of a matrix. This can be problematic, for example, in a regression context where the X matrix is sparse.

One example where standardizing is important is in estimating a linear regression with ℓ_1-penalty (also known as lasso regression):

$$\hat{\beta}^{\ell_1}(\lambda) \in \arg\min_b \left\{ ||y - Xb||_2^2 + \lambda ||b||_1 \right\}.$$

Here, the starting value for λ can be found by

$$\lambda_1 = \max_j |y^\top Z_j|, \qquad Z_j = \frac{X_j - \hat{\mu}_j}{\hat{\sigma}_j}$$

where j is an index running through the columns of X, $\hat{\mu}_j$ is the mean and $\hat{\sigma}_j$ is the standard deviation of column X_j.

Write a function which takes X and y as arguments and returns λ_1. Make sure that the function does not destroy the sparsity in X. Call the function for the following data:

```
set.seed(1)
X <- matrix(as.numeric(runif(2000) > 0.9), ncol = 20)
y  <- X[,1] * 0.1 + rnorm(100, sd = 0.1)
```

Task 31

Linear regression models are often used for prediction purposes i.e., to predict new observations y^{new} based on a collection of x^{new} values. Ideally, when comparing different models, we should focus on how well the model predicts on unseen data, not only by in-sample measures of fit such as R^2, which measure how well the model fits the given data. For this purpose, a common strategy is to split the data at hand into a train and a test sample. The train sample will be used for estimating the regression coefficients, while the test data will be used to check how well the model is predicting the response variable.

Consider the following data where the goal is to explore the relation of heat capacity for a type of acid and temperature.

```
data("heatcap", package = "GLMsData")
```

For this data often a polynomial regression is employed:

$$\texttt{Cp} = \beta_0 + \beta_1 \texttt{Temp} + \beta_2 \texttt{Temp}^2 + \beta_3 \texttt{Temp}^3 ... + \epsilon = \beta_0 + \sum_{k=1}^{K} \beta_k \texttt{Temp}^k + \epsilon$$

The model selection exercise here is concerned with choosing the degree of the polynomial K to include in the model.

a.

Split the `heatcap` data set into a train set containing 15 observations (roughly 80% of the sample) and a test set containing 3 observations. The observations should be randomly allocated to either train or test set.

b.

For the train set estimate the polynomial regression for $K = 5$. Specify the formula in `lm()` "by hand": `Cp ~ Temp + I(Temp ^ 2) + ...`. Check the summary.

c.

If K is large it is inconvenient to type the formula by hand. The `poly()` function in R can be used instead.

```
formula <- Cp ~ poly(Temp, 5)
```

Estimate the linear regression. Compare the coefficients and the goodness of fit of this model and the one in b. *Hint: Note that by default* ***`poly`*** *will create orthogonal polynomials.*

d.

For the linear model in c, use the `predict(lm.object, ...)` function to predict the heat capacity `Cp` on the test sample based on the fitted regression model on the train data. Compute the root mean squared error between the predicted and the observed `Cp` on the test sample:

$$RMSE = \sqrt{\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\hat{y}_i - y_i^{\text{observed}})^2}$$

e.

Repeat the prediction exercise in d. for $K = 1, \ldots, 7$. Choose the degree K which delivers the smallest RMSE.

Task 32

Consider the following data, which contains the delay (in minutes) in the departure of a random sample flights for two different airlines "AA" and "AB" operating at Vienna International Airport. Note that negative numbers represent departures before the scheduled time.

```
yAA <- c(2.5, 2.9, 4.5, 5, 5.5, 6.5, 7.5, 8.48, 8.49, 8.5, 8.5, 8.5,
         8.9, 9, 9.1, 9.2, 11.5, 28.5)
yAB <- c(-10, -9.9, -8.1, -8, -7.5, -0.9, -0.5, -0.4, 0, 0, 1, 8.5, 8.6,
         8.6, 8.7, 8.8, 9, 9.5, 10, 10.1, 10.5, 15, 32)
```

a.

Look at the distribution of delays for the two airlines. Generate boxplots, histograms and density plots for the two airlines (superimposed, not in separate plots). Add appropriate coloring and legends where needed. Comment on what you see in terms of differences between the distributions (location, dispersion, skewness).

b.

Perform a two sample t-test to check whether the difference in delays among the two airlines is significantly significant (assume a 10% α-level). Comment on the R output and interpret the results.

c.

Look at normal qqplots for the two groups. Do you see deviations from normality? What would this imply for the t-test performed?

d.

If you think the sample t-test is the most appropriate test for this data, clearly state the arguments on why this is the case. Otherwise, perform in R another test for the difference in location between two distributions, which you consider more appropriate than the two sample t-test for this data.

Task 33

In this exercise you should examine how the level accuracy of the one sample t test depends on the underlying distribution of the X variable. Let $X_1, \ldots, X_n$ be an i.i.d. sample with mean μ and consider the hypothesis $H_0 : \mu = 0$. The standard test is to reject when $|T| > c_0$ where $T = \sqrt{n}\bar{x}/s$ is the test statistic, $\bar{x}$ and s are the sample mean and standard deviation respectively, c_0 is the $1 - \alpha/2$ quantile of the t-distribution with $n - 1$ degrees of freedom for a nominal significance level α.

Remember to set a seed at the beginning of the code.

a.

Run 10000 repetitions to estimate the empirical Type I error rate (effective significance level) for standard normal data when $n = 15$. Assume a nominal α of 5%. Compute also the standard error of the empirical Type I error rate.

Hint: After simulating the mean zero data which is in accordance to the null hypothesis, run the one sample t test with a significance level of α. Record whether you reject the null or not. The empirical Type I error

rate is percentage of rejections over the repetitions. For computing the standard error, use the fact that this estimator is a proportion.

b.

Repeat a. with

- exponential data (with rate 1) $X \sim \text{Exp}(1)$
- a gamma distributed variable that is the sum of three exponentials with rate 1 $X = Y_1 + Y_2 + Y_3, Y_j \sim \text{Exp}(1), j = 1, 2, 3.$
- a Cauchy distribution with location 0 and scale 1 $X \sim \text{Cauchy}(0, 1)$ (note that for the Cauchy the second and first moments do not exist so the central limit theorem does not hold)

Make sure you translate the data so that expectation of X is equal to zero in each case.

c.

Repeat a. and b. when $n = 50$.

d.

Summarize your findings and discuss the results.

Task 34

Especially when dealing with time series data, it is quite common that error terms are serially correlated (also called auto-correlated). You should examine the effect of having auto-correlated errors in a simple regression framework which assumes iid errors and ignores the serial correlations.

$$y_t = \beta_0 + \beta_1 x_t + \epsilon_t, \qquad \epsilon_t \overset{iid}{\sim} N(0, \sigma^2)$$

Using simulation, you will investigate the behavior of the standard least-squares estimate of β_1. (Note that one can also analyze the impact of the serial correlation analytically, but this is a computational class :-))

a.

Write a function `sim_y_ar_errors` which takes as arguments n, x (the independent random variable) and parameters β_1, β_2, a, σ_η^2, and returns a response vector y simulated from the regression model above, where the errors are serially correlated rather then iid normal.

For creating auto-correlated errors use the following autoregressive AR(1) model $\epsilon_t = a\epsilon_{t-1} + \eta_t$, where $\eta \overset{iid}{\sim} N(0, \sigma_\eta^2)$. *Hint: you can use the* `arima.sim()` *function in R to generate this type of errors.*

b.

Set up a simulation study where for $M = 1000$ replications you generate a response y by drawing a different set of errors in each replication (remember to set a seed at the beginning of the code).

- Use $n = 20$
- As an independent variable assume $X \sim \text{Unif}(-1, 1)$ – you can use the same X in all repetitions, as we consider x to be deterministic in the model.
- Assume $\beta_0 = 0$, $\beta_1 = 2$, $\sigma_\eta^2 = 1$ throughout the analysis.
- For a you should use the scenarios: $a = \{-0.9, -0.1, 0, 0.1, 0.9\}$. Note that $a = 0$ corresponds to the no violation case.

c.

For each scenario ($a = \{-0.9, -0.1, 0, 0.1, 0.9\}$), estimate the bias $\hat{\beta}_1 - E(\hat{\beta}_1)$ and the variability $Var(\hat{\beta}_1)$ for the standard OLS estimator based on the M repetitions. Discuss your results. Compare with the bias and variability you nominally have from the elementary least-squares theory that does not take serial correlations into account (see e.g., Chapter 6, slide 14).

Hint: you can use `lm` *to get* $\hat{\beta}_1$ *for each replication or the closed form formula.*

Task 35

Many statistical learning applications require the prediction of a response variable that takes on categorical values, rather than numerical values. The binary case, where the response variable can take one of two values or factor levels, is the most commonly encountered:

$$y_i = \begin{cases} 1, & \text{category}_i = A \\ 0, & \text{category}_i = B \end{cases}$$

a.

In principle there is no "algorithmic" reason why we cannot estimate a linear regression for the type of problem above (such a model is sometimes referred to as the linear probability model).

$$P(y = 1) = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p + \epsilon$$

However, many of the regression assumptions would be violated.

State and explain at least two problems with using the linear regression for a response which only takes zero/one values.

b.

A more common model for binary response is the logistic regression model, where the probability of observing a one is modeled as:

$$p_i = P(y_i = 1) = \frac{\exp(x_i^\top \beta)}{1 + \exp(x_i^\top \beta)}.$$

Note that the model above is a member of the class of generalized linear models (GLMs). When we observe a sample on n observations, likelihood function of such a model is given by

$$\mathcal{L}(\beta; y) = \prod_{i=1}^{n} p_i^{y_i} (1 - p_i)^{1-y_i}$$

Derive the log likelihood as well as the gradient and the Hessian matrix of the log likelihood for the logistic regression model.

c.

Implement a function based on the Newton-Raphson algorithm which finds the vector of coefficients β which minimizes the negative log-likelihood function of the logistic regression model. For a convergence criterion you can use Euclidean distance between $\hat{\beta}^{(i)}$ and $\hat{\beta}^{(i+1)}$

$$\sqrt{\left(\hat{\beta}^{(i+1)} - \hat{\beta}^{(i)}\right)^{\top} \left(\hat{\beta}^{(i+1)} - \hat{\beta}^{(i)}\right)} < \text{tol}.$$

Allow a maximum of `maxit` iterations in the function. If convergence is not achieved before `maxit` iterations, you should print a warning using `warning("Maximum iterations reached without convergence!")`

Important: Avoid inverting the Hessian matrix (for a hint see slide 5 of Chapter 9).

```
logistic_nr <- function(y, X, maxit = 500, tol = 1e-5) {

}
```

d.

Run the linear probability model and the logistic regression model (using `logistic_nr`) for the following `X` and `y`.

```
set.seed(1234)
n <- 1000
p <- 2
beta <- c(0.2, 2, 1)
X <- cbind(1, matrix(rnorm(n * p), ncol = p))
mu <- 1 / (1 + exp(-X %*% beta))
y <- as.numeric(runif(n) > mu)
```

Compare the coefficients obtained by your `logistic_nr` function with the ones of the linear probability model and with the coefficients obtained when running the built-in function in R for logistic regression:

```
glm(formula, family = binomial())
```

Task 36

How many zeros does the function $f(x) = sin(10x) - x$ have? Find all these using a root solver of your choice. *Hint: inspecting the graph of the function will be helpful.*

Task 37

Rosenbrock's banana functionis a commonly used function to test optimization algorithms, given by

$$f(x, y) = (a - x)^2 + b(y - x^2)^2$$

It has a single global minimum at (a, a^2) where $f(x, y) = 0$.

a.

Use a contour plot to inspect the function for $a = 1$, $b = 100$. Why is this function difficult to optimize? *Hint: you can look at* `filled.contour()`.

b.

Try to find the minimum of f with $a = 1$, $b = 100$ using a starting value of (0, 10). Try different optimizers in R in case the first one fails.

Task 38

Using built-in functions in R (and its extension packages), find non-negative x_1, x_2, x_3 and x_4 to minimize

$$C(x) = x_1 + 3x_2 + 4x_3 + x_4$$

subject to the constraints

$$x_1 - 2x_2 \geq 9, \quad 3x_2 + x_3 \geq 9, \quad x_2 + x_4 \geq 10.$$

Task 39

Consider a portfolio management problem where the portofolio weights should be allocated to a collection of three stocks such that the following function is minimized:

$$Q(x) = \frac{k}{2} x^\top V x - m^\top x$$

where k is a risk aversion parameter which is fixed in advance. Assume that the mean and covariance of the monthly returns for the three stocks are

$$m = (0.0427, 0.0015, 0.0285)^\top, \qquad V = \begin{pmatrix} 0.1000^2 & 0.0018 & 0.0011 \\ 0.0018 & 0.1044^2 & 0.0026 \\ 0.0011 & 0.0026 & 0.1411^2 \end{pmatrix}$$

a.

Find the optimal weights x under the assumption that short-selling is **not allowed** for k = 4 and k = 1, respectively. How does being less risk-averse affect the investor's behavior?

b.

Find the optimal weights x under the assumption that short-selling is **allowed** for k = 4 and k = 1, respectively. How does being less risk-averse affect the investor's behavior?