R Oefeningen

Inleiding R Cursus

Longhow Lam

longhowlam@gmail.com



[Session 1. Data types and structures in R 2](#__RefHeading___Toc719_1123257060)

[1. Data types 2](#__RefHeading___Toc721_1123257060)

[2. Logical data type. 2](#__RefHeading___Toc723_1123257060)

[3. Character data type and factor data type. 2](#__RefHeading___Toc725_1123257060)

[4. Creating sequences. 2](#__RefHeading___Toc727_1123257060)

[5. Creating factors from double data. 2](#__RefHeading___Toc729_1123257060)

[6. Creating a data.frame 2](#__RefHeading___Toc731_1123257060)

[7. Creating a list 3](#__RefHeading___Toc733_1123257060)

[8. Importing some data 3](#__RefHeading___Toc735_1123257060)

[Session 2. Data preparation 4](#__RefHeading___Toc737_1123257060)

[1. String manipulation 4](#__RefHeading___Toc739_1123257060)

[2. Dates and times. 4](#__RefHeading___Toc741_1123257060)

[3. Basic data manipulation 1 4](#__RefHeading___Toc743_1123257060)

[4. Basic data manipulation 2 4](#__RefHeading___Toc745_1123257060)

[5. Car data van gaspedaal 4](#__RefHeading___Toc747_1123257060)

[6. Split data and stack data 5](#__RefHeading___Toc749_1123257060)

[7. Sampling van data 5](#__RefHeading___Toc751_1123257060)

[Session 3. Data visualization 6](#__RefHeading___Toc753_1123257060)

[1. plotting using ggplot2 6](#__RefHeading___Toc755_1123257060)

[2. Smoothing plots 6](#__RefHeading___Toc757_1123257060)

[3. Plotly graphs 6](#__RefHeading___Toc759_1123257060)

[4. Leaflet graphs 7](#__RefHeading___Toc761_1123257060)

[5. Leaflet graphs polygons 7](#__RefHeading___Toc763_1123257060)

[Session 4. Machine learning 9](#__RefHeading___Toc765_1123257060)

[1. Predicting car prices from ‘gaspedaal.nl’ with linear regression 9](#__RefHeading___Toc767_1123257060)

[2. Predicting car prices with Random forests 9](#__RefHeading___Toc769_1123257060)

[3. Classification with house prices 10](#__RefHeading___Toc771_1123257060)

[4. Market basket analysis 10](#__RefHeading___Toc773_1123257060)

# Session 1. Data types and structures in R

## 1. Data types

1. Create the object x1 with the numbers 1, 3.51, 0.7 and 5.
2. Show the data class of x1.
3. Transform x1 into an integer data type and store the result in x2.
4. Use the round function to round x1 and store the result in x3.
5. Is x3 of data type integer?

## 2. Logical data type.

1. Create the vector y1 with the numbers 1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.
2. Create a logical vector y2 from y1. An element of y2 should be TRUE if the corresponding element of y1 is larger than 5.
3. How many elements are smaller than 5?

## 3. Character data type and factor data type.

1. Create the character vector y1, with the characters “high”, “low”, “high”, “high”, “low”.
2. Create the factor variable y2 from y1. Print the levels of y2.
3. Transform y2 to a numeric vector y3, so that high becomes 2 and low becomes 1.

## 4. Creating sequences.

1. Create a vector x, with elements (1,2,3,4,..,100).
2. Create a vector y, with elements (0,5,10,...,500).
3. Create a vector z1, with elements (1,1,1,2,2,2,...,50,50,50).
4. Create a vector z2, with elements (1,2,2,3,3,3,.....,10,....10).
5. Create a vector z3, with elements (1,2,2,3,4,4,5,6,6,7,8,8,...,50,50).

## 5. Creating factors from double data.

1. Generate a vector x with 100 random normal numbers.
2. Use x to generate a factor vector y with three levels: “low”, “average” and “high”. If an element in x is:

* between -Inf and -1.5 then the level in y should be “low”,
* between -1.5 and 1.5 then the level in y should be “average”,
* between 1.5 and Inf the level in y should be “high”.

Use the function cut.

## 6. Creating a data.frame

1. Create a data frame df1 from the two vectors x and y from the previous exercise
2. Rename the columns x and y into index and income
3. Create an additional factor column Gender, randomly filled with “M” or “F”

## 7. Creating a list

1. Create a list L1 with 4 elements

* X1 = 100 random normal numbers,
* X2 = 100 random uniform numbers between 0 and 1
* X3 = the numbers one to 20
* x4 = the numbers 50 to 1

B. Give an overview of the components of L1

C. Calculate the mean of the four vectors in list L1 using the lapply function

## 8. Importing some data

In the data directory there are some dummy text files: f1.csv, f2.txt and f3.txt

Import these sets in the ‘correct’ way. The resulting datasets in R should all have 3 columns.

# Session 2. Data preparation

## 1. String manipulation

Copy the following vector to R.

X = c("abd 06-456", "blab la (06)-123.45678", "hoi 06-12123478", "mijn nr 0689452312")

1. For **this exercise** a valid mobile phone number starts with 06 followed by hyphen or space or not and then 8 consecutive digits. What are the valid numbers in x, use a regular expression?
2. Extract the phone numbers from x into a new vector y

## 2. Dates and times.

1. Create a date object from the characters "31-jan-05" "22-Sep-07".
2. Create a date object from the characters "31-jan-05 3:03:20" "22-Sep-07 22:29:56".
3. Add 1 months the dates from exercise (B), what happens to 31-Jan?
4. Add 4 weeks to the dates from exercise (B).
5. Create a date sequence vector x with the dates: 31-1-2016, 29-2-2016, 31-03-2016,……,31-12-2016

For the following two exercises we are going to use restaurant data in the file **Restaurants.csv** in the data folder. Use the read\_csv function to import this data set.

## 3. Basic data manipulation 1

1. Create a subset of the restaurants whose name is “kota radja”, how many are there?
2. Give a top ten restaurants in Amsterdam based on price.
3. Create an extra column that has the values zero or one. 1 if Kitchen type is Chinees and 0 otherwise.
4. Create a sub set of the data, filter only the restaurants that have price smaller than 25.
5. Give the number per kitchen types of that sub set.

## 4. Basic data manipulation 2

1. In the data folder there is a Postcode\_NL.RDs file, import that set.
2. Use it to enrich the restaurant data set with an extra column provincie.
3. Count the number of restaurants per province of the whole set.
4. Give the max and min prices per province.
5. In the column “keuken” there are some mistakes. A price where there should have been a kitchen type. Remove all kitchen types that have a digit in.

## 5. Car data van gaspedaal

In the data folder there is a file R data set AllCarsGasPedaal.Rds import this data set using readRDS. It contains car data scraped from the website gaspedaal.nl, each row represents a car that was for sale.

1. List the names of the columns of the data set.
2. In this data set there is a column bjkm which is character, containing the kilometerstand, extract it and convert it to numeric so that we can calculate with it.
3. Select only the brands Audi, Renault and BMW put those records in a new data frame.

## 6. Split data and stack data

Consider the following simple data set with 3 rows:

Df = data.frame(

V1 = c(“a,b,r”, “p,q”, “p,q,w,z”)

)

I need the following set:

Id Item

1 a

1 b

1 r

2 p

2 q

3 p

3 q

3 w

3 z

How to this? use tidyr functions separate and gather.

## 7. Sampling van data

Kijk naar sample\_n of sample\_frac uit dplyr.

A. Create a df1 data frame with 100 columns and 1000 rows. Filled with random numbers between 0 and 1.

B. Create df2 which samples 500 rows from df1

C. Create df3 which samples 50% of the rows from df1

# Session 3. Data visualization

Use the set AllCarsGasPedaal.Rds from the previous exercise

## 1. plotting using ggplot2

1. Create a scatterplot of the VraagPrijs and KMStand (The amount of KM driven).
2. Remove outlying points and create plot again using alpha = 0.01.
3. Filter the set so that we only have the top eight brands (The column ‘Merk’) (based on # observations).
4. Create a new scatter plot, now the points should be colored by brand, using the subset of the previous exercise.
5. Create a facets by brand, each facet should have a histogram of the price. If we have 8 brands then create a grid of plots with two rows and 4 columns.
6. Create facets by brand and type, each facet should contain a scatter plot between price and KM stand.

## 2. Smoothing plots

Smoothing lines are useful to get an idea of the relation between X and Y in a scatterplot (with many points). They can be used for continuous X and Y data, but also for continuous X and binary Y. Use geom\_smooth

Let’s use the AllCarsGasPedaal again for continuous X and Y

1. Create a scatter plot of VraagPrijs and KMStand again but now add a smoothed line through the points to get an idea of the relation between KM and Price.
2. Also create a smoothing line per brand, only the top 8 brands

Let’s use the titanic\_train data set in the titanic package for continuous X and binary Y

1. Create a scatter plot of Age and Survived. Not much interesting to see right?
2. Now add a smoothing line. Interesting relation….
3. Create facets for the variable Sex
4. Create facets for the variables Sex and Pclass

## 3. Plotly graphs

With plotly you can create interactive graphs. Let’s use the AllCarsGasPedaal data again.

* 1. Creating interactive scatter plots with 435K data points makes the plot very slow, first sample randomly 10% from the data and then plot a scatter with plotly KMStand against VraagPrijs.
  2. Now create the same plot but when you hover over the points it should display name of the car, in the variable ‘auto’.
  3. Create also a plot where the points are colored according to age. You need to calculate age first.
  4. There are some outlying ages and prices, remove them so that the plot becomes better.
  5. Create per Brand (Merk) a boxplot of the Price, because there are too many brands, focus on the top 25 brands (based on counts in the data set. Also remove outlying prices (price > 250K).

## 4. Leaflet graphs

Leaflet graphs are nice to visualize geographic data on maps. We’ll use the Restaurants.csv data set in the data directory. The longitude and latitude coordinates are in the columns LATs and LONGs

* 1. Import the data and select the restaurants in Utrecht.
  2. Plot all the Utrecht restaurant on an interactive leaflet.
  3. Leave out the restaurant with ‘wrong’ coordinates
  4. Now create a leaflet where a popup text will appear with the name of the restaurant

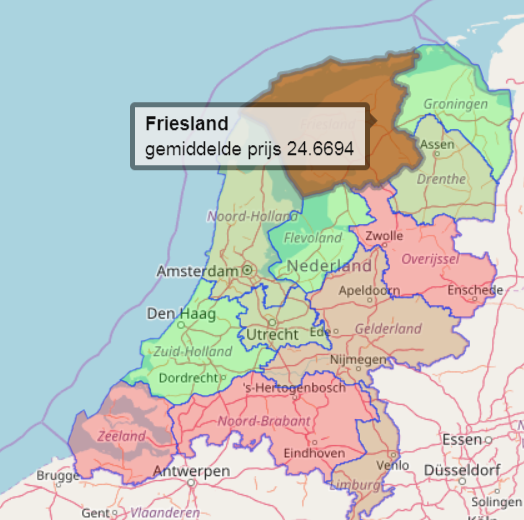
## 5. Leaflet graphs polygons

Instead of putting markers or circles on a leaflet, polygons can also be placed on an interactive map. We use the provinces of the Netherlands. Data is already present in the shape file **TopGrenzen-prov-actueel.shp**

* 1. Import the TopGrenzen shape file as in the lecture.
  2. Plot the polygons of the Dutch provinces on a leaflet with the function **addPolygons**.

The polygons can have different colors, for each province we can calculate the average prijs for a menu in the restaurant, and color the polygon accordingly. First calculate the average prijs per province.

* 1. To do that we need to tell in which province a restaurant is, import the data set **postcodes\_NL.RDs** in the data directory.
  2. Join the restaurant data with this postcode data set on postocde so that we know for each restaurant in which province they are.
  3. Now group by province and calculate per province the average prijs (be aware of NA prices!!)
  4. Join the prov polygons data with the average prices.
  5. Define a four color scale for the price with the function.
  6. Now create a leaflet and use addPolygons to draw per province the price.
  7. In addPolygons you can specify highlight options, experiment with that.



# Session 4. Machine learning

## 1. Predicting car prices from ‘gaspedaal.nl’ with linear regression

We are going to use the data AllCarsGasPedaal.Rds again, but now for modeling. We use regression models to predict the price of a car.

1. Start with a simple linear regression (the function lm) , predict VraagPrijs (the target) with KMStand (the input variable).
2. What is the R2of the model.
3. You may find some very large outliers in KM stand and VraagPrijs, plot those variables in a scatter plot to see them. Remove outlying points.
4. Predit the VraagPrijs with KMStand again, what is the R2 now?

When we plot Prijs against KMStand we see that it is a non linear relation, how can we get that into the linear model? We could use buckets (piece-wise linear constants) or we can transform with splines.

1. Create an additional column “KMBuckets” in the data set, that KMStand buckets. Use the function cut.
2. Now use KMBuckets instead of KMStand, did the R2 improve?
3. Attach the library **splines**, and use ns(KMStand, 7) as a predictor in the function lm. Instead of fitting a straight line it will fit a smoothed line.
4. Add additional variables, like age of car and Merk, brandstof and Transmissie, what is the R2 now? Be careful, with Merk there are too many categories, look at how many there are. Use forcats fct\_lump to reduce this number.

## 2. Predicting car prices with Random forests

As an alternative to linear regression for car prices we can use random forests with the **h2o** package.

1. Start simple predict Vraagprijs with KMStand.
2. What is the R2 ? You need to calculate this your self, since h2o does not provide this. Calculate the prediction on the train data so that you have observed house prices (the target) and the predictions. Then you can calculate the R2 as the ratio of squared differences, between the model predictions and the mean.

