IODS course project

Subam Kathet

Contents

Introduction to Open Data Science - Course Project

IODS Course Project

SUBAM KATHET

This course was recommended by a friend who is in the masters program in data science at the University of Helsinki. Data management, mining, data analytic and machine learning among many other within the same sphere are the next generation skill set everyone is recommended to acquire and here I am. I am a bit nervous, very exited and mostly curious to take a deep dive into the world of data science.

About the project

Here is the link to my github webpage.

https://iamsubam.github.io/IODS-project/

And here is the link to my course diary.

https://github.com/iamsubam/IODS-project

Week 1: Start me up !! The book and material

I have only had some time to browse through the R for Health Data Science. Coming from a background of experimental epidemiology, I was drawn immediately by linear and logistic regression because this is something I often rely on in my work. I looked into survival analysis briefly because of my interest and found it quite interesting. Although I need to practice a lot before I can get my hands around the analysis. I think the book gives a great over view of essential statistical analysis required on a fundamental level. Some knowledge of statistics can be of great advantage as R platform is already designed with a steep learning curve.

```
# This is a so-called "R chunk" where you can write R code.

date()
```

```
## [1] "Tue Nov 15 21:30:07 2022"
```

```
# Trying to check if the chunk works or not. It is usually a struggle especially when R version is outd
```

Week 2: Regression and model validation

This set consists of a few numbered exercises. Go to each exercise in turn and do as follows:

1. Read the brief description of the exercise.
2. Run the (possible) pre-exercise-code chunk.
3. Follow the instructions to fix the R code!

2.0 INSTALL THE REQUIRED PACKAGES FIRST!

One or more extra packages (in addition to `tidyverse`) will be needed below.

```
# Select (with mouse or arrow keys) the install.packages("...") and
# run it (by Ctrl+Enter / Cmd+Enter):

# install.packages("GGally")
```

2.1 Reading data from the web

The first step of data analysis with R is reading data into R. This is done with a function. Which function and function arguments to use to do this, depends on the original format of the data.

Conveniently in R, the same functions for reading data can usually be used whether the data is saved locally on your computer or somewhere else behind a web URL.

After the correct function has been identified and data read into R, the data will usually be in R `data.frame` format. Te dimensions of a data frame are (n,d), where n is the number of rows (the observations) and d the number of columns (the variables).

The purpose of this course is to expose you to some basic and more advanced tasks of programming and data analysis with R.

Instructions

- Read the `lrn14` data frame to memory with `read.table()`. There is information related to the data here
- Use `dim()` on the data frame to look at the dimensions of the data. How many rows and colums does the data have?
- Look at the structure of the data with `str()`.

Hint: - For both functions you can pass a data frame as the first (unnamed) argument.

R code

```
# This is a code chunk in RStudio editor.
# Work with the exercise in this chunk, step-by-step. Fix the R code!

# read the data into memory
lrn14 <- read.table("http://www.helsinki.fi/~kvehkala/JYTmooc/JYTOPKYS3-data.txt", sep="\t", header=TRU

# Look at the dimensions of the data


# Look at the structure of the data
#use .txt file to import data set for better description.
# Preliminary results available at http://www.slideshare.net/kimmovehkalahti/the-relationship-between-l
#Total respondents n=183, total question n=60, so 184 rows including heading and 60 columns
#The code as respective column heading represents a question related to the survey and number. Each SN
```

2.2 Scaling variables

The next step is wrangling the data into a format that is easy to analyze. We will wrangle our data for the next few exercises.

A neat thing about R is that may operations are *vectorized*. It means that a single operation can affect all elements of a vector. This is often convenient.

The column `Attitude` in `lrn14` is a sum of 10 questions related to students attitude towards statistics, each measured on the Likert scale (1-5). Here we'll scale the combination variable back to the 1-5 scale.

Instructions

- Execute the example codes to see how vectorized division works
- Use vector division to create a new column `attitude` in the `lrn14` data frame, where each observation of `Attitude` is scaled back to the original scale of the questions, by dividing it with the number of questions.

Hint: - Assign ‘Attitude divided by 10’ to the new column ’attitude.

R code

```
# This is a code chunk in RStudio editor.
# Work with the exercise in this chunk, step-by-step. Fix the R code!

#lrn14 is available

# divide each number in a vector
c(1,2,3,4,5) / 2
```

```
## [1] 0.5 1.0 1.5 2.0 2.5
```

```
# print the "Attitude" column vector of the lrn14 data
lrn14$Attitude
```

```
##   [1] 37 31 25 35 37 38 35 29 38 21 39 38 24 30 26 25 41 26 26 17 27 39 34 27 23
##  [26] 37 44 41 24 37 25 30 34 32 20 24 42 16 31 38 38 33 17 25 32 35 32 42 31 39
##  [51] 19 21 25 32 32 26 23 38 28 33 48 40 40 47 23 31 27 41 34 34 25 21 14 19 37
##  [76] 41 32 28 41 25 28 38 31 35 36 26 44 45 32 20 39 25 33 35 33 30 29 33 33 35
## [101] 36 42 37 28 42 22 32 50 47 36 36 29 35 40 35 32 26 20 27 32 33 39 33 30 37
## [126] 14 30 25 29 39 36 29 21 31 24 40 31 23 28 37 26 24 30 29 32 28 29 24 31 19
## [151] 20 38 34 37 29 23 41 27 35 34 32 33 33 35 32 31 24 28 17 19 32 35 24 38 21
## [176] 29 19 20 42 41 37 36 18
```

```
# divide each number in the column vector
lrn14$Attitude / 10
```

```
##   [1] 3.7 3.1 2.5 3.5 3.7 3.8 3.5 2.9 3.8 2.1 3.9 3.8 2.4 3.0 2.6 2.5 4.1 2.6
##  [19] 2.6 1.7 2.7 3.9 3.4 2.7 2.3 3.7 4.4 4.1 2.4 3.7 2.5 3.0 3.4 3.2 2.0 2.4
##  [37] 4.2 1.6 3.1 3.8 3.8 3.3 1.7 2.5 3.2 3.5 3.2 4.2 3.1 3.9 1.9 2.1 2.5 3.2
##  [55] 3.2 2.6 2.3 3.8 2.8 3.3 4.8 4.0 4.0 4.7 2.3 3.1 2.7 4.1 3.4 3.4 2.5 2.1
##  [73] 1.4 1.9 3.7 4.1 3.2 2.8 4.1 2.5 2.8 3.8 3.1 3.5 3.6 2.6 4.4 4.5 3.2 2.0
##  [91] 3.9 2.5 3.3 3.5 3.3 3.0 2.9 3.3 3.3 3.5 3.6 4.2 3.7 2.8 4.2 2.2 3.2 5.0
## [109] 4.7 3.6 3.6 2.9 3.5 4.0 3.5 3.2 2.6 2.0 2.7 3.2 3.3 3.9 3.3 3.0 3.7 1.4
## [127] 3.0 2.5 2.9 3.9 3.6 2.9 2.1 3.1 2.4 4.0 3.1 2.3 2.8 3.7 2.6 2.4 3.0 2.9
## [145] 3.2 2.8 2.9 2.4 3.1 1.9 2.0 3.8 3.4 3.7 2.9 2.3 4.1 2.7 3.5 3.4 3.2 3.3
## [163] 3.3 3.5 3.2 3.1 2.4 2.8 1.7 1.9 3.2 3.5 2.4 3.8 2.1 2.9 1.9 2.0 4.2 4.1
## [181] 3.7 3.6 1.8
```

```
# create column 'attitude' by scaling the column "Attitude"
lrn14$attitude <- "Attitude"
```

2.3 Combining variables

Our data includes many questions that can be thought to measure the same *dimension*. You can read more about the data and the variables here. Here we'll combine multiple questions into combination variables. Useful functions for summation with data frames in R are

function	description
`colSums(df)`	returns a sum of each column in `df`
`rowSums(df)`	returns a sum of each row in `df`
`colMeans(df)`	returns the mean of each column in `df`
`rowMeans(df)`	return the mean of each row in `df`

We'll combine the use of `rowMeans()`with the `select()` function from the **dplyr** library to average the answers of selected questions. See how it is done from the example codes.

Instructions

- Access the **dplyr** library
- Execute the example codes to create the combination variables 'deep' and 'surf' as columns in `lrn14`
- Select the columns related to strategic learning from `lrn14`
- Create the combination variable 'stra' as a column in `lrn14`

Hints: - Columns related to strategic learning are in the object `strategic_questions`. Use it for selecting the correct columns. - Use the function `rowMeans()` identically to the examples

R code

```
# Work with the exercise in this chunk, step-by-step. Fix the R code!
# lrn14 is available

# Access the dplyr library
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# questions related to deep, surface and strategic learning
deep_questions <- c("D03", "D11", "D19", "D27", "D07", "D14", "D22", "D30","D06",  "D15", "D23", "D31")
surface_questions <- c("SU02","SU10","SU18","SU26", "SU05","SU13","SU21","SU29","SU08","SU16","SU24","SU
strategic_questions <- c("ST01","ST09","ST17","ST25","ST04","ST12","ST20","ST28")

# select the columns related to deep learning
deep_columns <- select(lrn14, one_of(deep_questions))
# and create column 'deep' by averaging
lrn14$deep <- rowMeans(deep_columns)

# select the columns related to surface learning
surface_columns <- select(lrn14, one_of(surface_questions))
# and create column 'surf' by averaging
lrn14$surf <- rowMeans(surface_columns)

# select the columns related to strategic learning
strategic_columns <- select(lrn14, one_of(strategic_questions))
# and create column 'stra' by averaging
lrn14$stra <- rowMeans(strategic_columns)
```

(more chapters to be added similarly as we proceed with the course!)