Week ONE

This week was all about getting to know the programs and how to use them. I started by reading the GitHub instructions carefully since it was the most unfamiliar tool for me. I created an account for GitHub website and installed GitHub desktop from which I can update my diary with weekly excersices. On the GitHub website I could copy the project template made by course assistants. I could then modify those template for my own course diary.

I have been using R already for some years so this week's DataCamp exercises were quite easy for me. They introduced the basics of R. First exercise was about loading a dataset and looking at what's in it. With already provided code I drew a graph about people's attitudes versus their exam points.

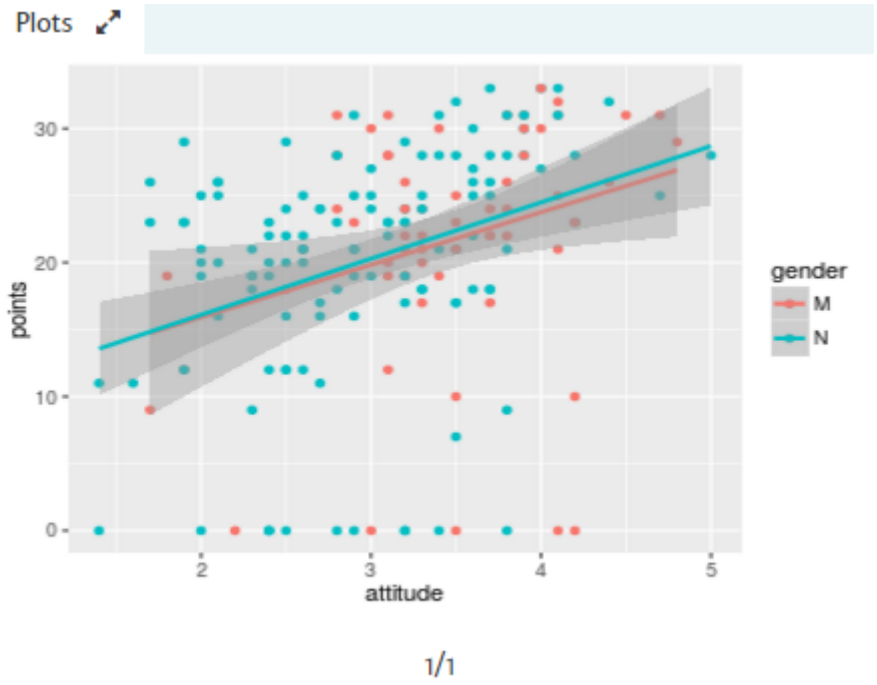


Figure 1: Scatterplot about people's attitudes and exam points.

The rest of the exercises were quite straight-forward as well and by following the instructions they could be done pretty easily. It is always good to refresh one's memory by doing the basic exercises because, to be honest, often I still have to look even the basics up in Google because I have such a short memory :) The most useful exercises for me were the ones were I had to create functions. Here follows an example of a function I created. It counts the number of apples I have if we know the number of apples I had before (my_apples) and the number of apples that have gone bad.

```
# New function here!
good_apples_count <- function(apples, bad_apples=4) return(apples - bad_apples)

# How many good apples do you have?
good_apples_count(my_apples)
```

I hadn't used RMarkdown before so there were definately many new things to learn this week! On this cheatsheet I could find some useful tips on how everything works. First of all I learned the very basics. I learned to write on a document and how to add graphics and code to the document. Besides modifying the

document templates, I also created a README-file for my GitHub repository page. Finally I also tried different themes on my course diary and ended up with *sandstone* because it seemed clear and simple enough for my taste.

Week TWO

This week was about regression and model validation. So in practice, we are trying to explain variables with other variables in the data. In the case of continuous variable to explain, that's done by creating a linear regression model with one dependent variable (the variable to be explained) and one or more explanatory variables. This week we are discovering linear models with some continuous variables.

First of all I created a dataset for my analysis. The Rscript with all the codes I used to create that can be found in my repository in a file called create_learning.R. If you would like to have a more careful look at the variables in the original dataset, you can find the documentation here.

The next thing I did was reading the created dataset back into RStudio. First I needed to set a working directory into the folder where my dataset is and then load the csv-file from there.

```
# Set working directory and call load the dataset into RStudio
setwd("C:/Users/Paula/Documents/GitHub/IODS-project")
students2014<-read.table("learning2014.csv",sep=",",header=T)
```

Reading through this weeks DataCamp exercises I realized that this week I wil need the packages dplyr, ggplot2 and GGally. Since I already installed them earlier (with the command install.packages("packagename")) I only had to call them from my package library.

```
# Call packages needed this week, ggplot2, dplyr and GGally from the library
library(ggplot2)
library(dplyr)
library(GGally)
```

When my dataset was read into RStudio, I first took a good look at it. It has 166 observations and 7 variables. The variables are gender, age, attitude, deep, surf, stra and points. By looking at the structure and the summary of the data the reader can get a clearer image of what kind of variables are they and what kind of values they get in this dataset.

```
str(students2014)
```

```
## 'data.frame':    166 obs. of  7 variables:
##  $ gender  : Factor w/ 2 levels "F","M": 1 2 1 2 2 1 2 1 2 1 ...
##  $ age     : int  53 55 49 53 49 38 50 37 37 42 ...
##  $ attitude: num  3.7 3.1 2.5 3.5 3.7 3.8 3.5 2.9 3.8 2.1 ...
##  $ deep    : num  3.75 2.88 3.88 3.5 3.75 ...
##  $ stra    : num  3.38 2.75 3.62 3.12 3.62 ...
##  $ surf    : num  2.58 3.17 2.25 2.25 2.83 ...
##  $ points  : int  25 12 24 10 22 21 21 31 24 26 ...
```

```
summary(students2014)
```

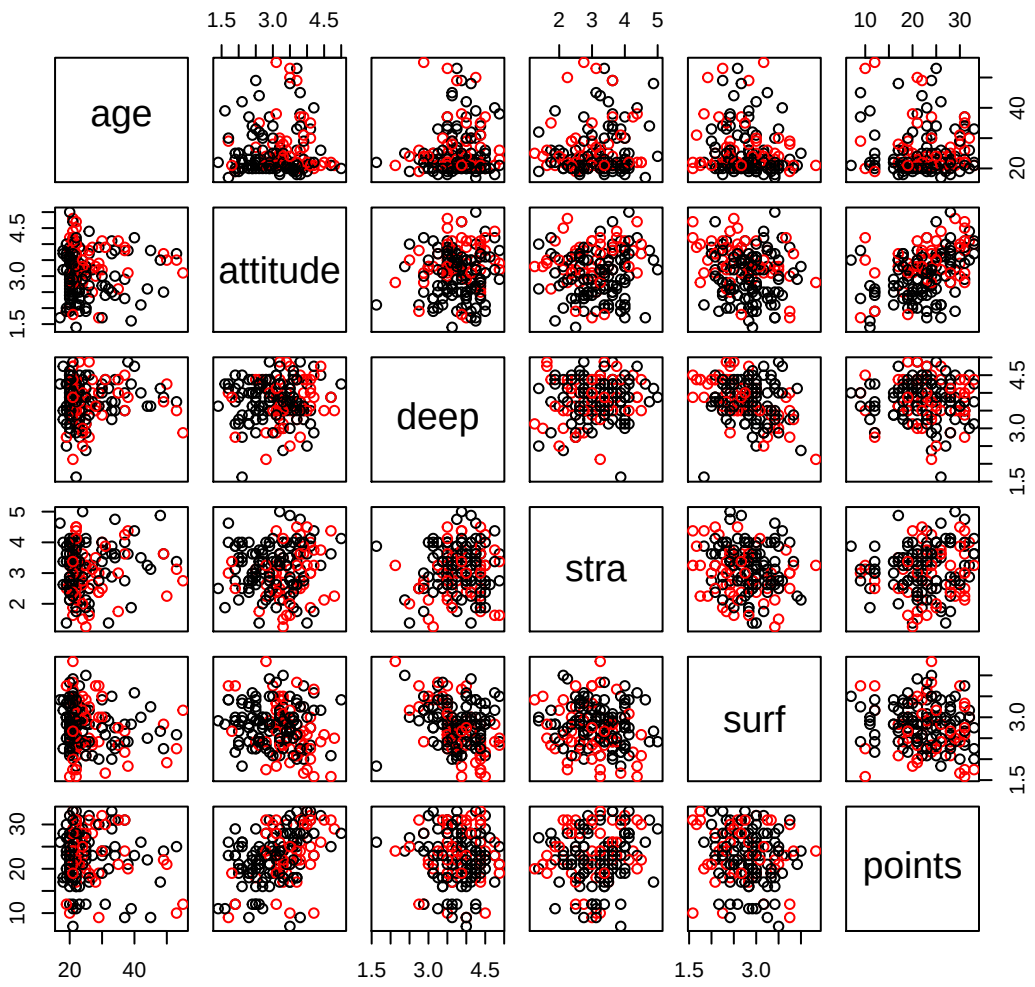
```
##  gender       age           attitude          deep            stra
##  F:110   Min.   :17.00   Min.   :1.400   Min.   :1.625   Min.   :1.250
```

```
##  M: 56   1st Qu.:21.00   1st Qu.:2.600   1st Qu.:3.500   1st Qu.:2.625
##          Median :22.00   Median :3.200   Median :3.875   Median :3.188
##          Mean   :25.51   Mean   :3.143   Mean   :3.796   Mean   :3.121
##          3rd Qu.:27.00   3rd Qu.:3.700   3rd Qu.:4.250   3rd Qu.:3.625
##          Max.   :55.00   Max.   :5.000   Max.   :4.875   Max.   :5.000
##       surf           points
##  Min.   :1.583   Min.   : 7.00
##  1st Qu.:2.417   1st Qu.:19.00
##  Median :2.833   Median :23.00
##  Mean   :2.787   Mean   :22.72
##  3rd Qu.:3.167   3rd Qu.:27.75
##  Max.   :4.333   Max.   :33.00
```

We can see that gender is a factor variable that has two levels, M for male and F for female, age and points are integer variables: Age varies between 17 and 55 while the median age is 22 years, points vary between 7 and 33 and the median value is 23. The median value is the value we get if we put all the values in order and take the one that is in the middle. The rest of the variables are numeric. In fact, deep, stra and surf are variables created by combining other variables in the way presented in the create_learning2014.R-script. deep contains variables connected to deep learning, stra variables connected to studying strategies and surf variables connected to surface learning. attitude is just the original Attitude variable divided by 10 so it will be on the same scale as the rest of the numeric variables in this data. points-variable tells us how much points each student in the dataset got on the exam.

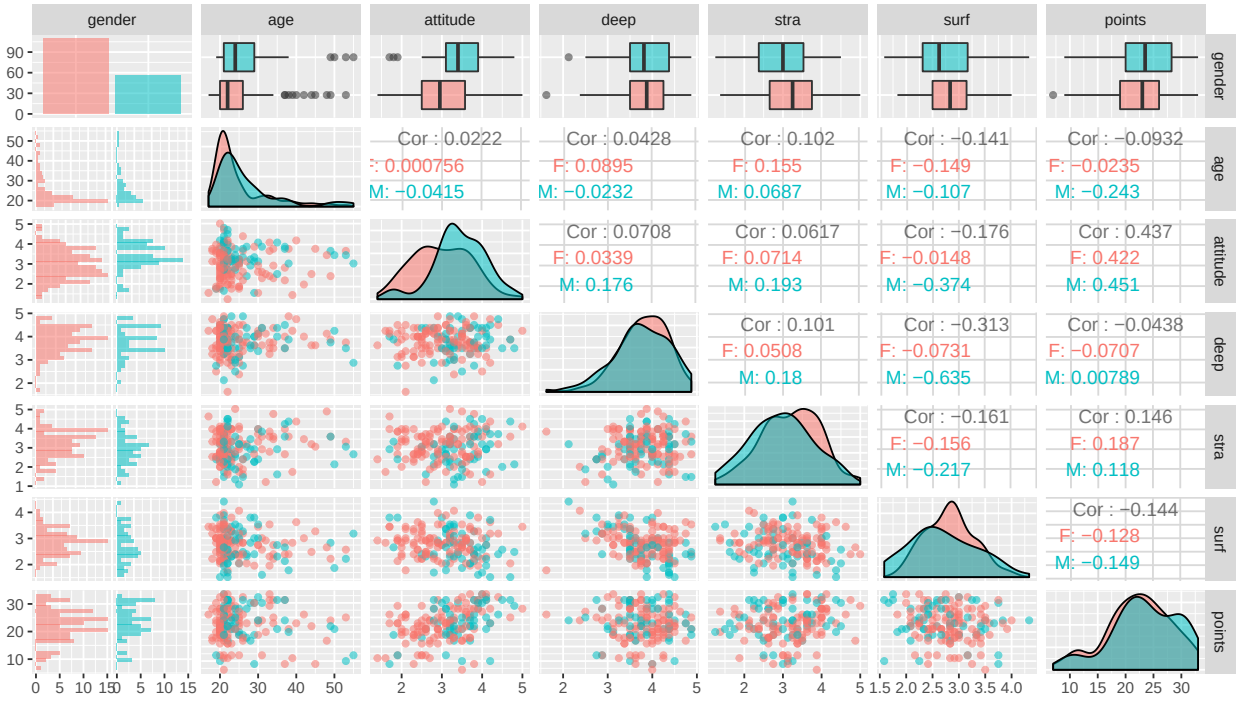
The task was to find variables that could explain the variation of the variable points. So I took a look at the scatterplots between each of the variables in the data but left out gender because for a binary variable scatterplot is not really a meaningful way to go.

```
# Draw a scatter plot matrix of all the variables except in students2014.
pairs(students2014[-1],col=students2014$gender,
      main="Scatterplot of all variables in the data")
```

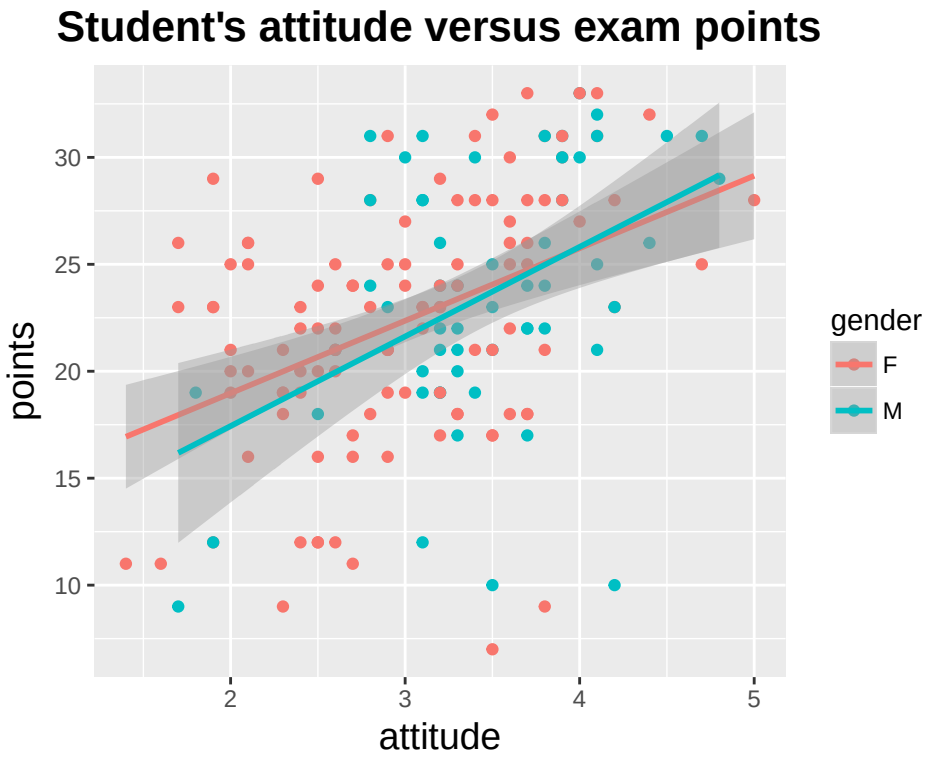


We can see that attitude seems to correlate somewhat strongly with points. Maybe some kind of correlation could be seen also between some other variables but at least for me this graph is not very clear. So I tried to look at the correlations better by plotting it with the instructions found from this week's DataCamp. I drew a plot that presents the correlations of each variable with each variable, for both genders separately and for all the observations, a scatterplot between each pair of variables and a density plot of each variable. For gender that is a binary variable it presents a boxplot that compares how the summary statistics of each variables vary according to gender. It also presents a histogram of a frequency distribution for each variable for both genders separately. In all the plots males are marked with the blueish color and women with red.

```
# create and draw a more advanced plot matrix with ggpairs()
p2 <- ggpairs(students2014, mapping = aes(col=gender,alpha=0.3),
              lower = list(combo = wrap("facethist", bins = 20))) +
  ggtitle("All variables against each other") +
  theme(plot.title = element_text(hjust = 0.5,size=20,face='bold'))
p2
```




```
# Define and draw a plot about students' attitude versus exam points and draw it
p1 <- ggplot(students2014, aes(x = attitude, y = points, col=gender)) +
  geom_point() + geom_smooth(method="lm") +
  ggtitle("Student's attitude versus exam points")+
  theme(plot.title = element_text(hjust = 0.5,size=16,face='bold'),
        axis.title= element_text(hjust = 0.5,size=14))
p1
```




```
# Try to explain points with other variables in the data by fitting
# a linear model
my_model1<-lm(points~attitude+stra+surf,data=students2014)
summary(my_model1)
```

```
##
## Call:
## lm(formula = points ~ attitude + stra + surf, data = students2014)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.1550  -3.4346   0.5156   3.6401  10.8952
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.0171     3.6837   2.991  0.00322 **
## attitude      3.3952     0.5741   5.913 1.93e-08 ***
## stra          0.8531     0.5416   1.575  0.11716
## surf         -0.5861     0.8014  -0.731  0.46563
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.296 on 162 degrees of freedom
## Multiple R-squared:  0.2074, Adjusted R-squared:  0.1927
## F-statistic: 14.13 on 3 and 162 DF,  p-value: 3.156e-08
```

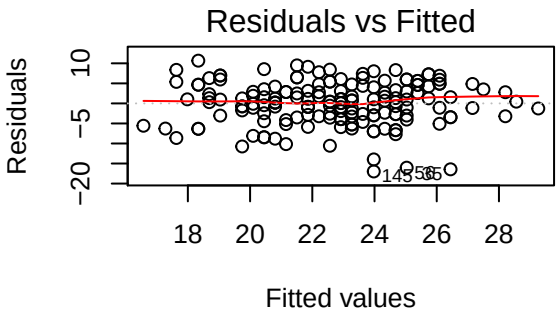
```
my_model2<-lm(points~attitude+stra,data=students2014)
summary(my_model2)
```

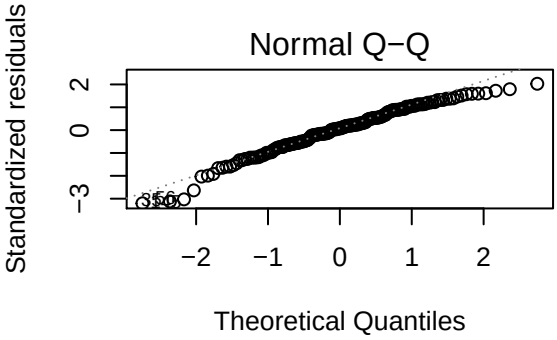
```
##
## Call:
## lm(formula = points ~ attitude + stra, data = students2014)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.6436  -3.3113   0.5575   3.7928  10.9295
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.9729     2.3959   3.745  0.00025 ***
## attitude      3.4658     0.5652   6.132 6.31e-09 ***
## stra          0.9137     0.5345   1.709  0.08927 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.289 on 163 degrees of freedom
## Multiple R-squared:  0.2048, Adjusted R-squared:  0.1951
## F-statistic: 20.99 on 2 and 163 DF,  p-value: 7.734e-09
```

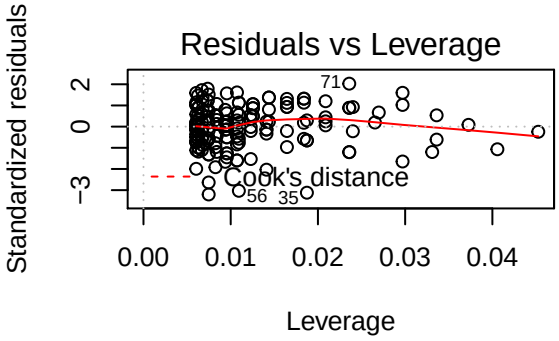
```
my_model3<-lm(points~attitude,data=students2014)
summary(my_model3)
```

```
##
## Call:
## lm(formula = points ~ attitude, data = students2014)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.9763  -3.2119   0.4339   4.1534  10.6645
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.6372     1.8303   6.358 1.95e-09 ***
## attitude      3.5255     0.5674   6.214 4.12e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.32 on 164 degrees of freedom
## Multiple R-squared:  0.1906, Adjusted R-squared:  0.1856
## F-statistic: 38.61 on 1 and 164 DF,  p-value: 4.119e-09
```

```
# Draw diagnostic plots for the model 3 using the plot() function. Choose the plots 1, 2 and 5
par(mfrow=c(2,2))
plot(my_model3,which=c(1,2,5))
```


Residuals vs Fitted
Residuals
Fitted values
18
20
22
24
26
28
−20
−5
10
145
56
65


Normal Q−Q
Standardized residuals
Theoretical Quantiles
−2
−1
0
1
2
−3
0
2


Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
0.00
0.01
0.02
0.03
0.04
−3
0
2
71
56
35