Exercises Day 1 - 1

You may want to work through the (very) short introduction to R (available at `https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf`) before starting these exercises. You can skip section 11 on programming.

Practical questions

Question 1:

```
a <- c(1,6,NA,8,9,10)
mean(a)

## [1] NA
```

a. This gives `NA`, why? Compute the mean using an argument of the `mean()` function.

Solution: The result is NA because there is a missing value

```
mean(a, na.rm=TRUE)

## [1] 6.8
```

b. Now, remove the `NA` from the vector instead and compute the mean.

Solution: The result is NA because there is a missing value

```
mean(na.omit(a))

## [1] 6.8

# or:
mean(a[!is.na(a)])

## [1] 6.8
```

Question 2:

a. Create the following matrix in `R` and store the result in an object

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

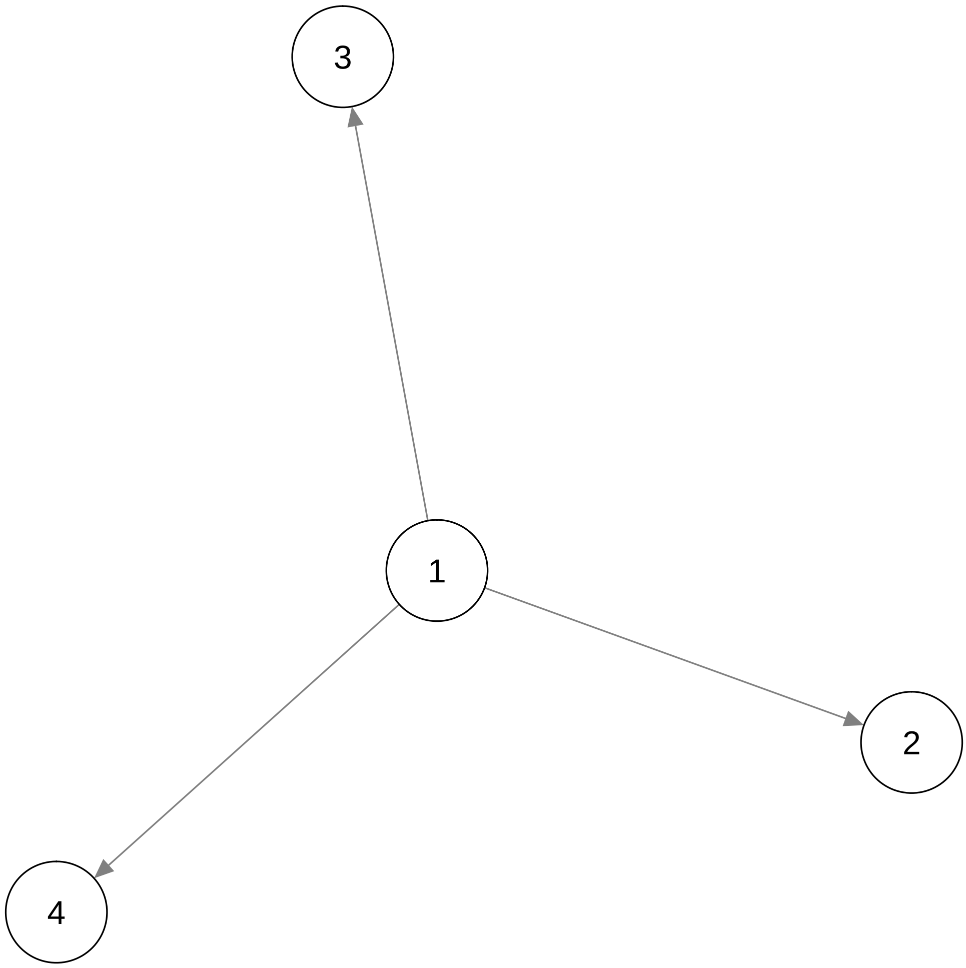
Solution:

```
mymatrix <- matrix(c(
  0,1,1,1,
  0,0,0,0,
  0,0,0,0,
  0,0,0,0
),4,4,byrow=TRUE)
```

b. Install and load the qgraph package and run the function `qgraph(input)` with the object containing your matrix as input. Copy the resulting plot in your report.

Solution:

```
# install.packages("qgraph")
library("qgraph")
qgraph(mymatrix)
```



Question 3:

a. Create a *list* in `R` containing as first element the vector $7, 8, 2, 5$, as second element the vector $2, 8, 1, 7$ and as third element the vector $7, 6, 10, 4$. Call the list `mylist`

Solution:

```
mylist <- list(
  c(7,8,2,5),
  c(2,8,1,7),
  c(7,6,10,4)
)
```

b. Run the commands `sapply(mylist, mean)` and `sapply(mylist, sd)`. Can you explain what happened? What do the `sapply`, `mean` and `sd` functions do?

Solution:

```
sapply(mylist, mean)

## [1] 5.50 4.50 6.75

sapply(mylist, sd)

## [1] 2.645751 3.511885 2.500000
```

I computed the means and standard deviations of all three vectores. Sapply applies a function to each element of a list, mean computes the mean and sd computes the standard deviation.

c. What do the following two commands to? How do they relate to one-another?

```
cbind(mylist[[1]], mylist[[2]], mylist[[3]])
do.call(cbind, mylist)
```

Solution: Both commants put the vectors next to each-otheras columns in a matrix. The second command (using `do.call`) is a more automated version of writing the first command. `do.call` calls a function (first argument), using all elements of a list (second argument) as arguments.

Question 4:

a. Install and load the R package `psych` into R. Once `psych` is loaded, load the 'bfi' dataset:

```
# Load bfi dataset:
data(bfi)

# Look at data:
View(bfi)
```

Where can you find more information about this dataset and what the variables represent?

In the help file:

```r
?bfi
```

b. How many men are in the sample? And how many women?

The help file tells me that males are encoded by 1 and females by 2

```r
# Number of males
sum(bfi$gender == 1)

## [1] 919

# Number of females
sum(bfi$gender == 2)

## [1] 1881
```

c. Create a separate dataset containing only the first 25 columns (the items). Use the psych package to perform an "Exploratory Factor Analysis" (EFA) on this dataset with 5 factors, using "promax" rotation. Where did you find the information needed to perform a EFA using the psych package?

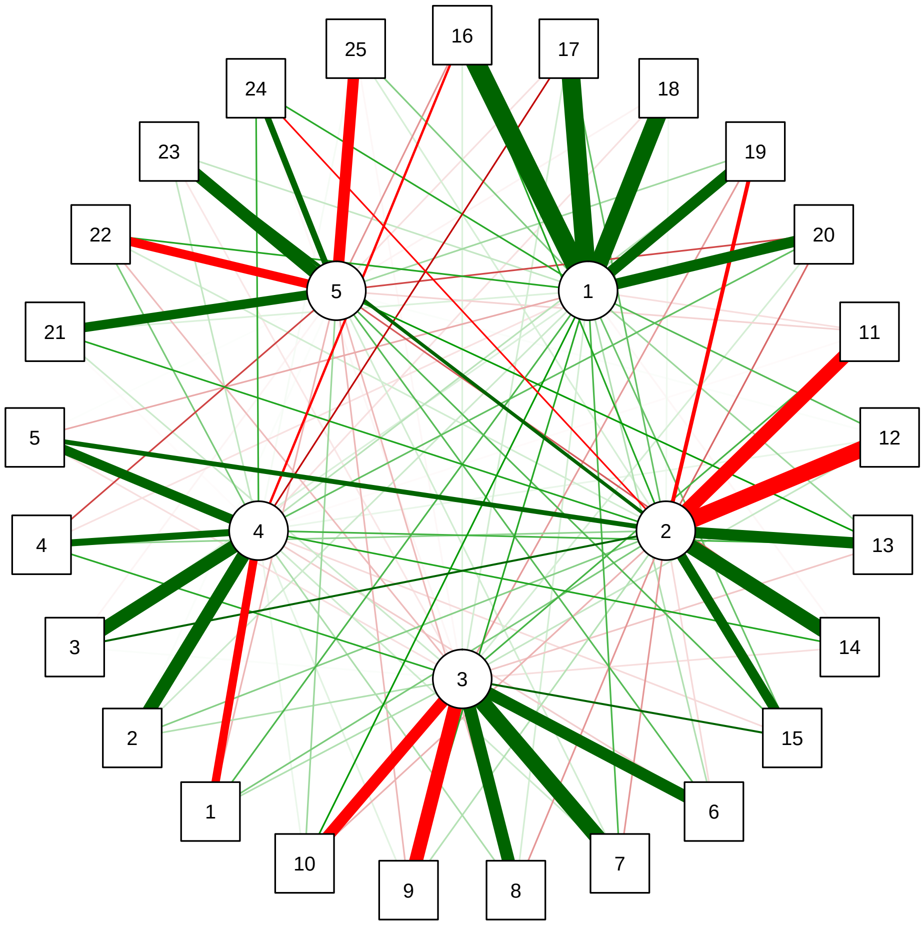
You can find information in the online documentation.

```r
bfiSub <- bfi[,1:25]
efa <- fa(bfiSub,5,rotate = "promax")

## Loading required namespace:  GPArotation
```

d. Make a plot of the estimated factor loadings using the `qgraph.loadings` function of the *qgraph* package. Tip, you might need the `loadings` function.

```r
library("qgraph")
qgraph.loadings(loadings(efa))
```



Question 5

Run the following codes in R:

```
set.seed(112)
Nperson <- 1000
Ability <- rnorm(Nperson)
Item1 <- 1*(exp(Ability)/(1+exp(Ability)) > runif(Nperson))
Item2 <- 1*(exp(Ability)/(1+exp(Ability)) > runif(Nperson))
```

This simulates three objects: `Ability` denotes the mathematical ability of 1000 subjects, and `Item1` and `Item2` represent scores (0 is incorrect, 1 is correct) on two mathematical questions. Those of you with a background in methods will recognize this as the Rasch model. In this model, people with a higher (lower) level of ability have a higher chance of making both items correct (incorrect).

a. Investigate the correlation between the two items. Is the correlation significant? Report both the correlation and the *p*-value.

Solution:

```
cor.test(Item1, Item2)

##
##  Pearson's product-moment correlation
##
## data:  Item1 and Item2
## t = 6.5303, df = 998, p-value = 1.043e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1422256 0.2611496
## sample estimates:
##       cor
## 0.2024338
```

b. Find a way to investigate the *partial correlation* between item 1 and item 2, after controlling for ability. Again, report both the correlation and the *p*-value.

Solution:

```
library("ppcor")

## Loading required package:  MASS

pcor.test(Item1, Item2, Ability)

##     estimate  p.value statistic    n gp  Method
## 1 0.03742729 0.237247  1.182607 1000  1 pearson
```

c. What happened? Can you explain why this happened?

Solution: After conditioning on the latent ability the item scores are no longer correlated. This concept is also known as *local independence*.

Question 6

A company wants to know how job performance relates to IQ, motivation and social support. They collect data on 60 employees, resulting in the file `job_performance.sav` on Dropbox[1] The company made the crucial mistake of asking academics to analyze the data, who cannot afford a copy of SPSS. Hence, we need to analyze the data using R.

a. To load the data, we need to set the working directory. What is a working directory? And how do we set it?

Solution: The directory in which R looks for files. Set it with `setwd()`

b. Load the data into R as a data frame. Tip, the foreign package contains a function for this, but it requires an argument `to.data.frame=TRUE`. There are also other packages you can use.

Solution:

```
library("foreign")
mydata <- read.spss("job_performance.sav", to.data.frame = TRUE)
```

c. Compute and report the correlation matrix of the four numeric items

Solution:

```
cor(mydata[,-1])

##           perf          iq         mot         soc
## perf 1.0000000  0.47378037 0.63486569  0.39692818
## iq   0.4737804  1.00000000 0.04687791 -0.09180517
## mot  0.6348657  0.04687791 1.00000000  0.36339641
## soc  0.3969282 -0.09180517 0.36339641  1.00000000
```

d. Run the following command:

```
fit <- lm(perf ~ iq + mot + soc, data = mydata)
```

Where `mydata` is the object containing your dataset. What analysis did you perform?

Solution: A multiple regression analysis

[1] I have taken this example from https://www.spss-tutorials.com/linear-regression-in-spss-example/

d. Investigate the object `fit` to see if IQ is a significant predictor of performance.

Solution:

```
fit <- lm(perf ~ iq + mot + soc, data = mydata)
summary(fit)

##
## Call:
## lm(formula = perf ~ iq + mot + soc, data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1376  -2.8375  -0.1956   3.4518  11.1904
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.13146    6.34603   2.857  0.00599 **
## iq           0.26487    0.04440   5.965 1.73e-07 ***
## mot          0.30820    0.05001   6.163 8.24e-08 ***
## soc          0.16396    0.05551   2.953  0.00459 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.844 on 56 degrees of freedom
## Multiple R-squared:  0.6545,Adjusted R-squared:  0.636
## F-statistic: 35.36 on 3 and 56 DF,  p-value: 5.908e-13
```

IQ is a significant predictor of performance.