FA/SEM Assignment 1 report

Päivimaria Mäkelä

2023-01-23

In my weekly report, the text of the original assignment is in italics, and my own additions are in normal font.

Basics of Exploratory Factor Analysis and Linear Regression

Start and get data

The data set that we use only in the Assignment 1, is a famous demo data used in the statistical computing environment Survo (www.survo.fi), authored by professor of Statistics, Seppo Mustonen.

The data includes the 48 world's best athletes of decathlon in 1973, their individual scores and the height and weight of those athletes.

The decathlon sports are

Day 1	**Day 2**
• 100 metres	• 110 metres hurdles
• Long jump	• Discus throw
• Shot put	• Pole vault
• High jump	• Javelin throw
• 400 metres	• 1500 metres

Stop for a minute to think about the phenomenon: *the physical fitness of a human-being. What kind of dimensions might the physical fitness include, what do you think? The first ones you might think about may include at least SPEED and POWER (I will use those in my example code below). But, how about other dimensions of physical fitness? What might they be?*

Roughly divided, the decathlon includes three types of sports. Sports where you run, sports where you jump and sports where you throw things. Some of these overlap: for example, high jump and javelin throw require running before jumping. On the other hand, hurdles requires jumping as well as running.

Here we think that those dimensions (SPEED, POWER, etc.) represent latent variables (factors) that cannot be measured directly. In this example, they are measured by those 10 decathlon sports, the scores of which are included in the data set.

Power and speed are natural measures of physical fitness. In addition to these, at least endurance comes to mind. Granted, I don't know if endurance is a proper English sports term.

Let us proceed!

Table 2: First six observations of the Decathlon data

Name	Points	Run100m	Longjump	Shotput	Hijump	Run400m	Hurdles	Discus	Polevlt	Javelin	Run1500m	Height	Weight
Skowron	8206	853	931	725	857	838	903	772	981	818	528	184	81
Hedmark	8188	853	853	814	769	833	914	855	884	975	438	195	90
Le_Roy	8140	879	951	799	779	838	881	819	1028	758	408	191	90
Zeilbaue	8136	826	931	793	865	875	891	729	909	774	543	192	84
Zigert	8134	879	840	924	857	788	892	866	920	671	497	198	105
Bennett	8121	905	859	647	779	938	859	651	1028	794	661	173	68

```
# load the libraries (install the packages first!)
library(tidyverse) # install.packages("tidyverse")
library(corrplot) # install.packages("corrplot")
library(psych) # install.packages("psych")
library(GGally) # install.packages("GGally")

# additional packages

# car: Companion to Applied Regression
#install.packages("car")
library(car)

# Tools to Accompany the 'psych' Package for Psychological Research
#install.packages("psychTools")
library(psychTools)

# Construct Complex Table with 'kable' and Pipe Syntax
#install.packages("kableExtra")
library(kableExtra)
```

```
# read the data from the Survo website:
deca <- read.table("https://survo.fi/data/Decathlon.txt", header = TRUE, sep = '\t')
```

```
# browse the data:
#View(deca)
head(deca) %>%
  kbl(caption = "First six observations of the Decathlon data") %>%
  kable_classic(full_width = T)
```

```
# (as you can see, the names are shortened to max 8 characters)
# you can also see, kableExtra prints the table where ever it wants when knitting.
# It also adds numbering automatically.

# names as row names
row.names(deca) <- deca$Name
```

Based on a quick Google search, I came to the conclusion that decathlon scores are not a linear transformation of performance (in meters or seconds to points). This makes me wonder if it would be better to create a factorial model based on performance.

Table 3: Summary statistics of the 1st day sports

	Min.	1st Qu.	Median	3rd Qu.	Max.
Run100m	712	780.00	828	879.00	932
Longjump	725	806.00	836	869.50	951
Shotput	604	708.75	751	774.75	924
Hijump	689	769.00	804	857.00	925
Run400m	699	784.00	819	839.00	938

```
# check the structure of the data:
str(deca)
```

```
## 'data.frame':    48 obs. of  14 variables:
##  $ Name    : chr  "Skowrone" "Hedmark" "Le_Roy" "Zeilbaue" ...
##  $ Points  : int  8206 8188 8140 8136 8134 8121 8100 8020 8016 7977 ...
##  $ Run100m : int  853 853 879 826 879 905 879 853 804 853 ...
##  $ Longjump: int  931 853 951 931 840 859 848 828 848 830 ...
##  $ Shotput : int  725 814 799 793 924 647 785 772 795 815 ...
##  $ Hijump  : int  857 769 779 865 857 779 804 751 831 822 ...
##  $ Run400m : int  838 833 838 875 788 938 766 838 819 784 ...
##  $ Hurdles : int  903 914 881 891 892 859 807 987 837 817 ...
##  $ Discus  : int  772 855 819 729 866 651 897 748 801 762 ...
##  $ Polevlt : int  981 884 1028 909 920 1028 909 960 884 859 ...
##  $ Javelin : int  818 975 758 774 671 794 820 755 755 848 ...
##  $ Run1500m: int  528 438 408 543 497 661 585 528 642 587 ...
##  $ Height  : int  184 195 191 192 198 173 190 184 189 186 ...
##  $ Weight  : int  81 90 90 84 105 68 90 81 89 87 ...
```

```
# or, alternatively, using the tidyverse tools:
#glimpse(deca)
```

Know your data

While I'm at it, I will use the opportunity to compile some code I may need during the course.

The 1st day sports

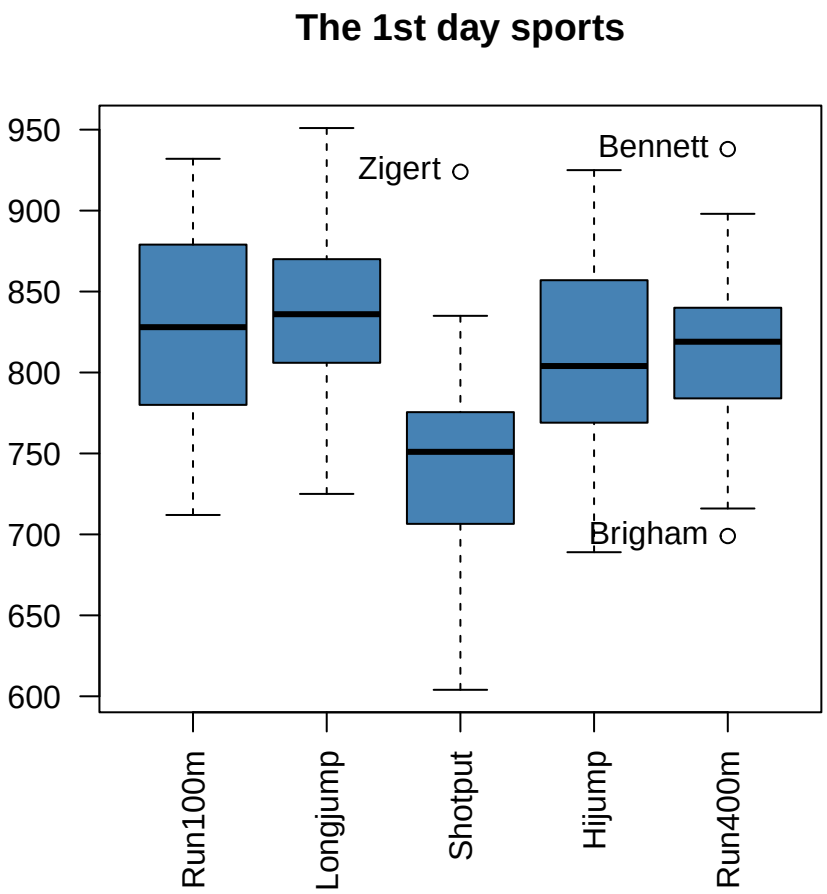
```
# calculate some statistics
deca_des1  <- describe(deca[, 3:7], ranges = FALSE, quant = c(0, 0.25, 0.5, 0.75, 1))
deca_des1 <- as.data.frame(deca_des1)
names(deca_des1) <- c("vars", "n", "mean", "sd", "skew", "kurtosis", "se",
                      "Min.", "1st Qu.", "Median", "3rd Qu.", "Max.")
```

```
# tables
# Summary
deca_des1[8:12] %>%
  kbl(caption = "Summary statistics of the 1st day sports") %>%
  kable_classic(full_width = T)
```

Table 4: Descriptive statistics of the 1st day sports

	n	mean	sd	skew	kurtosis	se
Run100m	48	828.19	59.30	-0.08	-1.14	8.56
Longjump	48	840.19	50.73	0.09	-0.16	7.32
Shotput	48	740.77	61.83	-0.08	0.43	8.92
Hijump	48	805.85	64.81	0.03	-0.90	9.35
Run400m	48	813.50	49.80	0.01	-0.20	7.19

```
Boxplot(deca[, 3:7], col="steelblue", main="The 1st day sports", las=2)
```




```
## [1] "Zigert"  "Brigham" "Bennett"
```

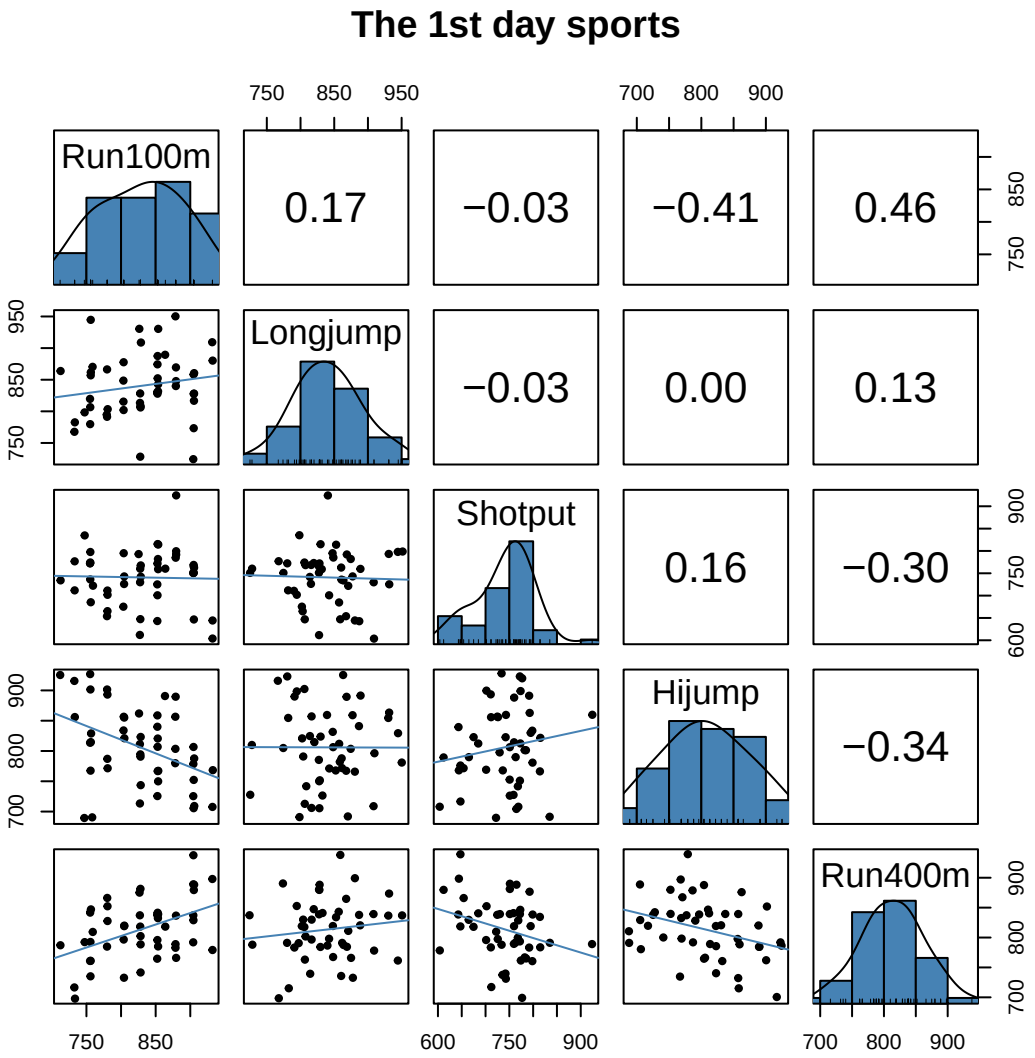
```
# Descriptives
round(deca_des1[,2:7], 2) %>%
  kbl(caption = "Descriptive statistics of the 1st day sports") %>%
  kable_classic(full_width = T)
```

I don't know decathlon as a sport, so I can't say if the scores in the data are typical for the sports. There are only a few outliers in the first day's results, but no athlete has performed exceptionally well or poorly in more than one sport.

Based on the plots and descriptive statistics, the distributions of the variables do not look badly skewed.

A more accurate interpretation of the statistics would be more meaningful if I knew the decathlon better and could relate the means and standard deviations of the data to the statistics typical of the sport.

```
# visualize the 1st day sports with a scatterplot:
#deca %>% select(Run100m:Run400m) %>% ggpairs()

pairs.panels(deca[, 3:7], smooth = FALSE, scale=FALSE, ellipses = FALSE,
             jiggle=TRUE, density = TRUE, lm=TRUE,
             hist.col = "steelblue", col="steelblue",
             main="The 1st day sports")
```



There are not very strong correlations between sports on the first day. In my opinion, the most surprising thing is that the correlation between long jump and high jump scores is zero. On the other hand, the negative correlation between the 100m run points and the high jump points is not surprising at all.

Table 5: Summary statistics of the 2st day sports

	Min.	1st Qu.	Median	3rd Qu.	Max.
Hurdles	726	824.50	849.5	885.00	987
Discus	607	716.50	747.0	784.75	897
Polevlt	780	859.00	909.0	932.00	1052
Javelin	650	716.75	756.5	794.75	975
Run1500m	378	503.75	558.0	605.75	696

The 2nd day sports

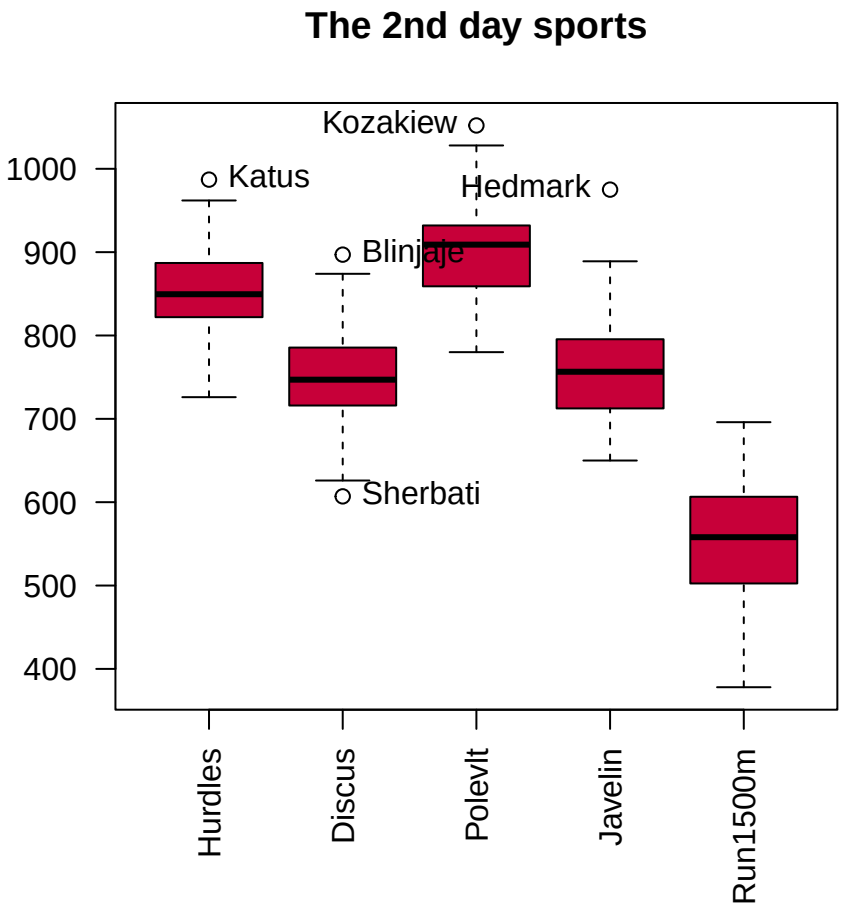
```
# calculate some statistics
deca_des2  <- describe(deca[, 8:12], ranges = FALSE, quant = c(0, 0.25, 0.5, 0.75, 1))
deca_des2 <- as.data.frame(deca_des2)
names(deca_des2) <- c("vars", "n", "mean", "sd", "skew", "kurtosis", "se",
                      "Min.", "1st Qu.", "Median", "3rd Qu.", "Max.")
```

```
# tables
# Summary
deca_des2[8:12] %>%
  kbl(caption = "Summary statistics of the 2st day sports") %>%
  kable_classic(full_width = T)
```

```
Boxplot(deca[, 8:12], col="#C70039", main="The 2nd day sports", las=2)
```

Table 6: Descriptive statistics of the 2st day sports

	n	mean	sd	skew	kurtosis	se
Hurdles	48	852.88	54.20	-0.04	-0.05	7.82
Discus	48	747.46	62.28	0.08	-0.04	8.99
Polevlt	48	900.27	63.04	0.36	-0.38	9.10
Javelin	48	760.02	63.94	0.77	1.01	9.23
Run1500m	48	554.62	76.67	-0.22	-0.63	11.07



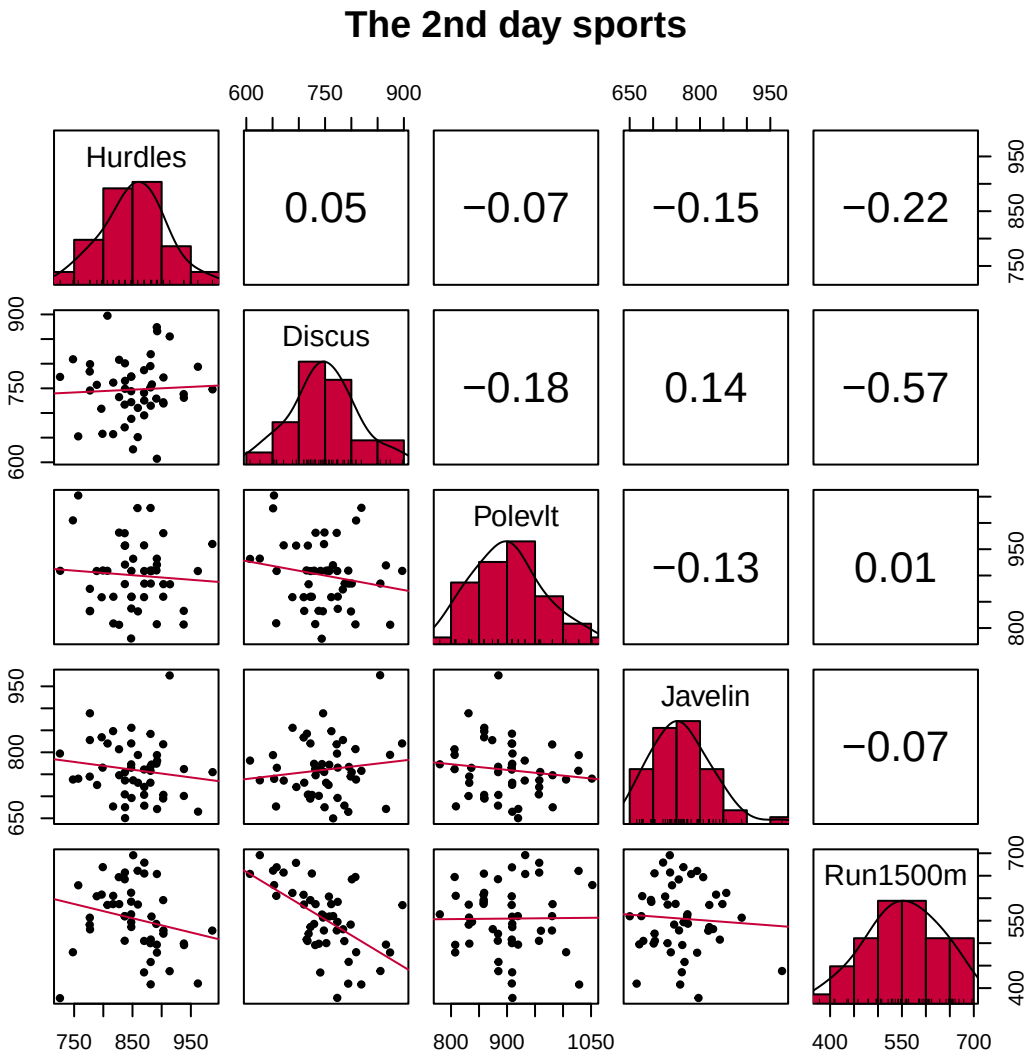

```
## [1] "Katus"    "Sherbati" "Blinjaje" "Kozakiew" "Hedmark"
```

```
# Descriptives
round(deca_des2[,2:7], 2) %>%
  kbl(caption = "Descriptive statistics of the 2st day sports") %>%
  kable_classic(full_width = T)
```

Again, I can't say if the scores in the data are typical for the sports. And again, there are only a few outliers in the second day's results, but no athlete has performed exceptionally well or poorly in more than one sport.

```
# visualize the 2nd day sports with a scatterplot:
#deca %>% select(Hurdles:Run1500m) %>% ggpairs()
```

```
pairs.panels(deca[, 8:12], smooth = FALSE, scale=FALSE, ellipses = FALSE,
             jiggle=TRUE, density = TRUE, lm=TRUE,
             hist.col = "#C70039", col="#C70039",
             main="The 2nd day sports")
```



The strongest correlation of scores in sports on the second day is between the discus throw and the 1500m run - and that is negative. Otherwise, the correlations are not very strong. It might be possible to draw preliminary conclusions regarding the factor model, but I don't know enough about the subject to dare to speculate.

Based on the descriptors and key figures, the distribution of the javelin scores in particular looks skewed.

Height and Weight

Table 7: Summary statistics of height and weight

	Min.	1st Qu.	Median	3rd Qu.	Max.
Height	173	184	188	190.00	198
Weight	68	82	86	89.25	105

Table 8: Descriptive statistics of height and weight

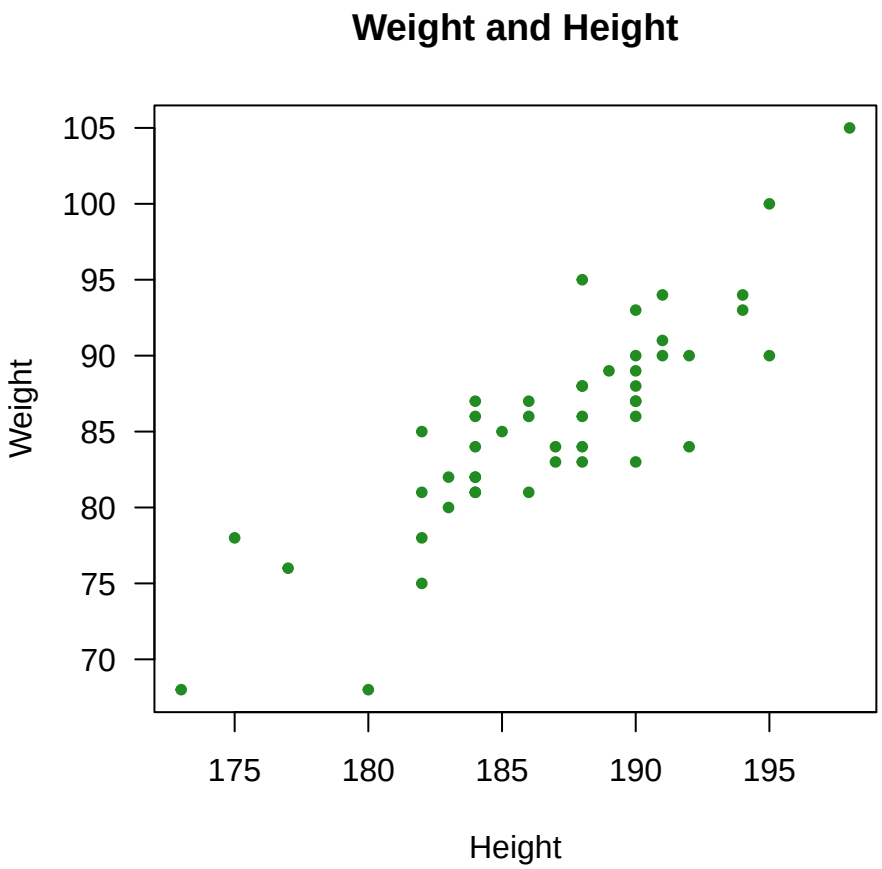
	n	mean	sd	skew	kurtosis	se
Height	48	186.96	5.09	-0.44	0.26	0.73
Weight	48	85.56	6.85	-0.04	1.11	0.99

```
# calculate some statistics
deca_des3  <- describe(deca[, 13:14], ranges = FALSE, quant = c(0, 0.25, 0.5, 0.75, 1))
deca_des3 <- as.data.frame(deca_des3)
names(deca_des3) <- c("vars", "n", "mean", "sd", "skew", "kurtosis", "se",
                      "Min.", "1st Qu.", "Median", "3rd Qu.", "Max.")
```

```
# tables
# Summary
deca_des3[8:12] %>%
  kbl(caption = "Summary statistics of height and weight") %>%
  kable_classic(full_width = T)
```

```
# Descriptives
round(deca_des3[,2:7], 2) %>%
  kbl(caption = "Descriptive statistics of height and weight") %>%
  kable_classic(full_width = T)
```

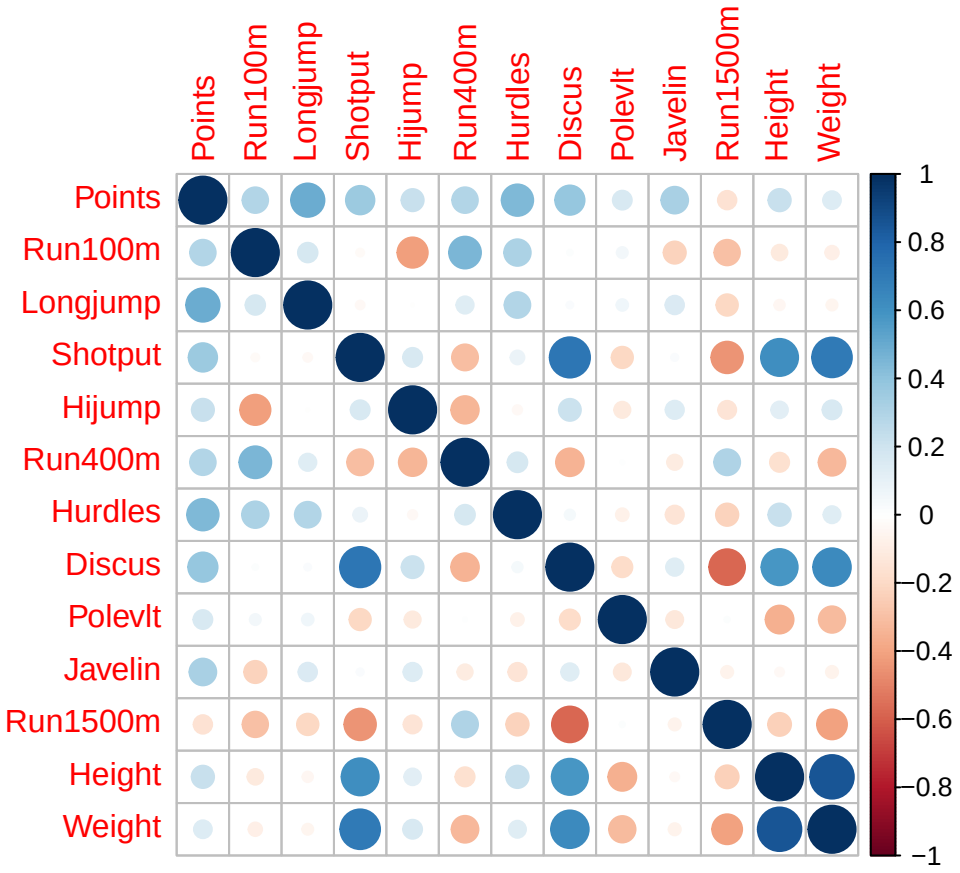
```
# visualize the scatterplot of Height and Weight:
#deca %>% ggplot(aes(x = Height, y = Weight)) + geom_point()

plot(NULL, xaxt = 'n', yaxt = 'n',
     xlim = range(min(deca$Height):max(deca$Height)),
     ylim = range(min(deca$Weight):max(deca$Weight)),
     xlab = "Height", ylab = "Weight",
     main = "Weight and Height")
axis(1, at=c(seq(from=min(deca$Height)+2, to=max(deca$Height)+2, by=5)),
     labels=c(seq(from=min(deca$Height)+2, to=max(deca$Height)+2, by=5)))
axis(2, at=c(seq(from=min(deca$Weight)+2, to=max(deca$Weight)+2, by=5)),
     labels=c(seq(from=min(deca$Weight)+2, to=max(deca$Weight)+2, by=5)), las=1)
points(x = deca$Height, y = deca$Weight, pch = 20,col = "#228b22")
```



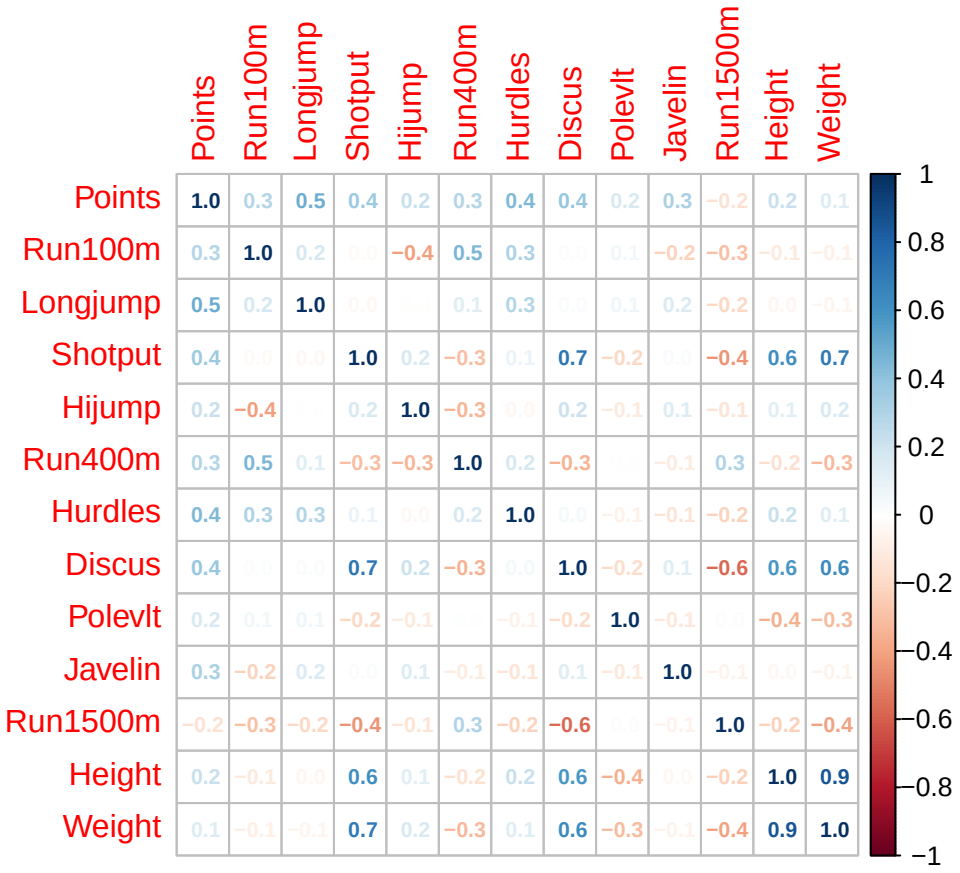
Even though I can't think of anything worth commenting on about the distributions of height and weight, it is still good to have an idea of the range, mean and standard deviation of the values of the variable. From the scatter pattern, we can conclude that there is a strong linear relationship between height and weight.

Correlation plots

```
# check the (visual) correlation matrix of all the numeric variables:
# (here we use the `pipe`, an essential part of the tidyverse style of R)
deca %>% select(-Name) %>% cor() %>% corrplot()
```




```
# another variation (see the help page of corrplot() for more arguments)
deca %>% select(-Name) %>% cor() %>%
  corrplot(method = 'number', number.cex = 0.7, number.digits = 1)
```



Both of these correlation plots are great, although I prefer the one where I can also see the numerical values of the correlations. The graph would be even better if you could also find out about the values of correlations close to zero. There is a fairly strong correlation between scores in some sports (eg the correlation between discus and shot put is 0.7 and the correlation between discus and 1500m is -0.60). However, by simply looking at the correlation matrix, I would not be able to decide which of the variables load on the same factors.

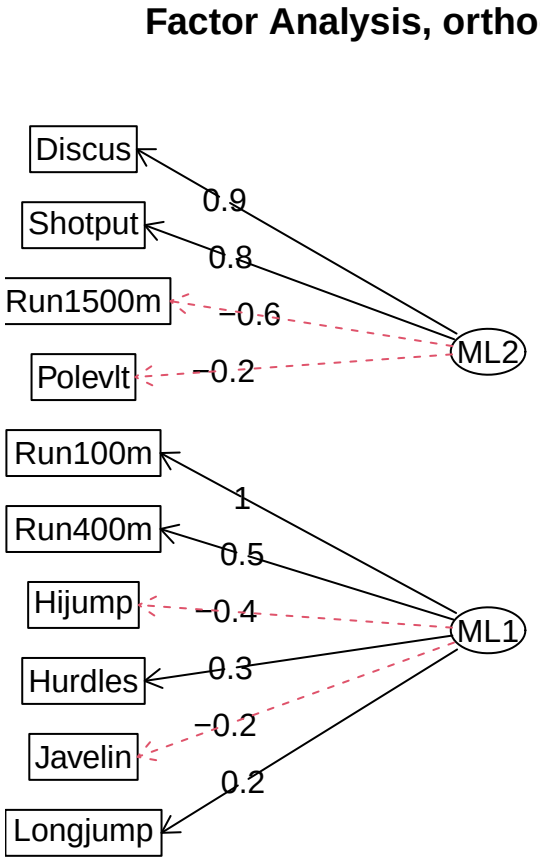
Exploratory Factor Analysis

Run Exploratory Factor Analysis of the 10 scores variables with different number of factors (`nfactors`). If you thought about the number of the dimensions above, that should give you a good starting point - but try different variations, anyway.

Two factors

```
# Exploratory Factor Analysis using only the 10 scores variables:
# (We will use only the Maximum Likelihood estimation and Varimax rotation here)
fadeca2 <- fa(deca %>% select(Run100m:Run1500m), nfactors=2, fm="ml",
              rotate = "varimax", scores = "regression")
```

```r
# Draw a diagram showing the factors and the items with arrows and factor loadings:
fa.diagram(fadeca2, cut = 0, main = "Factor Analysis, orthogonal rotation")
```




```r
# Check the factor matrix (especially factor loadings and communalities, column h2):
#print(fadeca2, digits = 2, sort = TRUE)

# Some statistics

fadeca2$loadings
```

```
##
## Loadings:
##           ML2    ML1
## Run100m          0.997
## Longjump         0.172
## Shotput   0.779
## Hijump    0.252 -0.417
## Run400m  -0.405  0.464
## Hurdles          0.316
## Discus    0.911
## Polevlt  -0.191
```

Table 9: Factor communalities, 2 factors

	x
Run100m	0.995
Longjump	0.031
Shotput	0.609
Hijump	0.238
Run400m	0.379
Hurdles	0.107
Discus	0.830
Polevlt	0.040
Javelin	0.066
Run1500m	0.492

```
## Javelin   0.125 -0.224
## Run1500m -0.643 -0.280
##
##                  ML2   ML1
## SS loadings    2.139 1.648
## Proportion Var 0.214 0.165
## Cumulative Var 0.214 0.379
```

```
round(fadeca2$communalities, 3) %>%
  kbl(caption = "Factor communalities, 2 factors") %>%
  kable_classic(full_width = F)
```

```
# degrees of freedom
# I don know how to extract these from the fadeca statistics,
# so I'll just calculate it with the formula in MABS4FASEM p. 10
df2 <- 0.5*(10 - 2)^2 - 0.5*(10 + 2)

Model1 <- c(2, fadeca2$STATISTIC, df2, fadeca2$PVAL, sum(fadeca2$communalities))
round(Model1, 2)
```

```
## [1]  2.00 20.11 26.00  0.79  3.79
```

All square sums of factor loadings are over 1.

Test of the hypothesis 'is two factors sufficient' gives the chi-square test statistics of 20.11 with 26 degrees of freedom and p-value of 0,79. This suggests that two factors are sufficient.

On p. 14 in MABS4FASEM we have a guideline for *a simple structure*. According to it, every row of factor loadings should have at least one zero. This pattern has four rows with no zeros. Accordingly, each column should have at least as many zeros as the number of factors. There are three zeros in each column, so this will come true after all.

The communalities of the factors tells how well variables are explained by the model. The closer the statistics is to 1, the better the variable is explained. The model explains well some of the variables, such as the 100 m run and the discus throw. Instead, for example, the communality of the long jump is only 0.031, the pole jump is 0.040 and the javelin throw is only 0.066.

```
# Save the factor scores as a new data set, copy the athlete names from the original set,
# and rename the factor score variables according to the interpretation:
scores2 <- as.data.frame(fadeca2$scores) %>%
  mutate(Name = deca$Name) %>%
  rename(SPEED = ML1, POWER = ML2)

# Save the factor scores (by regression method) as new variables in the original data set:
decaNEW2 <- left_join(deca, scores2)
```

Three factors

```
# Exploratory Factor Analysis using only the 10 scores variables:
# (We will use only the Maximum Likelihood estimation and Varimax rotation here)
fadeca3 <- fa(deca %>% select(Run100m:Run1500m), nfactors=3, fm="ml",
              rotate = "varimax", scores = "regression")
```

```
# Draw a diagram showing the factors and the items with arrows and factor loadings:
fa.diagram(fadeca3, cut = 0, main = "Factor Analysis, orthogonal rotation")
```

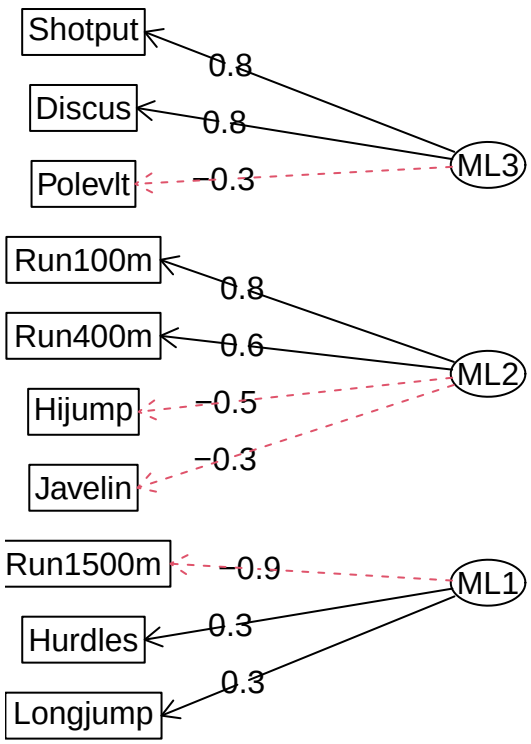


Table 10: Factor communalities, 3 factors

	x
Run100m	0.886
Longjump	0.082
Shotput	0.733
Hijump	0.275
Run400m	0.471
Hurdles	0.135
Discus	0.745
Polevlt	0.073
Javelin	0.072
Run1500m	0.995

```
# Check the factor matrix (especially factor loadings and communalities, column h2):
#print(fadeca3, digits = 2, sort = TRUE)

# Some statistics

fadeca3$loadings
```

```
##
## Loadings:
##          ML3    ML2    ML1
## Run100m          0.778  0.529
## Longjump                0.272
## Shotput   0.831 -0.160  0.132
## Hijump    0.126 -0.508
## Run400m  -0.236  0.642
## Hurdles          0.220  0.288
## Discus    0.791 -0.213  0.274
## Polevlt  -0.255
## Javelin         -0.265
## Run1500m -0.345  0.255 -0.901
##
##                  ML3   ML2   ML1
## SS loadings    1.582 1.533 1.354
## Proportion Var 0.158 0.153 0.135
## Cumulative Var 0.158 0.312 0.447
```

```
round(fadeca3$communalities, 3) %>%
  kbl(caption = "Factor communalities, 3 factors") %>%
  kable_classic(full_width = F)
```

```
# degrees of freedom
# I don know how to extract these from the fadeca statistics,
# so I'll just calculate it with the formula in MABS4FASEM p. 10
df3 <- 0.5*(10 - 3)^2 - 0.5*(10 + 3)

Model2 <- c(3, fadeca3$STATISTIC, df3, fadeca3$PVAL, sum(fadeca3$communalities))
round(Model2, 2)
```

```
## [1]  3.00 10.01 18.00  0.93  4.47
```

All square sums of factor loadings are over 1.

Test of the hypothesis 'is two factors sufficient' gives the chi-square test statistics of 10.01 with 18 degrees of freedom and p-value of 0,93. This suggests that three factors are sufficient.

Now only one row has no zeros in the loadings of the factor, but correspondingly, there are only two zeros in the factor 2 column (not three, as would be desirable).

When we look at the communalities, we notice that the 1500-meter run, the 100-meter run, the shot put, and the discus throw are well explained by this model, but there are still variables that the model does not explain very well.

Four factors

```
# Exploratory Factor Analysis using only the 10 scores variables:
# (We will use only the Maximum Likelihood estimation and Varimax rotation here)
fadeca4 <- fa(deca %>% select(Run100m:Run1500m), nfactors=4, fm="ml",
              rotate = "varimax", scores = "regression")
```

```
# Draw a diagram showing the factors and the items with arrows and factor loadings:
fa.diagram(fadeca4, cut = 0, main = "Factor Analysis, orthogonal rotation")
```

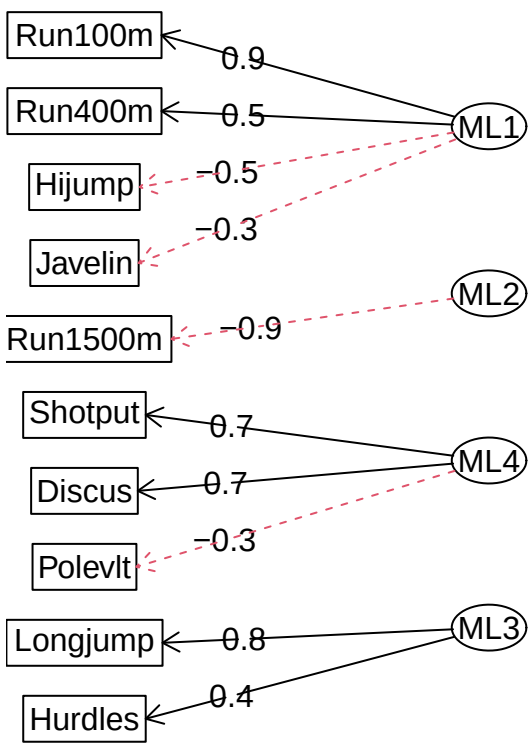


Table 11: Factor communalities, 4 factors

	x
Run100m	0.958
Longjump	0.577
Shotput	0.707
Hijump	0.275
Run400m	0.536
Hurdles	0.210
Discus	0.773
Polevlt	0.092
Javelin	0.117
Run1500m	0.820

```
# Check the factor matrix (especially factor loadings and communalities, column h2):
#print(fadeca4, digits = 2, sort = TRUE)

# Some statistics

fadeca4$loadings
```

```
##
## Loadings:
##          ML1    ML2    ML4    ML3
## Run100m   0.884  0.277         0.314
## Longjump                       0.752
## Shotput  -0.116  0.394  0.733
## Hijump   -0.493  0.146
## Run400m   0.545 -0.402         0.262
## Hurdles   0.182  0.120         0.400
## Discus   -0.140  0.534  0.683
## Polevlt                -0.298
## Javelin  -0.302                0.147
## Run1500m        -0.854 -0.153 -0.258
##
##                  ML1   ML2   ML4   ML3
## SS loadings    1.488 1.446 1.145 0.984
## Proportion Var 0.149 0.145 0.114 0.098
## Cumulative Var 0.149 0.293 0.408 0.506
```

```
round(fadeca4$communalities, 3) %>%
  kbl(caption = "Factor communalities, 4 factors") %>%
  kable_classic(full_width = F)
```

```
# degrees of freedom
# I don know how to extract these from the fadeca statistics,
# so I'll just calculate it with the formula in MABS4FASEM p. 10
df4 <- 0.5*(10 - 4)^2 - 0.5*(10 + 4)

Model3 <- c(4, fadeca4$STATISTIC, df4, fadeca4$PVAL, sum(fadeca4$communalities))
round(Model3, 2)
```

Table 12: Some statistics of the models

	Number of Factors	Chi-square	df	p-value	Sum of Communalities
Model1	2	20.11	26	0.79	3.79
Model2	3	10.01	18	0.93	4.47
Model3	4	4.97	11	0.93	5.06

```
## [1]  4.00  4.97 11.00  0.93  5.06
```

Not all square sums of factor loadings are over 1.

Test of the hypothesis 'is two factors sufficient' gives the chi-square test statistics of 4,97 with 18 degrees of freedom and p-value of 0,93. This suggests that four factors are sufficient.

Now all rows of factor loadings have at least one zero, but two columns have less than 4 zeros.

When we look at the communalities, we notice that even four factors do not explain all the variables very well, for example the communality of the pole vault is only 0.092.

Try to interpret the (rotated) factor solutions. ***What do you think is the best number of factors and why?***

```
# Combine statistics
models <- t(data.frame(Model1, Model2, Model3))
colnames(models) <- c("Number of Factors",
                      "Chi-square", "df", "p-value",
                      "Sum of Communalities")
round(models, 2) %>%
  kbl(caption = "Some statistics of the models") %>%
  kable_classic(full_width = T)
```

No model was anywhere near "perfect," but based on the results above, I would be inclined to go with the three-factor model. I think it explains the phenomenon better than the two-factor model, and I think that the four-factor model would be an over-explanation.

In the table 12, I have compiled some key statistics of the models, which can be used to compare the models.

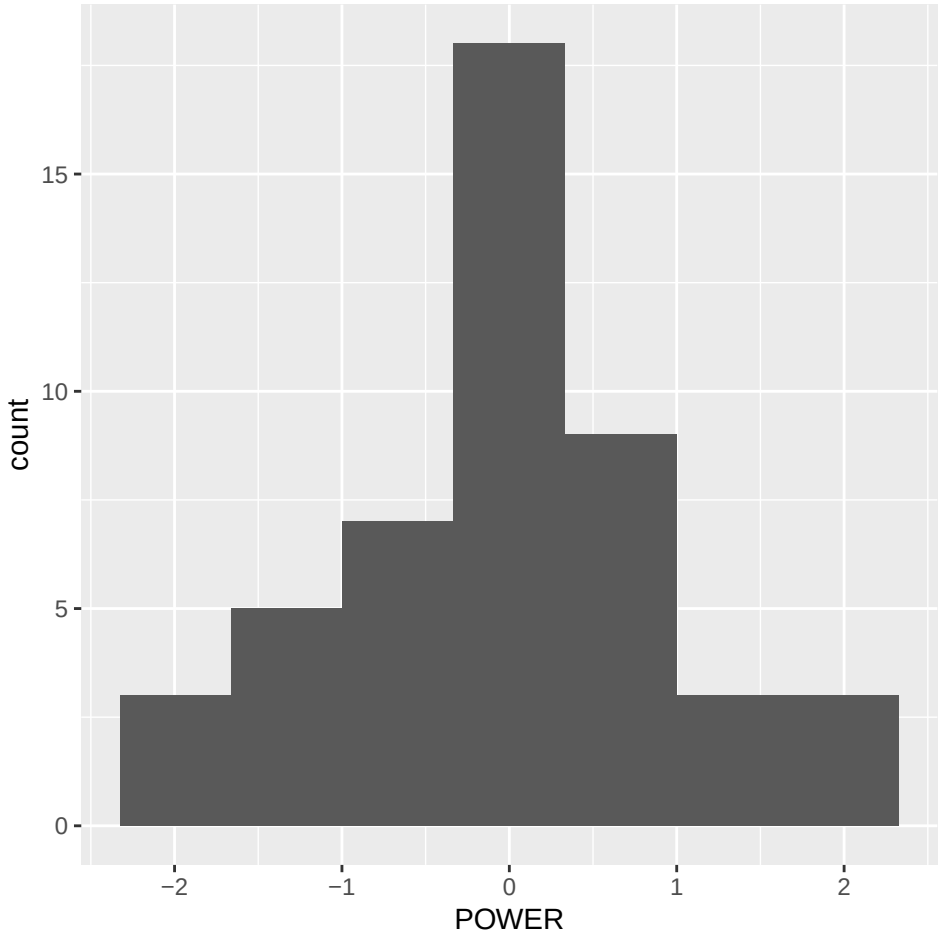
Unfortunately, however, I can't quite interpret what the three factors I found are. When I look at the factor loadings, specifically which variables load on which factor, they don't seem to make any sense.

Draw your best factor model (manually), take a photo of it and attach it in your report (or submit it as a separate graphics file, e.g., JPG or PNG).

Linear Regression

Practice linear regression e.g. with the Weight and Height variables and the factor scores.

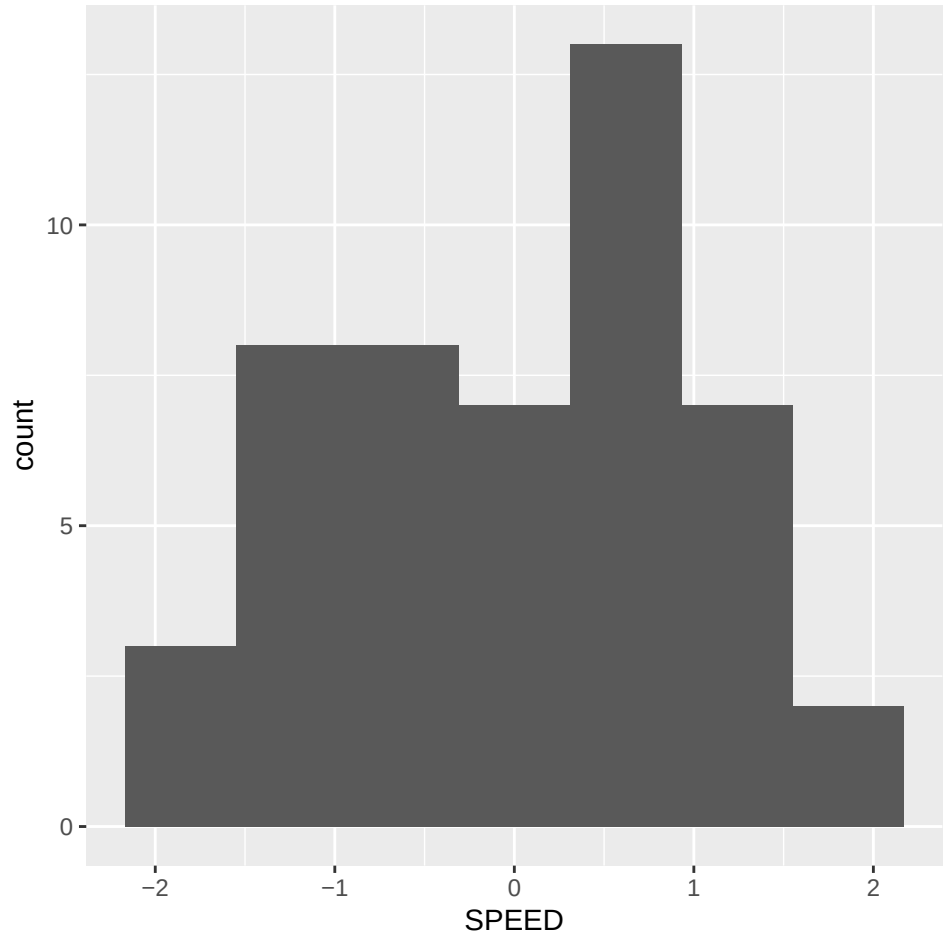
```
decaNEW2 %>%
  ggplot(aes(x = POWER)) +
  geom_histogram(bins = 7)
```




```
decaNEW2 %>%
  ggplot(aes(x = SPEED)) +
  geom_histogram(bins = 7)
```

Table 13: Some summary statistics

	Min.	1st Qu.	Median	3rd Qu.	Max.
POWER	-1.947191	-0.6012156	0.1073405	0.5216116	2.038157
SPEED	-1.953783	-0.8024772	0.0099311	0.8146515	1.762104



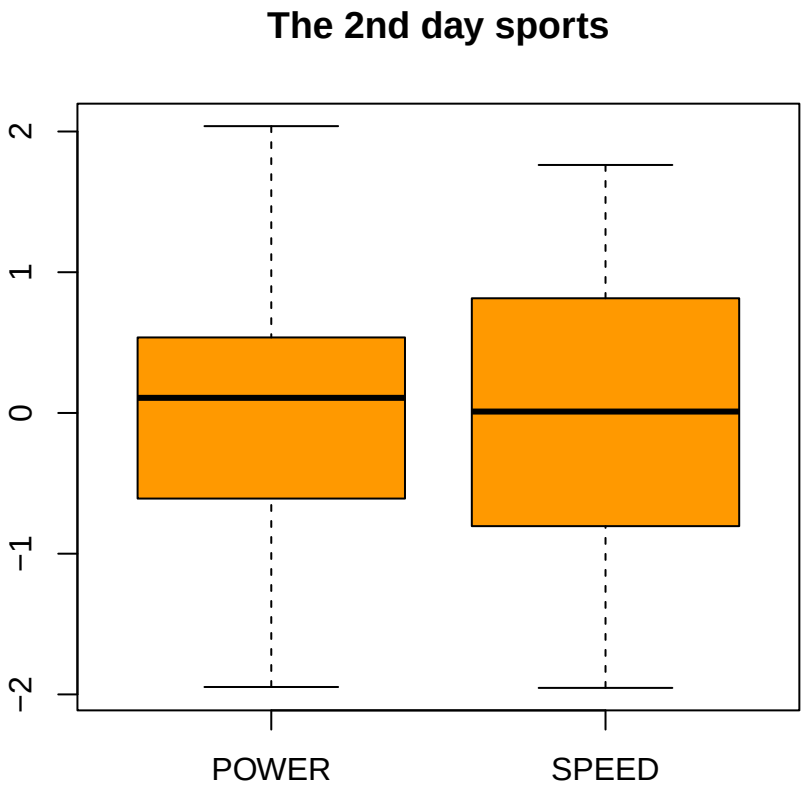
(These histograms are giving me a slight headache.)

```
# calculate some statistics
decaNEW2_des  <- describe(decaNEW2[, 15:16], ranges = FALSE, quant = c(0, 0.25, 0.5, 0.75, 1))
decaNEW2_des <- as.data.frame(decaNEW2_des)
names(decaNEW2_des) <- c("vars", "n", "mean", "sd", "skew", "kurtosis", "se",
                         "Min.", "1st Qu.", "Median", "3rd Qu.", "Max.")
# tables
# Summary
decaNEW2_des[8:12] %>%
  kbl(caption = "Some summary statistics") %>%
  kable_classic(full_width = T)
```

```
Boxplot(decaNEW2[, 15:16], col="#FF9900", main="The 2nd day sports")
```

Table 14: Some descriptive statistics

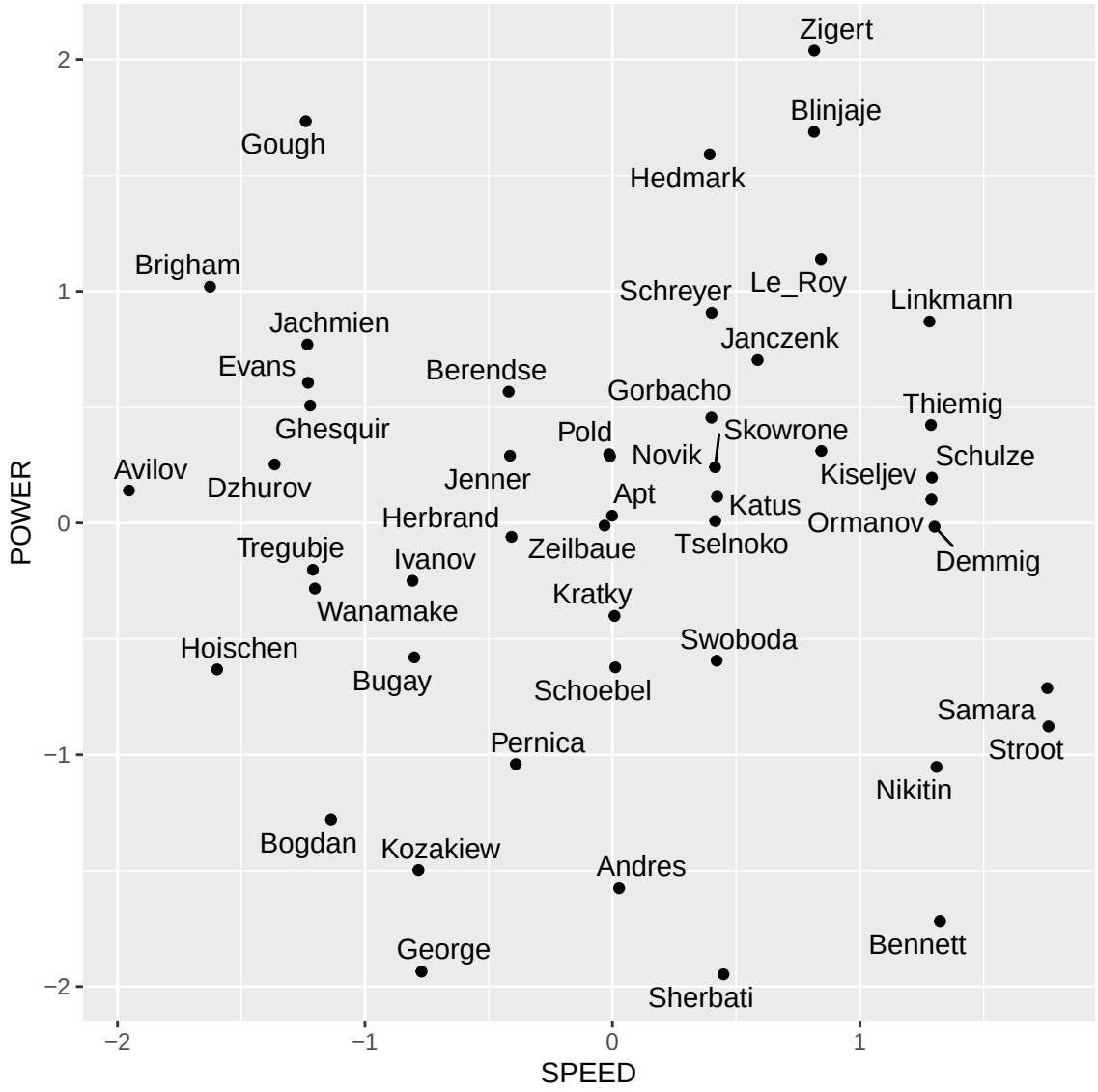
	n	mean	sd	skew	kurtosis	se
POWER	48	0	0.94	-0.13	-0.32	0.14
SPEED	48	0	1.00	-0.08	-1.11	0.14




```
# Descriptives
round(decaNEW2_des[,2:7], 2) %>%
  kbl(caption = "Some descriptive statistics") %>%
  kable_classic(full_width = T)
```

If you look more closely, the distributions of the variables are not as badly skewed as it seemed from the histograms. In particular, the distribution of SPEED is quite symmetrical, but also flat. However the power variable could be used as a tolerable response variable for the regression model. Perhaps. However, it would be good to look at the matter in other ways than just by calculating some statistics.

```
# Take care of the overlapping text labels properly:
# (the default geom_text() with check_overlap = TRUE is far from perfect)
library(ggrepel) # install.packages("ggrepel")
decaNEW2 %>%
  ggplot(aes(y = POWER, x = SPEED, label = Name)) +
  geom_point() + geom_text_repel()
```




```
round(cor(decaNEW2$POWER, decaNEW2$SPEED, method = "pearson"), 4)
```

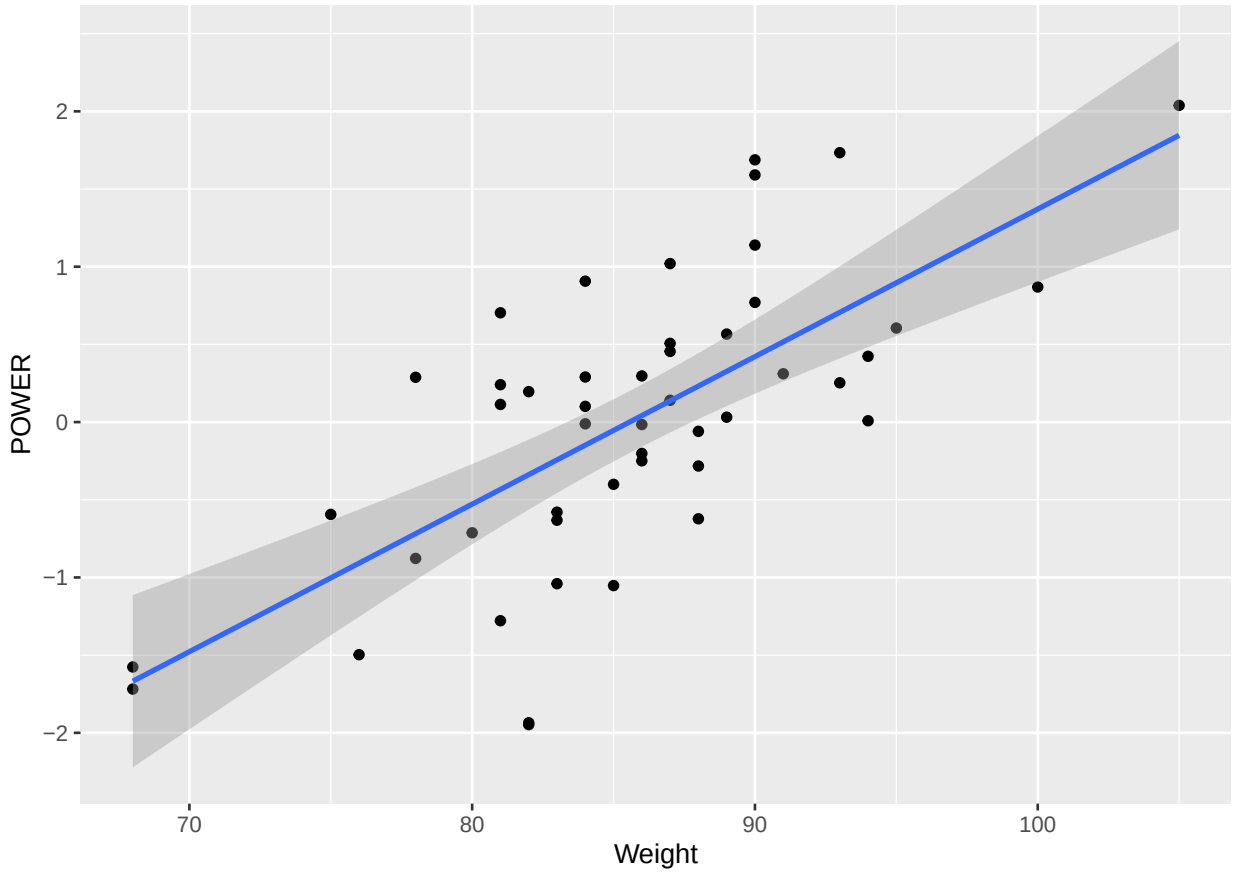
```
## [1] 0.0021
```

```
round(cor(decaNEW2$POWER, decaNEW2$SPEED, method = "spearman"), 4)
```

```
## [1] -0.0636
```

There is no correlation between the variables, non what so ever.

```
decaNEW2 %>%
  ggplot(aes(y = POWER, x = Weight)) +
  geom_point() + geom_smooth(method = "lm")
```




```
round(cor(decaNEW2$POWER, decaNEW2$Weight, method = "pearson"), 3)
```

```
## [1] 0.691
```

This already seems more promising, there is quite a strong correlation between the variables POWER and Weight.

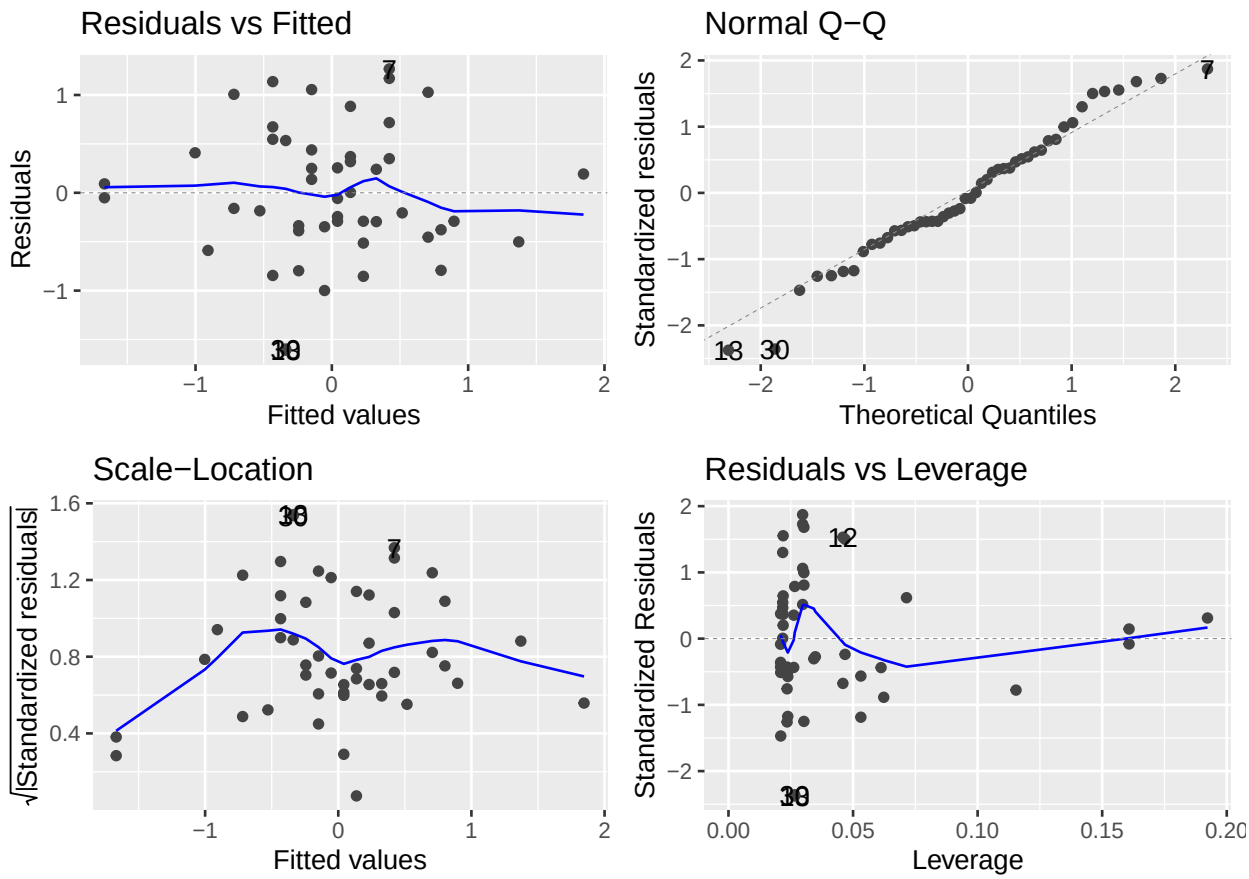
```
lmdeca <- lm(POWER ~ Weight, data = decaNEW2)
summary(lmdeca)
```

```
##
## Call:
## lm(formula = POWER ~ Weight, data = decaNEW2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.60888 -0.38052 -0.05411  0.41634  1.26650
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.12547    1.25568  -6.471 5.68e-08 ***
## Weight       0.09497    0.01463   6.491 5.30e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6868 on 46 degrees of freedom
## Multiple R-squared:  0.4781, Adjusted R-squared:  0.4667
## F-statistic: 42.14 on 1 and 46 DF,  p-value: 5.296e-08
```

From the values of the variable (table 7), it can be concluded that the unit of weight is the kilogram. So, when the weight increases by one kilogram, the POWER-score increases by one unit. However, according to this model, weight only explains about 47% of the variation of power.

```
library(ggfortify) # install.packages("ggfortify")
autoplot(lmdeca)
```



Based on the plots, it seems that the number 13 (maybe 12) and 30 are anomalous findings in the model and cause the most challenges in model diagnostics. The variation of the residual terms seems to be approximately the same in almost the entire range of the fitted values. The Normal Q-Q plot also looks reasonably good: it is quite common that there is a small variation at the edges of the data. On the Scale-Location plot a horizontal line would be nice. This is not horizontal. I don't like this, but I also can't say how disturbing this actually is. There are some outliers that cause leverage, but there are no outliers that exceed three standard deviations, which is good. Again, I don't think this plot is very wrong, but I'm not sure.

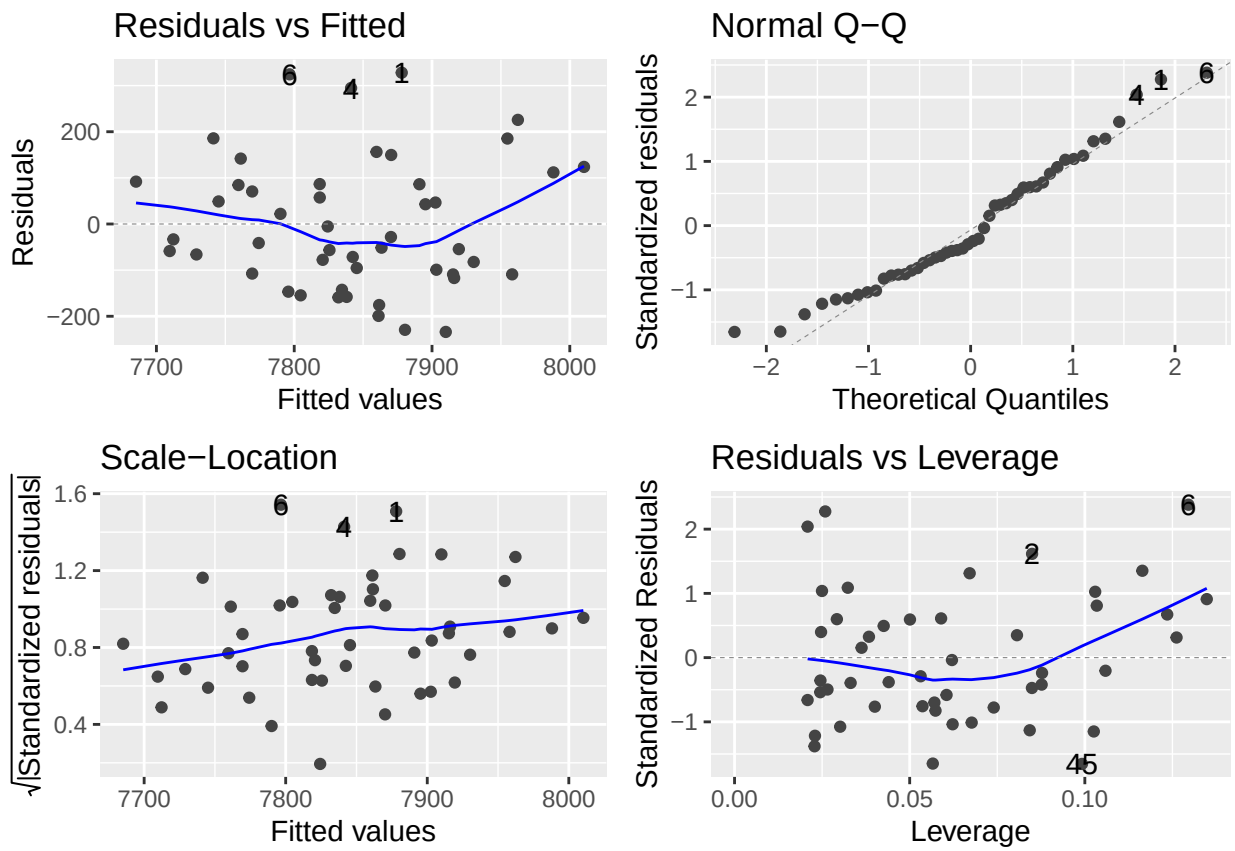
Continue with different regression models, for example:

```
lmdeca <- lm(Points ~ POWER + SPEED, data = decaNEW2)
summary(lmdeca)
```

```
##
## Call:
## lm(formula = Points ~ POWER + SPEED, data = decaNEW2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -234.00 -107.92  -37.28   87.81  327.97
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7843.48      21.08 372.062  < 2e-16 ***
## POWER          63.19      22.65   2.789  0.00771 **
## SPEED          46.64      21.36   2.184  0.03422 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 146.1 on 45 degrees of freedom
## Multiple R-squared:  0.2184, Adjusted R-squared:  0.1837
## F-statistic: 6.288 on 2 and 45 DF,  p-value: 0.003907
```

I believe POWER and SPEED score are variables of the same order of magnitude, so the coefficients of the regression model are comparable. If so, it would seem that POWER has more of an impact on overall decathlon points than speed. Unfortunately, this model is not very explanatory (adjusted R-squared is only 0.1837).

```
library(ggfortify)
autoplot(lmdeca)
```



There is more going on in the Residuals vs. Fitted -plot (than in the previous model), but Scale-Location and Residuals vs. Leverage look nicer. I like this model.

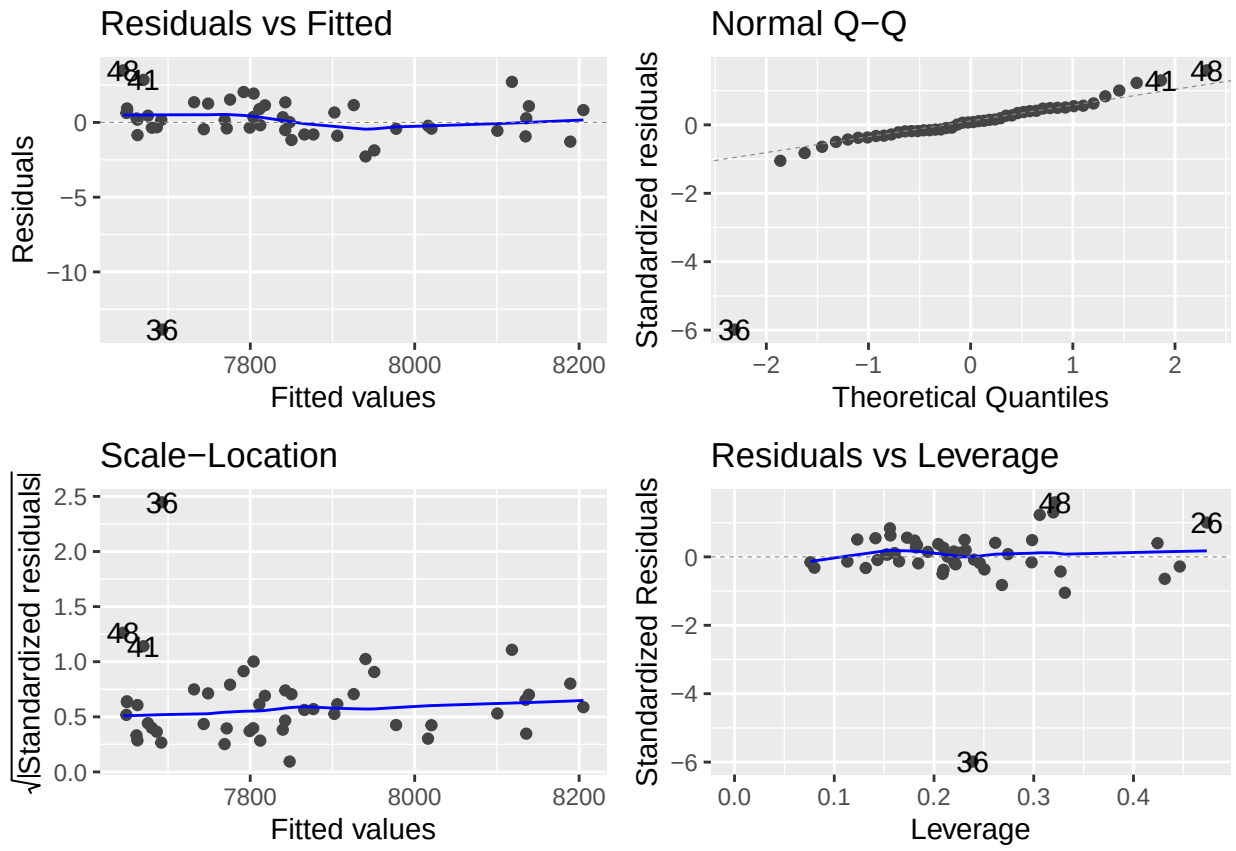
Additional experiment using the original Points and the 10 score variables:

```
# this should be a trivial model, as Points is the sum of the 10 score variables:
lmdeca <- lm(Points ~ Run100m + Longjump + Shotput + Hijump + Run400m +
               Hurdles + Discus + Polevlt + Javelin + Run1500m, data = decaNEW2)

# surprise! what is wrong here? can you spot the problem from the diagnostic plots?
# look at the residuals and their statistics, too!
summary(lmdeca)
```

```
##
## Call:
## lm(formula = Points ~ Run100m + Longjump + Shotput + Hijump +
##     Run400m + Hurdles + Discus + Polevlt + Javelin + Run1500m,
##     data = decaNEW2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8567  -0.4672   0.1791   0.9758   3.4881
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.600494  20.295237  -0.079    0.938
```

```
## Run100m      1.003235   0.009576 104.766   <2e-16 ***
## Longjump     1.002517   0.008462 118.468   <2e-16 ***
## Shotput      1.002573   0.009380 106.890   <2e-16 ***
## Hijump       0.996382   0.006963 143.090   <2e-16 ***
## Run400m      0.995210   0.010500  94.781   <2e-16 ***
## Hurdles      1.010072   0.008098 124.729   <2e-16 ***
## Discus       1.005394   0.010199  98.576   <2e-16 ***
## Polevlt      0.986195   0.006472 152.380   <2e-16 ***
## Javelin      1.000152   0.006603 151.462   <2e-16 ***
## Run1500m     1.002043   0.007527 133.122   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.654 on 37 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9997
## F-statistic: 1.743e+04 on 10 and 37 DF,  p-value: < 2.2e-16
```

```
library(ggfortify)
autoplot(lmdeca)
```



Okay, Mr. Kozakiew has done something very different than the other participants...

```
# fix the problem by computing the sum of the 10 score variables by yourself:
deca2 <- decaNEW2 %>%
  mutate(SUM = rowSums(across(Run100m:Run1500m)),
         diff = Points - SUM)
```

```
# check the difference (diff) between Points and SUM!
#View(deca2)
deca2[36, ]
```

```
##        Name Points Run100m Longjump Shotput Hijump Run400m Hurdles Discus
## 36 Kozakiew   7679     780      804     665    788     829     757    653
##    Polevlt Javelin Run1500m Height Weight     POWER      SPEED  SUM diff
## 36    1052     740      629    177     76 -1.497213 -0.7839209 7697  -18
```

... Or not. There seems to be a clerical error, the sum 7697 was typed wrong as 7679

```
# how about now?
lmdeca <- lm(SUM ~ Run100m + Longjump + Shotput + Hijump + Run400m +
               Hurdles + Discus + Polevlt + Javelin + Run1500m, data = deca2)
summary(lmdeca)
```

```
##
## Call:
## lm(formula = SUM ~ Run100m + Longjump + Shotput + Hijump + Run400m +
##     Hurdles + Discus + Polevlt + Javelin + Run1500m, data = deca2)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.238e-13 -7.170e-14 -1.293e-14  2.147e-14  1.200e-12
##
## Coefficients:
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept) -5.776e-12  1.707e-12 -3.384e+00   0.0017 **
## Run100m      1.000e+00  8.053e-16  1.242e+15   <2e-16 ***
## Longjump     1.000e+00  7.116e-16  1.405e+15   <2e-16 ***
## Shotput      1.000e+00  7.888e-16  1.268e+15   <2e-16 ***
## Hijump       1.000e+00  5.856e-16  1.708e+15   <2e-16 ***
## Run400m      1.000e+00  8.830e-16  1.133e+15   <2e-16 ***
## Hurdles      1.000e+00  6.810e-16  1.468e+15   <2e-16 ***
## Discus       1.000e+00  8.577e-16  1.166e+15   <2e-16 ***
## Polevlt      1.000e+00  5.443e-16  1.837e+15   <2e-16 ***
## Javelin      1.000e+00  5.553e-16  1.801e+15   <2e-16 ***
## Run1500m     1.000e+00  6.330e-16  1.580e+15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.232e-13 on 37 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 2.457e+30 on 10 and 37 DF,  p-value: < 2.2e-16
```

This looks nice. All slopes are one and standard errors are zero. Could it be that the Total Points are obtained by adding up the scores of the sports performances...?

Draw the regression model (manually), take a photo of it and attach it in your report (or submit it as a separate graphics file, e.g., JPG or PNG).

Bonus exercise

Check the lavaan tutorial page at https://lavaan.ugent.be/tutorial/cfa.html and estimate a CFA model of the physical fitness using the decathlon data. Use your best model as the basis. Compare the results between EFA and CFA.

Note: Use the option standardized = TRUE in the summary() function.

Define a CFA model

Hypothesis:

Physical fitness is a multidimensional construct composed of three factors:

- Power (POWER = Shotput + Discus + Polevlt)
- Speed (SPEED = Run100m + Run400m + Hijump + Javelin)
- Something else (OTHER = Run1500m + Hurdles + Longjump)
  (I would love to call this endurance, but it would not make any sense :-D)

```
library(lavaan)

mdl <- 'POWER =~ Shotput + Discus + Polevlt
        SPEED =~ Run100m + Run400m + Hijump + Javelin
        OTHER =~ Run1500m + Hurdles + Longjump'
```

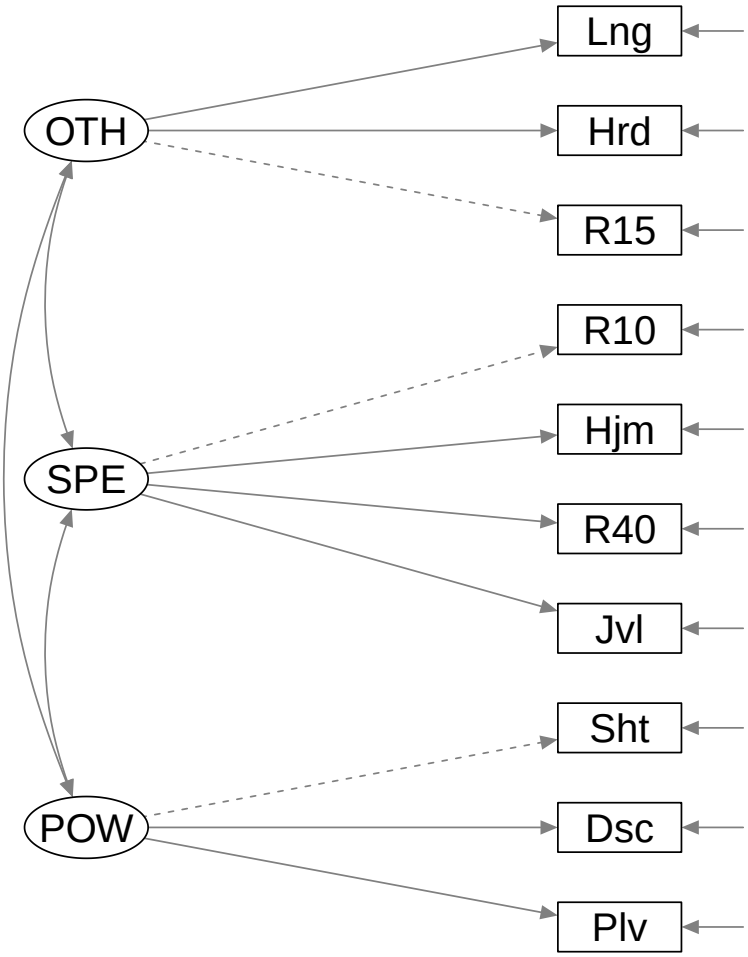
```
cfa <- cfa(mdl, data = deca[, 3:12])
cfa
```

Estimate the CFA model

```
## lavaan 0.6-11 ended normally after 224 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        23
##
##   Number of observations                            48
##
## Model Test User Model:
##
##   Test statistic                                35.214
##   Degrees of freedom                                32
##   P-value (Chi-square)                           0.319
```

Draw the graphs

```
library(semPlot)

semPaths(cfa,
    what = "path", whatLabels = "name", style = "lisrel", layout = "tree2",
    intercepts = FALSE, residuals = TRUE, thresholds = FALSE, reorder = FALSE,
    rotation = 2, sizeLat = 10, sizeLat2 = 5,
    sizeMan = 10, sizeMan2 = 4, manifests = rev(colnames(deca[, 3:12])),
    latents = c("POWER","SPEED", "OTHER"))
```



Numerical summary of the model

```
summary(cfa, fit.measures = TRUE, standardized = TRUE)
```

```
## lavaan 0.6-11 ended normally after 224 iterations
##
##   Estimator                                         ML
##   Optimization method                           NLMINB
##   Number of model parameters                        23
##
##   Number of observations                            48
##
## Model Test User Model:
##
##   Test statistic                                35.214
##   Degrees of freedom                                32
##   P-value (Chi-square)                           0.319
##
## Model Test Baseline Model:
##
##   Test statistic                               130.134
##   Degrees of freedom                                45
##   P-value                                        0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)                    0.962
##   Tucker-Lewis Index (TLI)                       0.947
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)              -2595.552
##   Loglikelihood unrestricted model (H1)      -2577.945
##
##   Akaike (AIC)                                5237.104
##   Bayesian (BIC)                              5280.142
##   Sample-size adjusted Bayesian (BIC)         5207.986
##
## Root Mean Square Error of Approximation:
##
##   RMSEA                                          0.046
##   90 Percent confidence interval - lower         0.000
##   90 Percent confidence interval - upper         0.119
##   P-value RMSEA <= 0.05                          0.495
##
## Standardized Root Mean Square Residual:
##
##   SRMR                                           0.143
##
## Parameter Estimates:
##
##   Standard errors                             Standard
##   Information                                 Expected
```

```
##   Information saturated (h1) model          Structured
##
## Latent Variables:
##                   Estimate   Std.Err  z-value  P(>|z|)   Std.lv   Std.all
##   POWER =~
##     Shotput            1.000                               47.267    0.767
##     Discus             1.257    0.257    4.902    0.000    59.438    0.956
##     Polevlt           -0.254    0.197   -1.292    0.196   -12.026   -0.193
##   SPEED =~
##     Run100m            1.000                              101.196    1.684
##     Run400m            0.140    0.153    0.919    0.358    14.194    0.289
##     Hijump            -0.130    0.150   -0.868    0.385   -13.171   -0.206
##     Javelin           -0.069    0.089   -0.770    0.441    -6.962   -0.110
##   OTHER =~
##     Run1500m           1.000                               88.630    1.152
##     Hurdles           -0.098    0.110   -0.895    0.371    -8.720   -0.162
##     Longjump          -0.096    0.105   -0.913    0.361    -8.518   -0.170
##
## Covariances:
##                   Estimate   Std.Err  z-value  P(>|z|)   Std.lv   Std.all
##   POWER ~~
##     SPEED            645.339  417.605    1.545    0.122     0.135    0.135
##     OTHER          -2219.859  748.134   -2.967    0.003    -0.530   -0.530
##   SPEED ~~
##     OTHER          -2142.771  717.584   -2.986    0.003    -0.239   -0.239
##
## Variances:
##                   Estimate   Std.Err  z-value  P(>|z|)   Std.lv   Std.all
##    .Shotput         1568.472  484.341    3.238    0.001  1568.472    0.412
##    .Discus           329.119  577.394    0.570    0.569   329.119    0.085
##    .Polevlt         3749.594  767.870    4.883    0.000  3749.594    0.963
##    .Run100m        -6628.217 9882.735   -0.671    0.502 -6628.217   -1.835
##    .Run400m         2216.447  483.640    4.583    0.000  2216.447    0.917
##    .Hijump          3929.545  805.298    4.880    0.000  3929.545    0.958
##    .Javelin         3951.737  801.659    4.929    0.000  3951.737    0.988
##    .Run1500m       -1935.624 5593.484   -0.346    0.729 -1935.624   -0.327
##    .Hurdles         2804.263  572.676    4.897    0.000  2804.263    0.974
##    .Longjump        2450.433  500.846    4.893    0.000  2450.433    0.971
##     POWER           2234.141  795.029    2.810    0.005     1.000    1.000
##     SPEED          10240.651 9723.739    1.053    0.292     1.000    1.000
##     OTHER           7855.328 5695.170    1.379    0.168     1.000    1.000
```

Model Test User Model:

Test statistic 35.214

Degrees of freedom 32

p-value (Chi-square) 0.319

In the statistical test of the factor analysis model, the null hypothesis is 'the model fits the data. In the three-factor model, the value of the chi-square test variable is 35,214 with 32 degrees of freedom, so the p-value of the test is 0,319. The null hypothesis should not be rejected, indicating the model does fit the data.

CFI and TLI

The comparative Fit Index (CFI) of the model is 0.962, bigger than 0.95. Also, Tucker-Lewis Index (TLI) of the model is 0.947, close to one (I believe, is it close enough?). These both indicate a well-fitting model.

RMSEA

Finally, RMSEA is 0.046, less than 0.05. This indicates good precision.

**** THE END ****