Compulsory Exercise 3 in TMA4267 Statistical Linear Models, Spring 2021

Sander Ruud, Alexander J Ohrt

18 april, 2021

Design of Experiments (DOE)

The purpose of this exercise is to provide insight and training in planning, performing and analyzing a statistical experiment, as well as to report the results.

High-score in Tetris

A k-factor two-level experiment is performed in order to determine how various factors influence score in the game of *Tetris*. The aim of the game is to attain the highest score possible. This is interesting, because Tetris is a nostalgic game and it is fun to infer some factors that could boost our performance. Our prior knowledge when it comes to factors that influence performance in the game is that concentration is key, as in many other situations in life. Besides this, in-game performance has never been investigated systematically by us, which makes this experiment more interesting, at least for the authors of the study. Hopefully, we can achieve higher scores and better performances in Tetris, after this statistical experiment.

Selection of Factors and Levels

Some factors that are believed to be relevant to the problem are

colnames(ExperimentDesign) <- c("Bright", "Coffee", "Music", "Seating", "Surroundings", "Device")

- A - Screen brightness: 10% or 100%.
- B - Coffee break: Is the game played while drinking coffee at least once during the game?
- C - Music: Is the game played with or without listening to Darude Sandstorm?
- D - Seating: Is the game played while laying in bed or while sitting comfortably at a desk?
- E - Surroundings: Is the game played in a quiet room or in front of the TV?
- F - Device: Is the game played on a mobile phone or on a laptop with a larger screen?

We expect an interaction between seating and device, since the mobile phone might be a lot easier to handle while laying in bed compared to the laptop, which might give more confidence when seated at a desk. Moreover, an interaction between low screen brightness and surroundings could make it even more difficult to concentrate, because it might be even harder to block out external disturbances when the screen brightness is low compared to when it is high. Also, an interaction between screen brightness and device might be reasonable, since a larger screen might not demand the same amount of brightness for reasonable legibility, compared to the smaller screen. More interactions might be present, but those are the ones I thought of off the top of my head.

The levels that are used in the experiments are briefly described in the list above. These are reasonable since we expect them to lead to different results in the response and they are easy to implement and monitor in practice. With the goal of the statistical experiment in mind, these are especially relevant, since all these factors are very easy to control when a subject wants to play the game. This means that should some of the levels in the factors be inferred as favourable to gaming performance, it is easy for a player to adjust to these in everyday life.

Selection of Response Variable

The response variable is the score in the game after 30 seconds. Each game is played by the same subject, within a time span of approximately one hour. Several response variables might be interesting however, e.g. an average score over several games with the same levels of the factors. These types of responses are very easy to measure, since Tetris simply outputs the score after each completed game. However, we have chosen to only register one score per combination of levels of factors, since it makes it possible to experiment with a larger amount of factors with the same amount of runs.

When it comes to the accuracy of these responses, it is reasonable to assume that the algorithm for calculating the score is the same in each game, which means that they should be reliable. Assuming that the reader is familiar with Tetris, since the pieces received are random each time, we will expect a random element, i.e. there will be a lot of variance.

Choice of Design

This experiment will follow a 2^6 full factorial design. In this way, all the factors presented may be tested in this experiment. Since the experiment has 2^6 runs, replications will not be performed, in order to save time and effort. However, replicates might be desirable for results with less variability and may be performed at a later time.

Blocking is not necessary, since the experiment is performed by one person, in a relatively short time frame of approximately one hour.

Desired resolution? Snakker man om resolution i et fullfaktorielt design, slik vi har gjort her?

Implementation of the Experiment

We create a game mode that lasts for 30 seconds. For each combination of levels of the factors, one single game is played, and the game outputs a score. These scores are registered and added to the `results`-vector below.

Note that the results are presented in an orderly fashion below, since they were performed in an orderly fashion. This is a problem with the implementation of this experiment, since randomization is not used. This could have been fixed simply by sampling rows randomly via R from the dataframe `ExperimentDesign`, which is shown in the following. However, this would make it harder to log which combinations of levels have been performed at each time during the experiment, which leads to larger time consumption.

Since the experiment is very simple to perform, each experiment is a genuine run replicate, which means that it reflects the total variability of the experiment. All the same necessary steps are performed for each of the 2^6 runs, i.e. ... Stemmer dette? Kan man f.eks si at det ikke er genuine run replicate, fordi du er mer sliten mot slutten av alle forsøkene enn mot starten? Burde man f.eks tatt 5 min pause etter hvert spill, for å endre på dette? Kanskje du har noen tanker og vil fullføre dette? Nå prøver jeg bare å nevne noe for alle keywordsene på BB, slik at vi kommer inn på alt der ;) Så regner jeg med at det blir godkjent.

Analysis of Data

```
# importing packages
# require(FrF2)
```

```
results <- c(7863,6977,6597,8982,5056,4352,7814,5472,6450,4031,4915,5365,3213,4540,7917,
             4842,6968,6717,7192,6767,5821,6387,064,4742,5246,3150,5463,7588,6430,3573,
             5867,4608,1564,728,1574,362,1673,3063,2816,1500,2845,496,2184,911,1241,
             3107,1657,1803,2552,2358,2218,2527,1460,403,1714,3391,2006,1789,4118,
```

```r
             577,2018,658,4528,662)

results
```

```
#>  [1] 7863 6977 6597 8982 5056 4352 7814 5472 6450 4031 4915 5365 3213 4540 7917
#> [16] 4842 6968 6717 7192 6767 5821 6387   64 4742 5246 3150 5463 7588 6430 3573
#> [31] 5867 4608 1564  728 1574  362 1673 3063 2816 1500 2845  496 2184  911 1241
#> [46] 3107 1657 1803 2552 2358 2218 2527 1460  403 1714 3391 2006 1789 4118  577
#> [61] 2018  658 4528  662
```

```r
length(results)
```

```
#> [1] 64
```

```r
?gl

ExperimentDesign <- expand.grid(A = gl(2, 1, labels = c("+", "-")),
                           B = gl(2, 1, labels = c("+", "-")),
                           C = gl(2, 1, labels = c("+", "-")),
                           D = gl(2, 1, labels = c("+", "-")),
                           E = gl(2, 1, labels = c("+", "-")),
                           FF = gl(2, 1, labels = c("+", "-"))) # FALSE IS RESERVED


# colnames(ExperimentDesign) <- c("Bright", "Coffee", "Music", "Seating", "Surroundings", "Device")


#View(ExperimentDesign)
ExperimentDesign
```

```
#>    A B C D E FF
#> 1  + + + + +  +
#> 2  - + + + +  +
#> 3  + - + + +  +
#> 4  - - + + +  +
#> 5  + + - + +  +
#> 6  - + - + +  +
#> 7  + - - + +  +
#> 8  - - - + +  +
#> 9  + + + - +  +
#> 10 - + + - +  +
#> 11 + - + - +  +
#> 12 - - + - +  +
#> 13 + + - - +  +
#> 14 - + - - +  +
#> 15 + - - - +  +
#> 16 - - - - +  +
#> 17 + + + + -  +
#> 18 - + + + -  +
#> 19 + - + + -  +
#> 20 - - + + -  +
#> 21 + + - + -  +
#> 22 - + - + -  +
#> 23 + - - + -  +
#> 24 - - - + -  +
```

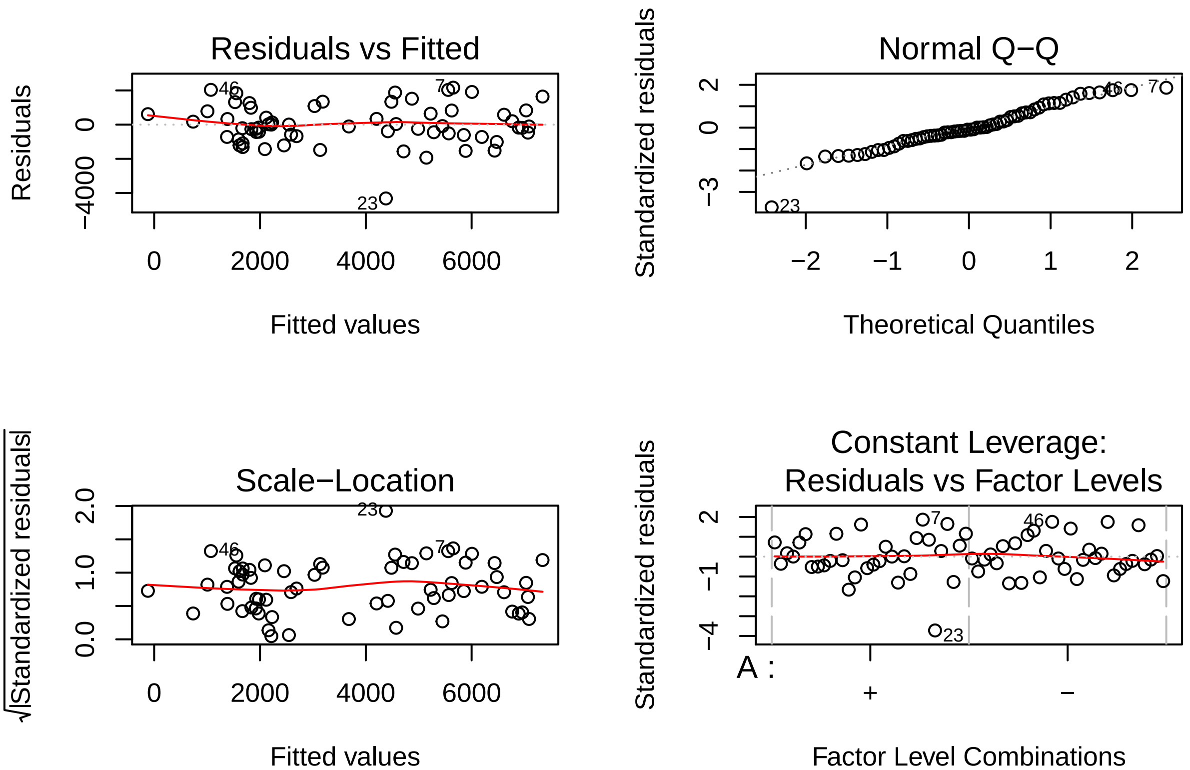
```
#> 25 + + + - -  +
#> 26 - + + - -  +
#> 27 + - + - -  +
#> 28 - - + - -  +
#> 29 + + - - -  +
#> 30 - + - - -  +
#> 31 + - - - -  +
#> 32 - - - - -  +
#> 33 + + + + +  -
#> 34 - + + + +  -
#> 35 + - + + +  -
#> 36 - - + + +  -
#> 37 + + - + +  -
#> 38 - + - + +  -
#> 39 + - - + +  -
#> 40 - - - + +  -
#> 41 + + + - +  -
#> 42 - + + - +  -
#> 43 + - + - +  -
#> 44 - - + - +  -
#> 45 + + - - +  -
#> 46 - + - - +  -
#> 47 + - - - +  -
#> 48 - - - - +  -
#> 49 + + + + -  -
#> 50 - + + + -  -
#> 51 + - + + -  -
#> 52 - - + + -  -
#> 53 + + - + -  -
#> 54 - + - + -  -
#> 55 + - - + -  -
#> 56 - - - + -  -
#> 57 + + + - -  -
#> 58 - + + - -  -
#> 59 + - + - -  -
#> 60 - - + - -  -
#> 61 + + - - -  -
#> 62 - + - - -  -
#> 63 + - - - -  -
#> 64 - - - - -  -
```

```
#print(FrF2(nruns=2^4,nfactors=4, randomize=FALSE))
# må sette default.levels = c(1, -1)

#plan = FrF2(nruns = 2^6, nfactors=6, randomize=FALSE)
#plan <- add.response(plan, results)

attach(ExperimentDesign)
fit <- lm(results~(A+B+C+D+E+FF)^2, data=ExperimentDesign)
summary(fit)
```

```
#>
#> Call:
#> lm(formula = results ~ (A + B + C + D + E + FF)^2, data = ExperimentDesign)
#>
```

```
#> Residuals:
#>     Min      1Q  Median      3Q     Max 
#> -4315.7  -623.8  -107.6   792.6  2161.9 
#> 
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)    
#> (Intercept)  7030.25     840.05   8.369 1.72e-10 ***
#> A-             54.75     877.40   0.062   0.9505    
#> B-             34.44     877.40   0.039   0.9689    
#> C-          -1462.75     877.40  -1.667   0.1029    
#> D-          -1406.62     877.40  -1.603   0.1164    
#> E-           -262.75     877.40  -0.299   0.7661    
#> FF-         -5048.81     877.40  -5.754 8.96e-07 ***
#> A-:B-         221.12     716.40   0.309   0.7591    
#> A-:C-         265.25     716.40   0.370   0.7131    
#> A-:D-       -1261.25     716.40  -1.761   0.0856 .  
#> A-:E-          67.50     716.40   0.094   0.9254    
#> A-:FF-       -440.63     716.40  -0.615   0.5418    
#> B-:C-          50.13     716.40   0.070   0.9446    
#> B-:D-         776.38     716.40   1.084   0.2847    
#> B-:E-        -188.87     716.40  -0.264   0.7933    
#> B-:FF-       -177.50     716.40  -0.248   0.8055    
#> C-:D-         984.25     716.40   1.374   0.1768    
#> C-:E-        -820.75     716.40  -1.146   0.2584    
#> C-:FF-       1403.63     716.40   1.959   0.0567 .  
#> D-:E-         492.25     716.40   0.687   0.4958    
#> D-:FF-        954.38     716.40   1.332   0.1900    
#> E-:FF-        828.63     716.40   1.157   0.2540    
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#> 
#> Residual standard error: 1433 on 42 degrees of freedom
#> Multiple R-squared:  0.7599, Adjusted R-squared:  0.6399 
#> F-statistic:  6.33 on 21 and 42 DF,  p-value: 2.101e-07
```

```r
# Check of model assumptions.
par(mfrow=c(2,2))
plot(fit) # Finnes vel nicere måte å plotte på sikkert, men har det ikke i hodet akk nå.
```



From the residual plot in the top left, the residuals do not seem to break the assumption of homoscedasticity, since no clear pattern is observed. Moreover, the Q-Q-plot does not signal that the model assumptions are not fulfilled either.

Conclusion and Recommendations

In the full factorial model, the only two significant factors are the intercept, and the device. When playing on mobile, the expected score decreases with about 5000 points, with a p-value of $\approx 9e-7$, which is not insignificant.