# Auswertung des Helikopterexperiments

Danuscha Große-Hering

05.Juli 2021

```
#setwd("C:\\Users\\danus\\OneDrive\\Uni\\4.Semester\\Grundlagen der Versuchspalnung\\1.Experiment\\GdV-
```

## Versuchsablauf

## Screening

```
library(SixSigma)


ExperimentDesign  = expand.grid(A = c(-1, 1), B = c(-1,1), C = c(-1, 1), D = c(-1,1), E = c(-1,1) )
"F" = ExperimentDesign$A * ExperimentDesign$C * ExperimentDesign$D
G = ExperimentDesign$A * ExperimentDesign$B * ExperimentDesign$C




ExperimentDesign$F = F
ExperimentDesign$G = G
```

```
Screening <- read.csv("Screening.CSV", sep = ";",dec = ",")
S <- Screening[order(Screening[,2]),]
S <-cbind(S, ExperimentDesign)
```

```
summary(lm(Zeit.ohne.Klammer ~A+B+C+D+E+F+G, data= S))
```

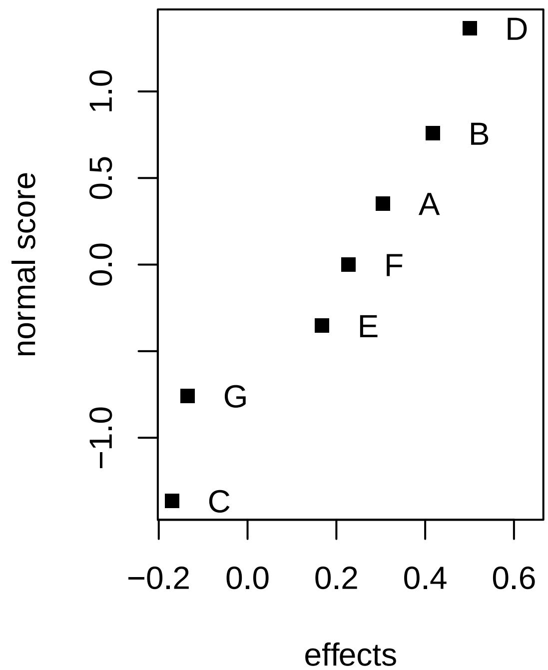
```
##
## Call:
## lm(formula = Zeit.ohne.Klammer ~ A + B + C + D + E + F + G, data = S)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.5694 -0.4650  0.1294  0.3719  2.1131
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.13563    0.13399  30.866   <2e-16 ***
## A            0.15250    0.13399   1.138   0.2663
## B            0.20875    0.13399   1.558   0.1323
## C           -0.08500    0.13399  -0.634   0.5318
## D            0.25000    0.13399   1.866   0.0743 .
## E            0.08375    0.13399   0.625   0.5378
## F            0.11375    0.13399   0.849   0.4043
```

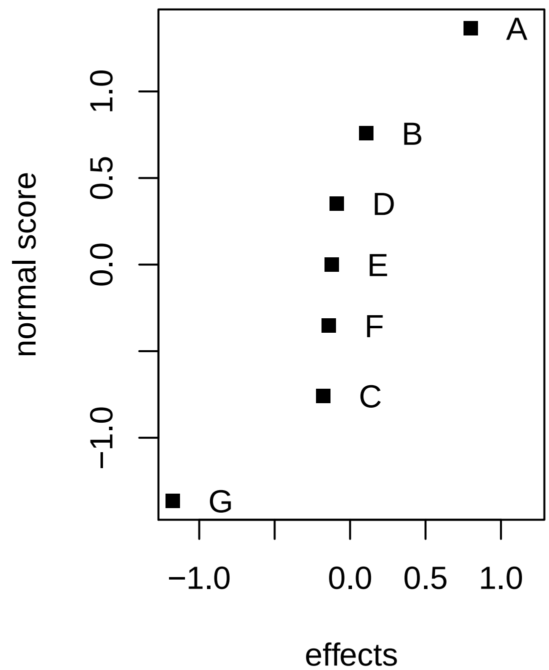
```
## G           -0.06750    0.13399  -0.504   0.6190
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7579 on 24 degrees of freedom
## Multiple R-squared:  0.2721, Adjusted R-squared:  0.0598
## F-statistic: 1.282 on 7 and 24 DF,  p-value: 0.3009
```

```
summary(lm(Zeit.mit.Klammer ~A+B+C+D+E+F+G, data= S))
```

```
##
## Call:
## lm(formula = Zeit.mit.Klammer ~ A + B + C + D + E + F + G, data = S)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2962 -0.5911 -0.2037  0.4916  1.4875
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.70781    0.15986  23.194  < 2e-16 ***
## A            0.39906    0.15986   2.496  0.01982 *
## B            0.05281    0.15986   0.330  0.74399
## C           -0.08906    0.15986  -0.557  0.58260
## D           -0.04469    0.15986  -0.280  0.78223
## E           -0.06156    0.15986  -0.385  0.70356
## F           -0.07031    0.15986  -0.440  0.66399
## G           -0.58781    0.15986  -3.677  0.00119 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9043 on 24 degrees of freedom
## Multiple R-squared:  0.4618, Adjusted R-squared:  0.3048
## F-statistic: 2.942 on 7 and 24 DF,  p-value: 0.02257
```

```
#install.packages("BsMD")
library(BsMD)
par(mfrow = c(1,2))
DanielPlot(lm(Zeit.ohne.Klammer ~ A +B + C + D+E+F+G, data = S), pch = 15)
DanielPlot(lm(Zeit.mit.Klammer ~A + B + C + D+E+F+G, data= S), pch = 15)
```




## Optimierung

```
Optimierung <-read.csv("Optimierung.CSV", sep = ";",dec = ",")

#erster Durchlauf
set.seed(1735)
p1 <-expand.grid(A = c(-1,0, 1),  G = c(-1,0,1) )
p1 <- cbind(1:9,p1)
#zweiter Durchlauf
set.seed(1736)
s <-sample(1:9,9)
p2 <-p1[s,]


p <- rbind(p1,p2)
names(p)[1] <- "Nr."
a <-round(sqrt(0.5*(sqrt(9*18)-9)), digits=3)
p[which(p$A*p$G ==0),2:3] <- p[which(p$A*p$G ==0),2:3] *a

o1 <-cbind(Optimierung,p[1:9,2:3])

o2 <- o1[,-3]
o3 <- o1[,-2]
```

```r
names(o2)[2] <- "Zeit"
names(o3)[2] <- "Zeit"

o <- rbind(o2,o3)
o$Asquare <- o$A^2
o$Gsquare <- o$G^2

lm(Zeit ~A + Asquare+G+Gsquare, data = o)
```

```
##
## Call:
## lm(formula = Zeit ~ A + Asquare + G + Gsquare, data = o)
##
## Coefficients:
## (Intercept)            A      Asquare            G      Gsquare
##    3.620074     0.375286    -0.009844    -0.584445     0.114269
```

```r
##MIT Wechselwirkung
MM = lm(Zeit ~A * G  + Asquare+Gsquare, data = o)
thetaM <- MM$coefficients
modellM <- function(t){
  thetaM[1] + thetaM[2] * t[1] + thetaM[4] * t[1]^2 + thetaM[3] * t[2] +
    thetaM[5] * t[2]^2 + thetaM[6] * t[1] * t[2]
}

MIT = optim(c(1,1),  function(x) -modellM(x), method = "L-BFGS-B",
             lower = c(-1.365, -1.365), upper = c(1.365,1.365))

modellM(c(1.365, -1.365))
```

```
## (Intercept)
##    5.678983
```

```r
MIT
```

```
## $par
## [1]  1.365 -1.365
##
## $value
## [1] -5.678983
##
## $counts
## function gradient
##        5        5
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL"
```

```r
## OHNE

M = lm(Zeit ~A + G  + Asquare+Gsquare, data = o)
theta <- M$coefficients
modell <- function(t){
```

```r
  theta[1] + theta[2] * t[1] + theta[4] * t[1]^2 + theta[3] * t[2] +
    theta[5] * t[2]^2 #+ theta[6] * t[1] * t[2]
}

OHNE = optim(c(0.5,0.5),  function(x) -modell(x), method = "L-BFGS-B",
      lower = c(-1.365, -1.365), upper = c(1.365,1.365))
OHNE
```

```
## $par
## [1]  1.365 -1.365
##
## $value
## [1] -5.124674
##
## $counts
## function gradient
##        5        5
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: NORM OF PROJECTED GRADIENT <= PGTOL"
```

$\Rightarrow f(x_1, x_2) = 3.47 + 00.44 \cdot x_1 + 0.06 \cdot x_1^2 - 0.67 \cdot x_2 + 0.29 \cdot x_2^2 \Rightarrow \frac{\partial f}{\partial x} = \begin{pmatrix} 0.44 + 0.12 \cdot x_1 \\ -0.67 + 0.58 \cdot x_2 \end{pmatrix}$

```r
y_observ=c(1.2,0.5,1.5,1.3,0.2,1.4)
data_set=data.frame(y=y_observ)
data_set$A = c(-1,0,1,-1,0,1)
data_set$Asquare = data_set$A^2
lm(y~A+Asquare, data=data_set)
```

```
##
## Call:
## lm(formula = y ~ A + Asquare, data = data_set)
##
## Coefficients:
## (Intercept)            A      Asquare
##        0.35         0.10         1.00
```