

Bilag 5

R-Koder til brug for benchmarking af spildevandsselskaber

Juli 2020
version 2



Bilag 5 – R-Koder til brug for benchmarking af spildevandsselskaber – version 2

Konkurrence- og Forbrugerstyrelsen
Forsyningssekretariatet
Carl Jacobsens Vej 35
2500 Valby
Tlf.: +45 41 71 50 00
E-mail: kfst@kfst.dk

Bilag 5 til Totaløkonomisk benchmarking for spildevandsselskaber – R-Koder til brug for benchmarking af spildevandsselskaber – version 2 er udarbejdet af Forsyningssekretariatet.

Juli 2020

Indhold

Kapitel 1	4
Korrigerede netvolumenmål	4
1.1 R-koder til brug for beregningen de to korrektioner af netvolumenmålene	4
1.2 Alderskorrigeret netvolumenmål	4
1.3 Tæthedskorrigeret netvolumenmål	5
Kapitel 2	7
Benchmarking	7
2.1 R-koder til brug for DEA og SFA modellen	7
2.2 DEA	7
2.3 SFA	8
2.4 SFA-modeller for outliers	11
2.5 Samlet oversigt over scorer	13
Kapitel 3	14
Costdriversammensætning	14
3.1 R-koder til brug for analyse af costdriversammensætning	14
3.2 Kode for costdriversammensætning	14

Kapitel 1

Korrigerede netvolumenmål

1.1 R-koder til brug for beregningen af de to korrektioner af netvolumenmålene

Dette kapitel viser koderne, som bliver brugt til at lave de korrigerede netvolumenmål. Koderne kan kopieres direkte ind i softwareprogrammet R. Koderne laver regressionerne, der ligger til grund for beregningen af de alderskorrigerede og tæthedskorrigerede netvolumenmål. I *Bilag 2 – Beregning af de korrigerede netvolumenmål* fremgår resultaterne af regressionerne sammen med en forklaring på metoden.

Beskrivelser af variable findes i *Bilag 7 – Data til R-koder v2*.

1.2 Alderskorrigeret netvolumenmål

```
## Indlæsning af datasæt
library(readxl)
Data <- read_excel("Bilag 7 - Data til R-koder v2.xlsx")

## I koden bruges følgende forkortelser for variable
# OPEX_Pr_Net = "Udregnes som FADO over OPEX Netvolumen"
# CAPEX_Pr_Net = "Udregnes som anlægsomkostninger over CAPEX
# Netvolumen"

#####
#### Udregning af OPEX_Pr_Net og CAPEX_Pr_net for 2017 og 2018 ####
#####

OPEX_Pr_Net_2017<-Data$FADO_2017/Data$OPEX_2017
OPEX_Pr_Net_2018<-Data$FADO_2018/Data$OPEX_2018
CAPEX_Pr_Net_2017<-Data$Anlægsomkostninger_2017/Data$CAPEX_2017
CAPEX_Pr_Net_2018<-Data$Anlægsomkostninger_2018/Data$CAPEX_2018

#####
#### Alderskorrigeret OPEX netvolmuenmål for 2017 og 2018 ####
#####

OPEX_alder_2017 <- lm(OPEX_Pr_Net_2017 ~ Data$Alder_2017)
OPEX_alder_2018 <- lm(OPEX_Pr_Net_2018 ~ Data$Alder_2018)

# Finder mulige outliers med Cook' Distance
i<- max(cooks.distance(OPEX_alder_2017))
j<- max(cooks.distance(OPEX_alder_2018))

# Her laves en iterativ outlier analyse, hvor de selskaber
# med en høj cooks distance værdi analyseres for at
# vurdere om det er en outlier
```

```

#Der er ingen outliers

summary(OPEX_alder_2017)
summary(OPEX_alder_2018)

#####
#### Alderskorrigeret CAPEX netvolumenmål for 2017 og 2018 ####
#####

CAPEX_alder_2017 <- lm(CAPEX_Pr_Net_2017 ~ Data$Alder_2017)
CAPEX_alder_2018 <- lm(CAPEX_Pr_Net_2018 ~ Data$Alder_2018)

# Finder mulige outliers med Cook' Distance
k <- max(cooks.distance(CAPEX_alder_2017))
l <- max(cooks.distance(CAPEX_alder_2018))

# Her laves en iterativ outlier analyse, hvor de selskaber
# med en høj cooks distance værdi analyseres for at
# vurdere om det er en outlier

# Der er ingen outliers

summary(CAPEX_alder_2017)
summary(CAPEX_alder_2018)

```

1.3 Tæthedskorrigeret netvolumenmål

```

# Indlæsning af datasæt
library(readxl)
Data <- read_excel("Bilag 7 - Data til R-koder v2.xlsx")
Data_OPEX <- Data
Data_CAPEX <- Data

# I koden bruges følgende forkortelser
#OPEX_Pr_Net = "Udregnes som FADO over OPEX Netvolumen"
#CAPEX_Pr_Net = "Udregnes som anlægsomkostninger over CAPEX
#Netvolumen"

#####
#### Udregning af OPEX_Pr_Net og CAPEX_Pr_net for 2017 og 2018 ####
#####

Data_OPEX$OPEX_Pr_Net_2017<-Data_OPEX$FADO_2017/Data_OPEX$OPEX_2017
Data_OPEX$OPEX_Pr_Net_2018<-Data_OPEX$FADO_2018/Data_OPEX$OPEX_2018

Data_CAPEX$CAPEX_Pr_Net_2017<-
Data_CAPEX$Anlægsomkostninger_2017/Data_CAPEX$CAPEX_2017

Data_CAPEX$CAPEX_Pr_Net_2018<-
Data_CAPEX$Anlægsomkostninger_2018/Data_CAPEX$CAPEX_2018

#####
#### Tæthedskorrigeret OPEX netvolumenmål ####
#####

# Vi fjerner alle selskaber der kun har renseaktivitet

```

```
Data_OPEX <- Data_OPEX[-c(which(Data_OPEX$Type == "r")),]

OPEX_Tæt_2017 <- lm(Data_OPEX$OPEX_Pr_Net_2017 ~
Data_OPEX$Tæt_2017)
OPEX_Tæt_2018 <- lm(Data_OPEX$OPEX_Pr_Net_2018 ~
Data_OPEX$Tæt_2018)

#Finder mulige outliers med Cook's Distance
m <- max(cooks.distance(OPEX_Tæt_2017))
n <- max(cooks.distance(OPEX_Tæt_2018))

# Her laves en itterativ outlier analyse, hvor de selskaber
# med en høj cooks distance værdi analyseres for at
# vurdere om det er en outlier

# Der er ingen outliers

summary(OPEX_Tæt_2017)
summary(OPEX_Tæt_2018)

#=====#
### Tæthedskorrigeret CAPEX netvolumenmål      ###
#=====#

# Vi fjerner alle selskaber der kun har renseaktivitet
Data_CAPEX <- Data_CAPEX[-c(which(Data_CAPEX$Type == "r")),]

CAPEX_Tæt_2017 <- lm(Data_CAPEX$CAPEX_Pr_Net_2017 ~
Data_CAPEX$Tæt_2017)
CAPEX_Tæt_2018 <- lm(Data_CAPEX$CAPEX_Pr_Net_2018 ~
Data_CAPEX$Tæt_2018)

# Finder mulige outliers med Cook's Distance
o <- max(cooks.distance(CAPEX_Tæt_2017))
p <- max(cooks.distance(CAPEX_Tæt_2018))

# Her laves en itterativ outlier analyse, hvor de selskaber
# med en høj cooks distance værdi analyseres for at
# vurdere om det er en outlier

# Der er ingen outliers

summary(CAPEX_Tæt_2017)
summary(CAPEX_Tæt_2018)
```

Kapitel 2

Benchmarking

2.1 R-koder til brug for DEA og SFA modellen

Dette kapitel viser R-koder for DEA og SFA benchmarking. Koderne kan kopieres direkte ind i softwareprogrammet R. Koderne bruges til at fastsætte fronterne og beregne efficiensscore for selskaberne. I *Bilag 3 – Fronterne i DEA og SFA* beskrives fremgangsmetoden for fastsættelse af fronterne.

Beskrivelser af variable findes i *Bilag 7 – Data til R-koder v2*.

2.2 DEA

```
# Indlæser pakke "Benchmarking", for at kunne bruge "sfa"
# og "dea"
library(Benchmarking)

# Indlæsning af datasæt, indeholdende alle Netvolumenmål og
# FATO
library(readxl)
Data <- read_excel("Bilag 7 - Data til R-koder v2.xlsx")

#### Datasæt til Fronten
Front_Data <- Data

# Fjerner alle selskaber som kun varetager rens eller
# transport

Front_Data <- Front_Data[-c(which(Front_Data$Type == "r")),]
# Renseselskaber fjernes
Front_Data <- Front_Data[-c(which(Front_Data$Type == "t")),]
# Transportselskaber fjernes

# Selskaber der ikke skal indgå i fronten fjernes

Front_Data <- Front_Data[-c(which(Front_Data$ID ==
"S036")),] # Hedensted
Front_Data <- Front_Data[-c(which(Front_Data$ID ==
"S105")),] # Vesthimmerland
Front_Data <- Front_Data[-c(which(Front_Data$ID ==
"S007")),] # Billund

# Definerer variable til brug for at sætte fronten
Input_Front_DEA <- Front_Data$FATO_Front
Output_Front_DEA <- cbind(Front_Data$OPEX_Front,
Front_Data$OPEX_Tæt_Front, Front_Data$OPEX_Alder_Front,
Front_Data$CAPEX_Front, Front_Data$CAPEX_Alder_Front)

# Laver variable til brug for beregning af efficiensscore
# hvor der er taget højde for særlige forhold

Input <- Data$FATO_U_SF
```

```

Output <- cbind(Data$OPEX, Data$OPEX_Tæt, Data$OPEX_Alder,
Data$CAPEX, Data$CAPEX_Alder)

#Beregning af DEA modellen

A <- dea(Input, Output, RTS="crs", FAST = TRUE,
ORIENTATION = "in",
XREF = Input_Front_DEA, YREF = Output_Front_DEA)
summary(A)

# Sætter efficiensen for alle DEA-scoringer over 1 ned til 1
for(i in 1:length(A)){
  A[i] <- min(A[i],1)
}
A

```

2.3 SFA

```

# Indlæsning af datasæt med alle Netvolumenmål og FATO
library(readxl)
Data <- read_excel("Bilag 7 - Data til R-koder v2.xlsx")

#### Kode til fastholdelse af front i SFA
# For at kunne fastholde en front i SFA, skal følgende kode
# køres inden SFA beregningerne påbegyndes

te.sfa_PB_all <- function (object,x_all,y_all){
  lambda <- object$lambda
  s2 <- object$sigma2
  coefficients <- as.matrix(object$coef,ncol=1)
  n_obs_all <- dim(x_all)[1]
  x_constant_all <- as.matrix(cbind(rep(1,n_obs_all),x_all))
  #colnames(x_constant_all)[1]<-"(intercepts)"
  #colnames(x_constant_all)<-NULL
  residuals_all <- y_all- x_constant_all %*%
t(coefficients)[1,]
  ustar <- -residuals_all * lambda^2/(1 + lambda^2)
  sstar <- lambda/(1 + lambda^2) * sqrt(s2)
  TE = pnorm(ustar/sstar - sstar)/pnorm(ustar/sstar) *
exp(sstar^2/2 - ustar)
  colnames(TE) <- "te"
  return(array(TE))
}

#=====#
#### SFA med ukorrigerede netvolumenmål ####
#=====#

#Indlæser data
Front_Ukorr <- Data

# For at finde eventuelle outliers beregnes Cook's distance

outlier_Ukorr <- lm(log(Front_Ukorr$FATO) ~
log(Front_Ukorr$OPEX_Front) +
log(Front_Ukorr$CAPEX_Front))
j <- max(cooks.distance(outlier_Ukorr))

```

```

# Selskaber som ikke skal være med i frontfastsættelse
# fjernes her

Front_Ukorr <- Front_Ukorr[-c(which(Front_Ukorr$Type ==
"r")),] # renseselskaber fjernes
Front_Ukorr <- Front_Ukorr[-c(which(Front_Ukorr$Type ==
"t")),] # Transportselskaber fjernes
Front_Ukorr <- Front_Ukorr[-c(which(Front_Ukorr$ID ==
"S036")),] # Hedensted fjernes

# Laver input og output til fastsættelse af SFA-fronten med
# de ukorrigerede netvolumenmål

Input_Front_SFA_ukorr <- Front_Ukorr$FATO_Front
Output_Front_SFA_ukorr <- cbind(Front_Ukorr$OPEX_Front,
Front_Ukorr$CAPEX_Front)

# Laver SFA-fronten med de ukorrigerede netvolumenmål

O1 <- sfa(-log(Output_Front_SFA_ukorr),-
log(Input_Front_SFA_ukorr))
summary(O1)

# Laver input og output til beregning af efficiensscorer
# med de ukorrigerede netvolumenmål, hvor der er taget højde
# for særlige forhold

Input_ukorr <- Data$FATO_U_SF
Output_ukorr <- cbind(Data$OPEX, Data$CAPEX)

# Beregner SFA-scorerne med de ukorrigerede netvolumenmål

B1 <- te.sfa_PB_all(O1,-log(Output_ukorr),-log(Input_ukorr))
mean(B1)

#####
#### SFA med ALDERskorrigerede netvolumenmål ####
#####

# Indlæser data
Front_alder <- Data

# For at finde eventuelle outliers beregnes
# Cook's distance mål

outlier_Alder <- lm(log(Front_alder$FATO) ~
log(Front_alder$OPEX_Alder) + log(Front_alder$CAPEX_Alder))
k <- max(cooks.distance(outlier_Alder))

# Selskaber som ikke skal være med i frontfastsættelse
# fjernes her

Front_alder <- Data
Front_alder <- Front_alder [-c(which(Front_alder$Type ==
"r")),] # renseselskaber fjernes
Front_alder <- Front_alder [-c(which(Front_alder$Type ==
"t")),] # Transportselskaber fjernes
Front_alder <- Front_alder [-c(which(Front_alder$ID ==

```

```

"S036")),] # Hedensted fjernes

# Laver input og output til fastsættelse af SFA-fronten med
# de alderskorrigerede netvolumenmål

Input_Front_SFA_alder <- Front_alder$FATO_Front
Output_Front_SFA_alder <- cbind(Front_alder$OPEX_Alder_Front,
Front_alder$CAPEX_Alder_Front)

# Laver SFA-fronten med de alderskorrigerede netvolumenmål

O2 <- sfa(-log(Output_Front_SFA_alder), -
log(Input_Front_SFA_alder))
summary(O2)

# Laver input og output til beregning af efficiensscorer
# med de alderskorrigerede netvolumenmål, hvor der er taget
# højde for særlige forhold

Input_Alder <- Data$FATO_U_SF
Output_Alder <- cbind(Data$OPEX_Alder, Data$CAPEX_Alder)

# Beregner SFA-scorerne med de ukorrigerede netvolumenmål

B2 <- te.sfa_PB_all(O2, -log(Output_Alder), -log(Input_Alder))
mean(B2)

#####
#### SFA med Tæthedskorrigerede netvolumenmål ####
#####

# Indlæser data
Front_tæthed<-Data

# For at finde eventuelle outliers, køres nedenstående som
# beregner Cook's distance mål
outlier_Tæthed <- lm(log(Front_tæthed$FATO) ~
log(Front_tæthed$OPEX_Tæt))
j <- max(cooks.distance(outlier_Tæthed))

# Selskaber som ikke skal være med i frontfastsættelse
# fjernes her

Front_tæthed <- Data
Front_tæthed <- Front_tæthed[-c(which(Front_tæthed$Type ==
"r")),] # Renseselskaber fjernes
Front_tæthed <- Front_tæthed[-c(which(Front_tæthed$Type ==
"t")),] # Transportselskaber fjernes
Front_tæthed <- Front_tæthed[-c(which(Front_tæthed$ID ==
"S036")),] # Hedensted fjernes

# Laver input og output til fastsættelse af SFA-fronten med
# de Tæthedskorrigerede netvolumenmål

Input_Front_SFA_Tæt <- Front_tæthed$FATO_Front
Output_Front_SFA_Tæt <- cbind(Front_tæthed$OPEX_Tæt_Front,
Front_tæthed$CAPEX_Front)

```

```

# Laver SFA-fronten med de Tæthedskorrigerede netvolumenmål

O3 <- sfa(-log(Output_Front_SFA_Tæt), -
log(Input_Front_SFA_Tæt))
summary(O3)

# Laver input og output til beregning af efficiensscorer
# med de Tæthedskorrigerede netvolumenmål, hvor der er taget
# højde for særlige forhold

Input_Tæt <- Data$FATO_U_SF
Output_Tæt <- cbind(Data$OPEX_Tæt, Data$CAPEX)

# Beregner SFA-scorerne med de Tæthedskorrigerede
# netvolumenmål

B3 <- te.sfa_PB_all(O3, -log(Output_Tæt), -log(Input_Tæt))

```

2.4 SFA-modeller for outliers

```

#####
#### Modeller for renseselskaberne #####
#####

Front_R <- Data
Front_R <- Front_R[-c(which(Front_R$Type == "t")),]
# Transportselskaber fjernes
Front_R <- Front_R[-c(which(Front_R$ID == "S036")),]
# Hedensted fjernes

Input_Front_SFA_R <- Front_R$FATO_Front

#### Ukorrigerede netvolumenmål ####
Output_Front_SFA_R <- cbind(Front_R$OPEX_Front,
Front_R$CAPEX_Front)

O1_R <- sfa(-log(Output_Front_SFA_R), -log(Input_Front_SFA_R))
B1_R <- te.sfa_PB_all(O1_R, -log(Output_ukorr), -log(Input_ukorr))
B1[Data$Type=="r"] <- B1_R[Data$Type=="r"]
summary(O1_R)

#### Alderskorrigerede netvolumenmål ####
Output_Front_SFA_R_Alder <- cbind(Front_R$OPEX_Alder_Front,
Front_R$CAPEX_Alder_Front)

O2_R <- sfa(-log(Output_Front_SFA_R_Alder), -log(Input_Front_SFA_R))
B2_R <- te.sfa_PB_all(O2_R, -log(Output_Alder), -log(Input_Alder))
B2[Data$Type=="r"] <- B2_R[Data$Type=="r"]
summary(O2_R)

#### Tæthedskorrigerede netvolumenmål ####
Output_Front_SFA_R_Tæt <- cbind(Front_R$OPEX_Tæt_Front,
Front_R$CAPEX_Front)

O3_R <- sfa(-log(Output_Front_SFA_R_Tæt), -log(Input_Front_SFA_R))
B3_R <- te.sfa_PB_all(O3_R, -log(Output_Tæt), -log(Input_Tæt))
B3[Data$Type=="r"] <- B3_R[Data$Type=="r"]
summary(O3_R)

#####

```

```

#### Modeller for transportselskaberne          ####
#####

Front_T <- Data
Front_T <- Front_T[-c(which(Front_T$Type == "r")),]
# Renseselskaber fjernes
Front_T <- Front_T[-c(which(Front_T$ID == "S036")),]
# Hedensted fjernes

Input_Front_SFA_T <- Front_T$FATO_Front

#### Ukorrigerede netvolumenmål ####
Output_Front_SFA_T <- cbind(Front_T$OPEX_Front,
Front_T$CAPEX_Front)

O1_T <- sfa(-log(Output_Front_SFA_T),-log(Input_Front_SFA_T))
B1_T <- te.sfa_PB_all(O1_T,-log(Output_ukorr),-log(Input_ukorr))
B1[Data$Type=="t"] <- B1_T[Data$Type=="t"]
summary(O1_T)

#### Alderskorrigerede netvolumenmål ####
Output_Front_SFA_T_Alder <- cbind(Front_T$OPEX_Alder_Front,
Front_T$CAPEX_Alder_Front)

O2_T <- sfa(-log(Output_Front_SFA_T_Alder),-log(Input_Front_SFA_T))
B2_T <- te.sfa_PB_all(O2_T,-log(Output_Alder),-log(Input_Alder))
B2[Data$Type=="t"] <- B2_T[Data$Type=="t"]
summary(O2_T)

#### Tæthedskorrigerede netvolumenmål ####
Output_Front_SFA_T_Tæt <- cbind(Front_T$OPEX_Tæt_Front,
Front_T$CAPEX_Front)

O3_T <- sfa(-log(Output_Front_SFA_T_Tæt),-log(Input_Front_SFA_T))
B3_T <- te.sfa_PB_all(O3_T,-log(Output_Tæt),-log(Input_Tæt))
B3[Data$Type=="t"] <- B3_T[Data$Type=="t"]
summary(O3_T)

#####
#### Modeller for s036 (Hedensted)          ####
#####

Front_H <- Data
Front_H <- Front_H[-c(which(Front_H$Type == "r")),]
# Renseselskaber fjernes
Front_H <- Front_H[-c(which(Front_H$Type == "t")),]
# Transportselskaber fjernes

Input_Front_SFA_H <- Front_H$FATO_Front

#### Ukorrigerede netvolumenmål ####
Output_Front_SFA_H <- cbind(Front_H$OPEX_Front,
Front_H$CAPEX_Front)

O1_H <- sfa(-log(Output_Front_SFA_H),-log(Input_Front_SFA_H))
B1_H <- te.sfa_PB_all(O1_H,-log(Output_ukorr),-log(Input_ukorr))
B1[Data$ID=="S036"] <- B1_H[Data$ID=="S036"]
summary(O1_H)

#### Alderskorrigerede netvolumenmål ####
Output_Front_SFA_H_Alder <- cbind(Front_H$OPEX_Alder_Front,
Front_H$CAPEX_Alder_Front)

```

```
O2_H <- sfa(-log(Output_Front_SFA_H_Alder), -log(Input_Front_SFA_H))
B2_H <- te.sfa_PB_all(O2_H, -log(Output_Alder), -log(Input_Alder))
B2[Data$ID=="S036"] <- B2_H[Data$ID=="S036"]
summary(O2_H)

#### Tæthedskorrigerede netvolumenmål ####
Output_Front_SFA_H_Tæt <- cbind(Front_H$OPEX_Tæt_Front,
Front_H$CAPEX_Front)

O3_H <- sfa(-log(Output_Front_SFA_H_Tæt), -log(Input_Front_SFA_H))
B3_H <- te.sfa_PB_all(O3_H, -log(Output_Tæt), -log(Input_Tæt))
B3[Data$ID=="S036"] <- B3_H[Data$ID=="S036"]
summary(O3_H)
```

2.5 Samlet oversigt over scorer

```
#### Finder den højeste SFA score ####

B <- rep(NA, dim(Data)[1])
for(i in 1:length(B)){
  B[i] <- max(B1[i], B2[i], B3[i])
}
B

#### Finder BOT score ####
C <- rep(NA, dim(Data)[1])
for(i in 1:length(A)){
  C[i] <- max(A[i], B[i])
}
C

#### Datatabel for de endelige scorer ####
Data_scorer <- data.frame(cbind(Data$ID, Data$Navn, A, B, C))
colnames(Data_scorer) <-
c("ID", "NAVN", "DEA_Score", "SFA_Score", "BOT_Score")
View(Data_scorer)
```

Kapitel 3

Costdriversammensætning

3.1 R-koder til brug for analyse af costdriversammensætning

Dette kapitel viser R-koderne brugt til at lave analysen af costdriversammensætningen. I *Bilag 4 – Costdriversammensætning* beskrives metoden for analysen samt resultaterne for regressionerne.

Beskrivelser af variable findes i *Bilag 7 – Data til R-koder v2*.

3.2 Kode for costdriversammensætning

```
library(car)

# Indlæsning af datasæt
library(readxl)
Data <- read_excel("Bilag 7 - Data til R-koder v2.xlsx")

# Data$Score = "Best of two efficiensscore fra DEA og SFA
# benchmarking"

#####
##### OPEX #####
#####

#### Ledninger

# Regression
ledninger <- lm(Data$Score ~ Data$Ledninger)

# Finder mulige outliers med Cook's distance
j <- max(cooks.distance(ledninger))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(ledninger)

#### Pumpestationer

# Regression
pumpestationer <- lm(Data$Score ~ Data$Pumpestationer)

# Finder mulige outliers med Cook's distance
k <- max(cooks.distance(pumpestationer))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(pumpestationer)

#### Regnvandsbassiner
```

```
# Regression
regnvandsbassiner <- lm(Data$Score ~ Data$Regnvandsbassiner)

# Finder mulige outliers med Cook's distance
l <- max(cooks.distance(regnvandsbassiner))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(regnvandsbassiner)

#### Spildevandsbassiner

# Regression
spildevandsbassiner <- lm(Data$Score ~ Data$Spildevandsbassiner)

# Finder mulige outliers med Cook's distance
m <- max(cooks.distance(spildevandsbassiner))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(spildevandsbassiner)

#### Renseanlæg

# Regression
renseanlæg <- lm(Data$Score ~ Data$Renseanlæg)

# Finder mulige outliers med Cook's distance
n <- max(cooks.distance(renseanlæg))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(renseanlæg)

#### Minirensesanlæg

# Regression
minirensesanlæg <- lm(Data$Score ~ Data$Minirensesanlæg)

# Finder mulige outliers med Cook's distance
o <- max(cooks.distance(minirensesanlæg))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(minirensesanlæg)

#### Slambehandling

# Regression
slambehandling <- lm(Data$Score ~ Data$Slambehandling)

# Finder mulige outliers med Cook's distance
p <- max(cooks.distance(slambehandling))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(slambehandling)
```

```
#### Slamdisponering

# Regression
slamdisponering <- lm(Data$Score ~ Data$Slamdisponering)

# Finder mulige outliers med Cook's distance
q <- max(cooks.distance(slamdisponering))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(slamdisponering)

#### Kunder

# Regression
kunder <- lm(Data$Score ~ Data$Kunder)

# Finder mulige outliers med Cook's distance
r <- max(cooks.distance(kunder))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(kunder)

#### Generel administration

# Regression
general_adm <- lm(Data$Score ~ Data$Generel_Adm)

# Finder mulige outliers med Cook's distance
s <- max(cooks.distance(general_adm))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(general_adm)

##### CAPEX #####

#### Produktionsanlæg

# Regression
produktionsanlæg <- lm(Data$Score ~ Data$Produktionsanlæg)

# Finder mulige outliers med Cook's distance
t <- max(cooks.distance(produktionsanlæg))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(produktionsanlæg)

#### Distributionsanlæg

# Regression
distributionsanlæg <- lm(Data$Score ~ Data$Distributionsanlæg)

# Finder mulige outliers med Cook's distance
u <- max(cooks.distance(distributionsanlæg))
```

```
# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(distributionsanlæg)

#### Fællesfunktionsanlæg

# Regression
fællesfunktionsanlæg <- lm(Data$Score ~ Data$Fællesfunktionsanlæg)

# Finder mulige outliers med Cook's distance
v <- max(cooks.distance(fællesfunktionsanlæg))

# Modellens resultat
# (Eventuelle outliers fjernes og modellen køres igen)
summary(fællesfunktionsanlæg)
```
