Bilag 7

R-koder til brug for benchmarking af spildevandsselskaber

September 2021

Konkurrence- og Forbrugerstyrelsen

Bilag 7 – R-koder til brug for benchmarking af spildeevandsselskaber

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

Bilag 7 – R-koder til brug for benchmarking spildevandsselskaber til Metode for beregning af individuelle effektiviseringskrav – Benchmarking af spildevandsselskaber til brug for de økonomiske rammer 2022-2023 er udarbejdet af Forsyningssekretariatet.

September 2021

Indhold

Kapitel 1 4
Korrigerede netvolumenmål 4

1.1 R-koder til brug for beregningen af korrigerede netvolumenmål 4
1.2 Korrigerede netvolumenmål til brug for benchmarkingen 4

Kapitel 2 6
Benchmarking 6

2.1 R-koder til brug for DEA og SFA modellen 6
2.2 DEA 6
2.3 SFA 9
2.4 Samlet oversigt over scorer 12

Kapitel 3 13
Costdriversammensætning 13

3.1 R-koder til brug for analyse af costdriversammensætning 13
3.2 Kode for costdriversammensætning 13

Kapitel 1
Korrigerede netvolumenmål

1.1 R-koder til brug for beregningen af korrigerede netvolumenmål

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 korrigerede netvolumenmål. I *Bilag 3 – Beregning af de korrigerede netvolumenmål* fremgår resultaterne af regressionerne sammen med en forklaring på metoden.

Beskrivelser af variable findes i *Bilag 6 – Data til R-koder.*

1.2 Korrigerede netvolumenmål til brug for benchmarkingen

```
####---- Indlæsning af data ----####
# Tallene i datasættet er gennemsnit af 2019 og 2020
library(readxl)
Data <- read_excel("Bilag 6 - Data til R-koder.xlsx")

#=====================================#
#### Korrektion for alder og tæthed ####
#=====================================#

####---- Data til korrektion af OPEX- og CAPEX-netvolumenmål ----
####
Data_korrektion<-Data[-which(Data$TYPE=="r"),]

#-------------------------------#
####---- OPEX korrektion ----####
#-------------------------------#

# OPEX regressionen
OPEX_Fit<-
lm(Data_korrektion$FADO_FRONT/Data_korrektion$OPEX_U_FRONT~Data_kor
rektion$ALDER_FRONT+Data_korrektion$TAETHED_FRONT)
summary(OPEX_Fit)

# Finder mulige outliers med Cook's Distance - selskaber med Cook's
distance værdier over 0,5 undersøges nærmere.
cbind(Data_korrektion$NAVN,Data_korrektion$ID,cooks.distance(OPEX_F
it))[order(cooks.distance(OPEX_Fit)),]

#-------------------------------#
####---- CAPEX korrektion ----####
#-------------------------------#

# CAPEX regression
CAPEX_Fit<-
lm(Data_korrektion$INV_OMK_FRONT/Data_korrektion$CAPEX_U_FRONT~Data
_korrektion$ALDER_FRONT+Data_korrektion$TAETHED_FRONT)
summary(CAPEX_Fit)
```

```
# Finder mulige outliers med Cook's Distance - selskaber med Cook's
distance værdier over 0,5 undersøges nærmere.
cbind(Data_korrektion$NAVN,Data_korrektion$ID,cooks.distance(CAPEX_
Fit))[order(cooks.distance(CAPEX_Fit)),]

#================================================#
#### Beregning af de korrigerede netvolumenmål ####
#================================================#
# Koefficienterne fra regressionerne OPEX_Fit og CAPEX_Fit bruges
til at beregne de korrigerede netvolumenmål
# Der afrundes til seks decimaler, så resultaterne stemmer overens
med Bilag 1 - Data til brug for fastlæggelse af de individuelle
effektiviseringskrav til Metode for beregning af individuelle
effektiviseringskrav - benchmarking til brug for de økonomiske
rammer 2022-2023

# Korrigeret netvolumenmål for alle selskaber - OPEX
OPEX_Kor<-
(round(coef(OPEX_Fit)[1],6)+round(coef(OPEX_Fit)[2],6)*Data$ALDER_F
RONT+round(coef(OPEX_Fit)[3],6)*Data$TAETHED_FRONT)*Data$OPEX_U_FRO
NT

# Korrigeret netvolumenmål for alle selskaber - CAPEX
CAPEX_Kor<-
(round(coef(CAPEX_Fit)[1],6)+round(coef(CAPEX_Fit)[2],6)*Data$ALDER
_FRONT+round(coef(CAPEX_Fit)[3],6)*Data$TAETHED_FRONT)*Data$CAPEX_U
_FRONT

####---- Datasæt med endelige korrigerede netvolumenmål ----####
data.frame(cbind(Data$ID,Data$NAVN,OPEX_Kor,CAPEX_Kor))
```

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 efficiensscorer for selskaberne. I *Bilag 3 – Fronterne i DEA og SFA* beskrives fremgangsmetoden for fastsættelse af fronterne.

Beskrivelser af variable findes i *Bilag 6 – Data til R-koder.*

2.2 DEA

```
####---- Indlæsning af data ----####

library(readxl)

# Tallene i datasættet er gennemsnit af 2019 og 2020
Data <- read_excel("Bilag 6 - Data til R-koder.xlsx ")

#--------------------#
####---- DEA ----####
#--------------------#

# Indlæser pakke til brug for DEA benchmarking
library(Benchmarking)

####---- Input og output til benchmarking ----####
Input <- Data$OMKG
Output <- cbind(Data$OPEX_KOR, Data$CAPEX_KOR)

####---- Front data ----####
# Omdøber data, så vi kan arbejde med det til fronten
Front_dea <- Data

# Definerer selskaber, der ikke skal indgå i fronten for hverken
rense- transport eller fællesselskaber (outliers for DEA)
Fjern_dea<-c("S045","S078","S094","S005")
# "S045" - HOFOR Spildevand København A/S
# "S078" - Ringsted Centralrenseanlæg A/S
# "S094" - Thisted Renseanlæg A/S
# "S005" - ASSENS RENSNING A/S

# Fjerner selskaber, der ikke skal indgå i fronten
Front_dea <- Front_dea[-c(which(Front_dea$ID %in% Fjern_dea)),]

#-----------------------------------------------#
####---- DEA for selskaber af typen "t+r" ----####
#-----------------------------------------------#

####---- Fronten for selskaber af typen "t+r" ----####
# Kun selskaber af typen "t+r" indgår i fronten
Front_dea_tr<-Front_dea[which(Front_dea$TYPE=="t+r"),]
```

```
####---- Input og output til fronten ----####
Input_Front_DEA_tr <- Front_dea_tr$OMKG_FRONT
Output_Front_DEA_tr <- cbind(Front_dea_tr$OPEX_KOR_FRONT,
Front_dea_tr$CAPEX_KOR_FRONT)

####---- DEA benchmarking -----####
# Scorer for selskaber af typen "t+r"
dea_tr <- dea(Input, Output, RTS="crs", ORIENTATION = "in", XREF =
Input_Front_DEA_tr, YREF = Output_Front_DEA_tr)
dea_tr_eff<- dea_tr$eff

####---- Her printes navn og ID på frontselskaber ----####
cbind(ID=Data$ID[which(dea_tr_eff ==
1)],Navn=Data$NAVN[which(dea_tr_eff == 1)])

#-----------------------------------------------#
####---- DEA for selskaber af typen "t" ----####
#-----------------------------------------------#

####---- Fronten for selskaber af typen "t" ----####
# Kun selskaber af typen "t+r" eller "t" indgår i fronten
Front_dea_t<-
Front_dea[which(Front_dea$TYPE=="t+r"|Front_dea$TYPE=="t"),]

####---- Input og output til fronten ----####
Input_Front_DEA_t <- Front_dea_t$OMKG_FRONT
Output_Front_DEA_t <- cbind(Front_dea_t$OPEX_KOR_FRONT,
Front_dea_t$CAPEX_KOR_FRONT)

####---- DEA benchmarking -----####
# Scorer for selskaber af typen "t"
dea_t <- dea(Input, Output, RTS="crs", ORIENTATION = "in", XREF =
Input_Front_DEA_t, YREF = Output_Front_DEA_t)
dea_t_eff<- dea_t$eff

####---- Her printes navn og ID på frontselskaber ----####
cbind(ID=Data$ID[which(dea_t_eff ==
1)],Navn=Data$NAVN[which(dea_t_eff == 1)])

#-----------------------------------------------#
####---- DEA for selskaber af typen "r" ----####
#-----------------------------------------------#

####---- Fronten for selskaber af typen "r" ----####
# Kun selskaber af typen "t+r" eller "r" indgår i fronten
Front_dea_r<-
Front_dea[which(Front_dea$TYPE=="t+r"|Front_dea$TYPE=="r"),]

####---- Input og output til fronten ----####
Input_Front_DEA_r <- Front_dea_r$OMKG_FRONT
Output_Front_DEA_r <- cbind(Front_dea_r$OPEX_KOR_FRONT,
Front_dea_r$CAPEX_KOR_FRONT)

####---- DEA benchmarking -----####
# Scorer for selskaber af typen "r"
dea_r <- dea(Input, Output, RTS="crs", ORIENTATION = "in", XREF =
Input_Front_DEA_r, YREF = Output_Front_DEA_r)
dea_r_eff<- dea_r$eff

####---- Her printes navn og ID på frontselskaber ----####
cbind(ID=Data$ID[which(dea_r_eff ==
1)],Navn=Data$NAVN[which(dea_r_eff == 1)])
```

```
#-----------------------------------------------#
####---- DEA scoren bindes på datasættet ----####
#-----------------------------------------------#

Data$dea_score<-
  ifelse( Data$TYPE == "t+r", dea_tr_eff,
          ifelse( Data$TYPE == "t", dea_t_eff,
                  ifelse( Data$TYPE == "r", dea_r_eff, NA ) ) )


#OBS! scoren erstattes for outliers, der ikke skal indgå i fronten
vha. koden nedenfor

#-------------------------------------#
####---- DEA for evt. outliers ----####
#-------------------------------------#

# Her køres DEA modellen for hver enkelt selskab, der ikke skal
indgå i fronten i ovenstående modeller
# Den enkelte outlier indgår altid selv i den model selskabet
benchmarkes i
# Øvrige outliers indgår ikke i frontfastlæggelsen for den enkelte
outlier

for (i in Fjern_dea){

  #### Front data
  Front_dea_outlier <- Data

  # Fjerner øvrige outliers for fronten
  Front_dea_outlier <- Front_dea_outlier[-
c(which(Front_dea_outlier$ID %in% Fjern_dea[-which(Fjern_dea==i)]
)),]

  # Bestemmer fronten for selskabstypen
  if(Data$TYPE[which(Data$ID==i)]=="t"){

    # Fronten for selskaber af typen "t"
    Front_dea_outlier<-
Front_dea_outlier[which(Front_dea_outlier$TYPE=="t+r"|Front_dea_out
lier$TYPE=="t"),]
  } else {

    if(Data$TYPE[which(Data$ID==i)]=="r"){

      # Fronten for selskaber af typen "r"
      Front_dea_outlier<-
Front_dea_outlier[which(Front_dea_outlier$TYPE=="t+r"|Front_dea_out
lier$TYPE=="r"),]
    }}

  #### DEA score for outlier

  Input_Front_DEA_outlier <- Front_dea_outlier$OMKG_FRONT
  Output_Front_DEA_outlier <-
cbind(Front_dea_outlier$OPEX_KOR_FRONT,
Front_dea_outlier$CAPEX_KOR_FRONT)

  dea_outlier <- dea(Input, Output, RTS="crs", ORIENTATION = "in",
XREF = Input_Front_DEA_outlier, YREF = Output_Front_DEA_outlier)

  # Scoren for outlier erstattes med den nye score
```

```
  Data$dea_score[which(Data$ID==i)]<-
dea_outlier$eff[which(Data$ID==i)]

}

####---- Endelige DEA-scorer ----####

# Sætter DEA-efficiensscoren for alle over én ned til én
# Relevant for selskaber, der efter frontfastlæggelsen har
ændringer til data, så de potentielt kan opnå en score over én

for(i in 1:length(Data$dea_score)){
   Data$dea_score[i] <- min(Data$dea_score[i],1)
}

# Endelige DEA-scorer
data.frame(cbind(ID=Data$ID,Navn=Data$NAVN,DEA_Score=Data$dea_score
))
mean(Data$dea_score)
```

2.3 SFA

```
#--------------------#
####---- SFA ----####
#--------------------#

####---- Indlæser pakker til brug for SFA benchmarking ----####
detach("package:Benchmarking", unload = TRUE)
# Detach "Benchmarking" pakken, da "frontier" pakkens funktionskald
skal bruges til SFA Benchmarking

install.packages("frontier", repos="http://R-Forge.R-project.org")
# OBS! Det er vigtigt, at denne version af Frontier pakken
installeres i R, da denne er opdateret med seneste ændringer

library(frontier)
library(micEcon)
library(quadprog)
library(miscTools)
library(Matrix)
library(matrixcalc)

####---- Data til fronten ----####
Front_sfa <- Data

####---- Data til benchmarking ----####
# Tager log til variablene til brug for translog modellen
BM_Data <- data.frame(Data$NAVN, Data$ID, log( Data$OMKG ), log(
Data$OPEX_KOR ), log( Data$CAPEX_KOR ))
names(BM_Data) <- c("ID", "Navn", "logOMKG", "logOpex", "logCapex")

#-----------------------------------------------#
####---- SFA for selskaber af typen "t+r" ----####
#-----------------------------------------------#

####---- Fronten for selskaber af typen "t+r" ----####
Front_sfa_tr<-Front_sfa[which(Front_sfa$TYPE=="t+r"),]

####---- Input og output til fronten ----####
# Tager log til variablene til brug for translog modellen
Front_sfa_tr$logOMKG <- log( Front_sfa_tr$OMKG_FRONT)
Front_sfa_tr$logOpex <- log( Front_sfa_tr$OPEX_KOR_FRONT )
```

```
Front_sfa_tr$logCapex <- log( Front_sfa_tr$CAPEX_KOR_FRONT)

####---- SFA benchmarking ----####
# Unrestricted estimation for SFA med translog funktionel form
sfaEst_tr <- sfa( logOMKG ~ logOpex + logCapex +
                    I(0.5*logOpex^2) + I(logOpex*logCapex) +
I(0.5*logCapex^2) ,
                  data = Front_sfa_tr, ineffDecrease = FALSE )
summary(sfaEst_tr)

sfa_tr<-efficiencies(sfaEst_tr, newdata = BM_Data) # Beregner
scorer for unrestricted model

####---- Tjekker om monotonicitet er brudt ----####
# OPEX
table((coef(sfaEst_tr)[2]+coef(sfaEst_tr)[4]*Front_sfa_tr$logOpex+c
oef(sfaEst_tr)[5]*Front_sfa_tr$logCapex) <0)
# CAPEX
table((coef(sfaEst_tr)[3]+coef(sfaEst_tr)[6]*Front_sfa_tr$logCapex+
coef(sfaEst_tr)[5]*Front_sfa_tr$logOpex) <0)

# monotonicitet er ikke brudt, så vi fortsætter med unrestricted
model

####--- Cook's distance til unrestricted model ----####
Front_sfa_tr$PCD_eff_tr<-cooks.distance(sfaEst_tr,target =
"efficiencies", asInData = TRUE)
PCD_tr<-
data.frame(cbind(Front_sfa_tr$ID,Front_sfa_tr$NAVN,Front_sfa_tr$TYP
E,Front_sfa_tr$PCD_eff_tr))
PCD_tr<-PCD_tr[order(PCD_tr[,4]),]
PCD_tr

#-----------------------------------------------#
####---- SFA for selskaber af typen "t" ----####
#-----------------------------------------------#

####---- Fronten for selskaber af typen "t" ----####
Front_sfa_t<-
Front_sfa[which(Front_sfa$TYPE=="t+r"|Front_sfa$TYPE=="t"),]

####---- Input og output til fronten ----####
# Tager log til variablene til brug for translog modellen
Front_sfa_t$logOMKG <- log( Front_sfa_t$OMKG_FRONT)
Front_sfa_t$logOpex <- log( Front_sfa_t$OPEX_KOR_FRONT )
Front_sfa_t$logCapex <- log( Front_sfa_t$CAPEX_KOR_FRONT)

####---- SFA benchmarking ----####
# Unrestricted estimation for SFA med translog funktionel form
sfaEst_t <- sfa( logOMKG ~ logOpex + logCapex +
                   I(0.5*logOpex^2) + I(logOpex*logCapex) +
I(0.5*logCapex^2) ,
                 data = Front_sfa_t, ineffDecrease = FALSE )
summary(sfaEst_t)

sfa_t<-efficiencies(sfaEst_t, newdata = BM_Data)  # Beregner scorer
for unrestricted model

# Bemærk, at dette ikke er den endelige model, da monotonicitet
ikke nødvendigvis er opfyldt for samtlige selskaber - dette
pålægges modellen herunder

####---- Tjekker om monotonicitet er brudt ----####
```

```
# OPEX
table((coef(sfaEst_t)[2]+coef(sfaEst_t)[4]*Front_sfa_t$logOpex+coef
(sfaEst_t)[5]*Front_sfa_t$logCapex) <0)
# CAPEX
table((coef(sfaEst_t)[3]+coef(sfaEst_t)[6]*Front_sfa_t$logCapex+coe
f(sfaEst_t)[5]*Front_sfa_t$logOpex) <0)

# monotonicitet er ikke brudt, så vi fortsætter med unrestricted
model

####--- Cook's distance til unrestricted model ----####
Front_sfa_t$PCD_eff_t<-cooks.distance(sfaEst_t,target =
"efficiencies", asInData = TRUE)
PCD_t<-
data.frame(cbind(Front_sfa_t$ID,Front_sfa_t$NAVN,Front_sfa_t$TYPE,F
ront_sfa_t$PCD_eff_t))
PCD_t<-PCD_t[order(PCD_t[,4]),]
PCD_t

#-----------------------------------------------#
####---- SFA for selskaber af typen "r" ----####
#-----------------------------------------------#

####---- Fronten for selskaber af typen "r" ----####
Front_sfa_r<-
Front_sfa[which(Front_sfa$TYPE=="t+r"|Front_sfa$TYPE=="r"),]

####---- Input og output til fronten ----####
# Tager log til variablene til brug for translog modellen
Front_sfa_r$logOMKG <- log( Front_sfa_r$OMKG_FRONT)
Front_sfa_r$logOpex <- log( Front_sfa_r$OPEX_KOR_FRONT )
Front_sfa_r$logCapex <- log( Front_sfa_r$CAPEX_KOR_FRONT)

####---- SFA benchmarking ----####
# Unrestricted estimation for SFA med translog funktionel form
sfaEst_r <- sfa( logOMKG ~ logOpex + logCapex +
                   I(0.5*logOpex^2) + I(logOpex*logCapex) +
I(0.5*logCapex^2) ,
                 data = Front_sfa_r, ineffDecrease = FALSE )
summary(sfaEst_r)

sfa_r<-efficiencies(sfaEst_r, newdata = BM_Data) # Beregner scorer
for unrestricted model

####---- Tjekker om monotonicitet er brudt ----####
# OPEX
table((coef(sfaEst_r)[2]+coef(sfaEst_r)[4]*Front_sfa_r$logOpex+coef
(sfaEst_r)[5]*Front_sfa_r$logCapex) <0)
# CAPEX
table((coef(sfaEst_r)[3]+coef(sfaEst_r)[6]*Front_sfa_r$logCapex+coe
f(sfaEst_r)[5]*Front_sfa_r$logOpex) <0)

# monotonicitet er ikke brudt, så vi fortsætter med unrestricted
model

####--- Cook's distance til unrestricted model ----####
Front_sfa_r$PCD_eff_r<-cooks.distance(sfaEst_r,target =
"efficiencies", asInData = TRUE)
PCD_r<-
data.frame(cbind(Front_sfa_r$ID,Front_sfa_r$NAVN,Front_sfa_r$TYPE,F
ront_sfa_r$PCD_eff_r))
PCD_r<-PCD_r[order(PCD_r[,4]),]
PCD_r
```

```
#-----------------------------------------------#
####---- SFA scoren bindes på datasættet ----####
#-----------------------------------------------#

Data$sfa_score<-
  ifelse( Data$TYPE == "t+r", sfa_tr,
          ifelse( Data$TYPE == "t", sfa_t,
                  ifelse( Data$TYPE == "r", sfa_r, NA ) ) )

mean(Data$sfa_score)
```

2.4 Samlet oversigt over scorer

```
####--- Finder BOT score ----####
A <- Data$dea_score
B <- Data$sfa_score

C <- rep(NA,dim(Data)[1])

for(i in 1:length(A)){
  C[i] <- max(A[i],B[i])
}

# BOT score
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 5 – Costdriveranalyse* beskrives metoden for analysen samt resultaterne for regressionerne.

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

3.2 Kode for costdriversammensætning

```
####---- Indlæsning af data ----####
# Tallene i datasættet er gennemsnit af 2019 og 2020
library(readxl)
Data <- read_excel("Bilag 6 - Data til R-koder.xlsx")

#--------------------#
####---- OPEX ----####
#--------------------#

####---- Ledninger ----####

# Den lineære sammenhæng mellem ledninger og efficiensscoren
udregnes
model <- lm(Data$Score ~ Data$Ledninger)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


#####---- Pumpestationer ----####

# Den lineære sammenhæng mellem pumper og efficiensscoren udregnes
model <- lm(Data$Score ~ Data$Pumpe)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Regnvandsbassiner ----####

# Den lineære sammenhæng mellem regnvandsbassiner og
efficiensscoren udregnes
model <- lm(Data$Score ~ Data$Regnbas)
summary(model)
```

```
# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Spildevandsbassiner ----####

# Den lineære sammenhæng mellem spildevandsbassiner og
efficiensscoren udregnes
model <- lm(Data$Score ~ Data$Spildebas)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Renseanlæg ----####

# Den lineære sammenhæng mellem renseanlæg og efficiensscoren
udregnes
model <- lm(Data$Score ~ Data$Rense)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Minirenseanlæg ----####

# Den lineære sammenhæng mellem Minirenseanlæg og efficiensscoren
udregnes
model <- lm(Data$Score ~ Data$Minirense)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Slambehandling ----####

# Den lineære sammenhæng mellem slambehandling og efficiensscoren
udregnes
model <- lm(Data$Score ~ Data$Behandling)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))
```

```
####---- Slamdisponering ----####

# Den lineære sammenhæng mellem slamdisponering og efficiensscoren
udregnes
model <- lm(Data$Score ~ Data$Disponering)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Kunder ----####

# Den lineære sammenhæng mellem kunder og efficiensscoren udregnes
model <- lm(Data$Score ~ Data$Kunder)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Generel administration ----####

# Den lineære sammenhæng mellem generel administration og
efficiensscoren udregnes
model <- lm(Data$Score ~ Data$Adm)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))

#-----------------------------------------#
####---- Costdriveranalysen for CAPEX ----####
#-----------------------------------------#

####---- Produktionsanlæg ----####

# Den lineære sammenhæng mellem produktionsanlæg og efficiensscoren
udregnes
model <- lm(Data$Score ~ Data$Prod)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))


####---- Distributionsanlæg ----####

# Den lineære sammenhæng mellem distributionsanlæg og
efficiensscoren udregnes
model <- lm(Data$Score ~ Data$Dist)
```

```
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))
## Fællesfunktionsanlæg


####---- Fællesfunktionsanlæg ----####
# Den lineære sammenhæng mellem fællesfunktionsanlæg og
efficiensscoren udregnes
model <- lm(Data$Score ~ Data$Faelles)
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))

## Ny model uden outlier
# "S055 - Kalundborg renseanlæg A/S" fjernes som outlier i modellen
model <- lm(Data$Score[-which(Data$ID=="S055")] ~ Data$Faelles[-
which(Data$ID=="S055")])
summary(model)

# Modellen tjekkes for outliers med en cooks distance større end
0.5
Data$NAVN[which(cooks.distance(model) ==
max(cooks.distance(model)))]
max(cooks.distance(model))
```