

Bilag 4

R-koder til brug for
benchmarking af
spildevandsselskaber

September 2023



Bilag 4 – R-koder til brug for benchmarking af spildevandsselskaber

Konkurrence- og Forbrugerstyrelsen

Forsyningssekretariatet

Carl Jacobsens Vej 35

2500 Valby

Tlf.: +45 41 71 50 00

E-mail: kfst@kfst.dk

Bilag 4 – R-koder til brug for benchmarking spildevandsselskaber – Benchmarking af spildevandsselskaber til brug for de økonomiske rammer 2024-2025 er udarbejdet af Forsyningssekretariatet.

September 2023

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 Data indlæses	6
2.3 DEA.....	6
2.4 SFA.....	8
2.5 Samlet oversigt over scorer	11
Kapitel 3	12
Costdriversammensætning.....	12
3.1 R-koder til brug for analyse af costdriversammensætning.....	12
3.2 Kode for costdriversammensætning	12

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.

Beskrivelser af variable samt data til kørsel af koderne kan findes i fanerne "Til R-koder" og "Beskrivelse af data til R-koder" i [Bilag 1 - Data til brug for fastlæggelse af de individuelle effektiviseringskrav](#).

Koderne kan kopieres direkte ind i softwareprogrammet R. Koderne laver regressionerne, der ligger til grund for beregningen af de korrigerede netvolumenmål. I [Bilag 2 - Beregning af de korrigerede netvolumenmål](#) fremgår resultaterne af regressionerne sammen med en forklaring på metoden.

1.2 Korrigerede netvolumenmål til brug for benchmarkingen

```
#####
#### Korrektion for alder og tæthed ####
#####

####---- Indlæsning af data ----####
library(readxl)
Data <- read_xlsx("Bilag 1 - Data til brug for fastlæggelse af de
individuelle effektiviseringskrav.xlsx", sheet = "Til R-koder")

# Fjerner selskaber der ikke har indberettet til benchmarking
Data <- Data[!Data$ID=="S088",]

# Data til korrektion af netvolumenmålene
Data_korrektion <- Data[-which(Data$TYPE=="r"),]

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

## OPEX med alle selskaber
OPEX_Fit <- lm(FADO_FROSSET/OPEX_U_FROSSET ~ALDER_FROSSET+
TAETHED_FROSSET , data=Data_korrektion)
cbind(Data_korrektion$NAVN,Data_korrektion$ID,
cooks.distance(OPEX_Fit))[order(cooks.distance(OPEX_Fit)),]
summary(OPEX_Fit)

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

## CAPEX med alle selskaber
CAPEX_Fit<-lm(INV_OMK_FROSSET/CAPEX_U_FROSSET~ALDER_FROSSET+
TAETHED_FROSSET , data=Data_korrektion)
```

```
cbind(Data_korrektion$NAVN,Data_korrektion$ID,  
cooks.distance(CAPEX_Fit))[order(cooks.distance(CAPEX_Fit)),]  
summary(CAPEX_Fit)
```

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 bruges til at fastsætte fronterne og beregne efficiensscorer for selskaberne.

2.2 Data indlæses

```
####---- Indlæsning af data ----####
# Indlæser pakke som bruges til at åbne data-fil
library(readxl)
Data <- read_xlsx("Bilag 1 - Data til brug for fastlæggelse af de
individuelle effektiviseringskrav.xlsx", sheet = "Til R-koder")
# Fjerner selskaber der ikke har indberettet til benchmarking
Data <- Data[!Data$ID=="S088",]
```

2.3 DEA

```
#=====#
#####      DEA-MODELLEN      #####
#=====#

# Indlæser pakke som bruges til DEA-benchmark
library(Benchmarking)

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

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

# Fjerner de selskaber som vi ikke mener skal være en del af
fronten (i nogen af modellerne)
Fjern_dea <-
c("S078", "S094", "S111", "S045", "S014", "S095", "S068", "S041")

# "S078" - Ringsted Centralrenseanlæg
# "S094" - Thisted Spildevand Transport
# "S111" - RINGKØBING-SKJERN RENSEANLÆG A/S
# "S045" - HOFOR Spildevand København A/S
# "S014" - Energi Viborg Vand A/S
# "S095" - Thisted Spildevand Transport A/S
# "S068" - MORSØ SPILDEVAND A/S
# "S041" - HOFOR Spildevand Albertslund A/S

Front_dea <- Front_dea[-c(which(Front_dea$ID %in% Fjern_dea)),]
```

```

#-----#
####--- trans+rens ----###
#-----#

####---- Input og output til fronten ----###
Input_Front_DEA_tr <- Front_dea[which(Front_dea$TYPE=="t+r"),
"FATO_FROSSET"]

Output_Front_DEA_tr <- Front_dea[which(Front_dea$TYPE=="t+r"),
c("OPEX_KOR_FROSSET", "CAPEX_KOR_FROSSET")]

# Scorer for selskaber af typen "t+r" (Kun selskaber af typen "t+r"
indgår i fronten)
dea_tr <- dea(Input, Output, RTS="crs", ORIENTATION = "in",
             XREF = Input_Front_DEA_tr, YREF =
Output_Front_DEA_tr)

####---- Her printes navn, ID og type på frontelskaber ----###
Data[which(dea_tr$eff == 1),c("NAVN","ID","TYPE")]

#-----#
####--- transport ----###
#-----#

####---- Input og output til fronten ----###
Input_Front_DEA_t <- Front_dea[which(Front_dea$TYPE=="t+r" |
Front_dea$TYPE=="t"), "FATO_FROSSET"]

Output_Front_DEA_t <- Front_dea[which(Front_dea$TYPE=="t+r" |
Front_dea$TYPE=="t"), c("OPEX_KOR_FROSSET", "CAPEX_KOR_FROSSET")]

# Scorer for selskaber af typen "t" (Selskaber af typen "t+r" samt
"t" indgår i fronten)
dea_t <- dea(Input, Output, RTS="crs", ORIENTATION = "in",
            XREF = Input_Front_DEA_t, YREF = Output_Front_DEA_t)

####---- Her printes navn, ID og type på frontelskaber ----###
Data[which(dea_t$eff == 1),c("NAVN","ID","TYPE")]

#-----#
####--- renseanlæg ----###
#-----#

####---- Input og output til fronten ----###
Input_Front_DEA_r <- Front_dea[which(Front_dea$TYPE=="t+r" |
Front_dea$TYPE=="r"), "FATO_FROSSET"]
Output_Front_DEA_r <- Front_dea[which(Front_dea$TYPE=="t+r" |
Front_dea$TYPE=="r"), c("OPEX_KOR_FROSSET", "CAPEX_KOR_FROSSET")]

# Scorer for selskaber af typen "r" (Selskaber af typen "t+r" samt
"r" indgår i fronten)
dea_r <- dea(Input, Output, RTS="crs", ORIENTATION = "in",
            XREF = Input_Front_DEA_r, YREF = Output_Front_DEA_r)

####---- Her printes navn, ID og type på frontelskaber ----###
Data[which(dea_r$eff == 1),c("NAVN","ID","TYPE")]

#-----#
####--- 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" ,
dear$eff, NA ) ) )

Data[Data$dea_score>1 , "dea_score"] <- 1

mean(Data$dea_score)
```

2.4 SFA

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

# OBS! DEA skal køres med "Benchmarking"-pakken, men SFA med
"frontier"-pakken

detach("package:Benchmarking", unload = TRUE)

# install.packages("frontier", repos="http://R-Forge.R-
project.org")

# det er vigtigt at det er denne version af frontierpakken der
hentes, da denne indeholder seneste udgave (1.1.9)
library(frontier)

# Tjek at frontier-versionen er korrekt
if(packageVersion("frontier") != "1.1.9"){
  warning("Advarsel: Den aktuelle version af 'frontier' er
forældet. Opdater til version 1.1.9 ")
  stop()}

# Omdøber data så vi kan arbejde med det til fronten
Front_sfa <- Data

# Data for alle til SFA-modellen
BM_Data <- data.frame(Data$NAVN, Data$ID, log( Data$FATO ), log(
Data$OPEX_KOR ), log( Data$CAPEX_KOR ))
names(BM_Data) <- c("ID", "Navn", "logFato", "logOpex", "logCapex")

#-----#
###--- trans+rens ----###
#-----#

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

###---- Front data ----###
Front_sfa_tr$logFato <- log( Front_sfa_tr$FATO_FROSSET)
Front_sfa_tr$logOpex <- log( Front_sfa_tr$OPEX_KOR_FROSSET )
Front_sfa_tr$logCapex <- log( Front_sfa_tr$CAPEX_KOR_FROSSET)

# unrestricted estimation af SFA-model
sfaEst_tr <- sfa( logFato ~ logOpex + logCapex + I(0.5*logOpex^2) +
I(logOpex*logCapex) + I(0.5*logCapex^2) , data = Front_sfa_tr,
ineffDecrease = FALSE )

###---- Tjek om monotonicitet er brudt ----###
```

```

# OPEX
table((coef(sfaEst_tr)[2]+coef(sfaEst_tr)[4]*Front_sfa_tr$logOpex+coef(sfaEst_tr)[5]*Front_sfa_tr$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)

####---- Cook's distance til unrestricted model ----####
Front_sfa_tr$PCD_eff_tr<-cooks.distance(sfaEst_tr,target =
"efficiencies", asInData = TRUE)

# Scorer for selskaber af typen "r" (Selskaber af typen "t+r"
indgår i fronten)
sfa_tr <- efficiencies(sfaEst_tr, newdata = BM_Data) # Beregner
scorer for unrestricted model

#-----#
####---- transport ----####
#-----#

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

####---- Front data ----####
Front_sfa_t$logFato <- log( Front_sfa_t$FATO_FROSSET)
Front_sfa_t$logOpex <- log( Front_sfa_t$OPEX_KOR_FROSSET )
Front_sfa_t$logCapex <- log( Front_sfa_t$CAPEX_KOR_FROSSET)

# unrestricted estimation af SFA-model
sfaEst_t <- sfa( logFato ~ logOpex + logCapex + I(0.5*logOpex^2) +
I(logOpex*logCapex) + I(0.5*logCapex^2) , data = Front_sfa_t,
ineffDecrease = FALSE )

####---- Tjek 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$logCapex+coef(sfaEst_t)[5]*Front_sfa_t$logOpex) <0)

####---- Tjek om monotonicitet er brudt - kun "transport" ----####
# OPEX
table((coef(sfaEst_t)[2]+coef(sfaEst_t)[4]*Front_sfa_t$logOpex[which(Front_sfa_t$TYPE=="t")]+coef(sfaEst_t)[5]*Front_sfa_t$logCapex[which(Front_sfa_t$TYPE=="t")]) <0)
# CAPEX
table((coef(sfaEst_t)[3]+coef(sfaEst_t)[6]*Front_sfa_t$logCapex[which(Front_sfa_t$TYPE=="t")]+coef(sfaEst_t)[5]*Front_sfa_t$logOpex[which(Front_sfa_t$TYPE=="t")]) <0)

####---- Cook's distance til unrestricted model ----####
Front_sfa_t$PCD_eff_t <- cooks.distance(sfaEst_t,target =
"efficiencies", asInData = TRUE)

# Scorer for selskaber af typen "r" (Selskaber af typen "t+r" samt
"t" indgår i fronten)
sfa_t <- efficiencies(sfaEst_t, newdata = BM_Data)

#-----#

```

```

####--- renseanlæg ----####
#-----#

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

####--- Front data ----####
Front_sfa_r$logFato <- log( Front_sfa_r$FATO_FROSSET)
Front_sfa_r$logOpex <- log( Front_sfa_r$OPEX_KOR_FROSSET )
Front_sfa_r$logCapex <- log( Front_sfa_r$CAPEX_KOR_FROSSET)

# unrestricted estimation af SFA-model
sfaEst_r <- sfa( logFato ~ logOpex + logCapex + I(0.5*logOpex^2) +
I(logOpex*logCapex) + I(0.5*logCapex^2) ,
               data = Front_sfa_r, ineffDecrease = FALSE )

####--- Tjek om monotonicitet er brudt - frontelskaber ----####
# 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)

####--- Tjek om monotonicitet er brudt - kun renseselskaber (dem
der får score fra modellen) ----####
# OPEX
table((coef(sfaEst_r)[2]+coef(sfaEst_r)[4]*BM_Data$logOpex[which(Fr
ont_sfa$TYPE=="r")]+coef(sfaEst_r)[5]*BM_Data$logCapex[which(Front_
sfa$TYPE=="r")]) <0)
# CAPEX
table((coef(sfaEst_r)[3]+coef(sfaEst_r)[6]*BM_Data$logCapex[which(F
ront_sfa$TYPE=="r")]+coef(sfaEst_r)[5]*BM_Data$logOpex[which(Front_
sfa$TYPE=="r")]) <0)

####--- Tjek om monotonicitet er brudt - kun renseselskaber (dem
der får score fra modellen) ----####
# OPEX
table((coef(sfaEst_r)[2]+coef(sfaEst_r)[4]*BM_Data$logOpex[which(Da
ta$TYPE=="r")]+coef(sfaEst_r)[5]*BM_Data$logCapex[which(Data$TYPE==
"r")]) <0)
# CAPEX
table((coef(sfaEst_r)[3]+coef(sfaEst_r)[6]*BM_Data$logCapex[which(D
ata$TYPE=="r")]+coef(sfaEst_r)[5]*BM_Data$logOpex[which(Data$TYPE==
"r")]) <0)

####--- Cook's distance til unrestricted model ----####
Front_sfa_r$PCD_eff_r<-cooks.distance(sfaEst_r,target =
"efficiencies", asInData = TRUE)

# Scorer for selskaber af typen "r" (Selskaber af typen "t+r" samt
"r" indgår i fronten)
sfa_r <- efficiencies(sfaEst_r, newdata = BM_Data)

#-----#
####--- 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)

#summary(sfaEst_tr)
#summary(sfaEst_t)
#summary(sfaEst_r)
```

2.5 Samlet oversigt over scorer

```
####---- Dataframe med endelige DEA- og SFA-scorer ----####
Data_scorer <- Data[ ,c("NAVN","ID","dea_score","sfa_score")]
```

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 3 – Costdriveranalyse](#) beskrives metoden for analysen samt resultaterne for regressionerne.

3.2 Kode for costdriversammensætning

```
#=====#
#### Costdriveranalyse til benchmarking ####
#=====#

####---- Indlæsning af data ----####
library(readxl)
Data <- read_xlsx("Bilag 1 - Data til brug for fastlæggelse af de
individuelle effektiviseringskrav.xlsx", sheet = "Til R-koder")

# Fjerner selskaber der ikke har indberettet til benchmarking
Data <- Data[!Data$ID=="S088",]

####----Udregner costdriverandelene----####
#OPEX costdrivere
Data$Ledninger_And <- Data$Ledninger/Data$OPEX_U
Data$Pumpe_And <- Data$Pumpe/Data$OPEX_U
Data$Regnbas_And <- Data$Regnbas/Data$OPEX_U
Data$Spildebas_And <- Data$Spildebas/Data$OPEX_U
Data$Rense_And <- Data$Rense/Data$OPEX_U
Data$Minirenses_And <- Data$Minirenses/Data$OPEX_U
Data$Behandling_And <- Data$Behandling/Data$OPEX_U
Data$Disponering_And <- Data$Disponering/Data$OPEX_U
Data$Kunder_And <- Data$Kunder/Data$OPEX_U
Data$Adm_And <- Data$Adm/Data$OPEX_U

#CAPEX costdrivere
Data$Prod_And <- Data$Prod / Data$CAPEX_U
Data$Dist_And <- Data$Dist / Data$CAPEX_U
Data$Faelles_And <- Data$Faelles / Data$CAPEX_U

#=====#
#### Regressioner for OPEX-costdrivere ####
#=====#

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

## Sammenhæng mellem ledninger og efficiensscoren
LmLedninger <- lm(BoT~Ledninger_And,data = Data)
summary(LmLedninger) # Endelig regressionsmodel

# Cooks distance - ledninger
plot(sort(cooks.distance(LmLedninger)))
CD_ledninger<-cbind(Data$NAVN,Data$ID,cooks.distance(LmLedninger))
```

```
CD_ledninger[order(cooks.distance(LmLedninger)),] # Ingen selskaber
fjernes som outlier
```

```
#-----#
###---- Pumpestationer ----###
#-----#
## Sammenhæng mellem pumpestationer og efficiensscoren
LmPumpe <- lm(BoT~Pumpe_And,data = Data)
summary(LmPumpe) # Endelig regressionsmodel

# Cooks distance - Pumpestationer
plot(sort(cooks.distance(LmPumpe)))
CD_pumpestationer<-cbind(Data$NAVN,cooks.distance(LmPumpe))
CD_pumpestationer[order(cooks.distance(LmPumpe)),] # Ingen
selskaber fjernes som outlier
```

```
#-----#
###---- Regnvandsbassiner ----###
#-----#
## Sammenhæng mellem regnvandsbassiner og efficiensscoren
LmRegnbas <- lm(BoT~Regnbas_And,data = Data)
summary(LmRegnbas) # Endelig regressionsmodel

# Cooks distance - regnvandsbassiner
plot(sort(cooks.distance(LmRegnbas)))
CD_regnbas<-cbind(Data$NAVN, Data$ID,cooks.distance(LmRegnbas))
CD_regnbas[order(cooks.distance(LmRegnbas)),]
```

```
#-----#
###---- Spildevandsbassiner ----###
#-----#
## Sammenhæng mellem spildevandsbassiner og efficiensscore
LmSpildebas <- lm(BoT~Spildebas_And,data = Data)
summary(LmSpildebas) # Endelig regressionsmodel

# Cooks distance - Spildevandsbassiner
plot(sort(cooks.distance(LmSpildebas)))
CD_spildebas<-cbind(Data$NAVN, Data$ID,cooks.distance(LmSpildebas))
CD_spildebas[order(cooks.distance(LmSpildebas)),] # Ingen selskaber
fjernes som outlier
```

```
#-----#
###---- Renseanlæg ----###
#-----#
## Sammenhæng mellem renseanlæg og efficiensscore
LmRense <- lm(BoT~Rense_And,data = Data)
summary(LmRense) # Endelig regressionsmodel

# Cooks distance - Renseanlæg
plot(sort(cooks.distance(LmRense)))
CD_rense<-cbind(Data$NAVN, Data$ID,cooks.distance(LmRense))
CD_rense[order(cooks.distance(LmRense)),] # Ingen selskaber fjernes
som outlier
```

```
#-----#
###---- Minirenseanlæg ----###
#-----#
## Sammenhæng mellem minirenseanlæg og efficiensscore
```

```

LmMinirensen <- lm(BoT~Minirensen_And,data = Data)
summary(LmMinirensen) # Endelig regressionsmodel

# Cooks distance - Minirensen anlæg
plot(sort(cooks.distance(LmMinirensen)))
CD_minirensen<-cbind(Data$NAVN, Data$ID,cooks.distance(LmMinirensen))
CD_minirensen[order(cooks.distance(LmMinirensen)),] # Ingen selskaber
fjernes som outlier

#-----#
###---- Slambehandling ----###
#-----#
## Sammenhæng mellem slambehandling og efficiensscore
LmBehandling <- lm(BoT~Behandling_And,data = Data)
summary(LmBehandling) # Endelig regressionsmodel

# Cooks distance - Slambehandling
plot(sort(cooks.distance(LmBehandling)))
CD_behandling<-cbind(Data$NAVN,
Data$ID,cooks.distance(LmBehandling))
CD_behandling[order(cooks.distance(LmBehandling)),] # Ingen
selskaber fjernes som outlier

#-----#
###---- Slamdisponering ----###
#-----#
## Sammenhæng mellem slamdisponering og efficiensscore
LmDisponering <- lm(BoT~Disponering_And,data = Data)
summary(LmDisponering) # Endelig regressionsmodel

# Cooks distance - Slambehandling
plot(sort(cooks.distance(LmDisponering)))
CD_disponering<-cbind(Data$NAVN,
Data$ID,cooks.distance(LmDisponering))
CD_disponering[order(cooks.distance(LmDisponering)),] # Ingen
selskaber fjernes som outlier

#-----#
###---- Kunder ----###
#-----#
## Sammenhæng mellem kunder og efficiensscore
LmKunder <- lm(BoT~Kunder_And,data = Data)
summary(LmKunder) # Endelig regressionsmodel

# Cooks distance - Kunder
plot(sort(cooks.distance(LmKunder)))
CD_kunder<-cbind(Data$NAVN, Data$ID,cooks.distance(LmKunder))
CD_kunder[order(cooks.distance(LmKunder)),] # Ingen selskaber
fjernes som outlier

#-----#
###---- Generel administration ----###
#-----#
## Sammenhæng mellem generel administration og efficiensscore
LmAdm <- lm(BoT~Adm_And,data = Data)
summary(LmAdm) # Endelig regressionsmodel

# Cooks distance - Generel administration
plot(sort(cooks.distance(LmAdm)))
CD_behandling<-cbind(Data$NAVN, Data$ID,cooks.distance(LmAdm))

```

```

CD_behandling[order(cooks.distance(LmAdm)),] # Ingen selskaber
fjernes som outlier

#####
### Regressioner for CAPEX-costdrivere ###
#####

#-----#
###---- Produktionsanlæg ----###
#-----#
## Sammenhæng mellem produktionsanlæg og efficiensscore
LmProd <- lm(BoT~Prod_And,data = Data)
summary(LmProd) # Endelig regressionsmodel

# Cooks distance - Generel administration
plot(sort(cooks.distance(LmProd)))
CD_prod<-cbind(Data$NAVN, Data$ID,cooks.distance(LmProd))
CD_prod[order(cooks.distance(LmProd)),] # Ingen selskaber fjernes
som outlier

#-----#
###---- Distributionsanlæg ----###
#-----#
## Sammenhæng mellem distributionsanlæg og efficiensscore
LmDist <- lm(BoT~Dist_And,data = Data)
summary(LmDist) # Endelig regressionsmodel

# Cooks distance - distributionsanlæg
plot(sort(cooks.distance(LmDist)))
CD_dist<-cbind(Data$NAVN, Data$ID,cooks.distance(LmDist))
CD_dist[order(cooks.distance(LmDist)),] # Ingen selskaber fjernes
som outlier

#-----#
###---- Fællesfunktionsanlæg ----###
#-----#

## Sammenhæng mellem fællesfunktionsanlæg og efficiensscore - inden
outlier fjernes
LmFaelles <- lm(BoT~Faelles_And,data = Data)
summary(LmFaelles)

# Cooks distance - Fællesfunktionsanlæg - inden outlier fjernes
plot(sort(cooks.distance(LmFaelles)))
CD_faelles<-cbind(Data$NAVN, Data$ID,cooks.distance(LmFaelles))
CD_faelles[order(cooks.distance(LmFaelles)),]

## Regressionsmodel for fællesfunktionsanlæg uden outlier
Faelles_outlier<-"S101" # Outlier defineres - Vejen Renseanlæg
LmFaelles_u_out <- lm(BoT~Faelles_And,data = Data[-which(Data$ID ==
Faelles_outlier),])
summary(LmFaelles_u_out) # Endelig regressionsmodel

# Cooks distance - fællesfunktionsanlæg - for model uden outlier
plot(sort(cooks.distance(LmFaelles_u_out)))
CD_faelles_u_out<-cbind(Data$NAVN[-which(Data$ID %in%
Faelles_outlier)],Data$ID[-which(Data$ID %in%
Faelles_outlier)],cooks.distance(LmFaelles_u_out))
CD_faelles_u_out[order(cooks.distance(LmFaelles_u_out)),]

```

```
## Endelig regressionsmodel for fællesfunktionsanlæg uden outlier
Faelles_outlier<-c("S101","S055") # Outlier defineres - Vejen
Renseanlæg og Kalundborg Renseanlæg A/S
LmFaelles_u_out <- lm(BoT~Faelles_And,data = Data[-which(Data$ID
%in% Faelles_outlier),])
summary(LmFaelles_u_out) # Endelig regressionsmodel

# Cooks distance - fællesfunktionsanlæg - for endelig model uden
outlier
plot(sort(cooks.distance(LmFaelles_u_out)))
CD_faelles_u_out<-cbind(Data$NAVN[-which(Data$ID %in%
Faelles_outlier)],Data$ID[-which(Data$ID %in%
Faelles_outlier)],cooks.distance(LmFaelles_u_out))
CD_faelles_u_out[order(cooks.distance(LmFaelles_u_out)),]
```