



Bilag 6 – R-koder

ANALYSE

DEN 24. APRIL 2020

FORSYNINGSTILSYNET
Torvegade 10
3300 Frederiksværk

Tlf. 4171 5400
post@forsyningstilsynet.dk
www.forsyningstilsynet.dk

Indhold

1	INDLEDNING.....	3
2	STATISTIKPROGRAMMET R.....	3
	TILFØJELSESPROGRAMMET R-STUDIO.....	3
3	FORSYNNINGSTILSYNETS R-KODER.....	3
	BENCHMARKINGMODEL.R.....	4
	DATAGRUNDLAG.R.....	9
	PAKKER.R.....	10
	FUNKTIONER.R.....	10
	DEA-OUTLIER.R.....	11
	TEST FOR SKALAAFKAST.R.....	12
	SFA-TEST.R.....	13

1 INDLEDNING

Dette bilag redegør for beregningsprogrammet, R, som anvendes til at beregne fjernvarmevirksomhedernes effektiviseringspotentiale og beskriver de koder, der anvendes i R til at beregne benchmarkingmodellen.

Til benchmarking af fjernvarmevirksomhedernes økonomiske potentialer frem mod 2030 har Forsyningstilsynet valgt at anvende metoder for beregning af fjernvarmevirksomhedernes effektiviseringspotentiale benævnt henholdsvis Data Envelopment Analysis (DEA) og Stochastic Frontier Analysis (SFA), som er fagligt anerkendte metoder, der anvendes blandt nationale og internationale regulatorer.

Metoderne bygger på regnetekniske formler, som ikke er en del af standardpakken i Microsoft Excel. Derfor har Forsyningstilsynet valgt at anvende R til at beregne effektiviseringspotentialerne.

2 STATISTIKPROGRAMMET R

Programmet R er et gratis software, som kan downloades på internettet via følgende hjemmeside: <https://www.r-project.org/>.

Statistikprogrammets største fordel er mulighederne for databehandling og analyser, som er nødvendige i den anvendte metode for benchmarking.

Programmet har dog en ulempe med hensyn til den visuelle præsentation, hvor Excel ofte er bedre til at give et dataoverblik og til at illustrere, hvordan talværdierne bliver summeret. For at sikre den størst mulig gennemsigtighed i metoden for benchmarking vil Forsyningstilsynet udarbejde en Excel-projektmappe, der indeholder beregninger af fjernvarmevirksomhedernes totalomkostninger samt af de enkelte ydelser anvendt i modellen.

TILFØJELSESPROGRAMMET R-STUDIO

Forsyningstilsynet har valgt at anvende et tilføjesprogram kaldet for R-studio. Tilføjesprogrammet R-studio er et gratis software, som kan downloades på internettet via følgende hjemmeside: <https://rstudio.com/>.

Det skal bemærkes, at R-studio er et tilføjesprogram, som ikke kan installeres alene. Statistikprogrammet R skal installeres først, og dernæst kan R-studio installeres og tilføjes som en udvidelse til R.

I de R-koder, som Forsyningstilsynet har udarbejdet, kan der være enkelte kommandoer, som ikke bliver læst ens i henholdsvis R og R-studio. Derfor anbefaler Forsyningstilsynet, at udvidelsesprogrammet R-studio anvendes, når R-koder skal læses.

3 FORSYNNINGSTILSYNETS R-KODER

Forsyningstilsynets R-koder (R-script) består af syv individuelle R-koder:

- Benchmarkingmodel.R
- Datagrundlag.R
- Pakker.R
- Funktioner.R
- DEA-outliers.R
- Test for skalaafkast.R
- SFA-test.R

BENCHMARKINGMODEL.R

Denne R-kode indeholder de primære koder til Forsyningstilsynets benchmarkingmodel. For at kunne køre hele benchmarkingmodellen er det vigtigt, at alle øvrige R-koder er placeret i samme mappe på computeren.¹ Dette skyldes, at koden indhenter kommandoer og resultater fra de øvrige R-koder, hvilket skulle gøre de primære koder lettere at læse, hvis brugeren ikke ønsker at gå i dybden med hele modellen. For dem, der ønsker at kigge dybere ned i modellens bagvedliggende antagelser, kan de øvrige R-koder åbnes selvstændigt.

Når koden køres, skal R-programmet have instruktion om, hvor arbejdsmappen er placeret på computeren. Arbejdsmappen er den mappe, hvori alle R-koder og alt datamateriale er gemt. Når koden med "setwd(choose.dir())" bliver kørt, vil R automatisk bede brugeren om at angive, hvor arbejdsmappen er placeret på computeren. Bemærk, at der kan være enkelte tilfælde hvor boksen, som skal anvendes til at angive placeringen, ikke åbner op automatisk, men i stedet blinker nede i Windows' Taskbar.

```

1  #####-----#####
2  #####-----  FORSYNINGSTILSYNETS SCRIPT TIL BENCHMARKING  -----#####
3  #####-----#####
4
5
6  #####      Først renses data, værdier og funktioner i R
7  rm( list=ls() )
8
9  setwd( choose.dir() )
10
11 #-----  GENERELLE INDSTILLINGER  -----#
12
13 SEKTOR <- "PRODUKTION"  # vælg enten PRODUKTION eller TRANSPORT
14 # vælg enten at segmentere eller ej:
15 SEGMENT <- "PRODUKTION"
16 #Hvis der ikke skal segmenteres, skriv da PRODUKTION eller TRANSPORT
17 # For Produktion kan der segmenteres på brændsler: GRØN eller FOSSIL
18 # For Transport kan der segmenteres på bymæssighed: STORBY eller ANDET
19 DEA_RETURNS_TO_SCALE<-"vrs" # vælg fx vrs, crs eller irs
20 F_cut_off <- 0.2 #vælg et cutof for F-test
21
22
23 #-----  RELEVANTE DATA, PAKKER OG FUNKTIONER  -----#
24

```

¹ Det samme gør sig også gældende for alt datamateriale

```

25 ##### Her indlæses datamaterialet,
26 #herefter installeres Benchmarking-pakke og funktioner
27 source("Datagrundlag.r") ; data_basis <- datagrundlag[,]
28
29 ##### Reduceres data yderligere hvis der segmenteres
30 if (SEGMENT=="GRØN"){
31 data_basis = subset(data_basis, Biobraendsel==1|
32 IndustrielOverskudsvarme==1|Andet==1)}
33 if (SEGMENT=="FOSSIL"){
34 data_basis = subset(data_basis, Naturgas==1|Olie==1|Kul==1)}
35 if (SEGMENT=="STORBY"){
36 data_basis = subset(data_basis, Storby==1 )}
37 if (SEGMENT=="ANDET"){
38 data_basis = subset(data_basis, AndenBymaessighed==1|Barmarksvaerk==1 )}
39
40 source("Pakker.R")
41 source("Funktioner.R")
42
43
44 #----- MODELOPSETNING -----#
45
46 ##### Omkostningsgrundlaget og ydelser defineres
47 if (SEKTOR=="PRODUKTION"){
48 #----- Produktionsmodel -----#
49 omkostninger <- c("Produktion_MedProportionalFaellesAdministration_gns")
50 modelydelser <- c("SamletProduktionAfVarmeMWh_gns",
51 "VarmeProduktionsKapacitetKorrektionMW_gns" ,
52 "SamletProduktionElMWh_gns","ElProduktionsKapacitetKorrektionMW_gns")
53
54 ##### Outlier grupperinger
55 PROD_SPIDSBELASTNING <- c("Vestforsyning Varme A/S",
56 "ANDELSSELSKABET MØLHOLM VARMEVÆRK",
57 "HIRTSHALS FJERNVARME",
58 "Padborg Fjernvarme A.m.b.a.",
59 "TRE-FOR VARME, KOLDING A/S",
60 "I/S Norfors Fjernvarme",
61 "Nyborg Forsyning Varme A/S",
62 "Solrød Fjernvarme a m b a",
63 "NÆSTVED VARMEVÆRK",
64 "GANDRUP - VESTER-HASSING VARMEVÆRK A M B A",
65 "Kerteminde Forsyning - Varme A/S",
66 "STRUER FORSYNING FJERNVARME A/S",
67 "Aalborg Varme A/S",
68 "Guldborgsund Varme",
69 "CTR I/S",
70 "FREDERICIA FJERNVARME A.M.B.A",
71 "VarmePlan Aarhus",
72 "Rønne Varme A/S",
73 "Gauerslund Fjernvarme",
74 "ESBJERG VARME A/S",
75 "NORDBY FJERNVARME A M B A",
76 "HOFOR Fjernvarme P/S",
77 "Hornbæk Fjernvarme A.m.b.a.",
78 "GIVE FJERNVARME A M B A",
79 "Kyndbyværket")
80
81 PROD_AFFALD <- c( "Energnist - Esbjerg",
82 "Fjernvarme Fyn Affaldsenergi A/S",
83 "GRENAA FORBRÆNDING A/S",
84 "Hammel Fjernvarme Amba",

```

```
85         "Fjernvarme Horsens - Horsens Kraftvarmeværk A/S",
86         "Energnist Kolding",
87         "Maabjerg Energy Center - BioHeat&Power (Måbjergværket)",
88         "I/S Norfors affaldsværk",
89         "RENO NORD I/S",
90         "SVENDBORG KRAFTVARME A/S",
91         "Sønderborg Kraftvarmeværk I/S",
92         "I/S Kraftvarmeværk Thisted",
93         "I/S Vestforbrænding Glostrup",
94         "Aars Fjernvarme Amba",
95         "AVA Varme")
96
97 PROD_SÆRLIGE_VÆRKER <- c( "FJERNVARME FYN PRODUKTION A/S - Blok 7")
98
99 PROD_UNDER_50_TJ <- c( "Arden Varmeværk",
100         "Balling Fjernvarmeværk Amba",
101         "ESBJERG VARME A/S",
102         "FREDERICIA FJERNVARME A.M.B.A",
103         "GANDRUP - VESTER-HASSING          VARMEVÆRK A M B A",
104         "Gauerslund Fjernvarme",
105         "GIVE FJERNVARME A M B A",
106         "Odsherred Varme A/S",
107         "Guldborgsund Varme",
108         "HIRTSHALS FJERNVARME",
109         "HOFOR Fjernvarme P/S",
110         "Hornbæk Fjernvarme A.m.b.a.",
111         "Hvalsø Kraftvarmeværk a.m.b.a.",
112         "Kerteminde Forsyning - Varme A/S",
113         "Kyndbyværket",
114         "Laurbjerg Kraftvarmeværk AMBA",
115         "NORDBY FJERNVARME A M B A",
116         "NR ALSLEV FJERNVARMEVÆRK A M B A",
117         "Nyborg Forsyning Varme A/S",
118         "NÆSTVED VARMEVÆRK",
119         "Padborg Fjernvarme A.m.b.a.",
120         "Rønne Varme A/S",
121         "Solrød Fjernvarme a m b a",
122         "TRE-FOR VARME, KOLDING A/S",
123         "VORDINGBORG FJERNVARME A/S",
124         "Værum-Ørum Kraftvarmeværk A.m.b.a.",
125         "ØSTERLUND VARME A/S")
126
127 PROD_SÆRLIGE_AFSKRIVNINGER <- c( "#Disse outliers kan ikke offentligøres
128                                   #på nuværende tidspunkt")
129
130 PROD_SFA_OUTLIERS<- c("Ikast Værkerne Varme A/S - Bording Kraftvarme
131 værk",
132         "EnergiGruppen Jylland Varme",
133         "Hjallerup Fjernvarmeværk",
134         "HOLTE FJERNVARME A M B A",
135         "Maabjerg Energy Center - BioGas A/S",
136         "NRGi Lokalvarme A/S",
137         "NØRRE NEBEL FJERNVARME",
138         "Refa, Stubbekøbing Fjernvarme A/S",
139         "Svanemølleværket",
140         "Thisted Varmeforsyning Amba",
141         "Tønder Fjernvarmeselskab A.m.b.a.",
142         "VORDINGBORG FJERNVARME A/S",
143         "Helsingør Kraftvarmeværk A/S",
144         "Hjallerup Fjernvarmeværk",
```



```
205 ##### Reducerer datamængden til kun det relevante data
206 data_alle <- data_basis[ , c("Pnummer", "CVR" , "Vaerk" ,
207 omkostninger , modelydelser) ]
208
209 ##### Outliers fjernes
210 if (SEKTOR=="PRODUKTION"){
211 data_alle <- data_alle[!data_alle$Vaerk%in%PROD_SPIDSBELASTNING,]
212 data_alle <- data_alle[!data_alle$Vaerk%in%PROD_AFFALD,]
213 data_alle <- data_alle[!data_alle$Vaerk%in%PROD_SÆRLIGE_VÆRKER,]
214 dea_front <- data_alle[!data_alle$Vaerk%in%PROD_UNDER_50_TJ,]
215 dea_front <- data_alle[!data_alle$Vaerk%in%PROD_SÆRLIGE_AFSKRIVNINGER,]
216 dea_front <- data_alle[!data_alle$Vaerk%in%PROD_SFA_OUTLIERS,]
217 }
218 if (SEKTOR=="TRANSPORT"){
219 data_alle <- data_alle[!data_alle$Vaerk%in%TRANS_TRANSMISSION,]
220 data_alle <- data_alle[!data_alle$Vaerk%in%TRANS_GARTNERE,]
221 data_alle <- data_alle[!data_alle$Vaerk%in%TRANS_SÆRLIGE_VÆRKER,]
222 dea_front <- data_alle[!data_alle$Vaerk%in%TRANS_SÆRLIGE_AFSKRIVNINGER,]
223 dea_front <- data_alle[!data_alle$Vaerk%in%TRANS_SFA_outliers,]
224 }
225
226 ##### Endeligt datasæt
227 data <- data_alle
228
229
230 #----- DEA-MODELLEN KØRES -----#
231
232 ##### Test for repræsentative fjernvarmevirksomheder findes i
233 #source-koden
234 source("DEA-outlier.R") ; dea_test
235 forste_kvartil=quantile(sdea_test,probs = 0.25)
236 tredje_kvartil=quantile(sdea_test,probs = 0.75)
237 # Formel: 3. kvartil + 1.5 * (3. kvartil - 1 kvartil)
238 Super_eff_cut_off <- tredje_kvartil + 1.5*(tredje_kvartil-forste_kvartil)
239 outliers <- (dea_test$F_test <=F_cut_off |
240 dea_test$sdea_test >=Super_eff_cut_off)
241
242 ##### Banker-test for skalaafkast
243 source("Test for skalaafkast.R")
244 dea_front <- dea_front[c(!outliers),]
245
246 ##### DEA regnes for alle repræsentative fjernvarmevirksomheder
247 dea <- dea( X = data[ , omkostninger ] ,
248 Y = data[ , modelydelser ] ,
249 XREF = dea_front[ , omkostninger ] ,
250 YREF = dea_front[ , modelydelser ] ,
251 ORIENTATION = "in" , RTS = DEA_RETURNS_TO_SCALE )
252
253 ##### Resultaterne fra DEA normaliseres og gemmes
254 dea_res <- as.matrix( ifelse( dea$eff > 1 , 1 , dea$eff ) )
255
256
257 #----- SFA-MODELLEN KØRES -----#
258 ##### Cutoff for cooks udregnes
259 # Formel: 4/(n-4-1)
260 if (SEGMENT=="PRODUKTION"){
261 sfacutoff <- 0.02325581
262 }
263 if (SEGMENT=="GRØN"){
```



```

264 sfacutoff <- 0.04210526
265 }
266 if (SEGMENT=="FOSSIL"){
267 sfacutoff <- 0.05555556
268 }
269 if (SEGMENT=="TRANSPORT"){
270 sfacutoff <- 0.02234637
271 }
272 if (SEGMENT=="STORBY"){
273 sfacutoff <- 0.2222222
274 }
275 if (SEGMENT=="ANDET"){
276 sfacutoff <- 0.02564103
277 }
278
279 ##### Fronten til log-linær-model findes i source-koden
280 source("SFA-test.r") ;
281 cookplot_log[order(cooks_log),]
282 plot(sort(cooks.distance(ols_log), decreasing = FALSE),
283 main = "Cooks Distance i SFA-modellen",ylab = "Cooks Distance" )
284
285 ##### Resultaterne fra SFA i loglineær beregnes og gemmes
286 sfa_log <- te.mul.sfa.all( X = as.matrix( -sfa_output_alle_log ) ,
287 Y = as.matrix( -sfa_input_alle_log ) ,
288 REF = Benchmarking::sfa( as.matrix( -sfa_front_output_log ) ,
289 as.matrix( -sfa_front_input_log ) ) )
290
291 #----- DEA- OG SFA-RESULTATER SAMLES -----#
292
293 ##### Potentialer beregnes som bedst-af-to enten af DEA-modellen
294 #eller SFA-modellen
295 pot <- 1-(pmax(dea_res,sfa_log)) ; pot[pot=="NaN"] <- 0
296 resultater <- cbind(data[,c("Vaerk","CVR")],dea_res,sfa_log,pot)
297 resultater
298
299 #----- MAKSIMALE POTENTIALE I MODELLEN -----#
300 kvatil=quantile(resultater[!resultater[,"dea_res"]==1,"pot"],
301 probs = 0.75)
302 loft <- rep(kvatil,length(data$Vaerk))
303 resultat <- pmin(pot,loft)
304
305 ##### Vægtet potentiel for hele branchen
306 if (SEKTOR=="PRODUKTION"){
307 weighted.mean(resultat,
308 data_alle$Produktion_MedProportionalFaellesAdministration_gns,
309 na.rm = FALSE)
310 }
311 if (SEKTOR=="TRANSPORT"){
312 weighted.mean(resultat,
313 data$Transport_MedProportionalFaellesAdministration_gns,
314 na.rm = FALSE)
315 }

```

DATAGRUNDLAG.R

For at gøre selve benchmarkingmetoden letlæselig og gennemskuelig har Forsynings-tilsynet valgt at foretage datakonstruktioner i en særskilt kode kaldet 'Datagrundlag.R'.

I denne kode indlæses data fra en csv.fil. Filen indeholder forskellige variabler for fjernvarmevirksomhedernes økonomiske og tekniske data samt virksomhedernes totalomkostninger (TOTEX), som er beregnet i overensstemmelse med bilag 2 om omkostningsgrundlag.

```

1  #####-----#####
2  #####-----  FORSYNINGSTILSYNETS SCRIPT TIL BENCHMARKING  -----#####
3  #####-----#####
4
5
6  #----- RÅDATA INDLÆSES -----#
7  if (SEKTOR=="PRODUKTION"){
8  indberetning <- read.csv( "DatasetRProd_200320.csv" , sep = ";" )}
9  if (SEKTOR=="TRANSPORT"){
10 indberetning <- read.csv( "DatasetRTrans_230320.csv" , sep = ";" )}
11
12 #----- SÆRLIGE VÆRDIER ERSTATTES -----#
13 indberetning[is.na(indberetning)] <- 0
14
15 #----- DATA MERGES SAMMEN -----#
16 datagrundlag <- indberetning

```

PAKKER.R

I standardindstillingerne i R og R-studio er det ikke muligt at foretage de beregninger, som skal anvendes for at foretage en beregning af en DEA-model. Benchmarkingmodellen kan først køre, når de korrekte tilføjelsespakker er downloadet, installeret og aktiveret i R. Alle nødvendige pakker er derfor samlet i denne kode, og når koden bliver udført, vil R automatisk tilføje de pakker, som er nødvendige for at køre Forsyningstilsynets benchmarkingmodel i R.

```

1  #####-----#####
2  #####-----  FORSYNINGSTILSYNETS SCRIPT TIL BENCHMARKING  -----#####
3  #####-----#####
4
5
6  #----- PAKKER DOWNLOADES OG INSTALLERES/OPDATERES -----#
7  #####-----#####
8  install.packages("Benchmarking")
9  install.packages("AER")
10 install.packages("ggplot2")
11
12
13 #----- INSTALLEREDE PAKKER AKTIVERES -----#
14 #####-----#####
15 library(Benchmarking)
16 library(AER)
17 library(ggplot2)

```

FUNKTIONER.R

Denne koder indeholder funktioner, som Forsyningstilsynet har udarbejdet specifikt til at foretage beregninger i benchmarkingmodellen, fordi flere af de regneoperationer,

som er nødvendige, fx fastfrysning af fronten, ikke er en del af hverken standard R-programmet eller de pakker, som bliver aktiveret i Pakker.R.

```

1  #####-----#####
2  #####-----  FORSYNINGSTILSYNETS SCRIPT TIL BENCHMARKING  -----#####
3  #####-----#####
4
5
6  #-----  Fast SFA-front til den additive model (LOG-LINEÆR)  -----#
7  te.mul.sfa.all <- function(X,Y,REF)
8  { lambda <- REF$lambda
9    s2 <- REF$sigma2
10 coefficients <- as.matrix(REF$coef,ncol=1)
11 K <- length(Y)
12 x_constant_all <- as.matrix(cbind(rep(1,K),X))
13 residuals_all <- Y-x_constant_all %*% t(coefficients)[1,]
14 ustar <- -residuals_all * lambda^2/(1 + lambda^2)
15 sstar <- lambda/(1 + lambda^2) * sqrt(s2)
16 te.mul = pnorm(ustar/sstar - sstar)/pnorm(ustar/sstar)
17 * exp(sstar^2/2-ustar)
18 class(te.mul) <- "matrix"
19 colnames(te.mul) <- "te.mul"
20 return(array(te.mul))

```

DEA-OUTLIER.R

I benchmarkingmetoden bliver den estimerede DEA-front testet for, om der er fjernvarmevirksomhederne med en kraftig påvirkning af den målte effektivitet for hele branchen. Denne R-kode foretager disse tests og indeholder en test af signifikant påvirkning samt en test af superefficiens. Det skal bemærkes, at koderne ikke indeholder en mekanisk udelukkelse af fjernvarmevirksomheder fra fronten, men at resultaterne af de tests, som bliver kørt, danner grundlag for en konkret vurdering af, om enkelte fjernvarmevirksomheder skal udelukkes fra fronten.

```

1  #####-----#####
2  #####-----  FORSYNINGSTILSYNETS SCRIPT TIL BENCHMARKING  -----#####
3  #####-----#####
4
5
6  #-----  DATA FOR REPRÆSENTATIVE FJERNVARMEVIRKSOMHEDER INDLÆSES  -----#
7  x <- as.matrix( dea_front[ omkostninger ] )
8  y <- as.matrix( dea_front[ modelydelser ] )
9  dea_front <- dea_front
10
11 #-----  BEREGN F_TEST SCORER  -----#
12 F_test <- NULL
13 for (i in 1:dim( x )[1])
14 { A <- dea( x[ -i , ] , y[ -i , ] , ORIENTATION="in" ,
15 RTS=DEA_RETURNS_TO_SCALE )$eff
16 B <- dea( x , y , ORIENTATION="in" ,
17 RTS=DEA_RETURNS_TO_SCALE )$eff[ -i ]
18 C <- pf( sum((1-A)^2)/sum((1-B)^2) , dim( x )[1] - 1 , dim( x )[1] - 1 )
19 F_test <- rbind( F_test , C )
20
21 #-----  SUPEREFFICIENS  -----#
22 sdea_test <- sdea( x , y , ORIENTATION="in" ,
23 RTS=DEA_RETURNS_TO_SCALE )$eff

```

```

24
25 #----- TESTRESULTATER SAMLES -----#
26 dea_test <- cbind.data.frame(dea_front[c("Vaerk")],F_test,sdea_test)

```

TEST FOR SKALAAFKAST.R

I denne kode foretages en Banker-test af, om der statistisk kan eftervises konstant skalaafkast i DEA-modellen. Testene foretages for varierende, stigende og konstant skalaafkast. I testen skal der løbende vurderes, om der er outliers i hver af modellerne, dvs. at det ikke nødvendigvis er de samme fjernvarmevirksohederne, der er outliers i modellen med konstant skalaafkast som i modellerne med henholdsvis stigende eller varierende skalaafkast.

```

1 #####-----#####
2 #####----- FORSYNINGSTILSYNETS SCRIPT TIL BENCHMARKING -----#####
3 #####-----#####
4
5
6 #----- DATA FOR REPRESENTATIVE FJERNVARMEVIRKSOMHEDER INDLÆSES -----#
7 x <- as.matrix( dea_front[ omkostninger ] )
8 y <- as.matrix( dea_front[ modelydelser ] )
9
10 #----- OUTLIER-TESTS FOR DE TRE MODELSPECIFIKATIONER -----#
11 for (rts in c("crs","irs","vrs"))
12 { F_test <- NULL
13   for (i in 1:dim( x )[1])
14   { A <- dea( x[ -i , ] , y[ -i , ] , ORIENTATION="in" , RTS=rts )$eff
15     B <- dea( x , y , ORIENTATION="in" , RTS=rts )$eff[ -i ]
16     C <- pf( sum((1-A)^2)/sum((1-B)^2) , dim( x )[1] - 1 , dim( x )[1] - 1 )
17     F_test <- rbind( F_test , C ) }
18   sdea_test <- sdea( x , y , ORIENTATION="in" , RTS=rts )$eff
19   assign( paste("outliertest_",rts,sep="" ) ,
20     cbind.data.frame(dea_front[c("Vaerk")],F_test,sdea_test))}
21
22 colnames( outliertest_crs ) <- c("Vaerk","F_test_crs","sdea_test_crs")
23 ; colnames( outliertest_irs ) <- c("Vaerk","F_test_irs","sdea_test_irs")
24 ; colnames( outliertest_vrs ) <- c("Vaerk","F_test_vrs","sdea_test_vrs")
25 merge(outliertest_crs,merge(outliertest_irs,outliertest_vrs))
26
27 #----- OUTLIERS I DE TRE MODELSPECIFIKATIONER FJERNES -----#
28 outliers_in_RTS_test_crs <- (outliertest_crs$F_test_crs <=F_cut_off
29   | outliertest_crs$sdea_test_crs >=Super_eff_cut_off)
30 outliers_in_RTS_test_irs <- (outliertest_irs$F_test_irs <=F_cut_off
31   | outliertest_irs$sdea_test_irs >=Super_eff_cut_off)
32 outliers_in_RTS_test_vrs <- (outliertest_vrs$F_test_vrs <=F_cut_off
33   | outliertest_vrs$sdea_test_vrs >=Super_eff_cut_off)
34
35 data_test_crs <- dea_front[!outliers_in_RTS_test_crs,]
36 data_test_irs <- dea_front[!outliers_in_RTS_test_irs,]
37 data_test_vrs <- dea_front[!outliers_in_RTS_test_vrs,]
38
39 #----- RESULTATER FRA DE TRE MODELSPECIFIKATIONER -----#
40 dea_crs <- dea(data_test_crs[,omkostninger]
41 ,data_test_crs[,modelydelser],ORIENTATION="in",RTS="crs")
42 dea_irs <- dea(data_test_irs[,omkostninger]
43 ,data_test_irs[,modelydelser],ORIENTATION="in",RTS="irs")
44 dea_vrs <- dea(data_test_vrs[,omkostninger]

```

```

45 ,data_test_vrs[,modelydelser],ORIENTATION="in",RTS="vrs")
46
47 #---    MATRIX MED SANDSYNLIGHEDER PÅ IKKE-INVERTEREDE VÆRDIER    ---#
48 print (rbind(c(pf(sum((1-dea_crs$eff)^2)/sum((1-dea_crs$eff)^2),
49   dim(data_test_crs)[1]-1,dim(data_test_crs)[1]-1 ) ,
50   pf(sum((1-dea_crs$eff)^2)/sum((1-dea_irs$eff)^2),
51   dim(data_test_crs)[1]-1,dim(data_test_irs)[1]-1 ) ,
52   pf(sum((1-dea_crs$eff)^2)/sum((1-dea_vrs$eff)^2),
53   dim(data_test_crs)[1]-1,dim(data_test_vrs)[1]-1 )) ,
54 c(pf(sum((1-dea_irs$eff)^2)/sum((1-dea_crs$eff)^2),
55   dim(data_test_irs)[1]-1,dim(data_test_crs)[1]-1 ) ,
56   pf(sum((1-dea_irs$eff)^2)/sum((1-dea_irs$eff)^2),
57   dim(data_test_irs)[1]-1,dim(data_test_irs)[1]-1 ) ,
58   pf(sum((1-dea_irs$eff)^2)/sum((1-dea_vrs$eff)^2),
59   dim(data_test_irs)[1]-1,dim(data_test_vrs)[1]-1 )) ,
60 c(pf(sum((1-dea_vrs$eff)^2)/sum((1-dea_crs$eff)^2),
61   dim(data_test_vrs)[1]-1,dim(data_test_crs)[1]-1 ) ,
62   pf(sum((1-dea_vrs$eff)^2)/sum((1-dea_irs$eff)^2),
63   dim(data_test_vrs)[1]-1,dim(data_test_irs)[1]-1 ) ,
64   pf(sum((1-dea_vrs$eff)^2)/sum((1-dea_vrs$eff)^2),
65   dim(data_test_vrs)[1]-1,dim(data_test_vrs)[1]-1 ))) )
66
67 #-----    MATRIX MED SANDSYNLIGHEDER PÅ INVERTEREDE VÆRDIER    -----#
68 print (rbind(c(pf(sum((1-1/dea_crs$eff)*(1-1/dea_crs$eff))/sum((1-
69 1/dea_crs$eff)*(1-1/dea_crs$eff)), dim(data_test_crs)[1]-
70 1,dim(data_test_crs)[1]-1 ) ,
71   pf(sum((1-1/dea_crs$eff)*(1-1/dea_crs$eff))/sum((1-
72 1/dea_irs$eff)*(1-1/dea_irs$eff)), dim(data_test_crs)[1]-
73 1,dim(data_test_irs)[1]-1 ) ,
74   pf(sum((1-1/dea_crs$eff)*(1-1/dea_crs$eff))/sum((1-
75 1/dea_vrs$eff)*(1-1/dea_vrs$eff)), dim(data_test_crs)[1]-
76 1,dim(data_test_vrs)[1]-1 )) ,
77 c(pf(sum((1-1/dea_irs$eff)*(1-1/dea_irs$eff))/sum((1-1/dea_crs$eff)*(1-
78 1/dea_crs$eff)), dim(data_test_irs)[1]-1,dim(data_test_crs)[1]-1 ) ,
79   pf(sum((1-1/dea_irs$eff)*(1-1/dea_irs$eff))/sum((1-
80 1/dea_irs$eff)*(1-1/dea_irs$eff)), dim(data_test_irs)[1]-
81 1,dim(data_test_irs)[1]-1 ) ,
82   pf(sum((1-1/dea_irs$eff)*(1-1/dea_irs$eff))/sum((1-
83 1/dea_vrs$eff)*(1-1/dea_vrs$eff)), dim(data_test_irs)[1]-
84 1,dim(data_test_vrs)[1]-1 )) ,
85 c(pf(sum((1-1/dea_vrs$eff)*(1-1/dea_vrs$eff))/sum((1-1/dea_crs$eff)*(1-
86 1/dea_crs$eff)), dim(data_test_vrs)[1]-1,dim(data_test_crs)[1]-1 ) ,
87   pf(sum((1-1/dea_vrs$eff)*(1-1/dea_vrs$eff))/sum((1-
88 1/dea_irs$eff)*(1-1/dea_irs$eff)), dim(data_test_vrs)[1]-
89 1,dim(data_test_irs)[1]-1 ) ,
90   pf(sum((1-1/dea_vrs$eff)*(1-1/dea_vrs$eff))/sum((1-
91 1/dea_vrs$eff)*(1-1/dea_vrs$eff)), dim(data_test_vrs)[1]-
92 1,dim(data_test_vrs)[1]-1 ))) )

```

SFA-TEST.R

I benchmarkingmetoden bliver fjernvarmevirksomhederne testet for, om der er fjernvarmevirksomheder med en kraftig påvirkning af de estimerede parameter, og dermed på de estimerede resultater af en lineær regression af totalomkostningerne på modelydelserne. Denne kode indeholder denne test, som foretages ved at beregne Cooks distance.

```

1 #####-----#####
2 #####-----  FORSYNINGSTILSYNETS SCRIPT TIL BENCHMARKING  -----#####
3 #####-----#####
4
5
6 #-----          DATA FOR ALLE NETVIRKSOMHEDER INDLÆSES          -----#
7 data_alle_log <- data[ , names(data) %in% c( "Vaerk" , "CVR" ,
8   "Pnummer" , omkostninger , modelydelser ) ]
9
10 #-----          DATA TRANSFORMERES          -----#
11 sfa_input_alle_log <- log(as.matrix(data_alle_log[omkostninger]))
12 sfa_output_alle_log <- log(as.matrix(data_alle_log[modelydelser]))
13 abs(sfa_output_alle_log)
14 sfa_output_alle_log[is.infinite(sfa_output_alle_log)] <- 0
15
16 #-----          MATERIALET REDUCERES TIL RELEVANTE DATA          -----#
17 sfadata <- cbind( data_alle_log[ , names(data) %in% c( "Vaerk" , "CVR" ,
18   "Pnummer" ) ] , sfa_input_alle_log , sfa_output_alle_log )
19
20 #-----          INPUT OG OUTPUT DEFINERES          -----#
21 sfa_input_log <- as.matrix(sfadata[omkostninger])
22 sfa_output_log <- as.matrix(sfadata[modelydelser])
23 #abs(sfa_output_log)
24 #sfa_output_log[is.infinite(sfa_output_log)] <- 0
25
26 #-----          COOKS DISTANCE FOR AT IDENTIFICERE OUTLIERS          -----#
27 ols_log <- lm(sfa_input_alle_log[,omkostninger] ~ sfa_output_log)
28 cooks_log <- cooks.distance(ols_log) ;
29 cookplot_log <- cbind(sfadata["Vaerk"],cooks_log)
30 sfa_cutoff <- sfacutoff
31 cooks <- ifelse(cooks.distance(ols_log) > sfa_cutoff , 1 , 0)
32 sfadata <- cbind( data[ , names(data) %in% c( "Vaerk" , "CVR" ) ] ,
33   sfa_input_log , sfa_output_log, cooks)
34 sfa_outlier_log <- NULL ;
35   sfa_outlier_log <- sfadata[ (sfadata$cooks == 1) ,]
36
37 #---          TRANSFORMERET DATA FOR REPRÆSENTATIVE NETVIRKSOMHEDER          ---#
38 sfa_front_log <- sfadata[ ( sfadata$cooks == 0 ) , ]
39 sfa_front_input_log <- sfa_front_log[ , omkostninger ]
40 sfa_front_output_log <- sfa_front_log[ , modelydelser ]

```