

13

Stochastic Frontier Analysis for Healthcare, with Illustrations in R

APPENDIX 13.1 R CODE FOR SYNTHETIC DATA-SET GENERATION

Due to confidentiality, we cannot provide the Queensland hospital data set used in our illustrations. As an alternative, we provide a synthetic data set to the interested readers to illustrate the SFMs introduced in this chapter with R.

The R code used to generate the synthetic panel data set is as noted in Box 13.1, in which we assume a scenario of 100 decision-making units (e.g., firms, factories, banks, and hospitals) in a period of four years. The production process contains three inputs, one output, and three binary environmental factors.

When using a replicable data generation process (i.e., setting the seed of the random number generator as 99 in the following code), the readers can obtain the inefficiency estimation as reported in Table 13.4 and Figure 13.2 from R.

TABLE 13.4 *Statistics of the estimated level of inefficiency (%) by different models with the synthetic data set*

Model	Mean	Std. dev.	Min	Q1	Median	Q3	Max
ALS77	44.98%	6.83%	30.21%	39.96%	44.74%	49.49%	67.23%
SS84	58.44%	20.76%	0.00%	41.93%	59.78%	70.44%	100.00%
PL81	39.66%	10.87%	18.27%	30.54%	38.91%	45.16%	64.35%
CSS90	52.75%	26.72%	0.00%	36.63%	56.69%	73.52%	100.00%
BC92	56.86%	9.47%	30.21%	49.06%	57.69%	62.86%	74.73%
Go5	0.01%	0.00%	0.01%	0.01%	0.01%	0.01%	0.01%
KLH14T	0.50%	0.00%	0.49%	0.49%	0.50%	0.50%	0.50%
KLH14P	2.79%	0.05%	2.65%	2.75%	2.79%	2.82%	2.89%
BC95	21.24%	15.81%	0.00%	13.53%	23.47%	23.47%	52.05%
SVKZ	39.14%	25.60%	0.00%	0.00%	51.25%	59.26%	73.00%

2 *Stochastic Frontier Analysis for Healthcare, with Illustrations in R*

```

1 rm(list=ls())
2 graphics.off()
3 # Generate synthetic panel data set
  -----
4
5 set.seed(99) # Set fixed seed for synthetic data
6
7 # Unit ID and Year:
8 id = sort(rep(1:100, times = 4))
9 Year = rep(1:4, times = 100)
10 t = Year
11 t2 = t^2
12
13 # Output
14 Y = rnorm(400, mean=-2, sd=1.6)
15
16 # Inputs
17 X1 = rnorm(400, mean=3.3, sd=1.3)
18 X2 = rnorm(400, mean=-1.6, sd=1.5)
19 X3 = rnorm(400, mean=13.6, sd=2)
20
21 # Environmental
22 #Assume 20% of units are 1 in Z1
23 Z1 = rep(c(1, 0), times= c(20,80))[order(runif(100))]
24 Z1 = rep(Z1, each = 4) # Replicate for a panel data set
25 Z2 = rep(c(1, 0), times= c(30,70))[order(runif(100))]
26 Z2 = rep(Z2, each = 4)
27 Z3 = rep(c(1, 0), times= c(80,20))[order(runif(100))]
28 Z3 = rep(Z3, each = 4)
29
30 # Composed data set
31 data = as.data.frame(cbind(id, Year, Y, X1, X2, X3, Z1, Z2, Z3, t, t2))
32 # Convert to panel data
33 library(plm)
34 paneldata<- pdata.frame(data, c("id","Year"))
35
36 # Define formulas for different models
37 form = Y ~ X1 + X2 + X3 # For ALS77/SS84/PL81/BC92/KLH14
38 formz = Y ~ X1 + X2 + X3 | Z1 + Z2 + Z3 # For BC95
39 formt = Y~ X1 + X2 + X3 + factor(id) # For G05
40 formc = Y~ X1 + X2 + X3 + t + t2 # For CSS90
41
42 fZ1 = as.factor(Z1)
43 fZ2 = as.factor(Z2)
44 fZ3 = as.factor(Z3)
45
46 forms = Y ~ X1 + X2 + X3 + fZ1 + fZ2 + fZ3 # For SVKZ
47 forms3 = ehat3 ~ X1 + X2 + X3 + fZ1 + fZ2 + fZ3 #For SVKZ
48
49 # The synthetic data set can be then applied with the SFM code as
  illustrated in Appendix B.

```

BOX 13.10: *Synthetic data set generation process*

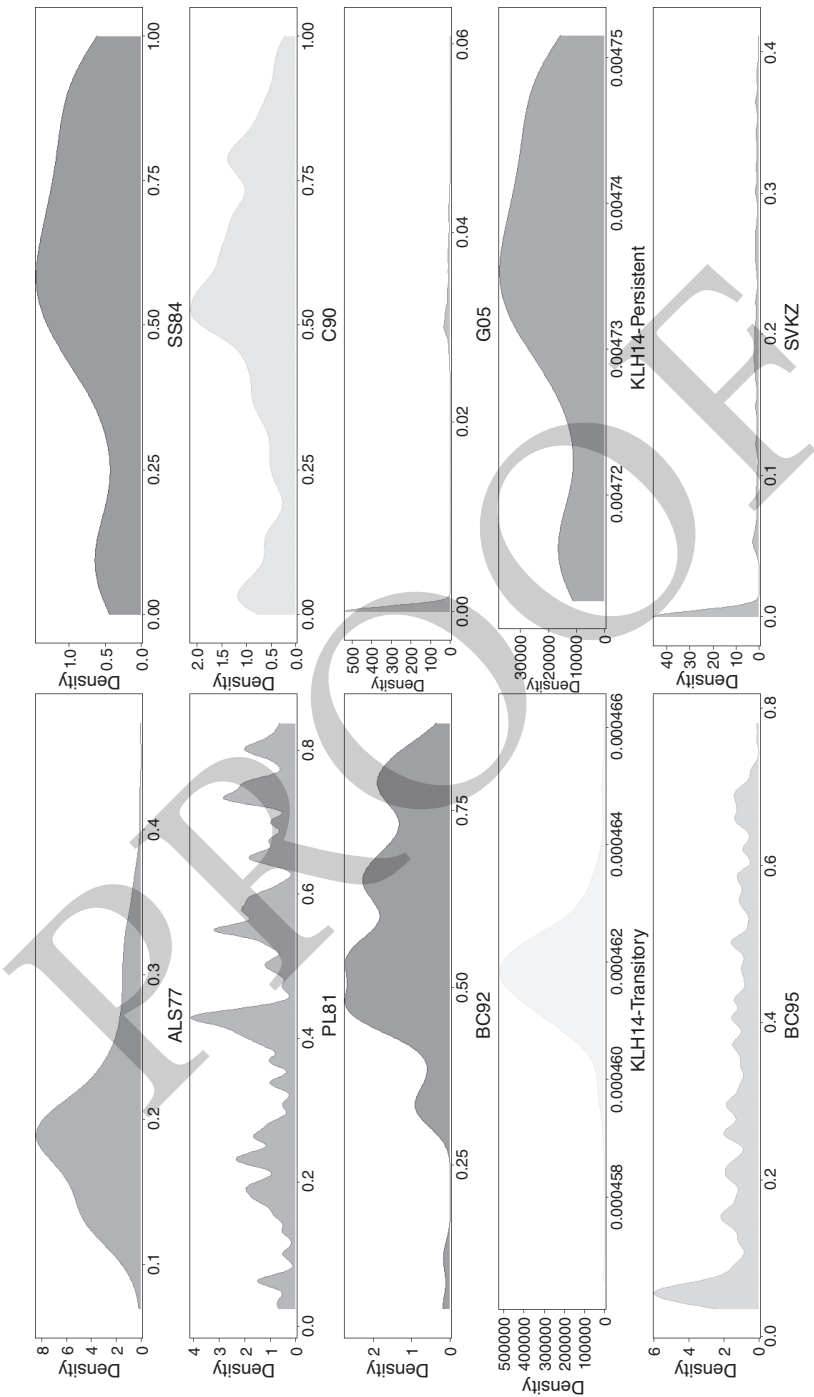


FIGURE 13.2 Estimated kernel densities of the inefficiency level of different SFMs with the synthetic data set

4 *Stochastic Frontier Analysis for Healthcare, with Illustrations in R*

```
> summary(als77)
Error Components Frontier (see Battese & Coelli 1992)
Inefficiency decreases the endogenous variable (as in a production function)
The dependent variable is logged
Iterative ML estimation terminated after 8 iterations:
log likelihood values and parameters of two successive iterations
are within the tolerance limit

final maximum likelihood estimates
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.751327    1.184760 -1.4782  0.13935
x1           0.032920    0.063208  0.5208  0.60249
x2           0.012927    0.053784  0.2403  0.81006
x3           0.010792    0.043463  0.2483  0.80391
sigmaSq      3.057531    1.393798  2.1937  0.02826 *
gamma        0.188451    0.623408  0.3023  0.76243
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
log likelihood value: -765.5158

cross-sectional data
total number of observations = 400

mean efficiency: 0.5973101
```

FIGURE 13.3 Model estimation of ALS77 with the synthetic data set

Moreover, taking the output of the ALS77 model as an example, the R output with our synthetic data set is as shown in Figure 13.3. The production function is estimated after eight iterations with a final log likelihood value of -765.5158 . The parameters estimated are as listed in the second column, which however, are all insignificant in this case. The estimated variances of the two error terms (i.e., σ_u^2 and σ_v^2) are reported afterward, where $\sigma^2 = \sigma_u^2 + \sigma_v^2$ and $\gamma = \sigma_u^2 / \sigma^2$. Moreover, the mean efficiency, which as noted in Section 13.2.1 as $E[\exp(-u)]$ is also reported at the end as 59.73%, whereas the individual inefficiency level can be estimated with the subsequent commands.

APPENDIX 13.2 FULL R CODE FOR THE EMPIRICAL ILLUSTRATIONS

```

1 rm(list=ls())
2 graphics.off()
3
4 # Data Process for the Queensland hospital data for illustrations
5 -----
6 # Read data
7 data <- read.csv("QLD.csv")
8 names(data)[names(data)=="HOSID"] <- "id"
9 names(data)[names(data)=="Yeardummy"] <- "Year"
10
11 # Convert to panel data
12 library(plm)
13 paneldata<- pdata.frame(data, c("id","Year"))
14 # Define formulas for different models
15 form = lAggout ~ lBEDS + lAgglabours + lSUPP
16 formz = lAggout ~ lBEDS + lAgglabours + lSUPP | TEACH + Remote + Small
17 formt = lAggout ~ lBEDS + lAgglabours + lSUPP + factor(id)
18 t = data$Year
19 t2 = t^2
20 formc = lAggout ~ lBEDS + lAgglabours + lSUPP + t + t2
21
22 # Define factor variables and formulas for SVKZ
23 attach(data)
24 fTEACH = as.factor(TEACH)
25 fRemote = as.factor(Remote)
26 fSmall = as.factor(Small)
27 forms = lAggout ~ lBEDS + lAgglabours + lSUPP + fTEACH + fRemote +
28         fSmall
29 forms3 = ehat3 ~ lBEDS + lAgglabours + lSUPP + fTEACH + fRemote +
30         fSmall
31
32 # Define output variable
33 Y = lAggout
34
35 #####
36 ### The following code can be applied with the synthetic data ###
37 ### or the Queensland hospital data processed above. ###
38 #####
39
40 # Code for SFMs -----
41 library(frontier)
42 ##### ALS77 #####
43 attach(data)
44 # Model estimation
45 als77 <- sfa(form, data = data, ineffDecrease = T, truncNorm = F,
46             timeEffect = F)
47 summary(als77)
48 # Individual inefficiency by JLMS
49 # Coefficients
50 als77coef <- coef(als77, which = "mle", extraPar = T)
51 # epsilon from fitted values
52 fals77 <- fitted(als77, asInData = T)
53 ei = Y - fals77

```

6 *Stochastic Frontier Analysis for Healthcare, with Illustrations in R*

```

51 # E(ui|ei)
52 us2 = (als77coef[["sigmaU"]])^2
53 vs2 = (als77coef[["sigmaV"]])^2
54 sigmastar = sqrt((vs2*us2)/(vs2+us2))
55 ustari = (-us2*ei)/(vs2+us2)
56 uals77 = ((sigmastar*dnorm(ustari/sigmastar))/(pnorm(ustari/sigmastar
    )))+ustari
57 inals77 = 1 - exp(-uals77) # Inefficiency estimation of individual
    unit
58 summary(inals77)
59
60 ##### SS84 #####
61 attach(paneldata)
62 # Fixed effects for ui
63 ss84 <- plm(form, data = paneldata,
64             model = "within", index = c("id","Year"), effect = "
                individual")
65 summary(ss84)
66 # ai and uihat
67 ai = as.numeric(unname(fixef(ss84)))
68 uss84 = max(ai) - ai
69 inss84 = (uss84 - min(uss84))/(max(uss84)-min(uss84)) # Relative
    inefficiency of individual unit
70 summary(inss84)
71
72 ##### PL81 #####
73 attach(paneldata)
74 # Fixed effects
75 pl81f <- plm(form, data = paneldata,
76             model = "random", index = c("id","Year"), effect = "
                individual")
77 summary(pl81f)
78 # Extract initial values
79 sigmau2 = as.numeric(ercomp(pl81f)[["sigma2"]][2])
80 sigmav2 = as.numeric(ercomp(pl81f)[["sigma2"]][1])
81 sigmasq = sigmau2+sigmav2
82 gamma = sigmau2/sigmasq
83
84 init_para = c(as.numeric(pl81f$coefficients), sigmasq, gamma)
85 # MLE with initial values
86 pl81 <- sfa(form, data = paneldata, ineffDecrease = T, truncNorm = F,
    timeEffect = F,
87             startVal = init_para)
88 summary(pl81)
89
90 # Estimate individual inefficiency
91 inpl81 <- 1-inefficiencies(pl81, asInData = T, logDepVar = T, minusU =
    T)
92 summary(inpl81)
93
94 ##### CSS90 #####
95 attach(paneldata)
96 # Fixed effects for ui
97 c90ori <- plm(form, data = paneldata,
98             model = "within", index = c("HOSID","Yeardummy"),
                effect = "twoways")
99 summary(c90ori)

```

```

I00 ait = lAggout - lBEDS*as.numeric(c90ori[["coefficients"]])[1] -
      lAgglabours*as.numeric(c90ori[["coefficients"]])[2] - lSUPP*as.
      numeric(c90ori[["coefficients"]])[3]
I01
I02 # Calculate ajt
I03 ti = c(1, 2, 3, 4)
I04 ti2 = ti^2
I05 aith = numeric()
I06
I07 for (i in 1:length(unique(id))) {
I08   aith = c(aith, as.numeric(unname(
I09     lm(ait[(4*(i-1)+1):(4*(i-1)+4)] ~ ti2 + ti)$fitted.values)))
I10 }
I11
I12 c90 = cbind(aith, t)
I13 # at that for each year
I14 ajtyear = aggregate(aith ~ t, data = c90, max)
I15 # Compute uithat
I16 c90 = as.data.frame(merge(c90, ajtyear, by = "t"))
I17 colnames(c90) <- c("t", "ait", "ajt")
I18 uc90 = c90$ajt - c90$ait
I19 summary(uc90)
I20 inc90 = (uc90 - min(uc90))/(max(uc90) - min(uc90)) # Relative
      inefficiency of individual unit
I21 summary(inc90)
I22
I23 ##### BC92 #####
I24 attach(paneldata)
I25 # Model estimation
I26 bc92 <- sfa(form, data = paneldata, ineffDecrease = T, truncNorm = T,
      timeEffect = T)
I27 summary(bc92, effic = F, logDepVar = T, effMinusU = T)
I28
I29 # Estimate individual inefficiency
I30 inbc92 <- 1-inefficiencies(bc92, asInData = T, logDepVar = T, minusU =
      T)
I31 summary(inbc92)
I32
I33 ##### G05 #####
I34 attach(paneldata)
I35 # Model estimation
I36 g05 <- sfa(formt, data = paneldata, ineffDecrease = T, truncNorm = F,
      timeEffect = T)
I37 summary(g05, effic = F, logDepVar = T, effMinusU = T)
I38
I39 # Estimate individual inefficiency
I40 ing05 <- 1-inefficiencies(g05, asInData = T, logDepVar = T, minusU = T
      )
I41 summary(ing05)
I42
I43 ##### KLH14 #####
I44 attach(paneldata)
I45 # Fixed effects for alhai and epsilonit
I46 klh14ori <- plm(form, data = paneldata,
I47   model = "within", index = c("id", "Year"), effect = "
      individual")
I48 summary(klh14ori)
I49 # alhai

```

8 *Stochastic Frontier Analysis for Healthcare, with Illustrations in R*

```

150 | alphas = as.numeric(unname(fixef(klh14ori)))
151 | # epsilon
152 | epsilon = as.numeric(klh14ori[["residuals"]])
153 | summary(alphas)
154 | summary(epsilon)
155 | # Constant for sfa estimation
156 | constant = 1
157 | # Persistent inefficiency
158 | klh14per = as.data.frame(cbind(alphas, constant))
159 | klh14p <- sfa(alphas~constant-1, data = klh14per, ineffDecrease = T,
160 |             truncNorm = F, timeEffect = F)
161 | summary(klh14p, effci = F, logDepVar = T, effMinusU = T)
162 | # Estimate individual persistent inefficiency
163 | inklh14p <- l-efficiencies (klh14p, asInData = T, logDepVar = T,
164 |                          minusU = T)
165 | summary(inklh14p)
166 | # Transitory inefficiency
167 | klh14tran = as.data.frame(cbind(epsilon, constant))
168 | klh14t <- sfa(epsilon~constant-1, data = klh14tran, ineffDecrease = T,
169 |             truncNorm = F, timeEffect = T)
170 | summary(klh14t, effci = F, logDepVar = T, effMinusU = T)
171 | # Estimate individual transitory inefficiency
172 | inklh14t <- l-efficiencies (klh14t, asInData = T, logDepVar = T,
173 |                          minusU = T)
174 | summary(inklh14t)
175 | ##### BC95 #####
176 | attach(paneldata)
177 | # Model estimation
178 | bc95 <- sfa(formz, data = paneldata, ineffDecrease = T, truncNorm = T,
179 |           timeEffect = F)
180 | summary(bc95, effci = F, logDepVar = T, effMinusU = T)
181 | # Estimate individual inefficiency
182 | inbc95 <- l-efficiencies (bc95, asInData = T, logDepVar = T, minusU =
183 |                          T)
184 | summary(inbc95)
185 | ##### SVKZ #####
186 | library(np)
187 | # Bandwidths selection
188 | bws.r1 <- npregbw(forms, regtype="ll", data=data, bwmethod = "cv.ls",
189 |                 ckertype = "epanechnikov")
189 | # Estimate conditional mean and extract fitted values and residuals
190 | r1.est <- npreg(bws=bws.r1, gradients=TRUE)
191 | r1hat <- fitted(r1.est)
192 | elhat <- residuals(r1.est)
193 | # Generate for r3
194 | ehat3 <- elhat^3
195 |
196 | # Bandwidths selection for r3 (skewness measures)
197 | bws.r3 <- npregbw(forms3, data=data, regtype="ll", bwmethod = "cv.ls",
198 |                 ckertype = "epanechnikov")
199 | # Estimate conditional mean and extract fitted values and residuals
200 | r3.est <- npreg(bws=bws.r3, gradients=TRUE)

```

```

200 r3hat <- fitted(r3.est)
201 e3hat <- residuals(r3.est)
202
203 # Estimate individual inefficiency
204 sigu3.hat <- sqrt(pi/2)*(pi/(pi-4))*r3hat
205 sigu.hat <- apply(cbind(0,sigu3.hat),1,FUN=max)^(1/3)
206 muhat <- sqrt(2/pi)*sigu.hat
207 summary(muhat)
208 insvkz = 1 - exp(-muhat) # Inefficiency estimation of individual unit
209 summary(insvkz)
210
211 # Statistics of results -----
212 library(qpcR)
213 # Combine inefficiency vectors of different length
214 ineffs <- qpcR::cbind.na(inals77, inss84, inpl81, inc90, inbc92,
215   ing05, inklh14t, inklh14p, inbc95, insvkz)
215 # Apply statistical analysis to each model
216 stat = list("mean" = apply(ineffs, 2, mean, na.rm = T),
217   "sd" = apply(ineffs, 2, sd, na.rm = T),
218   "min" = apply(ineffs, 2, min, na.rm = T),
219   "Q1" = apply(ineffs, 2, quantile, probs=0.25, na.rm = T),
220   "Median" = apply(ineffs, 2, quantile, probs=0.5, na.rm =
221     T),
222   "Q3" = apply(ineffs, 2, quantile, probs=0.75, na.rm = T),
223   "max" = apply(ineffs, 2, max, na.rm = T))
224
225 write.csv(stat, file = 'Stats of inefficiency.csv')
226
227 # Correlation between estimates and confidence interval of mean
228   inefficiency
229 cor(ineffs, method = c("spearman"))
230 library(tidyverse)
231 ci <- function(x) {
232   t.test(x, conf.level = 0.95)$conf.int
233 }
234 apply(ineffs, 2, ci)
235
236 # Rank of individual unit -----
237 # Ranks of panel data setting
238 Rankp = cbind(data$id, data$Year,
239   rank(inals77),
240   as.integer(as.factor(rank(inpl81, ties.method = "
241     average"))), #Tune for duplicate values
242   as.integer(as.factor(rank(inc90, ties.method = "average
243     "))),
244   rank(inbc92),
245   rank(inbc95),
246   rank(ing05),
247   rank(inklh14t),
248   as.integer(as.factor(rank(insvkz, ties.method = "
249     average"))))
250
251 # Rename for Rankp
252 Name <- c("id", "Year", "ALS77", "PL81", "CSS90", "BC92", "BC95", "G05", "
253   KLH14-transitory", "SVKZ")
254 colnames(Rankp) <- Name
255
256 # Ranks of cross-sectional data setting

```

```

251 Rankc = cbind(unique(data$id), #Extract id for individual unit
252               rank(inss84),
253               rank(inklhl14p))
254
255 # Rename for Rankc
256 Name <- c("id", "SS84", "KLH14-persistent")
257 colnames(Rankc) <- Name
258
259 cor(Rankp, method = c("spearman"))
260 cor(Rankc, method = c("spearman"))
261
262 # Kernel densities of estimations -----
263 require("ggplot2")
264 .df <- na.omit(data.frame(x = inals77))
265 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
266 .als<-ggplot(data = .df, aes(x = x, y = ..density..)) +
267   # Epanechnikov kernel and CV bandwidth
268   geom_density(
269     kernel = "gaussian",
270     bw = "ucv",
271     alpha = 0.5,
272     #Here for single group: color and fill without aes()
273     color = "brown1", fill = "brown1",
274     #shut the legend
275     show.legend = FALSE
276   ) +
277   scale_y_continuous(expand = c(0.01, 0)) +
278   xlab("ALS77") +
279   ylab("Density") +
280
281   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
      sans")
282 print(.als)
283 rm(.df, .nbins)
284
285 require("ggplot2")
286 .df <- na.omit(data.frame(x = inss84))
287 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
288 .ss84<-ggplot(data = .df, aes(x = x, y = ..density..)) +
289   # Epanechnikov kernel and CV bandwidth
290   geom_density(
291     kernel = "gaussian",
292     bw = "ucv",
293     alpha = 0.5,
294     #Here for single group: color and fill without aes()
295     color = "darkgreen", fill = "darkgreen",
296     #shut the legend
297     show.legend = FALSE
298   ) +
299   scale_y_continuous(expand = c(0.01, 0)) +
300   xlab("SS84") +
301   ylab("Density") +
302
303   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
      sans")
304 print(.ss84)
305 rm(.df, .nbins)
306

```

```

307 require("ggplot2")
308 .df <- na.omit(data.frame(x = inpl81))
309 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
310 .pl81<-ggplot(data = .df, aes(x = x, y = ..density..)) +
311   # Epanechnikov kernel and CV bandwidth
312   geom_density(
313     kernel = "gaussian",
314     bw = "ucv",
315     alpha = 0.5,
316     #Here for single group: color and fill without aes()
317     color = "dodgerblue", fill = "dodgerblue",
318     #shut the legend
319     show.legend = FALSE
320   ) +
321   scale_y_continuous(expand = c(0.01, 0)) +
322   xlab("PL81") +
323   ylab("Density") +
324
325   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
     sans")
326 print(.pl81)
327 rm(.df, .nbins)
328
329 require("ggplot2")
330 .df <- na.omit(data.frame(x = inc90))
331 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
332 .c90<-ggplot(data = .df, aes(x = x, y = ..density..)) +
333   # Epanechnikov kernel and CV bandwidth
334   geom_density(
335     kernel = "gaussian",
336     bw = "ucv",
337     alpha = 0.5,
338     #Here for single group: color and fill without aes()
339     color = "gold", fill = "gold",
340     #shut the legend
341     show.legend = FALSE
342   ) +
343   scale_y_continuous(expand = c(0.01, 0)) +
344   xlab("C90") +
345   ylab("Density") +
346
347   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
     sans")
348 print(.c90)
349 rm(.df, .nbins)
350
351 require("ggplot2")
352 .df <- na.omit(data.frame(x = inbc92))
353 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
354 .bc92<-ggplot(data = .df, aes(x = x, y = ..density..)) +
355   # Epanechnikov kernel and CV bandwidth
356   geom_density(
357     kernel = "gaussian",
358     bw = "ucv",
359     alpha = 0.5,
360     #Here for single group: color and fill without aes()
361     color = "blueviolet", fill = "blueviolet",
362     #shut the legend

```

```

363     show.legend = FALSE
364   ) +
365   scale_y_continuous(expand = c(0.01, 0)) +
366   xlab("BC92") +
367   ylab("Density") +
368
369   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
      sans")
370 print(.bc92)
371 rm(.df, .nbins)
372
373 require("ggplot2")
374 .df <- na.omit(data.frame(x = ing05))
375 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
376 .g05<-ggplot(data = .df, aes(x = x, y = ..density..)) +
377   # Epanechnikov kernel and CV bandwidth
378   geom_density(
379     kernel = "gaussian",
380     bw = "ucv",
381     alpha = 0.5,
382     #Here for single group: color and fill without aes()
383     color = "coral1", fill = "coral1",
384     #shut the legend
385     show.legend = FALSE
386   ) +
387   scale_y_continuous(expand = c(0.01, 0)) +
388   xlab("G05") +
389   ylab("Density") +
390
391   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
      sans")
392 print(.g05)
393 rm(.df, .nbins)
394
395 require("ggplot2")
396 # Disable scientific notation
397 options(scipen=1000)
398
399 .df <- na.omit(data.frame(x = inklh14t))
400 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
401 .klh14t<-ggplot(data = .df, aes(x = x, y = ..density..)) +
402   # Epanechnikov kernel and CV bandwidth
403   geom_density(
404     kernel = "gaussian",
405     bw = "ucv",
406     alpha = 0.5,
407     #Here for single group: color and fill without aes()
408     color = "darkolivegreen1", fill = "darkolivegreen1",
409     #shut the legend
410     show.legend = FALSE
411   ) +
412   scale_y_continuous(expand = c(0.01, 0)) +
413   xlab("KLH14-Transitory") +
414   ylab("Density") +
415
416   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
      sans")
417 print(.klh14t)

```

```

418 rm(.df, .nbins)
419
420 require("ggplot2")
421 .df <- na.omit(data.frame(x = inklh14p))
422 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
423 .klh14p<-ggplot(data = .df, aes(x = x, y = ..density..)) +
424   # Epanechnikov kernel and CV bandwidth
425   geom_density(
426     kernel = "gaussian",
427     bw = "ucv",
428     alpha = 0.5,
429     #Here for single group: color and fill without aes()
430     color = "royalblue", fill = "royalblue",
431     #shut the legend
432     show.legend = FALSE
433   ) +
434   scale_y_continuous(expand = c(0.01, 0)) +
435   xlab("KLH14-Persistent") +
436   ylab("Density") +
437
438   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
    sans")
439 print(.klh14p)
440 rm(.df, .nbins)
441
442 require("ggplot2")
443 .df <- na.omit(data.frame(x = inbc95))
444 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
445 .bc95<-ggplot(data = .df, aes(x = x, y = ..density..)) +
446   # Epanechnikov kernel and CV bandwidth
447   geom_density(
448     kernel = "gaussian",
449     bw = "ucv",
450     alpha = 0.5,
451     #Here for single group: color and fill without aes()
452     color = "khaki3", fill = "khaki3",
453     #shut the legend
454     show.legend = FALSE
455   ) +
456   scale_y_continuous(expand = c(0.01, 0)) +
457   xlab("BC95") +
458   ylab("Density") +
459
460   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
    sans")
461 print(.bc95)
462 rm(.df, .nbins)
463
464 require("ggplot2")
465 .df <- na.omit(data.frame(x = insvkz))
466 .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
467 .svkz<-ggplot(data = .df, aes(x = x, y = ..density..)) +
468   # Epanechnikov kernel and CV bandwidth
469   geom_density(
470     kernel = "gaussian",
471     bw = "ucv",
472     alpha = 0.5,
473     #Here for single group: color and fill without aes()

```

```
474     color = "mediumorchid1", fill = "mediumorchid1",
475     #shut the legend
476     show.legend = FALSE
477   ) +
478   scale_y_continuous(expand = c(0.01, 0)) +
479   xlab("SVKZ") +
480   ylab("Density") +
481
482   RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
    sans")
483 print(.svkz)
484 rm(.df, .nbins)
485
486 library("ggpubr")
487 Inefficiencies <- ggarrange(.als, .ss84, .pl81, .c90, .bc92, .g05, .
    klh14t, .klh14p, .bc95, .svkz,
488                               ncol = 2, nrow = 5)
489 Inefficiencies
490 #-----END-----
```