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%
G05	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%

```
rm(list=ls())
graphics.off()
# Generate synthetic panel data set ------------------------------------

set.seed(99)  # Set fixed seed for synthetic data

# Unit ID and Year:
id = sort(rep(1:100, times = 4))
Year = rep(1:4, times = 100)
t = Year
t2 = t^2

# Output
Y = rnorm(400, mean=-2, sd=1.6)

# Inputs
X1 = rnorm(400, mean=3.3, sd=1.3)
X2 = rnorm(400, mean=-1.6, sd=1.5)
X3 = rnorm(400, mean=13.6, sd=2)

# Environmental
#Assume 20% of units are 1 in Z1
Z1 = rep(c(1, 0), times= c(20,80))[order(runif(100))]
Z1 = rep(Z1, each = 4) # Replicate for a panel data set
Z2 = rep(c(1, 0), times= c(30,70))[order(runif(100))]
Z2 = rep(Z2, each = 4)
Z3 = rep(c(1, 0), times= c(80,20))[order(runif(100))]
Z3 = rep(Z3, each = 4)

# Composed data set
data = as.data.frame(cbind(id, Year, Y, X1, X2, X3, Z1, Z2, Z3, t, t2))
# Convert to panel data
library(plm)
paneldata<- pdata.frame(data, c("id","Year"))

# Define formulas for different models
form = Y ~ X1 + X2 + X3 # For ALS77/SS84/PL81/BC92/KLH14
formz = Y ~ X1 + X2 + X3 | Z1 + Z2 + Z3 # For BC95
formt = Y~ X1 + X2 + X3 + factor(id) # For G05
formc = Y~ X1 + X2 + X3 + t + t2 # For CSS90

fZ1 = as.factor(Z1)
fZ2 = as.factor(Z2)
fZ3 = as.factor(Z3)

forms = Y ~ X1 + X2 + X3 + fZ1 + fZ2 + fZ3 # For SVKZ
forms3 = ehat3 ~ X1 + X2 + X3 + fZ1 + fZ2 + fZ3 #For SVKZ

# 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*

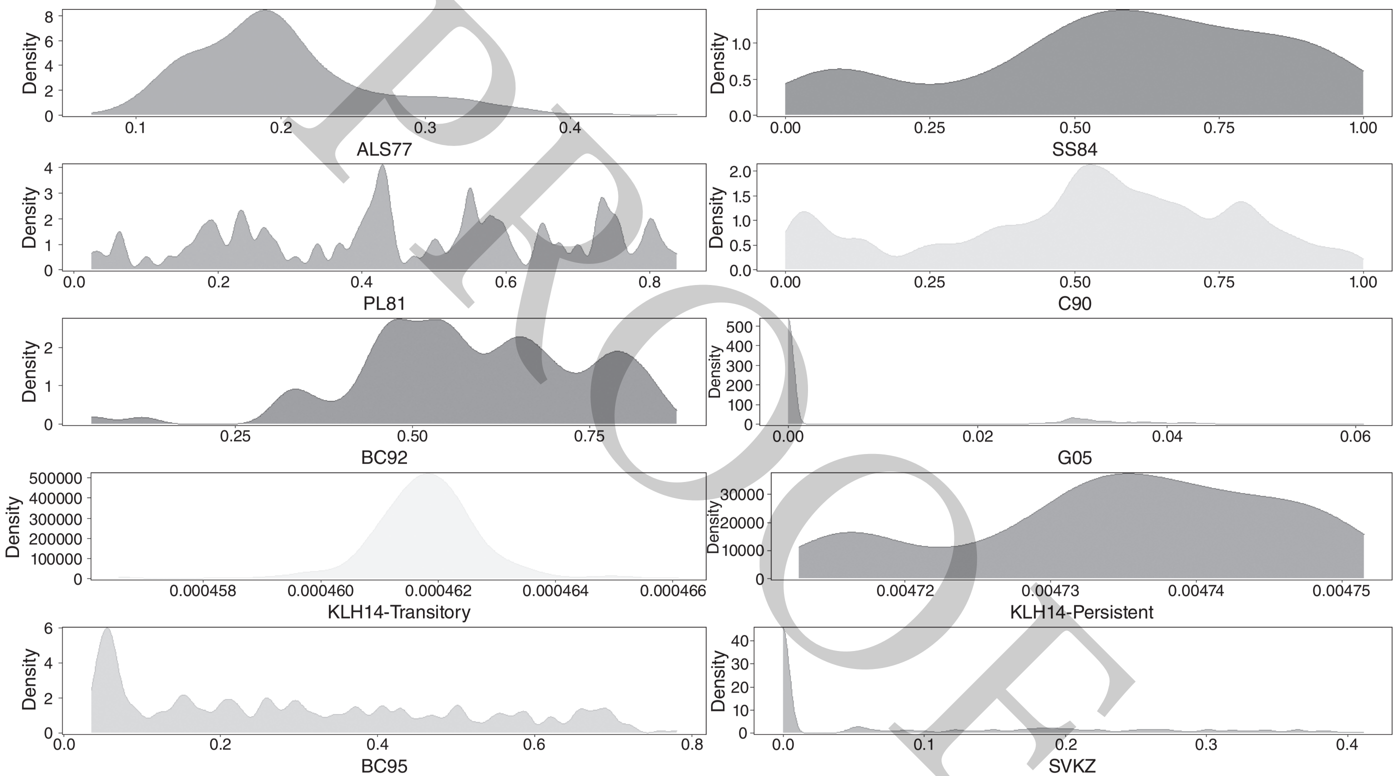

FIGURE 13.2 Estimated kernel densities of the inefficiency level of different SFMs with the synthetic data set

```
> 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

```
rm(list=ls())
graphics.off()

# Data Process for the Queensland hospital data for illustrations ------------

# Read data
data <- read.csv("QLD.csv")
names(data)[names(data)=="HOSID"] <- "id"
names(data)[names(data)=="Yeardummy"] <- "Year"

# Convert to panel data
library(plm)
paneldata<- pdata.frame(data, c("id","Year"))
# Define formulas for different models
form = lAggout ~ lBEDS + lAgglabours + lSUPP
formz = lAggout ~ lBEDS + lAgglabours + lSUPP | TEACH + Remote + Small
formt = lAggout ~ lBEDS + lAgglabours + lSUPP + factor(id)
t = data$Year
t2 = t^2
formc = lAggout ~ lBEDS + lAgglabours + lSUPP + t + t2

# Define factor variables and formulas for SVKZ
attach (data)
fTEACH = as.factor(TEACH)
fRemote = as.factor(Remote)
fSmall = as.factor(Small)
forms = lAggout ~ lBEDS + lAgglabours + lSUPP + fTEACH + fRemote + fSmall
forms3 = ehat3 ~ lBEDS + lAgglabours + lSUPP + fTEACH + fRemote + fSmall

# Define output variable
Y = lAggout

###############################################################
### The following code can be applied with the synthetic data ###
###     or the Queensland hospital data processed above.      ###
###############################################################

# Code for SFMs ------------
library(frontier)
######################## ALS77 ########################
attach(data)
# Model estimation
als77 <- sfa (form, data = data, ineffDecrease = T, truncNorm = F, timeEffect = F)
summary(als77)
# Individual inefficiency by JLMS
  # Coefficients
  als77coef <- coef(als77, which = "mle", extraPar = T)
  # epsilon from fitted values
  fals77 <- fitted(als77, asInData = T)
  ei = Y - fals77
```

```
  # E(ui|ei)
  us2 = (als77coef[["sigmaU"]])^2
  vs2 = (als77coef[["sigmaV"]])^2
  sigmastar = sqrt((vs2*us2)/(vs2+us2))
  ustari = (-us2*ei)/(vs2+us2)
  uals77 = ((sigmastar*dnorm(ustari/sigmastar))/(pnorm(ustari/sigmastar
      )))+ustari
  inals77 = 1 - exp(-uals77) # Inefficiency estimation of individual
      unit
  summary(inals77)

######################### SS84 #########################
attach(paneldata)
# Fixed effects for ui
ss84 <- plm(form, data = paneldata,
            model = "within", index = c("id","Year"), effect = "
                individual")
summary(ss84)
  # ai and uihat
  ai = as.numeric(unname(fixef(ss84)))
  uss84 = max(ai) - ai
  inss84 = (uss84 - min(uss84))/(max(uss84)-min(uss84)) # Relative
      inefficiency of individual unit
  summary(inss84)

######################### PL81 #########################
attach(paneldata)
# Fixed effects
pl81f <- plm(form, data = paneldata,
            model = "random", index = c("id","Year"), effect = "
                individual")
summary(pl81f)
  # Extract initial values
  sigmau2 = as.numeric(ercomp(pl81f)[["sigma2"]][2])
  sigmav2 = as.numeric(ercomp(pl81f)[["sigma2"]][1])
  sigmasq = sigmau2+sigmav2
  gamma = sigmau2/sigmasq

  init_para = c(as.numeric(pl81f$coefficients), sigmasq, gamma)
# MLE with initial values
pl81 <- sfa (form, data = paneldata, ineffDecrease = T, truncNorm = F,
    timeEffect =F,
             startVal = init_para)
summary(pl81)

  # Estimate individual inefficiency
  inpl81 <- 1-efficiencies (pl81, asInData = T, logDepVar = T, minusU =
       T)
  summary(inpl81)

######################### CSS90 #########################
attach(paneldata)
# Fixed effects for ui
c90ori <- plm(formc, data = paneldata,
                model = "within", index = c("HOSID","Yeardummy"),
                    effect = "twoways")
summary(c90ori)
```

```
ait = lAggout - lBEDS*as.numeric(c90ori[["coefficients"]])[1] - lAgglabours*as.numeric(c90ori[["coefficients"]])[2] - lSUPP*as.numeric(c90ori[["coefficients"]])[3]

  # Calculate ajt
  ti = c(1, 2, 3, 4)
  ti2 = ti^2
  aith = numeric()

  for (i in 1:length(unique(id))){
    aith = c(aith, as.numeric(unname(
      lm(ait[(4*(i-1)+1):(4*(i-1)+4)] ~ ti2 + ti)$fitted.values)))
  }

  c90 = cbind(aith, t)
  # athat for each year
  ajtyear = aggregate(aith ~ t, data = c90, max)
  # Compute uithat
  c90 = as.data.frame(merge(c90, ajtyear, by = "t"))
  colnames(c90) <- c("t","ait","ajt")
  uc90 = c90$ajt - c90$ait
  summary(uc90)
  inc90 = (uc90 - min(uc90))/(max(uc90)-min(uc90)) # Relative inefficiency of individual unit
  summary(inc90)

########################## BC92 ##########################
attach(paneldata)
# Model estimation
bc92 <- sfa (form, data = paneldata, ineffDecrease = T, truncNorm = T, timeEffect = T)
summary(bc92, effic = F, logDepVar = T, effMinusU = T)

  # Estimate individual inefficiency
  inbc92 <- 1-efficiencies (bc92, asInData = T, logDepVar = T, minusU = T)
  summary(inbc92)

########################## G05 ##########################
attach(paneldata)
# Model estimation
g05 <- sfa (formt, data = paneldata, ineffDecrease = T, truncNorm = F, timeEffect = T)
summary(g05, effic = F, logDepVar = T, effMinusU = T)

  # Estimate individual inefficiency
  ing05 <- 1-efficiencies (g05, asInData = T, logDepVar = T, minusU = T)
  summary(ing05)

########################## KLH14 ##########################
attach(paneldata)
# Fixed effects for alphai and epsilonit
klh14ori <- plm(form, data = paneldata,
              model = "within", index = c("id","Year"), effect = "individual")
summary(klh14ori)
# alphai
```

```
alphai = as.numeric(unname(fixef(klh14ori)))
# epsilonit
epsiloni = as.numeric(klh14ori[["residuals"]])
summary(alphai)
summary(epsiloni)
# Constant for sfa estimation
constant = 1
# Persistent inefficiency
klh14per = as.data.frame(cbind(alphai,constant))
klh14p <- sfa(alphai~constant-1, data = klh14per, ineffDecrease = T,
     truncNorm = F, timeEffect = F)
summary(klh14p, effic = F, logDepVar = T, effMinusU = T)

  # Estimate individual persistent inefficiency
  inklh14p <- 1-efficiencies (klh14p, asInData = T, logDepVar = T,
       minusU = T)
  summary(inklh14p)

# Transitory inefficiency
klh14tran = as.data.frame(cbind(epsiloni,constant))
klh14t <- sfa(epsiloni~constant-1, data = klh14tran, ineffDecrease = T,
      truncNorm = F, timeEffect = T)
summary(klh14t, effic = F, logDepVar = T, effMinusU = T)

  # Estimate individual transitory inefficiency
  inklh14t <- 1-efficiencies (klh14t, asInData = T, logDepVar = T,
       minusU = T)
  summary(inklh14t)

########################## BC95 ##########################
attach(paneldata)
# Model estimation
bc95 <- sfa(formz, data = paneldata, ineffDecrease = T, truncNorm = T,
     timeEffect = F)
summary(bc95, effic = F, logDepVar = T, effMinusU = T)

  # Estimate individual inefficiency
  inbc95 <- 1-efficiencies (bc95, asInData = T, logDepVar = T, minusU =
       T)
  summary(inbc95)
########################## SVKZ ##########################
library(np)

# Bandwidths selection
bws.r1 <- npregbw(forms, regtype="ll", data=data, bwmethod = "cv.ls",
     ckertype = "epanechnikov")
# Estimate conditional mean and extract fitted values and residuals
r1.est <- npreg(bws=bws.r1, gradients=TRUE)
r1hat   <- fitted(r1.est)
e1hat  <- residuals(r1.est)
# Generate for r3
ehat3 <- e1hat^3

# Bandwidths selection for r3 (skewness measures)
bws.r3 <- npregbw(forms3, data=data, regtype="ll", bwmethod = "cv.ls",
     ckertype = "epanechnikov")
# Estimate conditional mean and extract fitted values and residuals
r3.est  <- npreg(bws=bws.r3, gradients=TRUE)
```

```
r3hat  <- fitted(r3.est)
e3hat  <- residuals(r3.est)

  # Estimate individual inefficiency
  sigu3.hat <- sqrt(pi/2)*(pi/(pi-4))*r3hat
  sigu.hat <- apply(cbind(0,sigu3.hat),1,FUN=max)^(1/3)
  muhat <- sqrt(2/pi)*sigu.hat
  summary(muhat)
  insvkz = 1 - exp(-muhat) # Inefficiency estimation of individual unit
  summary(insvkz)

# Statistics of results ------------
  library(qpcR)
  # Combine inefficiency vectors of different length
  ineffs <- qpcR:::cbind.na(inals77, inss84, inpl81, inc90, inbc92,
      ing05, inklh14t, inklh14p,inbc95, insvkz)
  # Apply statistical analysis to each model
  stat = list("mean" = apply(ineffs, 2, mean, na.rm = T),
              "sd" = apply(ineffs, 2, sd, na.rm = T),
              "min" = apply(ineffs, 2, min, na.rm = T),
              "Q1" = apply(ineffs, 2, quantile, probs=0.25, na.rm = T),
              "Median" = apply(ineffs, 2, quantile, probs=0.5, na.rm =
                  T),
              "Q3" = apply(ineffs, 2, quantile, probs=0.75, na.rm = T),
              "max" = apply(ineffs, 2, max, na.rm = T))

  write.csv(stat, file = 'Stats of inefficiency.csv')

  # Correlation between estimates and confidence interval of mean
      inefficiency
  cor(ineffs, method = c("spearman"))
  library(tidyverse)
  ci <- function(x) {
    t.test(x, conf.level = 0.95)$conf.int
  }
  apply(ineffs, 2, ci)

# Rank of individual unit ------------
  # Ranks of panel data setting
  Rankp = cbind(data$id, data$Year,
                rank(inals77),
                as.integer(as.factor(rank(inpl81, ties.method = "
                    average"))), #Tune for duplicate values
                as.integer(as.factor(rank(inc90, ties.method = "average
                    "))),
                rank(inbc92),
                rank(inbc95),
                rank(ing05),
                rank(inklh14t),
                as.integer(as.factor(rank(insvkz, ties.method = "
                    average"))))

  # Rename for Rankp
  Name <- c("id","Year","ALS77","PL81","CSS90","BC92","BC95","G05","
    KLH14-transitory","SVKZ")
  colnames(Rankp) <- Name

  # Ranks of cross-sectional data setting
```

```
  Rankc = cbind(unique(data$id), #Extract id for individual unit
                rank(inss84),
                rank(inklh14p))

  # Rename for Rankc
  Name <- c("id","SS84", "KLH14-persistent")
  colnames(Rankc) <- Name

  cor(Rankp, method = c("spearman"))
  cor(Rankc, method = c("spearman"))

# Kernel densities of estimations ------------
  require("ggplot2")
  .df <- na.omit(data.frame(x = inals77))
  .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
  .als<-ggplot(data = .df, aes(x = x, y = ..density..)) +
    # Epanechnikov kernel and CV bandwidth
    geom_density(
      kernel = "gaussian",
      bw = "ucv",
      alpha = 0.5,
      #Here for single group: color and fill without aes()
      color = "brown1", fill = "brown1",
      #shut the legend
      show.legend = FALSE
    ) +
    scale_y_continuous(expand = c(0.01, 0)) +
    xlab("ALS77") +
    ylab("Density") +

    RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
  print(.als)
  rm(.df, .nbins)

  require("ggplot2")
  .df <- na.omit(data.frame(x = inss84))
  .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
  .ss84<-ggplot(data = .df, aes(x = x, y = ..density..)) +
    # Epanechnikov kernel and CV bandwidth
    geom_density(
      kernel = "gaussian",
      bw = "ucv",
      alpha = 0.5,
      #Here for single group: color and fill without aes()
      color = "darkgreen", fill = "darkgreen",
      #shut the legend
      show.legend = FALSE
    ) +
    scale_y_continuous(expand = c(0.01, 0)) +
    xlab("SS84") +
    ylab("Density") +

    RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
  print(.ss84)
  rm(.df, .nbins)
```

```
require("ggplot2")
.df <- na.omit(data.frame(x = inpl81))
.nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
.pl81<-ggplot(data = .df, aes(x = x, y = ..density..)) +
  # Epanechnikov kernel and CV bandwidth
  geom_density(
    kernel = "gaussian",
    bw = "ucv",
    alpha = 0.5,
    #Here for single group: color and fill without aes()
    color = "dodgerblue", fill = "dodgerblue",
    #shut the legend
    show.legend = FALSE
  ) +
  scale_y_continuous(expand = c(0.01, 0)) +
  xlab("PL81") +
  ylab("Density") +

  RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
      sans")
print(.pl81)
rm(.df, .nbins)

require("ggplot2")
.df <- na.omit(data.frame(x = inc90))
.nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
.c90<-ggplot(data = .df, aes(x = x, y = ..density..)) +
  # Epanechnikov kernel and CV bandwidth
  geom_density(
    kernel = "gaussian",
    bw = "ucv",
    alpha = 0.5,
    #Here for single group: color and fill without aes()
    color = "gold", fill = "gold",
    #shut the legend
    show.legend = FALSE
  ) +
  scale_y_continuous(expand = c(0.01, 0)) +
  xlab("C90") +
  ylab("Density") +

  RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "
      sans")
print(.c90)
rm(.df, .nbins)

require("ggplot2")
.df <- na.omit(data.frame(x = inbc92))
.nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
.bc92<-ggplot(data = .df, aes(x = x, y = ..density..)) +
  # Epanechnikov kernel and CV bandwidth
  geom_density(
    kernel = "gaussian",
    bw = "ucv",
    alpha = 0.5,
    #Here for single group: color and fill without aes()
    color = "blueviolet", fill = "blueviolet",
    #shut the legend
```

```
      show.legend = FALSE
    ) +
    scale_y_continuous(expand = c(0.01, 0)) +
    xlab("BC92") +
    ylab("Density") +

    RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
  print(.bc92)
  rm(.df, .nbins)

  require("ggplot2")
  .df <- na.omit(data.frame(x = ing05))
  .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
  .g05<-ggplot(data = .df, aes(x = x, y = ..density..)) +
    # Epanechnikov kernel and CV bandwidth
    geom_density(
      kernel = "gaussian",
      bw = "ucv",
      alpha = 0.5,
      #Here for single group: color and fill without aes()
      color = "coral1", fill = "coral1",
      #shut the legend
      show.legend = FALSE
    ) +
    scale_y_continuous(expand = c(0.01, 0)) +
    xlab("G05") +
    ylab("Density") +

    RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
  print(.g05)
  rm(.df, .nbins)

  require("ggplot2")
  # Disable scientific notation
  options(scipen=1000)

  .df <- na.omit(data.frame(x = inklh14t))
  .nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
  .klh14t<-ggplot(data = .df, aes(x = x, y = ..density..)) +
    # Epanechnikov kernel and CV bandwidth
    geom_density(
      kernel = "gaussian",
      bw = "ucv",
      alpha = 0.5,
      #Here for single group: color and fill without aes()
      color = "darkolivegreen1", fill = "darkolivegreen1",
      #shut the legend
      show.legend = FALSE
    ) +
    scale_y_continuous(expand = c(0.01, 0)) +
    xlab("KLH14-Transitory") +
    ylab("Density") +

    RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
  print(.klh14t)
```

```
rm(.df, .nbins)

require("ggplot2")
.df <- na.omit(data.frame(x = inklh14p))
.nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
.klh14p<-ggplot(data = .df, aes(x = x, y = ..density..)) +
  # Epanechnikov kernel and CV bandwidth
  geom_density(
    kernel = "gaussian",
    bw = "ucv",
    alpha = 0.5,
    #Here for single group: color and fill without aes()
    color = "royalblue", fill = "royalblue",
    #shut the legend
    show.legend = FALSE
  ) +
  scale_y_continuous(expand = c(0.01, 0)) +
  xlab("KLH14-Persistent") +
  ylab("Density") +

  RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
print(.klh14p)
rm(.df, .nbins)

require("ggplot2")
.df <- na.omit(data.frame(x = inbc95))
.nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
.bc95<-ggplot(data = .df, aes(x = x, y = ..density..)) +
  # Epanechnikov kernel and CV bandwidth
  geom_density(
    kernel = "gaussian",
    bw = "ucv",
    alpha = 0.5,
    #Here for single group: color and fill without aes()
    color = "khaki3", fill = "khaki3",
    #shut the legend
    show.legend = FALSE
  ) +
  scale_y_continuous(expand = c(0.01, 0)) +
  xlab("BC95") +
  ylab("Density") +

  RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
print(.bc95)
rm(.df, .nbins)

require("ggplot2")
.df <- na.omit(data.frame(x = insvkz))
.nbins <- pretty(range(.df$x), n = nclass.FD(.df$x), min.n = 1)
.svkz<-ggplot(data = .df, aes(x = x, y = ..density..)) +
  # Epanechnikov kernel and CV bandwidth
  geom_density(
    kernel = "gaussian",
    bw = "ucv",
    alpha = 0.5,
    #Here for single group: color and fill without aes()
```

```
      color = "mediumorchid1", fill = "mediumorchid1",
      #shut the legend
      show.legend = FALSE
    ) +
    scale_y_continuous(expand = c(0.01, 0)) +
    xlab("SVKZ") +
    ylab("Density") +

    RcmdrPlugin.KMggplot2::theme_simple(base_size = 14, base_family = "sans")
  print(.svkz)
  rm(.df, .nbins)

  library("ggpubr")
  Inefficiencies <- ggarrange(.als, .ss84, .pl81, .c90, .bc92, .g05, .klh14t, .klh14p, .bc95, .svkz,
                              ncol = 2, nrow = 5)
  Inefficiencies
#----------------------END----------------------------
```