Códigos utilizados no laboratório 7

Camille Menezes e Michel Miler

Maio de 2023

1 Atividade 1 - Diagnóstico do modelo

```
library(GGally)
library(tidyverse)
library(patchwork)
library(lmtest)
library(MASS)
library(datasets)
library(broom)
library(lindia)
library(dplyr)
library(summarytools)
library(gtsummary)
library(rigr)
library(olsrr)
library(gridExtra)
library(tikzDevice)
library(grDevices)

dados <- stackloss
names(dados)

#Análise descritiva
my_fn <- function(data, mapping, ...){
  p <- ggplot(data = data, mapping = mapping) +
    geom_point() +
    geom_smooth(method=lm,
                fill="blue", color="blue", ...)
  p
}

ggpairs(dados,
        upper = list(continuous = "cor"),
        lower = list(continuous = my_fn),
        axisLabels="none")

summary(dados)

boxp <- function(y,name){
  ggplot(dados) +
    aes(x=1,y=dados[,y]) +
    geom_boxplot() +
```

```
    #geom_jitter(color="black", size=0.4, alpha=0.9) +
    stat_summary(fun=mean, geom="point", shape=4, size=3) +
    scale_x_discrete("",breaks = NULL) +
    geom_text(aes(label=ifelse( dados[,y] > as.numeric(quantile(dados[,y])[4]
                                + 1.5*(quantile(dados[,y])[4] - quantile(dados[,y])[2])),
                                          as.character(rownames(dados)),'')),
              hjust=-0.7,vjust=0) +
    labs(y=name) +
    theme_minimal() +
    if (y == 4){
      annotate("text", x = Inf, y = Inf, hjust = 1, vjust = 1,
               colour = "black", label = paste("X = Média"))
    }
  else{
    theme_minimal()
  }
}

boxp(1, "Corrente de ar refrigerado") | boxp(2, "Temperatura de resfriamento") |
  boxp(3,"Concentração de ácido") | boxp(4, "Eficiência industrial")


# 1
model <- lm(stack.loss ~ Air.Flow + Water.Temp + Acid.Conc., data = dados)
summary(model)
anova(model)

MyAnova <- function(modelo, nd){
  m1 <- modelo
  np <- dim(anova(m1))[1]
  SQReg <- round(sum(anova(m1)$"Sum Sq"[1:(np-1)]), nd)
  glReg <- sum(anova(m1)$"Df"[1:(np-1)])
  SQRes <- round(anova(m1)$"Sum Sq"[np], nd)
  glRes <- anova(m1)$"Df"[np]
  SQTotal <- round(SQReg + SQRes, nd)
  glTotal <- glReg + glRes
  QMReg <- round(SQReg/glReg, nd)
  QMRes <- round(SQRes/glRes, nd)
  MyF <- round(QMReg/QMRes, nd)
  vpF <- ifelse(pf(MyF, glReg, glRes, lower.tail = F) < 0.0001, "<0.001",
                roud(pf(MyF, glReg, glRes, lower.tail = F), nd))
  ncolunas <- c("Fonte de Variação", "SQ", "gl", "F", "valor p")
  Tanova <- data.frame(FV = c("Regressão",
                              "Resíduos",
                              "Total"),
                       gl = c(glReg, glRes, glTotal),
                       SQ = c(SQReg, SQRes, SQTotal),
                       QM = c(QMReg, QMRes, " "),
                       Est.F = c(MyF, " ", " "),
                       valor.p = c(vpF, " ", " ")
  )
  Tanova
}
```

```r
MyAnova(model, 2)


# 2
shapiro.test(model$residuals)
gqtest(model)
dwtest(model)

a <- ggplot(model)+
  geom_point(aes(x=.fitted,y=studres(model)))+
  labs(x="Valores preditos \n A",y="Resíduos studentizados")+
  theme_bw()

b <- ggplot(model, aes(sample=studres(model)))+
  geom_qq()+
  geom_qq_line(color="red")+
  labs(x="Quantil teórico \n B", y="Resíduos studentizados")+
  theme_bw()

c <- ggplot(model)+
  geom_histogram(aes(x=.resid),binwidth=2.2,color="black", fill="white")+
  labs(y="Frequência",x="Resíduos \n C")+
  theme_bw()

a | b | c


# 3
m_infl <- influence.measures(model)
metricas <- as.data.frame(m_infl$infmat)
names(metricas)

#leverage
g <- ols_prep_rstudlev_data(model)
d <- g$levrstud
d$txt <- ifelse(d$color == "normal", NA, d$obs)
f <- d[d$color == "outlier", c("obs", "leverage", "rstudent")]
colnames(f) <- c("observation", "leverage", "stud_resid")
a <- ggplot(d, aes(leverage, rstudent, label = txt)) +
  geom_point(aes(colour = fct_color)) +
  scale_color_manual(labels = c("normal","ponto de alavanca","outlier",
  "outlier e ponto de alavanca"),
  values = c("black","blue", "red", "green")) +
  xlim(g$minx, g$maxx) +
  ylim(g$miny, g$maxy) +
  labs(colour = "Observação", x = "Pontos de alavanca \n B", y = "Resíduo studentizado") +
  geom_hline(yintercept = c(2,-2), colour = "black") +
  geom_vline(xintercept = g$lev_thrsh, colour = "black") +
  geom_text(vjust = -1, size = 3, colour = "black") +
  theme_bw()+
  annotate("text", x = Inf, y = Inf, hjust = 1.2, vjust = 1.5,
           colour = "darkred", label = paste("Limite:", round(g$lev_thrsh, 3)))
```

```r
#distância de Cook
k <- ols_prep_cdplot_data(model)
d <- ols_prep_outlier_obs(k)
f <- ols_prep_cdplot_outliers(k)
b <- ggplot(d, aes(x = obs, y = cd, label = txt)) +
  geom_bar(width = 0.3, stat = "identity",
           aes(fill = fct_color)) +
  scale_fill_manual(values = c("black", "red")) +
  labs(fill = "Observação") +
  ylim(0, k$maxx) +
  labs(x = "Observação \n A", y = "Distância de Cook") +
  geom_hline(yintercept = 0, colour = "gray") +
  geom_hline(yintercept = k$ts, colour = "red") +
  geom_text(hjust = -0.2, nudge_x = 0.05, size = 3, na.rm = TRUE) +
  theme_bw()+
  annotate("text", x = Inf, y = Inf, hjust = 1.2, vjust = 1.5,
           colour = "darkred", label = paste("Limite", round(k$ts, 3)))

b | a

#DFBETAs
ols_dfbetas <- function (model, print_plot = TRUE)
{  obs <- NULL
txt <- NULL
dfb <- dfbetas(model)
n <- nrow(dfb)
np <- ncol(dfb)
threshold <- 2/sqrt(n)
myplots <- list()
outliers <- list()
colnames(dfb) <- c("Intercepto",
                   "Corrente de ar refrigerado",
                   "Temperatura de resfriamento",
                   "Concentração de ácido")
for (i in seq_len(np)) {
  dbetas <- dfb[, i]
  df_data <- data.frame(obs = seq_len(n), dbetas = dbetas)
  d <- ols_prep_dfbeta_data(df_data, threshold)
  f <- ols_prep_dfbeta_outliers(d)
  p <- ggplot(d, aes(x = obs, y = dbetas,
                     label = txt, ymin = 0, ymax = dbetas)) +
    geom_linerange(colour = "black") +
    geom_hline(yintercept = c(0, threshold, -threshold),
               colour = "red") +
    labs(x="",y="") +
    ggtitle(paste(colnames(dfb)[i])) +
    theme_bw()+
    geom_text(hjust = -0.2, nudge_x = 0.15, size = 3,
              colour = "black", na.rm = TRUE)
  myplots[[i]] <- p
  outliers[[i]] <- f
}
if (print_plot) {
  marrangeGrob(myplots, nrow = 2, ncol = 2, top = quote(paste("")),
```

```
  left="DFBETAS",bottom="Observações")
}
}

ols_dfbetas(model)

#DFFITs
dbetas <- NULL
obs <- NULL
txt <- NULL
dffitsm <- unlist(dffits(model))
k <- length(coef(model))
n <- nrow(dados)
dffits_t <- sqrt(k/n) * 2
title <- names(model.frame(model))[1]
dfits_data <- data.frame(obs = seq_len(n), dbetas = dffitsm)
d <- ols_prep_dfbeta_data(dfits_data, dffits_t)
f <- ols_prep_dfbeta_outliers(d)
a <- ggplot(d, aes(x = obs, y = dbetas, label = txt, ymin = 0,
                   ymax = dffitsm)) +
  geom_linerange(colour = "black") +
  geom_hline(yintercept = c(0, dffits_t, -dffits_t), colour = "red") +
  labs(x = "Observação", y = "DFFITS") +
  theme_bw()+
  geom_text(hjust = -0.2, nudge_x = 0.15, size = 4, colour = "black",
            na.rm = TRUE) +
  annotate("text", x = Inf, y = Inf, hjust = 1.5,
           vjust = 1.5, colour = "darkred",
           label = paste("|Limite| =", round(dffits_t, 2)))

#COVRATIOs
cr <- data.frame(covratio(model))
cr['obs'] <- seq(1,nrow(dados))
limite <- (3*length(model$coefficients) ) / (nrow(dados)-length(model$coefficients))

b <- ggplot(data=cr,aes(x = obs, y = (1-covratio.model.), label = obs))+
  geom_line() +
  geom_hline(yintercept = c(limite, -1*limite),colour = "red")+
  geom_text(aes(label=ifelse(abs(1-covratio.model.)>limite,as.character(obs),'')),
  hjust=-0.2,vjust=0) +
  labs(x="Observação",y="COVRATIO")+
  theme_bw() +
  annotate("text", x = Inf, y = Inf, hjust = 1.5,
           vjust = 1.5, colour = "darkred", label = paste("|Limite| =", round(limite, 3)))

a | b


# 4
# gráfico da regressão parcial
fit1   <- lm(stack.loss ~ Water.Temp + Acid.Conc.,  data=dados)
fit1.1 <- lm(Air.Flow   ~ Water.Temp + Acid.Conc.,  data=dados)

fit2   <- lm(stack.loss ~ Air.Flow + Acid.Conc., data=dados)
```

```r
fit2.1 <- lm(Water.Temp ~ Air.Flow + Acid.Conc., data=dados)

fit3   <- lm(stack.loss ~ Air.Flow + Water.Temp, data=dados)
fit3.1 <- lm(Acid.Conc. ~ Air.Flow + Water.Temp, data=dados)

a <- ggplot()+
  aes(x = fit1.1$residuals, y = fit1$residuals) +
  geom_point()+
    geom_smooth(method = lm,se = FALSE) +
  labs(x = "res(X1~X2+X3) \n A",y="res(Y~X2+X3)")+
  theme_bw()

b <- ggplot() +
  aes(x = fit2.1$residuals, y = fit2$residuals) +
  geom_point() +
  geom_smooth(method = lm,se = FALSE) +
  labs(x = "res(X2~X1+X3) \n B",y="res(Y~X1+X3)") +
  theme_bw()

c <- ggplot() +
  aes(x = fit3.1$residuals, y = fit3$residuals) +
  geom_point() +
  geom_smooth(method = lm,se = FALSE) +
  labs(x = "res(X3~X1+X2) \n C",y="res(Y~X1+X2)") +
  theme_bw()

a | b | c

# Gráfico dos resíduos parciais
a <- ggplot()+
  aes(x = dados[,1], y = (model$residuals + model$coefficients[2]*dados[,1])) +
  geom_point()+
  geom_smooth(method = lm,se = FALSE) +
  labs(x = "X1 \n A",y=expression(paste("res(Y~X1+X2+X3) +", beta[1], "X1")))+
  theme_bw()

b <- ggplot() +
  aes(x = dados[,2], y = (model$residuals + model$coefficients[3]*dados[,2])) +
  geom_point() +
  geom_smooth(method = lm,se = FALSE) +
  labs(x = "X2 \n B",y=expression(paste("res(Y~X1+X2+X3) +", beta[2],"X2"))) +
  theme_bw()

c <- ggplot() +
  aes(x = dados[,3], y = (model$residuals + model$coefficients[4]*dados[,3])) +
  geom_point() +
  geom_smooth(method = lm,se = FALSE) +
  labs(x = "X3 \n C",y=expression(paste("res(Y~X1+X2+X3) +", beta[3],"X3"))) +
  theme_bw()

a | b | c
```

2 Atividade 2 - Colinearidade

```
evap <- read_delim("Lab07.txt", delim = "\t",
                   escape_double = FALSE, trim_ws = TRUE)
View(evap)
evap = data.frame(evap)
legenda=c('Intercepto','Temp. do ar diária máxima','Temp. do ar diária mínima',
          'Temp. média do ar','Temp. máxima diária do solo',
          'Temp. mínima diária do solo','Temp. média do solo',
          'Umidade relativa diária máxima','Umidade relativa diária mínima',
          'Umidade relativa média','Vento Total')

#columnLabels=c(legenda[c(5,6,7,2,3,4,8,9,10,11)],"Evaporação do solo")

ggpairs(evap[,4:14],upper = list(continuous = wrap("points", size = 1)),
        lower = list(continuous = wrap("blank")))

ggpairs(evap[,4:14],
        lower = list(continuous = "density", combo = "box_no_facet"),
        upper = list(continuous = wrap("points", size = 1), combo = "dot_no_facet"))

ggpairs(evap[,4:14],
        lower = list(continuous = wrap("points", size = 1)),
        upper = list(continuous = wrap("points", size = 1)))

boxp <- function(y,name){
  ggplot(evap) +
    aes(x=1,y=evap[,y]) +
    geom_boxplot() +
    #geom_jitter(color="black", size=0.4, alpha=0.9) +
    stat_summary(fun=mean, geom="point", shape=4, size=3) +
    scale_x_discrete("",breaks = NULL) +
    geom_text(aes(label=ifelse(evap[,y] > as.numeric(quantile(evap[,y])[4] +
    1.5*(quantile(evap[,y])[4] - quantile(evap[,y])[2])),
                               as.character(rownames(evap)),'')),
              hjust=-0.3,vjust=0) +
    geom_text(aes(label=ifelse(evap[,y] < as.numeric(quantile(evap[,y])[2] -
    1.5*(quantile(evap[,y])[4] - quantile(evap[,y])[2])),
                              as.character(rownames(evap)),'')),
              hjust=-0.3,vjust=0) +
    labs(y=name) +
    theme_minimal() +
    if (y == 4){
      annotate("text", x = Inf, y = Inf, hjust = 1, vjust = 1,
               colour = "black", label = paste("X = Média"))
    }
  else{
    theme_minimal()
  }
}

boxp(4, legenda[5]) | boxp(5, legenda[6]) |
boxp(6,legenda[7]) | boxp(7, legenda[2])|boxp(8, legenda[3]) | boxp(9, legenda[4])
```

```
boxp(10,legenda[8]) | boxp(11, legenda[9])| boxp(12, legenda[10]) |
boxp(13,legenda[11]) | boxp(14, "Evaporação do solo")


#1
modelo <- lm(EVAP~MAXAT+MINAT+AVAT+MAXST+MINST+AVST+MAXH+MINH+AVH+WIND,data=evap)
summary(modelo)
MyAnova(modelo,3)


#2
tab <- anova(modelo)
R <-  sum(tab$`Sum Sq`[1:length(tab$`Sum Sq`)-1])/
(sum(tab$`Sum Sq`[1:length(tab$`Sum Sq`)-1])+tab$`Sum Sq`[length(tab$`Sum Sq`)])
print(R)


#3
cor(evap[,4:14])

shapiro.test(modelo$residuals)
gqtest(modelo)
dwtest(modelo)

a <- ggplot(modelo)+
  geom_point(aes(x=.fitted,y=studres(modelo)))+
  labs(x="Valores preditos \n A",y="Resíduos studentizados")+
  theme_minimal()

b <- ggplot(modelo, aes(sample=studres(modelo)))+
  geom_qq()+
  geom_qq_line(color="red")+
  labs(x="Quantil teórico \n B", y="Resíduos studentizados")+
  theme_minimal()

c <- ggplot(modelo)+
  geom_histogram(aes(x=.resid),binwidth=2.2,color="black", fill="white")+
  labs(y="Frequência",x="Resíduos \n C")+
  theme_minimal()

a | b | c

g <- ols_prep_rstudlev_data(modelo)
d <- g$levrstud
d$txt <- ifelse(d$color == "normal", NA, d$obs)
f <- d[d$color == "outlier", c("obs", "leverage", "rstudent")]
colnames(f) <- c("observation", "leverage", "stud_resid")
a <- ggplot(d, aes(leverage, rstudent, label = txt)) +
  geom_point(aes(colour = fct_color)) +
  scale_color_manual(labels = c("normal","ponto de alavanca",
  "outlier","outlier e ponto de alavanca"),values = c("black","blue", "red", "green")) +
  xlim(g$minx, g$maxx) +
  ylim(g$miny, g$maxy) +
  labs(colour = "Observação", x = "Pontos de alavanca", y = "Resíduo studentizado") +
```

```
  geom_hline(yintercept = c(2,-2), colour = "black") +
  geom_vline(xintercept = g$lev_thrsh, colour = "black") +
  geom_text(vjust = -1, size = 3, colour = "black") +
  theme_bw()+
  annotate("text", x = Inf, y = Inf, hjust = 1.2, vjust = 1.5,
           colour = "darkred", label = paste("Limite:", round(g$lev_thrsh, 3)))

#distância de Cook
k <- ols_prep_cdplot_data(modelo)
d <- ols_prep_outlier_obs(k)
f <- ols_prep_cdplot_outliers(k)
b <- ggplot(d, aes(x = obs, y = cd, label = txt)) +
  geom_bar(width = 0.3, stat = "identity",
           aes(fill = fct_color)) +
  scale_fill_manual(values = c("black", "red")) +
  labs(fill = "Observação") +
  ylim(0, k$maxx) +
  labs(x = "Observação", y = "Distância de Cook") +
  geom_hline(yintercept = 0, colour = "gray") +
  geom_hline(yintercept = k$ts, colour = "red") +
  geom_text(hjust = -0.2, nudge_x = 0.05, size = 3, na.rm = TRUE) +
  theme_bw()+
  annotate("text", x = Inf, y = Inf, hjust = 1.2, vjust = 1.5,
           colour = "darkred", label = paste("Limite", round(k$ts, 3)))

b | a

#DFBETAs
ols_dfbetas <- function (modelo,l=1,u=FALSE,coln=NULL, print_plot = TRUE){
  obs <- NULL
  txt <- NULL
  dfb <- dfbetas(modelo)
  if (u==FALSE){
    u = ncol(dfb)
  }
  dfb<-dfb[,l:u]
  n<- nrow(dfb)
  if (is.null(n)){
    n<-length(dfb)
  }
  np <- ncol(dfb)
  threshold <- 2/sqrt(n)
  myplots <- list()
  outliers <- list()
  colnames(dfb)<-coln
  for (i in seq_len(np)) {
    dbetas <- dfb[, i]
    df_data <- data.frame(obs = seq_len(n), dbetas = dbetas)
    d <- ols_prep_dfbeta_data(df_data, threshold)
    f <- ols_prep_dfbeta_outliers(d)
    p <- ggplot(d, aes(x = obs, y = dbetas,
                       label = txt, ymin = 0, ymax = dbetas)) +
      geom_linerange(colour = "black") +
      geom_hline(yintercept = c(0, threshold, -threshold),
```

```
                colour = "red") +
      labs(x="",y="") +
      ggtitle(paste(colnames(dfb)[i])) +
      theme_bw()+
      geom_text(hjust = -0.2, nudge_x = 0.15, size = 3,
                colour = "black", na.rm = TRUE)
    myplots[[i]] <- p
    outliers[[i]] <- f
  }
  if (print_plot) {
    marrangeGrob(myplots, nrow = 2, ncol =2 , top = quote(paste("")),
    left="DFBETAS",bottom="Observações")
  }
}



ols_dfbetas(modelo,1,4,legenda[1:4])
ols_dfbetas(modelo,5,7,legenda[5:7])
ols_dfbetas(modelo,8,11,legenda[8:11])

#DFFITs
dbetas <- NULL
obs <- NULL
txt <- NULL
dffitsm <- unlist(dffits(modelo))
k <- length(coef(modelo))
n <- nrow(evap)
dffits_t <- sqrt(k/n) * 2
title <- names(model.frame(modelo))[1]
dfits_data <- data.frame(obs = seq_len(n), dbetas = dffitsm)
d <- ols_prep_dfbeta_data(dfits_data, dffits_t)
f <- ols_prep_dfbeta_outliers(d)
a <- ggplot(d, aes(x = obs, y = dbetas, label = txt, ymin = 0,
                   ymax = dffitsm)) + geom_linerange(colour = "black") +
  geom_hline(yintercept = c(0, dffits_t, -dffits_t), colour = "red") +
  labs(x = "Observação", y = "DFFITS") +
  theme_bw()+
  geom_text(hjust = -0.2, nudge_x = 0.15, size = 4, colour = "black",
            na.rm = TRUE) +
  annotate("text", x = Inf, y = Inf, hjust = 1.5,
           vjust = 1.5, colour = "darkred",
           label = paste("|Limite| =", round(dffits_t, 2)))

#COVRATIOs
cr = data.frame(covratio(modelo))
cr['obs'] = seq(1,nrow(evap))
limite = (3*length(modelo$coefficients) ) / (nrow(evap)-length(modelo$coefficients))

b <- ggplot(data=cr,aes(x = obs,y = (1-covratio.modelo.), label = obs))+
  geom_line() +
  geom_hline(yintercept = c(limite, -1*limite),colour = "red")+
  geom_text(aes(label=ifelse(abs(1-covratio.modelo.)>limite,
  as.character(obs),''')),hjust=-0.2,vjust=0) +
```

```
  labs(x="Observação",y="COVRATIO")+
  theme_bw() +
  annotate("text", x = Inf, y = Inf, hjust = 1.5,
           vjust = 1.5, colour = "darkred", label = paste("|Limite| =", round(limite, 3)))


a | b


#5
ml_maxat = lm(MAXAT~MINAT+AVAT+MAXST+MINST+AVST+MAXH+MINH+AVH+WIND,data=evap)
ml_minat = lm(MINAT~AVAT+MAXST+MINST+AVST+MAXH+MINH+AVH+WIND+MAXAT,data=evap)
ml_avat = lm(AVAT~MAXST+MINST+AVST+MAXH+MINH+AVH+WIND+MAXAT+MINAT,data=evap)
ml_maxst = lm(MAXST~MINST+AVST+MAXH+MINH+AVH+WIND+MAXAT+MINAT+AVAT,data=evap)
ml_minst = lm(MINST~AVST+MAXH+MINH+AVH+WIND+MAXAT+MINAT+AVAT+MAXST,data=evap)
ml_avst = lm(AVST~MAXH+MINH+AVH+WIND+MAXAT+MINAT+AVAT+MAXST+MINST,data=evap)
ml_maxh = lm(MAXH~MINH+AVH+WIND+MAXAT+MINAT+AVAT+MAXST+MINST+AVST,data=evap)
ml_minh = lm(MINH~AVH+WIND+MAXAT+MINAT+AVAT+MAXST+MINST+AVST+MAXH,data=evap)
ml_avh = lm(AVH~WIND+MAXAT+MINAT+AVAT+MAXST+MINST+AVST+MAXH+MINH,data=evap)
ml_wind = lm(WIND~MAXAT+MINAT+AVAT+MAXST+MINST+AVST+MAXH+MINH+AVH,data=evap)

mls = list(ml_maxat,ml_minat,ml_avat,ml_maxst,ml_minst,ml_avst,
        ml_maxh,ml_minh,ml_avh,ml_wind)
Rj2 = c()
i=1
for (ml in mls){
  tab<-anova(ml)
  Rj2[i] = sum(tab$`Sum Sq`[-length(tab$`Sum Sq`)])/
  (sum(tab$`Sum Sq`[-length(tab$`Sum Sq`)-1])+tab$`Sum Sq`[length(tab$`Sum Sq`)])
  print(Rj2[i])
  i=i+1
}

VIF = 1/(1-Rj2)
VIF

tabela <- matrix(c(Rj2,VIF),nrow=10,ncol=2)
tabela <- data.frame(tabela)
colnames(tabela) <- c('Rj2','VIF')
rownames(tabela) <- legenda[2:11]
tabela

#6

evap2 <- evap[-39,-c(4,5,6)]
modelo2 <- lm(EVAP~MAXAT+MINAT+AVAT+MAXH+MINH+AVH+WIND,data=evap2)

summary(modelo2)
MyAnova(modelo2,3)

shapiro.test(modelo2$residuals)
gqtest(modelo2)
dwtest(modelo2)
```

```r
a <- ggplot(modelo2)+
  geom_point(aes(x=.fitted,y=studres(modelo2)))+
  labs(x="Valores preditos \n A",y="Resíduos studentizados")+
  theme_minimal()

b <- ggplot(modelo2, aes(sample=studres(modelo2)))+
  geom_qq()+
  geom_qq_line(color="red")+
  labs(x="Quantil teórico \n B", y="Resíduos studentizados")+
  theme_minimal()

c <- ggplot(modelo2)+
  geom_histogram(aes(x=.resid),binwidth=2.2,color="black", fill="white")+
  labs(y="Frequência",x="Resíduos \n C")+
  theme_minimal()

a | b | c

g <- ols_prep_rstudlev_data(modelo2)
d <- g$levrstud
d$txt <- ifelse(d$color == "normal", NA, d$obs)
f <- d[d$color == "outlier", c("obs", "leverage", "rstudent")]
colnames(f) <- c("observation", "leverage", "stud_resid")
a <- ggplot(d, aes(leverage, rstudent, label = txt)) +
  geom_point(aes(colour = fct_color)) +
  scale_color_manual(labels = c("normal","ponto de alavanca","outlier",
  "outlier e ponto de alavanca"),values = c("black","blue", "red", "green")) +
  xlim(g$minx, g$maxx) +
  ylim(g$miny, g$maxy) +
  labs(colour = "Observação", x = "Pontos de alavanca", y = "Resíduo studentizado") +
  geom_hline(yintercept = c(2,-2), colour = "black") +
  geom_vline(xintercept = g$lev_thrsh, colour = "black") +
  geom_text(vjust = -1, size = 3, colour = "black") +
  theme_bw()+
  annotate("text", x = Inf, y = Inf, hjust = 1.2, vjust = 1.5,
           colour = "darkred", label = paste("Limite:", round(g$lev_thrsh, 3)))

#distância de Cook
k <- ols_prep_cdplot_data(modelo2)
d <- ols_prep_outlier_obs(k)
f <- ols_prep_cdplot_outliers(k)
b <- ggplot(d, aes(x = obs, y = cd, label = txt)) +
  geom_bar(width = 0.3, stat = "identity",
           aes(fill = fct_color)) +
  scale_fill_manual(values = c("black", "red")) +
  labs(fill = "Observação") +
  ylim(0, k$maxx) +
  labs(x = "Observação", y = "Distância de Cook") +
  geom_hline(yintercept = 0, colour = "gray") +
  geom_hline(yintercept = k$ts, colour = "red") +
  geom_text(hjust = -0.2, nudge_x = 0.05, size = 3, na.rm = TRUE) +
  theme_bw()+
  annotate("text", x = Inf, y = Inf, hjust = 1.2, vjust = 1.5,
           colour = "darkred", label = paste("Limite", round(k$ts, 3)))
```

```
b | a

#DFBETAs
ols_dfbetas <- function (modelo2,l=1,u=FALSE,coln=NULL, print_plot = TRUE){
  obs <- NULL
  txt <- NULL
  dfb <- dfbetas(modelo2)
  if (u==FALSE){
    u = ncol(dfb)
  }
  dfb<-dfb[,l:u]
  n<- nrow(dfb)
  if (is.null(n)){
    n<-length(dfb)
  }
  np <- ncol(dfb)
  threshold <- 2/sqrt(n)
  myplots <- list()
  outliers <- list()
  colnames(dfb)<-coln
  for (i in seq_len(np)) {
    dbetas <- dfb[, i]
    df_data <- data.frame(obs = seq_len(n), dbetas = dbetas)
    d <- ols_prep_dfbeta_data(df_data, threshold)
    f <- ols_prep_dfbeta_outliers(d)
    p <- ggplot(d, aes(x = obs, y = dbetas,
                       label = txt, ymin = 0, ymax = dbetas)) +
      geom_linerange(colour = "black") +
      geom_hline(yintercept = c(0, threshold, -threshold),
                 colour = "red") +
      labs(x="",y="") +
      ggtitle(paste(colnames(dfb)[i])) +
      theme_bw()+
      geom_text(hjust = -0.2, nudge_x = 0.15, size = 3,
                colour = "black", na.rm = TRUE)
    myplots[[i]] <- p
    outliers[[i]] <- f
  }
  if (print_plot) {
    marrangeGrob(myplots, nrow = 2, ncol =2 , top = quote(paste("")),
    left="DFBETAS",bottom="Observações")
  }
}

legenda=c('Intercepto','Temp. do ar diária máxima','Temp. do ar diária mínima',
          'Temp. média do ar','Umidade relativa diária máxima',
          'Umidade relativa diária mínima',
          'Umidade relativa diária média','Vento Total','Temp. mínima diária do solo')

ols_dfbetas(modelo2,1,4,legenda[1:4])
ols_dfbetas(modelo2,5,8,legenda[5:8])

#DFFITs
```

```r
dbetas <- NULL
obs <- NULL
txt <- NULL
dffitsm <- unlist(dffits(modelo2))
k <- length(coef(modelo2))
n <- nrow(evap2)
dffits_t <- sqrt(k/n) * 2
title <- names(model.frame(modelo2))[1]
dfits_data <- data.frame(obs = seq_len(n), dbetas = dffitsm)
d <- ols_prep_dfbeta_data(dfits_data, dffits_t)
f <- ols_prep_dfbeta_outliers(d)
a <- ggplot(d, aes(x = obs, y = dbetas, label = txt, ymin = 0,
                   ymax = dffitsm)) + geom_linerange(colour = "black") +
  geom_hline(yintercept = c(0, dffits_t, -dffits_t), colour = "red") +
  labs(x = "Observação", y = "DFFITS") +
  theme_bw()+
  geom_text(hjust = -0.2, nudge_x = 0.15, size = 4, colour = "black",
            na.rm = TRUE) +
  annotate("text", x = Inf, y = Inf, hjust = 1.5,
           vjust = 1.5, colour = "darkred",
           label = paste("|Limite| =", round(dffits_t, 2)))

#COVRATIOs
cr <- data.frame(covratio(modelo2))
cr['obs'] <-seq(1,nrow(evap2))
limite <- (3*length(modelo2$coefficients) ) / (nrow(evap2)-length(modelo2$coefficients))
b <- ggplot(data=cr,aes(x = obs,y = (1-covratio.modelo2.), label = obs))+
  geom_line() +
  geom_hline(yintercept = c(limite, -1*limite),colour = "red")+
  geom_text(aes(label=ifelse(abs(1-covratio.modelo2.)>limite,as.character(obs),'')),
  hjust=-0.2,vjust=0) +
  labs(x="Observação",y="COVRATIO")+
  theme_bw() +
  annotate("text", x = Inf, y = Inf, hjust = 1.5,
           vjust = 1.5, colour = "darkred", label = paste("|Limite| =", round(limite, 3)))

a | b



tab <- anova(modelo)
SqRegc <- sum(tab$`Sum Sq`[-length(tab$`Sum Sq`)])
SqResc <- tab$`Sum Sq`[length(tab$`Sum Sq`)]
tab2 <- anova(modelo2)
SqRegr <- sum(tab2$`Sum Sq`[-length(tab2$`Sum Sq`)])

Fr <- (SqRegc - SqRegr)*(nrow(evap)-length(tab$`Sum Sq`))/(SqResc*length(tab2$`Sum Sq`))

1-pf(Fr,4,nrow(evap)-8)
```