

```

#Questão 1 Prof. Marcos Toebe

# Carregar pacotes readxl
library(readxl)

# Link para o arquivo Excel no repositório
url_excel
<- "https://raw.githubusercontent.com/lpradebon/Quest-o1_Prof_Marcos_Toebe/main/Dados_quest_1.xlsx"

# Cria um arquivo temporário para salvar o download
temp_file <- tempfile(fileext = ".xlsx")

# Baixar o arquivo do GitHub
download.file(url = url_excel, destfile = temp_file,
               mode = "wb", quiet = TRUE)

# Listar as planilhas dentro do arquivo (opcional)
sheets <- excel_sheets(temp_file)
print(sheets)

# Importar planilha Paranaíba
Dados1 <- read_excel(temp_file, sheet = sheets[1])

#Limpar o arquivo temporário (opcional)
unlink(temp_file)

# Visualizar os dados
print(head(Dados2))

#chamar os pacotes necessários

library(remote)
remotes::install_github("lpradebon/DimExperimental")
library(readxl)
library(rio)
library(DimExp)

#####CURVATURA MÁXIMA EM FUNÇÃO DO CV

head(Dados1)
library(dplyr)
E1<-filter(Dados1,REP == "1")

A<-calc_paranaiba(dados      = E1,
                    nlin      = 6,
                    ncol      = 8,
                    nrep      = 1,
                    col_inicio = 4,
                    digits    = 3
)
)

```

```

E2<- filter(Dados1, REP=="2")

B<-calc_paranaiba(dados      = E2,
                    nlin       = 6,
                    ncol       = 8,
                    nrep       = 1,
                    col_inicio = 4,
                    digits     = 3
)
)

E3<- filter(Dados1, REP=="3")

C<-calc_paranaiba(dados      = E3,
                    nlin       = 6,
                    ncol       = 8,
                    nrep       = 1,
                    col_inicio = 4,
                    digits     = 3
)
)

#####
##### Exportar Resultados
library(rio)
export(list(E1=A, E2=B, E3=C),
       file = "Resultado_Paranaíba.xlsx")

#####
# Importar planilha Meier_Lessman
Dados2 <- read_excel(temp_file, sheet = sheets[2])

# Limpar o arquivo temporário (opcional)
unlink(temp_file)

# Visualizar os dados
print(head(Dados2))

#####
# MÁXIMA CURVATURA MODIFICADA
library(DimExp)
library(readxl)
library(tidyverse)

E1<-filter(Dados2, REP =="1")

D<-with(E1, cal_max_curv_mod(Xo, CV, plot = T))

E2<-filter(Dados2, REP =="2")

E<-with(E2, cal_max_curv_mod(Xo, CV, plot = T))

E3<-filter(Dados2, REP =="3")

```

```

G<-with(E3, cal_max_curv_mod(Xo, CV, plot = T))

##### MODELO LINEAR COM PLATÔ

#####ENSAIO1#####

#CV das parcelas simuladas
ps <- c(1, 2, 4, 8, 2, 4, 8, 16, 3, 6,
12, 24, 6, 12, 24)

#CV das parcelas simuladas
cv <- c(19.55309038, 15.23512639, 10.0533745, 8.203086387,
14.021032,
12.02564594, 7.707005924, 7.207523603, 11.51373415,
9.876254603,
11.59399511, 6.690432438, 7.504701872, 5.791576016,
3.084453732)

linear_plateau <- function(x, b0, b1, Xo) {
  ifelse(x < Xo, b0 + b1*x, b0 + b1*Xo)
}

fit <- nls(cv ~ linear_plateau(ps, b0, b1, Xo),
            start = list(b0 = max(cv), b1 = -1, Xo = 5),
            algorithm = "port",
            lower = c(-Inf, -Inf, min(ps)),
            upper = c(Inf, 0, max(ps)))

# Parâmetros
b0 <- coef(fit)[1]
b1 <- coef(fit)[2]
Xo <- coef(fit)[3]
Yo <- linear_plateau(Xo, b0, b1, Xo)

cv_pred <- predict(fit)
R2 <- 1 - sum((cv - cv_pred)^2) / sum((cv - mean(cv))^2)

plot(ps, cv, pch = 19, col = "blue",
      xlab = "Tamanho da parcela (m2)",
      ylab = "CV (%)",
      main = "Modelo Linear-Platô (Paranaíba, Ferreira, Morais, 2009)",
      xlim = c(0, max(ps) + 5),
      ylim = c(0, max(cv) + 6),
      xaxt = "n", yaxt = "n")

axis(2, at = seq(0, max(cv) + 6, by = 2))
axis(1, at = seq(0, max(ps) + 5, by = 2))

# Plota a reta da regressão
x_reg <- seq(0, Xo, length.out = 200)
lines(x_reg, b0 + b1*x_reg, lwd = 2)

```

```

# Plota a reta da regressão
segments(Xo, Yo, max(ps) + 5, Yo, col = "black", lwd = 2)

# Plota o ponto ótimo
points(Xo, Yo, pch = 19, col = "red")

# Plota as linhas do ponto ótimo
segments(0, Yo, Xo, Yo, col="red", lty=2)
segments(Xo, 0, Xo, Yo, col="red", lty=2)

eq_text <- paste0(
  "CV(X) = ", format(round(b0,3), decimal.mark=","),
  " - ", format(round(abs(b1),3), decimal.mark=","),
  "X se X ≤ ",
  format(round(Xo,2), decimal.mark=","),
  "\n",
  "CV(X) = ", format(round(Yo,2), decimal.mark=","),
  " se X > ",
  format(round(Xo,2), decimal.mark=","),
  "\n",
  "R² = ", format(round(R2,2), decimal.mark=","))
)

lim <- par("usr")

# Plota o texto
text(
  x = (lim[1] + lim[2]) / 2,
  y = lim[4] * 0.88,
  labels = eq_text,
  cex = 1.1
)

#####ensaio 2#####
#Tamanhos de parcelas simulados
ps <- c(1, 2, 4, 8, 2, 4, 8, 16, 3, 6,
12, 24, 6, 12, 24)

#CV das parcelas simuladas
cv <- c(18.27515113, 13.51272155, 9.549630661, 10.40275337,
12.26072061,
7.300500267, 4.955269561, 0.525821357, 8.521396194,
5.621487839,
3.750894746, 3.359700677, 5.364230744, 3.748309565,
1.607065185)

linear_plateau <- function(x, b0, b1, Xo) {
  ifelse(x < Xo, b0 + b1*x, b0 + b1*Xo)
}

fit <- nls(cv ~ linear_plateau(ps, b0, b1, Xo),
           start = list(b0 = max(cv), b1 = -1, Xo = 5),
           algorithm = "port",
           lower = c(-Inf, -Inf, min(ps)),
           upper = c(Inf, 0, max(ps)))

# Parâmetros

```

```

b0 <- coef(fit)[1]
b1 <- coef(fit)[2]
Xo <- coef(fit)[3]
Yo <- linear_plateau(Xo, b0, b1, Xo)

cv_pred <- predict(fit)
R2 <- 1 - sum((cv - cv_pred)^2) / sum((cv - mean(cv))^2)

plot(ps, cv, pch = 19, col = "blue",
      xlab = "Tamanho da parcela (m2)",
      ylab = "CV (%)",
      main = "Modelo Linear-Platô (Paranaíba, Ferreira, Morais, 2009)",
      xlim = c(0, max(ps) + 5),
      ylim = c(0, max(cv) + 6),
      xaxt = "n", yaxt = "n")

axis(2, at = seq(0, max(cv) + 6, by = 2))
axis(1, at = seq(0, max(ps) + 5, by = 2))

# Plota a reta de regressão
x_reg <- seq(0, Xo, length.out = 200)
lines(x_reg, b0 + b1*x_reg, lwd = 2)

# Plota o platô
segments(Xo, Yo, max(ps) + 5, Yo, col = "black", lwd = 2)

# Plota o ponto ótimo
points(Xo, Yo, pch = 19, col = "red")

# Linhas do ponto ótimo
segments(0, Yo, Xo, Yo, col="red", lty=2)
segments(Xo, 0, Xo, Yo, col="red", lty=2)

eq_text <- paste0(
  "CV(X) = ", format(round(b0,3), decimal.mark=","),
  " - ", format(round(abs(b1),3), decimal.mark=","),
  "X se X ≤ ",
  format(round(Xo,2), decimal.mark=","),
  "\n",
  "CV(X) = ", format(round(Yo,2), decimal.mark=","),
  " se X > ",
  format(round(Xo,2), decimal.mark=","),
  "\n",
  "R2 = ", format(round(R2,2), decimal.mark=","))
)

lim <- par("usr")

# Plotar o texto
text(
  x = (lim[1] + lim[2]) / 2,
  y = lim[4] * 0.88,
  labels = eq_text,
  cex = 1.1
)

#####ensaio3#####

```

```

#Tamanhos de parcelas simulados
ps <- c(1,      2,      4,      8, 2,    4,      8,      16,      3,      6,
12,     24,     6,     12,     24)

#CV das parcelas simuladas
cv <- c(15.8871252,      11.27495015,      8.934562939,      4.356418236,
10.79152267,      7.487783047,      6.032368718,      4.819426449,      9.499316313,
5.707748363,      10.62093603,      0.596433924,      7.391978765,      5.544005975,
4.214524446)

linear_plateau <- function(x, b0, b1, Xo) {
  ifelse(x < Xo, b0 + b1*x, b0 + b1*Xo)
}

fit <- nls(cv ~ linear_plateau(ps, b0, b1, Xo),
            start = list(b0 = max(cv), b1 = -1, Xo = 5),
            algorithm = "port",
            lower = c(-Inf, -Inf, min(ps)),
            upper = c(Inf, 0, max(ps)))

# Parâmetros
b0 <- coef(fit)[1]
b1 <- coef(fit)[2]
Xo <- coef(fit)[3]
Yo <- linear_plateau(Xo, b0, b1, Xo)

cv_pred <- predict(fit)
R2 <- 1 - sum((cv - cv_pred)^2) / sum((cv - mean(cv))^2)

plot(ps, cv, pch = 19, col = "blue",
      xlab = "Tamanho da parcela (m2)",
      ylab = "CV (%)",
      main = "Modelo Linear-Platô (Paranaíba, Ferreira, Morais, 2009)",
      xlim = c(0, max(ps) + 5),
      ylim = c(0, max(cv) + 6),
      xaxt = "n", yaxt = "n")

axis(2, at = seq(0, max(cv) + 6, by = 2))
axis(1, at = seq(0, max(ps) + 5, by = 2))

# Plota a reta da regressão
x_reg <- seq(0, Xo, length.out = 200)
lines(x_reg, b0 + b1*x_reg, lwd = 2)

# Plota o platô
segments(Xo, Yo, max(ps) + 5, Yo, col = "black", lwd = 2)

#Plota o ponto ótimo
points(Xo, Yo, pch = 19, col = "red")

# Plota as linhas do ponto ótimo

```

```

segments(0, Yo, Xo, Yo, col="red", lty=2)
segments(Xo, 0, Xo, Yo, col="red", lty=2)

eq_text <- paste0(
  "CV(X) = ", format(round(b0,3), decimal.mark=","),
  " - ", format(round(abs(b1),3), decimal.mark=","), "X se X ≤ ",
  format(round(Xo,2), decimal.mark=",","\n"),
  "CV(X) = ", format(round(Yo,2), decimal.mark=",","\n"), " se X > ",
  format(round(Xo,2), decimal.mark=",","\n"),
  "R² = ", format(round(R2,2), decimal.mark=","))
)

# coordenadas do gráfico
lim <- par("usr")
# Plota o texto
text(
  x = (lim[1] + lim[2]) / 2,
  y = lim[4] * 0.88,
  labels = eq_text,
  cex = 1.1
)

##### Número de repetições #####
cv <- 8.92
alpha <- 0.05
tratamentos <- 3:50
DMS <- c(10, 20, 30, 40, 50)

#####
#Cálculo para DIC
DIC<-calc_repeticoes(tratamentos, cv, DMS, design = "DIC")

#####
#Calculo para DBC
DBC<-calc_repeticoes(tratamentos, cv, DMS, design = "DBC")
#####

library(rio)
export(list(DIC=DIC, DBC=DBC), file = "resultado_nrep.xlsx")

```