

```

library(readxl)

#link para o arquivo Excel no repositório
url_excel <-
"https://raw.githubusercontent.com/lpradebon/Quest-es_Prof_Fernando_Machado_Haes
baert/main/3_1_Simulacoes_AR1_TMG31B3RF2.xlsx"

temp_file <- tempfile(fileext = ".xlsx")

download.file(url = url_excel, destfile = temp_file, mode = "wb", quiet = TRUE)

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

#Chama a planilha de dados para o rho 0
C0 <- read_excel(temp_file, sheet = sheets[1])

unlink(temp_file)

library(rio)
library(tidyverse)

#####
##### Rho 0,00 #####
#####

# Visualizar dados
head(C0)

calc_paranaiba <- function(dados,
                           nlin,
                           nrep,
                           col_inicio = 3,
                           digits = 3) {

# Cálculo para um bloco (1 repetição)
calculaBloco <- function(x) {

# Estatísticas básicas
  media <- mean(x)
  desvio <- sd(x)
  cv <- (desvio / media) * 100
  var1 <- desvio^2

# Erro em relação à média
  err <- x - media
  soma2 <- sum(err^2)

# Caminhamento em LINHA
  erro_linha <- c()
  for (i in 1:nrow(x)) {
    if (i %% 2 == 1) {
      erro_linha <- c(erro_linha, err[i, ])
    } else {
      erro_linha <- c(erro_linha, err[i, ncol(x):1])
    }
  }
  return(list(media = media, desvio = desvio, cv = cv, var1 = var1, err = err, soma2 = soma2, erro_linha = erro_linha))
}

return(calculaBloco)
}

```

```

        }

# rho apenas no sentido linha
rho_l <- sum(
  erro_linha[-1] * erro_linha[-length(erro_linha)]
) / soma2

# Tamanho ótimo da parcela
tamanho_otimo <- 10 * (2 * (1 - rho_l^2) * var1 * media)^(1/3) / media

# CV do tamanho ótimo
cv_otimo <- 100 *
  sqrt((1 - rho_l^2) * var1 / media^2) /
  sqrt(tamanho_otimo)

data.frame(
  Estatistica = c(
    "média",
    "rho_l",
    "Tamanho ótimo da parcela",
    "Coeficiente de variação (%)",
    "Coeficiente de variação da parcela ótima (%)"
  ),
  Valor = c(
    media,
    rho_l,
    tamanho_otimo,
    cv,
    cv_otimo
  ),
  stringsAsFactors = FALSE
)
}

# Loop nas repetições
resultados <- vector("list", nrep)

for (rep in 1:nrep) {

  lin_ini <- (rep - 1) * nlin + 1
  lin_fim <- rep * nlin

  bloco <- matrix(
    unlist(
      dados[lin_ini:lin_fim, col_inicio:(col_inicio + 7)]
    ),
    nrow = nlin,
    byrow = FALSE
  )

  res <- calcula_bloco(bloco)
  res$Repeticao <- rep
}

```

```

resultados[[rep]] <- res
}

tabela_final <- do.call(rbind, resultados)
tabela_final$Valor <- round(tabela_final$Valor, digits)
rownames(tabela_final) <- NULL

return(tabela_final)
}

C0_result <- calc_paranaiba(
  dados      = C0,
  nlin       = 18,
  nrep       = 1,
  col_inicio = 3,
  digits     = 3
)
C0_result

#####
##### Rho 0,20 #####
#Chama a planilha de dados para o rho 0,20
C20 <- read_excel(temp_file, sheet = sheets[2])

unlink(temp_file)

# Visualizar dados
head(C20)

calc_paranaiba <- function(dados,
                           nlin,
                           nrep,
                           col_inicio = 3,
                           digits = 3) {

# Cálculo para um bloco (1 repetição)
calcula_bloco <- function(x) {

# Estatísticas básicas
  media   <- mean(x)
  desvio  <- sd(x)
  cv      <- (desvio / media) * 100
  var1    <- desvio^2

# Erro em relação à média
  err     <- x - media
  soma2 <- sum(err^2)

# Caminhamento em LINHA
  erro_linha <- c()
  for (i in 1:nrow(x)) {
    if (i %% 2 == 1) {

```

```

        erro_linha <- c(erro_linha, err[i, ])
    } else {
        erro_linha <- c(erro_linha, err[i, ncol(x):1])
    }
}

# rho apenas no sentido linha
rho_1 <- sum(
    erro_linha[-1] * erro_linha[-length(erro_linha)])
) / soma2

# Tamanho ótimo da parcela
tamanho_otimo <- 10 * (2 * (1 - rho_1^2) * var1 * media)^(1/3) / media

# CV do tamanho ótimo
cv_otimo <- 100 *
sqrt((1 - rho_1^2) * var1 / media^2) /
sqrt(tamanho_otimo)

data.frame(
    Estatistica = c(
        "média",
        "rho_1",
        "Tamanho ótimo da parcela",
        "Coeficiente de variação (%)",
        "Coeficiente de variação da parcela ótima (%)"
    ),
    Valor = c(
        media,
        rho_1,
        tamanho_otimo,
        cv,
        cv_otimo
    ),
    stringsAsFactors = FALSE
)
}

# Loop nas repetições

resultados <- vector("list", nrep)

for (rep in 1:nrep) {

    lin_ini <- (rep - 1) * nlin + 1
    lin_fim <- rep * nlin

    bloco <- matrix(
        unlist(
            dados[lin_ini:lin_fim, col_inicio:(col_inicio + 7)]
        ),
        nrow = nlin,
        byrow = FALSE
    )
}

```

```

)
res <- calcula_bloco(bloco)
res$Repeticao <- rep

resultados[[rep]] <- res
}

tabela_final <- do.call(rbind, resultados)
tabela_final$Valor <- round(tabela_final$Valor, digits)
rownames(tabela_final) <- NULL

return(tabela_final)
}

C20_result <- calc_paranaiba(
  dados      = C20,
  nlin       = 18,
  nrep       = 1,
  col_inicio = 3,
  digits     = 3
)
C20_result

#####
##### Rho 0,50 #####
#####

#Chama a planilha de dados para o rho 0,50
C50 <- read_excel(temp_file, sheet = sheets[3])

unlink(temp_file)
# Visualizar dados
head(C50)

calc_paranaiba <- function(dados,
                           nlin,
                           nrep,
                           col_inicio = 3,
                           digits = 3) {

# Cálculo para um bloco (1 repetição)
calcula_bloco <- function(x) {

# Estatísticas básicas
  media  <- mean(x)
  desvio <- sd(x)
  cv     <- (desvio / media) * 100
  var1   <- desvio^2

# Erro em relação à média
  err    <- x - media
  soma2 <- sum(err^2)
}

```

```

# Caminhamento em LINHA
  erro_linha <- c()
  for (i in 1:nrow(x)) {
    if (i %% 2 == 1) {
      erro_linha <- c(erro_linha, err[i, ])
    } else {
      erro_linha <- c(erro_linha, err[i, ncol(x):1])
    }
  }

# rho no sentido linha
  rho_l <- sum(
    erro_linha[-1] * erro_linha[-length(erro_linha)])
  ) / soma2

# Tamanho ótimo da parcela
  tamanho_otimo <- 10 * (2 * (1 - rho_l^2) * var1 * media)^(1/3) / media

# CV do tamanho ótimo
  cv_otimo <- 100 *
    sqrt((1 - rho_l^2) * var1 / media^2) /
    sqrt(tamanho_otimo)

data.frame(
  Estatistica = c(
    "média",
    "rho_l",
    "Tamanho ótimo da parcela",
    "Coeficiente de variação (%)",
    "Coeficiente de variação da parcela ótima (%)"
  ),
  Valor = c(
    media,
    rho_l,
    tamanho_otimo,
    cv,
    cv_otimo
  ),
  stringsAsFactors = FALSE
)
}

# Loop nas repetições

resultados <- vector("list", nrep)

for (rep in 1:nrep) {

  lin_ini <- (rep - 1) * nlin + 1
  lin_fim <- rep * nlin
}

```

```

bloco <- matrix(
  unlist(
    dados[lin_ini:lin_fim, col_inicio:(col_inicio + 7)]
  ),
  nrow = nlin,
  byrow = FALSE
)

res <- calcula_bloco(bloco)
res$Repeticao <- rep

resultados[[rep]] <- res
}

tabela_final <- do.call(rbind, resultados)
tabela_final$Valor <- round(tabela_final$Valor, digits)
rownames(tabela_final) <- NULL

return(tabela_final)
}

C50_result <- calc_paranaiba(
  dados      = C50,
  nlin       = 18,
  nrep       = 1,
  col_inicio = 3,
  digits     = 3
)
C50_result

#####
#Chama a planilha de dados para o rho 0,80 #####
C80 <- read_excel(temp_file, sheet = sheets[4])

unlink(temp_file)

# Visualizar dados
head(C80)

calc_paranaiba <- function(dados,
                           nlin,
                           nrep,
                           col_inicio = 3,
                           digits = 3) {

# Cálculo para um bloco (1 repetição)
calcula_bloco <- function(x) {

```

```

# Estatísticas básicas
  media  <- mean(x)
  desvio <- sd(x)
  cv      <- (desvio / media) * 100
  var1   <- desvio^2

# Erro em relação à média
  err    <- x - media
  soma2 <- sum(err^2)

  erro_linha <- c()
  for (i in 1:nrow(x)) {
    if (i %% 2 == 1) {
      erro_linha <- c(erro_linha, err[i, ])
    } else {
      erro_linha <- c(erro_linha, err[i, ncol(x):1])
    }
  }

# rho apenas no sentido linha
  rho_1 <- sum(
    erro_linha[-1] * erro_linha[-length(erro_linha)])
  ) / soma2

# Tamanho ótimo da parcela com base em rho_1
  tamanho_otimo <- 10 * (2 * (1 - rho_1^2) * var1 * media)^(1/3) / media

# CV do tamanho ótimo
  cv_otimo <- 100 *
    sqrt((1 - rho_1^2) * var1 / media^2) /
    sqrt(tamanho_otimo)

  data.frame(
    Estatistica = c(
      "média",
      "rho_1",
      "Tamanho ótimo da parcela",
      "Coeficiente de variação (%)",
      "Coeficiente de variação da parcela ótima (%)"
    ),
    Valor = c(
      media,
      rho_1,
      tamanho_otimo,
      cv,
      cv_otimo
    ),
    stringsAsFactors = FALSE
  )
}

resultados <- vector("list", nrep)

```

```

for (rep in 1:nrep) {

  lin_ini <- (rep - 1) * nlin + 1
  lin_fim <- rep * nlin

  bloco <- matrix(
    unlist(
      dados[lin_ini:lin_fim, col_inicio:(col_inicio + 7)]
    ),
    nrow = nlin,
    byrow = FALSE
  )

  res <- calcula_bloco(bloco)
  res$Repeticao <- rep

  resultados[[rep]] <- res
}

tbla_final <- do.call(rbind, resultados)
tbla_final$Valor <- round(tbla_final$Valor, digits)
rownames(tbla_final) <- NULL

return(tbla_final)
}

C80_result <- calc_paranaiba(
  dados      = C80,
  nlin       = 18,
  nrep       = 1,
  col_inicio = 3,
  digits     = 3
)

C80_result
#####Exportar dados
library(rio)
export(list(C0_result=C0_result, C20_result=C20_result, C50_result=C50_result,
C80_result=C80_result),
      file = "3_2_Resultado_X0_AR(1).xlsx")

#Número ótimo de repetições para cada cenário de autocorrelação
library(DimExp)
C0_result
cv <- 13.39
alpha <- 0.05
tratamentos <- c(4,6,8,10)
DMS <- c(10)

P0_DIC <- calc_repeticoes(

```

```
tratamentos = tratamentos,
CV_percent  = cv,
d_percent   = DMS,
alpha        = alpha,
design       = "DIC",
plot         = TRUE
)
```

P0\_DIC

```
P0_DBC <- calc_repeticoes(
tratamentos = tratamentos,
CV_percent  = cv,
d_percent   = DMS,
alpha        = alpha,
design       = "DBC",
plot         = TRUE
)
```

P0\_DBC

```
#  
C20_result  
cv <- 13.10  
alpha <- 0.05  
tratamentos <- c(4,6,8,10)  
DMS <- c(10)
```

```
P20_DIC <- calc_repeticoes(
tratamentos = tratamentos,
CV_percent  = cv,
d_percent   = DMS,
alpha        = alpha,
design       = "DIC",
plot         = TRUE
)
```

P20\_DIC

```
P20_DBC <- calc_repeticoes(
tratamentos = tratamentos,
CV_percent  = cv,
d_percent   = DMS,
alpha        = alpha,
design       = "DBC",
plot         = TRUE
)
```

P20\_DBC

```
#  
C50_result  
cv <- 13.52
```

```
alpha <- 0.05
tratamentos <- c(4,6,8,10)
DMS <- c(10)

P50_DIC <- calc_repeticoes(
  tratamentos = tratamentos,
  CV_percent = cv,
  d_percent = DMS,
  alpha = alpha,
  design = "DIC",
  plot = TRUE
)
```

```
P50_DIC
```

```
P50_DBC <- calc_repeticoes(
  tratamentos = tratamentos,
  CV_percent = cv,
  d_percent = DMS,
  alpha = alpha,
  design = "DBC",
  plot = TRUE
)
```

```
P50_DBC
```

```
#  
C80_result  
cv <- 13.26  
alpha <- 0.05  
tratamentos <- c(4,6,8,10)  
DMS <- c(10)
```

```
P80_DIC <- calc_repeticoes(
  tratamentos = tratamentos,
  CV_percent = cv,
  d_percent = DMS,
  alpha = alpha,
  design = "DIC",
  plot = TRUE
)
```

```
P80_DIC
```

```
P80_DBC <- calc_repeticoes(
  tratamentos = tratamentos,
  CV_percent = cv,
  d_percent = DMS,
  alpha = alpha,
  design = "DBC",
  plot = TRUE
)
```

P80\_DBC

```
export(list(P0_DIC=P0_DIC,
           P0_DBC=P0_DBC,
           P20_DIC=P20_DIC,
           P20_DBC=P20_DBC,
           P50_DIC=P50_DIC,
           P50_DBC=P50_DBC,
           P80_DIC=P80_DIC,
           P80_DBC=P80_DBC), file = "Número de repetições variando
tratamento_p_delineamento .xlsx")

##### heatmap #####
if (!requireNamespace("agronomymap", quietly = TRUE)) {
  remotes::install_github("agronomy4future/agronomymap", force = TRUE)
}
library(readxl)
library(remotes)
library(agronomymap)
library(ggplot2)

#link para o arquivo Excel no repositório
url_excel <-
"https://raw.githubusercontent.com/lpradebon/Quest-es_Prof_Fernando_Machado_Haes
baert/main/3_5Heats_maps.xlsx"
temp_file <- tempfile(fileext = ".xlsx")

download.file(url = url_excel, destfile = temp_file, mode = "wb")
if (!file.exists(temp_file)) stop("Download falhou.")

#Lista as planilhas dentro do arquivo
abas <- excel_sheets(temp_file)
print(abas)

#Chama a planilha de dados
C0 <- lapply(abas, function(a) read_excel(temp_file, sheet = a))
names(C0) <- abas
unlink(temp_file)

#Seleciona a primeira planilha como data.frame
df <- C0[[1]]

df[] <- lapply(df, function(col) suppressWarnings(as.numeric(col)))

X <- unlist(df)
X <- X[is.finite(X)]

if (length(X) == 0) stop("Nenhum valor numérico válido encontrado na aba.")

##### Plotar heatmap_rho_0 #####
agronomymap(df,
            map = c("LINHA", "COLUNA"),
            variable = "C0_NCP",
```

```

grid_res = 100,
palette = "Blues",
fill_limits = c(0, 4),
fill_breaks = seq(0, 4, 1),
label_title = "NCP (unidades)",
label_size = 12,
label_key = 12,
label_position = "right",
label_key_height = 1.2,
label_key_width = 0.7,
xlab = "LINHAS",
ylab = "COLUNAS",
point_size = 1,
point_shape = 21,
point_color = "red",
point_fill = "grey25",
add_border = TRUE,
axis_units = TRUE) +
coord_flip() +
labs(
  title = "Correlação espacial - p = 0",
  x = "Linhas",
  y = "Colunas"
)

##### Heatmap_rho_0,20 #####
agronomymap(df,
  map = c("LINHA", "COLUNA"),
  variable = "C020_NCP",
  grid_res = 100,
  palette = "Blues",
  fill_limits = c(0, 4),
  fill_breaks = seq(0, 4, 1),
  label_title = "NCP (unidades)",
  label_size = 12,
  label_key = 12,
  label_position = "right",
  label_key_height = 1.2,
  label_key_width = 0.7,
  xlab = "LINHAS",
  ylab = "COLUNAS",
  point_size = 1,
  point_shape = 21,
  point_color = "red",
  point_fill = "grey25",
  add_border = TRUE,
  axis_units = TRUE) +
coord_flip() +
labs(
  title = "Correlação espacial - p = 0,20",
  x = "Linhas",
  y = "Colunas"
)

```

```

##### Heatmap_rho_0,50 #####
agronomymap(df,
  map = c("LINHA", "COLUNA"),
  variable = "C050_NCP",
  grid_res = 100,
  palette = "Blues",
  fill_limits = c(0, 4),
  fill_breaks = seq(0, 4, 1),
  label_title = "NCP (unidades)",
  label_size = 12,
  label_key = 12,
  label_position = "right",
  label_key_height = 1.2,
  label_key_width = 0.7,
  xlab = "LINHAS",
  ylab = "COLUNAS",
  point_size = 1,
  point_shape = 21,
  point_color = "red",
  point_fill = "grey25",
  add_border = TRUE,
  axis_units = TRUE) +
coord_flip() +
labs(
  title = "Correlação espacial - p = 0,50",
  x = "Linhas",
  y = "Colunas"
)

##### Heatmap_rho_0,80 #####
agronomymap(df,
  map= c("LINHA","COLUNA"),
  variable= "C080_NCP",
  grid_res = 100,
  palette= "Blues",
  fill_limits= c(0, 4),
  fill_breaks= seq(0, 4, 1),
  # label
  label_title= "NCP (unidades)",
  label_size= 12,
  label_key= 12,
  label_position= "right",
  label_key_height= 1.2,
  label_key_width= 0.7,
  xlab= "LINHAS",
  ylab= "COLUNAS",
  # point aesthetics
  point_size= 1,
  point_shape= 21,
  point_color= "red",
  point_fill= "grey25",
  #border and axis unit
  add_border= TRUE,

```

```
axis_units= TRUE)+coord_flip()+
labs(
  title = "Correlação espacial - p = 0,80",
  x = "Linhas",
  y = "Colunas")
```