

exploratory.R

```
#####
# 0. Carregar subset das máquinas
#####

load_machines_subset <- function() {
  library(rrcov)
  data("machines")

  start <- which(rownames(machines) == "hp-3000/64")
  end   <- which(rownames(machines) == "ibm-4331-2")

  subset <- machines[start:end, ]
  return(subset)
}

subset <- load_machines_subset()

#####
# 1. Exploratory Data Analysis (EDA)
#####

library(ggplot2)
library(GGally)
library(psych)
library(corrplot)

# Remover os rownames para facilitar gráficos
df <- subset
df$Machine <- rownames(df)
rownames(df) <- NULL

#####
# 1.1 Estatísticas descritivas
#####

# Summary
summary(df)

# Média
means <- colMeans(df[ , -ncol(df)])
means

# Mediana
medians <- apply(df[ , -ncol(df)], 2, median)
medians

# Trimmed mean (10%)
trimmed_means <- apply(df[ , -ncol(df)], 2, mean, trim = 0.1)
trimmed_means

# Winsorized mean (10%)
winsor_means <- apply(df[ , -ncol(df)], 2, winsor.mean)
winsor_means

# Variâncias
variances <- apply(df[ , -ncol(df)], 2, var)
```

exploratory.R

variances

```
# MAD (Median Absolute Deviation)
mads <- apply(df[ , -ncol(df)], 2, mad)
mads

#####
# 1.2 Covariância, Variância Total, Generalized Variance
#####

S <- cov(df[ , -ncol(df)]) # matriz de covariância

# Variância Total = soma das variâncias
total_variance <- sum(diag(S))
total_variance

# Generalized Variance (determinante)
generalized_variance <- det(S)
generalized_variance

#####
# 1.3 Distâncias de Mahalanobis
#####

center <- colMeans(df[ , -ncol(df)])
md <- mahalanobis(df[ , -ncol(df)], center, S)

df$Mahalanobis <- md
md

# Outliers potenciais (nível 97.5%)
cutoff <- qchisq(0.975, df = ncol(df)-1)
which(md > cutoff)

#####
# 1.4 Gráficos
#####

# Histogramas
par(mfrow=c(3,3))
for(col in names(df)[-c(ncol(df), length(names(df)))]){
  hist(df[[col]], main=paste("Histograma de", col), xlab=col)
}

# Boxplots
par(mfrow=c(3,3))
for(col in names(df)[-c(ncol(df), length(names(df)))]){
  boxplot(df[[col]], main=paste("Boxplot de", col))
}

# Scatterplot matrix
GGally::ggpairs(df[, sapply(df, is.numeric)])

# Correlograma
corrplot(cor(df[sapply(df, is.numeric)]),
         method = "color", addCoef.col = "black")
```

exploratory.R

