#########################################################

#IBI5086-2023

#Introdução a Métodos Estatísticos para a Bioinformática

#Profa. Júlia Maria Pavan Soler

#Comandos básicos do R para apoio à Aula2

#########################################################

#Lendo os dados PULSE vistos em Aula

dados = read.table('D:\\IBI5086-2023\\Aula2-170823\\pulse.csv', head=T, sep=";", dec=",")

dados

head(dados)

str(dados)

attach(dados)

names(dados)

dados[,1:3]

summary(dados[,1:2])

table(Ran) #Ran=1: correu Ran=2: não correu

table(Ran,Sex)

#Estatísticas resumo (descritivas) da Pulsação em repouso de Estudantes

mean(P1) #média amostral

sd(P1) #desvio padrão

sd(P1)/sqrt(length(P1)) #erro padrão

library(psych)

describe(P1) #pulsação em repouso

describe(P2) #Não faz sentido, pois há estudantes com Ran=1 e Ran=2

describe.by(P1,factor(Ran)) #estatísticas descritivas

describe.by(P2,factor(Ran))

#coeficiente de variação (CV: medida de variabilidade adimensional)

sd(P1[Ran==1])/mean(P1[Ran==1])

sd(P1[Ran==2])/mean(P1[Ran==2])

sd(P2[Ran==1])/mean(P2[Ran==1])

sd(P2[Ran==2])/mean(P2[Ran==2])

boxplot(P1[Ran==1],P2[Ran==1],P1[Ran==2],P2[Ran==2],

main="Boxplot Pulsação",

at=c(1,2,4,5),

names=c("P1:Ran1","P2:Ran1","P1:Ran2","P2:Ran2"),

col=c("red","red","white", "white"))

boxplot(Altura[Ran==1],Altura[Ran==2],

main="Boxplot Altura",

names=c("Ran1","Ran2"),

col=c("red","white")) #Há observações atípicas?

boxplot(Altura[Ran==1],plot=F)$out

boxplot(Peso[Ran==1],Peso[Ran==2],

main="Boxplot Peso",

names=c("Ran1","Ran2"),

col=c("red","white"))

bp<-boxplot(Peso[Ran==2],plot=F)

names(bp)

bp$stats

bp$conf

bp$names

fu<-table(Ran,Fu)

fu

install.packages("tigerstats")

library(tigerstats)

rowPerc(fu)

sx<-table(Ran,Sex)

sx

rowPerc)sx)

at<-table(Ran,Ativ)

at

rowPerc(at)

#Convertendo o formato dos dados

#Pulse está em "wide format"

#P1 P2 ...

head(dados)

dat.wide<-dados

dat.wide

names(dat.wide)

library(reshape2)

#Convertendo do formato wide para o long

dat.long <- melt(dat.wide,id.vars=c("Ran","Fu","Sex","Altura", "Peso", "Ativ"))

head(dat.long)

#adicionando o identificador de indivíduos (row.names)

id<-c(row.names(dat.wide))

dat.wide.id<-cbind(id,dat.wide)

head(dat.wide.id)

dat.long.id <- melt(dat.wide.id,id.vars=c("Ran","Fu","Sex","Altura", "Peso", "Ativ","id"))

dat.long.id

#Convertendo do formato wide para long

dat.wide <- dcast(dat.long, Ran + Fu + Sex + Altura + Peso + Ativ ~ variable, value.var="value")

head(dat.wide)

#Intervalos de Concentração dos dados

#e Intervalos de Confiança para o parâmetro mi (Média populacional da resposta)

#Cálculos para o grupo Ran=1

lRan1=c(mean(P1[Ran==1])-2\*sd(P1[Ran==1]),mean(P2[Ran==1])-2\*sd(P2[Ran==1]))

uRan1=c(mean(P1[Ran==1])+2\*sd(P1[Ran==1]),mean(P2[Ran==1])+2\*sd(P2[Ran==1]))

medRan1=c(mean(P1[Ran==1]),mean(P2[Ran==1]))

lRan1

uRan1

medRan1

#Interprete o intervalo de valores!

require(sciplot)

#Intervalos de Concentração dos Dados

#Sob distribuição aprox Normal, dos dados,

#Média -/+ 2\*desvio padrão

#É esperado que aproximadamente 95% dos valores centrais da amostra estejam neste Intervalo

par(mfrow=c(1,2))

lineplot.CI(factor(dat.long$variable[Ran==1]), dat.long$value[Ran==1], type="p", las=1,

xlab="Ran=1", ylab="Pulse", main="Média e 2\*sd",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*sd(x), mean(x)+2\*sd(x)))

lineplot.CI(factor(dat.long$variable[Ran==2]), dat.long$value[Ran==2], type="p", las=1,

xlab="Ran=2", ylab="Pulse", main="Média e 2\*sd",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*sd(x), mean(x)+2\*sd(x)))

#Intervalos a 95% de Confiança para a Média Populacional da resposta sob estudo

par(mfrow=c(1,2))

lineplot.CI(factor(dat.long$variable[Ran==1]), dat.long$value[Ran==1], type="p", las=1,

xlab="Ran=1", ylab="Pulse", main="Média e 2\*se",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*se(x), mean(x)+2\*se(x)))

lineplot.CI(factor(dat.long$variable[Ran==2]), dat.long$value[Ran==2], type="p", las=1,

xlab="Ran=2", ylab="Pulse", main="Média e 2\*se",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*se(x), mean(x)+2\*se(x)))

#Explorando a estrutura de Amostras pareadas (dados longitudinais: P1 e P2)

#Variável Diferença entre as Pulsações

d<-P2-P1

d

cbind(Ran,P1,P2,d)

par(mfrow=c(1,2))

lineplot.CI(factor(Ran), d, type="p", las=1,

xlab="Ran", ylab="Diferença d", main="Dif\_Média e 2\*sd",

ylim=c(-20, 50),

ci.fun= function(x) c(mean(x)-2\*sd(x), mean(x)+2\*sd(x)))

lineplot.CI(factor(Ran), d, type="p", las=1,

xlab="Ran", ylab="Diferença d", main="Dif\_Média e 2\*se",

ylim=c(-20, 50),

ci.fun= function(x) c(mean(x)-2\*se(x), mean(x)+2\*se(x)),

abline(h=0, col = "lightgray"))

abline(h=0, col = "lightgray")

#Interprete os resultados para a variável diferença

#Compare com a análise de P2 a seguir

lineplot.CI(factor(Ran), P2, type="p", las=1,

xlab="Ran", ylab="P2", main="Média e 2\*se",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*se(x), mean(x)+2\*se(x)))

#Considerando Dados Independentes (35 estudantes independentes dos demais 57)

par(mfrow=c(1,2))

lineplot.CI(factor(Ran), P1, type="p", las=1,

xlab="Ran", ylab="P1", main="Média e 2\*sd",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*sd(x), mean(x)+2\*sd(x)))

lineplot.CI(factor(Ran), P2, type="p", las=1,

xlab="Ran", ylab="P2", main="Média e 2\*sd",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*sd(x), mean(x)+2\*sd(x)))

#Considerando Dados Independentes (35 estudantes independentes dos demais 57)

par(mfrow=c(1,2))

lineplot.CI(factor(Ran), P1, type="p", las=1,

xlab="Ran", ylab="P2", main="Média e 2\*se",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*se(x), mean(x)+2\*se(x)))

lineplot.CI(factor(Ran), P2, type="p", las=1,

xlab="Ran", ylab="P2", main="Média e 2\*se",

ylim=c(50,135),

ci.fun= function(x) c(mean(x)-2\*se(x), mean(x)+2\*se(x)))

#Perfis de Médias de Pulse de acordo com Ran1 e Ran2

interaction.plot(dat.long$variable,dat.long$Ran,dat.long$value, type="b", col=c(1:3),

leg.bty="o", leg.bg="beige", lwd=2, pch=c(18,24,22),

xlab="Ran",

ylab="Pulse",

main="Perfis de Médias-Interação")

#Agora vamos simular dados sob a estrutura de PULSE

#comando rnorm: simula dados da Normal univariada (uma única variável)

#comando mvrnorm: simula dados da Normal multivariada (para "p" variáveis correlacionadas)

#Discuta os 3 diferentes cenários de Simulação a seguir

#Qual você adotaria?

#Cenário 1

n<-92

mi1<-mean(P1)

sigma1<-sd(P1)

#set.seed(47995)

P1sim<-rnorm(n,mi1,sigma1)

P1sim

boxplot(P1,P1sim)

mi2<-mean(P2)

sigma2<-sd(P2)

P2sim<-rnorm(n,mi2,sigma2)

P2sim

boxplot(P2,P2sim)

Ransim<-c(rep(1,35),rep(2,57))

Ransim

datsim1<-data.frame(P1sim, P2sim, Ransim)

datsim1

#Cenário 2

n1<-35

n2<-57

mi1.1<-mean(P1[Ran==1])

sigma1.1<-sd(P1[Ran==1])

mi1.2<-mean(P1[Ran==2])

sigma1.2<-sd(P1[Ran==2])

P1.1sim<-rnorm(n1,mi1.1,sigma1.1)

P1.2sim<-rnorm(n2,mi1.2,sigma1.2)

n1<-35

n2<-57

mi2.1<-mean(P2[Ran==1])

sigma2.1<-sd(P2[Ran==1])

mi2.2<-mean(P2[Ran==2])

sigma2.2<-sd(P2[Ran==2])

P2.1sim<-rnorm(n1,mi2.1,sigma2.1)

P2.2sim<-rnorm(n2,mi2.2,sigma2.2)

par(mfrow=c(2,2))

boxplot(P1[Ran==1],P1.1sim, main="P1\_Ran=1")

boxplot(P2[Ran==1],P2.1sim, main="P2\_Ran=1")

boxplot(P1[Ran==2],P1.2sim, main="P1\_Ran=2")

boxplot(P2[Ran==2],P2.2sim, main="P2\_Ran=2")

datsim2<-data.frame(matrix(c(P1.1sim, P1.2sim, P2.1sim, P2.2sim),92,2))

datsim2<-cbind(datsim2,Ransim)

datsim2

#Cenário 3

#Coeficiente de correlação entre P1 e P2 para Ran=1

cor1<-cor(P1[Ran==1],P2[Ran==1])

cor1

#Cor(x,y) = cov(x,y)/sigma(x)\*sigma(y)

#Coeficiente de covariância

cov1<-cov(P1[Ran==1],P2[Ran==1])

cov1

vmi.1<-c(mi1.1,mi2.1)

vmi.1 #vetor de médias

mcov.1<-matrix(c(sigma1.1^2, cov1, cov1, sigma2.1^2),2,2)

#matrix de variâncias e covariâncias

mcov.1

require(MASS)

#Gerando dados da Normal bivariada

dat.1.covsim<-mvrnorm(n1,vmi.1,mcov.1)

dat.1.covsim

#Coeficiente de correlação entre P1 e P2 para Ran=2

cor2<-cor(P1[Ran==2],P2[Ran==2])

cor2

#Coeficiente de covariância

cov2<-cov(P1[Ran==2],P2[Ran==2])

cov2

vmi.2<-c(mi1.2,mi2.2)

vmi.2 #vetor de médias

mcov.2<-matrix(c(sigma1.2^2, cov2, cov2, sigma2.2^2),2,2)

#matrix de variâncias e covariâncias

mcov.2

require(MASS)

dat.2.covsim<-mvrnorm(n2,vmi.2,mcov.2)

dat.2.covsim

datsim3<-data.frame(rbind(dat.1.covsim,dat.2.covsim))

datsim3<-cbind(datsim3,Ransim)

datsim3

#Intervalos de Condiança e Testes t - Amostras pareadas

install.packages("Rmisc")

require(Rmisc)

CI(d[Ran==1],0.95)

CI(d[Ran==2],0.95)

#t.test(x, y = NULL,

# alternative = c("two.sided", "less", "greater"),

# mu = 0, paired = FALSE, var.equal = FALSE,

# conf.level = 0.95, ...)

t.test(d[Ran==1],conf.level=0.95, alternative = "two.sided") #teste bicaudal

t.test(d[Ran==1],conf.level=0.95, alternative = "greater") #teste unicaudal

t.test(d[Ran==2],conf.level=0.95, alternative = "two.sided") #teste bicaudal

t.test(d[Ran==2],conf.level=0.95, alternative = "greater") #teste unicaudal

#Intervalos de Condiança e Testes t - Amostras pareadas

t.test(P2 ~ factor(Ran),conf.level=0.95, alternative = "two.sided") #teste bicaudal

#Note que o número de graus de liberdade (df) tem correção para heterocedasticidade

#Verificar se a suposição de homocedasticidade (variâncias homogêneas) é válida

bartlett.test(P2~factor(Ran))

#H0:Variâncias homogêneas

#o valor-p (nível descritivo do teste) indica a rejeição de H0

bartlett.test(P1~factor(Ran))

t.test(P1~factor(Ran),conf.level=0.95, var.equal=TRUE)

ttP1<-t.test(P1~factor(Ran),conf.level=0.95, var.equal=TRUE)

names(ttP1)

ttP1$estimate

ttP1$statistic

sc2<-((35-1)\*(sd(P1[Ran==1]))^2+(57-1)\*(sd(P1[Ran==2])^2))/(35+57-2)

sc2

sqrt(sc2)

###Modelos mais Gerais

#Ajuste do modelo M1

#y=b0+b1P1c+e

P1c<-P1-mean(P1)

plot(P2~P1c)

m1 <- lm(P2~P1c,dados)

summary(m1)

names(m1)

anova(m1)

m1$coefficients

plot(m1$fitted.values,m1$residuals, main="Resíduo x Ajustado")

abline(h=0)

lines(lowess(m1$fitted.values,m1$residuals),col="red")

plot(P2~P1c)

abline(m1)

#Análise de diagnóstico das premissas do modelo

qqnorm(m1$residuals,ylab="Residuos", main=NULL)

qqline(m1$residuals)

title("Gráfico Normal de Probabilidade dos Resíduos")

# Teste de normalidade de Shapiro-Wilk

# H0: distribuição é normal

# p-valor > 0.05 não rejeita H0

shapiro.test(m1$residuals)

#Ajuste do modelo M2

#y=b0+b2Ran+e

plot(P2~Ran)

points(c(1,2),c(mean(P2[Ran==1]),mean(P2[Ran==2])),col="red",pch=19)

plot(P2~factor(Ran))

points(c(1,2),c(mean(P2[Ran==1]),mean(P2[Ran==2])),col="red",pch=19)

m2<-lm(P2~factor(Ran))

summary(m2)

anova(m2)

m2$coefficients #igual às médias dos grupos

mean(P2[Ran=="0"]) #intercept

mean(P2[Ran=="1"]) #intercept + Ran

plot(m2$fitted.values,m2$residuals)

abline(h=0)

lines(lowess(m2$fitted.values,m2$residuals),col="red")

plot(P2~Ran)

abline(m2)

#Ajuste do modelo M3

#y=b0+b1P1c+b2Ran+e

Ran

library(car)

Ran<-recode(Ran,"1='1';2='0'")

Ran

plot(P2~P1c, pch=23, bg=c('red', 'blue')[factor(Ran)])

m3<-lm(P2~P1c+Ran)

summary(m3)

anova(m3)

m3$coefficients

plot(m3$fitted.values,m3$residuals)

abline(h=0)

lines(lowess(m3$fitted.values,m3$residuals),col="red")

b0<-m3$coefficients[1]

b0

names(b0)<-NULL

b0

b1<-m3$coefficients[2]

names(b1)<-NULL

b2<-m3$coefficients[3]

names(b2)<-NULL

plot(P2~P1c, pch=23, bg=c('red', 'blue')[factor(Ran)])

abline(b0,b1, col="red")

abline((b0+b2),b1,col="blue")

#Ajuste do modelo M4

#y=b0+b1P1c+b2Ran+b3(P1c\*Ran)+e

P1cRan<-P1c\*Ran

plot(P2~P1c, pch=23, bg=c('red', 'blue')[factor(Ran)])

m4<-lm(P2~P1c+Ran+P1cRan)

summary(m4)

anova(m4)

m4$coefficients

plot(m4$fitted.values,m4$residuals)

abline(h=0)

lines(lowess(m4$fitted.values,m4$residuals),col="red")

b0<-m4$coefficients[1]

b0

names(b0)<-NULL

b0

b1<-m4$coefficients[2]

names(b1)<-NULL

b2<-m4$coefficients[3]

names(b2)<-NULL

b3<-m4$coefficients[4]

names(b3)<-NULL

plot(P2~P1c, pch=23, bg=c('red', 'blue')[factor(Ran)])

abline(b0,b1, col="red")

abline((b0+b2),(b1+b3),col="blue")