Universidade Federal Fluminense (UFF)

Instituto de Matemática e Estatística (IME)

Departamento de Estatística (GET)

Projeto de Monitoria 2015

Comandos utilizados para a elaboração do gabarito dos exercícios de análise de regressão linear aplicados: uma abordagem usando o programa R

Monitora: Rosana Gayer Carvalho

Professor: Dr. José Rodrigo de Moraes

Universidade Federal Fluminense (UFF)

Instituto de Matemática e Estatística (IME)

Departamento de Estatística (GET)

Disciplina: Modelos Lineares I

Professor: José Rodrigo de Moraes

Monitora: Rosana Gayer Carvalho

Comandos utilizados para resolução dos Exercícios de monitoria de Modelos Lineares I

Assuntos: Modelos de regressão linear simples e múltipla

Modelo de regressão logística binária

```
#Exercício 1:
install.packages("foreign")

require(foreign)

base=read.spss("abastecimento.sav",to.data.frame=T)

#c)

modelo=lm(base$Taxaabast~base$Reg_Norte+base$Reg_Nordeste+base$Reg_Sudeste
+base$Reg_Sul) ; summary(modelo)


#Exercício 2:
base=read.spss("idh.sav",to.data.frame=T)

#a)

cor(base$IDH,base$DENSIDADE)

plot(base$DENSIDADE,base$IDH,xlab="Densidade demográfica
(hab/km²)",ylab="IDH",pch=19)

#b)

modelo=lm(base$IDH~base$DENSIDADE) ; summary(modelo)

#d)
res_student=rstandard(modelo)
IDH_est=fitted.values(modelo)
plot(IDH_est, res_student,xlab="IDH estimado",ylab="Resíduos
estudentizados",pch=19)
abline(h=0)
```

```
qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

#Obs: Para confirmar

shapiro.test(res_student)

#e)

plot(base$ln_DENSIDADE,base$IDH,xlab="Densidade demográfica (em escala logarítmica)",ylab="IDH",pch=19)

#f)

ln_DENSIDADE=log(base$DENSIDADE)

cor(base$IDH,ln_DENSIDADE)

modelo=lm(base$IDH~ln_DENSIDADE)

summary(modelo)

res_student=rstandard(modelo)

res_student

IDH_est=fitted.values(modelo)

IDH_est

plot(IDH_est, res_student,xlab="IDH estimado",ylab="Resíduos estudentizados",pch=19)

abline(h=0)

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

#Obs: Para confirmar

shapiro.test(res_student)


#Exercício 3:

base=read.spss("producao_banana.sav",to.data.frame=T)

base
```



```
#a)

plot(base$Area,base$Prod_banana,xlab="Área destinada à colheita (em hectares)",ylab="Produção de banana (em R$1.000)",pch=19)

cor(base$Area,base$Prod_banana)

#b)

modelo=lm(base$Prod_banana~base$Area)

summary(modelo)

res_student=rstandard(modelo)

res_student

Prod_est=fitted.values(modelo)

Prod_est

plot(Prod_est, res_student,xlab="Produção estimada de banana (em R$1.000)", ylab=" Resíduos estudentizados",pch=19)

abline(h=0)

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

#Obs: Para confirmar

shapiro.test(res_student)

#c)

#Residuos brutos

res_b=residuals(modelo)

res_b

#Residuos estudentizados

res_estudent=rstandard(modelo)

res_estudent

#Teste de White

#REGRESSÃ AUXILIAR

#Para testar a homocedasticidade

res_b_quad=res_b*res_b

res_b_quad
```

```
Area_quad=(base$Area)^2

Area_quad

modelo_res_b_quad=lm(res_b_quad~base$Area+Area_quad)

modelo_res_b_quad

summary(modelo_res_b_quad)

#d)

#Transformando os dados:

#1 MODO: Para tentar tirar a heterocedasticidade

Prod_transf=base$Prod_banana/base$Area

Area_transf=1/base$Area

modelo=lm(Prod_transf~Area_transf)

summary(modelo)

res_student=rstandard(modelo)

res_student

Prod_transf_est=fitted.values(modelo)

Prod_transf_est

plot(Prod_transf_est, res_student,xlab="Produção estimada de banana (em R$1.000) do modelo transformado", ylab=" Resíduos estudentizados",pch=19)

abline(h=0)

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)


#Exercício 4:

base=read.spss("Exercício4_Nfornecer.sav",to.data.frame=T)

base

str(base)

Estabelecimento=as.numeric(as.character(base$Estab))

Desp=as.numeric(as.character(base$Despesa))

base=data.frame(Estabelecimento, Desp)
```



```
str(base)

#a)

plot(base$Estabelecimento,base$Desp,xlab="Número de estabelecimentos",ylab="Despesa total (em R$1.000)",pch=19)

#b)

modelo=lm(base$Desp~base$Estabelecimento)

summary(modelo)

res_student=rstandard(modelo)

res_student

Desp_est=fitted.values(modelo)

Desp_est

plot(Desp_est, res_student,xlab="Despesas totais estimadas  (em R$1.000)", ylab=" Resíduos estudentizados",pch=19)

abline(h=0)

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

#Obs: Para confirmar

shapiro.test(res_student)

#c)

plot(ln_Estabelecimento,ln_Desp,xlab="Logarítmo do Número de estabelecimentos",ylab="Logarítmo da Despesa total (em R$1.000)",pch=19)

#d)

ln_Estabelecimento=log10(base$Estabelecimento)

ln_Desp=log10(base$Desp)

modelo=lm(ln_Desp~ln_Estabelecimento)

summary(modelo)

res_student=rstandard(modelo)

res_student

Desp_ln_est=fitted.values(modelo)

Desp_ln_est
```

```
plot(Desp_ln_est, res_student,xlab=" Logarítmo das despesas totais estimadas  (em R$1.000)", ylab=" Resíduos estudentizados",pch=19)

abline(h=0)

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

#Obs: Para confirmar

shapiro.test(res_student)

#e)

10^(1.54526)


#Exercício 5:

base=read.spss("Exercicio5_Nfornecer.sav",to.data.frame=T)

base

#a)

plot(base$Pes,base$Q_vendida,xlab="Número de pés colhidos (em 1.000 pés)",ylab="Quantidade vendida (em 1.000 frutos)",pch=19)

plot(base$Area,base$Q_vendida,xlab="Área colhida (em hectares)",ylab="Quantidade vendida (em 1.000 frutos)",pch=19)

#b)

modelo=lm(base$Q_vendida~base$Pes+base$Area)

summary(modelo)

#Método gráfico

pairs(base,pch=19)

cor(base)

#Método estatístico

install.packages("car")

require(car)

vif(modelo)

#Logo não existe problema de multicolinearidade
```



```
res_student=rstandard(modelo)

res_student

Q_vendida_est=fitted.values(modelo)

Q_vendida_est

plot(Q_vendida_est, res_student,xlab="Estimativa da quantidade vendida de coco-da-baía (em 1.000 frutos)",ylab="Resíduos estudentizados",pch=19)

abline(h=0)

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

#Obs: Para confirmar

shapiro.test(res_student)

#Modelo selecionado

modelo=lm(base$Q_vendida~base$Area)

summary(modelo)

res_student=rstandard(modelo)

res_student

Q_vendida_est=fitted.values(modelo)

Q_vendida_est

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

shapiro.test(res_student)

#c)

ln_Q_vendida=log10(base$Q_vendida)

ln_Area=log10(base$Area)

modelo=lm(ln_Q_vendida~ln_Area)

summary(modelo)

#d)

res_student=rstandard(modelo)

res_student
```

```
Q_vendida_est=fitted.values(modelo)

Q_vendida_est

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

shapiro.test(res_student)


#Exercício 6:

base=read.spss("Exericio6_Nfornecer.sav",to.data.frame=T)

base

str(base)

#a)

Q_produzida=(base$Q_produzida)*1000

rend=Q_produzida/base$Area

rend

Valor=base$Valor

Valor

base=data.frame(rend,Valor)

base

#b)

plot(base$rend,base$Valor,pch=19,xlab="Rendimento médio de milho (Kg/ha)",ylab="Valor da produção de milho (em R$1.000)")

#c)

modelo=lm(base$Valor~base$rend)

summary(modelo)

res_student=rstandard(modelo)

res_student

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)
```



```
#Para confirmar!

shapiro.test(res_student)

#d)

ln_Valor=log(base$Valor)

ln_Valor

modelo=lm(ln_Valor~base$rend)

summary(modelo)

#e)

plot(base$rend,ln_Valor,pch=19,xlab="Rendimento médio de milho (Kg/ha)",ylab="Logarítmo Valor da produção de milho (em R$1.000)")

res_student=rstandard(modelo)

res_student

Valor_est=fitted.values(modelo)

Valor_est

plot(Valor_est, res_student,xlab="Valor estimado da produção de milho (em R$ 1.000)",ylab="Resíduos estudentizados",pch=19)

abline(h=0)

qqnorm(res_student,xlab="Quantis da normal padrão",ylab="Quantis dos resíduos estudentizados",pch=19)

qqline(res_student)

shapiro.test(res_student)


#Exercício 7:

base=read.spss("Exercicio7_Nfornecer_1.sav",to.data.frame=T)

base

str(base)

Valor=as.numeric(as.character(base$Valor))

Valor

Rend=base$Rendimento

Rend

base=data.frame(Rend,Valor)
```

```
base

str(base)

#a)

plot(base$Rend,base$Valor,pch=19,xlab="Rendimento médio de feijão (Kg/ha)",ylab="Valor da produção de feijão (em R$1.000)")

#b)

modelo=lm(base$Valor~base$Rend)

summary(modelo)


#Exercício 8:

base=read.spss("deslizamento.sav")

base

head(base)

str(base)

#constras= mudando a categoria de referÃªncia

mod1=glm(Deslizamento~Regioes+Tamanho,binomial,base,contras=list(Regioes="contr.SAS",Tamanho="contr.SAS"))

summary(mod1)

probest=fitted.values(mod1)

summary(probest)

lnchance_deslizamento=log(probest/(1-probest))

lnchance_deslizamento

#Avaliacao da capacidade preditiva do modelo

Ychapeu=ifelse(probest>=0.5,1,0)

Ychapeu

#1- deslizamento

#0- sem deslizamento

tabela=table(base$Deslizamento,Ychapeu)

tabela
```



```
prop.table(tabela)

TG= 0.81294828+0.02642507

TG

prop.table(tabela,1)

#S=P(Ychpaeu=1/Y=1)=0.15642458

#E=P(Ychpaeu=0/Y=0)=0.97819668


#Exercício9:

base=read.spss("inundacoes.sav")

base

head(base)

str(base)

#constras= mudando a categoria de referÃªncia

mod2=glm(Inundacao~Regioes+Tamanho,binomial,base,contras=list(Regioes="contr.SAS",Tamanho="contr.SAS"))

summary(mod2)

probest=fitted.values(mod2)

summary(probest)

lnchance_inundacao=log(probest/(1-probest))

lnchance_inundacao

#Avaliacao da capacidade preditiva do modelo

Ychapeu=ifelse(probest>=0.5,1,0)

Ychapeu

#1- inundacao

#0- sem inundacao

tabela=table(base$Inundacao,Ychapeu)

tabela

prop.table(tabela)

TG=0.69093255+0.05068190

TG
```

```
prop.table(tabela,1)

#S=P(Ychpaeu=1/Y=1)=0.17471410

#E=P(Ychpaeu=0/Y=0)=0.97326064


#Grafico INSTALAR!

install.packages("lattice")

require(lattice)

xyplot(probest~base$Regioes,group=base$Tamanho,type="l",xlab="Grandes regiões brasileiras",ylab="Probabilidade estimada de enchente ou inundação", auto.key = list(space = "right"))


#Questão 10

base=read.spss("conhecimento.sav")

base

head(base)

str(base)

#constras= mudando a categoria de referÃªncia

mod2=glm(Conhecimento~Sexo+Faixa_etaria+Escolaridade+Tamanho,binomial,base,contras=list(Escolaridade="contr.SAS",Faixa_etaria="contr.SAS"))

summary(mod2)

probest=fitted.values(mod2)

summary(probest)

lnchance_conhecimento=log(probest/(1-probest))

lnchance_conhecimento

#Avaliacao da capacidade preditiva do modelo

Ychapeu=ifelse(probest>=0.5,1,0)

Ychapeu

#1- conhecimento

#0- sem conhecimento

#Como tem “NA” :
```



```
a=base$Conhecimento[which(base$Conhecimento=="Sim"|base$Conhecimento=="Não")]

a

length(a)

length(Ychapeu)

tabela=table(a,Ychapeu)

tabela

prop.table(tabela)

TG=0.58241758+0.12087912

TG

prop.table(tabela,1)

#S=P(Ychpaeu=1/Y=1)=0,92982456

#E=P(Ychpaeu=0/Y=0)=0,32352941
```