

Rotinas computacionais implementadas no R utilizadas no trabalho

As rotinas computacionais dos métodos descritos neste trabalho foram implementadas no software livre R (R Development Core Team, 2012) e estão descritas a seguir.

1. Simulação dos Dados:

```
#Função simula conjunto de dados para treinamento

####Argumentos da função
#n_ambiente = número de ambientes avaliados no estudo
#desv_pad =desvio padrão obtido por meio de
coeficiente de variação e da média do experimento em
estudo
#beta0 = média do experimento em estudo
#n_training = número de genótipos simulados
#gl = graus de liberdade da análise conjunta
#QMR = quadrado médio do resíduo da análise conjunta
#rep = número de repetições
#I_ord = índice ambiental ordenado
```

1.1. Dados assimétricos à direita

```
###Simulação dos valores fenotípicos

simula<-function(n_ambientes,QMR,nrep,beta0,beta1,n_simul,
I_ord)
{
y_obs<-matrix(0,nrow=n_simul,ncol=n_ambientes,byrow = T)
for (i in 1:n_simul)
{
erro<-rexp(20,0.006)
y_obs[i,]<-beta0+beta1*I_ord+erro
dados<<-y_obs
}
}
#####

####PACOTES UTILIZADOS: quantreg e mblm

###Defina os argumento da função de acordo com o
experimento avaliado
#número de indivíduos para o treinamento de cada classe
n_simul=100
#QMR
```

```

QMR=55851
#Número de repetições
nrep=2
#Média geral
beta0=1176
#beta1
beta1=1
#número de ambientes
n_ambientes=20
#Índice ambiental ordenado
I_ord<-c(-695,-681,-541,-508,-472,-351,-283,-237,-164,-
163,-29,33,45,89,112,154,203,864,1071,1553)
gl=1729
simula(n_ambientes,QMR,nrep,beta0,beta1,n_simul,I_ord)

nrepp=100
p_acertos<-matrix(NA,nrepp,5)
EQMbeta1r<-matrix(NA,nrepp,5)
EQMbeta0r<-matrix(NA,nrepp,5)

for(h in 1:nrepp)
{

n_simul=100
#QMR
QMR=55851
#Número de repetições
nrep=2
#Média geral
beta0=1176
#beta1
beta1=1
#número de ambientes
n_ambientes=20
#Índice ambiental ordenado
I_ord<-c(-695,-681,-541,-508,-472,-351,-283,-237,-164,-
163,-29,33,45,89,112,154,203,864,1071,1553)
gl=1729
simula(n_ambientes,QMR,nrep,beta0,beta1,n_simul,I_ord)

#####regressão MQO #Eberhart e Russell (1966)

difbeta1<-matrix(NA,n_simul,1)
difbeta0<-matrix(NA,n_simul,1)
teste<-matrix(NA,n_simul,1)
for(i in 1:n_simul)
{
    reg<-lm(dados[i,]~I_ord)
    b=reg$coefficients
    beta1<-b[2]
    difbeta1[i,1]<-(beta1-1)
}

```

```

beta0<-b[1]
difbeta0[i,1]<-(beta0-1176)
t<-(beta1-1)/sqrt((QMR/2)/sum(I_ord^2))
p_valor<-2*(pt(abs(t), gl, lower.tail=F))
if(p_valor>0.05)
{teste[i,1]<-1}
else {teste[i,1]<-0}

}
EQMbeta1<-sum(difbeta1^2)/n_simul
EQMbeta0<-sum(difbeta0^2)/n_simul
#EQMbeta0
p_acertos1<-(sum(teste)/n_simul)*100
#p_acertos1

p_acertos[h,1]<-p_acertos1
EQMbeta1r[h,1]<-EQMbeta1
EQMbeta0r[h,1]<-EQMbeta0

#####REGRESSÃO QUANTILICA TAU=0.25

difbeta11<-matrix(NA,n_simul,1)
difbeta00<-matrix(NA,n_simul,1)
teste1<-matrix(NA,n_simul,1)
for(i in 1:n_simul)
{
  reg1<-rq(dados[i,]~I_ord,tau=0.25)
  b1=reg1$coefficients
  beta1<-b1[2]
  difbeta11[i,1]<-(beta1-1)
  beta0<-b1[1]
  difbeta00[i,1]<-(beta0-1176)
  t<-(beta1-1)/sqrt((QMR/2)/sum(I_ord^2))
  p_valor<-2*(pt(abs(t), gl, lower.tail=F))
  if(p_valor>0.05)
  {teste1[i,1]<-1}
  else {teste1[i,1]<-0}
}

EQMbeta11<-sum(difbeta11^2)/n_simul
#EQMbeta11
EQMbeta00<-sum(difbeta00^2)/n_simul
#EQMbeta00
p_acertos11<-(sum(teste1)/n_simul)*100
#p_acertos11

EQMbeta1r[h,2]<-EQMbeta11
EQMbeta0r[h,2]<-EQMbeta00
p_acertos[h,2]<-p_acertos11

```

```
#####REGRESSÃO QUANTILICA TAU=0.5
```

```
difbeta111<-matrix(NA,n_simul,1)
difbeta000<-matrix(NA,n_simul,1)
teste1<-matrix(NA,n_simul,1)
for(i in 1:n_simul)
{
  reg1<-rq(dados[i,]~I_ord,tau=0.5)
  b1=reg1$coefficients
  beta1<-b1[2]
  difbeta111[i,1]<-(beta1-1)
  beta0<-b1[1]
  difbeta000[i,1]<-(beta0-1176)
  t<-(beta1-1)/sqrt((QMR/2)/sum(I_ord^2))
  p_valor<-2*(pt(abs(t),gl,lower.tail=F))
  if(p_valor>0.05)
  {teste1[i,1]<-1}
  else {teste1[i,1]<-0}
}
```

```
EQMbeta111<-sum(difbeta111^2)/n_simul
#EQMbeta111
EQMbeta000<-sum(difbeta000^2)/n_simul
#EQMbeta000
p_acertos111<-(sum(teste1)/n_simul)*100
#p_acertos111
```

```
EQMbeta1r[h,3]<-EQMbeta111
EQMbeta0r[h,3]<-EQMbeta000
p_acertos[h,3]<-p_acertos111
```

```
#####REGRESSÃO QUANTILICA TAU = 0.75
```

```
difbeta1111<-matrix(NA,n_simul,1)
difbeta0000<-matrix(NA,n_simul,1)
teste1<-matrix(NA,n_simul,1)
for(i in 1:n_simul)
{
  reg1<-rq(dados[i,]~I_ord,tau=0.75)
  b1=reg1$coefficients
  beta1<-b1[2]
  difbeta1111[i,1]<-(beta1-1)
  beta0<-b1[1]
  difbeta0000[i,1]<-(beta0-1176)
  t<-(beta1-1)/sqrt((QMR/2)/sum(I_ord^2))
  p_valor<-2*(pt(abs(t),gl,lower.tail=F))
  if(p_valor>0.05)
  {teste1[i,1]<-1}
```

```

    else {teste1[i,1]<-0}
}

EQMbeta1111<-sum(difbeta1111^2)/n_simul
#EQMbeta1111
EQMbeta0000<-sum(difbeta0000^2)/n_simul
#EQMbeta0000
p_acertos1111<-(sum(teste1)/n_simul)*100
#p_acertos1111

```

```

EQMbeta1r[h,4]<-EQMbeta1111
EQMbeta0r[h,4]<-EQMbeta0000
p_acertos[h,4]<-p_acertos1111

```

#####REGRESSÃO NÃO PARAMETRICA

```

difbeta11111<-matrix(NA,n_simul,1)
difbeta00000<-matrix(NA,n_simul,1)
teste<-matrix(NA,n_simul,1)
for(i in 1:n_simul)
{

y<-dados[i,]
x<-I_ord

    reg<-mblm(y~x)
    b=reg$coefficients
    beta1<-b[2]
    difbeta11111[i,1]<-(beta1-1)
    beta0<-b[1]
    difbeta00000[i,1]<-(beta0-1176)
    t<-(beta1-1)/sqrt((QMR/2)/sum(I_ord^2))
    p_valor<-2*(pt(abs(t), gl, lower.tail=F))
    if(p_valor>0.05)
    {teste[i,1]<-1}
    else {teste[i,1]<-0}

}
EQMbeta11111<-sum(difbeta11111^2)/n_simul
EQMbeta00000<-sum(difbeta00000^2)/n_simul
p_acertos11111<-(sum(teste)/n_simul)*100

```

```

p_acertos[h,5]<-p_acertos11111
EQMbeta1r[h,5]<-EQMbeta11111
EQMbeta0r[h,5]<-EQMbeta00000

```

```

}
colMeans(p_acertos)
colMeans(EQMbeta0r)
colMeans(EQMbeta1r)

```

1.2. Dados assimétricos à esquerda

```

###Simulação dos valores fenotípicos

simula<-
function(n_ambientes,QMR,nrep,beta0,beta1,n_simul,
I_ord)
{
y_obs<-matrix(0,nrow=n_simul,ncol=n_ambientes,byrow =
T)
for (i in 1:n_simul)
{
erro<<-(-rexp(20,sqrt(nrep/QMR)))
y_obs1<-beta0+beta1*I_ord+erro
y_obs[i,]<-y_obs1+(min(y_obs1)*-1)
dados<<-y_obs
}
}
#####

```

A partir daqui é idêntico à simulação à direita.

1.3. Dados simétricos

```

###Simula valores fenotípicos

simula<-function(n_ambientes,QMR,nrep,beta0,beta1,n_simul,
I_ord)
{
y_obs<-matrix(0,nrow=n_simul,ncol=n_ambientes,byrow = T)
for (i in 1:n_simul)
{
erro<<-rnorm(20,beta0,sqrt(QMR/2))
y_obs[i,]<-beta0+beta1*I_ord+erro
dados<<-y_obs
}
}
#####

```

A partir daqui é idêntico à simulação à direita.

1.4. Dados assimétricos à direita com presença de *outlier*

```
simula<-function(n_ambientes,beta0,beta1,I_ord)
{
  erro<-rexp(20,0.006)
  y_obs<-beta0+beta1*I_ord+erro
}

###Defina os argumentos da função de acordo com o
experimento avaliado
#Média geral
beta0=1176
#beta1
beta1=1
#número de ambientes
n_ambientes=20
#Índice ambiental ordenado
I_ord<-c(-695,-681,-541,-508,-472,-351,-283,-237,-164,-
163,-29,33,45,89,112,154,203,864,1071,1553)

n_genotipos=100

##com outliers
  cont<-0
  dados<-matrix(0,n_genotipos,n_ambientes,byrow=T)
  i<-1
  repeat
  {
    z1<-simula(n_ambientes,beta0,beta1,I_ord)
    reg<-lm(z1~I_ord)
    h=dfbeta(reg)
    if(sum(h[,2]>=abs(0.44))>0)
    {
      dados[i,]<-z1
      cont<-cont+1
      i<-i+1
    }
    if(cont==n_genotipos)
      break
  }

#####
```

A partir daqui é idêntico à simulação à direita.

1.5. Dados assimétricos à esquerda com presença de outlier

```
simula<-function(n_ambientes,beta0,beta1,I_ord)
{
  erro<-(-rexp(20,0.006))
  y_obs1<-beta0+beta1*I_ord+(erro)
  y_obs<-y_obs1+(min(y_obs1)*-1)
  dados<-y_obs
}

###Defina os argumentos da função de acordo com o
experimento avaliado
#QMR
QMR=55851
#Número de repetições
nrep=2
#Média geral
beta0=1176
#beta1
beta1=-1
#número de ambientes
n_ambientes=20
#Índice ambiental ordenado
I_ord<-c(-695,-681,-541,-508,-472,-351,-283,-237,-164,-
163,-29,33,45,89,112,154,203,864,1071,1553)

n_genotipos=100

##com outliers
cont<-0
dados<-matrix(0,n_genotipos,n_ambientes,byrow=T)
i<-1
repeat
{
  z1<-simula(n_ambientes,beta0,beta1,I_ord)
  z1<-(-z1+(max(z1)))
  reg<-lm(z1~I_ord)
  h=dfbeta(reg)
  if(sum(h[,2]>=abs(0.44))>0)
  {
    dados[i,]<-z1
    cont<-cont+1
    i<-i+1
  }
  if(cont==n_genotipos)
  break
}
#####
```


A partir daqui é idêntico à simulação à direita.

1.6. Dados simétricos com presença de *outlier*

```
simula<-function(n_ambientes,beta0,beta1,I_ord)
{
  erro<-rnorm(20,beta0,sqrt(QMR/2))
  y_obs<-beta0+beta1*I_ord+erro
}

###Defina os argumentos da função de acordo com o
experimento avaliado
#QMR
QMR=55851
#Número de repetições
nrep=2
#Média geral
beta0=1176
#beta1
beta1=1
#número de ambientes
n_ambientes=20
#Índice ambiental ordenado
I_ord<-c(-695,-681,-541,-508,-472,-351,-283,-237,-164,-
163,-29,33,45,89,112,154,203,864,1071,1553)

n_genotipos=100

##com outliers
cont<-0
dados<-matrix(0,n_genotipos,n_ambientes,byrow=T)
i<-1
repeat
{
  z1<-simula(n_ambientes,beta0,beta1,I_ord)
  z1[20]<-4500
  reg<-lm(z1~I_ord)
  h=dfbeta(reg)
  if(sum(h[,2]>=abs(0.44))>0)
  {
    dados[i,]<-z1
    cont<-cont+1
    i<-i+1
  }
  if(cont==n_genotipos)
  break
}
#####
```

A partir daqui é idêntico à simulação à direita.

2. Dados Reais

2.1. Adaptabilidade

```
#####MUDAR PARA O DIRETÓRIO ONDE SE ENCONTRA O
CONJUNTO DE DADOS#####
```

```
####PACOTES UTILIZADOS: quantreg e mblm
```

```
dados=scan("dados_reais.txt")
dados1<-matrix(dados,92,21,byrow=T)
dados1[1,]
I_ord<-c(-695,-681,-541,-508,-472,-351,-283,-237,-164,-
163,-29,33,45,89,112,154,203,864,1071,1553)
I_ord
QMR=55851
nrep=2
beta0=1176
gl=1729
```

```
#####Eberhart and Russell(1966):
teste<-matrix(NA,92,1)
beta1<-matrix(NA,92,1)
beta0<-matrix(NA,92,1)
```

```
for(i in 1:92)
{
  reg<-lm(dados1[i,-21]~I_ord)
  b=reg$coefficients
  beta1[i,1]<-b[2]
  beta0[i,1]<-b[1]
  t<-(beta1-1)/sqrt((QMR/2)/sum(I_ord^2))
  p_valor<-2*(pt(abs(t), gl,lower.tail=F))
  teste<-(p_valor>0.05)
}
```

```
write.table(beta1,"beta1.txt",row.names=FALSE,quote=FALSE)
```

```
#####REGRESSÃO NÃO PARAMÉTRICA
```

```
testeT<-matrix(NA,92,1)
```

```

beta1T<-matrix(NA,92,1)
beta0T<-matrix(NA,92,1)

for(i in 1:92)
{
y<-dados1[i,-21]
x<-I_ord
  reg<-mblm(y~x)
  b=reg$coefficients
  beta1T[i,1]<-b[2]
  beta0T[i,1]<-b[1]
  t<-(beta1T-1)/sqrt((QMR/2)/sum(I_ord^2))
  p_valor<-2*(pt(abs(t), gl, lower.tail=F))
  testeT<-(p_valor>0.05)

}

write.table(beta1T, "beta1T.txt", row.names=FALSE, quote=FALSE
)

#####REGRESSÃO QUANTILICA TAU=0.25

testeQ<-matrix(NA,92,1)
beta1Q<-matrix(NA,92,1)
beta0Q<-matrix(NA,92,1)

for(i in 1:92)
{
  reg1<-rq(dados1[i,-21]~I_ord,tau=0.25)
  b=reg1$coefficients
  beta1Q[i,1]<-b[2]
  beta0Q[i,1]<-b[1]
  t<-(beta1Q-1)/sqrt((QMR/2)/sum(I_ord^2))
  p_valor<-2*(pt(abs(t), gl, lower.tail=F))
  testeQ<-(p_valor>0.05)

}

write.table(beta1Q, "beta1Q0.25.txt", row.names=FALSE, quote=F
ALSE)

```

```
#####REGRESSÃO QUANTILICA TAU=0.50
```

```
teste0.5Q<-matrix(NA,92,1)  
beta10.5Q<-matrix(NA,92,1)  
beta00.5Q<-matrix(NA,92,1)
```

```
for(i in 1:92)  
{  
  reg1<-rq(dados1[i,-21]~I_ord,tau=0.50)  
  b1=reg1$coefficients  
  beta10.5Q[i,1]<-b1[2]  
  beta00.5Q[i,1]<-b1[1]  
  t<-(beta10.5Q-1)/sqrt((QMR/2)/sum(I_ord^2))  
  p_valor<-2*(pt(abs(t),gl,lower.tail=F))  
  teste0.5Q<-(p_valor>0.05)  
}
```

```
write.table(beta10.5Q,"beta1Q0.5.txt",row.names=FALSE,quote  
=FALSE)
```

```
#####REGRESSÃO QUANTILICA TAU=0.75
```

```
teste0.75Q<-matrix(NA,92,1)  
beta10.75Q<-matrix(NA,92,1)  
beta00.75Q<-matrix(NA,92,1)
```

```
for(i in 1:92)  
{  
  reg1<-rq(dados1[i,-21]~I_ord,tau=0.75)  
  b1=reg1$coefficients  
  beta10.75Q[i,1]<-b1[2]  
  beta00.75Q[i,1]<-b1[1]  
  t<-(beta10.75Q-1)/sqrt((QMR/2)/sum(I_ord^2))  
  p_valor<-2*(pt(abs(t),gl,lower.tail=F))  
  teste0.75Q<-(p_valor>0.05)  
}
```

```
write.table(beta1Q0.75,"beta1Q0.75.txt",row.names=FALSE,quo  
te=FALSE)
```

2.2. Estabilidade

```
#####MUDAR PARA O DIRETÓRIO ONDE SE ENCONTRA O CONJUNTO DE DADOS#####
```

```
####PACOTES UTILIZADOS: quantreg e mblm
```

```
dados=scan("dados_reais.txt")
dados1<-matrix(dados,92,21,byrow=T)
dados1[1,]
I_ord<-c(-695,-681,-541,-508,-472,-351,-283,-237,-164,-163,-29,33,45,89,112,154,203,864,1071,1553)
I_ord
QMR=55851
nrep=2
beta0=1176
gl=1729
```

```
#####REGRESSÃO NÃO PARAMÉTRICA
```

```
testeT<-matrix(NA,92,1)
beta1T<-matrix(NA,92,1)
beta0T<-matrix(NA,92,1)
var1=matrix(NA,92,1)
var2=matrix(NA,92,1)
R2=matrix(NA,92,1)
```

```
for(i in 1:92)
{
```

```
  y<-dados1[i,-21]
```

```
  x<-I_ord
```

```
    reg<-mblm(y~x)
    b=reg$coefficients
    beta1T[i,1]<-b[2]
    beta0T[i,1]<-b[1]
    t<-(beta1T-1)/sqrt((QMR/2)/sum(I_ord^2))
    p_valor<-2*(pt(abs(t), gl,lower.tail=F))
    testeT<-(p_valor>0.05)
    var1[i,1]<-var(dados1[i,-21])
```

```

var2[i,1]<-var(reg$fitted.values)
R2[i,1]<-var2[i,1]/var1[i,1]

}

#####REGRESSÃO QUANTILICA TAU = 0.25

testeQ<-matrix(NA,92,1)
beta1Q<-matrix(NA,92,1)
beta0Q<-matrix(NA,92,1)
var1=matrix(NA, 92,1)
var2=matrix(NA, 92,1)
R2=matrix(NA, 92,1)

for(i in 1:92)
{
  reg1<-rq(dados1[i,-21]~I_ord,tau=0.25)
  b=reg1$coefficients
  beta1Q[i,1]<-b[2]
  beta0Q[i,1]<-b[1]
  t<-(beta1Q-1)/sqrt((QMR/2)/sum(I_ord^2))
  p_valor<-2*(pt(abs(t),gl,lower.tail=F))
  testeQ<-(p_valor>0.05)
  var1[i,1]<-var(dados1[i,-21])
  var2[i,1]<-var(reg1$fitted.values)
  R2[i,1]<-var2[i,1]/var1[i,1]
}

#####REGRESSÃO QUANTILICA TAU=0.5

testeQ<-matrix(NA,92,1)
beta1Q<-matrix(NA,92,1)
beta0Q<-matrix(NA,92,1)
var1=matrix(NA, 92,1)
var2=matrix(NA, 92,1)
R2=matrix(NA, 92,1)

for(i in 1:92)
{
  reg1<-rq(dados1[i,-21]~I_ord,tau=0.50)
  b=reg1$coefficients
  beta1Q[i,1]<-b[2]
  beta0Q[i,1]<-b[1]
}

```

```

t<-(beta1Q-1)/sqrt((QMR/2)/sum(I_ord^2))
p_valor<-2*(pt(abs(t),g1,lower.tail=F))
testeQ<-(p_valor>0.05)
var1[i,1]<-var(dados1[i,-21])
var2[i,1]<-var(reg1$fitted.values)
R2[i,1]<-var2[i,1]/var1[i,1]
}

```

3. Teste de D'Agostino (1970)

```
#####PACOTE UTILIZADO: moments
```

```

dados=scan("dados_reais.txt")
dados1<-matrix(dados,92,21,byrow=T)
dim(dados1)

```

```

teste<-matrix(NA,92,1)
p_valor<-matrix(NA,92,1)
skewnes<-matrix(NA,92,1)

```

```

for(i in 1:92)
{

```

```

x= dados1[i,-21]

```

```

skewnes[i,1]<-skewness(x)
teste_agostino<-agostino.test(x)
teste_agostino
p<-teste_agostino$p.value
p_valor[i,1]<-p
teste<-(p_valor>0.05)

```

```

}

```

```

teste

```