

# ME720 - Modelos Lineares Generalizados

Profa. Larissa Avila Matos

Resolução Atividade

## Questão 1

```
ajustePoisson <-function(y,X,init,eps=1e-6,maxiter=50){

  beta <- init
  n<-dim(X)[1]
  p=length(t(beta))
  out <- matrix(NA, nrow=maxiter+1,ncol=p)
  out[1,] <- t(init)
  i <- 1

  continue <- T
  while (continue) {
    i <- i+1
    beta.o <- beta
    W<-diag(n)
    diag(W)<-exp(X%*%beta.o)
    mu<-exp(X%*%beta.o)
    beta <- beta.o + solve(t(X)%*%W%*%X)%*%t(X)%*%(y-mu)
    if (sum(is.na(beta))>0){stop("NA nas estimativas")}
    out[i,] <- t(beta)
    continue <- (abs(beta-beta.o) > eps) && (i <= maxiter)
  }
  if (i > maxiter) {
    warning("Máximo número de iterações atingido")
  }

  out <- out[!is.na(out[,1]),]
  est=out[i,]

  v.ajustado <- exp(X%*%est)
  preditor.linear=X%*%est
  desvio <- sum(2*(y*(ifelse(y==0,0,log(y))-log(v.ajustado))-y+v.ajustado))
  AIC <- -2*(sum(dpois(y,v.ajustado,log=T))) + 2*p
  BIC <- -2*(sum(dpois(y,v.ajustado,log=T))) + 2*p*log(n)

  H<-diag(sqrtm(W)%*%X%*%solve(t(X)%*%W%*%X)%*%t(X)%*%solve(sqrtm(W)))
  res <- sqrt(2*(y*(ifelse(y==0,0,log(y))-log(v.ajustado))-y+v.ajustado))/sqrt(1-H)
  residuos <- ifelse(y-v.ajustado>0,res,-res)
  residuos.pearson <- (y-v.ajustado)/sqrt(v.ajustado*(1-H))

  W<-diag(length(y))
  diag(W)<-exp(X%*%est)
  I=t(X)%*%W%*%X
  se=sqrt(diag(solve(I)))
  z.value=est/se
  p.value=pnorm(abs(z.value), lower.tail = F)

  z.alpha<-qnorm(0.975)
  LI=est-z.alpha*se
  LS=est+z.alpha*se

  saida<-list(out=out, est=est, iter=i, se=se, v.ajustado=v.ajustado, preditor.linear=preditor.linear,
    z.value=z.value, p.value=p.value, LI=LI, LS=LS,AIC=AIC, BIC=BIC, desvio=desvio, gl.res=(n-p),
    residuos=residuos, residuos.pearson=residuos.pearson)
  return(saida)
}
```

```

mlgPoisson = function(y,X,init,eps=1e-6,maxiter=50){

  # Validações

  ly <- length(y)
  if (nrow(X) != ly){stop("Número de linhas na matriz X difere do número de observações em y")}

  if (ncol(X) != nrow(init)){stop("Valores inválidos para os parâmetros iniciais")}

  if(maxiter <= 0 | maxiter%%1 != 0) stop("maxiter precisa ser um inteiro positivo.")

  if(eps <=0 | eps > 1) stop("0 erro precisa pertencer ao intervalo (0,1]")

  # Outputs

  out = ajustePoisson(y,X,init,eps=1e-6,maxiter=50)

  betas   = round(out$est, 4)
  se       = round(out$se, 4)
  z.value  = round(out$z.value, 4)
  p.value  = out$p.value
  LI       = round(out$LI, 2)
  LS       = round(out$LS, 2)

  # Criterios

  desvio = round(out$desvio,3)
  desvioinf = matrix(c(desvio,out$gl.res),1,2)
  AIC      = round(out$AIC,3)
  BIC      = round(out$BIC,3)
  iter     = out$iter

  Estimativas      = cbind(betas,se,z.value,p.value)
  colx              = ncol(as.matrix(X))

  namesx = paste0('x',1)
  IC=paste0('[' ,LI[1] ,';' ,LS[1] ,"]")
  if(ncol(as.matrix(X))>1)
  {
    for(i in 2:ncol(as.matrix(X))){namesx = cbind(namesx, paste0('x',i))
    IC = rbind(IC,paste0('[' ,LI[i] ,';' ,LS[i] ,"]"))}
  }

  Estimativas<-data.frame(Estimativas,IC=IC)
  dimnames(Estimativas) = list(c(namesx[1:colx]),c("Estimativas", "SE", "valor z", "p-valor", "IC's"))
  dimnames(desvioinf) = list(c('Desvio: '),c(" ", 'g.l'))
  names(iter) = c('Número de iterações: ')
  criteria = as.matrix(c(AIC,BIC))
  dimnames(criteria) = list(c("AIC:", "BIC:"),c(" "))

  cat('\n')
  cat('-----\n')
  cat('      MLG para o modelo Poisson com função de ligação log      \n')
  cat('-----\n')
  print(Estimativas)
  cat('-----\n')
  print(desvioinf)
  cat('-----\n')
  print(iter)
  cat('-----\n')
  cat('Critérios de seleção do Modelo \n')
  print(criteria)
  cat('-----\n')

  class(out) <- "mlgPoisson"
  return(invisible(out))

}

```

## Exemplo Exposição de Bactérias

```
bacterias<-c(175,108,95,82,71,50,49,31,28,17,16,11)
exposicao<-c(1,2,3,4,5,6,7,8,9,10,11,12)
```

Ajuste glm:

```
fit<-glm(bacterias~exposicao,family=poisson(link = "log"))
summary(fit)
```

Call:

```
glm(formula = bacterias ~ exposicao, family = poisson(link = "log"))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-1.7703	-0.5715	-0.1019	0.5496	1.2794

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	5.30557	0.06348	83.58	<2e-16 ***
exposicao	-0.22890	0.01270	-18.02	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 393.6292 on 11 degrees of freedom  
 Residual deviance: 8.4215 on 10 degrees of freedom  
 AIC: 80.182

Number of Fisher Scoring iterations: 4

Ajuste glmPoisson:

```
X<-cbind(rep(1,length(exposicao)),exposicao)
init<-matrix(solve(t(X)%*%X)%*%t(X)%*%sqrt(bacterias),2,1)
fit.mlg<-mlgPoisson(bacterias,X,init,eps=1e-6,maxiter=50)
```

-----  
 MLG para o modelo Poisson com função de ligação log  
 -----

	Estimativas	SE	valor z	p-valor	IC's
x1	5.3056	0.0635	83.5750	0.000000e+00	[5.18;5.43]
x2	-0.2289	0.0127	-18.0234	6.386627e-73	[-0.25;-0.2]

-----  
 g.l  
 Desvio: 8.422 10  
 -----

Número de iterações:  
 12  
 -----

Critérios de seleção do Modelo

AIC: 80.182  
 BIC: 86.122  
 -----

Resíduo desvio padronizado:

```
residuos<-cbind(residuals.glm(fit),fit.mlg$residuos)
colnames(residuos)<-c("glm","mlgPoisson")
residuos
```

	glm	mlgPoisson
1	1.14871848	1.15037616
2	-1.77034442	-1.77216391
3	-0.64055838	-0.64105455
4	0.15107587	0.15117908
5	0.84173046	0.84236463

```
6 -0.14314040 -0.14328145
7 1.27943323 1.28119813
8 -0.22657370 -0.22701238
9 0.45220360 0.45340480
10 -0.78002614 -0.78278823
11 -0.06067891 -0.06095752
12 -0.54845042 -0.55163662
```

```
# residuals.glm(object, type), com type = c("deviance", "pearson", "working", "response", "partial")
```

Resíduo de pearson padronizado:

```
residuos<-cbind(residuals.glm(fit,type = "pearson"),fit.mlg$residuos.pearson)
colnames(residuos)<-c("glm","mlgPoisson")
residuos
```

	glm	mlgPoisson
1	1.16596377	1.16764635
2	-1.72344485	-1.72521613
3	-0.63372990	-0.63422077
4	0.15149889	0.15160239
5	0.85634914	0.85699432
6	-0.14266151	-0.14280209
7	1.32157970	1.32340274
8	-0.22506270	-0.22549845
9	0.45888083	0.46009976
10	-0.75724781	-0.75992923
11	-0.06052647	-0.06080437
12	-0.53431842	-0.53742251

Valores ajustados:

```
valores.ajustados<-cbind(bacterias,fit$fitted.values,fit.mlg$v.ajustado)
colnames(valores.ajustados)<-c("bacterias","glm","mlgPoisson")
valores.ajustados
```

	bacterias	glm	mlgPoisson
1	175	160.24051	160.24051
2	108	127.45716	127.45716
3	95	101.38091	101.38091
4	82	80.63955	80.63955
5	71	64.14163	64.14163
6	50	51.01900	51.01900
7	49	40.58110	40.58110
8	31	32.27868	32.27868
9	28	25.67484	25.67484
10	17	20.42206	20.42206
11	16	16.24394	16.24394
12	11	12.92062	12.92062

Preditor Linear:

```
preditor.linear<-cbind(fit$linear.predictors,fit.mlg$preditor.linear)
colnames(preditor.linear)<-c("glm","mlgPoisson")
preditor.linear
```

	glm	mlgPoisson
1	5.076676	5.076676
2	4.847780	4.847780
3	4.618885	4.618885
4	4.389989	4.389989
5	4.161094	4.161094
6	3.932198	3.932198
7	3.703302	3.703302
8	3.474407	3.474407
9	3.245511	3.245511
10	3.016616	3.016616
11	2.787720	2.787720
12	2.558825	2.558825

### Questão 1.c

Gerando um conjunto de dados:

```
x=cbind(rep(1,100),runif(100,0,2))
beta<-c(1,3)
y=rpois(n=100,lambda=exp(x%*%beta))
```

Ajuste usando o glm:

```
fit=glm(y~x-1,family=poisson(link = "log"))
summary(fit)
```

Call:

```
glm(formula = y ~ x - 1, family = poisson(link = "log"))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.63742	-0.87580	0.06836	0.81999	2.11842

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
x1	0.98779	0.04154	23.78	<2e-16 ***
x2	3.00987	0.02487	121.02	<2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 172159.99 on 100 degrees of freedom  
 Residual deviance: 114.85 on 98 degrees of freedom  
 AIC: 693.58

Number of Fisher Scoring iterations: 4

Ajuste usando a função criada:

```
init=solve(t(x)%*%x)%*%t(x)%*%sqrt(y)
fit.mlg<-mlgPoisson(y,x,init,eps=1e-6,maxiter=50)
```

-----  
 MLG para o modelo Poisson com função de ligação log  
 -----

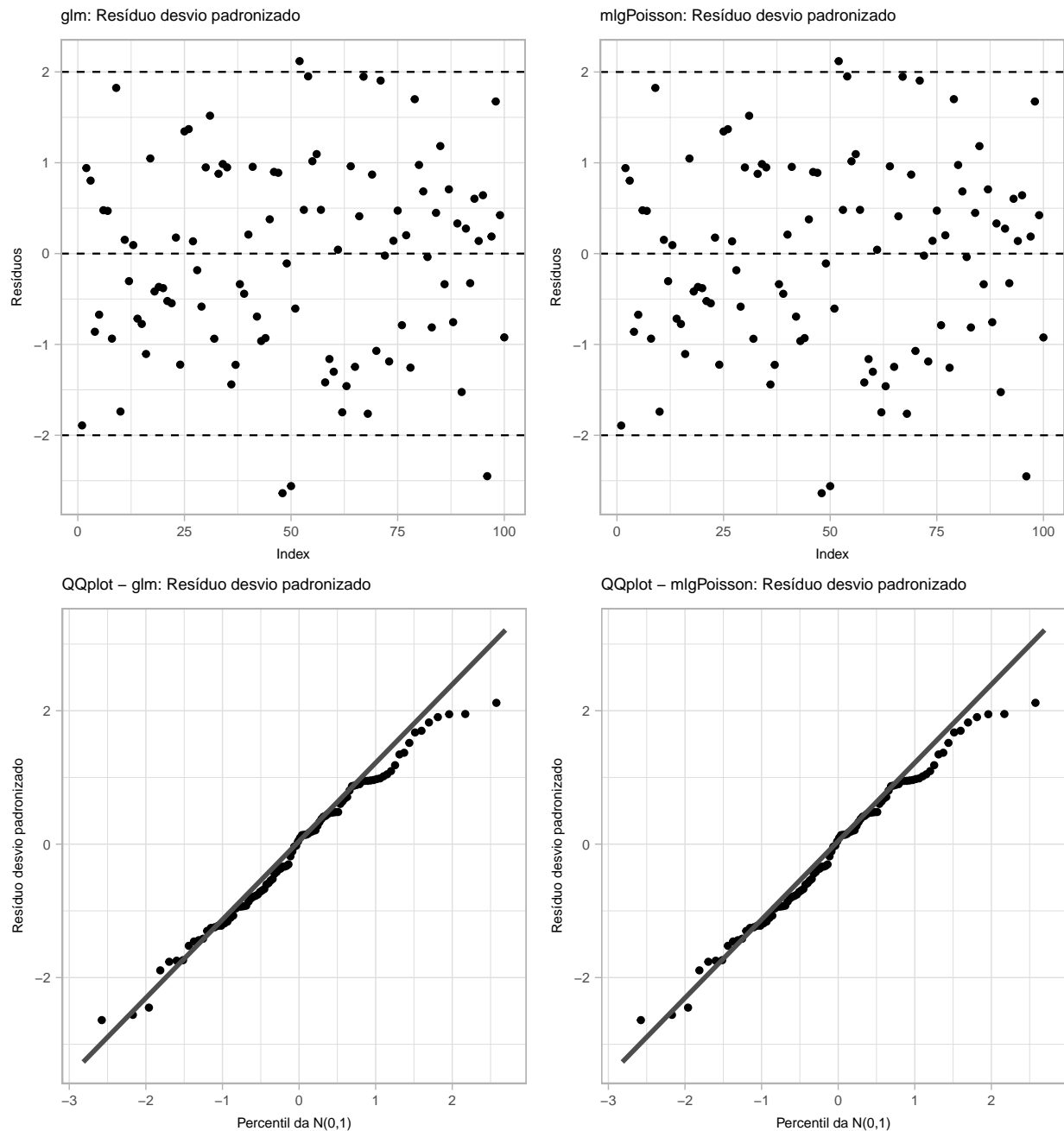
	Estimativas	SE	valor z	p-valor	IC's
x1	0.9878	0.0415	23.7785	2.788136e-125	[0.91;1.07]
x2	3.0099	0.0249	121.0170	0.000000e+00	[2.96;3.06]

-----  
 g.l  
 Desvio: 114.855 98  
 -----

Número de iterações:  
 21  
 -----

Critérios de seleção do Modelo

AIC: 693.575  
 BIC: 707.996  
 -----



### Questão 1.d - Estudo de simulação.

```

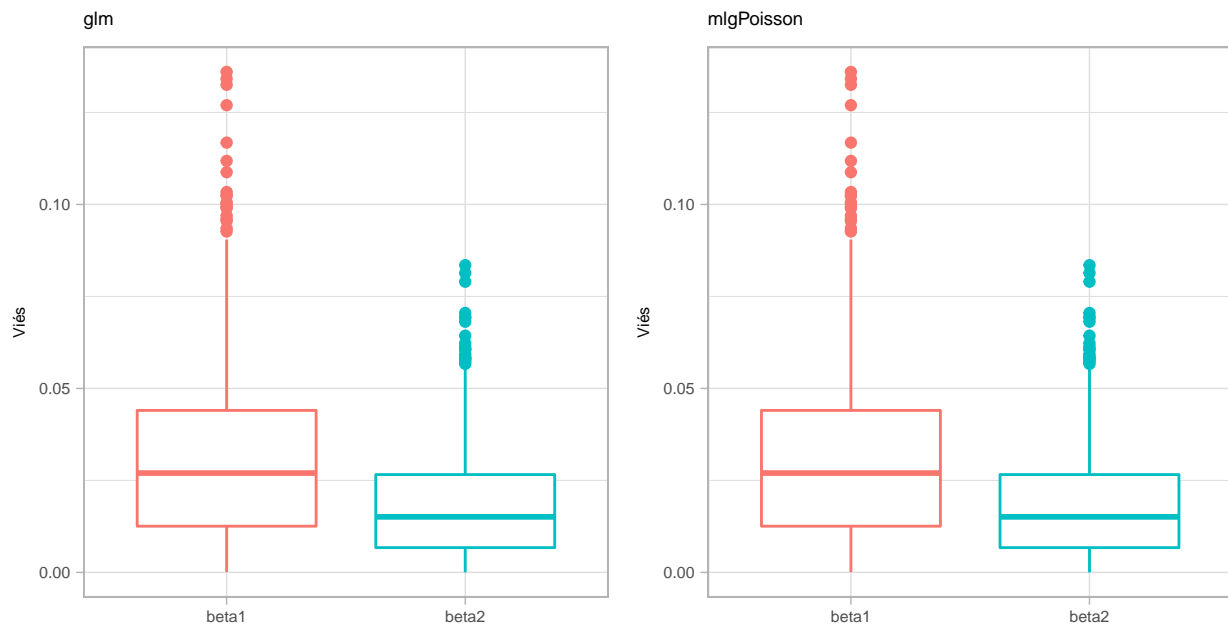
N=1000
betasglm<-betas<-matrix(NA,N,2)
aicglm<-aic<-desvioglm<-desvio<-NULL

for(i in 1:N){
  set.seed(1000*i)
  x=cbind(rep(1,100),runif(100,0,2))
  beta<-c(1,3)
  y=rpois(n=100,lambda=exp(x%*%beta))
  fit.glm=glm(y~x-1,family=poisson(link = "log"))
  betasglm[i,]<-fit.glm$coef
  aicglm[i]<-fit.glm$aic
  desvioglm[i]<-fit.glm$deviance
  init<-matrix(solve(t(x)%*%x)%*%t(x)%*%sqrt(y),2,1)
  fit.mlgPoisson=ajustePoisson(y,x,init,eps=1e-6,maxiter=100)
  betas[i, ]<-fit.mlgPoisson$est
  aic[i]<-fit.mlgPoisson$AIC
}

```

```
desvio[i]<-fit.mlgPoisson$desvio
}
```

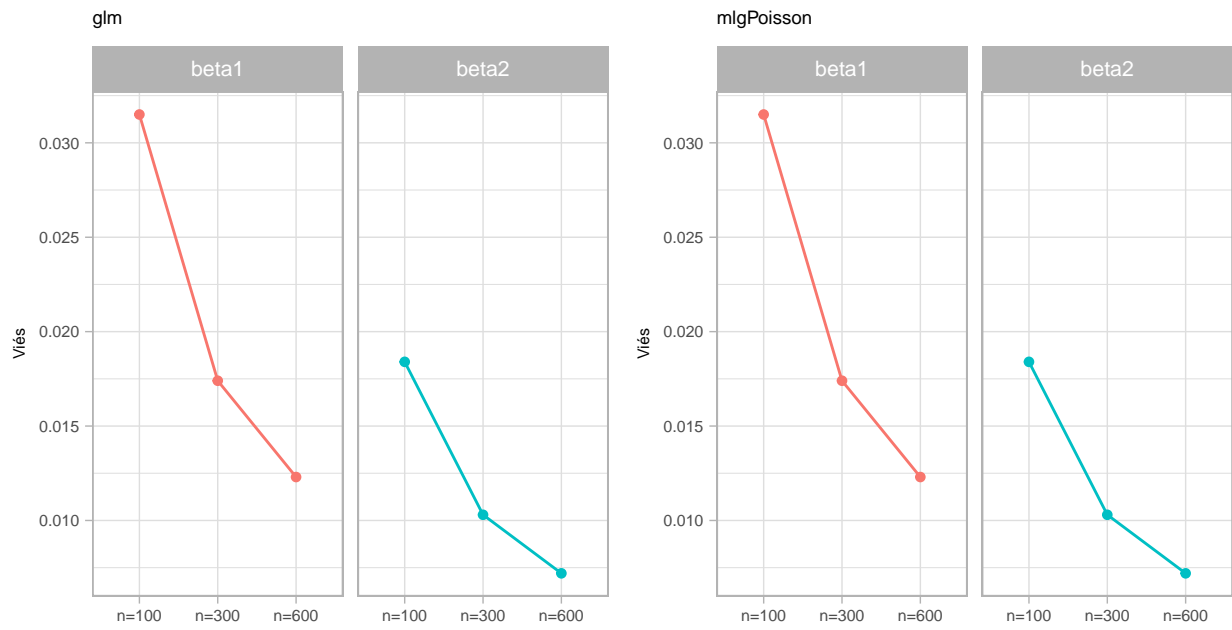
```
beta<-matrix(c(1,3),1000,2,byrow=T)
viesglm<-abs(betasglm-beta)
vies<-abs(betas-beta)
```



	glm	mlgPoisson
$\beta_1$	1.0003	1.0003
$\beta_2$	2.9999	2.9999
AIC	684.744	684.744
Desvio	98.887	98.887

### Questão 1.e - Estudo de simulação.

	n=100		n=300		n=600	
	glm	mlgPoisson	glm	mlgPoisson	glm	mlgPoisson
$\beta_1$	1.0003	1.0003	0.9999	0.9999	0.9996	0.9996
$\beta_2$	2.9999	2.9999	3	3	3.0002	3.0002
AIC	684.744	684.744	2051.196	2051.196	4099.517	4099.517
Desvio	98.887	98.887	302.329	302.329	606.15	606.15





### Questão 3

Gerando um conjunto de dados:

```
x1=rnorm(100,10,2)
x2=rnorm(100,5,1)
x=cbind(x1,x2)
beta<-c(-1,2)
y=rbinom(n = 100, size = 1, prob = exp(x%*%beta)/(1+exp(x%*%beta)))
```

Ajuste usando o glm:

```
fit=glm(y~x1+x2-1,family=binomial(link = "logit"))
summary(fit)
```

Call:

```
glm(formula = y ~ x1 + x2 - 1, family = binomial(link = "logit"))
```

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.84641	-0.50652	-0.02992	0.62527	1.76608

Coefficients:

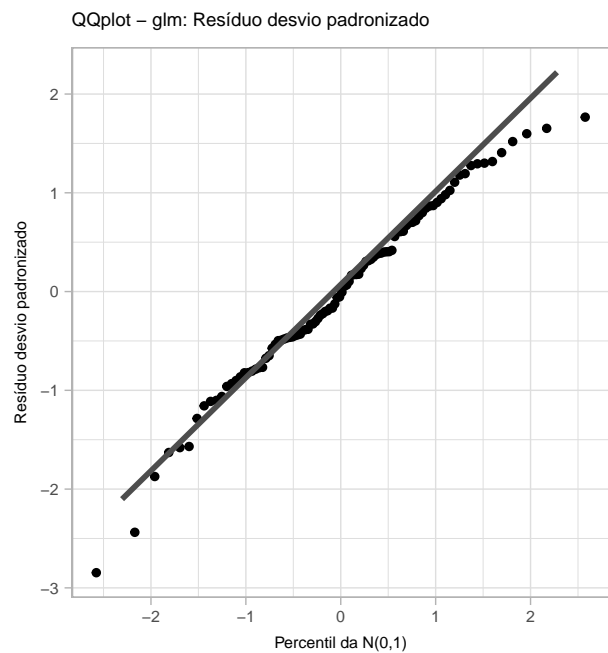
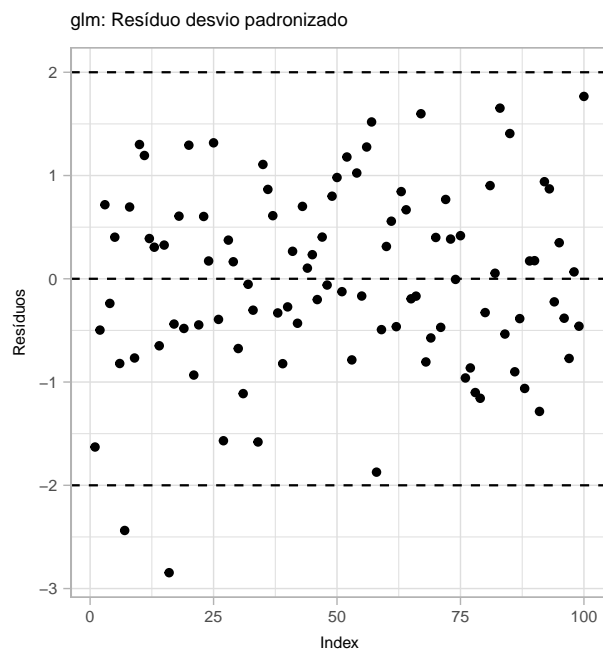
	Estimate	Std. Error	z value	Pr(> z )
x1	-0.9601	0.1926	-4.985	6.19e-07 ***
x2	1.9293	0.3904	4.942	7.74e-07 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 138.629 on 100 degrees of freedom  
Residual deviance: 79.162 on 98 degrees of freedom  
AIC: 83.162

Number of Fisher Scoring iterations: 6



### Questão 3.d - Estudo de simulação.

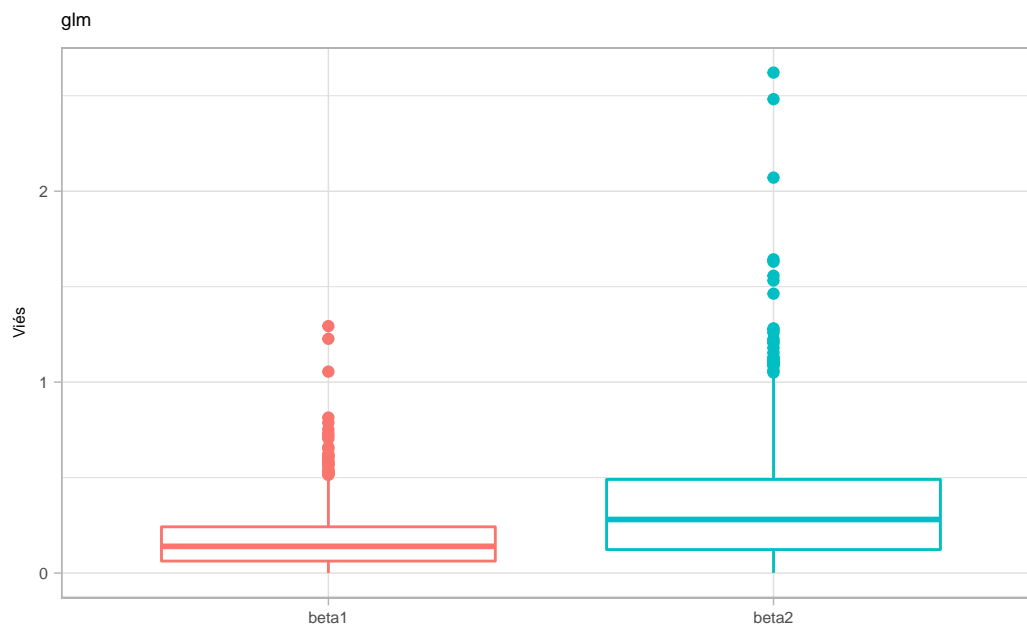
```

N=1000
betasglm<-matrix(NA,N,2)
aicglm<-desvioglm<-NULL

for(i in 1:N){
  set.seed(1000*i)
  x1=rnorm(100,10,2)
  x2=rnorm(100,5,1)
  x=cbind(x1,x2)
  beta<-c(-1,2)
  y=rbinom(n = 100, size = 1, prob = exp(x%*%beta)/(1+exp(x%*%beta)))
  fit.glm=glm(y~x1+x2-1,family=binomial(link = "logit"))
  betasglm[i,]<-fit.glm$coef
  aicglm[i]<-fit.glm$aic
  desvioglm[i]<-fit.glm$deviance
}

beta<-matrix(c(-1,2),1000,2,byrow=T)
viesglm<-abs(betasglm-beta)

```



	n=100	n=300	n=500
$\beta_1$ (-1)	-1.0536	-1.018	-1.0116
$\beta_2$ (2)	2.1059	2.0378	2.0226
AIC	77.416	227.604	377.958
Desvio	73.416	223.604	373.958

