tp5

Trabajo práctico 5

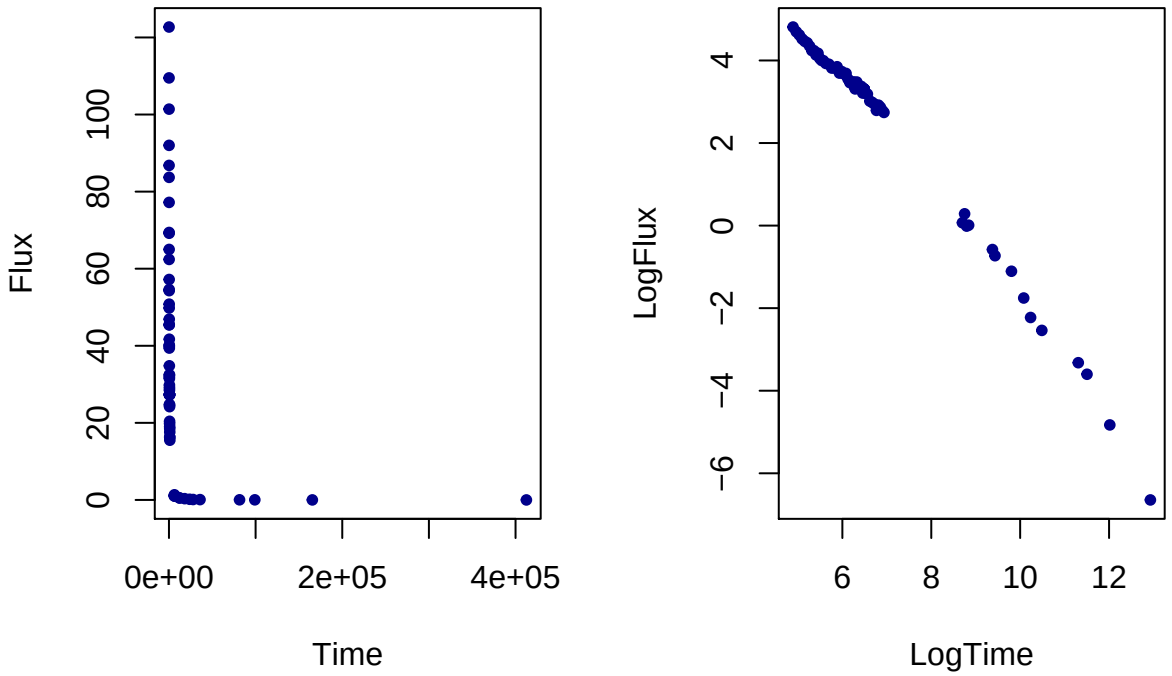
Ejercicio 1

Cargamos la librería y la base de datos GRB

```
library(HoRM)
library(ggplot2)
data(GRB)
```

a) Graficamos la variable Flux en función de Time, y lo mismo pero aplicando logaritmos

```
par(mfrow=c(1,2))
plot(GRB$TIME,GRB$FLUX,pch=20,col="darkblue",xlab="Time",ylab="Flux")
plot(log(GRB$TIME),log(GRB$FLUX),pch=20,col="darkblue",
     xlab="LogTime",ylab="LogFlux")
```




```
par(mfrow=c(1,1))
```

Observamos que la escala más adecuada es utilizando logaritmos

b) Parece razonable realizar el ajuste usando un modelo lineal de a trozos, observando el gráfico podría partirse en 2 o 3 partes.

c) Realizamos el ajuste de los tres modelos según lo propone el enunciado del ejercicio:

```
#definimos las transformaciones de las variables

Ltime<-log(GRB$TIME)
Ltime7<-(log(GRB$TIME)-7)*(log(GRB$TIME)>7)
Ltime8<-(log(GRB$TIME)-8.5)*(log(GRB$TIME)>8.5)
```

```r
#realizamos los tres ajustes
ajusteM1<- glm(GRB$FLUX~ Ltime,family = Gamma(link=log))
ajusteM2<- glm(GRB$FLUX~ Ltime+Ltime7,family = Gamma(link=log))
ajusteM3<- glm(GRB$FLUX~ Ltime+Ltime8,family = Gamma(link=log))

#analizamos las salidas
summary(ajusteM1)
```

```
##
## Call:
## glm(formula = GRB$FLUX ~ Ltime, family = Gamma(link = log))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.96427  -0.26609   0.01729   0.20662   0.31707
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.66499    0.11859   98.36   <2e-16 ***
## Ltime       -1.32737    0.01646  -80.66   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.06092683)
##
##     Null deviance: 143.5982  on 62  degrees of freedom
## Residual deviance:   4.3423  on 61  degrees of freedom
## AIC: 319.36
##
## Number of Fisher Scoring iterations: 7
```

```r
summary(ajusteM2)
```

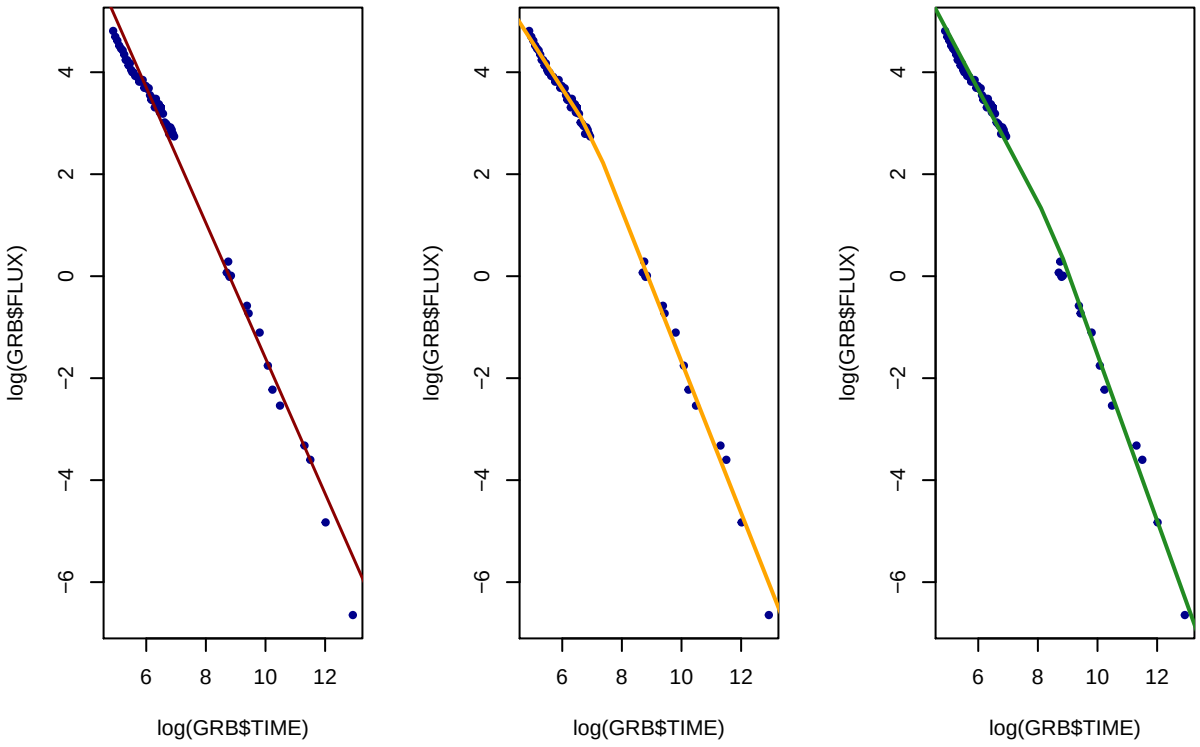
```
##
## Call:
## glm(formula = GRB$FLUX ~ Ltime + Ltime7, family = Gamma(link = log))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.56594  -0.07827  -0.00316   0.07467   0.32247
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.13456    0.18229   50.11   <2e-16 ***
## Ltime       -0.90942    0.02986  -30.46   <2e-16 ***
## Ltime7      -0.57274    0.03895  -14.70   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.01625884)
##
##     Null deviance: 143.5982  on 62  degrees of freedom
## Residual deviance:   1.0273  on 60  degrees of freedom
## AIC: 229.99
```

```
##
## Number of Fisher Scoring iterations: 5
```

```
summary(ajusteM3)
```

```
##
## Call:
## glm(formula = GRB$FLUX ~ Ltime + Ltime8, family = Gamma(link = log))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.46521  -0.12634  -0.01121   0.11988   0.41152
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.28587    0.16405  62.701  < 2e-16 ***
## Ltime       -1.10413    0.02586 -42.693  < 2e-16 ***
## Ltime8      -0.52268    0.05308  -9.848 3.81e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.03030429)
##
##     Null deviance: 143.5982  on 62  degrees of freedom
## Residual deviance:   1.8675  on 60  degrees of freedom
## AIC: 267.78
##
## Number of Fisher Scoring iterations: 6
```

```
#observamos gráficamente los ajustes
grilla<-seq(0,14,length=20)
y1<-ajusteM1$coefficients[1]+ajusteM1$coefficients[2]*grilla

y2<-ajusteM2$coefficients[1]+ajusteM2$coefficients[2]*grilla+
  ajusteM2$coefficients[3]*(grilla-7)*(grilla>7)

y3<-ajusteM3$coefficients[1]+ajusteM3$coefficients[2]*grilla+
  ajusteM3$coefficients[3]*(grilla-8.5)*(grilla>8.5)

par(mfrow=c(1,3))
plot(log(GRB$TIME),log(GRB$FLUX),pch=20,col="darkblue")
lines(grilla,y1,col="darkred",lwd=1.5)

plot(log(GRB$TIME),log(GRB$FLUX),pch=20,col="darkblue")
lines(grilla,y2,col="orange",lwd=2)

plot(log(GRB$TIME),log(GRB$FLUX),pch=20,col="darkblue")
lines(grilla,y3,col="forestgreen",lwd=2)
```




```
par(mfrow=c(1,1))


#podemos comparar los modelos anidados usando anova
anova(ajusteM1,ajusteM3,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: GRB$FLUX ~ Ltime
## Model 2: GRB$FLUX ~ Ltime + Ltime8
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        61     4.3423
## 2        60     1.8675  1   2.4748 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

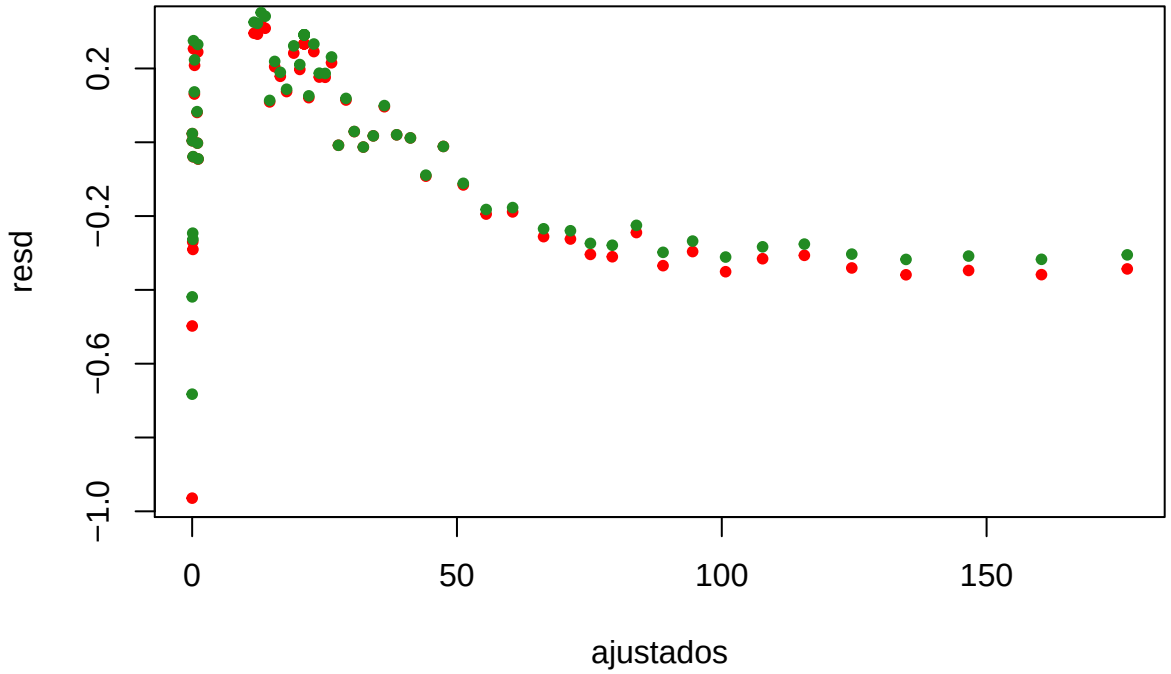
```
anova(ajusteM1,ajusteM2,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: GRB$FLUX ~ Ltime
## Model 2: GRB$FLUX ~ Ltime + Ltime7
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        61     4.3423
## 2        60     1.0273  1    3.315 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

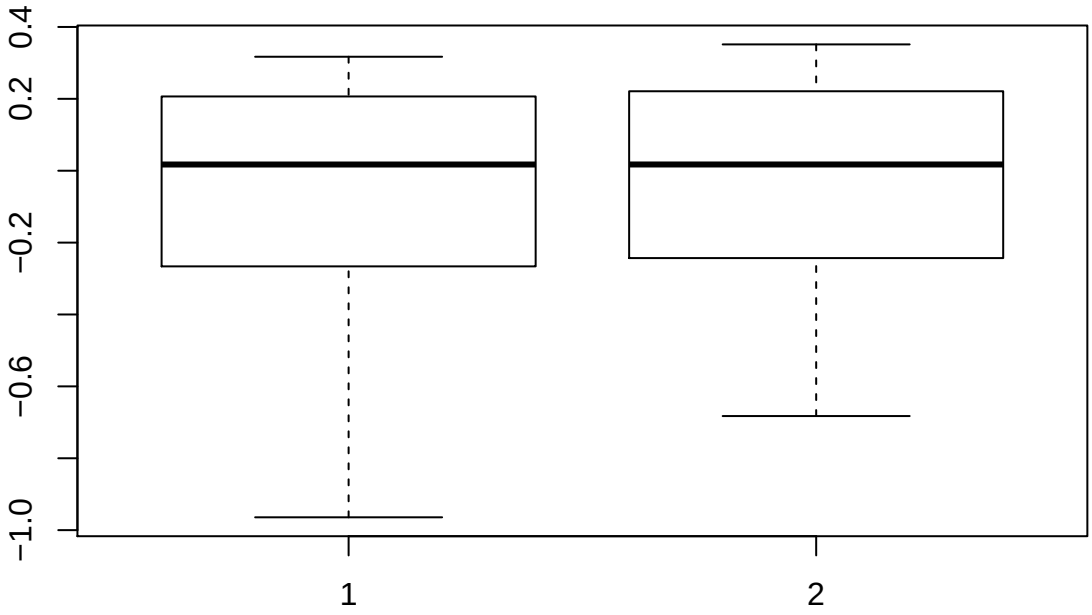
En los gráficos observamos que los modelos 2 y 3 se ajustan mucho mejor a los datos. Podemos realizar la tabla de ANOVA para comparar el modelo 1 con el 3 o el 1 con el 2, ya que están anidados, pero no así para comparar el modelo 2 con el 3. Para comparar estos dos modelos observamos los valores de AIC que aparecen en la salida de glm. A partir de toda esta información concluimos que el modelo 2 da un mejor ajuste.

d) Analizamos los residuos Deviance y Pearson. Esperamos ver puntos sin estructura al rededor del 0.

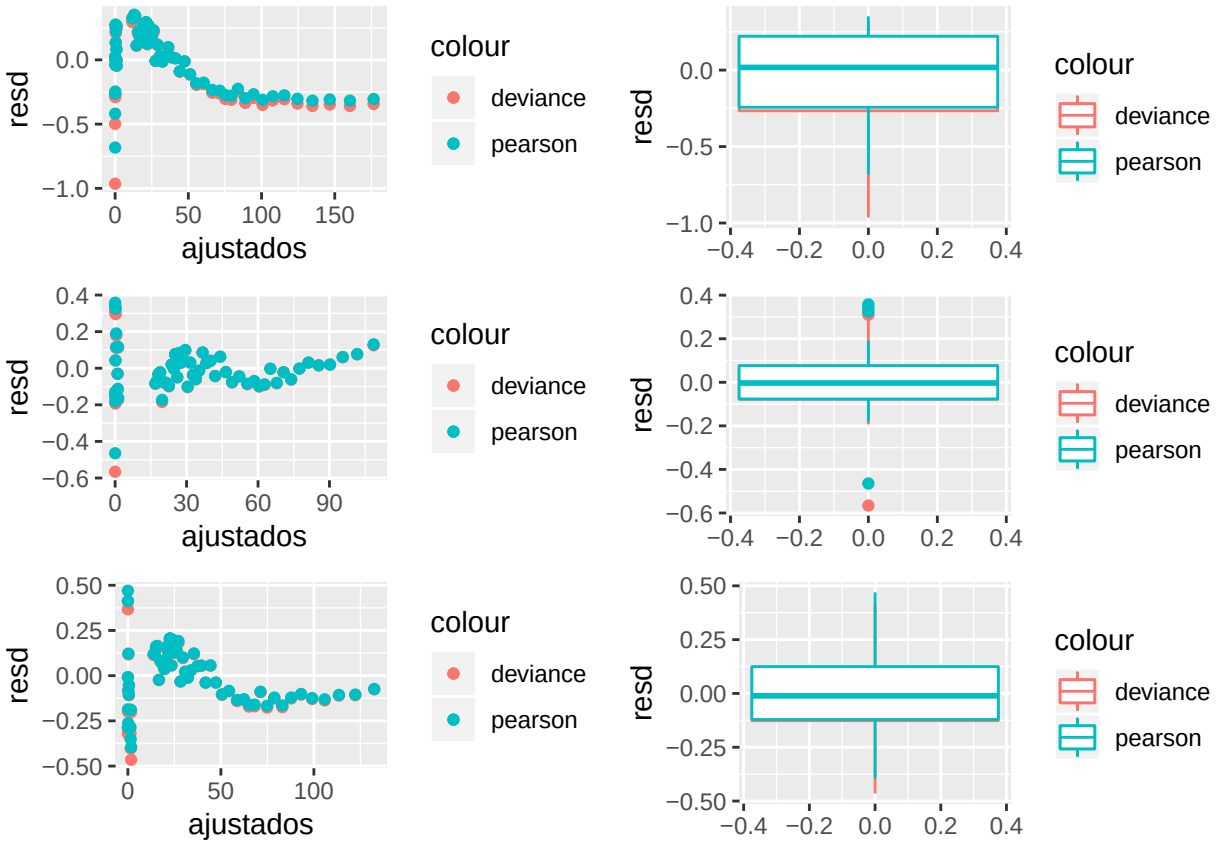
```
resd<-resid(ajusteM1,type="deviance")
resp<-resid(ajusteM1,type="pearson")
ajustados<-ajusteM1$fitted.values

datos<-data.frame(cbind(ajustados,resd,resp))

plot(ajustados,resd,pch=20,col="red")
points(ajustados,resp,pch=20,col="forestgreen")
```




```
boxplot(resd,resp)
```



```r
g1<-ggplot(datos)+
  geom_point(aes(ajustados,resd,col="deviance"))+
  geom_point(aes(ajustados,resp,col="pearson"))

g2<-ggplot(datos)+
  geom_boxplot(aes(y=resd,col="deviance"))+
  geom_boxplot(aes(y=resp,col="pearson"))

resd<-resid(ajusteM2,type="deviance")
resp<-resid(ajusteM2,type="pearson")
ajustados<-ajusteM2$fitted.values

datos<-data.frame(cbind(ajustados,resd,resp))

g3<-ggplot(datos)+
  geom_point(aes(ajustados,resd,col="deviance"))+
  geom_point(aes(ajustados,resp,col="pearson"))

g4<-ggplot(datos)+
  geom_boxplot(aes(y=resd,col="deviance"))+
  geom_boxplot(aes(y=resp,col="pearson"))

resd<-resid(ajusteM3,type="deviance")
resp<-resid(ajusteM3,type="pearson")
ajustados<-ajusteM3$fitted.values

datos<-data.frame(cbind(ajustados,resd,resp))

g5<-ggplot(datos)+
  geom_point(aes(ajustados,resd,col="deviance"))+
  geom_point(aes(ajustados,resp,col="pearson"))

g6<-ggplot(datos)+
  geom_boxplot(aes(y=resd,col="deviance"))+
  geom_boxplot(aes(y=resp,col="pearson"))


library(gridExtra)
grid.arrange(g1, g2, g3, g4,g5,g6, ncol=2, nrow =3)
```



En todos los gráficos se observa estructura, pero de todos el que menos estructura presenta es el del modelo 2.

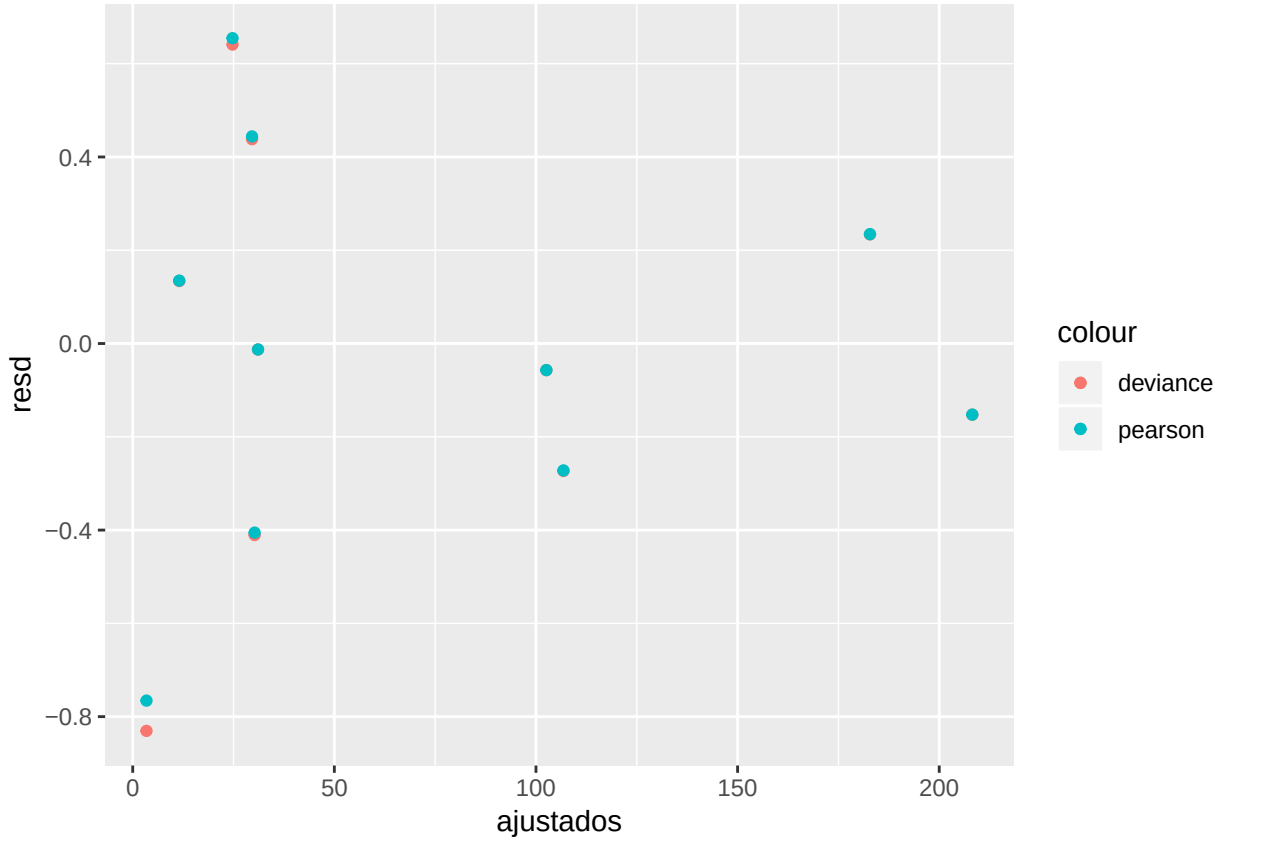
Ejercicio 2

Analizamos los residuos para el ejercicio 2 del trabajo práctico 4

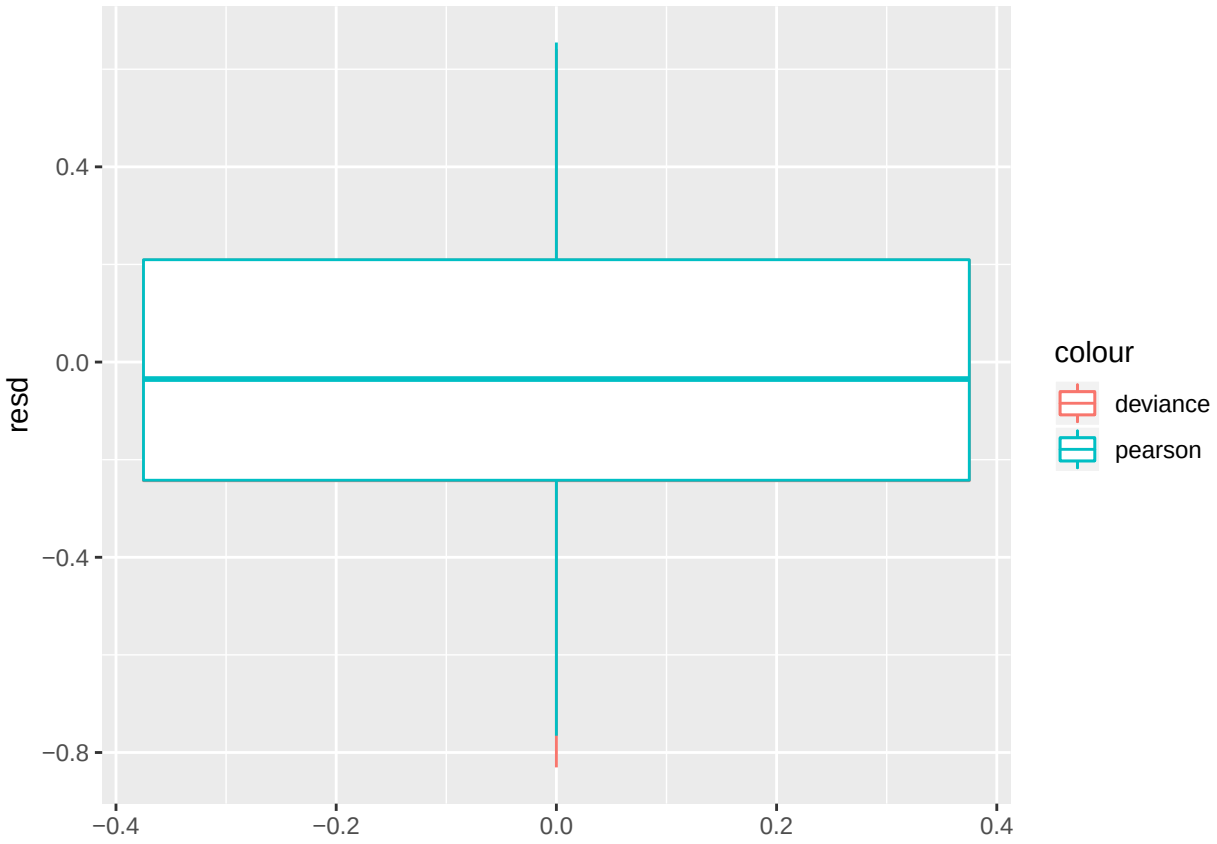
```
y <- c(32,104,206,186,102,2,12,28,28,31)
smoke<- c(rep(1,5),rep(0,5))
age <- c(1:5,1:5)
pop<- c(52407,43248,28612,12663,5317,18790,10673,5710,2585,1462)

logpop<- log(pop)
age2 <- age*age
smkage<- smoke*age
ajuste<-glm(y~age+age2+smoke+smkage, offset = logpop, family = poisson,x=T)

resd<-resid(ajuste,type="deviance")
resp<-resid(ajuste,type="pearson")
ajustados<-ajuste$fitted.values

datos<-data.frame(cbind(ajustados,resd,resp))

g7<-ggplot(datos)+
  geom_point(aes(ajustados,resd,col="deviance"))+
  geom_point(aes(ajustados,resp,col="pearson"))

g8<-ggplot(datos)+
```

```
  geom_boxplot(aes(y=resd,col="deviance"))+
  geom_boxplot(aes(y=resp,col="pearson"))
```

```
g7
```




```
g8
```



Ejercicio 3

Bondad de ajuste del modelo Binomial

Analizamos nuevamente los datos birthwt.

a) EN primer lugar transformamos en factores a las variables categóricas según el enunciado:

```
birth<- read.table("birthwt.txt",header=T)
frace<-factor(birth$race)
fsmoke<-factor(birth$smoke)
fht<-factor(birth$ht)
fui<-factor(birth$ui)
```

b) Creamos la variable nueva LWD

```
lwd <- 1*(birth$lwt< 110)
lwd<- rep(0,length(birth$lwt))
lwd<- 1*(birth$lwt==1)
```

c) Realizamos el ajuste y analizamos la significación de las variables

```
ajuste<- glm(low~age+lwd+frace+fsmoke+fht+fui, data=birth,family=binomial)
summary(ajuste)
```

```
##
## Call:
## glm(formula = low ~ age + lwd + frace + fsmoke + fht + fui, family = binomial,
##     data = birth)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6638  -0.8295  -0.5586   1.0909   2.1003
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.34436    0.90536  -1.485  0.13757
## age         -0.03102    0.03464  -0.896  0.37046
## lwd               NA         NA      NA       NA
## frace2       0.99809    0.50210   1.988  0.04683 *
## frace3       1.03863    0.41824   2.483  0.01302 *
## fsmoke1      1.07485    0.38221   2.812  0.00492 **
## fht1         1.35424    0.62728   2.159  0.03086 *
## fui1         0.97804    0.44149   2.215  0.02674 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 210.35  on 182  degrees of freedom
## AIC: 224.35
##
## Number of Fisher Scoring iterations: 4
```

d) Análizamos los estadísticos de bondad de ajuste, esto es: calculamos el estadístico Deviance y Pearson y los comparamos con sus grados de libertad. De ser bueno el ajuste, estos valores deberían ser cercanos a 1.

```
sum(resid(ajuste,type="deviance")*resid(ajuste,type="deviance"))/summary(ajuste)$df.residual
```

```
## [1] 1.155778
```

```
sum(resid(ajuste,type="pearson")*resid(ajuste,type="pearson"))/summary(ajuste)$df.residual
```

```
## [1] 1.004264
```

e) Realizamos el test de Hosmer-Lemeshow para evaluar la significación de la bondad de ajuste

```
library(ResourceSelection)
```

```
## ResourceSelection 0.3-5   2019-07-22
```

```
hoslem.test(birth$low, ajuste$fitted.values, g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  birth$low, ajuste$fitted.values
## X-squared = 12.77, df = 8, p-value = 0.12
```

En ambos casos concluimos que el ajuste es bueno. Sería bueno un p-valor mayor a 0.2 en el test para poder suponer que el modelo es correcto. Pero como las medidas del ítem anterior dieron cercanas a 1 y el p-valor es mayor a 0.1 podemos concluir que el ajuste es válido.