Cómputo Estadístico (Tarea1)

José Antonio García Ramírez

Agosto 23, 2018

1. *Elija una distribución de la familia exponencial (excepto normal, binomial y poisson ya que esto se impartirá en la clase). Escriba los tres componentes (componente aleatoria, componente sistemática y función de enlace o link) del modelo lineal generalizado. Defina y derive θ, ϕ, $a()$, $b()$, $c()$, $E(Y)$, η, $E(\frac{\partial l}{\partial \beta_j})$ y $E[(\frac{\partial^2 l}{\partial \beta_j \beta_i})]$ como se mostró en la clase según la distribución que eligió para esta tarea.*

El pasado martes 14 de agosto, en clase de cómputo estadístico vimos la siguiente forma cerrada de la familia exponencial, para los casos vistos en clase (distribuciones con solo dos parámetros), cuando la distribución de interes posee paámetros de localización y de escala.

$$f(y;\theta) = exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

Donde θ es el paámetro de escala y ϕ es el parámetro de localización, definición que puede consultarse en (Myers et al. 2012).

Sin embargo para el caso de las funciones de densidad con cualquier número de parámetros existe una definición más general.

$$f(y;\theta) = h(y)exp\left\{T(y)\eta(\theta) - b(\theta)\right\}$$

Donde $\theta \in R^k$ es el vector de parámetros, definición que puede consultarse en (Bickel and Doksum 2001).

Elegí la distribución de Rayleigh, por sus aplicaciones en teoría de confiabilidad. Y la lleve a la forma vista en clase considerando que esta distribución sólo tiene parámetro.

Así se tiene:

$f(y;\sigma^2) = \frac{y}{\sigma^2}exp\{-y^2/(2\sigma^2)\} = exp\{\ln(\frac{y}{\sigma^2})\}exp\{-y^2/(2\sigma^2)\} = exp\{\ln(\frac{y}{\sigma^2}) - y^2/(2\sigma^2)\} = exp\{y^2(-1/(2\sigma^2)) + \ln(\sigma^2) + \ln(y)\}$

Considerando un cambio de varable, $\theta = -1/(2\sigma^2) \Rightarrow \sigma^2 = -1/(2\theta)$ tenemos que: $g(\mu) = \eta, \eta(\theta) = \theta, T(y) = y^2, b(\theta) = \ln(-1/(2\theta)) = -\ln(-2\theta), a(\phi) = 1$ y $c(y,\phi) = c(y) = \ln(y)$

Se tiene que la definición de (Bickel and Doksum 2001) coincide con la de (Myers et al. 2012), con $a(\phi) = 1$.

La segunda parte de la tarea consiste en deducir $\frac{\partial l}{\partial \beta}$ y $E\left(\frac{\partial^2 l}{\partial^2 \beta_j \beta_i}\right)$,para deducir la entrada i-ésima de $\frac{\partial}{\partial \beta}$ al igual que la entrada (j,i) de $E\left(\frac{\partial^2 l}{\partial^2 \beta_j \beta_i}\right)$, realizaré unos cálculos para facilitar la notación y ser más breve.

$$\frac{\partial}{\partial \theta}\ln(f(y;\sigma^2)) = \frac{\partial}{\partial \theta}(T(y)\theta - b(\theta) + c(y)) = T(y) - b'(\theta)$$

Igualando a cero y sacando esperanzas tenemos:

$$E(T(y)) = b'(\theta) = \mu$$

De nuestros cursos de inferencia estadística sabemos que $E(l''(\theta,y)) = -E(l'(\theta,y)^2)$, por lo que $Var(l'(\theta,y)) = -E(l''(\theta,y)) = b''(\theta)$

Entonces

$$\frac{\partial l}{\partial \beta_i} = \frac{\partial l}{\partial \theta}\frac{\partial \theta}{\partial \mu}\frac{\partial \mu}{\partial \eta}\frac{\partial \eta}{\partial \beta_i}$$

Falta calcular

$$\frac{\partial l}{\partial \theta} = (T(y) - b'(\theta)) = T(y) - \mu$$

$$\frac{\partial \theta}{\partial \mu} = 1/b''(\theta)$$

$$\frac{\partial \mu}{\partial \eta} = 1$$

$$\frac{\partial \eta}{\partial \beta_i} = x_i$$

Entonces

$$\frac{\partial l}{\partial \beta_i} = (T(y) - \mu)(1/b''(\theta))x_i = x_i \frac{y^2 - b'(\theta)}{b''(\theta)} = x_i((y^2\theta + 1)/(-\theta^3)) = x_i\theta(y^2\theta + 1)$$

Finalmente

$$\frac{\partial^2 l}{\partial \beta_j \beta_i} = \frac{\partial}{\partial \theta} x_i\theta(y^2\theta + 1)\frac{\partial \theta}{\partial \mu}\frac{\mu}{\eta}\frac{\eta}{\beta_j} = -x_i x_j(1 - 2y^2 + (1/\theta))$$

Y

$$E(\frac{\partial^2 l}{\partial \beta_j \beta_i}) = -x_i x_j(1 - 2E(y^2) + (1/\theta)) = -x_i x_j(1 - 2(-1/\theta) + (1/\theta)) = -x_i x_j(1 + 3/\theta)$$

Así $\theta = -1/(2\sigma^2)$, ϕ no existe para esta distribución i.e. vale cero,$a(\phi) = 1$, $b(\theta) = \ln(-1/(2\theta)) = -\ln(-2\theta)$, $c(y, \phi) = c(y) = \ln(y)$, $E(T(y) = y^2) = -b'(\theta)$, la función de link es la identidad $\eta(\theta) = \theta$ y al considerar verosimilitudes de una muestra con n observaciones y m variables $\frac{\partial l}{\partial \beta_i} = \sum_{k=1}^{n} x_{ki}\theta(y^2\theta + 1)$ y $E(\frac{\partial^2 l}{\partial \beta_j \beta_i}) = -\sum_{k=0}^{n} x_{ki}x_{kj}(1 + 3/\theta)$, donde x_{ki} es la k-ésima observacion de la variable i-ésima para calcular $(E(\frac{\partial^2 l}{\partial \beta_j \beta_i}))^{-1}$ requerimos de una muestra fija.

2. *Haga un grupo de dos estudiantes y cada grupo debe ajustarse al modelo lineal generalizado en uno de de los conjuntos de datos de los archivos:* **cocmo_nuermic.arff**, **detatrieve.arff**, **desharnails.arff**, **humans_numeric.arff**, **nasa_numeric.arff**, **usp05-ft.arff**, **usp05.arff** *y* **cocomonasa.arff** *disponibles en http://tunedit.org/repo/PROMISE/EffortPrediction y http://promise.site.uottawa.ca/SERepository/datasets/cocomonasa.arff .*

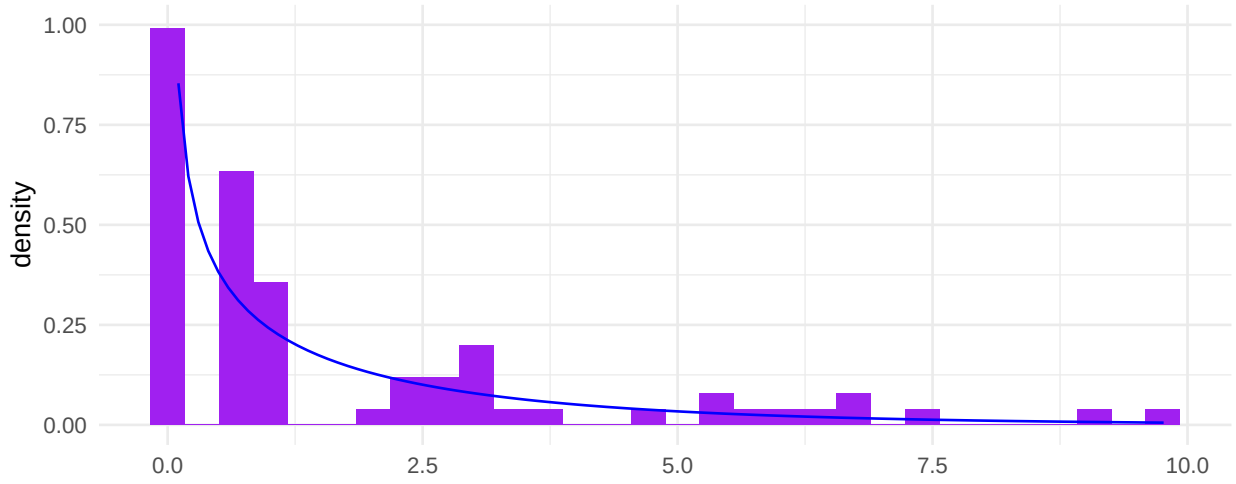
Junto a mi compañero Adrián Alejandro Rodriguez decidimos trabajar con el archivo **humans_numeric.arff**, pues relaciona conceptos de ingeniería de software y estimación humana empírica (algo muy común la industria del desarrollo del software).

Comenzamos considerando la distribución de la variable de respuesta que es el cociente entre la diferencia del tiempo estimado por una persona y el que en verdad tardó el desarrollo entre el mismo tiempo de desarrollo verdadero, denotaremos como y a esta variable (los detalles del contenido de las variables se puede consultar en http://tunedit.org/repo/PROMISE/EffortPrediction.

Después de varios intentos (entre distribuciones lognormales) ajustamos una distribución gamma.

```
rm(list = ls())
link <- 'log'
data <- read.csv('data.csv')
library(MASS)
param <- fitdistr(data$y,"gamma") ## fitting gamma pdf parameters
library(ggplot2)
ggplot(data = data, aes(x = y)) + geom_histogram(aes(y=..density.., fill = I('purple')),
position="identity") +  stat_function(fun = dgamma,args = list(shape=param$estimate[1],
```

```
rate =param$estimate[2]),colour=I('blue')) +
  theme_minimal() +   xlab('') + ylim(c(0 ,1))
```




```
set.seed(0)
ks.test(data$y, 'pgamma', param$estimate[1], param$estimate[2])
```

```
## Warning in ks.test(data$y, "pgamma", param$estimate[1], param$estimate[2]):
## ties should not be present for the Kolmogorov-Smirnov test
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  data$y
## D = 0.16641, p-value = 0.03141
## alternative hypothesis: two-sided
```

Si bien, visualmente, el ajuste parece adecuado realizamos una el test de Kolmogorov-Smirnov para afirmar que la distribución de nuestra variable a predecir es en efecto una gamma. El resultado del test es el rechazo de la hipótesis de igualdad en distribución sin embargo el p-value en considerable ($>.03$) y tomando en cuenta el gran sesgo positivo de la distribución muestral además de que no cubre todo el soporte de la v.a. gamma, consideramos prudente suponer que la variable de respuesta es una gamma.

En un inicio consideramos estimaciones de OLS sobre la variable log(y), los resultados eran modelos con parámetros individuales y en conjunto significativos sin embargo los residuales distaban de ser normales.

Se procedió a usar glm con diferentes funciones de densidad (gaussiana, inversa de la gaussiana,...) y diversos links sin embargo el resultado el modelo aceptado y presentado es un glm con distribución gamma con link 'log'.

Después de estimar los parámetros y utilizar el algoritmo de stepwise, para eliminar parámetros y por ende variables en vista de que originalmente tenemos 16 variables y solo 75 observaciones (lo que puede disminuir la calidad de las estimaciones por la maldición de la dimensionalidad) el modelo propuesto es el siguiente:

```
data$Degree <- factor(data$Degree)
no.lineal <- glm(y ~ ., family = Gamma(link=link), data = data )
summary(no.lineal)
no.lineal2 <- step(no.lineal)
```

```
summary(no.lineal2)
```

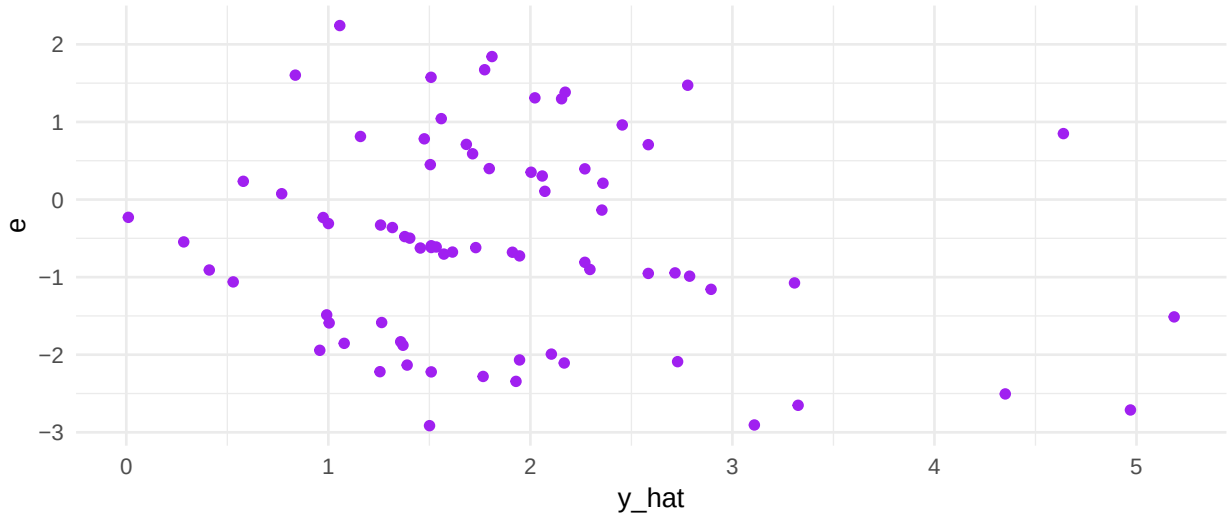
```
##
## Call:
## glm(formula = y ~ MgmtUGCourses + MgmtGCourses + Total.Workshops +
##     TotalLangExp + Hardware.Proj.Mgmt.Exp, family = Gamma(link = link),
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9143  -1.5878  -0.6183   0.3972   2.2420
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.66597    0.20411   3.263 0.001717 **
## MgmtUGCourses           -0.15765    0.08424  -1.871 0.065534 .
## MgmtGCourses             0.28325    0.09981   2.838 0.005961 **
## Total.Workshops         -0.05692    0.01489  -3.823 0.000285 ***
## TotalLangExp            -0.03634    0.01814  -2.004 0.049034 *
## Hardware.Proj.Mgmt.Exp   0.07276    0.04089   1.779 0.079593 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.469669)
##
##     Null deviance: 166.85  on 74  degrees of freedom
## Residual deviance: 147.10  on 69  degrees of freedom
## AIC: 220.48
##
## Number of Fisher Scoring iterations: 13
```

```
e <- residuals(no.lineal2, c ="deviance")
```

$y = MgmtUGCourses + MgmtGCourses + Total.Workshops + TotalLangExp + Hardware.Proj.Mgmt.Exp + \epsilon$

Podemos apreciar que sólo dos coeficientes no son significativos al 95%, sin embargo elegimos este modelo como el mejor. A continuación graficamos los residuales de la deviance (que son más sencillos de interpretar y los cuales son asintóticamente normales por la ley de los grandes números y el TLC.

```
e <- residuals(no.lineal2, c ="deviance")
a <- data.frame(y_hat=fitted(no.lineal2), res.deviance=e)
a$index <- 1:dim(a)[1]
ggplot(a, aes(y_hat, e)) + geom_point(aes(colour=I('purple'))) +
  theme_minimal()
```




```
cor(a$res.deviance, a$y_hat)
```

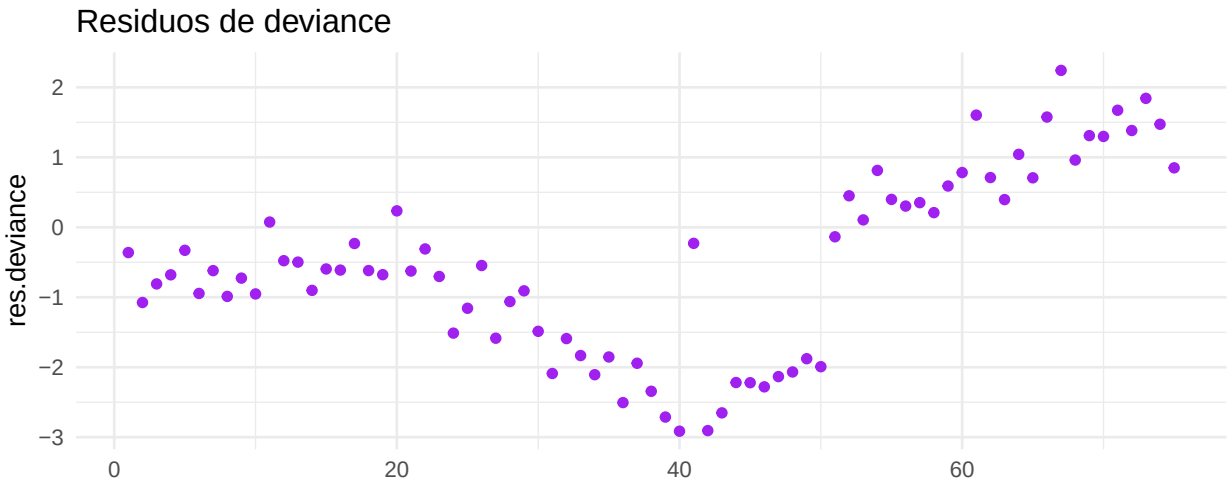
```
## [1] -0.1729037
```

Podemos apreciar que los residuos están poco correlacionados con los valores ajustados, aunque se nota una tendencia en ellos, cuando checamos su normalidad vemos que estos pasan el test de Anderson-Darling de normalidad con una confianza de 95%, en vista de que estos residuos deben de ser normales como lo podemos apreciar en la siguiente gráfica los residuales son normales, el intervalo de confianza se calcula con el estadístico t pertinente en vista de que estimamos la varianza.

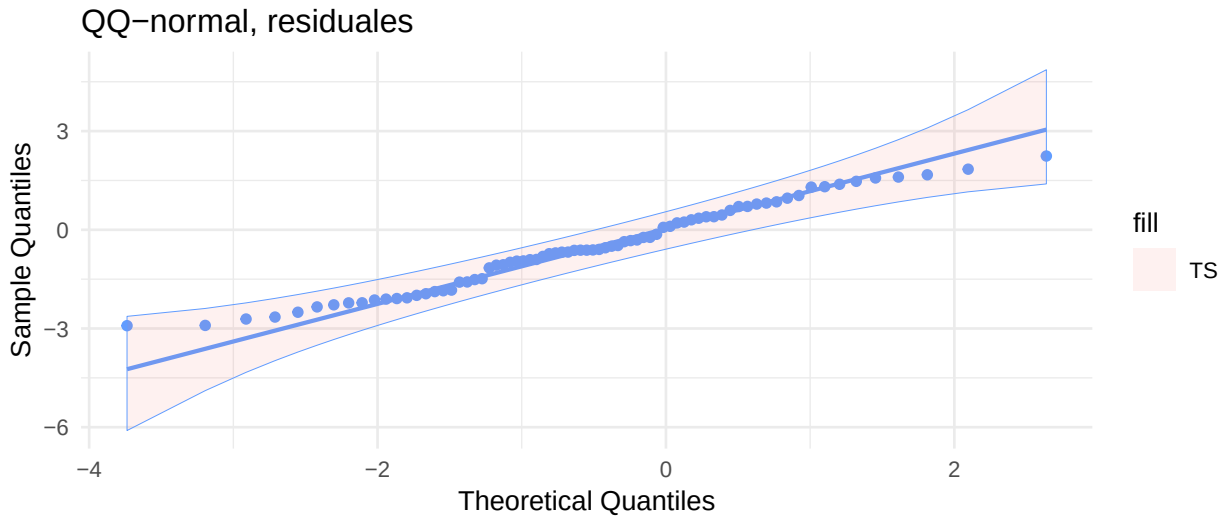
```
library(nortest)
ad.test(e)
```

```
##
##  Anderson-Darling normality test
##
## data:  e
## A = 0.45822, p-value = 0.2569
```

```
ggplot(a, aes(index, res.deviance) )+  geom_point(aes(colour=I('purple'))) +
  theme_minimal() + ggtitle('Residuos de deviance') +xlab('')
```



```r
library(qqplotr)
set.seed(0)
ggplot(data = a, mapping = aes(sample = res.deviance, color = I('#619CFF')) ) +
  stat_qq_line() +  stat_qq_point() +
  geom_qq_band(bandType = "ts", mapping = aes(fill = "TS"), alpha = 0.1) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles")+ theme_minimal() + ggtitle('QQ-normal, resi
```



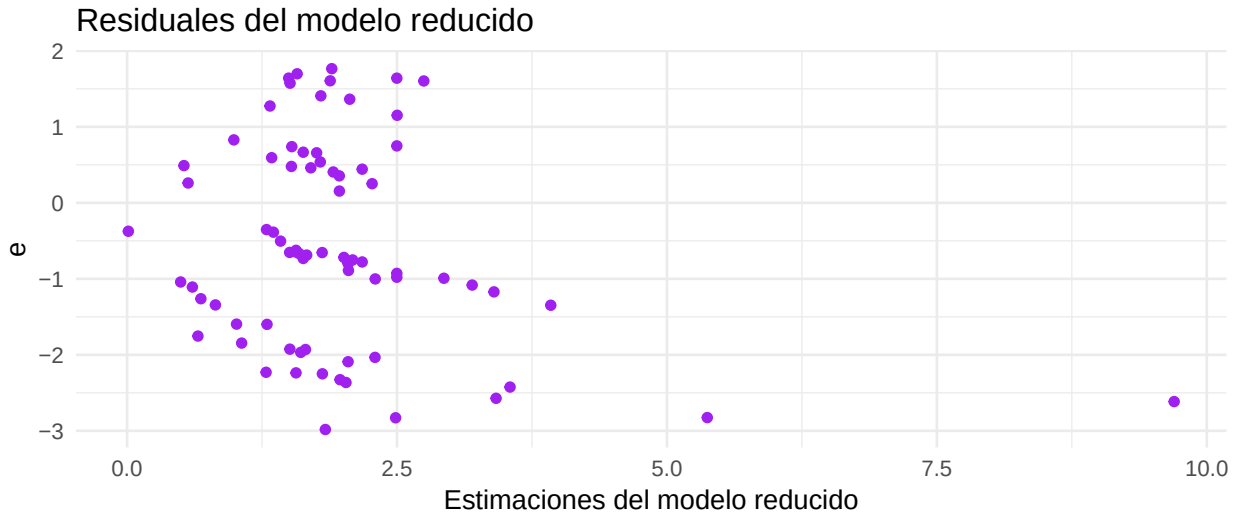
Procedemos a estimar otro modelo semejante al original pero reducido con los coeficientes poco significativos, por parsimonia, y esperando tener mejores residuos de deviance.

```r
#artesanal
no.lineal.reducido <- glm(y ~  MgmtGCourses + Total.Workshops +
                   TotalLangExp , family = Gamma(link=link), data = data )
summary(no.lineal.reducido)
```

```
##
## Call:
## glm(formula = y ~ MgmtGCourses + Total.Workshops + TotalLangExp,
##     family = Gamma(link = link), data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.9824  -1.5974  -0.7179   0.4842   1.7650
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.71639    0.19608   3.654 0.000492 ***
## MgmtGCourses      0.19923    0.06909   2.884 0.005200 **
## Total.Workshops -0.06062    0.01332  -4.550 2.16e-05 ***
## TotalLangExp    -0.03838    0.01771  -2.167 0.033584 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.409896)
##
##     Null deviance: 166.85  on 74  degrees of freedom
## Residual deviance: 153.44  on 71  degrees of freedom
```

```
## AIC: 220.5
##
## Number of Fisher Scoring iterations: 15
```

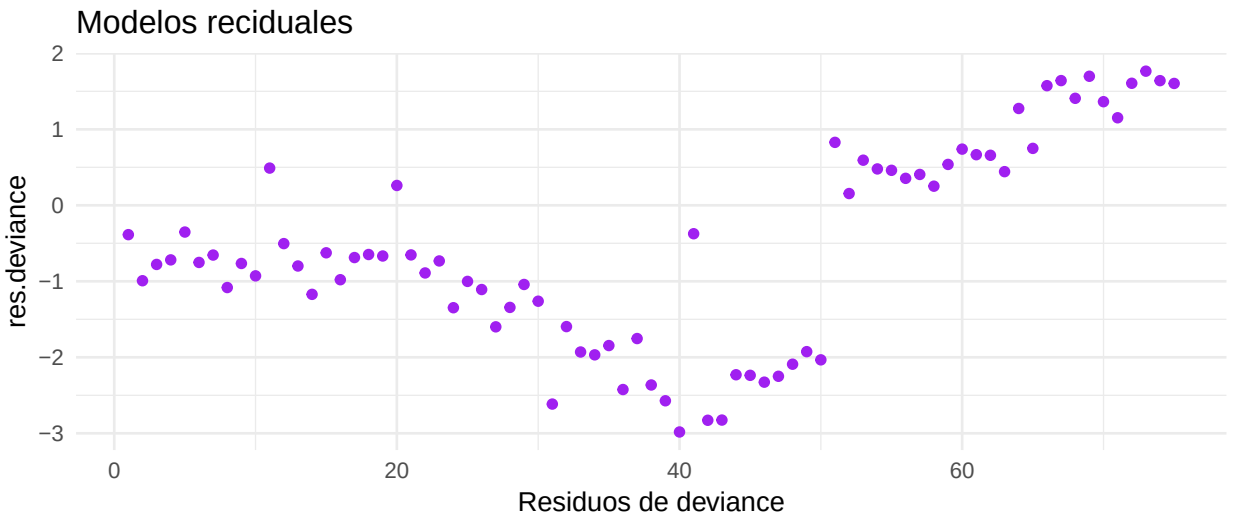
```
e <- residuals(no.lineal.reducido, c ="deviance")
a <- data.frame(y_hat=fitted(no.lineal.reducido), res.deviance=e)
a$index <- 1:dim(a)[1]
ggplot(a, aes(y_hat, e)) + geom_point(aes(colour=I('purple'))) +
  theme_minimal() + ggtitle('Residuales del modelo reducido') +
  xlab('Estimaciones del modelo reducido')
```




```
cor(a$res.deviance, a$y_hat)
```

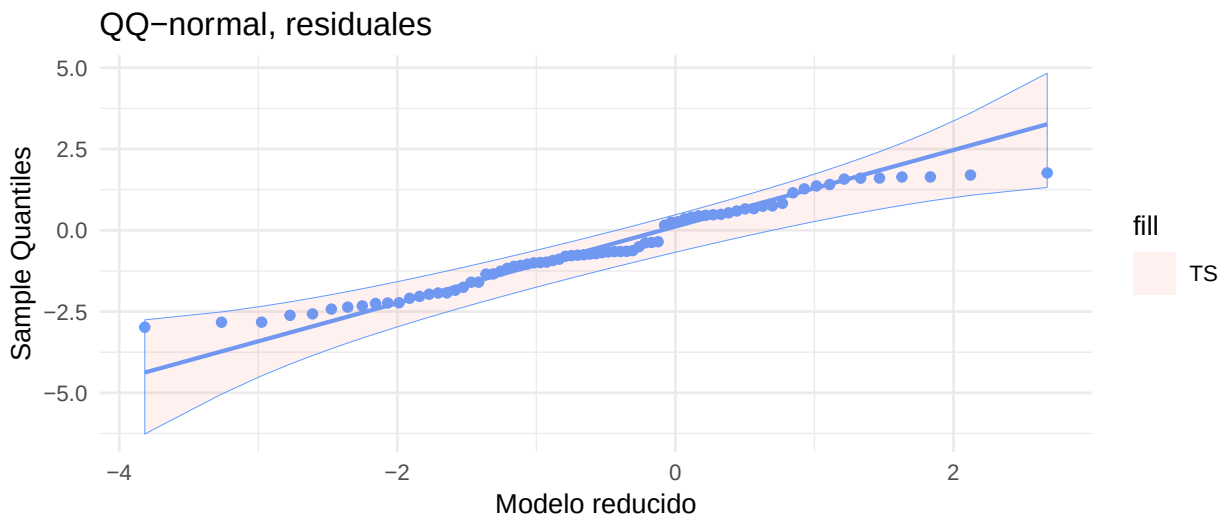
```
## [1] -0.2576133
```

```
ggplot(a, aes(index, res.deviance) )+  geom_point(aes(colour=I('purple'))) +
  theme_minimal() + ggtitle('Modelos reciduales') +xlab('Residuos de deviance')
```




```
set.seed(0)
ggplot(data = a, mapping = aes(sample = res.deviance, color = I('#619CFF')) ) +
  stat_qq_line() +  stat_qq_point() +
```

```
  geom_qq_band(bandType = "ts", mapping = aes(fill = "TS"), alpha = 0.1) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles")+ theme_minimal() + ggtitle('QQ-normal, resi
```




```
ad.test(e)
```

```
##
##  Anderson-Darling normality test
##
## data:  e
## A = 0.79817, p-value = 0.03694
```

Como podemos apreciar, estos residuales del modelo reducido no pasan el test de Anderson-Darling, pese a que el modelo tiene dos parámetros menos.

En los inicios exploramos modelos con diferentes factores de diversas variables, sin embargo no encontramos que alguno de ellos fuese significativo. A continuación realizamos una prueba anova (en realidad es una prueba de deviance no de varianza) con la hipótesis nula de que el modelo reducido ajusta mejor que él no reducido (el primero). La implementación actual de la función anova del ambiente R no arroja un p-value sin embargo como sabemos que el cociente de las variance de ambos modelos son aproximadamente distribuidos como una F asintóticamente podemos realizar el cálculo del cociente de verosimilitudes 147.10/153.44=0.9586809 y contrastarlo contra el quantil 0.9586809 de una distribucion $F_{69,71} = 1.517737$. Como el estadístico calculado es menor al quantil entonces no rechazamos la hipótesis de que los dos parámetros sean no significativos en conjunto. Sin embargo como los residuales del primero sí son normale sy la diferencia de deviances es pequeña optamos por el primer modelo.

```
anova( no.lineal.reducido, no.lineal2)
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ MgmtGCourses + Total.Workshops + TotalLangExp
## Model 2: y ~ MgmtUGCourses + MgmtGCourses + Total.Workshops + TotalLangExp +
##     Hardware.Proj.Mgmt.Exp
##   Resid. Df Resid. Dev Df Deviance
## 1        71     153.44
## 2        69     147.10  2   6.3393
```

```
(147.10/153.44)
```

```
## [1] 0.9586809
```

```r
qf(147.10/153.44, 69,71)
```

```
## [1] 1.517737
```

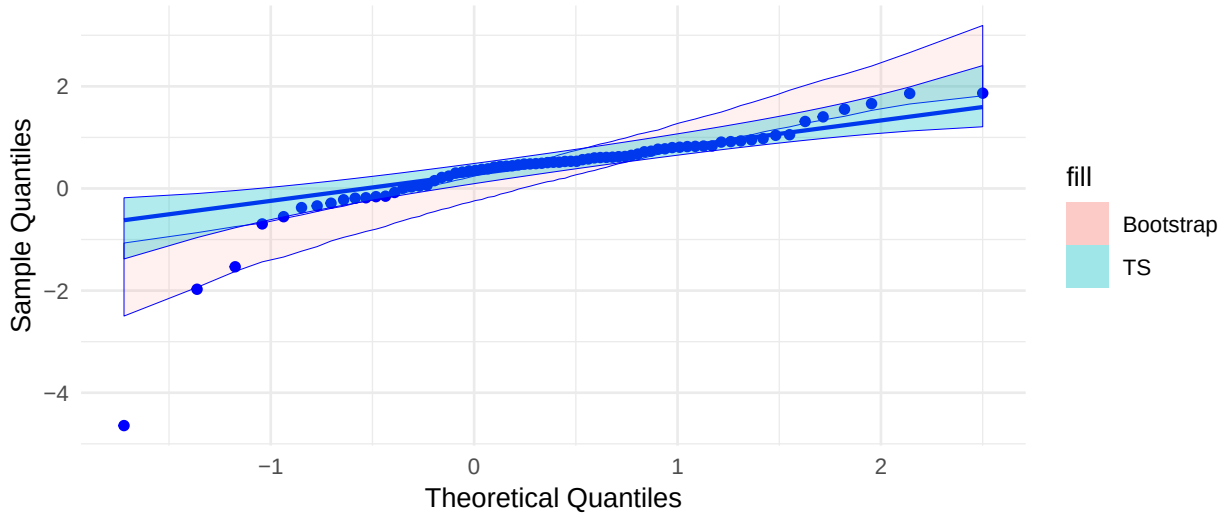
Intervalos de confianza para los parámetros del primer modelo, el modelo que determinamos como mejor.

```r
confint(no.lineal2)
```

```
##                                 2.5 %       97.5 %
## (Intercept)              0.2914957950  1.063883858
## MgmtUGCourses           -0.3231872511  0.025951259
## MgmtGCourses             0.0454740252  0.499452708
## Total.Workshops         -0.0820572749 -0.007839038
## TotalLangExp            -0.0807707800  0.013240102
## Hardware.Proj.Mgmt.Exp -0.0004028774  0.165252833
```

A continuación presentamos los intervalos de confianza (que se construyen con un cuantil con distribución gamma dado que los residuos de deviance son aproximadamente normales asintóticamente) para la respuesta media de nuestras observaciones al igual que los de bootstrap.

```r
preds <- predict(no.lineal, data,  se.fit = TRUE) # para obtener los errores estandar
critval <- qt(.95,71) ## los df los saque del anova
upr <- preds$fit + (critval * preds$se.fit)
lwr <- preds$fit - (critval * preds$se.fit)
fit <- preds$fit
CI.mean <- data.frame(l=lwr, mean=fit, u=upr)
##################
set.seed(0)
smp <- data.frame(norm = CI.mean$mean)
ggplot(data = smp, mapping = aes(sample = norm, colour=I('blue'))) +
  geom_qq_band(bandType = "boot", mapping = aes(fill = "Bootstrap"),
               alpha = 0.1) + stat_qq_line() +   stat_qq_point() +
  geom_qq_band(bandType = "ts", mapping = aes(fill = "TS"), alpha = 0.3) +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles")+ theme_minimal()
```



3. *Derive el algoritmo EM-soft para su distribución elegida en el inciso 1. Deducir ambos pasos E y M claramente.*

Bickel, Peter J., and Kjell A. Doksum. 2001. *Mathematical Statistics: Basic Ideas and Selected Topics*. 2nd ed. Vol. 1. Prentice-Hall.

Myers, Raymond H., Douglas C. Montgomery, G. Geoffrey Vining, and Timothy J. Robinson. 2012. *Generalized Linear Models: With Applications in Engineering and the Sciences: Second Edition*. John Wiley; Sons Inc. doi:10.1002/9780470556986.