Modelos Lineales Generalizados

Fernando Anorve

3/11/2020

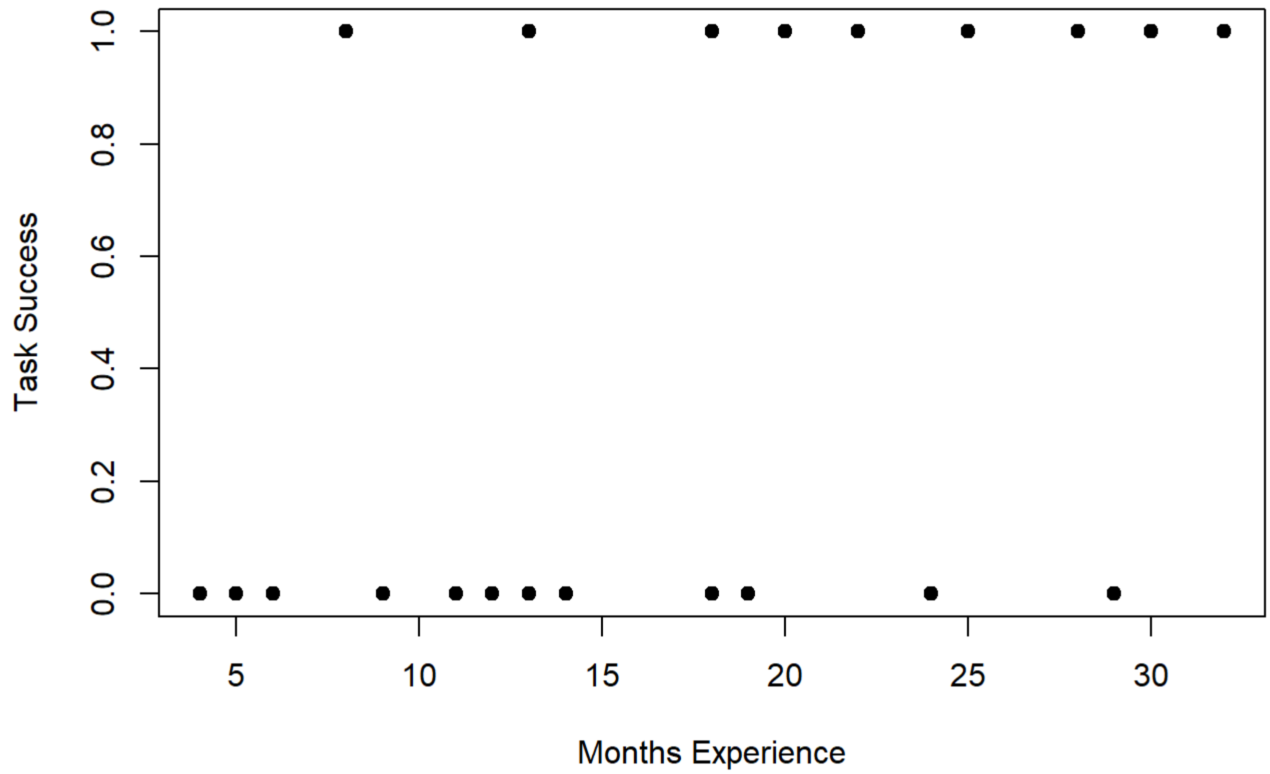
Para ajustar un modelo lineal generalizado, usaremos la función glm. Funciona de forma muy similar a la función lm, a diferencia de que indicamos el parámetro de familia.

Family	Link
Gaussian	identity
binomial	logit,probit o cloglog
Poisson	log,identity o sqrt
gamma	inverse,identity, o log
inverse.gaussian	"1/mu^2"

Nota: cuando se usa la primera familia (Gaussian) no se necesita especificar el link

Ejemplo 1 - Logística

```
dat=read.table("CompAnalyst.txt",header=T)
names(dat)[1] <- "MonthsExperience"

plot(dat,xlab="Months Experience",ylab="Task Success", pch=20,cex=1.5)
```



Se puede observar una relación entre el éxito en una tarea con la experiencia en meses.

```
fit = glm(TaskSuccess ~ MonthsExperience, data=dat, family=binomial(link="logit"))
summary(fit)
```

```
##
## Call:
## glm(formula = TaskSuccess ~ MonthsExperience, family = binomial(link = "logit"),
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8992  -0.7509  -0.4140   0.7992   1.9624
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.05970    1.25935  -2.430   0.0151 *
## MonthsExperience  0.16149    0.06498   2.485   0.0129 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 34.296  on 24  degrees of freedom
## Residual deviance: 25.425  on 23  degrees of freedom
## AIC: 29.425
##
## Number of Fisher Scoring iterations: 4
```

$$\ln\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

Se pueden calcular los intervalos de confianza de β_0 y β_1

```
confint(fit)
```

```
## Waiting for profiling to be done...
```

```
##                        2.5 %     97.5 %
## (Intercept)      -6.03725238 -0.9160349
## MonthsExperience  0.05002505  0.3140397
```

¿Cómo se interpreta β_1?

Cuando x aumenta una unidad, suponiendo que todo lo demás es constante (en este caso no hay más covariables, duh) las log-posibilidades (log-odds) aumentan en promedio β_1

$$\frac{p(x)}{1-p(x)} = \exp(\beta_0 + \beta_1 x) = \exp(\beta_0) \cdot \exp(\beta_1 x)$$

$$\frac{p(x)}{1-p(x+1)} = \exp(\beta_0) \cdot \exp(\beta_1 x + \beta_1) = \exp(\beta_0) \cdot \exp(\beta_1 x) \cdot \exp(\beta_1)$$

Cuando x aumenta una unidad, las posibilidades (odds) se multiplican por un factor $\exp(\beta_1)$. Para ello podemos hallar los intervalos de confianza en escala exponencial.

```r
exp(coef(fit))
```

```
##      (Intercept) MonthsExperience
##       0.04690196       1.17525591
```

```r
exp(confint(fit))
```

```
## Waiting for profiling to be done...
```

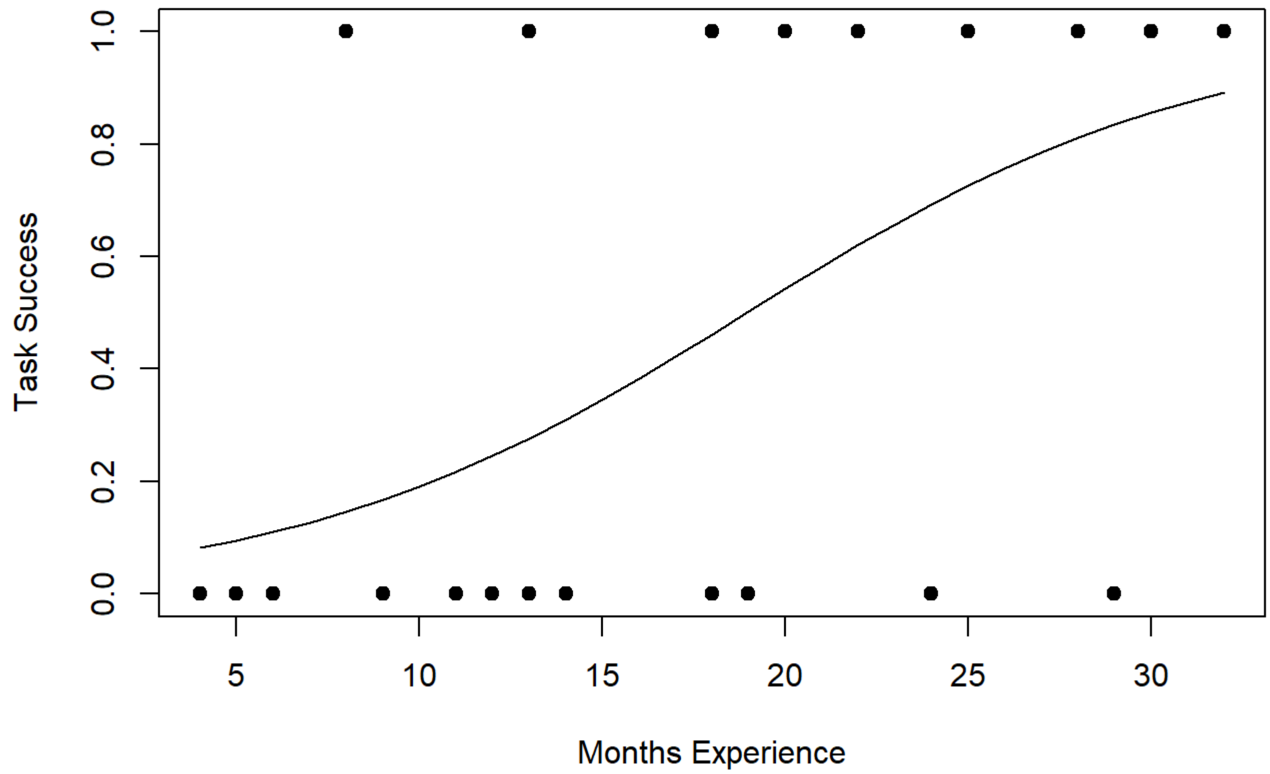
```
##                        2.5 %    97.5 %
## (Intercept)      0.002388112 0.4001024
## MonthsExperience 1.051297434 1.3689441
```

¿Cómo hallar (y graficar) $\hat{p}(x)$? Primero hallamos las predicciones

```r
# Los valores de x a usar
predMonths=seq(4,32,by=1)

# Usamos la función para predecir predict.glm, que predice el valor de logit
logit.hat=predict.glm(fit,newdata=data.frame(MonthsExperience=predMonths),se.fit=TRUE)

# hay que sacar el inverso de logit!
prob.hat=exp(logit.hat$fit)/(1+exp(logit.hat$fit))
```

Y luego las graficamos!

```r
plot(dat,xlab="Months Experience",ylab="Task Success", pch=20,cex=1.5,xlim=c(4,32))
lines(predMonths,prob.hat)
```

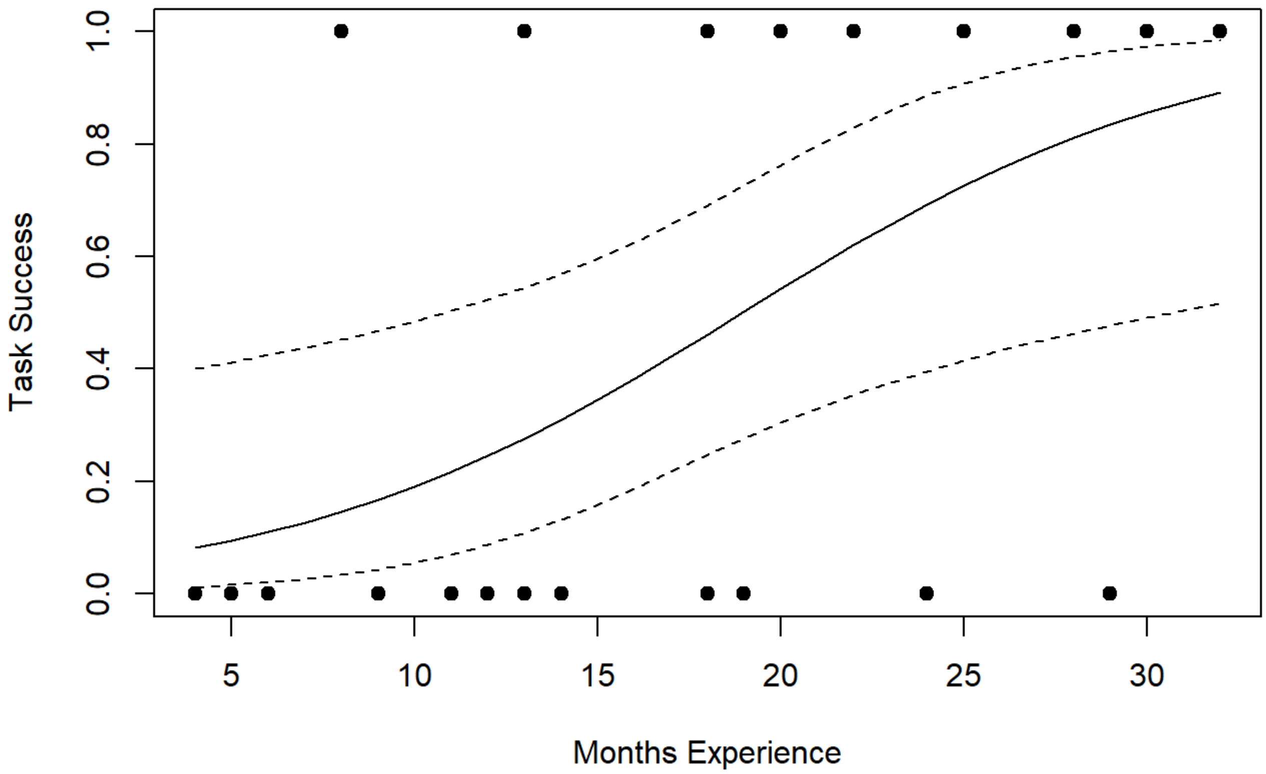


También podemos graficar los intervalos de confianza. Como en el caso anterior. Primero calculamos, y luego graficamos

```
prob.lwr=exp(logit.hat$fit-1.96*logit.hat$se.fit)/(1+exp(logit.hat$fit-1.96*logit.hat$se.fit))
prob.upr=exp(logit.hat$fit+1.96*logit.hat$se.fit)/(1+exp(logit.hat$fit+1.96*logit.hat$se.fit))
```

Notad que los intervalos se calcularon multiplicando los errores estándar por valores críticos de distribución normal.

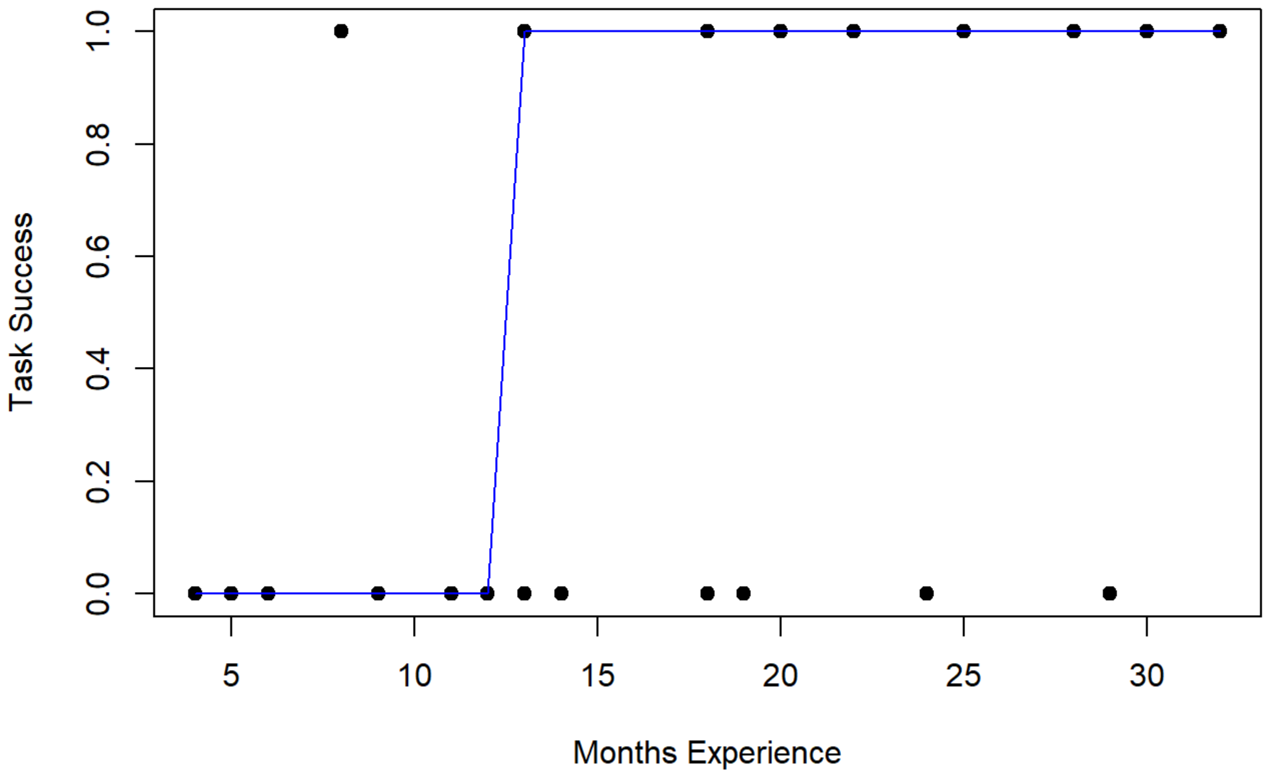
```
plot(dat,xlab="Months Experience",ylab="Task Success", pch=20,cex=1.5,xlim=c(4,32))
lines(predMonths,prob.hat)
lines(predMonths,prob.lwr,lty=2)
lines(predMonths,prob.upr,lty=2)
```



Podemos hacer predicciones más puntuales usando un umbral:

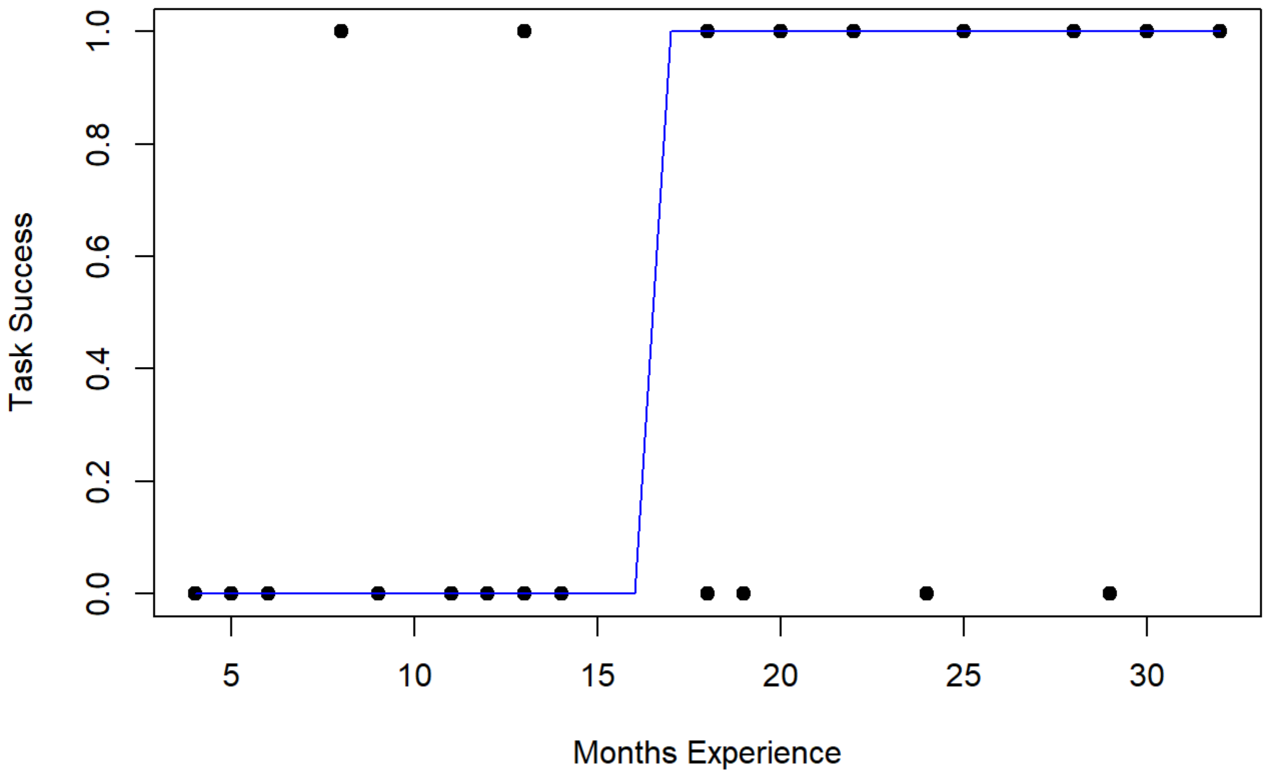
```
for (umbral in c(0.25, 0.40, 0.5 , 0.75)) {
    print(paste0("umbral: ",umbral))
    prob.hat.point = ifelse(prob.hat < umbral,0,1)

    plot(dat,xlab="Months Experience",ylab="Task Success", pch=20,cex=1.5,xlim=c(4,32))
    lines(predMonths,prob.hat.point, col="blue")
}
```

```
## [1] "umbral: 0.25"
```



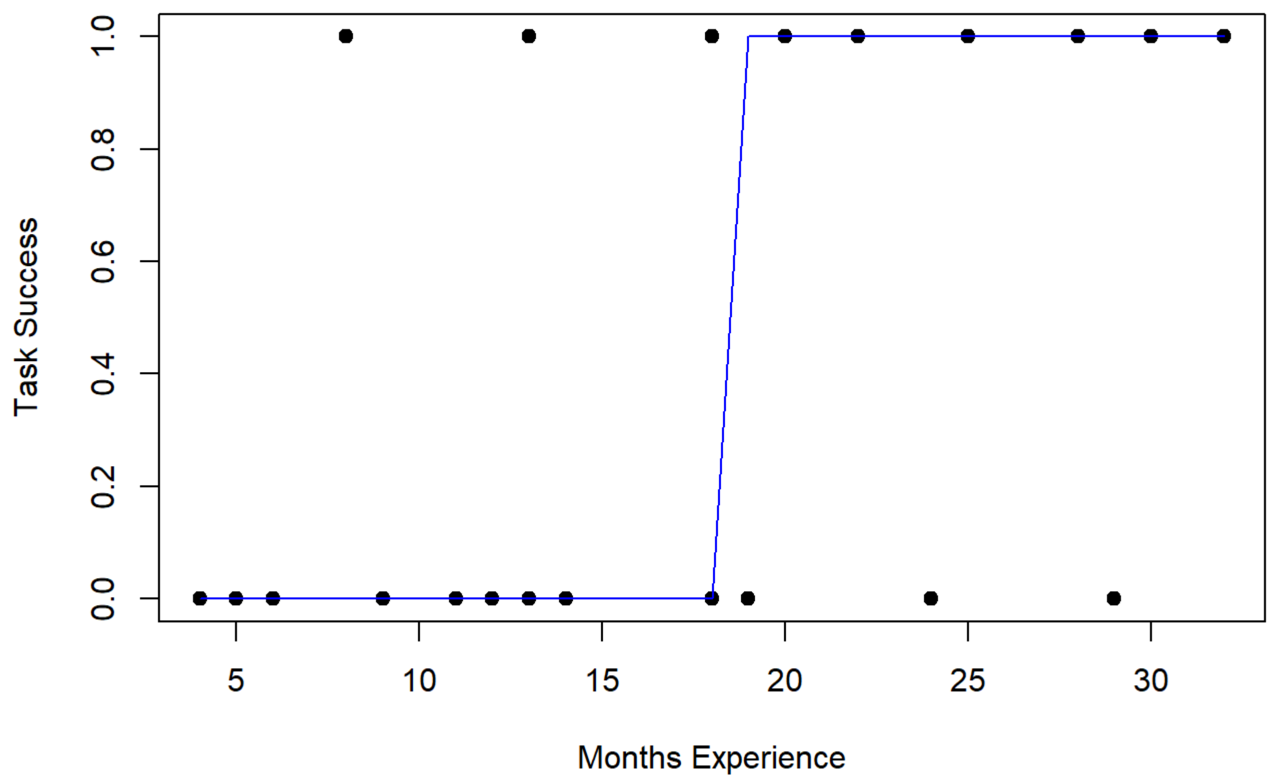
Task Success
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
30
Months Experience


```
## [1] "umbral: 0.4"
```



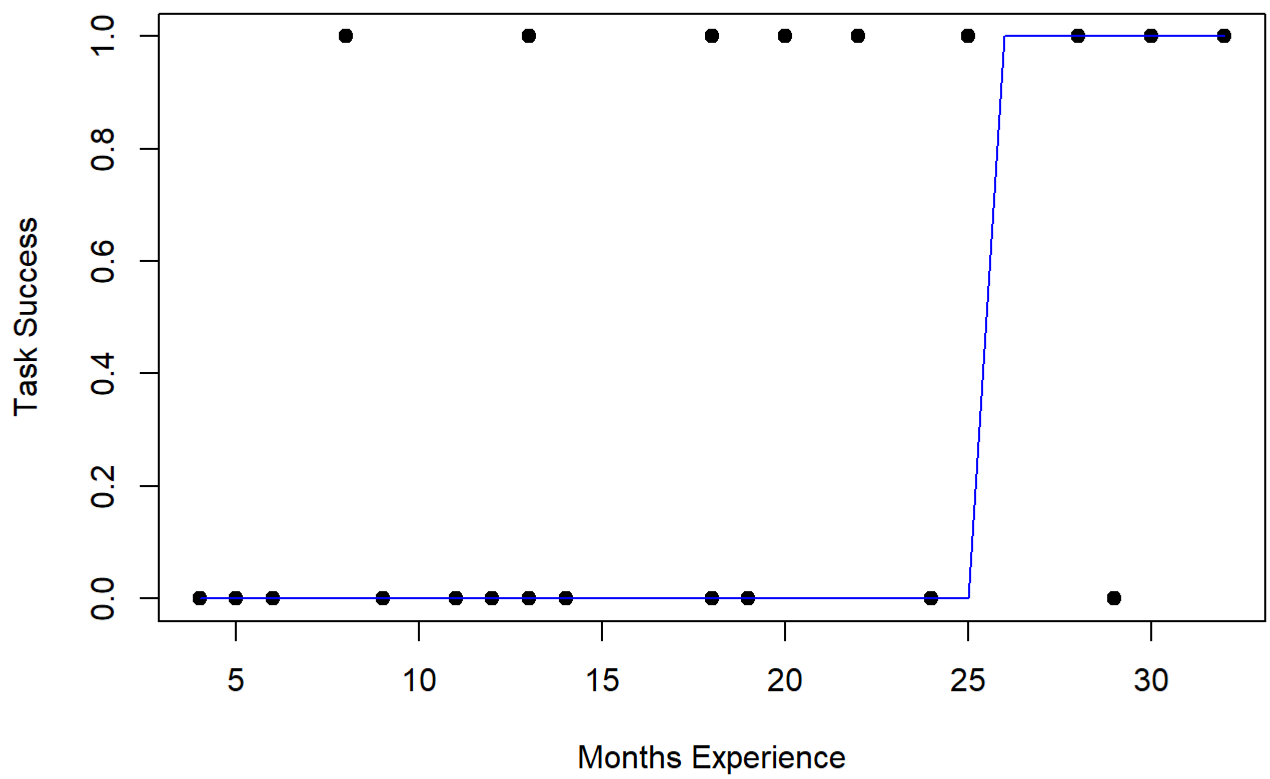
Task Success
0.0
0.2
0.4
0.6
0.8
1.0
5
10
15
20
25
30
Months Experience


```
## [1] "umbral: 0.5"
```



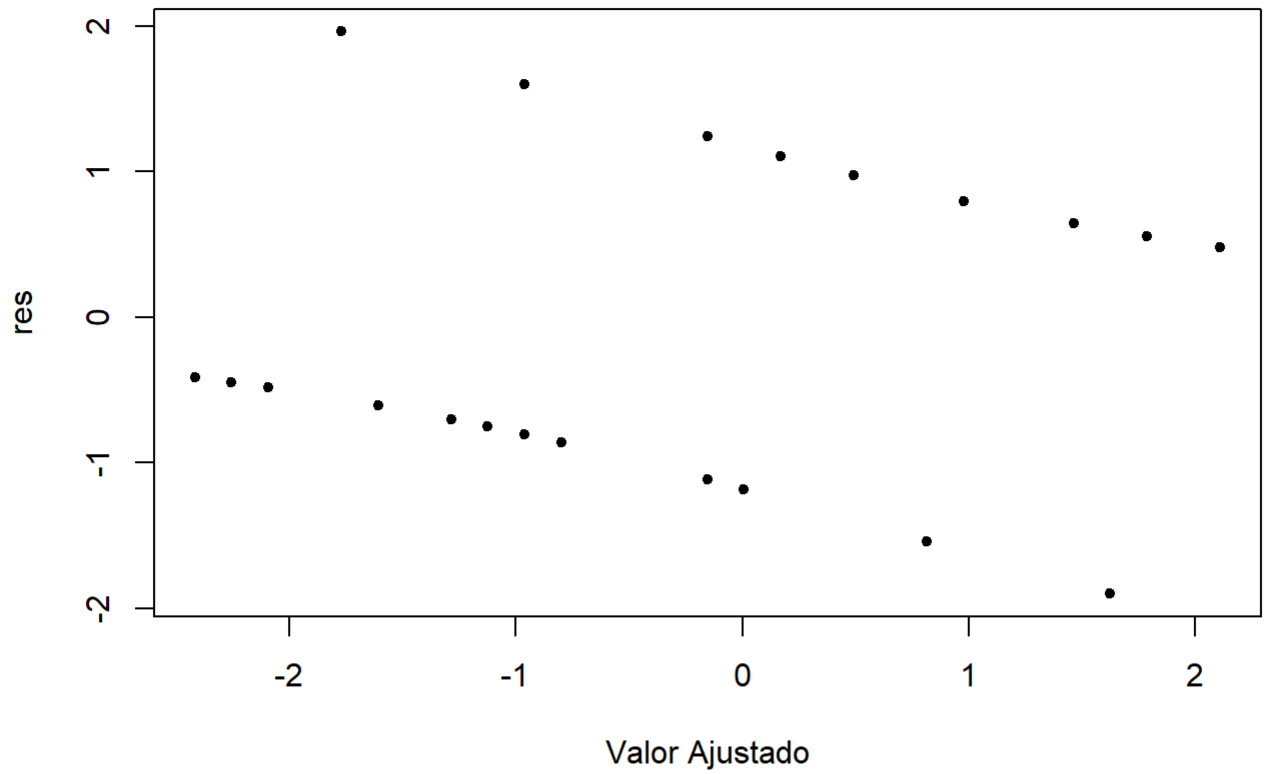
1.0
0.8
0.6
0.4
0.2
0.0
Task Success
5
10
15
20
25
30
Months Experience


```
## [1] "umbral: 0.75"
```

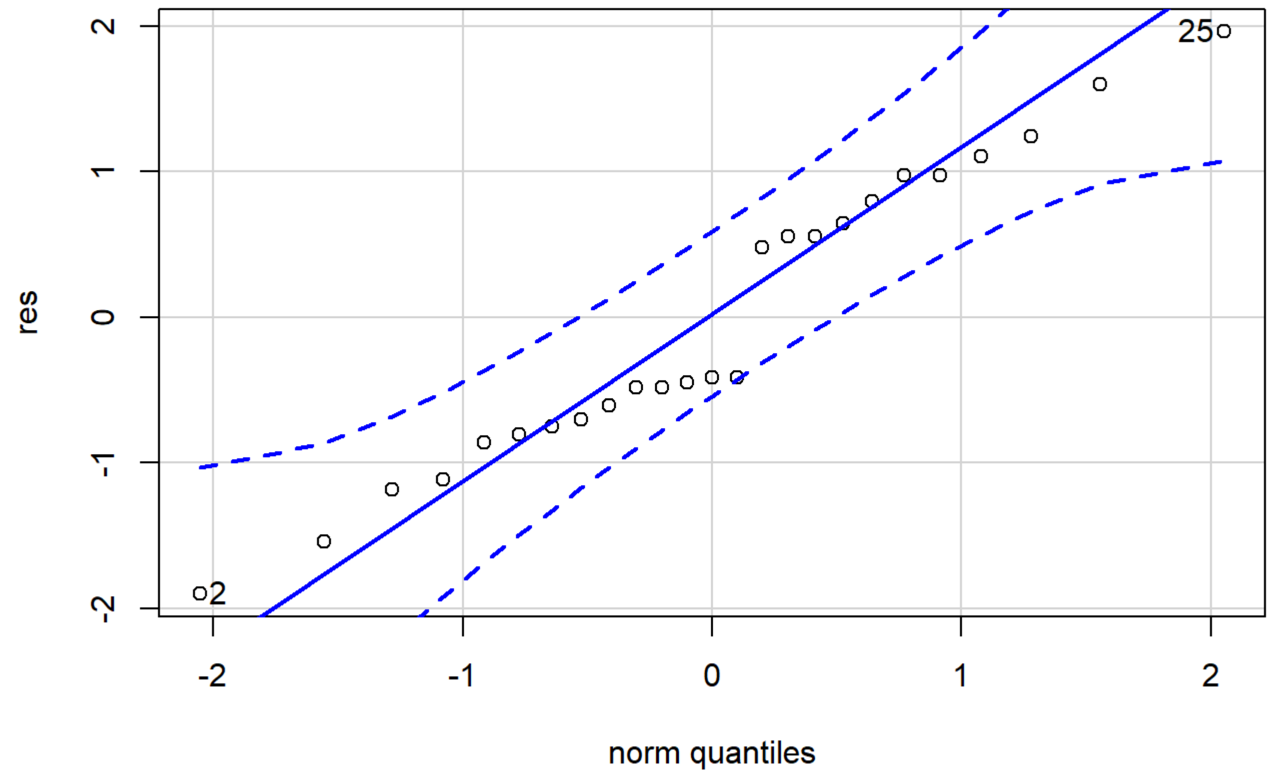


¿Qué nos dicen los residuales?

```
library(car)
res=residuals(fit)

plot((predict(fit)),res,pch=20,xlab = "Valor Ajustado")
```




```
qqPlot(res)
```




```
## [1] 25  2
```

En realidad, cuando hay sólo una observación por valor de covariable los residuales pueden ser engañosos.

Además, tampoco me fiaría mucho del qqplot, puesto que la normalidad sólo se alcanza de manera asintótica

¿Cómo usar R para comparar modelos anidados?

Modelo simple vs. modelo “completo”: Al usar la tabla de anova nos muestra la diferencia entre devianzas

```
fit0=glm(TaskSuccess~1,data=dat,family=binomial(link="logit"))
anova(fit0,fit)
```

```
## Analysis of Deviance Table
##
## Model 1: TaskSuccess ~ 1
## Model 2: TaskSuccess ~ MonthsExperience
##   Resid. Df Resid. Dev Df Deviance
## 1        24     34.296
## 2        23     25.425  1   8.8719
```

No obstante, también podemos hacer una prueba de chi cuadrada para comparar!

```
anova(fit0,fit,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: TaskSuccess ~ 1
## Model 2: TaskSuccess ~ MonthsExperience
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        24     34.296
## 2        23     25.425  1   8.8719 0.002896 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

También está esta otra forma:

```
anova(fit,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: TaskSuccess
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                               24     34.296
## MonthsExperience  1   8.8719        23     25.425 0.002896 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

El modelo naive está “lejos” del modelo completo, con una diferencia de 8.87 en devianza. La prueba de chi cuadrada sugiere que esta diferencia es significativa, por lo que nuestro modelo parece ser útil!

Ejemplo 2 - Logística Múltiple

En el siguiente ejemplo, se intenta relacionar dos covariables cuantitativas (edad y años de experiencia) con una respuesta binaria que indica la terminación o la permanencia de diversos empleados en una empresa

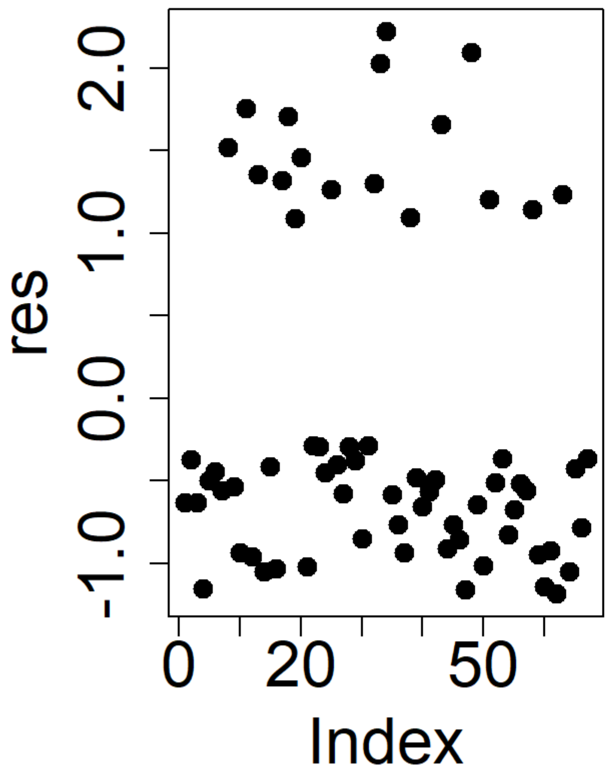
```
dat=read.csv("engall.csv",header=T)
fit = glm(Terminated ~ Age + YrofService, data = dat, family = binomial(link="logit"))
summary(fit)
```

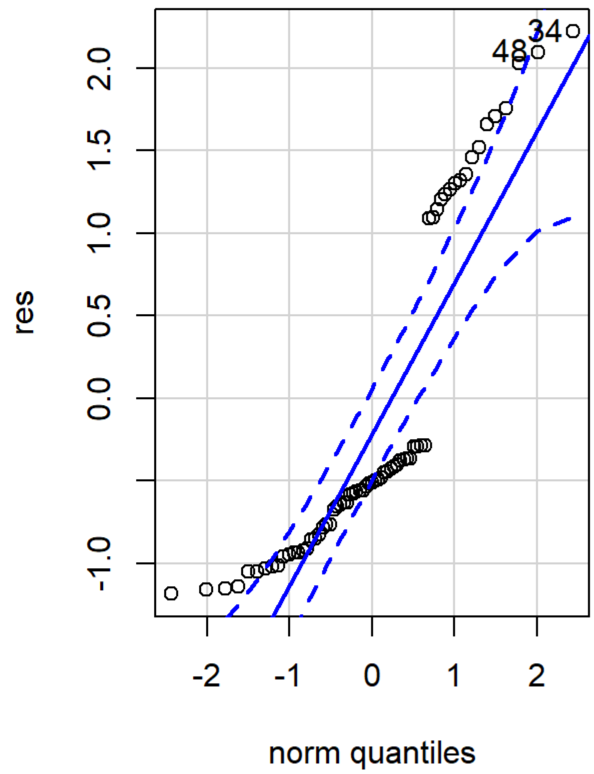
```
##
## Call:
## glm(formula = Terminated ~ Age + YrofService, family = binomial(link = "logit"),
##     data = dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.1834  -0.8359  -0.5112   0.4029   2.2201
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.03750    1.68490  -2.990  0.00279 **
## Age          0.08573    0.04117   2.082  0.03733 *
## YrofService -0.01147    0.04497  -0.255  0.79862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 75.897  on 66  degrees of freedom
## Residual deviance: 67.014  on 64  degrees of freedom
## AIC: 73.014
##
## Number of Fisher Scoring iterations: 4
```

A continuación se presenta la gráfica de residuales. A comparación del caso lineal, no es tan trivial usar las gráficas resultantes para evaluar el ajuste del modelo logístico. De hecho, cuando hay pocas respuestas (como en este caso donde Y = 0,1), los gráficos de residuos son poco informativos.

Por lo tanto: no hay de que preocuparse (por ahora)

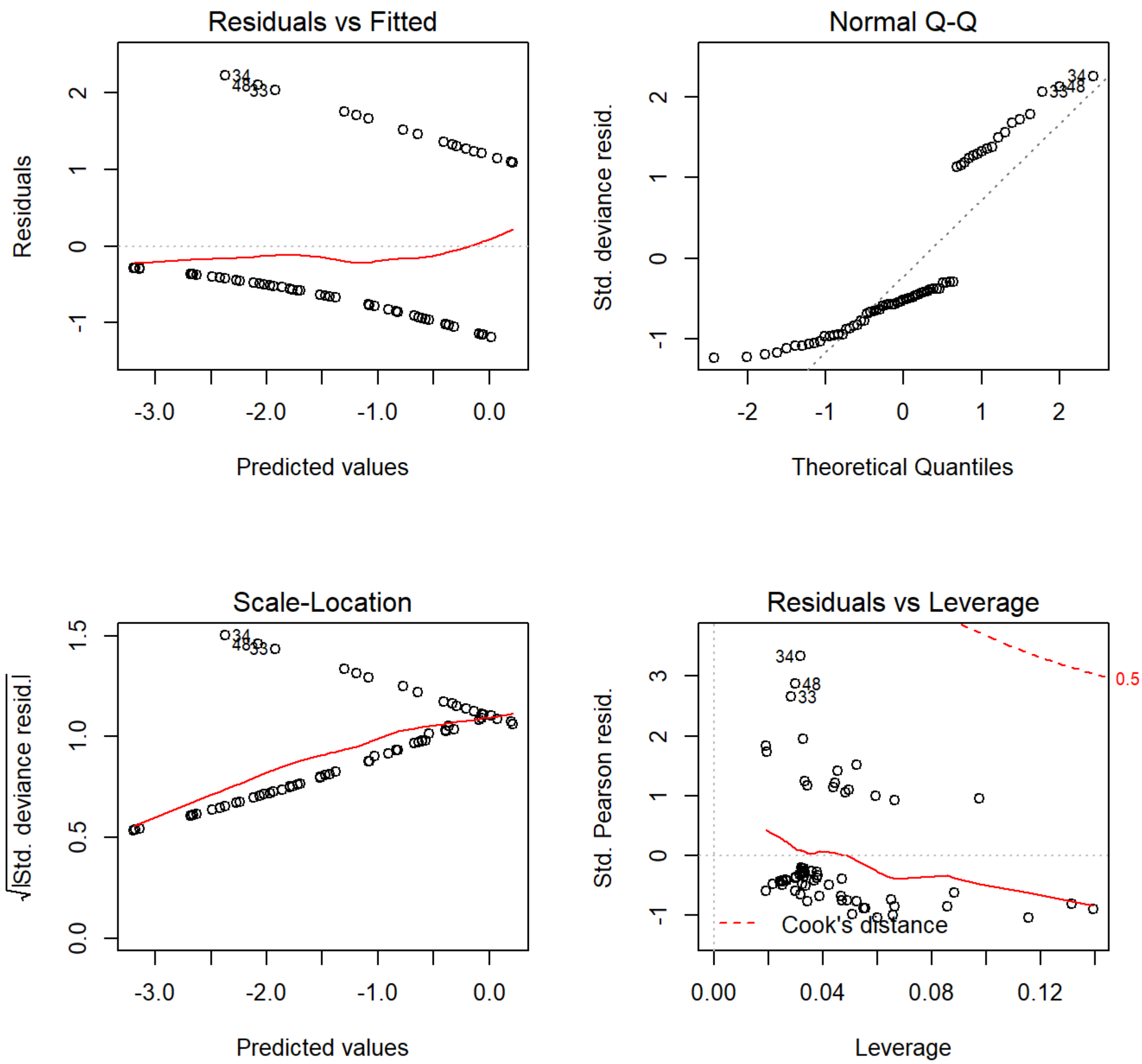
```
res=residuals(fit)
par(mfrow=c(1,2))
plot(res,pch=20,cex=2,cex.lab=2,cex.axis=2)
qqPlot(res)
```



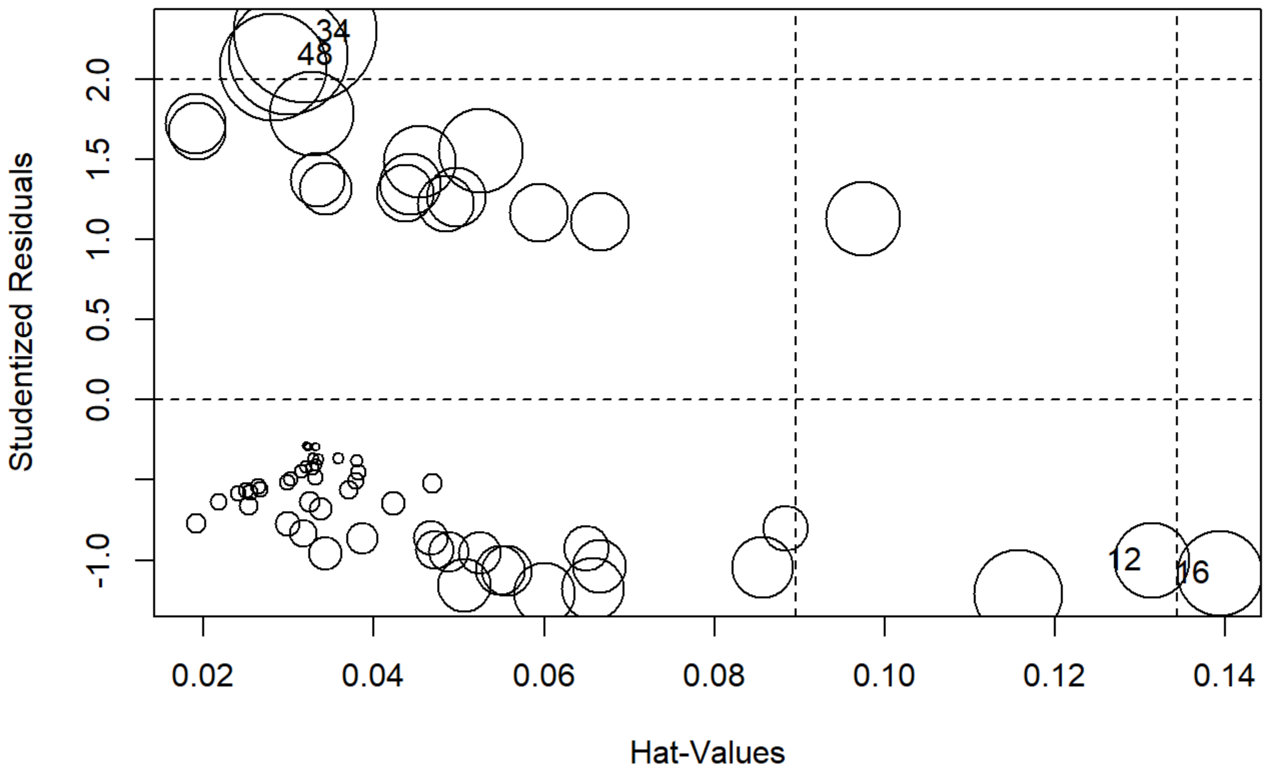



```
## [1] 34 48
```

```
par(mfrow=c(2,2))
plot(fit)
```




```
influencePlot(fit)
```




```
##      StudRes        Hat      CookD
## 12 -1.002422 0.13136802 0.03375442
## 16 -1.081492 0.13934306 0.04366131
## 34  2.298670 0.03196043 0.12228233
## 48  2.154182 0.02991316 0.08473716
```

Haciendo una prueba:

```
anova(fit,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Terminated
##
## Terms added sequentially (first to last)
##
##
##             Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                          66     75.897
## Age          1   8.8181        65     67.079 0.002982 **
## YrofService  1   0.0649        64     67.014 0.798875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ejemplo 3 - Logística Múltiple

El siguiente conjunto de datos fue extraido del American Community Survey de 2010 en el estado de Nueva York, del cual se obtuvo un subconjunto de 22,745 filas y 18 columnas.

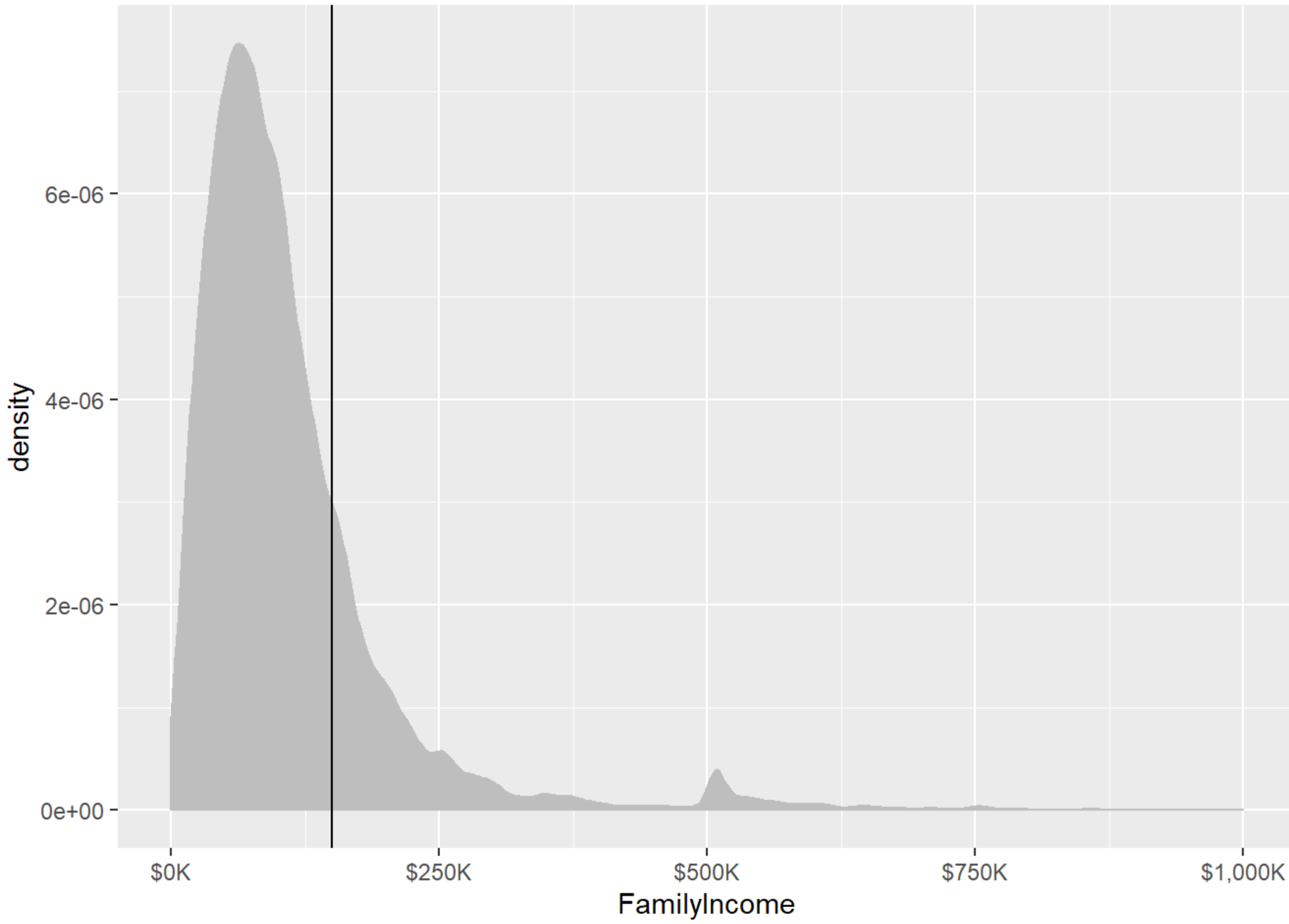
De este conjunto de datos se puede uno preguntar si un hogar tiene ingresos mayores que $150,000

```
acs <- read.table("http://jaredlander.com/data/acs_ny.csv", sep=",", header=TRUE, stringsAsFacto
rs=FALSE)
acs$Income <- with(acs, FamilyIncome >= 150000)
# head(acs)
```

Podemos hacer una gráfica para ver el ingreso de cada familia:

```
library(ggplot2)
library(useful) # para el label de multiple.dollar
 ggplot(acs, aes(x=FamilyIncome)) +
  geom_density(fill="grey", color="grey") +
  geom_vline(xintercept=150000)  +
  scale_x_continuous(label=multiple.dollar, limits=c(0, 1000000))
```

```
## Warning: Removed 13 rows containing non-finite values (stat_density).
```

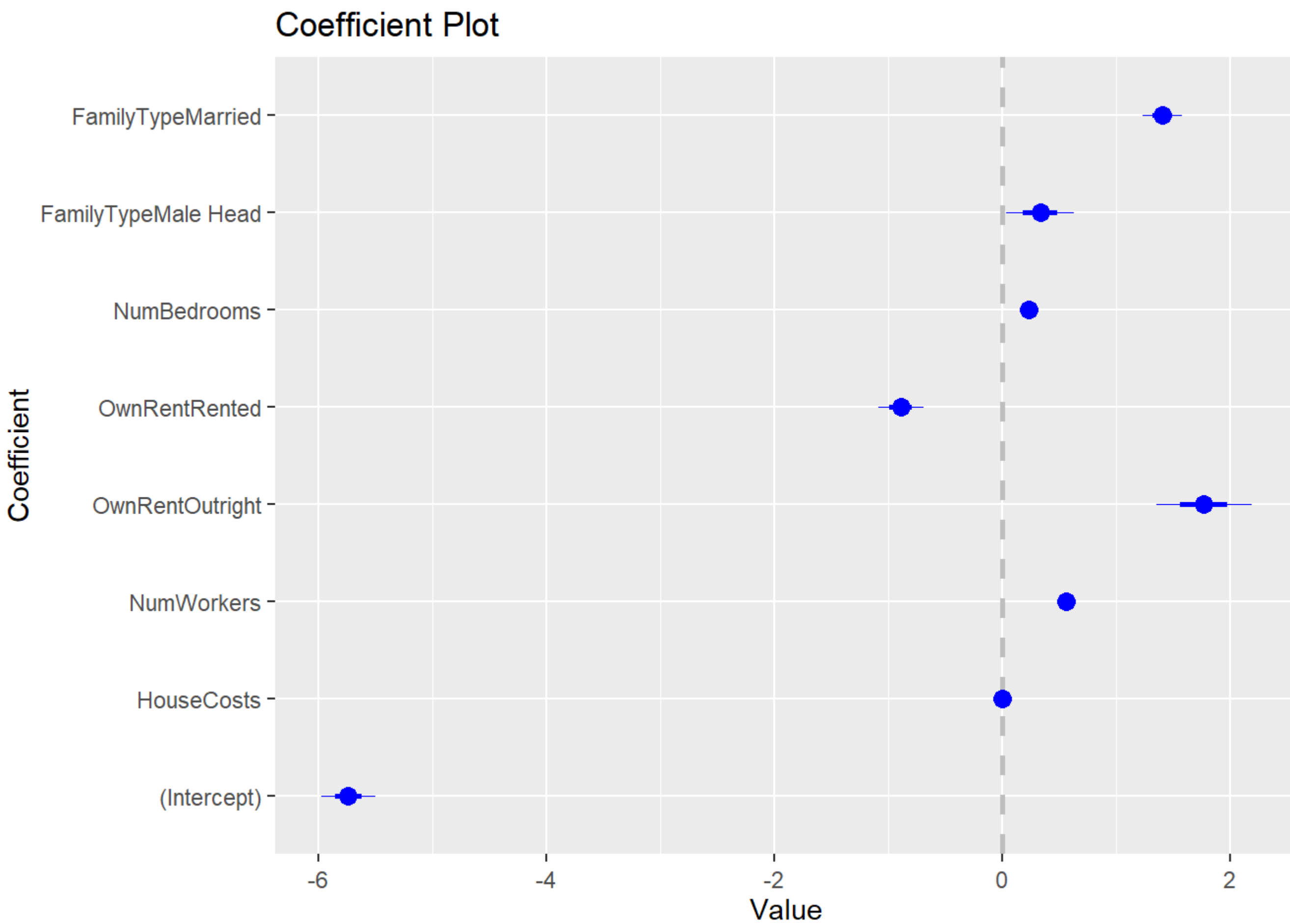



```
income1 <- glm(Income ~ HouseCosts + NumWorkers + OwnRent +
                 NumBedrooms + FamilyType, data=acs,
               family=binomial(link="logit"))
summary(income1)
```

```
##
## Call:
## glm(formula = Income ~ HouseCosts + NumWorkers + OwnRent + NumBedrooms +
##     FamilyType, family = binomial(link = "logit"), data = acs)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8452  -0.6246  -0.4231  -0.1743   2.9503
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -5.738e+00  1.185e-01 -48.421   <2e-16 ***
## HouseCosts           7.398e-04  1.724e-05  42.908   <2e-16 ***
## NumWorkers           5.611e-01  2.588e-02  21.684   <2e-16 ***
## OwnRentOutright      1.772e+00  2.075e-01   8.541   <2e-16 ***
## OwnRentRented       -8.886e-01  1.002e-01  -8.872   <2e-16 ***
## NumBedrooms          2.339e-01  1.683e-02  13.895   <2e-16 ***
## FamilyTypeMale Head  3.336e-01  1.472e-01   2.266   0.0235 *
## FamilyTypeMarried    1.405e+00  8.704e-02  16.143   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 22808  on 22744  degrees of freedom
## Residual deviance: 18073  on 22737  degrees of freedom
## AIC: 18089
##
## Number of Fisher Scoring iterations: 6
```

Gráfica de los coeficientes

```
library(coefplot)
coefplot(income1)
```



ANOVA

```
anova(income1,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Income
##
## Terms added sequentially (first to last)
##
##
##             Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                        22744      22808
## HouseCosts   1  3134.20     22743      19673 < 2.2e-16 ***
## NumWorkers   1   771.16     22742      18902 < 2.2e-16 ***
## OwnRent      2   234.99     22740      18667 < 2.2e-16 ***
## NumBedrooms  1   171.10     22739      18496 < 2.2e-16 ***
## FamilyType   2   422.83     22737      18073 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notad la aclaración de que los términos fueron agregados secuencialmente (i.e. ¡el orden importa!)

```
income2 <- glm(Income ~ FamilyType + OwnRent + HouseCosts +
                 NumBedrooms + NumWorkers, data=acs,
               family=binomial(link="logit"))
anova(income2,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Income
##
## Terms added sequentially (first to last)
##
##
##             Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                        22744      22808
## FamilyType   2   866.18     22742      21941 < 2.2e-16 ***
## OwnRent      2   325.02     22740      21616 < 2.2e-16 ***
## HouseCosts   1  2777.76     22739      18839 < 2.2e-16 ***
## NumBedrooms  1   262.12     22738      18577 < 2.2e-16 ***
## NumWorkers   1   503.19     22737      18073 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

¿Cómo predecir?

Por simplicidad, tomemos cinco observaciones del conjunto de datos

```
set.seed(2020)
x = sample(22745,5)
new_data = acs[x,c("HouseCosts" ,"NumWorkers","OwnRent",
                 "NumBedrooms","FamilyType")]

logit.hat=predict.glm(income1,newdata=new_data,se.fit=TRUE)
new_data$Income = acs$Income[x]
new_data$logit = logit.hat$fit
```

Tenemos los valores ajustados en la escala logit, lo convertimos a probabilidad estimada

```
prob.hat=exp(logit.hat$fit)/(1+exp(logit.hat$fit))
new_data$proba = prob.hat

new_data
```

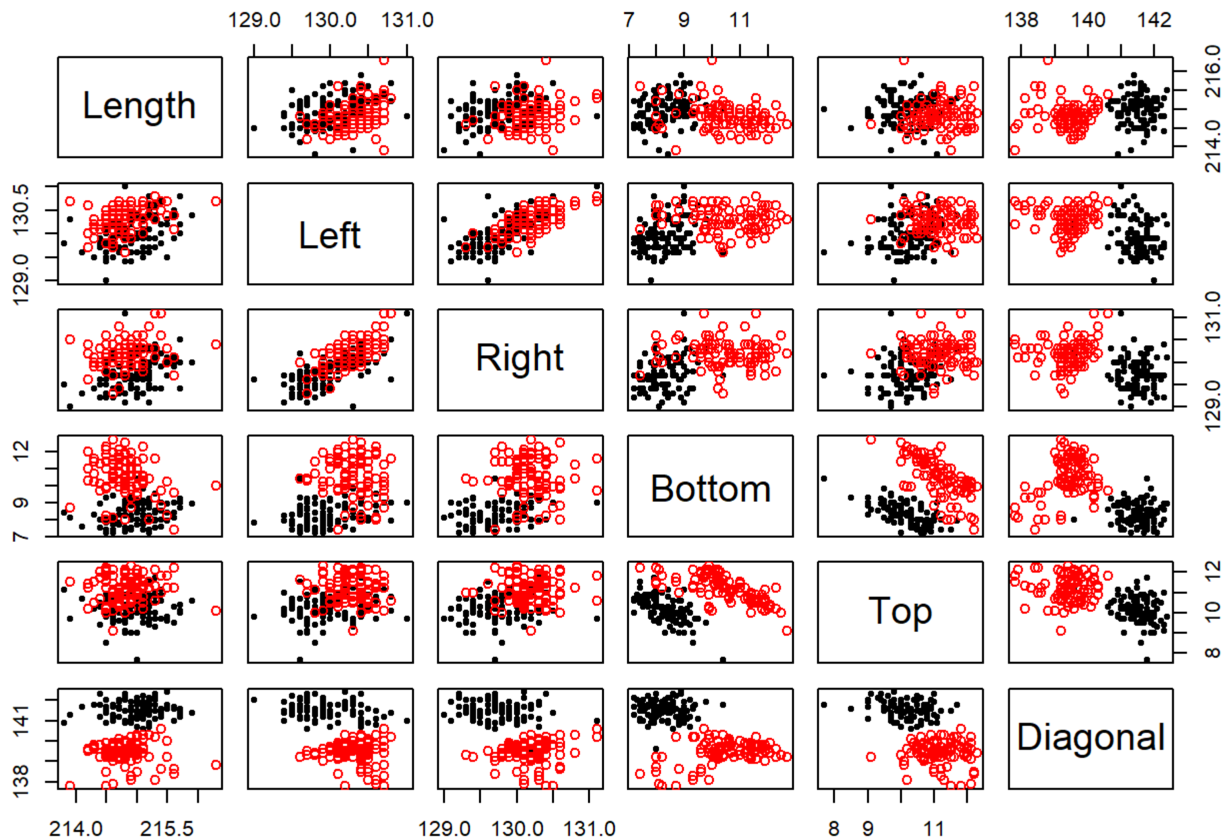
```
##      HouseCosts NumWorkers  OwnRent NumBedrooms  FamilyType Income
## 9628        650          1 Mortgage           3     Married  FALSE
## 7767         60          0 Outright           4 Female Head  FALSE
## 8920        700          2 Mortgage           3     Married  FALSE
## 4417       4800          1 Mortgage           3     Married   TRUE
## 8465       1200          1   Rented           3     Married  FALSE
##           logit      proba
## 9628 -2.5892019 0.06983661
## 7767 -2.9855688 0.04808210
## 8920 -1.9911335 0.12013700
## 4417  0.4809649 0.61797569
## 8465 -3.0708647 0.04432518
```

Ejemplo 4 - Dato curioso

Los datos en el archivo “banknote.txt” contienen información sobre 100 notas bancarias suizas falsas ($Y = 0$) y 100 genuinas ($Y = 1$). Para tratar de predecir las posibilidades de que de que Y tenga cierto valor, se incluyen seis medidas físicas de las notas en milímetros.

Primero, veamos la matriz de gráficos para estudiar el fenómeno:

```
banknote = read.table("banknote.txt",
               header=T)

pairs(banknote[,-7], col = banknote$Y+1, pch = banknote$Y+20)
```



Ahora veamos cómo sería un modelo logístico aplicado a estos datos:

```
banknote.lm <- glm(Y~.,dat=banknote, family = binomial(link="logit"))
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(banknote.lm)
```

```
##
## Call:
## glm(formula = Y ~ ., family = binomial(link = "logit"), data = banknote)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -8.216e-05  -2.100e-08   0.000e+00   2.100e-08   7.237e-05
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.013e+03  3.063e+07   0.000    1.000
## Length       3.084e+01  1.666e+05   0.000    1.000
## Left        -1.075e+01  3.254e+05   0.000    1.000
## Right        1.056e+01  2.541e+05   0.000    1.000
## Bottom       5.876e+01  4.354e+04   0.001    0.999
## Top          4.983e+01  3.730e+04   0.001    0.999
## Diagonal    -4.040e+01  3.655e+04  -0.001    0.999
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2.7726e+02  on 199  degrees of freedom
## Residual deviance: 2.0593e-08  on 193  degrees of freedom
## AIC: 14
##
## Number of Fisher Scoring iterations: 25
```

Hay un warning que nos dice que se detectó un ajuste perfecto para algunos puntos de los datos. La gráfica de arriba (Right x Diagonal) tiene una división lineal perfecta de los datos. Nótese que los resultados de aplicar un modelo logístico en este tipo de casos no es muy confiable puesto que $\hat{p}(x) = 1$

$$\frac{p(x)}{1-p(x)} = \exp(\beta_0 + \beta_1 x) = \exp(\beta_0) \cdot \exp(\beta_1 x)$$