Modelos lineales en Marketing: Validación

Series Temporales 2017/2018

X.Barber

Centro de Investigación Operativa
Universidad Miguel Hernández de Elche

Bondad del ajuste

Comporbando el modelo

- Una vez que tenemos un modelo adecuado, respecto a la significación de las variables que lo componen, podemos tener una cierta medida que nos indique si este modelo es o no del todo "bondadoso".
- Para ello utilizaremos inicialmente el AIC para compara entre modelos o el **pseudo-**R^2.

$$\text{pseudo} - R^2 = 1 - \frac{\text{Residual Deviance}}{\text{Model Deviance}}$$

- La interpretación de este valor es la siguiente para respuesta *gaussiana* es equivalente al R^2 de un ajuste por mínimos cuadrados ordinarios.

Fórmula que calcula un R^2 para datos con variable respuesta *Gaussiana*.

```
R2gauss <- function(y, model) {
    moy <- mean(y)
    N <- length(y)
    p <- length(model$coefficients) - 1
    SSres <- sum((y - predict(model))^2)
    SStot <- sum((y - moy)^2)
    R2 <- 1 - (SSres/SStot)
    Rajust <- 1 - (((1 - R2) * (N - 1))/(N - p - 1))
    return(data.frame(R2, Rajust, SSres, SStot))
}
```

Validación de los modelos

Comportamiento de los Residuos

- $e_{ij} \sim (0, \sigma^2)$, es decir, centrados en 0 y con distribución Normal.
- Homocedasticidad de los Residuos, es decir, que se comporten de forma aleatoria en los distintos grupos.
- Tampoco debe haber autocorrelación (Series Temporales).

Estas premisas se pueden estudiar de una forma gráfica y rápida, y en caso de dudas entrar al detalle realizando los Test correspondientes:

- Media 0, con un $t - test$
- Normalidad, con un `ks.test` o `shapiro.test`
- Homocedasticidad de residuos, con `bptest` o `ncvTest` entre otras alternativas.

Graficando los residuos

```
pdf("img/valida1.pdf")
par(mfrow = c(2, 2))  # init 4 charts in 1 panel
plot(modelo1c)
dev.off()
```

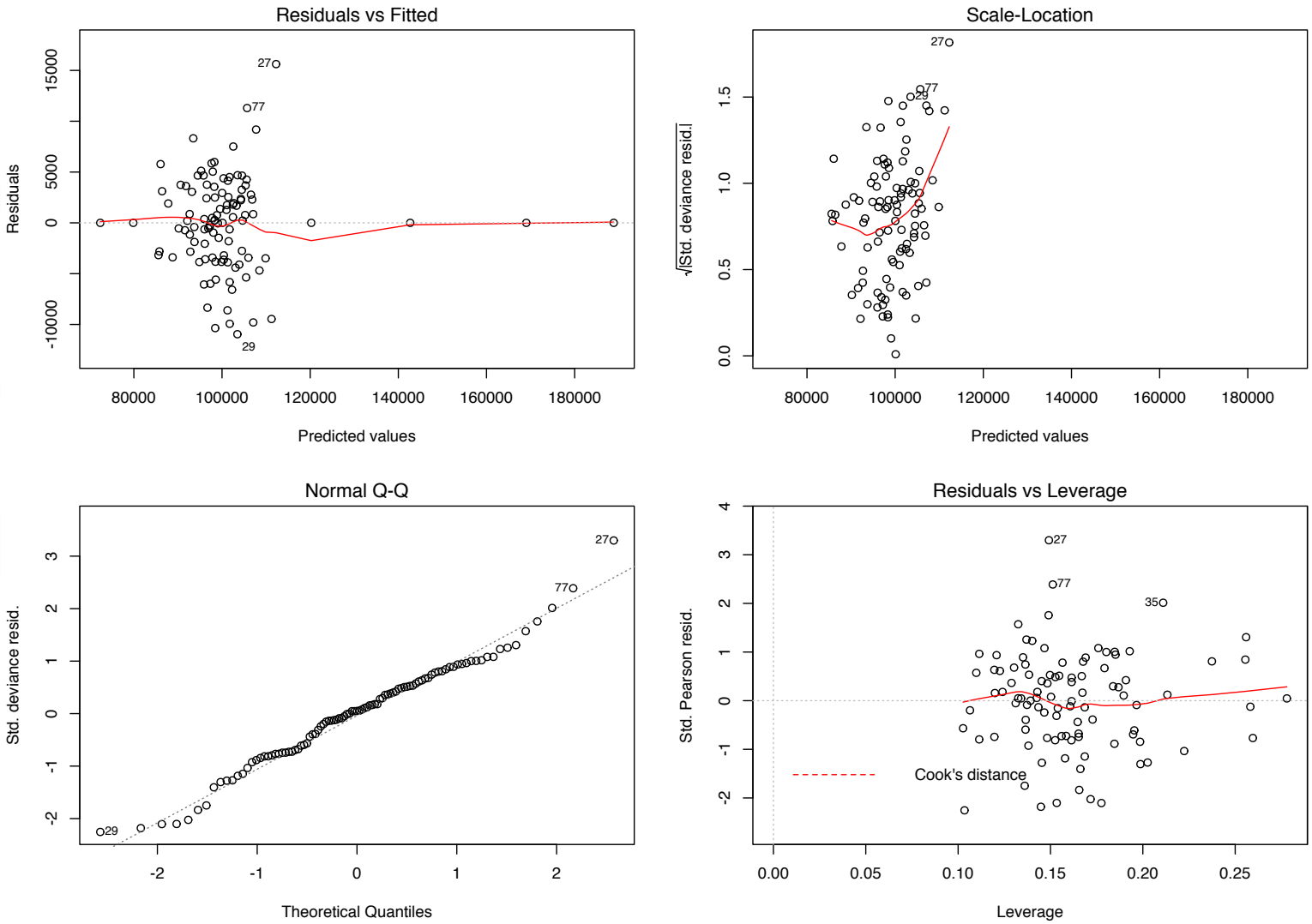
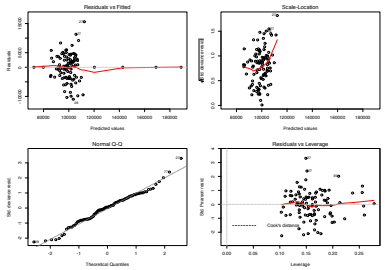

Residuals vs Fitted
Residuals
Predicted values
27
77
29
Scale-Location
√|Std. deviance resid.|
Predicted values
Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
Residuals vs Leverage
Std. Pearson resid.
Cook's distance
Leverage
35

Figure 1

Interpretación de los 4 gráficos

- Arriba izquierda: gráfico sobre homocedasticidad de los residuos, debe ser una nube de puntos sin patrón alguno.
- Abajo izquierda: gráfico sobre la normalidad de lso residuos, los residuos deben de estar sobre la diagonaly sin dibujar forma alguna.
- Arriba derecha: gráfico sobre la independencia de los residuos (los residuos se distribuyen por igual a lo largo de los rangos de los predictores), no debe tener patrón alguno (la línea roja debería ser recta).
- Abajo derecha: observaciones influyentes y atípicas, no debe de haber puntos muy alejados del resto.



```r
# Normalidad de los Residuos
normalidad <- shapiro.test(modelo1c$residuals)

# Heterocedasticidad de los Residuos
library(lmtest)
hetero1 <- bptest(modelo1c)
```

- El p-valor para la hipótesis de normalidad es .....
- El p-valor para la hipótesis de homocedasticidad es .....

Validado

Por lo que no rechazamos la Normalidad ni la Homocedasticidad de mis residuos. Damos por válido el modelo y se podría predecir si se desease, o estudiar con tenenimiento las diferencias existentes.

Validación Global

Otra alternativa para la validación de las hipótesis que debe de cumplir el modelo es la utilización del paquete `gvlma`

```
library(gvlma)
gvlma(modelo1c)
```

Ejemplo de Validación

Venta de caja de huevos

```
library(BCA)
data(Eggs)
fit.4<-glm(Cases ~ Egg.Pr + Easter +
             First.Week + Month +
           Beef.Pr + Cereal.Pr +
          Egg.Pr:Easter,
            family = "gaussian",
             data = Eggs)
```

```
##
## Call:
## glm(formula = Cases ~ Egg.Pr + Easter + First.Week + Month +
##     Beef.Pr + Cereal.Pr + Egg.Pr:Easter, family = "gaussian",
##     data = Eggs)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -10959.0   -3197.5       0.4    2940.8   15620.0
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                141694.03   21346.55   6.638 3.07e-09 ***
## Egg.Pr                       -420.27      83.58  -5.028 2.81e-06 ***
## EasterPre Easter          1288643.10  379463.63   3.396 0.001052 **
## EasterEaster               714789.67  232056.32   3.080 0.002805 **
## EasterPost Easter         -476819.01  304606.42  -1.565 0.121303
## First.WeekYes                5296.56    1294.16   4.093 9.84e-05 ***
## MonthFebruary               -2812.35    2490.22  -1.129 0.262001
## MonthMarch                  -2239.18    2648.38  -0.845 0.400267
## MonthApril                 -13785.33    3250.04  -4.242 5.74e-05 ***
## MonthMay                   -12556.06    2559.10  -4.906 4.56e-06 ***
## MonthJune                   -9632.83    2572.86  -3.744 0.000333 ***
## MonthJuly                  -11712.31    2362.50  -4.958 3.72e-06 ***
## MonthAugust                -15501.99    2507.50  -6.182 2.26e-08 ***
## MonthSeptember              -8617.15    2474.36  -3.483 0.000795 ***
## MonthOctober                -8959.11    2363.47  -3.791 0.000284 ***
## MonthNovember               -3136.47    2459.71  -1.275 0.205818
## MonthDecember                1519.74    2554.06   0.595 0.553444
## Beef.Pr                       250.63      78.08   3.210 0.001888 **
## Cereal.Pr                    -287.49     127.88  -2.248 0.027220 *
## Egg.Pr:EasterPre Easter   -12850.86    3876.13  -3.315 0.001359 **
## Egg.Pr:EasterEaster        -8104.30    2947.39  -2.750 0.007320 **
## Egg.Pr:EasterPost Easter    4587.81    3017.59   1.520 0.132222
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 26355714)
##
##     Null deviance: 2.1338e+10  on 104  degrees of freedom
## Residual deviance: 2.1875e+09  on  83  degrees of freedom
## AIC: 2113.4
```

```
R2gauss(Eggs$Cases, fit.4)
```

Table 1: Bondad de ajuste del modelo

R2	Rajust	SSres	SStot
0.897	0.872	2187524257	21338395482

Validación

```
# diagnostic plot
layout(matrix(c(1, 2, 3, 4), 2, 2))  # optional 4 graphs/page
plot(fit.4)
```

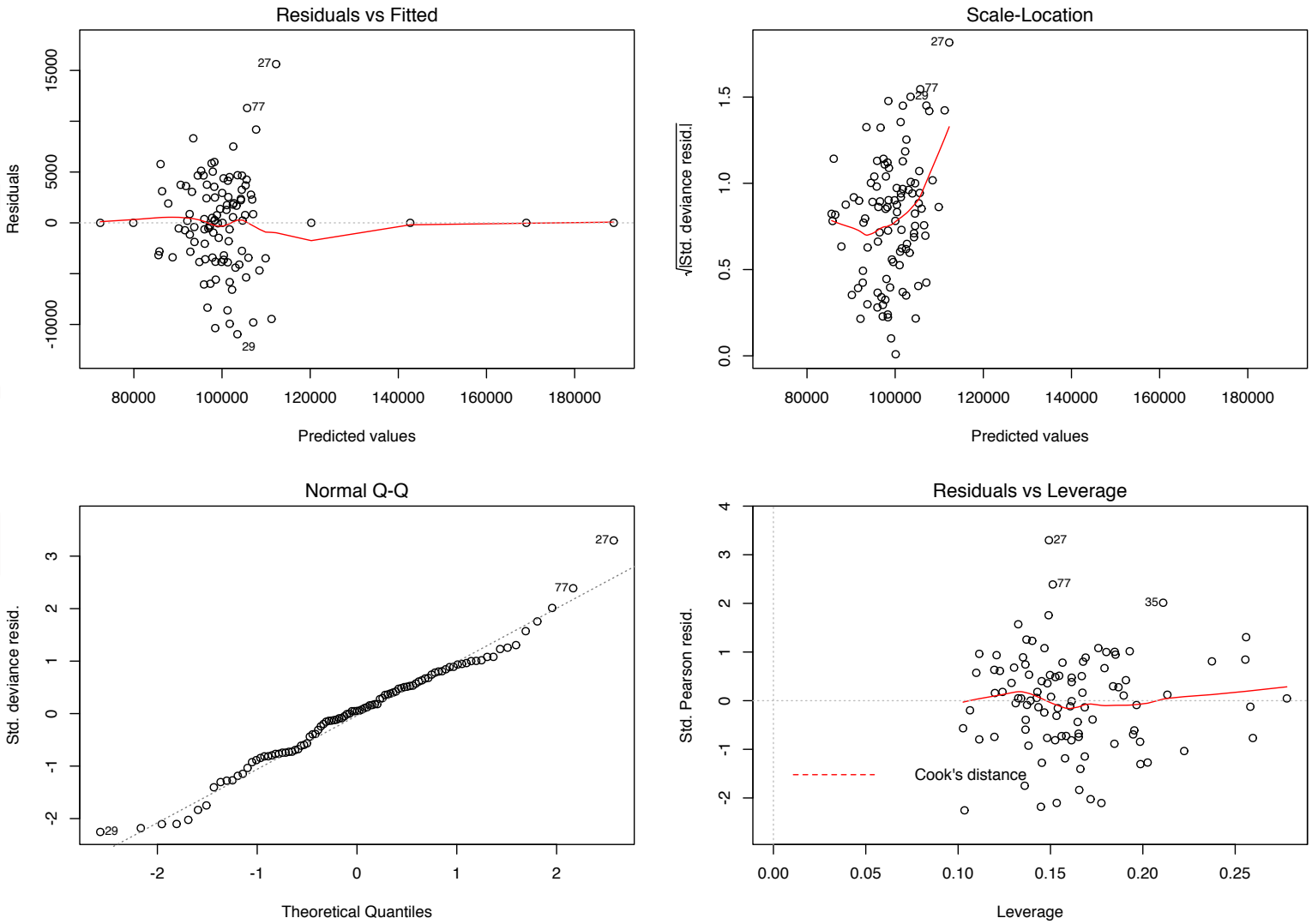

Residuals vs Fitted
Residuals
Predicted values
Scale-Location
√|Std. deviance resid.|
Predicted values
Normal Q-Q
Std. deviance resid.
Theoretical Quantiles
Residuals vs Leverage
Std. Pearson resid.
Cook's distance
Leverage

Figure 2

Del gráfico se extraen las siguientes conclusiones:

- Exceptuando algún punto sí parece que haya normalidad de los residuos.
- Pero parece que no existe Homocedasticidad.

Se puede contrastar de forma numérica para tener mayor seguridad.

Hipótesis de media de los residuos=0

```
residuos <- fit.4$residuals
t.test(residuos)
```

```
##
##  One Sample t-test
##
## data:  residuos
## t = -1.5761e-13, df = 104, p-value = 1
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -887.5566  887.5566
## sample estimates:
##    mean of x
## -7.05421e-11
```

Hipótesis de Normalidad

Preferimremos utilizar los residuos estutentizados:

```
res.st <- scale(fit.4$residuals)
## Shapiro-Wilk Normality Test
shapiro.test(res.st)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res.st
## W = 0.9795, p-value = 0.1037
```

Hipótesis de Homomocedasticidad

```
library(lmtest)
bptest(fit.4)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit.4
## BP = 27.492, df = 21, p-value = 0.1552
```

Hipótesis de autocorrelación

Atípicos (Outliers)

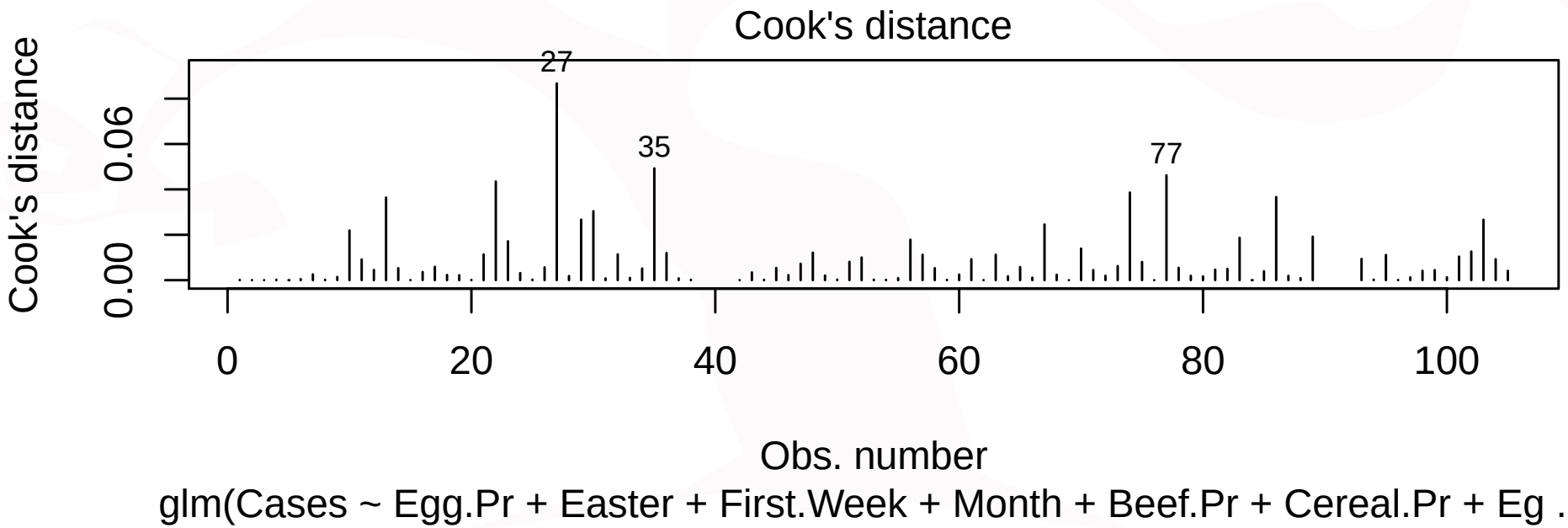
```
# Assessing Outliers
library(car)
outlierTest(fit.4)  # Bonferonni p-value for most extreme obs
```

```
##    rstudent unadjusted p-value Bonferonni p
## 27 3.517225         0.00043608     0.043172
```

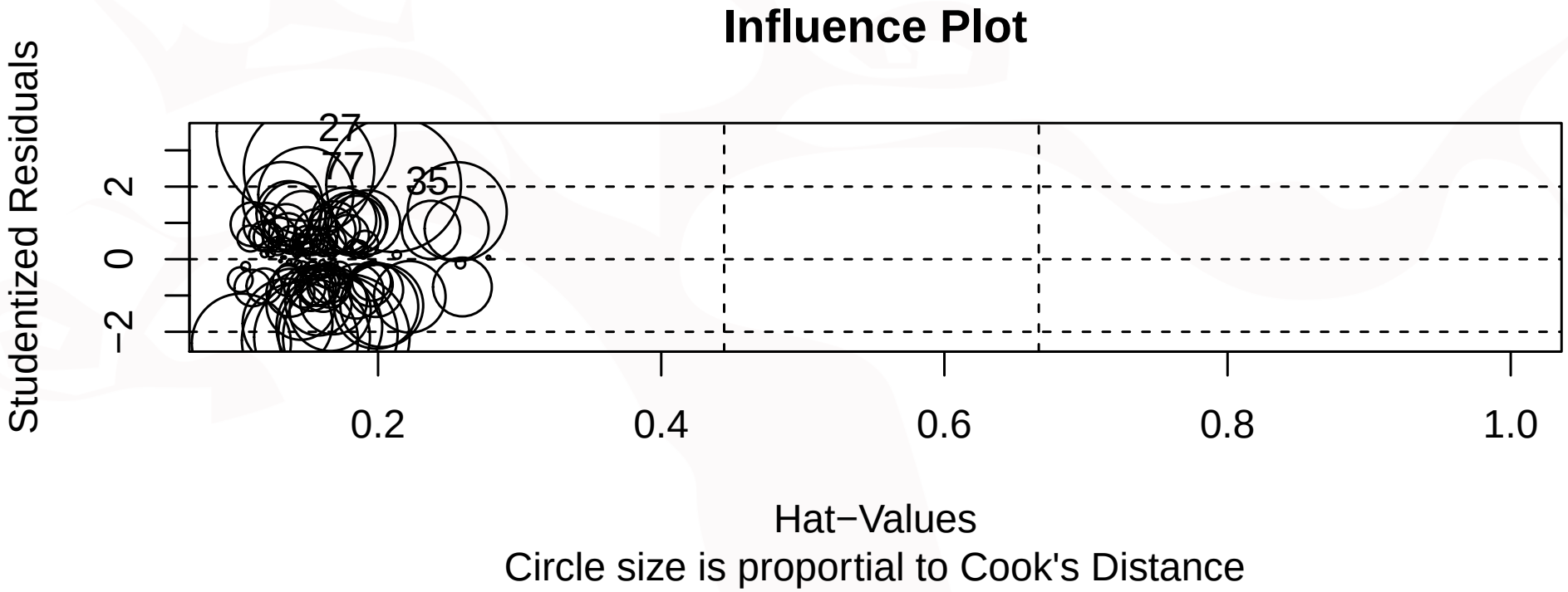
Influyentes

```
# Influential Observations added variable plots
avPlots(fit.4)
```

```
# Cook's D plot identify D values > 4/(n-k-1)
cutoff <- 4/((nrow(Eggs) - length(fit.4$coefficients) - 2))
plot(fit.4, which = 4, cook.levels = cutoff)
```



```
# Influence Plot
influencePlot(fit.4, id.method = "identify", main = "Influence Plot"
    sub = "Circle size is proportial to Cook's Distance")
```




```
##      StudRes       Hat       CookD
## 27 3.517225 0.1491577 0.08669905
## 35 2.051760 0.2109976 0.04926647
## 39      NaN 1.0000000         NaN
## 40      NaN 1.0000000         NaN
```

Multicolinealidad

```
# Evaluate Collinearity
vif(fit.4)  # variance inflation factors
```

```
##                       GVIF Df GVIF^(1/(2*Df))
## Egg.Pr        1.756347e+00  1        1.325272
## Easter        2.959536e+11  3       81.633911
## First.Week    1.176549e+00  1        1.084689
## Month         7.111277e+00 11        1.093264
## Beef.Pr       1.422014e+00  1        1.192482
## Cereal.Pr     1.675920e+00  1        1.294573
## Egg.Pr:Easter 2.950825e+11  3       81.593816
```

```
# Evaluate Collinearity
sqrt(vif(fit.4)) > 2  # problem?
```

```
##                  GVIF    Df GVIF^(1/(2*Df))
## Egg.Pr          FALSE FALSE             FALSE
## Easter           TRUE FALSE              TRUE
## First.Week      FALSE FALSE             FALSE
## Month            TRUE  TRUE             FALSE
## Beef.Pr         FALSE FALSE             FALSE
## Cereal.Pr       FALSE FALSE             FALSE
## Egg.Pr:Easter    TRUE FALSE              TRUE
```

No linealidad

En nuestro ejemplo no podemos hacer por haber una interacción significativa.

```
# Evaluate Nonlinearity component + residual plot
crPlots(fit.4)
# Ceres plots
ceresPlots(fit.4)
```

En nuestro ejemplo no podemos hacer por haber una interacción significativa.

```
# Evaluate Nonlinearity component + residual plot
crPlots(fit)
# Ceres plots
ceresPlots(fit.4)
```

Si pudisemos seríaalgo como así:

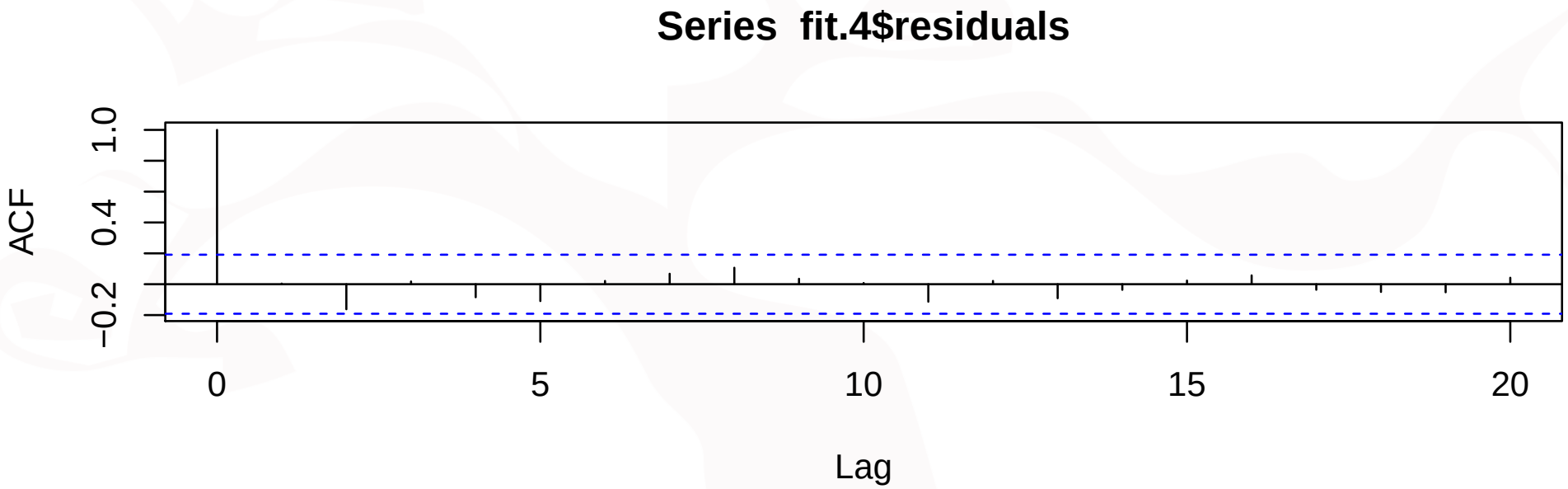
```
# Evaluate Nonlinearity # component + residual plot
crPlots(fit.4b)
```

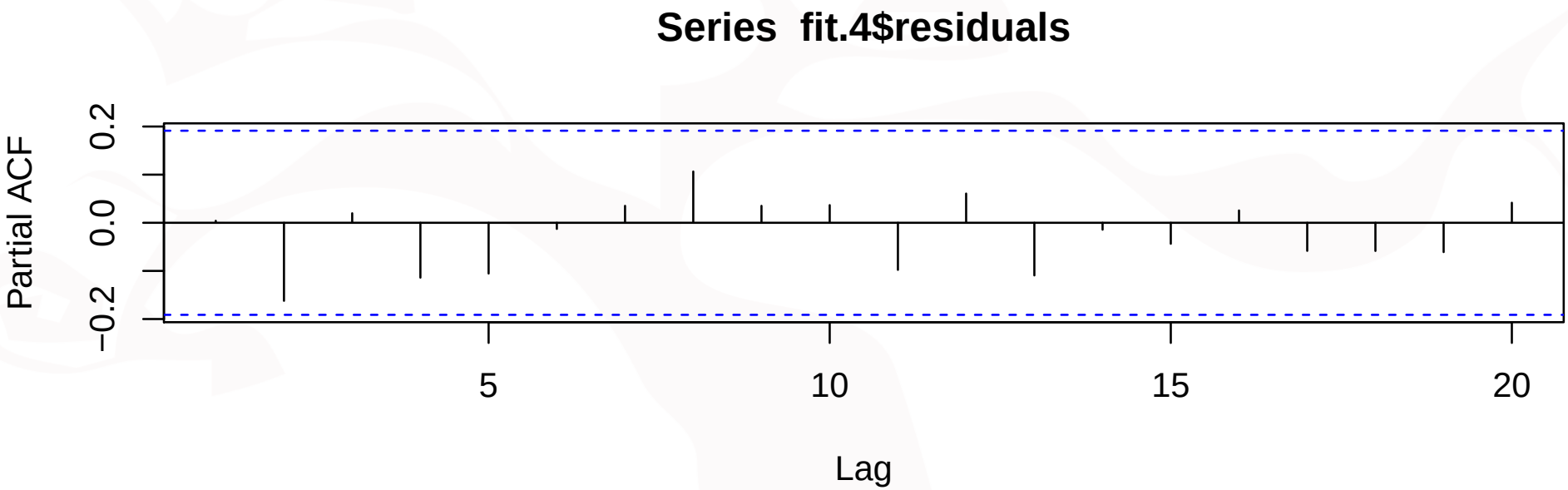
```
# Ceres plots
ceresPlots(fit.4b)
```

Autocorrelación

```
# Test for Autocorrelated Errors
durbinWatsonTest(fit.4)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     0.004144631      1.987962   0.228
##  Alternative hypothesis: rho != 0
```


Series fit.4$residuals
ACF
1.0
0.4
−0.2
0
5
10
15
20
Lag


Series fit.4$residuals
Partial ACF
0.2
0.0
−0.2
5
10
15
20
Lag

Estudio conjunto de la Validación

Solo para modelos "lm"

```
# Global test of model assumptions
gvmodel <- gvlma(fit.4)
summary(gvmodel)
```

Pero si lo ajustamos como un *lm*, por ser la variable respuesta *Gaussian*:

```
fit.4v <- lm(Cases ~ Egg.Pr + Easter + First.Week + Month + Beef.Pr
    Cereal.Pr + Egg.Pr:Easter, data = Eggs)
```

```
# Global test of model assumptions
gvmodel <- gvlma(fit.4v)
gvmodel
plot(gvmodel)
```

```
ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05

Call:
 gvlma(x = fit.4v)

                     Value p-value                Decision
Global Stat        5.81577  0.2133 Assumptions acceptable.
Skewness           0.05631  0.8124 Assumptions acceptable.
Kurtosis           3.05069  0.0807 Assumptions acceptable.
Link Function      1.88768  0.1695 Assumptions acceptable.
Heteroscedasticity 0.82109  0.3649 Assumptions acceptable.
```

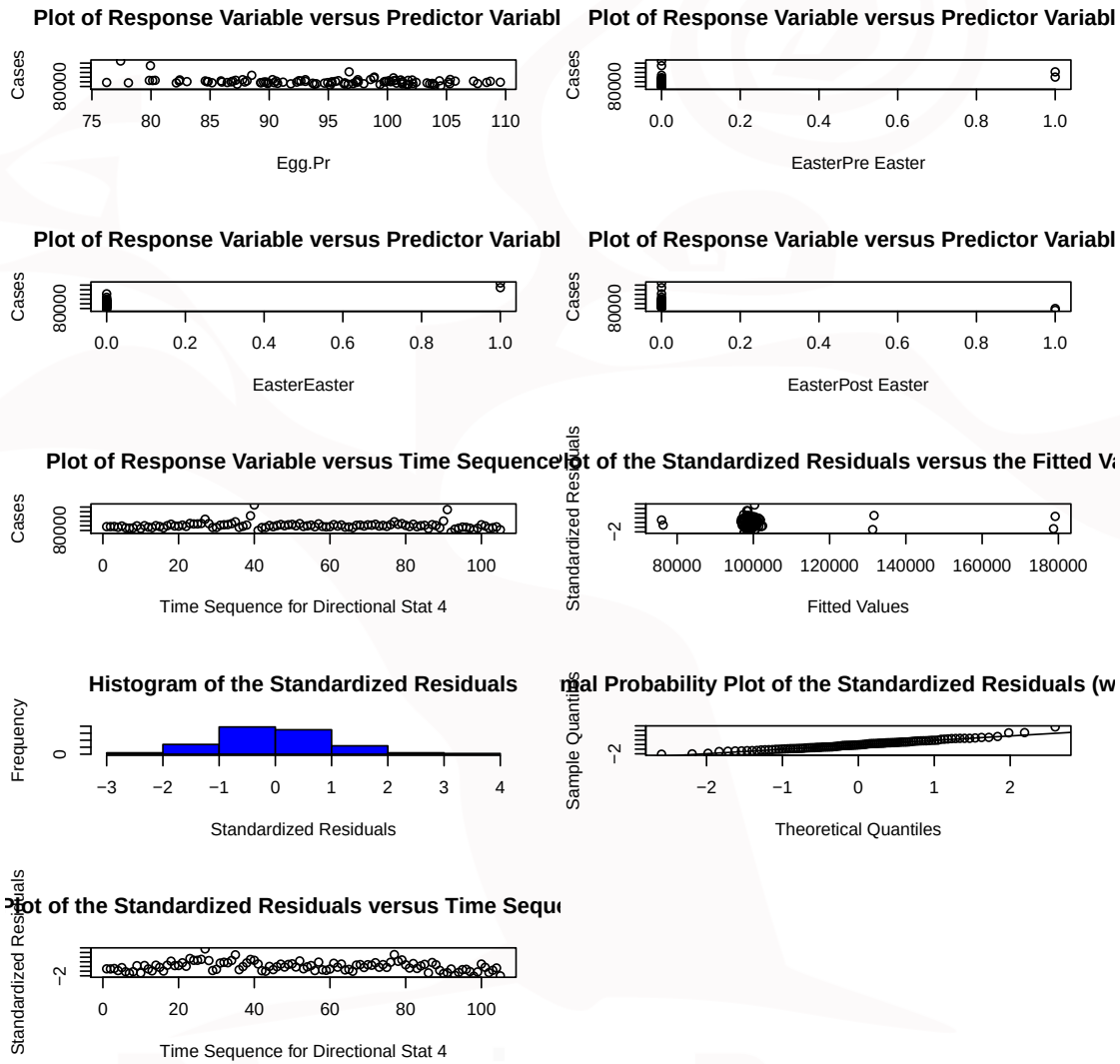

Plot of Response Variable versus Predictor Variabl
Cases
80000
75 80 85 90 95 100 105 110
Egg.Pr
Plot of Response Variable versus Predictor Variabl
Cases
80000
0.0 0.2 0.4 0.6 0.8 1.0
EasterPre Easter
Plot of Response Variable versus Predictor Variabl
Cases
80000
0.0 0.2 0.4 0.6 0.8 1.0
EasterEaster
Plot of Response Variable versus Predictor Variabl
Cases
80000
0.0 0.2 0.4 0.6 0.8 1.0
EasterPost Easter
Plot of Response Variable versus Time Sequence
Cases
80000
0 20 40 60 80 100
Time Sequence for Directional Stat 4
lot of the Standardized Residuals versus the Fitted V
Standardized Residuals
−2
80000 100000 120000 140000 160000 180000
Fitted Values
Histogram of the Standardized Residuals
Frequency
0
−3 −2 −1 0 1 2 3 4
Standardized Residuals
mal Probability Plot of the Standardized Residuals (w
Sample Quantiles
−2
−2 −1 0 1 2
Theoretical Quantiles
lot of the Standardized Residuals versus Time Sequ
Standardized Residuals
−2
0 20 40 60 80 100
Time Sequence for Directional Stat 4

Figure 3

Predicción

Nuevos datos

- Lo más importante para predecir de forma correcta cuando en el modelo tenemos variable "Factor" es no equivocarnos al escribir.
- No es lo mismo para R: **Febrary** que **February** que **february**
- El comando es sencillo: `predict(....)`

Comado predict

```
predict(object, newdata = NULL,
        type = c("link", "response", "terms"),
        se.fit = FALSE, dispersion = NULL, terms = NULL,
        na.action = na.pass, ...}
```

Ejemplo

Mis variables en el modelo son: - Egg.Pr Easter First.Week Month Beef.Pr Cereal.Pr

Debo de introducir valores reales en la predicción, para ellos mirare los rangos de estas variables y así poder predecir en consonancia con lso datos que he modelizado, de lo contrario la predicción no será fiable.

Rangos y "levels":

Egg.Pr: 76.25, 109.55

Beef.Pr: 129.79, 166.48

Cereal.Pr: 102.35, 130.28

Easter: Non Easter, Pre Easter, Easter, Post Easter

First.Week: No, Yes

Month: January, February, March, April, May, June, July, August, September, October, November, December

Egg.Pr: **100**

Beef.Pr: **130**

Cereal.Pr: **105**

Easter: **“Pre Easter”**

First.Week: **“No”**

Month: **October**

```
Nuevos.Datos<-data.frame(Egg.Pr=100,
                         Beef.Pr=130,
                         Cereal.Pr=105,
                         Easter="Easter",
                         First.Week="No",
                         Month="April")
predict(fit.4, Nuevos.Datos,type="response" ,se.fit=TRUE)
```

```
## $fit
##         1
## -7362.921
##
## $se.fit
## [1] 63140.58
##
## $residual.scale
## [1] 5133.782
```

- residual.scale = residual standard deviations

```
Nuevos.Datos<-data.frame(Egg.Pr=100,
                         Beef.Pr=130,
                         Cereal.Pr=105,
                         Easter="Pre Easter",
                         First.Week="Yes",
                         Month="March")
predict(fit.4, Nuevos.Datos,type="response" ,se.fit=TRUE)
```

```
## $fit
##        1
## 108677.6
##
## $se.fit
## [1] 9442.337
##
## $residual.scale
## [1] 5133.782
```

```
Nuevos.Datos<-data.frame(Egg.Pr=100,
                         Beef.Pr=130,
                         Cereal.Pr=105,
                         Easter="Post Easter",
                         First.Week="Yes",
                         Month="April")
predict(fit.4, Nuevos.Datos,type="response" ,se.fit=TRUE)
```

```
## $fit
##        1
## 75536.01
##
## $se.fit
## [1] 4997.605
##
## $residual.scale
## [1] 5133.782
```

```
Nuevos.Datos<-data.frame(Egg.Pr=100,
                         Beef.Pr=130,
                         Cereal.Pr=105,
                         Easter="Non Easter",
                         First.Week="Yes",
                         Month="October")
predict(fit.4, Nuevos.Datos,type="response" ,se.fit=TRUE)
```

```
## $fit
##        1
## 98400.15
##
## $se.fit
## [1] 2262.574
##
## $residual.scale
## [1] 5133.782
```

```
Nuevos.Datos<-data.frame(Egg.Pr=100,
                         Beef.Pr=130,
                         Cereal.Pr=105,
                         Easter="Non Easter",
                         First.Week="No",
                         Month="October")
predict(fit.4, Nuevos.Datos,type="response" ,se.fit=TRUE)
```

```
## $fit
##        1
## 93103.58
##
## $se.fit
## [1] 2185.813
##
## $residual.scale
## [1] 5133.782
```