videos

GLM

modelos lineales

- Los modelos lineales son la técnica estadística aplicada más útil. Sin embargo, no están exentos de limitaciones.
  - Los modelos de respuesta aditiva no tienen mucho sentido si la respuesta es discreta o estrictamente positiva.
  - Los modelos de error aditivo a menudo no tienen sentido, por ejemplo, si el resultado tiene que ser positivo.
  - Las transformaciones suelen ser difíciles de interpretar.
    - Es valioso modelar los datos en la escala en la que se recopilaron.
    - Las transformaciones particularmente interpelables, los logaritmos naturales en específico, no son aplicables para valores negativos o cero.

GLM

- Introducido en un artículo RSSB de 1972 por Nelder y Wedderburn.
- Involucra tres componentes
  - Un modelo de * familia exponencial * para la respuesta.
  - Un componente sistemático a través de un predictor lineal.
  - Una función de enlace que conecta los medios de la respuesta al predictor lineal.

ejemplo:modelo lineal

- Suponga que $Y_i \sim N(\mu_i, \sigma^2)$ (la distribución gaussiana es una distribución familiar exponencial).
- Defina el predictor lineal como $\eta_i = \sum_{k=1}^{p} X_{ik}\beta_k$.
- El enlace funciona como g de modo que $g(\mu) = \eta$.
  - Para modelos lineales $g(\mu) = \mu$ de modo que $\mu_i = \eta_i$
- Esto produce el mismo modelo de verosimilitud que nuestro modelo lineal gaussiano de error aditivo

$$Y_i = \sum_{k=1}^{p} X_{ik}\beta_k + \epsilon_i$$

donde $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$

ejemplo:regresion logistica

- Suponga que $Y_i \sim Bernoulli(\mu_i)$ de modo que $E[Y_i] = \mu_i$ donde $0 \leq \mu_i \leq 1$.
- Predictor lineal $\eta_i = \sum_{k=1}^{p} X_{ik}\beta_k$
- Función de enlace $g(\mu) = \eta = \log\left(\frac{\mu}{1-\mu}\right)$ g son las probabilidades de registro (naturales), conocidas como **logit**.
- Tenga en cuenta que podemos invertir la función logit como

$$\mu_i = \frac{\exp(\eta_i)}{1+\exp(\eta_i)} \quad \text{y} \quad 1 - \mu_i = \frac{1}{1+\exp(\eta_i)}$$

Por tanto, la probabilidad es

$$\prod_{i=1}^{n} \mu_i^{y_i}(1-\mu_i)^{1-y_i} = \exp\left(\sum_{i=1}^{n} y_i\eta_i\right)\prod_{i=1}^{n}(1+\ eta_i)^{-1}$$

ejemplo: regresion de poisson

- Asumir que $Y_i \sim Poisson(\mu_i)$ así que $E[Y_i] = \mu_i$ donde $0 \leq \mu_i$
- Predictor lineal $\eta_i = \sum_{k=1}^{p} X_{ik}\beta_k$
- Función de enlace $g(\mu) = \eta = \log(\mu)$
- Recuerrda que e^x es la inversa de $\log(x)$ así que:

$$\mu_i = e^{\eta_i}$$

Por tanto, la probabilidad(verosimilitud) es

$$\prod_{i=1}^{n}(y_i!)^{-1}\mu_i^{y_i}e^{-\mu_i} \propto \exp\left(\sum_{i=1}^{n} y_i\eta_i - \sum_{i=1}^{n}\mu_i\right)$$

algunas notas

- En cada caso, la única forma en que la probabilidad depende de los datos es a través de

$$\sum_{i=1}^{n} y_i\eta_i = \sum_{i=1}^{n} y_i \sum_{k=1}^{p} X_{ik}\beta_k = \sum_{k=1}^{p}\beta_k \sum_{i=1}^{n} X_{ik}y_i$$

Por lo tanto, si no necesitamos los datos completos, solo $\sum_{i=1}^{n} X_{ik}y_i$. Esta simplificación es una consecuencia de la elección de las funciones de enlace llamadas "canónicas". * (Esto tiene que derivarse). Todos los modelos alcanzan su máximo en la raíz de las llamadas ecuaciones normales.

$$0 = \sum_{i=1}^{n} \frac{(Y_i - \mu_i)}{Var(Y_i)} W_i$$

donde W_i son la derivada de la inversa de la función de enlace.

- pafra los modelos lineales $Var(Y_i) = \sigma^2$ es constante.
- Para el caso bernoulli $Var(Y_i) = \mu_i(1 - \mu_i)$
- Para el caso poisson $Var(Y_i) = \mu_i$.
- En los últimos casos, a menudo es relevante tener un modelo de varianza más flexible, incluso si no corresponde a una probabilidad real

$$0 = \sum_{i=1}^{n} \frac{(Y_i - \mu_i)}{\phi\mu_i(1 - \mu_i)} W_i \quad \text{and} \quad 0 = \sum_{i=1}^{n} \frac{(Y_i - \mu_i)}{\phi\mu_i} W_i$$

- Estas se denominan ecuaciones normales de 'cuasi-verosimilitud'
- Las ecuaciones normales deben resolverse iterativamente. Resultando en $\hat{\beta}_k$ y, si se incluye, $\hat{\phi}$.
- Las respuestas predictoras lineales predichas se pueden obtener como $\hat{\eta} = \sum_{k=1}^{p} X_k\hat{\beta}_k$ *Respuestas medias previstas como $\hat{\mu} = g^{-1}(\hat{\eta})$
- Los coeficientes se interpretan como

$$g(E[Y|X_k = x_k + 1, X_{\sim k} = x_{\sim k}]) - g(E[Y|X_k = x_k, X_{\sim k} = x_{\sim k}]) = \beta_k$$

o el cambio en la función de enlace de la respuesta esperada por cambio unitario en X_k manteniendo constantes otros regresores.

- Para hacerlo, se utilizan variaciones del algoritmo de Newon / Raphson.
- Las asintóticas se utilizan normalmente para inferencias.
- Muchas de las ideas de modelos lineales se pueden llevar a GLM.

Modelos lineales generalizados, datos binarios

- Con frecuencia nos preocupamos por los resultados que tienen dos valores
  - Vivo/muerto
  - Ganar/perder
  - Exito/fracaso
  - etc
- Resultados denominados binarios, Bernoulli o 0/1
- La recopilación de resultados binarios intercambiables para los mismos datos de covariables se denominan resultados binomiales.

```
load("C:/Users/luism/Downloads/ravensData.rda")
head(ravensData)
```

```
##   ravenWinNum ravenWin ravenScore opponentScore
## 1           1        W         24             9
## 2           1        W         38            35
## 3           1        W         28            13
## 4           1        W         34            31
## 5           1        W         44            13
## 6           0        L         23            24
```

Dado el caso que se hiciera con regresión lineal

$$RW_i = b_0 + b_1 RS_i + e_i$$

RW_i - 1 si gana un Ravens, 0 si no

RS_i - Número de puntos que anotó Ravens

b_0 - probabilidad de que los Ravens ganen si obtienen 0 puntos

b_1 - aumentar la probabilidad de que los Ravens ganen por cada punto adicional

e_i - variación residual debida

```
lmRavens <- lm(ravensData$ravenWinNum ~ ravensData$ravenScore)
summary(lmRavens)$coef
```

```
##                         Estimate  Std. Error  t value   Pr(>|t|)
## (Intercept)           0.28503172 0.256643165 1.110615 0.28135043
## ravensData$ravenScore 0.01589917 0.009058997 1.755069 0.09625261
```

salidas Binarias 0/1

$$RW_i$$

Probabilidad (0,1)

$$\Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})$$

razon de probabilidades (odds) $(0, \infty)$

$$\frac{\Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}{1 - \Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}$$

Log odds $(-\infty, \infty)$

$$\log\left(\frac{\Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}{1 - \Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}\right)$$

regresion lineal vs logistica

Lineal

$$RW_i = b_0 + b_1 RS_i + e_i$$

o

$$E[RW_i|RS_i, b_0, b_1] = b_0 + b_1 RS_i$$

Logistica

$$\Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1}) = \frac{\exp(\mathrm{b_0} + \mathrm{b_1}\mathrm{RS_i})}{1 + \exp(\mathrm{b_0} + \mathrm{b_1}\mathrm{RS_i})}$$

o

$$\log\left(\frac{\Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}{1 - \Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}\right) = b_0 + b_1 RS_i$$

interpretacion de la regresion logistica

$$\log\left(\frac{\Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}{1 - \Pr(\mathrm{RW_i}|\mathrm{RS_i}, \mathrm{b_0}, \mathrm{b_1})}\right) = b_0 + b_1 RS_i$$

b_0 -Registre las probabilidades de que los Ravens ganen si obtienen cero puntos

b_1 - Registrar la proporción de probabilidades de ganar por cada punto anotado (en comparación con cero puntos)

$\exp(b_1)$ -Razón de probabilidades de la probabilidad de ganar por cada punto anotado (en comparación con cero puntos) **ej**

- Imagina que estás jugando un juego en el que lanzas una moneda con probabilidad de éxito p.
- Si sale cara, ganas X. Si sale cruz, pierde Y.
- ¿Qué deberíamos establecer X y Y para que el juego sea justo?

$$E[ganancias] = Xp - Y(1 - p) = 0$$

- Implica

$$\frac{Y}{X} = \frac{p}{1 - p}$$

- Las "odds" (prporcion) se pueden decir como "¿Cuánto debería estar dispuesto a pagar por una probabilidad de p de ganar un dólar?"
  - (Si $p > 0.5$ tiene que pagar más si pierde de lo que obtiene si gana).
  - (Si $p < 0.5$ tiene que pagar menos si pierde de lo que obtiene si gana).

simulacion

```
x <- seq(-10, 10, length = 1000)
manipulate(
    plot(x, exp(beta0 + beta1 * x) / (1 + exp(beta0 + beta1 * x)),
         type = "l", lwd = 3, frame = FALSE),
    beta1 = slider(-2, 2, step = .1, initial = 2),
    beta0 = slider(-2, 2, step = .1, initial = 0)
    )
```

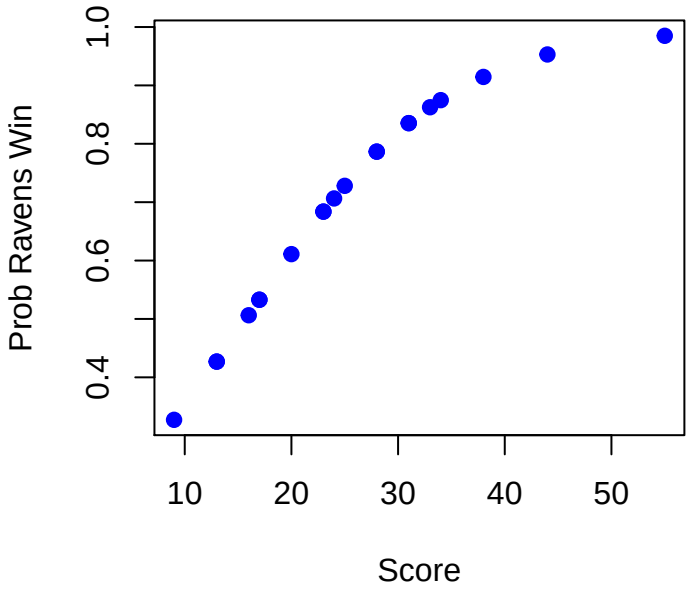
aplicando regresion logistica

```
logRegRavens <- glm(ravensData$ravenWinNum ~ ravensData$ravenScore,family="binomial")
summary(logRegRavens)
```

```
##
## Call:
## glm(formula = ravensData$ravenWinNum ~ ravensData$ravenScore,
##     family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7575  -1.0999   0.5305   0.8060   1.4947
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.68001    1.55412  -1.081     0.28
## ravensData$ravenScore  0.10658    0.06674   1.597     0.11
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24.435  on 19  degrees of freedom
## Residual deviance: 20.895  on 18  degrees of freedom
## AIC: 24.895
##
## Number of Fisher Scoring iterations: 5
```

valores ajustados

```
plot(ravensData$ravenScore,logRegRavens$fitted,pch=19,col="blue",xlab="Score",ylab="Prob Ravens Win")
```



Odds ratios e intervalos de confianza

```
exp(logRegRavens$coeff)
```

```
##           (Intercept) ravensData$ravenScore
##             0.1863724             1.1124694
```

```
exp(confint(logRegRavens))
```

```
##                              2.5 %   97.5 %
## (Intercept)           0.005674966 3.106384
## ravensData$ravenScore 0.996229662 1.303304
```

anova

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

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: ravensData$ravenWinNum
##
## Terms added sequentially (first to last)
##
##
##                       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
```

```
## NULL                                     19     24.435
## ravensData$ravenScore  1   3.5398        18     20.895  0.05991 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretando Odds Ratios

- No probabilidades
- Razón de probabilidades de 1 = sin diferencia en las probabilidades
- Logaritmo de la razón de probabilidades de 0 = sin diferencia en las probabilidades
- Odds ratio <0.5 o > 2 comúnmente un "efecto moderado"
- Riesgo relativo $\frac{\Pr(\mathrm{RW_i}|\mathrm{RS_i}=10)}{\Pr(\mathrm{RW_i}|\mathrm{RS_i}=0)}$ a menudo más fácil de interpretar, más difícil de estimar
- Para probabilidades pequeñas RR ≈ OR pero **¡no son lo mismo**!

Wikipedia on Odds Ratio

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

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: ravensData$ravenWinNum
##
## Terms added sequentially (first to last)
##
##
##                       Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                                     19     24.435
## ravensData$ravenScore  1   3.5398        18     20.895  0.05991 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Contar resultados, GLM de Poisson

Key ideas

- Muchos datos toman la forma de recuentos
  - Llamadas a un centro de llamadas
  - Número de casos de gripe en un área
  - Número de autos que cruzan un puente
- Los datos también pueden estar en forma de tarifas
  - Porcentaje de niños que aprueban un examen
  - Porcentaje de visitas a un sitio web desde un país
- La regresión lineal con transformación es una opción.

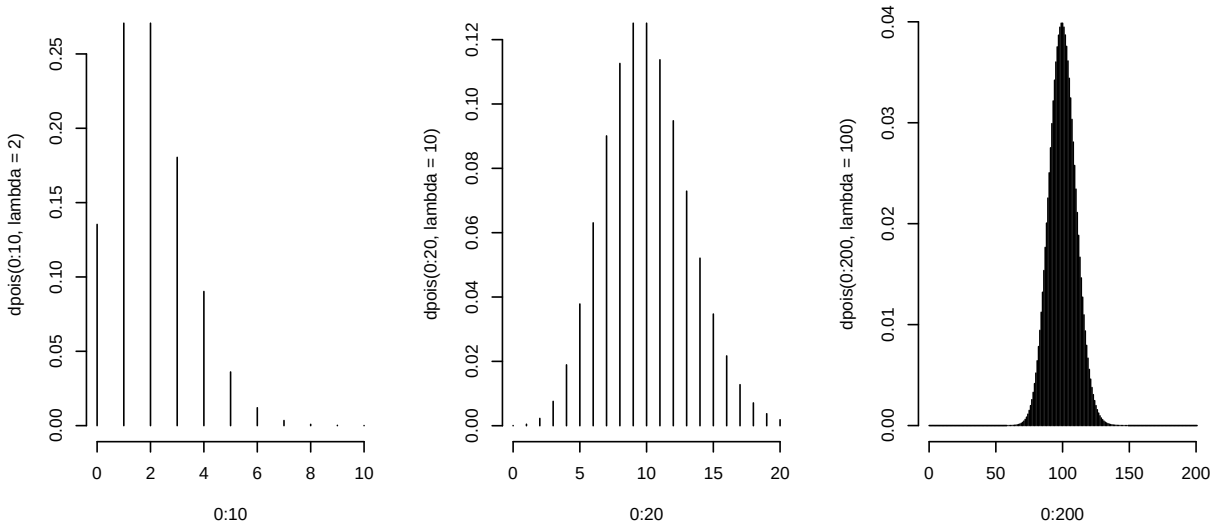
Poisson distribution

- La distribución de Poisson es un modelo útil para conteos y tasas

- Aquí se cuenta una tasa por tiempo de monitoreo
- Algunos ejemplos de usos de la distribución de Poisson
  - Modelado de visitas al tráfico web
  - Tasas de incidencia
  - Aproximación de probabilidades binomiales con p pequeños y n grandes
  - Análisis de datos de la tabla de contingencia
- $X \sim Poisson(t\lambda)$ si
$$P(X = x) = \frac{(t\lambda)^x e^{-t\lambda}}{x!}$$
Para $x = 0, 1, \ldots$.
- La media de la Poisson es $E[X] = t\lambda$, por lo tanto $E[X/t] = \lambda$
- La varianza de Poisson es $Var(X) = t\lambda$.
- El Poisson tiende a un normal como $t\lambda$ gets larse enlarge.

simulacion con diferentes $t\lambda$

```
par(mfrow = c(1, 3))
plot(0 : 10, dpois(0 : 10, lambda = 2), type = "h", frame = FALSE)
plot(0 : 20, dpois(0 : 20, lambda = 10), type = "h", frame = FALSE)
plot(0 : 200, dpois(0 : 200, lambda = 100), type = "h", frame = FALSE)
```



Example: Leek Group Website Traffic

- Considere los recuentos diarios del sitio web de Jeff Leek

http://biostat.jhsph.edu/~jleek/

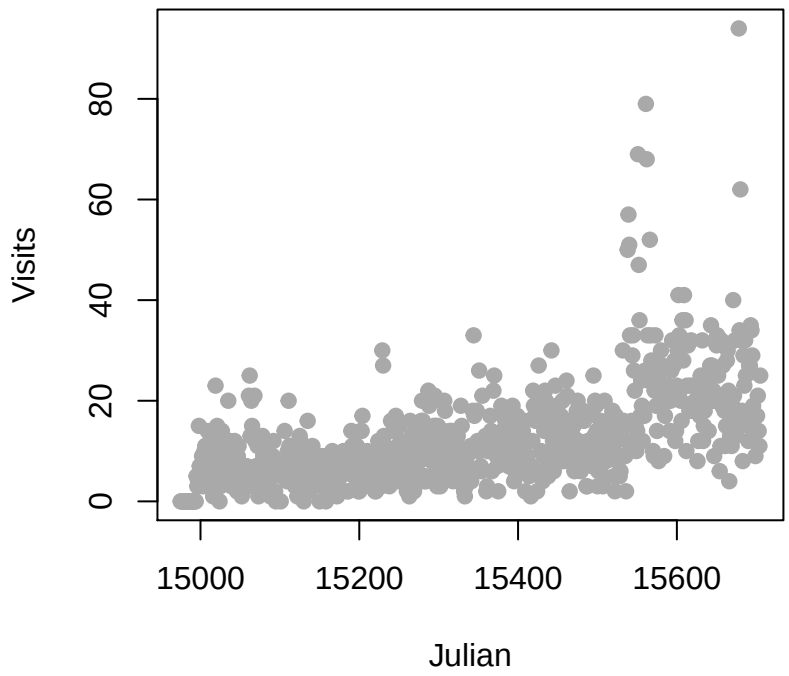
- Dado que la unidad de tiempo es siempre un día, establezca $t = 1$ y luego la media de Poisson se interpreta como visitas a la web por día. (Si establecemos $t = 24$, sería ser visitas web por hora).

```
load("C:/Users/luism/Documents/courses/07_RegressionModels/originalContent/003countOutcomes/data/gaData
gaData$julian <- julian(gaData$date)
head(gaData)
```

```
##         date visits simplystats julian
## 1 2011-01-01      0           0  14975
## 2 2011-01-02      0           0  14976
## 3 2011-01-03      0           0  14977
## 4 2011-01-04      0           0  14978
## 5 2011-01-05      0           0  14979
## 6 2011-01-06      0           0  14980
```

http://skardhamar.github.com/rga/

```
plot(gaData$julian,gaData$visits,pch=19,col="darkgrey",xlab="Julian",ylab="Visits")
```



Linear regression

$$NH_i = b_0 + b_1 JD_i + e_i$$

NH_i - número de visitas al sitio web

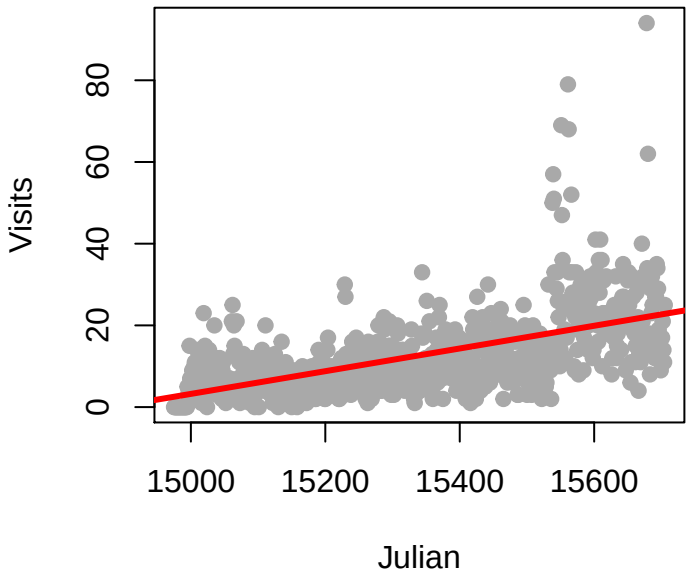
JD_i - día del año (día juliano)

b_0 - número de visitas en el día juliano 0 (01/01/1970)

b_1 - aumento en el número de visitas por unidad de día

e_i - variación debido a todo lo que no medimos

```
plot(gaData$julian,gaData$visits,pch=19,col="darkgrey",xlab="Julian",ylab="Visits")
lm1 <- lm(gaData$visits ~ gaData$julian)
abline(lm1,col="red",lwd=3)
```



Aparte, tomando el registro del resultado - Tomar el logaritmo natural del resultado tiene una interpretación específica. - Considere el modelo

$$\log(NH_i) = b_0 + b_1 JD_i + e_i$$

NH_i - número de visitas al sitio web

JD_i - día del año (día juliano)

b_0 - número de registro de visitas en el día juliano 0 (01/01/1970)

b_1 - aumento en el número de registros de visitas por día unitario

e_i - variación debido a todo lo que no medimos

Coeficientes exponenciadores

- $e^{E[\log(Y)]}$ media geométrica de Y.
  - Sin covariables, esto se estima por $e^{\frac{1}{n}\sum_{i=1}^{n}\log(y_i)} = (\prod_{i=1}^{n} y_i)^{1/n}$
- Cuando toma el logaritmo natural de resultados y ajusta un modelo de regresión, sus coeficientes exponenciados estiman cosas sobre las medias geométricas.
- e^{β_0} aciertos de la media geométrica estimada el día 0

- e^{β_1} aumento relativo estimado o disminución de los aciertos medios geométricos por día
- Hay un problema con los registros con cero recuentos, la adición de una constante funciona

```
round(exp(coef(lm(I(log(gaData$visits + 1)) ~ gaData$julian))), 5)
```

```
##   (Intercept) gaData$julian
##       0.00000       1.00231
```

Linear vs. Poisson regression

Lineal

$$NH_i = b_0 + b_1 JD_i + e_i$$

o

$$E[NH_i|JD_i, b_0, b_1] = b_0 + b_1 JD_i$$

Poisson/log-linear

$$\log\left(E[NH_i|JD_i, b_0, b_1]\right) = b_0 + b_1 JD_i$$

o

$$E[NH_i|JD_i, b_0, b_1] = \exp\left(b_0 + b_1 JD_i\right)$$

Diferencias multiplicativas

$$E[NH_i|JD_i, b_0, b_1] = \exp\left(b_0 + b_1 JD_i\right)$$

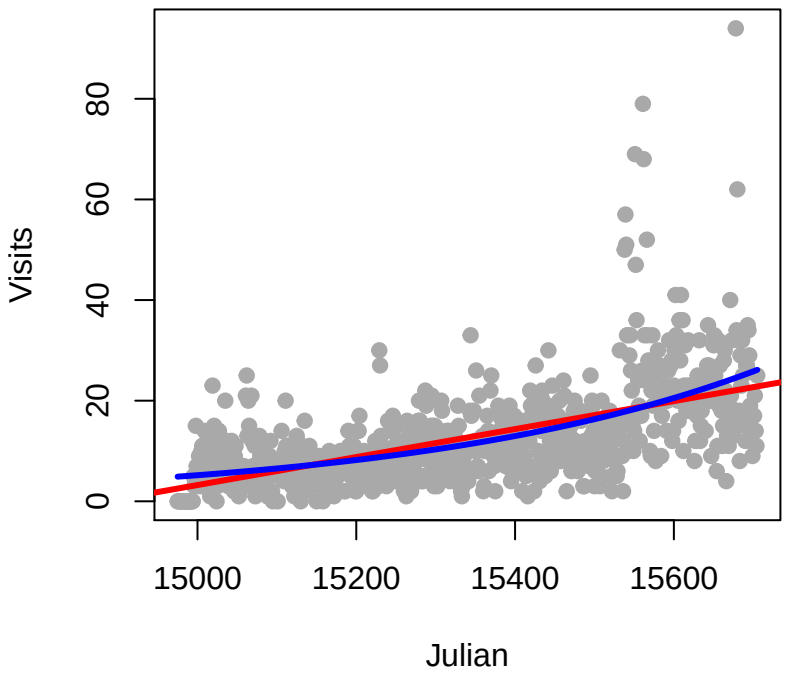
$$E[NH_i|JD_i, b_0, b_1] = \exp\left(b_0\right)\exp\left(b_1 JD_i\right)$$

<br

si JD_ise incrementa en una unidad, $E[NH_i|JD_i, b_0, b_1]$ es multiplicada por $\exp\left(b_1\right)$

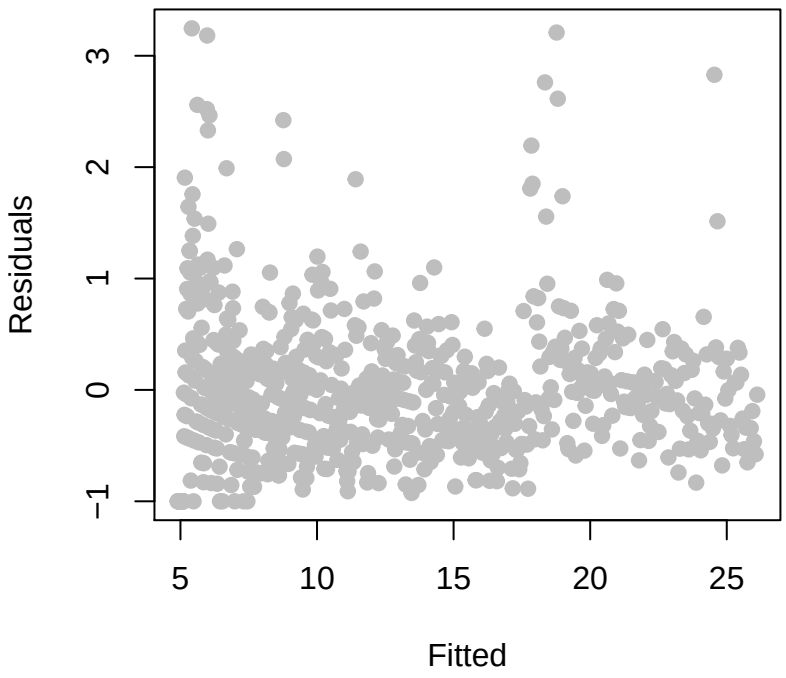
aplicacion de la distribucion de poisson

```
plot(gaData$julian,gaData$visits,pch=19,col="darkgrey",xlab="Julian",ylab="Visits")
glm1 <- glm(gaData$visits ~ gaData$julian,family="poisson")
abline(lm1,col="red",lwd=3); lines(gaData$julian,glm1$fitted,col="blue",lwd=3)
```



¿Relación media-varianza?

```
plot(glm1$fitted,glm1$residuals,pch=19,col="grey",ylab="Residuals",xlab="Fitted")
```



Errores estándar agnósticos del modelo

checar en el libro o en el video mas informacion sobre la relacion de media-varianza

```
library(sandwich)
confint.agnostic <- function (object, parm, level = 0.95, ...)
{
    cf <- coef(object); pnames <- names(cf)
    if (missing(parm))
        parm <- pnames
    else if (is.numeric(parm))
        parm <- pnames[parm]
    a <- (1 - level)/2; a <- c(a, 1 - a)
    pct <- stats:::format.perc(a, 3)
    fac <- qnorm(a)
    ci <- array(NA, dim = c(length(parm), 2L), dimnames = list(parm,
                                                               pct))
    ses <- sqrt(diag(sandwich::vcovHC(object)))[parm]
    ci[] <- cf[parm] + ses %o% fac
    ci
}
```

http://stackoverflow.com/questions/3817182/vcovhc-and-confidence-interval

estimacion de errores estandar

```
confint(glm1)
```

```
##                       2.5 %        97.5 %
## (Intercept)   -34.346577587 -31.159715656
## gaData$julian   0.002190043   0.002396461
```

```
confint.agnostic(glm1)
```

```
##                       2.5 %        97.5 %
## (Intercept)   -36.362674594 -29.136997254
## gaData$julian   0.002058147   0.002527955
```

tasas

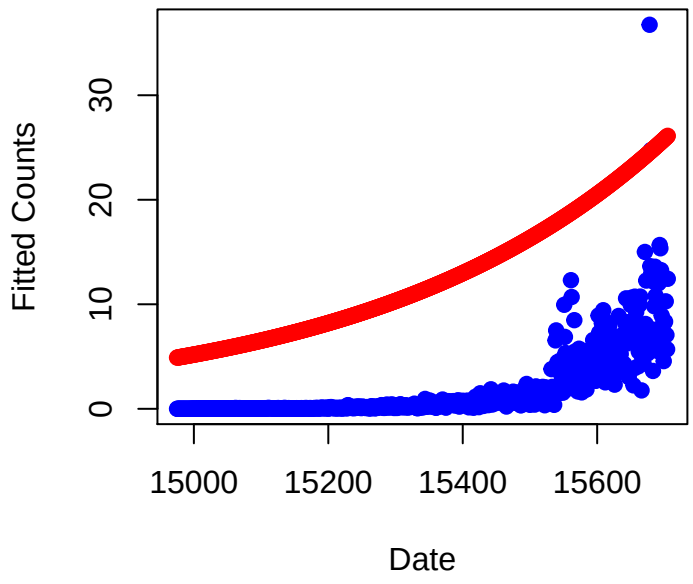
$$E[NHSS_i|JD_i, b_0, b_1]/NH_i = \exp\left(b_0 + b_1 JD_i\right)$$

$$\log\left(E[NHSS_i|JD_i, b_0, b_1]\right) - \log(NH_i) = b_0 + b_1 JD_i$$

$$\log\left(E[NHSS_i|JD_i, b_0, b_1]\right) = \log(NH_i) + b_0 + b_1 JD_i$$

**ajustando tasas en R

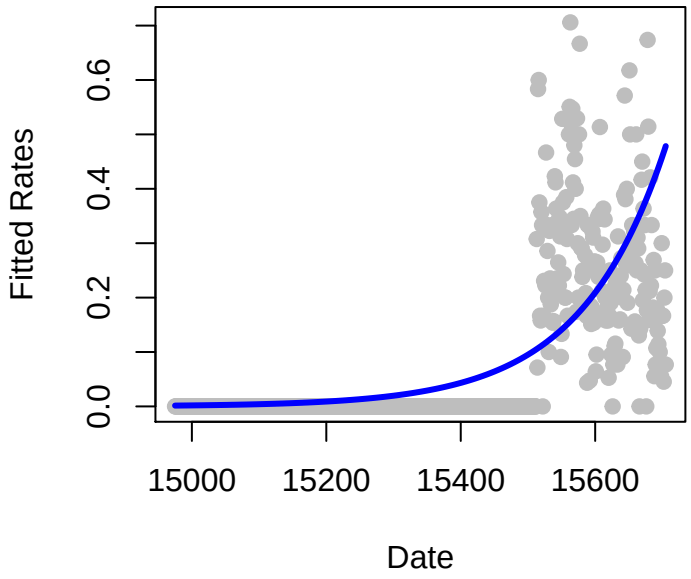
```
glm2 <- glm(gaData$simplystats ~ julian(gaData$date),offset=log(visits+1),
            family="poisson",data=gaData)
plot(julian(gaData$date),glm2$fitted,col="blue",pch=19,xlab="Date",ylab="Fitted Counts")
points(julian(gaData$date),glm1$fitted,col="red",pch=19)
```



More information

- Log-linear models and multiway tables
- Wikipedia on Poisson regression, Wikipedia on overdispersion
- Regression models for count data in R
- pscl package - the function *zeroinfl* fits zero inflated models.

```
glm2 <- glm(gaData$simplystats ~ julian(gaData$date),offset=log(visits+1),
            family="poisson",data=gaData)
plot(julian(gaData$date),gaData$simplystats/(gaData$visits+1),col="grey",xlab="Date",
     ylab="Fitted Rates",pch=19)
lines(julian(gaData$date),glm2$fitted/(gaData$visits+1),col="blue",lwd=3)
```



bonus

Cómo ajustar funciones usando modelos lineales

regresion spline

- Considere un modelo $Y_i = f(X_i) + \epsilon$.
- ¿Cómo podemos ajustar un modelo de este tipo utilizando modelos lineales (llamado suavizado de gráficos de dispersión)?
- Considere el modelo

$$Y_i = \beta_0 + \beta_1 X_i + \sum_{k=1}^{d} (x_i - \xi_k)_+ \gamma_k + \epsilon_i$$

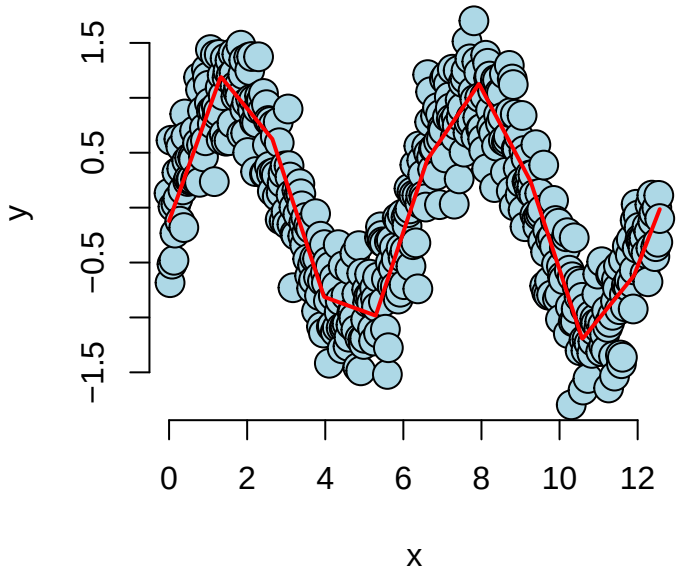
donde $(a)_+ = a$ si $a > 0$ y 0 de lo contrario y $\xi_1 \leq ... \leq \xi_d$ son puntos de nudo conocidos. * Demuéstrese a sí mismo que la función media

$$\beta_0 + \beta_1 X_i + \sum_{k=1}^{d} (x_i - \xi_k)_+ \gamma_k$$

Es continuo en los puntos de nudo.

Simulated example:

```
n <- 500; x <- seq(0, 4 * pi, length = n); y <- sin(x) + rnorm(n, sd = .3)
knots <- seq(0, 8 * pi, length = 20);
splineTerms <- sapply(knots, function(knot) (x > knot) * (x - knot))
xMat <- cbind(1, x, splineTerms)
yhat <- predict(lm(y ~ xMat - 1))
plot(x, y, frame = FALSE, pch = 21, bg = "lightblue", cex = 2)
lines(x, yhat, col = "red", lwd = 2)
```

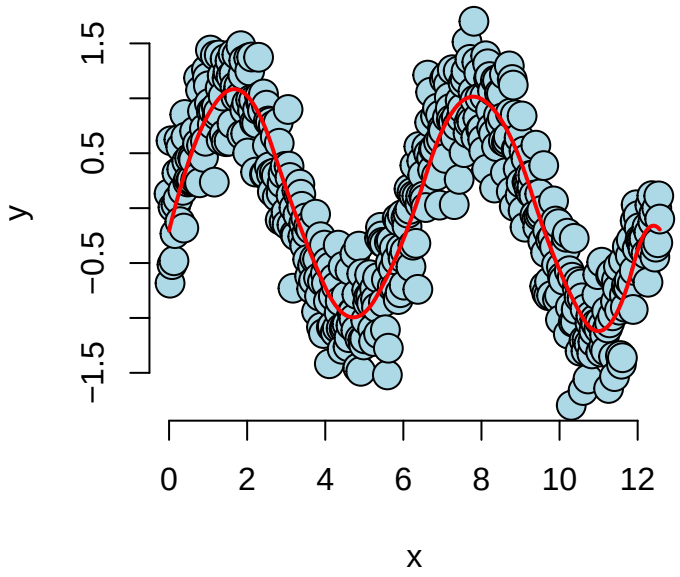


Sumar términos al cuadrado

- Agregar términos al cuadrado lo hace continuamente diferenciable en los puntos de nudo.
- La adición de términos cúbicos hace que sea dos veces diferenciable de forma continua en los puntos de nudo; etcétera.

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \sum_{k=1}^{d} (x_i - \xi_k)_+^2 \gamma_k + \epsilon_i$$

```r
splineTerms <- sapply(knots, function(knot) (x > knot) * (x - knot)^2)
xMat <- cbind(1, x, x^2, splineTerms)
yhat <- predict(lm(y ~ xMat - 1))
plot(x, y, frame = FALSE, pch = 21, bg = "lightblue", cex = 2)
lines(x, yhat, col = "red", lwd = 2)
```



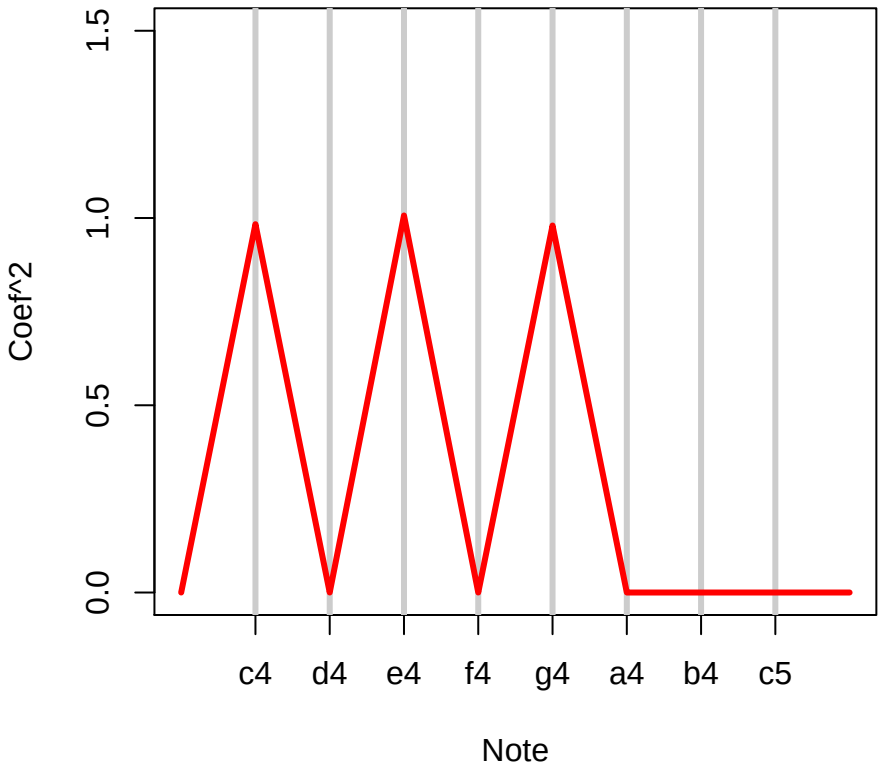
Notas

- La colección de regresores se llama base.
  - La gente ha pasado **mucho** tiempo pensando en las bases de este tipo de problema. Por lo tanto, considere esto solo como un adelanto.
- Los términos de un solo punto(de quiebre) pueden ajustarse a procesos similares a los de un palo de hockey.
- Estas bases también se pueden utilizar en GLM.
- no sabemos cuantos puntoa de ruptura poner
- Un problema con estos enfoques es la gran cantidad de parámetros introducidos.
  - Requiere algún método de la denominada regularización.

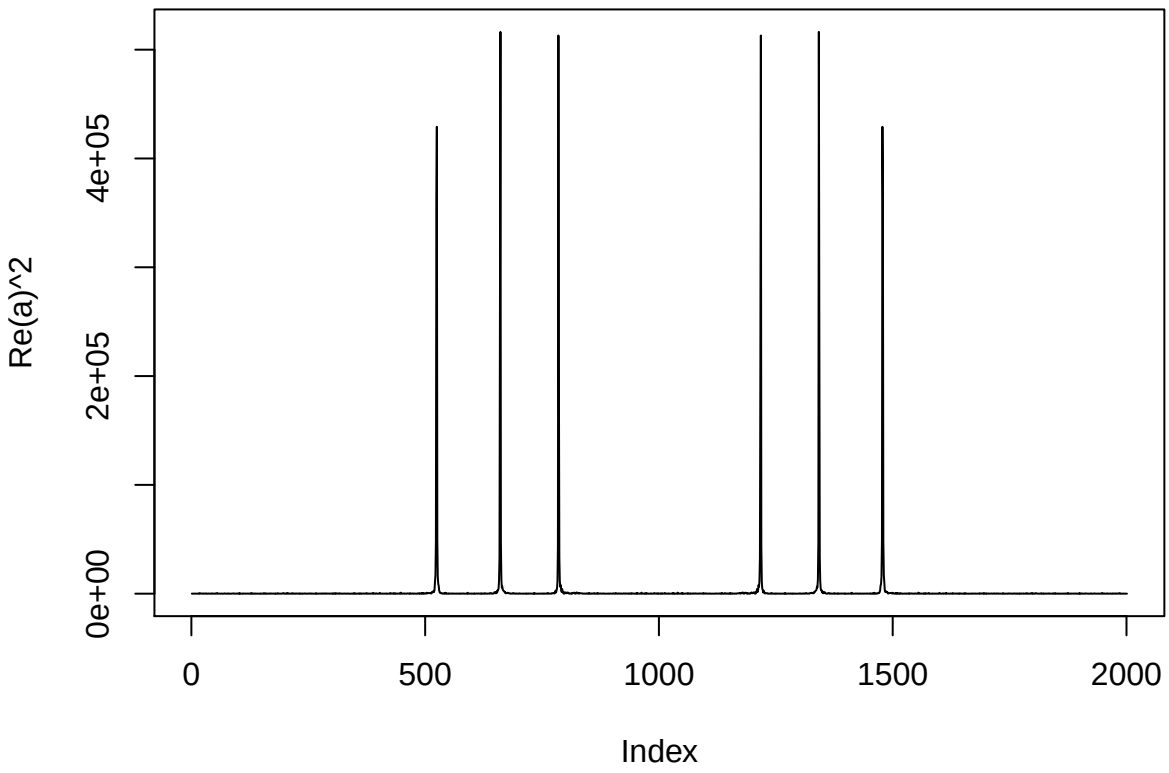
Harmonicos usinando linear models

```r
##Chord finder, playing the white keys on a piano from octave c4 - c5
notes4 <- c(261.63, 293.66, 329.63, 349.23, 392.00, 440.00, 493.88, 523.25)
t <- seq(0, 2, by = .001); n <- length(t)
c4 <- sin(2 * pi * notes4[1] * t); e4 <- sin(2 * pi * notes4[3] * t);
g4 <- sin(2 * pi * notes4[5] * t)
```

```
chord <- c4 + e4 + g4 + rnorm(n, 0, 0.3)
x <- sapply(notes4, function(freq) sin(2 * pi * freq * t))
fit <- lm(chord ~ x - 1)
```




```
##(How you would really do it)
a <- fft(chord); plot(Re(a)^2, type = "l")
```



mas info: transformada discreta de Fourier

ir más allá con estas cosas, sería eso, modelos lineales generalizados y datos longitudinales multinivel(audio) # swirl

Factores de inflación de varianza

En el modelado, nuestro interés radica en representaciones parsimoniosas e interpretables de los datos que mejoran nuestra comprensión de los fenómenos en estudio. La omisión de variables da como resultado un sesgo en los coeficientes de interés, a menos que sus regresores no estén correlacionados con los omitidos. Por otro lado, la inclusión de cualquier variable nueva aumenta los errores estándar (reales, no estimados) de otros regresores. Por tanto, no queremos introducir variables en el modelo sin hacer nada. Esta lección trata sobre el segundo de estos dos temas, que se conoce como inflación de varianza.

Usaremos simulaciones para ilustrar la inflación de la varianza. El código fuente para estas simulaciones está en un archivo llamado vifSims.R que he copiado en su directorio de trabajo y he intentado mostrarlo en su editor de código fuente. Si no pude mostrarlo, debe abrirlo manualmente.

```
makelms <- function(x1, x2, x3){
  # Simulate a dependent variable, y, as x1
  # plus a normally distributed error of mean 0 and
  # standard deviation .3.
  y <- x1 + rnorm(length(x1), sd = .3)
```

```
  # Find the coefficient of x1 in 3 nested linear
  # models, the first including only the predictor x1,
  # the second x1 and x2, the third x1, x2, and x3.
  c(coef(lm(y ~ x1))[2],
    coef(lm(y ~ x1 + x2))[2],
    coef(lm(y ~ x1 + x2 + x3))[2])
}

# Regressor generation process 1.
rgp1 <- function(){
  print("Processing. Please wait.")
  # number of samples per simulation
  n <- 100
  # number of simulations
  nosim <- 1000
  # set seed for reproducibility
  set.seed(4321)
  # Point A
  x1 <- rnorm(n)
  x2 <- rnorm(n)
  x3 <- rnorm(n)
  # Point B
  betas <- sapply(1 : nosim, function(i)makelms(x1, x2, x3))
  round(apply(betas, 1, var), 5)
}

# Regressor generation process 2.
rgp2 <- function(){
  print("Processing. Please wait.")
  # number of samples per simulation
  n <- 100
  # number of simulations
  nosim <- 1000
  # set seed for reproducibility
  set.seed(4321)
  # Point C
  x1 <- rnorm(n)
  x2 <- x1/sqrt(2) + rnorm(n) /sqrt(2)
  x3 <- x1 * 0.95 + rnorm(n) * sqrt(1 - 0.95^2)
  # Point D
  betas <- sapply(1 : nosim, function(i)makelms(x1, x2, x3))
  round(apply(betas, 1, var), 5)
}
```

Encuentre la función, makelms, en la parte superior de vifSims.R. La expresión final en makelms crea 3 modelos lineales. El primero, lm (y ~ x1), predice y en términos de x_1, el segundo predice y en términos de x_1 y x_2, el tercero en términos de los tres regresores. El segundo coeficiente de cada modelo, por ejemplo coef (lm (y ~ x1)) [2], se extrae y se devuelve en un vector de 3 longitudes. ¿Qué representa este segundo coeficiente?. El coeficiente de x_1, En makelms, la variable dependiente simulada, y, depende de x_1

La función rgp1 () calcula la varianza en las estimaciones del coeficiente de x1 en cada uno de los tres modelos, y ~ x1, y ~ x1 + x2 e y ~ x1 + x2 + x3. (Los resultados se redondean a 5 lugares decimales para una visualización conveniente). Esta simulación se aproxima a la varianza (es decir, el error estándar al cuadrado) del coeficiente de x1 en cada uno de estos tres modelos. Recuerde que la inflación de la varianza

se debe a regresores correlacionados y que en rgp1 () los regresores no están correlacionados. Ejecute la simulación rgp1 () ahora. Se paciente. Se tarda un poco.

```
rgp1()
```

```
## [1] "Processing. Please wait."
```

```
##      x1      x1      x1
## 0.00110 0.00111 0.00112
```

Las varianzas en cada uno de los tres modelos son aproximadamente iguales, como se esperaba, ya que los otros regresores, x2 y x3, no están correlacionados con el regresor de interés, x1. Sin embargo, en rgp2 (), x2 y x3 dependen de x1, por lo que deberíamos esperar un efecto. De las expresiones que asignan x2 y x3 que siguen al punto C, ¿cuál está más fuertemente correlacionada con x1?,R=x_3

```
rgp2()
```

```
## [1] "Processing. Please wait."
```

```
##      x1      x1      x1
## 0.00110 0.00240 0.00981
```

En este caso, la inflación de la varianza debido a regresores correlacionados es clara y es más pronunciada en el tercer modelo, y ~ x1 + x2 + x3, ya que x3 es el regresor más fuertemente correlacionado con x1.

En estas dos simulaciones, teníamos 1000 muestras de coeficientes estimados, por lo que pudimos calcular la varianza de la muestra para ilustrar el efecto. En un caso real, solo tenemos un conjunto de coeficientes y dependemos de estimaciones teóricas. Sin embargo, las estimaciones teóricas contienen una constante desconocida de proporcionalidad. Por lo tanto, dependemos de razones de estimaciones teóricas llamadas factores de inflación de varianza o VIF.

Un factor de inflación de varianza (VIF) es una razón de varianzas estimadas, la varianza debida a incluir el i-ésimo regresor, dividida por la debida a incluir un regresor ideal correspondiente que no está correlacionado con los demás. Los VIF se pueden calcular directamente, pero el paquete car proporciona un método conveniente para el propósito, como ilustraremos utilizando los datos suizos del paquete de conjuntos de datos.

Según su documentación, el conjunto de datos suizos consiste en una medida de fertilidad estandarizada e indicadores socioeconómicos para cada una de las 47 provincias de habla francesa de Suiza alrededor de 1888, cuando las tasas de fertilidad suizas comenzaron a caer. Escriba head (swiss) o View (swiss) para examinar los datos

```
data(swiss)
head(swiss)
```

```
##              Fertility Agriculture Examination Education Catholic
## Courtelary        80.2        17.0          15        12     9.96
## Delemont          83.1        45.1           6         9    84.84
## Franches-Mnt      92.5        39.7           5         5    93.40
## Moutier           85.8        36.5          12         7    33.77
## Neuveville        76.9        43.5          17        15     5.16
## Porrentruy        76.1        35.3           9         7    90.57
##              Infant.Mortality
```

```
## Courtelary               22.2
## Delemont                 22.2
## Franches-Mnt             20.2
## Moutier                  20.3
## Neuveville               20.6
## Porrentruy               26.6
```

Se pensaba que la fertilidad dependía de cinco factores socioeconómicos: el porcentaje de hombres que trabajaban en agricultura, el porcentaje de reclutas que recibieron la calificación más alta en el examen del ejército, el porcentaje de reclutas con educación más allá de la escuela primaria, el porcentaje de la población que era católica romana. , y la tasa de Mortalidad Infantil en la provincia. Utilice la regresión lineal para modelar la fertilidad en términos de estos cinco regresores y una intersección. Almacene el modelo en una variable llamada mdl.

```
mdl <- lm(Fertility ~ ., swiss)
```

Calcular los VIF para cada uno de los regresores.

```
library(car)
vif(mdl)
```

```
##      Agriculture      Examination        Education         Catholic
##         2.284129         3.675420         2.774943         1.937160
## Infant.Mortality
##         1.107542
```

Estos VIF muestran, para cada coeficiente de regresión, la inflación de la varianza debido a la inclusión de todos los demás. Por ejemplo, la varianza en el coeficiente estimado de Educación es 2.774943 veces lo que podría haber sido si la Educación no estuviera correlacionada con los otros regresores. Dado que es probable que la educación y la puntuación en un examen estén correlacionados, podemos suponer que la mayor parte de la inflación de la varianza para Educación se debe a que se incluye el examen.

Haga un segundo modelo lineal de fertilidad en el que se omita el examen, pero se incluyen los otros cuatro regresores. Almacene el resultado en una variable llamada mdl2.

```
mdl2 <- lm(Fertility ~ . -Examination, swiss)
 vif(mdl2)
```

```
##      Agriculture        Education         Catholic Infant.Mortality
##         2.147153         1.816361         1.299916         1.107528
```

Como se esperaba, omitir el examen ha disminuido notablemente el VIF para educación, de 2.774943 a 1.816361. Tenga en cuenta que omitir el examen casi no ha tenido ningún efecto en el VIF para la mortalidad infantil. Es probable que el examen y la mortalidad infantil no estén fuertemente correlacionados. Ahora, antes de terminar esta lección, repasemos varios puntos importantes.

Un VIF describe el aumento en la varianza de un coeficiente debido a la correlación de su regresor con los otros regresores. ¿Cuál es la relación de un VIF con el error estándar de su coeficiente? R=VIF es el cuadrado de la inflación del error estándar.

Si un regresor está fuertemente correlacionado con otros y, por lo tanto, aumentará sus VIF, ¿por qué no deberíamos simplemente excluirlo? R: Excluirlo podría sesgar las estimaciones de coeficientes de los regresores con los que está correlacionado.

Los problemas de inflación de la varianza y el sesgo debido a regresores excluidos involucran regresores correlacionados. Sin embargo, existen métodos, como el análisis factorial o el análisis de componentes principales, que pueden convertir regresores en un conjunto equivalente no correlacionado. ¿Por qué entonces, al modelar, no deberíamos simplemente usar regresores no correlacionados y evitar todos los problemas? R: El uso de regresores convertidos puede dificultar la interpretación.

salidas binarias

Con frecuencia nos preocupamos por los resultados que tienen dos valores, como vivo o muerto, ganar o perder, éxito o fracaso. Estos resultados se denominan binarios, Bernoulli o 0/1. Una colección de resultados binarios intercambiables para los mismos datos de covariables se denomina resultados binomiales. (Los resultados son intercambiables si su orden no importa).

En esta unidad usaremos glm () para modelar un proceso con un resultado binario y un predictor continuo. También aprenderemos cómo interpretar los coeficientes glm y cómo encontrar intervalos de confianza. Pero primero, hablemos de las probabilidades.

Los Baltimore Ravens son un equipo de la Liga de fútbol americano. En el juego de postemporada (campeonato) ganan aproximadamente 2/3 de sus juegos. En otras palabras, ganan aproximadamente el doble de lo que pierden. Si quisiera apostar por ellos, tendría que ofrecer probabilidades de 2 a 1; si perdieran, te pagaría $2, pero si ganaran, me pagarías solo $1. De esa manera, a la larga, en muchas apuestas, ambos esperaríamos ganar tanto dinero como perdimos.

Durante la temporada regular, los Ravens ganan alrededor del 55% de sus juegos. ¿Qué probabilidades tendría que ofrecer en la temporada regular? R=55 a 45

Ahora suponga que queremos ver cómo las probabilidades de los Ravens dependen de su ofensiva. En otras palabras, queremos modelar cómo p, o alguna función de él, depende de cuántos puntos sean capaces de anotar los Ravens. Por supuesto, no podemos observar p, solo podemos observar las victorias, las derrotas y las puntuaciones asociadas. Aquí hay un diagrama de caja del valor de una temporada de tales observaciones.

Podemos ver que los Ravens tienden a ganar más cuando suman más puntos. De hecho, alrededor de 3/4 de sus pérdidas están en o por debajo de un cierto puntaje y aproximadamente 3/4 de sus victorias están en o por encima de él. ¿De qué puntuación estoy hablando? (Recuerde que los recuadros violetas representan el 50% de las muestras y las "T" el 25%).R=23

Hubo 9 juegos en los que los Ravens anotaron 23 puntos o menos. Ganaron 4 de estos juegos, por lo que podríamos suponer que su probabilidad de ganar, dado que obtienen 23 puntos o menos, es de aproximadamente 1/2.

Hubo 11 juegos en los que los Ravens anotaron 24 puntos o más. Ganaron todos menos uno de estos. Verifique esto comprobando los datos usted mismo. Está en un marco de datos llamado ravenData. Mírelo escribiendo ravenData o View (ravenData).

```
library(dplyr)
ravenData<-select(ravensData, ravenWinNum, ravenWin, ravenScore)
ravenData
```

```
##    ravenWinNum ravenWin ravenScore
## 1            1        W         24
## 2            1        W         38
## 3            1        W         28
## 4            1        W         34
## 5            1        W         44
## 6            0        L         23
## 7            1        W         31
```

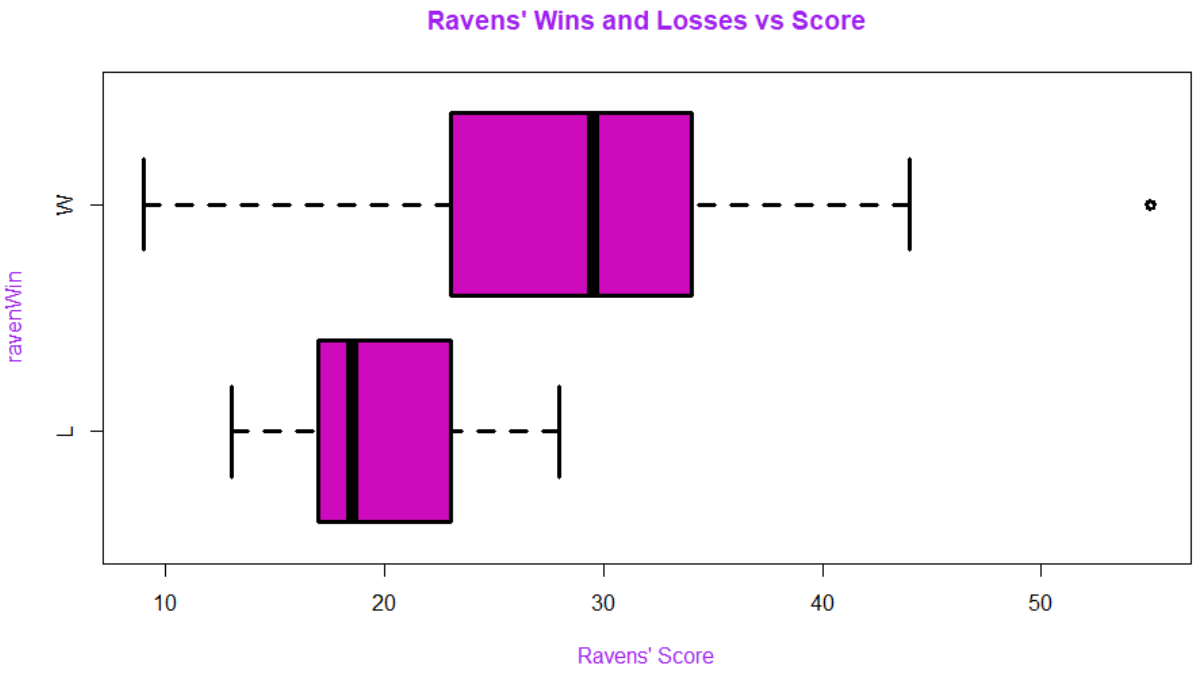


Figure 1: A caption

```
## 8          1       W       23
## 9          1       W        9
## 10         1       W       31
## 11         0       L       13
## 12         1       W       25
## 13         1       W       55
## 14         1       W       13
## 15         1       W       16
## 16         0       L       20
## 17         0       L       28
## 18         0       L       17
## 19         1       W       33
## 20         0       L       17
```

Vemos una transición bastante rápida en el récord de victorias / derrotas de los Ravens entre 23 y 28 puntos. Con 23 puntos y menos ganan aproximadamente la mitad de sus juegos, entre 24 y 28 puntos ganan 3 de 4, y por encima de 28 puntos los ganan todos. De esto, obtenemos una idea muy cruda de la correspondencia entre los puntos anotados y la probabilidad de ganar. Obtenemos una curva en forma de S, un graffiti S de todos modos.

Por supuesto, esperaríamos que una curva real fuera más suave. Por ejemplo, no esperaríamos que los Ravens ganen la mitad de los juegos en los que anotaron cero puntos, ni todos los juegos en los que anotaron más de 28. Un modelo lineal generalizado que tiene estas propiedades supone que las probabilidades logarítmicas de una victoria depende linealmente de la puntuación. Es decir,$log(\frac{p}{1-p}) = \beta_0 + \beta_1 * puntuacion$ La función de enlace, $log(\frac{p}{1-p})$, se llama logit, y el proceso de encontrar los mejores β_0, β_1 se llama regresión logística.

Los "mejores" β_0, β_1 son aquellos que maximizan la probabilidad del récord real de victorias / derrotas. Con base en la puntuación de un juego, β_0, β_1 nos dan un logaritmo de probabilidades, que podemos convertir

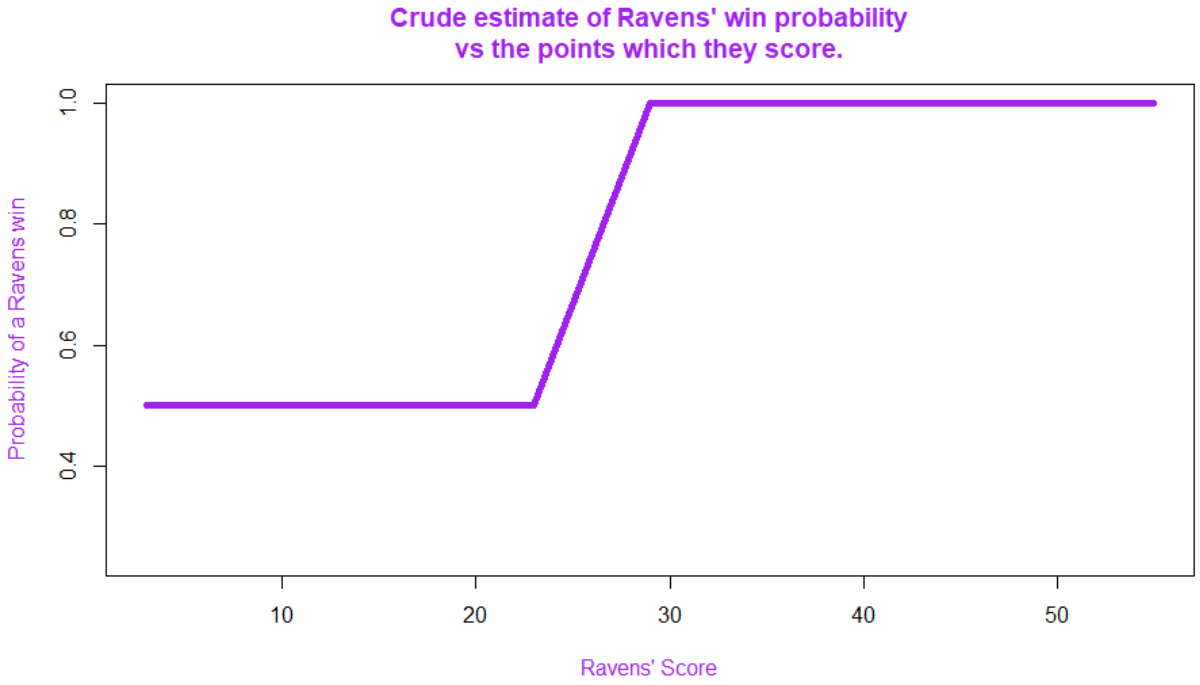


Figure 2: A caption

en una probabilidad, p, de ganar. Nos gustaría que p sea alto para los puntajes de los juegos ganadores y bajo para los puntajes de las derrotas.

Podemos usar la función glm () de R para encontrar β_0, β_1 h que maximizan la probabilidad de nuestras observaciones. Volviendo al marco de datos, queremos predecir los resultados binarios, ravenWinNum, a partir de los puntos anotados, ravenScore. Esto corresponde a la fórmula, ravenWinNum ~ ravenScore, que es el primer argumento de glm. El segundo argumento, familia, describe los resultados, que en nuestro caso son binomiales. El tercer argumento son los datos, ravenData. Llame a glm con estos parámetros y almacene el resultado en una variable llamada mdl.

```
mdl <- glm(ravenWinNum ~ ravenScore, binomial, ravenData)
```

Las probabilidades estimadas por regresión logística usando glm () están representadas por la curva negra. Es más razonable que nuestra estimación bruta en varios aspectos: aumenta sin problemas con la puntuación, estima que 15 puntos dan a los Ravens un 50% de posibilidades de ganar, que 28 puntos les dan un 80% de posibilidades y que 55 puntos dan lugar a una victoria. muy probable (98%) pero no absolutamente seguro.

El modelo es menos creíble con puntuaciones inferiores a 9. Por supuesto, no hay datos en esa región; los Ravens anotaron al menos 9 puntos en cada juego. El modelo les da un 33% de posibilidades de ganar si obtienen 9 puntos, lo que puede ser razonable, pero también les da un 16% de posibilidades de ganar incluso si no obtienen puntos. Podemos usar la función predict () de R para ver las estimaciones del modelo para puntajes más bajos. La función tomará mdl y un marco de datos de puntuaciones como argumentos y devolverá las probabilidades de registro para las puntuaciones dadas. Llame a predict (mdl, data.frame (ravenScore = c (0, 3, 6))) y almacene el resultado en una variable llamada lodds.

```
lodds <- predict(mdl, data.frame(ravenScore=c(0, 3, 6)))
```

Como predecir () nos da logaritmos de probabilidades, tendremos que convertir a probabilidades. Para

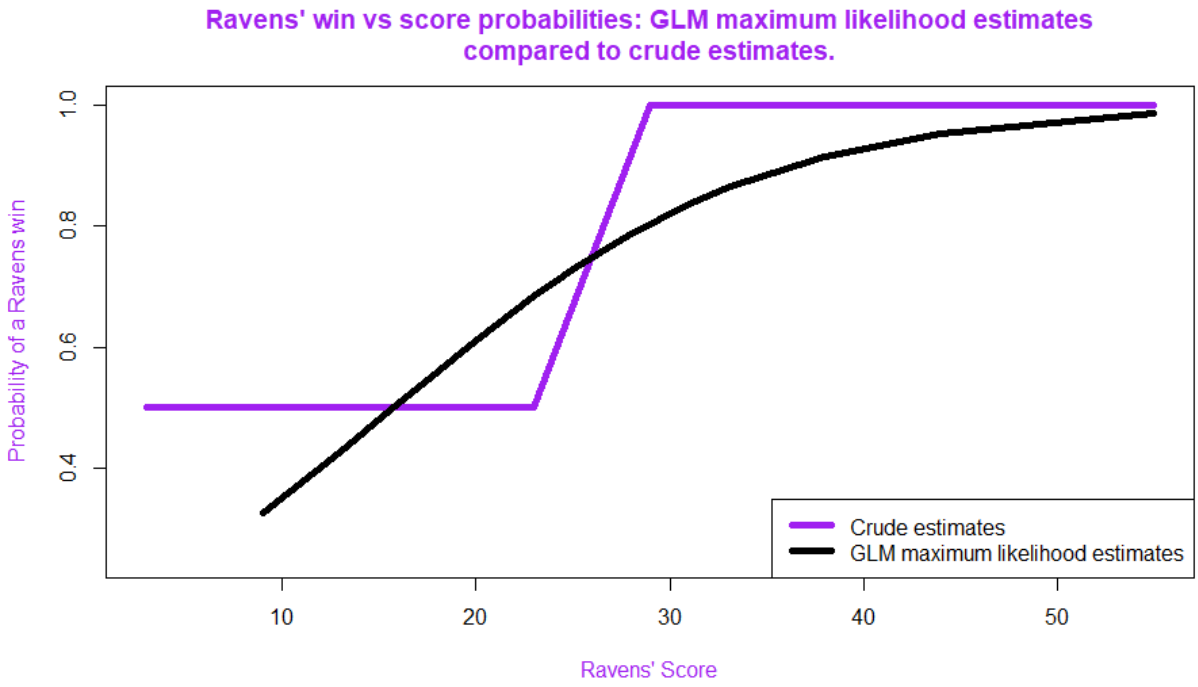


Figure 3: A caption

convertir las probabilidades de registro en probabilidades, use exp (lodds) / (1 + exp (lodds)). No se moleste en almacenar el resultado en una variable. No lo necesitaremos.

```
exp(lodds)/(1+exp(lodds))
```

```
##         1         2         3
## 0.1570943 0.2041977 0.2610505
```

Como puede ver, una persona podría ganar mucho dinero apostando contra este modelo. Cuando los Ravens no obtienen puntos, al modelo le podrían gustar probabilidades de 16 a 84. Sin embargo, resulta que el modelo no está tan seguro de sí mismo. Al escribir resumen (mdl), puede ver que los coeficientes estimados están dentro de 2 errores estándar de cero. Mira el resumen ahora.

```
summary(mdl)
```

```
##
## Call:
## glm(formula = ravenWinNum ~ ravenScore, family = binomial, data = ravenData)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7575  -1.0999   0.5305   0.8060   1.4947
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.68001    1.55412  -1.081     0.28
```

```
## ravenScore   0.10658    0.06674   1.597     0.11
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 24.435  on 19  degrees of freedom
## Residual deviance: 20.895  on 18  degrees of freedom
## AIC: 24.895
##
## Number of Fisher Scoring iterations: 5
```

Los coeficientes estiman las probabilidades logarítmicas como una función lineal de los puntos obtenidos. Tienen una interpretación natural en términos de probabilidades porque, si $\beta_0 + \beta_1 * puntuacion$ estima el registro de probabilidades, entonces $e^{\beta_0+\beta_1*puntuacion} = e^{\beta_0} * e^{\beta_1*puntuacion}$ estima las probabilidades. Así e^{β_0} es la probabilidad de ganar con una puntuación de 0 (en nuestro caso 16/84) y e^{β_1} es el factor por el cual las probabilidades de ganar aumentan con cada punto anotado. En nuestro caso $e^{\beta_1} = e^{0.10658} = 1.11$. En otras palabras, las probabilidades de ganar aumentan en un 11% por cada punto anotado.

Sin embargo, los coeficientes tienen errores estándar relativamente grandes. Un intervalo de confianza del 95% es aproximadamente 2 errores estándar a cada lado de un coeficiente. La función confint () de R encontrará los límites inferior y superior exactos de los intervalos de confianza del 95% para los coeficientes b0 y b1. Para obtener los intervalos correspondientes para exp (b0) y exp (b1), simplemente exponencializaríamos la salida de confint (mdl). Hacer esto ahora.

```
exp(confint(mdl))
```

```
##                   2.5 %   97.5 %
## (Intercept) 0.005674966 3.106384
## ravenScore  0.996229662 1.303304
```

El límite de confianza más bajo en las probabilidades de ganar con una puntuación de 0 es cercano a cero, lo que parece mucho más realista que la cifra de 16/84 del modelo de máxima verosimilitud. Ahora mire el límite inferior de exp (b1), el coeficiente exponenciado de ravenScore. ¿Cómo sugiere que las probabilidades de ganar se verán afectadas por cada punto adicional anotado?

El límite de confianza más bajo en exp (b1) sugiere que las probabilidades de ganar disminuirían ligeramente con cada punto adicional anotado. Evidentemente, esto no es realista. Por supuesto, los intervalos de confianza se basan en suposiciones de muestras grandes y nuestra muestra consta de solo 20 juegos. De hecho, la versión GLM del análisis de varianza mostrará que si ignoramos los puntajes por completo, no lo hacemos mucho peor.

La regresión lineal minimiza la diferencia al cuadrado entre las observaciones pronosticadas y reales, es decir, minimiza la varianza del residual. Si un predictor adicional reduce significativamente la varianza del residual, el predictor se considera importante. La desviación extiende esta idea a la regresión lineal generalizada, utilizando verosimilitudes logarítmicas (negativas) en lugar de la varianza. (Para obtener una explicación detallada, consulte las diapositivas y conferencias.) Para ver el análisis de la desviación de nuestro modelo, escriba anova (mdl).

```
anova(mdl)
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: ravenWinNum
```

```
##
## Terms added sequentially (first to last)
##
##
##            Df Deviance Resid. Df Resid. Dev
## NULL                           19     24.435
## ravenScore  1   3.5398        18     20.895
```

El valor, 3.5398, etiquetado como la desviación de ravenScore, es en realidad la diferencia entre la desviación de nuestro modelo, que incluye una pendiente, y la de un modelo que incluye solo una intersección, b0. Este valor tiene una distribución central de chi-cuadrado (para muestras grandes) con 1 grado de libertad (2 parámetros menos 1 parámetro, o equivalentemente 19-18). La hipótesis nula es que el coeficiente de ravenScore es cero. Para rechazar con seguridad esta hipótesis, querríamos que 3,5398 fuera mayor que el percentil 95 de la distribución de chi-cuadrado con un grado de libertad. Utilice qchisq (0,95, 1) para calcular el umbral de este percentil.

```
qchisq(0.95, 1)
```

```
## [1] 3.841459
```

Como puede ver, 3.5398 está cerca pero por debajo del umbral del percentil 95, 3.841459, por lo que se consideraría consistente con la hipótesis nula al nivel convencional del 5%. En otras palabras, ravenScore agrega muy poco a un modelo que solo adivina que los Ravens ganan con una probabilidad del 70% (su récord real esa temporada) o las probabilidades de 7 a 3 son casi tan buenas. Si lo desea, puede verificar esto usando mdl0 <- glm (ravenWinNum ~ 1, binomial, ravenData), pero esto concluye el ejemplo de Resultados binarios. Gracias.

salidas contables

Muchos datos toman la forma de recuentos. Pueden ser llamadas a un centro de llamadas, cantidad de casos de gripe en un área o cantidad de automóviles que cruzan un puente. Los datos también pueden estar en forma de tasas, por ejemplo, porcentaje de niños que aprueban una prueba. En esta lección usaremos la regresión de Poisson para analizar las visitas diarias a un sitio web a medida que crece la popularidad del sitio web y para analizar el porcentaje de visitas que se deben a referencias de un sitio diferente.

Las visitas a un sitio web tienden a ocurrir de forma independiente, una a la vez, a un cierto índice promedio. La distribución de Poisson describe procesos aleatorios de este tipo. Un proceso de Poisson se caracteriza por un solo parámetro, la tasa de ocurrencia esperada, que generalmente se denomina λ. En nuestro caso, λ serán visitas esperadas por día. Por supuesto, a medida que el sitio web se vuelva más popular, lambda crecerá. En otras palabras, nuestra lambda *dependerá del tiempo* . Usaremos la regresión de Poisson para modelar esta dependencia.

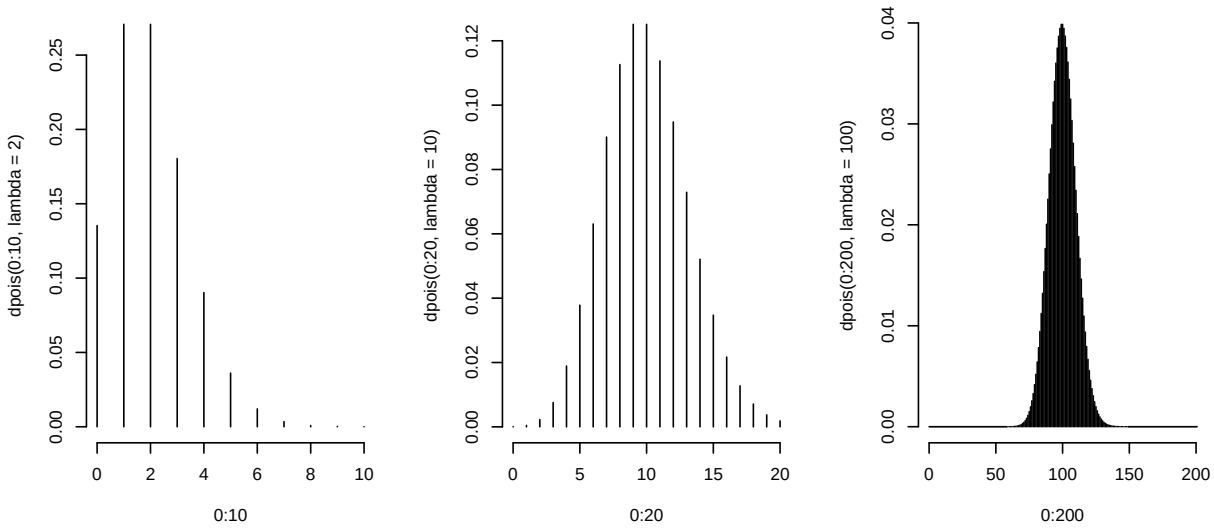
De manera algo notable, la varianza de un proceso de Poisson tiene el mismo valor que su media, λ. Puede ilustrar esto rápidamente generando, digamos, n = 1000 muestras de un proceso de Poisson usando rpois (n, lambda) de R y calculando la varianza de la muestra. Por ejemplo, escriba var (rpois (1000, 50)). La varianza de la muestra no será exactamente igual al valor teórico, por supuesto, pero estará bastante cerca.

```
var(rpois(1000, 50))
```

```
## [1] 50.89962
```

Los recuentos generados por un proceso de Poisson son, estrictamente hablando, ligeramente diferentes de las sumas normalizadas del teorema del límite central. Sin embargo, los recuentos en un período de tiempo dado representarán sumas de un mayor número de términos a medida que aumenta lambda. De hecho, se puede demostrar formalmente que para una lambda grande, una distribución de Poisson se aproxima bien a una normal. La figura ilustra este efecto. Muestra una progresión desde una función de masa de probabilidad de Poisson escasa y asimétrica a la izquierda, a una curva densa en forma de campana a la derecha, ya que lambda varía de 2 a 100.

```
par(mfrow = c(1, 3))
plot(0 : 10, dpois(0 : 10, lambda = 2), type = "h", frame = FALSE)
plot(0 : 20, dpois(0 : 20, lambda = 10), type = "h", frame = FALSE)
plot(0 : 200, dpois(0 : 200, lambda = 100), type = "h", frame = FALSE)
```



En una regresión de Poisson, se supone que el logaritmo de lambda es una función lineal de los predictores. Dado que intentaremos modelar el crecimiento de las visitas a un sitio web, el registro de lambda será una función lineal de la fecha: $log(\lambda) = \beta_0 + \beta_1 * date$. Esto implica que el número medio de visitas por día, lambda, es exponencial en la fecha: $\lambda = exp(\beta_0) * exp(\beta_1)^{fecha}$. El crecimiento exponencial también es sugerido por la curva negra suave dibujada a través de los datos. Por lo tanto, exp (b1) representaría el porcentaje por el cual las visitas crecen por día.

Nuestros datos están en un marco de datos llamado hits.

```
hits<-select(gaData,date ,visits, simplystats)
```

Hay tres columnas de datos etiquetadas como fecha, visitas y simplemente estadísticas, respectivamente. La columna de Simplystats registra el número de visitas que se deben a referencias de otro sitio, el blog Simply Statistics. Regresaremos a esa columna más tarde. Por ahora, nos interesan las columnas de fecha y visitas. La fecha será nuestro predictor.

Nuestras fechas están representadas en términos de la clase R, Fecha. Verifique esto escribiendo class (hits [, 'date']), o algo equivalente.

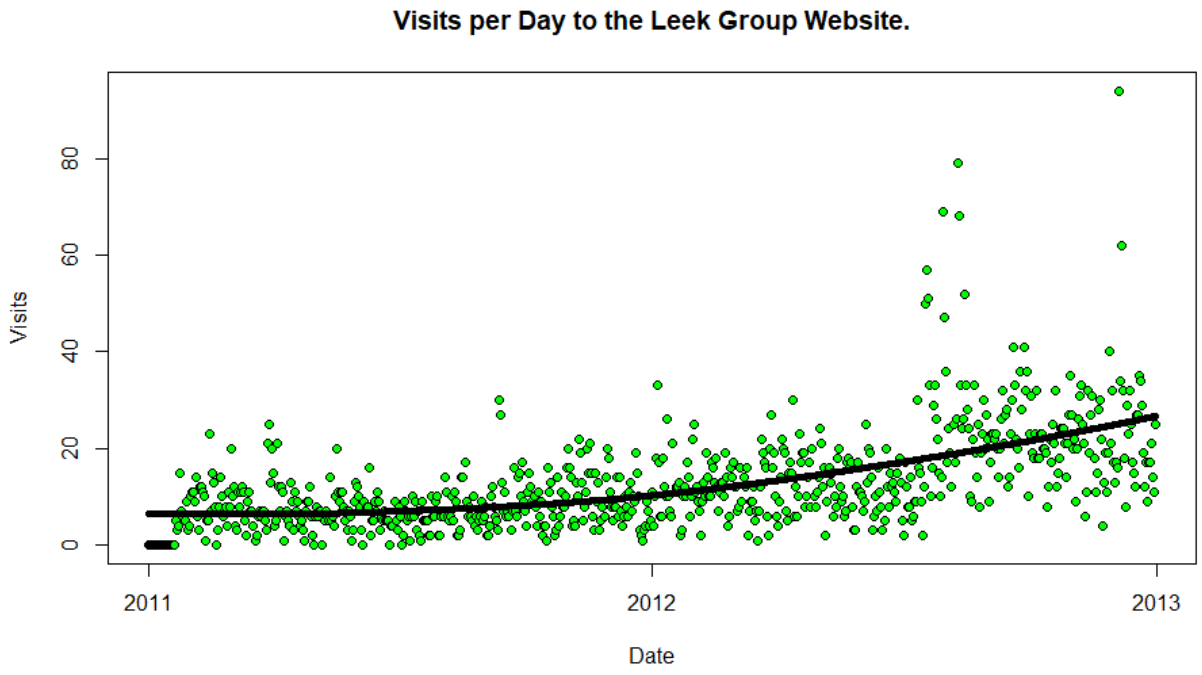


Figure 4: A caption

```
class(hits[,'date'])
```

```
## [1] "Date"
```

La clase Date de R representa las fechas como días desde o antes del 1 de enero de 1970. Son esencialmente números y, hasta cierto punto, pueden tratarse como tales. Las fechas, por ejemplo, se pueden sumar o restar, o fácilmente convertirlas en números.

```
as.integer(head(hits[,'date']))
```

```
## [1] 14975 14976 14977 14978 14979 14980
```

Las propiedades aritméticas de las fechas nos permiten utilizarlas como predictores. Usaremos la regresión de Poisson para predecir $log(\lambda)$ como una función lineal de la fecha de una manera que maximice la probabilidad de los recuentos que realmente vemos. Nuestra fórmula serán visitas ~ fecha. Dado que nuestros resultados (visitas) son recuentos, nuestra familia será 'poisson' y nuestro tercer argumento serán los datos, aciertos. Cree tal modelo y guárdelo en una variable llamada mdl usando la siguiente expresión o algo equivalente, mdl <- glm (visitas ~ fecha, poisson, hits)

```
mdl <- glm(visits ~ date, poisson, hits)
```

La figura sugiere que nuestra regresión de Poisson se ajusta muy bien a los datos. La línea negra es la lambda estimada, o el número medio de visitas por día. Vemos que la media de visitas por día aumentó de alrededor de 5 a principios de 2011 a alrededor de 10 en 2012, ya alrededor de 20 a finales de 2013. Aproximadamente se duplica cada año.

Escriba resumen (mdl) para examinar los coeficientes estimados y su importancia.

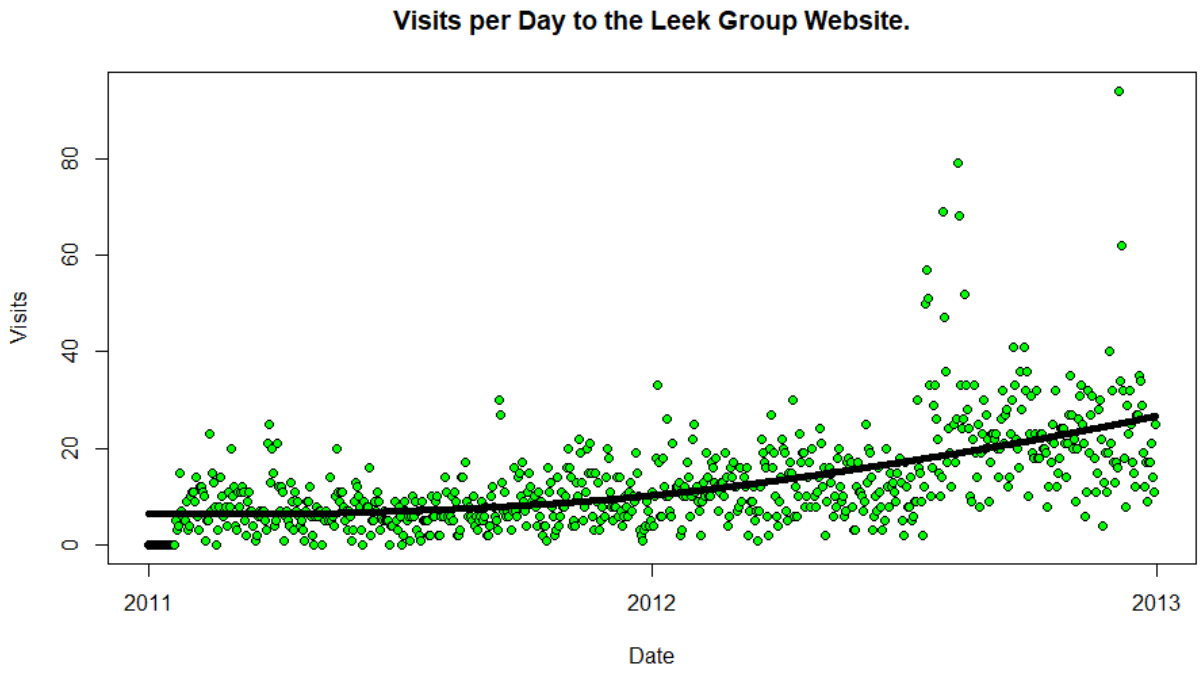


Figure 5: A caption

```
summary(mdl)
```

```
##
## Call:
## glm(formula = visits ~ date, family = poisson, data = hits)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0466  -1.5908  -0.3198   0.9128  10.6545
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.275e+01  8.130e-01  -40.28   <2e-16 ***
## date         2.293e-03  5.266e-05   43.55   <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: 5150.0  on 730  degrees of freedom
## Residual deviance: 3121.6  on 729  degrees of freedom
## AIC: 6069.6
##
## Number of Fisher Scoring iterations: 5
```

Ambos coeficientes son significativos, siendo mucho más de dos errores estándar desde cero. La desviación residual también es significativamente menor que la nula, lo que indica un efecto fuerte. (**Recuerde que**

la diferencia entre la desviación nula y residual es aproximadamente chi-cuadrado con 1 grado de libertad checar). El coeficiente de intersección, b0, solo representa log promedios de aciertos en la fecha 0 de R, es decir, el 1 de enero de 1970. Lo ignoraremos y nos enfocaremos en el coeficiente de fecha, b1, ya que exp (b1) estimará el porcentaje en el que las visitas promedio aumentan por día de la vida del sitio.

Obtenga el intervalo de confianza del 95% para exp (b1) exponenciando confint (mdl, 'date')

```
exp(confint(mdl, 'date'))
```

```
##    2.5 %   97.5 %
## 1.002192 1.002399
```

Se estima que las visitas aumentan en un factor de entre 1,002192 y 1,002399 por día. Es decir, entre 0,2192% y 0,2399% por día. En realidad, esto representa más del doble cada año.

Nuestro modelo parece una descripción bastante buena de los datos, pero ningún modelo es perfecto y, a menudo, podemos aprender sobre un proceso de generación de datos buscando las deficiencias de un modelo. Como se muestra en la figura, una cosa acerca de nuestro modelo es la 'inflación cero' en las dos primeras semanas de enero de 2011, antes de que el sitio tuviera visitas. El modelo sobreestima sistemáticamente el número de visitas durante este tiempo. Una cosa menos obvia es que la desviación estándar de los datos puede estar aumentando con lambda más rápido de lo que permite un modelo de Poisson. Esta posibilidad se puede ver en el gráfico de la derecha comparando visualmente la extensión de los puntos verdes con la desviación estándar predicha por el modelo (guiones negros). Además, hay cuatro o cinco ráfagas de popularidad durante las cuales el número de visitas supera con creces las dos desviaciones sobre el promedio. Quizás esto se deba a menciones en otro sitio.

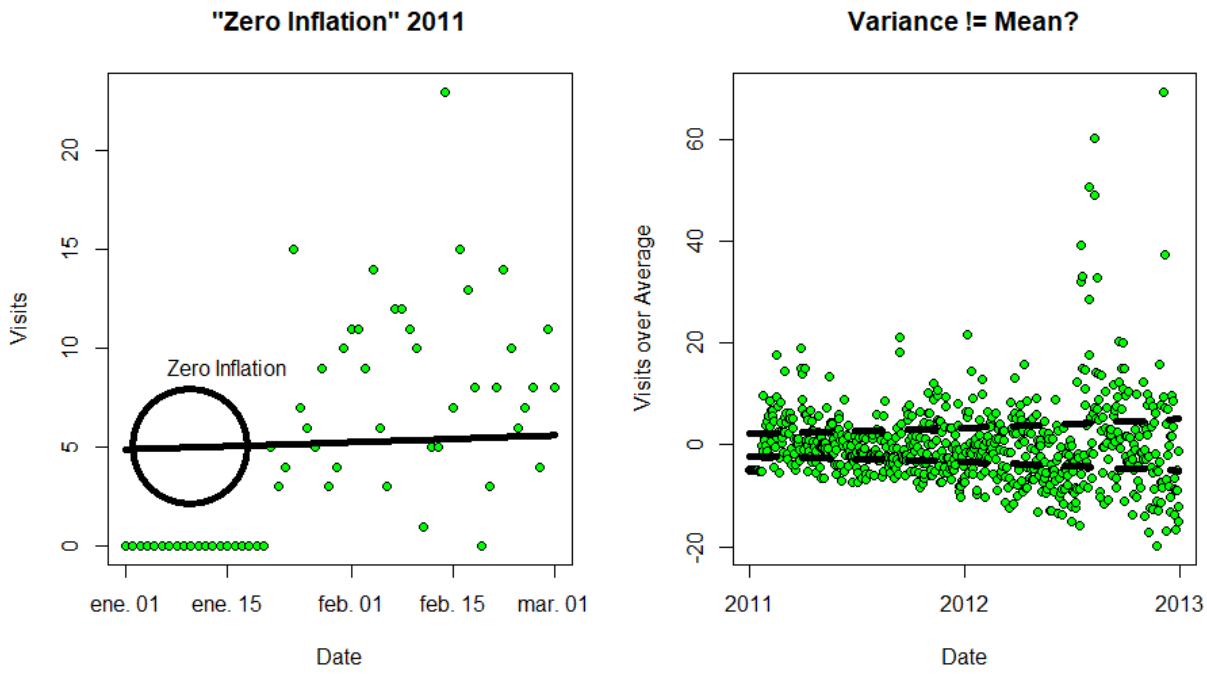


Figure 6: A caption

Parece que al menos algunos de ellos lo son. La columna de Simplystats de nuestros datos registra el número de visitas al sitio de Leek Group que provienen del sitio relacionado, Simply Statistics. (Es decir, visitas debido a clics en un enlace al Grupo Leek que apareció en una publicación de Simply Statisics).

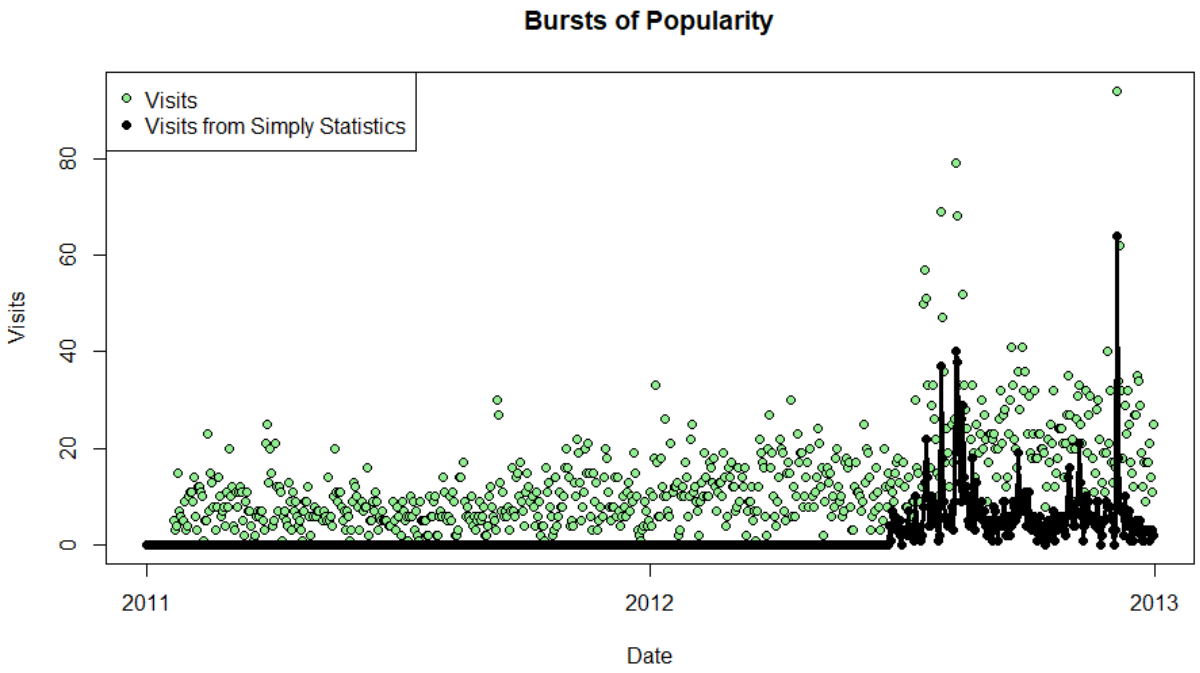


Figure 7: A caption

En la figura, el número máximo de visitas se produjo a finales de 2012. Las visitas del blog Simply Statistics también alcanzaron su máximo ese día. Para encontrar la fecha exacta podemos usar which.max (hits [, 'visitas']). Hacer esto ahora.

```
 a<-which.max(hits[,'visits'])
hits[a,]
```

```
##           date visits simplystats
## 704 2012-12-04     94          64
```

El número máximo de visitas, 94, se produjo el 4 de diciembre de 2012, de las cuales 64 provinieron del blog Simply Statistics. Podríamos considerar las 64 visitas como un evento especial, por encima de lo normal. ¿Se puede atribuir la diferencia, 94-64 = 30 visitas, al tráfico normal estimado por nuestro modelo? Para verificar, necesitaremos el valor de lambda el 4 de diciembre de 2012. Esta será la entrada 704 del elemento ajustado.values de nuestro modelo. Extraiga mdl $ fit.values [704] y guárdelo en una variable denominada lambda.

```
lambda <- mdl$fitted.values[704]
```

El número de visitas explicadas por nuestro modelo el 4 de diciembre de 2012 son las de una variable aleatoria de Poisson con lambda media. Podemos encontrar el percentil 95 de esta distribución usando qpois (.95, lambda). Prueba esto ahora.

```
qpois(.95, lambda)
```

```
## [1] 33
```

Entonces, el 95% de las veces veríamos 33 visitas o menos, por lo que 30 visitas no serían raras según nuestro modelo. Parecería que el 4 de diciembre de 2012, el altísimo número de visitas se debió a referencias de Simply Statistics. Para medir la importancia de las referencias de Simply Statistics, es posible que deseemos modelar la proporción de tráfico que representan dichas referencias. Hacerlo también ilustrará el uso del parámetro de glm, offset, para modelar frecuencias y proporciones.

Un proceso de Poisson genera recuentos y los recuentos son números enteros, 0, 1, 2, 3, etc. Una proporción es una fracción. Entonces, ¿cómo puede un proceso de Poisson modelar una proporción? El truco consiste en incluir el denominador de la fracción, o más precisamente su logaritmo, como compensación. Recuerde que en nuestro conjunto de datos, 'Simplystats' son las visitas de Simply Statistics, y 'Visitas' es el número total de visitas. Nos gustaría modelar la fracción simplemente estadísticas / visitas, pero para evitar la división por cero usaremos simplemente estadísticas / (visitas + 1). Un modelo de Poisson supone que log (lambda) es una combinación lineal de predictores. Supongamos que asumimos que log (lambda) = log (visitas + 1) + b0 + b1 * fecha. En otras palabras, si insistimos en que el coeficiente de log (visitas + 1) sea igual a 1, estamos prediciendo el logaritmo de visitas medias de Simply Statistics como una proporción del total de visitas: log (lambda / (visitas + 1)) = b0 + b1 * fecha.

El parámetro de glm, offset, tiene precisamente este efecto. Fija el coeficiente de la compensación en 1. Para crear un modelo para la proporción de visitas de Simply Statistics, dejamos offset = log (visitas + 1). Cree un modelo de Poisson ahora y guárdelo como una variable llamada mdl2.

```
mdl2 <- glm(formula = simplystats ~ date, family = poisson, data = hits, offset = log(visits + 1))
```

Aunque el resumen (mdl2) mostrará que los coeficientes estimados son significativamente diferentes de cero, el modelo en realidad no es impresionante. Podemos ilustrar por qué mirando el 4 de diciembre de 2012, una vez más. Ese día hubo 64 visitas reales de Simply Statistics. Sin embargo, según mdl2, 64 visitas serían extremadamente improbables. Puede verificar esta debilidad en el modelo encontrando el percentil 95 de mdl2 para ese día. Recordando que el 4 de diciembre de 2012 fue la muestra 704, encuentre qpois (.95, mdl2 $ valores ajustados [704]).

```
qpois(.95, mdl2$fitted.values[704])
```

```
## [1] 47
```

Sobreajuste y desajuste

La lección de factores de inflación de varianza demostró que la inclusión de nuevas variables aumentará los errores estándar de las estimaciones de coeficientes de otros regresores correlacionados. Por lo tanto, no queremos lanzar variables sin hacer nada al modelo. Por otro lado, la omisión de variables da como resultado un sesgo en los coeficientes de regresores que se correlacionan con los omitidos. En esta lección demostramos el efecto de las variables omitidas y discutimos el uso de ANOVA para construir representaciones parsimoniosas e interpretables de los datos.

Primero, me gustaría ilustrar cómo la omisión de un regresor correlacionado puede sesgar las estimaciones de un coeficiente. El código fuente relevante está en un archivo llamado fiting.R que he copiado en su directorio de trabajo y he intentado mostrarlo en su editor de código fuente. Si no pude mostrarlo, debe abrirlo manualmente.

```
simbias <- function(seed=8765){
  # The default seed guarantees a nice histogram. This is the only
  # reason that accepting the default, x1c <- simbias(), is required in the lesson.
  # The effect will be evident with other seeds as well.
  set.seed(seed)
```

```
  temp <- rnorm(100)
  # Point A
  x1 <- (temp + rnorm(100))/sqrt(2)
  x2 <- (temp + rnorm(100))/sqrt(2)
  x3 <- rnorm(100)
  # Function to simulate regression of y on 2 variables.
  f <- function(k){
    # Point B
    y <- x1 + x2 + x3 + .3*rnorm(100)
    # Point C
    c(lm(y ~ x1 + x2)$coef[2],
       lm(y ~ x1 + x3)$coef[2])
  }
  # Point D
  sapply(1:150, f)
}

# Illustrate the effect of bogus regressors on residual squared error.
bogus <- function(){
  temp <- swiss
  # Add 41 columns of random regressors to a copy of the swiss data.
  for(n in 1:41){temp[,paste0("random",n)] <- rnorm(nrow(temp))}
  # Define a function to compute the deviance of Fertility regressed
  # on all regressors up to column n. The function, deviance(model), computes
  # the residual sum of squares of the model given as its argument.
  f <- function(n){deviance(lm(Fertility ~ ., temp[,1:n]))}
  # Apply f to data from n=6, i.e., the legitimate regressors,
  # through n=47, i.e., a full complement of bogus regressors.
  rss <- sapply(6:47, f)
  # Display result.
  plot(0:41, rss, xlab="Number of bogus regressors.", ylab="Residual squared error.",
       main="Residual Squared Error for Swiss Data\nUsing Irrelevant (Bogus) Regressors",
       pch=21, bg='red')
}

# Plot histograms illustrating bias in estimates of a regressor
# coefficient 1) when an uncorrelated regressor is missing and
# 2) when a correlated regressor is missing.
x1hist <- function(x1c){
  p1 <- hist(x1c[1,], plot=FALSE)
  p2 <- hist(x1c[2,], plot=FALSE)
  yrange <- c(0, max(p1$counts, p2$counts))
  plot(p1, col=rgb(0,0,1,1/4), xlim=range(x1c), ylim=yrange, xlab="Estimated coefficient of x1",
        main="Bias Effect of Omitted Regressor")
  plot(p2, col=rgb(1,0,0,1/4), xlim=range(x1c), ylim=yrange, add=TRUE)
  legend(1.1, 40, c("Uncorrelated regressor, x3, omitted", "Correlated regressor, x2, omitted"),
         fill=c(rgb(0,0,1,1/4), rgb(1,0,0,1/4)))
}
```

x1 y x2 estan correlacionados

Dentro de simbias () se define otra función, f (n). Forma una variable dependiente, y, y en el punto C devuelve el coeficiente de x1 estimado por dos modelos, y ~ x1 + x2 e y ~ x1 + x3. Falta un regresor en cada modelo. En la expresión para y (punto B), ¿cuál es el coeficiente real de x1?

```
x1c <- simbias()
```

El coeficiente real de x1 es 1. Habiendo sido advertido de que omitir un regresor correlacionado sesgaría las estimaciones del coeficiente de x1, esperaríamos que la estimación media de la segunda fila de x1c esté más lejos de 1 que la media de la primera fila de x1c. Usando apply (x1c, 1, mean), encuentre las medias de cada fila.

```
apply(x1c, 1, mean)
```

```
##       x1       x1
## 1.034403 1.476944
```

Se muestran los histogramas de estimaciones de la primera fila (azul) y la segunda fila (rojo) de x1c. Las estimaciones de la segunda fila son claramente más de dos desviaciones estándar del valor correcto de 1, y el sesgo debido a la omisión del regresor correlacionado es evidente. (El código que produjo esta figura es secundario a la lección, pero está disponible como la función x1hist (), en la parte inferior del ajuste. R.)

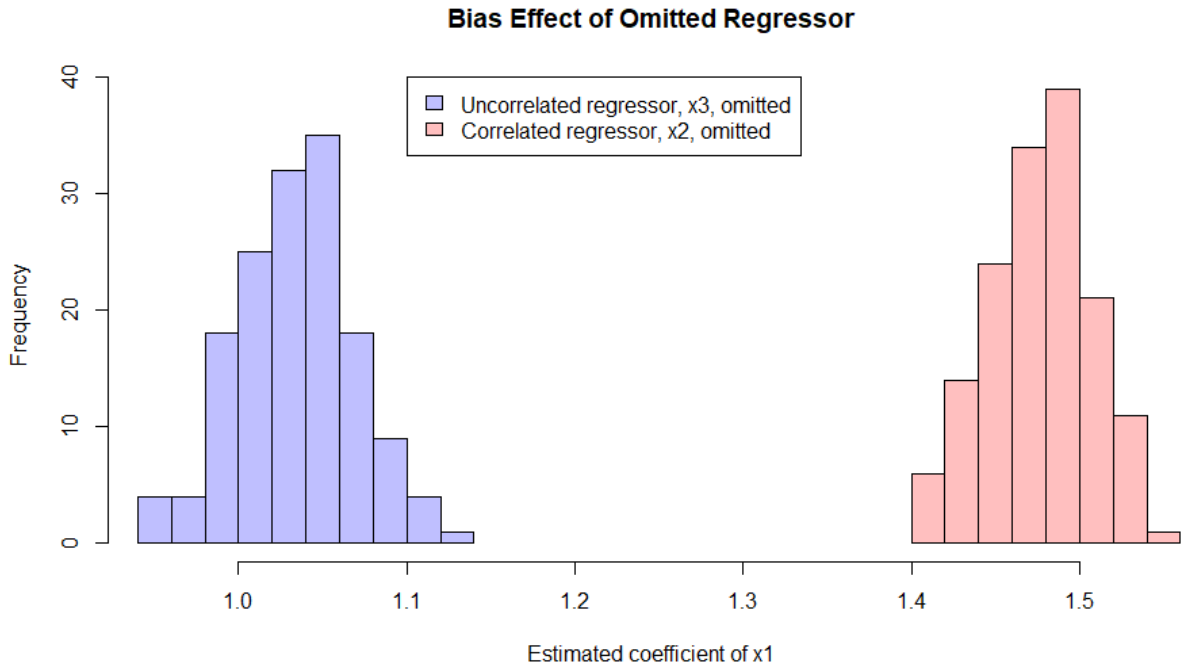


Figure 8: A caption

Agregar incluso regresores irrelevantes puede hacer que un modelo tienda a un ajuste perfecto. Ilustramos esto agregando regresores aleatorios a los datos suizos y retrocediendo progresivamente en más de ellos. A medida que el número de regresores se acerca al número de puntos de datos (47), la suma de cuadrados residual, también conocida como la desviación, se acerca a 0. (El código fuente de esta figura se puede encontrar como función falsa () en el ajuste.

En la figura, agregar regresores aleatorios disminuyó la desviación, pero sería un error creer que tales disminuciones son significativas. Para evaluar la importancia, debemos tener en cuenta que la adición de regresores reduce los grados de libertad residuales. El análisis de varianza (ANOVA) es una forma útil de cuantificar la importancia de regresores adicionales. Para ejemplificar su uso, usaremos los datos suizos.

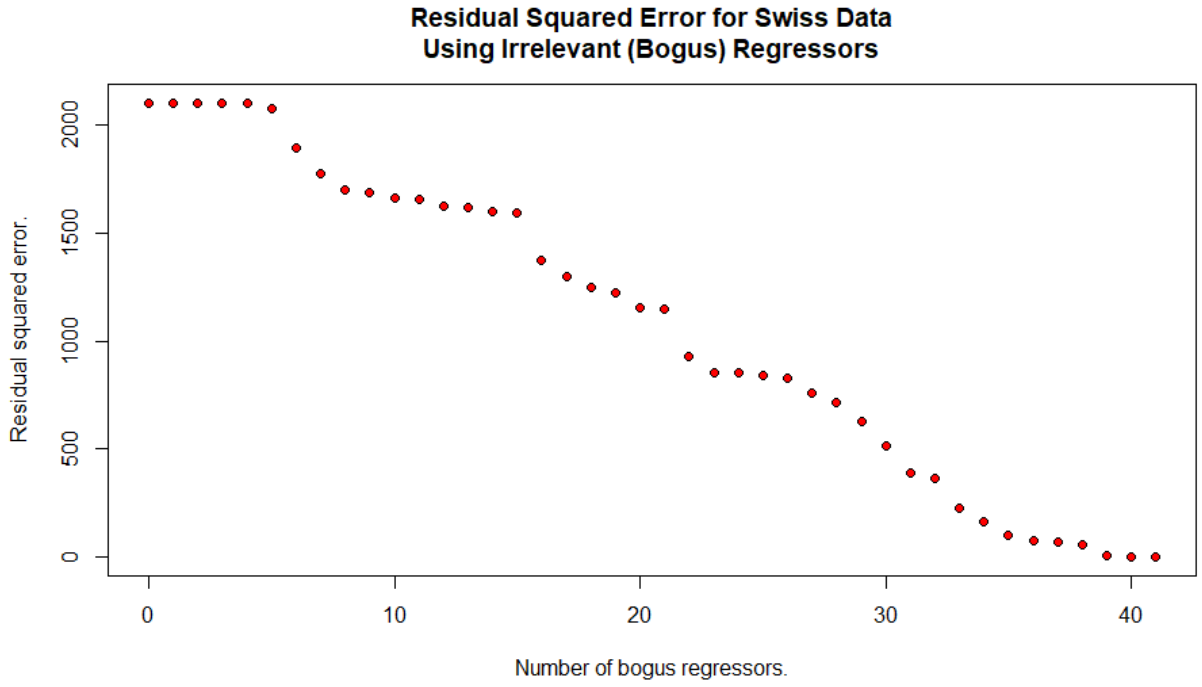


Figure 9: A caption

Recuerde que el conjunto de datos suizos consiste en una medida estandarizada de fertilidad e indicadores socioeconómicos para cada una de las 47 provincias francófonas de Suiza en 1888. Se pensaba que la fertilidad dependía de una intersección y cinco factores denotados como Agricultura, Examen, Educación, Católica y Mortalidad infantil. Para comenzar nuestro ejemplo de ANOVA, haga una regresión de la fertilidad en agricultura y almacene el resultado en una variable denominada ajuste1

```
fit1 <- lm(Fertility ~ Agriculture, swiss)
```

Cree otro modelo, denominado fit3, mediante la regresión de la fertilidad en agricultura y dos regresores adicionales, examen y educación.

```
fit3 <- lm(Fertility ~ Agriculture + Examination + Education, swiss)
```

Ahora usaremos anova para evaluar la importancia de los dos regresores agregados. La hipótesis nula es que los regresores agregados no son significativos. Lo explicaremos en detalle en breve, pero ahora solo aplique la prueba de significancia ingresando anova (ajuste1, ajuste3).

```
anova(fit1, fit3)
```

```
## Analysis of Variance Table
##
## Model 1: Fertility ~ Agriculture
## Model 2: Fertility ~ Agriculture + Examination + Education
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1     45 6283.1
## 2     43 3180.9  2    3102.2 20.968 4.407e-07 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Los tres asteriscos, ***, en la parte inferior derecha de la tabla impresa indican que la hipótesis nula se rechaza en el nivel 0.001, por lo que al menos uno de los dos regresores adicionales es significativo. El rechazo se basa en una prueba F de cola derecha, Pr (> F), aplicada a un valor F.

Una estadística F es una razón de dos sumas de cuadrados divididas por sus respectivos grados de libertad. Si las dos sumas escaladas son independientes y centralmente distribuidas en chi-cuadrado con la misma varianza, el estadístico tendrá una distribución F con parámetros dados por los dos grados de libertad. En nuestro caso, las dos sumas son sumas residuales de cuadrados que, como sabemos, tienen una media cero, por lo tanto, son chi-cuadrado centralmente siempre que los propios residuos estén distribuidos normalmente. Las dos sumas relevantes se dan en la columna RSS (suma de cuadrados residual) de la tabla. Qué so 6283.1 and 3180.9

La función de R, desviance (modelo), calcula la suma residual de cuadrados, también conocida como desviación, del modelo lineal dado como argumento. Usando la desviación (ajuste3), verifique que 3180.9 es la suma de cuadrados residual de ajuste3. (Por supuesto, fit3 se llama Modelo 2 en la tabla).

```
deviance(fit3)
```

```
## [1] 3180.925
```

En los siguientes pasos, mostraremos cómo calcular el valor F, 20.968, que aparece en la tabla impresa por anova (). Comenzaremos con el denominador, que es la suma de cuadrados residual de fit3 dividida por sus grados de libertad. Fit3 tiene 43 grados de libertad residuales. Esta cifra se obtiene restando 4, el número de predictores de fit3 (el 3 nombrado y el intercepto) de 47, el número de muestras en suizo. Almacene el valor de la desviación (ajuste3) / 43 en una variable denominada d.

```
d <- deviance(fit3)/43
```

El numerador es la diferencia, desviación (ajuste1) -desviación (ajuste3), dividida por la diferencia en los grados residuales de libertad de ajuste1 y ajuste3, es decir, 2. Este cálculo requiere una justificación teórica que omitimos, pero la idea esencial es que fit3 tiene 2 predictores además de los de fit1. Calcule el numerador y guárdelo en una variable llamada n.

```
n <- (deviance(fit1) - deviance(fit3))/2
```

```
 n/d
```

```
## [1] 20.96783
```

Ahora calcularemos el valor p, que es la probabilidad de que un valor de n / d o mayor se extraiga de una distribución F que tiene los parámetros 2 y 43. Este valor se dio como 4.407e-07 en la columna etiquetada Pr (> F) en la tabla impresa por anova (), un valor muy poco probable si la hipótesis nula fuera cierta. Calcule este valor p usando pf (n / d, 2, 43, lower.tail = FALSE)

```
pf (n / d, 2, 43, lower.tail = FALSE)
```

```
## [1] 4.406913e-07
```

Con base en el valor p calculado, es extremadamente improbable un falso rechazo de la hipótesis nula. Estamos seguros de que el ajuste3 es significativamente mejor que el ajuste1, con una salvedad: el análisis de varianza es sensible a la suposición de que los residuos del modelo son aproximadamente normales. Si no es así, podríamos obtener un valor p pequeño por esa razón. Por lo tanto, vale la pena probar la normalidad de los residuos. La prueba de Shapiro-Wilk es rápida y fácil en R. La normalidad es su hipótesis nula. Utilice shapiro.test (fit3 $ residuals) para probar el residual de fit3.

```
shapiro.test (fit3 $ residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  fit3$residuals
## W = 0.97276, p-value = 0.336
```

El valor p de Shapiro-Wilk de 0,336 no rechaza la normalidad, lo que respalda la confianza en nuestro análisis de varianza. Para ilustrar el uso de anova () con más de dos modelos, he construido fit5 y fit6 usando los primeros 5 y los 6 regresores (incluida la intersección) respectivamente. Por tanto, fit1, fit3, fit5 y fit6 forman una secuencia anidada de modelos; los regresores de uno se incluyen en los del siguiente. Ingrese anova (fit1, fit3, fit5, fit6) en el indicador R ahora para obtener el sabor.

```
fit5 <- lm(Fertility ~ Agriculture + Examination + Education+Catholic, swiss)
fit6 <- lm(Fertility ~ Agriculture + Examination + Education+Catholic+Infant.Mortality, swiss)
anova(fit1, fit3, fit5, fit6)
```

```
## Analysis of Variance Table
##
## Model 1: Fertility ~ Agriculture
## Model 2: Fertility ~ Agriculture + Examination + Education
## Model 3: Fertility ~ Agriculture + Examination + Education + Catholic
## Model 4: Fertility ~ Agriculture + Examination + Education + Catholic +
##     Infant.Mortality
##   Res.Df    RSS Df Sum of Sq       F    Pr(>F)
## 1     45 6283.1
## 2     43 3180.9  2   3102.19 30.2107 8.638e-09 ***
## 3     42 2513.8  1    667.13 12.9937 0.0008387 ***
## 4     41 2105.0  1    408.75  7.9612 0.0073357 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Parece que cada modelo es una mejora significativa con respecto a su predecesor