Regresion 6

Diego VZ

2023-05-09

PREGUNTA 1

```
library(faraway)

# Cargamos el dataset
data(happy)

# Estandarizamos las variables predictoras
happy_std <- scale(happy[, -1])

# Modelo de regresión sin escalar
model1 <- lm(happy ~ ., data = happy)

# Modelo de regresión escalado
model <- lm(happy ~ happy_std, data = happy)

# Resumen del modelo
summary(model)

##
## Call:
## lm(formula = happy ~ happy_std, data = happy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7186 -0.5779 -0.1172  0.6340  2.0651
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     6.74359    0.16948  39.790  < 2e-16 ***
## happy_stdmoney  0.33741    0.18364   1.837   0.0749 .
## happy_stdsex   -0.06967    0.19569  -0.356   0.7240
## happy_stdlove   1.23336    0.18986   6.496 1.97e-07 ***
## happy_stdwork   0.48214    0.20193   2.388   0.0227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.058 on 34 degrees of freedom
## Multiple R-squared:  0.7102, Adjusted R-squared:  0.6761
## F-statistic: 20.83 on 4 and 34 DF,  p-value: 9.364e-09

summary(model1)
```

```
##
## Call:
## lm(formula = happy ~ ., data = happy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7186 -0.5779 -0.1172  0.6340  2.0651
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.072081   0.852543  -0.085   0.9331
## money        0.009578   0.005213   1.837   0.0749 .
## sex         -0.149008   0.418525  -0.356   0.7240
## love         1.919279   0.295451   6.496 1.97e-07 ***
## work         0.476079   0.199389   2.388   0.0227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.058 on 34 degrees of freedom
## Multiple R-squared:  0.7102, Adjusted R-squared:  0.6761
## F-statistic: 20.83 on 4 and 34 DF,  p-value: 9.364e-09
```

Al comparar los resultados de ambos modelos, podemos ver que los coeficientes y los niveles de significancia son los mismos para ambas versiones del modelo. Esto se debe a que la escala de las variables predictoras no afecta la relación entre las variables predictoras y la variable de respuesta. Sin embargo, el uso de variables predictoras escaladas puede hacer que la interpretación del modelo sea más sencilla. En el modelo escalado, los coeficientes representan el cambio en la variable de respuesta asociado con un cambio de una desviación estándar en la variable predictora correspondiente. Esto puede hacer que sea más fácil comparar la importancia relativa de cada variable predictora en el modelo. En resumen, ambos modelos proporcionan información similar sobre la relación entre las variables predictoras y la variable de respuesta. Sin embargo, el modelo escalado puede hacer que la interpretación del modelo sea más sencilla y puede facilitar la comparación de la importancia relativa de cada variable predictora en el modelo.

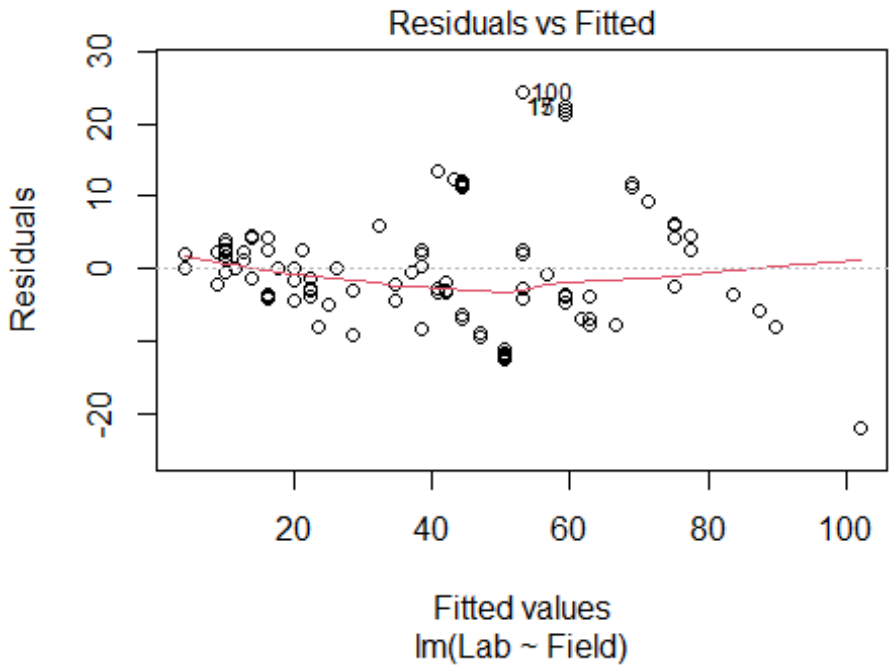
PREGUNTA 2

```
model <- lm(Lab ~ Field, data = pipeline)
summary(model)
```

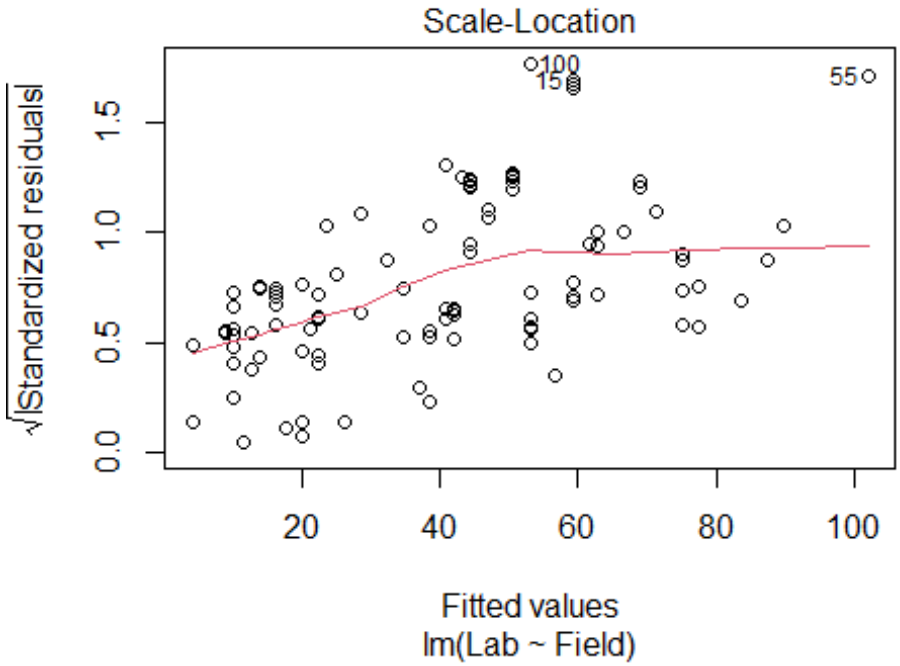
```
##
## Call:
## lm(formula = Lab ~ Field, data = pipeline)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.985  -4.072  -1.431   2.504  24.334
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.96750    1.57479  -1.249    0.214
## Field        1.22297    0.04107  29.778   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.865 on 105 degrees of freedom
## Multiple R-squared:  0.8941, Adjusted R-squared:  0.8931
## F-statistic: 886.7 on 1 and 105 DF,  p-value: < 2.2e-16
```

```
plot(model, which = 1) # Gráfico de residuos versus valores ajustados
```

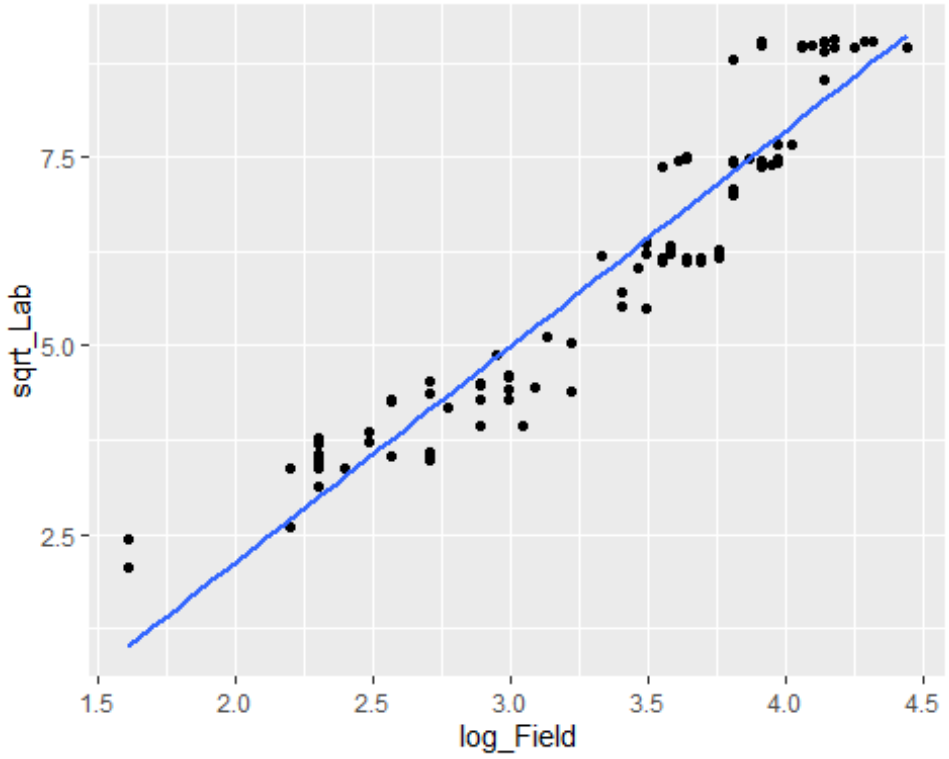



```
plot(model, which = 3) # Gráfico de residuos versus valores de la
variable predictora
```




```
#Según la salida de la función summary(model), no hay evidencia de un problema de varianza no constante en el modelo de regresión lineal. El error estándar residual es 7.865, lo cual es relativamente pequeño en comparación con el rango de la variable de respuesta (Lab). Además, se puede utilizar el gráfico de residuos versus valores ajustados para verificar la varianza no constante. Si la dispersión de los residuos es aproximadamente constante en todos los valores de los valores ajustados, entonces no hay evidencia de varianza no constante. Si la dispersión de los residuos aumenta o disminuye a medida que aumentan los valores ajustados, entonces puede haber un problema de varianza no constante.

#Observando los plots vemos que no se da el caso y que la dispersión de los residuos es aproximadamente constante.

# Ordenar los datos por la variable predictora "Field"
i <- order(pipeline$Field)
npipe <- pipeline[i,]

# Dividir los datos en grupos y calcular la media de "Field" y la varianza de "Lab" dentro de cada grupo
ff <- gl(12, 9)[-108]
meanfield <- unlist(lapply(split(npipe$Field, ff), mean))
varlab <- unlist(lapply(split(npipe$Lab, ff), var))

# Regresar el logaritmo de la varianza de "Lab" en función del logaritmo de la media de "Field"
```

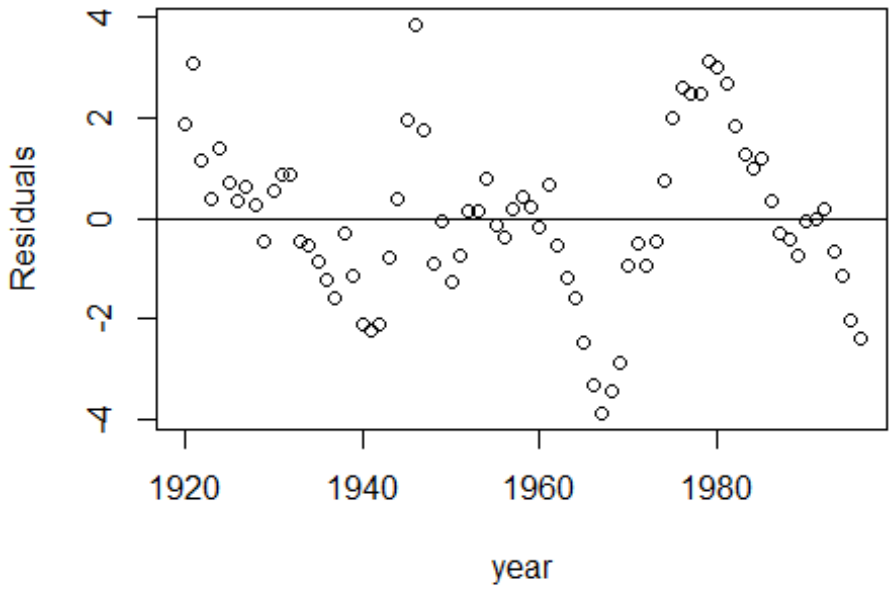
```r
logmeanfield <- log(meanfield)
logvarlab <- log(varlab)
fit <- lm(logvarlab ~ logmeanfield)
a0 <- exp(fit$coefficients[1])
a1 <- fit$coefficients[2]

# Calcular los pesos para cada observación
weights <- a0 * (pipeline$Field)^a1

# Ajustar el modelo de regresión lineal ponderada (WLS) de "Lab" en
función de "Field"
model_wls <- lm(Lab ~ Field, data = pipeline, weights = weights)

# Mostrar el resumen del modelo WLS
summary(model_wls)

##
## Call:
## lm(formula = Lab ~ Field, data = pipeline, weights = weights)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -206.43  -25.11  -10.54   10.56  173.45
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.07787    2.49792  -0.031    0.975
## Field        1.18057    0.05185  22.768   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.67 on 105 degrees of freedom
## Multiple R-squared:  0.8316, Adjusted R-squared:   0.83
## F-statistic: 518.4 on 1 and 105 DF,  p-value: < 2.2e-16

library(ggplot2)

# Transformar las variables
pipeline$log_Field <- log(pipeline$Field)
pipeline$sqrt_Lab <- sqrt(pipeline$Lab)

# Crear el gráfico de dispersión
ggplot(pipeline, aes(x = log_Field, y = sqrt_Lab)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)

## `geom_smooth()` using formula = 'y ~ x'
```



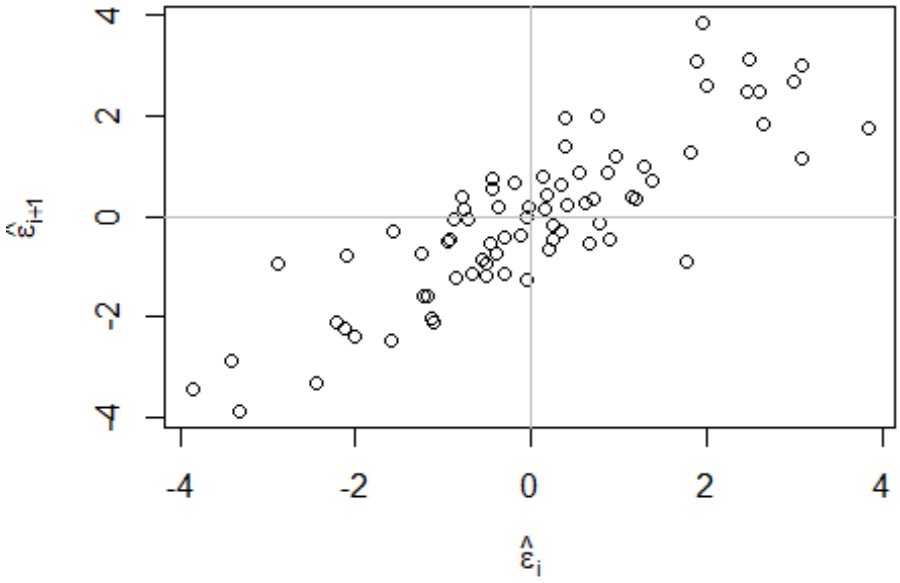
PREGUNTA 3

```
# Ajustar el modelo de regresión
lmod<-lm(divorce~unemployed+femlab+marriage+birth+military, data=divusa)
summary(lmod)

##
## Call:
## lm(formula = divorce ~ unemployed + femlab + marriage + birth +
##     military, data = divusa)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8611 -0.8916 -0.0496  0.8650  3.8300
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.48784    3.39378   0.733   0.4659
## unemployed  -0.11125    0.05592  -1.989   0.0505 .
## femlab       0.38365    0.03059  12.543  < 2e-16 ***
## marriage     0.11867    0.02441   4.861 6.77e-06 ***
## birth       -0.12996    0.01560  -8.333 4.03e-12 ***
## military    -0.02673    0.01425  -1.876   0.0647 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.65 on 71 degrees of freedom
```

```
## Multiple R-squared:  0.9208, Adjusted R-squared:  0.9152
## F-statistic: 165.1 on 5 and 71 DF,  p-value: < 2.2e-16

# Realizar los gráficos de verificación de errores correlacionados
plot(residuals(lmod) ~ year, na.omit(divusa), ylab="Residuals")
abline(h=0)
```




```
n <- length(residuals(lmod))
plot(tail(residuals(lmod),n-1) ~ head(residuals(lmod),n-1), xlab=
expression(hat(epsilon)[i]),ylab=expression(hat(epsilon)[i+1]))
abline(h=0,v=0,col=grey(0.75))
```




```
cor(residuals(lmod)[-1],residuals(lmod)[-length(residuals(lmod))])
```

```
## [1] 0.8469792
```

```
# Hay una larga secuencia de puntos por encima y por debajo de la línea.
Los gráficos sugieren una correlación serial positiva y una tendencia
lineal entre residuos sucesivos. Existe una correlación de 0,847 entre
residuos sucesivos.

# Ajustar el modelo GLS con estructura de correlación AR(1)
library(nlme)
glmod <- gls(divorce~unemployed+femlab+marriage+birth+military,
data=divusa, correlation=corAR1(form=~year), method="ML")
summary(glmod)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: divorce ~ unemployed + femlab + marriage + birth + military
##   Data: divusa
##        AIC      BIC    logLik
##   179.9523 198.7027 -81.97613
##
## Correlation Structure: AR(1)
##  Formula: ~year
##  Parameter estimate(s):
##       Phi
## 0.9715486
##
```

```
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) -7.059682  5.547193 -1.272658  0.2073
## unemployed   0.107643  0.045915  2.344395  0.0219
## femlab       0.312085  0.095151  3.279878  0.0016
## marriage     0.164326  0.022897  7.176766  0.0000
## birth       -0.049909  0.022012 -2.267345  0.0264
## military     0.017946  0.014271  1.257544  0.2127
##
##  Correlation:
##             (Intr) unmply femlab marrig birth
## unemployed -0.420
## femlab     -0.802  0.240
## marriage   -0.516  0.607  0.307
## birth      -0.379  0.041  0.066 -0.094
## military   -0.036  0.436 -0.311  0.530  0.128
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.4509327 -0.9760939 -0.6164694  1.1375377  2.1593261
##
## Residual standard error: 2.907665
## Degrees of freedom: 77 total; 71 residual

intervals(glmod, which="var-cov")

## Approximate 95% confidence intervals
##
##  Correlation structure:
##         lower      est.     upper
## Phi 0.6527537 0.9715486 0.9980196
##
##  Residual standard error:
##     lower      est.     upper
##  0.797364  2.907665 10.603078

summary(glmod)

## Generalized least squares fit by maximum likelihood
##   Model: divorce ~ unemployed + femlab + marriage + birth + military
##   Data: divusa
##        AIC      BIC    logLik
##   179.9523 198.7027 -81.97613
##
## Correlation Structure: AR(1)
##  Formula: ~year
##  Parameter estimate(s):
##       Phi
## 0.9715486
##
## Coefficients:
```

```
##                 Value Std.Error   t-value p-value
## (Intercept) -7.059682  5.547193 -1.272658  0.2073
## unemployed   0.107643  0.045915  2.344395  0.0219
## femlab       0.312085  0.095151  3.279878  0.0016
## marriage     0.164326  0.022897  7.176766  0.0000
## birth       -0.049909  0.022012 -2.267345  0.0264
## military     0.017946  0.014271  1.257544  0.2127
##
##  Correlation:
##             (Intr) unmply femlab marrig birth
## unemployed -0.420
## femlab     -0.802  0.240
## marriage   -0.516  0.607  0.307
## birth      -0.379  0.041  0.066 -0.094
## military   -0.036  0.436 -0.311  0.530  0.128
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -1.4509327 -0.9760939 -0.6164694  1.1375377  2.1593261
##
## Residual standard error: 2.907665
## Degrees of freedom: 77 total; 71 residual
```

 # La correlación estimada (phi) es 0,97154. Esto es muy alto. Además, el intervalo de confianza es (0,6528097, 0,9980192), que no contiene 0. Por lo tanto, la correlación AR(1) es significativa.

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

 # Obtener el resumen del modelo GLS

```
 dwtest(divorce~unemployed+femlab+marriage+birth+military, data=divusa)
```

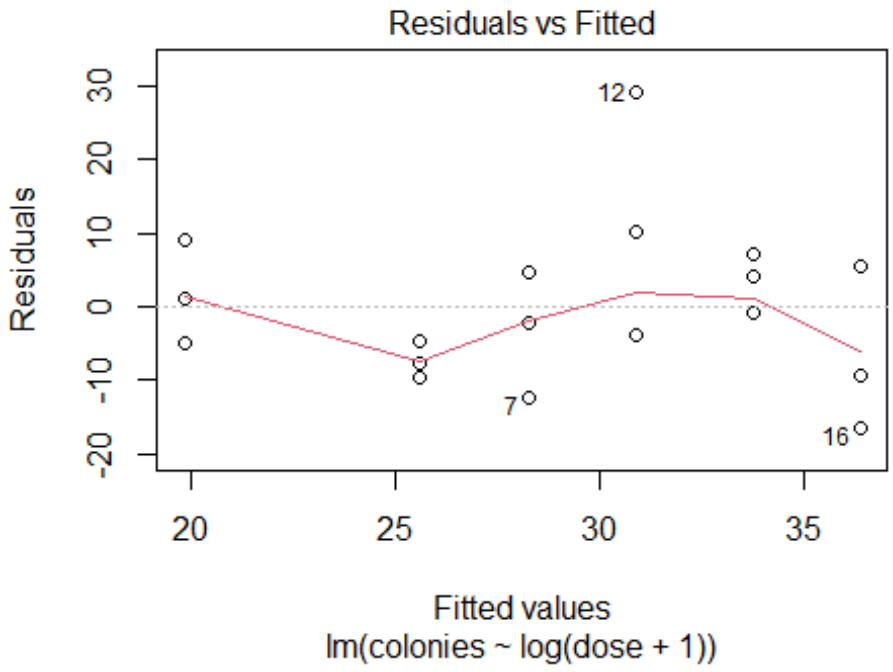
```
##
##  Durbin-Watson test
##
## data:  divorce ~ unemployed + femlab + marriage + birth + military
## DW = 0.29988, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

#El resultado del test de Durbin-Watson indica que el valor del estadístico DW es 0.29988 y el valor p es menor que 2.2e-16, lo que

```
sugiere que hay evidencia significativa de autocorrelación positiva en los residuos del modelo de regresión. La hipótesis alternativa es que la autocorrelación verdadera es mayor que cero.
#El valor del estadístico DW varía entre 0 y 4, y un valor cercano a 0 indica autocorrelación positiva, mientras que un valor cercano a 4 indica autocorrelación negativa. En este caso, el valor de DW es muy cercano a 0, lo que sugiere que hay autocorrelación positiva en los residuos.
#Por lo tanto, se puede concluir que el modelo de regresión lineal simple no es adecuado para estos datos, ya que no tiene en cuenta la autocorrelación positiva en los residuos. Es necesario ajustar un modelo que tenga en cuenta la autocorrelación, como un modelo de regresión con errores autorregresivos (AR) o un modelo de series de tiempo.
```

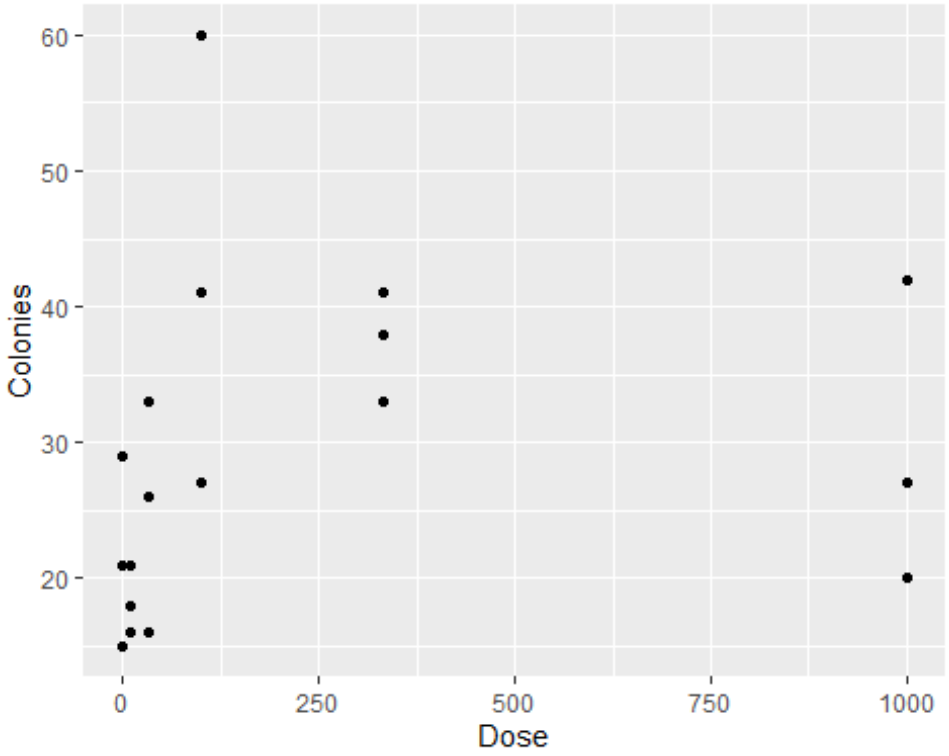
PREGUNTA 4

```
# Fit a linear model with colonies as the response and log(dose+1) as the predictor
model <- lm(colonies ~ log(dose+1), data = salmonella)

# Check for lack of fit
plot(model, which = 1)
```

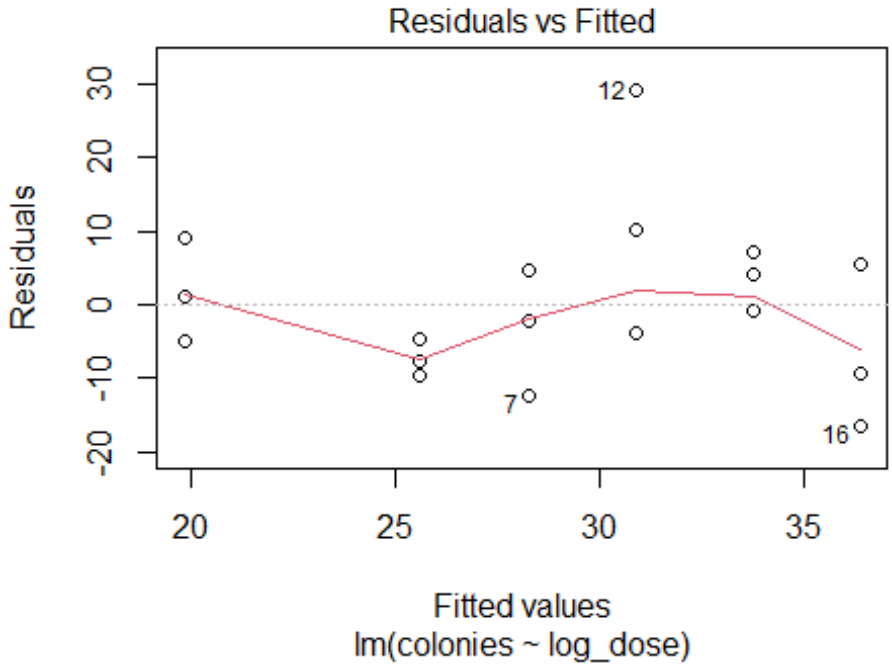



```
summary(aov(model))

##               Df Sum Sq Mean Sq F value Pr(>F)
## log(dose + 1)  1  530.7   530.7   4.514 0.0495 *
## Residuals     16 1881.1   117.6
```

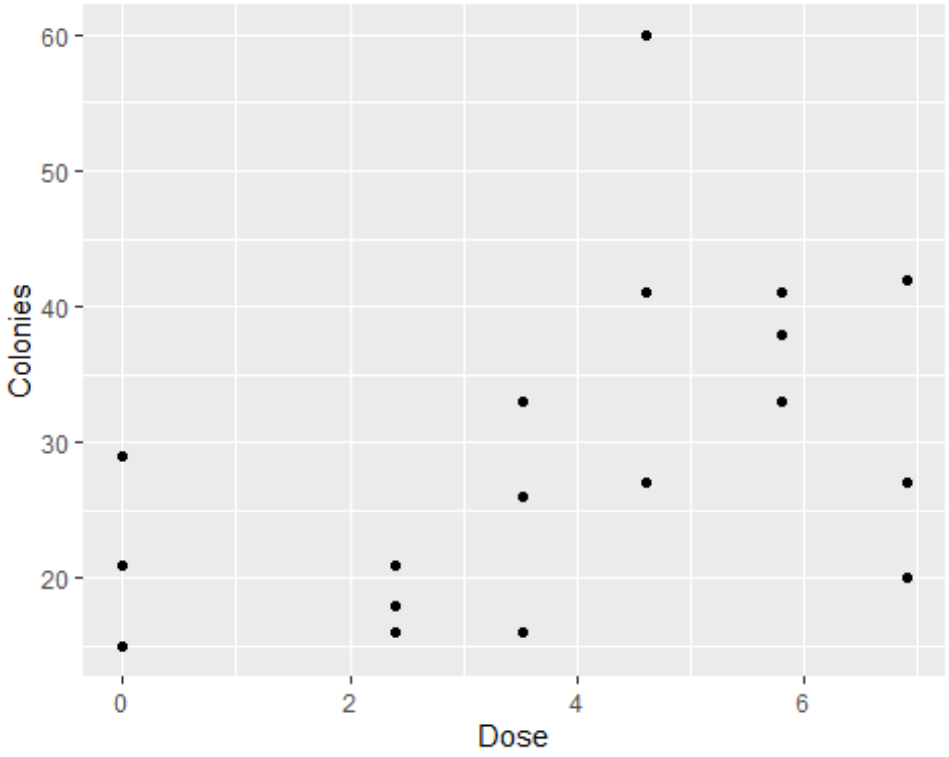
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Create a scatter plot of colonies vs. dose
ggplot(salmonella, aes(x = dose, y = colonies)) +
geom_point() +
labs(x = "Dose", y = "Colonies")
```




```
salmonella$log_dose <- log(salmonella$dose + 1)
# Fit a linear model with colonies as the response and log_dose as the
predictor
model <- lm(colonies ~ log_dose, data = salmonella)

# Check for lack of fit
plot(model, which = 1)
```




```
summary(aov(model))

##             Df Sum Sq Mean Sq F value Pr(>F)
## log_dose     1  530.7   530.7   4.514 0.0495 *
## Residuals   16 1881.1   117.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

ggplot(salmonella, aes(x = log_dose, y = colonies)) +
geom_point() +
labs(x = "Dose", y = "Colonies")
```




```
#La aplicación del logaritmo definitivamente mejora la explicación que la
dosis tiene sobre las colonias. Construyamos un modelo utilizando glm.
Recordando que el modelo lineal generalizado (GLM) es una generalización
flexible de la regresión lineal ordinaria que permite variables de
respuesta que tienen modelos de distribución de errores distintos a una
distribución normal.

# Ajustar el modelo GLM con la distribución de Poisson
model1 <- glm(colonies ~ dose, family = poisson, data = salmonella)
# Imprimir los resultados del modelo
summary(model1)

##
## Call:
## glm(formula = colonies ~ dose, family = poisson, data = salmonella)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 3.3219950  0.0540292  61.485   <2e-16 ***
## dose        0.0001901  0.0001172   1.622    0.105
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 78.358  on 17  degrees of freedom
## Residual deviance: 75.806  on 16  degrees of freedom
```

```
## AIC: 172.34
##
## Number of Fisher Scoring iterations: 4

#El resumen del modelo GLM muestra un valor alto de p = 0.105 para la
variable 'dose'. Parece haber falta de ajuste.

# Ajustar el modelo GLM con log(dose+1)
model2 <- glm(colonies ~ log_dose, family = poisson, data = salmonella)
# Imprimir los resultados del modelo
summary(model2)

##
## Call:
## glm(formula = colonies ~ log_dose, family = poisson, data =
salmonella)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.01989    0.09712  31.095  < 2e-16 ***
## log_dose     0.08585    0.02018   4.255 2.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 78.358  on 17  degrees of freedom
## Residual deviance: 59.629  on 16  degrees of freedom
## AIC: 156.17
##
## Number of Fisher Scoring iterations: 4

#Aunque vemos un valor p mejor con el predictor transformado, los
residuos no muestran una distribución igualmente distribuida alrededor de
la línea y = 0. Hay una varianza de errores desigual/no constante.
También descubrimos que cuando aplicamos la función lm al conjunto de
datos de salmonella, el coeficiente de determinación R cuadrado es débil,
lo que significa que no hay un buen ajuste al conjunto de datos. Sigue
habiendo falta de ajuste.
```

PREGUNTA 5

```
# Ajustar el modelo lineal
fatmod <- lm(brozek ~ age + weight + height + neck + chest + abdom + hip
+ thigh + knee + ankle + biceps + forearm + wrist, data=fat)
# Imprimir los resultados del modelo
summary(fatmod)

##
## Call:
## lm(formula = brozek ~ age + weight + height + neck + chest +
```

```
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist,
##     data = fat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.264  -2.572  -0.097   2.898   9.327
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -15.29255   16.06992  -0.952  0.34225
## age           0.05679    0.02996   1.895  0.05929 .
## weight       -0.08031    0.04958  -1.620  0.10660
## height       -0.06460    0.08893  -0.726  0.46830
## neck         -0.43754    0.21533  -2.032  0.04327 *
## chest        -0.02360    0.09184  -0.257  0.79740
## abdom         0.88543    0.08008  11.057  < 2e-16 ***
## hip          -0.19842    0.13516  -1.468  0.14341
## thigh         0.23190    0.13372   1.734  0.08418 .
## knee         -0.01168    0.22414  -0.052  0.95850
## ankle         0.16354    0.20514   0.797  0.42614
## biceps        0.15280    0.15851   0.964  0.33605
## forearm       0.43049    0.18445   2.334  0.02044 *
## wrist        -1.47654    0.49552  -2.980  0.00318 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.988 on 238 degrees of freedom
## Multiple R-squared:  0.749,  Adjusted R-squared:  0.7353
## F-statistic: 54.63 on 13 and 238 DF,  p-value: < 2.2e-16
```

```
library(MASS)
rfatmod <- rlm(brozek ~ age + weight + height + neck + chest + abdom +
hip + thigh + knee + ankle + biceps + forearm + wrist, data=fat)
summary(rfatmod)
```

```
##
## Call: rlm(formula = brozek ~ age + weight + height + neck + chest +
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist,
##     data = fat)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3964  -2.7352  -0.1171   2.8008   9.4446
##
## Coefficients:
##             Value    Std. Error t value
## (Intercept) -11.3460  17.1216    -0.6627
## age           0.0650   0.0319     2.0368
## weight       -0.0643   0.0528    -1.2163
## height       -0.0625   0.0948    -0.6595
## neck         -0.4553   0.2294    -1.9846
```

```
## chest        -0.0256   0.0978    -0.2614
## abdom         0.8778   0.0853    10.2891
## hip          -0.2142   0.1440    -1.4872
## thigh         0.2632   0.1425     1.8473
## knee         -0.1076   0.2388    -0.4505
## ankle         0.1815   0.2186     0.8306
## biceps        0.1367   0.1689     0.8091
## forearm       0.4152   0.1965     2.1126
## wrist        -1.5739   0.5279    -2.9812
##
## Residual standard error: 4.073 on 238 degrees of freedom
```

#El intercepto es más bajo y cada uno de los predictores (en general) tienen un efecto marginalmente mayor en el resultado. Los predictores más predictivos en el modelo lm original son adom, wrist y forearm, y también lo son en el modelo de Huber, pero lo que varía son algunos de los predictores menores. En el modelo de Huber, la edad parece ser significativa, mientras que no lo es en el modelo lm básico. El error estándar es en realidad mejor en el original, pero solo modestamente.

```
z <- rfatmod$w
names(z) <- row.names(fat)
head(sort(z))
```

```
##       224       207        39       231       225        81
## 0.5269652 0.5800712 0.5987751 0.6260611 0.6304030 0.6367342
```

```
# Los casos 224 y 207 tienen los valores más bajos.
```

```
fat[224,]
```

```
##     brozek siri density age weight height adipos  free neck chest
abdom  hip
## 224    6.1  5.2  1.0874  55 142.25  67.25   22.2 133.6 35.2  92.7
82.8 91.9
##     thigh knee ankle biceps forearm wrist
## 224  54.4 35.2  22.5   29.4    26.8    17
```

```
fat[207,]
```

```
##     brozek siri density age weight height adipos  free neck chest
abdom   hip
## 207   31.7 32.9   1.025  44    166   65.5   27.2 113.5 39.1 100.6
93.9 100.1
##     thigh knee ankle biceps forearm wrist
## 207  58.9 37.6  21.4   33.1    29.5  17.3
```

```
head(fat[order(fat$brozek),])
```

```
##     brozek siri density age weight height adipos  free neck chest
abdom  hip
## 182    0.0  0.0  1.1089  40 118.50  68.00   18.1 118.5 33.8  79.3
```

```
69.4 85.0
## 172    1.9  0.7  1.0983  35 125.75  65.50   20.6 123.4 34.0  90.8
75.0 89.2
## 171    4.1  3.0  1.0926  35 152.25  67.75   23.4 146.1 37.0  92.2
81.9 92.8
## 26     4.6  3.7  1.0911  27 159.25  71.50   21.9 151.9 35.7  89.6
79.7 96.5
## 29     4.7  3.7  1.0910  27 133.25  64.75   22.4 127.0 36.4  93.5
73.9 88.5
## 55     4.9  3.9  1.0906  42 136.25  67.50   21.1 129.6 37.8  87.6
77.6 88.6
##     thigh knee ankle biceps forearm wrist
## 182  47.2 33.5  20.2   27.7    24.6  16.5
## 172  50.0 34.8  22.0   24.8    25.9  16.9
## 171  54.7 36.2  22.1   30.4    27.4  17.7
## 26   55.0 36.7  22.5   29.9    28.2  17.7
## 29   50.1 34.5  21.3   30.5    27.9  17.2
## 55   51.9 34.9  22.5   27.7    27.5  18.5
```

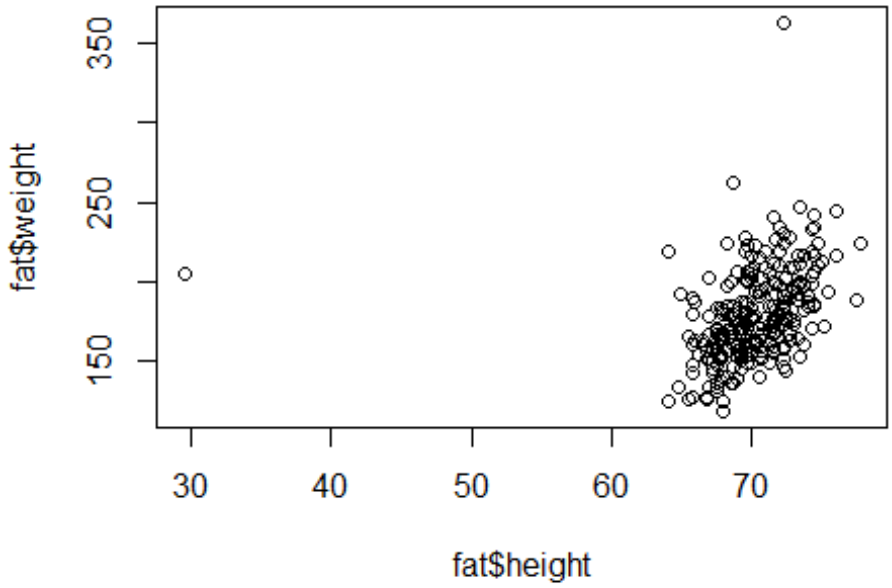
```
head(fat[order(fat$brozek, decreasing = TRUE),])
```

```
##     brozek siri density age weight height adipos  free neck chest
abdom   hip
## 216   45.1 47.5  0.9950  51 219.00  64.00   37.6 120.2 41.2 119.8
122.1 112.8
## 36    38.2 40.1  1.0101  49 191.75  65.00   32.0 118.4 38.4 118.5
113.1 113.8
## 192   36.5 38.1  1.0140  42 244.25  76.00   29.8 155.2 41.8 115.2
113.7 112.4
## 169   34.7 34.3  1.0180  35 228.25  69.50   33.3 149.3 40.4 114.9
115.9 111.9
## 39    33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2
148.1 147.7
## 242   33.6 35.0  1.0207  65 224.50  68.25   33.9 149.2 38.8 119.6
118.0 114.3
##     thigh knee ankle biceps forearm wrist
## 216  62.5 36.9  23.6   34.7    29.1  18.4
## 36   61.9 38.3  21.9   32.0    29.8  17.0
## 192  68.5 45.0  25.5   37.1    31.2  19.9
## 169  74.4 40.6  24.0   36.1    31.8  18.8
## 39   87.3 49.1  29.6   45.0    29.0  21.4
## 242  61.3 42.1  23.4   34.9    30.1  19.4
```

```
colMeans(fat)
```

```
##     brozek       siri    density        age     weight     height
adipos
##  18.938492  19.150794   1.055574  44.884921 178.924405  70.148810
25.436905
##       free       neck      chest      abdom        hip      thigh
knee
```

```
## 143.713889  37.992063 100.824206  92.555952  99.904762  59.405952
38.590476
##      ankle     biceps    forearm      wrist
##  23.102381  32.273413  28.663889  18.229762

# Lo interesante es que se encuentran cerca del extremo alto y bajo de
los valores para el peso de Brozek, pero hay que tener en cuenta que
ninguno de ellos está muy cerca de ser el más bajo para ninguna de las
medidas. Lo que también es inusual es que si se observan las medias de
columna (colMeans) para el conjunto de datos "fat", los valores de las
filas 224 y 207 no tienen valores que parezcan variar mucho de la media,
parecen "relativamente" típicos.

plot(fat$weight ~ fat$height)
identify(fat$weight ~ fat$height)
```




```
## integer(0)

# Los puntos 39 y 42 fueron identificados como valores atípicos. Uno
tiene la altura más baja y el otro el peso más alto y están
sustancialmente separados del resto de los puntos.
# Sin embargo, estos no son los dos puntos con el peso más bajo
identificados en la pregunta anterior (b), aunque "42" con la altura más
baja es el tercer peso físico más bajo. Pero el punto 39, con el peso
físico más alto, en realidad tiene un peso de modelo de 1, lo que
significa que se le da un peso completo en el modelo.
```

```
fat[39,]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
hip
## 39   33.8 35.2  1.0202  46 363.15  72.25   48.9 240.5 51.2 136.2 148.1
147.7
##    thigh knee ankle biceps forearm wrist
## 39  87.3 49.1  29.6     45      29  21.4
```

```
fat[42,]
```

```
##    brozek siri density age weight height adipos  free neck chest abdom
hip
## 42   31.7 32.9   1.025  44    205   29.5   29.9 140.1 36.6   106 104.3
115.5
##    thigh knee ankle biceps forearm wrist
## 42  70.6 42.5  23.7   33.6    28.7  17.4
```

```
z[39]
```

```
##        39
## 0.5987751
```

```
z[42]
```

```
## 42
##  1
```

```
# Es comúnmente asumido que los casos con los pesos más bajos en el
modelo son valores atípicos. Sin embargo, ser un valor atípico no
necesariamente significa que un caso sea poco importante para el modelo,
al menos según el algoritmo de Huber. El algoritmo de Huber solo asigna
pesos más bajos a los valores que se desvían significativamente del error
cuadrático medio.

# Para identificar los casos con los errores más altos y más bajos
(diferencia entre los valores predichos y los valores reales), podemos
ordenar los errores. Al hacerlo, podemos ver que las predicciones para
las filas 207 y 224 están más alejadas del valor real.

min(predict(fatmod) - fat$brozek)
```

```
## [1] -9.32674
```

```
max(predict(fatmod) - fat$brozek)
```

```
## [1] 10.26353
```

```
errors <- predict(fatmod) - fat$brozek
errors[order(errors)]
```

```
##          207           81           82          128          135
192
## -9.326740246 -8.480250444 -8.310690775 -7.937236953 -7.758955591 -
7.323569740
##          249          119          121           33            3
24
## -7.222483537 -7.057953104 -6.922060347 -6.374215955 -6.252231350 -
6.099412184
##          115          148           38           76           62
216
## -6.068221177 -6.048565123 -5.972179308 -5.905070445 -5.781976643 -
5.760728538
##          156          202          153           23           67
195
## -5.753483484 -5.665321123 -5.607709967 -5.607066085 -5.600336102 -
5.567206164
##          208           25           86          127           84
17
## -5.496702951 -5.408701711 -5.272933099 -5.261188961 -5.239335593 -
5.138519868
##           44          215          200          137          104
235
## -5.128290958 -4.892854878 -4.748428662 -4.747473150 -4.740088194 -
4.700175029
##          134           28          109          197           71
120
## -4.590060672 -4.547318115 -4.520872194 -4.513017439 -4.466810428 -
4.260100079
##          252          139          143          213            6
175
## -4.176970668 -4.051587183 -4.013073267 -3.856960302 -3.709195183 -
3.689518678
##          167           46          144          240          173
141
## -3.672680204 -3.661451588 -3.658081445 -3.584511978 -3.541652183 -
3.538373408
##           66          160          228           18          138
234
## -3.426388034 -3.292389956 -3.191149954 -3.148293222 -3.055496422 -
3.002179264
##          117           94          237           63          100
13
## -2.949819930 -2.928179976 -2.910291908 -2.894011636 -2.773960791 -
2.766336569
##           47           78           59          122          103
196
## -2.755563313 -2.620682562 -2.620063863 -2.594752168 -2.531944167 -
2.485774218
##          101          132            7          157           36
243
```

```
## -2.451910868 -2.313650732 -2.310623343 -2.308508365 -2.239497617 -
2.220884542
##          217          241           65          129           74
212
## -2.016906272 -1.956784895 -1.943050272 -1.891207796 -1.825364485 -
1.793654500
##          178          146          179          206          181
10
## -1.624201324 -1.621389342 -1.618837460 -1.517487855 -1.412340955 -
1.394875743
##          251            5          105          114          246
88
## -1.371373735 -1.355656321 -1.312001824 -1.273753572 -1.215500947 -
1.198519465
##          168          166           96          113          191
145
## -1.165096790 -1.129126835 -1.012550774 -0.994287810 -0.979269809 -
0.936836446
##          203          151          199           61          219
159
## -0.912585051 -0.783202256 -0.768405351 -0.699878050 -0.629847610 -
0.615146285
##          123           92          111          102           42
110
## -0.614063380 -0.577461426 -0.461265211 -0.438073261 -0.397557661 -
0.348638044
##           58          165          116          174          136
106
## -0.315270117 -0.283844742 -0.266894612 -0.242014491 -0.226214261 -
0.122204790
##           37           99           40           35          130
90
## -0.039775918 -0.004164290  0.003380545  0.032541671  0.085673834
0.090389476
##          154          164          131          244           77
85
##  0.103646852  0.161977787  0.188430945  0.198381292  0.272175438
0.278121162
##          245          226          185          118          247
19
##  0.336389040  0.352395229  0.376412091  0.425856244  0.441460975
0.611702470
##          155          149           70           56          205
233
##  0.688095922  0.733336556  0.771006154  0.802323028  0.886750166
0.929834140
##          124          133          176           91           27
64
##  1.067504958  1.071808823  1.089736027  1.110899192  1.143719656
1.296101092
```

```
##           50            8          169          218          150
4
##  1.321479840  1.346472137  1.354520090  1.371188704  1.379469960
1.397600909
##          188           79          125           60          242
68
##  1.463068549  1.470454965  1.535017900  1.543936943  1.557764745
1.740667507
##          239           21          193          194           15
190
##  1.759099800  1.818916398  1.838677020  1.850908182  1.905346337
1.943330571
##          189           16          198           11           69
229
##  1.971224424  1.972940322  1.992984700  2.063488268  2.131570179
2.162792720
##          161           93           41          186           34
73
##  2.228433106  2.231387704  2.250878879  2.289392886  2.291908594
2.296554594
##          163           29          152           31           87
43
##  2.365234487  2.369271885  2.417510053  2.422538099  2.431043322
2.487680848
##          214          177            2           52           30
220
##  2.530132369  2.535439065  2.559765439  2.609298207  2.628319455
2.652202377
##          162          142           45          147          201
126
##  2.826430491  2.833008837  2.855322308  3.039754680  3.130436174
3.230399417
##          230          248          170          210           55
1
##  3.367332565  3.370326238  3.383059932  3.426305657  3.474885433
3.564508677
##          108          236           14           48           72
187
##  3.564998529  3.566693218  3.596004602  3.605176496  3.660259456
3.729414039
##          209           54          227          112           51
89
##  3.863018374  3.936430547  3.957901882  4.120180330  4.124763307
4.220435703
##           49          222           83          183           22
184
##  4.233347035  4.235489955  4.260587556  4.361475626  4.406119506
4.406373970
##           26           12           75           32          232
98
```

```
##  4.497675174  4.675528543  4.754878386  4.875461736  4.894576083
5.022766447
##           57            9          158           95          211
182
##  5.034121476  5.113356443  5.175276754  5.231336135  5.258555963
5.309336175
##          223           80           53          107           20
180
##  5.335983091  5.548460835  5.643654028  6.093045743  6.140421251
6.281388794
##          238           97          172          250          140
221
##  6.667960176  6.723220533  6.925153108  7.116068918  7.726238505
7.831497228
##          171           39          204          231          225
224
##  7.911169314  8.170497848  8.371973433  8.609910081  8.736079635
10.263532226
```

PREGUNTA 6

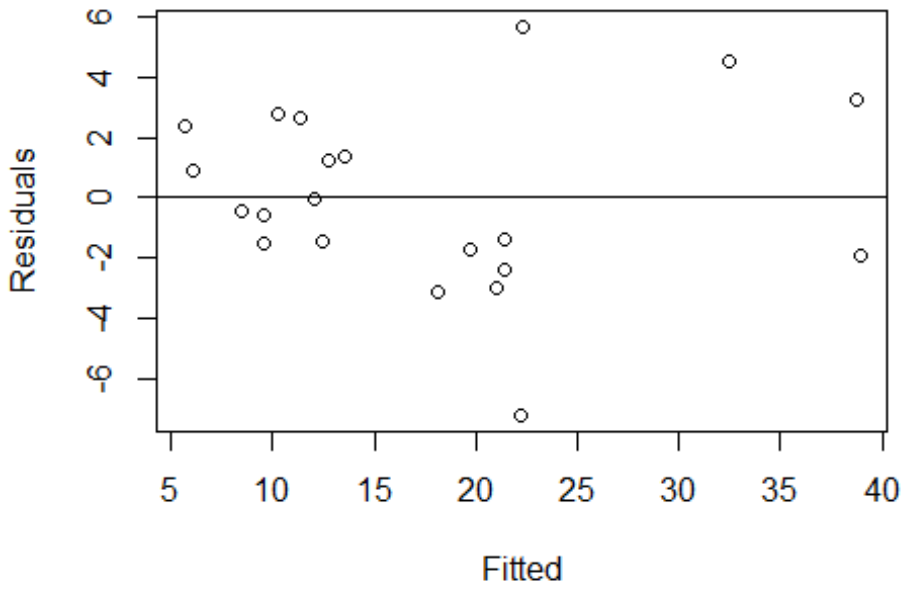
```
data(stackloss, package="faraway")
```

```
## Warning in data(stackloss, package = "faraway"): data set 'stackloss'
not found
```

```
lm.fit <- lm(stack.loss ~ . , data=stackloss)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = stack.loss ~ ., data = stackloss)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2377 -1.7117 -0.4551  2.3614  5.6978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -39.9197    11.8960  -3.356  0.00375 **
## Air.Flow      0.7156     0.1349   5.307  5.8e-05 ***
## Water.Temp    1.2953     0.3680   3.520  0.00263 **
## Acid.Conc.   -0.1521     0.1563  -0.973  0.34405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.243 on 17 degrees of freedom
## Multiple R-squared:  0.9136, Adjusted R-squared:  0.8983
## F-statistic:  59.9 on 3 and 17 DF,  p-value: 3.016e-09
```

```r
plot(fitted(lm.fit),residuals(lm.fit),xlab="Fitted",ylab="Residuals")
abline(h=0)
```




```r
# Vemos que puede haber una asociación con la varianza de los residuos y el valor de la respuesta


library(quantreg)
```

```
## Loading required package: SparseM
```

```
##
## Attaching package: 'SparseM'
```

```
## The following object is masked from 'package:base':
##
##     backsolve
```

```r
lm.fit1 <- rq(stack.loss ~ ., data= stackloss)
summary(lm.fit1)
```

```
##
## Call: rq(formula = stack.loss ~ ., data = stackloss)
##
## tau: [1] 0.5
##
## Coefficients:
##             coefficients lower bd  upper bd
```

```
## (Intercept) -39.68986    -41.61973 -29.67754
## Air.Flow      0.83188      0.51278   1.14117
## Water.Temp    0.57391      0.32182   1.41090
## Acid.Conc.   -0.06087     -0.21348  -0.02891

library(MASS)
lm.fit2 <- rlm(stack.loss ~ . ,data=stackloss)
summary(lm.fit2)

##
## Call: rlm(formula = stack.loss ~ ., data = stackloss)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -8.91753 -1.73127  0.06187  1.54306  6.50163
##
## Coefficients:
##             Value    Std. Error t value
## (Intercept) -41.0265   9.8073    -4.1832
## Air.Flow      0.8294   0.1112     7.4597
## Water.Temp    0.9261   0.3034     3.0524
## Acid.Conc.   -0.1278   0.1289    -0.9922
##
## Residual standard error: 2.441 on 17 degrees of freedom

fit2.weights <- lm.fit2$w
names(fit2.weights) <- row.names(stackloss)
head(sort(fit2.weights),10)

##        21         4         3         1         2         5         6
7
## 0.3681411 0.5049409 0.7858871 1.0000000 1.0000000 1.0000000 1.0000000
1.0000000
##         8         9
## 1.0000000 1.0000000

# 21, 4 y 3 tienen valores menores a 1

sq.fit <- ltsreg(stack.loss ~ . ,data=stackloss)
coef(sq.fit)

## (Intercept)    Air.Flow  Water.Temp  Acid.Conc.
## -33.9317376   0.7500000   0.3297872  -0.0212766

# Revisamos apalancamiento
library(pander)
df <-stackloss
numPredictors <- ( ncol(df)-1)
hatv <- hatvalues(lm.fit)
lev.cut <- (numPredictors+1) *2 * 1/ nrow(df)
high.leverage <- df[hatv > lev.cut,]
pander(high.leverage, caption = "High Leverage Data Elements")
```

High Leverage Data Elements

	Air.Flow	Water.Temp	Acid.Conc.	stack.loss
17	50	19	72	8

```
# Revisamos outliers
studentized.residuals <- rstudent(lm.fit)
max.residual <-
studentized.residuals[which.max(abs(studentized.residuals))]
range.residuals <- range(studentized.residuals)
names(range.residuals) <- c("left", "right")
pander(data.frame(range.residuals=t(range.residuals)), caption="Range of
Studentized residuals")
```

Range of Studentized residuals

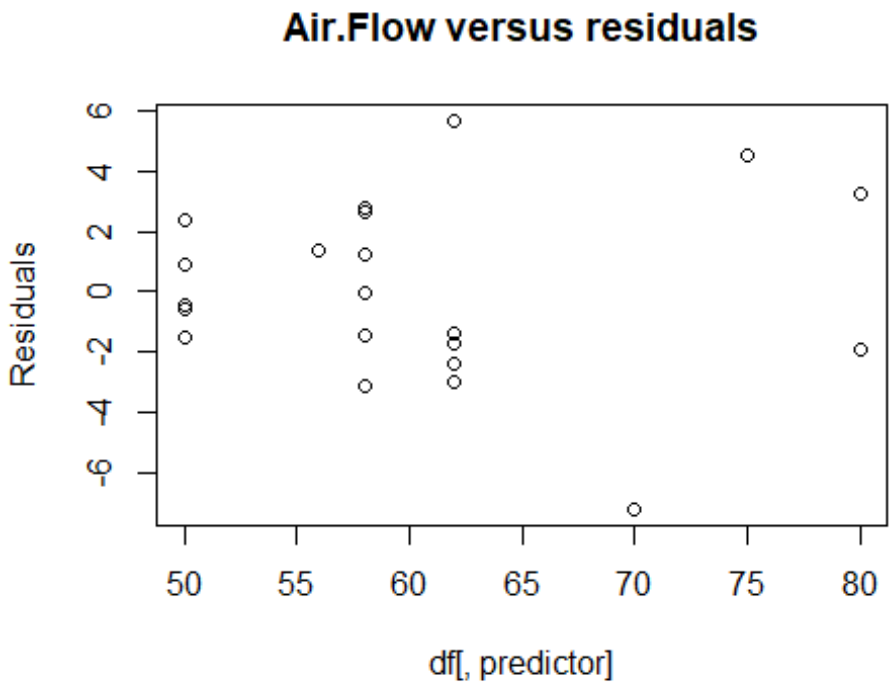
range.residuals.left	range.residuals.right
-3.33	2.052

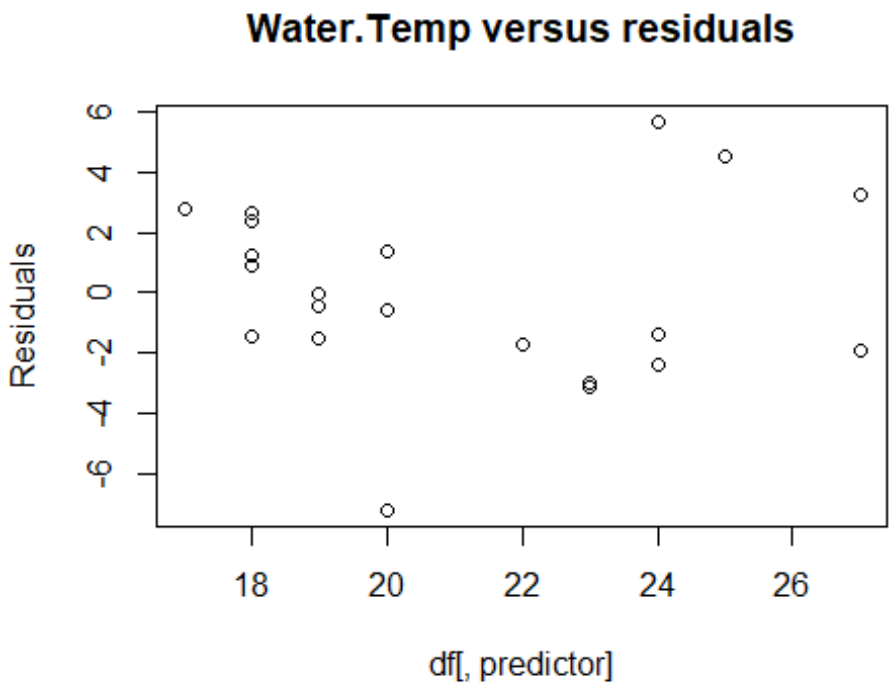
```
p<-numPredictors+1
n<-nrow(df)
t.val.alpha <- qt(.05/(n*2),n-p-1)
pander(data.frame(t.val.alpha = t.val.alpha), caption = "Bonferroni
corrected t-value")
```

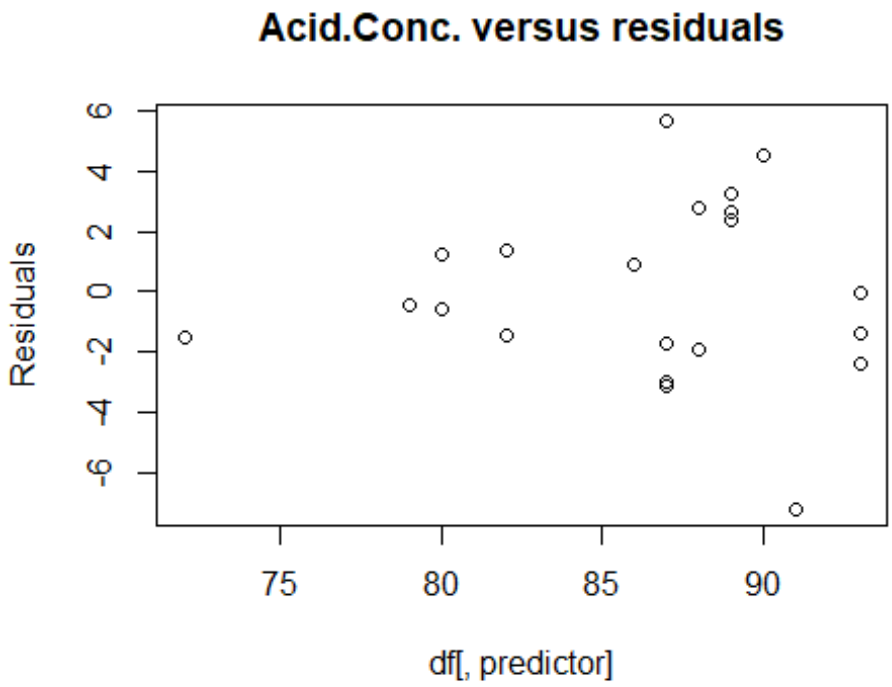
Bonferroni corrected t-value

t.val.alpha
-3.604

```
outlier.index <- abs(studentized.residuals) > abs(t.val.alpha)
outliers <- df[outlier.index==TRUE,]
if(nrow(outliers)>=1)
{
  pander(outliers, caption = "outliers")
}

predictors <-names(lm.fit$coefficients)
predictors <- predictors[2:length(predictors)]
for(i in 1:length(predictors))
{
  predictor <- predictors[i]

  plot(df[,predictor],residuals(lm.fit),xlab=,ylab="Residuals",main =
paste(predictor, " versus residuals", sep = ''))
}
```

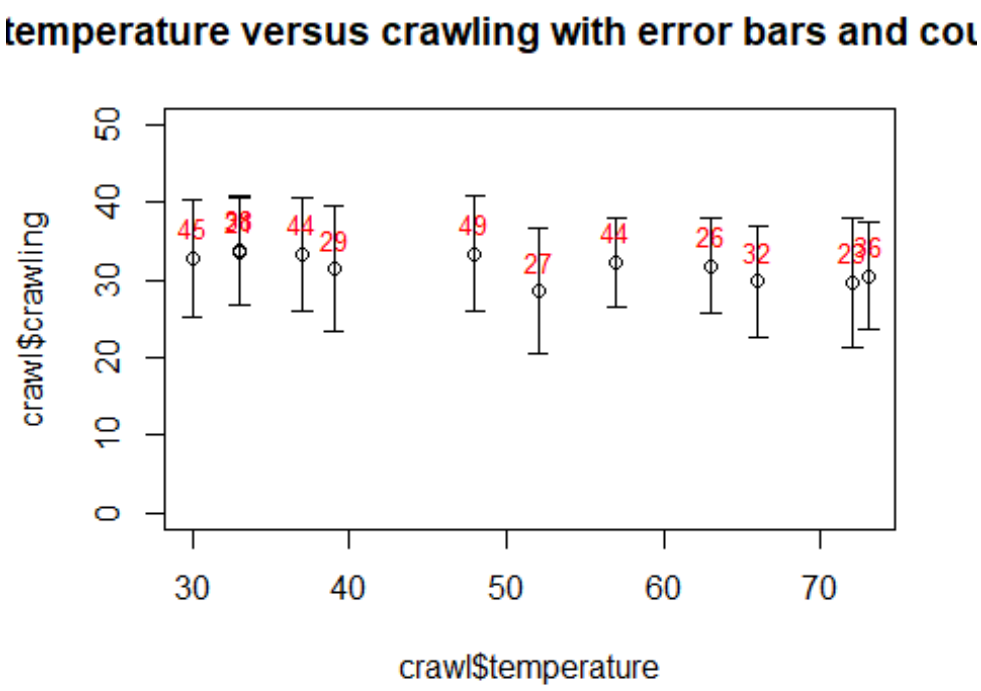

Air.Flow versus residuals
Residuals
6
4
2
0
-2
-4
-6
50
55
60
65
70
75
80
df[, predictor]


Water.Temp versus residuals
Residuals
6
4
2
0
-2
-4
-6
18
20
22
24
26
df[, predictor]



PREGUNTA 7

```
library(latex2exp)
library(nlme)
data(crawl, package="faraway")
plot(crawl$temperature,crawl$crawling, ylim = c(0,50), main =
"temperature versus crawling with error bars and counts ")
arrows(crawl$temperature, crawl$crawling-crawl$SD,crawl$temperature,
crawl$crawling+crawl$SD, length=0.05, angle=90, code=3)
 text(crawl$temperature, crawl$crawling, labels=crawl$n, cex= .8, pos=3,
col='red')
```

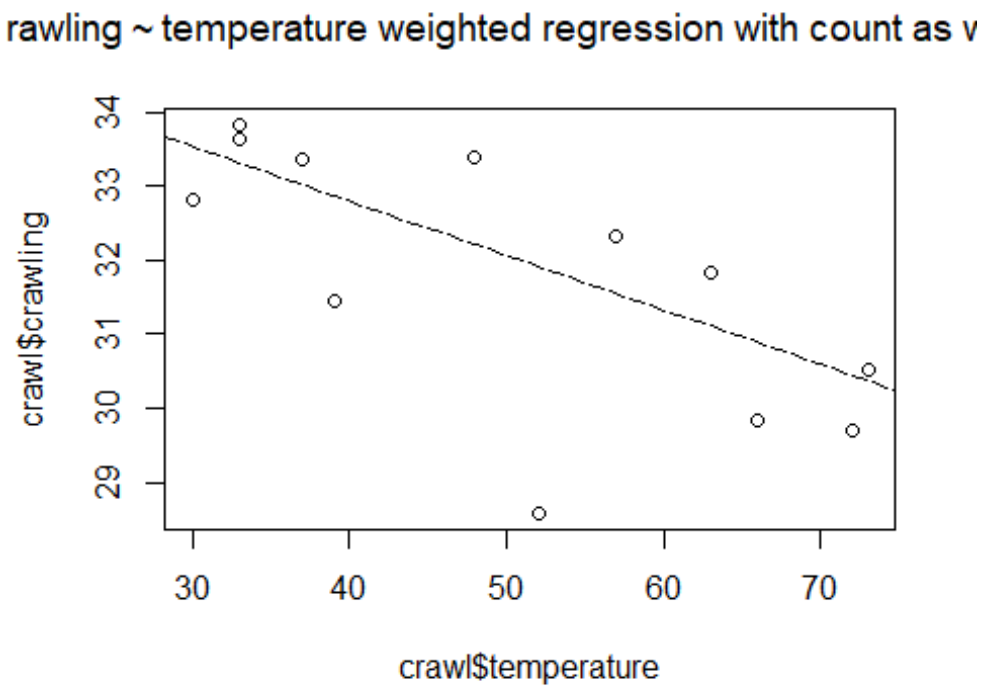



```
# Ajustamos el modelo de regresión con los pesos basados en la varianza inversa de la variable respuesta crawling:
 wts <- crawl$n/crawl$SD^2
lm1 <- lm(crawling ~ temperature, data=crawl, weights = wts)
summary(lm1)

##
## Call:
## lm(formula = crawling ~ temperature, data = crawl, weights = wts)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1504 -0.6817  0.1688  0.4941  1.1009
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.73262    1.21153   29.49 4.69e-11 ***
## temperature -0.07332    0.02328   -3.15   0.0103 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9772 on 10 degrees of freedom
## Multiple R-squared:  0.4981, Adjusted R-squared:  0.4479
## F-statistic: 9.923 on 1 and 10 DF,  p-value: 0.01033

#  Ajusta un modelo de regresión lineal ponderada al conjunto de datos crawl utilizando la función gls del paquete nlme
```

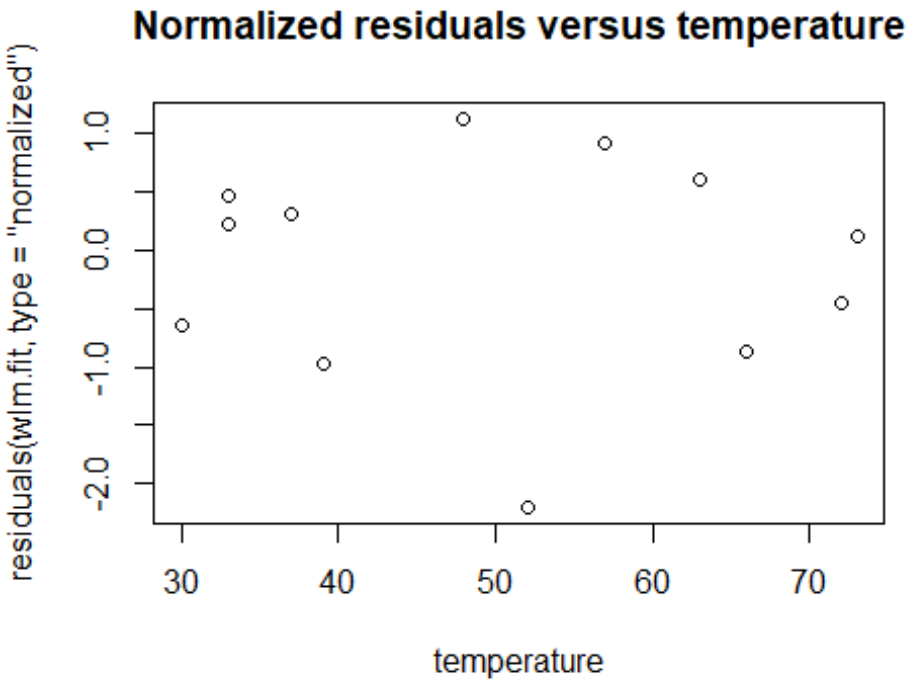
```r
wlm.fit <- gls(crawling ~ temperature, data=crawl, weights = ~ SD^2/n)
summary(wlm.fit)
```

```
## Generalized least squares fit by REML
##   Model: crawling ~ temperature
##   Data: crawl
##        AIC      BIC    logLik
##   48.76397 49.67173 -21.38199
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~SD^2/n
##
## Coefficients:
##                Value Std.Error   t-value p-value
## (Intercept) 35.73262 1.2115286 29.493832  0.0000
## temperature -0.07332 0.0232771 -3.150053  0.0103
##
##  Correlation:
##             (Intr)
## temperature -0.96
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.2007008 -0.6976329  0.1727593  0.5057059  1.1266156
##
## Residual standard error: 0.9771564
## Degrees of freedom: 12 total; 10 residual
```

```r
plot(crawl$temperature, crawl$crawling, main = TeX("$crawling \\sim
temperature$ weighted regression with count as weight"))
abline(coef(wlm.fit), lty = 5)
```




```
plot(residuals(wlm.fit, type = "normalized") ~ temperature, crawl, main =
"Normalized residuals versus temperature")

plot(residuals(wlm.fit, type = "normalized") ~ temperature, crawl, main =
"Normalized residuals versus temperature")
```




```
library(pracma)

##
## Attaching package: 'pracma'

## The following object is masked from 'package:faraway':
##
##     logit

SSpe<- pracma::dot(crawl$n, crawl$SD)
total <- sum(crawl$n)
groups <- nrow(crawl)
sigma.sq.estimated <- SSpe/ (total-groups)
pander(data.frame(se=sqrt(sigma.sq.estimated)), caption="estimated SD
from the repeated predictor values")
```

estimated SD from the repeated predictor values

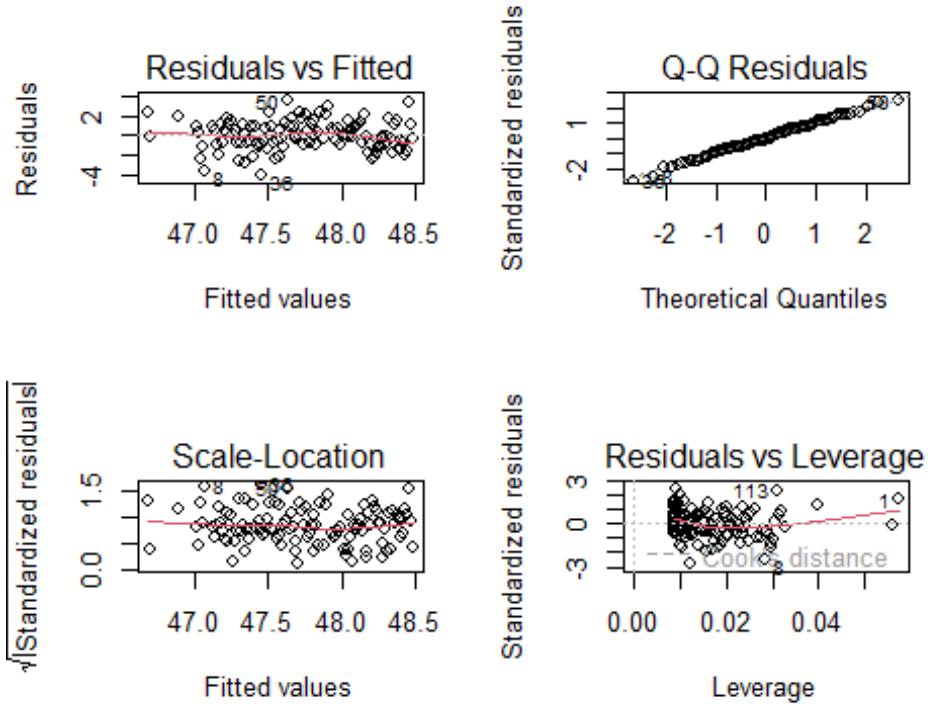
se
2.718

PREGUNTA 8

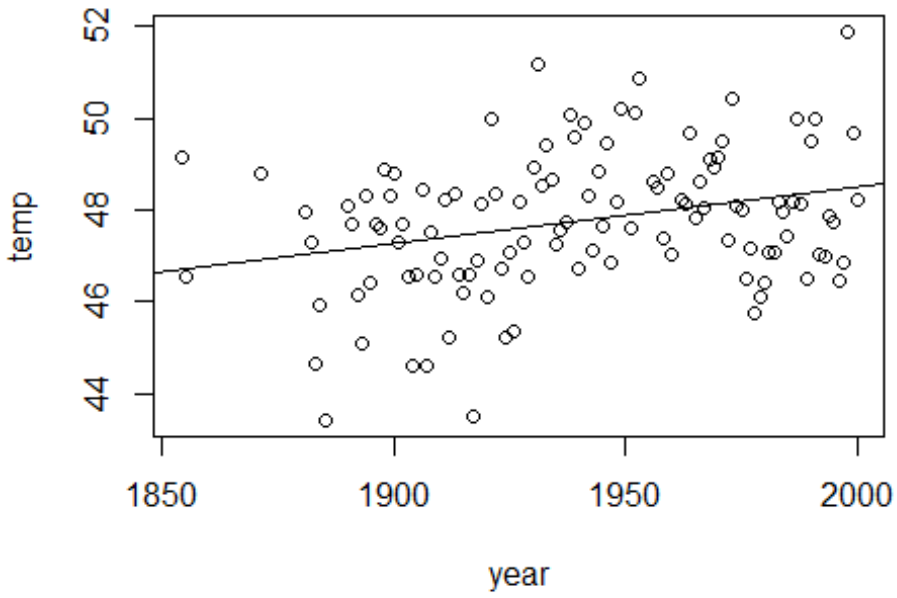
```
library(nlme)
lmod<-lm(temp~year, data= aatemp)
summary(lmod)
```

```
##
## Call:
## lm(formula = temp ~ year, data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9843 -0.9113 -0.0820  0.9946  3.5343
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 24.005510   7.310781   3.284  0.00136 **
## year         0.012237   0.003768   3.247  0.00153 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.466 on 113 degrees of freedom
## Multiple R-squared:  0.08536,    Adjusted R-squared:  0.07727
## F-statistic: 10.55 on 1 and 113 DF,  p-value: 0.001533
```

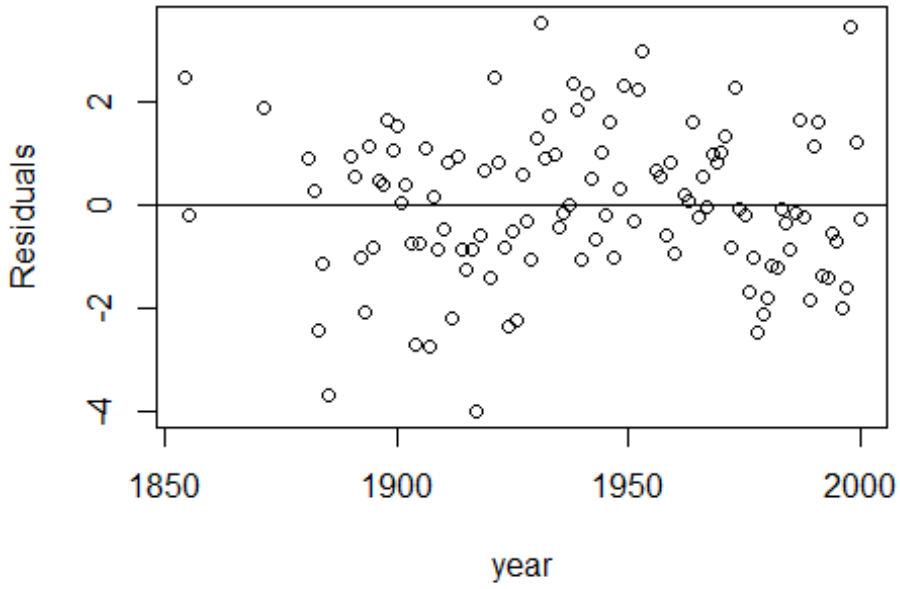
```
par(mfrow=c(2,2))
plot(lmod)
```



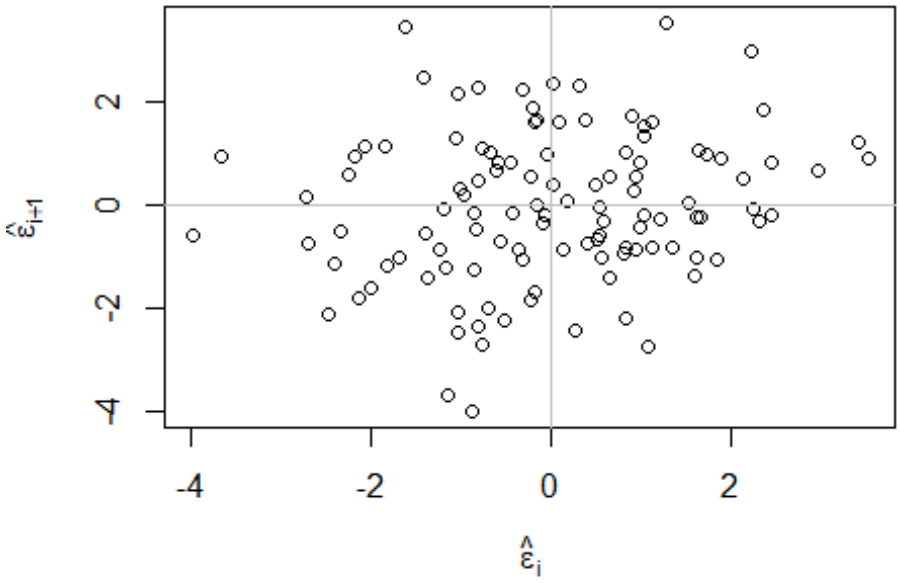

```
par(mfrow=c(1,1))
plot(temp~year, data=aatemp)
abline(coefficients(lmod))
```




```
# Los datos muestran una tendencia lineal en el gráfico QQ, esto sugiere
que los datos tienen una relación lineal y siguen una distribución
normal.

plot(residuals(lmod) ~ year, na.omit(aatemp), ylab="Residuals")
abline(h=0)
```




```
n <- length(residuals(lmod))
plot(tail(residuals(lmod),n-1) ~ head(residuals(lmod),n-1), xlab=
expression(hat(epsilon)[i]),ylab=expression(hat(epsilon)[i+1]))
abline(h=0,v=0,col=grey(0.75))
```




```
# Se observan largas secuencias de puntos por encima o por debajo de la
línea. Esto sugiere que los valores de la serie de datos están
correlacionados positivamente, ya que los valores tienden a ser similares
entre sí durante un período de tiempo prolongado.

glmod <- gls(temp~year, correlation=corAR1 (form=~year),
data=na.omit(aatemp))
summary(glmod)

## Generalized least squares fit by REML
##   Model: temp ~ year
##   Data: na.omit(aatemp)
##        AIC     BIC    logLik
##   426.5694 437.479 -209.2847
##
## Correlation Structure: ARMA(1,0)
##  Formula: ~year
##  Parameter estimate(s):
##      Phi1
## 0.2303887
##
## Coefficients:
##                Value Std.Error  t-value p-value
## (Intercept) 25.18407  8.971864 2.807006  0.0059
## year         0.01164  0.004626 2.516015  0.0133
##
##  Correlation:
```

```
##      (Intr)
## year -1
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.7230803 -0.6321970 -0.0520135  0.6645795  2.3775123
##
## Residual standard error: 1.475718
## Degrees of freedom: 115 total; 113 residual
```

```
intervals(glmod,which="var-cov")
```

```
## Approximate 95% confidence intervals
##
##  Correlation structure:
##           lower      est.     upper
## Phi1 0.02952135 0.2303887 0.4133708
##
##  Residual standard error:
##    lower     est.    upper
## 1.284121 1.475718 1.695902
```

```
#Se observa correlación positiva según el valor de correlación estimada.
Se observa una tendencia lineal ya que el valor P sigue siendo
significativo
```

```
mean(aatemp$year)
```

```
## [1] 1939.739
```

```
lmod1<-lm(temp~poly(year-1939,8),data=aatemp)
summary(lmod1)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 8), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6086 -0.8600 -0.2385  1.0608  3.3975
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            47.7426     0.1313 363.579  < 2e-16 ***
## poly(year - 1939, 8)1   4.7616     1.4082   3.381  0.00101 **
## poly(year - 1939, 8)2  -0.9071     1.4082  -0.644  0.52085
## poly(year - 1939, 8)3  -3.3132     1.4082  -2.353  0.02047 *
## poly(year - 1939, 8)4   2.4383     1.4082   1.732  0.08626 .
## poly(year - 1939, 8)5   3.3824     1.4082   2.402  0.01805 *
## poly(year - 1939, 8)6   1.2124     1.4082   0.861  0.39118
## poly(year - 1939, 8)7  -0.9373     1.4082  -0.666  0.50713
```

```
## poly(year - 1939, 8)8  -1.1011     1.4082  -0.782  0.43600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.408 on 106 degrees of freedom
## Multiple R-squared:  0.2086, Adjusted R-squared:  0.1489
## F-statistic: 3.494 on 8 and 106 DF,  p-value: 0.001284
```

```
lmod2<-lm(temp~poly(year-1939,7),data=aatemp)
summary(lmod2)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 7), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5922 -0.9032 -0.2322  0.9880  3.2941
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            47.7426     0.1311 364.241  < 2e-16 ***
## poly(year - 1939, 7)1   4.7616     1.4056   3.388 0.000988 ***
## poly(year - 1939, 7)2  -0.9071     1.4056  -0.645 0.520083
## poly(year - 1939, 7)3  -3.3132     1.4056  -2.357 0.020234 *
## poly(year - 1939, 7)4   2.4383     1.4056   1.735 0.085672 .
## poly(year - 1939, 7)5   3.3824     1.4056   2.406 0.017828 *
## poly(year - 1939, 7)6   1.2124     1.4056   0.863 0.390303
## poly(year - 1939, 7)7  -0.9373     1.4056  -0.667 0.506341
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.406 on 107 degrees of freedom
## Multiple R-squared:  0.2041, Adjusted R-squared:  0.152
## F-statistic: 3.919 on 7 and 107 DF,  p-value: 0.0007651
```

```
lmod3<-lm(temp~poly(year-1939,6),data=aatemp)
summary(lmod3)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 6), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6846 -0.8825 -0.1428  0.9388  3.2950
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            47.7426     0.1307 365.181  < 2e-16 ***
## poly(year - 1939, 6)1   4.7616     1.4020   3.396 0.000957 ***
```

```
## poly(year - 1939, 6)2  -0.9071     1.4020  -0.647 0.518996
## poly(year - 1939, 6)3  -3.3132     1.4020  -2.363 0.019905 *
## poly(year - 1939, 6)4   2.4383     1.4020   1.739 0.084851 .
## poly(year - 1939, 6)5   3.3824     1.4020   2.413 0.017527 *
## poly(year - 1939, 6)6   1.2124     1.4020   0.865 0.389067
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.402 on 108 degrees of freedom
## Multiple R-squared:  0.2008, Adjusted R-squared:  0.1564
## F-statistic: 4.522 on 6 and 108 DF,  p-value: 0.0003978
```

```
lmod4<-lm(temp~poly(year-1939,5),data=aatemp)
summary(lmod4)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 5), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7142 -0.9198 -0.1420  0.9903  3.2364
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            47.7426     0.1306 365.604  < 2e-16 ***
## poly(year - 1939, 5)1   4.7616     1.4004   3.400 0.000942 ***
## poly(year - 1939, 5)2  -0.9071     1.4004  -0.648 0.518500
## poly(year - 1939, 5)3  -3.3132     1.4004  -2.366 0.019749 *
## poly(year - 1939, 5)4   2.4383     1.4004   1.741 0.084470 .
## poly(year - 1939, 5)5   3.3824     1.4004   2.415 0.017384 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.4 on 109 degrees of freedom
## Multiple R-squared:  0.1952, Adjusted R-squared:  0.1583
## F-statistic: 5.289 on 5 and 109 DF,  p-value: 0.0002176
```

```
lmod5<-lm(temp~poly(year-1939,4),data=aatemp)
summary(lmod5)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 4), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0085 -0.9618 -0.0913  0.9926  3.7370
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)            47.7426     0.1334 357.827  < 2e-16 ***
## poly(year - 1939, 4)1   4.7616     1.4308   3.328  0.00119 **
## poly(year - 1939, 4)2  -0.9071     1.4308  -0.634  0.52741
## poly(year - 1939, 4)3  -3.3132     1.4308  -2.316  0.02243 *
## poly(year - 1939, 4)4   2.4383     1.4308   1.704  0.09117 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.431 on 110 degrees of freedom
## Multiple R-squared:  0.1522, Adjusted R-squared:  0.1213
## F-statistic: 4.936 on 4 and 110 DF,  p-value: 0.001068
```

```
lmod6<-lm(temp~poly(year-1939,3),data=aatemp)
summary(lmod6)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 3), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8557 -0.9646 -0.1552  1.0485  4.1538
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            47.7426     0.1346 354.796   <2e-16 ***
## poly(year - 1939, 3)1   4.7616     1.4430   3.300   0.0013 **
## poly(year - 1939, 3)2  -0.9071     1.4430  -0.629   0.5309
## poly(year - 1939, 3)3  -3.3132     1.4430  -2.296   0.0236 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.443 on 111 degrees of freedom
## Multiple R-squared:  0.1298, Adjusted R-squared:  0.1063
## F-statistic: 5.518 on 3 and 111 DF,  p-value: 0.001436
```

```
lmod7<-lm(temp~poly(year-1939,2),data=aatemp)
summary(lmod7)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 2), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0412 -0.9538 -0.0624  0.9959  3.5820
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            47.7426     0.1371 348.218  < 2e-16 ***
## poly(year - 1939, 2)1   4.7616     1.4703   3.239  0.00158 **
```

```
## poly(year - 1939, 2)2  -0.9071     1.4703  -0.617  0.53851
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.47 on 112 degrees of freedom
## Multiple R-squared:  0.08846,    Adjusted R-squared:  0.07218
## F-statistic: 5.434 on 2 and 112 DF,  p-value: 0.005591
```

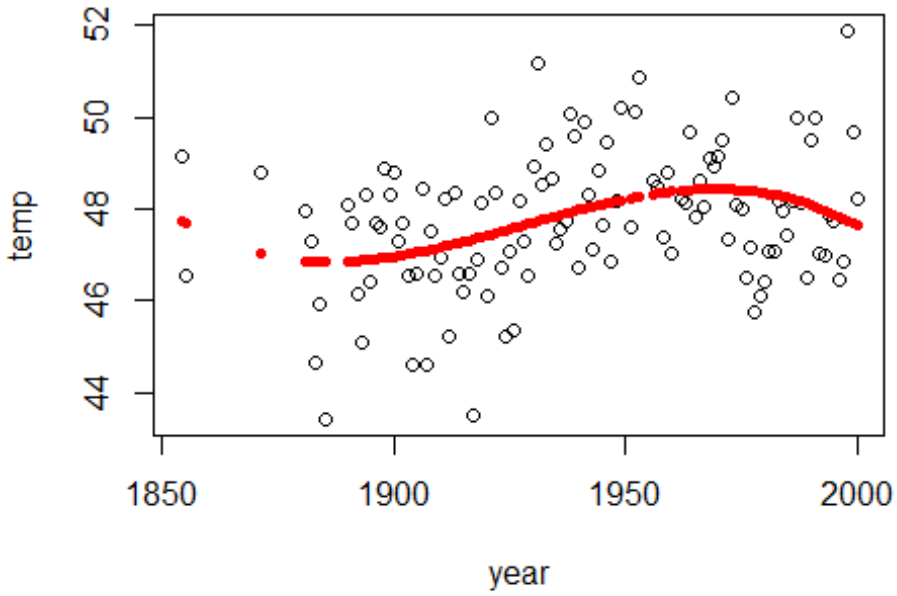
```
lmod8<-lm(temp~poly(year-1939,1),data=aatemp)
summary(lmod8)
```

```
##
## Call:
## lm(formula = temp ~ poly(year - 1939, 1), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9843 -0.9113 -0.0820  0.9946  3.5343
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           47.7426     0.1367 349.176  < 2e-16 ***
## poly(year - 1939, 1)   4.7616     1.4663   3.247  0.00153 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.466 on 113 degrees of freedom
## Multiple R-squared:  0.08536,    Adjusted R-squared:  0.07727
## F-statistic: 10.55 on 1 and 113 DF,  p-value: 0.001533
```

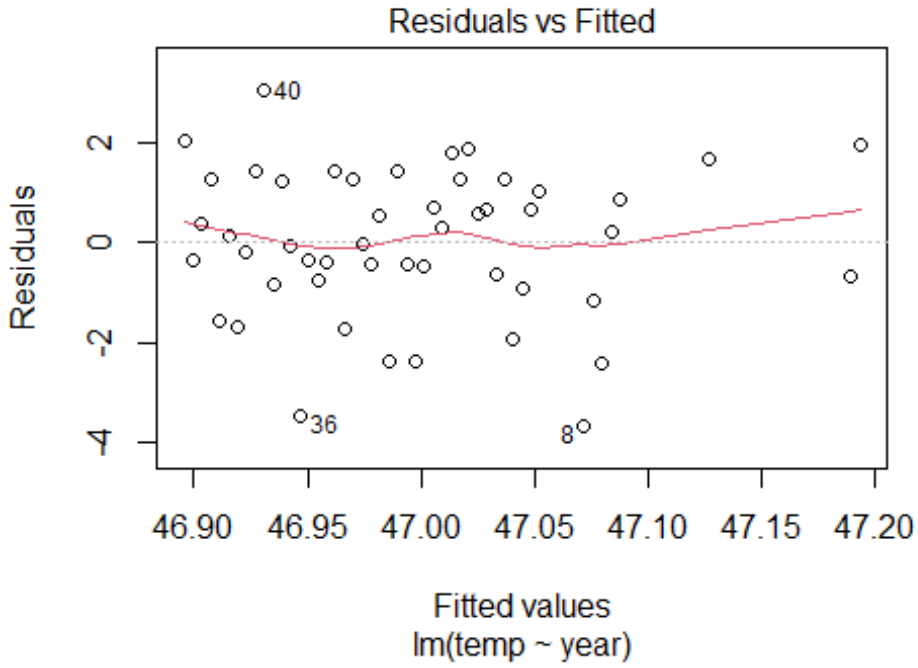
```
lmodx<-lm(temp~poly(year,8), data=aatemp)
summary(lmodx)
```

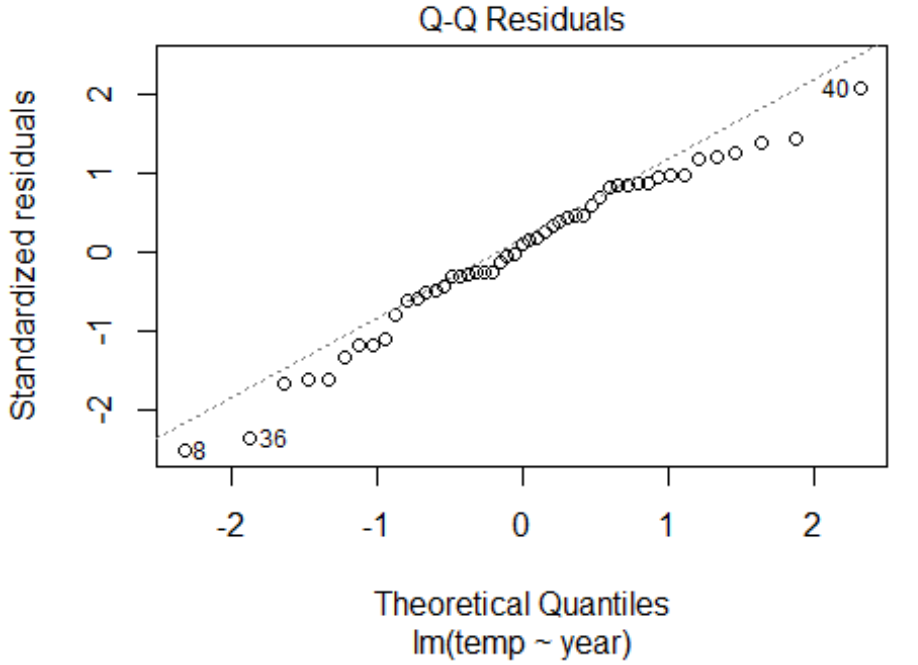
```
##
## Call:
## lm(formula = temp ~ poly(year, 8), data = aatemp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6086 -0.8600 -0.2385  1.0608  3.3975
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     47.7426     0.1313 363.579  < 2e-16 ***
## poly(year, 8)1   4.7616     1.4082   3.381  0.00101 **
## poly(year, 8)2  -0.9071     1.4082  -0.644  0.52085
## poly(year, 8)3  -3.3132     1.4082  -2.353  0.02047 *
## poly(year, 8)4   2.4383     1.4082   1.732  0.08626 .
## poly(year, 8)5   3.3824     1.4082   2.402  0.01805 *
## poly(year, 8)6   1.2124     1.4082   0.861  0.39118
## poly(year, 8)7  -0.9373     1.4082  -0.666  0.50713
```

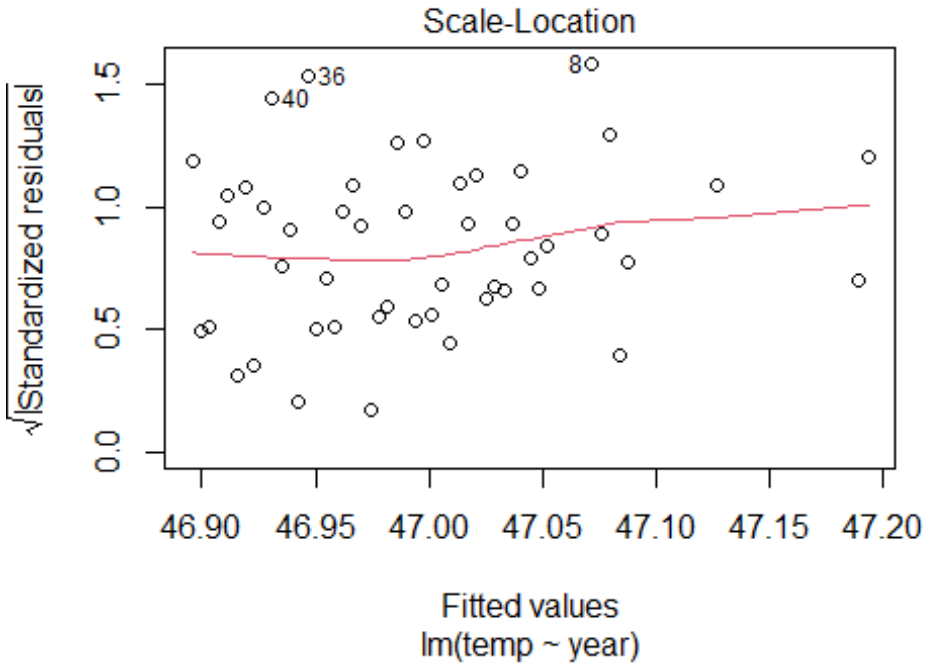
```
## poly(year, 8)8  -1.1011     1.4082  -0.782  0.43600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.408 on 106 degrees of freedom
## Multiple R-squared:  0.2086, Adjusted R-squared:  0.1489
## F-statistic: 3.494 on 8 and 106 DF,  p-value: 0.001284

#El modelo con grado 5 es el modelo polinomial más alto ya que el valor
de P es significativo. Podemos graficar el modelo ajustado.
attach(aatemp)
plot(temp~year, data=aatemp)
points(year, fitted(lmod6),col="red", pch=20)
```

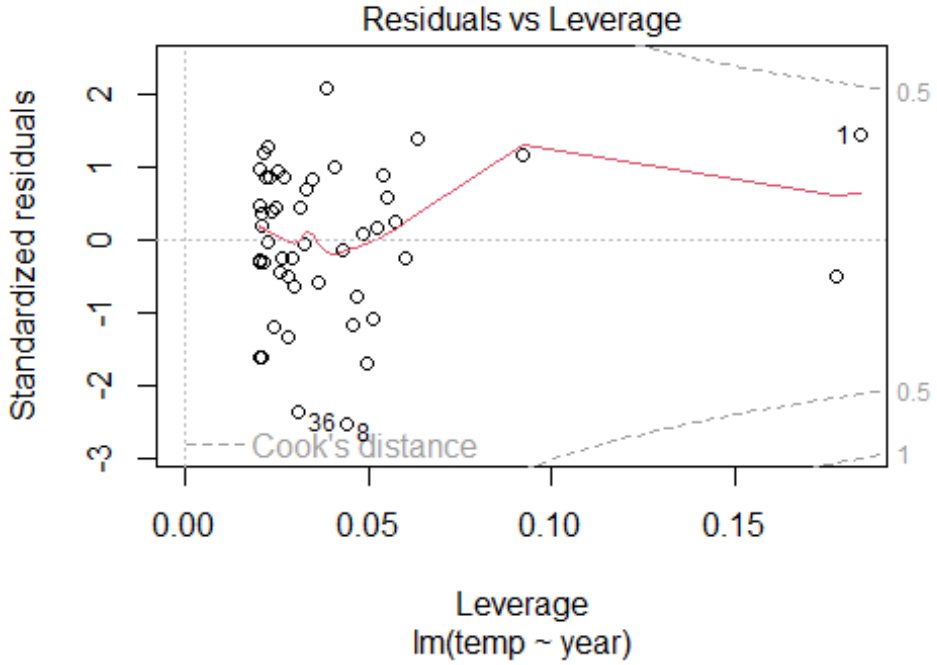



```
l1<-lm(temp~year, aatemp,subset=(year<=1930))
l2<-lm(temp~year, aatemp, subset=(year>1930))
plot(l1)
```


Residuals vs Fitted
Residuals
40
36
8
2
0
-2
-4
46.90
46.95
47.00
47.05
47.10
47.15
47.20
Fitted values
lm(temp ~ year)

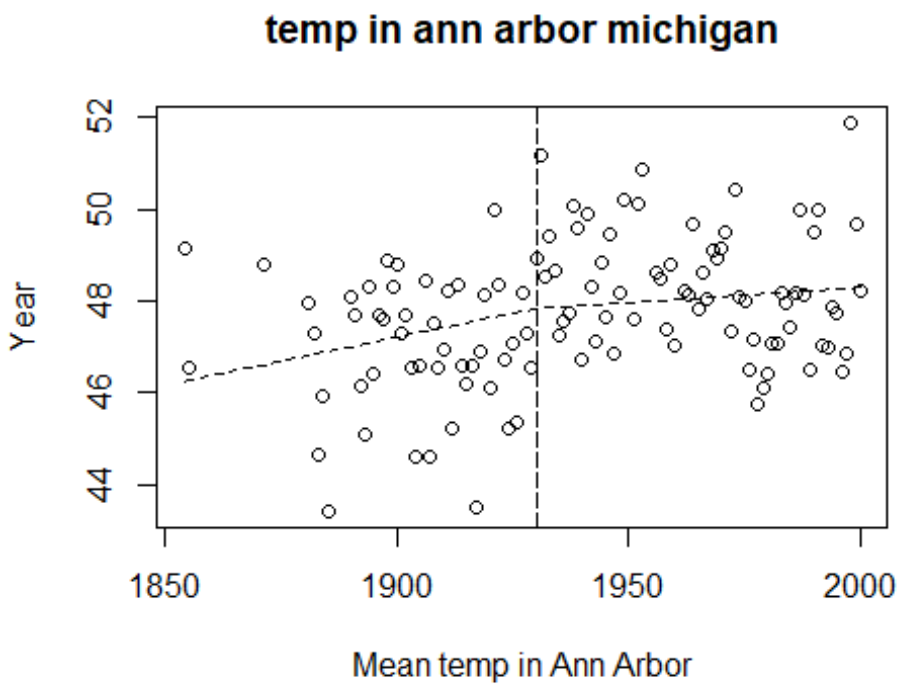

Q-Q Residuals
Standardized residuals
40
36
8
2
1
0
-1
-2
-2
-1
0
1
2
Theoretical Quantiles
lm(temp ~ year)



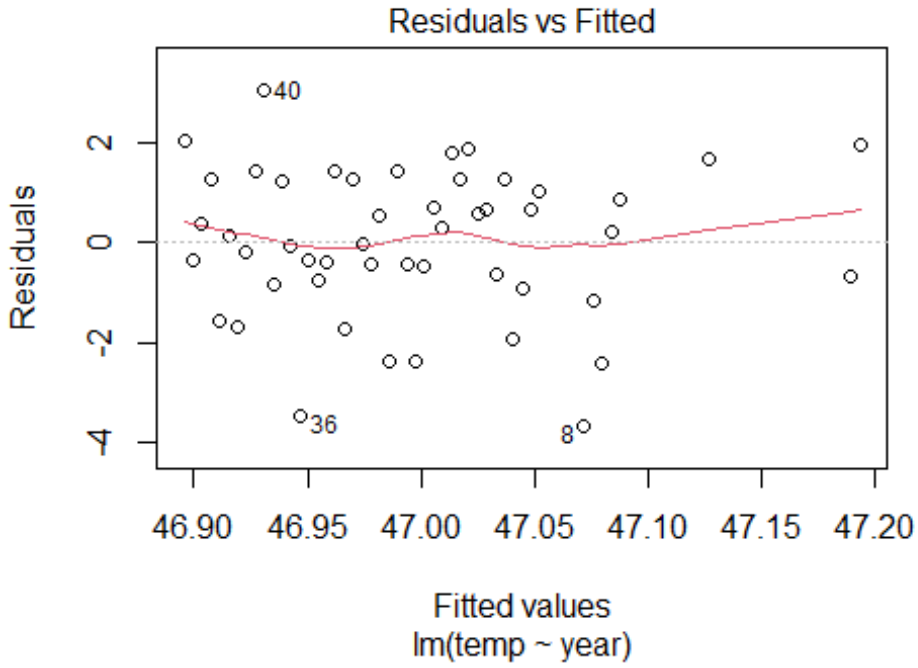


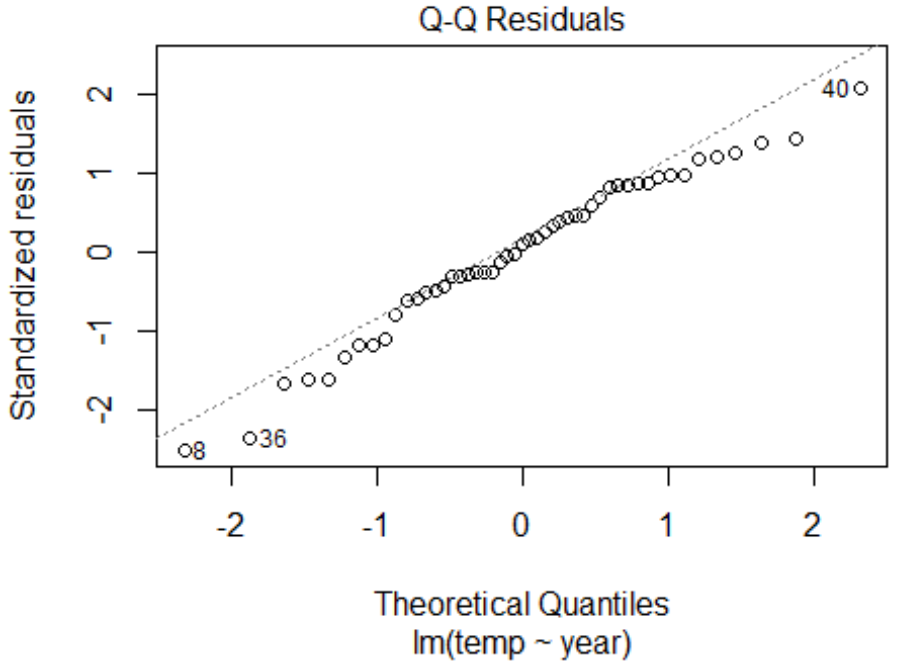

```r
plot(temp~year,aatemp,xlab="Mean temp in Ann Arbor",ylab="Year",
main="temp in ann arbor michigan")
abline(v=1930, lty=5)
lhs<-function(x)ifelse(x<=1930, 1930-x,0)
```

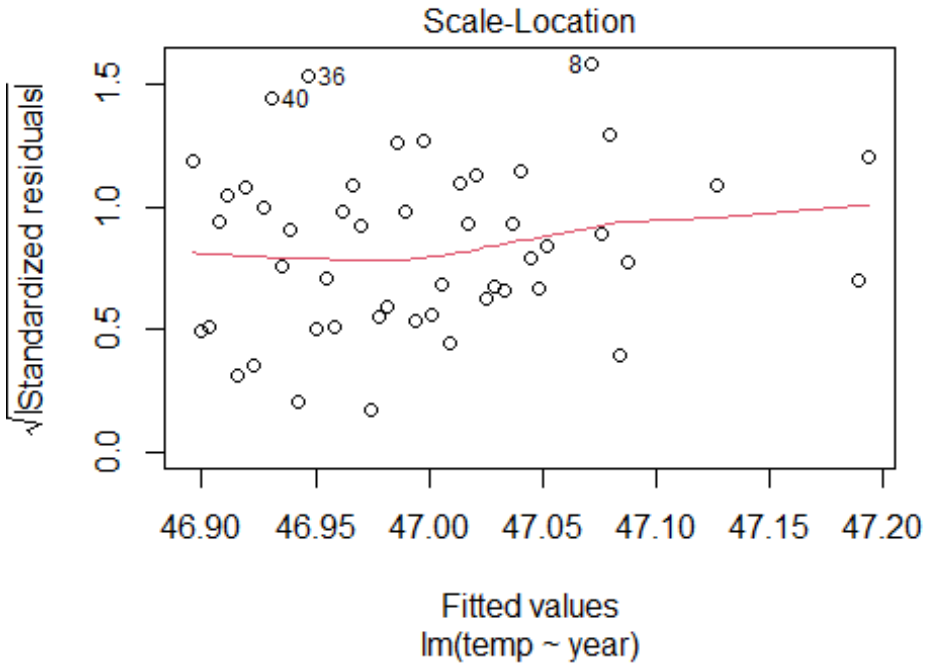
```
rhs<-function(x)ifelse(x>1930, x-1930,0)
gmod<-lm(temp~lhs(year)+rhs(year),aatemp)
x<-seq(1854,2000,by=1)
py<-gmod$coef[1]+gmod$coef[2]*lhs(x)+gmod$coef[3]*rhs(x)
lines(x,py,lty=2)
```

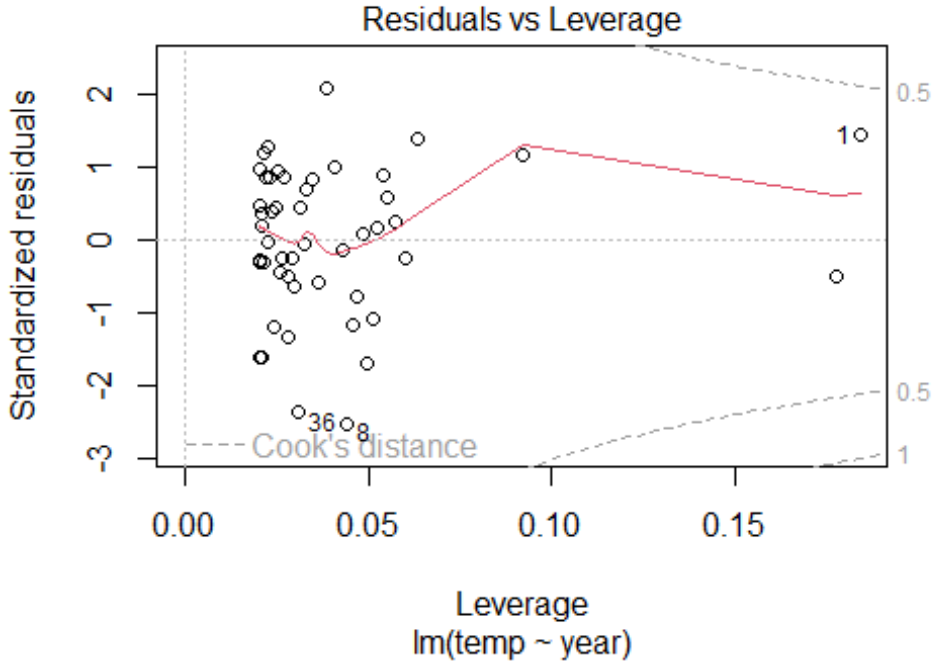



```
l1<-lm(temp~year, aatemp,subset=(year<=1930))
l2<-lm(temp~year, aatemp, subset=(year>1930))
plot(l1)
```


Residuals vs Fitted
Residuals
40
36
8
2
0
-2
-4
46.90
46.95
47.00
47.05
47.10
47.15
47.20
Fitted values
lm(temp ~ year)

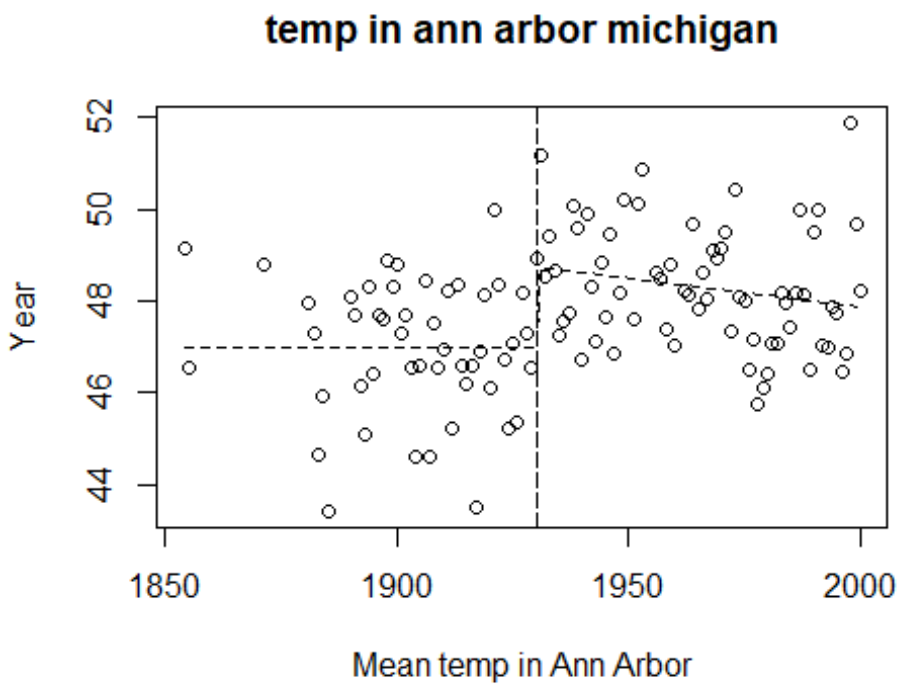

Q-Q Residuals
Standardized residuals
40
36
8
2
1
0
-1
-2
-2
-1
0
1
2
Theoretical Quantiles
lm(temp ~ year)



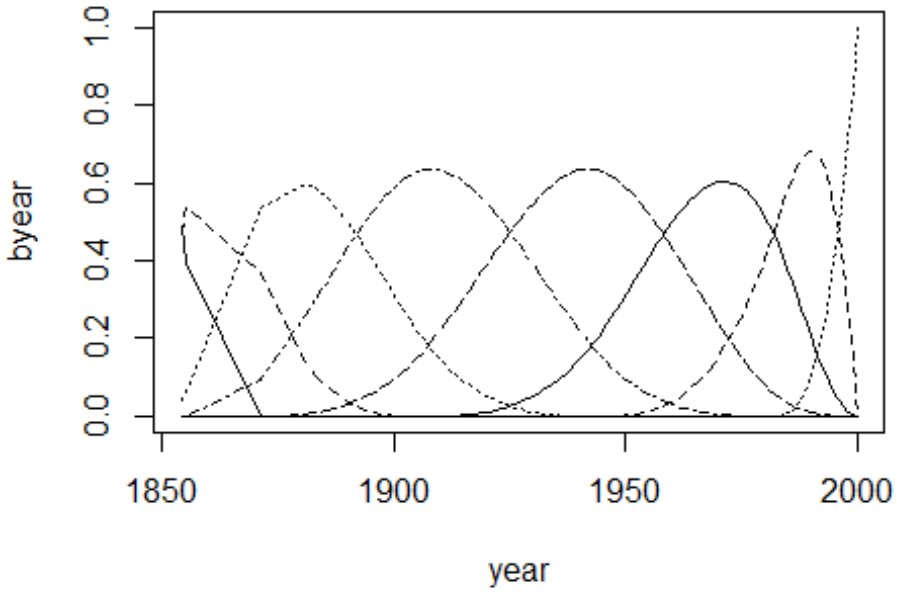



```
plot(temp~year,aatemp,xlab="Mean temp in Ann Arbor",ylab="Year",
main="temp in ann arbor michigan")
abline(v=1930, lty=5)
lhs<-function(x)ifelse(x<=1930, 47.74,0)
```

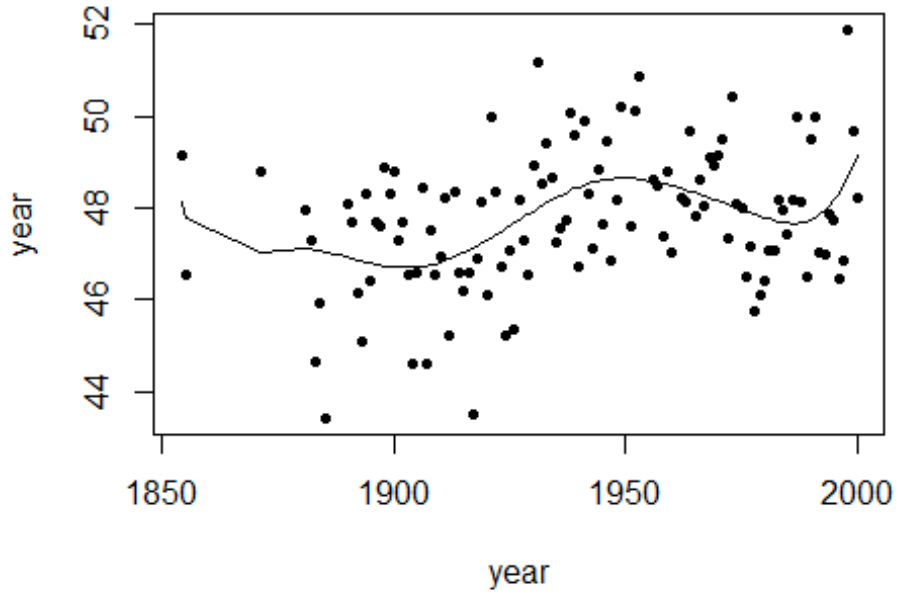
```
rhs<-function(x)ifelse(x>1930, x-1930,0)
gmod<-lm(temp~lhs(year)+rhs(year),aatemp)
x<-seq(1854,2000,by=1)
py<-gmod$coef[1]+gmod$coef[2]*lhs(x)+gmod$coef[3]*rhs(x)
lines(x,py,lty=2)
```




```
# Vemos que en el segundo modelo, la constante es la media de la
temperatura. Parece haber errores basándonos en el modelo.
attach(aatemp)

## The following objects are masked from aatemp (pos = 3):
##
##     temp, year

library(splines)
nudos<-
c(1850,1850,1850,1850,1868.75,1906.25,1943.75,1981.25,2000,2000,2000,2000
)
byear<-splineDesign(nudos,year)
lmodb<-lm(temp~byear-1, data=aatemp)
matplot(year, byear, type="l", col=1)
```




```
matplot(year, cbind(temp,lmodb$fit), type="pl", ylab="year",
pch=20,lty=1,col=1)
```




```
#No hay mejor ajuste respecto al modelo en línea recta.
```

PREGUNTA 9

```r
data(odor)
odor_model = lm(odor ~.^2 + I(temp^2) + I(gas^2) + I(pack^2),data = odor)
summary(odor_model)
```

```
##
## Call:
## lm(formula = odor ~ .^2 + I(temp^2) + I(gas^2) + I(pack^2), data =
odor)
##
## Residuals:
##        1        2        3        4        5        6        7
8
## -20.6250  -6.8750   6.8750  20.6250  15.5000   1.7500  -1.7500 -
15.5000
##        9       10       11       12       13       14       15
##   5.1250 -22.3750  22.3750  -5.1250  -0.3333  -4.3333   4.6667
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -30.667     12.978  -2.363  0.06451 .
## temp         -12.125      7.947  -1.526  0.18761
## gas          -17.000      7.947  -2.139  0.08542 .
## pack         -21.375      7.947  -2.690  0.04332 *
## I(temp^2)     32.083     11.698   2.743  0.04067 *
## I(gas^2)      47.833     11.698   4.089  0.00946 **
## I(pack^2)      6.083     11.698   0.520  0.62524
## temp:gas       8.250     11.239   0.734  0.49588
## temp:pack      1.500     11.239   0.133  0.89903
## gas:pack      -1.750     11.239  -0.156  0.88236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.48 on 5 degrees of freedom
## Multiple R-squared:  0.882,  Adjusted R-squared:  0.6696
## F-statistic: 4.152 on 9 and 5 DF,  p-value: 0.06569
```

```r
# 9 grados de libertad en el numerador y 5 grados de libertad en el
denominador.
odor_model_2 = lm(odor~. + I(temp^2) + I(gas^2) + I(pack^2), data = odor)
summary(odor_model_2)
```

```
##
## Call:
## lm(formula = odor ~ . + I(temp^2) + I(gas^2) + I(pack^2), data = odor)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.625  -9.625  -1.375   4.021  28.875
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -30.667     10.840  -2.829   0.0222 *
## temp         -12.125      6.638  -1.827   0.1052
## gas          -17.000      6.638  -2.561   0.0336 *
## pack         -21.375      6.638  -3.220   0.0122 *
## I(temp^2)     32.083      9.771   3.284   0.0111 *
## I(gas^2)      47.833      9.771   4.896   0.0012 **
## I(pack^2)      6.083      9.771   0.623   0.5509
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.77 on 8 degrees of freedom
## Multiple R-squared:  0.8683, Adjusted R-squared:  0.7695
## F-statistic: 8.789 on 6 and 8 DF,  p-value: 0.003616

anova(odor_model_2,odor_model)

## Analysis of Variance Table
##
## Model 1: odor ~ temp + gas + pack + I(temp^2) + I(gas^2) + I(pack^2)
## Model 2: odor ~ (temp + gas + pack)^2 + I(temp^2) + I(gas^2) +
I(pack^2)
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1      8 2819.9
## 2      5 2526.4  3     293.5 0.1936 0.8965

#8 grados de libertad en el numerador y 5 grados de libertad en el
denominador.
#Aunque el p valor es menor en el primer modelo ambos fallan en rechazar
la hipótesis nula

predict_func <- function(x) predict(odor_model_2, newdata =
data.frame(temp = x[1], gas = x[2], pack = x[3]))
result <- optim(c(0, 0, 0), predict_func)
result$par

## [1] 0.1889666 0.1777468 1.7568309

# [1] 0.1889666 0.1777468 1.7568309 son los valores mínimos.
```

PREGUNTA DE REGRESIÓN LOGÍSTICA

```
ex28_49 <-
read.csv("D:/Materiales_Complementarios_de_Regresion_Logistica/Materiales
_Complementarios_de_Regresion_Logistica/ex28_49.dat")
datos <- ex28_49
head(datos)

##   X diabetes weight waist cholratio
## 1 1        0    121    29       3.6
## 2 2        0    218    46       6.9
```

```
## 3 3        0    256    49       6.2
## 4 4        0    119    33       6.5
## 5 5        1    183    44       8.9
## 6 6        0    190    36       3.6

modelo <- glm(diabetes ~ weight, data = datos, family = binomial)
summary(modelo)

##
## Call:
## glm(formula = diabetes ~ weight, family = binomial, data = datos)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.569051   0.644306  -5.539 3.04e-08 ***
## weight       0.010122   0.003325   3.044  0.00233 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 330.12  on 385  degrees of freedom
## Residual deviance: 321.01  on 384  degrees of freedom
## AIC: 325.01
##
## Number of Fisher Scoring iterations: 4

# Sí, el coeficiente para la variable de peso es significativamente
diferente de cero, ya que el valor p asociado con el coeficiente es menor
que 0.05. Por lo tanto, podemos concluir que hay una relación
significativa entre el peso y la diabetes tipo 2. Además, como el
coeficiente para la variable de peso es positivo, podemos decir que un
mayor peso aumenta las probabilidades de tener diabetes tipo 2.

modelo1 <- glm(diabetes ~ weight + waist, data = datos, family =
binomial)
summary(modelo1)

##
## Call:
## glm(formula = diabetes ~ weight + waist, family = binomial, data =
datos)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.487312   1.081683  -5.997    2e-09 ***
## weight      -0.009302   0.006379  -1.458 0.144786
## waist        0.165519   0.046306   3.574 0.000351 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 330.12  on 385  degrees of freedom
## Residual deviance: 307.99  on 383  degrees of freedom
## AIC: 313.99
##
## Number of Fisher Scoring iterations: 5
```

```
#Observando el modelo vemos que solo contribuye la variable cintura
porque su valor de p es menor a 0.05.
#Al agregar la variable de cintura al modelo, el coeficiente de la
pendiente para la variable de peso disminuye en magnitud y significancia,
lo que indica que la variable de cintura está explicando parte de la
variabilidad en la variable de respuesta que antes se atribuía a la
variable de peso
modelo2 <- glm(diabetes ~ waist, data = datos, family = binomial)
summary(modelo2)
```

```
##
## Call:
## glm(formula = diabetes ~ waist, family = binomial, data = datos)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.9658     1.0097  -5.909 3.45e-09 ***
## waist         0.1086     0.0248   4.380 1.19e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 330.12  on 385  degrees of freedom
## Residual deviance: 310.14  on 384  degrees of freedom
## AIC: 314.14
##
## Number of Fisher Scoring iterations: 5
```

```
# En términos de qué modelo preferir, dependerá de los objetivos
específicos del análisis. Si el objetivo es maximizar la precisión
predictiva, entonces el modelo que incluye tanto el peso como la cintura
como variables explicativas podría ser preferible, ya que incluye más
información. Sin embargo, si el objetivo es simplificar el modelo y
reducir la complejidad, entonces el modelo que solo incluye la variable
de cintura podría ser preferible.
modelo3 <- glm(diabetes ~ waist + cholratio, data = datos, family =
binomial)
summary(modelo3)
```

```
##
## Call:
## glm(formula = diabetes ~ waist + cholratio, family = binomial,
```

```
##     data = datos)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.84900    1.11265  -6.156 7.48e-10 ***
## waist        0.08822    0.02675   3.299 0.000972 ***
## cholratio    0.34153    0.08902   3.837 0.000125 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 330.12  on 385  degrees of freedom
## Residual deviance: 293.01  on 383  degrees of freedom
## AIC: 299.01
##
## Number of Fisher Scoring iterations: 5
```

```
#Ambos coeficientes son significativos, ya que los valores p para ambas variables son menores que 0.05.
exp(coef(modelo3))
```

```
## (Intercept)       waist   cholratio
## 0.001060512 1.092232980 1.407103053
```

```
# Los odds ratios correspondientes son exp(0.088) = 1.092 para la variable de cintura y exp(0.342) = 1.407 para la variable de relación colesterol/HDL. En otras palabras, por cada unidad de aumento en la cintura, las probabilidades de tener diabetes tipo 2 aumentan en un 9.2%, manteniendo constante la variable de relación colesterol/HDL. Por cada unidad de aumento en la relación colesterol/HDL, las probabilidades de tener diabetes tipo 2 aumentan en un 40.7%, manteniendo constante la variable de cintura.
```