Introduccion a ML. Ejercicios

Laura Sudupe Medinilla

17/2/2021

```
# Función para cargar todos los paquetes necesarios
LoadLibraries <- function() {
  myLibraries <- c("faraway", "ggplot2", "dplyr", "cowplot", "scales", "aplot",
                   "XML", "httr", "RCurl")
  invisible(lapply(myLibraries, library, character.only = TRUE))
}
LoadLibraries()
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

EJERCICIOS FARADAY.

1. The dataset teengamb concerns a study of teenage gambling in Britain. Make a numerical and graphical summary of the data, commenting on any features that you find interesting. Limit the output you present to a quantity that a busy reader would find sufficient to get a basic understanding of the data.

```
# Cargamos el paquete
data <- teengamb
rownames(data) <- c(1:nrow(data))
```

```
#Vamos a ver los datos que lo componen
head(data)
```

```
##   sex status income verbal gamble
## 1   1     51   2.00      8    0.0
## 2   1     28   2.50      8    0.0
## 3   1     37   2.00      6    0.0
## 4   1     28   7.00      4    7.3
## 5   1     65   2.00      8   19.6
## 6   1     61   3.47      6    0.1
```

```r
dim(data) #número de filas y columnas
```

```
## [1] 47  5
```

```r
str(data) #estructura de las variables
```

```
## 'data.frame':    47 obs. of  5 variables:
##  $ sex   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ status: int  51 28 37 28 65 61 28 27 43 18 ...
##  $ income: num  2 2.5 2 7 2 3.47 5.5 6.42 2 6 ...
##  $ verbal: int  8 8 6 4 8 6 7 5 6 7 ...
##  $ gamble: num  0 0 0 7.3 19.6 0.1 1.45 6.6 1.7 0.1 ...
```

Vemos que tenemos 5 columnas y 47 filas. Todos los valores son numericos, vemos que la variable sex es categorica

```r
data$sex <- factor(data$sex, levels = c(0,1), labels = c("man", "woman"))
summary(data)
```

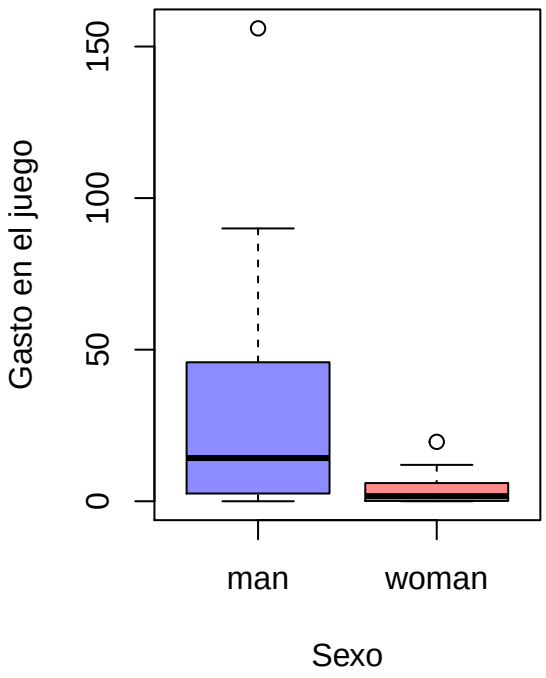
```
##     sex         status          income           verbal          gamble      
##  man  :28   Min.   :18.00   Min.   : 0.600   Min.   : 1.00   Min.   :  0.0  
##  woman:19   1st Qu.:28.00   1st Qu.: 2.000   1st Qu.: 6.00   1st Qu.:  1.1  
##             Median :43.00   Median : 3.250   Median : 7.00   Median :  6.0  
##             Mean   :45.23   Mean   : 4.642   Mean   : 6.66   Mean   : 19.3  
##             3rd Qu.:61.50   3rd Qu.: 6.210   3rd Qu.: 8.00   3rd Qu.: 19.4  
##             Max.   :75.00   Max.   :15.000   Max.   :10.00   Max.   :156.0
```

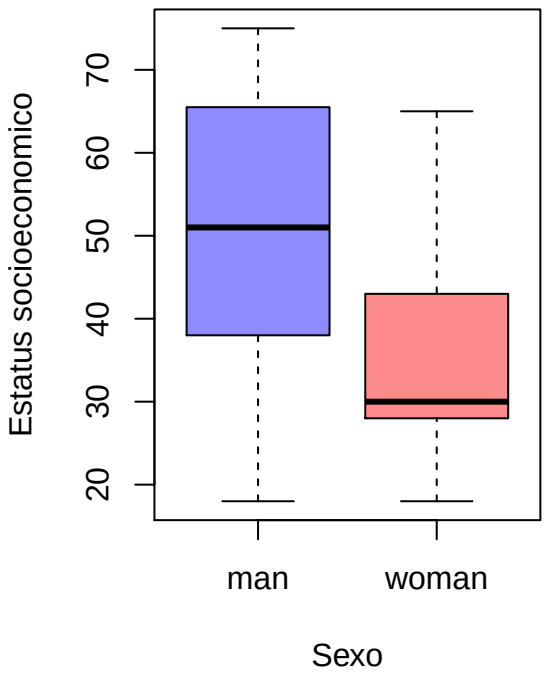
```r
sum(is.na(data))
```

```
## [1] 0
```

Vamos a realizar algunas representaciónes para estudiar los datos

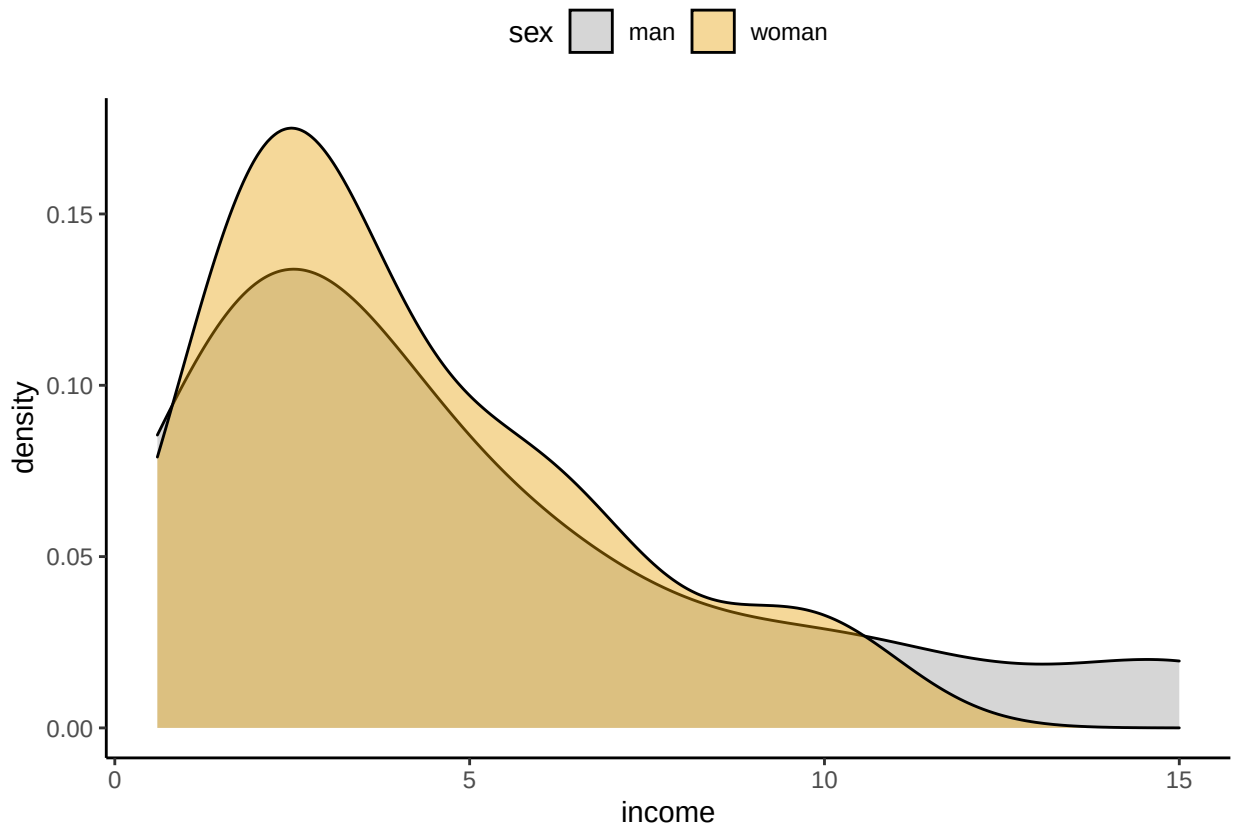
```r
#Veamos a ver los valores de gamble y status  segun el sexo

par(mfrow=c(1,2))
boxplot(data$gamble ~ data$sex, xlab= "Sexo", ylab="Gasto en el juego",
        col= c(col= rgb(red = 0, green = 0, blue = 1, alpha= 0.45),
               col= rgb(red = 1, green = 0, blue = 0, alpha = 0.45)))

boxplot(data$status ~ data$sex, xlab= "Sexo", ylab="Estatus socioeconomico",
        col= c(col= rgb(red = 0, green = 0, blue = 1, alpha= 0.45),
               col= rgb(red = 1, green = 0, blue = 0, alpha = 0.45)))
```





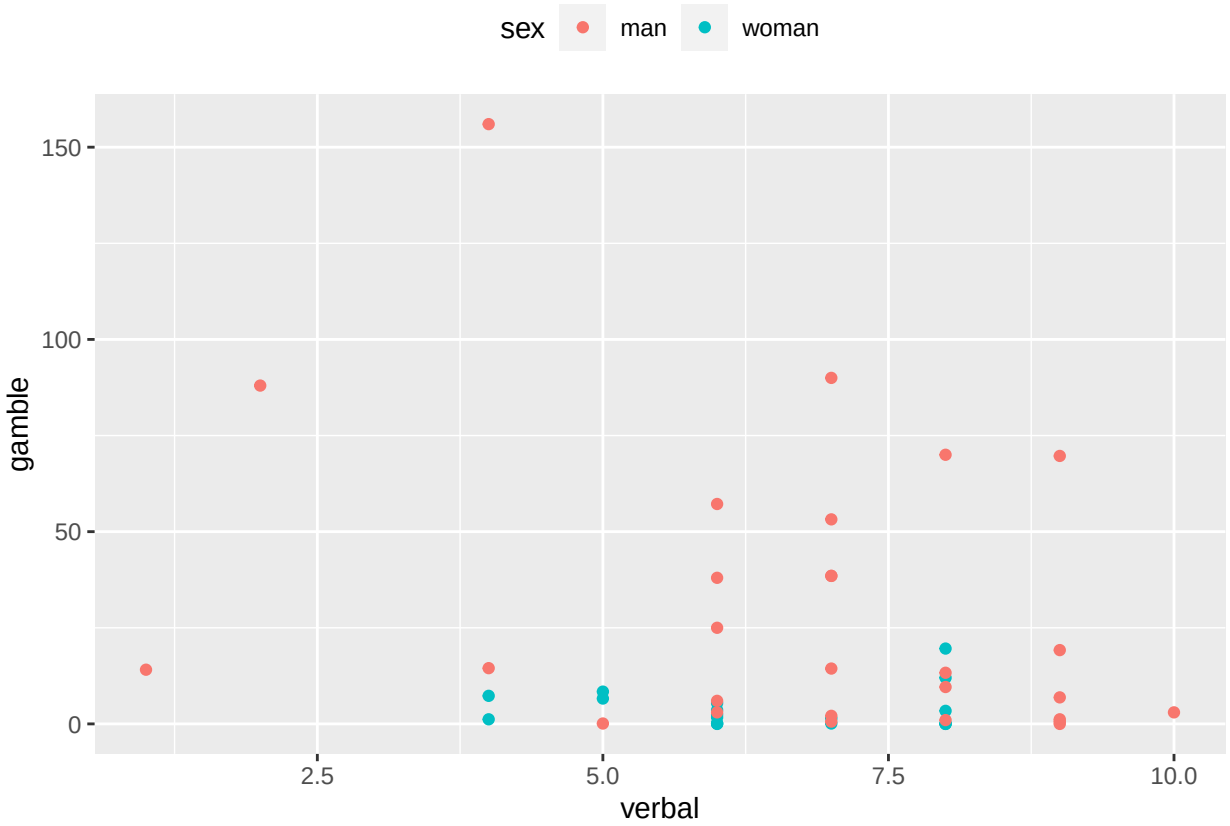
Vemos que hay una diferencia considerable, para el grupo de los hombres tenemos mayores valores tanto en el gasto en el juego como en el estatus socioeconomico

```
#Veamos como se distribuye los ingresos dependiendo el sexo
data%>%
ggplot(aes(x = income, fill = sex)) + geom_density(alpha=0.4) +
scale_fill_manual(values=c("#999999", "#E69F00", "#56B4E9")) +
theme_classic() + theme(legend.position="top")
```



Vemos que hay mayor densidad de mujeres que tienen ingresos mas bajos

```
#Vamos a ver la relación entre la puntuación verbal y el gasto en el juego por
#sexos

ggplot(data, aes(x= verbal, y=gamble, color = sex)) + geom_point()+
  theme(legend.position = "top", legend.direction = "horizontal")
```



Observamos que para valores de puntuación verbal entre 4 y 8 las mujeres tienen un gasto mensual menor en el juego.

2. The dataset uswages is drawn as a sample from the Current Population Survey in 1988. Make a numerical and graphical summary of the data as in the previous question.

```
data2 <- uswages
head(data2)
```

```
##         wage educ exper race smsa ne mw so we pt
## 6085  771.60   18    18    0    1  1  0  0  0  0
## 23701 617.28   15    20    0    1  0  0  0  1  0
## 16208 957.83   16     9    0    1  0  0  1  0  0
## 2720  617.28   12    24    0    1  1  0  0  0  0
## 9723  902.18   14    12    0    1  0  1  0  0  0
## 22239 299.15   12    33    0    1  0  0  0  1  0
```

```
summary(data2)
```

```
##       wage             educ           exper           race
##  Min.   :  50.39   Min.   : 0.00   Min.   :-2.00   Min.   :0.000
##  1st Qu.: 308.64   1st Qu.:12.00   1st Qu.: 8.00   1st Qu.:0.000
##  Median : 522.32   Median :12.00   Median :15.00   Median :0.000
##  Mean   : 608.12   Mean   :13.11   Mean   :18.41   Mean   :0.078
##  3rd Qu.: 783.48   3rd Qu.:16.00   3rd Qu.:27.00   3rd Qu.:0.000
```

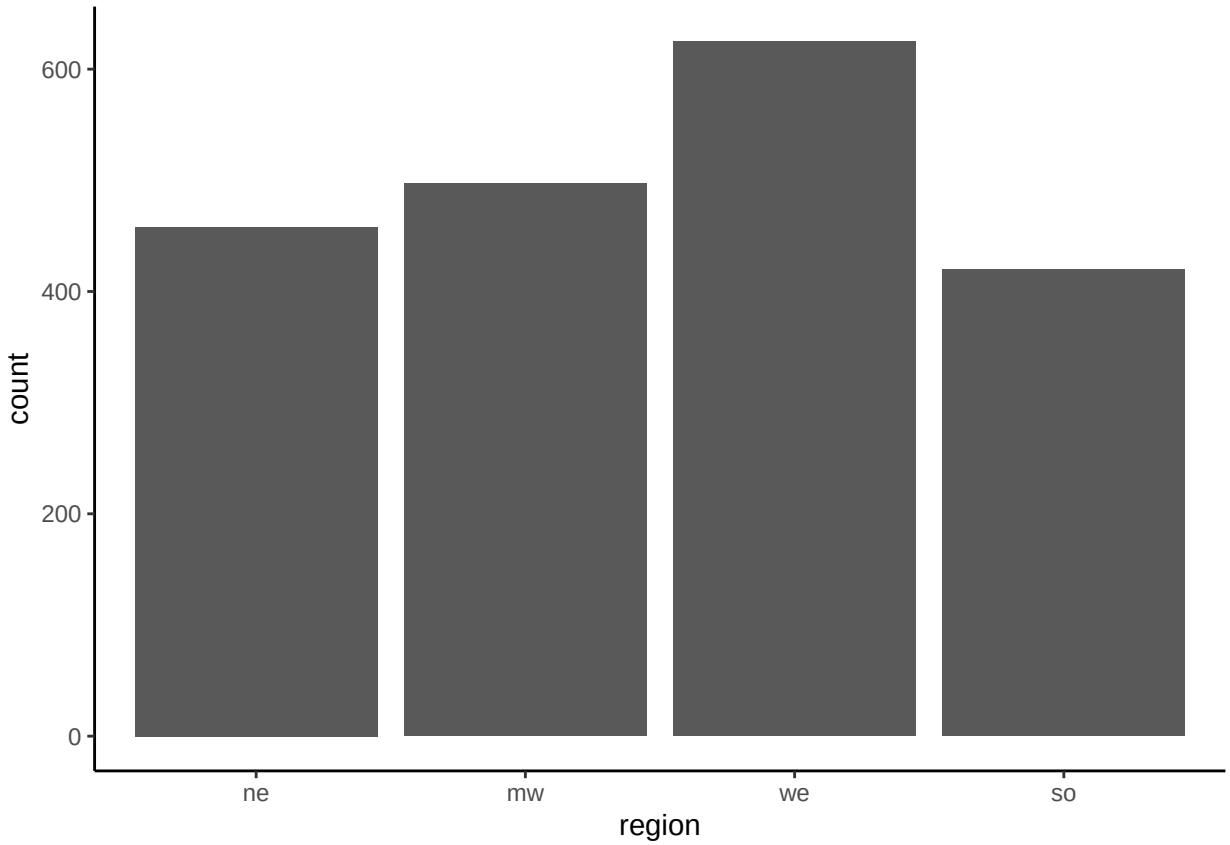
```
##  Max.   :7716.05   Max.   :18.00   Max.   :59.00   Max.   :1.000
##       smsa             ne              mw              so
##  Min.   :0.000   Min.   :0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.000   Median :0.000   Median :0.0000   Median :0.0000
##  Mean   :0.756   Mean   :0.229   Mean   :0.2485   Mean   :0.3125
##  3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:1.0000
##  Max.   :1.000   Max.   :1.000   Max.   :1.0000   Max.   :1.0000
##        we              pt
##  Min.   :0.00   Min.   :0.0000
##  1st Qu.:0.00   1st Qu.:0.0000
##  Median :0.00   Median :0.0000
##  Mean   :0.21   Mean   :0.0925
##  3rd Qu.:0.00   3rd Qu.:0.0000
##  Max.   :1.00   Max.   :1.0000
```

```
#Creamos una columna region y añadimos el distrito
data2$mw <- ifelse(data2$mw == 1, 2, data2$mw)
data2$so <- ifelse(data2$so == 1, 3, data2$so)
data2$we <- ifelse(data2$we == 1, 4, data2$we)


data2$region <- ifelse(data2$ne == 1, data2$ne, data2$mw)
data2$region <- ifelse(data2$region == 0, data2$we, data2$region)
data2$region <- ifelse(data2$region == 0, data2$so, data2$region)
data2$region <- as.factor(data2$region)
levels(data2$region) <- c("ne", "mw", "we", "so")



str(data2)
```

```
## 'data.frame':    2000 obs. of  11 variables:
##  $ wage  : num  772 617 958 617 902 ...
##  $ educ  : int  18 15 16 12 14 12 16 16 12 12 ...
##  $ exper : int  18 20 9 24 12 33 42 0 36 37 ...
##  $ race  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ smsa  : int  1 1 1 1 1 1 1 1 1 0 ...
##  $ ne    : int  1 0 0 1 0 0 0 0 0 0 ...
##  $ mw    : num  0 0 0 0 2 0 0 2 0 2 ...
##  $ so    : num  0 0 3 0 0 0 3 0 0 0 ...
##  $ we    : num  0 4 0 0 0 4 0 0 4 0 ...
##  $ pt    : int  0 0 0 0 0 0 1 1 1 0 ...
##  $ region: Factor w/ 4 levels "ne","mw","we",..: 1 4 3 1 2 4 3 2 4 2 ...
```

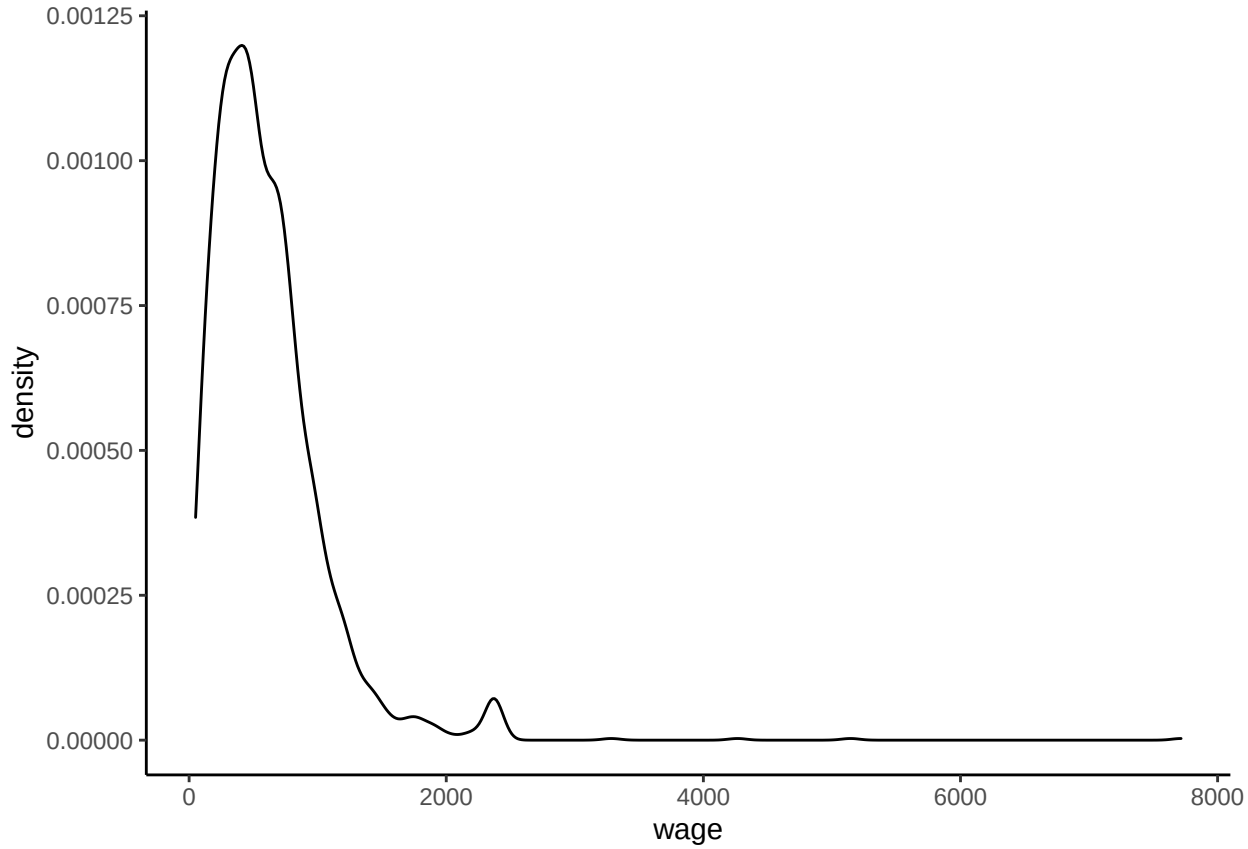
Vamos a realizar algunas modificaciones para mayor entendimiento de los datos.

```
ggplot(data2, aes(region), width=2) + geom_bar(aes(fill=race)) + theme_classic()
```

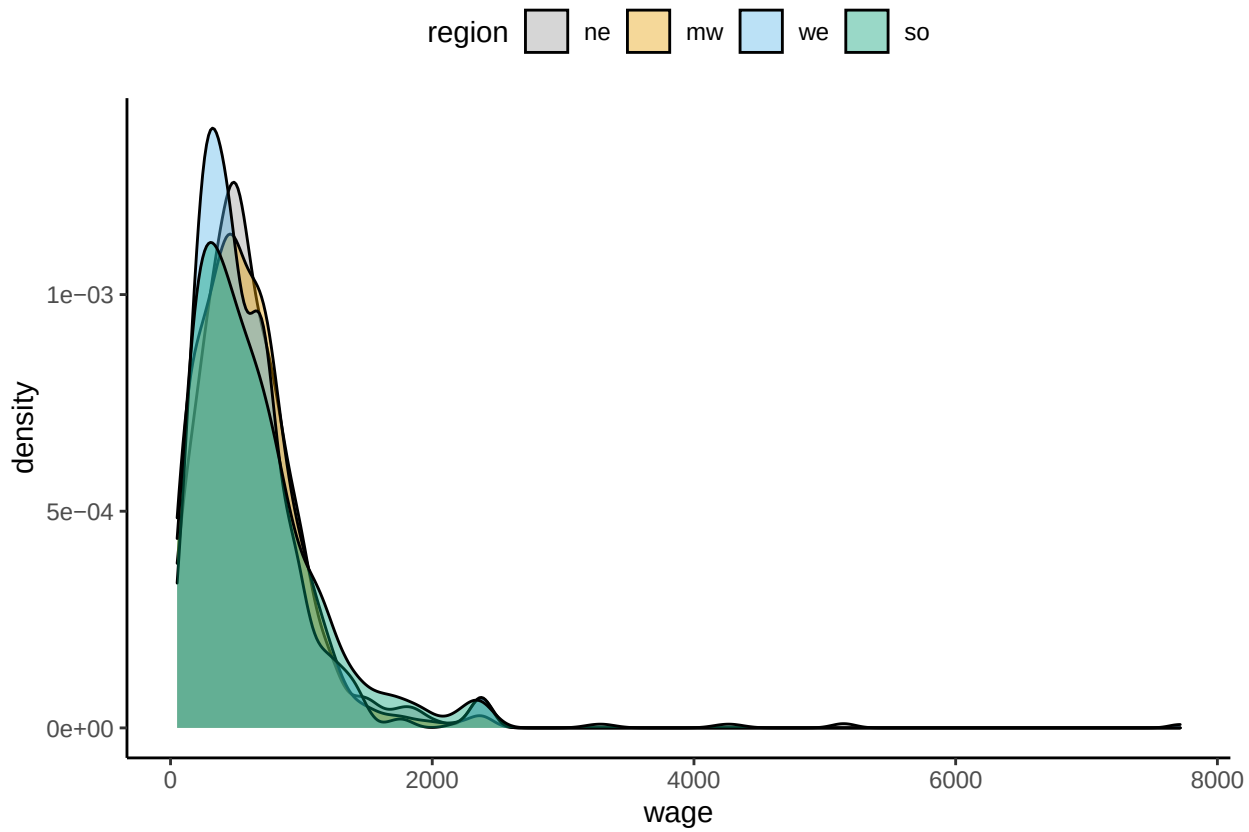


Se puede ver que la proporcion de personas de raza negra y blanca esta muy desajustado

```
#Veamos como se distribuye la cantidad de ganancias dependiendo de la raza y de
#la zona

par(mfrow=c(2,1))
data2%>%
ggplot(aes(x = wage, fill = race)) + geom_density(alpha=0.4) +
scale_fill_manual(values=c("#999999", "#E69F00", "#56B4E9")) +
theme_classic() + theme(legend.position="top")
```

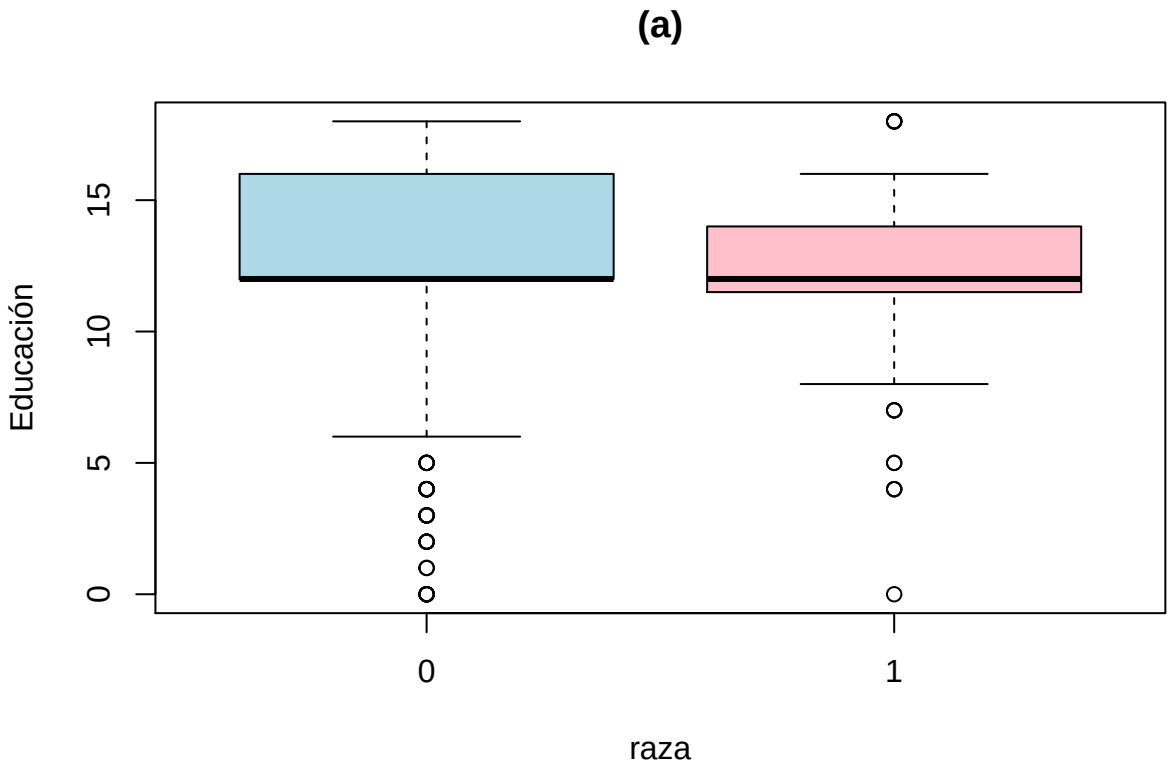



```
data2%>%
ggplot(aes(x = wage, fill = region)) + geom_density(alpha=0.4) +
scale_fill_manual(values=c("#999999", "#E69F00", "#56B4E9", "#009E73")) +
theme_classic() + theme(legend.position="top")
```



Vemos que hay mayor densidad en cuanto a menor ingresos en la raza negra. A su vez, en cuanto a los distritos se mantiene bstante igualado con una densidad mayor en menor ingresos en el sur.

```
boxplot(educ ~ race, data=data2,
        col=c('lightblue', 'pink'),
        xlab='raza', main='(a)',
        ylab='Educación')
```



Aunque tenemos outliers, vemos que los niveles de educación estan por debajo en la raza negra

EJERCICIOS CARMONA.

3. Consideremos el problema de tráfico planteado en el apartado 1.2 de este capítulo, con la variable independiente densidad y la variable dependiente raíz cuadrada de la velocidad. Con los datos proporcionados en la tabla 1.1

```
dens <- c(12.7,17.0,66.0,50.0,87.8,81.4,75.6,66.2,81.1,62.8,77.0,89.6,
18.3,19.1,16.5,22.2,18.6,66.0,60.3,56.0,66.3,61.7,66.6,67.8)
vel <- c(62.4,50.7,17.1,25.9,12.4,13.4,13.7,17.9,13.8,17.9,15.8,12.6,
51.2,50.8,54.7,46.5,46.3,16.9,19.8,21.2,18.3,18.0,16.6,18.3)
rvel <- sqrt(vel)
```

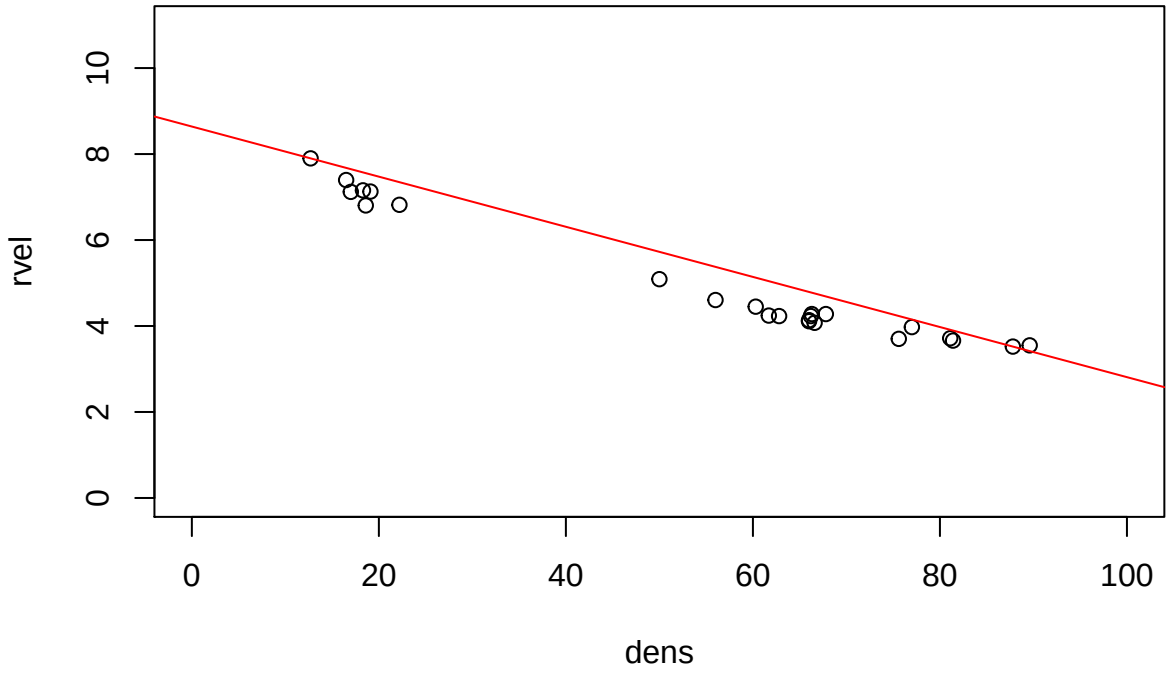
realizar el siguiente proceso:

(a) Dibujar la nube de puntos y la recta que pasa por los puntos (12.7,62.4) y (87.8,12.4). Dibujar el gráfico de los residuos con la densidad y el gráfico con las predicciones. Calcular la suma de cuadrados de los residuos.

```
# Vamos a calcular la ecuación de la recta que pasa por los puntos
x <- c(12.7, 87.8)
y <- sqrt(c(62.4, 12.4))
reg_lin <- lm(y ~ x)
reg_lin
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##      8.6397      -0.0583
```

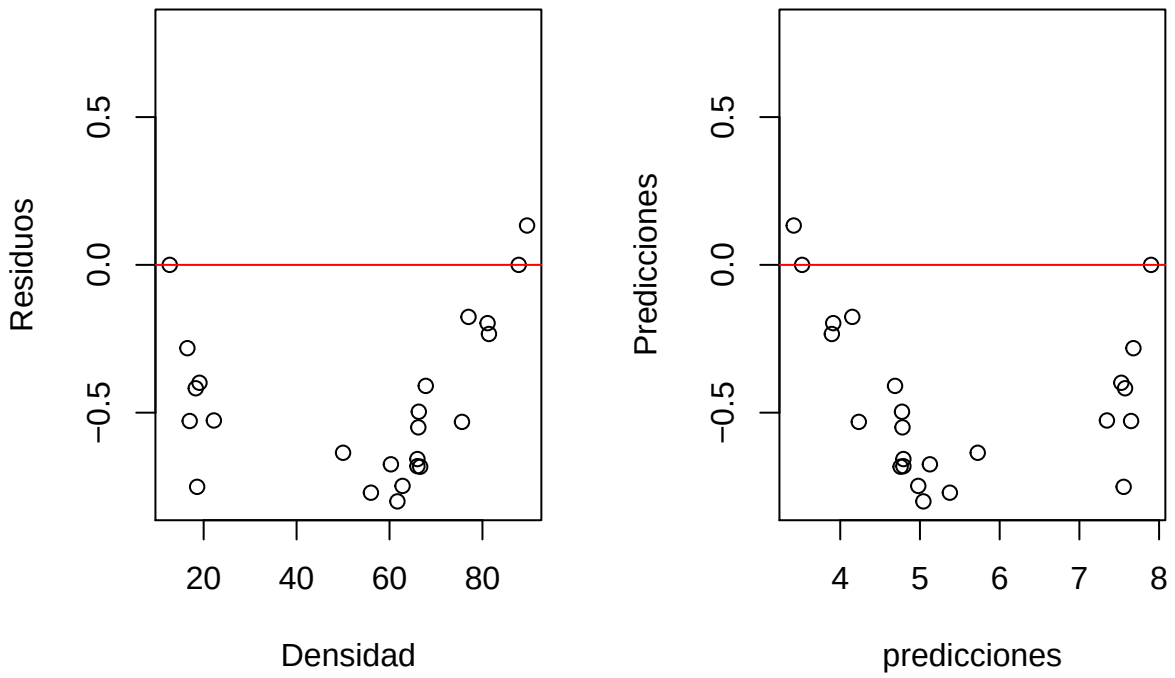
```r
#Dibujamos la nube de puntos y añadimos la recta que pasa por los puntos
plot(rvel~dens,xlim=c(0 ,100), ylim=c(0,11))
abline(reg_lin, col="red")
```




```r
#Dibujamos el grafico de residuos vs densidad y residuos vs predicciones
#Calculamos las predicciones con los coeficientes de la recta que pasa por los
#puntos

predicciones <- reg_lin$coef[1] + reg_lin$coef[2] * dens

#Calculamos los residuos
e <- rvel - predicciones

par(mfrow=c(1,2))
plot(dens, e, xlab="Densidad", ylab="Residuos", ylim=c(-0.8,0.8))
abline(h=0,col="red")
plot(predicciones, e, xlba="Densidad", ylab ="Predicciones", ylim=c(-0.8,0.8))
```

```
## Warning in plot.window(...): "xlba" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlba" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in box(...): "xlba" is not a graphical parameter

## Warning in title(...): "xlba" is not a graphical parameter
```

```
abline(h=0,col="red")
```



Tenemos un patron de dispersión no aleatorio en los residuos, esto indica que no se cumple el supuesto de varianza constante en los errores del modelo. Tenemos una tendencia de abanico.

```
#Calculamos la suma de los cuadrados del residuo
SCE <- sum((e)^2)
SCE
```

```
## [1] 6.689836
```

(b) Hallar la recta de regresión simple. Dibujar el gráfico de los residuos con la densidad y el gráfico con las predicciones. Calcular la suma de cuadrados de los residuos.

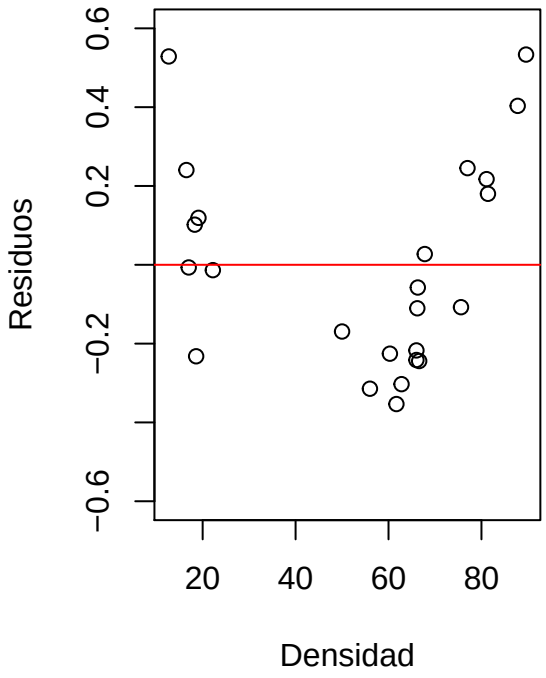
```
#Hallamos la recta
reg_lin_simple <- lm(rvel ~ dens)
reg_lin_simple
```

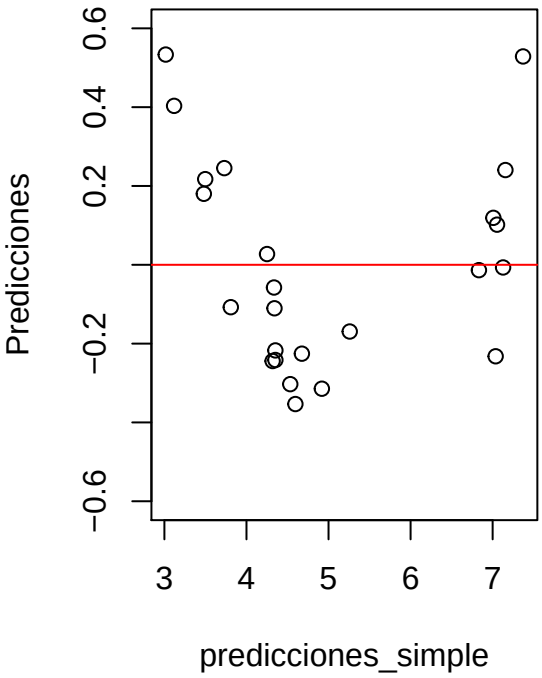
```
##
## Call:
## lm(formula = rvel ~ dens)
##
## Coefficients:
## (Intercept)         dens
##     8.08981     -0.05663
```

```
#Hallamos los residuos y las predicciones
residuos_simple <- reg_lin_simple$residual
predicciones_simple <- rvel - residuos_simple

#Dibujamos las graficas
par(mfrow=c(1,2))
plot(dens, residuos_simple, xlab="Densidad", ylab="Residuos", ylim=c(-0.6,0.6))
abline(h=0,col="red")
plot(predicciones_simple, residuos_simple, xlba="Densidad", ylab ="Predicciones"
     , ylim=c(-0.6,0.6))
```

```
## Warning in plot.window(...): "xlba" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlba" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in box(...): "xlba" is not a graphical parameter

## Warning in title(...): "xlba" is not a graphical parameter
```

```
abline(h=0,col="red")
```






```
#Calculamos suma de cuadrados de los residuos
SCE_simple <- sum((residuos_simple)^2)
SCE_simple
```

```
## [1] 1.591218
```

Estan mas cerca del 0 pero siguen teniendo una tendencia. La suma de los cuadrados de los residuos se ha reducido

(c) Mejorar el modelo anterior considerando una regresión parabólica. Dibujar el gráfico de los residuos con la densidad y el gráfico con las predicciones. Calcular la suma de cuadrados de los residuos.

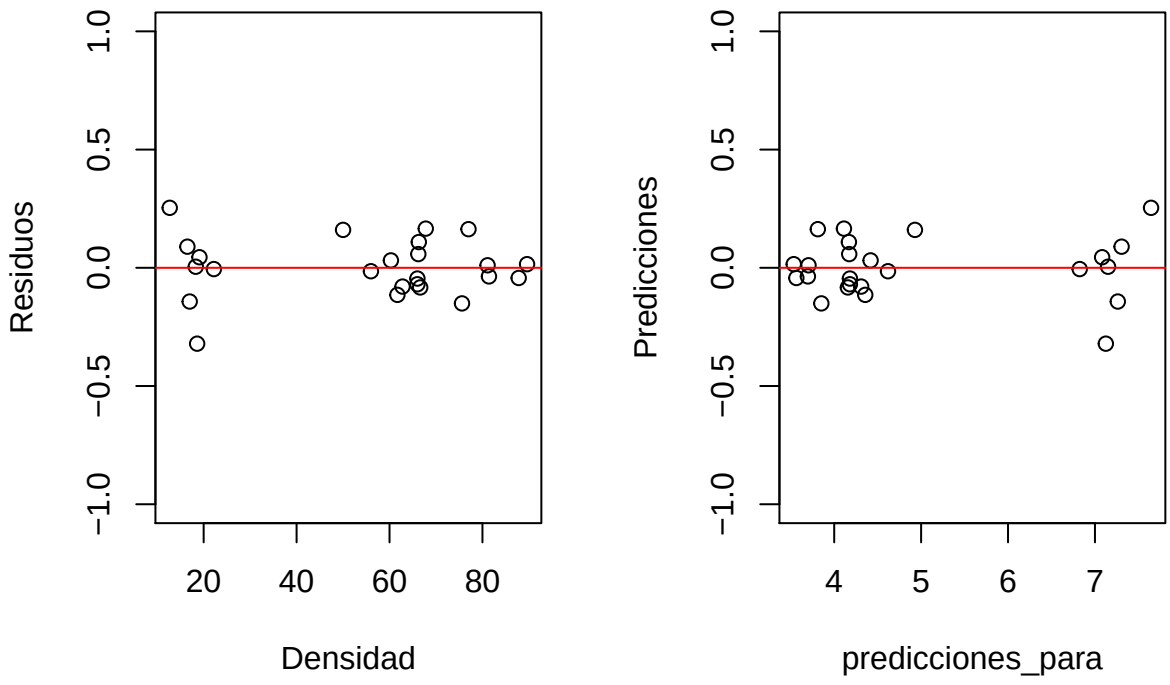
```
#Ajustamos a una regresion polinomica
reg_parabolica <- lm(rvel ~ poly(dens,2))
reg_parabolica
```

```
##
## Call:
## lm(formula = rvel ~ poly(dens, 2))
##
## Coefficients:
##    (Intercept)  poly(dens, 2)1  poly(dens, 2)2
##          5.007          -6.994           1.113
```

```r
#Hallamos los residuos y las predicciones
residuos_para <- reg_parabolica$residual
predicciones_para <- rvel - residuos_para

#Dibujamos las graficas
par(mfrow=c(1,2))
plot(dens, residuos_para, xlab="Densidad", ylab="Residuos", ylim=c(-1,1))
abline(h=0,col="red")
plot(predicciones_para, residuos_para, xlba="Densidad", ylab ="Predicciones"
     , ylim=c(-1,1))
```

```
## Warning in plot.window(...): "xlba" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlba" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in box(...): "xlba" is not a graphical parameter

## Warning in title(...): "xlba" is not a graphical parameter
```

```r
abline(h=0,col="red")
```



```r
#Calculamos suma de cuadrados de los residuos
SCE_para <- sum((residuos_para)^2)
SCE_para
```

```
## [1] 0.3534143
```

(d) Calcular la capacidad de la carretera o punto de máximo flujo. Recordar que flujo = vel × densidad.

Para encontrar el punto maximo, tenemos que representar la funcion del flujo respecto de la densidad. Una vez que tenemos la curva, encontrar cuando la derivada vale 0. No se como tendria que hacerlo en R.

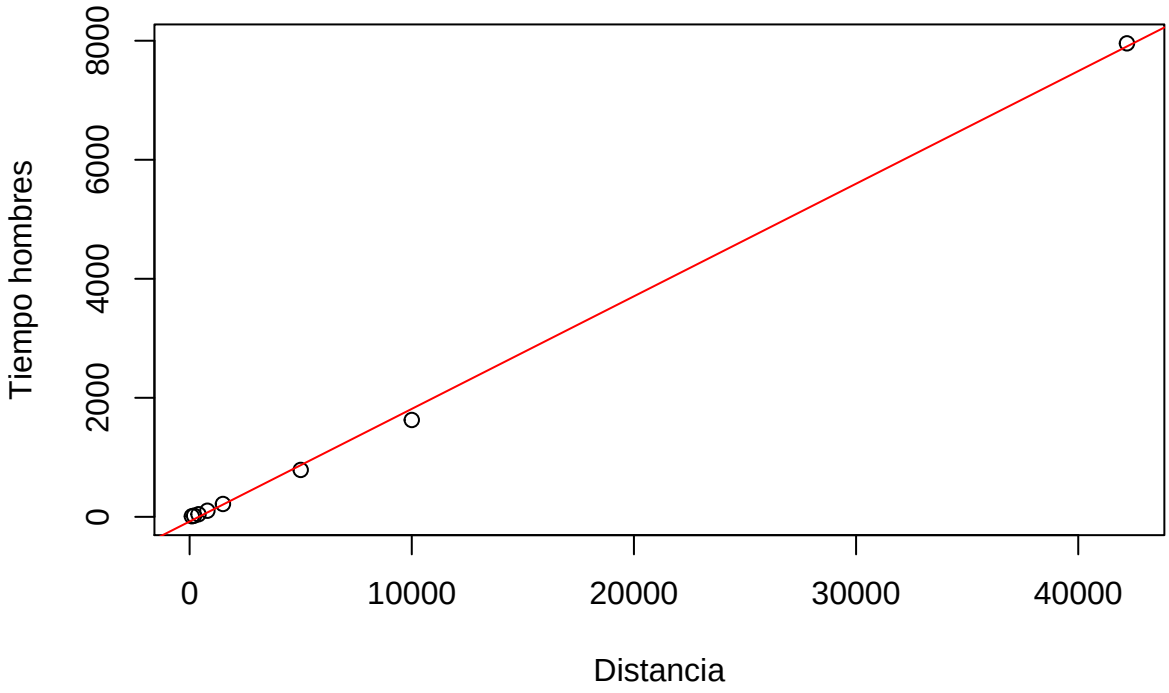
4. La siguiente tabla contiene los mejores tiempos conseguidos en algunas pruebas de velocidad en los Juegos Olímpicos de Atlanta:

```r
distancia <- c(100, 200, 400, 800, 1500, 5000, 10000, 42192)
tiempo_h <- c(9.84, 19.32, 43.19, 102.58, 215.78, 787.96, 1627.34, 7956)
tiempo_m <- c(10.94, 22.12, 48.25, 117.73, 240.83, 899.88, 1861.63, 8765.00)
```

Si tomamos como variable regresora o independiente la distancia (metros) y como variable respuesta o dependiente el tiempo (segundos):

(a) Calcular la recta de regresión simple con los datos de los hombres y dibujarla. Dibujar el gráfico de los residuos con la distancia y el gráfico con las predicciones. Calcular la suma de cuadrados de los residuos y el R2.

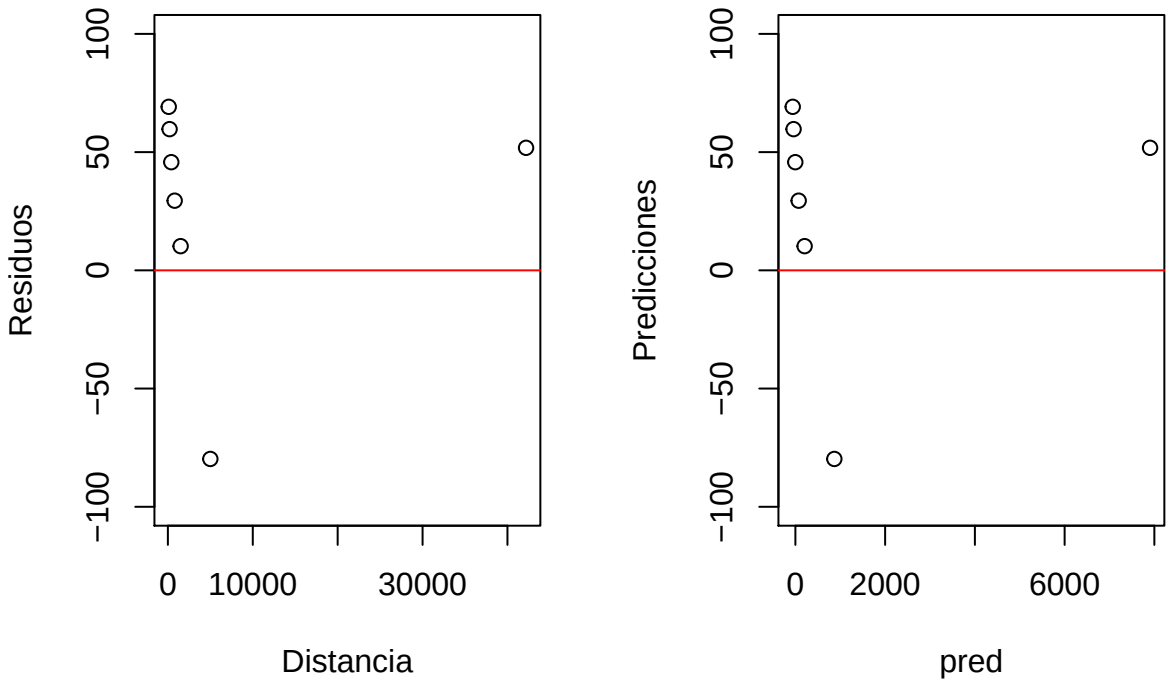
```r
#Recta de regresión
reg_sim_hombres <- lm(tiempo_h ~  distancia)
plot(distancia, tiempo_h, xlab= "Distancia",
                      ylab="Tiempo hombres")
abline(reg_sim_hombres, col="red")
```




```
#Hallamos los residuos y las predicciones
e <- reg_sim_hombres$residuals
pred <- tiempo_h - e

#Dibujamos las graficas
par(mfrow=c(1,2))
plot(distancia, e, xlab="Distancia", ylab="Residuos", ylim=c(-100,100))
abline(h=0,col="red")
plot(pred, e, xlba="Distancia", ylab ="Predicciones"
     , ylim=c(-100,100))
```

```
## Warning in plot.window(...): "xlba" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlba" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in box(...): "xlba" is not a graphical parameter

## Warning in title(...): "xlba" is not a graphical parameter
```

```
abline(h=0,col="red")
```




```
#Calculamos suma de cuadrados de los residuos
SCE_h <- sum((e)^2)
SCE_h
```

```
## [1] 55189
```

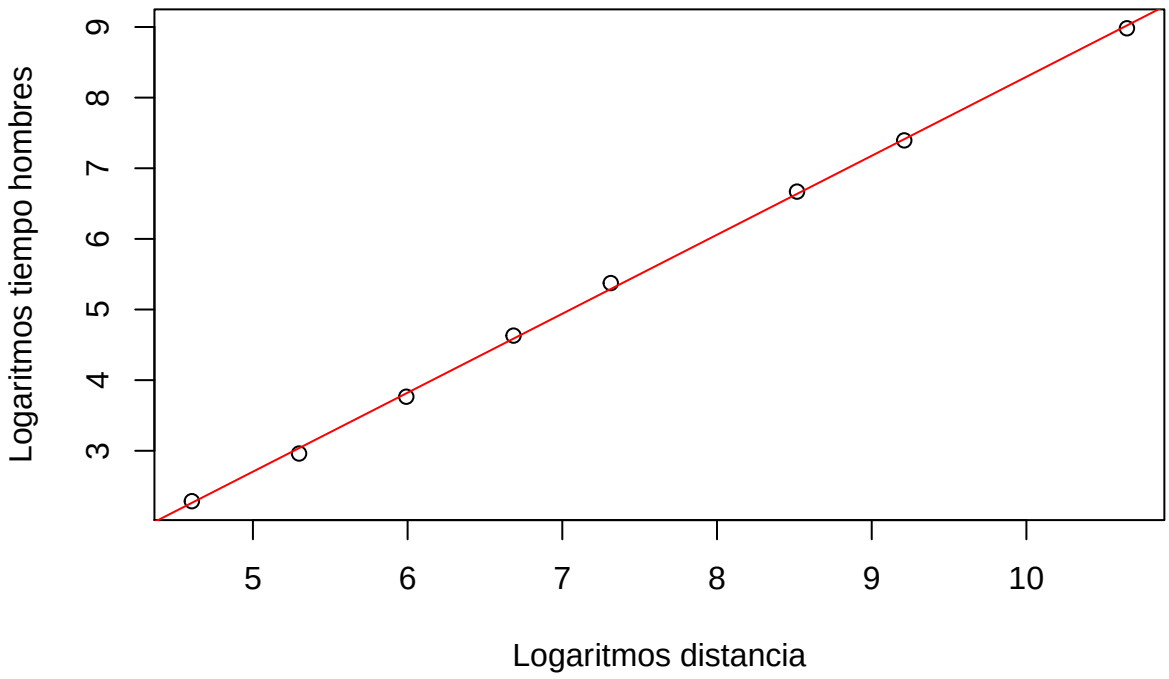
```
#R2 es "Multiple R-squared"
summary(reg_sim_hombres)
```

```
##
## Call:
## lm(formula = tiempo_h ~ distancia)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -186.35  -12.27   37.60   53.79   69.16
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -78.234245  38.827257  -2.015   0.0905 .
## distancia     0.189193   0.002514  75.256 3.71e-10 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 95.91 on 6 degrees of freedom
## Multiple R-squared:  0.9989, Adjusted R-squared:  0.9988
## F-statistic:  5663 on 1 and 6 DF,  p-value: 3.705e-10
```

(b) Repetir el apartado anterior utilizando los logaritmos de las variables tiempo y distancia.

```
#Calculamos los logaritmos
log_distancia <- log(distancia)
log_tiempo_h <- log(tiempo_h)

#Recta de regresión
reg_sim_hombres_log <- lm(log_tiempo_h ~log_distancia)
plot(log_distancia, log_tiempo_h, xlab= "Logaritmos distancia",
                      ylab="Logaritmos tiempo hombres")
abline(reg_sim_hombres_log, col="red")
```

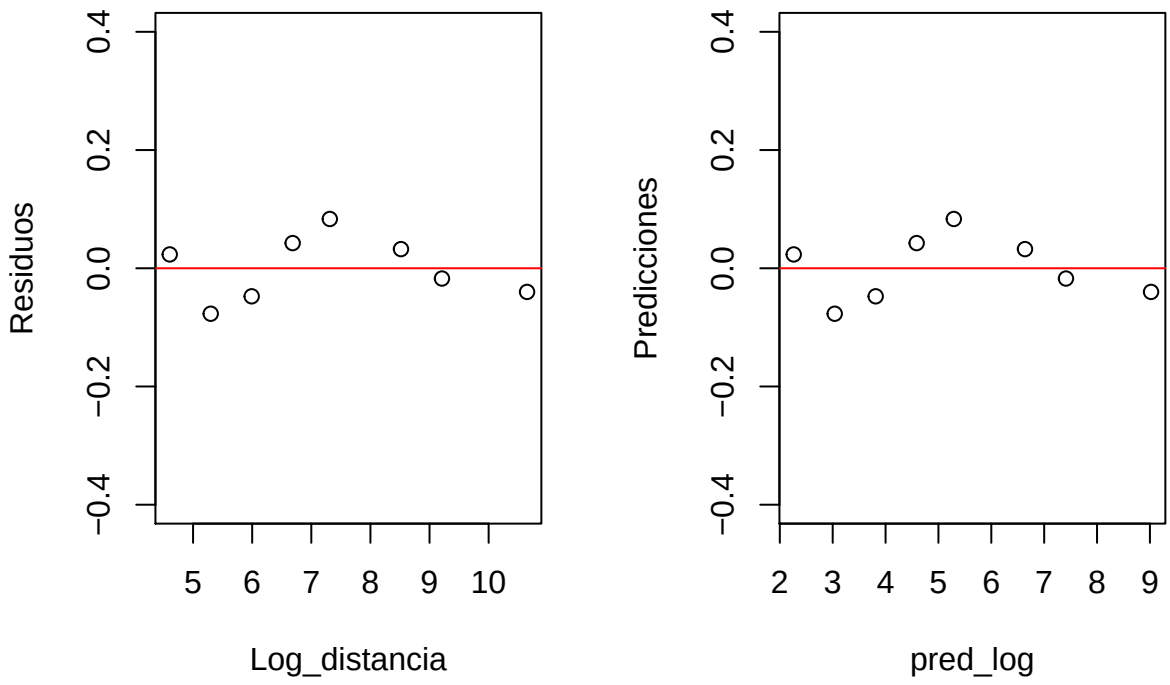



```
#Hallamos los residuos y las predicciones
e_log <- reg_sim_hombres_log$residuals
pred_log <- log_tiempo_h - e_log

#Dibujamos las graficas
par(mfrow=c(1,2))
plot(log_distancia, e_log, xlab="Log_distancia", ylab="Residuos", ylim=c(-0.4,0.4))
```

```
abline(h=0,col="red")
plot(pred_log, e_log, xlba="Log_distancia", ylab ="Predicciones"
     , ylim=c(-0.4,0.4))
```

```
## Warning in plot.window(...): "xlba" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlba" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in box(...): "xlba" is not a graphical parameter

## Warning in title(...): "xlba" is not a graphical parameter
```

```
abline(h=0,col="red")
```




```
#Calculamos suma de cuadrados de los residuos
SCE_h <- sum((e_log)^2)
SCE_h
```

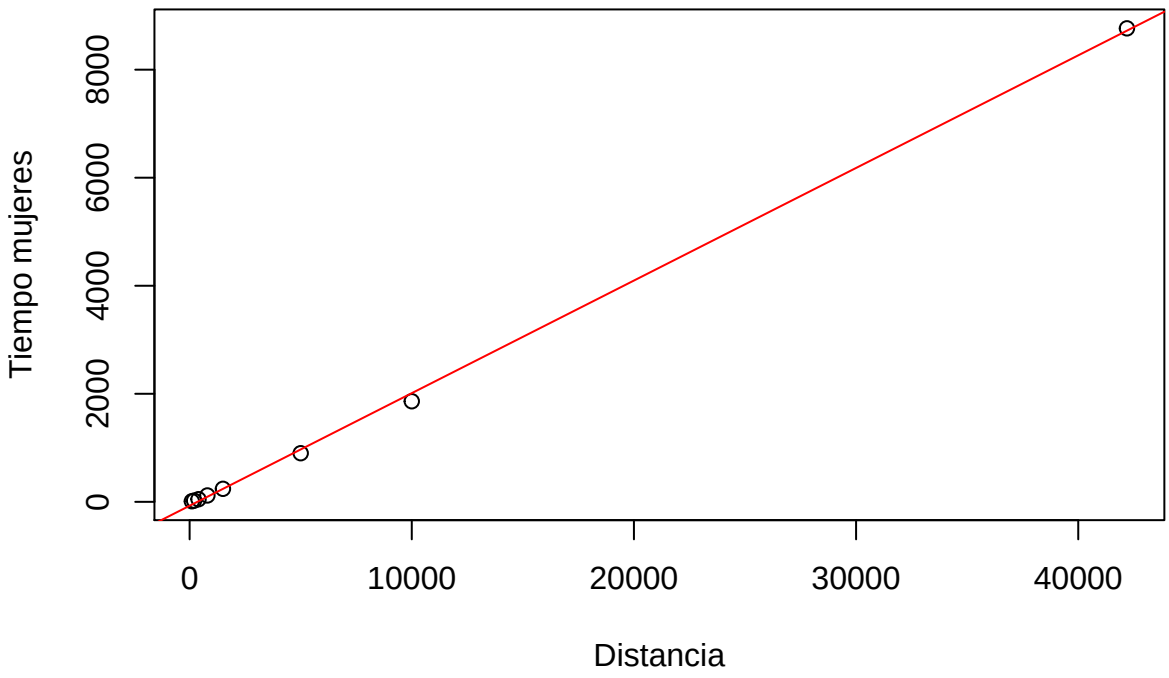
```
## [1] 0.02042962
```

```
#R2 es "Multiple R-squared"
summary(reg_sim_hombres_log)
```

```
##
## Call:
## lm(formula = log_tiempo_h ~ log_distancia)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.076910 -0.041887  0.003026  0.034941  0.083359
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.88596    0.08066  -35.78 3.18e-08 ***
## log_distancia  1.11809    0.01071  104.44 5.19e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05835 on 6 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 1.091e+04 on 1 and 6 DF,  p-value: 5.192e-11
```

(c) Repetir los dos apartados anteriores utilizando los datos de las mujeres.

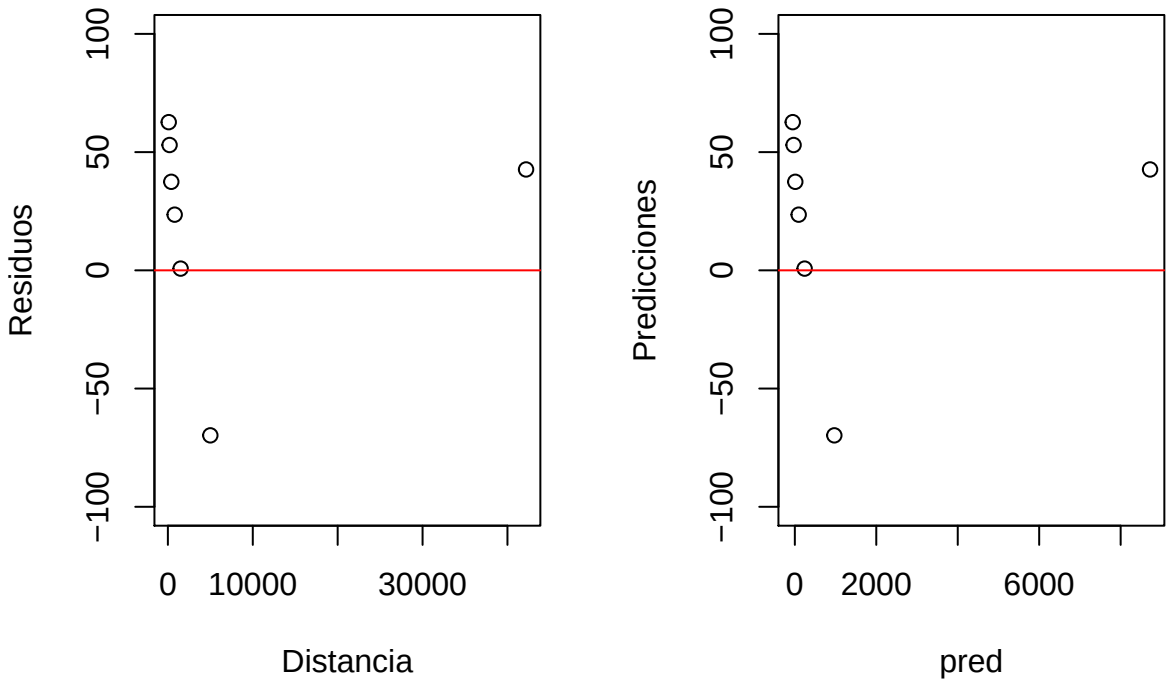
```
#Recta de regresión
reg_sim_mujeres <- lm(tiempo_m ~  distancia)
plot(distancia, tiempo_m, xlab= "Distancia",
                     ylab="Tiempo mujeres")
abline(reg_sim_mujeres, col="red")
```




```
#Hallamos los residuos y las predicciones
e <- reg_sim_mujeres$residuals
pred <- tiempo_m - e

#Dibujamos las graficas
par(mfrow=c(1,2))
plot(distancia, e, xlab="Distancia", ylab="Residuos", ylim=c(-100,100))
abline(h=0,col="red")
plot(pred, e, xlba="Distancia", ylab ="Predicciones"
     , ylim=c(-100,100))
```

```
## Warning in plot.window(...): "xlba" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlba" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in box(...): "xlba" is not a graphical parameter

## Warning in title(...): "xlba" is not a graphical parameter
```

```
abline(h=0,col="red")
```




```
#Calculamos suma de cuadrados de los residuos
SCE_h <- sum((e)^2)
SCE_h
```

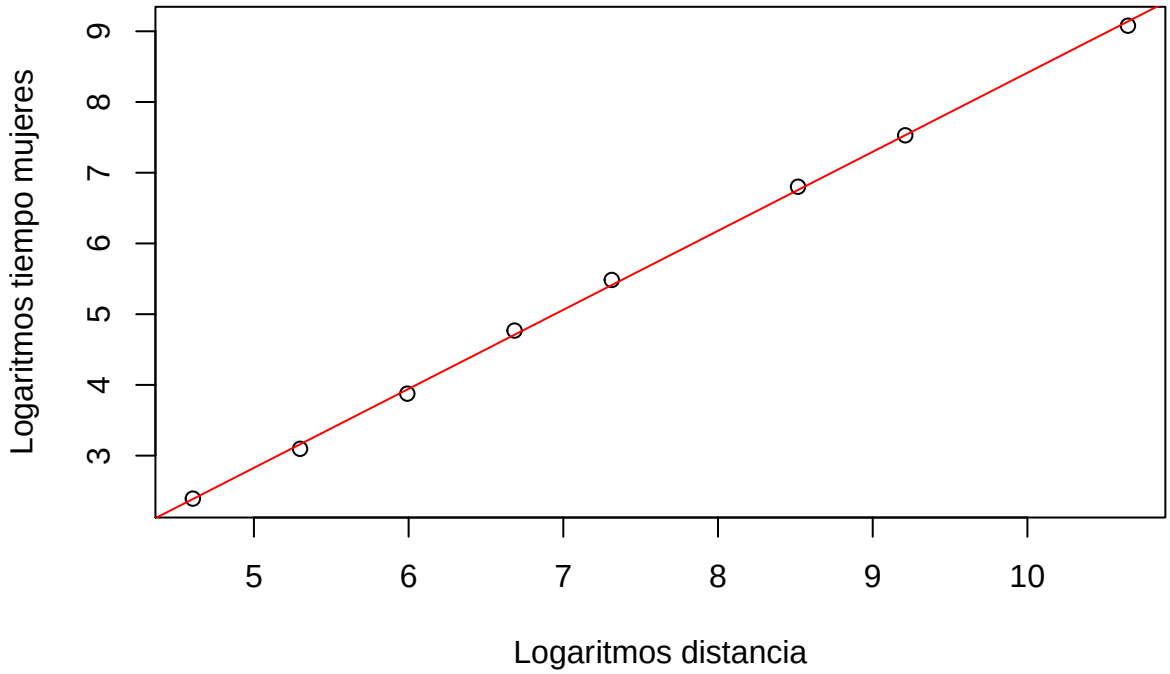
```
## [1] 37973.26
```

```
#R2 es "Multiple R-squared"
summary(reg_sim_mujeres)
```

```
##
## Call:
## lm(formula = tiempo_m ~ distancia)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -150.29  -16.90   30.50   45.25   62.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -72.579108  32.206936  -2.254   0.0651 .
## distancia     0.208450   0.002085  99.960 6.76e-11 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 79.55 on 6 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9993
## F-statistic:  9992 on 1 and 6 DF,  p-value: 6.756e-11
```

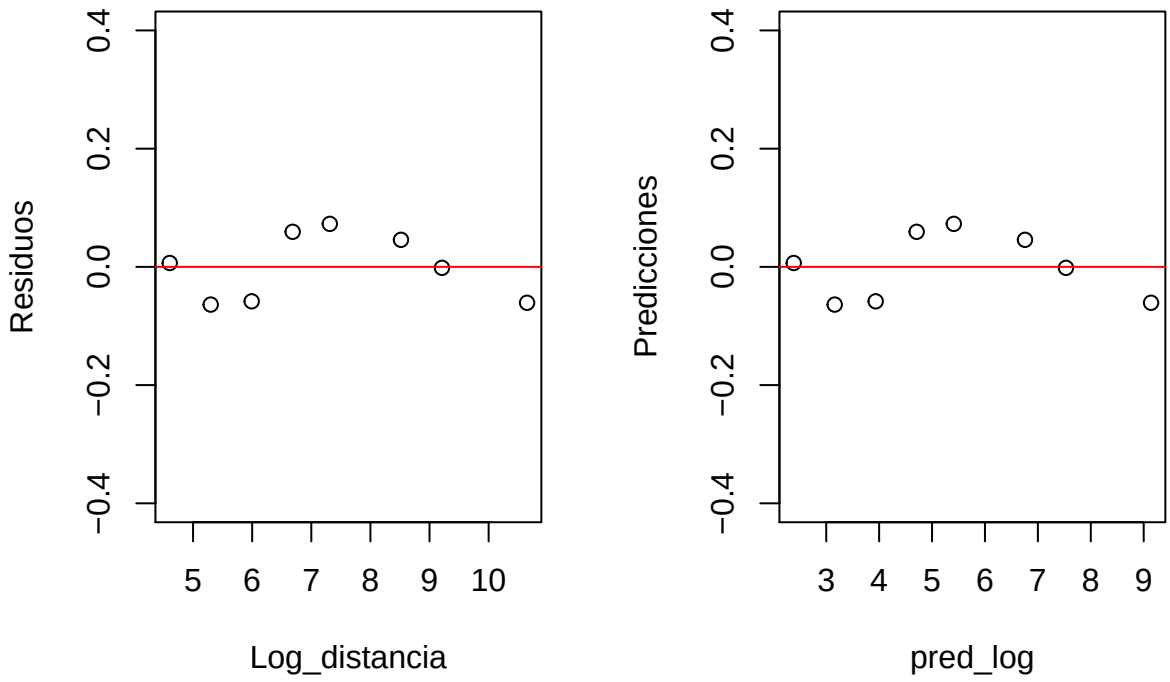
```
#Calculamos los logaritmos
log_tiempo_m <- log(tiempo_m)

#Recta de regresión
reg_sim_mujeres_log <- lm(log_tiempo_m ~log_distancia)
plot(log_distancia, log_tiempo_m, xlab= "Logaritmos distancia",
                      ylab="Logaritmos tiempo mujeres")
abline(reg_sim_mujeres_log, col="red")
```




```
#Hallamos los residuos y las predicciones
e_log <- reg_sim_mujeres_log$residuals
pred_log <- log_tiempo_m - e_log

#Dibujamos las graficas
par(mfrow=c(1,2))
plot(log_distancia, e_log, xlab="Log_distancia", ylab="Residuos", ylim=c(-0.4,0.4))
abline(h=0,col="red")
plot(pred_log, e_log, xlba="Log_distancia", ylab ="Predicciones"
     , ylim=c(-0.4,0.4))
```

```
## Warning in plot.window(...): "xlba" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "xlba" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "xlba" is not a
## graphical parameter

## Warning in box(...): "xlba" is not a graphical parameter

## Warning in title(...): "xlba" is not a graphical parameter
```

```
abline(h=0,col="red")
```




```
#Calculamos suma de cuadrados de los residuos
SCE_h <- sum((e_log)^2)
SCE_h
```

```
## [1] 0.02210668
```

```r
#R2 es "Multiple R-squared"
summary(reg_sim_mujeres_log)
```

```
##
## Call:
## lm(formula = log_tiempo_m ~ log_distancia)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.063794 -0.058885  0.002453  0.049190  0.072736
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.75908    0.08390  -32.88 5.26e-08 ***
## log_distancia  1.11721    0.01114  100.33 6.61e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0607 on 6 degrees of freedom
## Multiple R-squared:  0.9994, Adjusted R-squared:  0.9993
## F-statistic: 1.007e+04 on 1 and 6 DF,  p-value: 6.609e-11
```

OTROS EJERCICIOS.

1. Con los datos de la tensión arterial sistólica y la edad de los 69 pacientes que podemos encontrar en la web de www.fisterra.com https://www.fisterra.com/mbe/investiga/regre_lineal_simple/regre_lineal_simple.asp

Calcular los coeficientes de regresión de la recta mínimo cuadrática.

```r
#El siguiente codigo lo he copiado del foro

url <- "http://www.fisterra.com/mbe/investiga/regre_lineal_simple/regre_lineal_simple.asp"
tbls_xml <- readHTMLTable(url)
typeof(tbls_xml)
```

```
## [1] "list"
```

```r
length(tbls_xml)
```

```
## [1] 26
```

```r
#  De las 26 tablas hay que buscar cual es la que nos interesa (la 11).
datos <- readHTMLTable(url, which=11, header=F, skip.rows = 1:2,
                       colClasses = rep("integer", 6))
```

```
## Warning in asMethod(object): NAs introducidos por coerción

## Warning in asMethod(object): NAs introducidos por coerción

## Warning in asMethod(object): NAs introducidos por coerción
```

```r
head(datos)
```

```
##   V1  V2 V3 V4  V5 V6
## 1  1 114 17 36 156 47
## 2  2 134 18 37 159 47
## 3  3 124 19 38 130 48
## 4  4 128 19 39 157 48
## 5  5 116 20 40 142 50
## 6  6 120 21 41 144 50
```

```r
#Ordenamos el dataframe
colnames(datos) <- c("n","tension","edad","n","tension","edad")
datos <- rbind(datos[,1:3],datos[,4:6])
datos <- datos[-70,]
```

```r
#Coeficientes de la recta de regresión
recta <- lm(tension ~ edad, datos)
coeficientes <- recta$coef
coeficientes
```

```
## (Intercept)        edad
## 103.3526585   0.9835585
```