Practica2_Inferencia

Miguel Ángel Gragera García

9/2/2022

Cargar datos

```
#install.packages("readr")
#install.packages("dplyr")
#install.packages("ggplot2")
#install.packages("GGally")
library (dplyr)
```

```
##
## 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
```

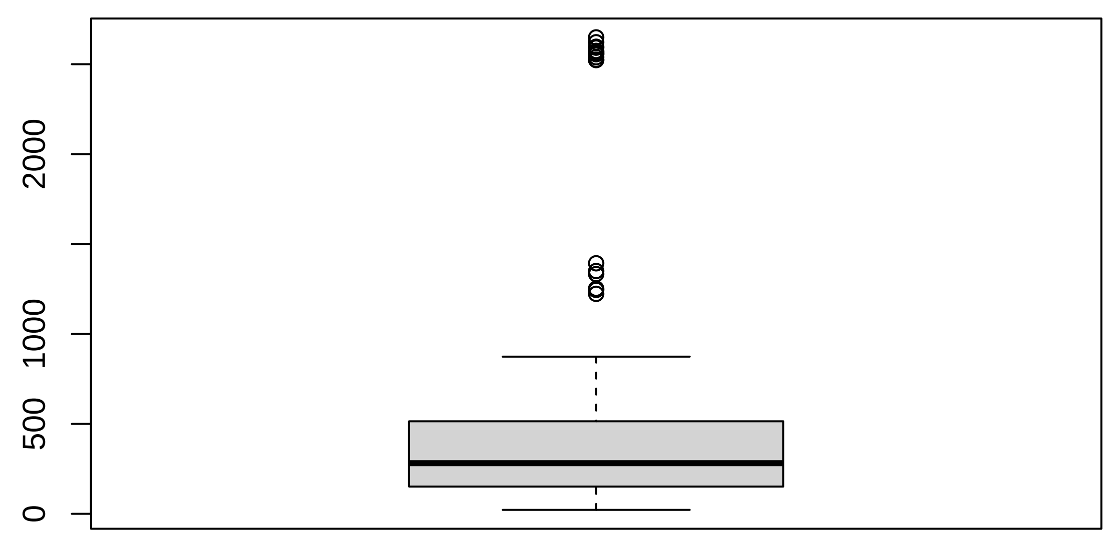
```
ruta_Excel_CalidadAire <- "CalidadAire14_19_zonaProv.csv"
data_CalidadAire <- read.csv(ruta_Excel_CalidadAire, header=TRUE, sep=';',
                             dec = ',')
CalidadAire <- select(data_CalidadAire, Year, CodProv, Population,
          PM10.population.weighted.average..ug.m3.,
          PM2.5.population.weighted.average..ug.m3.,
          NO2.population.weighted.average..ug.m3.,
          O3.SOMO35.population.weighted.average..ug.days.m3.)

names (CalidadAire) = c("Year", "CodProv", "Population", "Factor_PM10",
                        "Factor_PM2.5", "Factor_NO2", "Factor_O3")

CalidadAire_Final <- CalidadAire %>%
  group_by(CodProv, Year) %>%
  summarise_all(sum)

ruta_Excel_datosCP <- "Datos_CP_14_19_prov.csv"
data_datosCP <- read.csv(ruta_Excel_datosCP, header=TRUE, sep=";")
names (data_datosCP) = c("Year", "CodProv", "CIE10", "Provincia",
                         "SumaTotal", "value_f", "value_m")
```

```r
data_final <- merge(data_datosCP, CalidadAire_Final, by = c("Year", "CodProv"))
data_final$Prevalencia = round( (data_final$SumaTotal * 100000) / data_final$Population , 2 )
head(data_final)
```

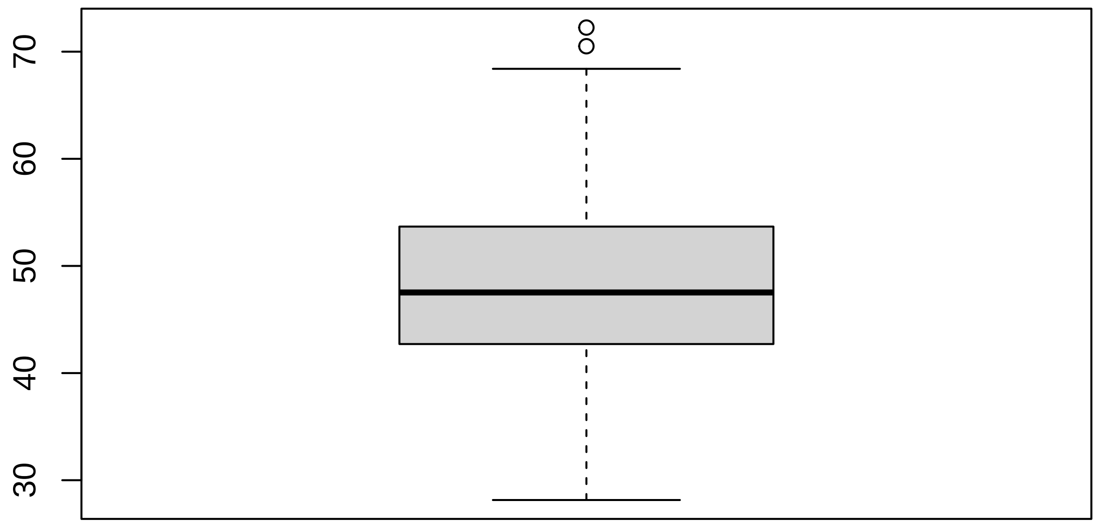
```
##   Year CodProv   CIE10            Provincia SumaTotal value_f value_m
## 1 2014       1 C33-C34         Araba/Ã\201lava       146      25     121
## 2 2014      10 C33-C34              CÃ¡ceres       226      30     196
## 3 2014      11 C33-C34                CÃ¡diz       507      70     437
## 4 2014      12 C33-C34 CastellÃ³n/CastellÃ³       238      30     208
## 5 2014      13 C33-C34          Ciudad Real       263      38     225
## 6 2014      14 C33-C34              CÃ³rdoba       297      34     263
##   Population Factor_PM10 Factor_PM2.5 Factor_NO2 Factor_O3 Prevalencia
## 1     323249        17.0         10.5       17.9    3744.9       45.17
## 2     415041        13.7          6.9        9.0    4113.5       54.45
## 3    1171305        26.3         13.4       16.6    4979.7       43.29
## 4     582572        16.7          9.2       12.9    5876.8       40.85
## 5     531721        18.5          9.7       15.2    6640.5       49.46
## 6     810185        21.4         11.8       16.0    6258.7       36.66
```

BoxPlot

You can also embed plots, for example:



```r
boxplot(data_final$Prevalencia)
```



Correlación

```r
Data_Correlacion_SumaTotal <- select(data_final, Factor_PM10, Factor_PM2.5, Factor_NO2, Factor_O3, Preva
round(cor(Data_Correlacion_SumaTotal),4)
```

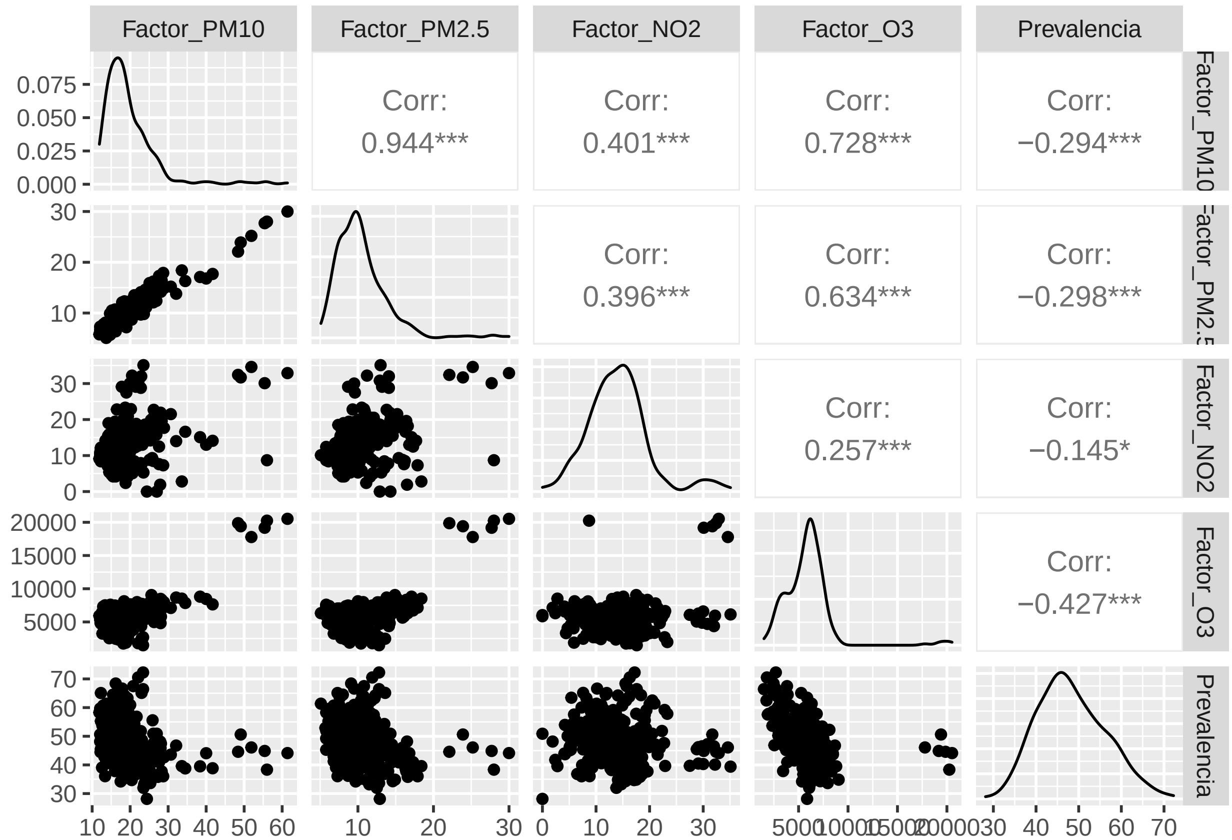
```
##              Factor_PM10 Factor_PM2.5 Factor_NO2 Factor_O3 Prevalencia
## Factor_PM10       1.0000       0.9443     0.4010    0.7284     -0.2943
## Factor_PM2.5      0.9443       1.0000     0.3959    0.6341     -0.2979
## Factor_NO2        0.4010       0.3959     1.0000    0.2572     -0.1448
## Factor_O3         0.7284       0.6341     0.2572    1.0000     -0.4273
## Prevalencia      -0.2943      -0.2979    -0.1448   -0.4273      1.0000
```

```r
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```r
ggpairs(Data_Correlacion_SumaTotal, lower = list(continuos="smooth"), diag = list(continuos="barDiag"),
```

```
## Warning in warn_if_args_exist(list(...)): Extra arguments: "axislabels" are
## being ignored. If these are meant to be aesthetics, submit them using the
## 'mapping' variable within ggpairs with ggplot2::aes or ggplot2::aes_string.
```



Selección variables menor pValues para la regresión

```
modelo <- lm(Prevalencia~Factor_PM10 + Factor_PM2.5 + Factor_NO2 + Factor_O3, data = Data_Correlacion_Su
summary(modelo)
```

```
##
## Call:
## lm(formula = Prevalencia ~ Factor_PM10 + Factor_PM2.5 + Factor_NO2 +
##     Factor_O3, data = Data_Correlacion_SumaTotal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21.4463  -5.2897  -0.8009   4.5632  21.2394
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  57.6134943  1.4430049  39.926  < 2e-16 ***
## Factor_PM10   0.5716805  0.2120611   2.696  0.00742 **
## Factor_PM2.5 -0.9740537  0.3695113  -2.636  0.00883 **
## Factor_NO2   -0.0602769  0.0774903  -0.778  0.43727
## Factor_O3    -0.0016041  0.0002477  -6.475 3.96e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.129 on 295 degrees of freedom
## Multiple R-squared:  0.2041, Adjusted R-squared:  0.1933
## F-statistic: 18.91 on 4 and 295 DF,  p-value: 7.413e-14
```

Regresión lineal múltiple

Se seleccionan las variables factor03, factorPM2.5, FactorPm10

```
library(caTools)
library(caret)
```

```
## Loading required package: lattice
```

```
set.seed(123)
split = sample.split(Data_Correlacion_SumaTotal$Prevalencia, SplitRatio = 0.8)
training_set = subset(Data_Correlacion_SumaTotal, split == TRUE)
testing_set = subset(Data_Correlacion_SumaTotal, split == FALSE)

regression =  lm(Prevalencia~Factor_PM10 + Factor_PM2.5  + Factor_O3, data = training_set)

y_pred = predict(regression, newdata = testing_set)

RMSE(y_pred, testing_set$Prevalencia)
```

```
## [1] 6.839546
```

```
R2(y_pred, testing_set$Prevalencia)
```

```
## [1] 0.18926
```

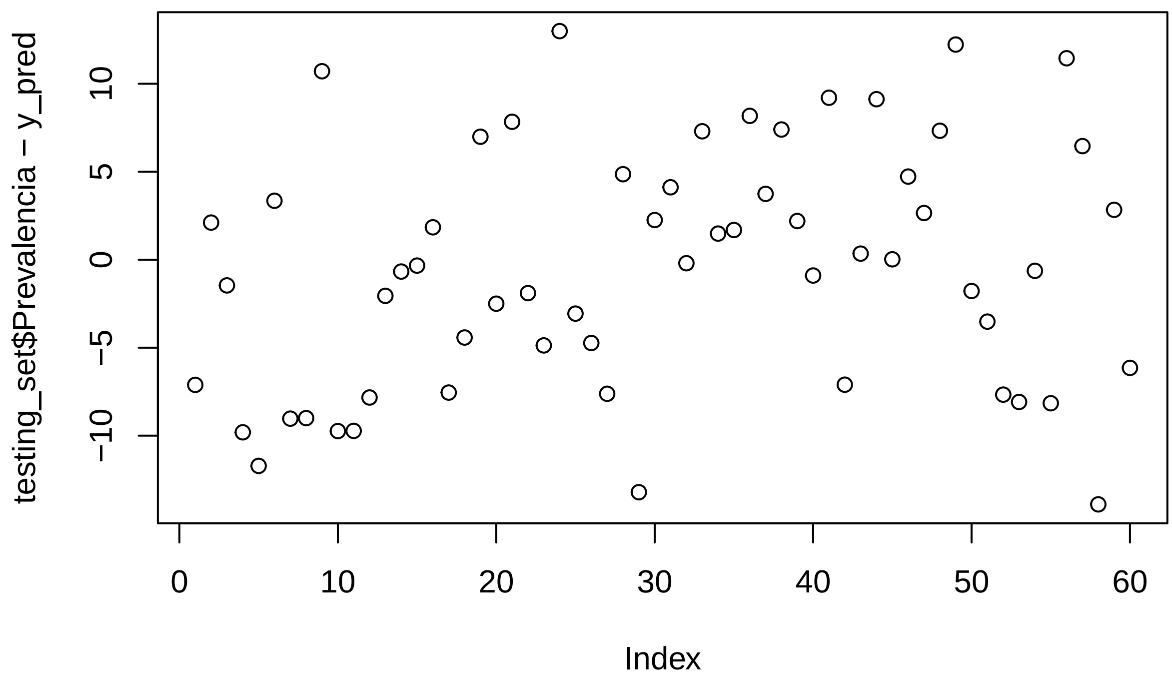
```
y_pred
```

```
##        4        5        8       11       16       20       21       24
## 47.96502 47.35136 47.16020 48.11990 46.42775 52.10031 48.72348 47.67302
##       31       32       34       50       53       59       65       67
## 50.03943 48.51907 48.63853 49.51171 47.45319 45.68329 47.68539 49.59891
##       68       69       87       88       89      104      106      107
## 47.04195 49.12227 46.97153 46.40889 46.44290 47.28960 45.66223 53.63975
##      111      114      118      126      132      137      139      145
## 45.34335 47.91541 49.16012 51.29127 52.05276 46.81609 48.49458 51.08925
##      151      173      179      181      189      190      193      195
## 51.77219 45.99525 56.16272 54.30543 48.10815 47.89123 47.27343 48.51892
##      202      206      219      220      222      230      238      240
## 49.40668 47.58466 49.37406 50.77834 48.04740 47.80661 46.68681 47.80244
##      248      249      260      261      262      264      271      277
## 33.86352 48.27946 46.18887 46.40885 48.45555 50.56096 48.63812 52.35125
##      294      296      297      300
## 50.49440 50.62383 50.07646 49.03823
```

```
testing_set$Prevalencia
```

```
##  [1] 40.85 49.46 45.70 38.31 34.71 55.45 39.69 38.67 60.75 38.78 38.91 41.68
## [13] 45.40 45.01 47.35 51.44 39.49 44.70 53.96 43.91 54.28 45.39 40.79 66.63
## [25] 42.28 43.18 41.54 56.15 38.84 49.07 52.61 50.89 59.07 47.48 57.85 62.48
## [37] 51.85 55.29 49.47 47.62 58.61 40.48 49.72 59.90 48.07 52.53 49.34 55.13
## [49] 46.09 46.50 42.67 38.74 40.37 49.93 40.48 63.80 56.95 36.72 52.91 42.89
```

```
y_compared = data.frame(y_pred,testing_set$Prevalencia )

plot( testing_set$Prevalencia  - y_pred)
```



Refresión SVR

```
library(e1071)
regression = svm(formula = Prevalencia~Factor_PM10 + Factor_PM2.5  + Factor_O3,
                 data = training_set,
                 type = "eps-regression",
                 kernel = "radial")
y_pred = predict(regression, newdata =testing_set )
RMSE(y_pred, testing_set$Prevalencia)
```

```
## [1] 5.474367
```

```r
R2(y_pred, testing_set$Prevalencia)
```

```
## [1] 0.5070083
```

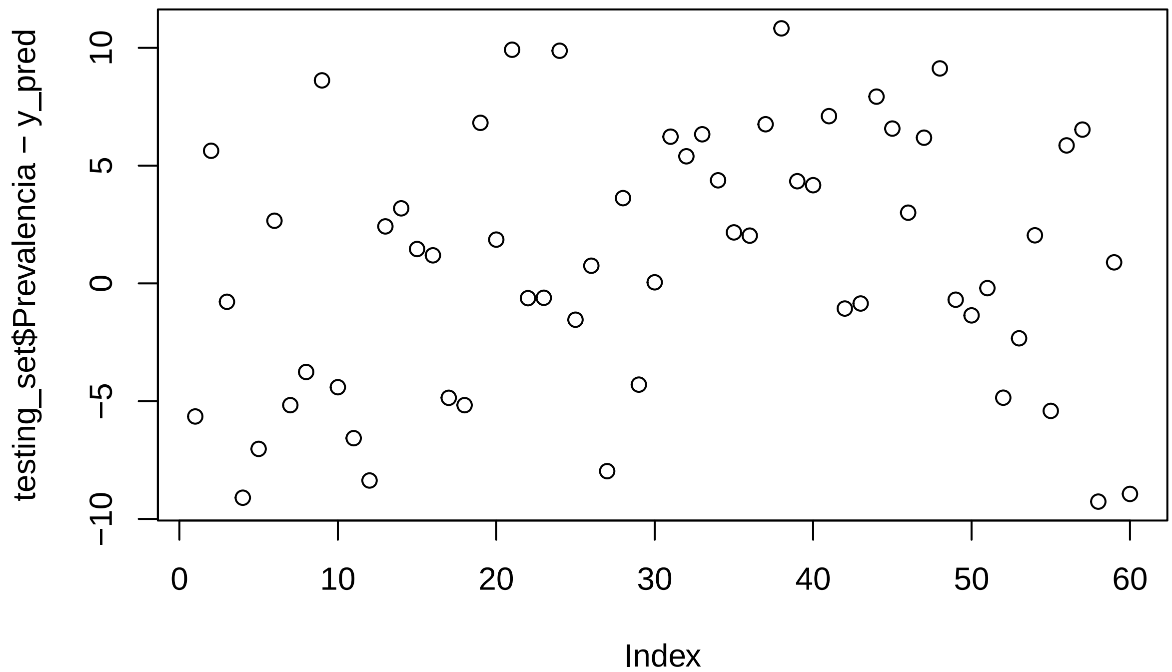
```r
y_pred
```

```
##        4        5        8       11       16       20       21       24 
## 46.49961 43.82946 46.48122 47.41000 41.73681 52.78973 44.85953 42.43125 
##       31       32       34       50       53       59       65       67 
## 52.13097 43.18692 45.48120 50.04827 42.98048 41.82346 45.89178 50.24911 
##       68       69       87       88       89      104      106      107 
## 44.34831 49.86929 47.14289 42.05096 44.36055 46.01726 41.40091 56.75346 
##      111      114      118      126      132      137      139      145 
## 43.82296 42.43035 49.51143 52.53057 43.13840 49.02398 46.38078 45.49459 
##      151      173      179      181      189      190      193      195 
## 52.73979 43.10351 55.68531 60.45243 45.09327 44.46284 45.13177 43.45093 
##      202      206      219      220      222      230      238      240 
## 51.51010 41.54740 50.57426 51.97107 41.49676 49.52806 43.15434 46.00300 
##      248      249      260      261      262      264      271      277 
## 46.78299 47.85749 42.87062 43.59363 42.70161 47.88859 45.89347 57.94199 
##      294      296      297      300 
## 50.42030 45.98476 52.01665 51.82837
```

```r
testing_set$Prevalencia
```

```
##  [1] 40.85 49.46 45.70 38.31 34.71 55.45 39.69 38.67 60.75 38.78 38.91 41.68
## [13] 45.40 45.01 47.35 51.44 39.49 44.70 53.96 43.91 54.28 45.39 40.79 66.63
## [25] 42.28 43.18 41.54 56.15 38.84 49.07 52.61 50.89 59.07 47.48 57.85 62.48
## [37] 51.85 55.29 49.47 47.62 58.61 40.48 49.72 59.90 48.07 52.53 49.34 55.13
## [49] 46.09 46.50 42.67 38.74 40.37 49.93 40.48 63.80 56.95 36.72 52.91 42.89
```

```r
y_compared = data.frame(y_pred,testing_set$Prevalencia )

plot( testing_set$Prevalencia  - y_pred)
```



Regresión RF

```
library(randomForest)
```

```
## randomForest 4.7-1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':
##
##     margin

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
set.seed(1234)
regression = randomForest(x = training_set[, 1:4],
                          y = training_set$Prevalencia,
```

```r
                                ntree = 100)
y_pred = predict(regression, newdata =testing_set )
RMSE(y_pred, testing_set$Prevalencia)
```

```
## [1] 5.72547
```

```r
R2(y_pred, testing_set$Prevalencia)
```

```
## [1] 0.4430676
```

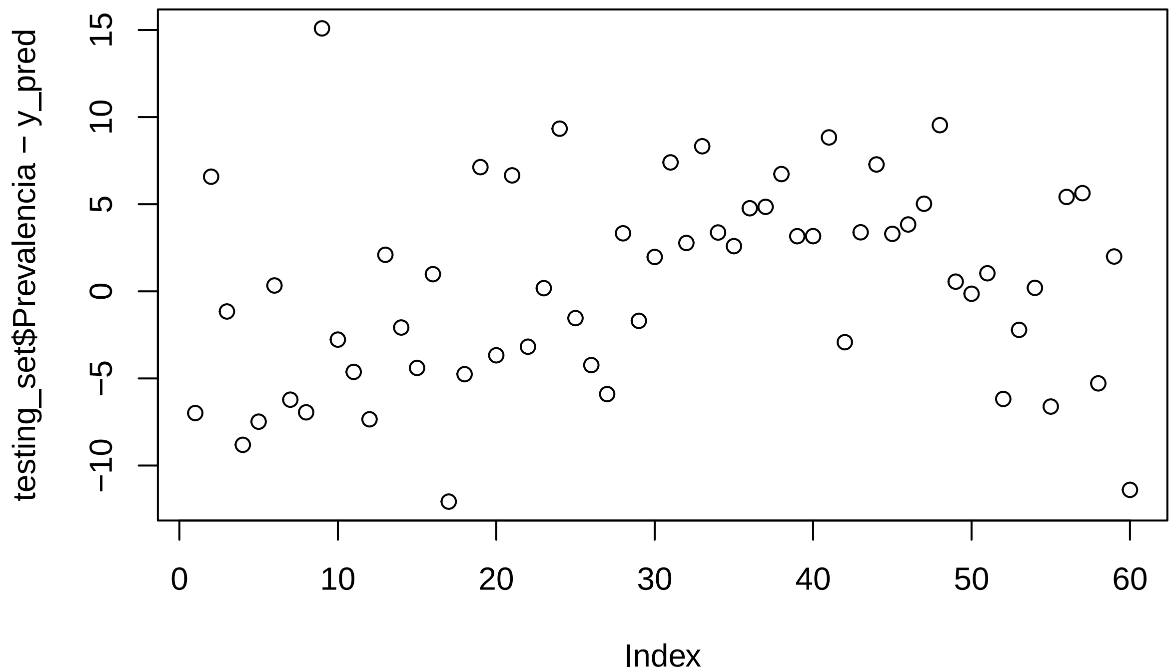
```r
y_pred
```

```
##        4        5        8       11       16       20       21       24
## 47.83692 42.87965 46.85679 47.12104 42.19231 55.11470 45.90915 45.61557
##       31       32       34       50       53       59       65       67
## 45.65535 41.55078 43.53490 49.02617 43.30266 47.08764 51.74716 50.45528
##       68       69       87       88       89      104      106      107
## 51.56108 49.45451 46.83133 47.58241 47.62439 48.56820 40.60756 57.29395
##      111      114      118      126      132      137      139      145
## 43.81743 47.41464 47.43941 52.81947 40.53229 47.09726 45.20984 48.11468
##      151      173      179      181      189      190      193      195
## 50.74210 44.10323 55.25721 57.71128 47.00006 48.55874 46.30458 44.45282
##      202      206      219      220      222      230      238      240
## 49.77555 43.39928 46.33174 52.61790 44.77107 48.69246 44.31264 45.59141
##      248      249      260      261      262      264      271      277
## 45.53448 46.63955 41.63241 44.91799 42.58298 49.73060 47.09435 58.38111
##      294      296      297      300
## 51.31381 42.00340 50.91037 54.28621
```

```r
testing_set$Prevalencia
```

```
##  [1] 40.85 49.46 45.70 38.31 34.71 55.45 39.69 38.67 60.75 38.78 38.91 41.68
## [13] 45.40 45.01 47.35 51.44 39.49 44.70 53.96 43.91 54.28 45.39 40.79 66.63
## [25] 42.28 43.18 41.54 56.15 38.84 49.07 52.61 50.89 59.07 47.48 57.85 62.48
## [37] 51.85 55.29 49.47 47.62 58.61 40.48 49.72 59.90 48.07 52.53 49.34 55.13
## [49] 46.09 46.50 42.67 38.74 40.37 49.93 40.48 63.80 56.95 36.72 52.91 42.89
```

```r
y_compared = data.frame(y_pred,testing_set$Prevalencia )

plot( testing_set$Prevalencia  - y_pred)
```



Intervalo de confianza bilateral para la diferencia de medias

```
n <- length(data_final$Prevalencia)    # El tamaño válido de la muestra
media <- mean(data_final$Prevalencia) # la media
desv <- sd(data_final$Prevalencia)  # La desviación estándar. Datos históricos
nivelconfianza = 0.80

error.est <- desv/sqrt(n) # Calculamos el error estándar
margen.error <- 1.644854 * error.est # nivel de confianza de 90%


lim.inf <- media - margen.error # Límite inferior del intervalo
lim.inf
```

```
## [1] 47.62279
```

```
lim.sup <- media + margen.error # Límite superior del intervalo
lim.sup
```

```
## [1] 49.13028
```

```r
#install.packages("BSDA")
library(BSDA)
```

```
##
## Attaching package: 'BSDA'

## The following object is masked from 'package:datasets':
##
##     Orange
```

```r
zsum.test(mean.x=media,sigma.x=desv, n.x=n,conf.level=nivelconfianza)
```

```
##
##  One-sample z-Test
##
## data:  Summarized x
## z = 105.57, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 80 percent confidence interval:
##  47.78927 48.96380
## sample estimates:
## mean of x
##  48.37653
```

```r
par(mfrow=c(1, 2))
require(car)  # Debe instalar antes el paquete car
```

```
## Loading required package: car

## Loading required package: carData

##
## Attaching package: 'carData'

## The following objects are masked from 'package:BSDA':
##
##     Vocab, Wool

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
qqPlot(data_final$Prevalencia, pch=19,
       main='QQplot para la prevalencia',
       xlab='Cuantiles teóricos',
       ylab='Cuantiles muestrales')
```

```
## [1] 227 127
```

```r
hist(data_final$Prevalencia , freq=TRUE,
     main='Histograma para la prevalencia',
     xlab='Prevalencia',
     ylab='Frecuencia')
```