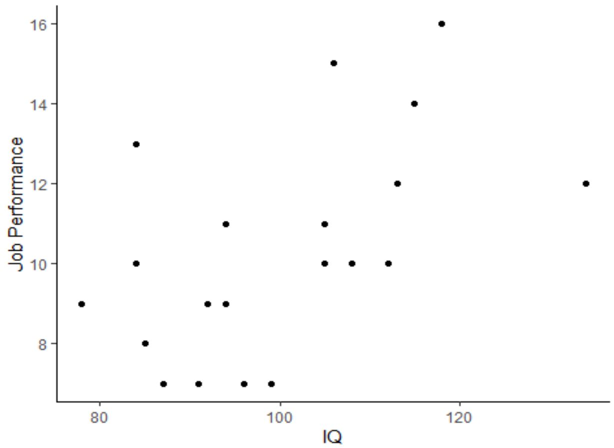
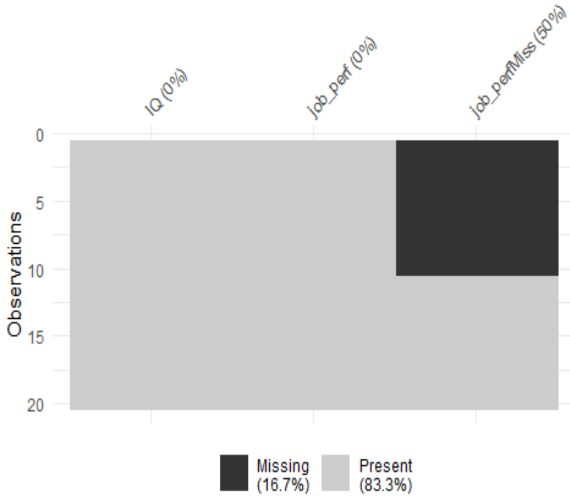
T2UD1

```
#TEMA 2:

#DATOS FALTANTES

library(dplyr)

library(naniar)
library(ggplot2)
library(simputation)

ejemplo<- read.table("ejemplo.csv", header=T, sep=";")

ejemplo

##     IQ job_perf job_perfMiss
## 1   78        9           NA
## 2   84       13           NA
## 3   84       10           NA
## 4   85        8           NA
## 5   87        7           NA
## 6   91        7           NA
## 7   92        9           NA
## 8   94        9           NA
## 9   94       11           NA
## 10  96        7           NA
## 11  99        7            7
## 12 105       10           10


ggplot(ejemplo, aes(IQ, job_perf) ) +
  geom_point() +
  xlab("IQ") +
  ylab("Job Performance") +
  theme_classic()
```

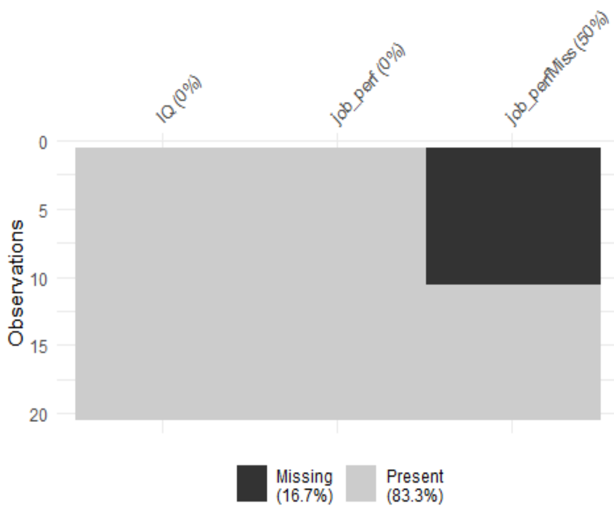



```
#Resumen basico de los datos faltantes

n_miss(ejemplo)

## [1] 10

n_miss_row(ejemplo)#En cada registro comprueba si hay algÃºn valor perdid
o, si el registro tiene un valor perdido en una variable pondrÃ¡ 1, si el
resgistro tiene valor

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

#perdido en dos variables pondrÃ¡ un 2, asÃ- sucesivamente.

prop_miss(ejemplo)

## [1] 0.1666667

n_complete(ejemplo)

## [1] 50

n_complete_row(ejemplo)

##  [1] 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3

prop_miss_case(ejemplo)#Calcula la proporciÃ³n de casos (filas) que conti
enen valores faltantes.

## [1] 0.5

prop_complete_case(ejemplo)#Calcula la proporciÃ³n de casos (filas) que c
ontienen valores completos

## [1] 0.5
```

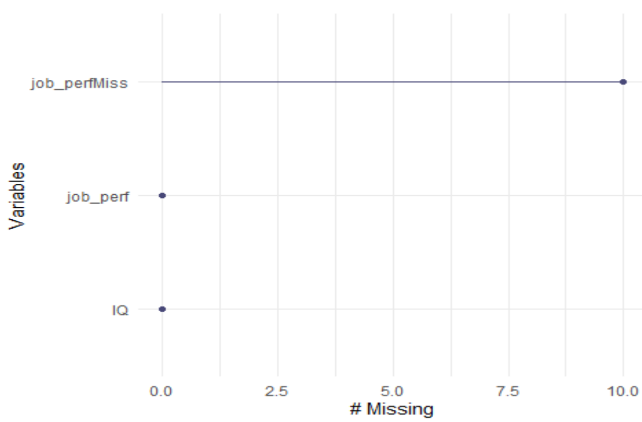
```r
# Hacemos un estudio de los datos perdidos. Utilizamos el paquete naniar(
)

# Numero de casos perdidos en cada variable y su porcentaje.

miss_var_summary(ejemplo)

## # A tibble: 3 × 3
##   variable     n_miss pct_miss
##   <chr>         <int>    <dbl>
## 1 job_perfMiss     10       50
## 2 IQ                0        0
## 3 job_perf          0        0

# Estudio registro a registro sobre el numero de variables que ese regist
ro no tiene dato, y su porcentaje

miss_case_summary(ejemplo)

## # A tibble: 20 × 3
##     case n_miss pct_miss
##    <int>  <int>    <dbl>
##  1     1      1     33.3
##  2     2      1     33.3
##  3     3      1     33.3
##  4     4      1     33.3
##  5     5      1     33.3
##  6     6      1     33.3
##  7     7      1     33.3


# Tabla resumen
miss_var_table(ejemplo) #en 2 variables no hay ningÃºn caso perdido, en 1
variable hay 10 casos perdidos.

## # A tibble: 2 × 3
##   n_miss_in_var n_vars pct_vars
##           <int>  <int>    <dbl>
## 1             0      2     66.7
## 2            10      1     33.3

miss_case_table(ejemplo)# 10 casos sin ningun valor perdido (50%) y 10 ca
sos con algun valor perdido

## # A tibble: 2 × 3
##   n_miss_in_case n_cases pct_cases
##            <int>   <int>     <dbl>
## 1              0      10        50
## 2              1      10        50

vis_miss(ejemplo)#10/60*100=16.7% de missing, 50/60*100=83.3%
```



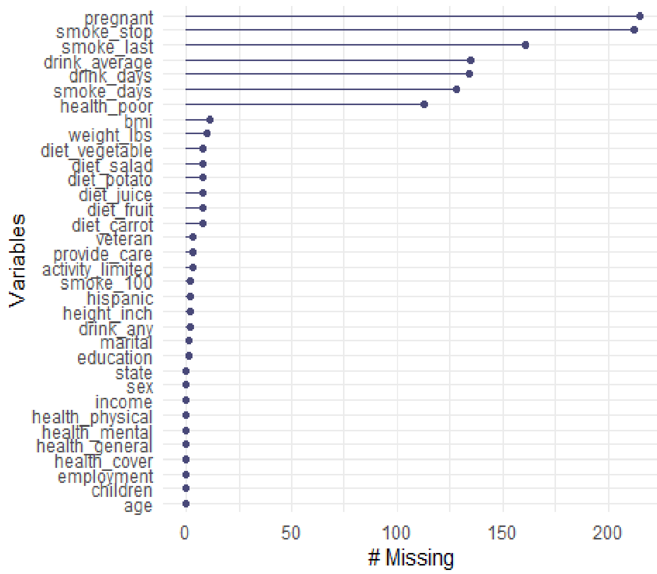

```
vis_miss(ejemplo, cluster = T)#agrupa los valores perdidos
```



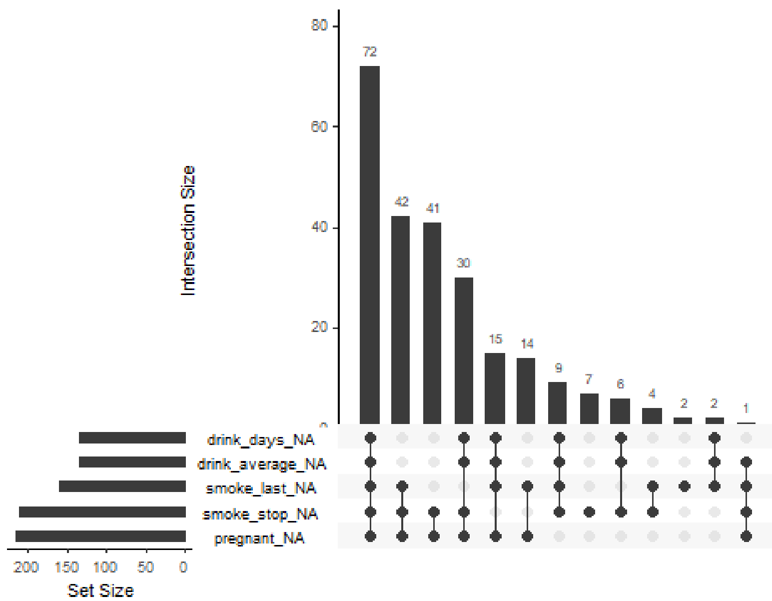

```
gg_miss_var(ejemplo) #anÃ¡logo a miss_var_summary
```



```
gg_miss_var(riskfactors)#con la base de datos riskfactors
```

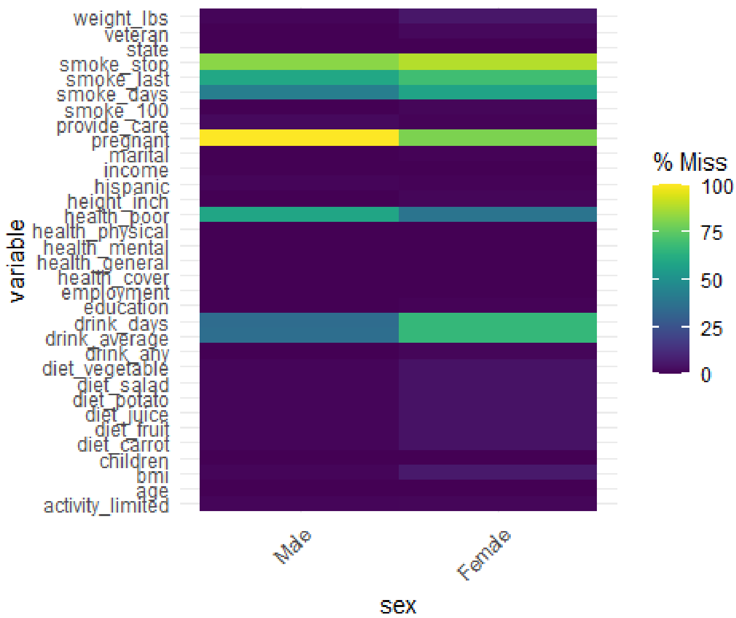



```
gg_miss_upset(riskfactors)#VisualizaciÃ³n de patrones, # muestra el nÃºme
ro de combinaciones de valores faltantes que coexisten, el 72 de la barra
significa que hay dos registros que tienen
```

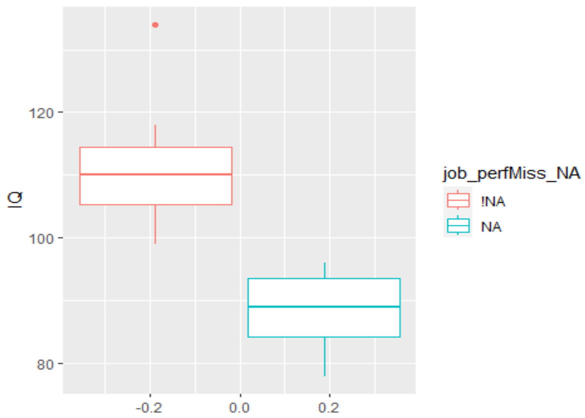



```
#perdidos en esas 5 variables, 7 casos que tienen valores perdidos en smo
ke_stop_NA y no en el resto.
```

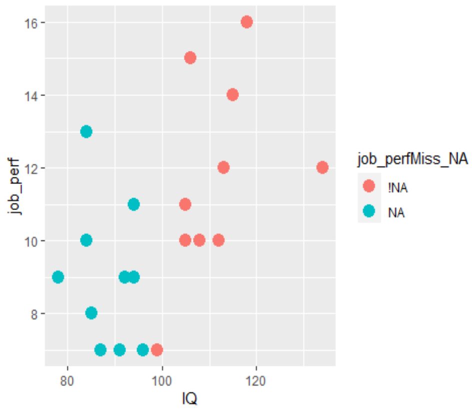
```r
gg_miss_fct(x = riskfactors, fct = sex)
```




```r
#comprobación de si los datos faltantes son aleatorios o no

ejemplo %>%
  bind_shadow() %>%
  group_by(job_perfMiss_NA) %>%
  summarize(mean = mean(IQ))

## # A tibble: 2 × 2
##   job_perfMiss_NA  mean
##   <fct>           <dbl>
## 1 !NA             112.
## 2 NA               88.5

ejemplo %>%
  bind_shadow() %>%
  ggplot(aes(y = IQ, color = job_perfMiss_NA))+
  geom_boxplot()
```

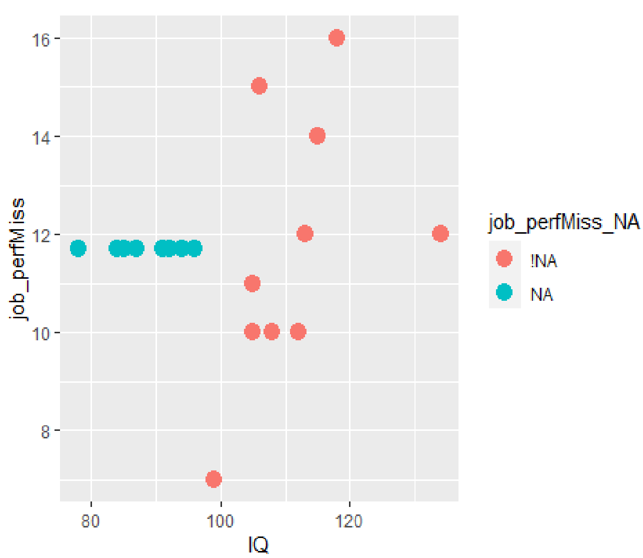


```
ejemplo %>%
  bind_shadow() %>%
  ggplot(aes(x=IQ , y = job_perf, color = job_perfMiss_NA))+
  geom_point(size = 4)
```

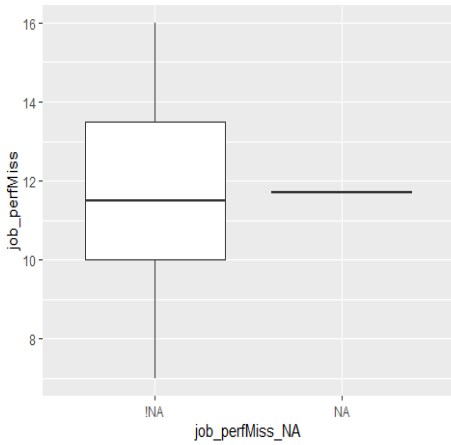



```
#ComprobaciÃ³n si los datos son Completamente Aleatorios (MCAR). Test de
Little
mcar_test(ejemplo)

## # A tibble: 1 × 4
##   statistic    df  p.value missing.patterns
##       <dbl> <dbl>    <dbl>            <int>
## 1      14.9     2 0.000592                2

#Diferencias significativas, los datos ausentes no pueden ser clasificado
s como MCAR



#Soluciones a los datos faltantes

#1.- Utilizar solo aquellas observaciones con datos completos (omitir los
registros que poseen valores faltantes)

ejemplo_cc <- ejemplo %>%
  na.omit()

ejemplo_cc

##      IQ job_perf job_perfMiss
## 11   99        7            7
## 12 105       10           10
## 13 105       11           11
## 14 106       15           15
## 15 108       10           10
…
```

```
#2.- Metodos de imputacion

# Imputacion por la media

library(ggplot2)
library(naniar)

ejemplo_impute_mean <- ejemplo %>%
  bind_shadow(only_miss = TRUE) %>%
  impute_mean_all()

head(ejemplo_impute_mean)

## # A tibble: 6 × 4
##      IQ job_perf job_perfMiss job_perfMiss_NA
##   <dbl>    <dbl>        <dbl> <fct>
## 1    78        9         11.7 NA
## 2    84       13         11.7 NA
## 3    84       10         11.7 NA
## 4    85        8         11.7 NA
## 5    87        7         11.7 NA
## 6    91        7         11.7 NA

ggplot(ejemplo_impute_mean,
       aes(x = IQ,y = job_perfMiss,color = job_perfMiss_NA)) +
  geom_point(size = 4)
```

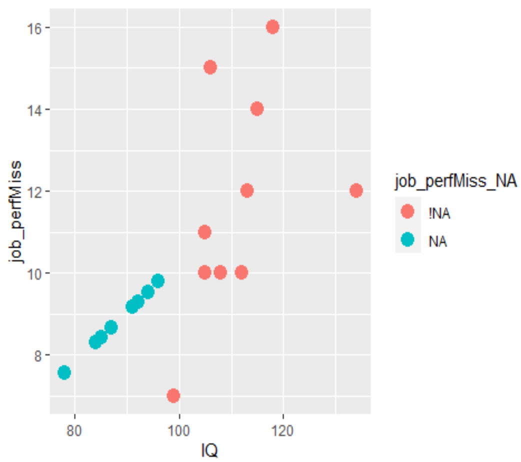



```
ggplot(ejemplo_impute_mean,
       aes(x = job_perfMiss_NA,y = job_perfMiss)) +
  geom_boxplot()
```



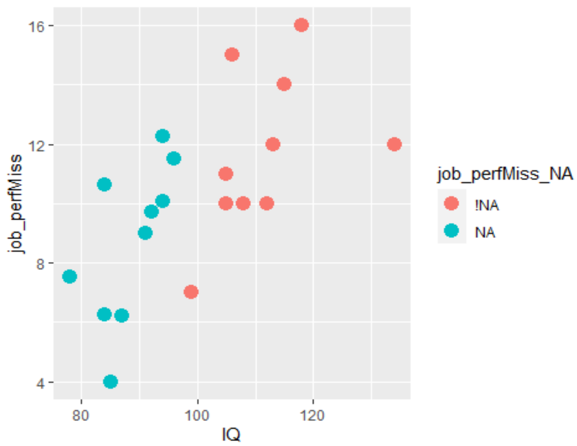

```
#Imputacion regresion (no estocastica)

library(simputation)
library(naniar)
library(ggplot2)
library(dplyr)

str(ejemplo)

## 'data.frame':    20 obs. of  3 variables:
##  $ IQ          : int  78 84 84 85 87 91 92 94 94 96 ...
##  $ job_perf    : int  9 13 10 8 7 7 9 9 11 7 ...
##  $ job_perfMiss: int  NA NA NA NA NA NA NA NA NA NA ...

ejemplo$IQ <- as.numeric(ejemplo$IQ)
ejemplo$job_perf <- as.numeric(ejemplo$job_perf)
ejemplo$job_perfMiss <- as.numeric(ejemplo$job_perfMiss)

str(ejemplo)

## 'data.frame':    20 obs. of  3 variables:
##  $ IQ          : num  78 84 84 85 87 91 92 94 94 96 ...
##  $ job_perf    : num  9 13 10 8 7 7 9 9 11 7 ...
##  $ job_perfMiss: num  NA NA NA NA NA NA NA NA NA NA ...

lm(data = ejemplo,job_perfMiss ~ IQ)

##
## Call:
## lm(formula = job_perfMiss ~ IQ, data = ejemplo)
##
## Coefficients:
## (Intercept)           IQ
##     -2.0646       0.1234
```

```
ejemplo_impute_lm <-
  ejemplo %>%
  bind_shadow() %>%
  impute_lm(job_perfMiss ~ IQ)

ejemplo_impute_lm

## # A tibble: 20 × 6
##        IQ job_perf job_perfMiss IQ_NA job_perf_NA job_perfMiss_NA
##   * <dbl>    <dbl>        <dbl> <fct> <fct>       <fct>
##   1    78        9         7.56 !NA   !NA         NA
##   2    84       13         8.31 !NA   !NA         NA
##   3    84       10         8.31 !NA   !NA         NA

…

ggplot(ejemplo_impute_lm, aes(x = IQ, y = job_perfMiss, color = job_perfM
iss_NA)) +
  geom_point(size = 4)
```




```
#Imputacion regresion lineal estocastica

ejemplo_impute_slm <-
  ejemplo %>%
  bind_shadow() %>%
  impute_lm(job_perfMiss ~ IQ, add_residual = "normal")

head(ejemplo_impute_slm)

## # A tibble: 6 × 6
##      IQ job_perf job_perfMiss IQ_NA job_perf_NA job_perfMiss_NA
##   <dbl>    <dbl>        <dbl> <fct> <fct>       <fct>
## 1    78        9         7.55 !NA   !NA         NA
## 2    84       13         6.27 !NA   !NA         NA
## 3    84       10        10.6  !NA   !NA         NA
## 4    85        8         4.00 !NA   !NA         NA
## 5    87        7         6.21 !NA   !NA         NA
## 6    91        7         8.98 !NA   !NA         NA
```

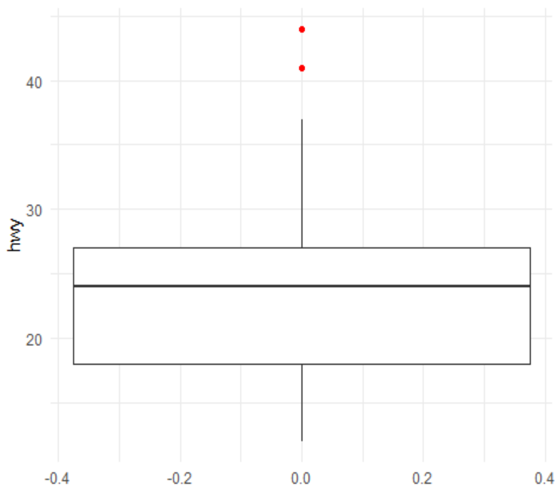
```r
ggplot(ejemplo_impute_slm,
       aes(x = IQ,
           y = job_perfMiss,
           color = job_perfMiss_NA)) +
  geom_point(size = 4)
```




```r
#IMPUTACION HOTDECK

library(validate)#contiene la base de datos retailers

data(retailers)
str(retailers)

## 'data.frame':    60 obs. of  10 variables:
##  $ size       : Factor w/ 4 levels "sc0","sc1","sc2",..: 1 4 4 4 4 1 4
2 4 3 ...
##  $ incl.prob  : num  0.02 0.14 0.14 0.14 0.14 0.02 0.14 0.02 0.14 0.05
...
##  $ staff      : int  75 9 NA NA NA 1 5 3 6 5 ...
##  $ turnover   : int  NA 1607 6886 3861 NA 25 NA 404 2596 NA ...
##  $ other.rev  : int  NA NA -33 13 37 NA NA 13 NA NA ...
##  $ total.rev  : int  1130 1607 6919 3874 5602 25 1335 417 2596 NA ...
##  $ staff.costs: int  NA 131 324 290 314 NA 135 NA 147 NA ...
##  $ total.costs: int  18915 1544 6493 3600 5530 22 136 342 2486 NA ...
##  $ profit     : int  20045 63 426 274 72 3 1 75 110 NA ...
##  $ vat        : int  NA NA NA NA NA NA 1346 NA NA NA ...

head(retailers, 10)

##    size incl.prob staff turnover other.rev total.rev staff.costs total.costs
## 1   sc0      0.02    75       NA        NA      1130          NA       18915
## 2   sc3      0.14     9     1607        NA      1607         131        1544
## 3   sc3      0.14    NA     6886       -33      6919         324        6493
## 4   sc3      0.14    NA     3861        13      3874         290        3600
## 5   sc3      0.14    NA       NA        37      5602         314        5530
## 6   sc0      0.02     1       25        NA        25          NA          22
## 7   sc3      0.14     5       NA        NA      1335         135         136
## 8   sc1      0.02     3      404        13       417          NA         342
## 9   sc3      0.14     6     2596        NA      2596         147        2486
….
```

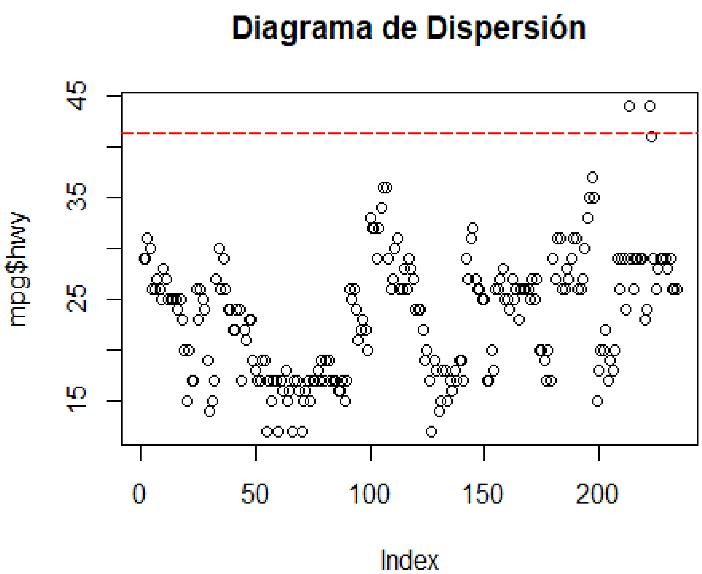
```r
#imputacion aleatoria HOTDECK
set.seed(1)
ret1_hd<- impute_rhd(retailers, turnover + other.rev + total.rev ~ size )

head(ret1_hd, 10)

##    size incl.prob staff turnover other.rev total.rev staff.costs total.costs
## 1   sc0      0.02    75      359         9      1130          NA       18915
## 2   sc3      0.14     9     1607     98350      1607         131        1544
## 3   sc3      0.14    NA     6886       -33      6919         324        6493
## 4   sc3      0.14    NA     3861        13      3874         290        3600
## 5   sc3      0.14    NA     2649        37      5602         314        5530
## 6   sc0      0.02     1       25       622        25          NA          22
## 7   sc3      0.14     5     4445        20      1335         135         136
## 8   sc1      0.02     3      404        13       417          NA         342
## 9   sc3      0.14     6     2596        32      2596         147        2486
## 10  sc2      0.05     5     1175         4       206          NA          NA


#imputacion secuencial

ret1_shd <- impute_shd(retailers, turnover ~ size + profit)
head(ret1_shd, 10)

##    size incl.prob staff turnover other.rev total.rev staff.costs total.costs
## 1   sc0      0.02    75      839        NA      1130          NA       18915
## 2   sc3      0.14     9     1607        NA      1607         131        1544
## 3   sc3      0.14    NA     6886       -33      6919         324        6493
## 4   sc3      0.14    NA     3861        13      3874         290        3600
## 5   sc3      0.14    NA     1607        37      5602         314        5530
## 6   sc0      0.02     1       25        NA        25          NA          22
## 7   sc3      0.14     5     2333        NA      1335         135         136
## 8   sc1      0.02     3      404        13       417          NA         342
## 9   sc3      0.14     6     2596        NA      2596         147        2486
## 10  sc2      0.05     5      690        NA        NA          NA          NA


head(retailers)

##   size incl.prob staff turnover other.rev total.rev staff.costs total.costs
## 1  sc0      0.02    75       NA        NA      1130          NA       18915
## 2  sc3      0.14     9     1607        NA      1607         131        1544
## 3  sc3      0.14    NA     6886       -33      6919         324        6493
## 4  sc3      0.14    NA     3861        13      3874         290        3600
## 5  sc3      0.14    NA       NA        37      5602         314        5530
## 6  sc0      0.02     1       25        NA        25          NA          22
##   profit vat
## 1  20045  NA
## 2     63  NA
## 3    426  NA
## 4    274  NA
## 5     72  NA
## 6      3  NA
```

```r
####################################################################
################################OUTLIERS############################

library(outliers)
library(ggplot2)
library(dplyr)


### todo con la variable mpg$hwy

ggplot(mpg, aes(y = hwy)) +
  geom_boxplot(outlier.colour = "red") +
  theme_minimal()
```

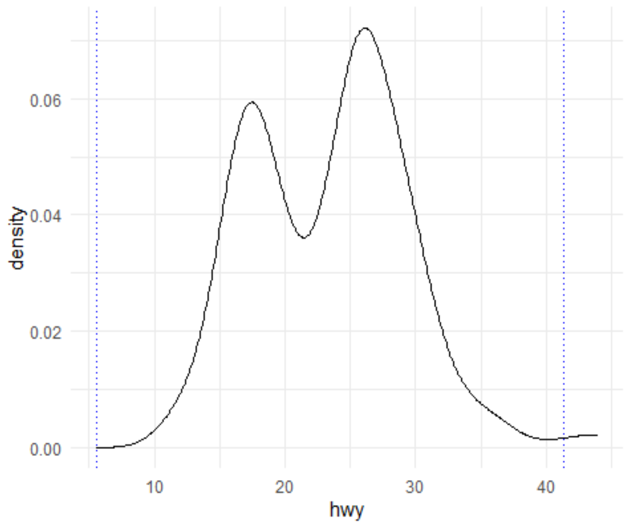



```r
boxplot.stats(mpg$hwy)

## $stats
## [1] 12 18 24 27 37
##
## $n
## [1] 234
##
## $conf
## [1] 23.07041 24.92959
##
## $out
## [1] 44 44 41

boxplot.stats(mpg$hwy)$out

## [1] 44 44 41

which(mpg$hwy %in% boxplot.stats(mpg$hwy)$out)

## [1] 213 222 223
```

```
###realizamos un gráfico de dispersión y marcamos los outliers potenciale
s
#a 3 desviaciones típicas de la media

# Para este ejemplo utilizamos 3 desviaciones típicas

outliers_max<-mean(mpg$hwy)+3*sd(mpg$hwy);outliers_max

## [1] 41.3041

outliers_min<-mean(mpg$hwy)-3*sd(mpg$hwy); outliers_min

## [1] 5.576241

plot(mpg$hwy, main="Diagrama de Dispersión")
abline(h=c(outliers_max,outliers_min), col="red",lty=5)
```

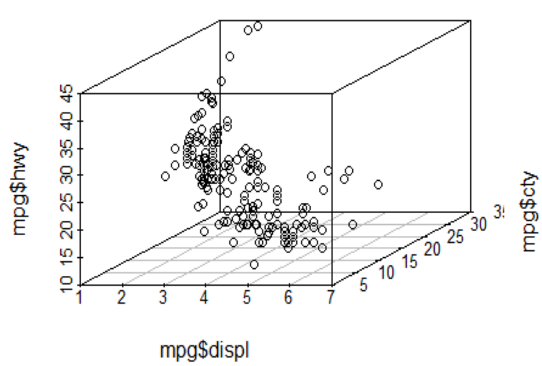



```
####

ggplot(mpg, aes(x = hwy)) +
  geom_density()+
  geom_vline(xintercept = c(outliers_min,outliers_max),
             linetype="dotted",color="blue") + #agregamos lineas vertical
es, para identificar outliers
  theme_minimal()
```

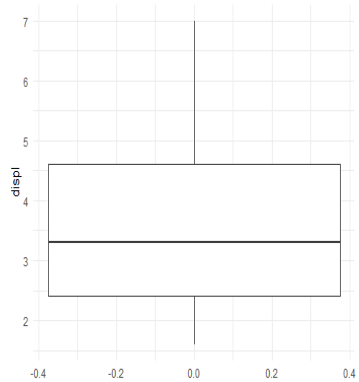



```
#TEST PARA DETECTAR CASOS ATIPICOS

#Test de grubbs
library(outliers)

grubbs.test(mpg$hwy)

##
##  Grubbs test for one outlier
##
## data:  mpg$hwy
## G = 3.45274, U = 0.94862, p-value = 0.05555
## alternative hypothesis: highest value 44 is an outlier

#Test de dixon, para pocos datos
#Seleccionamos los primeros 20 registros de la variable wage

submpg <- mpg %>%
  slice(1:20)

dixon.test(submpg$hwy)

##
##  Dixon test for outliers
##
## data:  submpg$hwy
## Q = 0.57143, p-value = 0.006508
## alternative hypothesis: lowest value 15 is an outlier

#Prueba de Rosner

library(EnvStats)
```

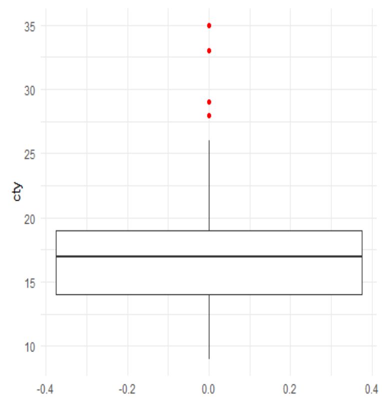
```
test <- rosnerTest(mpg$hwy,alpha = 0.05,k = 3)

test$all.stats

##   i   Mean.i     SD.i Value Obs.Num    R.i+1 lambda.i+1 Outlier
## 1 0 23.44017 5.954643    44     213 3.452739   3.652091   FALSE
## 2 1 23.35193 5.812124    44     222 3.552586   3.650836   FALSE
## 3 2 23.26293 5.663340    41     223 3.131909   3.649575   FALSE

#quitar las observaciones outliers

mpg %>%
  slice(-213)

## # A tibble: 233 × 11
##    manufacturer model      displ  year   cyl trans drv     cty   hwy fl    class
##    <chr>        <chr>      <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>
##  1 audi         a4           1.8  1999     4 auto… f        18    29 p     comp…
##  2 audi         a4           1.8  1999     4 manu… f        21    29 p     comp…
##  3 audi         a4           2    2008     4 manu… f        20    31 p     comp…
##  4 audi         a4           2    2008     4 auto… f        21    30 p     comp…
##  5 audi         a4           2.8  1999     6 auto… f        16    26 p     comp…
##  6 audi         a4           2.8  1999     6 manu… f        18    26 p     comp…
##  7 audi         a4           3.1  2008     6 auto… f        18    27 p     comp…
##  8 audi         a4 quattro   1.8  1999     4 manu… 4        18    26 p     comp…
##  9 audi         a4 quattro   1.8  1999     4 auto… 4        16    25 p     comp…
## 10 audi         a4 quattro   2    2008     4 manu… 4        20    28 p     comp…
## # … with 223 more rows

###Análisis conjunto, multivariantes

library(scatterplot3d) # Observamos los 3 exámenes

scatterplot3d(mpg$displ, mpg$cty, mpg$hwy)
```



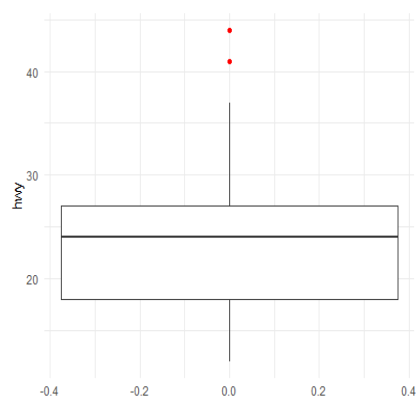
```
ggplot(mpg, aes(y = displ)) +
  geom_boxplot(outlier.colour = "red") +
  theme_minimal()
```




```
ggplot(mpg, aes(y = cty)) +
  geom_boxplot(outlier.colour = "red") +
  theme_minimal()
```



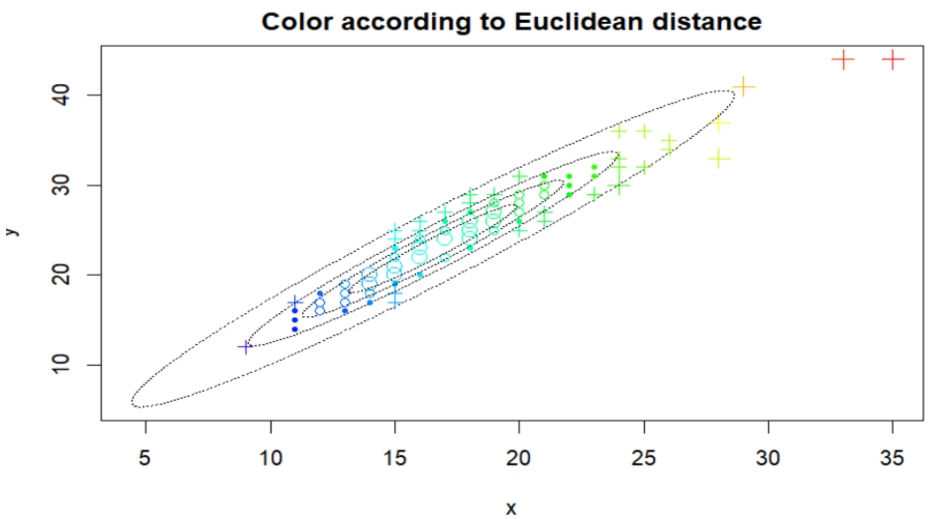

```
ggplot(mpg, aes(y = hwy)) +
  geom_boxplot(outlier.colour = "red") +
  theme_minimal()
```



```r
#Distancias de mahalanobis

data <- mpg %>%
  select(displ,cty,hwy)

vector_medias = colMeans(data)
matriz_var_cov = cov(data)



# Creamos una variable con la distancia
data$maha = sqrt(mahalanobis(data,vector_medias,matriz_var_cov))


# Los 6 registros mas distantes según la distancia de Mahalanobis

top_maha <- data %>%
  top_n(6, maha) %>%
  print()

##   displ cty hwy     maha
## 1   6.2  16  26 3.905398
## 2   6.2  15  25 3.785788
## 3   7.0  15  24 4.370385
## 4   1.6  28  33 4.149844
## 5   1.9  33  44 4.872089
## 6   1.9  35  44 5.968596

#####Análisis con dos variables


library(mvoutlier)

## Loading required package: sgeostat

Z <- cbind(mpg$cty, mpg$hwy)

color.plot(Z)

$outliers

  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

 [18] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

 [35] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

 [52] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

 [69] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

 [86] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE

[103] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
[120] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[137] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[154] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[171] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[188] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[222]  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

$md
  [1] 2.5040297 1.0166907 2.1186177 1.1053570 2.3716575 0.6899169 1.2747255 0.6899169 1.7472385 0.7848141
 [11] 0.6557755 2.6470233 0.8681960 0.8681960 2.6470233 2.0284363 0.8681960 0.5063281 0.6822739 1.2451580
 [21] 0.6822739 1.0033782 1.0504160 2.3716575 1.4159195 2.3716575 2.6470233 2.0284363 0.6187648 1.4394136
 [31] 1.2451580 1.6015536 0.6557755 1.2973904 0.6899169 2.5040297 1.4909615 0.7248898 0.2588740 0.1783561
 [41] 0.1783561 0.2588740 0.2588740 1.6896130 0.8231378 0.3643304 0.5063281 0.5063281 1.1791769 1.0405264
 [51] 1.0033782 1.6015536 0.6187648 0.6187648 1.7036371 1.6896130 1.2451580 1.0033782 1.0033782 1.7036371
 [61] 1.0033782 1.3458747 0.7919622 1.2451580 1.0782743 1.7036371 1.0033782 1.0033782 1.0782743 1.7036371
 [71] 1.2451580 1.3458747 1.0033782 1.2451580 1.6896130 1.6896130 1.3514901 1.6015536 1.1791769 1.6015536
 [81] 1.0149354 1.0149354 1.0033782 1.6015536 1.6015536 1.4726794 1.4726794 1.0033782 1.2451580 1.0033782
 [91] 0.6899169 0.3309058 1.4909615 1.1240012 0.3643304 0.8231378 1.4159195 0.8231378 0.6822739 4.6157223
[101] 1.9177726 2.6264551 2.4117874 1.9177726 2.5690121 2.0227702 2.2982700 1.0166907 0.6899169 1.2747255
[111] 1.1053570 1.4806828 0.6899169 0.6899169 1.1219267 0.6195295 1.6940634 0.7848141 0.9433824 0.2588740
[121] 1.1240012 0.2588740 1.0202345 1.1791769 0.6085809 1.6015536 1.7036371 0.6187648 0.7919622 1.4394136
[131] 1.2451580 1.3514901 1.3514901 1.2451580 1.6896130 1.3458747 1.3514901 1.6015536 1.0149354 1.0149354
[141] 1.0033782 1.0166907 0.6557755 1.6014429 1.4533581 0.6557755 0.6195295 0.6899169 1.0583219 1.0583219
[151] 1.6015536 2.4037455 0.6822739 1.3514901 0.6899169 2.3716575 2.1153300 1.8858203 1.7472385 0.3309058
[161] 0.7248898 0.9433824 1.0583219 1.3951561 1.3129680 2.2724253 0.6195295 0.6195295 0.6195295 1.9456115
[171] 0.9433824 1.0583219 0.9433824 0.6085809 1.3892646 1.1791769 2.4037455 1.3892646 1.6015536 1.0166907
[181] 1.7334103 1.4806828 1.4806828 0.6899169 0.6899169 1.1219267 1.7334103 1.0166907 1.4806828 1.2454374
[191] 0.6899169 0.6899169 1.2747255 2.7514496 1.7044707 2.2774256 2.8974913 2.2774256 1.2451580 0.7919622
[201] 0.6085809 1.3892646 1.0202345 2.4037455 1.1791769 1.7868964 1.3892646 1.0166907 0.6195295 1.0166907
[211] 1.6117639 0.2588740 3.9385183 1.0166907 0.6195295 1.6117639 1.0166907 1.0166907 1.0166907 0.5063281
[221] 0.2588740 5.1893184 2.8143381 1.0166907 0.6195295 0.7848141 1.0603533 1.0166907 2.5040297 1.1219267
[231] 1.0166907 2.3716575 0.6899169 1.4909615

$euclidean
  [1] 3.59772997 4.06534456 4.15364152 4.18498696 2.91273073 3.21260380 3.33746870 3.21260380 2.77894428
```

```
 [10] 3.78040599 3.49610283 2.64002766 2.93014232 2.93014232 2.64002766 2.50293471 2.93014232 2.52364985
 [19] 1.83469613 0.74186992 1.83469613 1.31691304 1.14574307 2.91273073 2.36981505 2.91273073 2.64002766
 [28] 2.50293471 1.71553827 0.63981106 0.74186992 1.50539370 3.49610283 4.35079372 3.21260380 3.59772997
 [37] 3.05731958 2.97561595 2.80725312 2.40344646 2.40344646 2.80725312 2.80725312 1.00080562 2.24137674
 [46] 2.11847141 2.52364985 2.52364985 1.89352563 1.60512876 1.31691304 1.50539370 1.71553827 1.71553827
 [55] 0.07463137 1.00080562 0.74186992 1.31691304 1.31691304 0.07463137 1.31691304 0.86474787 1.42985596
 [64] 0.74186992 1.02903069 0.07463137 1.31691304 1.31691304 1.02903069 0.07463137 0.74186992 0.86474787
 [73] 1.31691304 0.74186992 1.00080562 1.00080562 1.27395263 1.50539370 1.89352563 1.50539370 1.55277857
 [82] 1.55277857 1.31691304 1.50539370 1.50539370 1.21673207 1.21673207 1.31691304 0.74186992 1.31691304
 [91] 3.21260380 3.09182053 3.05731958 2.64905028 2.11847141 2.24137674 2.36981505 2.24137674 1.83469613
[100] 5.74800948 4.92286837 5.09717329 4.41225839 4.92286837 5.49610320 5.56280734 5.40354355 4.06534456
[109] 3.21260380 3.33746870 4.18498696 4.30785451 3.21260380 3.21260380 3.61897973 3.37710836 3.74535694
[118] 3.78040599 3.66294829 2.80725312 2.64905028 2.80725312 2.57677271 1.89352563 2.00211829 1.50539370
[127] 0.07463137 1.71553827 1.42985596 0.63981106 0.74186992 1.27395263 1.27395263 0.74186992 1.00080562
[136] 0.86474787 1.27395263 1.50539370 1.55277857 1.55277857 1.31691304 4.06534456 3.49610283 4.63665916
[145] 4.75378879 3.49610283 3.37710836 3.21260380 3.26242116 3.26242116 1.50539370 1.70545535 1.83469613
[154] 1.27395263 3.21260380 2.91273073 3.18827181 3.46597412 2.77894428 3.09182053 2.97561595 3.66294829
[163] 3.26242116 3.54955153 2.86454734 3.72883208 3.37710836 3.37710836 3.37710836 3.44061724 3.66294829
[172] 3.26242116 3.66294829 2.00211829 2.18203182 1.89352563 1.70545535 2.18203182 1.50539370 4.06534456
[181] 3.83693404 4.30785451 4.30785451 3.21260380 3.21260380 3.61897973 3.83693404 4.06534456 4.30785451
[190] 4.46910512 3.21260380 3.21260380 3.33746870 4.70013178 5.03894213 5.61037124 6.18295102 5.61037124
[199] 0.74186992 1.42985596 2.00211829 2.18203182 2.57677271 1.70545535 1.89352563 1.79410314 2.18203182
[208] 4.06534456 3.37710836 4.06534456 4.23583717 2.80725312 7.84391106 4.06534456 3.37710836 4.23583717
[217] 4.06534456 4.06534456 4.06534456 2.52364985 2.80725312 8.19255504 6.81973959 4.06534456 3.37710836
[226] 3.78040599 3.90155787 4.06534456 3.59772997 3.61897973 4.06534456 2.91273073 3.21260380 3.05731958


which(color.plot(Z)$outlier == TRUE)
```



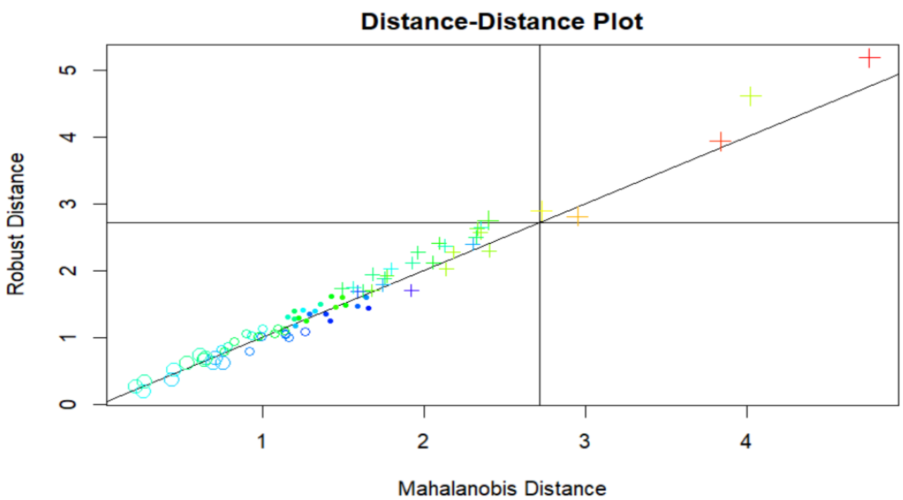

```
## [1] 213 222 223
```

```
dd.plot(Z)
```

```
$outliers
  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [15] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [29] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [43] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [57] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [71] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
 [99] FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[113] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[127] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[141] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[155] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[169] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[183] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE FALSE FALSE
[197]  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[211] FALSE FALSE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE FALSE
[225] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

$md.cla
  [1] 2.3224758 0.9730747 2.0578582 1.1348786 2.1306086 0.6491658 1.1936268 0.6491658
```

```
  [9] 1.5622094 0.7660355 0.6416056 2.3548859 0.7849115 0.7849115 2.3548859 1.7960350
 [17] 0.7849115 0.4527687 0.7066728 1.4193407 0.7066728 1.1649785 1.1421809 2.1306086
 [25] 1.2490633 2.1306086 2.3548859 1.7960350 0.7586461 1.6543154 1.4193407 1.6402287
 [33] 0.6416056 1.2214129 0.6491658 2.3224758 1.3565272 0.6142067 0.2148714 0.2617168
 [41] 0.2617168 0.2148714 0.2148714 1.5914008 0.7407702 0.4375905 0.4527687 0.4527687
 [49] 1.1997777 1.1427446 1.1649785 1.6402287 0.7586461 0.7586461 1.9215241 1.5914008
 [57] 1.4193407 1.1649785 1.1649785 1.9215241 1.1649785 1.3951561 0.9206546 1.4193407
 [65] 1.2622979 1.9215241 1.1649785 1.1649785 1.2622979 1.9215241 1.4193407 1.3951561
 [73] 1.1649785 1.4193407 1.5914008 1.5914008 1.2922546 1.6402287 1.1997777 1.6402287
 [81] 0.9960917 0.9960917 1.1649785 1.6402287 1.6402287 1.5877369 1.5877369 1.1649785
 [89] 1.4193407 1.1649785 0.6491658 0.2687958 1.3565272 0.9978311 0.4375905 0.7407702
 [97] 1.2490633 0.7407702 0.7066728 4.0227314 1.7710261 2.3307371 2.0940586 1.7710261
[105] 2.3496515 2.1384443 2.4034978 0.9730747 0.6491658 1.1936268 1.1348786 1.5110223
[113] 0.6491658 0.6491658 1.0928107 0.5322066 1.6220285 0.7660355 0.8241042 0.2148714
[121] 0.9978311 0.2148714 0.9321096 1.1997777 0.6988520 1.6402287 1.9215241 0.7586461
[129] 0.9206546 1.6543154 1.4193407 1.2922546 1.2922546 1.4193407 1.5914008 1.3951561
[137] 1.2922546 1.6402287 0.9960917 0.9960917 1.1649785 0.9730747 0.6416056 1.4905488
[145] 1.4564712 0.6416056 0.5322066 0.6491658 0.8994408 0.8994408 1.6402287 2.3030637
[153] 0.7066728 1.2922546 0.6491658 2.1306086 1.9283178 1.7556302 1.5622094 0.2687958
[161] 0.6142067 0.8241042 0.8994408 1.1940054 1.1560330 1.9617929 0.5322066 0.5322066
[169] 0.5322066 1.6811644 0.8241042 0.8994408 0.8241042 0.6988520 1.3258881 1.1997777
[177] 2.3030637 1.3258881 1.6402287 0.9730747 1.4923857 1.5110223 1.5110223 0.6491658
[185] 0.6491658 1.0928107 1.4923857 0.9730747 1.5110223 1.2697139 0.6491658 0.6491658
[193] 1.1936268 2.3961180 1.6779019 2.1813256 2.7277559 2.1813256 1.4193407 0.9206546
[201] 0.6988520 1.3258881 0.9321096 2.3030637 1.1997777 1.7449077 1.3258881 0.9730747
[209] 0.5322066 0.9730747 1.4232671 0.2148714 3.8378886 0.9730747 0.5322066 1.4232671
[217] 0.9730747 0.9730747 0.9730747 0.4527687 0.2148714 4.7596224 2.9511749 0.9730747
[225] 0.5322066 0.7660355 1.0717447 0.9730747 2.3224758 1.0928107 0.9730747 2.1306086
[233] 0.6491658 1.3565272

$md.rob
  [1] 2.5040297 1.0166907 2.1186177 1.1053570 2.3716575 0.6899169 1.2747255 0.6899169
  [9] 1.7472385 0.7848141 0.6557755 2.6470233 0.8681960 0.8681960 2.6470233 2.0284363
 [17] 0.8681960 0.5063281 0.6822739 1.2451580 0.6822739 1.0033782 1.0504160 2.3716575
 [25] 1.4159195 2.3716575 2.6470233 2.0284363 0.6187648 1.4394136 1.2451580 1.6015536
 [33] 0.6557755 1.2973904 0.6899169 2.5040297 1.4909615 0.7248898 0.2588740 0.1783561
```

```
 [41] 0.1783561 0.2588740 0.2588740 1.6896130 0.8231378 0.3643304 0.5063281 0.5063281
 [49] 1.1791769 1.0405264 1.0033782 1.6015536 0.6187648 0.6187648 1.7036371 1.6896130
 [57] 1.2451580 1.0033782 1.0033782 1.7036371 1.0033782 1.3458747 0.7919622 1.2451580
 [65] 1.0782743 1.7036371 1.0033782 1.0033782 1.0782743 1.7036371 1.2451580 1.3458747
 [73] 1.0033782 1.2451580 1.6896130 1.6896130 1.3514901 1.6015536 1.1791769 1.6015536
 [81] 1.0149354 1.0149354 1.0033782 1.6015536 1.6015536 1.4726794 1.4726794 1.0033782
 [89] 1.2451580 1.0033782 0.6899169 0.3309058 1.4909615 1.1240012 0.3643304 0.8231378
 [97] 1.4159195 0.8231378 0.6822739 4.6157223 1.9177726 2.6264551 2.4117874 1.9177726
[105] 2.5690121 2.0227702 2.2982700 1.0166907 0.6899169 1.2747255 1.1053570 1.4806828
[113] 0.6899169 0.6899169 1.1219267 0.6195295 1.6940634 0.7848141 0.9433824 0.2588740
[121] 1.1240012 0.2588740 1.0202345 1.1791769 0.6085809 1.6015536 1.7036371 0.6187648
[129] 0.7919622 1.4394136 1.2451580 1.3514901 1.3514901 1.2451580 1.6896130 1.3458747
[137] 1.3514901 1.6015536 1.0149354 1.0149354 1.0033782 1.0166907 0.6557755 1.6014429
[145] 1.4533581 0.6557755 0.6195295 0.6899169 1.0583219 1.0583219 1.6015536 2.4037455
[153] 0.6822739 1.3514901 0.6899169 2.3716575 2.1153300 1.8858203 1.7472385 0.3309058
[161] 0.7248898 0.9433824 1.0583219 1.3951561 1.3129680 2.2724253 0.6195295 0.6195295
[169] 0.6195295 1.9456115 0.9433824 1.0583219 0.9433824 0.6085809 1.3892646 1.1791769
[177] 2.4037455 1.3892646 1.6015536 1.0166907 1.7334103 1.4806828 1.4806828 0.6899169
[185] 0.6899169 1.1219267 1.7334103 1.0166907 1.4806828 1.2454374 0.6899169 0.6899169
[193] 1.2747255 2.7514496 1.7044707 2.2774256 2.8974913 2.2774256 1.2451580 0.7919622
[201] 0.6085809 1.3892646 1.0202345 2.4037455 1.1791769 1.7868964 1.3892646 1.0166907
[209] 0.6195295 1.0166907 1.6117639 0.2588740 3.9385183 1.0166907 0.6195295 1.6117639
[217] 1.0166907 1.0166907 1.0166907 0.5063281 0.2588740 5.1893184 2.8143381 1.0166907
[225] 0.6195295 0.7848141 1.0603533 1.0166907 2.5040297 1.1219267 1.0166907 2.3716575
[233] 0.6899169 1.4909615

which(dd.plot(Z)$outliers == TRUE)
[1] 100 194 197 213 222 223
```




```
Y <- as.matrix(mpg[, c("cty", "hwy")])
res <- aq.plot(Y)
```