Actividad 2.6. Análisis discriminante

Franco Mendoza Muraira A01383399

2023-11-17

1. Designa tu variable categórica como variable dependiente para una clasificación y tus variables numéricas como variables independientes.

```
## Loading required package: ggplot2
```

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

```
##           id            date   price bedrooms bathrooms sqft_living sqft_lot
## 1 7129300520 20141013T000000  221900        3      1.00        1180     5650
## 2 6414100192 20141209T000000  538000        3      2.25        2570     7242
## 3 5631500400 20150225T000000  180000        2      1.00         770    10000
## 4 2487200875 20141209T000000  604000        4      3.00        1960     5000
## 5 1954400510 20150218T000000  510000        3      2.00        1680     8080
## 6 7237550310 20140512T000000 1225000        4      4.50        5420   101930
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0         3     7       1180             0     1955
## 2      2          0    0         3     7       2170           400     1951
## 3      1          0    0         3     6        770             0     1933
## 4      1          0    0         5     7       1050           910     1965
## 5      1          0    0         3     8       1680             0     1987
## 6      1          0    0         3    11       3890          1530     2001
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2         1991   98125 47.7210 -122.319          1690       7639
## 3            0   98028 47.7379 -122.233          2720       8062
## 4            0   98136 47.5208 -122.393          1360       5000
## 5            0   98074 47.6168 -122.045          1800       7503
## 6            0   98053 47.6561 -122.005          4760     101930
```

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
```

```
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...

## 'data.frame':    21097 obs. of  22 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
##  $ Category     : Factor w/ 3 levels "high","low","medium": 2 3 2 3 3 1 2 2 2 2 ...

##   bedrooms bathrooms sqft_living sqft_lot floors condition grade sqft_above
## 1        3      1.00        1180     5650      1         3     7       1180
## 2        3      2.25        2570     7242      2         3     7       2170
## 3        2      1.00         770    10000      1         3     6        770
## 4        4      3.00        1960     5000      1         5     7       1050
## 5        3      2.00        1680     8080      1         3     8       1680
##   yr_built zipcode     lat     long sqft_living15 sqft_lot15 Category
## 1     1955   98178 47.5112 -122.257          1340       5650      low
## 2     1951   98125 47.7210 -122.319          1690       7639   medium
## 3     1933   98028 47.7379 -122.233          2720       8062      low
## 4     1965   98136 47.5208 -122.393          1360       5000   medium
## 5     1987   98074 47.6168 -122.045          1800       7503   medium
```

2. Acota tu base de datos realizando un muestreo aleatorio de 300 observaciones.

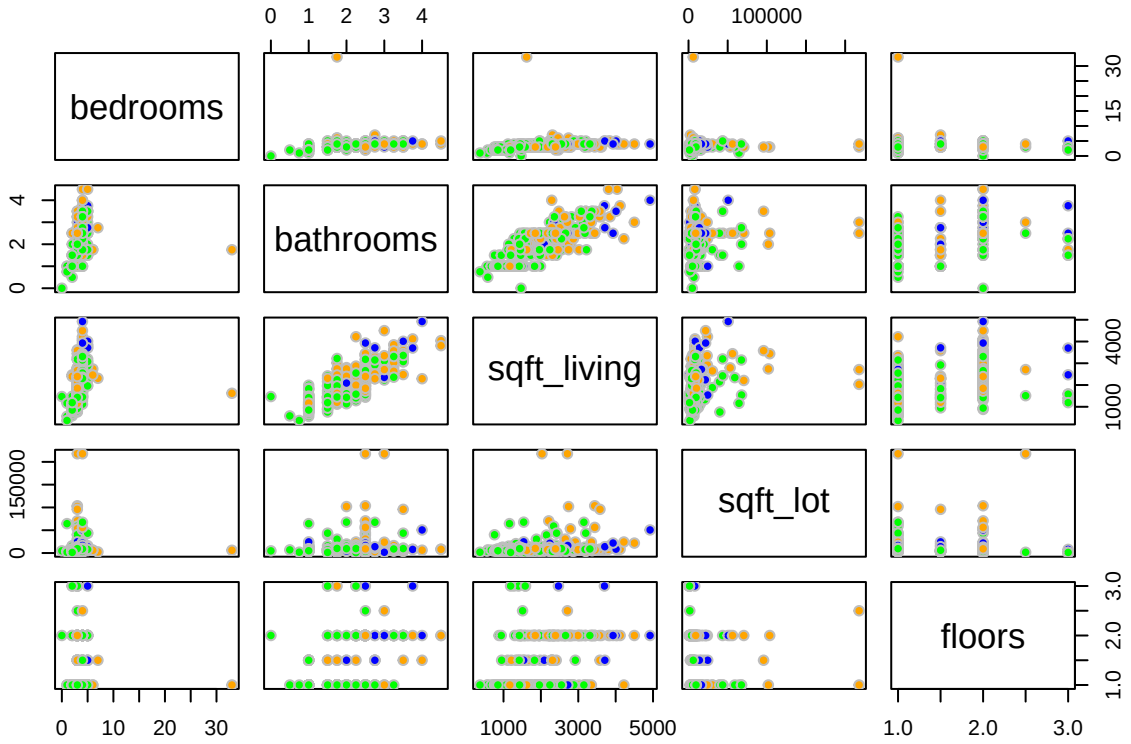
```
set.seed(123)
M_sample <- M2[sample(nrow(M2), 300), ]
```

3. Muestre gráficamente la segmentación original de los datos. Realiza un gráfico de dispersión donde se identifiquen las diferentes categorías de tu base de datos. ¿Qué variable o variables discriminan mejor?

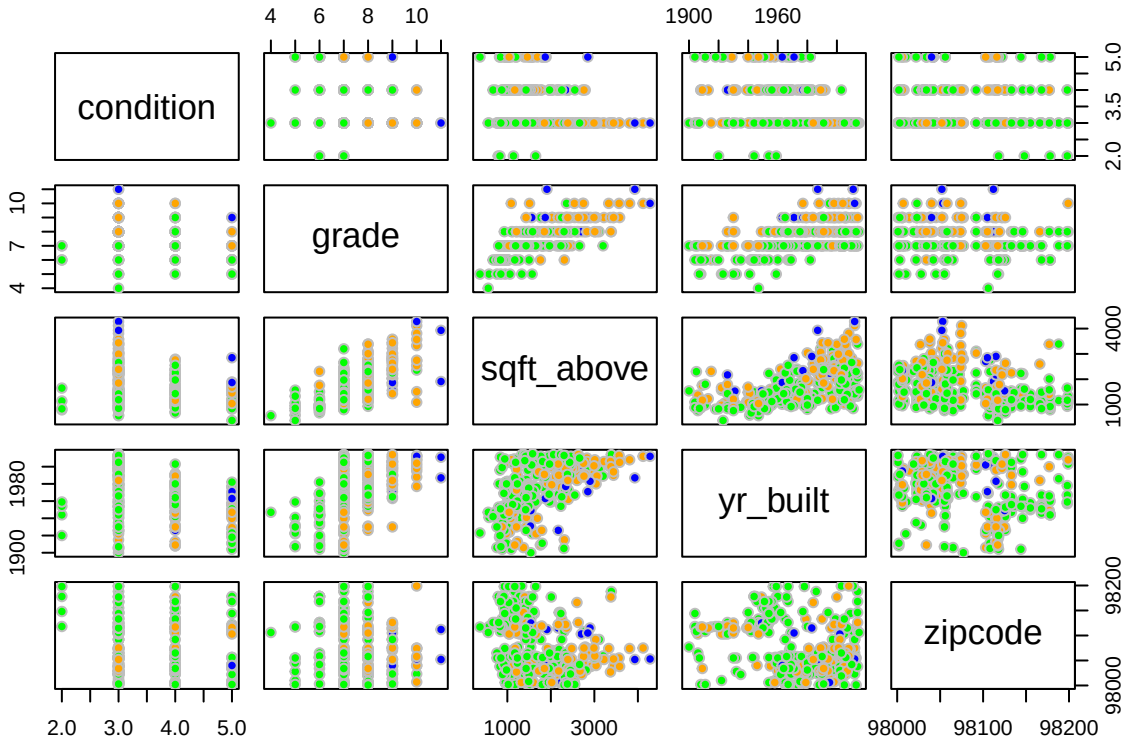
```
library(ggplot2)
#Asignamos un color a cada especie
color = c(low="blue",medium="green",high="orange")
color
```

```
##      low   medium     high
##   "blue"  "green" "orange"
```

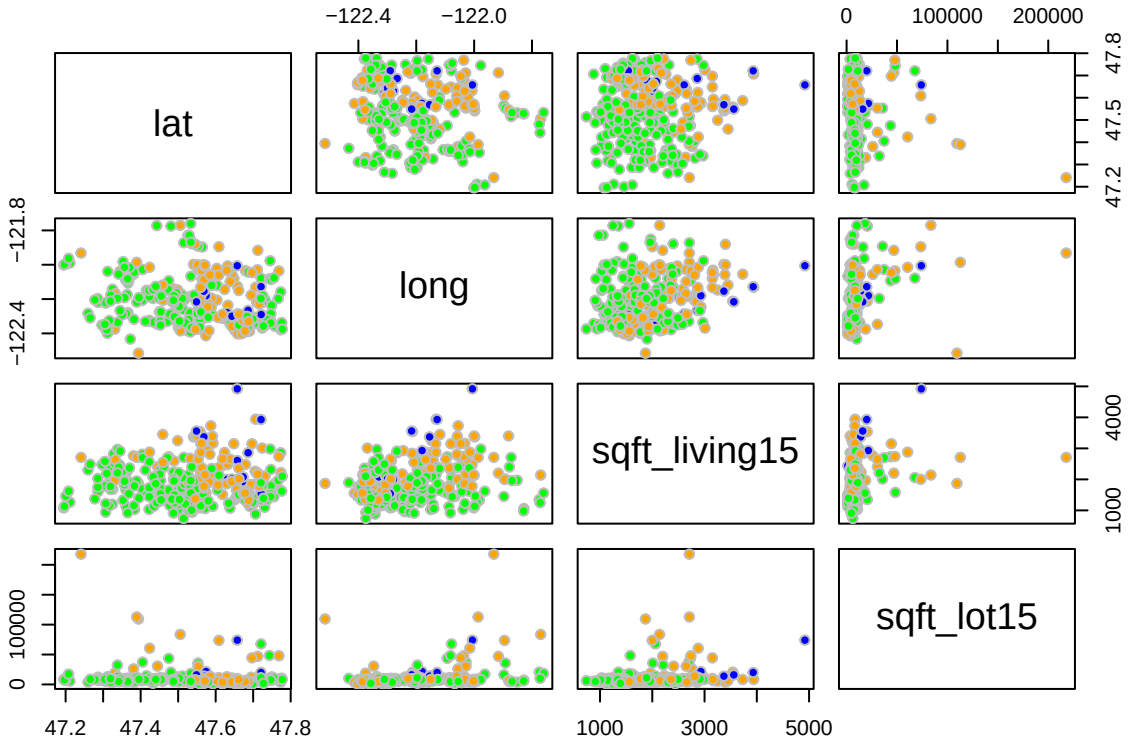
```
#Creamos un vector con el color corresponidente a cada observacion de acuerdo a la columna Species
col.ind=color[M_sample$Category]
plot(M_sample[1:5],pch=21,bg=col.ind,col="gray")
```




```
plot(M_sample[6:10],pch=21,bg=col.ind,col="gray")
```




```
plot(M_sample[11:14],pch=21,bg=col.ind,col="gray")
```



Debido a la baja utilidad que nos dan las variables de zipcode, yr_built, lat y long las eliminaremos, ya que tambien hacen mucho cambio en las graficas.

También como no se ve mucha discriminación en las otras variables, solo dejamos las de condition, grade, sqft_living, y sqft_lot15. En estas se nota mucho la discriminación en las categorias.

```
M_nueva= M_sample[,-c(1,2,3,4,5,8,9,10,11,12)]
head(M_nueva,5)
```

```
##       condition grade sqft_living15 sqft_lot15 Category
## 19291         3     7          1610       4756      low
## 19341         3     8          1910       9348   medium
## 3058          4     7          1610      10640      low
## 1878          3     9          3170       6285   medium
## 3453          3    11          2440       1229     high
```

4. Realiza un análisis discriminante para responder las siguientes preguntas:

a) Obtener la media para cada variable predictora en función del grupo:

```
numeric_cols <- M_nueva[, 1:4]
categories <- M_nueva$Category
```

```r
means_by_category <- aggregate(numeric_cols, by = list(Category = categories), FUN = mean)
print(means_by_category[,-6])
```

```
##   Category condition    grade sqft_living15 sqft_lot15
## 1     high  3.538462 8.846154      2781.538   14086.69
## 2      low  3.442105 7.073684      1683.811    8464.40
## 3   medium  3.402062 8.123711      2244.227   15514.86
```

Aquí tenemos las medias de las variables predictoras que seleccionamos en el análisis discriminante.

b) Mostrar las probabilidades a priori para las diferentes clases:

```r
prior_prob <- prop.table(table(M_nueva$Category))
print(prior_prob)
```

```
##
##       high        low     medium
## 0.04333333 0.63333333 0.32333333
```

c) Determinar la función discriminante lineal:

```r
lda_model <- lda(Category ~ ., data = M_nueva)
print(lda_model)
```

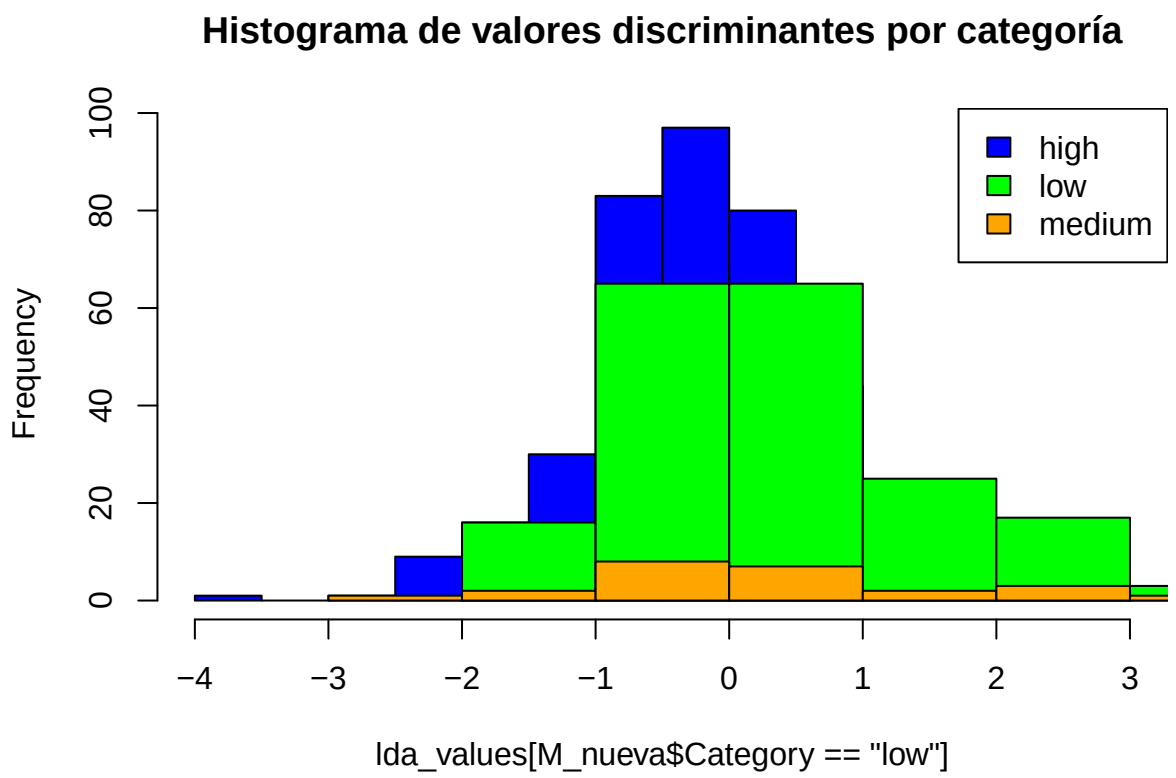
```
## Call:
## lda(Category ~ ., data = M_nueva)
##
## Prior probabilities of groups:
##       high        low     medium
## 0.04333333 0.63333333 0.32333333
##
## Group means:
##        condition    grade sqft_living15 sqft_lot15
## high    3.538462 8.846154      2781.538   14086.69
## low     3.442105 7.073684      1683.811    8464.40
## medium  3.402062 8.123711      2244.227   15514.86
##
## Coefficients of linear discriminants:
##                          LD1           LD2
## condition     4.196187e-01 -5.798928e-01
## grade         7.403019e-01  4.765653e-01
## sqft_living15 7.823153e-04 -1.261759e-03
## sqft_lot15    4.199328e-06  4.219879e-05
##
## Proportion of trace:
##    LD1    LD2
## 0.9801 0.0199
```

Este es nuestro modelo/ funcion discriminante lineal, podemos tambien observar las probabilidades a priori de las clasificaciones. low: 0.633, medium: 0.323 y high: 0.043.

Tambien podemos observar los coeficientes de las variables discriminantes del modelo, el coeficiente más alto siendo el de grade con 7.403e-01 en el LD1 y 4.7656e-01 en el LD2. La proporción de trace del modelo es .9801 en el LD1 y 0.0199 en el LD2.

d) Graficar el histograma de valores discriminantes en cada grupo:

```
lda_values <- predict(lda_model)$x
hist(lda_values[M_nueva$Category == "low"], col = "blue", main = "Histograma de valores discriminantes p
hist(lda_values[M_nueva$Category == "medium"], col = "green", add = TRUE)
hist(lda_values[M_nueva$Category == "high"], col = "orange", add = TRUE)
legend("topright", legend = levels(M_nueva$Category), fill = c("blue", "green", "orange"))
```




```
predicted = predict(lda_model)

names(predicted)
```

```
## [1] "class"     "posterior" "x"
```

```
head(predicted$class)
```

```
## [1] low    low    low    medium medium low
## Levels: high low medium
```

```
head(predicted$posterior)
```

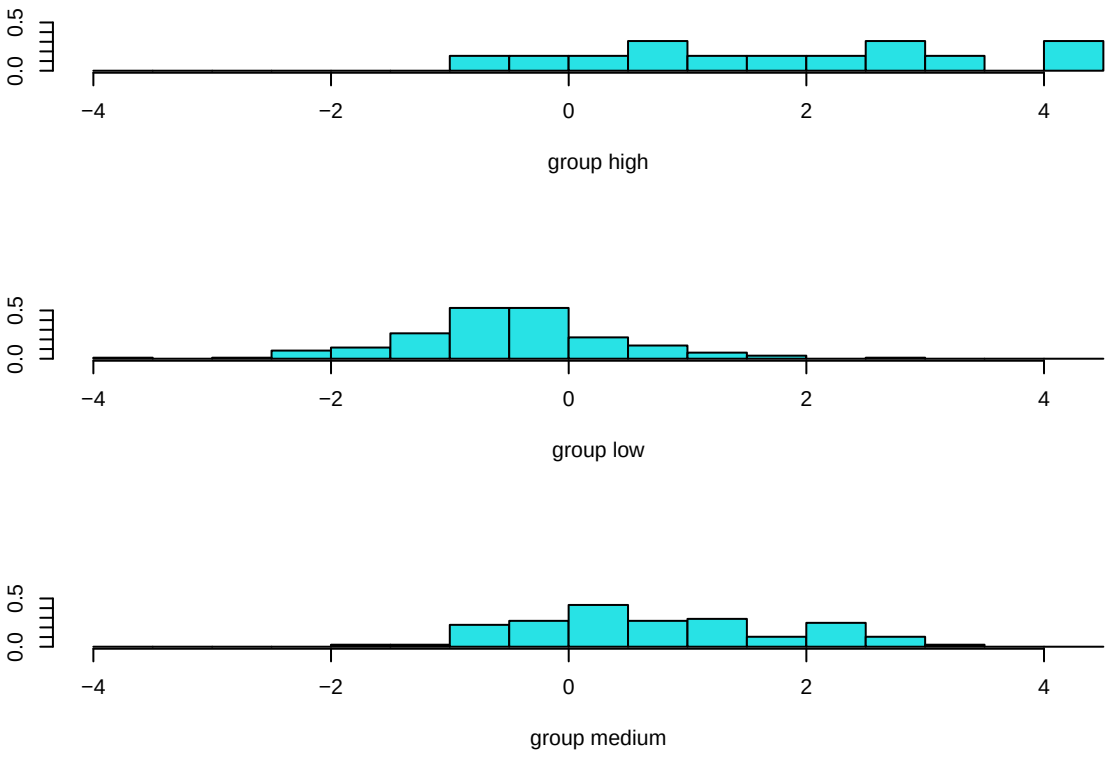
```
##              high        low    medium
## 19291 0.002120339 0.85639014 0.1414895
## 19341 0.012858710 0.62499298 0.3621483
## 3058  0.005871926 0.77995137 0.2141767
## 1878  0.223886465 0.15140311 0.6247104
## 3453  0.272301863 0.04590924 0.6817889
## 11897 0.007264017 0.77322425 0.2195117
```

```
head(predicted$x)
```

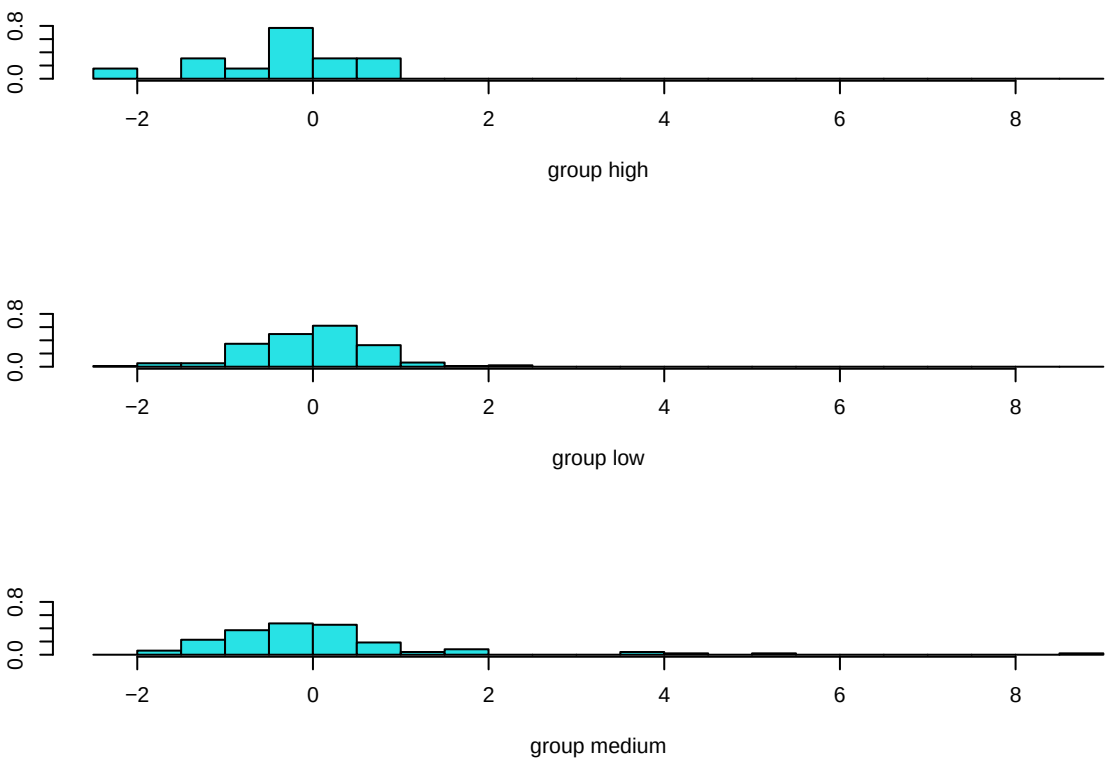
```
##              LD1        LD2
## 19291 -0.8074645  0.1365835
## 19341  0.1868153  0.4283980
## 3058  -0.3631370 -0.1950116
## 1878   1.8999719 -0.8141076
## 3453   2.7882537  0.8467499
## 11897 -0.3096939 -0.4790530
```

e) Mostrar gráficamente la segmentación de los datos con predicciones del modelo:

```
ldahist(data=predicted$x[,1],g=M_nueva$Category, main="Histograma de la función discriminante LD1")
```

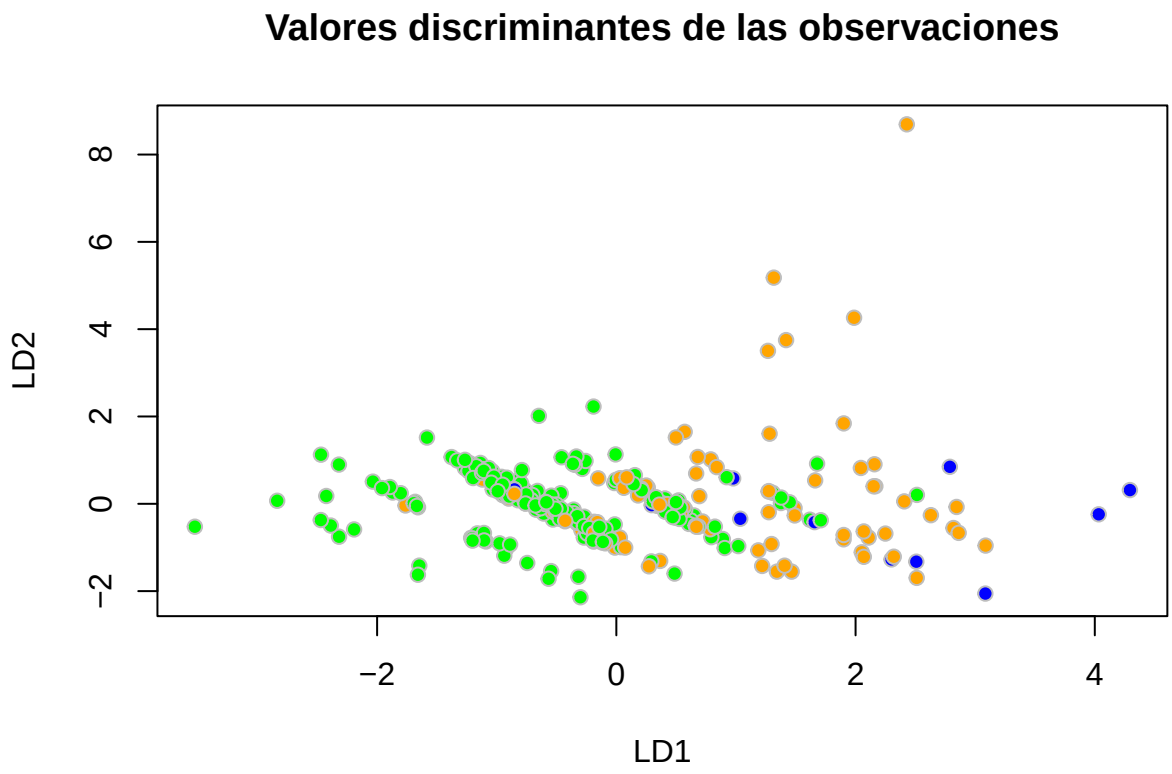



```
ldahist(data=predicted$x[,2],g=M_nueva$Category, main="Histograma de la función discriminante LD2")
```



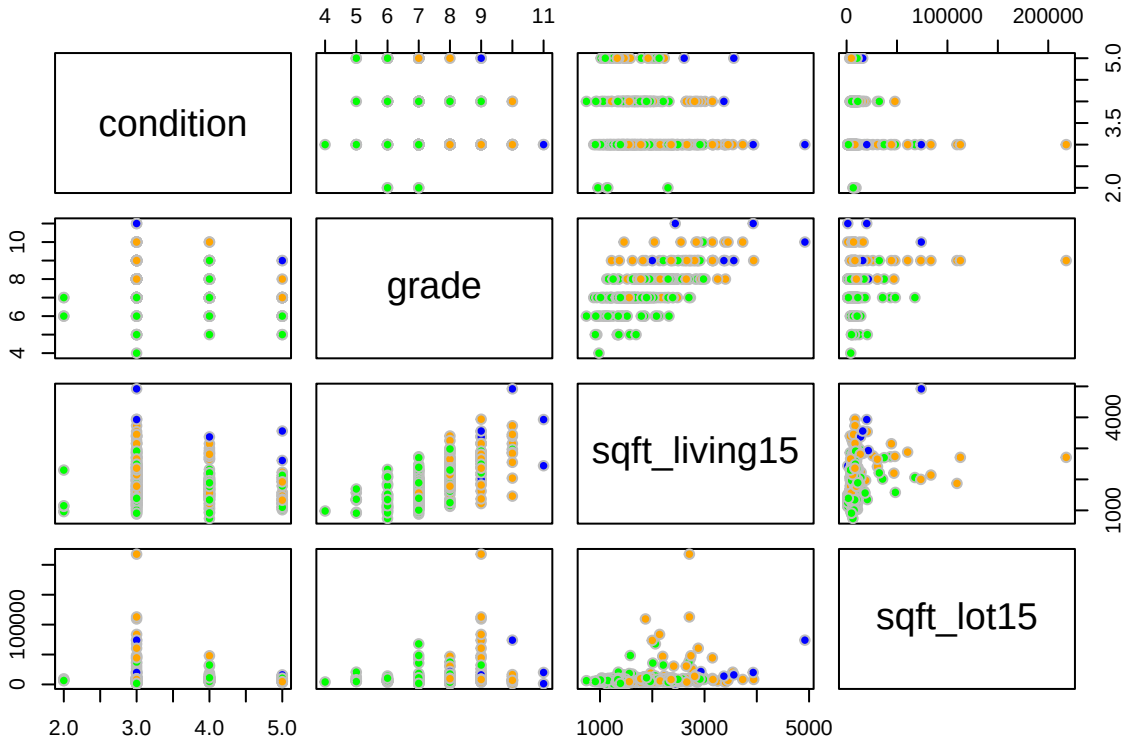
Visualizaremos la forma en que cada uno de las funciones discriminantes lineales separan las tres clases diferentes

```
#definimos los datos a graficar
plot(LD2~LD1,data = predicted$x,pch=21,col="gray",bg=col.ind,main="Valores discriminantes de las observa
```

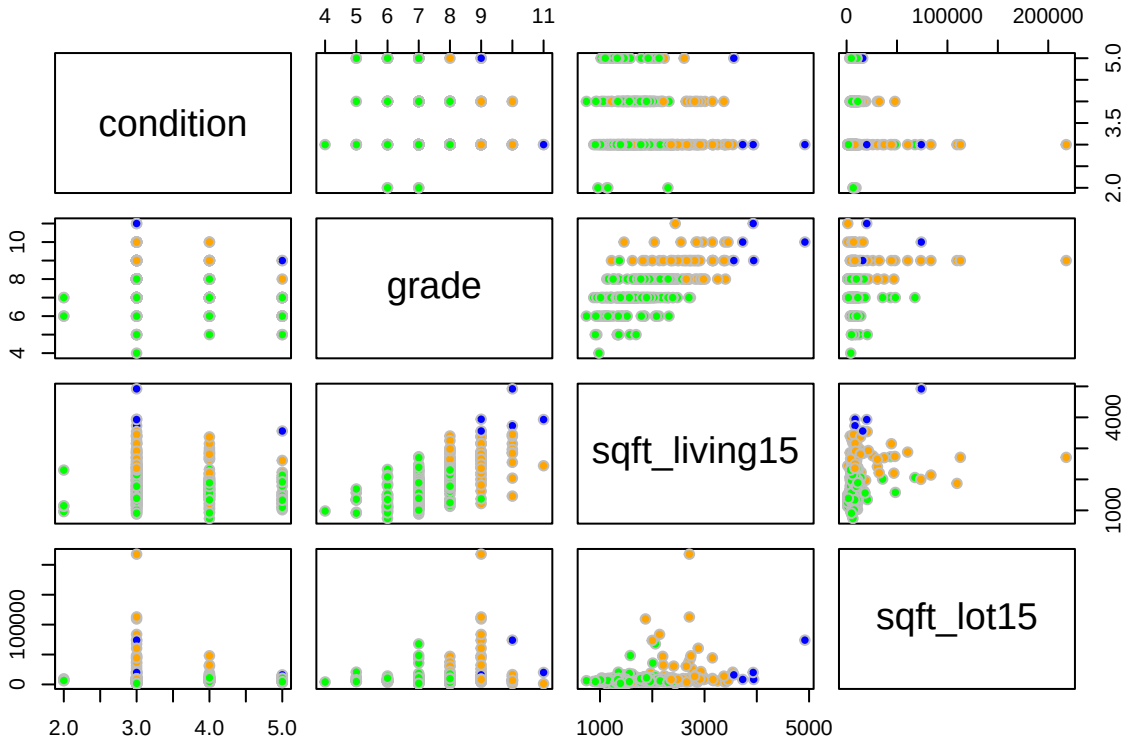


Visualizaremos la clasificación realizada por las predicciones:

```
#Asignamos un color a cada categoria
color2=c(low='blue',medium='green',high='orange')

#Creamos un vector con el color corresponidente a cada observacion de acuerdo a la columna Category
col.ind2=color2[predicted$class]

#Graficos de dispersion con el color de acuerdo al tipo de categoria
plot(M_nueva[-5],pch=21,bg=col.ind,col='gray')
```




```
plot(M_nueva[-5],pch=21,bg=col.ind2,col='gray')
```



Podemos ver los scatter plots de las variables seleccionadas, junto con las predicciones que se hicieron con el modelo.

f) Evaluar la precisión del modelo:

```
confusion_matrix <- table(predicted$class, M_nueva$Category)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
print(paste("Precisión del modelo:", round(accuracy * 100, 2), "%"))
```

```
## [1] "Precisión del modelo: 73.67 %"
```

```
print(confusion_matrix)
```

```
##
##           high low medium
##   high       3   0      2
##   low        4 175     52
##   medium     6  15     43
```

```
# porcentaje de observaciones clasificadas erróneamente
rate=1-mean(predicted$class==M_nueva$Category)

cat("\n El porcentaje de observaciones clasificadas erróneamente es: ",rate*100,"%")
```

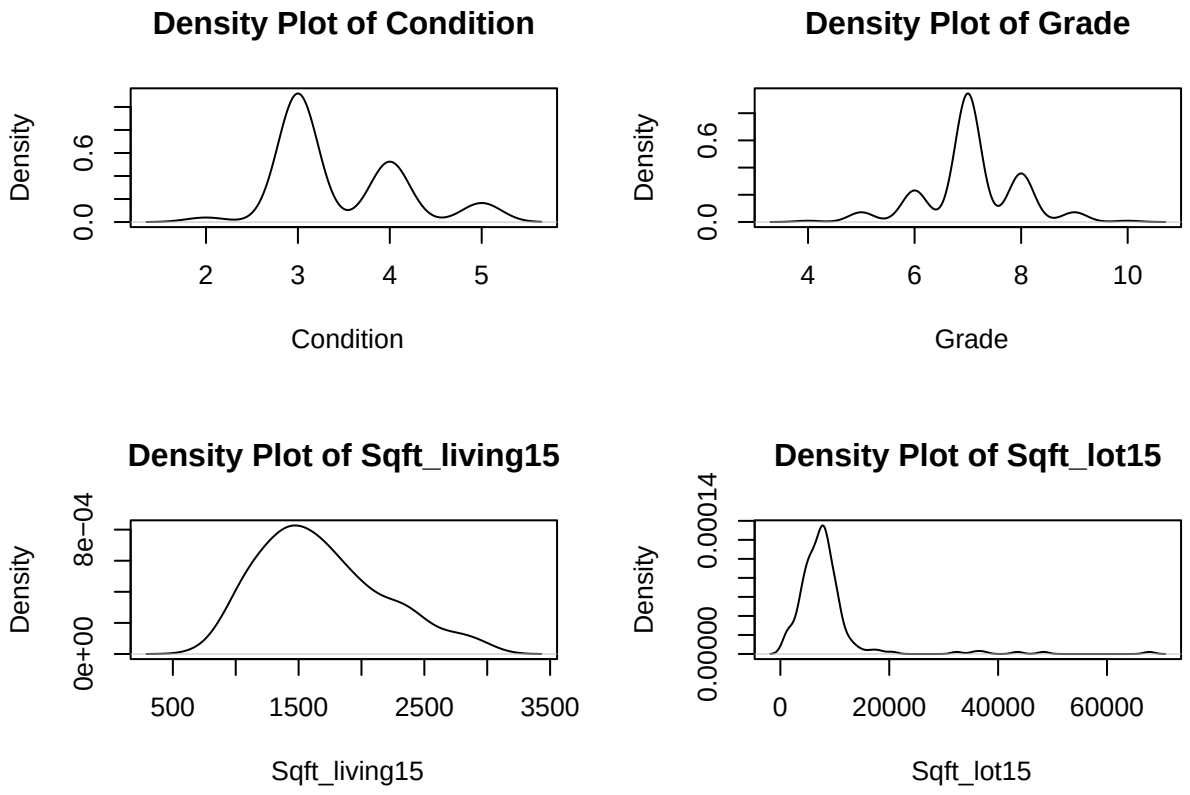
```
##
##  El porcentaje de observaciones clasificadas erróneamente es:  26.33333 %
```

Podemos ver que la precisión del modelo es bastante alta, con una de 73.67%, en la matriz de confusión se ven algunos pocos errores, que son significantes en este caso, cosa que podría ser debido a la baja cantidad de datos.

5. Validar los supuestos del modelo:

Low

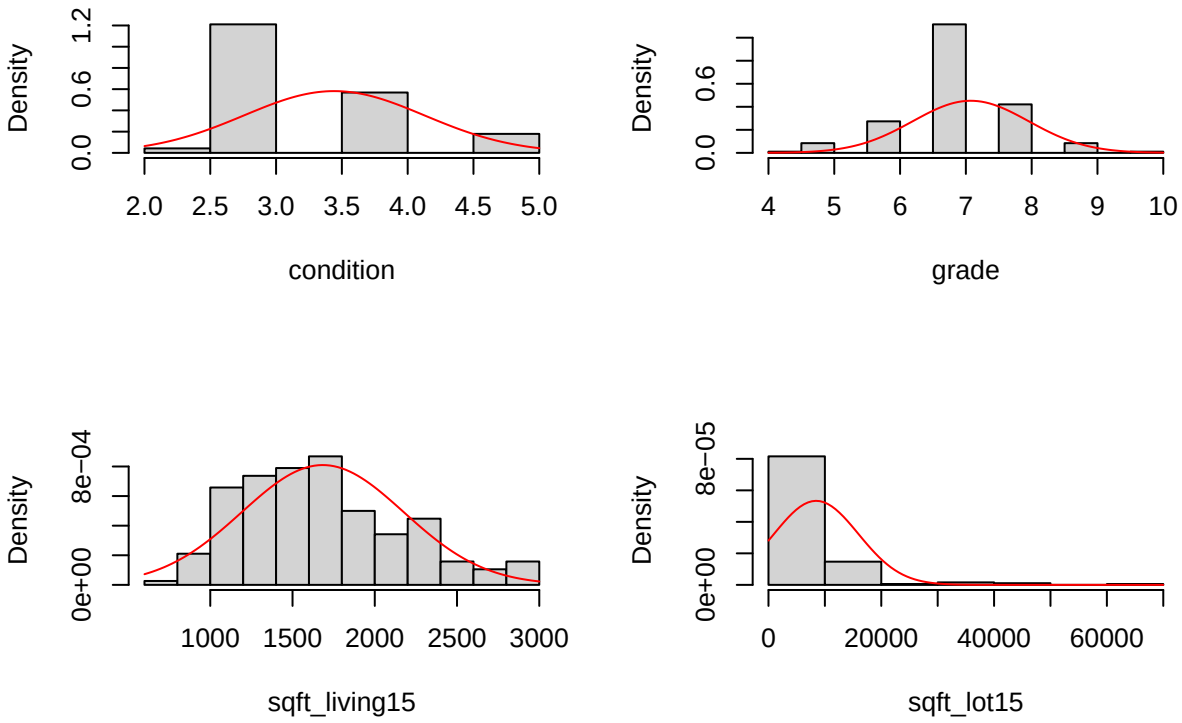
```
# Graficos de densidad
low = M_nueva[M_nueva[,5]=='low',]
medium = M_nueva[M_nueva[,5]=='medium',]
high = M_nueva[M_nueva[,5]=='high',]

par(mfrow=c(2,2))
plot(density(low$condition),main = "Density Plot of Condition",xlab =' Condition')
plot(density(low$grade),main = "Density Plot of Grade",xlab =' Grade')
plot(density(low$sqft_living15),main = "Density Plot of Sqft_living15",xlab =' Sqft_living15')
plot(density(low$sqft_lot15),main = "Density Plot of Sqft_lot15",xlab =' Sqft_lot15')
```



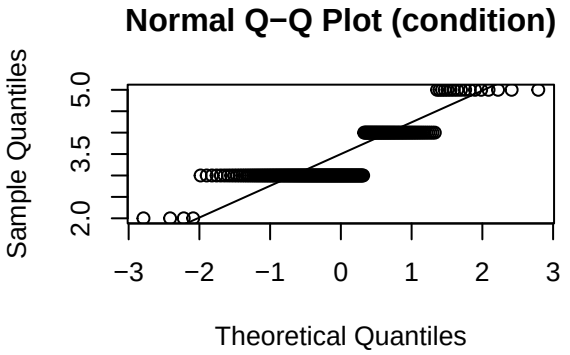
```
head(low)
```

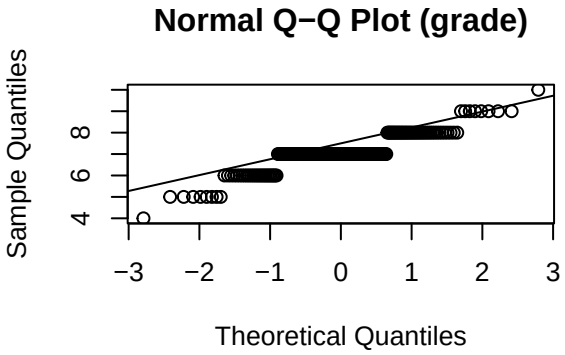
```
##       condition grade sqft_living15 sqft_lot15 Category
## 19291         3     7          1610       4756      low
## 3058          4     7          1610      10640      low
## 11897         4     7          1700       6600      low
## 4875          4     7          1720       7575      low
## 16506         4     7          1830       8000      low
## 2822          4     6          1500       4800      low
```

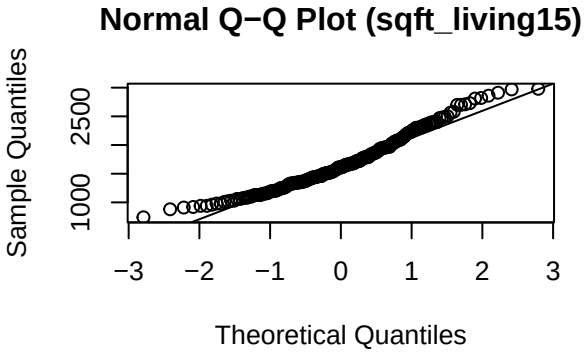
```
lowhist = mvn(data=low[,1:4],mvnTest = "royston",univariatePlot = "histogram")
```

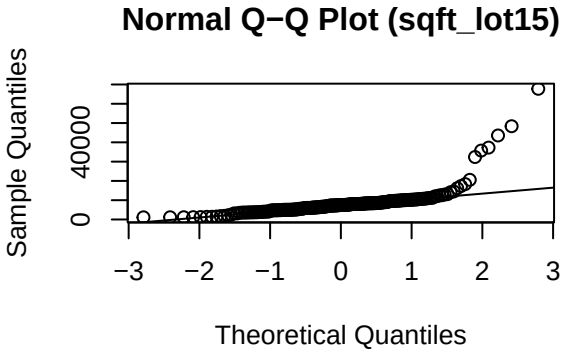



```
lowqq = mvn(data = low[,1:4],mvnTest = "royston",univariatePlot = "qqplot")
```









Pruebas de normalidad

H_0 : Los datos tienen una distribución normal.

H_1 : Los datos no tienen una distribución normal.

```
mvn(data=low[,1:4],mvnTest = "mardia")
```

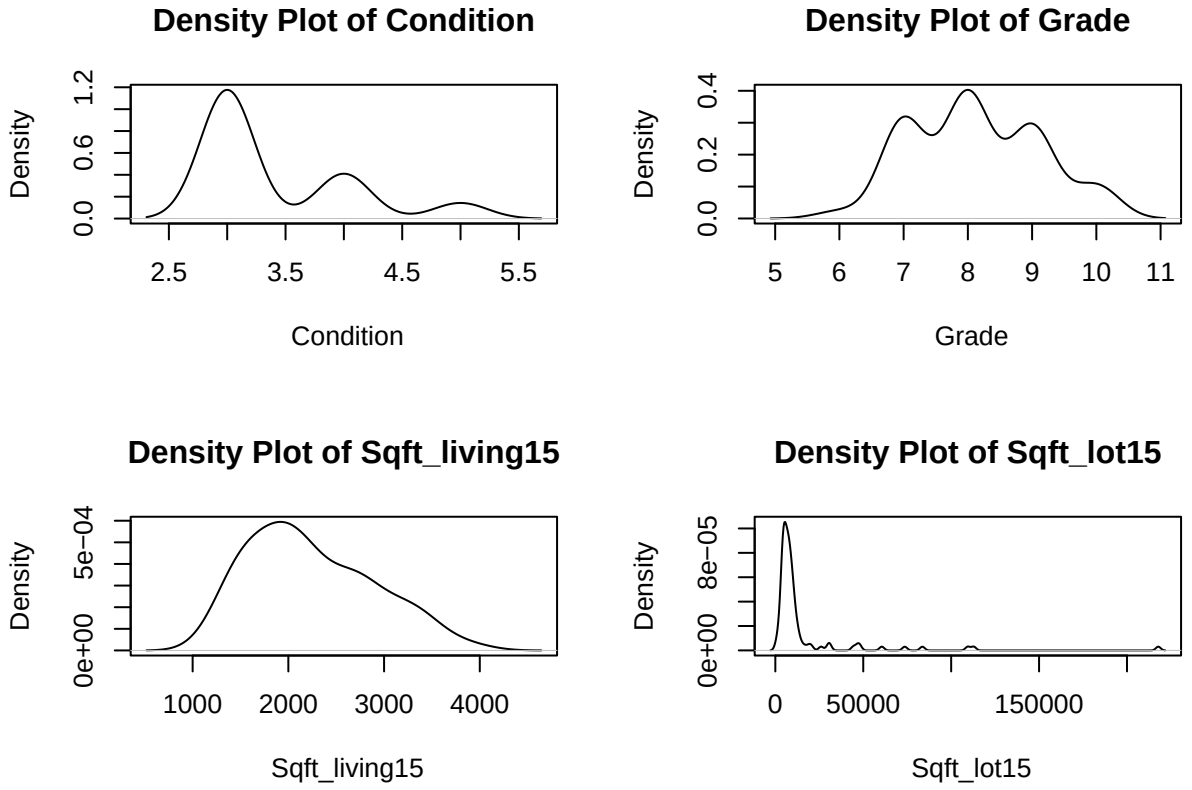
```
## $multivariateNormality
##              Test        Statistic               p value Result
## 1 Mardia Skewness 803.942412707119 2.06026794904789e-157     NO
## 2 Mardia Kurtosis 27.6924723127827                     0     NO
## 3             MVN             <NA>                 <NA>     NO
##
## $univariateNormality
##               Test      Variable Statistic   p value Normality
## 1 Anderson-Darling   condition     23.4291  <0.001      NO
## 2 Anderson-Darling     grade       13.4210  <0.001      NO
## 3 Anderson-Darling sqft_living15    1.8986   1e-04      NO
## 4 Anderson-Darling  sqft_lot15     22.2091  <0.001      NO
##
## $Descriptives
##                 n        Mean      Std.Dev Median  Min   Max   25th   75th
## condition     190    3.442105    0.6856934      3    2     5    3.0    4.0
## grade         190    7.073684    0.8818224      7    4    10    7.0    8.0
```

```
## sqft_living15 190 1683.810526  492.7343918   1610  740  2980 1332.5 1970.0
## sqft_lot15    190 8464.400000 7478.8968203   7500 1126 67756 5127.5 9301.5
##                      Skew    Kurtosis
## condition       0.8443517  0.02514699
## grade          -0.1424850  1.21866847
## sqft_living15   0.5921870 -0.28408466
## sqft_lot15      4.6620556 27.38405843
```

Debido a que los valores p de las pruebas de mardia de normalidad multivariada son menores al α de 0.05, se concluye que los datos de la categoria high no tienen una distribución normal.

Medium

```
par(mfrow=c(2,2))
plot(density(medium$condition),main = "Density Plot of Condition",xlab =' Condition')
plot(density(medium$grade),main = "Density Plot of Grade",xlab =' Grade')
plot(density(medium$sqft_living15),main = "Density Plot of Sqft_living15",xlab =' Sqft_living15')
plot(density(medium$sqft_lot15),main = "Density Plot of Sqft_lot15",xlab =' Sqft_lot15')
```

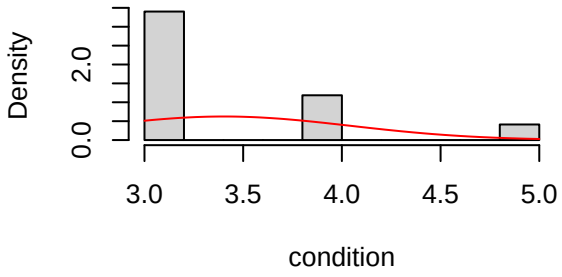


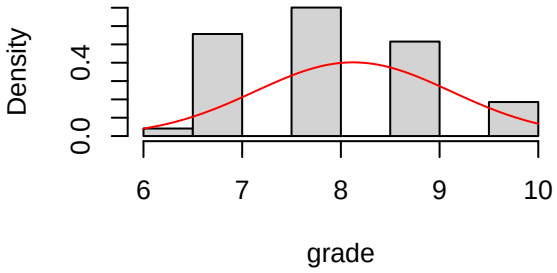

```
head(medium)
```

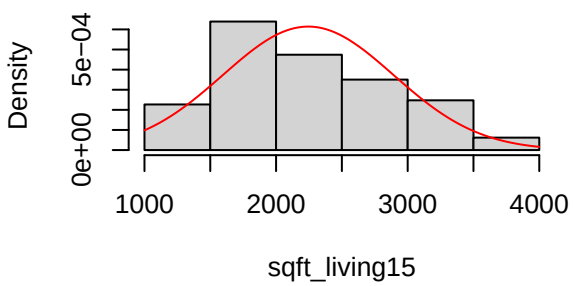
```
##       condition grade sqft_living15 sqft_lot15 Category
## 19341         3     8          1910       9348   medium
```

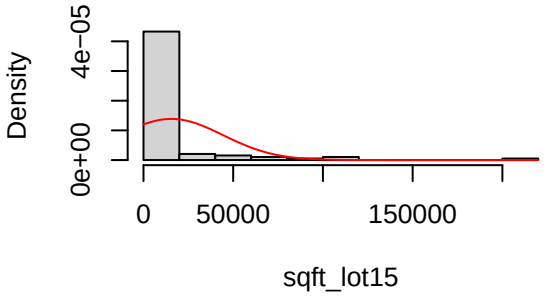
```
## 1878          3     9          3170       6285   medium
## 6902          5     7          2010       9943   medium
## 12917         3    10          3150       7515   medium
## 9857          4     8          1970      18893   medium
## 17336         3     6          1340       3825   medium
```

```
mediumhist = mvn(data=medium[,1:4],mvnTest = "royston",univariatePlot = "histogram")
```

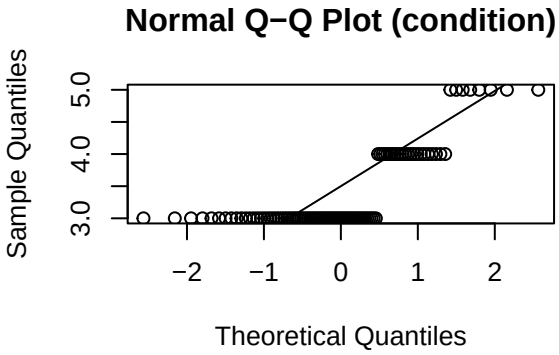


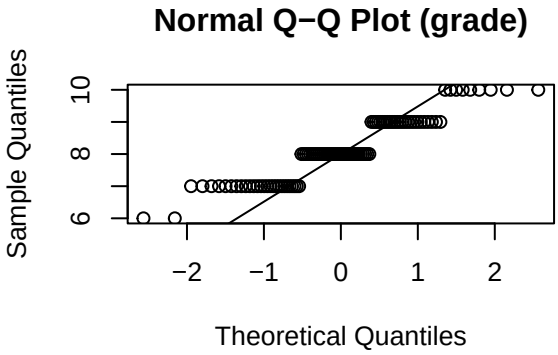


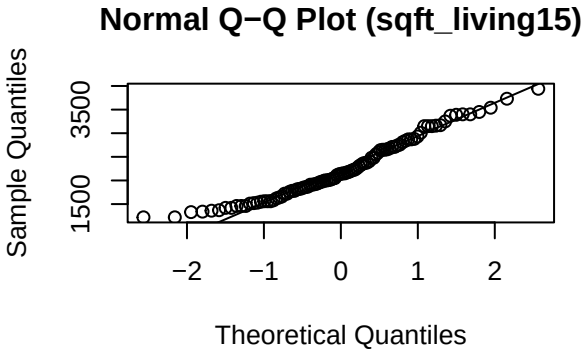


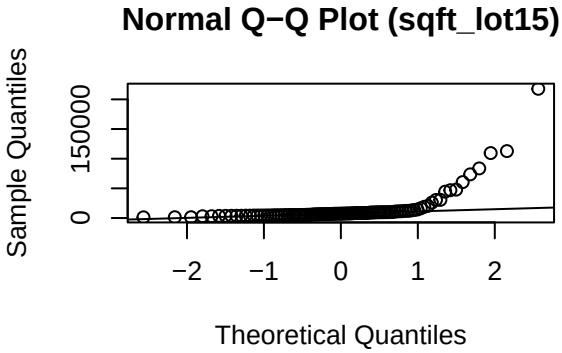



```
mediumqq = mvn(data = medium[,1:4],mvnTest = "royston",univariatePlot = "qqplot")
```









Pruebas de normalidad

H_0 : Los datos tienen una distribución normal.

H_1 : Los datos no tienen una distribución normal.

```
mvn(data=medium[,1:4],mvnTest = "mardia")
```

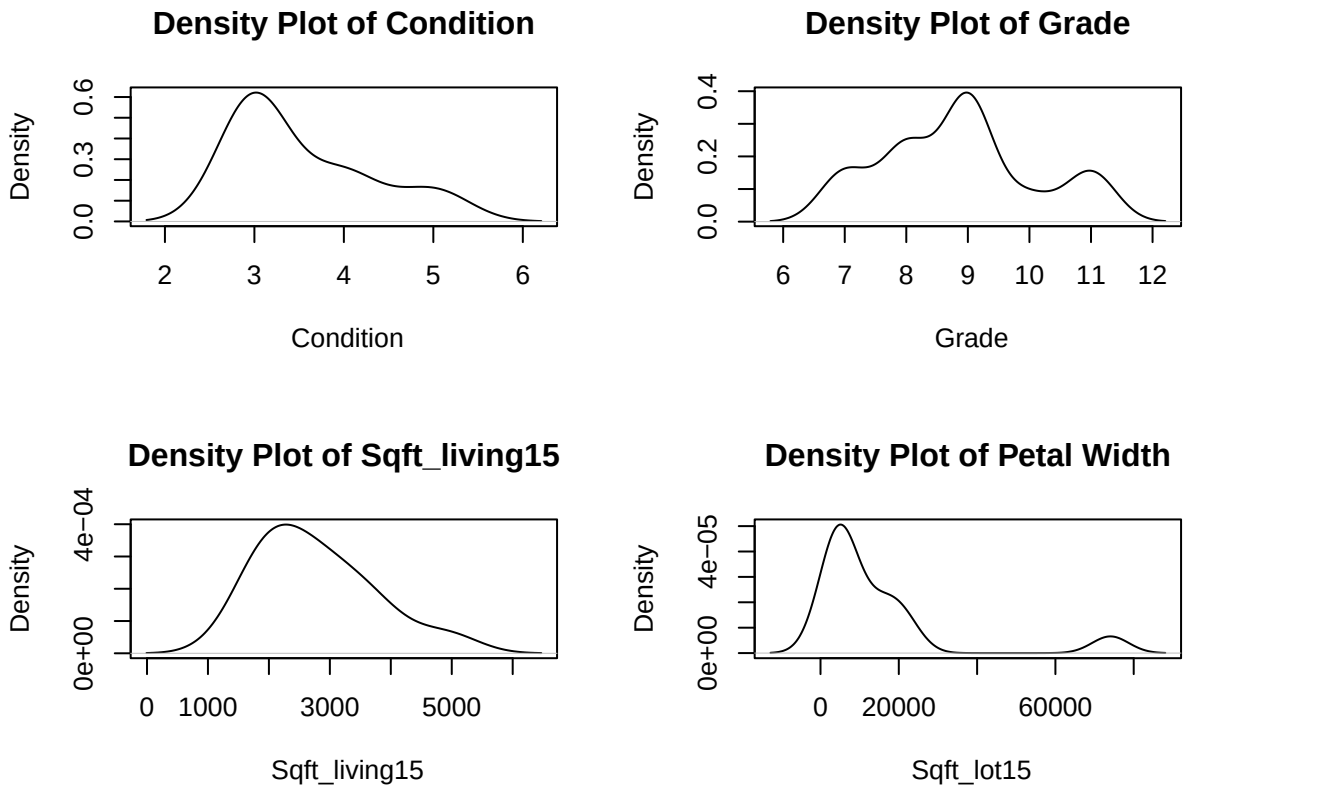
```
## $multivariateNormality
##              Test        Statistic              p value Result
## 1 Mardia Skewness 419.046827169528 2.26693433390328e-76     NO
## 2 Mardia Kurtosis 16.0269854594108                    0     NO
## 3             MVN             <NA>                 <NA>     NO
##
## $univariateNormality
##               Test      Variable Statistic   p value Normality
## 1 Anderson-Darling   condition     16.0872  <0.001      NO
## 2 Anderson-Darling     grade        4.3770  <0.001      NO
## 3 Anderson-Darling sqft_living15    1.1839  0.0041      NO
## 4 Anderson-Darling  sqft_lot15     20.0313  <0.001      NO
##
## $Descriptives
##                n         Mean      Std.Dev Median  Min    Max 25th 75th
## condition     97     3.402062 6.399836e-01      3    3      5    3    4
## grade         97     8.123711 9.922379e-01      8    6     10    7    9
```

```
## sqft_living15 97  2244.226804 6.494546e+02   2140 1220   3940 1760 2710
## sqft_lot15    97 15514.855670 2.877538e+04   7393 1206 217800 5074 9998
##                      Skew   Kurtosis
## condition     1.3084545  0.4920722
## grade         0.1977490 -0.7662365
## sqft_living15 0.5076553 -0.6360857
## sqft_lot15    4.6123444 25.3078766
```

Debido a que los valores p de las pruebas de mardia de normalidad multivariada son menores al α de 0.05, se concluye que los datos de la categoria medium no tienen una distribución normal.

High

```
par(mfrow=c(2,2))
plot(density(high$condition),main = "Density Plot of Condition",xlab =' Condition')
plot(density(high$grade),main = "Density Plot of Grade",xlab =' Grade')
plot(density(high$sqft_living15),main = "Density Plot of Sqft_living15",xlab =' Sqft_living15')
plot(density(high$sqft_lot15),main = "Density Plot of Petal Width",xlab =' Sqft_lot15')
```

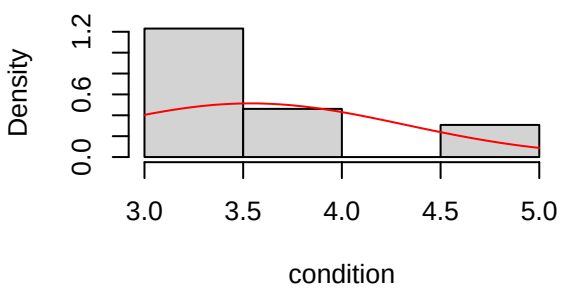


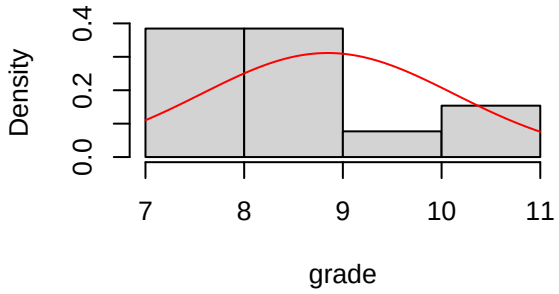

```
head(high)
```

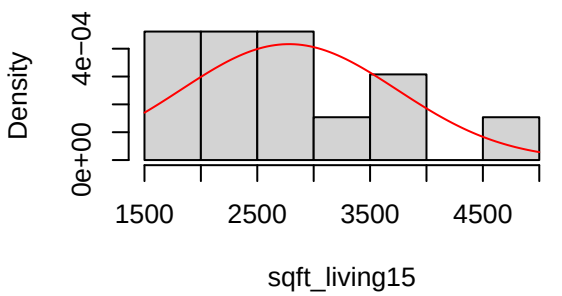
```
##       condition grade sqft_living15 sqft_lot15 Category
## 3453          3    11          2440       1229     high
```

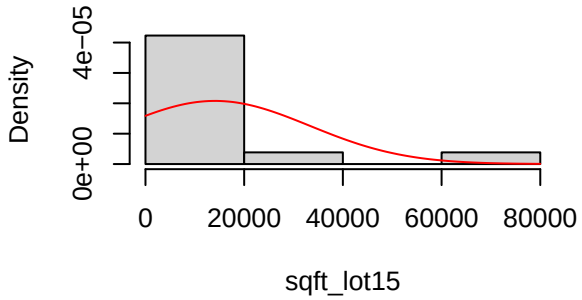
```
## 8178          4     7          1910       4590     high
## 11269         3     9          2860       6360     high
## 8048          3     9          2000       4380     high
## 3548          3     8          2080       3570     high
## 21249         3     8          2930      21569     high
```

```
highhist = mvn(data=high[,1:4],mvnTest = "royston",univariatePlot = "histogram")
```

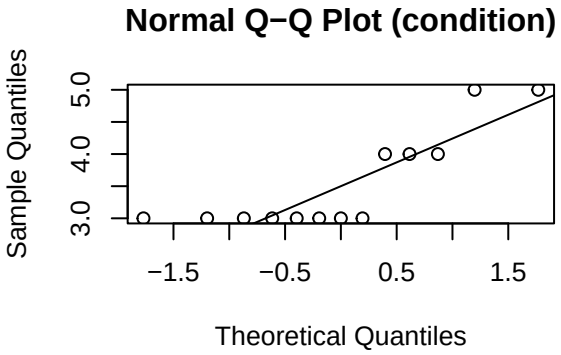


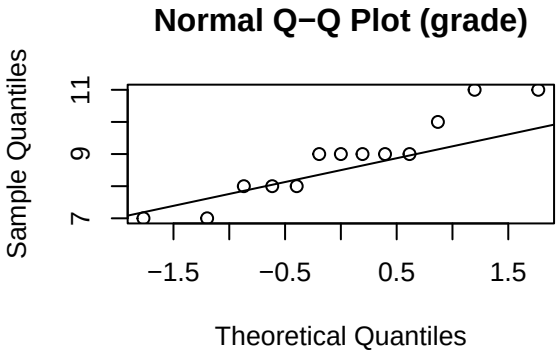


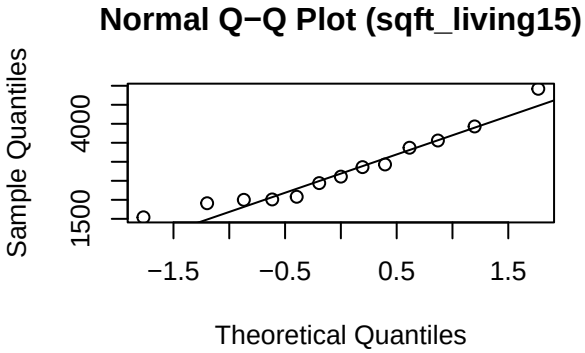


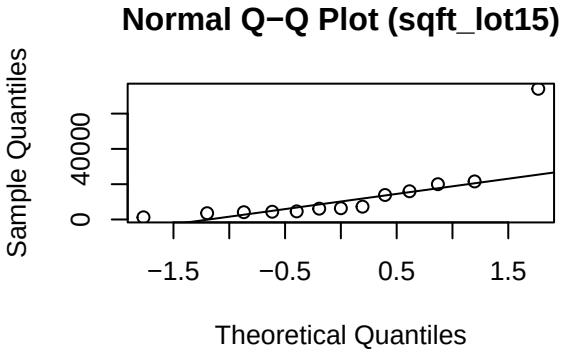



```
highqq = mvn(data = high[,1:4],mvnTest = "royston",univariatePlot = "qqplot")
```









Pruebas de normalidad

H_0 : Los datos tienen una distribución normal.

H_1 : Los datos no tienen una distribución normal.

```
mvn(data=high[,1:4],mvnTest = "mardia")
```

```
## $multivariateNormality
##              Test          Statistic            p value Result
## 1 Mardia Skewness   34.6491996656417 0.0220542093734105     NO
## 2 Mardia Kurtosis -0.244775765992987  0.806630059134789    YES
## 3             MVN               <NA>               <NA>     NO
##
## $univariateNormality
##               Test      Variable Statistic   p value Normality
## 1 Anderson-Darling   condition      1.7311   1e-04      NO
## 2 Anderson-Darling     grade        0.5130  0.1573      YES
## 3 Anderson-Darling sqft_living15    0.3488  0.4176      YES
## 4 Anderson-Darling  sqft_lot15      1.9007  <0.001      NO
##
## $Descriptives
##                n         Mean      Std.Dev Median  Min   Max 25th  75th
## condition     13     3.538462     0.776250      3    3     5    3     4
## grade         13     8.846154     1.281025      9    7    11    8     9
```

```
## sqft_living15 13  2781.538462   958.200102   2610 1540  4920 2010  3370
## sqft_lot15    13 14086.692308 19191.309745   6360 1229 74052 4380 15993
##                      Skew   Kurtosis
## condition     0.8699666 -0.8952805
## grade         0.2611224 -1.0122677
## sqft_living15 0.7056710 -0.5292406
## sqft_lot15    2.2737206  4.3779339
```

Debido a que uno de los valores p de las pruebas de mardia de normalidad multivariada es menor al α de 0.05, se concluye que los datos de la categoria high no tienen una distribución normal.

Prueba de homogeneidad de matrices de covarianza

Ho: Var-cov M low = var cov M medium = var-cov M high

Ha: al menos dos matrices no son iguales

```
library(heplots)
```

```
## Loading required package: broom
```

```
library(car)
```

```
## Loading required package: carData
```

```
library(carData)
library(broom)

boxM(M_nueva[,-5],M_nueva[,5],conf.level=0.99)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  M_nueva[, -5]
## Chi-Sq (approx.) = 265.46, df = 20, p-value < 2.2e-16
```

Debido a que el valor p es muy bajo, se tiene suficiente evidencia para rechazar nuestra hipotesis nula, por lo que se cncluye que al menos dos de las matrices de covarianza no son iguales.

Como no se encontró normalidad en los datos de las categorías, los siguientes pasos serían hacer una transformación de los datos buscando esta normalidad, ya sea con técnicas como box'cox o Yeo-Johnson.