Actividad 2_5

Ricardo kaleb Flores Alfonso

2024-10-04

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

```
library(caret)
```

```
## Cargando paquete requerido: ggplot2
```

```
## Cargando paquete requerido: lattice
```

```
library(MASS)

M=read.csv("kc_house_data.csv")
head(M)
```

```
##           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
```

```
str(M)
```

```
## '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 ...
```

```
#Remover observaciones con precio mayor a  $1.5M
M <- subset(M, price <= 1500000)

#Agregar una nueva variable categórica: Category
M$Category <- factor(ifelse(M$price < 500000, "low", ifelse(M$price < 1000000, "medium","high")))

#Estructura de los datos
str(M)
```

```
## '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 ...
```

```
# Nombres de columnas del data set
all_cols <- names(M)
```

```r
#Crear un data frame para la columna categoría
Category <- data.frame(M[,22])

#Se eliminan las primeras tres columnas (ID, Date, y Category)
M <-M[,-c(1:3,22)]

# Identificar variables con varianza cercana a cero: remove_cols
remove_cols <- nearZeroVar(M)

# Remover variables con varianza cercana a cero
M2 <- M[,-remove_cols]


#Agregar la variable categoría al data frame
M2$Category <- Category[,1]
```

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

```r
set.seed(42)
sampled_data <- M2[sample(nrow(M2), 300), ]
```

3. Gráfico de la segmentación original de los datos

¿Qué variable o variables discriminan mejor?

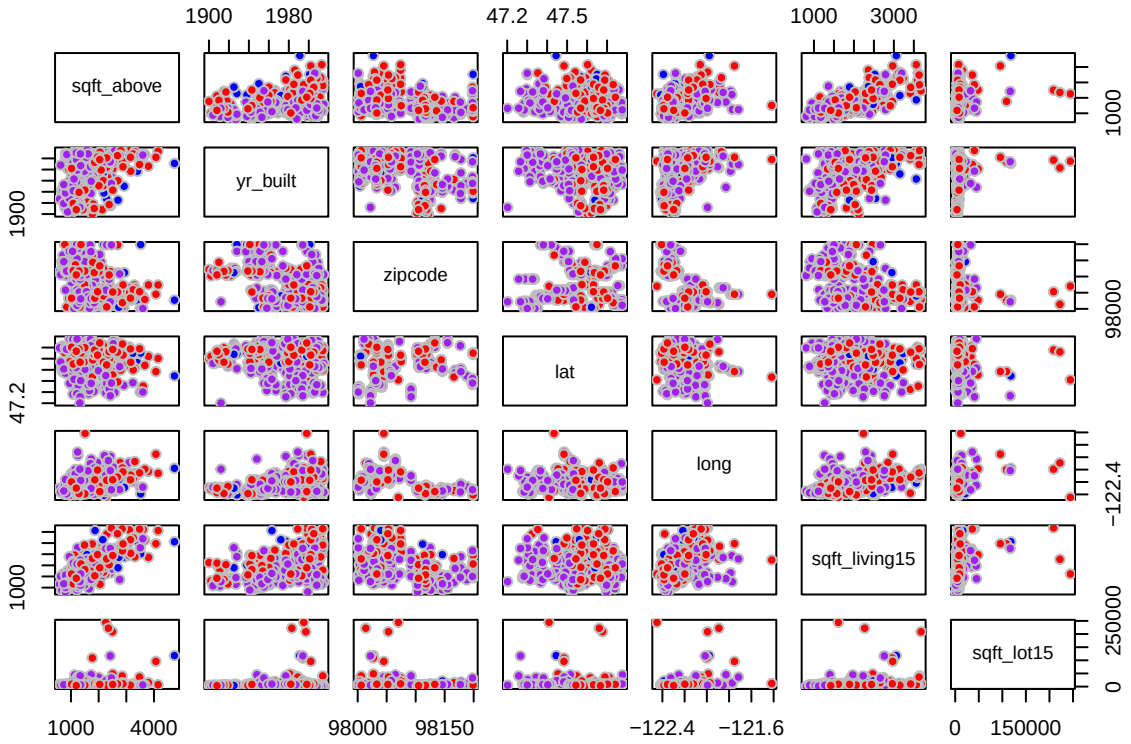
```r
#Asignamos un color a cada categoria
color = c(high="blue", medium ="purple", low = "red")
color
```

```
##     high   medium      low
##   "blue" "purple"    "red"
```

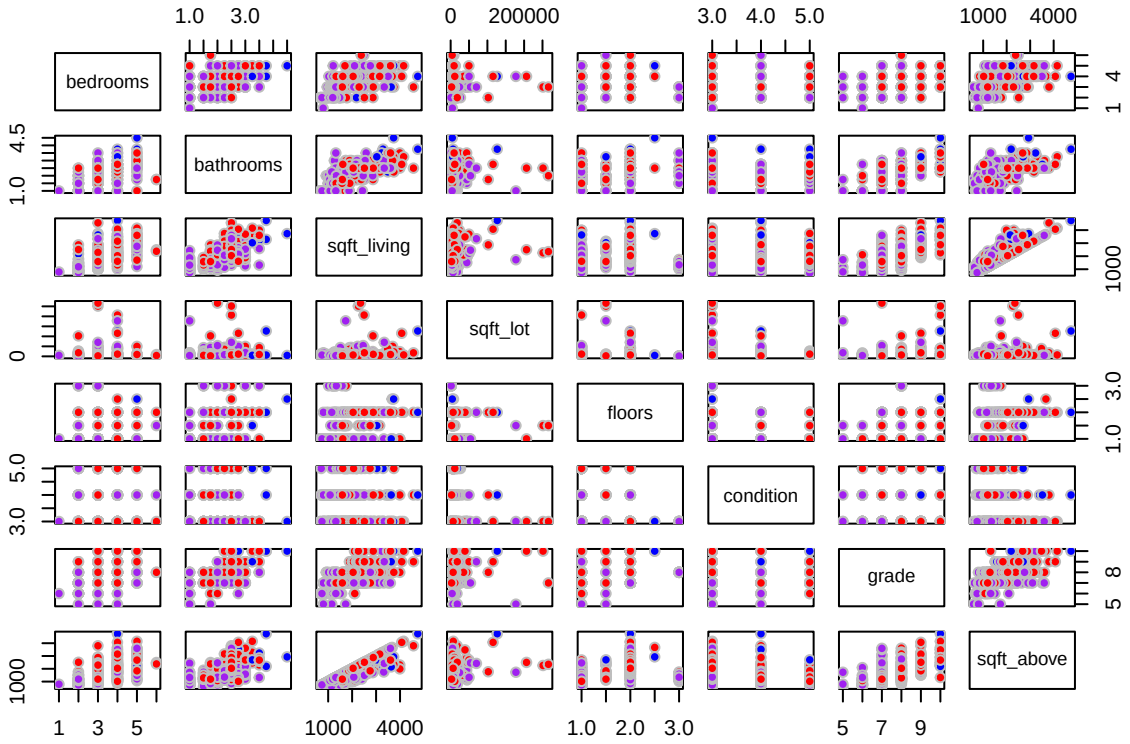
```r
#Creamos un vector con el color corresponidente a cada observacion de acuerdo a la columna categoria
col.ind = color[sampled_data$Category]
head(col.ind)
```

```
##   medium   medium   medium      low      low     high
## "purple" "purple" "purple"    "red"    "red"   "blue"
```

```r
#Graficos de dispersion con el color de acuerdo a la categoria
plot(sampled_data[,c(8:14)], pch=21, bg=col.ind, col = "gray")
```




```
plot(sampled_data[,c(1:8)], pch=21, bg=col.ind, col = "gray")
```



Las variables que mejor discriminan son las de condición, medidas del la sala, la medida del techo, la cantidad de baños y habitaciones.

4. Realiza un análisis discriminante para responder las siguientes preguntas

##a) Obtenga la media para cada variable predictora en función del grupo

```
aggregate(. ~ Category, data = sampled_data, FUN = mean)
```

```
##   Category bedrooms bathrooms sqft_living  sqft_lot   floors condition    grade
## 1     high 4.000000  2.942308    3277.846 17930.077 1.769231  4.000000 9.307692
## 2      low 3.115385  1.864011    1655.643  9919.374 1.395604  3.384615 7.120879
## 3   medium 3.561905  2.292857    2381.952 18127.524 1.609524  3.447619 8.114286
##   sqft_above yr_built  zipcode      lat      long sqft_living15 sqft_lot15
## 1   2787.846 1969.462 98089.38 47.59053 -122.2435      2707.692  17787.000
## 2   1456.907 1973.769 98080.74 47.53491 -122.2208      1691.231   8612.533
## 3   2014.762 1970.438 98082.90 47.60793 -122.2055      2243.857  16651.886
```

Vemos que la que tiene cada categoria varia mucho en el tamaño de la sala, del espacio para aparcar, del espacio hacia el techo, asi como el rating.

b) Muestre las probabilidades a priori para las diferentes clases, es decir, la distribución de datos en función de la variable dependiente

```
prop.table(table(sampled_data$Category))
```

```
##
##       high        low     medium
## 0.04333333 0.60666667 0.35000000
```

Para esta muestra de datos, se tiene una alta cantidad de casos de viviendas baratas y medianas, sin embargo de viviendas caras existen muy pocas en la muestra. ## c) Determine y escriba la(s) funcion(es) lineal(es) discriminante(s).

```
lda.model = lda(Category~., data=sampled_data)
lda.model
```

```
## Call:
## lda(Category ~ ., data = sampled_data)
##
## Prior probabilities of groups:
##       high        low     medium
## 0.04333333 0.60666667 0.35000000
##
## Group means:
##        bedrooms bathrooms sqft_living  sqft_lot   floors condition    grade
## high   4.000000  2.942308    3277.846 17930.077 1.769231  4.000000 9.307692
## low    3.115385  1.864011    1655.643  9919.374 1.395604  3.384615 7.120879
## medium 3.561905  2.292857    2381.952 18127.524 1.609524  3.447619 8.114286
##        sqft_above yr_built  zipcode      lat      long sqft_living15 sqft_lot15
## high     2787.846 1969.462 98089.38 47.59053 -122.2435      2707.692  17787.000
## low      1456.907 1973.769 98080.74 47.53491 -122.2208      1691.231   8612.533
## medium   2014.762 1970.438 98082.90 47.60793 -122.2055      2243.857  16651.886
##
## Coefficients of linear discriminants:
##                          LD1           LD2
## bedrooms       8.764129e-02  4.690601e-01
## bathrooms     -3.678549e-01 -6.243953e-01
## sqft_living   -2.767951e-04  6.172273e-04
## sqft_lot      -6.154824e-06 -2.034430e-06
## floors        -3.667238e-01  9.449598e-01
## condition     -3.615611e-01 -8.024377e-01
## grade         -7.478038e-01 -2.671797e-01
## sqft_above    -2.141892e-04 -1.746022e-03
## yr_built       2.440716e-02 -8.572032e-03
## zipcode       -9.092161e-04  9.096379e-04
## lat           -2.202360e+00  3.440434e+00
## long           7.460542e-01  2.539110e+00
## sqft_living15 -6.396115e-04  1.112111e-03
## sqft_lot15     3.860747e-06  1.327515e-05
##
## Proportion of trace:
##    LD1    LD2
## 0.9492 0.0508
```

$$LD1 = 0.0876 \cdot bedrooms - 0.3678 \cdot bathrooms - 0.0003 \cdot sqft_living - 0.00001 \cdot sqft_lot - 0.3667 \cdot floors - 0.3615 \cdot condition - 0.7478 \cdot g$$

$$LD2 = 0.4690 \cdot bedrooms - 0.6243 \cdot bathrooms + 0.0006 \cdot sqft_living - 0.00001 \cdot sqft_lot + 0.9450 \cdot floors - 0.8024 \cdot condition - 0.2671 \cdot g$$

```r
predicted = predict(lda.model)

names(predicted)
```

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

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

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

```r
#Se dan las probabilidades a posteriori de acuerdo a la clase a la que podría pertenecer
head(predicted$posterior)
```

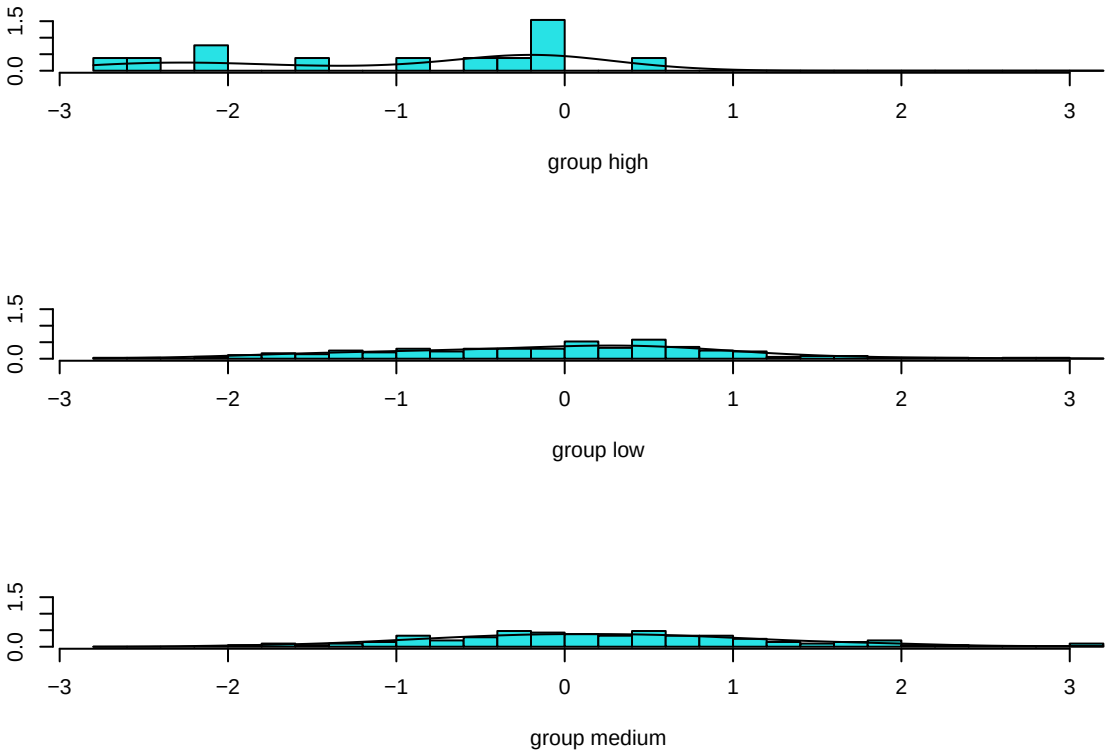
```
##                high          low     medium
## 19192 1.216231e-04 8.334882e-01 0.16639017
## 9497  3.023828e-06 9.208096e-01 0.07918742
## 1276  3.586538e-06 9.642340e-01 0.03576245
## 15869 3.841238e-01 1.527149e-02 0.60060475
## 9024  9.666660e-03 3.198872e-03 0.98713447
## 10512 9.059742e-01 2.786284e-05 0.09399791
```

```r
#valores discriminantes lineales
head(predicted$x)
```

```
##              LD1        LD2
## 19192  0.3743874 -0.3503483
## 9497   1.0427038  0.9336078
## 1276   1.2804853 -0.2636748
## 15869 -2.5289734 -0.9641605
## 9024  -2.9122935  3.1178954
## 10512 -4.6465804  0.4164552
```

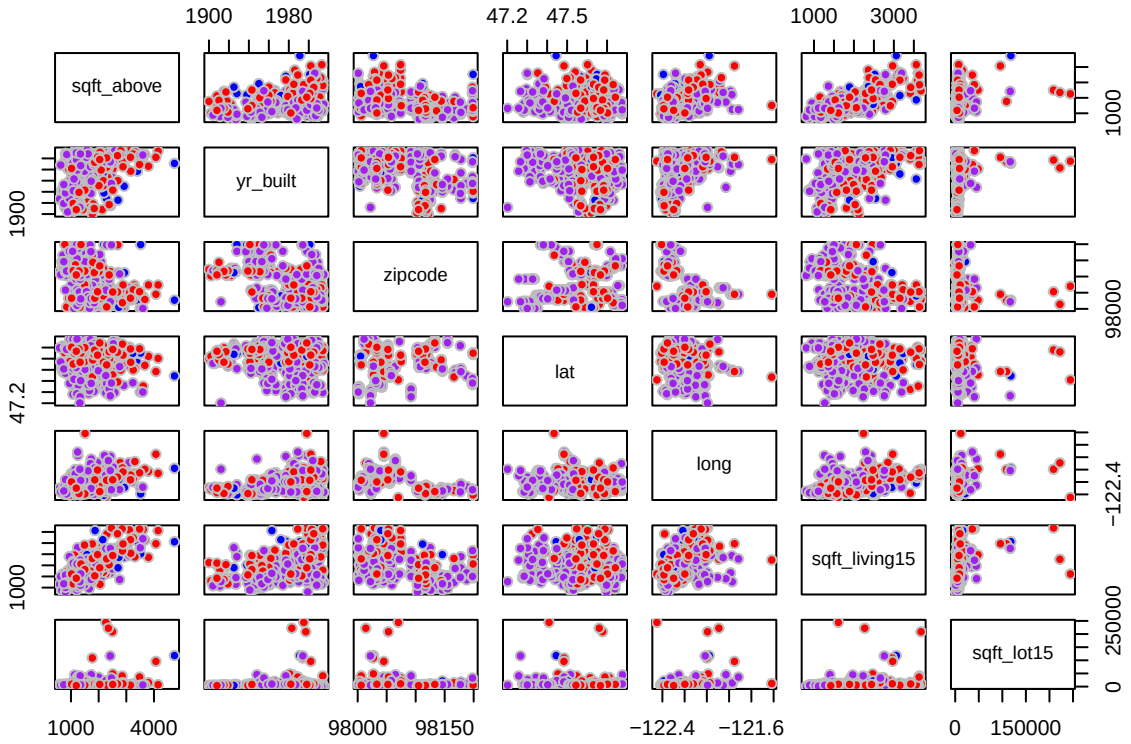
d) Grafique el histograma de valores discriminantes en cada grupo.

```r
#ldahist(data=predicted$x[,1],g=sampled_data$Category,type="both",main="Histograma de la función discri
ldahist(data=predicted$x[,2],g=sampled_data$Category,type="both",main="Histograma de la función discrim
```

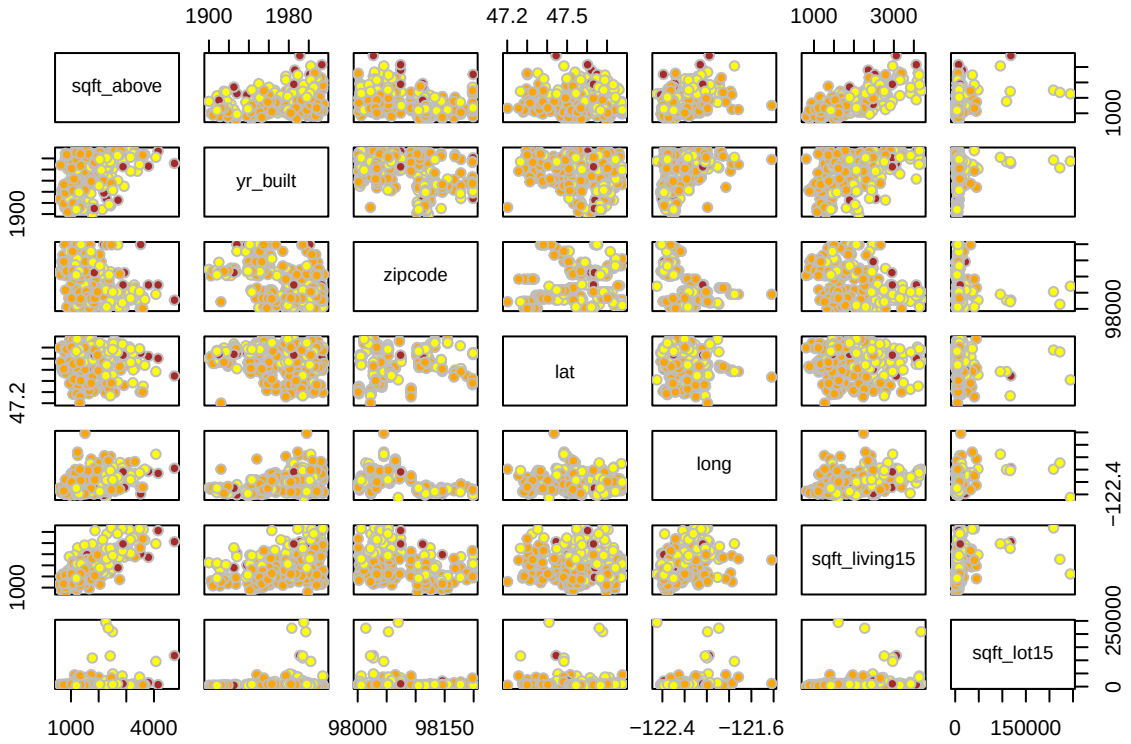


e) Muestre gráficamente la segmentación de los datos. Realiza el gráfico de dispersión con las predicciones hechas por el modelo.

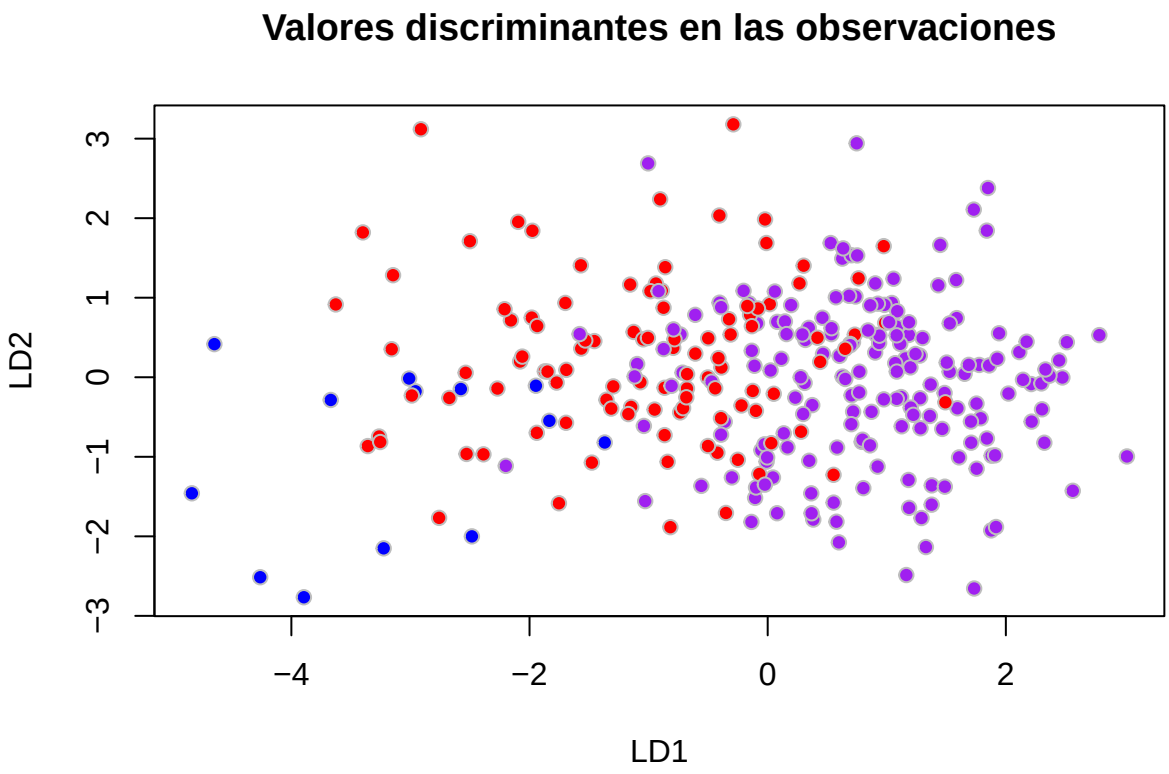
```
#Asignamos un color a cada especie
color2=c(high="brown",medium="orange",low="yellow")

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

#Graficos de dispersion con el color de acuerdo al tipo de especie
#Graficos de dispersion con el color de acuerdo a la categoria
plot(sampled_data[,c(8:14)], pch=21, bg=col.ind, col = "gray")
```




```
plot(sampled_data[,c(8:14)], pch=21, bg=col.ind2, col = "gray")
```




```
plot(LD2~LD1, data=predicted$x, pch=21,col="gray",bg=col.ind,main="Valores discriminantes en las observa
```



Vemos que ambos componentes principales logran discriminar de una buena manera las tres categorias sin embargo si existe una mezcla importante entre las clases bajas y medias.

f) Evalúe la precisión del modelo. ¿El modelo es bueno para pronosticar? Indique el porcentaje de predicciones erróneas y la tabla de contingencia.

```
table(pred=predicted$class, true=sampled_data$Category)
```

```
##         true
## pred     high low medium
##   high      7   0      4
##   low       0 165     33
##   medium    6  17     68
```

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


cat("\n El modelo tiene un porcentaje de error de: ",rate*100,"%")
```

```
##
##  El modelo tiene un porcentaje de error de:  20 %
```

El modelo es bueno para pronosticar a un 20%, a pesar de que esto es un porcentaje alto, para el caso de las casas yo considero que puede ser utilizado dependiendo lo que se busque categorizar.

5) Valide los supuestos del modelo

```
library(heplots)
```

```
## Cargando paquete requerido: broom
```

```
library(MVN)


mvn(sampled_data[, c(1:14)])
```

```
## $multivariateNormality
##            Test      HZ p value MVN
## 1 Henze-Zirkler 1.42539       0  NO
##
## $univariateNormality
##                Test      Variable Statistic   p value Normality
## 1  Anderson-Darling   bedrooms      17.9044  <0.001      NO
## 2  Anderson-Darling   bathrooms      6.5877  <0.001      NO
## 3  Anderson-Darling  sqft_living     3.0728  <0.001      NO
## 4  Anderson-Darling   sqft_lot      67.8810  <0.001      NO
## 5  Anderson-Darling    floors       35.8379  <0.001      NO
## 6  Anderson-Darling   condition     48.9457  <0.001      NO
## 7  Anderson-Darling     grade       20.0644  <0.001      NO
## 8  Anderson-Darling  sqft_above      7.1202  <0.001      NO
## 9  Anderson-Darling   yr_built       3.9199  <0.001      NO
## 10 Anderson-Darling    zipcode       5.9280  <0.001      NO
## 11 Anderson-Darling      lat         4.1250  <0.001      NO
## 12 Anderson-Darling     long         6.5784  <0.001      NO
## 13 Anderson-Darling sqft_living15    4.8355  <0.001      NO
## 14 Anderson-Darling  sqft_lot15     70.8617  <0.001      NO
##
## $Descriptives
##                  n         Mean      Std.Dev      Median       Min         Max
## bedrooms       300     3.310000 8.581110e-01     3.00000     1.000      6.0000
## bathrooms      300     2.060833 6.835435e-01     2.12500     1.000      4.5000
## sqft_living    300  1980.146667 7.971343e+02  1880.00000   670.000   4740.0000
## sqft_lot       300 13139.356667 2.870241e+04  7038.00000   711.000 266151.0000
## floors         300     1.486667 5.545394e-01     1.00000     1.000      3.0000
## condition      300     3.433333 6.785452e-01     3.00000     3.000      5.0000
## grade          300     7.563333 1.011302e+00     7.00000     5.000     10.0000
## sqft_above     300  1709.830000 7.550965e+02  1495.00000   590.000   4740.0000
## yr_built       300  1972.416667 2.920135e+01  1977.00000  1900.000   2015.0000
## zipcode        300 98081.873333 5.401577e+01 98073.00000 98001.000  98199.0000
## lat            300    47.562878 1.362122e-01    47.56855    47.202     47.7769
## long           300  -122.216440 1.521223e-01  -122.25600  -122.449   -121.4170
## sqft_living15  300  1928.696667 6.355784e+02  1780.00000   806.000   3680.0000
## sqft_lot15     300 11823.866667 2.550011e+04  7202.50000   748.000 244372.0000
##                       25th       75th       Skew    Kurtosis
## bedrooms           3.00000     4.0000  0.3445023 -0.03500722
## bathrooms          1.50000     2.5000  0.0317932 -0.30976640
## sqft_living     1340.00000  2382.5000  0.7444143  0.25993400
## sqft_lot        4800.00000  9699.2500  6.3598589 45.21646525
## floors             1.00000     2.0000  0.6680909 -0.48938449
```

```
## condition         3.00000     4.0000  1.2656347  0.25024184
## grade             7.00000     8.0000  0.6399637  0.25398596
## sqft_above     1150.00000  2190.0000  1.0290753  0.77075114
## yr_built       1951.00000  1998.0000 -0.4629483 -0.70205948
## zipcode       98034.00000 98118.0000  0.4185979 -0.77444536
## lat              47.48445    47.6836 -0.4686872 -0.75375074
## long           -122.33825  -122.1255  1.2381962  2.34991395
## sqft_living15  1460.00000  2352.5000  0.7287091 -0.09824494
## sqft_lot15     4971.00000  9073.5000  6.7435979 50.42959494
```

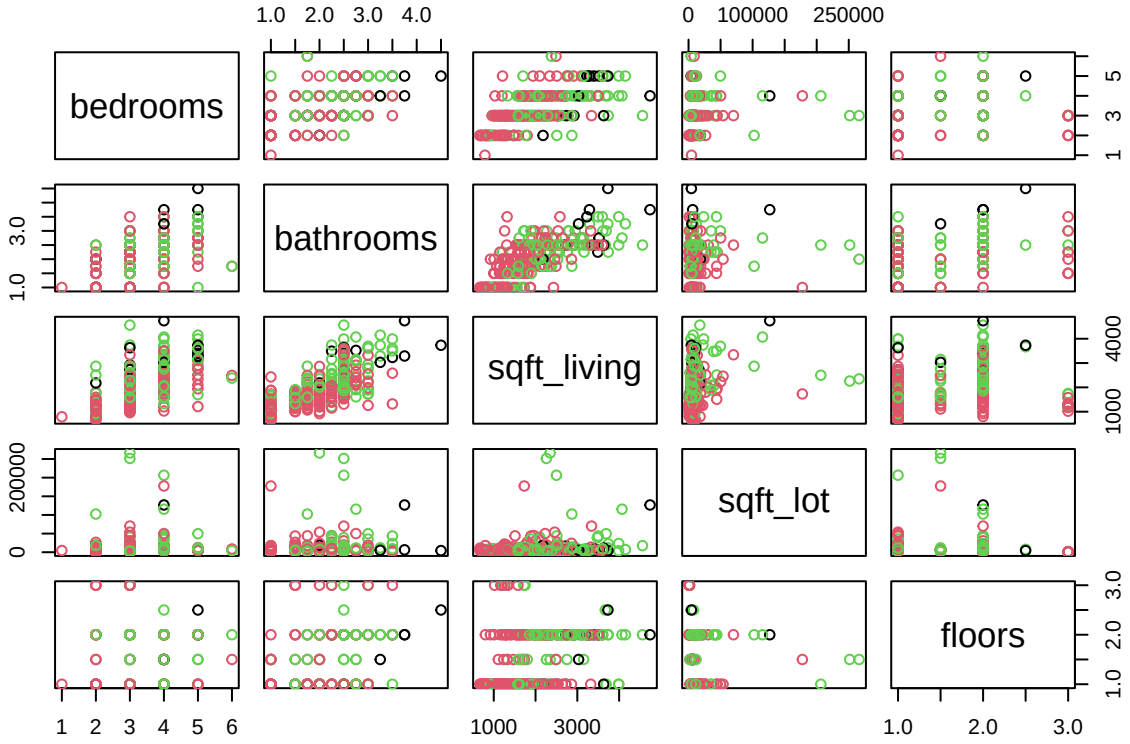
```
# Homocedasticidad
boxM(sampled_data[, c(1:14)], sampled_data$Category)
```

```
## Warning in boxM.default(sampled_data[, c(1:14)], sampled_data$Category): there
## are one or more levels with less observations than variables!
```

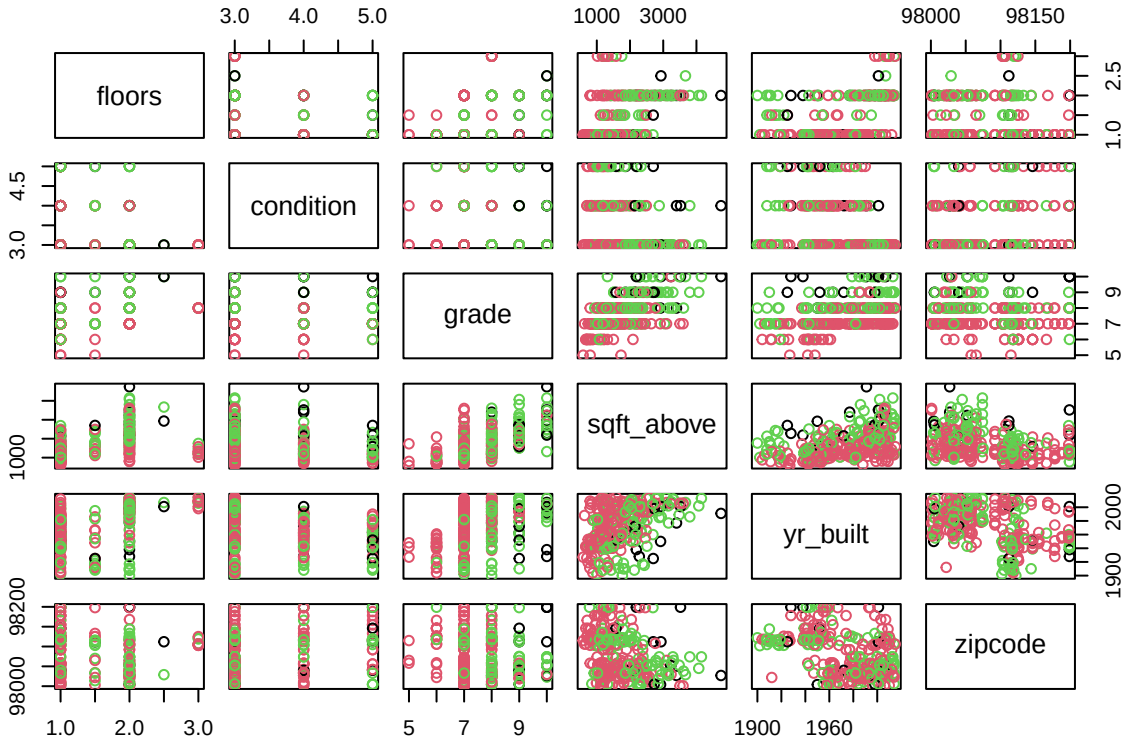
```
## Warning in log(unlist(lapply(mats, det))): Se han producido NaNs
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  sampled_data[, c(1:14)]
## Chi-Sq (approx.) = NaN, df = 210, p-value = NA
```

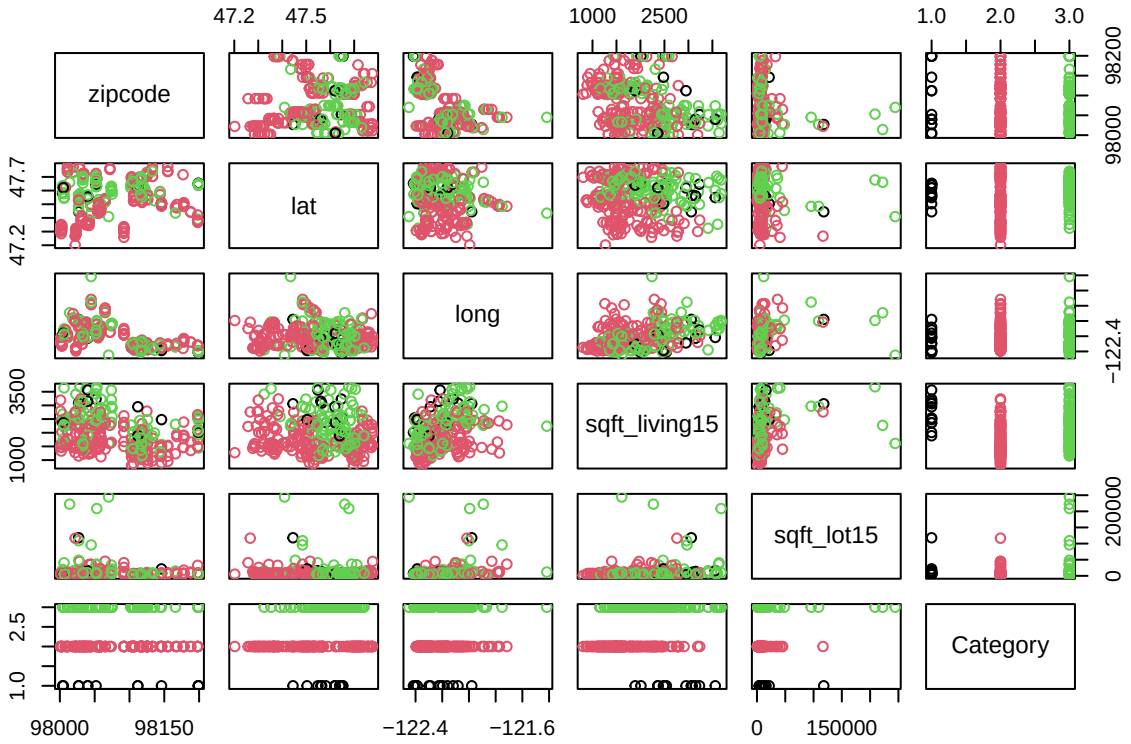
```
# Gráficos de dispersión para cada par de variables por grupo
pairs(sampled_data[, c(1:5)], col=sampled_data$Category)
```




```
pairs(sampled_data[, c(5:10)], col=sampled_data$Category)
```




```
pairs(sampled_data[, c(10:15)], col=sampled_data$Category)
```




```
# Prueba de multicolinealidad
library(car)
```

```
## Cargando paquete requerido: carData
```

```
vif_model <- lm(Category ~ bedrooms + bathrooms + sqft_living + sqft_lot + floors + condition + grade +
```

```
## Warning in model.response(mf, "numeric"): using type = "numeric" with a factor
## response will be ignored
```

```
## Warning in Ops.factor(y, z$residuals): '-' no es significativo para factores
```

```
vif(vif_model)
```

```
## Warning in Ops.factor(r, 2): '^' no es significativo para factores
```

```
## Warning in cov2cor(v): diag(V) had non-positive or NA entries; the non-finite
## result may be dubious
```

```
##      bedrooms     bathrooms   sqft_living      sqft_lot        floors
##           NaN           NaN           NaN           NaN           NaN
##     condition         grade    sqft_above      yr_built       zipcode
##           NaN           NaN           NaN           NaN           NaN
##           lat          long sqft_living15    sqft_lot15
##           NaN           NaN           NaN           NaN
```

El modelo obtenido no pasa los supuestos de normalidad multivariada, pues tiene un valor de p menor a 0.05, tampoco las variables por separado cumplen los supuestos de normalidad. El modelo no pasa el test de homocedasticidad, pues su valor de p es menor a 0.05. El modelo tampoco pasa los supuestos de multicolinealidad. Por esto se concluye que el modelo no es significativo para describir las variables.