PRACTICA 2: LIMPIEZA Y VALIDACIÓN DE LOS DATOS

Sinthia Elizabeth Guaigua Guanopatin

2022-06-03

---

1 Desarrollo

1.1 Descripción del dataset.

1.1.1 Dataset

El conjunto de datos objeto de análisis se ha obtenido a partir de este enlace en Kaggle, del siguiente link: https://www.kaggle.com/datasets/syuzai/perth-house-prices (https://www.kaggle.com/datasets/syuzai/perth-house-prices)

El fichero tiene formato csv, contiene información de viviendas de venta.

Los campos de este conjunto de datos, encontramos los siguientes:

address : Dirección física de la propiedad

suburb :localidad específica en Perth;

price : Precio al que se vendió una propiedad (AUD)

bedrooms : Número de dormitorios

bathrooms : Número de baños

garage :Número de plazas de garaje

land_area : Superficie total del terreno (m^2)

floor_area : Superficie interior (m^2)

buil_year : Año en que se construyó la propiedad

CBD_dist : Distancia desde el centro de Perth (m)

nearest_stn : La estación de transporte público más cercana a la propiedad

nearest_stn_dst : La distancia de la estación más cercana (m)

date_sold : Mes y año en que se vendió la propiedad

NEAREST_SCH_RANK: Ranking de cercanía a la escuela

¿Por qué es importante?

Se ha seleccionado este proyecto y set de datos pues es similar al estudio que actualmente me encuentro realizando, debido a que me dedico a medio tiempo a la venta de casas.

El objetivo es:

- Predecir precios justos y competitivos de viviendas de venta.
- Identificar que variables influyen para poder predecir el valor de una vivienda

1.2 Integración y selección de los datos de interés a analizar.

Carga de Librerías:

```
if(!require(ggplot2)){
    install.packages('ggplot2', repos='http://cran.us.r-project.org')
    library(ggplot2)
}
```

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

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
if(!require(ggpubr)){
    install.packages('ggpubr', repos='http://cran.us.r-project.org')
    library(ggpubr)
}
```

```
## Loading required package: ggpubr
```

```
## Warning: package 'ggpubr' was built under R version 4.1.3
```

```
if(!require(grid)){
    install.packages('grid', repos='http://cran.us.r-project.org')
    library(grid)
}
```

```
## Loading required package: grid
```

```
if(!require(gridExtra)){
    install.packages('gridExtra', repos='http://cran.us.r-project.org')
    library(gridExtra)
}
```

```
## Loading required package: gridExtra
```

```
## Warning: package 'gridExtra' was built under R version 4.1.3
```

```
if(!require(C50)){
    install.packages('C50', repos='http://cran.us.r-project.org')
    library(C50)
}
```

```
## Loading required package: C50
```

```
## Warning: package 'C50' was built under R version 4.1.3
```

```
if (!require('papeR')) {install.packages('papeR'); library('papeR')
}
```

```
## Loading required package: papeR
```

```
## Warning: package 'papeR' was built under R version 4.1.3
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 4.1.3
```

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

```
## Warning: package 'carData' was built under R version 4.1.3
```

```
## Loading required package: xtable
```

```
## Warning: package 'xtable' was built under R version 4.1.3
```

```
## Registered S3 method overwritten by 'papeR':
##   method    from
##   Anova.lme car
```

```
##
## Attaching package: 'papeR'
```

```
## The following object is masked from 'package:utils':
##
##     toLatex
```

```
if (!require('corrplot')) {install.packages('corrplot'); library('corrplot')
}
```

```
## Loading required package: corrplot
```

```
## Warning: package 'corrplot' was built under R version 4.1.3
```

```
## corrplot 0.92 loaded
```

```
if(!require(gmodels)){
    install.packages('gmodels', repos='http://cran.us.r-project.org')
    library(gmodels)
}
```

```
## Loading required package: gmodels
```

```
## Warning: package 'gmodels' was built under R version 4.1.3
```

```
if (!require('dplyr')) {install.packages('dplyr'); library('dplyr')
}
```

```
## Loading required package: dplyr
```

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

```
## The following objects are masked from 'package:papeR':
##
##     summarise, summarize
```

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

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

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

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

```
if (!require('stringr')) {install.packages('stringr'); library('stringr')
}
```

```
## Loading required package: stringr
```

```
if (!require('nortest')) {install.packages('nortest'); library('nortest')
}
```

```
## Loading required package: nortest
```

```
if (!require('scales')) {install.packages('scales'); library('scales')
}
```

```
## Loading required package: scales
```

```
## Warning: package 'scales' was built under R version 4.1.3
```

```
if (!require('broom')) {install.packages('broom'); library('broom')
}
```

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

```
## Warning: package 'broom' was built under R version 4.1.3
```

Carga de los datos:

```
data_set<-read.csv("./data.csv",header=T,sep=",")
attach(data_set)
```

1.3 Limpieza de los datos.

Para empezar, calculamos las dimensiones de la base de datos mediante la función str(). Obtenemos que disponemos de 33656 registros (filas) y 19 variables (columnas).

Se procede a eliminar los atributos que no aportan a nuestro estudio, entre estos están atributos con valores tipo texto imposibles de categorizar(NEAREST_STN,ADDRESS,NEAREST_SCH,DATE_SOLD) y atributos no necesarios(LONGITUDE, LATITUDE,POSTCODE).

```
data_set[c("NEAREST_STN","ADDRESS","NEAREST_SCH","DATE_SOLD","LONGITUDE","LATITUDE", "POSTCOD
E")] <- NULL
```

Se puede observar que existen variables con diferente tipo de dato, por ejemplo garage ,buil_year deberían ser del tipo numérico.

```
str(data_set)
```

```
## 'data.frame':    33656 obs. of  12 variables:
##  $ SUBURB          : chr  "South Lake" "Wandi" "Camillo" "Bellevue" ...
##  $ PRICE           : int  565000 365000 287000 255000 325000 409000 400000 370000 565000 6
85000 ...
##  $ BEDROOMS        : int  4 3 3 2 4 4 3 4 4 3 ...
##  $ BATHROOMS       : int  2 2 1 1 1 2 2 2 2 2 ...
##  $ GARAGE          : chr  "2" "2" "1" "2" ...
##  $ LAND_AREA       : int  600 351 719 651 466 759 386 468 875 552 ...
##  $ FLOOR_AREA      : int  160 139 86 59 131 118 132 158 168 126 ...
##  $ BUILD_YEAR      : chr  "2003" "2013" "1979" "1953" ...
##  $ CBD_DIST        : int  18300 26900 22600 17900 11200 27300 28200 41700 12100 5900 ...
##  $ NEAREST_STN_DIST: int  1800 4900 1900 3600 2000 1000 3700 1100 2500 508 ...
##  $ NEAREST_SCH_DIST: num  0.828 5.524 1.649 1.571 1.515 ...
##  $ NEAREST_SCH_RANK: int  NA 129 113 NA NA NA NA NA NA 29 ...
```

Remplazamos los valores en letras NULL por NA y cambiamos el tipo de dato para las columnas buil_year, garage a numéricas

```
data_set[data_set == "NULL"]<- NA
data_set$BUILD_YEAR=as.numeric(data_set$BUILD_YEAR)
data_set$GARAGE=as.numeric(data_set$GARAGE)
data_set$LAND_AREA=as.numeric(data_set$LAND_AREA)
data_set$FLOOR_AREA=as.numeric(data_set$FLOOR_AREA)
data_set$PRICE=as.numeric(data_set$PRICE)
```

Se considerá crear las siguientes variables:

- Una variable sobre el Precio/M2 (superficie terreno total).

```
data_set$Precio_M2_total<- (data_set$PRICE/data_set$LAND_AREA)
```

Al crear las nuevas variables, se procede a eliminar las variables: LAND_AREA,PRICE, debido a que se estaría redundando la información.

```
data_set[c("LAND_AREA","BATHROOMS","BEDROOMS","GARAGE")] <- NULL
```

Se identifica que las variables NEAREST_STN_DIST,NEAREST_SCH_DIST,CBD_DIST no estan en la misma escala, por lo que se procede a cambiar las variables NEAREST_STN_DIST,CBD_DIST dividiendo para 1000

```
data_set$NEAREST_STN_DIST<- (data_set$NEAREST_STN_DIST/1000)
data_set$CBD_DIST<- (data_set$CBD_DIST/1000)
data_set$PRICE<- (data_set$PRICE/1000)
```

```
str(data_set)
```

```
## 'data.frame':    33656 obs. of  9 variables:
##  $ SUBURB          : chr  "South Lake" "Wandi" "Camillo" "Bellevue" ...
##  $ PRICE           : num  565 365 287 255 325 409 400 370 565 685 ...
##  $ FLOOR_AREA      : num  160 139 86 59 131 118 132 158 168 126 ...
##  $ BUILD_YEAR      : num  2003 2013 1979 1953 1998 ...
##  $ CBD_DIST        : num  18.3 26.9 22.6 17.9 11.2 27.3 28.2 41.7 12.1 5.9 ...
##  $ NEAREST_STN_DIST: num  1.8 4.9 1.9 3.6 2 1 3.7 1.1 2.5 0.508 ...
##  $ NEAREST_SCH_DIST: num  0.828 5.524 1.649 1.571 1.515 ...
##  $ NEAREST_SCH_RANK: int  NA 129 113 NA NA NA NA NA NA 29 ...
##  $ Precio_M2_total : num  942 1040 399 392 697 ...
```

1.3.1 ¿Los datos contienen ceros o elementos vacíos?

Mostraremos para cada atributo la cantidad de valores perdidos mediante la función summary. Se identifica que si existen valores nulos para las columnas GARAGE, BUILD_YEAR y NEAREST_SCH_RANK. No existen valores vacios.

```
NAcount <- apply(is.na(data_set), 2, sum) >= 1
NAcount[NAcount==TRUE]
```

```
##       BUILD_YEAR NEAREST_SCH_RANK
##             TRUE             TRUE
```

```
WScount <- colSums(data_set == "")
WScount[WScount==TRUE]
```

```
## <NA> <NA>
##   NA   NA
```

Como parte de la preparación de los datos, miraremos si hay valores missing .

```
missing <- data_set[is.na(data_set),]
dim(missing)
```

```
## [1] 14107     9
```

Observamos fácilmente que si hay valores missing y, por tanto, deberemos preparar los datos en este sentido.

Imputación de valores para el atributo “BUILD_YEAR”

Reemplazamos los valores missing (na), por la media, porque si no tiene información nos inclinamos obtener un promedio de los años del resto de registros, pues colocar un valor 0 se convertiría en un outlier

```
data_set$BUILD_YEAR[is.na(data_set$BUILD_YEAR)]<-mean(data_set$BUILD_YEAR, na.rm = T)
data_set$BUILD_YEAR<-round(data_set$BUILD_YEAR, digits = 0)
```

Imputación de valores para el atributo “NEAREST_SCH_RANK” Se identifica que 10952, registros son nulos de un total 33656, lo cual corresponde a mas del 30% por ese motivo se procede a eliminar la columna de nuestro data_set

```
summary (data_set$NEAREST_SCH_RANK)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1.00   39.00   68.00   72.67  105.00  139.00   10952
```

```
data_set[c("NEAREST_SCH_RANK")] <- NULL
```

1.3.2 Identifica y gestiona los valores extremos.

Se muestra las estadísticas de las diferentes variables.

```
options(width = 90)
papeR::summarize(data_set)
```

```
## Factors are dropped from the summary
```

```
##                         N      Mean     SD       Min      Q1  Median      Q3     Max
## 1            PRICE 33656    637.07 355.83     51.00  410.00  535.50  760.00 2440.00
## 2       FLOOR_AREA 33656    183.50  72.10      1.00  130.00  172.00  222.50  870.00
## 3       BUILD_YEAR 33656   1989.73  19.96   1868.00 1980.00 1993.00 2004.00 2017.00
## 4         CBD_DIST 33656     19.78  11.36      0.68   11.20   17.50   26.60   59.80
## 5 NEAREST_STN_DIST 33656      4.52   4.50      0.05    1.80    3.20    5.30   35.50
## 6 NEAREST_SCH_DIST 33656      1.82   1.75      0.07    0.88    1.35    2.10   23.25
## 7  Precio_M2_total 33656    998.77 802.31      0.37  509.05  824.78 1248.09 9944.75
```

Se identifica que existen valores extremos sin embargo pueden ser valores reales los cuales dependan del lugar en el que se encuentren.

Valores atípicos Se realiza una representación en diagrama de cajas para las variables numéricas. Para la identificación de los valores atípico se hace uso de una gráfica tipo box-plot, en donde se observan los valores basados en cuartiles.

Variable CBD_DIST: Para aquellos valores atípicos identificados se reemplaza por el valor NaN, luego estos serán analizados para identificar si se eliminan o se realiza un proceso de regresión para colocar el valor.

```
valoresAtipicos <- boxplot.stats(data_set$CBD_DIST)$out
print("El tamaño de valores atípicos es:")
```

```
## [1] "El tamaño de valores atípicos es:"
```

```
length(valoresAtipicos)
```

```
## [1] 661
```

```
data_set$CBD_DIST[data_set$CBD_DIST %in% valoresAtipicos] <- NaN
```

Variable BUILD_YEAR: Para aquellos valores atípicos identificados se reemplaza por el valor NaN, luego estos serán analizados para identificar si se eliminan o se realiza un proceso de regresión para colocar el valor.

```
valoresAtipicos <- boxplot.stats(data_set$BUILD_YEAR)$out
print("El tamaño de valores atípicos es:")
```

```
## [1] "El tamaño de valores atípicos es:"
```

```
length(valoresAtipicos)
```

```
## [1] 1183
```

```
data_set$BUILD_YEAR[data_set$BUILD_YEAR %in% valoresAtipicos] <- NaN
```

Variable NEAREST_STN_DIST: Para aquellos valores atípicos identificados se reemplaza por el valor NaN, luego estos serán analizados para identificar si se eliminan o se realiza un proceso de regresión para colocar el valor.

```
valoresAtipicos <- boxplot.stats(data_set$NEAREST_STN_DIST)$out
print("El tamaño de valores atípicos es:")
```

```
## [1] "El tamaño de valores atípicos es:"
```

```
length(valoresAtipicos)
```

```
## [1] 3034
```

```
data_set$NEAREST_STN_DIST[data_set$NEAREST_STN_DIST %in% valoresAtipicos] <- NaN
```

Variable NEAREST_SCH_DIST: Para aquellos valores atípicos identificados se reemplaza por el valor NaN, luego estos serán analizados para identificar si se eliminan o se realiza un proceso de regresión para colocar el valor.

```
valoresAtipicos <- boxplot.stats(data_set$NEAREST_SCH_DIST)$out
print("El tamaño de valores atípicos es:")
```

```
## [1] "El tamaño de valores atípicos es:"
```

```
length(valoresAtipicos)
```

```
## [1] 2296
```

```
data_set$NEAREST_SCH_DIST[data_set$NEAREST_SCH_DIST %in% valoresAtipicos] <- NaN
```

Variable Precio_M2_total: Para aquellos valores atípicos identificados se reemplaza por el valor NaN, luego estos serán analizados para identificar si se eliminan o se realiza un proceso de regresión para colocar el valor.

```
valoresAtipicos <- boxplot.stats(data_set$Precio_M2_total)$out
print("El tamaño de valores atípicos es:")
```

```
## [1] "El tamaño de valores atípicos es:"
```

```
length(valoresAtipicos)
```

```
## [1] 2113
```

```
data_set$Precio_M2_total[data_set$Precio_M2_total %in% valoresAtipicos] <- NaN
```

Variable PRICE: Para aquellos valores atípicos identificados se reemplaza por el valor NaN, luego estos serán analizados para identificar si se eliminan o se realiza un proceso de regresión para colocar el valor.

```
valoresAtipicos <- boxplot.stats(data_set$PRICE)$out
print("El tamaño de valores atípicos es:")
```

```
## [1] "El tamaño de valores atípicos es:"
```

```
length(valoresAtipicos)
```

```
## [1] 2112
```

```
data_set$PRICE[data_set$PRICE %in% valoresAtipicos] <- NaN
```

Calculo de valores atipicos, se llega a la conclusión que estos registros estan mal registrados

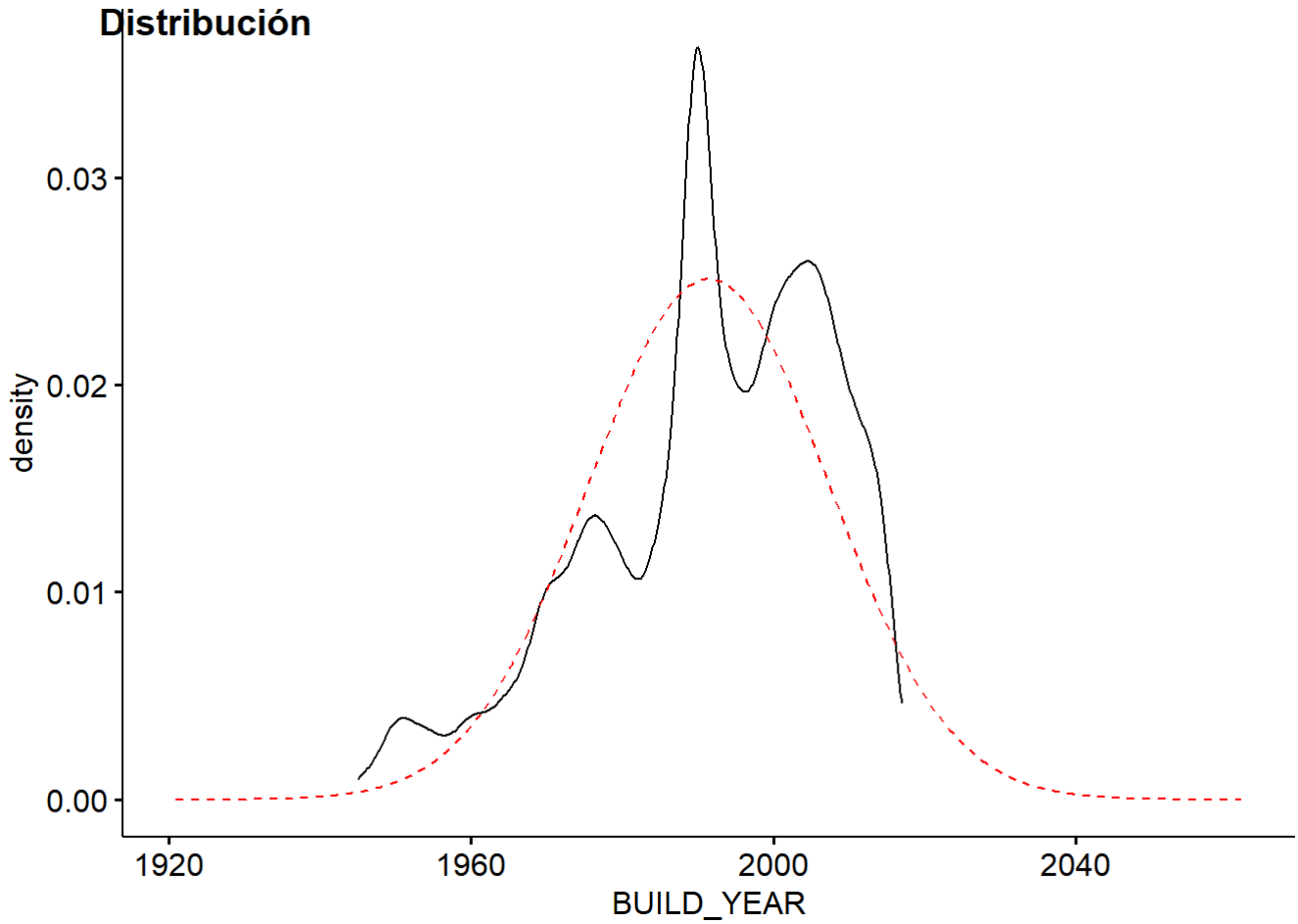
```
data_set <- data_set[!is.na(data_set$BUILD_YEAR),]
data_set <- data_set[!is.na(data_set$CBD_DIST),]
data_set <- data_set[!is.na(data_set$NEAREST_STN_DIST),]
data_set <- data_set[!is.na(data_set$NEAREST_SCH_DIST),]
data_set <- data_set[!is.na(data_set$Precio_M2_total),]
data_set <- data_set[!is.na(data_set$PRICE),]
```

1.4 Análisis de los datos.

Métodos de discretización

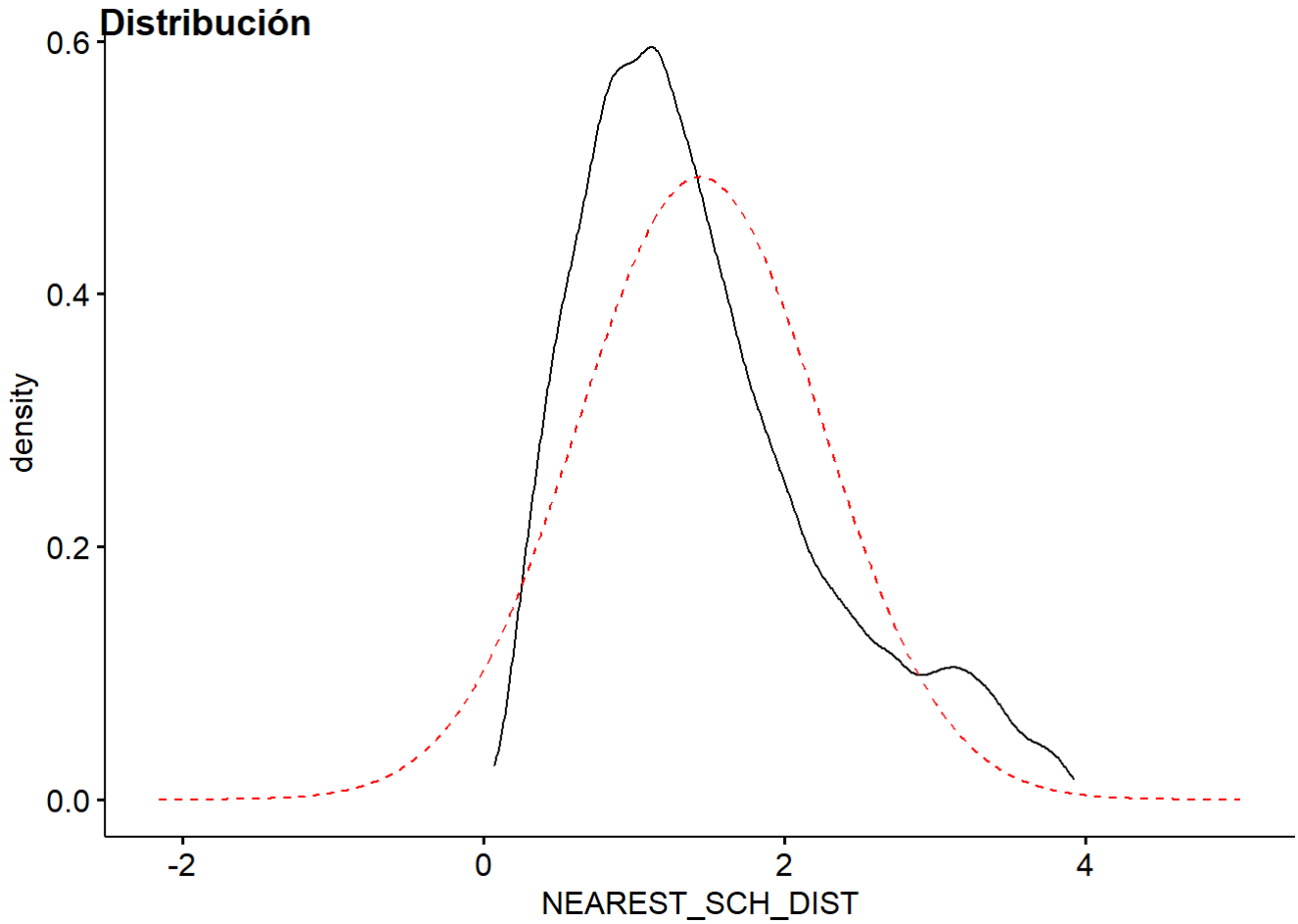
BUILD_YEAR,existe una marcada diferencia entre la mediana y la media luego se puede presumir que existen valores atípicos en los datos. Adicionalmente la variable no sigue una distribución normal.

```
xplot <- ggdensity(data_set, "BUILD_YEAR", fill = "BUILD_YEAR",
                   palette = "jco")+
  stat_overlay_normal_density(color = "red", linetype = "dashed")
ggarrange(xplot,nrow = 1, labels = "Distribución")
```



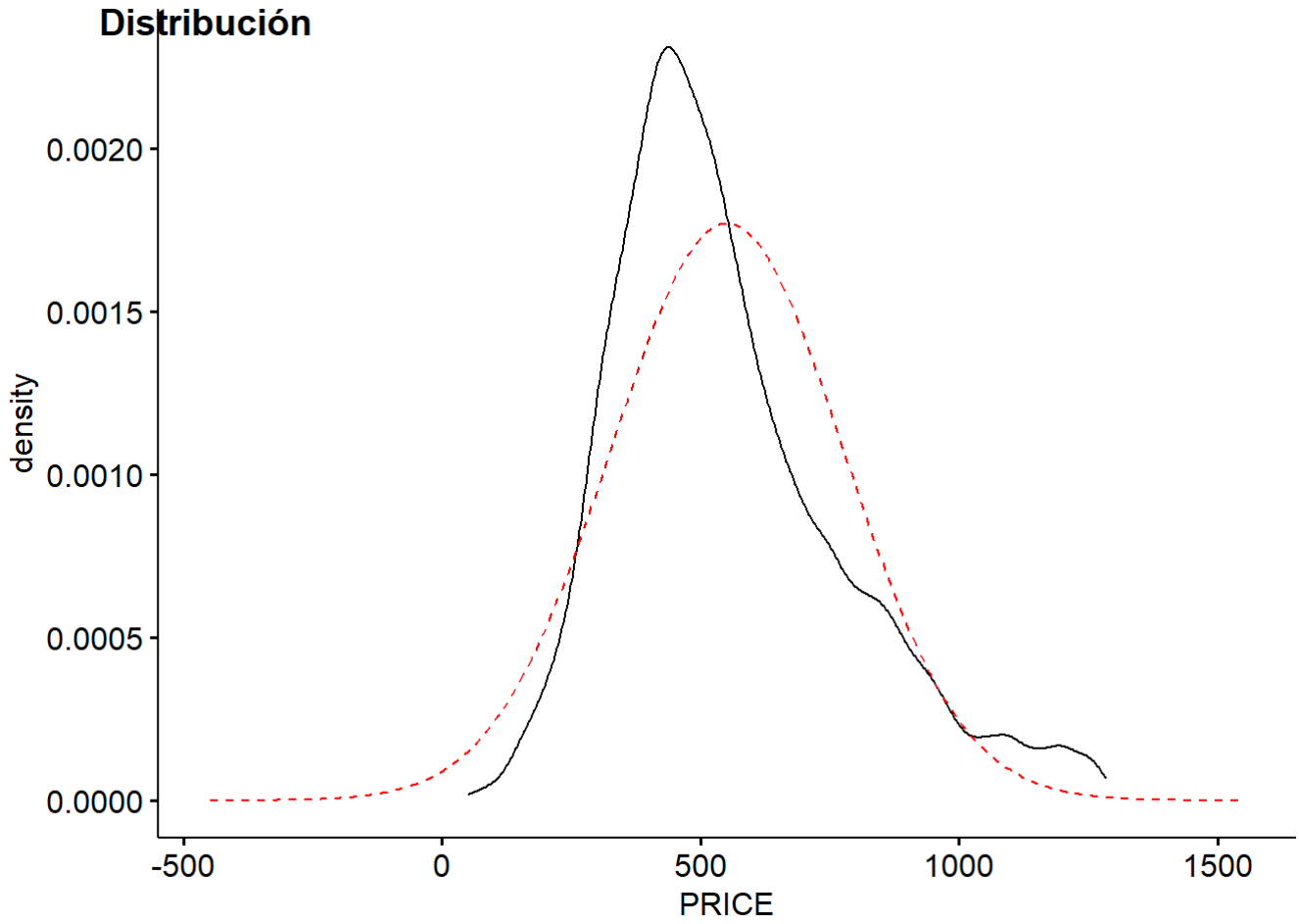
NEAREST_SCH_DIST, existe una marcada diferencia entre la mediana y la media luego se puede presumir que existen valores atípicos en los datos. Adicionalmente la variable no sigue una distribución normal.

```
xplot <- ggdensity(data_set, "NEAREST_SCH_DIST", fill = "NEAREST_SCH_DIST",
                   palette = "jco")+
  stat_overlay_normal_density(color = "red", linetype = "dashed")
ggarrange(xplot,nrow = 1, labels = "Distribución")
```



Precio_M2_total, existe una marcada diferencia entre la mediana y la media luego se puede presumir que existen valores atípicos en los datos. Adicionalmente la variable no sigue una distribución normal.

```
xplot <- ggdensity(data_set, "PRICE", fill = "PRICE",
                   palette = "jco")+
  stat_overlay_normal_density(color = "red", linetype = "dashed")
ggarrange(xplot,nrow = 1, labels = "Distribución")
```



En el conjunto de datos analizado se puede observar las variables continuas BUILD_YEAR CBD_DIST,NEAREST_STN_DIST,NEAREST_SCH_DIST,Precio_M2_total, son candidatas a realizar procesos de discretización para generar nuevos clústeres que puedan representar los datos. Se seleccionar estas variables porque su estructura puede ser representada como rangos de clústeres.

Para esto se introduce nuevas variables SEG_BUILD_YEAR SEG_CBD_DIST,SEG_NEAREST_STN_DIST,SEG_NEAREST_SCH_DIST,SEG_Precio_M2_total, en donde los rangos tienen la misma amplitud debido a que para ciertos casos no hay muchos registros con esas condiciones. La discretización de datos nos permitirá utilizarlos posteriormente en los análisis visuales.

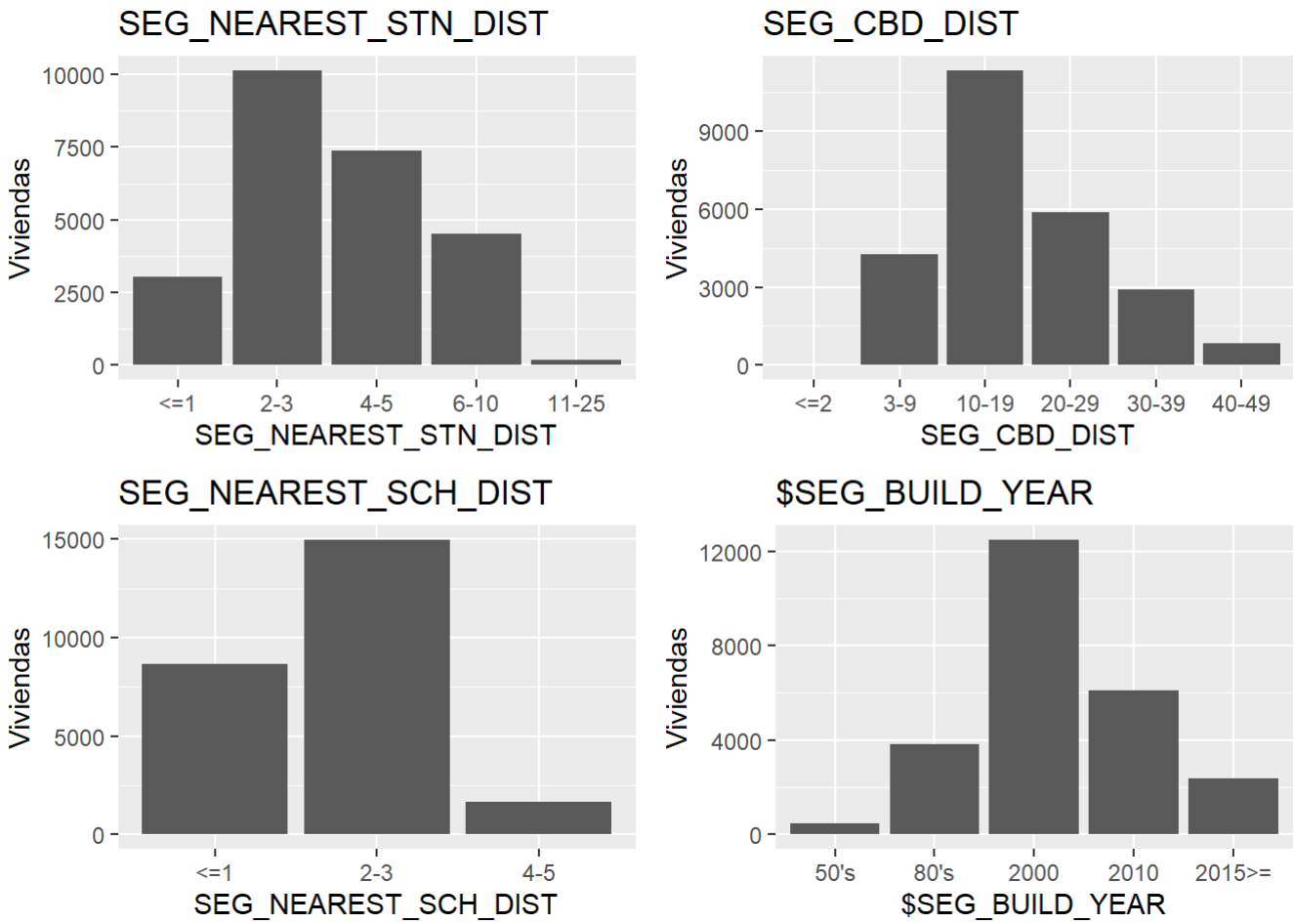
```
data_set["SEG_BUILD_YEAR"] <-
  cut(data_set$BUILD_YEAR, breaks = c(0,1950,1975,2000,2010,2017),
      labels = c("50's","80's","2000","2010","2015>="))

data_set["SEG_CBD_DIST"] <-
  cut(data_set$CBD_DIST, breaks = c(0,2,10,20,30,40,50,60),
      labels = c("<=2","3-9","10-19","20-29","30-39","40-49","50>="))
data_set["SEG_NEAREST_STN_DIST"] <-
  cut(data_set$NEAREST_STN_DIST, breaks = c(0,1,3,5,10,25,35),
      labels = c("<=1","2-3","4-5","6-10","11-25","26>="))
data_set["SEG_NEAREST_SCH_DIST"] <-
  cut(data_set$NEAREST_SCH_DIST, breaks = c(0,1,3,5,10,25,50),
      labels = c("<=1","2-3","4-5","6-10","11-24","25>="))
data_set["SEG_Precio_M2_total"] <-
  cut(data_set$Precio_M2_total, breaks = c(0,1,500,1000,2000,5000,8000,10000),
      labels = c("<=1","2-500","501-1000","1001-2000","2001-5000","5001-8000","8001>="))
data_set["SEG_Precio"] <-
  cut(data_set$PRICE, breaks = c(0,500,750,1000,1500,2000,3000),
      labels = c("<=500","501-750","751-1000","1001-1500","1501-2000","2001>="))

data_set$SEG_BUILD_YEAR <- as.factor(data_set$SEG_BUILD_YEAR)
data_set$SEG_CBD_DIST <- as.factor(data_set$SEG_CBD_DIST)
data_set$SEG_NEAREST_STN_DIST <- as.factor(data_set$SEG_NEAREST_STN_DIST)
data_set$SEG_NEAREST_SCH_DIST <- as.factor(data_set$SEG_NEAREST_SCH_DIST)
data_set$SEG_Precio_M2_total <- as.factor(data_set$SEG_Precio_M2_total)
data_set$SEG_Precio <- as.factor(data_set$SEG_Precio)
```

1.4.1 Selección de los grupos de datos que se quieren analizar/comparar

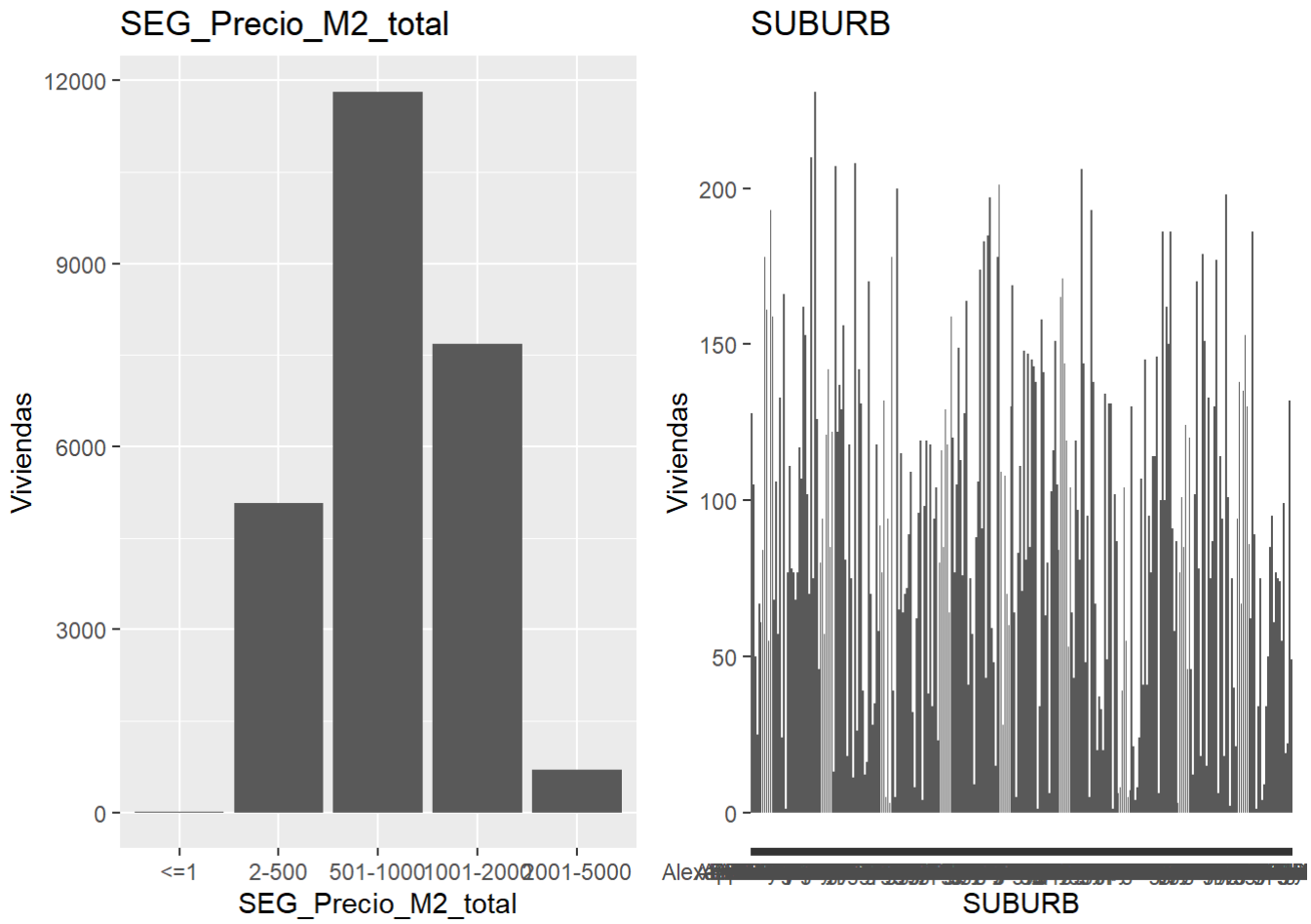
Para un conocimiento mayor sobre los datos, tenemos a nuestro alcance unas herramientas muy valiosas: las herramientas de visualización. Para dichas visualizaciones, haremos uso de los paquetes ggplot2, gridExtra y grid de R.

Siempre es importante analizar los datos que tenemos ya que las conclusiones dependerán de las características de la muestra.

```
grid.newpage()
plotbyChecking_balance<-ggplot(data_set,aes(SEG_BUILD_YEAR ))+geom_bar() +labs(x="$SEG_BUILD_
YEAR", y="Viviendas")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("blue"
,"#008000"))+ggtitle("$SEG_BUILD_YEAR ")

plotbyCredit_history<-ggplot(data_set,aes(SEG_CBD_DIST))+geom_bar() +labs(x="SEG_CBD_DIST", y
="Viviendas")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("blue","#00800
0"))+ggtitle("SEG_CBD_DIST")

plotbysavings_balance<-ggplot(data_set,aes(SEG_NEAREST_STN_DIST))+geom_bar() +labs(x="SEG_NEA
REST_STN_DIST", y="Viviendas")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values
=c("blue","#008000"))+ggtitle("SEG_NEAREST_STN_DIST")

plotbyemployment_length<-ggplot(data_set,aes(SEG_NEAREST_SCH_DIST))+geom_bar() +labs(x="SEG_N
EAREST_SCH_DIST", y="Viviendas")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(valu
es=c("blue","#008000"))+ggtitle("SEG_NEAREST_SCH_DIST")

grid.arrange(plotbysavings_balance,plotbyCredit_history,plotbyemployment_length,plotbyCheckin
g_balance,ncol=2)
```



```
grid.newpage()
plotbypurpose<-ggplot(data_set,aes(SEG_Precio_M2_total ))+geom_bar() +labs(x="SEG_Precio_M2_t
otal", y="Viviendas")+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("blue"
,"#008000"))+ggtitle("SEG_Precio_M2_total")

plotbyother_debtors<-ggplot(data_set,aes(SUBURB))+geom_bar() +labs(x="SUBURB", y="Viviendas")
+ guides(fill=guide_legend(title=""))+ scale_fill_manual(values=c("blue","#008000"))+ggtitle(
"SUBURB")

grid.arrange(plotbypurpose,plotbyother_debtors,ncol=2)
```



La varialbe SUBURB al ser una variable del tipo Char no se ha podido categorizar, muestra una grafica no entendible. Eliminamos la variable SUBURB.

```
data_set[c("SUBURB")] <- NULL
```

La variable NEAREST_SCH_DIST, se identifica que hay mas viviendas compradas cuando una escula se encuentra mas cerca.

1.4.2 Comprobación de la normalidad y homogeneidad de la varianza.

Para la comprobación de que los valores que toman nuestras variables cuantitativas provienen de una población distribuida normalmente, utilizaremos la prueba de normalidad de AndersonDarling. Así, se comprueba que para que cada prueba se obtiene un p-valor superior al nivel de significación prefijado α = 0, 05. Si esto se cumple, entonces se considera que variable en cuestión sigue una distribución normal. Se identifica que ninguna variable es normal.

```
alpha = 0.05
col.names = colnames(data_set)
for (i in 1:ncol(data_set)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
  if (is.integer(data_set[,i]) | is.numeric(data_set[,i])) {
    p_val = ad.test(data_set[,i])$p.value
    if (p_val < alpha) {
      cat(col.names[i])
      # Format output
      if (i < ncol(data_set) - 1) cat(", ")
      if (i %% 3 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal:
## PRICE, FLOOR_AREA, BUILD_YEAR,
## CBD_DIST, NEAREST_STN_DIST, NEAREST_SCH_DIST,
## Precio_M2_total,
```

Normalizamos utilizando la función “rescale” que realiza un escalado de la variables en un rango [0,1] es decir, por la diferencia:

```
data_set_norm <- data_set[c( "CBD_DIST","NEAREST_STN_DIST","NEAREST_SCH_DIST","PRICE","BUILD_
YEAR","FLOOR_AREA")]

data_set_norm <- sapply(data_set_norm, rescale)
head(data_set_norm)
```

```
##       CBD_DIST NEAREST_STN_DIST NEAREST_SCH_DIST     PRICE BUILD_YEAR FLOOR_AREA
## [1,] 0.3600394       0.16778267        0.1967326 0.4165316  0.8055556 0.18750000
## [2,] 0.4482391       0.17734838        0.4099359 0.1912480  0.4722222 0.10023585
## [3,] 0.3518348       0.33996556        0.3897342 0.1653160  0.1111111 0.06839623
## [4,] 0.2144073       0.18691410        0.3750644 0.2220421  0.7361111 0.15330189
## [5,] 0.5446434       0.09125694        0.3003371 0.2901135  0.6388889 0.13797170
## [6,] 0.5631038       0.34953128        0.6272206 0.2828201  0.9583333 0.15448113
```

Seguidamente, pasamos a estudiar la homogeneidad de varianzas mediante la aplicación de un test de Fligner-Killeen. En este caso, estudiaremos esta homogeneidad en cuanto a los grupos conformados por los vehículos que presentan un motor turbo frente a un motor estándar. En el siguiente test, la hipótesis nula consiste en que ambas varianzas son iguales.

```
fligner.test(PRICE ~ NEAREST_STN_DIST, data = data_set_norm)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  PRICE by NEAREST_STN_DIST
## Fligner-Killeen:med chi-squared = 1908.9, df = 900, p-value < 2.2e-16
```

```
fligner.test(PRICE ~ NEAREST_SCH_DIST, data = data_set_norm)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  PRICE by NEAREST_SCH_DIST
## Fligner-Killeen:med chi-squared = 24173, df = 24997, p-value = 0.9999
```

```
fligner.test(PRICE ~ CBD_DIST, data = data_set_norm)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  PRICE by CBD_DIST
## Fligner-Killeen:med chi-squared = 1982.3, df = 491, p-value < 2.2e-16
```

```
fligner.test(PRICE ~ BUILD_YEAR, data = data_set_norm)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  PRICE by BUILD_YEAR
## Fligner-Killeen:med chi-squared = 463.54, df = 72, p-value < 2.2e-16
```

```
fligner.test(PRICE ~ FLOOR_AREA, data = data_set_norm)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  PRICE by FLOOR_AREA
## Fligner-Killeen:med chi-squared = 1301.7, df = 462, p-value < 2.2e-16
```

Solo para la variable NEAREST_SCH_DIST, obtenemos un p-valor superior a 0,05, aceptamos la hipótesis de que las varianzas de ambas muestras son homogéneas.

1.4.3 Aplicación de pruebas estadísticas para comparar los grupos de datos.

Analisis de Correlaciones

Procedemos a realizar un análisis de correlación entre las distintas variables para determinar cuáles de ellas ejercen una mayor influencia sobre el precio final de la vivienda. Para ello, se utilizará el coeficiente de correlación de Spearman, puesto que hemos visto que tenemos datos que no siguen una distribución normal. Se identifica que el precio se relaciona con la varible NEAREST_SCH_DIST

```
cor.test(data_set$PRICE,data_set$NEAREST_STN_DIST, method="spearman")
```

```
## Warning in cor.test.default(data_set$PRICE, data_set$NEAREST_STN_DIST, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_set$PRICE and data_set$NEAREST_STN_DIST
## S = 2.5712e+12, p-value = 6.722e-12
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## 0.04316365
```

```
cor.test(data_set$PRICE,data_set$CBD_DIST, method="spearman")
```

```
## Warning in cor.test.default(data_set$PRICE, data_set$CBD_DIST, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_set$PRICE and data_set$CBD_DIST
## S = 3.5781e+12, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.3315394
```

```
cor.test(data_set$PRICE,data_set$NEAREST_SCH_DIST, method="spearman")
```

```
## Warning in cor.test.default(data_set$PRICE, data_set$NEAREST_SCH_DIST, method =
## "spearman"): Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_set$PRICE and data_set$NEAREST_SCH_DIST
## S = 2.5976e+12, p-value = 1.142e-07
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.0333534
```

```
cor.test(data_set$Precio_M2_total,data_set$NEAREST_SCH_DIST, method="spearman")
```

```
## Warning in cor.test.default(data_set$Precio_M2_total, data_set$NEAREST_SCH_DIST, : Cannot
## compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_set$Precio_M2_total and data_set$NEAREST_SCH_DIST
## S = 2.7978e+12, p-value = 5.968e-11
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##        rho
## -0.0411598
```

```
cor.test(data_set$PRICE,data_set$BUILD_YEAR, method="spearman")
```

```
## Warning in cor.test.default(data_set$PRICE, data_set$BUILD_YEAR, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_set$PRICE and data_set$BUILD_YEAR
## S = 2.701e+12, p-value = 0.4148
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##          rho
## -0.005130821
```

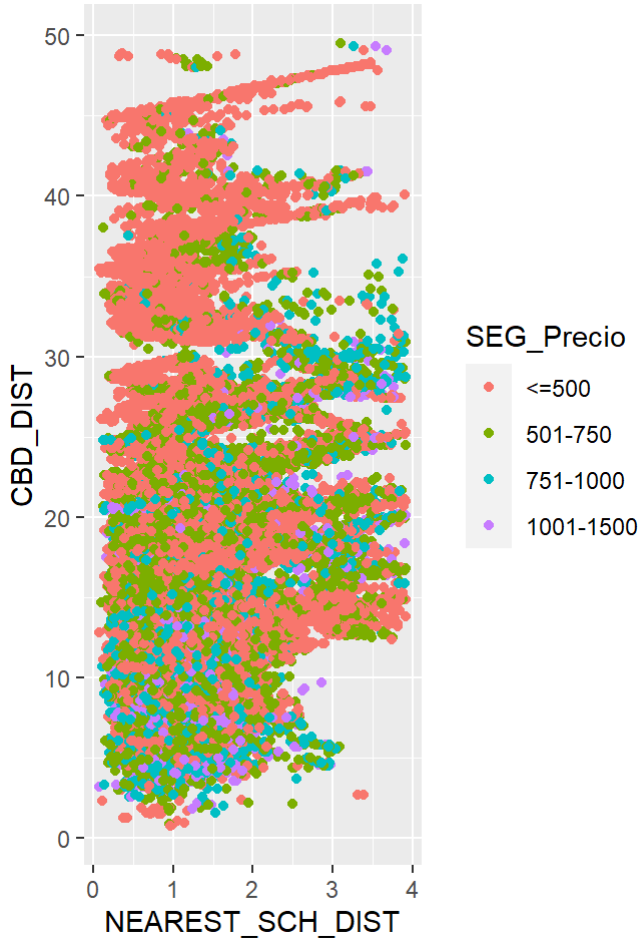
```
cor.test(data_set$PRICE,data_set$FLOOR_AREA, method="spearman")
```

```
## Warning in cor.test.default(data_set$PRICE, data_set$FLOOR_AREA, method = "spearman"):
## Cannot compute exact p-value with ties
```

```
##
##  Spearman's rank correlation rho
##
## data:  data_set$PRICE and data_set$FLOOR_AREA
## S = 1.1063e+12, p-value < 2.2e-16
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
##       rho
## 0.5883133
```

Se mantiene la relación entre el precio y la cercania a una escuela NEAREST_SCH_DIST y CBD_DIST.

```
  plot_dot_plot_precio <- function(data_set,color ){
  p3<-ggplot(data = data_set, aes_string(x="NEAREST_SCH_DIST", y="CBD_DIST")) +
              geom_point(aes_string(colour=color))
  lay <- rbind(c(1,2))
  grid.arrange(p3, nrow = 1,layout_matrix = lay)
}

plot_dot_plot_precio(na.omit(data_set),"SEG_Precio")
```



Regresión Lineal

Revisión de relación entre variables se reconfirma que las variables “CBD_DIST”,“NEAREST_STN_DIST”,“NEAREST_SCH_DIST”, “FLOOR_AREA” son las que mas se relacionan con el precio.

Se calculará un modelo de regresión lineal utilizando regresores cuantitativos para poder realizar las predicciones de los precios.

Regresores cuantitativos con mayor coeficiente de correlación con respecto al precio: “CBD_DIST”,“NEAREST_STN_DIST”,“NEAREST_SCH_DIST”

```
distancia_centro=data_set$CBD_DIST
distancia_estacion_bus=data_set$NEAREST_STN_DIST
distancia_escuela=data_set$NEAREST_SCH_DIST
area_construida=data_set$FLOOR_AREA
```

Variable a predecir “PRECIO” por metro cuadrado

```
precio=data_set$Precio_M2_total
```

```
modelo1 = lm(precio~distancia_escuela,data=data_set)
modelo2 = lm(precio~distancia_escuela+distancia_centro,data=data_set)
modelo3 = lm(precio~distancia_escuela+distancia_estacion_bus,data=data_set)
modelo4 = lm(precio~distancia_escuela+distancia_estacion_bus+distancia_centro+area_construid
a,data=data_set)
modelo5 = lm(precio~distancia_estacion_bus+distancia_centro,data=data_set)
modelo6 = lm(precio~distancia_centro,data=data_set)
modelo7 = lm(precio~area_construida,data=data_set)

# Tabla con los coeficientes de determinación de cada modelo
tabla.coeficientes <- matrix(c(1, summary(modelo1)$r.squared,
2, summary(modelo2)$r.squared,
3, summary(modelo3)$r.squared,
4, summary(modelo4)$r.squared,
5, summary(modelo5)$r.squared,
6, summary(modelo6)$r.squared,
7, summary(modelo7)$r.squared),
ncol = 2, byrow = TRUE)
colnames(tabla.coeficientes) <- c("Modelo", "R^2")
tabla.coeficientes
```

```
##      Modelo         R^2
## [1,]      1 0.004489146
## [2,]      2 0.108002288
## [3,]      3 0.030230581
## [4,]      4 0.187799417
## [5,]      5 0.134474083
## [6,]      6 0.106998037
## [7,]      7 0.033727564
```

En este caso, tenemos que el cuarto modelo es el más conveniente dado que tiene un mayor coeficiente de determinación. Sin embargo el valor del coeficiente es bastante bajo.

Ahora, empleando este modelo, podemos proceder a realizar predicciones de precios de vehículos como el siguiente:

```
newdata <- data.frame(
distancia_centro = 17.50,
distancia_estacion_bus = 3.2,
distancia_escuela = 1.35,
area_construida=500
)
# Predecir el precio
predict(modelo4, newdata)
```

```
##        1
## 1460.896
```

En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes.

1.5 Representación de los resultados a partir de tablas y gráficas.

Este apartado se puede responder a lo largo de la práctica, sin necesidad de concentrar todas las representaciones en este punto de la práctica.

1.5.1 ¿cuáles son las conclusiones?

Inicialmente se han realizado tareas de limpieza, identificando campos vacíos y outliers, se eliminaron variables que no tenían suficiente relación con las variables. Se procede a normalizar los valores pues estaban con valores muy altos. Mediante la discretización,se puede observar las variables mas importantes que interfieren para identificar el precio de una viviendo, llegando a la conclusión que las variables mas importantes son cercanía a la escuela SEG_NEAREST_SCH_DIST y cercania al centro(CBD_DIST). En base a estas variables se pudo predecir el valor de una construcción.

Los modelos son faciles de implementar con los algoritmos que presta los paquetes, sin embargo si se debe tener mucho cuidado con las variables del set de datos que se va ha manejar, y esto depende de un buen analisis de la información y comparativa entre variables, adicionalmente del tratamiento previo que se debe dar a los mismos.

1.5.2 ¿Los resultados permiten responder al problema?

Si, se pudo predecir el valor de una vivienda en base a las variables que mas influyen en el calculo del precio a vender.

1.6 Código: Hay que adjuntar el código, preferiblemente en R,

```
write.csv(data_set, file="data_out.csv")
```