



Hoja de Trabajo 3

Brian Carrillo - 21108
Diego Alonzo - 20172
Carlos López - 21666

Guatemala, 28 de febrero del 2024

1. Descargue los conjuntos de datos de la plataforma Kaggle.

 test	2/22/2024 3:48 PM	Microsoft Excel Co...	441 KB
 train	2/22/2024 3:48 PM	Microsoft Excel Co...	450 KB

Ya se descargaron y se logueo para poder descargarlos.

2. Análisis exploratorio extenso de los datos. Explique bien todos los hallazgos. No ponga solo gráficas y código. Debe llegar a conclusiones interesantes para poder predecir. Explique el preprocesamiento que necesitó hacer.

Esto sería qué significa cada columna:

- a. SalePrice: el precio de venta de la propiedad en dólares. Esta es la variable objetivo que queremos predecir.
- b. MSSubClass: La clase de construcción
- c. MSZoning: La clasificación general de zonificación
- d. LotFrontage: Pies lineales de calle conectados a la propiedad
- e. LotArea: Tamaño del lote en pies cuadrados
- f. Street: Tipo de acceso a la carretera
- g. Alley: Tipo de acceso al callejón
- h. LotShape: Forma general de la propiedad
- i. LandContour: Planitud de la propiedad
- j. Utilities: Tipo de servicios disponibles
- k. LotConfig: Configuración del lote
- l. LandSlope: Pendiente de la propiedad
- m. Neighborhood: Ubicaciones físicas dentro de los límites de la ciudad de Ames
- n. Condition1: Proximidad a carretera principal o ferrocarril
- o. Condition2: Proximidad a carretera principal o ferrocarril (si hay una segunda presente)
- p. BldgType: Tipo de vivienda
- q. HouseStyle: Estilo de vivienda
- r. OverallQual: Calidad general de material y acabado
- s. OverallCond: Calificación general de condición
- t. YearBuilt: Fecha original de construcción
- u. YearRemodAdd: Fecha de remodelación
- v. RoofStyle: Tipo de techo
- w. RoofMatl: Material del techo
- x. Exterior1st: Revestimiento exterior de la casa
- y. Exterior2nd: Segundo revestimiento exterior de la casa (si hay más de un material)
- z. MasVnrType: Tipo de revestimiento de mampostería
- aa. MasVnrArea: Área de revestimiento de mampostería en pies cuadrados
- bb. ExterQual: Calidad del material exterior
- cc. ExterCond: Condición actual del material exterior
- dd. Foundation: Tipo de cimentación
- ee. BsmtQual: Altura del sótano
- ff. BsmtCond: Condición general del sótano
- gg. BsmtExposure: Paredes del sótano a nivel del suelo o de jardín
- hh. BsmtFinType1: Calidad del área terminada del sótano
- ii. BsmtFinSF1: Pies cuadrados terminados de tipo 1
- jj. BsmtFinType2: Calidad de la segunda área terminada (si está presente)
- kk. BsmtFinSF2: Pies cuadrados terminados de tipo 2

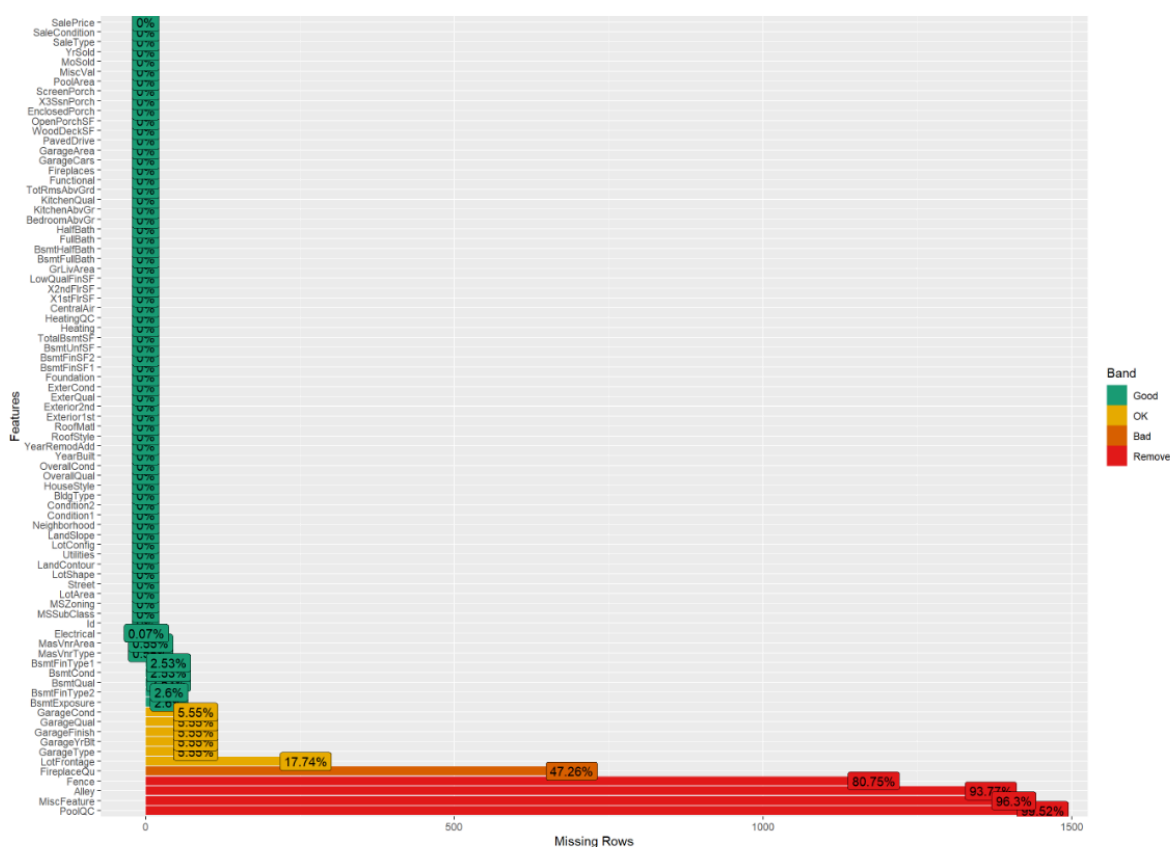
- ll. BsmtUnfSF: Pies cuadrados sin terminar del área del sótano
- mm. TotalBsmtSF: Pies cuadrados totales del área del sótano
- nn. Heating: Tipo de calefacción
- oo. HeatingQC: Calidad y condición de la calefacción
- pp. CentralAir: Aire acondicionado central
- qq. Electrical: Sistema eléctrico
- rr. 1stFlrSF: Pies cuadrados del primer piso
- ss. 2ndFlrSF: Pies cuadrados del segundo piso
- tt. LowQualFinSF: Pies cuadrados terminados de baja calidad (todos los pisos)
- uu. GrLivArea: Pies cuadrados de área habitable sobre el nivel del suelo.
- vv. BsmtFullBath: Baños completos en el sótano
- ww. BsmtHalfBath: Medios baños en el sótano
- xx. FullBath: Baños completos sobre el nivel del suelo
- yy. HalfBath: Medios baños sobre el nivel del suelo
- zz. Bedroom: Número de dormitorios sobre el nivel del sótano
- aaa. Kitchen: Número de cocinas
- bbb. KitchenQual: Calidad de la cocina
- ccc. TotRmsAbvGrd: Total de habitaciones sobre el nivel del suelo (no incluye baños)
- ddd. Functional: Calificación de funcionalidad del hogar
- eee. Fireplaces: Número de chimeneas
- fff. FireplaceQu: Calidad de la chimenea
- ggg. GarageType: Ubicación del garaje
- hhh. GarageYrBlt: Año en que se construyó el garaje
- iii. GarageFinish: Acabado interior del garaje
- jjj. GarageCars: Tamaño del garaje en capacidad para automóviles
- kkk. GarageArea: Tamaño del garaje en pies cuadrados
- lll. GarageQual: Calidad del garaje
- mmm. GarageCond: Condición del garaje
- nnn. PavedDrive: Entrada pavimentada
- ooo. WoodDeckSF: Área de terraza de madera en pies cuadrados
- ppp. OpenPorchSF: Área de porche abierto en pies cuadrados
- qqq. EnclosedPorch: Área de porche cerrado en pies cuadrados
- rrr. 3SsnPorch: Área de porche de tres estaciones en pies cuadrados
- sss. ScreenPorch: Área de porche con pantalla en pies cuadrados
- ttt. PoolArea: Área de piscina en pies cuadrados
- uuu. PoolQC: Calidad de la piscina
- vvv. Fence: Calidad de la cerca
- www. MiscFeature: Característica miscelánea no cubierta en otras categorías
- xxx. MiscVal: Valor en dólares de la característica miscelánea
- yyy. MoSold: Mes de venta
- zzz. YrSold: Año de venta
- aaaa. SaleType: Tipo de venta
- bbbb. SaleCondition: Condición de venta

Dado que queríamos conocer los datos y descubrir qué tan completos se encontraban utilizamos data explorer para explorarlos. Por lo que en primer lugar, se debe de denotar que no se posee un set de datos muy grande dado que solo hay 1460 valores. Además se logró ver que ninguna columna está completamente vacía, a su vez existen 81 diferentes variables y de ellas 43 son discretas y 38 son continuas.

Raw Counts

Name	Value
Rows	1,460
Columns	81
Discrete columns	43
Continuous columns	38
All missing columns	0
Missing observations	6,965
Complete Rows	0
Total observations	118,260
Memory allocation	737.6 Kb

Missing Data Profile



Por lo que al revisar la gráfica de missing data profile se encontró que estas columnas se encontraban sin datos en general, sobretodo:

cccc. PoolQC: que es la calidad de la piscina por ende tiene sentido que no hayan datos completos ya que no todas las casas poseerán piscina.

```
> sum(!is.na(train[, "PoolQC"]))
[1] 7
> train$PoolQC[!is.na(train$PoolQC)]
[1] "Ex" "Fa" "Gd" "Ex" "Gd" "Fa" "Gd"
```

Solo 7 casas tienen alguna calidad en la piscina, y de las demás no se sabe qué significa cada una.

dddd. MiscFeature: que es una característica compuesta que no está cubierta en otras categorías.

```
> sum(!is.na(train[, "MiscFeature"]))
[1] 54
> train$MiscFeature[!is.na(train$MiscFeature)]
[1] "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed"
[16] "Gar2" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Othr" "Shed" "Shed"
[31] "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Shed" "Othr" "Shed" "Shed" "Shed" "Shed" "Shed"
[46] "Shed" "Shed" "Shed" "Shed" "Gar2" "Shed" "Shed" "TenC" "Shed"
```

Solo 54 no son na y después predomina shed, aunque incluso en esta categoría que es casi un otros existe un subgrupo que es de others.

eeee. Alley: callejón que está compuesto mayormente por n/a, aunque los demás valores son si está pavimentado o sin pavimentar.

```
> sum(!is.na(train[, "Alley"]))
[1] 91
> train$Alley[!is.na(train$Alley)]
[1] "Grv" "Pave" "Pave" "Grv" "Pave" "Grv" "Grv" "Pave" "Pave" "Grv" "Grv" "Grv" "Grv" "Pave"
[15] "Pave" "Grv" "Pave" "Grv" "Grv" "Pave" "Pave" "Pave" "Pave" "Grv" "Grv" "Grv" "Grv" "Grv"
[29] "Grv" "Pave" "Pave" "Grv" "Grv" "Pave" "Pave" "Grv" "Grv" "Pave" "Grv" "Grv" "Pave" "Grv"
[43] "Grv" "Pave" "Pave" "Grv" "Grv" "Grv" "Grv" "Pave" "Pave" "Pave" "Pave" "Grv" "Grv" "Pave"
[57] "Pave" "Pave" "Pave" "Grv" "Grv" "Grv" "Pave" "Pave" "Grv" "Grv" "Grv" "Pave" "Pave" "Grv"
[71] "Pave" "Grv" "Grv" "Pave" "Grv" "Grv" "Pave" "Pave" "Grv" "Grv" "Pave" "Pave" "Grv" "Grv"
[85] "Grv" "Pave" "Grv" "Pave" "Grv" "Grv" "Pave"
```

Y solamente 91 filas son no nulas.

ffff. Fence: por otro lado esta característica brinda la calidad de la cerca que bordea la casa, sin embargo, no todas las casas poseen cerca. Por ello realmente tener o no una cerca no es un dato relevante.

```
> sum(!is.na(train[, "Fence"]))
[1] 281
> train$Fence[!is.na(train$Fence)]
[1] "MnPrv" "GdWo" "GdPrv" "MnPrv" "GdPrv" "MnPrv" "MnPrv" "MnPrv" "GdWo" "MnPrv" "MnPrv" "MnPrv" "MnPrv"
[14] "MnPrv" "MnPrv" "GdPrv" "GdPrv" "GdWo" "GdWo" "MnPrv" "MnPrv" "MnPrv" "GdWo" "MnPrv" "MnPrv" "MnPrv"
[27] "MnPrv" "MnWw" "GdPrv" "MnPrv" "MnPrv" "GdPrv" "MnPrv" "MnPrv" "MnPrv" "GdWo" "GdWo" "MnPrv" "GdWo"
[40] "MnPrv" "MnPrv" "MnPrv" "GdPrv" "GdPrv" "MnPrv" "GdPrv" "MnPrv" "MnPrv" "MnWw" "GdWo" "MnPrv" "MnPrv"
[53] "MnPrv" "MnPrv" "MnPrv" "MnPrv" "MnPrv" "GdWo" "MnPrv" "GdWo" "GdWo" "GdPrv" "GdPrv" "MnPrv" "GdPrv"
[66] "MnPrv" "GdPrv" "MnPrv" "GdPrv" "GdWo" "MnPrv" "MnPrv" "GdPrv" "MnPrv" "GdWo" "MnPrv" "MnPrv" "MnPrv"
[79] "GdWo" "GdWo" "MnPrv" "MnPrv" "GdWo" "GdPrv" "GdWo" "MnPrv" "MnPrv" "GdPrv" "MnPrv" "MnPrv" "MnPrv"
```

Solamente 281 filas no son nulas.

gggg. FireplaceQu: es la característica que dice la calidad de la chimenea, en este caso no todas las casas tienen chimeneas, por ende no es del todo relevante. Y aunque podría ser categorizado con la moda realmente al ser únicamente la calidad de la chimenea no es realmente necesario dado que existe la característica fireplaces que indica el número de chimeneas dentro de la casa.

```
> sum(!is.na(train[, "FireplaceQu"]))
[1] 770
> train$FireplaceQu[!is.na(train$FireplaceQu)]
[1] "TA" "TA" "Gd" "TA" "Gd" "TA" "TA" "TA" "Gd" "Gd" "Fa" "TA" "Gd" "Gd" "Gd" "TA" "TA" "Gd" "Gd" "Gd"
[21] "Gd" "Gd" "Gd" "TA" "TA" "Gd" "Gd" "Ex" "Gd" "Gd" "TA" "Gd" "Gd" "Gd" "Gd" "TA" "Gd" "TA" "Gd"
[41] "Gd" "TA" "TA" "Gd" "Gd" "TA" "TA" "TA" "Gd" "TA" "Gd" "TA" "Gd" "Gd" "TA" "Po" "TA" "Gd" "Gd" "Gd"
[61] "TA" "TA" "TA" "Fa" "Gd" "TA" "TA" "Gd" "Fa" "TA" "Po" "Gd" "Gd" "Gd" "Gd" "Gd" "Gd" "Gd" "Gd" "Gd"
[81] "Gd" "Gd" "Gd" "TA" "Gd" "TA" "TA" "TA" "Gd" "TA" "Gd" "Gd" "TA" "Gd" "Gd" "TA" "TA" "Gd" "TA" "TA"
[101] "Gd" "Ex" "Gd" "Fa" "Gd" "TA" "Po" "Gd" "TA" "Fa" "TA" "TA" "TA" "Ex" "TA" "Fa" "TA" "TA" "Po" "TA"
```

Aquí solamente 770 no son nulos.

```
> distinct(train,train$FireplaceQu, .keep_all = FALSE)
train$FireplaceQu
1                <NA>
2                 TA
3                 Gd
4                 Fa
5                 Ex
6                 Po
```

Y solo son 6 categorías.

hhhh. LotFrontage: es la característica que indica los pies de banqueta que se poseen en la propiedad (el frente). Esto podría ser útil aunque en primera instancia los 259 datos restantes se tendrían que trabajar, ya sea asumiendo que no tienen cerca, que tienen el promedio o la mediana para no sesgar.

```
> sum(!is.na(train[, "LotFrontage"]))
[1] 1201
> train$LotFrontage[!is.na(train$LotFrontage)]
[1] 65 80 68 60 84 85 75 51 50 70 85 91 51 72 66 70 101 57 75 44 110 60 98 47 60
[26] 50 85 70 60 108 112 74 68 65 84 115 70 61 48 84 33 66 52 110 68 60 100 24 89 66
[51] 60 63 60 44 50 76 72 47 81 95 69 74 85 60 21 50 72 60 100 32 78 80 121 122 40
[76] 105 60 60 85 80 60 69 78 73 85 77 77 64 94 75 60 50 85 105 75 77 61 34 74 90
[101] 65 50 75 55 48 60 55 69 69 88 75 78 80 82 73 65 70 78 71 78 70 24 51 63 120
[126] 107 84 60 60 92 100 134 110 95 55 40 62 86 62 141 44 80 47 84 97 63 60 54 60 63
```

iiii. GarageFinish: este indica si el garage fue terminado o no o si está parcialmente terminado (RFn).

```
> sum(!is.na(train[, "GarageFinish"]))
[1] 1379
> train$GarageFinish[!is.na(train$GarageFinish)]
[1] "RFn" "RFn" "RFn" "Unf" "RFn" "Unf" "RFn" "RFn" "Unf" "RFn" "Unf" "Fin" "Unf" "RFn" "RFn" "Unf" "Fin"
[18] "Unf" "Unf" "Unf" "RFn" "Unf" "RFn" "Unf" "Unf" "RFn" "Unf" "RFn" "RFn" "Unf" "Unf" "Unf" "RFn" "RFn"
[35] "Fin" "Fin" "Unf" "Fin" "Unf" "RFn" "RFn" "RFn" "Unf" "RFn" "RFn" "RFn" "Unf" "Fin" "Unf" "Unf"
[52] "Fin" "Unf" "RFn" "Fin" "RFn" "Fin" "Unf" "Unf" "Unf" "RFn" "Unf" "RFn" "RFn" "RFn" "Unf" "Fin"
[69] "Fin" "Unf" "Fin" "Unf" "Unf" "Unf" "Unf" "Unf" "Unf" "Fin" "Fin" "RFn" "Unf" "Fin" "Unf" "Fin" "RFn"
[86] "Unf" "Unf" "Unf" "Unf" "RFn" "Fin" "RFn" "Fin" "Unf" "RFn" "Fin" "Unf" "RFn" "Unf" "RFn" "Unf" "Unf"
[103] "RFn" "Unf" "Fin" "Fin" "Unf" "Unf" "Unf" "Unf" "Unf" "Unf" "Fin" "Unf" "Unf" "Unf" "RFn" "Unf" "RFn"
```

```
> distinct(train,train$GarageFinish, .keep_all = FALSE)
train$GarageFinish
1                RFn
2                Unf
3                Fin
4                <NA>
```

Posee 1379 valores diferentes a na y son 3 categorías diferentes, finalizados, no finalizados y parcialmente terminados.

jjjj. GarageCond: indica la condición en la cuál se encuentra el garage, de manera que indica si es típico (TA), aceptable (Fa), bueno (Gd), pobre (PO) o excelente (Ex).

```
> sum(!is.na(train[, "GarageCond"]))
[1] 1379
> distinct(train,train$GarageCond, .keep_all = FALSE)
train$GarageCond
1                TA
2                Fa
3                <NA>
4                Gd
5                Po
6                Ex
> train$GarageCond[!is.na(train$GarageCond)]
[1] "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA"
[21] "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "Fa" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA"
```

Así mismo, tiene 1379 filas que no son nulos.

kkkk. GarageQual: Es similar a garage cond, de manera que a partir de la misma categorización divide en grupos la calidad del garage y tiene igual 1379 valores distintos a nulos.

```
> sum(!is.na(train[, "GarageQual"]))
[1] 1379
> distinct(train, train$GarageQual, .keep_all = FALSE)
  train$GarageQual
1              TA
2              Fa
3              Gd
4             <NA>
5              Ex
6              Po
> train$GarageQual[!is.na(train$GarageQual)]
[1] "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "Fa" "Gd" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA"
[21] "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "Fa" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA"
[41] "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA" "TA"
```

IIII. GarageYrBlt: En este caso solo indica el año en el cuál fue construido el garage y posee igualmente 1379 valores.

mmmm. GarageType: En este caso solo indica qué tipo de garage es.

```
> sum(!is.na(train[, "GarageType"]))
[1] 1379
> distinct(train, train$GarageType, .keep_all = FALSE)
  train$GarageType
1      Attchd
2      Detchd
3      BuiltIn
4      CarPort
5      <NA>
6      Basement
7      2Types
> train$GarageType[!is.na(train$GarageType)]
[1] "Attchd" "Attchd" "Attchd" "Detchd" "Attchd" "Attchd" "Attchd" "Attchd" "Detchd" "Attchd"
[11] "Detchd" "BuiltIn" "Detchd" "Attchd" "Attchd" "Detchd" "Attchd" "CarPort" "Detchd" "Attchd"
```

De manera que se clasifican en adjunto a la casa, separado de la casa, construido en una parte estructural de la casa, existe una cochera, el garage está en el sótano y que existe más de un tipo de garage en la casa.

Puede ser que dependiendo del tipo de garage indique que una casa tenga o no un mejor precio.

Ahora, lo primero que realizaremos será eliminar las columnas que no son útiles en absoluto, PoolQC, MiscFeature, Alley, Fence y FireplaceQu puesto que según el análisis la data que se posee es menor al 47.26%.

```
> invalidos<-c("PoolQC", "MiscFeature", "Alley", "Fence", "FireplaceQu")
> train1<-train[,-which(names(train)%in% invalidos)]
> View(train1)
> create_report(
+   train1,
+   output_file = "train_report1.html",
+   output_dir = getwd(),
+   config = configure_report(),
+   report_title = "Houses Data Report"
+ )
```

processing file: report.rmd

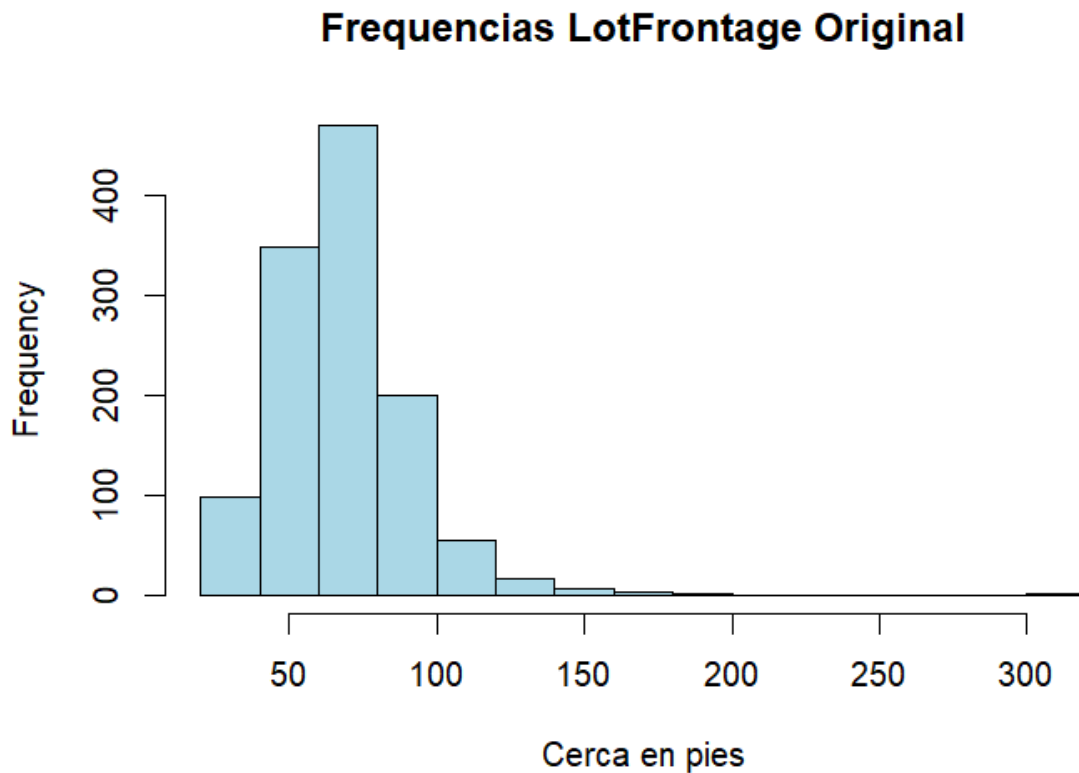
|.....| 57% [plot normal] aa|

Para las otras columnas que no poseen tantos datos pues se hará un relleno con dichas columnas dado que puede que lleguen a ser importantes para el análisis del valor de la casa.

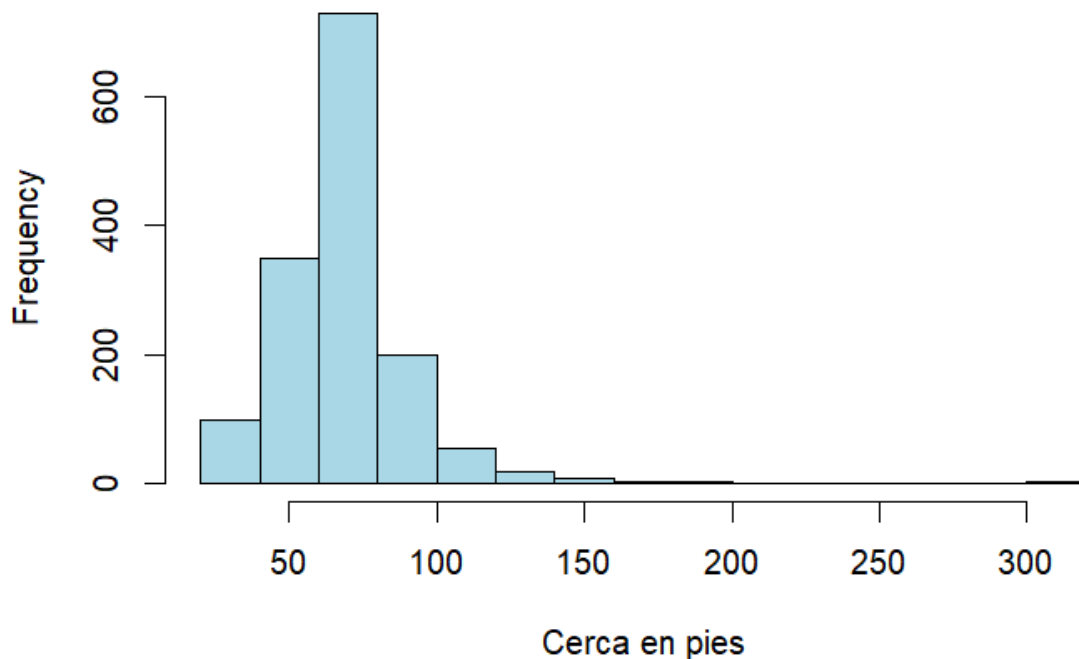
nnnn. LotFrontage:

```
> mean(train$LotFrontage[!is.na(train$LotFrontage)])
[1] 70.04996
> median(train$LotFrontage[!is.na(train$LotFrontage)])
[1] 69
> train1$LotFrontage[is.na(train1$LotFrontage)] <- median(train$LotFrontage[!is.na(train$LotFrontage)])
> train1$LotFrontage
[1] 65 80 68 60 84 85 75 69 51 50 70 85 69 91 69 51 69 72 66 70 101 57 75 44 69
[26] 110 60 98 47 60 50 69 85 70 60 108 112 74 68 65 84 115 69 69 70 61 48 84 33 66
[51] 69 52 110 68 60 100 24 89 66 60 63 60 44 50 69 76 69 72 47 81 95 69 74 85 60
```

Se rellenaron sus datos con la media y se realizó un histograma de frecuencias para observar cambios en los datos y comparar cómo estaban los datos previo a el relleno de datos y luego del mismo.



Frecuencias LotFrontage Modificado

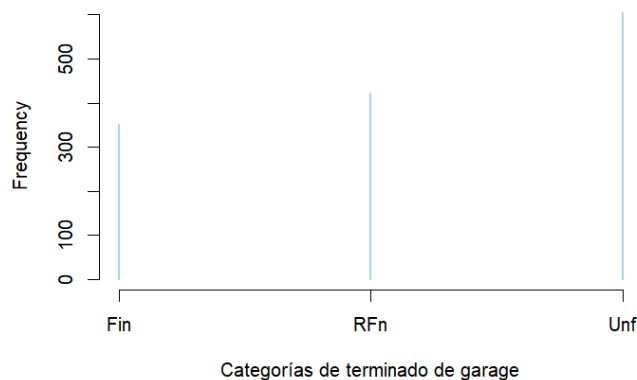


Y tal cómo se puede observar no existe un gran cambio, tan solo se concentraron los valores en la mediana pero únicamente eso, realmente la gráfica previo al rellenado ya tenía una forma leptocúrtica y solo se hizo un poco más evidente luego de la modificación, así mismo, no se nota que haya generado algún sesgo.

oooo. GarageFinish: en el caso de esta variable cualitativa se calculó la moda y se modificó en virtud de ello.

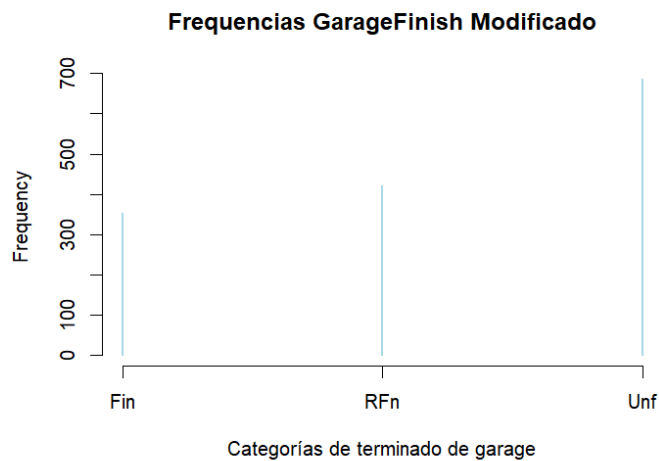
```
> table(train1$GarageFinish)
Fin RFn Unf
352 422 605
> names(table(train1$GarageFinish))[which.max(table(train1$GarageFinish))]
[1] "Unf"
```

Frecuencias GarageFinish Original



De manera que Unf era la moda, y dado que garageFinish solo le hacían falta 81 registros lo que equivale a un 5.55% de registros entonces se decidió realizar ese

rellenado.



```
> table(train1$GarageFinish)
```

```
Fin RFn Unf  
352 422 686
```

pppp. GarageCond: En este caso pues se realizó un procedimiento similar al anterior dado que es una variable categórica, por ende se buscó la moda que en este caso resultó ser típico garage condition, cosa que tiene sentido.

```
> table(train1$GarageCond)
```

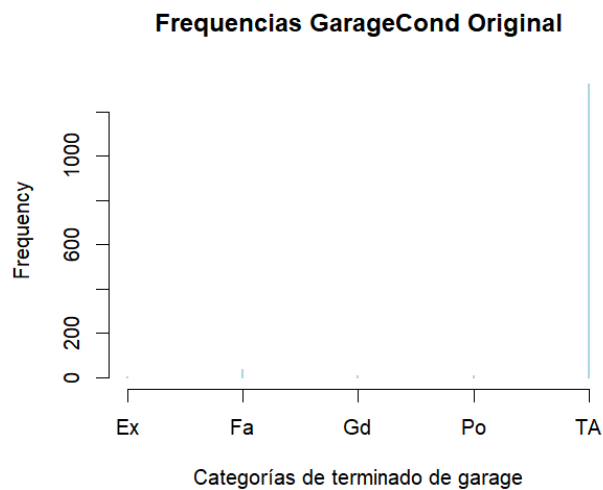
```
Ex Fa Gd Po TA  
2 35 9 7 1326
```

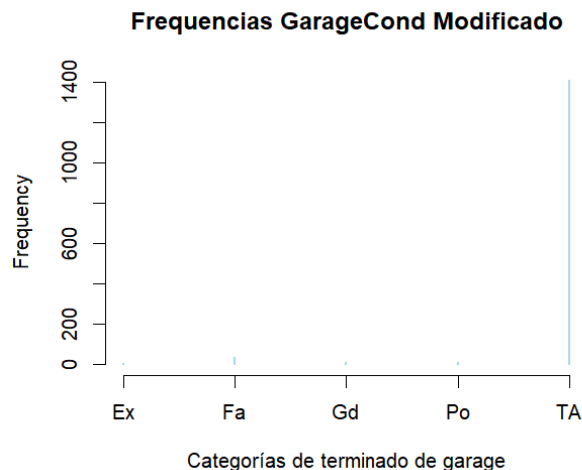
```
> names(table(train1$GarageCond))[which.max(table(train1$GarageCond))]
```

```
[1] "TA"
```

```
> |
```

Y se procedió a rellenar





Por lo que esta diferencia en los garages realmente no es significativa para determinar si la condición de los garajes son realmente influyentes al momento de darle un precio a la casa cómo tal.

Sin embargo, al explorar estas 3 cosas nos dimos cuenta de que realmente era más conveniente desechar estas rows dado que realmente no se tenía datos en esos 81 registros.

```
> nrow(train[!is.na(train$GarageFinish) & !is.na(train$GarageCond) & !is.na(train$GarageQual) & !is.na(train$GarageYrBlt) & !is.na(train$GarageType),])
[1] 1379
```

Por lo que se realizó eso precisamente.

```
> train1<-train1[complete.cases(train1$GarageYrBlt),]
> nrow(train1)
[1] 1379
```

Quedando en total 1379 valores. Ahora, respecto a las demás variables que quedan con datos nulos se podría trabajar tal cuál cómo están sin embargo preferimos modificarlas para que queden lo más limpias posibles

qqqq. Electrical: de manera que es una variable cuantitativa y categoriza el tipo de sistema eléctrico.

```
> nrow(train1[is.na(train1$Electrical),])
[1] 1
```

Para esta variable tan solo 1 valor es na, por ende solo cambiaremos ese valor por la moda.

```
> table(train1$Electrical)

FuseA FuseF FuseP   Mix SBrkr
  81    22     2     1 1272
> names(table(train1$Electrical))[which.max(table(train1$Electrical))]
[1] "SBrkr"
```

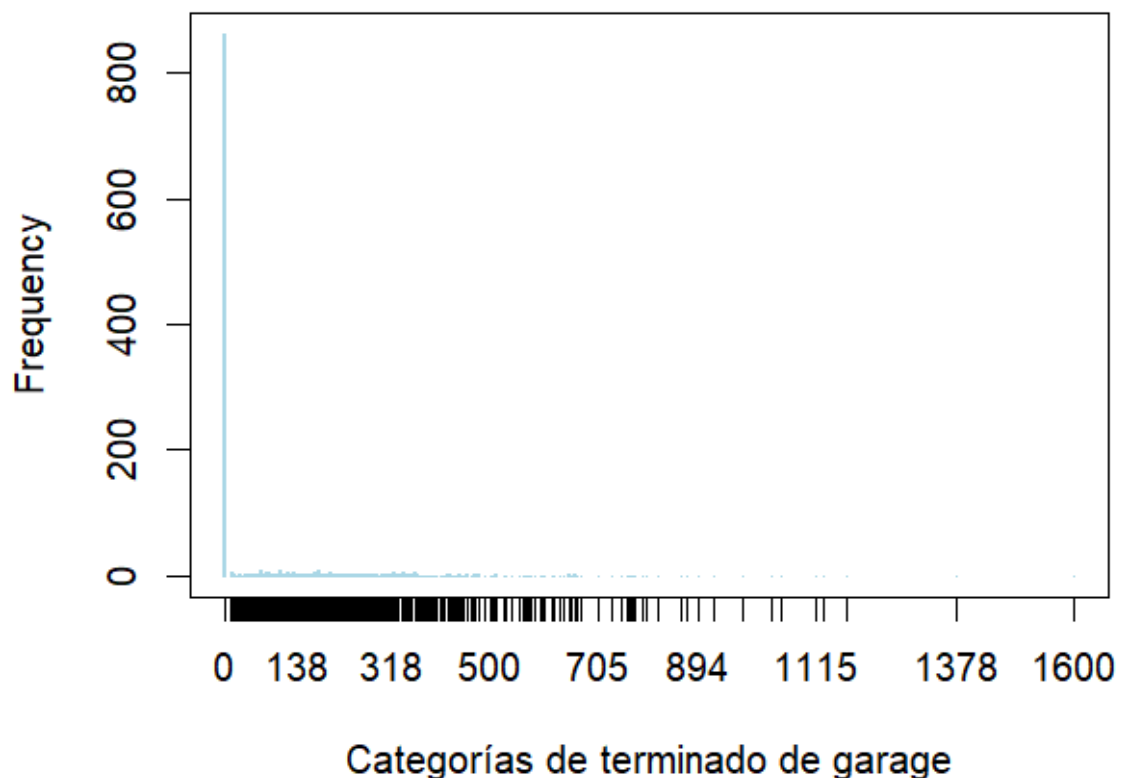
Para esta se encontró que la moda con bastante diferencia es SBrkr para el sistema eléctrico.

Y se realizó el cambio en el único valor con na.

```
> train1$Electrical[is.na(train1$Electrical)] <- names(table(train1$Electrical))[which.max(table(train1$Electrical))]
```

rrrr. MasVnrArea: es una variable cualitativa que indica el área en pies cuadrados del revestimiento de mampostería. Se realizó un procedimiento en el cuál se insertó la media que era 0 en los valores nulos y realmente no hubo una diferencia significativa, sin embargo, lo que se denota en esto, es que realmente no suele realizarse este procedimiento de mampostería, por lo que en realidad no es muy necesario para determinar el precio de una vivienda.

Freuencias MasVnr Modificado



ssss. MasVnrType: es una variable para indicar el tipo de mampostería aplicada, es decir el tipo de revestimiento de mampostería.

```
> nrow(train1[is.na(train1$MasVnrType),])
[1] 8
> distinct(train1, train1$MasVnrType, .keep_all = FALSE)
  train1$MasVnrType
1      BrkFace
2         None
3        Stone
4      BrkCmn
5         <NA>
> table(train1$MasVnrType)

  BrkCmn BrkFace   None   Stone
    15    439    789    128
> names(table(train1$MasVnrType))[which.max(table(train1$MasVnrType))]
[1] "None"
> train1$MasVnrType[is.na(train1$MasVnrType)] <- names(table(train1$MasVnrType))[which.max(table(train1$MasVnrType))]
```

De manera que para esta variable se obtuvo la moda para que reemplazara, de forma que la moda eran aquellas que no tenían mampostería y cómo realmente solo se tenían que reemplazar 8 filas pues no había realmente mucho problema.

tttt. BsmtFinType1: es una variable cuantitativa que indica la calidad del área del sótano acabado.

uuuu. BsmtCond: es una variable cuantitativa que indica la condición general del sótano.

vvvv. BsmtQual: es una variable cuantitativa que indica la altura del sótano.

www. BsmtFinType2: es una variable cuantitativa que indica la calidad del segundo sótano acabado si es que hay.

xxxx. BsmtExposure: es una variable cuantitativa que indica el nivel de los jardines del sótano.

Para estas últimas 5 variables revisamos si sucedía algo similar que en garage, y efectivamente, si se eliminaran todas las filas con na de estas variables quedarían 1347 y para cada variable quedaban entre 1348 y 1349 por lo que se eliminaron las variables nulas de bsmt.

```
> nrow(train1[!is.na(train1$BsmtExposure) & !is.na(train1$BsmtFinType2) & !is.na(train1$BsmtQual) & !is.na(train1$BsmtCond) & !is.na(train1$BsmtFinType1),])  
[1] 1347
```

Dado que las variables eran categóricas, realmente decidimos que no era necesario trabajar los datos para eliminar esos valores

```
> train2<-train1[complete.cases(train1$BsmtExposure , train1$BsmtFinType2 , train1$BsmtQual , train1$BsmtCond , train1$BsmtFinType1),]  
> nrow(train2)  
[1] 1347
```

Por lo que luego de realizar el preprocesamiento de datos se eliminó únicamente un 9.23% de las filas, aunque sí se debieron eliminar columnas que realmente no contribuyen en nada dado que estaban más vacías que llenas.

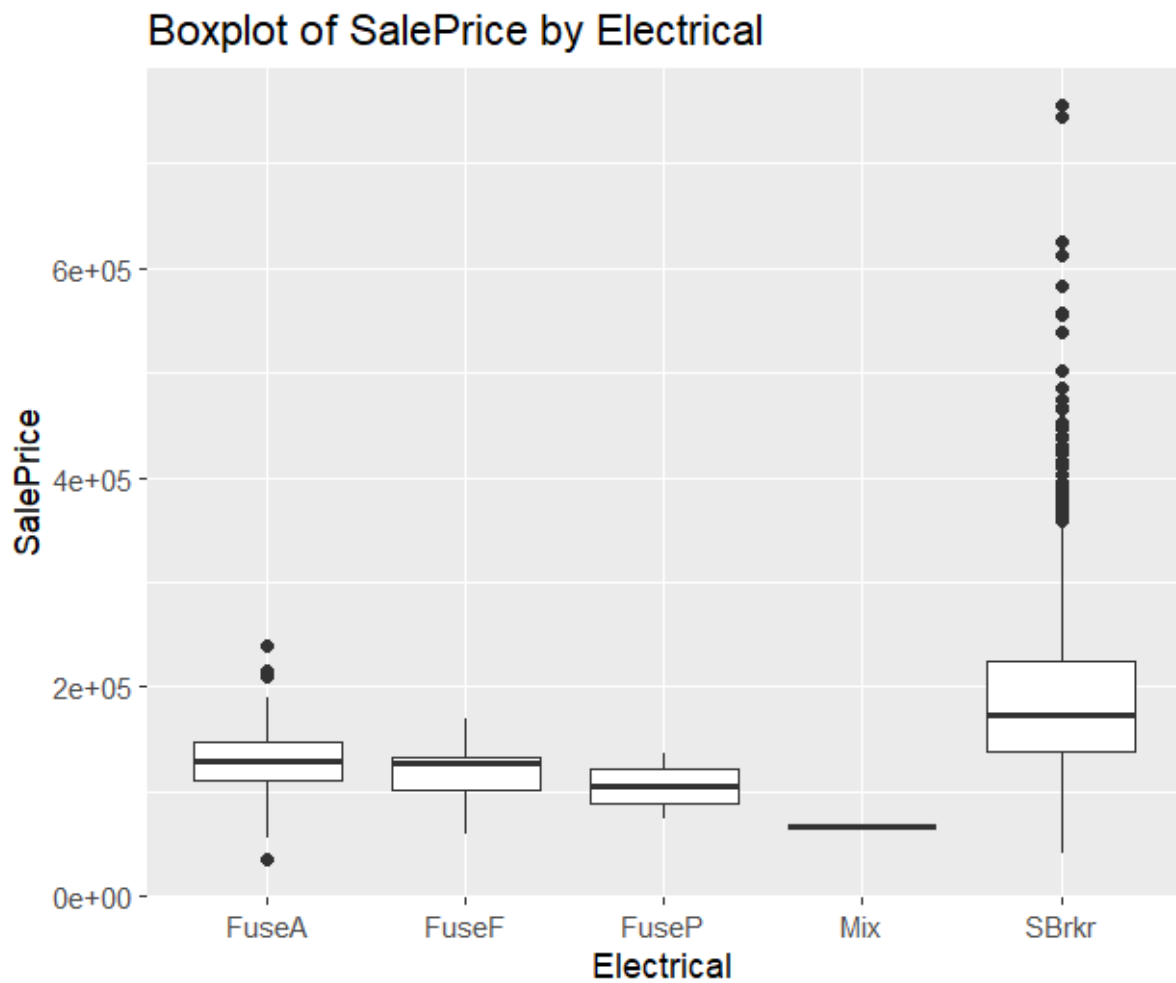
Así mismo, se generó otro reporte de datos mediante el explorador y se obtuvieron métricas interesantes, sin embargo, no se podían leer los datos para los coeficientes de correlación, por ende se tuvo que hacer manualmente.

```
> cor(train2[, sapply(train2, is.numeric)], train2$SalePrice)
      [,1]
Id      -0.02711515
MSSubClass -0.08121471
LotFrontage 0.32950269
LotArea     0.25386523
OverallQual 0.78449935
OverallCond -0.10980074
YearBuilt   0.50368564
YearRemodAdd 0.50074516
MasVnrArea  0.46088559
BsmtFinSF1  0.36228596
BsmtFinSF2 -0.03211421
BsmtUnfSF   0.19045410
TotalBsmtSF 0.60223157
X1stFlrSF   0.60342489
X2ndFlrSF   0.30762411
LowQualFinSF -0.01031102
GrLivArea   0.71003135
BsmtFullBath 0.21190228
BsmtHalfBath -0.03090956
FullBath    0.56674964
HalfBath    0.25999468
BedroomAbvGr 0.16513967
KitchenAbvGr -0.10947402
TotRmsAbvGrd 0.54881935
Fireplaces   0.44336341
GarageYrBlt  0.48137164
GarageCars   0.64104769
GarageArea   0.60875460
WoodDeckSF   0.30588813
OpenPorchSF  0.32746766
EnclosedPorch -0.12712611
X3SsnPorch   0.04153028
ScreenPorch  0.09512312
PoolArea     0.09117677
MiscVal      -0.01720622
MoSold       0.04272306
YrSold       -0.02350432
SalePrice    1.00000000
```

Por lo que se puede observar que aquellas variables cualitativas que poseen cierta correlación con el precio de venta son: OverallQual, GrLivArea, GarageCars, GarageArea, X1stFlrSF y TotalBsmtSF.

Ahora observamos que para cada variable no categórica esta sería su relación con

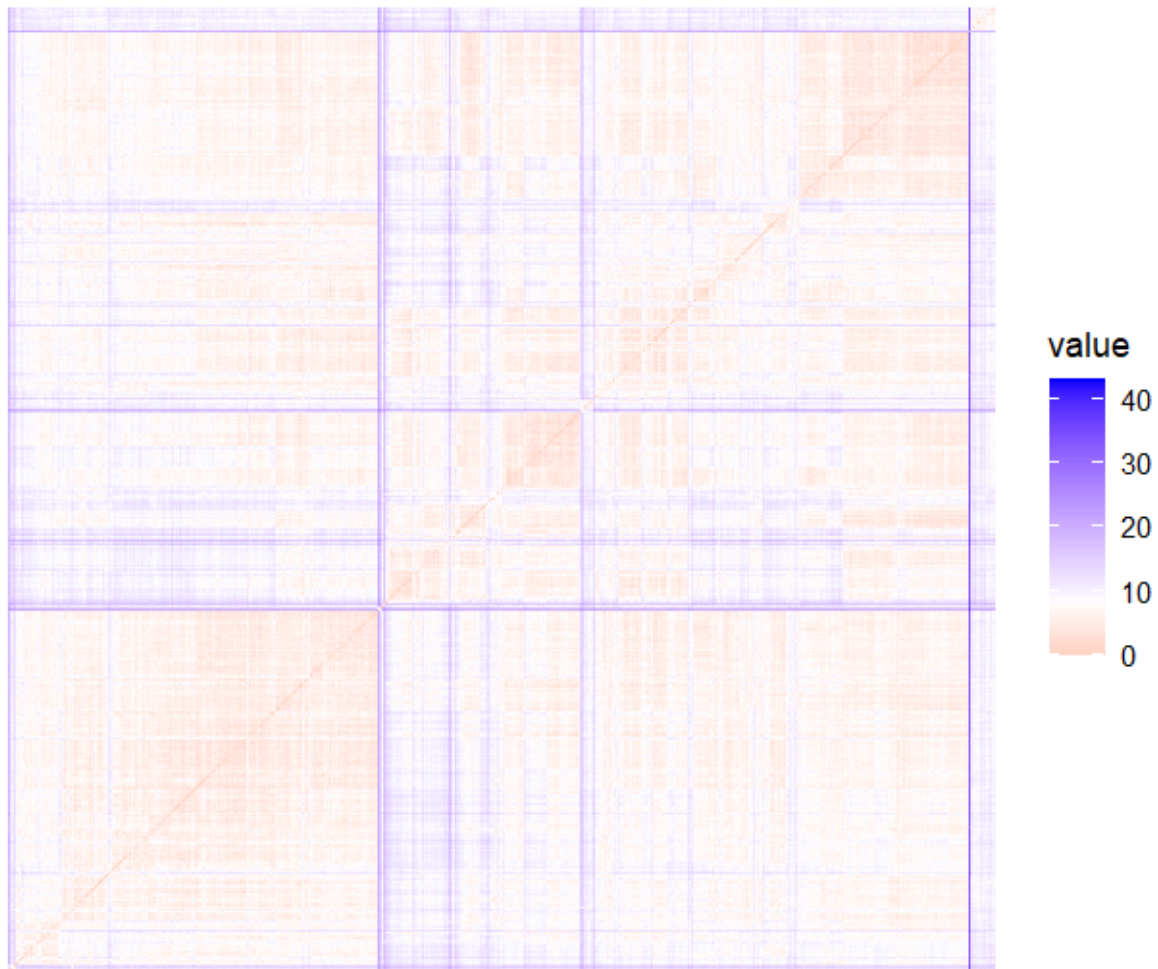
SalePrice:



3. Incluya un análisis de grupos en el análisis exploratorio. Explique las características de los grupos.

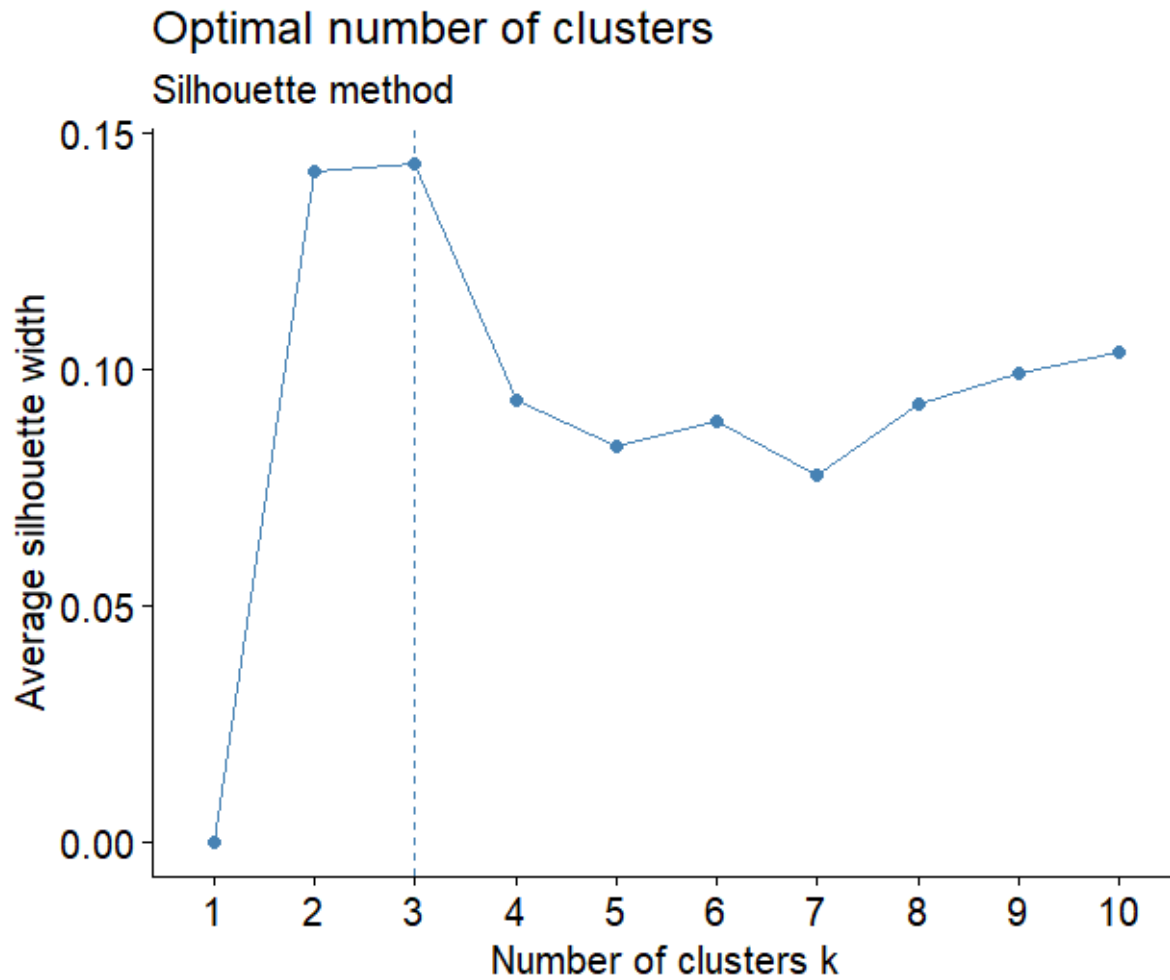
El valor estadístico de hopkins está alejado de 0.5 por lo que los datos no son aleatorios y hay altas posibilidades de que sea factible el agrupamiento.

```
> hopkins(train3[,apply(train2, is.numeric)])  
[1] 1
```

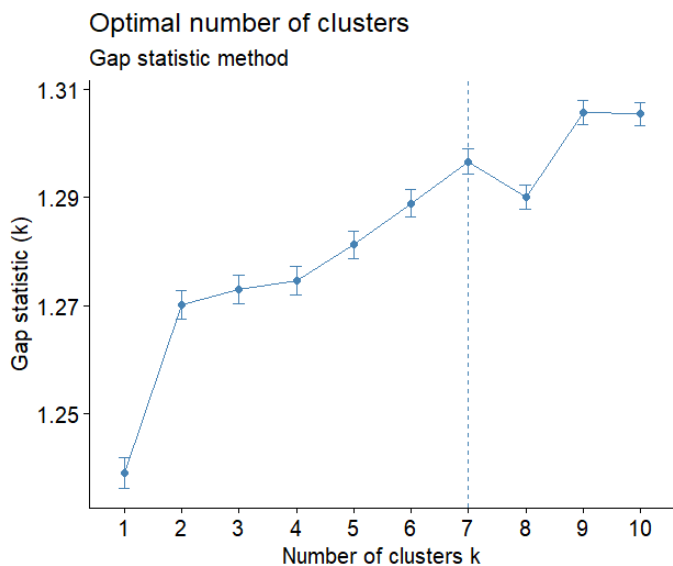


En el gráfico VAT se observa al menos un grupo separado de los otros que se ven menos claro pero corrobora que al menos existe una división entre los grupos por lo que sí es necesario realizar un clustering.

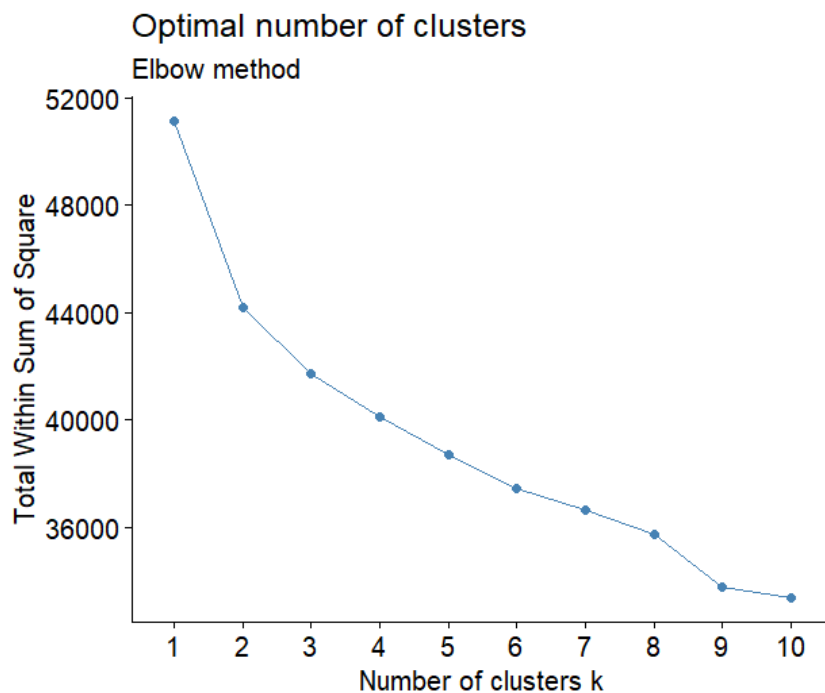
Número óptimo de clústers->bajo el método de la silueta se recomienda 3 lo siguiente:



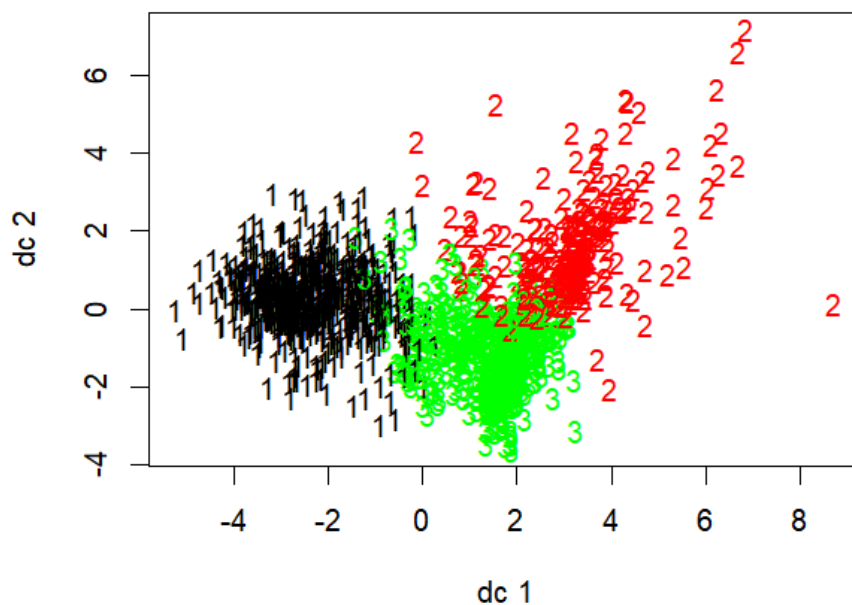
Para el método del gap se recomiendan 7.



Para el método del codo se recomiendan entre 2 y 3



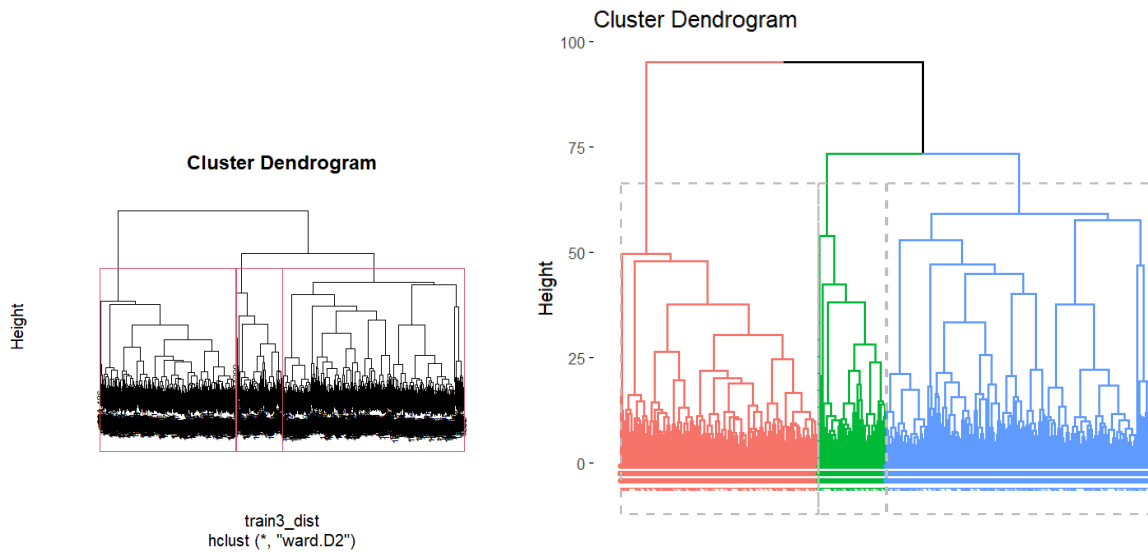
De manera que se eligió 3 como el número ideal de clústers por lo que utilizando k means se encontró que:



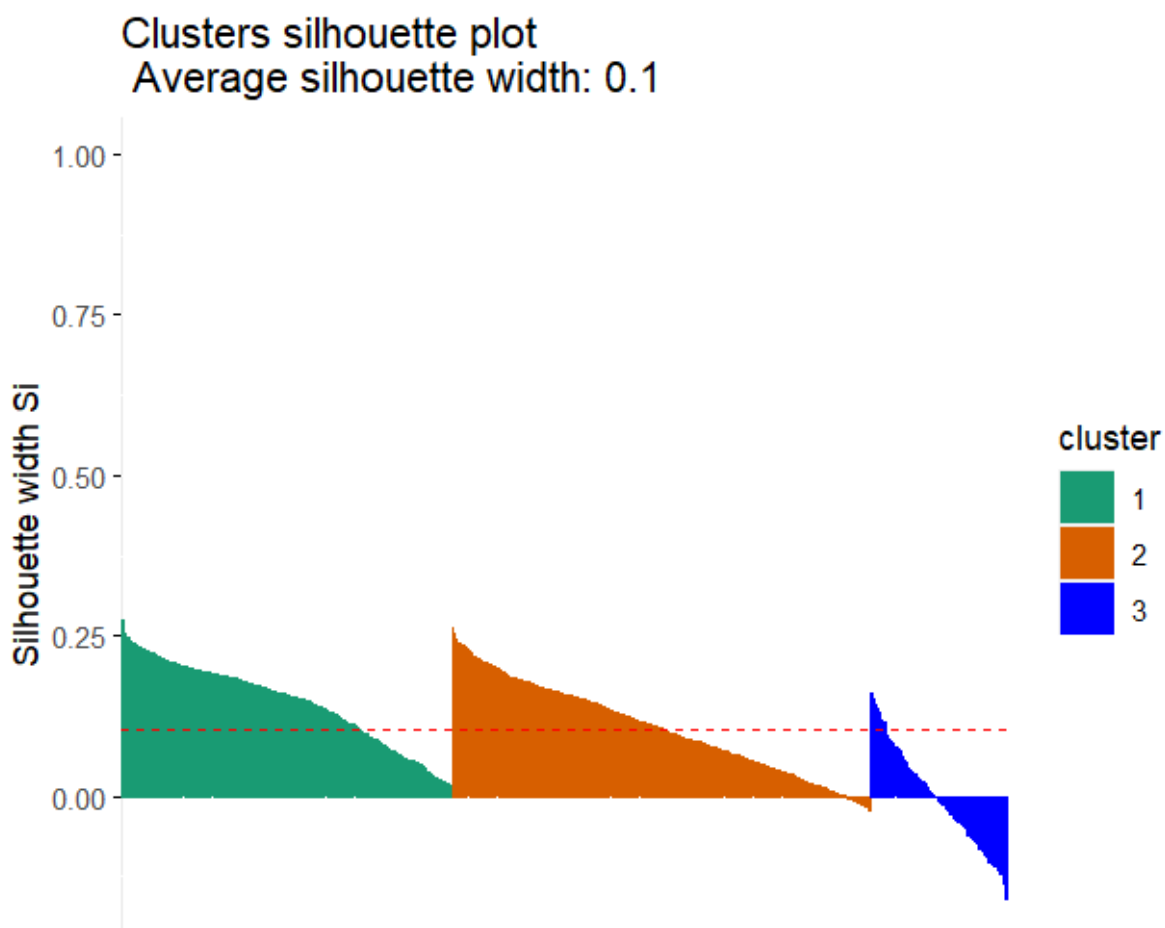
Se poseen 3 grupos bastante bien definidos de los datos.

```
> km$size
[1] 622 237 488
> ### Variabilidad intragrupo
> km$withinss
[1] 18105.33 11080.99 12548.32
```

Así mismo se denota que no están tan desbalanceados en virtud de tener muchos o pocos datos, tal vez el grupo de en medio se encuentra un poco desbalanceado pero nada más. Así mismo, el dendrograma de clúster quedó así:



Y al momento de comprobar con la medida de silueta quedó así.



De manera que se puede observar que realmente no quedó la distribución de clústers tan alejada de la realidad, dado que no hay tantos datos que redistribuir.

4. Divida el set de datos preprocesados en dos conjuntos: Entrenamiento y prueba. Describa el criterio que usó para crear los conjuntos: número de filas de cada uno, estratificado o no, balanceado o no, etc. Si le proveen un conjunto de datos de prueba y tiene suficientes datos, tómelo como de validación, pero haga sus propios conjuntos de prueba.

```
# DIVISION DE DATOS EN TRAIN Y PRUEBA
```{r}
Version sin normalizar
corte <- sample(nrow(train2),nrow(train2)*0.7)
train_copy<-train2[corte,]
test_copy<-train2[-corte,]
```

```{r}
columns <- c(sapply(train_copy, is.numeric))
train_copy<-train_copy[,c(sapply(train_copy, is.numeric))]
test_copy<-test_copy[,c(sapply(test_copy, is.numeric))]
```

```{r}
Normalizacion
train_copy<-as.data.frame(scale(train_copy))
test_copy<-as.data.frame(scale(test_copy))
```

```{r}
train_copy<-train_copy[complete.cases(train_copy),]
test_copy<-test_copy[complete.cases(test_copy),]
```

```{r}
train_copy[is.na(columns),]
```
```

De manera que se hizo una copia de train2 que sería el set de datos ya limpio, sin embargo al momento de normalizar se generaban algunos nulos por lo que se volvió a limpiar, y quedaron

estas proporciones:

```
{r}  
train_copy[is.na(columns),]
```

Description: df [0 x 38]

0 rows | 1-10 of 38 columns

```
{r}  
test_copy[is.na(columns),]
```

Description: df [0 x 38]

0 rows | 1-10 of 38 columns

```
{r}  
nrow(train_copy)  
nrow(test_copy)
```

```
[1] 942  
[1] 405
```

5. Haga ingeniería de características, ¿qué variables cree que puedan ser mejor es predictor es para el precio de las casas? Explique en que basó la selección o no de las variables.

a. Correlación

Lo que se planteó para seleccionar las variables para el precio de las casas fue en primer lugar chequear las variables de las cuáles existe una correlación. Por lo que se puede observar que aquellas variables cualitativas que poseen cierta correlación con el precio de venta son:

| | [,1] |
|---------------|---------------|
| Id | -0.0287518538 |
| MSSubClass | -0.0722375057 |
| LotFrontage | 0.3359873563 |
| LotArea | 0.2337342367 |
| OverallQual | 0.7778155898 |
| OverallCond | -0.0956793242 |
| YearBuilt | 0.4845743325 |
| YearRemodAdd | 0.4902634553 |
| MasVnrArea | 0.4749499042 |
| BsmtFinSF1 | 0.3577889948 |
| BsmtFinSF2 | -0.0316896485 |
| BsmtUnfSF | 0.1892491605 |
| TotalBsmtSF | 0.5855272184 |
| X1stFlrSF | 0.6026378665 |
| X2ndFlrSF | 0.3328113185 |
| LowQualFinSF | -0.0001476608 |
| GrLivArea | 0.7186020852 |
| BsmtFullBath | 0.2128993670 |
| BsmtHalfBath | -0.0323704594 |
| FullBath | 0.5928649057 |
| HalfBath | 0.2658206832 |
| BedroomAbvGr | 0.1927303213 |
| KitchenAbvGr | -0.1044993810 |
| TotRmsAbvGrd | 0.5591601702 |
| Fireplaces | 0.4696777964 |
| GarageYrBlt | 0.4677760938 |
| GarageCars | 0.6488829438 |
| GarageArea | 0.6237946256 |
| WoodDeckSF | 0.3009699856 |
| OpenPorchSF | 0.3001866225 |
| EnclosedPorch | -0.1308749297 |
| X3SsnPorch | 0.0442052021 |
| ScreenPorch | 0.1165145552 |
| PoolArea | 0.1165251983 |
| MiscVal | -0.0213493477 |
| MoSold | 0.0340588417 |
| YrSold | -0.0202119698 |
| SalePrice | 1.0000000000 |

De manera que se puede observar que:

- OverallQual: Calidad general de material y acabado. Tiene sentido que tenga una correlación alta con el precio de venta dado que básicamente estima la calidad media de la casa basado en sus acabados y los materiales utilizados. No es lo mismo una
- GrLivArea: Pies cuadrados de área habitable sobre el nivel del suelo.
- GarageCars: Tamaño del garaje en capacidad para automóviles.
- GarageArea: Tamaño del garaje en pies cuadrados.
- YearBuilt: Fecha original de construcción.
- YearRemodAdd: Fecha de remodelación.
- X1stFlrSF: Pies cuadrados del primer piso.
- TotalBsmtSF: Pies cuadrados totales del área del sótano

Se puede observar que todas estas variables tienen una correlación relativamente alta con la variable train\$SalePrice. Tanto para los normalizados como para los no normalizados.

Debido a que overallQual tiene la mejor correlación respecto a SalePrice. Sin embargo, esta variable es categórica, dado que realmente categoriza los materiales con los que se

construyó la casa, por ende, se tomará la siguiente variable que tenga mejor coeficiente de correlación, que en tal caso es GrLivArea, esta será la variable que se utilizará para modelar la regresión lineal simple.

6. Todos los resultados deben ser reproducibles por lo que debe fijar que los conjuntos de entrenamiento y prueba sean los mismos siempre que se ejecute el código.

Es por ello que se utilizará una copia de los datos ya preprocesados, de manera que no haya algún problema al momento de utilizarlos. Así mismo, se utilizarán los normalizados para este propósito y así no tener problemas en caso de tener que hacer un modelo con regularizaciones.

7. Seleccione una de las variables y haga un modelo univariado de regresión lineal para predecir el precio de las casas. Analice el modelo (resumen, residuos, resultados de la predicción). Muéstrelo gráficamente.

De manera que el modelo basado únicamente en GrLivArea queda con los siguientes dos resúmenes, no normalizado y normalizado respectivamente:

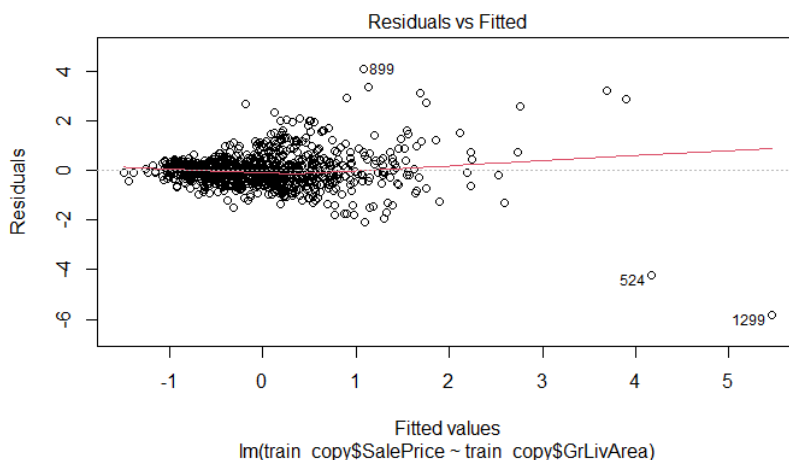
```
call:
lm(formula = train_copy$SalePrice ~ train_copy$GrLivArea, data = train_copy)

Residuals:
    Min       1Q   Median       3Q      Max
-5.8120 -0.3527 -0.0356  0.2583  4.0802

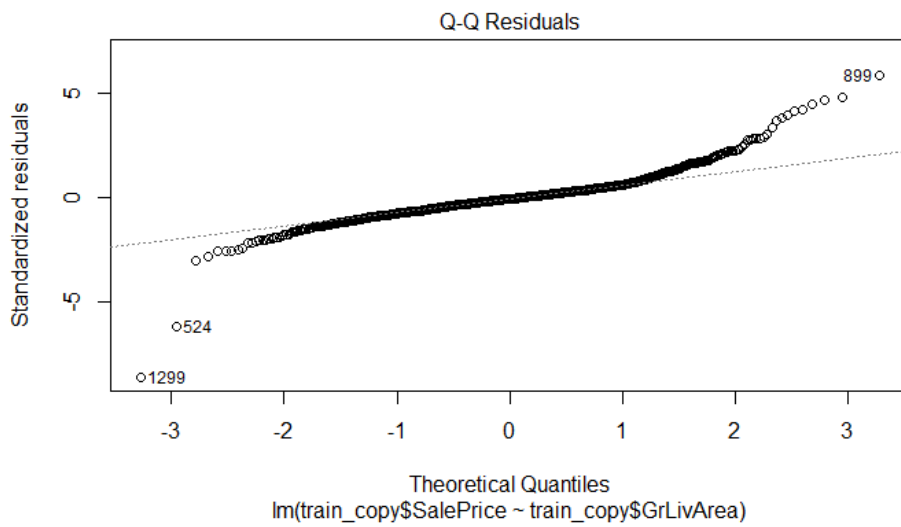
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.609e-18  2.267e-02   0.00    1
train_copy$GrLivArea  7.186e-01  2.268e-02  31.68 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6958 on 940 degrees of freedom
Multiple R-squared:  0.5164,    Adjusted R-squared:  0.5159
F-statistic: 1004 on 1 and 940 DF,  p-value: < 2.2e-16
```

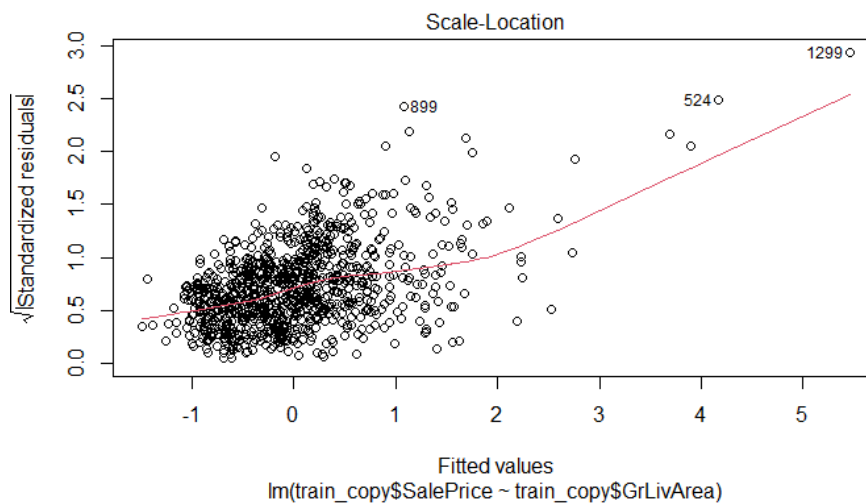
De manera que se puede observar que no tiene un error estándar muy alto respecto a la variable (esto puede ser debido a que está normalizado), aunque en definitiva, no tiene un buen ajuste dado que su R^2 es de 0.5164.



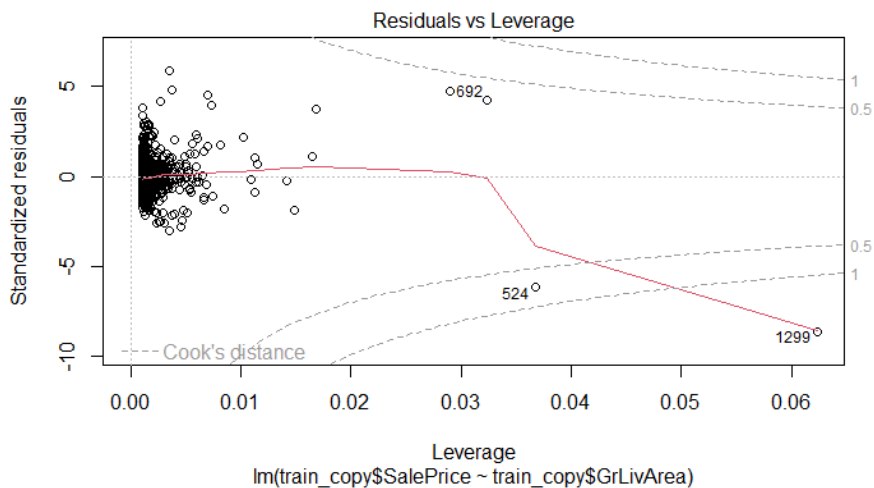
De manera que se puede observar que sí sigue un patrón lineal aunque se concentran muchos de los datos en una cierta región, esto puede ser debido a que existen puntos que están algo alejados de la región principal.



Se observa que los residuos no tiene una distribución normal aunque hay algunos sectores en los cuáles pareciera que sí sigue una distribución normal.



Así mismo, se puede observar que la variable utilizada para la predicción es inválida dado que los residuos carecen de cumplir homocedasticidad al estar concentrados en un sector en específico y no estar distribuidos a través del mapa.



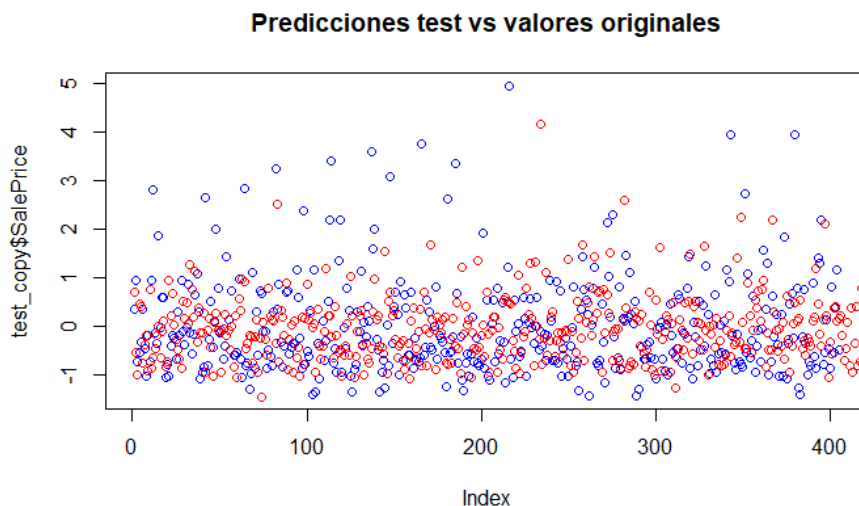
Finalmente se observan que existen puntos que son influyentes ya que por ejemplo el 692, 524 y 1299 realmente están muy alejados de lo que debería de ser.

Por lo que se puede observar que realmente los datos no son realmente representados por el modelo lineal, indicado tanto por el R^2 como el gráfico, al momento de realizar el test de lillie se obtuvieron los siguientes resultados:

Lilliefors (Kolmogorov-Smirnov) normality test

```
data: SNS_Model$residuals
D = 0.10965, p-value < 2.2e-16
```

Por lo que se rechaza la hipótesis nula y por ende los resultados no siguen una distribución normal.



Finalmente se descarta este modelo ya que cómo se puede observar, existe una diferencia muy grande entre las predicciones y los valores originales dónde los valores originales son los azules y las predicciones los rojos. Y finalmente se observó que el RMSE era de 1.261685 lo que denota el mal modelo dado que tiene 1.26 medio.

8. **Haga un modelo de regresión lineal con todas las variables numéricas para predecir el precio de las casas. Analice el modelo (resumen, residuos, resultados de la predicción).**

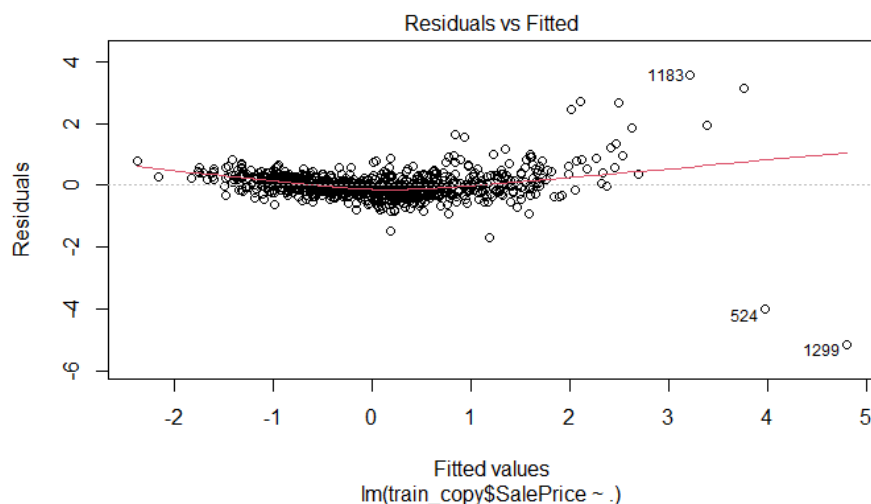
Muestre el modelo gráficamente.

De manera que luego de ejecutar se tuvo el siguiente resultado para el modelo de regresión lineal multivariado tanto para el no normalizado cómo para el normalizado respectivamente:

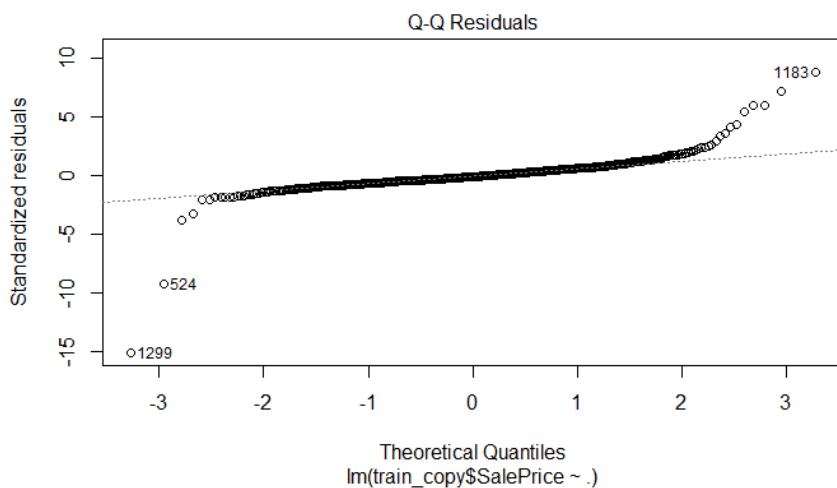
```
GarageYrBlt    -3.699e-02  3.263e-02  -1.134  0.257175
GarageCars      1.364e-01  3.066e-02   4.449  9.69e-06 ***
GarageArea      6.895e-03  3.178e-02   0.217  0.828265
WoodDeckSF      2.644e-02  1.674e-02   1.579  0.114601
OpenPorchSF    -2.905e-02  1.702e-02  -1.706  0.088326 .
EnclosedPorch   1.589e-02  1.709e-02   0.930  0.352680
X3SsnPorch      9.163e-03  1.532e-02   0.598  0.549975
ScreenPorch     3.301e-02  1.605e-02   2.057  0.039977 *
PoolArea       -6.835e-03  1.623e-02  -0.421  0.673723
MiscVal        -3.262e-03  1.526e-02  -0.214  0.830825
MoSold         -1.413e-04  1.547e-02  -0.009  0.992713
YrSold         -1.250e-02  1.552e-02  -0.805  0.420907
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4621 on 906 degrees of freedom
Multiple R-squared:  0.7944,    Adjusted R-squared:  0.7865 
F-statistic: 100 on 35 and 906 DF,  p-value: < 2.2e-16
```

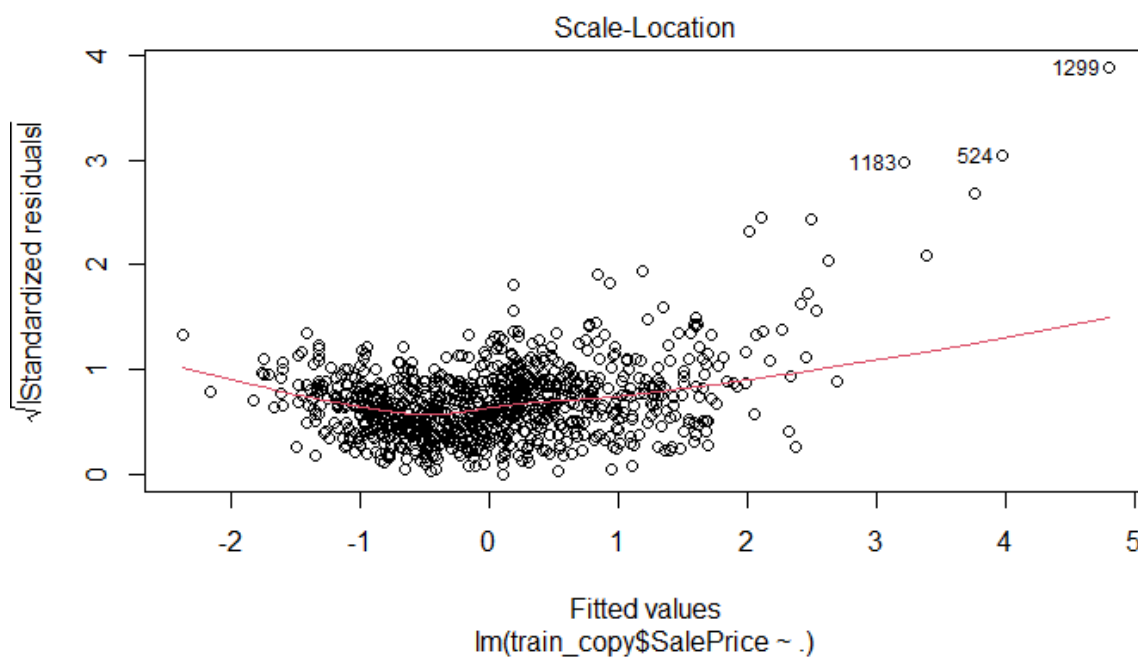
De manera que se observa que tiene un R ajustado de 0.7865 por lo que es bastante decente y explicaría bastante bien el comportamiento, así mismo, no tiene un error estándar muy alto cómo se observa y tiene varias variables que describen correctamente el modelo.



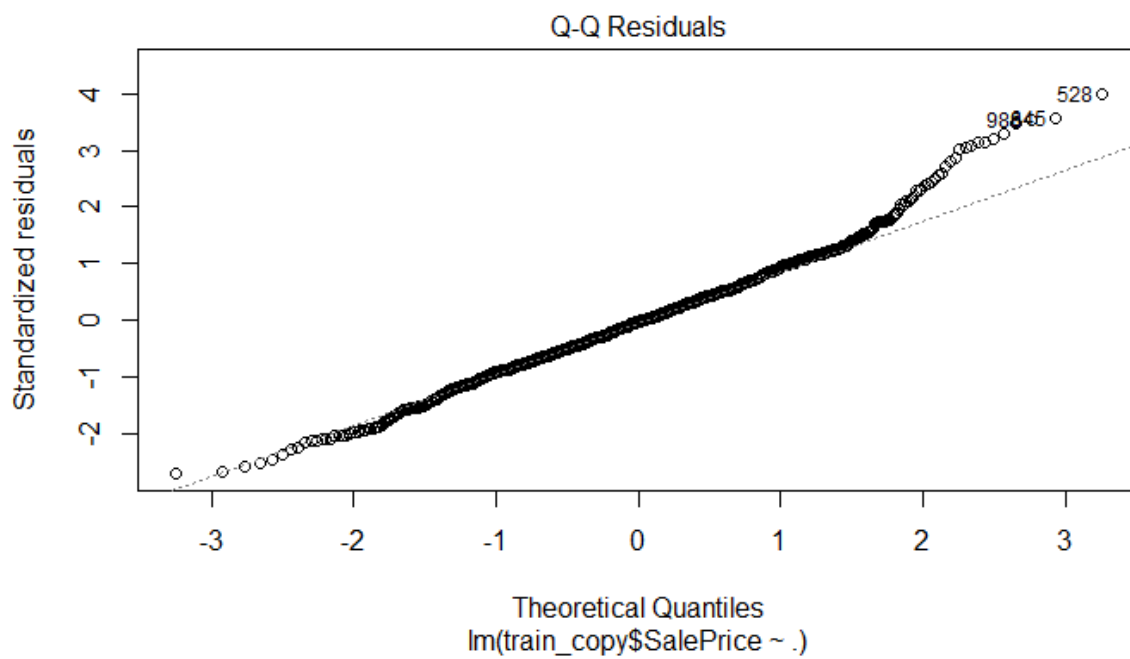
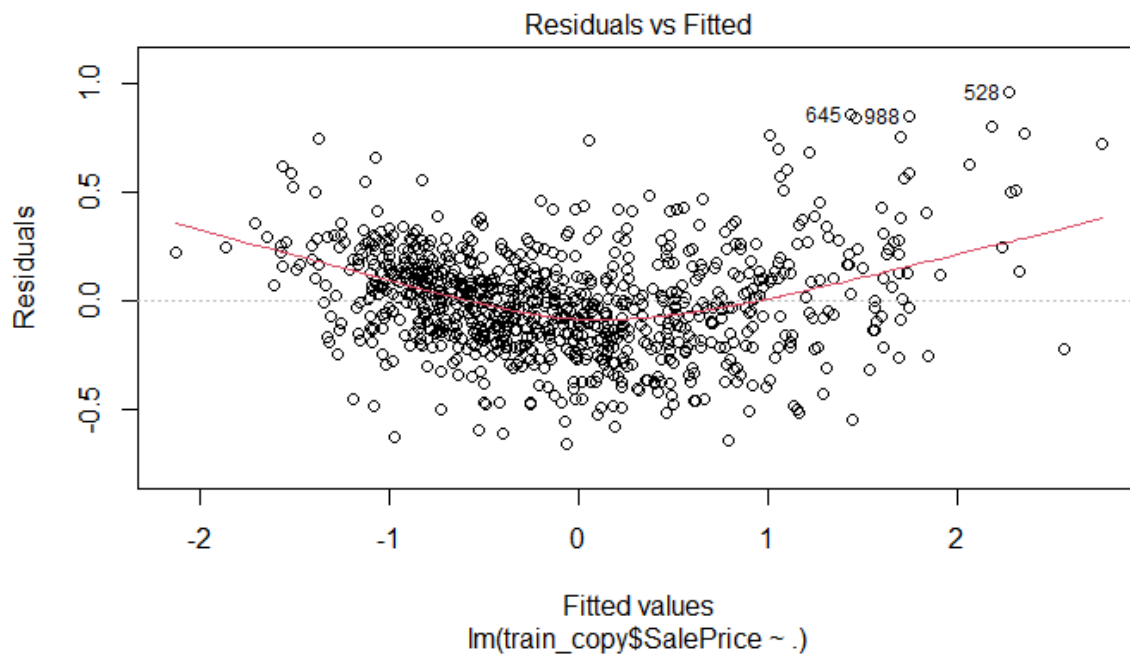
En primer lugar, parece que sigue una forma lineal por lo que hacer una regresión lineal no es una mala idea realmente. Y está un poco concentrado en cierta región, sin embargo, puede que se vea así porque existen puntos palanca que puedan estar moviendo el modelo.

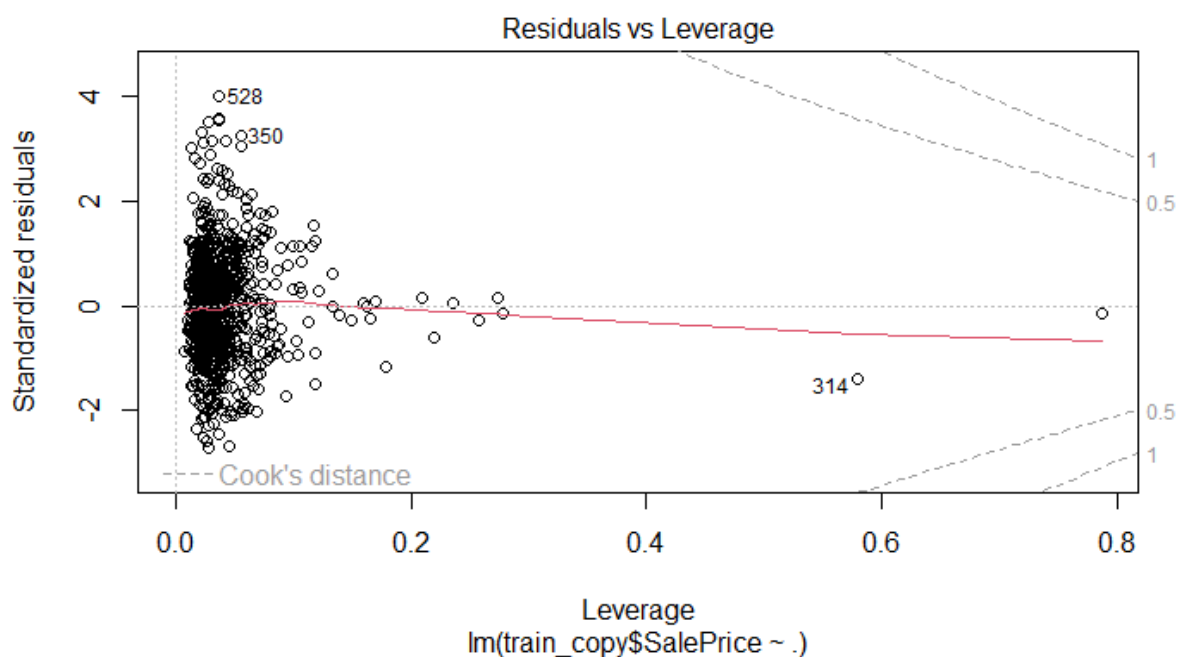
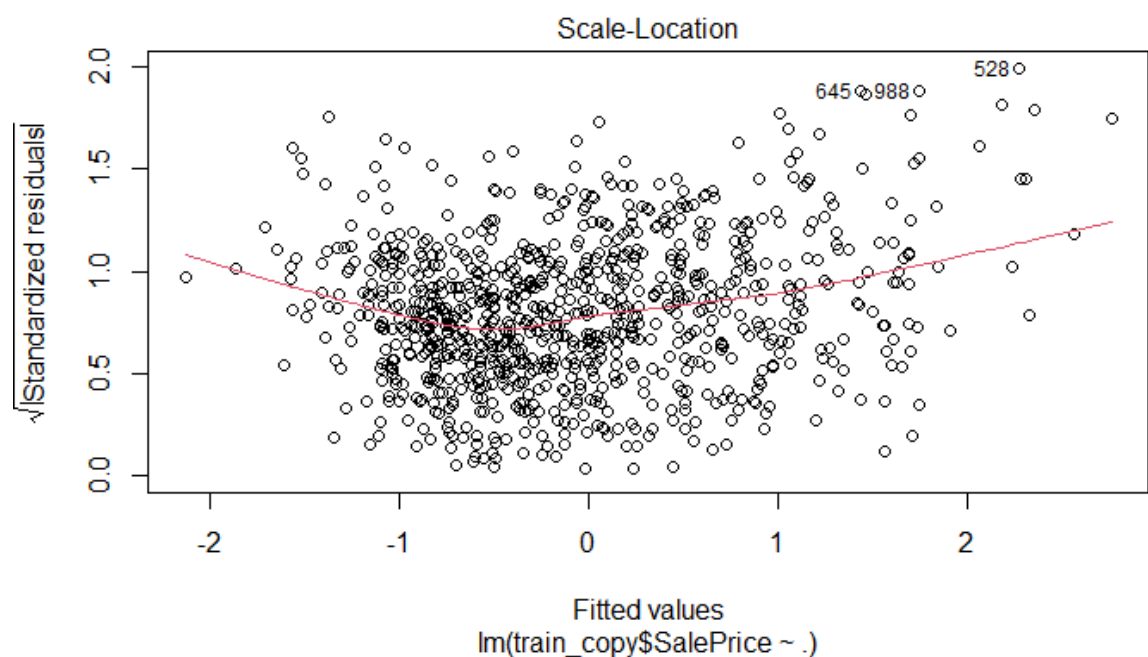


Así mismo, se observa que el modelo parece ser en su mayoría lineal tal cómo se puede observar en el gráfico QQ.



Se puede observar que no es homocedasticidad, aunque es probable que tenga que ver con el hecho de que hayan puntos palanca por lo que se realizará esa corrección.



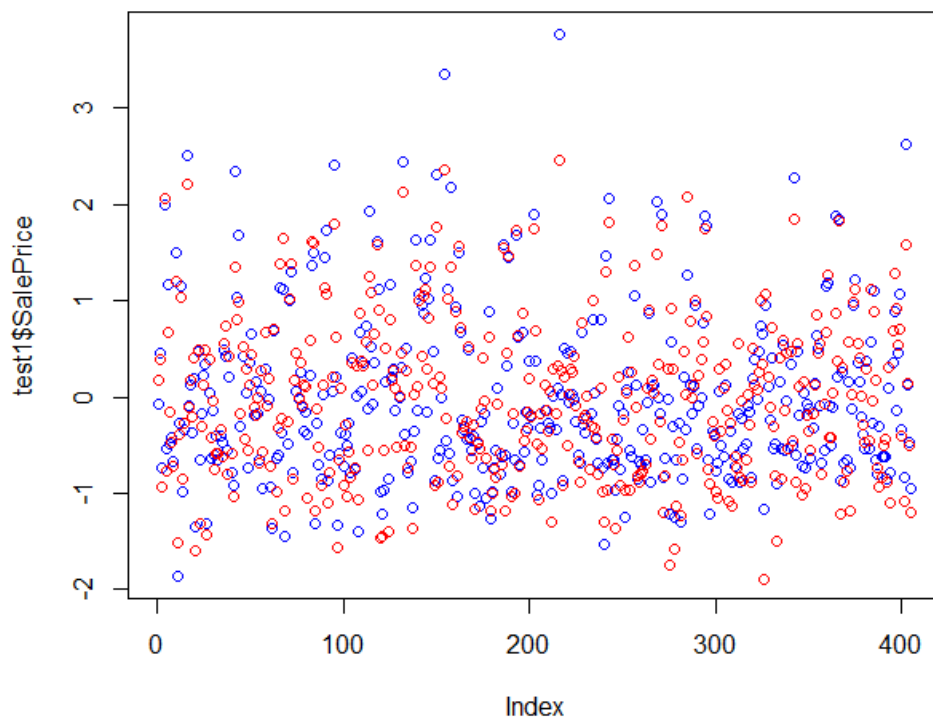


De manera que se puede observar que si es homocedastico además de que luce como una normal en el QQ y aún existen algunos puntos palanca.

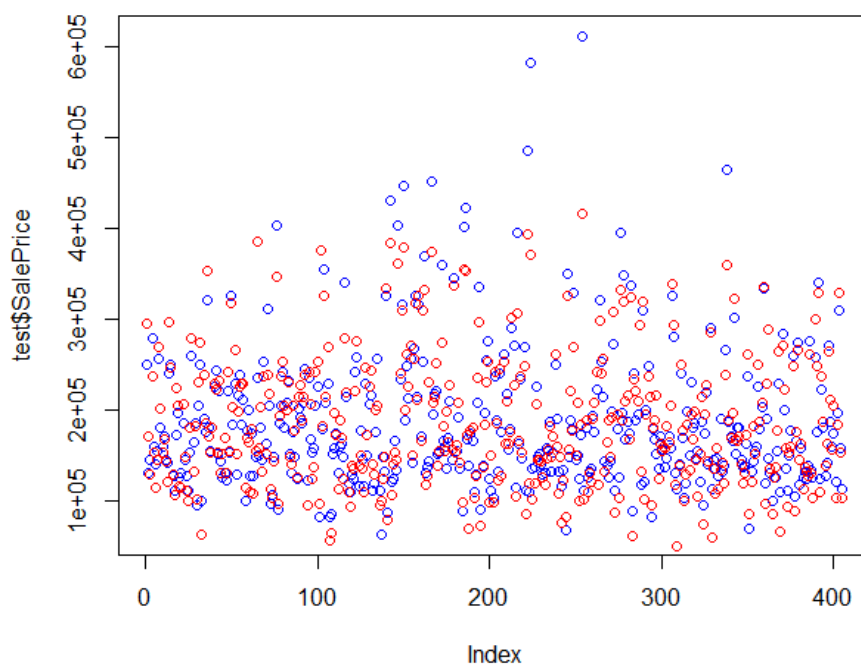
```
lag Autocorrelation D-w Statistic p-value
1 -0.03027743 2.0603 0.372
Alternative hypothesis: rho != 0
```

Así mismo, se demuestra mediante el test de Durbin Watson que las variables no tienen autocorrelación por lo que sí es un modelo válido.

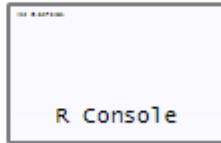
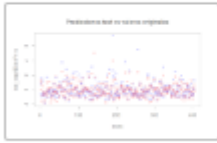
Predicciones vs valores originales normalizados



Predicciones vs valores originales



```
RMSE(pSNM_Model, test_copy$SalePrice)
```



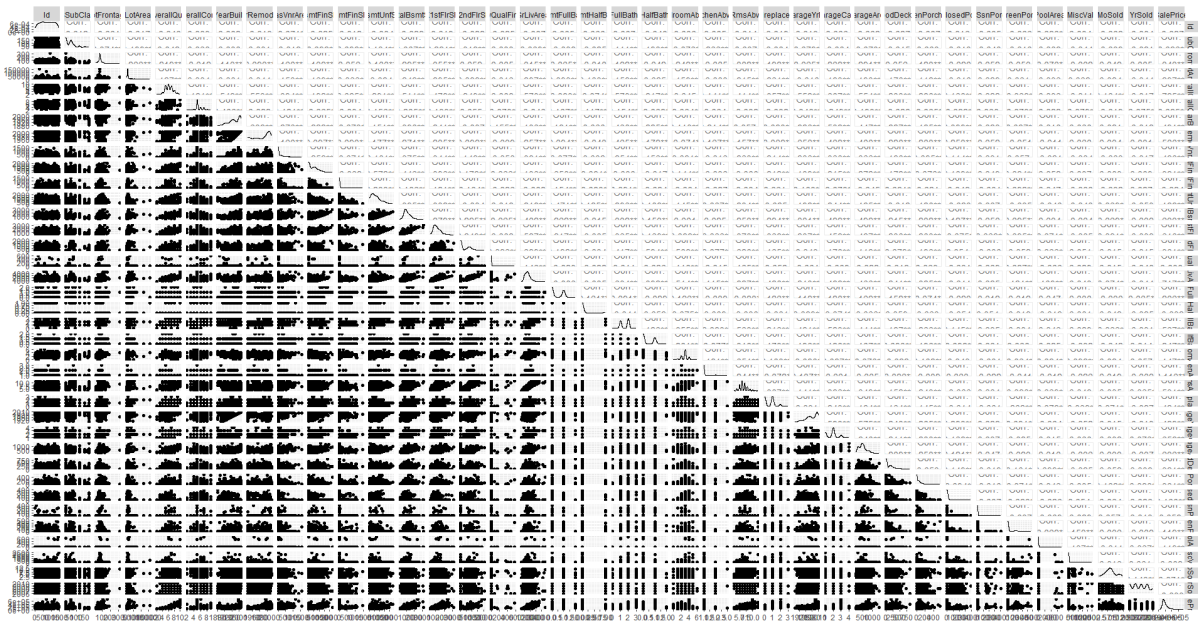
```
[1] 0.6375481
```

Realizando el test de lillie se rechaza el hecho de que los residuos sean normales ya que son menores a 0.05

```
data: SNM_Model$residuals
D = 0.10891, p-value < 2.2e-16
```

Por lo que el modelo en definitiva no es el mejor para predecir y es un modelo inválido ya que es homocedastico, tiene un valor medio cuadrado decente pero sus residuos no son normales.

- Analice el modelo. Determine si hay multicolinealidad entre las variables, y cuáles son las que aportan al modelo. Haga un análisis de correlación de las características del modelo y especifique si el modelo se adapta bien a los datos. Explique si hay sobreajuste (overfitting) o no. En caso de existir sobreajuste, haga otro modelo que lo corrija.



Aquí no se logra observar nada, sin embargo, con el test de durbin se obtuvo lo ya dicho, no existe multicolinealidad de las variables.

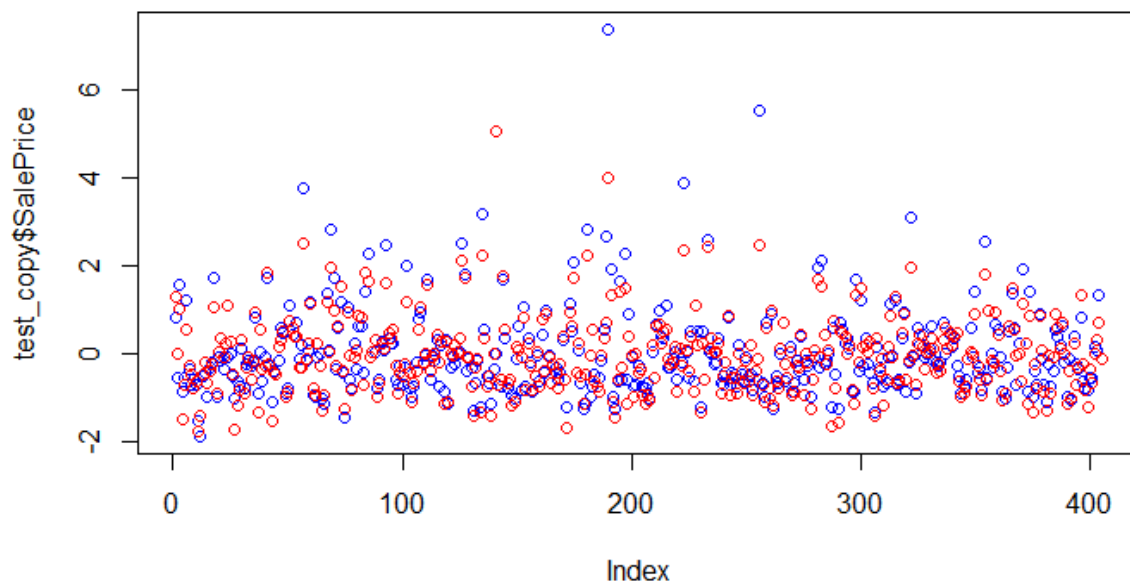
```
lag Autocorrelation D-W Statistic p-value
1 -0.03027743 2.0603 0.372
Alternative hypothesis: rho != 0
```

Así mismo las variables que aportan al modelo son las siguientes:

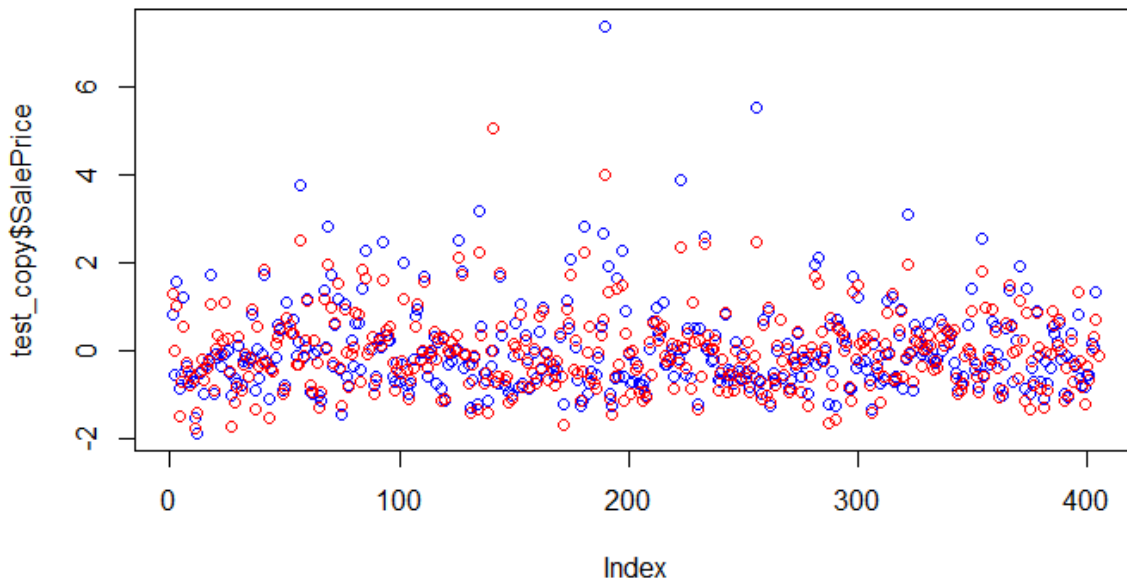
| | Estimate | Std. Error | t value | Pr(> t) | |
|--------------|------------|------------|---------|----------|-----|
| (Intercept) | -1.080e+06 | 1.587e+05 | -6.803 | 2.10e-11 | *** |
| Id | 5.453e+00 | 2.859e+00 | 1.907 | 0.056913 | . |
| MSSubClass | -6.228e+01 | 3.716e+01 | -1.676 | 0.094150 | . |
| LotFrontage | 1.174e+02 | 7.255e+01 | 1.618 | 0.106128 | |
| LotArea | 9.123e-01 | 3.220e-01 | 2.833 | 0.004731 | ** |
| OverallQual | 1.552e+04 | 1.603e+03 | 9.686 | < 2e-16 | *** |
| OverallCond | 4.588e+03 | 1.455e+03 | 3.154 | 0.001677 | ** |
| YearBuilt | 2.474e+02 | 7.114e+01 | 3.477 | 0.000536 | *** |
| YearRemodAdd | 2.664e+02 | 9.093e+01 | 2.929 | 0.003499 | ** |
| MasVnrArea | 2.342e+01 | 7.804e+00 | 3.001 | 0.002777 | ** |
| BsmtFinSF1 | 7.867e+01 | 8.556e+00 | 9.195 | < 2e-16 | *** |
| BsmtFinSF2 | 6.006e+01 | 1.102e+01 | 5.451 | 6.83e-08 | *** |
| BsmtUnfSF | 5.590e+01 | 8.435e+00 | 6.627 | 6.58e-11 | *** |
| X1stFlrSF | 1.636e+01 | 9.123e+00 | 1.793 | 0.073325 | . |
| X2ndFlrSF | 5.153e+01 | 5.435e+00 | 9.481 | < 2e-16 | *** |
| BedroomAbvGr | -1.054e+04 | 2.377e+03 | -4.435 | 1.06e-05 | *** |
| KitchenAbvGr | -3.102e+04 | 7.706e+03 | -4.026 | 6.26e-05 | *** |
| TotRmsAbvGrd | 5.356e+03 | 1.573e+03 | 3.404 | 0.000699 | *** |
| Fireplaces | 6.409e+03 | 2.373e+03 | 2.701 | 0.007071 | ** |
| GarageCars | 1.029e+04 | 2.639e+03 | 3.899 | 0.000105 | *** |
| WoodDeckSF | 1.685e+01 | 1.085e+01 | 1.553 | 0.120915 | |
| PoolArea | 4.891e+01 | 2.725e+01 | 1.795 | 0.073029 | . |

 signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

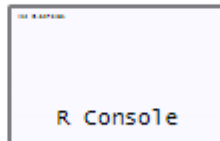
Predicciones Train vs valores originales



Predicciones test vs valores originales



```
RMSE(psNM_Model, test_copy$SalePrice)
```

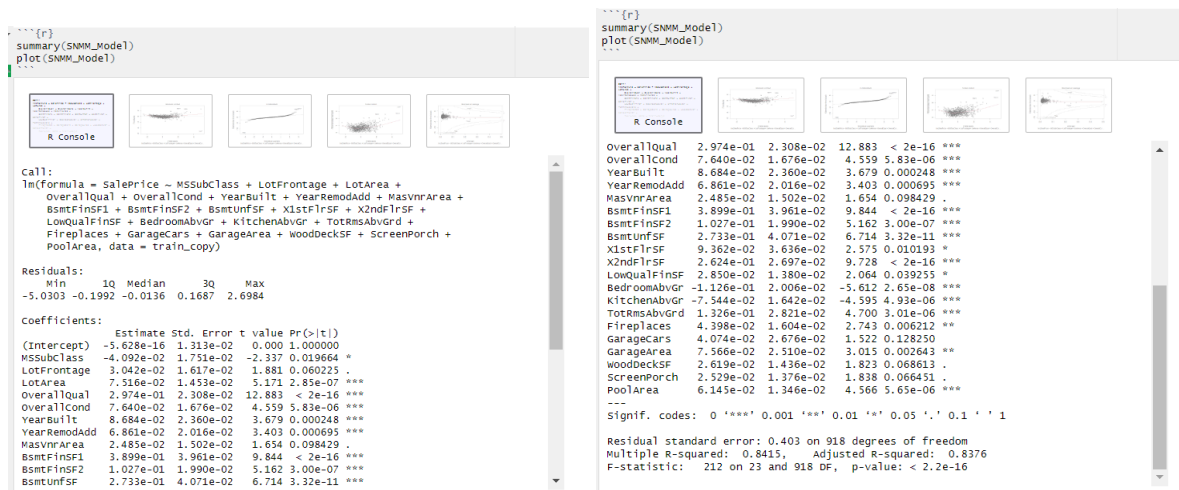


```
[1] 0.6375481
```

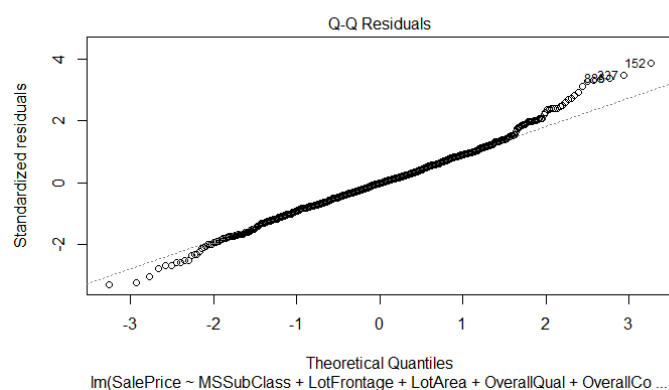
```
Residual standard error: 0.2459 on 849 degrees of freedom  
Multiple R-squared: 0.9182, Adjusted R-squared: 0.9149  
F-statistic: 280.1 on 34 and 849 DF, p-value: < 2.2e-16
```

El modelo probablemente está en overfitting y es inválido por lo ya dicho, sus residuos no siguen una distribución normal. Aunque realmente tal y cómo se observa en la gráfica la diferencia entre los valores originales en el test es bastante distinta, a su vez en el train tampoco es del todo exacta aunque si se puede observar que sigue una tendencia algo exacta en algunas de las regresiones cómo se observa en la gráfica de las predicciones del test.

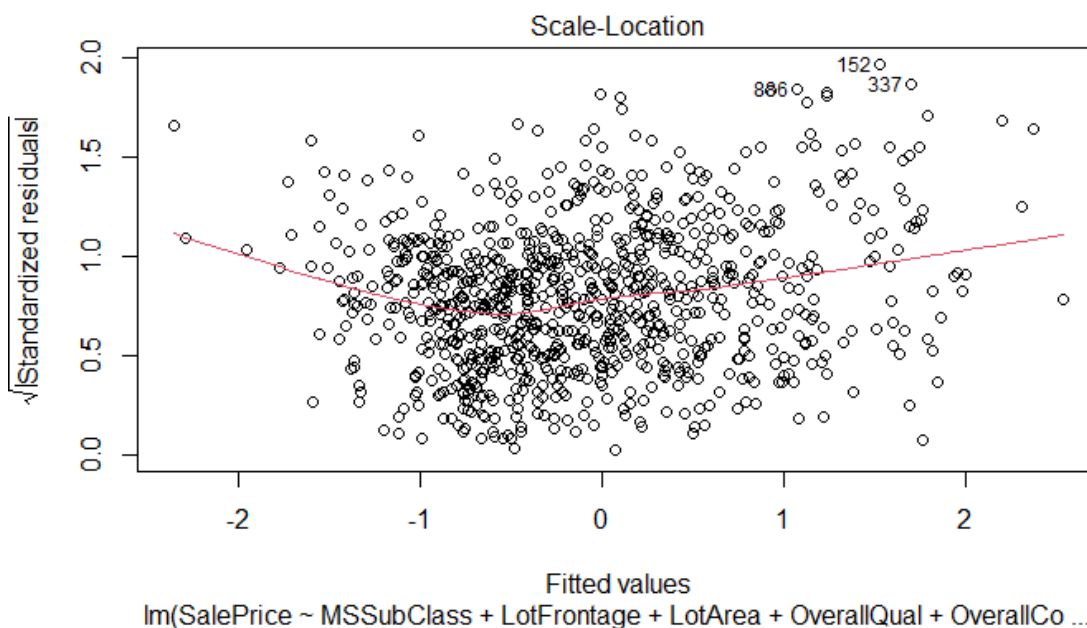
10. Si tiene multicolinealidad o sobreajuste, haga un modelo con las variables que sean mejores predictoras del precio de las casas. Determine la calidad del modelo realizando un análisis de los residuos. Muéstrela gráficamente.



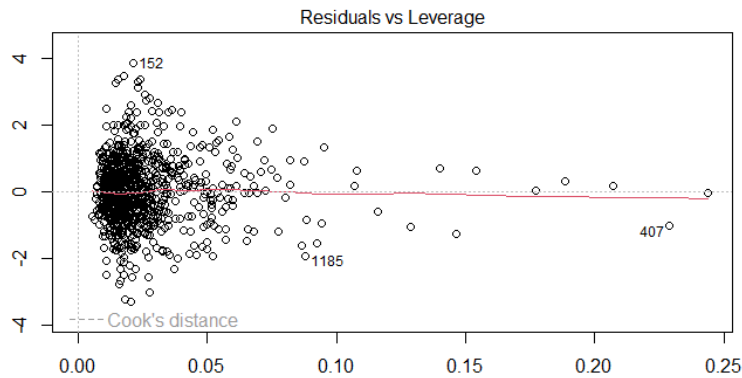
De manera que se hizo un modelo que seleccionó las características que mejor explican los datos.



Sigue una tendencia normal, tal cómo se observa en el gráfico de qq.



Así mismo, se observa que cumple con la homocedasticidad.

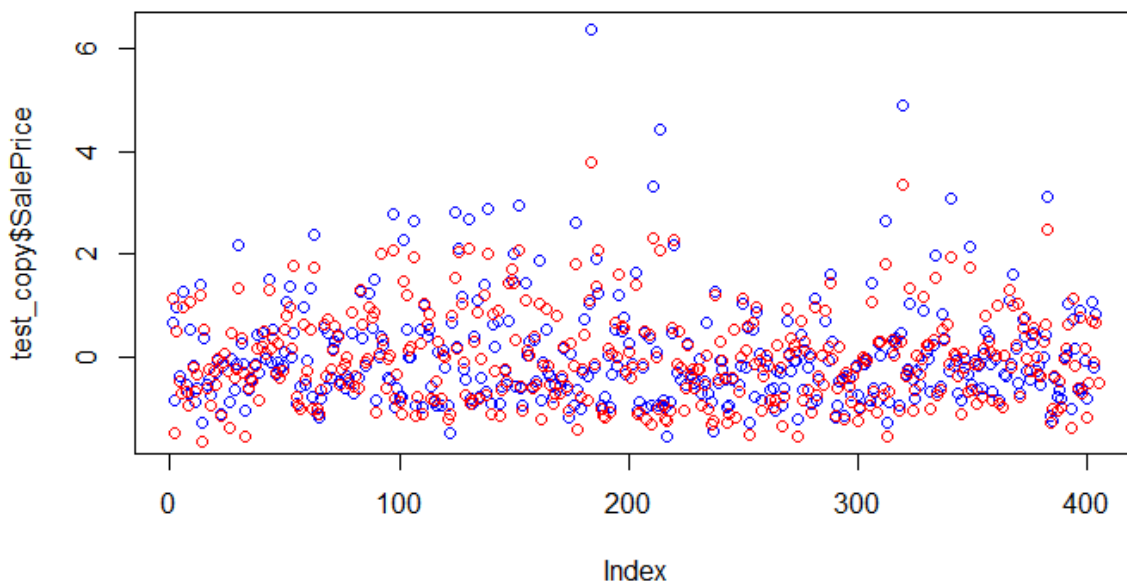


Se ve una significativa menor cantidad de palancas.

```
lag Autocorrelation D-w Statistic p-value
1      0.0197405      1.95968      0.546
Alternative hypothesis: rho != 0
```

No existe la autocorrelación.

Predicciones test vs valores originales



Y se observa que realmente explica algo mejor los datos en la gráfica respectiva.

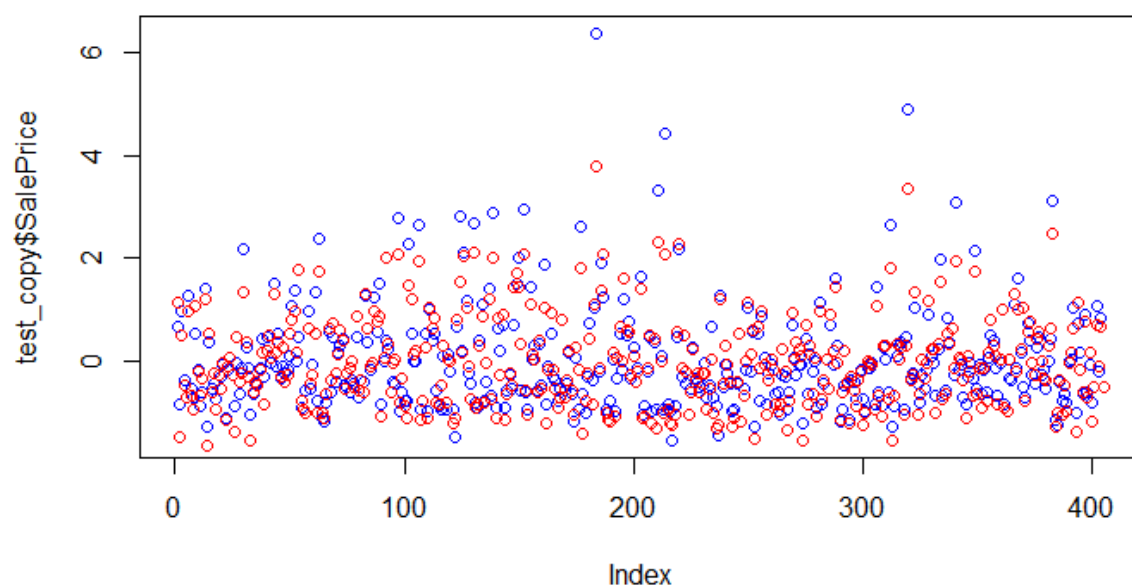
Nota:

Se realizaron otros 2 modelos pero esos no se cubrirán aquí aunque igual estarán en la comparación final.

11. Utilice cada modelo con el conjunto de prueba y determine la eficiencia del algoritmo para predecir el precio de las casas. ¿Qué tan bien lo hizo? ¿Qué medidas usó para determinar la calidad de la predicción?

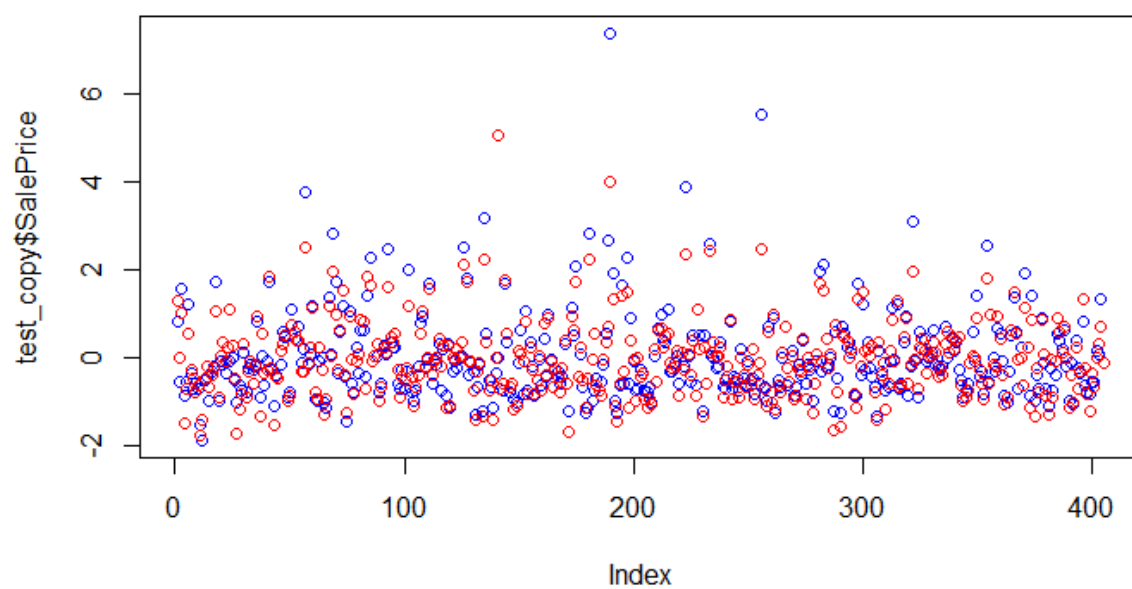
Predictores seleccionados

Predicciones test vs valores originales



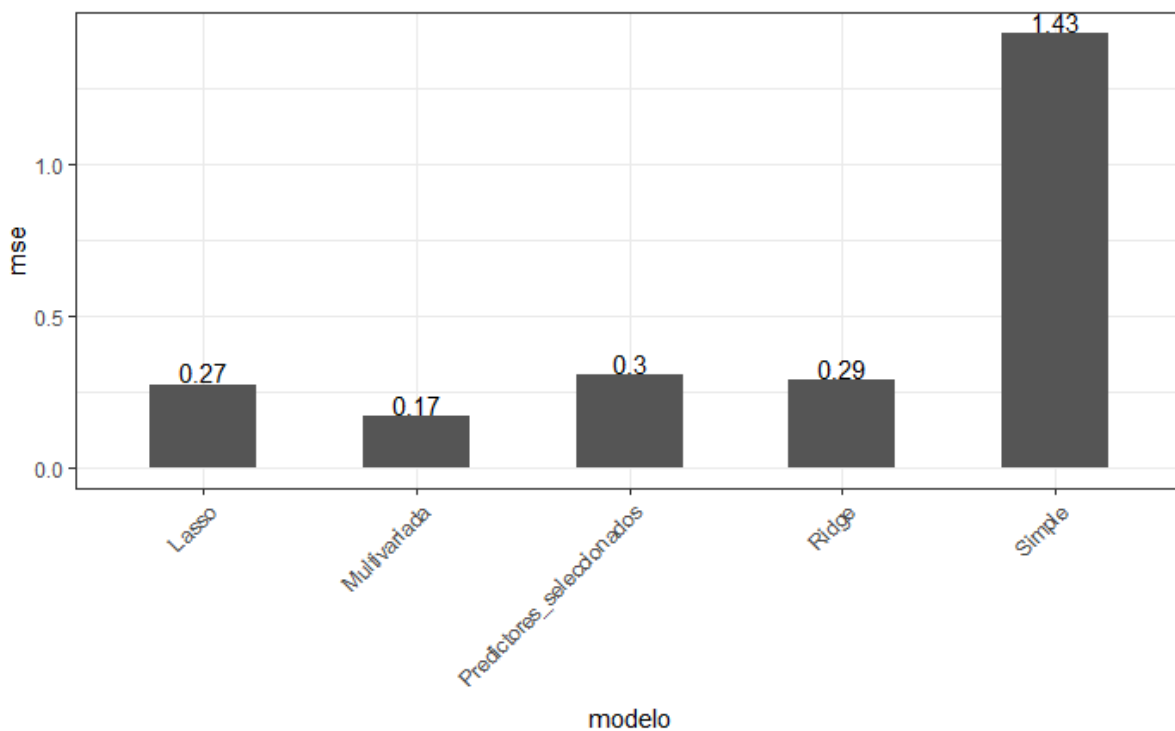
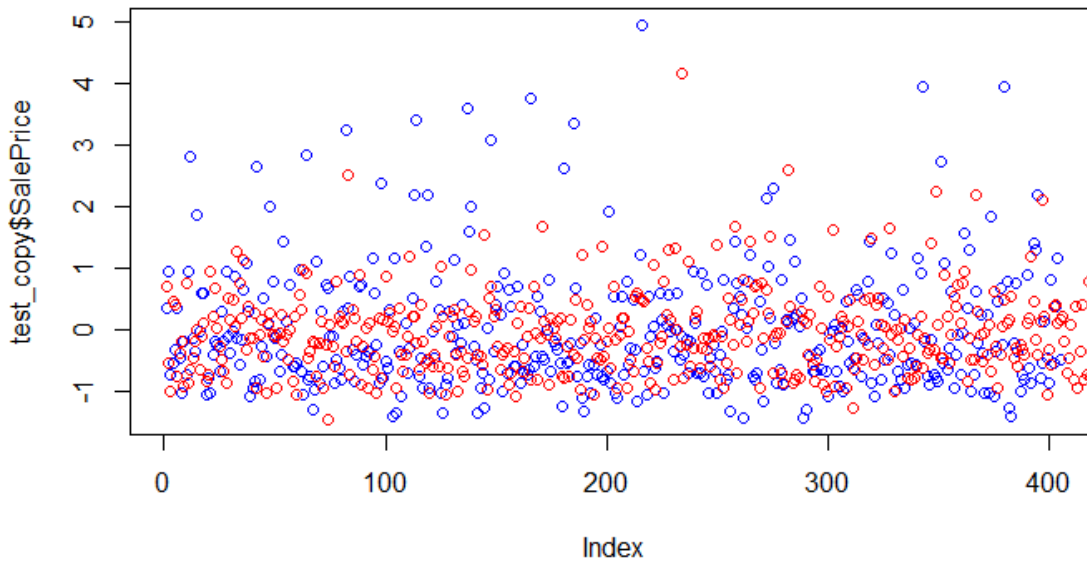
Regresión lineal múltiple

Predicciones test vs valores originales



Regresión lineal simple

Predicciones test vs valores originales



De manera que se puede observar que la regresión lineal simple no lo hizo del todo bien, la multivariada lo hizo bastante bien al igual que la de los predictores seleccionados. Aunque la multivariada y la simple realmente son modelos no aceptables.

12. Discuta sobre la efectividad de los modelos. ¿Cuál lo hizo mejor? ¿Cuál es el mejor modelo para predecir el precio de las casas? Haga los gráficos que crea que le pueden ayudar en la discusión.

Por lo tanto, la que mejor lo hizo fue la de predictores seleccionados dado que cumple con las características que se requieren al analizar sus residuos además que tiene un error medio

cuadrado no tan alto. Obviamente, se están obviando los modelos Ridge y Lasso dado que no son parte de esta discusión. Se debe de denotar que todos los gráficos que soportan esta conclusión están en los incisos anteriores.