Escogiendo Hogar

Andrea Tellería

March 13, 2016

# 1. Objetivos:

Dado que un amigo cercano tiene planeado irse a estudiar al extrangero próximamente (a Italia, Roma), se convirtió en una necesidad el buscar un lugar en el que quedarse y llamar hogar mientras duren sus estudios allí. En vista de esto y con ayuda de la plataforma *sturent*, se concentró en encontrar un buen hogar.

Nuestro amigo, al tener poco capital disponible, necesita escoger con ideas claras el lugar que va a ocupar, en base a una lista de lugares disponibles, la cual fue provista por *sturent*, planea conseguirlo. Para ayudarlo a lograr su onjetivo, se tratará de predecir el precio de los apartamentos en base a los apartamentos disponibles y escoger, de esta manera, el mejor entre ellos con ayuda de un modelo de regresión líneal.

# 2. Solución

## 2.1 Instalación de librerías necesarias

Primero que nada es necesario la instalación de ciertos paquetes que fueron usados a lo largo del análisis, para ello se procede a verificar si estos ya han sido instalados de no estarlo, realiza la instalación pertinente.

################################################################  
# Instalación de Paquetes necesarios  
################################################################  
  
#Creamos la función que recibe los paquetes  
install = function(pkg){  
 #Si ya esta instalado, no lo instala.  
 if (!require(pkg, character.only = TRUE)) {  
 install.packages(pkg)  
 if (!require(pkg, character.only = TRUE)) stop(paste("load failure:", pkg))  
 }  
}  
install("foreach")

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.2.3

#Seleccionamos los archivos que queremos instalar  
archive = c("rJava", "shiny", "rmarkdown", "gdata", "jsonlite", "RCurl")  
foreach(i = archive) %do% install(i)

## Loading required package: rJava

## Warning: package 'rJava' was built under R version 3.2.3

## Loading required package: shiny

## Warning: package 'shiny' was built under R version 3.2.3

## Loading required package: rmarkdown

## Loading required package: gdata

## Warning: package 'gdata' was built under R version 3.2.3

## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.

##

## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.

##   
## Attaching package: 'gdata'

## The following object is masked from 'package:stats':  
##   
## nobs

## The following object is masked from 'package:utils':  
##   
## object.size

## Loading required package: jsonlite

## Warning: package 'jsonlite' was built under R version 3.2.2

##   
## Attaching package: 'jsonlite'

## The following object is masked from 'package:shiny':  
##   
## validate

## Loading required package: RCurl

## Warning: package 'RCurl' was built under R version 3.2.3

## Loading required package: bitops

## Warning: package 'bitops' was built under R version 3.2.3

##   
## Attaching package: 'RCurl'

## The following object is masked from 'package:rJava':  
##   
## clone

## 2.2 Cargando la data

Debido a que los datos se encuentran en un documento excel (extensión .xslx), se usó la función que viene con el paquete *gdata*, *read.xls*, se cargó todo dentro de una variable tipo *data frame*, la que posteriormente se preprocesaria con la finalidad de dejar la data limpia y usable.

Dado que se uso el sistema operativo Windows para hacer el análisis, cabe destacar la necesidad de colocar como atributo para perl, la dirección del compilador (Una forma de no hacer esto es agregar perl a la variables del sistema).

################################################################  
# Lectura de archivos y carga de datos  
################################################################  
  
df = read.xls ("hogares.xlsx", sheet = 1, header = TRUE,   
 perl = "C:\\Strawberry\\Perl\\bin\\perl.exe")

## 2.3 Preprocesamiento

Lo primero que se notó es que la data se encontraba bastante complicada de analizar, con numerosas habitaciones mezcladas en un sólo registro y palabras con caracteres especiales. Por lo cual la necesidad de preprocesar la información se hizo básico.

### 2.3.1 Eliminación de columnas

Cuando se comenzó con el análisis lo primero fue dicernir aquellas variables a las que se les daría importancia durante el análisis y aquellas que se decidió dejar de lado. Esta decisión fue en gran medida tomada con la idea del tiempo que conllevaría el preprocesamiento de algunas columnas, al mismo tiempo junto con otros datos que se decidió no aportaban nada al estudio.

################################################################  
# Preprocesamiento  
################################################################  
  
#Eliminación de columnas que no aportan información  
#o requieren un gran trabajo de preprocesamiento  
colnames(df)[2] <- "Direccion" #Se elimina el acento del nombre  
colnames(df)[5] <- "Descripcion" #Se elimina el acento del nombre  
df$Tipo.de.Inmueble <- NULL  
df$Foto <- NULL  
df$Descripcion <- NULL  
df$Notas <- NULL

### 2.3.2 Limpiando la variable *Piso*

Para limpiar los *string*, se usó *gsub*, en el caso de la variable Piso, debido a sus características básicas, se requirió del uso de esta función para conseguir los pisos correctos.

#Limpiando los datos  
df$Piso <- gsub("[^TRS[:digit:]]", "", df$Piso)  
df$Piso[df$Piso=='ST'] <- 'S'  
df$Piso <- as.factor(df$Piso)  
df <- df[-c(74), ]

### 2.3.3 Índices de replicación

Como se comentó antes, a lo largo de toda la data ocurre que un registro hace referencia a más de una habitación, como éste caso es usual se trató de saber cuales filas era necesario replicar ya que en relaidad eran más de una, para realizar esto se decidió buscar una lista a la que nos referiremos como índice, el cual nos daría el número de repeticiones que sería necesrio hacer.

#### 2.3.3.1 Índice de precios y habitación

Como ya se dijo, la idea era buscar un índice (lista de números enteros) cuyo número indicaría la cantidad de veces que se estaban separando un mismo registro. Dada la columna de precios es obvio que muchas veces hay varios valores en una misma fila, la idea era separar estos y saber cuántas veces se hizo esto.

Para conseguirlo se optó por crear una función que nos retornaria con la lista directamente. Luego con la función *sapply*, quedaríamos con lo deseado.

Análogo ocurre con la habitación, en éste caso, además, de pasó a realizar una limpieza de tal forma que la información contenida en ambas columnas fuese más fácil de tratar y de utilizar.

#Indice de habitacion  
indx.hab = function(str){  
 str <- gsub("INTERO APPARTAMENTO", "1", str)  
 str <- gsub("MINI APPARTAMENTO", "1", str)  
 str <- gsub("MONOLOCALE", "1", str)  
 str <- gsub("[^[:digit:]]", "", str)  
 str <- ifelse (as.numeric(str)>9, 1, str)  
 return (str)  
}  
  
#Generando indice de habitación  
hab <- gsub("/", ";", df$Habitaciones.Disponibles)  
hab <- toupper(hab)  
idx\_hab <- sapply(hab, indx.hab, USE.NAMES = FALSE)  
idx\_hab <- as.numeric(idx\_hab)  
  
#Indices del precio, limpieza de precio  
df$Precio.Mensual <- toupper(df$Precio.Mensual)  
df$Precio.Mensual <- gsub("\n", "", df$Precio.Mensual)  
df$Precio.Mensual <- gsub(",", ";", df$Precio.Mensual)  
df$Precio.Mensual <- gsub("¬", ";", df$Precio.Mensual)  
  
#Función para conseguir índice de precio  
rep.reg = function (str){  
 str <- gsub("[^;[:digit:]]", "", str)  
 str <- strsplit(str, ";")  
 str <- unlist(str)  
 str <- str[str!= ""]  
 return (length(str))  
}  
  
#Generando indice con información de la columna de precios  
pre <- df$Precio.Mensual  
idx\_price <- sapply(pre, rep.reg, USE.NAMES = FALSE)  
idx\_price[5]<- 2  
idx\_price[18]<- 2

#### 2.3.3.2 *Verdaderos* índices

La información anterior nos dice cuántas veces se replica y las replicas que se generaros de dividir la data que se tenía. Ahora, se tiene que saber el total de replicas que el set de datos debe tener así como replicar aquellos registros tanto de habitaciones como de precios que también necesitan replicarse.

Por ejemplo, en el caso del precio, al dividirse en el caso de las habitaciones con diferentes costos, ya se duplicaron algunas filas, sin embargo en el caso de muchas habitaciones con el mismo precio, ésto no ocurrió, hay que tener por ende, en cuenta estos cosos especiales y tartar con ellos para conseguir los resultados esperados.

Para ello se generó una función la cual nos daría el *verdadero* índice que buscamos dada la resta entre el indce de precios y el del total de habitaciones.

Una vez se tienen las listas (índices) que deseabamos replicamos.

#Función para hayar el indice de replicación para los precios  
true.idx = function(idx){  
 for (i in seq(1,length(idx))) {  
 if (idx[i]>0) {  
 idx[i] <- idx[i] + 1  
 } else {  
 idx[i] <- 1  
 }  
 }  
 return(idx)  
}  
  
#Indice de replicación para los precios  
idx <- true.idx(idx\_hab - idx\_price)  
  
#Replicando filas que necesitan ser replicadas  
pre <- rep(pre, idx)  
hab <- rep(hab, idx\_hab)

### 2.3.4 Limpieza de precios

Una vez se replicó aquello que necesitamos que se replique es hora de limpiar nuestros precios, para hacer esto hay que separar aquellos precios que están en una misma fila para que éstas sean varias, eliminar todos los caractéres a exepción de los números, teniendo como consideración que los precios de los que se hablan a lo largo del estudio son todos euros.

Una vez tenemos los dígitos que queríamos los guardamos en una variable la cual va a ser nuestros nuevos precios en nuestro *dataframe*, antes de esto sin embargo, necesitaremos un par de variables extra

#Función para limpiar los precios  
rep.reg2 = function (str){  
 str <- gsub("[^;[:digit:]]", "", str)  
 str <- strsplit(str, ";")  
 str <- unlist(str)  
 str <- str[str!= ""]  
 return (str)  
}  
  
#Limpiando los precios  
clean\_price <- sapply(pre, rep.reg2, USE.NAMES = FALSE)  
clean\_price <- unlist(clean\_price)  
clean\_price[8] <- ""  
clean\_price[9] <- ""  
clean\_price[29] <- ""  
clean\_price <- as.numeric(clean\_price)  
clean\_price[7] <- clean\_price[7]+55  
clean\_price[6] <- clean\_price[6]+55  
clean\_price[28] <- clean\_price[28]+50  
clean\_price[27] <- clean\_price[27]+50  
clean\_price <- clean\_price[!is.na(clean\_price)]

### 2.3.5 Limpieza de habitaciones

Similar a lo ocurrido con los precios, las habitaciones deben ser limpiadas dándoles un formato más lógico. Una vez replicadas las casillas que necesitaban serlo, se pasó a eliminar de las cadenas de caracteres los caractéres especiales que aparecieron en ellos con el fin de dejar sólo el tipo de habitación a la que se refería. Entre los tipos de habitaciones resultantes se tenían:

1. SINGOLA
2. APPARTAMENTO (el cual contiene MINI APPARTAMENTO y MONOCALE)
3. DOPIA
4. POSTO LETTO

Además se agregó en aquellas habitaciones con servicios excluídos el costo de estos. Borrando cualquier registro que se generó debido al número del impuesto.

#Función para limpiar las habitaciones  
clean.hab = function(str){  
 str <- gsub("\\s", ";", str)  
 str <- gsub("[^;[:upper:]]", "", str)  
 str <- strsplit(str, ";")  
 str <- unlist(str)  
 str <- str[str!= ""]  
 return (str)  
}  
  
#Limpiando las habitaciones  
clean\_hab <- sapply(hab, clean.hab, USE.NAMES = FALSE)  
clean\_hab <- unlist(clean\_hab)  
clean\_hab <- clean\_hab[clean\_hab != "MINI"]  
clean\_hab <- clean\_hab[clean\_hab != "PRIVATO"]  
clean\_hab <- clean\_hab[clean\_hab != "E"]  
clean\_hab <- clean\_hab[clean\_hab != "O"]  
clean\_hab <- clean\_hab[clean\_hab != "LETTO"]  
clean\_hab <- clean\_hab[clean\_hab != "BAGNO"]  
clean\_hab <- clean\_hab[clean\_hab != "CON"]  
clean\_hab <- clean\_hab[clean\_hab != "USO"]  
clean\_hab <- clean\_hab[clean\_hab != "INTERO"]  
clean\_hab <- append(clean\_hab, "SINGOLA", 131)  
clean\_hab <- gsub("POSTO", "POSTO LETTO", clean\_hab)  
clean\_hab <- gsub("SINGOLE", "SINGOLA", clean\_hab)  
clean\_hab <- gsub("DOPPIE", "DOPPIA", clean\_hab)

### 2.3.6 Replicando el *data set*

Como ya se ha discutido repetidamente, varias filas contenian varias habitaciones, como ya se tienen las replicas de habitaciones (en las que se tuvo en cuenta no contar aquellas que venían de la separación de las cadenas, en casos donde las habitaciones eran de diferentes tipos) y el de los precios (análogo al índice de habitación, en este índice no se tomó en cuenta los datos de la duplicación que venía con separar precios diferentes).

Ahora bien, lo que se desea es el índice de replicación el conjunto de datos entero. En este caso es aquel con todas las habitaciones disponibles, éste con la finalidad de obtener direcciones, pisos, etc; que vienen con el *data set* original.

#Calculando la cantidad total de habitaciones  
#Indice total  
hab <- df$Habitaciones.Disponibles  
hab <- toupper(hab)  
hab <- gsub("INTERO APPARTAMENTO", "1", hab)  
hab <- gsub("MINI APPARTAMENTO", "1", hab)  
hab <- gsub("MONOLOCALE", "1", hab)  
hab <- gsub("\n1", ";1", hab)  
hab <- gsub("E;O\nUSO", ";1", hab)  
hab <- gsub("USO", ";1", hab)  
  
#Función para obtener el índice con la información de las habitaciones  
indx.tot = function(str){  
 str <- gsub("[^;[:digit:]]", "", str)  
 str <- strsplit(str, ";")  
 str <- unlist(str)  
 str <- str[str!= ""]  
 str <- as.numeric(str)  
 return (sum(str))  
}  
  
#Generando indice para el data set  
idx\_tot <- sapply(hab, indx.tot, USE.NAMES = FALSE)  
df <- df[rep(row.names(df), as.numeric(idx\_tot)),]  
df$Habitaciones.Disponibles <- clean\_hab  
df$Precio.Mensual <- clean\_price

### 2.3.7 Direcciones

Primero que nada, junto con la ayuda brindada por el grupo docente de la materia, el cuál facilitó de la función mediante las cuales se emitían las distancias (ya fuese en metros o en tiempo) entre direcciones. Se notó un error que tendía a ocurrir a menudo.

Lo primero es que ciertas direcciones tenían problemas con el **API de *google maps***, en estos casos sólo la dirección sin el distrito, o sólo el distrito sin la específica dirección eran suficiente para resolver el problema, con esto en mente se ideó una función que retornaría la dirección con la que se buscaría luego con ayuda del **API de *google maps***. Cabe destacar que en este punto todas las direcciones arrojan resultadas, aunque no se pudo confirmar si eran las direcciones correctas o no.

#Buscando la dirección completa  
origen\_list <- paste(df$Distrito, df$Direccion)  
origen\_list <- gsub("'", "", origen\_list)  
origen\_list <- gsub("\n/", " ", origen\_list)  
origen\_list <- gsub("\n", " ", origen\_list)  
origen\_list <- gsub(" ", " ", origen\_list)  
country <- "Italia"  
destny\_list <- "Universita di Roma Sapienza"  
origen\_list <- paste(origen\_list, country)  
  
#Limpiando las direcciones para qe la búsqueda arroje resultados  
clean.dirs = function(dir) {  
 dir <- ifelse(grepl("Pigneto", dir), "Pigneto", dir)  
 dir <- ifelse(grepl("Batteria Nomentana", dir), "Batteria Nomentana", dir)  
 dir <- ifelse(grepl("San Giovanni", dir), "San Giovanni", dir)  
 dir <- ifelse(grepl("Via Endert", dir), "Via Enderta Roma Italia", dir)  
 dir <- ifelse(grepl("Via Tripolitana", dir), "Via Tripolitania", dir)  
 dir <- ifelse(grepl("Via Filippo Marchetti", dir), "Via Filippo Marchetti", dir)  
 dir <- ifelse(grepl("Montesacro Nomentano Via Cimone Italia", dir), "Montesacro Nomentano Italia", dir)  
 dir <- ifelse(grepl("Casilina Tuscolana Piazza Giuseppe Cardinali Italia", dir), "Casilina Tuscolana Italia", dir)  
 dir <- ifelse(grepl("Baldo degli Ubaldi", dir), "Via Baldo degli Ubaldi Italia", dir)  
 dir <- ifelse(grepl("Balduina Via Lattanzio Italia", dir), "Via Lattanzio Italia", dir)  
 dir <- ifelse(grepl("Bravetta Via del Fontanile Arenato Italia", dir), "Via del Fontanile Arenato Italia", dir)  
 dir <- ifelse(grepl("Cipro Via Bragadin Italia", dir), "Via Bragadin Italia", dir)  
 dir <- ifelse(grepl("Cornelia Piazza", dir), "Cornelia Piazza Roma Italia", dir)  
 dir <- ifelse(grepl("Cornelia Largo", dir), "Cornelia Largo Roma Italia", dir)  
 dir <- ifelse(grepl("Via Giovanni Tamassia Italia", dir), "Via Giovanni Tamassia Italia", dir)  
 dir <- ifelse(grepl("Gregorio VII Via Gregorio VII", dir), "Via Gregorio VII Italia", dir)  
 dir <- ifelse(grepl("Via Francesco Pacelli Italia", dir), "Via Francesco Pacelli Italia", dir)  
 dir <- ifelse(grepl("Via Montesanto Italia", dir), "Via Montesanto Italia", dir)  
 dir <- ifelse(grepl("Magliana Via Biagio a Ripoli Italia", dir), "Magliana Italia", dir)  
 dir <- ifelse(grepl("Via Ascanio Fenizi Italia", dir), "Via Ascanio Fenizi Italia", dir)  
 dir <- ifelse(grepl("Via della Villa Lucina Italia", dir), "Via della Villa Lucina Italia", dir)  
 dir <- ifelse(grepl("Via Gabriello Chiabrera Italia", dir), "Via Gabriello Chiabrera Italia", dir)  
 dir <- ifelse(grepl("Via Antonino Pio", dir), "Via Antonino Pio Italia", dir)  
 dir <- ifelse(grepl("Largo Luigi Antonelli Italia", dir), "Largo Luigi Antonelli Italia", dir)  
 dir <- ifelse(grepl("Via Cristoforo Colombo 348 Italia", dir), "Via Cristoforo Colombo 348 Italia", dir)  
 dir <- ifelse(grepl("Via Pellegrino Matteucci Italia", dir), "Via Pellegrino Matteucci Italia", dir)  
 dir <- ifelse(grepl("Via Ettore Rolli Italia", dir), "Via Ettore Rolli Italia", dir)  
 dir <- ifelse(grepl("Largo Enea Bortolotti Italia", dir), "Largo Enea Bortolotti Italia", dir)  
 dir <- ifelse(grepl("Largo Giuseppe Veratti Italia", dir), "Largo Giuseppe Veratti Italia", dir)  
 dir <- ifelse(grepl("Via Antonio Roiti Italia", dir), "Via Antonio Roiti Italia", dir)  
 dir <- ifelse(grepl("Lungotevere degli Inventori Italia", dir), "Lungotevere degli Inventori Italia", dir)  
 dir <- ifelse(grepl("Via Federico Guarducci Italia", dir), "Via Federico Guarducci Italia", dir)  
 dir <- ifelse(grepl("Via Pietro Blaserna Italia", dir), "Via Pietro Blaserna Italia", dir)  
 dir <- ifelse(grepl("Via Antonino Lo Surdo Italia", dir), "Via Antonino Lo Surdo Italia", dir)  
 dir <- ifelse(grepl("Via Eugenio Barsanti Italia", dir), "Via Eugenio Barsanti Italia", dir)  
 dir <- ifelse(grepl("Lungotevere Dante Italia", dir), "Lungotevere Dante Italia", dir)  
 dir <- ifelse(grepl("Via Francesco Maurolico Italia", dir), "Via Francesco Maurolico Italia", dir)  
 dir <- ifelse(grepl("Via Giuseppe Veronesi Italia", dir), "Via Giuseppe Veronesi Italia", dir)  
 dir <- ifelse(grepl("Via Luigi Biolchini Italia", dir), "Via Luigi Biolchini Italia", dir)  
 dir <- ifelse(grepl("Via Bernardino Molinari Italia", dir), "Via Bernardino Molinari Italia", dir)  
}  
  
#Limpiando las direcciones  
origen\_list<-sapply(origen\_list, clean.dirs)

### 2.3.8 *Google Maps*

Una vez tuvo las direcciones deseadas (así como el destino), se definió el *Key* del **API**, con la que fue posible establecer el contacto con *google maps*, una vez se tuvieron estos elementos, se llamó a la función *parse\_data*, la cual fue completada a la hora del estudio y no fue provista originalmente por el grupo de estudio. A esta función se le envió la lista de direcciones de oriden, el destino y el **API *Key***.

# Seleccionar google\_api.R en su sistema de archivos, api key y llamar a la función  
# source(file.choose())  
api\_key = "AIzaSyCS6AqTNIPGnPzIMGpyfianRvq5X1bwu9s"  
datos <- parse\_data(origen\_list, destny\_list, api\_key)  
df$Distancia <- datos

### 2.3.9 parse\_data

La función parse\_data fue dada con la intención de ser completada al momento del estudio, con esta función se buscaron las distancias dadas las direcciones de origen y destino. Debido a complicaciones a la hora de usar el API de google, se decidió separar la lista de origen inicial, de tal forma que no hubiese problemas.

Cada lista parcial llama a la función que ofrece los servicios del API de google y luego todasse mezclan de nuevo a forma de tener las distancias deseadas. La función retorna justo esta lista de distacias resultante.

#Recibiendo una lista de direcciones con su destino, retorna la distancia en metros  
parse\_data = function(origen\_list, destny\_list, api\_key, mode = "driving", language = "es"){  
 #Separar la lista en pedazos para que puedan ser procesada por el API de Google  
 origen\_list2<-origen\_list[1:40]  
 origen\_list3<-origen\_list[41:80]  
 origen\_list4<-origen\_list[81:120]  
 origen\_list5<-origen\_list[121:160]  
   
 # Colocar su API Key   
 api\_key = "AIzaSyA5e9bq0\_Nmyix8ytYbMyOEbsOhVdY83Bc"  
 api\_url = get\_url(origen\_list2, destny\_list, api\_key)  
 datos = get\_data(api\_url)  
 datos2 <- unlist(datos$rows)  
 api\_url = get\_url(origen\_list3, destny\_list, api\_key)  
 datos = get\_data(api\_url)  
 datos3 <- unlist(datos$rows)  
 api\_url = get\_url(origen\_list4, destny\_list, api\_key)  
 datos = get\_data(api\_url)  
 datos4 <- unlist(datos$rows)  
 api\_url = get\_url(origen\_list5, destny\_list, api\_key)  
 datos = get\_data(api\_url)  
 datos5 <- unlist(datos$rows)  
   
 #combinando los resultados parciales  
 datos <- append(datos2, datos3, after = length(datos2))  
 datos <- append(datos, datos4, after = length(datos))  
 datos <- append(datos, datos5, after = length(datos))  
 datos <- datos[grep("elements.duration.value", names(datos))]  
 return (datos)  
}

## 2.4 Regresión líneal

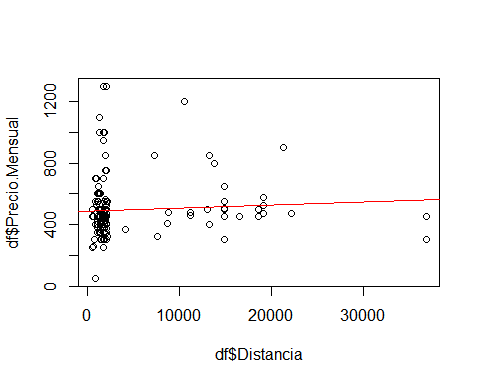
Una vz tenemos la data deseada en un *data frame*, pasamos al verdadero estudio dónde con ayuda de una regresión líneal se busca conseguir el mejor apartamento (siempre con la idea del costo del mismo en mente) para nuestro amigo, quien planea irse a estudiar fuera.

Para conseguir el modelo usando regresión líneal se usó de la función *lm*, provista por R dentro del paquete básico (*stats*).

Como se tenía en cuenta que lo importante era el dinero se decidió hacer varios modelos (uno por cada variable), donde se trabajaría con el precio en contra de la característica que el resto d las variables tuviesen para ofrecer.

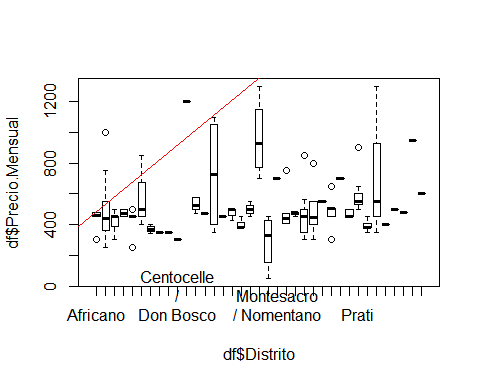
Así pues se consiguieron diversos modelos.

################################################################  
# Regresión Lineal  
################################################################  
  
#Modelo con Precio Vs. Distancia (distancia en metros)  
df$Distancia <- as.numeric(df$Distancia)  
lr <- lm(df$Precio.Mensual ~ df$Distancia, na.action = NULL)  
plot(df$Precio.Mensual ~ df$Distancia, df)  
abline(lr, col="red")



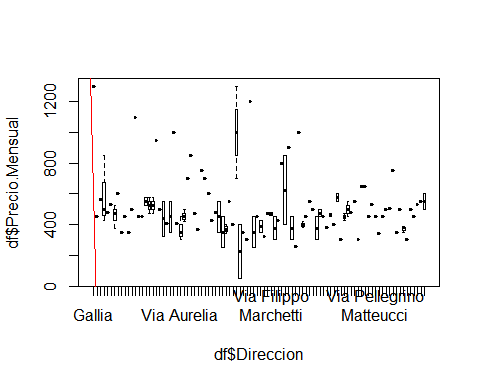
#Modelo con Precio Vs. Distrito  
lr <- lm(df$Precio.Mensual ~ df$Distrito, na.action = NULL)  
plot(df$Precio.Mensual ~ df$Distrito, df)  
abline(lr, col="red")

## Warning in abline(lr, col = "red"): only using the first two of 37  
## regression coefficients



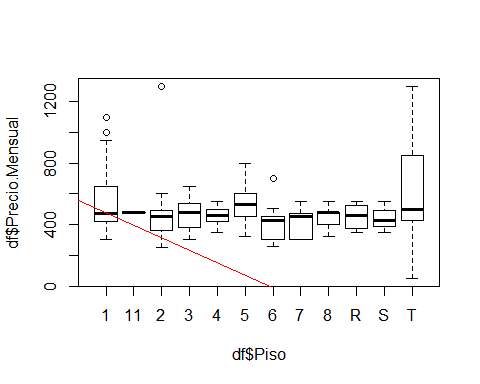
#Modelo con Precio Vs. Dirección  
lr <- lm(df$Precio.Mensual ~ df$Direccion, na.action = NULL)  
plot(df$Precio.Mensual ~ df$Direccion, df)  
abline(lr, col="red")

## Warning in abline(lr, col = "red"): only using the first two of 96  
## regression coefficients



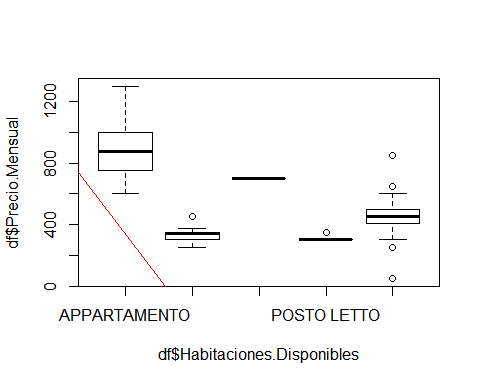
#Modelo con Precio Vs. Piso  
lr <- lm(df$Precio.Mensual ~ df$Piso, na.action = NULL)  
plot(df$Precio.Mensual ~ df$Piso, df)  
abline(lr, col="red")

## Warning in abline(lr, col = "red"): only using the first two of 12  
## regression coefficients



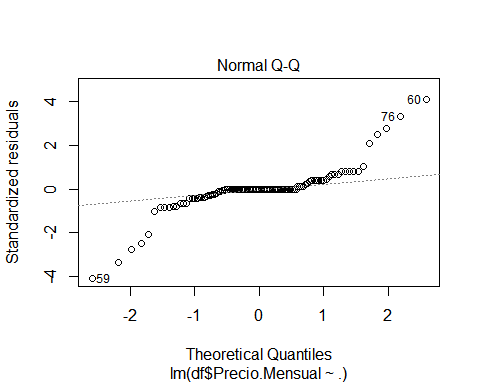
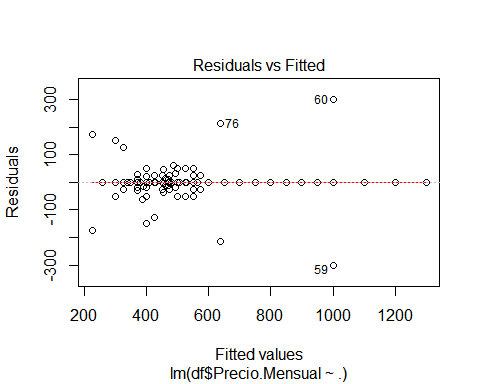
#Modelo con Precio Vs. Habitación  
df$Habitaciones.Disponibles <- as.factor(df$Habitaciones.Disponibles)  
lr <- lm(df$Precio.Mensual ~ df$Habitaciones.Disponibles, na.action = NULL)  
plot(df$Precio.Mensual ~ df$Habitaciones.Disponibles, df)  
abline(lr, col="red")

## Warning in abline(lr, col = "red"): only using the first two of 5  
## regression coefficients

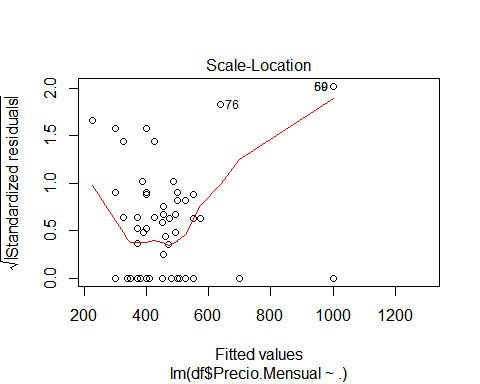


#Modelo con todas las varibles que forman parte de la vista generada  
lr\_all <- lm(df$Precio.Mensual ~ ., df, na.action = NULL)  
plot(lr\_all)

## Warning: not plotting observations with leverage one:  
## 1, 3, 8, 11, 16, 21, 22, 27, 30, 43, 54, 57, 58, 59, 64, 65, 66, 67, 79, 80, 81, 82, 83, 84, 85, 88, 91, 95, 102, 105, 106, 107, 108, 109, 110, 114, 115, 116, 117, 118, 123, 124, 129, 130, 136, 141, 142, 150, 151, 152, 153, 154, 157, 158, 159, 160

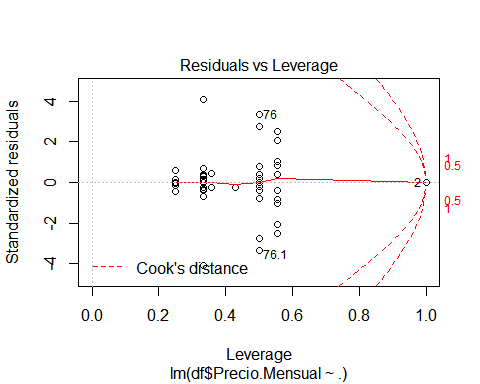


## Warning: not plotting observations with leverage one:  
## 1, 3, 8, 11, 16, 21, 22, 27, 30, 43, 54, 57, 58, 59, 64, 65, 66, 67, 79, 80, 81, 82, 83, 84, 85, 88, 91, 95, 102, 105, 106, 107, 108, 109, 110, 114, 115, 116, 117, 118, 123, 124, 129, 130, 136, 141, 142, 150, 151, 152, 153, 154, 157, 158, 159, 160



## Warning in sqrt(crit \* p \* (1 - hh)/hh): NaNs produced

## Warning in sqrt(crit \* p \* (1 - hh)/hh): NaNs produced



summary(lr\_all)

##   
## Call:  
## lm(formula = df$Precio.Mensual ~ ., data = df, na.action = NULL)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -300 0 0 0 300   
##   
## Coefficients: (46 not defined because of singularities)  
## Estimate Std. Error t value  
## (Intercept) 3.800e+02 1.562e+02 2.432  
## DistritoArdeatino -3.000e+01 1.804e+02 -0.166  
## DistritoBaldo degli\nUbaldi 7.500e+01 1.626e+02 0.461  
## DistritoBalduina -3.550e+02 2.116e+02 -1.678  
## DistritoBatteria\nNomentana 7.000e+01 1.688e+02 0.415  
## DistritoBologna 2.000e+01 1.804e+02 0.111  
## DistritoBravetta -4.400e+02 3.232e+02 -1.362  
## DistritoCasilina\n/ Tuscolana -4.600e+02 3.253e+02 -1.414  
## DistritoCentocelle -4.600e+02 3.315e+02 -1.388  
## DistritoCentocelle\n/\nDon Bosco -8.000e+01 1.626e+02 -0.492  
## DistritoCentro 3.700e+02 2.210e+02 1.674  
## DistritoCentro\n/ Manzoni -2.850e+02 3.232e+02 -0.882  
## DistritoCipro 9.000e+01 1.804e+02 0.499  
## DistritoCornelia 6.200e+02 1.804e+02 3.436  
## DistritoGianicolense -3.600e+02 3.315e+02 -1.086  
## DistritoGregorio VII 1.200e+02 1.688e+02 0.711  
## DistritoLaurentina -3.000e+01 2.017e+02 -0.149  
## DistritoLepanto 1.200e+02 1.647e+02 0.729  
## DistritoMagliana 1.700e+02 2.083e+02 0.816  
## DistritoMontesacro 7.000e+01 1.804e+02 0.388  
## DistritoMontesacro\n/ Nomentano 3.200e+02 1.804e+02 1.774  
## DistritoMonteverde\n/ Trastevere 7.000e+01 1.647e+02 0.425  
## DistritoOstiense\n/ Garbatella 7.000e+01 1.804e+02 0.388  
## DistritoOstiense\n/ Marconi 6.089e+01 1.739e+02 0.350  
## DistritoOstiense\n/\nSan Paolo -4.045e-12 1.276e+02 0.000  
## DistritoParioli 1.700e+02 1.804e+02 0.942  
## DistritoPigneto 1.250e+02 1.688e+02 0.741  
## DistritoPonte Milvio -1.300e+02 2.210e+02 -0.588  
## DistritoPortuense 1.200e+02 1.688e+02 0.711  
## DistritoPrati 1.700e+02 1.688e+02 1.007  
## DistritoPrenestino -4.400e+02 3.232e+02 -1.362  
## DistritoSan Giovanni 4.900e+02 3.315e+02 1.478  
## DistritoSan Giovanni\n/\nPonte Lungo 2.000e+01 1.688e+02 0.119  
## DistritoSan Giovanni\n/\nRe di Roma -3.300e+02 2.116e+02 -1.560  
## DistritoSan Pietro -3.300e+02 3.315e+02 -0.996  
## DistritoTestaccio 5.700e+02 1.804e+02 3.159  
## DistritoTor Sapienza -2.100e+02 3.315e+02 -0.634  
## DireccionLargo Boccea -5.500e+02 1.276e+02 -4.311  
## DireccionLargo Enea\nBortolotti -3.059e+02 2.423e+02 -1.262  
## DireccionLargo Giuseppe\nVeratti 1.966e+02 9.941e+01 1.978  
## DireccionLargo Luigi\nAntonelli 1.000e+02 1.804e+02 0.554  
## DireccionLungotevere degli Inventori 8.911e+01 1.181e+02 0.754  
## DireccionLungotevere\nDante 5.107e+01 8.693e+01 0.588  
## DireccionPiazza Cesare de\nCupis NA NA NA  
## DireccionPiazza di villa\nCarpegna -6.500e+02 1.276e+02 -5.095  
## DireccionPiazza Fonteiana NA NA NA  
## DireccionPiazza Giuseppe\nCardinali NA NA NA  
## DireccionPiazza Massa\nCarrara -3.300e+02 2.116e+02 -1.560  
## DireccionPiazza\nGiureconsulti 1.000e+02 1.276e+02 0.784  
## DireccionRoberto\nSanseverino 8.000e+01 1.042e+02 0.768  
## DireccionVia Adolfo Rava' 1.000e+02 1.276e+02 0.784  
## Direccionvia Alfieri int. 11 4.550e+02 2.270e+02 2.004  
## Direccionvia Alfieri int. 12 4.300e+02 2.240e+02 1.919  
## Direccionvia Alfieri int. 8 NA NA NA  
## DireccionVia Amerigo\nVespucci NA NA NA  
## DireccionVia Anglona NA NA NA  
## DireccionVia Antonino Lo\nSurdo 4.705e+01 9.260e+01 0.508  
## DireccionVia Antonino Pio\nF19/20 3.000e+01 1.804e+02 0.166  
## DireccionVia Antonio Roiti 5.955e+01 9.260e+01 0.643  
## DireccionVia Ardeatina 2.000e+02 1.804e+02 1.109  
## DireccionVia Ascanio Fenizi -4.700e+02 2.270e+02 -2.070  
## DireccionVia Aurelia -5.050e+02 1.496e+02 -3.376  
## Direccionvia Baldo degli\nUbaldi NA NA NA  
## DireccionVia Bernardino\nMolinari NA NA NA  
## DireccionVia Biagio a\nRipoli 3.000e+02 1.647e+02 1.821  
## DireccionVia Bragadin NA NA NA  
## DireccionVia Casamari -7.089e+01 1.181e+02 -0.600  
## DireccionVia Cesare\nPascarella 3.000e+02 1.042e+02 2.880  
## DireccionVia Cimone NA NA NA  
## DireccionVia Cola di\nRienzo 5.000e+01 1.105e+02 0.453  
## DireccionVia Cristoforo\nColombo 8.000e+01 1.276e+02 0.627  
## DireccionVia Cristoforo\nColombo 348 3.000e+01 1.042e+02 0.288  
## DireccionVia Dacia -3.696e+02 2.396e+02 -1.542  
## DireccionVia dei Georgofili -3.796e+02 2.396e+02 -1.584  
## DireccionVia del Fontanile\nArenato NA NA NA  
## DireccionVia dell'Accademia Antiquaria 2.000e+02 1.105e+02 1.810  
## DireccionVia della Villa\nLucina 2.000e+01 1.804e+02 0.111  
## DireccionVia della\nMagliana Nuova\n342 NA NA NA  
## Direccionvia delle Alpi\nApuane -6.750e+02 1.688e+02 -4.000  
## DireccionVia delle Palme NA NA NA  
## DireccionVia delle\nAccademie -4.200e+02 2.210e+02 -1.901  
## DireccionVia di Panico NA NA NA  
## Direccionvia di Sacco\nPastore -4.796e+02 2.309e+02 -2.077  
## DireccionVia EndertÃ  -3.600e+02 3.315e+02 -1.086  
## DireccionVia Ettore Rolli -1.089e+01 1.181e+02 -0.092  
## DireccionVia Eugenio\nBarsanti -1.500e+01 1.105e+02 -0.136  
## DireccionVia Federico\nGuarducci 3.411e+01 1.181e+02 0.289  
## DireccionVia Filippo\nMarchetti 9.000e+01 1.626e+02 0.553  
## DireccionVia Francesco\nMaurolico 1.221e+02 1.442e+02 0.847  
## DireccionVia Francesco\nPacelli -5.250e+02 1.688e+02 -3.111  
## DireccionVia Gabriello\nChiabrera 4.200e+02 1.804e+02 2.328  
## DireccionVia Gatteschi NA NA NA  
## DireccionVia Giordano\nBruno 3.500e+02 1.105e+02 3.168  
## DireccionVia Giovanni De\nAgostini 5.795e+01 1.442e+02 0.402  
## Direccionvia Giovanni\nConti -8.911e+01 1.342e+02 -0.664  
## DireccionVia Giovanni\nTamassia NA NA NA  
## DireccionVia Girolamo\nCocconi 4.600e+02 2.240e+02 2.053  
## DireccionVia Giuseppe\nVeronesi 9.107e+00 1.181e+02 0.077  
## DireccionVia Golametto -7.077e-14 1.105e+02 0.000  
## DireccionVia Gregorio VII\nint. 13 NA NA NA  
## DireccionVia Guido\nCastelnuovo 1.221e+02 1.442e+02 0.847  
## DireccionVia Lattanzio NA NA NA  
## DireccionVia Laurentina 32 1.000e+02 1.562e+02 0.640  
## DireccionVia Laurentina\n197 3.000e+01 1.105e+02 0.272  
## DireccionVia Luigi Biolchini 1.591e+02 1.486e+02 1.071  
## DireccionVia Macedonia NA NA NA  
## DireccionVia Marcantonio\nColonna 2.500e+01 9.021e+01 0.277  
## Direccionvia Mario Lizzani NA NA NA  
## DireccionVia Monte Verde NA NA NA  
## DireccionVia Montesanto NA NA NA  
## DireccionVia Niccolo' III NA NA NA  
## Direccionvia Ostiense 224 3.075e+02 2.116e+02 1.453  
## Direccionvia Ostiense 459 NA NA NA  
## DireccionVia Ottaviano 1.000e+02 1.105e+02 0.905  
## Direccionvia Padre\nRaffaele Melis 1.450e+02 1.105e+02 1.312  
## Direccionvia Pantelleria NA NA NA  
## DireccionVia Paolo Emilio -2.000e+01 1.105e+02 -0.181  
## DireccionVia Pellegrino\nMatteucci NA NA NA  
## DireccionVia Pietro\nBlaserna NA NA NA  
## DireccionVia Portuense\n471\nInt. 10 -4.800e+02 2.270e+02 -2.114  
## DireccionVia Portuense\n471\nInt. 14 NA NA NA  
## Direccionvia Prenestina NA NA NA  
## DireccionVia Roberto\nFerruzzi -5.000e+01 1.804e+02 -0.277  
## DireccionVia San Roberto\nBellarimino NA NA NA  
## DireccionVia San Tommaso d'Aquino -4.800e+02 2.387e+02 -2.011  
## DireccionVia Silvio Benco NA NA NA  
## DireccionVia Tripolitana NA NA NA  
## DireccionVia Val d'Ossola -3.800e+02 2.387e+02 -1.592  
## DireccionVia Valsassina NA NA NA  
## DireccionViale Angelico -2.000e+01 1.105e+02 -0.181  
## DireccionViale Bruno\nBuozzi NA NA NA  
## DireccionViale Giulio\nCesare NA NA NA  
## Piso11 NA NA NA  
## Piso2 4.300e+02 2.116e+02 2.032  
## Piso3 NA NA NA  
## Piso4 NA NA NA  
## Piso5 NA NA NA  
## Piso6 NA NA NA  
## Piso7 -1.375e+02 1.105e+02 -1.245  
## Piso8 NA NA NA  
## PisoR NA NA NA  
## PisoS NA NA NA  
## PisoT 4.500e+02 1.276e+02 3.527  
## Habitaciones.DisponiblesDOPPIA -1.009e+02 4.176e+01 -2.416  
## Habitaciones.DisponiblesMONOLOCALE NA NA NA  
## Habitaciones.DisponiblesPOSTO LETTO -8.000e+01 1.276e+02 -0.627  
## Habitaciones.DisponiblesSINGOLA NA NA NA  
## Distancia NA NA NA  
## Pr(>|t|)   
## (Intercept) 0.018123 \*   
## DistritoArdeatino 0.868517   
## DistritoBaldo degli\nUbaldi 0.646400   
## DistritoBalduina 0.098730 .   
## DistritoBatteria\nNomentana 0.679841   
## DistritoBologna 0.912117   
## DistritoBravetta 0.178614   
## DistritoCasilina\n/ Tuscolana 0.162632   
## DistritoCentocelle 0.170500   
## DistritoCentocelle\n/\nDon Bosco 0.624635   
## DistritoCentro 0.099429 .   
## DistritoCentro\n/ Manzoni 0.381473   
## DistritoCipro 0.619784   
## DistritoCornelia 0.001096 \*\*   
## DistritoGianicolense 0.281919   
## DistritoGregorio VII 0.479913   
## DistritoLaurentina 0.882288   
## DistritoLepanto 0.469183   
## DistritoMagliana 0.417836   
## DistritoMontesacro 0.699450   
## DistritoMontesacro\n/ Nomentano 0.081374 .   
## DistritoMonteverde\n/ Trastevere 0.672400   
## DistritoOstiense\n/ Garbatella 0.699450   
## DistritoOstiense\n/ Marconi 0.727424   
## DistritoOstiense\n/\nSan Paolo 1.000000   
## DistritoParioli 0.349978   
## DistritoPigneto 0.461888   
## DistritoPonte Milvio 0.558604   
## DistritoPortuense 0.479913   
## DistritoPrati 0.317976   
## DistritoPrenestino 0.178614   
## DistritoSan Giovanni 0.144729   
## DistritoSan Giovanni\n/\nPonte Lungo 0.906077   
## DistritoSan Giovanni\n/\nRe di Roma 0.124242   
## DistritoSan Pietro 0.323575   
## DistritoTestaccio 0.002512 \*\*   
## DistritoTor Sapienza 0.528852   
## DireccionLargo Boccea 6.38e-05 \*\*\*  
## DireccionLargo Enea\nBortolotti 0.211838   
## DireccionLargo Giuseppe\nVeratti 0.052710 .   
## DireccionLargo Luigi\nAntonelli 0.581532   
## DireccionLungotevere degli Inventori 0.453649   
## DireccionLungotevere\nDante 0.559144   
## DireccionPiazza Cesare de\nCupis NA   
## DireccionPiazza di villa\nCarpegna 3.98e-06 \*\*\*  
## DireccionPiazza Fonteiana NA   
## DireccionPiazza Giuseppe\nCardinali NA   
## DireccionPiazza Massa\nCarrara 0.124242   
## DireccionPiazza\nGiureconsulti 0.436323   
## DireccionRoberto\nSanseverino 0.445601   
## DireccionVia Adolfo Rava' 0.436323   
## Direccionvia Alfieri int. 11 0.049727 \*   
## Direccionvia Alfieri int. 12 0.059846 .   
## Direccionvia Alfieri int. 8 NA   
## DireccionVia Amerigo\nVespucci NA   
## DireccionVia Anglona NA   
## DireccionVia Antonino Lo\nSurdo 0.613266   
## DireccionVia Antonino Pio\nF19/20 0.868517   
## DireccionVia Antonio Roiti 0.522655   
## DireccionVia Ardeatina 0.272214   
## DireccionVia Ascanio Fenizi 0.042890 \*   
## DireccionVia Aurelia 0.001319 \*\*   
## Direccionvia Baldo degli\nUbaldi NA   
## DireccionVia Bernardino\nMolinari NA   
## DireccionVia Biagio a\nRipoli 0.073692 .   
## DireccionVia Bragadin NA   
## DireccionVia Casamari 0.550704   
## DireccionVia Cesare\nPascarella 0.005563 \*\*   
## DireccionVia Cimone NA   
## DireccionVia Cola di\nRienzo 0.652561   
## DireccionVia Cristoforo\nColombo 0.533072   
## DireccionVia Cristoforo\nColombo 348 0.774372   
## DireccionVia Dacia 0.128399   
## DireccionVia dei Georgofili 0.118585   
## DireccionVia del Fontanile\nArenato NA   
## DireccionVia dell'Accademia Antiquaria 0.075446 .   
## DireccionVia della Villa\nLucina 0.912117   
## DireccionVia della\nMagliana Nuova\n342 NA   
## Direccionvia delle Alpi\nApuane 0.000182 \*\*\*  
## DireccionVia delle Palme NA   
## DireccionVia delle\nAccademie 0.062313 .   
## DireccionVia di Panico NA   
## Direccionvia di Sacco\nPastore 0.042284 \*   
## DireccionVia EndertÃ  0.281919   
## DireccionVia Ettore Rolli 0.926838   
## DireccionVia Eugenio\nBarsanti 0.892477   
## DireccionVia Federico\nGuarducci 0.773790   
## DireccionVia Filippo\nMarchetti 0.582113   
## DireccionVia Francesco\nMaurolico 0.400654   
## DireccionVia Francesco\nPacelli 0.002893 \*\*   
## DireccionVia Gabriello\nChiabrera 0.023428 \*   
## DireccionVia Gatteschi NA   
## DireccionVia Giordano\nBruno 0.002450 \*\*   
## DireccionVia Giovanni De\nAgostini 0.689182   
## Direccionvia Giovanni\nConti 0.509447   
## DireccionVia Giovanni\nTamassia NA   
## DireccionVia Girolamo\nCocconi 0.044552 \*   
## DireccionVia Giuseppe\nVeronesi 0.938805   
## DireccionVia Golametto 1.000000   
## DireccionVia Gregorio VII\nint. 13 NA   
## DireccionVia Guido\nCastelnuovo 0.400654   
## DireccionVia Lattanzio NA   
## DireccionVia Laurentina 32 0.524691   
## DireccionVia Laurentina\n197 0.786947   
## DireccionVia Luigi Biolchini 0.288809   
## DireccionVia Macedonia NA   
## DireccionVia Marcantonio\nColonna 0.782666   
## Direccionvia Mario Lizzani NA   
## DireccionVia Monte Verde NA   
## DireccionVia Montesanto NA   
## DireccionVia Niccolo' III NA   
## Direccionvia Ostiense 224 0.151483   
## Direccionvia Ostiense 459 NA   
## DireccionVia Ottaviano 0.369156   
## Direccionvia Padre\nRaffaele Melis 0.194556   
## Direccionvia Pantelleria NA   
## DireccionVia Paolo Emilio 0.856983   
## DireccionVia Pellegrino\nMatteucci NA   
## DireccionVia Pietro\nBlaserna NA   
## DireccionVia Portuense\n471\nInt. 10 0.038799 \*   
## DireccionVia Portuense\n471\nInt. 14 NA   
## Direccionvia Prenestina NA   
## DireccionVia Roberto\nFerruzzi 0.782666   
## DireccionVia San Roberto\nBellarimino NA   
## DireccionVia San Tommaso d'Aquino 0.048970 \*   
## DireccionVia Silvio Benco NA   
## DireccionVia Tripolitana NA   
## DireccionVia Val d'Ossola 0.116792   
## DireccionVia Valsassina NA   
## DireccionViale Angelico 0.856983   
## DireccionViale Bruno\nBuozzi NA   
## DireccionViale Giulio\nCesare NA   
## Piso11 NA   
## Piso2 0.046691 \*   
## Piso3 NA   
## Piso4 NA   
## Piso5 NA   
## Piso6 NA   
## Piso7 0.218316   
## Piso8 NA   
## PisoR NA   
## PisoS NA   
## PisoT 0.000828 \*\*\*  
## Habitaciones.DisponiblesDOPPIA 0.018859 \*   
## Habitaciones.DisponiblesMONOLOCALE NA   
## Habitaciones.DisponiblesPOSTO LETTO 0.533072   
## Habitaciones.DisponiblesSINGOLA NA   
## Distancia NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 90.21 on 58 degrees of freedom  
## Multiple R-squared: 0.9161, Adjusted R-squared: 0.77   
## F-statistic: 6.27 on 101 and 58 DF, p-value: 2.506e-12

Si se nota, el último modelofue realizado con respecto a todas las varibles al mismo tiempo. Además de emitir gráficas que hablan como el modelo como tal se hizo un resumen del modelo con ayuda de la función *summary*.

Si vemos en nuestros diversas gráficas cuando se habla de costo, este generalmente viene anclado a ciertas características, por ejemplo el tipo de inmuegle del que se habla (apartamento, cama, doble, individual), sin embargo siempre hay algunos valores que vale la pena revisar de cerca.

Similar ocurre con la distancia, aunque en este caso es llamativo como hay algunos que ofrecen muy poca de ésta (distancia) a un precio bastante atractivo. Como en el caso anterior, valdría la pena estudiar a estos individuos.

Si bien no sé sabe cuál dirección es más atractiva que otra el gráfico siguiente nos ofrece una clara imagen de cuales tienden a ser más costosos. Similar ocurre al distrito.

Cabe destacar que como varias de las variables con las que se compara al método son factores, sus gráficas nos ofrecen información clara acerca de la media y lo mucho que tienden a variar los precios dependiendo de ciertas características. Esto es obvio en la gráfica de Precio Vs. Piso, dónde como puede verse las terrazas tienden a variar ampliamente, donde hay unas con muy altos costos y otras con bajos. En el caso de algunas otras variables no hay tanta variación en la información.

Las gráficas finales, que viene del modelo contra todas las variables nos ofrecen información interesante sobre los residuales que quedaron de realizar el modelo.