# Informe_Hogares

PedroFlores

13 de marzo de 2016

```
## Loading required package: jsonlite

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

## Loading required package: xlsx

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

## Loading required package: rJava

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

## Loading required package: xlsxjars

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

## Loading required package: readxl

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

## Loading required package: curl

## Warning: package 'curl' was built under R version 3.2.3
```

Se preparan los datos antes de usar el google_api:

-Se borran columnas que no serán utilizadas:

```
data$Piso <- NULL
data$Foto <- NULL
```

Se quitan los saltos de linea:

```
data$Direccion = str_replace_all(data$Direccion, "[\r\n]" , " ")
data$Distrito = str_replace_all(data$Distrito, "[\r\n]" , " ")
data$Tipo_de_Inmueble = str_replace_all(data$Tipo_de_Inmueble, "[\r\n]" ,
" ")
data$Habitaciones_disponibles =
str_replace_all(data$Habitaciones_disponibles, "[\r\n]" , " ")
data$Precio_mensual = str_replace_all(data$Precio_mensual, "[\r\n]" , "
")
data$Notas = str_replace_all(data$Notas, "[\r\n]" , " ")
```

Se aplica el api de google 2 veces: En el primer ciclo se busca con la dirección y distrito, y las direcciones no encontradas, se iteraran en un segundo for que tiene como destino unicamente la dirección

```
destino = c("Sapienza-Università di Roma")

data$Distancia_al_Campus_en_km <- 0


for(i in seq(1:nrow(data)))
{
  inicio = paste(data$Direccion[i],",",data$Distrito[i],sep="")

  api_url = get_url(inicio, destino, api_key)

  documento = get_data(api_url)

  resultado = parse_data(documento)

  if(resultado$status == "OK"){
  data$Distancia_al_Campus_en_km[i]=resultado$distance$value/1000
  }




}

for(i in seq(1:nrow(data)))
{
  if (data$Distancia_al_Campus_en_km[i]==0) {
  inicio = data$Direccion[i]

  api_url = get_url(inicio, destino, api_key)

  documento = get_data(api_url)

  resultado = parse_data(documento)

  if(resultado$status == "OK"){
    data$Distancia_al_Campus_en_km[i]=resultado$distance$value/1000
  }
                                        }

}
```

Se eliminan las filas de direcciones que no se consiguieron

```
data<-data[!(data$Distancia_al_Campus_en_km==0),]
```

Se numeriza la columna de sexo , 1 para hombres, 2 para mujeres, 3 para ambos

```
sexo <- data$Notas
sexo <- 0



sexo[grepl("(ragazzi)", data$Notas)] =  as.character(1)
sexo[grepl("(ragazze)", data$Notas)] =  as.character(2)
sexo[grepl("(ragazze/i) | (ragazzi/e) | (ragazzi/ragazze) |
(ragazze/ragazzi)", data$Notas)] =  as.character(3)


for(i in 1:length(sexo)){
  if(is.na(sexo[i]))
    sexo[i]=as.character(1)
}



#agrego la columna sexo
data$Sexo = 0
sexo <- as.character(sexo)
data$Sexo <-sexo
```

Se clasifican los tipos de apartamentos: Mini appartamento 0 , Appartamento 1 , monolocale 2

```
tipo <- rep(NA,length(data$Tipo_de_Inmueble))


for(i in 1:length(data$Tipo_de_Inmueble)){
  if(data$Tipo_de_Inmueble[i]=="Mini\nAppartamento" ||
data$Tipo_de_Inmueble[i]=="Mini appartamento" )
    tipo[i]=as.character(0)

  if(data$Tipo_de_Inmueble[i]=="Apartamento" ||
data$Tipo_de_Inmueble[i]=="Apparrtamento" ||
data$Tipo_de_Inmueble[i]=="Appartameno" ||
data$Tipo_de_Inmueble[i]=="Appartamenti" ||
data$Tipo_de_Inmueble[i]=="Appartamento" )
    tipo[i]=as.character(1)

  if(data$Tipo_de_Inmueble[i]=="Monolocale")
    tipo[i]=as.character(2)


 }

for(i in 1:length(tipo)){
  if(is.na(tipo[i]))
```

```
    tipo[i]=as.character(1)
}


data$Tipo_de_Inmueble <- tipo
```

Se crean y clasifican columnas donde hay pagos de: agua, internet, calefacción y condominio

```
agua <- rep(0,length(data$Precio_mensual))
internet <- rep(0,length(data$Precio_mensual))
calefaccion <- rep(0,length(data$Precio_mensual))
condominio <- rep(0,length(data$Precio_mensual))


agua[grepl("(acqua)", data$Precio_mensual)] = 1
internet[grepl("(internet inclu)", data$Precio_mensual)] = 1
calefaccion[grepl("(risca)", data$Precio_mensual)] = 1
condominio[grepl("(condominio)", data$Precio_mensual)] = 1


agua[grepl("(TUTTO INCLUSO)", data$Precio_mensual)] = 1
internet[grepl("(TUTTO INCLUSO)", data$Precio_mensual)] = 1
calefaccion[grepl("(TUTTO INCLUSO)", data$Precio_mensual)] = 1
condominio[grepl("(TUTTO INCLUSO)", data$Precio_mensual)] = 1

agua[grepl("(Tutto incluso)", data$Precio_mensual)] = 1
internet[grepl("(Tutto incluso)", data$Precio_mensual)] = 1
calefaccion[grepl("(Tutto incluso)", data$Precio_mensual)] = 1
condominio[grepl("(Tutto incluso)", data$Precio_mensual)] = 1


data$agua <- agua
data$internet <- internet
data$calefaccion <- calefaccion
data$condominio <- condominio
```

Se eligen las columnas con habitaciones simples y menor precio

```
#precio mensual

dop <- grep("(dop)+.", data$Habitaciones_disponibles)
pos <- grep("(posto)+.", data$Habitaciones_disponibles)
prueba <- data
prueba <- prueba[-dop,]
prueba <- prueba[-pos,]


#elegir el minimo precio
p <- gsub("\\D", "", prueba$Precio_mensual)
```

```
p <- as.numeric(p)
precio_hab <- integer()

for(i in 1:length(p)){
if (str_length(p[i]) > 4){
tam <- str_length(p[i])
aux <- p[i]
min_ <- integer()
  while(tam >= 3){
    num <- 3
    tam <- tam - num
    aux2 <- aux%/%(1*10^tam)
    min_ <- c(min_, aux2)
    aux <- aux%%(1*10^tam)
                }
    precio_hab[i] <- min(min_)
                        }
  else{
    precio_hab[i] <- p[i]
      }
                     }

prueba$Precio_hab <-precio_hab
```

Descartar columnas que no se usaran en el estudio

```
prueba$Distrito <- NULL
pruebaDireccion <- NULL
prueba$Descripcion <- NULL
prueba$Notas <- NULL
```

Descartar columnas que no se usaran en el estudio

```
prueba$Distancia_al_Campus <- as.numeric(prueba$Distancia_al_Campus)
prueba$Sexo <- as.numeric(prueba$Sexo)
prueba$agua <- as.numeric(prueba$agua)
prueba$internet <- as.numeric(prueba$internet)
prueba$calefaccion <- as.numeric(prueba$calefaccion)
prueba$condominio <- as.numeric(prueba$condominio)
prueba$Precio_hab <- as.numeric(prueba$Precio_hab)
```

Separar la data de hombres y mujeres

```
#separar data:  1 hombres.,  2 mujeres

hombres <- prueba
hombres <- hombres[hombres$Sexo ==1 | hombres$Sexo == 3, ]



mujeres <- prueba
mujeres <- mujeres[mujeres$Sexo==2 | mujeres$Sexo==3,  ]
```

```
hombres$Sexo <- NULL
mujeres$Sexo <- NULL


# distancia, agua, inter, calefaccion,condominio, precio
array <- c(6,7,8,9,10,11)


precios_hombre <- hombres$Precio_hab
precios_mujer <- mujeres$Precio_hab


hombres <- as.data.frame(lapply(hombres[array], normalize))
mujeres <- as.data.frame(lapply(mujeres[array], normalize))


hombres$Precio_hab <- precios_hombre
mujeres$Precio_hab <- precios_mujer
```

Hombres, training y testing

```
muestra <-  sample(nrow(hombres), floor(nrow(hombres) * 0.8))
training_hombres <- hombres[muestra, ]

test <- hombres[-muestra, ]
test_hombres <- test




#creacion y aplicacion del modelo

modelo <- lm(training_hombres$Precio_hab ~ ., training_hombres)

test_hombres$regresion <- predict(modelo, newdata = test_hombres)

test_hombres$precision <- abs(test_hombres$Precio_hab -
abs(test_hombres$regresion))
```

Mujeres, training y testing

```
#mujeres training y testing

muestra2 <-  sample(nrow(mujeres), floor(nrow(mujeres) * 0.8))
training_mujeres <- mujeres[muestra2, ]

test2 <- mujeres[-muestra2, ]
test_mujeres <- test2
```

```r
#creacion y aplicacion del modelo
modelo2 <- lm(training_mujeres$Precio_hab ~ ., training_mujeres)

test_mujeres$regresion <- predict(modelo2, newdata = test_mujeres)

test_mujeres$precision <- abs(test_mujeres$Precio_hab -
abs(test_mujeres$regresion))
```

Eleccion de hombres y mujeres

```r
M <- test_hombres[test_hombres$precision == max(test_hombres$precision),
]

View(M[ ,c(5,6)])

F <- test_mujeres[test_mujeres$precision == max(test_mujeres$precision),
]

View(F[ ,c(5,6)])
```