Analisis_911

Cesar Velasco - Belen Paredes

25/01/2021

INTRODUCCIÓN

En este análisis supervisado se utiliza un dataset de las llamadas al 911 del condado de Montgomery en la Commonwealth of Pennsylvania[1]. Los datos se recopilan desde el 12 de diciembre de 2015 hasta el 20 de Julio de 2020.

Para este análisis de datos, hemos decidido utilizar las siguientes técnicas:

- Regresión Lineal

Se utiliza para estimar valores reales en función de variables continuas. Aquí, establecemos la relación entre variables independientes y dependientes ajustando una mejor línea.

- KNN

Clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecer. [3]

GIT: https://github.com/CAVA1611/fid-analisis

```
chooseCRANmirror(graphics=FALSE, ind=1)
knitr::opts_chunk$set(echo = TRUE)

#Instalacion paquetes y librerias

#Paquetes vistos en clases

install.packages("dplyr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/qjx5f99x7l7820pg2h6vnmmm0000gn/T//Rtmp5HRrmG/downloaded_packages
```

```
library(dplyr)
```

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

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

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

```r
install.packages("ggplot2")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/qjx5f99x717820pg2h6vnmmm0000gn/T//Rtmp5HRrmG/downloaded_packages
```

```r
library(ggplot2)
install.packages("caret")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/qjx5f99x717820pg2h6vnmmm0000gn/T//Rtmp5HRrmG/downloaded_packages
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
#Paquetes  investigados

install.packages("lubridate")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/qjx5f99x717820pg2h6vnmmm0000gn/T//Rtmp5HRrmG/downloaded_packages
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```r
#Libreria tidyr
library(tidyr)
#Libreria scales
library(scales)
#
install.packages("caTools")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/qjx5f99x717820pg2h6vnmmm0000gn/T//Rtmp5HRrmG/downloaded_packages
```

```r
library(caTools)
```

```r
#lectura del dataset 911.csv
emergency_calls <- read.csv("/Volumes/T7/Spain/Sevilla/Fundamentos de Datos/envio de revision/FID-analis

#file.exists("/Volumes/T7/Spain/Sevilla/Fundamentos de Datos/envio de #revision/FID-analisis-envio/fid-

# Informacion del dataset
dim(emergency_calls)
```

```
## [1] 663522      9
```

```r
str(emergency_calls)
```

```
## 'data.frame':    663522 obs. of  9 variables:
```

```
##  $ lat      : num  40.3 40.3 40.1 40.1 40.3 ...
##  $ lng      : num  -75.6 -75.3 -75.4 -75.3 -75.6 ...
##  $ desc     : chr  "REINDEER CT & DEAD END;  NEW HANOVER; Station 332; 2015-12-10 @ 17:10:52;" "BRIA
##  $ zip      : int  19525 19446 19401 19401 NA 19446 19044 19426 19438 19462 ...
##  $ title    : chr  "EMS: BACK PAINS/INJURY" "EMS: DIABETIC EMERGENCY" "Fire: GAS-ODOR/LEAK" "EMS: CA
##  $ timeStamp: chr  "2015-12-10 17:10:52" "2015-12-10 17:29:21" "2015-12-10 14:39:21" "2015-12-10 16:
##  $ twp      : chr  "NEW HANOVER" "HATFIELD TOWNSHIP" "NORRISTOWN" "NORRISTOWN" ...
##  $ addr     : chr  "REINDEER CT & DEAD END" "BRIAR PATH & WHITEMARSH LN" "HAWS AVE" "AIRY ST & SWEDE
##  $ e        : int  1 1 1 1 1 1 1 1 1 1 ...
```

```
head(emergency_calls)
```

```
##        lat       lng
## 1 40.29788 -75.58129
## 2 40.25806 -75.26468
## 3 40.12118 -75.35198
## 4 40.11615 -75.34351
## 5 40.25149 -75.60335
## 6 40.25347 -75.28324
##                                                                                desc
## 1           REINDEER CT & DEAD END;  NEW HANOVER; Station 332; 2015-12-10 @ 17:10:52;
## 2 BRIAR PATH & WHITEMARSH LN;  HATFIELD TOWNSHIP; Station 345; 2015-12-10 @ 17:29:21;
## 3                          HAWS AVE; NORRISTOWN; 2015-12-10 @ 14:39:21-Station:STA27;
## 4              AIRY ST & SWEDE ST;  NORRISTOWN; Station 308A; 2015-12-10 @ 16:47:36;
## 5    CHERRYWOOD CT & DEAD END;  LOWER POTTSGROVE; Station 329; 2015-12-10 @ 16:56:52;
## 6              CANNON AVE & W 9TH ST;  LANSDALE; Station 345; 2015-12-10 @ 15:39:04;
##     zip                   title           timeStamp               twp
## 1 19525  EMS: BACK PAINS/INJURY 2015-12-10 17:10:52       NEW HANOVER
## 2 19446 EMS: DIABETIC EMERGENCY 2015-12-10 17:29:21 HATFIELD TOWNSHIP
## 3 19401     Fire: GAS-ODOR/LEAK 2015-12-10 14:39:21        NORRISTOWN
## 4 19401  EMS: CARDIAC EMERGENCY 2015-12-10 16:47:36        NORRISTOWN
## 5    NA          EMS: DIZZINESS 2015-12-10 16:56:52  LOWER POTTSGROVE
## 6 19446        EMS: HEAD INJURY 2015-12-10 15:39:04          LANSDALE
##                          addr e
## 1     REINDEER CT & DEAD END 1
## 2 BRIAR PATH & WHITEMARSH LN 1
## 3                   HAWS AVE 1
## 4         AIRY ST & SWEDE ST 1
## 5   CHERRYWOOD CT & DEAD END 1
## 6      CANNON AVE & W 9TH ST 1
```

Realizamos la lectura de los datos a analizar, verificamos la relevancia de las columnas y su significado.

Explicación de los atributos:

-lat: Latitud del lugar de la llamada -Ing: Longitud del lugar de la llamada -desc: Descripción de la llamada de emergencia -zip: Código postal -Title: Tipo de llamada de emergencia(EMS: Servicio médico de emergencia, Fire: Accidente de incendio, Trafic: Accidente de tráfico) -TimeStamp: AAAA-MM-DD HH: MM: SS -Twp: Municipio -addr: Dirección -e: Variable ficticia (siempre 1)

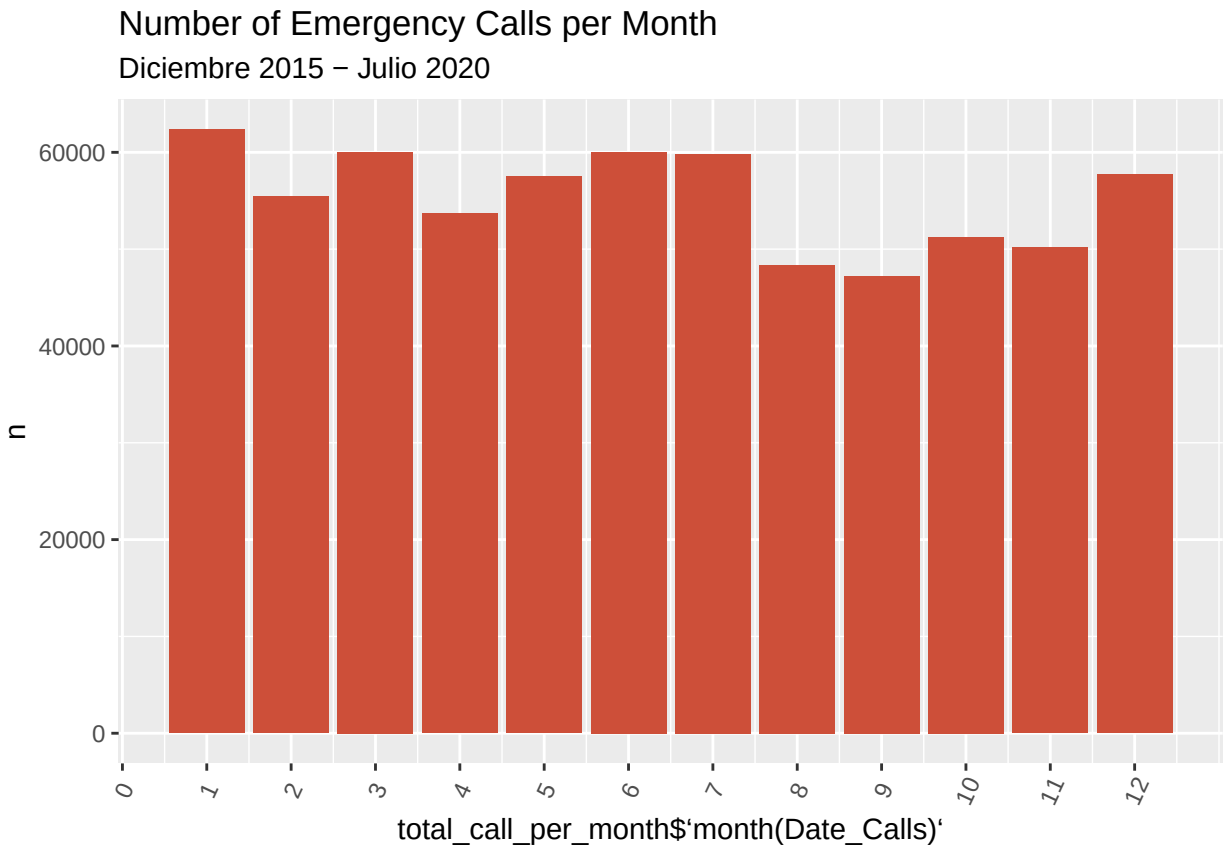
```
#Creción de nuevas variables a partir de variables existentes y eliminacion  de  variables innecesarias

#Separacion la columna titlle en dos (Types,Subtypes)
emergency_calls <- separate(emergency_calls,title,c("Types","Subtypes"),sep=":")

#Separacion la columna timeStamp en dos (Date_Calls, Hour_Calls)
emergency_calls <- separate(emergency_calls,timeStamp,c("Date_Calls","Hour_Calls"),sep=" ")
```

```r
# Elimanacion columna "e"
emergency_calls$e <- NULL

#Nueva Dimension del DataSet
dim(emergency_calls)
```

```
## [1] 663522     10
```

```r
#Cambio de clase a "date" para trabajo con Fechas (lubridate)
emergency_calls$Date_Calls <- as.Date(emergency_calls$Date_Calls)

emergency_calls$Hour_Calls <- hms(emergency_calls$Hour_Calls)


num_filas <- dim(emergency_calls)
```

```r
#LLAMADAS MENSUALES

#Grafico del numero de llamadas por mes (duracte los 5 años)
 total_call_per_month <-emergency_calls%>%
  group_by(month(Date_Calls))%>%
  count()%>%
  arrange(desc(n))


ggplot(total_call_per_month, aes(x=total_call_per_month$`month(Date_Calls)`, y=n)) +
  geom_bar(stat ="identity", fill="tomato3") +
  scale_x_continuous(breaks=seq(0, 12, 1)) +
  labs(title="Number of Emergency Calls per Month",
       subtitle="Diciembre 2015 - Julio 2020") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```




```
total_call_per_month
```

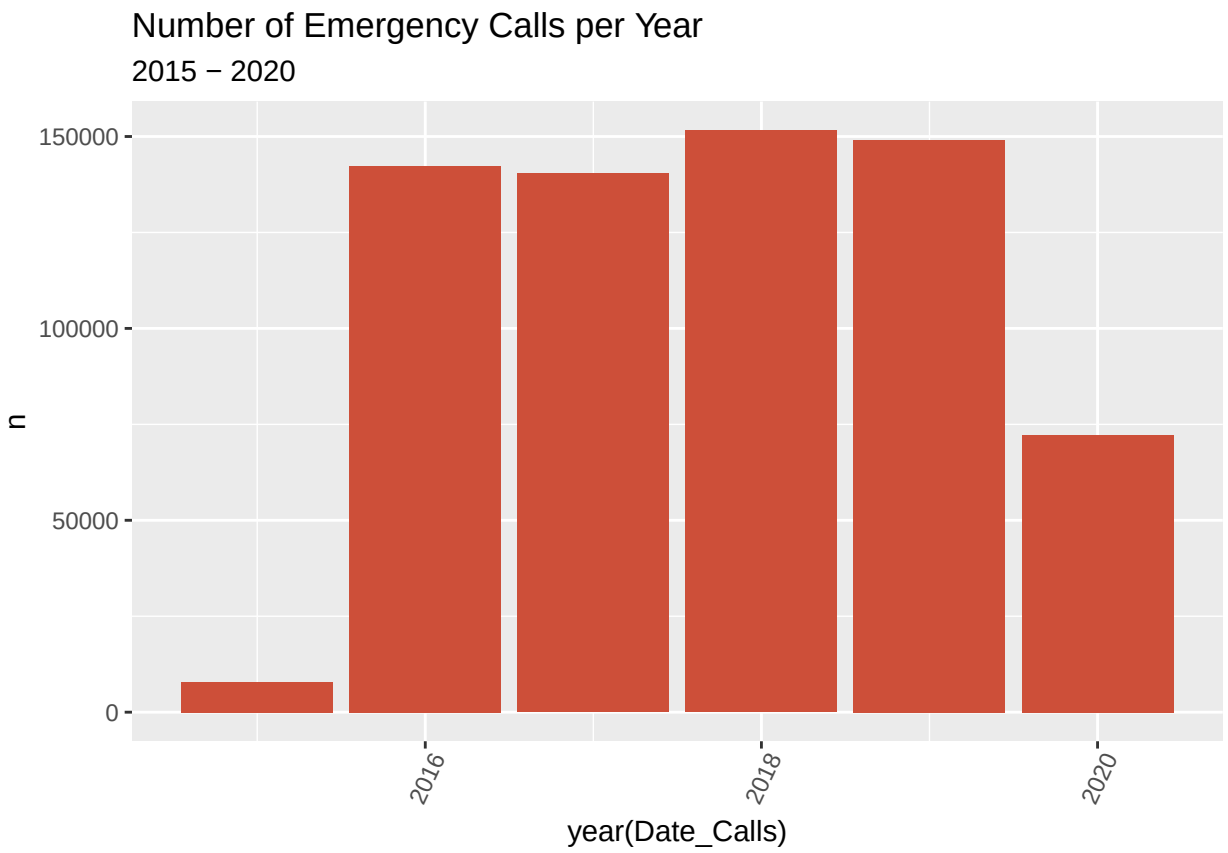
```
## # A tibble: 12 x 2
## # Groups:   month(Date_Calls) [12]
##    `month(Date_Calls)`     n
##                  <dbl> <int>
##  1                   1 62336
##  2                   6 60034
##  3                   3 60027
##  4                   7 59832
##  5                  12 57692
##  6                   5 57509
##  7                   2 55427
##  8                   4 53671
##  9                  10 51240
## 10                  11 50182
## 11                   8 48359
## 12                   9 47213
```

Llamadas Mensuales

-La frecuencia de las llamadas al 911 en Enero es mayor mientras que en noviembre la frecuencia baja

```
#numero de llamadas por años

total_calls_per_year <- emergency_calls%>%
  group_by(year(Date_Calls))%>%
  count()
```

```
ggplot(total_calls_per_year, aes(x=`year(Date_Calls)`, y=n)) +
  geom_bar(stat ="identity", fill="tomato3") +
  labs(title="Number of Emergency Calls per Year",
       subtitle="2015 - 2020") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```




```
total_calls_per_year
```

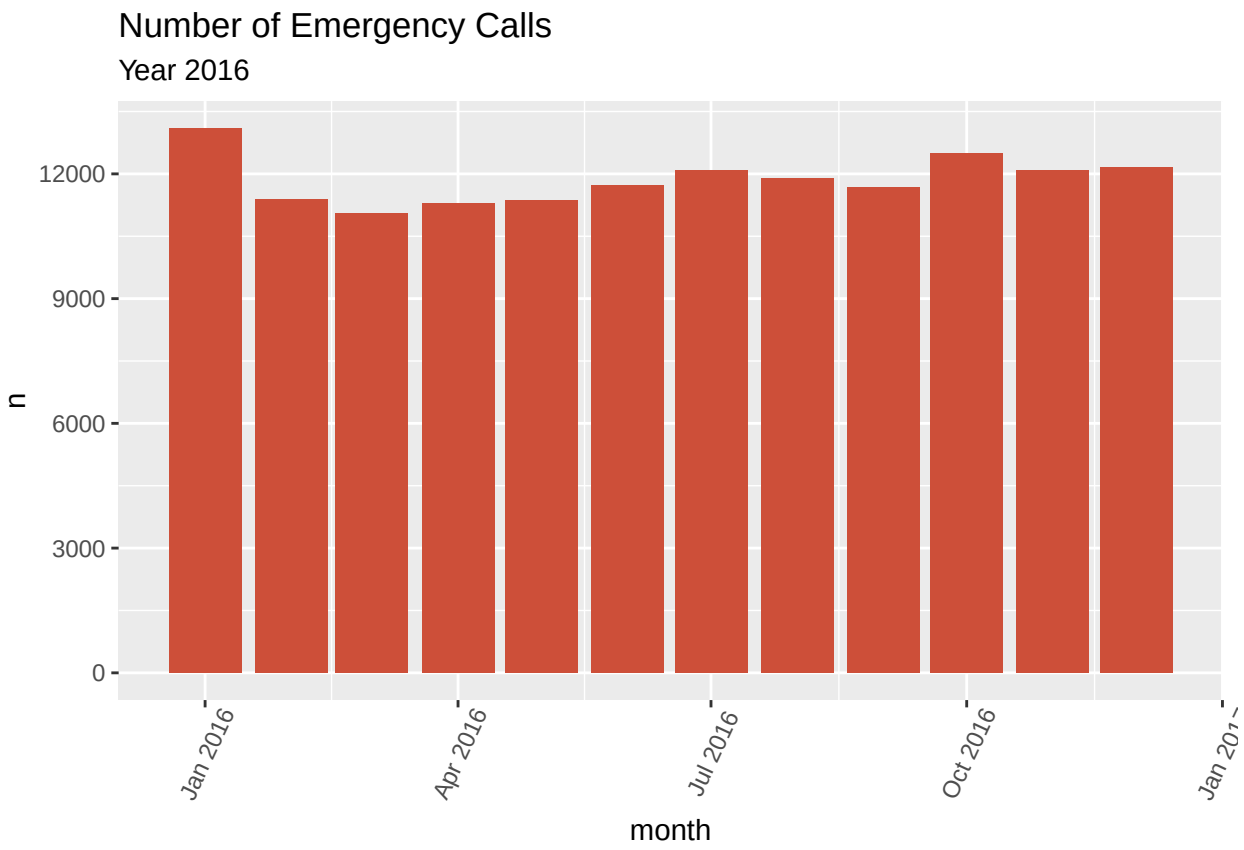
```
## # A tibble: 6 x 2
## # Groups:   year(Date_Calls) [6]
##   `year(Date_Calls)`      n
##                <dbl>  <int>
## 1               2015   7916
## 2               2016 142360
## 3               2017 140343
## 4               2018 151527
## 5               2019 149118
## 6               2020  72258
```

```
#class(emergency_calls$Hour_Calls)
#emergency_calls$Date_Calls <- as.Date(emergency_calls$Date_Calls)
#class(emergency_calls$Date_Calls)
```

Llamadas Años

-En el año 2015 y año 2020 el registro de llamadas corresonde a 1 mes y 7 meses respectivamente, se puede observar en el grafico que el mayor numero de llamadas se dio en el año 2018 y el menor numero de llamadas en el año 2017

```r
#Grafico numero de llamadas por mes para el 2016
calls_per_month_2016 <- emergency_calls%>%
  select(Date_Calls)%>%
  filter(Date_Calls>= "2016-01-01" & Date_Calls<="2016-12-31")%>%
  group_by(month=floor_date(Date_Calls, "month"))%>%
  count()

ggplot(calls_per_month_2016, aes(x=month, y=n)) +
  geom_bar(stat ="identity", fill="tomato3") +
  labs(title="Number of Emergency Calls",
       subtitle="Year 2016") +
  scale_y_continuous(breaks=seq(0, 15000, 3000))+
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```

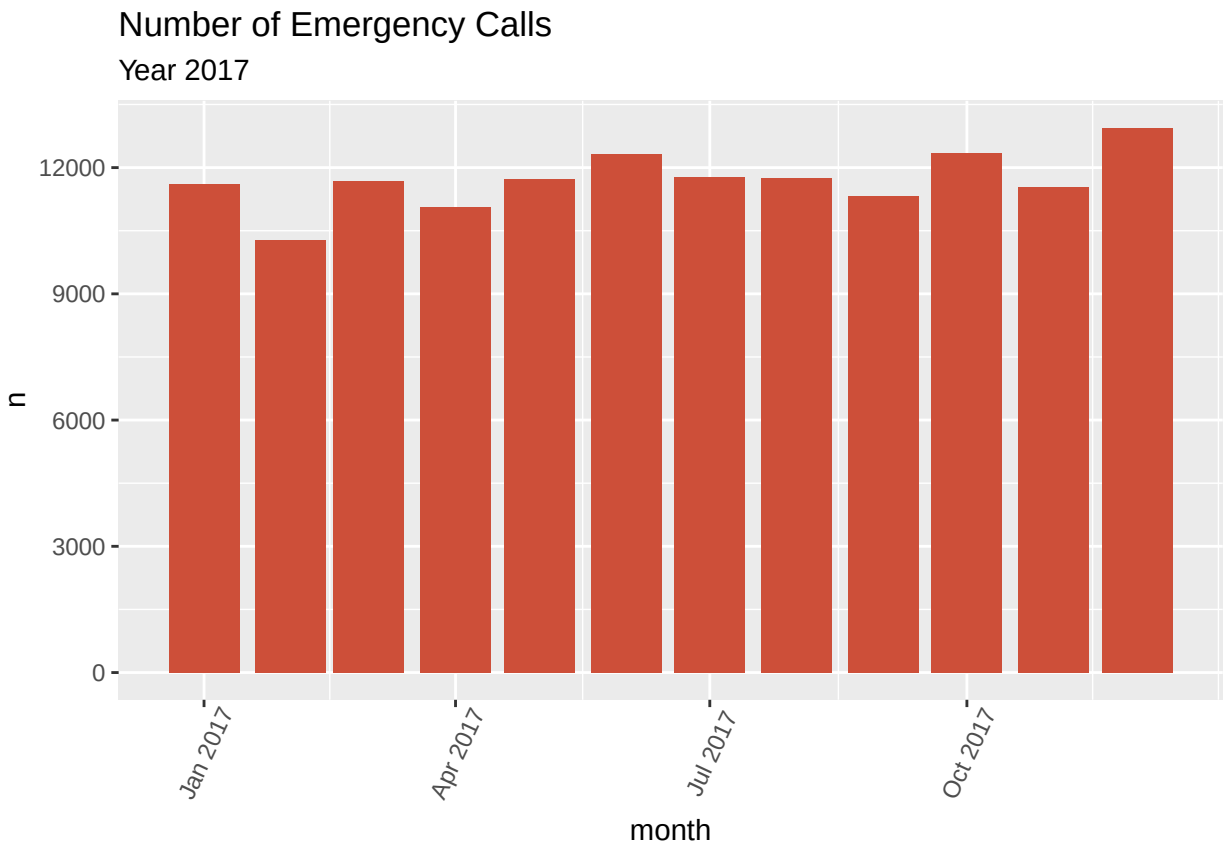


Llamadas Año 2016

-En enero del 2016 se registra el mayor numero de llamadas 13096 de un total de 142360

```r
#Grafico numero de llamadas por mes para el 2017
calls_per_month_2017 <- emergency_calls%>%
  select(Date_Calls)%>%
  filter(Date_Calls>= "2017-01-01" & Date_Calls<="2017-12-31")%>%
  group_by(month=floor_date(Date_Calls, "month"))%>%
  count()

ggplot(calls_per_month_2017, aes(x=month, y=n)) +
  geom_bar(stat ="identity", fill="tomato3") +
  labs(title="Number of Emergency Calls",
       subtitle="Year 2017") +
```

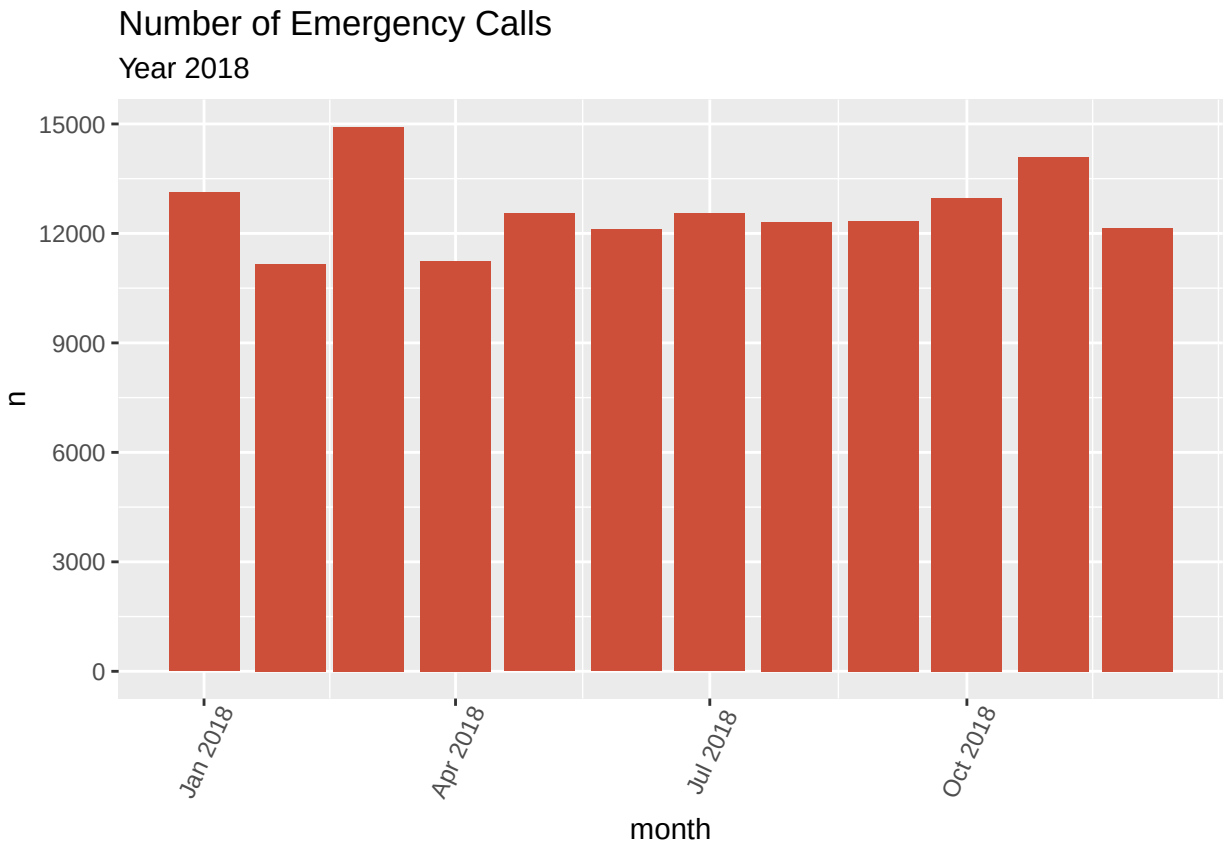
```
   theme(axis.text.x = element_text(angle=65, vjust=0.6))+
  scale_y_continuous(breaks=seq(0, 15000, 3000))
```



Llamadas Año 2017

-En diciembre del 2017 se registra el mayor numero de llamadas 12941de un total de 140343

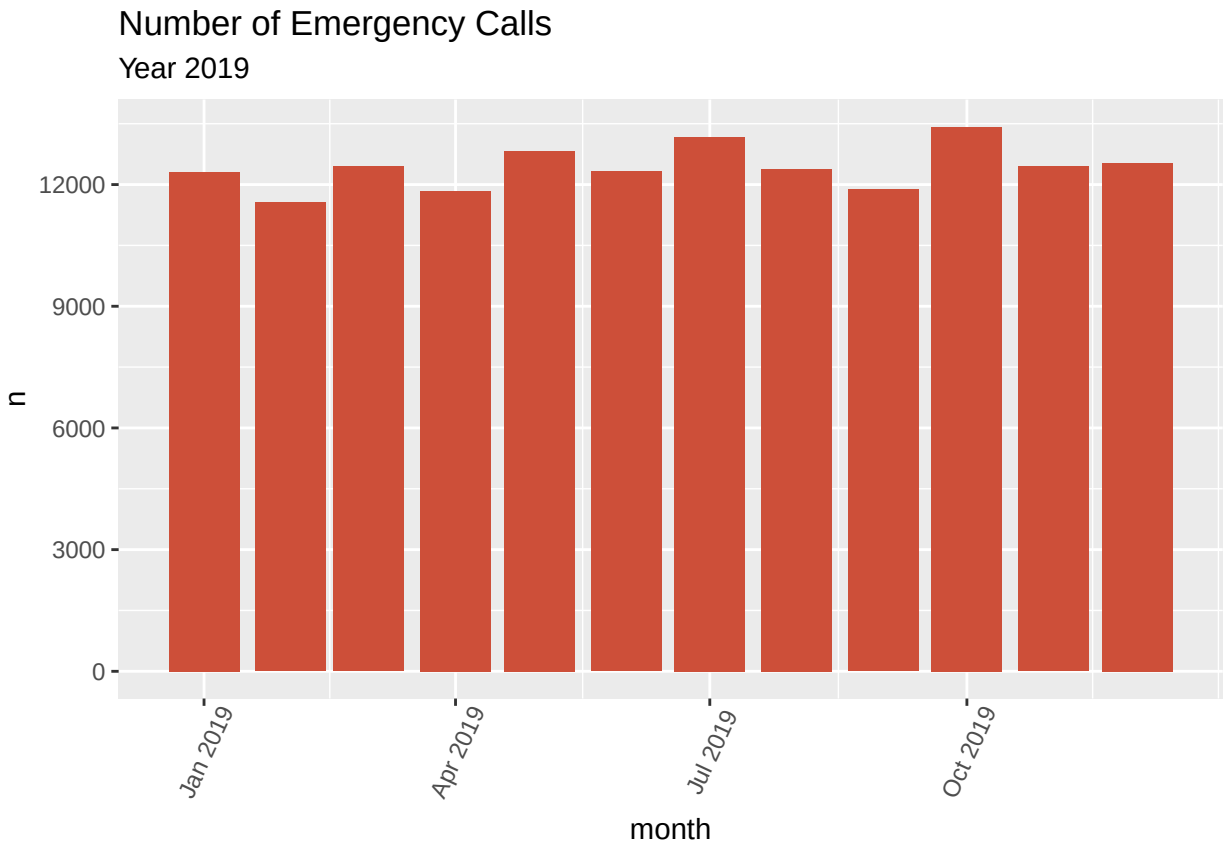
```
#Grafico numero de llamadas por mes para el 2018
calls_per_month_2018 <- emergency_calls%>%
  select(Date_Calls)%>%
  filter(Date_Calls>= "2018-01-01" & Date_Calls<="2018-12-31")%>%
  group_by(month=floor_date(Date_Calls, "month"))%>%
  count()

ggplot(calls_per_month_2018, aes(x=month, y=n)) +
  geom_bar(stat ="identity", fill="tomato3") +
  labs(title="Number of Emergency Calls",
       subtitle="Year 2018") +
  scale_y_continuous(breaks=seq(0, 15000, 3000))+
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



Llamadas Año 2018

-En marzo del 2018 se registra el mayor numero de llamadas 14923 de un total de 151527

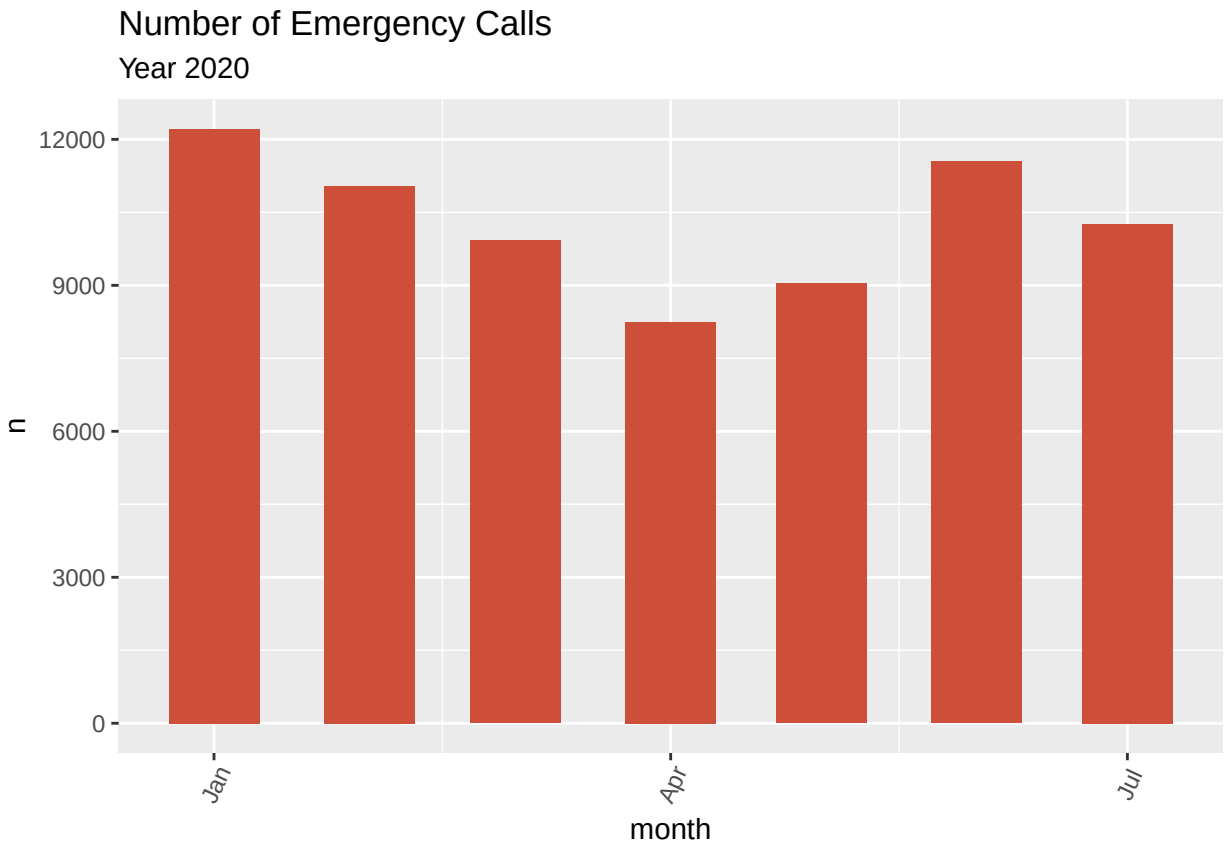
```
#Grafico numero de llamadas por mes para el 2019
calls_per_month_2019 <- emergency_calls%>%
  select(Date_Calls)%>%
  filter(Date_Calls>= "2019-01-01" & Date_Calls<="2019-12-31")%>%
  group_by(month=floor_date(Date_Calls, "month"))%>%
  count()

ggplot(calls_per_month_2019, aes(x=month, y=n)) +
  geom_bar(stat ="identity", fill="tomato3") +
  labs(title="Number of Emergency Calls",
       subtitle="Year 2019") +
  scale_y_continuous(breaks=seq(0, 15000, 3000))+
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



Llamadas Año 2019

-En octubre del 2019 se registra el mayor numero de llamadas 13425 de un total de 149118

```
#Grafico numero de llamadas por mes para el 2020
calls_per_month_2020 <- emergency_calls%>%
  select(Date_Calls)%>%
  filter(Date_Calls>= "2020-01-01" & Date_Calls<="2020-12-31")%>%
  group_by(month=floor_date(Date_Calls, "month"))%>%
  count()

ggplot(calls_per_month_2020, aes(x=month, y=n)) +
  geom_bar(stat ="identity", fill="tomato3", width=18) +
  labs(title="Number of Emergency Calls",
       subtitle="Year 2020") +
  scale_y_continuous(breaks=seq(0, 15000, 3000))+
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```



Llamadas Año 2020

-En enero del 2020 se registra el mayor numero de llamadas 12208 de un total de 72258

```
#contamos el numero de llamadas por Township

 aux <- emergency_calls %>%
  group_by(twp) %>%
  count()%>%
  arrange(desc(n))
head(aux)
```

```
## # A tibble: 6 x 2
## # Groups:   twp [6]
##   twp              n
##   <chr>        <int>
## 1 LOWER MERION 55490
## 2 ABINGTON     39947
## 3 NORRISTOWN   37633
## 4 UPPER MERION 36010
## 5 CHELTENHAM   30574
## 6 POTTSTOWN    27387
```

```
#aux contiene el numero total de twp (69)
aux
```

```
## # A tibble: 69 x 2
## # Groups:   twp [69]
##    twp                  n
##    <chr>            <int>
```

```
##  1 LOWER MERION     55490
##  2 ABINGTON         39947
##  3 NORRISTOWN       37633
##  4 UPPER MERION     36010
##  5 CHELTENHAM       30574
##  6 POTTSTOWN        27387
##  7 UPPER MORELAND   22932
##  8 LOWER PROVIDENCE 22476
##  9 PLYMOUTH         20116
## 10 UPPER DUBLIN     18862
## # ... with 59 more rows
```

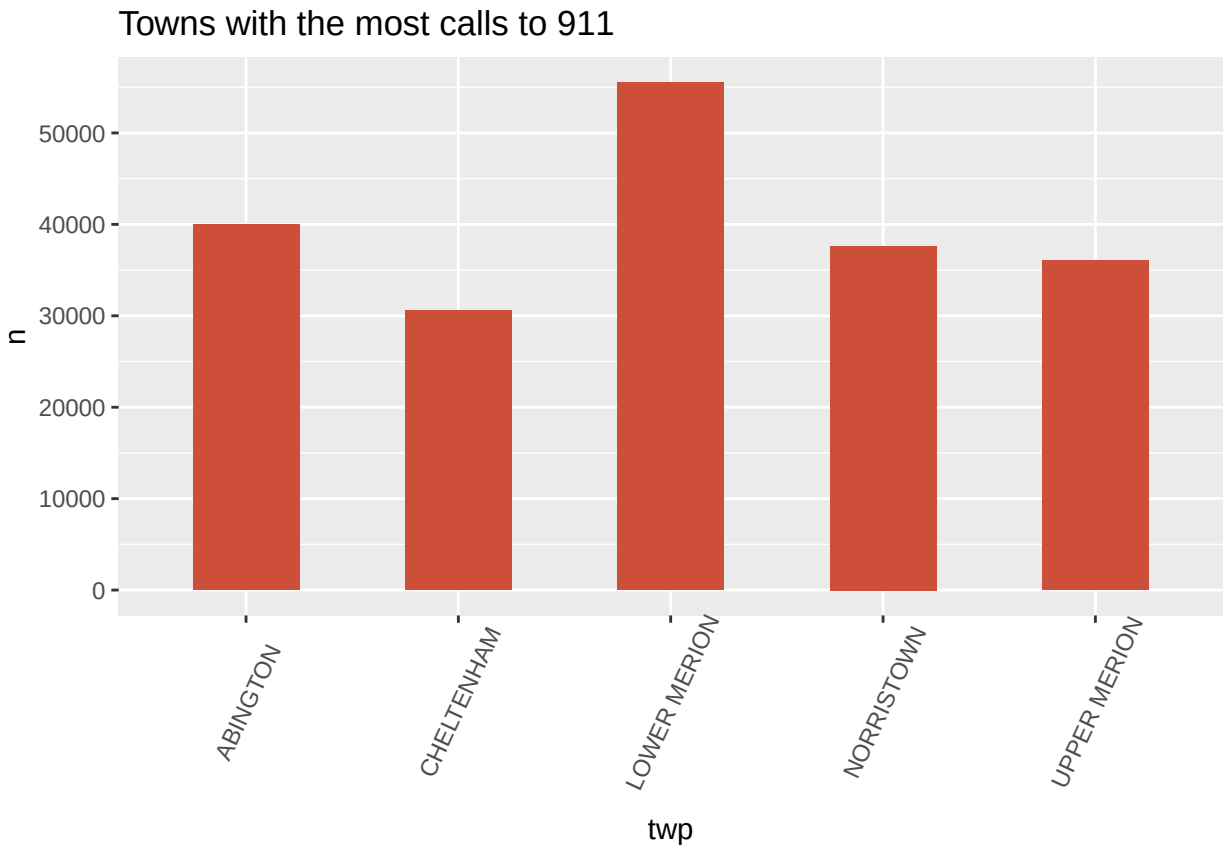
```
# We found the town with the most emergency calls to 911
mas_llamadas<-aux %>%
  head(1,1)

mas_llamadas
```

```
## # A tibble: 1 x 2
## # Groups:   twp [1]
##   twp              n
##   <chr>        <int>
## 1 LOWER MERION 55490
```

```
print(paste("The town with the most 911 calls in Montgomery County, Pennsylvania es : ", mas_llamadas$tw
```

```
## [1] "The town with the most 911 calls in Montgomery County, Pennsylvania es :  LOWER MERION con 5549
```

```
# Graficamos las 5 ciudades con mas llamadas al 911

ggplot(aux [1:5,], aes(x=twp, y=n)) +
  geom_bar(stat ="identity", fill="tomato3", width=0.5) +
 scale_y_continuous(breaks=seq(0, 70000, 10000)) +
  labs(title="Towns with the most calls to 911") +
  theme(axis.text.x = element_text(angle=65, vjust=0.6))
```




```
dim(aux)
```

```
## [1] 69  2
```

```
#Grafico de las direcciones que han realizado mas llamadas

direcciones_masllamadas <- emergency_calls %>%
  group_by(addr) %>%
  count()%>%
  arrange(desc(n))
head(direcciones_masllamadas)
```

```
## # A tibble: 6 x 2
## # Groups:   addr [6]
##   addr                                 n
##   <chr>                            <int>
## 1 SHANNONDELL DR & SHANNONDELL BLVD  7285
## 2 MAIN ST & OLD SUMNEYTOWN PIKE      2576
## 3 THE FAIRWAY  & RYDAL RD            1986
## 4 EAGLEVILLE RD & SUNDERLAND DR      1618
## 5 EVERGREEN RD & W LIGHTCAP RD       1591
## 6 GERMANTOWN PIKE & HANNAH AVE       1557
```

```
#direcciones_llamadas contiene el numero total de add (41292)
direcciones_masllamadas
```

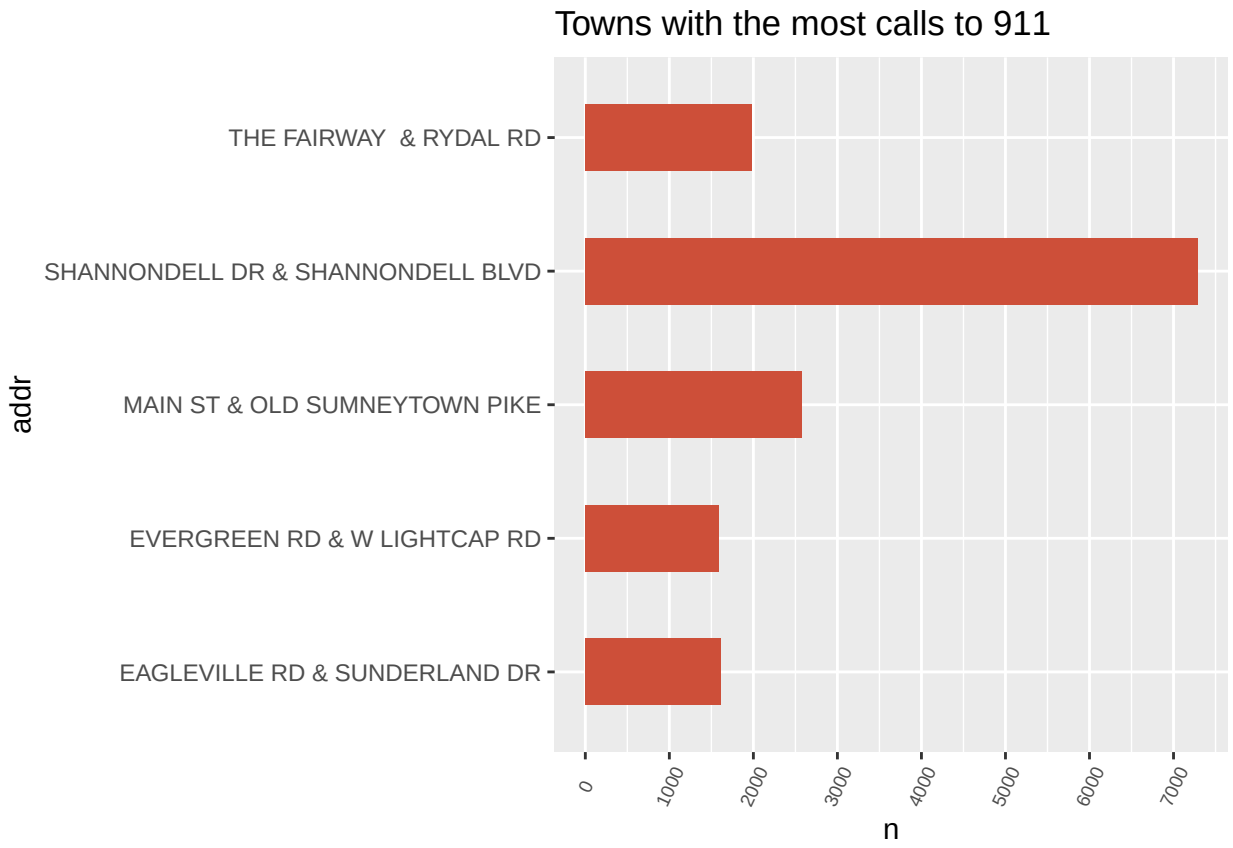
```
## # A tibble: 41,292 x 2
## # Groups:   addr [41,292]
##    addr                                   n
```

```
##    <chr>                                <int>
##  1 SHANNONDELL DR & SHANNONDELL BLVD     7285
##  2 MAIN ST & OLD SUMNEYTOWN PIKE         2576
##  3 THE FAIRWAY  & RYDAL RD               1986
##  4 EAGLEVILLE RD & SUNDERLAND DR         1618
##  5 EVERGREEN RD & W LIGHTCAP RD          1591
##  6 GERMANTOWN PIKE & HANNAH AVE          1557
##  7 GULPH RD & KIRK AVE                   1440
##  8 BLACK ROCK RD & S TRAPPE RD           1425
##  9 DAVISVILLE RD & PENNYPACK RD          1367
## 10 SCHUYLKILL EXPY & WEADLEY RD OVERPASS  1287
## # ... with 41,282 more rows
```

```
fil <- dim(direcciones_masllamadas)#


# Encontramos la dirección  con mas llamadas de emergencia al 911
top1 <- direcciones_masllamadas %>%
  head(1,1)

#Direccion con mas llamadas al 911  con respecto al total de direcciones (41292)
print(paste("The address with the most calls to 911: ", top1$addr,"con",top1$n, "llamadas","(Total addr
```

```
## [1] "The address with the most calls to 911:  SHANNONDELL DR & SHANNONDELL BLVD con 7285 llamadas (T
```

```
# Grafica de las direcciones VS cantidad de llamadas
ggplot(direcciones_masllamadas [1:5,], aes(y=addr, x=n)) +
  geom_bar(stat ="identity", fill="tomato3", width=0.5) +
  theme(axis.text.x = element_text(angle=65, vjust=0.53, size =7 ))+
 scale_x_continuous(breaks=seq(0, 7000, 1000))+
  labs(title="Towns with the most calls to 911")
```




```
# Encontramos el numero de llamadas segun el tipo de emergencia
freqt.calls <-as.data.frame(table(emergency_calls$Types))

colnames(freqt.calls) <- c('Type_emergencia', 'Number_of_calls')

#Tipo de accidente conmas llamadas
print(paste("The type of accident with the most calls to 911: ", freqt.calls$`Type_emergencia`[1],"con"
```

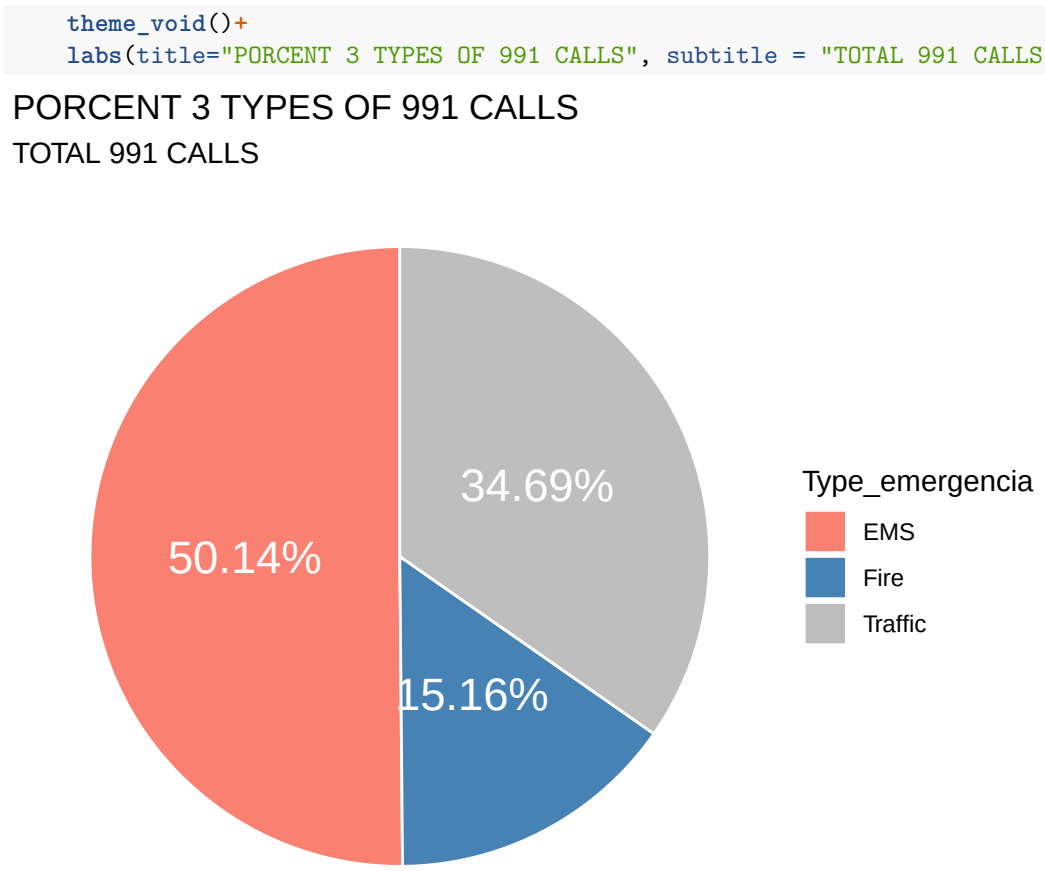
```
## [1] "The type of accident with the most calls to 911:  EMS con 332692 llamadas"
```

```
# Encontramos en porcentaje el numero de llamadas segun su tipo de emergencia
porcentaje_typescall <- freqt.calls %>%
mutate (Porcentaje = (freqt.calls$`Number_of_calls`/(num_filas[1])*100) )
head(porcentaje_typescall)
```

```
##   Type_emergencia Number_of_calls Porcentaje
## 1             EMS          332692   50.14031
## 2            Fire          100622   15.16483
## 3         Traffic          230208   34.69486
```

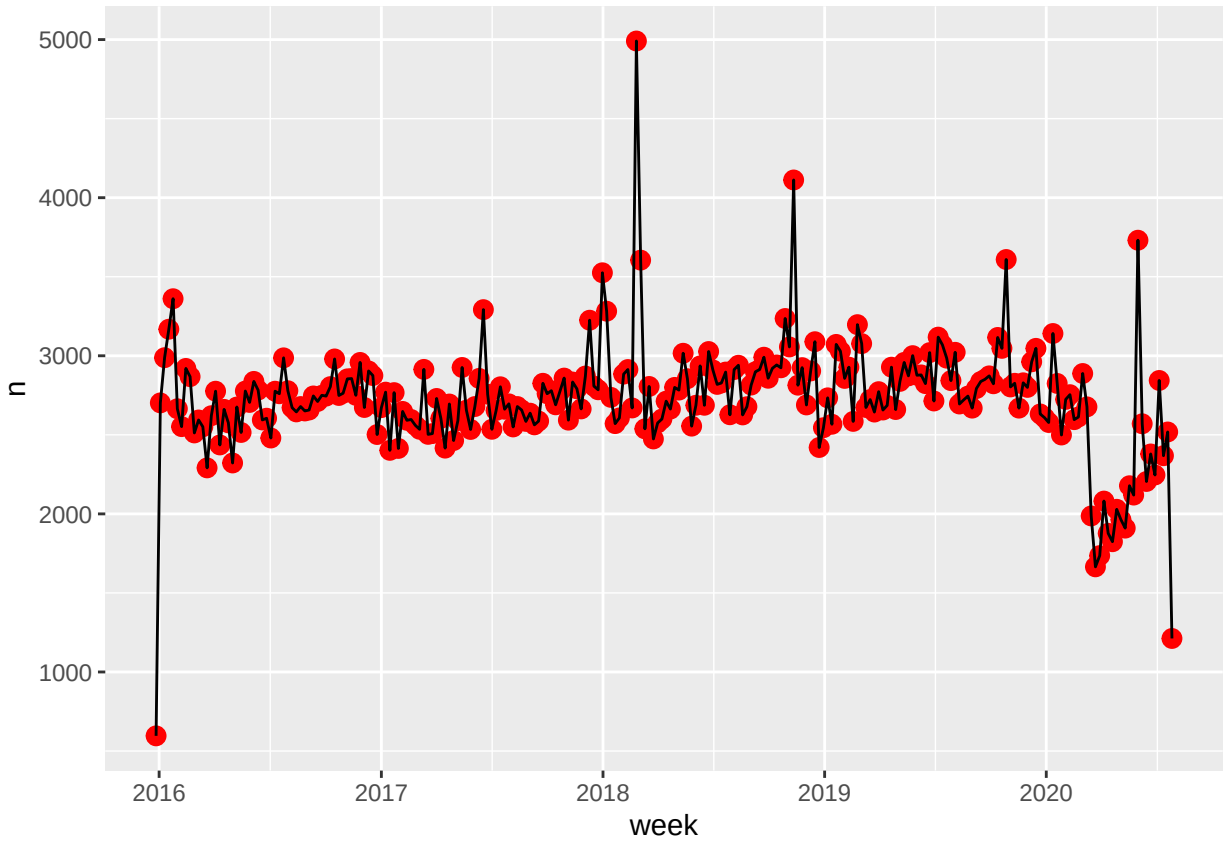
```
# Representacion grafica del porcentaje de llamadas segun el tipo de emergencia

ggplot(porcentaje_typescall,aes(x="",y=Porcentaje, fill=Type_emergencia,))+
  geom_bar(stat = "identity",
           color="white")+
    geom_text(aes(label=percent(Porcentaje/100,0.01)),
              position=position_stack(vjust=0.5),color="white",size=6)+
  coord_polar(theta = "y")+
  scale_fill_manual(values=c("salmon","steelblue","gray"))+
```

```
    theme_void()+
    labs(title="PORCENT 3 TYPES OF 991 CALLS", subtitle = "TOTAL 991 CALLS" )
```



EMS ha representa el 50,14% y le siguen Tráfico (34,69%) y Fuego (15,16%) .

```
#METODO SUPERVISADO

# REGRESION LINEAL

#En el eje x se representa la semana y e el eje y el numero de llamadas, se pretende encontrar una func

total_calls_per_week <- emergency_calls%>%
   select(Date_Calls)%>%
  filter(Date_Calls>= "2016-01-01" & Date_Calls<="2020-12-31")%>%
  group_by(week=floor_date(Date_Calls, "week"))%>%
  count()

#Grafico del numero de llamadas por semana
ggplot(total_calls_per_week, aes(x=week, y=n)) +
  geom_point(colour = "red", size = 3) +
    geom_path()
```



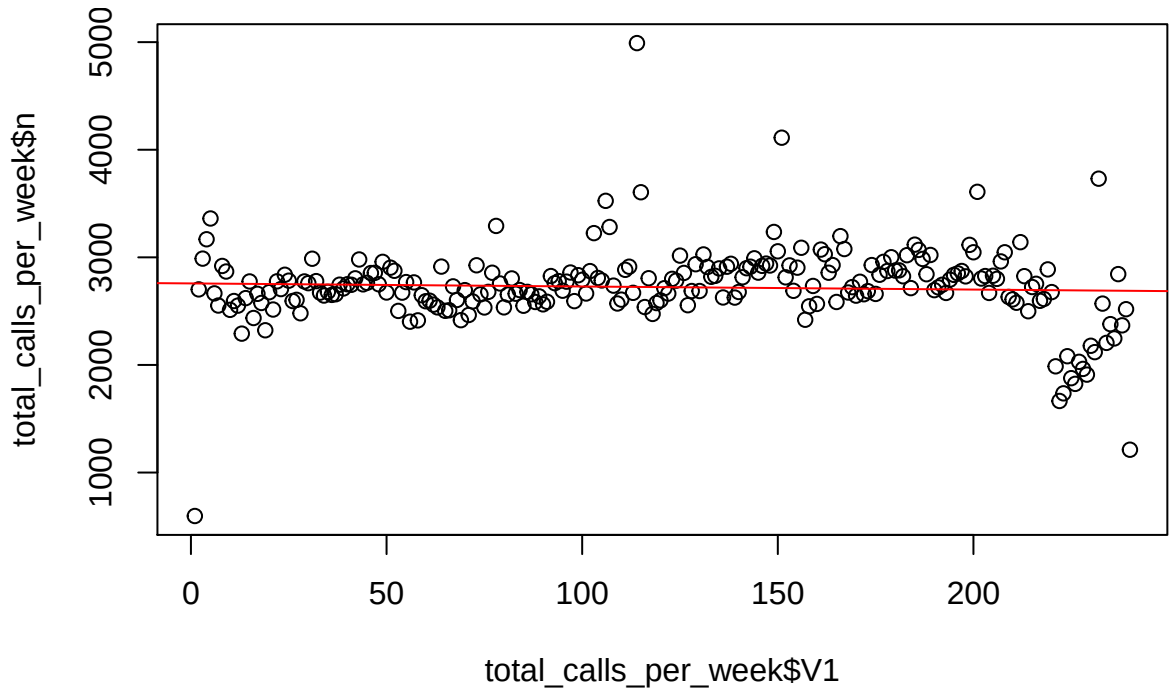

```
#Enumeracion de los meses
num_of_week <- matrix(1:240, nrow=240, ncol=1)
num_of_week <- as.data.frame(num_of_week)

# Union del numero de mes con el data set total_call_per_week
# total_calls_per_week dataset numero de semana (v1) y numero de llamadas (n)
aux <- cbind(num_of_week, total_calls_per_week)
total_calls_per_week <- aux
total_calls_per_week$week<- NULL


#Separacion en training y test
set.seed(4532)
pcte <- sample(2, nrow(total_calls_per_week), replace = TRUE, prob = c(0.8,0.2))
train_data_calls <- total_calls_per_week[pcte==1,]
test_data_calls <- total_calls_per_week[pcte==2,]

#Plot del numero de llamadas por semana desde Diciembre 2015 hasta Julio 2020
plot(x=total_calls_per_week$V1, y=total_calls_per_week$n)

#regresison lineal aplicada al training data
mylm <- lm(n ~ ., data = train_data_calls)
mylm$coefficients
```

```
##  (Intercept)           V1
## 2757.6662726   -0.2860435
```

```r
#Prediccion para la semana (revisar archivo aux)
new <- test_data_calls[16,]
prediccion <-  predict(mylm, new)
prediccion
```

```
##       76
## 2735.927
```

```r
abline(mylm$coefficients, col="red")
```



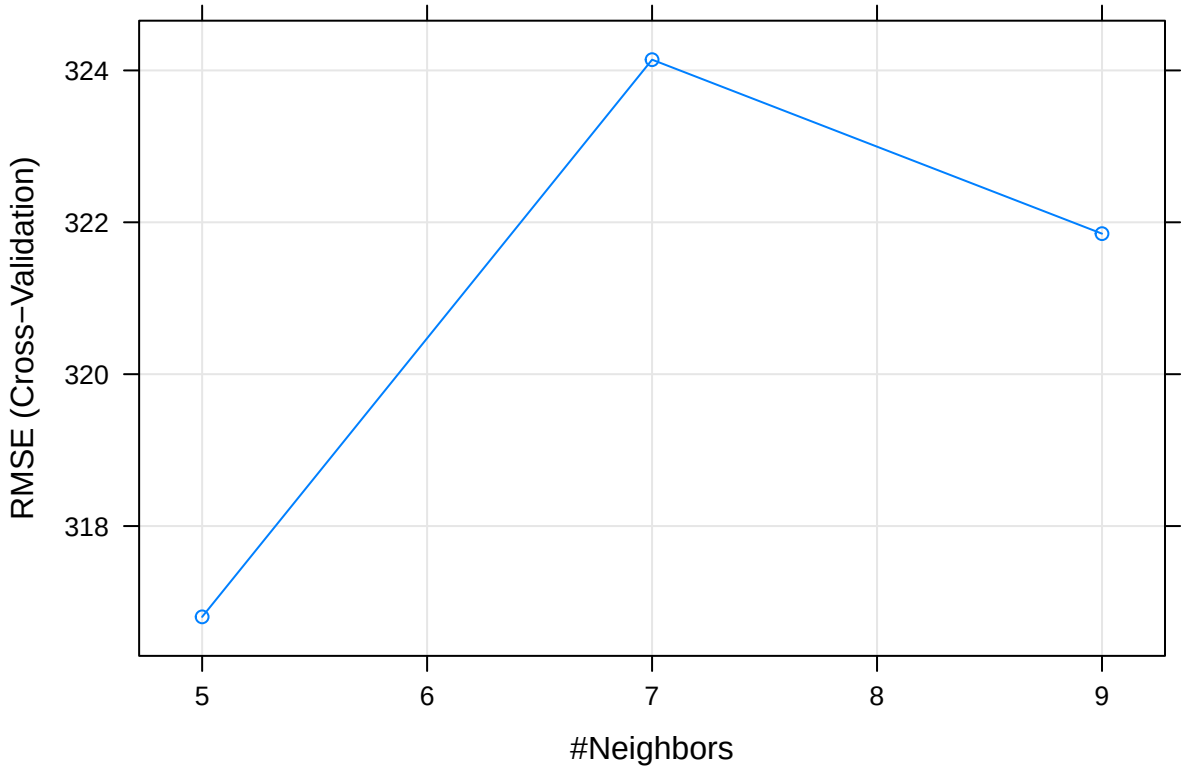

```r
#Informacion sobre el metodo aplicado
summary(mylm)
```

```
##
## Call:
## lm(formula = n ~ ., data = train_data_calls)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2161.38  -133.83     7.51   165.84  2265.94
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2757.6663    57.7891  47.720   <2e-16 ***
## V1            -0.2860     0.4119  -0.694    0.488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 391.7 on 192 degrees of freedom
## Multiple R-squared:  0.002505,   Adjusted R-squared:  -0.00269
## F-statistic: 0.4823 on 1 and 192 DF,  p-value: 0.4882
```

```r
# KNN utilizando el paquete caret

#
set.seed(582)
fitControl <- trainControl(
              method = "cv", #Metodo utilizado cross-validation
              number = 5) # numero de submuestras
#Aplicamos la funcion train()
calls_knn <- train(n ~., total_calls_per_week,
                method = "knn",
                trControl= fitControl)

calls_knn
```

```
## k-Nearest Neighbors
##
## 240 samples
##   1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 191, 192, 192, 192, 193
## Resampling results across tuning parameters:
##
##   k  RMSE      Rsquared   MAE
##   5  316.8043  0.2911661  197.4639
##   7  324.1414  0.2528265  199.6497
##   9  321.8506  0.2578855  197.9931
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 5.
```

```r
plot(calls_knn)
```



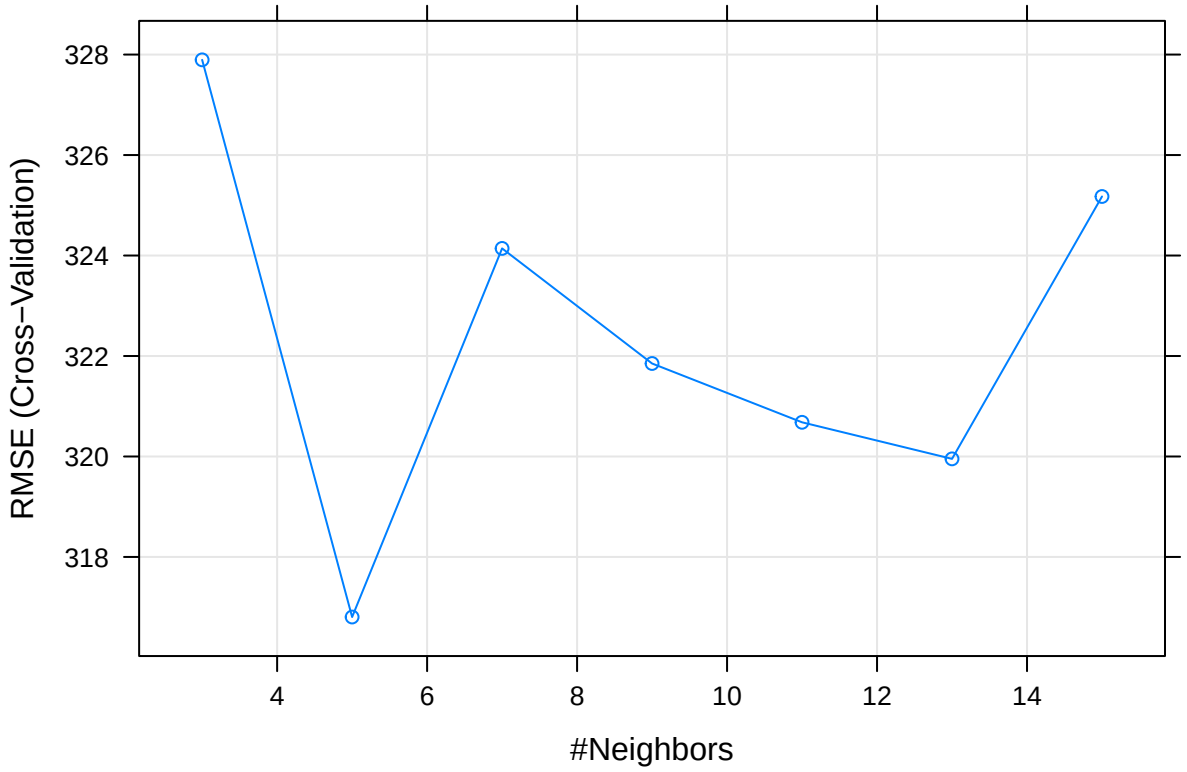
Para visualizar de mejor manera el valos de k.

```
# Se realizara con mas valores de K para apreciar de mejor manera la grfaica

set.seed(582)
valores <- expand.grid(k = seq(3, 15, 2))
fitControl <- trainControl(
              method = "cv", #Metodo utilizado cross-validation
              number = 5) # numero de submuestras

#Aplicamos la funcion train()
calls_knn <- train(n ~., total_calls_per_week,
                method = "knn",
                trControl= fitControl,
                tuneGrid = valores)


calls_knn
```

```
## k-Nearest Neighbors
##
## 240 samples
##   1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 191, 192, 192, 192, 193
## Resampling results across tuning parameters:
##
##   k   RMSE      Rsquared   MAE
```

```
##     3  327.8947  0.2755673  200.2328
##     5  316.8043  0.2911661  197.4639
##     7  324.1414  0.2528265  199.6497
##     9  321.8506  0.2578855  197.9931
##    11  320.6803  0.2664714  198.8116
##    13  319.9526  0.2681025  199.8264
##    15  325.1750  0.2490981  202.7284
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 5.
```

```r
plot(calls_knn)
```




```r
#Utilizado para power BI

write.csv(emergency_calls,file="emergency_call_PBI.csv")

accidentsEMS <- emergency_calls %>%

  filter(Types== "EMS")%>%
   group_by(Subtypes) %>%
  count()%>%
 arrange(desc(n))

accidentsfire <- emergency_calls %>%

  filter(Types== "Fire")%>%
   group_by(Subtypes) %>%
  count()%>%
 arrange(desc(n))
```

```
accidentsTraffic <- emergency_calls %>%

  filter(Types== "Traffic")%>%
   group_by(Subtypes) %>%
  count()%>%
 arrange(desc(n))
```

CONCLUSIONES

En base a la informacion obtenida por el metodo de Regresión Lineal se concluye que no se puede predecir el numero de llamadas semanales al 911, puesto que sus valores de R-squared: 0.002505, y p-value: 0.4882, valores que se encuentran lejanos al ideal cercano a 1 y menor a 0.05 repectivamente

El método de KNN se ajusta de mejor manera a nuestros datos, puesto que su valor R-squared 0.2911661, es más cercano a 1.

Los dos métodos utilizados no han proporcionado una adecuada predicción del numero de llamadas semanales, por lo que se deberia probar con otros métodos.

```
#######
```

BIBLIOGRAFIA

[1] Dataset : https://www.kaggle.com/mchirico/montcoalert [2] metodos: https://riptutorial.com/es/machine-learning/example/32209/regresion-lineal

[3] lubridate :https://cran.r-project.org/web/packages/lubridate/vignettes/lubridate.html https://www.doctormetrics.com/lubridate-manipulando-fechas-en-r/ https://www.rdocumentation.org/packages/lubridate/versions/1.7.9.2

[4]tidy: https://rpubs.com/jaortega/151936 [5]Escales: https://rpubs.com/luis_bolanos/537899 para manejar porcentajes [6]Sample: https://www.rdocumentation.org/packages/sp/versions/1.4-5/topics/spsample

[6]ggplot-bar: https://www.r-graph-gallery.com/218-basic-barplots-with-ggplot2.html https://rstudio-pubs-static.s3.amazonaws.com/345827_ef9eb854d51943ca9433ff814e1fdf2a.html http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html [7]caret: https://www.rdocumentation.org/packages/caret/versions/6.0-86/topics/trainControl