

Graficos iniciales

Juan felipe Múnera Vergara

10/6/2019

Carga librerias

```
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
```

Carga de datos

```
##Lectura de las 6 bases de datos desde el directorio

getwd() #CHECKEAR QUE EL DIRECTORIO SEA EL DONDE ESTA EL ARCHIVO

## [1] "C:/Users/JF_Mra/Documents/Unal S6/Series/Git_repo/series"

files_names= list.files(pattern="*.csv", path = paste(getwd(),"/Data",sep=""))
path <- paste(getwd(),"/Data/",files_names[-6],sep="")
df <- do.call(rbind, lapply(path, function(x) read.csv(x, stringsAsFactors = FALSE,encoding = "UTF-8",

#La base de datos del 2019 tiene una columna extra, ha de leerse aparte
#Contiene datos hasta Junio
df_2019 <- read.csv(paste(getwd(),"/Data/",files_names[6],sep=""), header=T,
                    encoding="UTF-8")

#Seleccionamos las columnas relevantes
cols <- c("FECHA", "HORA", "DIA", "MES", "PERIODO", "DIA_NOMBRE", "COMUNA")
df <- df[, cols]
df_2019 <- df_2019[,cols]

df_accidentes <- rbind(df, df_2019)

#Se toman solo las 16 comunas correspondientes al área metropolitana
#Que coinciden con ser las comunas con mayor accidentalidad
```

```

comunas <- count(df_accidentes, COMUNA)
comunas <- head(comunas[order(-comunas$n),][1], 16)

df_accidentes <- df_accidentes[df_accidentes$COMUNA %in% as.vector(unlist(comunas)),]

## Se transformas characters a String.
df_accidentes <- df_accidentes%>%
  mutate(
    DIA_NOMBRE = factor(DIA_NOMBRE),
    COMUNA = factor(COMUNA),
    FECHA= as.Date(FECHA, "%Y-%m-%dT00:00:00.000Z")
  )

```

Agrupamiento a nivel Mensual

```

#Agrupación de los accidentes a escala mensual
df_mes <- df_accidentes%>%group_by(MES, PERIODO)%>%
  summarise(
    accidentes = n()
  )

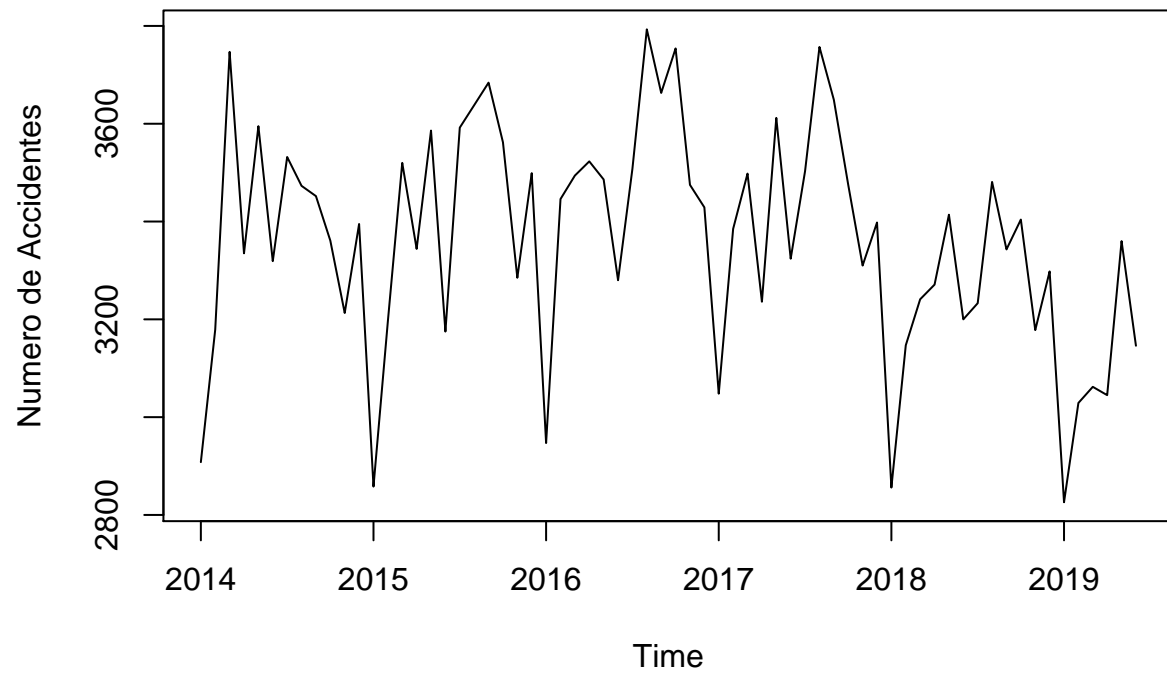
#Ordenamiento cronologico para crear la serie de tiempo:
df_mes <- df_mes[order(df_mes$PERIODO, df_mes$MES), ]

ts_mes <- ts(df_mes$accidentes, start=c(2014,1),frequency=12)

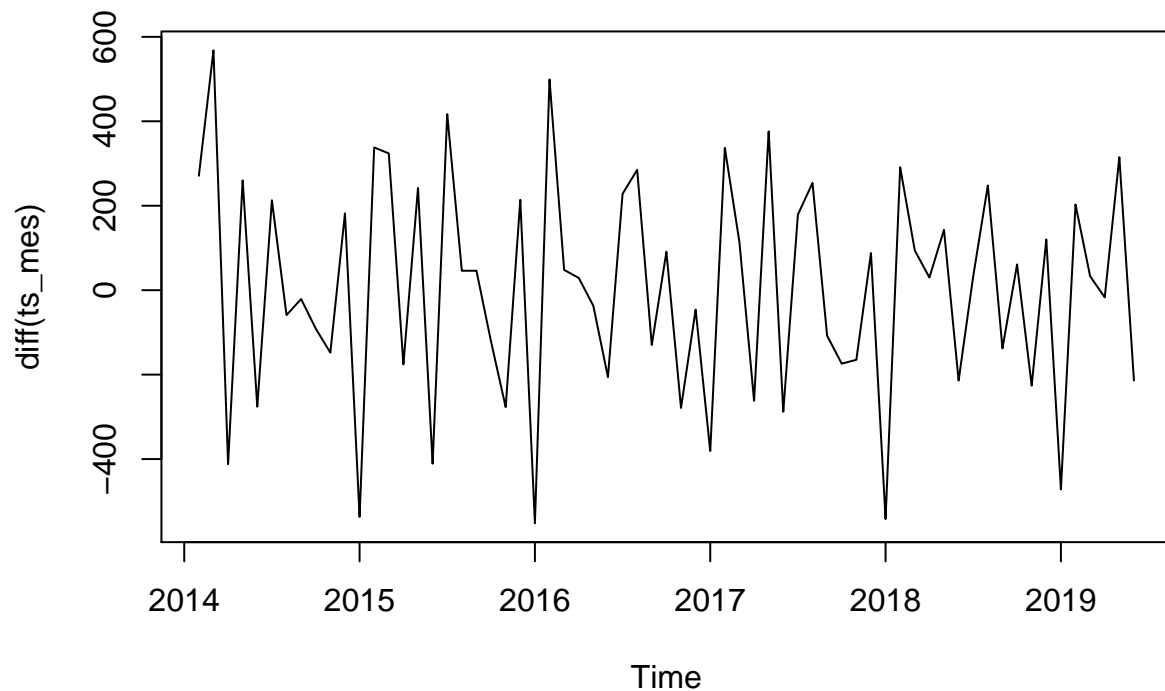
plot.ts(ts_mes, ylab="Numero de Accidentes",
  main="Numero de accidentes Mensuales en Medellín")

```

Numero de accidentes Mensuales en Medellín



```
#Componente estacional:  
plot(diff(ts_mes))
```



Agrupamiento a nivel semanal

```
#Agrupación de los accidentes a escala semanal
df_sem <- df_accidentes%>%
  mutate(SEMANA= strftime(FECHA, format = "%V"))%>%
  group_by(SEMANA, PERIODO)%>%
  summarise(
    accidentes = n()
  )

#Ordenamiento cronológico para crear la serie de tiempo:
df_sem <- df_sem[order(df_sem$PERIODO, df_sem$SEMANA), ]

library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.6.1
```

```
##
```

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

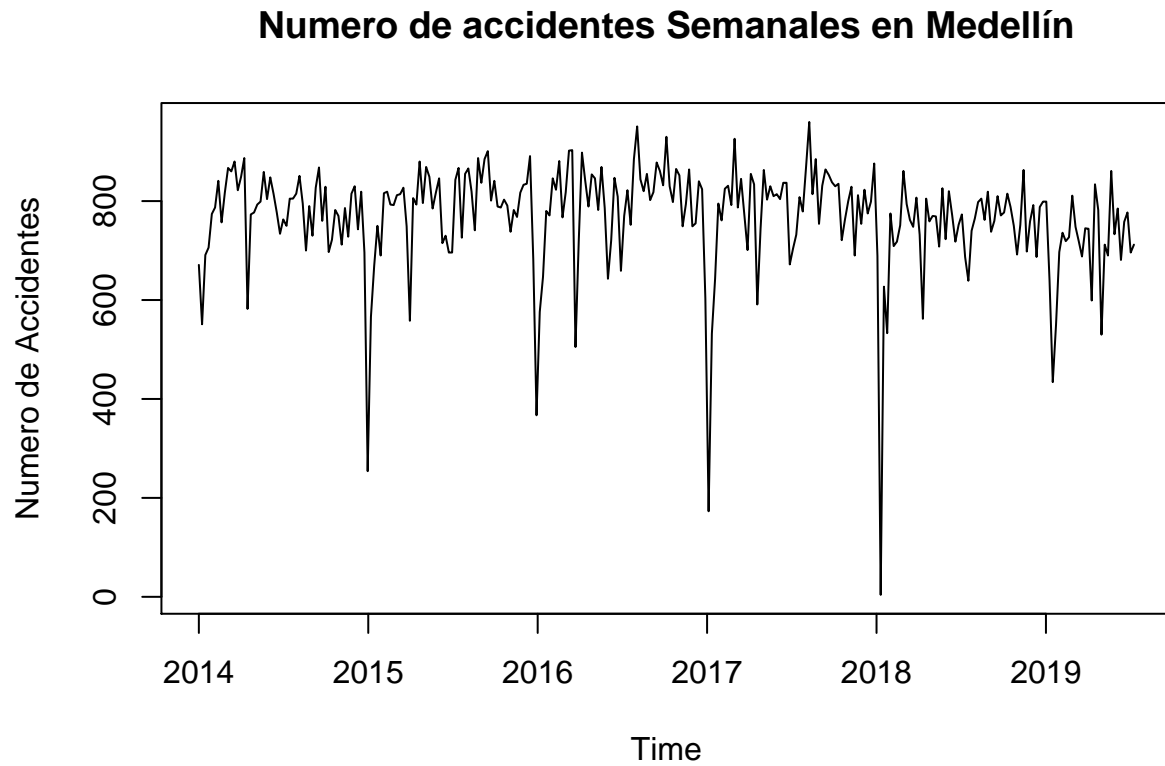
```
## The following object is masked from 'package:base':
```

```
##
```

```
## date
```

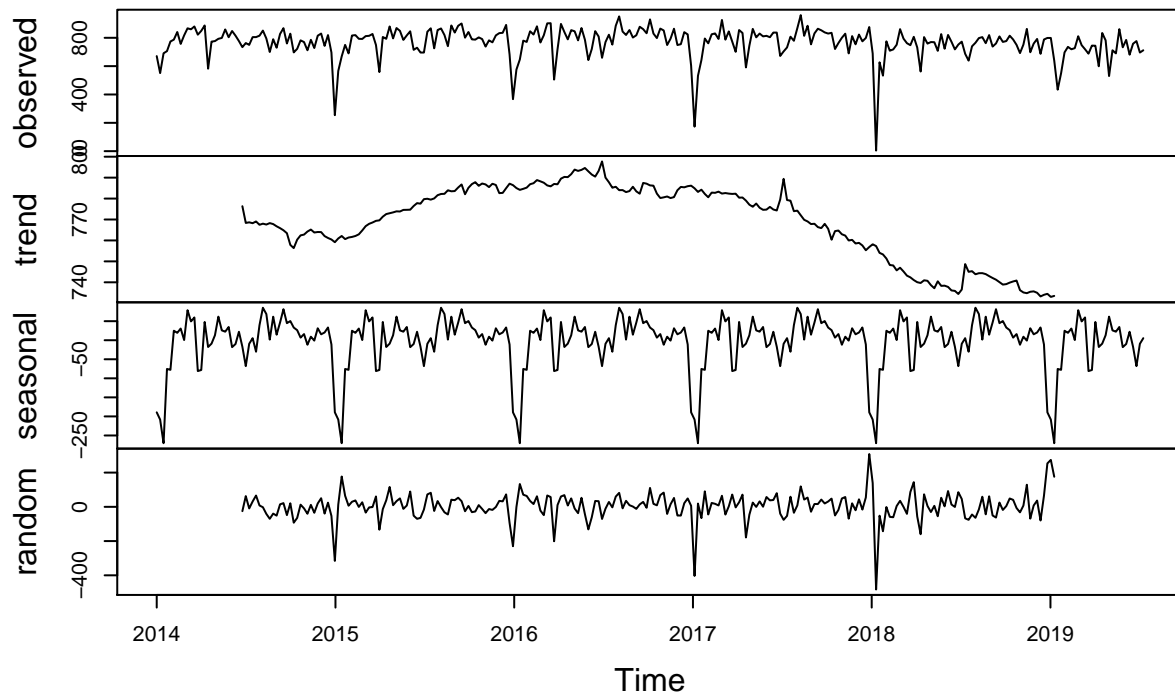
```
ts_semanal <- ts(df_sem$accidentes, freq=365.25/7,
  start=decimal_date(ymd("2014-01-01")))

plot.ts(ts_semanal, ylab="Numero de Accidentes",
  main="Numero de accidentes Semanales en Medellín")
```

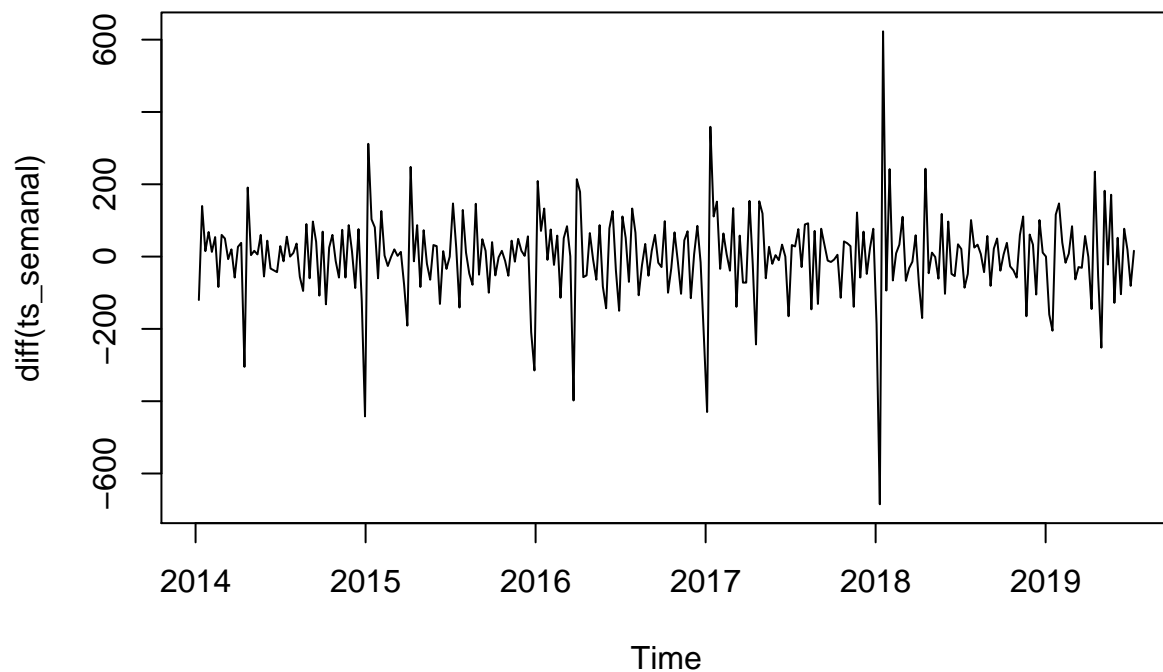


```
plot(decompose(ts_semanal, type="additive"))
```

Decomposition of additive time series



```
plot(diff(ts_semanal))
```



Agrupamiento por día

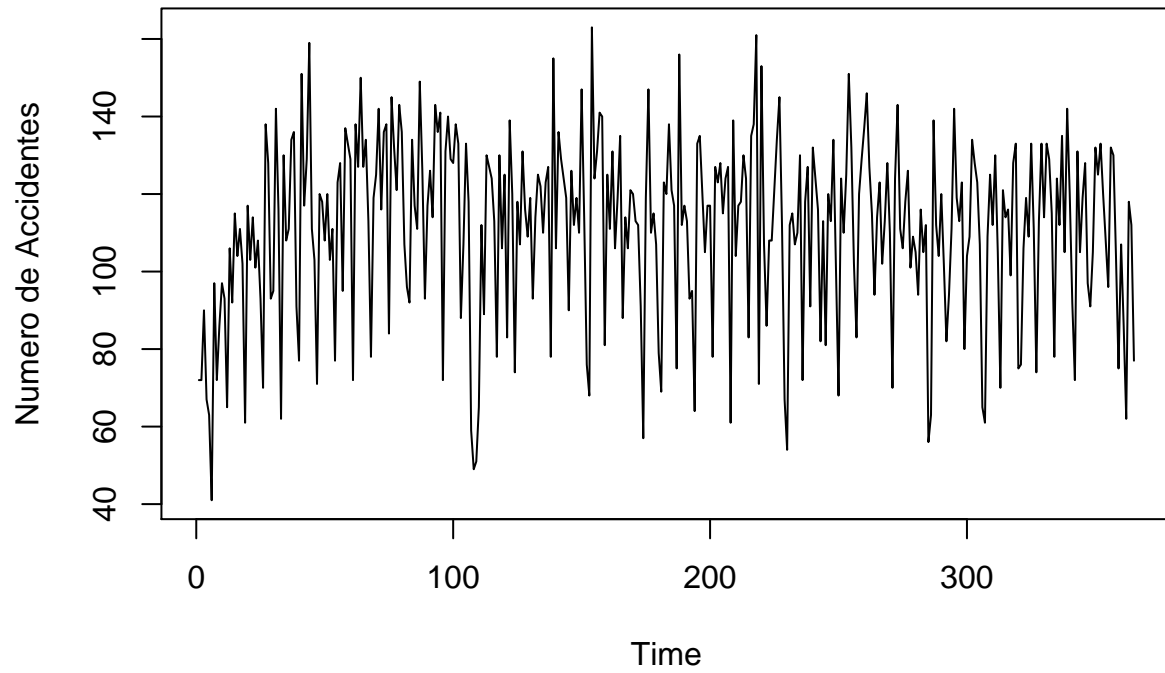
```
#Agrupación de los accidentes a escala diaria
df_dia <- df_accidentes%>%
  group_by(DIA, MES, PERIODO)%>%
  summarise(
    accidentes = n()
  )

#Ordenamiento cronologico para crear la serie de tiempo:
df_dia <- df_dia[order(df_dia$PERIODO,df_dia$MES, df_dia$DIA), ]

ts_dia <- ts(df_dia$accidentes, freq=365,
  start=decimal_date(ymd("2014-01-01")))

plot.ts(ts_dia[1:365], ylab="Numero de Accidentes",
  main="Accidentes diarios en Medellín 2014")
```

Accidentes diarios en Medellín 2014



```
plot.ts(ts_dia[366:(2*365)], ylab="Numero de Accidentes",  
        main="Accidentes diarios en Medellín 2015")
```


Accidentes diarios en Medellín 2015

