NYPD Data

pablo

4/29/2023

Importing the Data

We read the data directly from the below URL, ad display the frist rows to get an idea of the schema.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
url_i<-"https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"

d<-read.csv(url_i)
head(d)
```

```
##   INCIDENT_KEY OCCUR_DATE OCCUR_TIME     BORO LOC_OF_OCCUR_DESC PRECINCT
## 1    228798151 05/27/2021   21:30:00   QUEENS                       105
## 2    137471050 06/27/2014   17:40:00    BRONX                        40
## 3    147998800 11/21/2015   03:56:00   QUEENS                       108
## 4    146837977 10/09/2015   18:30:00    BRONX                        44
## 5     58921844 02/19/2009   22:58:00    BRONX                        47
## 6    219559682 10/21/2020   21:36:00 BROOKLYN                        81
##   JURISDICTION_CODE LOC_CLASSFCTN_DESC LOCATION_DESC STATISTICAL_MURDER_FLAG
## 1                 0                                                   false
## 2                 0                                                   false
## 3                 0                                                    true
## 4                 0                                                   false
## 5                 0                                                    true
## 6                 0                                                    true
##   PERP_AGE_GROUP PERP_SEX PERP_RACE VIC_AGE_GROUP VIC_SEX       VIC_RACE
## 1                                          18-24       M          BLACK
## 2                                          18-24       M          BLACK
## 3                                          25-44       M          WHITE
```

```
## 4                                          <18       M WHITE HISPANIC
## 5          25-44        M     BLACK         45-64       M          BLACK
## 6                                          25-44       M          BLACK
##   X_COORD_CD Y_COORD_CD Latitude Longitude
## 1    1058925   180924.0 40.66296 -73.73084
## 2    1005028   234516.0 40.81035 -73.92494
## 3    1007668   209836.5 40.74261 -73.91549
## 4    1006537   244511.1 40.83778 -73.91946
## 5    1024922   262189.4 40.88624 -73.85291
## 6    1004234   186461.7 40.67846 -73.92795
##                                      Lon_Lat
## 1 POINT (-73.73083868899994 40.662964620000025)
## 2  POINT (-73.92494232599995 40.81035186300006)
## 3  POINT (-73.91549174199997 40.74260663300004)
## 4  POINT (-73.91945661499994 40.83778200300003)
## 5  POINT (-73.85290950899997 40.88623791800006)
## 6 POINT (-73.92795224099996 40.678456718000064)
```

```
length(d$INCIDENT_KEY)
```

```
## [1] 27312
```

Next we check the class type of each column,

```
sapply(d,typeof)
```

```
##            INCIDENT_KEY              OCCUR_DATE              OCCUR_TIME
##               "integer"             "character"             "character"
##                    BORO       LOC_OF_OCCUR_DESC                PRECINCT
##             "character"             "character"               "integer"
##       JURISDICTION_CODE      LOC_CLASSFCTN_DESC           LOCATION_DESC
##                "integer"             "character"             "character"
## STATISTICAL_MURDER_FLAG          PERP_AGE_GROUP                PERP_SEX
##             "character"             "character"             "character"
##               PERP_RACE           VIC_AGE_GROUP                 VIC_SEX
##             "character"             "character"             "character"
##                VIC_RACE              X_COORD_CD              Y_COORD_CD
##             "character"                "double"                "double"
##                Latitude               Longitude                 Lon_Lat
##                "double"                "double"             "character"
```

We have to convert the dates from string to a date object

```
d['OCCUR_DATE2'] <- as.Date(d$OCCUR_DATE,"%m/%d/%Y")
head(d$OCCUR_DATE2)
```

```
## [1] "2021-05-27" "2014-06-27" "2015-11-21" "2015-10-09" "2009-02-19"
## [6] "2020-10-21"
```

We see that most of our variables are categorical, so it is better to expore by using frequenices. We can ignore the coordinates as we won't conduct a geostatistical analysis.

```
pct_na<-sapply(d,function(x){sum(is.na(x))/length(x)*100})
names <-names(d)

df<-data.frame(names,pct_na)
df
```

```
##                                       names      pct_na
## INCIDENT_KEY                     INCIDENT_KEY 0.000000000
## OCCUR_DATE                       OCCUR_DATE 0.000000000
## OCCUR_TIME                       OCCUR_TIME 0.000000000
## BORO                                   BORO 0.000000000
## LOC_OF_OCCUR_DESC           LOC_OF_OCCUR_DESC 0.000000000
## PRECINCT                             PRECINCT 0.000000000
## JURISDICTION_CODE           JURISDICTION_CODE 0.007322789
## LOC_CLASSFCTN_DESC         LOC_CLASSFCTN_DESC 0.000000000
## LOCATION_DESC                 LOCATION_DESC 0.000000000
## STATISTICAL_MURDER_FLAG STATISTICAL_MURDER_FLAG 0.000000000
## PERP_AGE_GROUP                PERP_AGE_GROUP 0.000000000
## PERP_SEX                            PERP_SEX 0.000000000
## PERP_RACE                          PERP_RACE 0.000000000
## VIC_AGE_GROUP                  VIC_AGE_GROUP 0.000000000
## VIC_SEX                              VIC_SEX 0.000000000
## VIC_RACE                            VIC_RACE 0.000000000
## X_COORD_CD                        X_COORD_CD 0.000000000
## Y_COORD_CD                        Y_COORD_CD 0.000000000
## Latitude                            Latitude 0.036613943
## Longitude                          Longitude 0.036613943
## Lon_Lat                              Lon_Lat 0.000000000
## OCCUR_DATE2                      OCCUR_DATE2 0.000000000
```

For the Categorical let's make a few frequencies

```
##  [1] "OCCUR_DATE"              "OCCUR_TIME"
##  [3] "BORO"                    "LOC_OF_OCCUR_DESC"
##  [5] "LOC_CLASSFCTN_DESC"      "LOCATION_DESC"
##  [7] "STATISTICAL_MURDER_FLAG" "PERP_AGE_GROUP"
##  [9] "PERP_SEX"                "PERP_RACE"
## [11] "VIC_AGE_GROUP"           "VIC_SEX"
## [13] "VIC_RACE"                "Lon_Lat"
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## i Please use `as_tibble()` instead.
## i The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## # A tibble: 5 x 2
##   x                 n
##   <chr>         <dbl>
## 1 BRONX         0.291
## 2 BROOKLYN      0.400
```

```
## 3 MANHATTAN     0.131
## 4 QUEENS        0.150
## 5 STATEN ISLAND 0.0284
## # A tibble: 3 x 2
##   x              n
##   <chr>      <dbl>
## 1 ""        0.937
## 2 "INSIDE"  0.00886
## 3 "OUTSIDE" 0.0540
## # A tibble: 10 x 2
##    x                   n
##    <chr>           <dbl>
##  1 ""            0.937
##  2 "COMMERCIAL"  0.00366
##  3 "DWELLING"    0.00465
##  4 "HOUSING"     0.0103
##  5 "OTHER"       0.00114
##  6 "PARKING LOT" 0.000256
##  7 "PLAYGROUND"  0.00110
##  8 "STREET"      0.0404
##  9 "TRANSIT"     0.000549
## 10 "VEHICLE"     0.000842
## # A tibble: 41 x 2
##    x                          n
##    <chr>                  <dbl>
##  1 ""                   0.548
##  2 "(null)"             0.0358
##  3 "ATM"                0.0000366
##  4 "BANK"               0.000110
##  5 "BAR/NIGHT CLUB"     0.0230
##  6 "BEAUTY/NAIL SALON" 0.00410
##  7 "CANDY STORE"        0.000256
##  8 "CHAIN STORE"        0.000183
##  9 "CHECK CASH"         0.0000366
## 10 "CLOTHING BOUTIQUE" 0.000513
## # i 31 more rows
## # A tibble: 2 x 2
##   x         n
##   <chr> <dbl>
## 1 false 0.807
## 2 true  0.193
## # A tibble: 11 x 2
##    x                n
##    <chr>        <dbl>
##  1 ""        0.342
##  2 "(null)"  0.0234
##  3 "<18"     0.0583
##  4 "1020"    0.0000366
##  5 "18-24"   0.228
##  6 "224"     0.0000366
##  7 "25-44"   0.208
##  8 "45-64"   0.0226
##  9 "65+"     0.00220
## 10 "940"     0.0000366
```

```
## 11 "UNKNOWN" 0.115
## # A tibble: 5 x 2
##   x             n
##   <chr>     <dbl>
## 1 ""       0.341
## 2 "(null)" 0.0234
## 3 "F"      0.0155
## 4 "M"      0.565
## 5 "U"      0.0549
## # A tibble: 9 x 2
##   x                                    n
##   <chr>                            <dbl>
## 1 ""                             0.341
## 2 "(null)"                       0.0234
## 3 "AMERICAN INDIAN/ALASKAN NATIVE" 0.0000732
## 4 "ASIAN / PACIFIC ISLANDER"       0.00564
## 5 "BLACK"                        0.419
## 6 "BLACK HISPANIC"               0.0481
## 7 "UNKNOWN"                      0.0672
## 8 "WHITE"                        0.0104
## 9 "WHITE HISPANIC"               0.0857
## # A tibble: 7 x 2
##   x              n
##   <chr>      <dbl>
## 1 <18     0.104
## 2 1022    0.0000366
## 3 18-24   0.369
## 4 25-44   0.450
## 5 45-64   0.0682
## 6 65+     0.00663
## 7 UNKNOWN 0.00223
## # A tibble: 3 x 2
##   x           n
##   <chr>   <dbl>
## 1 F     0.0957
## 2 M     0.904
## 3 U     0.000403
## # A tibble: 7 x 2
##   x                                   n
##   <chr>                           <dbl>
## 1 AMERICAN INDIAN/ALASKAN NATIVE 0.000366
## 2 ASIAN / PACIFIC ISLANDER       0.0148
## 3 BLACK                          0.712
## 4 BLACK HISPANIC                 0.0969
## 5 UNKNOWN                        0.00242
## 6 WHITE                          0.0256
## 7 WHITE HISPANIC                 0.148

##   BORO        LOC_OF_OCCUR_DESC LOC_CLASSFCTN_DESC LOCATION_DESC
## x character,5 character,3       character,10       character,41
## n numeric,5   numeric,3         numeric,10         numeric,41
##   STATISTICAL_MURDER_FLAG PERP_AGE_GROUP PERP_SEX    PERP_RACE   VIC_AGE_GROUP
## x character,2             character,11   character,5 character,9 character,7
## n numeric,2               numeric,11     numeric,5   numeric,9   numeric,7
```

```
##   VIC_SEX     VIC_RACE
## x character,3 character,7
## n numeric,3   numeric,7
```

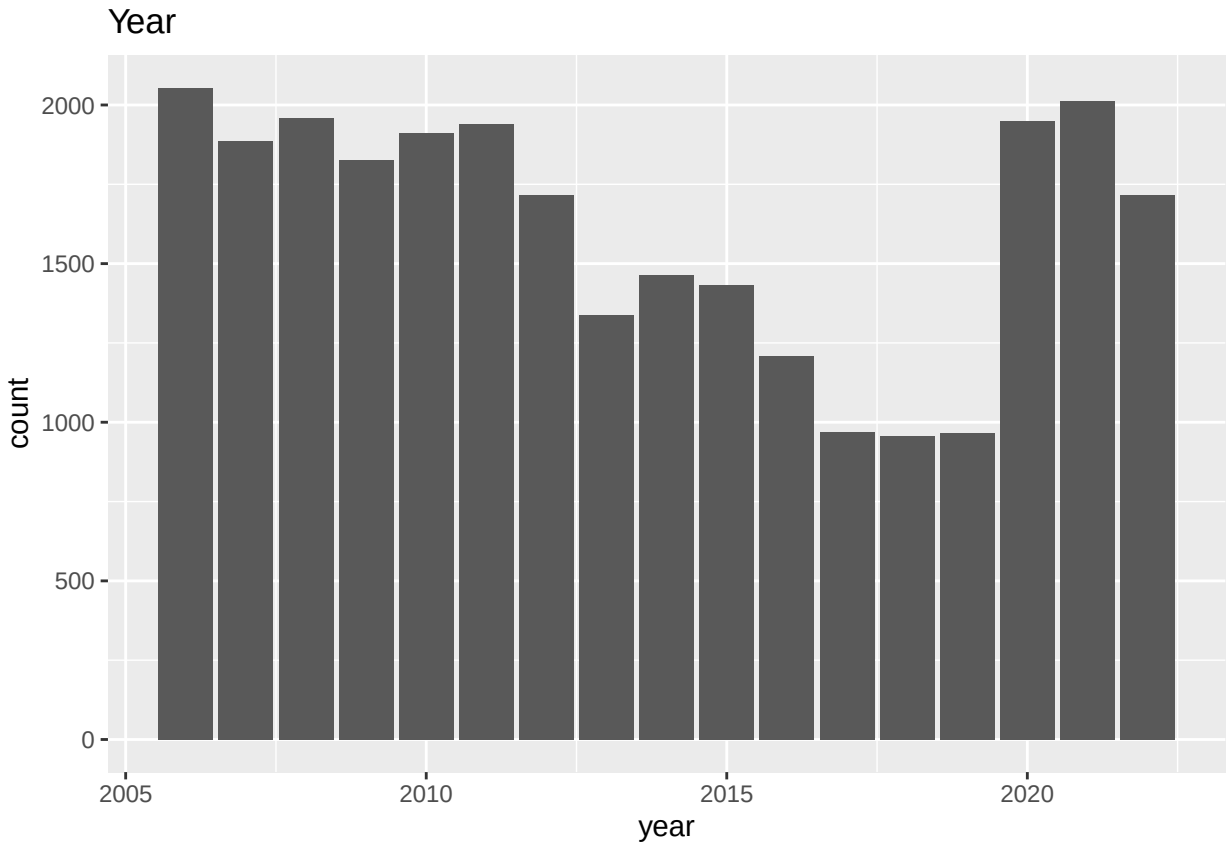
1.- We see that LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC have 93% missing so we can't use those columns. 2.- Sex of the perpetrator is empty for 36%, but it is safe to impute M 3.- Sex of the victim has no missing values and 90% is male.

So we see an obvious pattern, that males are way overrepresented as victims ad pepetrators in this type of violent crime. Which matches our intuition

Graphical Presentation of frequencies

Lets se the Frequencies Graphically

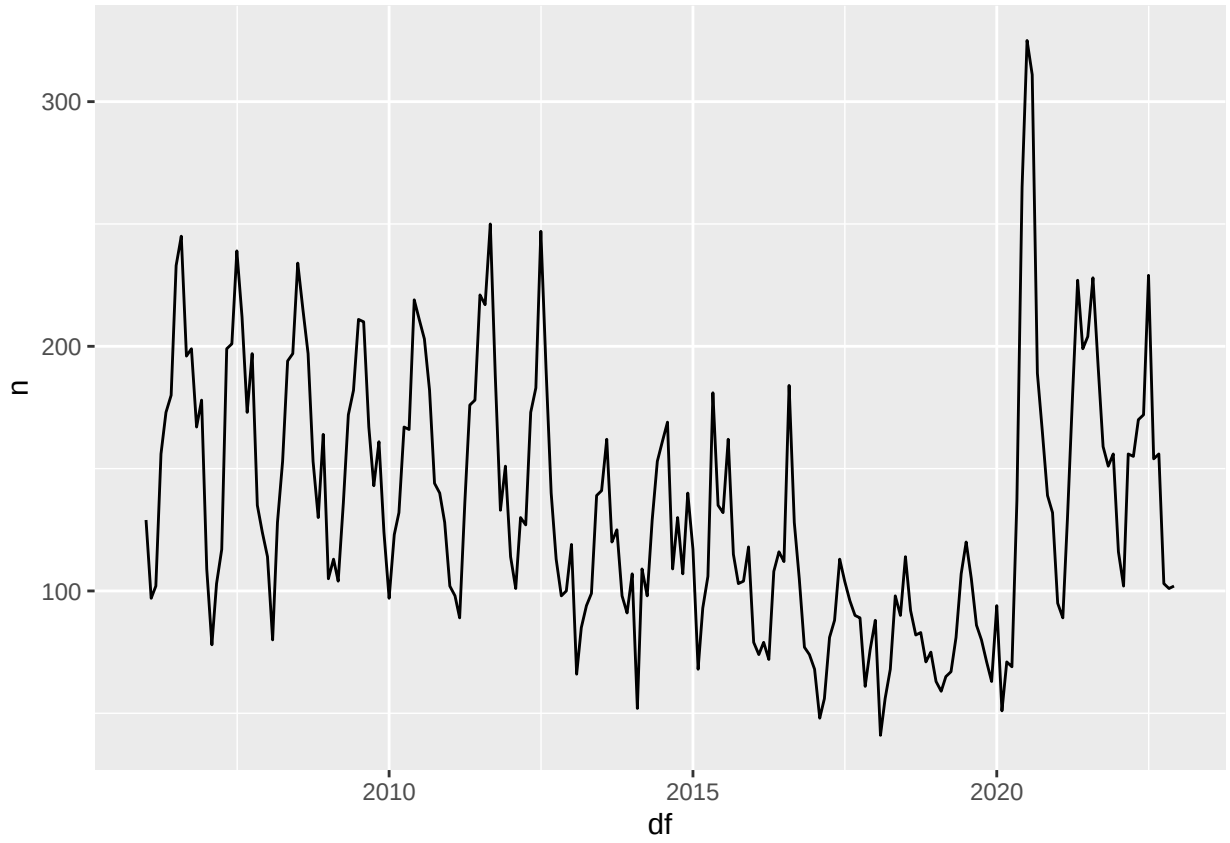
```
library(ggplot2)
ggplot(data=data)+geom_bar(aes(x=BORO))+ggtitle("Borough")
ggplot(data=data)+geom_bar(aes(x=PERP_AGE_GROUP))+ggtitle("Perpetrator Age Group")

ggplot(data=data)+geom_bar(aes(x=PERP_RACE))+ggtitle("Perpetrator Race")+coord_flip()

ggplot(data=data)+geom_bar(aes(x=VIC_RACE))+ggtitle("Victim Race")+coord_flip()
```



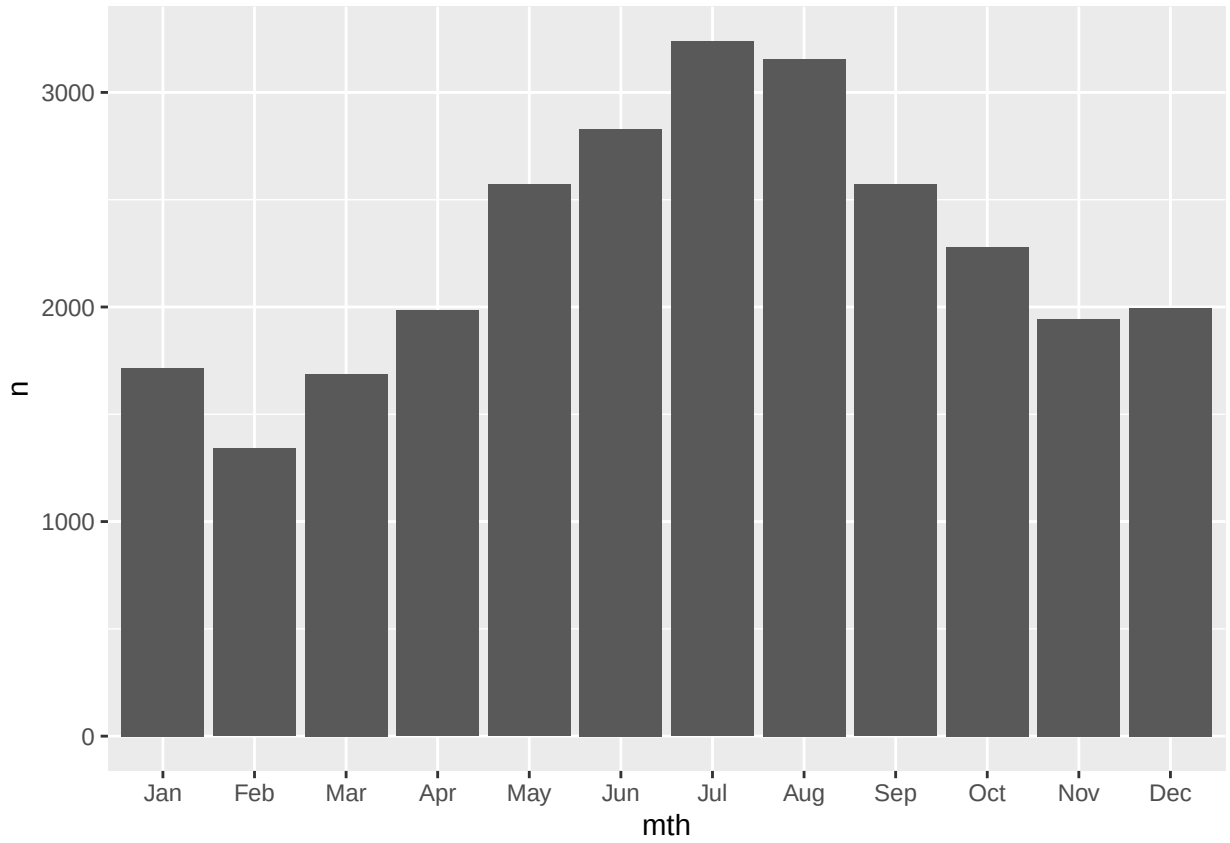
Seasonality

Lets group the events by month and see if there are any patterns

```
## # A tibble: 6 x 2
##   df             n
##   <date>     <int>
## 1 2006-01-01   129
## 2 2006-02-01    97
## 3 2006-03-01   102
## 4 2006-04-01   156
## 5 2006-05-01   173
## 6 2006-06-01   180
```




```
## # A tibble: 6 x 2
##   mth       n
##   <ord> <int>
## 1 Jan    1716
## 2 Feb    1340
## 3 Mar    1688
## 4 Apr    1983
## 5 May    2571
## 6 Jun    2829
```



And we see a spike in Summer.

We see a clear Seasonality Patter.

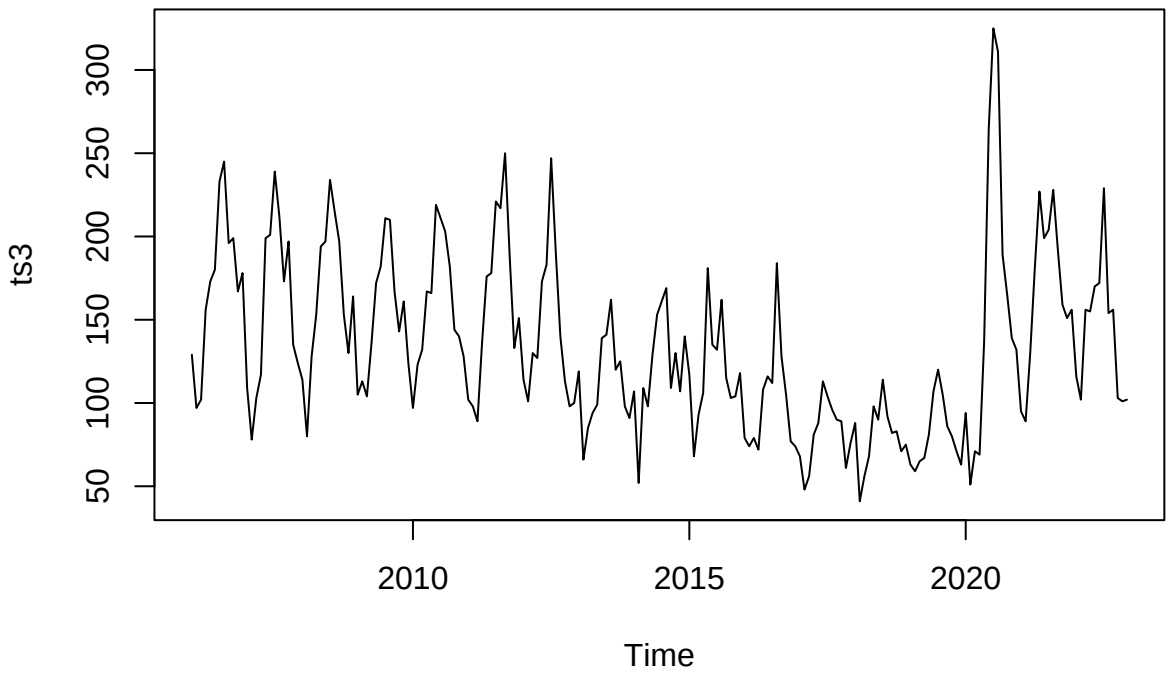
Test Seasonality

We will try to figure out the seasonality

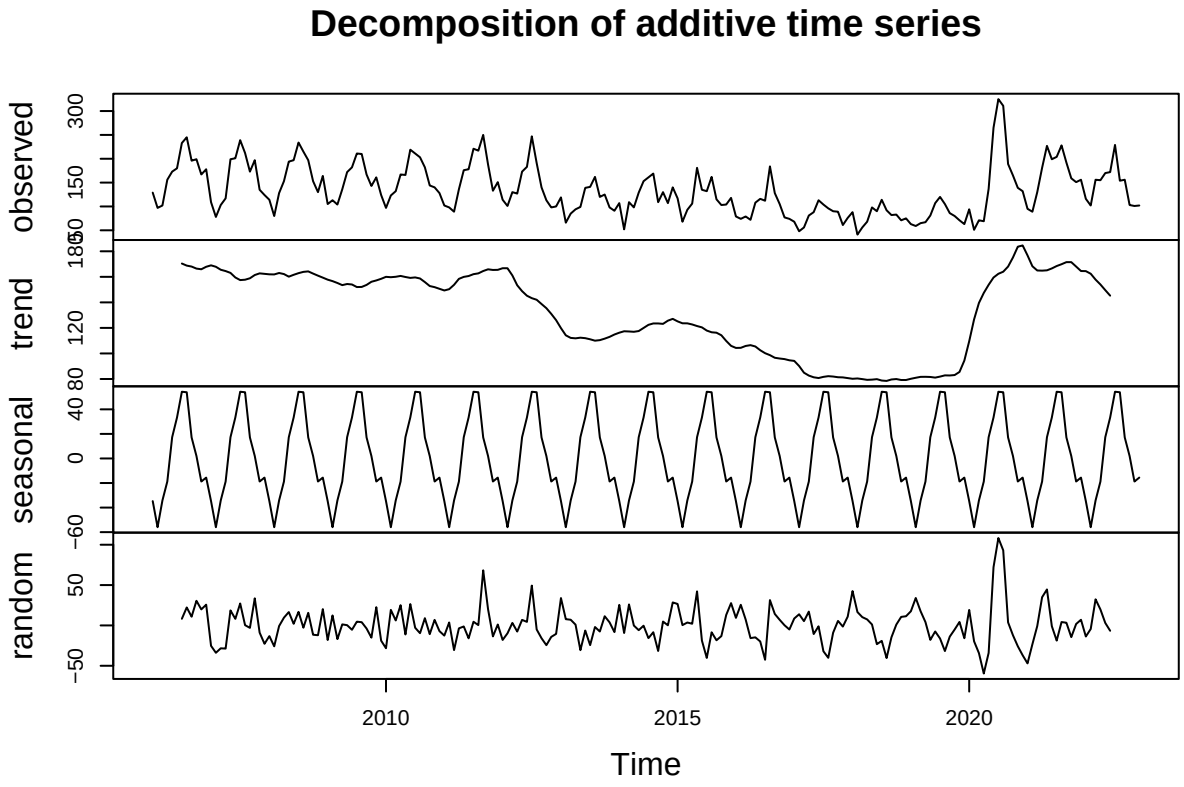
```
ts2<- d %>%
  group_by(df ) %>%
  summarise( n = n())
head(ts2)
```

```
## # A tibble: 6 x 2
##   df             n
##   <date>     <int>
## 1 2006-01-01   129
## 2 2006-02-01    97
## 3 2006-03-01   102
## 4 2006-04-01   156
## 5 2006-05-01   173
## 6 2006-06-01   180
```

```
ts3<-ts(ts2$n,frequency = 12,start=c(2006,1))
plot(ts3)
```




```
ts_components <- decompose(ts3)
plot(ts_components)
```




```
summary(ts_components)
```

```
##          Length Class  Mode
## x        204    ts     numeric
## seasonal 204    ts     numeric
## trend    204    ts     numeric
## random   204    ts     numeric
## figure    12    -none- numeric
## type       1    -none- character
```

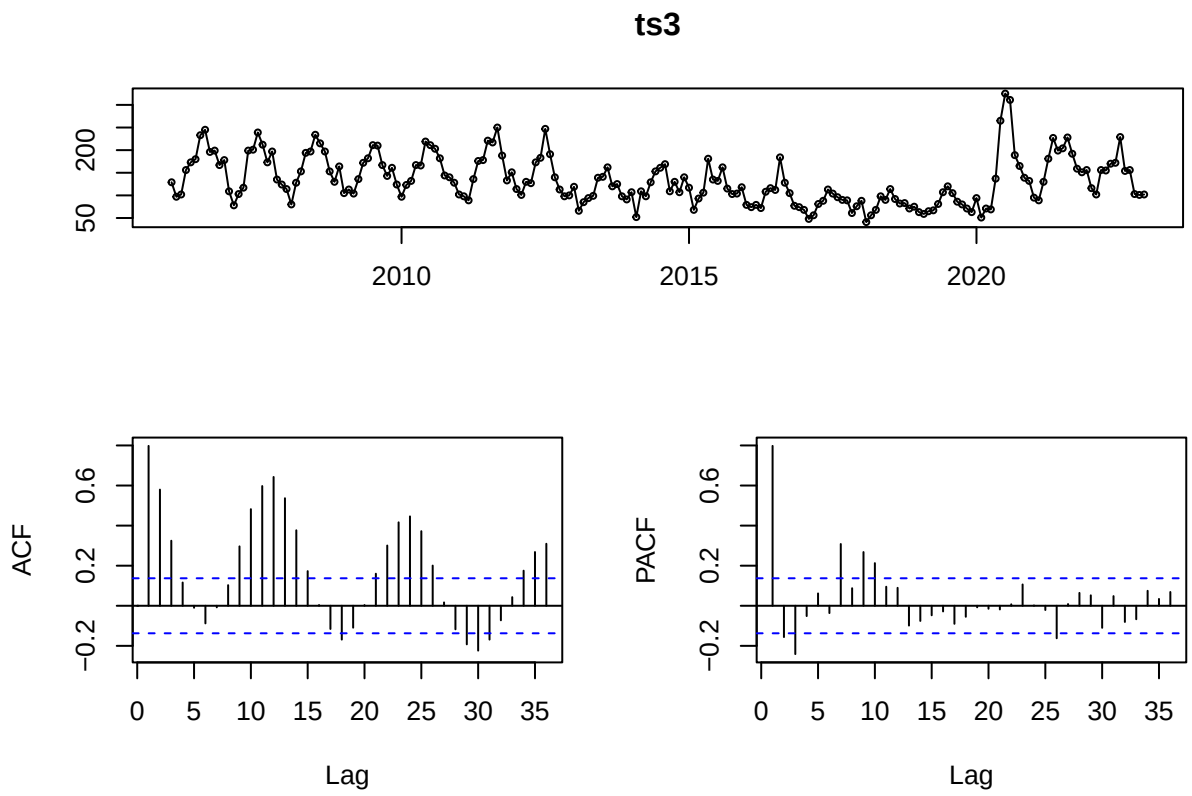
Notice in trend the the downward slope and the structural breakdown due to COVID which caused asharp increase and brought us back to pre 2010 levels. We see a strong seasonality component

Below, we fit an arima model and we can see that the seasonal component is h

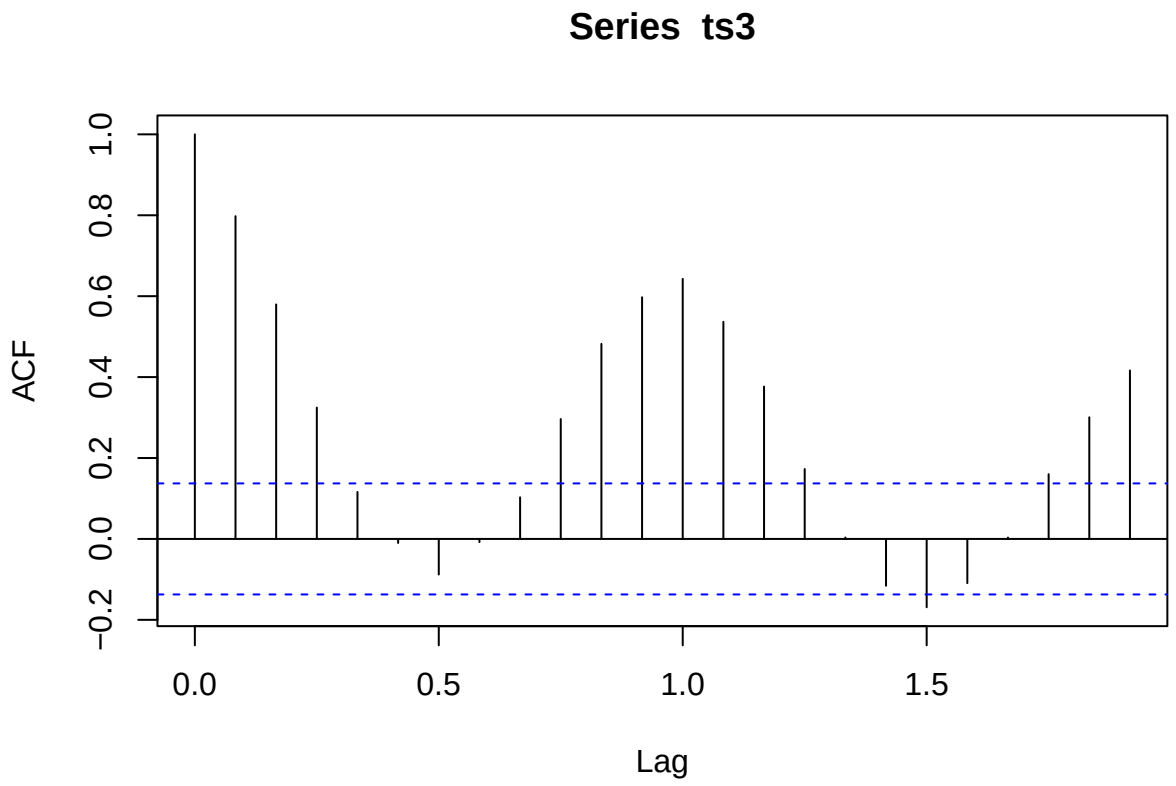
```
###

library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

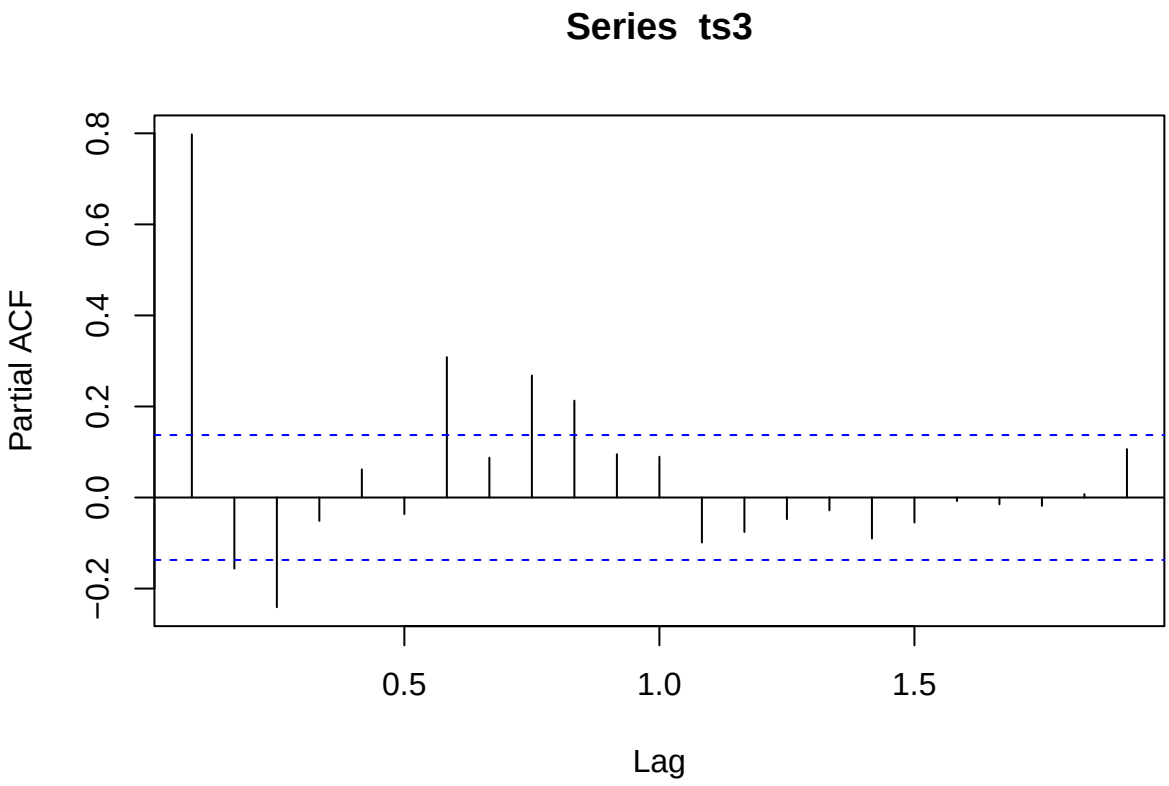
```
tsdisplay(ts3)
```



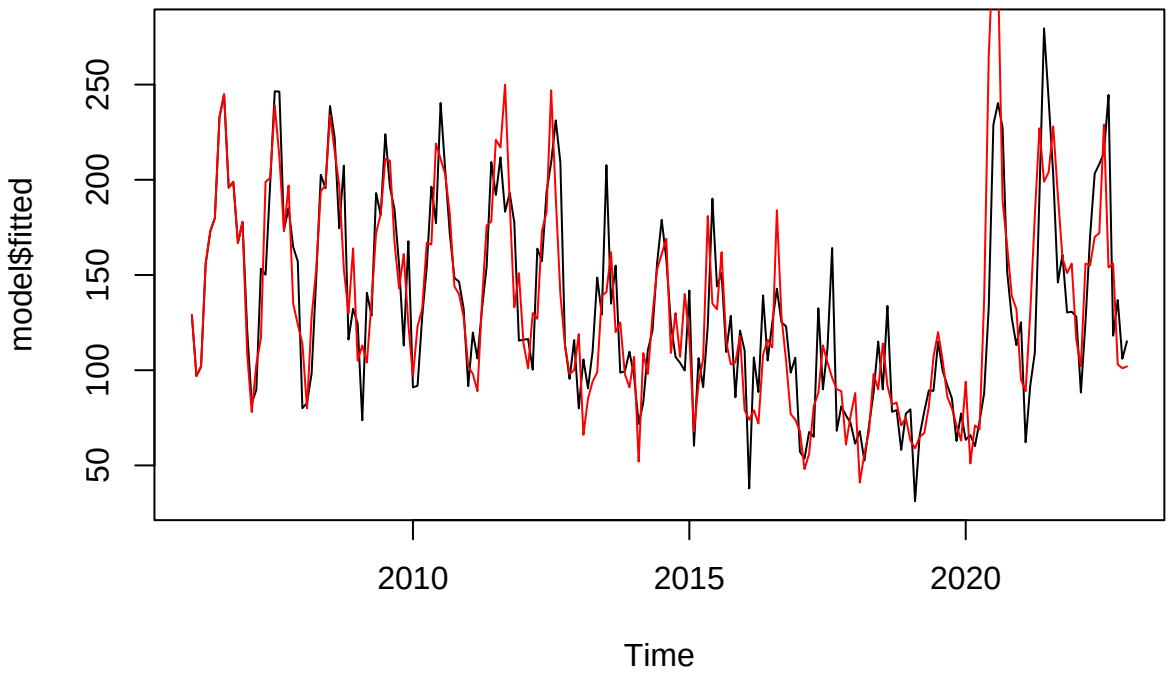

```
plot(acf(ts3))
```



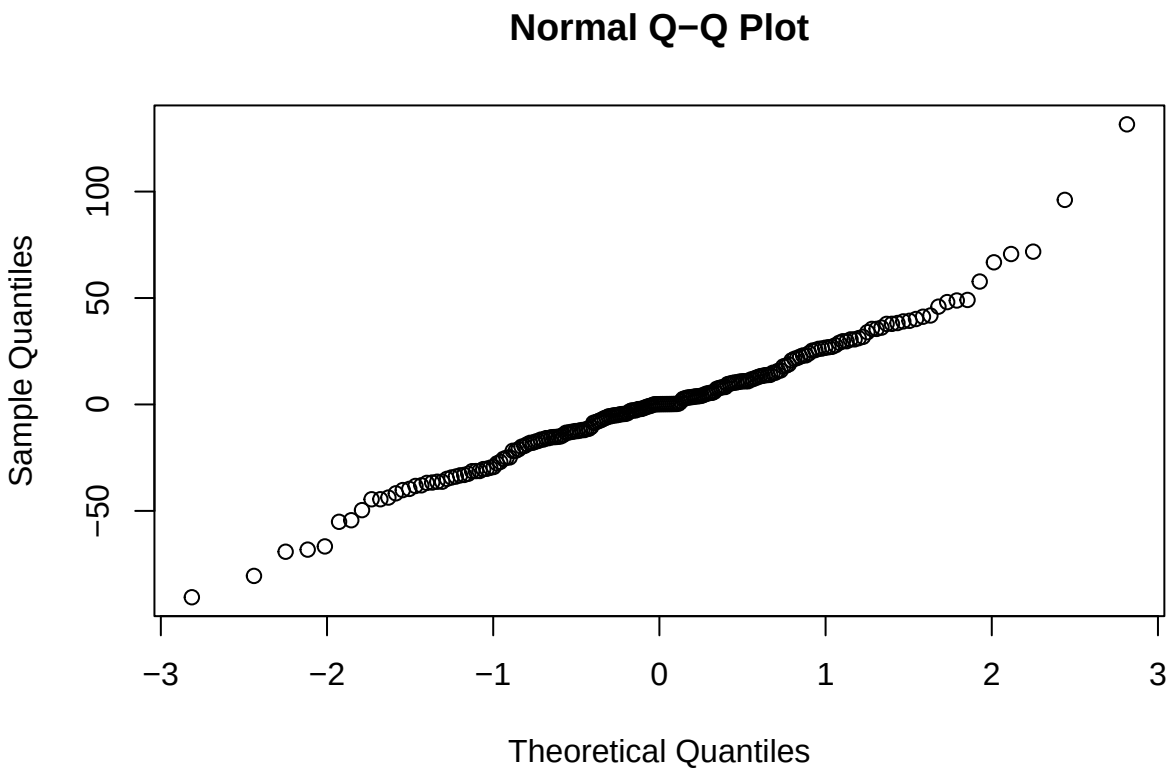

```
plot(pacf(ts3))
```




```
model<-auto.arima(ts3,seasonal = TRUE)
plot(model$fitted)
lines(ts3, col='red')
```




```
qqnorm(model$residuals)
```




```
summary(model)
```

```
## Series: ts3
## ARIMA(1,0,0)(1,1,0)[12] with drift
##
## Coefficients:
##          ar1     sar1    drift
##       0.6803  -0.3835  -0.1351
## s.e.  0.0532   0.0689   0.4114
##
## sigma^2 = 917.1:  log likelihood = -927.02
## AIC=1862.04   AICc=1862.25   BIC=1875.07
##
## Training set error measures:
##                       ME     RMSE     MAE       MPE     MAPE     MASE
## Training set -0.02437165 29.14876 21.3069 -2.240472 16.99186 0.767672
##                      ACF1
## Training set -0.08032225
```