NYPD Shooting Incident Data

cloud-erik

20 5 2021

Question of interest

After I saw the date first time my primary interest was to see if there is a correlation between number of shootings and the location. So to make clusters visible if there are some. During analyses a second so far unknown question came up. That was if there is any correlation to the season or time of day.

I will address during this analyses both of theses questions.

Librarys

Load used library

```
library(lubridate)
```

```
##
## Attaching package: ’lubridate’

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.5     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(ggmap)
```

```
## Google’s Terms of Service: https://cloud.google.com/maps-platform/terms/.
```

```
## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```
library(ggplot2)
library(timetk)
```

Data Import

First importing the “NYPD Shooting Incident Data (Historic)” Dataset as CSV file from https://catalog.data.gov and create a summary.

```
url_source <- "https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD"
shootings <- read.csv(url_source)
summary(shootings)
```

```
##   INCIDENT_KEY        OCCUR_DATE         OCCUR_TIME            BORO
##  Min.   :  9953245   Length:23568       Length:23568       Length:23568
##  1st Qu.: 55317014   Class :character   Class :character   Class :character
##  Median : 83365370   Mode  :character   Mode  :character   Mode  :character
##  Mean   :102218616
##  3rd Qu.:150772442
##  Max.   :222473262
##
##     PRECINCT      JURISDICTION_CODE LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Min.   :0.0000    Length:23568       Length:23568
##  1st Qu.: 44.00   1st Qu.:0.0000    Class :character   Class :character
##  Median : 69.00   Median :0.0000    Mode  :character   Mode  :character
##  Mean   : 66.21   Mean   :0.3323
##  3rd Qu.: 81.00   3rd Qu.:0.0000
##  Max.   :123.00   Max.   :2.0000
##                   NA’s   :2
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  Length:23568       Length:23568       Length:23568       Length:23568
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_SEX            VIC_RACE          X_COORD_CD         Y_COORD_CD
##  Length:23568       Length:23568       Length:23568       Length:23568
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
##     Latitude       Longitude        Lon_Lat
##  Min.   :40.51   Min.   :-74.25   Length:23568
##  1st Qu.:40.67   1st Qu.:-73.94   Class :character
##  Median :40.70   Median :-73.92   Mode  :character
##  Mean   :40.74   Mean   :-73.91
##  3rd Qu.:40.82   3rd Qu.:-73.88
##  Max.   :40.91   Max.   :-73.70
##
```

cleaning up the data

- convert date and time column to one datetime
- drop not needed columns X_COORD_CD, Y_COORD_CD, Lon_Lat, PRECINCT, JURISDICTION_CODE, LOCATION_DESC, BORO and keep only one location (Latitude, Longitude)

After cleaning the data create again a summary

```
shootings <- shootings %>%
  unite(OCCUR_DATE, OCCUR_DATE:OCCUR_TIME) %>%
  mutate(OCCUR_DATE = mdy_hms(OCCUR_DATE)) %>%
  select(-c(INCIDENT_KEY, BORO, X_COORD_CD, Y_COORD_CD, Lon_Lat, PRECINCT,
    JURISDICTION_CODE, LOCATION_DESC)) %>%
  rename(Murder = "STATISTICAL_MURDER_FLAG")
summary(shootings)
```

```
##    OCCUR_DATE                         Murder          PERP_AGE_GROUP
##  Min.   :2006-01-01 02:00:00   Length:23568       Length:23568
##  1st Qu.:2008-12-30 04:27:00   Class :character   Class :character
##  Median :2012-02-26 03:35:00   Mode  :character   Mode  :character
##  Mean   :2012-10-04 05:23:12
##  3rd Qu.:2016-02-28 00:01:00
##  Max.   :2020-12-31 23:45:00
##    PERP_SEX          PERP_RACE         VIC_AGE_GROUP        VIC_SEX
##  Length:23568       Length:23568       Length:23568       Length:23568
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    VIC_RACE            Latitude       Longitude
##  Length:23568       Min.   :40.51   Min.   :-74.25
##  Class :character   1st Qu.:40.67   1st Qu.:-73.94
##  Mode  :character   Median :40.70   Median :-73.92
##                     Mean   :40.74   Mean   :-73.91
##                     3rd Qu.:40.82   3rd Qu.:-73.88
##                     Max.   :40.91   Max.   :-73.70
```

Visulize

Because the shootings are spatial data I decided to do first a visualization on a map to get a better overview if there are any clusters.

first sort the data frame with murder at the tail to make sure that murder are not overprinted.

```r
shootings <- shootings[order(shootings$Murder),]
```

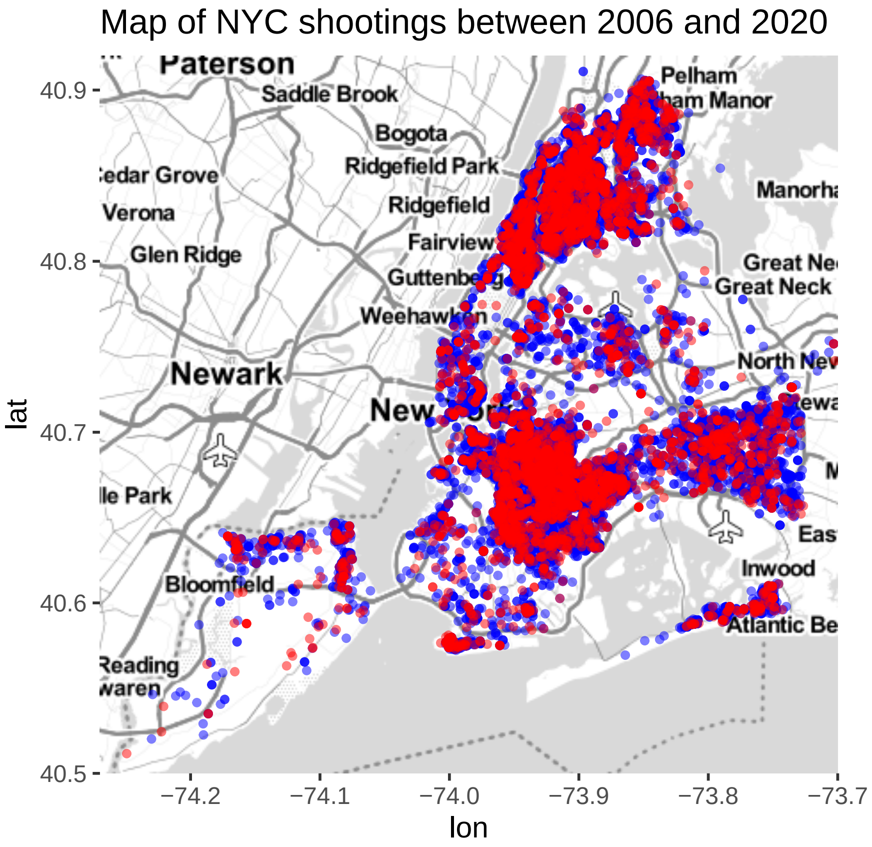
Map with shootings without murder in blue and with murder in red

First I plot all shootings on a stamen map of New York, shootings without murder in blue and with murder in red.

```r
myLocation<-c(-74.27, 40.5, -73.7, 40.92) # New York
myMap <- get_stamenmap(bbox=myLocation, maptype="toner-lite", crop=TRUE)
```

```
## Source : http://tile.stamen.com/toner-lite/10/300/384.png

## Source : http://tile.stamen.com/toner-lite/10/301/384.png

## Source : http://tile.stamen.com/toner-lite/10/302/384.png

## Source : http://tile.stamen.com/toner-lite/10/300/385.png

## Source : http://tile.stamen.com/toner-lite/10/301/385.png

## Source : http://tile.stamen.com/toner-lite/10/302/385.png
```

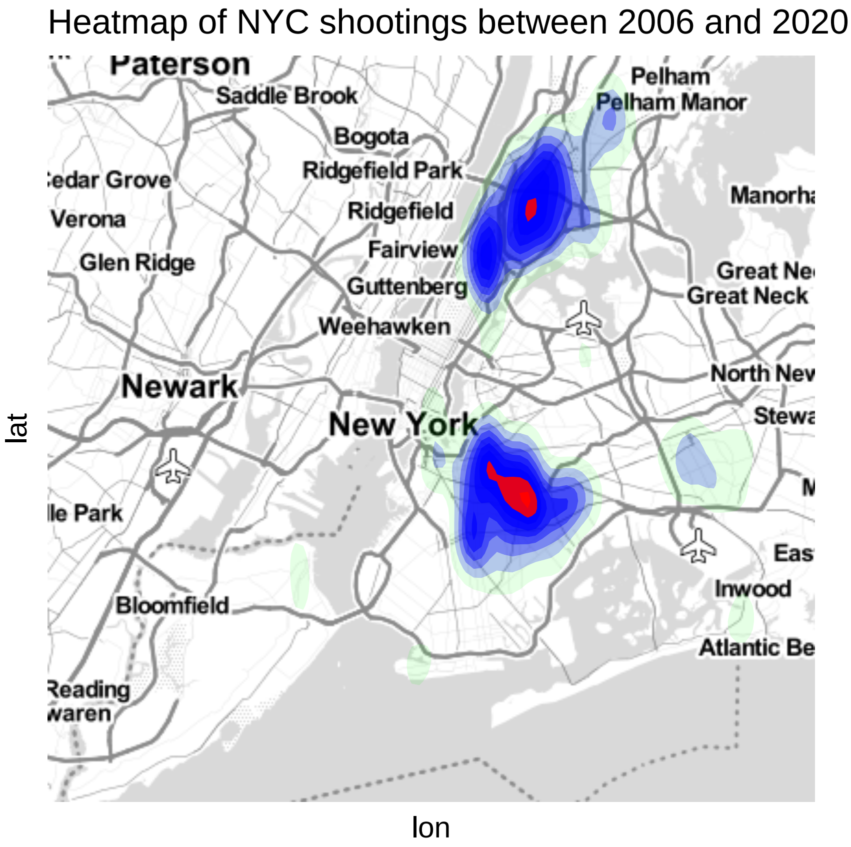
```r
ggmap(myMap)+
geom_point(aes(x=Longitude, y=Latitude), data=shootings, alpha=0.5,
           color=ifelse(shootings$Murder=="true", "red", "blue"), size = 1) +
ggtitle("Map of NYC shootings between 2006 and 2020")
```



Heatmap wihh shootings without murder in blue and with murder in red

To make cluster more visual I visualize same data again with a heat-map starting from green up to shootings without murder in blue or with murder in red.

```
ggmap(myMap)+
stat_density2d(aes(x=Longitude, y=Latitude,fill=..level.., alpha=..level..),
               data=shootings, geom="polygon")+
scale_fill_gradient(low = "green", high = ifelse(shootings$Murder=="true", "red", "blue"))+
theme(axis.ticks = element_blank(),
axis.text = element_blank(),
legend.position="none") +
ggtitle("Heatmap of NYC shootings between 2006 and 2020")
```

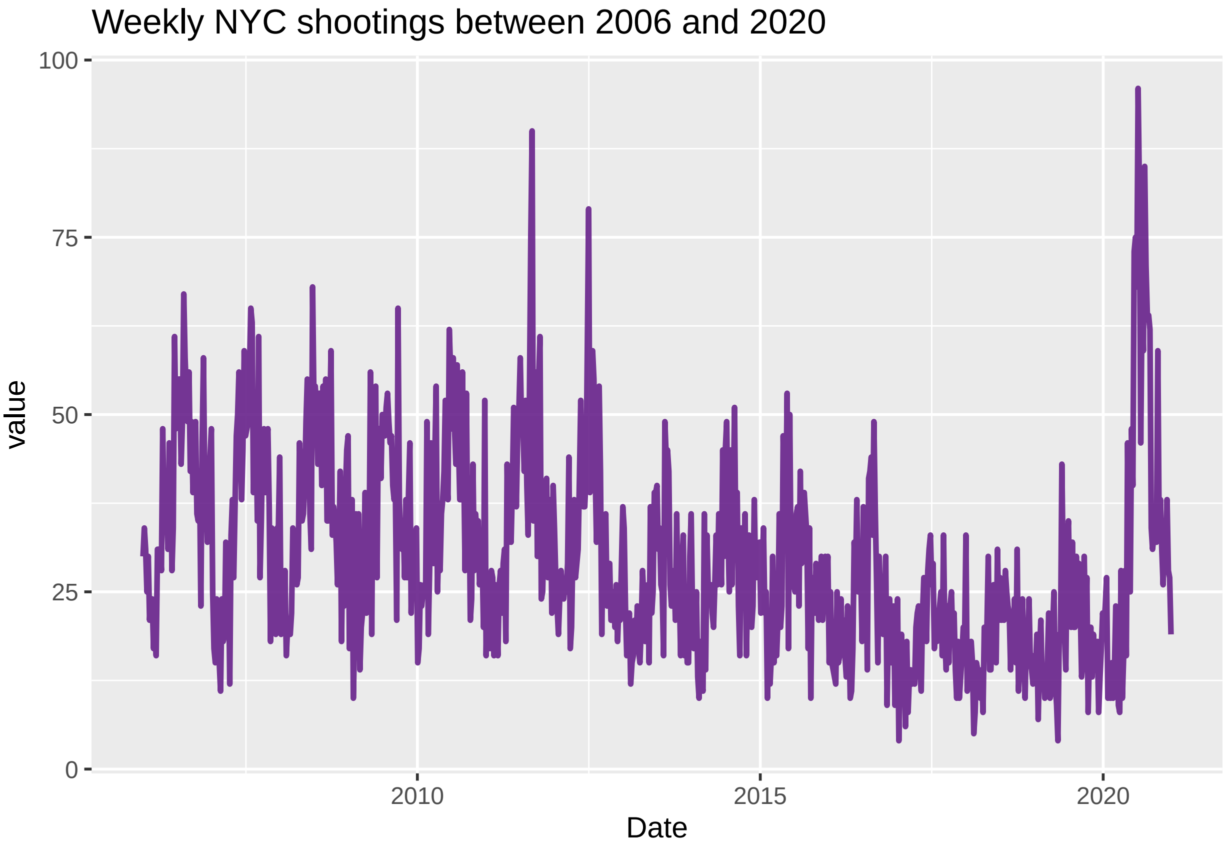


So it look that there happens significant more shootings and murder in the area of Brooklyn and Bronx than in the other areas.

Timeline

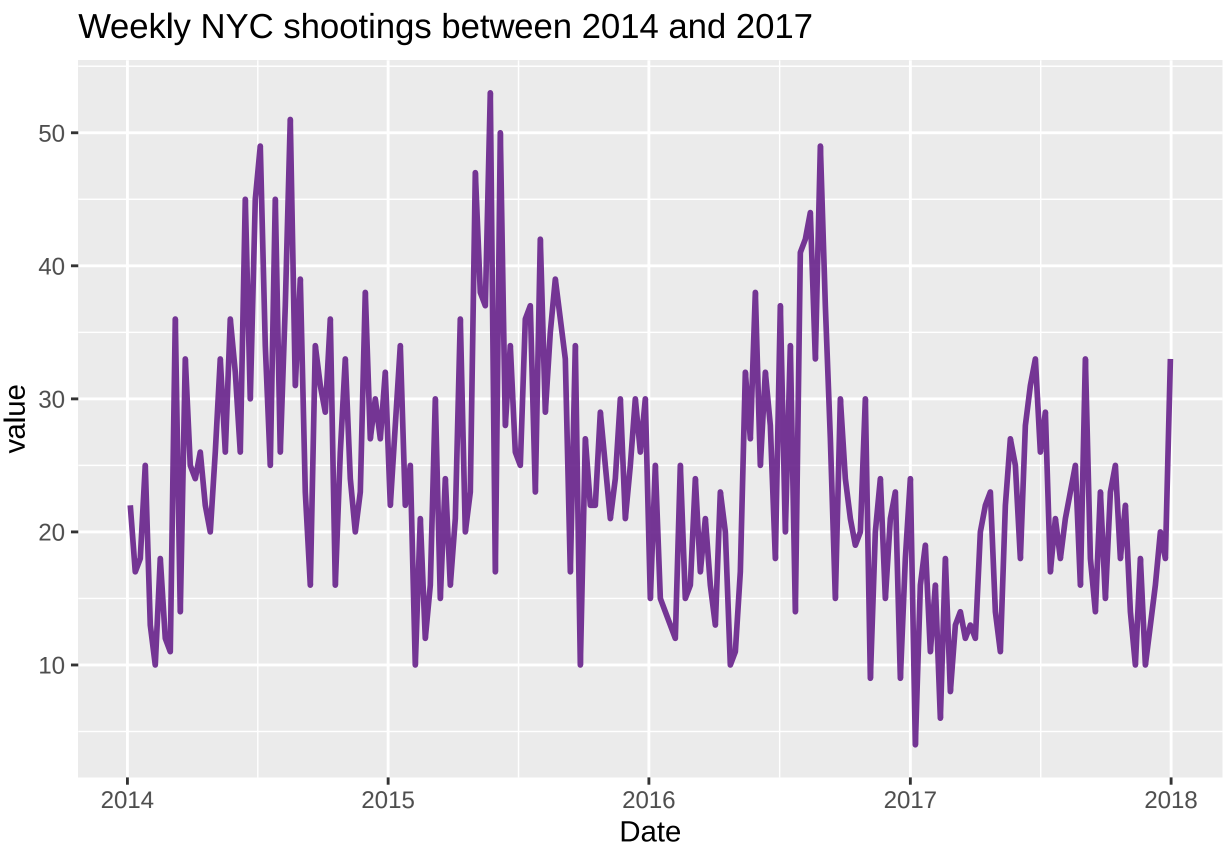
To see if there is any trend over time I plotted data on a time-line aggregated by week.

```r
shootings_byweek <- summarise_by_time(shootings, .date_var = OCCUR_DATE,
                                      .by = "week", value = n())
p <- ggplot(shootings_byweek, aes(x = OCCUR_DATE, y = value)) +
geom_line(color="darkorchid4", size=1, alpha=0.9, linetype=1) +
#geom_point(color="darkorchid4") +
labs(x="Date") +
ggtitle("Weekly NYC shootings between 2006 and 2020")
p
```



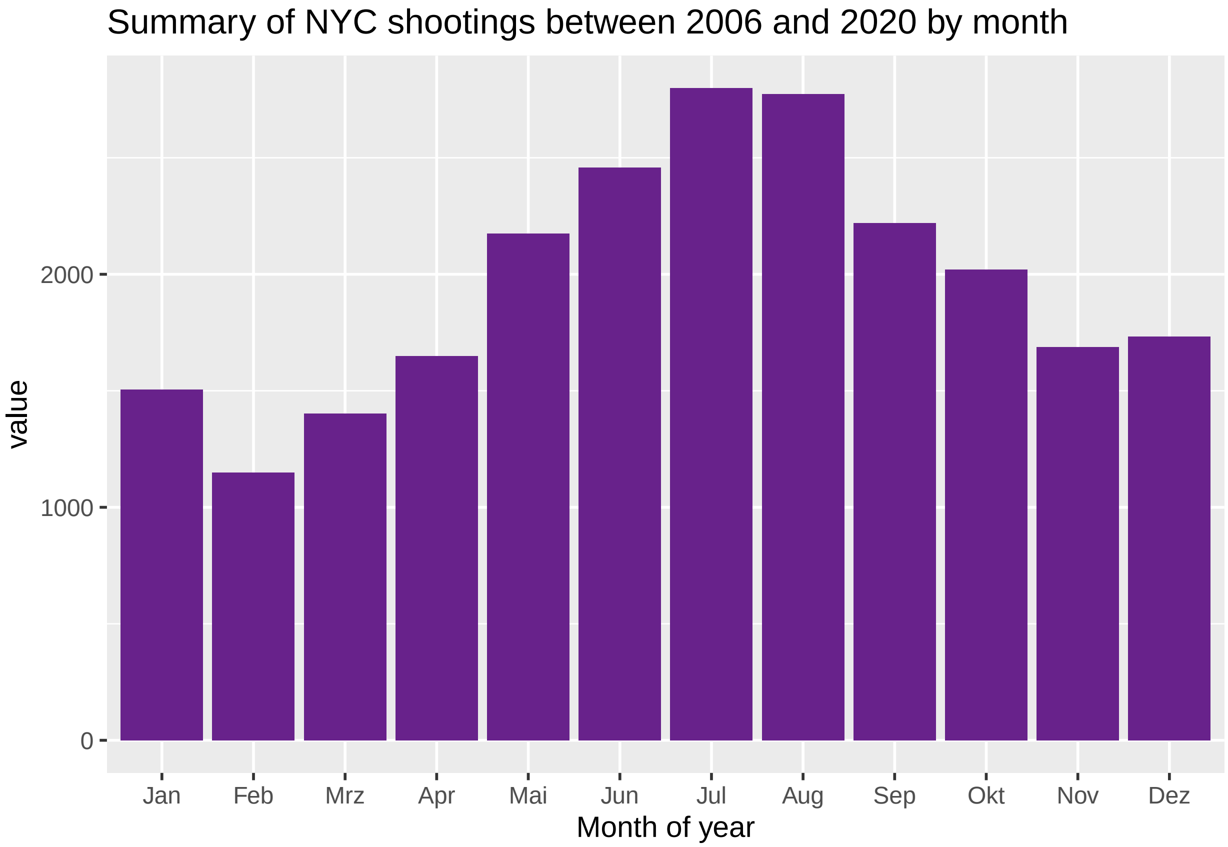
It looks like these are some seasonal variances over the time and there is a peak of shootings in 2020. Take only Data from one four Years to show the seasonal variance.

```
shootings_byweek2015 <- filter_by_time(shootings_byweek, .date_var = OCCUR_DATE,
                                      .start_date = "2014-01-01", .end_date = "2017-12-31")
p <- ggplot(shootings_byweek2015, aes(x = OCCUR_DATE, label=TRUE, y = value)) +
geom_line(color="darkorchid4", size=1, alpha=0.9, linetype=1) +
#geom_point() +
labs(x="Date") +
ggtitle("Weekly NYC shootings between 2014 and 2017")
p
```



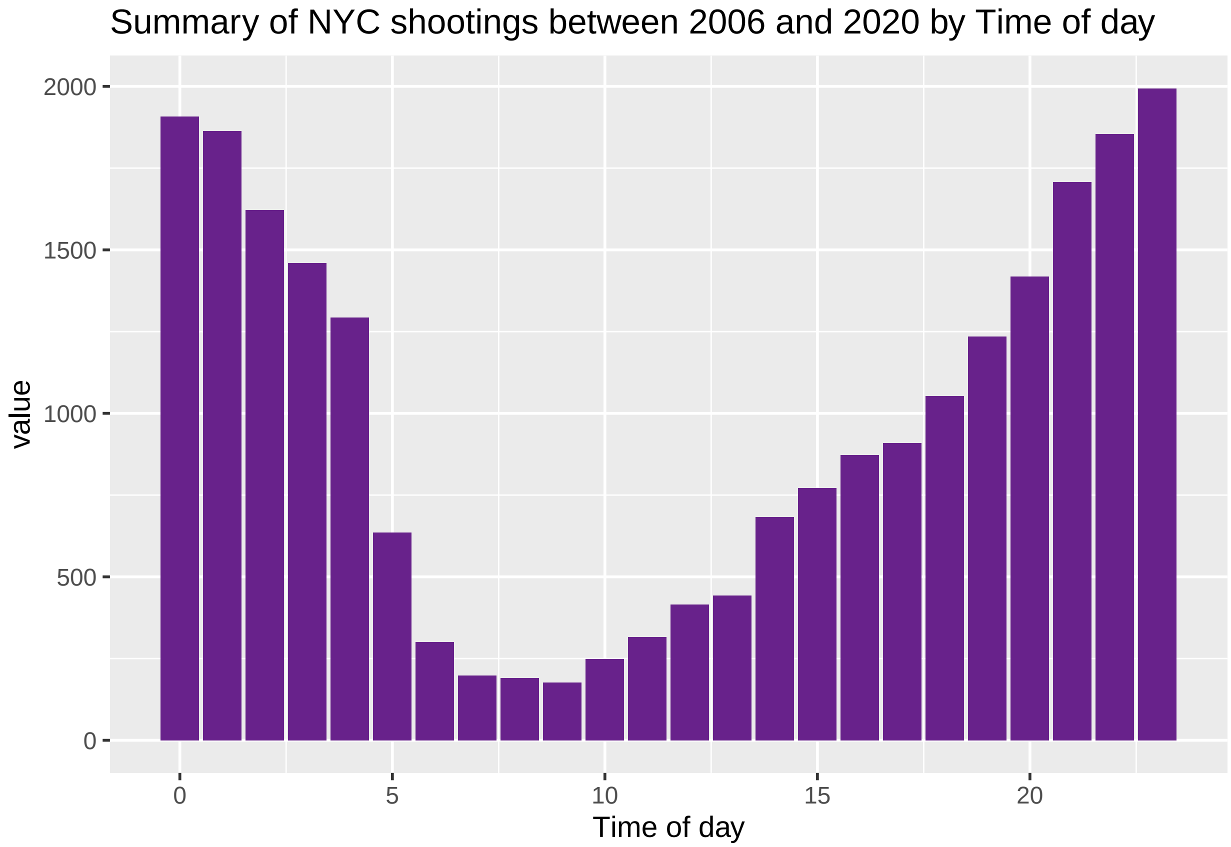
So it look that there happens more shootings during summer and lesser during winter time. To verify this lets make a histogram of shootings by month.

```
shootings_bymonth <- summarise_by_time(shootings, .date_var = OCCUR_DATE,
                                       .by = "month", value = n())
p <- ggplot(shootings_bymonth, aes(x = month(OCCUR_DATE, label=TRUE), y = value)) +
geom_bar(stat = "identity", fill = "darkorchid4") +
labs(x="Month of year") +
ggtitle("Summary of NYC shootings between 2006 and 2020 by month")
p
```



Finally have a look at the time shootings happen. So create a histogram by hour of the day.

```
shootings_byhour <- summarise_by_time(shootings, .date_var = OCCUR_DATE,
                                      .by = "hour", value = n())
p <- ggplot(shootings_byhour, aes(x = hour(OCCUR_DATE), y = value)) +
geom_bar(stat = "identity", fill = "darkorchid4") +
labs(x="Time of day") +
ggtitle("Summary of NYC shootings between 2006 and 2020 by Time of day")
p
```



Bias

Because shootings are surpassing events I expect that there is no great bias like in other criminal statistics with minor crime that could be e.g. somehow correlated with the presence of police. But I assume that shootings will be recognized always with or without police and so all shootings should be part of the official statistics. Also the correlation between murder and shootings without murder shows that there seems minor bias in the data.

The spatial data should take into account the density of population in more detailed analysis.

A major bias is of course witch additional information is included within the data. So simply because e.g. race is included in the data it implies that there could be a correlation. That's in my opinion dangerous because it could lead to wrong and maybe discriminating results if its not cleaned and leveled carefully.

Conclusion

Concerning the correlation between number of shootings and the location it is especially on a heat map clearly visible that there are two main areas where shootings without and also with murder happens most. One is in The Bronx and the other in north of Brooklyn.

The outcome of the second question was really surprising to me, but it showed up, that there are significant more shootings in summer compared to winter. And even more prominent is the difference within the time of day. Between seven o'clock in the evening and four o'clock in the morning happens massive more shootings compared to the morning (6-11).

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=German_Germany.1252  LC_CTYPE=German_Germany.1252
## [3] LC_MONETARY=German_Germany.1252 LC_NUMERIC=C
## [5] LC_TIME=German_Germany.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] timetk_2.6.1     ggmap_3.0.0      forcats_0.5.1    stringr_1.4.0
##  [5] dplyr_1.0.5      purrr_0.3.4      readr_1.4.0      tidyr_1.1.3
##  [9] tibble_3.0.5     ggplot2_3.3.3    tidyverse_1.3.0  lubridate_1.7.10
##
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6        fs_1.5.0            xts_0.12.1
##  [4] httr_1.4.2          tools_4.0.3         backports_1.2.1
##  [7] R6_2.5.0            rpart_4.1-15        DBI_1.1.1
## [10] colorspace_2.0-0    nnet_7.3-14         withr_2.4.1
## [13] sp_1.4-5            tidyselect_1.1.0    curl_4.3
## [16] compiler_4.0.3      cli_2.2.0           rvest_1.0.0
## [19] xml2_1.3.2          isoband_0.2.4       labeling_0.4.2
## [22] scales_1.1.1        digest_0.6.27       rmarkdown_2.7
## [25] jpeg_0.1-8.1        pkgconfig_2.0.3     htmltools_0.5.1.1
## [28] parallelly_1.25.0   dbplyr_2.1.0        rlang_0.4.10
## [31] readxl_1.3.1        rstudioapi_0.13     farver_2.1.0
## [34] generics_0.1.0      zoo_1.8-9           jsonlite_1.7.2
## [37] magrittr_2.0.1      Matrix_1.2-18       Rcpp_1.0.6
## [40] munsell_0.5.0       fansi_0.4.2         lifecycle_1.0.0
## [43] furrr_0.2.2         stringi_1.5.3       yaml_2.2.1
## [46] MASS_7.3-53         plyr_1.8.6          recipes_0.1.16
## [49] grid_4.0.3          parallel_4.0.3      listenv_0.8.0
## [52] crayon_1.3.4        lattice_0.20-41     haven_2.3.1
## [55] splines_4.0.3       hms_1.0.0           knitr_1.30
## [58] pillar_1.4.7        rjson_0.2.20        codetools_0.2-16
## [61] reprex_1.0.0        glue_1.4.2          evaluate_0.14
## [64] rsample_0.1.0       modelr_0.1.8        png_0.1-7
## [67] vctrs_0.3.6         RgoogleMaps_1.4.5.3 cellranger_1.1.0
## [70] gtable_0.3.0        future_1.21.0       assertthat_0.2.1
## [73] xfun_0.20           gower_0.2.2         prodlim_2019.11.13
## [76] broom_0.7.5         class_7.3-17        survival_3.2-7
## [79] timeDate_3043.102   lava_1.6.9          globals_0.14.0
## [82] ellipsis_0.3.1      ipred_0.9-11
```