NYPD Shooting Incident Data Analysis

Input file

- Title: NYPD Shooting Incident Data (Historic)
- Url: https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv
- Dataset description: List of every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year.

```
options(repr.plot.width=30, repr.plot.height=8)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
url_in <- 'https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv'
NYPD <- read_csv(url_in)
```

```
## Rows: 23585 Columns: 19
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): OCCUR_DATE, BORO, LOCATION_DESC, PERP_AGE_GROUP, PERP_SEX, PERP_R...
## dbl   (7): INCIDENT_KEY, PRECINCT, JURISDICTION_CODE, X_COORD_CD, Y_COORD_CD...
## lgl   (1): STATISTICAL_MURDER_FLAG
## time  (1): OCCUR_TIME
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Check the data structure

```
library(dplyr)
glimpse(NYPD)
```

```
## Rows: 23,585
## Columns: 19
## $ INCIDENT_KEY            <dbl> 24050482, 77673979, 203350417, 80584527, 90843~
## $ OCCUR_DATE              <chr> "08/27/2006", "03/11/2011", "10/06/2019", "09/~
## $ OCCUR_TIME              <time> 05:35:00, 12:03:00, 01:09:00, 03:35:00, 21:16~
## $ BORO                    <chr> "BRONX", "QUEENS", "BROOKLYN", "BRONX", "QUEEN~
## $ PRECINCT                <dbl> 52, 106, 77, 40, 100, 67, 77, 81, 101, 106, 71~
## $ JURISDICTION_CODE       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ LOCATION_DESC           <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ STATISTICAL_MURDER_FLAG <lgl> TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE~
## $ PERP_AGE_GROUP          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ PERP_SEX                <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ PERP_RACE               <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ VIC_AGE_GROUP           <chr> "25-44", "65+", "18-24", "<18", "18-24", "<18"~
## $ VIC_SEX                 <chr> "F", "M", "F", "M", "M", "M", "M", "M", "M", "~
## $ VIC_RACE                <chr> "BLACK HISPANIC", "WHITE", "BLACK", "BLACK", "~
## $ X_COORD_CD              <dbl> 1017542, 1027543, 995325, 1007453, 1041267, 10~
## $ Y_COORD_CD              <dbl> 255918.9, 186095.0, 185155.0, 233952.0, 157133~
## $ Latitude                <dbl> 40.86906, 40.67737, 40.67489, 40.80880, 40.597~
## $ Longitude               <dbl> -73.87963, -73.84392, -73.96008, -73.91618, -7~
## $ Lon_Lat                 <chr> "POINT (-73.87963173099996 40.86905819000003)"~
```

Possible bias

1. Different boroughs may have different security levels, which means they have different numbers of shooting incidents. Brooklyn probably has a higher crime rate than other boro's.
2. Different age groups can have different shootings incident rate. 20s may be more inclined to shoot.

Analytics plan

1. Will check the number of incident by boro and age group to verify the bias above.
2. Also leverage modeling method to find the relation between number of death and number of incidents.

Data transform : Change OCCUR_DATE to date format

```
NYPD$OCCUR_DATE <- as.Date(NYPD$OCCUR_DATE,format='%m/%d/%Y')
```

Count the number of incident by each boro in NY in 2020 to see which area has more cases

```
NYPD_by_boro_2020 <- NYPD %>%
  mutate(OCCUR_DATE = as.Date(OCCUR_DATE,format='%m/%d/%Y') ) %>%
  filter( between(OCCUR_DATE, as.Date("2020-01-01"), as.Date("2020-12-31"))  ) %>%
  group_by(BORO) %>%
  summarize(cases =n()) %>%
  select(BORO,cases) %>%
  ungroup()

NYPD_by_boro_2020
```

```
## # A tibble: 5 x 2
##   BORO          cases
##   <chr>         <int>
## 1 BRONX           504
## 2 BROOKLYN        819
## 3 MANHATTAN       272
## 4 QUEENS          303
## 5 STATEN ISLAND    50
```

Get the boro with the highest number of shooting incident

```
NYPD_by_boro_2020 %>%
  slice_max(cases, n=1)
```

```
## # A tibble: 1 x 2
##   BORO      cases
##   <chr>     <int>
## 1 BROOKLYN    819
```

Get the death rate of shooting incident for each boro in 2020

```
NYPD_death_rate_by_boro <- NYPD %>%
    mutate(OCCUR_DATE = as.Date(OCCUR_DATE,format='%m/%d/%Y') ) %>%
    filter( between(OCCUR_DATE, as.Date("2020-01-01"), as.Date("2020-12-31"))  ) %>%
    group_by(BORO) %>%
    summarize(cases =n(),deaths = sum(STATISTICAL_MURDER_FLAG)) %>%
    mutate(deaths_rate = round(deaths / cases,3)) %>%
    ungroup()
NYPD_death_rate_by_boro
```

```
## # A tibble: 5 x 4
##   BORO          cases deaths deaths_rate
##   <chr>         <int>  <int>       <dbl>
## 1 BRONX           504     86       0.171
## 2 BROOKLYN        819    161       0.197
## 3 MANHATTAN       272     47       0.173
## 4 QUEENS          303     56       0.185
## 5 STATEN ISLAND    50     16       0.32
```

Get the boro with the highest death rate in shooting incident

```
NYPD_death_rate_by_boro %>%
  slice_max(deaths_rate, n=1)
```

```
## # A tibble: 1 x 4
##   BORO          cases deaths deaths_rate
##   <chr>         <int>  <int>       <dbl>
## 1 STATEN ISLAND    50     16        0.32
```

Count the number of incident by Perpetrator's age group in 2020

```
NYPD_by_age <- NYPD %>%
    mutate(OCCUR_DATE = as.Date(OCCUR_DATE,format='%m/%d/%Y') ) %>%
    filter( between(OCCUR_DATE, as.Date("2020-01-01"), as.Date("2020-12-31"))  ) %>%
    filter(! is.na(PERP_AGE_GROUP)  ) %>%
    group_by(PERP_AGE_GROUP) %>%
    summarize(cases =n()) %>%
    select(PERP_AGE_GROUP,cases) %>%
    ungroup()
NYPD_by_age
```

```
## # A tibble: 5 x 2
##   PERP_AGE_GROUP cases
##   <chr>          <int>
## 1 <18               77
## 2 18-24            298
## 3 25-44            428
## 4 45-64             72
## 5 65+                3
```

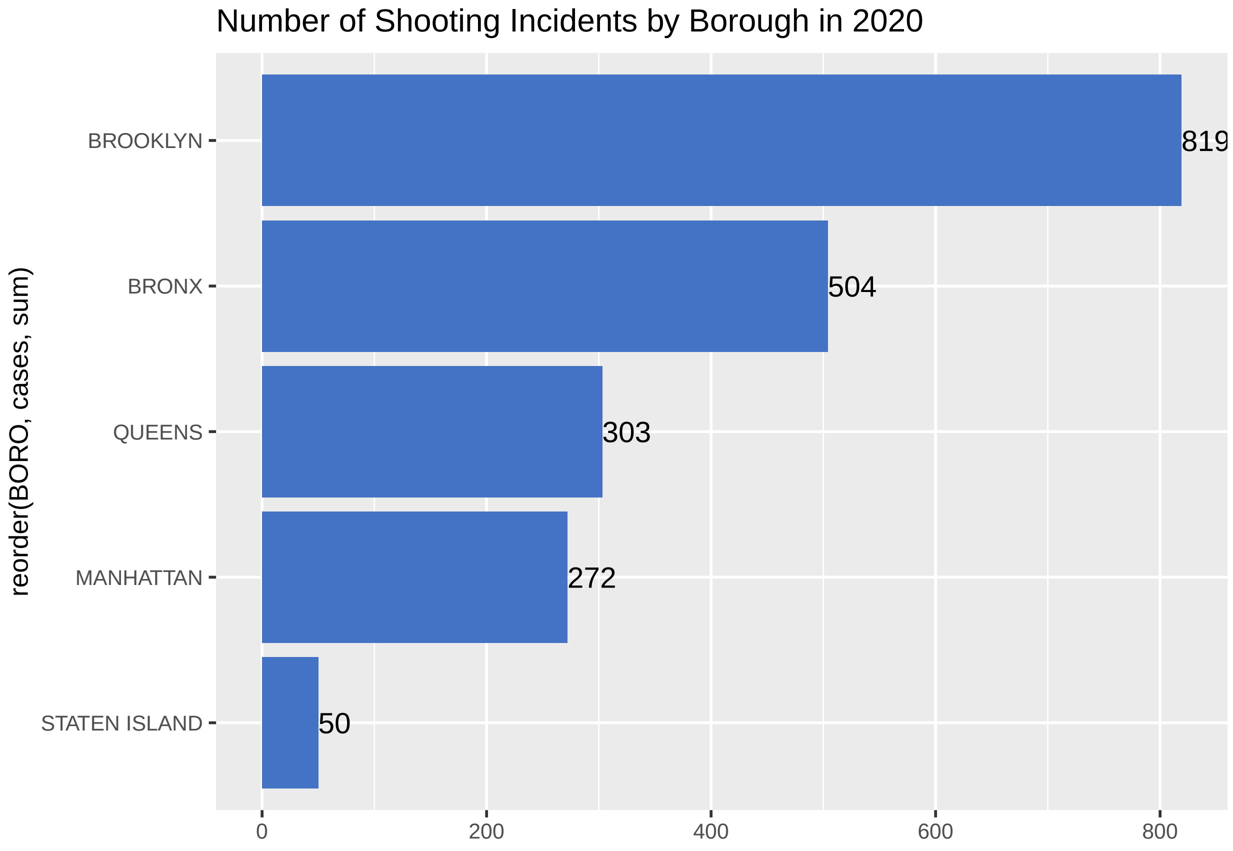
Get the age group with the highest number of incident

```
NYPD_by_age %>%
  slice_max(cases, n=1)
```

```
## # A tibble: 1 x 2
##   PERP_AGE_GROUP cases
##   <chr>          <int>
## 1 25-44            428
```

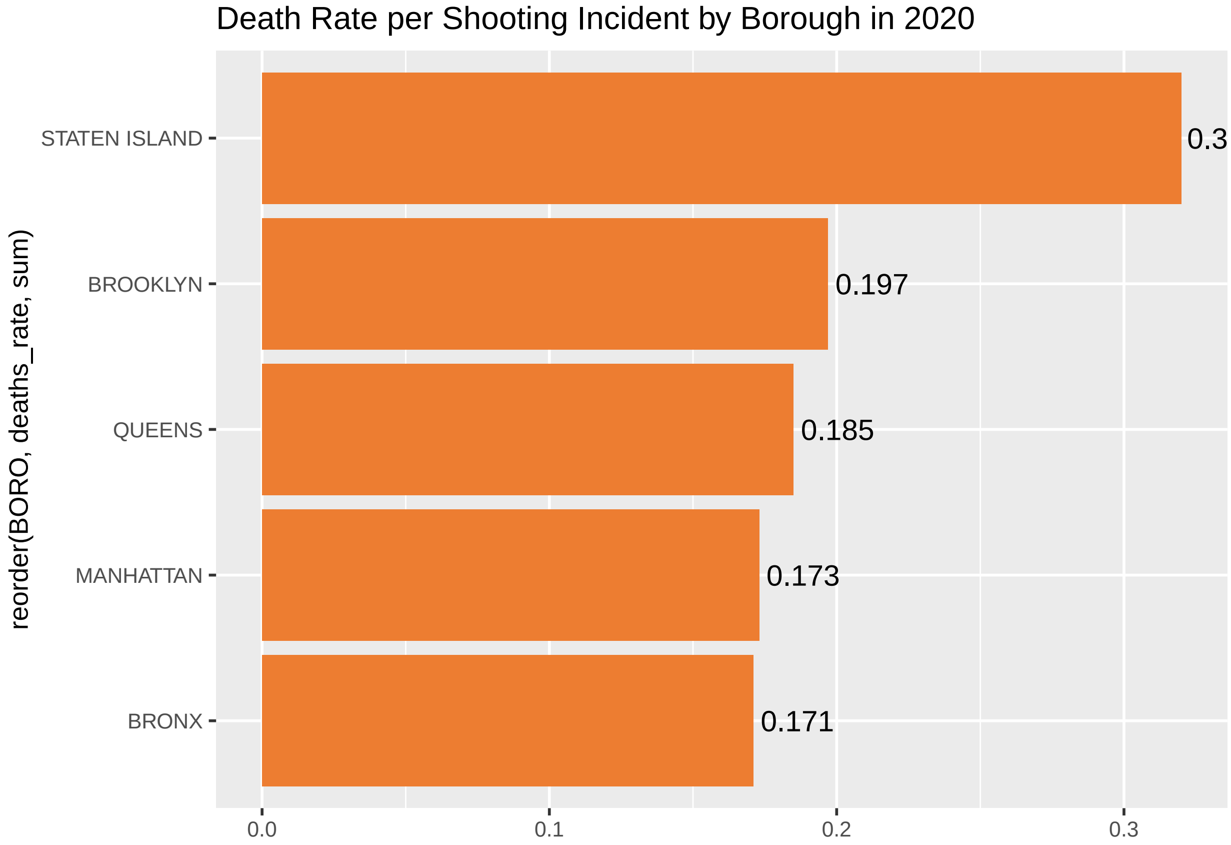
Visualize number of cases by boro

```
options(repr.plot.width=30, repr.plot.height=8)
ggplot(NYPD_by_boro_2020, aes(reorder(BORO, cases, sum), cases)) +  geom_col(fill = "#4472C4") +
  geom_text(aes(label=cases), position=position_dodge(width=0.9), hjust=0) +
  coord_flip() +
  labs(title = "Number of Shooting Incidents by Borough in 2020", y= NULL)+
  theme(text=element_text(size=10))
```



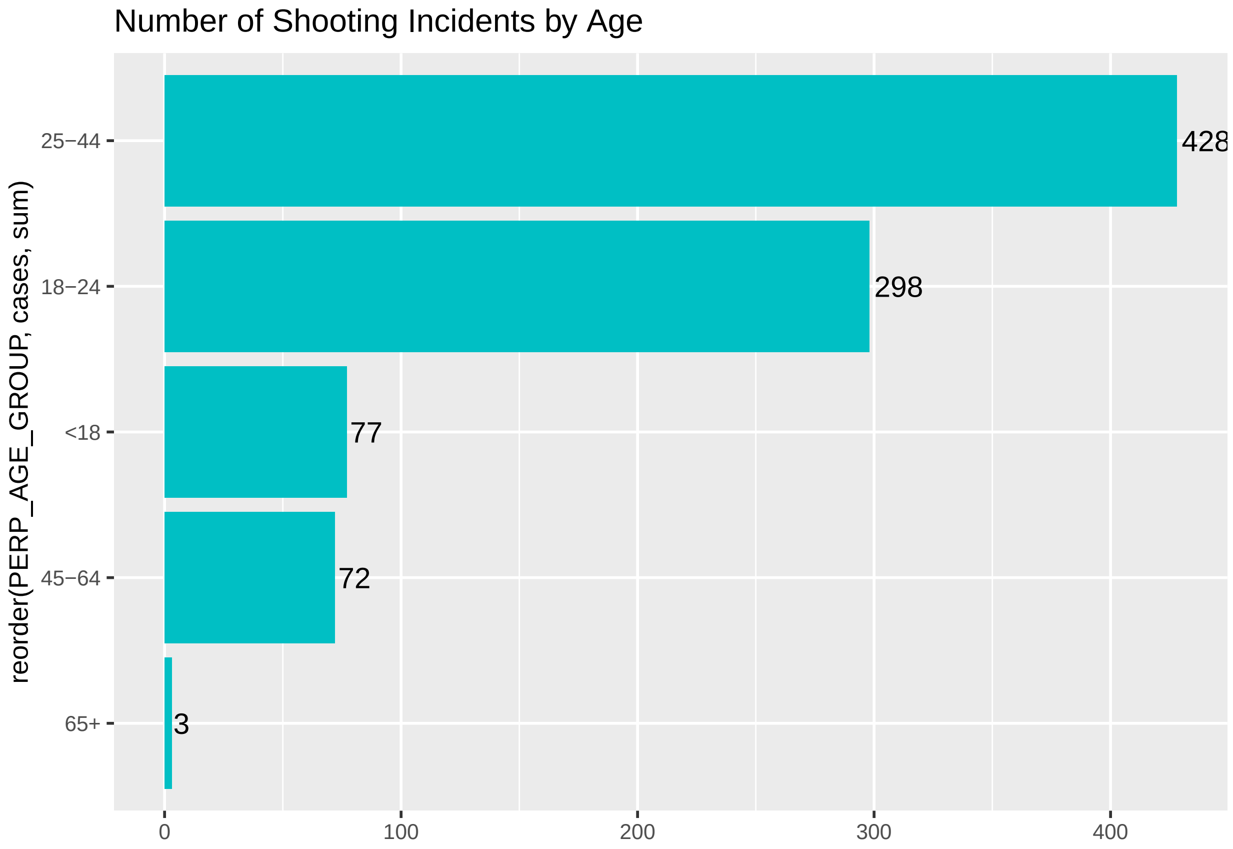
Visualize death rate per incident by boro

```
options(repr.plot.width=30, repr.plot.height=8)
ggplot(NYPD_death_rate_by_boro, aes(reorder(BORO, deaths_rate, sum), deaths_rate)) +  geom_col(fill = "#
  geom_text(aes(label=deaths_rate), position=position_dodge(width=0.9), hjust=-0.1) +
  coord_flip() +
  labs(title = "Death Rate per Shooting Incident by Borough in 2020", y= NULL) +
  theme(text=element_text(size=10)) #change font size of legend title
```



Visualize number of cases by age group

```
options(repr.plot.width=30, repr.plot.height=8)
ggplot(NYPD_by_age, aes(reorder(PERP_AGE_GROUP, cases, sum), cases)) +  geom_col(fill = "#00BFC4") + co
    geom_text(aes(label=cases), position=position_dodge(width=0.9), hjust=-0.1) +
    labs(title = "Number of Shooting Incidents by Age", y= NULL) +
    theme(text=element_text(size=10))
```



Build model to see the relationship between number of deaths and number of shooting incidents

```
NYPD_by_month <- NYPD %>%
    mutate(OCCUR_DATE = as.Date(OCCUR_DATE,format='%m/%d/%Y') ) %>%
    mutate(OCCUR_MONTH = strftime(OCCUR_DATE,format='%Y/%m') ) %>%
    mutate(Month = strftime(OCCUR_DATE,format='%m') ) %>%
#     filter( between(OCCUR_DATE, as.Date("2020-01-01"), as.Date("2020-12-31"))  ) %>%
    group_by(OCCUR_MONTH,Month) %>%
    summarize(deaths = sum(STATISTICAL_MURDER_FLAG), cases= n())
```

```
## `summarise()` has grouped output by 'OCCUR_MONTH'. You can override using the `.groups` argument.
```

Build model to see the relationship between number of deaths and number of shooting incidents

```
mod <- lm(deaths  ~ cases  , data = NYPD_by_month)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths ~ cases, data = NYPD_by_month)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3916  -3.8531  -0.0315   3.5552  21.5726
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.36199    1.13586   1.199    0.232
## cases        0.18041    0.00804  22.438   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.697 on 178 degrees of freedom
## Multiple R-squared:  0.7388, Adjusted R-squared:  0.7373
## F-statistic: 503.5 on 1 and 178 DF,  p-value: < 2.2e-16
```

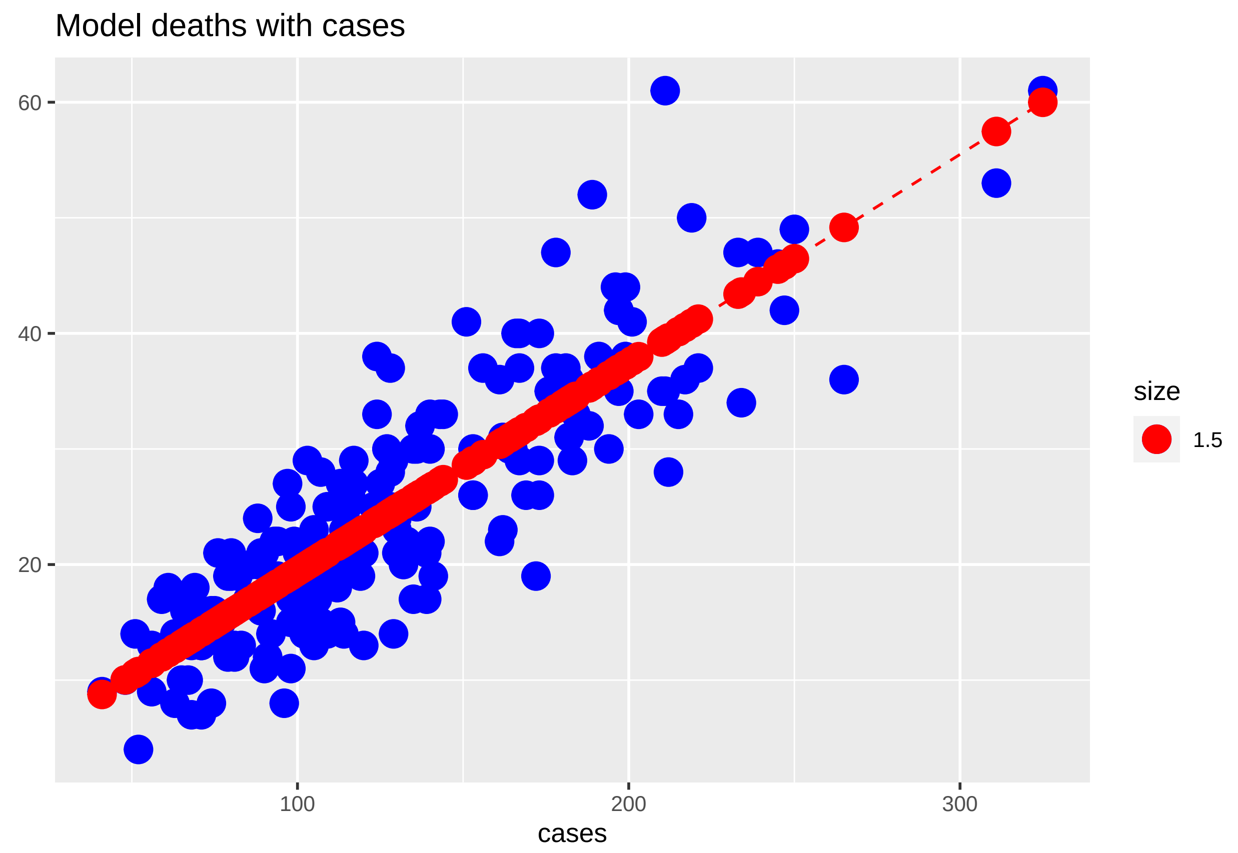
Predict number of deaths with model

```
pred <- tibble(pred = predict(mod))

NYPD_by_month_w_pred <- cbind(NYPD_by_month,pred)
```

Plot predicted deaths and actual deaths

```
options(repr.plot.width=30, repr.plot.height=8)
NYPD_by_month_w_pred %>% ggplot() +
   geom_point(aes(x=cases, y=deaths, size = 1.5), color = "blue") +
   geom_point(aes(x = cases, y = pred,size = 1.5), color = "red") +
   geom_line(aes(x = cases, y = pred ),linetype = "dashed", color = "red") +
   labs(title = "Model deaths with cases", y= NULL) +
   theme(
#        legend.position="bottom",
    text=element_text(size=10)) #change font size of legend title
```



Build model to see the relationship between number of shooting incidents and calendar month

```
mod_m <- lm(cases ~ Month , data = NYPD_by_month)
summary(mod_m)
```

```
##
## Call:
## lm(formula = cases ~ Month, data = NYPD_by_month)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -92.93 -29.02   1.90  27.07 138.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  100.333     10.781   9.306  < 2e-16 ***
## Month02      -23.733     15.247  -1.557  0.12145
## Month03       -6.867     15.247  -0.450  0.65303
## Month04        9.467     15.247   0.621  0.53552
## Month05       44.600     15.247   2.925  0.00392 **
## Month06       63.533     15.247   4.167 4.93e-05 ***
## Month07       86.667     15.247   5.684 5.69e-08 ***
## Month08       84.600     15.247   5.549 1.10e-07 ***
## Month09       47.933     15.247   3.144  0.00197 **
## Month10       34.133     15.247   2.239  0.02649 *
## Month11       12.467     15.247   0.818  0.41472
## Month12       15.533     15.247   1.019  0.30977
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 41.76 on 168 degrees of freedom
## Multiple R-squared:  0.4167, Adjusted R-squared:  0.3785
## F-statistic: 10.91 on 11 and 168 DF,  p-value: 4.6e-15
```

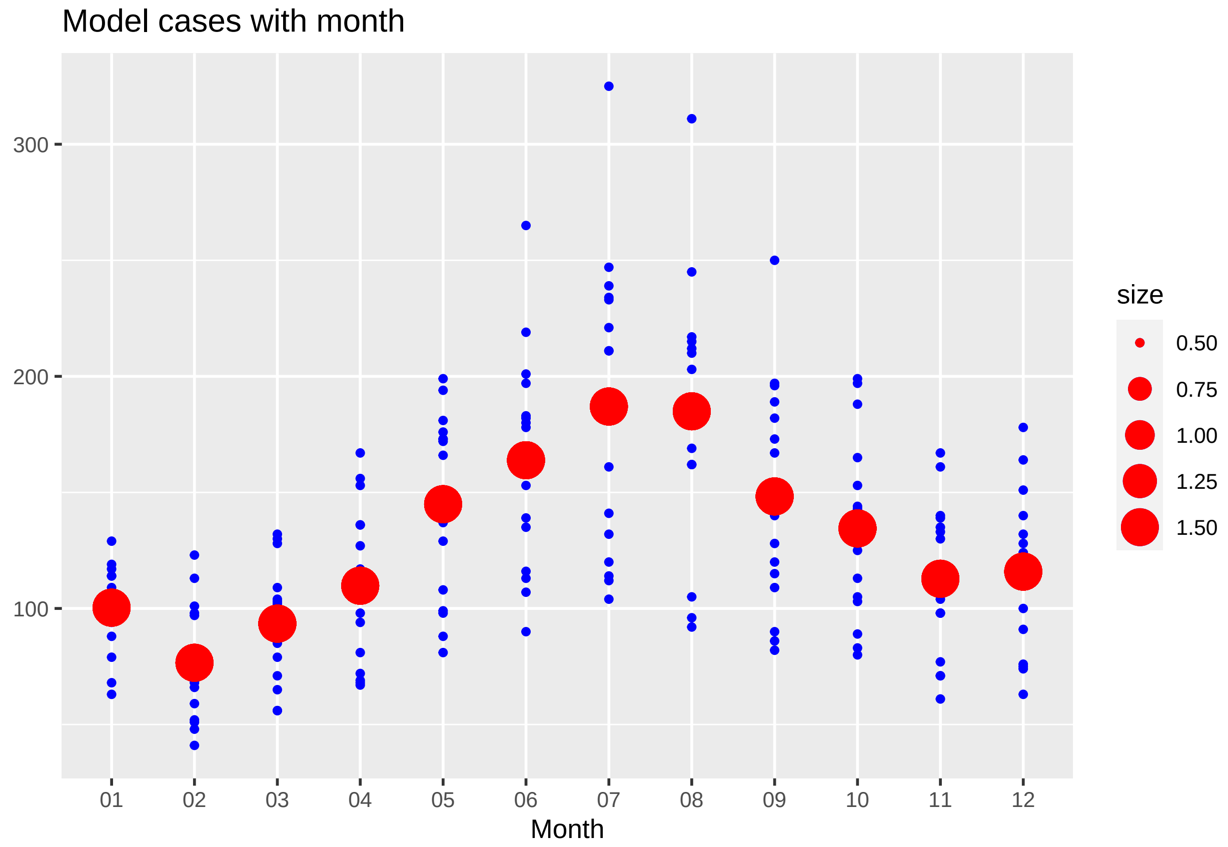
Predict number of cases with model

```
pred_m <- tibble(pred_m = predict(mod_m))

NYPD_by_month_w_pred <- cbind(NYPD_by_month_w_pred,pred_m)
```

Plot predicted deaths and actual deaths

```
options(repr.plot.width=30, repr.plot.height=8)
NYPD_by_month_w_pred %>% ggplot() +
   geom_point(aes(x=Month, y=cases, size = 0.5), color = "blue") +
   geom_line(aes(x = Month, y = pred_m ),linetype = "dashed", color = "red") +
   geom_point(aes(x = Month, y = pred_m,size = 1.5), color = "red") +
   labs(title = "Model cases with month", y= NULL) +
   theme(
#        legend.position="bottom",
    text=element_text(size=10)) #change font size of legend title
```



Conclusion

1. Brooklyn has had more shootings than any other borough.
2. The Staten Island has the highest fatality rate from shootings.
3. In 2020 in NY, the shootings were mainly committed by people who are 25~44.
4. 18% chance of dying in a shooting incident.
5. There were more shootings in July and August than any other month.

my session info

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18363)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936
## [2] LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.936
## system code page: 65001
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
```

```
##
## other attached packages:
## [1] forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7     purrr_0.3.4
## [5] readr_2.1.1     tidyr_1.1.4     tibble_3.1.6    ggplot2_3.3.5
## [9] tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8       lubridate_1.8.0  assertthat_0.2.1 digest_0.6.29
##  [5] utf8_1.2.2       R6_2.5.1         cellranger_1.1.0 backports_1.4.1
##  [9] reprex_2.0.1     evaluate_0.14    highr_0.9        httr_1.4.2
## [13] pillar_1.6.4     rlang_0.4.12     curl_4.3.2       readxl_1.3.1
## [17] rstudioapi_0.13  rmarkdown_2.11   labeling_0.4.2   bit_4.0.4
## [21] munsell_0.5.0    broom_0.7.11     compiler_4.1.2   modelr_0.1.8
## [25] xfun_0.29        pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.1
## [29] fansi_1.0.0      crayon_1.4.2     tzdb_0.2.0       dbplyr_2.1.1
## [33] withr_2.4.3      grid_4.1.2       jsonlite_1.7.2   gtable_0.3.0
## [37] lifecycle_1.0.1  DBI_1.1.2        magrittr_2.0.1   scales_1.1.1
## [41] cli_3.1.0        stringi_1.7.6    vroom_1.5.7      farver_2.1.0
## [45] fs_1.5.2         xml2_1.3.3       ellipsis_0.3.2   generics_0.1.1
## [49] vctrs_0.3.8      tools_4.1.2      bit64_4.0.5      glue_1.6.0
## [53] hms_1.1.1        parallel_4.1.2   fastmap_1.1.0    yaml_2.2.1
## [57] colorspace_2.0-2 rvest_1.0.2      knitr_1.37       haven_2.4.3
```