NYPD Shooting Incidents Data Analysis

Engin Deniz Dogu

2024-09-20

NYPD Shooting Data

Reading data…

```
nypd_shooting_data <- readr::read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv?accessType=DOWNLOAD")
```

```
## Rows: 28562 Columns: 21
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (12): OCCUR_DATE, BORO, LOC_OF_OCCUR_DESC, LOC_CLASSFCTN_DESC, LOCATION...
## 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.
```

Data cleaning steps…

```
# Date conversion
nypd_shooting_data$OCCUR_DATE <- as.Date(nypd_shooting_data$OCCUR_DATE, "%m/%d/%Y")

# Calculate total incidents in each BORO
total_incidents_in_boro <- nypd_shooting_data |> dplyr::group_by(BORO) |>
  dplyr::summarize(cases=dplyr::n(),cases_per_thou = dplyr::n()/1000,
                   deaths_per_thou=sum(STATISTICAL_MURDER_FLAG==TRUE)/1000)

# Daily cases
cases_and_deaths_daily <- nypd_shooting_data |> dplyr::group_by(OCCUR_DATE) |>
  dplyr::summarize(cases=dplyr::n(),cases_per_thou = dplyr::n()/1000,
                   deaths_per_thou=sum(STATISTICAL_MURDER_FLAG==TRUE)/1000)

# Data Exploration (unique values)
boro_uniq <- unique(nypd_shooting_data["BORO"])
perp_race_uniq <- unique(nypd_shooting_data["PERP_RACE"])
perp_sex_uniq <- unique(nypd_shooting_data["PERP_SEX"])
perp_age_group_uniq <- unique(nypd_shooting_data["PERP_AGE_GROUP"])
vic_race_uniq <- unique(nypd_shooting_data["VIC_RACE"])
vic_sex_uniq <- unique(nypd_shooting_data["VIC_SEX"])
vic_age_group_uniq <- unique(nypd_shooting_data["VIC_AGE_GROUP"])
location_desc_uniq <- unique(nypd_shooting_data["LOCATION_DESC"])

# Discovery & Factors
boro_f <- c(unique(nypd_shooting_data$BORO))

perp_race_f <- subset(perp_race_uniq,!is.na(perp_race_uniq$PERP_RACE)
                      & perp_race_uniq$PERP_RACE != "(null)")$PERP_RACE

perp_sex_f <- subset(perp_sex_uniq,!is.na(perp_sex_uniq$PERP_SEX)
                      & perp_sex_uniq$PERP_SEX != "(null)")$PERP_SEX

perp_age_group_uniq_f <- subset(perp_age_group_uniq,!is.na(perp_age_group_uniq$PERP_AGE_GROUP)
                      & perp_age_group_uniq$PERP_AGE_GROUP != "(null)"
                      & perp_age_group_uniq$PERP_AGE_GROUP != "1020"
                      & perp_age_group_uniq$PERP_AGE_GROUP != "940"
                      & perp_age_group_uniq$PERP_AGE_GROUP != "224"
                      & perp_age_group_uniq$PERP_AGE_GROUP != "1028"
```

```
                      )$PERP_AGE_GROUP

vic_race_uniq_f <- subset(vic_race_uniq,!is.na(vic_race_uniq$VIC_RACE)
                      & vic_race_uniq$VIC_RACE != "(null)")$VIC_RACE

vic_sex_uniq_f <- subset(vic_sex_uniq,!is.na(vic_sex_uniq$VIC_SEX)
                      & vic_sex_uniq$VIC_SEX != "(null)")$VIC_SEX

vic_age_group_uniq_f <- subset(vic_age_group_uniq,!is.na(vic_age_group_uniq$VIC_AGE_GROUP)
                      & vic_age_group_uniq$VIC_AGE_GROUP != "(null)"
                      & vic_age_group_uniq$VIC_AGE_GROUP != "1022"
                      )$VIC_AGE_GROUP

location_desc_uniq_f <- subset(location_desc_uniq,!is.na(location_desc_uniq$LOCATION_DESC)
                      & location_desc_uniq$LOCATION_DESC != "(null)"
                      )$LOCATION_DESC

# Remove unwanted columns
nypd_shooting_data <- subset(nypd_shooting_data,
                             select=-c(Lon_Lat,X_COORD_CD,Y_COORD_CD,JURISDICTION_CODE,OCCUR_TIME))

# Remove unwanted rows
nypd_shooting_data <- subset(nypd_shooting_data,nypd_shooting_data$PERP_RACE %in% perp_race_f)
nypd_shooting_data <- subset(nypd_shooting_data,nypd_shooting_data$PERP_SEX %in% perp_sex_f)
nypd_shooting_data <- subset(nypd_shooting_data,nypd_shooting_data$PERP_AGE_GROUP %in% perp_age_group_uniq_f)
nypd_shooting_data <- subset(nypd_shooting_data,nypd_shooting_data$VIC_RACE %in% vic_race_uniq_f)
nypd_shooting_data <- subset(nypd_shooting_data,nypd_shooting_data$VIC_SEX %in% vic_sex_uniq_f)
nypd_shooting_data <- subset(nypd_shooting_data,nypd_shooting_data$VIC_AGE_GROUP %in% vic_age_group_uniq_f)
nypd_shooting_data <- subset(nypd_shooting_data,nypd_shooting_data$LOCATION_DESC %in% location_desc_uniq_f)

# Summary and missing data check
summary(nypd_shooting_data)
```

```
##   INCIDENT_KEY         OCCUR_DATE             BORO           LOC_OF_OCCUR_DESC
##  Min.   :  9953245   Min.   :2006-01-01   Length:8017        Length:8017
##  1st Qu.: 47156715   1st Qu.:2008-06-10   Class :character   Class :character
##  Median : 77888546   Median :2011-03-23   Mode  :character   Mode  :character
##  Mean   :105747095   Mean   :2012-11-09
##  3rd Qu.:162945059   3rd Qu.:2017-03-19
##  Max.   :279472658   Max.   :2023-12-24
##
##     PRECINCT      LOC_CLASSFCTN_DESC LOCATION_DESC      STATISTICAL_MURDER_FLAG
##  Min.   :  1.00   Length:8017        Length:8017        Mode :logical
##  1st Qu.: 43.00   Class :character   Class :character   FALSE:6153
##  Median : 67.00   Mode  :character   Mode  :character   TRUE :1864
##  Mean   : 64.94
##  3rd Qu.: 81.00
##  Max.   :123.00
##
##  PERP_AGE_GROUP       PERP_SEX          PERP_RACE         VIC_AGE_GROUP
##  Length:8017        Length:8017        Length:8017        Length:8017
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    VIC_SEX            VIC_RACE            Latitude       Longitude
##  Length:8017        Length:8017        Min.   :40.53   Min.   :-74.21
##  Class :character   Class :character   1st Qu.:40.67   1st Qu.:-73.95
##  Mode  :character   Mode  :character   Median :40.71   Median :-73.92
##                                        Mean   :40.74   Mean   :-73.91
##                                        3rd Qu.:40.82   3rd Qu.:-73.89
##                                        Max.   :40.91   Max.   :-73.71
##                                        NA's   :13      NA's   :13
```

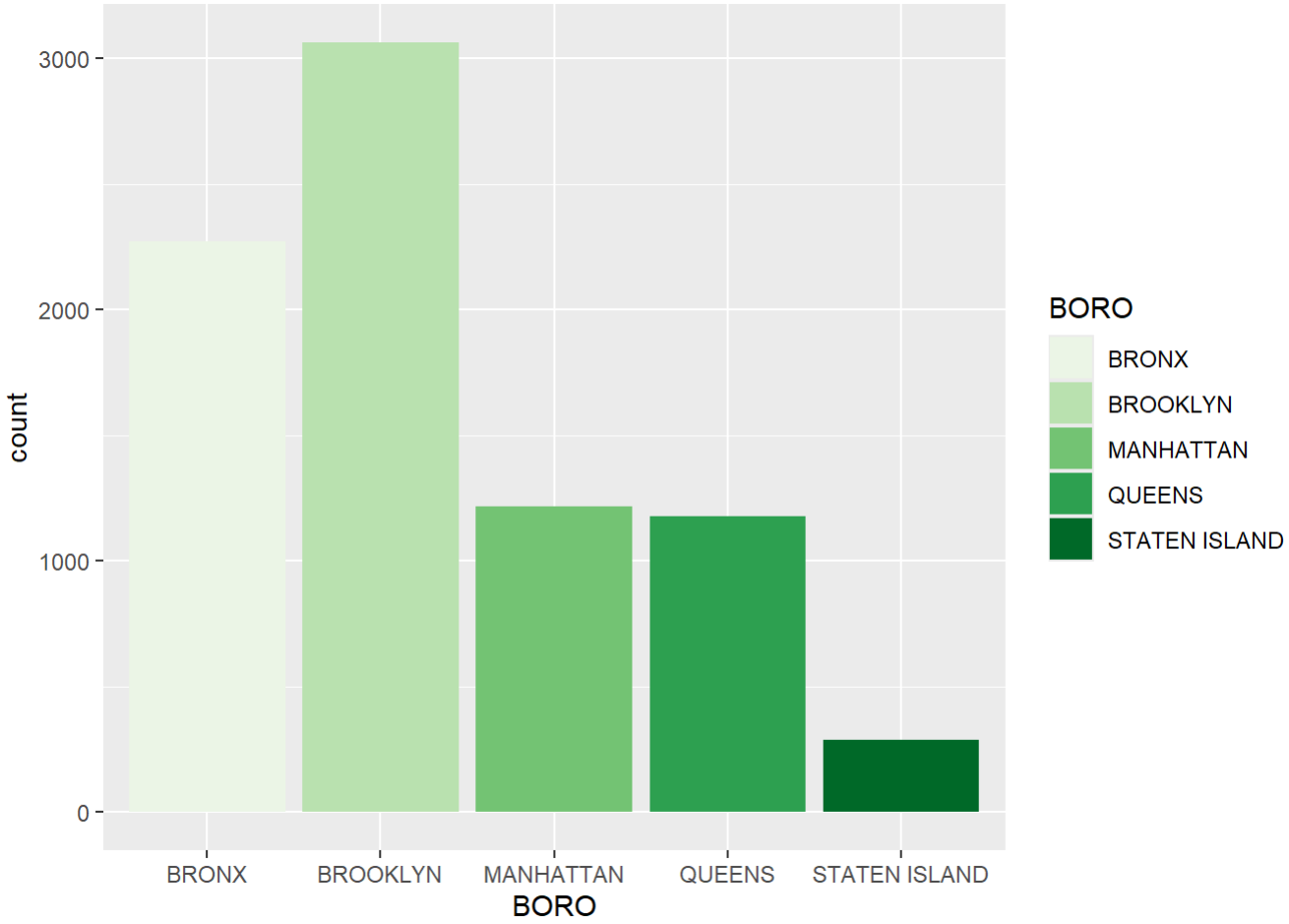
```
shooting_totals <- nypd_shooting_data |> dplyr::group_by(BORO, OCCUR_DATE) |>
  dplyr::summarize(cases=sum(INCIDENT_KEY), deaths=sum(STATISTICAL_MURDER_FLAG==TRUE))
```

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

```
shooting_totals <- subset(shooting_totals,deaths>0)
```

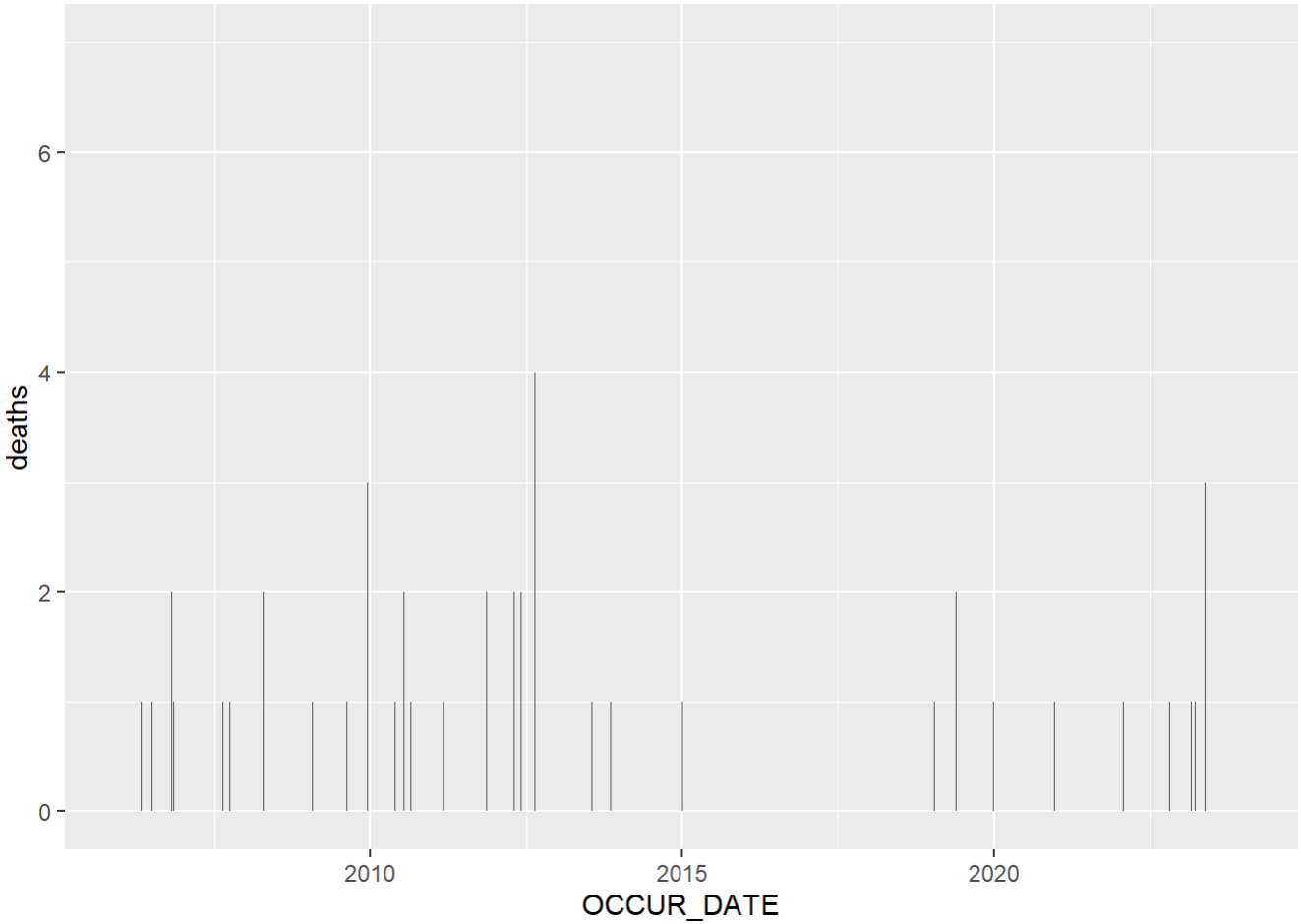
Total Cases by BORO

```
ggplot2::ggplot(nypd_shooting_data, ggplot2::aes(x=BORO,fill=BORO)) + ggplot2::geom_bar() + ggplot2::scale_fill_brewer(palet
te="Greens")
```

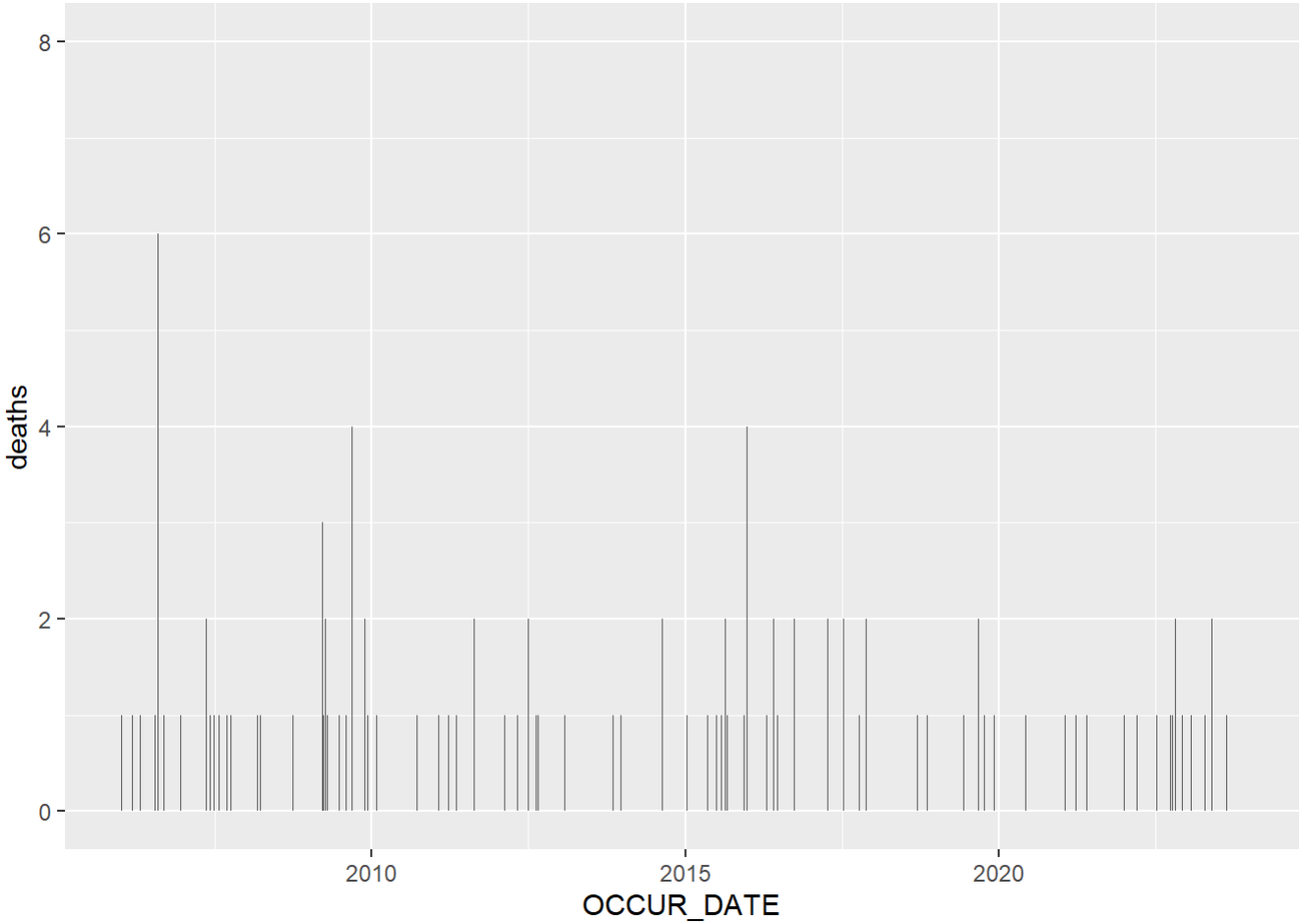


Comparison of Daily Deaths in MANHATTAN & BROOKLYN

```
boro_selection <- "MANHATTAN"
selection <- subset(shooting_totals, BORO==boro_selection)
ggplot2::ggplot(selection, ggplot2::aes(x = OCCUR_DATE, y = deaths)) + ggplot2::geom_col()
```




```
boro_selection <- "BROOKLYN"
selection <- subset(shooting_totals, BORO==boro_selection)
ggplot2::ggplot(selection, ggplot2::aes(x = OCCUR_DATE, y = deaths)) + ggplot2::geom_col()
```



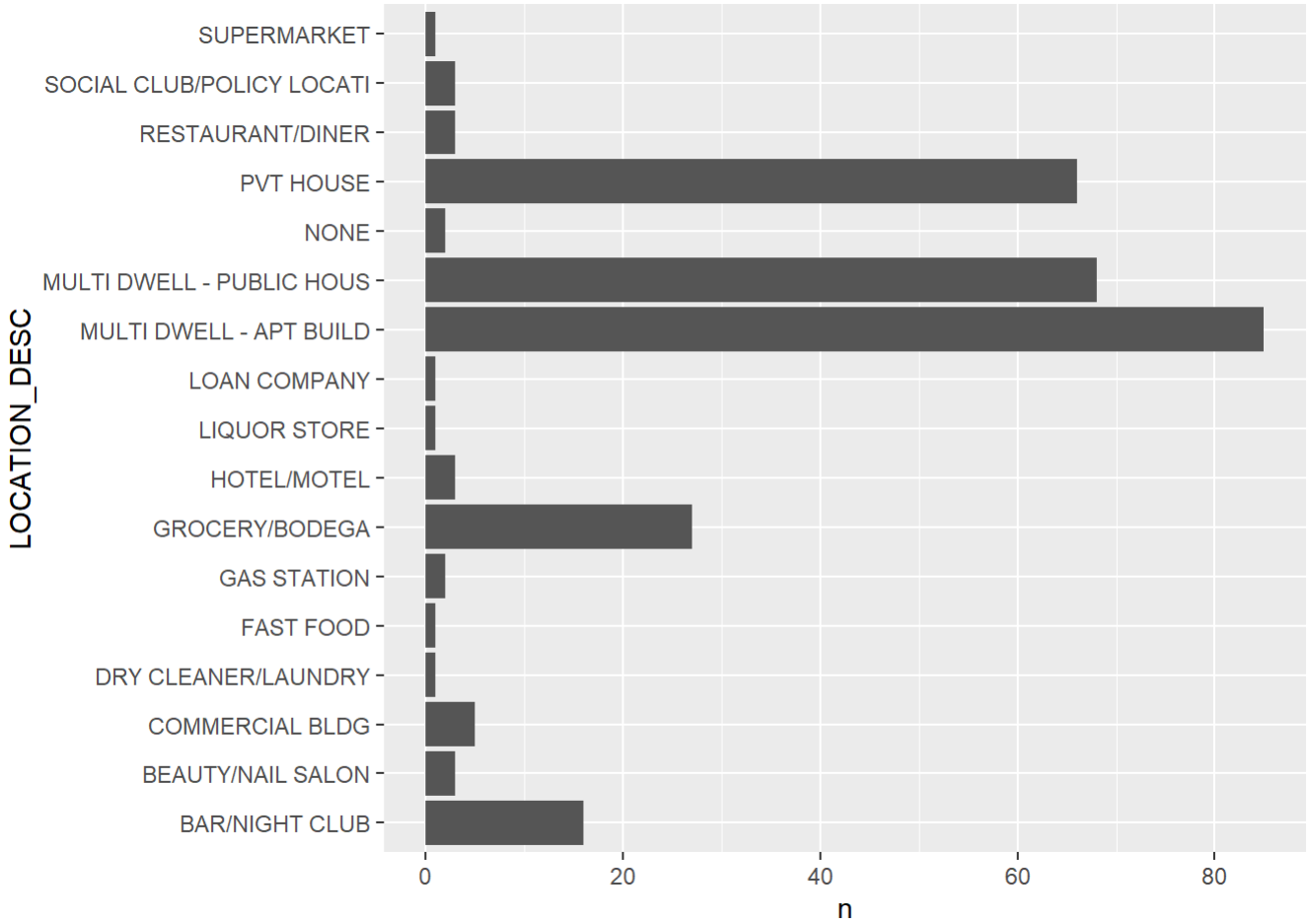
Number of Cases in Staten Island by Location Type

It seems that public housing, private housing and apartment buildings have the most shooting cases which might be interesting to look into.

```
cases_by_location <- nypd_shooting_data |> dplyr::group_by(BORO,LOCATION_DESC) |> dplyr::summarize(n = dplyr::n())
```

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

```
boro_selection <- "STATEN ISLAND"
selection <- subset(cases_by_location, BORO==boro_selection)
ggplot2::ggplot(selection, ggplot2::aes(x = n, y = LOCATION_DESC)) + ggplot2::geom_col()
```



Modelling 1

Deaths per Thousand by BORO (predicted values in red). Deaths per thousand is highly correlated with the number of cases.

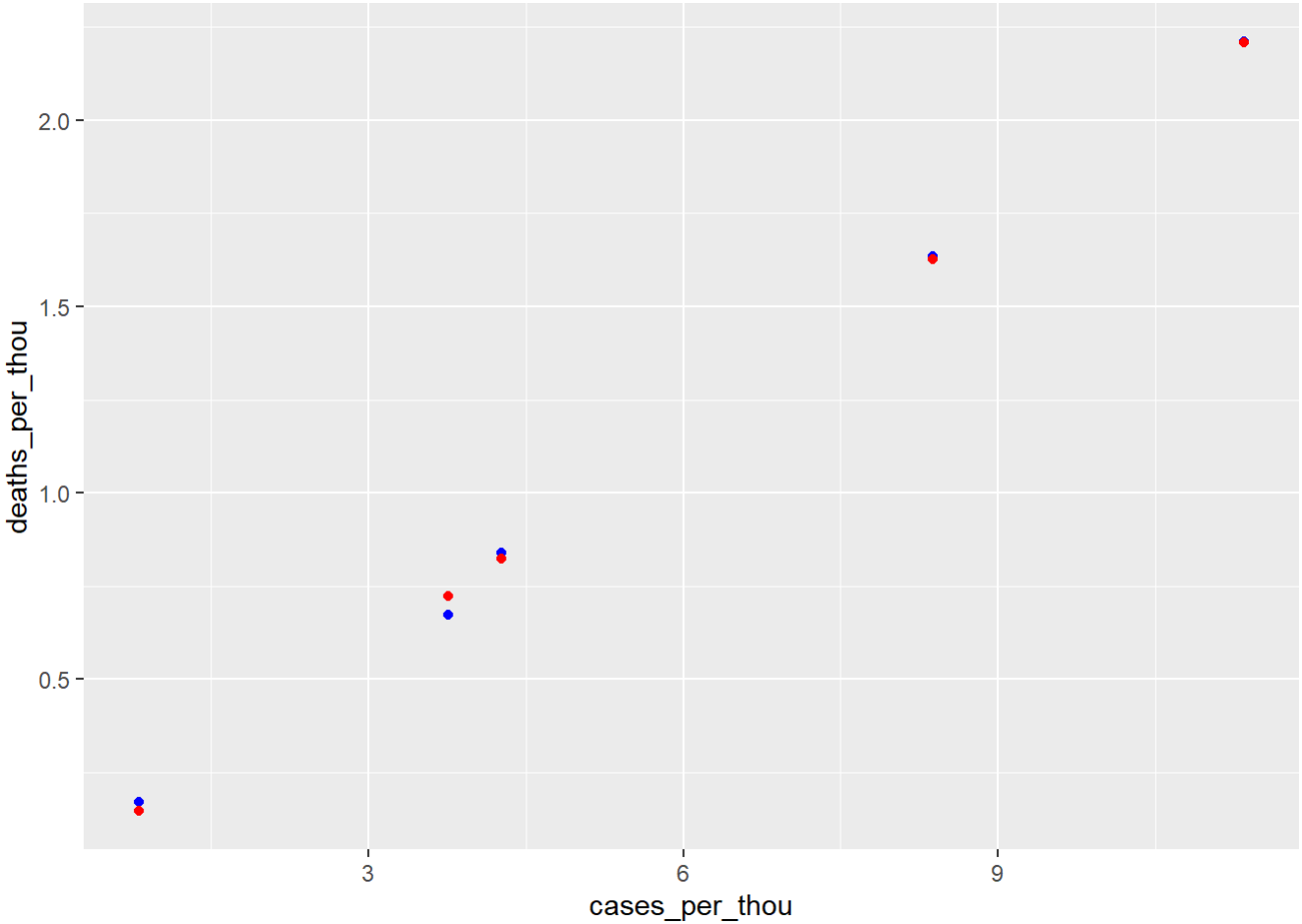
```
model_1 <- lm(deaths_per_thou ~ cases_per_thou, data=total_incidents_in_boro)

summary(model_1)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = total_incidents_in_boro)
##
## Residuals:
##         1         2         3         4         5
##  0.008014  0.003321 -0.051859  0.016622  0.023901
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.011685   0.028497   -0.41    0.709
## cases_per_thou  0.195519   0.004184   46.73 2.16e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0347 on 3 degrees of freedom
## Multiple R-squared:  0.9986, Adjusted R-squared:  0.9982
## F-statistic:  2184 on 1 and 3 DF,  p-value: 2.158e-05
```

```
total_incidents_in_boro_pred <- total_incidents_in_boro |>
  dplyr::mutate(pred=predict(model_1))

total_incidents_in_boro_pred |> ggplot2::ggplot() +
  ggplot2::geom_point(ggplot2::aes(x=cases_per_thou, y=deaths_per_thou), color="blue") +
  ggplot2::geom_point(ggplot2::aes(x=cases_per_thou, y=pred),color="red")
```



Modelling 2

Deaths per Thousand by Date (predicted values in red). It seems listing by date does not produce a good fit!

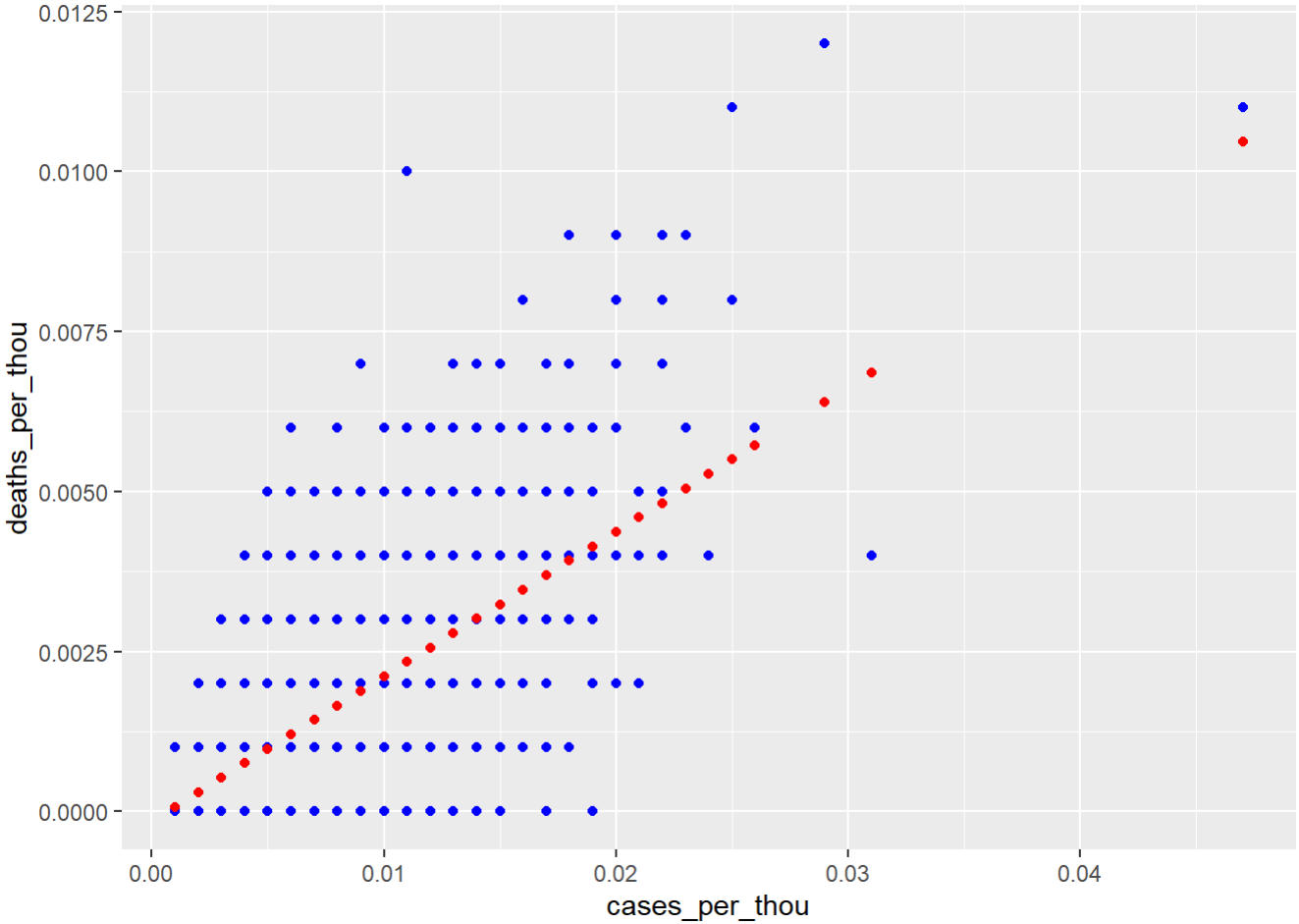
```
model_2 <- lm(deaths_per_thou ~ cases_per_thou, data=cases_and_deaths_daily)

summary(model_2)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = cases_and_deaths_daily)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0041414 -0.0005256 -0.0000736  0.0004744  0.0076665
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.523e-04  2.081e-05  -7.321 2.77e-13 ***
## cases_per_thou  2.260e-01  3.538e-03  63.870  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0009817 on 6093 degrees of freedom
## Multiple R-squared:  0.401,  Adjusted R-squared:  0.4009
## F-statistic:  4079 on 1 and 6093 DF,  p-value: < 2.2e-16
```

```
cases_and_deaths_daily_pred <- cases_and_deaths_daily |>
  dplyr::mutate(pred=predict(model_2))

cases_and_deaths_daily_pred |> ggplot2::ggplot() +
  ggplot2::geom_point(ggplot2::aes(x=cases_per_thou, y=deaths_per_thou), color="blue") +
  ggplot2::geom_point(ggplot2::aes(x=cases_per_thou, y=pred),color="red")
```



Sources of Bias and Final Thoughts

Potential sources of bias in NYPD shooting data:

- Gender
- Race
- Age
- Location

When I first looked at the I thought locations like bars/night clubs would have more shooting cases. However the seems to point at a different direction. It seems that most of the shootings are located in houses and apartments. My bias was that I considered houses to be safe. However, when I look at the analysis high numbers of shooting cases in houses and apartments actually make sense. Since these are private spaces (you

wouldn’t have cameras, police nearby etc.) it is possible to have a higher number of shooting cases related to these areas.

Session Summary

```
## R version 4.4.1 (2024-06-14 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: Europe/Istanbul
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] bit_4.0.5          gtable_0.3.5       jsonlite_1.8.8     highr_0.11
##  [5] dplyr_1.1.4        compiler_4.4.1     crayon_1.5.3       tidyselect_1.2.1
##  [9] parallel_4.4.1     jquerylib_0.1.4    scales_1.3.0       yaml_2.3.10
## [13] fastmap_1.2.0      ggplot2_3.5.1      readr_2.1.5        R6_2.5.1
## [17] labeling_0.4.3     generics_0.1.3     curl_5.2.2         knitr_1.48
## [21] tibble_3.2.1       munsell_0.5.1      RColorBrewer_1.1-3 bslib_0.8.0
## [25] pillar_1.9.0       tzdb_0.4.0         rlang_1.1.4        utf8_1.2.4
## [29] cachem_1.1.0       xfun_0.47          sass_0.4.9         bit64_4.0.5
## [33] cli_3.6.3          withr_3.0.1        magrittr_2.0.3     digest_0.6.37
## [37] grid_4.4.1         vroom_1.6.5        rstudioapi_0.16.0  hms_1.1.3
## [41] lifecycle_1.0.4    vctrs_0.6.5        evaluate_1.0.0     glue_1.7.0
## [45] farver_2.1.2       colorspace_2.1-1   fansi_1.0.6        rmarkdown_2.28
## [49] tools_4.4.1        pkgconfig_2.0.3    htmltools_0.5.8.1
```