# NYPD Shooting Data Analysis

Seth Porter

2022-04-04

## Contents

```r
library(conflicted)

# Tidyverse and sub-projects
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.0     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.1     v tibble    3.2.0
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
```

```r
# if this fails, please install with `install.packages("tidyverse")` in the console
library(lubridate)

# The new fancy error messages breaks LaTeX rendering at least on
# my machine, so avoid it.
conflict_prefer("filter", "dplyr")
```

```
## [conflicted] Will prefer dplyr::filter over any other package.
```

```
conflict_prefer("lag", "dplyr")
```

```
## [conflicted] Will prefer dplyr::lag over any other package.
```

# Overview

This analysis considers a historical dataset of shooting incidents in New York City, as reported by the NYPD.

Shooting incidents are extreme events which can end or radically change the lives of both the victim and the shooter in a single action. Understanding their patterns is crucial to policy and intervention decisions from law enforcement, other city agencies, and non-governmental actors.

## Where is the data from?

The dataset is sourced from https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic, with interpretation footnotes at https://bit.ly/3KSLRjA. It is described as a "List of every shooting incident that occurred in NYC going back to 2006 through the end of the previous calendar year."

It also clarifies that a shooting with no injuries is not included: "Only valid shooting incidents resulting in an injured victim are included in this release."

Be cautious; this data is from a law enforcement agency reporting on its own jurisdiction, and may have obvious as well as unexpected biases or gaps in the collection and reporting process.

## Secondary data sources

In addition to the primary data, some secondary sources are used to contextualize or suggest possible causal variables:

- Central Park temperature data from https://www.weather.gov/media/okx/Climate/CentralPark/DailyAvgTNormals.pdf
- Population data from https://www.census.gov/quickfacts/newyorkcitynewyork

## Self-links and Source

This document is derived from an R Markdown notebook, with source available in Github repo: https://github.com/symmatree/data-science/tree/main/r:

- This document: https://github.com/symmatree/data-science/raw/main/r/NYPDShootingData.Rmd (source) / pdf
- A slide deck summarizing the conclusions of this analysis, eliding discussion of the details of data cleaning and other "methods" topics: Source (https://github.com/symmatree/data-science/raw/main/r/NYPDShootingDataSlides.Rmd) / pdf / pptx

# Import

The first step in analysis is to import the data. In this case we read from https://catalog.data.gov/dataset/nypd-shooting-incident-data-historic:

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

```
## Rows: 25596 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.
```

# Tidy

Some fields need to be converted appropriately. Refer to https://data.cityofnewyork.us/api/views/833y-fsy8/files/e4e3d86c-348f-4a16-a17f-19480c089429?download=true&filename=NYPD_Shootings_Incident_Level_Data_Footnotes.pdf for background on the dataset.

## Dates

## Missing Values

> Null values may also appear in instances where information was not available or unknown at the time of the report and should be considered as either "Unknown/Not Available/Not Reported."

```
summary(is.na(raw_incidents))
```

```
##  INCIDENT_KEY    OCCUR_DATE      OCCUR_TIME         BORO
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:25596     FALSE:25596     FALSE:25596     FALSE:25596
##
##   PRECINCT       JURISDICTION_CODE LOCATION_DESC   STATISTICAL_MURDER_FLAG
##  Mode :logical   Mode :logical     Mode :logical   Mode :logical
##  FALSE:25596     FALSE:25594       FALSE:10619     FALSE:25596
##                  TRUE :2           TRUE :14977
##  PERP_AGE_GROUP   PERP_SEX       PERP_RACE       VIC_AGE_GROUP
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:16252     FALSE:16286     FALSE:16286     FALSE:25596
##  TRUE :9344      TRUE :9310      TRUE :9310
##   VIC_SEX          VIC_RACE       X_COORD_CD      Y_COORD_CD
##  Mode :logical   Mode :logical   Mode :logical   Mode :logical
##  FALSE:25596     FALSE:25596     FALSE:25596     FALSE:25596
##
##   Latitude       Longitude        Lon_Lat
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:25596     FALSE:25596     FALSE:25596
##
```

Several fields have substantial levels of missingness (TRUE counts in the table above): `LOCATION_DESC`, `PERP_{AGE_GROUP,SEX,RACE}`. In addition, `JURISDICTION_CODE` has two missing values.

The high missingness fields all have explicit values for "unknown":

- `LOCATION_DESC` has a `NONE` level
- `PERP_AGE` has `UNKNOWN`
- `PERP_SEX` has `U`
- `PERP_RACE` has `UNKNOWN`

so we can collapse the NA values onto these enums. `JURISDICTION_CODE` does **not** have this explicit encoding for missingness, but only two values are NA. These records appear unexceptional, and we will simply discard them.

(An alternative would be to infer the majority class, `Patrol`).

### Locations

There are several concerns about the geolocation fields (various police stations and prisons are used as placeholder values, among other things). For the moment, discard these fields and focus on the categorical data.

Exploring the geolocation of shootings, and correlating against the other variables, is an entire analysis of its own.

### Jurisdiction

Per the dataset description:

> `JURISDICTION_CODE`: Jurisdiction where the shooting incident occurred. Jurisdiction codes 0(Patrol), 1(Transit) and 2(Housing) represent NYPD whilst codes 3 and more represent non NYPD jurisdictions

We can recode the jurisdiction integers with human-readable values.

## PERP_AGE_GROUP bad values

PERP_AGE_GROUP has some bad values (1020, 224, and 940):

```r
raw_incidents %>% count(PERP_AGE_GROUP, sort=TRUE)
```

```
## # A tibble: 10 x 2
##    PERP_AGE_GROUP     n
##    <chr>          <int>
##  1 <NA>            9344
##  2 18-24           5844
##  3 25-44           5202
##  4 UNKNOWN         3148
##  5 <18             1463
##  6 45-64            535
##  7 65+               57
##  8 1020               1
##  9 224                1
## 10 940                1
```

## Categorical fields

Most of the string fields are actually categorical and should be converted to factors once their values are cleaned up:

- PERP and VIC versions of:
  - gender (‘SEX’)
  - RACE
  - AGE_GROUP
- BORO
- PRECINCT is categorical though it looks like a number (police precincts are numbered but are nominal not ordinal data, similar to zip codes). However, we will discard it from this analysis because it cannot be well analyzed without considering geolocation data, because the effect of the institution of the precinct itself is inherently confounded with the local environment.
- LOCATION_DESC is a category with 39 distinct values

## Execute tidying

```
gender_levels <- c("M", "F", "U")
# Use the data to generate these lists, the values are long and messy
race_levels <- levels(factor(raw_incidents$VIC_RACE))
# PERP_AGE_GROUP has some values that VIC_AGE_GROUP doesn't, but they're bad.
age_levels = levels(factor(raw_incidents$VIC_AGE_GROUP))
boro_levels <- c( "BRONX", "BROOKLYN", "MANHATTAN", "QUEENS", "STATEN ISLAND")
# use parse_factor to warn if we get unexpected values
incidents <- raw_incidents %>%
    filter(!is.na(JURISDICTION_CODE)) %>%
    mutate(OCCUR_DATE=mdy(OCCUR_DATE),
           LOCATION_DESC=replace_na(LOCATION_DESC, "NONE"),
           PERP_RACE=replace_na(PERP_RACE, "UNKNOWN"),
           PERP_AGE_GROUP=replace_na(PERP_AGE_GROUP, "UNKNOWN"),
           PERP_SEX=replace_na(PERP_SEX, "U")) %>%
    mutate(PERP_AGE_GROUP=recode(PERP_AGE_GROUP,
           "1020"= "UNKNOWN",
           "224" = "UNKNOWN",
           "940" = "UNKNOWN")) %>%
    mutate(PERP_SEX=parse_factor(PERP_SEX, levels=gender_levels),
           VIC_SEX=parse_factor(VIC_SEX, levels=gender_levels),
           PERP_RACE=parse_factor(PERP_RACE, levels=race_levels),
           VIC_RACE=parse_factor(VIC_RACE, levels=race_levels),
           BORO=parse_factor(BORO, levels=boro_levels),
           # PRECINCT=factor(PRECINCT),
           LOCATION_DESC=factor(LOCATION_DESC),
           JURISDICTION_CODE = fct_recode(factor(JURISDICTION_CODE),
                                          "Patrol" = "0",
                                          "Transit" = "1",
                                          "Housing" = "2"),
           PERP_AGE_GROUP=parse_factor(PERP_AGE_GROUP, age_levels),
           VIC_AGE_GROUP=parse_factor(VIC_AGE_GROUP, age_levels)) %>%
    select(-c("Lon_Lat", "INCIDENT_KEY", "X_COORD_CD", "Y_COORD_CD", "Latitude", "Longitude", "PRECINCT
```

# Explore

Rather than explore the data one *analysis* at a time, we will instead focus on one or more *columns* at a time and explore each in appropriate ways. However, an initial summary of the data is a useful starting point:

```
summary(incidents)
```

```
##    OCCUR_DATE          OCCUR_TIME                  BORO       JURISDICTION_CODE
##  Min.   :2006-01-01   Length:25594      BRONX        : 7402   Patrol :21321
##  1st Qu.:2009-05-10   Class1:hms        BROOKLYN     :10365   Transit:   59
##  Median :2012-08-26   Class2:difftime   MANHATTAN    : 3264   Housing: 4214
##  Mean   :2013-06-13   Mode  :numeric    QUEENS       : 3827
##  3rd Qu.:2017-06-30                     STATEN ISLAND:  736
##  Max.   :2021-12-31
##
##                    LOCATION_DESC   STATISTICAL_MURDER_FLAG PERP_AGE_GROUP
##  NONE                     :15151   Mode :logical           <18    : 1462
##  MULTI DWELL - PUBLIC HOUS: 4559   FALSE:20666             18-24  : 5844
##  MULTI DWELL - APT BUILD  : 2664   TRUE :4928              25-44  : 5202
##  PVT HOUSE                :  893                           45-64  :  535
##  GROCERY/BODEGA           :  622                           65+    :   57
##  BAR/NIGHT CLUB           :  588                           UNKNOWN:12494
##  (Other)                  : 1117
##  PERP_SEX                           PERP_RACE     VIC_AGE_GROUP   VIC_SEX
##  M:14414   AMERICAN INDIAN/ALASKAN NATIVE:    2   <18    : 2681   M:23180
##  F:  371   ASIAN / PACIFIC ISLANDER      :  141   18-24  : 9603   F: 2403
##  U:10809   BLACK                         :10667   25-44  :11385   U:   11
##            BLACK HISPANIC                : 1203   45-64  : 1698
##            UNKNOWN                       :11146   65+    :  167
##            WHITE                         :  272   UNKNOWN:   60
##            WHITE HISPANIC                : 2163
##                            VIC_RACE
##  AMERICAN INDIAN/ALASKAN NATIVE:    9
##  ASIAN / PACIFIC ISLANDER      :  354
##  BLACK                         :18280
##  BLACK HISPANIC                : 2485
##  UNKNOWN                       :   65
##  WHITE                         :  660
##  WHITE HISPANIC                : 3741
```

## Date of incident: `OCCUR_DATE`

### Trends over Time: 2020 was BAD

Consider annual incident totals, over time:

```
yearly <- incidents %>% count(year=floor_date(OCCUR_DATE, "year"),
                              year_num=year(floor_date(OCCUR_DATE, "year")))
print(yearly)
```

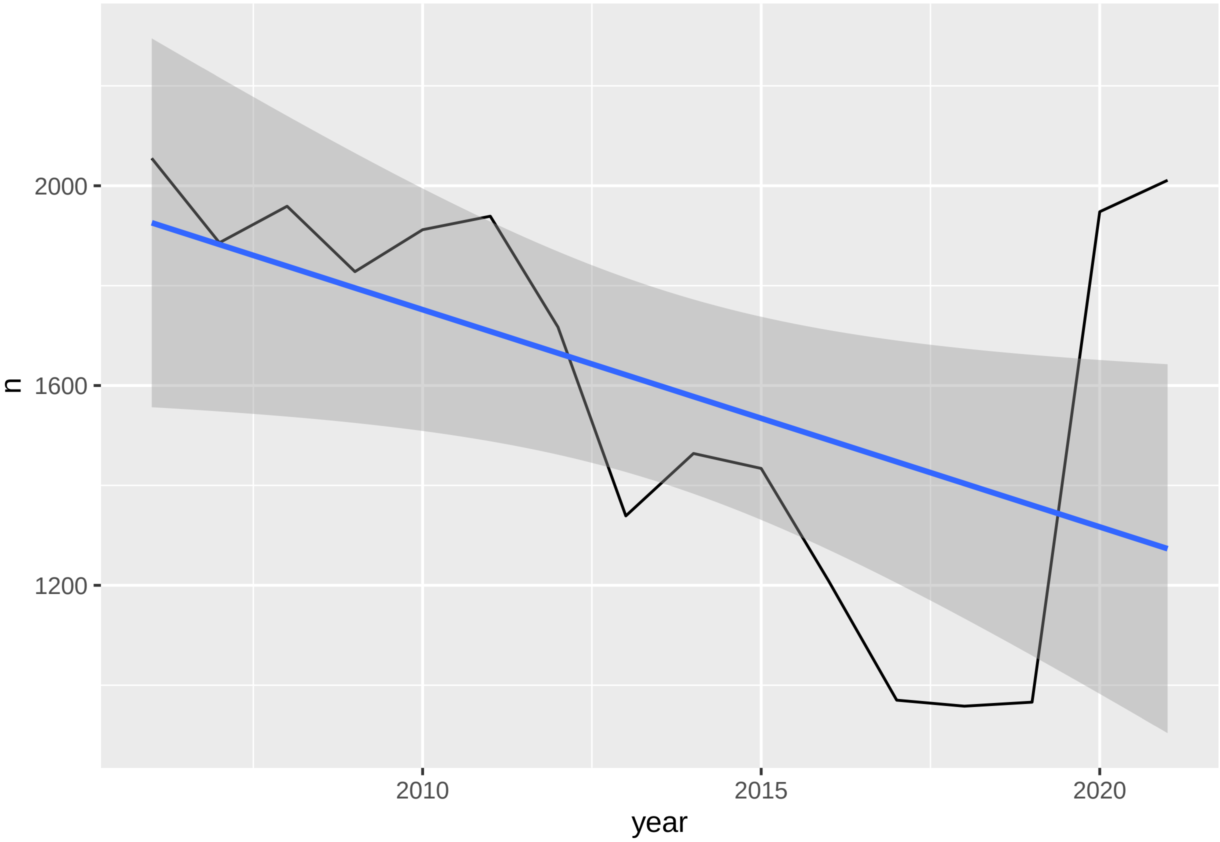
```
## # A tibble: 16 x 3
##    year       year_num     n
##    <date>        <dbl> <int>
##  1 2006-01-01     2006  2055
##  2 2007-01-01     2007  1886
##  3 2008-01-01     2008  1959
##  4 2009-01-01     2009  1828
##  5 2010-01-01     2010  1912
##  6 2011-01-01     2011  1939
##  7 2012-01-01     2012  1717
##  8 2013-01-01     2013  1339
##  9 2014-01-01     2014  1464
## 10 2015-01-01     2015  1434
## 11 2016-01-01     2016  1208
## 12 2017-01-01     2017   970
## 13 2018-01-01     2018   958
## 14 2019-01-01     2019   966
## 15 2020-01-01     2020  1948
## 16 2021-01-01     2021  2011
```

```
yearly_fit <- lm(formula = yearly$n ~ yearly$year)
print(summary(yearly_fit))
```

```
##
## Call:
## lm(formula = yearly$n ~ yearly$year)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -477.33 -282.48   18.13  136.88  737.68
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3491.91225  855.60216   4.081  0.00112 **
## yearly$year   -0.11910    0.05355  -2.224  0.04311 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 360.7 on 14 degrees of freedom
## Multiple R-squared:  0.2611, Adjusted R-squared:  0.2083
## F-statistic: 4.946 on 1 and 14 DF,  p-value: 0.04311
```

```
yearly %>%
  ggplot(aes(year, n)) +
    geom_line() +
    geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



2020 was clearly a *wild* outlier. There was a strong linear trend from 2006 through 2019, completely reversed in 2020. Without further investigation, we cannot fully assume the cause, but "in some way caused by the pandemic" seems like a reasonable starting point.

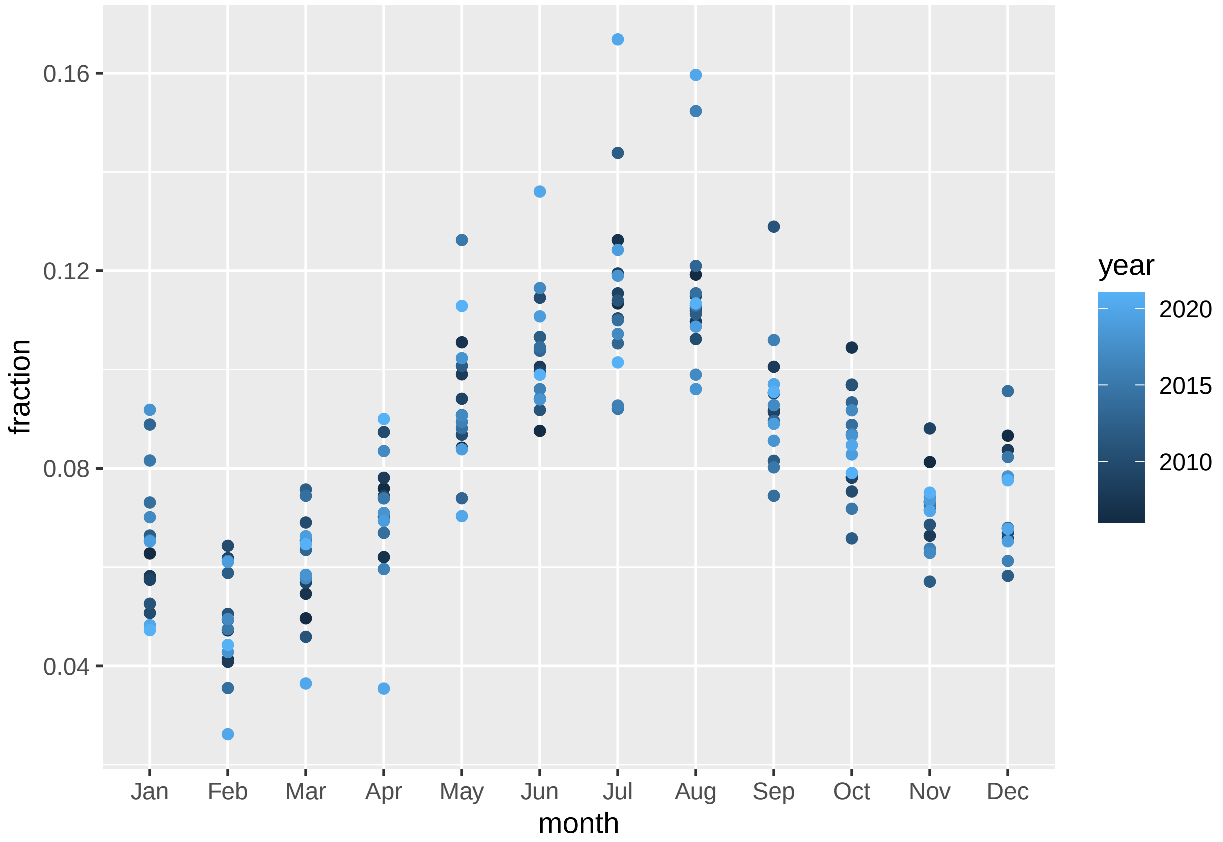
Note that this is such an outlier that we cannot usefully detrend the data to remove the steady year-over-year drop, so instead we must simply remember that the overall volume of incidents roughly halved between 2006 and 2019, then rebounded to 2006 levels in 2020. We can also consider proportional data normalized by each year's totals.

Rather than consider all temporal trends at once, we will look at the trends within each category in turn.

### Month of Year

A different aspect of the date is any cycle within the year. We can plot which fraction of incidents occur in a given month (e.g. "February") and observe what appears to be a sinusoidal pattern.

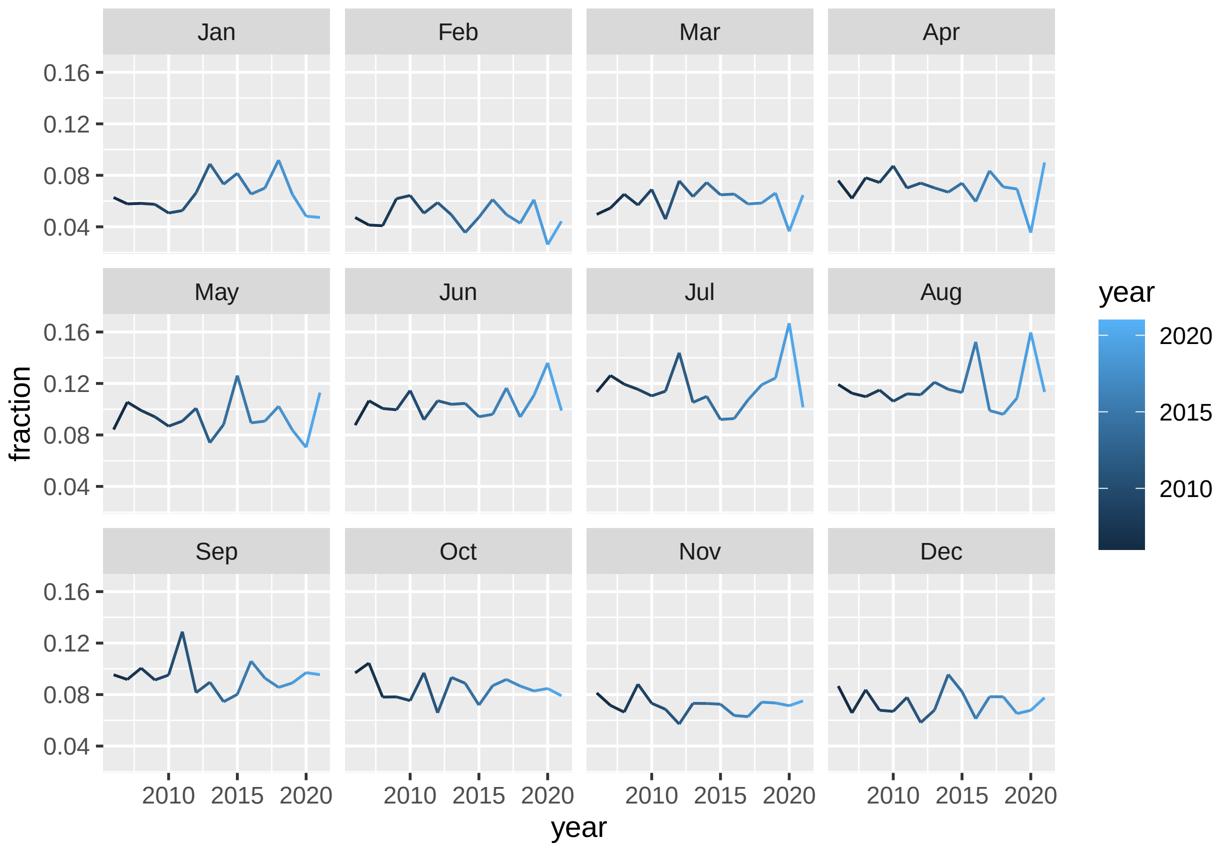
```
monthly = incidents %>%
  count(month=month(floor_date(OCCUR_DATE, "month"), label=TRUE),
        month_num=month(floor_date(OCCUR_DATE, "month")),
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n/n.total)

# The overall trend masks the seasonality in the raw plot:
#monthly %>%
#  ggplot(aes(month, n, color=year)) +
#    geom_point()

monthly %>%
  ggplot(aes(month, fraction, color=year)) +
    geom_point()
```



#### Trend over time

The overlaid plot, above, shows an aggregate pattern but might hide an evolution over time. Plotting the fraction of annual crime in each month, longitudinally over years, would reveal this:

```
monthly %>%
  ggplot(aes(x=year, y=fraction, color=year)) +
    geom_line() +
    facet_wrap(~month)
```



The only immediately visible trend is a shift in 2020 from January through April, toward June, July and August. This could be tentatively read as a covid-based signal partially overriding the established annual cycle.

**Sinusoidal Model** Ignoring for the moment the pandemic shift, We can fit a scaled and offset sine wave to this data:

```
sine_model <- nls(fraction ~ v_scale*sin(2*pi*month_num/12.0+phase)+offset, data=monthly, start=list(v_
print(sine_model)
```

```
## Nonlinear regression model
##   model: fraction ~ v_scale * sin(2 * pi * month_num/12 + phase) + offset
##    data: monthly
## v_scale   phase  offset
## 0.02813 3.93814 0.08333
##  residual sum-of-squares: 0.0369
##
## Number of iterations to convergence: 4
## Achieved convergence tolerance: 4.517e-08
```

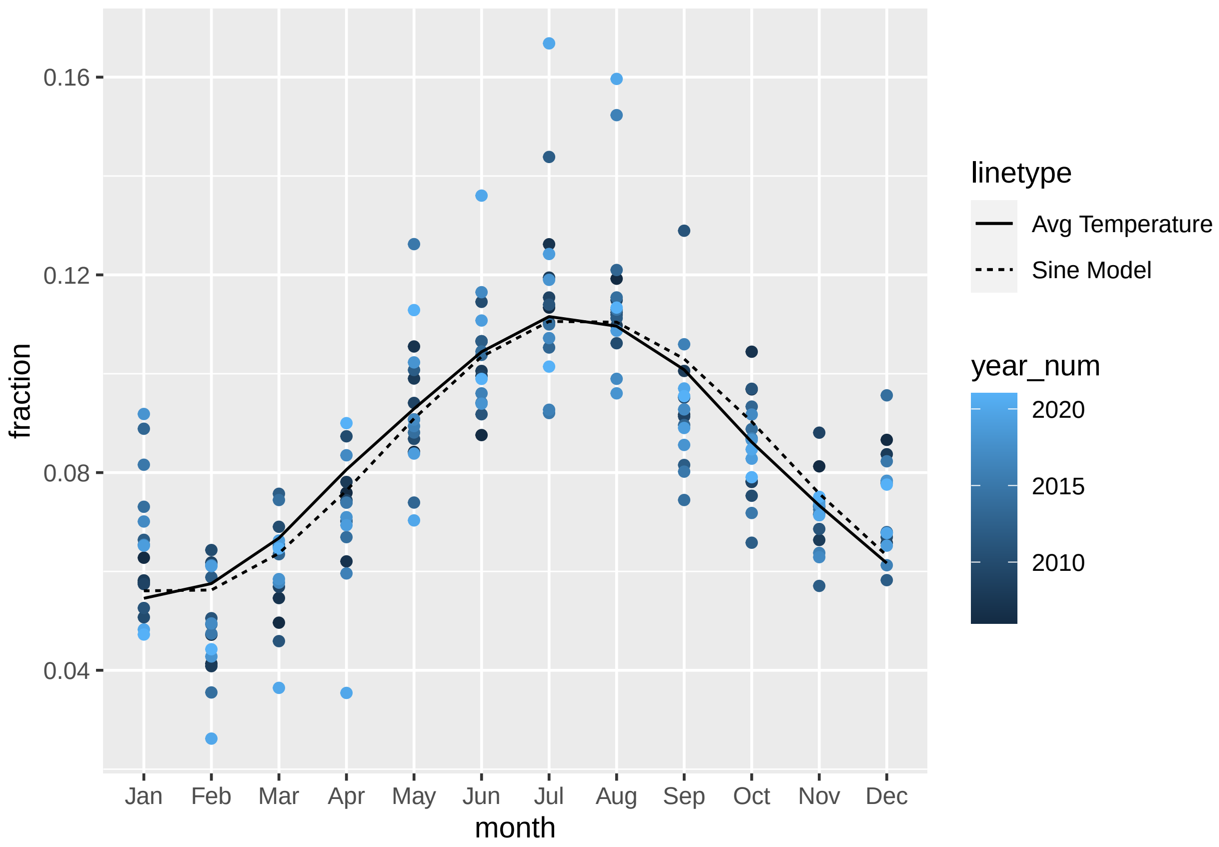
```
monthly_with_sine_pred <- monthly %>%
  mutate(predicted=predict(sine_model))

sine_rmse = sqrt(mean((monthly_with_sine_pred$fraction - monthly_with_sine_pred$predicted)^2))
print(c("RMSE of model: ", sine_rmse))
```

```
## [1] "RMSE of model: "    "0.0138635971179457"
```

The root-mean-squared-error (RMSE) of ~0.014 can be interpreted in the units (fraction of the yearly incident volume), so this model predicts the actual monthly fraction of crime with an error of 1.4 percentage points, which is rather small compared to the magnitude of the values (largely between 5 and 10 percentage points); this is a strong result for such a simple model.

Plotting the model we can confirm this fit visually. We also overlay a plot of the monthly average temperature in Central Park, per https://www.weather.gov/media/okx/Climate/CentralPark/DailyAvgTNormals.pdf, which, while not establishing a causal relationship, is certainly suggestive.

```r
month_df = tibble(month_num=seq(1, 12))
month_df$predicted=predict(sine_model, newdata=month_df)

# Per https://www.weather.gov/media/okx/Climate/CentralPark/DailyAvgTNormals.pdf
nyc_average_temps = c(33.6, 35.9, 43.0, 53.7, 63.2, 72.1, 77.6, 76.1, 69.3, 58, 48.1, 39.1)
month_df$average_temp <- nyc_average_temps

# Let the computer figure out the right scale and offset to align the temp
# data with the sine model outputs.
temp_model <- nls(predicted ~ temp_scale*average_temp+temp_offset,
                  data=month_df,
                  start=list(temp_scale=1, temp_offset=0))
month_df$scaled_temp <- predict(temp_model)


ggplot() +
    geom_point(data=monthly_with_sine_pred, mapping=aes(x=month, y=fraction, color=year_num)) +
    geom_line(data=month_df, mapping=aes(x=month_num, y=predicted, linetype="Sine Model")) +
    geom_line(data=month_df, mapping=aes(x=month_num, y=scaled_temp, linetype="Avg Temperature"))
```



We can assert that there is a strong annual cycle that should be investigated, and a very interesting correlation with temperature, which suggests interpretations around how / where people spend their time in different seasons.

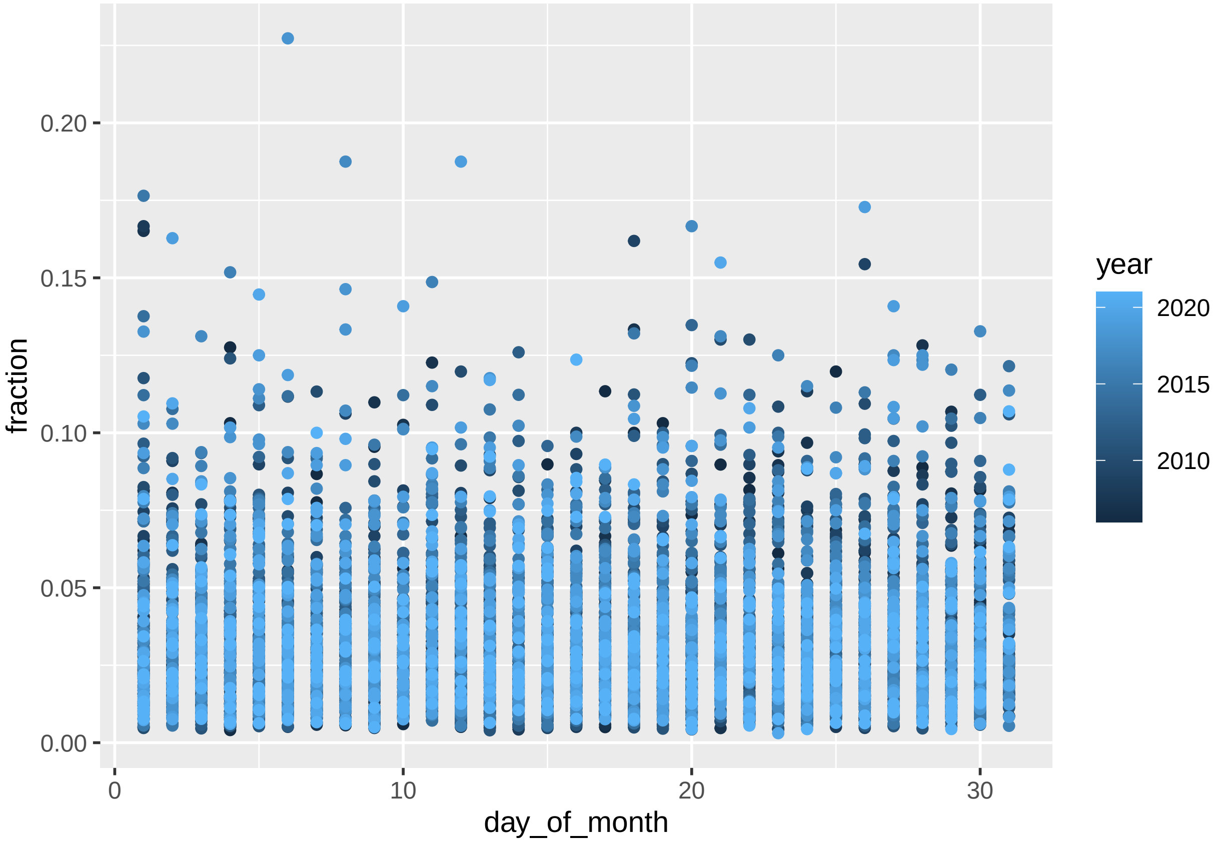
Note that the 2020 outliers are clearly visible in this plot, lower after the pandemic response began in February, and higher in the summer.

This concludes our look at the annual cycles in the data.

### Day of month

A similar analysis by day-of-month shows no noticeable pattern on first examination; there may be subtle structure here but it would not be the most promising topic to investigate first.

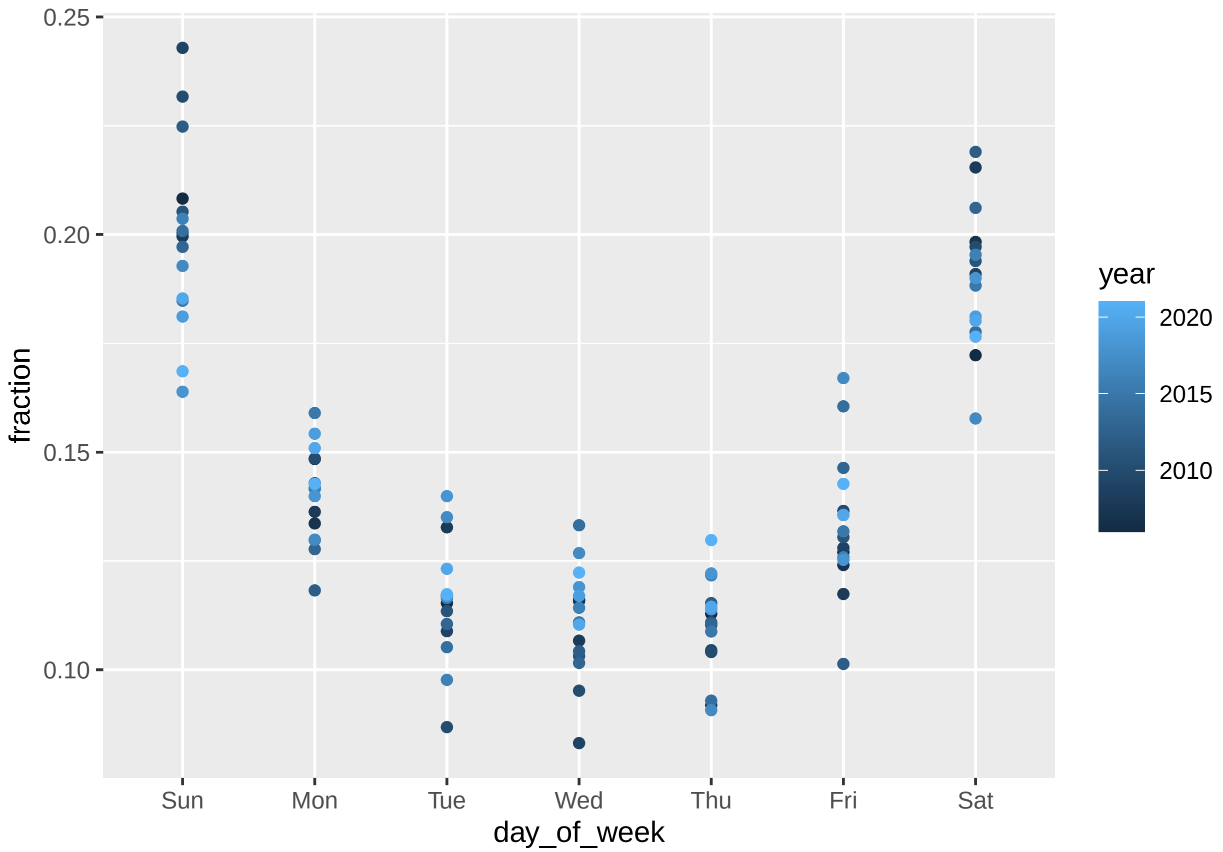
```
day_of_month = incidents %>%
  count(day_of_month=mday(OCCUR_DATE),
        month=floor_date(OCCUR_DATE, "month"),
        month_num=month(floor_date(OCCUR_DATE, "month")),
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(monthly, by=c("month_num","year"), suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given month
  mutate(fraction=n/n.total)

day_of_month %>%
  ggplot(aes(day_of_month, fraction, color=year, group=month)) +
    geom_point()
```



### Day of Week

We can plot what fraction of annual incidents occur on each day of the week, with one point per year, to look for overall trends:

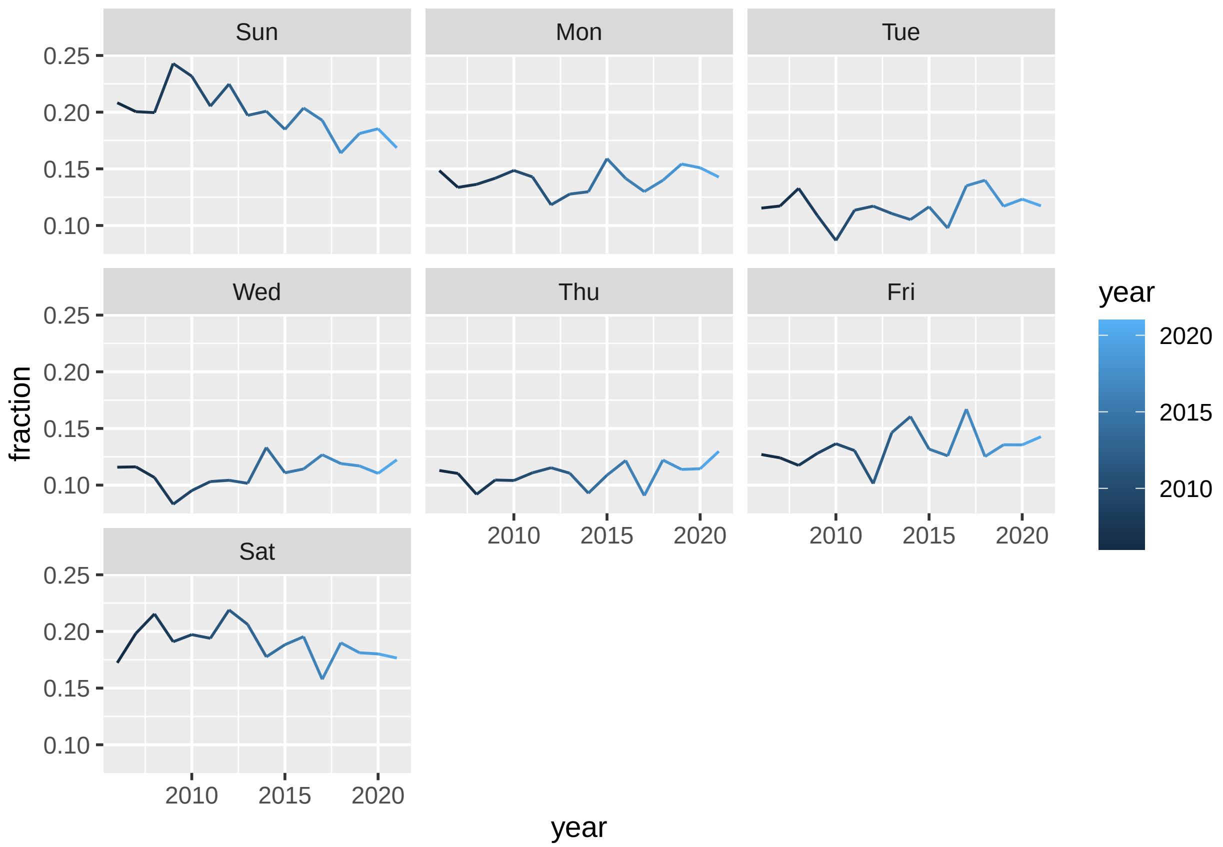
```
day_of_week = incidents %>%
  count(day_of_week=wday(OCCUR_DATE, label=TRUE),
        day_num=wday(OCCUR_DATE),
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given year
  mutate(fraction=n/n.total)

day_of_week %>%
  ggplot(aes(day_of_week, fraction, color=year)) +
    geom_point()
```



We can see that there is a strong weekly cycle to this data, again looking roughly sinusoidal.

**Trends over time** Plotting the per-day trends over time:

```
day_of_week %>%
  ggplot(aes(x=year, y=fraction, color=year)) +
    geom_line() +
    facet_wrap(~day_of_week)
```



we can see that the pattern is relatively stable over time, unless one sees significance in the possible downward trend in Sunday shootings. (Unlike the monthly data, there is no major deviation from the pattern in 2020.)

```
sine_day_model <- nls(fraction ~ v_scale*sin(2*pi*day_num/7.0+phase)+offset, data=day_of_week, start=li
print(sine_day_model)
```

**Sinusoidal model**

```
## Nonlinear regression model
##   model: fraction ~ v_scale * sin(2 * pi * day_num/7 + phase) + offset
##    data: day_of_week
## v_scale   phase  offset
## 0.04592 1.02457 0.14286
##  residual sum-of-squares: 0.03819
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 1.715e-08
```

```
day_of_week_with_sine_pred <- day_of_week %>%
  mutate(predicted=predict(sine_day_model))

sine_rmse = sqrt(mean((day_of_week_with_sine_pred$fraction - day_of_week_with_sine_pred$predicted)^2))
print(c("RMSE of model: ", sine_rmse))
```

```
## [1] "RMSE of model: "   "0.018466149058027"
```

Much as with the monthly data, the fraction of annual incidents on a given day of the week are predicted quite well with this model, as seen by the RMSE of ~2 percentage points for data largely between 12 and 22 percentage points.

### Joint Model

Given the success of these two models in explaining a great deal of the variation at the weekly and monthly level, it seems reasonable to consider a joint model to predict daily totals based on two composed sine waves, one computed over day-of-week and one over month-of-year.

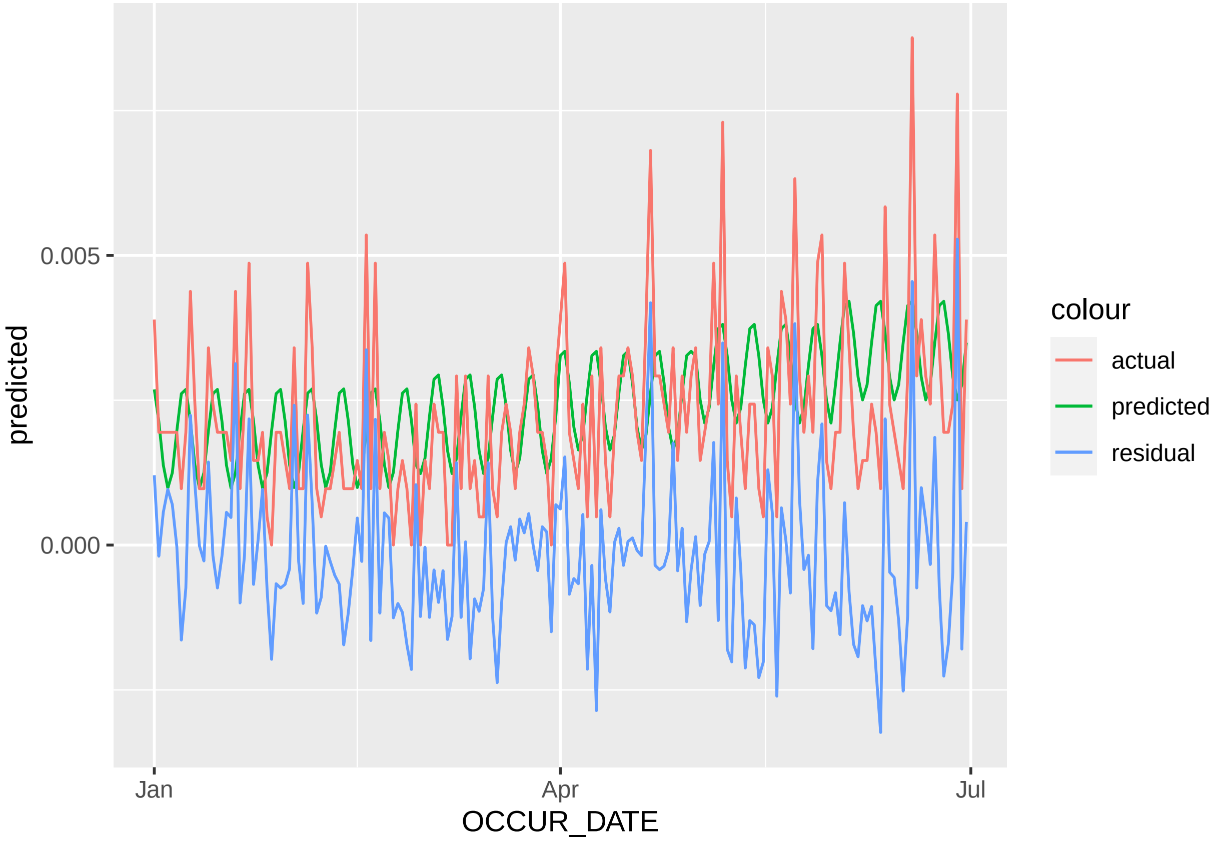
```
# Generate
all_days <-
  tibble(OCCUR_DATE=seq(from=min(incidents$OCCUR_DATE), to=max(incidents$OCCUR_DATE), by=1)) %>%
  mutate(
    day_num=wday(OCCUR_DATE),
    month_num=month(floor_date(OCCUR_DATE, "month")),
    year=floor_date(OCCUR_DATE, "year"))

daily_counts <- incidents %>%
  count(OCCUR_DATE=OCCUR_DATE) %>%
  right_join(all_days, by="OCCUR_DATE") %>%
  mutate(n=replace_na(n, 0)) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given year
  mutate(fraction=n/n.total)

sine_joint_model <- nls(
  fraction ~ week_scale*sin(2*pi*day_num/7.0+week_phase)+month_scale*sin(2*pi*month_num/12.0+month_phase
  data=daily_counts, start=list(week_scale=1, week_phase=0, month_scale=1, month_phase=0, offset=0.1))
print(sine_joint_model)
```

```
## Nonlinear regression model
##   model: fraction ~ week_scale * sin(2 * pi * day_num/7 + week_phase) +     month_scale * sin(2 * pi
##    data: daily_counts
##  week_scale  week_phase month_scale month_phase      offset
##   0.0008806   1.0250767  -0.0008994   0.8046291   0.0027337
##  residual sum-of-squares: 0.02356
##
## Number of iterations to convergence: 7
## Achieved convergence tolerance: 2.86e-09
```

```
daily_counts_with_sine_pred <- daily_counts %>%
  mutate(predicted=predict(sine_joint_model))

sine_joint_rmse = sqrt(
  mean(
    (daily_counts_with_sine_pred$fraction - daily_counts_with_sine_pred$predicted)^2))
print(c("RMSE of joint model: ", sine_joint_rmse))
```

```
## [1] "RMSE of joint model: " "0.00200796051403927"
```

```
#hist(daily_counts$fraction)
daily_counts_with_sine_pred$resid = daily_counts_with_sine_pred$fraction - daily_counts_with_sine_pred$p
daily_counts_with_sine_pred %>%
  # Plot becomes unreadable with too many observations
  filter(OCCUR_DATE < ymd("2006-07-01")) %>%
  ggplot() +
    geom_line(aes(x=OCCUR_DATE, y=predicted, color="predicted")) +
    geom_line(aes(x=OCCUR_DATE, y=fraction, color="actual")) +
    geom_line(aes(x=OCCUR_DATE, y=resid, color="residual"))
```



This model is rather underwhelming numerically - the RMSE is as of the same order as the signal - and visually there is no obvious explanation. The green (prediction) line is not unreasonable compared to the red actuals, but there is simply a great deal of day-to-day variation in the actuals which are not predicted by this model.

This concludes our analysis of specifically temporal patterns, though we will consider the trend of each variable as we come to it.

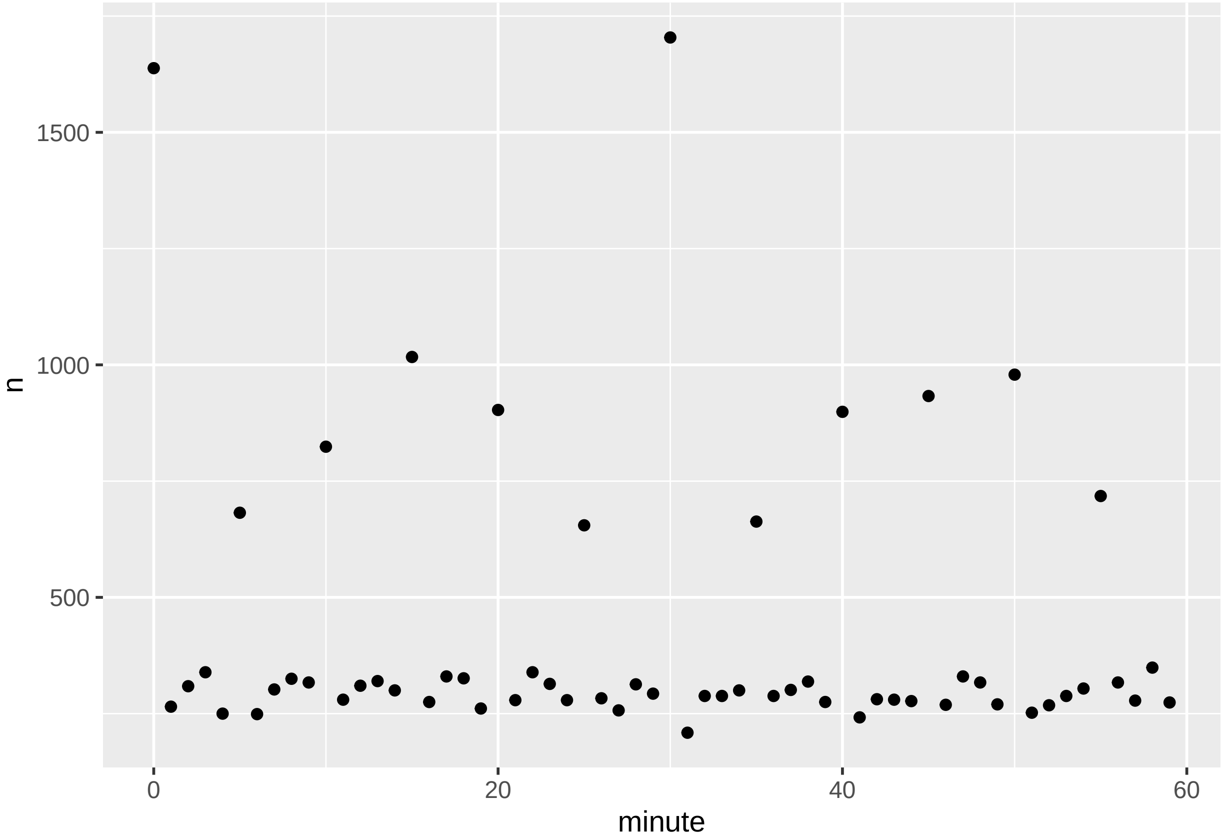
### Time of day: `OCCUR_TIME`

We consider now the time of day when the incident is recorded.

### Minute of hour

It seems unlikely that shootings happen more often at the beginning of an hour than the end, say, but let us consider:

```
incidents %>%
  count(minute=minute(OCCUR_TIME)) %>%
  ggplot(aes(x=minute, y=n)) +
    geom_point()
```



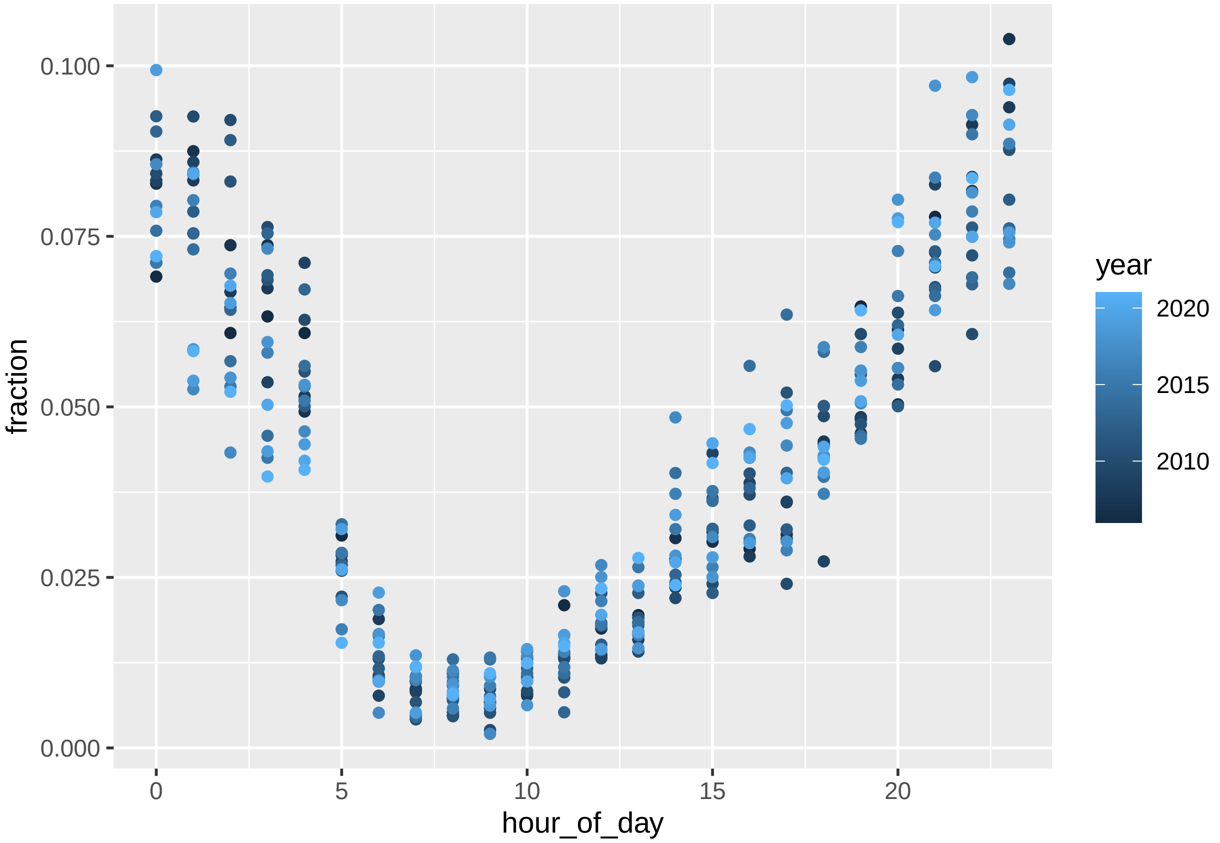
```
#incidents %>%
#   count(minute=minute(OCCUR_TIME))
```

The two highest peaks are at 0 and 30 minutes, with lower peaks at five minute intervals; counts are much lower between this values. This almost certainly demonstrates rounding in the data collection, *not* an actual semantic pattern to the time of incidents. Were it actually the case that "murders happen on the half hour", we would expect to see high counts for 29 and 31 minutes as well, due to differing clocks and the difficulty of scheduling a shooting. In this data, however, the *lowest* points occur on each side of a spike, which strongly suggests that adjacent values are being rounded to the spike value.

### Hour of day

The simplest way to avoid being misled by these rounded values is to discard them. (Alternatively, we could bucket at 30 minute intervals, since that is roughly the granularity with which the data is **actually** collected.) We can then plot, for each year, the portion of incidents occuring in each hour of the day:

```
hour_of_day = incidents %>%
  count(hour_of_day=hour(OCCUR_TIME),
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given year
  mutate(fraction=n/n.total)

hour_of_day %>%
  ggplot(aes(hour_of_day, fraction, color=year)) +
    geom_point()
```

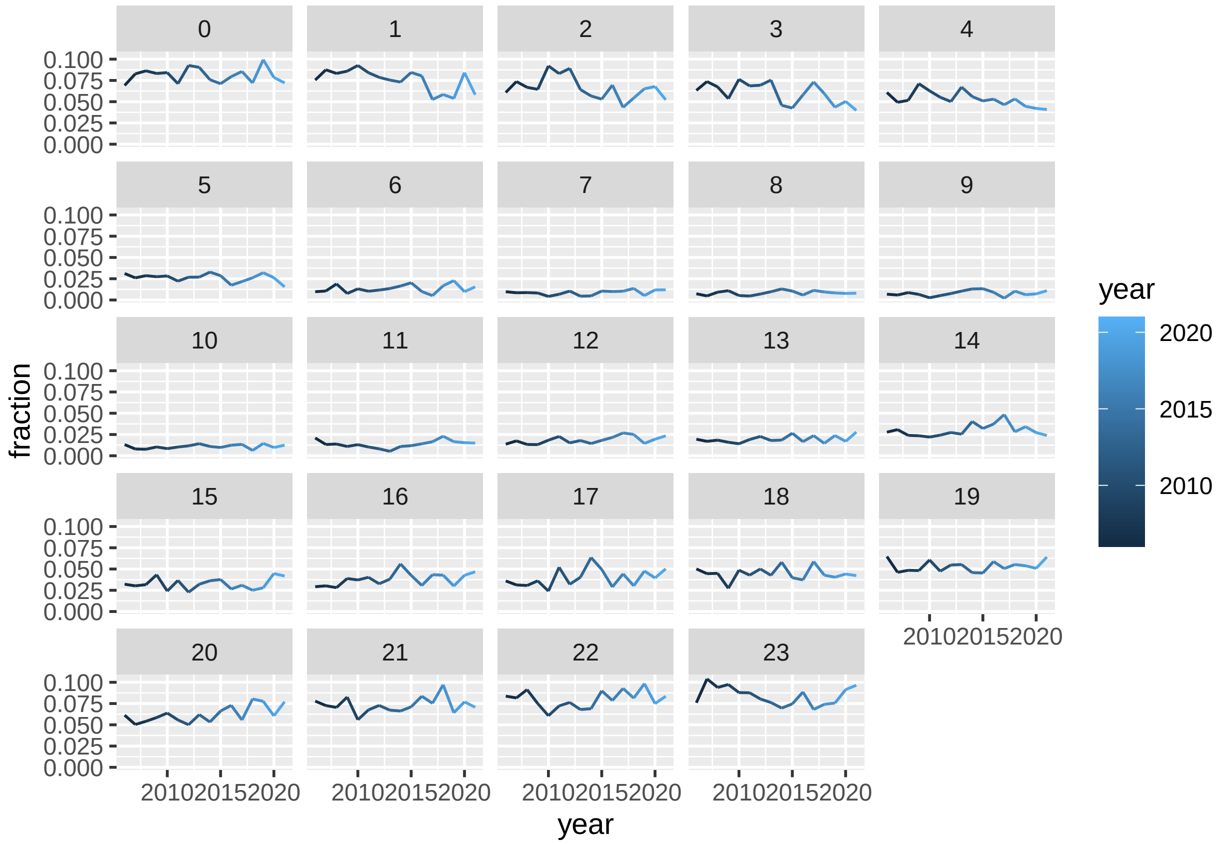


This is not nicely sinusoidal like the weekly and monthly cycles, but there is a clear pattern here; incidents occur in the afternoon and especially overnight, but rarely in the dawn and waking hours.

It is worth noting that bars in New York City close at 4am; this may help explain the timing of the downward discontinuity in the following hour.

Considering the trends over time for each hour:

```
hour_of_day %>%
  ggplot(aes(x=year, y=fraction, color=year)) +
    geom_line() +
    facet_wrap(~hour_of_day)
```

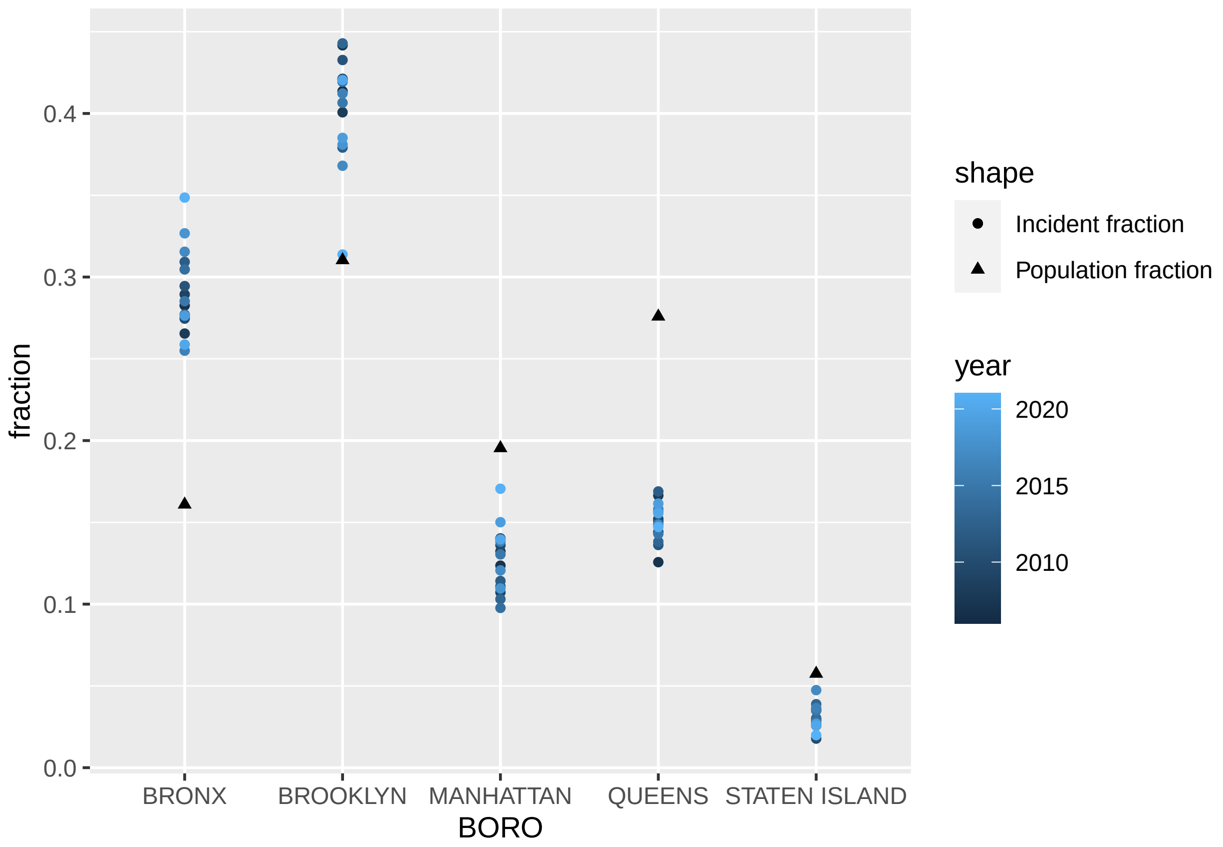


we see no glaring shifts occurring over time.

## Borough

We can plot borough population data, from https://en.wikipedia.org/wiki/Boroughs_of_New_York_City, against the fraction of incidents in each borough:

```r
# rounded to 0.1 million, from <https://en.wikipedia.org/wiki/Boroughs_of_New_York_City>
boro_pops = tibble(BORO=levels(incidents$BORO),
                   pop=c(1.4,2.7,1.7,2.4,0.5)) %>%
  mutate(pop_frac=pop/sum(pop))

by_boro = incidents %>%
  count(BORO=BORO,
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given year
  mutate(fraction=n/n.total)

ggplot() +
  geom_point(data=by_boro, aes(x=BORO, y=fraction, color=year, shape="Incident fraction")) +
  geom_point(data=boro_pops, aes(x=BORO, y=pop_frac, shape="Population fraction"))
```



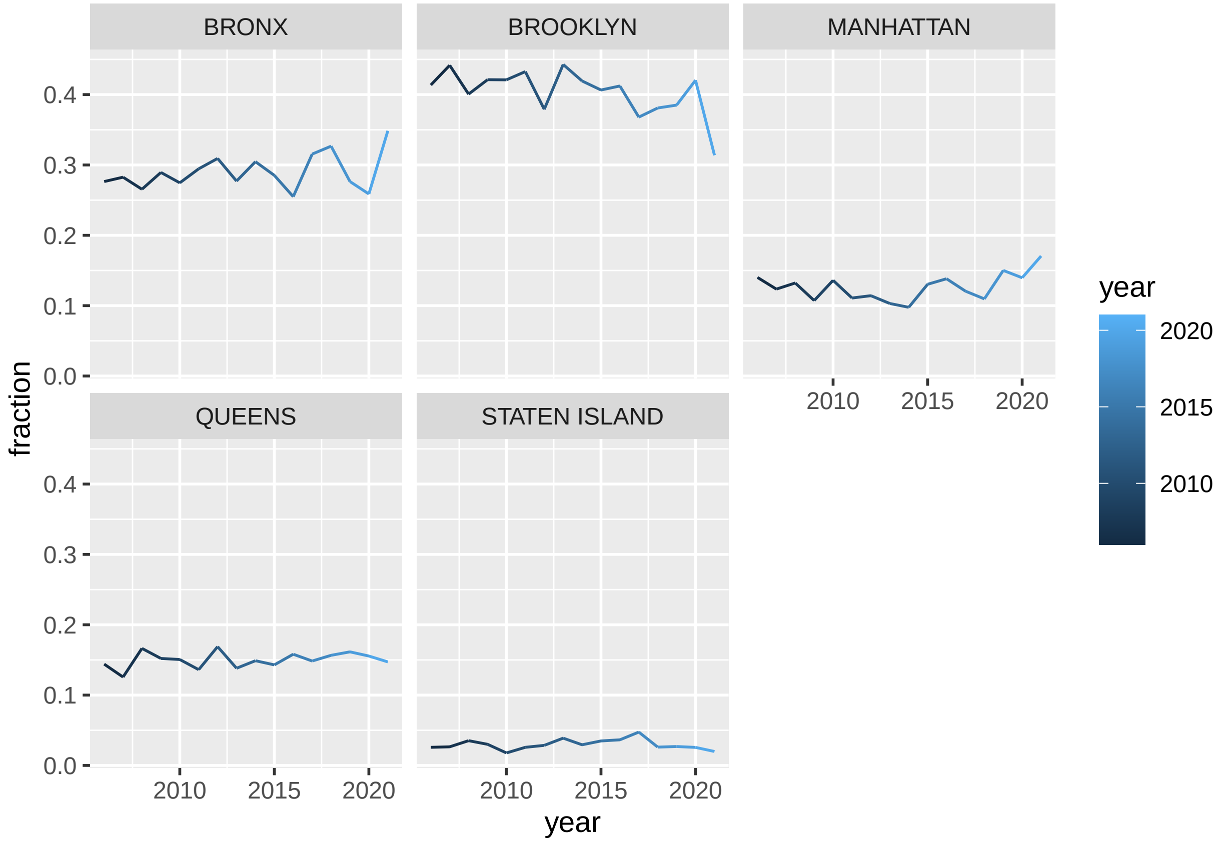
We see that the Bronx and Brooklyn have a greater proportional share of incidents than of population, while Manhattan and Queens have a smaller share of incidents than residents, and Staten Island has roughly “its fair share”.

Understanding the causes of these differences is beyond this paper, but some possible factors include:

- Residential population is not instantaneous population; incidents do not necessarily happen in people’s homes.
- Further, the strong time-of-day cycle in incidents may factor in; most incidents happen in the evening hours, so if people are more likely to visit a particular borough at night, it might have a sharply higher population *during those hours*.
- There are many differences between boroughs beyond simple population; many socioeconomic, demographic, and other factors might influence the rate of incidents.

Plotting the per-borough split over time, the pattern has been quite stable:

```
by_boro %>%
  ggplot(aes(year, fraction, color=year)) +
    geom_line() +
    facet_wrap(~BORO)
```

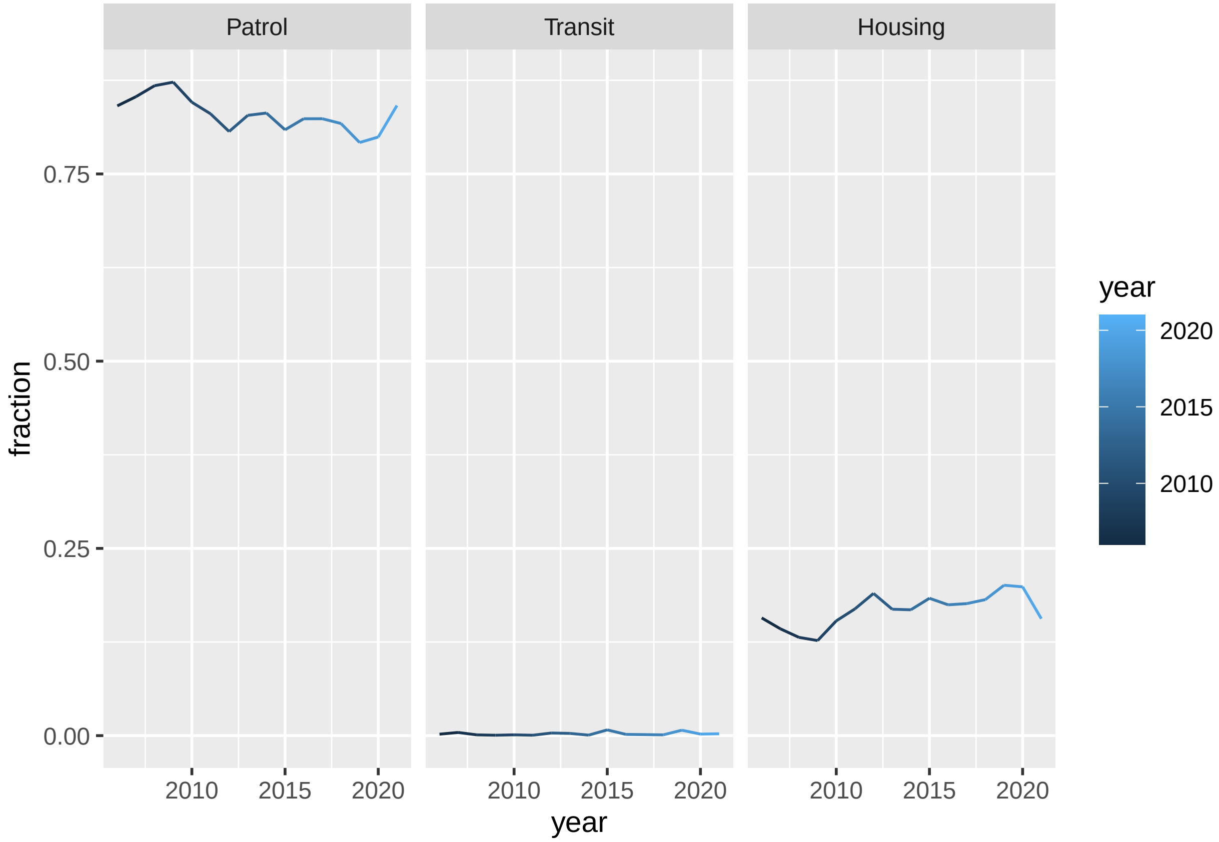


## Jurisdiction

Several police jurisdictions are represented in the dataset. These represent first: differing *institutions*, possibly reporting or acting in different ways, and secondly: different *locations* where incidents occur, whether in public housing, mass transit, or elsewhere.

The proportion of incidents in the different juridictions has been stable over time:

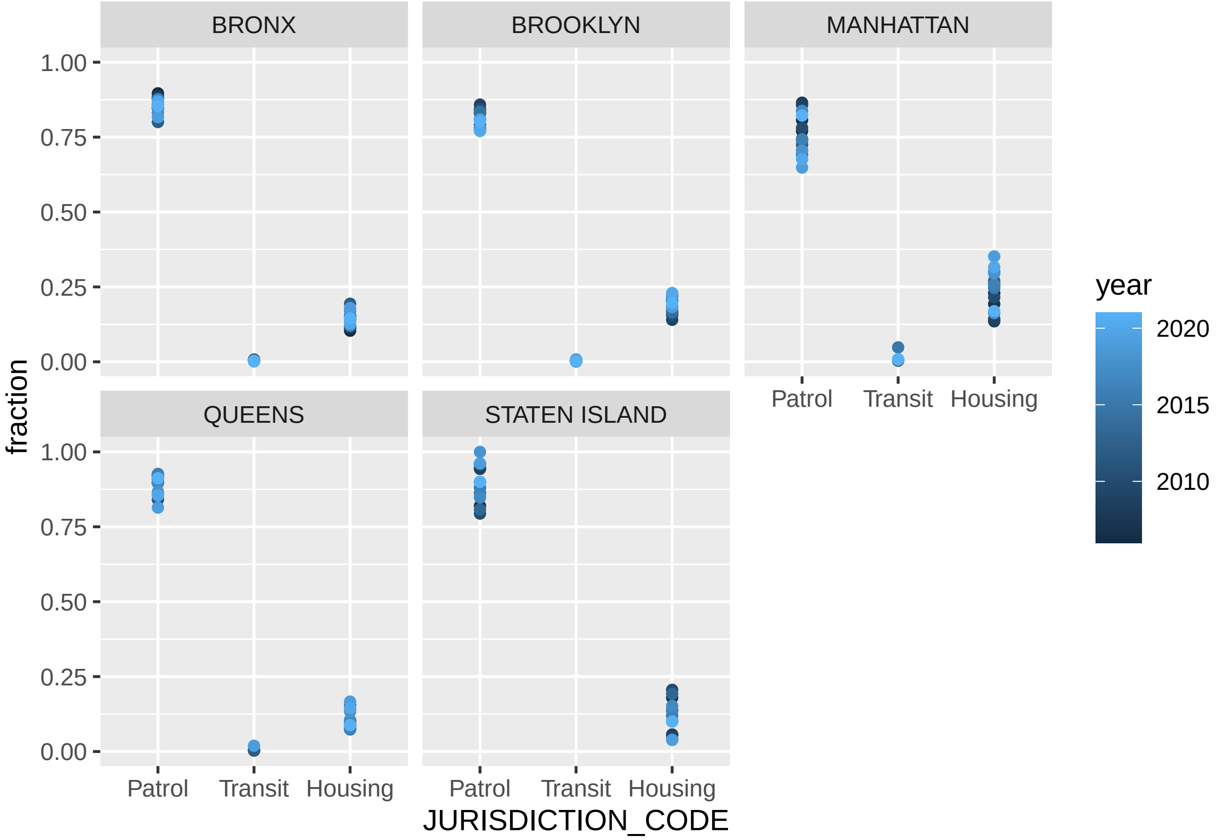
```
by_juris = incidents %>%
  count(JURISDICTION_CODE=JURISDICTION_CODE,
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given year
  mutate(fraction=n/n.total)

by_juris %>%
ggplot() +
  geom_line(aes(x=year, y=fraction, color=year)) +
  facet_wrap(~JURISDICTION_CODE)
```



We can demonstrate at a coarse level how Jurisdiction is confounded with location by considering the fraction of annual per-borough incidents which fall into each jurisdiction:

```
# This version scaled by the total annual incidents, which meant that
# boroughs with more total incidents appeared to have different
# transit / housing / patrol ratios.
#
#by_juris_boro = incidents %>%
#  count(JURISDICTION_CODE=JURISDICTION_CODE,
#        BORO=BORO,
#        year=floor_date(OCCUR_DATE, "year")) %>%
#  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given year
#  mutate(fraction=n/n.total)



by_juris_boro = incidents %>%
  count(JURISDICTION_CODE=JURISDICTION_CODE,
        BORO=BORO,
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(incidents %>%
              count(year=floor_date(OCCUR_DATE, "year"),
                    BORO=BORO),
            by=c("year", "BORO"), suffix=c("", ".total")) %>%
  # n.total is the total incidents in that borough in the given year
  mutate(fraction=n/n.total)
```

```
by_juris_boro %>%
ggplot() +
  geom_point(aes(x=JURISDICTION_CODE, y=fraction, color=year)) +
  facet_wrap(~BORO)
```



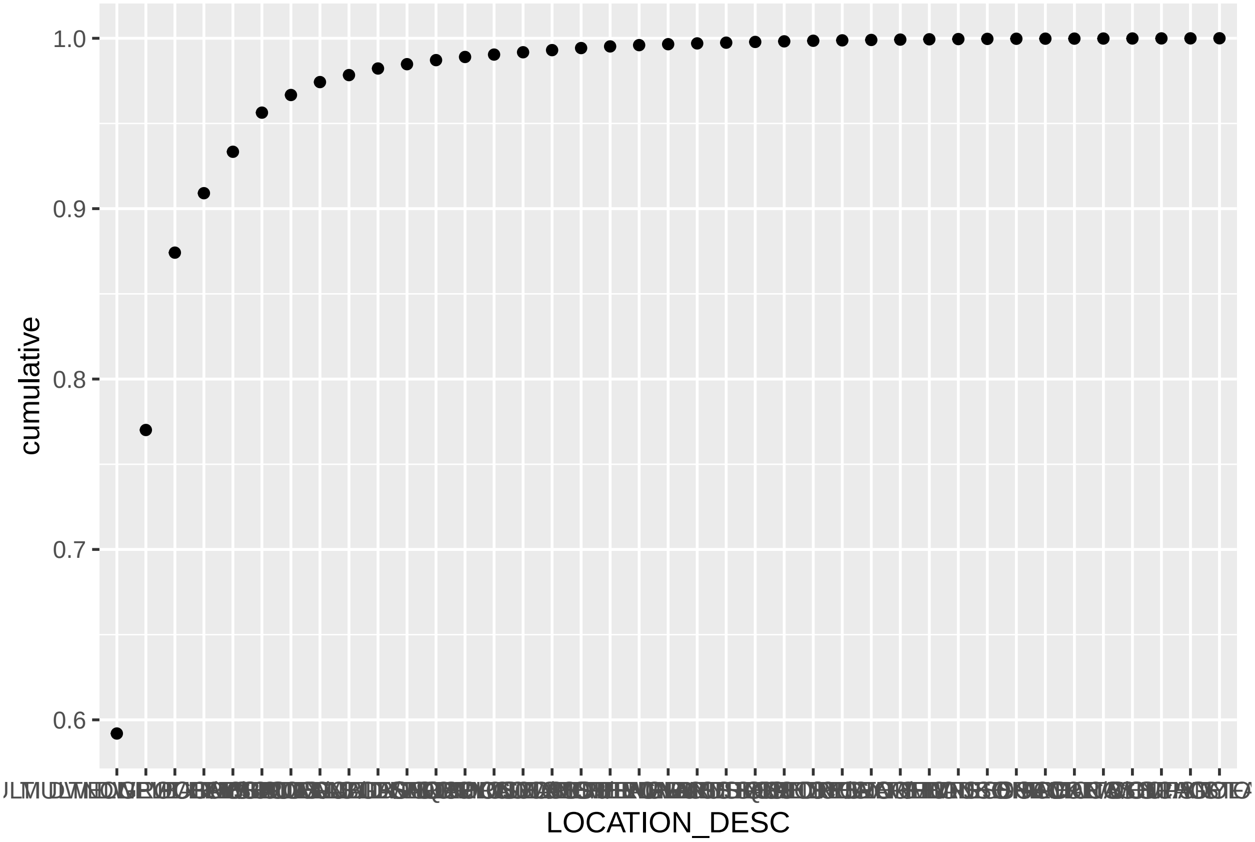
Note that in the original version of this chart, the per-borough, per-jurisdiction count was scaled as a fraction of the *total* annual incidents, not the *per borough* annual incidents. The result was that boroughs with higher overall incident counts appeared to have different jurisdictional splits. However this effect disappears when properly normalized as seen here.

## Location Desc

There are many location categories, but the most common 6 cover 95% of the total incidents:

```
by_location <- incidents %>%
  count(LOCATION_DESC=LOCATION_DESC) %>%
  arrange(desc(n)) %>%
  mutate(fraction=n/sum(n)) %>%
  mutate(LOCATION_DESC=fct_reorder(LOCATION_DESC, desc(fraction))) %>%
  mutate(cumulative=cumsum(fraction))

by_location %>%
#  filter(fraction > 0.0001)
```

```
  ggplot(aes(x=LOCATION_DESC, y=cumulative)) +
    geom_point()
```



```
top_locs <- head(by_location$LOCATION_DESC, 6)

by_location %>%
  filter(LOCATION_DESC %in% top_locs)
```

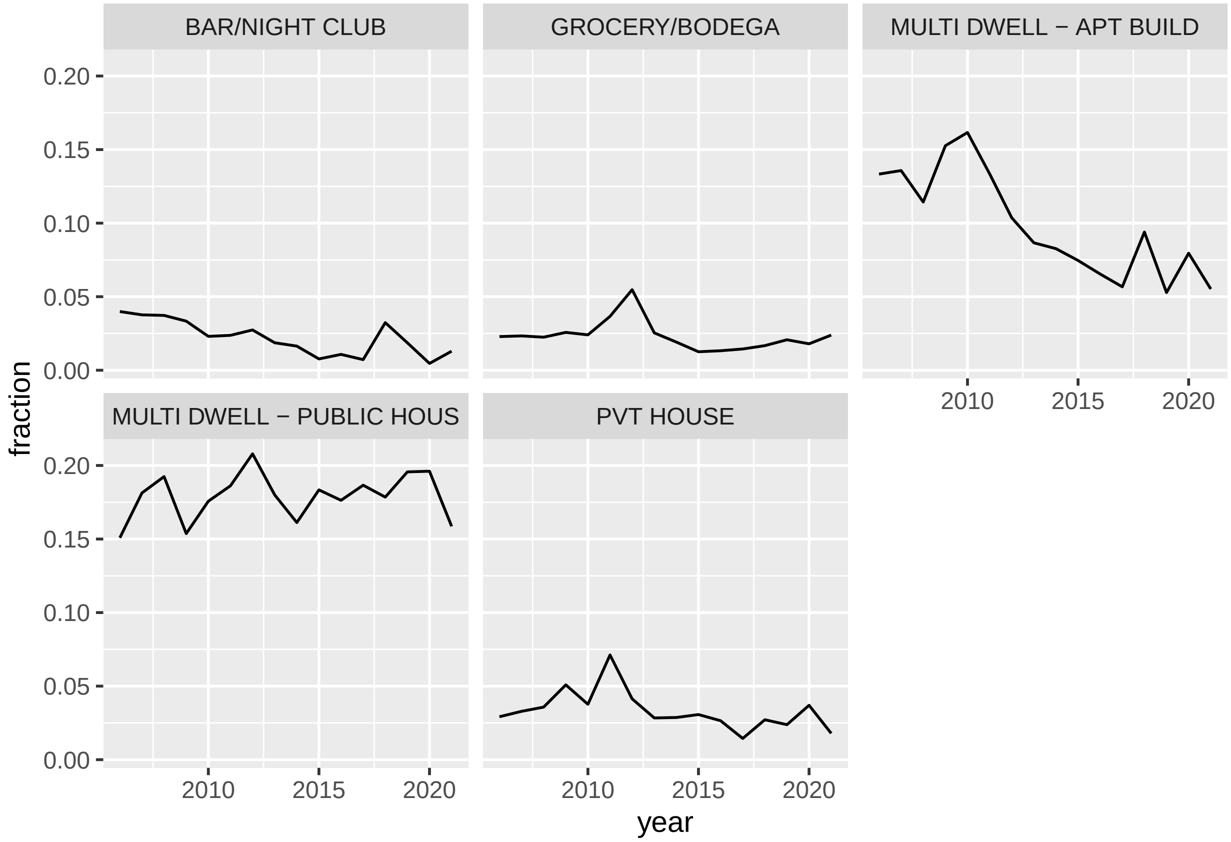
```
## # A tibble: 6 x 4
##   LOCATION_DESC                  n fraction cumulative
##   <fct>                      <int>    <dbl>      <dbl>
## 1 NONE                       15151   0.592       0.592
## 2 MULTI DWELL - PUBLIC HOUS  4559   0.178       0.770
## 3 MULTI DWELL - APT BUILD    2664   0.104       0.874
## 4 PVT HOUSE                    893   0.0349      0.909
## 5 GROCERY/BODEGA               622   0.0243      0.933
## 6 BAR/NIGHT CLUB               588   0.0230      0.956
```

```
#%>%
#  ggplot() +
#    geom_point(aes(x=LOCATION_DESC, y=fraction))
```

An important thing to note is that a full 58% of the incidents have location "NONE". It is unclear (and likely varies between reporters) whether this means "unknown" or "outside" or "none of the listed options". In the extreme, if these NONEs were all misclassified from a single location, then that location would immediately be the most common, regardless the other counts. Therefore this entire section of analysis must be considered very cautiously.

There is some defense for considering the most frequent values; these locations apparently are clear enough that reporters use them, and they occur often enough that it is unlikely they are all reporting errors. On that tenuous basis we will look at trends over time among these six:

```
incidents %>%
  count(LOCATION_DESC=LOCATION_DESC,
        year=floor_date(OCCUR_DATE, "year")) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  # n.total is the total incidents in the given year
  mutate(fraction=n/n.total) %>%
  # Drop NONE off the beginning, we don't know what it means and it
  # crushes the vertical scale.
  filter(LOCATION_DESC %in% tail(top_locs, 5)) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~LOCATION_DESC)
```



As with some other columns, graphs of the raw totals would suggest a shift toward public housing and apartment buildings in 2020, but they are in fact largely unchanged in fraction of annual incidents, as seen here.

## Victim Demographics

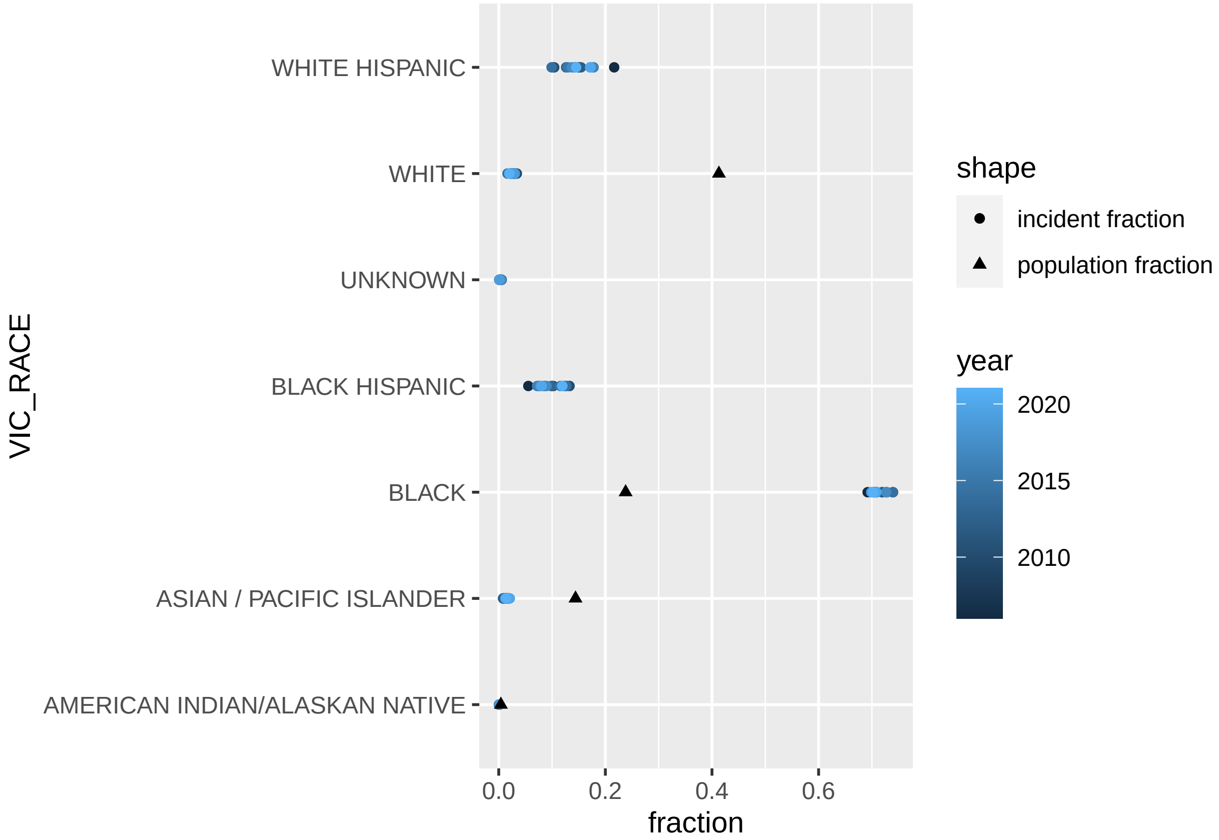
### VIC_RACE

We can use census demographic data from https://www.census.gov/quickfacts/newyorkcitynewyork to consider victim race in light of population demographics. One immediate concern is that the census ACS categories are different than the reporting categories in this incident data; in particular, the ACS does not split into BLACK HISPANIC and WHITE_HISPANIC, and does encode "Two or more Races".

The plot below presents population markers only for the aligned categories. However we should note that the difference in categories is itself a direct source of bias: if we cannot easily measure and compare these measures across different datasets, that serves to obscure the true differential impact of shooting incidents.

```
nyc_demo = tibble(
  VIC_RACE=levels(incidents$VIC_RACE),
  # Per <https://www.census.gov/quickfacts/newyorkcitynewyork>. This source
  # does not split out values for BLACK HISPANIC vs WHITE HISPANIC, nor
  # UNKNOWN
  fraction=c(0.4, 14.3+0.1, 23.8, NA, NA, 41.3, NA)/100.0
)

by_vic_race = incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        VIC_RACE=VIC_RACE) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total)
ggplot() +
  geom_point(data=by_vic_race, aes(VIC_RACE, fraction, shape="incident fraction", color=year)) +
  geom_point(data=nyc_demo, aes(VIC_RACE, fraction, shape="population fraction")) +
  coord_flip()
```

```
## Warning: Removed 3 rows containing missing values (‘geom_point()‘).
```



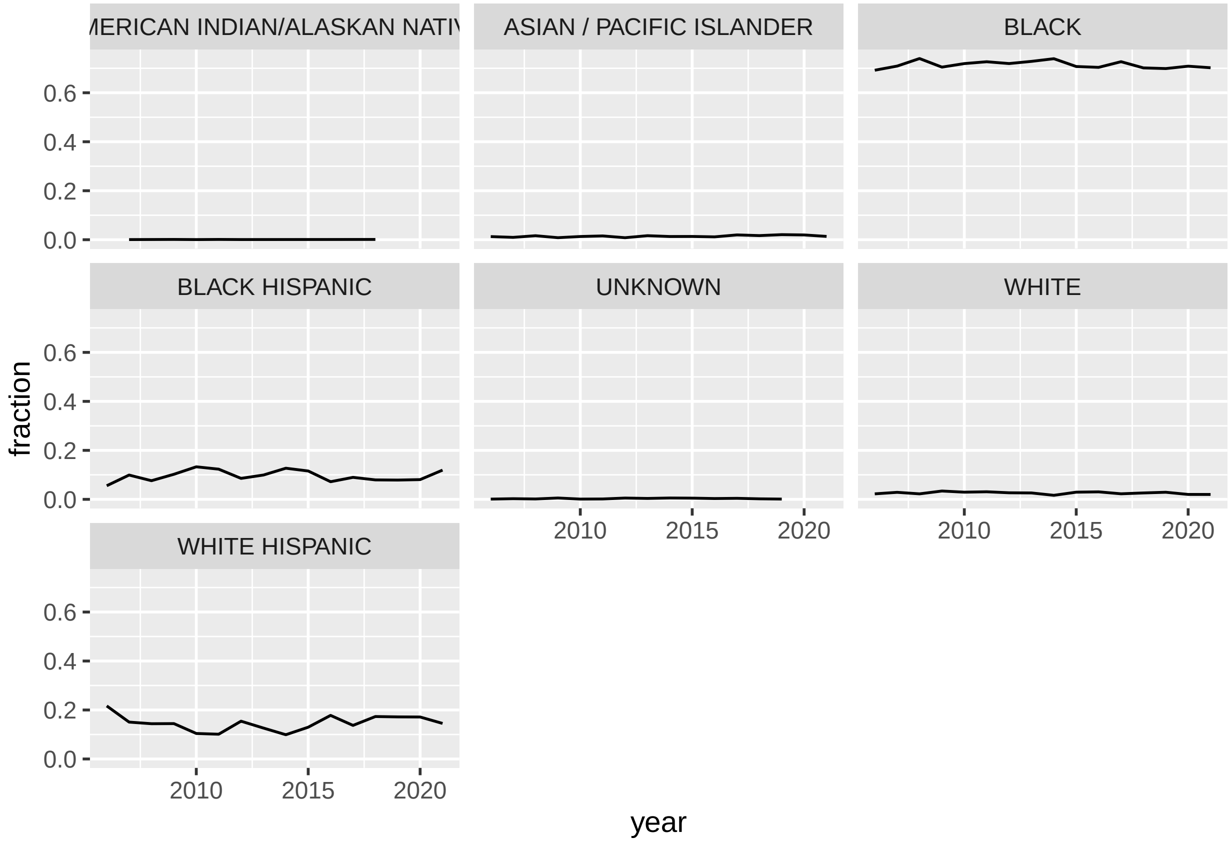
From the limited data available, the primary conclusion is that people coded as BLACK are the victims in incidents far more often than their portion of the population as a whole, and those coded as WHITE, far less often; ASIAN/PACIFIC ISLANDER is also a lower proportion though less extremely than WHITE.

The incident-fraction values are clearly quite divergent from the population-fraction values. This reveals a true disparate impact of incidents across the races; however there are several issues to confirm, to understand the context and potential for over- or under-stating these figures:

- Who determines the race recorded for a victim? Is it the victim themselves? (Unlikely in the event of a "successful" shooting.) Is it a police officer, or someone else, or does it vary?
- What are the guidelines for characterizing races? Are there incentives on the person recording, or their organization, to prefer to record in one way or another?
- How does this compare to the way race is recorded in the American Community Survey, which provides the population figures? Are there incentive structures there to skew the data one way or another?
- In particular, how do the "Two or more Races" and "Hispanic or Latino" census categories relate to the BLACK HISPANIC and WHITE HISPANIC incident categories?

Plotting the race of victims over time, the proportions are relatively stable:

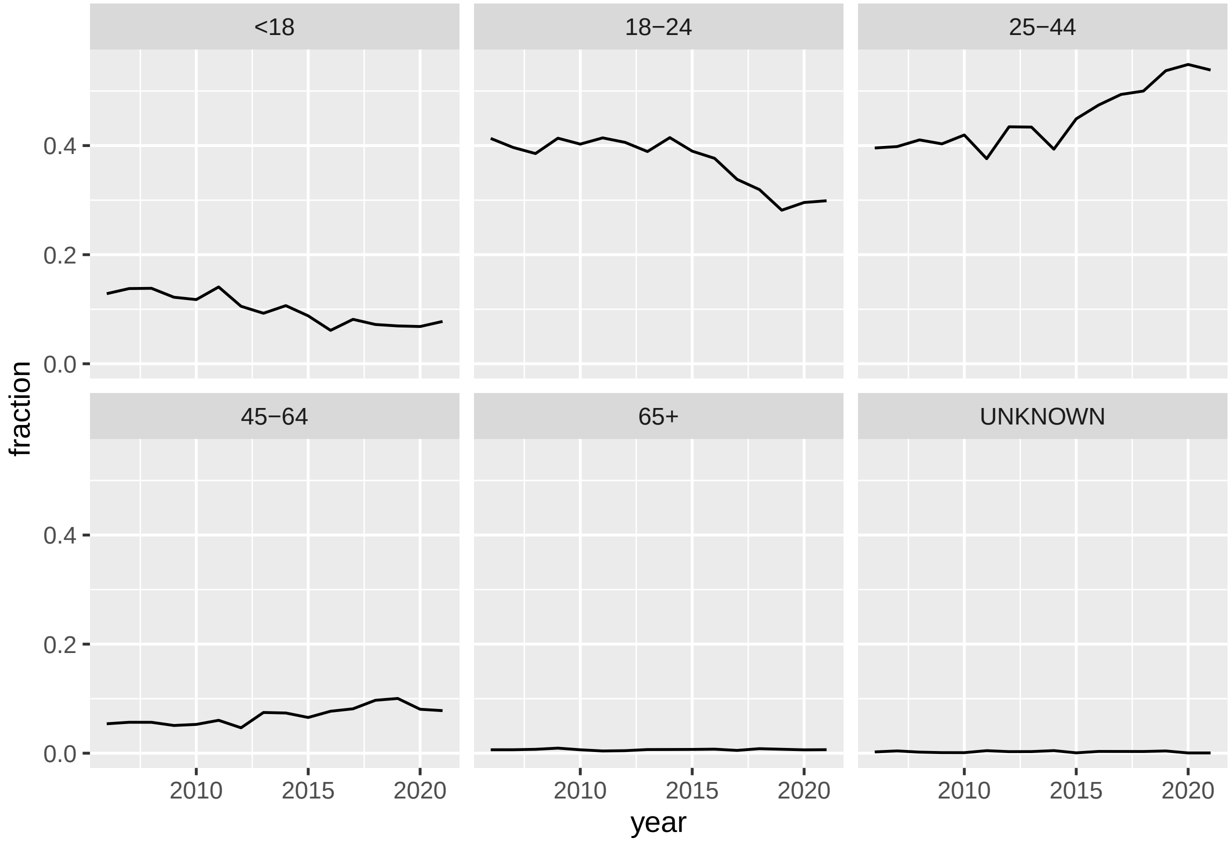
```
incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        VIC_RACE=VIC_RACE) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~VIC_RACE)
```



### VIC_AGE

We can examine the portion of annual incidents in each victim age group, over the years:

```
incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        VIC_AGE_GROUP=VIC_AGE_GROUP) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~VIC_AGE_GROUP)
```

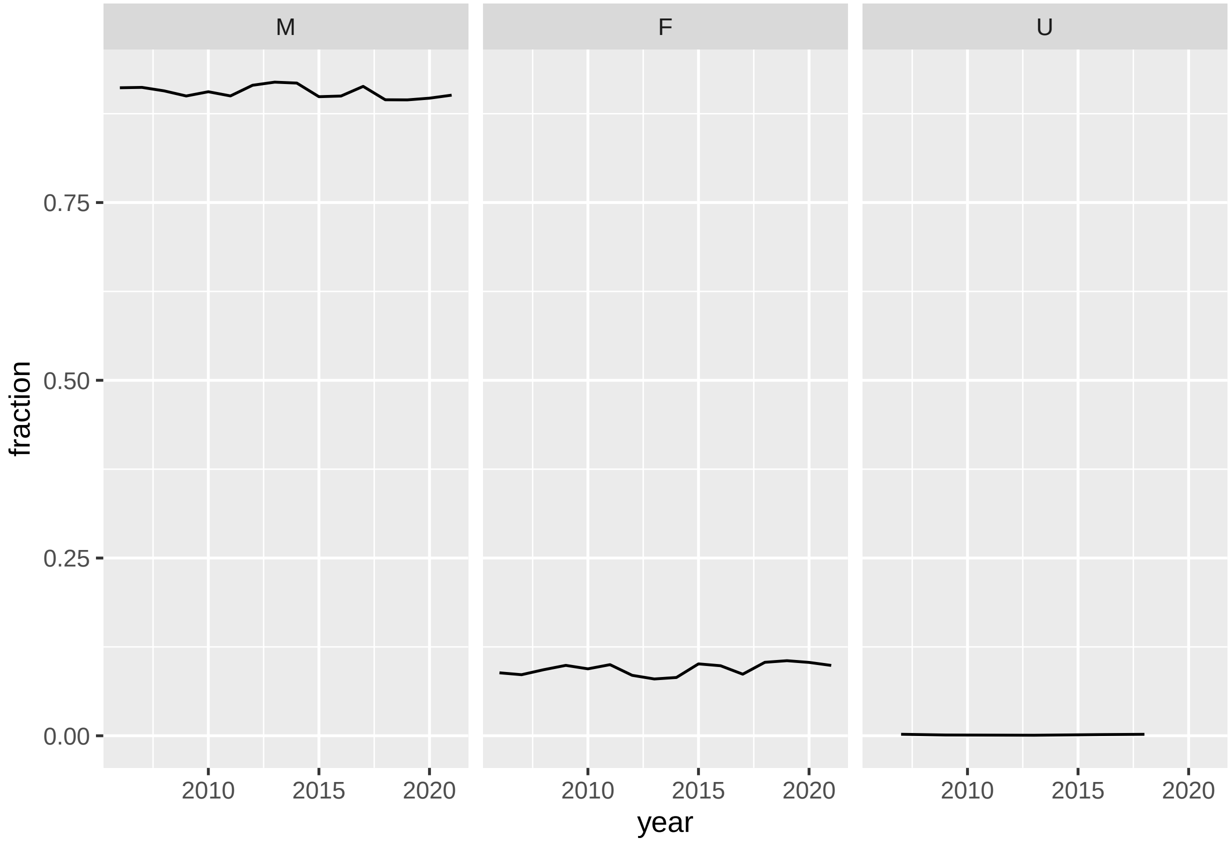


This reflects a modest shift over the past 5 years from 18-24 year old victims toward 25-44 year old victims. One angle for investigation would be to compare this shift to any overall shift in the age distribution of New Yorkers during this time window.

### VIC_SEX

The dataset presents a very "traditionalist" view of gender (which is likely what is intended), including referring to it as "sex" and structuring it as a strict dichotomy. This very encoding limits some kinds of analyses: for example it is impossible to assess the impact of shooting incidents on non-binary individuals when they are simply invisible in the data.

```
incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        VIC_SEX=VIC_SEX) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~VIC_SEX)
```

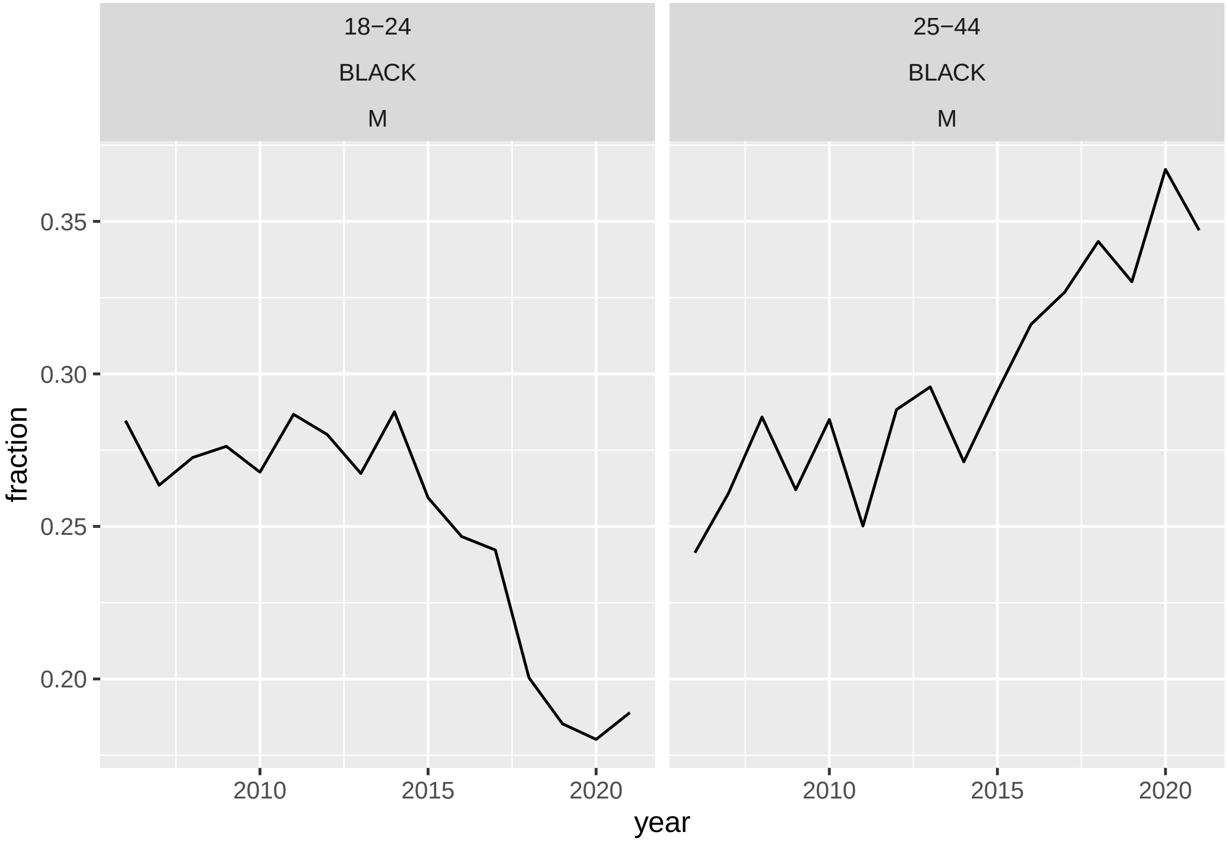


Victims are male far more often than female or unknown gender, consistently over time. This is subject to the same coding and reporting questions as VIC_RACE, but it seems equally certain that there is a real disparity here, though the exact magnitude would need to be carefully examined.

### Jointly: Age, Gender and Race

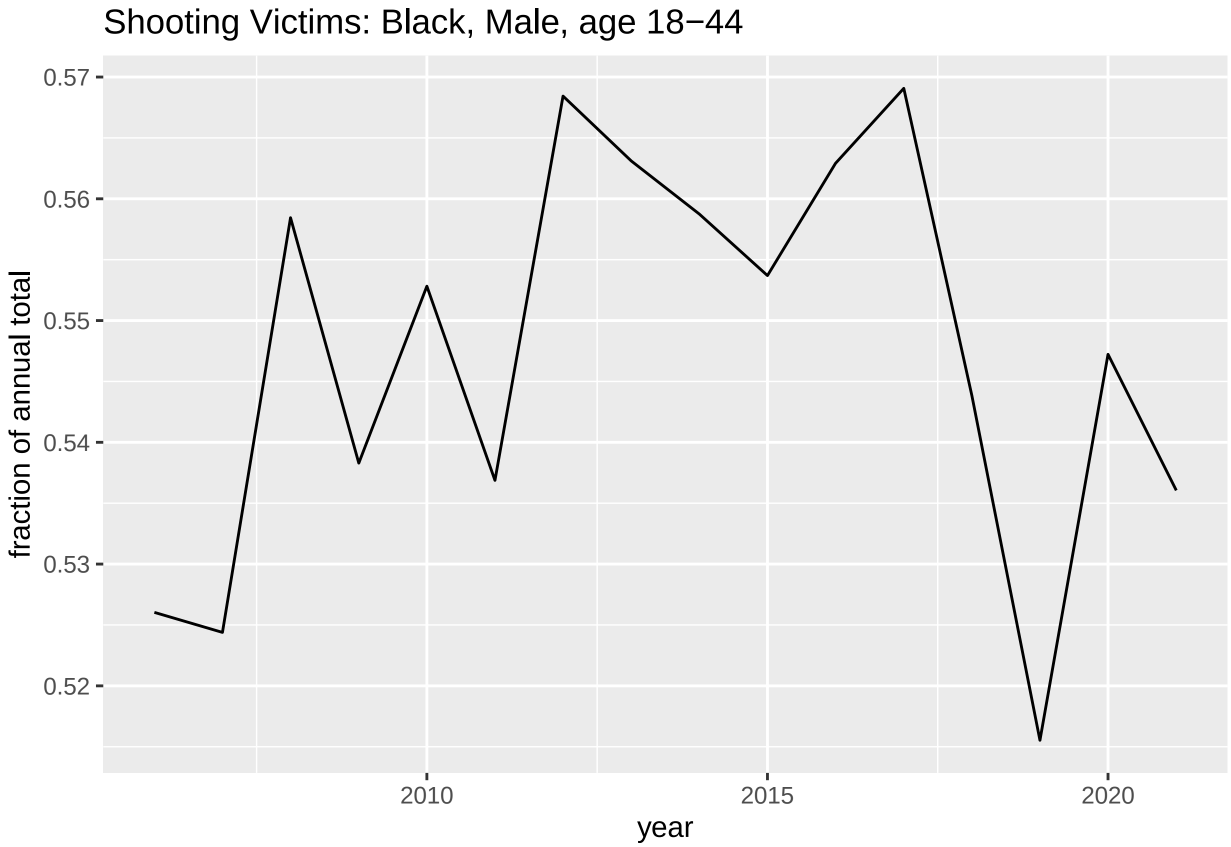
There are a combinatorial number of age, race and gender graphs we could examine, but the key story is told by just two, if we filter to the intersectional categories which suffer more than 10% of the city's incidents each year:

```
vic_demo = incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        VIC_RACE=VIC_RACE,
        VIC_AGE_GROUP=VIC_AGE_GROUP,
        VIC_SEX=VIC_SEX) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total)

#vic_demo %>%
#  filter(VIC_SEX=="M") %>%
#  ggplot(aes(year, fraction)) +
#    geom_line() +
#    facet_grid(rows=vars(VIC_RACE), cols=vars(VIC_AGE_GROUP))

vic_demo %>%
  filter(fraction > 0.1) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~VIC_AGE_GROUP + VIC_RACE + VIC_SEX)
```



Shooting victimhood is distributed between these two groups, along an arbitrary boundary at age 25. If we sum them together:

```
vic_demo %>%
  filter(fraction > 0.1) %>%
  group_by(year) %>%
  summarize(fraction=sum(fraction)) %>%
  ggplot(aes(year, fraction)) +
    ggtitle("Shooting Victims: Black, Male, age 18-44") +
    ylab("fraction of annual total") +
    geom_line()
```



Black men between the ages of 18 and 44 suffer slightly more than half of all shooting incidents in the city. Further research could establish a true population proportion for this group; if we assume (incorrectly) that age, gender and race are independently distributed, we can use figures from https://www.census.gov/quickfacts/fact/table/newyorkcitynewyork to estimate this group as around 7% of the total population:

```
frac_female=0.523
frac_lt_18=0.207
frac_gt_65=0.149
frac_black=0.238

print(c("Approx fraction 18-65 black males: ", (1-frac_female) * (1-(frac_lt_18+frac_gt_65)) * frac_bla
```

```
## [1] "Approx fraction 18-65 black males: " "0.073110744"
```

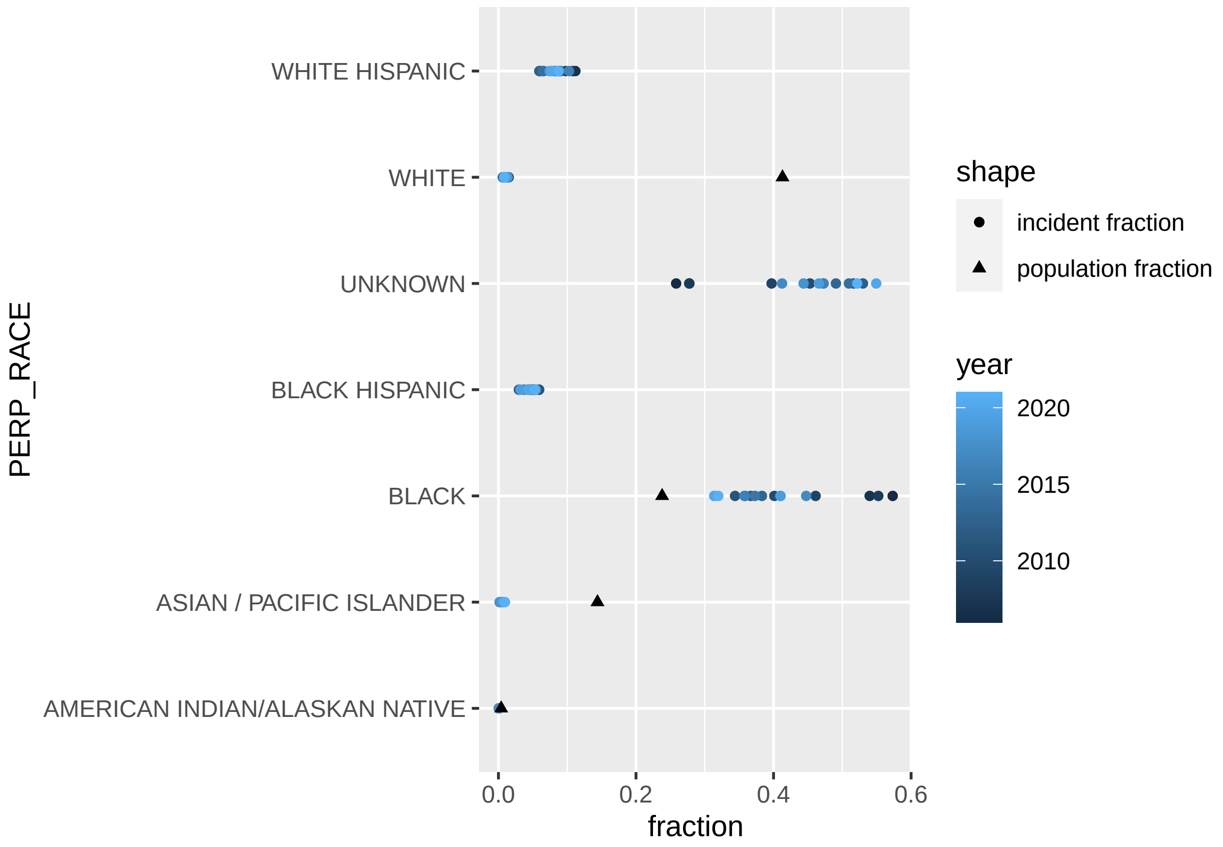
## Perpetrator Demographics

We can repeat a similar analysis for the perpetrator, though in many more cases this information is unknown. Note that all of the data provenance and recording / encoding questions from victim demographics apply here as well, though crucially the reporters' incentives may differ when recording the attributes of the perpetrator compared to those of the victim (so the questions are similar but the answers may differ).

### PERP_RACE

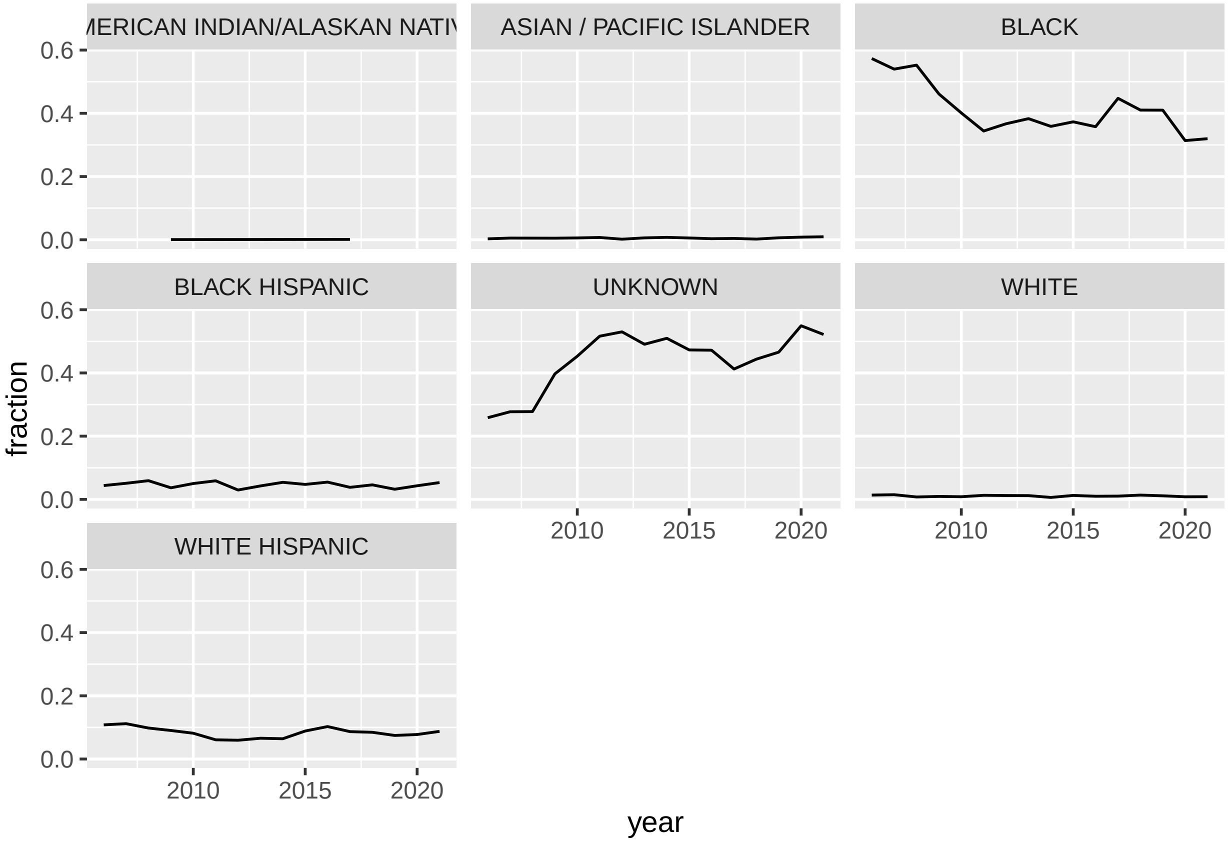
```
by_perp_race = incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        PERP_RACE=PERP_RACE) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total)
ggplot() +
  geom_point(data=by_perp_race, aes(PERP_RACE, fraction, shape="incident fraction", color=year)) +
  geom_point(data=nyc_demo, aes(VIC_RACE, fraction, shape="population fraction")) +
  coord_flip()
```

```
## Warning: Removed 3 rows containing missing values (‘geom_point()‘).
```



The perpetrator race data is similar to the victim race, in terms of over-representation for black-coded individuals and under-representation for white-coded and asian/pacific-islander-coded individuals. The major different dynamic is the large block of UNKNOWN values, as well as a wider spread in the annual values for the different classifications. Examining the trends over time:

```
incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        PERP_RACE=PERP_RACE) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~PERP_RACE)
```



we see that BLACK and UNKNOWN reports have traded positions over the dataset, with the UNKNOWN portion rising as the BLACK portion drops; other categories appear to remain roughly constant. This pattern could represent many different things:

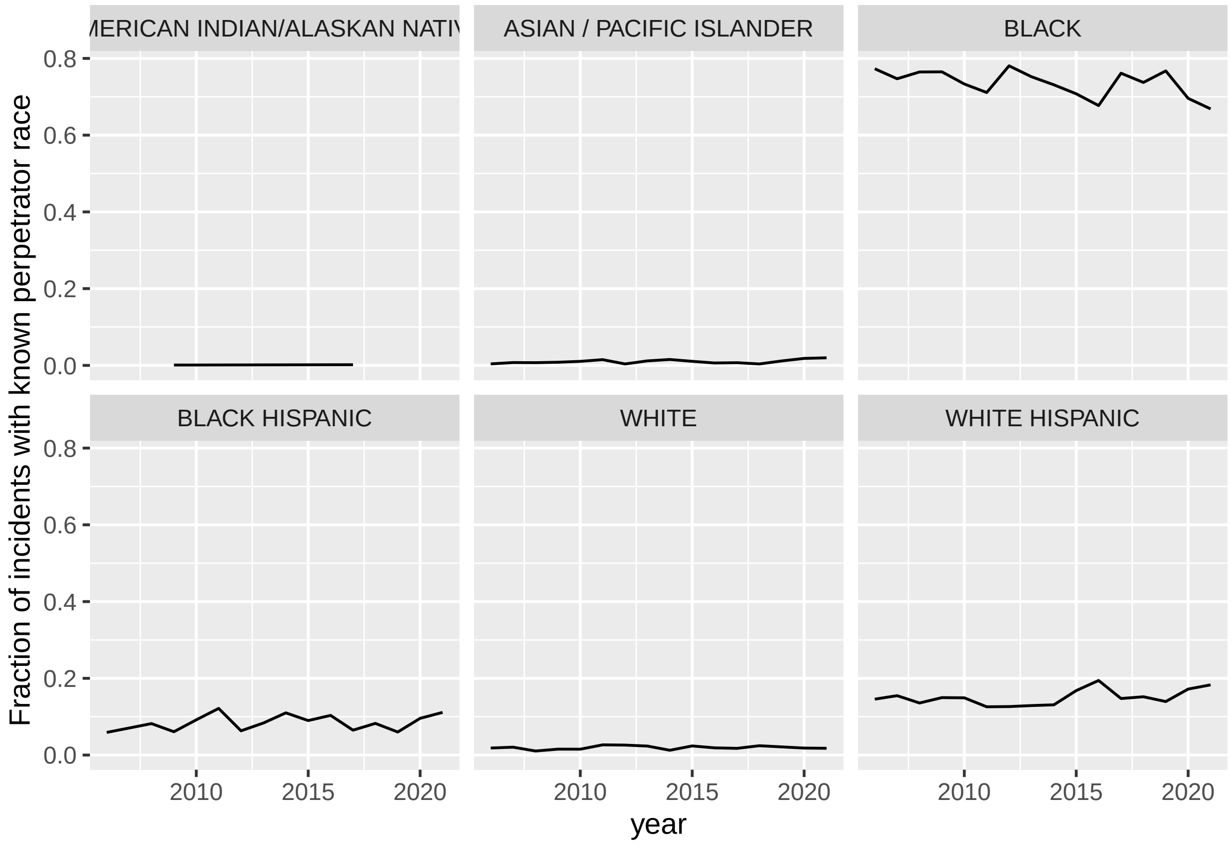
- Maybe fewer BLACK-coded individuals are perpetrating incidents, and this data reflects that; the UNKNOWNs might in fact all be recorded as another category if they could be identified.
- As the plurality of perpetrators have historically been recorded as black, this could signify that fewer perpetrators are being caught overall, and the trend is simply more visible for the BLACK category.
- Subtly differently, if the data is updated when a perpatrator is identified, it is possible that the rise in UNKNOWN signifies shootings that are not yet solved, and in the future those incidents may be reassigned once a perpatrator is identified.
- Reporting and recording standards may have changed over time, and an identification of a black perpetrator may be less of an immediate default than it once was.
- The source of this data may be less objective than it presents. For example, if the `PERP_*` fields are based on a victim's or witness's description, rather than an independently identified individual, it is possible that fewer descriptions are confident about the race of the perpetrator than previously.

Separately from this shift over time, we should also exercise caution about interpreting the high BLACK proportion, as this may be based on individuals who are caught by police and charged with a crime, which may in turn reflect many layers of bias and inequity in the police, prosecutor's office, and other institutions.

We can also filter out UNKNOWN entries and plot the breakdown of the remaining annual incidents:

```
known_annual <- incidents %>%
  filter(PERP_RACE != "UNKNOWN") %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        PERP_RACE=PERP_RACE)
```

```
known_annual %>%
  left_join(
    known_annual %>%
      group_by(year) %>%
      summarize(annual_total=sum(n)),
    by=c('year')
  ) %>%
  mutate(fraction=n / annual_total) %>%
  ggplot(aes(year, fraction)) +
    ylab("Fraction of incidents with known perpetrator race") +
    geom_line() +
    facet_wrap(~PERP_RACE)
```

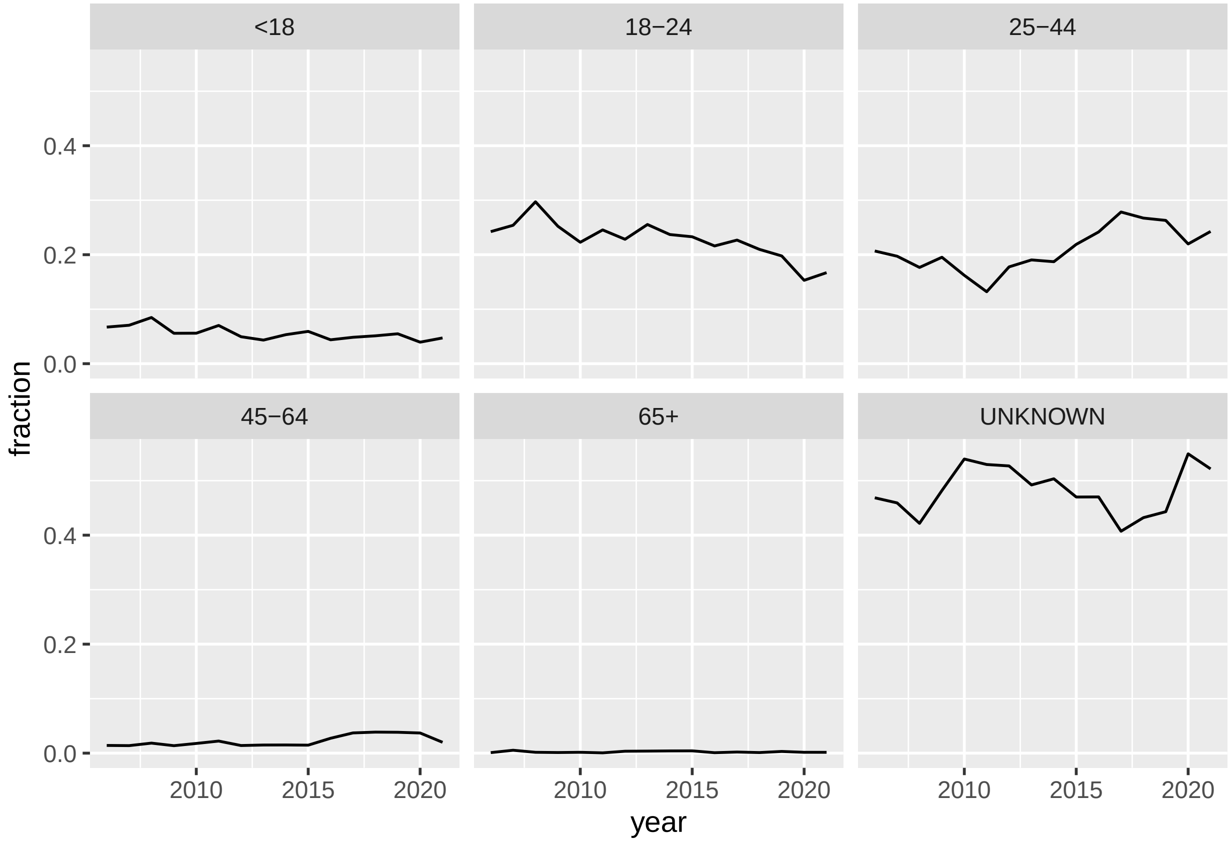


The stability seen in these proportions suggests that the increasing fraction of UNKNOWN perpetrators drives all the changes seen in the previous set of graphs, and in fact the breakdown has been steady over time.

### PERP_AGE

We can examine the portion of annual incidents in each victim age group, over the years:

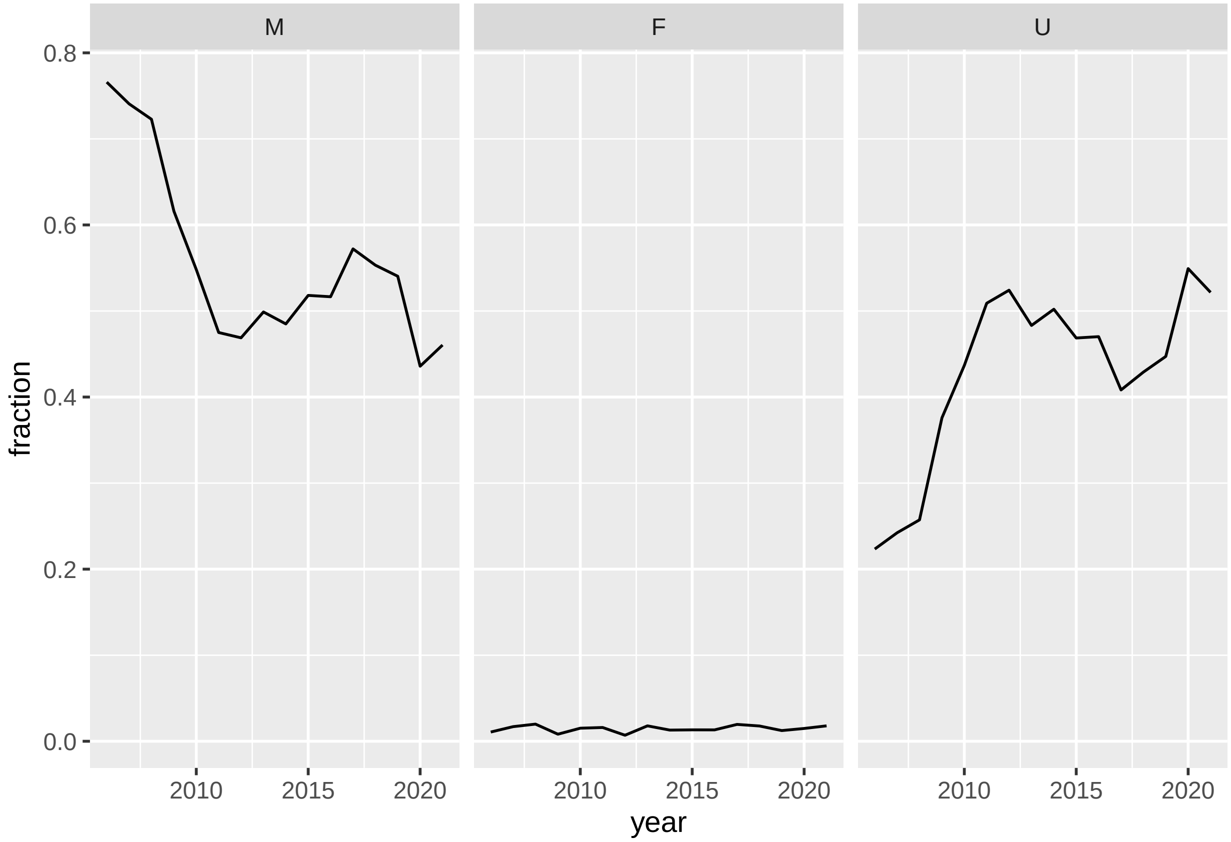
```
incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        PERP_AGE_GROUP=PERP_AGE_GROUP) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~PERP_AGE_GROUP)
```



Similar to race, the same patterns emerge from PERP and from VIC age data; most perpetrators are 18-44, though the magnitude of those lines is lower due to the high fraction of UNKNOWN ages.

### PERP_SEX

```
incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        PERP_SEX=PERP_SEX) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~PERP_SEX)
```

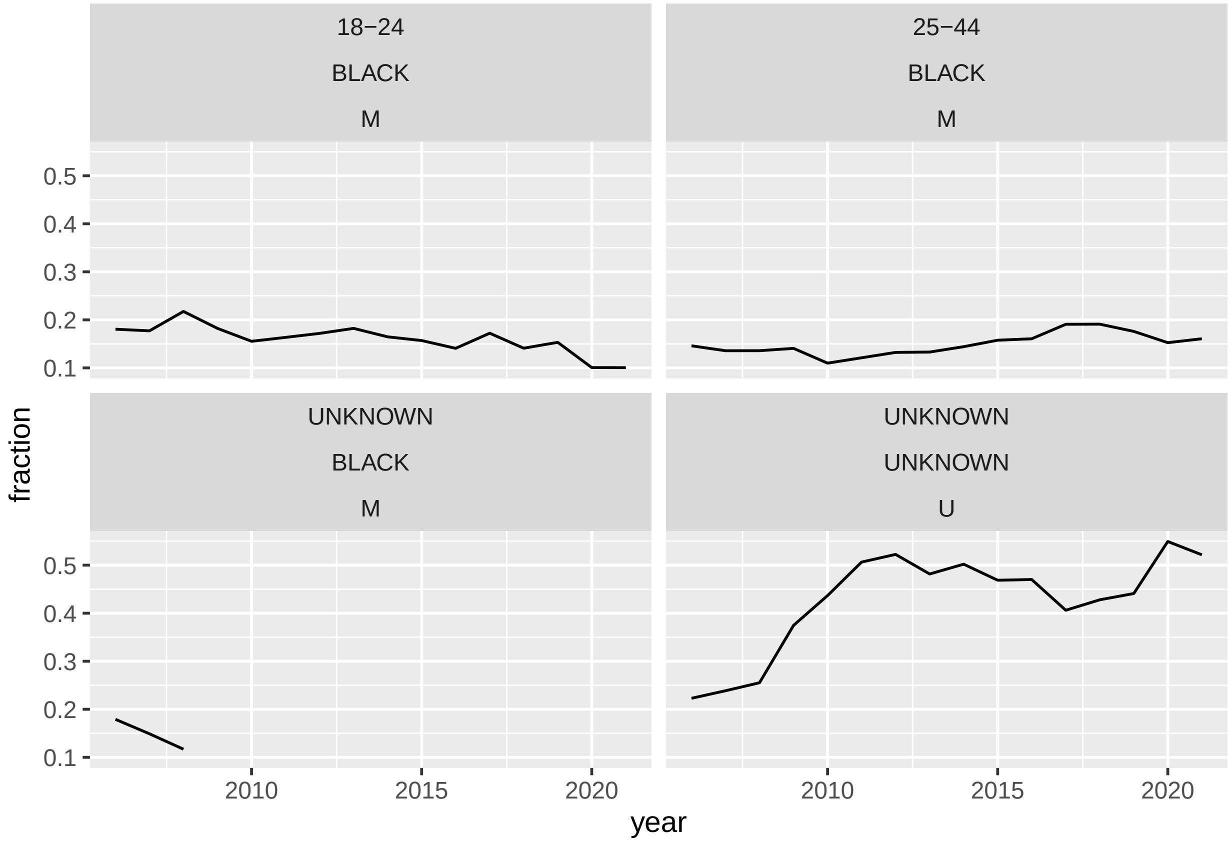


As with victims, most perpetrators are male, with an increasing fraction of UNKNOWN as the dates approach the present day.

### Jointly: Perpetrators

Again let us consider the higher-prevalence Age/Gender/Race groups:

```
perp_demo = incidents %>%
  count(year=floor_date(OCCUR_DATE, "year"),
        PERP_RACE=PERP_RACE,
        PERP_AGE_GROUP=PERP_AGE_GROUP,
        PERP_SEX=PERP_SEX) %>%
  left_join(yearly, by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total)

#vic_demo %>%
#  filter(VIC_SEX=="M") %>%
#  ggplot(aes(year, fraction)) +
#    geom_line() +
#    facet_grid(rows=vars(VIC_RACE), cols=vars(VIC_AGE_GROUP))

perp_demo %>%
  filter(fraction > 0.1) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    facet_wrap(~PERP_AGE_GROUP + PERP_RACE + PERP_SEX)
```

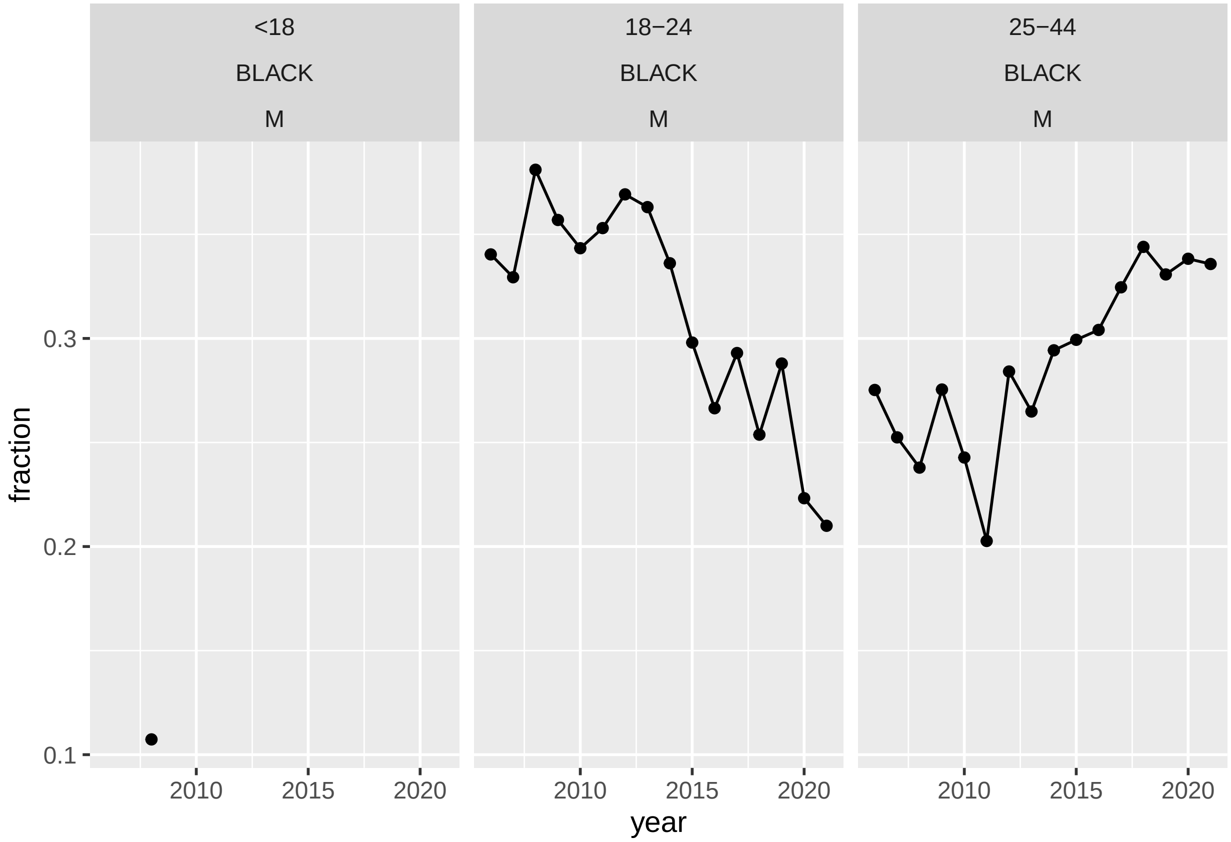


Where not unknown, this is the same group as the victims: male, black, aged 18-44. The same demographic group is both the most common perpetrator and victim in these incidents, as encoded within this dataset.

We can again consider only those incidents with a known perpetrator:

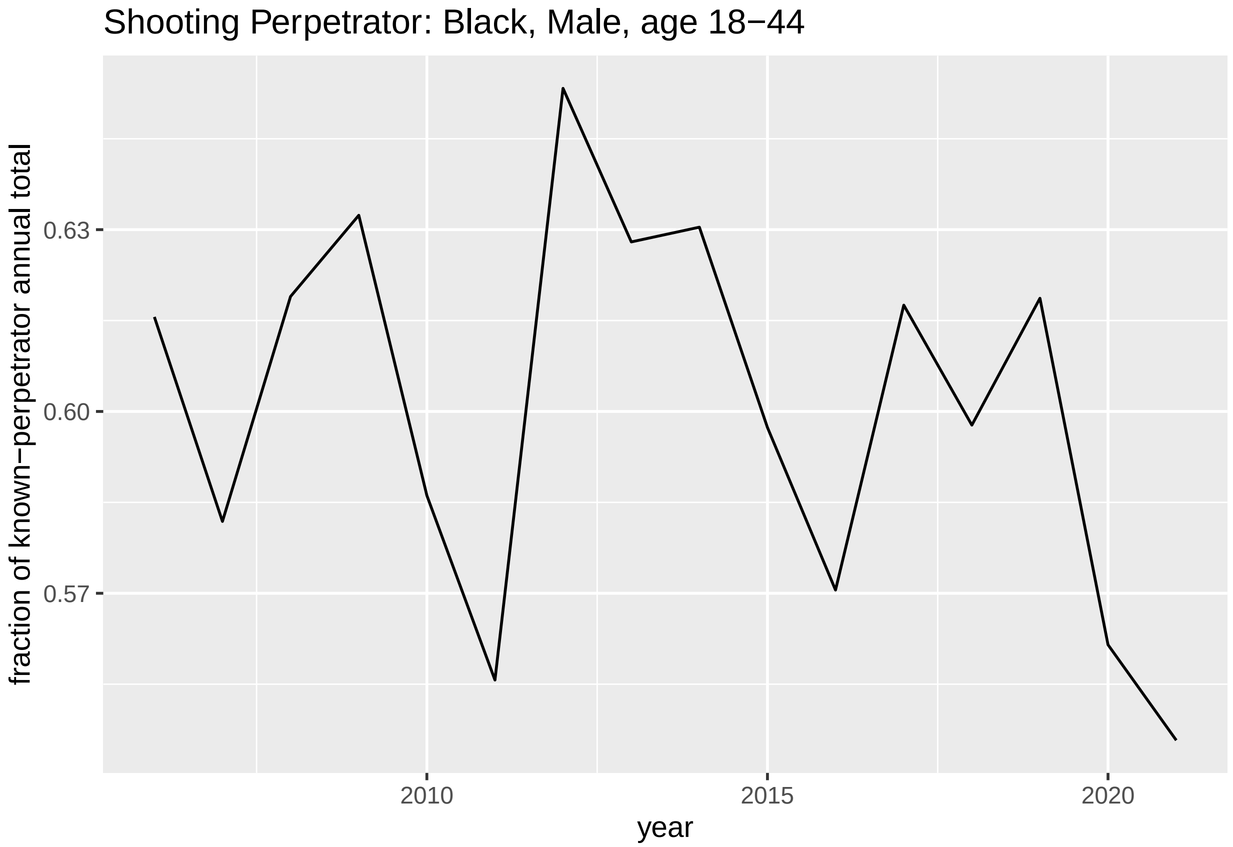
```
known_perp_demo = incidents %>%
  filter((PERP_RACE != "UNKNOWN") & (PERP_AGE_GROUP != "UNKNOWN")) %>%
  mutate(year=floor_date(OCCUR_DATE, "year"))
yearly_perp_known_demo <- known_perp_demo %>%
  count(year=year,
        PERP_RACE=PERP_RACE,
        PERP_AGE_GROUP=PERP_AGE_GROUP,
        PERP_SEX=PERP_SEX) %>%
  left_join(known_perp_demo %>% count(year=year),
            by="year", suffix=c("", ".total")) %>%
  mutate(fraction=n / n.total)

yearly_perp_known_demo %>%
  filter(fraction > 0.1) %>%
  ggplot(aes(year, fraction)) +
    geom_line() +
    geom_point() +
    facet_wrap(~PERP_AGE_GROUP + PERP_RACE + PERP_SEX)
```

```
## ‘geom_line()‘: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```



We see a very similar pattern to victim demographics. Paralleling our analysis from there, we can combine the slices:

```
yearly_perp_known_demo %>%
  # Bump filter slightly higher to drop a single year of Black Male <18
  # which is valid but makes the age group more diffuse.
  filter(fraction > 0.15) %>%
  group_by(year) %>%
  summarize(fraction=sum(fraction))  %>%
  ggplot(aes(year, fraction)) +
    ggtitle("Shooting Perpetrator: Black, Male, age 18-44") +
    ylab("fraction of known-perpetrator annual total") +
    geom_line()
```



Despite some mild ups and downs (note the narrow vertical scale on this chart), the overall proportion has been stable and high. As seen in the victim demographic analysis, this is vastly out of proportion to the population fraction of this demographic group.

## Bias Considerations

There are numerous sources of possible bias in this analysis, both within the original data and in the analysis processes adopted in this paper.

### Within the Data Source

On a column-by-column basis, many fields encode possible biases on the parts of the process and the individuals generating and encoding the data. As noted above, demographic fields like race and gender cannot be measured objectively, and this leads to many opportunities for the as-reported data to reflect the preconceptions of those generating it, or to align with the reporting trends which would most benefit them.

The perpetrator's demographic columns are even more fertile ground for possible bias and preconception to emerge and be encoded into the data; the lack of transparency about the meaning and source of these fields in the "footnotes" PDF is itself a warning sign that this data may be intentionally opaque or allowing itself to be misinterpreted.

Both the "sex" and "race" fields demonstrate a structural type of bias, inherent in the dataset schema and largely independent of the biases of the individual reporters. By specifying only "M", "F", "U" fields for sex, people of other genders are invisible in the dataset, and their impact or victimhood cannot be assessed. Similarly, the "race" levels are quite arbitrary, and because they do not align with the levels in Census data, it makes it impossible to do some kinds of baselining or normalization.

One very important bias in the data is invisible: those incidents which are not recorded. Are there types of shooting incident, or possible shooting incidents, that are not present in the dataset? Is there any judgement call in what gets recorded as a shooting incident, and is there any influence on that judgement to record those incidents which fit some set of expected parameters or narratives?

### Within the Analysis

The author has a strong prior expectation that gun violence and hence shooting incidents will be concentrated, in both victim and perpetrator, among the poorest and most disadvantaged populations. He also notes that many types of law enforcement misconduct or bias have historically impacted the same populations; as this dataset is apparently gathered by the NYPD, both of those factors may influence both who is involved in shooting incidents, and how those incidents are recorded. In an attempt to mitigate the impact of authorial bias, caution has been taken to lead with the data in all cases, for example sorting by category prevalence rather than selecting data for specific racial categories.

In a more procedural area, the author assumes that analyzing the *proportion of annual incidents* is a meaningful way to correct for annual fluctuations in the level of incidents. This assumption is broadly validated by several columns which appear to have meaningful changes when considered in absolute counts, but those flatten to a stable proportion when considered as a proportion of annual incidents. However, this proportional analysis serves to hide the actual scale of counts in each category or level. If there are psychologically or otherwise meaningful thresholds in absolute counts – for example a hypothetical regulation might trigger an outreach program for demographic groups suffering more than 1,000 annual shooting incidents – those would be invisible in the analysis presented here.

## Conclusion

This is a complex and highly structured dataset. There is clear periodic structure, at multiple scales (time of day versus time of year, for example) in the times when incidents occur. There is substantial skew in the demographic breakdown of both victims and perpetrators, compared to the overall population of the city. We must use caution in drawing conclusions here, especially without more information about the source and meaning of these fields. However, it appears that individuals involved in shootings, whether as victims or perpetrators, are vastly more likely to be recorded as BLACK, MALE, and in the 18-24 or 25-44 age groups.

## Appendix

### Session Info

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 23403)
##
## 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
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.2  forcats_1.0.0    stringr_1.5.0    dplyr_1.1.0
##  [5] purrr_1.0.1      readr_2.1.4      tidyr_1.3.0      tibble_3.2.0
##  [9] ggplot2_3.4.1    tidyverse_2.0.0  conflicted_1.2.0
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0 xfun_0.37        splines_4.2.2    lattice_0.20-45
##  [5] colorspace_2.1-0 vctrs_0.5.2      generics_0.1.3   htmltools_0.5.4
##  [9] yaml_2.3.7       mgcv_1.8-41      utf8_1.2.3       rlang_1.0.6
## [13] pillar_1.8.1     glue_1.6.2       withr_2.5.0      bit64_4.0.5
## [17] lifecycle_1.0.3  munsell_0.5.0    gtable_0.3.1     memoise_2.0.1
## [21] evaluate_0.20    labeling_0.4.2   knitr_1.42       tzdb_0.3.0
## [25] fastmap_1.1.1    parallel_4.2.2   curl_5.0.0       fansi_1.0.4
## [29] highr_0.10       scales_1.2.1     cachem_1.0.7     vroom_1.6.1
## [33] farver_2.1.1     bit_4.0.5        hms_1.1.2        digest_0.6.31
## [37] stringi_1.7.12   grid_4.2.2       cli_3.6.0        tools_4.2.2
## [41] magrittr_2.0.3   crayon_1.5.2     pkgconfig_2.0.3  ellipsis_0.3.2
## [45] Matrix_1.5-1     timechange_0.2.0 rmarkdown_2.20   rstudioapi_0.14
## [49] R6_2.5.1         nlme_3.1-160     compiler_4.2.2
```