ShootingProject

Peter Crona

2023-01-06

Question

Do male and female perpetrators commit crimes at different times of the day?

Get, clean and extract relevant data

This analysis uses "NYPD Shooting Incident Data (Historic)", see https://data.cityofnewyork.us/Public-Safety/NYPD-Shooting-Incident-Data-Historic-/833y-fsy8 for details.

For the sake of our analysis, we only need PERP_SEX, which contains M for male and F for female, and OCCUR_TIME which contains the time of day of the crime / shooting incident.

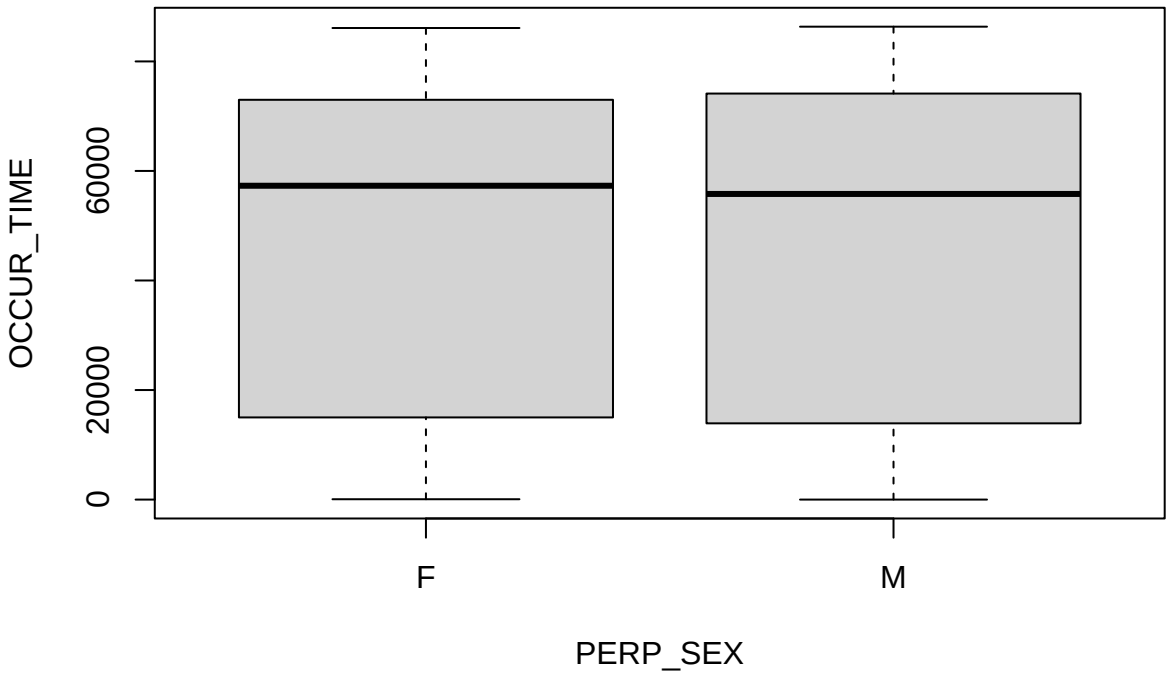
```
data <- read_csv("https://data.cityofnewyork.us/api/views/833y-fsy8/rows.csv") %>%
  dplyr::select(OCCUR_TIME, PERP_SEX) %>%
  drop_na() %>%
  filter(PERP_SEX == "M" | PERP_SEX == "F")
```

```
## 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.
```

Visualize using boxplot

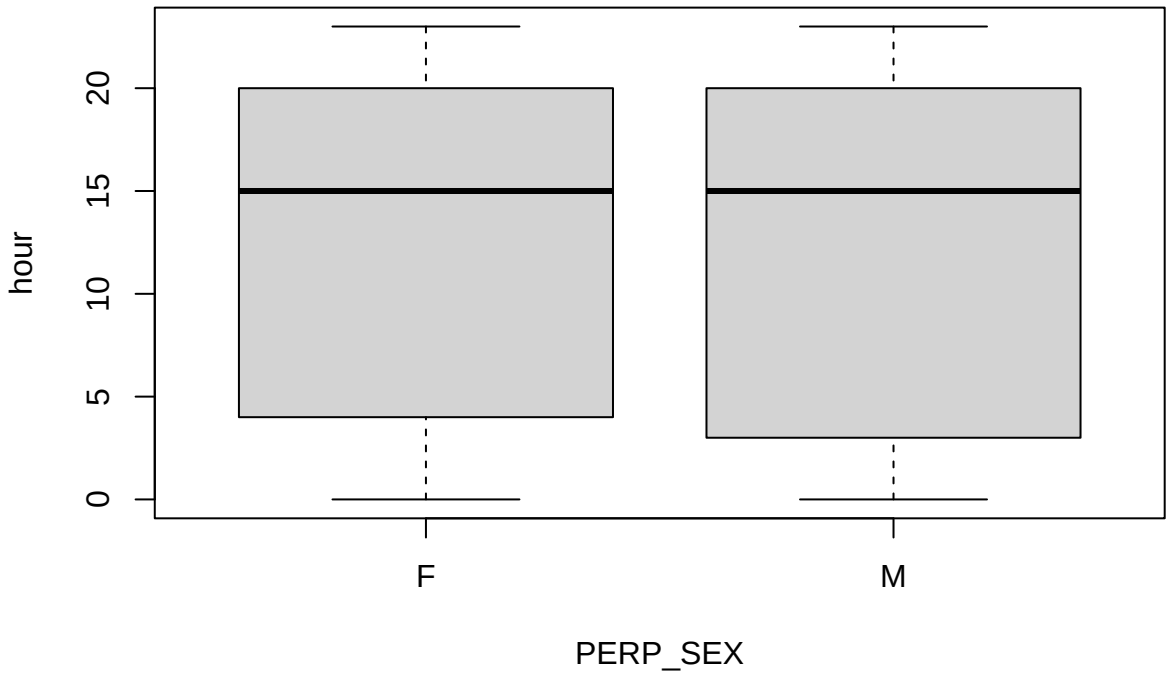
We start by visualizing the data to see if we immediately can spot a difference:

```
boxplot(OCCUR_TIME ~ PERP_SEX, data = data)
```



The visualization is not especially readable as we show time as seconds rather than hour of day. To make it more readable, we use hour of day rather than seconds.

```
data_w_hour <- data %>%
  mutate(hour = hour(OCCUR_TIME))

boxplot(hour ~ PERP_SEX, data = data_w_hour)
```



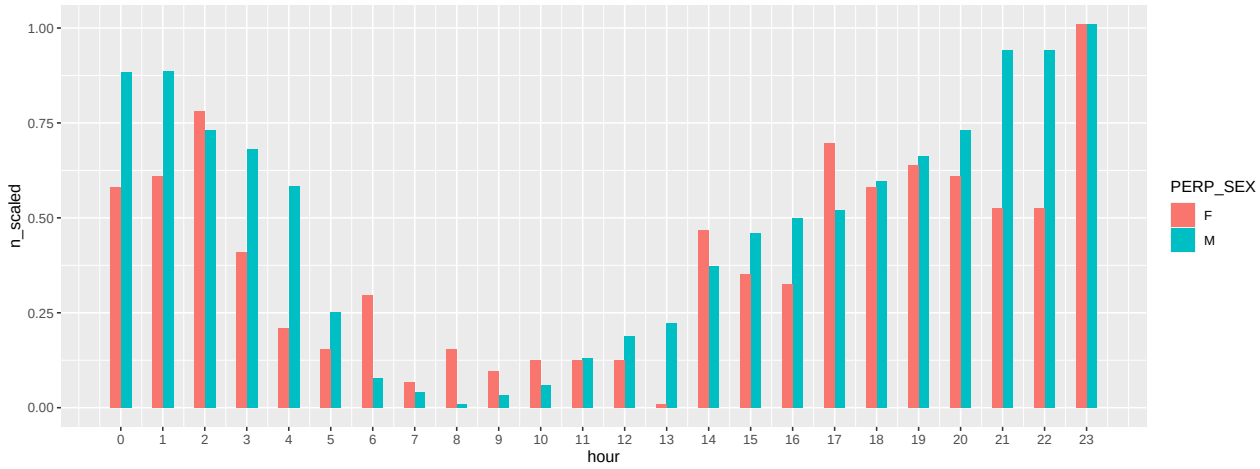
Based on visual inspection of above boxplot, it does not look like there's much of a difference.

However, it might be that crimes do not follow a normal distribution. And we perhaps run into trouble due to hour of day being cyclic (mod 24). We can continue our visual inspection by plotting male and female

crime times using a bar chart.

To do so, we first group by (hour, gender) and get the count for each group and scale for the genders (so that the hour with most for a gender becomes 1 and the one with least 0, or 0.01 for better visualization). Then we plot it.

```
count_by_sex <- data_w_hour %>% count(hour, PERP_SEX)

scale_001_to_1 <- function(x) {
  0.01 + (x - min(x)) / (max(x) - min(x)) # note 0.01 added to avoid 0 height bar
}

scaler <- function(x) {
  x["n_scaled"] = scale_001_to_1(x["n"])
  x
}

count_by_sex_w_scaled <- rbind(
  scaler(count_by_sex %>% filter(PERP_SEX == "M")),
  scaler(count_by_sex %>% filter(PERP_SEX == "F")))

ID <- 0:23 # Used to force all hours being shown
ggplot(count_by_sex_w_scaled, aes(hour, n_scaled, fill = PERP_SEX)) +
  geom_bar(stat="identity", position="dodge", width=0.5) +
  scale_x_continuous("hour", labels = as.character(ID), breaks = ID)
```

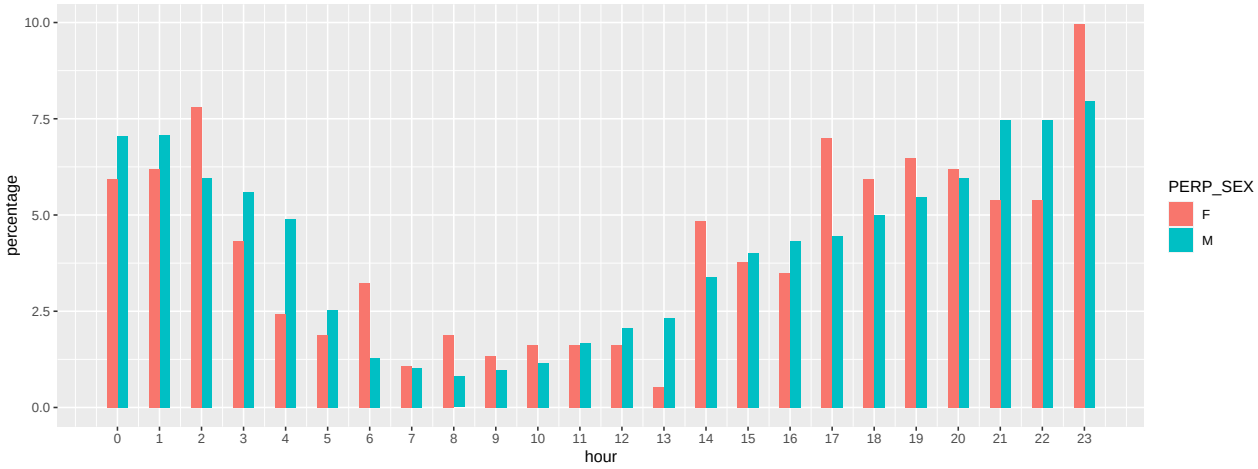


Now we can see a difference. For instance, women commit the fewest crimes around 13, whereas men commit the fewest crimes around 8.

However, it would be more readable to show percentage of crimes happening a specific hour for the gender, as this is easier to interpret. It works for us since the maximum percentage of crimes happening a specific hour is similar for men and women. We can update our scaler to rather calculate the percentage and plot again.

```
count_by_sex <- data_w_hour %>% count(hour, PERP_SEX)

calculate_percentage <- function(x) {
  (x / sum(x)) * 100
}

scaler <- function(x) {
  x["percentage"] = calculate_percentage(x["n"])
```

```
  x
}

count_by_sex_w_percentage <- rbind(
  scaler(count_by_sex %>% filter(PERP_SEX == "M")),
  scaler(count_by_sex %>% filter(PERP_SEX == "F")))

ID <- 0:23 # Used to force all hours being shown
ggplot(count_by_sex_w_percentage, aes(hour, percentage, fill = PERP_SEX)) +
  geom_bar(stat="identity", position="dodge", width=0.5) +
  scale_x_continuous("hour", labels = as.character(ID), breaks = ID)
```



We can see that the peak for female perpetrators is even greater at 23 compared to men.

To easier see when the most crimes are commited, we can sort the data:

```
only_female <- count_by_sex_w_percentage %>% filter(PERP_SEX == "F")
only_male <- count_by_sex_w_percentage %>% filter(PERP_SEX == "M")

only_female[order(only_female$percentage),] %>% map_df(rev)
```

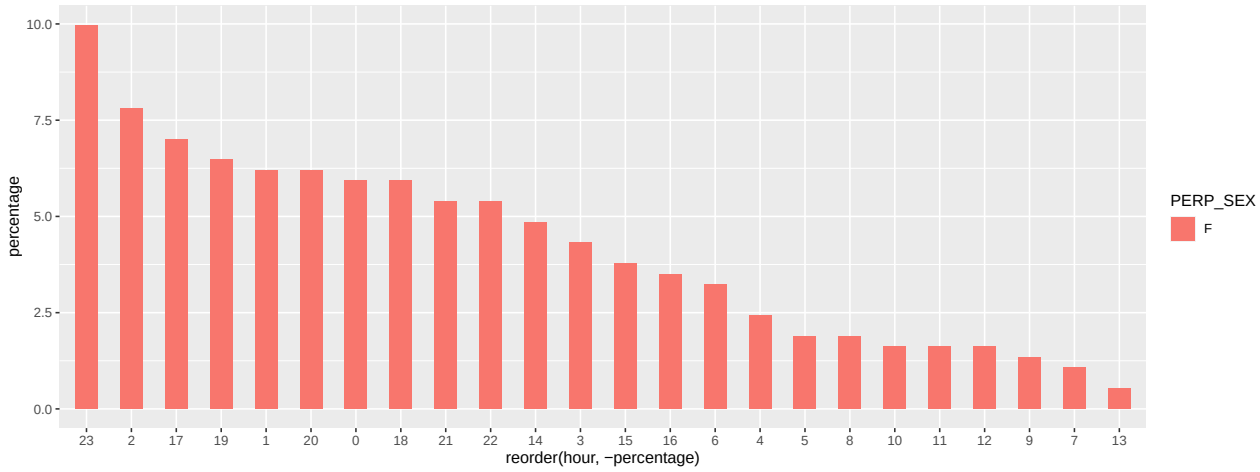
```
## # A tibble: 24 x 4
##     hour PERP_SEX     n percentage
##    <int> <chr>    <int>      <dbl>
##  1    23 F           37       9.97
##  2     2 F           29       7.82
##  3    17 F           26       7.01
##  4    19 F           24       6.47
##  5    20 F           23       6.20
##  6     1 F           23       6.20
##  7    18 F           22       5.93
##  8     0 F           22       5.93
##  9    22 F           20       5.39
## 10    21 F           20       5.39
## # ... with 14 more rows
```

```
only_male[order(only_male$percentage),] %>% map_df(rev)
```

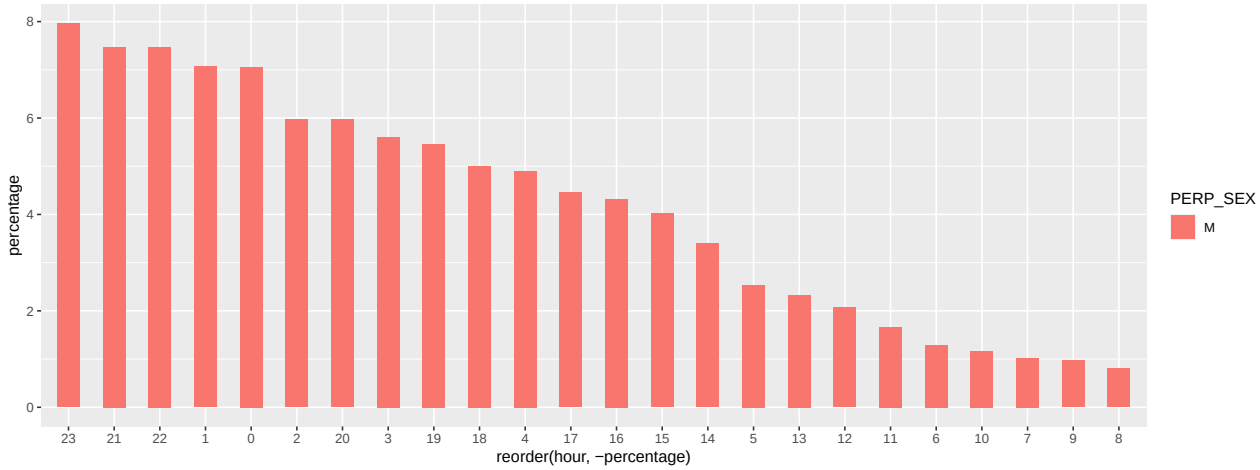
```
## # A tibble: 24 x 4
##     hour PERP_SEX     n percentage
##    <int> <chr>    <int>      <dbl>
```

```
##  1    23 M         1148       7.96
##  2    22 M         1078       7.48
##  3    21 M         1078       7.48
##  4     1 M         1019       7.07
##  5     0 M         1017       7.05
##  6    20 M          860       5.97
##  7     2 M          860       5.97
##  8     3 M          808       5.60
##  9    19 M          788       5.47
## 10    18 M          722       5.01
## # ... with 14 more rows
```

```
ggplot(only_female, aes(x = reorder(hour, -percentage), percentage, fill = PERP_SEX)) +
  geom_bar(stat="identity", position="dodge", width=0.5)
```



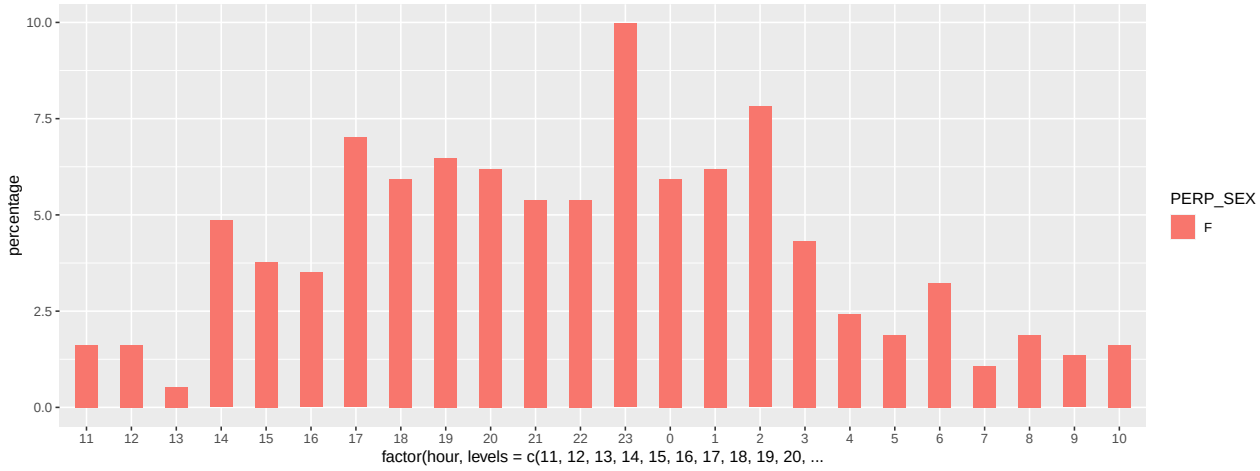

```
ggplot(only_male, aes(x = reorder(hour, -percentage), percentage, fill = PERP_SEX)) +
  geom_bar(stat="identity", position="dodge", width=0.5)
```



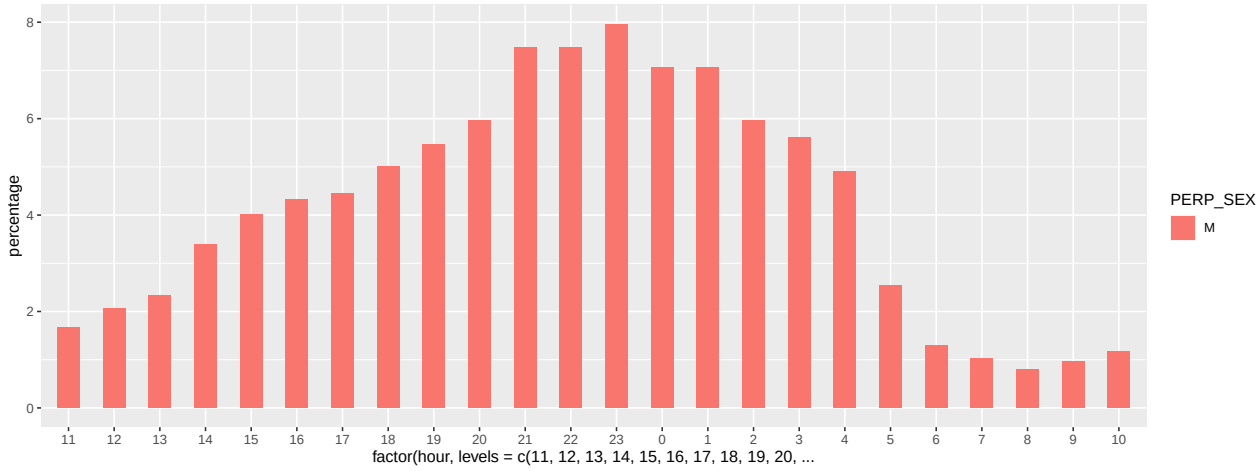
The tables and visualizations make it easy to see what hours of the day most crimes are committed by men and women.

Another interesting observation is that it looks like we have something that resembles a normal distribution with mean hour 23. To explorer this further, we can plot using a custom order where 23 is in the "middle".

```
ggplot(only_female, aes(factor(hour, levels=c(11,12,13,
                                              14,15,16,
                                              17,18,19,
                                              20,21,22,
                                              23,0,1,
                                              2,3,4,
                                              5,6,7,
                                              8,9,10)), percentage, fill = PERP_SEX)) +
  geom_bar(stat="identity", position="dodge", width=0.5)
```




```
ggplot(only_male, aes(factor(hour, levels=c(11,12,13,
                                            14,15,16,
                                            17,18,19,
                                            20,21,22,
                                            23,0,1,
                                            2,3,4,
                                            5,6,7,
                                            8,9,10)), percentage, fill = PERP_SEX)) +
  geom_bar(stat="identity", position="dodge", width=0.5)
```



This helps us see that there is a difference between men and women in when they commit crimes. Female perpetrators commit their crimes a bit more spread out, with the second and third peak at 2am and 5pm respectively. Men commit crimes more concentrated around 23 (the primary peak). Second and third peak is found at 9pm and 10pm for men.

We can model our data using a normal distribution, so we can tell the likelihood of a crime being committed by male and female perpetrators at a specific time.

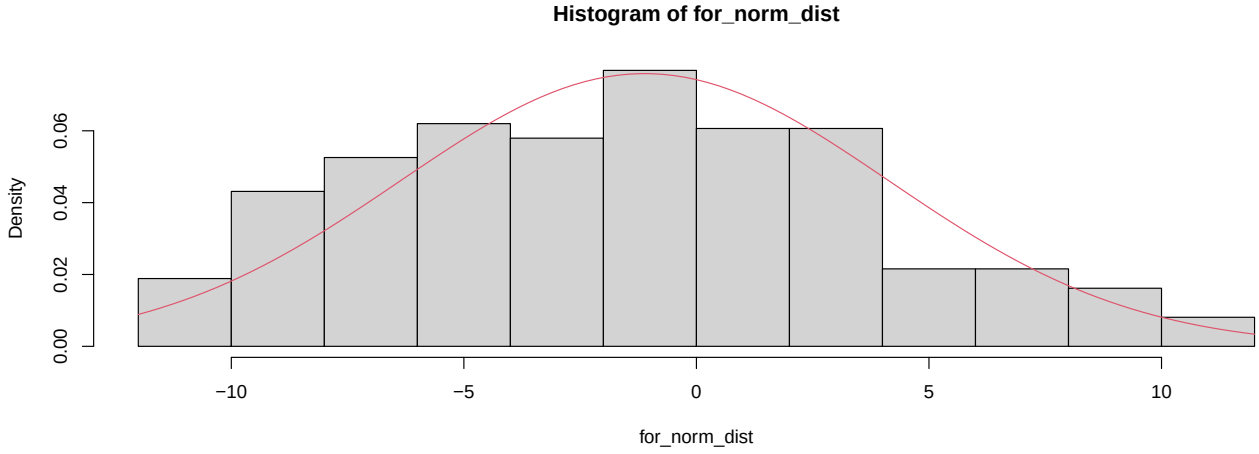
Model

To model as a normal distribution we must change how we represent the time of day. We need to reflect that there is one hour between 23 and 0 for instance.

An easy way is to take the offset from 23. So 23 becomes 0, 0 becomes 1, 22 becomes -1, etc. This will allow us to fit a normal distribution to the data, which we can then use for looking up the likelihood of a crime being committed.

For women:

```
only_female_w_centered_hour <- only_female %>%
  mutate(hour_centered = case_when(
    .$hour >= 11 & .$hour < 23 ~ .$hour - 23,
    .$hour >= 0 & .$hour < 11 ~ .$hour + 1,
    TRUE ~ 0
  ))

for_norm_dist <- only_female_w_centered_hour %>% uncount(n) %>% pull(hour_centered)

fit <- fitdistr(for_norm_dist, "normal")
ignore <- class(fit)

para <- fit$estimate

hist(for_norm_dist, prob = TRUE)
curve(dnorm(x, para[1], para[2]), col = 2, add = TRUE)
```



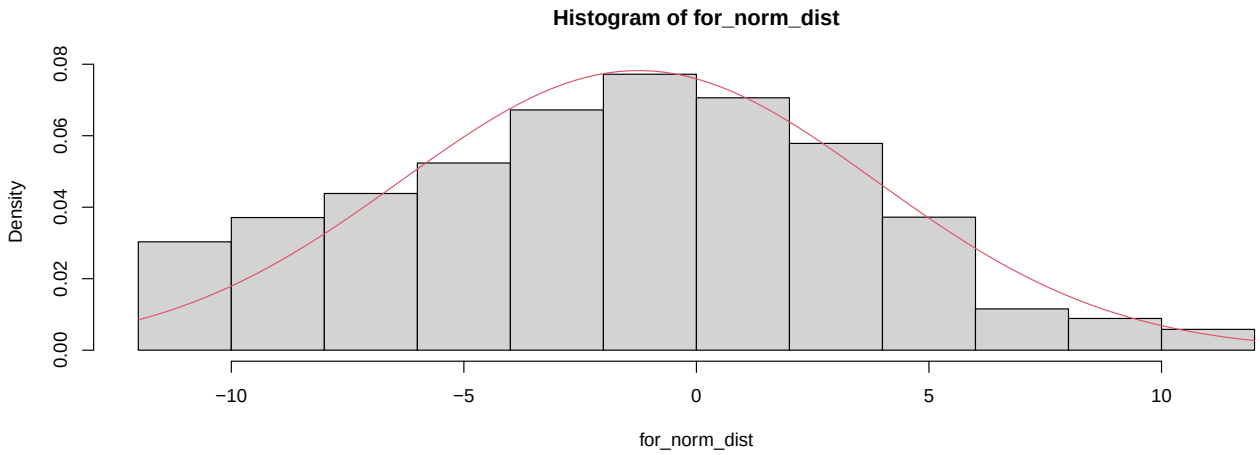

```
para
```

```
##      mean        sd
## -1.113208  5.254505
```

Same for men:

```
only_male_w_centered_hour <- only_male %>%
  mutate(hour_centered = case_when(
    .$hour >= 11 & .$hour < 23 ~ .$hour - 23,
    .$hour >= 0 & .$hour < 11 ~ .$hour + 1,
```

```
    TRUE ~ 0
  ))

for_norm_dist <- only_male_w_centered_hour %>% uncount(n) %>% pull(hour_centered)

fit <- fitdistr(for_norm_dist, "normal")
ignore <- class(fit)

para <- fit$estimate

hist(for_norm_dist, prob = TRUE)
curve(dnorm(x, para[1], para[2]), col = 2, add = TRUE)
```




```
para
```

```
##      mean        sd
## -1.247433  5.099267
```

A normal distribution does not fit perfect as the distribution is slightly biased to crimes not happening during morning (6-10), especially for men. We can try a Weibull model. This model does not support <= 0, so we must change our scale. We can do so by counting hours after 11 + 2. So 11 becomes 2 and 12 becomes 3, etc.

```
only_male_w_centered_hour <- only_male %>%
  mutate(hour_centered = case_when(
    .$hour >= 11 & .$hour <= 23 ~ 2 + .$hour - 11,
    .$hour >= 0 & .$hour < 11 ~ .$hour + 15
  ))

only_male_w_centered_hour
```

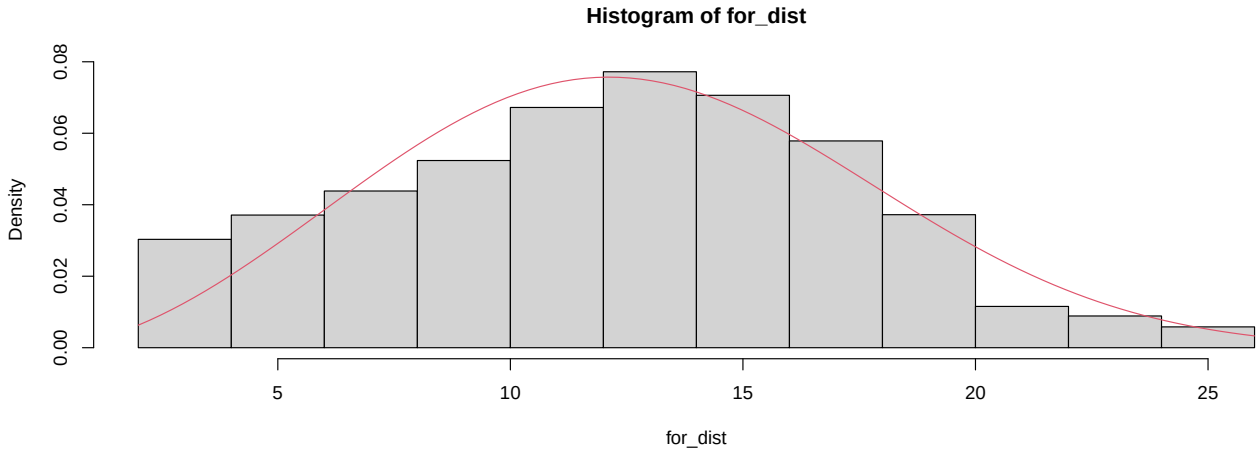
```
## # A tibble: 24 x 5
##     hour PERP_SEX     n percentage hour_centered
##    <int> <chr>    <int>      <dbl>         <dbl>
##  1     0 M         1017       7.05            15
##  2     1 M         1019       7.07            16
##  3     2 M          860       5.97            17
##  4     3 M          808       5.60            18
##  5     4 M          707       4.90            19
##  6     5 M          366       2.54            20
##  7     6 M          186       1.29            21
##  8     7 M          147       1.02            22
```

```
##  9     8 M          116      0.805            23
## 10     9 M          140      0.971            24
## # ... with 14 more rows
```

```
for_dist <- only_male_w_centered_hour %>% uncount(n) %>% pull(hour_centered)

fit <- fitdistr(for_dist, "weibull")
```

```
## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```
ignore <- class(fit)

para <- fit$estimate

hist(for_dist, prob = TRUE)
curve(dweibull(x, para[1], para[2]), col = 2, add = TRUE)
```




```
para
```

```
##     shape     scale
##  2.728533 14.324903
```

Both models are decent fits, but fail to capture that fewer crimes happen in the early morning hours (6-10).

Conclusion

Noteworthy is that the data contains a lot more incidents with male than female perpetrators. The data itself can not tell us whether this is due to catching more male than female perpetrators, or that men indeed commit more crimes. This is a sort of bias, as our model will be better for men. Another potential bias is that the data is likely based on traditional gender identities. Presumably, gender is the biological gender. It could be that we would get different results if we were to use gender that the perpetrator identify as.

The lower amount of data for female perpetrators means that the data for female perpetrators is less statistically robust. It might be influenced by outliers.

However, this analysis does suggest that there's a difference in the time distribution of crimes for male and

female perpetrators.

Neither a normal distribution or weibull distribution model are perfect fits for the data, but do capture the general trend.

SessionInfo

```
sessionInfo()
```

```
## R version 4.2.3 (2023-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Arch Linux
##
## Matrix products: default
## BLAS:   /usr/lib/libblas.so.3.11.0
## LAPACK: /usr/lib/liblapack.so.3.11.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] MASS_7.3-58.2    scales_1.2.1     lubridate_1.9.0  timechange_0.1.1
##  [5] forcats_0.5.2    stringr_1.5.0    dplyr_1.0.10     purrr_1.0.1
##  [9] readr_2.1.3      tidyr_1.3.0      tibble_3.1.8     ggplot2_3.4.0
## [13] tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] assertthat_0.2.1    digest_0.6.31       utf8_1.2.2
##  [4] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
##  [7] reprex_2.0.2        evaluate_0.19       highr_0.10
## [10] httr_1.4.4          pillar_1.8.1        rlang_1.1.0
## [13] googlesheets4_1.0.1 curl_4.3.3          readxl_1.4.1
## [16] rstudioapi_0.14     rmarkdown_2.19      labeling_0.4.2
## [19] googledrive_2.0.0   bit_4.0.5           munsell_0.5.0
## [22] broom_1.0.2         compiler_4.2.3      modelr_0.1.10
## [25] xfun_0.36           pkgconfig_2.0.3     htmltools_0.5.4
## [28] tidyselect_1.2.0    fansi_1.0.3         crayon_1.5.2
## [31] tzdb_0.3.0          dbplyr_2.2.1        withr_2.5.0
## [34] grid_4.2.3          jsonlite_1.8.4      gtable_0.3.1
## [37] lifecycle_1.0.3     DBI_1.1.3           magrittr_2.0.3
## [40] cli_3.5.0           stringi_1.7.8       vroom_1.6.0
## [43] farver_2.1.1        fs_1.5.2            xml2_1.3.3
## [46] ellipsis_0.3.2      generics_0.1.3      vctrs_0.6.1
## [49] tools_4.2.3         bit64_4.0.5         glue_1.6.2
## [52] hms_1.1.2           parallel_4.2.3      fastmap_1.1.0
## [55] yaml_2.3.6          colorspace_2.0-3    gargle_1.2.1
```

```
## [58] rvest_1.0.3         knitr_1.41          haven_2.5.1
```