Term Project-Analysis

Brynn Woolley

1. Setup and Load Data

```r
library(tidyverse)
library(sf)
library(lubridate)
library(here)
library(MASS)      # negative binomial
library(janitor)
library(patchwork)

set.seed(506)
```

```r
## Load cleaned datasets
collisions_york_2023_sf <- readRDS(
  here::here("Term Project", "Data", "collisions_york_2023_sf.rds")
)

collisions_york_multi_sf <- readRDS(
  here::here("Term Project", "Data", "collisions_york_2020_2024_sf.rds")
)

## Sanity checks
nrow(collisions_york_2023_sf)
```

```
[1] 314
```

```r
nrow(collisions_york_multi_sf)
```

```
[1] 1478
```

```
range(collisions_york_multi_sf$date)
```

```
[1] "2020-01-04" "2024-12-30"
```

2. Spatial and Temporal Aggregation

2.1 Construct Hex Grid

```
## Create bounding box
york_bbox <- st_bbox(collisions_york_multi_sf)


## Create hex grid (~2km spacing)
hex_grid <- st_make_grid(
  st_as_sfc(york_bbox),
  cellsize = 2000,
  square = FALSE
) %>%
  st_sf(hex_id = seq_along(.), geometry = .)


## Keep only hexes that intersect any crash (multi-year)
hex_grid <- hex_grid[
  st_intersects(hex_grid, collisions_york_multi_sf, sparse = FALSE) %>%
    apply(1, any),
]

## Quick check
nrow(hex_grid)
```

```
[1] 74
```

Hexagon size was selected to balance spatial resolution with sufficient crash counts per spatial unit, yielding 74 hexes across the City of York.

2.2 Aggregate to hex by month

```r
## ---- hex-month-aggregation ----

aggregate_hex_month <- function(collisions_sf, hex_grid) {

  ## Add month variable
  collisions_sf <- collisions_sf %>%
    mutate(month = floor_date(date, "month"))

  ## Define full month sequence for this dataset
  all_months <- seq(
    from = min(collisions_sf$month),
    to   = max(collisions_sf$month),
    by   = "month"
  )

  ## Spatial join to hex grid
  collisions_hex <- collisions_sf %>%
    st_join(hex_grid) %>%
    filter(!is.na(hex_id))

  ## Aggregate to hex × month and balance panel
  hex_month_counts <- collisions_hex %>%
    st_drop_geometry() %>%
    count(hex_id, month, name = "crash_count") %>%
    complete(
      hex_id,
      month = all_months,
      fill = list(crash_count = 0)
    ) %>%
    arrange(hex_id, month) %>%
    mutate(
      days_in_month = days_in_month(month),
      log_days = log(days_in_month)
    )

  list(
    hex_month = hex_month_counts,
    collisions_hex = collisions_hex
  )
}

## Aggregate 2023 data
```

```r
agg_2023 <- aggregate_hex_month(
  collisions_sf = collisions_york_2023_sf,
  hex_grid = hex_grid
)

hex_month_2023 <- agg_2023$hex_month
collisions_hex_2023 <- agg_2023$collisions_hex

## Aggregate multi-year data
agg_multi <- aggregate_hex_month(
  collisions_sf = collisions_york_multi_sf,
  hex_grid = hex_grid
)

hex_month_multi <- agg_multi$hex_month
collisions_hex_multi <- agg_multi$collisions_hex
```

3. Negative Binomial Models

```r
## ---- model-helpers ----

## Modal road category by hex-month
make_road_covs <- function(collisions_hex_df) {
  collisions_hex_df %>%
    st_drop_geometry() %>%
    filter(!is.na(road_cat)) %>%
    group_by(hex_id, month) %>%
    summarise(
      road_cat = names(sort(table(road_cat), decreasing = TRUE))[1],
      .groups = "drop"
    )
}

## Environmental indicators by hex-month (handle NA weather_bin safely)
make_env_covs <- function(collisions_hex_df) {
  collisions_hex_df %>%
    st_drop_geometry() %>%
    group_by(hex_id, month) %>%
    summarise(
      any_dark = any(light_bin == "Dark", na.rm = TRUE),
      any_bad_weather = any(weather_bin == "Other", na.rm = TRUE),
      .groups = "drop"
    )
}

## Attach covariates to the balanced hex-month panel
attach_covariates <- function(hex_month_df, collisions_hex_df) {
  road_covs <- make_road_covs(collisions_hex_df)
  env_covs  <- make_env_covs(collisions_hex_df)

  hex_month_df %>%
    left_join(road_covs, by = c("hex_id", "month")) %>%
    left_join(env_covs,  by = c("hex_id", "month")) %>%
    mutate(
      road_cat = replace_na(as.character(road_cat), "standard"),
      road_cat = factor(road_cat, levels = c("standard", "divided", "complex", "other")),
      any_dark = replace_na(any_dark, FALSE),
      any_bad_weather = replace_na(any_bad_weather, FALSE)
    )
}
```

3.1 2023 Analysis

```
## ---- models-2023 ----

df_2023 <- attach_covariates(hex_month_2023, collisions_hex_2023) %>%
  mutate(month_fe = factor(month))

nb_base_2023 <- glm.nb(
  crash_count ~ month_fe + offset(log_days),
  data = df_2023
)
summary(nb_base_2023)
```

```

Call:
glm.nb(formula = crash_count ~ month_fe + offset(log_days), data = df_2023,
    init.theta = 0.3839790263, link = log)

Coefficients:
```

```
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)        -3.984034   0.288819 -13.794   <2e-16 ***
month_fe2023-02-01 -0.003578   0.412960  -0.009    0.993    
month_fe2023-03-01 -0.033902   0.409856  -0.083    0.934    
month_fe2023-04-01 -0.277365   0.423029  -0.656    0.512    
month_fe2023-05-01 -0.143101   0.414681  -0.345    0.730    
month_fe2023-06-01  0.097328   0.405893   0.240    0.810    
month_fe2023-07-01  0.095310   0.404725   0.235    0.814    
month_fe2023-08-01 -0.182322   0.416532  -0.438    0.662    
month_fe2023-09-01 -0.423969   0.431429  -0.983    0.326    
month_fe2023-10-01 -0.143101   0.414681  -0.345    0.730    
month_fe2023-11-01 -0.036203   0.411355  -0.088    0.930    
month_fe2023-12-01 -0.567984   0.438546  -1.295    0.195    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.384) family taken to be 1)

    Null deviance: 428.79  on 623  degrees of freedom
Residual deviance: 424.05  on 612  degrees of freedom
AIC: 1183

Number of Fisher Scoring iterations: 1


              Theta:  0.3840 
          Std. Err.:  0.0582 

 2 x log-likelihood:  -1157.0210 
```

```r
nb_road_2023 <- glm.nb(
  crash_count ~ month_fe + road_cat + offset(log_days),
  data = df_2023
)
summary(nb_road_2023)
```

```

Call:
glm.nb(formula = crash_count ~ month_fe + road_cat + offset(log_days), 
    data = df_2023, init.theta = 0.3942407992, link = log)

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)    
```

```
(Intercept)        -4.04895    0.29067 -13.930   <2e-16 ***
month_fe2023-02-01  0.03897    0.41340   0.094    0.925    
month_fe2023-03-01 -0.01159    0.41165  -0.028    0.978    
month_fe2023-04-01 -0.26879    0.42323  -0.635    0.525    
month_fe2023-05-01 -0.07818    0.41440  -0.189    0.850    
month_fe2023-06-01  0.13983    0.40466   0.346    0.730    
month_fe2023-07-01  0.11897    0.40402   0.294    0.768    
month_fe2023-08-01 -0.11740    0.41626  -0.282    0.778    
month_fe2023-09-01 -0.41667    0.43240  -0.964    0.335    
month_fe2023-10-01 -0.16856    0.41683  -0.404    0.686    
month_fe2023-11-01  0.02871    0.41108   0.070    0.944    
month_fe2023-12-01 -0.50307    0.43828  -1.148    0.251    
road_catdivided     0.84638    0.71186   1.189    0.234    
road_catcomplex     0.99827    0.69836   1.429    0.153    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.3942) family taken to be 1)

    Null deviance: 433.78  on 623  degrees of freedom
Residual deviance: 425.18  on 610  degrees of freedom
AIC: 1183.3

Number of Fisher Scoring iterations: 1

              Theta:  0.3942 
          Std. Err.:  0.0604 

 2 x log-likelihood:  -1153.2610 
```

```r
nb_full_2023 <- glm.nb(
  crash_count ~ month_fe + road_cat + any_dark + any_bad_weather + offset(log_days),
  data = df_2023
)

summary(nb_full_2023)
```

```

Call:
glm.nb(formula = crash_count ~ month_fe + road_cat + any_dark + 
    any_bad_weather + offset(log_days), data = df_2023, init.theta = 3.104440524, 
    link = log)
```

```
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)         -5.44530    0.26657 -20.427  < 2e-16 ***
month_fe2023-02-01   0.63390    0.33845   1.873 0.061081 .
month_fe2023-03-01   0.70450    0.33815   2.083 0.037215 *
month_fe2023-04-01   0.14136    0.36228   0.390 0.696390
month_fe2023-05-01   0.98749    0.33937   2.910 0.003617 **
month_fe2023-06-01   1.26859    0.32948   3.850 0.000118 ***
month_fe2023-07-01   0.80009    0.32843   2.436 0.014847 *
month_fe2023-08-01   0.86529    0.33811   2.559 0.010492 *
month_fe2023-09-01   0.62329    0.35458   1.758 0.078778 .
month_fe2023-10-01   0.01785    0.34451   0.052 0.958682
month_fe2023-11-01   0.25077    0.34813   0.720 0.471327
month_fe2023-12-01  -0.09703    0.37824  -0.257 0.797538
road_catdivided      0.63855    0.43122   1.481 0.138660
road_catcomplex      0.96852    0.42353   2.287 0.022209 *
any_darkTRUE         1.91086    0.17401  10.982  < 2e-16 ***
any_bad_weatherTRUE  1.25459    0.19353   6.483 9.01e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(3.1044) family taken to be 1)

    Null deviance: 786.79  on 623  degrees of freedom
Residual deviance: 444.08  on 608  degrees of freedom
AIC: 961.15

Number of Fisher Scoring iterations: 1


              Theta:  3.10
          Std. Err.:  1.06
Warning while fitting theta: alternation limit reached

 2 x log-likelihood:  -927.147
```

```
AIC(nb_base_2023, nb_road_2023, nb_full_2023)
```

```
             df       AIC
nb_base_2023 13 1183.0210
nb_road_2023 15 1183.2607
nb_full_2023 17  961.1474
```

3.2 Multi-Year Models

```
## ---- models-multi ----

df_multi <- attach_covariates(hex_month_multi, collisions_hex_multi) %>%
  mutate(
    month_of_year = factor(month(month), levels = 1:12, labels = month.abb),
    year_fe = factor(year(month))
  )

nb_base_multi <- glm.nb(
  crash_count ~ month_of_year + year_fe + offset(log_days),
  data = df_multi
)
summary(nb_base_multi)
```

```

Call:
glm.nb(formula = crash_count ~ month_of_year + year_fe + offset(log_days),
    data = df_multi, init.theta = 0.2616666503, link = log)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      -4.54775    0.15576 -29.198   <2e-16 ***
month_of_yearFeb  0.07611    0.18844   0.404    0.686
month_of_yearMar -0.20992    0.19273  -1.089    0.276
month_of_yearApr -0.16777    0.19251  -0.872    0.383
month_of_yearMay -0.35321    0.19633  -1.799    0.072 .
month_of_yearJun -0.01505    0.18916  -0.080    0.937
month_of_yearJul -0.05962    0.18941  -0.315    0.753
month_of_yearAug -0.11953    0.19068  -0.627    0.531
month_of_yearSep -0.07855    0.19050  -0.412    0.680
month_of_yearOct -0.02715    0.18875  -0.144    0.886
month_of_yearNov  0.06615    0.18756   0.353    0.724
month_of_yearDec -0.11877    0.19066  -0.623    0.533
year_fe2021       0.11596    0.12566   0.923    0.356
year_fe2022       0.11073    0.12573   0.881    0.378
year_fe2023       0.17518    0.12488   1.403    0.161
year_fe2024       0.14427    0.12524   1.152    0.249
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for Negative Binomial(0.2617) family taken to be 1)

    Null deviance: 2365.4  on 4439  degrees of freedom
Residual deviance: 2355.1  on 4424  degrees of freedom
AIC: 6368

Number of Fisher Scoring iterations: 1


              Theta:  0.2617
          Std. Err.:  0.0173

 2 x log-likelihood:  -6333.9920
```

```r
nb_road_multi <- glm.nb(
  crash_count ~ month_of_year + year_fe + road_cat + offset(log_days),
  data = df_multi
)
summary(nb_road_multi)
```

```
Call:
glm.nb(formula = crash_count ~ month_of_year + year_fe + road_cat +
    offset(log_days), data = df_multi, init.theta = 0.2806931341,
    link = log)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)      -4.635773   0.154529 -29.999  < 2e-16 ***
month_of_yearFeb  0.045311   0.186609   0.243 0.808149
month_of_yearMar -0.201183   0.190695  -1.055 0.291428
month_of_yearApr -0.138904   0.190034  -0.731 0.464813
month_of_yearMay -0.345484   0.194329  -1.778 0.075432 .
month_of_yearJun -0.008633   0.186784  -0.046 0.963136
month_of_yearJul -0.051749   0.187078  -0.277 0.782074
month_of_yearAug -0.107475   0.188391  -0.570 0.568345
month_of_yearSep -0.122343   0.188997  -0.647 0.517420
month_of_yearOct -0.022452   0.186420  -0.120 0.904135
month_of_yearNov  0.036583   0.185720   0.197 0.843844
month_of_yearDec -0.126875   0.188745  -0.672 0.501453
year_fe2021       0.138468   0.124783   1.110 0.267142
year_fe2022       0.157975   0.124633   1.268 0.204969
```

```
year_fe2023       0.225422   0.123838   1.820 0.068715 .  
year_fe2024       0.149038   0.124364   1.198 0.230764    
road_catdivided   1.256937   0.337898   3.720 0.000199 ***
road_catcomplex   1.341102   0.255271   5.254 1.49e-07 ***
road_catother     1.214265   1.236154   0.982 0.325956    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(0.2807) family taken to be 1)

    Null deviance: 2440.2  on 4439  degrees of freedom
Residual deviance: 2378.8  on 4421  degrees of freedom
AIC: 6324.4

Number of Fisher Scoring iterations: 1


              Theta:  0.2807 
          Std. Err.:  0.0190 

 2 x log-likelihood:  -6284.4460 
```

```r
nb_full_multi <- glm.nb(
  crash_count ~ month_of_year + year_fe + road_cat + any_dark + any_bad_weather + offset(log_
  data = df_multi
)

summary(nb_full_multi)
```

```

Call:
glm.nb(formula = crash_count ~ month_of_year + year_fe + road_cat + 
    any_dark + any_bad_weather + offset(log_days), data = df_multi, 
    init.theta = 1.416843824, link = log)

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)    
(Intercept)         -5.71167    0.14306 -39.925  < 2e-16 ***
month_of_yearFeb     0.22042    0.16464   1.339  0.18064    
month_of_yearMar     0.23960    0.17046   1.406  0.15983    
month_of_yearApr     0.40845    0.16900   2.417  0.01566 *  
month_of_yearMay     0.50597    0.16937   2.987  0.00281 ** 
month_of_yearJun     0.82057    0.16115   5.092 3.54e-07 ***
```

```
month_of_yearJul     0.76068    0.16131   4.716 2.41e-06 ***
month_of_yearAug     0.67666    0.16227   4.170 3.05e-05 ***
month_of_yearSep     0.49745    0.16466   3.021  0.00252 ** 
month_of_yearOct     0.12513    0.16566   0.755  0.45003    
month_of_yearNov     0.03548    0.16676   0.213  0.83153    
month_of_yearDec    -0.31195    0.17258  -1.808  0.07068 .  
year_fe2021          0.18409    0.10580   1.740  0.08187 .  
year_fe2022          0.14548    0.10720   1.357  0.17474    
year_fe2023          0.08375    0.10742   0.780  0.43557    
year_fe2024          0.22499    0.10578   2.127  0.03343 *  
road_catdivided      0.92165    0.20704   4.451 8.53e-06 ***
road_catcomplex      1.20228    0.15754   7.632 2.32e-14 ***
road_catother        1.37244    0.76000   1.806  0.07094 .  
any_darkTRUE         2.25947    0.09101  24.827  < 2e-16 ***
any_bad_weatherTRUE  1.37117    0.10050  13.643  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(1.4168) family taken to be 1)

    Null deviance: 4110.7  on 4439  degrees of freedom
Residual deviance: 2575.0  on 4419  degrees of freedom
AIC: 5294.2

Number of Fisher Scoring iterations: 1

              Theta:  1.417 
          Std. Err.:  0.158 

 2 x log-likelihood:  -5250.178 
```

```
AIC(nb_base_multi, nb_road_multi, nb_full_multi)
```

```
              df      AIC
nb_base_multi 17 6367.992
nb_road_multi 20 6324.446
nb_full_multi 22 5294.178
```

3.3 Side by Side comparison

```r
## ---- irr-extraction-and-filtering ----

irr <- function(model) {
  coefs <- coef(summary(model))
  tibble(
    term = rownames(coefs),
    estimate = coefs[, "Estimate"],
    se = coefs[, "Std. Error"],
    irr = exp(estimate),
    irr_low = exp(estimate - 1.96 * se),
    irr_high = exp(estimate + 1.96 * se),
    p = coefs[, "Pr(>|z|)"]
  )
}

keep_terms <- c(
  "road_catdivided",
  "road_catcomplex",
  "any_dark",
  "any_bad_weather"
)

irr_2023 <- irr(nb_full_2023) %>%
  filter(term %in% keep_terms) %>%
  mutate(dataset = "2023")

irr_multi <- irr(nb_full_multi) %>%
  filter(term %in% keep_terms) %>%
  mutate(dataset = "2020-2024")

print(irr_2023)
```

```
# A tibble: 2 x 8
  term            estimate    se   irr irr_low irr_high      p dataset
  <chr>              <dbl> <dbl> <dbl>   <dbl>    <dbl>  <dbl> <chr>
1 road_catdivided    0.639 0.431  1.89   0.813     4.41 0.139  2023
2 road_catcomplex    0.969 0.424  2.63   1.15      6.04 0.0222 2023
```

```r
print(irr_multi)
```

```
# A tibble: 2 x 8
  term            estimate    se   irr irr_low irr_high        p dataset
  <chr>              <dbl> <dbl> <dbl>   <dbl>    <dbl>    <dbl> <chr>
1 road_catdivided    0.922 0.207  2.51    1.68     3.77 8.53e- 6 2020-2024
2 road_catcomplex    1.20  0.158  3.33    2.44     4.53 2.32e-14 2020-2024
```

4. Visualizations

```r
## ---- eda-time-series-prep ----

plot_df <- collisions_york_multi_sf %>%
  mutate(
    month = floor_date(date, "month"),
    light_simple = if_else(light_bin == "Daylight", "Daylight", "Dark"),
    road_simple = if_else(road_cat == "standard", "Standard", "Other")
  ) %>%
  st_drop_geometry()

p1 <- ggplot(plot_df, aes(x = month)) +
  geom_bar(width = 25, fill = "grey60") +
  scale_x_date(
    date_breaks = "1 year",
    labels = NULL
  ) +
  labs(
    title = "Total collisions by month",
    y = "Number of collisions",
    x = NULL
  ) +
  theme_minimal(base_size = 11) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank()
  )

p2 <- ggplot(plot_df, aes(x = month, fill = light_simple)) +
  geom_bar(width = 25) +
  scale_fill_manual(
    values = c("Daylight" = "#FDB863", "Dark" = "#5E3C99"),
    name = "Lighting"
  ) +
  scale_x_date(
    date_breaks = "1 year",
    labels = NULL
  ) +
  labs(
    title = "Collisions by lighting condition",
    y = "Number of collisions",
    x = NULL
  ) +
  theme_minimal(base_size = 11) +
  theme(
    axis.text.x = element_blank(),
    axis.ticks.x = element_blank(),
    legend.position = "bottom"
  )

p3 <- ggplot(plot_df, aes(x = month, fill = road_simple)) +
  geom_bar(width = 25) +
  scale_fill_manual(
    values = c("Standard" = "#4DAF4A", "Other" = "#E41A1C"),
    name = "Road type"
  ) +
  scale_x_date(
    date_breaks = "1 year",
    date_labels = "%Y"
  ) +
  labs(
    title = "Collisions by roadway type",
    y = "Number of collisions",
    x = "Year"
  ) +
  theme_minimal(base_size = 11) +
  theme(
    legend.position = "bottom"
  )

final_fig <- (p1 / p2 / p3) +
  plot_annotation(
    title = "Monthly road collisions in York (2020-2024)",
    theme = theme(
      plot.title = element_text(hjust = 0.5)
    )
  )

ggsave(
  filename = here::here("Term Project", "Figures", "monthly_collisions_york_2020_2024.png"),
  plot = final_fig,
  width = 10,
  height = 8,
  dpi = 300
)
```

GitHub Link

- Repo: https://github.com/brynnwoolley/STATS-506#
