HW 03 - Road traffic accidents

Tina Huynh

Contents

Figure 1: Photo by Clark Van Der Beken on Unsplash

In this assignment we'll look at traffic accidents in Edinburgh. The data are made available online by the UK Government. It covers all recorded accidents in Edinburgh in 2018 and some of the variables were modified for the purposes of this assignment.

1 Getting started

IMPORTANT: If there is no GitHub repo created for you for this assignment, it means I didn't have your GitHub username as of when I assigned the homework. Please let me know your GitHub username asap, and I can create your repo.

Go to the course GitHub organization and locate your homework repo, which should be named `hw-03-YOUR_GITHUB_USERNAME`. Grab the URL of the repo, and clone it in RStudio. First, open the R Markdown document `hw-03.Rmd` and Knit it. Make sure it compiles without errors. The output will be in the file markdown `.md` file with the same name.

1.1 Warm up

Before we introduce the data, let's warm up with some simple exercises.

- Update the YAML, changing the author name to your name, and **knit** the document.
- Commit your changes with a meaningful commit message.
- Push your changes to GitHub.
- Go to your repo on GitHub and confirm that your changes are visible in your Rmd **and** md files. If anything is missing, commit and push again.

1.2 Packages

We'll use the **tidyverse** package for much of the data wrangling and visualization and the data lives in the **dsbox** package. These packages are already installed for you. You can load them by running the following in your Console:

```r
library(tidyverse)
library(dsbox)
```

1.3 Data

The data can be found in the **dsbox** package, and it's called `accidents`. Since the dataset is distributed with the package, we don't need to load it separately; it becomes available to us when we load the package. You can find out more about the dataset by inspecting its documentation, which you can access by running `?accidents` in the Console or using the Help menu in RStudio to search for `accidents`. You can also find this information here.

2 Exercises

1. How many observations (rows) does the dataset have? Instead of hard coding the number in your answer, use inline code.

The dataset contains 768 observations.

Tired of typing your password? Chances are your browser has already saved your password, but if not, you can ask Git to save (cache) your password for a period of time, where you indicate the period of time in seconds. For example, if you want it to cache your password for 1 hour, that would be 3,600 seconds. To do so, run the following *in the Console*: `usethis::use_git_config(credential.helper = "cache --timeout=3600")`. If you want to cache it for a longer time, you can adjust the number of seconds in the code.

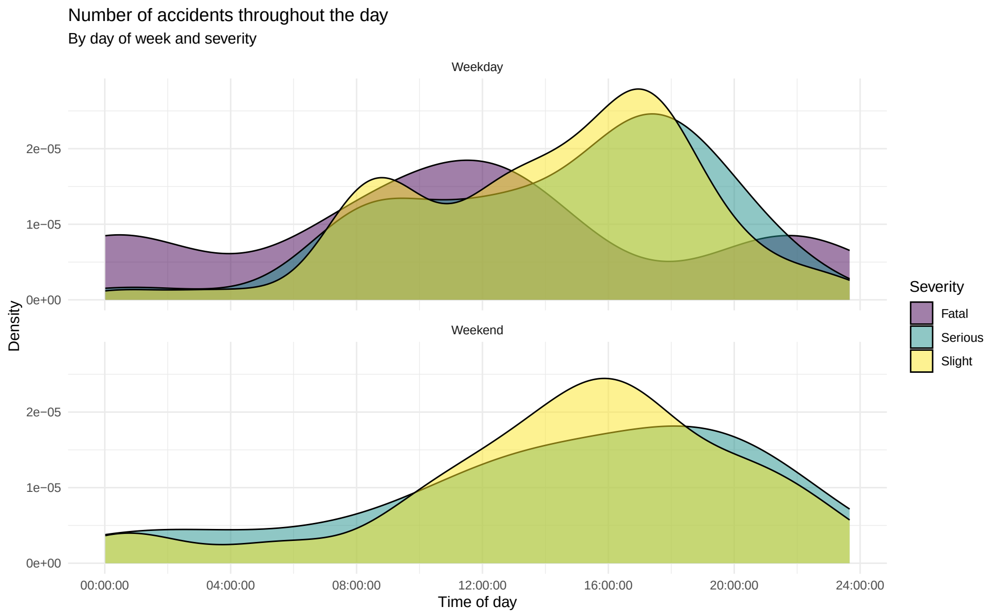
2. Run `View(accidents)` in your Console to view the data in the data viewer. What does each row in the dataset represent?

Each row in the dataset represents a distinct traffic accident that occurred in Edinburgh in 2018. Each accident entry includes detailed information about the time and location of the incident, weather and road conditions at the time, the severity of the accident, and various characteristics of the vehicles and casualties involved.

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

3. Recreate the following plot, and describe in context of the data. In your answer, don't forget to label your R chunk as well (where it says `label-me-1`). Your label should be short, informative, shouldn't include spaces, and shouldn't shouldn't repeat a previous label.

```r
accidents %>%
  mutate(day_of_week_type = if_else(day_of_week %in% c("Saturday", "Sunday"), "Weekend", "Weekday")) %>
  ggplot(aes(x = time, fill = severity)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~day_of_week_type, ncol = 1) +
  theme_minimal() +
  scale_fill_viridis_d() +
  labs(
    title = "Number of accidents throughout the day",
    subtitle = "By day of week and severity",
    x = "Time of day",
    y = "Density",
    fill = "Severity"
  )
```



The density plot above reveals critical temporal patterns in Edinburgh's traffic accidents, comparing weekday and weekend incidents by time of day and severity.

On weekdays, the data shows two prominent accident peaks: one during the morning rush hour (approximately 8-9 AM) and a larger, more sustained peak during the afternoon/evening commute (3-6 PM). This bimodal distribution strongly correlates with typical commuting patterns, suggesting that traffic congestion during work commutes significantly contributes to accident frequency. Notably, the afternoon peak is more pronounced and extends over a longer period, potentially reflecting the less synchronized nature of evening commutes compared to morning ones.

Weekend accident patterns differ substantially, with accidents more evenly distributed throughout the day and reaching their highest frequency in the afternoon and early evening hours. The absence of a morning peak on weekends reflects the reduced early-day traffic when commuting pressures are absent. Instead, the gradual increase in accidents throughout the day likely corresponds to increased leisure travel, shopping activities, and potentially social events later in the day.

Regarding severity, "Slight" accidents (indicated in purple) constitute the majority of incidents across both weekday and weekend periods. However, "Serious" accidents (shown in green-blue) maintain a relatively consistent proportion throughout the day. Fatal accidents, while rare, do not show a clear temporal pattern in this visualization, which is expected given their fortunately infrequent occurrence.

The overall lower density of weekend accidents compared to weekday incidents suggests that commuter traffic significantly influences accident rates in Edinburgh. This comprehensive temporal analysis provides valuable insights for traffic management authorities, who might consider targeted safety measures during high-risk periods, particularly weekday commuting hours when accident densities reach their peak.

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards.*

4. Create another data visualization based on these data and interpret it. You can choose any variables and any type of visualization you like, but it must have at least three variables, e.g. a scatterplot of x vs. y isn't enough, but if points are colored by z, that's fine. In your answer, don't forget to label your R chunk as well (where it says `label-me-2`).

```r
# First, examine the speed limit and weather distributions
accidents %>%
  count(speed_limit) %>%
  arrange(desc(n))
```

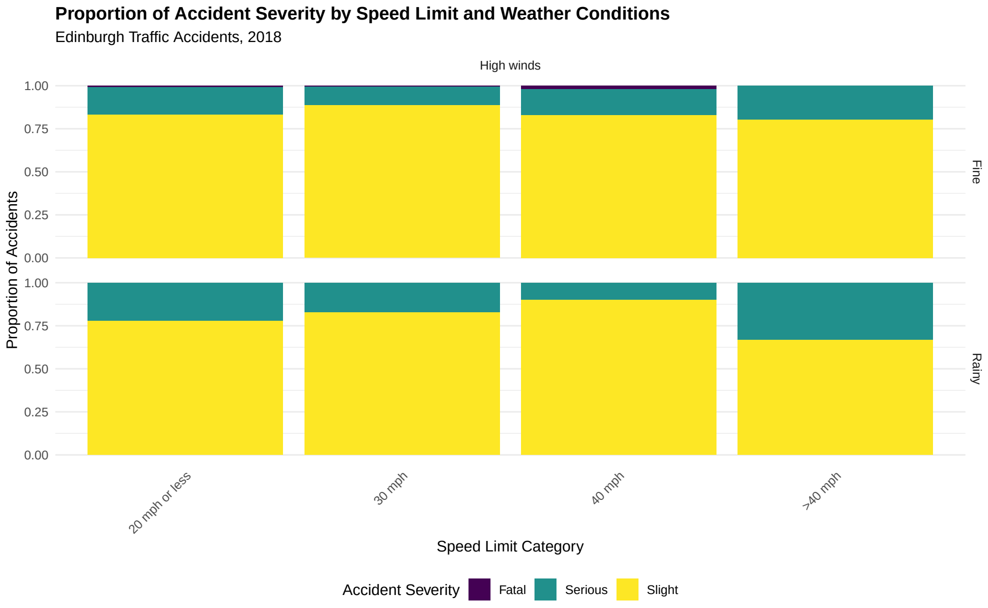
```
## # A tibble: 6 x 2
##   speed_limit     n
##         <dbl> <int>
## 1          20   379
## 2          30   246
## 3          40    64
## 4          70    47
## 5          50    19
## 6          60    13
```

```r
accidents %>%
  count(weather) %>%
  arrange(desc(n))
```

```
## # A tibble: 9 x 2
##   weather                       n
##   <fct>                     <int>
## 1 Fine + no high winds        618
## 2 Raining + no high winds      65
## 3 Unknown                      29
## 4 Other                        15
## 5 Fine + high winds            14
## 6 Raining + high winds       13
## 7 Snowing + no high winds    11
## 8 Snowing + high winds        2
## 9 Fog or mist                 1
```

```
# Filter out the very infrequent categories for clearer visualization
accidents_filtered <- accidents %>%
  filter(weather %in% c("Fine + no high winds", "Raining + no high winds", "Fine + high winds", "Raining
  mutate(
    weather_type = case_when(
      str_detect(weather, "Fine") ~ "Fine",
      str_detect(weather, "Rain") ~ "Rainy",
      TRUE ~ weather
    ),
    wind_condition = if_else(str_detect(weather, "high winds"), "High winds", "No high winds"),
    speed_category = case_when(
      speed_limit <= 20 ~ "20 mph or less",
      speed_limit <= 30 ~ "30 mph",
      speed_limit <= 40 ~ "40 mph",
      TRUE ~ ">40 mph"
    ),
    speed_category = factor(speed_category, levels = c("20 mph or less", "30 mph", "40 mph", ">40 mph")
  )

# Create visualization of weather conditions, speed limits, and accident severity
ggplot(accidents_filtered, aes(x = speed_category, fill = severity)) +
  geom_bar(position = "fill") +
  facet_grid(weather_type ~ wind_condition) +
  scale_fill_viridis_d() +
  theme_minimal() +
  labs(
    title = "Proportion of Accident Severity by Speed Limit and Weather Conditions",
    subtitle = "Edinburgh Traffic Accidents, 2018",
    x = "Speed Limit Category",
    y = "Proportion of Accidents",
    fill = "Accident Severity"
  ) +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1),
    plot.title = element_text(face = "bold"),
    panel.grid.major.x = element_blank(),
    legend.position = "bottom"
  )
```



This visualization examines the relationship between speed limits, weather conditions, and accident severity in Edinburgh's 2018 traffic incidents, revealing several noteworthy patterns with significant road safety implications.

The analysis reveals that roads with higher speed limits consistently demonstrate greater proportions of serious and fatal accidents across all weather conditions. This progressive relationship is particularly evident in the 40 mph and >40 mph categories, where the proportion of severe outcomes increases substantially compared to lower-speed zones. This pattern aligns with established traffic safety principles that higher speeds generate greater kinetic energy during collisions, resulting in more severe injuries.

Weather conditions exhibit a complex influence on accident severity. Fine weather without high winds shows relatively consistent severity distributions across speed categories, likely reflecting "baseline" accident patterns. However, the introduction of high winds significantly alters this pattern, particularly in higher speed zones. In 40 mph zones during fine weather with high winds, the proportion of serious accidents increases noticeably compared to the same speed zones without high winds. This suggests that wind conditions may compromise vehicle stability or driver control, especially at higher speeds.

Rainy conditions demonstrate an interesting pattern: they generally show higher proportions of slight accidents in the 30 mph zones compared to fine weather. This counterintuitive finding may reflect adaptive behavior, where drivers exercise greater caution during visible adverse conditions, potentially reducing accident severity despite increased occurrence. However, when rain combines with high winds in 20 mph or less zones, we observe a notably higher proportion of serious accidents, indicating that this combination of adverse conditions may be particularly hazardous in city centers and residential areas with lower speed limits.

The data shows few fatal accidents across all categories, making pattern identification difficult for this severity level. However, their relative absence in the 20 mph or less zones across all weather conditions reinforces the safety benefits of lower urban speed limits.

These findings have significant implications for traffic management policy. They suggest that speed limit reductions could be particularly beneficial during adverse weather conditions, and that warning systems for high wind conditions might be especially important on higher-speed roadways. The data also supports the continued implementation of 20 mph zones in urban areas, which consistently show the lowest proportion of serious and fatal accidents across all weather conditions analyzed.

Knit, *commit, and push your changes to GitHub with an appropriate commit message. Make sure to commit and push all changed files so that your Git pane is cleared up afterwards and review the md document* *on GitHub to make sure you're happy with the final state of your work.* Once you decide that you are done with the lab, choose the knit drop down and select `Knit to tufte_handout` to generate a pdf. Download and submit that pdf to Canvas.