Response Time

Mandy Langlois

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becon
```

```r
library(lubridate)
```

Load the data from the CSV.

```r
data = read_csv(here::here("Fire_incidents.csv"))
```

```
Rows: 229047 Columns: 20
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (14): incident_number, incident_type_description, arrive_date_time, clea...
dbl  (6): X, Y, OBJECTID, incident_type, exposure, station

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.
```

```r
head(data)
```

```
# A tibble: 6 x 20
      X     Y OBJECTID incident_number incident_type incident_type_description
  <dbl> <dbl>    <dbl> <chr>                   <dbl> <chr>
1 -78.6  35.9   474765 07-0031665                 NA NULL
2 -78.7  35.8   474766 08-0017918                 NA NULL
3 -78.6  35.8   474767 08-0032426                 NA NULL
4 -78.6  35.8   474768 07-0023051                444 Power line down
5 -78.5  35.9   474769 13-0030918                150 Outside rubbish fire Other
6 -78.6  35.9   474770 14-0004846                510 Person in distress Other
# i 14 more variables: arrive_date_time <chr>, cleared_date_time <chr>,
#   dispatch_date_time <chr>, exposure <dbl>, platoon <chr>, station <dbl>,
#   address <chr>, address2 <chr>, apt_room <chr>, GlobalID <chr>,
#   CreationDate <chr>, Creator <chr>, EditDate <chr>, Editor <chr>
```

Make sure the dates and times are in the right format before any calculations.

Calculate the response times.

```r
dispatch_date_time = data |>
  mutate(dispatch_date_time = ymd_hms(dispatch_date_time), arrive_date_time = ymd_hms(arrive_


average_response_time = dispatch_date_time |>
  mutate(response_time = arrive_date_time - dispatch_date_time) |>
  summarize(response_time = mean(response_time, na.rm=T))

average_response_time
```

```
# A tibble: 1 x 1
  response_time
  <drtn>
1 318.7497 secs
```

Find the average response time by station.

```r
response_by_station = dispatch_date_time |>
  mutate(response_time = arrive_date_time - dispatch_date_time) |>
  group_by(station) |>
  summarize(response_time = mean(response_time, na.rm=T))

highest_response_time = response_by_station |>
  filter(response_time == max(response_time, na.rm=T))

lowest_response_time = response_by_station |>
  filter(response_time == min(response_time, na.rm=T))

response_by_station
```

```
# A tibble: 31 x 2
   station response_time
     <dbl> <drtn>
 1       0 450.0000 secs
 2       1 246.7053 secs
 3       2 353.0248 secs
 4       3 284.8110 secs
 5       4 326.0335 secs
 6       5 273.7687 secs
 7       6 311.8862 secs
 8       7 284.7177 secs
 9       8 333.7066 secs
10       9 291.6525 secs
# i 21 more rows
```

```r
highest_response_time
```

```
# A tibble: 1 x 2
  station response_time
    <dbl> <drtn>
1      29 495.764 secs
```

```r
lowest_response_time
```

```
# A tibble: 1 x 2
  station response_time
    <dbl> <drtn>
1      13 223 secs
```

Finding the station with the highest and lowest average response over time.

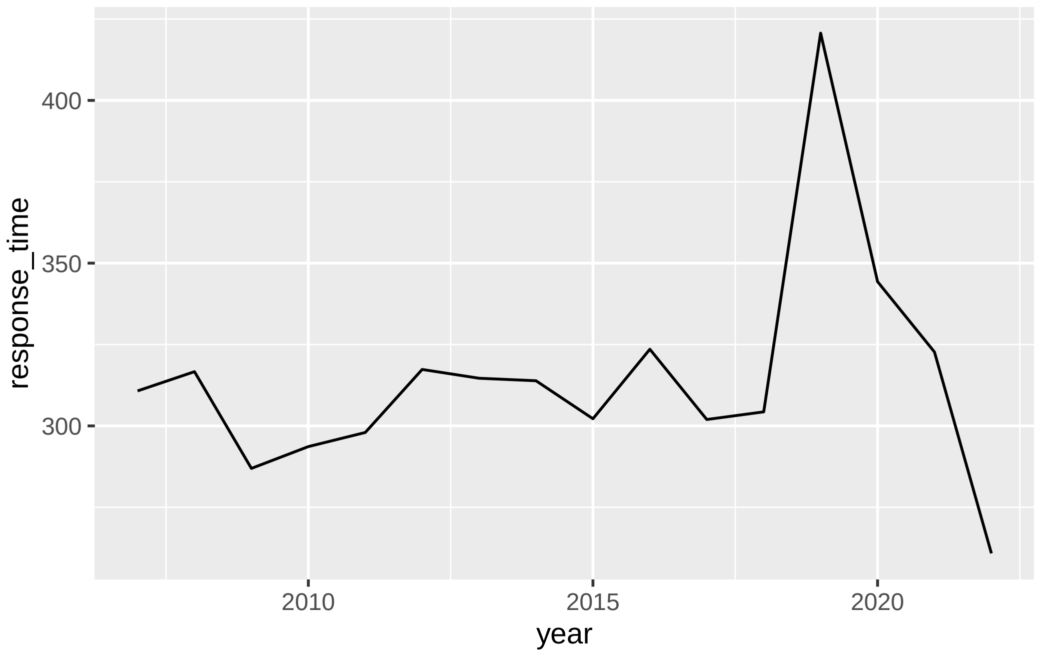
```r
response_over_time = data |>
  mutate(dispatch_date_time = ymd_hms(dispatch_date_time), arrive_date_time = ymd_hms(arrive_

response_over_time = response_over_time |>
  mutate(response_time = arrive_date_time - dispatch_date_time)

response_over_time = response_over_time |>
  mutate(year = year(dispatch_date_time))

response_by_year = response_over_time |>
  group_by(year) |>
  summarize(response_time = mean(response_time, na.rm=T ))

response_by_year
```

```
# A tibble: 17 x 2
    year response_time
   <dbl> <drtn>
 1  2007 310.7516 secs
 2  2008 316.6669 secs
 3  2009 286.9414 secs
 4  2010 293.6406 secs
 5  2011 297.9799 secs
 6  2012 317.3400 secs
 7  2013 314.6415 secs
 8  2014 313.8668 secs
 9  2015 302.1926 secs
10  2016 323.5524 secs
11  2017 301.9666 secs
12  2018 304.3143 secs
13  2019 420.6703 secs
14  2020 344.3216 secs
15  2021 322.6979 secs
16  2022 260.8378 secs
17    NA      NaN secs
```

```r
ggplot(response_by_year, aes(x=year, y=response_time)) +
  geom_line()
```

```
Don't know how to automatically pick scale for object of type <difftime>.
Defaulting to continuous.
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_line()`).
```



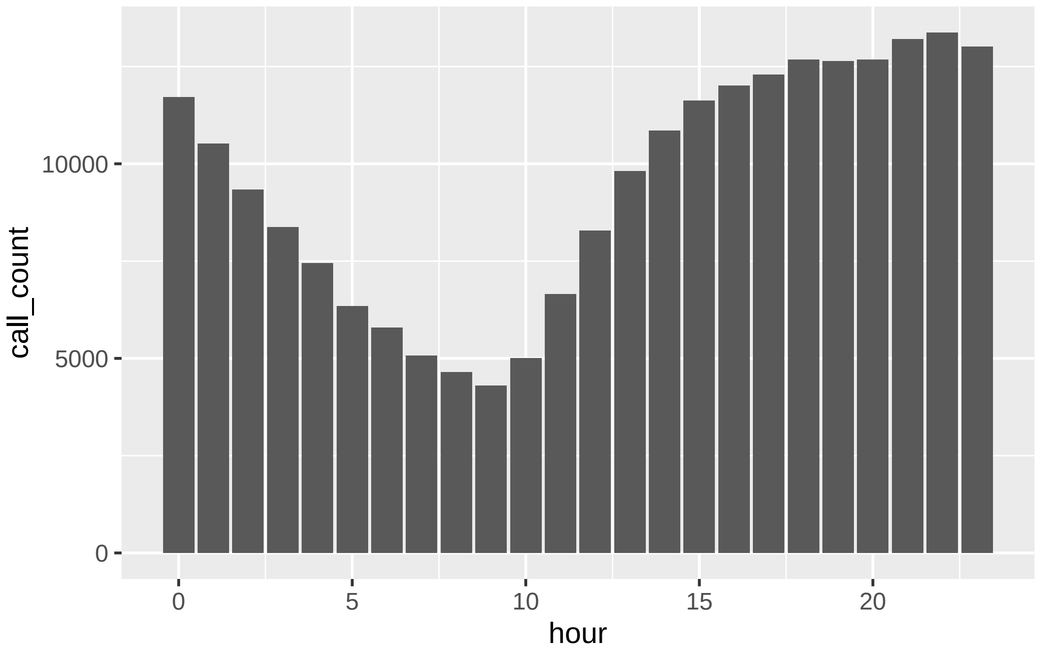
Calculating the times of day fire calls are most likely to occur.

```
fire_calls_by_time = data |>
  mutate(dispatch_date_time = ymd_hms(dispatch_date_time))

fire_calls_by_time = fire_calls_by_time |>
  mutate(hour = hour(dispatch_date_time))

calls_by_hour = fire_calls_by_time |>
group_by(hour) |>
  summarize(call_count = length(dispatch_date_time))

calls_by_hour
```

```
# A tibble: 25 x 2
    hour call_count
   <int>      <int>
 1     0      11714
 2     1      10512
 3     2       9340
 4     3       8366
 5     4       7449
 6     5       6337
 7     6       5785
 8     7       5065
 9     8       4640
10     9       4303
# i 15 more rows
```

```
ggplot(calls_by_hour, aes(x=hour, y=call_count)) +
  geom_col()
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```



Calculating how many calls are recorded in this dataset.

figure out how many of them are actual fires.

```
fire_calls = data |>
  filter(incident_type >= 100, incident_type < 200)

total_fire_calls = length(data$incident_type)

actual_fires = sum(!is.na(fire_calls$incident_type))

total_fire_calls
```

```
[1] 229047
```

```
actual_fires
```

```
[1] 17231
```

Determine the average response time to actual fires.

Determine if the response time is faster than the average response time for all incidents.

```
dispatch_date_time_fires = data |>
   filter(incident_type >= 100, incident_type < 200) |>
   mutate(dispatch_date_time = ymd_hms(dispatch_date_time), arrive_date_time = ymd_hms(arrive


average_response_time_fires = dispatch_date_time_fires |>
  mutate(response_time = arrive_date_time - dispatch_date_time) |>
  summarize(average_response_time_fires = mean(response_time, na.rm=T))

dispatch_date_time_all = data |>
   mutate(dispatch_date_time = ymd_hms(dispatch_date_time), arrive_date_time = ymd_hms(arrive
  filter(!is.na(dispatch_date_time), !is.na(arrive_date_time))

  average_response_time_all = dispatch_date_time_all |>
    mutate(response_time = arrive_date_time - dispatch_date_time) |>
    summarize(average_response_time_all = mean(response_time, na.rm=T))

  average_response_time_fires
```

```
# A tibble: 1 x 1
  average_response_time_fires
  <drtn>
1 310.9837 secs
```

```
  average_response_time_all
```

```
# A tibble: 1 x 1
  average_response_time_all
  <drtn>
1 318.7497 secs
```

Calculating response time by station for actual fires.

Determine which station have the highest and lowest average response time for actual fires.

```
dispatch_date_time_fires = data |>
  filter(incident_type >= 100, incident_type < 200) |>
  mutate(dispatch_date_time = ymd_hms(dispatch_date_time), arrive_date_time = ymd_hms(arrive_
  filter(!is.na(dispatch_date_time), !is.na(arrive_date_time)) |>
  mutate(response_time = arrive_date_time - dispatch_date_time)

average_response_time_by_station_fires = dispatch_date_time_fires |>
    group_by(station) |>
    summarize(average_response_time = mean(response_time, na.rm=T))

highest_response_time_station_fires = average_response_time_by_station_fires |>
  filter(average_response_time == max(average_response_time, na.rm=T))

lowest_response_time_station_fires = average_response_time_by_station_fires |>
  filter(average_response_time == min(average_response_time, na.rm=T))

average_response_time_by_station_fires
```

```
# A tibble: 29 x 2
   station average_response_time
     <dbl> <drtn>
 1       1 250.0814 secs
 2       2 439.1138 secs
 3       3 232.7666 secs
 4       4 276.9760 secs
 5       5 271.0142 secs
 6       6 289.4477 secs
 7       7 240.8027 secs
 8       8 295.1253 secs
 9       9 446.2697 secs
10      10 278.3915 secs
# i 19 more rows
```

```
highest_response_time_station_fires
```

```
# A tibble: 1 x 2
  station average_response_time
    <dbl> <drtn>
1      23 586.3713 secs
```

```
lowest_response_time_station_fires
```

```
# A tibble: 1 x 2
  station average_response_time
    <dbl> <drtn>
1       3 232.7666 secs
```

Determine if wake forest county fire response times have been going up or down for actual fires.

determine the reason behind the changes.

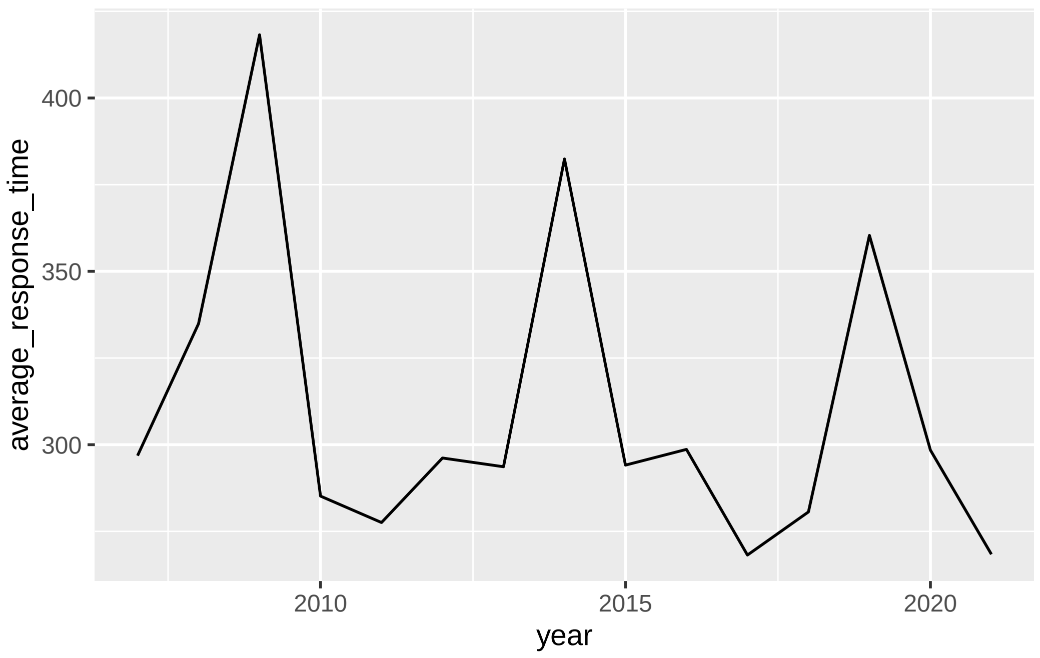
```
dispatch_date_time = data |>
  filter(incident_type >= 100, incident_type < 200) |>
  mutate(dispatch_date_time = ymd_hms(dispatch_date_time), arrive_date_time = ymd_hms(arrive_
  filter(!is.na(dispatch_date_time),!is.na(arrive_date_time)) |>
  mutate(response_time = arrive_date_time - dispatch_date_time)
```

```r
dispatch_date_time_fires = dispatch_date_time_fires |>
  mutate(year = year(dispatch_date_time))

average_response_time_by_year = dispatch_date_time_fires |>
  group_by(year) |>
  summarize(average_response_time = mean(response_time, na.rm=T))

average_response_time_by_year
```

```
# A tibble: 15 x 2
    year average_response_time
   <dbl> <drtn>
 1  2007 296.8353 secs
 2  2008 334.9061 secs
 3  2009 418.2293 secs
 4  2010 285.1706 secs
 5  2011 277.5476 secs
 6  2012 296.1646 secs
 7  2013 293.6396 secs
 8  2014 382.4182 secs
 9  2015 294.1111 secs
10  2016 298.6482 secs
11  2017 268.1704 secs
12  2018 280.6117 secs
13  2019 360.3776 secs
14  2020 298.4371 secs
15  2021 268.3878 secs
```

```r
ggplot(average_response_time_by_year, aes(x=year, y=average_response_time)) +
  geom_line()
```

```
Don't know how to automatically pick scale for object of type <difftime>.
Defaulting to continuous.
```



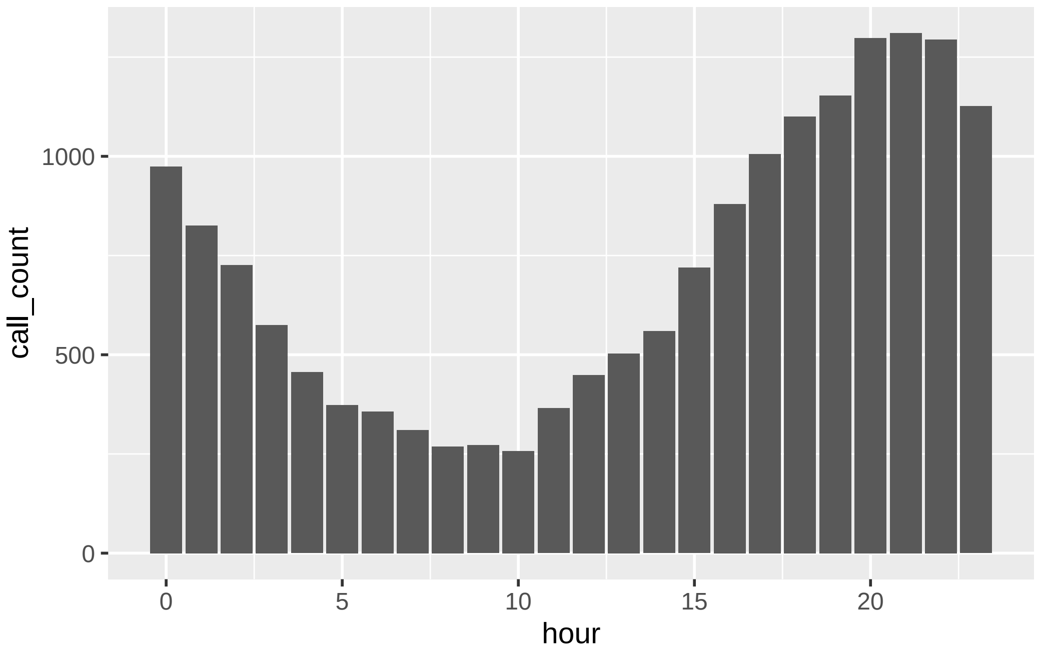
Determine what times of day are fire calls most likely occur.

```
fire_calls = data |>
  filter(incident_type >= 100, incident_type < 200) |>
  mutate(dispatch_date_time = ymd_hms(dispatch_date_time))

fire_calls = fire_calls |>
  mutate(hour = hour(dispatch_date_time))

calls_by_hour = fire_calls |>
  group_by(hour) |>
  summarize(call_count = length(dispatch_date_time))

calls_by_hour
```

```
# A tibble: 25 x 2
    hour call_count
   <int>      <int>
 1     0        974
 2     1        826
 3     2        726
 4     3         575
 5     4         456
 6     5         373
 7     6         357
 8     7         310
 9     8         269
10     9         272
# i 15 more rows
```

```r
ggplot(calls_by_hour, aes(x=hour, y=call_count)) +
  geom_col()
```

```
Warning: Removed 1 row containing missing values or values outside the scale range
(`geom_col()`).
```



Final write up

Q1. According to the table above, on average it takes 318.7497 seconds for Wake County Fire to respond to incidents.

Q2. The response times do vary across all stations, station 29 has the highest response time with a response time of 495.764 secs. Station 13 has the lowest response time with 223 secs.

Q3. Wake County Fire's response time has been been going down over time, down to less than 300 secs after 2020. this may be due to an improvement in technology, an increase in personnel could also be a factor in reducing the response time.

Q4. Fire calls are most likely to occur from 21:00-23:00 hours.

Q5. 229047 calls are to Wake County Fire are recorded in this dataset, and 17231 of those calls are actual fires.

Q6. The average response time for actual fires is 310.9837 secs, the average response time for all incidents is 318.7497. Therefore, the response time to actual fires averages was faster.

Q7.2 The average response time across all station varies. Station 23 had the highest average response time with 586.3713 secs, station 3 had the lowest average response time with 232.7666 secs.

Q7.3 Over time the average response time for wake county fire has been going down, this decline may be due to improved infrastructures that lead to better route options and a reduction in traffic.

Q7.4 Fire calls are most likely to occur for actual fires from 19:00-23:00, these fires could be happening because of faulty or unattended equipment, technology, everyone charges their phones/tablets before they go to sleep not caring about the dangers that may come with such practices.

Repository link:

https://github.com/ashmandy/plan372-hw2.git