# Problem Set 7

name?!

date?!

For the first part of this problem set we will work with motor vehicle crash data from New York City. You can read more about this publicly available data set on their website.

## Part 1

### Questions 1 - 5

The data is called "Motor_Vehicle_Collisions_Crashes". We want you to perform the following:

1. Rename the column names to lower-case and replace spaces with an underscore.
2. Select only:
    - crash_date
    - number_of_persons_injured
    - contributing_factor_vehicle_1
    - vehicle_type_code_1
3. Drop all rows with an NA value
4. Lower case the vehicle_type_code_1 variable and replace spacess with a dash.
5. Filter the data for vehicles that have a count/appear in the data set 500 times or more
    - Hints: group_by(), mutate(), n(), filter()
6. Calculate the percentage by vehicle

We have grouped the questions below to push you to perform commands with less code.

**Questions 1-3**

```
df_motor <- df_motor %>%
  # lower case and remove spaces
  rename_with(~ tolower(gsub(" ","_", .x, fixed=TRUE))) %>%
  # select certain columns
  select(crash_date, crash_time,
         number_of_persons_injured,
         contributing_factor_vehicle_1,
         vehicle_type_code_1) %>%
  # drop NA rows
  drop_na()
```

**Questions 4-5**

```
# lower case vehicles and add dash between spaces
df_motor <- df_motor %>%
  mutate(vehicle_type_code_1 =
           gsub(" ", "-", ignore.case=T, tolower(vehicle_type_code_1))) %>%
  # organize by vehicles
  group_by(vehicle_type_code_1) %>%
  # create a variable for counts
  mutate(count = n()) %>%
  # filter counts > 500
  filter(count > 500)
```

**Question 6**

```
# calculate percentage by vehicle
df_motor %>%
  group_by(vehicle_type_code_1) %>%
  summarize(count = n(),
            perc = count/nrow(df_motor))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 13 x 3
##    vehicle_type_code_1                count    perc
##    <chr>                              <int>   <dbl>
##  1 ambulance                            692 0.00375
##  2 bike                                1825 0.00989
##  3 box-truck                           3830 0.0208
##  4 bus                                 2862 0.0155
##  5 convertible                          577 0.00313
##  6 dump                                 543 0.00294
##  7 motorcycle                          1214 0.00658
##  8 pick-up-truck                       5411 0.0293
##  9 sedan                              85181 0.461
## 10 station-wagon/sport-utility-vehicle 71728 0.389
## 11 taxi                                8104 0.0439
## 12 tractor-truck-diesel                1434 0.00777
## 13 van                                 1177 0.00638
```

## Part 2

### Question 7 / Challenge

Use the readxl library and load two data sets from the "two_data_sheets" file. There's a parameter that you can specify which sheet to load. In this case, we have data about rat reaction time in sheet 1 and home visits in sheet 2.

```
library(readxl)
df_rats <- read_excel("../data/two_data_sheets.xlsx", 1)
df_home <- read_excel("../data/two_data_sheets.xlsx", 2)
```

**Question 8**

For the rats data, pivot the data frame from wide to long format. We want the 1, 2, 3 columns, which represent the amount of cheese placed in a maze, to transform into a column called "cheese". The values in the cheese column will be the time, which represents the amount of time the rat took to complete the maze. Please use the head() function to print your data frame.

```
# convert from wide to long
df_rats$subject <- factor(df_rats$subject)
df_rats_long <- df_rats %>%
  pivot_longer(c(`1`, `2`, `3`), names_to = "cheese", values_to = "time")

head(df_rats_long)
```

```
## # A tibble: 6 x 3
##   subject cheese  time
##   <fct>   <chr>  <dbl>
## 1 rat_101 1      14.4
## 2 rat_101 2       9.01
## 3 rat_101 3       8.20
## 4 rat_102 1      11.7
## 5 rat_102 2       8.59
## 6 rat_102 3       8.49
```

### Question 9

Compute the mean and standard deviation of the maze time.

```
df_rats_long %>%
  # organize by amount of cheese
  group_by(cheese) %>%
  # summarize
  summarize(mean = mean(time), # mean function
            sd = sd(time))     # standard deviation function
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 3
##   cheese  mean    sd
##   <chr>  <dbl> <dbl>
## 1 1      12.8  1.43
## 2 2       9.88 0.904
## 3 3       8.51 0.279
```

### Question 10

Last one (we promise). With the home visits data, pivot the data frame from long to wide. We want the names from the action column to become unique columns and the values to represent the counts. Please use the head() function to print your data frame.

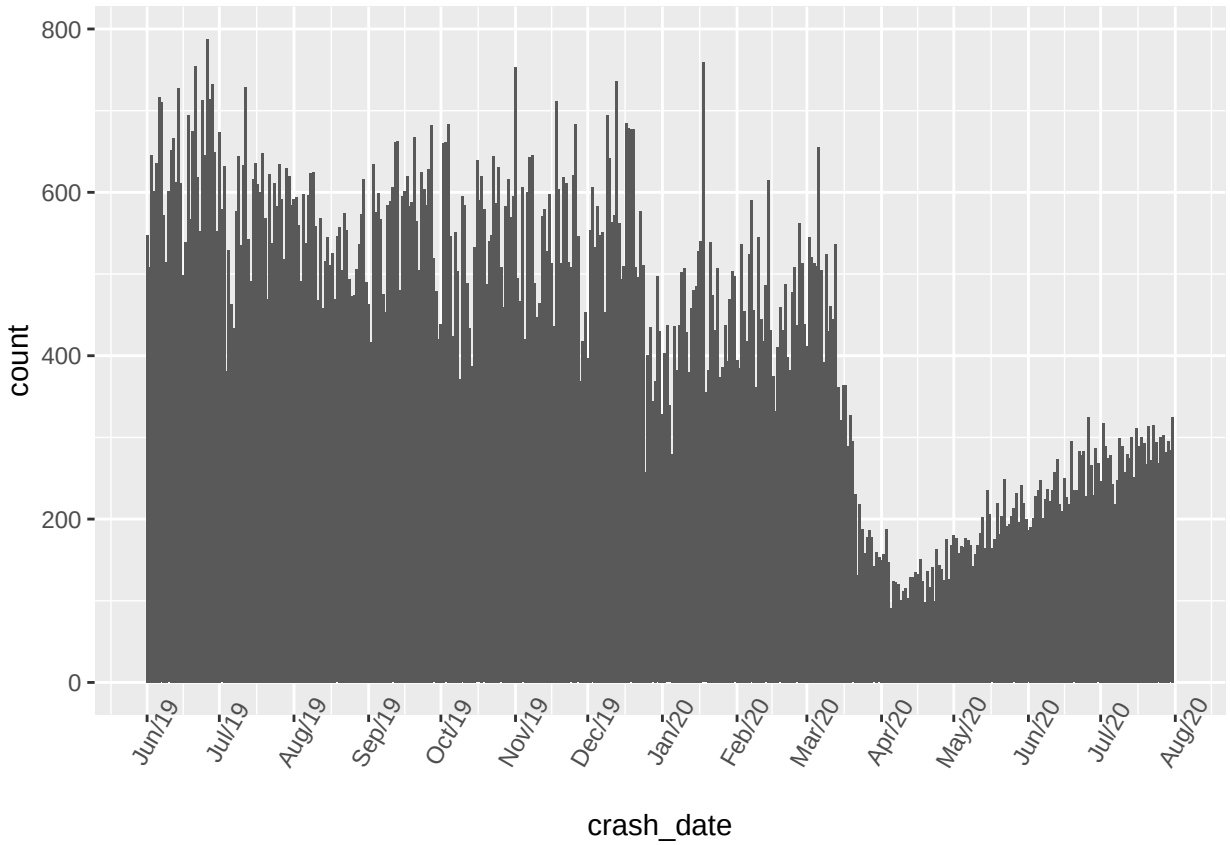
```
# pivot from long to wide
df_home_wide <- df_home %>%
  pivot_wider(names_from = action, values_from = count)

head(df_home_wide)
```

```
## # A tibble: 6 x 5
##   location       year interview `home visit` questionnaire
##   <chr>         <dbl>     <dbl>        <dbl>         <dbl>
## 1 Washington DC  2015       103           76           200
## 2 Washington DC  2016        71           43           168
## 3 Washington DC  2017        45           60            90
## 4 St Louis       2015        90           86           210
## 5 St Louis       2016        95           82           175
## 6 St Louis       2017        78           71           106
```

```
# visualize crashes by date
df_motor %>%
  mutate(crash_date = mdy(crash_date)) %>%
  group_by(crash_date) %>%
  summarize(count = n()) %>%
  ggplot(aes(x = crash_date, y = count)) +
  geom_col() +
  scale_x_date(date_breaks = "1 month", date_labels="%b/%y") +
  theme(axis.text.x = element_text(angle=60))
```

**not including**

```
## `summarise()` ungrouping output (override with `.groups` argument)
```


800
600
400
200
0
count
Jun/19
Jul/19
Aug/19
Sep/19
Oct/19
Nov/19
Dec/19
Jan/20
Feb/20
Mar/20
Apr/20
May/20
Jun/20
Jul/20
Aug/20
crash_date