PHW251 Problem Set 7

Teaching Team

Part 1

For part 1 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.

The data file is called "Motor_Vehicle_Collisions_Crashes.csv". 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 that contain an NA value.
4. Make the values in the vehicle_type_code_1 variable all lowercase and replace the spaces with a dash.
5. Filter the data for vehicles that have a count of at least 500 (appear in the data set 500 times or more)
    - Hints: group_by(), mutate(), n(), filter()
6. Calculate the percentage of accidents by vehicle type
7. Which vehicle group accounted for 1.55% (0.0155) of the accidents?

We have grouped the questions below to push you to perform commands with less code. As you're building your code we recommend going line by line to test, then combining to perform multiple steps in one command.

Questions 1-3

```
# YOUR CODE HERE

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

dim(df_motor)
```

```
## [1] 188989      4
```

```
head(df_motor)
```

```
## # A tibble: 6 x 4
##   crash_date number_of_persons_inju~1 contributing_factor_~2 vehicle_type_code_1
##   <chr>                         <dbl> <chr>                  <chr>
## 1 9/11/19                           0 Unspecified            Sedan
## 2 12/7/19                           1 Unspecified            Sedan
## 3 12/7/19                           0 Passing or Lane Usage~ Sedan
## 4 12/7/19                           1 Unsafe Speed           Sedan
## 5 12/7/19                           0 Passing or Lane Usage~ Sedan
## 6 12/9/19                           0 Oversized Vehicle      Ambulance
## # i abbreviated names: 1: number_of_persons_injured,
## #   2: contributing_factor_vehicle_1
```

Questions 4-5

```
# YOUR CODE HERE

# 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)

head(df_motor)
```

```
## # A tibble: 6 x 5
## # Groups:   vehicle_type_code_1 [2]
##   crash_date number_of_persons_inju~1 contributing_factor_~2 vehicle_type_code_1
##   <chr>                         <dbl> <chr>                  <chr>
## 1 9/11/19                           0 Unspecified            sedan
## 2 12/7/19                           1 Unspecified            sedan
## 3 12/7/19                           0 Passing or Lane Usage~ sedan
## 4 12/7/19                           1 Unsafe Speed           sedan
## 5 12/7/19                           0 Passing or Lane Usage~ sedan
## 6 12/9/19                           0 Oversized Vehicle      ambulance
## # i abbreviated names: 1: number_of_persons_injured,
## #   2: contributing_factor_vehicle_1
## # i 1 more variable: count <int>
```

```
min(df_motor$count)
```

```
## [1] 543
```

```
unique(df_motor$vehicle_type_code_1)
```

```
##  [1] "sedan"                                "ambulance"
##  [3] "taxi"                                 "station-wagon/sport-utility-vehicle"
##  [5] "motorcycle"                           "box-truck"
##  [7] "pick-up-truck"                        "van"
##  [9] "tractor-truck-diesel"                 "bike"
## [11] "dump"                                 "bus"
## [13] "convertible"
```

Question 6

```
# YOUR CODE HERE

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

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

Question 7

WRITE YOUR ANSWER HERE

Buses account for 1.55% of the accidents.

- count: 2862
- perc: 0.0155

Please note, if you try to filter for where perc == 0.0155, you will not get the correct answer unless you round perc to the same number of digits first.

Part 2

For this part we will work with four tables that are relational to each other. The following keys link the tables together:

- patient_id: patients, schedule
- visit_id: schedule, visits
- doctor_id: visits, doctors

Question 8

You've been asked to collect information on patients who are actually on the schedule. To start this task, you need to join the patient data to the schedule data, since we only want to keep the observations that are present in both the patient data AND the schedule data.

Which kind of join do you use?

WRITE YOUR ANSWER HERE **inner join**

How many observations do you see in your joined data set? Notice that some patients have multiple visits.

```r
# YOUR CODE HERE

# inner join by patient_id
inner.join.patient <- patients %>%
  inner_join(schedule, by = "patient_id")

head(inner.join.patient)
```

```
## # A tibble: 6 x 8
##   patient_id   age race_ethnicity   gender_identity height weight visit_id date
##        <dbl> <dbl> <chr>            <chr>            <dbl>  <dbl>    <dbl> <chr>
## 1       1000    54 Asian            woman              163     57       17 7/5/~
## 2       1001    60 Hispanic, Latin~ woman              190     80        1 1/2/~
## 3       1001    60 Hispanic, Latin~ woman              190     80       37 2/7/~
## 4       1001    60 Hispanic, Latin~ woman              190     80       53 8/3/~
## 5       1001    60 Hispanic, Latin~ woman              190     80       80 3/7/~
## 6       1001    60 Hispanic, Latin~ woman              190     80       83 4/7/~
```

WRITE YOUR ANSWER HERE **124 observations**

Question 9

In the visits data, we have a variable called "follow_up" where Y means a follow-up is needed and N means a follow-up is not needed. How many patients require a follow-up? You will want to first make a join and then subset. Start with the data frame created in the previous question.

```
# YOUR CODE HERE

left.follow.up <- inner.join.patient %>%
  left_join(visits, by = "visit_id")

# two ways we can filter:
follow.up <- left.follow.up %>% filter(follow_up == "Y")
follow.up <- left.follow.up[which(left.follow.up$follow_up == "Y"), ]

# make sure we count unique patients who need follow-up
length(unique(follow.up$patient_id))
```

```
## [1] 27
```

```
# or
follow.up %>% tally()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    27
```

Which join did you use?

WRITE YOUR ANSWER HERE **left join**

How many patients need a follow-up?

WRITE YOUR ANSWER HERE **27**

In this instance, there are actually multiple join types that will give you the same answer due to the question and how the data is structured. However, this doesn't apply to all join scenarios!

```
# Can get the same answer with an inner join

inner.follow.up <- inner.join.patient %>%
  inner_join(visits, by = "visit_id")

follow.up <- inner.follow.up %>% filter(follow_up == "Y")

# make sure we count unique patients who need follow-up
length(unique(follow.up$patient_id))
```

```
## [1] 27
```

```r
# Can get the same answer with a right join

right.follow.up <- inner.join.patient %>%
  right_join(visits, by = "visit_id")

follow.up <- right.follow.up %>% filter(follow_up == "Y")

# make sure we count unique patients who need follow-up
length(unique(follow.up$patient_id))
```

```
## [1] 27
```

Question 10

Which doctors do these patients need follow-up with? Print out each doctor's name.

```
# YOUR CODE HERE

doctors.contact <- follow.up %>%
  left_join(doctors, by = "doctor_id")

unique(doctors.contact$doctor)
```

```
##  [1] "Ariadne Anthony"   "Millie Albert"     "Ellesha Castaneda"
##  [4] "Bea Frame"         "Vera Irwin"        "Cade Gale"
##  [7] "Estelle Landry"    "Wiktoria Travis"   "Huzaifa Chung"
## [10] "Jamie-Lee Wilder"  "Jeremy Camacho"    "Daanyaal Griffin"
## [13] "Ammar Phelps"      "Rabia Browning"    "Amritpal Goodman"
## [16] "Merlin Jacobs"     "Tudor Moran"
```

Which join did you use?

WRITE YOUR ANSWER HERE **left join**