Homework 3: Further Cleaning

Econ 245

Overview

In this assignment, you will be using use-of-force (UOF) data from the Louisville Metro Police. UOF reports are filed whenever a police officer uses force in a confrontation with a civilian. Below is a description of some of the most important columns:

- `date_of_occurrence` - the date the UOF incident occurred.
- `time_of_occurrence` - the time the UOF incident occurred.
- `citizen_injured` - whether the citizen was injured in the incident.
- `service_rendered` - how the police officer was called to the scene.
- `reason_for_force` - why force was used.
- `force_used_{number}` - the type of force used.
- `force_used_{number}_effective` - whether the force was effective.
- `name_badge_number` - the officer name and badge number.
- `badge_number` - only the badge number of the officer.

To Receive Credit

- Save the scripting file (i.e. your R program file) as assignment_3.R. Make sure your capitalization is correct as the autograder is case-sensitive.
- Make sure all changes to the original dataset are done within the R script.

Things that can break the autograder:

- Please read the `autograder_instructions.pdf` file on CANVAS.

Part 1: Coding Assignment

Remember to start your answer with the following code:

```
rm(list=ls()) # clear the environment
setwd(dirname(rstudioapi::getSourceEditorContext()$path))

#-------Import necessary packages here--------------------#
# Note: Using other packages than the ones listed below can break the autograder
library(tidyverse)
library(lubridate)
library(dplyr)
library(janitor)

#------ Uploading PERMID --------------------------------#
PERMID <- "AD1273BD" #Type your PERMID within the quotation marks
PERMID <- as.numeric(gsub("\\D", "", PERMID)) #Don't touch
set.seed(PERMID) #Don't touch

#------ Answer ------------------------------------------#
```

For each homework assignment, words colored in magenta indicate a variable/vector/tibble that will be graded by the autograder. Pay close attention to these colored texts and be sure not to miss any.

1. Read in the UOF data using `read_csv`. Save this tibble as uof.
2. For this question, we will be analyzing when UOF incidents most frequently occur. To do this, the `count` and `mutate` functions will be helpful.
    a) Using the uof tibble, create a new column called `hour` which is the hour of the `time_of_occurrence` column. This can be done using the `lubridate::hour` function. Once this has been done, find the hour that is the most frequent for UOF reports. Save this hour number in a vector named frequent_hour. This vector should have 1 element, and that element should be an integer (not a string!). **Note:** Make sure to not update the uof tibble.
    b) Using a similar stragety to 2a, create a new column called `month` which is the month of the `time_of_occurrence` column. Then, find the month that has the *least* amount of UOF incidents. Save this month in a vector with the name least_frequent_month. It will be helpful to use the `lubridate::month` function. **Note:** Make sure to not update the uof tibble.
    c) Using a similar strategy to 2a and 2b, create a new column named `day` which is the day of the `time_of_occurrence` column. Then, find the day of the week (i.e. Mon/Tue/Wed etc.) that has the most number of UOF reports. Save this day in a vector named most_frequent_day. **Hint:** The `lubridate::wday` function will be helpful here. **Note:** Once again, make sure to not update the uof tibble.
    d) Let's take a look at the distribution of days within a month to see if there is any relationship as to when UOF incidents happen most. For instance, it may be the case that police officers have more instances of UOF near the end of the month if they are interacting with civilians more to reach a quota. While a bar graph would be most easy to analyze, we are going to create a tibble containing totals since graphics have not yet been covered in depth. To do this, you will follow a process similar to 2a/2b/2c, except you will use the `lubridate::day` function. The goal is to create a tibble similar to Table 1 except with all 31 days. You will need to use the `janitor::adorn_totals` function to create totals as the last row of your tibble. Hence, the final tibble should have 32 rows and 3 columns. Save this tibble as day_distribution. Note that column order matters.

Table 1: Example of table to be created in Question 2d.

day	n	fraction
17	133	0.0490775
16	112	0.0413284
12	110	0.0405904
24	107	0.0394834
28	103	0.0380074
Total	565	0.2084871

3. This question will involve analyzing the types of force used in UOF incidents.

   a) Using the uof tibble, find all unique categories of `force_used_1`. Save this as a vector named force_used_1.
   b) Using the uof tibble, find all unique categories of `force_used_2`. Save this as a vector named force_used_2.
   c) It is likely that the two vectors created in 3a and 3b have values unique to themselves (i.e., one vector contains something the other does not have). In this question, we'd like to extract *all* the unique types of forced used from `force_used_1`, `force_used_2`,. . . ,`force_used_8` and put them into a vector. To do this, you will need to complete the following code and save it as all_force. Note that this code is taking all distinct values from multiple columns, transposing these values (with the `t` function) and then turning all of these values into a vector (with the `c` function):

```
uof %>%
  distinct() %>% ## edit this line
  t() %>% ## do not edit this line
  c() %>% ## do not edit this line
  ## put one final function here
```

   d) We will now define a new vector named violent_force, which will be the violent UOF tactics used. Copy the following code:

```
## copy and run this line of code
violent_force <- c("take down", "hobble", "ecw cartridge deployed", "knee strike(s)",
                   "12 ga. sock round", "take-down", "impact weapon",
                   "kick", "deadly force used")
```

   e) Using the uof tibble, create a new column named `violent_uof_1` which is an binary variable equal to 1 if `force_used_1` contains any of the values from the vector violent_force and 0 otherwise. Remember to update the uof tibble to reflect these changes.
   f) Now, recreate Table 2. To do this, you will need to do several steps in a long pipe. First filter to include only violent UOF incidents. Next you will need to use the `count`, `mutate`, and `janitor::adorn_totals` functions to complete the table. Save this table as violent_force_service_table. Note that column order matters.

Table 2: Example of table to be created in Question 3f.

service_rendered	n	fraction
call for service	323	0.5086614
effecting an arrest	109	0.1716535
self initiated	46	0.0724409
criminal investigation	34	0.0535433
traffic stop	32	0.0503937
crime in progress	26	0.0409449
warrant service	16	0.0251969
other	13	0.0204724
vehicle pursuit	10	0.0157480
field interview contact	7	0.0110236
mental inquest warrant	7	0.0110236
escape from custody	4	0.0062992
mental inquest interview	3	0.0047244
NA	3	0.0047244
crowd control	1	0.0015748
special event	1	0.0015748
Total	635	1.0000000

4. For this final question, you will be recreating Table 3.

   a) First, using the uof tibble, filter `citizen_gender` to only include males and females, exclude all `NA` values from `citizen_race`, and create a new column named `force_used_1_effective_binary` which is a binary variable equal to 1 if `force_used_1_effective` is equal to "yes" and 0 otherwise. Save this new tibble as uof_filtered.

   b) Finally, using `group_by`, `summarize`, `janitor::adorn_totals`, and `mutate`, group by `citizen_gender` and `citize_race` and use the `summarize` function to create two summary statistics: the first named `effective_1` which is the sum of `force_used_1_effective_binary`, and the second named `counts` which is the number of observations in each grouping. Next you will need to use the `janitor::adorn_totals` and `mutate` functions to complete the Table 3. Note that the ordering of these functions is extremely important to getting the correct table. Most importantly, pay attention to the Total row to make sure your statistics look similar. Save this tibble as uof_filtered_table.

Table 3: Example of the table to be created in Question 4b.

citizen_gender	citizen_race	effective_1	counts	fraction_effective
female	black	65	144	0.4513889
female	hispanic	3	4	0.7500000
female	other	2	6	0.3333333
female	white	102	167	0.6107784
male	asian	5	6	0.8333333
male	black	592	1172	0.5051195
male	hispanic	41	86	0.4767442
male	other	10	35	0.2857143
male	white	518	1064	0.4868421
Total	-	1338	2684	0.4985097