project_4.Rmd

Ken Wood

8/12/2024

Project Description

In Part Four of the course project, you will extract data from the project_fire_911_calls.csv file, then use that data to examine how fire-related 911 calls varied over the course of one month. The data set contains information on emergency calls in Montgomery County, Pennsylvania, that happened in July 2019. This data set has three variables:

desc gives a description of each call in the form "M/D/Y H:M (month/day/year hour:minute), street address, township, Station X (where X is the station number, such as"33" or "308A")" *zip* is the ZIP Code of the call location *title* gives the reason for the call in the format "Type of Service: Specific Reason". In this data set, the only Type of Service is Fire.

You want to begin your analysis of how call volume varies over one month by creating a barplot that depicts how the daily number of fire-related calls changes over the course of the month. Before you make this plot, however, you need to clean up the data set.

Run the following code chunk to load and view this data set, then complete the steps below:

```
library(tidyverse)
library(utf8)
library(ellipsis)
library(lubridate)
library(dplyr)

calls <- read_csv("project_fire_911_calls.csv")
head(calls)
```

```
## # A tibble: 6 x 3
##   desc                                                        zip title
##   <chr>                                                     <dbl> <chr>
## 1 7/1/2019 0:09, GOLFVIEW DR & SHOAL CREEK DR, WHITPAIN, Station 33  19422 Fire~
## 2 7/1/2019 1:22, HIGHLAND AVE & DEAD END, JENKINTOWN, Station 96        NA Fire~
## 3 7/1/2019 2:13, ROCK CREEK RD & IDLEWILD RD, LOWER MERION, Station~ 19035 Fire~
## 4 7/1/2019 4:44, MOUNTAINVIEW AVE & HIGHLAND AVE, SCHWENKSVILLE, St~ 19473 Fire~
## 5 7/1/2019 5:07, MAIN ST & N 3RD AVE, ROYERSFORD, Station 98         19468 Fire~
## 6 7/1/2019 5:16, LANCASTER AVE & GRIFFIN LN, LOWER MERION, Station ~    NA Fire~
```

Step 1

Fill in the following code chunk to extract (a) the date and time of the call, (b) the street address, (c) the township, and (d) the station from the desc variable. For example, the first call would yield "7/1/2019 0:09", " GOLFVIEW DR & SHOAL CREEK DR", " WHITPAIN", and " Station 33". Next, you will need to remove any excess whitespace from these strings. Then, add each of these new variables to the data frame. For the date-time variable, you will need to convert the strings to date-time objects with the appropriate lubridate function then add these objects to the data frame as a new variable.

```
# Initialize vectors to store each of the new variables (date and time of call, street address, town, s
dts <- c()
addresses <- c()
towns <- c()
stations <- c()

for(i in 1:nrow(calls)) { # loop over emergency calls

  # get the description of the i^th call:
  callI <- calls[i, "desc"]

  # split the description text based on "," --> gives a vector of substrings:
  splitCallDesc <- str_split(callI, ",", simplify = TRUE)

  # store the date and time of the call, street address, town, and station:
  dts[i] <- splitCallDesc[1]
  addresses[i] <- splitCallDesc[2]
  towns[i] <- splitCallDesc[3]
  stations[i] <- splitCallDesc[4]

}

# Remove excess white space:
addresses <- str_trim(addresses)
towns <- str_trim(towns)
stations <- str_trim(stations)

# Add the new variables to the data frame:
calls2 <- calls # first create new data frame that doesn't overwrite the original
calls2$address <- addresses
calls2$town    <- towns
calls2$station <- stations
calls2$dt      <- mdy_hm(dts)     # use mdy_hm() to convert to date-time object since have the MONTH/DA
head(calls2)
```

```
## # A tibble: 6 x 7
##   desc                       zip title address town  station dt
##   <chr>                    <dbl> <chr> <chr>   <chr> <chr>   <dttm>
## 1 7/1/2019 0:09, GOLFVIEW~ 19422 Fire~ GOLFVI~ WHIT~ Statio~ 2019-07-01 00:09:00
## 2 7/1/2019 1:22, HIGHLAND~    NA Fire~ HIGHLA~ JENK~ Statio~ 2019-07-01 01:22:00
## 3 7/1/2019 2:13, ROCK CRE~ 19035 Fire~ ROCK C~ LOWE~ Statio~ 2019-07-01 02:13:00
## 4 7/1/2019 4:44, MOUNTAIN~ 19473 Fire~ MOUNTA~ SCHW~ Statio~ 2019-07-01 04:44:00
## 5 7/1/2019 5:07, MAIN ST ~ 19468 Fire~ MAIN S~ ROYE~ Statio~ 2019-07-01 05:07:00
## 6 7/1/2019 5:16, LANCASTE~    NA Fire~ LANCAS~ LOWE~ Statio~ 2019-07-01 05:16:00
```

Step 2

The dt variable contains both the date (month, day, and year), as well as the time (hour, minute, and second) of each call. Use a function from the lubridate package to extract only the day from the dt variable. Use this function to create another variable that specifies just the day and save it to calls2$day.

```
# Extract just the day from the entire date-time object and save in a variable called "day"
calls2$day <- day(calls2$dt)
head(calls2)
```

```
## # A tibble: 6 x 8
##   desc                 zip title address town  station dt                    day
##   <chr>              <dbl> <chr> <chr>   <chr> <chr>   <dttm>              <int>
## 1 7/1/2019 0:09, GO~ 19422 Fire~ GOLFVI~ WHIT~ Statio~ 2019-07-01 00:09:00     1
## 2 7/1/2019 1:22, HI~    NA Fire~ HIGHLA~ JENK~ Statio~ 2019-07-01 01:22:00     1
## 3 7/1/2019 2:13, RO~ 19035 Fire~ ROCK C~ LOWE~ Statio~ 2019-07-01 02:13:00     1
## 4 7/1/2019 4:44, MO~ 19473 Fire~ MOUNTA~ SCHW~ Statio~ 2019-07-01 04:44:00     1
## 5 7/1/2019 5:07, MA~ 19468 Fire~ MAIN S~ ROYE~ Statio~ 2019-07-01 05:07:00     1
## 6 7/1/2019 5:16, LA~    NA Fire~ LANCAS~ LOWE~ Statio~ 2019-07-01 05:16:00     1
```

Step 3

Fill in the following code chunk to compute the number of fire department calls for each station on each day of the month. Once filled in, this code chunk will produce a new data frame that gives the number of calls on each day of the month for each station.

```
callsPerDay <- calls2 %>%
  group_by(station,day) %>%
  summarise(NumCalls = n(), .groups='drop')

head(callsPerDay)
```

```
## # A tibble: 6 x 3
##   station     day NumCalls
##   <chr>     <int>    <int>
## 1 Station 1     3        1
## 2 Station 1     5        1
## 3 Station 1     6        2
## 4 Station 1     9        1
## 5 Station 1    11        5
## 6 Station 1    12        1
```

Step 4

Complete the following code chunk to make a barplot from the callsPerDay data frame that shows the number of fire department calls on each day of the month.

```
callsPerDay2 <- calls2 %>%
  group_by(day) %>%
  summarise(NumCalls = n(), .groups='drop')

head(callsPerDay2)
```

```
## # A tibble: 6 x 2
##     day NumCalls
##   <int>    <int>
## 1     1       43
## 2     2       60
## 3     3       80
## 4     4       67
## 5     5       65
## 6     6       66
```

```
ggplot(data = callsPerDay2, mapping = aes(x = day , y = NumCalls )) +
  geom_bar(stat = "identity") +
  xlab("Day") +
```

```
  ylab("Number of Calls") +
  ggtitle("Number of Fire Department Calls on Each Day of July 2019")
```

Step 5

Using the plot you created in Step 4, determine which day had an unusually high amount of fire activity. Then, use the data frame you made in Step 3 to determine which station responded to most of the calls on this day. Type your answers below.

```
# Get list of stations and number of calls for each station on July 11.

july11 <- callsPerDay %>% filter(day == 11)

# Find the station that had the most calls on July 11.
july11max <- july11[which.max(july11$NumCalls),]
july11max
```

```
## # A tibble: 1 x 3
##   station      day NumCalls
##   <chr>      <int>    <int>
## 1 Station 69    11       33
```