NYPD Motor Vehicle Collisions

Andrea J. Elhajj

6/10/2019

Please give all numerical answers to 10 digits of precision.

The city of New York has collected data on every automobile collision in city limits since mid-2012. Colisions are broken down by borough, zip code, latitude/longitude, and street name. Each entry describes injuries/deaths, collision causes, and vehicle types involved. The data can be downloaded from: https??data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95

Download the "NYPD Motor Vehicle Collisions" dataset in .csv format. The download link can be found under the "Export" tab. Information on the variables can be found on this page, as well, along with a preview of the rows of the dataset. For all questions, do not use data occurring after December 31, 2018.

What is the total number of persons injured in the dataset (up to December 31, 2018?)

```
library(readr)
library(stringr)
library(tidyr)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

options(digits=10) # Give all numerical answers to 10 digits of precision
df <- read_csv("NYPD_Motor_Vehicle_Collisions.csv") # Read in csv file

## Parsed with column specification:
## cols(
##   .default = col_character(),
##   TIME = col_time(format = ""),
##   `ZIP CODE` = col_double(),
##   LATITUDE = col_double(),
##   LONGITUDE = col_double(),
##   `NUMBER OF PERSONS INJURED` = col_double(),
##   `NUMBER OF PERSONS KILLED` = col_double(),
```

```
##   `NUMBER OF PEDESTRIANS INJURED` = col_double(),
##   `NUMBER OF PEDESTRIANS KILLED` = col_double(),
##   `NUMBER OF CYCLIST INJURED` = col_double(),
##   `NUMBER OF CYCLIST KILLED` = col_double(),
##   `NUMBER OF MOTORIST INJURED` = col_double(),
##   `NUMBER OF MOTORIST KILLED` = col_double(),
##   `UNIQUE KEY` = col_double()
## )

## See spec(...) for full column specifications.
```

```
df$YEAR <- format(as.Date(df$DATE, format="%m/%d/%Y"),"%Y") # Extract year,
create column
df$MONTH <- format(as.Date(df$DATE, format="%m/%d/%Y"),"%m") # Extract month,
create column
df <- df[df$YEAR != '2019',] # Do not use data occurring after December 31,
2018
dup <- df[duplicated(df$`UNIQUE KEY`),] # Check if any rows are duplicates -
no
# Total number of persons injured
injuries <- sum(df$`NUMBER OF PERSONS INJURED`, na.rm=TRUE)
injuries
```

```
## [1] 368034
```

What proportion of all collisions in 2016 occurred in Brooklyn? Only consider entries with a non-null value for BOROUGH.

```
Boro_2016 <- df %>% # Select 2016 collisions
        filter(YEAR == '2016') %>%
        select(BOROUGH)
Boro_2016 <- na.omit(Boro_2016) # Only consider entries with a non-null value
for BOROUGH
Brook_Prop_2016 <- table(Boro_2016)[c('BROOKLYN')]/sum(table(Boro_2016))
Brook_Prop_2016
```

```
##     BROOKLYN
## 0.3096198006
```

What proportion of collisions in 2016 resulted in injury or death of a cyclist?

```
Collis_2016 <- df %>% # Select 2016 collisions
        filter(YEAR == '2016') %>%
        select(DATE, `NUMBER OF CYCLIST INJURED`, `NUMBER OF CYCLIST KILLED`)
Cyc_2016 <- Collis_2016 %>%
        filter(`NUMBER OF CYCLIST INJURED` != 0 | `NUMBER OF CYCLIST KILLED`
!= 0) #Keep rows where cyclist injured or killed

Cyc_Prop_2016 <- nrow(Cyc_2016)/nrow(Collis_2016)
Cyc_Prop_2016
```

```
## [1] 0.02165483687
```

For each borough, compute the number of accidents per capita involving alcohol in 2017. Report the highest rate among the 5 boroughs. Use populations as given by https://wikipedia.org/wiki/Demographics_of_New_York_City.

```
Alc_2017 <- df %>%
  filter(YEAR == '2017', str_detect(`CONTRIBUTING FACTOR VEHICLE 1`,
fixed('Alcohol', ignore_case = TRUE)) |
           str_detect(`CONTRIBUTING FACTOR VEHICLE 2`, fixed('Alcohol',
ignore_case = TRUE)) |
           str_detect(`CONTRIBUTING FACTOR VEHICLE 3`, fixed('Alcohol',
ignore_case = TRUE)) |
           str_detect(`CONTRIBUTING FACTOR VEHICLE 4`, fixed('Alcohol',
ignore_case = TRUE)) |
           str_detect(`CONTRIBUTING FACTOR VEHICLE 5`, fixed('Alcohol',
ignore_case = TRUE))) %>%
  select(BOROUGH) %>%
  filter(!is.na(BOROUGH)) # Must have entry for BOROUGH, remove NA
Alc_Bronx_2017 <- table(Alc_2017)[c('BRONX')]/1471160
Alc_Brookl_2017 <- table(Alc_2017)[c('BROOKLYN')]/2648771
Alc_Manhat_2017 <- table(Alc_2017)[c('MANHATTAN')]/1664727
Alc_Queens_2017 <- table(Alc_2017)[c('QUEENS')]/2358582
Alc_Staten_2017 <- table(Alc_2017)[c('STATEN ISLAND')]/479458
Max_Alc_Prop_2017 <- max(Alc_Bronx_2017, Alc_Brookl_2017, Alc_Manhat_2017,
Alc_Queens_2017, Alc_Staten_2017)
Max_Alc_Prop_2017
```

```
## [1] 0.0002272752156
```

Obtain the number of vehicles involved in each collision in 2016. Group the collisions by zip code and compute the sum of all vehicles involved in collisions in each zip code, then report the maximum of these values.

```
Zip_2016 <- df %>%
  filter(YEAR == '2016') %>%
  mutate(`VEHICLE TYPE CODE 1` = 1) %>% # Assume 1 vehicle involved in each
accident
  mutate(`VEHICLE TYPE CODE 2` = ifelse(is.na(`VEHICLE TYPE CODE 2`), 0, 1))
%>% # Replace with 0 if na, else enter 1
  mutate(`VEHICLE TYPE CODE 3` = ifelse(is.na(`VEHICLE TYPE CODE 3`), 0, 1))
%>%
  mutate(`VEHICLE TYPE CODE 4` = ifelse(is.na(`VEHICLE TYPE CODE 4`), 0, 1))
%>%
  mutate(`VEHICLE TYPE CODE 5` = ifelse(is.na(`VEHICLE TYPE CODE 5`), 0, 1))
%>%
  mutate(`CONTRIBUTING FACTOR VEHICLE 1` = 1) %>% # Assume 1 vehicle involved
in each accident
  mutate(`CONTRIBUTING FACTOR VEHICLE 2` = ifelse(is.na(`CONTRIBUTING FACTOR
VEHICLE 2`), 0, 1)) %>%
  mutate(`CONTRIBUTING FACTOR VEHICLE 3` = ifelse(is.na(`CONTRIBUTING FACTOR
VEHICLE 3`), 0, 1)) %>%
```

```
  mutate(`CONTRIBUTING FACTOR VEHICLE 4` = ifelse(is.na(`CONTRIBUTING FACTOR
VEHICLE 4`), 0, 1)) %>%
  mutate(`CONTRIBUTING FACTOR VEHICLE 5` = ifelse(is.na(`CONTRIBUTING FACTOR
VEHICLE 5`), 0, 1)) %>%
  mutate(`TOTAL VEHICLE TYPE` = select(.,`VEHICLE TYPE CODE 1`:`VEHICLE TYPE
CODE 5`) %>% rowSums()) %>% # Sum across the 5 rows
  mutate(`TOTAL CONTRIBUTING FACTOR` = select(.,`CONTRIBUTING FACTOR VEHICLE
1`:`CONTRIBUTING FACTOR VEHICLE 5`) %>% rowSums()) %>% # Repeat
  mutate(`TOTAL VEHICLES INVOLVED` = ifelse((`TOTAL VEHICLE TYPE` >= `TOTAL
CONTRIBUTING FACTOR`),`TOTAL VEHICLE TYPE`, `TOTAL CONTRIBUTING FACTOR`)) #
Take larger value
max(tapply(Zip_2016$`TOTAL VEHICLES INVOLVED`, Zip_2016$`ZIP CODE`, FUN=sum))

## [1] 5941
```

Consider the total number of collisions each year from 2013-2018. Is there an apparent trend? Fit a linear regression for the number of collisions per year and report its slope.

```
collisions_per_yr <- df %>%
        filter(YEAR >= '2013') %>%
        mutate(Collision = 1) %>%
        select(YEAR, Collision)
per_year <- data.frame(matrix(tapply(collisions_per_yr$`Collision`,
collisions_per_yr$`YEAR`, FUN=sum)))
colnames(per_year) <- c('Collisions')
Years <- c(2013, 2014, 2015, 2016, 2017, 2018)
per_year <- cbind(Years, per_year)
lm.Collisions <- lm(formula = Collisions ~ Years, data = per_year)
lm.Collisions

##
## Call:
## lm(formula = Collisions ~ Years, data = per_year)
##
## Coefficients:
##   (Intercept)          Years
## -12776339.181       6448.171

# The following two functions allow the slope to be displayed to 10 decimal
places:
specify_decimal <- function(x, k) format(round(x, k), nsmall=k)
new_summary  <- function(lmcoef, digits) {
        coefs <- as.data.frame(lmcoef)
        coefs[] <- lapply(coefs, function(x) specify_decimal(x, digits))
        coefs

}
model_summ <- new_summary(summary(lm.Collisions)$coefficients, 10)
slope <- model_summ$Estimate[2]
slope
```

```
## [1] "     6448.1714285721"
```

Do winter driving conditions lead to more multi-car collisions? Compute the rate of multi car collisions as the proportion of the number of collisions involving 3 or more cars to the total number of collisions for each month of 2017. Calculate the chi-square test statistic for testing whether a collision is more likely to involve 3 or more cars in January than in May.

```
Monthly_2017 <- df %>%
        filter(YEAR == '2017') %>%
        mutate(`VEHICLE TYPE CODE 1` = 1) %>% # Assume 1 vehicle involved in
each accident
        mutate(`VEHICLE TYPE CODE 2` = ifelse(is.na(`VEHICLE TYPE CODE 2`),
0, 1)) %>%
        mutate(`VEHICLE TYPE CODE 3` = ifelse(is.na(`VEHICLE TYPE CODE 3`),
0, 1)) %>%
        mutate(`VEHICLE TYPE CODE 4` = ifelse(is.na(`VEHICLE TYPE CODE 4`),
0, 1)) %>%
        mutate(`VEHICLE TYPE CODE 5` = ifelse(is.na(`VEHICLE TYPE CODE 5`),
0, 1)) %>%
        mutate(`CONTRIBUTING FACTOR VEHICLE 1` = 1) %>% # Assume 1 vehicle
involved in each accident
        mutate(`CONTRIBUTING FACTOR VEHICLE 2` = ifelse(is.na(`CONTRIBUTING
FACTOR VEHICLE 2`), 0, 1)) %>%
        mutate(`CONTRIBUTING FACTOR VEHICLE 3` = ifelse(is.na(`CONTRIBUTING
FACTOR VEHICLE 3`), 0, 1)) %>%
        mutate(`CONTRIBUTING FACTOR VEHICLE 4` = ifelse(is.na(`CONTRIBUTING
FACTOR VEHICLE 4`), 0, 1)) %>%
        mutate(`CONTRIBUTING FACTOR VEHICLE 5` = ifelse(is.na(`CONTRIBUTING
FACTOR VEHICLE 5`), 0, 1)) %>%
        mutate(`TOTAL VEHICLE TYPE` = select(.,`VEHICLE TYPE CODE 1`:`VEHICLE
TYPE CODE 5`) %>% rowSums()) %>%
        mutate(`TOTAL CONTRIBUTING FACTOR` = select(.,`CONTRIBUTING FACTOR
VEHICLE 1`:`CONTRIBUTING FACTOR VEHICLE 5`) %>% rowSums()) %>%
        mutate(`TOTAL VEHICLES INVOLVED` = ifelse((`TOTAL VEHICLE TYPE` >=
`TOTAL CONTRIBUTING FACTOR`),`TOTAL VEHICLE TYPE`, `TOTAL CONTRIBUTING
FACTOR`)) %>%
        mutate(Collision = 1)
ThreeMore_Monthly_2017 <- Monthly_2017 %>%
        dplyr::filter(`TOTAL VEHICLES INVOLVED` >= 3) %>% # Select only
collisions involving 3 or more vehicles
        select(MONTH, Collision)

Monthly_2017 <- Monthly_2017 %>%
        select(MONTH, Collision)
Monthly_2017 <- tapply(Monthly_2017$Collision, Monthly_2017$MONTH, FUN=sum)
ThreeMore_Monthly_2017 <- tapply(ThreeMore_Monthly_2017$Collision,
ThreeMore_Monthly_2017$MONTH, FUN=sum)
Less3_Monthly_2017 <- Monthly_2017 - ThreeMore_Monthly_2017
ContingencyTable <- matrix(c(ThreeMore_Monthly_2017[1], Monthly_2017[1]-
ThreeMore_Monthly_2017[1], ThreeMore_Monthly_2017[5], Monthly_2017[5]-
```

```
ThreeMore_Monthly_2017[5]), byrow = TRUE, 2, 2)
chisq <- chisq.test(ContingencyTable, correct = FALSE)
print(chisq$statistic, digits = 14)
```

```
##       X-squared
## 5292.9126425118
```

We can use collision locations to estimate the areas of the zip code regions. Represent each as an ellipse with semi-axes given by a single standard deviation of the longitude and latitude. For collisions in 2017, estimate the number of collisions per square kilometer. Note: Some entries may have invalid or incorrect (latitude, longitude) coordinates. Drop any values that are invalid or seem unreasonable for New York City.

```
EllipseInfo <- df %>%
        filter(YEAR == '2017', !is.na(`ZIP CODE`)) %>%
        mutate(Collision = 1) %>%
        select(`ZIP CODE`,LATITUDE, LONGITUDE, Collision) %>%
        group_by((`ZIP CODE`)) %>%
        summarise(`LAT SD` = sd(LATITUDE, na.rm = TRUE), `LONG SD` =
sd(LONGITUDE, na.rm = TRUE), `TOTAL COLLISIONS` = sum(Collision)) %>%
        filter(`TOTAL COLLISIONS` >= 1000) %>%
        # AT LATITUDE 40 DEGREES (NORTH OR SOUTH): 1 degree of latitude =
111.03 km, 1 degree of longitude =  85.39 km:
        mutate(AREA = (pi*(`LAT SD`*111.03)*(`LONG SD`*85.39))) %>%
        mutate(`NO COLLISIONS PER SQ KM` = (`TOTAL COLLISIONS`/AREA))
max_Col_per_sqkm <- max(EllipseInfo$`NO COLLISIONS PER SQ KM`)
max_Col_per_sqkm
```

```
## [1] 5316.144689
```