GU4206_GR5206 Fall 2021 Midterm lt2846

Lili Tan

10/22/2021

Part 0: Instructions

The STAT GU4206/GR5206 midterm is open notes, open book(s), open computer and online resources are allowed. Students are **not** allowed to communicate with any other people regarding the exam with the exception of the instructor (Gabriel Young) and the TAs. This includes emailing fellow students, using WeChat and other similar forms of communication. If there is any suspicion of one or more students cheating, further investigation will take place. If students do not follow the guidelines, they will receive a zero on the exam and potentially face more severe consequences. The exam will be posted on Friday, 10/22/2021 at 10:15AM (ET). Students are required to submit both the .pdf (or .html) and .rmd files on Canvas by Friday, at 12:55PM (ET). Students must budget their time appropriately to successfully knit and submit their exam by the deadline. Late exams will not be accepted

A few more recommendations follow:

- Don't forget to submit both the correct .rmd file and at least one of your .html or .pdf files.
- Save your .rmd regularly to avoid any problems if your computer crashes.
- Please ensure your output is tidy. Do not print pages and pages of data. Doing so will result in points deducted.
- Please stop working on the exam at least 15 minutes before it's deadline to make sure your RMarkdown file knitts properly.
- Please use the Zoom chat log to ask questions. Do not use the microphone to ask questions because it's disruptive to the class.

Part I: NYC Party Data and Warm-Up

The following dataset **NYC_party.csv** contains all noise complaint calls that were received by the city police with complaint type "Loud music/Party" in 2016. The data contains the time of the call, time of the police response, coordinates and part of the city.

```
party <- read.csv("NYC_party.csv")
dim(party)
```

```
## [1] 20000     8
```

Problem 1 [1 points]

Display the names of each variable. There are 8 variables in total.

```r
# Problem1
names(party)
```

```
## [1] "Created.Date"  "Closed.Date"   "Location.Type" "Incident.Zip"
## [5] "City"          "Borough"       "Latitude"      "Longitude"
```

Problem 2 [5 points]

Produce the same task shown below using only one line of R code. That is, compute **out** in one line of R code without using a loop. If you can't solve this problem in one line, try your best for partial credit.

```r
some_number <- NULL
for (i in 1:nrow(party)) {
  logical_value <- party$Location.Type[i] == "House of Worship"
  if (logical_value==T) {
    some_number[i] <- 1
    } else {some_number[i] <- 0}
}
out <- some_number[1]/nrow(party)
for (i in 2:nrow(party)) {
  out <- out + some_number[i]/nrow(party)
}
out
```

```
## [1] 0.00325
```

```r
# Problem2
mean(party$Location.Type == "House of Worship")
```

```
## [1] 0.00325
```

Part II: NYC Party Data and Some Data Science Tasks

Problem 3 [3 points]

Write a function named **check_num_na** that counts the number of NAs in a vector. Check your function on the vector **party$Longitude**.

```r
# Problem3
check_num_na <- function(v){
  num_na <- sum(is.na(v))
  return(num_na)
}
check_num_na(party$Longitude)
```

```
## [1] 142
```

Problem 4 [2 points]

Apply your function **check_num_na** to each column of the **party** dataframe. To accomplish this task, choose an appropriate function from the apply family. How many NAs do Latitude and Longitude have?

```
# Problem4
apply(party, 2, check_num_na)
```

```
##  Created.Date   Closed.Date Location.Type  Incident.Zip          City
##             0             0             0           100             0
##       Borough      Latitude     Longitude
##             0           142           142
```

Both of Latitude and Longitude have 142 NAs.

Problem 5 [5 points]

Remove all rows in the **party** dataframe with NAs in both Latitude and Longitude. Name your new dataframe **party_new** and use this updated dataframe for problems 6-11.

```
# Problem5
party_new <- party[!is.na(party$Latitude)&!is.na(party$Longitude),]
```

Problem 6 [4 points]

Create a table showing the raw counts of **Location.Type** with the frequency displayed in decreasing order.

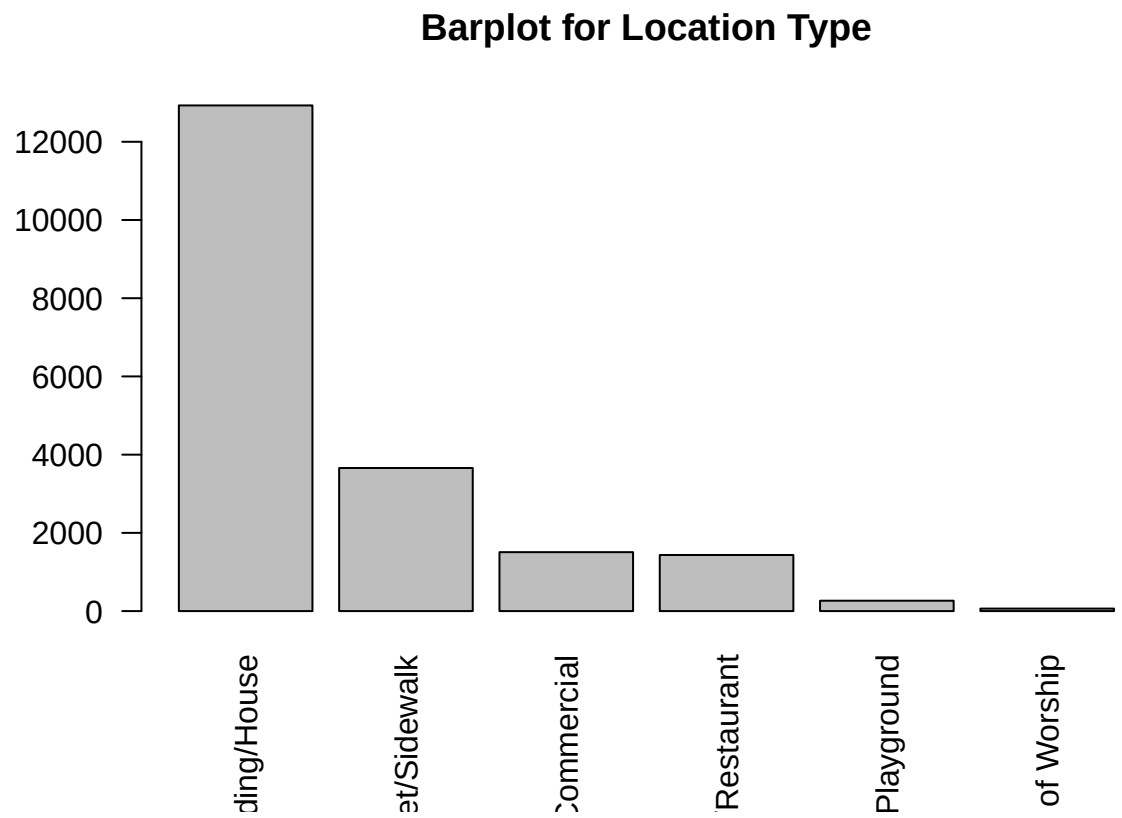
```
# Problem6
table_location <- sort(table(party_new$Location.Type),decreasing = T)
table_location
```

```
##
## Residential Building/House            Street/Sidewalk
##                      12929                       3659
##           Store/Commercial        Club/Bar/Restaurant
##                       1507                       1434
##            Park/Playground          House of Worship
##                        265                         64
```

Problem 7 [5 points]

Using ggplot or base R, create a barplot that shows the raw counts of **Location.Type** with the frequency displayed in decreasing order. This is similar to a Pareto chart. Your plot should look nice.

```
# Problem7
barplot(table_location,main = "Barplot for Location Type",las=2)
```



Problem 8 [5 points]

Given that a randomly chosen noise violation is in the **BRONX**, what is the estimated probability that the **Location.Type** is on the **Street/Sidewalk**?

```
# Problem8
mean(party_new$Location.Type[party_new$Borough=="BRONX"]=="Street/Sidewalk")
```

```
## [1] 0.181991
```

Problem 9 [5 points]

Write a function named **Street_party**, with input **Borough**, that computes the estimated conditional probability that the **Location.Type** is on the **Street/Sidewalk** given its **Borough**. Test your function on the **BRONX**, i.e., **Street_party("BRONX")** should produce the same answer as problem 8.

```
# Problem9
Street_party <- function(B){
  prob <- mean(party_new$Location.Type=="Street/Sidewalk"
               &party_new$Borough==B)/mean(party_new$Borough==B)
  return(prob)
}
Street_party("BRONX")
```

```
## [1] 0.181991
```

Problem 10 [5 points]

Apply your function **Street_party** to the vector **my_Borough** defined below. To accomplish this task, choose an appropriate function from the apply family.

```
# Problem10
my_Borough <- levels(factor(party_new$Borough))
sapply(my_Borough,Street_party)
```

```
##         BRONX      BROOKLYN     MANHATTAN        QUEENS STATEN ISLAND
##    0.18199097    0.16254301    0.27627149    0.08658644    0.09475806
```

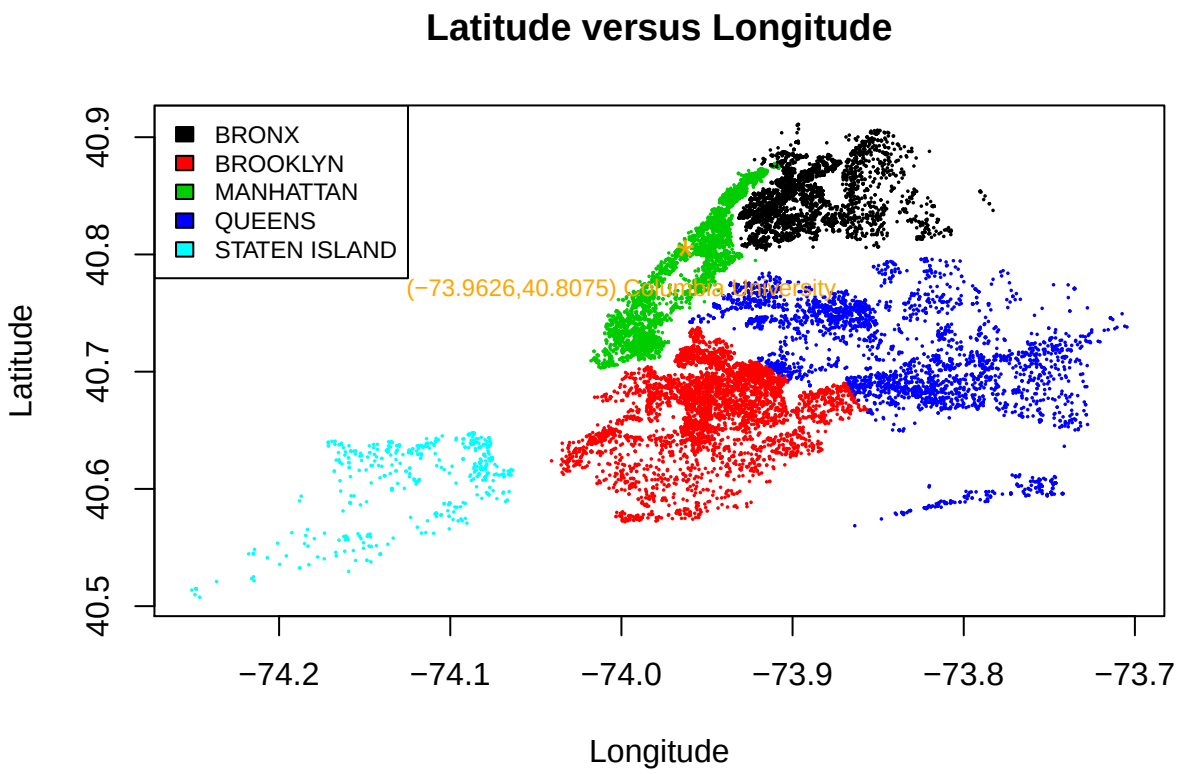
Which Borough produces the highest likelihood of noise violations from the **Street/Sidewalk**? Does this surprise you?

Manhattan produces the highest likelihood of noise violations(0.27627149) from the Street/Sidewalk. Is does not surprise me since Manhattan is the most flourishing place in New York.

Problem 11 [10 points] (base R graphics)

Using **Base R**, create a scatterplot of **Latitude** versus **Longitude**, split by **Borough**. I.e., each Borough should have a different color. Make sure to include appropriate labels, a legend and adjust the point size/point type to be **cex=.25** and **pch=16** respectively. Also include an additional point and text for the location of Columbia University. Note that Columbia is located at **Longitude=-73.9626** and **Latitude=40.8075**.

```
# Problem11
plot(party_new$Longitude, party_new$Latitude,xlab = "Longitude",ylab = "Latitude",
     main = "Latitude versus Longitude",col=factor(party_new$Borough),cex=0.25,pch=16)
legend("topleft",legend=levels(factor(party_new$Borough)),
fill=1:length(levels(factor(party_new$Borough))), cex=0.75)
points(x=-73.9626,y=40.8075, pch = "*", col = "orange",cex=1.5)
text(x=-74,y=40.77,"(-73.9626,40.8075) Columbia University",col = "orange",cex=0.75)
```



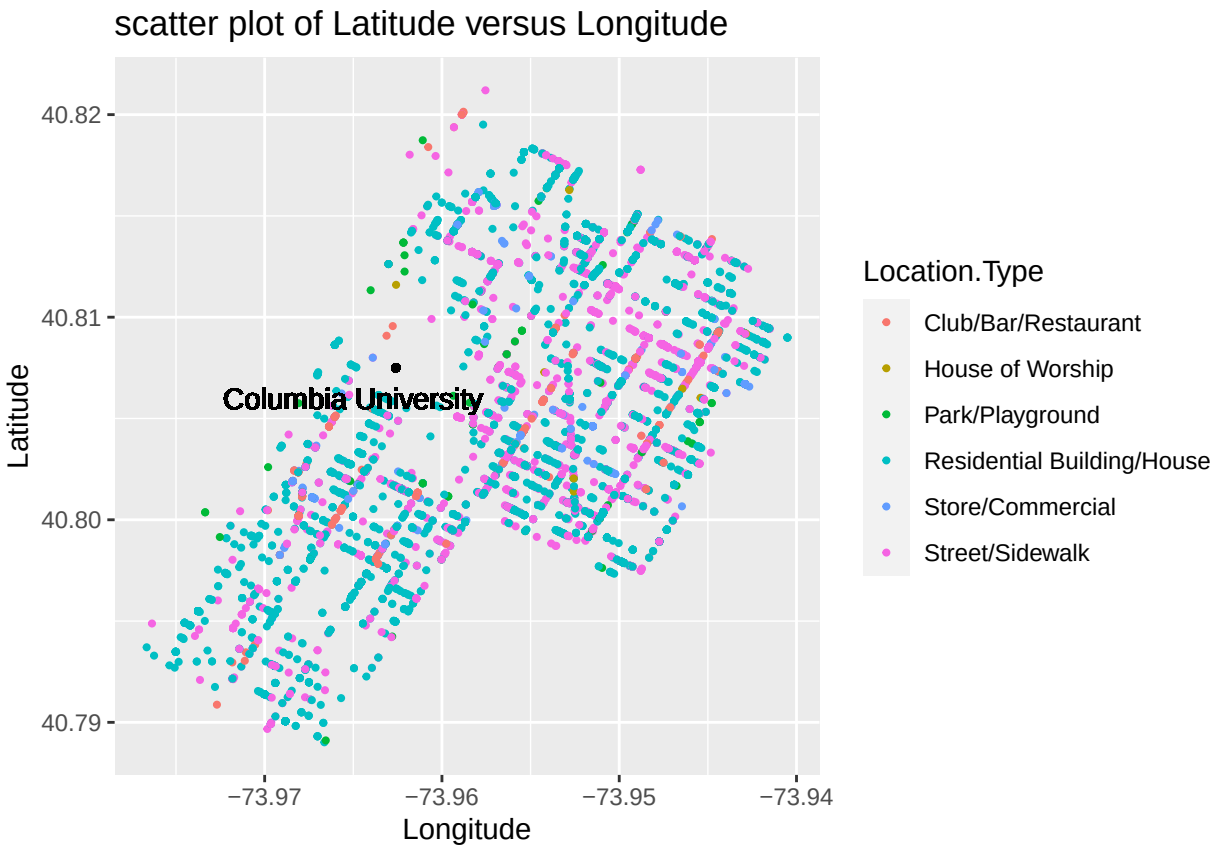
Problem 12 [10 points] (ggplot)

Consider a new dataset **Columbia_party.csv** that includes 7225 noise complaint calls recorded in 2016 related to zip codes 10025, 10026, and 10027. The variables are the same as **NYC_party.csv**. Using **ggplot**, create a scatterplot of **Latitude** versus **Longitude**, split by **Location.Type**. I.e., each Location.Type should have a different color. Also include an additional point for the location of Columbia University. Note that Columbia is located at **Longitude=-73.9626** and **Latitude=40.8075**.

```
near_columbia <- read.csv("Columbia_party.csv")
dim(near_columbia)
```

```
## [1] 7225    8
```

```
# Problem12
library(ggplot2)
ggplot(data=near_columbia) +
  geom_point(aes(x=Longitude,y=Latitude,col=Location.Type),cex=0.75)+
  labs(title = "scatter plot of Latitude versus Longitude",
       x="Longitude",y="Latitude")+
  geom_point(aes(x=-73.9626,y=40.8075),color="black",cex=1)+
  geom_text(aes(x=-73.965,y=40.806,label="Columbia University"),col="black")
```



Part III: Non-parametric procedures

The dataset **CO2__Ford__Chevy.csv** contains official records of CO2 emissions based on various cars. The included variables are **Make** and **CO2__Emissions**.

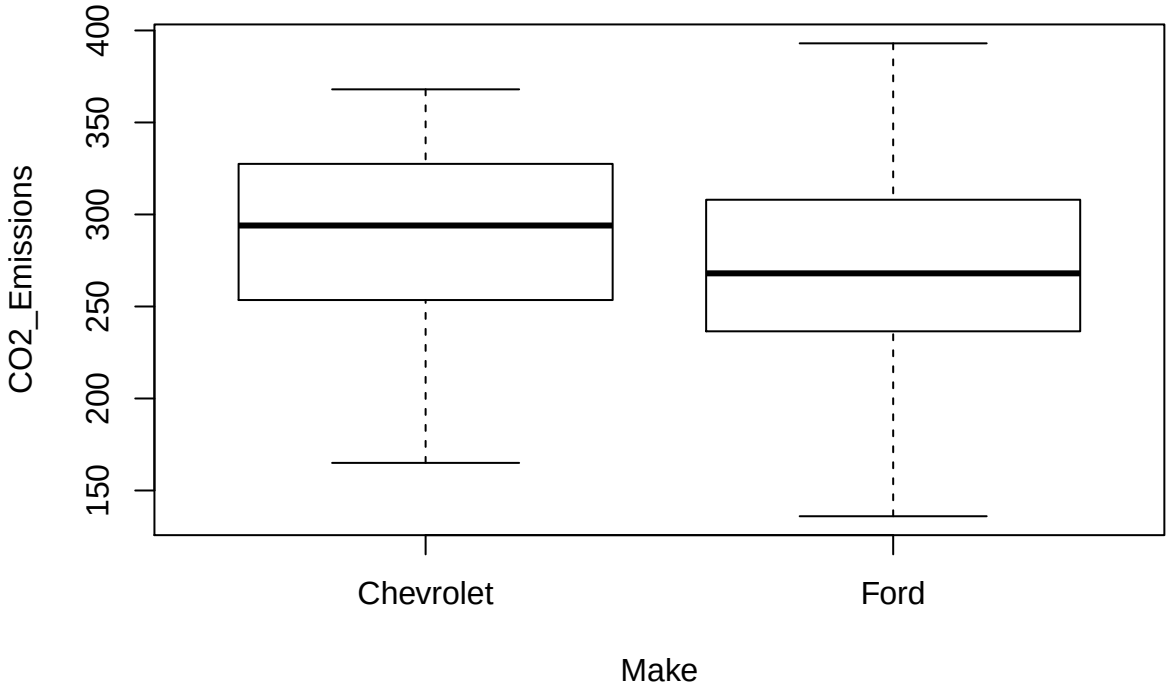
```
data <- read.csv("CO2_Ford_Chevy.csv")
dim(data)
```

```
## [1] 171   2
```

Problem 13 [5 points]

Create an appropriate graphic that shows the the average CO2 emission rate split by Make (Chevrolet and Ford), i.e., showing a continuous variable versus a categorical variable.

```
# Problem13
plot(data)
```



Problem 14 [5 points]

Our research question follows: Does the true average CO2 emission rate differ between Chevrolet and Ford? This question can be formulated as a hypothesis testing procedure with null/alternative pair:

$$H_0 : \mu_C - \mu_F = 0 \quad \text{vs.} \quad H_A : \mu_C - \mu_F \neq 0$$

At 95% confidence (or 5% significance), run a traditional two-sample t-test to answer the above research question. The R function of interest is **t.test()**.

```
# Problem14
t.test(x=data$CO2_Emissions[data$Make=="Chevrolet"],y=data$CO2_Emissions[data$Make=="Ford"])
```

```
##
##  Welch Two Sample t-test
##
## data:  data$CO2_Emissions[data$Make == "Chevrolet"] and data$CO2_Emissions[data$Make == "Ford"]
## t = 1.8357, df = 168.2, p-value = 0.06817
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.105455 30.415800
## sample estimates:
## mean of x mean of y
##  285.3333  270.6782
```

Since we get p-value=0.06817 which is greater than 0.05, we fail to reject H0. Hence, we conclude that the true average CO2 emission rate is NOT differ between Chevrolet and Ford.

Problem 15 [15 points]

Run a bootstrap procedure to test if the true average CO2 emission rate differs between Chevrolet and Ford. Use confidence level 95%. The null/alternative pair of interest is the same as problem 14:

$$H_0 : \mu_C - \mu_F = 0 \quad \text{vs.} \quad H_A : \mu_C - \mu_F \neq 0$$

Is your statistical conclusion the same as problem 14?

```
# Problem15
set.seed(1)
B <- 1000
n <- 171
mu12 <- NULL
for (b in 1:B) {
  re_index <- sample(1:n, n, replace = TRUE)
  re_data <- data[re_index,]
  re_mu1 <- mean(re_data$CO2_Emissions[re_data$Make=="Chevrolet"])
  re_mu2 <- mean(re_data$CO2_Emissions[re_data$Make=="Ford"])
  mu12[b] <- re_mu1-re_mu2
}
mu1 <- mean(data$CO2_Emissions[data$Make=="Chevrolet"])
mu2 <- mean(data$CO2_Emissions[data$Make=="Ford"])
Cl <- 2*(mu1-mu2)-quantile(mu12, probs = c(0.975))
Cu <- 2*(mu1-mu2)-quantile(mu12, probs = c(0.025))
int <- c(Cl, Cu)
int
```

```
##     97.5%      2.5%
## -1.253821 30.459555
```

Since the interval we get is [-1.253821,30.459555] i.e. 0 is in the interval, we conclude that the true average CO2 emission rate is NOT differ between Chevrolet and Ford. My statistical conclusion is as same as in problem 14.

Problem 16 [10 points]

Run a permutation test to answer the same research question from problems 14 and 15.

$$H_0 : \mu_F - \mu_C = 0 \quad \text{vs.} \quad H_A : \mu_F - \mu_C \neq 0$$

Is your statistical conclusion the same as problems 14 and 15?

```
# Problem 16
D <- abs(mu1-mu2)
nf <- sum(data$Make=="Ford")
nc <- sum(data$Make=="Chevrolet")
P <- 100000
sample_diff <- rep(NA, P)
```

```r
for (i in 1:P){
  perm_data <- data$CO2_Emissions[sample(1:(nf+nc))]
  meanf <- mean(perm_data[1:nf])
  meanc <- mean(perm_data[-(1:nf)])
  sample_diff[i] <- meanc-meanf
}
pval <- mean(abs(sample_diff) >= D)
pval
```

```
## [1] 0.06869
```

Here, we get p-value=0.06892 from permutation test which is greater than 0.05, so we fail to reject H0. Hence, we conclude that the true average CO2 emission rate is NOT differ between Chevrolet and Ford. My statistical conclusion is as same as in problem 14 and 15.

Problem 17 [5 points]

In your expert opinion, is the bootstrap procedure equivalent to the permutation test? Your answer should be at least two sentences but less than a two paragraphs.

I think bootstrap procedure is somehow equivalent to the permutation test but not completely. For the conclusion, in most cases, we can get same conclusion from bootstrap and permutation. However, the permutation test is better for testing hypotheses and bootstrap procedure is better for estimating confidence intervals. For permutation test, it is more powerful and easier when we want to test a specific null hypothesis. For bootstrap procedure, it estimates the variability of the sampling process, and it works well for estimating confidence intervals.