11/10

Below are the tasks I have done this week:

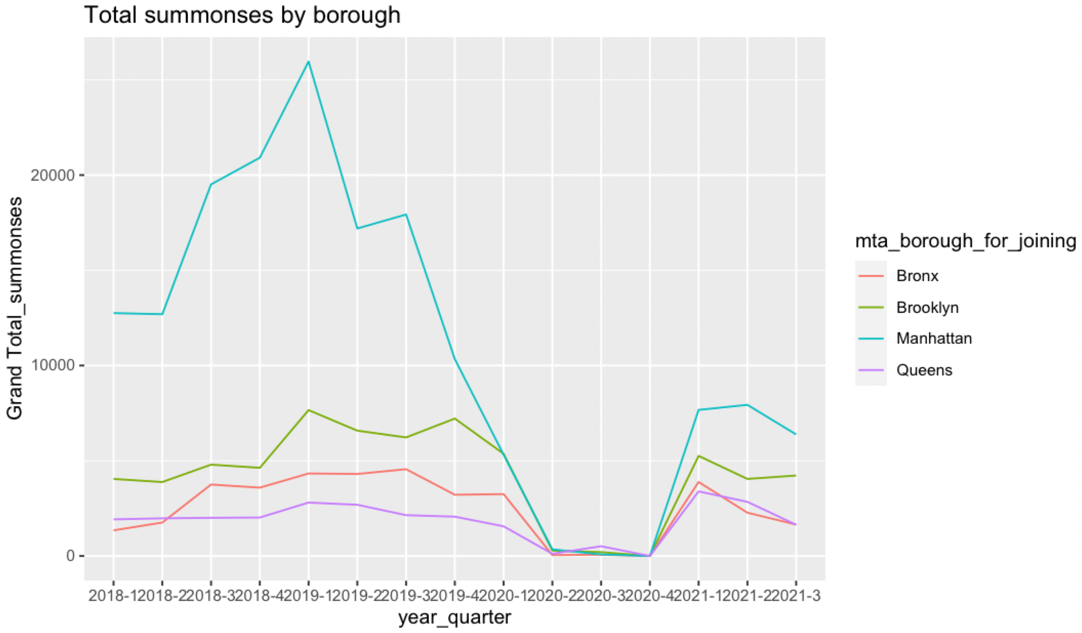
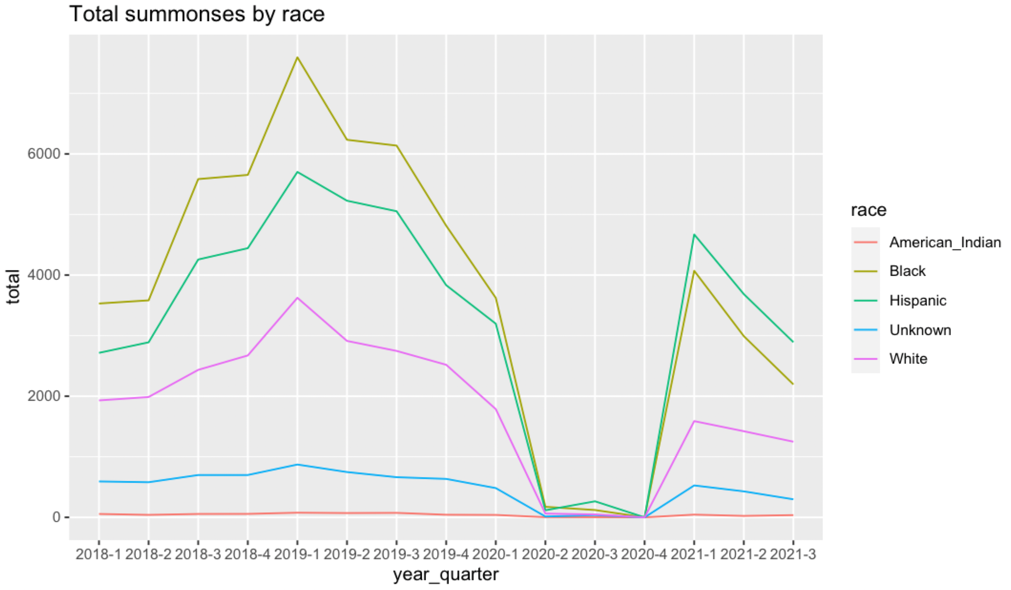
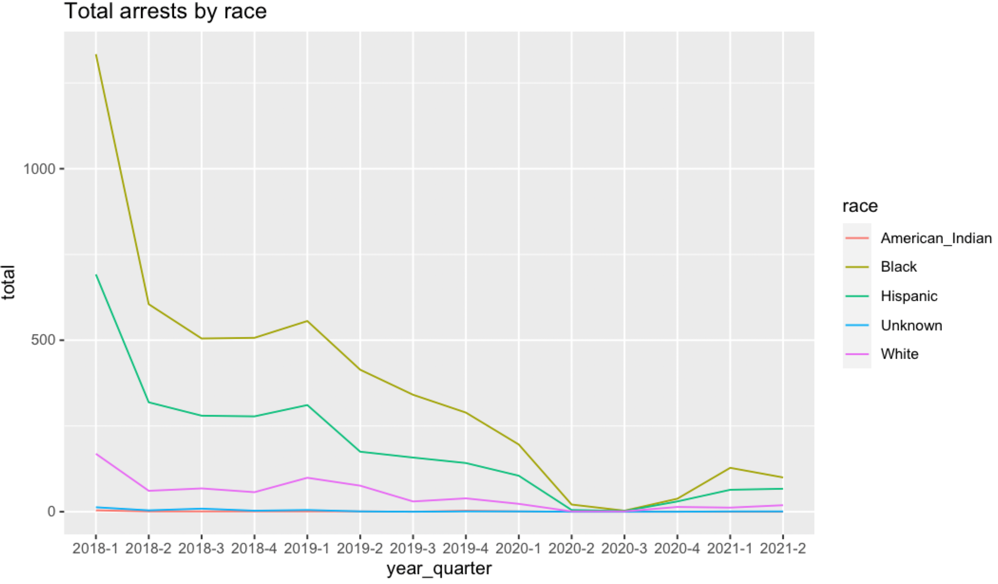
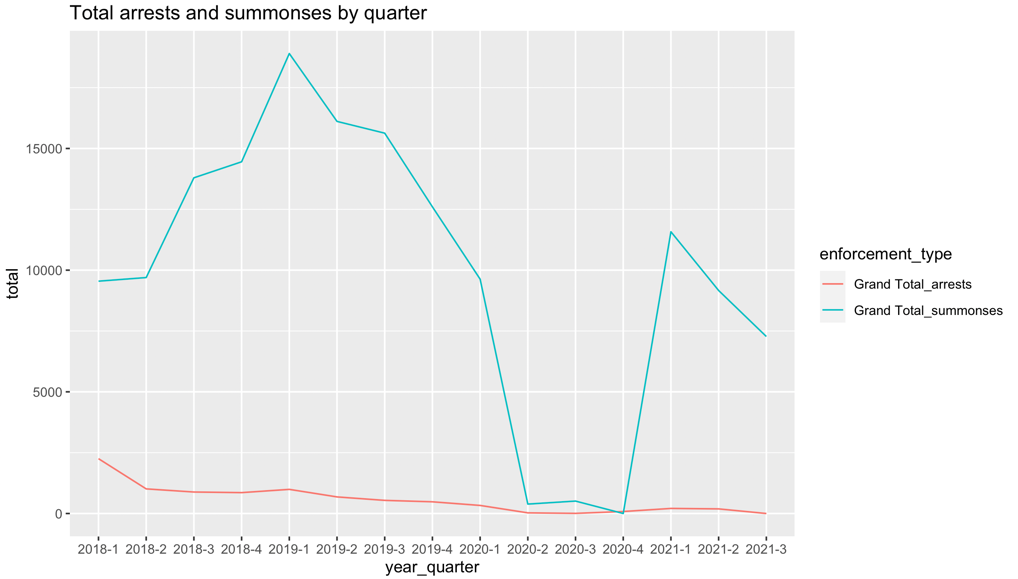
1. **Made “master.R” script shorter** by creating separate R files for different functions/stages.
   1. "01\_load\_raw\_arrests\_and\_summonses\_quarterly\_data.R"
   2. "02-1\_get\_quarterly\_arrests\_data\_tidy.R"
   3. "02-2\_get\_quarterly\_summonses\_data\_tidy.R"
   4. “03\_stack\_tidy\_quarterly\_data\_by\_enforcement\_type.R”
2. **Identified and started fixing NYPD naming issues** (It’s turned out I had valid concerns!)
   1. Firstly checked any inconsistencies/potential issues in the stacked **arrests** dataset.
      1. Found that in a18q3, “MYRTLE-WILLOUGHBY AVENUES - G line” was incorrectly assigned to "Manhattan." In the other quarters, the station was correctly assigned to “Brooklyn.” (FIXED)
   2. Secondly checked any inconsistencies/potential issues in the stacked **summonses** dataset.
      1. Found that “42 ST.-PORT AUTHORITY BUS TERMINAL” in other quarters is named as “42 ST.-PORT AUTHORITY BUS TERM” in s20q3. (FIXED)
      2. **[District # Office]** Four boroughs have “DISTRICT # OFFICE” values in the ‘Street\_Names’ column: Queens, 2; Manhattan, 4; Brooklyn, 2; Bronx, 2. I could easily find the matching station for offices located in Queens and Brooklyn. However, I couldn’t determine which ‘Street\_Names’ and ‘All\_Lines’ values to assign for three district offices in Manhattan, and one in Bronx, for two main reasons. First, some offices have multiple stations nearby so it is hard to determine one matching station (i.e. District 2 Office). Second, some offices have only one station multiple lines go through, but the station appears as multiple stations with different lines in the data. (i.e. District 4 Office). I would love to look at the four district offices below. I also listed possible options I found from existing nypd names below each office. (FIX THIS WHEN CHANGING UNITS TO MTA NAMES)
         1. (Manhattan) DISTRICT 1 OFFICE
            1. 59 ST.-COLUMBUS CIRCLE - "A,B,C,D,E" OR "1”
         2. (Manhattan) DISTRICT 2 OFFICE
            1. CANAL STREET - "A,C,E" OR "N,Q,R"
            2. FRANKLIN STREET – “1”
         3. (Manhattan) DISTRICT 4 OFFICE
            1. UNION SQUARE – “N,Q,R”
            2. 14 ST.-UNION SQUARE – “L”
            3. The two look like the same station.
         4. (Bronx) DISTRICT 11 OFFICE
            1. 161 ST.-YANKEE STADIUM – “B,D” OR “4”
   3. Thirdly stacked all nypd stations from both stacked and arrests datasets, and checked any inconsistencies across enforcement types.
      1. Found that “HALSEY STREET – L line” is assigned to different boroughs in different enforcement types datasets. It is assigned to “Queens” in the arrests data, and “Brooklyn” in the summonses data. Googled and found the station is located between the two boroughs. So I looked it up from mta names and assigned it to “Brooklyn” accordingly. (FIXED)
      2. Found one station named consistently within each enforcement type, but inconsistently across enforcement types. “111 ST.-ROOSEVELT AVE” in arrests, is named as “111 St-Roosevelt Ave” Changed its name to “111 ST.-ROOSEVELT AVE” for consistency with other nypd station names. (FIXED)
      3. Found three streets having overlapping lines within the summons dataset. (FIXED)
         1. [116 STREET] In s20q3, the street has “A,B,C” line, while it has “B,C” line in other quarters. Googled and found that “A” and “B,C” lines are two different stations. Changed “A,B,C” to “B,C.” (FIXED)
         2. [UNION SQUARE] In s20q3, the street has “N,R” line, while it has “N,Q,R” line in other quarters. Changed “N,R” to “N,Q,R.” (FIXED)
         3. [36 STREET] The street has “B,M,N,R” and “D,N,R” line in Brooklyn, and “M,R” line in Queens. After looking it up from mta names, fixed to “D,N,R” in Brooklyn, and “M,R” in Queens. (FIXED)
      4. **[UNKNOWN(arrests) vs OFF-SYSTEM(summonses)]** Arrests without station information appear as “UNKNOWN” in the arrests dataset, while summonses without station information appear as “OFF-SYSTEM” in the summonses dataset. (FIXED)

11/18

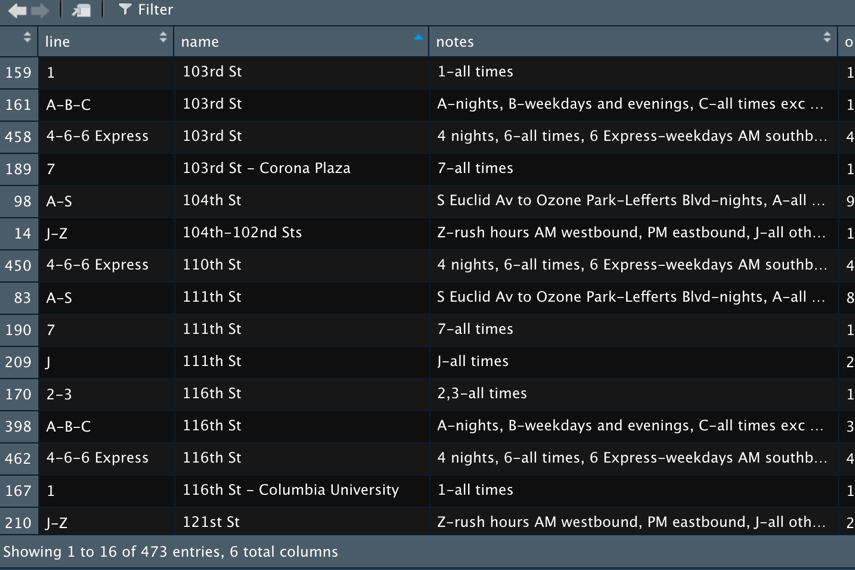
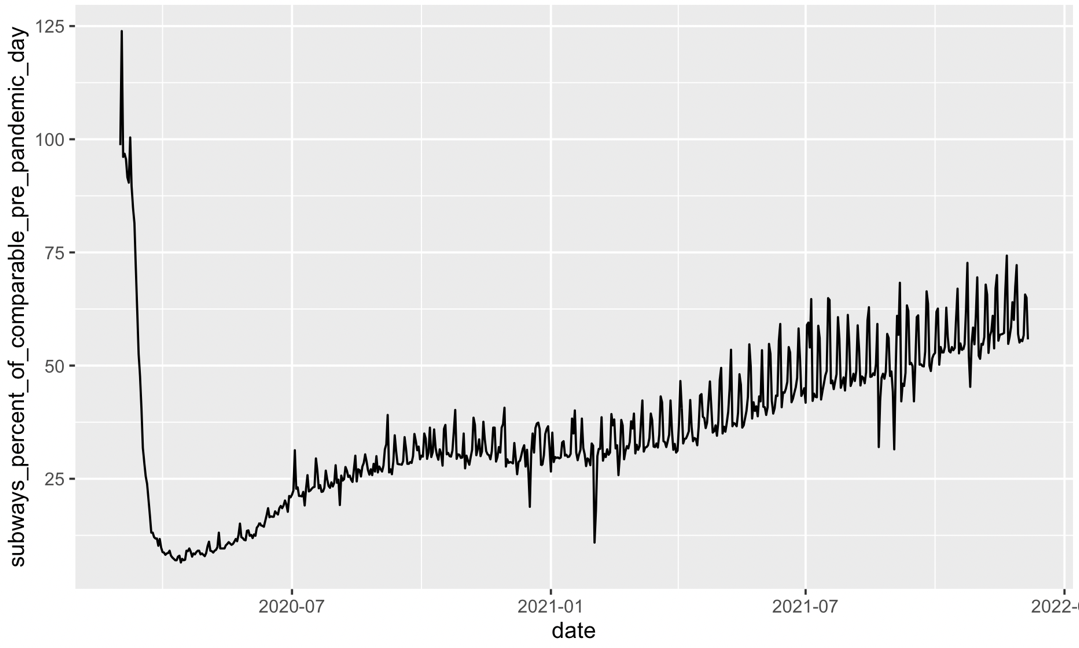
1. Fixed all the things mentioned above.
2. Added the functionality that sum the values within ‘Year-Quarter-Borough-Street\_Names-All\_Lines’ pairs, to incorporate new stations derived from ‘DISTRICT # OFFICE.’
3. Finished joining clean, stacked nypd arrests and summonses datasets
4. Created the fourth script file called ‘04\_fix\_nypd\_naming\_issues\_and\_join\_stacked\_arrests\_and\_summonses\_datasets.’ The tasks above are shown in this file.
5. Changed the unit of analysis from nypd\_names to mta\_names
   1. Created a crosswalk table
      1. Fixed three (+1) district offices I couldn’t fix earlier because multiple stations were nearby. I couldn’t fix “DISTRICT 2 OFFICE” even at this stage b/c it had multiple options for mta\_names. Options are as below:
         1. ->> Canal St (A,C,E)
         2. Canal St (J,N,Q,R,W,Z,G)
   2. Created “mta\_tidy” data using the crosswalk table.
      1. Need to see where Na values are coming from.
   3. Removed “Borough Modifications” part because the “crosswalk” table can be used for that functionality.

12/2

1. Fixed scripts to incorporate 2021 Q1-2 arrests and Q1-3 summonses datasets. 2021 Q3 arrests dataset is yet to be released.
   1. Arrests – New datasets are in the same format as the last one from 2020 Q4.
   2. Summonses – New datasets are in the same format as the last one from 2020 Q3. (Remember, we currently do not have the summonses data for 2020 Q4 because the link to download it on NYPD’s website downloads the dataset for Q3 in the same year instead.)
   3. Note that the most recent quarterly arrests and summonses datasets have the same structure, but different column names.
2. Sent a FOIL request for the correct 2020 Q4 summonses dataset.
3. Identified where NA values were coming from, and fixed it. In the fourth script, where I fixed nypd naming issues one by one, I had messed up one line of code and assigned a new correct value to a wrong column.
4. Created the fifth script called ‘05\_make\_mta\_tidy’.
5. Started adding additional notes and finding parts that can be improved. Also, started thinking of how to better cut sections, and when is best to write and reload files.
6. Did some Exploratory Data Analysis for fun. Some of the interesting findings are:
   1. Pre-covid, Blacks were more summoned than any other racial groups, but since covid, Hispanics have been more summoned than Blacks. However, when it comes to arrests, Blacks remain the most arrested group.
   2. I took a closer look at the summonses data by Borough, as defined by MTA. I found that the number of summonses was down in all boroughs in 2021, compared to pre-covid, except for the Queens. Unless the ridership in Queens remained intact, unlike the other boroughs, that indicates either that the level of Fare Evasion enforcement in Queens got more intense, or that a higher proportion of riders in Queens were caught.



12/7

1. Finished organizing directory.
   1. Created all the materials to a new folder named “NYPDFareEvasionProject” under the “Fare Evasion” folder. This will be our final folder and host all the raw datasets and codes.
2. Imported 2019 ACS 5-year block group-level racial composition and poverty status estimates, using the tidycensus package. Used tables “B03002 (Hispanic or Latino Origin by Race)” and “B17001 (Poverty Status by Sex by Age)”, “B17020 (Poverty Status by Age)”
   1. ISSUE: “B17001” and “B17020” don’t seem available at the block group-level. “B17010 (Poverty Status of Families by Family Type by Presence of Related Children)”. But the unit for B17010 is families. How do you know which table is available at which levels?
3. Downloaded and took a look at NYC Subway ‘Stations’ geospatial data.
   1. Station names are in a format completely different from what we have previously worked with. It seems easier to enter our final mta name for each station in a new column by hand and join this data with our final mta\_tidy data on that column, rather than directly joining it with our final mta\_tidy data. There are only 473 unique stations in the geospatial data, so it shouldn’t take too long.
   2. A glimpse at the data:
4. Downloaded and took a look at NYC Subway ‘Lines’ geospatial data.
   1. For mapping purpose, the subway ‘lines’ data might need to be joined with subway ‘stations’ data.
5. Downloaded and took a look at daily MTA ridership data.
   1. NOTE: The data shows the proportions of daily ridership to the equivalent pre-pandemic day between 3/1/19 and 2/29/20. **Daily numbers are available between Mar. 1, 2020 – Dec. 6, 2021 only, currently**.
   2. An exploratory analysis shows that the proportion of ridership to comparable pre-pandemic day has continuously increased, but still around 60% level.
   3. 
6. Started exploring MTA [Developer Resources](http://web.mta.info/developers/developer-data-terms.html#data). Maybe we can use some of these resources to get the most granular ridership data?

12/8

The main task for this week was data and code validation. I have read through, improved, and added notes to seven R scripts:

1. Master.R

2. 01\_load\_raw\_arrests\_and\_summonses\_quarterly\_data.R

**3.** **02-1\_get\_quarterly\_arrests\_data\_tidy.R**

4. **02-2\_get\_quarterly\_summonses\_data\_tidy.R** (Will fix after finalizing the above script)

5. 03\_stack\_tidy\_quarterly\_arrests\_and\_summonses\_data\_by\_enforcement\_type.R

6. 04\_fix\_nypd\_naming\_issues\_and\_join\_stacked\_arrests\_and\_summonses\_datasets.R

7. 05\_make\_mta\_tidy.R

I spent most of my work hours this week fixing the highlighted scripts. The two scripts share a large portion of codes, so I basically fixed the first script, and will copy and paste the fixed codes to the second script. For the other scripts, I mostly added notes. Here are things I fixed in the highlighted scripts:

**MINOR** CHANGES

1. More intuitive object naming.

i.e.) n -> max\_line\_length

2. Removed redundant codes.

i.e.) changing NA values to “”, and then switching them back to NA values.

3. Removed unnecessary object creations

i.e.) data -> data\_1 -> data\_2 -> data\_3 -> data\_4 -> data\_unknown -> data\_final

-> data -> data\_1 -> data\_2 -> data\_final

**MAJOR** CHANGES

4. Changed some codes to use more intuitive, and commonly used function.

i.e.) apply(data, 1, function(x) any(x %in% “BRONX”))

-> apply(data, 1, function(x) any(grepl(“BRONX”, x))

5. Changed street name filtering criteria

filter(SL >= 8 & str\_count(BSL, “,” <5)

-> Filter(str\_length(BSL) > 1 & !str\_detect(substr(BSL, 1, 2), “,”))

a. IMPLICATIONS: Fixing filtering criteria removed issues we earlier had, and thus, tackled in the later part of the script. So now we can get rid of a large portion of the script.

12/17

1. Gave a second look Garret’s and my codes identifying rows with a borough name.
   1. I compared following three codes:  
      1) apply(data, 1, function(x) any(**grepl**("BRONX", x))) (Yun’s)   
      2) apply(data, 1, function(x) any(x **%in%** "BRONX")) (Garret’s I) # TRUE or FALSE   
      3) apply(data, 1, function(x) any(x **==** "BRONX")) (Garret’s II) # TRUE, FALSE or NA
   2. I found an issue with **my** code. The major difference between Garret’s code and mine is that Garret’s code detects rows with the exact string “BRONX” while my code also detects rows with any string that CONTAINS the string “BRONX”. For example, unlike Garrent’s, my code detects a row with the station name “BRONX PARK EAST” as a row showing borough information “BRONX”, which is incorrect.
   3. **In conclusion, Garret’s code and mine did different things; Garret’s code was correct; so I switched the code back to Garret’s.**
   4. I also figured out why Garret had used “%in%” instead of “==” although having only one string to check. In general, “%in%” is used when having a list of strings to check. For example, “x **%in%** c("BRONX", "QUEENS", "MANHATTAN")”. Theoretically, ‘x **%in%** "BRONX"’ and ‘x **==** "BRONX"’ should do the same job telling if a row has the borough information. However, the codes produce different results. ‘x **%in%** "BRONX"’ produces either “TRUE” or “FALSE” while ‘x **==** "BRONX"’ produces NA values as well when a value in any column is NA. This difference does not affect the following codes in any way but I chose Garret’s initial code ‘x **%in%** "BRONX"’ because working with strings with NA values does not seem desirable.
   5. Is there any more intuitive code to perform the same task?
2. Fixed Lines 1:4 – 1:7 as soon as possible.
   1. Codes fixing was spread throughout scripts 02\_1, 02\_2, and 03. The earliest stage possible where the issue can be fixed is 02\_1 and 02\_2. So I fixed in 02-1 and 02-2.
   2. The code is between Line 115 and 129 in Script 02\_1.
3. Fixed the syncing issue.
   1. You could not easily access the updated version of the new project directory “NYPDFareEvasionProject” because I was working with a copied “Fare Evasion” folder, which was not synced with the one you shared with me. The reason I had created the new directory under MY Drive after copying and pasting the whole “Fare Evasion” folder to MY Drive, is because Google Drive only allowed syncing files in MY Drive, not SHARED one.
   2. SOLUTION: I created a shortcut to the original “FareEvasion” folder you had shared with me under MY Drive. That way, I can easily make changes to the directory from my local computer without creating and sharing a new directory with you.
4. Reset all the file paths used in each script to reflect the change in the directory path.
5. [IN PROGRESS] Save cleaned datasets at the end of every script.
   1. DONE: Script: 01, 02-1, 02-2, 03,
6. [IN PROGRESS] Make sure each script works.
   1. DONE: Script: 01, 02-1, 02-2, 03,

Next steps: 3) clean notes; 4) Github?

01/27

1. Fixed missing “UNKNOWN” and “OFF-SYSTEM” rows in Scripts 02-1 and 02-2.
2. Changed the way we save datasets at the end of every script, from saving individual datasets as separate csv files, to saving all of them as an RData object.
   1. Now every script starts loading files from the previous script.
   2. I also saved as separate csv files the datasets that might be used separately – ‘all\_arrests’ or ‘all\_summonses’, for example.
3. Made sure each script works!
4. Created a tool that (1) scrapes all the download links in MTA Developer Resources, and (2) downloads and merges all bi-weekly datasets, but MTA blocked my access.
5. Reached out to MTA developers to ask how to easily access their station-level weekly fare data.
6. Started a new script “06\_collect\_data\_for\_mapping.R” under “02\_mapping” folder
7. Finished collecting ACS block group-level poverty data.
   1. Out of many poverty metrics, I chose “**ratio of income to poverty level**” although it is not very intuitive because of its availability at the census block group level. (There are 6 categories, which range from under 0.50 to 2.00 and over. In general, a ratio less than 1 means that the income is less than the poverty level. When the ratio equals 1, the income and poverty level are the same, and when the ratio is greater than 1, the income is higher than the poverty level.)
   2. Out of many tables using the “ratio of income to poverty level” metric, I selected **B17002** for consistency with racial composition figures, as it shows the number of “residents”, not the number of “families.”

Next steps: 1) crosswalk ‘mta\_tidy’ to subway shape file, using Prof. Stolper’s data (can’t remember where it was); 2) police district shape files as well? ; 3) Start creating a Shiny map so that I actually know what to put on the map; 4) finish cleaning notes

CSS Fare Evasion > Data

02/03

1. Explore various NYC subway stations and lines resources and choose the best shapefile.
   1. NYC Open Data
2. Create a crosswalk table from ‘mta\_tidy’ to subway stations and lines shapefiles
   1. Source: (1) NYC Open Data – ‘stations’ last updated on Aug. 2019, ‘lines’ last update on Sep. 2018; (2) CUNY updates subway stops and routes at least once a year. Last updated in May, 2020.
3. Changes in s- new openings? Shutdowns?
4. Subway first ->