nycsubwayguide.com. (n.d.).

GEOG 596B

# Visualize Subway Turnstile Data

Capstone Project

Peter Rechtlich
5-12-2020
Advisor: Dr. James O’Brien

# Abstract

### *Background*

For the first time in its history, the New York City Subway System has suspended 24/7 operations due to the 2020 Global Pandemic. The associated virus thrives for extended periods of time on metal surfaces and consequently, cleaning operations on the metal interiors of all NYC Subway Cars will be performed between 1:00 am to 05:00 am time period every day for the foreseeable future.

This project started out with a focus on tracking subway data, specifically focusing on tracking commuter congestion along Subway Trainline 7 during the months of the U.S. Open Tennis Grand Slam Tennis Championships. Due to these unprecedented times, and the fact that major sports venues are being cancelled (including the 2020 French and Wimbledon Grand Slam Tennis Championships), the focus of the data for this project was redirected to tracking commuter flow during the first four months of 2020.

### *Method*

The initial method for this project was to use the Python programming language to extract and transform the data and then use AppStudio for ArcGIS for data analysis/presentation. Instead, project direction shifted toward using the R programming language and the Tableau Data Visualization Development Environment Suite, respectively.

### *Results*

A data visualization application has been uploaded to the web through infrastructure set in place via Tableau technology. This application is interactive and has the capability to visualize the information in an animated sequence.

### *Conclusions*

A good amount of additional training in software application development capabilities was needed to get an online solution out onto the web. There were also new resources uncovered that will provide further growth in data visualization understanding. Another takeaway was the availability of online data visualization communities (check out Nathan Yau's FlowingData Community, Chart Chats with 'The Big Book of Dashboards' folks, and Makeover Monday).

I want to thank my Penn State Capstone Advisor Dr. James O'Brien for his guidance and advice, all my Penn State instructors throughout the Penn State MGIS experience for their passion and commitment, and my wife Kelly, my brothers Chris and Walt, and the rest of my family for their support in helping me through the work/school/life balance.

# Background

Penn State GEOG486 ‘CARTOGRAPHY AND VISUALIZATION’ was the inspiration for this project. Back then, emphasis during the course was on using ESRI’s ArcMap software application and using it to learn how to communicate through maps (i.e. map types, map design, typographical design, layout essentials, building a legend, choosing map symbols, visual encoding, designing for multiple map scales, types of color schemes, choosing projections, building terrain layers, etc.). Online content for that same course currently has a section within Lesson 9 called ‘When Not to Map’. A paragraph within this section resonates around this project.

> “When designing data visualizations, maps often provide an invaluable source for insight generation. However, they are not necessarily always the best choice for your data – even if the data contain spatial information.”

Indeed, this project experienced going beyond the boundaries of map presentation to help answers beyond the map. Some of the charts for dashboard visualizations include time-series line charts, enhanced bubble charts, line charts, bar charts, and symbol maps. Below is a sampling of the charts just mentioned, created using a ‘Sample Superstore’ dataset that comes associated to the Tableau Desktop Data Visualization Development Environment.

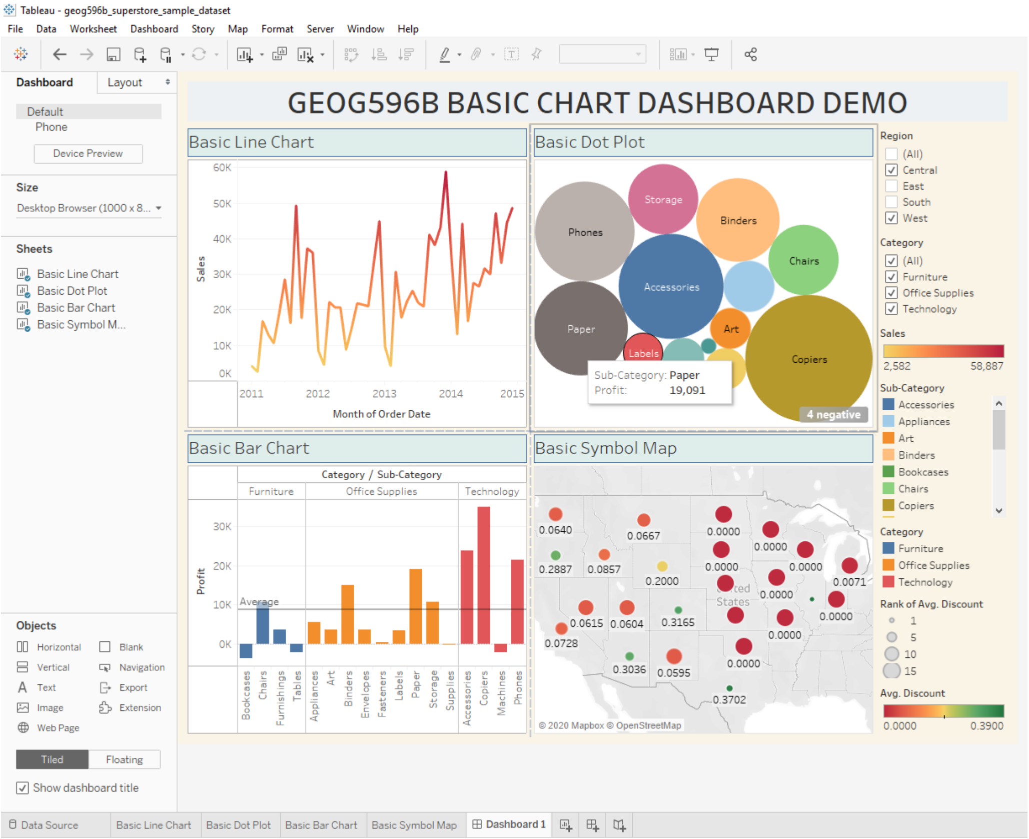


*Figure 1- Dashboard displaying Sample Superstore data*

Additional knowledge/insight was acquired throughout the project by studying soft skills associated to data visualization (i.e. historical contributions, philosophy, and fundamentals).

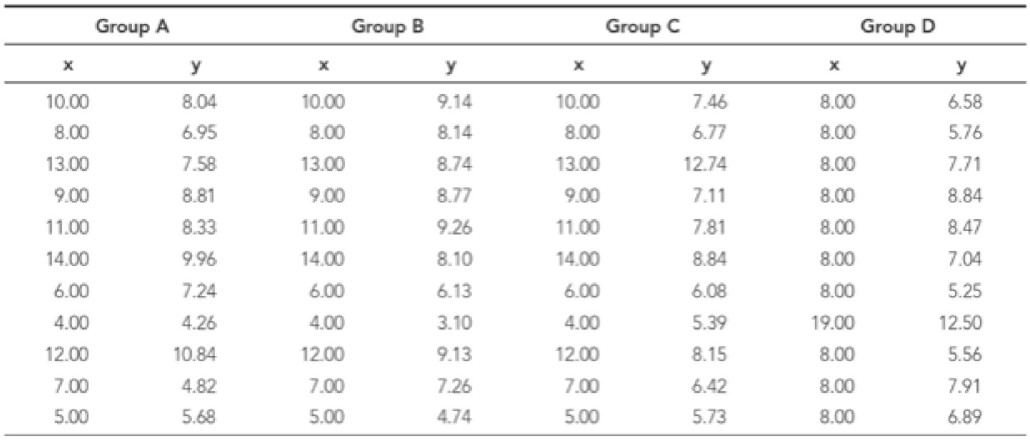
## Soft Skill Takeaways

### *Historical*

Important data visualization pioneers that provided perspective to the efforts of this project included Rene Descartes (inventor of the 2-D coordinate system), Jacque Bertins (first to present proper use of visual variables; GEOG486 'Lesson 1, Section: Symbol Design), and Edward Tufte (showcased Minard's Napoleon March map in self-published books on Information Graphics; proponent against use of Pie Charts; GEOG486 'Lesson 5', Section: Flow Mapping/Multivariate Glyphs).

### *Philosophy*

Frank Anscombe (British Statistician; creator Anscombe's quartet) demonstrated how four datasets that are nearly identical with simple descriptive statistics will appear very different when graphed.



| Group A | | Group B | | Group C | | Group D | |
|---|---|---|---|---|---|---|---|
| x | y | x | y | x | y | x | y |
| 10.00 | 8.04 | 10.00 | 9.14 | 10.00 | 7.46 | 8.00 | 6.58 |
| 8.00 | 6.95 | 8.00 | 8.14 | 8.00 | 6.77 | 8.00 | 5.76 |
| 13.00 | 7.58 | 13.00 | 8.74 | 13.00 | 12.74 | 8.00 | 7.71 |
| 9.00 | 8.81 | 9.00 | 8.77 | 9.00 | 7.11 | 8.00 | 8.84 |
| 11.00 | 8.33 | 11.00 | 9.26 | 11.00 | 7.81 | 8.00 | 8.47 |
| 14.00 | 9.96 | 14.00 | 8.10 | 14.00 | 8.84 | 8.00 | 7.04 |
| 6.00 | 7.24 | 6.00 | 6.13 | 6.00 | 6.08 | 8.00 | 5.25 |
| 4.00 | 4.26 | 4.00 | 3.10 | 4.00 | 5.39 | 19.00 | 12.50 |
| 12.00 | 10.84 | 12.00 | 9.13 | 12.00 | 8.15 | 8.00 | 5.56 |
| 7.00 | 4.82 | 7.00 | 7.26 | 7.00 | 6.42 | 8.00 | 7.91 |
| 5.00 | 5.68 | 5.00 | 4.74 | 5.00 | 5.73 | 8.00 | 6.89 |

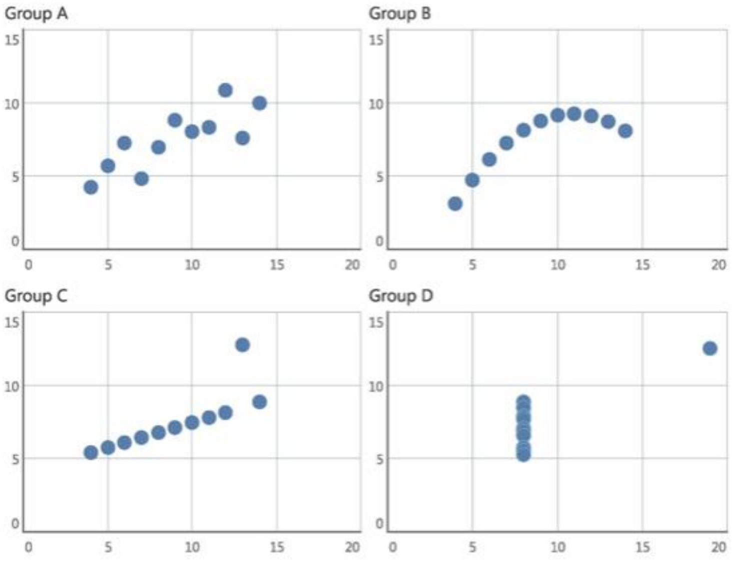
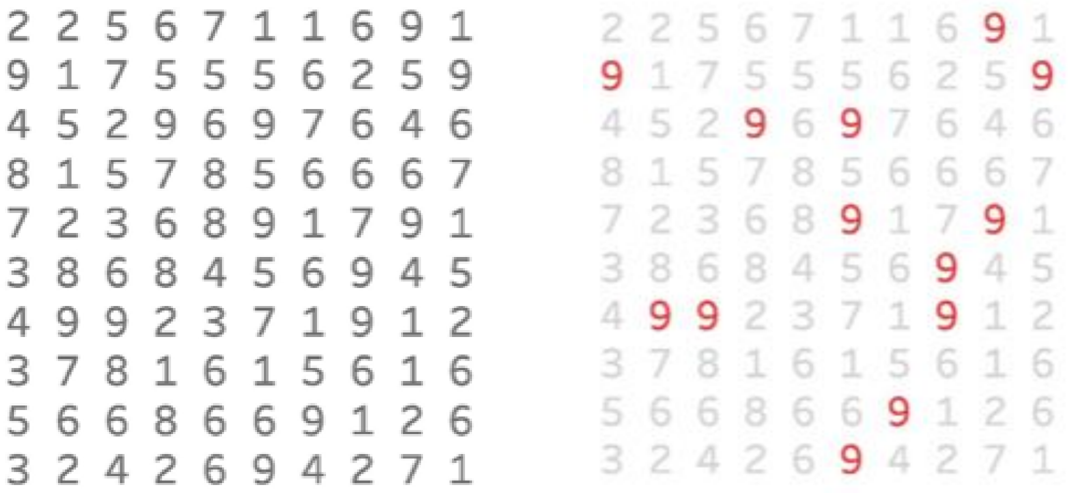


*Figure 2 – Anscombe's Quartet (Wexler, Shaffer, Cotgreave.2017)*

*Fundamentals*

Pre-attentive attributes can be described as "…things our brains process in milliseconds, before we pay attention to everything else" (Wexler, Shaffer, Cotgreave. 2017). How/if preattentive attributes are implemented literally determines how viewers will interpret the communique behind a data visualization. Example: Below is a 9 x 9 matrix of random integers from 1 to 9. Find all the 9's in each matrix. Which way is quicker?



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 5 | 6 | 7 | 1 | 1 | 6 | 9 | 1 |
| 9 | 1 | 7 | 5 | 5 | 5 | 6 | 2 | 5 | 9 |
| 4 | 5 | 2 | 9 | 6 | 9 | 7 | 6 | 4 | 6 |
| 8 | 1 | 5 | 7 | 8 | 5 | 6 | 6 | 6 | 7 |
| 7 | 2 | 3 | 6 | 8 | 9 | 1 | 7 | 9 | 1 |
| 3 | 8 | 6 | 8 | 4 | 5 | 6 | 9 | 4 | 5 |
| 4 | 9 | 9 | 2 | 3 | 7 | 1 | 9 | 1 | 2 |
| 3 | 7 | 8 | 1 | 6 | 1 | 5 | 6 | 1 | 6 |
| 5 | 6 | 6 | 8 | 6 | 6 | 9 | 1 | 2 | 6 |
| 3 | 2 | 4 | 2 | 6 | 9 | 4 | 2 | 7 | 1 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 2 | 2 | 5 | 6 | 7 | 1 | 1 | 6 | 9 | 1 |
| 9 | 1 | 7 | 5 | 5 | 5 | 6 | 2 | 5 | 9 |
| 4 | 5 | 2 | 9 | 6 | 9 | 7 | 6 | 4 | 6 |
| 8 | 1 | 5 | 7 | 8 | 5 | 6 | 6 | 6 | 7 |
| 7 | 2 | 3 | 6 | 8 | 9 | 1 | 7 | 9 | 1 |
| 3 | 8 | 6 | 8 | 4 | 5 | 6 | 9 | 4 | 5 |
| 4 | 9 | 9 | 2 | 3 | 7 | 1 | 9 | 1 | 2 |
| 3 | 7 | 8 | 1 | 6 | 1 | 5 | 6 | 1 | 6 |
| 5 | 6 | 6 | 8 | 6 | 6 | 9 | 1 | 2 | 6 |
| 3 | 2 | 4 | 2 | 6 | 9 | 4 | 2 | 7 | 1 |

*Figure 3 – Pre-attentive attributes example (Wexler, Shaffer, Cotgreave.2017).*

## Additional training

If I had eight hours to chop down a tree,
I'd spend six sharpening my axe.

- ABRAHAM LINCOLN

The path toward getting the necessary tech skills for this data visualization project required building upon Python programming skills developed during GEOG485 'GIS PROGRAMMING AND SOFTWARE DEVELOPMENT' as well as building upon 'R' programming skills first introduced in' in GEOG586 'GEOGRAPHIC INFORMATION ANALYSIS'. 'Point Pattern Analysis (PPA).

Below is a chart of training resources explored during this project. There were also some online video tutorials that Tableau puts out that were quite helpful.

*Table 1 - Data Visualization training*

| Order | Technology | Additional training |
|---|---|---|
| 1 | Python programming | DataCamp module(s): Introduction to Python, Intermediate Python, Importing Data in Python (Part 1), Introduction to Data Science, Cleaning Data in Python, pandas Foundation, Introduction to Data Visualization in Python, Time Series Analysis in Python, Manipulating DataFrames with pandas, Statisical Thinking in Python, Web Scraping in Python |
| 2 | R programming | DataCamp module(s): Introduction to R, Cleaning Data in R, Introduction to the TidyVerse, Data Manipulation in R with dplyr, Importing Data in R (Part 1), Introduction to Data Visualization with ggplot2, Intermediate R, Joining Data in R; with dplyr, Exploratory Data Analysis in R: Case Study, Working with Dates and Times in R, Times Series Analysis in R; joined as a member of FLOWINGDATA (a lot of R programming examples associated to R) |
| 3 | Tableau Desktop | DataCamp module(s): Introduction to Tableau; virtual online training certificate from St. Louis University Workforce (TAB.PRO – TAB100, TAB200) |
| 4 | Tableau Prep | Online training videos through Tableau website (https://www.tableau.com/learn/get-started) |
| 5 | Microsoft Excel | No additional training taken |
| 6 | Power BI | virtual online training certificate from St. Louis University Workforce (DA1.PRO – BID615, TAB100, PYT100) ~ tool not implemented for this project (similar to Tableau Desktop) |

## Objective

The objective of this project was to create an interactive mobile application that tracks New York City commuter congestion for subway stations along the NYC MTA Subway Trainline 7 route within the New York City Subway System. This route runs regularly from downtown Times Square out east to the neighborhood of Flushing in the New York City of Queens.

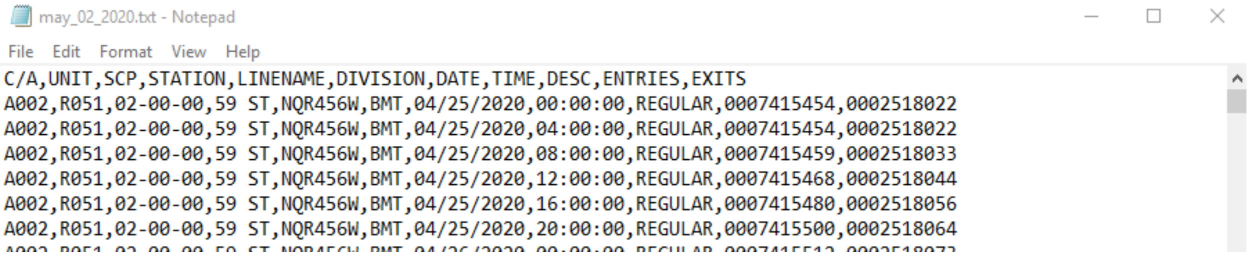
## Description of the Data

The data used for this project is New York City subway turnstile data that gets generated by the Metropolitan Transportation Authority (MTA) 'Developer' website. Turnstiles are mechanical/electronic units that have been installed at the entrances to the different subway stations spread out across the New York City Subway System. They attempt to record every commuter that enters the station, which is consequently electronically captured and consolidated into one of the six 4-hour audit time periods depending on commuter entry time. A weekly report of all turnstile data then is generated and posted online within t the Metropolitan Transportation System (MTS) ‘Developer’ website.

The descriptions of the dataset fields, as described from the MTA website, are as follows:

- C/A – Control Area
- UNIT – Remote Unit for a station
- SCP – Subunit Channel Position represent a specific address for a device
- STATION – Represents the station name where the device is located at
- LINENAME – Represents all train lines that can be boarded at this station
- DIVISION – Represents the ‘Line’ that the station originally belonged to (i.e. BMT, IRT, IND)
- DATE – Represents the data (MM-DD-YY)
- TIME – Represents the time (hh:mm:ss) for a scheduled audit event
- DESC – Represents the ‘REGULAR’ audit event (which normally occurs every 4 hours)
    - Audits may occur at more than the 4 hour intervals due to planning, or troubleshooting activities
    - There may be a ‘RECOVR AUD’ entry – this refers to a missed audit that was recovered
- ENTRIES – the cumulative entry register value for a device
- EXITS – the cumulative exit register value for a device

A representative sample of what this data looks like is shown below. There are 11 fields but only a couple of them are needed for this visualization. The key columns for analysis are ‘UNIT’, ‘SCP’, STATION’ and ‘ENTRIES’.



```
C/A,UNIT,SCP,STATION,LINENAME,DIVISION,DATE,TIME,DESC,ENTRIES,EXITS
A002,R051,02-00-00,59 ST,NQR456W,BMT,04/25/2020,00:00:00,REGULAR,0007415454,0002518022
A002,R051,02-00-00,59 ST,NQR456W,BMT,04/25/2020,04:00:00,REGULAR,0007415454,0002518022
A002,R051,02-00-00,59 ST,NQR456W,BMT,04/25/2020,08:00:00,REGULAR,0007415459,0002518033
A002,R051,02-00-00,59 ST,NQR456W,BMT,04/25/2020,12:00:00,REGULAR,0007415468,0002518044
A002,R051,02-00-00,59 ST,NQR456W,BMT,04/25/2020,16:00:00,REGULAR,0007415480,0002518056
A002,R051,02-00-00,59 ST,NQR456W,BMT,04/25/2020,20:00:00,REGULAR,0007415500,0002518064
```

*Figure 4- NYC MTA Subway Turnstile Data text file*

## Problem Statement

There were three primary obstacles to overcome in this project.

### Obstacle #1 – Data Acquisition/Transformation

There are two parts to the first obstacle:

- Find a suitable method to extract the data from the New York City Metropolitan Transportation Authority’s ‘Developer’ website
- Transform the acquired data such that commuter counts are captured via an appropriate differential calculation method

## Obstacle #2 – Data Analysis/Visualization

The second challenge was about finding an appropriate development environment and then efficiently implementing an appropriate analysis of the data.

## Obstacle # - Data Presentation

The third challenge was about getting the data visualization solution into an interactive format and posted to the web

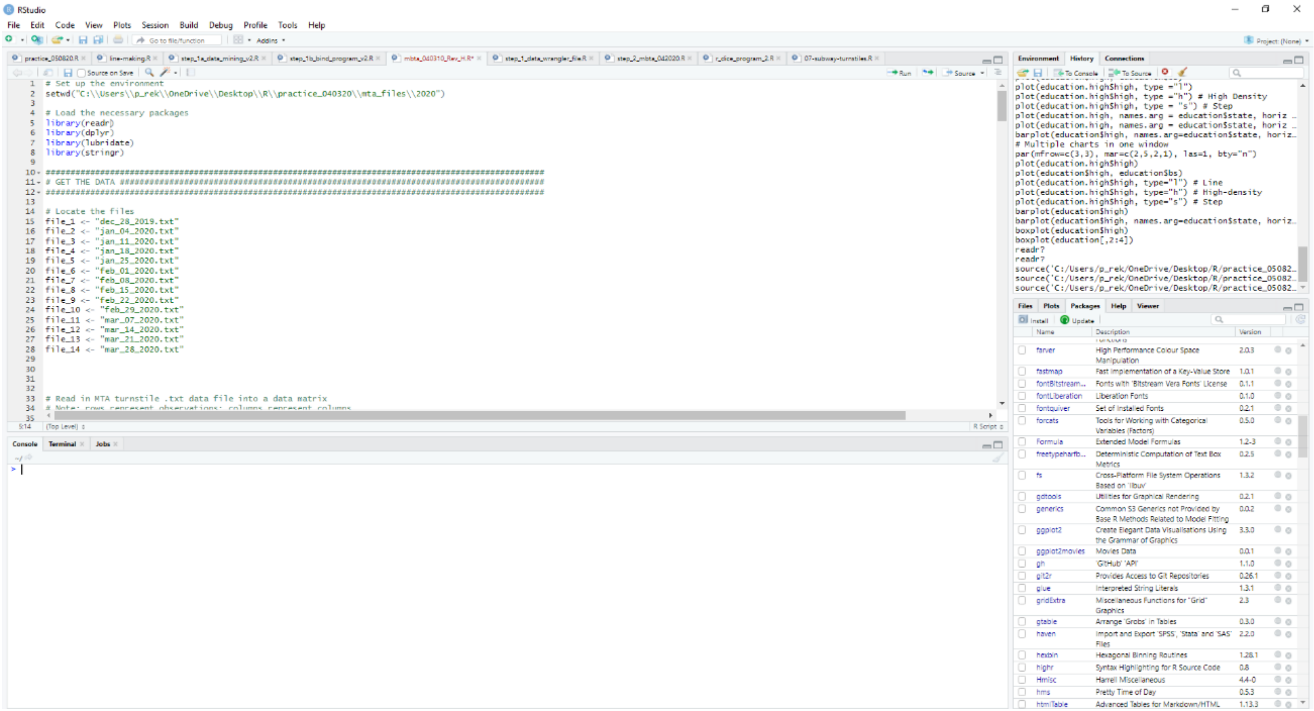
# Methodology

The work that needed to be done included capturing data, transforming it, cleaning it, analyzing it, visualizing it, and then presenting it.

## R programming

The project went with R programming to capture and transform the data. A couple of R packages worth remembering (courtesy mostly from referencing of the R Studio data wrangling cheat sheet): https://rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf .

R Studio was used as the development environment to manage R code for this project.



```
# Set up the environment
setwd("C:\\Users\\p_rek\\OneDrive\\Desktop\\R\\practice_040320\\mta_files\\2020")

# Load the necessary packages
library(readr)
library(dplyr)
library(lubridate)
library(stringr)

##########################################################################################
# GET THE DATA ###########################################################################
##########################################################################################

# Locate the files
file_1 <- "dec_28_2019.txt"
file_2 <- "jan_04_2020.txt"
file_3 <- "jan_11_2020.txt"
file_4 <- "jan_18_2020.txt"
file_5 <- "jan_25_2020.txt"
file_6 <- "feb_01_2020.txt"
file_7 <- "feb_08_2020.txt"
file_8 <- "feb_15_2020.txt"
file_9 <- "feb_22_2020.txt"
file_10 <- "feb_29_2020.txt"
file_11 <- "mar_07_2020.txt"
file_12 <- "mar_14_2020.txt"
file_13 <- "mar_21_2020.txt"
file_14 <- "mar_28_2020.txt"




# Read in MTA turnstile .txt data file into a data matrix
# Note: rows represent observations; columns represent columns
```

```
> |
```

```
plot(education.high$high, type ="l")
plot(education.high$high, type ="h") # High Density
plot(education.high$high, type = "s") # Step
plot(education.high, names.arg = education$state, horiz …
plot(education.high, names.arg = education$state, horiz …
barplot(education$high, names.arg=education$state, horiz…
# Multiple charts in one window
par(mfrow=c(3,3), mar=c(2,5,2,1), las=1, bty="n")
plot(education.high$high)
plot(education$high, education$bs)
plot(education.high$high, type="l") # Line
plot(education.high$high, type="h") # High-density
plot(education.high$high, type="s") # Step
barplot(education$high)
barplot(education$high, names.arg=education$state, horiz…
boxplot(education$high)
boxplot(education[,2:4])
readr?
readr?
source('C:/Users/p_rek/OneDrive/Desktop/R/practice_05082…
source('C:/Users/p_rek/OneDrive/Desktop/R/practice_05082…
source('C:/Users/p_rek/OneDrive/Desktop/R/practice_05082…
```

| | Name | Description | Version |
|---|---|---|---|
| ☐ | farver | High Performance Colour Space Manipulation | 2.0.3 |
| ☐ | fastmap | Fast Implementation of a Key-Value Store | 1.0.1 |
| ☐ | fontBitstream… | Fonts with 'Bitstream Vera Fonts' License | 0.1.1 |
| ☐ | fontLiberation | Liberation Fonts | 0.1.0 |
| ☐ | fontquiver | Set of Installed Fonts | 0.2.1 |
| ☐ | forcats | Tools for Working with Categorical Variables (Factors) | 0.5.0 |
| ☐ | Formula | Extended Model Formulas | 1.2-3 |
| ☐ | freetypeharfb… | Deterministic Computation of Text Box Metrics | 0.2.5 |
| ☐ | fs | Cross-Platform File System Operations Based on 'libuv' | 1.3.2 |
| ☐ | gdtools | Utilities for Graphical Rendering | 0.2.1 |
| ☐ | generics | Common S3 Generics not Provided by Base R Methods Related to Model Fitting | 0.0.2 |
| ☐ | ggplot2 | Create Elegant Data Visualisations Using the Grammar of Graphics | 3.3.0 |
| ☐ | ggplot2movies | Movies Data | 0.0.1 |
| ☐ | gh | 'GitHub' 'API' | 1.1.0 |
| ☐ | git2r | Provides Access to Git Repositories | 0.26.1 |
| ☐ | glue | Interpreted String Literals | 1.3.1 |
| ☐ | gridExtra | Miscellaneous Functions for "Grid" Graphics | 2.3 |
| ☐ | gtable | Arrange 'Grobs' in Tables | 0.3.0 |
| ☐ | haven | Import and Export 'SPSS', 'Stata' and 'SAS' Files | 2.2.0 |
| ☐ | hexbin | Hexagonal Binning Routines | 1.28.1 |
| ☐ | highr | Syntax Highlighting for R Source Code | 0.8 |
| ☐ | Hmisc | Harrell Miscellaneous | 4.4-0 |
| ☐ | hms | Pretty Time of Day | 0.5.3 |
| ☐ | htmlTable | Advanced Tables for Markdown/HTML | 1.13.3 |

*Figure 5- The R Studio Development Environment*

### *Downloading files*

When setting up a new program script in R, it seemed helpful to always include code that configures the environment. Also necessary at the start of each script was the installation of R packages. These packages consequently pull in functions which thus extend the coding capabilities for the script.

```
# Set up the environment

setwd("C:\\Users\\Desktop\\mgis\\geog596B\\part_1a_data_mining_with_download_file\\mta_files")

# Load packages

library(util)
```

After setting up the environment and calling the packages, the process was pull the data. After some trial and error, the download.file() function ended up being used for this purpose.

```
# Download the text files

download.file("http://web.mta.info/developers/data/nyct/turnstile/turnstile_200111.txt", "jan_11_2020.txt")
download.file("http://web.mta.info/developers/data/nyct/turnstile/turnstile_200118.txt", "jan_18_2020.txt")
download.file("http://web.mta.info/developers/data/nyct/turnstile/turnstile_200125.txt", "jan_25_2020.txt")
:
:
download.file("http://web.mta.info/developers/data/nyct/turnstile/turnstile_200328.txt", "mar_28_2020.txt")
```

### *Getting a feel for the data*

Early in the data cleaning routine, it proved helpful to utilize functions that gave some understanding of the data.

```
# Verify that turnstile files are in data.frame format
class(totals)

# Check the shape of the dataframe being used to house turnstile data
dim(totals)

# View the column names of the turnstile dataframes
names(totals)

# View the structure of the turnstile data
str(totals)

# Look at the structure using dplyr:: glimpse()
glimpse(totals)
```

Along with understanding the structure of the data, the following functions were used to get a snapshot of what the data itself looked like.

```
# Display the first 6 rows of the data
head(totals)
```

```
# Display the last 10 rows of the data
tail(totals, n = 10)
```

The functions below are a group of functions used for helping find missing values

```
# Count missing values
sum(is.na(totals))

# Find missing values
summary(totals)

# Find indices of NAs within the 'ENTRIES' column
ind <- which(is.na(totals$ENTRIES))

# Look at the full rows for records missing a date
totals[ind,]
```

### *Transforming the data*

The code below reflects some sample code of the steps used to transform the text files into clean datasets. First off, loading in some popular data wrangling packages:

```
# Load the necessary packages

library(readr)
library(dplyr)
library(lubridate)
library(stringr)
```

Step 1: Text files got assigned to a variable (note: 14 files in the actual code)

```
# Locate the files
file_1 <- "dec_28_2019.txt"
file_2 <- "jan_04_2020.txt"
:
file_14 <- "mar_28_2020.txt"
```

Step 2: Text files were then converted into a data matrixes (note: 14 files in actual code)

```
# Read in MTA turnstile .txt data file into a data matrix
raw_turnstile_data_1 <- read.delim(file_1, header = TRUE, sep = ",")
raw_turnstile_data_2 <- read.delim(file_2, header = TRUE, sep = ",")
:
raw_turnstile_data_14 <- read.delim(file_14, header = TRUE, sep = ",")
```

Step 3: Data matrixes were converted into data frames (note: 14 files in actual code)

```
# Create dataframes for each of the files
turnstile_1 <- data.frame(raw_turnstile_data_1)
turnstile_2 <- data.frame(raw_turnstile_data_2)
:
```

```
turnstile_14 <- data.frame(raw_turnstile_data_14)
```

Step 4: Columns were renamed  (note: 14 files in actual code)

```
# Rename the 'C.A.' column in each dataframe to 'BOOTH'
# (i.e. the first column of the dataframe)
colnames(turnstile_1)[1] <- "BOOTH"
colnames(turnstile_2)[1] <- "BOOTH"
:
colnames(turnstile_14)[1] <- "BOOTH"
```

Step 6: Filtering operations were performed on the ‘DESC’ column to capture valid audit records  (note: 14 files in actual code)

```
# Filter the 'Description' variable...keep only 'REGULAR' or 'RECOVR AUD' observation values
turnstile_1 <- turnstile_1[which(turnstile_1$DESC == "REGULAR" | turnstile_1$DESC == "RECOVR AUD"),]
turnstile_2 <- turnstile_2[which(turnstile_2$DESC == "REGULAR" | turnstile_2$DESC == "RECOVR AUD"),]
:
turnstile_14 <- turnstile_14[which(turnstile_14$DESC == "REGULAR" | turnstile_14$DESC == "RECOVR AUD"),]
```

Step 7:  Turnstile units were grouped (note: 14 files in actual code)

```
# Group the turnstiles (note: the first argument is the data to be operated on)
turnstile_1$diff <- ave(turnstile_1$ENTRIES, turnstile_1$BOOTH, turnstile_1$SCP, FUN=function(x) c(0, diff(x)))
turnstile_2$diff <- ave(turnstile_2$ENTRIES, turnstile_2$BOOTH, turnstile_2$SCP, FUN=function(x) c(0, diff(x)))
:
turnstile_14$diff <- ave(turnstile_14$ENTRIES, turnstile_14$BOOTH, turnstile_14$SCP, FUN=function(x) c(0, diff(x)))
```

Step 8:  Additional clean up was performed (note: 14 files in actual code)

```
# Remove negative entries
turnstile_1 <- turnstile_1[which(turnstile_1$diff > 0),]
turnstile_2 <- turnstile_2[which(turnstile_2$diff > 0),]
:
turnstile_14 <- turnstile_14[which(turnstile_14$diff > 0),]
```

Step 9:  Three terminals were chosen in the downtown Manhattan (high congestion) and one in an outlying neighborhood in the Queens borough of New York City (note: 14 files in actual code)

```
# Select Terminals
terminals_1 <- turnstile_1[which(turnstile_1$STATION == "TIMES SQ-42 ST" | turnstile_1$STATION == "42 ST-PORT AUTH" | turnstile_1$STATION == "GRD CNTRL-42 ST" | turnstile_1$STATION == "METS-WILLETS PT"),]
terminals_2 <- turnstile_2[which(turnstile_2$STATION == "TIMES SQ-42 ST" | turnstile_2$STATION == "42 ST-PORT AUTH" | turnstile_2$STATION == "GRD CNTRL-42 ST" | turnstile_2$STATION == "METS-WILLETS PT"),]
:
terminals_14 <- turnstile_14[which(turnstile_14$STATION == "TIMES SQ-42 ST" | turnstile_14$STATION == "42 ST-PORT AUTH" | turnstile_14$STATION == "GRD CNTRL-42 ST" | turnstile_14$STATION == "METS-WILLETS PT"),]
```

Step 10:  Calculation were performed to get turnstile ‘Interval’ data (note: 14 files in actual code)

```
# Sum all the entries by day.
# The first parameter of aggregate defines the subset of data, in this case, the diff, STATION and DATE
# The second parameter of aggregate is the dataframe.
# The third parameter is the summary statistic, in this case the sum of the subset
# The final parameter defines what to do for missing values (na.rm = TRUE removes missing values)
daily_entries_1 <-
aggregate(cbind(terminals_1$diff)~terminals_1$STATION+terminals_1$DATE+terminals_1$TIME,
data=terminals_1, sum, na.rm=TRUE)
daily_entries_2 <-
aggregate(cbind(terminals_2$diff)~terminals_2$STATION+terminals_2$DATE+terminals_2$TIME,
data=terminals_2, sum, na.rm=TRUE)
:
daily_entries_14 <-
aggregate(cbind(terminals_14$diff)~terminals_14$STATION+terminals_14$DATE+terminals_14$TIME,
data=terminals_14, sum, na.rm=TRUE)
```

Step 11: Column names were renamed (note: only first of 14 files in actual code shown here)

```
colnames(daily_entries_1)[1] <- "STATION"
colnames(daily_entries_1)[2] <- "DATE"
colnames(daily_entries_1)[3] <- "TIME"
colnames(daily_entries_1)[4] <- "ENTRIES"
```

Step 12: The rbind function was used to seam together the 14 files that would represent the 2020 dataset

```
# Combine the data for all fourteen files
totals <- rbind(daily_entries_1, daily_entries_2, daily_entries_3,
            daily_entries_4, daily_entries_5, daily_entries_6,
            daily_entries_7, daily_entries_8, daily_entries_9,
            daily_entries_10, daily_entries_11, daily_entries_12,
            daily_entries_13, daily_entries_14)
```

Step 13: The lubridate package was used to format the date column

```
# Convert date column to proper date format using lubridate's ymd()
totals$DATE <- mdy(totals$DATE)
```

Step 14: Generated the excel spreadsheet representing a clean dataset

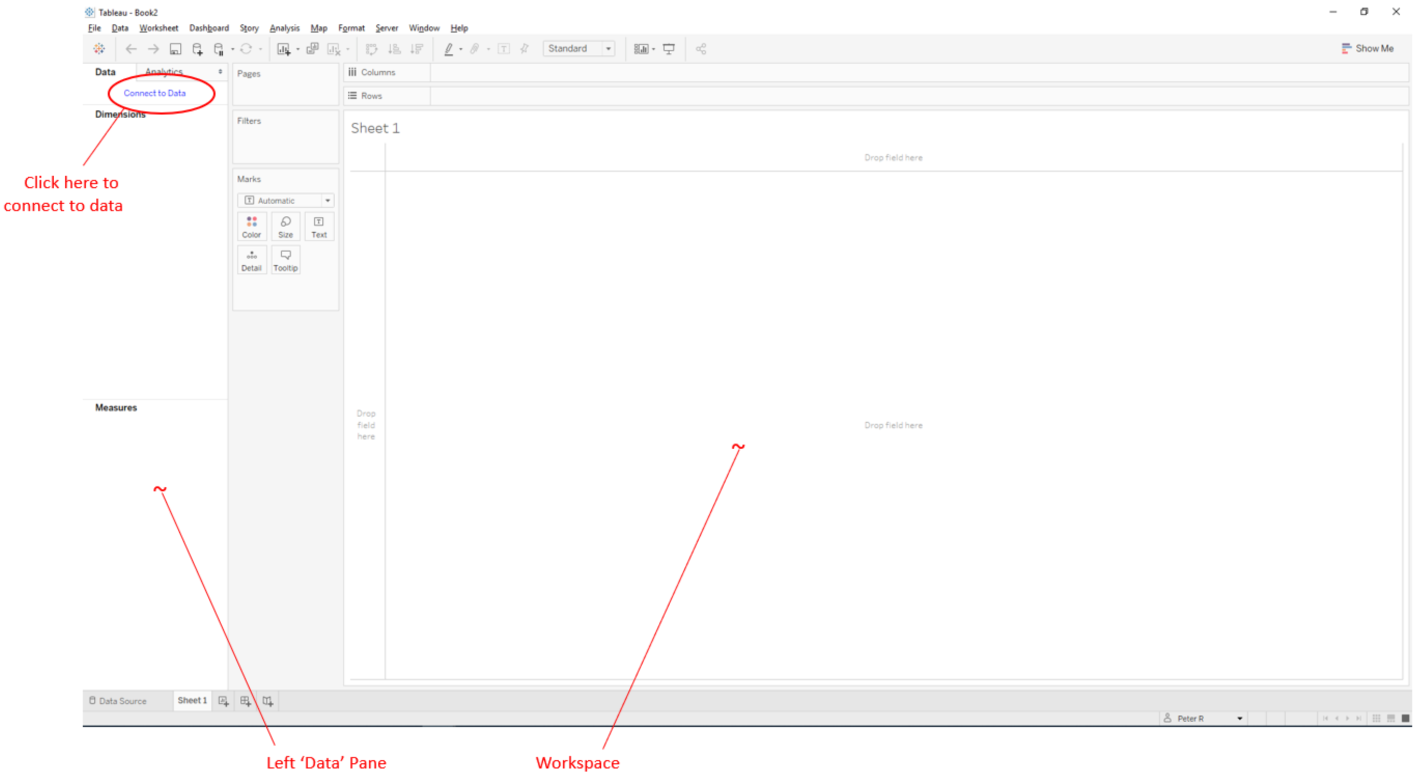
```
# Create new, improved file

write.xlsx(totals,'mta_2020.xlsx')
```

It should be noted that these steps had to be performed 3 separate times to capture tracking curves for the years of 2018, 2019, and 2020.

## Tableau Desktop

Once the data was cleaned and ready to go, the data was pulled into Tableau Desktop for data analysis and presentation. Tableau is a software application development environment popular for data analysis and data visualization. It is versatile in its ability to pull in data sources for a large variety of data formats. This project focused on working with .txt, .csv, and .xlsx files. This project tried to follow the mantra of keeping data acquisition/manipulation (i.e. Tableau Prep Builder/Microsoft Excel) and data analysis/visualization efforts (i.e. Tableau Desktop) separate if possible.

When first opening a new Tableau Desktop 'Workbook', the first window that normally gets displayed is a blank worksheet (i.e. Sheet 1). The Tableau Desktop worksheet format is similar to the Microsoft Excel worksheet tab layout. No data is yet available in the left 'Data' Panel. Clicking the link called 'Connect to Data' opens a separate blue option panel offering a multitude of connection options to load data into the Tableau Desktop application. A resulting upload operation will display the 'Added Data' under the 'Connections' column in the left navigation panel for the 'Data Source' tab.



Click here to connect to data

Left 'Data' Pane

Workspace

*Figure 6 - Blank Tableau Desktop Application Environment*

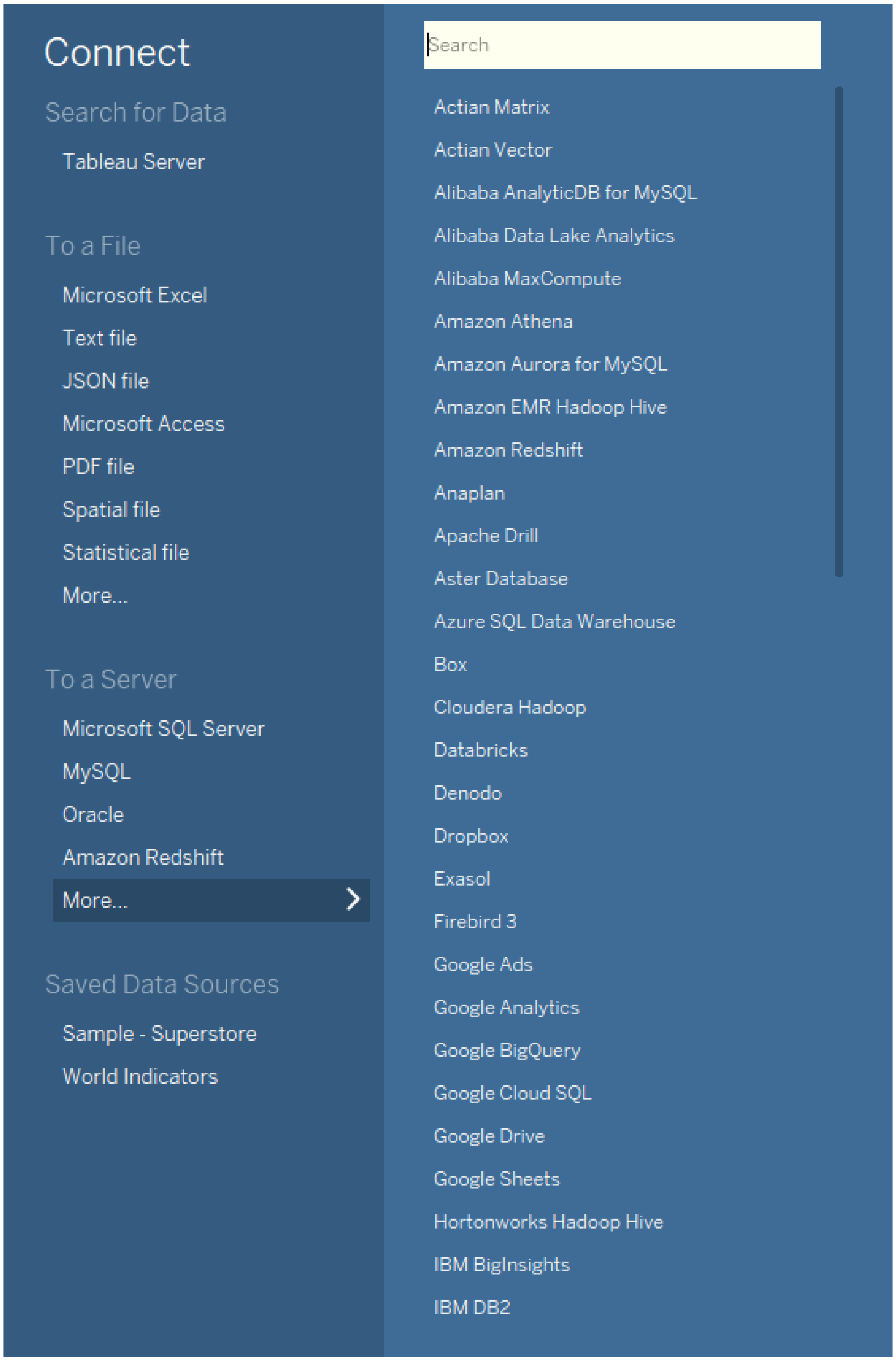
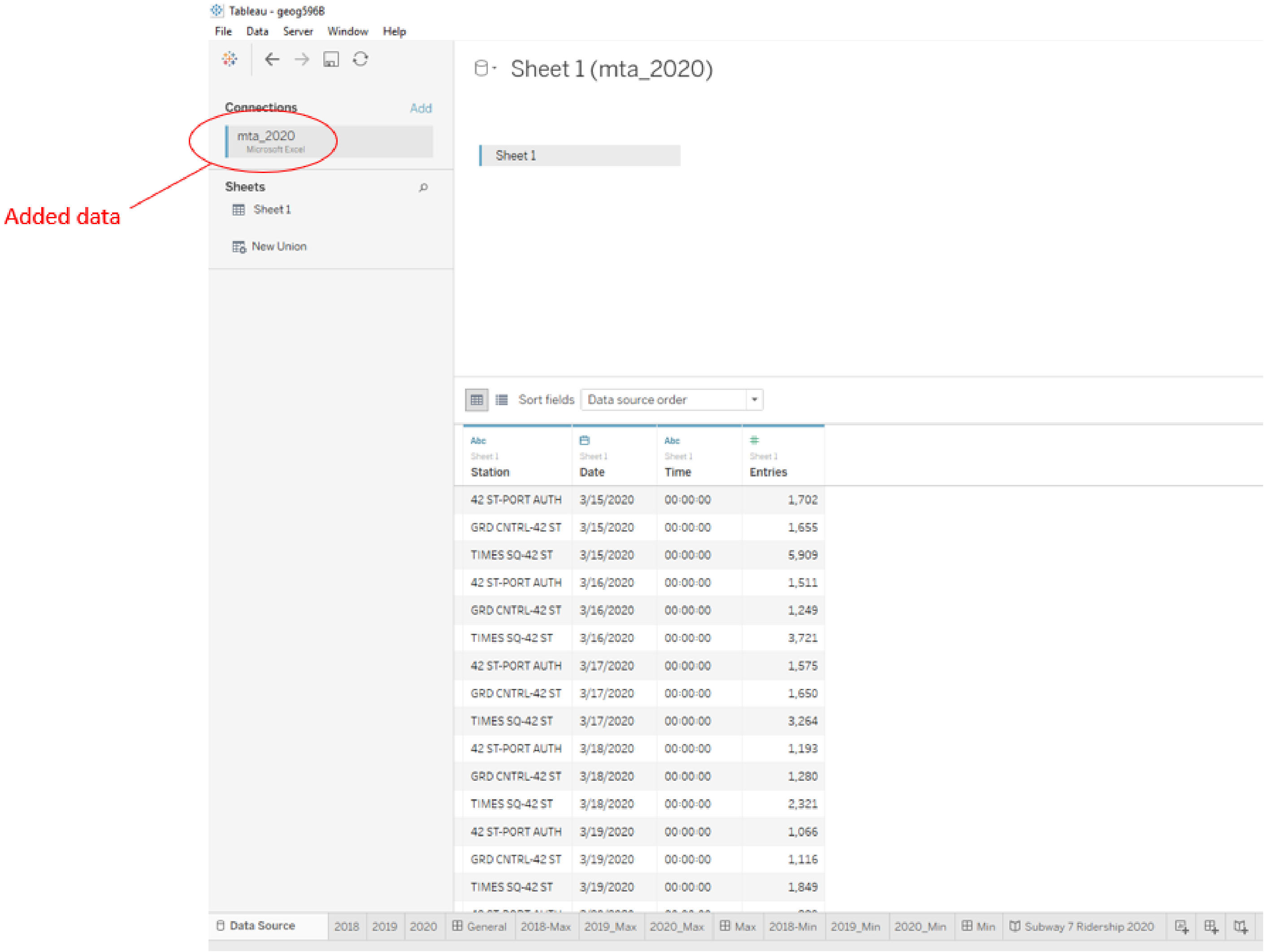


*Figure 7- Panel for connecting to data*



| Station | Date | Time | Entries |
|---|---|---|---|
| 42 ST-PORT AUTH | 3/15/2020 | 00:00:00 | 1,702 |
| GRD CNTRL-42 ST | 3/15/2020 | 00:00:00 | 1,655 |
| TIMES SQ-42 ST | 3/15/2020 | 00:00:00 | 5,909 |
| 42 ST-PORT AUTH | 3/16/2020 | 00:00:00 | 1,511 |
| GRD CNTRL-42 ST | 3/16/2020 | 00:00:00 | 1,249 |
| TIMES SQ-42 ST | 3/16/2020 | 00:00:00 | 3,721 |
| 42 ST-PORT AUTH | 3/17/2020 | 00:00:00 | 1,575 |
| GRD CNTRL-42 ST | 3/17/2020 | 00:00:00 | 1,650 |
| TIMES SQ-42 ST | 3/17/2020 | 00:00:00 | 3,264 |
| 42 ST-PORT AUTH | 3/18/2020 | 00:00:00 | 1,193 |
| GRD CNTRL-42 ST | 3/18/2020 | 00:00:00 | 1,280 |
| TIMES SQ-42 ST | 3/18/2020 | 00:00:00 | 2,321 |
| 42 ST-PORT AUTH | 3/19/2020 | 00:00:00 | 1,066 |
| GRD CNTRL-42 ST | 3/19/2020 | 00:00:00 | 1,116 |
| TIMES SQ-42 ST | 3/19/2020 | 00:00:00 | 1,849 |

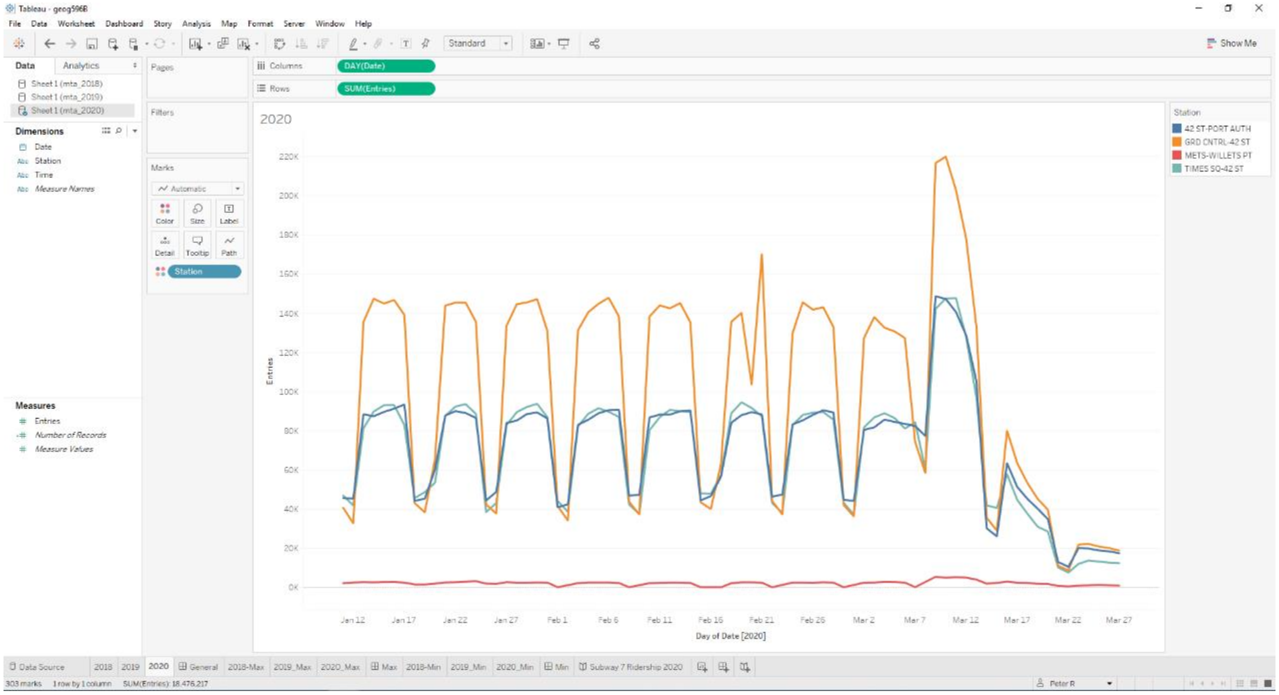
*Figure 8 - Tableau Desktop Data Source page*

*Example of a Tableau Desktop Worksheet*

Once data was pulled into Tableau Desktop, it got displayed in the left ‘Data’ pane. Each data field that shows up in the Data pane gets categorized by Tableau Desktop as either a Dimension (blue pill) or a Measure (green pill).

- Dimensions (blue pills) represent categorical data and provide a means to control the granularity of the data
- Measures (green pills) represent aggregates or continuous data that, for this project, provided continuous data that can normally be used for, say, time-based analysis

The layout for the time-based presentation below pulled in the ‘DATE’ field aggregation along the x-axis and displayed the ‘ENTRIES’ data field along the y-axis, where the ‘STATION’ dimension was configured to control the color of the data being presented for the different subway stations that were being analyzed.



Columns: DAY(Date)

Rows: SUM(Entries)

Color: Station

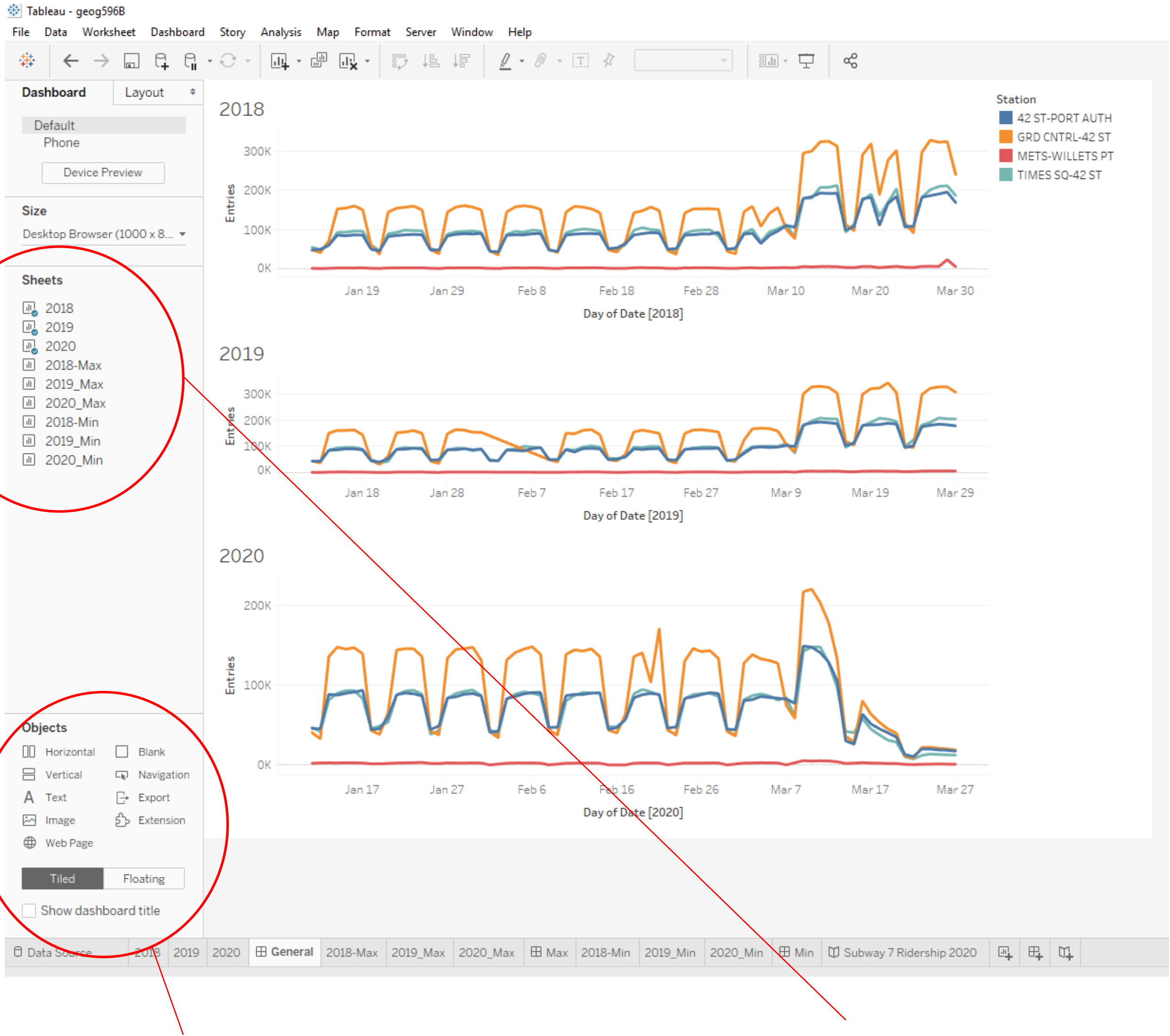
303 marks 1 row by 1 column SUM(Entries): 18,476,217

*Figure 9- New York City Subway Turnstile Counts for the months of January through March, 2020*

Note: The Tableau Desktop Worksheet environment provides the capabilities to filter the data, color the data, change the size of the data, and add tooltips. Also, it is very easy to change how data gets graphically displayed (i.e. switching from a line plot representation to, say, a bar chart)

### *Example of a Tableau Desktop Dashboard*

Multiple Tableau Desktop Worksheets can be created within a Tableau Desktop Workbook. A Tableau Desktop Dashboard, then, can take these Tableau Desktop Worksheets and present them together on a singular view. To aid in the layout of these spreadsheets, the 'Objects' pane in the lower left corner of the development environment allows for adding additional features that can enhance the viewability of the data visualization being presented (i.e. text items like titles or sub-titles, and also images.



Objects available for use to help mold the dashboard design around 'Sheets' that get pulled into the dashboard

Items that are listed in the 'Sheets' section of the Dashboard are the same 'Worksheets' that show up in the tabs section at the bottom of the Tableau Desktop Workbook

*Figure 10- New York City Subway Turnstile Counts for the months of January through March for 2018, 2019, and 2020*

### *Example of a Tableau Desktop Dashboard exported online to Tableau Public*

Tableau Public is a free online service where views and dashboards created in Tableau Desktop can be uploaded to the web. Registration was required to get started but once an account gets created, Tableau Desktop provides features within the application development environment that allow for effective upload of Tableau Desktop Worksheets and Tableau Desktop Dashboards.

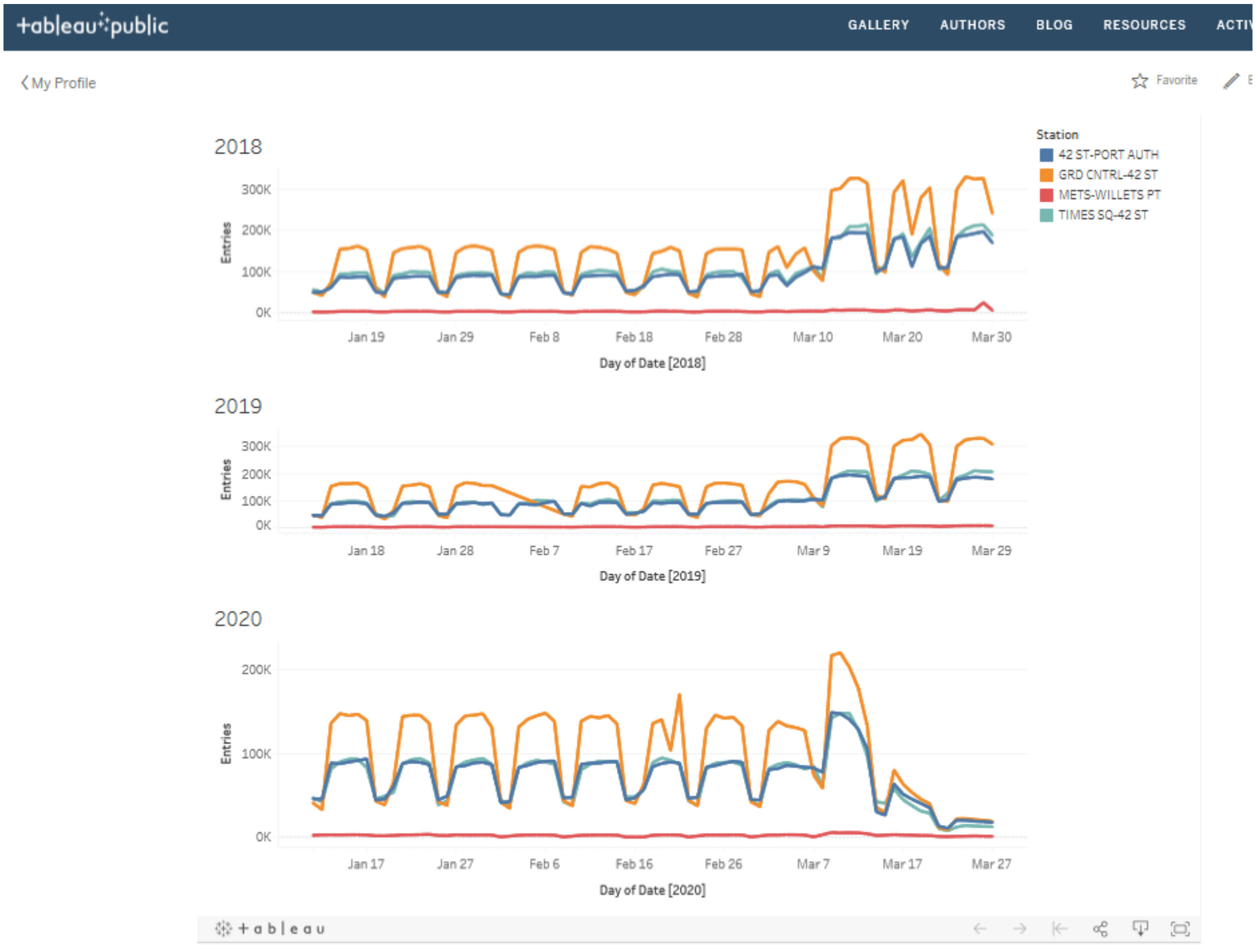


*Figure 11- Subway Turnstile Dashboard Visualization displayed online on Tableau Public*

Note: A Tableau Desktop Worksheet or Tableau Desktop Dashboard that is ready for upload has to be converted into a Tableau Data Extract (TDE) before the Tableau Public upload features within Tableau Desktop would perform the operation. This required switching the data connection for the Tableau Desktop Workbook from ‘Live’ to ‘Extract’ within the ‘Data Source’ window.

### *Data visualization developed in Tableau Online/Mobile*

There are additional offerings that Tableau puts out to support a professional business environment. They are called Tableau Server, Tableau Online, and Tableau Mobile. These options were explored during the project by signing up for a 14-day trial of Tableau Online and Tableau Mobile. A mobile presentation was configured for an Apple ipad using the subway turnstile data for this project.

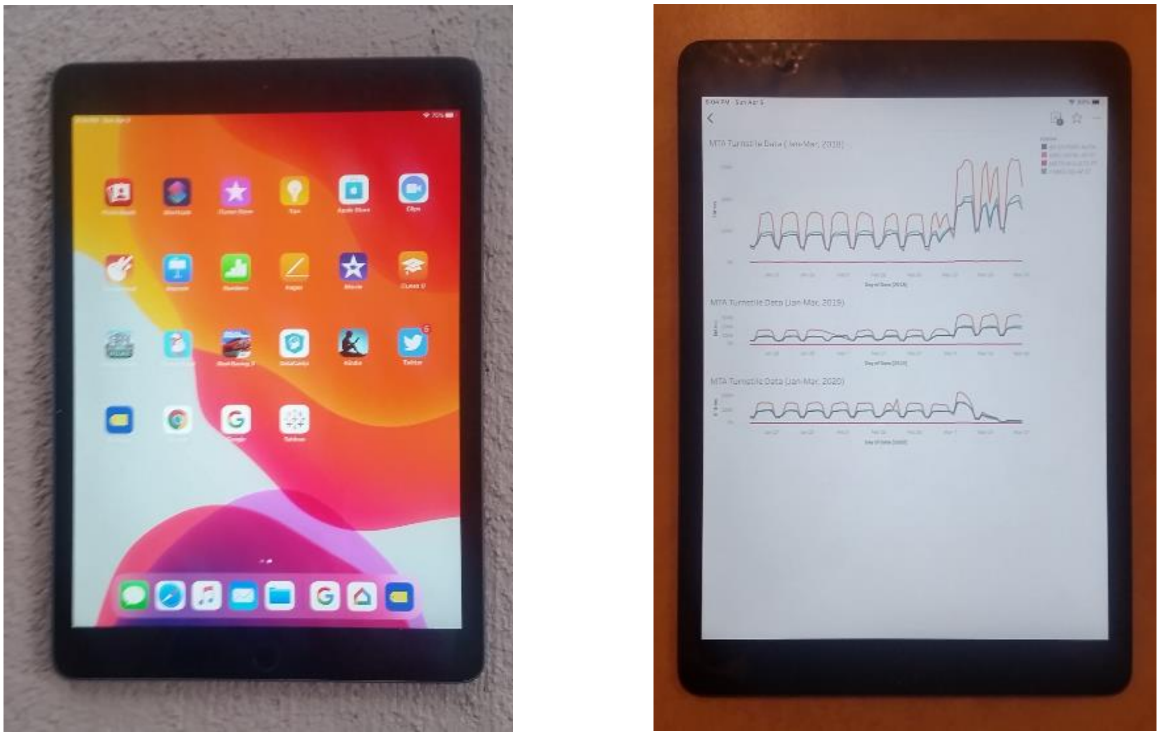

MTA Turnstile Data (Jan-Mar, 2018)
MTA Turnstile Data (Jan-Mar, 2019)
MTA Turnstile Data (Jan-Mar, 2020)

*Figure 12- NYC Subway Turnstile Dashboard Visualization displayed using Tableau Online/Mobile*

### *Dashboard Design Information Graphic (Version 1)*

There is a forum called Makeover Mondays that presents a data visualization every week and challenges participants to answer 3 questions: What did you like about the data visualization? What don't you like? How can you make it better?

Below is an early iteration of a dashboard design for subway turnstile data that incorporates data visualization fundamentals acquired over the past couple of months.

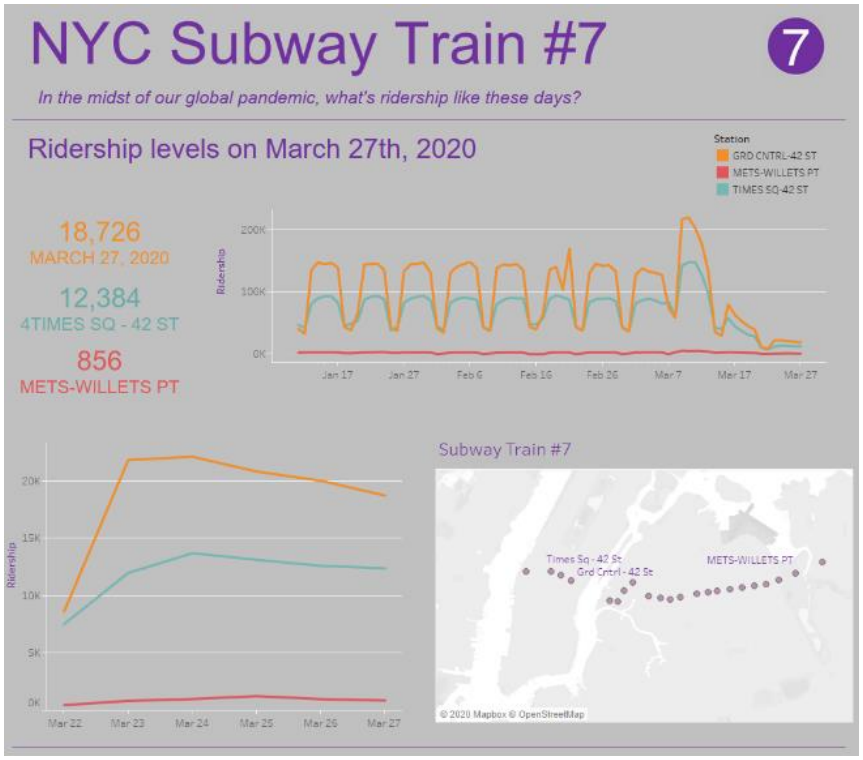


*Figure 13- Information Graphic using Tableau Desktop Application*

### What I liked…

- The simplicity of the line chart showcases the drop in ridership when social distancing was encouraged
- Adding a map to the visualization provides perspective to the data shown in the line charts

### What I didn't like…

- Overuse of colors in this data visualization (not incorporating pre-attentive attribute best practices)
  - The color purple represents the colors of the NYC Subway Trainline 7
  - the gray background was chosen to resemble the metal subway trains
  - three random basic colors
- Manual entry encourages error (orange 'March 27, 2020' text should have read 'GRD CNTRL – 42 ST'

### What would I do different…

See 'Dashboard Design Information Graphic (Version 2) section

### *Tableau Prep Builder*

The Tableau Prep Builder Software Application was used to try for a cleaning the data to a deeper level of granularity. It provided a more visually intuitive experience for cleaning the data. The application follows a process flow format. Cleaning operations ultimately output to a .csv file. This section provides a little overview on each step in the process that was taken.

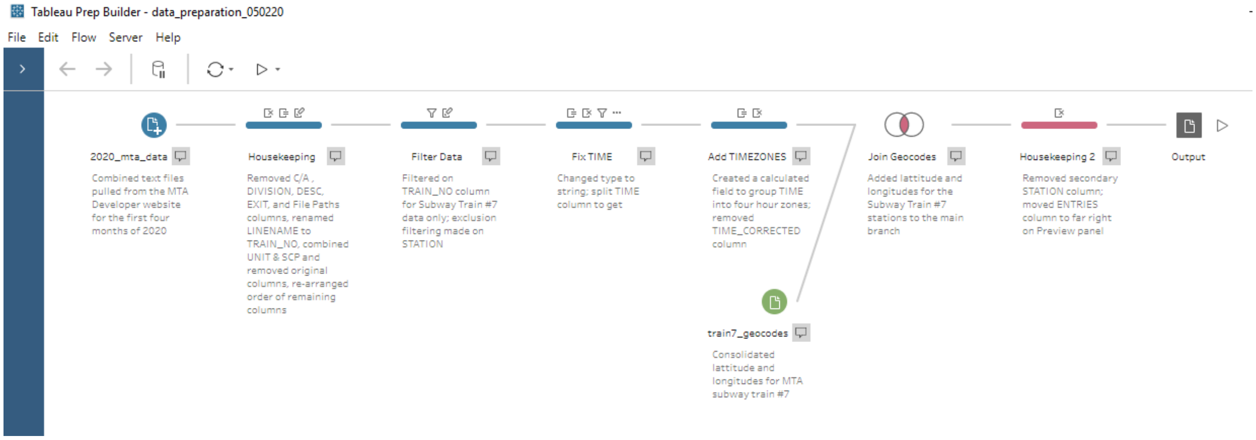


Tableau Prep Builder - data_preparation_050220
File Edit Flow Server Help

2020_mta_data – Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping – Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data – Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME – Changed type to string; split TIME column to get

Add TIMEZONES – Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

train7_geocodes – Consolidated lattitude and longitudes for MTA subway train #7

Join Geocodes – Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2 – Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output

*Figure 14- Flow Pane section of the Tableau Prep Builder Application*

Tableau Prep Step #1 – Files from the MTA Developer website were initially downloaded to a directory using the R programming language. This step then goes and connects to one of those files. The 'Multiple Files' tab then needed to be selected and the 'Applied' button was clicked. All files, consequently, then got pulled in for analysis (i.e. January, 2020 through April, 2020).

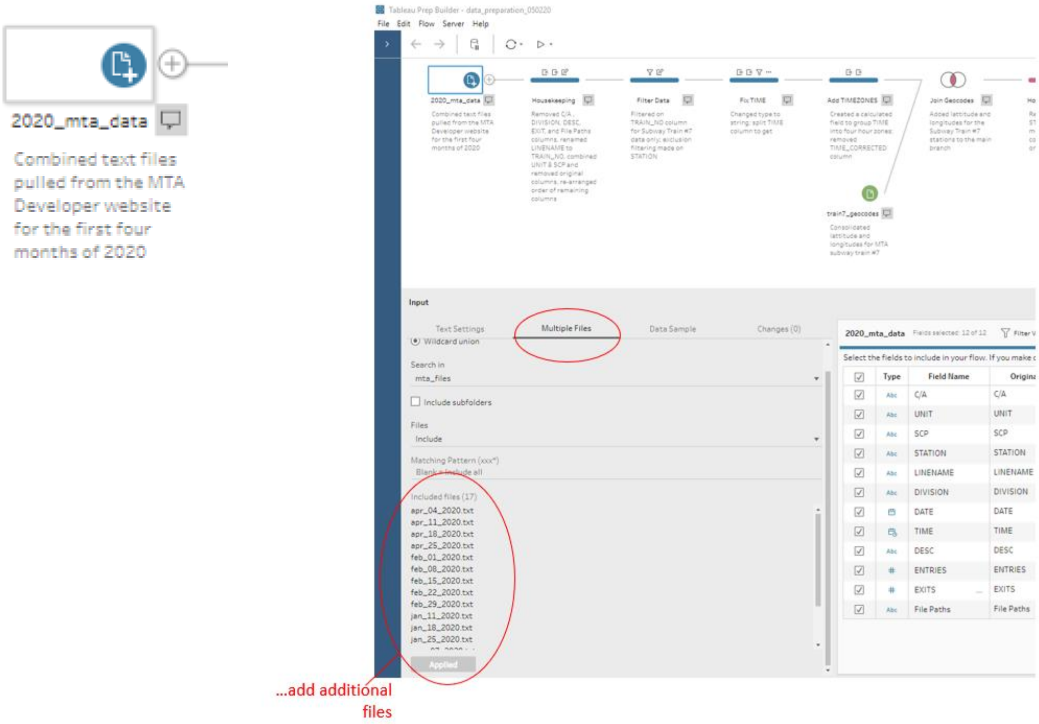

2020_mta_data – Combined text files pulled from the MTA Developer website for the first four months of 2020

Tableau Prep Builder - data_preparation_050220
File Edit Flow Server Help
Input
Text Settings
Multiple Files
Data Sample
Changes (0)

● Wildcard union

Search in: mta_files

□ Include subfolders

Files: Include

Matching Pattern (xxx*): Blank = Include all

Included files (17)
apr_04_2020.txt
apr_11_2020.txt
apr_18_2020.txt
apr_25_2020.txt
feb_01_2020.txt
feb_08_2020.txt
feb_15_2020.txt
feb_22_2020.txt
feb_29_2020.txt
jan_11_2020.txt
jan_18_2020.txt
jan_25_2020.txt

2020_mta_data Fields selected: 12 of 12

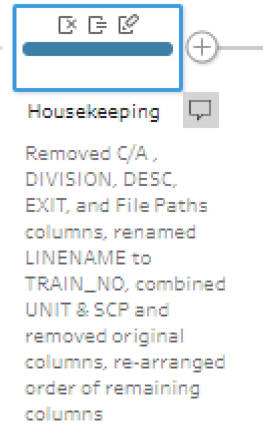
Select the fields to include in your flow. If you make c

| ☑ | Type | Field Name | Origina |
|---|---|---|---|
| ☑ | Abc | C/A | C/A |
| ☑ | Abc | UNIT | UNIT |
| ☑ | Abc | SCP | SCP |
| ☑ | Abc | STATION | STATION |
| ☑ | Abc | LINENAME | LINENAME |
| ☑ | Abc | DIVISION | DIVISION |
| ☑ | | DATE | DATE |
| ☑ | | TIME | TIME |
| ☑ | Abc | DESC | DESC |
| ☑ | # | ENTRIES | ENTRIES |
| ☑ | # | EXITS | EXITS |
| ☑ | Abc | File Paths | File Paths |

...add additional files

*Figure 15- Tableau Prep Multiple File Input phase*

Tableau Prep Step #2 – This step basically cleaned out some of the columns in the Tableau Prep Builder's Profile Pane that were not needed for the analysis. In addition, Prep has the capability of combining two columns together. For this project, a new column was created that stringed together the values in the UNIT column with the SCP column.



Housekeeping

Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

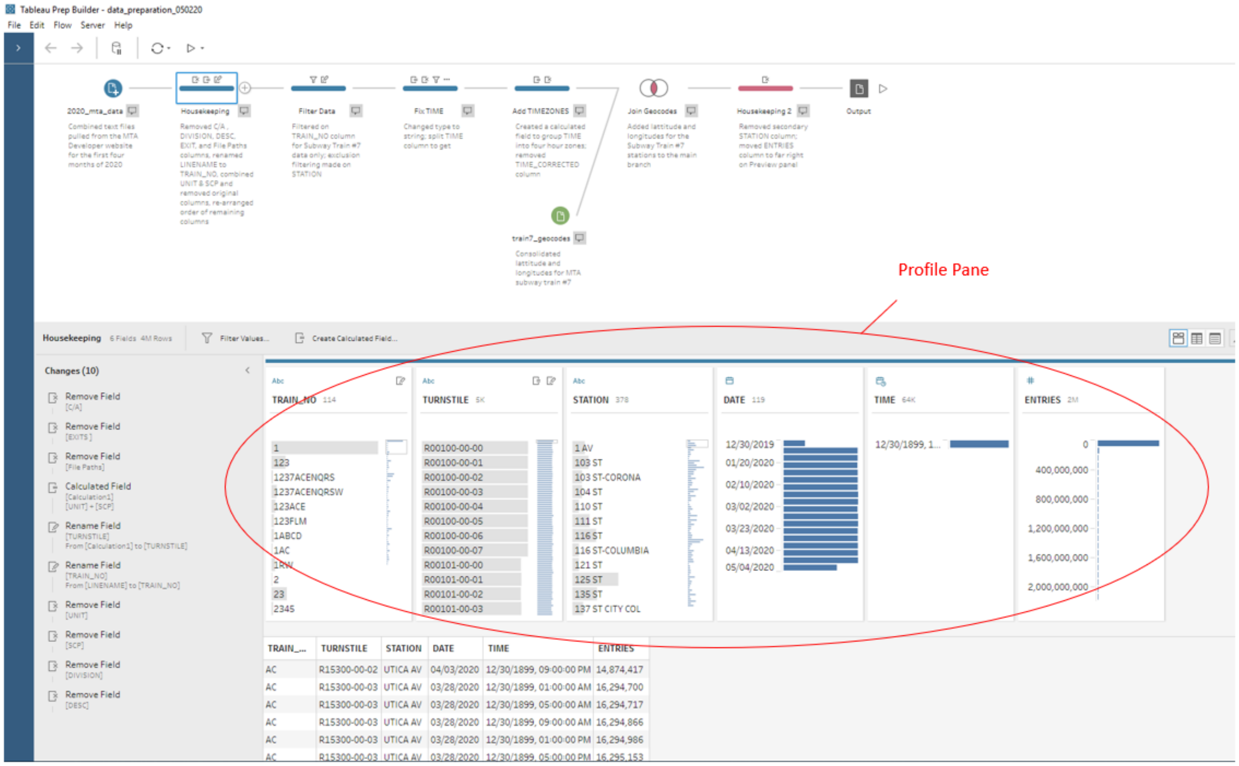


Tableau Prep Builder - data_preparation_050220

2020_mta_data: Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping: Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data: Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME: Changed type to string; split TIME column to get

Add TIMEZONES: Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

train7_geocodes: Consolidated lattitude and longitudes for MTA subway train #7

Join Geocodes: Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2: Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output

Profile Pane

Housekeeping 6 Fields 4M Rows

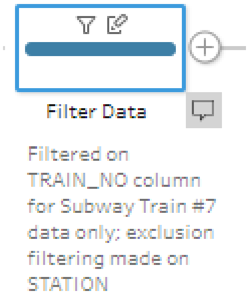
Changes (10)

- Remove Field [C/A]
- Remove Field [EXITS ]
- Remove Field [File Paths]
- Calculated Field [Calculation1] [UNIT] + [SCP]
- Rename Field [TURNSTILE] From [Calculation1] to [TURNSTILE]
- Rename Field [TRAIN_NO] From [LINENAME] to [TRAIN_NO]
- Remove Field [UNIT]
- Remove Field [SCP]
- Remove Field [DIVISION]
- Remove Field [DESC]

| TRAIN_... | TURNSTILE | STATION | DATE | TIME | ENTRIES |
|---|---|---|---|---|---|
| AC | R15300-00-02 | UTICA AV | 04/03/2020 | 12/30/1899, 09:00:00 PM | 14,874,417 |
| AC | R15300-00-03 | UTICA AV | 03/28/2020 | 12/30/1899, 01:00:00 AM | 16,294,700 |
| AC | R15300-00-03 | UTICA AV | 03/28/2020 | 12/30/1899, 05:00:00 AM | 16,294,717 |
| AC | R15300-00-03 | UTICA AV | 03/28/2020 | 12/30/1899, 09:00:00 AM | 16,294,866 |
| AC | R15300-00-03 | UTICA AV | 03/28/2020 | 12/30/1899, 01:00:00 PM | 16,294,986 |
| AC | R15300-00-03 | UTICA AV | 03/28/2020 | 12/30/1899, 05:00:00 PM | 16,295,153 |

*Figure 16- Tableau Prep Housekeeping phase*

Tableau Prep Step #3 – This step focused on filtering data for just Subway Trainline 7. There is a filter option located in the 'TRAIN_NO' panel within the Tableau Prep Builder Profile Pane.



Filter Data

Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

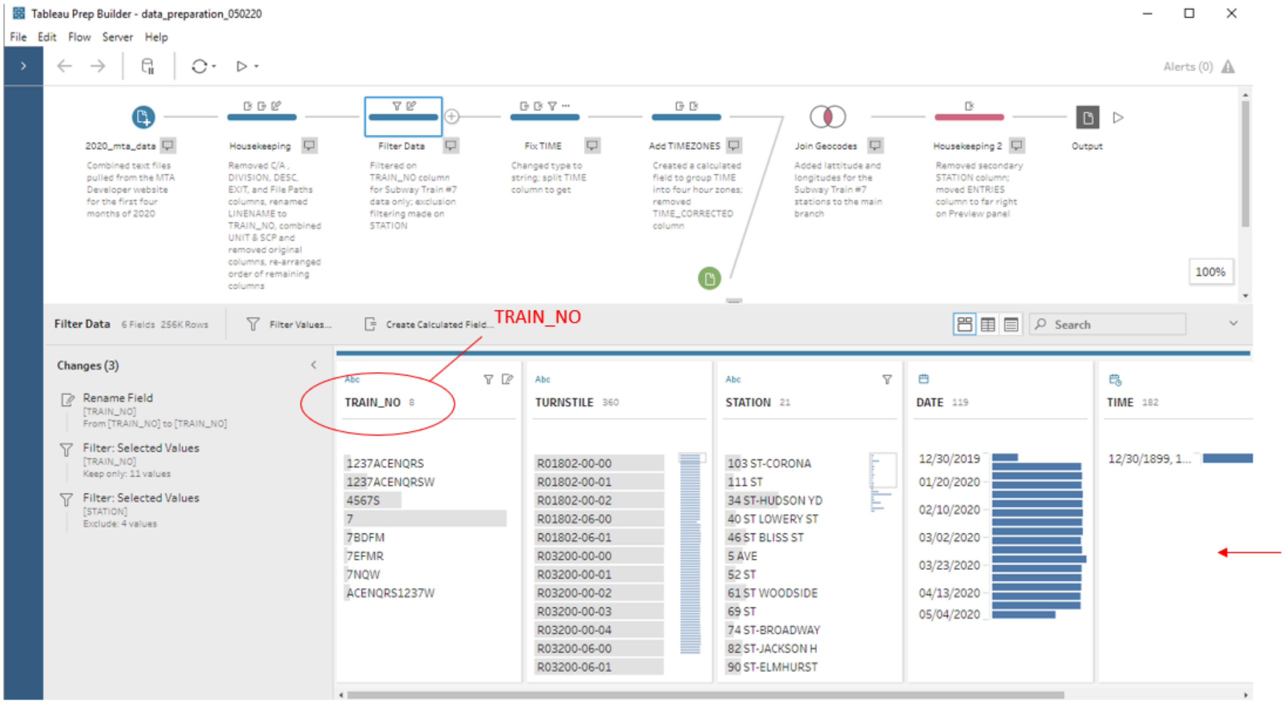


Tableau Prep Builder - data_preparation_050220

2020_mta_data: Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping: Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data: Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME: Changed type to string; split TIME column to get

Add TIMEZONES: Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

Join Geocodes: Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2: Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output

Filter Data 6 Fields 256K Rows

Changes (3)

Rename Field
[TRAIN_NO]
From [TRAIN_NO] to [TRAIN_NO]

Filter: Selected Values
[TRAIN_NO]
Keep only: 11 values

Filter: Selected Values
[STATION]
Exclude: 4 values

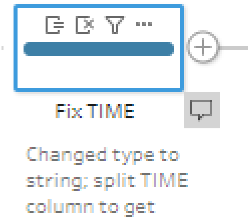
| TRAIN_NO 8 | TURNSTILE 360 | STATION 21 | DATE 119 | TIME 182 |
|---|---|---|---|---|
| 1237ACENQRS | R01802-00-00 | 103 ST-CORONA | 12/30/2019 | 12/30/1899, 1... |
| 1237ACENQRSW | R01802-00-01 | 111 ST | 01/20/2020 | |
| 4567S | R01802-00-02 | 34 ST-HUDSON YD | 02/10/2020 | |
| 7 | R01802-06-00 | 40 ST LOWERY ST | 03/02/2020 | |
| 7BDFM | R01802-06-01 | 46 ST BLISS ST | 03/23/2020 | |
| 7EFMR | R03200-00-00 | 5 AVE | 04/13/2020 | |
| 7NQW | R03200-00-01 | 52 ST | 05/04/2020 | |
| ACENQRS1237W | R03200-00-02 | 61 ST WOODSIDE | | |
| | R03200-00-03 | 69 ST | | |
| | R03200-00-04 | 74 ST-BROADWAY | | |
| | R03200-06-00 | 82 ST-JACKSON H | | |
| | R03200-06-01 | 90 ST-ELMHURST | | |

TRAIN_NO

Profile Pane

*Figure 17- Tableau Prep Filtering phase*

Tableau Prep Step 4 – This step focused getting started with configuring ‘time’ for this data set. It needed a date type to a string type conversion, some string manipulation, and then conversion to an integer type. The new column created got called ‘TIME_CORRECTED’.



Fix TIME

Changed type to string; split TIME column to get

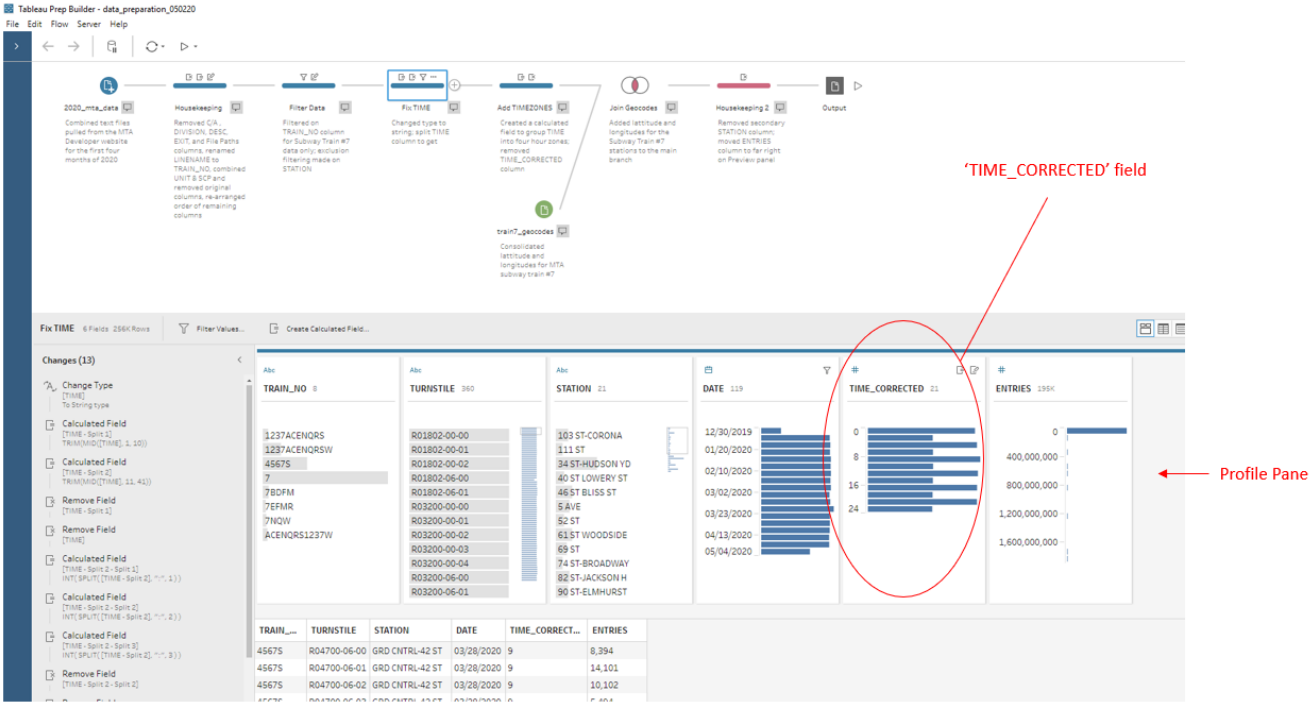


Tableau Prep Builder - data_preparation_050220

File Edit Flow Server Help

2020_mta_data – Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping – Removed C/A., DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data – Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME – Changed type to string; split TIME column to get

Add TIMEZONES – Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

train7_geocodes – Consolidated lattitude and longitudes for MTA subway train #7

Join Geocodes – Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2 – Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output

Fix TIME 6 Fields 256K Rows | Filter Values... | Create Calculated Field...

Changes (13)

- Change Type [TIME] To String type
- Calculated Field [TIME - Split 1] TRIM(MID([TIME], 1, 10))
- Calculated Field [TIME - Split 2] TRIM(MID([TIME], 11, 41))
- Remove Field [TIME - Split 1]
- Remove Field [TIME]
- Calculated Field [TIME - Split 2 - Split 1] INT( SPLIT( [TIME - Split 2], ":", 1 ) )
- Calculated Field [TIME - Split 2 - Split 2] INT( SPLIT( [TIME - Split 2], ":", 2 ) )
- Calculated Field [TIME - Split 2 - Split 3] INT( SPLIT( [TIME - Split 2], ":", 3 ) )
- Remove Field [TIME - Split 2 - Split 2]

TRAIN_NO 8: 1237ACENQRS, 1237ACENQRSW, 4567S, 7, 7BDFM, 7EFMR, 7NQW, ACENQRS1237W

TURNSTILE 360: R01802-00-00, R01802-00-01, R01802-00-02, R01802-06-00, R01802-06-01, R03200-00-00, R03200-00-01, R03200-00-02, R03200-00-03, R03200-00-04, R03200-06-00, R03200-06-01

STATION 21: 103 ST-CORONA, 111 ST, 34 ST-HUDSON YD, 40 ST LOWERY ST, 46 ST BLISS ST, 5 AVE, 52 ST, 61 ST WOODSIDE, 69 ST, 74 ST-BROADWAY, 82 ST-JACKSON H, 90 ST-ELMHURST

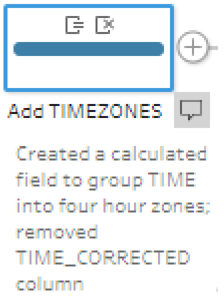
| TRAIN_... | TURNSTILE | STATION | DATE | TIME_CORRECT... | ENTRIES |
|---|---|---|---|---|---|
| 4567S | R04700-06-00 | GRD CNTRL-42 ST | 03/28/2020 | 9 | 8,394 |
| 4567S | R04700-06-01 | GRD CNTRL-42 ST | 03/28/2020 | 9 | 14,101 |
| 4567S | R04700-06-02 | GRD CNTRL-42 ST | 03/28/2020 | 9 | 10,102 |

‘TIME_CORRECTED’ field

Profile Pane

*Figure 18- Tableau Prep Fixing Time phase*

Tableau Prep Step 5 – This step focuses on creating a representation of the timezones in string format. Notice that the 'TIME_CORRECTED' field was removed as it was no longer needed.



Add TIMEZONES

Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

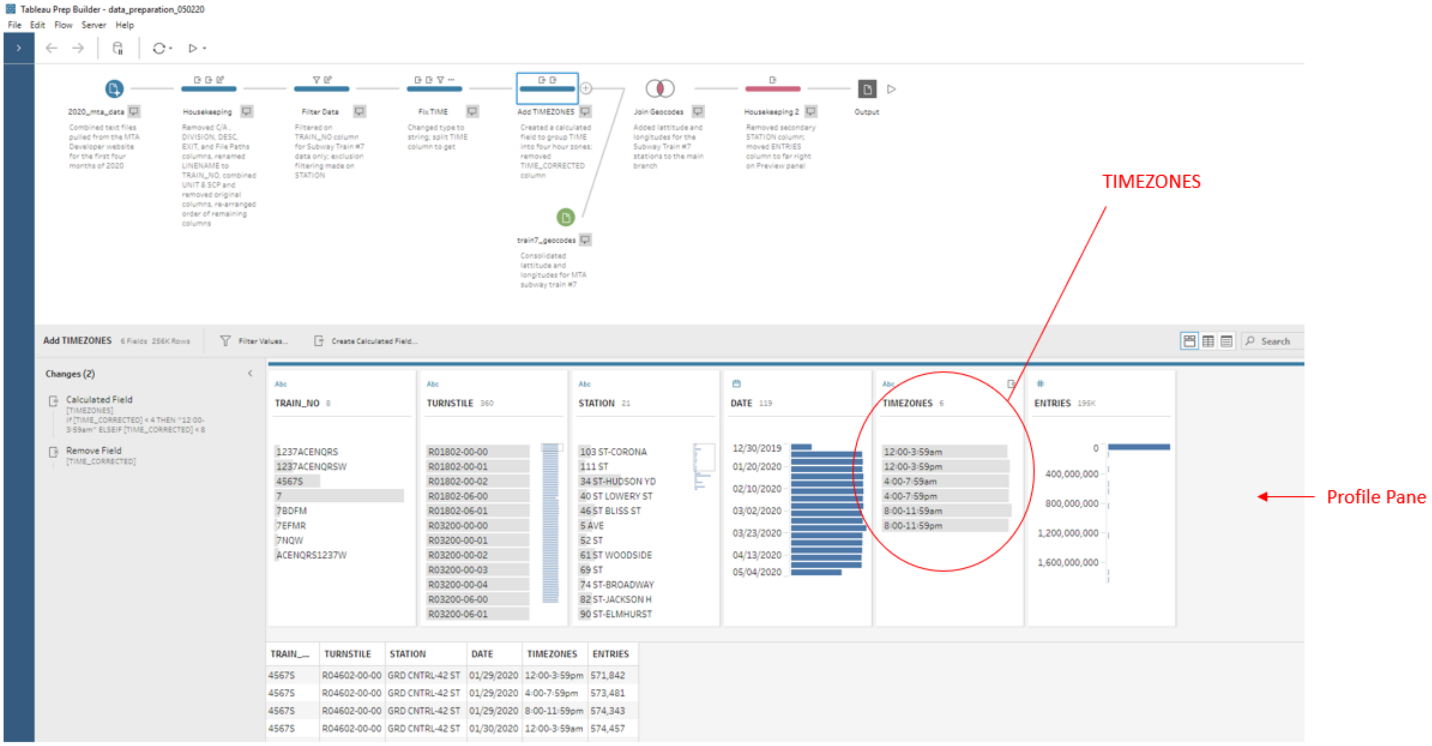


Tableau Prep Builder - data_preparation_050220

2020_mta_data – Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping – Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data – Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME – Changed type to string; split TIME column to get

Add TIMEZONES – Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

train7_geocodes – Consolidated lattitude and longitudes for MTA subway train #7

Join Geocodes – Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2 – Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output

Add TIMEZONES 6 Fields 256K Rows

Changes (2)

Calculated Field
[TIMEZONES]
If [TIME_CORRECTED] < 4 THEN "12:00-3:59am" ELSEIF [TIME_CORRECTED] < 8

Remove Field
[TIME_CORRECTED]

TRAIN_NO 8: 1237ACENQRS, 1237ACENQRSW, 4567S, 7, 7BDFM, 7EFMR, 7NQW, ACENQRS1237W

TURNSTILE 360: R01802-00-00, R01802-00-01, R01802-00-02, R01802-06-00, R01802-06-01, R03200-00-00, R03200-00-01, R03200-00-02, R03200-00-03, R03200-00-04, R03200-06-00, R03200-06-01

STATION 21: 103 ST-CORONA, 111 ST, 34 ST-HUDSON YD, 40 ST LOWERY ST, 46 ST BLISS ST, 5 AVE, 52 ST, 61 ST WOODSIDE, 69 ST, 74 ST-BROADWAY, 82 ST-JACKSON H, 90 ST-ELMHURST

DATE 119: 12/30/2019, 01/20/2020, 02/10/2020, 03/02/2020, 03/23/2020, 04/13/2020, 05/04/2020

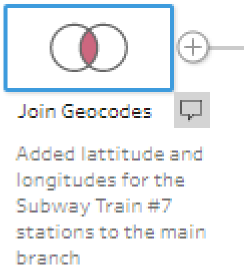
TIMEZONES 6: 12:00-3:59am, 12:00-3:59pm, 4:00-7:59am, 4:00-7:59pm, 8:00-11:59am, 8:00-11:59pm

ENTRIES 195K: 0, 400,000,000, 800,000,000, 1,200,000,000, 1,600,000,000

| TRAIN_... | TURNSTILE | STATION | DATE | TIMEZONES | ENTRIES |
|---|---|---|---|---|---|
| 4567S | R04602-00-00 | GRD CNTRL-42 ST | 01/29/2020 | 12:00-3:59pm | 571,842 |
| 4567S | R04602-00-00 | GRD CNTRL-42 ST | 01/29/2020 | 4:00-7:59pm | 573,481 |
| 4567S | R04602-00-00 | GRD CNTRL-42 ST | 01/29/2020 | 8:00-11:59pm | 574,343 |
| 4567S | R04602-00-00 | GRD CNTRL-42 ST | 01/30/2020 | 12:00-3:59am | 574,457 |

*Figure 19- Tableau Prep Time Zone phase*

Tableau Prep Step 6 – This step is about joining 'LATITUDE' AND 'LONGITUDE' values. These values are located in a separate file that gets pulled into the Tableau Prep main branch of the 'Prep' flow. Join activity can be viewed in the Profile Pane (highlighted by a Pink bar)



Join Geocodes

Added lattitude and longitudes for the Subway Train #7 stations to the main branch

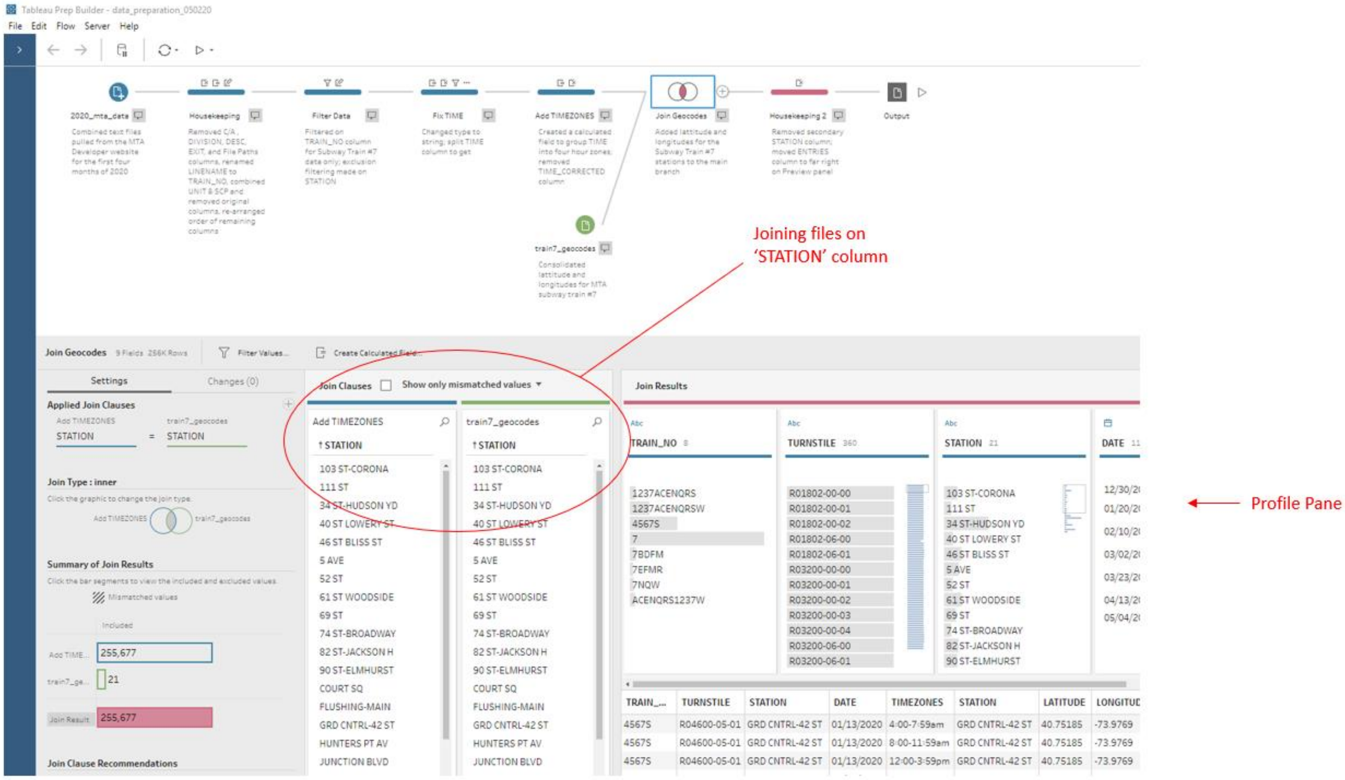


Tableau Prep Builder - data_preparation_050220
File Edit Flow Server Help

2020_mta_data: Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping: Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data: Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME: Changed type to string; split TIME column to get

Add TIMEZONES: Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

train7_geocodes: Consolidated lattitude and longitudes for MTA subway train #7

Join Geocodes: Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2: Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output

Joining files on 'STATION' column

Join Geocodes 9 Fields 256K Rows

Applied Join Clauses

Add TIMEZONES STATION = train7_geocodes STATION

Join Type : inner

Click the graphic to change the join type.

Summary of Join Results

Click the bar segments to view the included and excluded values.

▨ Mismatched values

| | Included |
|---|---|
| Add TIME... | 255,677 |
| train7_ge... | 21 |
| Join Result | 255,677 |

Join Clause Recommendations

Join Clauses □ Show only mismatched values ▾

| Add TIMEZONES ↑STATION | train7_geocodes ↑STATION |
|---|---|
| 103 ST-CORONA | 103 ST-CORONA |
| 111 ST | 111 ST |
| 34 ST-HUDSON YD | 34 ST-HUDSON YD |
| 40 ST LOWERY ST | 40 ST LOWERY ST |
| 46 ST BLISS ST | 46 ST BLISS ST |
| 5 AVE | 5 AVE |
| 52 ST | 52 ST |
| 61 ST WOODSIDE | 61 ST WOODSIDE |
| 69 ST | 69 ST |
| 74 ST-BROADWAY | 74 ST-BROADWAY |
| 82 ST-JACKSON H | 82 ST-JACKSON H |
| 90 ST-ELMHURST | 90 ST-ELMHURST |
| COURT SQ | COURT SQ |
| FLUSHING-MAIN | FLUSHING-MAIN |
| GRD CNTRL-42 ST | GRD CNTRL-42 ST |
| HUNTERS PT AV | HUNTERS PT AV |
| JUNCTION BLVD | JUNCTION BLVD |

Join Results

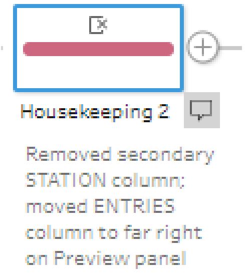
| Abc TRAIN_NO 8 | Abc TURNSTILE 360 | Abc STATION 21 | DATE 11 |
|---|---|---|---|
| 1237ACENQRS | R01802-00-00 | 103 ST-CORONA | 12/30/2( |
| 1237ACENQRSW | R01802-00-01 | 111 ST | 01/20/2( |
| 4567S | R01802-00-02 | 34 ST-HUDSON YD | 02/10/2( |
| 7 | R01802-06-00 | 40 ST LOWERY ST | 03/02/2( |
| 7BDFM | R01802-06-01 | 46 ST BLISS ST | 03/23/2( |
| 7EFMR | R03200-00-00 | 5 AVE | 04/13/2( |
| 7NQW | R03200-00-01 | 52 ST | 05/04/2( |
| ACENQRS1237W | R03200-00-02 | 61 ST WOODSIDE | |
| | R03200-00-03 | 69 ST | |
| | R03200-00-04 | 74 ST-BROADWAY | |
| | R03200-06-00 | 82 ST-JACKSON H | |
| | R03200-06-01 | 90 ST-ELMHURST | |

| TRAIN_... | TURNSTILE | STATION | DATE | TIMEZONES | STATION | LATITUDE | LONGITUD |
|---|---|---|---|---|---|---|---|
| 4567S | R04600-05-01 | GRD CNTRL-42 ST | 01/13/2020 | 4:00-7:59am | GRD CNTRL-42 ST | 40.75185 | -73.9769 |
| 4567S | R04600-05-01 | GRD CNTRL-42 ST | 01/13/2020 | 8:00-11:59am | GRD CNTRL-42 ST | 40.75185 | -73.9769 |
| 4567S | R04600-05-01 | GRD CNTRL-42 ST | 01/13/2020 | 12:00-3:59pm | GRD CNTRL-42 ST | 40.75185 | -73.9769 |

Profile Pane

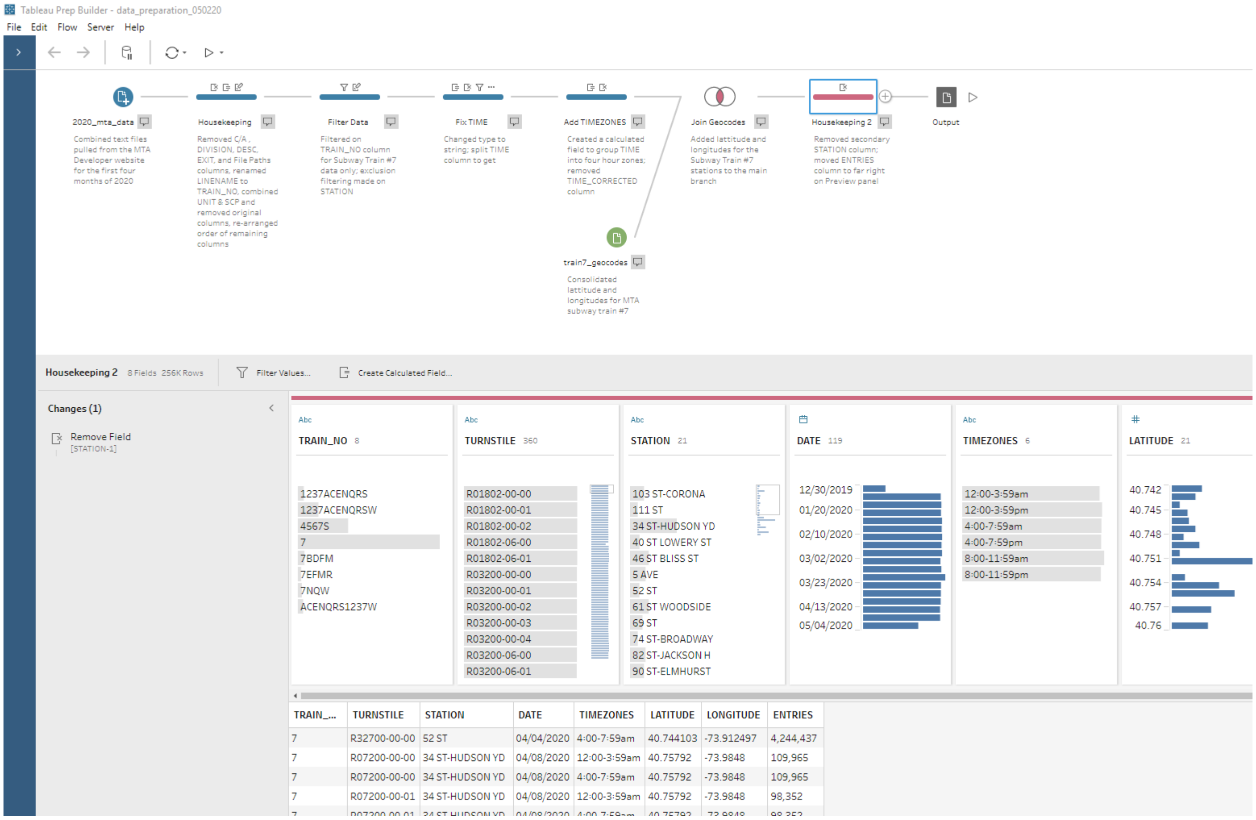
*Figure 20- Tableau Prep Joining Operations*

Tableau Prep Step 7 – Minor clean-up step. Secondary 'STATION-1' column was removed. It was the result of the Join operation in the prior step. The recommendation is to create logical steps in the Tableau Prep Builder process flow that allow for clarity when flow is reviewed at a later time.



Housekeeping 2

Removed secondary STATION column; moved ENTRIES column to far right on Preview panel



2020_mta_data: Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping: Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data: Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME: Changed type to string; split TIME column to get

Add TIMEZONES: Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

train7_geocodes: Consolidated lattitude and longitudes for MTA subway train #7

Join Geocodes: Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2: Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output

Housekeeping 2 8 Fields 256K Rows

Changes (1)

Remove Field [STATION-1]

Abc TRAIN_NO 8: 1237ACENQRS, 1237ACENQRSW, 4567S, 7, 7BDFM, 7EFMR, 7NQW, ACENQRS1237W

Abc TURNSTILE 360: R01802-00-00, R01802-00-01, R01802-00-02, R01802-06-00, R01802-06-01, R03200-00-00, R03200-00-01, R03200-00-02, R03200-00-03, R03200-00-04, R03200-06-00, R03200-06-01

Abc STATION 21: 103 ST-CORONA, 111 ST, 34 ST-HUDSON YD, 40 ST LOWERY ST, 46 ST BLISS ST, 5 AVE, 52 ST, 61 ST WOODSIDE, 69 ST, 74 ST-BROADWAY, 82 ST-JACKSON H, 90 ST-ELMHURST

DATE 119: 12/30/2019, 01/20/2020, 02/10/2020, 03/02/2020, 03/23/2020, 04/13/2020, 05/04/2020

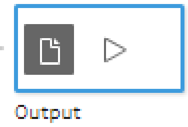
Abc TIMEZONES 6: 12:00-3:59am, 12:00-3:59pm, 4:00-7:59am, 4:00-7:59pm, 8:00-11:59am, 8:00-11:59pm

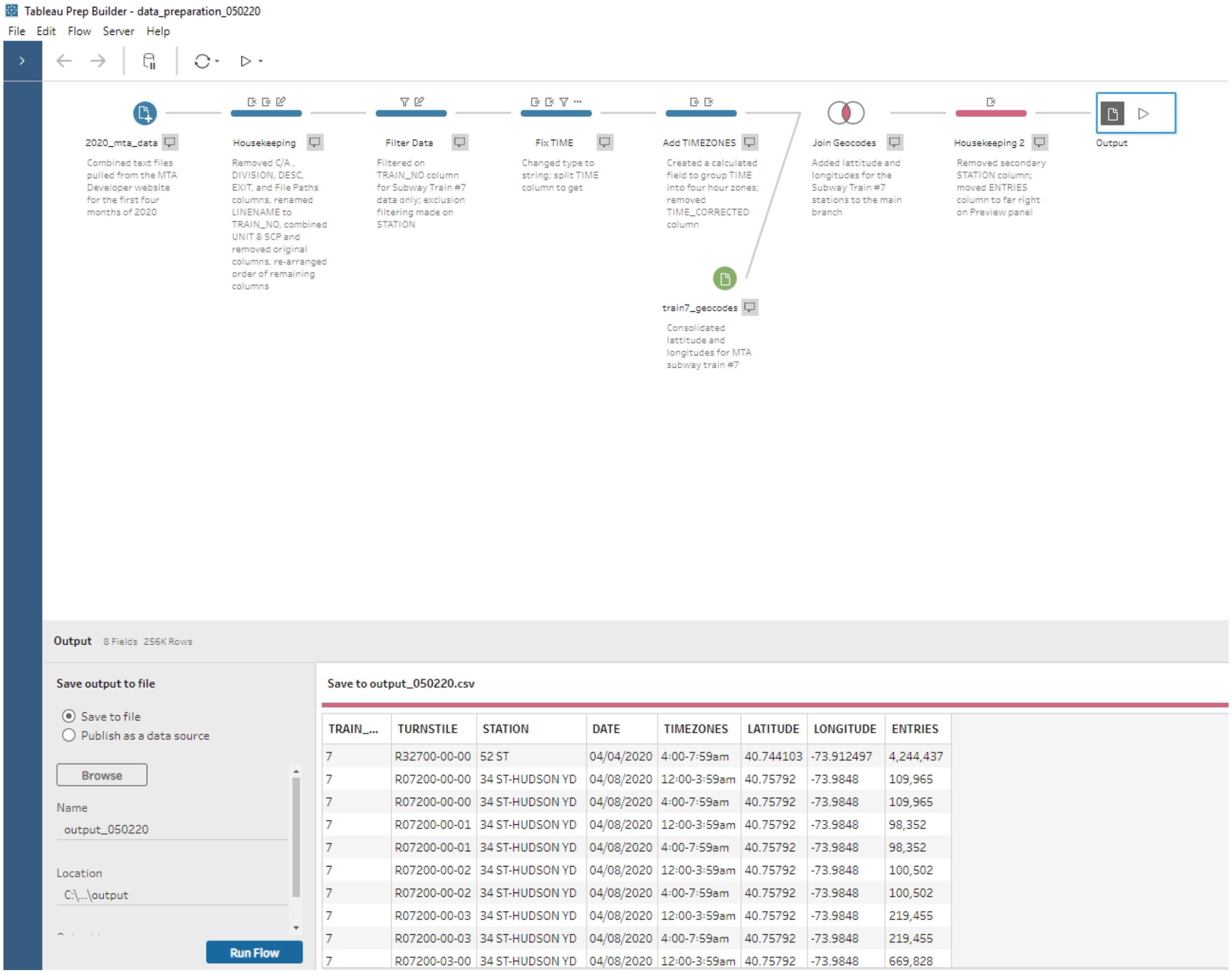
# LATITUDE 21: 40.742, 40.745, 40.748, 40.751, 40.754, 40.757, 40.76

| TRAIN_... | TURNSTILE | STATION | DATE | TIMEZONES | LATITUDE | LONGITUDE | ENTRIES |
|---|---|---|---|---|---|---|---|
| 7 | R32700-00-00 | 52 ST | 04/04/2020 | 4:00-7:59am | 40.744103 | -73.912497 | 4,244,437 |
| 7 | R07200-00-00 | 34 ST-HUDSON YD | 04/08/2020 | 12:00-3:59am | 40.75792 | -73.9848 | 109,965 |
| 7 | R07200-00-00 | 34 ST-HUDSON YD | 04/08/2020 | 4:00-7:59am | 40.75792 | -73.9848 | 109,965 |
| 7 | R07200-00-01 | 34 ST-HUDSON YD | 04/08/2020 | 12:00-3:59am | 40.75792 | -73.9848 | 98,352 |
| 7 | R07200-00-01 | 34 ST-HUDSON YD | 04/08/2020 | 4:00-7:59am | 40.75792 | -73.9848 | 98,352 |

*Figure 21- Tableau Prep Housecleaning phase (Part 2)*

Tableau Prep Step 8 – This step in the flow generates a .csv file. Next stop: Microsoft Excel for differential aggregation work





2020_mta_data: Combined text files pulled from the MTA Developer website for the first four months of 2020

Housekeeping: Removed C/A , DIVISION, DESC, EXIT, and File Paths columns, renamed LINENAME to TRAIN_NO, combined UNIT & SCP and removed original columns, re-arranged order of remaining columns

Filter Data: Filtered on TRAIN_NO column for Subway Train #7 data only; exclusion filtering made on STATION

Fix TIME: Changed type to string; split TIME column to get

Add TIMEZONES: Created a calculated field to group TIME into four hour zones; removed TIME_CORRECTED column

train7_geocodes: Consolidated lattitude and longitudes for MTA subway train #7

Join Geocodes: Added lattitude and longitudes for the Subway Train #7 stations to the main branch

Housekeeping 2: Removed secondary STATION column; moved ENTRIES column to far right on Preview panel

Output 8 Fields 256K Rows

Save output to file

● Save to file
○ Publish as a data source

Name: output_050220

Location: C:\...\output

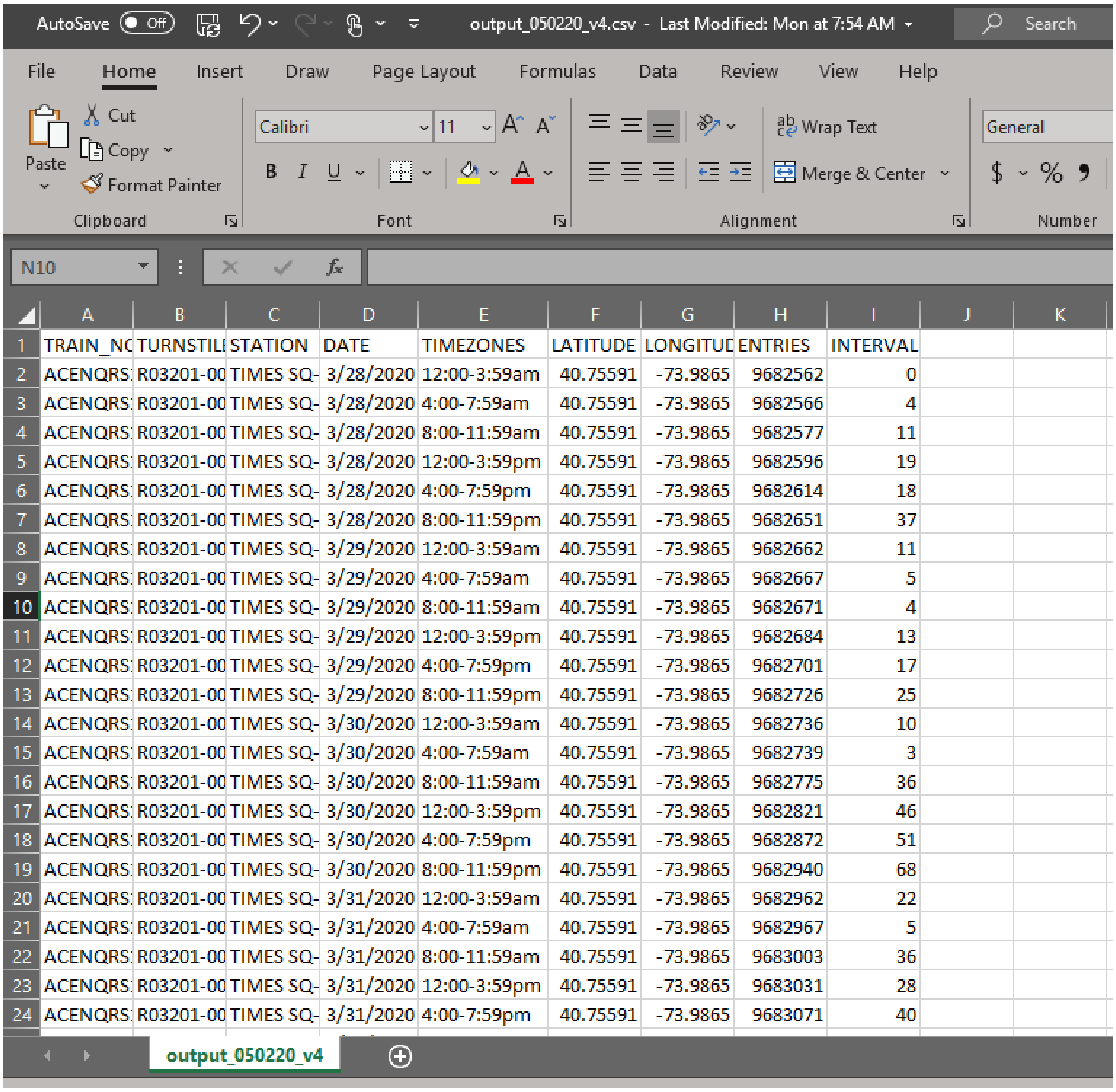
Save to output_050220.csv

| TRAIN_... | TURNSTILE | STATION | DATE | TIMEZONES | LATITUDE | LONGITUDE | ENTRIES |
|---|---|---|---|---|---|---|---|
| 7 | R32700-00-00 | 52 ST | 04/04/2020 | 4:00-7:59am | 40.744103 | -73.912497 | 4,244,437 |
| 7 | R07200-00-00 | 34 ST-HUDSON YD | 04/08/2020 | 12:00-3:59am | 40.75792 | -73.9848 | 109,965 |
| 7 | R07200-00-00 | 34 ST-HUDSON YD | 04/08/2020 | 4:00-7:59am | 40.75792 | -73.9848 | 109,965 |
| 7 | R07200-00-01 | 34 ST-HUDSON YD | 04/08/2020 | 12:00-3:59am | 40.75792 | -73.9848 | 98,352 |
| 7 | R07200-00-01 | 34 ST-HUDSON YD | 04/08/2020 | 4:00-7:59am | 40.75792 | -73.9848 | 98,352 |
| 7 | R07200-00-02 | 34 ST-HUDSON YD | 04/08/2020 | 12:00-3:59am | 40.75792 | -73.9848 | 100,502 |
| 7 | R07200-00-02 | 34 ST-HUDSON YD | 04/08/2020 | 4:00-7:59am | 40.75792 | -73.9848 | 100,502 |
| 7 | R07200-00-03 | 34 ST-HUDSON YD | 04/08/2020 | 12:00-3:59am | 40.75792 | -73.9848 | 219,455 |
| 7 | R07200-00-03 | 34 ST-HUDSON YD | 04/08/2020 | 4:00-7:59am | 40.75792 | -73.9848 | 219,455 |
| 7 | R07200-03-00 | 34 ST-HUDSON YD | 04/08/2020 | 12:00-3:59am | 40.75792 | -73.9848 | 669,828 |

*Figure 22- Tableau Prep Output phase*

### *Microsoft Excel*

Microsoft Excel was used as the application development environment for performing the differential aggregation needed on the 'ENTRIES' values from the Tableau Prep Builder output file. Microsoft Excel, contrary to popular knowledge, is a pretty powerful data preparation and visualization tool and provided the project with a quick solution using the 'ENTRIES' column to generate the 'INTERVAL' column values that represent commuter counts per turnstile unit for given time period.

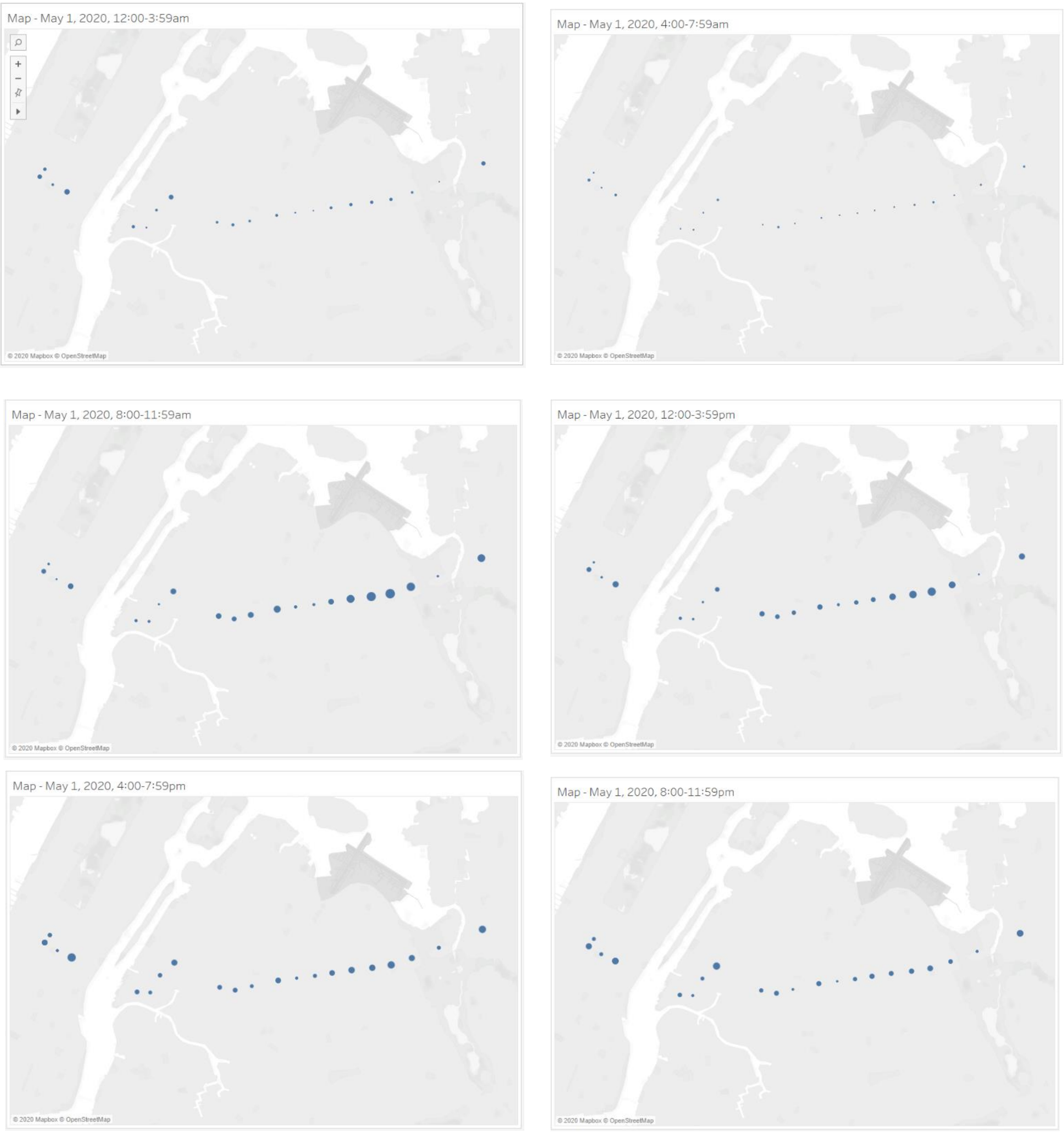


| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | TRAIN_NC | TURNSTILI | STATION | DATE | TIMEZONES | LATITUDE | LONGITUD | ENTRIES | INTERVAL |
| 2 | ACENQRS | R03201-00 | TIMES SQ- | 3/28/2020 | 12:00-3:59am | 40.75591 | -73.9865 | 9682562 | 0 |
| 3 | ACENQRS | R03201-00 | TIMES SQ- | 3/28/2020 | 4:00-7:59am | 40.75591 | -73.9865 | 9682566 | 4 |
| 4 | ACENQRS | R03201-00 | TIMES SQ- | 3/28/2020 | 8:00-11:59am | 40.75591 | -73.9865 | 9682577 | 11 |
| 5 | ACENQRS | R03201-00 | TIMES SQ- | 3/28/2020 | 12:00-3:59pm | 40.75591 | -73.9865 | 9682596 | 19 |
| 6 | ACENQRS | R03201-00 | TIMES SQ- | 3/28/2020 | 4:00-7:59pm | 40.75591 | -73.9865 | 9682614 | 18 |
| 7 | ACENQRS | R03201-00 | TIMES SQ- | 3/28/2020 | 8:00-11:59pm | 40.75591 | -73.9865 | 9682651 | 37 |
| 8 | ACENQRS | R03201-00 | TIMES SQ- | 3/29/2020 | 12:00-3:59am | 40.75591 | -73.9865 | 9682662 | 11 |
| 9 | ACENQRS | R03201-00 | TIMES SQ- | 3/29/2020 | 4:00-7:59am | 40.75591 | -73.9865 | 9682667 | 5 |
| 10 | ACENQRS | R03201-00 | TIMES SQ- | 3/29/2020 | 8:00-11:59am | 40.75591 | -73.9865 | 9682671 | 4 |
| 11 | ACENQRS | R03201-00 | TIMES SQ- | 3/29/2020 | 12:00-3:59pm | 40.75591 | -73.9865 | 9682684 | 13 |
| 12 | ACENQRS | R03201-00 | TIMES SQ- | 3/29/2020 | 4:00-7:59pm | 40.75591 | -73.9865 | 9682701 | 17 |
| 13 | ACENQRS | R03201-00 | TIMES SQ- | 3/29/2020 | 8:00-11:59pm | 40.75591 | -73.9865 | 9682726 | 25 |
| 14 | ACENQRS | R03201-00 | TIMES SQ- | 3/30/2020 | 12:00-3:59am | 40.75591 | -73.9865 | 9682736 | 10 |
| 15 | ACENQRS | R03201-00 | TIMES SQ- | 3/30/2020 | 4:00-7:59am | 40.75591 | -73.9865 | 9682739 | 3 |
| 16 | ACENQRS | R03201-00 | TIMES SQ- | 3/30/2020 | 8:00-11:59am | 40.75591 | -73.9865 | 9682775 | 36 |
| 17 | ACENQRS | R03201-00 | TIMES SQ- | 3/30/2020 | 12:00-3:59pm | 40.75591 | -73.9865 | 9682821 | 46 |
| 18 | ACENQRS | R03201-00 | TIMES SQ- | 3/30/2020 | 4:00-7:59pm | 40.75591 | -73.9865 | 9682872 | 51 |
| 19 | ACENQRS | R03201-00 | TIMES SQ- | 3/30/2020 | 8:00-11:59pm | 40.75591 | -73.9865 | 9682940 | 68 |
| 20 | ACENQRS | R03201-00 | TIMES SQ- | 3/31/2020 | 12:00-3:59am | 40.75591 | -73.9865 | 9682962 | 22 |
| 21 | ACENQRS | R03201-00 | TIMES SQ- | 3/31/2020 | 4:00-7:59am | 40.75591 | -73.9865 | 9682967 | 5 |
| 22 | ACENQRS | R03201-00 | TIMES SQ- | 3/31/2020 | 8:00-11:59am | 40.75591 | -73.9865 | 9683003 | 36 |
| 23 | ACENQRS | R03201-00 | TIMES SQ- | 3/31/2020 | 12:00-3:59pm | 40.75591 | -73.9865 | 9683031 | 28 |
| 24 | ACENQRS | R03201-00 | TIMES SQ- | 3/31/2020 | 4:00-7:59pm | 40.75591 | -73.9865 | 9683071 | 40 |

*Figure 23- Microsoft Excel aggregation phase*

*Tableau Desktop (Version 2 data visualization output)*

The sequence of snapshots below showcases different time sequences of the commuter congestion data for May 1st, 2020. Tableau Desktop has the capability to perform time-sequenced visualizations of underlying data at different snapshots in time which can be run interactively or by animation.


Map - May 1, 2020, 12:00-3:59am

Map - May 1, 2020, 4:00-7:59am

Map - May 1, 2020, 8:00-11:59am

Map - May 1, 2020, 12:00-3:59pm

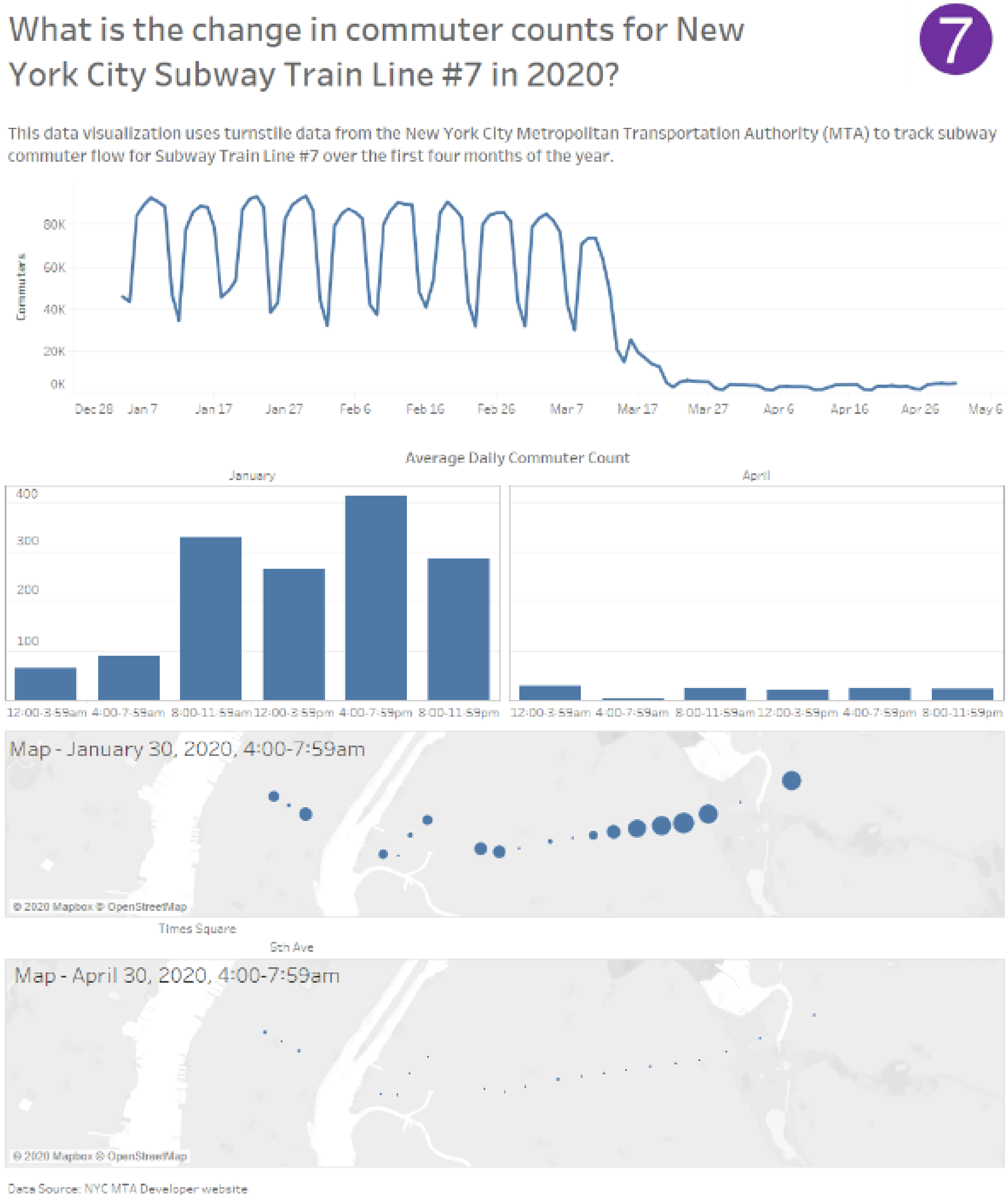
Map - May 1, 2020, 4:00-7:59pm

Map - May 1, 2020, 8:00-11:59pm

*Figure 24- Screenshots of data for time periods for May 1st, 2020*

### *Dashboard Design Information Graphic (Version 2)*

This information graphic uses the cleaned dataset generated by the Microsoft Excel spreadsheet. It displays a line plot of 2020 commuter flow, two bar charts for analyze the average 2020 commuter flow for the months of January and April and then two maps that take a snapshot of the 4:00 am to 7:59 am time period for two different days



What is the change in commuter counts for New York City Subway Train Line #7 in 2020?

This data visualization uses turnstile data from the New York City Metropolitan Transportation Authority (MTA) to track subway commuter flow for Subway Train Line #7 over the first four months of the year.

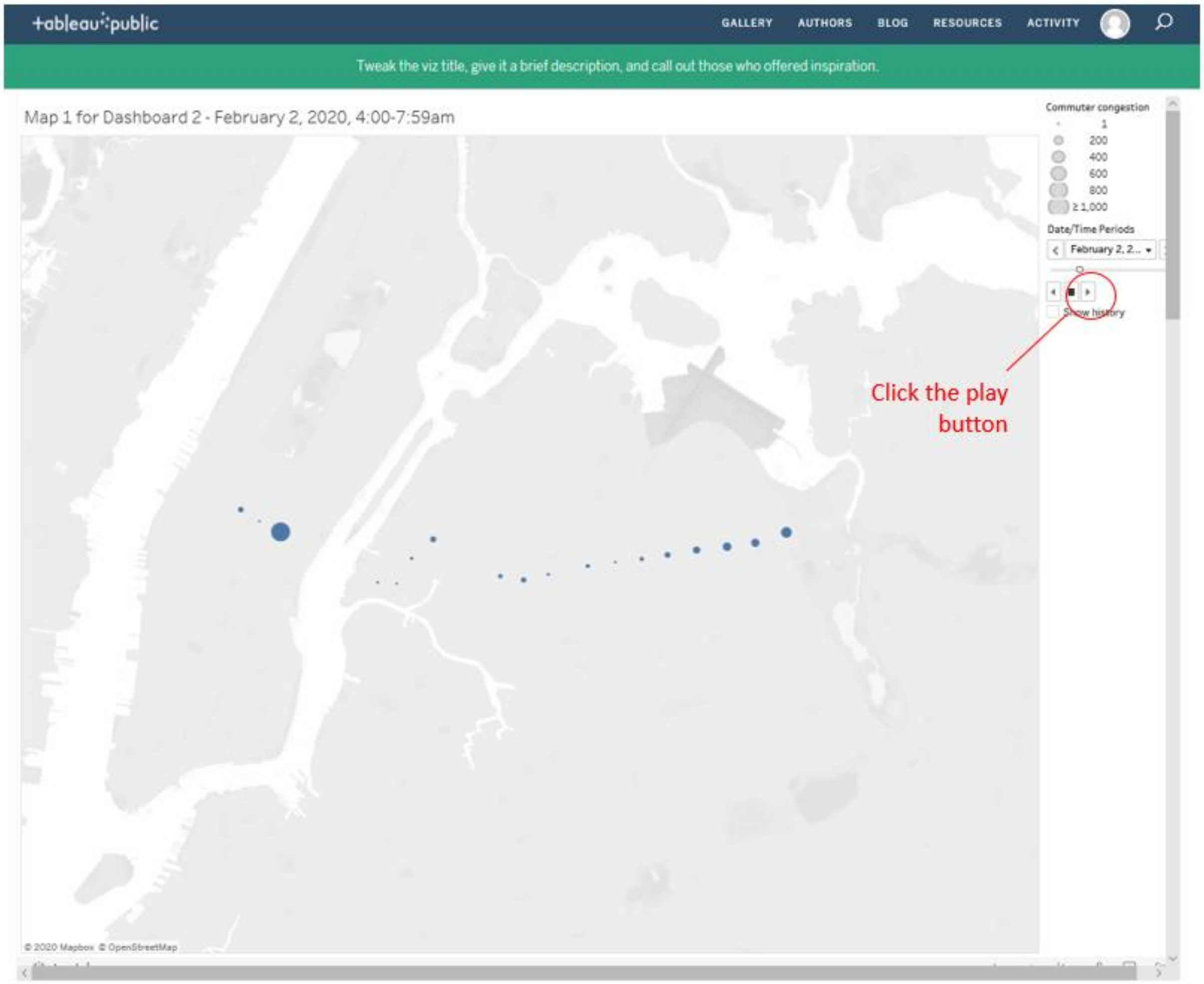
Data Source: NYC MTA Developer website

*Figure 25- Information Graphic using Tableau Desktop Dashboard (Version 2)*

## Results

Tableau suite of data visualization functionality includes the ability to create data visualizations that can incorporate interactivity. In addition, this interactivity can be harnessed  to generate animation sequences. This capability can then be uploaded to the internet.  Below is a link to Tableau Public that can be used to interactively interface with the online application, in this case one of the map views created during this project. Users can also view the data by running it as an animation sequence (i.e. click the 'Play' button).

https://public.tableau.com/profile/peter.r5258#!/vizhome/mta_2020_subway_train_7_subway_turnstile_data_tracker_v10/Map1forDashboard2?publish=yes



Tweak the viz title, give it a brief description, and call out those who offered inspiration.

Map 1 for Dashboard 2 - February 2, 2020, 4:00-7:59am

Click the play button

*Figure 26- Interactive Subway Turnstile Map loaded into Tableau Public*

## Challenges/Limitations

- Need to find a good way to interactively add labels to the map portion of the dashboard
- Objects available for dashboard design do not provide options to draw leader lines
- The animation sequence became beneficial in spotting time frames that showed missing data values

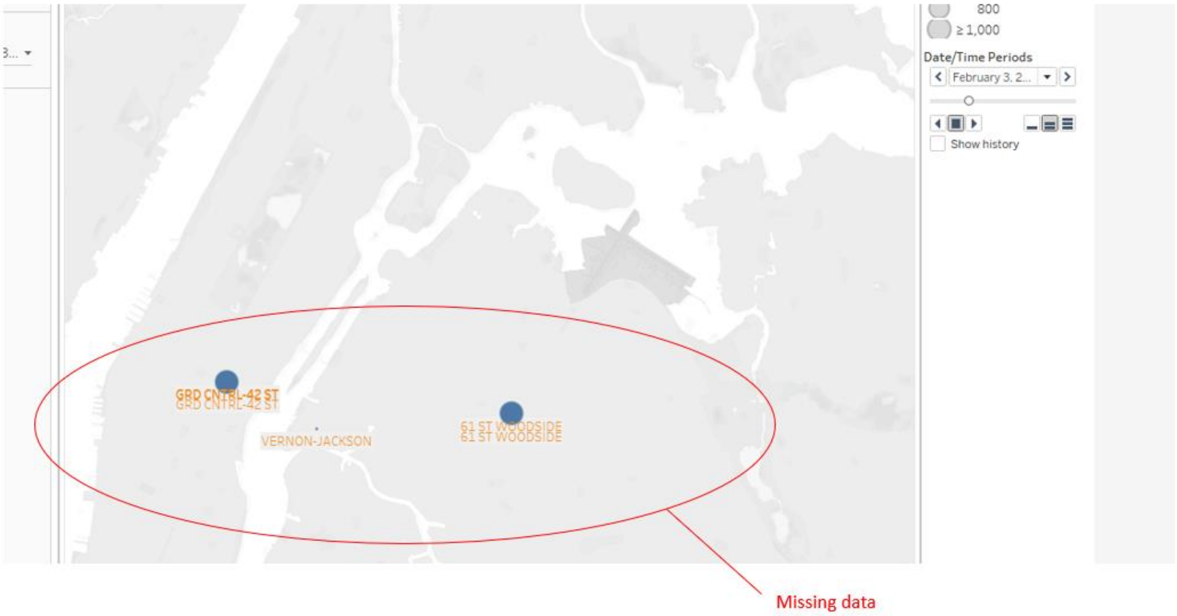


*Figure 27- Work in Progress*

## References

Cairo, Alberto. (2019). *How Charts Lie*. W.W. Norton & Company, Inc., 500 Fifth Avenue, New York, N.Y. 10110

census.gov. (n.d.). *Quick Facts: New York City. United States Census Bureau*. Retrieved on November 9, 2019 from https://www.census.gov/quickfacts/fact/table/newyorkcitynewyork/PST045218

chriswhong.com. (2015). *Visualizing the MTA's Turnstile Data.* Retrieved on November 9, 2019 from https://chriswhong.com/open-data/visualizing-the-mtas-turnstile-data/

Costello, Tim; Blackshear, Lori. (2020). *Prepare Your Data for Tableau*. A Practical guide to the Tableau Data Prep Tool. Apress

Craig, James; Scala, Irene Korol; Bevington, William. (2006). *Designing with Type: the essential guide to typography*. Watson-Guptill Publications, a division of VNU Business Media, Inc., 770 Broadway, New York, NY 10003

datacamp.com. Retrieved on 10/24/19 from https://www.datacamp.com/home (need an account)

De Vries, Andrie; Meys, Joris. (2015). *R for Dummies*. John Wiley & Sons, Inc., Hoboken, New Jersey

rdocumentation.org.(n.d.). download.file, RDocumentation. Retrieved on 5/10/2020 from https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/download.file

Few, Stephen. (2013). *Information Dashboard Design.* Analytics Press. Publishers: Koomey, Jonathan G.

Field, Andy; Miles, Jeremy; Fields, Zoe. (2012). *DISCOVERING STATISTICS USING R.* SAGE Publications Ltd, 1 Oliver's Yard, 55 City Road, London EC1Y 1SP

flowingdata.com. (n.d.). Main page of the 'Flowingdata' Learning Blog. Retrieved on May 10, 2020 from https://flowingdata.com/

Grolemund, Garrett. Hands-On Programming with R. O'Reilly Media, Inc.

github.com. Reading May and Beg. Of June Data. Retrieved on May 10, 2020 from https://github.com/lpalova/MTA-analysis/blob/master/Lucia-Benson-project.ipynb

groups.google.com. (n.d.). *Access MTA Data -API Key.* Retrieved on November 8, 2019 from https://groups.google.com/forum/#!topic/mtadeveloperresources/6Bou6mrMYnc%5B1-25%5D

help.tableau.com. (n.d.). *Show, Hide, and Format Mark Labels.* Tableau Desktop and Authoring Help. Retrieved on May 10, 2020 from https://help.tableau.com/current/pro/desktop/en-us/annotations_marklabels_showhideworksheet.htm

help.tableau.com. (n.d.). Save Workbooks to Tableau Public. Tableau Desktop and Web Authoring Help. Retrieved on May 10, 2020 from https://help.tableau.com/current/pro/desktop/en-us/publish_workbooks_tableaupublic.htm

Jones, Ben. (2020). *AVOIDING DATA PITFALLS.* John Wiley & Sons, Inc. Hoboken, New Jersey.

Knaflic, Cole Nussbaumer. (2015). *storytelling with data.* John Wiley & Sons, Inc. Hoboken, New Jersey.

Knaflic, Cole Nussbaumer. (2020). *storytelling with data, let’s practice.* John Wiley & Sons, Inc. Hoboken, New Jersey.

Kriebel, Andy; Murray, Eva.(2018). *Makeover Monday, Improving How We Visualize and Analyze Data, One Chart at a Time*. John Wiley & Sopns, Inc, Hoboken, New Jersey.

McCandless, David. (2014). *Knowledge is Beautiful.* Harper Collins Publishers. 77-85 Fulham Palace Road, Hammersmith, London W6 8JB

McDaniel, Eileen, PHD; McDaniel Stephen. (2011-2012). *The Accidental Analyst. Show Your Data Who’s Boss.* Freakalytics, LLC. Seattle, WA.

medium.com. (Feb 2, 2018). *NYC Turnstile Data*. Retrieved on November 8, 2019. https://medium.com/@michaelkduchak/nyc-turnstile-data-fb5fc7019a21

Milligen, Joshua N. (2019). *Learning Tableau 2019.* Packt Publishing Ltd.

Nelson, John. (2019). *Atlas of Design (Volume Three). UFO Sitings Dashboard.* Editors: Matthews, Samuel; Elmer, Martin. Published by North American Cartographic Information Society.

new.mta.info, (n.d.). *Developer page.* Retrieved on 10/23/19 from https://new.mta.info/developers/open-data

nycsubwayguide.com. (n.d.). Minh T. Nguyen. *The Absolute Beginner’s Guide to the New York Subway*. Retrieved on November 6th, 2019 from http://www.nycsubwayguide.com/subway/default.aspx

Parker, Spencer. (May 9, 2018). *Tableau Prep: How to Cleanse Your Data and Prepare it for ~~World Domination~~ Analysis.* Retrieved from https://interworks.com/blog/sparker/2018/05/09/tableau-prep-how-to-cleanse-your-data-and-prepare-it-for-world-domination-analysis

rstudio-pubs-static.s3.amazonaws.com. (14 June, 2015). Raymond Carl. *NYC Subway Data*. Retrieved on November 8, 2019 from https://rstudio-pubs-static.s3.amazonaws.com/92916_4afa18f2f01344f49ea182770095e7db.html

rstudio-pubs-static.s3.amazonaws.com. (23 June, 2015). Raymond Carl. *MTA Subway Data v2*. Retrieved on November 8, 2019 from https://rstudio-pubs-static.s3.amazonaws.com/92744_51eb2f1ecce040caaea481928ed43d6f.html

rstudio.com. (n.d.). Data Wrangling with dplyr and tidyr Cheat sheet R Studio. Retrieved on May 10, 2020 from https://rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf

rviews.rstudio.com. What is the tidyverse? Retrieved on May 10, 2020 from https://rviews.rstudio.com/2017/06/08/what-is-the-tidyverse/

Schmuller PhD, Joseph. (2017). *Statistical Analysis with R for dummies.* John Wiley & Sons, Inc., Hoboken, New Jersey

tableau.com.(n.d.). Free Training Transcript: The Input Step. Retrieved on May 10, 2020 from https://www.tableau.com/sites/default/files/the_input_step_0.pdf

tableau.com. (2019). Understanding Pill Types. Retrieved on May 11th, 2020 from https://www.tableau.com/learn/tutorials/on-demand/understanding-pill-types?_ga=2.122845387.155557573.1584445803-423265706.1581652010&_fsi=B3XQDnzS&signin=b470c38192cb74a160311844b9c15736

Tomlinson, Roger. (2003). Thinking About GIS. Esri Press, *The 10-stage GIS planning methodology*. (p. 13 – 17). 380 New York street, Redlands, California. 10475 Crosspoint Boulevard, Indianapolis, IN 46256

transitdatatoolkit.com. (n.d.). 7 – *Subway Turnstile Data.* Retrieved on November 8, 2019 from http://transitdatatoolkit.com/lessons/subway-turnstile-data/

vita.had.co.nz. (n.d.). *Tidy Data. Journal of Statistical Software.* Retrieved on November 9, 2019 from http://vita.had.co.nz/papers/tidy-data.pdf

Ware, Colin.(2018). *VISUAL THINKING for DESIGN*. Elsevisor, Inc.

web.mta.info. (n.d.). *The Weekender*. Retrieved on November 8, 2019 from http://web.mta.info/weekender.html

Wexler, Steve; Shaffer, Jeffrey; Cotgrove, Andy. (2017). *THE BIG BOOK OF DASHBOARDS.* John Wiley & Sons, Hoboken, New Jersey.

Yau, Nathan. (2011). *VISUALIZE THIS.* Wiley Publishing, Inc. 10475 Crosspoint Boulevard. Indianapolis, IN 46256

Zandbergen, Paul A. (2013). Esri Press. *Python Scripting for ArcGIS*. 380 New