Using Data Visualization to Explore the Problem with the MTA's Operations

Jessica Padilla

September 27, 2019

Introduction

The Metropolitan Transportation Authority (MTA) is responsible for overseeing the transportation network throughout New York City and its surrounding regions. The MTA consists of New York City Transit, Long Island Railroad, Metro-North Railroad, Staten Island Railway, Regional Bus Operations, and B&T (Bridges and Tunnels). For this project, we focus on the New York City Transit subways provided by the MTA.

In 1979, the subway system was completely deteoriated. In order to revive the system, the MTA needed more funds than the State of New York could provide annually. As a result, the MTA continuously sold bonds and took on several loans to rebuild infrastructure, in hopes that the debt would eventually be paid off by fare revenue (Rivera 1-3). However, many decades later, there is a decrease in subway ridership and an increase in alternative forms of transportation. Due to all these factors, the MTA debt is expected to approach 42 billion dollars by the year 2022 (Walker 1-2).

Since May 2003, the MTA has changed the subway fare 6 times to try to offset the debt ("New York City Transit Fares"). To aggravate the situation, the MTA is hoping to introduce a $50 billion plan for capital improvements ("Public Can Weigh in on MTA's $50 Billion Capital Plan"). They are also in the process of hiring an additional 500 transit cops to deal with the increase in subway assaults and fare evasion (Meyer, David et. al).

Here we take an in depth look at the MTA's subway services. We also discuss how finances can be re-allocated to assist with the MTA's ever-increasing debt and in the long run, provide a better transportation system for commuters.

Methods

MTA revenue, expenses, and budget data were found on the "MTA Fiscal Dashboard" of the Citizens Budget Commission website (https://cbcny.org/research/mta-fiscal-dashboard).

Average subway ridership totals for weekdays and weekends were retrieved from the official MTA website (http://web.mta.info/nyct/facts/ridership/).

Information in regards to MTA subway train totals, commute times, and incidents was obtained through the "Performance Dashboard" of the MTA website (http://dashboard.mta.info).

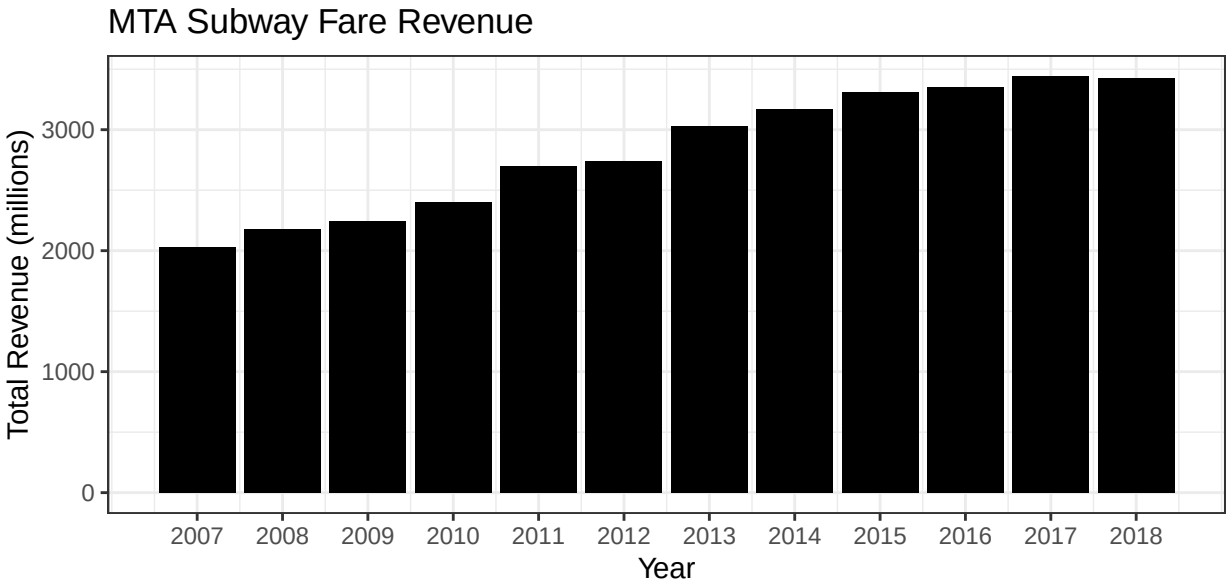
Most of the data was exported in the form of .csv files and uploaded onto GitHub (https://github.com/jessicapadilla/mta_finances). If there was no export option available, data was retrieved from the site using web scraping. This involved extracting tables from the html code within the webpage and converting them into data tables.

All of the data was, then, imported into R. The rvest package within R was used to obtain data from the .csv files and websites. The tidyverse package was used for data cleaning, analysis, and visuzalition. The gridExtra, RColorBrewer, and ggpubr packages within R were also used to enhance the aesthetics of the graphs in this paper.

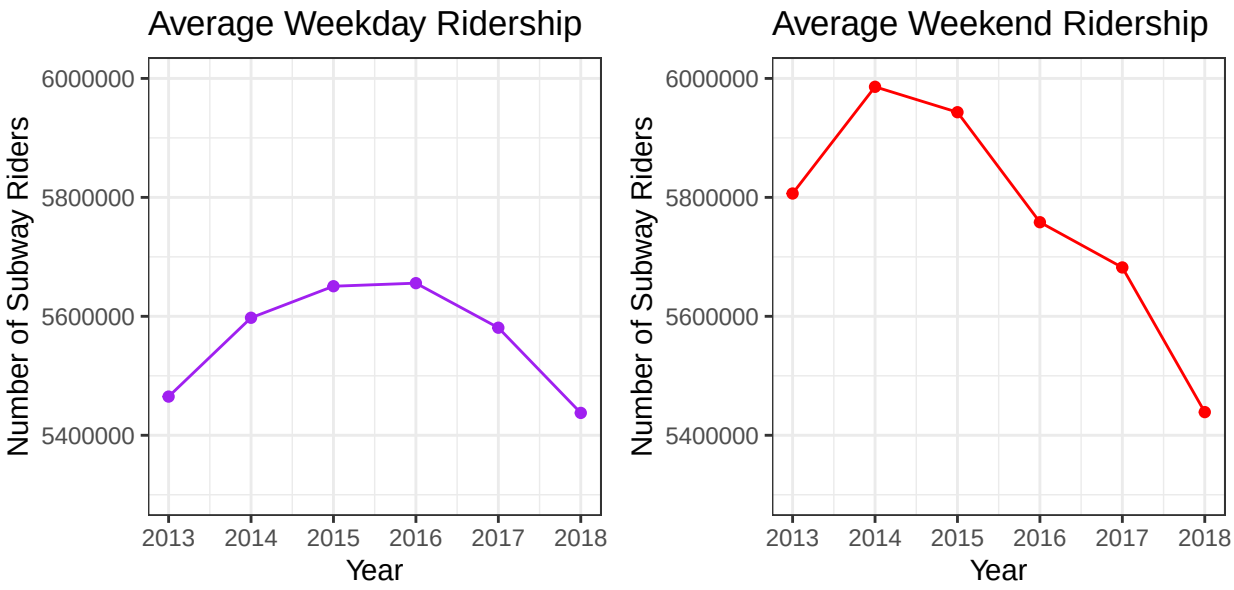
The R code for this project can be found within the Supplementary Materials section of this paper and on GitHub (https://github.com/jessicapadilla/mta_finances/blob/master/code.R).

Results

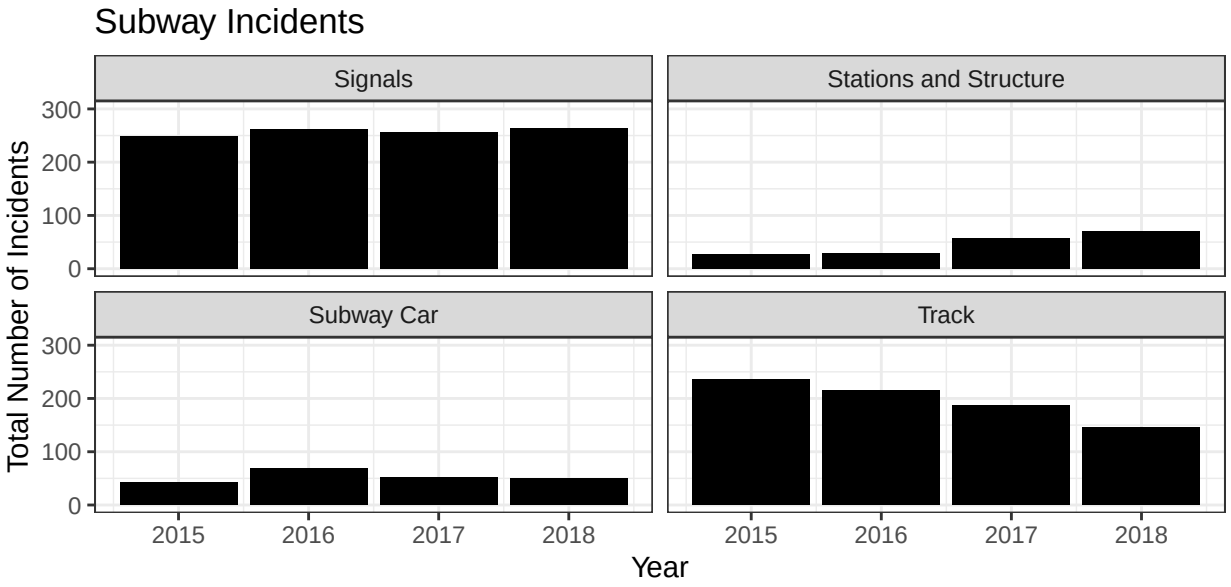
The debt of the MTA has grown significantly over the years and is projected to reach $42 billion by the year 2022 (Walker 1-2). While fare revenue has been able to offset a bit of the debt, fare revenue has reached a standstill. The MTA Fiscal Dashboard data obtained from the Citizens Budget Commission shows that although the fare revenue is much larger than what it was in 2007, the amount of fare revenue has slowed down since 2013. In fact, there was a slight decrease in fare revenue in 2018.



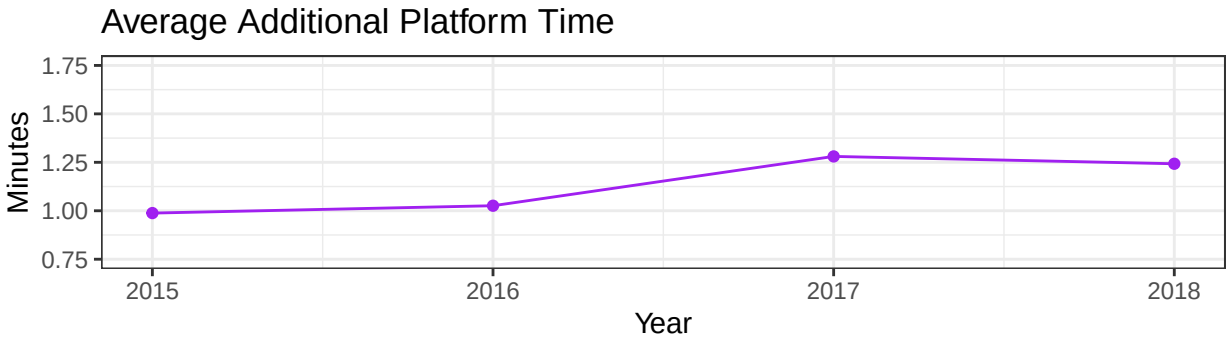
The decrease in fare revenue in 2018 can be explained by the downward trend in subway ridership. Data obtained from the MTA site showed that average weekday ridership was 5,465,034 in 2013, but declined to 5,437,587 in 2018. Average weekend ridership also declined significantly from 5,806,517 in 2013 to 5,438,947 in 2018.

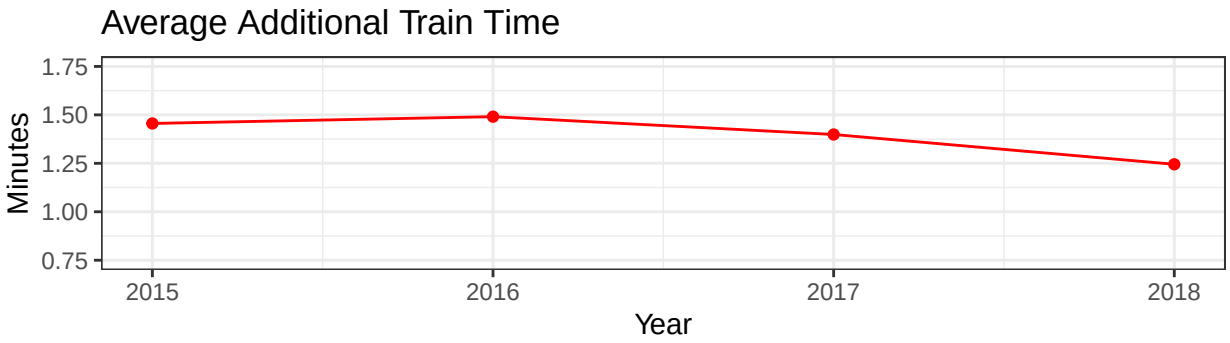


Over the years, subway riders have expressed their discontent over the number of subway incidents that delayed their commutes. The MTA Performance Dashboard shows that signal incidents have not improved and occur approximately 250 times every year. The number of station and structure incidents have also increased from 27 in 2013 to 70 in 2018. Subway car incidents have stabilized to about 40 per year, while track incidents have improved (235 in 2015 versus 146 in 2018) but still remain a significant problem.

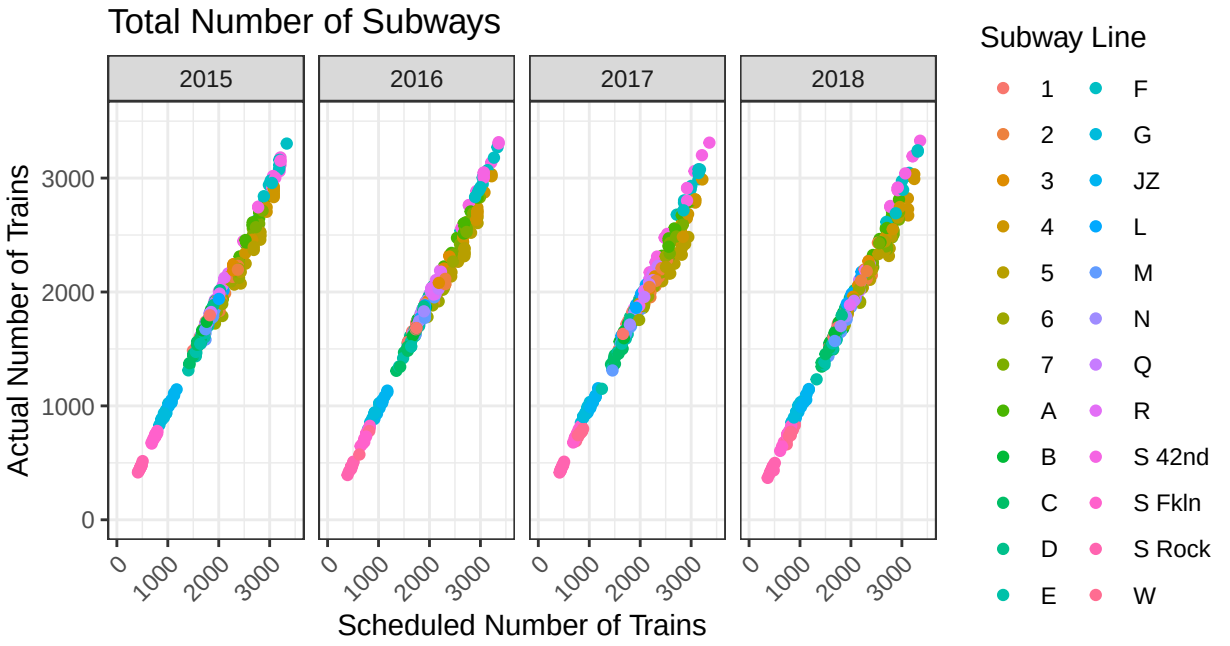


The problematic subway infrastructure has led to passengers waiting on the platform for approximately an additional 1.00 to 1.25 minutes and waiting on the train for approximately an additional 1.25 to 1.50 minutes.



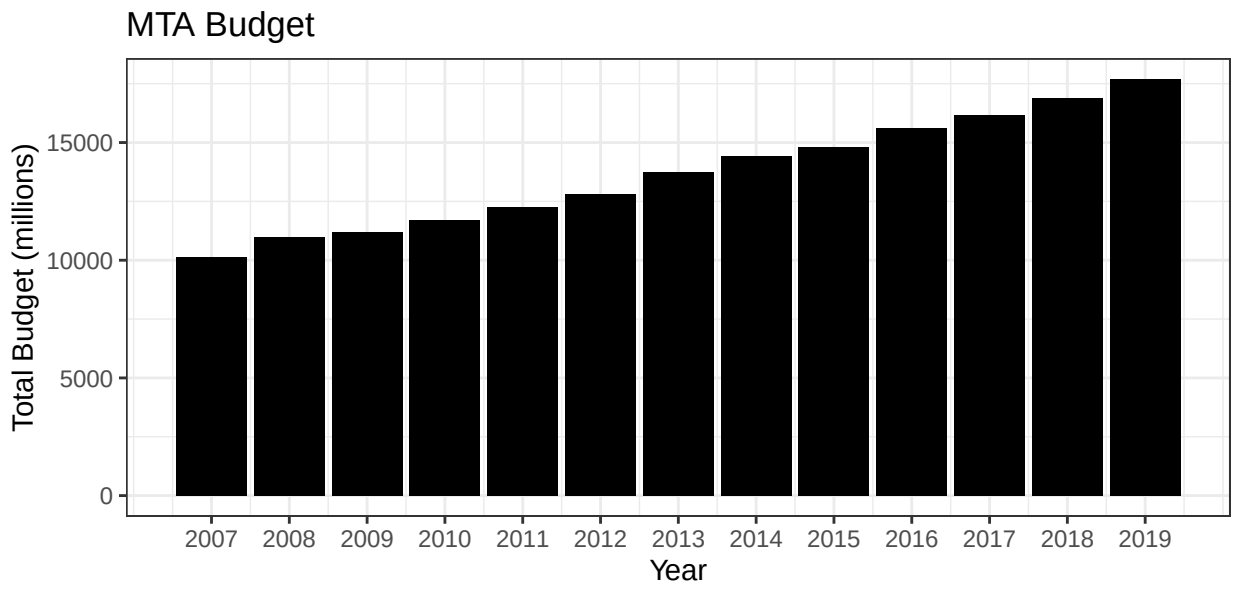


Data from the MTA site showed that the MTA also had a number of train cancellations throughout the years. To demonstate this, the number of scheduled trains was plotted against the number of actual trains. If the number of scheduled trains equalled the number of actual trains, the plotted dots would overlap and form a perfect line. However, as seen in the graphs below, the imperfect line indicates that the number of scheduled trains rarely equalled the number of actual trains. This also means that there were fewer actual trains in service.

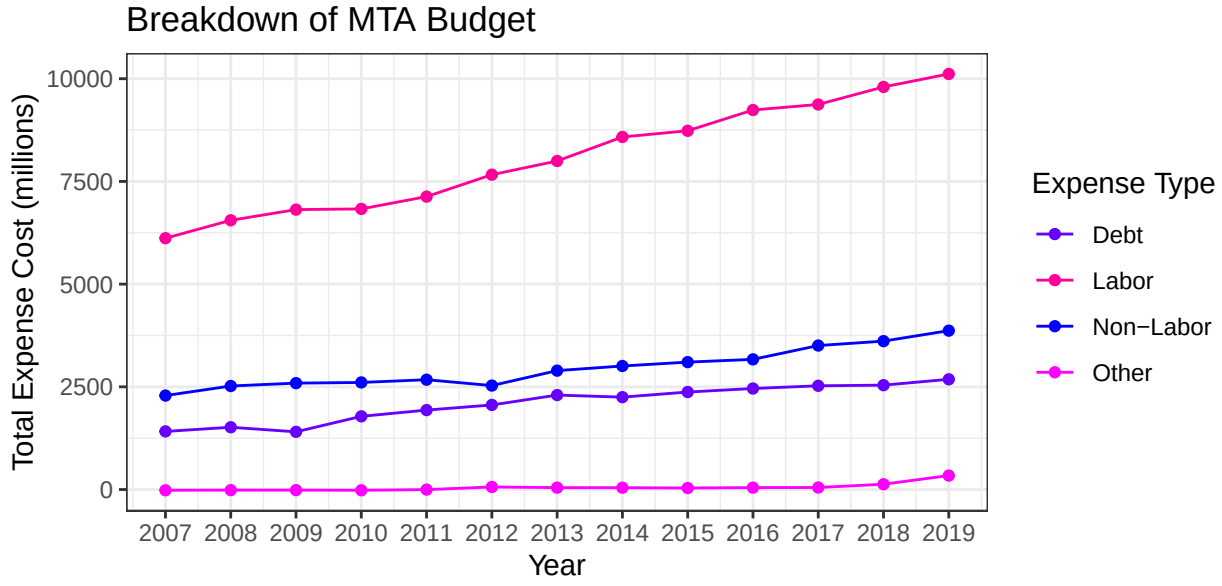


In the midst of all the problems with subway service, the MTA hopes to utilize another $50 billion to repair

subway stations, to add new signal systems, and to modernize the look of subway cars. This is in addition to the escalating MTA budget shown below (Meyer, David et. al). But is such a massive new expense necessary?

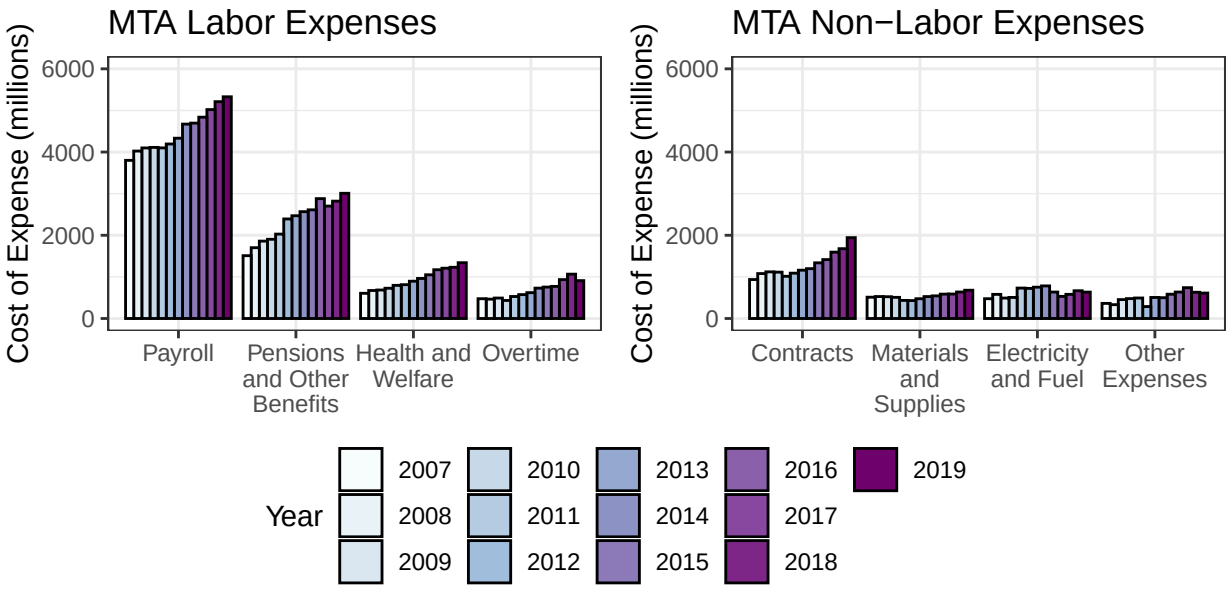


A breakdown of the MTA budget shows that it is comprised of four types of expenses: debt, labor, non-labor, and other. While the debt, non-labor, and other expenses have increased since 2007, the labor expenses have almost doubled in the same time period.



The labor expenses include payroll, pensions and other benefits, health and welfare, and overtime, while the non-labor expenses include contracts, materials and supplies, electricity and fuel, and other expenses. Data shows that all of the labor expenses have increased significantly throughout the years. The non-labor expenses for contracts have also increased as well, but the amount spent on the rest of the non-labor expenses remains relatively the same.

A side-by-side comparison also reveals that the MTA spends more on payroll alone than all of the non-labor expenses combined. In addition, the MTA spends more on overtime than on either materials and supplies or electricity and fuel.



Conclusion

The MTA's plan to tackle its debt with the use of fare revenue is no longer a realistic plan, due to the decrease in subway ridership throughout the week. This could possibly be due to the crippling infrastructure which includes signal, track, subway car, and station issues. These lead to either delays in commute time and train cancellations.

With the new plan unveiled by the MTA to spend another $50 billion in improving the subway system, one needs to ask if a new burden of debt is necessary. Upon inspecting the MTA's expenses, payroll was revealed to cost the most out all types of expenses. Moreover, overtime costs more than either materials and supplies or electricity and fuel. If money from payroll and/or overtime were reallocated towards subway system improvements, then perhaps the MTA would not need to spend another $50 billion. If the subway system was renovated and repaired, perhaps more commuters would start to use the subway again and lead to an increase in fare revenue. Improvements to the MTA are absolutely necessary, especially if it plans to stay dominant in the transportation industry amongst new transportation rivals, such as CitiBike, Uber, and Lyft.

References

"Introduction to Subway Ridership", *MTA*, web.mta.info/nyct/facts/ridership.

Meyer, David et al. "MTA Hiring 500 New Transit Cops in Wake of Rising Assaults", *New York Post*, NYP Holdings, Inc., 12 Sept. 2019, nypost.com/2019/09/12/mta-hiring-500-new-transit-cops-in-wake-of-rising-assaults.

"MTA Fiscal Dashboard." *Citizens Budget Commission*, Citizens Budget Commission, 31 Aug. 2019, cbcny.org/research/mta-fiscal-dashboard.

"MTA Subway Performance Dashboard"", *MTA*, dashboard.mta.info.

"New York City Transit Fares", *Wikipedia*, Wikipedia, en.wikipedia.org/wiki/New_York_City_transit_fares.

"Public Can Weigh in on MTA's $50 Billion Capital Plan." *AP*, The Associated Press, 12 Nov. 2019, apnews.com/de4925cbb8634c0fa126ef31e3c6a959.

Rivera, Ray. "M.T.A. and Its Debt, and How They Got That Way." *The New York Times*, The New York Times, 26 July 2008, nytimes.com/2008/07/26/nyregion/26mta.html.

Walker, Ameena. "MTA Could Face $42B in Outstanding Debt by 2022: Report." *Curbed New York*, Vox Media, 11 Oct. 2018, ny.curbed.com/2018/10/11/17964786/mta-budget-deficit-debt-report-thomas-dinapoli.

Supplementary Material

This data science project was completed with the use of the R code below.

```
## load libraries
library(tidyverse)
library(rvest)
library(gridExtra)
library(RColorBrewer)
library(ggpubr)

## retrieve the mta revenue data
mta_revenue <- read_csv(
  "https://github.com/jessicapadilla/problem_with_mta/raw/master/mta_revenue.csv")

## check the structure of the mta revenue data
str(mta_revenue)

## select needed columns and rename them
mta_revenue <- mta_revenue %>%
  select(c("Business Line (group)", "Business Line",
           "Description1", "Revenue and Expense Type",
           "Year", "Value")) %>%
  rename(business_line_group = "Business Line (group)",
         business_line = "Business Line",
         revenue_description = "Description1",
         revenue_type = "Revenue and Expense Type",
         mta_year = "Year",
         revenue_value = "Value")

## change all NA values in the revenue value column to 0
mta_revenue$revenue_value[is.na(mta_revenue$revenue_value)] = 0
```

```
## check the mta revenue data
head(mta_revenue)

## create a graph for subway fare revenue totals each year
## exclude year 2019 since the year is not yet completed
mta_revenue %>% filter(mta_year <= 2018 & revenue_description == "Subway") %>%
  ggplot(aes(mta_year, revenue_value)) +
  geom_bar(stat = "identity", fill = "black") +
  scale_x_continuous(breaks = seq(2007, 2018, 1)) +
  xlab("Year") + ylab("Total Revenue (millions)") +
  ggtitle("MTA Subway Fare Revenue") +
  theme_bw()

## set the link for subway ridership data
subway_ridership_url <- "http://web.mta.info/nyct/facts/ridership/"

## get the html code from the webpage
subway_ridership_html <- read_html(subway_ridership_url)

## get the html nodes
subway_ridership_nodes <- subway_ridership_html %>% html_nodes("table")

## find the subway table
subway_ridership_nodes

## turn the table to a data frame
subway_ridership <- subway_ridership_nodes[[2]] %>% html_table

## check the structure of the subway ridership data
str(subway_ridership)

## select the needed columns
## rename the year column
## then convert the data to a tibble
subway_ridership <- subway_ridership %>%
  select(c("Year", "Average Weekday", "Average Weekend")) %>%
  rename(mta_year = "Year") %>% as_tibble()

## gather the columns
subway_ridership <- subway_ridership %>%
  gather(category, totals, "Average Weekday":"Average Weekend")

## remove commas from the totals column and convert it to numbers
subway_ridership$totals <- subway_ridership$totals %>%
  str_replace_all(",", "") %>% as.numeric()

## create a graph for average weekday subway ridership
weekday <- subway_ridership %>%
  filter(category == "Average Weekday") %>%
  ggplot(aes(mta_year, totals)) +
  geom_point(color = "purple") + geom_line(color = "purple") +
  scale_x_continuous(breaks = seq(2013, 2018, 1)) +
  coord_cartesian(ylim = c(5300000, 6000000)) +
```

```
  xlab("Year") + ylab("Number of Subway Riders") +
  ggtitle("Average Weekday Ridership") +
  theme_bw()

## create a graph for average weekend subway ridership
weekend <- subway_ridership %>%
  filter(category == "Average Weekend") %>%
  ggplot(aes(mta_year, totals)) +
  geom_point(color = "red") + geom_line(color = "red") +
  scale_x_continuous(breaks = seq(2013, 2018, 1)) +
  coord_cartesian(ylim = c(5300000, 6000000)) +
  xlab("Year") + ylab("Number of Subway Riders") +
  ggtitle("Average Weekend Ridership") +
  theme_bw()

## put the ridership graphs side by side
grid.arrange(weekday, weekend, ncol = 2)

## retrieve subway incidents data
subway_incidents <- read_csv(
  "https://github.com/jessicapadilla/problem_with_mta/raw/master/
  subway_major_incidents.csv")

## check the structure of the subway incidents table
str(subway_incidents)

## remove the division column and rename the category column
subway_incidents <- subway_incidents %>% select(-division) %>%
  rename(type_of_issue = category)

## add a year column using the values from the month column
subway_incidents <- subway_incidents %>% mutate(mta_year = month)

## edit the year column by removing the month
## change the year to numbers
subway_incidents$mta_year <- subway_incidents$mta_year %>%
  str_replace("^(\\d{4})-(\\d{2})$", "\\1") %>% as.numeric()

## check the different types of issues
unique(subway_incidents$type_of_issue)

## reorder the types of issue
subway_incidents$type_of_issue <- factor(subway_incidents$type_of_issue,
                                levels = c("Signals",
                                           "Stations and Structure",
                                           "Subway Car", "Track",
                                           "Persons on Trackbed/Police/Medical",
                                           "Other"))

## check the subway incidents table
head(subway_incidents)

## create a graph showing the number of subway incidents per year
```

```
## exclude 2019 since the year is not yet completed
subway_incidents %>%
  filter(mta_year <= 2018 & type_of_issue %in%
           c("Signals", "Stations and Structure", "Subway Car", "Track")) %>%
  group_by(mta_year, type_of_issue) %>%
  summarize(total_count = sum(count)) %>%
  ggplot(aes(mta_year, total_count)) +
  geom_bar(stat = "identity", fill = "black") +
  coord_cartesian(ylim = c(0, 300)) +
  facet_wrap(.~type_of_issue, ncol = 2) +
  xlab("Year") + ylab("Total Number of Incidents") +
  ggtitle("Subway Incidents") + theme_bw()

## retrieve subway platform times data
subway_platform_times <- read_csv(
  "https://github.com/jessicapadilla/problem_with_mta/raw/master/
  subway_platform_time.csv")

## check the structure of the subway platform table
str(subway_platform_times)

## remove the division and number of passengers columns
## rename the platform time column
subway_platform_times <- subway_platform_times %>%
  select(-c("division", "num_passengers")) %>%
  rename(additional_platform_time = "additional platform time")

## add a year column by using the values from the month column
subway_platform_times <- subway_platform_times %>% mutate(mta_year = month)

## edit the year column by removing the month and change the year to numbers
subway_platform_times$mta_year <- subway_platform_times$mta_year %>%
  str_replace("^(\\d{4})-(\\d{2})$", "\\1") %>% as.numeric()

## check the subway platform table
head(subway_platform_times)

## create a graph for additional platform times
## exclude 2019 since the year is not yet completed
add_platform <- subway_platform_times %>%
  filter(mta_year <= 2018) %>% group_by(mta_year) %>%
  summarize(average_additional_platform = mean(additional_platform_time)) %>%
  ggplot(aes(mta_year, average_additional_platform)) +
  geom_point(color = "purple") + geom_line(color = "purple") +
  coord_cartesian(ylim = c(0.75, 1.75)) +
  xlab("Year") + ylab("Minutes") +
  ggtitle("Average Additional Platform Time") +
  theme_bw()

## retrieve subway train times data
subway_train_times <- read_csv("https://github.com/jessicapadilla/problem_with_mta/raw/
                               master/subway_train_time.csv")
```

```
## check the structure of the subway train times table
str(subway_train_times)

## remove the division and number of passengers columns
## rename the train time column
subway_train_times <- subway_train_times %>%
  select(-c("division", "num_passengers")) %>%
  rename(additional_train_time = "additional train time")

## add a year column by using the values from the month column
subway_train_times <- subway_train_times %>% mutate(mta_year = month)

## edit the year column by removing the month and change the year to numbers
subway_train_times$mta_year <- subway_train_times$mta_year %>%
  str_replace("^(\\d{4})-(\\d{2})$", "\\1") %>% as.numeric()

## check the subway train times table
head(subway_train_times)

## create a graph for additional train times
## exclude 2019 since the year is not yet completed
add_train <- subway_train_times %>%
  filter(mta_year <= 2018) %>% group_by(mta_year) %>%
  summarize(average_additional_train = mean(additional_train_time)) %>%
  ggplot(aes(mta_year, average_additional_train)) +
  geom_point(color = "red") + geom_line(color = "red") +
  coord_cartesian(ylim = c(0.75, 1.75)) +
  xlab("Year") + ylab("Minutes") +
  ggtitle("Average Additional Train Time") +
  theme_bw()

## put the additional time graphs side by side
grid.arrange(add_platform, add_train, ncol = 1)

## retrieve subway delivered data
subway_delivered <- read_csv("https://github.com/jessicapadilla/problem_with_mta/raw/
                             master/subway_service_delivered.csv")

## check the structure of the subway delivered table
str(subway_delivered)

## remove the division column and rename some columns
subway_delivered <- subway_delivered %>% select(-division) %>%
  rename(scheduled_trains = num_sched_trains,
         actual_trains = num_actual_trains)

## add a year column by using the values from the month column
subway_delivered <- subway_delivered %>% mutate(mta_year = month)

## edit the year column by removing the month and change the year to numbers
subway_delivered$mta_year <- subway_delivered$mta_year %>%
  str_replace("^(\\d{4})-(\\d{2})$", "\\1") %>% as.numeric()
```

```r
## check the subway delivered times table
head(subway_delivered)

## create a graph showing the number of subways each year
## exclude year 2019 since the year is not yet completed
subway_delivered %>% filter(mta_year <= 2018) %>%
  mutate(mta_year = factor(mta_year)) %>%
  ggplot(aes(scheduled_trains, actual_trains, col = line)) +
  geom_point() + facet_grid(. ~ mta_year) +
  scale_color_discrete(name = "Subway Line") + xlim(0, 3500) + ylim(0, 3500) +
  xlab("Scheduled Number of Trains") + ylab("Actual Number of Trains") +
  ggtitle("Total Number of Subways") + theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

## retrieve mta budget data
mta_budget <- read_csv("https://github.com/jessicapadilla/problem_with_mta/raw/master/
                       mta_budget.csv")

## check the structure of the mta budget table
str(mta_budget)

## remove unwanted columns and rename some columns
mta_budget <- mta_budget %>%
  select(c("Business Line", "Description1",
           "Revenue and Expense Type",
           "Year", "Value")) %>%
  rename(business_line = "Business Line",
         expense_description = "Description1",
         expense_type = "Revenue and Expense Type",
         mta_year = "Year",
         expense_cost = "Value")

## check how many expense types there are
unique(mta_budget$expense_type)

## rename the expense types
mta_budget$expense_type <- mta_budget$expense_type %>%
  str_replace("Labor Expense", "Labor") %>%
  str_replace("Non-labor Expense", "Non-Labor") %>%
  str_replace("Debt Service", "Debt") %>%
  str_replace("Other Expense Adjustments", "Other")

## check how many expense descriptions there are for each expense type
unique(mta_budget$expense_description[mta_budget$expense_type == "Labor"])
unique(mta_budget$expense_description[mta_budget$expense_type == "Non-Labor"])
unique(mta_budget$expense_description[mta_budget$expense_type == "Debt"])
unique(mta_budget$expense_description[mta_budget$expense_type == "Other"])

## rename some of the expense descriptions
mta_budget <- mta_budget %>% mutate(expense_description = case_when(
  expense_type == "Labor" & expense_description == "Payroll" ~ "Payroll",
  expense_type == "Labor" & expense_description == "Overtime" ~ "Overtime",
  expense_type == "Labor" & expense_description == "Health and Welfare" ~
```

```
    "Health and Welfare",
  expense_type == "Labor" & expense_description %in%
    c("OPEB Current Payment", "Pensions", "Pensions Offset", "Other Fringe Benefits") ~
    "Pensions and Other Benefits",
  expense_type == "Labor" & expense_description == "Reimbursable Overhead" ~
    "Reimbursable Overhead",
  expense_type == "Non-Labor" & expense_description %in%
    c("Electric Power", "Fuel") ~ "Electricity and Fuel",
  expense_type == "Non-Labor" & expense_description %in%
    c("Insurance", "Claims", "Other Business Expenses") ~ "Other Expenses",
  expense_type == "Non-Labor" & expense_description %in%
    c("Paratransit Service Contracts", "Maintenance and Other Operating Contracts",
      "Professional Service Contracts") ~ "Contracts",
  expense_type == "Non-Labor" & expense_description == "Materials and Supplies" ~
    "Materials and Supplies",
  expense_type == "Debt Services" & expense_description %in%
    c("Total MTA Bus Debt Service", "Total CRR Debt Service",
      "Total NYCT Debt Service", "Total SIRTOA Debt Service",
      "Total MTA HQ Debt Service for 2 Broadway Certificates of Participation",
      "Total B&T Debt Service", "BAB Subsidy") ~ "Debt Service"))

## check the mta budget table
head(mta_budget)

## create a graph for mta budget totals each year
## exclude any data after 2019 and any expenses with a value of 0 or below
mta_budget %>% group_by(mta_year, expense_type) %>%
  filter(mta_year <= 2019 & expense_cost > 0) %>%
  summarize(total_cost = sum(expense_cost)) %>%
  ggplot(aes(mta_year, total_cost)) +
  geom_bar(stat = "identity", fill = "black") +
  scale_x_continuous(breaks = seq(2007, 2019, 1)) +
  xlab("Year") + ylab("Total Budget (millions)") +
  ggtitle("MTA Budget") +
  theme_bw()

## create a line graph for breakdown of mta budget
## exclude any data after 2019
mta_budget %>% group_by(mta_year, expense_type) %>%
  filter(mta_year <= 2019) %>%
  summarize(total_cost = sum(expense_cost)) %>%
  ggplot(aes(mta_year, total_cost, col = expense_type)) +
  geom_point() + geom_line() +
  scale_color_manual(values = c("#6600FF","#FF0099", "#0000FF", "#FF00FF"),
                     name = "Expense Type") +
  scale_x_continuous(breaks = seq(2007, 2019, 1)) +
  xlab("Year") + ylab("Total Expense Cost (millions)") +
  ggtitle("Breakdown of MTA Budget") +
  theme_bw()

## set up the amount of colors needed for a bar plot
nb.cols <- 13
```

```r
## assign the number of colors to a palette
mycolors <- colorRampPalette(brewer.pal(8, "BuPu"))(nb.cols)

## create a graph for labor expenses
## exclude any data after 2019
labor <- mta_budget %>%
  filter(expense_type == "Labor" & mta_year <= 2019) %>%
  mutate(mta_year = factor(mta_year)) %>%
  group_by(mta_year, expense_description) %>%
  summarize(total_expense_cost = sum(expense_cost)) %>%
  filter(total_expense_cost > 0) %>%
  mutate(expense_description = reorder(expense_description, -total_expense_cost)) %>%
  ggplot(aes(expense_description, total_expense_cost, fill = mta_year)) +
  geom_bar(position = "dodge", stat = "identity", colour = "black") +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  ylim(0, 6000) +
  scale_fill_manual(values = mycolors, name = "Year") +
  ylab("Cost of Expense (millions)") +
  ggtitle("MTA Labor Expenses") + theme_bw() +
  theme(axis.title.x = element_blank())

## create a graph for non-labor expenses
## exclude any data after 2019
non_labor <- mta_budget %>%
  filter(expense_type == "Non-Labor" & mta_year <= 2019) %>%
  mutate(mta_year = factor(mta_year)) %>%
  group_by(mta_year, expense_description) %>%
  summarize(total_expense_cost = sum(expense_cost)) %>%
  filter(total_expense_cost > 0) %>%
  mutate(expense_description = reorder(expense_description, -total_expense_cost)) %>%
  ggplot(aes(expense_description, total_expense_cost, fill = mta_year)) +
  geom_bar(position = "dodge", stat = "identity", colour = "black") +
  scale_x_discrete(labels = function(x) str_wrap(x, width = 10)) +
  ylim(0, 6000) +
  scale_fill_manual(values = mycolors, name = "Year") +
  ylab("Cost of Expense (millions)") +
  ggtitle("MTA Non-Labor Expenses") + theme_bw() +
  theme(axis.title.x = element_blank())

## arrange the graphs side by side
ggarrange(labor, non_labor, ncol = 2, nrow = 1,
          common.legend = TRUE, legend = "bottom")
```