

# mbta\_615final

Danya Zhang

2022-12-14

```
# subsetting data to the first week of 2022
data <- read.csv("/Users/dz/Documents/MSSP/GitHub/MA615 final/Final-Project-MBTA/TravelTimes_2022/2022-01-01-2022-01-07.csv")
df <- as.data.frame(table(data$service_date))
sum <- sum(df[c(1:7), 2])
data <- head(data, sum)
```

#Getting to know the data #Summary Statistics

```
# summary statistics
stats <- data %>%
  group_by(from_stop_id, to_stop_id) %>%
  summarise(max = max(travel_time_sec), min = min(travel_time_sec), mean = mean(travel_time_sec),
            sd = sd(travel_time_sec))
```

```
## `summarise()` has grouped output by 'from_stop_id'. You can override using the
## `.groups` argument.
```

```
head(stats)
```

```
## # A tibble: 6 x 6
## # Groups:   from_stop_id [1]
##   from_stop_id to_stop_id max    min mean    sd
##         <int>    <int> <int> <int> <dbl> <dbl>
## 1         70110        70112    610     12  104.  45.2
## 2         70110        70114   1111     2  183.  66.3
## 3         70110        70116   1189    80  253.  71.4
## 4         70110        70120   1316   176  365.  80.9
## 5         70110        70124   1642   269  489. 100.
## 6         70110        70126   1767   324  597. 113.
```

#Data Cleaning

```
# there some travel times that are clearly impossible. possible data
# entry error
short_ind <- data[which(data$travel_time_sec < 10), ]
remove_ind <- as.numeric(dimnames(short_ind)[[1]])
data <- data[-remove_ind, ]

# subsetting based on the same from_stop_id, to_stop_id, route_id,
# direction_id
subset1 <- data %>%
  filter(from_stop_id == 70134 & to_stop_id == 170136 & route_id == "Green-B" &
         direction_id == 1)

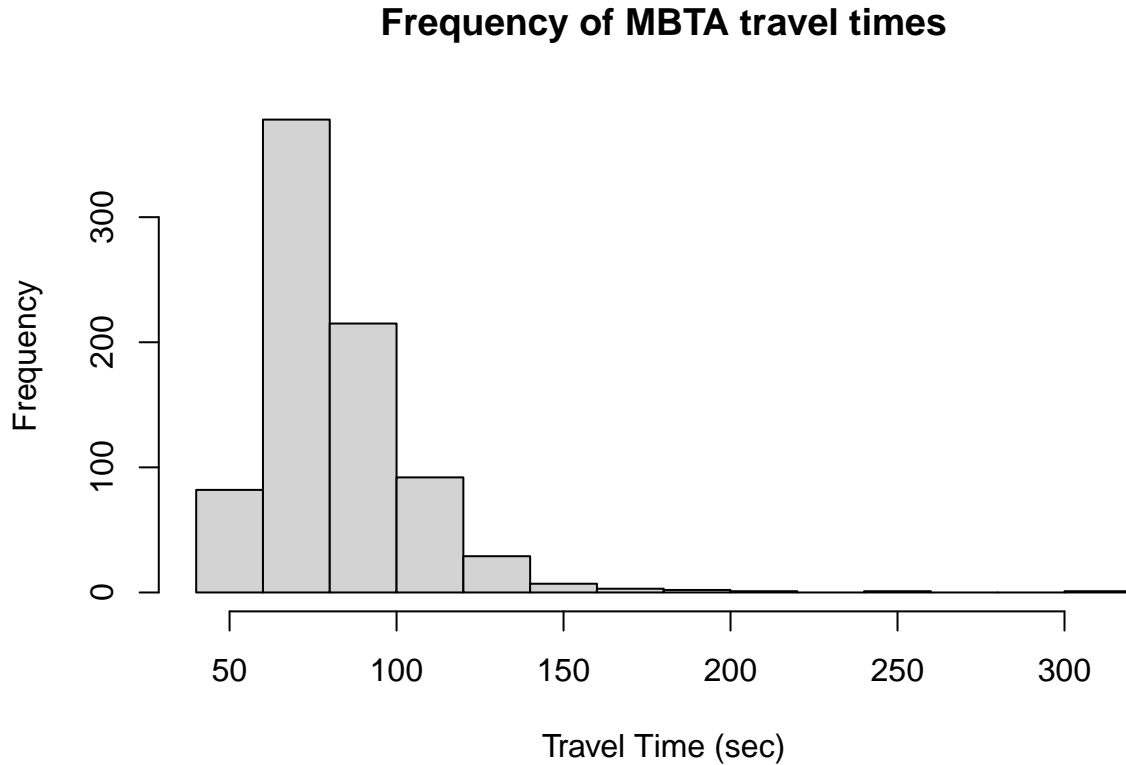
# first week of 2022
```

```
subset1 <- as.data.frame(subset1)
```

```
#Visualizations #Histogram
```

```
# histogram
```

```
hist(subset1$travel_time_sec, xlab = "Travel Time (sec)", main = "Frequency of MBTA travel times")
```



```
#Boxplot
```

```
library(ggplot2)
```

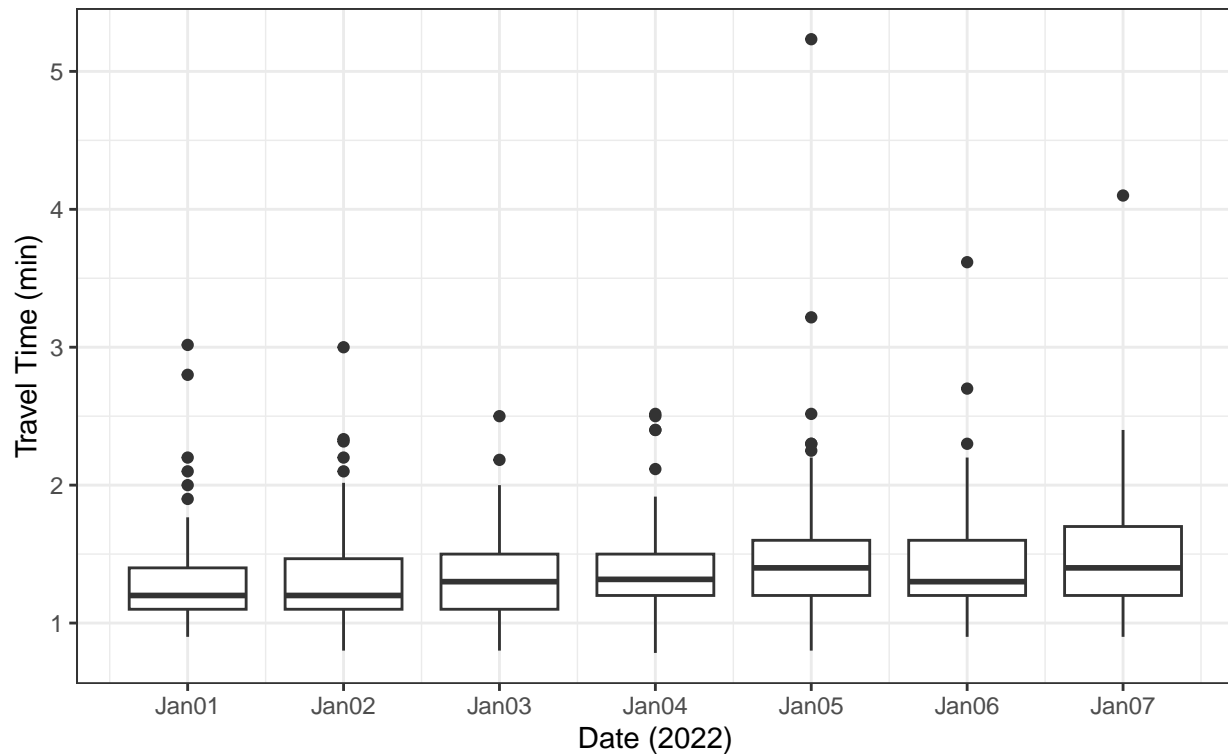
```
# boxplot
```

```
subset1 %>%
```

```
  ggplot(aes(x = as.Date(service_date), y = travel_time_sec/60, group = service_date)) +  
  geom_boxplot() + labs(title = "Boxplot of travel times", x = "Date (2022)",  
    y = "Travel Time (min)", subtitle = "stop 70134 to stop 170136, greenline-B, direction_id 1") +  
  scale_x_date(date_breaks = "1 day", date_label = "%b%d") + theme_bw()
```

## Boxplot of travel times

stop 70134 to stop 170136, greenline-B, direction\_id 1



*# looks pretty consistent. There are outliers but this could be  
# amounted to vehicle malfunctions, accidents, emergencies, unruly  
# passengers, etc.*

## Scatter Plot

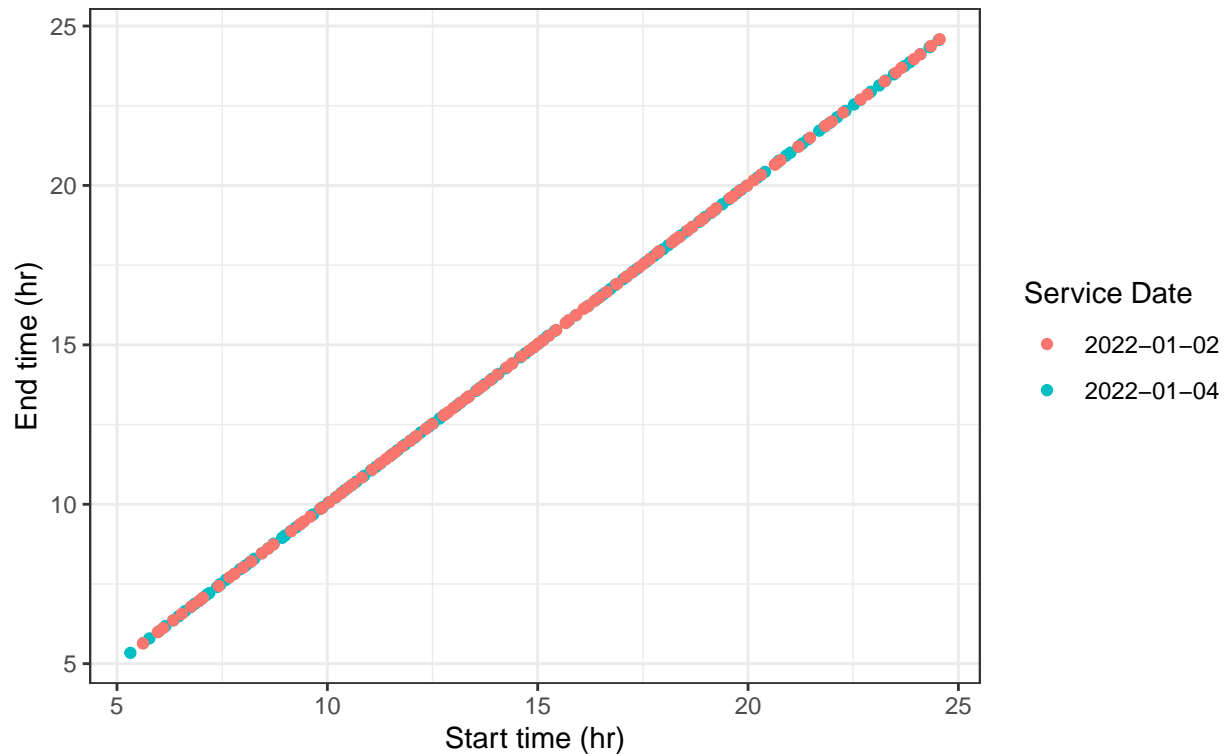
To see if there is variability between start time and end time based on day of the week, I have selected Sunday and Tuesday to represent the weekend and weekdays.

```
subset1_tues <- subset1[which(subset1$service_date == "2022-01-04"), ]
subset1_sun <- subset1[which(subset1$service_date == "2022-01-02"), ]
subset1_suntues <- rbind(subset1_tues, subset1_sun)

ggplot(subset1_suntues, aes(start_time_sec/60/60, end_time_sec/60/60, group = service_date,
  color = service_date)) + geom_point() + labs(title = "End Time vs. Start Time",
  x = "Start time (hr)", y = "End time (hr)", subtitle = "stop 70134 to stop 170136, greenline-B, dir",
  guides(color = guide_legend(title = "Service Date"))) + theme_bw()
```

## End Time vs. Start Time

stop 70134 to stop 170136, greenline-B, direction\_id 1



Scatter plots appear linear and stacked on top of each other. Seems like there is little to no difference, meaning the MBTA is fairly reliable and times between days are consistent.

```
# library(mapsapi) key <- 'AIzaSyBb1P6Czwie4KeTOBAKlHLoCzzUXRh9qyA'
# #580 to prudential doc <- mp_directions(origin = 'Boston
# University', destination = 'Newbury Street', alternatives = TRUE,
# key = key, quiet = TRUE) r <- mp_get_routes(doc) #map visualization
# library(leaflet) pal <- colorFactor(palette = 'Dark2', domain =
# r$alternative_id) leaflet() %>%
# addProviderTiles('CartoDB.DarkMatter') %>% addPolylines(data = r,
# opacity = 1, weight = 7, color = ~pal(alternative_id))
```