MA615_final_report

Yirong Yuan

2022-12-14

<Introduction>

This report analyzes the operation of MBTA's subway, bus, and ferry. To limit the project size, I picked a week randomly from the past 12 months as my dataset. For data on the subway, the first month is Oct. 2021 and the last month is Sept. 2022. It is the same as the data of the bus. For the ferry, the first month is Jan. 2021, and the last month is Dec.2021.

```
library(tidyverse)
library(ggplot2)
library(leaflet)
library(rgdal)
library(sf)
library(magrittr)
library(viridis)
library(hrbrthemes)

##import subway data
HR_selected<-read.csv("~/Desktop/HR_selected.csv")
LR_selected<-read.csv("~/Desktop/LR_selected.csv")

##import data for mapping
MBTA_icon <- makeIcon(
  iconUrl = "https://upload.wikimedia.org/wikipedia/commons/thumb/6/64/MBTA.svg/240px-MBTA.svg.png",iconWidth = 1
5, iconHeight = 15)
MBTA_sub <- readOGR(dsn="~/desktop/mbta_rapid_transit", layer="MBTA_ARC")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/yirong/Desktop/mbta_rapid_transit", layer: "MBTA_ARC"
## with 141 features
## It has 4 fields
```

```
MBTA_sub <- spTransform(MBTA_sub, CRS("+proj=longlat +ellps=GRS80"))
MBTA_sub_stops <- readOGR(dsn="~/desktop/mbta_rapid_transit", layer="MBTA_NODE")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/yirong/Desktop/mbta_rapid_transit", layer: "MBTA_NODE"
## with 170 features
## It has 4 fields
```

```
MBTA_sub_stops <- spTransform(MBTA_sub_stops, CRS("+proj=longlat +ellps=GRS80"))

MBTA_bus <- readOGR(dsn="~/desktop/mbtabus", layer="MBTABUSROUTES_ARC")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/yirong/Desktop/mbtabus", layer: "MBTABUSROUTES_ARC"
## with 923 features
## It has 11 fields
## Integer64 fields read as strings:  CTPS_ROU_2
```

```
MBTA_bus <- spTransform(MBTA_bus, CRS("+proj=longlat +ellps=GRS80"))
MBTA_bus_stops <- readOGR(dsn="~/desktop/mbtabus", layer="MBTABUSSTOPS_PT")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/yirong/Desktop/mbtabus", layer: "MBTABUSSTOPS_PT"
## with 7810 features
## It has 4 fields
## Integer64 fields read as strings:  STOP_ID
```

```
MBTA_bus_stops <- spTransform(MBTA_bus_stops, CRS("+proj=longlat +ellps=GRS80"))

MBTA_ferry <- readOGR(dsn="~/desktop/Ferry_Routes", layer="Ferry_Routes")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/yirong/Desktop/Ferry_Routes", layer: "Ferry_Routes"
## with 41 features
## It has 14 fields
```

```
MBTA_ferry <- spTransform(MBTA_ferry, CRS("+proj=longlat +ellps=GRS80"))
MBTA_ferry_stops <- readOGR(dsn="~/desktop/Seaports", layer="Seaports")
```

```
## OGR data source with driver: ESRI Shapefile
## Source: "/Users/yirong/Desktop/Seaports", layer: "Seaports"
## with 30 features
## It has 14 fields
```

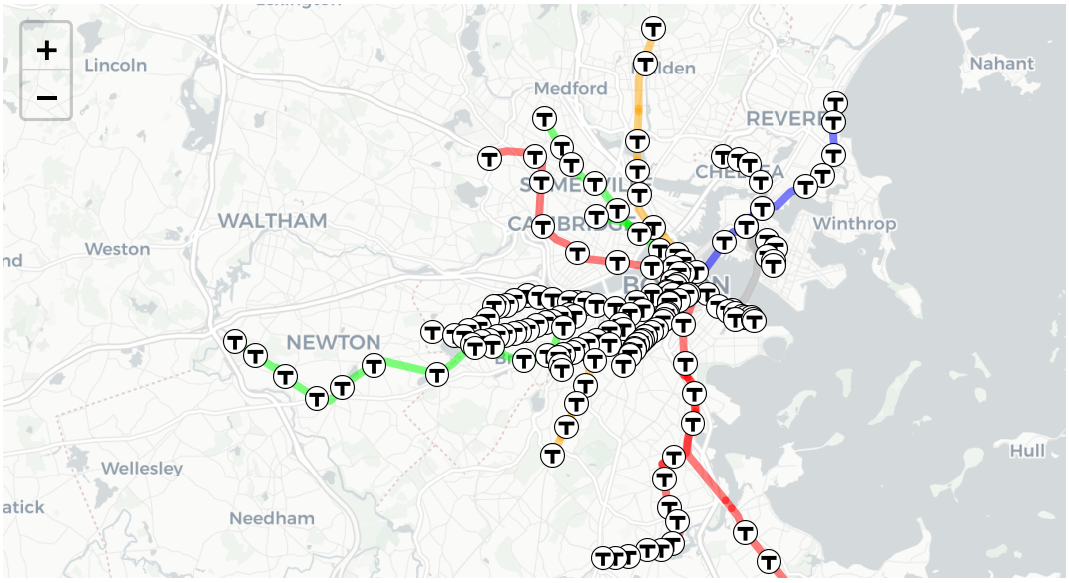
```
MBTA_ferry_stops <- spTransform(MBTA_ferry_stops, CRS("+proj=longlat +ellps=GRS80"))

##import bus data
BUS_selected<-read.csv("~/Desktop/BUS_selected.csv")
##import ferry data
Ferry_selected<-read.csv("~/Desktop/Ferry_selected.csv")
```

<Routes and Stops>

I used three leaflet plots to display the routes and stops of the MBTA subway, bus, and ferry.

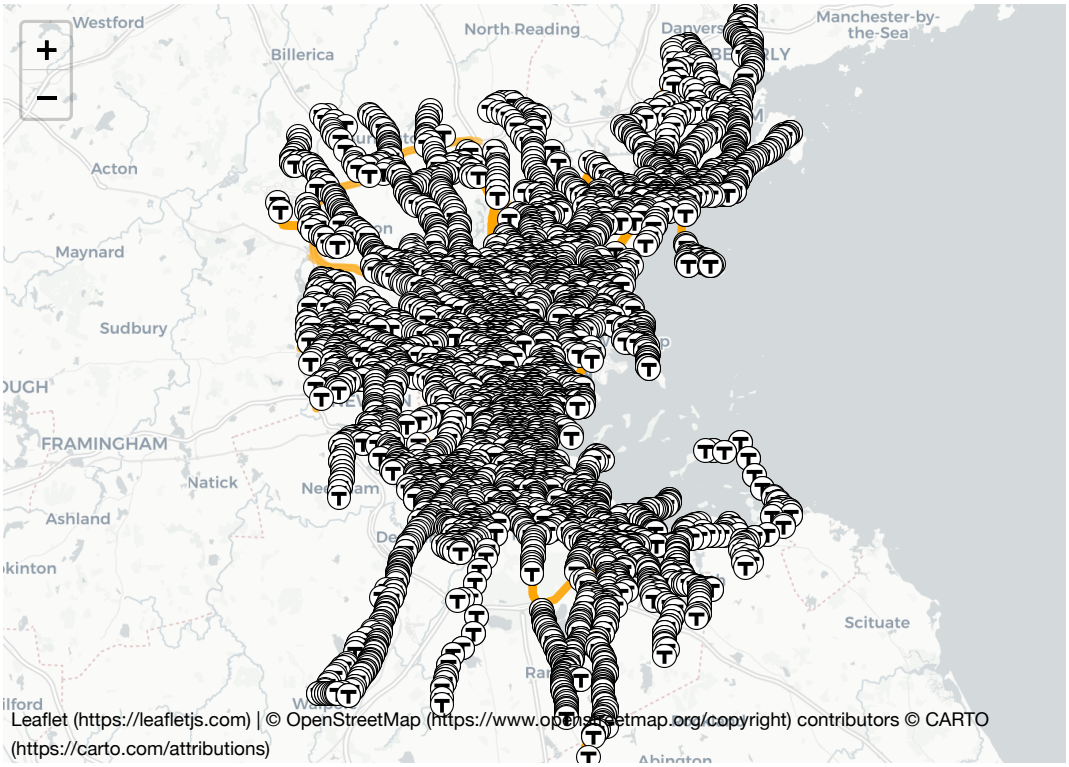
MBTA Subway routes and stops

```
pal <- colorFactor(palette = c('blue','green','orange','red','grey'),
                   domain = c('BLUE','GREEN','ORANGE','RED','SILVER'))
leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolylines(data = MBTA_sub, color=~pal(LINE)) %>%
  addMarkers(data=MBTA_sub_stops,icon=MBTA_icon)
```



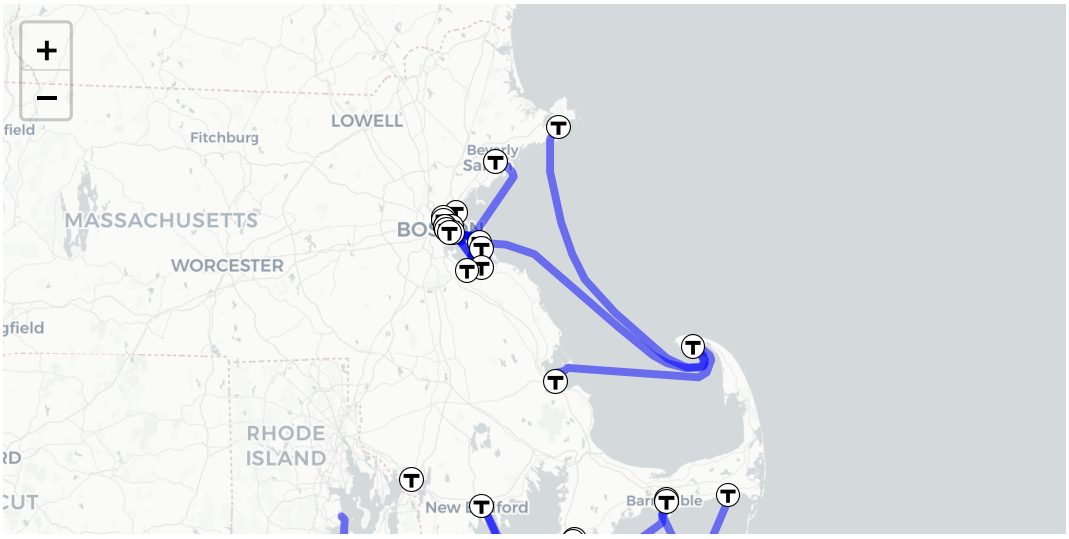
MBTA Bus routes and stops

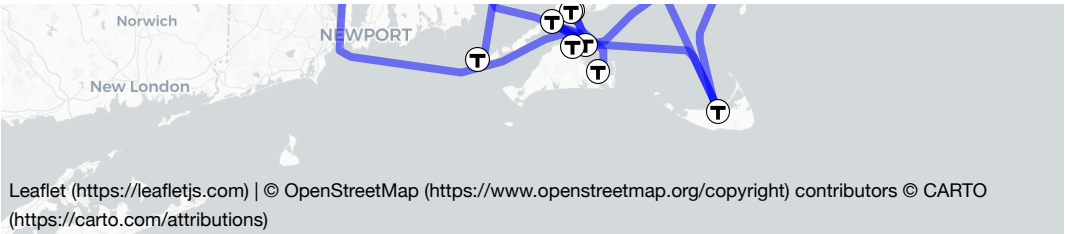
```
leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolylines(data = MBTA_bus, color="orange") %>%
  addMarkers(data=MBTA_bus_stops,icon=MBTA_icon)
```



MBTA Ferry routes and stops

```
leaflet() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addPolylines(data = MBTA_ferry, color="blue")%>%
  addMarkers(data=MBTA_ferry_stops,icon=MBTA_icon)
```





<Data visualization>

I re-organized data and visually shows the operation conditions of subway, bus, and ferry.

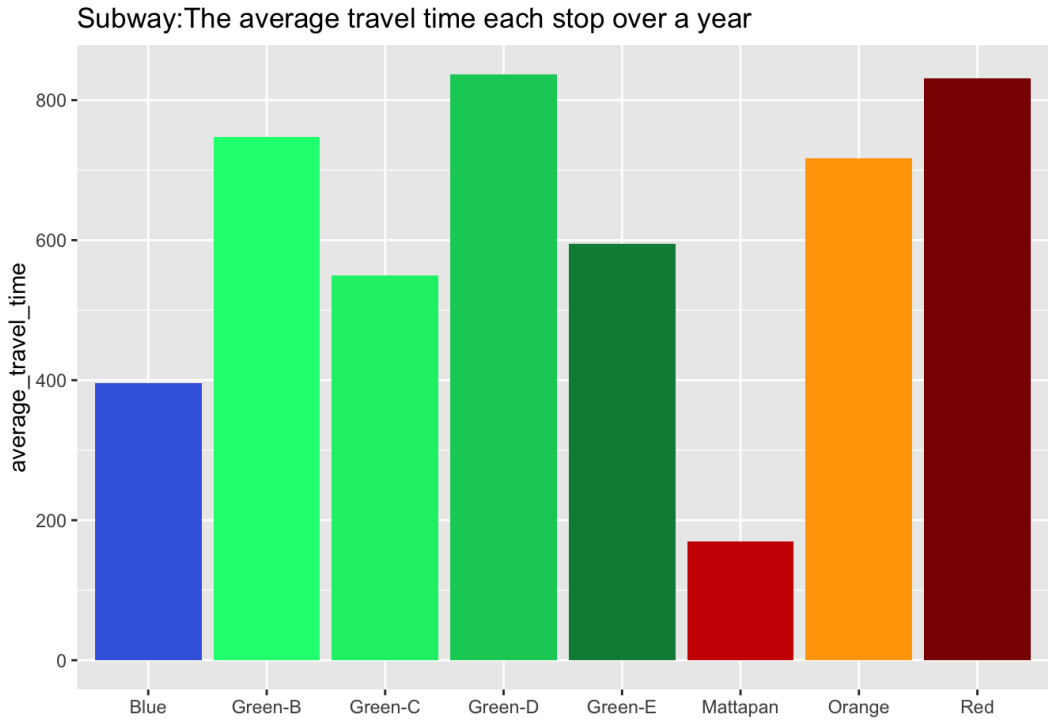
Subway data organize

```
##Calculate the average travel time each stop over a year for HR
HR_year<- HR_selected %>% select("from_stop_id","to_stop_id","route_id","travel_time_sec","service_date","week")%
>%group_by(route_id) %>% summarise(count = median(travel_time_sec))

##Calculate the average travel time per stop per week for a year for HR
HR_week<- HR_selected %>% select("from_stop_id","to_stop_id","route_id","travel_time_sec","service_date","week")%
>%group_by(route_id,week) %>% summarise(count = median(travel_time_sec))

#Calculate the average travel time each stop over a year for LR
LR_year<- LR_selected %>% select("from_stop_id","to_stop_id","route_id","travel_time_sec","service_date","week")%
>%group_by(route_id) %>% summarise(count = median(travel_time_sec))

##Calculate the average travel time per stop each day of a week for a year for LR
LR_week<- LR_selected %>% select("from_stop_id","to_stop_id","route_id","travel_time_sec","service_date","week")%
>%group_by(route_id,week) %>% summarise(count = median(travel_time_sec))

##Combine HR and LR
Subway_year <-bind_rows(HR_year,LR_year)
Subway_week <-bind_rows(HR_week,LR_week)

##Rename variable
Subway_year %<>%rename(average_travel_time = count)
Subway_week %<>%rename(average_travel_time = count)
```

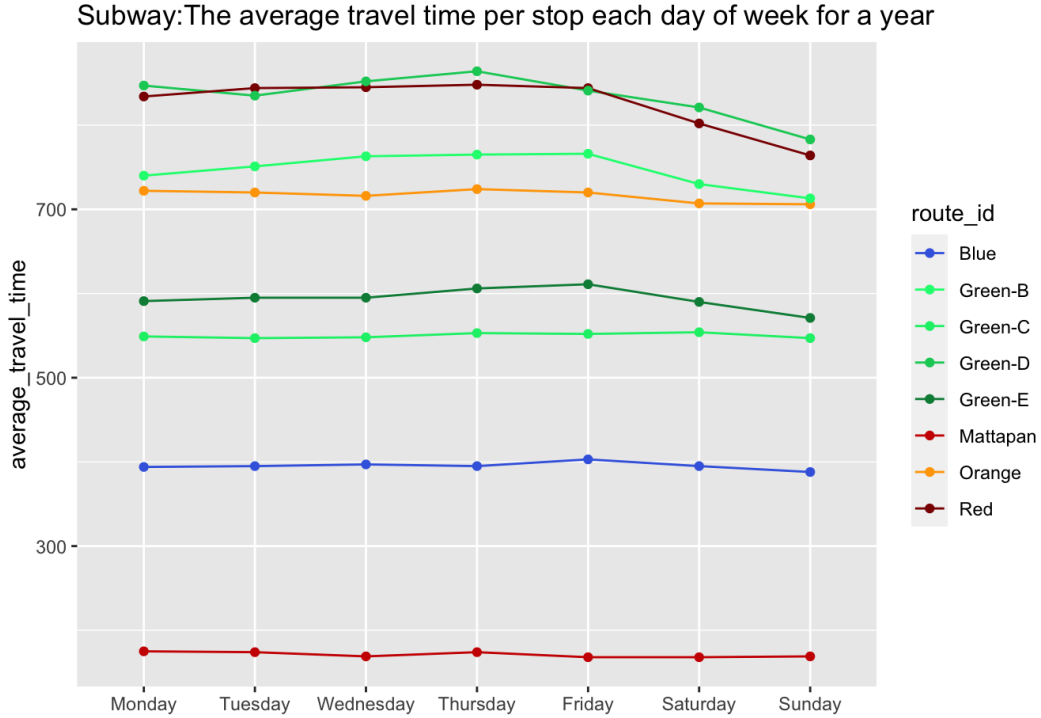
<Subway Data visualization>

The first plot displays different subway lines that have different average travel times between each stop. The Mattapan line has the lowest average travel time between each stop. Red line and Green-D have the highest average travel time between each stop. The second plot shows the average travel times of each line always peak on Thursday and Friday.

```
colors<-c("royalblue","springgreen","springgreen2","springgreen3","springgreen4","red3","orange","red4")
 ggplot(Subway_year, aes(x=route_id, y=average_travel_time,fill=route_id)) +
    geom_bar(stat="identity") +
   scale_fill_manual(values =colors )+
   ggtitle("Subway:The average travel time each stop over a year")+
   xlab("")+
   theme(legend.position="none")
```




```
week_order<- c("Monday", "Tuesday", "Wednesday","Thursday","Friday","Saturday",
                                    "Sunday")

p1<-ggplot(Subway_week, aes(x=week, y=average_travel_time, group=route_id)) +
  geom_line(aes(color=route_id))+
  geom_point(aes(color=route_id))
p2<-p1+ scale_x_discrete(limits = week_order)
p2+ ggtitle("Subway:The average travel time per stop each day of week for a year")+
  scale_color_manual(values=colors)+
   xlab("")
```



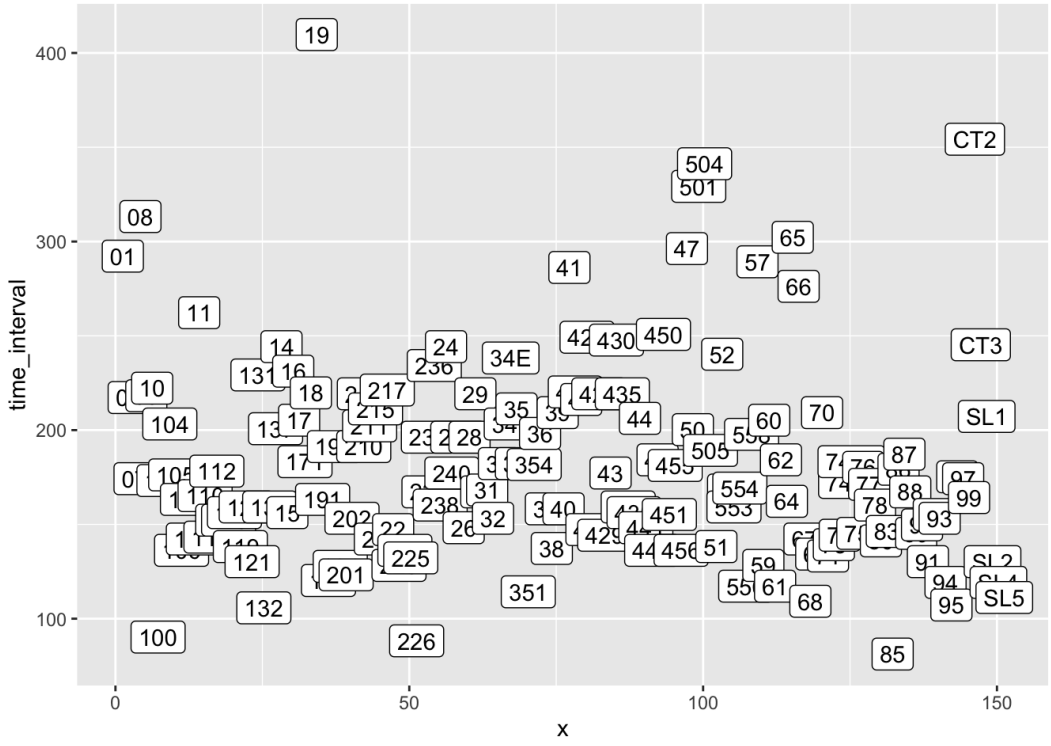
##Bus data organize

```
##filter empty rows in data
BUS_sub<-BUS_selected%>%filter(BUS_selected$actual!="")

##Calculate the difference between scheduled time and actual time
BUS_sub$actual <- as.POSIXct(BUS_sub$actual)
BUS_sub$scheduled <- as.POSIXct(BUS_sub$scheduled)
BUS_sub$time_interval<-difftime(BUS_sub$actual, BUS_sub$scheduled, units="secs")
BUS_sub %<>% separate(col=time_interval,into =c("time_interval","unit"),sep=" ")
BUS_sub %<>%select(-unit)

##Change time_interval to absolute value
BUS_sub$time_interval<-abs(as.numeric(BUS_sub$time_interval))

#Calculate the average of the difference between the estimated and actual times
##of the bus for a year
BUS_year<- BUS_sub %>% select(service_date,route_id,time_interval)%>%
  group_by(route_id)%>%
  summarise(time_interval = median(time_interval))

##Calculates the average of the difference between the estimated and actual
##times for each day of a week for the bus
BUS_week<- BUS_sub %>% select(week,route_id,time_interval)%>%
   group_by(week)%>%
  summarise(time_interval = median(time_interval))
```

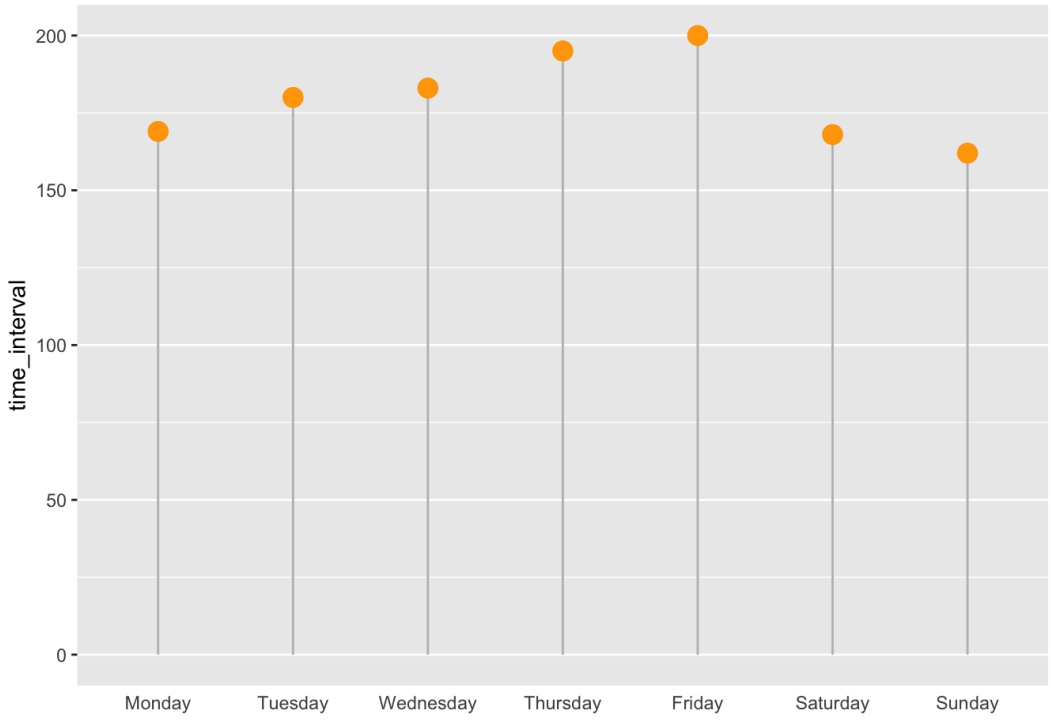
<Bus data visualization>

The first plot shows the average difference between the estimated and actual times of the bus for a year. We can see that BUS 19 has the highest average time difference between the estimated and actual and BUS 85 has the lowest average time difference between the estimated and actual. The second plot shows the average difference between the estimated and actual times for each day of a week for the bus. On Friday, the buses always have the highest difference time between the estimated and actual.

```
BUS_year$x<-row_number(BUS_year$route_id)
ggplot(BUS_year, aes(x=x, y=time_interval)) +
  geom_point() +
  geom_label(
    label=BUS_year$route_id,
    nudge_x = 0.25, nudge_y = 0.25, )
```



```
week_order<- c("Monday", "Tuesday", "Wednesday","Thursday","Friday","Saturday",
                                        "Sunday")
ggplot(BUS_week, aes(x=week, y=time_interval)) +
  geom_segment( aes(x=week, xend=week, y=0, yend=time_interval), color="grey") +
  geom_point( color="orange", size=4) +
  theme(
    panel.grid.major.x = element_blank(),
    panel.border = element_blank(),
    axis.ticks.x = element_blank()
  ) +
  scale_x_discrete(limits = week_order)+
  xlab("") +
  ylab("time_interval")
```



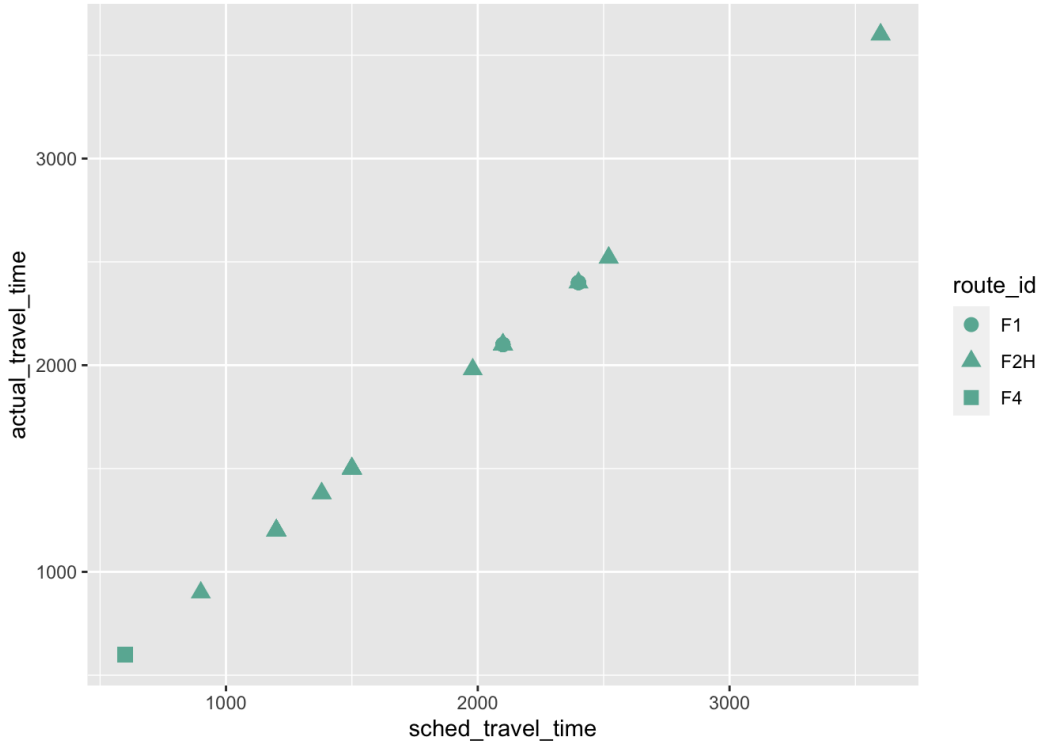
##Ferry data organize

```
##Calculate mbta scheduled travel time
Ferry_selected$mbta_sched_departure <- as.POSIXct(Ferry_selected$mbta_sched_departure)
Ferry_selected$mbta_sched_arrival<-as.POSIXct(Ferry_selected$mbta_sched_arrival)
Ferry_selected$sched_travel_time<-difftime(Ferry_selected$mbta_sched_arrival,
                                           Ferry_selected$mbta_sched_departure ,units="secs")
Ferry_selected$sched_travel_time<-as.numeric(Ferry_selected$sched_travel_time)

##Calculate actual travel time
Ferry_selected$actual_departure <- as.POSIXct(Ferry_selected$actual_departure)
Ferry_selected$actual_arrival<-as.POSIXct(Ferry_selected$actual_arrival)
Ferry_selected$actual_travel_time<-difftime(Ferry_selected$mbta_sched_arrival,
                                           Ferry_selected$mbta_sched_departure ,units="secs")
Ferry_selected$actual_travel_time<-as.numeric(Ferry_selected$actual_travel_time)

##Calculate the absolute value of time difference between scheduled arrive and actual arrive
Ferry_selected$arrival_interval<-difftime(Ferry_selected$actual_arrival,
                                           Ferry_selected$mbta_sched_arrival ,units="secs")
Ferry_selected$arrival_interval<-abs(as.numeric(Ferry_selected$arrival_interval))
##subset data
Ferry_sub1<-Ferry_selected%>%select(route_id,sched_travel_time,actual_travel_time)
Ferry_sub2<-Ferry_selected%>%select(route_id,arrival_interval)
```

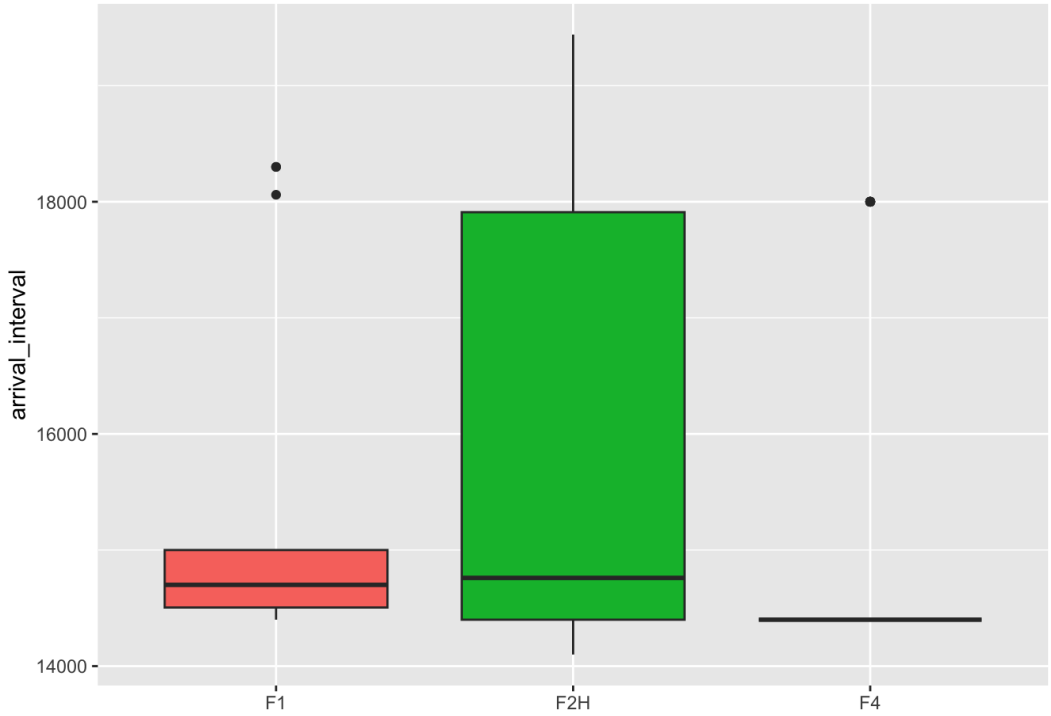
<Ferry data visualization>

The first plot shows that the actual travel time and scheduled travel time are very close because the slope of these scatter points is close to 45 degrees. The Second plot shows that the F2H ferry route has the largest range of time difference between scheduled and actual arrival times.

```
ggplot(Ferry_sub1, aes(x=sched_travel_time, y=actual_travel_time, shape=route_id)) +
    geom_point(size=3,color="#69b3a2")
```




```
ggplot(Ferry_sub2, aes(x=route_id, y=arrival_interval, fill=route_id)) +
  geom_boxplot() +
  xlab("route_id") +
  theme(legend.position="none") +
  xlab("") +
  xlab("")
```



<Conculsion>

According to the results of EDA, we can see that the travel time between stops of each line varies greatly and is different in a week. The EDA of the MBTA bus and ferry shows that the actual arrival time has different from the scheduled arrival time. The MBTA transportations always arrive earlier or later than scheduled.