615 Final Project–MBTA EDA

Yaquan Yang

2022-12-12

Subway data

This analysis uses data from November 2021 to October 2022 for the Red line, Blue line, and Green-B line to analyze.

- Red Line: From Alewife in the northwest to Braintree in the southeast.
- Blue Line: From Wonderland to Bowdoin via Airport and Downtown.
- Green-B: From Government Center to Boston College, passes through Boston University and Fenway Park.

These three subway lines contain several important transportation hubs in the Boston area that we believe are worth analyzing.

```
data.files<-list.files('MetroData',pattern='csv$',full=T)
data.files
```

```
## [1] "MetroData/2022-Q1_HRTravelTimes.csv" "MetroData/2022-Q1_LRTrave
lTimes.csv"
## [3] "MetroData/2022-Q2_HRTravelTimes.csv" "MetroData/2022-Q2_LRTrave
lTimes.csv"
## [5] "MetroData/2022-Q3_HRTravelTimes.csv" "MetroData/2022-Q3_LRTrave
lTimes.csv"
## [7] "MetroData/HRTravelTimesQ4_21.csv"    "MetroData/LRTravelTimesQ4
_21.csv"
```

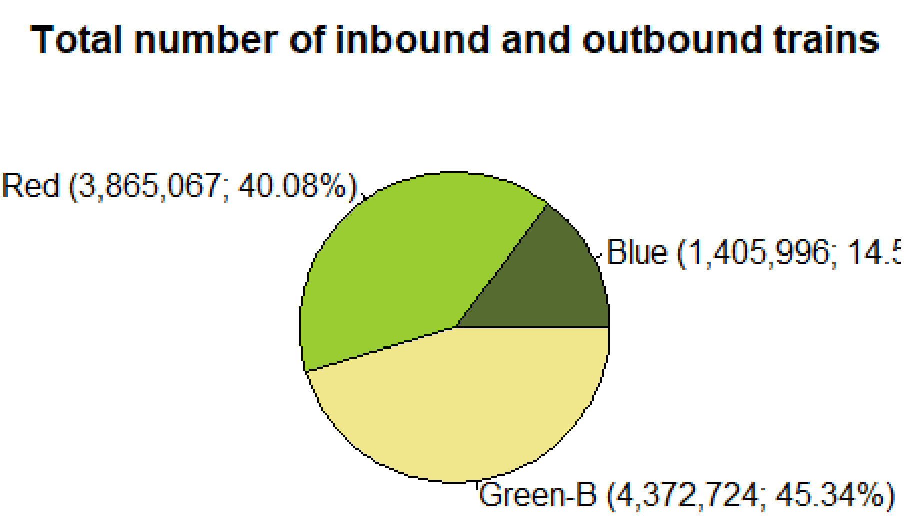
```
dat<-lapply(setNames(,data.files),function(x)
{
   fread(x)->inter
   inter[,fromFile:=x]
   inter[,day:=mday(service_date)]
   inter<-inter[day %in% 1:7]
   inter<-inter[route_id %in% c('Red','Blue','Green-B')]
})


rbindlist(dat)->dat
dat[,startTime:=as.ITime(service_date)+start_time_sec]
dat[,endTime:=as.ITime(service_date)+end_time_sec]
```

```
fread('stops.txt')->sites
setNames(sites[,stop_name],sites[,stop_id])->sites.name

dat[,from_stop_name:=sites.name[as.character(from_stop_id)]]
dat[,to_stop_name:=sites.name[as.character(to_stop_id)]]
```

Subway EDA

Total number of inbound and outbound trains for different lines (Red, Blue, Green-B) during the year

```
pie.data<-dat[,.N,route_id]
pie.data[,percent:=percent(N/sum(N),0.01)]
pie.data[,label:=paste0(route_id," (",comma(N),"; ",percent,")")]


pie(pie.data$N,
    pie.data$label,
    col=c('darkolivegreen','olivedrab3','khaki'), main= "Total number o
f inbound and outbound trains")
```



From the above graph we can see that the traffic flow on the red and green-B lines is much higher than the blue line, which means that the red and green-B lines are busier than the blue line most of the time.

Total number of inbound and outbound trains at different stations

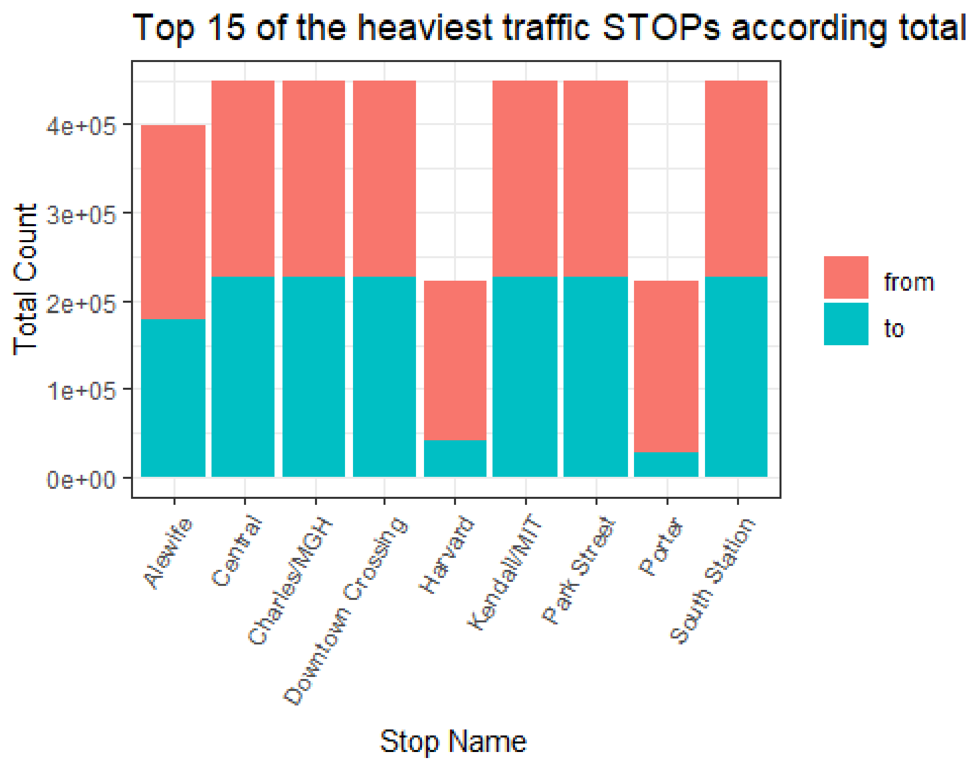
```
fromVolume<-dat[,.N,from_stop_id][order(-N)]
names(fromVolume)<-c('stop_id','from_count')
toVolume<-dat[,.N,to_stop_id][order(-N)]
names(toVolume)<-c('stop_id','to_count')
siteVolume<-fromVolume[toVolume,on=.(stop_id)]
siteVolume[,stop_name:=sites.name[as.character(stop_id)]]
siteVolume[,total_count:=from_count+to_count]
siteVolume<-siteVolume[order(-total_count)]
siteVolume<-siteVolume[,.(stop_id,stop_name,from_count,to_count,total_c
ount)]

siteVolume %>%
head(15) %>%
    flextable() %>%
    width(j=2,width=2) %>%
    theme_vanilla() %>%
    set_caption('Top 15 of the heaviest traffic STOPs according total c
ount')
```

Top 15 of the heaviest traffic STOPs according total count

stop_id	stop_name	from_count	to_count	total_count
70,061	Alewife	220,236	178,925	399,161
70,080	South Station	123,024	101,909	224,933
70,078	Downtown Crossing	108,928	115,971	224,899
70,069	Central	169,433	55,404	224,837
70,071	Kendall/MIT	155,351	69,420	224,771
70,075	Park Street	127,255	97,240	224,495
70,077	Downtown Crossing	113,410	110,981	224,391
70,076	Park Street	94,728	129,655	224,383
70,073	Charles/MGH	141,136	83,238	224,374

stop_id	stop_name	from_count	to_count	total_count
70,072	Kendall/MIT	66,781	157,516	224,297
70,070	Central	52,694	171,556	224,250
70,074	Charles/MGH	80,726	143,362	224,088
70,079	South Station	99,265	124,774	224,039
70,067	Harvard	180,621	41,562	222,183
70,065	Porter	194,501	27,667	222,168

```
site.plot.data<-siteVolume %>% head(15)
site.plot.data[,stop_name:=ordered(stop_name)]
site.plot.data<-site.plot.data[,.(stop_name,from_count,to_count)]
site.plot.data<-melt(site.plot.data,id.var='stop_name')
ggplot(site.plot.data,aes(x=stop_name,y=value,fill=gsub('_count','',var
iable)))+
geom_bar(stat='identity')+
theme_bw()+
guides(x=guide_axis(angle=60))+
labs(x='Stop Name',
     y='Total Count',fill='',title = "Top 15 of the heaviest traffic ST
OPs according total count")
```



We can see that the busiest stops are "Alewife", "South Station" and "Downtown Crossing" Distribution of subway traffic and time

Distribution of subway traffic within 24 hours

```
dat[,startHour:=hour(startTime)]
hour.data<-dat[,.(count=.N),startHour][order(startHour)]
hour.data<-hour.data[data.table(startHour=0:23),on=.(startHour)]

flextable(hour.data) %>%
    width(j=2,width=2) %>%
    theme_vanilla() %>%
    set_caption('Distribution of subway traffic within 24-hours')
```

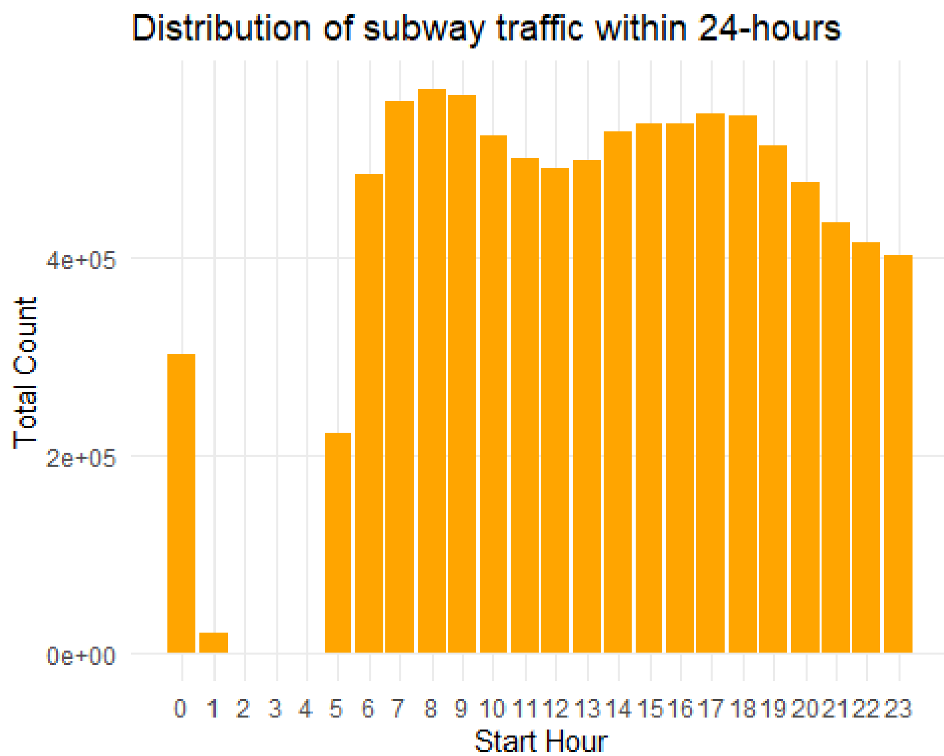
Distribution of subway traffic within 24-hours

startHour	count
0	302,318
1	21,408

startHour	count
2	
3	
4	1,028
5	222,061
6	484,134
7	555,949
8	568,769
9	563,181
10	522,785
11	499,405
12	489,238
13	497,052
14	525,445
15	534,193
16	534,980
17	543,641
18	542,866
19	511,315
20	474,652
21	434,494
22	412,942
23	401,931

```
ggplot(hour.data,aes(x=startHour,y=count))+
    geom_bar(stat='identity',fill='orange')+
    theme_minimal()+
    scale_x_continuous(breaks=0:23)+
    theme(panel.grid.minor=element_blank())+
```

```
    labs(x='Start Hour',
     y='Total Count',fill='',title = "Distribution of subway traffic wi
thin 24-hours")
```



The morning peak of the subway is at 7:00-9:00 a.m., the evening peak is at 4:00-7:00 p.m., and almost all trains stop running from 1:00 a.m. to 4:00 a.m.

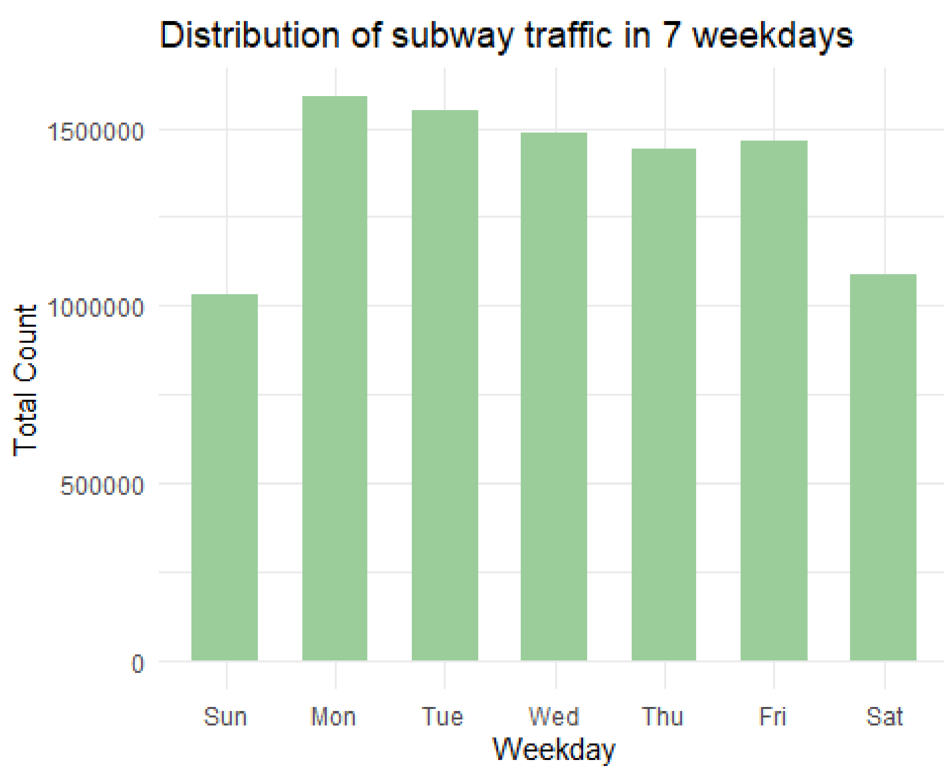
Weekday

```
weekdays.name<-strsplit('Sun,Mon,Tue,Wed,Thu,Fri,Sat',',')[[1]]

dat[,weekday:=wday(service_date)]
dat[,weekDay:=weekdays.name[as.integer(weekday)]]
dat[,weekDay:=ordered(weekDay,weekdays.name)]

week.data<-dat[,.(count=.N),weekDay][order(weekDay)]

flextable(week.data) %>%
    width(j=2,width=2) %>%
    theme_vanilla() %>%
    set_caption('Distribution of subway traffic in 7 weekdays')
```

Distribution of subway traffic in 7 weekdays

weekDay	count
Sun	1,029,821
Mon	1,588,406
Tue	1,547,402
Wed	1,486,562
Thu	1,443,433
Fri	1,462,784
Sat	1,085,379

```
ggplot(week.data,aes(y=count,x=weekDay))+
  geom_bar(stat='identity',fill='darkseagreen3',width=0.6)+
  theme_minimal()+
    labs(x='Weekday',
     y='Total Count',fill='',title = "Distribution of subway traffic in
 7 weekdays")
```



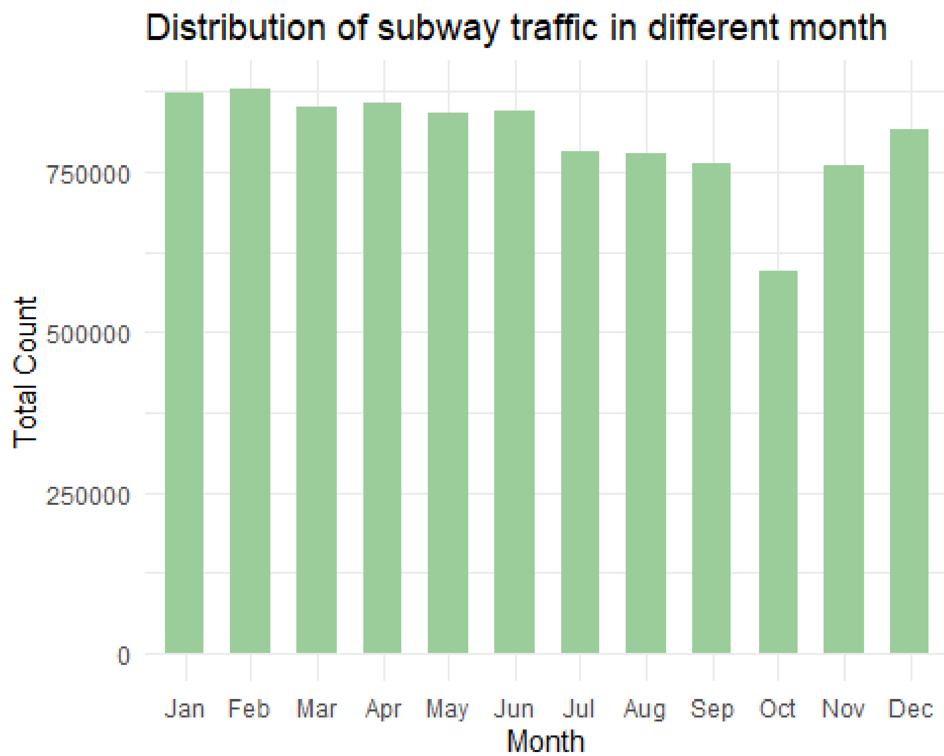
Month

```
dat[,Month:=month(service_date)]
dat[,MonthName:=month.abb[Month]]
dat[,MonthName:=ordered(MonthName,month.abb)]

month.data<-dat[,.(count=.N),MonthName][order(MonthName)]

flextable(month.data) %>%
    width(j=2,width=2) %>%
    theme_vanilla() %>%
    set_caption('Distribution of subway traffic in different month')
```

Distribution of subway traffic in different month

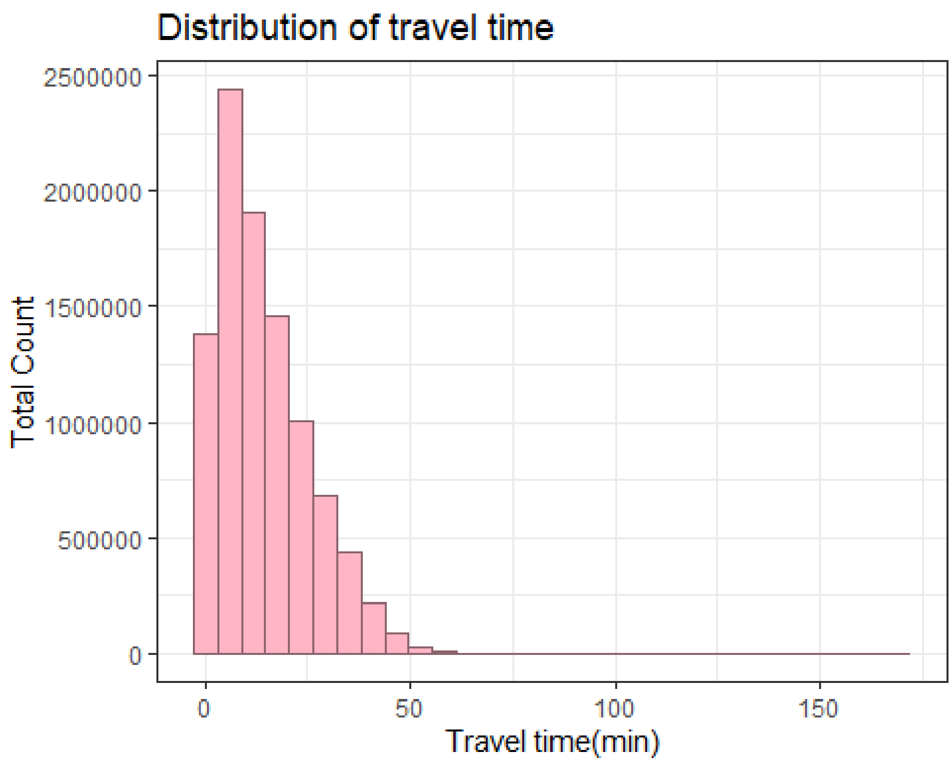
MonthName	count
Jan	873,378
Feb	879,392
Mar	850,672
Apr	857,848
May	842,091
Jun	844,162
Jul	781,057
Aug	779,973
Sep	763,749
Oct	595,694
Nov	760,787
Dec	814,984

```
ggplot(month.data,aes(y=count,x=MonthName))+
  geom_bar(stat='identity',fill='darkseagreen3',width=0.6)+
  theme_minimal()+
  labs(x='Month',
     y='Total Count',fill='',title = "Distribution of subway traffic in different month")
```



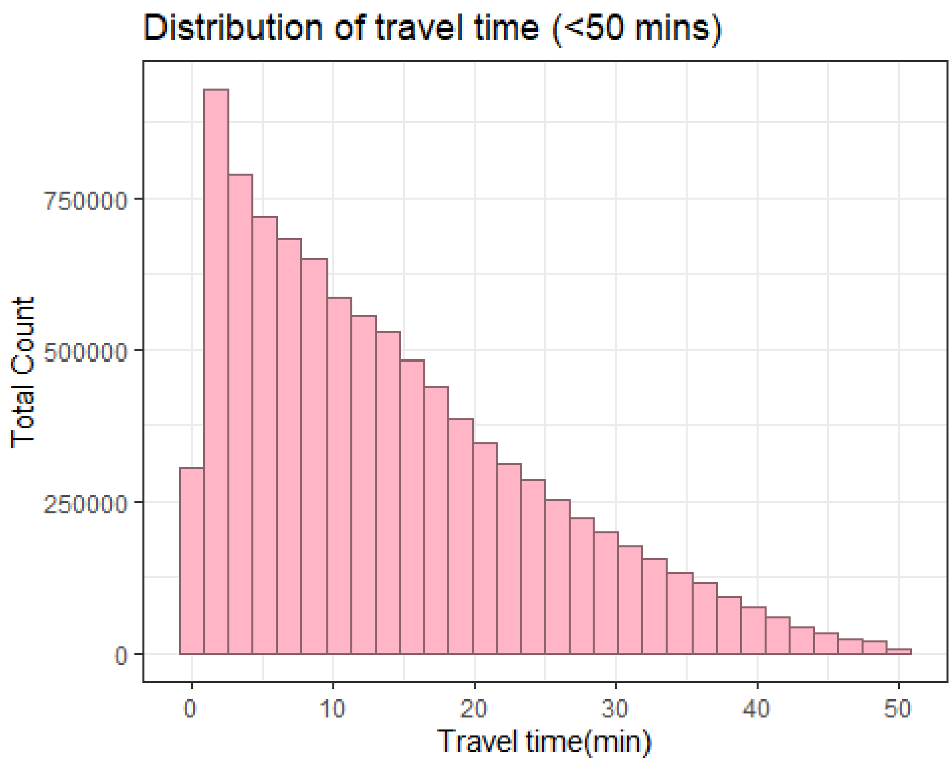
Distribution of travel_time

```
ggplot(dat,aes(travel_time_sec/60))+
geom_histogram(bins=30,fill='pink1',colour='pink4')+
theme_bw()+
    labs(x='Travel time(min)',
     y='Total Count',fill='',title = "Distribution of travel time")
```



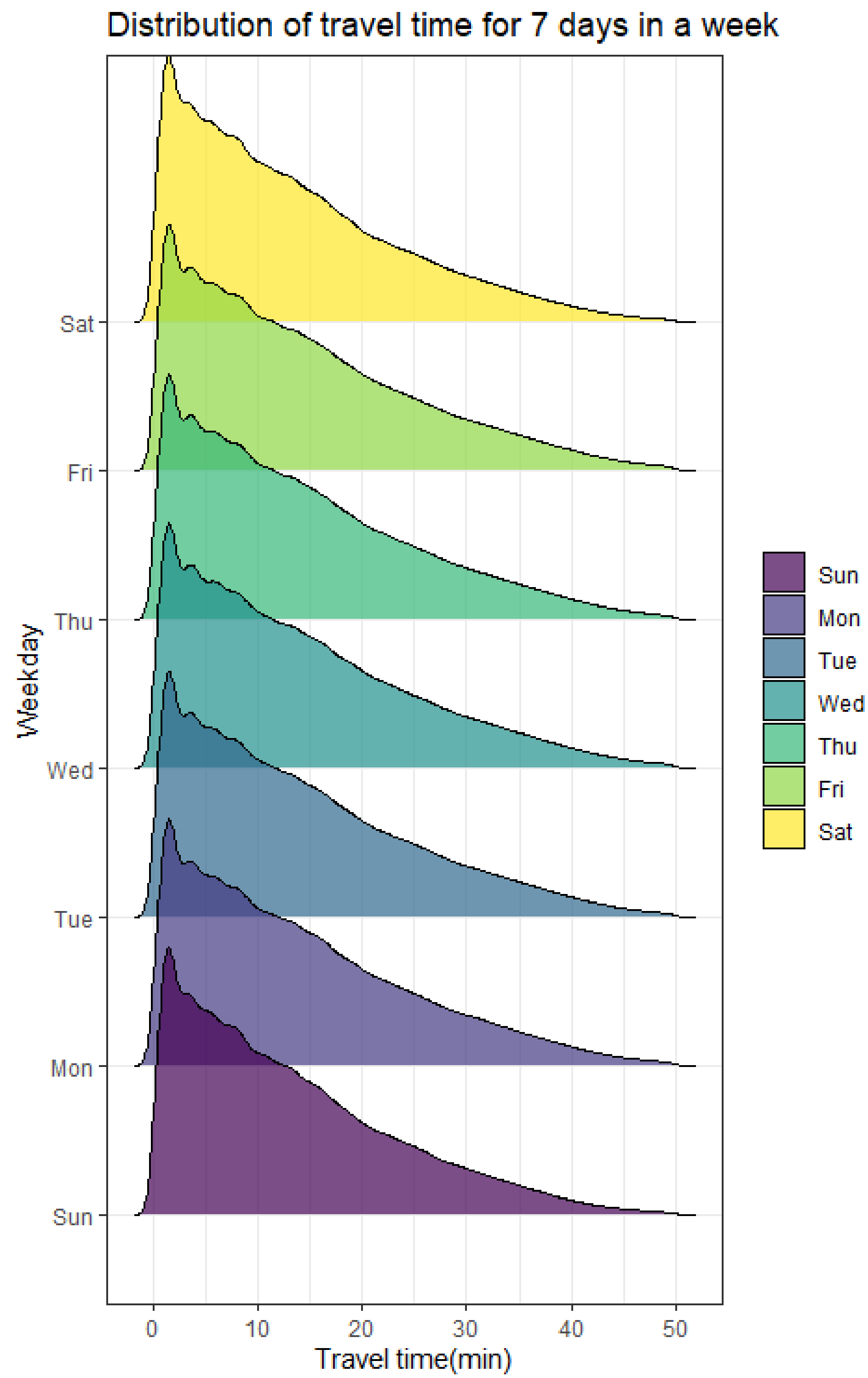
From the above chart, we can see that most of the travel_time are located within 50 minutes, let's look at the distribution of travel_time < 50 mins:

```
ggplot(dat[travel_time_sec<=3000],aes(travel_time_sec/60))+
geom_histogram(bins=30,fill='pink1',colour='pink4')+
theme_bw()+
        labs(x='Travel time(min)',
     y='Total Count',fill='',title = "Distribution of travel time (<50 
mins)")
```



Distribution of travel_time for 7 days in a week:

```
library(ggridges)

ggplot(dat[travel_time_sec<=3000],aes(x=travel_time_sec/60,y=weekDay,fill=weekDay))+
   geom_density_ridges(alpha=0.7)+
   theme_bw()+
    labs(x='Travel time(min)',
     y='Weekday',fill='',title = "Distribution of travel time for 7 days in a week")
```



The results show that the distribution is similar.

Bus Data

We choose three routes: No.8, No.56 and No.71 for analysis

```
(bus.data.files<-list.files('BusData',pattern='csv$',full=T))

##  [1] "BusData/MBTA-Bus-Arrival-Departure-Times_2021-10.csv"
##  [2] "BusData/MBTA-Bus-Arrival-Departure-Times_2021-11.csv"
##  [3] "BusData/MBTA-Bus-Arrival-Departure-Times_2021-12.csv"
##  [4] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-01.csv"
##  [5] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-02.csv"
##  [6] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-03.csv"
##  [7] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-04.csv"
##  [8] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-05.csv"
##  [9] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-06.csv"
## [10] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-07.csv"
## [11] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-08.csv"
## [12] "BusData/MBTA-Bus-Arrival-Departure-Times_2022-09.csv"

bus<-lapply(setNames(,bus.data.files),function(x)
{
fread(x)->inter
})

bus<-rbindlist(bus)
bus<-bus[route_id %in% c('08','57','71')]
bus<-bus[!point_type %in% c('Pullout','Pullback')]

bus[,day:=mday(service_date)]
bus<-bus[day %in% 1:7]

bus<-bus[,.(service_date,route_id,stop_id,point_type,scheduled,actual,d
ay)]
bus<-na.omit(bus)

bus[,timeDiff:=as.numeric(actual-scheduled)]
bus[,sum(abs(timeDiff)<=1800)/.N]

## [1] 0.987855

bus<-bus[abs(timeDiff)<=1800]

fread('stops.txt')->sites
setNames(sites[,stop_name],sites[,stop_id])->sites.name

bus[,stop_name:=sites.name[as.character(stop_id)]]

weekdays.name<-strsplit('Sun,Mon,Tue,Wed,Thu,Fri,Sat',',')[[1]]
bus[,weekday:=wday(service_date)]
bus[,weekDay:=weekdays.name[as.integer(weekday)]]
```

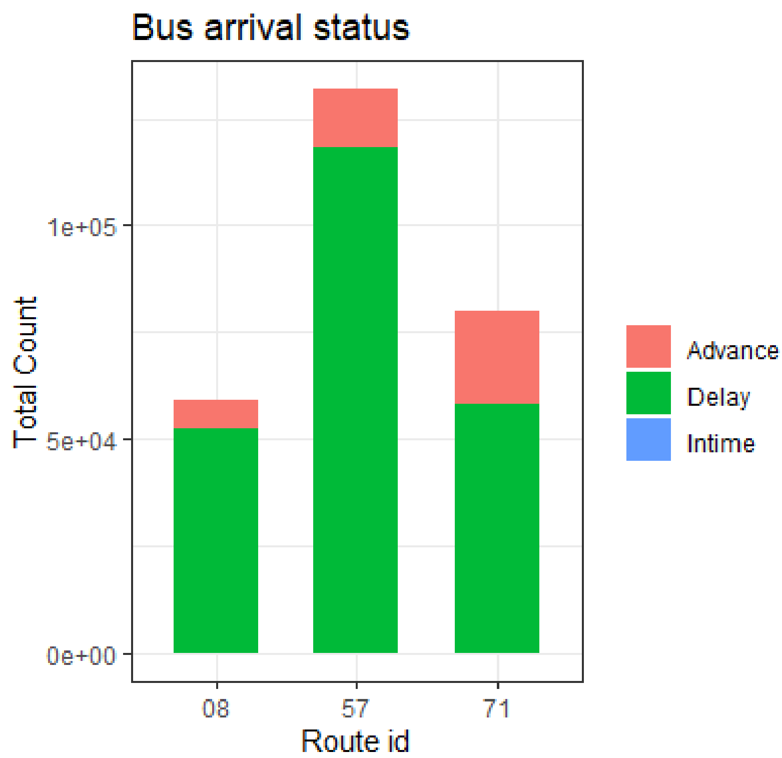
```
bus[,weekDay:=ordered(weekDay,weekdays.name)]
```

Bus EDA

```
library(data.table)
library(ggplot2)
library(stringr)
library(RColorBrewer)
library(flextable)
library(dplyr)
library(scales)
library(ggridges)

bus[,scheduledHour:=as.character(hour(scheduled))]
bus[,scheduledHour:=ordered(scheduledHour,0:23)]
bus[,type:=fcase(timeDiff==0,'Intime',timeDiff>0,'Delay',timeDiff<0,'Ad
vance')]
```

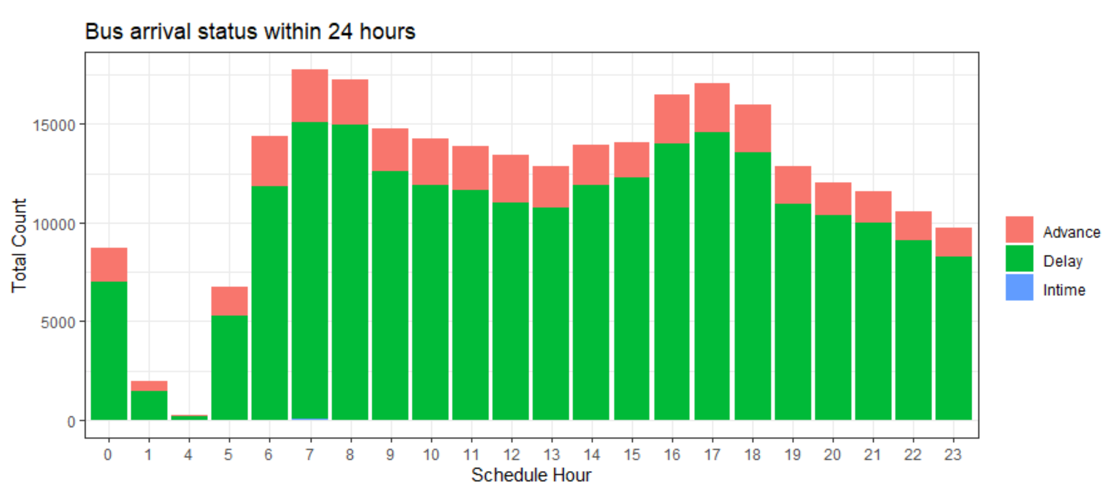
The distribution of Delay/Intime/Advance in the three routes

```
plot1.data<-bus[,.(count=.N),.(type,route_id)]

ggplot(plot1.data,aes(x=route_id,y=count,fill=type))+
geom_bar(stat='identity',width=0.6)+
theme_bw()+
    labs(x='Route id',
     y='Total Count',fill='',title = "Bus arrival status")
```



The distribution of Delay/Intime/Advance within 24 hours

```
plot2.data<-bus[,.(count=.N),.(type,scheduledHour)]

ggplot(plot2.data,aes(x=scheduledHour,y=count,fill=type))+
geom_bar(stat='identity')+
theme_bw()+
    labs(x='Schedule Hour',
     y='Total Count',fill='',title = "Bus arrival status within 24 hour
s")
```



Top 15 sites with the most delays

```
bus[type=='Delay',.(sumDelayTime=sum(timeDiff)),.(stop_name)][order(-su
mDelayTime)] %>%
head(15) %>%
    flextable() %>%
    width(j=1,width=4) %>%
    theme_vanilla() %>%
    set_caption('Top 15 sites with the most delays')
```

Top 15 sites with the most delays

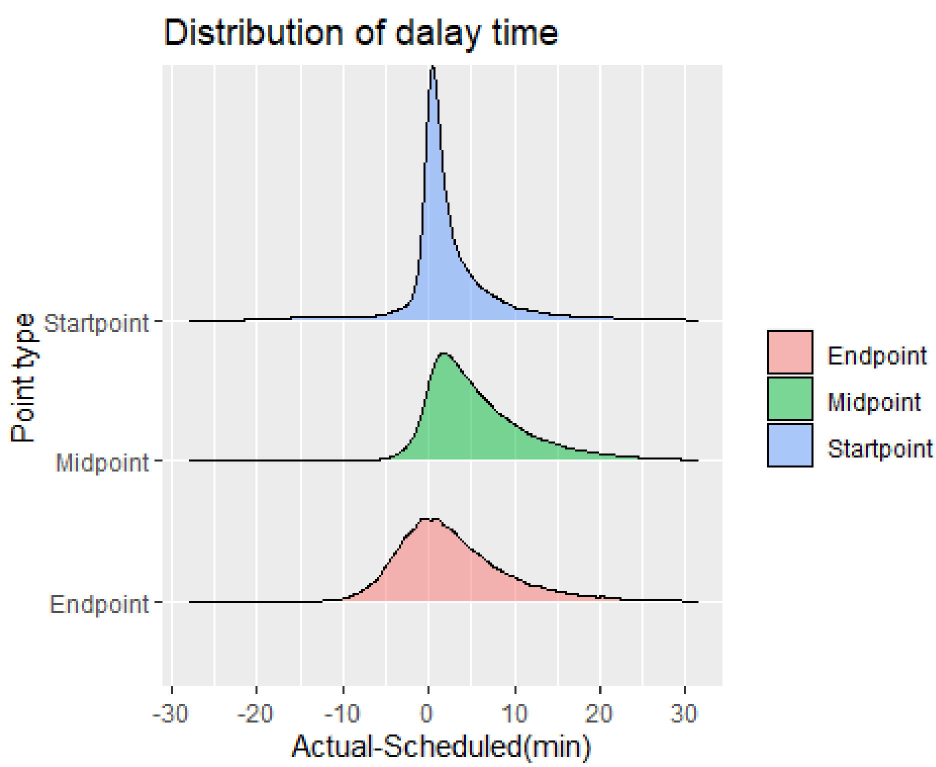
stop_name	sumDelayTime
Tremont St @ Washington St	6,781,864
Kenmore	5,679,976
Watertown Yard	4,278,675
Park St @ Tremont St	4,242,160
Washington St @ Market St	4,194,802
Commonwealth Ave @ Carlton St	4,104,640
Brighton Ave @ Commonwealth Ave	4,053,787
Cambridge St @ N Beacon St	3,782,957
Brighton Ave @ Cambridge St	3,692,237
1079 Commonwealth Ave	3,317,490
Washington St @ Chestnut Hill Ave	3,144,337
Commonwealth Ave @ University Rd	2,738,398

stop_name	sumDelayTime
Nubian	2,575,028
Ruggles St @ Huntington Ave	2,573,213
Ruggles	2,450,635

The stations with the most severe delays are “Tremont St @ Washington St”, “Kenmore” and “Watertown Yard”.

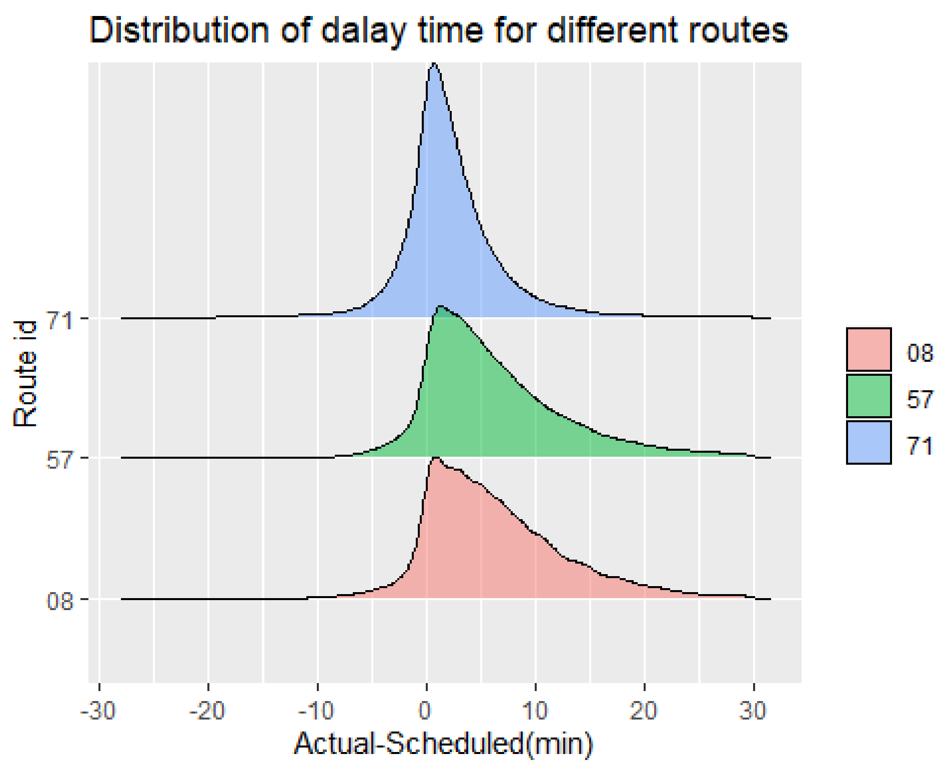
Distribution of dalay time

```
ggplot(bus,aes(y=point_type,x=timeDiff/60,fill=point_type))+
  geom_density_ridges(alpha=0.5)+
  labs(x='Actual-Scheduled(min)',
     y='Point type',fill='',title = "Distribution of dalay time")

## Picking joint bandwidth of 0.455
```



From the above chart can see the Startpoint punctuality is high, the Midpoint has a high probability of delay, but the Endpoint, some instead arrived in time or even early. And most of the bus are delayed within 10 minutes.

```
ggplot(bus,aes(y=route_id,x=timeDiff/60,fill=route_id))+
  geom_density_ridges(alpha=0.5)+
  labs(x='Actual-Scheduled(min)',
     y='Route id',fill='',title = "Distribution of dalay time for diffe
rent routes")

## Picking joint bandwidth of 0.46
```

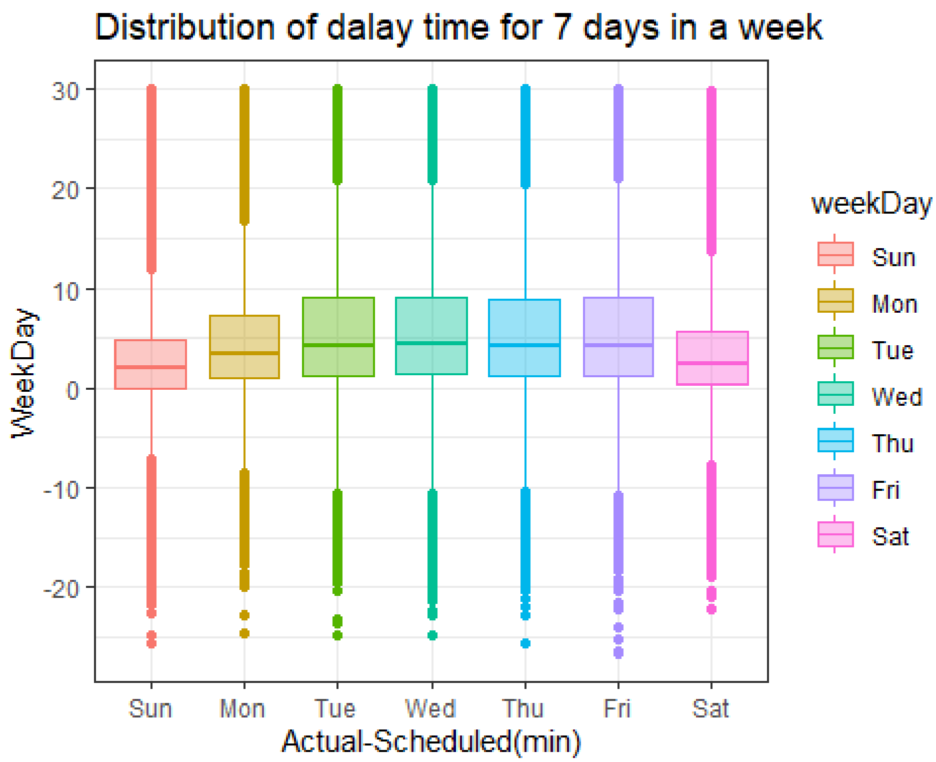


As you can see from the chart above, route 71 is more punctual and 08 and 57 are more delayed.

Distribution of dalay time for 7 days in a week

```
ggplot(bus,aes(x=weekDay,y=timeDiff/60,colour=weekDay,fill = after_scal
e(alpha(colour, 0.4))))+
   geom_boxplot()+
   theme_bw()+
   scale_colour_hue()+
  labs(x='Actual-Scheduled(min)',
```

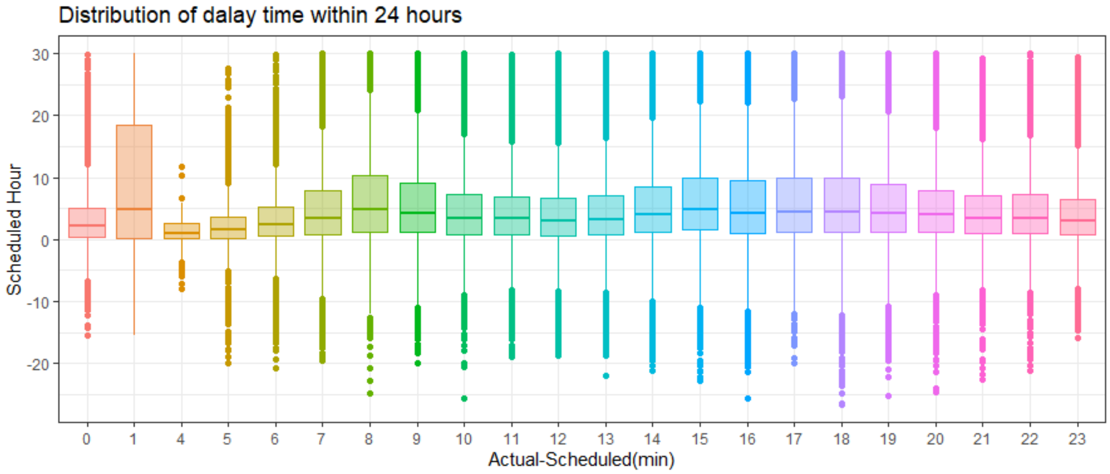
```
    y='WeekDay',fill='',title = "Distribution of dalay time for 7 days
 in a week")
```



The delay is more severe on weekdays than on Saturdays and Sundays.

Distribution of dalay time within 24 hours

```
ggplot(bus,aes(x=scheduledHour,y=timeDiff/60,colour=scheduledHour,fill
= after_scale(alpha(colour, 0.4))))+
   geom_boxplot()+
   theme_bw()+
   scale_colour_hue()+
   theme(legend.position='none')+
  labs(x='Actual-Scheduled(min)',
     y='Scheduled Hour',fill='',title = "Distribution of dalay time wit
hin 24 hours")
```



The delay varies from time to time, but during commuting hours, the delay is more serious. And in the morning and evening rush hour delays can exceed ten minutes.