Final Project

Zhi Tu

2022-12-17

Data from GTFS

First, We get the data from the GTFS site: https://cdn.mbta.com/archive/archived_feeds.txt After examining all these files, stop_times, stops, directions, and trip are most useful to use. For example, I can left_join trip onto the stop_times by the column `trip_id`. Also, I can left_join directions onto the stop_times by the column `route_id` and `direction_id`. Since the file uploaded onto github cannot exceed 100mb, I concise all the files under the limits.

```
##    trip_id      departure_time stop_id stop_sequence route_id direction_id
## 1 52554916 2022-12-17 05:02:00   70276             1 Mattapan            1
## 2 52554916 2022-12-17 05:03:00   70274             2 Mattapan            1
## 3 52554916 2022-12-17 05:04:00   70272             3 Mattapan            1
## 4 52554916 2022-12-17 05:06:00   70270             4 Mattapan            1
## 5 52554916 2022-12-17 05:07:00   70268             5 Mattapan            1
## 6 52554916 2022-12-17 05:08:00   70266             6 Mattapan            1
```

On the above is the head rows of processed stoptime

Bus

After getting all the stoptimes for different public transportation from GFTS, I would like to get stop informations. For buses, I read the bus stop info from site: https://mbta-massdot.opendata.arcgis.com/maps/MassDOT::mbta-systemwide-gtfs-map/about. After reading the file, I remove NAs in the Routes of buses and select columns with useful infos.

Next, I find the processed bus arrival and departure times file on site: https://mbta-massdot.opendata.arcgis.com/datasets/mbta-bus-arrival-departure-times-2021/about.

```
##   from_stop_id              start_time to_stop_id                end_time  time
## 1           64 1900-01-01 05:58:01.000          2 1900-01-01 05:58:54.000  0.88
## 2           64 1900-01-01 05:58:01.000      10590 1900-01-01 06:05:29.000  7.47
## 3           64 1900-01-01 05:58:01.000        188 1900-01-01 06:10:59.000 12.97
## 4           64 1900-01-01 05:58:01.000         93 1900-01-01 06:14:11.000 16.17
## 5           64 1900-01-01 05:58:01.000         97 1900-01-01 06:16:43.000 18.70
## 6           64 1900-01-01 05:58:01.000        102 1900-01-01 06:21:48.000 23.78
```

As you can see, we need to convert the departure time on each stop to the time period between stops. Specifically, if the route has stop order from 1 to 9, then we need to have 8! combinations of time period to be calculated.

Then, we can left join the departure stop info and the arrival stop info onto the combination to get all the time period from all the routes of buses.

Ferry

Next, we do the same thing for the ferry stop info. After getting the data, I found some names are named different across the file especially between the stop info and stoptimes. So I need to change one so that they can match to each other.

```
head(ferry_stop)
```

```
##           X        Y OBJECTID          stop_id   stop_name      platform_name
## 1 -71.02734 42.35979   638612       Boat-Logan       LOGAN Hingham/Hull Ferry
## 2 -71.05253 42.37276   638607 Boat-Charlestown Charlestown  Charlestown Ferry
## 3 -70.93043 42.31974   638608      Boat-George     Georges Hingham/Hull Ferry
## 4 -70.91984 42.25396   638609     Boat-Hingham     Hingham Hingham/Hull Ferry
## 5 -70.92022 42.30325   638610        Boat-Hull        Hull Hingham/Hull Ferry
## 6 -71.04196 42.36587   638611       Boat-Lewis Lewis Wharf  East Boston Ferry
##                           stop_address municipality
## 1 Harborside Dr, East Boston, MA 02128       Boston
## 2             Pier 4, Boston, MA 02129       Boston
## 3          George's Island, Boston, MA       Boston
## 4    28 Shipyard Dr, Hingham, MA 02043      Hingham
## 5          180 Main St, Hull, MA 02045         Hull
## 6        65 Lewis St, Boston, MA 02128       Boston
##                                    stop_url
## 1       https://www.mbta.com/stops/Boat-Logan
## 2 https://www.mbta.com/stops/Boat-Charlestown
## 3      https://www.mbta.com/stops/Boat-George
## 4     https://www.mbta.com/stops/Boat-Hingham
## 5        https://www.mbta.com/stops/Boat-Hull
## 6       https://www.mbta.com/stops/Boat-Lewis
```

The data I got the daily info from:https://mbta-massdot.opendata.arcgis.com/search?tags=ferry I then clean the data so that it can be match to the ferry stop info file.

```
head(ferry_daily)
```

```
##   route_id departure_terminal    actual_departure      actual_arrival
## 1       F1            Hingham 2018-11-05 17:45:00 2018-11-05 18:20:00
## 2       F1        Rowes Wharf 2018-11-05 18:31:00 2018-11-05 19:07:00
## 3       F1            Hingham 2018-11-05 06:50:00 2018-11-05 07:30:00
## 4       F1        Rowes Wharf 2018-11-05 07:35:00 2018-11-05 08:13:00
## 5       F1            Hingham 2018-11-05 08:20:00 2018-11-05 09:00:00
## 6       F1        Rowes Wharf 2018-11-05 16:00:00 2018-11-05 16:38:00
##   arrival_terminal time
## 1      Rowes Wharf   35
## 2          Hingham   36
## 3      Rowes Wharf   40
## 4          Hingham   38
## 5      Rowes Wharf   40
## 6          Hingham   38
```

Commuter Rail

Commuter Rail is exactly same the buses, we need to calculate the time period based on the departure time. Here is the results:

```
head(rail_stop)
```

```
##      stop_id       stop_name stop_lat  stop_lon
## 1  CM-0493-S Wareham Village 41.75833 -70.71472
## 2  CM-0547-S    Buzzards Bay 41.74480 -70.61623
## 3  CM-0564-S          Bourne 41.74650 -70.58877
## 4  CM-0790-S         Hyannis 41.66022 -70.27658
## 5    DB-0095       Readville 42.23841 -71.13325
## 6 FB-0095-04       Readville 42.23841 -71.13325
```

```
head(rail)
```

```
## # A tibble: 6 x 7
## # Groups:   trip_id [1]
##   trip_id  route~1 from_~2 start_time          to_st~3 end_time             time
##   <chr>    <chr>   <chr>   <dttm>              <chr>   <dttm>              <dbl>
## 1 CR-5508~ CR-Mid~ MM-035~ 2022-12-17 05:15:00 MM-027~ 2022-12-17 05:25:00    10
## 2 CR-5508~ CR-Mid~ MM-035~ 2022-12-17 05:15:00 MM-021~ 2022-12-17 05:32:00    17
## 3 CR-5508~ CR-Mid~ MM-035~ 2022-12-17 05:15:00 MM-0200 2022-12-17 05:36:00    21
## 4 CR-5508~ CR-Mid~ MM-035~ 2022-12-17 05:15:00 MM-0186 2022-12-17 05:39:00    24
## 5 CR-5508~ CR-Mid~ MM-035~ 2022-12-17 05:15:00 MM-015~ 2022-12-17 05:44:00    29
## 6 CR-5508~ CR-Mid~ MM-035~ 2022-12-17 05:15:00 MM-0109 2022-12-17 05:50:00    35
## # ... with abbreviated variable names 1: route_id, 2: from_stop_id,
## #   3: to_stop_id
```

MTBA data

Lastly, we got the MTBA data from the travel time file: https://mbta-massdot.opendata.arcgis.com/datasets/mbta-travel-times-2021/about

```
head(tt_q1_2021_lr)
```

```
##   route_id from_stop_id start_time_sec to_stop_id end_time_sec travel_time_sec
## 1  Green-B        70111          50344      70107        50476             132
## 2  Green-B        70111          50740      70107        50842             102
## 3  Green-B        70111          25744      70107        26039             295
## 4  Green-B        70111          26440      70107        26632             192
## 5  Green-B        70111          26440      70107        26594             154
## 6  Green-B        70111          26645      70107        26772             127
```

Graph

I start the exploration on the MTBA data. At the beginning, I separate the subway by its `route_id` to different lines.

```
unique(tt_q1_2021_lr$route_id)
```

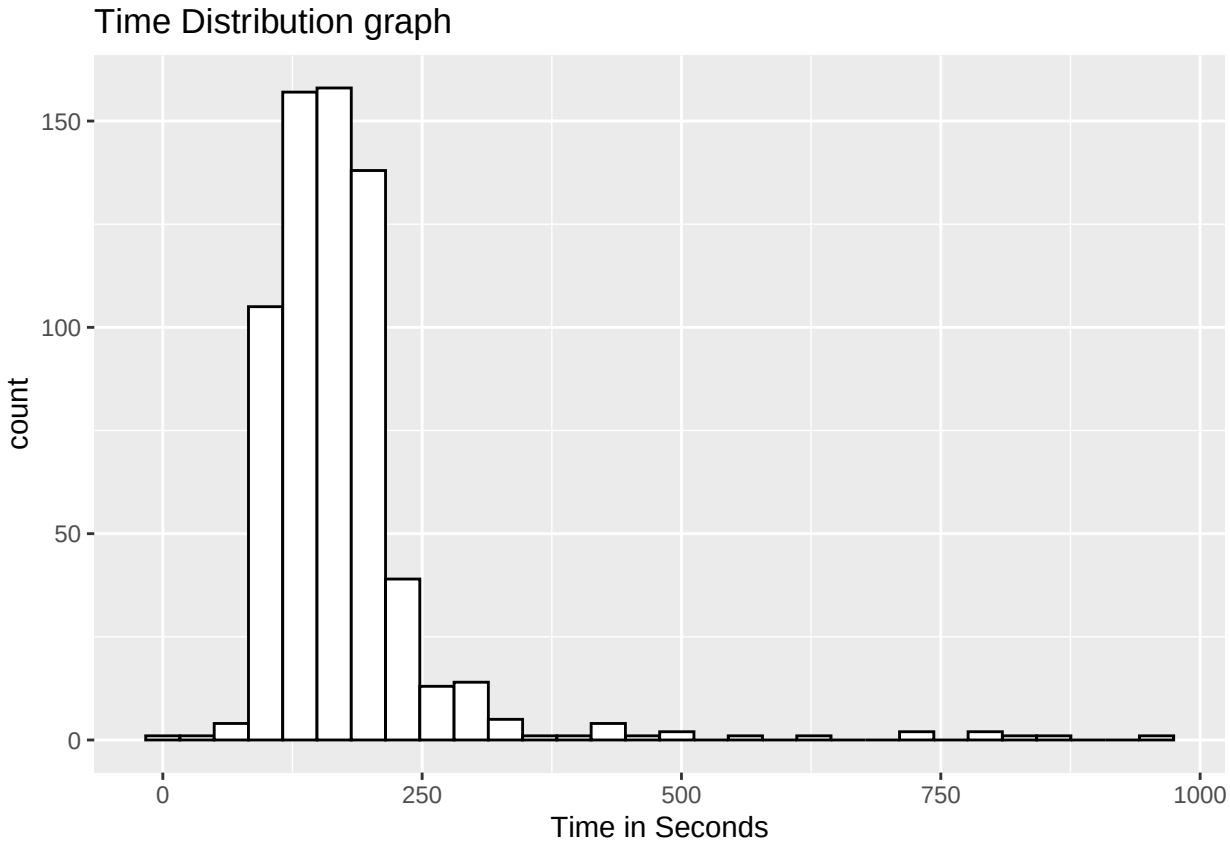
```
## [1] "Green-B"  "Green-C"  "Green-D"  "Green-E"  "Mattapan" "Orange"   "Blue"
## [8] "Red"
```

Take the Green-B line as an example:

```
##   route_id from_stop_id start_time_sec to_stop_id end_time_sec travel_time_sec
## 1  Green-B        70111          50344      70107        50476             132
## 2  Green-B        70111          50740      70107        50842             102
## 3  Green-B        70111          25744      70107        26039             295
## 4  Green-B        70111          26440      70107        26632             192
## 5  Green-B        70111          26440      70107        26594             154
## 6  Green-B        70111          26645      70107        26772             127
```

Here is the graph of time distribution between stops that I would like to include in my shinyApp:

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



In addition to graph, I would also like a summary table for the time distribution between specific stops.

```
summary(green_b$travel_time_sec)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   127.0   162.0   174.8   193.0   958.0
```