Frequency Extraction

Instructions on finding headway data from: http://tidytransit.r-transit.org/articles/frequency.html#:~:text=The%20get_stop_frequency%20function%20simply%20counts%20the%20number%20of,of%20departures%2C%20and%20rounds%20to%20the%20nearest%20inte

the # <- symbol disables the code. If you want to run the code, get rid of the symbol

load libraries

```
library(sf)
library(tidytransit)
library(pryr)
library(dplyr)
library(ggplot2)
library(readr)
library(rgdal)
library(crsuggest)
library(tmap)
```

If you have a URL source of the gtfs zip file, you can replace this URL and get rid of the "#"

```
#gtfs <- read_gtfs("http://web.mta.info/developers/data/nyct/subway/google_transit.zip")
```

After downloading the GTFS zip file, replace the file path with the one in here

```
zip <- "N:/Projects/MeadHunt/Clearwater_4615600_220262_01/DATA/GTFS.zip"
outDir <- substring(zip, 1, nchar(zip)-4)


dir.create(outDir)
setwd(outDir)

unzip(zip, exdir = outDir)
trips <- read_csv("trips.txt")
routes <- read_csv("routes.txt")
stop_times <- read_csv("stop_times.txt", col_types= cols(arrival_time = col_character(), departure_time = col_cha
racter()))
```

Make st_length is in sf, add shape lengths

```
gtfs <- read_gtfs("N:/Projects/MeadHunt/Clearwater_4615600_220262_01/DATA/GTFS.zip")
gtfs <- set_servicepattern(gtfs)

gtfs <- gtfs_as_sf(gtfs)
gtfs$shapes$length <- st_length(gtfs$shapes)

shape_lengths <- gtfs$shapes %>%
  as.data.frame() %>%
  select(shape_id, length, -geometry)
```

Join all the gtfs tables together

```
service_pattern_summary <- gtfs$trips %>%
  left_join(gtfs$.$servicepatterns, by="service_id") %>%
  left_join(shape_lengths, by="shape_id") %>%
  left_join(gtfs$stop_times, by="trip_id") %>%
  group_by(servicepattern_id) %>%
  summarise(
    trips = n(),
    routes = n_distinct(route_id),
    total_distance_per_day_km = sum(as.numeric(length), na.rm=TRUE)/1e3,
    route_avg_distance_km = (sum(as.numeric(length), na.rm=TRUE)/1e3)/(trips*routes),
    stops=(n_distinct(stop_id)/2))
```

Grouping by id and summarizing

```
service_pattern_summary <- gtfs$.$dates_servicepatterns %>%
  group_by(servicepattern_id) %>%
  summarise(days_in_service = n()) %>%
  left_join(service_pattern_summary, by="servicepattern_id")
```

This provides a table of the different service patterns and includes information with each one

```
knitr::kable(service_pattern_summary)
```

servicepattern_id	days_in_service	trips	routes	total_distance_per_day_km	route_avg_distance_km	stops
s_04313a1	1	66106	32	1752644	0.8285197	1814.0
s_1053c6e	48	127558	44	3608277	0.6428941	2197.5
s_149b004	16	128736	44	3644394	0.6433875	2197.5
s_27b4ba1	2	60932	30	1659931	0.9080783	1786.0
s_28dd690	9	105909	40	2929943	0.6916181	2123.0
s_38e78d6	14	102618	39	2895149	0.7234072	2103.0
s_5382c2b	5	139997	46	3947643	0.6130012	2231.5
s_697fc68	5	141291	46	3981354	0.6125739	2230.5
s_6b14600	9	64202	31	1714791	0.8615908	1802.0
s_8a0539a	15	127558	44	3608277	0.6428941	2197.5
s_8bd74b8	16	60631	30	1651465	0.9079321	1786.0
s_913ad6d	5	107632	41	2960044	0.6707688	2134.5
s_954b400	14	139988	46	3947567	0.6130288	2231.5
s_9c1cb0b	1	73521	30	2035797	0.9230000	1786.0
s_b27334b	5	66017	32	1748800	0.8278171	1813.0
s_f7dbdba	27	137651	45	3909124	0.6310846	2220.0
s_faa5e20	9	137660	45	3909200	0.6310556	2220.0
s_fce3ae0	9	139568	45	3951254	0.6291245	2219.0

Pick the service pattern with the most days in service or the most trips to summarize the most common patterns of service in the study area. Copy the servicepattern id and replace the one in this code with your own. The pattern with the most days in service for this example is s_1053c6e

```
service_ids <- gtfs$.$servicepattern %>%
  filter(servicepattern_id == 's_1053c6e') %>%
  pull(service_id)

head(service_ids) %>%
  knitr::kable()
```

The below code shows how many trips fall under each of these service_ids on the trips table, and how they relate to routes.

```
gtfs$trips %>%
  filter(service_id %in% service_ids) %>%
  group_by(service_id, route_id) %>%
  summarise(count = n()) %>%
  head() %>%
  knitr::kable()
```

Map Headways by Route for both AM and PM peak hours

Load in study border

```
service_area <- st_read("N:/Projects/MeadHunt/Clearwater_4615600_220262_01/DATA/clearwater_service_area/service_a
rea.shp")
```

The get_stop_frequency function used below simply counts the number of departures within the time frame to get departures per stop. Then, TO GET HEADWAYS, it divides the number of seconds (between start_time and end_time) by the number of departures, and rounds to the nearest integer.

The code below creates both an AM and PM variable with a 7 am start / 9 am end and 4pm start / 6 pm end time respectfully

```
am_route_freq <- get_route_frequency(gtfs, service_ids = service_ids,
                                     start_time = 7*3600, end_time = 9*3600)
head(am_route_freq) %>%
  knitr::kable()

pm_route_freq <- get_route_frequency(gtfs, service_ids = service_ids,
                                     start_time = 16*3600, end_time = 18*3600)
head(pm_route_freq) %>%
  knitr::kable()
```

Get_route_geometry needs a gtfs object that includes shapes as simple feature data frames

```
routes_sf <- get_route_geometry(gtfs, service_ids = service_ids)
```

Then we join the geometries to the calculated frequencies:

```
routes_sf_am <- routes_sf %>%
  inner_join(am_route_freq, by = 'route_id')

routes_sf_pm <- routes_sf %>%
  inner_join(pm_route_freq, by = 'route_id')
```

Get suggested projections.. first result for this example is 6443

```
suggest_crs(routes_sf)
```

Convert the study area and routes to the suggested projection

```
service_area_crs <- st_transform(service_area, 6443)
routes_crs_am <- sf::st_transform(routes_sf_am, 6443)
```

Make list of routes needed in study area

```
list_data <- list(18, 19, 52, "52LX", 60, 61, 62, 65, 66, 67, 73, 76, 78, 812, 814, "JTC", "JTNB", "JTSB", "SCBT"
)
```

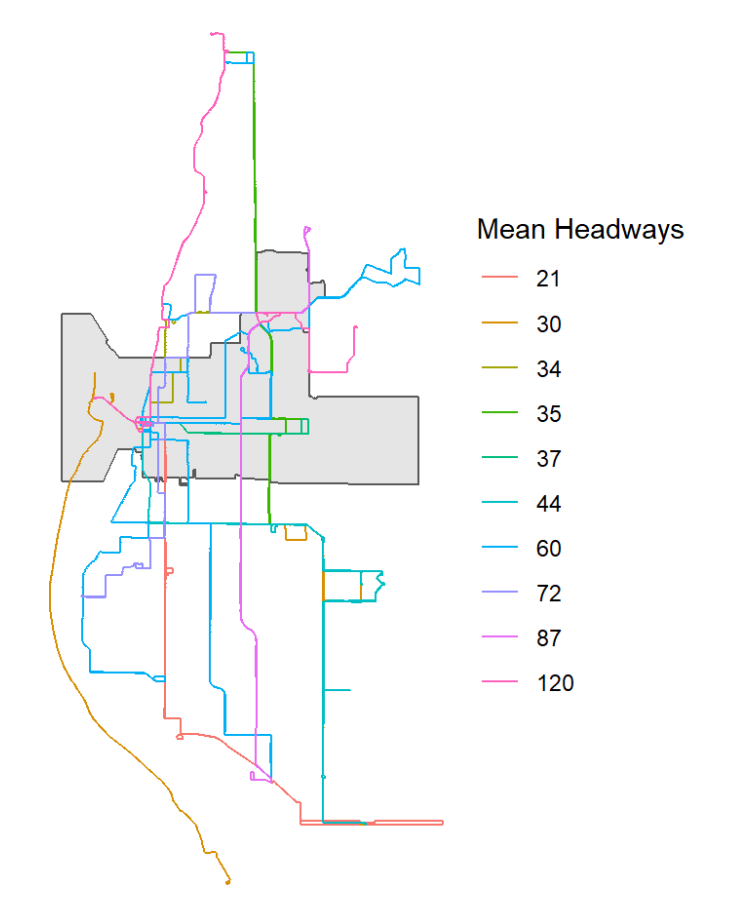
Prepare to plot the routes with mean headways for the morning. Manipulate fields

```
#convert headways from seconds to minutes and filter median headways to be less than 60 min
routes_crs_am <- routes_crs_am %>%
  mutate(mean_headway_minutes = mean_headways/60) %>%
  mutate(median_headway_minutes = median_headways/60)  %>%
  filter(route_id %in% list_data)  # %>%                              #this filters routes with our list we made
   #filter(mean_headways < 60*60)
                        #enable this line if you want to filter routes with mean headways of 60 minutes or less
                        #this will affect the results in the future sections

#round decimal places
routes_sf_round_am <- routes_crs_am %>%
  mutate_if(is.numeric,
            round,
            digits = 0)
```

Plot headways using the data with rounded decimal places

```
  ggplot(data=routes_sf_round_am) +
  geom_sf(data=service_area_crs) +
  geom_sf(aes(colour=as.factor(mean_headway_minutes))) +
  labs(color = "Mean Headways") +
  #geom_sf_text(aes(label=route_id)) +                         #enable this line if you want to see route labels
  theme_void()                                                  #gets rid of coordinates
```



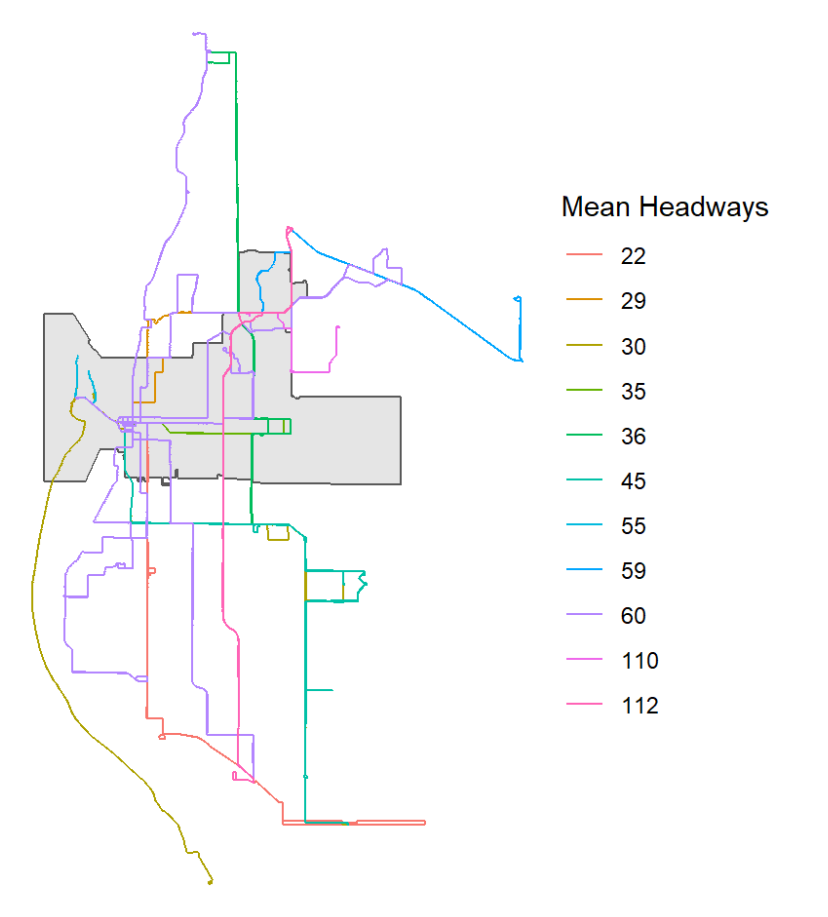

```
                                                    #do theme_bw() instead to create coordinate grid around pl
ot
```

Convert to an appropriate coordinate reference system Convert headways from seconds to minutes and filter median headways to be less than 60 min

```
routes_crs_pm <- sf::st_transform(routes_sf_pm, 6443)
routes_crs_pm <- routes_crs_pm %>%
  mutate(mean_headway_minutes = mean_headways/60) %>%
  mutate(median_headway_minutes = median_headways/60)  %>%
  filter(route_id %in% list_data)  # %>%                              #filter routes with our list we made
   #filter(mean_headways < 60*60)
                        #enable this line if you want to filter routes with mean headways of 60 minutes or less
                        #this will affect the results in the future sections

#round decimal places
routes_sf_round_pm <- routes_crs_pm %>%
  mutate_if(is.numeric,
            round,
            digits = 0)
```

```
#plot headways using the data with rounded decimal places
  ggplot(data=routes_sf_round_pm) +
  geom_sf(data=service_area_crs) +
  geom_sf(aes(colour=as.factor(mean_headway_minutes))) +
  labs(color = "Mean Headways") +
  #geom_sf_text(aes(label=route_id)) +                         #enable this line if you want to see route labels
  theme_void()
```



If you want the headway shapefile as it is, just change the file path: But we will calculate frequencies in the following code

```
#st_write(routes_sf_round_am, "N:/Projects/MeadHunt/Clearwater_4615600_220262_01/DATA/R_GTFS_Extract/MeanHeadways
_am.shp")
#st_write(routes_sf_round_pm, "N:/Projects/MeadHunt/Clearwater_4615600_220262_01/DATA/R_GTFS_Extract/MeanHeadways
_am.shp")
```

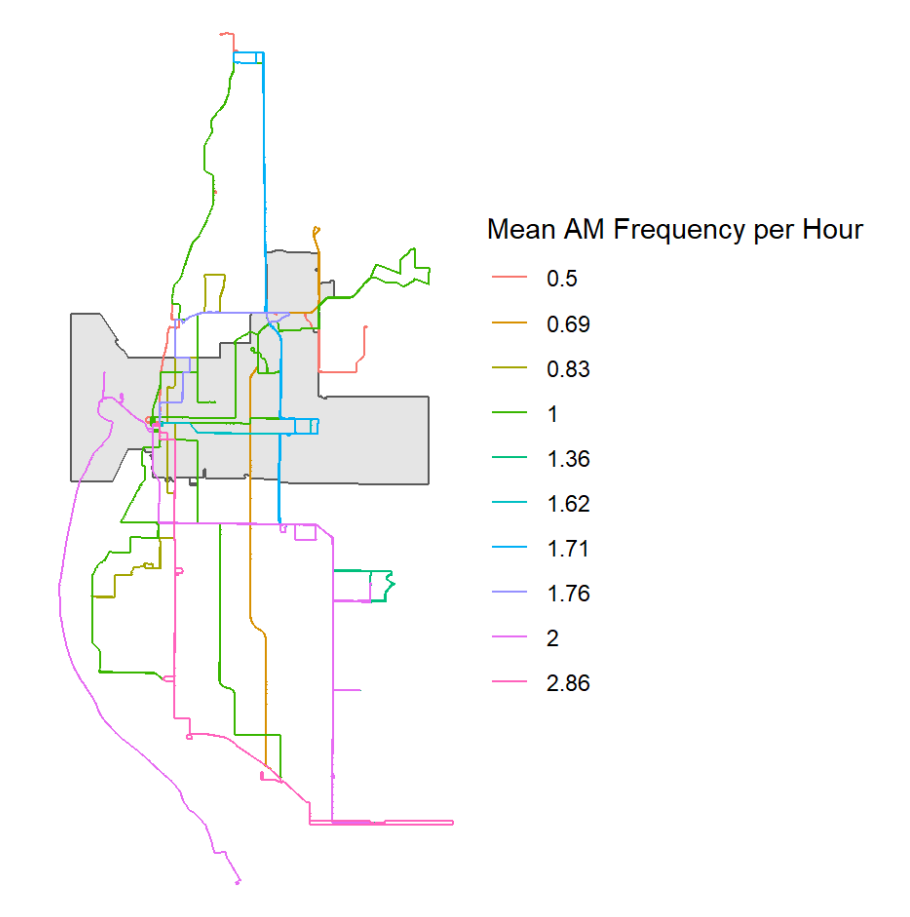
Getting frequency from headways

```
#get frequency per hour by dividing 60 by headway in minutes
routes_frequency_am <- routes_sf_round_am %>%
  mutate(mean_frequency = 60/mean_headway_minutes)

#round decimal places
routes_freq_round_am <- routes_frequency_am %>%
  mutate_if(is.numeric,
            round,
            digits = 2)
```

Plot headways using the data with rounded decimal places

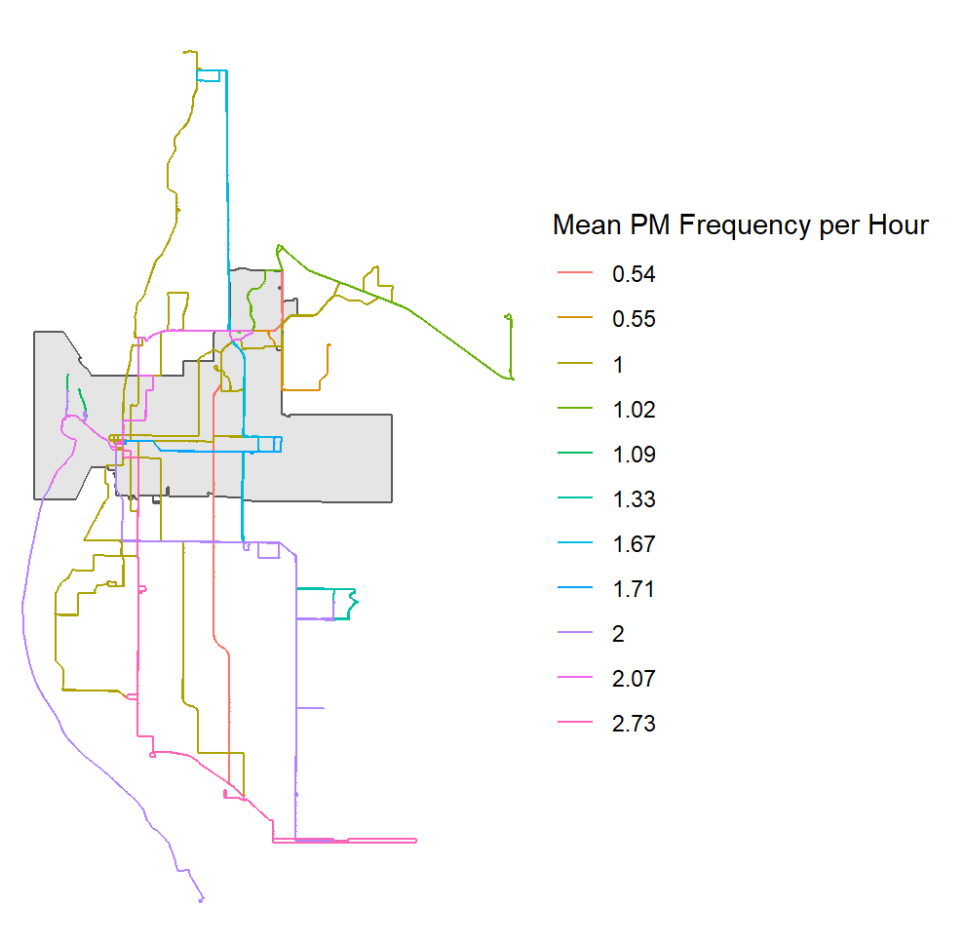
```
  ggplot(data=routes_freq_round_am) +
  geom_sf(data=service_area_crs) +
  geom_sf(aes(colour=as.factor(mean_frequency))) +
  labs(color = "Mean AM Frequency per Hour") +
  #geom_sf_text(aes(label=route_id)) +                    #enable this line if labels are wanted
  theme_void()
```




```
#get frequency per hour by dividing 60 by headway in minutes
routes_frequency_pm <- routes_sf_round_pm %>%
  mutate(mean_frequency = 60/mean_headway_minutes)

#round decimal places
routes_freq_round_pm <- routes_frequency_pm %>%
  mutate_if(is.numeric,
            round,
            digits = 2)
```

```
#plot headways using the data with rounded decimal places
  ggplot(data=routes_freq_round_pm) +
  geom_sf(data=service_area_crs) +
  geom_sf(aes(colour=as.factor(mean_frequency))) +
  labs(color = "Mean PM Frequency per Hour") +
  #geom_sf_text(aes(label=route_id)) +                    #enable this line if labels are wanted
  theme_void()
```



This is how to download all the mean frequencies for each route as a single shapefile Just delete the # sign the replace the file path with your own

```
#st_write(routes_freq_round_am, "N:/Projects/MeadHunt/Clearwater_4615600_220262_01/DATA/R_GTFS_Extract/frequency_
AM.shp")

#st_write(routes_freq_round_pm, "N:/Projects/MeadHunt/Clearwater_4615600_220262_01/DATA/R_GTFS_Extract/frequency_
PM.shp")
```