Project_Analysis

Jayant Yadav

2023-02-11

Loading the required Packages and Libraries

```r
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2
## --

## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Load The Dataset

The data is stored in excel files that first loaded into R environment know moving these excel files to the dataframe and combining them into a single data frame

loading the data into dataframe

```r
df1 <- read_csv("202101-divvy-tripdata.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)

## Rows: 96834 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): ride_id, rideable_type, started_at, ended_at, start_station_name,...
## dbl   (4): start_lat, start_lng, end_lat, end_lng
## time  (1): ride_length
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df2 <- read_csv("202102-divvy-tripdata.csv")
```

```
## Rows: 49622 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (10): ride_id, rideable_type, started_at, ended_at, start_station_name,...
## dbl   (4): start_lat, start_lng, end_lat, end_lng
## time  (1): ride_length
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Combining both the Dataframe

```
complete_data <- rbind(df1,df2)
```

now we have all the dataset in a single dataframe (complete_data), so there is no need of df1 and df2 we will remove them from the memory.

Removing the temporary Dataframe

```
rm(df1)
rm(df2)
```

load and view the Complete data

Now we have the required data in one place so we will load the data and have a look on the dataset what are the different fields we have in our dataset and if any data cleaning is required or not.

```
view(complete_data)

head(complete_data)
```

```
## # A tibble: 6 x 15
##   ride_id         ridea~1 start~2 ended~3 start~4 start~5 end_s~6 end_s~7 start~8
##   <chr>           <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>     <dbl>
## 1 E19E6F1B8D4C4~ electr~ 23-01-~ 23-01-~ Califo~ 17660   <NA>    <NA>       41.9
## 2 DC88F20C2C55F~ electr~ 27-01-~ 27-01-~ Califo~ 17660   <NA>    <NA>       41.9
## 3 EC45C94683FE3~ electr~ 21-01-~ 21-01-~ Califo~ 17660   <NA>    <NA>       41.9
## 4 4FA453A75AE37~ electr~ 07-01-~ 07-01-~ Califo~ 17660   <NA>    <NA>       41.9
## 5 BE5E8EB4E7263~ electr~ 23-01-~ 23-01-~ Califo~ 17660   <NA>    <NA>       41.9
## 6 5D8969F88C773~ electr~ 09-01-~ 09-01-~ Califo~ 17660   <NA>    <NA>       41.9
## # ... with 6 more variables: start_lng <dbl>, end_lat <dbl>, end_lng <dbl>,
## #   member_casual <chr>, ride_length <time>, day_of_week <chr>, and abbreviated
## #   variable names 1: rideable_type, 2: started_at, 3: ended_at,
## #   4: start_station_name, 5: start_station_id, 6: end_station_name,
## #   7: end_station_id, 8: start_lat
```

```
nrow(complete_data)
```

```
## [1] 146456
```

```
colnames(complete_data)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"      "ride_length"        "day_of_week"
```

```
str(complete_data)
```

```
## spc_tbl_ [146,456 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:146456] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA4
##  $ rideable_type     : chr [1:146456] "electric_bike" "electric_bike" "electric_bike" "electric_bike
##  $ started_at        : chr [1:146456] "23-01-2021 16:14" "27-01-2021 18:43" "21-01-2021 22:35" "07-0
##  $ ended_at          : chr [1:146456] "23-01-2021 16:24" "27-01-2021 18:47" "21-01-2021 22:37" "07-0
##  $ start_station_name: chr [1:146456] "California Ave & Cortez St" "California Ave & Cortez St" "Cal
##  $ start_station_id  : chr [1:146456] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:146456] NA NA NA NA ...
##  $ end_station_id    : chr [1:146456] NA NA NA NA ...
##  $ start_lat         : num [1:146456] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:146456] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:146456] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:146456] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:146456] "member" "member" "member" "member" ...
##  $ ride_length       : 'hms' num [1:146456] 00:10:25 00:04:04 00:01:20 00:11:42 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_week       : chr [1:146456] "Saturday" "Wednesday" "Thursday" "Thursday" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character(),
##   ..   ride_length = col_time(format = ""),
##   ..   day_of_week = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
glimpse(complete_data)
```

```
## Rows: 146,456
## Columns: 15
## $ ride_id            <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F", "EC45C94683~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <chr> "23-01-2021 16:14", "27-01-2021 18:43", "21-01-2021~
## $ ended_at           <chr> "23-01-2021 16:24", "27-01-2021 18:47", "21-01-2021~
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor~
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augu~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "657", "13258",~
## $ start_lat          <dbl> 41.90034, 41.90033, 41.90031, 41.90040, 41.90033, 4~
## $ start_lng          <dbl> -87.69674, -87.69671, -87.69664, -87.69666, -87.696~
## $ end_lat            <dbl> 41.89000, 41.90000, 41.90000, 41.92000, 41.90000, 4~
## $ end_lng            <dbl> -87.72000, -87.69000, -87.70000, -87.69000, -87.700~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
## $ ride_length        <time> 00:10:25, 00:04:04, 00:01:20, 00:11:42, 00:00:43, ~
## $ day_of_week        <chr> "Saturday", "Wednesday", "Thursday", "Thursday", "S~
```

After looking into the data, it was found that below 2 mentioned variables have incorrect data types: * started_at * ended_at

Changing the datatype of both the variables to the Date type.

```
complete_data$started_at <- as.Date(complete_data$started_at,format = "%d-%m-%y")
complete_data$ended_at <- as.Date(complete_data$ended_at,format = "%d-%m-%y")
```

Now will check the complete summary of our data that will give us the brief idea about each variables.

```
summary(complete_data)
```

```
##    ride_id          rideable_type        started_at
##  Length:146456      Length:146456      Min.   :2020-01-01
##  Class :character   Class :character   1st Qu.:2020-01-12
##  Mode  :character   Mode  :character   Median :2020-01-22
##                                        Mean   :2020-01-26
##                                        3rd Qu.:2020-02-09
##                                        Max.   :2020-02-28
##
##     ended_at          start_station_name start_station_id   end_station_name
##  Min.   :2020-01-01   Length:146456      Length:146456      Length:146456
##  1st Qu.:2020-01-12   Class :character   Class :character   Class :character
##  Median :2020-01-22   Mode  :character   Mode  :character   Mode  :character
##  Mean   :2020-01-26
##  3rd Qu.:2020-02-09
##  Max.   :2020-03-05
##
##  end_station_id       start_lat       start_lng         end_lat
##  Length:146456      Min.   :41.64   Min.   :-87.78   Min.   :41.54
##  Class :character   1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88
##  Mode  :character   Median :41.90   Median :-87.64   Median :41.90
##                     Mean   :41.90   Mean   :-87.65   Mean   :41.90
##                     3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93
##                     Max.   :42.06   Max.   :-87.53   Max.   :42.07
##                                                      NA's   :317
##     end_lng        member_casual       ride_length       day_of_week
##  Min.   :-87.81   Length:146456      Length:146456      Length:146456
##  1st Qu.:-87.66   Class :character   Class1:hms         Class :character
##  Median :-87.64   Mode  :character   Class2:difftime    Mode  :character
##  Mean   :-87.65                      Mode  :numeric
##  3rd Qu.:-87.63
##  Max.   :-87.51
##  NA's   :317
```

Cleaning Data

Now we will clean or filter our data, in the given dataset there are some 0 values in ride_length in this some data is not proper. How this field is calculated : Ride_length = started_at - ended_at so there are some records where both the started_at and ended_at are same or there are some records in which ended_at is greater than started_at.

So, removing all these incorrect data and considering only where ride_length is graeter than zero.

```
complete_data <-complete_data %>%
  filter(ride_length>0)
print("Cleaned data Count:")
```

```
## [1] "Cleaned data Count:"
```

```
nrow(complete_data)
```

```
## [1] 146446
```

Descriptive Analytics

overall Data

```
complete_data %>%
  summarize(mean(ride_length),max(ride_length))
```

```
## # A tibble: 1 x 2
##   `mean(ride_length)` `max(ride_length)`
##   <drtn>              <drtn>
## 1 960.7154 secs       85658 secs
```

Based on member and weekdays

```
complete_data %>%
  group_by(member_casual, day_of_week) %>%
  summarise(count=n(),min(ride_length),max(ride_length),mean(ride_length),median(ride_length)) %>%
  arrange(member_casual,count)
```

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

```
## # A tibble: 14 x 7
## # Groups:   member_casual [2]
##    member_casual day_of_week count `min(ride_length)` max(rid~1 mean(~2 median~3
##    <chr>         <chr>       <int> <drtn>             <drtn>    <drtn>  <time>
##  1 casual        Monday       2673 2 secs             59074 se~ 1284.3~ 12'18.0"
##  2 casual        Tuesday      2889 1 secs             85658 se~ 1456.7~ 12'23.0"
##  3 casual        Wednesday    3216 1 secs             84850 se~ 1366.1~ 12'13.5"
##  4 casual        Thursday     3390 1 secs             67359 se~ 1210.3~ 11'59.0"
##  5 casual        Sunday       4257 2 secs             85335 se~ 1571.5~ 14'05.0"
##  6 casual        Friday       4334 1 secs             82726 se~ 1392.6~ 12'02.0"
##  7 casual        Saturday     7488 2 secs             85126 se~ 1908.7~ 16'58.5"
##  8 member        Sunday      12863 1 secs             80253 se~  882.0~ 09'25.0"
##  9 member        Monday      15138 1 secs             84290 se~  844.8~ 08'52.0"
## 10 member        Tuesday     16297 1 secs             75714 se~  787.6~ 09'06.0"
## 11 member        Wednesday   17820 1 secs             81170 se~  827.9~ 09'02.5"
## 12 member        Thursday    17997 1 secs             80188 se~  763.3~ 08'49.0"
## 13 member        Saturday    18847 1 secs             67142 se~  886.5~ 10'10.0"
## 14 member        Friday      19237 1 secs             66756 se~  798.6~ 08'50.0"
## # ... with abbreviated variable names 1: `max(ride_length)`,
## #   2: `mean(ride_length)`, 3: `median(ride_length)`
```

Average ride length by user type and weekdays

```
complete_data %>%
  group_by(member_casual,day_of_week) %>%
  summarise(average_duration = mean(ride_length)) %>%
  arrange(member_casual,-average_duration)
```

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

```
## # A tibble: 14 x 3
## # Groups:   member_casual [2]
##    member_casual day_of_week average_duration
##    <chr>         <chr>       <drtn>
##  1 casual        Saturday    1908.7714 secs
##  2 casual        Sunday      1571.5600 secs
##  3 casual        Tuesday     1456.7189 secs
##  4 casual        Friday      1392.6913 secs
##  5 casual        Wednesday   1366.1872 secs
##  6 casual        Monday      1284.3124 secs
##  7 casual        Thursday    1210.3991 secs
##  8 member        Saturday     886.5955 secs
##  9 member        Sunday       882.0686 secs
## 10 member        Monday       844.8131 secs
## 11 member        Wednesday    827.9261 secs
## 12 member        Friday       798.6073 secs
## 13 member        Tuesday      787.6345 secs
## 14 member        Thursday     763.3842 secs
```

Summarizing the data

```
complete_data %>%
  group_by(member_casual) %>%
  summarize(count=n(),total_pop=nrow(complete_data),share=count/total_pop*100,mean(ride_length))
```

```
## # A tibble: 2 x 5
##   member_casual  count total_pop share `mean(ride_length)`
##   <chr>          <int>     <int> <dbl> <drtn>
## 1 casual         28247    146446  19.3 1527.8534 secs
## 2 member        118199    146446  80.7  825.1817 secs
```

Visualizing the Data

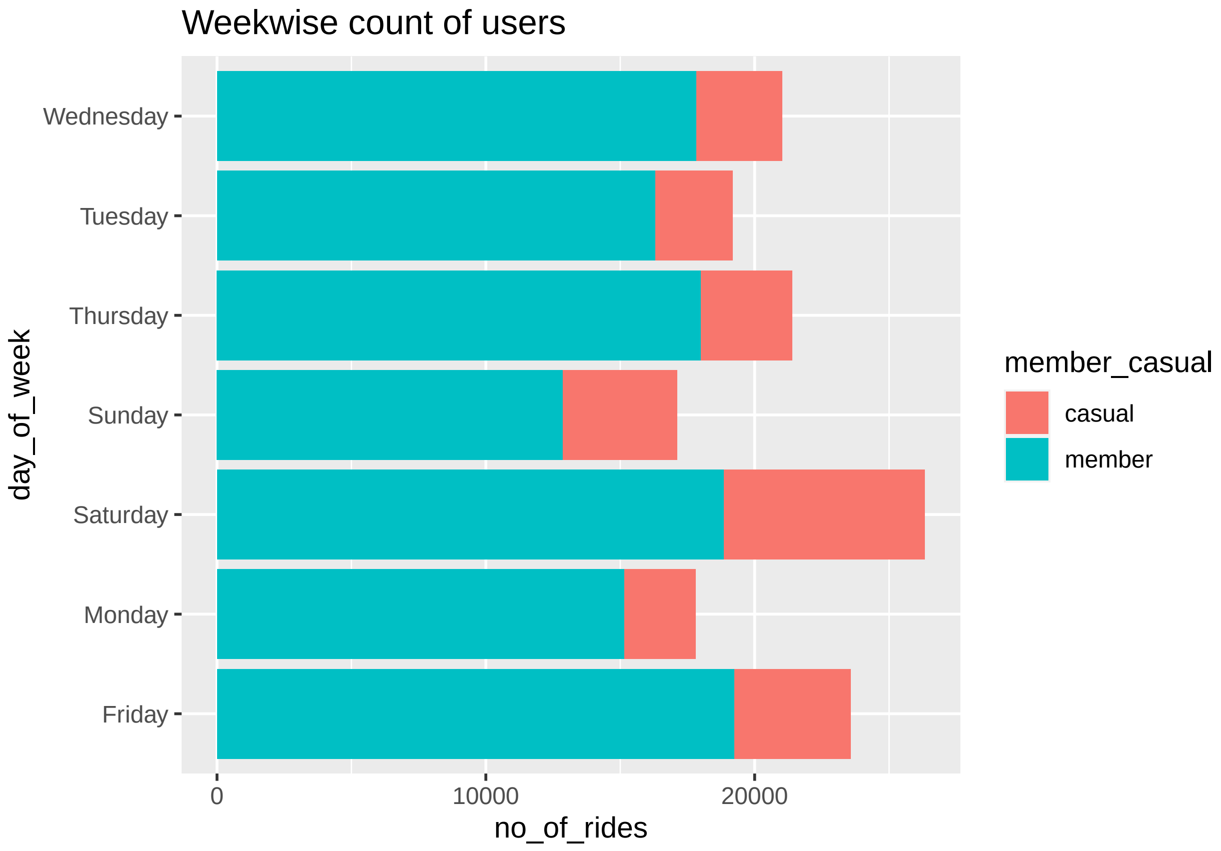
Now we will look the data visually to get a better idea of our analysis.

Weekwise count of users

This data shows us the count of rides in different week days, it is visible from the graph that casual members has slightly higher count than regular on Saturday.

```
complete_data %>%
  group_by(member_casual,day_of_week) %>%
  summarise(no_of_rides= n()) %>%
  ggplot(aes(y=day_of_week,x=no_of_rides, fill=member_casual))+geom_col()+
  labs(title = "Weekwise count of users")
```

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

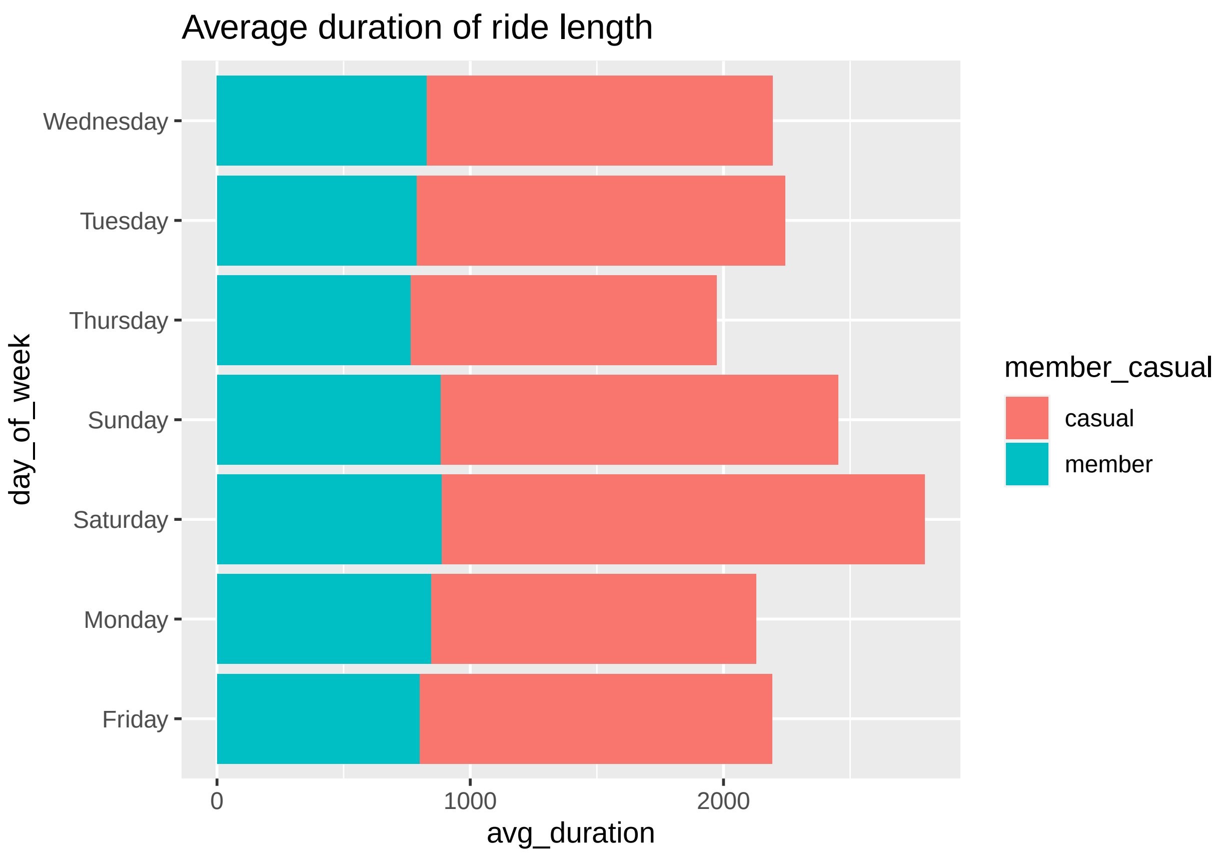


Average ride length of users

This data shows us the average duration of ride length of members and casual users. With the below mentioned visuals it is clear that those who are not members are generally those people who prefer bikes for longer duration.

```
complete_data %>%
  group_by(member_casual,day_of_week) %>%
  summarise(avg_duration = mean(ride_length)) %>%
  ggplot(mapping = aes(x=avg_duration, y=day_of_week, fill=member_casual))+geom_col()+
  labs(title="Average duration of ride length")
```

```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```

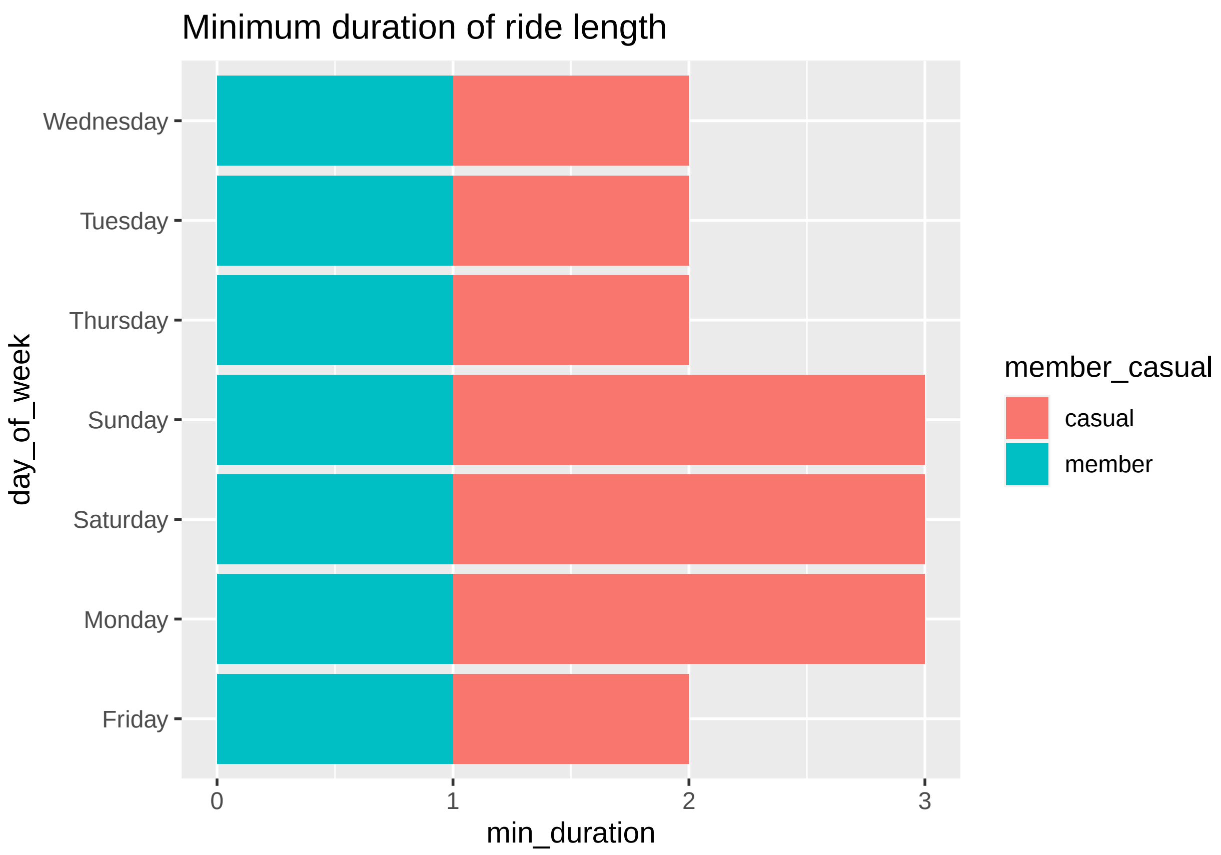


Minimum duration ride length of users

This data shows us the minimum duration of ride length of members and casual users.

```
complete_data %>%
  group_by(member_casual,day_of_week) %>%
  summarise(min_duration = min(ride_length)) %>%
  ggplot(mapping = aes(x=min_duration, y=day_of_week, fill=member_casual))+geom_col()+
  labs(title="Minimum duration of ride length")
```

```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```

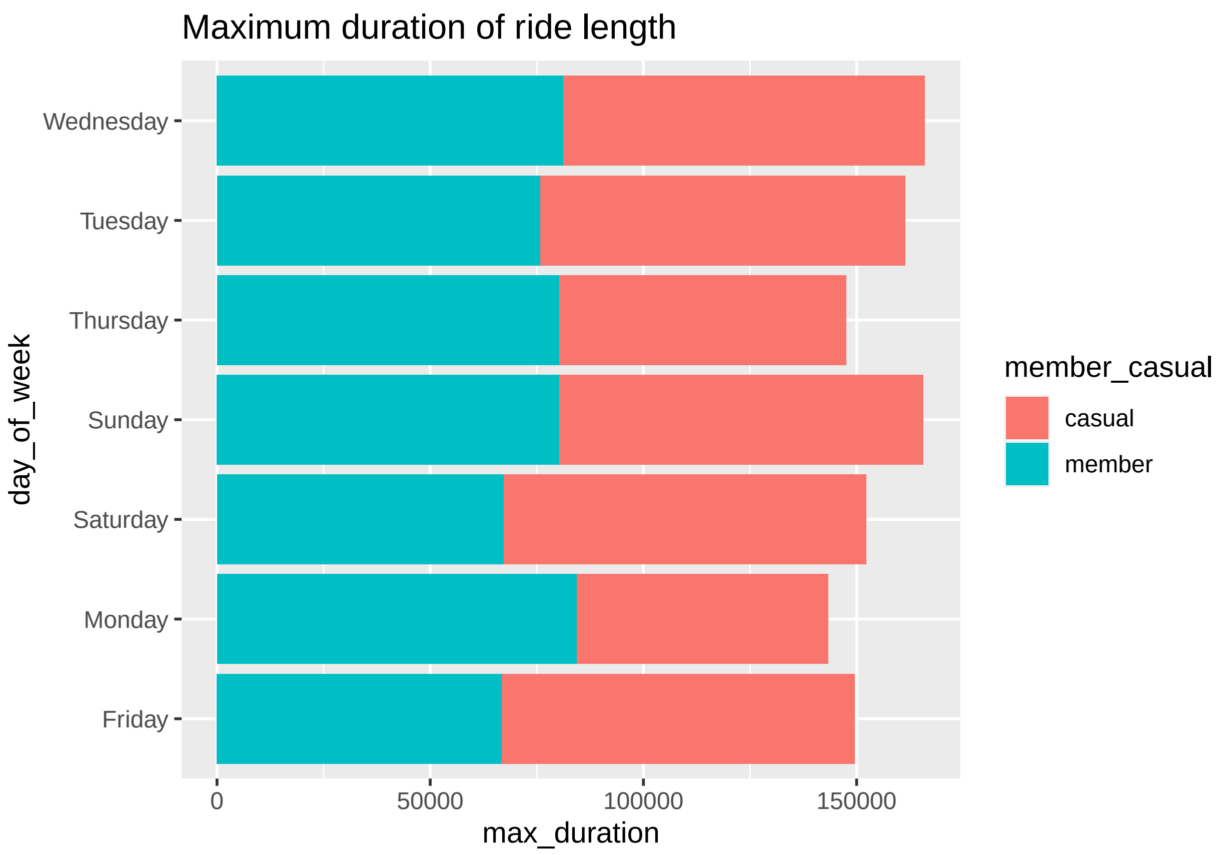


Maximum duration ride length of users

This data shows us the maximum duration of ride length of members and casual users.

```r
complete_data %>%
  group_by(member_casual,day_of_week) %>%
  summarise(max_duration = max(ride_length)) %>%
  ggplot(mapping = aes(x=max_duration, y=day_of_week, fill=member_casual))+geom_col()+
  labs(title="Maximum duration of ride length")
```

```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```


Maximum duration of ride length
Wednesday
Tuesday
Thursday
Sunday
Saturday
Monday
Friday
day_of_week
0
50000
100000
150000
max_duration
member_casual
casual
member