

Data cleaning

Maria Braeuner

2021

```
library(tidyverse)
library(lubridate) #date functions
```

The first part of this document details the steps to clean, transform & prepare the raw data from Cyclistic for analysis. The second part is the initial exploratory analysis. Final analysis/results in CS1_Cyclistic_Report.Rmd.

Preparing

Import Data

#Load Data

```
Nov20 <- read_csv("Data/202011-divvy-tripdata.csv")
Dec20 <- read_csv("Data/202012-divvy-tripdata.csv")
Jan21 <- read_csv("Data/202101-divvy-tripdata.csv")
Feb21 <- read_csv("Data/202102-divvy-tripdata.csv")
Mar21 <- read_csv("Data/202103-divvy-tripdata.csv")
Apr21 <- read_csv("Data/202104-divvy-tripdata.csv")
May21 <- read_csv("Data/202105-divvy-tripdata.csv")
Jun21 <- read_csv("Data/202106-divvy-tripdata.csv")
Jul21 <- read_csv("Data/202107-divvy-tripdata.csv")
Ago21 <- read_csv("Data/202108-divvy-tripdata.csv")
Sep21 <- read_csv("Data/202109-divvy-tripdata.csv")
Oct21 <- read_csv("Data/202110-divvy-tripdata.csv")
```

*#Check all df's have the same structure: check if all column names
#are the same (e.g. would make it easier to merge into one dataframe)*

```
colnames(Nov20) == colnames(Dec20)
colnames(Nov20) == colnames(Jan21)
colnames(Nov20) == colnames(Feb21)
colnames(Nov20) == colnames(Mar21)
colnames(Nov20) == colnames(Apr21)
colnames(Nov20) == colnames(May21)
colnames(Nov20) == colnames(Jun21)
colnames(Nov20) == colnames(Jul21)
colnames(Nov20) == colnames(Ago21)
colnames(Nov20) == colnames(Sep21)
colnames(Nov20) == colnames(Oct21)
```

all TRUE

The data available for each df: ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id, end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual

Merge Data Merge 12 datasets into one.

```
#Merge into one df
cyclistic12 <- rbind(Nov20, Dec20, Jan21, Feb21,
                    Mar21, Apr21, May21, Jun21,
                    Jul21, Ago21, Sep21, Oct21)
```

```
#Check:
head(cyclistic12)
tail(cyclistic12)
dim(cyclistic12) # 5,378,834 x 13
```

Clean & prepare data for analysis

```
#check for data types
glimpse(cyclistic12) #rideable_type & member_casual are as chr, change to factors
cyclistic12$rideable_type <- as.factor(cyclistic12$rideable_type)
cyclistic12$member_casual <- as.factor(cyclistic12$member_casual)
glimpse(cyclistic12) #ok
```

```
summary(cyclistic12)
```

```
##   ride_id          rideable_type      started_at
## Length:5378834   classic_bike :3066970  Min.   :2020-11-01 00:00:08
## Class :character  docked_bike  : 464387  1st Qu.:2021-05-17 12:45:18
## Mode  :character  electric_bike:1847477  Median :2021-07-13 22:33:14
##                                     Mean   :2021-06-27 18:37:41
##                                     3rd Qu.:2021-09-02 18:18:14
##                                     Max.   :2021-10-31 23:59:49
##
##   ended_at          start_station_name start_station_id
## Min.   :2020-11-01 00:02:20  Length:5378834  Length:5378834
## 1st Qu.:2021-05-17 13:07:36  Class :character  Class :character
## Median :2021-07-13 22:57:23  Mode  :character  Mode  :character
## Mean   :2021-06-27 18:58:10
## 3rd Qu.:2021-09-02 18:35:16
## Max.   :2021-11-03 21:45:48
##
##   end_station_name  end_station_id      start_lat      start_lng
## Length:5378834     Length:5378834    Min.   :41.64    Min.   : -87.84
## Class :character   Class :character    1st Qu.:41.88    1st Qu.: -87.66
## Mode  :character   Mode  :character    Median :41.90    Median : -87.64
##                                     Mean   :41.90    Mean   : -87.65
##                                     3rd Qu.:41.93    3rd Qu.: -87.63
##                                     Max.   :42.08    Max.   : -87.52
##
##   end_lat      end_lng      member_casual
## Min.   :41.51  Min.   : -88.07  casual:2470517
## 1st Qu.:41.88  1st Qu.: -87.66  member:2908317
## Median :41.90  Median : -87.64
## Mean   :41.90  Mean   : -87.65
## 3rd Qu.:41.93  3rd Qu.: -87.63
## Max.   :42.17  Max.   : -87.44
## NA's   :4831   NA's   :4831
```

- ended_at includes 3 days of November 2021 (remove)
- end_lat & end_lang have 4831 NA's
- rideable_type categories: classic_bike, docked_bike, electric_bike
- member_casual categories: member & casual

```
length(unique(cyclistic12$start_station_name)) #check how many stations (815)
length(unique(cyclistic12$end_station_name)) #check how many stations (812)

#filter out the extra days of Nov2021
cyclistic <- cyclistic12 %>%
  filter(ended_at <= "2021-11-01 00:00:00")

summary(cyclistic)
dim(cyclistic) #5,378.531 x 13 - ok

#Add new columns for: Day, Month, Year, DayOfWeek (dow), time of day (tod) & tod_char, Season, ride_len.
cyclistic$date <- as.Date(cyclistic$started_at) #yyyy-mm-dd
cyclistic$month <- format(as.Date(cyclistic$date), "%m")
cyclistic$month <- as.numeric(cyclistic$month) #helps to create Seasons column;
cyclistic$day <- format(as.Date(cyclistic$date), "%d")
cyclistic$year <- format(as.Date(cyclistic$date), "%Y")
cyclistic$dow <- format(as.Date(cyclistic$date), "%A")
cyclistic$tod <- format(cyclistic$started_at, "%H:%M:%S")
cyclistic$ride_length <- difftime(cyclistic$ended_at, cyclistic$started_at) #in seconds
cyclistic$ride_length <- as.numeric(cyclistic$ride_length)
cyclistic$ride_length_min <- cyclistic$ride_length/60 #in min

cyclistic$season <- "Winter"
cyclistic$season[cyclistic$month>2&cyclistic$month<6] <- "Spring"
cyclistic$season[cyclistic$month>5&cyclistic$month<9] <- "Summer"
cyclistic$season[cyclistic$month>8&cyclistic$month<12] <- "Autumn"
cyclistic$season <- as.factor(cyclistic$season)
summary(cyclistic$season)

# Reference used here:
##Morning=[05:00-11:59] ; Afternoon=[12:00-17:59] ; Evening=[18:00-21:59] ; Night=[22:00-04:59]

tod_char <- format(cyclistic$started_at, "%H")
tod_char <- as.numeric(tod_char)
cyclistic$tod_char <- "Night"
cyclistic$tod_char[tod_char<12&tod_char>=5] <- "Morning"
cyclistic$tod_char[tod_char>=18&tod_char<22] <- "Evening"
cyclistic$tod_char[tod_char>=12&tod_char<18] <- "Afternoon"
cyclistic$tod_char <- as.factor(cyclistic$tod_char)

summary(cyclistic$tod_char)
glimpse(cyclistic)

summary(cyclistic) #there are negative ride_length vals.

cyclistic[cyclistic$ride_length_min <0, ] # 1393 negatives; started_at & ended_at could be inverted, bu

##Exclude these rows.
Cyclistic_Data <- cyclistic[!cyclistic$ride_length <0, ]
```

```

dim(cyclistic) # 5378531 x 23
dim(Cyclistic_Data) # 5377138 x 23
# 5378531 - 5377138 = 1393 = ok

#ride_id should not have duplicates:
length(Cyclistic_Data$ride_id) #5377138
n_distinct(Cyclistic_Data$ride_id) #5377138 ok!

max(Cyclistic_Data$ride_length_min) #55944.14, this is over 38 days.
#how many ride lengths exceed 24 hours?
Cyclistic_Data %>%
  summarize(weird_lengths = which(ride_length_min>1440))
#there are 3,800 trips of over 24 hours
# who's doing 24+? Check:
long_rides <- Cyclistic_Data %>%
  group_by(member_casual) %>%
  summarize(Long_Ride_Length = which(ride_length_min>1440))

long_rides <- long_rides %>%
  group_by(member_casual) %>%
  summarize(n = n(),
            mean_duration_minutes = mean(Long_Ride_Length))

#Station names have some that were tests and some NAs
##Remove testing rows. Leave NAs for now (some don't have station name but lat,lon)
CyclisticData <- Cyclistic_Data[!grepl("TEST",Cyclistic_Data$start_station_name), ]

CyclisticData <- CyclisticData[!grepl("TEST",CyclisticData$end_station_name), ]

CyclisticData <- CyclisticData[!grepl("TEST",CyclisticData$start_station_id), ]

CyclisticData <- CyclisticData[!grepl("TEST",CyclisticData$end_station_id), ]

rideable_type_check <- CyclisticData %>%
  group_by(month, year) %>%
  select(rideable_type, month, year) %>%
  count(rideable_type)

rideable_type_check

#order dow by dow instead of alphabetically:
CyclisticData$dow <- ordered(CyclisticData$dow, levels=c("Monday",
                                                         "Tuesday",
                                                         "Wednesday",
                                                         "Thursday",
                                                         "Friday",
                                                         "Saturday",
                                                         "Sunday"))

#order season
CyclisticData$season <- ordered(CyclisticData$season,
                                levels=c("Winter",

```

```
"Spring",
"Summer",
"Autumn"))
```

Keep in mind for analysis:

- Rideable_type for November 2020 includes only 2 categories: docked and electric; “classic_bike” appears from December 2020 onward.
- Reference used for “Time of Day” (tod_char):
 - Morning = [05:00-11:59]
 - Afternoon = [12:00-17:59]
 - Evening = [18:00-21:59]
 - Night = [22:00-04:59]
- There are 3,800 trips of over 24 hours still included in the dataset (mean_duration_minutes = mean() of the length of all trips over 1440 minutes (24 hours), in minutes):

```
long_rides
```

```
## # A tibble: 2 x 3
##   member_casual      n mean_duration_minutes
## * <fct>          <int>          <dbl>
## 1 casual          3343          1147415.
## 2 member           457          1443065.
```

Save cleaned data into a new .csv file.

Exploratory Analysis

```
head(CyclisticData)
```

```
## # A tibble: 6 x 23
##   ride_id rideable_type started_at      ended_at      start_station_n-
##   <chr>   <fct>         <dtm>         <dtm>         <chr>
## 1 BD0A6F~ electric_bike 2020-11-01 13:36:00 2020-11-01 13:45:40 Dearborn St & E-
## 2 96A7A7~ electric_bike 2020-11-01 10:03:26 2020-11-01 10:14:45 Franklin St & I-
## 3 C61526~ electric_bike 2020-11-01 00:34:05 2020-11-01 01:03:06 Lake Shore Dr &~
## 4 E533E8~ electric_bike 2020-11-01 00:45:16 2020-11-01 00:54:31 Leavitt St & Ch-
## 5 1C9F4E~ electric_bike 2020-11-01 15:43:25 2020-11-01 16:16:52 Buckingham Foun-
## 6 725958~ electric_bike 2020-11-14 15:55:17 2020-11-14 16:44:38 Wabash Ave & 16-
## # ... with 18 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <fct>, date <date>, month <dbl>, day <chr>,
## #   year <chr>, dow <ord>, tod <chr>, ride_length <dbl>, ride_length_min <dbl>,
## #   season <ord>, tod_char <fct>
```

How do casual customers and members differ in # of rides & average ride duration (in minutes)?

```
## Ride length ($ride_length &/or $ride_length_min)
```

```
CyclisticData %>%
  group_by(member_casual) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min))
```

```
## # A tibble: 2 x 3
##   member_casual no_rides avg_duration_min
```

```
## * <fct>          <int>          <dbl>
## 1 casual        2469725         31.8
## 2 member        2906470         14.0
```

How do casual customers and members differ in # of rides & average ride duration (in minutes) by ride_type used?

```
## Type of ride they use ($rideable_type)
CyclisticData %>%
  group_by(member_casual, rideable_type) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min))
```

```
## # A tibble: 6 x 4
## # Groups:   member_casual [2]
##   member_casual rideable_type no_rides avg_duration_min
##   <fct>         <fct>         <int>         <dbl>
## 1 casual       classic_bike    1226182        29.0
## 2 casual       docked_bike    350384         70.7
## 3 casual       electric_bike  893159         20.5
## 4 member       classic_bike   1840398         14.3
## 5 member       docked_bike   112700          14.2
## 6 member       electric_bike  953372          13.2
```

How do casual customers and members differ in # of rides & average ride duration (in minutes) by season?

```
## Use by $season
CyclisticData %>%
  group_by(member_casual, season) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min))
```

```
## # A tibble: 8 x 4
## # Groups:   member_casual [2]
##   member_casual season no_rides avg_duration_min
##   <fct>         <ord>     <int>         <dbl>
## 1 casual       Winter    58242         30.2
## 2 casual       Spring   477477         37.0
## 3 casual       Summer   1225272        32.1
## 4 casual       Autumn   708734         28.1
## 5 member       Winter   219297         13.7
## 6 member       Spring   619733         14.5
## 7 member       Summer   1130616        14.3
## 8 member       Autumn   936824         13.2
```

How do casual customers and members differ in # of rides & average ride duration (in minutes) by time of day?

```
## Use by $tod ($tod_char)
CyclisticData %>%
  group_by(member_casual, tod_char) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min))
```

```
## # A tibble: 8 x 4
## # Groups:   member_casual [2]
##   member_casual tod_char no_rides avg_duration_min
##   <fct>         <fct>     <int>         <dbl>
```

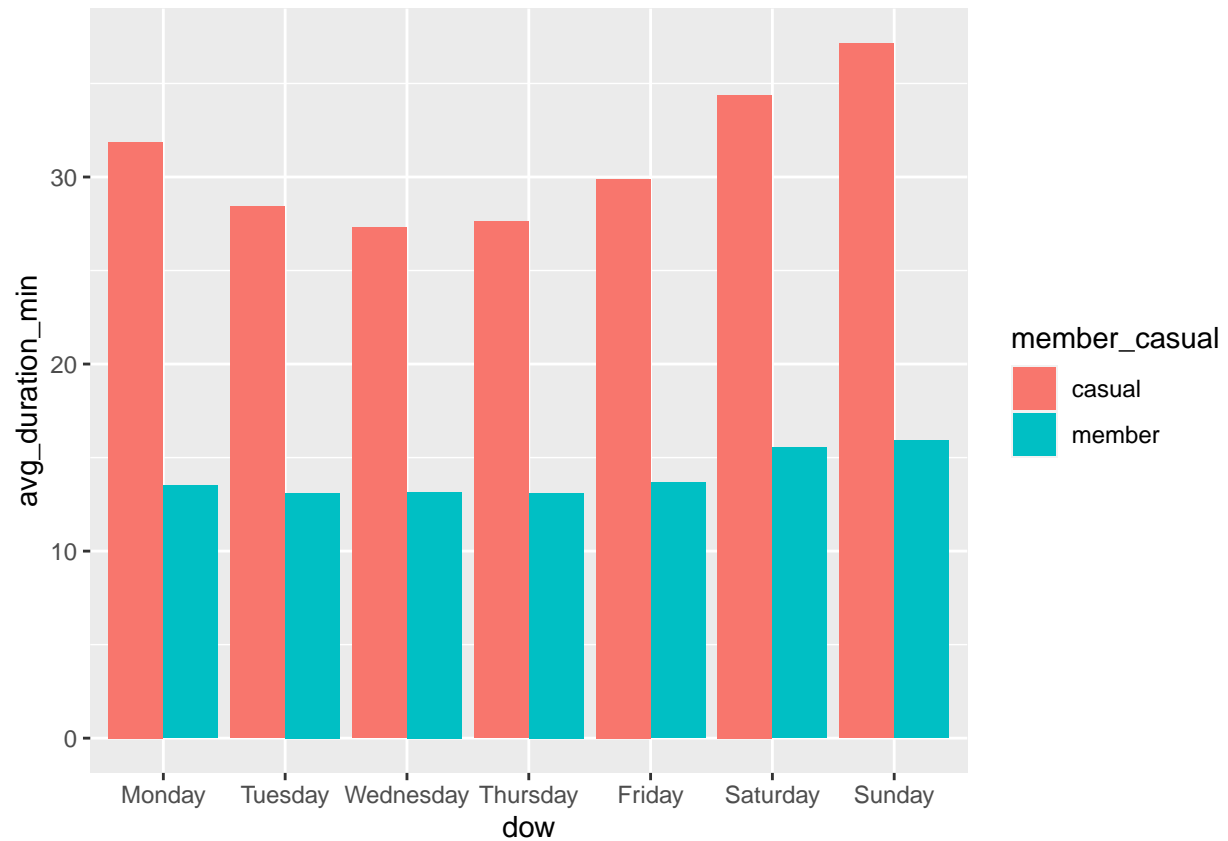
```
## 1 casual      Afternoon 1126975      32.7
## 2 casual      Evening   590029      30.6
## 3 casual      Morning   450933      29.3
## 4 casual      Night     301788      34.7
## 5 member      Afternoon 1256233      14.4
## 6 member      Evening   658254      14.2
## 7 member      Morning   799152      13.1
## 8 member      Night     192831      14.3
```

How do casual customers and members differ in # of rides & average ride duration (in minutes) by day of the week?

```
## Use by $dow
CyclisticData %>%
  group_by(member_casual, dow) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min))
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##   member_casual dow      no_rides avg_duration_min
##   <fct>         <ord>      <int>      <dbl>
## 1 casual      Monday      278216      31.9
## 2 casual      Tuesday     264197      28.4
## 3 casual      Wednesday   267398      27.3
## 4 casual      Thursday    277291      27.6
## 5 casual      Friday      354923      29.9
## 6 casual      Saturday    551752      34.4
## 7 casual      Sunday      475948      37.1
## 8 member      Monday      391174      13.5
## 9 member      Tuesday     431387      13.1
## 10 member     Wednesday   444184      13.2
## 11 member     Thursday    425458      13.1
## 12 member     Friday      425010      13.7
## 13 member     Saturday    421090      15.5
## 14 member     Sunday      368167      15.9
```

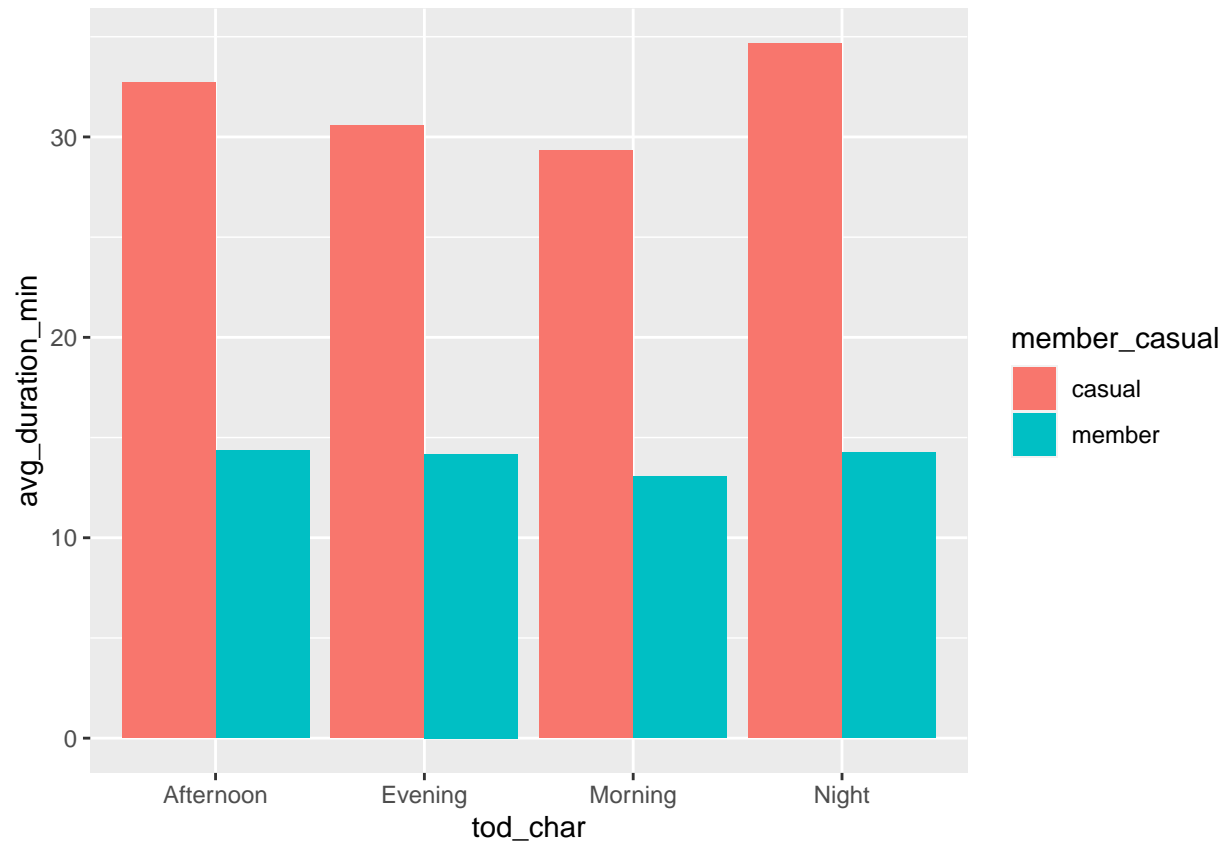
```
CyclisticData %>%
  group_by(member_casual, dow) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min)) %>%
  ggplot(mapping = aes(x = dow, y = avg_duration_min, fill = member_casual)) +
  geom_col(position="dodge")
```



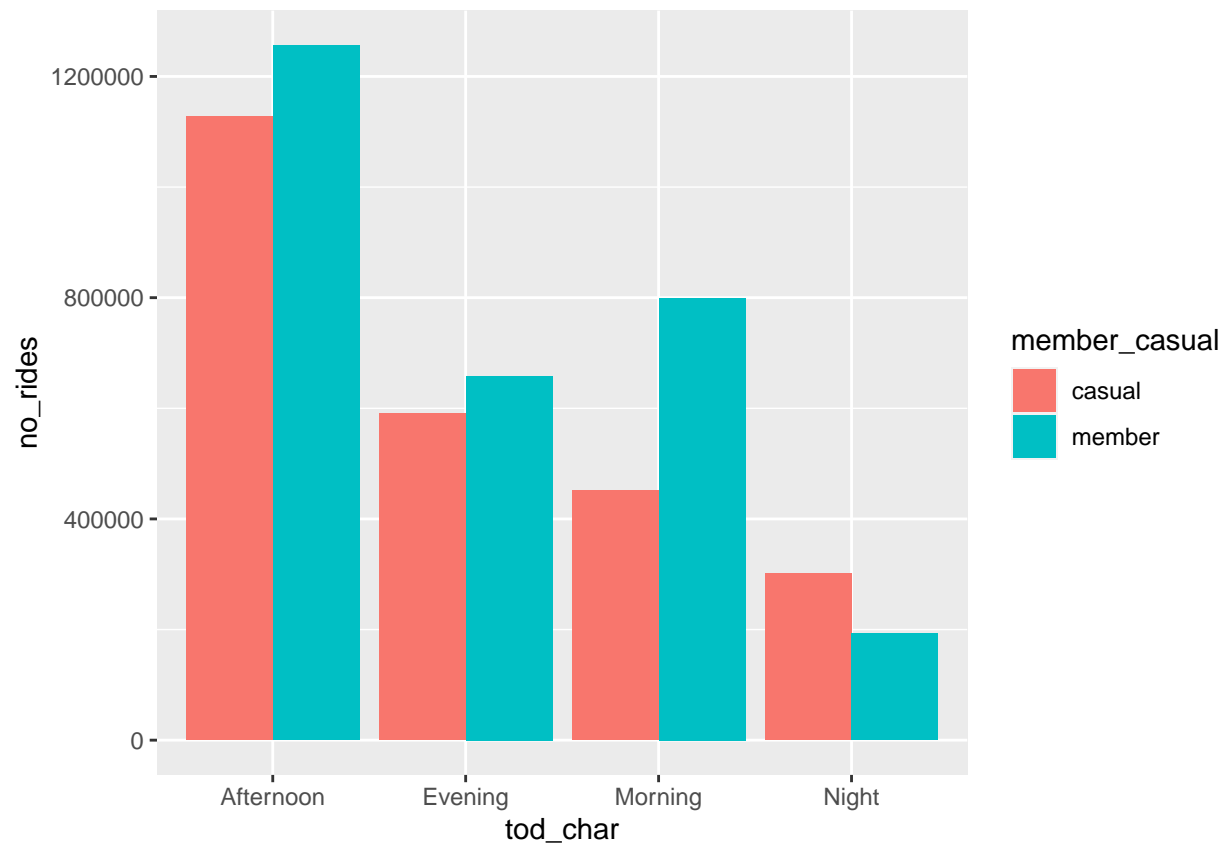
```
CyclisticData %>%
  group_by(member_casual, dow) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min)) %>%
  ggplot(mapping = aes(x = dow, y = no_rides, fill = member_casual)) +
  geom_col(position="dodge")
```




```
CyclisticData %>%
  group_by(member_casual, tod_char) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min)) %>%
  ggplot(mapping = aes(x = tod_char, y = avg_duration_min, fill = member_casual)) +
  geom_col(position="dodge")
```



```
CyclisticData %>%  
  group_by(member_casual, tod_char) %>%  
  summarise(no_rides = n(),  
            avg_duration_min = mean(ride_length_min)) %>%  
  ggplot(mapping = aes(x = tod_char, y = no_rides, fill = member_casual)) +  
  geom_col(position="dodge")
```



```
CyclisticData %>%
  group_by(member_casual, season) %>%
  summarise(no_rides = n(),
            avg_duration_min = mean(ride_length_min)) %>%
  ggplot(mapping = aes(x = season, y = no_rides, fill = member_casual)) +
  geom_col(position="dodge")
```

