Sample Case Study: Cyclistic

Allan Lam

2025-09-12

Setting up my environment

Notes: setting up my R environment by loading the 'tidyverse' and 'ggplot2' packages

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

Processing csv files

```
setwd("~/Desktop/Divvy_Trips")

Q1_2019 = read.csv("/Users/allanlam/Desktop/Divvy_Trips/Divvy_Trips_2019_Q1.csv") %>% mutate(year = "20
glimpse(Q1_2019)
```

```
## Rows: 365,069
## Columns: 15
## $ trip_id           <int> 21742443, 21742444, 21742445, 21742446, 21742447, 21~
## $ start_time        <chr> "1/1/19 0:04", "1/1/19 0:08", "1/1/19 0:13", "1/1/19~
## $ end_time          <chr> "1/1/19 0:11", "1/1/19 0:15", "1/1/19 0:27", "1/1/19~
## $ bikeid            <int> 2167, 4386, 1524, 252, 1170, 2437, 2708, 2796, 6205,~
## $ tripduration      <chr> "390", "441", "829", "1,783.00", "364", "216", "177"~
## $ from_station_id   <int> 199, 44, 15, 123, 173, 98, 98, 211, 150, 268, 299, 2~
## $ from_station_name <chr> "Wabash Ave & Grand Ave", "State St & Randolph St", ~
## $ to_station_id     <int> 84, 624, 644, 176, 35, 49, 49, 142, 148, 141, 295, 4~
## $ to_station_name   <chr> "Milwaukee Ave & Grand Ave", "Dearborn St & Van Bure~
## $ usertype          <chr> "Subscriber", "Subscriber", "Subscriber", "Subscribe~
## $ gender            <chr> "Male", "Female", "Female", "Male", "Male", "Female"~
## $ birthyear         <int> 1989, 1990, 1994, 1993, 1994, 1983, 1984, 1990, 1995~
## $ ride_length       <dbl> 0.004513889, 0.005104167, 0.009594907, 0.020636574, ~
## $ day_of_week       <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3~
## $ year              <chr> "2019", "2019", "2019", "2019", "2019", "2019", "201~
```

```
Q1_2020 = read.csv("/Users/allanlam/Desktop/Divvy_Trips/Divvy_Trips_2020_Q1.csv") %>% mutate(year = "202
glimpse(Q1_2020)
```

```
## Rows: 426,887
## Columns: 16
## $ ride_id            <chr> "EACB19130B0CDA4A", "8FED874C809DC021", "789F3C21E4~
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", "docke~
## $ started_at         <chr> "1/21/20 20:06", "1/30/20 14:22", "1/9/20 19:29", "~
## $ ended_at           <chr> "1/21/20 20:14", "1/30/20 14:26", "1/9/20 19:32", "~
## $ start_station_name <chr> "Western Ave & Leland Ave", "Clark St & Montrose Av~
## $ start_station_id   <int> 239, 234, 296, 51, 66, 212, 96, 96, 212, 38, 117, 1~
## $ end_station_name   <chr> "Clark St & Leland Ave", "Southport Ave & Irving Pa~
## $ end_station_id     <int> 326, 318, 117, 24, 212, 96, 212, 212, 96, 100, 632,~
## $ start_lat          <dbl> 41.9665, 41.9616, 41.9401, 41.8846, 41.8856, 41.889~
## $ start_lng          <dbl> -87.6884, -87.6660, -87.6455, -87.6319, -87.6418, -~
## $ end_lat            <dbl> 41.9671, 41.9542, 41.9402, 41.8918, 41.8899, 41.884~
## $ end_lng            <dbl> -87.6674, -87.6644, -87.6530, -87.6206, -87.6343, -~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
## $ ride_length        <chr> "0.005219907", "0.002581019", "0.001979167", "0.006~
## $ day_of_week        <int> 3, 5, 5, 2, 5, 6, 6, 6, 6, 6, 3, 4, 4, 5, 3, 3, 3, ~
## $ year               <chr> "2020", "2020", "2020", "2020", "2020", "2020", "20~
```

```
#change to numeric
Q1_2019$ride_length = as.numeric(Q1_2019$ride_length)
Q1_2020$ride_length = as.numeric(Q1_2020$ride_length)
```

```
## Warning: NAs introduced by coercion
```

```
#check for NAs
any((is.na(Q1_2019$ride_length)))
```

```
## [1] FALSE
```

```r
any((is.na(Q1_2020$ride_length)))
```

```
## [1] TRUE
```

```r
Q1_2020 = na.omit(Q1_2020) #Trips with 0 seconds, so omit for conciseness

#edit consistent naming for membership type
Q1_2020 = Q1_2020 %>%
  mutate(usertype = case_when(
    member_casual == "casual" ~ "Customer",
    member_casual == "member" ~ "Subscriber"
  ))
#check for NAs
any(is.na(Q1_2020$member_casual))
```

```
## [1] FALSE
```

```r
#rename column names for consistency
Q1_2020_renamed = Q1_2020 %>%
  rename (
    trip_id = ride_id,
    start_time = started_at,
    end_time = ended_at,
    from_station_name = start_station_name,
    to_station_name = end_station_name
  )

#select columns of interest for merging
Q1_2019_trimmed = Q1_2019 %>% select(trip_id, start_time, end_time, from_station_name, to_station_name,
Q1_2019_trimmed$trip_id = as.character(Q1_2019_trimmed$trip_id) #change to character for merging
Q1_2020_trimmed = Q1_2020_renamed %>% select(trip_id, start_time, end_time, from_station_name, to_statio
Q1_agg_trimmed = full_join(Q1_2019_trimmed, Q1_2020_trimmed)
```

```
## Joining with ‘by = join_by(trip_id, start_time, end_time, from_station_name,
## to_station_name, usertype, ride_length, day_of_week, year)‘
```

```r
#label day_of_week into weekdays
Q1_agg_trimmed = Q1_agg_trimmed %>%  mutate (day_of_week =
                              case_when(
                             day_of_week == "1" ~"Sunday",
                             day_of_week == "2" ~"Monday",
                             day_of_week == "3" ~"Tuesday",
                             day_of_week == "4" ~"Wednesday",
                             day_of_week == "5" ~"Thursday",
                             day_of_week == "6" ~"Friday",
                             day_of_week == "7" ~"Saturday"))
Q1_agg_trimmed$day_of_week = factor(
  Q1_agg_trimmed$day_of_week,
  levels = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")
)
```

```r
#clean data format for ride_length into minutes and seconds
Q1_agg_trimmed = Q1_agg_trimmed %>%
  mutate(ride_length_sec = round(ride_length * 24 * 60 * 60, 2),
         ride_length_min = round(ride_length * 24 * 60, 2),
         total_ride_trips =  n()
         )
glimpse(Q1_agg_trimmed)
```

```
## Rows: 791,839
## Columns: 12
## $ trip_id           <chr> "21742443", "21742444", "21742445", "21742446", "217~
## $ start_time        <chr> "1/1/19 0:04", "1/1/19 0:08", "1/1/19 0:13", "1/1/19~
## $ end_time          <chr> "1/1/19 0:11", "1/1/19 0:15", "1/1/19 0:27", "1/1/19~
## $ from_station_name <chr> "Wabash Ave & Grand Ave", "State St & Randolph St", ~
## $ to_station_name   <chr> "Milwaukee Ave & Grand Ave", "Dearborn St & Van Bure~
## $ usertype          <chr> "Subscriber", "Subscriber", "Subscriber", "Subscribe~
## $ ride_length       <dbl> 0.004513889, 0.005104167, 0.009594907, 0.020636574, ~
## $ day_of_week       <fct> Tuesday, Tuesday, Tuesday, Tuesday, Tuesday, Tuesday~
## $ year              <chr> "2019", "2019", "2019", "2019", "2019", "2019", "201~
## $ ride_length_sec   <dbl> 390, 441, 829, 1783, 364, 216, 177, 100, 1727, 336, ~
## $ ride_length_min   <dbl> 6.50, 7.35, 13.82, 29.72, 6.07, 3.60, 2.95, 1.67, 28~
## $ total_ride_trips  <int> 791839, 791839, 791839, 791839, 791839, 791839, 7918~
```

```r
#check for NAs
any(is.na(Q1_agg_trimmed$ride_length_sec))
```

```
## [1] FALSE
```

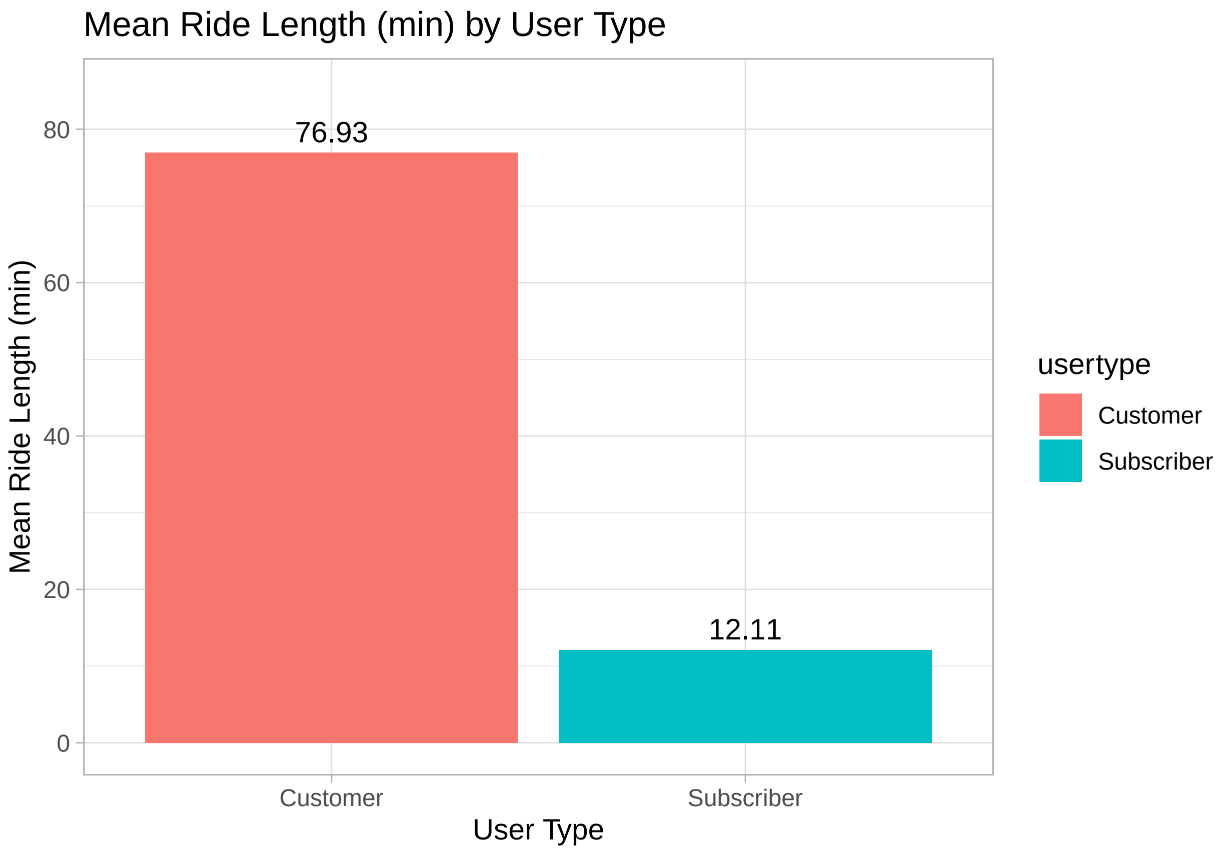
```r
any(is.na(Q1_agg_trimmed$ride_length_min))
```

```
## [1] FALSE
```

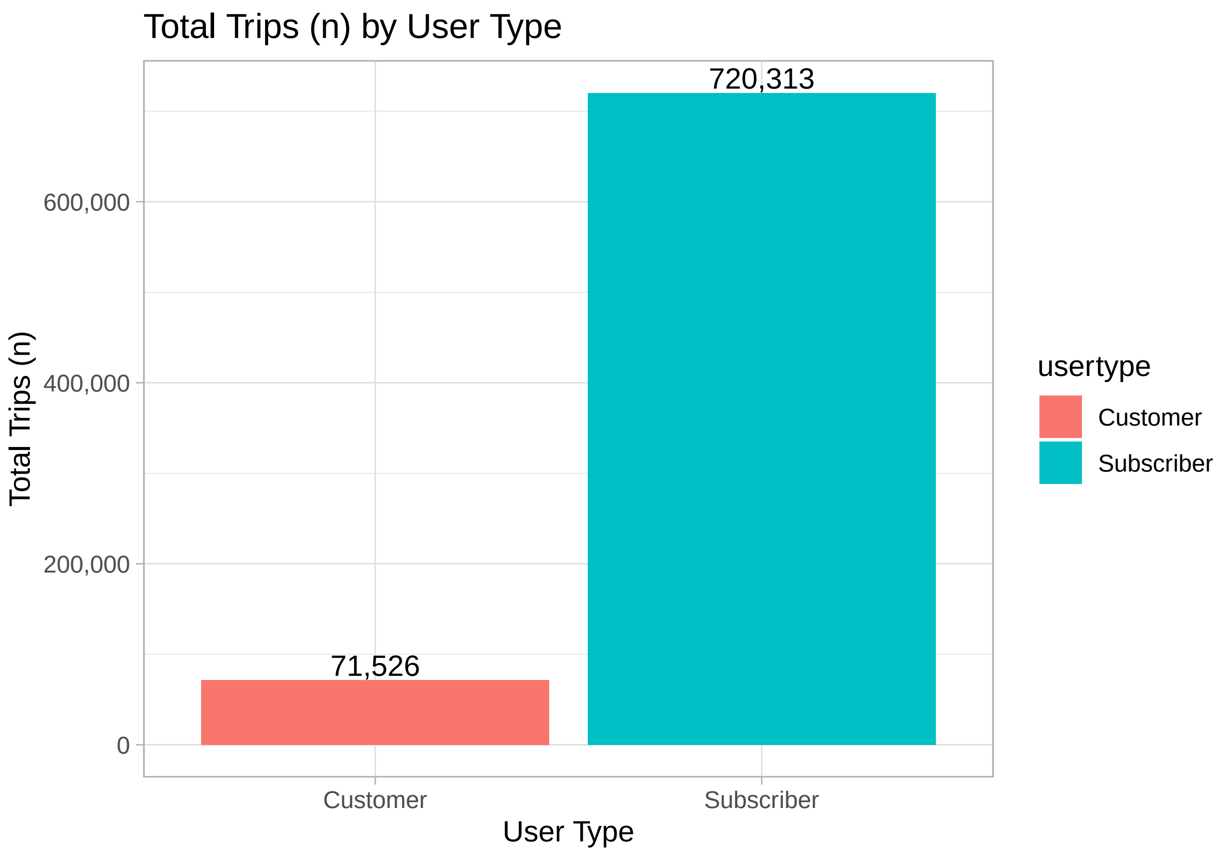
```r
# write_csv(Q1_agg_trimmed, "Q1_agg_trimmed.csv")
```

Data analysis

```r
# Calculate the descriptive stats of the data (by user type)
Q1_summary = Q1_agg_trimmed %>%
  group_by(usertype) %>%
  summarize(group_trips = n(),
            mean_ride_length_sec = mean(ride_length_sec, na.rm = TRUE),
            max_ride_length_sec = max(ride_length_sec, na.rm = TRUE),
            min_ride_length_sec = min(ride_length_sec, na.rm = TRUE),
            sd_ride_length_sec = sd(ride_length_sec, na.rm = TRUE),

            mean_ride_length_min = mean(ride_length_min, na.rm = TRUE),
            max_ride_length_min = max(ride_length_min, na.rm = TRUE),
            min_ride_length_min = min(ride_length_min, na.rm = TRUE),
            sd_ride_length_min = sd(ride_length_min, na.rm = TRUE),
            # find the most frquent week of usage
            mode_day_of_week = names(sort(table(day_of_week), decreasing = TRUE))[1])

# Bar plot: Mean ride length by user type
ggplot(data = Q1_summary, aes(x= usertype, y= mean_ride_length_min, fill = usertype)) +
  geom_col() +
  geom_text(aes(label = round(mean_ride_length_min,2)), vjust = -0.5) +
  labs(title = "Mean Ride Length (min) by User Type", y = "Mean Ride Length (min)", x = "User Type") +
  ylim(0,85) +
  theme_light()
```



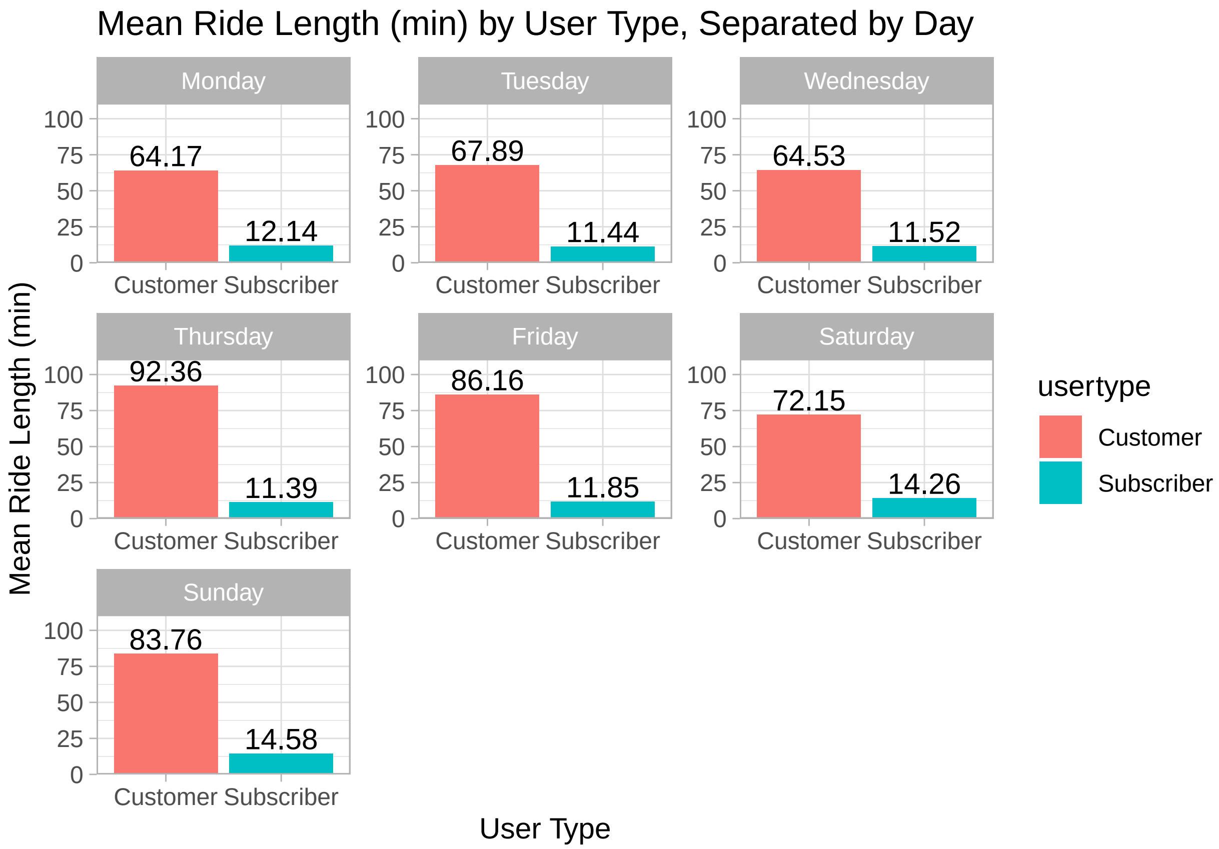
```r
# Bar plot: Total trips by user type
ggplot(data = Q1_summary, aes(x= usertype, y= group_trips, fill = usertype)) +
  geom_col() +
  geom_text(aes(label = scales::comma(group_trips)), vjust = -0.2) +
  labs(title = "Total Trips (n) by User Type", y = "Total Trips (n)", x = "User Type") +
  scale_y_continuous(labels = comma) +
  theme_light()
```



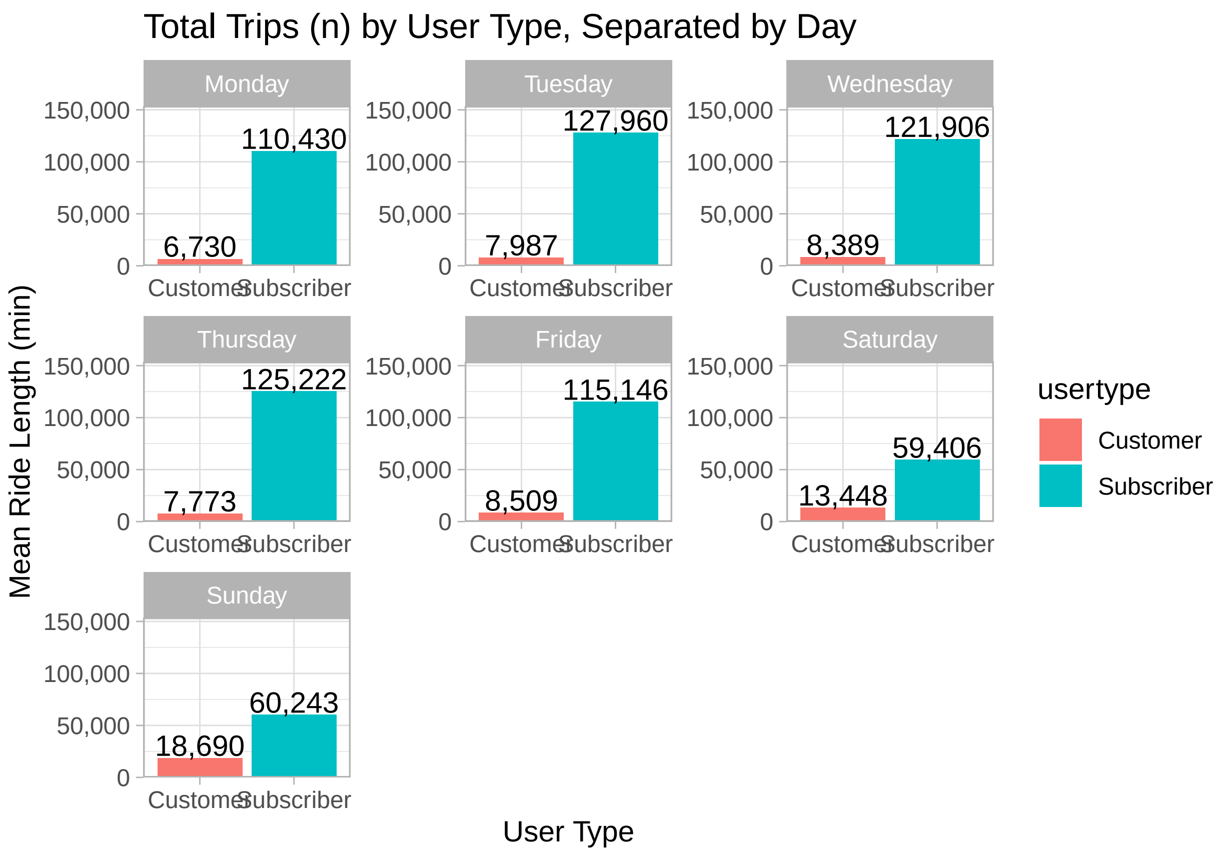
```
# Calculate the descriptive stats of the data (by user type and day of week)
Q1_summary_by_days = Q1_agg_trimmed %>%
  group_by(usertype, day_of_week) %>%
  summarize(group_trips = n(),
            mean_ride_length_sec = mean(ride_length_sec, na.rm = TRUE),
            max_ride_length_sec = max(ride_length_sec, na.rm = TRUE),
            min_ride_length_sec = min(ride_length_sec, na.rm = TRUE),
            sd_ride_length_sec = sd(ride_length_sec, na.rm = TRUE),

            mean_ride_length_min = mean(ride_length_min, na.rm = TRUE),
            max_ride_length_min = max(ride_length_min, na.rm = TRUE),
            min_ride_length_min = min(ride_length_min, na.rm = TRUE),
            sd_ride_length_min = sd(ride_length_min, na.rm = TRUE))
```

```
## ‘summarise()‘ has grouped output by ’usertype’. You can override using the
## ‘.groups‘ argument.
```

```
# Bar plot: Mean Ride Length by User Type, Faceted by Day
ggplot(data = Q1_summary_by_days, aes(x= usertype, y= mean_ride_length_min, fill = usertype)) +
  geom_col() +
  geom_text(aes(label = round(mean_ride_length_min,2)), vjust = -0.2) +
  facet_wrap(~ fct_relevel(day_of_week,
                           "Monday", "Tuesday", "Wednesday",
                           "Thursday", "Friday", "Saturday", "Sunday"), axes = "all") +
  labs(title = "Mean Ride Length (min) by User Type, Separated by Day", y = "Mean Ride Length (min)", x
  scale_y_continuous(expand = expansion(mult = c(0, 0.2)))+
  theme_light()
```



```
# Bar plot: Total Trips by User Type, Faceted by Day
ggplot(data = Q1_summary_by_days, aes(x= usertype, y= group_trips, fill = usertype)) +
  geom_col() +
  geom_text(aes(label = scales::comma(group_trips)), vjust = -0.1) +
   facet_wrap(~ fct_relevel(day_of_week,
                             "Monday", "Tuesday", "Wednesday",
                             "Thursday", "Friday", "Saturday", "Sunday"), axes = "all") +
  labs(title = "Total Trips (n) by User Type, Separated by Day", y = "Mean Ride Length (min)", x = "Use
  scale_y_continuous(labels = comma,expand = expansion(mult = c(0, 0.2)))+
  theme_light()
```



Statistical Modeling and Analysis

```r
# Drop extreme outliers
Q1_agg_trimmed = subset(Q1_agg_trimmed, ride_length_min > 0 & ride_length_min < 300)

# Generalized Linear Model with Gamma + log link (ride_length_min ~ usertype)
glm_model = glm(ride_length_min ~ usertype, data = Q1_agg_trimmed,
                 family = Gamma(link = "log"))
summary(glm_model)
```

```
##
## Call:
## glm(formula = ride_length_min ~ usertype, family = Gamma(link = "log"),
##     data = Q1_agg_trimmed)
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.429897   0.003365  1019.3   <2e-16 ***
## usertypeSubscriber -1.030570   0.003526  -292.3   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for Gamma family taken to be 0.7989199)
##
```

```
##     Null deviance: 554330  on 790353  degrees of freedom
## Residual deviance: 463888  on 790352  degrees of freedom
## AIC: 5372470
##
## Number of Fisher Scoring iterations: 6
```

```
glm_coef = coef(glm_model)
coef_exp = exp(glm_coef) #exp transform
glm_p = summary(glm_model)$coefficients[,4]
customer_mean_glm = coef_exp["(Intercept)"]
subscriber_ratio = coef_exp["usertypeSubscriber"]

# Print result sentence for glm
cat("Generalized Linear Model (Gamma with log link):\n")
```

```
## Generalized Linear Model (Gamma with log link):
```

```
cat("Average ride length for Customers is about", round(customer_mean_glm, 2), "minutes.\n")
```

```
## Average ride length for Customers is about 30.87 minutes.
```

```
if (subscriber_ratio < 1) {
  cat("Subscribers ride about", round(subscriber_ratio, 2),
      "times as long as Customers (i.e. shorter).\n")
} else {
  cat("Subscribers ride about", round(subscriber_ratio, 2),
      "times as long as Customers (i.e. longer).\n")
}
```

```
## Subscribers ride about 0.36 times as long as Customers (i.e. shorter).
```

```
cat("The p-value for this effect is", format.pval(glm_p["usertypeSubscriber"], digits = 3, scientific =
```

```
## The p-value for this effect is <0.0000000000000002
```

```
# Generalized Linear Model with Gamma + log link (group_trips ~ usertype)
glm_model = glm(group_trips ~ usertype, data = Q1_summary,
                 family = poisson(link = "log"))
summary(glm_model)
```

```
##
## Call:
## glm(formula = group_trips ~ usertype, family = poisson(link = "log"),
##     data = Q1_summary)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        11.177816   0.003739  2989.4   <2e-16 ***
## usertypeSubscriber  2.309625   0.003920   589.1   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance:  6.1740e+05  on 1  degrees of freedom
## Residual deviance: -1.1197e-10  on 0  degrees of freedom
## AIC: 32.341
##
## Number of Fisher Scoring iterations: 2
```

```
glm_coef = coef(glm_model)
coef_exp = exp(glm_coef) #exp transform
glm_p = summary(glm_model)$coefficients[,4]
customer_mean_glm = coef_exp["(Intercept)"]
subscriber_ratio = coef_exp["usertypeSubscriber"]

# Print result sentence for glm
cat("Generalized Linear Model (Gamma with log link):\n")
```

```
## Generalized Linear Model (Gamma with log link):
```

```
cat("Average number of trips for Customers is about", round(customer_mean_glm, 2), "trips.\n")
```

```
## Average number of trips for Customers is about 71526 trips.
```

```
if (subscriber_ratio < 1) {
  cat("Subscribers ride about", round(subscriber_ratio, 2),
      "times total trips as Customers (i.e. less).\n")
} else {
  cat("Subscribers ride about", round(subscriber_ratio, 2),
      "times total trips as Customers (i.e. more).\n")
}
```

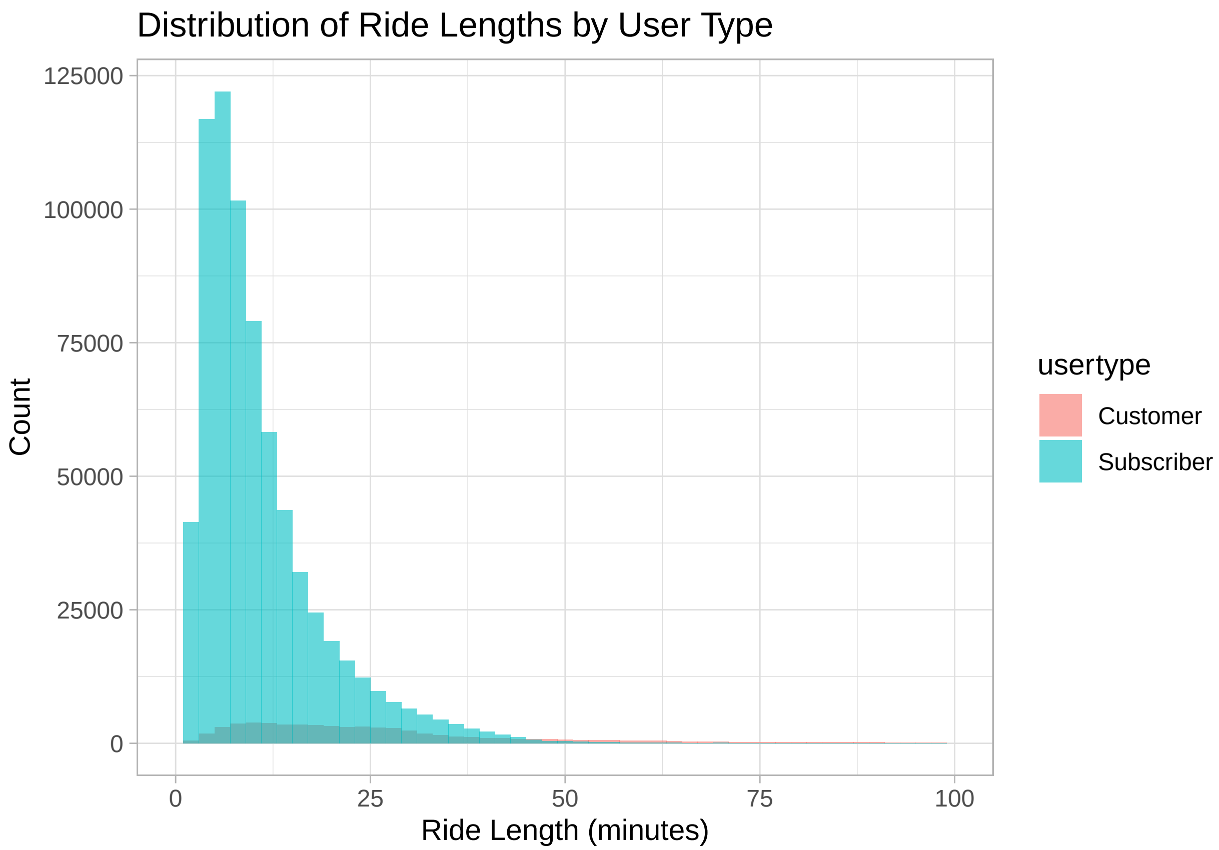
```
## Subscribers ride about 10.07 times total trips as Customers (i.e. more).
```

```
cat("The p-value for this effect is", format.pval(glm_p["usertypeSubscriber"], digits = 3, scientific =
```

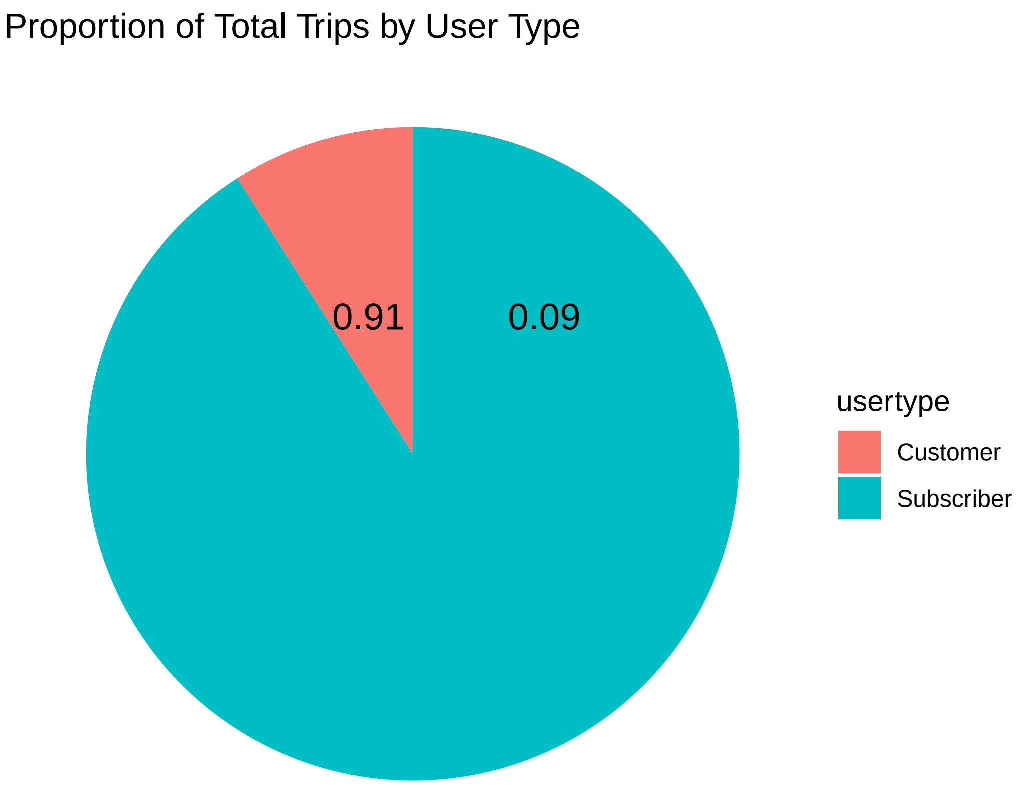
```
## The p-value for this effect is <0.0000000000000002
```

```
# Histogram: Distribution of Ride Lengths by User Type
ggplot(Q1_agg_trimmed, aes(x = ride_length_min, fill = usertype)) +
  geom_histogram(binwidth = 2, alpha = 0.6, position = "identity") +
  xlim(0, 100) +   # trim extreme rides
  labs(title = "Distribution of Ride Lengths by User Type",
       x = "Ride Length (minutes)", y = "Count")+
  theme_light()
```

```
## Warning: Removed 3764 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
## Warning: Removed 4 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```



```r
# Pie chart: Proportion of Total Trips by User Type
Q1_summary %>%
  mutate(prop = group_trips / sum(group_trips)) %>%
  ggplot(aes(x = "", y = prop, fill = usertype)) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_text(aes(label = round(prop,2)), hjust = -0.1, size = 5) +
  coord_polar(theta = "y") +
  labs(title = "Proportion of Total Trips by User Type") +
  theme_void()
```


Proportion of Total Trips by User Type
0.91
0.09
usertype
Customer
Subscriber