jumpman.r

riserate

Wed Sep 13 17:08:26 2017

```
#load need libraries
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(geosphere)
```

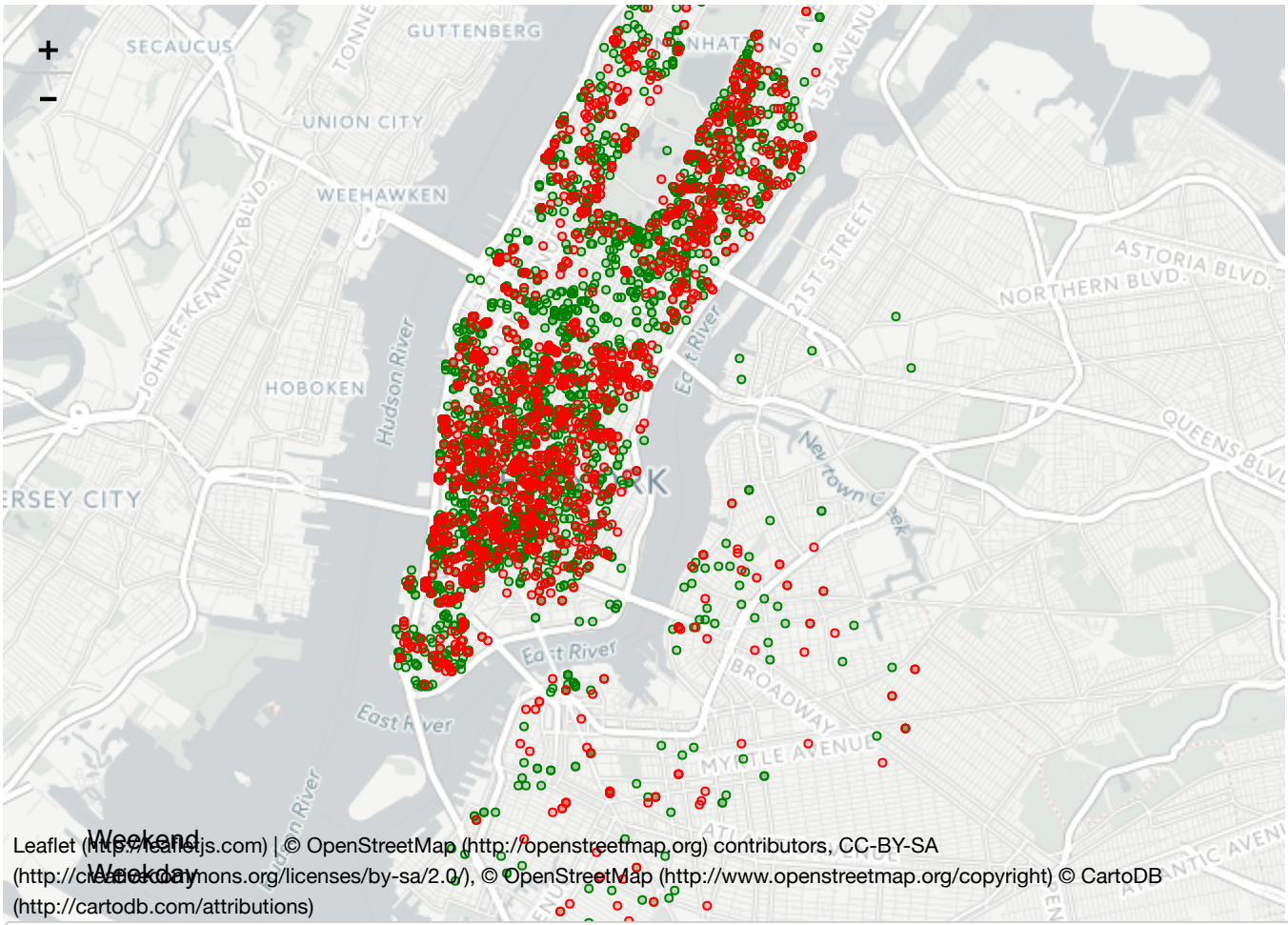
```
## Loading required package: sp
```

```
library(leaflet)
library(ggplot2)
#import the data set
df <- read.csv("~/r-code/jumpmen/CompanyData.csv",stringsAsFactors = FALSE)
head(df)
```

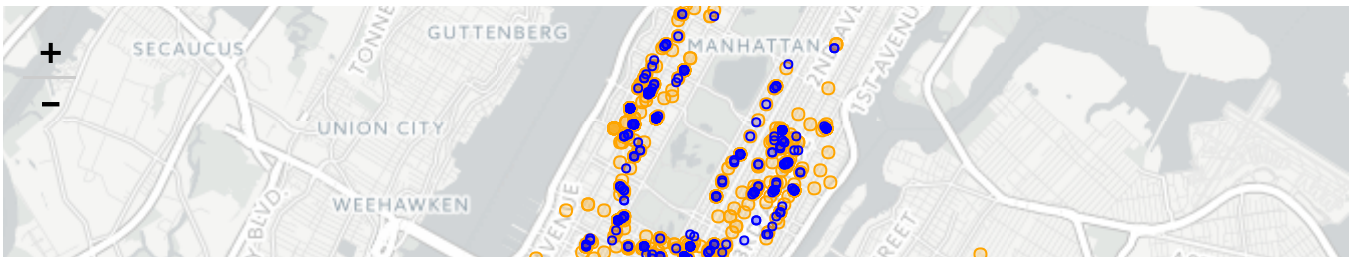
```
##   delivery_id customer_id jumpman_id vehicle_type        pickup_place
## 1     1457973      327168     162381          van           Melt Shop
## 2     1377056       64452     104533      bicycle Prince Street Pizza
## 3     1476547       83095     132725      bicycle          Bareburger
## 4     1485494      271149     157175      bicycle         Juice Press
## 5     1327707      122609     118095      bicycle   Blue Ribbon Sushi
## 6     1423142       75169      91932      bicycle    Tamarind TriBeCa
##   place_category                  item_name item_quantity
## 1       American                  Lemonade             1
## 2          Pizza       Neapolitan Rice Balls             3
## 3         Burger                  Bare Sodas             1
## 4      Juice Bar     OMG! My Favorite Juice!             1
## 5       Japanese Spicy Tuna & Tempura Flakes             2
## 6         Indian               Dum Aloo Gobi             1
##       item_category_name how_long_it_took_to_order pickup_lat pickup_lon
## 1              Beverages           00:19:58.582052   40.74461  -73.99074
## 2             Munchables           00:25:09.107093   40.72308  -73.99462
## 3                 Drinks           00:06:44.541717   40.72848  -73.99839
## 4    Cold Pressed Juices                             40.73887  -74.00275
## 5   Maki (Special Rolls)           00:03:45.035418   40.72611  -74.00249
## 6 Vegetarian Specialties           00:07:14.327405   40.71927  -74.00875
##   dropoff_lat dropoff_lon  when_the_delivery_started
## 1    40.75207   -73.98537 2014-10-26 13:51:59.898924
## 2    40.71972   -73.99186  2014-10-16 21:58:58.65491
## 3    40.72861   -73.99514 2014-10-28 21:39:52.654394
## 4    40.75126   -74.00563 2014-10-30 10:54:11.531894
## 5    40.70932   -74.01587 2014-10-10 00:07:18.450505
## 6    40.72568   -74.00062 2014-10-22 18:56:36.348939
##   when_the_Jumpman_arrived_at_pickup when_the_Jumpman_left_pickup
## 1
## 2         2014-10-16 22:26:02.120931   2014-10-16 22:48:23.091253
## 3         2014-10-28 21:37:18.793405    2014-10-28 21:59:09.98481
## 4         2014-10-30 11:04:17.759577   2014-10-30 11:16:37.895816
## 5         2014-10-10 00:14:42.702223   2014-10-10 00:25:19.400294
## 6         2014-10-22 19:18:49.953427    2014-10-22 19:27:10.57897
##   when_the_Jumpman_arrived_at_dropoff
## 1          2014-10-26 14:52:06.313088
## 2          2014-10-16 22:59:22.948873
## 3          2014-10-28 22:04:40.634962
## 4          2014-10-30 11:32:38.090061
## 5          2014-10-10 00:48:27.150595
## 6          2014-10-22 19:36:53.801191
```

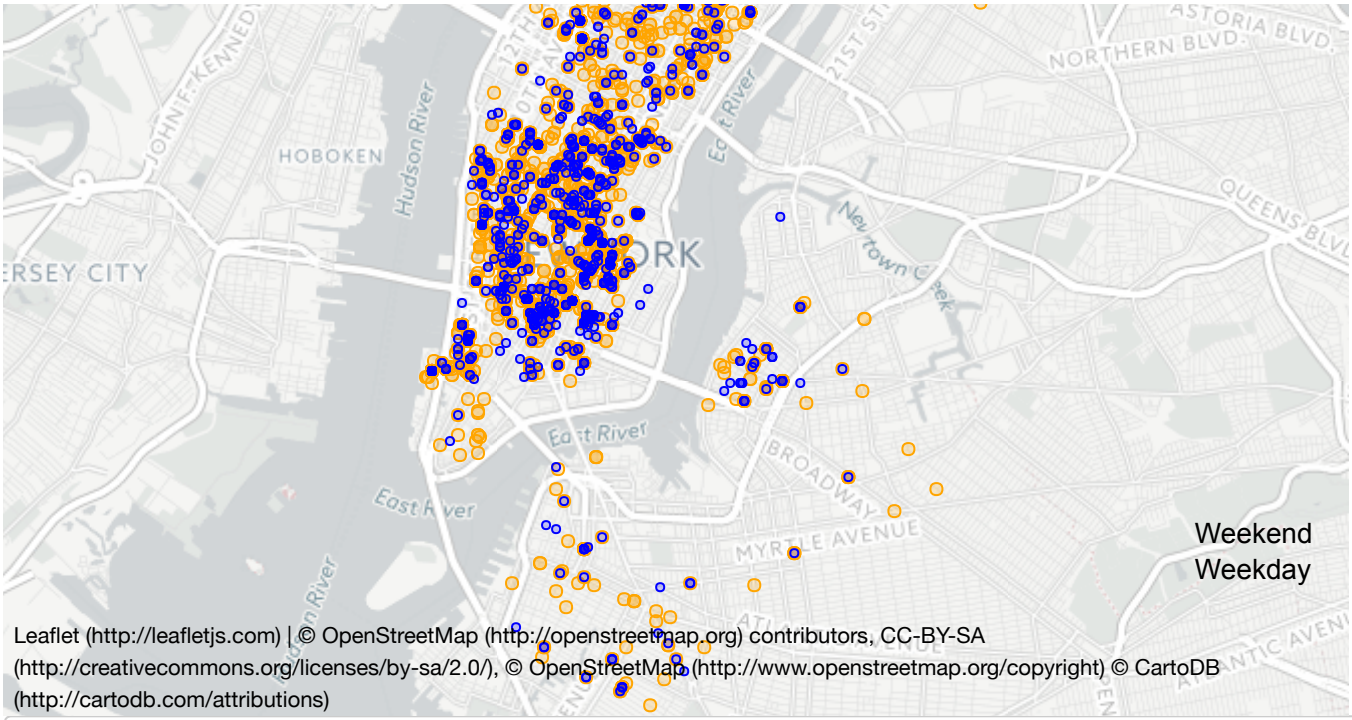
```
#lets take a look at the data frame
#lets check the integrity of the data missing values, etc...
#is.na(df) # returns TRUE of data is missing but also print a lot of extra pages

#looks like we have some item_quantity values missing
#the time stamps where imported as strings we need to convert them to a date format
df$when_the_delivery_started <- ymd_hms(substr(df$when_the_delivery_started,1,19))
df$when_the_Jumpman_arrived_at_pickup <- ymd_hms(substr(df$when_the_Jumpman_arrived_at_p
ickup,1,19))
df$when_the_Jumpman_left_pickup <- ymd_hms(substr(df$when_the_Jumpman_left_pickup,1,19))
df$when_the_Jumpman_arrived_at_dropoff <- ymd_hms(substr(df$when_the_Jumpman_arrived_at_
dropoff,1,19))

#it good to code days like monday, tuesday, etc.. into integers so we can use them numer
ical data. We will create new columns and
#add the below values.
df$wday_delivery_started <- wday(df$when_the_delivery_started)
df$weekend_delivery_started <- ifelse(df$wday_delivery_started %in% c(1,7),1,0)
df$day_delivery_started <- (day(df$when_the_delivery_started))

#creat time differenc columns for the jumpman timestamps
df$delivery_time <- difftime(df$when_the_Jumpman_arrived_at_dropoff,
                             df$when_the_Jumpman_left_pickup,
                             units="hours")
df$loading_time <- difftime(df$when_the_Jumpman_left_pickup,
                            df$when_the_Jumpman_arrived_at_pickup,
                            units="hours")
df$jumpman_arrival_time <- difftime(df$when_the_Jumpman_arrived_at_pickup,
                            df$when_the_delivery_started,
                            units="hours")

#delivery distance based off the lat and long to meters
df$delivery_distance <- 0
for(i in 1:nrow(df))
{
  df[i,'delivery_distance'] <- distm(c(df[i,"dropoff_lat"],df[i,"dropoff_lon"]),
                                     c(df[i,"pickup_lat"],df[i,"pickup_lon"]),
                                     fun=distHaversine)/1609.34
}

#compute the average jumpman speed and put in new column
df$jumpman_avg_speed <- df$delivery_distance/as.numeric(df$delivery_time)

#calculate average jumpman speed to delivery
df$jumpman_avg_speed <- df$delivery_distance/as.numeric(df$delivery_time)

#lets retreive the distinct values and discard the rest
df_unique <- df %>% distinct(delivery_id, .keep_all = TRUE)

#weekend vs weekday by dropoffs
leaflet() %>% setView(-73.972887,40.732828,zoom=12) %>% addTiles() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(data=subset(df_unique,weekend_delivery_started==0),
```

```
lat=~dropoff_lat,lng=~dropoff_lon,weight=1,radius=2,opacity=1,color="Green") %>%
  addCircleMarkers(data=subset(df_unique,weekend_delivery_started==1),

lat=~dropoff_lat,lng=~dropoff_lon,weight=1,radius=2,opacity=1,color="Red") %>%
  addLegend("bottomleft",colors =c("Red", "Green"),labels= c("Weekend","Weekday"),opacit
y = 1)
```




```
#weekend vs weekday by pickup
leaflet() %>% setView(-73.972887,40.732828,zoom=12) %>% addTiles() %>%
  addProviderTiles(providers$CartoDB.Positron) %>%
  addCircleMarkers(data=subset(df_unique,weekend_delivery_started==0),
                   lat=~pickup_lat,lng=~pickup_lon,weight=1,radius=3,opacity=1,color="Or
ange") %>%
  addCircleMarkers(data=subset(df_unique,weekend_delivery_started==1),
                   lat=~pickup_lat,lng=~pickup_lon,weight=1,radius=2,opacity=1,color="Bl
ue")%>%
  addLegend("bottomright",colors =c("Blue", "Orange"),labels= c("Weekend","Weekday"),opa
city = 1)
```

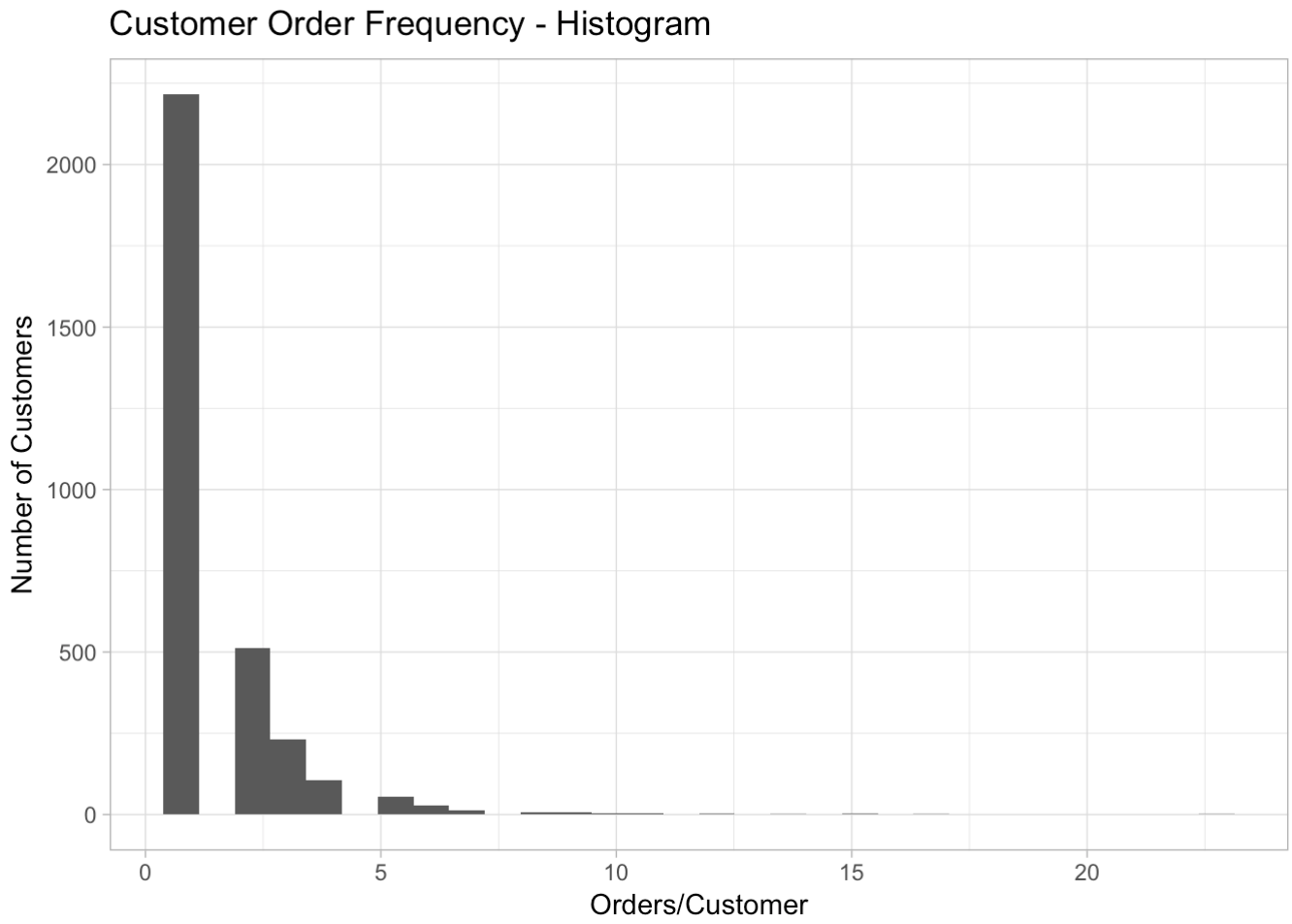




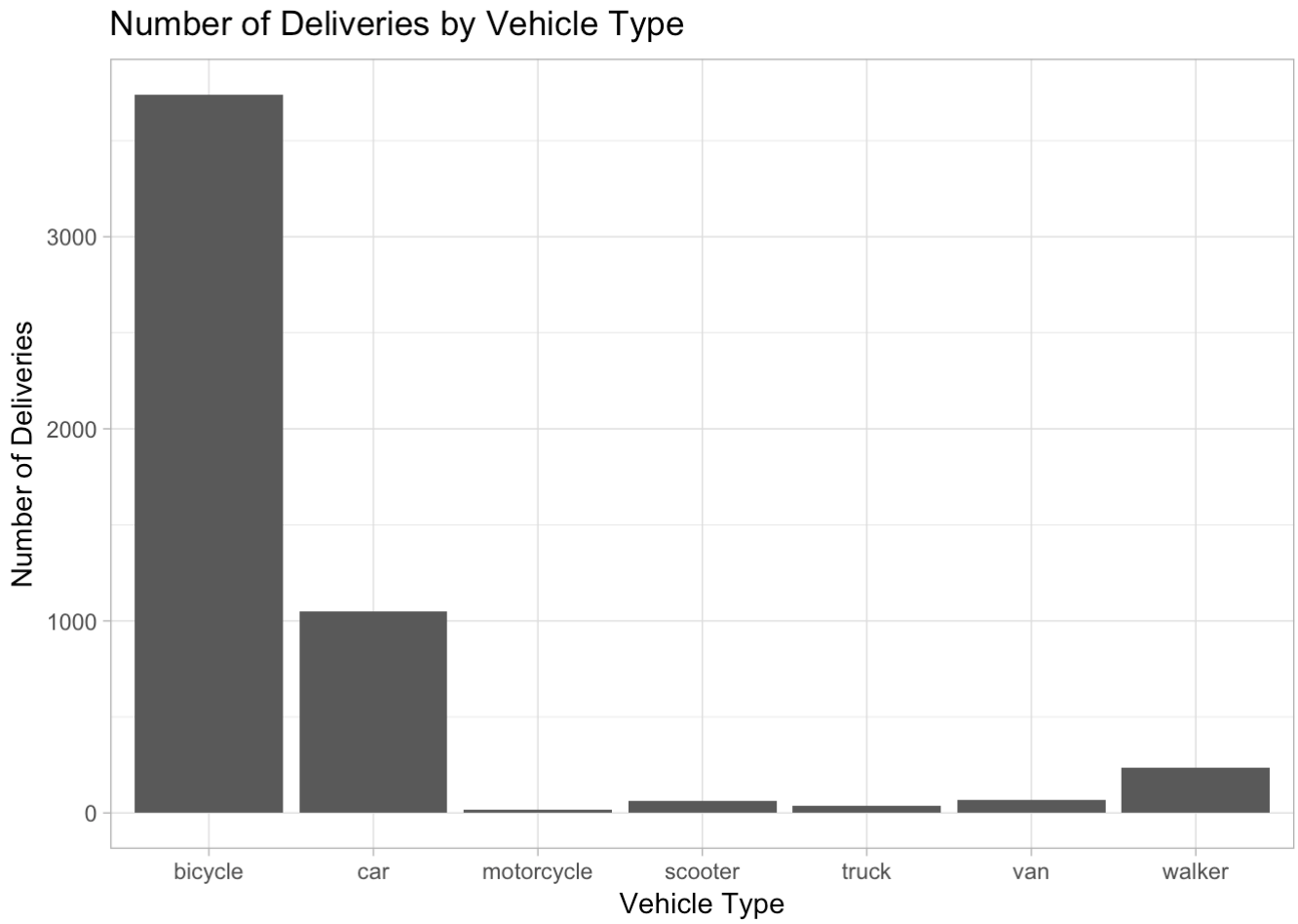

```
#unique number of customers
paste(length(unique(df_unique$customer_id))," Unique Customers")
```

```
## [1] "3192  Unique Customers"
```

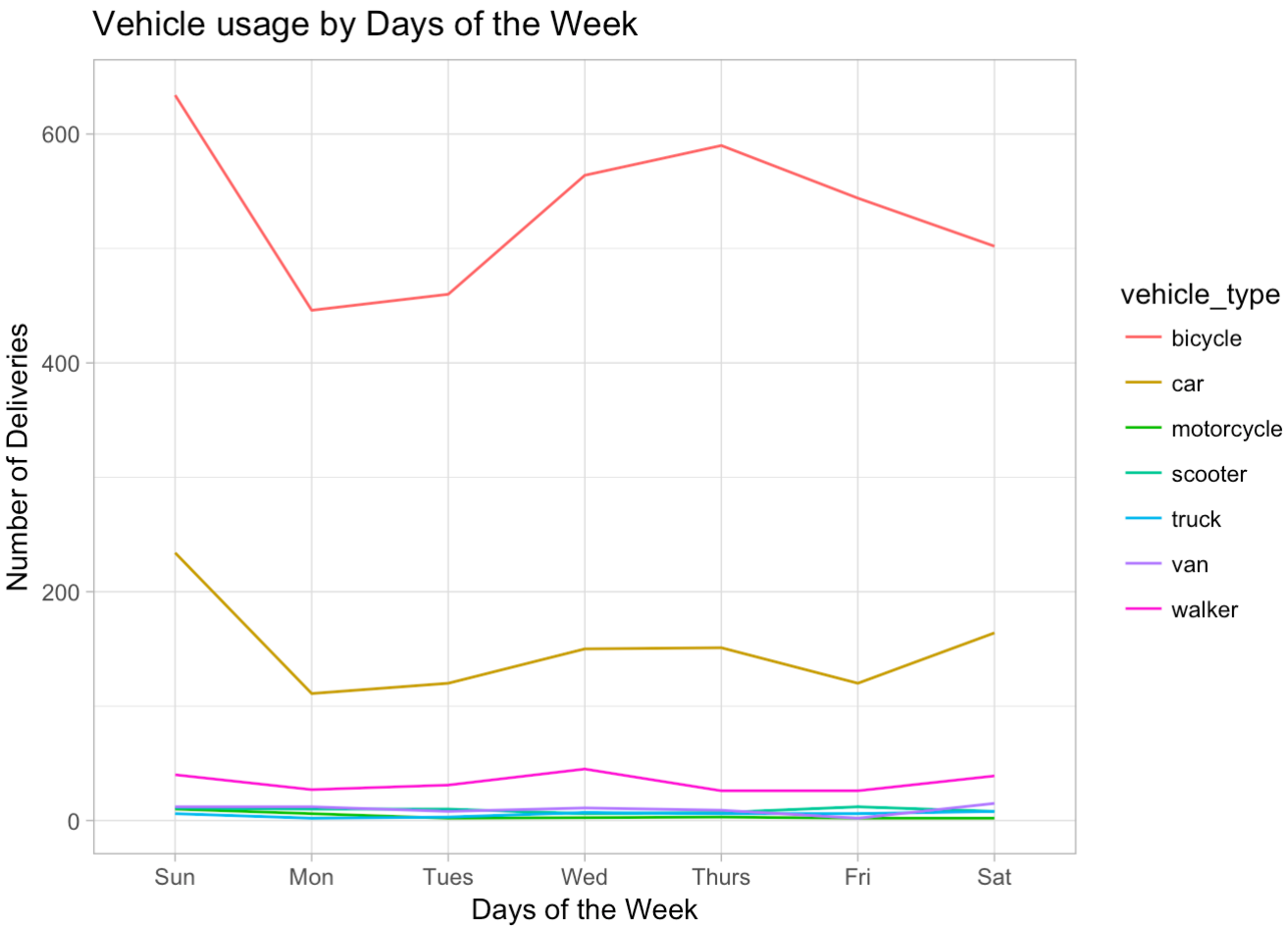
```
ggplot(data.frame(as.vector(table(df_unique$customer_id)))) +
  geom_histogram(bins=30,aes(x=as.vector.table.df_unique.customer_id..))+
  ggtitle("Customer Order Frequency - Histogram")+
  xlab("Orders/Customer")+
  ylab("Number of Customers") + theme_light()
```




```
#vehicle usage
ggplot(df_unique,aes(x=vehicle_type, 1,group=1)) +
  stat_summary(fun.y = sum,geom = "bar")+
  ggtitle("Number of Deliveries by Vehicle Type")+
  xlab("Vehicle Type")+ylab("Number of Deliveries") + theme_light()
```



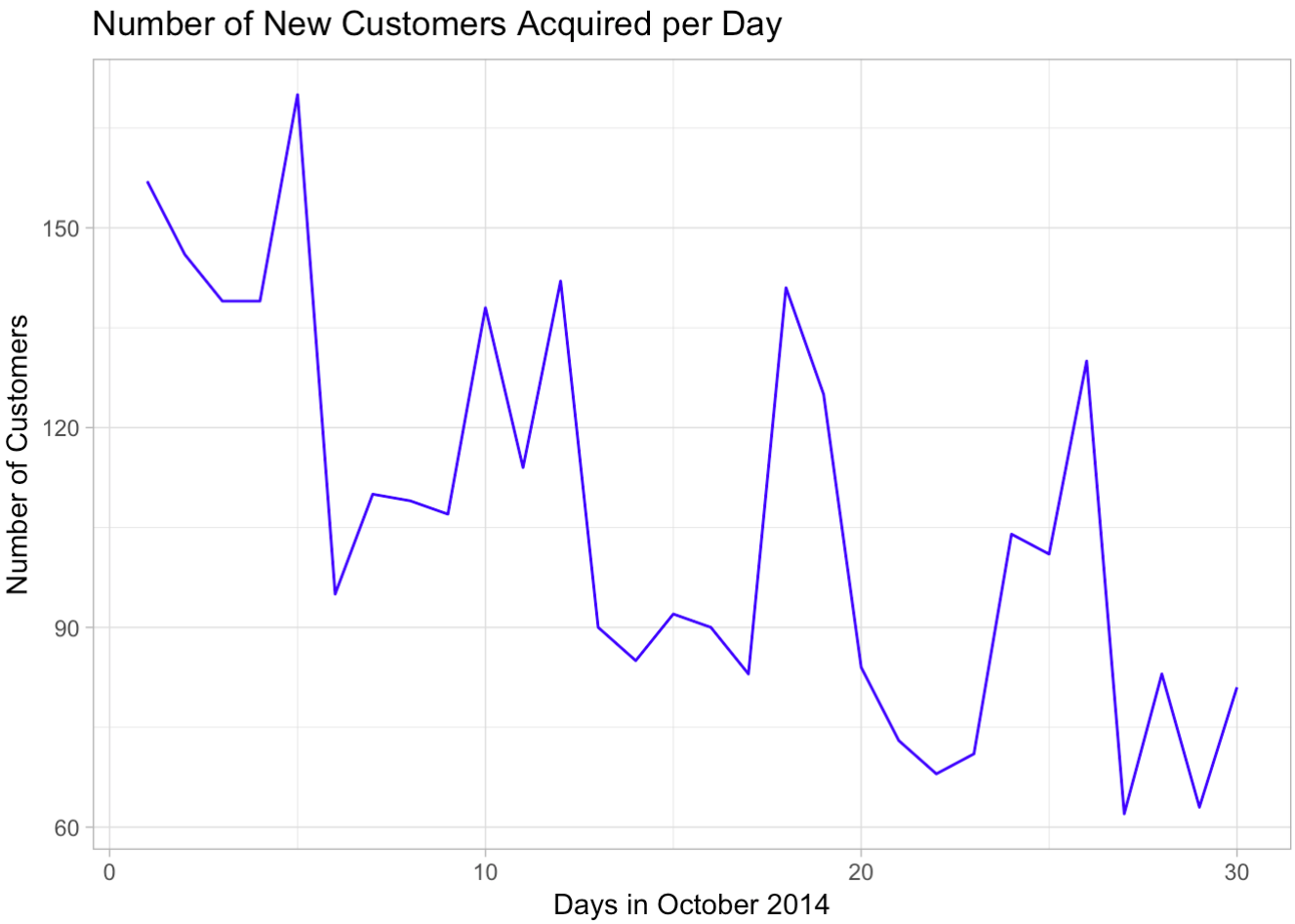

```
#vechile usage by days of the week
ggplot(df_unique,aes(x=wday(when_the_delivery_started,label=T), 1,group=vehicle_type,col
or=vehicle_type)) +
  stat_summary(fun.y = sum,geom = "line",size=.5)+
  ggtitle("Vehicle usage by Days of the Week")+
  xlab("Days of the Week")+ylab("Number of Deliveries") + theme_light()
```



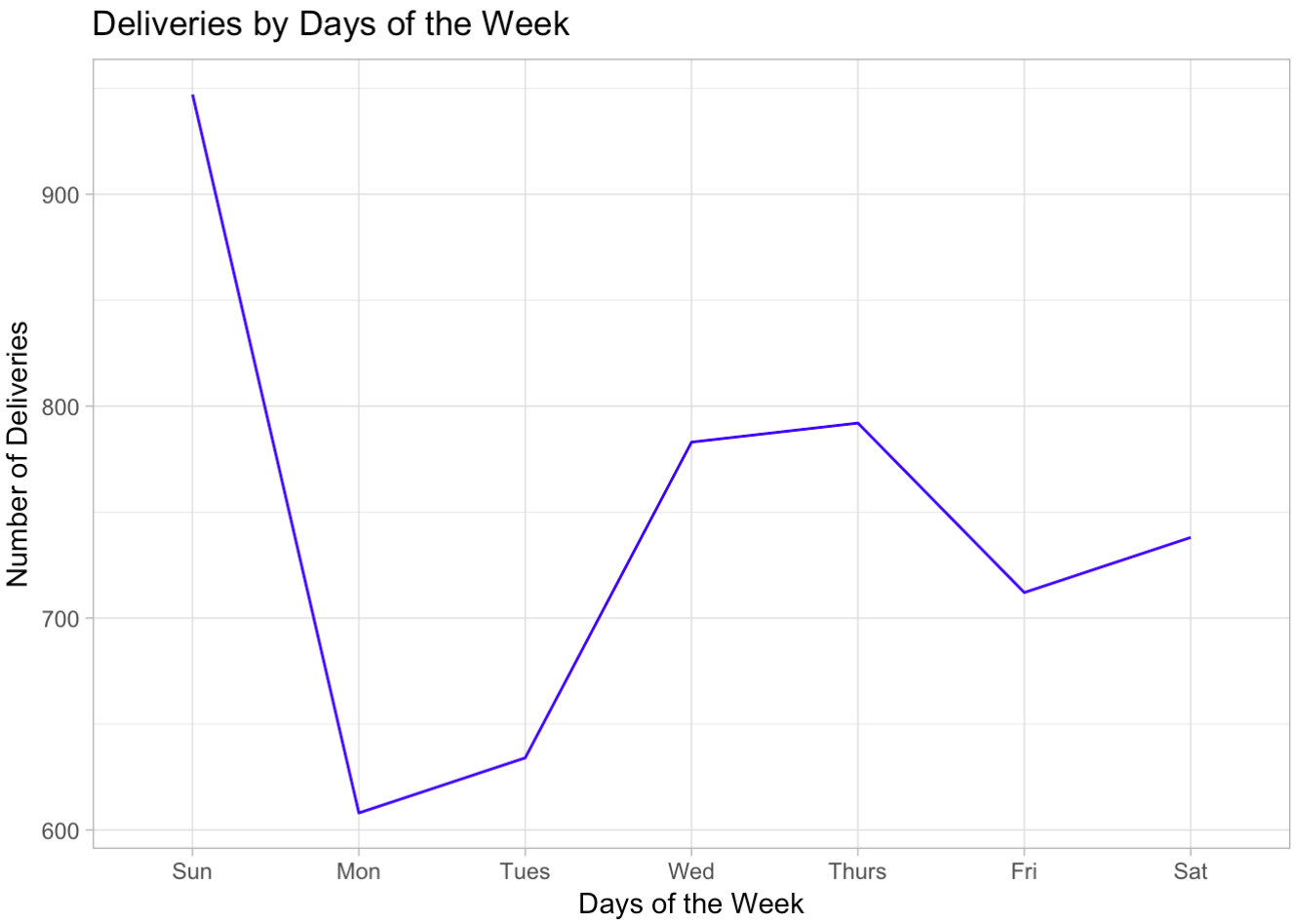

```
#range of dates deliver start
paste("Dates range from ",
      min(df_unique$when_the_delivery_started),
      " to ",
      max(df_unique$when_the_delivery_started)
      )
```

```
## [1] "Dates range from  2014-10-01 00:07:58  to  2014-10-30 23:08:43"
```

```
#customers acquired
cust_acq <- df_unique %>%
  group_by(customer_id) %>%
  summarise(first_day=min(day(when_the_delivery_started)))

ggplot(cust_acq,aes(x=first_day,y=1)) +
  stat_summary(fun.y=sum,geom="line", colour= "blue") +
  ggtitle("Number of New Customers Acquired per Day")+
  ylab("Number of Customers")+
  xlab("Days in October 2014") + theme_light()
```



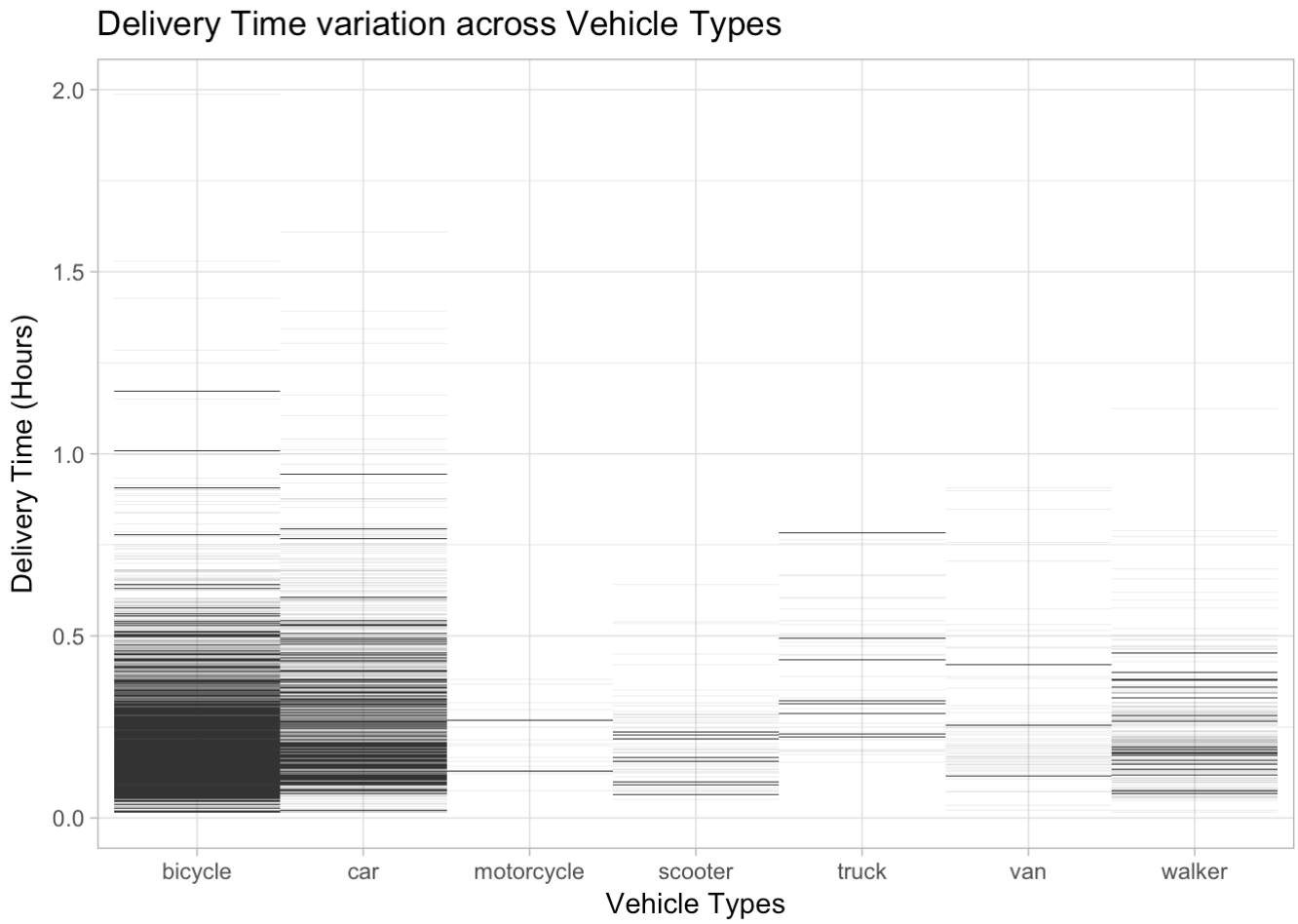

```
#delivery trends
ggplot(df_unique,aes(x=wday(when_the_delivery_started,label=T), 1,group=1)) +
  stat_summary(fun.y = sum,geom = "line", colour= "blue")+
  ggtitle("Deliveries by Days of the Week")+
  ylab("Number of Deliveries")+xlab("Days of the Week") + theme_light()
```




```
#delivery time by day of the week
ggplot(df_unique,aes(x=vehicle_type,y=delivery_time))+
  geom_tile()+
  ggtitle("Delivery Time variation across Vehicle Types")+
  xlab("Vehicle Types")+ylab("Delivery Time (Hours)") + theme_light()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to
continuous.
```

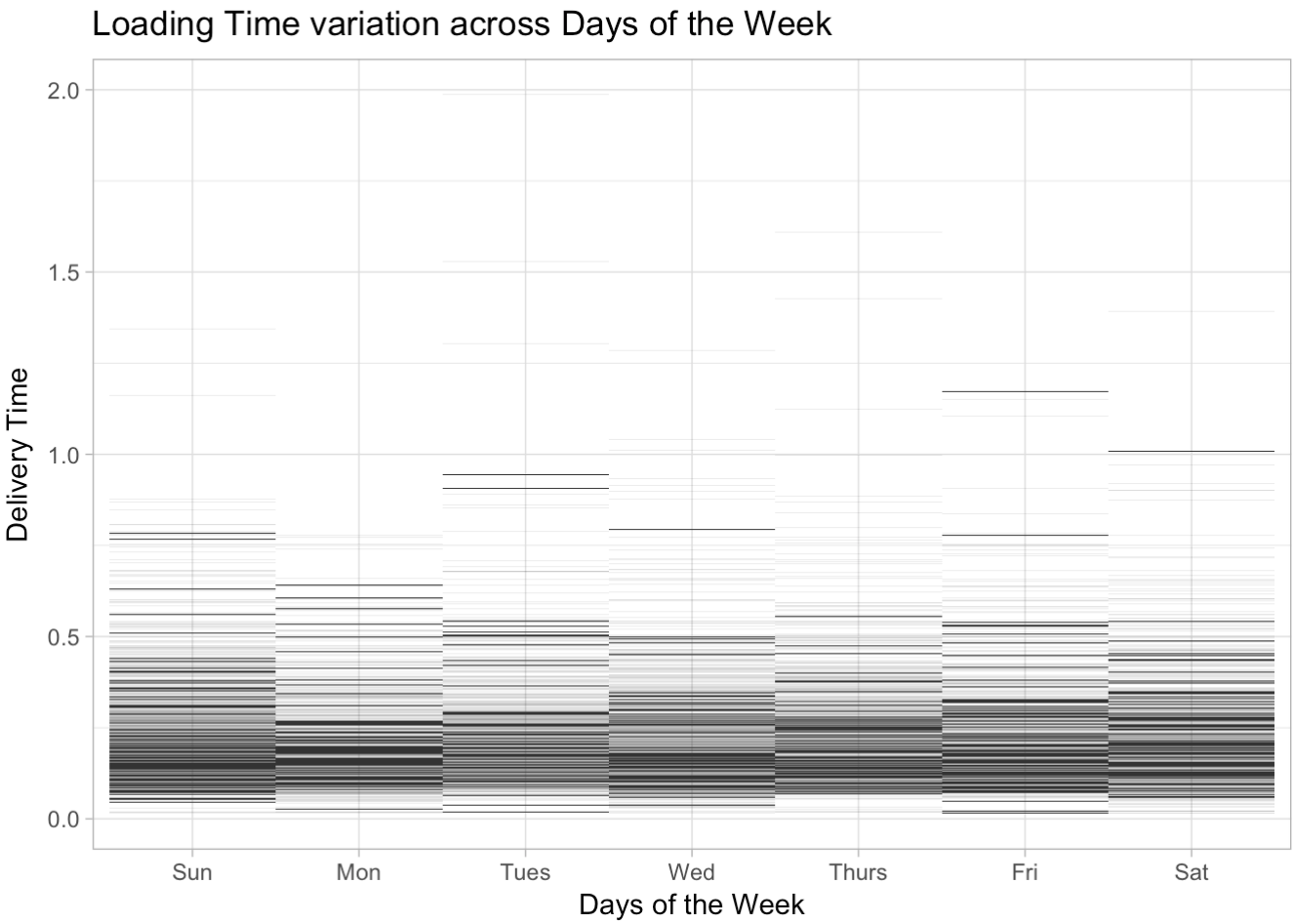
```
## Warning: Removed 495 rows containing missing values (geom_tile).
```




```
#jumpman arrival by day of the week
ggplot(df_unique,aes(x=wday(when_the_delivery_started,label=T),y=delivery_time))+
  geom_tile()+
  ggtitle("Loading Time variation across Days of the Week")+
  xlab("Days of the Week")+ylab("Delivery Time") + theme_light()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to
continuous.
```

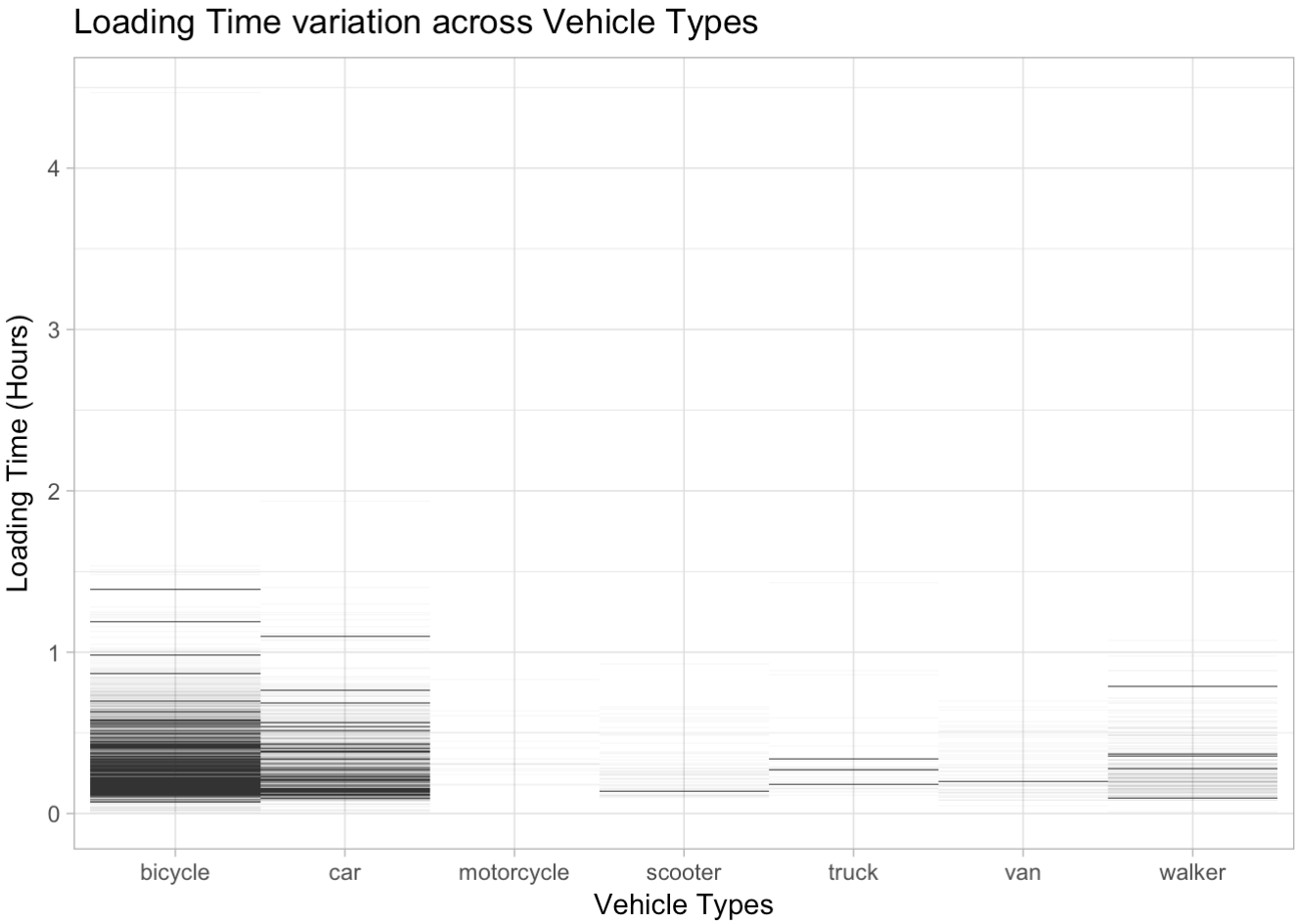
```
## Warning: Removed 495 rows containing missing values (geom_tile).
```



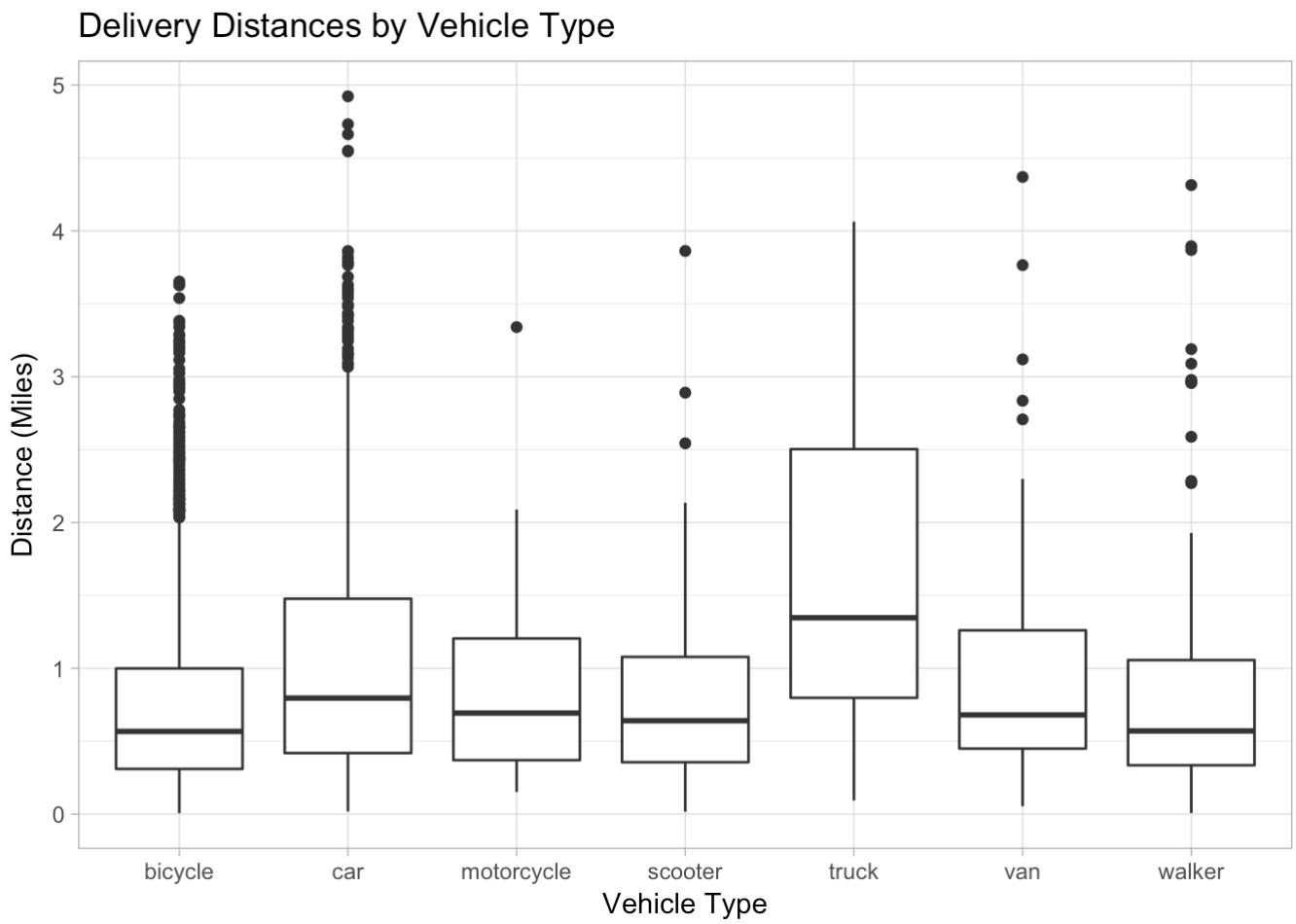

```
ggplot(df_unique,aes(x=vehicle_type,y=loading_time))+
  geom_tile()+
  ggtitle("Loading Time variation across Vehicle Types")+
  xlab("Vehicle Types")+ylab("Loading Time (Hours)") + theme_light()
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to
continuous.
```

```
## Warning: Removed 495 rows containing missing values (geom_tile).
```

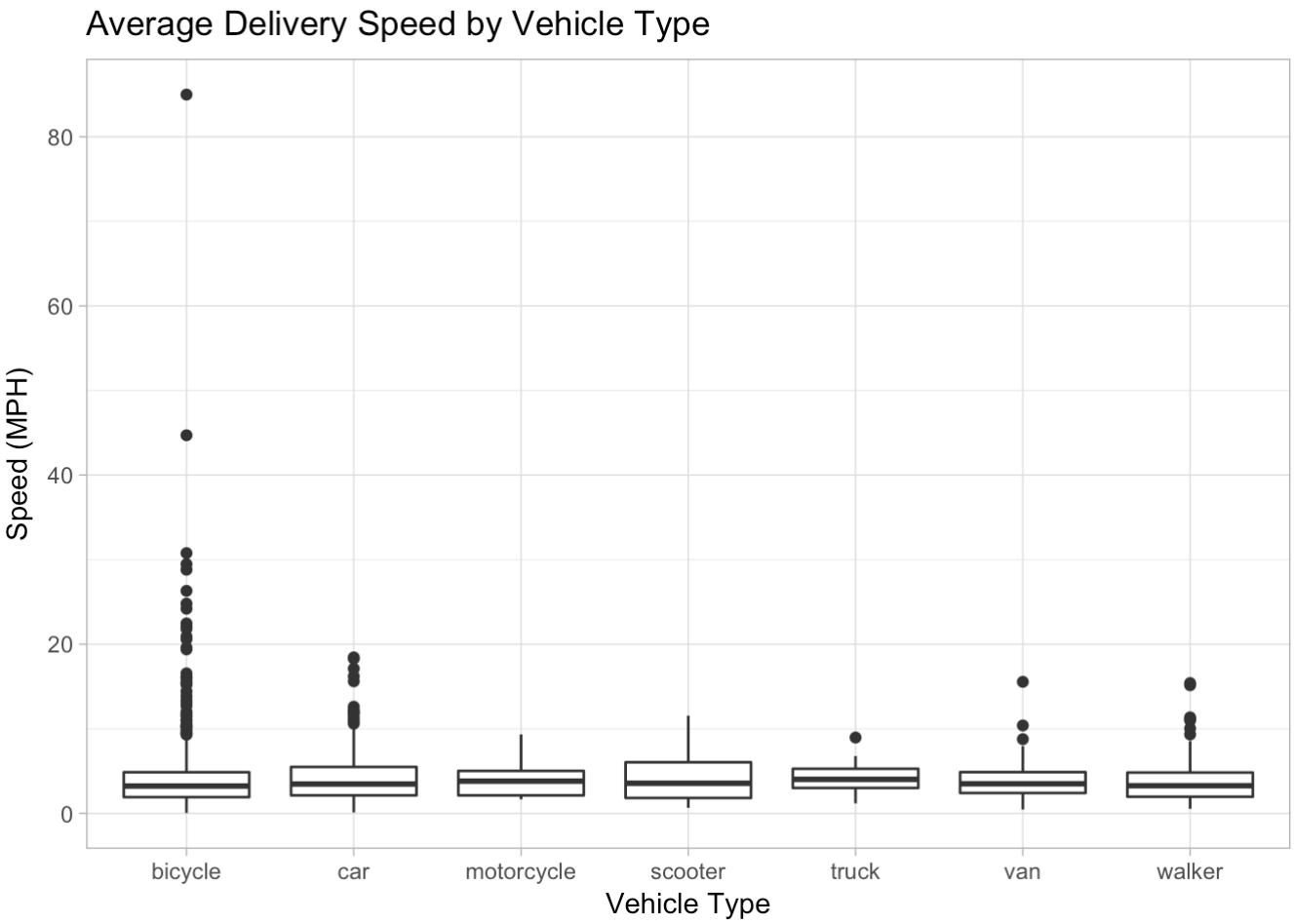



```
ggplot(df_unique,aes(x=vehicle_type,y=delivery_distance))+
  geom_boxplot()+
  ggtitle("Delivery Distances by Vehicle Type")+
  xlab("Vehicle Type")+
  ylab("Distance (Miles)") + theme_light()
```



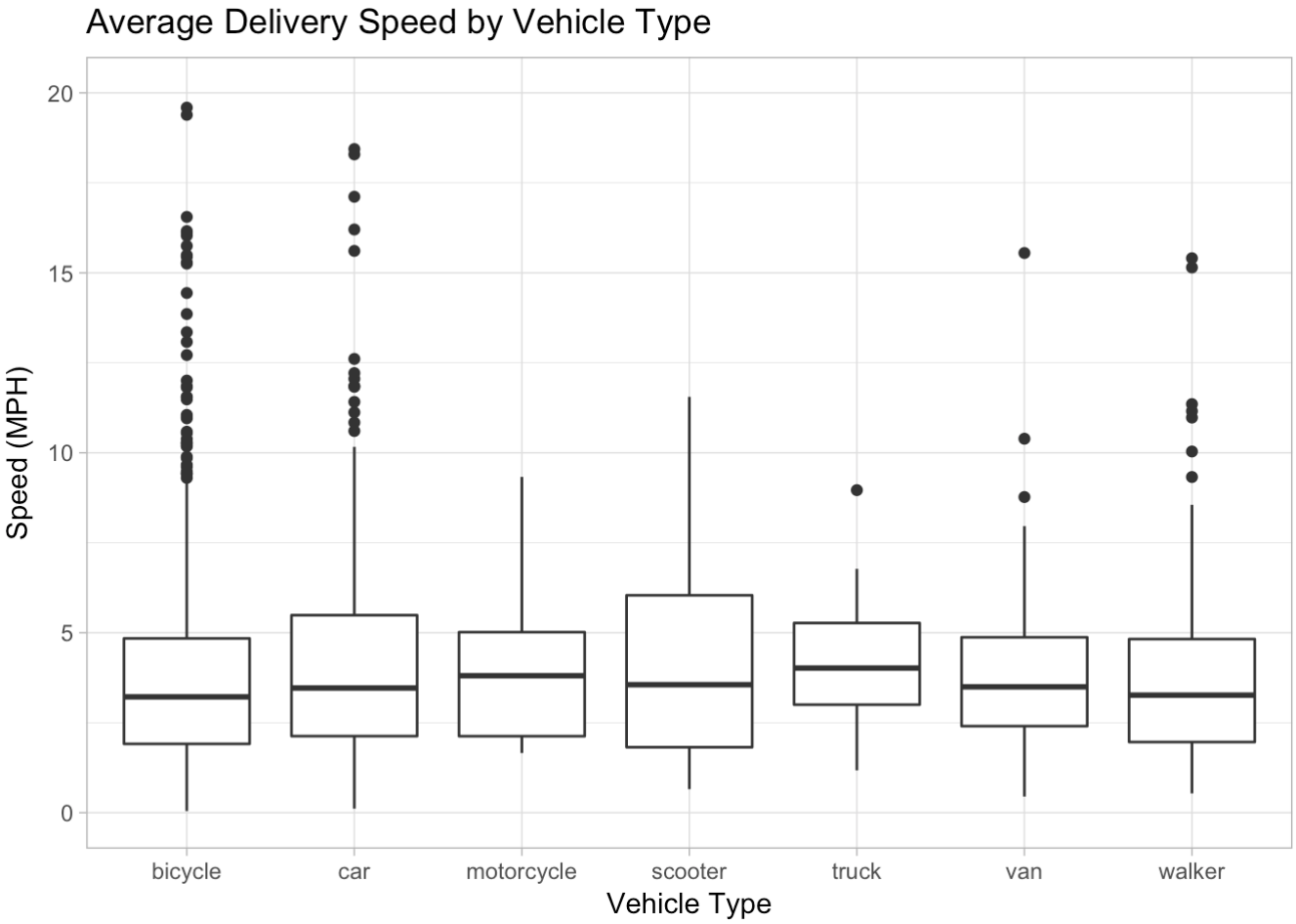

```
ggplot(df_unique,aes(x=vehicle_type,y=jumpman_avg_speed))+
  geom_boxplot()+
  ggtitle("Average Delivery Speed by Vehicle Type")+
  xlab("Vehicle Type")+
  ylab("Speed (MPH)") + theme_light()
```

```
## Warning: Removed 495 rows containing non-finite values (stat_boxplot).
```




```
ggplot(df_unique,aes(x=vehicle_type,y=jumpman_avg_speed))+
  geom_boxplot()+
  ggtitle("Average Delivery Speed by Vehicle Type")+
  xlab("Vehicle Type")+
  ylab("Speed (MPH)")+ylim(0,20) + theme_light()
```

```
## Warning: Removed 508 rows containing non-finite values (stat_boxplot).
```


Average Delivery Speed by Vehicle Type
Speed (MPH)
20
15
10
5
0
bicycle
car
motorcycle
scooter
truck
van
walker
Vehicle Type