# Team Bike

```
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(ggplot2)
library(readr)
library(lubridate)
```

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

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

```
library(chron)
```

```
##
## Attaching package: 'chron'
```

```
## The following objects are masked from 'package:lubridate':
##
##     days, hours, minutes, seconds, years
```

```
library(Rtsne)
NYC <- read_csv("~/Desktop/NYC-3.txt")
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   valid = col_datetime(format = ""),
##   tmpf = col_double(),
##   dwpf = col_double(),
##   relh = col_double(),
##   p01i = col_double(),
##   alti = col_double(),
##   vsby = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
# Removing the extraneous variables
NYC <- NYC %>%
  select(-drct,-p01i,-skyc1,-skyc2,-skyc3,-skyc4,skyl1,-skyl2,-skyl2,-skyl3,-skyl4,-metar)
randomsample <- read_csv("~/Desktop/compstats/ma154-project24-teambike/final_project/randomsample.csv")
```

```
## Parsed with column specification:
```

```
## cols(
##   X1 = col_integer(),
##   tripduration = col_integer(),
##   starttime = col_character(),
##   stoptime = col_character(),
##   start.station.id = col_integer(),
##   start.station.name = col_character(),
##   start.station.latitude = col_double(),
##   start.station.longitude = col_double(),
##   end.station.id = col_integer(),
##   end.station.name = col_character(),
##   end.station.latitude = col_double(),
##   end.station.longitude = col_double(),
##   bikeid = col_integer(),
##   usertype = col_character(),
##   birth.year = col_integer(),
##   gender = col_integer()
## )
```

```
# Parsing the CitiBikes Data
NYC$valid <- parse_date_time(NYC$valid,orders = "ymd HMS")



# Trying to get hourly data - one observation per hour, instead of multiple observations
NYC <- NYC %>%
  filter(minute(valid)=="51")
NYC <- NYC %>%
  mutate(Month=month(valid))

# Cieling the Hour
NYC$valid <- ceiling_date(NYC$valid, unit = "hour")

# Making Binary Variables for the Seasons
NYC <- NYC %>%
  mutate(summer=ifelse(Month=="6"|Month=="7"|Month=="8",1,0)) %>%
  mutate(spring=ifelse((Month=="3"|Month=="4"|Month=="5"),1,0)) %>%
  mutate(winter=ifelse((Month=="1"|Month=="2"|Month=="12"),1,0)) %>%
  mutate(fall=ifelse((Month=="9"|Month=="10"|Month=="11"),1,0))


# Trying to Find Day of the Week
NYC <- NYC %>%
  mutate(WeekDay=wday(as.Date(valid), label=TRUE)) %>%
  mutate(WorkingDay=ifelse((WeekDay=="Mon"|WeekDay=="Tues"|WeekDay=="Wed"|WeekDay=="Thurs"|WeekDay=="Fri

# Trying to find the holidays - Not quite working. Need to do this manually!
NYC <- NYC %>%
  mutate(holiday=is.weekend(valid))



# So I'm joing the two df's based on the date and the hour. The uid identifier is the date and hour.
```

```r
head(NYC$valid)
```

```
## [1] "2013-07-01 02:00:00 UTC" "2013-07-01 03:00:00 UTC"
## [3] "2013-07-01 04:00:00 UTC" "2013-07-01 05:00:00 UTC"
## [5] "2013-07-01 06:00:00 UTC" "2013-07-01 07:00:00 UTC"
```

```r
# take a look, I was able to join the two data frames based on the date and hour. I used hourly data wh

colnames(NYC)[2] <- "starttime"
```

```r
# Parsing all the start times into one format
mdy <- mdy_hms(randomsample$starttime)
```

```
## Warning: 285131 failed to parse.
```

```r
ymd <- ymd_hms(randomsample$starttime)
```

```
## Warning: 232836 failed to parse.
```

```r
f1 <- mdy_hm(randomsample$starttime)
```

```
## Warning: 482033 failed to parse.
```

```r
mdy[is.na(mdy)] <- ymd[is.na(mdy)] # some dates are ambiguous, here we give
randomsample$starttime <- mdy
randomsample$starttime[is.na(randomsample$starttime)] <- f1[is.na(randomsample$starttime)]

# Parsing all the end times into one format
mdy <- mdy_hms(randomsample$stoptime)
```

```
## Warning: 285131 failed to parse.
```

```r
ymd <- ymd_hms(randomsample$stoptime)
```

```
## Warning: 232836 failed to parse.
```

```r
f1 <- mdy_hm(randomsample$stoptime)
```

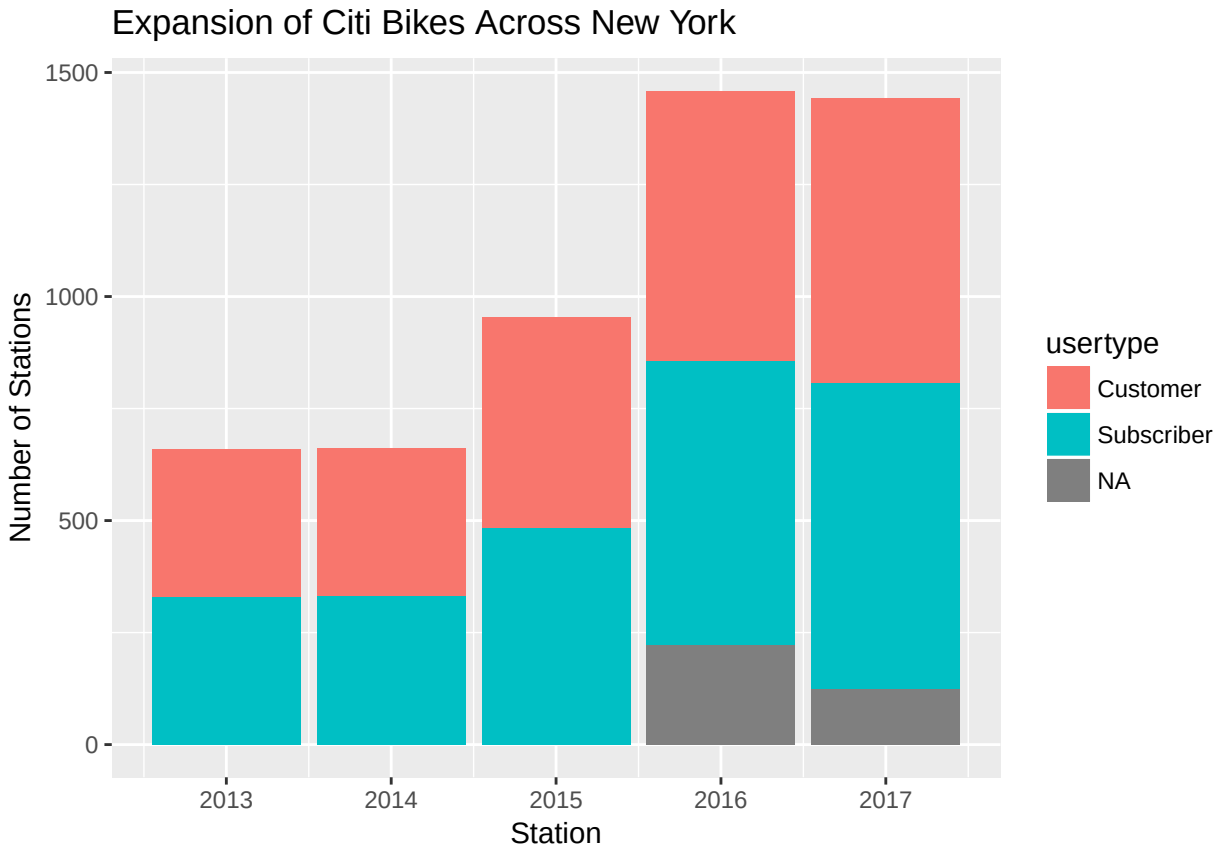
```
## Warning: 482033 failed to parse.
```

```r
mdy[is.na(mdy)] <- ymd[is.na(mdy)]
randomsample$stoptime <- mdy
randomsample$stoptime[is.na(randomsample$stoptime)] <- f1[is.na(randomsample$stoptime)]

# Flooring the Hour
randomsample$starttime <- floor_date(randomsample$starttime,unit = "hour")

# Joining the weather data and the random sample
combined <- inner_join(NYC,randomsample)
```

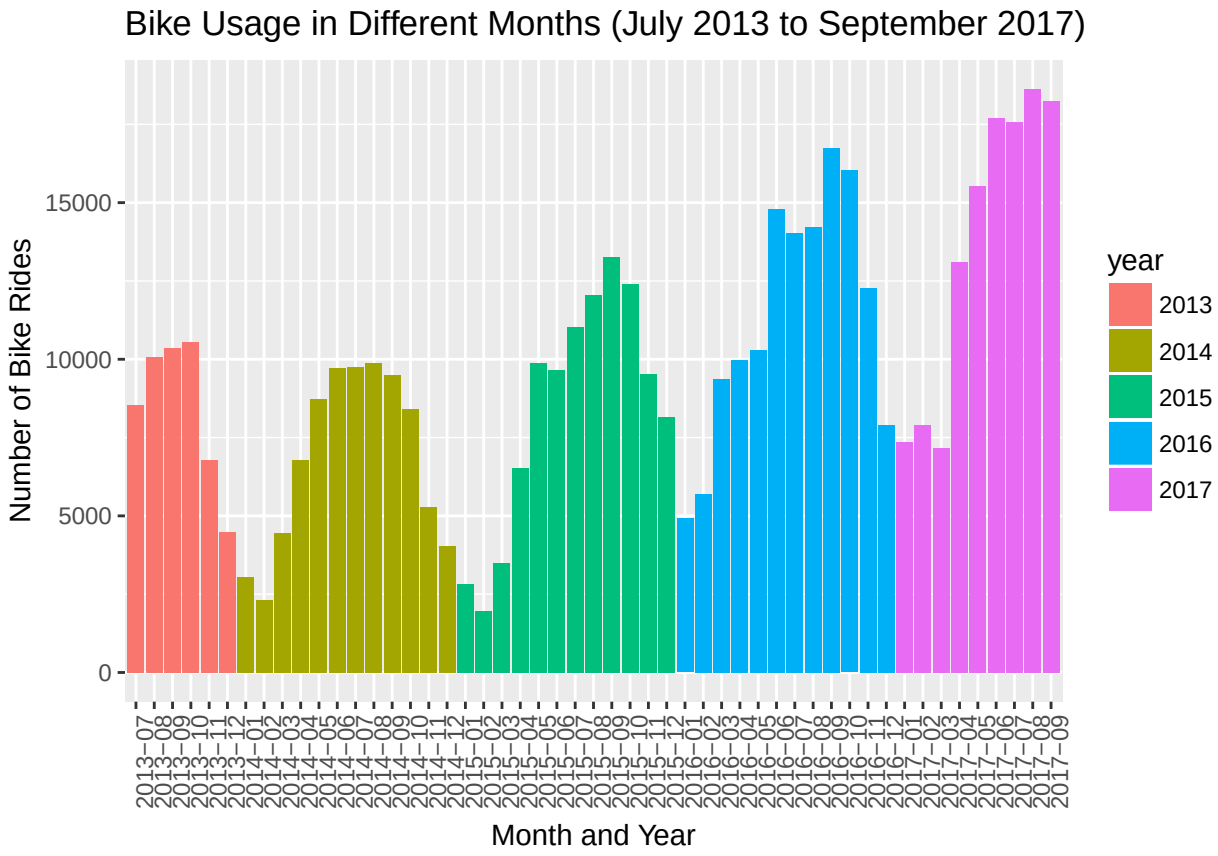
```
## Joining, by = "starttime"
```

```r
# Number of Stations Every Year
rs <- randomsample %>%
  mutate(year=year(starttime)) %>%
  group_by(year,usertype) %>%
  summarise(stations=n_distinct(start.station.id),rides=n())
```

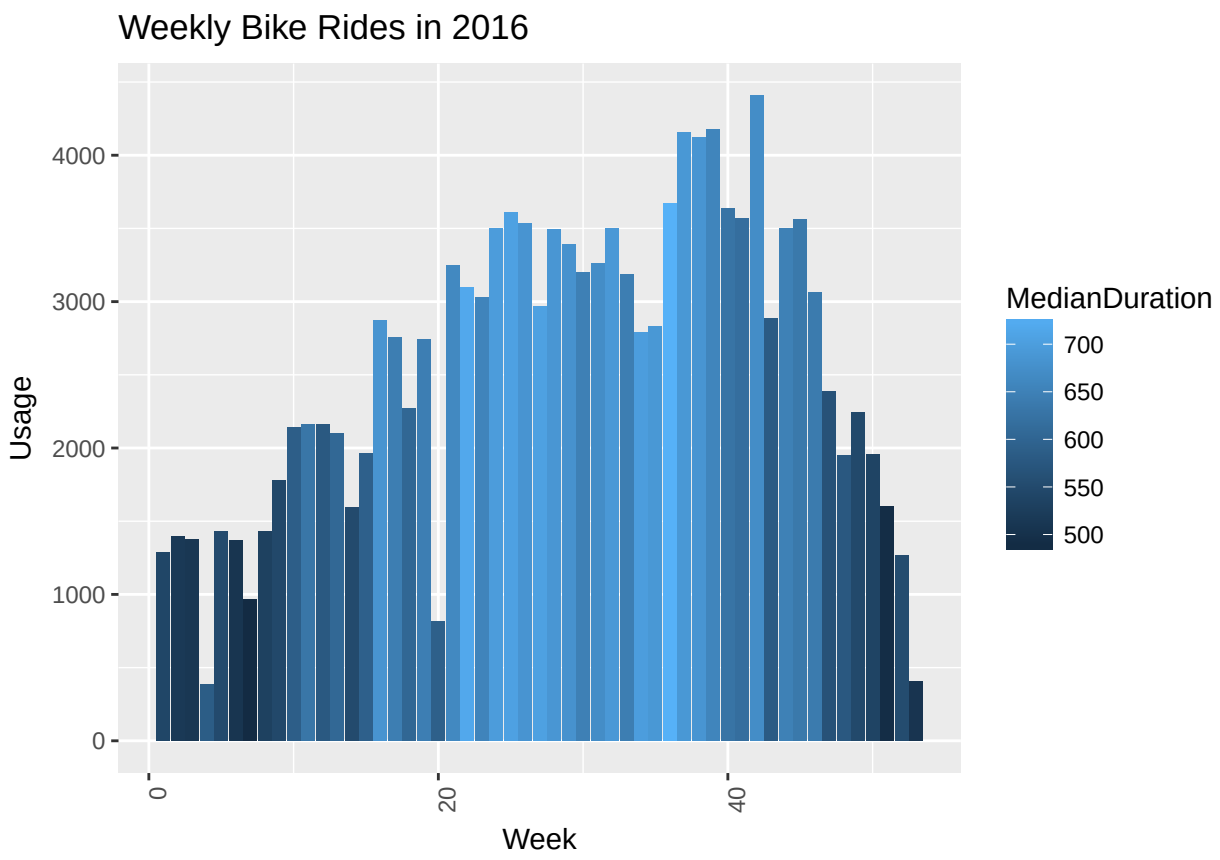
```r
# Plotting the expansion of Citi Bikes Across New York
ggplot(rs,aes(x=year,y=stations,fill=usertype))+geom_bar(stat="identity")+ylab("Number of Stations")+xla
```



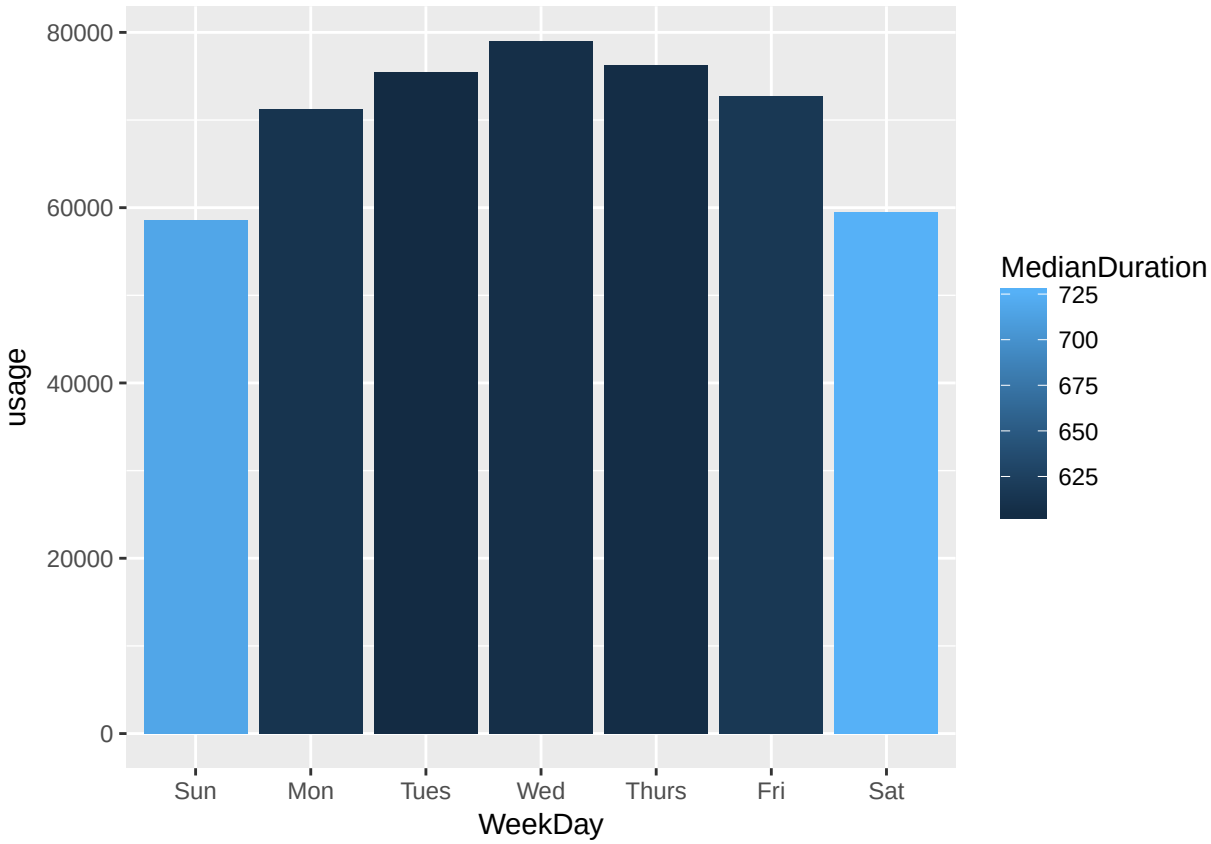

```r
# Exploratory Analysis -
combined1 <- combined%>%
  mutate(year=year(starttime),month=month(starttime),monyear=substr(starttime,1,7)) %>%
  group_by(monyear,year)%>%
  summarise(usage=n())

combined1$year <- as.factor(combined1$year)

# Graph to See the Monthly Usage
ggplot(combined1,aes(x=monyear,y=usage,fill=year))+geom_bar(stat="identity")+ theme(axis.text.x = eleme
```



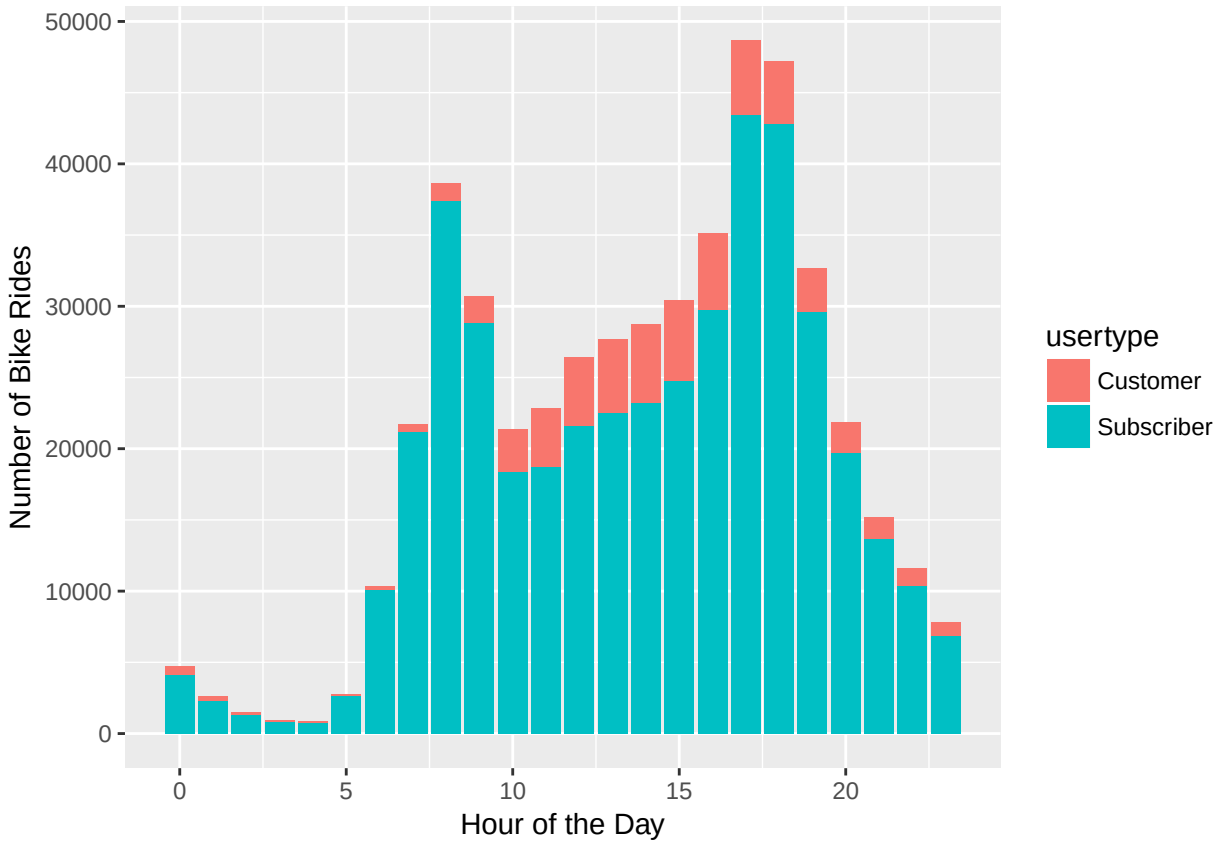

```
# Wrangling the data
combined2 <-  combined%>%
  mutate(year=year(starttime),week=week(starttime),monyear=substr(starttime,1,7)) %>%
  group_by(week,year)%>%
  summarise(usage=n(),MedianDuration=median(tripduration)) %>%
  mutate(wy=paste(week,"-",year)) %>%
  filter(year==2016) %>%
  arrange(year)

# Graph to See the Usage in 2016
ggplot(combined2,aes(x=week,y=usage,fill=MedianDuration))+geom_bar(stat="identity")+theme(axis.text.x =
```




```
# Wrangling
combined3 <- combined%>%
  mutate(year=year(starttime),month=month(starttime),monyear=substr(starttime,1,7)) %>%
  group_by(WeekDay)%>%
  summarise(usage=n(),MedianDuration=median(tripduration))
# Graphing to See the Week Day
ggplot(combined3,aes(x=WeekDay,y=usage,fill=MedianDuration))+geom_bar(stat = "identity")
```




```
combined4 <- combined%>%
  mutate(year=year(starttime),month=month(starttime),hour=hour(starttime)) %>%
  filter(!is.na(usertype))%>%
  group_by(hour,gender,usertype)%>%
  summarise(usage=n(),MedianDuration=median(tripduration))
# The Number of Rides for every hour - Will make binary variables for TSNE Clustering then.

ggplot(combined4,aes(x=hour,y=usage,fill=usertype))+geom_bar(stat="identity")+xlab("Hour of the Day")+y
```




```
# Making Different Levels for the Time of the Day
combined.clustering <- combined4 %>%
  mutate(EarlyMorning=ifelse(hour=="3"|hour=="4"|hour=="5"|hour=="6",1,0)) %>%
  mutate(Commuting=ifelse((hour=="7"|hour=="8"|hour=="9"),1,0)) %>%
  mutate(DayTime=ifelse((hour=="10"|hour=="11"|hour=="12"|hour=="13"|hour=="14"|hour=="15"),1,0)) %>%
  mutate(Evening=ifelse((hour=="16"|hour=="17"|hour=="18"|hour=="19"|hour=="20"),1,0)) %>%
  mutate(Night=ifelse((hour=="21"|hour=="22"|hour=="23"|hour=="0"|hour=="1"|hour=="2"),1,0))
combined4
```

```
## # A tibble: 141 x 5
## # Groups:   hour, gender [72]
##     hour gender   usertype usage MedianDuration
##    <int>  <int>      <chr> <int>          <dbl>
##  1     0      0   Customer   579         1077.0
##  2     0      0 Subscriber    31          640.0
##  3     0      1   Customer    15         1095.0
##  4     0      1 Subscriber  3351          527.0
##  5     0      2   Customer     2         1132.5
##  6     0      2 Subscriber   764          596.0
##  7     1      0   Customer   350         1048.0
##  8     1      0 Subscriber    16          636.0
##  9     1      1   Customer     9         1006.0
## 10     1      1 Subscriber  1877          521.0
## # ... with 131 more rows
```

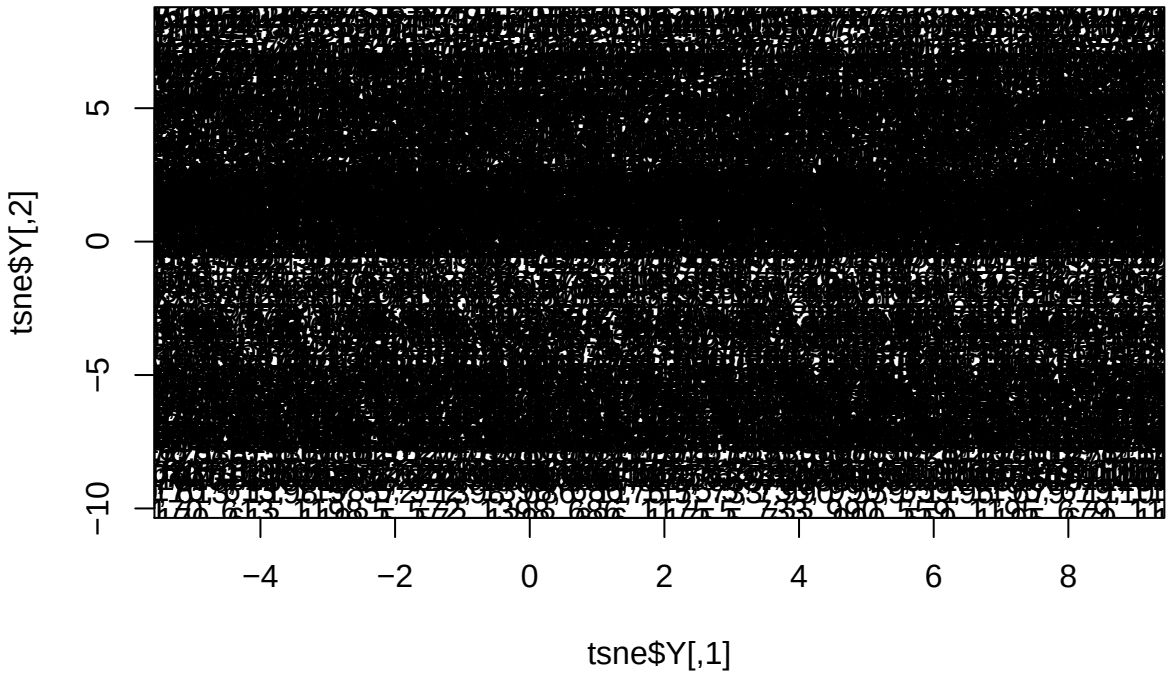
```
dim(combined4)
```

```
## [1] 141   5
```

```
combined4 <- combined4 %>%
  filter(!is.na(usertype)) %>%
  mutate(usertype=ifelse(usertype=="Customer",1,0))
combined4
```

```
## # A tibble: 141 x 5
## # Groups:   hour, gender [72]
##     hour gender usertype usage MedianDuration
##    <int>  <int>    <dbl> <int>          <dbl>
##  1     0      0        1   579         1077.0
##  2     0      0        0    31          640.0
##  3     0      1        1    15         1095.0
##  4     0      1        0  3351          527.0
##  5     0      2        1     2         1132.5
##  6     0      2        0   764          596.0
##  7     1      0        1   350         1048.0
##  8     1      0        0    16          636.0
##  9     1      1        1     9         1006.0
## 10     1      1        0  1877          521.0
## # ... with 131 more rows
```

```
# run Rtsne with default parameters - tsne
tsne <- Rtsne(as.matrix(combined4), check_duplicates = FALSE, pca = FALSE, perplexity=30, theta=0.5, di
cols <- rainbow(10)
plot(tsne$Y, t='n')
text(tsne$Y, labels=combined4[,5])
```




Rationale for Choosing the weather data variables:

The variables relating to the weather data chosen are air temperature in Fahrenheit(tmpf), Dew Point Temperature in Fahrenheit (dwpf), Relative Humidity in percentage (relh), Wind Speed in knots (sknt), Pressure altimeter in inches (alti), and visibility in miles (vsby). We chose these variables because each of these variables can have an impact on a biker's decision to choose or not to choose to bike and can significantly affect bike usage. We chose these variables to build a preliminary model and make predictions based off the

model. The other variables like wind direction in degrees from north, sky level coverages, sky level altitudes, and wind gust had some missing observations. We believe that these variables don't affect bike usage much and not including them would not affect the model.