Bike Sharing Service Analysis

CS 7280, Northeastern University

Author: Josh Gartman

March 29, 2016

Introduction:

Cities around the world are constantly looking for innovative ways to provide their residents with alternative forms of transportation that are fast, clean, and promote healthy lifestyles. Bike sharing services are an example of a way that residents can be encouraged to exercise while reducing traffic congestion and motor vehicle emissions. In Boston, Hubway operates approximately 150 bike sharing stations serving tens of thousands of customers annually. For services such as Hubway to be successful they must analyze the riding habits of their customers so they can better target their services, project costs and expand their reach. My project focuses on a data set containing information about trips taken using Hubway from the inception of the service in 2011 to the end of 2012. This data set is available at http://hubwaydatachallenge.org/. Specifically, my analysis intends to create a model of the duration of a rider's trip based on demographic information about that rider and information about the distance traveled on the trip itself. This information could be used to make usage projections based on the demographic information of current and potential markets.

Methods:

The most important method used in this analysis is multiple linear regression. Multiple linear regression models a response variable Y as a function of a set of predictor variables X_i. The formal description of a general linear regression model is $Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_{p-1} X_{i,p-1} + \epsilon_i$ where $\beta_0, \beta_1, ..., \beta_{p-1}$ are parameters, $X_{i1}, X_{i2}, ..., X_{i,p-1}$ are known constants and the ϵ_i are all independent $N(0, \sigma^2)$. In the given data set, it is possible that the same rider accounts for more than one trip which would violate the assumption that all the trips are independent. The data set does not contain any unique rider ID so it is not possible to definitively determine which observations a single rider may account for. An aggressive approach to dealing with this potential issue is to throw out all observations where the riders birth year, gender and home zip code have already appeared in a previous observation. Although this approach will likely throw out more data than is necessary it will ensure that the remaining observations are all from distinct Hubway users. Once the model has been fitted the analysis turns to diagnostics of the model itself. The first step of this analysis is to plot the fitted values from the model $\hat{Y}_i$ vs the residuals $Y_i - \hat{Y}_i$. The purpose of this step is to analyze whether there is systematic deviation or non-constant variance for the residuals at the various levels of the fitted values $\hat{Y}_i$. To test for non-constant variance the Breusch-Pagan test can be utilized. This procedure test whether the variance of the residuals σ_i is related to X by $log\sigma_i^2 = \gamma_0 + \gamma_1 X$. The null hypothesis is that $\gamma_1 = 0$ and therefore the variance is a constant. Another model diagnostic is a quantile-quantile plot of the residuals against the theoretical quantiles of the normal distribution. This will help elucidate if the residuals are normally distributed in accordance with the general linear model. After this, an F test for the regression relation is performed. This test relies on the test statistic $F* = \frac{MSR}{MSE}$ where MSR is regression mean squared and MSE is error mean squared. The procedure is designed to test if there is a regression relation between the response variable Y and the predictor variables X_i. If appropriate, the following relation can then be used with the model to establish confidence intervals for response at various levels of the predictor variables: $\hat{Y} \pm Ws\{\hat{Y}_h\}$ where $W^2 = pF(1-\alpha; p, n-p)$.

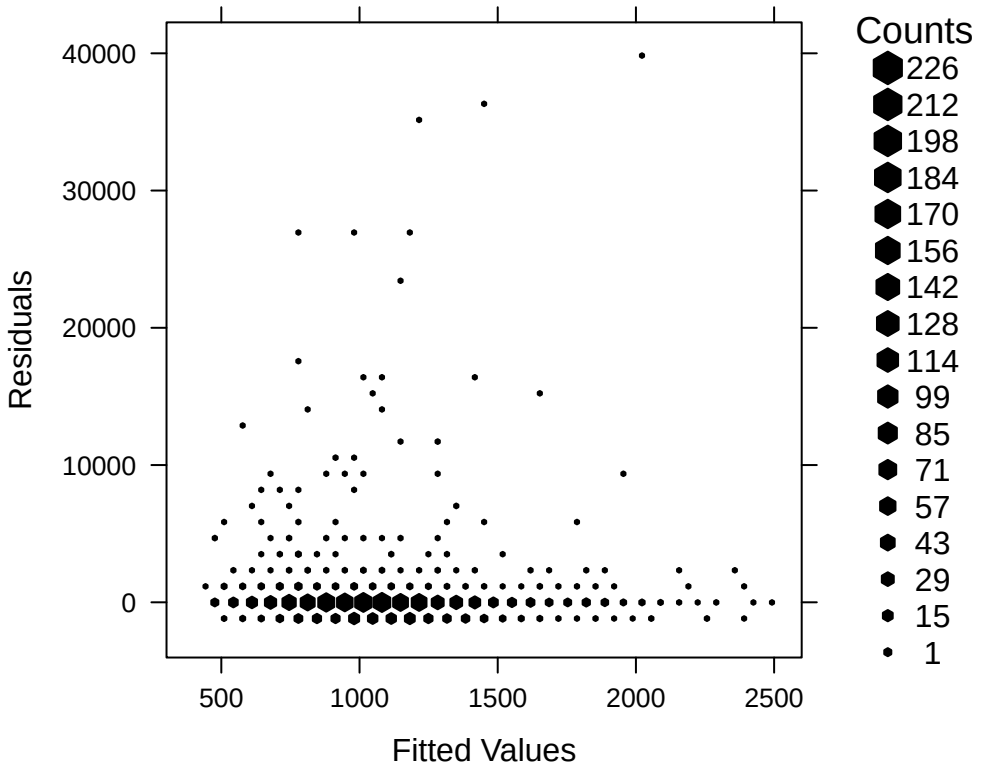
Results:

The estimated values of the regression coefficients for a multiple linear regression model that treats the duration of a trip as the response variable and distance beteween starting and ending station, rider age, and gender as predictor variables are shown in the table below. Rider age is in years, distance is in meters and the trip duration is in seconds. Gender has been recoded as scode where 1 is male and 0 is female.

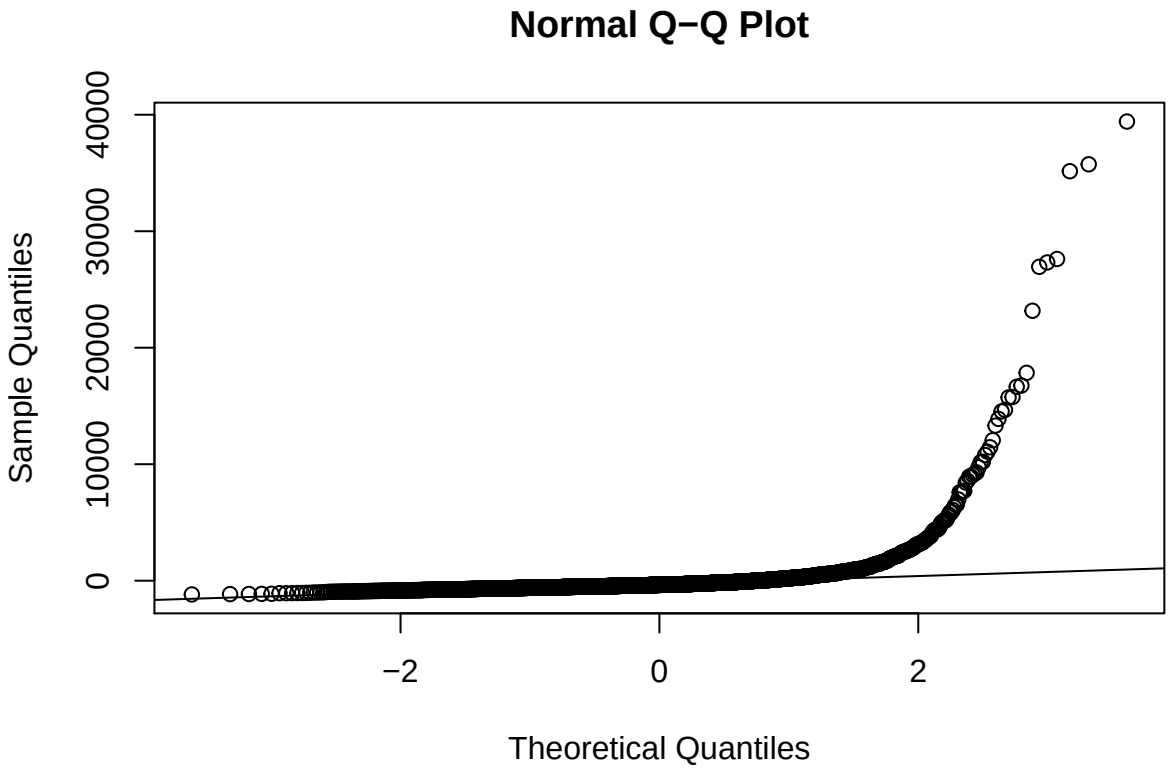
```
##
## Call:
## lm(formula = duration ~ dist + rider_age + scode, data = hubway_model_data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -1181   -532   -358    -63  39414
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  519.2692   133.2546   3.897 9.94e-05 ***
## dist           0.2196     0.0273   8.042 1.22e-15 ***
## rider_age      9.6317     2.8088   3.429 0.000613 ***
## scode1      -249.7586    71.3480  -3.501 0.000470 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1960 on 3290 degrees of freedom
## Multiple R-squared:  0.02561,    Adjusted R-squared:  0.02472
## F-statistic: 28.82 on 3 and 3290 DF,  p-value: < 2.2e-16
```

To examine the appropriateness of the model, the predicted values $\hat{Y}_i$ are plotted against the residuals $Y_i - \hat{Y}_i$.

```
## Warning in fun(maxcnt = quote(226L), trans = quote(NULL), inv =
## quote(NULL), : legend shows relative sizes
```



Observe that the residuals appear to have a few large positive values and many small negative values. Next, the assumption that the residuals are normally distributed is analyzed by a QQplot.



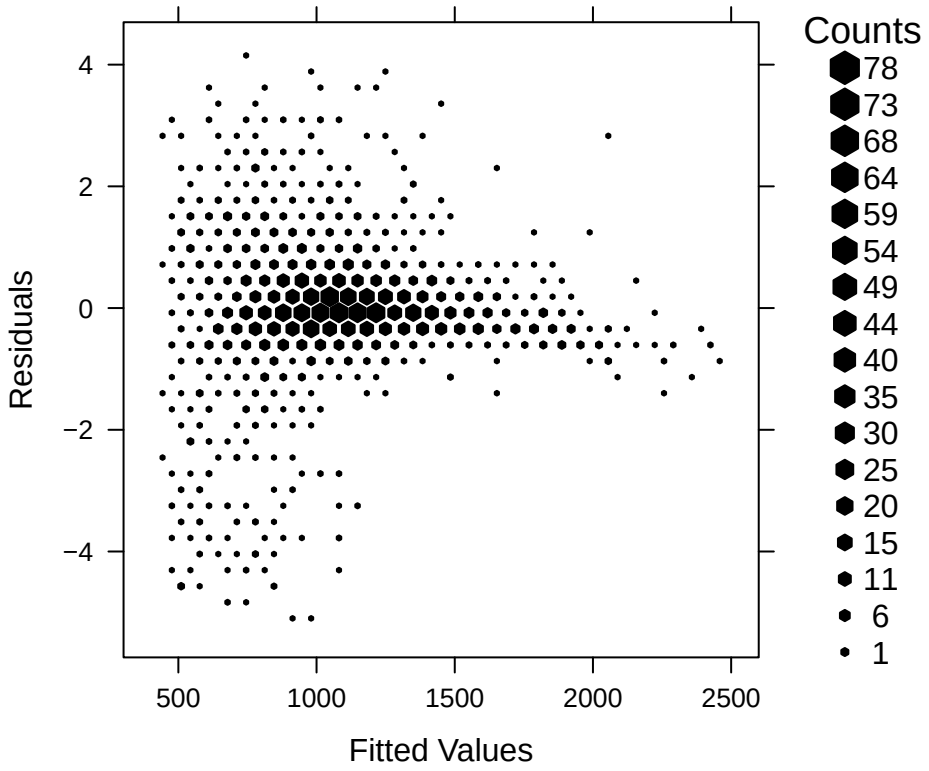
This plot indicates that there are significant deviations from normality for the larger values of residuals. The largest residuals are larger than what would be expected if the residuals were normally distributed. In such cases it may be appropriate to transform the response variable Y and see if the model assumptions might hold more closely for the transformed values. One such common transformation is a log transform of the response variable Y with the same group of predictor variables. This model is formally described as $Y_i` = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_{p-1} X_{i,p-1} + \epsilon_i$ where $Y_i` = log(Y_i)$ and all the other parameters are equivalent to the description given earlier. The summary for the regression coefficients for this model is shown below:

```
##
## Call:
## lm(formula = log(duration) ~ dist + rider_age + scode, data = hubway_model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0991 -0.3161 -0.0315  0.3417  4.0563
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.734e+00  6.497e-02  88.261  < 2e-16 ***
## dist         3.720e-04  1.331e-05  27.949  < 2e-16 ***
## rider_age    9.276e-03  1.369e-03   6.774 1.48e-11 ***
## scode1      -2.763e-01  3.478e-02  -7.942 2.71e-15 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9556 on 3290 degrees of freedom
## Multiple R-squared:  0.2115, Adjusted R-squared:  0.2108
## F-statistic: 294.2 on 3 and 3290 DF,  p-value: < 2.2e-16
```

Next, we examine the same plots as before to verify the model assumptions relating to the residuals.

```
## Warning in fun(maxcnt = quote(78L), trans = quote(NULL), inv =
## quote(NULL), : legend shows relative sizes
```



!

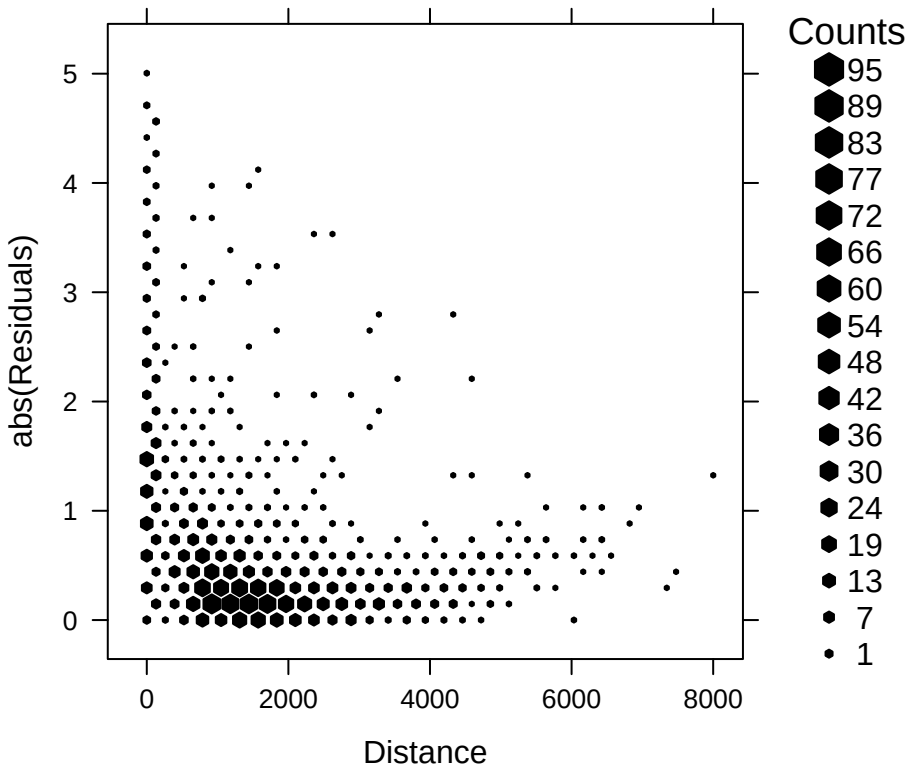
The quantile-quantile plot indicates that the largest and smallest values of the residuals are larger and smaller than what would be expected if the residuals were actually normally distributed. The residuals for the log model appear to be more evenly balanced around 0 however, the variance may not be constant. The Breusch-Pagan test can be used to establish non-constant variance of residuals.

```
## Warning: package 'car' was built under R version 3.2.4
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1049.235    Df = 1     p = 3.568687e-230
```

Using the test we can reject the null hypothesis that the residuals have constant variance with 95% confidence. In such cases a weighted regression may be appropriate. A plot of the absolute residuals against the predictor variable of trip distance reveals that the magnitude of the residuals appears to decrease when distance between beginning and ending station increases.

```
## Warning in fun(maxcnt = quote(95L), trans = quote(NULL), inv =
## quote(NULL), : legend shows relative sizes
```



The distribution appears similar to an exponential decay. This similarity was incorporated into the weights for the weighted regression. After some trial and error a weighting function of $exp(\sqrt{dist})$ was arrived at. Using this function a weighted regression model can be established. The coefficients of this model are summarized below.

```
##
## Call:
## lm(formula = log(duration) ~ dist + rider_age + scode, data = hubway_model_data,
##     weights = exp(sqrt(dist)))
##
## Weighted Residuals:
##        Min         1Q     Median         3Q        Max
## -1.431e+18 -2.536e+08 -6.011e+05  5.607e+03  3.235e+17
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.904e+00  1.767e-01  27.757   <2e-16 ***
## dist         2.035e-04  2.185e-05   9.314   <2e-16 ***
## rider_age    6.350e-02  6.285e-04 101.038   <2e-16 ***
## scode1      -1.075e+00  1.365e-02 -78.768   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 2.782e+16 on 3290 degrees of freedom
## Multiple R-squared:  0.9424, Adjusted R-squared:  0.9424
## F-statistic: 1.796e+04 on 3 and 3290 DF,  p-value: < 2.2e-16
```

The Breusch-Pagan test is again used to check for non-constant variance of the error terms

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.4811524    Df = 1     p = 0.4879008
```

We fail to reject the null hypothesis that the error terms have constant variance. Next, it is appropriate to test if there is a regression relation between the predictor and response variables using the test statistic $F* = \frac{MSR}{MSE}$.

```
##    value    numdf    dendf
## 17955.17     3.00  3290.00
```

The pvalue of F* is 0+. This indicates that there is a linear relationship between the predictor variables and the response variable. A correlation matrix with the predictor variables of rider age and distance between beginning and ending stations to find if multicolinearity exists.

```
##                  dist   rider_age
## dist       1.00000000 -0.06527881
## rider_age -0.06527881  1.00000000
```

This shows that the predictor variables of inter station distance and rider age aren't strongly correlated with one another. At this point it is appropriate to use the model to make predictions about new observations. Suppose Hubway were installing a new station and wanted to estimate the typical trip duration to a station 1000m away for various demographic groups of their users. For example, suppose they wanted to predict trip duration for males of age 25. The model is evaluated with the given parameters for the predictor variables. To interpret the result it is necessary to use the exponential function to transform the data back to the original scale. The units of the output are seconds.

```
##        fit      lwr      upr
## 1 520.4499 389.9309 694.6567
```

As discussed previously, the residuals of the regression relation appear to vary with the distance between starting and ending station of the trip. It would be useful to establish how the other predictor variables relate to trip duration when this distance is held constant. For this analysis trips between only a single starting and ending station are examined, removing potentially non-independent observations as before to ensure the independence of the data. The starting and ending pair of stations that account for the greatest number of observations are TD Garden (station #38) as the starting point with South Station (station #22) as the ending point. The following graph shows the relation between trip duration and rider age between these two stations with the observations color coded by gender.

It appears that trips by males (scode 1) are generally shorter in duration than those by females (scode 0) although the 95% confidence bands for the regression lines do have significant overlap. The trip duration also does seem to generally increase with age.

Discussion:

The most important takeaway of the previous analysis is the difficulty in using linear regression to model the duration of Hubway users trips based on their age, gender and distance between stations. The main technical issue is that the first couple of attempts to create regression models all violated the assumptions of linear regression. Specifically, the variance of the residuals of these models seemed to dramatically decrease as the distance of the trip increased. Part of the difficulty with the analysis is that Hubway riders have different purposes for their trips which can affect duration. For example, some users may be interested in using Hubway to get to work where as others may be using the service for sightseeing. This variation may account for the shortcomings of the regression model. This also elucidates the more general difficulty of trying to model human behavior with techniques like linear regression. Because there can be so much variability in human behavior these techniques often lack the explanatory power they may have in more predictable domains. The data set seems to lend itself readily to network analysis techniques. As an extension of the previous analysis, it could be worthwhile to try and model the data set as a weighted graph where the stations are nodes and the weighted edges between stations represent the distance between them or the probability that a trip starting at one station will end at the other. Such analysis could potentially provide more insight and lead to a deeper understanding of the usage habits of users of bike sharing services.

Appendix:

```
#Code and additional plots


#radius of the earth in meters.  This is used in the function to calculate distance based on
```

```r
#longitude and latitude of the starting and ending station of a trip
EARTH_RADIUS <- 6371000

#convert radians to degrees
rad2deg <- function(rad) {(rad * 180) / (pi)}

#convert degrees to radians
deg2rad <- function(deg) {(deg * pi) / (180)}

#the haversine function is used to give the distance in meters between a starting and ending
#longitude and latitude.  See https://en.wikipedia.org/wiki/Haversine_formula.
haversine <- function(lat1,long1,lat2,long2){
  phi1 <- deg2rad(lat1)

  phi2 <- deg2rad(lat2)

  delta_phi <- deg2rad(lat2 - lat1)

  delta_lambda <- deg2rad(long2 - long1)

  a <- (sin(delta_phi/2))^2 +
    cos(phi1)*cos(phi2) *
    (sin(delta_lambda/2))^2

  c <- 2 * atan2(sqrt(a), sqrt(1 - a))

  EARTH_RADIUS * c
}

library(plyr)

library(dplyr)

library(tidyr)

#observations of all trips taken
hubway_trips <- read.csv("C:/Users/Josh/Desktop/CS/NU/CS7280/hubway_2011_07_through_2013_11/hubway_trips

#station information, including station name and number, longitude and latitude
hubway_stations <- read.csv("C:/Users/Josh/Desktop/CS/NU/CS7280/hubway_2011_07_through_2013_11/hubway_s

#filter out trips lasting longer than 12 hours and remove any trips that have na values.
hubway_trips <- hubway_trips %>% na.omit() %>% filter(duration < 43200)

summary(hubway_trips)
```

```
##      seq_id         hubway_id         status          duration
##  Min.   :     1   Min.   :     8   Closed:350391   Min.   :    0.0
##  1st Qu.:153290   1st Qu.:173370                   1st Qu.:  346.0
##  Median :279593   Median :319312                   Median :  531.0
##  Mean   :283146   Mean   :321007                   Mean   :  680.1
##  3rd Qu.:414763   3rd Qu.:469314                   3rd Qu.:  825.0
##  Max.   :549286   Max.   :620312                   Max.   :42711.0
```

```
##
##               start_date       strt_statn                   end_date
##  9/13/2012 08:14:00:    16   Min.   : 3.00   6/26/2012 17:45:00:    15
##  8/21/2012 17:05:00:    15   1st Qu.:22.00   8/20/2012 08:35:00:    14
##  8/8/2012 07:55:00 :    15   Median :38.00   8/21/2012 08:54:00:    14
##  9/14/2012 17:11:00:    15   Mean   :36.73   5/29/2012 17:13:00:    13
##  9/24/2012 17:09:00:    15   3rd Qu.:50.00   8/15/2012 07:57:00:    13
##  9/12/2012 17:08:00:    14   Max.   :98.00   8/16/2012 17:16:00:    13
##  (Other)           :350301                   (Other)           :350309
##    end_statn        bike_nr            subsc_type        zip_code
##  Min.   : 3.00   B00401 :   716   Casual    :     0   '02118 : 52584
##  1st Qu.:22.00   B00145 :   694   Registered:350391   '02215 : 33371
##  Median :38.00   B00123 :   693                       '02116 : 30898
##  Mean   :36.69   B00444 :   676                       '02115 : 20728
##  3rd Qu.:50.00   B00107 :   674                       '02113 : 15087
##  Max.   :98.00   B00079 :   667                       '02114 : 12736
##                  (Other):346271                       (Other):184987
##    birth_date      gender
##  Min.   :1932         :     0
##  1st Qu.:1969   Female: 86945
##  Median :1979   Male  :263446
##  Mean   :1976
##  3rd Qu.:1985
##  Max.   :1995
##
```

```
#Add distance between starting and ending station as a new column

#Couldn't get join to work with columns of different names so create two new data frames from the
#hubway_stations data and join on those new column names
strt_hubway_stations <- plyr::rename(hubway_stations, replace = c("id" = "strt_statn",
  "lat" = "strt_lat", "lng" = "strt_lng")) %>%
  select(-station, -terminal,-municipal,-status)

end_hubway_stations <- plyr::rename(hubway_stations, replace = c("id" = "end_statn",
 "lat" = "end_lat", "lng" = "end_lng")) %>%
  select(-station, -terminal,-municipal,-status)

hubway_trips <- join(hubway_trips, strt_hubway_stations)
```

```
## Joining by: strt_statn
```

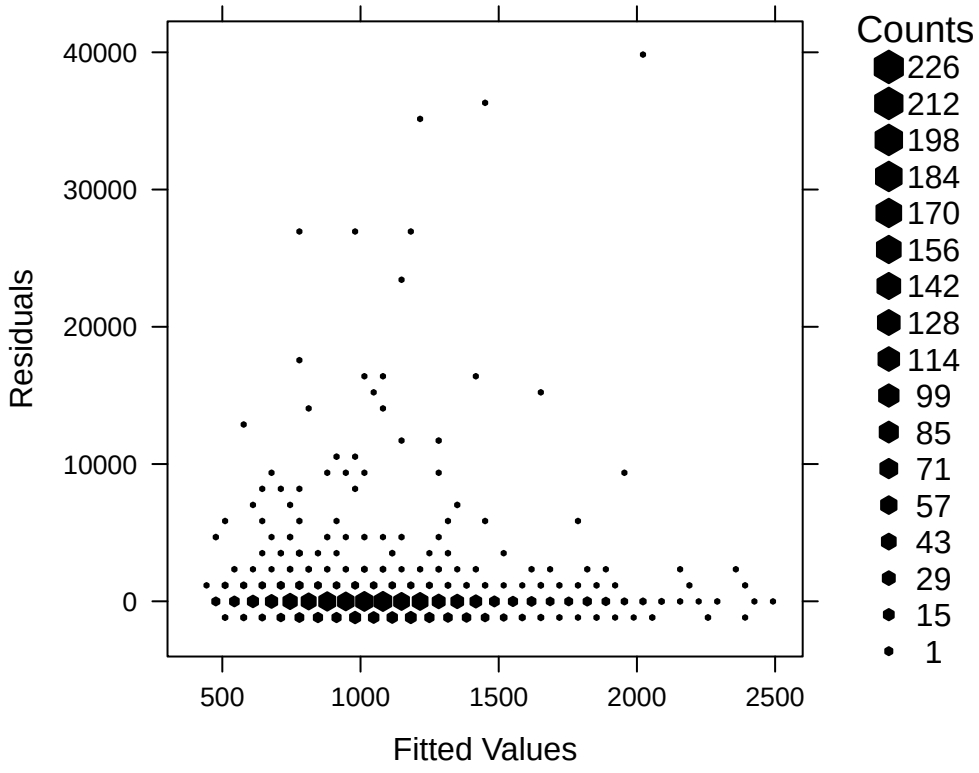
```
hubway_trips <- join(hubway_trips, end_hubway_stations)
```

```
## Joining by: end_statn
```

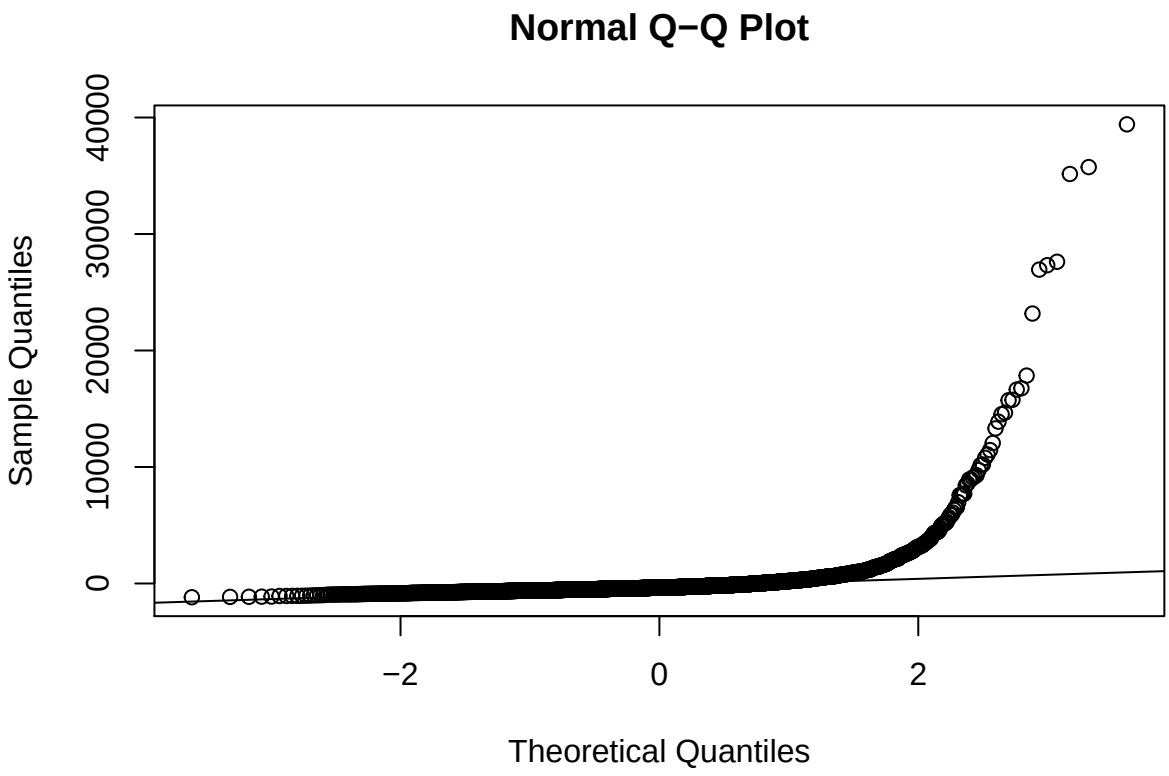
```
#add column with trip distance in meters
hubway_trips <- mutate(hubway_trips, dist = haversine(strt_lat, strt_lng, end_lat, end_lng))

#remove all observations that are duplicates of earlier observations based on zip code, gender
#and birth year. As described in the methods section, this is to ensure the independence of the data
deduped.hubway_trips <- hubway_trips[!duplicated(hubway_trips[,c('zip_code','birth_date','gender')]),]
```

```r
#recode male and female as 1 and 0 to simplify analysis
deduped.hubway_trips$scode <- revalue(deduped.hubway_trips$gender, c("Male" = 1, "Female" = 0))

#begin process of adding new column with rider age by first splitting start_date
#into the day, month and year in one column and time in another column
temporary_data_1 <- deduped.hubway_trips %>% tidyr::separate(start_date,
  into=c("start_day_mon_year","start_time")," ")

#continue by splitting the day month and year into 3 separate columns
temporary_data_2 <- temporary_data_1 %>% tidyr::separate(start_day_mon_year, into=c("start_day", "start_

#calculate the appoximate age of the rider by subtracting the year in which the observation
#was recorded from the birth year of the rider.  There is some subtlety here since, for
#example someone born in 1988 could be 3 different possible ages in 2012 depending on when
#their birthday is.  Since we don't have this information we just subtract the two years from
#one another.  This will likely even out on average.
hubway_model_data <- temporary_data_2 %>% dplyr::mutate(rider_age = as.numeric(start_year) - birth_date)

library(hexbin)

#create regression model with trip duration as response variable and rider age,
#trip distance and gender as predictor variables
hubway_fit <- lm(duration ~ dist+rider_age+scode, data = hubway_model_data)

#create regression model with the log of trip duration as response variabel and rider age,
#trip distance and gender as predictor variables
log_hubway_fit <- lm(log(duration)~dist+rider_age+scode,data=hubway_model_data)

#plot residuals vs fitted values to check for non-constant error variance
hexbinplot(hubway_fit$residuals~hubway_fit$fitted.values, aspect=1, bins=50,
           xlab = "Fitted Values", ylab="Residuals", style="lattice")
```

```
## Warning in fun(maxcnt = quote(226L), trans = quote(NULL), inv =
## quote(NULL), : legend shows relative sizes
```




```r
#check for normality of residuals
qqnorm(hubway_fit$residuals)
qqline(hubway_fit$residuals)
```

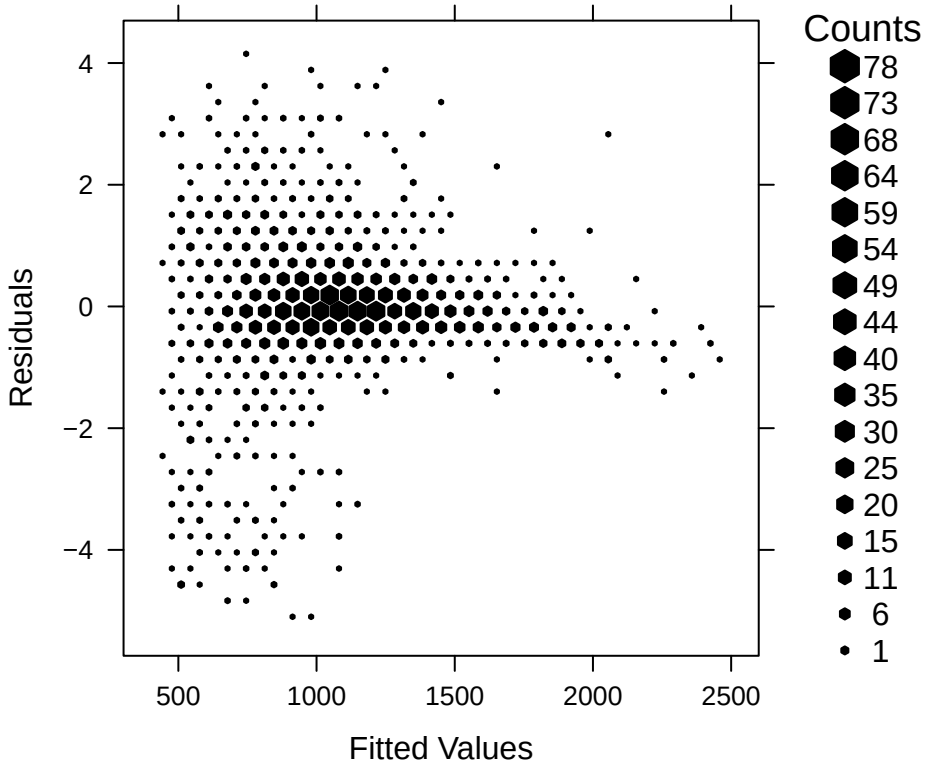



```
#summary of log model
summary(log_hubway_fit)
```

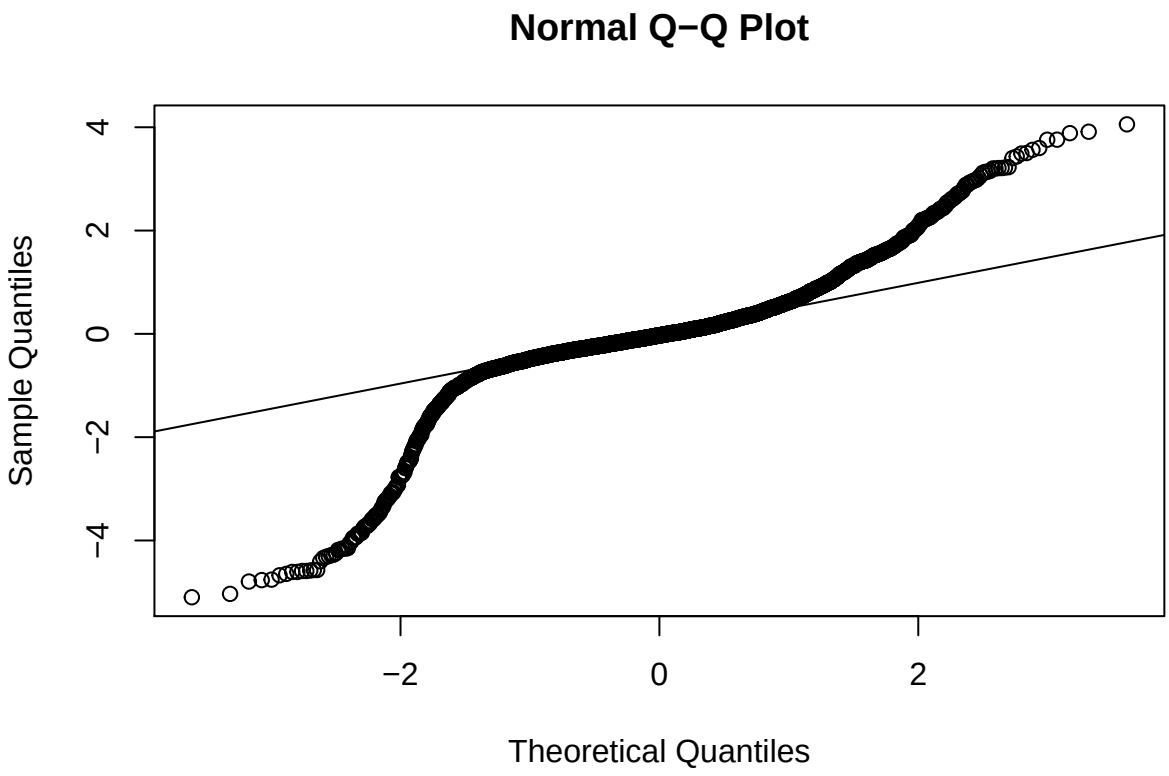
```
##
## Call:
## lm(formula = log(duration) ~ dist + rider_age + scode, data = hubway_model_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0991 -0.3161 -0.0315  0.3417  4.0563
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.734e+00  6.497e-02  88.261  < 2e-16 ***
## dist         3.720e-04  1.331e-05  27.949  < 2e-16 ***
## rider_age    9.276e-03  1.369e-03   6.774 1.48e-11 ***
## scode1      -2.763e-01  3.478e-02  -7.942 2.71e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9556 on 3290 degrees of freedom
## Multiple R-squared:  0.2115, Adjusted R-squared:  0.2108
## F-statistic: 294.2 on 3 and 3290 DF,  p-value: < 2.2e-16
```

```r
hexbinplot(log_hubway_fit$residuals~hubway_fit$fitted.values, aspect=1, bins=50, xlab = "Fitted Values"
```

```
## Warning in fun(maxcnt = quote(78L), trans = quote(NULL), inv =
## quote(NULL), : legend shows relative sizes
```




```r
qqnorm(log_hubway_fit$residuals)
qqline(log_hubway_fit$residuals)
```

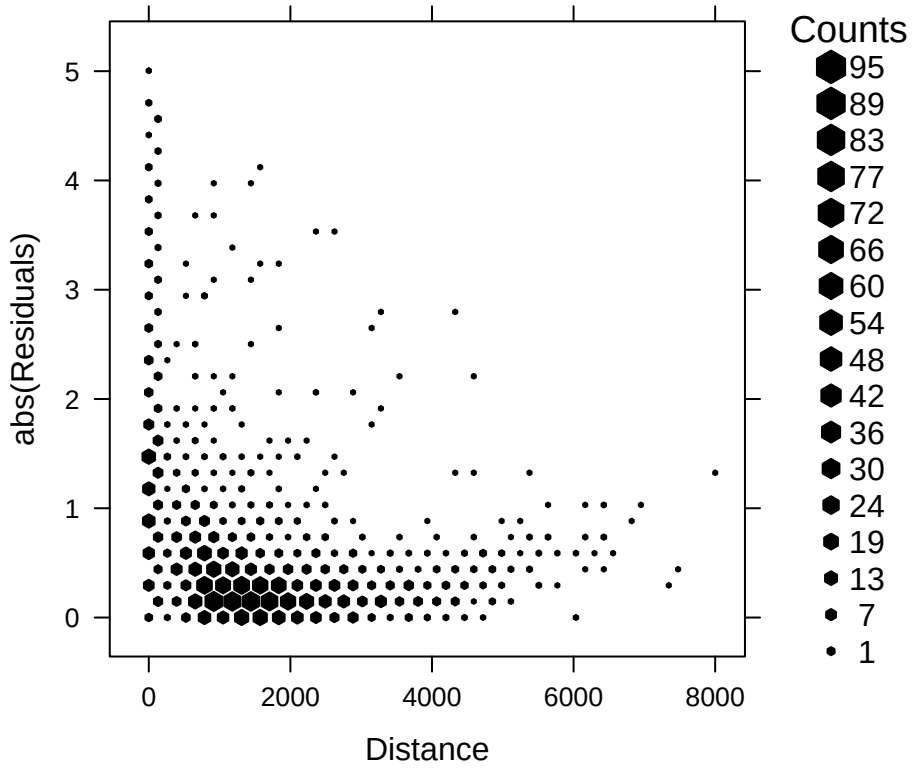



```
library(car)

#test log model for non-constant variance
ncvTest(log_hubway_fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 1049.235    Df = 1     p = 3.568687e-230
```

```
#plot residuals against distance
hexbinplot(abs(log_hubway_fit$residuals)~hubway_model_data$dist, aspect=1, bins=50, xlab = "Distance", y
```

```
## Warning in fun(maxcnt = quote(95L), trans = quote(NULL), inv =
## quote(NULL), : legend shows relative sizes
```




```
#test weighted model for non-constant variance
ncvTest(weighted_log_hubway_fit)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.4811524    Df = 1     p = 0.4879008
```

```
summary(weighted_log_hubway_fit)$fstatistic
```

```
##    value    numdf    dendf
## 17955.17     3.00  3290.00
```

```
#correlations between predictor variables
cor(select(hubway_model_data, dist, rider_age))
```

```
##                  dist   rider_age
## dist       1.00000000 -0.06527881
## rider_age -0.06527881  1.00000000
```

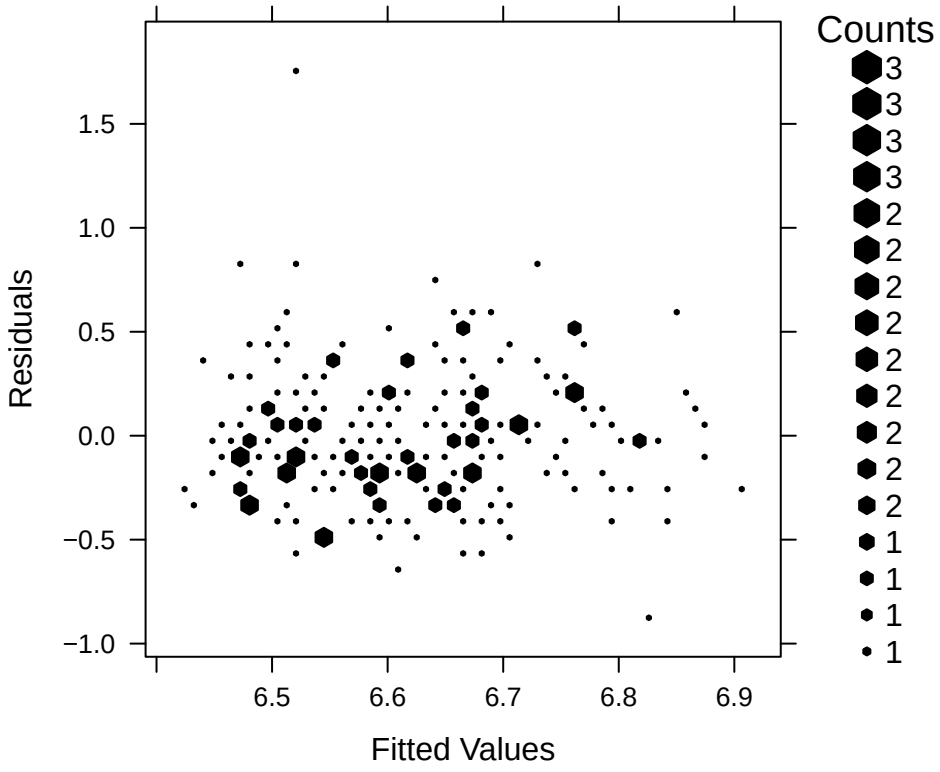
```
#create new data point
new_data_point <- data.frame(rider_age = c(35), dist = c(1000), scode = as.factor(c(1)))

#confidence interval for predicted response at new data point
exp(predict.lm(weighted_log_hubway_fit, newdata = new_data_point, interval="confidence"))
```

```
##        fit      lwr      upr
## 1 520.4499 389.9309 694.6567
```

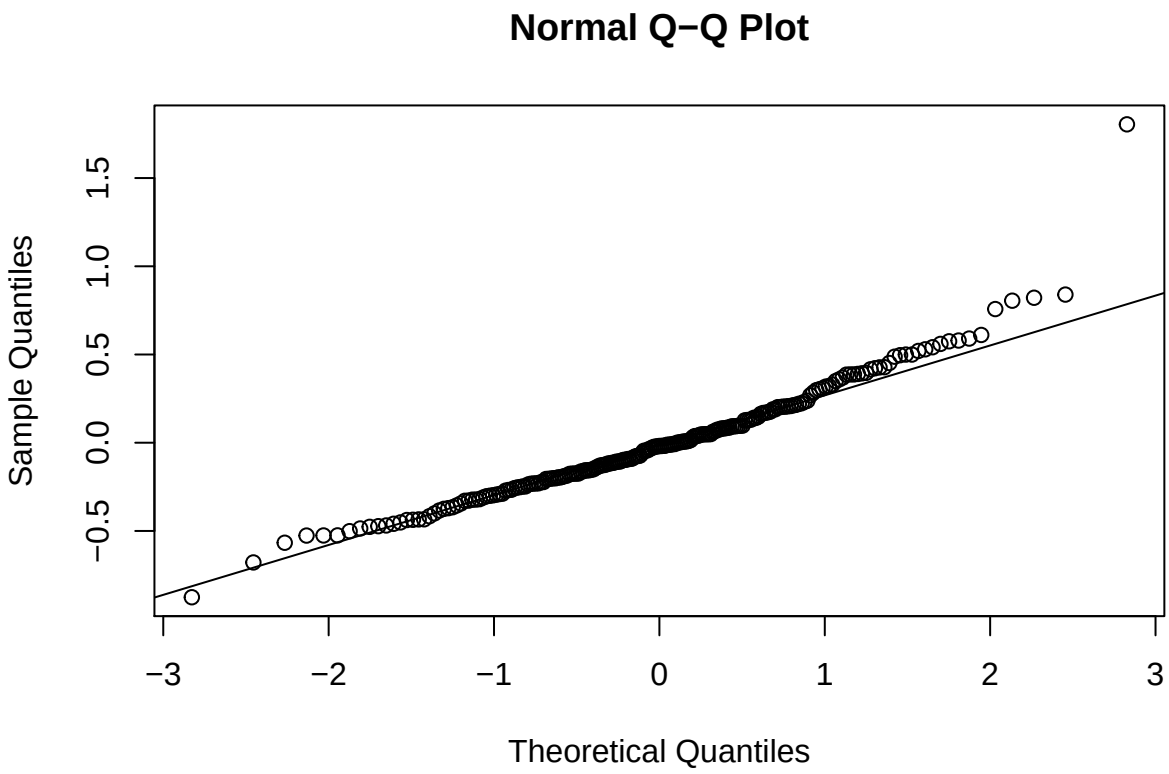
```
#select all trips beginning at station #38 and ending at #22
data.38_to_22_trips <- hubway_trips %>% filter(strt_statn == 38) %>% filter(end_statn ==22)

#remove any observations possibly from the same rider to ensure independence of the data
deduped.38_to_22_trips <- data.38_to_22_trips[!duplicated(data.38_to_22_trips[,c('zip_code','birth_date

deduped.38_to_22_trips$scode <- revalue(deduped.38_to_22_trips$gender, c("Male" = 1, "Female" = 0))

#begin process of adding new column with rider age by first splitting start_date into the day, month an
#year in one column and time in another column
temporary_data_3 <- deduped.38_to_22_trips %>% tidyr::separate(start_date, into=c("start_day_mon_year",'

#continue by splitting the day month and year into 3 separate columns
temporary_data_4 <- temporary_data_3 %>% tidyr::separate(start_day_mon_year, into=c("start_day", "start_

hubway_model_data_38_to_22 <- temporary_data_4 %>% dplyr::mutate(rider_age = as.numeric(start_year) - bi


library(ggplot2)

ggplot(select(hubway_model_data_38_to_22, duration,dist,scode,rider_age), aes(x=rider_age, y=log(duratio
```

```
#create regression model from hubway_model_data_38_to_22 and test for non-constant variance
log_hubway_fit_38_to_22 <- lm(log(duration)~rider_age+scode,data=hubway_model_data_38_to_22)

#plot of fitted values vs residuals for log_hubway_fit_38_to_22
hexbinplot(log_hubway_fit_38_to_22$residuals~log_hubway_fit_38_to_22$fitted.values, aspect=1, bins=50, i
```

```
## Warning in fun(maxcnt = quote(3L), trans = quote(NULL), inv =
## quote(NULL), : legend shows relative sizes
```




```
#qqplot for log_hubway_fit_38_to_22
qqnorm(log_hubway_fit_38_to_22$residuals)
qqline(log_hubway_fit_38_to_22$residuals)
```




```
#plot of Hubway stations on map:
library(ggmap)
```

```
## Warning: package 'ggmap' was built under R version 3.2.4
```

```
citation('ggmap')
```

```
##
## To cite ggmap in publications, please use:
##
##   D. Kahle and H. Wickham. ggmap: Spatial Visualization with
##   ggplot2. The R Journal, 5(1), 144-161. URL
##   http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     author = {David Kahle and Hadley Wickham},
##     title = {ggmap: Spatial Visualization with ggplot2},
##     journal = {The R Journal},
##     year = {2013},
##     volume = {5},
##     number = {1},
##     pages = {144--161},
```

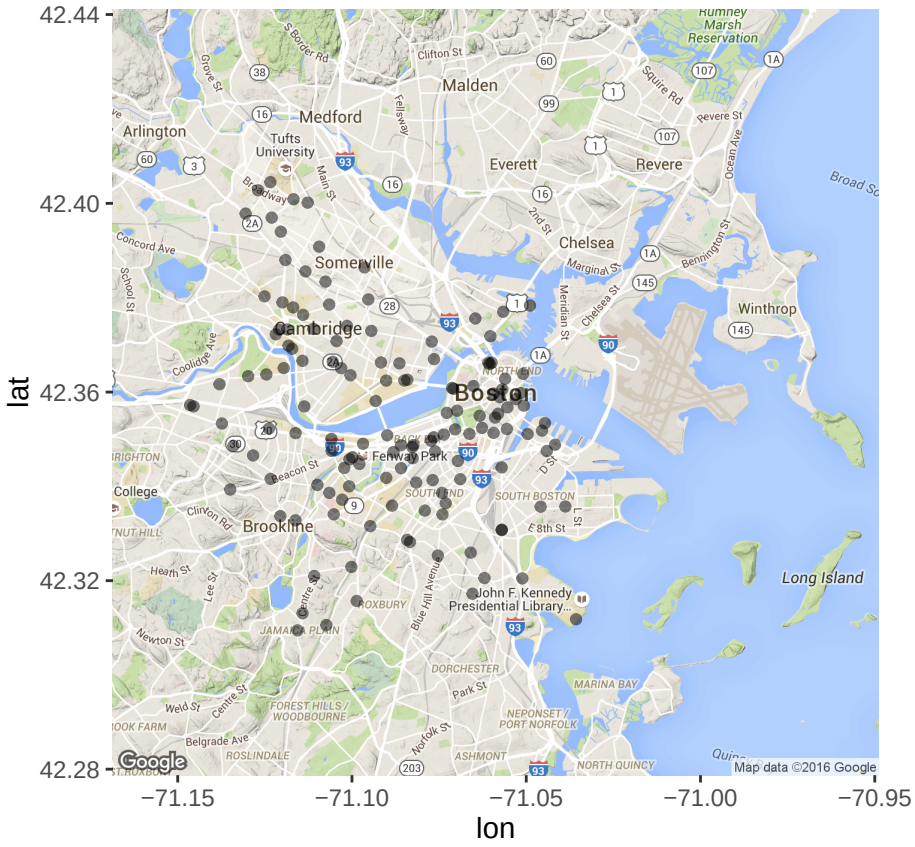
```
##     url = {http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf},
##   }
```

```
#####################################################
#Assorted visualizations of data set

map <- get_map(location = "Boston", zoom = 12)
```

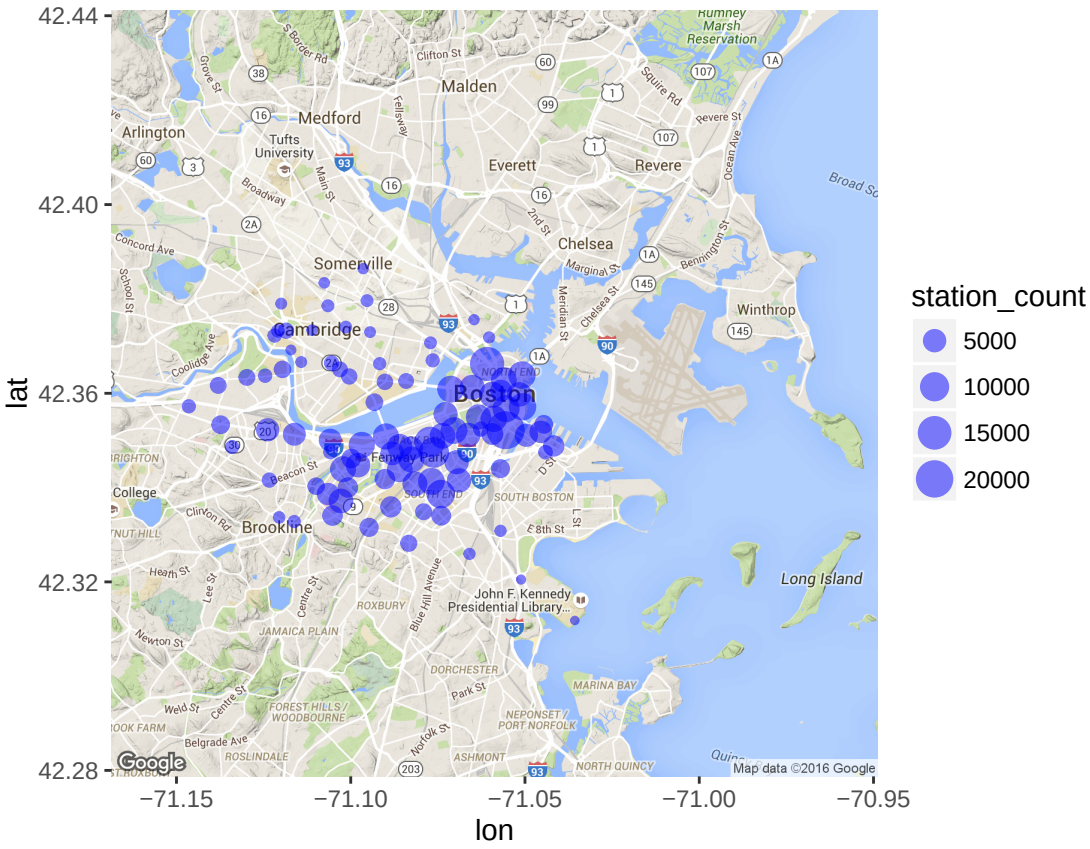
```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Boston&zoom=12&size=640x640&scal
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Boston&sensor=false
```

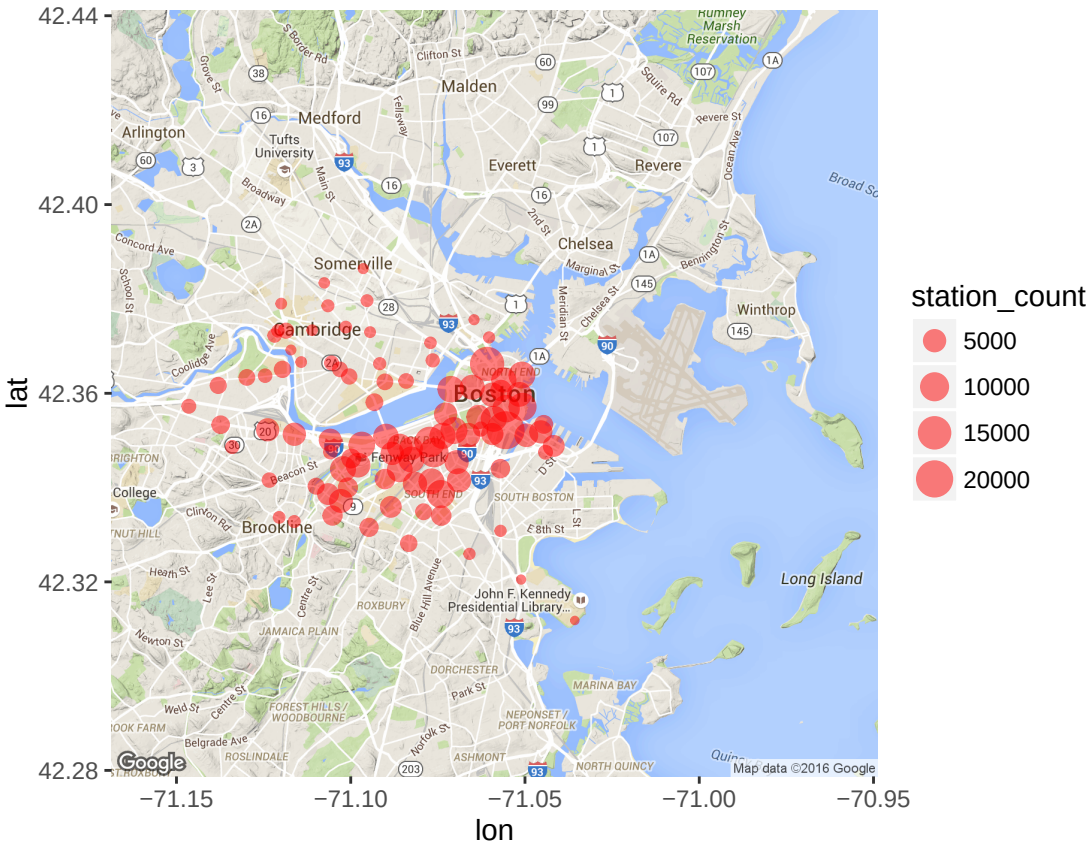
```
ggmap(map) + geom_point(aes(x = lng, y = lat), data = hubway_stations, alpha=.5)
```



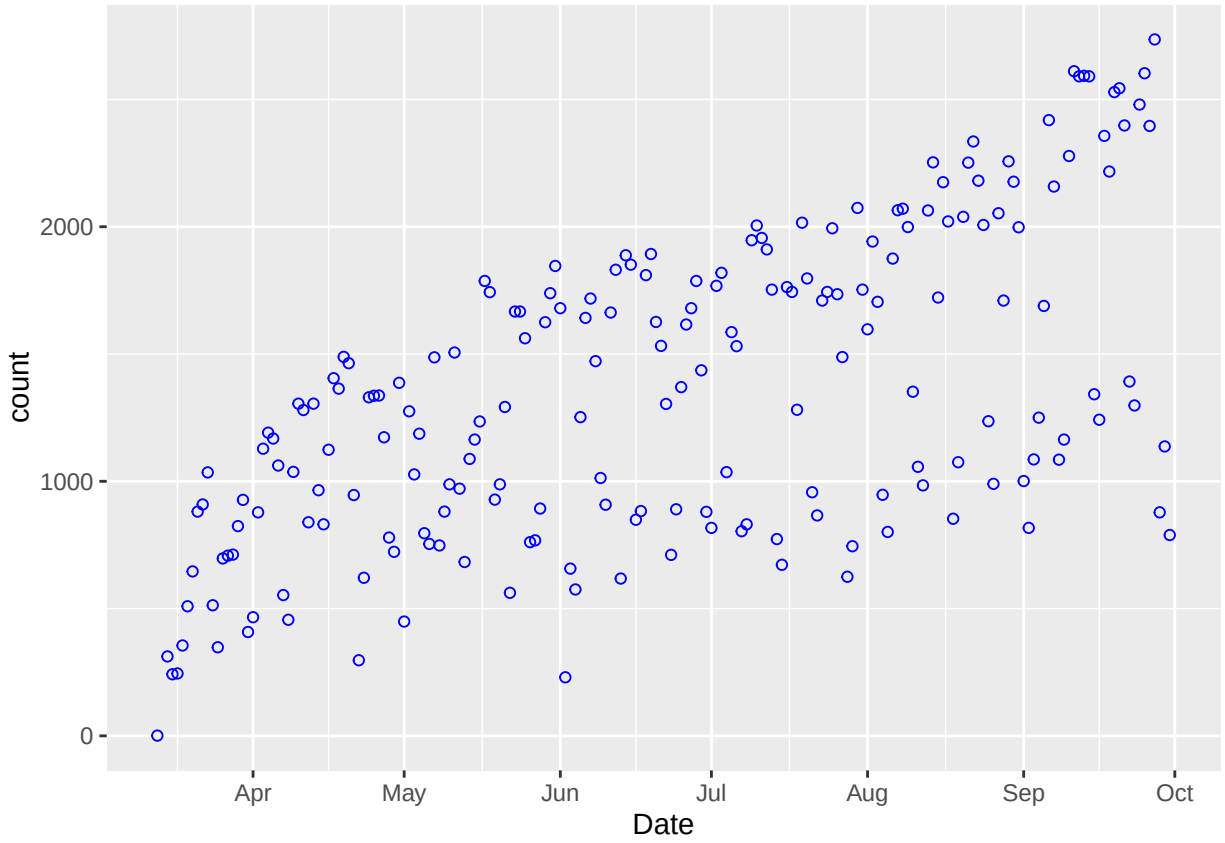

```
start_hubway_trips <- hubway_trips %>% group_by(strt_statn, strt_lat, strt_lng) %>% summarise(station_c

#plot of stations with size relative to number of trips beginning at that station
ggmap(map) + geom_point(aes(x = strt_lng, y = strt_lat, size=station_count), data = start_hubway_trips,
```




```
end_hubway_trips <- hubway_trips %>% group_by(end_statn, end_lat, end_lng) %>% summarise(station_count =

#plot of stations with size relative to number of trips ending at that station
ggmap(map) + geom_point(aes(x = end_lng, y = end_lat, size=station_count), data = end_hubway_trips, alp
```




```
#plot of total number of daily trips taken vs. Date for 2012
daily_hubway_trips <- hubway_trips %>%
  tidyr::separate(start_date, into=c("start_day_mon_year","start_time")," ") %>%
  filter(seq_id > 140521) %>%
  group_by(as.Date(start_day_mon_year, "%m/%d/%Y")) %>%
  summarise(count = n())

colnames(daily_hubway_trips)[1] <- "Date"

ggplot(daily_hubway_trips, aes(x=Date, y=count)) + geom_point(shape=1, colour="blue")
```



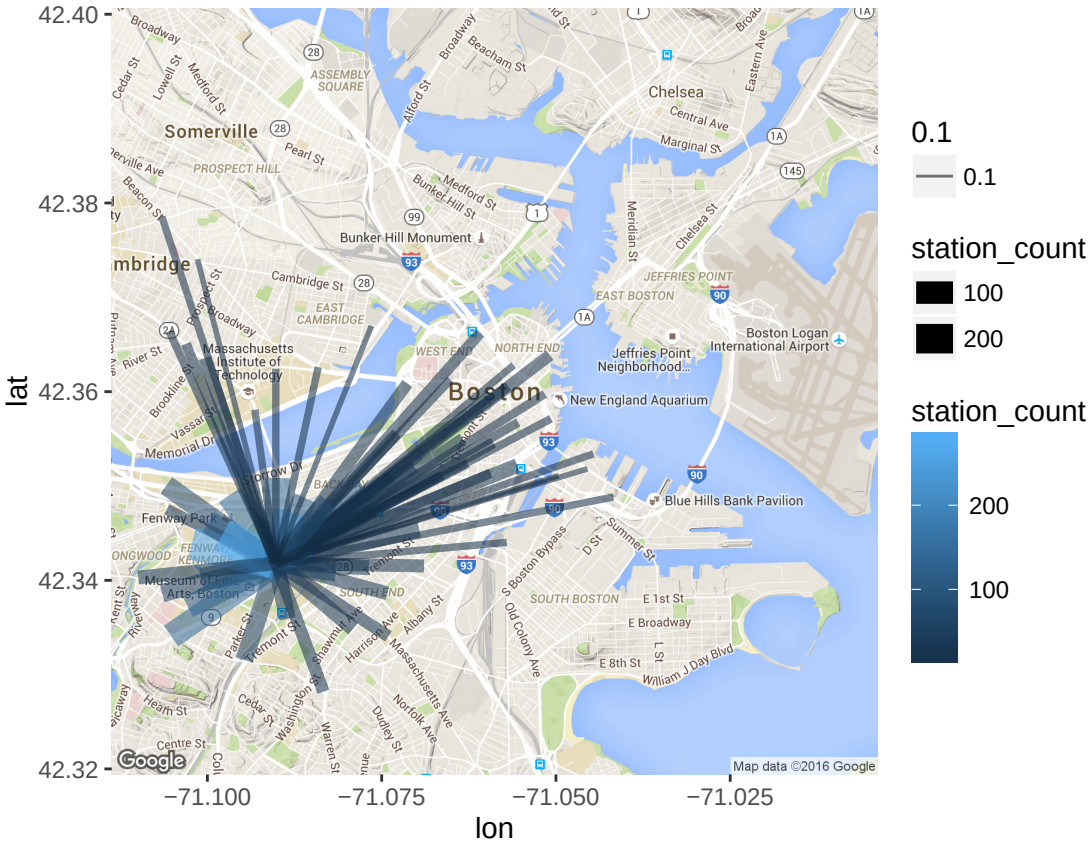

```
#get all trip starting at Northeastern
NEU_hubway_trips <- hubway_trips %>% filter(strt_statn==5)

NEU_hubway_trips <- NEU_hubway_trips %>% group_by(end_statn, end_lat, end_lng, strt_lat, strt_lng) %>% s

map <- get_map(location = "Boston", zoom = 13)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Boston&zoom=13&size=640x640&scale
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Boston&sensor=false
```

```
#map all trip starting at Northeastern
ggmap(map) + geom_segment(aes(x = strt_lng, xend = end_lng , y = strt_lat, yend= end_lat, size = statio
```

```
## Warning: Removed 14 rows containing missing values (geom_segment).
```


42.40
42.38
42.36
42.34
42.32
lat
−71.100
−71.075
−71.050
−71.025
lon
Somerville
Chelsea
Boston
Bunker Hill Monument
Massachusetts Institute of Technology
New England Aquarium
Jeffries Point Neighborhood...
Boston Logan International Airport
Blue Hills Bank Pavilion
Fenway Park
Museum of Fine Arts, Boston
Map data ©2016 Google
0.1
0.1
station_count
100
200
station_count
200
100