Data Analysis of Hubway bike trips

Elena Ines Saez Papachristou

Bike sharing programs stand as pillars in the sharing economy, mirroring the efficiency and benefits seen in other sharing initiatives like car-sharing programs. By offering convenient access to bicycles without the burden of ownership or maintenance expenses, bike shares contribute to both economic and environmental well-being. They encourage more people to cycle, reducing reliance on cars, and preventing an excess of unused bicycles from being produced.

Hubway is Boston's most successful bike sharing program. The company asks customers to pay a relatively modest annual or monthly fee, and in return gives customers access to bicycles parked at stations across Boston, Brookline, Cambridge, and Somerville. Additionally, 24-hour and 72-hour passes are available for purchase by non-members. Customers can take short rides for free and pay a nominal hourly rate for any journey lasting more than 30 minutes. Typically, riders will pick up a bike from a "dock" in one part of the city and drop it off at another dock.

By 2016, Hubway operated 185 stations and 1750 bicycles, with 5 million rides since launching in 2011.

In April 2017, Hubway held a Data Visualization Challenge at the Microsoft NERD Center in Cambridge, releasing trip data.

I will analyse data from Hubway trips.

Reading the data files `hubway_stations.csv` and `hubway_trips.csv` into separate dataframes.

```r
library(dplyr)
library(ggplot2)
```

```r
stations <- read.csv("hubway_stations.csv")

trips <- read.csv("hubway_trips.csv")
```

Taking a closer look at all the columns and understanding their types.

```r
str(stations)
```

```
## 'data.frame':    142 obs. of  7 variables:
##  $ id       : int  3 4 5 6 7 8 9 10 11 12 ...
##  $ terminal : chr  "B32006" "C32000" "B32012" "D32000" ...
##  $ station  : chr  "Colleges of the Fenway" "Tremont St. at Berkeley St." "Northeastern U / North Parking Lot"
"Cambridge St. at Joy St." ...
##  $ municipal: chr  "Boston" "Boston" "Boston" "Boston" ...
##  $ lat      : num  42.3 42.3 42.3 42.4 42.4 ...
##  $ lng      : num  -71.1 -71.1 -71.1 -71.1 -71 ...
##  $ status   : chr  "Existing" "Existing" "Existing" "Existing" ...
```

```r
str(trips)
```

```
## 'data.frame':    674350 obs. of  13 variables:
##  $ seq_id    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ hubway_id : int  8 9 10 11 12 13 14 15 16 17 ...
##  $ status    : chr  "Closed" "Closed" "Closed" "Closed" ...
##  $ duration  : int  9 220 56 64 12 19 24 7 8 1108 ...
##  $ start_date: chr  "2011-07-28 10:12:00" "2011-07-28 10:21:00" "2011-07-28 10:33:00" "2011-07-28 10:35:00" ..
.
##  $ strt_statn: int  23 23 23 23 23 23 23 23 23 47 ...
##  $ end_date  : chr  "2011-07-28 10:12:00" "2011-07-28 10:25:00" "2011-07-28 10:34:00" "2011-07-28 10:36:00" ..
.
##  $ end_statn : int  23 23 23 23 23 23 23 23 23 40 ...
##  $ bike_nr   : chr  "B00468" "B00554" "B00456" "B00554" ...
##  $ subsc_type: chr  "Registered" "Registered" "Registered" "Registered" ...
##  $ zip_code  : chr  "'97217" "'02215" "'02108" "'02116" ...
##  $ birth_date: int  1976 1966 1943 1981 1983 1951 1971 1971 1983 1994 ...
##  $ gender    : chr  "Male" "Male" "Male" "Female" ...
```

Getting some statistical information from the stations and trips data

```r
summary(stations)
```

```
##        id           terminal           station           municipal
##  Min.   :  3.00   Length:142         Length:142         Length:142
##  1st Qu.: 39.25   Class :character   Class :character   Class :character
##  Median : 74.50   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 74.32
##  3rd Qu.:109.75
##  Max.   :145.00
##       lat             lng            status
##  Min.   :42.31   Min.   :-71.15   Length:142
##  1st Qu.:42.34   1st Qu.:-71.11   Class :character
##  Median :42.35   Median :-71.09   Mode  :character
##  Mean   :42.35   Mean   :-71.09
##  3rd Qu.:42.37   3rd Qu.:-71.07
##  Max.   :42.40   Max.   :-71.04
```

```
summary(trips)
```

```
##      seq_id         hubway_id         status             duration
##  Min.   :     1   Min.   :     8   Length:674350      Min.   :   -6660
##  1st Qu.:168588   1st Qu.:191561   Class :character   1st Qu.:     405
##  Median :337176   Median :382519   Mode  :character   Median :     663
##  Mean   :337176   Mean   :381807                      Mean   :    1560
##  3rd Qu.:505763   3rd Qu.:571535                      3rd Qu.:    1161
##  Max.   :674350   Max.   :761917                      Max.   :11994458
##
##   start_date          strt_statn       end_date           end_statn
##  Length:674350      Min.   :  3.00   Length:674350      Min.   :  3.00
##  Class :character   1st Qu.: 22.00   Class :character   1st Qu.: 22.00
##  Mode  :character   Median : 40.00   Mode  :character   Median : 40.00
##                     Mean   : 41.18                      Mean   : 41.05
##                     3rd Qu.: 54.00                      3rd Qu.: 54.00
##                     Max.   :141.00                      Max.   :141.00
##                     NA's   :14                          NA's   :45
##    bike_nr           subsc_type          zip_code          birth_date
##  Length:674350      Length:674350      Length:674350      Min.   :1932
##  Class :character   Class :character   Class :character   1st Qu.:1969
##  Mode  :character   Mode  :character   Mode  :character   Median :1979
##                                                           Mean   :1976
##                                                           3rd Qu.:1985
##                                                           Max.   :1995
##                                                           NA's   :323706
##     gender
##  Length:674350
##  Class :character
##  Mode  :character
##
##
##
##
```

Removing all the rows with null values in any one (or more) of the columns and creating a new dataframe with the name `trips_clean` and `stations_clean`.

```
trips_clean <- filter(trips, rowSums(is.na(trips)) == 0)

stations_clean <- filter(stations, rowSums(is.na(stations)) == 0)
```

With the following code I firstly create a new variable `year` that shows the year that the trip occurred and then I compute the `hour` of the day. Then I find which year we have data from.

```
# Converting the date column to Date class
trips_clean$date <- as.Date(trips_clean$start_date)

# Extracting only the year from the date column
trips_clean$year <- as.integer(format(trips_clean$date, "%Y"))

# Extracting the hour from the date column
trips_clean$hour <- substr(trips_clean$start_date, 12, 13)
```

```
unique(trips_clean$year)
```

```
## [1] 2011 2012
```

We have data from the years 2011 and 2012.

Creating a new dataframe that includes only data from 2012 with the name `trips_2012` .

```
trips_2012 <- filter(trips_clean, year == 2012)
```

Creating a new variable `age` in the `trips_2012` dataframe that gives the age of the rider (at the time of the trip).

```
trips_2012$age <- trips_2012$year - trips_2012$birth_date
```

Removing the `birth_date` column from `trips_2012` .

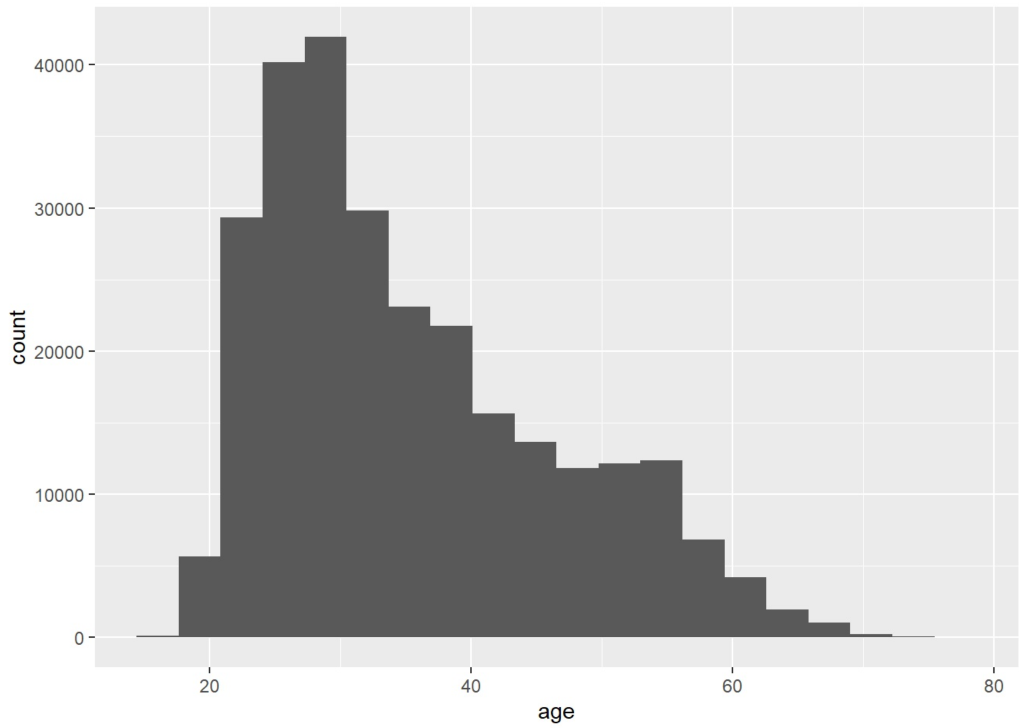
```
trips_2012 <- select(trips_2012, -birth_date)
```

Let's perform relevant EDA to answer the following question:

- Who? Who's using the bikes? More men or more women? Older or younger people?
- When is the biggest rush hour?

I will create relevant plots and compute summary statistics to answer the questions above using the `ggplot2` library.

```
ggplot(trips_2012, aes(x = age)) + geom_histogram(bins = 20)
```

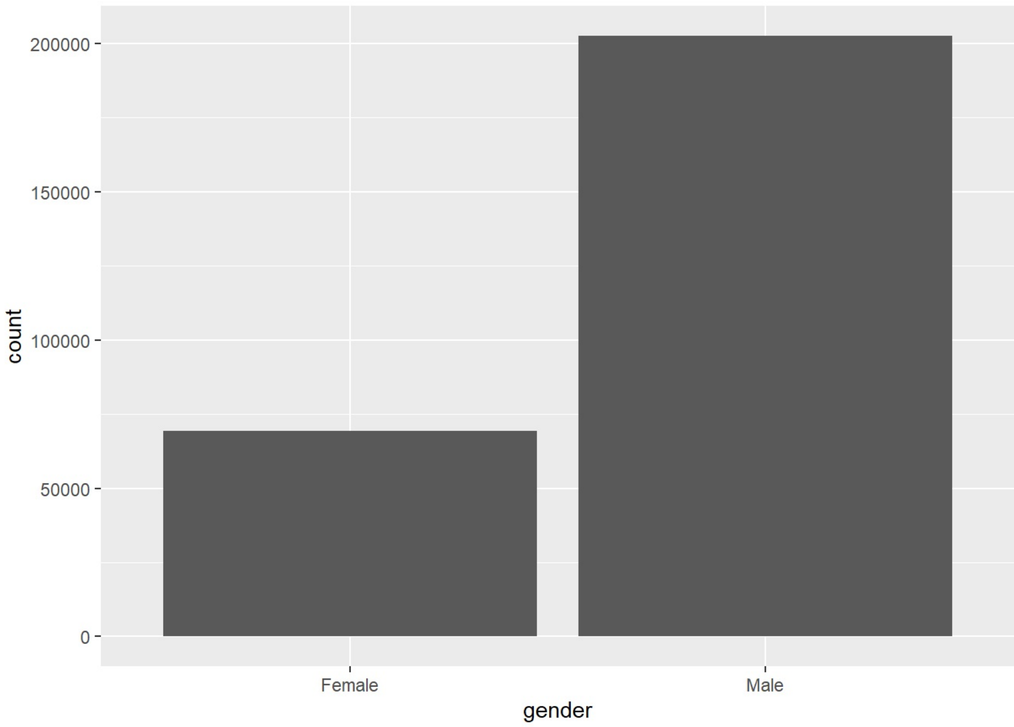



```
summary(trips_2012)
```

```
##      seq_id         hubway_id         status              duration      
##  Min.   :140522   Min.   :157655   Length:271916      Min.   :      0  
##  1st Qu.:236916   1st Qu.:271368   Class :character   1st Qu.:    347  
##  Median :339409   Median :384946   Mode  :character   Median :    533  
##  Mean   :342213   Mean   :388096                      Mean   :    751  
##  3rd Qu.:444895   3rd Qu.:503424                      3rd Qu.:    829  
##  Max.   :549286   Max.   :620312                      Max.   :5351083  
##   start_date          strt_statn      end_date           end_statn    
##  Length:271916      Min.   : 3.00   Length:271916      Min.   : 3.00  
##  Class :character   1st Qu.:22.00   Class :character   1st Qu.:22.00  
##  Mode  :character   Median :38.00   Mode  :character   Median :38.00  
##                     Mean   :37.81                      Mean   :37.74  
##                     3rd Qu.:52.00                      3rd Qu.:52.00  
##                     Max.   :98.00                      Max.   :98.00  
##    bike_nr           subsc_type          zip_code            gender         
##  Length:271916      Length:271916      Length:271916      Length:271916     
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##       date                 year          hour                age       
##  Min.   :2012-03-13   Min.   :2012   Length:271916      Min.   :17.00  
##  1st Qu.:2012-05-24   1st Qu.:2012   Class :character   1st Qu.:27.00  
##  Median :2012-07-13   Median :2012   Mode  :character   Median :32.00  
##  Mean   :2012-07-07   Mean   :2012                      Mean   :35.43  
##  3rd Qu.:2012-08-25   3rd Qu.:2012                      3rd Qu.:42.00  
##  Max.   :2012-09-30   Max.   :2012                      Max.   :78.00  
```

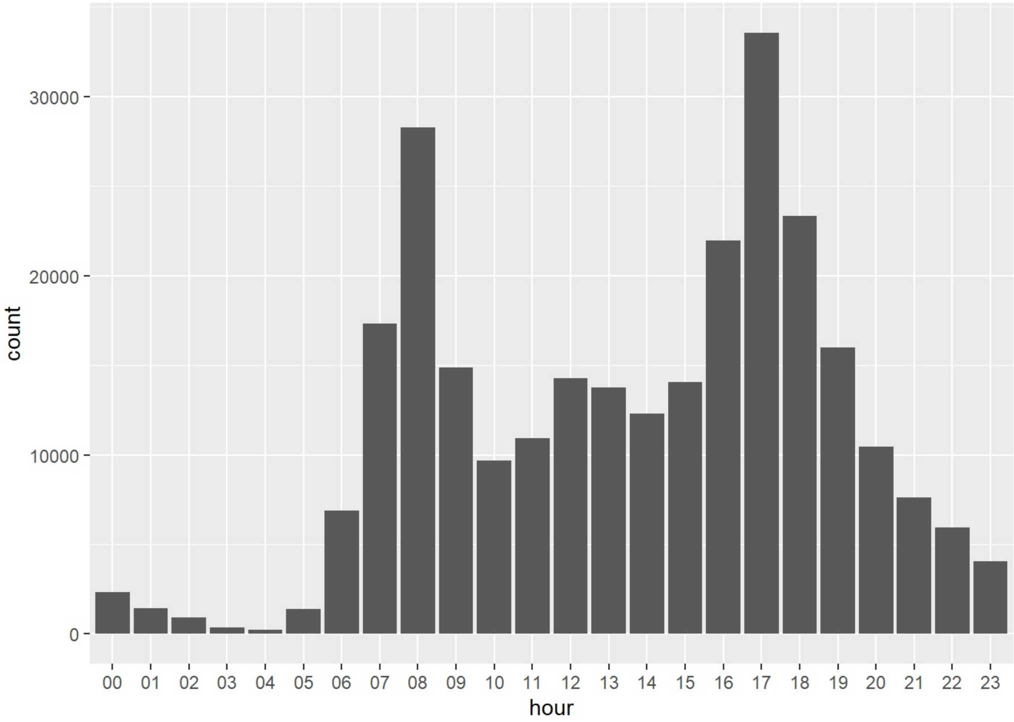
Based on the histogram above we conclude that the age group that uses bikes the most is around 20 - 35 years old, with a peak at the late 20s - 30. After the ages of 35-40, bike usage starts droping, so we can say that younger people use the bikes more than older people, with an exception to kids and teenagers, who don’t use the bikes. We can also reinforce these facts seeing the statistical summary. The min age is 17, the max 78, and the median is 32, which means half the people who use bikes are in the ages 17-32, which is a span of 15 years, compared to the other half (32-78) which is a span of 46 years.

```
ggplot(trips_2012, aes(x = gender)) + geom_bar()
```



Based on the barplot above that shows the number of bike rides by gender in 2012, we conclude that men used the bikes significantly more (the rides by women are approximately 1/3 of the ones by men).

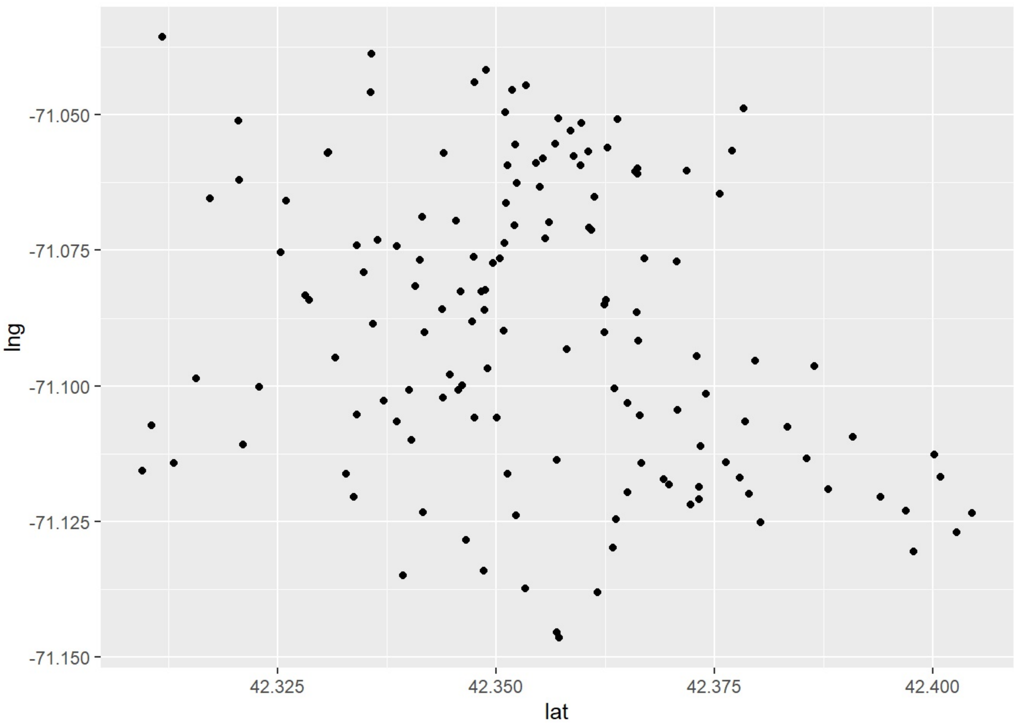
```
ggplot(trips_2012, aes(x = hour)) + geom_histogram(stat = "count")
```



The biggest rush hour is at 17:00 in the afternoon and the second biggest at 08:00 in the morning.

Creating plots to find out if there is any relation between the stations data.

```
ggplot(stations_clean, aes(x = lat, y = lng)) + geom_point()
```



At first look the relationship between the latitude and the longitude seems to be random.

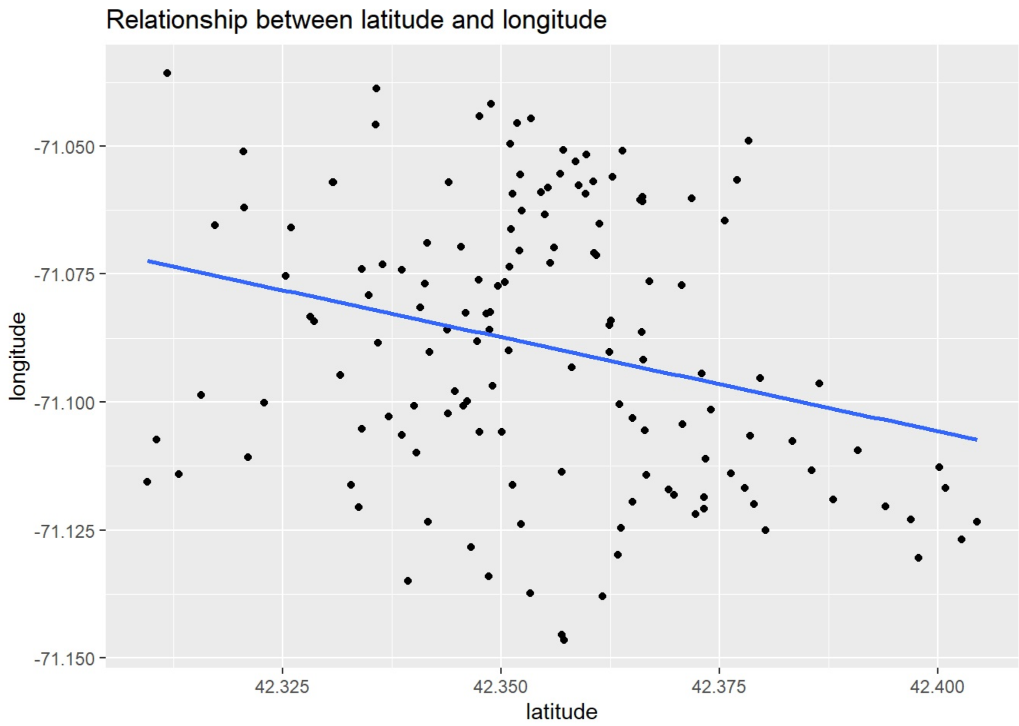
```
summarize(stations_clean, correlation = cor(lat, lng))
```

```
##   correlation
## 1  -0.2723766
```

But the correlation is -0.27 so there is a kind of linear correlation, even if not strong. Now we use linear regression to get more precise information:

```
ggplot(stations_clean, aes(x = lat, y = lng)) +
  geom_point() +
  labs(x = "latitude", y = "longitude",
       title = "Relationship between latitude and longitude") +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
model_lat_lng <- lm(lng ~ lat, data = stations_clean)

summary(model_lat_lng)
```

```
##
## Call:
## lm(formula = lng ~ lat, data = stations_clean)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.056473 -0.020004  0.000016  0.021176  0.048805
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -55.5203     4.6482 -11.945  < 2e-16 ***
## lat          -0.3676     0.1097  -3.349  0.00104 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02618 on 140 degrees of freedom
## Multiple R-squared:  0.07419,    Adjusted R-squared:  0.06758
## F-statistic: 11.22 on 1 and 140 DF,  p-value: 0.001041
```

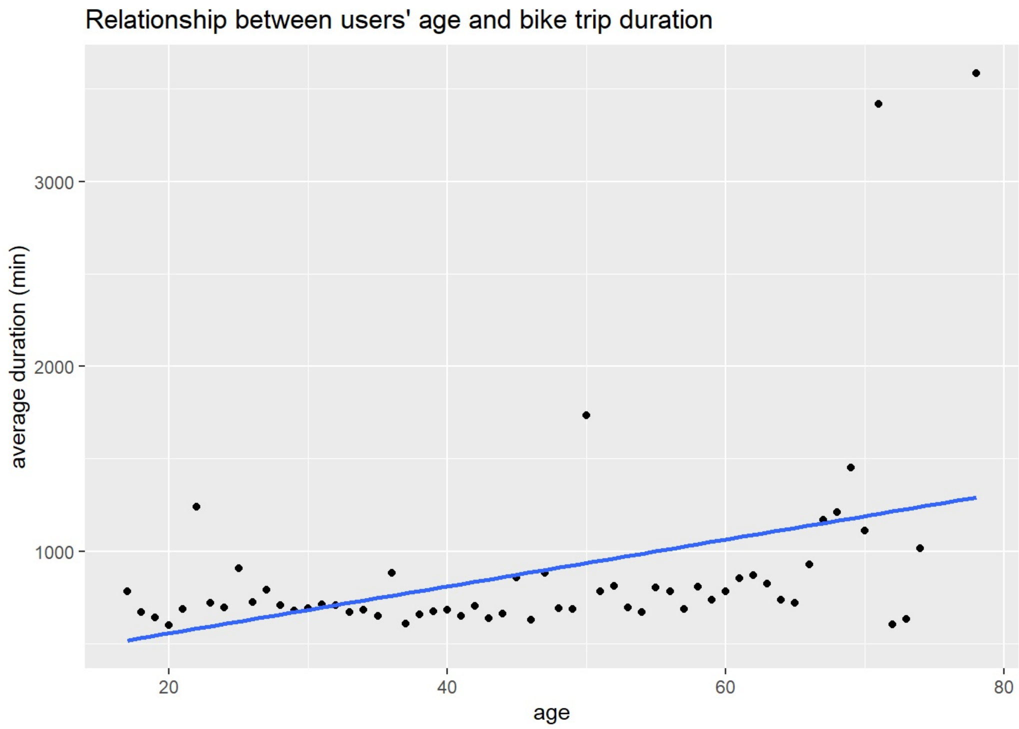
We have a negative slope, which coincides with the negative correlation, and the R-squared value is small, only 7.4% which means the correlation is not strong.

Based on all the above, the relationship between the latitude and the longitude seems to be inversely proportionate (when the latitude increases, the longitude decreases), but the data suggests that the correlation is not really strong, so it is probably safer to assume that the relationship between those two is mostly random.

How does user demographics impact the duration the bikes are being used? I will create two simple linear models and interpret the coefficients to answer the question above.

```
trips_2012 %>%
  group_by(age) %>%
  summarize(avg_duration = mean(duration)) %>%
  ggplot(aes(x = age, y = avg_duration)) + geom_point() + labs(x = "age", y = "average duration (min)", title = "
Relationship between users' age and bike trip duration") +
  geom_smooth(method = "lm", se = FALSE)
```

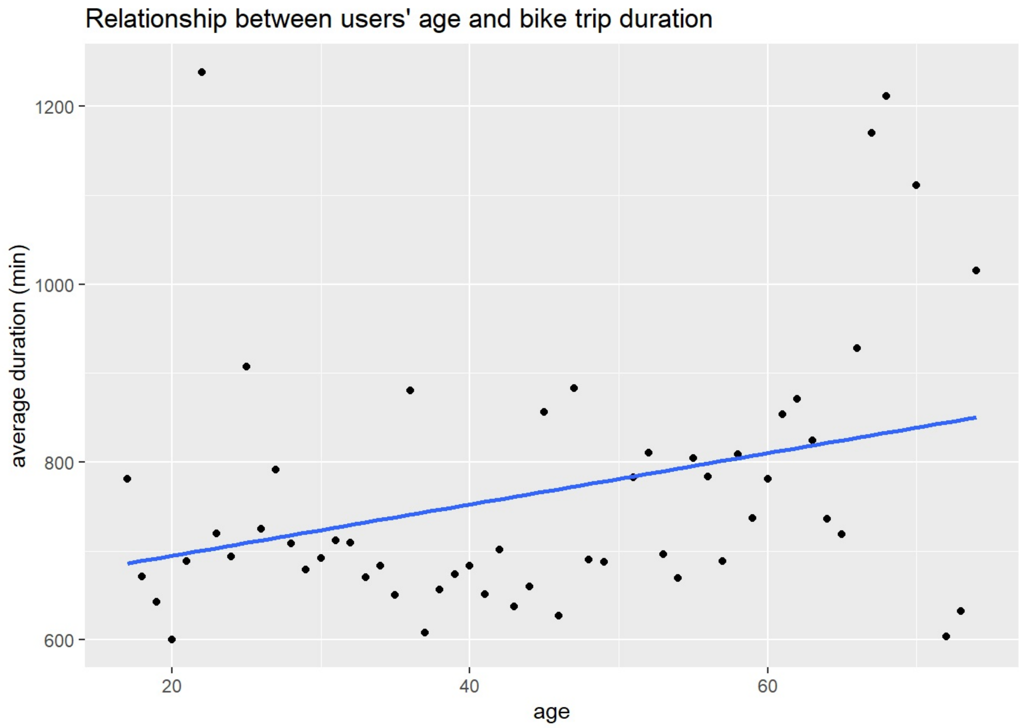
```
## `geom_smooth()` using formula = 'y ~ x'
```



We are going to do the graph again without the outliers, by filtering the average duration.

```
trips_2012 %>%
  group_by(age) %>%
  summarize(avg_duration = mean(duration)) %>%
  filter(avg_duration < 1250) %>%
  ggplot(aes(x = age, y = avg_duration)) + geom_point() + labs(x = "age", y = "average duration (min)", title = "
Relationship between users' age and bike trip duration") +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Now the linear regression model:

```
trips_2012_by_age <- group_by(trips_2012, age)
data <- summarize(trips_2012_by_age, avg_duration = mean(duration))
filter(data, avg_duration < 1250)
```

```
## # A tibble: 55 × 2
##      age avg_duration
##    <int>        <dbl>
##  1    17         781.
##  2    18         671.
##  3    19         643.
##  4    20         601.
##  5    21         688.
##  6    22        1239.
##  7    23         719.
##  8    24         694.
##  9    25         908.
## 10    26         725.
## # ℹ 45 more rows
```

```
model_dur <- lm(avg_duration ~ age, data = data)
summary(model_dur)
```

```
##
## Call:
## lm(formula = avg_duration ~ age, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -611.59 -223.61 -123.35   68.89 2294.63
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  302.958    184.872   1.639  0.10678
## age           12.668      3.763   3.366  0.00137 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 494.8 on 57 degrees of freedom
## Multiple R-squared:  0.1658, Adjusted R-squared:  0.1512
## F-statistic: 11.33 on 1 and 57 DF,  p-value: 0.001369
```

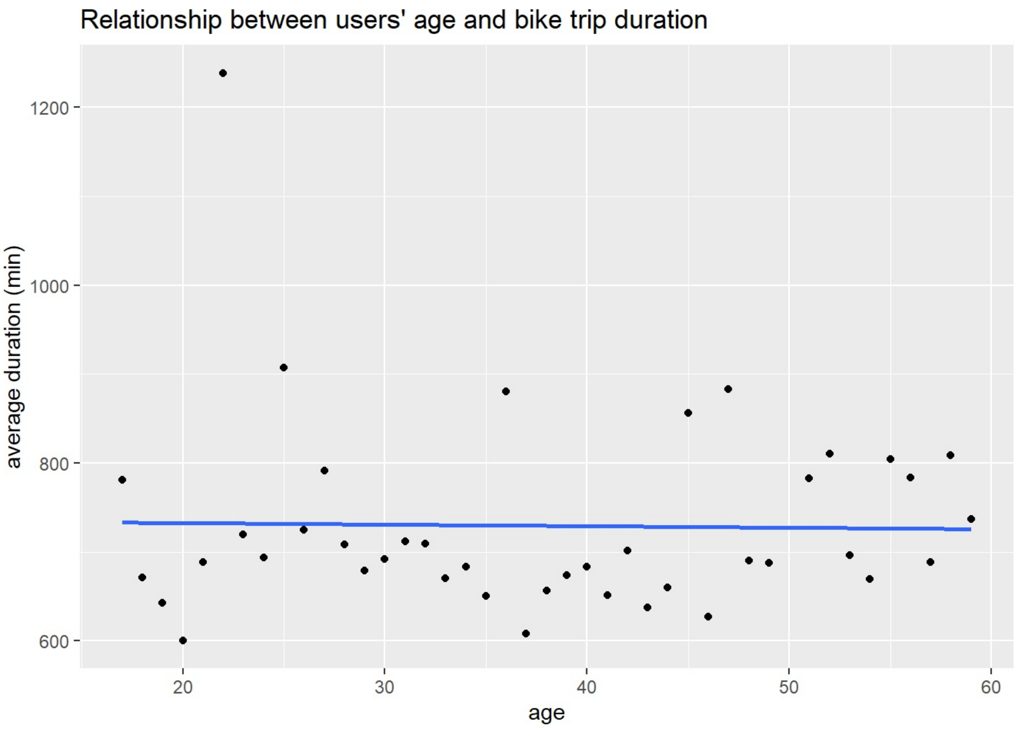
We can see that as the age increases, the average duration of the bike trip is longer. According to the linear regression model we have an intercept of value 302.958, and a slope of value 12.668. So our prediction line is: $y = 12.668*x + 302.958$

This means that if the age of the user increases by 1 year, the average trip duration increases by 12.668 minutes.

But if we look at the first graph, it seems like before the age of 60, where according to the histogram that we made earlier, the majority of our users are, the duration seems to be in a straight line. So for the integrity of our results, we are fitting the line again till the age of 60, and then again only for the ages above 60.

```
trips_2012 %>%
  group_by(age) %>%
  summarize(avg_duration = mean(duration)) %>%
  filter(avg_duration < 1250, age < 60) %>%
  ggplot(aes(x = age, y = avg_duration)) + geom_point() + labs(x = "age", y = "average duration (min)", title = "
Relationship between users' age and bike trip duration") +
  geom_smooth(method = "lm", se = FALSE)
```

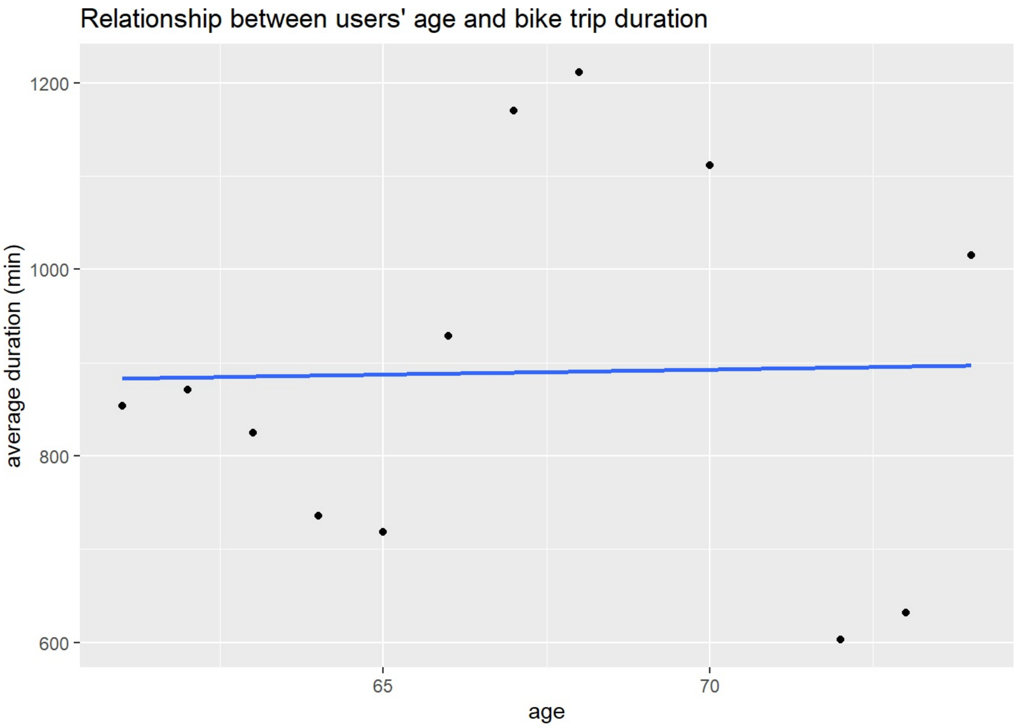
```
## `geom_smooth()` using formula = 'y ~ x'
```



We can see now that the line is almost straight, which means that all the users till the age of 60 use the bikes for approximately the same duration.

```
trips_2012 %>%
  group_by(age) %>%
  summarize(avg_duration = mean(duration)) %>%
  filter(avg_duration < 1250, age > 60) %>%
  ggplot(aes(x = age, y = avg_duration)) + geom_point() + labs(x = "age", y = "average duration (min)", title = "
Relationship between users' age and bike trip duration") +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

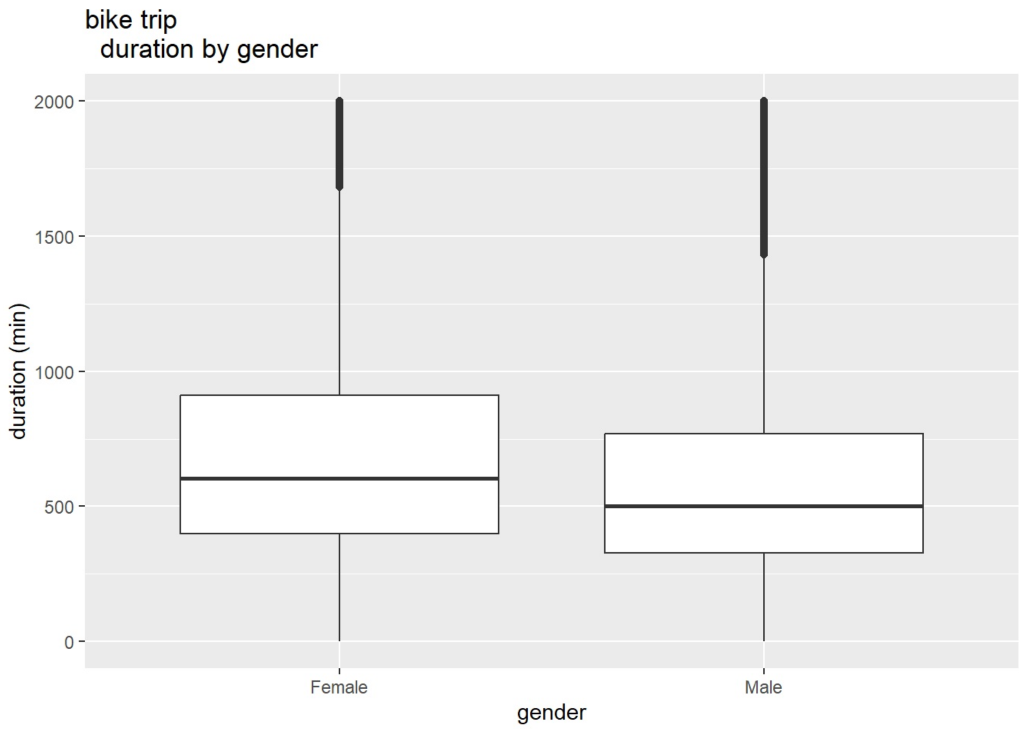


For the ages above 60, we can see now that the line is almost straight again but with a slightly increasing slope.

Based on these results, we can say that people over 60, even though less, use the bikes for a longer duration than people under 60. People under 60, use the bikes for approximately the same average duration.

Now for the gender:

```
ggplot(trips_2012, aes(x = gender, y = duration)) + geom_boxplot() +
  labs(x = "gender", y = "duration (min)", title = "bike trip
  duration by gender") + ylim(c(0, 2000))
```




```
model_gender <- lm(duration ~ gender, data = trips_2012)
summary(model_gender)
```

```
##
## Call:
## lm(formula = duration ~ gender, data = trips_2012)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##    -855    -399    -215      75 5350368
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   854.73      46.66  18.318  < 2e-16 ***
## genderMale   -139.81      54.06  -2.586  0.00971 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12290 on 271914 degrees of freedom
## Multiple R-squared:  2.46e-05,   Adjusted R-squared:  2.092e-05
## F-statistic: 6.688 on 1 and 271914 DF,  p-value: 0.009705
```

The intercept corresponds to the mean bike trip duration of women, and is 854.73 minutes.

The value -139.81 is the difference in the mean trip duration of men relative to women. So the mean bike trip duration of men is 854.73 - 139.81 = 714.92 minutes.

We conclude that in average women use the bikes for longer duration trips than men.

There are some questions that cannot be answered with simple graphing techniques. It requires combining different variables. Let us try to answer the question: How does the distance from the center of the city affect the bike usage?

The following code, firstly counts the number of checkout from each station. Then it combines the data from the trips and the stations to calculate the distance of each checkout station from the city center using the `haversine()` function. It returns a dataframe `counts` that contains columns for station ID, number of checkouts, latitude, longitude, and distance to the city center.

```r
haversine <- function(pt, lat2=42.355589, lon2=-71.060175) {
  # Calculating the great circle distance between two points on the earth

  # Extracting latitude and longitude of point pt
  lon1 <- pt[1]
  lat1 <- pt[2]

  # Converting decimal degrees to radians
  lon1 <- lon1 * pi / 180
  lat1 <- lat1 * pi / 180
  lon2 <- lon2 * pi / 180
  lat2 <- lat2 * pi / 180

  # Haversine formula
  dlon <- lon2 - lon1
  dlat <- lat2 - lat1
  a <- sin(dlat/2)^2 + cos(lat1) * cos(lat2) * sin(dlon/2)^2
  c <- 2 * asin(sqrt(a))
  r <- 3956 # Radius of earth in miles

  return(c * r)
}

get_distance <- function(trip_data, station_data){
    station_counts <- table(subset(trip_data, !is.na(strt_statn))$strt_statn)

    # Converting the result to a dataframe
  counts_df <- data.frame(
      id = as.numeric(names(station_counts)),
      checkouts = as.numeric(station_counts)
    )

    # Joining with station data
  counts_df <- merge(counts_df, station_data, by = "id")

  dist_to_center <- numeric()
  for (i in 1:nrow(counts_df)){
    dist_to_center <- rbind(dist_to_center, haversine(c(counts_df$lng[i], counts_df$lat[i])))}

  counts_df$dist_to_center <- dist_to_center

  return(counts_df)}

counts <- get_distance(trips_2012, stations_clean)
head(counts)
```

```
##   id checkouts terminal                                       station municipal
## 1  3      2298   B32006                       Colleges of the Fenway    Boston
## 2  4      4504   C32000                   Tremont St. at Berkeley St.    Boston
## 3  5      2133   B32012           Northeastern U / North Parking Lot    Boston
## 4  6      4524   D32000                      Cambridge St. at Joy St.    Boston
## 5  7      2019   A32000                                      Fan Pier    Boston
## 6  8      1621   A32001 Union Square - Brighton Ave. at Cambridge St.    Boston
##        lat       lng   status dist_to_center
## 1 42.34002 -71.10081 Existing      2.3357065
## 2 42.34539 -71.06962 Existing      0.8530953
## 3 42.34181 -71.09018 Existing      1.8024226
## 4 42.36129 -71.06514 Existing      0.4678034
## 5 42.35341 -71.04462 Existing      0.8075823
## 6 42.35333 -71.13731 Existing      3.9389523
```

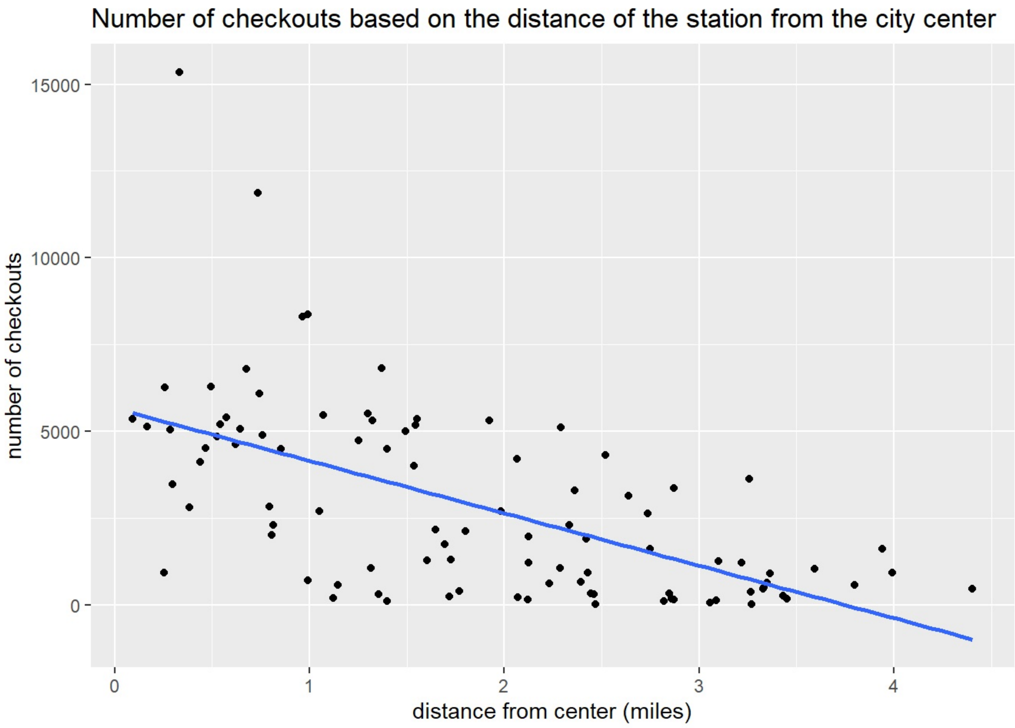
I will create a simple linear model to predict the number of checkouts based on the distance of the bikes from the centre of the city using the `counts` dataframe. Then, I will visualize the prediction against the data.

```r
model_checkouts <- lm(checkouts ~ dist_to_center, data = counts)
summary(model_checkouts)
```

```
##
## Call:
## lm(formula = checkouts ~ dist_to_center, data = counts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4355.2 -1365.3  -116.1  1252.6 10189.1
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5666.9      451.8  12.542  < 2e-16 ***
## dist_to_center  -1510.6      210.4  -7.179 1.69e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2212 on 93 degrees of freedom
## Multiple R-squared:  0.3566, Adjusted R-squared:  0.3496
## F-statistic: 51.54 on 1 and 93 DF,  p-value: 1.69e-10
```

```
ggplot(counts, aes(x = dist_to_center, y = checkouts)) + geom_point() + labs(x = "distance from center (miles)",
y = "number of checkouts", title = "Number of checkouts based on the distance of the station from the city center
") +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Based on the linear model, the number of checkouts decreases as we move further from the center. Our prediction line is:

y = -1510.6*x + 5666.9, where y is the number of checkouts and x is the distance from the center in miles.

Based on our linear model, what would most likely be the number of checkouts for a distance of 2.5 miles from the city center?

Our prediction model is the line: y = -1510.6 * x + 5666.9, where y are the checkouts and x the distance, so for x = 2.5 miles, y is:

```
-1510.6 * 2.5 + 5666.9
```

```
## [1] 1890.4
```

The right answer is 3, 1890 checkouts.