Interesting observations for Biketown PDX

Raymond Ford (raymond.anthony.ford@gmail.com)

2020-06-05

Contents

This notebook looks at some of the interesting observations noted when we first began to look at the raw data in the data preprocessing notebook, and we will extend these observations to the entire data set. In particular, we seek to identify observations (trips) where the rider traveled at an excessive speed, or perhaps may have been a time traveler (trips beginning or ending in the past/future). All of these observations can be attributed to some type of data entry error, but maybe, just maybe, there are time travelers riding the orange Biketown PDX bikes around town, or Marvels's Quicksilver/DC's Flash rides among us.

Preliminaries

Before we begin to look for these interesting observations/trips, we will need to import some `R` packages and bring in all of the raw data into `R`.

Required packages

We will use the following `R` packages for this endeavor. These packages allow us to create location plots/maps.

```
require(sp)
```

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

```
require(leaflet)
```

```
## Loading required package: leaflet
```

```
require(webshot)
```

```
## Loading required package: webshot
```

Bringing in the raw data

We first begin by bringing our raw data into R. Since all of the files we need to bring in are stored in /data/raw/ and no other files are located in that directory we will do the following:

1. Set our working directory to the folder containing the data for each of the 45 months.
2. Obtain the names of each of the files in /data/raw/ ending in .csv.
3. Use these names to create a list() object from the files found in #1.
4. Make this list() object into a R dataframe object.
5. Create a data frame dat.raw which we will use for the remainder of this notebook.

```
setwd("~/GoogleDrive/pdxbikes/pdxbikes/data/raw/")
file.names <- list.files(pattern="*.csv")
make.df <- lapply(file.names, read.csv)
rm(file.names) # We do not need these anymore, so it's best to remove them.
dat.raw <- do.call(rbind.data.frame, make.df)
rm(make.df)
```

Since /data/raw/ is hard coded, we obtain the above warning from R. We can move past this warning and continue on in our quest to identify interesting observations the raw data.

Time travelers

We first begin by converting StartDate and EndDate into "Date" objects, and then output a summary of the data.

```
dat.raw$StartDate <- as.Date(dat.raw$StartDate, format="%m/%d/%Y")
dat.raw$EndDate <- as.Date(dat.raw$EndDate, format="%m/%d/%Y")
summary(dat.raw)
```

```
##     RouteID             PaymentPlan
##  Min.   : 1282087   Casual    :711282
##  1st Qu.: 3591394   Subscriber:521022
##  Median : 7220719             :  4088
##  Mean   : 7226320
##  3rd Qu.:11064799
##  Max.   :13201070
##  NA's   :4088
##                                StartHub      StartLatitude   StartLongitude
##                                    :378982   Min.   :45.30   Min.   :-123.14
##  SW Salmon at Waterfront Park      : 27814   1st Qu.:45.52   1st Qu.:-122.68
##  SW Moody at Aerial Tram Terminal: 20167   Median :45.52   Median :-122.67
##  SW River at Montgomery            : 17924   Mean   :45.52   Mean   :-122.67
##  NW Everett at 22nd                : 17776   3rd Qu.:45.53   3rd Qu.:-122.66
##  NW 13th at Marshall               : 17093   Max.   :45.73   Max.   :  67.18
##  (Other)                           :756636   NA's   :4680    NA's   :4680
##    StartDate            StartTime
##  Min.   :2016-07-19           :   4088
##  1st Qu.:2017-07-13   17:06   :   2223
##  Median :2018-05-29   17:09   :   2205
##  Mean   :2018-05-02   17:08   :   2189
##  3rd Qu.:2019-03-30   17:07   :   2182
##  Max.   :2020-03-31   17:11   :   2175
##  NA's   :4088         (Other):1221330
```

```
##                                EndHub        EndLatitude     EndLongitude
##                                   :319512   Min.   :34.26   Min.   :-134.4
##  SW Salmon at Waterfront Park     : 31329   1st Qu.:45.52   1st Qu.:-122.7
##  NW 13th at Marshall              : 22421   Median :45.52   Median :-122.7
##  SW Moody at Aerial Tram Terminal: 22373   Mean   :45.52   Mean   :-122.7
##  NW Couch at 11th                 : 21178   3rd Qu.:45.53   3rd Qu.:-122.7
##  SW River at Montgomery           : 20746   Max.   :49.16   Max.   :  45.5
##  (Other)                          :798833   NA's   :4730    NA's   :4730
##     EndDate              EndTime              TripType           BikeID
##  Min.   :1969-12-31          :   4394              :1235063   Min.   : 5986
##  1st Qu.:2017-07-13   17:27  :   2255   commute    :    406   1st Qu.: 6302
##  Median :2018-05-29   17:29  :   2191   errand     :    172   Median : 6574
##  Mean   :2018-05-02   17:28  :   2155   recreation:    663   Mean   : 6960
##  3rd Qu.:2019-03-30   17:25  :   2142   work       :     88   3rd Qu.: 7155
##  Max.   :2080-01-05   17:18  :   2135                         Max.   :35537
##  NA's   :4394         (Other):1221120                         NA's   :4088
##           BikeName       Distance_Miles         Duration
##               :   4873   Min.   :    0.000          :   8662
##  0090 BIKETOWN:   1562   1st Qu.:    0.720   0:06:37:   1225
##  0153 BIKETOWN:   1547   Median :    1.300   0:06:39:   1216
##  0139 BIKETOWN:   1521   Mean   :    2.002   0:05:41:   1212
##  0646 BIKETOWN:   1518   3rd Qu.:    2.390   0:05:36:   1211
##  0584 BIKETOWN:   1508   Max.   :15527.180   0:06:07:   1210
##  (Other)      :1223863   NA's   :4088        (Other):1221656
##             RentalAccessPath  MultipleRental
##  keypad              :915062   Mode :logical
##  mobile              :185136   FALSE:1120063
##  keypad_rfid_card    :126201   TRUE :112241
##                      :  4088   NA's :4088
##  keypad_phone_number:  2829
##  web                 :  2205
##  (Other)             :   871
```

In the above output we notice the following:

1. All of the dates for `StartDate` are within the expected range of our data.
2. There are dates in `EndDate` that are outside the expected range of our data. In particular we see trips that ended in 1969 and 2080.

To take a closer look at the trips outside of our expected date range for our data we will obtain their observation numbers, create a subset of `dat.raw` and name it `time.travel`, and output a summary of this subset of the data with the following code.

```
known.dates <- as.factor(seq(as.Date("2016-07-19"), as.Date("2020-03-31"), "days"))
in.known.range <- which(as.factor(dat.raw$EndDate) %in% known.dates)
time.travel <- dat.raw[-in.known.range, ]
rm(in.known.range, known.dates)
summary(time.travel)
```

```
##     RouteID             PaymentPlan
##  Min.   : 1282195   Casual    : 150
##  1st Qu.: 1284672   Subscriber: 214
##  Median : 1586165             :4088
##  Mean   : 2215485
##  3rd Qu.: 2151635
##  Max.   :13142137
```

```
##  NA's   :4088
##                                          StartHub    StartLatitude
##                                              :4201   Min.   :45.49
##  SW Salmon at Waterfront Park                :  11   1st Qu.:45.51
##  SW 3rd at Ankeny                            :   9   Median :45.52
##  SW Broadway at Smith Memorial Student Union:   8   Mean   :45.52
##  SW 5th at Morrison                          :   7   3rd Qu.:45.53
##  SW Naito at Ankeny Plaza                    :   7   Max.   :45.56
##  (Other)                                     : 209   NA's   :4103
##  StartLongitude     StartDate            StartTime
##  Min.   :-122.7   Min.   :2016-07-19          :4088
##  1st Qu.:-122.7   1st Qu.:2016-07-19   11:24  :   5
##  Median :-122.7   Median :2016-09-01   11:23  :   4
##  Mean   :-122.7   Mean   :2016-11-19   11:25  :   4
##  3rd Qu.:-122.7   3rd Qu.:2016-12-03   11:27  :   4
##  Max.   :-122.6   Max.   :2020-03-04   13:07  :   4
##  NA's   :4103     NA's   :4088         (Other): 343
##                            EndHub      EndLatitude     EndLongitude
##                               :4295   Min.   :45.50   Min.   :-122.7
##  N Mississippi at Beech       :   6   1st Qu.:45.52   1st Qu.:-122.7
##  SE 9th at Belmont            :   6   Median :45.52   Median :-122.7
##  SW River at Montgomery       :   6   Mean   :45.52   Mean   :-122.7
##  SW Salmon at Waterfront Park:   6   3rd Qu.:45.53   3rd Qu.:-122.7
##  NW Johnson at Jamison Square:   5   Max.   :45.56   Max.   :-122.6
##  (Other)                      : 128   NA's   :4102    NA's   :4102
##     EndDate              EndTime           TripType        BikeID
##  Min.   :1969-12-31          :4394             :4446   Min.   :5991
##  1st Qu.:1969-12-31   16:30  :  30   commute   :   3   1st Qu.:6327
##  Median :1969-12-31   16:00  :  19   errand    :   1   Median :6604
##  Mean   :2010-07-31   17:30  :   2   recreation:   1   Mean   :6694
##  3rd Qu.:2080-01-05   11:28  :   1   work      :   1   3rd Qu.:7147
##  Max.   :2080-01-05   16:24  :   1                     Max.   :7501
##  NA's   :4394         (Other):   5                     NA's   :4088
##           BikeName    Distance_Miles           Duration
##               :4088   Min.   : 0.000               :4430
##  0794 BIKETOWN:   4   1st Qu.: 0.098   550617:17:53:   1
##  0172 BIKETOWN:   3   Median : 0.825   550700:52:38:   1
##  0257 BIKETOWN:   3   Mean   : 1.306   549572:39:56:   1
##  0654 BIKETOWN:   3   3rd Qu.: 1.812   549763:59:42:   1
##  0687 BIKETOWN:   3   Max.   :12.970   550041:15:29:   1
##  (Other)      : 348   NA's   :4088     (Other)     :  17
##          RentalAccessPath MultipleRental
##                  :4088    Mode :logical
##  keypad          : 320    FALSE:328
##  keypad_rfid_card:  20    TRUE :36
##  mobile          :  19    NA's :4088
##  unknown         :   4
##  web             :   1
##  (Other)         :   0
```

In the above output for `EndDate` we notice that in addition to trips ending in 1969 and 2080, there are 4394 trips which have no recorded end date. We will remove these and output a summary of the updated data set.

```
time.travel <- time.travel[!is.na(time.travel$EndDate), ]
dim(time.travel)
```

```
## [1] 58 19
```

```
summary(time.travel)
```

```
##     RouteID             PaymentPlan                        StartHub
##  Min.   : 2463277   Casual    :12                             :38
##  1st Qu.: 3386689   Subscriber:46   NW Raleigh at 21st       : 2
##  Median : 4602332              : 0   N Killingsworth at Albina: 1
##  Mean   : 5498699                    NW 18th at Northrup      : 1
##  3rd Qu.: 6816244                    NW Everett at 22nd       : 1
##  Max.   :13142137                    NW Naito at Ironside     : 1
##                                      (Other)                  :14
##  StartLatitude   StartLongitude     StartDate            StartTime
##  Min.   :45.51   Min.   :-122.7   Min.   :2017-03-10   14:00  : 2
##  1st Qu.:45.52   1st Qu.:-122.7   1st Qu.:2017-06-27   16:38  : 2
##  Median :45.52   Median :-122.7   Median :2017-09-24   0:22   : 1
##  Mean   :45.53   Mean   :-122.7   Mean   :2017-12-22   1:03   : 1
##  3rd Qu.:45.53   3rd Qu.:-122.7   3rd Qu.:2018-05-10   1:37   : 1
##  Max.   :45.56   Max.   :-122.6   Max.   :2020-03-04   10:37  : 1
##  NA's   :14      NA's   :14                            (Other):50
##                            EndHub    EndLatitude     EndLongitude
##                               :37   Min.   :45.50   Min.   :-122.7
##  SW 5th at Morrison           : 2   1st Qu.:45.52   1st Qu.:-122.7
##  SW Salmon at Waterfront Park: 2   Median :45.52   Median :-122.7
##  N Failing at Williams        : 1   Mean   :45.52   Mean   :-122.7
##  N Mississippi at Beech       : 1   3rd Qu.:45.53   3rd Qu.:-122.7
##  NE MLK at Knott              : 1   Max.   :45.55   Max.   :-122.6
##  (Other)                      :14   NA's   :10      NA's   :10
##     EndDate              EndTime         TripType      BikeID
##  Min.   :1969-12-31   16:30  :30             :56   Min.   :6034
##  1st Qu.:1969-12-31   16:00  :19   commute   : 1   1st Qu.:6282
##  Median :1969-12-31   17:30  : 2   errand    : 1   Median :6576
##  Mean   :2010-07-31   11:28  : 1   recreation: 0   Mean   :6704
##  3rd Qu.:2080-01-05   16:24  : 1   work      : 0   3rd Qu.:7233
##  Max.   :2080-01-05   17:47  : 1                   Max.   :7499
##                       (Other): 4
##              BikeName  Distance_Miles           Duration
##  0952 AIR MAX    : 2   Min.   : 0.000               :36
##  0985 AIR TRAINER: 2   1st Qu.: 0.000   550617:17:53: 1
##  0039 BIKETOWN   : 1   Median : 0.520   550700:52:38: 1
##  0046 BIKETOWN   : 1   Mean   : 1.773   549572:39:56: 1
##  0051 BIKETOWN   : 1   3rd Qu.: 2.277   549763:59:42: 1
##  0076 BIKETOWN   : 1   Max.   :12.970   550041:15:29: 1
##  (Other)         :50                    (Other)     :17
##          RentalAccessPath MultipleRental
##  keypad          :41      Mode :logical
##  keypad_rfid_card: 9      FALSE:51
##  mobile          : 8      TRUE :7
##  admin           : 0
##  web             : 0
##                  : 0
##  (Other)         : 0
```

We note that there are 58 observations not found in our expected date range, and some of them have missing starting and ending locations. To get a better idea of the unreasonable dates we will output a table of the

ending dates.

```
table(time.travel$EndDate)
```

```
##
## 1969-12-31 1971-06-09 2014-12-31 2023-11-27 2024-05-02 2074-07-10 2080-01-05
##         31          4          1          1          1          1         19
```

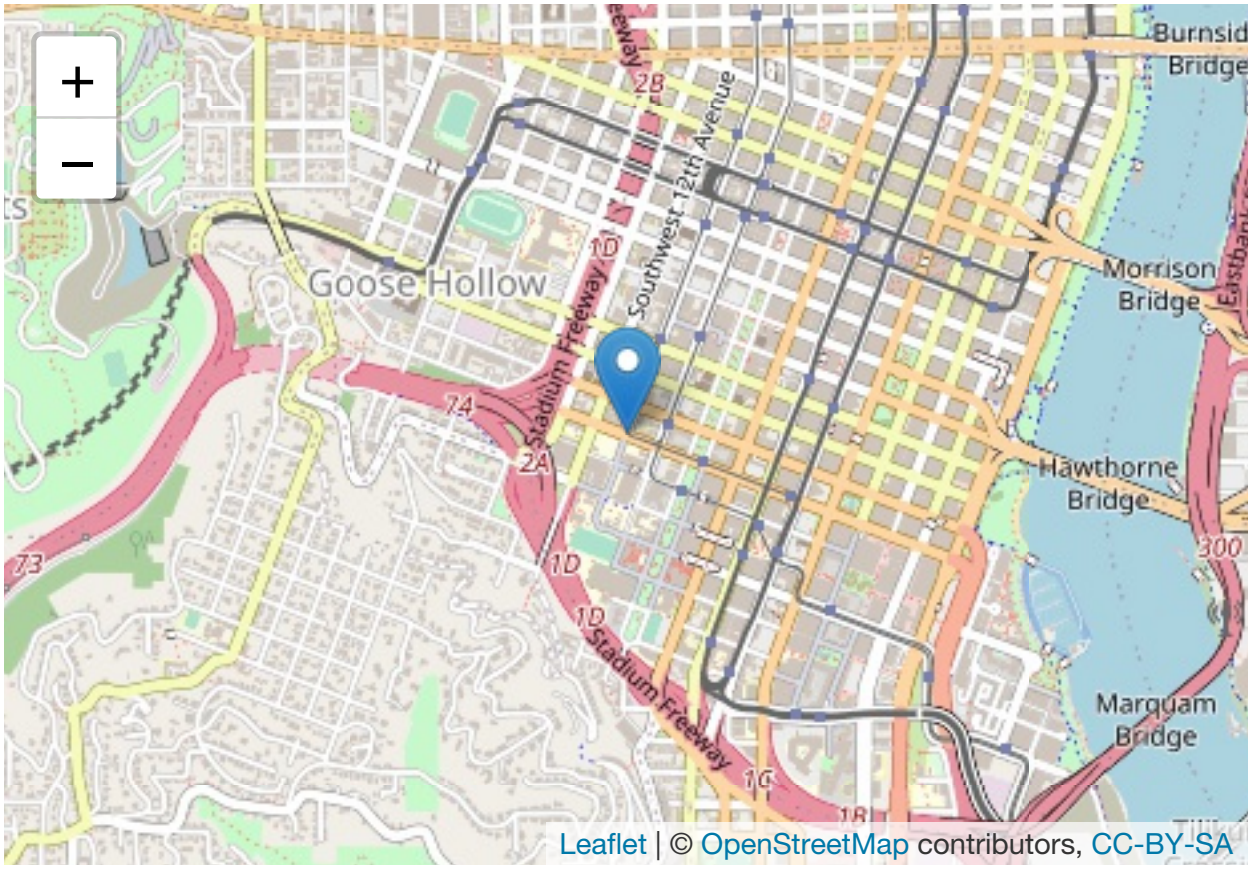
From the above output we notice that most of the ending dates are either in 1969 or 2080, with a few trips in between. The next time we would get the opportunity to meet one of these time travelers is 27 Nov. 2023. In fact, we can identify where this individual will return their bicycle rental with the following code.

```
time.travel[which(time.travel$EndDate=="2023-11-27"), ]
```

```
##         RouteID PaymentPlan StartHub StartLatitude StartLongitude  StartDate
## 896562 10783921  Subscriber               45.51136       -122.684 2019-02-12
##        StartTime EndHub EndLatitude EndLongitude    EndDate EndTime TripType
## 896562     16:59           45.51429    -122.6864 2023-11-27   16:24
##        BikeID      BikeName Distance_Miles    Duration RentalAccessPath
## 896562   6887 0450 BIKETOWN            2.3 41975:24:58           keypad
##        MultipleRental
## 896562          FALSE
```

While no `EndHub` is provided, we do have the ending longitude, latitude, and time for the bicycle's return. We can visualize this location with the code below, and perhaps meet up with the time traveler to share an IPA on a chilly November day in 2023. (Note: If you're viewing this in the PDF form of the notebook, then you will not be able to zoom in/out. This functionality is only available in the `.nb.html` version.)

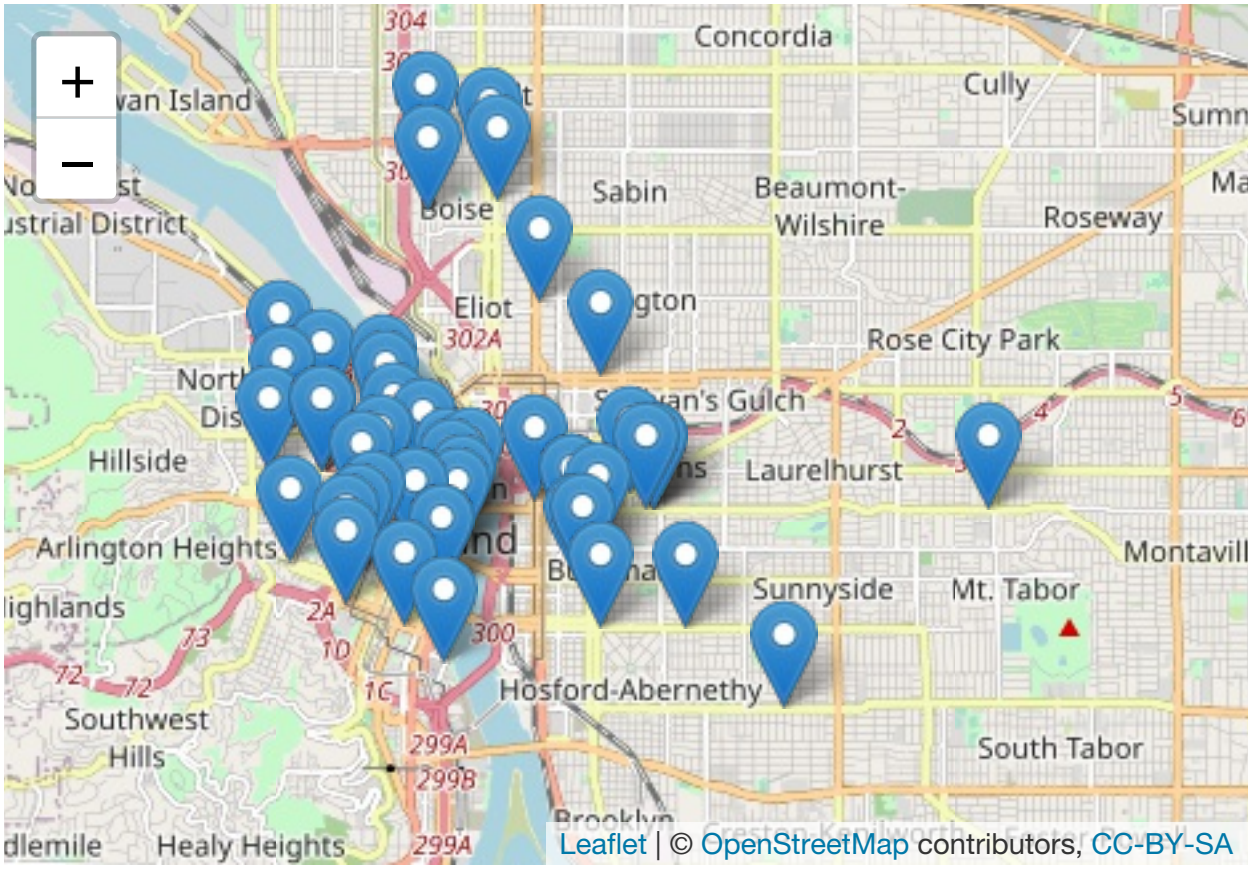
```
# This solution came from a solution posted on StackOverflow. Unfortunately, I
# cannot find the url at this time, but I will update with the appropriate link
# when I find it.
df <- data.frame(longitude = -122.6864, latitude = 45.51429)
coordinates(df) <- ~ longitude + latitude
leaflet(df) %>%
  addMarkers() %>%
  addTiles()
```



We can also visualize all the locations where all the time travelers will complete their trip with the code and plot below; we just need to remove the missing values for ten of the observations first.

```
df <- data.frame(longitude=na.omit(time.travel$EndLongitude), latitude=na.omit(time.travel$EndLatitude))

coordinates(df) <- ~ longitude + latitude
leaflet(df) %>%
  addMarkers() %>%
  addTiles()
```



Speed demons

In this section we will look at individual observation's/trip's that are not humanly possible to complete. For example, a rider's speed was in excess of what one would consider feasible on a bicycle—even under the best conditions.

Before we begin we will clean up our `R` environment.

```
rm(df, time.travel)
```

With that out of the way we will now create a new column of data known as `Mph` (miles per hour) from `dat.raw`. We do this by first converting `Duration` into a decimal value. Next with use this decimal value and `Distance_Miles` to compute the speed into miles per hour which is stored into `Mph`.

```
speed.demon <- dat.raw
rm(dat.raw) # We no longer need this.
speed.demon$Duration <- sapply(strsplit(as.character(speed.demon$Duration),":"),
       function(x) {
         x <- as.numeric(x)
         x[1]*60+x[2]+x[3]/60
       }
)
speed.demon <- speed.demon[!is.na(speed.demon$Duration), ]
speed.demon <- speed.demon[!is.na(speed.demon$Distance_Miles), ]
speed.demon$Mph <- (speed.demon$Distance_Miles / speed.demon$Duration) * 60
summary(speed.demon$Mph)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.00      4.17      5.90      6.90      7.58 310543.60
```

From the above we see that one rider traveled at around 310543 miles per hour. To understand just how insane this speed is we note that the engineering masterpiece known as Lockheed's SR-71 Blackbird travels at (a publicly disclosed speed of) 2200 mph.

We can try visualize how many other fast riders rode by plotting a stem-and-leaf plot of all the trips.

```
stem(speed.demon$Mph)
```

```
##
##   The decimal point is 4 digit(s) to the right of the |
##
##    0 | 000000000000000000000000000000000000000000000000000000000000000000000000000000+1227634
##    2 | 0266125
##    4 | 10147
##    6 | 6
##    8 | 3
##   10 |
##   12 |
##   14 | 3
##   16 |
##   18 |
##   20 |
##   22 |
##   24 |
##   26 |
##   28 |
##   30 | 1
```

We see that there will several trips exceeding the SR-71's top speed, and we will thus subset the data by only looking at the trips having a speed that exceeded 2200 mph.

```
speed.demon <- subset(speed.demon, Mph > 2200)
dim(speed.demon)
```

```
## [1] 27 20
```

With only 27 trips having a speed exceeding 2,200 mph we can output the entire subset.

```
speed.demon
```

```
##          RouteID PaymentPlan
## 28667    1374716      Casual
## 29073    1376084      Casual
## 119313   1819628  Subscriber
## 166368   2309921  Subscriber
## 256402   3124476      Casual
## 346528   3903517  Subscriber
## 391253   4298322  Subscriber
## 450310   5039933  Subscriber
## 508454   5941795      Casual
## 622385   7242673      Casual
## 651841   7629925  Subscriber
## 684532   7988584      Casual
## 808111   9714175  Subscriber
## 852728  10306714      Casual
```

```
## 870228  10503880  Subscriber
## 882057  10620001  Subscriber
## 884172  10643108  Subscriber
## 894634  10753147  Subscriber
## 900734  10841962  Subscriber
## 988408  11468802  Subscriber
## 1131150 12297891      Casual
## 1152647 12501941  Subscriber
## 1161463 12587277      Casual
## 1175708 12722459      Casual
## 1202118 12946914  Subscriber
## 1212359 13043904  Subscriber
## 1233320 13179222  Subscriber
##                                                                  StartHub
## 28667                                                    NW Couch at 11th
## 29073                                                   SE Water at Taylor
## 119313                                                   NW 8th at Everett
## 166368                                                 N Failing at Williams
## 256402
## 346528                                                   N Mason at Williams
## 391253                                                SW Naito at Ankeny Plaza
## 450310                                                        NE MLK at Knott
## 508454                                                     SE 14th at Stark
## 622385                                                  NW 11th at The Fields
## 651841                                                    NW 18th at Northrup
## 684532  SW Park at SW Wright (Oregon Holocaust Memorial) Community Corral
## 808111                                                     SE Clay at Water
## 852728                                NE 50th at Fremont - Community Corral
## 870228                                                      SW 12th at Clay
## 882057
## 884172
## 894634
## 900734                                                   NW 20th at Burnside
## 988408
## 1131150
## 1152647                                                  NW 20th at Burnside
## 1161463                                                   SE Belmont at 33rd
## 1175708                                                 SE 34th at Hawthorne
## 1202118
## 1212359
## 1233320
##         StartLatitude StartLongitude  StartDate StartTime
## 28667         45.52374     -122.68181 2016-08-02     16:45
## 29073         45.51515     -122.66578 2016-08-02     19:54
## 119313        45.52499     -122.67846 2016-10-04      7:51
## 166368        45.55076     -122.66687 2017-01-31      8:41
## 256402        45.30201     -121.74448 2017-06-07     12:59
## 346528        45.55335     -122.66691 2017-08-04     14:34
## 391253        45.52270     -122.67035 2017-09-01     12:16
## 450310        45.54180     -122.66148 2017-11-01      8:00
## 508454        45.51845     -122.65169 2018-03-11     15:32
## 622385        45.53233     -122.68243 2018-05-30     16:49
## 651841        45.53143     -122.68957 2018-06-18     14:51
## 684532        45.52182     -122.70441 2018-07-06     12:37
```

```
## 808111       45.51148     -122.66629 2018-09-26     17:08
## 852728       45.54828     -122.61116 2018-11-17     12:24
## 870228       45.51507     -122.68703 2018-12-19     14:24
## 882057       45.52297     -122.68923 2019-01-14      2:54
## 884172       45.49723     -122.62391 2019-01-18      0:33
## 894634       45.51789       67.18144 2019-02-07      7:51
## 900734       45.52331     -122.69326 2019-02-21     15:29
## 988408       45.56278       33.92405 2019-05-23     13:01
## 1131150      45.51294     -122.66167 2019-09-12     18:57
## 1152647      45.52331     -122.69326 2019-10-09     13:36
## 1161463      45.51650     -122.63075 2019-10-21     21:39
## 1175708      45.51191     -122.62948 2019-11-11     11:33
## 1202118      45.52830     -122.68233 2020-01-09      5:18
## 1212359      45.50978     -122.67802 2020-02-03     16:26
## 1233320      45.52519     -122.67753 2020-03-17     10:08
##                                          EndHub EndLatitude EndLongitude
## 28667               SW Salmon at Waterfront Park    45.51557   -122.67389
## 29073      SW Morrison at Pioneer Courthouse Sq.    45.51929   -122.67925
## 119313                        SE 7th at Burnside    45.52270   -122.65875
## 166368                           SW 5th at Main    45.51610   -122.67908
## 256402                  NE Wheeler at Multnomah    45.53087   -122.66589
## 346528                         NW Couch at 11th    45.52374   -122.68181
## 391253                           SW 5th at Main    45.51610   -122.67908
## 450310                       NE Multnomah at 9th    45.53141   -122.65746
## 508454                                          45.50867   -122.65875
## 622385                     NW 11th at The Fields    45.53233   -122.68243
## 651841                       NW 13th at Marshall    45.53080   -122.68442
## 684532               SW Yamhill at Director Park    45.51898   -122.68127
## 808111                                          45.51152   -122.65873
## 852728                                          45.51641   -122.68006
## 870228                                          45.52466     45.50117
## 882057                                          45.53153   -122.68143
## 884172  SE 29th and Gladstone - Community Corral    45.49334   -122.63576
## 894634                                          45.51750   -122.69262
## 900734                                          45.52773   -122.69449
## 988408                                          45.56269   -122.67498
## 1131150                                         45.51547   -122.67365
## 1152647                                         45.51202   -122.68361
## 1161463                                         45.51645   -122.64016
## 1175708                                         45.50480   -122.62958
## 1202118                                         45.52302   -122.67909
## 1212359                          SE 14th at Stark    45.51845   -122.65169
## 1233320                   NW Broadway at Everett    45.52508   -122.67756
##            EndDate EndTime TripType BikeID                       BikeName
## 28667   2016-08-02   17:03            5990                     0381 BIKETOWN
## 29073   2016-08-02   20:47            7283                     0722 BIKETOWN
## 119313  2016-10-04    8:01            7410                     0564 BIKETOWN
## 166368  2017-01-31    8:57            6392                    0306 BIKETOWN
## 256402  2017-06-07   13:11            6616                     0721 BIKETOWN
## 346528  2017-08-04   14:53            7184                     0428 BIKETOWN
## 391253  2017-09-01   12:24            6270                     0050 BIKETOWN
## 450310  2017-11-01    8:06            6612                     0794 BIKETOWN
## 508454  2018-03-11   15:42            6407                     0740 BIKETOWN
## 622385  2018-05-30   17:33            7344                  1027 AIR SAFARI
```

```
## 651841  2018-06-18   14:55           7178                      0672 BIKETOWN
## 684532  2018-07-06   12:54           6889                      0063 BIKETOWN
## 808111  2018-09-26   17:14           7497                      0576 BIKETOWN
## 852728  2018-11-17   13:01           6646                      0644 BIKETOWN
## 870228  2018-12-19   14:43           6059                      0816 BIKETOWN
## 882057  2019-01-14    4:02           7120                      0314 BIKETOWN
## 884172  2019-01-18    0:55           6555                      0322 BIKETOWN
## 894634  2019-02-07    7:53           6650                      0814 BIKETOWN
## 900734  2019-02-21   15:32           7355 1006  DESIGN BIKE, NE CYCLING LIFE
## 988408  2019-05-23   13:44           7206                   0553 MSS BIKETOWN
## 1131150 2019-09-12   19:11           6635                      0250 BIKETOWN
## 1152647 2019-10-09   13:49           6229                    0501 XM BIKETOWN
## 1161463 2019-10-21   21:46           6027                      0674 BIKETOWN
## 1175708 2019-11-11   11:39           6656                      0643 BIKETOWN
## 1202118 2020-01-09    5:27           6671                     0852 BIKETOWN
## 1212359 2020-02-03   16:41           6861                      0541 BIKETOWN
## 1233320 2020-03-17   10:14           7340                   1025 VETERANS BIKE
##         Distance_Miles  Duration RentalAccessPath MultipleRental        Mph
## 28667          5249.56 17.500000          keypad          FALSE  17998.491
## 29073          5253.49 53.016667          keypad          FALSE   5945.478
## 119313         5249.02  9.800000          keypad          FALSE  32136.857
## 166368         5247.45 15.800000          keypad          FALSE  19927.025
## 256402         5248.97 12.033333          keypad          FALSE  26172.150
## 346528         5247.41 18.733333          keypad          FALSE  16806.651
## 391253         5248.72  7.716667          keypad          FALSE  40810.782
## 450310         5246.44  6.183333          keypad          FALSE  50908.852
## 508454         5248.84 10.283333          keypad          FALSE  30625.323
## 622385         5251.59 43.966667          keypad          FALSE   7166.688
## 651841         5247.64  3.383333          keypad           TRUE  93061.596
## 684532         6307.82 17.116667          keypad          FALSE  22111.151
## 808111         5994.67  6.333333          mobile          FALSE  56791.611
## 852728         6698.23 37.600000 keypad_rfid_card          FALSE  10688.665
## 870228         6113.09 19.233333 keypad_rfid_card          FALSE  19070.298
## 882057        10398.01 67.600000          mobile          FALSE   9229.003
## 884172        10015.78 22.700000          mobile          FALSE  26473.427
## 894634         6125.21  2.400000          mobile          FALSE 153130.250
## 900734        15527.18  3.000000 keypad_rfid_card          FALSE 310543.600
## 988408         5990.42 43.416667 keypad_rfid_card          FALSE   8278.507
## 1131150       11989.44 13.350000          keypad           TRUE  53885.124
## 1152647        7764.97 13.266667          mobile          FALSE  35117.955
## 1161463        5994.72  7.200000          keypad          FALSE  49956.000
## 1175708        5994.92  5.433333          keypad          FALSE  66201.571
## 1202118        1590.71  8.316667          keypad          FALSE  11476.064
## 1212359        1589.50 15.650000          mobile          FALSE   6093.930
## 1233320        1593.68  6.666667          mobile          FALSE  14343.120
```

While we could dig deeper into what makes these speed demons unique, we have elected not to carry out this step. Our main goal with this entire project is to provide deliverables that do not rely on these interesting observations. So we will conclude this notebook here, and possibly take up this endeavor in the future.

Final thoughts

There are so many other interesting observations and analyses that this data set could be used for. I wanted to look at some interesting observations found in the data not humanly possible, as noted in a previous notebook. I personally enjoy finding these types of observations that seemingly defy logic/reality (i.e. a bicycle rider traveling faster than an SR-71, or potential time travelers).

Future work

1. Find the URL for the leaflet/map plotting solution from Stack Overflow and add it in the appropriate location in this notebook.
2. Replace the absolute paths with relative paths for bringing the data into `R`.

Session information

Below you will find the output from `sessionInfo()` to assist in reproducing the work shown in this notebook.

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS  10.15.4
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] webshot_0.5.2 leaflet_2.0.3 sp_1.4-2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.4.6      lattice_0.20-41   ps_1.3.2          digest_0.6.25
##  [5] R6_2.4.1          grid_3.5.2        jsonlite_1.6.1    magrittr_1.5
##  [9] evaluate_0.14     rlang_0.4.5       stringi_1.4.6     callr_3.4.3
## [13] rmarkdown_2.1     tools_3.5.2       stringr_1.4.0     htmlwidgets_1.5.1
## [17] crosstalk_1.1.0.1 processx_3.4.2    xfun_0.13         yaml_2.2.1
## [21] compiler_3.5.2    htmltools_0.4.0   knitr_1.28
```