SF bikeshare load & prep

Michael Hutson

December 6, 2018

SF BikeShare

The San Francisco Bay Area bikeshare program started as a joint public/private initiative, and is now operated by Motivate with sponsorship from Ford. Ford makes some usage data publicly available, covering rides from 28 June 2017 to 31 October 2018 (at the time of this writing).

The variables in the data set are:

- **Trip Duration** - in seconds
- **Start Time and Date** - When each ride began (i.e., bike removed from a dock)
- **End Time and Date** - When each ride ended (i.e., bike returned to a dock)
- **Start Station ID** - a unique ID for each station location (same as End)
- **Start Station Name** - the landmark or street intersection of the station
- **Start Station Latitude** - self explanatory, though we will see some stations have multiple sets of coordinates
- **Start Station Longitude**
- **End Station ID** - a unique ID for each station location (same as Start)
- **End Station Name** - the landmark or street intersection of the station
- **End Station Latitude** - self explanatory, though we will see some stations have multiple sets of coordinates
- **End Station Longitude**
- **Bike ID** - a unique ID for each bike
- **User Type** - "Subscriber" = member or "Customer" = casual
- **Member Year of Birth** - user sets birth year
- **Member Gender** - user identifies as male, female, or other
- **Bike share for all trip** - subsidized subscription program -> user type must be "Subscriber""

Load libraries import data

If you have already unzipped the files, using fread() from the `data.table` package and then coercing to a tibble is faster than using read_csv() from `readr`.

```
library(data.table)
library(magrittr)
library(tidyverse)
library(lubridate)
library(sp)
library(elevatr)
library(knitr)
```

```
# load 2017 data
path <- file.path("data", "2017-fordgobike-tripdata.csv")
data2017 <- data.table::fread(path) %>% as.tbl()

# 2018 data are spread across 9 files, so list them out
file_names <- list.files(path = "./data", pattern = "*2018")
file_paths <- paste("./data/", file_names, sep = "")

# load 2018 data into a list
```

```
data2018 <- map(file_paths, data.table::fread) %>% map(as.tbl)

data2017 %>% head()
```

```
## # A tibble: 6 x 15
##   duration_sec start_time end_time start_station_id start_station_n~
##          <int> <chr>      <chr>               <int> <chr>
## 1        80110 2017-12-3~ 2018-01~               74 Laguna St at Ha~
## 2        78800 2017-12-3~ 2018-01~              284 Yerba Buena Cen~
## 3        45768 2017-12-3~ 2018-01~              245 Downtown Berkel~
## 4        62172 2017-12-3~ 2018-01~               60 8th St at Ringo~
## 5        43603 2017-12-3~ 2018-01~              239 Bancroft Way at~
## 6         9226 2017-12-3~ 2018-01~               30 San Francisco C~
## # ... with 10 more variables: start_station_latitude <dbl>,
## #   start_station_longitude <dbl>, end_station_id <int>,
## #   end_station_name <chr>, end_station_latitude <dbl>,
## #   end_station_longitude <dbl>, bike_id <int>, user_type <chr>,
## #   member_birth_year <int>, member_gender <chr>
```

Wrangling, Pt 1

Goal 1: combine all data into a single data frame

Let's try collapsing all the 2018 data from a list into a single data frame

```
data2018 <- bind_rows(data2018)
```

```
## Error in bind_rows_(x, .id): Column `start_station_id` can't be converted from integer to character
```

Fail! There is a column type mismatch for start station ID, and possibly others. Let's figure it out:

```
# function definition: for a list of data tables ("list"),
# check that a given attribute ("attrib") of the columns
# is consistent across all items in the list
compare_cols <- function(list, attrib) {
  for (i in 1:(length(list)-1))
    print(identical(map(list[[i]], attrib), map(list[[i+1]], attrib)))
}

# do all column names match across each month?
compare_cols(data2018, names)
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

```
# do all column classes match across each month?
compare_cols(data2018, class)
```

```
## [1] TRUE
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] FALSE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
```

```
# no! changes from mo. 5 to mo. 6

early_cols <- data2018[[5]] %>% sapply(class) %>% unname() # get col classes of month 5
late_cols <- data2018[[6]] %>% sapply(class) %>% unname() # get col classes of month 6

which(early_cols != late_cols)
```

```
## [1] 4 8
```

```
# cols 4 & 8 are different: start_station_id and end_station_id
rm(early_cols, late_cols)

data2018[[5]] %>% str()
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    179125 obs. of  16 variables:
##  $ duration_sec           : int  56791 52797 43204 67102 58883 22858 2863 3189 3149 3136 ...
##  $ start_time             : chr  "2018-05-31 21:41:51.4750" "2018-05-31 18:39:53.7690" "2018-05-31 2
##  $ end_time               : chr  "2018-06-01 13:28:22.7220" "2018-06-01 09:19:51.5410" "2018-06-01 0
##  $ start_station_id       : int  44 186 17 106 16 163 197 61 61 61 ...
##  $ start_station_name     : chr  "Civic Center/UN Plaza BART Station (Market St at McAllister St)" "
##  $ start_station_latitude : num  37.8 37.8 37.8 37.8 37.8 ...
##  $ start_station_longitude: num  -122 -122 -122 -122 -122 ...
##  $ end_station_id         : int  78 338 93 47 30 212 197 8 8 8 ...
##  $ end_station_name       : chr  "Folsom St at 9th St" "13th St at Franklin St" "4th St at Mission Ba
##  $ end_station_latitude   : num  37.8 37.8 37.8 37.8 37.8 ...
##  $ end_station_longitude  : num  -122 -122 -122 -122 -122 ...
##  $ bike_id                : int  1230 3414 2677 4224 3392 1235 152 1109 2143 3374 ...
##  $ user_type              : chr  "Customer" "Subscriber" "Customer" "Subscriber" ...
##  $ member_birth_year      : int  NA 1983 NA 1979 1986 1992 1985 NA NA NA ...
##  $ member_gender          : chr  "" "Male" "" "Male" ...
##  $ bike_share_for_all_trip: chr  "No" "No" "No" "No" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
data2018[[6]] %>% str()
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    195968 obs. of  16 variables:
##  $ duration_sec           : int  59088 60358 63654 50508 51697 36708 46380 7224 4294 2209 ...
##  $ start_time             : chr  "2018-06-30 23:32:44.6590" "2018-06-30 21:48:19.5570" "2018-06-30 2
##  $ end_time               : chr  "2018-07-01 15:57:33.3160" "2018-07-01 14:34:18.1000" "2018-07-01 1
##  $ start_station_id       : chr  "76" "248" "23" "58" ...
##  $ start_station_name     : chr  "McCoppin St at Valencia St" "Telegraph Ave at Ashby Ave" "The Embar
##  $ start_station_latitude : num  37.8 37.9 37.8 37.8 37.8 ...
##  $ start_station_longitude: num  -122 -122 -122 -122 -122 ...
##  $ end_station_id         : chr  "95" "239" "50" "88" ...
##  $ end_station_name       : chr  "Sanchez St at 15th St" "Bancroft Way at Telegraph Ave" "2nd St at T
##  $ end_station_latitude   : num  37.8 37.9 37.8 37.8 37.9 ...
```

```
##  $ end_station_longitude  : num  -122 -122 -122 -122 -122 ...
##  $ bike_id                : int  2100 653 3235 3675 3232 577 1764 779 2491 4225 ...
##  $ user_type              : chr  "Subscriber" "Customer" "Subscriber" "Subscriber" ...
##  $ member_birth_year      : int  1975 NA 1962 1992 1989 NA NA 1989 1996 1963 ...
##  $ member_gender          : chr  "Male" "" "Female" "Male" ...
##  $ bike_share_for_all_trip: chr  "Yes" "No" "No" "No" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

So the station IDs switch to character starting in June 2018. The same problem crops up when combining 2017 with 2018.

```
# make two internally-consistent subsets of the data
data2018a <- bind_rows(data2018[1:5])
data2018b <- bind_rows(data2018[-c(1:5)])

# change station ID columns to char
data2018a %<>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))
# now all 2018 data should have consistent formatting

# bind all 2018 data together, overwriting initial list
data2018 <- bind_rows(data2018a, data2018b)
# success

## now to combine 2017 and 2018
cols2017 <- data2017 %>% sapply(class) %>% unname() # col classes
cols2018 <- data2018 %>% sapply(class) %>% unname() # col classes

which(cols2017 != cols2018)
```

```
## Warning in cols2017 != cols2018: longer object length is not a multiple of
## shorter object length
```

```
## [1]  4  8 16
```

```
str(data2017)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    519700 obs. of  15 variables:
##  $ duration_sec           : int  80110 78800 45768 62172 43603 9226 4507 4334 4150 4238 ...
##  $ start_time             : chr  "2017-12-31 16:57:39.6540" "2017-12-31 15:56:34.8420" "2017-12-31 2
##  $ end_time               : chr  "2018-01-01 15:12:50.2450" "2018-01-01 13:49:55.6170" "2018-01-01 1
##  $ start_station_id       : int  74 284 245 60 239 30 259 284 20 20 ...
##  $ start_station_name     : chr  "Laguna St at Hayes St" "Yerba Buena Center for the Arts (Howard St
##  $ start_station_latitude : num  37.8 37.8 37.9 37.8 37.9 ...
##  $ start_station_longitude: num  -122 -122 -122 -122 -122 ...
##  $ end_station_id         : int  43 96 245 5 247 30 259 284 20 20 ...
##  $ end_station_name       : chr  "San Francisco Public Library (Grove St at Hyde St)" "Dolores St at
##  $ end_station_latitude   : num  37.8 37.8 37.9 37.8 37.9 ...
##  $ end_station_longitude  : num  -122 -122 -122 -122 -122 ...
##  $ bike_id                : int  96 88 1094 2831 3167 1487 3539 1503 3125 2543 ...
##  $ user_type              : chr  "Customer" "Customer" "Customer" "Customer" ...
##  $ member_birth_year      : int  1987 1965 NA NA 1997 NA 1991 NA NA NA ...
##  $ member_gender          : chr  "Male" "Female" "" "" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(data2018)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    1732358 obs. of  16 variables:
##  $ duration_sec           : int  75284 85422 71576 61076 39966 6477 453 180 996 825 ...
##  $ start_time             : chr  "2018-01-31 22:52:35.2390" "2018-01-31 16:13:34.3510" "2018-01-31 14
##  $ end_time               : chr  "2018-02-01 19:47:19.8240" "2018-02-01 15:57:17.3100" "2018-02-01 10
##  $ start_station_id       : chr  "120" "15" "304" "75" ...
##  $ start_station_name     : chr  "Mission Dolores Park" "San Francisco Ferry Building (Harry Bridges
##  $ start_station_latitude : num  37.8 37.8 37.3 37.8 37.8 ...
##  $ start_station_longitude: num  -122 -122 -122 -122 -122 ...
##  $ end_station_id         : chr  "285" "15" "296" "47" ...
##  $ end_station_name       : chr  "Webster St at O'Farrell St" "San Francisco Ferry Building (Harry Bi
##  $ end_station_latitude   : num  37.8 37.8 37.3 37.8 37.8 ...
##  $ end_station_longitude  : num  -122 -122 -122 -122 -122 ...
##  $ bike_id                : int  2765 2815 3039 321 617 1306 3571 1403 3675 1453 ...
##  $ user_type              : chr  "Subscriber" "Customer" "Customer" "Customer" ...
##  $ member_birth_year      : int  1986 NA 1996 NA 1991 NA 1988 1980 1987 1994 ...
##  $ member_gender          : chr  "Male" "" "Male" "" ...
##  $ bike_share_for_all_trip: chr  "No" "No" "No" "No" ...
```

```
# cols 4 & 8 are different, and col 16 is new for 2018

# change station ID columns to char
data2017 %<>%
  mutate(start_station_id = as.character(start_station_id),
         end_station_id = as.character(end_station_id))

# combine years
data <- data2017 %>% bind_rows(data2018)

# cleanup
rm(cols2017, cols2018, data2017, data2018, data2018a, data2018b, path, file_names, file_paths)

str(data)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    2252058 obs. of  16 variables:
##  $ duration_sec           : int  80110 78800 45768 62172 43603 9226 4507 4334 4150 4238 ...
##  $ start_time             : chr  "2017-12-31 16:57:39.6540" "2017-12-31 15:56:34.8420" "2017-12-31 21
##  $ end_time               : chr  "2018-01-01 15:12:50.2450" "2018-01-01 13:49:55.6170" "2018-01-01 11
##  $ start_station_id       : chr  "74" "284" "245" "60" ...
##  $ start_station_name     : chr  "Laguna St at Hayes St" "Yerba Buena Center for the Arts (Howard St
##  $ start_station_latitude : num  37.8 37.8 37.9 37.8 37.9 ...
##  $ start_station_longitude: num  -122 -122 -122 -122 -122 ...
##  $ end_station_id         : chr  "43" "96" "245" "5" ...
##  $ end_station_name       : chr  "San Francisco Public Library (Grove St at Hyde St)" "Dolores St at
##  $ end_station_latitude   : num  37.8 37.8 37.9 37.8 37.9 ...
##  $ end_station_longitude  : num  -122 -122 -122 -122 -122 ...
##  $ bike_id                : int  96 88 1094 2831 3167 1487 3539 1503 3125 2543 ...
##  $ user_type              : chr  "Customer" "Customer" "Customer" "Customer" ...
##  $ member_birth_year      : int  1987 1965 NA NA 1997 NA 1991 NA NA NA ...
##  $ member_gender          : chr  "Male" "Female" "" "" ...
##  $ bike_share_for_all_trip: chr  NA NA NA NA ...
```

Success!

Goal 2: set the classes of variables

and a few more useful tweaks

Ridership patterns might cycle every week, so pull out the day of the week from ride start. And, come to think of it, the biggest difference might be by weekday vs weekend.

```
# convert column types
data %<>%
  mutate(member_gender = as.factor(member_gender),
         user_type = as.factor(user_type),
         bike_share_for_all_trip = as.factor(bike_share_for_all_trip),
         start_time = ymd_hms(start_time),
         end_time = ymd_hms(end_time),
         bike_id = as.character(bike_id))

# create new variables: weekday (split out the day of the week), hour (ride start hour)
data %<>%
  mutate(weekday = factor(wday(start_time, label = TRUE), ordered = FALSE),
         hour = hour(start_time))

# simplify weekday even further, to binary is_weekend
data %<>%
  mutate(is_weekend = ifelse(weekday %in% c("Sat", "Sun"), "wk_end", "wk_day"),
         is_weekend = as.factor(is_weekend))

# earlier, I converted station IDs to character. What do they look like?
data$start_station_id %>% unique() %>% sort()
```

```
##   [1] "10"   "100"  "101"  "102"  "104"  "105"  "106"  "107"  "108"  "109"
##  [11] "11"   "110"  "112"  "113"  "114"  "115"  "116"  "118"  "119"  "120"
##  [21] "121"  "122"  "123"  "124"  "125"  "126"  "127"  "129"  "13"   "130"
##  [31] "131"  "132"  "133"  "134"  "136"  "137"  "138"  "139"  "14"   "140"
##  [41] "141"  "142"  "144"  "145"  "146"  "147"  "148"  "149"  "15"   "150"
##  [51] "151"  "152"  "153"  "154"  "155"  "156"  "157"  "158"  "159"  "16"
##  [61] "160"  "162"  "163"  "164"  "166"  "167"  "168"  "169"  "17"   "170"
##  [71] "171"  "172"  "173"  "174"  "175"  "176"  "177"  "178"  "179"  "18"
##  [81] "180"  "181"  "182"  "183"  "184"  "185"  "186"  "187"  "188"  "189"
##  [91] "19"   "190"  "191"  "192"  "193"  "194"  "195"  "196"  "197"  "198"
## [101] "199"  "20"   "200"  "201"  "202"  "203"  "204"  "205"  "206"  "207"
## [111] "208"  "209"  "21"   "210"  "211"  "212"  "213"  "214"  "215"  "216"
## [121] "217"  "218"  "219"  "22"   "220"  "221"  "222"  "223"  "224"  "225"
## [131] "226"  "227"  "228"  "229"  "23"   "230"  "231"  "232"  "233"  "234"
## [141] "235"  "236"  "237"  "238"  "239"  "24"   "240"  "241"  "242"  "243"
## [151] "244"  "245"  "246"  "247"  "248"  "249"  "25"   "250"  "251"  "252"
## [161] "253"  "254"  "255"  "256"  "257"  "258"  "259"  "26"   "262"  "263"
## [171] "265"  "266"  "267"  "268"  "269"  "27"   "270"  "271"  "272"  "273"
## [181] "274"  "275"  "276"  "277"  "278"  "279"  "28"   "280"  "281"  "282"
## [191] "283"  "284"  "285"  "286"  "287"  "288"  "289"  "29"   "290"  "291"
## [201] "292"  "293"  "294"  "295"  "296"  "297"  "298"  "299"  "3"    "30"
## [211] "300"  "301"  "302"  "303"  "304"  "305"  "306"  "307"  "308"  "309"
## [221] "31"   "310"  "311"  "312"  "313"  "314"  "315"  "316"  "317"  "318"
## [231] "321"  "323"  "324"  "327"  "33"   "336"  "337"  "338"  "339"  "34"
## [241] "340"  "341"  "342"  "343"  "344"  "345"  "347"  "349"  "35"   "350"
## [251] "351"  "355"  "356"  "357"  "358"  "359"  "36"   "360"  "361"  "362"
```

```
## [261] "363"  "364"  "365"  "367"  "368"  "369"  "37"   "370"  "371"  "372"
## [271] "373"  "374"  "375"  "377"  "378"  "381"  "39"   "4"    "40"   "41"
## [281] "42"   "43"   "44"   "45"   "46"   "47"   "48"   "49"   "5"    "50"
## [291] "52"   "53"   "55"   "56"   "58"   "59"   "6"    "60"   "61"   "62"
## [301] "63"   "64"   "66"   "67"   "7"    "70"   "71"   "72"   "73"   "74"
## [311] "75"   "76"   "77"   "78"   "79"   "8"    "80"   "81"   "84"   "85"
## [321] "86"   "87"   "88"   "89"   "9"    "90"   "91"   "92"   "93"   "95"
## [331] "96"   "97"   "98"   "99"   "NULL"
```

```
# turn them back into numeric, converting "NULL" values into NAs
data %<>%
  mutate(start_station_id = as.numeric(start_station_id),
         end_station_id = as.numeric(end_station_id))
```

```
## Warning in evalq(as.numeric(start_station_id), <environment>): NAs
## introduced by coercion
```

```
## Warning in evalq(as.numeric(end_station_id), <environment>): NAs introduced
## by coercion
```

```
summary(data$start_station_id)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     3.0    30.0    81.0   113.2   180.0   381.0   11579
```

```
# convert "NULL" station names into NAs
data %<>%
  mutate(start_station_name = replace(start_station_name, which(start_station_name == "NULL"), NA),
         end_station_name = replace(end_station_name, which(end_station_name == "NULL"), NA))
```

Goal 3: address the NA stations

What is going on with all those NA stations? Count them up, and cross-check names and IDs:

```
# how many unknown stations?
sum(is.na(data$start_station_id))
```

```
## [1] 11579
```

```
sum(is.na(data$start_station_name))
```

```
## [1] 11579
```

```
data %>%
  filter(is.na(start_station_name)) %>%
  group_by(start_station_id) %>%
  summarise(count = n())
```

```
## # A tibble: 1 x 2
##   start_station_id count
##              <dbl> <int>
## 1               NA 11579
```

```
# all unnamed stations lack IDs

data %>%
  filter(is.na(start_station_id)) %>%
  group_by(start_station_name) %>%
  summarise(count = n())
```

```
## # A tibble: 1 x 2
##   start_station_name count
##   <chr>              <int>
## 1 <NA>               11579
```

```
# all no-ID stations also lack names - and it's the same number, so 1:1 match
```

For now, I will create a dataset without all those NA stations, since they are a small fraction of total. But we will return to them later, since there might be an important reason for all those NAs.

```
# what fraction?
sum(is.na(data$start_station_name)) / nrow(data)
```

```
## [1] 0.005141519
```

```
# approx. 0.5%

# further, all stations starting at NA also end at NA:
data %>%
  filter(is.na(start_station_name)) %>%
  group_by(end_station_name) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

```
## # A tibble: 1 x 2
##   end_station_name count
##   <chr>            <int>
## 1 <NA>             11579
```

```
# do the deed
data_clean <- data %>%
  filter(!is.na(start_station_name))
# now, all rides are to/from named stations

summary(data_clean)
```

```
##   duration_sec     start_time
##  Min.   :   61   Min.   :2017-06-28 09:47:36
##  1st Qu.:  358   1st Qu.:2018-01-16 16:29:06
##  Median :  566   Median :2018-05-28 14:18:18
##  Mean   :  915   Mean   :2018-05-01 19:31:28
##  3rd Qu.:  887   3rd Qu.:2018-08-24 18:31:31
##  Max.   :86369   Max.   :2018-11-30 23:58:26
##
##     end_time                  start_station_id start_station_name
##  Min.   :2017-06-28 09:52:55   Min.   :  3.0    Length:2240479
##  1st Qu.:2018-01-16 16:41:27   1st Qu.: 30.0    Class :character
##  Median :2018-05-28 14:39:19   Median : 81.0    Mode  :character
##  Mean   :2018-05-01 19:46:44   Mean   :113.2
##  3rd Qu.:2018-08-24 18:44:16   3rd Qu.:180.0
##  Max.   :2018-12-01 08:07:53   Max.   :381.0
##
##  start_station_latitude start_station_longitude end_station_id
##  Min.   :37.26          Min.   :-122.5          Min.   :  3.0
##  1st Qu.:37.77          1st Qu.:-122.4          1st Qu.: 28.0
##  Median :37.78          Median :-122.4          Median : 81.0
##  Mean   :37.77          Mean   :-122.4          Mean   :111.4
```

```
##  3rd Qu.:37.80          3rd Qu.:-122.3          3rd Qu.:179.0
##  Max.   :37.88          Max.   :-121.8          Max.   :381.0
##
##  end_station_name   end_station_latitude end_station_longitude
##  Length:2240479     Min.   :37.26        Min.   :-122.5
##  Class :character   1st Qu.:37.77        1st Qu.:-122.4
##  Mode  :character   Median :37.78        Median :-122.4
##                     Mean   :37.77        Mean   :-122.4
##                     3rd Qu.:37.80        3rd Qu.:-122.3
##                     Max.   :37.88        Max.   :-121.8
##
##    bike_id              user_type       member_birth_year
##  Length:2240479     Customer  : 370470   Min.   :1881
##  Class :character   Subscriber:1870009   1st Qu.:1977
##  Mode  :character                        Median :1985
##                                          Mean   :1982
##                                          3rd Qu.:1990
##                                          Max.   :2000
##                                          NA's   :171926
##  member_gender    bike_share_for_all_trip weekday           hour
##        : 171496   No  :1572618            Sun:185978   Min.   : 0.0
##  Female: 504593   Yes : 148161            Mon:352019   1st Qu.: 9.0
##  Male  :1533139   NA's: 519700            Tue:385160   Median :14.0
##  Other :  31251                           Wed:381171   Mean   :13.5
##                                           Thu:375377   3rd Qu.:17.0
##                                           Fri:350477   Max.   :23.0
##                                           Sat:210297
##   is_weekend
##  wk_day:1844204
##  wk_end: 396275
##
##
##
##
##
```

Wrangling, Pt 2

Goal 1: Summarize arrival/departure stats by station

```
departures <- data_clean %>%
  group_by(start_station_name) %>%
  summarise(departure_count = n()) %>%
  rename(station_name = start_station_name)

arrivals <- data_clean %>%
  group_by(end_station_name) %>%
  summarise(arrival_count = n()) %>%
  rename(station_name = end_station_name)

station_stats <- departures

# expand station_stats w/ arrivals, net change, proportional change
station_stats <- station_stats %>%
```

```
  left_join(arrivals) %>%
  mutate(net_change = arrival_count - departure_count,
         prop_inflow = net_change/departure_count)
```

```
## Joining, by = "station_name"
```

```
rm(arrivals, departures)

head(station_stats)
```

```
## # A tibble: 6 x 5
##   station_name         departure_count arrival_count net_change prop_inflow
##   <chr>                          <int>         <int>      <int>       <dbl>
## 1 10th Ave at E 15th ~             831           849         18      0.0217
## 2 10th St at Fallon St            4804          5837       1033      0.215
## 3 10th St at Universi~             610          1020        410      0.672
## 4 11th St at Bryant St           10621         12230       1609      0.151
## 5 11th St at Natoma St            9806         10010        204      0.0208
## 6 12th St at 4th Ave              2520          2628        108      0.0429
```

Hmm, already we can see that some station IDs are repeated under slightly different names. Let's dig in to this a little.

```
# make a table of station info (name, id, lat, long)
station_info <- data_clean %>%
  group_by(start_station_id,
           start_station_name,
           start_station_latitude,
           start_station_longitude) %>% # this is the most specific grouping of stations by location
  summarize(count = n()) %>%
  group_by(start_station_name) %>%
  rename(station_name = start_station_name,
         station_id = start_station_id,
         station_latitude = start_station_latitude,
         station_longitude = start_station_longitude)

station_info
```

```
## # A tibble: 361 x 5
## # Groups:   station_name [351]
##    station_id station_name          station_latitude station_longitu~ count
##         <dbl> <chr>                            <dbl>            <dbl> <int>
##  1          3 Powell St BART Stati~             37.8            -122. 39518
##  2          4 Cyril Magnin St at E~             37.8            -122.  8066
##  3          5 Powell St BART Stati~             37.8            -122. 31159
##  4          6 The Embarcadero at S~             37.8            -122. 45169
##  5          7 Frank H Ogawa Plaza               37.8            -122. 11058
##  6          8 The Embarcadero at V~             37.8            -122. 15009
##  7          9 Broadway at Battery ~             37.8            -122. 13600
##  8         10 Washington St at Kea~             37.8            -122.  9542
##  9         11 Davis St at Jackson ~             37.8            -122. 12464
## 10         13 Commercial St at Mon~             37.8            -122. 11485
## # ... with 351 more rows
```

```
length(unique(station_info$station_id))
```

```
## [1] 334
```

```
length(unique(station_info$station_name))
```

```
## [1] 351
```

```
length(unique(data_clean$start_station_latitude))
```

```
## [1] 354
```

```
length(unique(data_clean$start_station_longitude))
```

```
## [1] 353
```

At last check:

- 356 unique lat/long combos
- 349 unique latitudes
- 348 unique longitudes
- 346 unique station names
- 331 unique station IDs

Let's find the repeat stations

```
station_count <- station_info %>%
  summarize(station_count = n()) %>%
  arrange(desc(station_count))

head(station_count, n = 10)
```

```
## # A tibble: 10 x 2
##    station_name                    station_count
##    <chr>                                   <int>
##  1 Shattuck Ave at Hearst Ave                  3
##  2 21st Ave at International Blvd              2
##  3 22nd St Caltrain Station                    2
##  4 37th St at West St                          2
##  5 Downtown Berkeley BART                      2
##  6 Doyle St at 59th St                         2
##  7 North Berkeley BART Station                 2
##  8 S. 4th St at San Carlos St                  2
##  9 Tamien Station                              2
## 10 10th Ave at E 15th St                       1
```

```
id_count <- station_info %>%
  group_by(station_id) %>%
  summarise(n_per_id = n()) %>%
  arrange(desc(n_per_id))

head(id_count, n = 20)
```

```
## # A tibble: 20 x 2
##    station_id n_per_id
##         <dbl>    <int>
##  1        192        3
##  2        205        3
##  3        208        3
##  4        221        3
##  5        233        3
```

```
##  6        244        3
##  7         50        2
##  8        101        2
##  9        130        2
## 10        154        2
## 11        173        2
## 12        212        2
## 13        224        2
## 14        234        2
## 15        245        2
## 16        250        2
## 17        280        2
## 18        281        2
## 19        302        2
## 20        321        2
```

19 station IDs are associated with >1 lat/long location. 9 station names are associated with >1 lat/long locations. Probably different dock locations at the same general location?

The docks might be operated simultaneously, or sequentially (repositioned over time). Can check this later if necessary.

```
# join station stats
station_stats <- station_stats %>%
  full_join(station_info) %>%
  full_join(id_count) %>%
  full_join(station_count) %>%
  select(station_id, n_per_id,
         station_name,
         station_count,
         station_latitude,
         station_longitude,
         departure_count:prop_inflow) %>%
  arrange(desc(n_per_id), station_id)
```

```
## Joining, by = "station_name"
```

```
## Joining, by = "station_id"
```

```
## Joining, by = "station_name"
```

```
rm(id_count, station_info, station_count)

station_stats %>% head(n = 20)
```

```
## # A tibble: 20 x 10
##    station_id n_per_id station_name station_count station_latitude
##         <dbl>    <int> <chr>                <int>            <dbl>
##  1        192        3 37th St at ~             2             37.8
##  2        192        3 37th St at ~             2             37.8
##  3        192        3 MLK Jr Way ~             1             37.8
##  4        205        3 Miles Ave a~             1             37.8
##  5        205        3 Miles Ave a~             1             37.8
##  6        205        3 Shafter Ave~             1             37.8
##  7        208        3 S. 4th St a~             2             37.3
##  8        208        3 S. 4th St a~             2             37.3
##  9        208        3 William St ~             1             37.3
```

```
## 10        221        3 12th St at ~             1             37.8
## 11        221        3 6th Ave at ~             1             37.8
## 12        221        3 E 12th St a~             1             37.8
## 13        233        3 12th St at ~             1             37.8
## 14        233        3 4th Ave at ~             1             37.8
## 15        233        3 E 12th St a~             1             37.8
## 16        244        3 Shattuck Av~             3             37.9
## 17        244        3 Shattuck Av~             3             37.9
## 18        244        3 Shattuck Av~             3             37.9
## 19         50        2 2nd St at T~             1             37.8
## 20         50        2 2nd St at T~             1             37.8
## # ... with 5 more variables: station_longitude <dbl>,
## #   departure_count <int>, arrival_count <int>, net_change <int>,
## #   prop_inflow <dbl>
```

Let's investigate further: why do some stations seem to repeat silently, i.e. identical name but appear 2+ times?

```
temp <- station_stats %>%
  arrange(desc(station_count)) %>%
  filter(station_count > 1)

kable(temp)
```

station_id	n_per_id	station_name	station_count	station_latitude	station_longitude	departur
244	3	Shattuck Ave at Hearst Ave	3	37.87368	-122.2685	
244	3	Shattuck Ave at Hearst Ave	3	37.87375	-122.2686	
244	3	Shattuck Ave at Hearst Ave	3	37.87379	-122.2686	
192	3	37th St at West St	2	37.82670	-122.2718	
192	3	37th St at West St	2	37.82670	-122.2718	
208	3	S. 4th St at San Carlos St	2	37.33004	-121.8818	
208	3	S. 4th St at San Carlos St	2	37.33284	-121.8839	
130	2	22nd St Caltrain Station	2	37.75737	-122.3921	
130	2	22nd St Caltrain Station	2	37.75772	-122.3918	
154	2	Doyle St at 59th St	2	37.84192	-122.2880	
154	2	Doyle St at 59th St	2	37.84192	-122.2880	
224	2	21st Ave at International Blvd	2	37.78485	-122.2393	
224	2	21st Ave at International Blvd	2	37.78516	-122.2389	
245	2	Downtown Berkeley BART	2	37.87014	-122.2684	
245	2	Downtown Berkeley BART	2	37.87035	-122.2678	
250	2	North Berkeley BART Station	2	37.87356	-122.2831	
250	2	North Berkeley BART Station	2	37.87401	-122.2830	
302	2	Tamien Station	2	37.31285	-121.8829	
302	2	Tamien Station	2	37.34772	-121.8909	

```
# Shattuck at Hearst has very similar - but not identical - lat/long coordinates. What about the rest?
i <- seq(from = 4, to = 18, by = 2)
temp$station_latitude[i] - temp$station_latitude[i+1]
```

```
## [1] -6.135903e-07 -2.795780e-03 -3.494316e-04 -7.105427e-15 -3.019377e-04
## [6] -2.087000e-04 -4.561000e-04 -3.486707e-02
```

```
temp$station_longitude[i] - temp$station_longitude[i+1]
```

```
## [1] -1.560966e-06  2.091086e-03 -2.440810e-04  0.000000e+00 -3.896810e-04
## [6] -6.583000e-04 -7.400000e-05  7.915199e-03
```

```
identical(temp$station_longitude[2], temp$station_longitude[3])
```

```
## [1] FALSE
```

```
identical(temp$station_longitude[10], temp$station_longitude[11])
```

```
## [1] TRUE
```

```
rm(i, temp)
```

One station has identical longitudes but differing latitudes for its docks - this accounts for discrepancy btw the number of unique latitudes vs longitudes.

So the "silent" repeat stations (i.e., no name change but different lat-long) are: * Shattuck Ave at Hearst Ave; * 37th at West * S. 4th St at San Carlos St; * Doyle St at 59th St; * North Berkeley BART Station; and * Tamien Station

All have identical arrival and departure counts, since I initially grouped them by name. So we should pick a single set of lat-long coordinates for each.

```
# check the duplicate stations
station_stats %>% arrange(desc(station_count)) %>% print(n = 20)
```

```
## # A tibble: 361 x 10
##    station_id n_per_id station_name station_count station_latitude
##         <dbl>    <int> <chr>                <int>            <dbl>
##  1        244        3 Shattuck Av~             3             37.9
##  2        244        3 Shattuck Av~             3             37.9
##  3        244        3 Shattuck Av~             3             37.9
##  4        192        3 37th St at ~             2             37.8
##  5        192        3 37th St at ~             2             37.8
##  6        208        3 S. 4th St a~             2             37.3
##  7        208        3 S. 4th St a~             2             37.3
##  8        130        2 22nd St Cal~             2             37.8
##  9        130        2 22nd St Cal~             2             37.8
## 10        154        2 Doyle St at~             2             37.8
## 11        154        2 Doyle St at~             2             37.8
## 12        224        2 21st Ave at~             2             37.8
## 13        224        2 21st Ave at~             2             37.8
## 14        245        2 Downtown Be~             2             37.9
## 15        245        2 Downtown Be~             2             37.9
## 16        250        2 North Berke~             2             37.9
## 17        250        2 North Berke~             2             37.9
## 18        302        2 Tamien Stat~             2             37.3
## 19        302        2 Tamien Stat~             2             37.3
## 20        192        3 MLK Jr Way ~             1             37.8
## # ... with 341 more rows, and 5 more variables: station_longitude <dbl>,
## #   departure_count <int>, arrival_count <int>, net_change <int>,
## #   prop_inflow <dbl>
```

```
duplicates <- which(station_stats$station_count > 1)
station_stats[duplicates,] # yep, checks out
```

```
## # A tibble: 19 x 10
##    station_id n_per_id station_name station_count station_latitude
##         <dbl>    <int> <chr>                <int>            <dbl>
```

```
##  1        192        3 37th St at ~             2             37.8
##  2        192        3 37th St at ~             2             37.8
##  3        208        3 S. 4th St a~             2             37.3
##  4        208        3 S. 4th St a~             2             37.3
##  5        244        3 Shattuck Av~             3             37.9
##  6        244        3 Shattuck Av~             3             37.9
##  7        244        3 Shattuck Av~             3             37.9
##  8        130        2 22nd St Cal~             2             37.8
##  9        130        2 22nd St Cal~             2             37.8
## 10        154        2 Doyle St at~             2             37.8
## 11        154        2 Doyle St at~             2             37.8
## 12        224        2 21st Ave at~             2             37.8
## 13        224        2 21st Ave at~             2             37.8
## 14        245        2 Downtown Be~             2             37.9
## 15        245        2 Downtown Be~             2             37.9
## 16        250        2 North Berke~             2             37.9
## 17        250        2 North Berke~             2             37.9
## 18        302        2 Tamien Stat~             2             37.3
## 19        302        2 Tamien Stat~             2             37.3
## # ... with 5 more variables: station_longitude <dbl>,
## #   departure_count <int>, arrival_count <int>, net_change <int>,
## #   prop_inflow <dbl>
```

```
extras <- duplicates[c(2,4,6,7,9,11,13,15,17,19)]

# delete the extras
station_stats <- station_stats[-extras,]
rm(duplicates, extras)
```

And, now that there is only one set of lat-long coordinates per station name, station_count is superfluous, and n_per_id is outdated

```
station_stats <- station_stats %>%
  select(-station_count, -n_per_id)

station_stats %>% print(n = 10)
```

```
## # A tibble: 351 x 8
##    station_id station_name station_latitude station_longitu~
##         <dbl> <chr>                  <dbl>            <dbl>
##  1        192 37th St at ~            37.8            -122.
##  2        192 MLK Jr Way ~            37.8            -122.
##  3        205 Miles Ave a~            37.8            -122.
##  4        205 Miles Ave a~            37.8            -122.
##  5        205 Shafter Ave~            37.8            -122.
##  6        208 S. 4th St a~            37.3            -122.
##  7        208 William St ~            37.3            -122.
##  8        221 12th St at ~            37.8            -122.
##  9        221 6th Ave at ~            37.8            -122.
## 10        221 E 12th St a~            37.8            -122.
## # ... with 341 more rows, and 4 more variables: departure_count <int>,
## #   arrival_count <int>, net_change <int>, prop_inflow <dbl>
```

```
# recalculate station ID counts
new_id_counts <- station_stats %>%
  group_by(station_id) %>%
```

```
  summarise(n_per_id = n()) %>%
  arrange(desc(n_per_id))

# add in the new station ID counts
station_stats <- station_stats %>%
  full_join(new_id_counts) %>%
  select(station_id, n_per_id, station_name:prop_inflow) %>%
  arrange(desc(n_per_id))
```

```
## Joining, by = "station_id"
```

```
rm(new_id_counts)

station_stats %>% filter(n_per_id > 1) %>% kable()
```

station_id	n_per_id	station_name	station_latitude	station_longitude	d
205	3	Miles Ave at Cavour St	37.83880	-122.2587	
205	3	Miles Ave at Cavour St (Temporary Location)	37.83880	-122.2587	
205	3	Shafter Ave at Cavour St	37.83795	-122.2572	
221	3	12th St at 6th Ave	37.79435	-122.2539	
221	3	6th Ave at E 12th St (Temporary Location)	37.79440	-122.2538	
221	3	E 12th St at 6th Ave	37.79435	-122.2539	
233	3	12th St at 4th Ave	37.79581	-122.2556	
233	3	4th Ave at E 12th St (Temporary Location)	37.79591	-122.2555	
233	3	E 12th St at 4th Ave	37.79581	-122.2556	
192	2	37th St at West St	37.82670	-122.2718	
192	2	MLK Jr Way at 36th St (Temporary Location)	37.82579	-122.2694	
208	2	S. 4th St at San Carlos St	37.33004	-121.8818	
208	2	William St at 4th St (Temporary Location)	37.32996	-121.8819	
50	2	2nd St at Townsend St	37.78053	-122.3903	
50	2	2nd St at Townsend St - Coming Soon	37.78053	-122.3903	
101	2	Potrero Ave at 15th St (Temporary Location)	37.76663	-122.4077	
101	2	San Bruno Ave at 16th St	37.76601	-122.4057	
173	2	Shattuck Ave at 55th Ave	37.84036	-122.2645	
173	2	Shattuck Ave at 55th St	37.84036	-122.2645	
212	2	Mosswood Park	37.82493	-122.2605	
212	2	Webster St at MacArthur Blvd (Temporary Location)	37.82501	-122.2616	
234	2	Farnam St at Fruitvale Ave	37.77806	-122.2254	
234	2	Fruitvale Ave at International Blvd	37.77768	-122.2258	
280	2	6th St at San Fernando St (Temporary Location)	37.33704	-121.8841	
280	2	San Fernando at 7th St	37.33725	-121.8831	
281	2	9th St at San Fernando	37.33840	-121.8808	
281	2	9th St at San Fernando St	37.33840	-121.8808	
321	2	5th at Folsom	37.78015	-122.4031	
321	2	5th St at Folsom	37.78015	-122.4031	
358	2	Lane St at Van Dyke Ave	37.72925	-122.3924	
358	2	Williams Ave at 3rd St	37.72928	-122.3929	

So we can see how names can differ slightly, and therefore station use stats appear separate. To lump by ID, would need to recalculate.

Overall, station ID is associated w/ a general location, and different names (& data) are associated w/ slightly different positions of that station. So, either do the analysis by a single ID (and average the lat/long, and

re-calculate ridership by station ID) or keep stations separate by name for greater geographic precision.

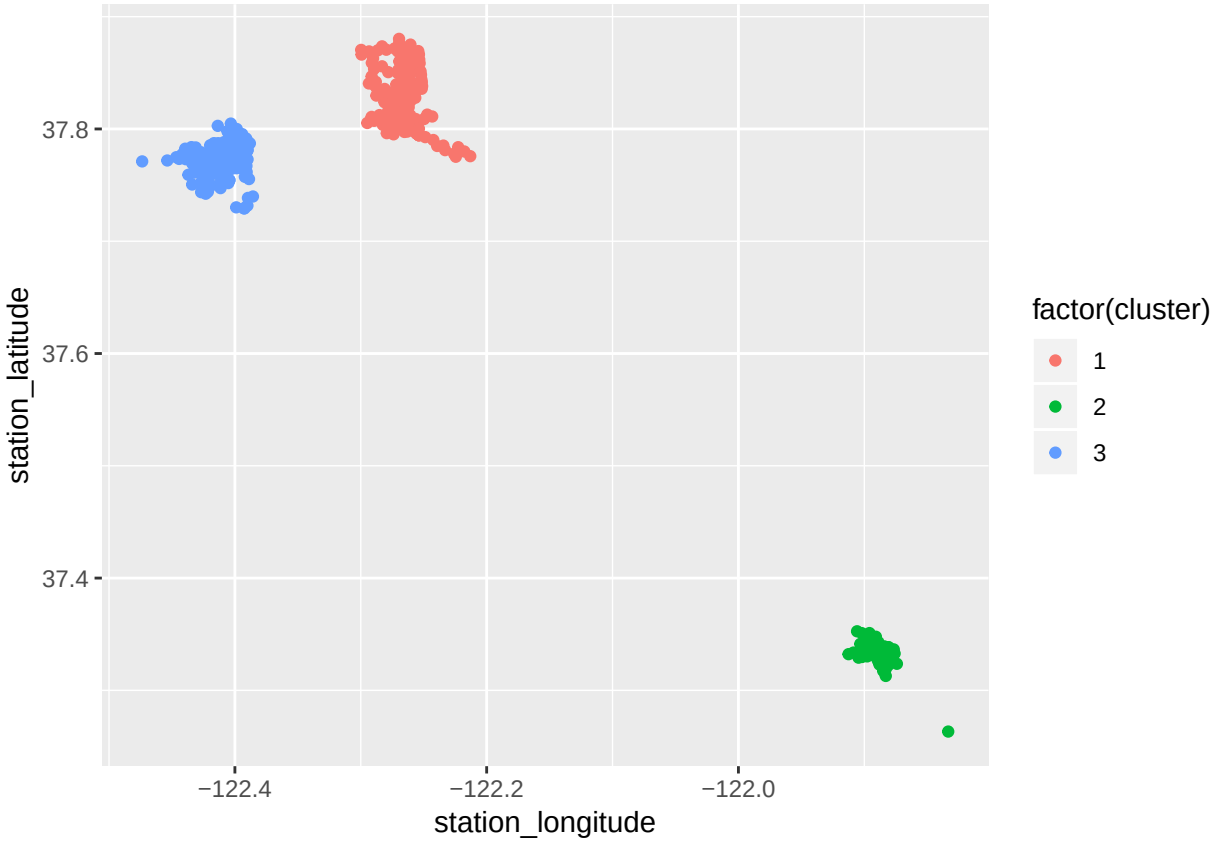
Goal 2: identify the city in which each station is located

```
# cluster analysis of stations by lat-long, just for practice

station_locations <- station_stats %>% select(station_latitude, station_longitude)
dist_stations <- dist(station_locations, method = "euclidean")
# no crossing the international date line, no rescaling, no problem!
hc_stations <- hclust(dist_stations, method = "complete")

# extract clusters
cluster_assigments <- cutree(hc_stations, k = 3)
stations_clustered <- mutate(station_locations, cluster = cluster_assigments)
```

Plot the stations in lat-long space, colored by cluster



Looking good. Now tag each station by its cluster, and rename the clusters as appropriate. In fact, even without the cluster analysis, it looks clear that SF, East Bay, and SJ stations do not overlap by longitude, so we can use that to label stations.

```
stations_clustered <- stations_clustered %>%
  mutate(cluster = recode(cluster, '1' = "EastBay",
                          '2' = "SanJose",
                          '3' = "SanFrancisco")) %>%
  rename(city = cluster)

stations_clustered %>%
```

```
  group_by(city) %>%
  summarise(min_long = min(station_longitude),
            max_long = max(station_longitude)) %>%
  kable(caption = "range of station longitudes, by city")
```

Table 3: range of station longitudes, by city

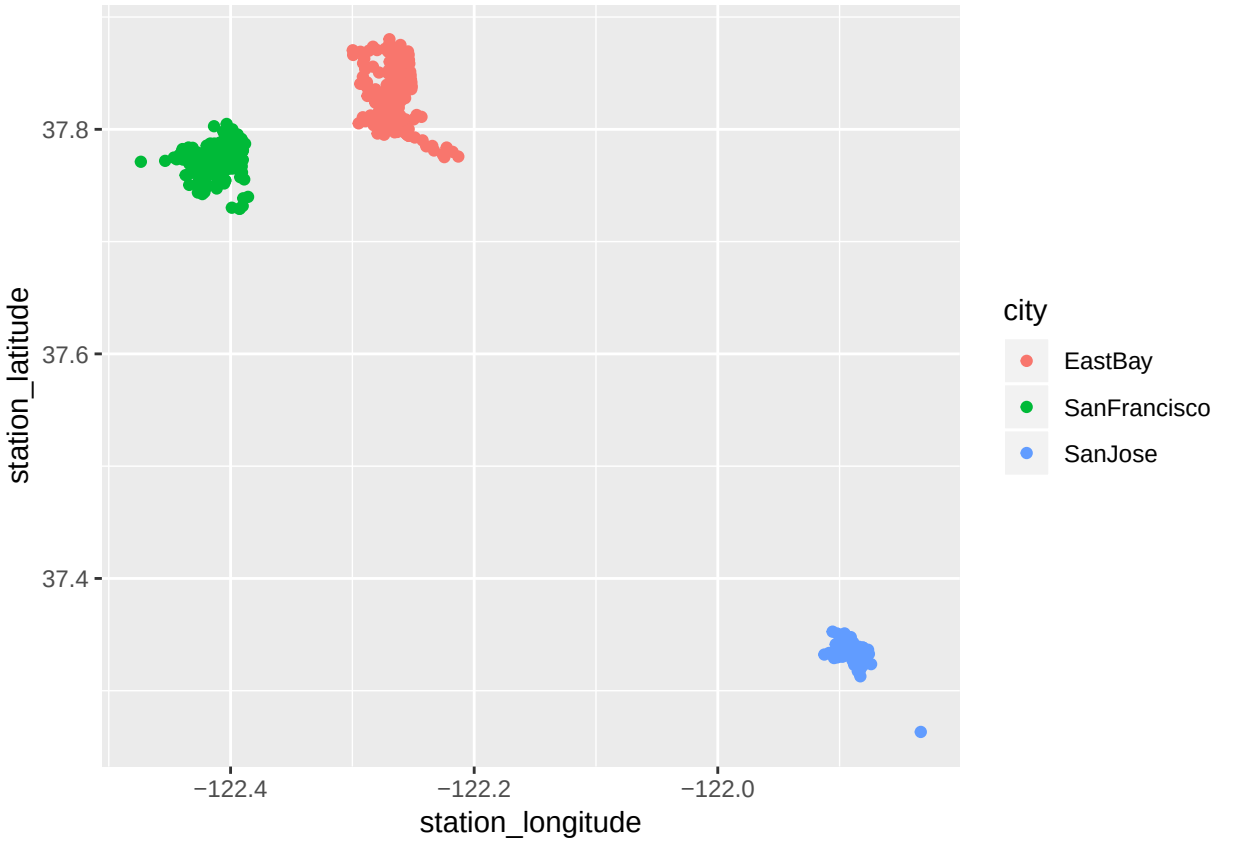
city	min_long	max_long
EastBay	-122.2997	-122.2130
SanFrancisco	-122.4737	-122.3857
SanJose	-121.9125	-121.8333

We can label each station with its city based on the longitude bin into which it falls:

```
label_city_by_long <- function(long) {
  if_else (long > -122, "SanJose",
           if_else (long < -122.3, "SanFrancisco",
                    "EastBay"))
}

# add city column to station_stats
station_stats <- station_stats %>%
  mutate(city = label_city_by_long(station_longitude)) %>%
  select(station_id:station_longitude, city, departure_count:prop_inflow)

ggplot(station_stats, aes(x = station_longitude, y = station_latitude, color = city)) + geom_point()
```




```
# success

rm(station_locations, hc_stations, stations_clustered, cluster_assigments, dist_stations)
```

Goal 3: identify stations attached to regional transit

For now, use text matching to find any station whose name contains at least one of the following keywords: * BART * train * station * ferry

```
transit <- unique(c(grep("BART", station_stats$station_name, value = TRUE),
                    grep("train", station_stats$station_name, ignore.case = TRUE, value = TRUE),
                    grep("station", station_stats$station_name, ignore.case = TRUE, value = TRUE),
                    grep("ferry", station_stats$station_name, ignore.case = TRUE, value = TRUE)))[-22]

station_stats <- station_stats %>%
  mutate(is_transit = station_name %in% transit)

rm(transit)
```

Goal 4: Add elevation to station_stats

Create a spatial points data frame from the lat/long coordinates for each station, then connect to the USGS Elevation Point Query Service. This takes some time (~2 seconds/station), so I save the output locally, then read it back in later (as long as I haven't added new data). The end product is the elevation, in meters, of each station.

```r
coord_df <- station_stats %>% select(station_longitude, station_latitude)
prj_dd <- "+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs"

# Create SpatialPoints
sp <- SpatialPoints(coord_df, proj4string = CRS(prj_dd))

# Create SpatialPointsDataFrame
spdf <- SpatialPointsDataFrame(sp, proj4string = CRS(prj_dd), data = station_stats)

# use USGS Elevation Point Query Service (slow, USA only)
spdf_elev_epqs <- get_elev_point(spdf, src = "epqs", units = "meters")
# spdf_elev_epqs # this took a while, so export the results for future use

readr::write_csv(as.data.frame(spdf_elev_epqs),
                 path = file.path("results", "station_elevation_df.csv"))
rm(coord_df, sp, spdf, spdf_elev_epqs, prj_dd)
```

```
## Parsed with column specification:
## cols(
##   station_id = col_double(),
##   n_per_id = col_double(),
##   station_name = col_character(),
##   station_latitude = col_double(),
##   station_longitude = col_double(),
##   city = col_character(),
##   departure_count = col_double(),
##   arrival_count = col_double(),
##   net_change = col_double(),
##   prop_inflow = col_double(),
##   is_transit = col_logical(),
##   elevation = col_double(),
##   elev_units = col_character(),
##   station_longitude.1 = col_double(),
##   station_latitude.1 = col_double()
## )
```

Add elevation to station info

```r
# align rows of the two data frames, just in case
spdf_elev_epqs %<>% arrange(station_id, station_name)
station_stats %<>% arrange(station_id, station_name)

# add it in
station_stats$elevation <- spdf_elev_epqs$elevation

rm(spdf_elev_epqs)
```

Goal 5: Reincorporate select station stats into primary data frame

Append elevation & transit linkage status to the start and end stations of each ride in 'data_clean'

```r
data_clean <- data_clean %>%
  arrange(start_time) %>%
  left_join(station_stats, by = c("start_station_name" = "station_name")) %>%
  mutate(start_station_city = city,
```

```
        start_station_is_transit = is_transit,
        start_station_elevation = elevation) %>%
  select(start_time:start_station_longitude, start_station_city:start_station_elevation,
        end_station_id:end_station_longitude, duration_sec, bike_id:is_weekend) %>%
  left_join(station_stats, by = c("end_station_name" = "station_name")) %>%
  mutate(end_station_city = city,
        end_station_is_transit = is_transit,
        end_station_elevation = elevation) %>%
  select(start_time, start_station_id:start_station_elevation,
        end_time, end_station_id:end_station_longitude, end_station_city:end_station_elevation,
        duration_sec:is_weekend) %>%
  mutate(elev_change = end_station_elevation - start_station_elevation)

data_clean %>% head(n = 10) %>% kable()
```

start_time	start_station_id	start_station_name	start_station_latitude
2017-06-28 09:47:36	21	Montgomery St BART Station (Market St at 2nd St)	37.78963
2017-06-28 09:47:41	58	Market St at 10th St	37.77662
2017-06-28 09:49:46	25	Howard St at 2nd St	37.78752
2017-06-28 09:50:59	81	Berry St at 4th St	37.77588
2017-06-28 09:56:39	66	3rd St at Townsend St	37.77874
2017-06-28 09:56:55	15	San Francisco Ferry Building (Harry Bridges Plaza)	37.79539
2017-06-28 09:58:33	23	The Embarcadero at Steuart St	37.79146
2017-06-28 10:00:54	81	Berry St at 4th St	37.77588
2017-06-28 10:00:59	66	3rd St at Townsend St	37.77874
2017-06-28 10:09:06	15	San Francisco Ferry Building (Harry Bridges Plaza)	37.79539