Filter the station data

Georgie Knight

23 August, 2016

The first problem to deal with is that the "status data" file is quite big. It contains a row of data for every minute. We will trim this data by selecting only the rows where there is a *change* in the data. That is a bike is added, taken away or the number of docks changes.

Dplyr and tidyr

Load the *dplyr* and *tidyr* packages which will help us wrangle the data:

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library("tidyr")
```

Loading up the data

Load the file for (1/9/14 - 31-8/15) into R:

```
status_data_original <-
  read.csv("C:/Users/Georgie/Desktop/BABS_Data/201508_status_data.csv")
status_data_frame <- data.frame(status_data_original)
status <- dplyr::tbl_df(status_data_frame)
```

The following will use the *lag* function to define a new column which checks if there has been a change or not and store this as a truth variable. We'll then filter using this column. We firstly create the new columns:

```
status  <-
  mutate(status,
         ch = (station_id == lag(station_id,1) &
               bikes_available == lag(bikes_available,1) &
               docks_available == lag(docks_available,1)))
```

Note that the first entry will be equal to NA so we change this to FALSE.

```
status$ch[1] = FALSE
```

Then we filter

```
status <- status %>% filter(ch == FALSE) %>% select(-ch)
```

Now we take a look and save the file for future use

```
## Observations: 1,135,974
## Variables: 4
## $ station_id      <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2...
## $ bikes_available <int> 15, 14, 15, 14, 15, 14, 15, 14, 15, 14, 15, 14...
## $ docks_available <int> 12, 13, 12, 13, 12, 13, 12, 13, 12, 13, 12, 13...
## $ time            <fctr> 2014-09-01 00:00:03, 2014-09-01 02:57:02, 201...
```

We've gone from nearly 37 million observations to just over 1.1 million without losing any information.