Lab 8

Enter Your Name and UNI Here

December __, 2016

Instructions

Before you leave lab today make sure that you upload a .pdf file to the canvas page (this should have a .pdf extension). This should be the PDF output after you have knitted the file, we don't need the .Rmd file (don't upload the one with the .Rmd extension). The file you upload to the Canvas page should be updated with commands you provide to answer each of the questions below. You can edit this file directly to produce your final solutions. Note, however, in the file you upload you should the above header to have the date, your name, and your UNI. Similarly, when you save the file you should replace **UNI** with your actualy UNI.

Introduction

Today we use the same data as last week's lab, recording the 2829 fastest men's and women's 100m sprint times.

1. Load the dataframes `sprint.m.fastest.csv` and `sprint.w.fastest.csv` and save them as `sprint.m.fastest` and `sprint.w.fastest`. These are the data you created in last week's lab that only stores the fastest time from each track meet.

```
setwd("C:/Users/cheny/Desktop/study/statistical computing and intro to data science/lab/lab
 8")

sprint.w.fastest <- read.csv('sprint.w.fastest.csv',header = TRUE)
sprint.m.fastest <- read.csv('sprint.m.fastest.csv',header = TRUE)
```

2. We want to merge the dataframes `sprint.m.fastest` and `sprint.w.fastest` over rows that correspond to times recorded at the same track meet. First find the common track meets between the two data frames, i.e. the common entries in `CityDate`. Hint: Use `intersect()`. Call the result `common.meets`.

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.4.2
```

```
common.meets <- merge(sprint.m.fastest,sprint.w.fastest,by='CityDate',all = FALSE)
```

3. Now compute the rows of each dataframe that correspond to these common track meets. Hint: Use `which()` and `is.element()`. Call the results `ind.m` and `ind.w`. Both should have length 385.

```
ind.m <- which(is.element(common.meets$CityDate,sprint.m.fastest$CityDate))
length(ind.m)
```

```
## [1] 385
```

```
ind.w <- which(is.element(common.meets$CityDate,sprint.w.fastest$CityDate))
length(ind.w)
```

```
## [1] 385
```

4. Now create a new dataframe that merges the columns of `sprint.m.fastest` with `sprint.f.fastest` , but keeping only rows that correspond to common track meets (these are indexed by `ind.m` and `ind.f`). Call the result `sprint` and arrange it so that the dataframe only has three columns: `MensTime` , `WomensTime` , and `CityDate` (the common track meet). Display the first five rows. Check to verify that the first index in `ind.m` corresponds to the same race as the first index in `ind.w` , the same for the second index, and so on…

```
sprint <- merge(sprint.m.fastest,sprint.w.fastest,all = TRUE)
head(sprint[ind.m,])
```

```
##   X Rank  Time Wind                     Name Country Birthdate
## 1 1    1  9.58  0.9               Usain Bolt     JAM  21.08.86
## 2 1    1 10.49  0.0 Florence Griffith-Joyner     USA  21.12.59
## 3 2    2  9.63  1.5               Usain Bolt     JAM  21.08.86
## 4 2    2 10.61  1.2 Florence Griffith-Joyner     USA  21.12.59
## 5 3    3  9.69  0.0               Usain Bolt     JAM  21.08.86
## 6 3    3 10.62  1.0 Florence Griffith-Joyner     USA  21.12.59
##           City       Date                CityDate
## 1       Berlin 16.08.2009       Berlin 16.08.2009
## 2 Indianapolis 16.07.1988 Indianapolis 16.07.1988
## 3       London 05.08.2012       London 05.08.2012
## 4 Indianapolis 17.07.1988 Indianapolis 17.07.1988
## 5      Beijing 16.08.2008      Beijing 16.08.2008
## 6        Seoul 24.09.1988        Seoul 24.09.1988
```

5. Note that the previous merge could have been done with the `merge()` function. Can you get the same result using `merge()` ?

```
Sprint <- merge(sprint.m.fastest,sprint.w.fastest,all = FALSE)
```