

# Introduction

---

## Introduction

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. In this assignment one will analyze a self monitor dataset; the data is collected by an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day. **The objective of the assignment is to show how tool such as knitr and R-mardown would help analysts in creating reproducible data analysis research project.**

## About the Dataset

The variables included in this dataset are:

1. **steps:** Number of steps taking in a 5-minute interval (missing values are coded as NA)
2. **date:** The date on which the measurement was taken in YYYY-MM-DD format
3. **interval:** Identifier for the 5-minute interval at which the measurements were taken.

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

## Loading and preprocessing the data

The activity data set is read in memory using `read.table()`, with comma is a separator and convert to a data frame. Some preliminary observations on the dataset:

1. There are total of 17,568 observation; after removing the obseravations that contain NAs, there are 15,264 observations remained.

2. The observations spanned through 61 days; of which only 53 days that contains non-NAs observations.

```
dat <- data.frame(read.table
                   ("~/Documents/Coursera/ReproR/GitTest/RepData_PeerAssessment1/activity.csv",
                    , header=TRUE, sep=","))

## Removing NA's
df <- na.omit(dat);
nrow(df)

## [1] 15264

## Number of days spanned with NA's
tDays <- unique(dat$date)
length(tDays)

## [1] 61

## Number of days spanned without NA's
t <- unique(df$date)
length(t)

## [1] 53
```

## What is mean total number of steps taken per day?

In this section, the observations with missing values are ignored, or not included in the analysis.

1. Calculate the total number of steps taken per day. The rows in the dataset can be grouped into each day, then one can sum up the steps of the observed intervals of the day. To do this, one can use `aggregate()` function – aggregate data base on date, and apply summation on the steps, as in the below line of code:

```
totalStepsPerDay <- aggregate(x=df$steps, list(date=df$date), FUN=sum, na.rm=TRUE);
totalStepsPerDay

##           date      x
## 1 2012-10-02    126
## 2 2012-10-03 11352
```

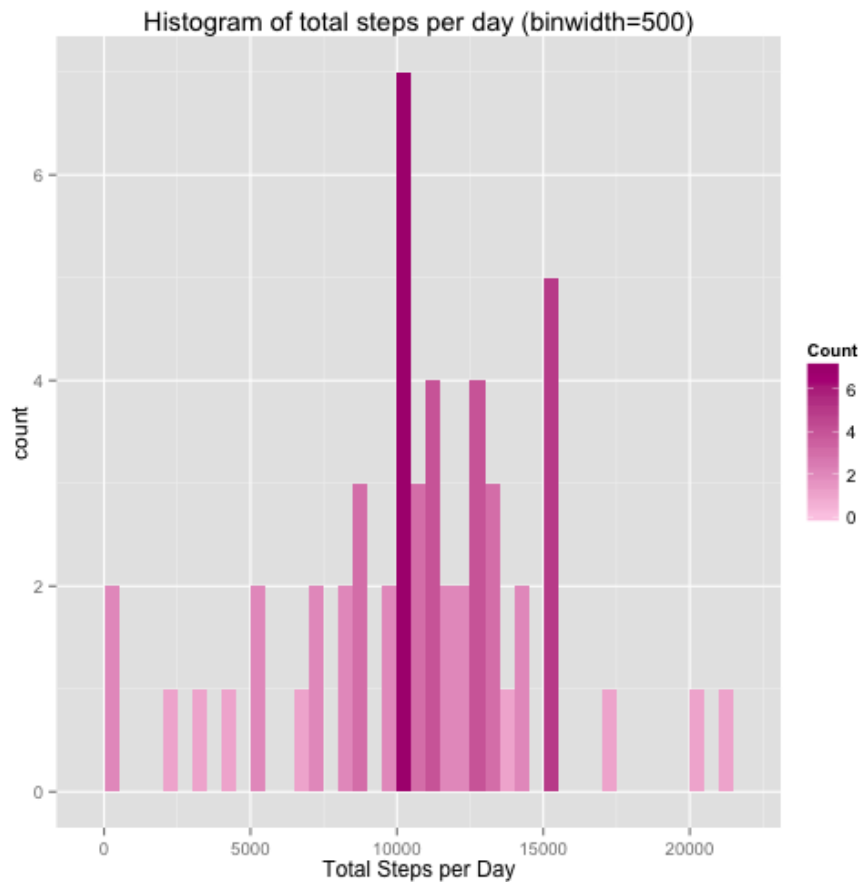
## 3 2012-10-04 12116  
## 4 2012-10-05 13294  
## 5 2012-10-06 15420  
## 6 2012-10-07 11015  
## 7 2012-10-09 12811  
## 8 2012-10-10 9900  
## 9 2012-10-11 10304  
## 10 2012-10-12 17382  
## 11 2012-10-13 12426  
## 12 2012-10-14 15098  
## 13 2012-10-15 10139  
## 14 2012-10-16 15084  
## 15 2012-10-17 13452  
## 16 2012-10-18 10056  
## 17 2012-10-19 11829  
## 18 2012-10-20 10395  
## 19 2012-10-21 8821  
## 20 2012-10-22 13460  
## 21 2012-10-23 8918  
## 22 2012-10-24 8355  
## 23 2012-10-25 2492  
## 24 2012-10-26 6778  
## 25 2012-10-27 10119  
## 26 2012-10-28 11458  
## 27 2012-10-29 5018  
## 28 2012-10-30 9819  
## 29 2012-10-31 15414  
## 30 2012-11-02 10600  
## 31 2012-11-03 10571  
## 32 2012-11-05 10439  
## 33 2012-11-06 8334  
## 34 2012-11-07 12883  
## 35 2012-11-08 3219  
## 36 2012-11-11 12608  
## 37 2012-11-12 10765  
## 38 2012-11-13 7336  
## 39 2012-11-15 41  
## 40 2012-11-16 5441  
## 41 2012-11-17 14339  
## 42 2012-11-18 15110  
## 43 2012-11-19 8841  
## 44 2012-11-20 4472  
## 45 2012-11-21 12787  
## 46 2012-11-22 20427  
## 47 2012-11-23 21194  
## 48 2012-11-24 14478

```
## 49 2012-11-25 11834
## 50 2012-11-26 11162
## 51 2012-11-27 13646
## 52 2012-11-28 10183
## 53 2012-11-29 7047
```

1. Histogram to show totals steps taken each day

```
library (ggplot2)

h <-ggplot(data=totalStepsPerDay, aes(x=x))+
  ggtitle("Histogram of total steps per day (binwidth=500)") +
  xlab("Total Steps per Day")
h + geom_histogram(binwidth=500, aes(fill = ..count..)) +
  scale_fill_gradient("Count", low="#fccde5", high="#ae017e")
```



1. Calculate and report the mean and median of the total steps taken per day

```
mu <- mean (totalStepsPerDay$x)
mu

## [1] 10766.19

med <- median (totalStepsPerDay$x)
med

## [1] 10765
```

## What is the average daily activity pattern?

1. Plotting the average steps taken per interval identifier, across all observed days

The interval identifier is a label for a five-minute interval during the date. The dataset will be grouped using the interval identifiers for all observed days; then the average steps is calculated for each group. The code is below.

```
aveStepsPerInterval <- aggregate(x=df$steps,list(Interval=df$interval), FUN=mean, na.rm=TRUE)
```

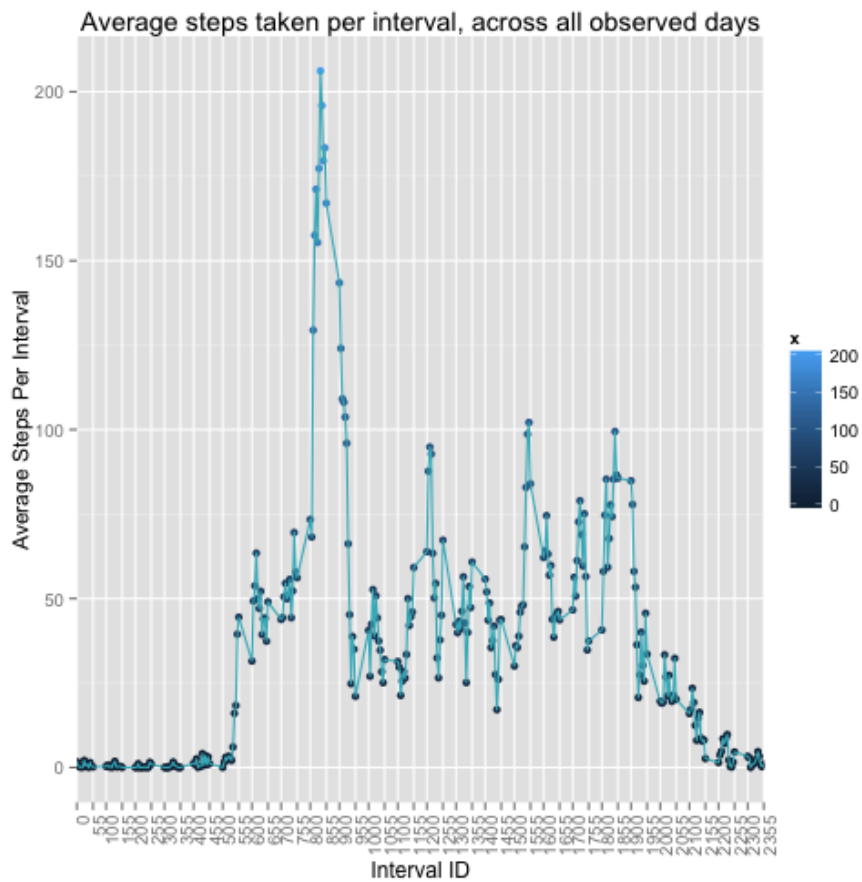
Below code will plot the average of steps per interval identifier, across the observed days.

```
library(scales)
x_breaks <- c(0,55,100,155,200,255,300,355,400, 455, 500, 555,
             600, 655, 700, 755, 800, 855, 900, 955, 1000, 1055,
             1100, 1155, 1200, 1255, 1300, 1355, 1400, 1455, 1500, 1555,
             1600, 1655, 1700, 1755, 1800, 1855, 1900, 1955, 2000, 2055,
             2100,2155,2200,2255,2300,2355)

avePlot <- ggplot(data=aveStepsPerInterval, aes(x = Interval, y = x, group=1)) +
  ggtitle("Average steps taken per interval, across all observed days") +
  xlab("Interval ID") +
  ylab("Average Steps Per Interval") +
  scale_x_discrete( breaks=x_breaks) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

avePlot <- avePlot +
  geom_point(aes(colour = x)) +
  geom_line(color="#41b6c4")

avePlot
```



1. Calculating the maximum average steps and identify the interval where the max is found

The maximum average steps is round up to be 206 steps, and it occurs at the time with interval identifier of 835.

```
mx <- round(max(aveStepsPerInterval$x));
mx
```

```
## [1] 206
```

```
indx <- which(aveStepsPerInterval$x==max(aveStepsPerInterval$x));
mxInterval <- aveStepsPerInterval$Interval[indx]
mxInterval
```

```
## [1] 835
```

**Imputing missing values**

**Are there differences in activity patterns between weekdays and weekends?**