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 monintor 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-mardow 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. *Each interval of the day is identified by a number; i.e. the identifier is same for 8:00 - 8:05 am every day of the year.*

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

I. 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 prelimiary 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
```

II. 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 – aggreage data base on date, and apply summaation 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
```

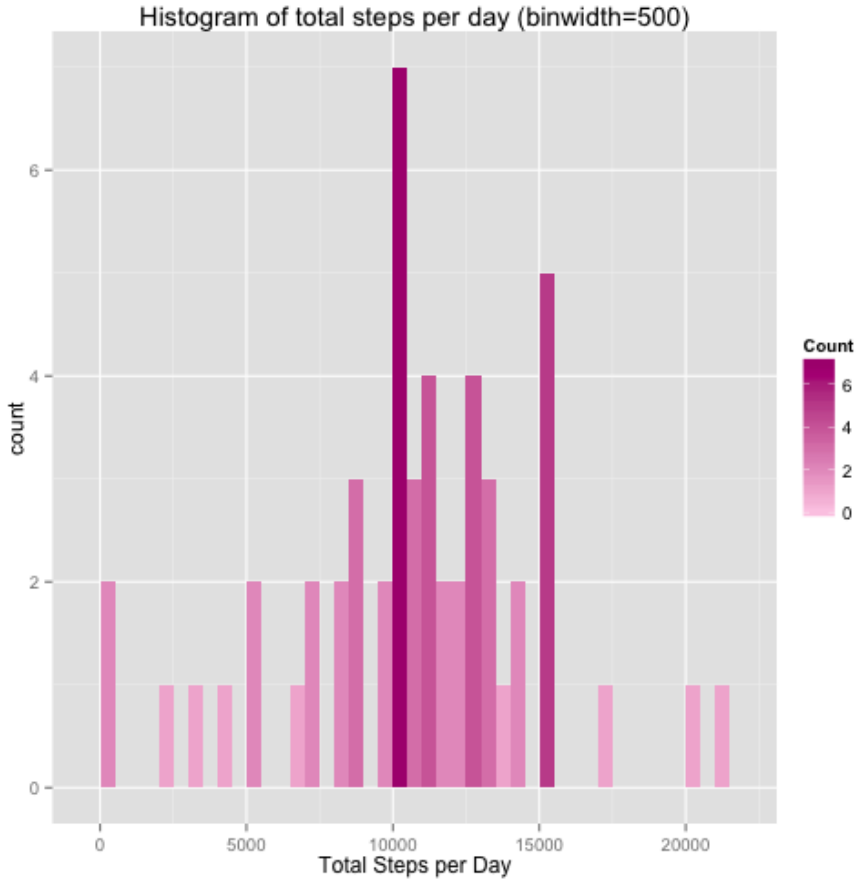
```
##          date     x
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## 11 2012-10-13 12426
## 12 2012-10-14 15098
## 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
```

III. What is the average daily activity pattern?

1. **Ploting 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 identifers for all observed days; then the averaage steps is calculated for each groups.

A note about the interval in the dataset– this is an *identifer* for a 5-minute interval, not the actual time of the day. For example 0-55 minutes of the day has the *interval identifier* of 0, 5, ..., 55, and the 60th-65th minute of the day has the identifer of 100. In order to plot the time series (continous time), an *xInterval* sequence is introduced–sequence of 0..1440, step by 5 minutes. In addition, in order to label the x-axis with time marks, a *x_breaks* variable is defined as a sequence of 0..1440, step by 30 minutes.

```
#Group data by its 5-minute interval identifer, across all days,
# and then calculate the average steps for the interval identifer
aveStepsPerInterval <-aggregate(x=df$steps,list(Interval=df$interval), FUN=mean, na.rm=TRUE)
## define a time equivalent of the interval identifier
xInterval <- seq(from=0, to=1435, by=5)
aveStepsPerInterval <- cbind(aveStepsPerInterval,xInterval)
```

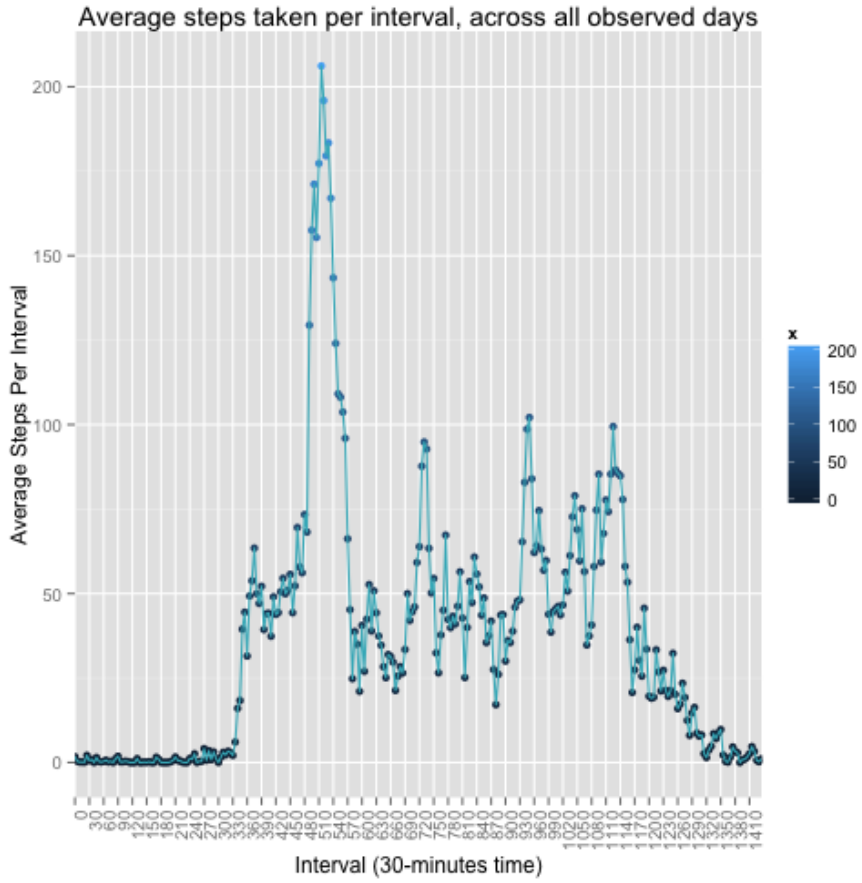
Below code will plot the average of steps per interval identifier, across the observer days.

```
library(scales)
x_breaks <- seq(from=0, to=1440, by=30);

avePlot <- ggplot(data=aveStepsPerInterval, aes(x = xInterval, y = x, group=1)) +
  ggtitle("Average steps taken per interval, across all observed days") +
  xlab("Interval (30-minutes time)") +
  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. **Calulating the maxinum average steps and identify the interval where the max is found**

The maxinum average steps is round up to be **206 steps**, and it occurs at **835 interval identifier or at 8:35 in the morning. **

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

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

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

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

IV. Imputing missing values

1. **Total number of missing values in the dataset**

The dataset **dat** read from *activity.csv* contains the NAs. The total number of rows with NAs is **2304**.

```
ind <- which(is.na(dat))
length(ind)
```

```
## [1] 2304
```

1. **Strategy on choosing number of steps that contain NAs** Using the mean of the average of steps for each interval. This number is calcuated to be 37 steps.

```
replacedBy <- round(mean(aveStepsPerInterval$x))
replacedBy
```

```
## [1] 37
```

1. **Replace all the NA's in the dataset** with *devised new* number of steps; the next line is to check that there is no more NAs in the dataset.

```
dat$steps[which(is.na(dat))] <- replacedBy
length(which(is.na(dat)))
```

```
## [1] 0
```

1. **Recalculate the total number of steps taken each day, its mean, median , and plot new histogram **

```
totalStepsPerDayNew <-aggregate(x=dat$steps,list(date=dat$date), FUN=sum);
muNew <- mean (totalStepsPerDayNew$x)
muNew
```

```
## [1] 10751.74
```

```
medNew <- median (totalStepsPerDayNew$x)
medNew
```

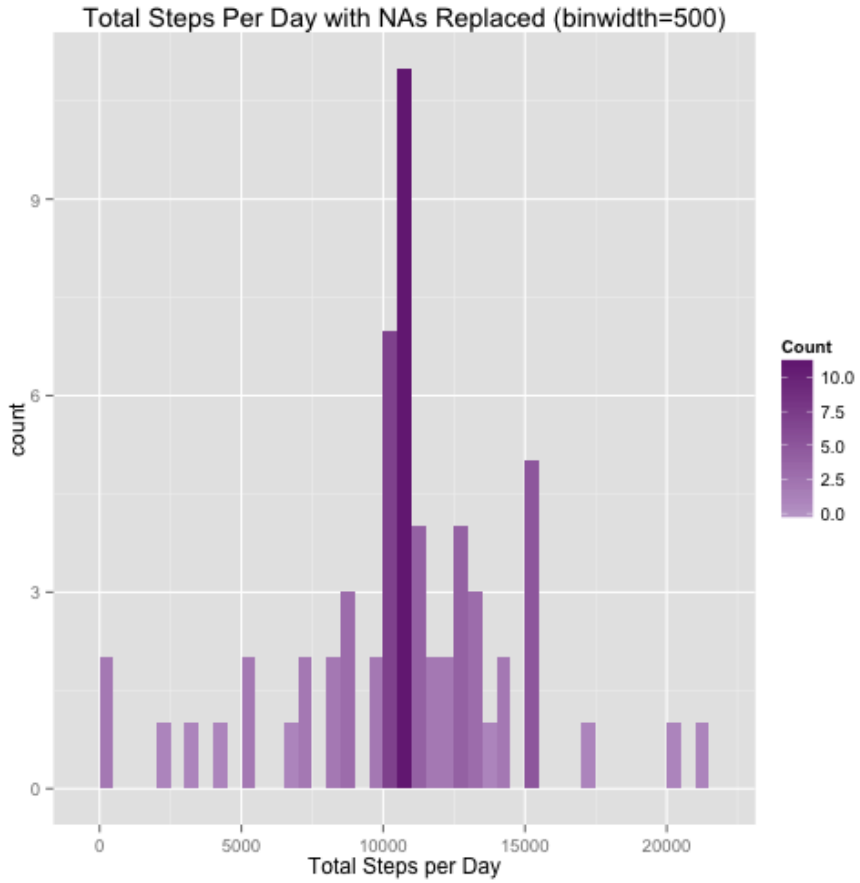
```
## [1] 10656
```

Following table show the adjusted mean and median of the dataset with replaced NAs, as compared to those of the dataset with NAs are being removed.

Measure	With NA replaced	Removed NAs
mean (steps)	10751.74	10766.19
median (steps)	10656	10765

Conclusion:

1. *the mean and median are different when NAs are replaced*
2. Since the steps is non-negative, if the steps are replaced with positive numbers, the mean and median will be higher. If the NAs are replaced with zeroes, the mean and median will be smaller.



V. Are there differences in activity patterns between weekdays and weekends?

In the last section the dataset *dat* has its NAs filled-in; the stragey in this particular exercise, a mean of all the average steps per interval was chosen to be the filled in number. This dataset *dat* will be using in this section for calculation.

1. **Create a new factor Weekday and Weekend** apply the weekdays() function tho the date column in derivign "Weekday" and "Weekend" factor, as in the following code.

```
dat$dayOfWeek <- as.factor(ifelse(weekdays(as.Date(dat$date)) %in% c("Saturday", "Sunday"),

head(dat)
```

```
##   steps       date interval dayOfWeek
## 1    37 2012-10-01        0   Weekday
## 2    37 2012-10-01        5   Weekday
## 3    37 2012-10-01       10   Weekday
## 4    37 2012-10-01       15   Weekday
## 5    37 2012-10-01       20   Weekday
## 6    37 2012-10-01       25   Weekday
```

1. **Separate out observations for the Weekdays vs Weekends**

```
wkdayDat <- subset(dat, dat$dayOfWeek=="Weekday")
wkendDat <- subset(dat, dat$dayOfWeek=="Weekend")
```

1. **Calculate average steps per interval for the Weekday and Weekend groups or subsets**

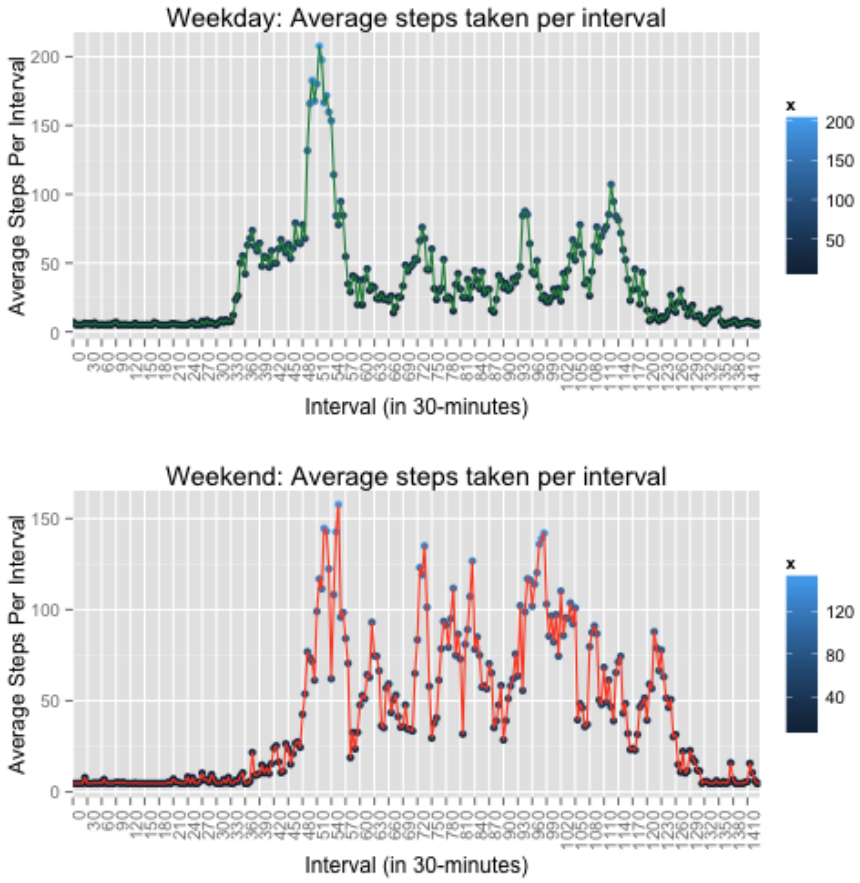
Similar in part 3, a real time sequence *xInterval* is bound into the data frame to plot a time series.

```
aveStepsPerIntervalWkday <-aggregate(x=wkdayDat$steps,list(Interval=wkdayDat$interval), FUN=
aveStepsPerIntervalWkend <-aggregate(x=wkendDat$steps,list(Interval=wkendDat$interval), FUN=

# Append a time interval to the data frame to graph the time series
aveStepsPerIntervalWkday <- cbind(aveStepsPerIntervalWkday,xInterval);
aveStepsPerIntervalWkend <- cbind(aveStepsPerIntervalWkend,xInterval);
```

1. **Time series plots of activity of Weekdays vs Weekends**

```
library(grid)
grid.newpage()
grid.draw(rbind(ggplotGrob(avePlotWkday), ggplotGrob(avePlotWkend), size = "last"))
```



Observation: Referencing the above figures, it is observed that there are more activities on the weekend than that of weekdays.

Since the *original dataset has its NAs replaced with an positive number of steps.* The replacement is an average steps of the mean per interval. **The filled-in dataset is used in this analysis.** Therefore, the above observation may not be true. One true thing is that in the original dataset, most NAs are found in the weekends. For such reason, it can be true that there are less activities during the weekend.