Reproducible Research - Week 2, Course Project

by NA

7 July 2016

This is a R markdown document for the course project 1 of *Reproducible Research* course. This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from 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 variables included in this dataset are:

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

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

Loading and pre-proceessing the data

```r
act <- read.csv('activity.csv')
summary(act)
```

```
##      steps                date          interval
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8
##  Median :  0.00   2012-10-03:  288   Median :1177.5
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840
```

```r
head(act)
```

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

```r
str(act)
```

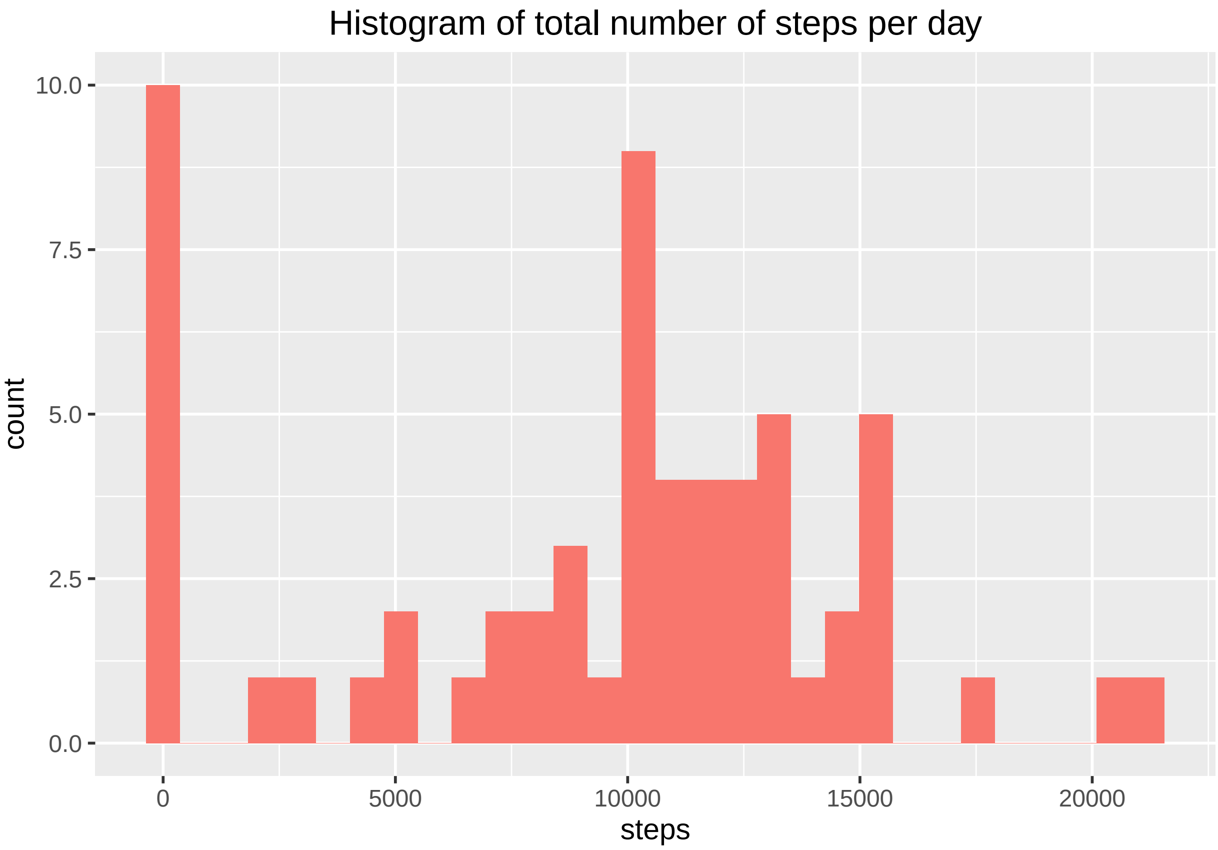
```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
library(dplyr)
act_by_date <- act %>% group_by(date)%>% summarise_each(funs(sum(.,na.rm=TRUE)))
act_by_date
```

```
## Source: local data frame [61 x 3]
##
##          date steps interval
##        <fctr> <int>    <int>
## 1  2012-10-01     0   339120
## 2  2012-10-02   126   339120
## 3  2012-10-03 11352   339120
## 4  2012-10-04 12116   339120
## 5  2012-10-05 13294   339120
## 6  2012-10-06 15420   339120
## 7  2012-10-07 11015   339120
## 8  2012-10-08     0   339120
## 9  2012-10-09 12811   339120
## 10 2012-10-10  9900   339120
## ..        ...   ...      ...
```

Histogram of total number of steps per day

```
library(ggplot2)
ggplot(act_by_date, aes(steps, fill = 'magenta'))+geom_histogram()+theme(legend.position = 'none')+ggti
```



Mean and Median number of steps taken each day

```
mean <- mean(act_by_date$steps, na.rm = TRUE)
median <- median(act_by_date$steps, na.rm = TRUE)
cat(paste('The mean number of steps taken each day is',mean, sep=' '))
```

```
## The mean number of steps taken each day is 9354.22950819672
```

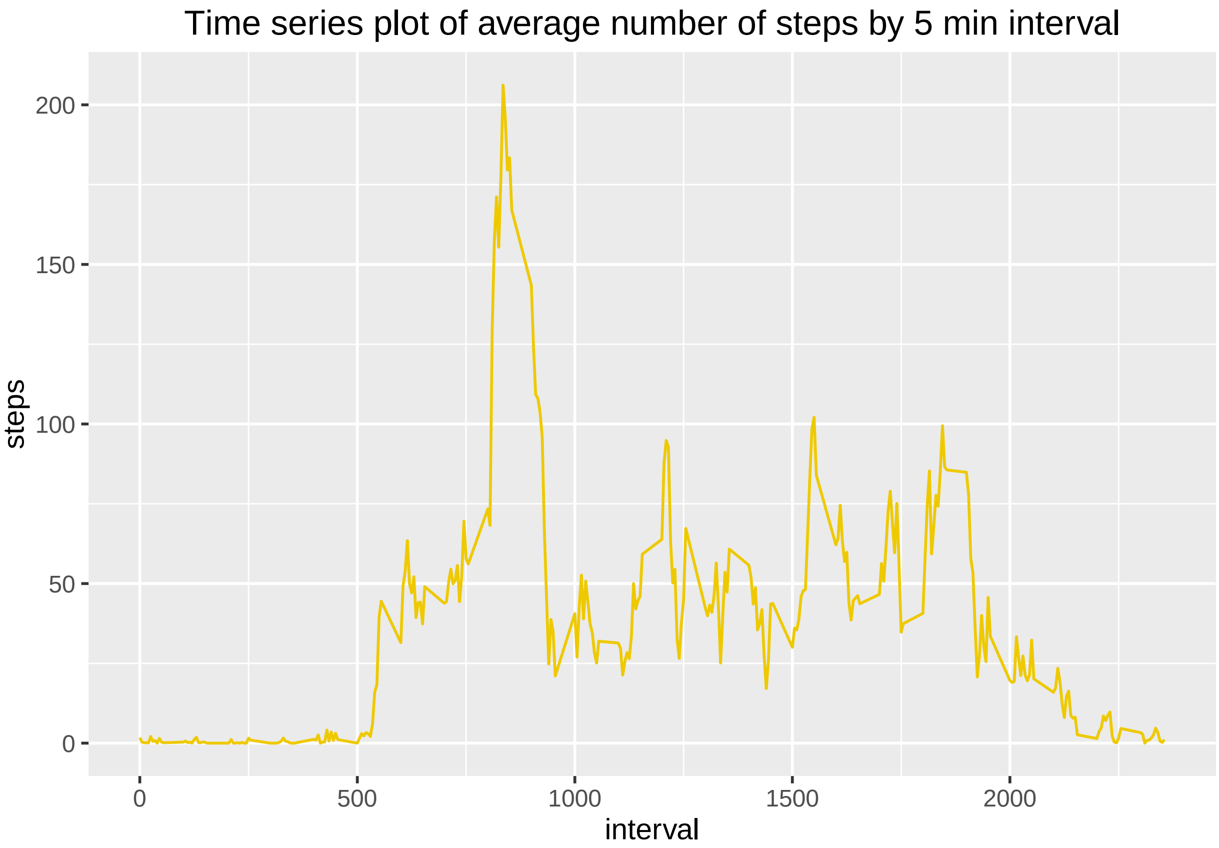
```
cat(paste('The median number of steps taken each day is',median, sep=' '))
```

```
## The median number of steps taken each day is 10395
```

Time series plot of average number of steps

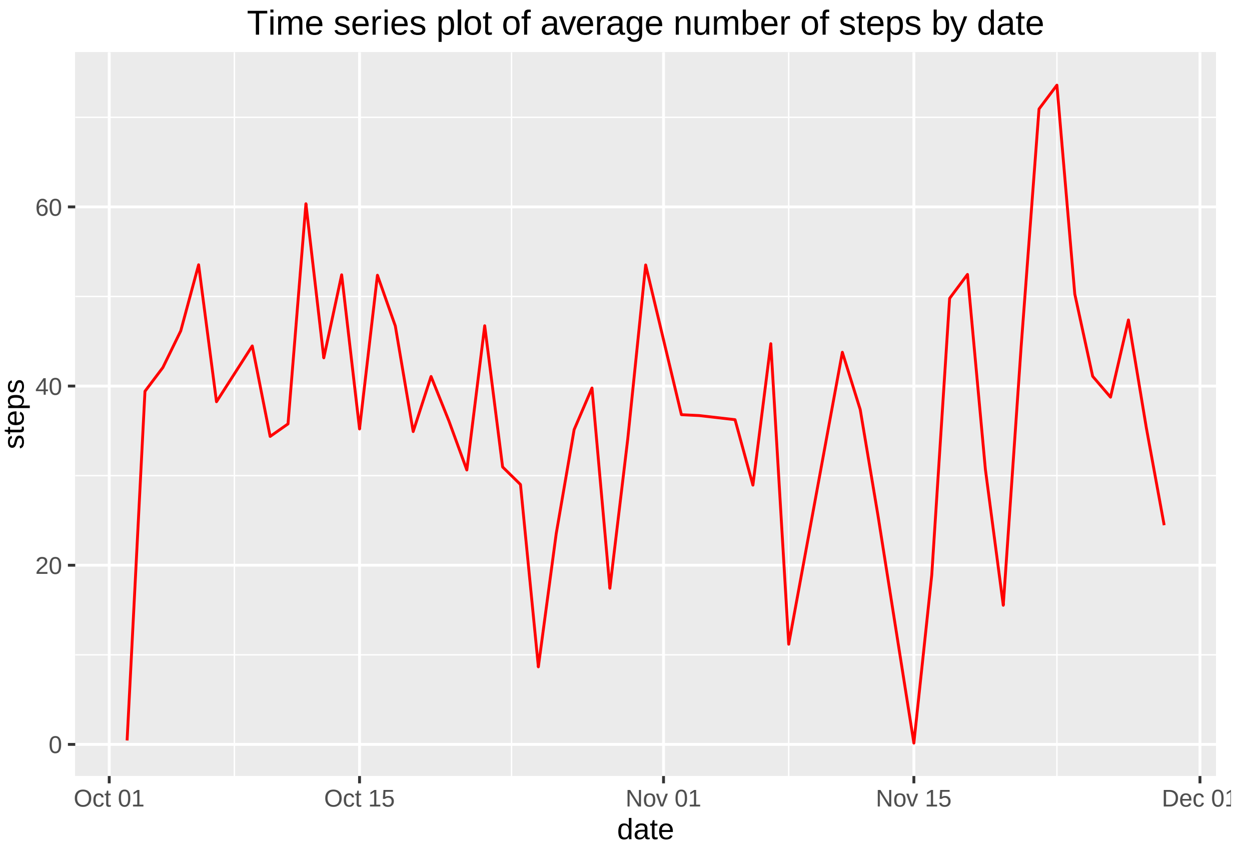
```
steps_by_interval <- act %>% group_by(interval)%>% summarise_each(funs(mean(.,na.rm=TRUE)))

ggplot(data = steps_by_interval, aes(x=interval, y = steps))+geom_line(color = 'gold2')+ggtitle('Time s
```



```
steps_by_date <- act %>% group_by(date)%>% summarise_each(funs(mean(.,na.rm=TRUE)))

library(lubridate)
steps_by_date$date <- ymd(steps_by_date$date)
ggplot(data = na.omit(steps_by_date), aes(x = date, y = steps))+geom_line(color = 'red')+ggtitle('Time
```



The 5-minute interval that, on average, contains the maximum number of steps

```
steps_by_interval[which.max(steps_by_interval$steps),]
```

```
## Source: local data frame [1 x 3]
##
##   interval    steps  date
##      <int>    <dbl> <lgl>
## 1      835 206.1698    NA
```

The interval 835 has the maximal number of steps (206).

Imputation of missing data

```
sum(is.na(act))
```

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

```
steps_by_interval
```

```
## Source: local data frame [288 x 3]
```

```
##
##    interval     steps  date
##       <int>     <dbl> <lgl>
## 1         0 1.7169811    NA
## 2         5 0.3396226    NA
## 3        10 0.1320755    NA
## 4        15 0.1509434    NA
## 5        20 0.0754717    NA
## 6        25 2.0943396    NA
## 7        30 0.5283019    NA
## 8        35 0.8679245    NA
## 9        40 0.0000000    NA
## 10       45 1.4716981    NA
## ..      ...       ...   ...
```

The total number of missing rows is 2304.

Strategy for imputation - missing values (NAs) to be replaced with the average value of 5-min interval

```
act_new <- act
NAs <- is.na(act_new$steps)
mean_interval <- tapply(act_new$steps, act_new$interval, mean, na.rm=TRUE, simplify=TRUE)
act_new$steps[NAs] <- mean_interval[as.character(act_new$interval[NAs])]
sum(is.na(act_new))
```

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

Missing values have been replaced and a new dataset(act_new) has been created.

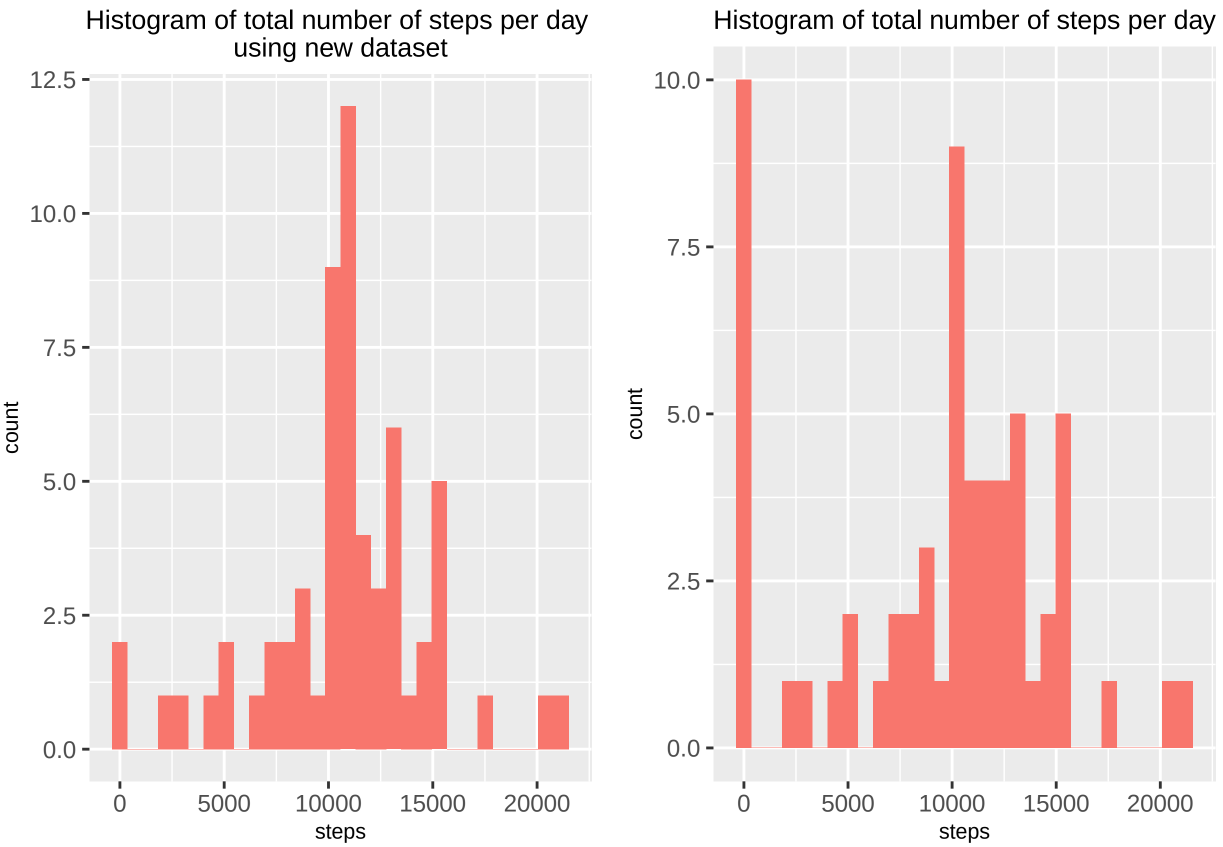
Tatal steps each day and mean/median total steps per day using new dataset

```
act_new_by_date <- act_new %>% group_by(date)%>% summarise_each(funs(sum(.,na.rm=TRUE)))

a<-ggplot(act_new_by_date, aes(steps, fill = 'magenta'))+geom_histogram()+theme(legend.position = 'none

b<-ggplot(act_by_date, aes(steps, fill = 'magenta'))+geom_histogram()+theme(legend.position = 'none')+gg
library(gridExtra)

grid.arrange(a,b, ncol = 2)
```



Mean and Median number of steps taken each day using new dataset

```
mean <- mean(act_new_by_date$steps, na.rm = TRUE)
median <- median(act_new_by_date$steps, na.rm = TRUE)
cat(paste('The mean number of steps taken each day is',mean, sep=' '))
```

```
## The mean number of steps taken each day is 10766.1886792453
```

```
cat(paste('The median number of steps taken each day is',median, sep=' '))
```

```
## The median number of steps taken each day is 10766.1886792453
```

After replacing NA values, the distribution of the data appears more gaussian and the mean and median became identical.

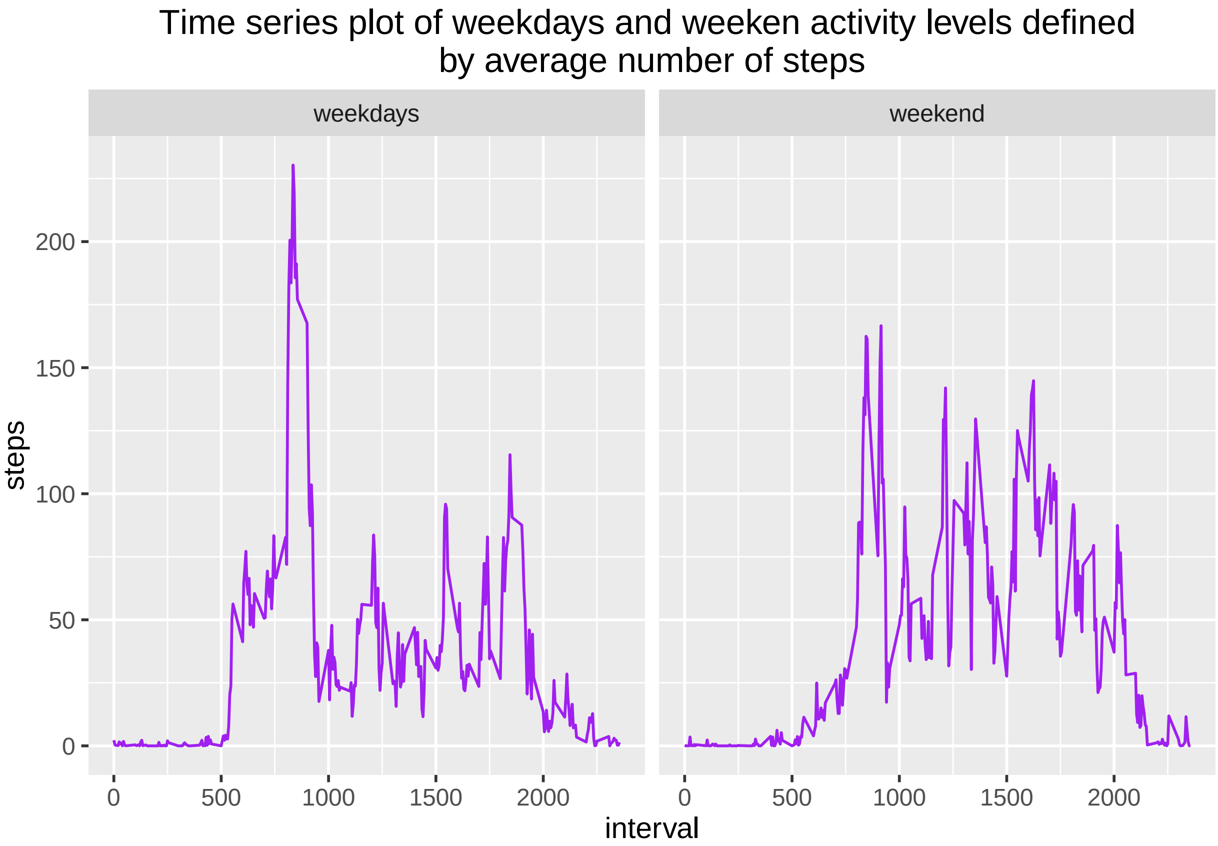
Activity patterns between weekends and weekdays

```
library(lubridate)
act_new$date <- ymd(act_new$date)
act_new$days <- weekdays(act_new$date)
act_new$wkdays <- ifelse(act_new$days == 'Saturday', 'weekend',
                         ifelse(act_new$days == 'Sunday', 'weekend',
                                'weekdays'))
act_new$wkdays <- as.factor(act_new$wkdays)
str(act_new)
```

```
## 'data.frame':    17568 obs. of  5 variables:
##  $ steps   : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ days    : chr  "Monday" "Monday" "Monday" "Monday" ...
##  $ wkdays  : Factor w/ 2 levels "weekdays","weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

```
steps_by_interval_new <- act_new %>% group_by(interval, wkdays)%>% summarise_each(funs(mean))

ggplot(data = steps_by_interval_new, aes(x=interval, y = steps))+geom_line(color = 'purple')+facet_wrap
```


Time series plot of weekdays and weeken activity levels defined
by average number of steps
weekdays
weekend
steps
interval