Reproducible Research: Peer Assessment 1

Abdul Rasheed Narejo

August 27, 2018

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. These type of devices are part of the "quantified self" movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

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 data for this assignment can be downloaded from the course web site:

Dataset: Activity monitoring data [52K] 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 and there are a total of 17,568 observations in this dataset.

Load required libraries

```r
library(dplyr) # load dplyr for data manipulation
library(ggthemes) # use themes to beautify graphs
library(ggplot2) # ggplot for data visualization
```

Loading and preprocessing the data

Load the data (read.csv())

```r
data <- read.csv("activity.csv")
summary(data$steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.00   37.38   12.00  806.00    2304
```

Process/transform the data (if necessary) into a format suitable for your analysis,

```r
# format date column as valid Date format
data$date <- as.Date(data$date)
```

```
# generate data summary
summary(data)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

back to top

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

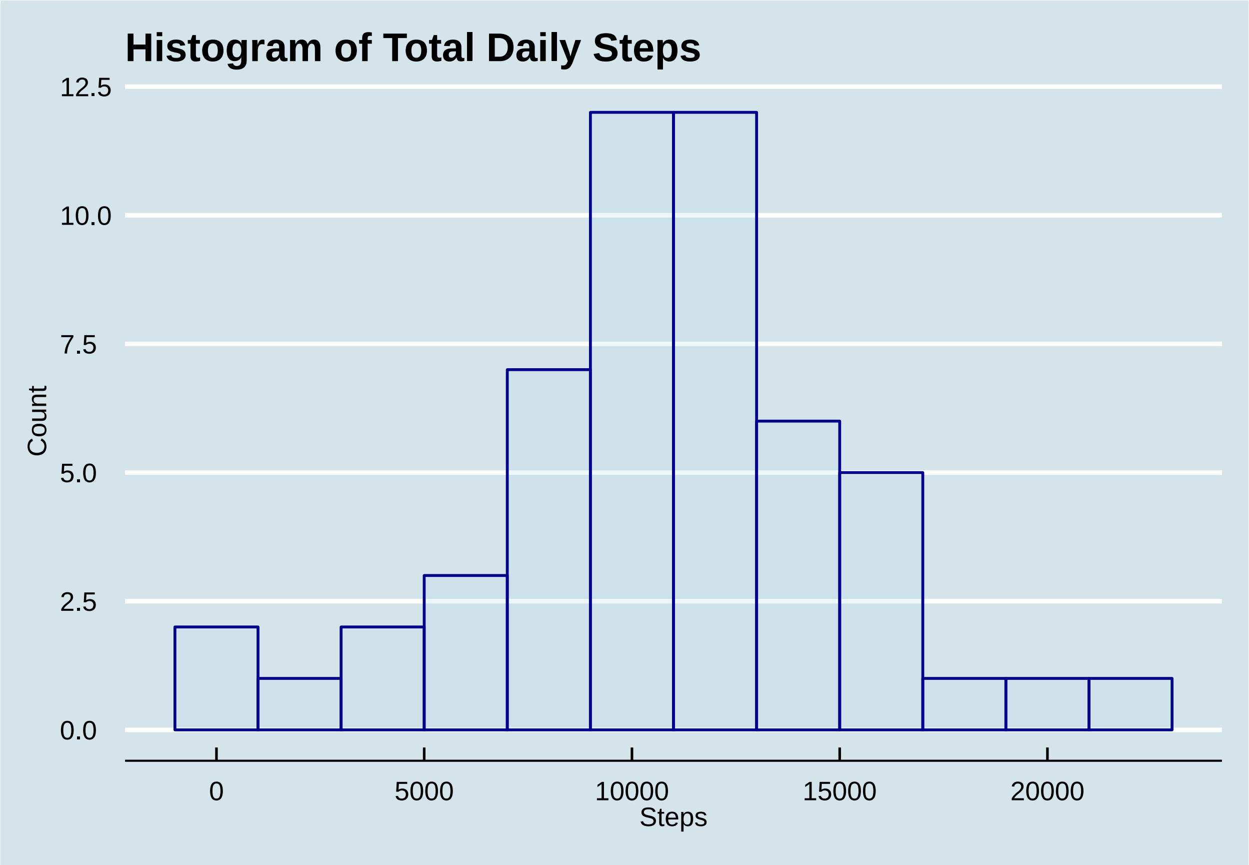
Calculate the total number of steps taken per day

```
# calculate total steps by each day
dailySteps <- data %>% group_by(date) %>% summarize(dailySteps=sum(steps))
summary(dailySteps)
```

```
##       date              dailySteps
##  Min.   :2012-10-01   Min.   :   41
##  1st Qu.:2012-10-16   1st Qu.: 8841
##  Median :2012-10-31   Median :10765
##  Mean   :2012-10-31   Mean   :10766
##  3rd Qu.:2012-11-15   3rd Qu.:13294
##  Max.   :2012-11-30   Max.   :21194
##                       NA's   :8
```

Make a histogram of the total number of steps taken each day

```
#hist(dailySteps1$dailySteps, breaks = 10)
ggplot(na.omit(dailySteps), aes(dailySteps)) +
    geom_histogram(binwidth = 2000,
                   col="darkblue",
                   fill="lightblue",
                   alpha = .2
                  ) +
    theme_economist() +
    labs(title="Histogram of Total Daily Steps") +
    labs(x="Steps", y="Count")
```



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

```
# calculate mean daily steps for all days
meanDailySteps <- round(mean(dailySteps$dailySteps, na.rm = TRUE))
meanDailySteps
```

```
## [1] 10766
```

NOTE: Mean daily steps are 10,766

```
# calculate median daily steps
medianDailySteps <- round(median(dailySteps$dailySteps, na.rm = TRUE))
medianDailySteps
```

```
## [1] 10765
```

NOTE: Meedian daily steps are 10,765

back to top

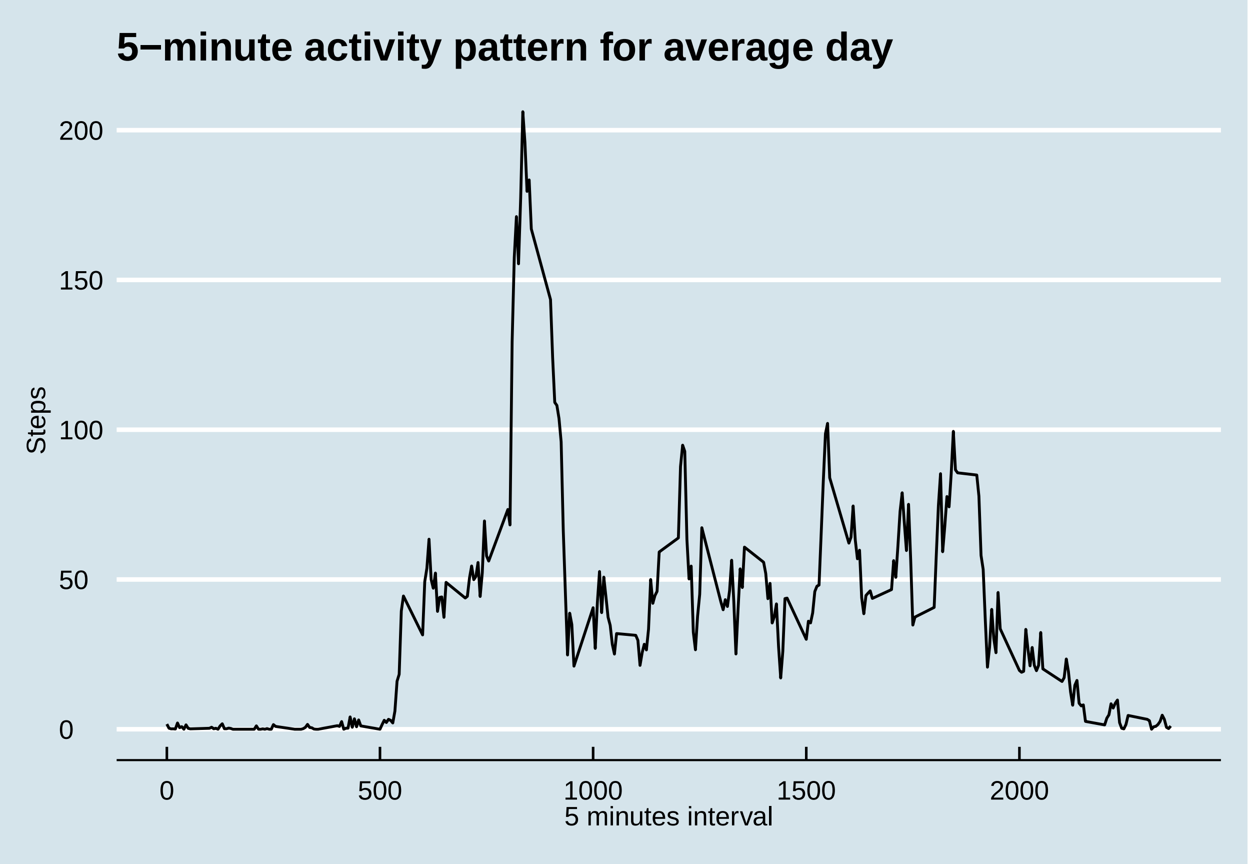
What is the average daily activity pattern?

Plot 5-minute interval and average number of steps taken

Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
# calculate average steps for every 5 minute interval during the day and save it as a new dataframe dai
dailyPattern <- data %>% group_by(interval) %>% summarize(meanActivity = mean(steps, na.rm = TRUE))

# plot average 5-minute activity trend using ggplot
ggplot(dailyPattern, aes(interval, meanActivity)) + geom_line() +
        theme_economist() +
        labs(title="5-minute activity pattern for average day") +
    labs(x="5 minutes interval", y="Steps")
```



Which 5-minute interval had maximum steps?

Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
maxStepsInterval <- dailyPattern[which.max(dailyPattern$meanActivity),]
maxStepsInterval
```

```
## # A tibble: 1 x 2
##   interval meanActivity
##      <int>        <dbl>
## 1      835         206.
```

Interval **835** had maximum average steps of **206**

Imputing missing values

Total Missing Values

Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
totalMissingValues <- sum(is.na(data$steps))
totalMissingValues
```

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

There are total 2304 number of total missing values

Stragety to fill missing values

Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

There is a fluctuation of activity based on the time of the day. Hence, for each missing value we can use average for same slot across all available values

Fill missing Values

Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
newData <- data %>%
             group_by(interval) %>%
             mutate(steps= ifelse(is.na(steps), mean(steps, na.rm=TRUE), steps))

# check for missing values in new DataFrame
sum(is.na(newData$steps))
```

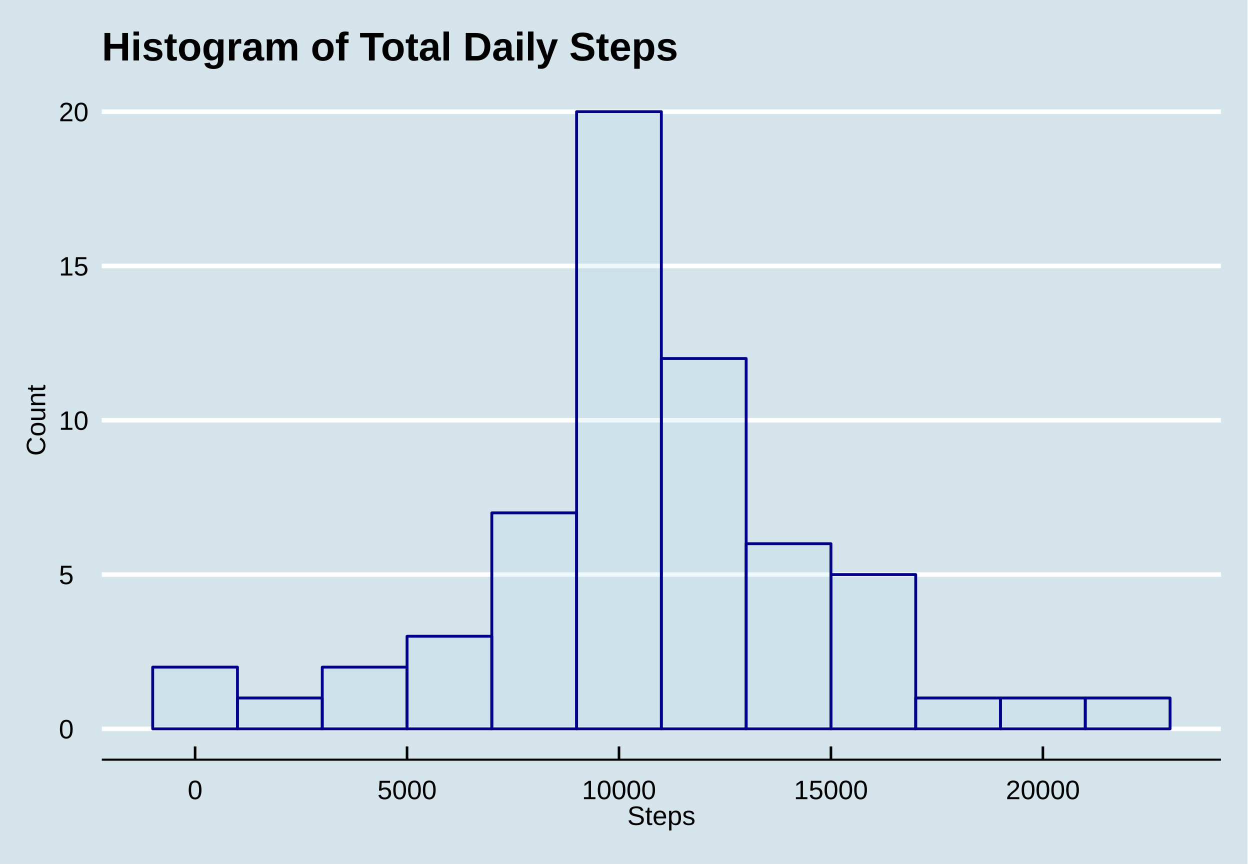
```
## [1] 0
```

Histogram of total steps each day, calculate mean and median

Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

```
# calculate total steps by each day
dailyStepsRevised <- newData %>% group_by(date) %>% summarize(dailySteps=sum(steps))
```

```
# generate histogram plot
ggplot(dailyStepsRevised, aes(dailySteps)) +
    geom_histogram(binwidth = 2000,
                   col="darkblue",
                    fill="lightblue",
                    alpha = .2
                   ) +
    theme_economist() +
    labs(title="Histogram of Total Daily Steps") +
    labs(x="Steps", y="Count")
```



```
# calculate mean daily steps for all days
meanDailyStepsRevised <- mean(dailyStepsRevised$dailySteps, na.rm = TRUE)
```

NOTE: Mean daily steps are 10,766.19 '

```
# calculate median daily steps
medianDailyStepsRevised <- median(dailyStepsRevised$dailySteps, na.rm = TRUE)
```

NOTE: Mean daily steps are 10,766.19

back to top

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

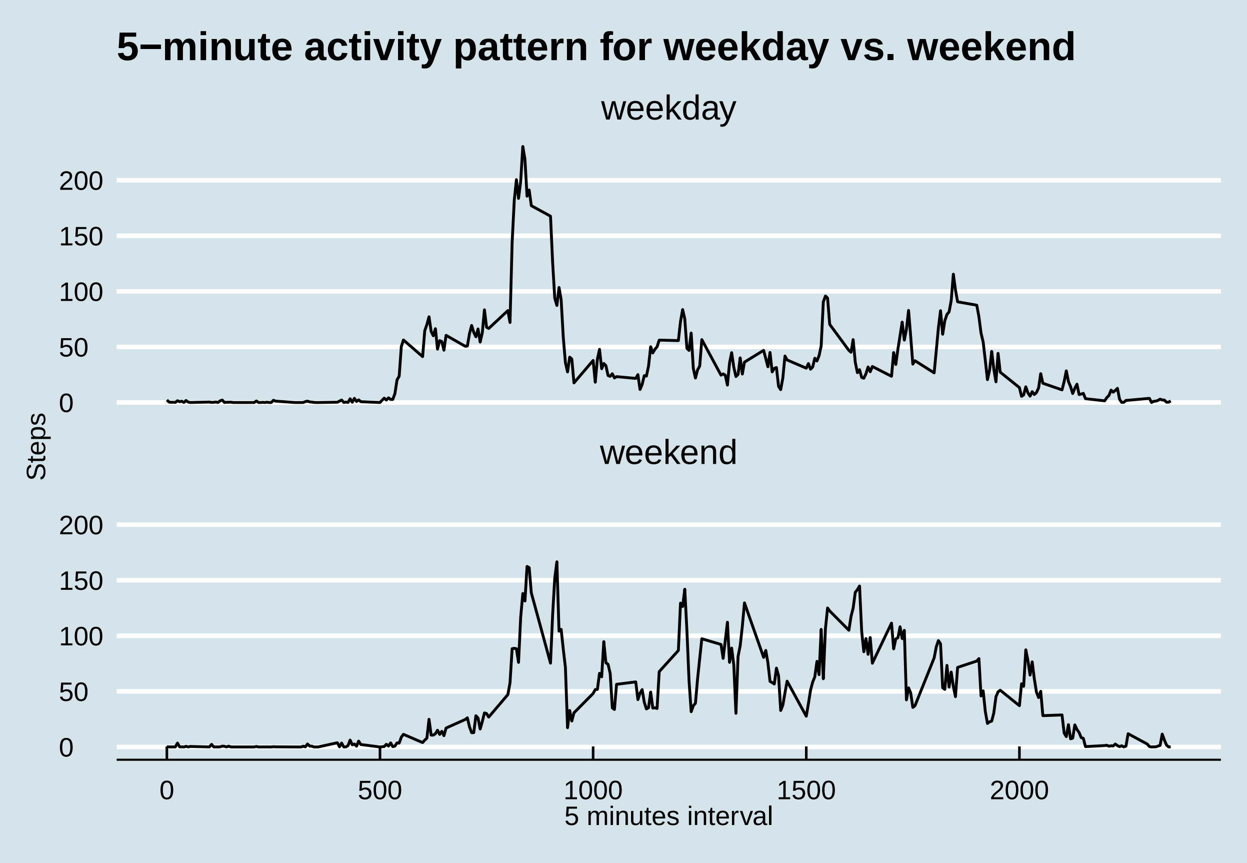
Make a panel plot containing a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
newData$dayOfWeek = "weekday"
newData[(weekdays(newData$date) %in% c("Saturday", "Sunday")),]$dayOfWeek = "weekend"
newData$dayOfWeek <- as.factor(newData$dayOfWeek)
table(newData$dayOfWeek)
```

```
##
## weekday weekend
##   12960    4608
```

```
weeklyData <- newData %>% group_by(dayOfWeek, interval) %>% summarize(meanActivity = mean(steps, na.rm
```

```
ggplot(weeklyData, aes(interval, meanActivity)) +
    geom_line() +
    facet_wrap(~dayOfWeek, ncol=1) +
    theme_economist() +
    labs(title="5-minute activity pattern for weekday vs. weekend") +
    labs(x="5 minutes interval", y="Steps")
```


5−minute activity pattern for weekday vs. weekend
weekday
Steps
weekend
5 minutes interval