Reproducible Research Peer Graded Project 1

Jason Carlson

3-September-2020

==========

R Markdown

This is an R Markdown document containing the code and the output associated with the peer graded assignment for the Reproducible Research course, produced by Johns Hopkins, and delivered through the Coursera platform. This markdown corresponds to Project 1 of the course.

At the start of this project, I forked/cloned the GitHub repository created for this assignment, which was located at the following URL: http://github.com/rdpeng/RepData_PeerAssessment1. The GitHub repository contains the dataset for the assignment.

Description of the dataset

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 includes 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 and there are a total of 17,568 observations in this dataset.

Loading and inspecting the data

- Using the read.csv command to assign the dataset to a variable called activity

```
activity <- read.csv("./activity.csv", header = TRUE, na.strings="NA")
```

- Viewing a summary of the data

```
summary(activity)
```

```
##      steps            date              interval     
##  Min.   :  0.00   Length:17568       Min.   :   0.0  
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8  
##  Median :  0.00   Mode  :character   Median :1177.5  
##  Mean   : 37.38                      Mean   :1177.5  
##  3rd Qu.: 12.00                      3rd Qu.:1766.2  
##  Max.   :806.00                      Max.   :2355.0  
##  NA's   :2304
```

- Viewing the header rows of the data

```
head(activity)
```

```
##   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
```

Pre-processing the data into a format suitable for the analysis

- Determining that the class of the date column is character:

```
class(activity$date)
```

```
## [1] "character"
```

- Converting the date column to the class of date

```
activity$date <- as.Date(activity$date, format="%Y-%m-%d")
```

- Confirming that the date column is now of class date

```
class(activity$date)
```

```
## [1] "Date"
```

- Create a new version of the dataset that has the NAs removed

```
activitysteps <- activity[!is.na(activity$step),]
```

- Confirm that there are no NAs in the new dataset

```
sum(is.na(activitysteps$step))
```

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

What is mean total number of steps taken per day?

- Calculate the total number of steps taken per day, by agreggating steps for each date, and including a sum, assigning this to a new dataset containing 53 observations corresponding to the unique days in the dataset.

```
steps_per_day <-aggregate(steps~date,data=activitysteps,sum,na.rm=TRUE)
```

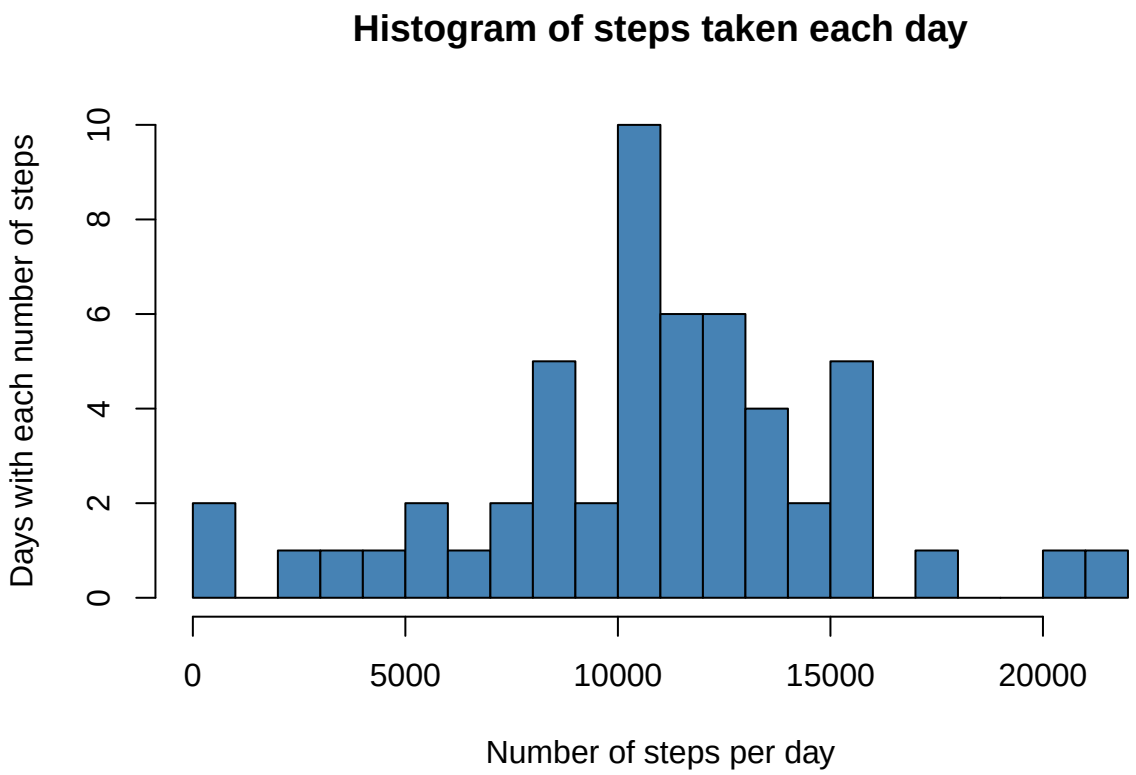
- View the header of the dataset containing the total number of steps each day

```
head(steps_per_day)
```

```
##         date steps
## 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
```

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

```
hist(steps_per_day$steps, main="Histogram of steps taken each day", xlab="Number of steps per day", ylab
```



- Calculate and report the mean and median of the total number of steps taken per day by showing the summary of the dataset containing the total steps per day

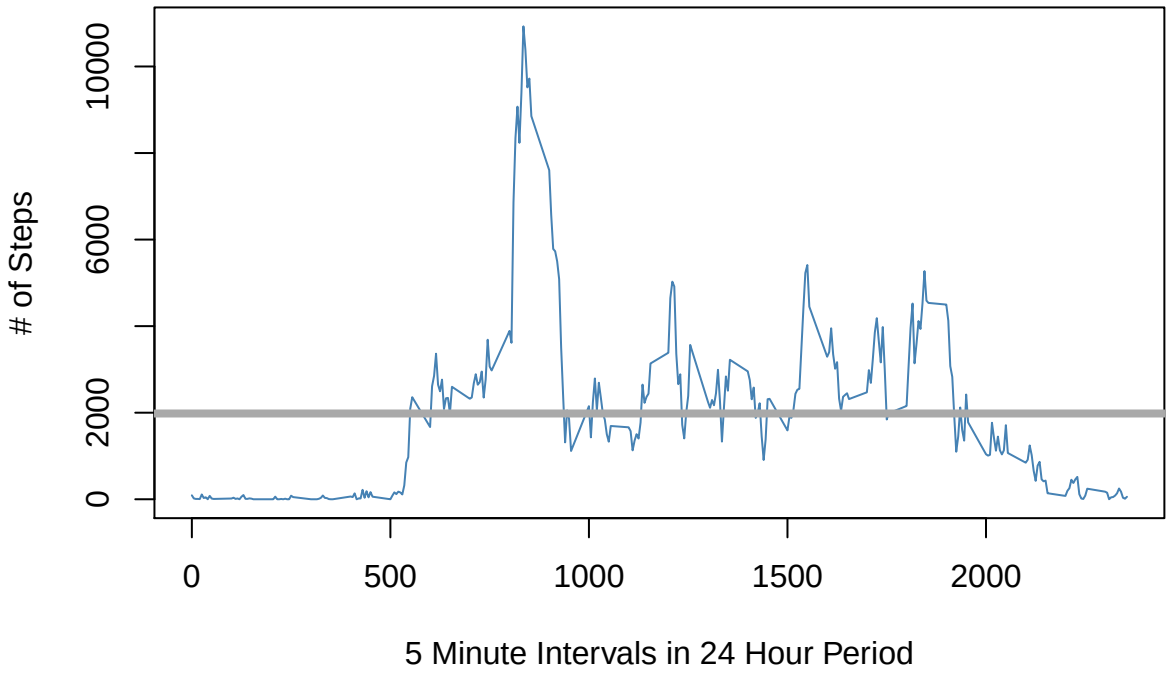
```
summary(steps_per_day)
```

```
##       date                steps
##  Min.   :2012-10-02   Min.   :   41
##  1st Qu.:2012-10-16   1st Qu.: 8841
##  Median :2012-10-29   Median :10765
##  Mean   :2012-10-30   Mean   :10766
##  3rd Qu.:2012-11-16   3rd Qu.:13294
##  Max.   :2012-11-29   Max.   :21194
```

What is the average daily activity pattern?

- Make 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 days (y-axis). First create a dataset that sums up the number of steps per 5 minute interval. Then create a horizontal line showing the average number of steps across all intervals.

```
steps_per_5_min_interval <-aggregate(steps~interval,data=activity,sum,na.rm=TRUE)
with(steps_per_5_min_interval, plot(interval, steps, type = "l", main = "Average Daily Activity Pattern
abline(h=mean(steps_per_5_min_interval$steps), col = "dark grey", lwd = 4)
```



- Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps? Load the dplyr package. Arrange the dataset in descending order of steps per interval and show the first row of the resulting dataset.

```
sorted_interval_data <- arrange(steps_per_5_min_interval, desc(steps))
sorted_interval_data[1,]
```

```
##   interval steps
## 1      835 10927
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

```
sum(is.na(activity$steps))
```

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

- 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. First, I'm going to calculate the average number of steps per 5 minute time interval, as was shown in the plot above, and then divide this by 53(the number of days included in the dataset), to obtain the average number of steps per time interval per day (which is 37.3826 steps per 5 minute time interval). Then I'll replace the NAs with this value, and show the header of this new dataset.

```
average_steps_per_interval <- mean(steps_per_5_min_interval$steps)
average_steps_per_interval
```

```
## [1] 1981.278
```

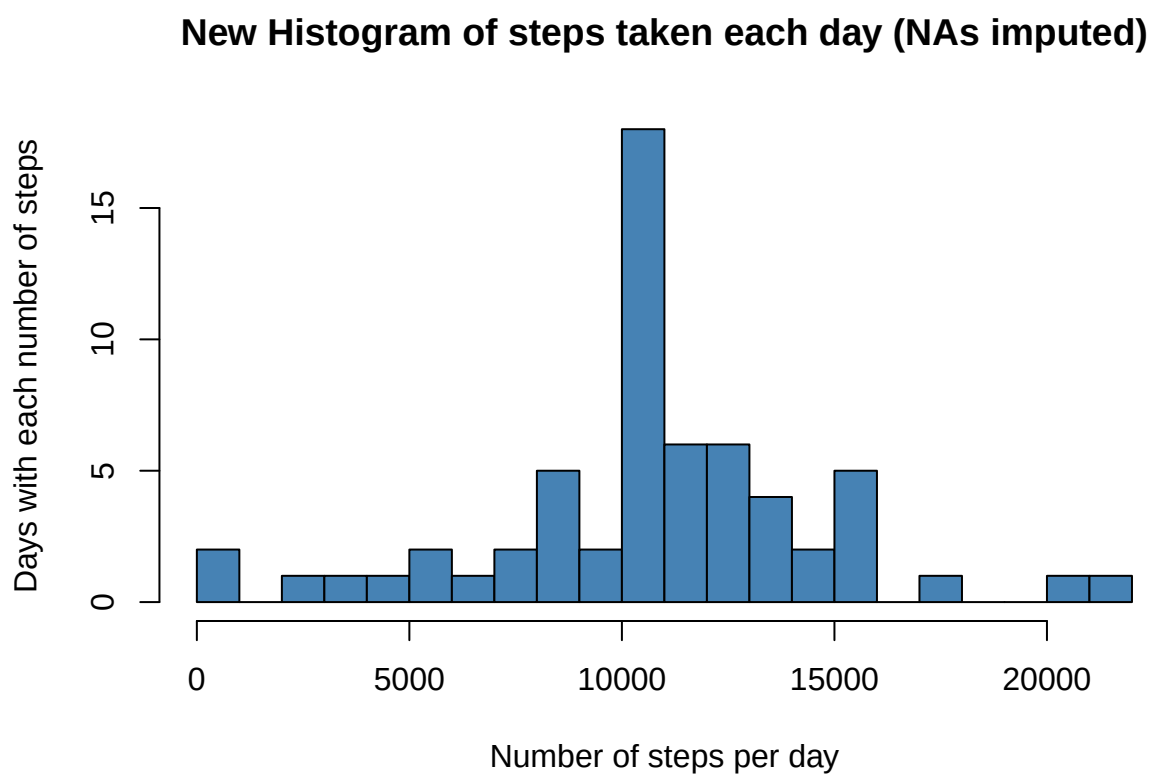
```
activity_with_NAs_replaced_by_interval_mean <-  mutate(activity, steps = ifelse(is.na(steps), mean(steps
head(activity_with_NAs_replaced_by_interval_mean)
```

```
##     steps       date interval
## 1 37.3826 2012-10-01        0
## 2 37.3826 2012-10-01        5
## 3 37.3826 2012-10-01       10
## 4 37.3826 2012-10-01       15
## 5 37.3826 2012-10-01       20
## 6 37.3826 2012-10-01       25
```

```
new_steps_per_day <-aggregate(steps~date,data=activity_with_NAs_replaced_by_interval_mean,sum,na.rm=TRU
```

- 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?* Analysis: The median and mean values are now both equal to 10,766, compared to the median the value of 10,765 for the median and 10,766 for the mean in the dataset before the NAs were imputed. There was very little change to the median and no change to the mean. The 1st qu changed from 8841 to 9819, while the 3rd qu changed from 13,294 to 12,811. There were more days that had around 10,000 to 11,0000 steps.

```
new_steps_per_day <-aggregate(steps~date,data=activity_with_NAs_replaced_by_interval_mean,sum,na.rm=TRU
hist(new_steps_per_day$step, main="New Histogram of steps taken each day (NAs imputed)", xlab="Number o
```




```
summary(new_steps_per_day)
```

```
##       date               steps
##  Min.   :2012-10-01   Min.   :   41
##  1st Qu.:2012-10-16   1st Qu.: 9819
##  Median :2012-10-31   Median :10766
```

```
##  Mean   :2012-10-31   Mean   :10766
##  3rd Qu.:2012-11-15   3rd Qu.:12811
##  Max.   :2012-11-30   Max.   :21194
```

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.

```
new_steps_per_day$week <- ifelse(weekdays(new_steps_per_day$date) %in% c("Saturday", "Sunday"), "Weekend
head(new_steps_per_day)
```

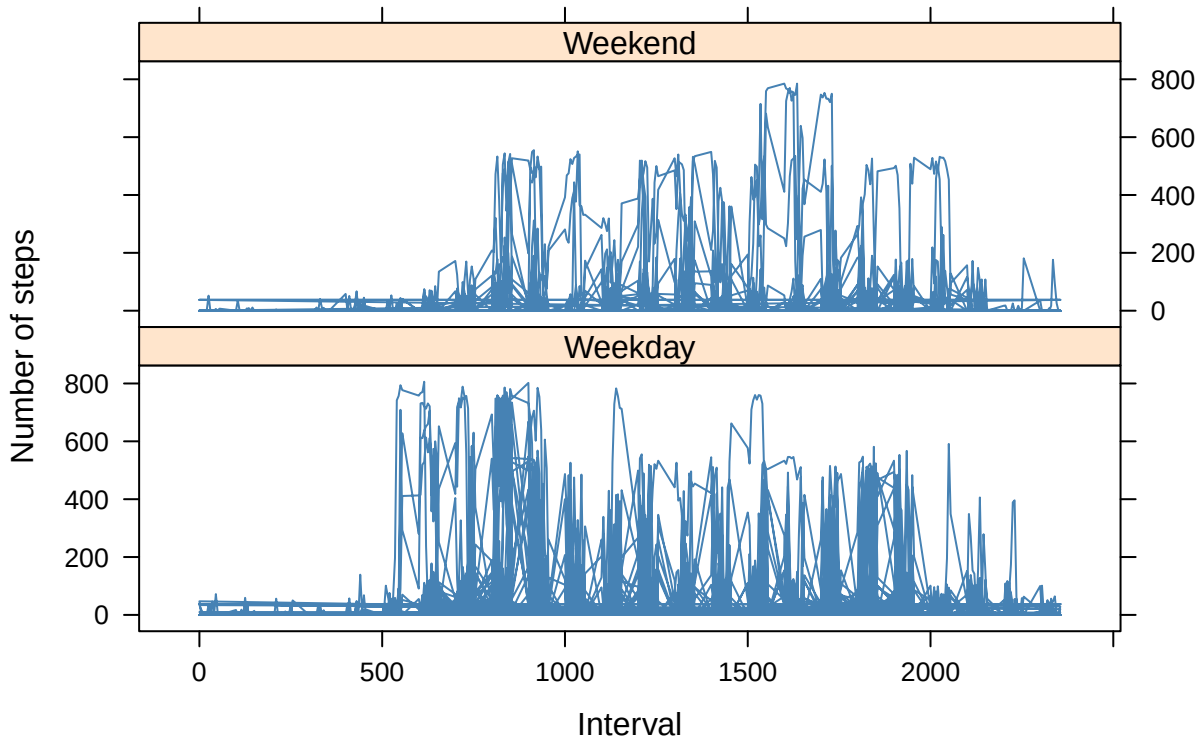
```
##         date    steps    week
## 1 2012-10-01 10766.19 Weekday
## 2 2012-10-02   126.00 Weekday
## 3 2012-10-03 11352.00 Weekday
## 4 2012-10-04 12116.00 Weekday
## 5 2012-10-05 13294.00 Weekday
## 6 2012-10-06 15420.00 Weekend
```

```
activity_with_NAs_replaced_by_interval_mean$week <- ifelse(weekdays(activity_with_NAs_replaced_by_interv
head(activity_with_NAs_replaced_by_interval_mean)
```

```
##     steps       date interval    week
## 1 37.3826 2012-10-01        0 Weekday
## 2 37.3826 2012-10-01        5 Weekday
## 3 37.3826 2012-10-01       10 Weekday
## 4 37.3826 2012-10-01       15 Weekday
## 5 37.3826 2012-10-01       20 Weekday
## 6 37.3826 2012-10-01       25 Weekday
```

- 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.* Analysis: On weekdays, people tend to take more steps earlier in the day, while people tend to sleep in laters and to begin taking more significant amounts of steps starting at about mid day.

```
xyplot(steps ~ interval | week, data = activity_with_NAs_replaced_by_interval_mean, layout = c(1,2), typ
```


Time Series Panel Plot by Weekend / Weekday
Weekend
Weekday
Number of steps
Interval
0
200
400
600
800
0
500
1000
1500
2000