# Reproducible Research: Peer Assessment 1

Suzannah Herron

## Data Overview

*Activity Monitoring Dataset* with:
- **steps**: Number of steps taken in a 5-min interval (missing values coded as `NA`)
- **date**: Date in YYYY-MM-DD format
- **interval**: Identifier for the 5-min interval in which measurement was taken

## 1. Loading and preprocessing the data

```
file_name <- "activity.zip"
if (!file.exists(file_name)) {
    url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
    download.file(url, file_name, method = "curl")
}

if (!file.exists("activity.csv")) {
    unquip(file_name)
}

activity <- read.csv("activity.csv")
```

Change `date` column from characters strings.

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
activity$date <- ymd(activity$date)
summary(activity)
```

```
##      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
```

There are 2304 `NA` values

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

Removing NA values

```
act_na <- activity[!is.na(activity$steps), ]
summary(act_na)
```

```
##      steps             date              interval
##  Min.   :  0.00   Min.   :2012-10-02   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-29   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-30   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-16   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-29   Max.   :2355.0
```

### 2.1 Calculate total number of steps taken per day

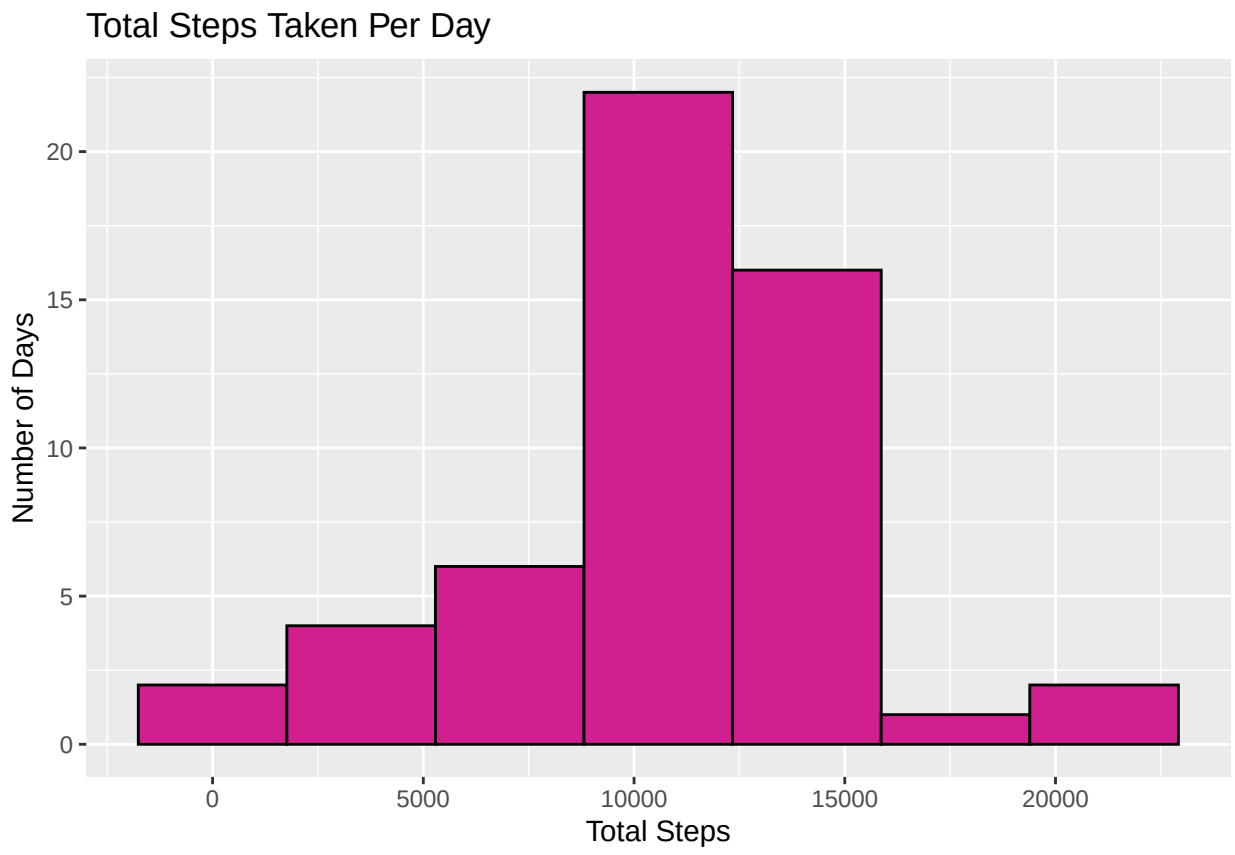
```
act_sum <- aggregate(steps ~ date, act_na, sum)
head(act_sum)
```

```
##         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
```

```
library(ggplot2)
```

### 2.2 Graph of Sum of Steps Taken Per Day

```
# png(filename = "total_steps_hist.png", width = 480, height = 480, bg = NA)
ggplot(data = act_sum) +
  geom_histogram(mapping = aes(steps), fill = "violetred", bins = 7, color = "black") +
  labs(title = "Total Steps Taken Per Day",
       x = "Total Steps",
       y = "Number of Days")
```



```
# dev.off()
```

### 2.3 Calculate average number of steps taken per day

```
act_mean <- mean(act_sum$steps)
act_med <- median(act_sum$steps)
act_mean
```

```
## [1] 10766.19
```

```
act_med
```

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

The mean of total steps per day is $1.0766189 \times 10^4$ and the median is 10765.

## 3. What is the average daily activity pattern?

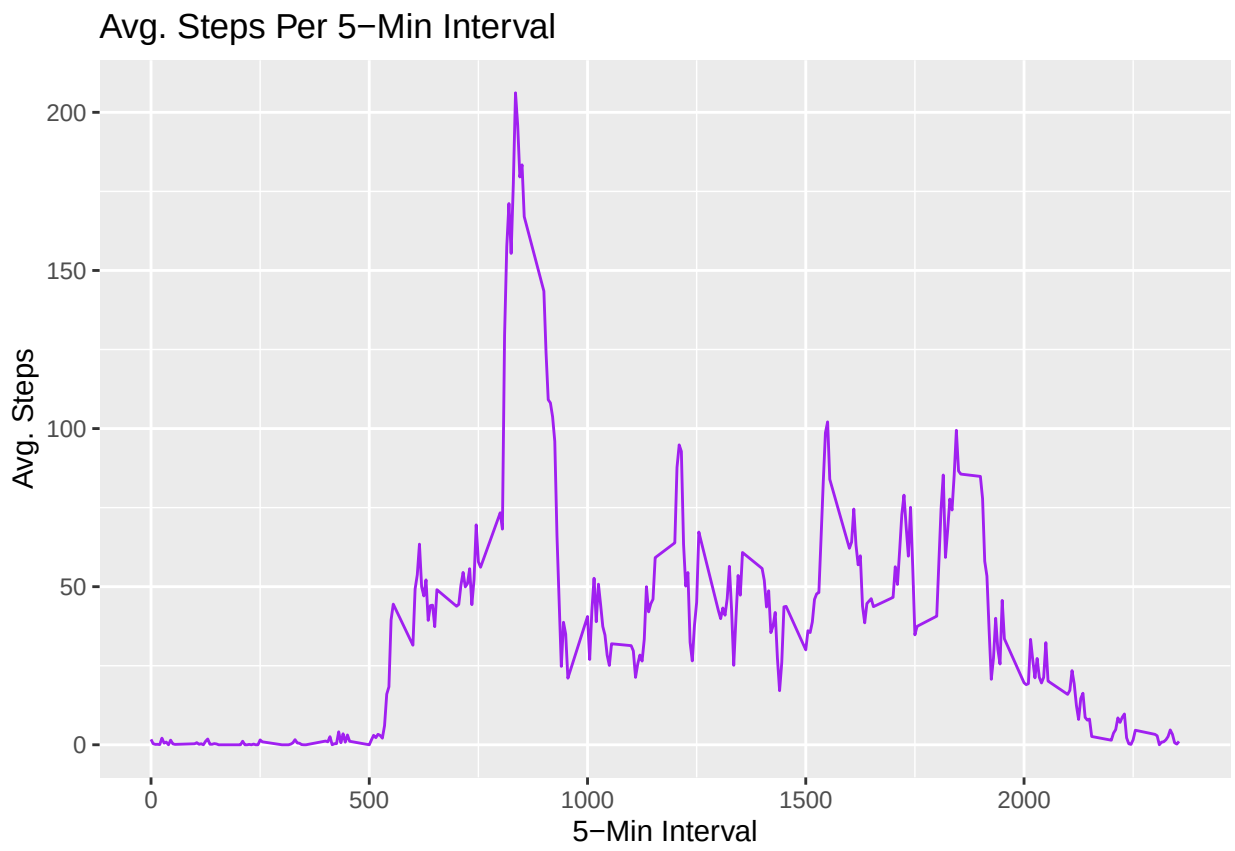
### 3.1 Make a time series plot of 5-min interval and avg num of steps taken averaged across all days

Calculate mean of steps across intervals

```
int_avg <- aggregate(steps ~ interval, act_na, mean)
head(int_avg)
```

```
##   interval     steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

```
# png(filename = "avg_intervals.png", width = 480, height = 480, bg = NA)
ggplot(data = int_avg) +
  geom_line(mapping = aes(x = interval, y = steps), color = "purple") +
  labs(title = "Avg. Steps Per 5-Min Interval",
       x = "5-Min Interval",
       y = "Avg. Steps")
```



```
# dev.off()
```

### 3.2 Which 5-min interval has most steps?

```
int_max <- int_avg[which.max(int_avg$steps), 1]
int_max
```

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

The 5-minute interval with the most steps is 835.

## 4. Imputing missing values

### 4.1 Calculate the total number of NAs

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

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

The total number of NAs in the dataset is 2304.
Alternatively, `summary()` will give you this info as well:

```
summary(activity)
```

```
##      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
```

### 4.2 Devise strategy for filling in missing values and create new dataset

Find mean of average of steps per interval.

```
avg <- mean(int_avg$steps)
avg
```

```
## [1] 37.3826
```

Create df with the NAs.

```
nas <- is.na(activity$steps)
```

Insert `avg` into values with `NA`

```
data_avg <- activity
data_avg[nas, 1] <- avg
head(data_avg)
```

```
##     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
```

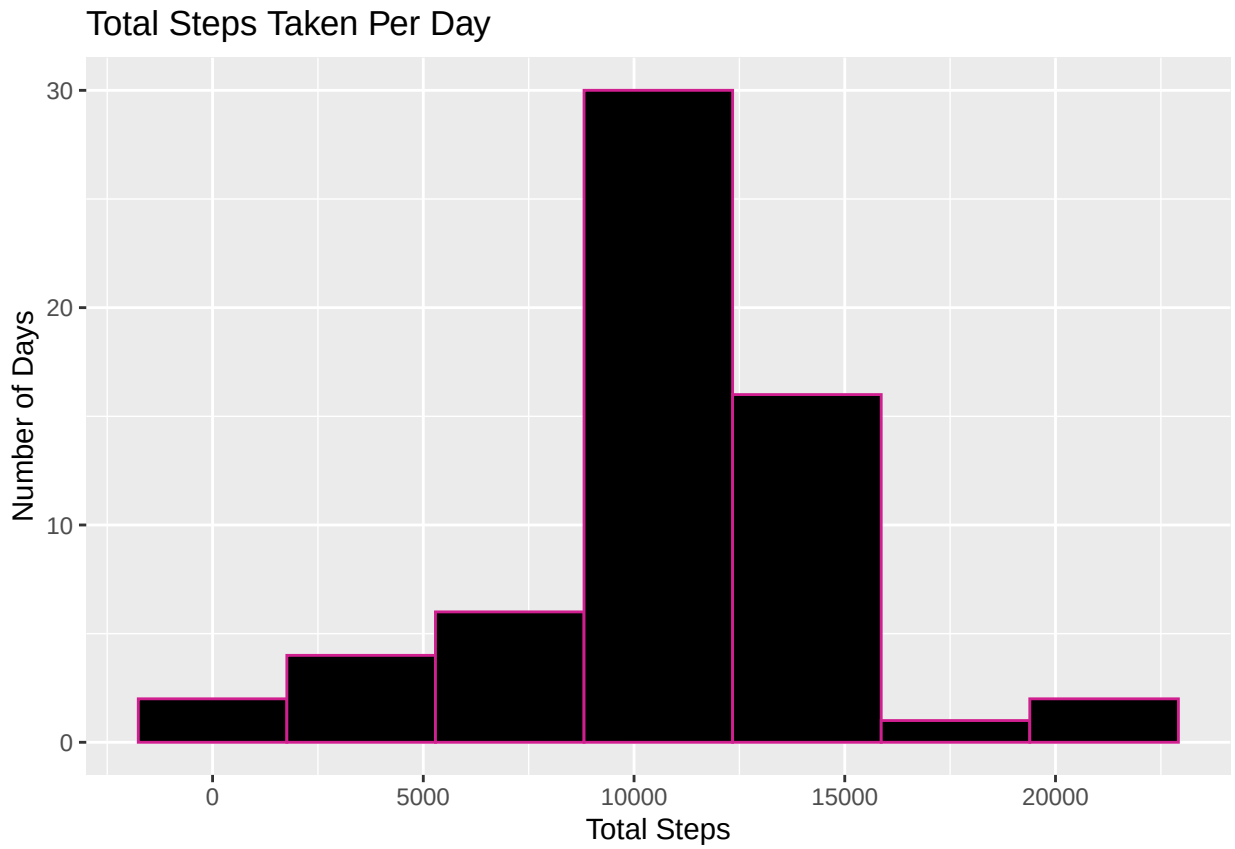
Use new `data_avg` dataset to calculate sum of step per day

```
data_sum <- aggregate(steps ~ date, data_avg, sum)
head(data_sum)
```

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

**4.3 Make histogram of total number of steps taken each day. Report mean and median. How does it differ?**

```
# png(filename = "total_per_day.png", width = 480, height = 480, bg = NA)
ggplot(data = data_sum) +
  geom_histogram(mapping = aes(steps), fill = "black", bins = 7, color = "violetred") +
  labs(title = "Total Steps Taken Per Day",
       x = "Total Steps",
       y = "Number of Days")
```




```
# dev.off()
```

```
data_mean <- mean(data_sum$steps)
data_med <- median(data_sum$steps)
data_mean
```

```
## [1] 10766.19
```

```
data_med
```

```
## [1] 10766.19
```

The mean without the `NA` values is $1.0766189 \times 10^4$ and with inserted averages, it is $1.0766189 \times 10^4$.
The median without the `NA` values is 10765 and with inserted averages, it is $1.0766189 \times 10^4$.

```
diff_mean <- act_mean - data_mean
diff_med <- act_med - data_med
diff_mean
```

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

```
diff_med
```

```
## [1] -1.188679
```

There is no impact extracting the NAs to the mean and a very negligible difference of -1.1886792 to the median.

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

### 5.1 Using data_sum, create a new variable factor with two levels of "weekday" and "weekend".

```
library(dplyr)
```

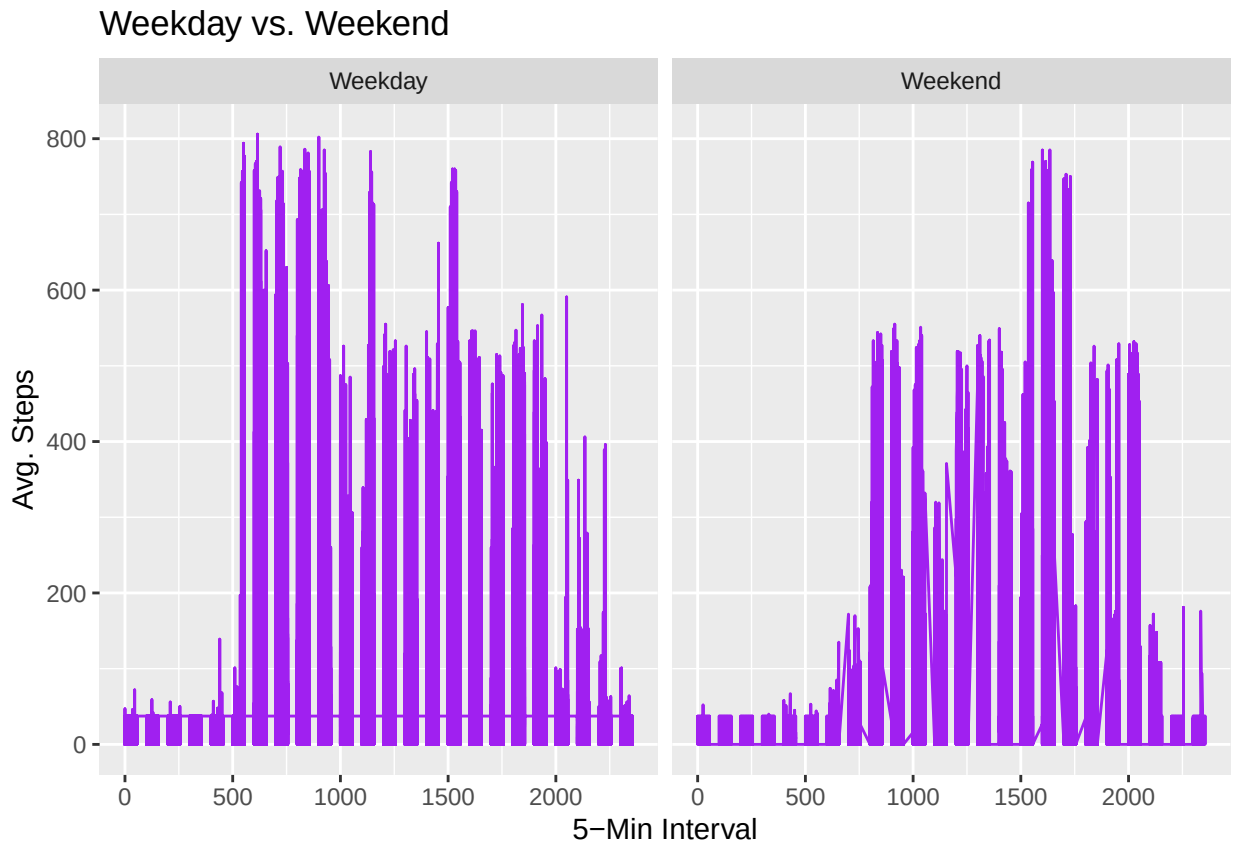
Using weekdays() to find day of the week then changing "Saturday" and "Sunday" to "Weekend" and the rest to "Weekday".

```
weekend <- mutate(data_avg, weekday = ifelse((weekdays(data_avg$date) == "Saturday" | weekdays(data_avg$
head(weekend)
```

```
##     steps       date interval weekday
## 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
```

### 5.2 Make a plot of weekends vs. weekdays

```
# png(filename = "weekend_intervals.png", width = 480, height = 480, bg = NA)
ggplot(data = weekend) +
  geom_line(mapping = aes(x = interval, y = steps), color = "purple") +
  facet_wrap(~weekday) +
  labs(title = "Weekday vs. Weekend",
       x = "5-Min Interval",
       y = "Avg. Steps")
```


Weekday vs. Weekend
Weekday
Weekend
Avg. Steps
800
600
400
200
0
0
500
1000
1500
2000
5−Min Interval

```
# dev.off()
```