Reproducible Research: Peer Assessment 1

James Portman

June 4, 2016

Data

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 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 preprocessing the data

1. Load the unzipped data into df_activity data frame.

```
df_activity <- read.csv("./activity.csv", header=TRUE)
str(df_activity) # Initial look at data.
```

```
## '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 ...
```

2. Transform the date into a date object.

```
df_activity$date <- as.Date(as.character(df_activity$date, "%Y%m%d"))
```

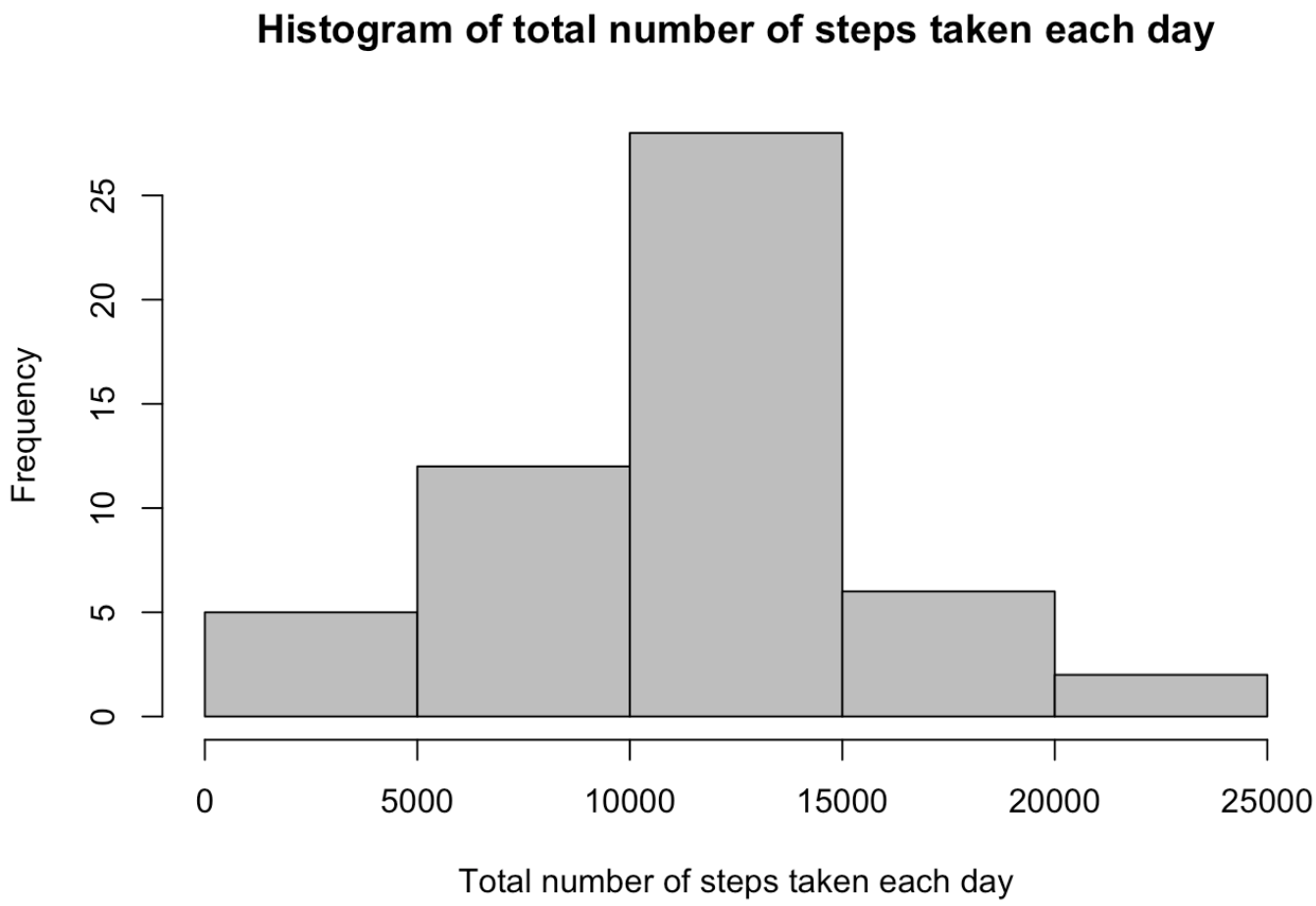
What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day

```
# Group data by date.
steps_per_day <- aggregate(steps ~ date, df_activity, sum)
```

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

```
hist(steps_per_day$steps,
    xlab = "Total number of steps taken each day",
    main = "Histogram of total number of steps taken each day",
    ylab = "Frequency",
    col  = "gray")
```



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

```
mean_steps <- mean(steps_per_day$steps)
print(mean_steps)
```

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

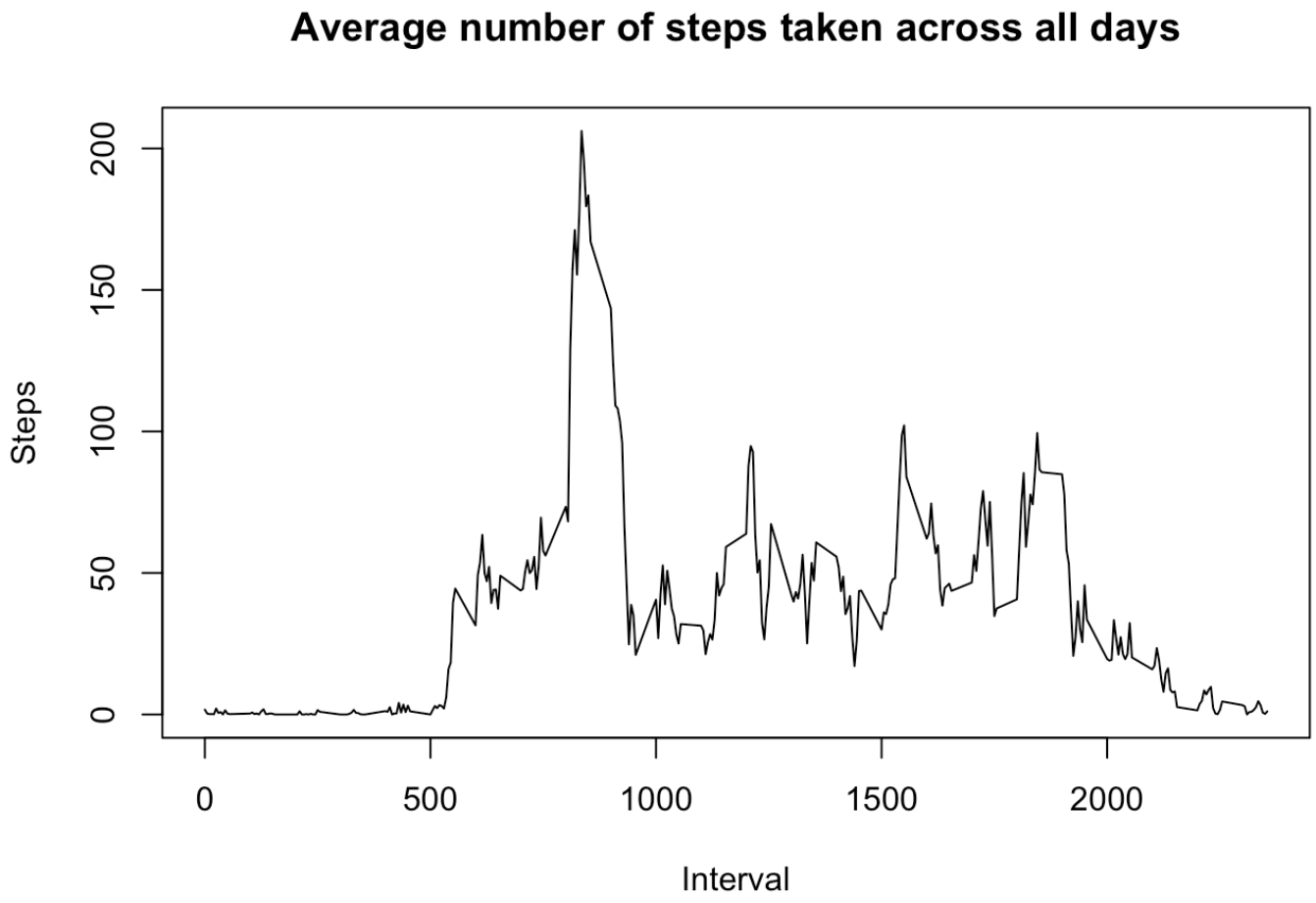
```
median_steps <- median(steps_per_day$steps)
print(median_steps)
```

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

What is the average daily activity pattern?

1. 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)

```
df_activity_interval <- aggregate(steps ~ interval, FUN=mean, data=df_activity)

plot(x=df_activity_interval$interval
   , y=df_activity_interval$steps
   , type="l"
   , main="Average number of steps taken across all days"
   , xlab="Interval"
   , ylab="Steps")
```



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

```
df_activity_order <- df_activity_interval[order(-df_activity_interval$steps),]
print(df_activity_order[1, ])
```

```
##     interval    steps
## 104      835 206.1698
```

Inputing missing values

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

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

```
number_NA <- sum(is.na(df_activity$steps))
print(number_NA)
```

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

2. 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.

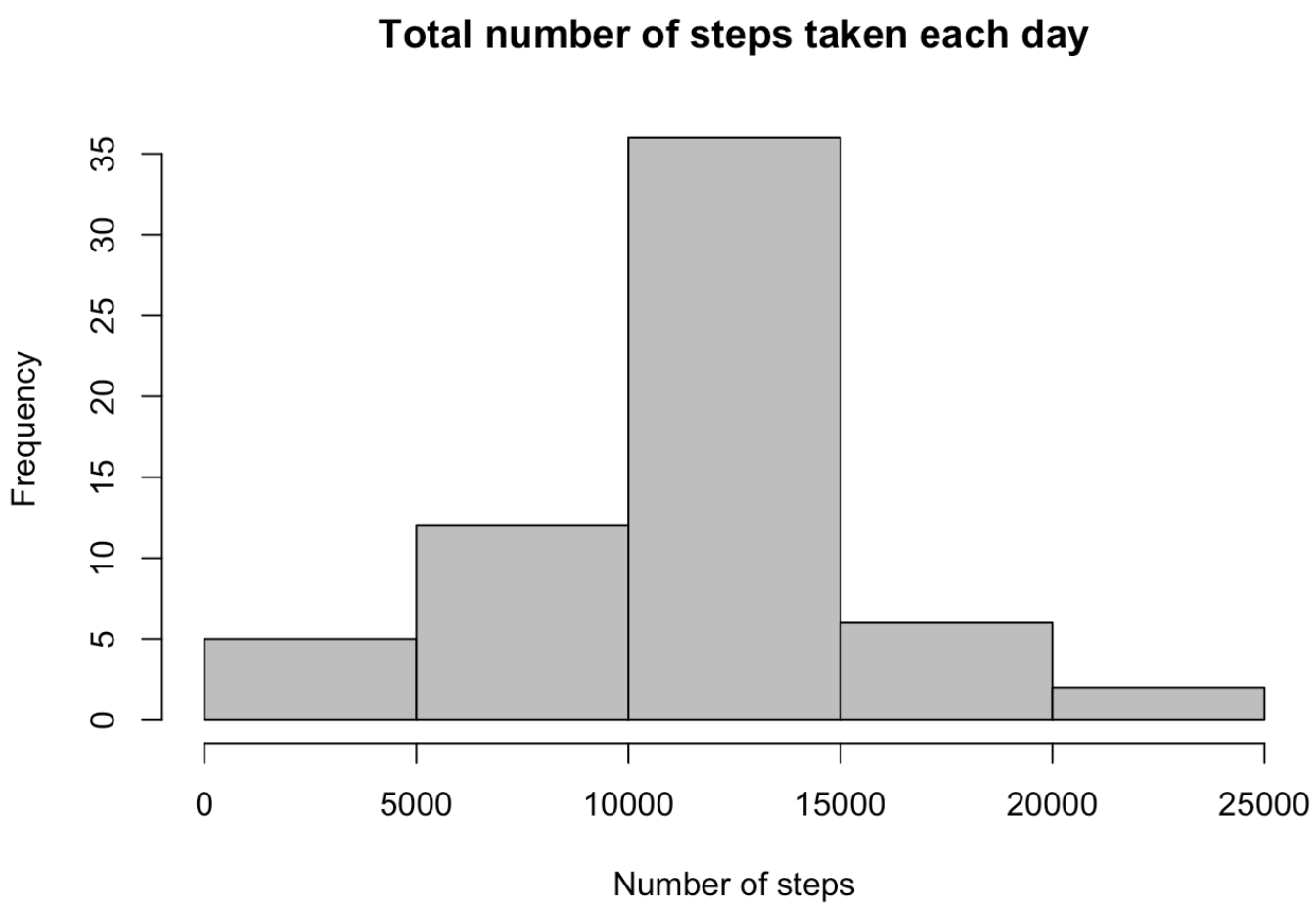
```
# Calculate the mean per 5-minute interval
df_interval_mean <- aggregate(steps ~ interval, FUN=mean, data=df_activity)
```

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

```
# Merge original and mean per interval dataset
df_merged <- merge(x = df_activity, y = df_interval_mean, by="interval")

# Add column steps with missing data filled with the interval mean
df_merged$steps <- ifelse(is.na(df_merged$steps.x), df_merged$steps.y, df_merged$s
teps.x)

# Subselect only steps, date and interval columns
df_new <- df_merged[c("steps", "date", "interval")]
```

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

```
# Aggregate by date and calculate sum
df_combined <- aggregate(steps ~ date, FUN=sum, data=df_new)
vt_agg_day_na <- df_combined$steps
names(vt_agg_day_na) <- df_combined$date

hist(df_combined$steps,
     xlab = "Number of steps",
     main = "Total number of steps taken each day",
     col  = "gray")
```



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

```
new_mean <- mean(df_combined$steps)
new_median <- median(df_combined$steps)

print(new_mean, digits = 15)
```

```
## [1] 10766.1886792453
```

```
print(new_median, digits = 15)
```

```
## [1] 10766.1886792453
```

6. 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?

Replacing the missing value with the average of the interval shifts the median closer to the mean.

Are there differences in activity patterns between weekdays and weekends?

1. 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.

```
df_new$weekdays <- as.factor(ifelse(weekdays(df_new$date) %in% c("Saturday", "Sund
ay"), "weekend", "weekday"))
```

2. Make a panel plot containing a time series plot 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.

```
library(lattice)
# aggregate by weekday and apply mean
df_new_weekdays <- aggregate(steps ~ weekdays + interval, FUN=mean, data=df_new)
```

```
xyplot(steps ~ interval | weekdays, df_new_weekdays
     , type = "l"
     , xlab = "Interval"
     , ylab = "Number of steps"
     , main = "Average steps taken, averaged across all weekday days or weekend da
ys"
     , layout = c(1, 2))
```

Average steps taken, averaged across all weekday days or weekend days