Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.3     ✓ purrr   0.3.4
## ✓ tibble  3.0.4     ✓ dplyr   1.0.2
## ✓ tidyr   1.1.2     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
unzip("activity.zip")
active <- read.csv("activity.csv")
active$date <- as.Date(active$date, format = "%Y-%m-%d")
```

What is mean total number of steps taken per day?

```
steps <- active %>%
        group_by(date) %>%
        summarize(mean_steps = mean(steps, na.rm = TRUE),
                  total_steps = sum(steps, na.rm = TRUE),
                  median_steps = median(steps, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
mean_before_imputatiaon <- mean(steps$total_steps, na.rm = TRUE)
median_before_imputation <- median(steps$total_steps, na.rm = TRUE)
mean_before_imputatiaon
```

```
## [1] 9354.23
```

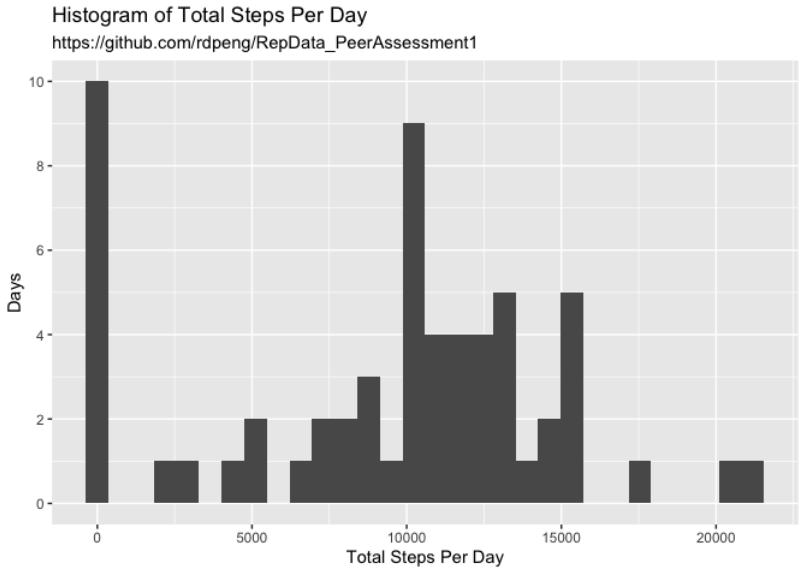
```
median_before_imputation
```

```
## [1] 10395
```

Before imputation to NAs, the mean number of steps per day is 9354.23 and the median is 10395.

```
## Make a histogram of the total number of steps taken each day
steps_hist <- ggplot(steps, aes(total_steps))
steps_hist + geom_histogram(na.rm = TRUE) +
        labs(title = "Histogram of Total Steps Per Day",
             subtitle = "https://github.com/rdpeng/RepData_PeerAssessment1",
             x = "Total Steps Per Day",
             y = "Days") +
        scale_y_continuous(breaks = c(0, 2, 4, 6, 8, 10), labels = c(0, 2, 4, 6, 8, 10))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

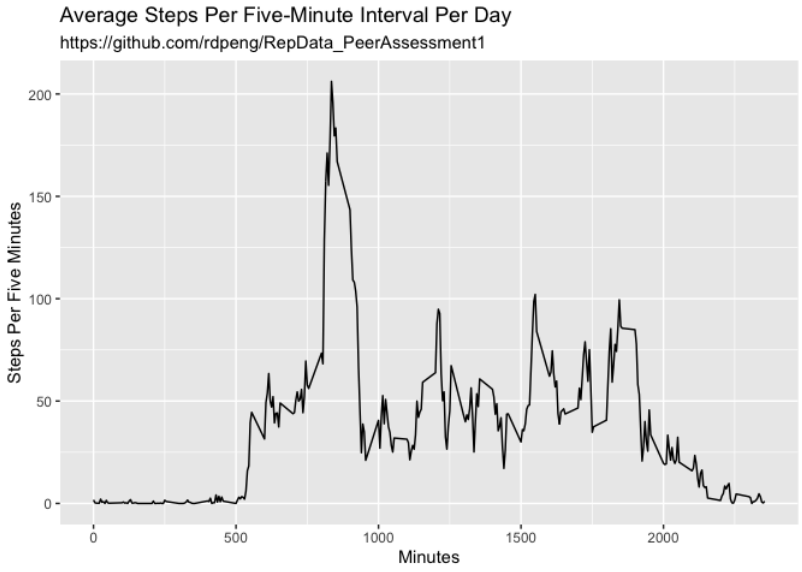


What is the average daily activity pattern?

```
intervals <- active %>%
        group_by(interval) %>%
        summarize(mean_steps = mean(steps, na.rm = TRUE))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
plot_intervals <- ggplot(intervals, aes(interval, mean_steps))
plot_intervals + geom_line() +
        labs(title = "Average Steps Per Five-Minute Interval Per Day",
             subtitle = "https://github.com/rdpeng/RepData_PeerAssessment1",
             x = "Minutes",
             y = "Steps Per Five Minutes")
```



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

```
max_steps <- max(intervals$mean_steps, na.rm = TRUE)
max_step_interval <- intervals$interval[intervals$mean_steps == max_steps]
hour_of_day <- max_step_interval %/% 60
minute_of_hour <- max_step_interval %% 60
hour_of_day
```

```
## [1] 13
```

```
minute_of_hour
```

```
## [1] 55
```

The five minute interval with maximum steps is from 835 to 840 minutes (13h55 to 14h). ## Imputing missing values

```
### (i.e. the total number of rows with NAs)
sum(!complete.cases(active))
```

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

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

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

There are 2304 cases with an NA and all of them are in the steps variable.

```
active_not_na <- active %>%
        select(steps, date, interval) %>%
        group_by(interval) %>%
        mutate(mean_steps = mean(steps, na.rm = TRUE))
active_not_na$steps[is.na(active_not_na$steps)] <-
        active_not_na$mean_steps[is.na(active_not_na$steps)]
```

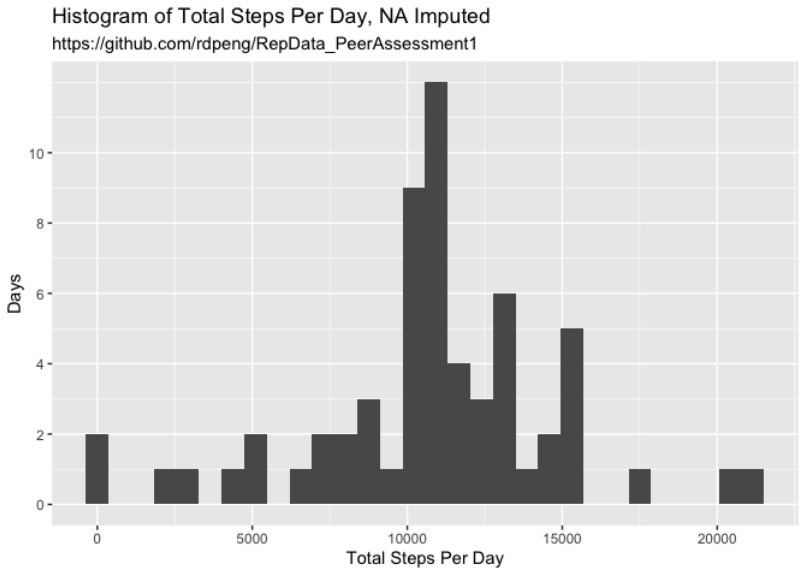
Histogram of Total Daily Steps After Imputing NAs

```
steps_not_na <- active_not_na %>%
        group_by(date) %>%
        summarize(total_steps = sum(steps))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
steps_not_na_hist <- ggplot(steps_not_na, aes(total_steps))
steps_not_na_hist + geom_histogram(na.rm = TRUE) +
        labs(title = "Histogram of Total Steps Per Day, NA Imputed",
             subtitle = "https://github.com/rdpeng/RepData_PeerAssessment1",
             x = "Total Steps Per Day",
             y = "Days") +
        scale_y_continuous(breaks = c(0, 2, 4, 6, 8, 10), labels = c(0, 2, 4, 6, 8, 10))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Calculate and report the mean

and median total number of steps taken per day.

```
mean_after_imputation <- mean(steps_not_na$total_steps)
median_after_imputation <- median(steps_not_na$total_steps)
mean_after_imputation
```

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

```
median_after_imputation
```

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

```
mean_after_imputation - mean_before_imputatiaon
```

```
## [1] 1411.959
```

```
median_after_imputation - median_before_imputation
```

```
## [1] 371.1887
```

After imputation, the mean and median are the same (10766.19). The decimal in a median results from imputation of means to NAs. The mean is 1411.959 larger and the median is 371.1887 larger than before imputation.

Are there differences in activity patterns between weekdays and weekends?

```
## Create a new factor variable in the dataset with two levels – "weekday" and "weekend"
weekday_index <- weekdays(active_not_na$date)
active_not_na$weekday[weekday_index %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")] <- "weekday"
```

```
## Warning: Unknown or uninitialised column: `weekday`.
```

```
active_not_na$weekday[weekday_index %in% c("Saturday", "Sunday")] <- "weekend"
active_not_na$weekday <- as.factor(active_not_na$weekday)
```

Panel Time Series of Weekday and Weekend Activity

```
weekday_steps <- active_not_na %>%
        group_by(weekday, interval) %>%
        summarize(average_steps = mean(steps))
```

```
## `summarise()` regrouping output by 'weekday' (override with `.groups` argument)
```

```
weekday_plot <- ggplot(weekday_steps, aes(interval, average_steps))
weekday_plot + geom_line() +
        labs(title = "Comparison of Weekday and Weekend Steps Per Five-Minute Interval Per Day",
             subtitle = "https://github.com/rdpeng/RepData_PeerAssessment1",
             x = "Minutes",
             y = "Steps Per Five Minutes") +
        facet_grid(weekday~.)
```