

Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

```
render("Activity monitoring data Analysis.Rmd", output_format = "word_document")
```

```
source("data.R")
```

What is mean total number of steps taken per day?

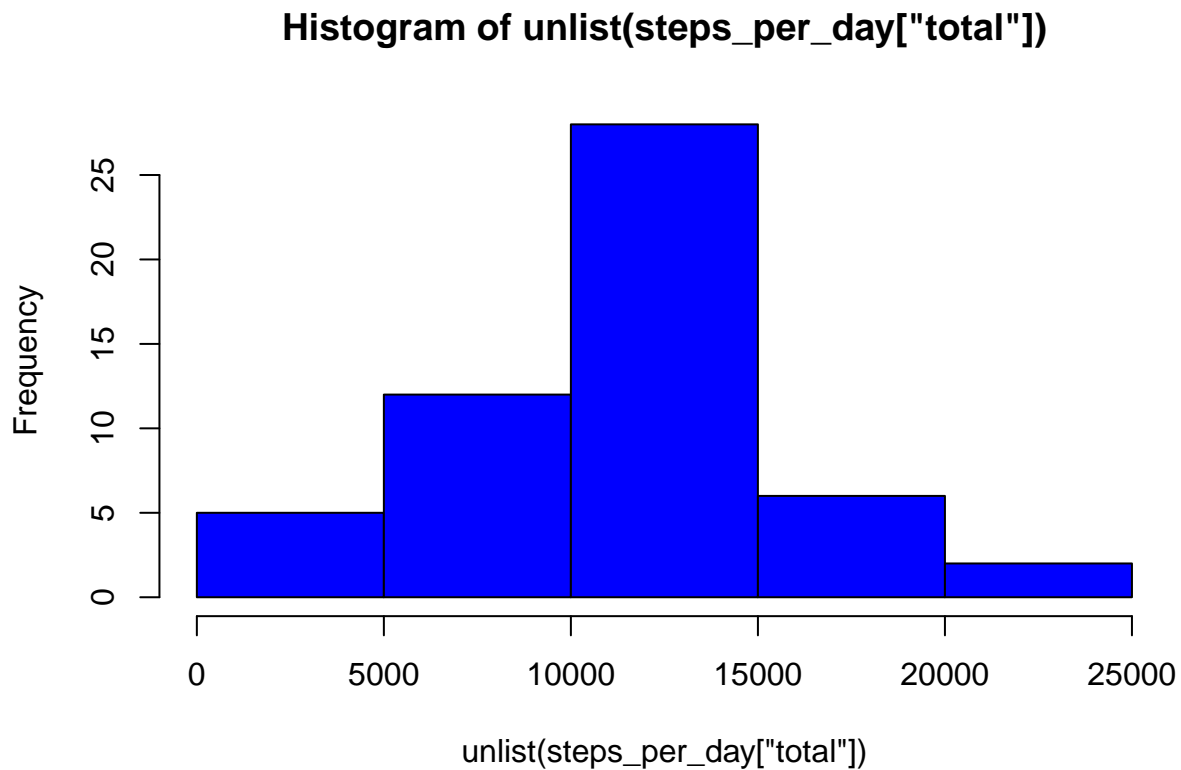
1. Calculate the total number of steps taken per day

```
tot <- complete_activities %>%  
  group_by(date) %>%  
  summarise(total=sum(steps))  
head(tot)
```

```
## # A tibble: 6 x 2  
##   date      total  
##   <fct>      <int>  
## 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
```

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

```
steps_per_day <- complete_activities %>%  
  group_by(date) %>%  
  summarise(total=sum(steps))  
hist(unlist(steps_per_day['total']), col = "blue")
```



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

```
mean(as.numeric(unlist(steps_per_day)))
```

```
## [1] 5398.453
```

```
median(as.numeric(unlist(steps_per_day)))
```

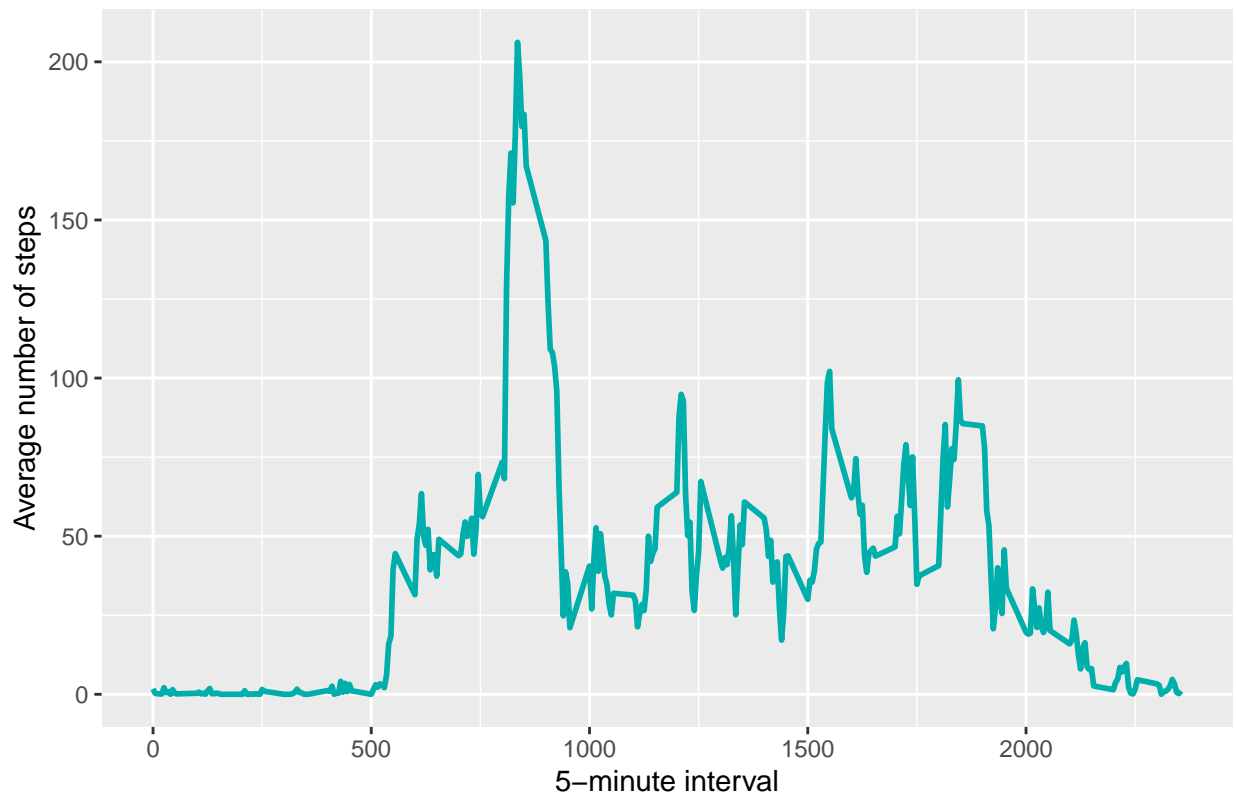
```
## [1] 59.5
```

What is the average daily activity pattern?

1. 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)

```
avrg <- complete_activities %>%
  group_by(interval) %>%
  summarize(avg_steps=mean(steps))
ggplot(data = avrg, aes(x=interval, y=avg_steps))+
  geom_line(color = "#00AFAB", size = 1) + xlab("5-minute interval") + ylab("Average number of steps")
```

Average steps taken during 5-minute interval



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

```
max_n_steps <- avrg$interval[which.max(avrg$avg_steps)]
max_n_steps
```

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

Imputing missing values

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

```
non_complete_count <- sum(!complete.cases(activities))
non_complete_count
```

```
## [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.

```

for (i in 1:nrow(complete_activities)) {
  if (is.na(complete_activities$steps[i])) {
    j <- which(complete_activities$interval[i] == avrg$interval)
    # Assign the value to replace the NA
    complete_activities$steps[i] <- avrg[j,]$avg_steps
  }
}

complete_activities$date <- as.Date(complete_activities$date)

```

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

```
all_activities <- activities
```

4. 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?

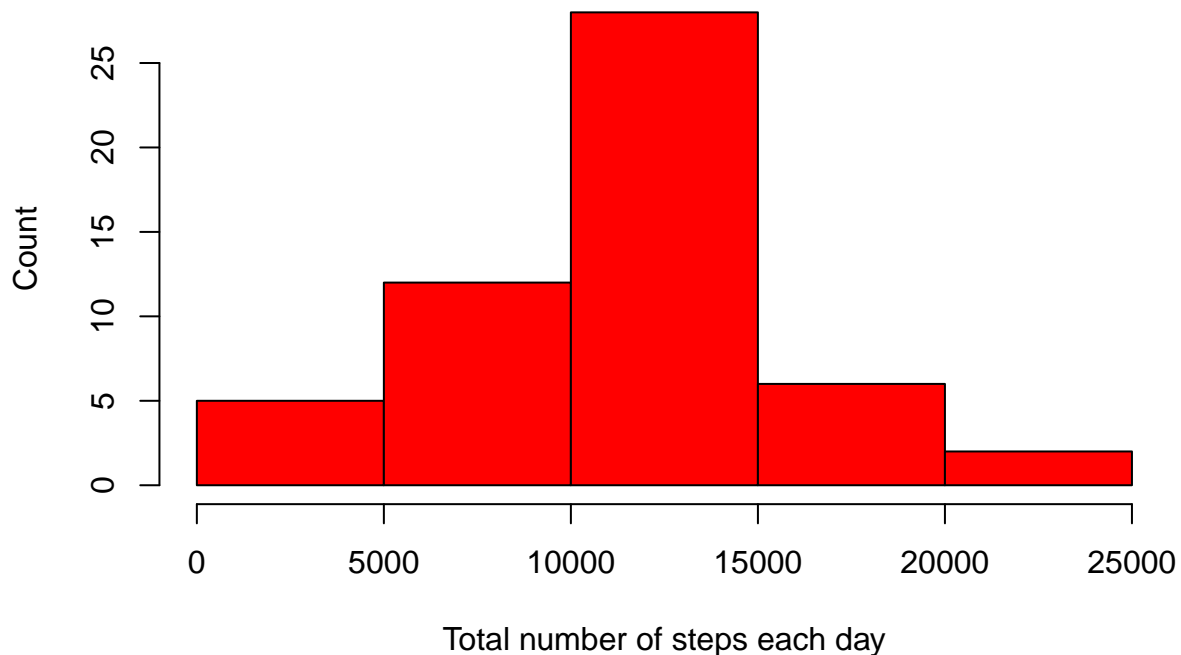
```

fillstepsperday <- complete_activities %>%
  group_by(date) %>%
  summarize(total_steps=sum(steps))

# Show histogram of steps per day
hist(fillstepsperday$total_steps,
      xlab="Total number of steps each day",
      ylab="Count",
      main="Histogram of total number of steps each day",
      col="red"
    )

```

Histogram of total number of steps each day



```
fillmeansteps <- mean(fillstepsperday$total_steps)
fillmeansteps
```

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

```
fillmediansteps <- median(fillstepsperday$total_steps)
```

The mean total number of steps per day is 1.0766189×10^4 . and the median is 10765.

The mean is unchanged, but the median moved slightly and is now equal to the mean. Inputting missing data based on an average leaves the average daily total number of steps the same while changing the median.

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.

```
complete_activities$day <- weekdays(complete_activities$date)
complete_activities$daytype <- "weekday"
complete_activities$daytype[complete_activities$day %in% c("Saturday", "Sunday")] <- "weekend"
```

2. 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.

```
dayaverage <- complete_activities %>%  
  group_by(daytype, interval) %>%  
  summarize(average_steps=mean(steps))  
  
qplot(interval, average_steps, data=dayaverage,  
  geom="line",  
  xlab="Interval",  
  ylab="Number of Steps (Average)",  
  main="Average steps taken Weekends vs. Weekdays",  
  facets =daytype ~ .)
```

