Peer-graded Assignment: Course Project 1

Commit containing full submission

1. Code for reading in the dataset and/or processing the data
2. Histogram of the total number of steps taken each day
3. Mean and median number of steps taken each day
4. Time series plot of the average number of steps taken
5. The 5-minute interval that, on average, contains the maximum number of steps
6. Code to describe and show a strategy for imputing missing data
7. Histogram of the total number of steps taken each day after missing values are imputed
8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends
9. All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

1. Code for reading in the dataset and/or processing the data

- Loaded the data from a zip file.

```
activity_dataset <- "repdata_data_activity.zip"
unzip(activity_dataset)
activity <- read.csv("activity.csv")
str(activity)
```

```
## ’data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

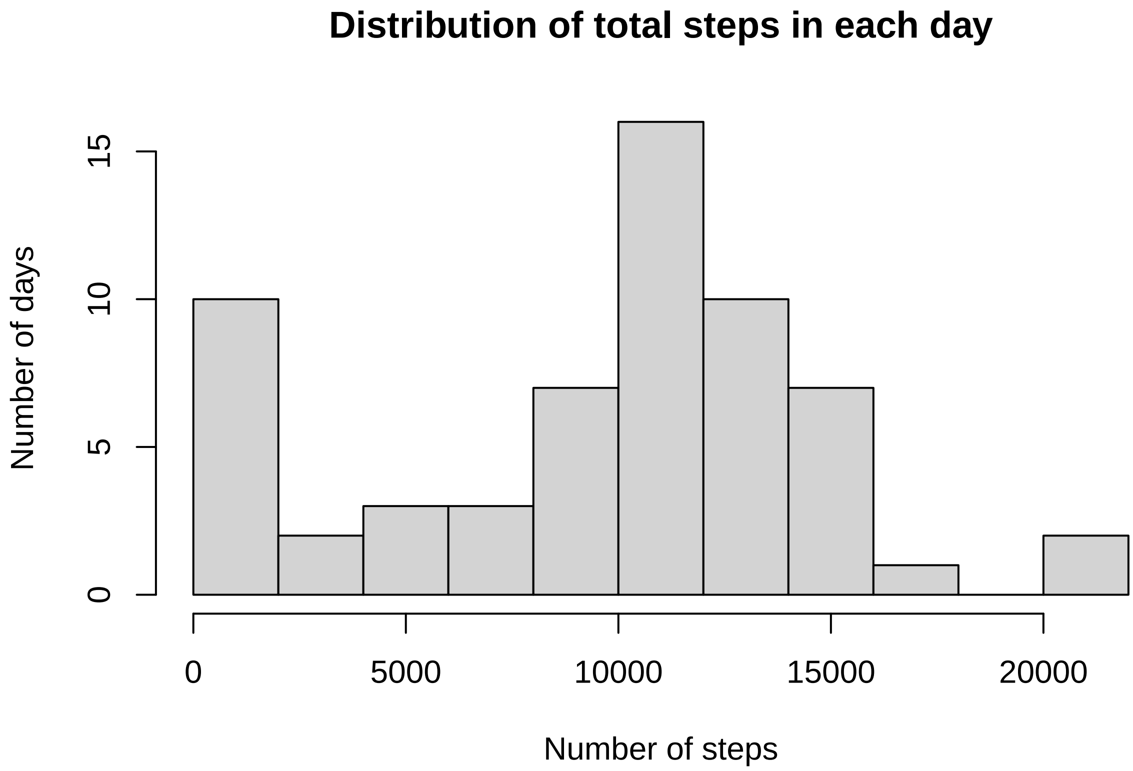
2. Histogram of the total number of steps taken each day

- Dates are character vector, therefore it shoudl be transformed to Date format. Used the lubricate package.
- The next step total number of steps in each day was calculated and assigned to "steps_per_day"
- The total number of steps were used for histogram.

```
library(lubridate)

activity$date <- as.Date(activity$date)
steps_per_day <- aggregate(activity$steps, by=list(activity$date), na.rm=TRUE, sum)

# Assigning column names
colnames(steps_per_day) <- c("date", "total_steps" )
hist(steps_per_day$total_steps, bin=50,breaks = 10,  xlab="Number of steps",ylab="Number of days",  mai
```



3. Mean and median number of steps taken each day

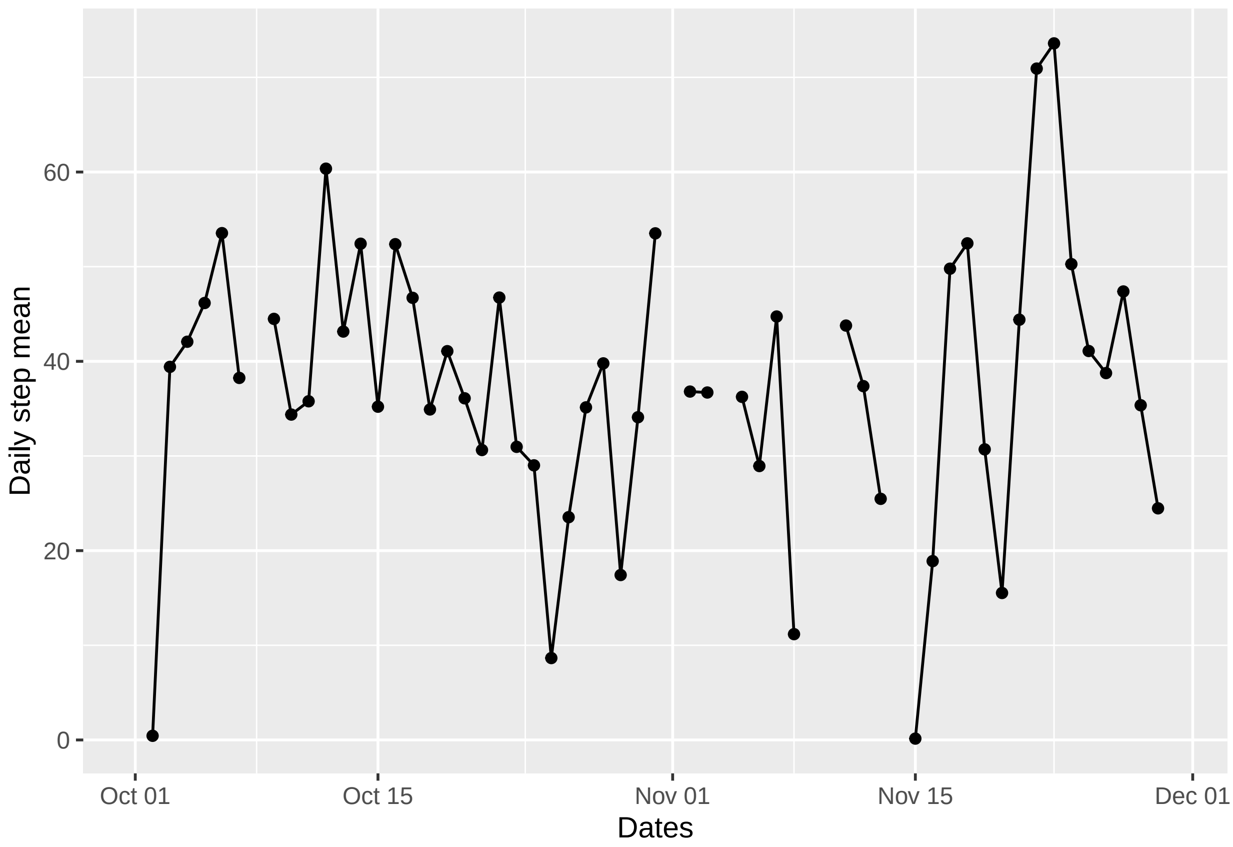
```
library(dplyr)
# calculate mean and median
steps_per_day_stat <- activity %>%
    group_by(date) %>%
     summarize_all(funs(mean=mean(steps), median=median(steps)))
daily_steps_stat<- steps_per_day_stat[c("date", "steps_mean", "steps_median")]

knitr::kable(daily_steps_stat[1:6,1:3], format = "markdown")
```

date	steps_mean	steps_median
2012-10-01	NA	NA
2012-10-02	0.43750	0
2012-10-03	39.41667	0
2012-10-04	42.06944	0
2012-10-05	46.15972	0
2012-10-06	53.54167	0

4. Time series plot of the average number of steps taken

You can also embed plots, for example:



5. The 5-minute interval that, on average, contains the maximum number of steps

```
the_5min_interval <- aggregate(activity$steps, by=list(activity$interval), na.rm=TRUE, mean)
colnames(the_5min_interval) <- c("interval", "average_steps" )
sub<- the_5min_interval[which(the_5min_interval$average_steps == max(the_5min_interval$average_steps)),]
cat("The interval with Max average steps:", sub$interval)
```

```
## The interval with Max average steps: 835
```

6. Code to describe and show a strategy for imputing missing data

As I could not find clear pattern in the missing data, replaced all the missing data with the average steps of interval across all days. I used the average of the 5-minute interval data that was calculated in the last step to replace the missing data.

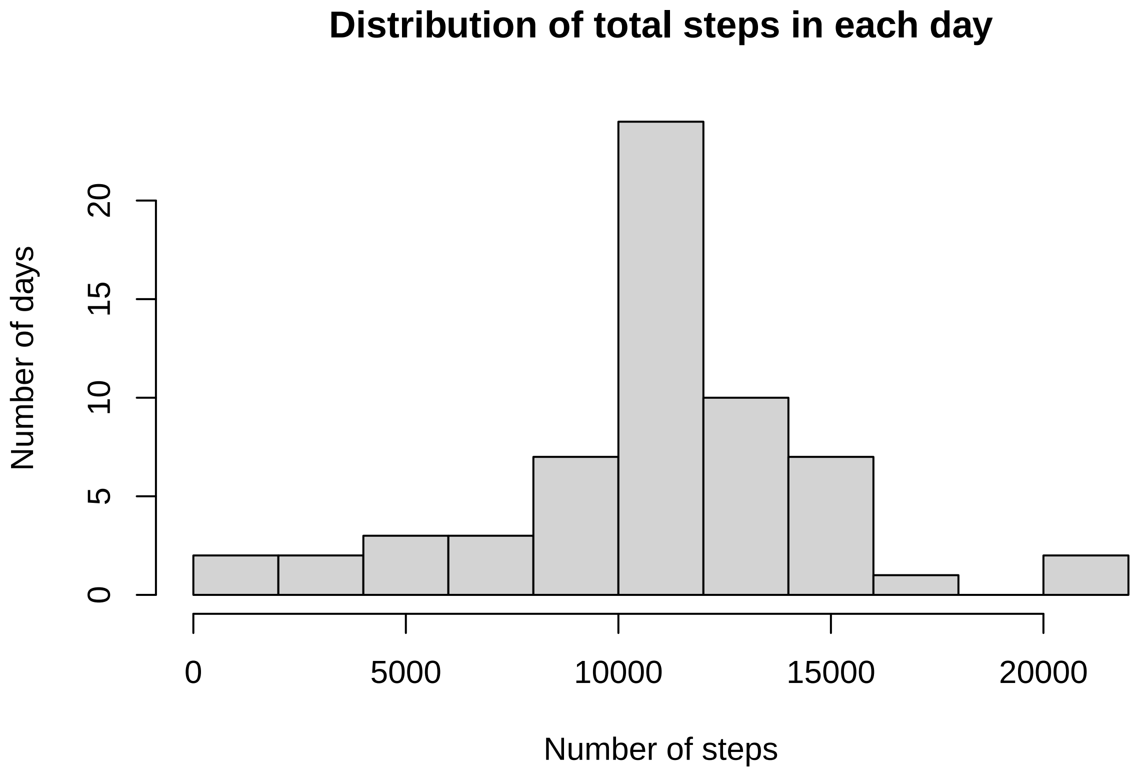
```
n <- 61
the_5min_interval1 <- do.call("rbind", replicate(n, the_5min_interval, simplify = FALSE))

# preserving the original data file
activity2<-activity
idx <- is.na(activity2$steps)
activity2$steps[idx] = the_5min_interval1$average_steps
```

7. Histogram of the total number of steps taken each day after missing values are imputed

```
steps_per_day_amputdata <- aggregate(activity2$steps, by=list(activity2$date), na.rm=TRUE, sum)

# Assigning column names
colnames(steps_per_day_amputdata) <- c("date", "total_steps" )
hist(steps_per_day_amputdata$total_steps, bin=50,breaks = 10,  xlab="Number of steps",ylab="Number of da
```



8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

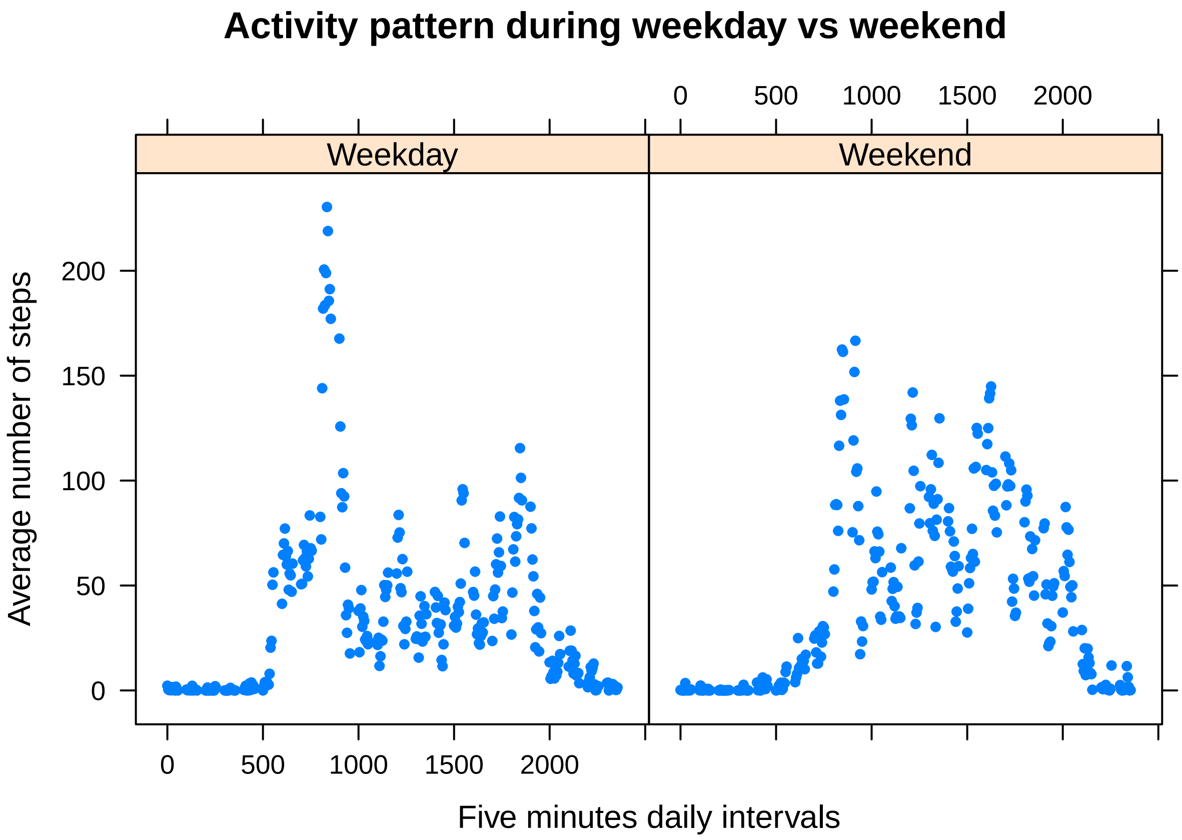
```
activity3<-activity2
activity3$date <- as.Date(activity3$date)

activity4<-activity3%>%
  mutate(day= ifelse(weekdays(activity3$date)=="Saturday" | weekdays(activity3$date)=="Sunday", "Weekend

daily_pattern<-activity4 %>%
  group_by(day,interval) %>%
  summarize(daily_steps=mean(steps))


library(lattice)
```

```
with(daily_pattern,
     xyplot(daily_steps ~ interval  | day, type = "p", colors="blue", pch=20,
            main = "Activity pattern during weekday vs weekend",
            xlab = "Five minutes daily intervals", ylab = "Average number of steps"))
```


Activity pattern during weekday vs weekend
Weekday
Weekend
Average number of steps
Five minutes daily intervals