Analysis of Activity from a Personal Activity Device

Instructions

It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the “quantified self” movement – a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data. This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

This assignment will be described in multiple parts. You will need to write a report that answers the questions detailed below. Ultimately, you will need to complete the entire assignment in a single R markdown document that can be processed by knitr and be transformed into an HTML file. For the plotting aspects of this assignment, feel free to use any plotting system in R (i.e., base, lattice, ggplot2)

Fork/clone the GitHub repository (http://github.com/rdpeng/RepData_PeerAssessment1) created for this assignment. You will submit this assignment by pushing your completed files into your forked repository on GitHub. The assignment submission will consist of the URL to your GitHub repository and the SHA-1 commit ID for your repository state. NOTE: The GitHub repository also contains the dataset for the assignment so you do not have to download the data separately.

DATA

The data for this assignment can be downloaded from here (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip)

The variables included in this dataset are:

- steps: Number of steps taking in a 5-minute interval (missing values are coded as `NA`)
- date: The date on which the measurement was taken in YYYY-MM-DD format
- 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.

Review Criteria

1.Repo

- Valid GitHub URL
- At least one commit beyond the original fork

- Valid SHA-1
- SHA-1 corresponds to a specific commit

2.Commit containing full submission

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

Loading and preprocessing the data

```
library(tidyverse)
library(plyr)
library(visdat)
library(ggthemes)
library(skimr)
```

Data Analysis

The following code downloads uploads the file, process and transforms the data. The Lubridate package can also covert dates.

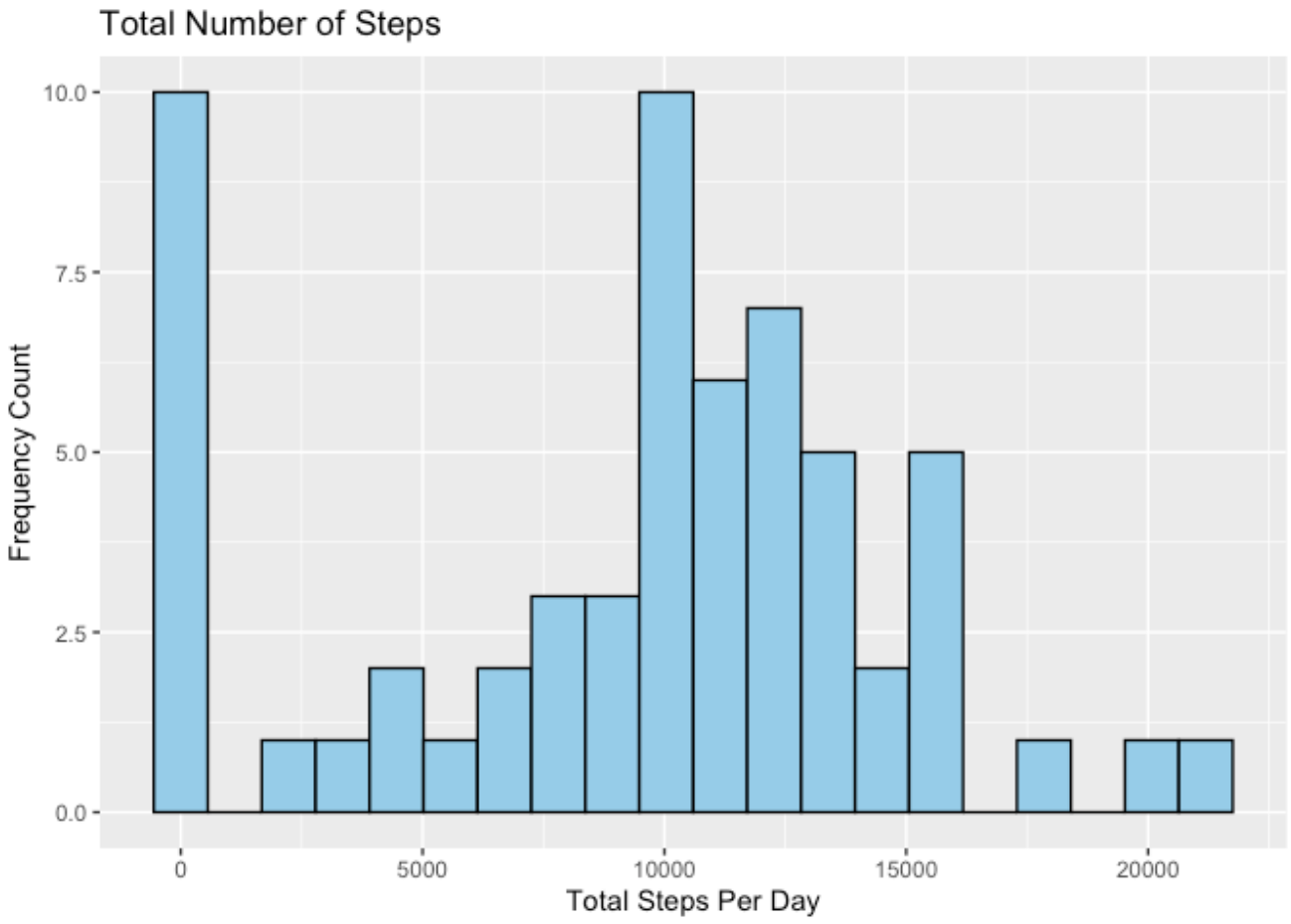
```
data <- read.csv("activity.csv", stringsAsFactors = FALSE)
data$date <- as.Date(data$date, "%Y-%m-%d")

data <- as.data.frame(data)
data1 <- data
```

Calculating the total number of steps. (Group_by can also be used for this specific task)

```
Q1 <- ddply(data, "date", summarise,
            nsteps = sum(steps, na.rm = TRUE))
```

Plotting Histogram with ggplot

```
ggplot(data = Q1, aes(x = nsteps)) +
  geom_histogram(fill = "skyblue", col="black",bins = 20) +
  xlab("Total Steps Per Day") +
  ylab("Frequency Count") +
  ggtitle("Total Number of Steps")
```



What is mean total number of steps taken per day?

Calculating the mean and median. Skimr provides a lot of useful information during data processing.

```
skim(data)
```

Data summary

Name	data
Number of rows	17568
Number of columns	3
_____	
Column type frequency:	
Date	1

numeric	2
_____	
Group variables	None

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date	0	1	2012-10-01	2012-11-30	2012-10-31	61

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
steps	2304	0.87	37.38	112.00	0	0.00	0.0	12.00	806	▇▁▁▁▁
interval	0	1.00	1177.50	692.45	0	588.75	1177.5	1766.25	2355	▇▇▇▇▇

```
mean(Q1$nsteps)
```

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

```
median(Q1$nsteps)
```

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

What is the average daily activity pattern?

Tapply allows for the means for each interval to be calculated. I would prefer to use group_by also works but it seemed to be a be erratic as is continued with this assignment.

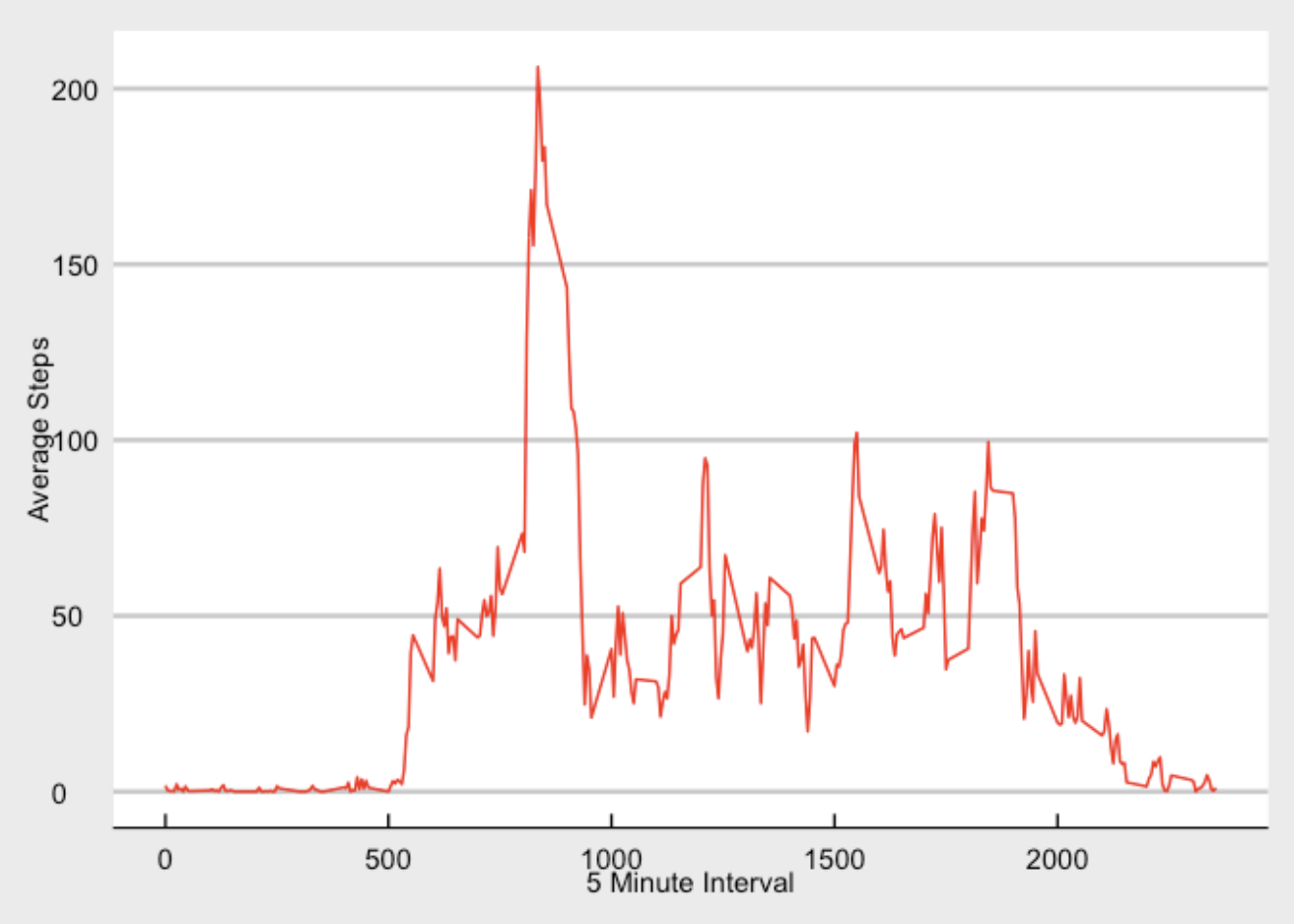
```
means <- with(na.omit(data), tapply(steps, interval, mean))
head(means, 5)
```

```
##         0         5        10        15        20
## 1.7169811 0.3396226 0.1320755 0.1509434 0.0754717
```

Average daily activity pattersn

The following code creates a time series plot where y (daily average steps) per x (5 minute intervals).

```
Q2 <- ddply(data, "interval", summarise,
            nSteps = mean(steps, na.rm = TRUE))

ggplot(data = Q2, aes(x= interval, y = nSteps)) +
  geom_line(color = "red") +
  xlab("5 Minute Interval")+
  ylab("Average Steps") +
  theme_economist_white()
```




```
means[which(means == max(means))]
```

```
##      835
## 206.1698
```

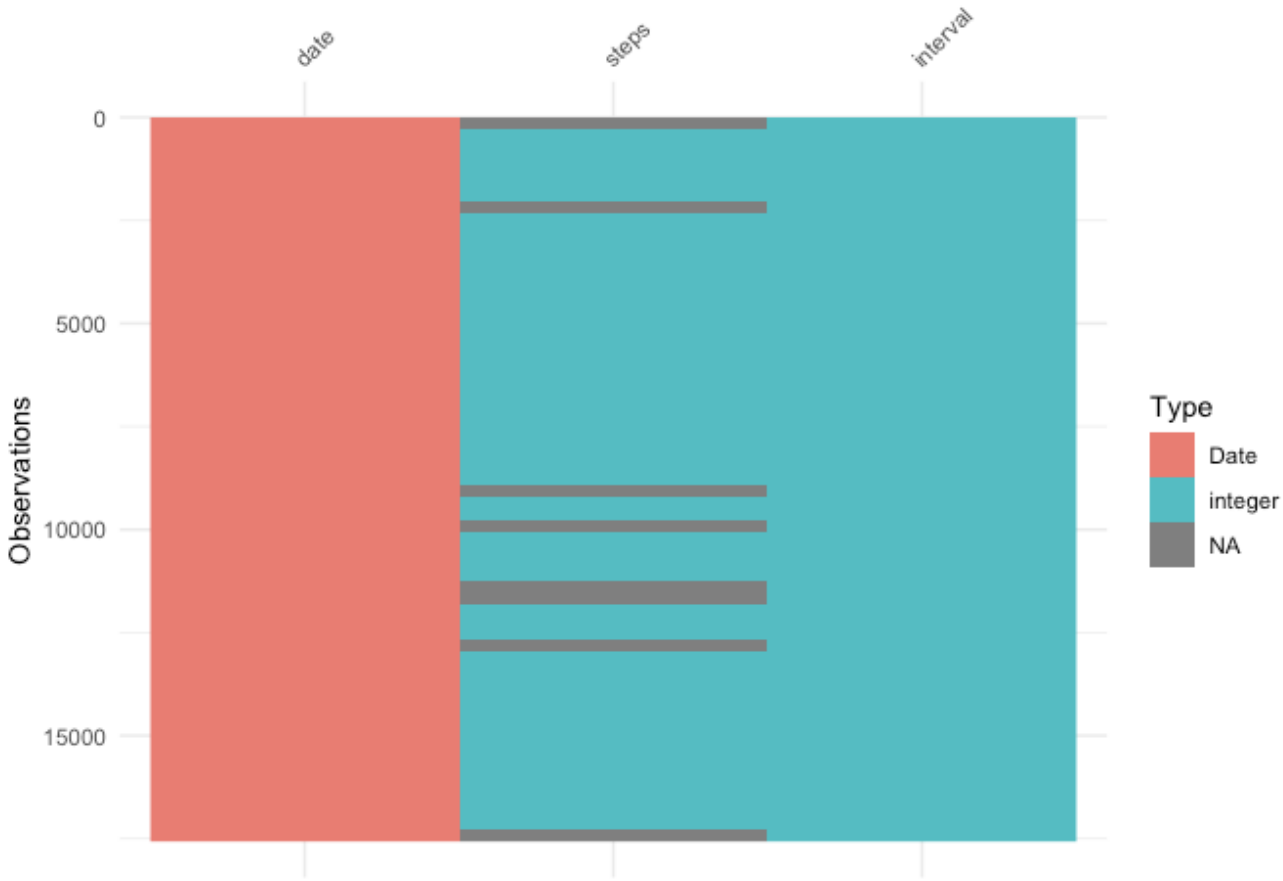
Imputing missing values

Visdat and skimr are helpful in visualizing the type of variable and if missing data is present.

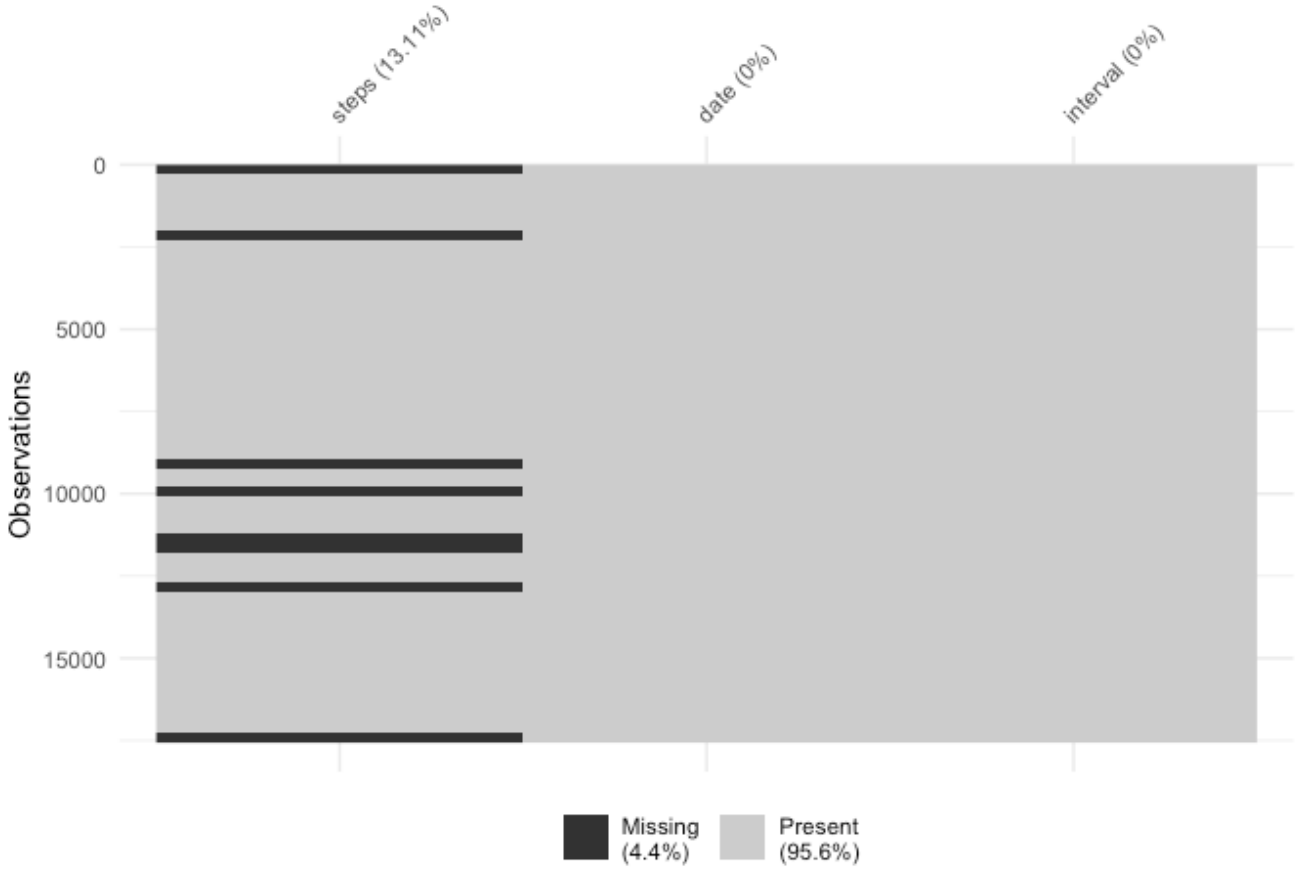
```
sum(is.na(data))
```

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

```
vis_dat(data)
```




```
vis_miss(data)
```




```
skim(data)
```

Data summary

Name	data
Number of rows	17568
Number of columns	3
_____	
Column type frequency:	
Date	1
numeric	2
_____	
Group variables	None

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date	0	1	2012-10-01	2012-11-30	2012-10-31	61

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
steps	2304	0.87	37.38	112.00	0	0.00	0.0	12.00	806	▇▁▁▁▁
interval	0	1.00	1177.50	692.45	0	588.75	1177.5	1766.25	2355	▇▇▇▇▇

```
nrow(data)
```

```
## [1] 17568
```

```
(2304/17568)
```

```
## [1] 0.1311475
```

```
head(data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

```
tail(data)
```

```
##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

Made a copy of the data.

```
data2 <- data
```

The following code served the following purposes:

- intUQ: helped in obtaining the number of unique “interval” values.
- rows: obtaining the rows of missing data.
- NAintr: number of missing values in ther interval column.
- STEPSna: missing values in the steps column.

```
intUQ <- unique(data2$interval)
rows <- nrow(data[is.na(data2), ])

NAintr <- data[is.na(data2), 3]
STEPSna <-data[is.na(data2), 1]
```

###**Creating a function that deals with missing values.**

Dinov's data science textbook (2018:79-81) was usefu in creating a function that deals with missing values.

- j: for missing row values
- i: for missing interval values

```
for (j in 1:2304){       # missing values
  for (i in 1:288) {     # number of intervals
    if(NAintr[j] == intUQ[i]) # if they are equal t
      STEPSna[j] <- means[i] # then replace with the means from the mneans columns
  }
}
```

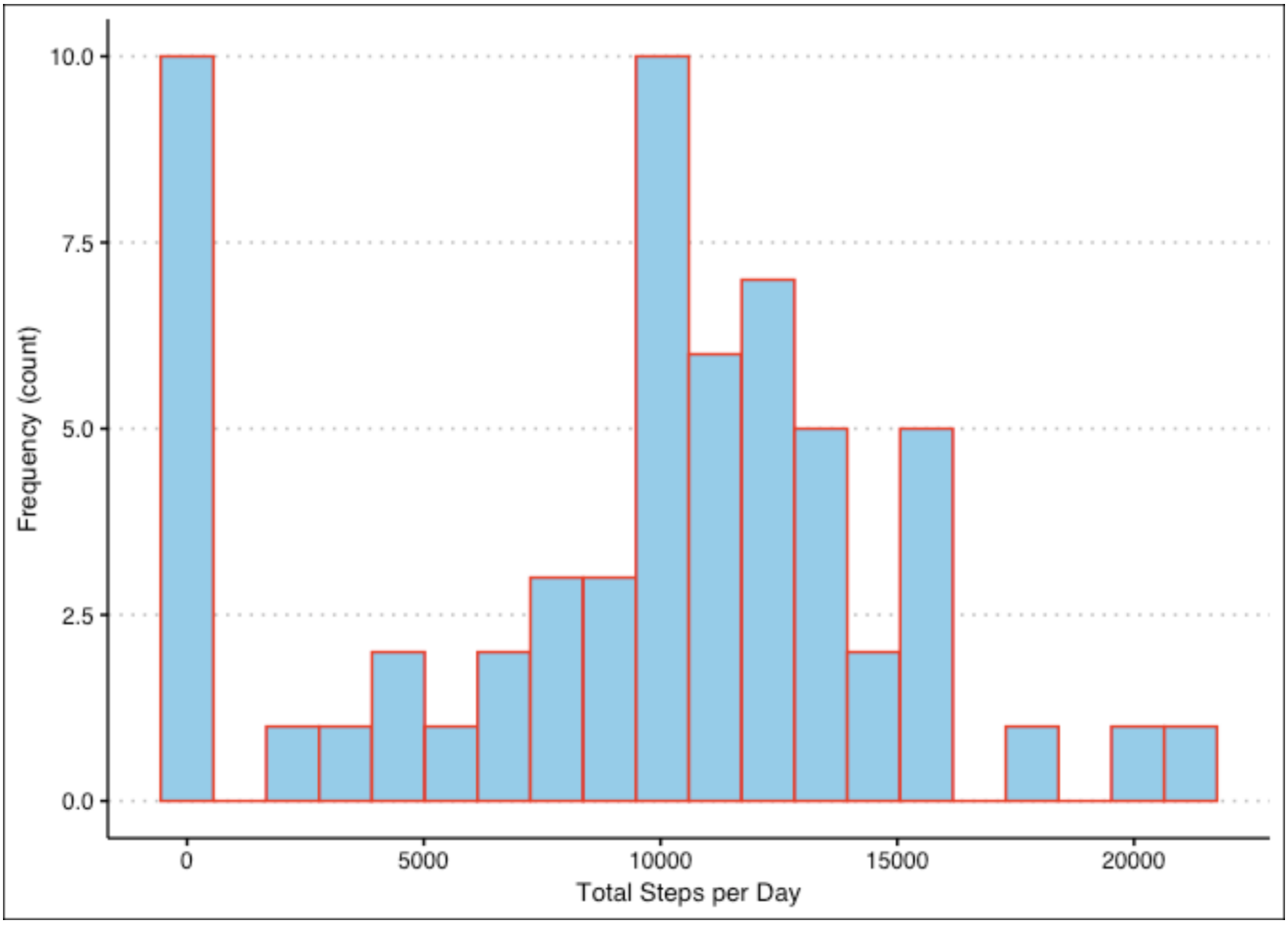
- indX: number of missing values from steps
- Missing steps (using indX as an index) are replaced

```
indX <- is.na(data$steps)
data$steps <- replace(data$steps, indX, STEPSna)
head(data)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

- 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?

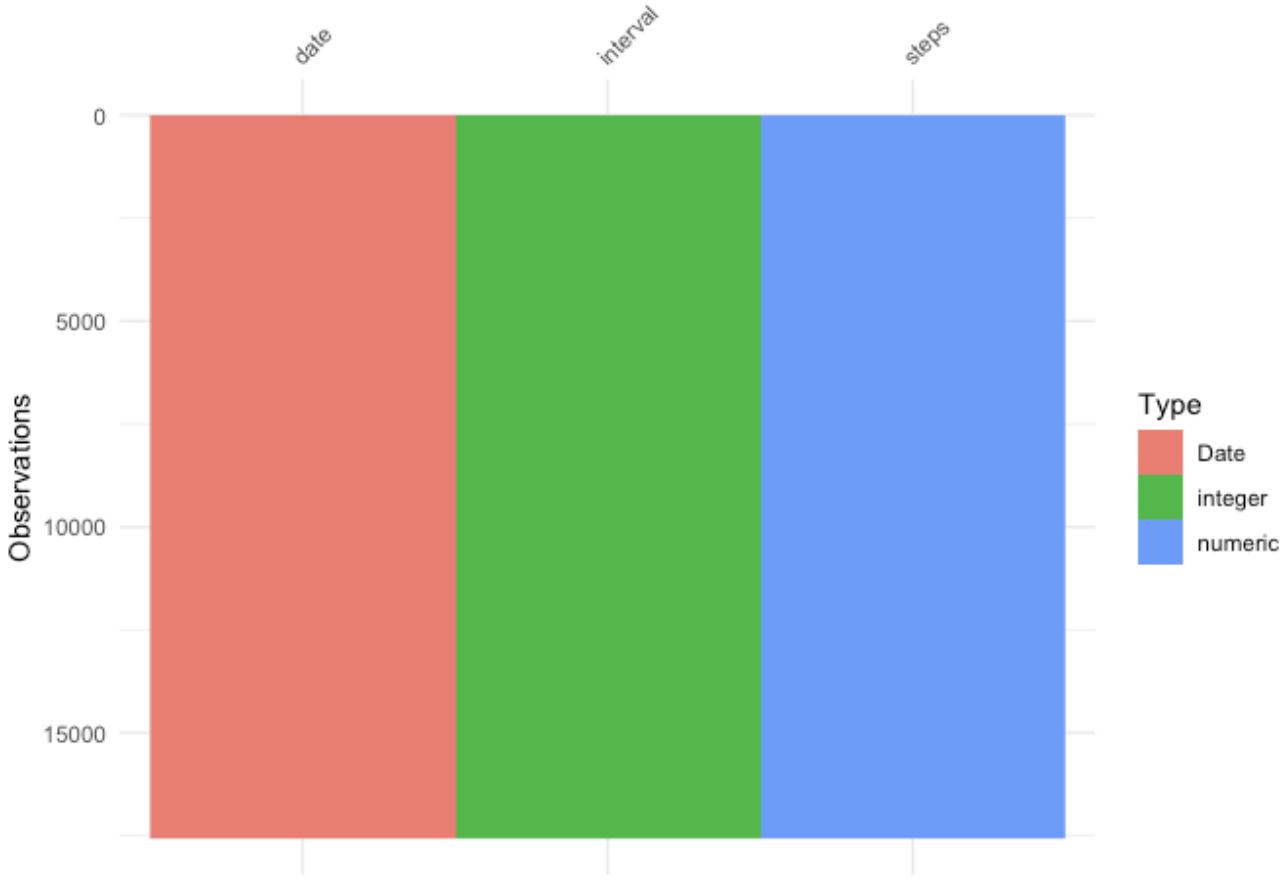
```
Q3 <- ddply(data2, "date", summarise, Nsteps = sum(steps, na.rm = TRUE))

ggplot(data = Q3, aes(x = Nsteps)) +
  geom_histogram(fill = "skyblue", col = "red", bins = 20) +
  xlab("Total Steps per Day") +
  ylab("Frequency (count)") +
  theme_clean()
```



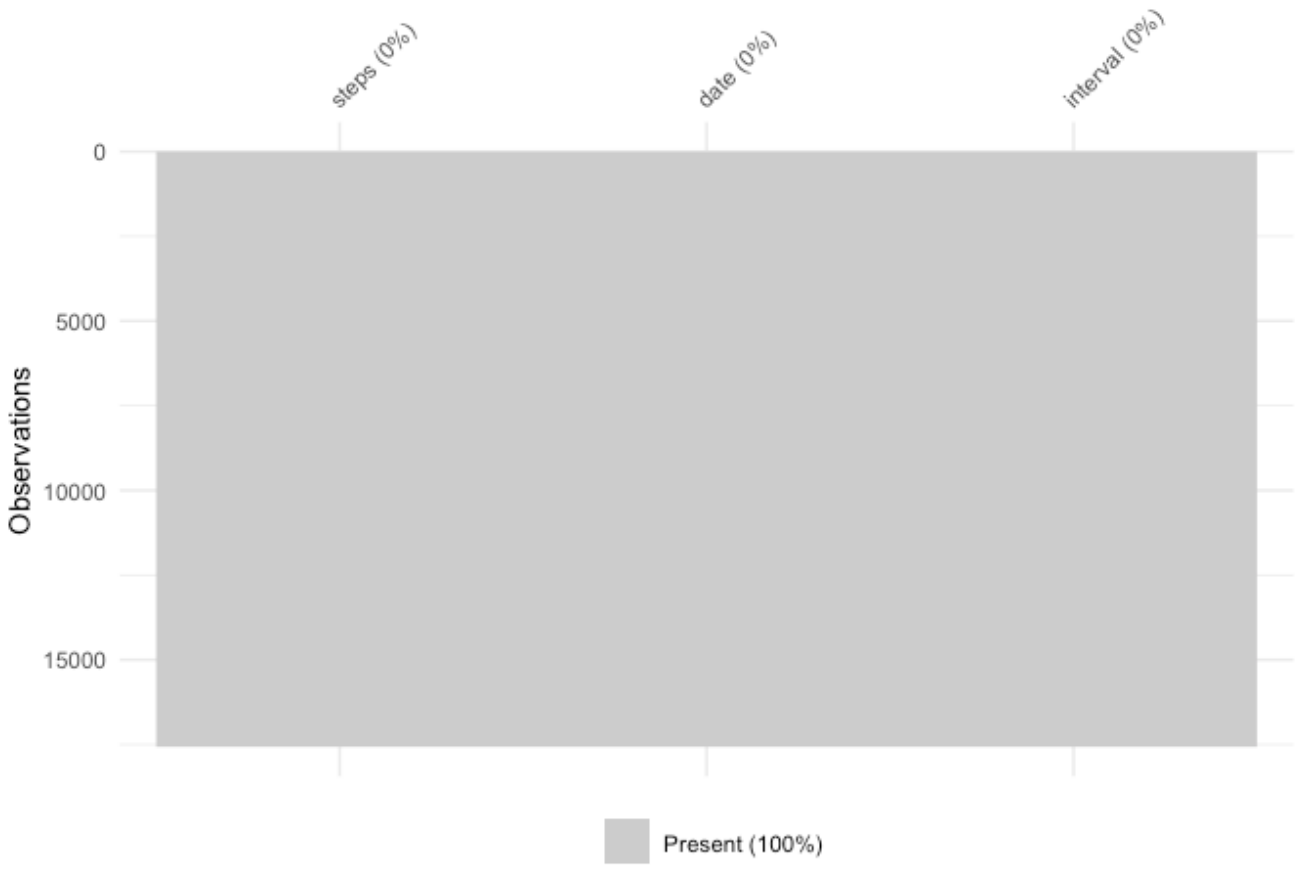
Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

- Calculated steps taken by day.

```
steps <- with(data = data, tapply(steps,date, sum))
vis_dat(data)
```




```
vis_miss(data)
```




```
mean(steps)
```

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

```
median(steps)
```

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

Answer: Yes, the replacement of NA values with interval means. As a result, the mean and median are similar.

Are there differences in activity patterns between weekdays and weekends?

A copy of the data was created.

```
Q3 <- data
Q3 <- mutate(Q3, day = weekdays(Q3$date))
Q3 <- as.data.frame(Q3)
```

This portion of the assignment requires that a new factor variable (“day”) be created with two levels differentiating the week: weekend and weekday.

```
weekdays <- c('Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday')
Q3$day <- factor((weekdays(Q3$date) %in% weekdays),
                 levels=c(FALSE, TRUE), labels=c('Weekend', 'Weekday'))
```

Two subsets of data were created.

```
weekdays <- subset(Q3, day == "Weekday")
weekends <- subset(Q3, day == "Weekend")
```

Before calculating the means, two subsets based on day were created. Although this approach may take a couple of more lines of code, it helped in following the process taken to answer this question.

```
wday <- aggregate(steps~interval, weekdays, mean)
wend <- aggregate(steps~interval, weekends, mean)
```

Merging the two subsets.

- Only the column for means steps during the weekend from the weekend subset was included.

```
days_activity <- cbind(wday,wend$steps)
```

Names were changed to reflect weekend and weekday.

```
names(days_activity)[2] <- "Weekday"
names(days_activity)[3] <- "Weekend"
```

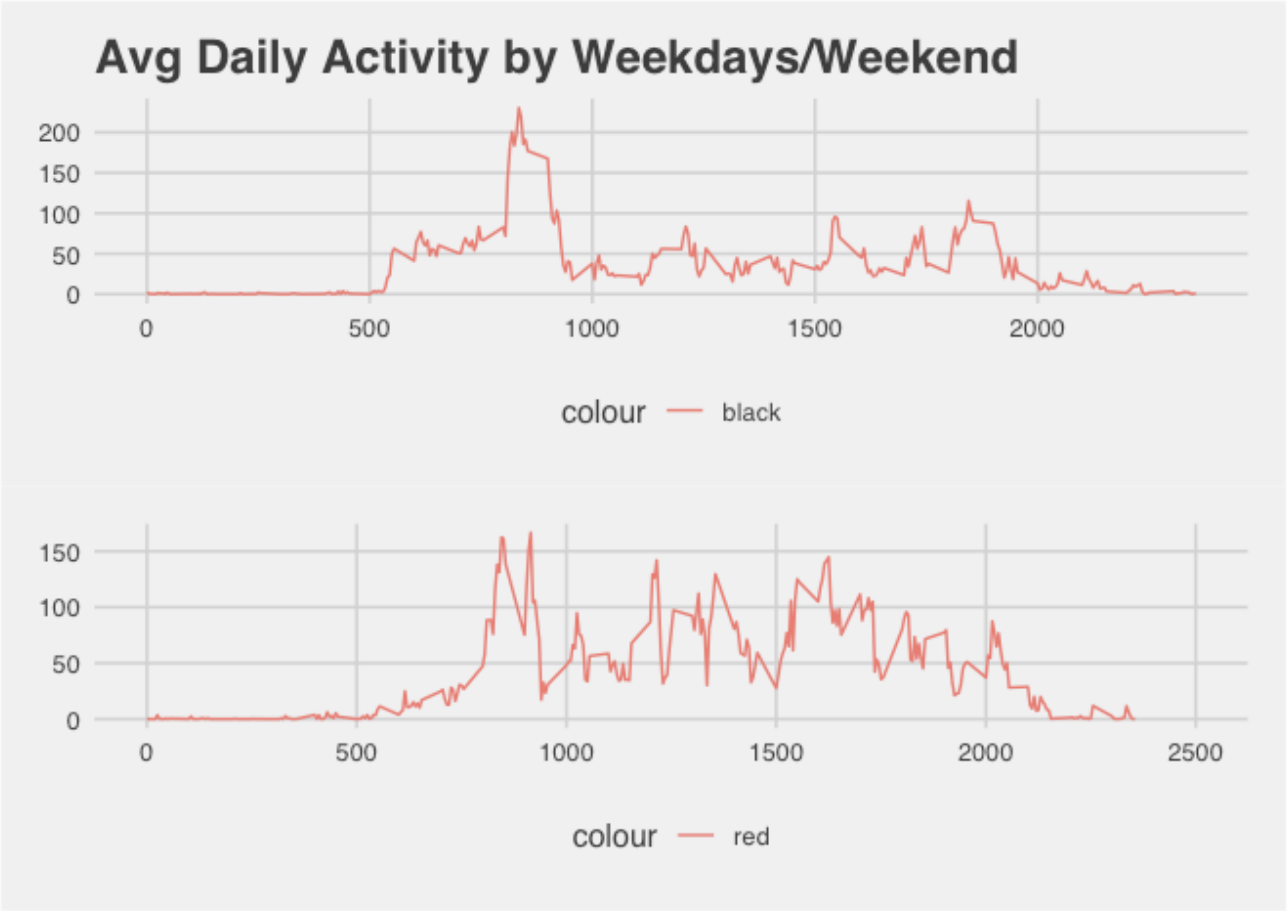
```
p1 <- ggplot(data = days_activity, aes(x = interval, y = Weekday, col = "black")) +
  geom_line() + theme_fivethirtyeight() + ggtitle("Avg Daily Activity by Weekdays/Weeken
d")
```

Patchwork is a ggplot packaged with some added feature. In this case, it made faceting this graph much easier.

```
library(patchwork)
```

```
p2 <- ggplot(data = days_activity, aes(x = interval, y = Weekend, col ="red")) +
  geom_line()+ theme_fivethirtyeight()
```

```
Final <- p1 / p2
Final + xlab("Interval") + ylab("Number of Steps") +
  xlim(0,2500)
```



- Are there differences in activity patterns between weekdays and weekends?

Answer:There is a difference between activity patterns between weekend and weekdays. Evidence shows there is a decrease in average steps per day in the weekends.