output: pdf_document: dev: png

Peer-graded Assignment: Course Project 1

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

Downloading file and loading data frame in working directory

```
Url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(Url, destfile = "./monitoring_data.zip", method = "curl")
unzip("monitoring_data.zip")
activity <- read.table(file = "./activity.csv", header = T, sep = ",")
```

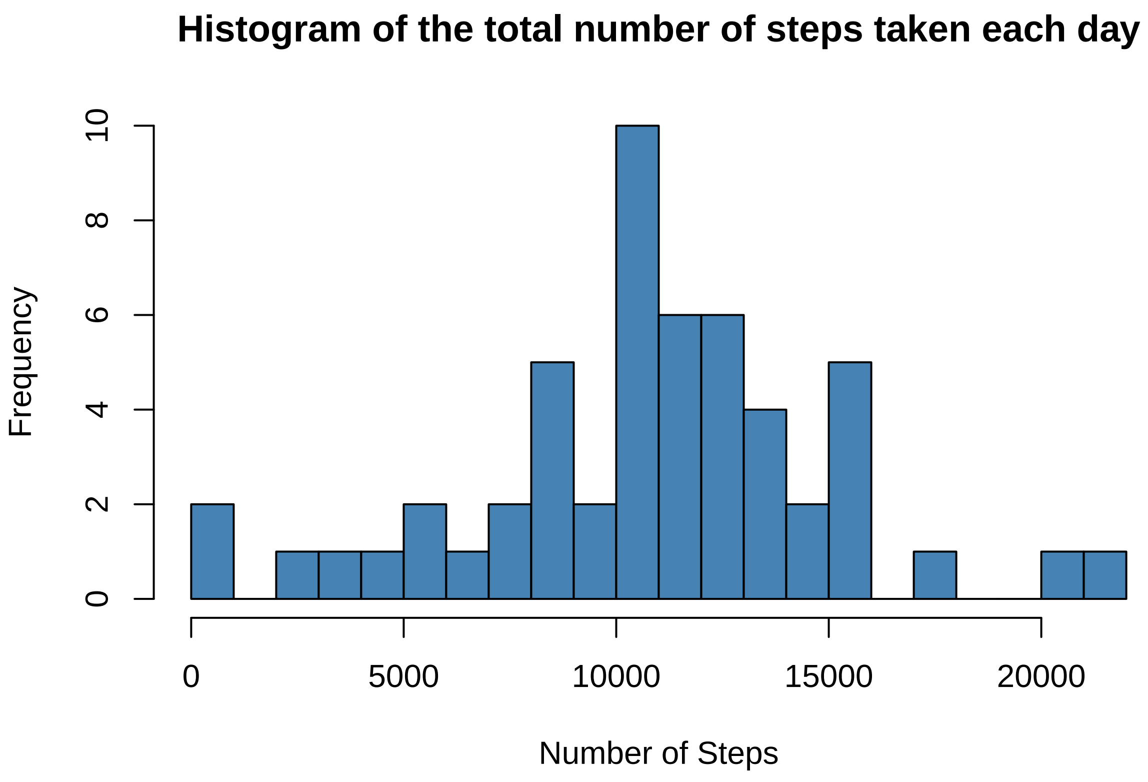
Setting a column with the proper date format and df with non NA

```
activity$date <- ymd(activity$date)
activity2 <- subset(activity, !is.na(activity[,1]))
```

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

Plotting data

```
step_two <- aggregate(data=activity2, steps ~ date, FUN = "sum")
hist(step_two$steps, main = "Histogram of the total number of steps taken each day", col="steelblue", xl
```



3. Mean and median number of steps taken each day

Setting Mean and Median

```
mean(step_two$steps)
```

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

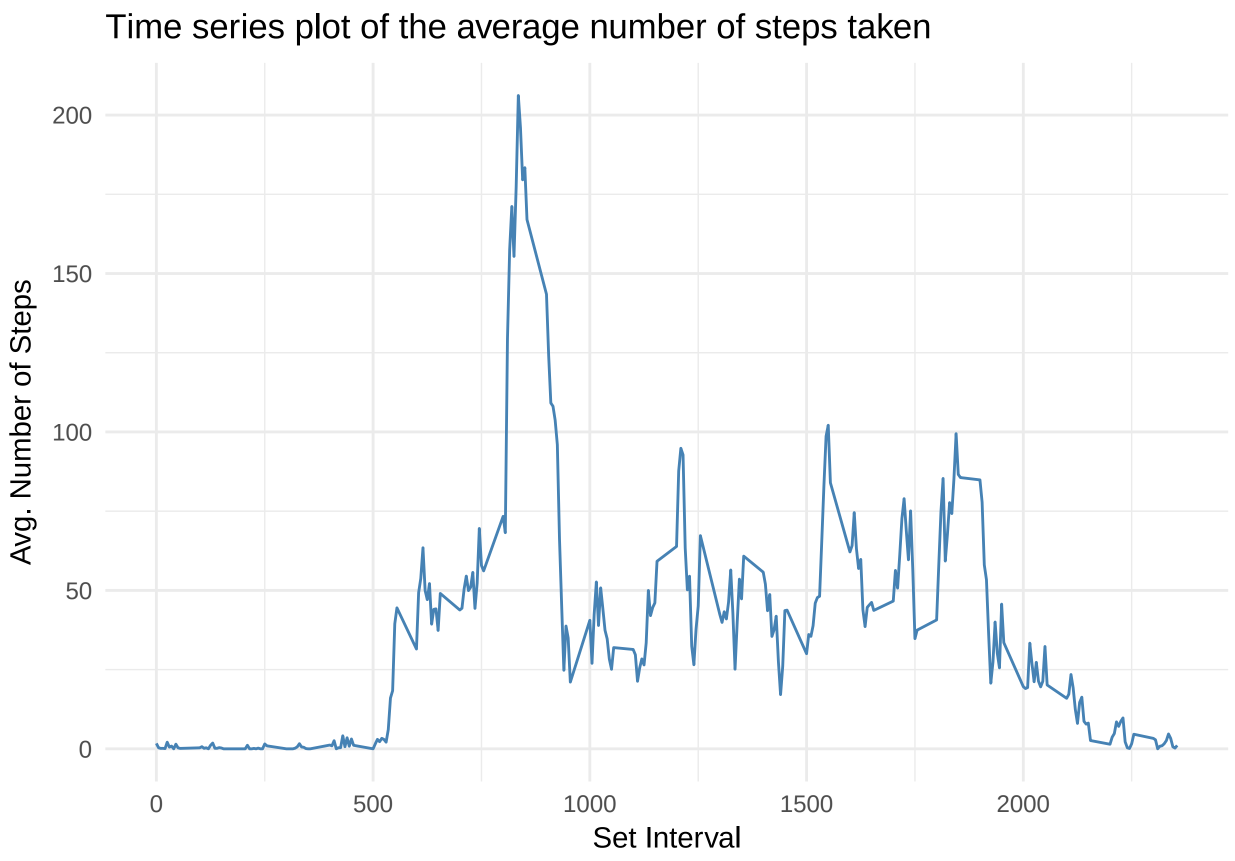
```
median(step_two$steps)
```

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

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

Aggregating data and setting ggplot

```
step_interval <- aggregate(data=activity2, steps ~ interval, FUN = "mean")
ggplot(step_interval, aes(x=interval, y=steps)) + geom_line(color="steelblue", size=0.5) + labs(title="
```



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

Setting code

```
info <- step_interval[which.max(step_interval$steps),]
info1 <- info$interval
info2 <- info$steps
```

The number 835 interval has the most number of steps (max), with 206.1698113 steps

Code to describe and show a strategy for imputing missing data

Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NA

```
sum(is.na(activity))
```

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

2304 missing values exist in the dataset provided

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.

We will proceed to fill in the missing values with the average steps of the available data.

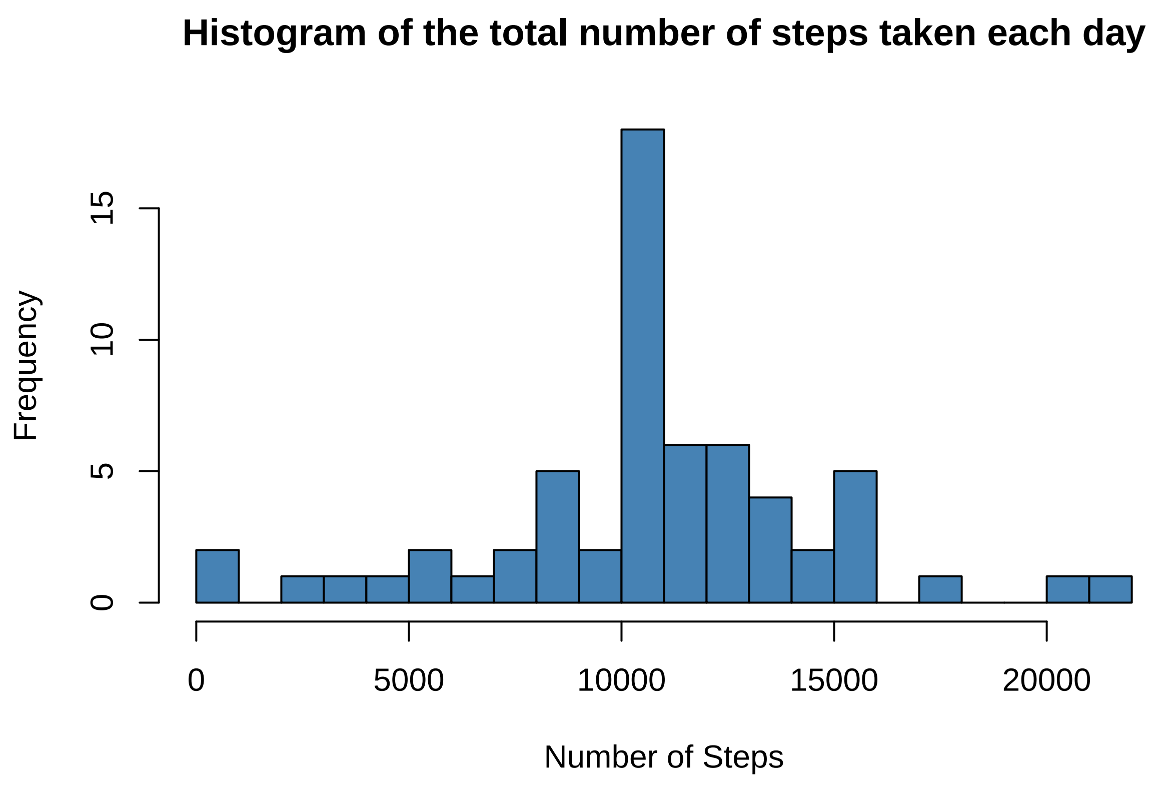
Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
activity3 <- subset(activity, is.na(activity[,1]))
activity4 <- merge(activity3, step_interval, by="interval")
activity5 <- select(activity4, c(4, 3, 1))
colnames(activity5) <- c("steps", "date", "interval")
activity_final <- rbind(activity2,activity5)
sum(is.na(activity_final))
```

```
## [1] 0
```

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

```
step_nine <- aggregate(data=activity_final, steps ~ date, FUN = "sum")
hist(step_nine$steps, main = "Histogram of the total number of steps taken each day", col="steelblue", 
```



```
mean(step_nine$steps)
```

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

```
median(step_nine$steps)
```

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

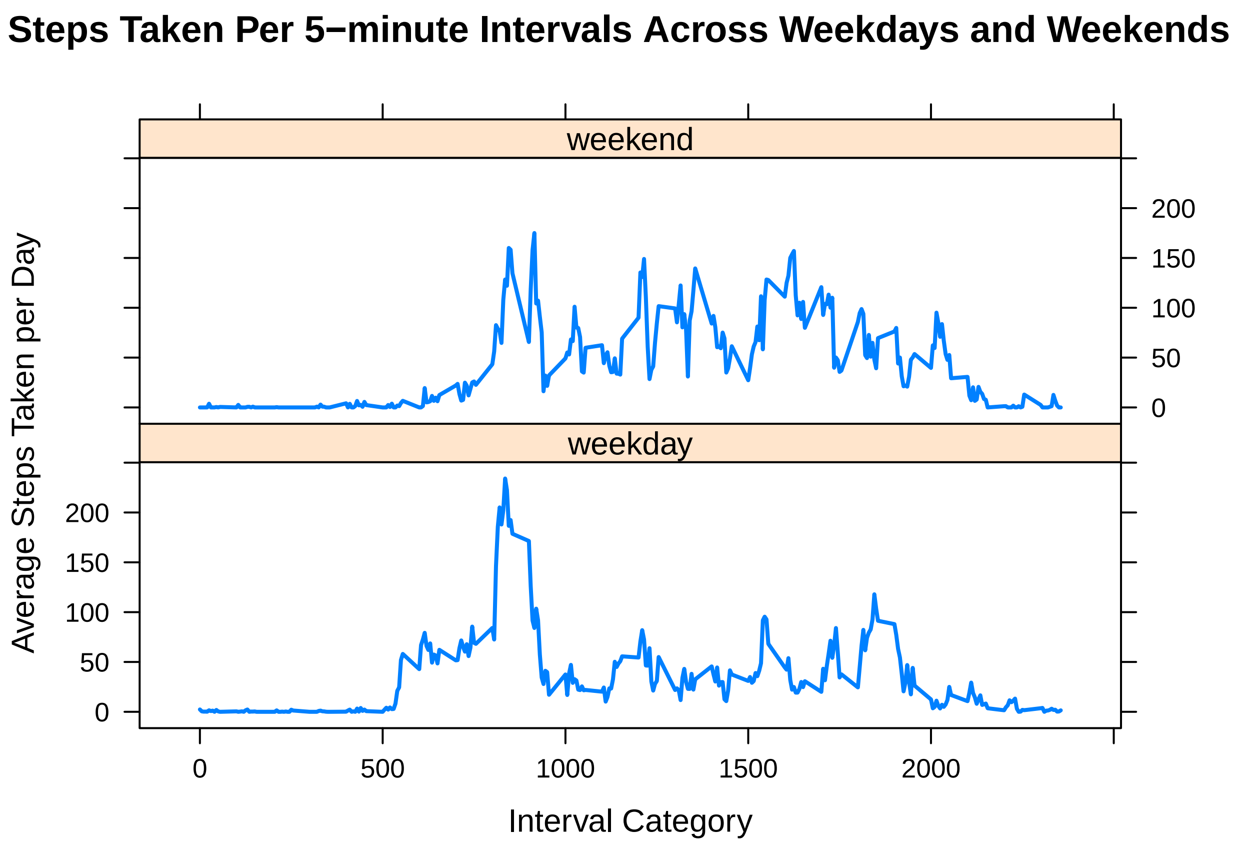
Setting up the non-missing values with average values generates a bias towards normalization, where the median is relatively close or equal to the mean, for each day. So the average remains the same, but the data tends to cluster to the middle, setting the median value closer to the mean.

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

Create a new factor variable in the dataset

```
activity2$week <- weekdays(activity2$date)
activity2$weekcat <- ""
activity2[activity2$week == "Saturday" | activity2$week == "Sunday", ]$weekcat <- "weekend"
activity2[!(activity2$week == "Saturday" | activity2$week == "Sunday"), ]$weekcat <- "weekday"
activity2$weekcat <- factor(activity2$weekcat)
```

```
step_eleven <- aggregate(data=activity2, steps ~ interval + weekcat, FUN=mean)
xyplot(steps ~ interval | weekcat, data = step_eleven, type = "l", lwd = 2,
       layout = c(1, 2),
       xlab = "Interval Category",
       ylab = "Average Steps Taken per Day",
       main = "Steps Taken Per 5-minute Intervals Across Weekdays and Weekends")
```


Steps Taken Per 5−minute Intervals Across Weekdays and Weekends
weekend
weekday
Average Steps Taken per Day
Interval Category
0
50
100
150
200
0
500
1000
1500
2000