Reproducible Research - Project 1

Hemanth P Mohanadas

May 8, 2016

Excercise/Activity Monitoring Data Project

Introduction

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.

The data for this assignment can be downloaded from the course web site:

Dataset: Activity monitoring data [52K] 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.

Questions:

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

Guidelines:

Assignment

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.

Throughout your report make sure you always include the code that you used to generate the output you present. When writing code chunks in the R markdown document, always use echo = TRUE so that someone else will be able to read the code. This assignment will be evaluated via peer assessment so it is essential that your peer evaluators be able to review the code for your analysis.

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 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.

Loading and preprocessing the data

Show any code that is needed to

-Load the data (i.e. read.csv()) -Process/transform the data (if necessary) into a format suitable for your analysis

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1. Calculate the total number of steps taken per day
2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day
3. Calculate and report the mean and median of the total number of steps taken per day

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

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)
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.
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
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?

Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

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.
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.

Submitting the Assignment

To submit the assignment:

1. Commit your completed PA1_template.Rmd file to the master branch of your git repository (you should already be on the master branch unless you created new ones)

2. Commit your PA1_template.md and PA1_template.html files produced by processing your R markdown file with knit2html() function in R (from the knitr package) by running the function from the console.

3. If your document has figures included (it should) then they should have been placed in the figure/ directory by default (unless you overrided the default). Add and commit the figure/ directory to your git repository so that the figures appear in the markdown file when it displays on github.

4. Push your master branch to GitHub.

5. Submit the URL to your GitHub repository for this assignment on the course web site. In addition to submitting the URL for your GitHub repository, you will need to submit the 40 character SHA-1 hash (as string of numbers from 0-9 and letters from a-f) that identifies the repository commit that contains the version of the files you want to submit. You can do this in GitHub by doing the following

6. Going to your GitHub repository web page for this assignment

7. Click on the "?? commits" link where ?? is the number of commits you have in the repository. For example, if you made a total of 10 commits to this repository, the link should say "10 commits".

8. You will see a list of commits that you have made to this repository. The most recent commit is at the very top. If this represents the version of the files you want to submit, then just click the "copy to clipboard" button on the right hand side that should appear when you hover over the SHA-1 hash. Paste this SHA-1 hash into the course web site when you submit your assignment. If you don't want to use the most recent commit, then go down and find the commit you want and copy the SHA-1 hash.

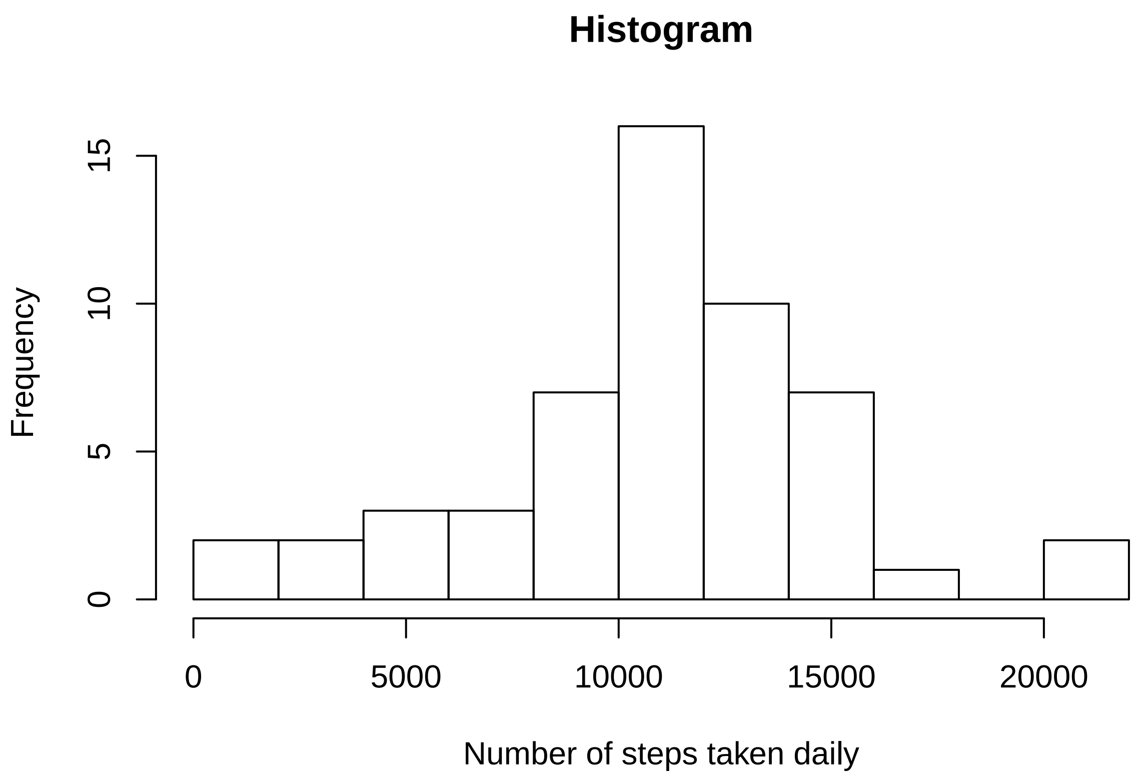
Question 1: Code for reading in the dataset and/or processing the data

```r
activity <- read.csv("C:/Personal/coursera course/Course 5 - Reproducible Research/Project 1/repdata-da
View(activity)
```

The data is loaded into Rstudio using read.csv function. Not much processing is done on the data even though the data had missing values, as we have to work on the missing values later half of the project.

Question 2: Histogram of the total number of steps taken each day

```
daily_steps<-aggregate(steps~date , data=activity, FUN=sum)
hist(daily_steps$steps, xlab="Number of steps taken daily", main="Histogram", breaks=10)
```



Question 3: Mean and median number of steps taken each day

```
summary(daily_steps$steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    8841   10760   10770   13290   21190
```

```
Median_daily_steps<-median(daily_steps$steps)
Mean_daily_steps<-mean(daily_steps$steps)
paste("The median of the total number of steps per day is" , Median_daily_steps)
```

```
## [1] "The median of the total number of steps per day is 10765"
```

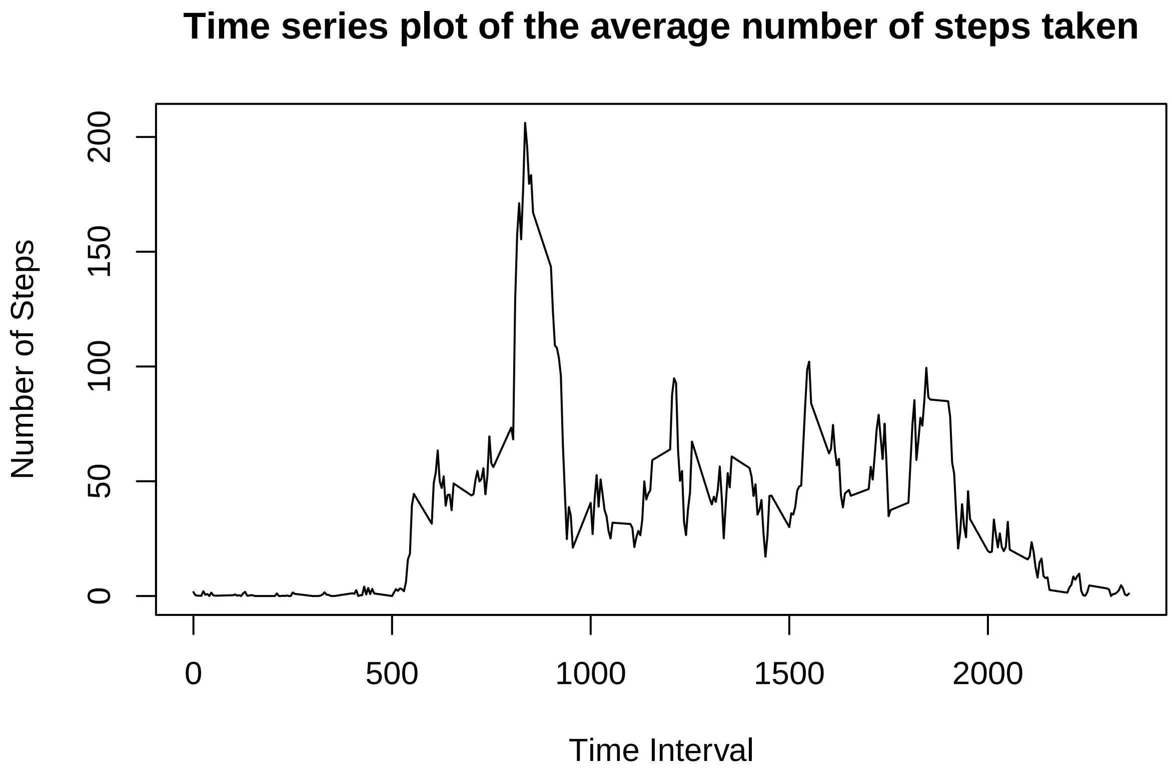
```
paste("The mean of the total number of steps per day is", Mean_daily_steps)
```

```
## [1] "The mean of the total number of steps per day is 10766.1886792453"
```

Shows the median, mean, min, max, 1st Quantile and 3rd Quantile

Question 4: Time series plot of the average number of steps taken

```
interval_steps<-aggregate(steps~interval,data=activity, FUN=mean)
plot(interval_steps$interval,interval_steps$steps, type="l", main="Time series plot of the average numbe
```



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

```
max_steps<-max(interval_steps$steps)

for (i in 1:288)
  {
  if (interval_steps$steps[i]==max_steps)
  interval_with_max_steps<-interval_steps$interval[i]
}
interval_with_max_steps
```

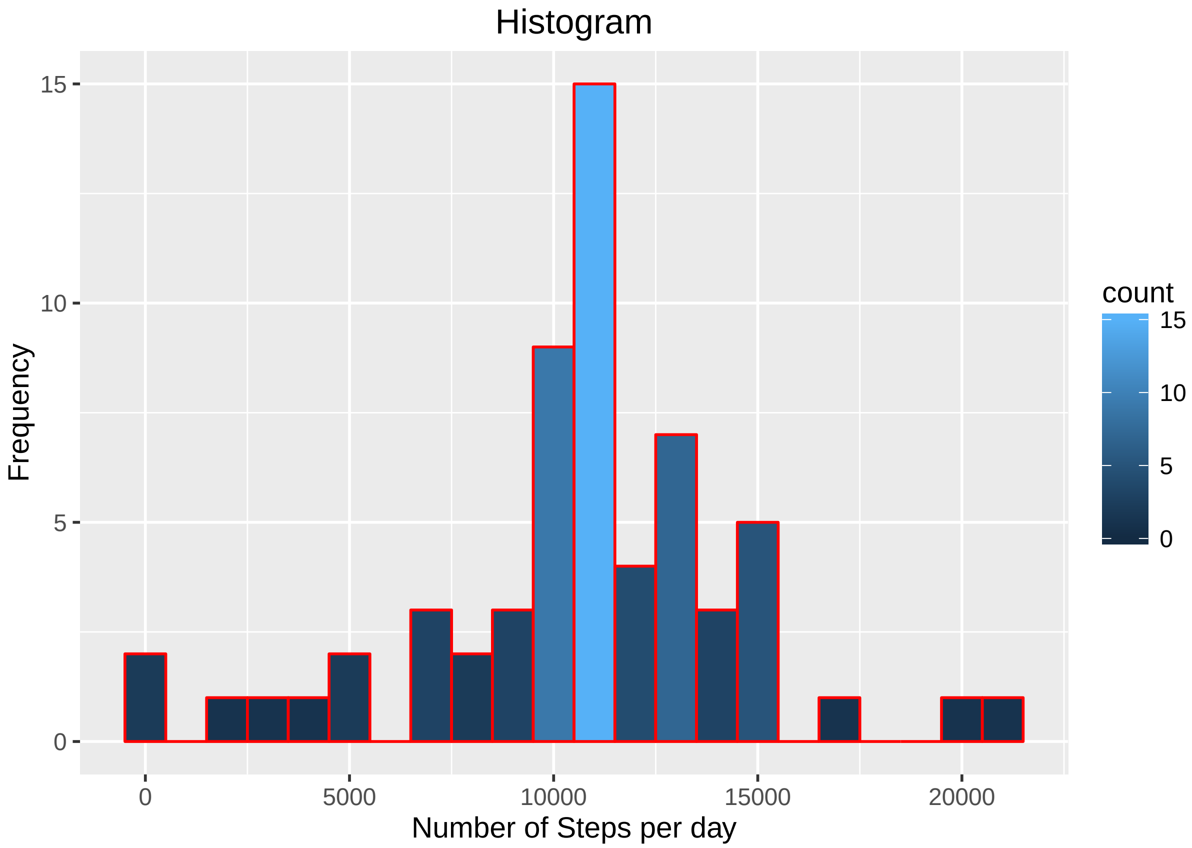
```
## [1] 835
```

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

```
fill_missing_data<-activity ## Create duplicate of data acitivity

for (i in 1:17568) {
  if(is.na(fill_missing_data$steps[i])) { ## Logical form if NA is present/not
    interval<-activity$interval[i]
    for (j in 1:288) {
      if(interval_steps$interval[j]==interval)
        fill_missing_data$steps[i]<-interval_steps$steps[j] ## filling the mean values into the missing
    }
  }
}
```

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

```
## Creating the sum of the steps taken each day
no_missing_value_average_steps<-aggregate(steps~date, data=fill_missing_data, FUN=sum)

## Histogram of the average number of steps after the missing values of the data is filled
library(ggplot2)
ggplot(data=no_missing_value_average_steps, aes(no_missing_value_average_steps$steps)) + geom_histogram
```

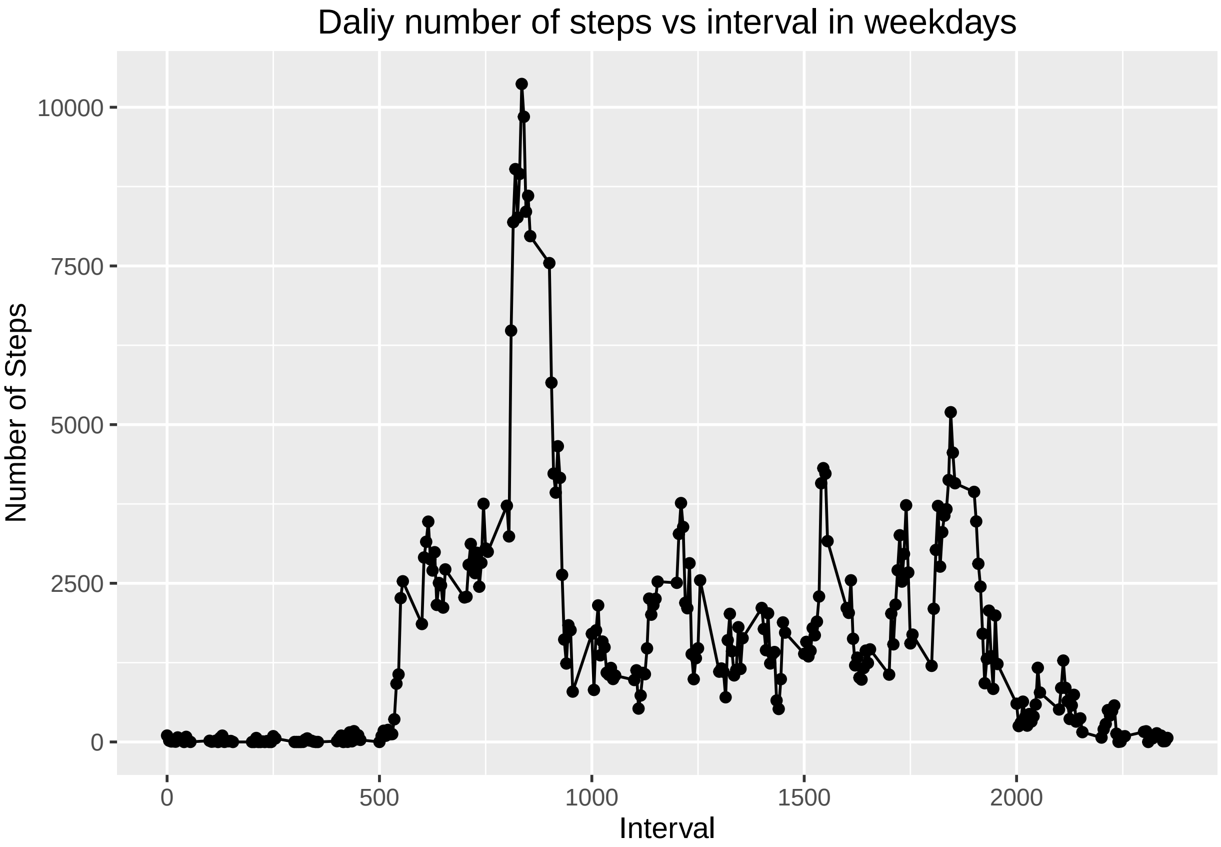


Question 8: Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

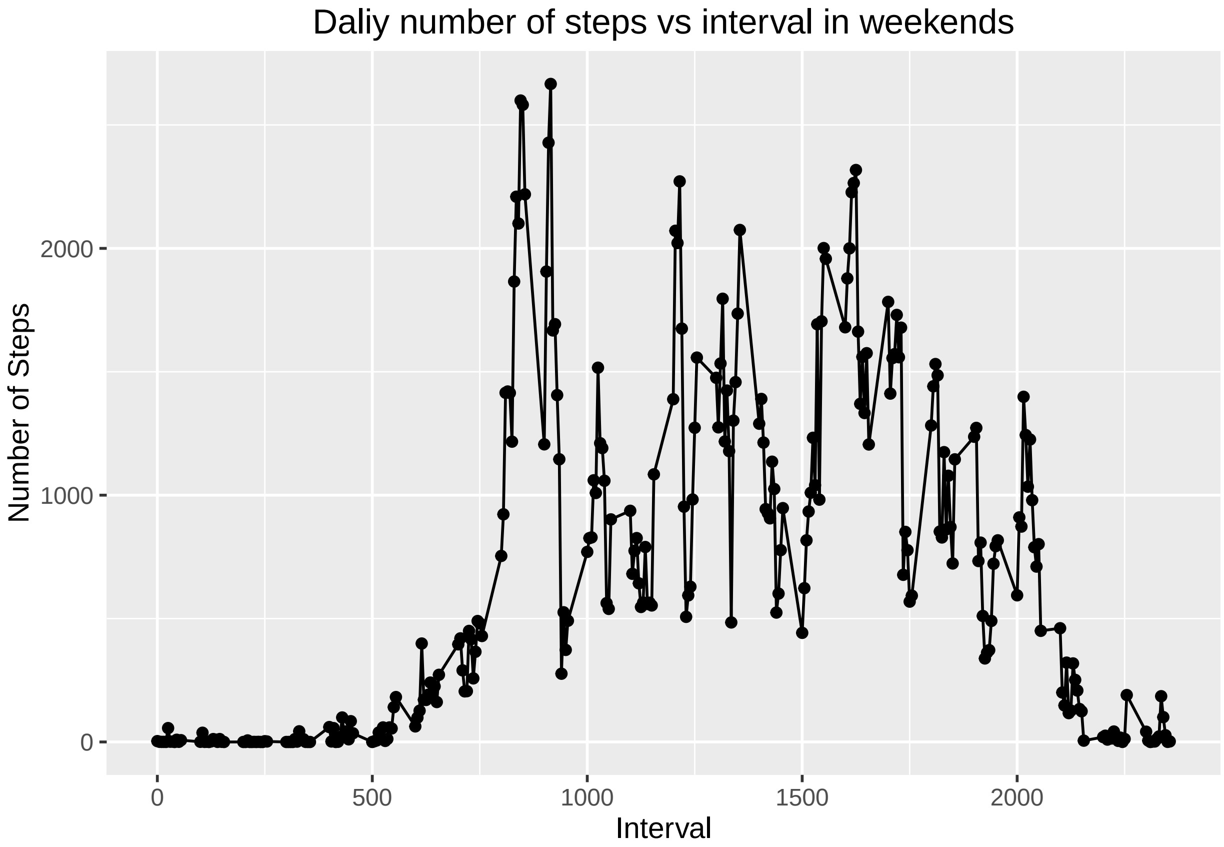
```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
week <- wday(fill_missing_data$date)
week_day <- week
for (i in 1:17568)
{
    if(week[i] == 1)
        week_day[i] <- 'weekend'
    if(week[i] == 2)
        week_day[i] <- 'weekday'
    if(week[i] == 3)
        week_day[i] <- 'weekday'
    if(week[i] == 4)
        week_day[i] <- 'weekday'
    if(week[i] == 5)
        week_day[i] <- 'weekday'
    if(week[i] == 6)
        week_day[i] <- 'weekday'
    if(week[i] == 7)
        week_day[i] <- 'weekend'
}

fill_missing_data$weekday<-week_day ## Creatring a col name weekday with weekdays value

weekday <- grep("weekday",fill_missing_data$weekday) ## finding the word weekday in the data frame
weekday_frame <- fill_missing_data[weekday,] ## creating a frame with only weekdays
weekend_frame <- fill_missing_data[-weekday,] ## creating frame with only weekends

daily_weekdays_steps<-aggregate(steps~interval, data=weekday_frame, FUN=sum)
daily_weekend_steps<-aggregate(steps~interval, data=weekend_frame, FUN=sum)

ggplot(data=daily_weekdays_steps, aes(daily_weekdays_steps$interval, daily_weekdays_steps$steps)) + geom
```



```
ggplot(data=daily_weekend_steps, aes(daily_weekend_steps$interval, daily_weekend_steps$steps)) + geom_po
```


Daliy number of steps vs interval in weekends
Number of Steps
2000
1000
0
Interval
0
500
1000
1500
2000