Reproducible Research Course Project

Somtirtha Roy

February 7th, 2016

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.

Analysis

Read in the data and check for the attributes.

```
library(reshape2)
library(dplyr)
library(ggplot2)

# Code for reading in the dataset and/or processing the data
activity = read.csv("activity.csv")
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

Daily activity pattern

We recast the data attributes form text to data objects so that we can manipulate them for our analysis later.

```
# casting the date attribute in the data as Date objects
activity$date = as.Date(activity$date)
```

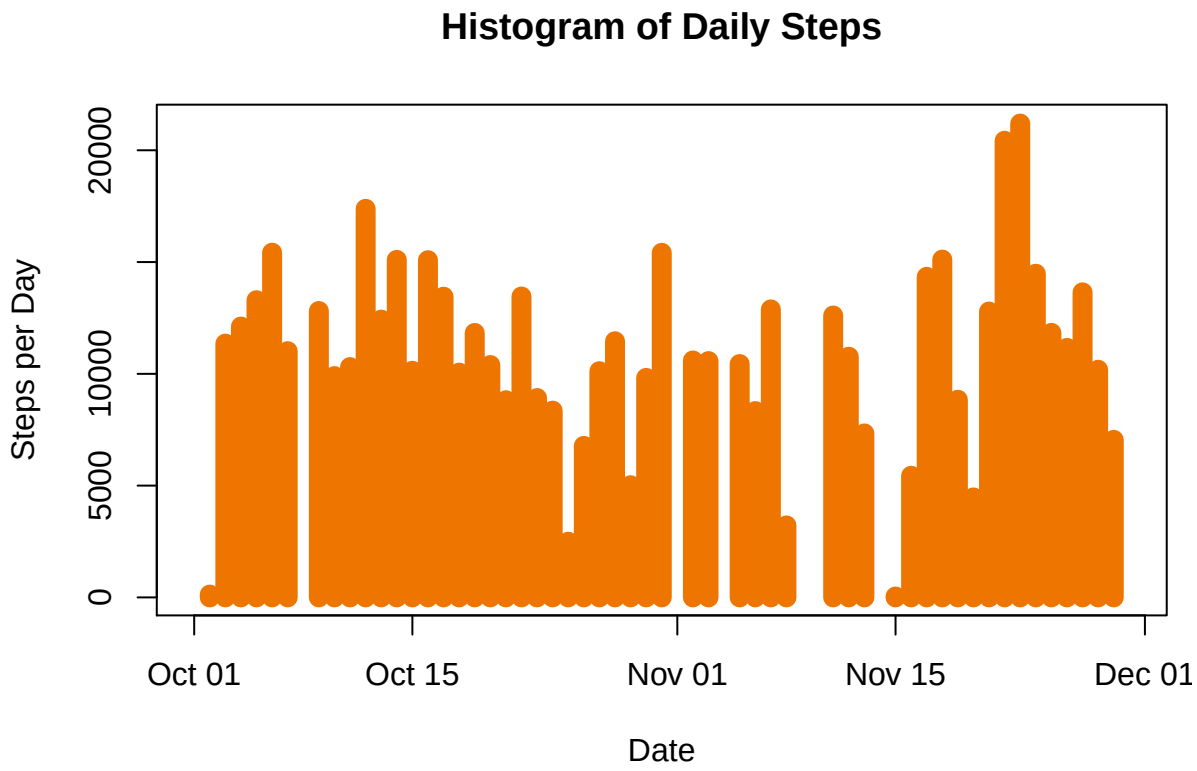
We use the reshape2 package to melt and recast the data to serve our purpose. Here we rehsape the data as all steps belonging to a particular day. This way we can manipulate the data by all steps of a certain day. We cast the melted data by the steps of each date adn sum them up to find the number of steps taken in paricular day.

```
# HISTOGRAM OF TOTAL STEPS ON EACH DAY
# Melt and recast data by date and step on each date
# id=date, measure=steps
activity_date = melt(activity, id.vars="date", measure.vars="steps", na.rm=FALSE)

# Sum all the steps within wach day to get teh total number of steps on each day
activity_sum_date = dcast(activity_date, date ~ variable, sum)
```

We plot the above casted data in a histogram. WE also calculate the mean and median of hte day now that we have the steps walked in the particular day. # Histogram of the total number of steps taken each day

```
plot(activity_sum_date$date, activity_sum_date$steps, type="h", main="Histogram of Daily Steps",
      xlab="Date", ylab="Steps per Day", col="darkorange2", lwd=10)
```




```
# Mean and median number of steps taken each day
mean(activity_sum_date$steps, na.rm=TRUE)
```

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

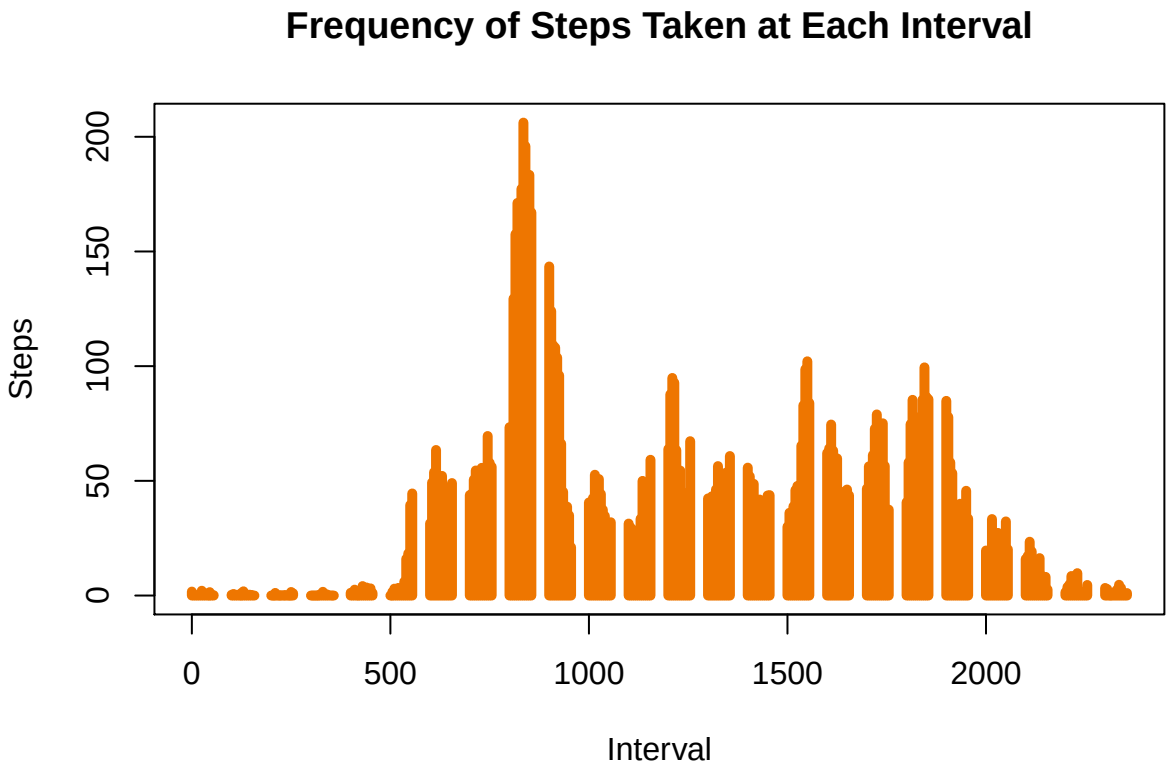
```
median(activity_sum_date$steps, na.rm=TRUE)
```

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

We are trying to determine the average steps taken in each interval across all days and the interval with the maximum average number of steps. As above we melt and cast the data into interval-steps and find the mean number of steps for each interval across all days. We plot this in a histogram to see the pattern of the steps taken in each interval. ## Daily interval pattern

```
# Time series plot of the average number of steps taken
# The 5-minute interval that, on average, contains the maximum number of steps
# Re-melt data frame to prep for casting by interval, including removing NA values so we can take the m
activity_interval = melt(activity, id.vars="interval", measure.vars="steps", na.rm=TRUE)

# Cast data frame to see mean steps per interval
activity_mean_interval = dcast(activity_interval, interval ~ variable, mean)
plot(activity_mean_interval$interval, activity_mean_interval$steps, type="h",
     main="Frequency of Steps Taken at Each Interval", xlab="Interval", ylab="Steps", col="darkorange2"
```



We find the maximum steps taken amonst all intervals.

```
# Interval that has maximum steps
activity_mean_interval$interval[which(activity_mean_interval$steps == max(activity_mean_interval$steps)
```

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

```
max(activity_mean_interval$steps)
```

```
## [1] 206.1698
```

Effect of missing values in data

We check the effect of missing data in this section by replacing the NA values with the mean number of steps.

```r
#IMPUTED MISSING VALUES
# Code to describe and show a strategy for imputing missing data
sum(is.na(activity$steps))
```

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

```r
activity_merge = merge(activity, activity_mean_interval, by="interval", suffixes=c(".act", ".spi"))
```

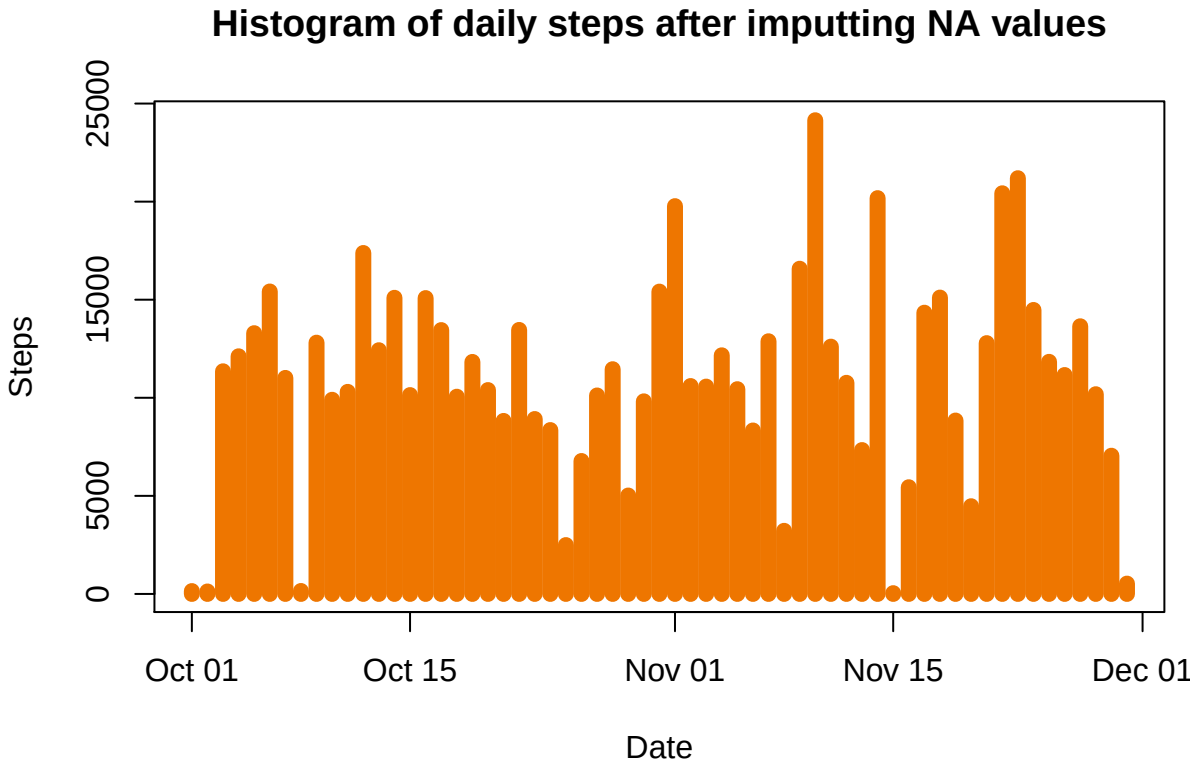
```r
# Get list of indexes where steps value = NA
naIndex = which(is.na(activity$steps))

# Replace NA values with value from steps.spi
activity[naIndex,"steps"] = activity_merge[naIndex,"steps.spi"]

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

activity_date_noNA = melt(activity, id.vars="date", measure.vars="steps", na.rm=FALSE)
activity_sum_noNA = dcast(activity_date_noNA, date ~ variable, sum)

# Plot histogram with frequency of steps by day
plot(activity_sum_noNA$date, activity_sum_noNA$steps, type="h", main="Histogram of daily steps after imp
     xlab="Date", ylab="Steps", col="darkorange2", lwd=8)
```




```r
mean(activity_sum_noNA$steps, na.rm=TRUE)
```

```
## [1] 10889.8
```

```
median(activity_sum_noNA$steps, na.rm=TRUE)
```

```
## [1] 11015
```

Differences in activity patterns between weekdays and weekends

We check the differences in activity of the person on weekdays and weekends.

```
# ACTIVITY PATTERN DURING WEEKDAYS AND WEEKENDS
# Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and week
activity = mutate(activity, weektype = ifelse(weekdays(activity$date) == "Saturday" | weekdays(activity$
activity$weektype = as.factor(activity$weektype)

activity_interval = activity %>% group_by(interval, weektype) %>% summarise(steps = mean(steps))
gp = ggplot(activity_interval, aes(x=interval, y=steps, color = weektype)) +
  geom_line() + facet_wrap(~weektype, ncol = 1, nrow=2)
print(gp)
```