Peer Assignment for Activity

Dipali Bagad

July 26, 2017

R Markdown

This is an R Markdown document. 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.

Loading and preprocessing the data

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(data.table)
```

```
## -------------------------------------------------------------------------
```

```
## data.table + dplyr code now lives in dtplyr.
## Please library(dtplyr)!
```

```
## ------------------------------------------------------------------------
```

```
##
## Attaching package: 'data.table'
```

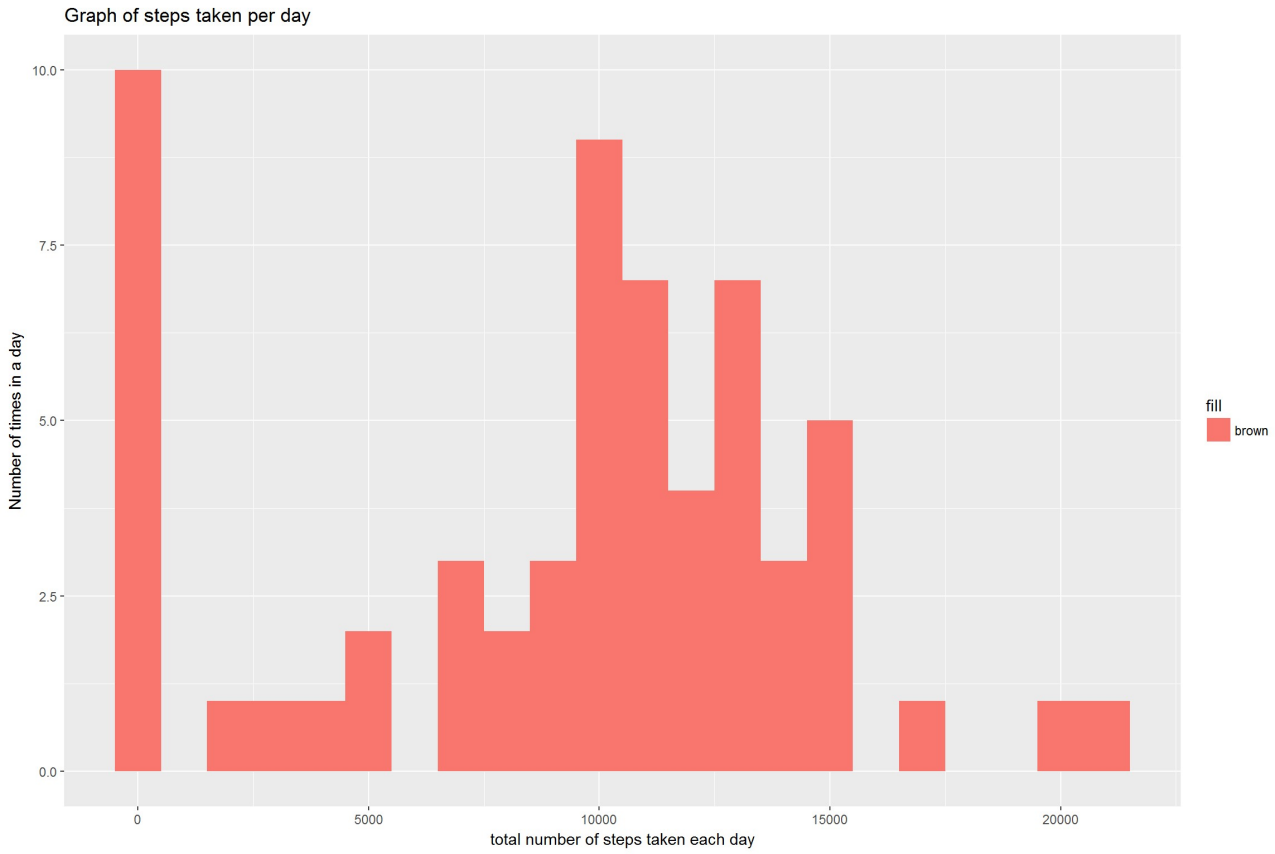
```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(ggplot2)


##Loading and preprocessing the data
activity_data <- read.csv("activity.csv")
#head(activity_data)

activity_data$date <- as.Date(activity_data$date, format = "%Y-%m-%d")
activity_data$interval <- as.factor(activity_data$interval)
#head(activity_data$date)
```

What is mean total number of steps taken per day?

```
##Question 1. What is mean total number of steps taken per day?

totalperday.steps <- tapply(activity_data$steps, activity_data$date, FUN=sum, na.rm=TR
UE)

  knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.path='instructions_fig/')
```

```
qplot(totalperday.steps, binwidth=1000, fill = "brown", main="Graph of steps taken per
day " ,xlab="total number of steps taken each day", ylab="Number of times in a day")
```




```
#calculate mean and median of steps per day
mean(totalperday.steps, na.rm=TRUE)
```

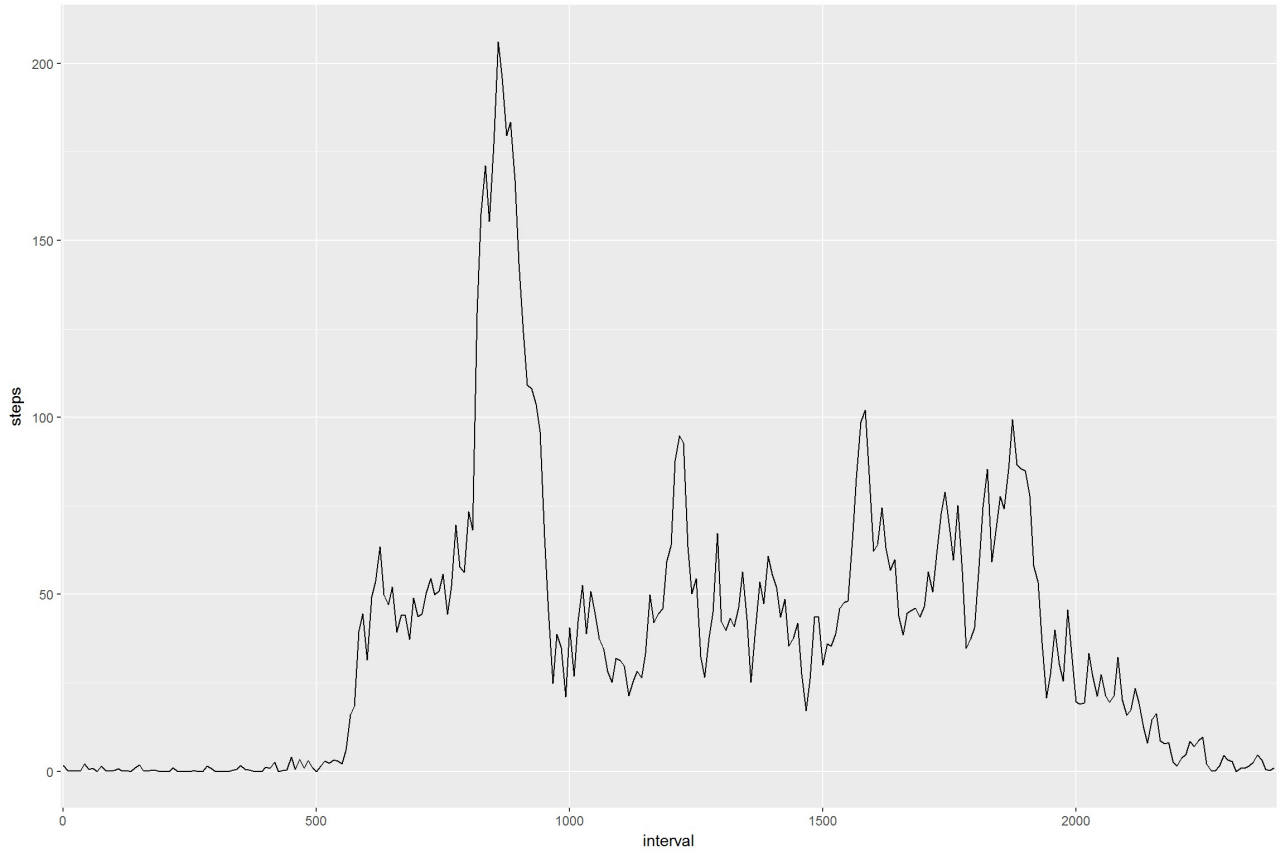
```
## [1] 9354.23
```

```
median(totalperday.steps, na.rm=TRUE)
```

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

What is the average daily activity pattern?

```
##Question 2.What is the average daily activity pattern?

daily_average <- aggregate(x=list(steps=activity_data$steps), by=list(interval=activit
y_data$interval),
                           FUN=mean, na.rm=TRUE)
#draw graphics
ggplot(daily_average, aes(x = interval, y = steps, group = 1)) +
  geom_line() + scale_x_discrete(breaks = seq(0, 3000, 500))
```




```
#calculate max of daily average activity pattern
daily_average[which.max(daily_average$steps),]
```

```
##     interval    steps
## 104      835 206.1698
```

Imputing missing values

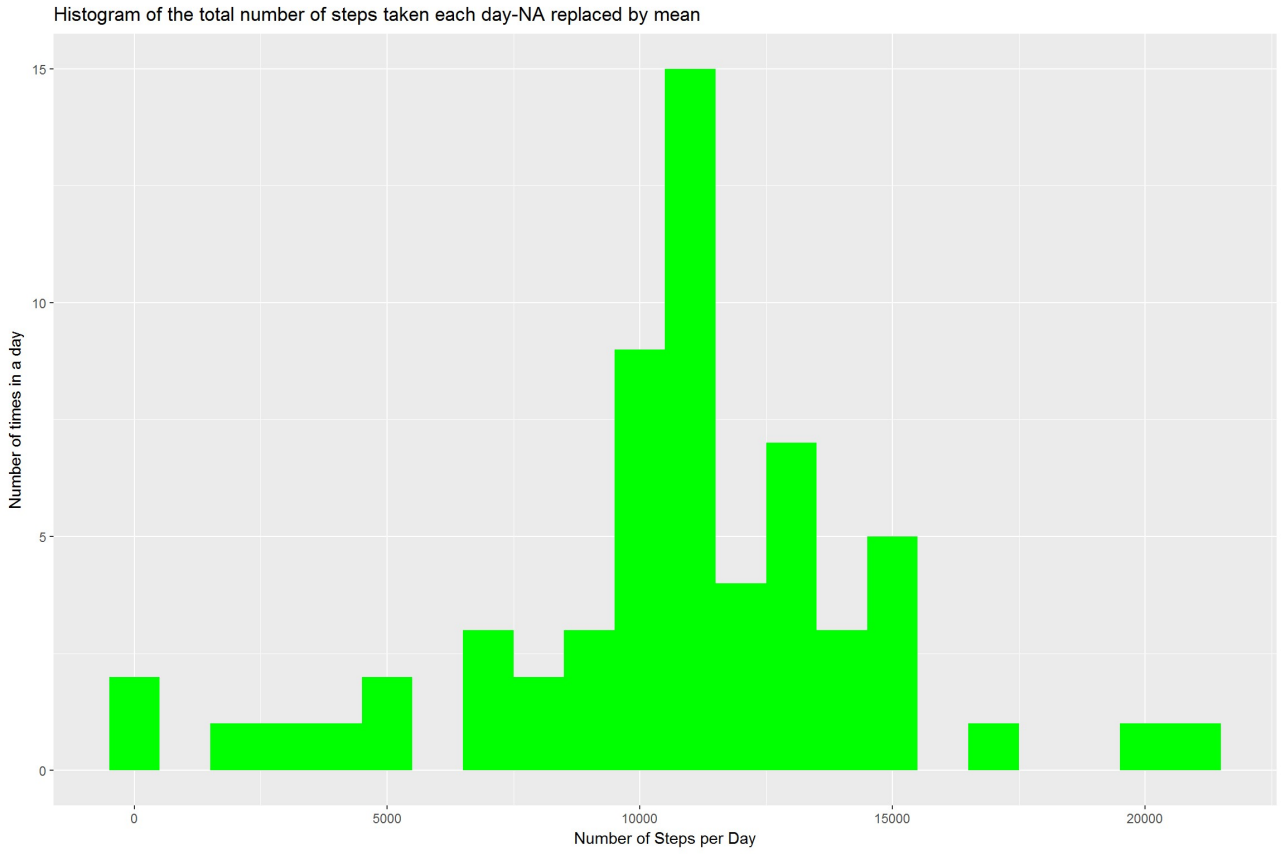
A strategy that is chosen for imputing a missing value is to use the mean value for that 5-minute interval in the dataset.

A new data frame final_sum_data is created and is equal to the original dataset but with the missing data filled in by using the mean value for that interval.

```
## Question 3.Imputing missing values

#number of missing values
missing_values <- sum(is.na(activity_data))
table(missing_values)
```

```
## missing_values
## 2304
##    1
```

```
na_pos <- which(is.na(activity_data$steps))

# Create a vector of means of steps
mean_vector <- rep(mean(activity_data$steps, na.rm=TRUE), times=length(na_pos))

# Replace the NAs by the means vector
activity_data[na_pos, "steps"] <- mean_vector

#aggregate data of steps by each day
final_sum_data <- aggregate(activity_data$steps, by=list(activity_data$date), FUN=sum)

# Rename the attributes
names(final_sum_data) <- c("date", "total")
# draw graphics
ggplot(final_sum_data, aes(x=total)) +
  geom_histogram(fill = "green", binwidth = 1000) +
  labs(title="Histogram of the total number of steps taken each day-NA replaced by mea
n",
       x = "Number of Steps per Day", y = "Number of times in a day")
```



```
#calculate mean and median for final processed data
mean(final_sum_data$total)
```

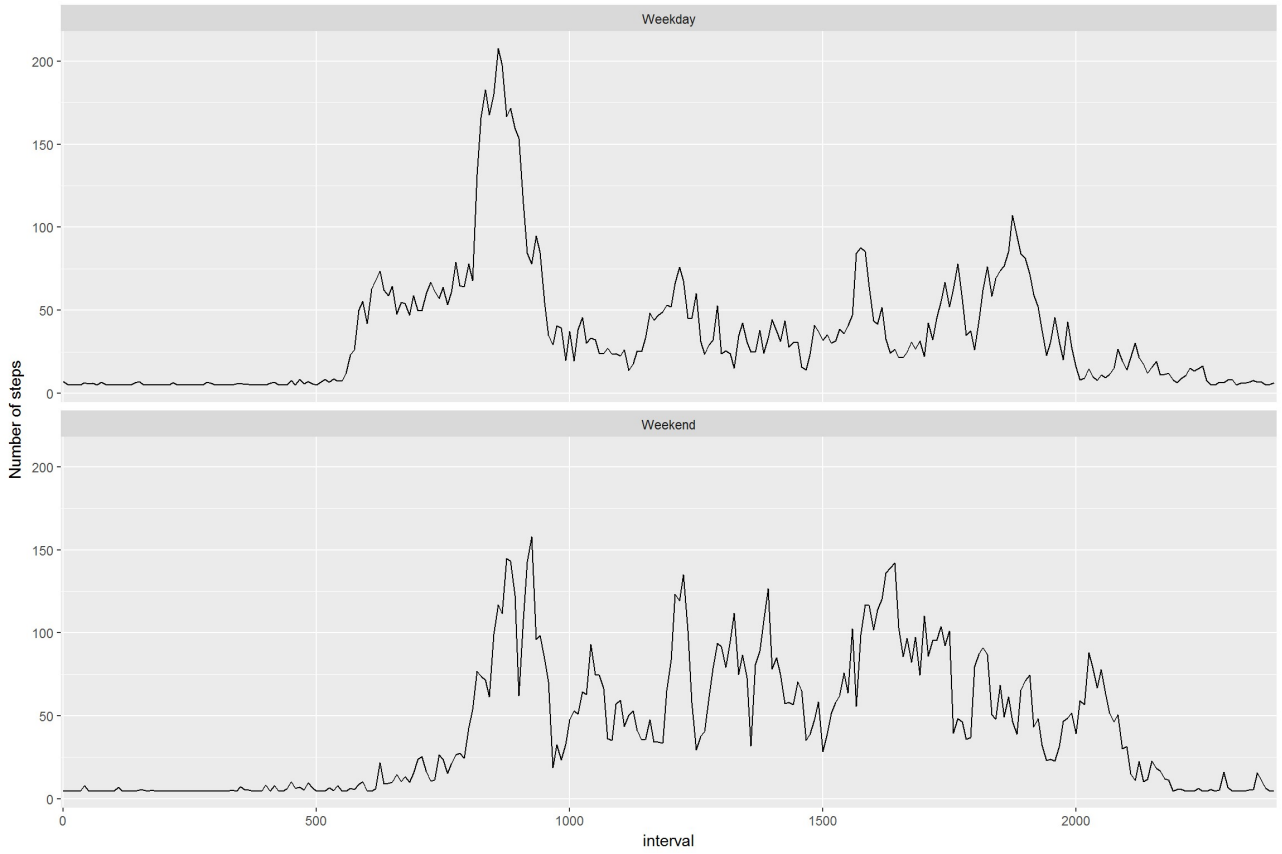
```
## [1] 10766.19
```

```
median(final_sum_data$total)
```

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

Are there differences in activity patterns between weekdays and weekends?

```
## Question 4.Are there differences in activity patterns between weekdays and weekend
s?

dataset_days <- activity_data %>%
  mutate(type_of_day = as.factor(format(date,"%a")))

dataset_days$type_of_day <- revalue(dataset_days$type_of_day, c("Mon"="Weekday", "Tu
e"="Weekday", "Wed"="Weekday",
                                                                "Thu"="Weekday", "Fr
i"="Weekday", "Sat"="Weekend",
                                                                "Sun"="Weekend"))

dataset_weekday <- subset(dataset_days, type_of_day == "Weekday")
dataset_weekend <- subset(dataset_days, type_of_day == "Weekend")

weekday_avg <- aggregate(dataset_weekday$steps, list(dataset_weekday$interval,dataset_
weekday$type_of_day), mean)
weekend_avg <- aggregate(dataset_weekend$steps, list(dataset_weekend$interval,dataset_
weekend$type_of_day), mean)

colnames(weekday_avg) <- c("interval", "type_of_day", "avg_steps")
colnames(weekend_avg) <- c("interval", "type_of_day", "avg_steps")

weekday_data <- rbind(weekday_avg, weekend_avg)
##draw graphics
ggplot(weekday_data, aes(x=interval, y=avg_steps, group=1)) + geom_line() +
  scale_x_discrete(breaks=seq(0,2500,500)) +
  facet_wrap(~ type_of_day, nrow=2) +
  ylab("Number of steps")
```


Weekday
Weekend
Number of steps
interval
0
50
100
150
200
0
500
1000
1500
2000