ReproResearchAssignment1

M Mach

1/15/2017

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.

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.

1. First, we must load the data from the working directory. The “activity.csv” file must be in your working directory for the data to load properly.

```
library(knitr)
library(ggplot2)
library(dplyr)
opts_chunk$set(echo=TRUE,results="show",cache=TRUE)
```

```
data <- read.csv("./activity.csv", header = TRUE)
data$date <- as.Date(data$date)
summary(data)
```

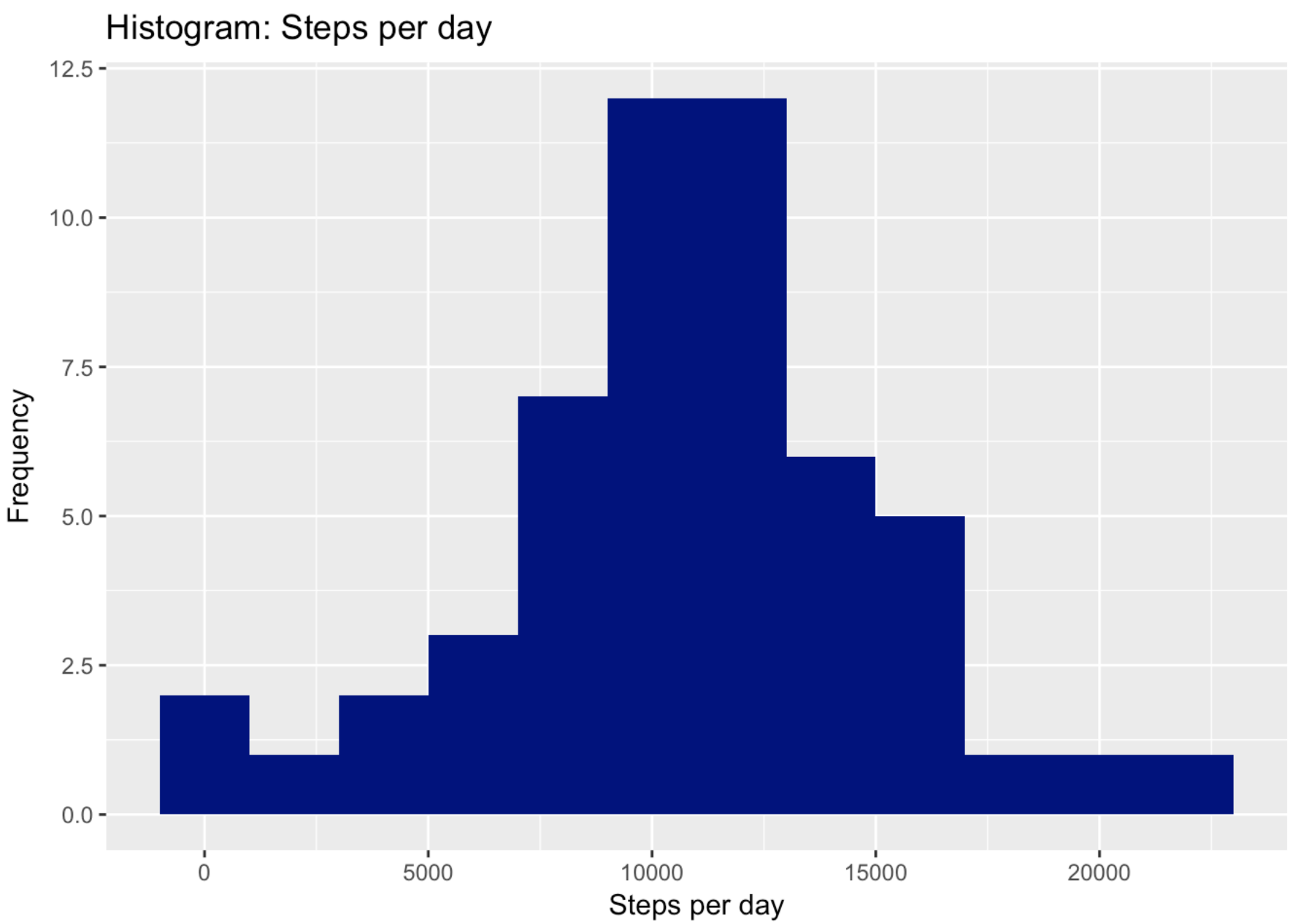
```
##      steps                date                  interval     
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0  
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8  
##  Median :  0.00   Median :2012-10-31   Median :1177.5  
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5  
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2  
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0  
##  NA's   :2304
```

2. Now that the data is loaded, we can procede with our analysis. First, we will create a histogram of the number of steps taken each day by the subject. To do this, we can group the data by “date” and summarize by “steps.” Next, we create a histogram using the grouped data.

```
#Group data
stepsByday <- data %>% filter(!is.na(steps)) %>% group_by(date) %>% summarize(steps = sum(steps)) %>% print
```

```
## # A tibble: 53 × 2
##          date steps
##        <date> <int>
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## # ... with 43 more rows
```

```
#Generate plot
ggplot(stepsByday, aes(x = steps)) + geom_histogram(fill = "navy blue", binwidth = 2000) + labs(title = "Histogram: Steps per day", x = "Steps per day", y = "Frequency")
```



3. We can use the same grouped data to provide a mean and median number of steps taken each day.

```
stepsMean <- mean(stepsByday$steps, na.rm=TRUE)
print(stepsMean)
```

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

```
stepsMedian <- median(stepsByday$steps, na.rm=TRUE)
print(stepsMedian)
```

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

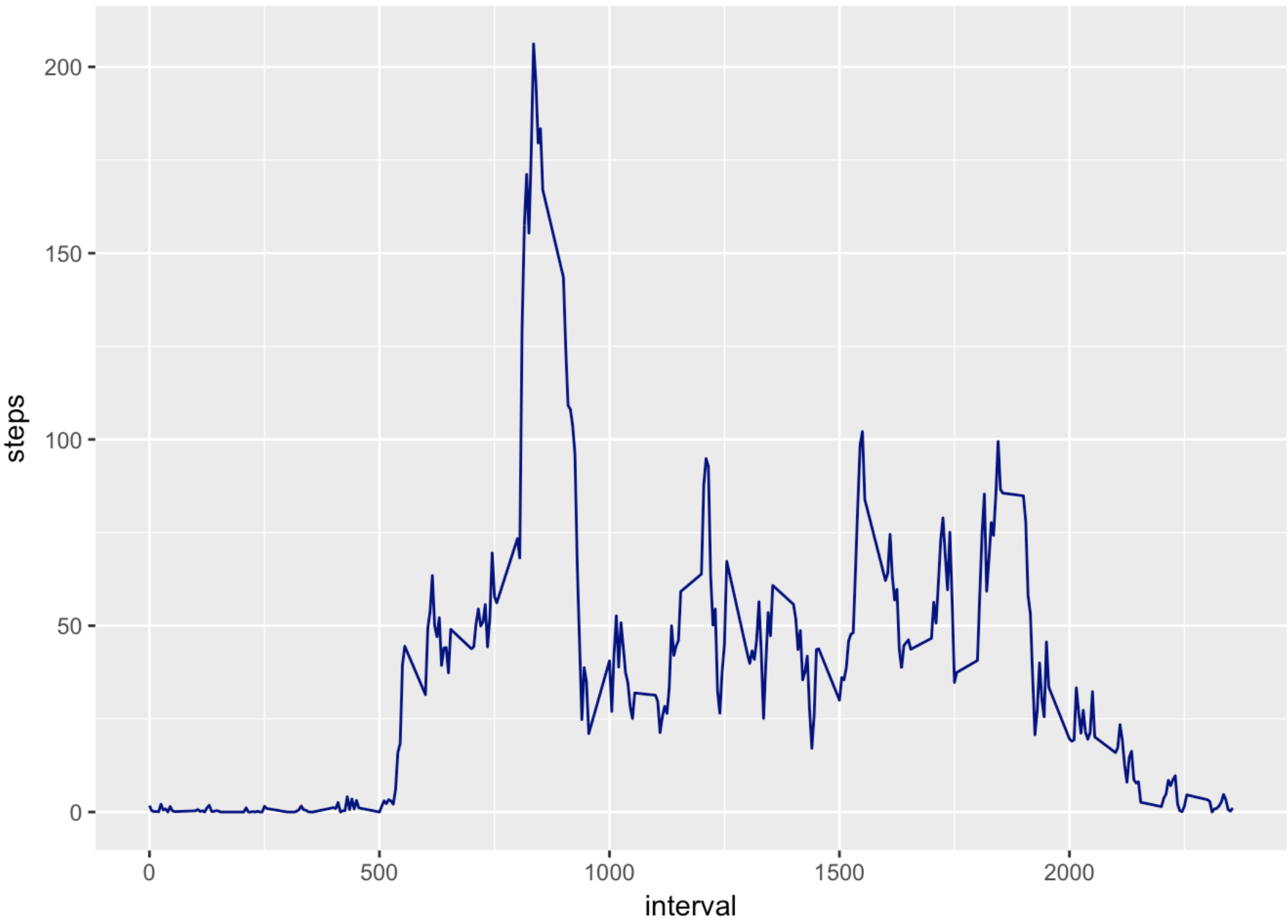
The histogram shows that the subject takes an average approximately 10,700 steps each day.

4. To see the average daily activity pattern for the subject, we can create a time series plot. Instead of grouping our data by “data” as we did earlier, we can group by “interval” and summarize by the mean number of “steps” for each interval.

```
#group and summarize data
interval <- data %>% filter(!is.na(steps)) %>% group_by(interval) %>% summarize(steps
= mean(steps)) %>% print
```

```
## # A tibble: 288 × 2
##    interval     steps
##       <int>     <dbl>
## 1         0 1.7169811
## 2         5 0.3396226
## 3        10 0.1320755
## 4        15 0.1509434
## 5        20 0.0754717
## 6        25 2.0943396
## 7        30 0.5283019
## 8        35 0.8679245
## 9        40 0.0000000
## 10       45 1.4716981
## # ... with 278 more rows
```

```
#generate plot
ggplot(interval, aes(x=interval, y=steps)) + geom_line(color = "navy blue")
```



The time series graph indicates that the subject usually takes more steps in the earlier part of the day.

5. Next, we can also show the 5 minute interval that contains the most steps on average.

```
interval[which.max(interval$steps),]
```

```
## # A tibble: 1 × 2
##   interval    steps
##      <int>    <dbl>
## 1      835 206.1698
```

6. The data currently contains several missing values. In order to impute these missing values, we can calculate the mean steps by interval, and then place this value in each missing value.

```
allData <- data
nas <- is.na(allData$steps)
meanInterval <- tapply(allData$steps, allData$interval, mean, na.rm=TRUE, simplify=TR
UE)
allData$steps[nas] <- meanInterval[as.character(allData$interval[nas])]
sum(is.na(data$steps))
```

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

```
sum(is.na(allData$steps))
```

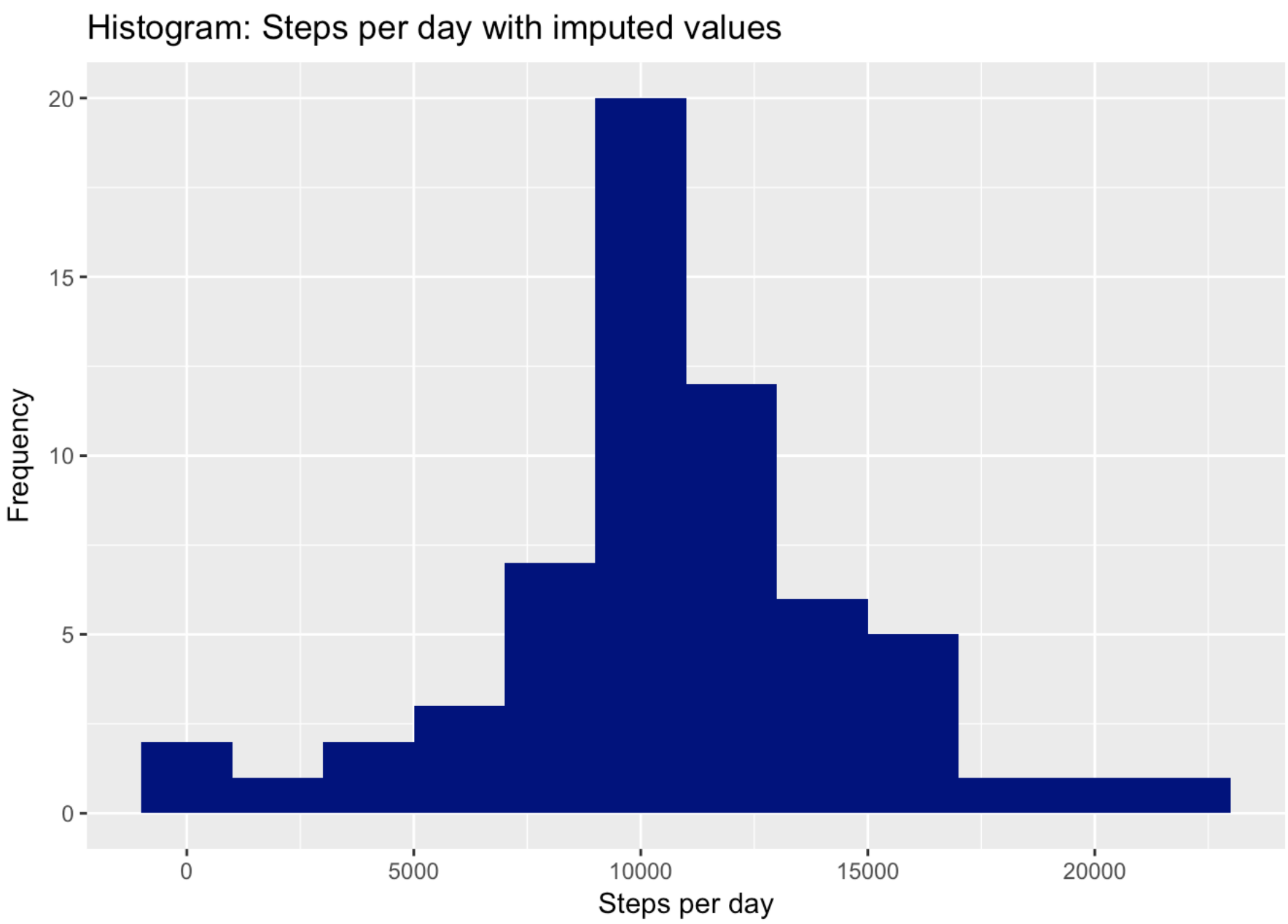
```
## [1] 0
```

7. Next, we can incorporate the imputed values into a new histogram. To do this, we must use the new imputed data to group values by date and summarize the data by steps taken. After this, we can plot the grouped data.

```
#group data
allSteps <- allData %>% filter(!is.na(steps)) %>% group_by(date) %>% summarize(steps
= sum(steps)) %>% print
```

```
## # A tibble: 61 × 2
##          date    steps
##        <date>    <dbl>
## 1  2012-10-01 10766.19
## 2  2012-10-02   126.00
## 3  2012-10-03 11352.00
## 4  2012-10-04 12116.00
## 5  2012-10-05 13294.00
## 6  2012-10-06 15420.00
## 7  2012-10-07 11015.00
## 8  2012-10-08 10766.19
## 9  2012-10-09 12811.00
## 10 2012-10-10  9900.00
## # ... with 51 more rows
```

```
#generate plot
ggplot(allSteps, aes(x = steps)) + geom_histogram(fill = "navy blue", binwidth = 2000
) + labs(title = "Histogram: Steps per day with imputed values", x = "Steps per day",
y = "Frequency")
```



The new Mean and Median of the data with imputed values:

```
meanAllSteps <- mean(allSteps$steps, na.rm = TRUE)
print(meanAllSteps)
```

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

```
medianAllSteps <- median(allSteps$steps, na.rm = TRUE)
print(medianAllSteps)
```

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

After taking into account the new imputed values, we can see that the mean and median are equal. Previsouly, the median was slightly lower than the mean, indicating a left skew. By imputing the missing values as the average, we have equalized the mean and median.

8. Using this new imputed data, we can create a new factor, “weektype,” that segments our data by values collected on weekdays and values collected on the weekend.

```
weekData <- mutate(allData, weektype = ifelse(weekdays(allData$date) == "Saturday" |
weekdays(allData$date) == "Sunday", "weekend", "weekday"))
weekData$weektype <- as.factor(weekData$weektype)
```

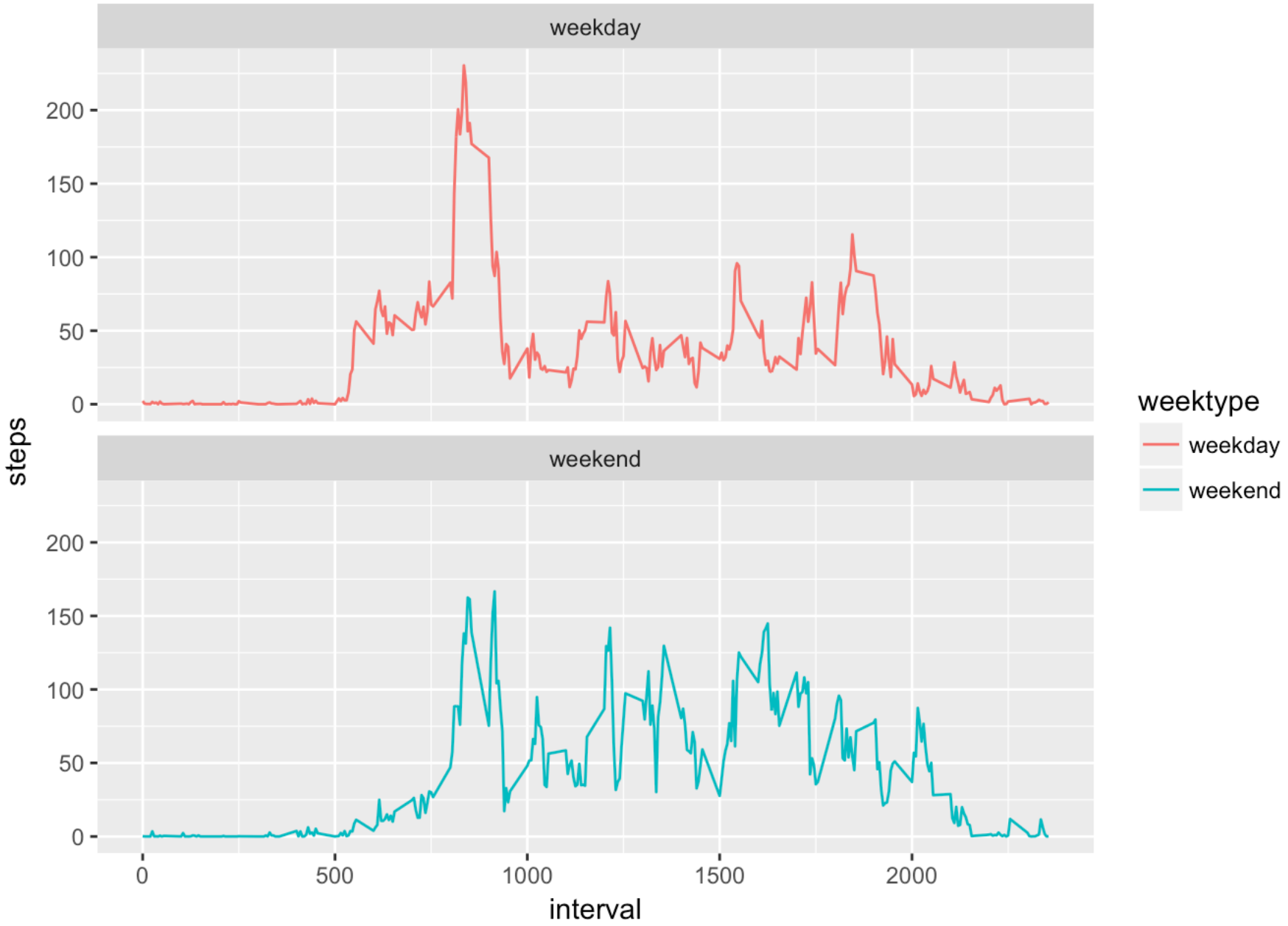
The new data and variable can be seen in the data header below.

```
head(weekData)
```

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

Using this new variable, we can create a new time series chart that shows the difference between steps taken during the work week and during the weekend.

```
#group data
allInterval <- weekData %>% group_by(interval, weektype) %>% summarise(steps = mean(s
teps))
#generage plot
graph <- ggplot(allInterval, aes(x=interval, y=steps, color = weektype)) + geom_line(
) + facet_wrap(~weektype, ncol = 1, nrow=2)
print(graph)
```



The new time series charts show that on weekdays the subject takes more steps in the early part of the day, then fewer steps through the rest of the day. However, on the weekend the subect is more active throughout the day. This may be a result of the subject's job. The subject may be working out or walking to work in the morning, then sitting at a desk all day. During the weekend, he may be taking less steps per interval, but more consistently active throughout the day.