PA1_template.Rmd

Venkat

7/26/2019

ReproducibleResearch: Week2 Assignment for Peer Review

Step1: Load and Read Data.

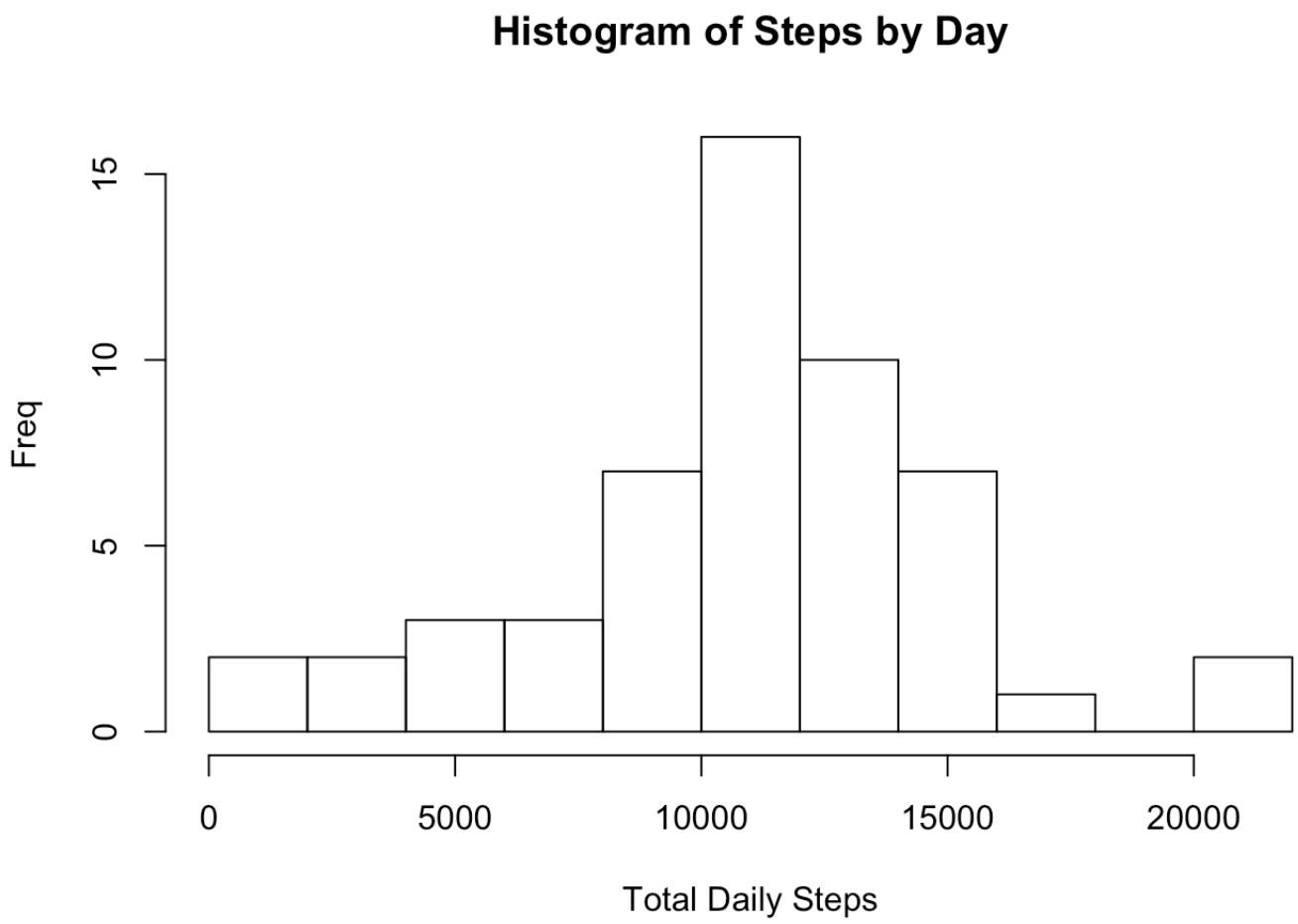
```
#download file directly from online and extract it
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip",
destfile = "activity.zip", mode = "wb")
unzip("activity.zip")

#Read the data and check the contents
activity_data <- read.csv("activity.csv", header = TRUE)
head(activity_data)
```

```
##   steps       date interval
## 1    NA 2012-10-01        0
## 2    NA 2012-10-01        5
## 3    NA 2012-10-01       10
## 4    NA 2012-10-01       15
## 5    NA 2012-10-01       20
## 6    NA 2012-10-01       25
```

Step 2: Histogram of total steps/day

```
# aggregate the 5 min data into day level
stepsbydate <- activity_data %>% select(date,steps) %>% group_by(date) %>% summarize(
totalsteps = sum(steps)) %>%na.omit()
#plot the histogram
hist(stepsbydate$totalsteps, xlab="Total Daily Steps", ylab="Freq", main="Histogram o
f Steps by Day", breaks = 15)
```



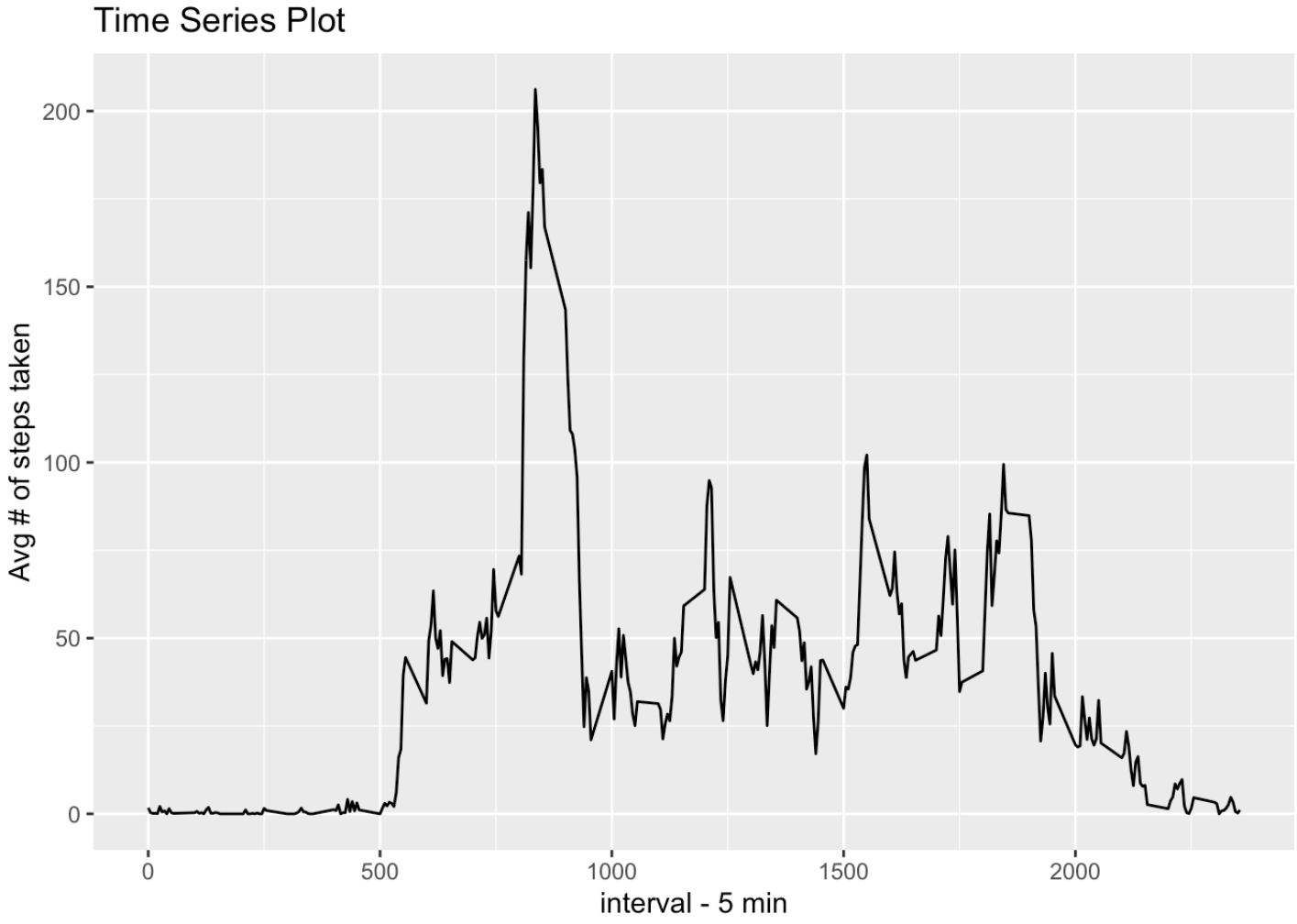
Step 3: Mean and Median of steps/day

```
stepsbydateMean <- mean(stepsbydate$totalsteps)
stepsbydateMedian <- median(stepsbydate$totalsteps)
```

- Mean: 1.076618910^{4}
- Median: 10765

Step 4: Time series Plot of Avg Steps taken per day

```
databyinterval <- activity_data %>% select(interval, steps) %>% na.omit() %>% group_b
y(interval) %>% summarize(totalsteps = mean(steps))
## Now plot the summary of steps
ggplot(databyinterval, aes(x=interval, y=totalsteps))+geom_line()+labs(title="Time Se
ries Plot", y="Avg # of steps taken", x="interval - 5 min")
```



Step 5: The 5-minute interval where average contains max #of steps

```
databyinterval[which(databyinterval$totalsteps == max(databyinterval$totalsteps)),]
```

```
## # A tibble: 1 x 2
##   interval totalsteps
##      <int>      <dbl>
## 1      835       206.
```

Step 6: Code to describe and show the strategy for imputing the missing data

```
count_of_missing_values = length(which(is.na(activity_data$steps)))
summary(activity_data)
```

```
##      steps               date          interval
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8
##  Median :  0.00   2012-10-03:  288   Median :1177.5
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840
```

- missing data rows: 2304
  Use impute function with mean and fill steps for the days they are missing.

```
library(dplyr)
replacewithmean <- function(num) replace(num, is.na(num),mean(num, na.rm = TRUE))
activitydata_nomissing <- activity_data %>% group_by(interval) %>% mutate(steps = rep
lacewithmean(steps))
head(activitydata_nomissing)
```

```
## # A tibble: 6 x 3
## # Groups:   interval [6]
##    steps date       interval
##    <dbl> <fct>         <int>
## 1 1.72   2012-10-01        0
## 2 0.340  2012-10-01        5
## 3 0.132  2012-10-01       10
## 4 0.151  2012-10-01       15
## 5 0.0755 2012-10-01       20
## 6 2.09   2012-10-01       25
```

```
new_activity_data = as.data.frame(activitydata_nomissing)
head(new_activity_data)
```

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

```
count_of_missing_values2 = length(which(is.na(new_activity_data$steps)))
```

- Number of missing values(steps): 0

```
summary(new_activity_data)
```

```
##      steps                date          interval     
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0  
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8  
##  Median :  0.00   2012-10-03:  288   Median :1177.5  
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5  
##  3rd Qu.: 27.00   2012-10-05:  288   3rd Qu.:1766.2  
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0  
##                   (Other)   :15840
```

Step 7: Histogram of Total steps taken for each day after missing values are imputed

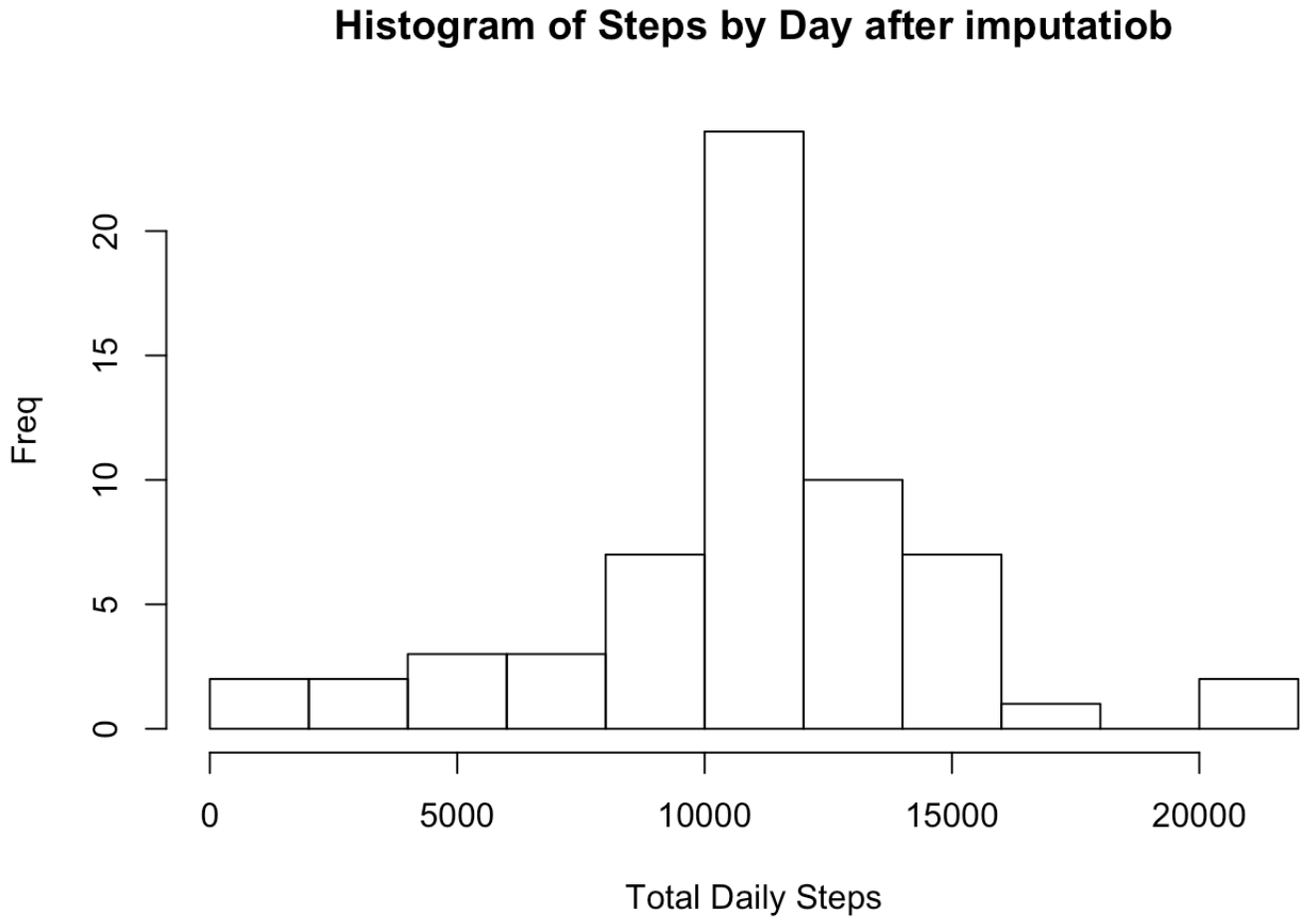
For the histogram sum up the steps for each day

```
day_summary <- aggregate(new_activity_data$steps, by=list(new_activity_data$date), su
m)

names(day_summary)[1] = "date"
names(day_summary)[2] = "totalsteps"
head(day_summary)
```

```
##         date totalsteps
## 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
```

```
hist(day_summary$totalsteps, xlab="Total Daily Steps",ylab="Freq" ,main="Histogram of Steps by Day after imputatiob", breaks = 15)
```



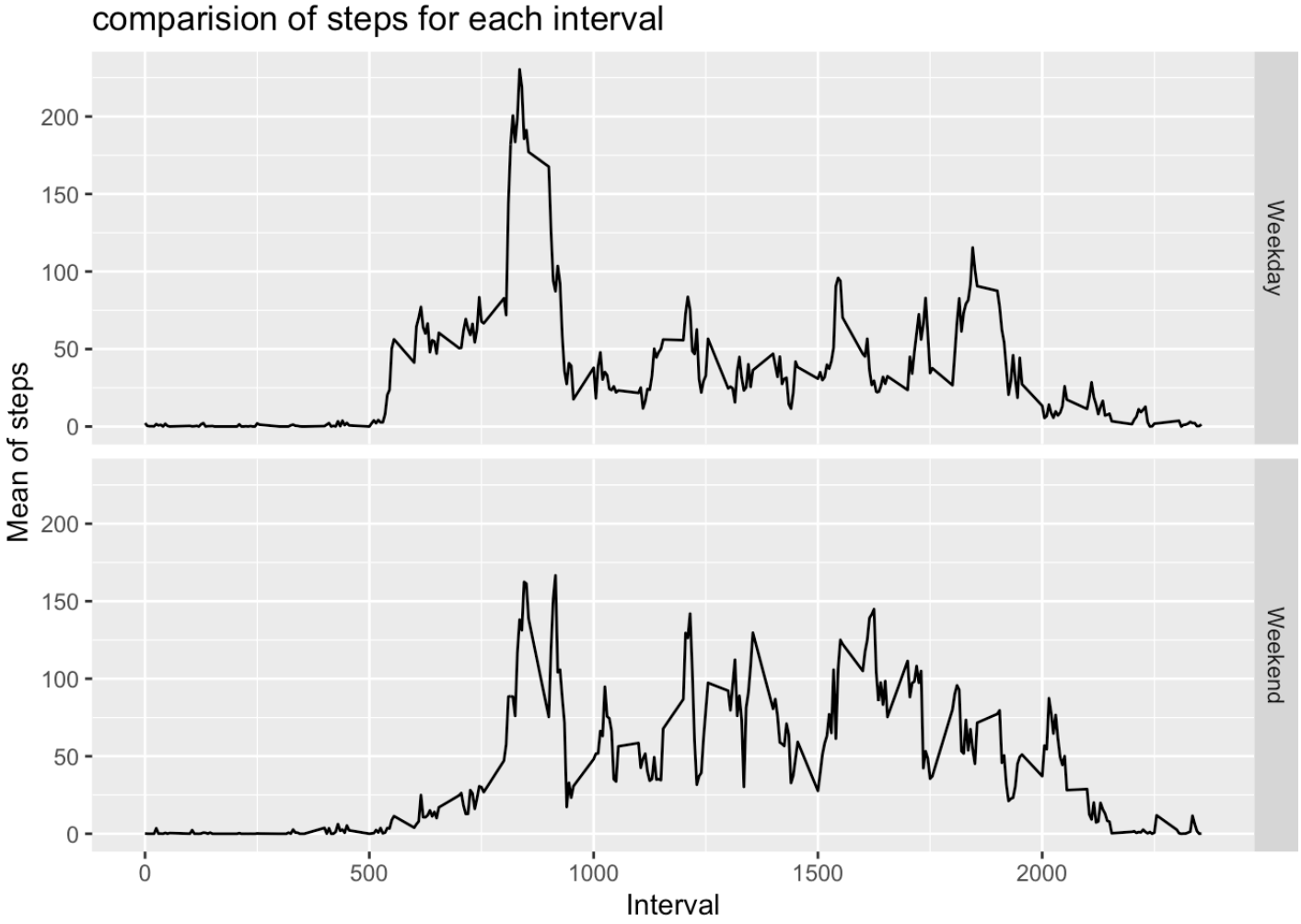
Step 8: Panel split comparing the average number of steps taken per 5min interval across week days and weekends

```
new_activity_data$weekend_flag <- ifelse(weekdays(as.Date(new_activity_data$date)) %in% c("Monday","Tuesday","Wednesday","Thursday","Friday"),"Weekday","Weekend")
head(new_activity_data)
```

```
##       steps       date interval weekend_flag
## 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
```

```
new_activity_data <- (new_activity_data %>% group_by(interval,weekend_flag) %>% summarise(Mean= mean(steps)))

ggplot(new_activity_data, mapping = aes(x=interval, y=Mean)) + geom_line()+
facet_grid(weekend_flag ~.) +xlab("Interval") +ylab("Mean of steps") + ggtitle("comparision of steps for each interval")
```



Step 9: All of R code needed to reproduce the results in the report.