Reproducible Research Week2 Course Project 1

Lulu

11/13/2020

Download and unzip the data

```
fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
if(!file.exists("activity.csv")) {
        download.file(fileUrl, destfile = "c5p1.zip", method = "curl")
        dateDownloaded <- date()
        unzip("c5p1.zip")
}
activity <- read.csv("activity.csv")
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day

```
# Load the library
library(dplyr)
```

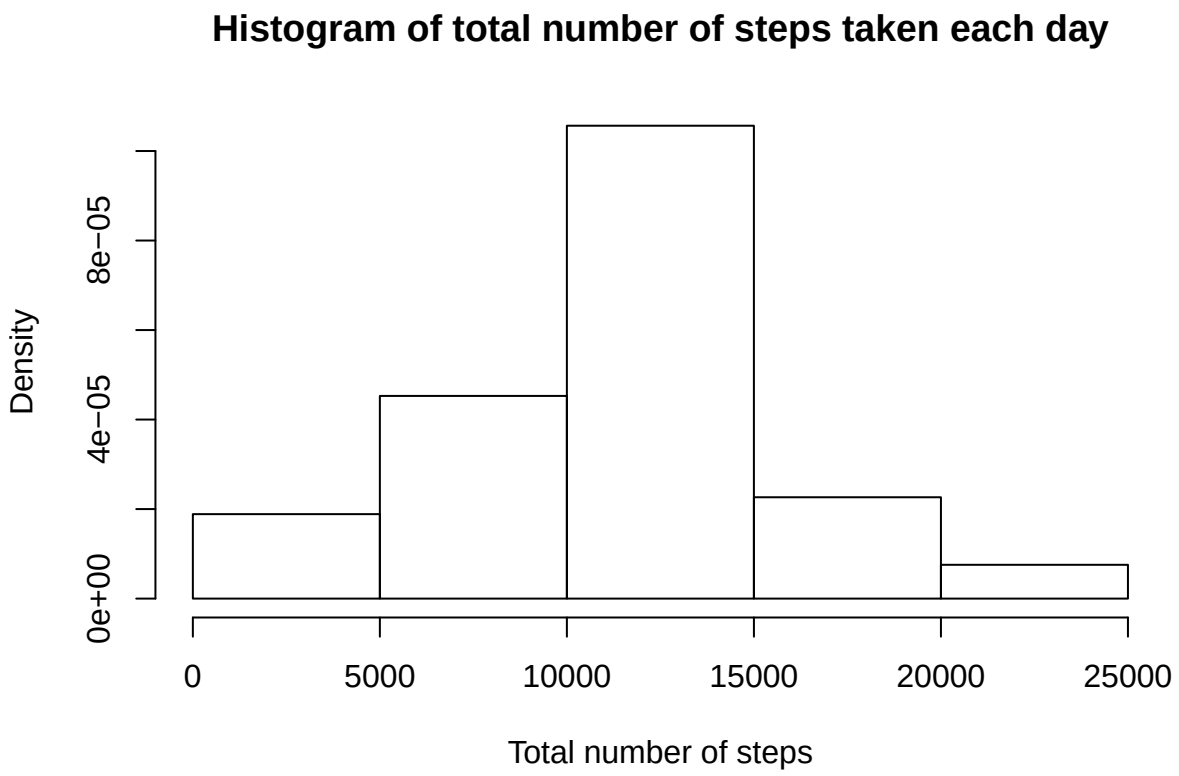
```
##
## Attaching package: 'dplyr'

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

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

```
library(ggplot2)
# Calculate the total number of steps taken per day
total <- activity%>%group_by(date)%>%summarize(sum(steps))
str(total)
```

```
## tibble [61 x 2] (S3: tbl_df/tbl/data.frame)
##  $ date      : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ sum(steps): int [1:61] NA 126 11352 12116 13294 15420 11015 NA 12811 9900 ...
```

2. Make a histogram of the total number of steps taken each day

```
hist(total$'sum(steps)', prob=TRUE, main="Histogram of total number of steps taken each day", xlab="Tota
```



3.Calculate and report the mean and median of the total number of steps taken per day

```
mean(total$`sum(steps)`, na.rm=TRUE)
```

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

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

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

The mean and median of the total number of steps taken per day are 10766.19 and 10765, respectively.

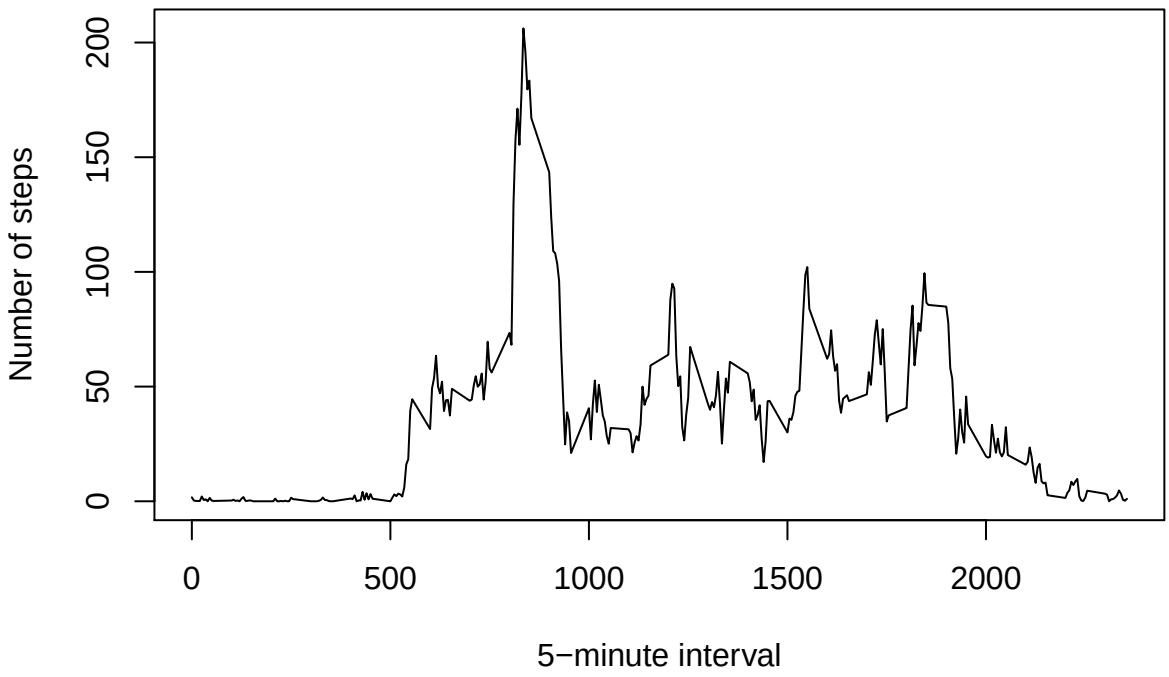
What is the average daily acitivity pattern?

1. Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```r
interval_average <- activity %>% group_by(interval) %>% summarise(mean_steps = mean(steps, na.rm=TRUE))
str(interval_average)
```

```
## tibble [288 x 2] (S3: tbl_df/tbl/data.frame)
##  $ interval  : int [1:288] 0 5 10 15 20 25 30 35 40 45 ...
##  $ mean_steps: num [1:288] 1.717 0.3396 0.1321 0.1509 0.0755 ...
```

```r
plot(interval_average, type="l", xlab="5-minute interval", ylab="Number of steps")
```



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

```r
interval_max <- interval_average$interval[which.max(interval_average$mean_steps)]
interval_max
```

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

The "835" 5-minute interval, on average across all the days in the dataset, contain the maximum number of steps.

Imputing missing values

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
missing_values <- sum(is.na(activity$steps))
missing_values
```

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

The total number of missing values in the dataset is 2304.

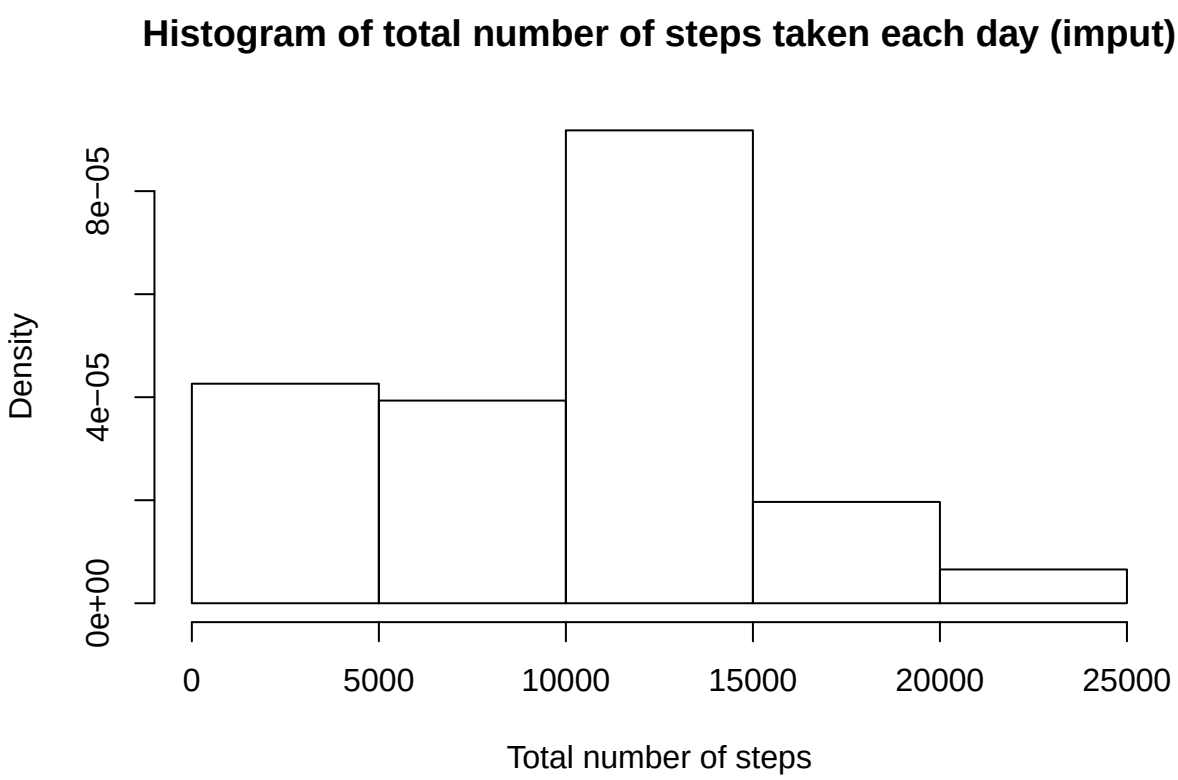
2. Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc. The missing values are imput with the mean for that 5-minute interal.

3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
activity_imput <- activity
for (i in 1:nrow(activity)){
        if(is.na(activity$steps[i])){
                activity_imput$steps[i] <- interval_average$mean_steps[which(activity$interval[i] %in% i
        }
}
```

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?

```
# Calculate the total number of steps taken per day
total_imput <- activity_imput%>%group_by(date)%>%summarize(total_steps =sum(steps))

# Make a histogram of the total number of steps taken each day
hist(total_imput$total_steps, prob=TRUE, main="Histogram of total number of steps taken each day (imput)
```




```
# Calculate the mean and median total number of steps taken per day
mean(total_imput$total_steps, na.rm=TRUE)
```

```
## [1] 9419.081
```

```
median(total_imput$total_steps, na.rm=TRUE)
```

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

After the missing values are filled, the mean and median total number of steps taken per day are reduced from 10766.19 and 10765 to 9410.081 and 10395, respectively. Imputing mssing data reduced the estimates of the total daily number of steps.

Are there differences in activity patterns between weekdays and weekends?

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.

```
activity_imput <- activity_imput %>% mutate(date=as.Date(date))
weekday <- weekdays(activity_imput$date)
for (i in 1:nrow(activity_imput)){
        if (weekday[i] %in% c("Saturday","Sunday")){
                weekday[i] <- "weekend"} else {
```

```
                        weekday[i] <- "weekday"
                }
}
weekday <- as.factor(weekday)
str(weekday)
```

```
##  Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

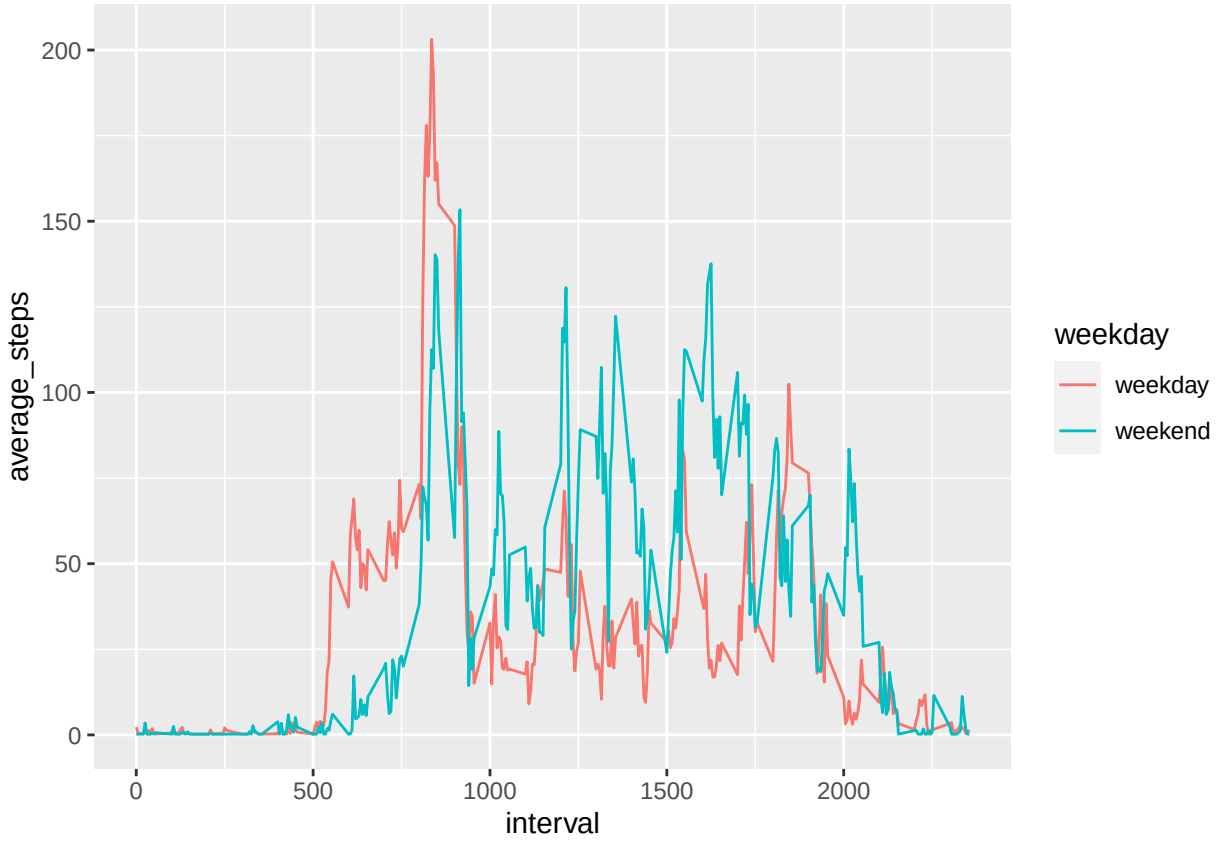
```
weekday.df <- data.frame(weekday)
activity_imput_weekday <- cbind(activity_imput, weekday.df)
```

2. Make a panel plot containing a time series plot of the 5 minute interval and the average number of steps taken, averaged across all weekday days or weekend days.

```
# Calculate the average number of steps taken across weekday or weekend
average_steps_df <- activity_imput_weekday %>% group_by(weekday, interval) %>% summarise(average_steps =
average_steps_df
```

```
## # A tibble: 576 x 3
## # Groups:   weekday [2]
##    weekday interval average_steps
##    <fct>      <int>         <dbl>
##  1 weekday        0         2.25
##  2 weekday        5         0.629
##  3 weekday       10         0.384
##  4 weekday       15         0.407
##  5 weekday       20         0.318
##  6 weekday       25         1.54
##  7 weekday       30         0.851
##  8 weekday       35         1.25
##  9 weekday       40         0.229
## 10 weekday       45         1.83
## # ... with 566 more rows
```

```
g <- ggplot(average_steps_df, aes(x=interval, y=average_steps, color=weekday))+geom_line()
g
```


average_steps
200
150
100
50
0
0
500
1000
1500
2000
interval
weekday
weekday
weekend