Peer-graded Assignment: Course Project 1

Xi Fang

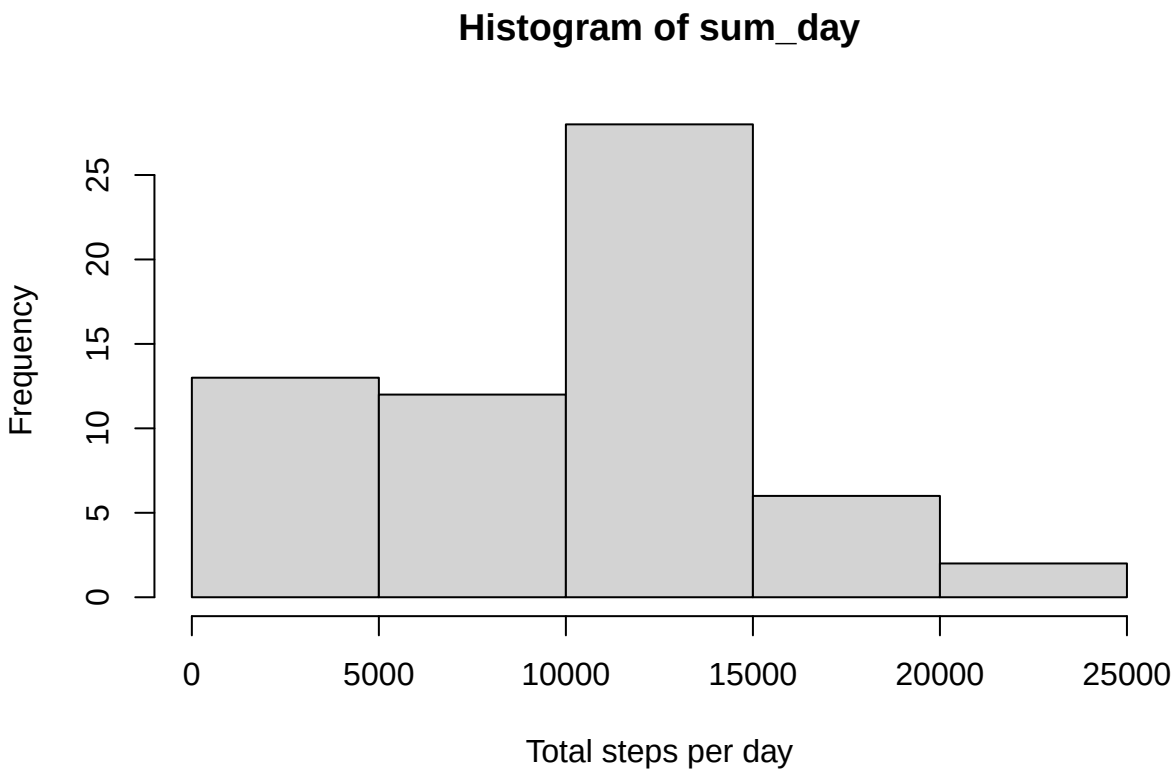
5/13/2020

Loading and preprocessing the data

```
activity <- read.csv("activity.csv")
activity$date <- as.Date(activity$date)
```

What is mean total number of steps taken per day?

```
# 1. Calculate the total number of steps taken per day
sum_day <- with(activity, tapply(steps, date,sum, na.rm = TRUE))
## an alternative way--returns a dataframe
steps_day <- aggregate(steps ~ date, activity, sum, na.rm= TRUE)

# 2. Make a histogram of the total number of steps taken each day
hist(sum_day, xlab = "Total steps per day", ylab = "Frequency")
```



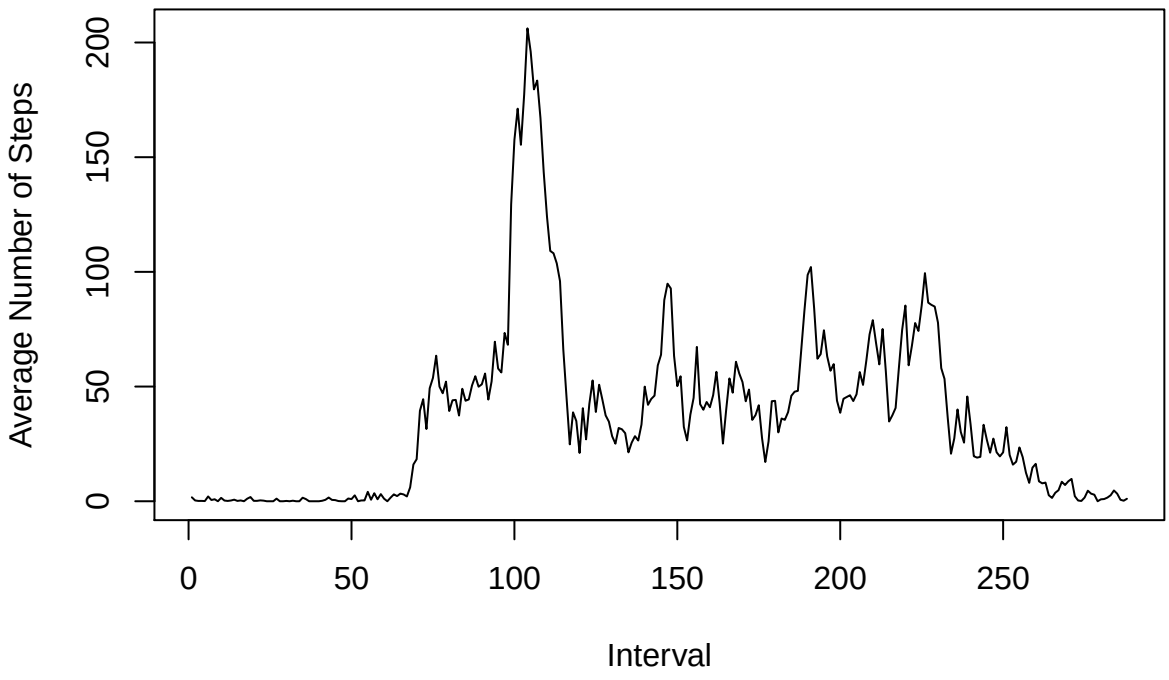

```
# 3. Calculate and report the mean and median of the total number of steps taken per day
data.frame("mean" = mean(sum_day), "median"=median(sum_day))
```

```
##      mean median
## 1 9354.23  10395
```

- The mean of the total number of steps taken per day is “9354.23”
- The median of the total number of steps taken per day is “10395”

What is the average daily activity pattern?

```
# 1. Make a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, a
mean_interval <- with(activity, tapply(steps, interval, mean, na.rm = TRUE))
plot(mean_interval,  xlab = "Interval", ylab = "Average Number of Steps", type = "l" )
```




```
# 2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum numbe
inter <- aggregate(steps ~ interval, activity, mean, na.rm= TRUE)
inter[which.max(inter$steps),]
```

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

The 5-minute intercal, on acerage across all the days in the dataset, that contains the maximun number of steps is '835'.

Imputing missing values

```
# 1. Calculate and report the total number of missing values in the dataset (i.e. the total number of r
sum(is.na(activity))
```

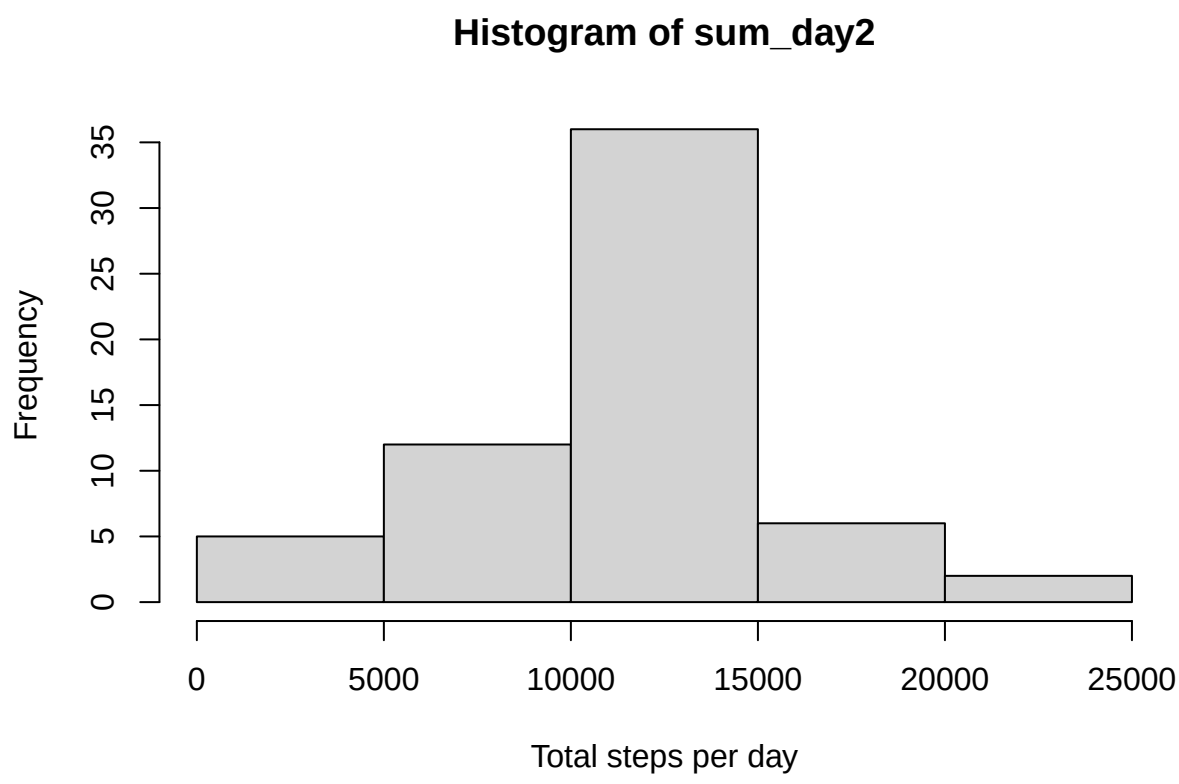
```
## [1] 2304
```

```
# 2. Devise a strategy for filling in all of the missing values with the mean for that interval in the
# 3. Create a new dataset that is equal to the original dataset but with the missing data filled in.
activity2 <- activity
for (i in 1:nrow(activity2))
{
```

```
    if (is.na(activity2$steps[i])) {
        activity2$steps[i] <- inter[which(activity2$interval[i] == inter$interval),]$steps
    }

}
summary(activity2)
```

```
##      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.: 27.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
```

```
## Make a histogram of the total number of steps taken each day and Calculate and report the mean and m
sum_day2 <- with(activity2, tapply(steps, date,sum, na.rm = TRUE))
hist(sum_day2, xlab = "Total steps per day", ylab = "Frequency")
```




```
data.frame("mean" = mean(sum_day2), "median"=median(sum_day2))
```

```
##       mean   median
## 1 10766.19 10766.19
```

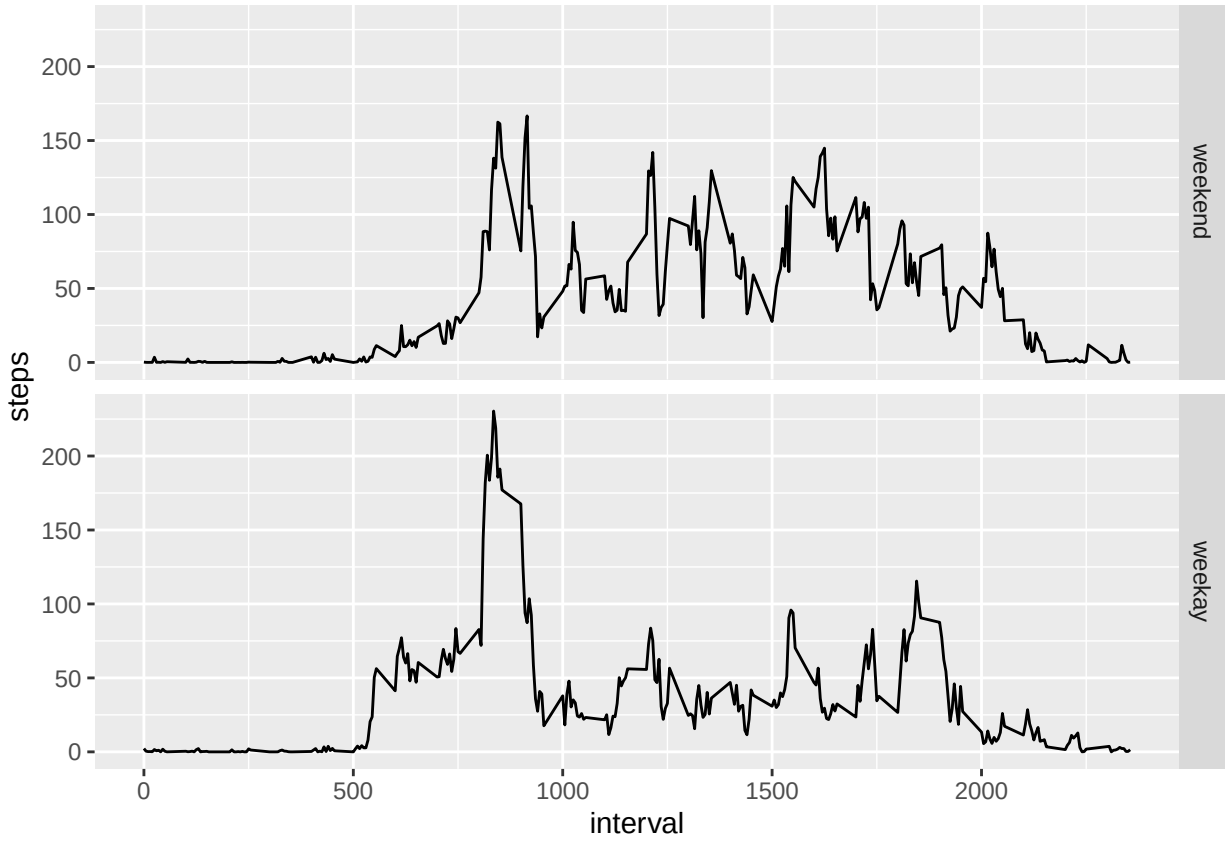
- The total number of missing values in the dataset is “2304”
- The mean of the total number of steps taken per day is “10766.19”
- The median of the total number of steps taken per day is “10766.19”
- After imputing the missing data, the new mean of total steps taken per day is the same as the median, and is the same as that of the old mean. The new histogram is more like a bell shape with less data at both ends of the plot.

Are there differences in activity patterns between weekdays and weekends?

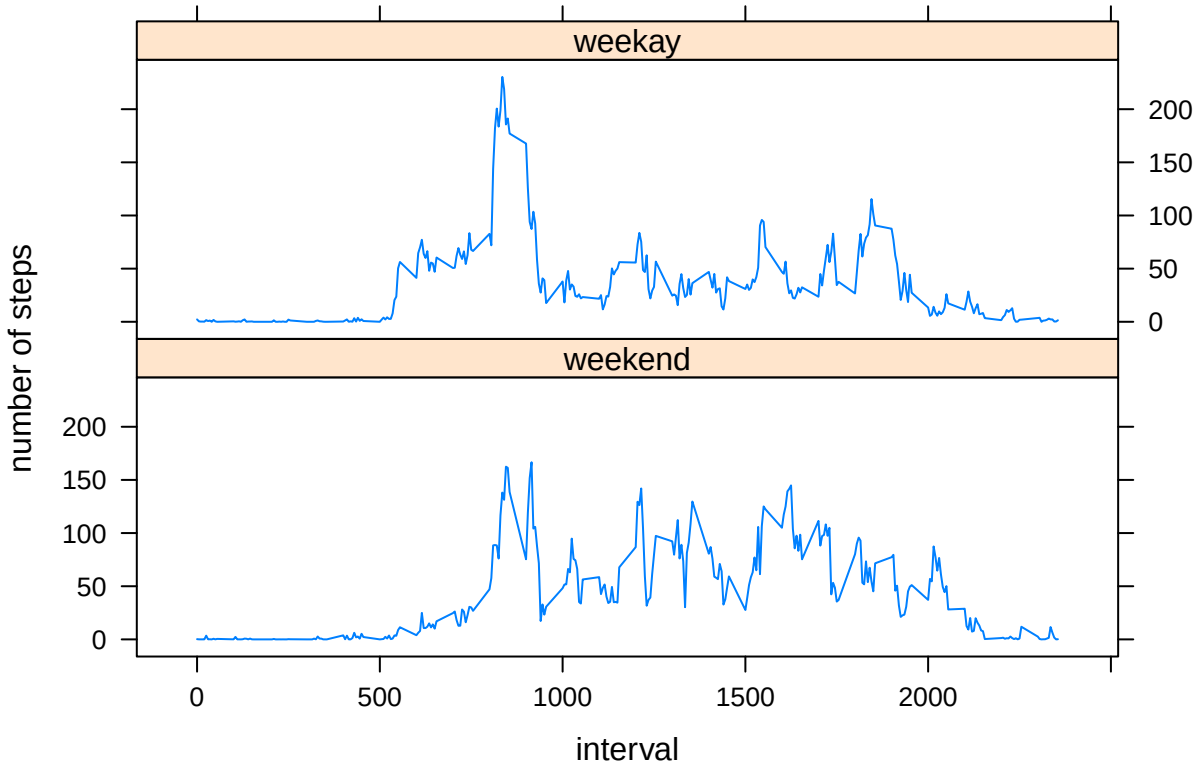
```
# 1. Create a new factor variable in the dataset with two levels - "weekday" and "weekend" indicating w
wd <- c('Monday','Tuesday', 'Wednesday','Thursday', 'Friday')
activity2$weekday <- factor((weekdays(activity2$date) %in% wd),
                            levels = c(FALSE, TRUE), labels = c('weekend','weekay'))

# 2. Make a panel plot containing a time series plot (i.e.type = "l") of the 5-minute interval (x-axis)
inter3 <- aggregate(steps ~ interval+weekday, activity2, mean)
table(inter3$weekday)
```

```
##
## weekend  weekay
##     288     288
```

```
library(ggplot2)
qplot(interval, steps,data = inter3, facets = weekday~.,
      geom = "line")
```




```
## alternative
library(lattice)
xyplot(steps~interval | factor(weekday),
       data=inter3,
       type="l",
       layout = c(1,2),
       xlab="interval",
       ylab="number of steps")
```


weekay
weekend
number of steps
interval
0
50
100
150
200
0
500
1000
1500
2000