Reproducible Research

By Jeff Gauzza

Setup

Load packages

```
library(lattice)
```

Loading and preprocessing the data

Load the data (i.e. read.csv()read.csv()) Process/transform the data (if necessary) into a format suitable for your analysis

```
# Read the data
dataFull = read.csv("activity.csv", header=TRUE)
```

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

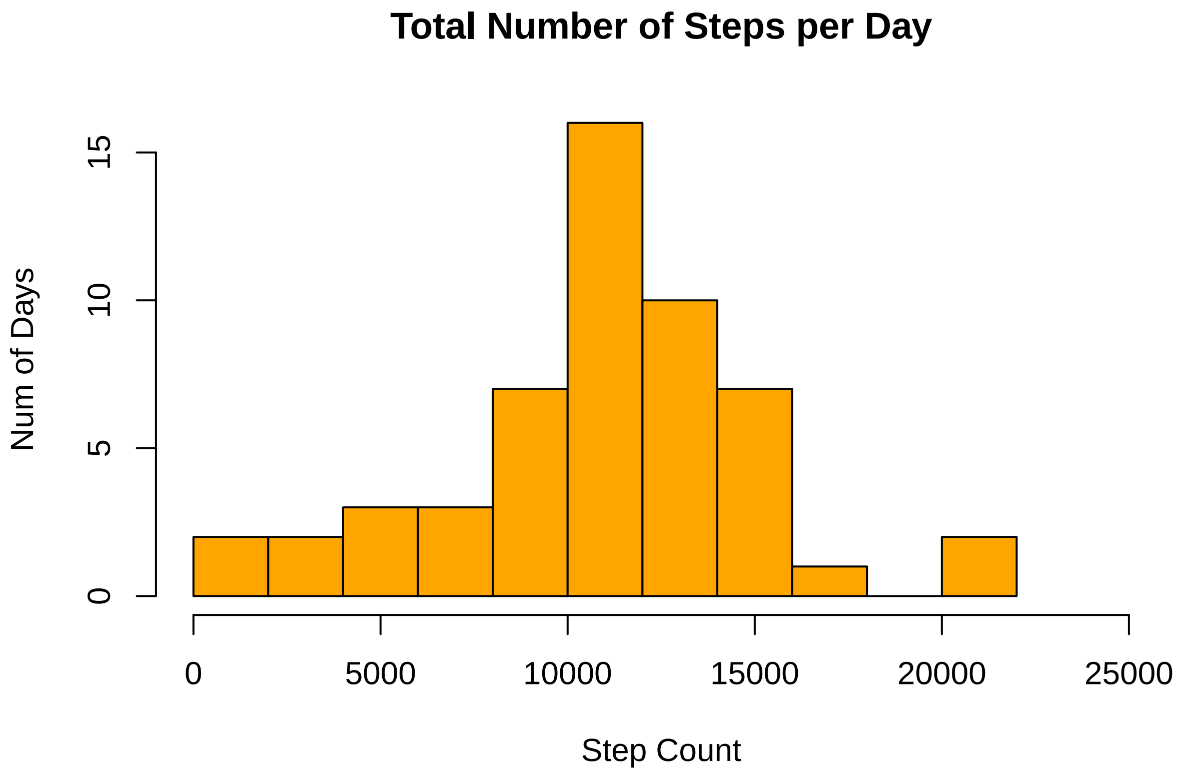
1. Calculate the total number of steps taken per day

```
spd <-aggregate(steps~date, dataFull, sum)
head(spd)
```

```
##         date steps
## 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
```

2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

```
hist(spd$steps,
     xlab="Step Count",
     ylab="Num of Days",
     main="Total Number of Steps per Day",
     col="orange",
     breaks=c(0,2000,4000,6000,8000,10000,12000,14000,16000,18000,20000,22000),
     xlim=c(0,25000))
```



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

```
c('Mean',mean(spd$steps))
```

```
## [1] "Mean"             "10766.1886792453"
```

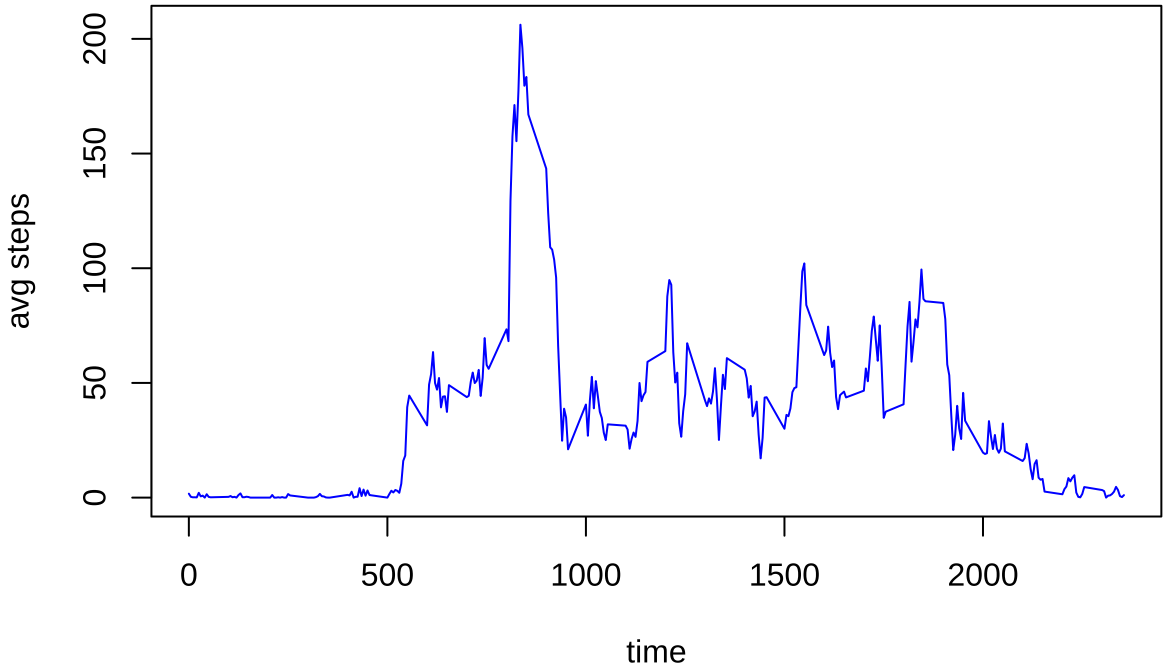
```
c('Median',median(spd$steps))
```

```
## [1] "Median" "10765"
```

What is the average daily activity pattern?

1. Make a time series plot (i.e. `type = "l"`type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
avgSteps<-aggregate(steps~interval, dataFull, mean)
with(avgSteps, plot(interval,
                    steps,
                    col = "blue",
                    xlab="time",
                    ylab="avg steps",
                    type = "l"))
```



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

```
c('Interval with Max step average',avgSteps$interval[which.max(avgSteps[,2])])
```

```
## [1] "Interval with Max step average" "835"
```

Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NANA). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

```
c('Total number of days/intervals missing values',sum(is.na(dataFull$steps)))
```

```
## [1] "Total number of days/intervals missing values"
## [2] "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.

For any missing values, the rounded mean for that same 5 minute time interval will be used to replace the NA.

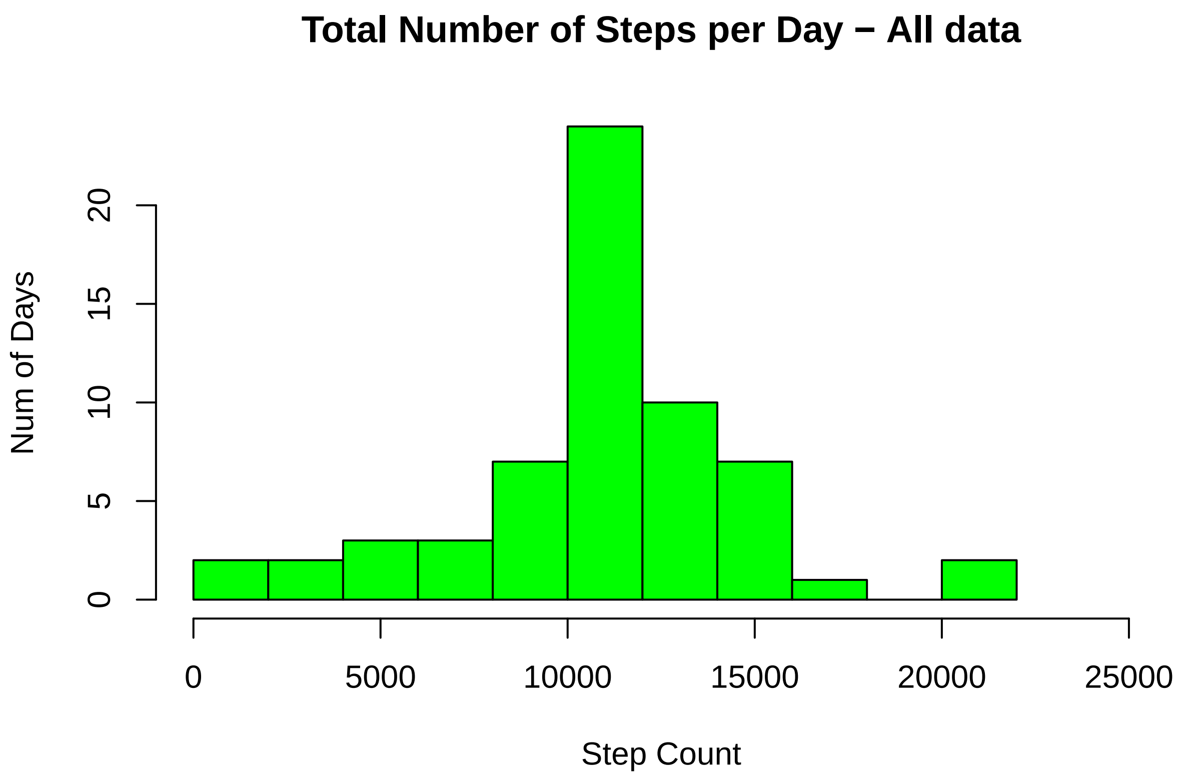
3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
dataFull$NoNA <- ifelse(is.na(dataFull$steps), round(avgSteps$steps[match(dataFull$interval, avgSteps$in
head(dataFull)
```

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

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.

```
spd2 <-aggregate(NoNA~date, dataFull, sum)
hist(spd2$NoNA,
     xlab="Step Count",
     ylab="Num of Days",
     main="Total Number of Steps per Day - All data",
     col="green",
     breaks=c(0,2000,4000,6000,8000,10000,12000,14000,16000,18000,20000,22000),
     xlim=c(0,25000))
```



```
c('Mean',mean(spd2$NoNA))
```

```
## [1] "Mean"             "10765.6393442623"
```

```
c('Median',median(spd2$NoNA))
```

```
## [1] "Median" "10762"
```

4.a. Do these values differ from the estimates from the first part of the assignment? Original values: "Mean" "10766.1886792453" "Median" "10765" New Values: "Mean" "10765.6393442623" "Median" "10762"

Yes, both the mean and median values dropped slightly with the data set contianing imputed data

4.b. What is the impact of imputing missing data on the estimates of the total daily number of steps?

For days that were mostly NA values, the totals increased significantly (See below 2012-10-01)

```
head(spd)
```

```
##         date steps
## 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
```

```
head(spd2)
```

```
##         date  NoNA
## 1 2012-10-01 10762
## 2 2012-10-02   126
## 3 2012-10-03 11352
## 4 2012-10-04 12116
## 5 2012-10-05 13294
## 6 2012-10-06 15420
```

Are there differences in activity patterns between weekdays and weekends?

For this part the `weekdays()`weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

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.

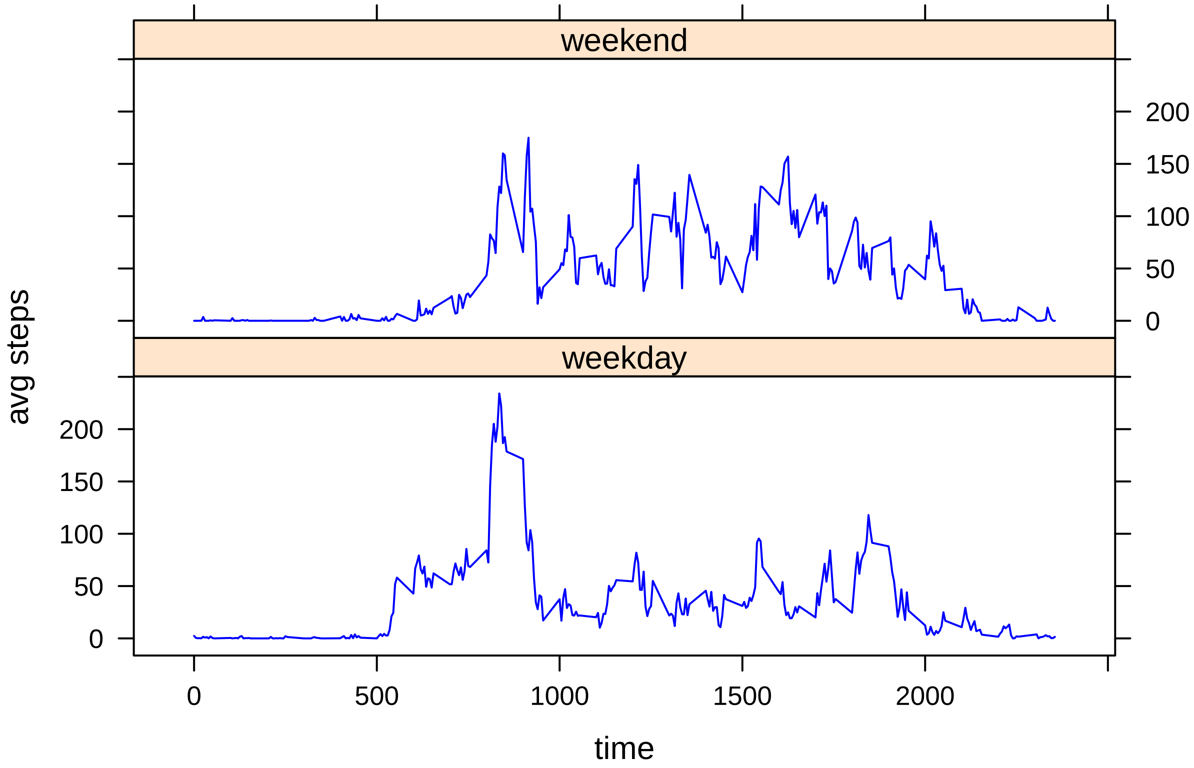
```
dataFull$date<-as.Date(dataFull$date)
dataFull$dayType <- ifelse(weekdays(dataFull$date)==’Saturday’ | weekdays(dataFull$date)==’Sunday’, ’wee
head(dataFull)
```

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

2. Make a panel plot containing a time series plot (i.e. `type = "l"`type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
spd3 <-aggregate(steps~interval+dayType, dataFull, FUN=mean)

xyplot(steps ~ interval | factor(dayType),
       layout = c(1, 2),
       col = "blue",
       xlab="time",
       ylab="avg steps",
       type="l",
       data=spd3)
```


weekend
weekday
avg steps
0
50
100
150
200
0
500
1000
1500
2000
time