Reproducible Research: Peer Assessment 1

Greg Verissimo

Overview

Assignment:

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.

The raw dataset for this assignment

- Was downloaded from here: Activity monitoring data (https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip)
- 57KB compressed (.zip) and 351KB uncompressed (.csv)
- There are a total of 17,568 observations contained the dataset

The variables included in this dataset are:

1. steps: Number of steps taking in a 5-minute interval (missing values are coded as "NA")
2. date: The date on which the measurement was taken in YYYY-MM-DD format
3. interval: Identifier for the 5-minute interval in which measurement was taken

Loading and preprocessing the data

Assignment:

Show any code that is needed to

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

First load required libraries:

```
## Load  libraries
library(reshape2)
library(ggplot2)
library(dplyr)
```

We loaded the raw dataset (.csv) into a dataframe (‘activity’) using read.csv()

```
## read raw dats (.csv file) into dataframe ('activity')
activity <- read.csv("./data/activity.csv",
                     header=TRUE,
                     colClasses=c(
                             "numeric",   ## steps: number of steps taken during 5min su
b-interval of a day
                             "character", ## date: the specific day of sample 'steps'
                             "numeric"    ## interval: the specific 5min sub-interval of
sample 'steps'
                     )
)
```

and converted variables (date, interval) to factors for later manipulation/reshaping.

```
## convert string ("YYYY-MM-DD") to class 'date' and then to 'factor'
activity$date <- as.factor(as.Date(activity$date, format="%Y-%m-%d"))
## convert interval to class 'factor'
activity$interval <- as.factor(activity$interval)
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  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: Factor w/ 288 levels "0","5","10","15",..: 1 2 3 4 5 6 7 8 9 10 ...
```

What is mean total number of steps taken per day?

Assignment:

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

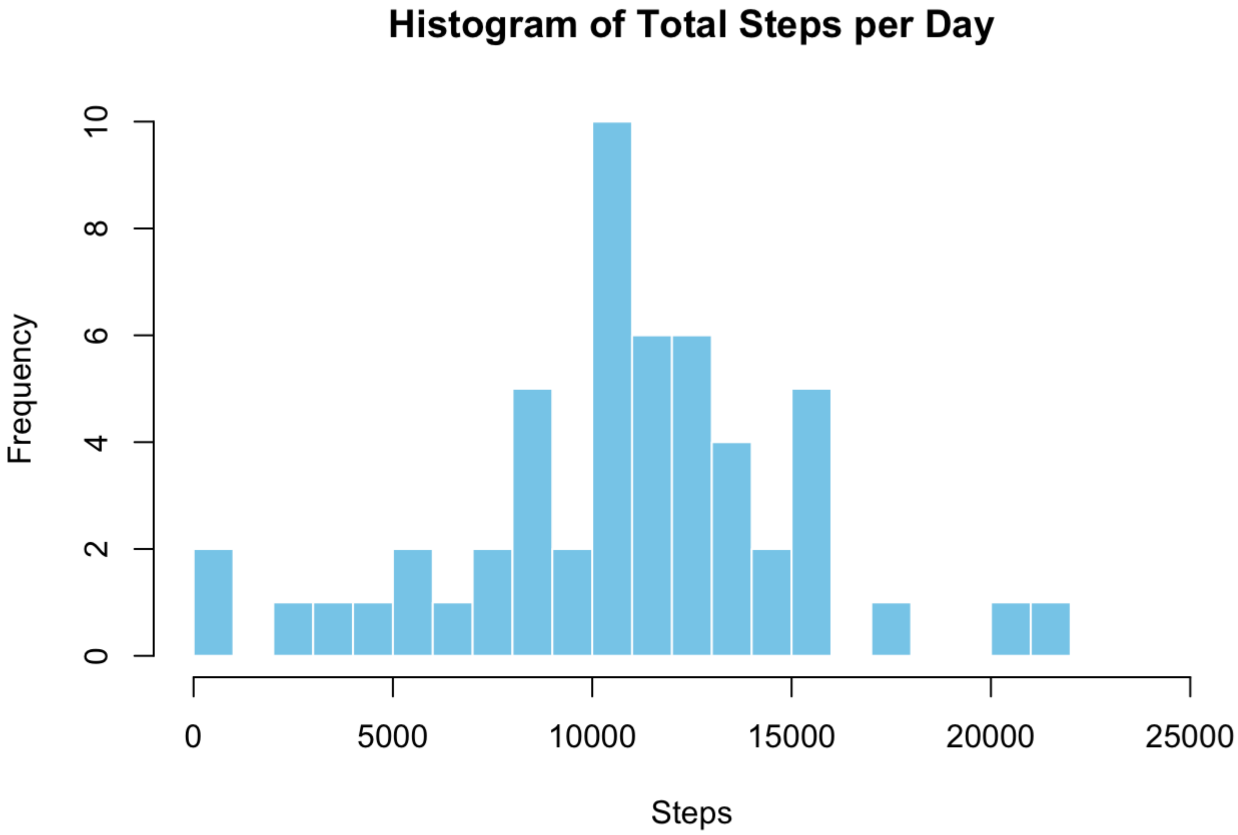
1. *Calculate the total number of steps taken per day*
2. *Make a histogram of the total number of steps taken each day. If you do not understand the difference between a histogram and a barplot, research the difference between them.*
3. *Calculate and report the mean and median of the total number of steps taken per day*

Using aggregate() to reshape the dataset around steps by date in combination with the function sum(), we added up all the 5 minute intervals for each day into daily totals over the sample period (10/1/12 to 11/30/12).

```
stepsPerDay <- aggregate(steps ~ date, data = activity, FUN = sum, na.rm = TRUE)
meanStepsPerDay <- format(round(mean(stepsPerDay$steps), 1),
                          scientific=FALSE, nsmall=1, big.mark=",")
medianStepsPerDay <- format(round(median(stepsPerDay$steps), 1),
                            scientific=FALSE, nsmall=1, big.mark=",")
```

We then created a histogram of those daily totals where you can see the subject most frequently walked between 10,000 and 11,000 steps each day:

```
## Make a histogram of the total number of steps taken each day. If you do not understan
d the difference between a histogram and a barplot, research the difference between the
m.*
par(mfrow=c(1,1), mar=c(4,4,4,2), oma=c(1,1,1,0))
histInfo <- hist(stepsPerDay$steps,
     xlim=c(0,25000), breaks = 25,
     col = "sky blue", border = "white",
     main="Histogram of Total Steps per Day",
     xlab="Steps", ylab="Frequency")
```



```
histInfo[c("mids", "counts")]
```

```
## $mids
##  [1]   500  1500  2500  3500  4500  5500  6500  7500  8500  9500 10500
## [12] 11500 12500 13500 14500 15500 16500 17500 18500 19500 20500 21500
## 
## $counts
##  [1]  2  0  1  1  1  2  1  2  5  2 10  6  6  4  2  5  0  1  0  0  1  1
```

The mean steps/day over the period was 10,766.2 and the median was 10,765.0.

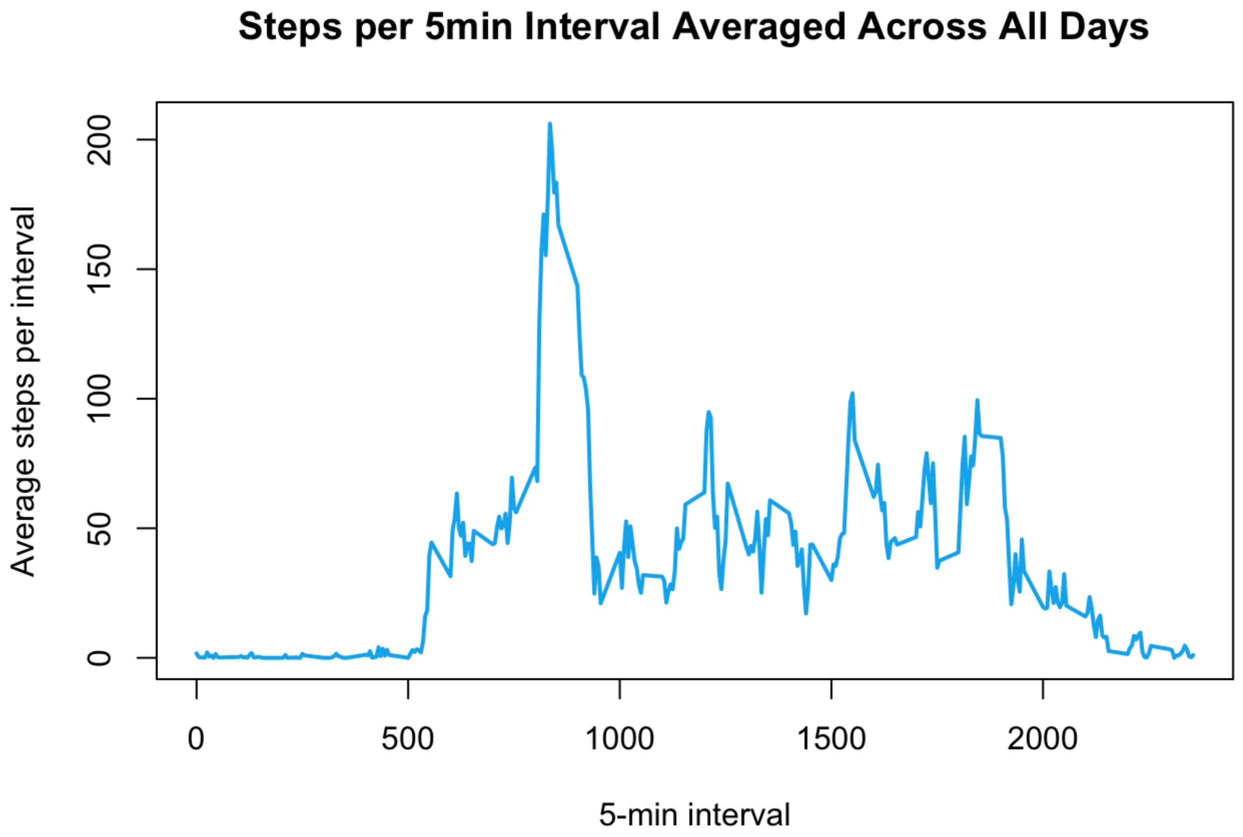
What is the average daily activity pattern?

Assignment:

1. *Make a time series plot (i.e. - type=“l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)*
2. *Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?*

Again using aggregate() –this time reshaping the dataset around steps by 5-min interval– in combination with the function mean(), we calculated the mean for each 5min interval over the sample period:

```
averageSteps <- aggregate(steps ~ interval, data = activity, mean, na.rm=TRUE)
maxAverageSteps <- averageSteps[which.max(averageSteps$steps), ]

par(mfrow=c(1,1), mar=c(4,4,4,2), oma=c(1,1,1,0))
plot(as.numeric(as.character(averageSteps$interval)), averageSteps$steps,
     type = "l",
     lwd = 2, col = "deep sky blue 2",
     main = "Steps per 5min Interval Averaged Across All Days",
     xlab = "5-min interval", ylab = "Average steps per interval" )
```



Imputing missing values

Assignment:

Note that there are a number of days/intervals where there are missing values (coded as NA). 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 NAs)*
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.*
3. *Create a new dataset that is equal to the original dataset but with the missing data filled in.*
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?*

Going back to the original (long) dataset, we counted the total quantity of missing step samples:

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

and found there are 2304 missing values.

Looking at missing values in the dataset, we saw the NAs are grouped in terms of entire days. **So we decided to impute these missing days with the average day for the sample period** that we calculated earlier *(ie - replace each NA sample interval with the corresponding average).*

We did this by first reshaping the dataset into wide format (activityWide) to move the dates (10/1~11/30) into separate columns.

```
## reshape long->wide to facilitate impute of NAs
activityWide <- dcast(activity, interval ~ date, value.var="steps")
```

We then looped over those dates, filtered each date for intervals containing NA values (‘missingVal’) and replaced those missing values with the corresponding averages calculated earlier (‘averageSteps’).

(Note that we also opted to round those imputed values to the nearest whole step)

```
## setup loop variables
firstDate <- which(colnames(activityWide)=="2012-10-01")
lastDate <- which(colnames(activityWide)=="2012-11-30")
activityWideImputed <- activityWide

## loop over daterange (Oct/1 to Nov/30)
for (i in firstDate:lastDate) {
        ## identify NA's for loop date
        missingVal <- is.na(activityWideImputed[ ,i])
        ## replace interval NA's with average steps for that interval (rounded to whole
 steps)
        activityWideImputed[missingVal, i] <- round(averageSteps$steps[missingVal],0)
}
```

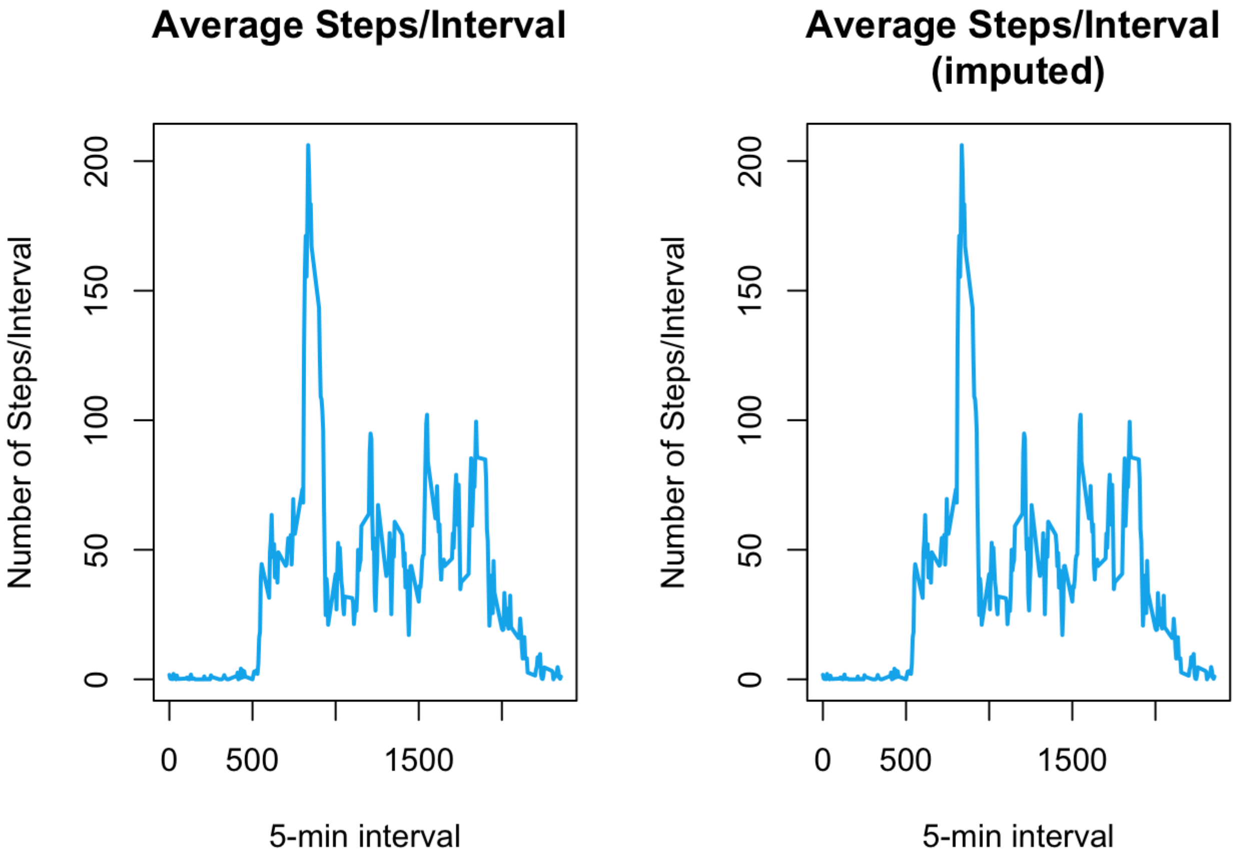
We then reshaped the data back to long format (‘activityImputed’) and recalculated the average steps per day (‘stepsPerDayImputed’) and average steps per 5min interval (averageStepsImputed) – similar to above.

```
## reshape wide->long
activityImputed <- melt(activityWideImputed,
                        id.vars=c("interval"),
                        variable.name="date",
                        value.name="steps")

## Calculate steps per day (imputed NA)
stepsPerDayImputed <- aggregate(steps ~ date, data = activityImputed, sum, na.rm = TRUE)
averageStepsImputed <- aggregate(steps ~ interval, data = activityImputed, mean, na.rm=T
RUE)
```

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

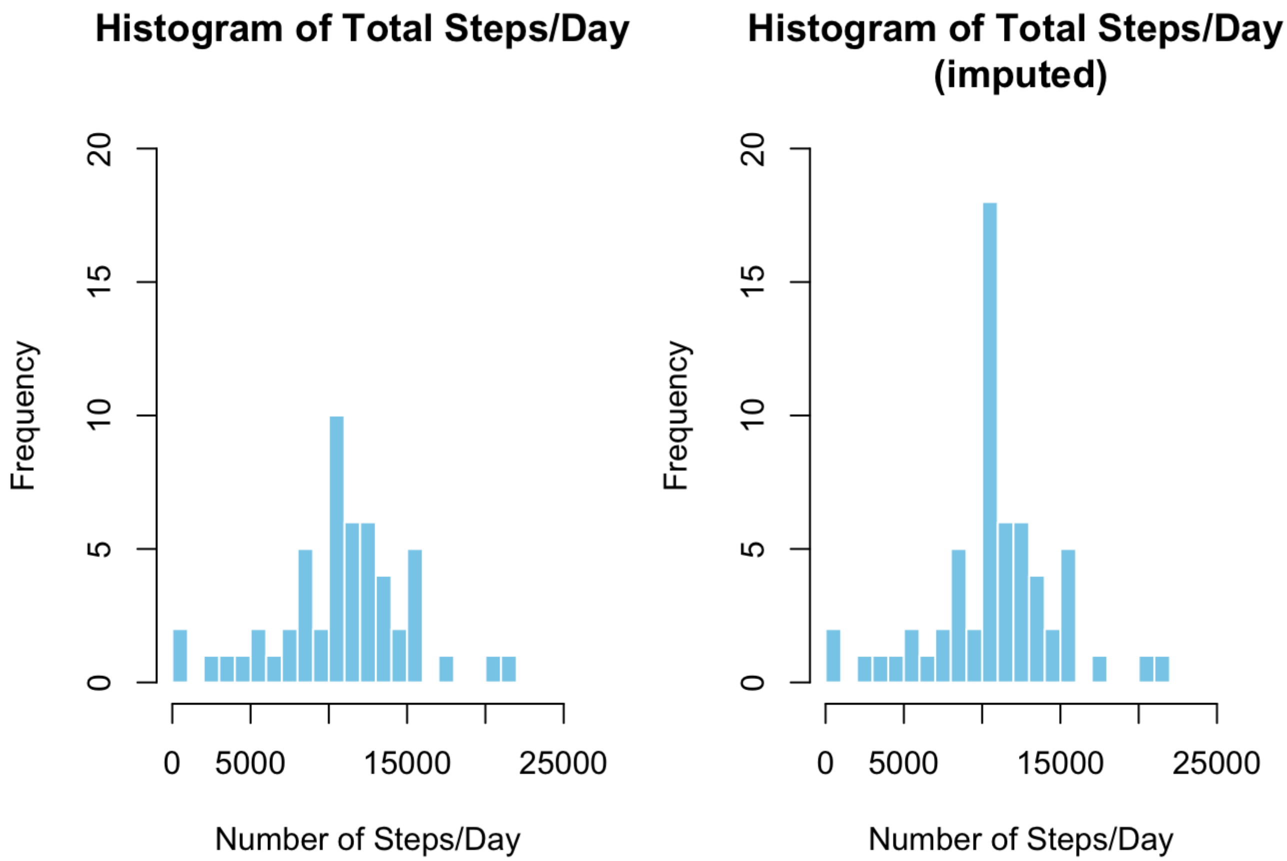
For comparison, we plotted both average daily steps as well as a histogram of steps/day and unimputed and imputed:

```
par(mfrow=c(1,2), mar=c(4,4,4,2), oma=c(1,1,1,0))

plot(as.numeric(as.character(averageSteps$interval)), averageSteps$steps,
     type = "l",
     lwd = 2, col = "deep sky blue 2",
     main = "Average Steps/Interval \n",
     xlab = "5-min interval", ylab = "Number of Steps/Interval" )
plot(as.numeric(as.character(averageStepsImputed$interval)), averageStepsImputed$steps,
     type = "l",
     lwd = 2, col = "deep sky blue 2",
     main = "Average Steps/Interval \n(imputed)",
     xlab = "5-min interval", ylab = "Number of Steps/Interval" )
```


Average Steps/Interval

Average Steps/Interval
(imputed)

```
hist(stepsPerDay$steps,
     xlim=c(0,25000), breaks = 25,
     ylim=c(0,20),
     col = "sky blue", border = "white",
     main="Histogram of Total Steps/Day \n",
     xlab="Number of Steps/Day", ylab="Frequency")
hist(stepsPerDayImputed$steps,
     xlim=c(0,25000), breaks = 25,
     ylim=c(0,20),
     col = "sky blue", border = "white",
     main="Histogram of Total Steps/Day \n(imputed)",
     xlab="Number of Steps/Day", ylab="Frequency")
```



```
meanStepsPerDayImputed <- format(round(mean(stepsPerDayImputed$steps), 1),
                                 scientific=FALSE, nsmall=1, big.mark=",")
medianStepsPerDayImputed <- format(round(median(stepsPerDayImputed$steps), 1),
                                   scientific=FALSE, nsmall=1, big.mark=",")
```

Imputing NAs as described above, the mean & median statistics changed very little:

- mean: 10,765.6 (vs. 10,766.2 unimputed)
- median: 10,765.6 (vs. 10,765.0unimputed)

We attribute this to there being (1) just 8 missing days (out of 61) and (2) we subsitiued average days for the missing ones (which would only strengthen the mean).

It is interesting to note (again, because we're imputing the average day for the missing days) that the the only change to the histogram was to heighten the central peak –which is to be expected.

Are there differences in activity patterns between weekdays and weekends?

Assignment:

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

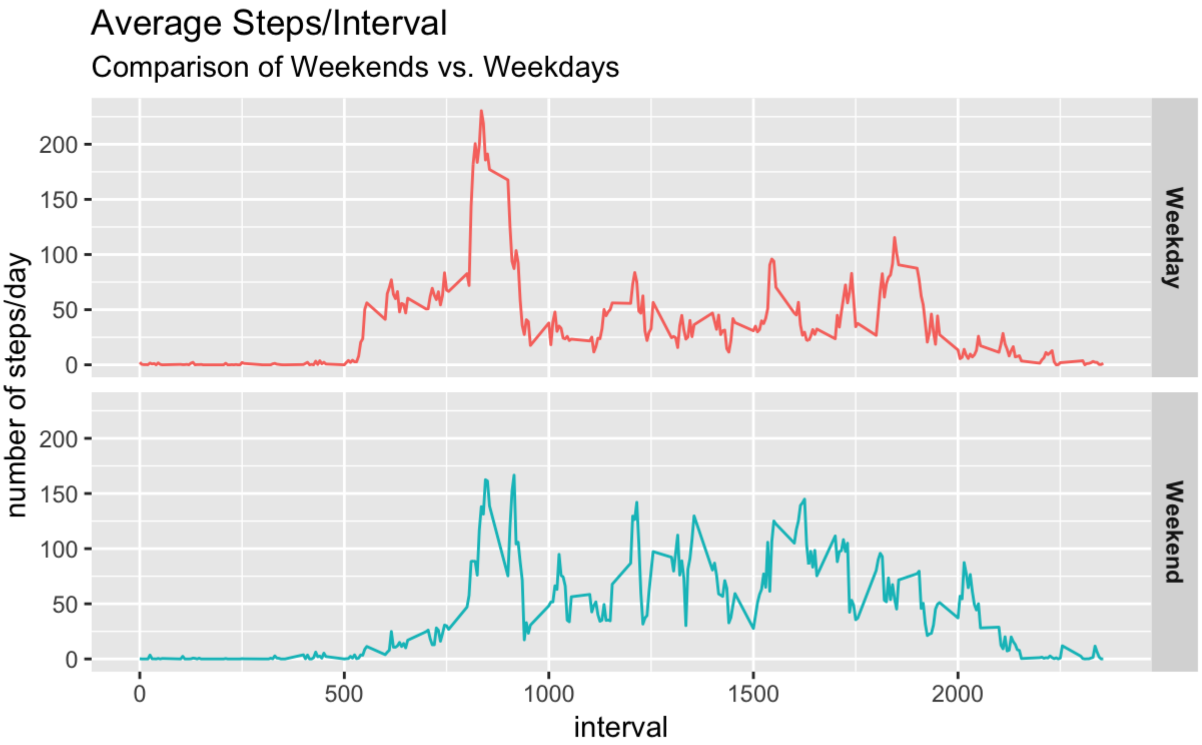
1. *Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.*
2. *Make a panel plot containing a time series plot (i.e. - type="l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged 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.*

With the hint to consider the weekday() function, it was pretty straightfoward to add a 2-level factor ("Weekend", "Weekday") which we implemented with a simple: **IF** (saturday|sunday) **THEN** weekend **ELSE** weekday

```
activityImputed2 <- activityImputed
activityImputed2$weekDayEnd <- ifelse(weekdays(as.Date(activityImputed2$date)) %in% c("S
aturday", "Sunday"),"Weekend", "Weekday" )
activityImputed2$weekDayEnd <- as.factor(activityImputed2$weekDayEnd)
```

And then reshaped the data around steps by interval (similar to above) and created the comparison plot using ggplot:

```
stepsPerDayImputed2 <- aggregate(steps ~ interval + weekDayEnd, activityImputed2, mean)

pl <- ggplot(filter(stepsPerDayImputed2, weekDayEnd %in% c("Weekend", "Weekday")),
             aes(x=as.numeric(as.character(interval)), y=steps, color=weekDayEnd) ) +
        geom_line() +
        facet_grid(weekDayEnd ~ ., scales="fixed", space="fixed") +
        labs(x="interval",
             y="number of steps/day",
             color= "Part of Week",
             title="Average Steps/Interval",
             subtitle="Comparison of Weekends vs. Weekdays"
             ) +
        theme(
                legend.position = "none",
                plot.margin = unit(c(1.5,1,1.5,1), "cm"),
                strip.text.y = element_text(face="bold")
                )
print(pl)
```



From the plots we observed the following:

- on the weekend start-of-day activity is much lower and ramps more gradually
  - …and the morning peak (around 800) is lower *(our guess: no commute)*
- on the weekend mid-day activity is significantly higher (roughly twice as high compared to weekdays)
- on the weekend there is a gradual rolloff at end-of-day
  - …with no peak (around 1800) *(our guess: no commute)*