Peer-graded Assignment: Course Project 1

PA1_template.Rmd

Teo Lo Piparo

Introduction

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 data for this assignment can be downloaded from the course web site:

Dataset: Activity monitoring data [52K]

The variables included in this dataset are:

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

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

Loading and preprocessing the data

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

```
fileUrl <-
    "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(fileUrl, destfile = "./repdata%2Fdata%2Factivity.zip", mode = "wb")
## Unzip file
unzip(
    "./repdata%2Fdata%2Factivity.zip",
    exdir = "./Unzipped"
)
## Read File
csv0 <- read.csv("./Unzipped/activity.csv", header = TRUE, sep = ",")
str(head(csv0))
```

```
## 'data.frame':    6 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25
```

What is mean total number of steps taken per day?

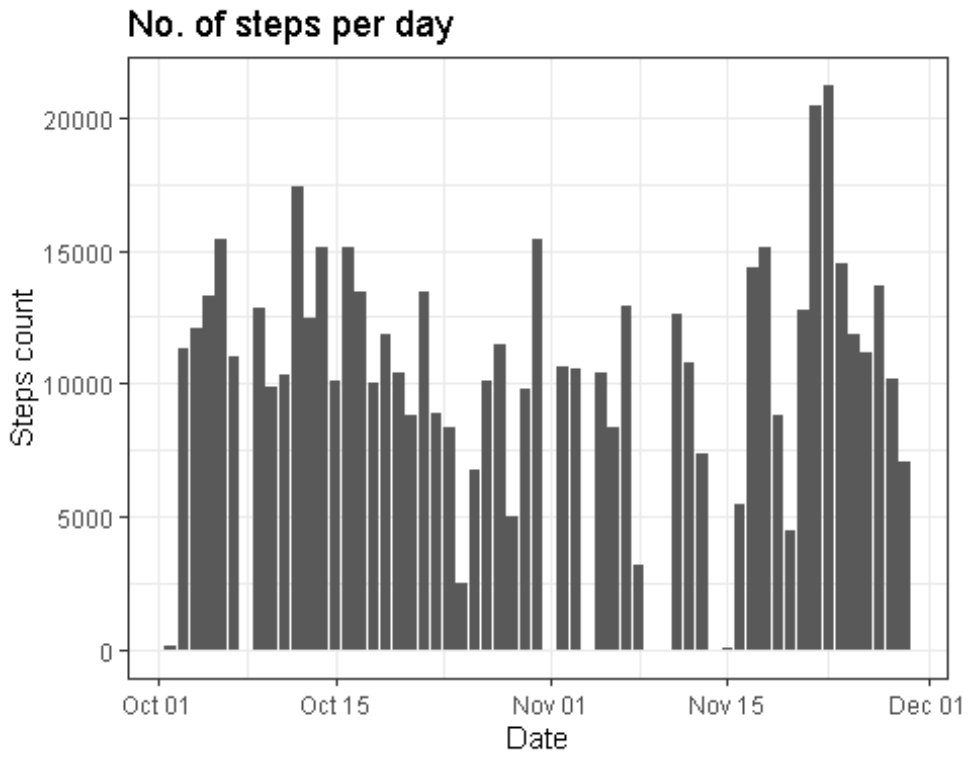
For this part of the assignment, you can ignore the missing values in the dataset.

- Calculate the total number of steps taken per day.

```
Sys.setlocale("LC_ALL", "English")
```

```
## [1] "LC_COLLATE=English_United States.1252;LC_CTYPE=English_United
States.1252;LC_MONETARY=English_United
States.1252;LC_NUMERIC=C;LC_TIME=English_United States.1252"
```

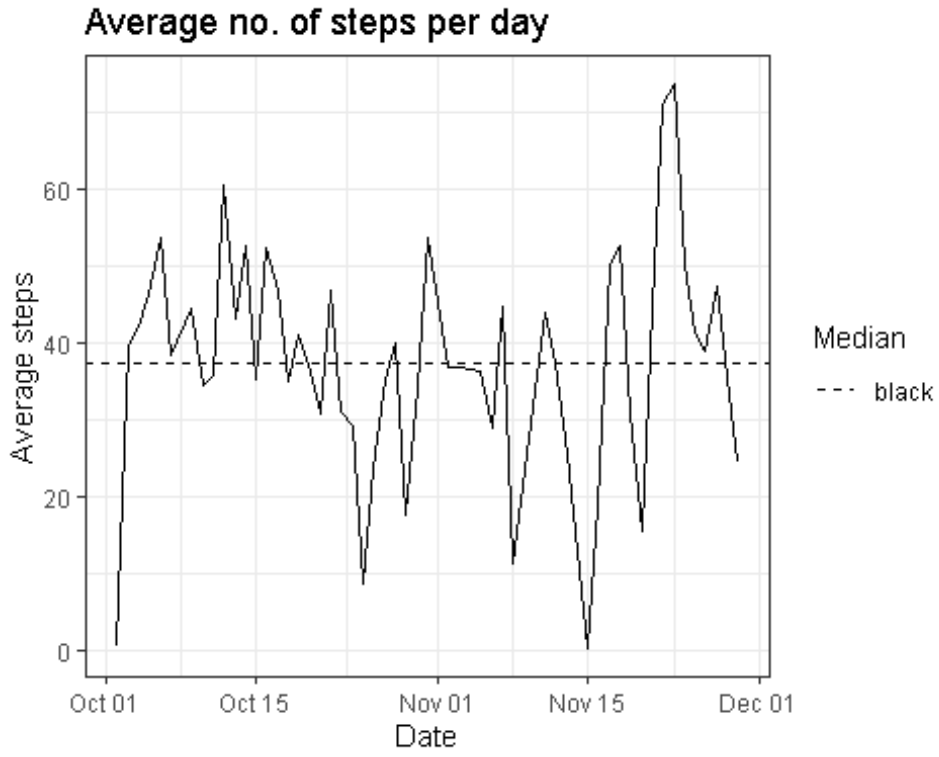
```
library(ggplot2)
step0 <- aggregate(steps~.,data = csv0, sum)
step0 <- step0[,c(1,3)] ;
step0[,1] <- as.Date(step0$date)
```



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.

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

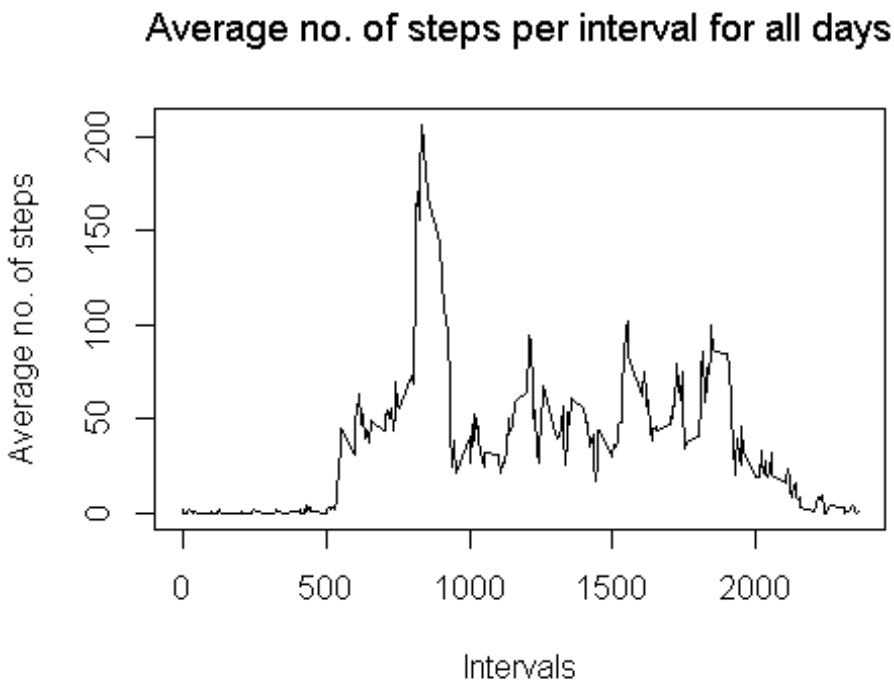
```
step1 <- aggregate(steps ~ ., data = step0, mean)
step2 <- median(step1[,2],na.rm = T)
```



What is the average daily activity pattern?

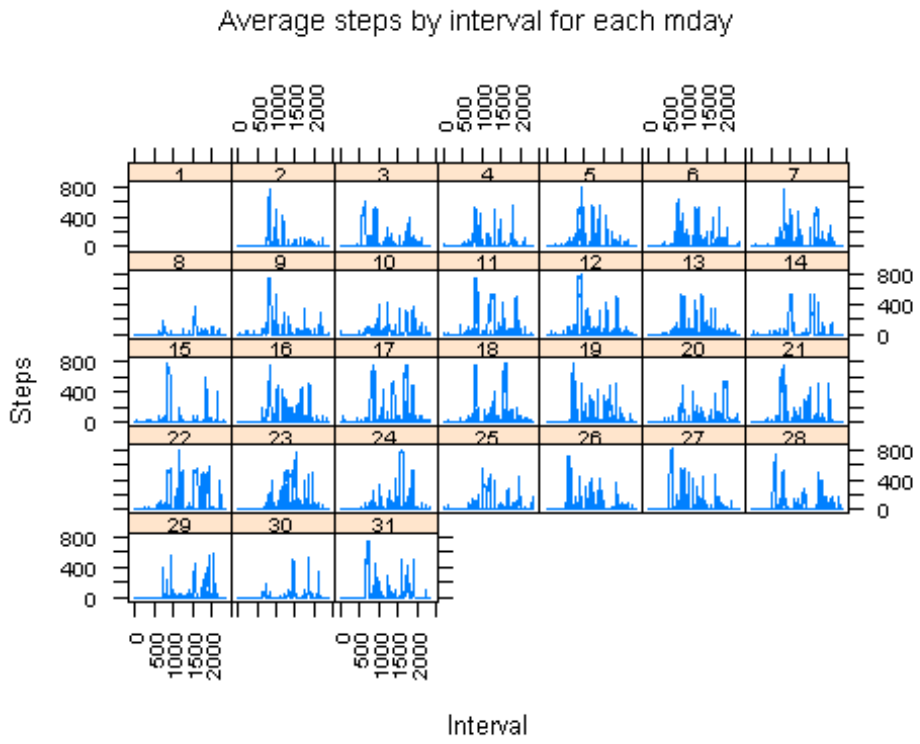
- Make a time series plot (i.e. `type = "l"`) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis).

```
step3 <- aggregate(steps ~ interval, data = csv0, mean)
step4 <- aggregate(steps ~ date, data = csv0, mean)
```



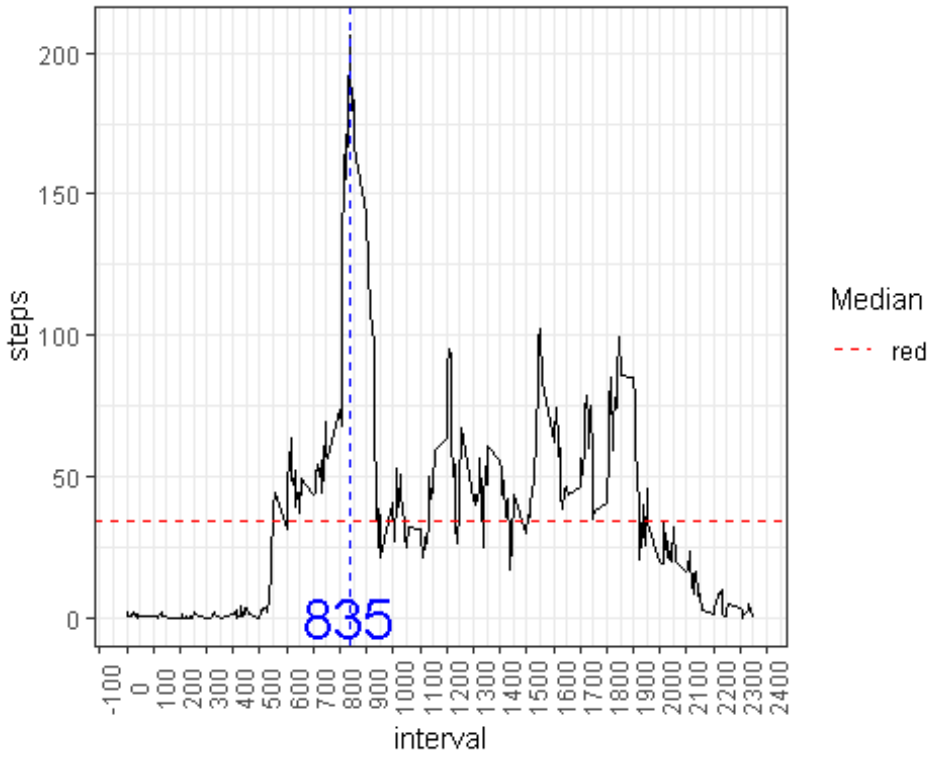
Another way of answering the above question:

```
##
## Attaching package: 'lubridate'

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



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

```
library(ggplot2)
step5 <- median(step3[,2],na.rm = T)
step6 <- subset(step3$interval, step3$steps == max(step3$steps, na.rm = T))
```

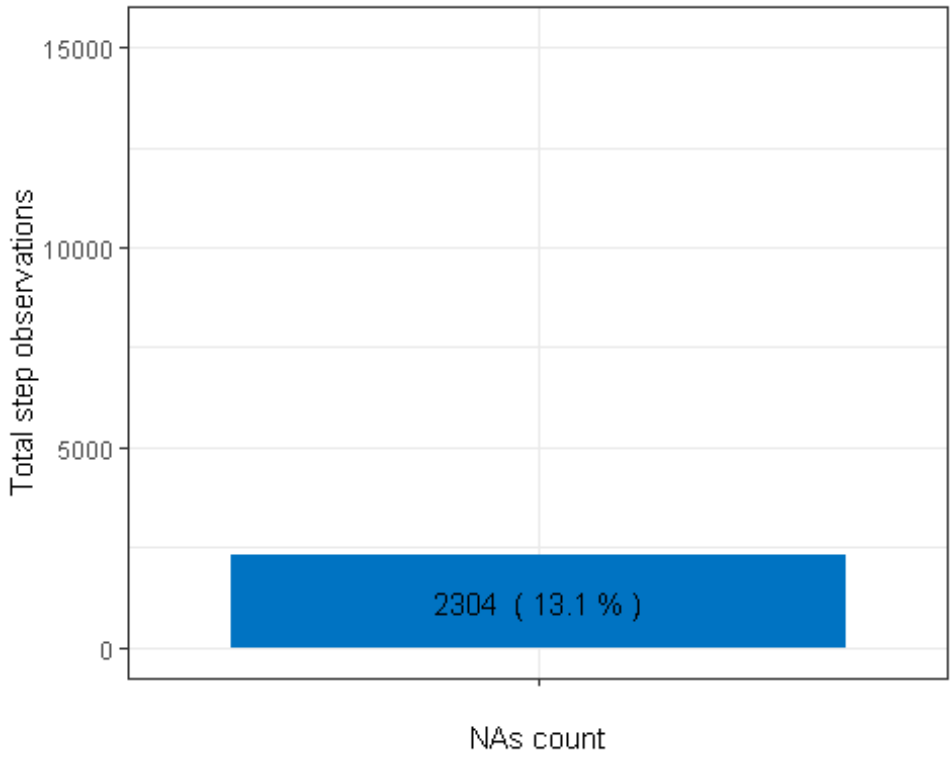


Imputing missing values

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.

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

```
library(data.table)

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

library(ggplot2)
step6 <- table(sum(is.na(csv0)))
step7 <- data.frame("NAs Steps"=csv0$steps, "NAs dates"=csv0$date, "NAs
intervals"=csv0$interval)
step7b <- data.frame(sapply(step7,function (x) sum(is.na(x)), simplify = T)) ;
step7b <- transpose(step7b)
colnames(step7b) <- c("NAsSteps", "NAsDates", "NAsIntervals")
```



- 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.

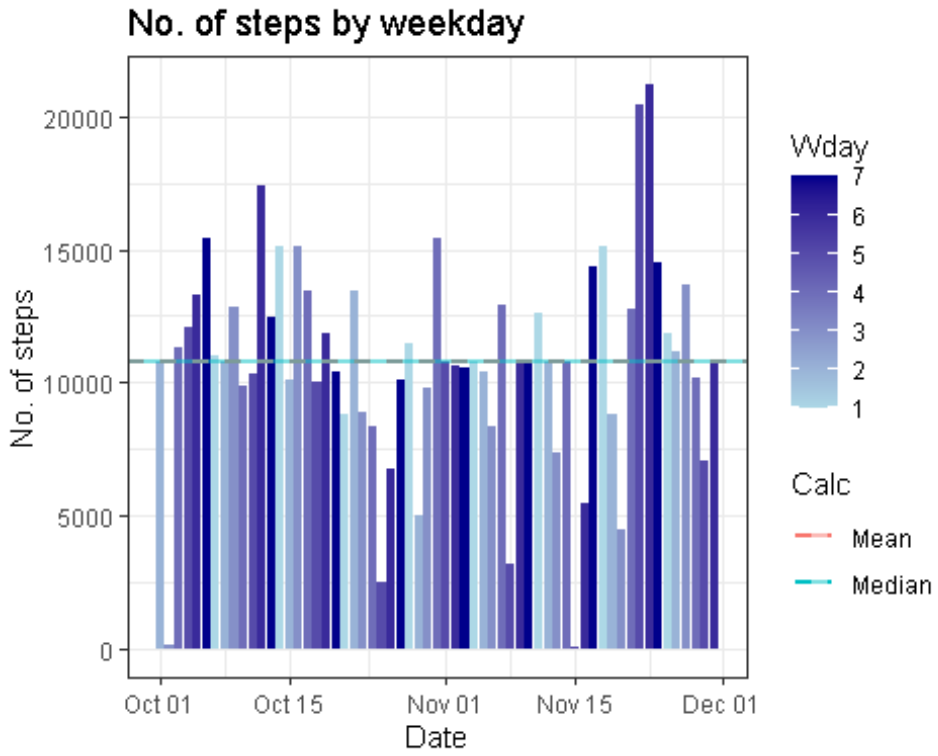
```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

step8 <- csv0
step8 <- step8 %>%
  ## Grouping by intervals
  group_by(interval) %>%
  ## Applying grouped mean for each respective NA
  mutate(steps = ifelse(is.na(steps), mean(steps, na.rm=T), steps))
  ## Rounding observations as integers
step8$steps <- format(round(step8$steps)); step8$steps <- as.integer(step8$steps)
```

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

```
step9 <- merge(csv0, step8, by = c("date","interval"))
step9 <- step9[,c(1,2,4)];
colnames(step9) <- c("date","interval","steps")
```

- 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.

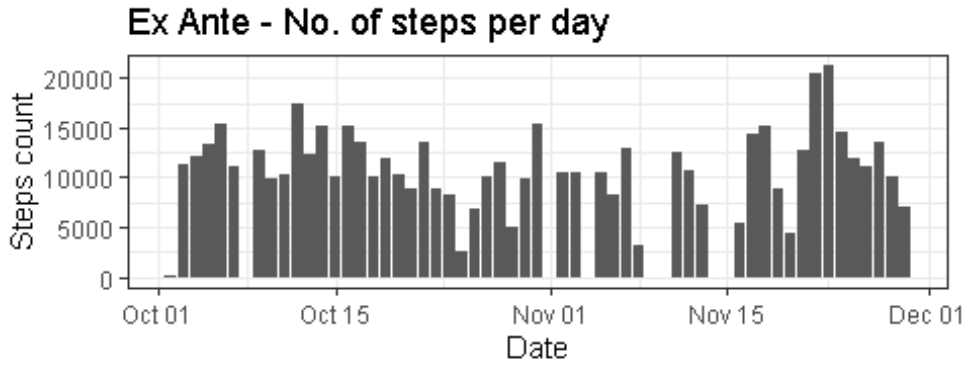
```
library(lubridate)
library(ggplot2)
step10 <- aggregate(steps ~ date, data=step9, sum)
step10mean <- mean(step10$steps)
step10median <- median(step10$steps)
```

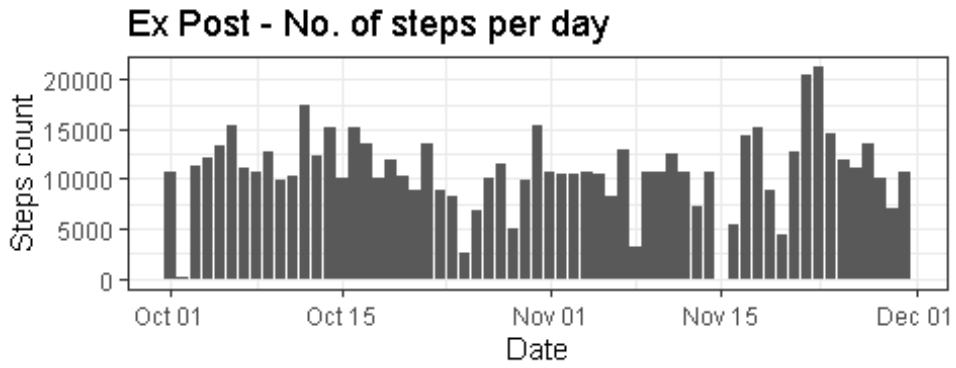


- Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

Both graphs presents equal values, except for the missing ones, however, the mean & median are affected making them similar to each other.

```
library(gridExtra)

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

grid.arrange(ante,post)
```





Are there differences in activity patterns between weekdays and weekends?

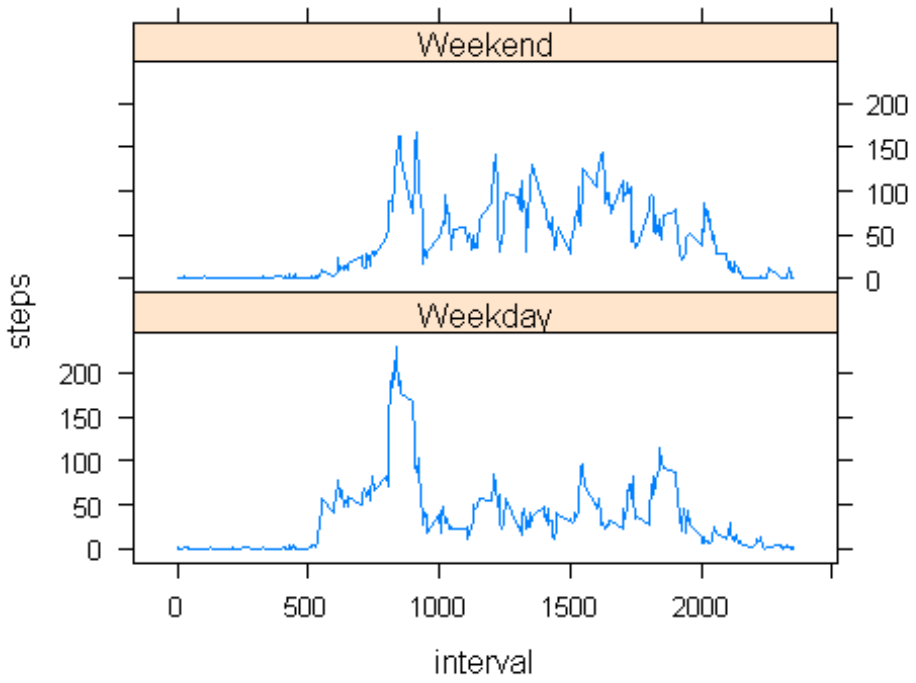
For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

- 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.

```
## [1] "LC_COLLATE=English_United States.1252;LC_CTYPE=English_United
States.1252;LC_MONETARY=English_United
States.1252;LC_NUMERIC=C;LC_TIME=English_United States.1252"
```

- 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.

```
library(lattice)
steptmp <- aggregate(steps ~ interval + wday, data=step9, mean)

xyplot(steps ~ interval | wday,type= "l", data= steptmp, layout=1:2)
```


Weekend
Weekday
steps
interval
0
50
100
150
200
0
500
1000
1500
2000