PA1_template

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.

Data

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.

Assignment

This assignment will be described in multiple parts. You will need to write a report that answers the questions detailed below. Ultimately, you will need to complete the entire assignment in a single R markdown document that can be processed by knitr and be transformed into an HTML file.

Throughout your report make sure you always include the code that you used to generate the output you present. When writing code chunks in the R markdown document, always use echo = TRUE so that someone else will be able to read the code. This assignment will be evaluated via peer assessment so it is essential that your peer evaluators be able to review the code for your analysis.

For the plotting aspects of this assignment, feel free to use any plotting system in R (i.e., base, lattice, ggplot2)

Fork/clone the GitHub repository created for this assignment. You will submit this assignment by pushing your completed files into your forked repository on GitHub. The assignment submission will consist of the URL to your GitHub repository and the SHA-1 commit ID for your repository state.

NOTE: The GitHub repository also contains the dataset for the assignment so you do not have to download the data separately.

Loading and preprocessing the data

Show any code that is needed to

Load the data (i.e. read.csv())

```
setwd("C:/Users/CLeibold/Desktop/R Files")
rm(list=ls())
activity <- read.csv("./activity.csv",colClasses = c("numeric", "character","integer"))
```

Process/transform the data (if necessary) into a format suitable for your analysis

```
dim(activity)
```

```
## [1] 17568     3
```

```
head(activity)
```

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

```
tail(activity)
```

```
##       steps       date interval
## 17563    NA 2012-11-30     2330
## 17564    NA 2012-11-30     2335
## 17565    NA 2012-11-30     2340
## 17566    NA 2012-11-30     2345
## 17567    NA 2012-11-30     2350
## 17568    NA 2012-11-30     2355
```

```
summary(activity)
```

```
##      steps            date              interval     
##  Min.   :  0.00   Length:17568       Min.   :   0.0  
##  1st Qu.:  0.00   Class :character   1st Qu.: 588.8  
##  Median :  0.00   Mode  :character   Median :1177.5  
##  Mean   : 37.38                      Mean   :1177.5  
##  3rd Qu.: 12.00                      3rd Qu.:1766.2  
##  Max.   :806.00                      Max.   :2355.0  
##  NA's   :2304
```

```
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  NA NA NA NA 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 30 35 40 45 ...
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 3.2.3
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.3
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.2.3
```

```
##
## Attaching package: 'lubridate'
##
## The following object is masked from 'package:plyr':
##
##     here
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.3
```

```
total.steps <- tapply(activity$steps, activity$date, FUN = sum, na.rm = TRUE)
activity$date <- ymd(activity$date)
```

Part One

What is mean total number of steps taken per day? Calculate and report the mean and median of the total number of steps taken per day.

```
mean(total.steps)
```

```
## [1] 9354.23
```

```
median(total.steps)
```

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

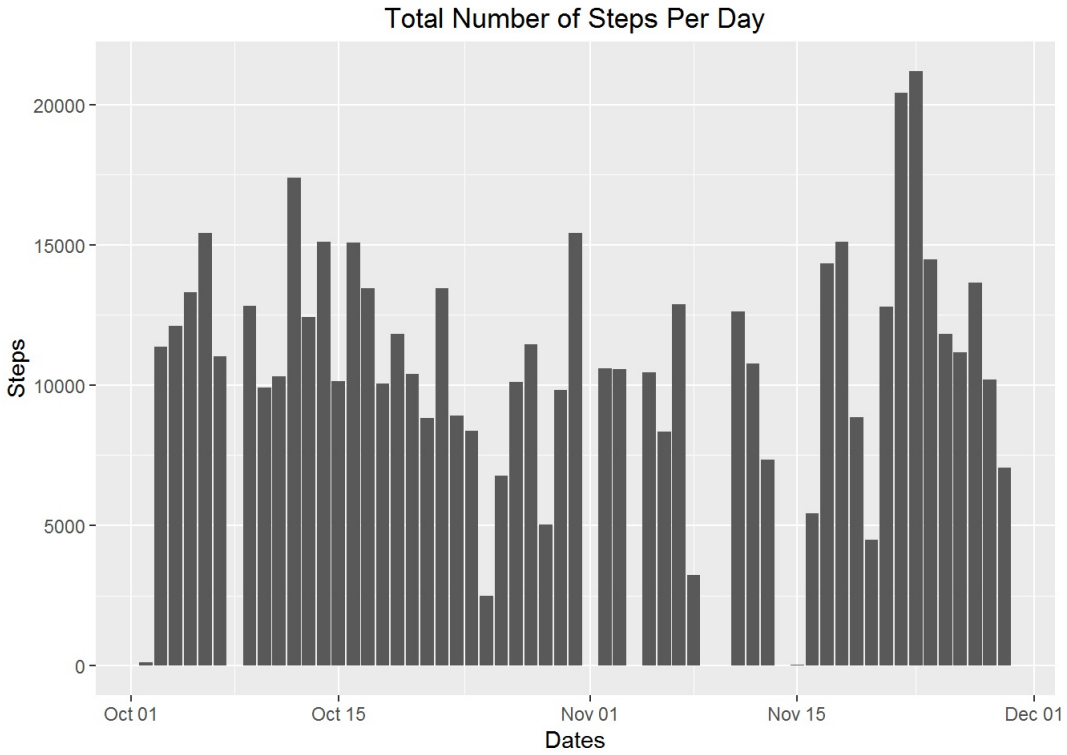
Calculate the total number of steps taken per day

```
steps <- activity %>%
  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(steps = sum(steps)) %>%
  print
```

```
## Source: local data frame [53 x 2]
##
##          date steps
##        (time) (dbl)
## 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
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## ..        ...   ...
```

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

```
ggplot(steps, aes(x=date, y=steps))+geom_bar(stat="identity")+ xlab("Dates")+ ylab("Steps")+ labs(title= "Total
 Number of Steps Per Day")
```



Part Two

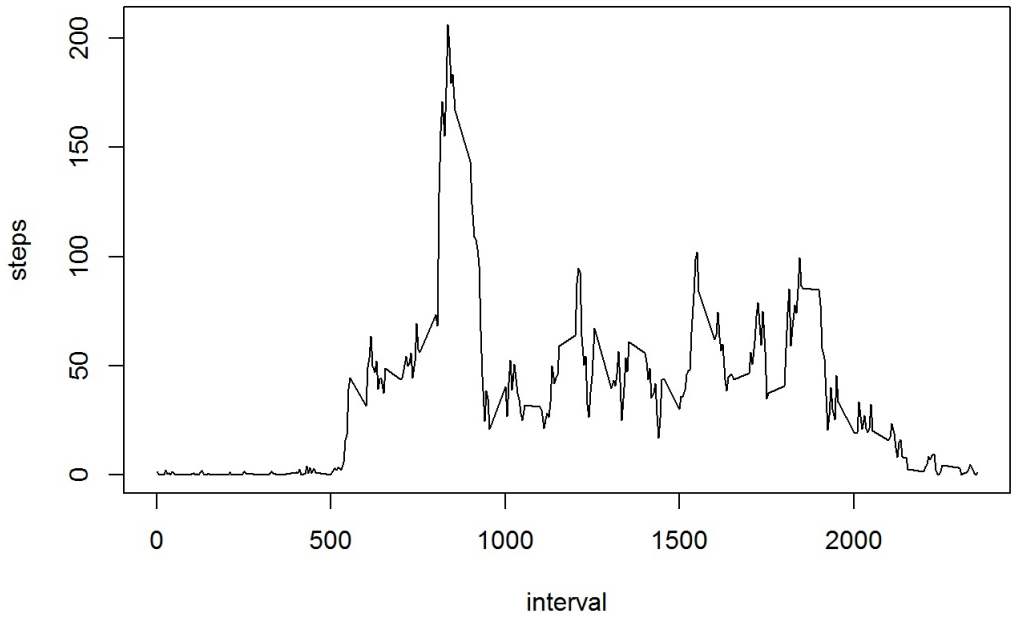
What is the average daily activity pattern?

```
daily <- activity %>%
        filter(!is.na(steps)) %>%
        group_by(interval) %>%
        summarize(steps=mean(steps)) %>%
        print
```

```
## Source: local data frame [288 x 2]
##
##    interval     steps
##       (int)     (dbl)
## 1         0 1.7169811
## 2         5 0.3396226
## 3        10 0.1320755
## 4        15 0.1509434
## 5        20 0.0754717
## 6        25 2.0943396
## 7        30 0.5283019
## 8        35 0.8679245
## 9        40 0.0000000
## 10       45 1.4716981
## ..      ...       ...
```

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)

```
plot(daily, type = "l")
```



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

```
daily[which.max(daily$steps), ]$interval
```

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

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)

```
missing <- sum(is.na(activity))
print(missing)
```

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

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

```
new <- activity %>%
        group_by(interval) %>%
        mutate(steps = ifelse(is.na(steps), mean(steps, na.rm=TRUE), steps))
summary(new)
```

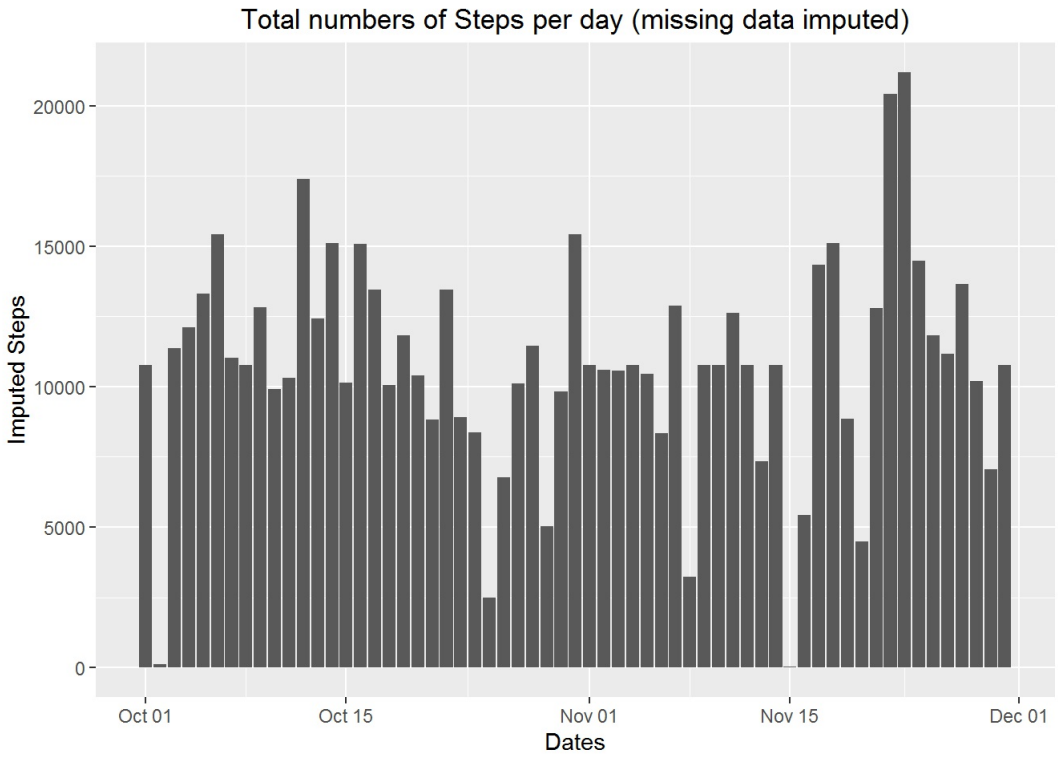
```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 27.00   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

```
new.steps <- new %>%
  group_by(date) %>%
  summarize(steps = sum(steps)) %>%
  print
```

```
## Source: local data frame [61 x 2]
##
##          date    steps
##        (time)    (dbl)
## 1  2012-10-01 10766.19
## 2  2012-10-02   126.00
## 3  2012-10-03 11352.00
## 4  2012-10-04 12116.00
## 5  2012-10-05 13294.00
## 6  2012-10-06 15420.00
## 7  2012-10-07 11015.00
## 8  2012-10-08 10766.19
## 9  2012-10-09 12811.00
## 10 2012-10-10  9900.00
## ..        ...      ...
```

```
ggplot(new.steps, aes(x=date, y=steps))+geom_bar(stat="identity")+ xlab("Dates")+ ylab("Imputed Steps")+ labs(t
itle= "Total numbers of Steps per day (missing data imputed)")
```



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

```
imputed.steps <- tapply(new$steps, new$date, FUN = sum, na.rm = TRUE)
new$date <- ymd(new$date)
mean(imputed.steps)
```

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

```
median(imputed.steps)
```

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

Do these values differ from the estimates from the first part of the assignment?

```
mean(total.steps)==mean(imputed.steps)
```

```
## [1] FALSE
```

```
median(total.steps)==median(imputed.steps)
```

```
## [1] FALSE
```

```
summary(total.steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    6778   10400    9354   12810   21190
```

```
summary(imputed.steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    9819   10770   10770   12810   21190
```

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

```
summary(imputed.steps) - summary(total.steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      41    3041     370    1416       0       0
```

The estimates of the number of steps increased by 41, 3041, 370, 1416, 0, 0.

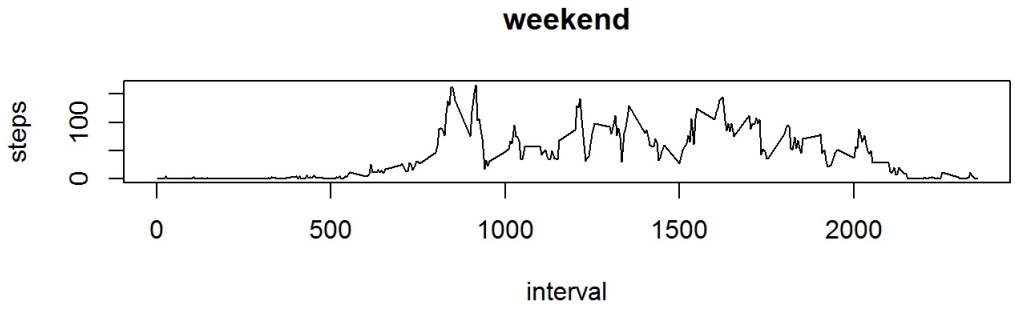
Part 3

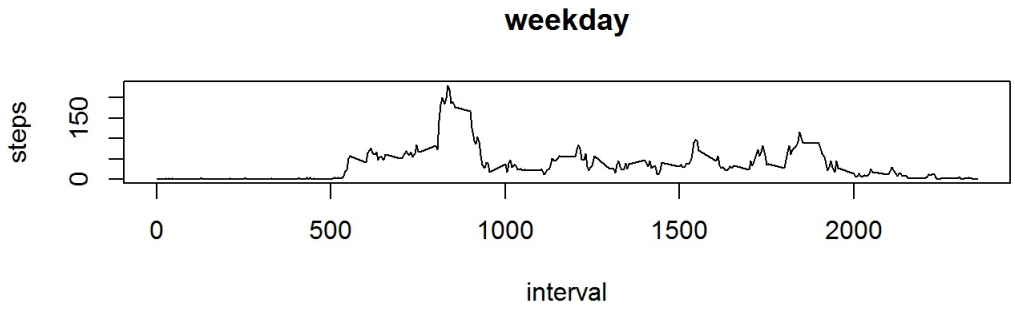
Are there differences in activity patterns between weekdays and weekends? For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part. Create a new factor variable in the dataset with two levels - “weekday” and “weekend” indicating whether a given date is a weekday or weekend day.

```
dayofweek <- function(date) {
    if (weekdays(as.Date(date)) %in% c("Saturday", "Sunday")) {
        "weekend"
    } else {
        "weekday"
    }
}
new$daytype <- as.factor(sapply(new$date, dayofweek))
```

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.#

```
par(mfrow = c(2, 1))
for (type in c("weekend", "weekday")) {
    steps.type <- aggregate(steps ~ interval, data = new, subset = new$daytype ==
        type, FUN = mean)
    plot(steps.type, type = "l", main = type)
}
```


weekend
steps
0
100
0
500
1000
1500
2000
interval


weekday
steps
0
150
0
500
1000
1500
2000
interval