ReproducibleResearch_CourseProject1

Jordan

January 1, 2017

Instructions

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.

```
activityData <- read.csv ("activity.csv", header = T, sep = ",", stringsAsFactors = F)
```

Convert Date Column to the Rigth Format

```
activityData$date <- as.Date(activityData$date, "%Y-%m-%d")
str(activityData)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Check New Data Frame

```
dim(activityData)
```

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

```
head(activityData)
```

```
##   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
```

ANALYSIS

Q1. What is mean total number of steps taken per day?

1. Calculate the total number of s teps per day Use dplyr to group and summarize the data and store it in the variable AvgDay (total number of steps per day and the mean number of daily steps)

```
library (dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2
```

```
##
## Attaching package: 'dplyr'
```

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

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

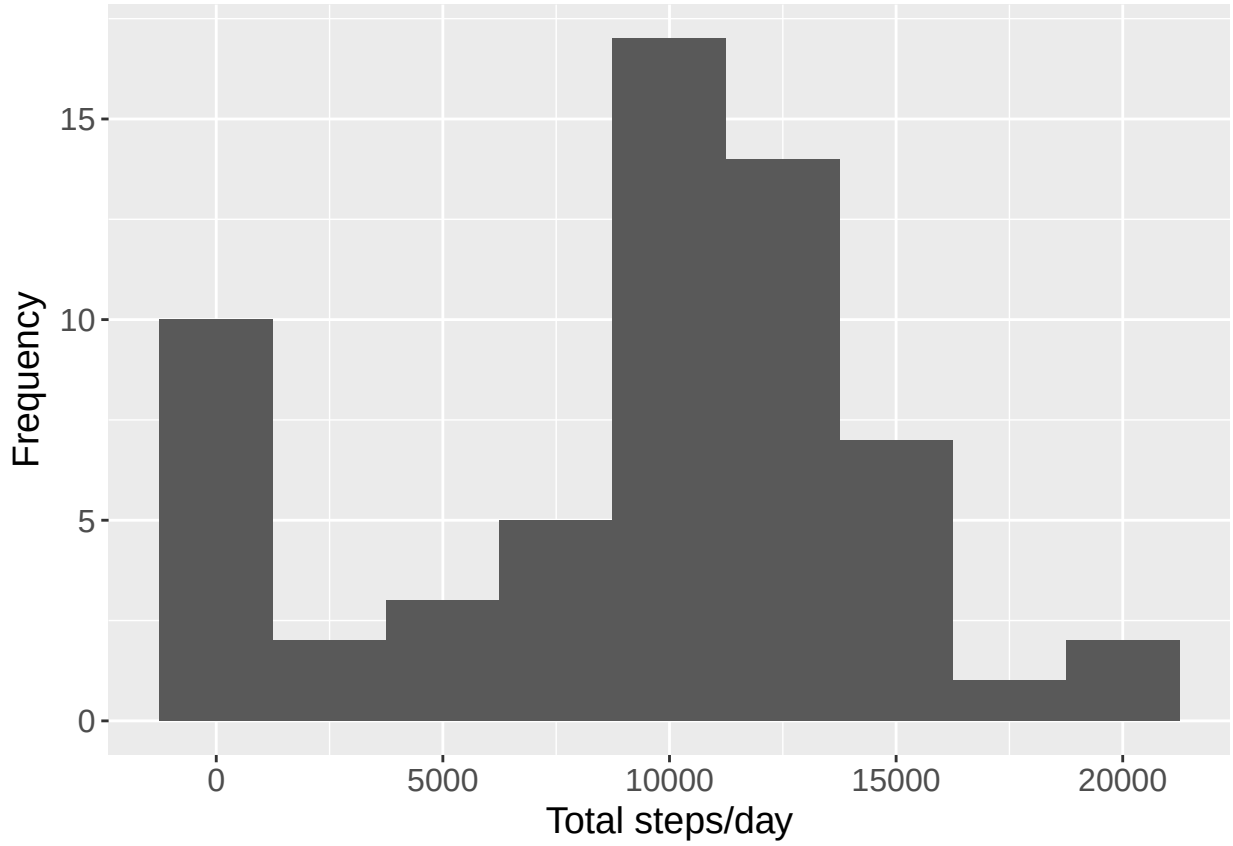
```
AvgDay <- activityData %>% group_by(date) %>%
          summarize(total.steps = sum(steps, na.rm = T),
                  mean.steps = mean(steps, na.rm = T))
```

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
g <- ggplot(AvgDay, aes(x=total.steps))
g + geom_histogram(binwidth = 2500) + theme(axis.text = element_text(size = 12),
      axis.title = element_text(size = 14)) + labs(y = "Frequency") + labs(x = "Total steps/day")
```



The histogram suggests that the median will be in the 10k-12k steps/day neighborhood. Aside from a lot of people who walk very little, the data is symetrically distributed around this median.

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

```
summary(AvgDay$total.steps)
```

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

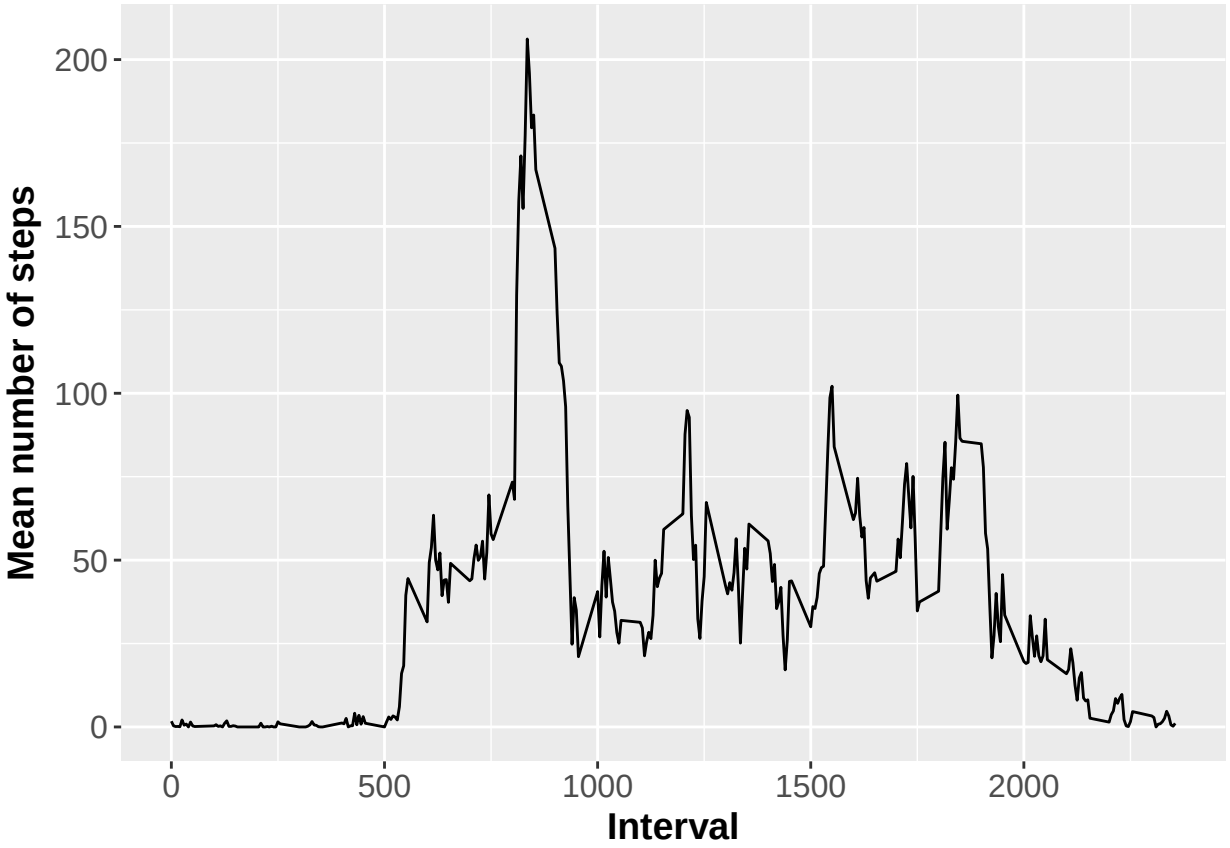
```
summary (AvgDay$mean.steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.1424 30.7000 37.3800 37.3800 46.1600 73.5900       8
```

Q2. What is the average daily activity pattern?

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)

```
AvgInterval <- activityData %>% group_by(interval) %>%
      summarize(mean.steps = mean(steps, na.rm = T))
g <- ggplot(AvgInterval, aes(x = interval, y = mean.steps))
g + geom_line() + theme(axis.text = element_text(size = 12),
      axis.title = element_text(size = 14, face = "bold")) +
      labs(y = "Mean number of steps") + labs(x = "Interval")
```



2. Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps? The highest step count occurs between time intervals 500 and 1000. The maximum average number of steps is: 206 and occurs in time interval #835

Q3. 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.

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

```
mean(is.na(activityData$steps))
```

```
## [1] 0.1311475
```

```
sum(is.na(activityData$steps))
```

```
## [1] 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. *A bit more than 13% of the data is missing. That is a lot, we should test the effect of imputing.*
   Some options are:

- Using the average steps during the day to fill in NAs within the same day. BUT the time series shows significant variation over the course of the day and and there are eight full days with no data.
- Using the average steps per interval. This is probably a better techinque.
- First, we will check for missing values in the interval column within AvgInterval, where we stored the mean number of steps for each 5 min interval:

```
sum(is.na(AvgInterval$mean.steps))
```

```
## [1] 0
```

No missing values! Good basis for imputing: 3. Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
newData <- activityData
```

Quoting others: In order to fill in missing values we check at each row if the column interval is NA, when the condition is true we look for the corresponding interval (index), we search for this particular interval in the AvgInterval data and extract it to a temporary variable values. Last we choose only the column of interest from values, which is the mean.steps and assign this number to the corresponding position in the newData set. We use a for loop to run through all the rows. (Note: there may be a more elegant way to do this perhaps using apply but couldnt make it work)

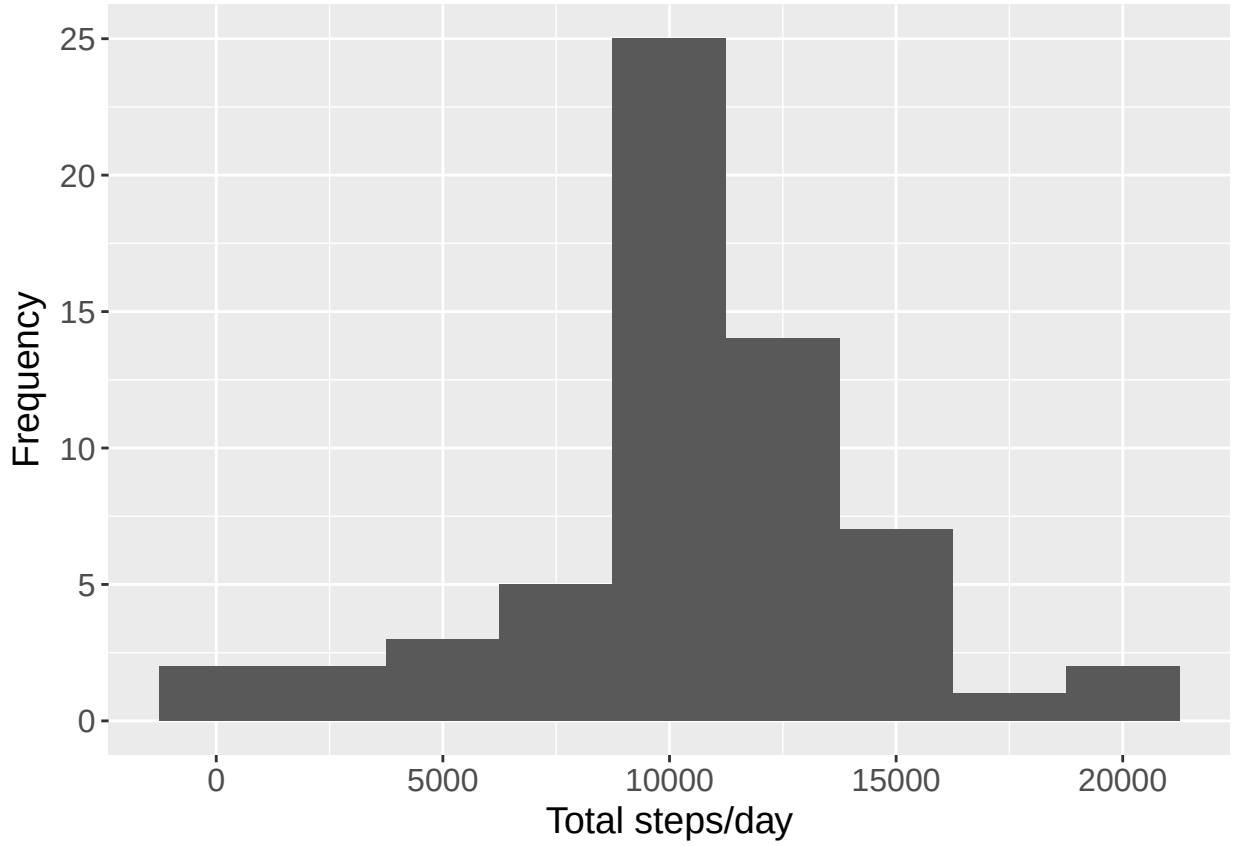
```
for (i in 1:nrow(newData)) {
      if (is.na(newData$steps[i])) {
            index <- newData$interval[i]
            value <- subset(AvgInterval, interval==index)
            newData$steps[i] <- value$mean.steps
      }
}
head(newData)
```

```
##       steps       date interval
## 1 1.7169811 2012-10-01        0
## 2 0.3396226 2012-10-01        5
## 3 0.1320755 2012-10-01       10
## 4 0.1509434 2012-10-01       15
## 5 0.0754717 2012-10-01       20
## 6 2.0943396 2012-10-01       25
```

```
newAvg <- newData %>% group_by(date) %>%
      summarize(total.steps = sum(steps, na.rm = T))
```

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?

```
g <- ggplot(newAvg, aes(x=total.steps))
g + geom_histogram(binwidth = 2500) + theme(axis.text = element_text(size = 12),
      axis.title = element_text(size = 14)) + labs(y = "Frequency") + labs(x = "Total steps/day")
```




```
summary (AvgDay$total.steps)
```

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

```
sd(AvgDay$total.steps, na.rm=T)
```

```
## [1] 5405.895
```

```
summary (newAvg$total.steps)
```

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

```
sd(newAvg$total.steps, na.rm=T)
```

```
## [1] 3974.391
```

No change to teh mean and median, but the 1st quantile of the new data is closer to the mean. The imputed data has a smaller standard deviation: the distribution is closer to the median.

Q4. 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.

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. Create a new column in newData containing the values weekend or weekday:

```
newData$day <- ifelse(weekdays(newData$date) %in% c("Saturday", "Sunday"), "weekend", "weekday")
```

Create two subsets, one containing the weekend and one containing the weekday data:

```
wkend <- filter(newData, day == "weekend")
wkday <- filter(newData, day == "weekday")
```

Group by the intervals and calculate the mean number of steps for each time interval. Since the day column is lost during the grouping, we add it again to the wkend and wday dataframes. Then merge both data frames into a new data frame named newInterval

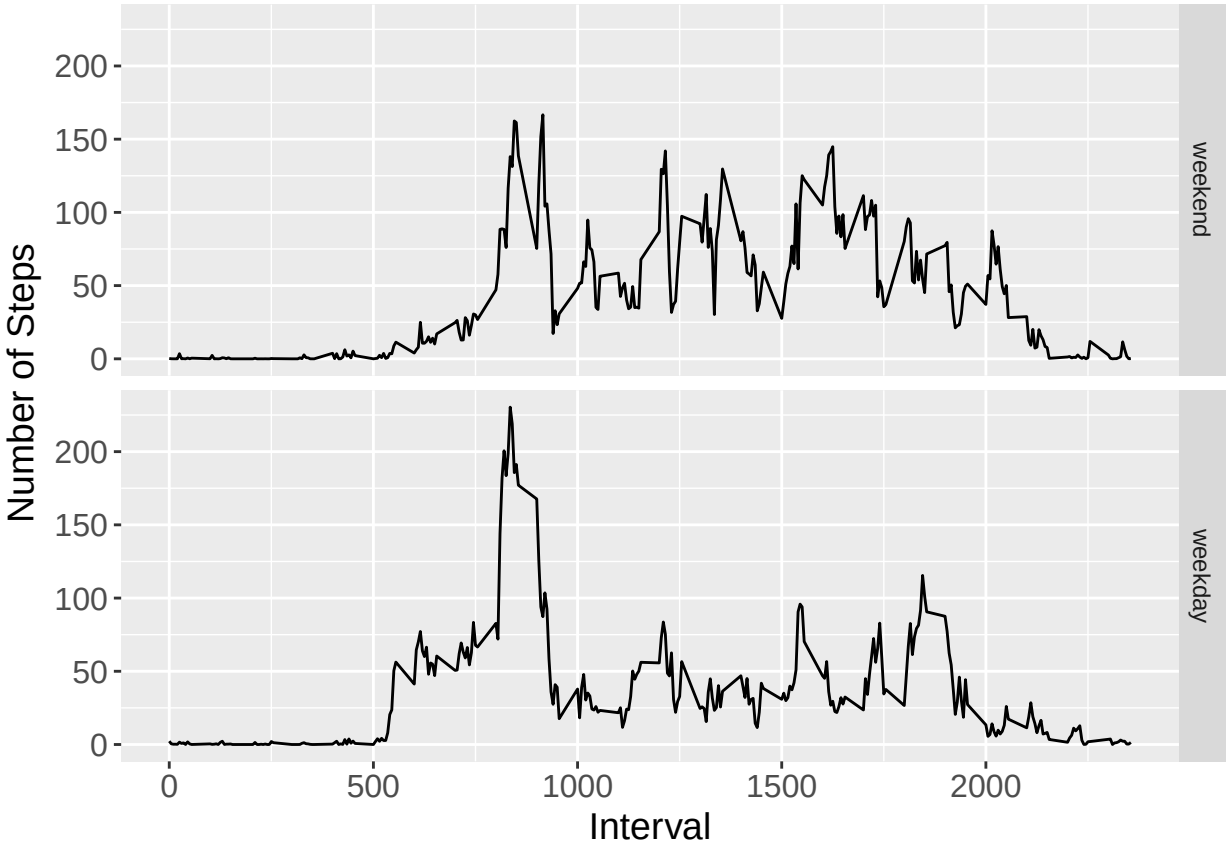
```
wkend <- wkend %>%
      group_by(interval) %>%
      summarize(mean.steps = mean(steps))
wkend$day <- "weekend"

wkday <- wkday %>%
      group_by(interval) %>%
      summarize(mean.steps = mean(steps))
wkday$day <- "weekday"

newInterval <- rbind(wkend, wkday)
newInterval$day <- as.factor(newInterval$day)
newInterval$day <- relevel(newInterval$day, "weekend")
```

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.

```
g <- ggplot (newInterval, aes (interval, mean.steps))
g + geom_line() + facet_grid (day~.) + theme(axis.text = element_text(size = 12),
      axis.title = element_text(size = 14)) + labs(y = "Number of Steps") + labs(x = "Interval")
```



So quite a bit off difference between weekends and weekdays in terms of activity. More activityy in the middle of the day on weekends - probably because people aren't sitting around at work...