PA1_template.Rmd

Victor Faria de Sousa

17 de setembro de 2019

Loading and preprocessing Data

Load and brief analysis and adjustment of the data

```
data <- read.csv("activity.csv")
#Data Adjustments
data$date <- as.Date(data$date)
#Creating a col for month factor
data$month <- as.numeric(format(data$date, "%m"))
#Removing NA Values
noNAData <- na.omit(data)
#after removing NA, adjusting rows sequence
rownames(noNAData) <- 1:nrow(noNAData)
# noNAData information
head(noNAData)
```

```
##   steps       date interval month
## 1     0 2012-10-02        0    10
## 2     0 2012-10-02        5    10
## 3     0 2012-10-02       10    10
## 4     0 2012-10-02       15    10
## 5     0 2012-10-02       20    10
## 6     0 2012-10-02       25    10
```

```
dim(noNAData)
```

```
## [1] 15264     4
```

```
# Complete data information:
dim(data)
```

```
## [1] 17568     4
```

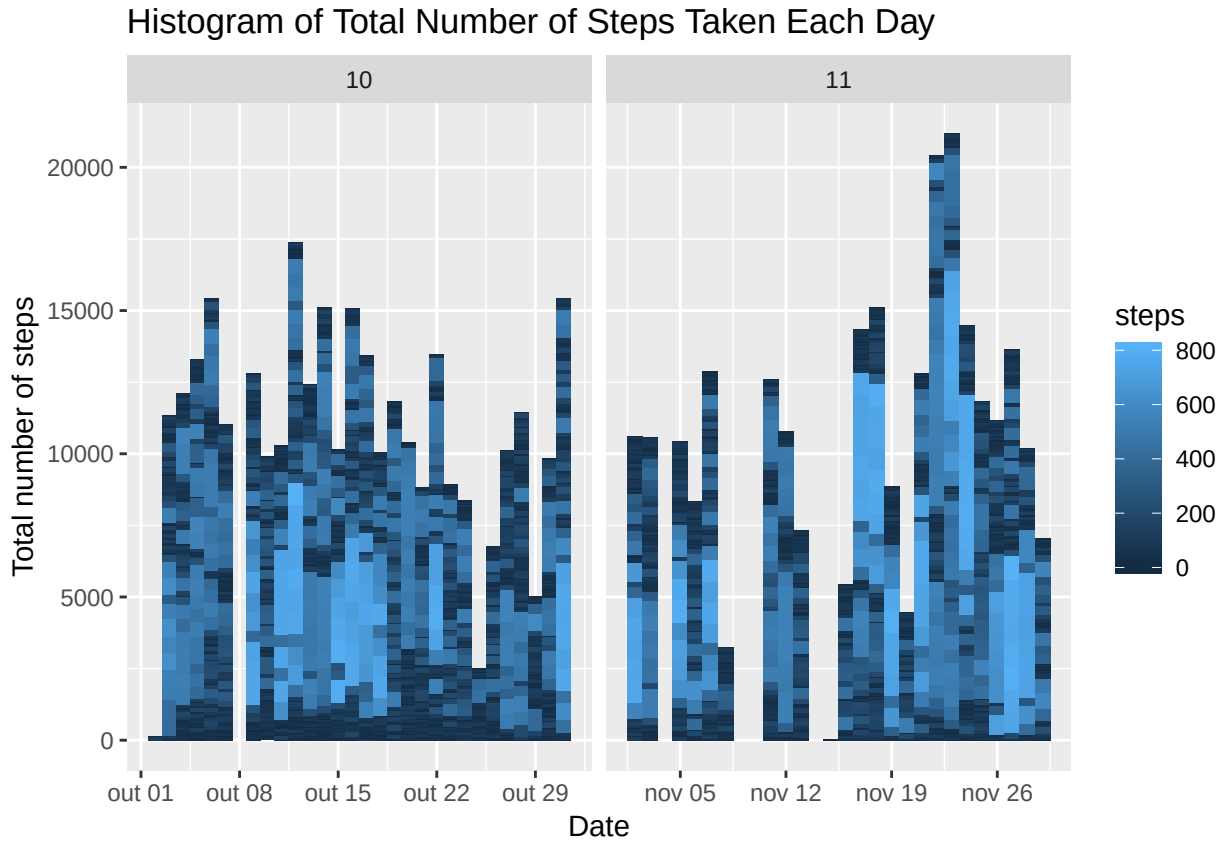
We are going to use the ggplot2 lib for plot:

```
library(ggplot2)
```

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

```
ggplot(noNAData, aes(date, steps, fill= steps)) +
            geom_bar(stat = "identity", width = 1) +
            facet_grid(. ~ month, scales = "free") +
            labs(title = "Histogram of Total Number of Steps Taken Each Day",
            x = "Date", y = "Total number of steps")
```



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

```
steps <- aggregate(noNAData$steps, list(Date = noNAData$date), FUN = "sum")$x
mean(steps)
```

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

- Median of total steps per day:

```
oldMedian <- median(steps)
median(steps)
```

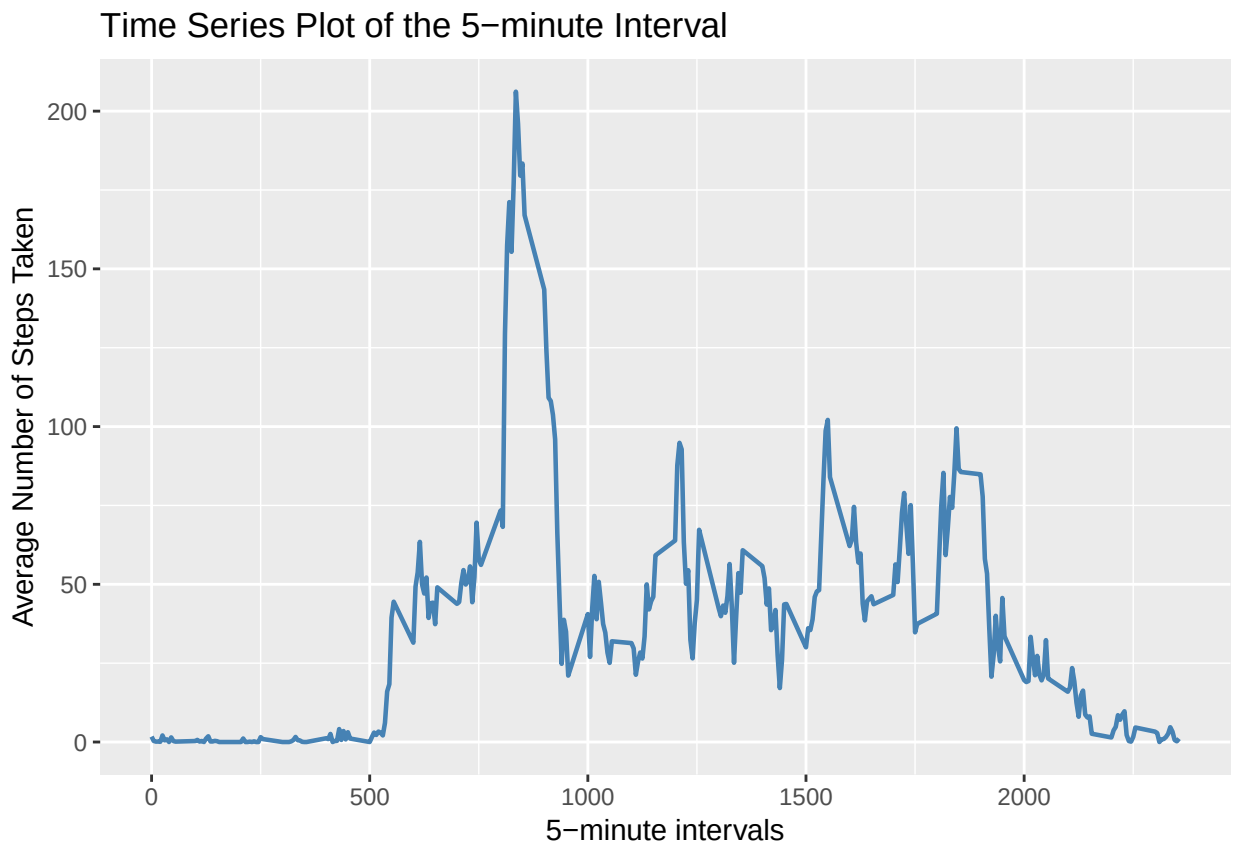
```
## [1] 10765
```

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)

```
averageSteps <- aggregate(noNAData$steps,
                  list(interval = as.numeric(as.character(noNAData$interval))),
                  FUN = "mean")

names(averageSteps)[2] <- "meanOfSteps"
ggplot(averageSteps, aes(interval, meanOfSteps)) +
            geom_line(color = "steelblue", size = 0.8) +
            labs(title = "Time Series Plot of the 5-minute Interval",
            x = "5-minute intervals", y = "Average Number of Steps Taken")
```



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

```
averageSteps[averageSteps$meanOfSteps == max(averageSteps$meanOfSteps), ]
```

```
##     interval meanOfSteps
## 104      835    206.1698
```

Imputing missing values

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

```
sum(is.na(data))
```

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

*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 this, i'll use the steps-mean to fill each step-NA.

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

```
newData <- data
for (i in 1:nrow(newData)) {
    if (is.na(newData$steps[i])) {
        newData$steps[i] <- averageSteps[which(newData$interval[i] == averageSteps$interval),
                                         ]$meanOfSteps
```

```
    }
}
head(newData)
```

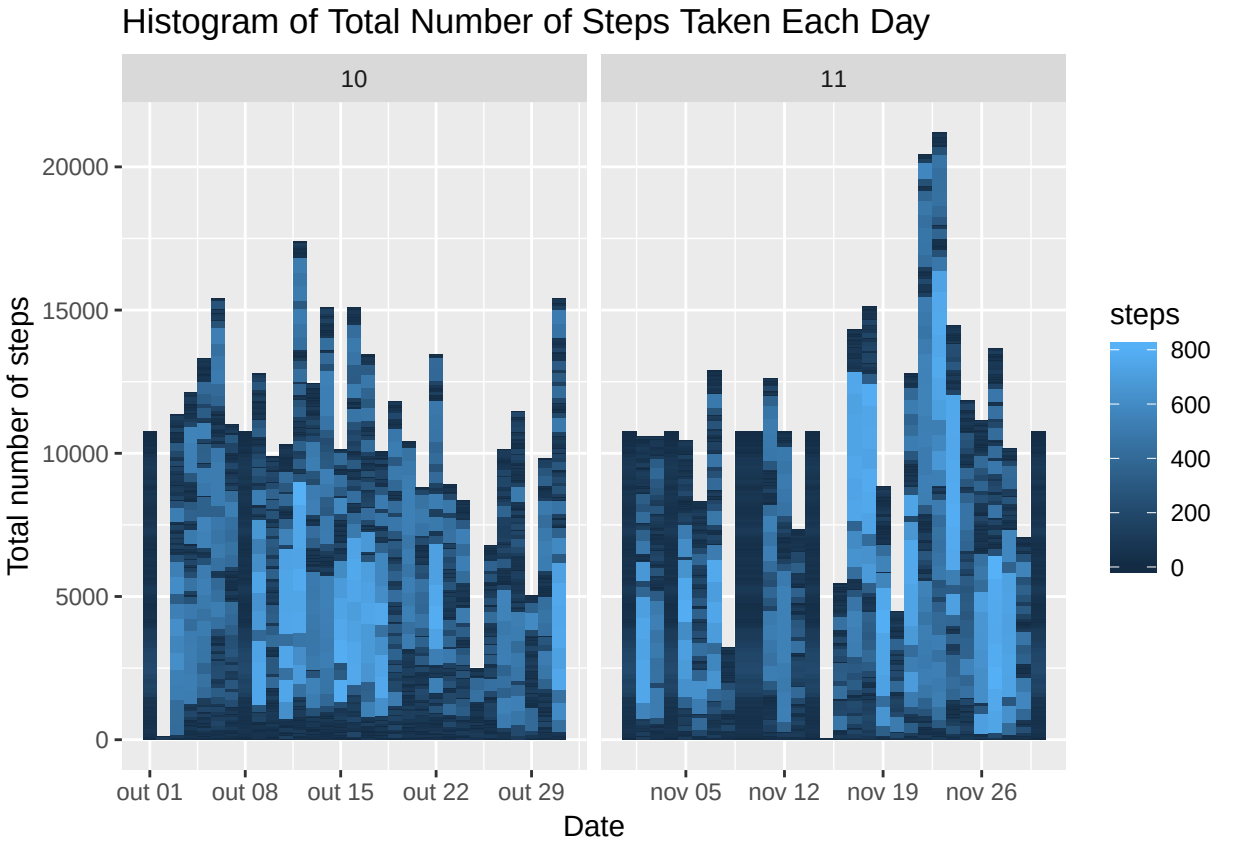
```
##       steps       date interval month
## 1 1.7169811 2012-10-01        0    10
## 2 0.3396226 2012-10-01        5    10
## 3 0.1320755 2012-10-01       10    10
## 4 0.1509434 2012-10-01       15    10
## 5 0.0754717 2012-10-01       20    10
## 6 2.0943396 2012-10-01       25    10
```

```
sum(is.na(newData))
```

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

- 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.

```
ggplot(newData, aes(date, steps, fill= steps)) +
            geom_bar(stat = "identity", width = 1) +
            facet_grid(. ~ month, scales = "free") +
            labs(title = "Histogram of Total Number of Steps Taken Each Day",
            x = "Date", y = "Total number of 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?

Mean total number of steps taken per day:

```
newSteps <- aggregate(newData$steps, list(Date = newData$date), FUN = "sum")$x
newMean <- mean(newSteps)
newMean
```

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

Median of total steps per day:

```
newMedian <- median(newSteps)
newMedian
```

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

Compare them with the two before imputing missing data:

```
oldMean <- mean(steps)
oldMedian <- median(steps)
newMean / oldMean
```

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

```
newMedian / oldMedian
```

```
## [1] 1.00011
```

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

The strategy for filling NA steps do not change de global mean; And the new median of total steps is slight increased.

Are there differences in activity patterns between weekdays and weekends?

- 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.

```
head(newData)
```

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

```
newData$weekdays <- factor(format(newData$date, "%w"))
levels(newData$weekdays)
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

```
levels(newData$weekdays) <- list(weekday = c("1", "2",
                                             "3",
                                             "4", "5"),
                                 weekend = c("6", "0"))
levels(newData$weekdays)
```

```
## [1] "weekday" "weekend"
```

```
table(newData$weekdays)
```

```
##
## weekday weekend
```

```
##    12960     4608
```

- 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).

```
finalAvgStp <- aggregate(newData$steps,
                         list(interval = as.numeric(as.character(newData$interval)),
                              weekdays = newData$weekdays),
                         FUN = "mean")
names(finalAvgStp)[3] <- "meanOfSteps"
library(lattice)
xyplot(finalAvgStp$meanOfSteps ~ finalAvgStp$interval | finalAvgStp$weekdays,
       layout = c(1, 2), type = "l",
       xlab = "Interval", ylab = "Number of steps")
```