PA1_template.rmd

Shujuan Huang

Friday, December 18, 2015

Loading and preprocessing the data

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

```
packages <- c("plyr", "lattice", "data.table", "ggplot2")
lapply(packages, require, character.only = TRUE)
```

```
## Loading required package: plyr
## Loading required package: lattice
## Loading required package: data.table
## Loading required package: ggplot2

## [[1]]
## [1] TRUE
##
## [[2]]
## [1] TRUE
##
## [[3]]
## [1] TRUE
##
## [[4]]
## [1] TRUE
```

```
setwd("C:/Users/shujuan/Desktop/coursera/reproductive research/repdata_data_activity")
activity<-read.csv("activity.csv", header=TRUE,sep=",")
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
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  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: int  0 5 10 15 20 25 30 35 40 45 ...
```

What is mean total number of steps taken per day?

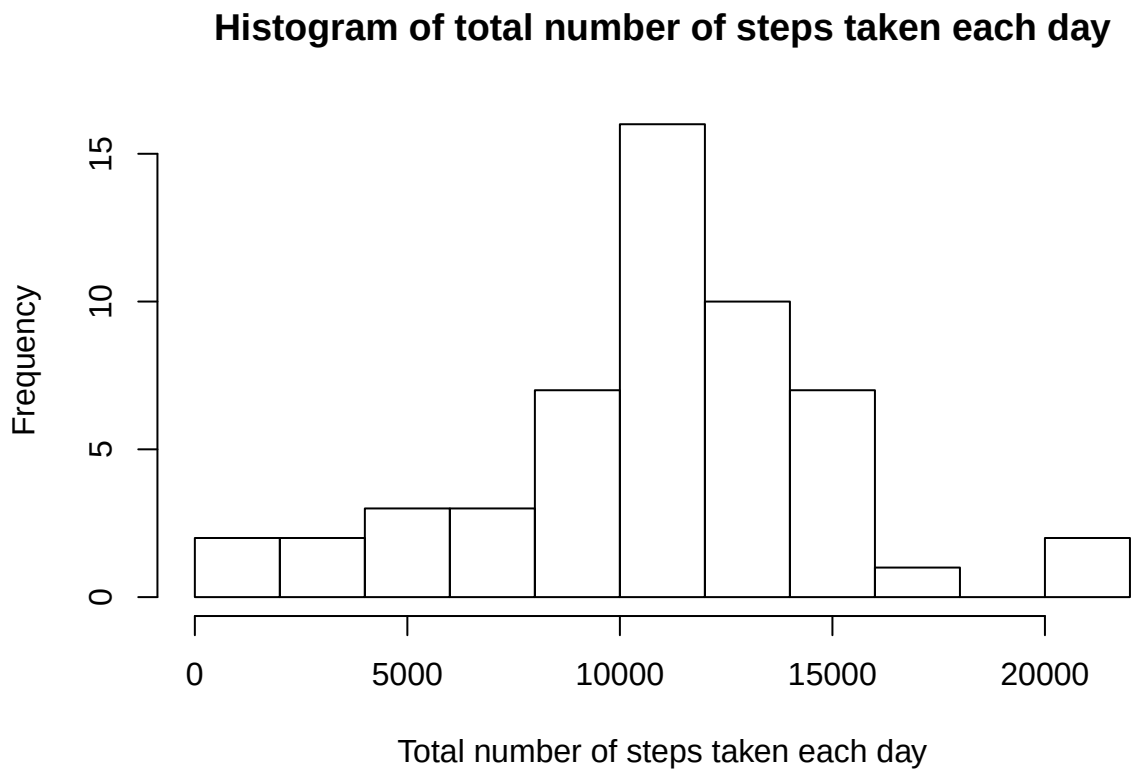
For this part of the assignment, you can ignore the missing values in the dataset.

1.Calculate the total number of steps taken per day

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

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

```
stepbydate <- setNames(aggregate(steps~as.Date(date), activity, sum, na.rm = TRUE),
                       c("date","steps"))
hist(stepbydate$steps, main="Histogram of total number of steps taken each day",
     xlab="Total number of steps taken each day",breaks=15)
```




```
paste("mean:", mean(stepbydate$steps))
```

```
## [1] "mean: 10766.1886792453"
```

```
paste("median:", median(stepbydate$steps))
```

```
## [1] "median: 10765"
```

What is the average daily activity pattern?

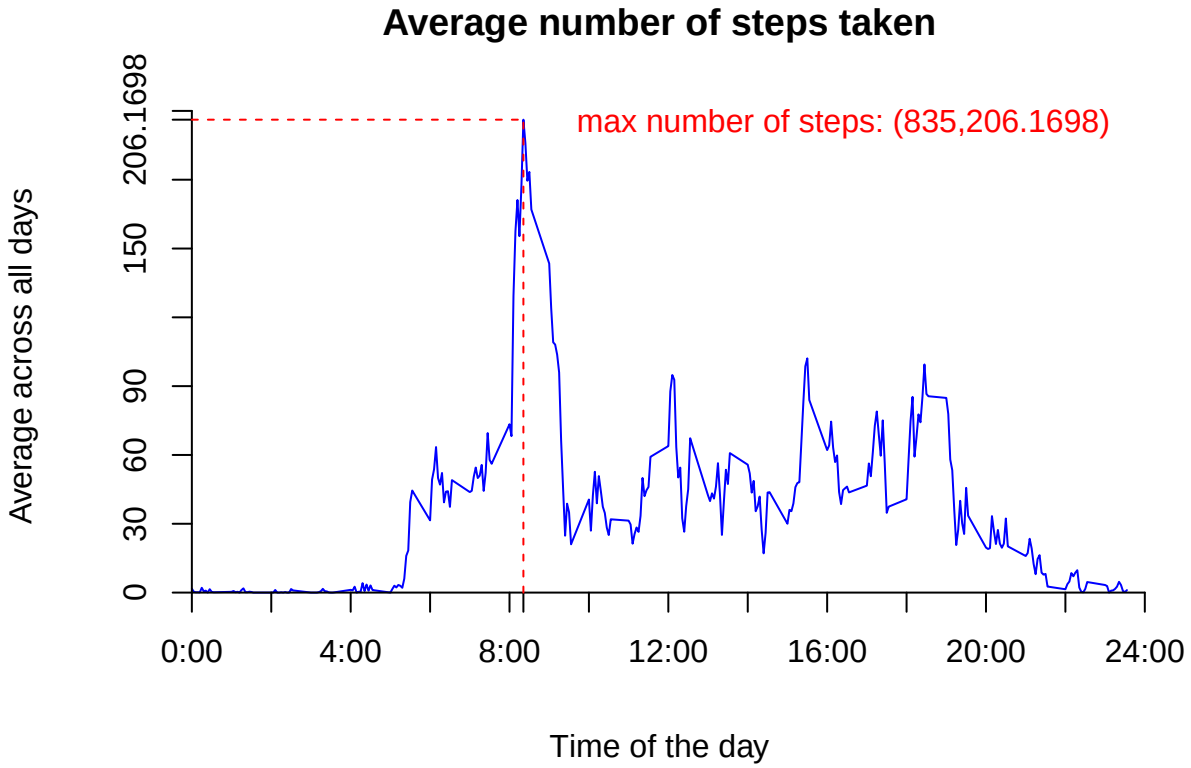
Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

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

```
stepbytime<- aggregate(steps ~ interval, data = activity, FUN = mean)
plot(stepbytime, type = "l", axes = F, xlab = "Time of the day",
     ylab = "Average across all days", main = "Average number of steps taken",
     col = "blue")
maxV <- which.max(stepbytime$steps)
stepbytime[maxV,]
```

```
##     interval    steps
## 104      835 206.1698
```

```
axis(1,at=c(seq(0,2400,200),835), label = paste(c(seq(0,24,2),8),
 c(rep(":00",13),":40"),sep=""), pos = 0)
axis(2, at=c(seq(0,210,30),206.1698), label = c(seq(0,210,30),206.1698), pos = 0)
segments(835, 0, 835, 206.1698, col = "red", lty = "dashed")
text(835,200, "max number of steps: (835,206.1698)", col = "red", adj = c(-.1, -.1))
segments(0, 206.1698, 835, 206.1698, col = "red", lty = "dashed")
```



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.

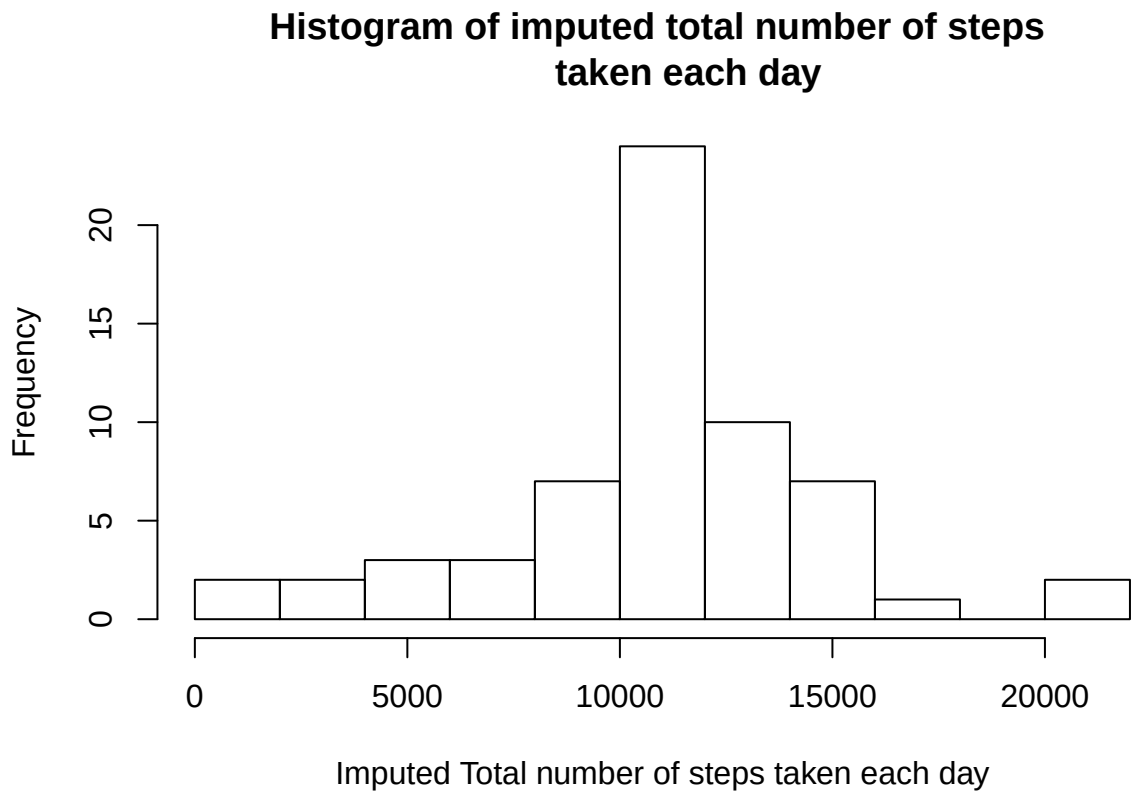
Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

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.

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

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?

```
newdata<- activity
newdata[is.na(activity$steps), ]$steps <- mean(activity$steps,na.rm=TRUE)
# head(newdata)
# head(activity)
stepbydate2 <- setNames(aggregate(steps~as.Date(date), newdata, sum, na.rm = TRUE),
                        c("date","steps"))
hist(stepbydate2$steps, breaks=15,main="Histogram of imputed total number of steps
     taken each day",     xlab="Imputed Total number of steps taken each day")
```



```
mean(stepbydate2$steps)
```

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

```
summary(stepbydate2$steps)
```

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

```
paste("mean:", mean(stepbydate2$steps))
```

```
## [1] "mean: 10766.1886792453"
```

```
paste("median:", median(stepbydate2$steps))
```

```
## [1] "median: 10766.1886792453"
```

```
paste("means difference:", mean(stepbydate2$steps)-mean(stepbydate$steps))
```

```
## [1] "means difference: 0"
```

```
paste("medians difference:", median(stepbydate2$steps)-median(stepbydate$steps))
```

```
## [1] "medians difference: 1.1886792452824"
```

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.

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.

```
str(newdata)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  37.4 37.4 37.4 37.4 37.4 ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
newdata$date <- as.Date(newdata$date, "%Y-%m-%d")
newdata$day <- weekdays(newdata$date)
newdata$ww <- c("weekday")
for (i in 1:nrow(newdata)){
  if (newdata$day[i] == "Saturday" || newdata$day[i] == "Sunday"){
    newdata$ww[i] <- "weekend"
  }
}
newdata$ww<- as.factor(newdata$ww)
str(newdata$ww)
```

```
##  Factor w/ 2 levels "weekday","weekend": 1 1 1 1 1 1 1 1 1 1 ...
```

```
calcu<- aggregate(steps ~ interval+ww, newdata, mean)
qplot(interval, steps, data=calcu, geom=c("line"), xlab="5-min intervals",
      ylab="steps mean", main="") + facet_wrap(~ ww, ncol=1)
```