PA1_template.Rmd

R Markdown

Load library "ggplot2"

```
library(ggplot2)
```

Loading and preprocessing the data Show any code that is needed to

```
if(!(file.exists("data_2Factivity.zip") && file.exists("Source_Classification_Code.rds"))) {
  archiveFile <- "data_2Factivity.zip"
  if(!file.exists(archiveFile)) {
    archiveURL <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
    download.file(url=archiveURL,destfile=archiveFile,method="curl")
  }
  unzip(archiveFile)
}
```

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

```
rawStepData<-read.csv(paste(getwd(),'/activity.csv',sep=''),na.strings = "NA", header = TRUE, sep = ","
```

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

```
rawStepData$date <- as.Date(rawStepData$date, format = "%Y-%m-%d")
```

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.

```
StepData<-rawStepData[!is.na(as.character(rawStepData$steps)),]
```

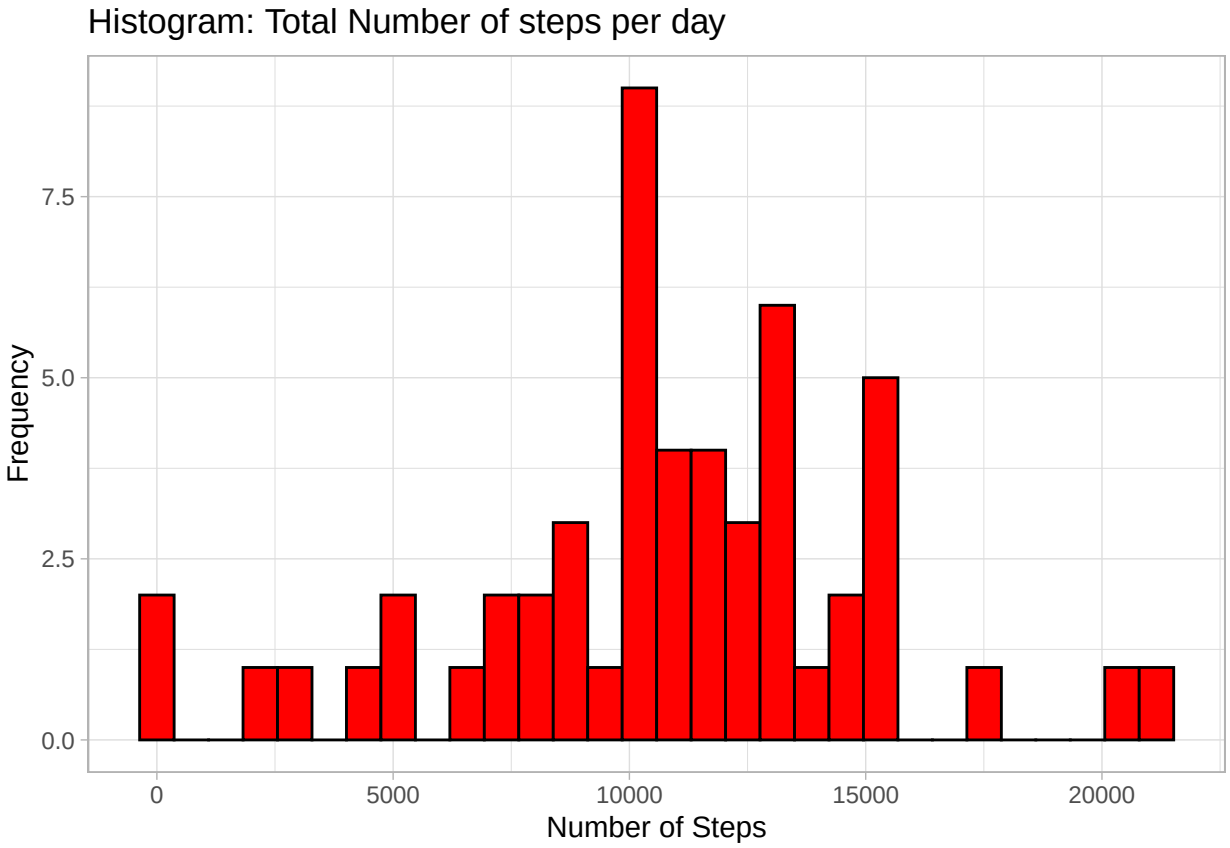
1. Calculate the total number of steps taken per day

```
summSteps<-aggregate(StepData$steps,by=list(StepData$date),FUN=sum,na.rm=TRUE)
colnames(summSteps) <- c("date", "steps")
```

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

```
gg1<-ggplot(summSteps, aes(x=steps))
gg1<-gg1+geom_histogram(color="black",fill="red")
gg1<-gg1+ylab("Frequency")
gg1<-gg1+xlab( "Number of Steps")
gg1<-gg1+ggtitle("Histogram: Total Number of steps per day")
gg1<-gg1+theme_light()
gg1
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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

```
mean(summStepts$steps)
```

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

```
median(summStepts$steps)
```

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

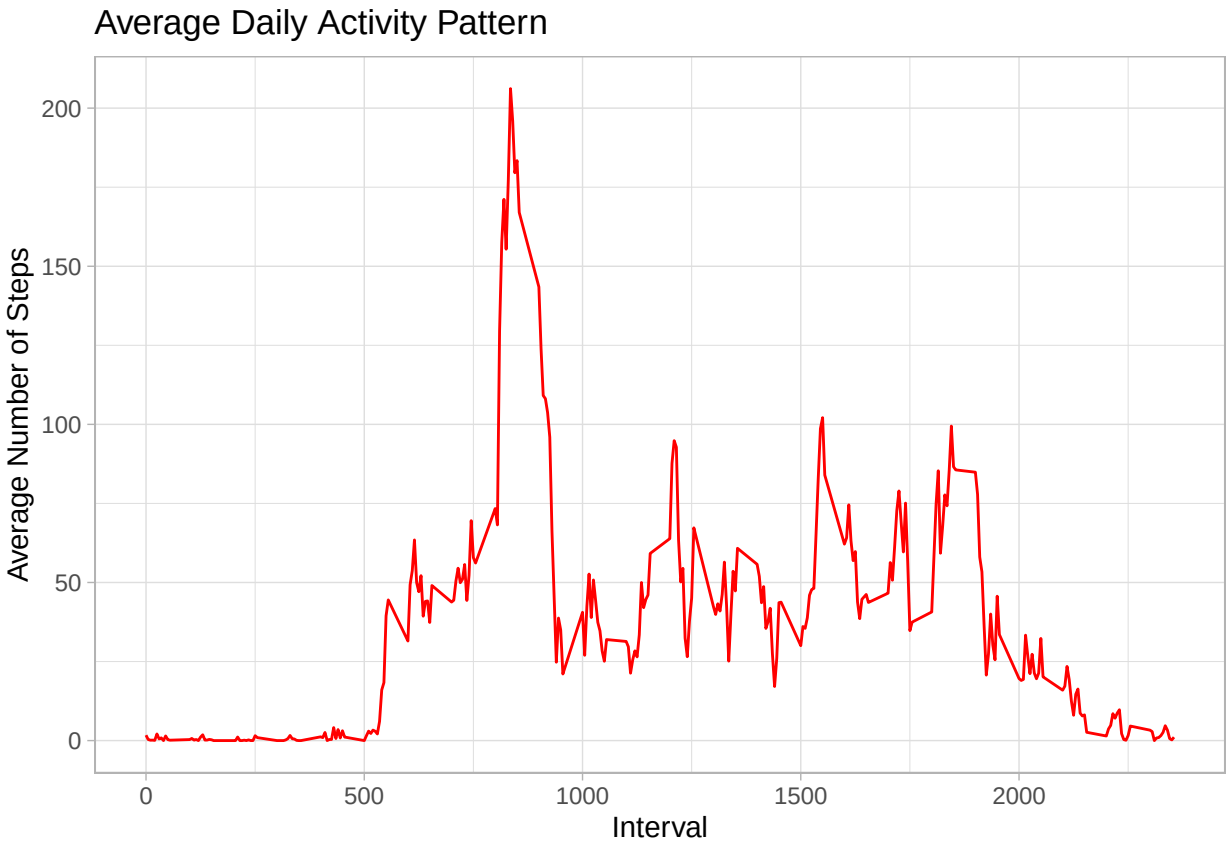
What is the average daily activity pattern?

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)

```
StepsPerInter<-aggregate(StepData$steps,by=list(StepData$interval),FUN=mean,na.rm=TRUE)
colnames(StepsPerInter) <- c("interval", "steps")

gg2<-ggplot(StepsPerInter, aes(x=interval, y=steps))
gg2<-gg2+geom_line(color="red")
```

```
gg2<-gg2+ylab("Average Number of Steps")
gg2<-gg2+xlab("Interval")
gg2<-gg2+ggtitle("Average Daily Activity Pattern")
gg2<-gg2+theme_light()
gg2
```



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

```
StepsPerInter[which.max(StepsPerInter$steps),]$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.

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

```
Nanumb<-sum(is.na(as.character(rawStepData$steps)))
Nanumb
```

```
## [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.

```
NA_index<- which(is.na(as.character(rawStepData$steps)))
stepDataComplete <- rawStepData

stepDataComplete[NA_index, ]$steps<-unlist(lapply(NA_index, FUN=function(NA_index){
  StepsPerInter[rawStepData[NA_index,]$interval==StepsPerInter$interval,]$steps
}))
```

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

```
summary(stepDataComplete)
```

```
##      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
```

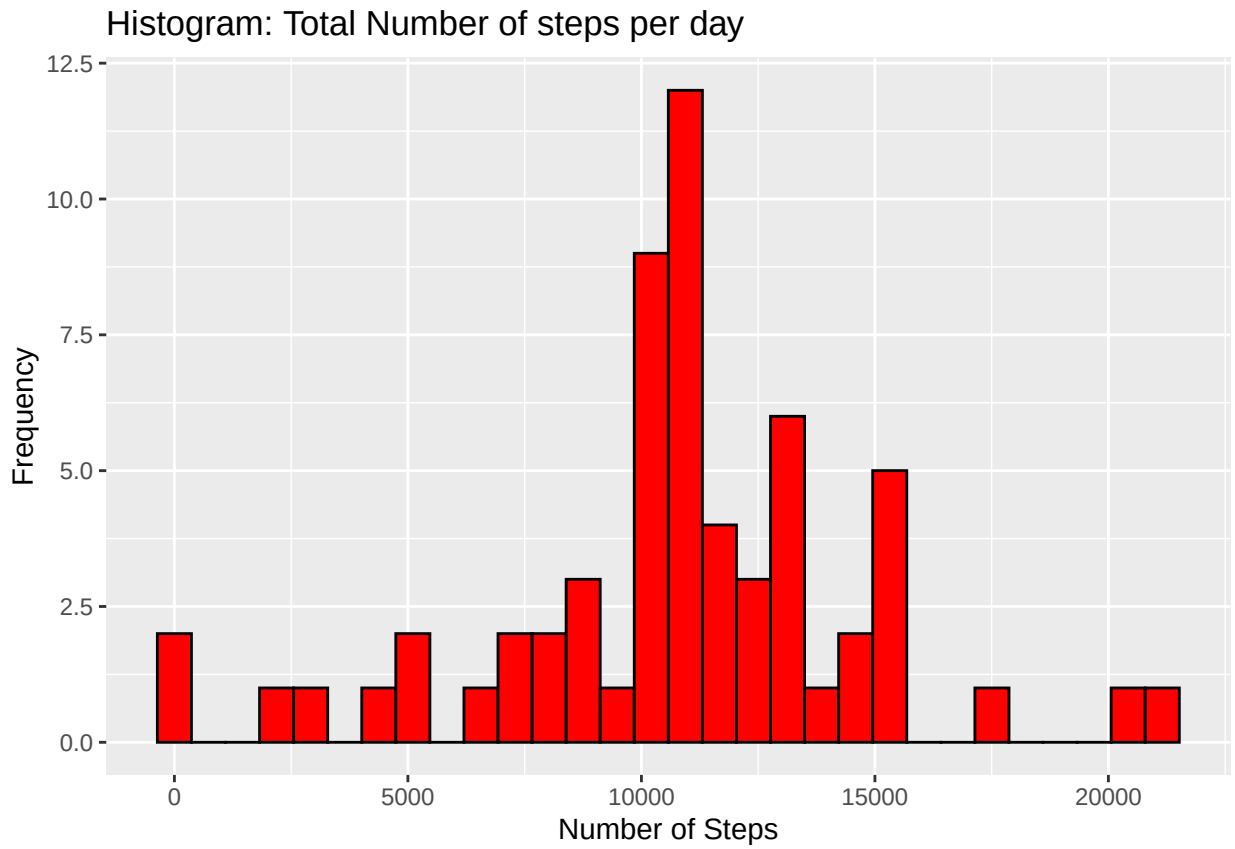
```
str(stepDataComplete)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

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?

```
summSteptsComplete<-aggregate(steps ~ date,data=stepDataComplete,FUN=sum)
colnames(summSteptsComplete) <- c("date", "steps")
gg3<-ggplot(summSteptsComplete,aes(x=steps))
gg3<-gg3+ geom_histogram(color="black",fill="red")
gg3<-gg3+ylab("Frequency")
gg3<-gg3+xlab( "Number of Steps")
gg3<-gg3+ggtitle("Histogram: Total Number of steps per day")
gg3
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
mean(summSteptsComplete$steps)
```

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

```
median(summSteptsComplete$steps)
```

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

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. #### textObservation: if you se this "Head" in Spanish, you don't may executed the code, because was compilate in an computer with 'English' like lenguaje per default.

```
unique(weekdays(stepDataComplete$date))
```

```
## [1] "Monday"    "Tuesday"   "Wednesday" "Thursday"  "Friday"    "Saturday"
## [7] "Sunday"
```

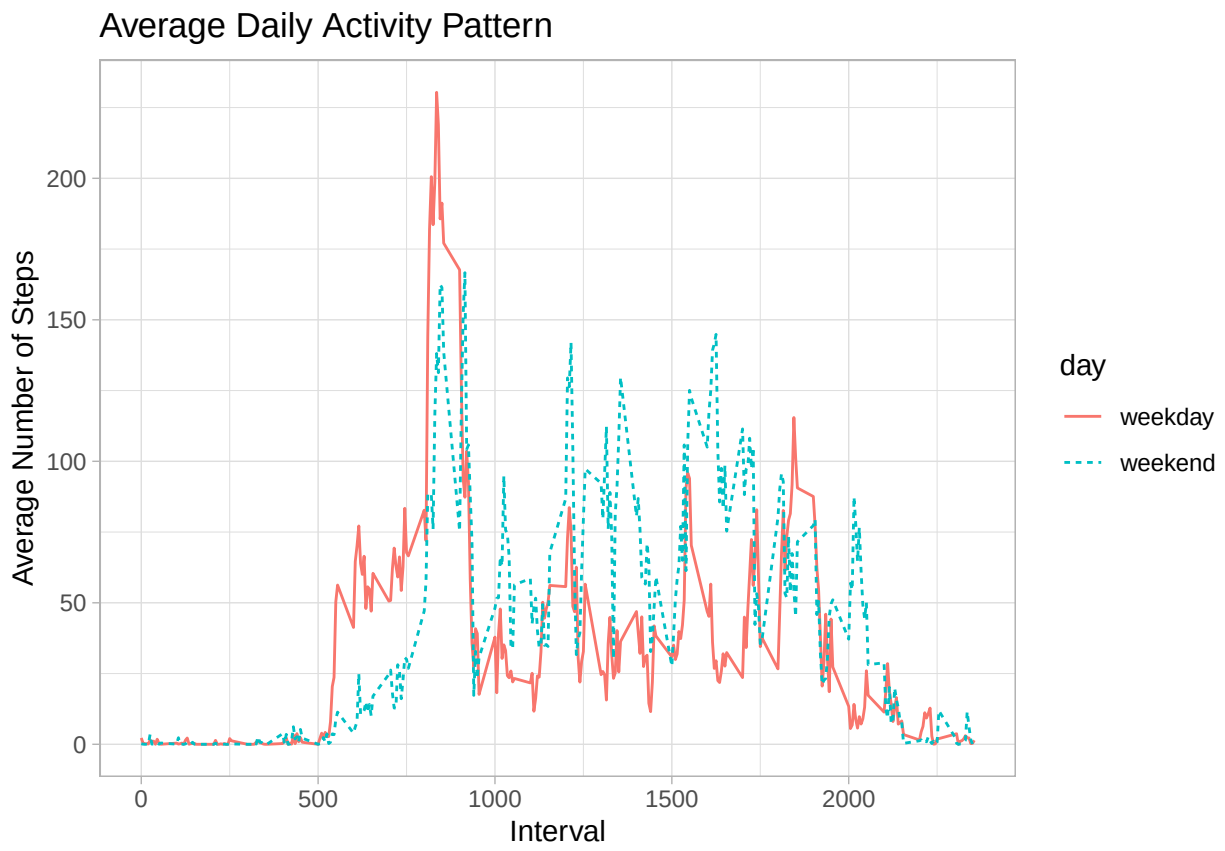
```
weekStepData<-stepDataComplete
weekStepData<-cbind(weekStepData,ifelse((weekdays(weekStepData$date) %in% c("Saturday","Sunday")),TRUE,
colnames(weekStepData)<-c( "steps","date", "interval","is_weekend")
```

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).

```
weekStepData_WEEKEND<-weekStepData[weekStepData$is_weekend,]
weekStepData_WEEKDAY<-weekStepData[!weekStepData$is_weekend,]
StepsPerInter_weekday<-aggregate(weekStepData_WEEKDAY$steps,by=list(weekStepData_WEEKDAY$interval),FUN=
StepsPerInter_weekend<-aggregate(weekStepData_WEEKEND$steps,by=list(weekStepData_WEEKEND$interval),FUN=
colnames(StepsPerInter_weekday) <- c("interval", "steps")
colnames(StepsPerInter_weekend) <- c("interval", "steps")
StepsPerInter_weekday$day<-"weekday"
StepsPerInter_weekend$day<-"weekend"
week_data <- rbind(StepsPerInter_weekday, StepsPerInter_weekend)
head(week_data)
```

```
##   interval      steps     day
## 1        0 2.25115304 weekday
## 2        5 0.44528302 weekday
## 3       10 0.17316562 weekday
## 4       15 0.19790356 weekday
## 5       20 0.09895178 weekday
## 6       25 1.59035639 weekday
```

```
gg4<-ggplot(week_data, aes(x=interval,y=steps))
gg4<-gg4+ geom_line(aes(color = day, linetype = day))
gg4<-gg4+ ylab("Average Number of Steps")
gg4<-gg4+ xlab("Interval")
gg4<-gg4+ ggtitle("Average Daily Activity Pattern")
gg4<-gg4+theme_light()
gg4
```




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

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