Course Project 1 of Reproducible Research

Edgar Alirio Rodridriguez

January 8 at 2018

Course Project 1 of Reproducible Research

Introduction

According to requests for the project assignment of "Reproducible Research" course in this file is condensed the information about: requests, code chunks that resolve these requests and the outcomes of code.

Loading and preprocessing the data

```
library(dplyr)
```

```
##
## 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
```

```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
#Download Zip file
zipUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
zipFile <-"repdata_activity.zip"


if (!file.exists(zipFile)) {
  download.file(zipUrl, zipFile, mode = "wb")
}
#unzip file
unzip(zipFile)

#1. Code for reading in the dataset and/or processing the data
dfActivityOrig <-read.table("activity.csv", sep = "," , stringsAsFactors = FALSE)
#Due to in the first row are the names of columns, it necessary to assign
#the appropriate names to columns and discharge this row
```

```
names(dfActivityOrig) <-dfActivityOrig[1,]
dfActivityOrig<-dfActivityOrig[2:17568, ]

#2. Cleaning data
dfActivity <-dfActivityOrig[complete.cases(dfActivityOrig),]
#Transforming "steps"" column into appropriate format
dfActivity$steps <-as.numeric(dfActivity$steps)
```

The data frame "dfActivity" has the cleaned data and "steps" column with the appropriate format. The data frame "dfActivityOrig" has raw data.

What is mean total number of steps taken per day?

1. Calculating Total Number steps, Mean and median number of steps taken each day

```
dfTotalSteps <-dfActivity %>% group_by(date) %>%
                          summarize(TotStepsDay=sum(steps),
                                    MedianStepsDay= median(steps),
                                    MeanStepsDay=mean(steps))
```

2. Histogram of the total number of steps taken each day

```
png("figure/Histogram_Total_Steps_Day_With_NO_NA.png")
hist(dfTotalSteps$TotStepsDay, breaks=10, main="Histogram of the Total Number of Steps Taken Each Day",
      xlab="Total Number of Steps by Day", ylab = "Frequency")
dev.off()
```

```
## pdf
##   2
```

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

```
head(dfTotalSteps)
```

```
## # A tibble: 6 x 4
##         date TotStepsDay MedianStepsDay MeanStepsDay
##        <chr>       <dbl>          <dbl>        <dbl>
## 1 2012-10-02         126              0      0.43750
## 2 2012-10-03       11352              0     39.41667
## 3 2012-10-04       12116              0     42.06944
## 4 2012-10-05       13294              0     46.15972
## 5 2012-10-06       15420              0     53.54167
## 6 2012-10-07       11015              0     38.24653
```

What is the average daily activity pattern?

1. Make a time series plot of the 5-minute interval and the average number of steps taken, averaged across all days

According to initial data is a result of monitoring the walking activity of one person during 53 days and data was collected at intervals of five minutes. Consequently, this person would have a daily routine that could be different some days from others or even among some periods of the day. For example, at nights, probably there was no data. That´s why common sense drives to calculate mean among same intervals of similar days.

```
dfActivity$interval<- as.numeric(dfActivity$interval)
i<-0
#Assigning one sequence ID (conInterval) to each interval for each day
dfActivity$conInterval <- tapply(dfActivity$date,dfActivity$interval,i=i+1)
#Identifying each day with a sequence ID of week, where 1 is for Mondays and 7 for Sundays
dfActivity$wday <-wday(dfActivity$date)
#Calculating mean for each interval and for each day
dfIntervalMean <-dfActivity %>% group_by(wday,conInterval) %>%
                  summarize(MeanStepsInterval=mean(steps, na.rm = TRUE))

totalrows<-nrow(dfActivity)
#Creating and initializing with zeros an interval means vector
MeanStepsInterval<-rep(0,totalrows)
dfActivity<-cbind(dfActivity,MeanStepsInterval)
#Assignig mean for each interval of each day
i<-1
while (i <= totalrows) {
  dfActivity[i,"MeanStepsInterval"]<- round(dfIntervalMean[((dfIntervalMean$wday==dfActivity[i,"wday"])
                                                    (dfIntervalMean$conInterval==dfActivity[i,"conInter
                                                     "MeanStepsInterval"], digits = 0)

  i<-i+1
}

totalrows<-nrow(dfActivity)
#Creating and initializing (with any value) a datetime interval vector in POSIXct format.
DateTimeInterval<-rep(as.POSIXct(as.Date("1972-02-15 00:00:00",
                                         tz="UTC",format="%Y-%m-%d %H:%M:%S"),totalrows))
dfActivity<-cbind(dfActivity,DateTimeInterval)
i<-1
while (i <= totalrows) {
  #Calculating the appropiate sequence of data time intervals in order to populate the x-axis.
  #Each interval takes 300 seconds, this value is multiplied for its respective ID interval and added t
  #start time of day (00:00:00)
  dfActivity[i,"DateTimeInterval"]<- as.POSIXct(as.Date(paste(dfActivity[i,"date"],"00:00:00",sep = " "
  +(300*dfActivity[i,"conInterval"])
  i<-i+1
}
png("figure/Plot_Mean_Steps_Intervals.png")
plot(x=as.Date(dfActivity$DateTimeInterval),y=dfActivity$MeanStepsInterval, type = "l",
     main = "Time Series of the 5-minute interval and Average Number of Steps",
     xlab="Intervals of 5-minutes during 53 days", ylab="Mean of Steps")
dev.off()
```

```
## pdf
##   2
```

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

```
maxSteps<- dfActivity[which.max(dfActivity$steps),]
print(maxSteps)
```

```
##       steps       date interval conInterval wday MeanStepsInterval
## 16493   806 2012-11-27      615          76    3                95
##          DateTimeInterval
## 16493 2012-11-26 19:00:00
```

Imputing missing values

Just as mentioned before, the strategy for filling missing values of steps at some intervals is to calculate the mean for each interval among a similar day of the week. ###1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
CountNA <- nrow(dfActivityOrig[is.na(dfActivityOrig$steps),])
print(paste("The Total Number of missing values in the dataset is:",CountNA))
```

```
## [1] "The Total Number of missing values in the dataset is: 2303"
```

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

```
dfActPattern <-dfActivityOrig
dfActPattern$wday <-wday(dfActPattern$date)
dfActPattern$interval <-as.numeric(dfActPattern$interval)
dfActPattern$steps <-as.numeric(dfActPattern$steps)
i<-0
dfActPattern$conInterval <- tapply(dfActPattern$date,dfActPattern$interval,i=i+1)
dfIntervalMean2 <-dfActPattern %>% group_by(wday,conInterval) %>% summarize(MeanStepsInterval=mean(steps
```

2. Strategy for filling in all of the missing values in the dataset

```
totalrows<-nrow(dfActPattern)
i<-1
while (i <= totalrows) {
  if (is.na(dfActPattern[i,"steps"])) {
    dfActPattern[i,"steps"]<- round(dfIntervalMean2[((dfIntervalMean2$wday==dfActPattern[i,"wday"]) &
                                          (dfIntervalMean2$conInterval==dfActPattern[i,"conInterval
                                          "MeanStepsInterval"], digits = 0)
  }
  i<-i+1
}
```

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

```
dfTotalSteps2 <-dfActPattern %>% group_by(date) %>%
                summarize(TotStepsDay=sum(steps), MedianStepsDay= median(steps),
```

```
                                                    MeanStepsDay=mean(steps))

png("figure/Histogram_Total_Steps_Day_With_Imputed_Values.png")
hist(dfTotalSteps2$TotStepsDay, breaks=10, main="Histogram of the Total Number of Steps Taken Each Day
     xlab="Total Number of Steps by Day", ylab = "Frequency")
dev.off()
```

```
## pdf
##   2
```

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

```
head(dfTotalSteps2)
```

```
## # A tibble: 6 x 4
##         date TotStepsDay MedianStepsDay MeanStepsDay
##        <chr>       <dbl>          <dbl>        <dbl>
## 1 2012-10-01        9978              8     34.64583
## 2 2012-10-02         126              0      0.43750
## 3 2012-10-03       11352              0     39.41667
## 4 2012-10-04       12116              0     42.06944
## 5 2012-10-05       13294              0     46.15972
## 6 2012-10-06       15420              0     53.54167
```

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?

```
#Ckeking
summary(dfActivity$steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   37.38   12.00  806.00
```

```
summary(dfActPattern$steps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   37.58   19.00  806.00
```

After reviewing the results of data with no "NA" (dfActivity) and data changing NA by mean of same intervals (dfActPattern), there are a slight difference of two decimals on mean and redistribution of the third quartile.

Are there differences in activity patterns between weekdays and weekends?

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.

```
totalrows<-nrow(dfActPattern)
TypeDay<-rep("NA",totalrows)
dfActPattern<-cbind(dfActPattern,TypeDay)
dfActPattern$TypeDay<-as.character(dfActPattern$TypeDay)
i<-1
while (i<=totalrows) {
  if(dfActPattern[i,"wday"]>5) {
```

```
    dfActPattern[i,"TypeDay"] <- c("Weekend")
    }
  else{
    dfActPattern[i,"TypeDay"] <- c("Weekday")
   }
  i<- i+1
}
dfActPattern$TypeDay<-as.factor(dfActPattern$TypeDay)
```

2.Make a panel plot containing a time series plot

```
totalrows<-nrow(dfActPattern)
#Creating and initializing (with any value) a datetime interval vector in POSIXct format.
DateTimeInterval<-rep(as.POSIXct(as.Date("1972-02-15 00:00:00",tz="UTC",format="%Y-%m-%d %H:%M:%S"),tota
dfActPattern<-cbind(dfActPattern,DateTimeInterval)
i<-1
while (i <= totalrows) {
  #Calculating the appropiate sequence of data time intervals in order to populate the x-axis.
  #Each interval takes 300 seconds, this value is multiplied for its respective ID interval and added t
  #start time of day (00:00:00)
  dfActPattern[i,"DateTimeInterval"]<- as.POSIXct(as.Date(paste(dfActPattern[i,"date"],"00:00:00",sep =
  i<-i+1
}

#Calculating mean steps by interval grouping by Type of Day
dfMeanStepsTypeDay<-dfActPattern %>% group_by(TypeDay,DateTimeInterval) %>% summarize(MeanStepsTypeDay=m
png("figure/Plot_difference_weekdays_weekend.png")
plotStepsTypeDay<- qplot(DateTimeInterval,MeanStepsTypeDay, data=dfMeanStepsTypeDay, facets=. ~ TypeDay
                         main="Differences in activity patterns between weekdays and weekends",
                         xlab = "Days", ylab = "Mean Steps by Interval", geom=c("line"))
print(plotStepsTypeDay)
dev.off()
```

```
## pdf
##   2
```