Reproducible\_Research\_Course\_Project\_1

Luis OLiveira

15/08/2020

### Loading Data

The data file “activity.csv” already unziped should be in the user current R working directory.

MainDataAtivity <- read.csv("activity.csv", colClasses=c("numeric", "Date", "numeric"))

## —-What is mean total number of steps taken per day?—

## Exclude NA steps rows

names(MainDataAtivity)

## [1] "steps" "date" "interval"

str(MainDataAtivity)

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

head(MainDataAtivity[which(!is.na(MainDataAtivity$steps)), ])

## steps date interval  
## 289 0 2012-10-02 0  
## 290 0 2012-10-02 5  
## 291 0 2012-10-02 10  
## 292 0 2012-10-02 15  
## 293 0 2012-10-02 20  
## 294 0 2012-10-02 25

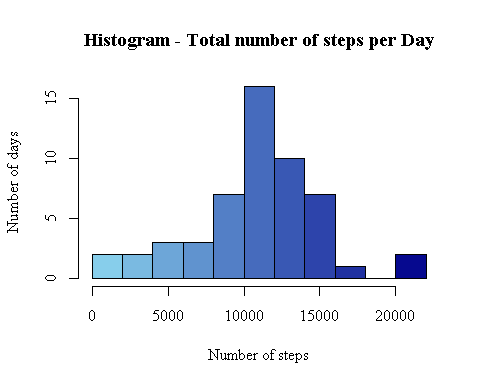
## Compute the total steps per day

totalstepsday <- aggregate(steps ~ date, MainDataAtivity, sum)   
head(totalstepsday)

## date steps  
## 1 2012-10-02 126  
## 2 2012-10-03 11352  
## 3 2012-10-04 12116  
## 4 2012-10-05 13294  
## 5 2012-10-06 15420  
## 6 2012-10-07 11015

## How to build the Histogram of the total number of steps taken each day

paletteBlue <- colorRampPalette(c("skyblue", "darkblue", "skyblue"))   
hist(totalstepsday$steps, breaks=10, xlab="Number of steps", ylab="Number of days",  
main="Histogram - Total number of steps per Day",   
col=paletteBlue(22), family="serif")



## Computing the mean and median number of steps taken each day

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

stepsstatistics <- summarise(totalstepsday, meantotalsteps=mean(totalstepsday$steps),   
mediantotalsteps=median(totalstepsday$steps))   
print(stepsstatistics)

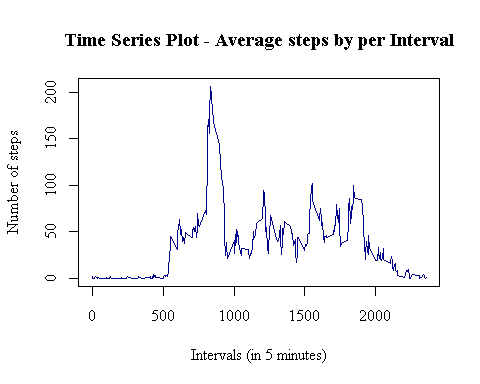
## meantotalsteps mediantotalsteps  
## 1 10766.19 10765

## —–What is the average daily activity pattern?—-

## Time series plot of the average number of steps taken  
  
meanstepsinterval <- aggregate(steps ~ interval, MainDataAtivity, mean)   
head(meanstepsinterval)

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

plot(x=meanstepsinterval$interval, y=meanstepsinterval$steps, type="l",   
main="Time Series Plot - Average steps by per Interval",   
ylab="Number of steps", xlab="Intervals (in 5 minutes)",   
col="darkblue", lwd=1.5, family="serif")



## The 5-minute interval that, on average, contains the max number of steps

meanstepsinterval[grep(max(meanstepsinterval$steps), meanstepsinterval$steps), ]

## interval steps  
## 104 835 206.1698

## —–Imputing missing values —–

## Code for imputing missing data

## Replace NA values in the steps variable by the mean of the interval

adjusteddata <- MainDataAtivity   
for(x in 1:17568) {   
 if(is.na(adjusteddata[x, 1])==TRUE) { adjusteddata[x, 1] <- meanstepsinterval[meanstepsinterval$interval %in% adjusteddata[x, 3], 2] } }   
head(adjusteddata)

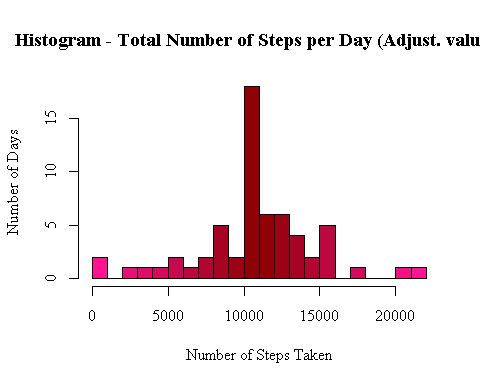
## steps date interval  
## 1 1.7169811 2012-10-01 0  
## 2 0.3396226 2012-10-01 5  
## 3 0.1320755 2012-10-01 10  
## 4 0.1509434 2012-10-01 15  
## 5 0.0754717 2012-10-01 20  
## 6 2.0943396 2012-10-01 25

## How to build the Histogram of the total number of steps taken each day after missing values are imputed

adjustedtotalstepsday <- aggregate(steps ~ date, adjusteddata, sum)   
head(adjustedtotalstepsday)

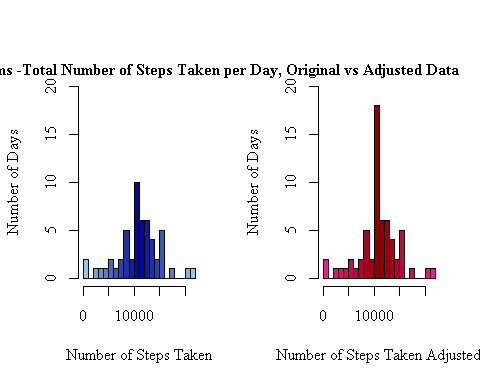
## date steps  
## 1 2012-10-01 10766.19  
## 2 2012-10-02 126.00  
## 3 2012-10-03 11352.00  
## 4 2012-10-04 12116.00  
## 5 2012-10-05 13294.00  
## 6 2012-10-06 15420.00

paletteRed <- colorRampPalette(c("deeppink", "darkred", "deeppink"))   
hist(adjustedtotalstepsday$steps, breaks=20, xlab="Number of Steps Taken", ylab="Number of Days",  
main="Histogram - Total Number of Steps per Day (Adjust. values)",   
col=paletteRed(22), family="serif")



## How to compare Histograms to see the impact of imputing vakues in missing data

par(mfrow = c(1, 2))   
hist(totalstepsday$steps, breaks=20, xlab="Number of Steps Taken", ylab="Number of Days",  
col=paletteBlue(22), family="serif", ylim=c(0, 20), main=NULL)   
hist(adjustedtotalstepsday$steps, breaks=20, xlab="Number of Steps Taken Adjusted", ylab="Number of Days",  
col=paletteRed(22), family="serif", ylim=c(0, 20), main=NULL)   
mtext("Histograms -Total Number of Steps Taken per Day, Original vs Adjusted Data",   
adj=0.95, family="serif", font=2)



##Analysis if there are differences in activity patterns between weekdays and weekends?

daysdata <- adjusteddata   
daysdata$days <- weekdays(daysdata$date)   
daysdata$weekday <- as.character(rep(0, times=17568))   
for(x in 1:17568) {   
 if(daysdata[x, 4] %in% c("Saturday", "Sunday")) { daysdata[x, 5] <- "Weekend" } else { daysdata[x, 5] <- "Weekday" } }   
daysdata$weekday <- factor(daysdata$weekday)   
head(daysdata)

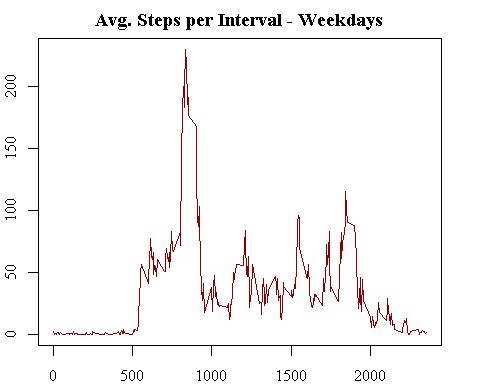
## steps date interval days weekday  
## 1 1.7169811 2012-10-01 0 Monday Weekday  
## 2 0.3396226 2012-10-01 5 Monday Weekday  
## 3 0.1320755 2012-10-01 10 Monday Weekday  
## 4 0.1509434 2012-10-01 15 Monday Weekday  
## 5 0.0754717 2012-10-01 20 Monday Weekday  
## 6 2.0943396 2012-10-01 25 Monday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

## Comparing the weekday and weekend data

weekdaydata <- daysdata[daysdata$weekday=="Weekday", ]   
weekenddata <- daysdata[daysdata$weekday=="Weekend", ]   
  
weekdayMean <- aggregate(steps ~ interval, weekdaydata, mean)   
weekendMean <- aggregate(steps ~ interval, weekenddata, mean)

## Making the plot chart to compare the two series

par(mar = rep(2, 4))  
plot(weekdayMean$interval, weekdayMean$steps, type="l",   
main="Avg. Steps per Interval - Weekdays",   
xlab="Intervals (5 mins)", ylab="Number Steps", family="serif",   
col="darkred", lwd=1.5, ylim=c(0, 230))



plot(weekendMean$interval, weekendMean$steps, type="l",   
main="Avg. Steps per Interval - Weekends",   
xlab="Intervals (5 mins)", ylab="Number Steps", family="serif",   
col="darkblue", lwd=1.5, ylim=c(0, 230))

