Peer-G-A-Course Proj-1

Phillip

7/22/2020

## Loading and processing the dataset “activity”.

Act\_Mon <- read.csv("activity.csv")  
head(Act\_Mon,3, echo = TRUE)

## steps date interval  
## 1 NA 2012-10-01 0  
## 2 NA 2012-10-01 5  
## 3 NA 2012-10-01 10

Assignment for project-1:

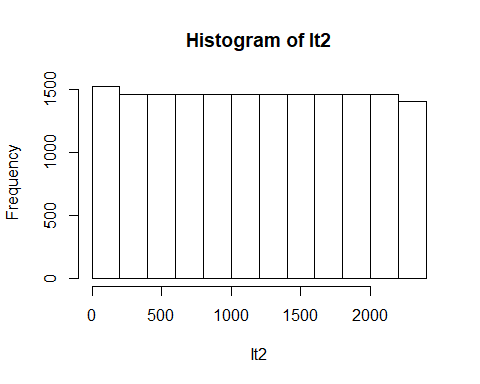
## The data shows the mean total number of steps taken per day:

It2 <- Act\_Mon$interval  
mean(It2)

## [1] 1177.5

## The histogram will show the total number of steps taken per day. and the summary will display the mean and median of the data.

It2 <- Act\_Mon$interval  
  
hist(It2)



## The summary function will display the mean and median of the data.

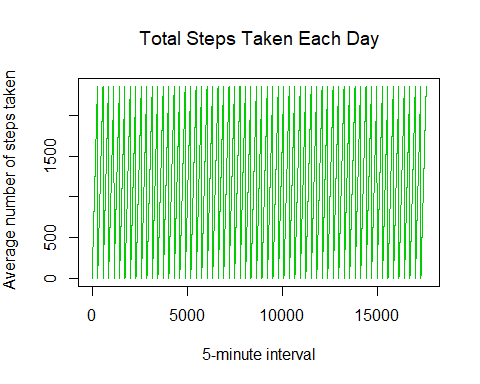
It2 <- Act\_Mon$interval  
  
summary(It2)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 588.8 1177.5 1177.5 1766.2 2355.0

## The plot shows the average daily activity pattern of average number of steps taken and which has the maximum number of steps.

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

It2 <- Act\_Mon$interval  
  
plot(Act\_Mon$interval, type = "l", main = expression("Total Steps Taken Each Day"),  
 xlab = "5-minute interval", ylab = expression("Average number of steps taken"), col = "green3")



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

It2 <- Act\_Mon$interval  
  
Mon\_st <- (Act\_Mon$steps)  
  
sum(Mon\_st, na.rm = TRUE)

## [1] 570608

P2 <- Act\_Mon$steps  
str(P2)

## int [1:17568] NA NA NA NA NA NA NA NA NA NA ...

P3\_st <- na.omit(P2)  
head(P3\_st)

## [1] 0 0 0 0 0 0

View(P3\_st)  
summary(P3\_st)

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

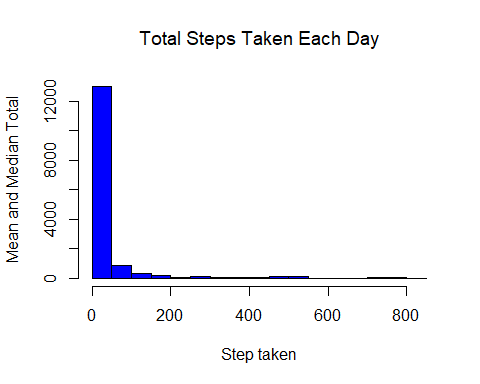
hist(P3\_st, type = "l", main = expression("Total Steps Taken Each Day"),  
 xlab = "Step taken", ylab = expression("Mean and Median Total"), col = "blue")

## Warning in plot.window(xlim, ylim, "", ...): graphical parameter "type" is  
## obsolete

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):  
## graphical parameter "type" is obsolete

## Warning in axis(1, ...): graphical parameter "type" is obsolete

## Warning in axis(2, ...): graphical parameter "type" is obsolete



## 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. 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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

DD2 <- plot(P3\_st, add = TRUE, type = "l", xlab = "Interval", ylab = expression("Number of steps"), col = "green")

## Warning in plot.window(...): "add" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "add" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "add" is not a  
## graphical parameter  
  
## Warning in axis(side = side, at = at, labels = labels, ...): "add" is not a  
## graphical parameter

## Warning in box(...): "add" is not a graphical parameter

## Warning in title(...): "add" is not a graphical parameter

