PA1\_template.Rmd

2023-06-25

# 1.0 Loading and pre-processing the data

#1.1 Ensure csv file is on the working directory

library(ggplot2)  
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

activity <- read.csv("activity.csv")

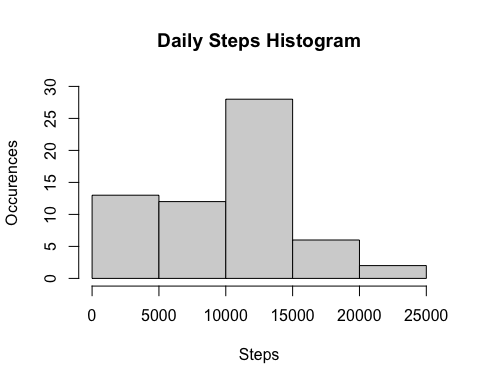
#2.0 What is mean total number of steps taken per day?

#2.1 Calculating the total steps per day

StepsPerDay <- activity %>%  
 group\_by(date) %>%  
 summarize(totalsteps = sum(steps, na.rm = TRUE)) #removes NA values

#2.2 Histogram of the total steps per day

hist(StepsPerDay$totalsteps, main = "Daily Steps Histogram",  
 xlab = "Steps", ylab = "Occurences", ylim=c(0,30))



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

mean <- round(mean(StepsPerDay$totalsteps))  
 median <- round(median(StepsPerDay$totalsteps))  
 print(paste("Mean: ", mean))

## [1] "Mean: 9354"

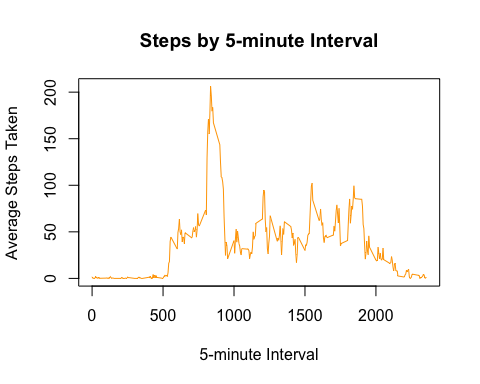
print(paste("Median: ",median))

## [1] "Median: 10395"

#3.0 What is the average daily activity pattern?

#3.1 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)

StepsPerInterval <- activity %>%  
 group\_by(interval) %>%  
 summarize(MeanSteps = mean(steps, na.rm = TRUE))  
   
 plot(StepsPerInterval$MeanSteps ~ StepsPerInterval$interval, main = "Steps by 5-minute Interval",   
 col='orange',type="l", xlab = "5-minute Interval", ylab="Average Steps Taken")



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

print(paste("5-minute interval with the maximum number of steps on average across all days:",  
 StepsPerInterval$interval[which.max(StepsPerInterval$MeanSteps)]))

## [1] "5-minute interval with the maximum number of steps on average across all days: 835"

print(paste("Average steps for the 5-minute interval with the maximum number of steps on average across all days:",  
 round(max(StepsPerInterval$MeanSteps))))

## [1] "Average steps for the 5-minute interval with the maximum number of steps on average across all days: 206"

#4.0 Imputing missing values

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

print(paste("Total number of rows with NAs:", sum(is.na(activity$steps))))

## [1] "Total number of rows with NAs: 2304"

#4.2 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.  
  
#4.3 Create a new dataset that is equal to the original dataset but with the missing data filled in.

#Before Imputing NA  
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

#After Imputing NA  
 ActivityImputingNA <- activity  
 for (i in 1:nrow(activity)){ #loop from 1 to 17568 row  
 if(is.na(activity$steps[i])){ #if step is NA, the mean for that 5-minute interval will be used  
 ActivityImputingNA$steps[i] <- StepsPerInterval$MeanSteps[ActivityImputingNA$interval[i] == StepsPerInterval$interval]  
 }  
 }  
 #The mean is now populated to the intervals with NA  
 head(ActivityImputingNA)

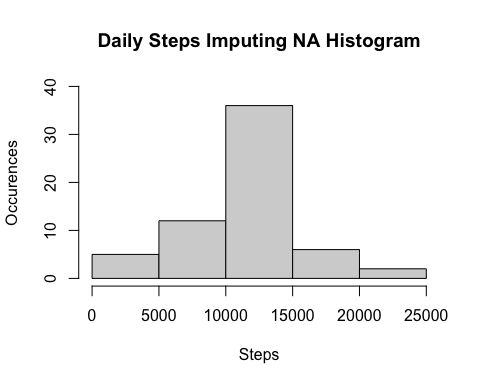
## steps date interval  
## 1 1.7169811 2012-10-01 0  
## 2 0.3396226 2012-10-01 5  
## 3 0.1320755 2012-10-01 10  
## 4 0.1509434 2012-10-01 15  
## 5 0.0754717 2012-10-01 20  
## 6 2.0943396 2012-10-01 25

#4.4 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?

StepsPerDayImputingNA <- ActivityImputingNA %>%  
 group\_by(date) %>%  
 summarize(totalsteps = round(sum(steps)))   
  
head(StepsPerDayImputingNA)

## # A tibble: 6 × 2  
## date totalsteps  
## <chr> <dbl>  
## 1 2012-10-01 10766  
## 2 2012-10-02 126  
## 3 2012-10-03 11352  
## 4 2012-10-04 12116  
## 5 2012-10-05 13294  
## 6 2012-10-06 15420

hist(StepsPerDayImputingNA$totalsteps, main="Daily Steps Imputing NA Histogram",  
 xlab = "Steps", ylab = "Occurences", ylim=c(0,40))



meanImputingNA <- round(mean(StepsPerDayImputingNA$totalsteps))  
 medianImputingNA <- round(median(StepsPerDayImputingNA$totalsteps))  
 print(paste("Mean Imputing NA:", meanImputingNA))

## [1] "Mean Imputing NA: 10766"

print(paste("Median Imputing NA:", medianImputingNA))

## [1] "Median Imputing NA: 10766"

#Compare Before and After Imputing NA  
#The values differ from the estimates from the first part of the assignment.  
#The mean and median increase after imputing missing values.

CompareNA <- data.frame(mean = c(mean, meanImputingNA), median = c(median, medianImputingNA))  
 rownames(CompareNA) <- c("Before Imputing NA", "After Imputing NA")  
 print(CompareNA)

## mean median  
## Before Imputing NA 9354 10395  
## After Imputing NA 10766 10766

#5.0 Are there differences in activity patterns between weekdays and weekends?

#5.1 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.

ActivityDay <- ActivityImputingNA  
 ActivityDay$date <- as.Date(ActivityDay$date)   
 ActivityDay$day <- ifelse(weekdays(ActivityDay$date) %in% c("Saturday","Sunday"),"Weekend","Weekday")  
 ActivityDay$day <- as.factor(ActivityDay$day)

#5.2 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.

ActivityWeekday <- filter(ActivityDay, ActivityDay$day == "Weekday")  
 ActivityWeekend <- filter(ActivityDay, ActivityDay$day == "Weekend")  
   
 #Weekday average number of steps  
 ActivityWeekday <- ActivityWeekday %>%  
 group\_by(interval) %>%  
 summarize(steps = mean(steps))  
 ActivityWeekday$day <- "Weekday"  
   
 #Weekend average number of steps  
 ActivityWeekend <- ActivityWeekend %>%  
 group\_by(interval) %>%  
 summarize(steps = mean(steps))  
 ActivityWeekend$day <- "Weekend"  
   
 #Combine Weekday and Weekend  
 ActivityWeekdayWeekend <- rbind(ActivityWeekday, ActivityWeekend)  
 ActivityWeekdayWeekend$day <- as.factor(ActivityWeekdayWeekend$day)  
   
 #plot  
 ggplot(ActivityWeekdayWeekend,   
 aes(interval,steps)) +  
 geom\_line() +  
 facet\_grid (day~.) +  
 labs(y = "Average Number of Steps") + labs(x = "Interval") +  
 ggtitle("Weekday vs Weekend Average Number of Steps")

