Reproducible Research - Project 1

Clayton Cafferata

6/13/2017

This is for Coursera's Reproducible Research Course Project 1.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.

The data for this assignment can be downloaded from the course web site:

Dataset: https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip

The variables included in this dataset are:

steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)  
date: The date on which the measurement was taken in YYYY-MM-DD format  
interval: Identifier for the 5-minute interval in which measurement was taken

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.

The work below assumes you have downloaded the dataset and unzipped it.

\*\* Part 1,2,3: Reading-in Data, Histogram, Mean and Median

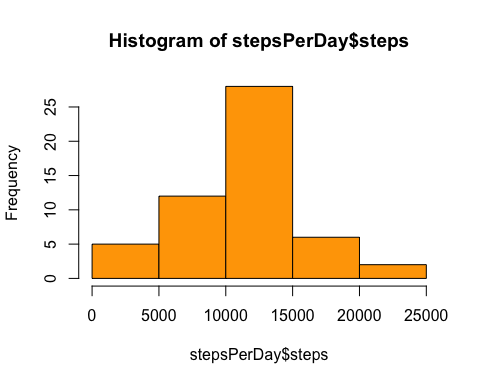
# load plotting program  
library(dplyr)

##   
## 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)  
  
# read the data file  
  
ActivityData <- read.csv("activity.csv")  
  
# total steps per day with NAs  
  
stepsPerDay <- aggregate(steps ~ date, ActivityData, sum, na.rm=TRUE)  
hist(stepsPerDay$steps, col = "orange")



# mean and median steps per day with NAs  
  
meanStepsPerDay <- mean(stepsPerDay$steps)  
medianStepsPerDay <- median(stepsPerDay$steps)  
  
meanStepsPerDay

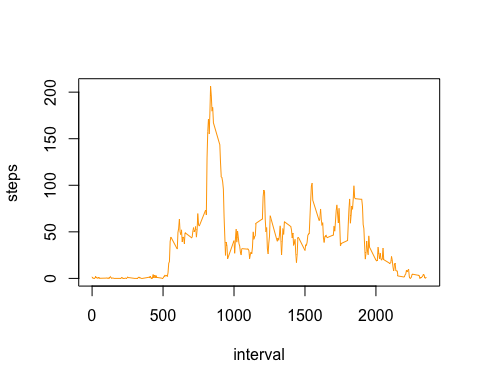
## [1] 10766.19

medianStepsPerDay

## [1] 10765

\*\* Part 4: Time Series of Average Steps Taken

# 5 minutes intervals for daily steps with NAs  
  
stepsPerInterval<-aggregate(steps~interval, data=ActivityData, mean, na.rm=TRUE)  
plot(steps~interval, data=stepsPerInterval, type="l", col = "orange")



\*\* Part 5: The 5 minute interval that contains on average the largest # of Steps

# max steps with NAs  
  
maxSteps <- stepsPerInterval[which.max(stepsPerInterval$steps),]$interval  
maxSteps

## [1] 835

\*\* Part 6: A Methodology to Find Missing Data and Replace it Systematically with Reasonable Estimates

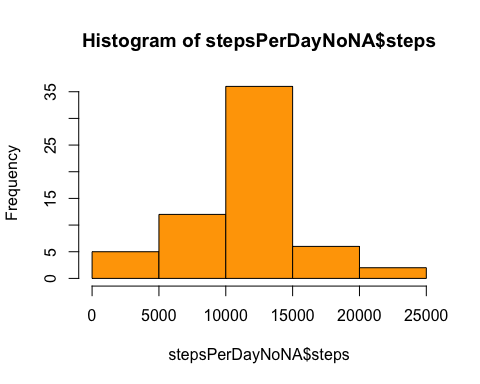
# number of NA's in data file  
  
totalNA <- sum(is.na(ActivityData))  
totalNA

## [1] 2304

# new data set to fill in NAs  
  
ActivityData2 <- ActivityData  
NAs <- nas<- is.na(ActivityData2$steps)  
  
# here we are basing the fill-in numbers on the average number of steps in the same 5 minute interval  
  
avginterval<- tapply(ActivityData2$steps, ActivityData2$interval, mean, na.rm=TRUE, simplify = TRUE)  
ActivityData2$steps[NAs] <- avginterval[as.character(ActivityData2$interval[NAs])]

\*\* Part 7: Histogram of Total Daily Steps with no NAs plus mean and median

# Histogram with no-NAs  
  
stepsPerDayNoNA <- aggregate(steps ~ date, data=ActivityData2, sum)  
hist(stepsPerDayNoNA$steps, col = "orange")



# mean and median steps per day without NAs  
  
meanStepsPerDayNoNA <- mean(stepsPerDayNoNA$steps)  
medianStepsPerDayNoNA <- median(stepsPerDayNoNA$steps)  
  
meanStepsPerDayNoNA

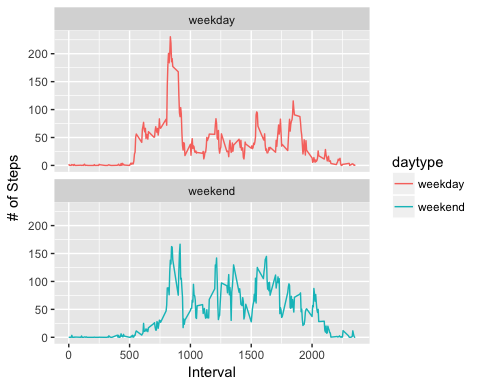
## [1] 10766.19

medianStepsPerDayNoNA

## [1] 10766.19

\*\* Part 8: Panel Plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

# Show differences in Activity patterns weekend vs weekday  
# Create new factor variable: weekday or weekend  
  
ActivityData3 <- ActivityData2  
weekend <- weekdays(as.Date(ActivityData3$date)) %in% c("Saturday","Sunday")  
ActivityData3$daytype <- "weekday"  
  
# replace "weekday" with "weekend" where day equals Saturday or Sunday  
ActivityData3$daytype[weekend == TRUE] <- "weekend"  
ActivityData3$daytype <- as.factor(ActivityData3$daytype)  
  
Interval2 <- ActivityData3%>%  
 group\_by(interval, daytype)%>%  
 summarise(avg\_steps2 = mean(steps))  
  
plot<- ggplot(Interval2, aes(x =interval , y=avg\_steps2, color=daytype)) + geom\_line() + labs(x = "Interval", y = "# of Steps") + facet\_wrap(~daytype, ncol = 1, nrow=2)  
print(plot)



\*\* Part 9: All of the R code needed to reproduce the results (numbers, plots, etc.) in the report see above for the code