RepData\_PeerAssessment1

Eric Bird

July 24, 2016

Loading and preprocessing the data

Show any code that is needed to

Load the data (i.e. 𝚛𝚎𝚊𝚍.𝚌𝚜𝚟()) Process/transform the data (if necessary) into a format suitable for your analysis

setwd("/Users/Prometheus/Documents/Data Science Specialization/Reproducible Research/")  
activitydata <- read.csv("activity 2.csv")  
str(activitydata)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

Date is of a factor class, need to switch to Date.

activitydata$date <- as.Date(activitydata$date)  
str(activitydata)

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

Transformed data into proper data set. Now remove NA from alternative data set.

activitydata2 <- subset(activitydata, activitydata$steps != 'NA')

What is mean total number of steps taken per day?

For this part of the assignment, you can ignore the missing values in the dataset.

1. Total number of steps taken per day. Using data set removing NA.

library(dplyr)

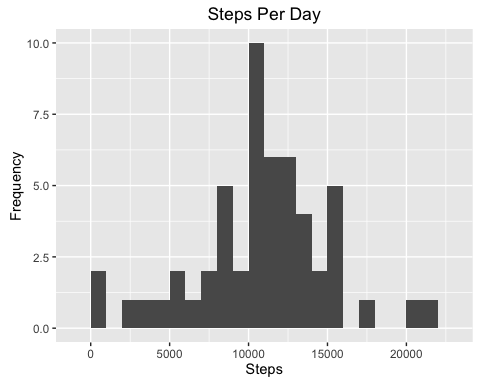
##   
## Attaching package: 'dplyr'  
##   
## The following objects are masked from 'package:stats':  
##   
## filter, lag  
##   
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

stepsperday <- activitydata2 %>% group\_by(date) %>% summarize(steps = sum(steps)) %>% print

## Source: local data frame [53 x 2]  
##   
## date steps  
## (date) (int)  
## 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  
## 7 2012-10-09 12811  
## 8 2012-10-10 9900  
## 9 2012-10-11 10304  
## 10 2012-10-12 17382  
## .. ... ...

1. Histogram of the total number of steps.

library(ggplot2)  
ggplot(stepsperday,aes(steps)) + geom\_histogram(binwidth = 1000) + labs(title = "Steps Per Day", x="Steps", y = "Frequency")



1. Mean and median of the total number of steps per day.

mean(stepsperday$steps)

## [1] 10766.19

median(stepsperday$steps)

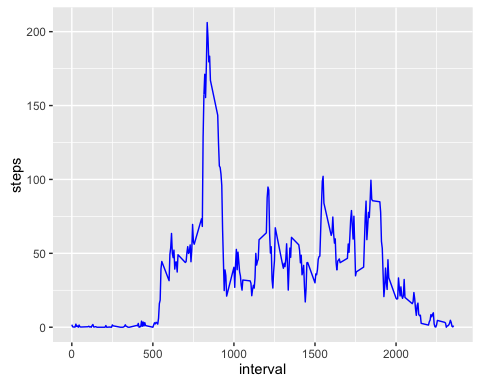
## [1] 10765

What is the average daily activity pattern? 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).

avginterval <- activitydata2 %>% group\_by(interval) %>% summarize(steps = mean(steps)) %>% print

## Source: local data frame [288 x 2]  
##   
## interval steps  
## (int) (dbl)  
## 1 0 1.7169811  
## 2 5 0.3396226  
## 3 10 0.1320755  
## 4 15 0.1509434  
## 5 20 0.0754717  
## 6 25 2.0943396  
## 7 30 0.5283019  
## 8 35 0.8679245  
## 9 40 0.0000000  
## 10 45 1.4716981  
## .. ... ...

ggplot(avginterval,aes(interval,steps)) + geom\_line(color = "blue")



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

filter(avginterval,avginterval$steps == max(avginterval$steps))

## Source: local data frame [1 x 2]  
##   
## interval steps  
## (int) (dbl)  
## 1 835 206.1698

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)

sum(is.na(activitydata$steps))

## [1] 2304

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.

Using the average step per interval data set created above, use that for filling in the na values.

narows <- filter(select(activitydata,interval,date),is.na(activitydata$steps))  
datafill <- right\_join(narows,avginterval,by="interval")

Now insert rows fixing NA values

fix\_actdata <- bind\_rows(activitydata2,select(datafill,steps,date,interval))

check NA values

sum(is.na(fix\_actdata))

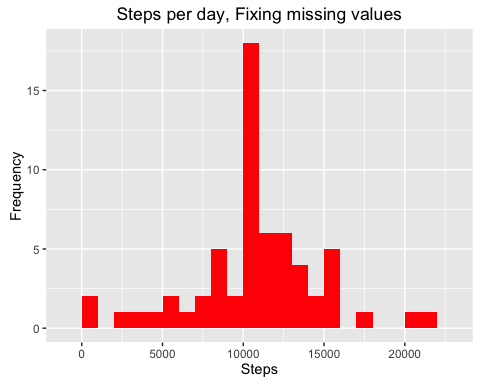
## [1] 0

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?

steps\_fixd <- fix\_actdata %>%  
 filter(!is.na(steps)) %>%  
 group\_by(date) %>%  
 summarize(steps = sum(steps)) %>%  
 print

## Source: local data frame [61 x 2]  
##   
## date steps  
## (date) (dbl)  
## 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  
## 7 2012-10-07 11015.00  
## 8 2012-10-08 10766.19  
## 9 2012-10-09 12811.00  
## 10 2012-10-10 9900.00  
## .. ... ...

ggplot(steps\_fixd, aes(x = steps)) +  
 geom\_histogram(fill = "red", binwidth = 1000) +  
 labs(title = "Steps per day, Fixing missing values", x = "Steps", y = "Frequency")



Mean and Median

mean(steps\_fixd$steps)

## [1] 10766.19

median(steps\_fixd$steps)

## [1] 10766.19

The changes is that the mean and median are exactly the same.

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.

fix\_actdata<- mutate(fix\_actdata, weektype = ifelse(weekdays(fix\_actdata$date) == "Saturday" | weekdays(fix\_actdata$date) == "Sunday", "weekend", "weekday"))  
head(fix\_actdata)

## Source: local data frame [6 x 4]  
##   
## steps date interval weektype  
## (dbl) (date) (int) (chr)  
## 1 0 2012-10-02 0 weekday  
## 2 0 2012-10-02 5 weekday  
## 3 0 2012-10-02 10 weekday  
## 4 0 2012-10-02 15 weekday  
## 5 0 2012-10-02 20 weekday  
## 6 0 2012-10-02 25 weekday

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.

interval\_fixd <- fix\_actdata %>%  
 group\_by(interval, weektype) %>%  
 summarise(steps = mean(steps))  
ggplot(interval\_fixd, aes(x=interval, y=steps, color = weektype)) +  
 geom\_line() +  
 facet\_wrap(~weektype, ncol = 1, nrow=2)

