Activity

Marius Nick

15 decembrie 2015

## Introduction

Personal activity monitoring devices allow for the collection of large amounts of data about personal movement and physical activity. These measurements can be used to improve a person’s health and to find behaviour patterns.

In this assignment I’ll analyse data from a personal activity monitoring device. This device collects data at 5 minute intervals throughout 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.

## Load Data and preprocesing

Data for the assignment is provided in the form of a zipped CSV format file. The following code block unzips the data file (if necessary) and loads the data into a data frame.

library(lubridate)  
library(stringr)

## Warning: package 'stringr' was built under R version 3.1.3

setwd("~/R/Reproducible Research\_1")  
activity<-read.csv("activity.csv", sep=",")  
summary(activity)

## steps date interval   
## Min. : 0.00 2012-10-01: 288 Min. : 0.0   
## 1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8   
## Median : 0.00 2012-10-03: 288 Median :1177.5   
## Mean : 37.38 2012-10-04: 288 Mean :1177.5   
## 3rd Qu.: 12.00 2012-10-05: 288 3rd Qu.:1766.2   
## Max. :806.00 2012-10-06: 288 Max. :2355.0   
## NA's :2304 (Other) :15840

#single days   
date<-unique(activity$date)  
#

## Calculate the total number of steps taken per day

i<-rep(0,length(date))  
stepsdays<-rep(0,length(date))  
#total steps per day  
for(i in 1:length(date)){stepsdays[i]<-sum(activity[activity$date==date[i],]$steps,na.rm = TRUE)}

## Mean and median of steps of days

Mean

mean(stepsdays)

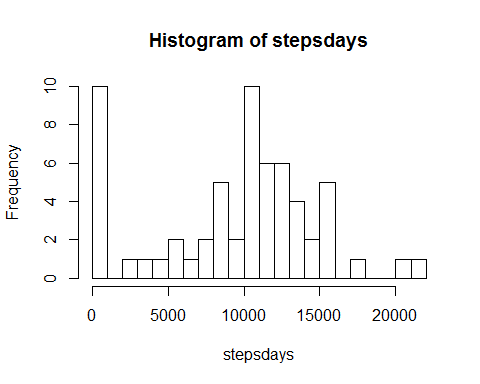
## [1] 9354.23

Median

median(stepsdays)

## [1] 10395

## Histogram of the total number of steps taken each day



Here the mean total number of steps is 9354. But, there are 8 days where the device records zero steps (note that the minimum of totsteps is 0). We can remove this days.

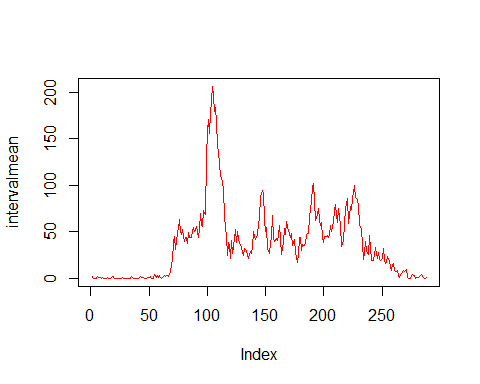
## Average daily activity pattern

time<-(strptime(str\_pad(as.character(activity[,3]),4,side="left",pad="0"),"%H%M"))  
#dattee days in activity as format time   
dattee<-strptime(as.character.Date(activity[,2]),"%Y-%m-%d")  
year(time)<-year(dattee)  
day(time)<-day(dattee)  
month(time)<-month(dattee)  
datte<-strptime(as.character.Date(date),"%Y-%m-%d")

## Average daily activity pattern

Now we create a profile of the activity in an “average” day.

# average number of steps taken, averaged across all days   
interval<-unique(activity$interval)  
 j<-0  
 intervalmean<-rep(0,length(interval))  
for( j in 1:length(interval)){intervalmean[j]<-mean(activity[activity$interval==interval[j],]$steps,na.rm=TRUE)}  
#plot average   
plot(intervalmean,type="l", col="red")



# Maximum step

max(intervalmean)

## [1] 206.1698

for (i in 1:length(interval)) {if ( max(intervalmean)==intervalmean[i]) print(interval[i]) }

## [1] 835

The maximum average number of steps is 206.20 occurs at 835.

# Number of Na.s

There are a number of rows with NA values in the data. These missing data may introduce bias in summaries or calculations. First we find the total number of rows which contain missing data. From the summary that I printed after loading the data at the start, I know that missing entries only occur in the steps column.

length(activity[is.na(activity$steps)==TRUE,]$steps)

## [1] 2304

activity1<-activity

## Imputing missing values

So are, 2304 rows contain missing data. We’ll replace rows with missing data with the mean value for that time interval (after recording which rows originally had missing data in activity, so as to be able to revert the changes should I need to). The corrected data is stored in a new data frame activity1.

interval<-unique(activity$interval)  
activity1<-activity  
  
intervalmean<-as.data.frame(cbind(intervalmean,interval))  
for(i in 1:length(activity1$steps))  
 {if(is.na(activity1$steps[i]))  
 {activity1$steps[i]<-intervalmean[intervalmean$interval==activity1$interval[i],]$intervalmean }   
 }  
  
stepsdays1<-rep(0,length((date)))  
for(i in 1:length(date)){stepsdays1[i]<-sum(activity1[activity1$date==date[i],]$steps)}

## Mean, median hitogram of steps of days whitout missing

summary((stepsdays1))

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 41 9819 10770 10770 12810 21190

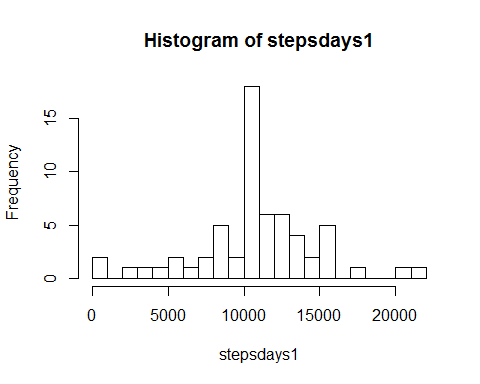
mean(stepsdays1)

## [1] 10766.19

median(stepsdays1)

## [1] 10766.19

hist(stepsdays1,n=20)



The mean total number of steps for the corrected dataset is 10767 steps, and the median is 10767 steps. Correcting the data we have a effect(small).

## Activity patterns in weekdays and weekends

wdat<-weekdays(datte)  
# Set my locale to en\_US so that the weekdays() function outputs names of days in English.   
Sys.setlocale(category = "LC\_ALL", locale = "English\_United States.1252")

## [1] "LC\_COLLATE=English\_United States.1252;LC\_CTYPE=English\_United States.1252;LC\_MONETARY=English\_United States.1252;LC\_NUMERIC=C;LC\_TIME=English\_United States.1252"

activity[,4]<-weekdays(strptime(as.character.Date(activity[,2]),"%Y-%m-%d"))  
act\_weekday<-activity[!(activity[,4]%in% c("Sunday","Saturday")),]  
act\_weekend<-activity[(activity[,4]%in% c("Sunday","Saturday")),]  
 in\_mean\_weekend<-rep(0,length(interval))  
for( j in 1:length(interval)){in\_mean\_weekend[j]<-mean(act\_weekend[act\_weekend$interval==interval[j],]$steps,na.rm=TRUE)}  
 in\_mean\_weekday<-rep(0,length(interval))  
for( j in 1:length(interval)){in\_mean\_weekday[j]<-mean(act\_weekday[act\_weekday$interval==interval[j],]$steps,na.rm=TRUE)}

We plot the time series for mean number of steps, separating weekdays (top panel, red) from weekends (bottom panel, blue).

par(mfrow=c(2,1),mar=c(4,4,2,1))  
plot(in\_mean\_weekday,type="l", col="red")  
plot(in\_mean\_weekend,type="l", col="blue")

