Course Project 1 Reproducible Research

Bliztacular

September 19, 2015

# Loading and preprocessing the data

Let's load the data (i.e. read.csv()), then process/transform the data (if necessary) into a format suitable for our analysis.

htr = c("integer", "character", "integer")  
actdata <- read.csv("activity.csv", head=TRUE, colClasses=htr, na.strings="NA")  
head(actdata)

## 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

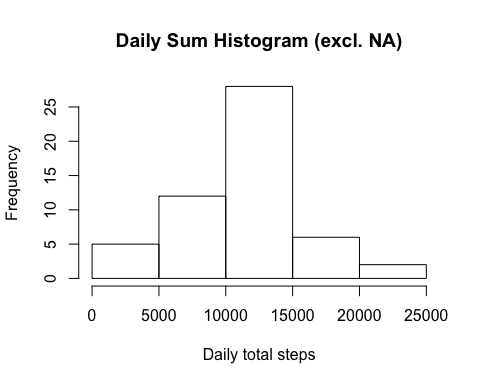
Now let's change the date to be readable as a date, and save a copy of the dataset with the NA values removed.

actdata$date <- as.Date(actdata$date)  
actnna <- subset(actdata, !is.na(actdata$steps))

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

Ignoring the missing values, let's calculate the total number of steps taken per day, make a histogram of the total number of steps taken each day, and then after we'll calculate and report the mean and median of the total number of steps taken per day.

dailysum <- tapply(actnna$steps, actnna$date, sum, na.rm=TRUE, simplify=T)  
dailysum <- dailysum[!is.na(dailysum)]  
  
hist(x=dailysum,  
 xlab="Daily total steps",  
 ylab="Frequency",  
 main="Daily Sum Histogram (excl. NA)")



Now that we've looked at the distribution a bit, let's calculate the mean and median.

Mean:

rmean <- mean(dailysum)

Median:

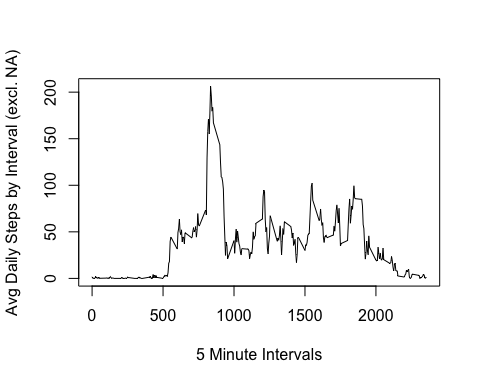
rmedian <- median(dailysum)

The mean is 1.076618910^{4} and the median is 10765.

## What is the average daily activity pattern?

Let's 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). Looking at the time series, which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

int\_avg <- tapply(actnna$steps, actnna$interval, mean, na.rm=TRUE, simplify=T)  
df\_ia <- data.frame(interval=as.integer(names(int\_avg)), avg=int\_avg)  
  
with(df\_ia,  
 plot(interval,  
 avg,  
 type="l",  
 xlab="5 Minute Intervals",  
 ylab="Avg Daily Steps by Interval (excl. NA)"))



So, which interval contains the maximum number of steps?

max\_steps <- max(df\_ia$avg)  
df\_ia[df\_ia$avg == max\_steps, ]

## interval avg  
## 835 835 206.1698

OK, so now we know that the busiest time has 206.1698113. Good to know!

## Imputing missing values

Let's calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

sum(is.na(actdata$steps))

## [1] 2304

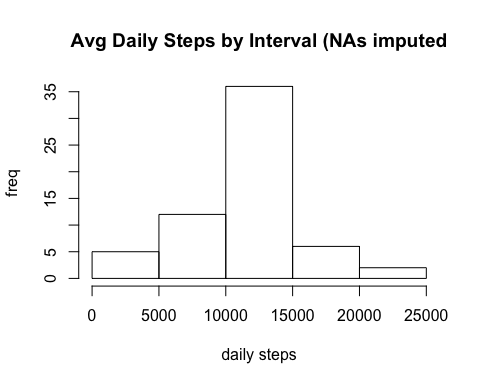
So the total number of missing variables is sum(is.na(actdata$steps)).

Now let's replace the missing values with the mean/median for that day, or the mean for that 5-minute interval, etc., then create a new dataset that is equal to the original dataset but with the missing data filled in.

df\_impute <- actdata  
nax <- is.na(df\_impute$steps)  
int\_avg <- tapply(actnna$steps, actnna$interval, mean, na.rm=TRUE, simplify=T)  
df\_impute$steps[nax] <- int\_avg[as.character(df\_impute$interval[nax])]

Now let's 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?

new\_dailysum <- tapply(df\_impute$steps, df\_impute$date, sum, na.rm=TRUE, simplify=T)  
  
hist(x=new\_dailysum,  
 xlab="daily steps",  
 ylab="freq",  
 main="Avg Daily Steps by Interval (NAs imputed")



And now to look at the mean and median of the new dataset with imputed values to replace the NAs:

Mean:

zmean <- mean(new\_dailysum)

Median:

zmedian <- median(new\_dailysum)

The new mean is 1.076618910^{4} and the new median is 1.076618910^{4}. Not too different!

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

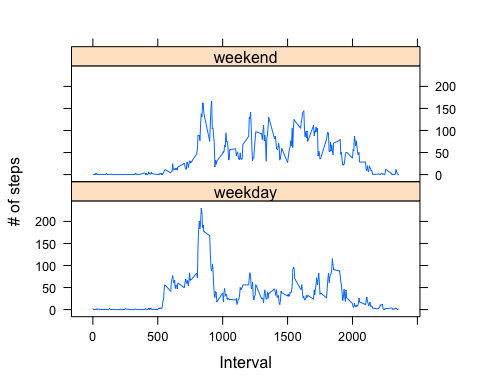
We'll use the weekdays() function here, and create a new factor variable in the dataset with two levels – “weekday” and “weekend” indicating whether a given date is a weekday or weekend day. Then we'll show the first few rows of the data with a 'wk' column added.

is\_weekday <- function(d) {  
 wd <- weekdays(d)  
 ifelse (wd == "Saturday" | wd == "Sunday", "weekend", "weekday")  
}  
  
wx <- sapply(df\_impute$date, is\_weekday)  
df\_impute$wk <- as.factor(wx)  
head(df\_impute)

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

Finally, we'll 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).

wk\_df <- aggregate(steps ~ wk+interval, data=df\_impute, FUN=mean)  
  
library(lattice)  
xyplot(steps ~ interval | factor(wk),  
 layout = c(1, 2),  
 xlab="Interval",  
 ylab="# of steps",  
 type="l",  
 lty=1,  
 data=wk\_df)



Looks like weekdays have a morning spike that we don't see with weekend days.

# Submitting the Assignment

Commit the your completed PA1\_template.Rmd file to the master branch of your git repository (you should already be on the master branch unless you created new ones)

Commit your PA1\_template.md and PA1\_template.html files produced by processing your R markdown file with knit2html() function in R (from the knitr package) by running the function from the console.

If your document has figures included (it should) then they should have been placed in the figure/ directory by default (unless you overrided the default). Add and commit the figure/ directory to yoru git repository so that the figures appear in the markdown file when it displays on github.

Push your master branch to GitHub.