Reproducibility on Activity Monitoring

Diego-MX

May 2016

This is a class project for **Coursera** and **Johns Hopkins**' course on Reproducible Research. For a link to the project's description click here.

Thanks to *Roger* -who runs this course- together with Jeff and Brian who run the whole Data Science Specialization from Johns Hopkins University.

For compiling the document I use `knitr` package in RStudio IDE with R. Other packages that are used for performing the required tasks can be seen in the following code. Initializing options are set as well.

```r
library(knitr);  library(magrittr);  library(lubridate);
library(dplyr);  library(reshape2);  library(data.table);
library(ggplot2);

inline_hook <- function(x) {
  require(magrittr)

  if (is.numeric(x)) {
    digits_ <- getOption('digits')
    x <- signif(x, digits=digits_) %>%
      formatC(digits=digits_, format='fg')
  }
  x_str <- paste(as.character(x), collapse=", ")
  return (x_str)
}

options(digits=3)
opts_chunk$set(fig.width=6, fig.height=3, fig.align='center')
knit_hooks$set(inline=inline_hook)
opts_knit $set(fig.path="./figure")
```

Loading and Preprocessing Data

To start with the data, let's consider several possibilities for accessing it. That is, starting from a url from which to download it, or a zip file to extract, or csv file to read from.
In addition, the data is stored as a *data table* and its dates transformed. The interval is also manipulated into its own class.

```r
act_file <- "./activity.csv"
act_zip  <- "./activity.zip"
act_url  <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"

if (!file.exists(act_file)) {
  if (!file.exists(act_zip)) {
    download.file(act_url, act_zip, method='curl')
  }
  unzip(act_zip)
}
```

```
stepsData <- read.csv(act_file) %>%
  as.data.table %>%
  mutate(date = ymd(date),
    hour    = floor(interval/100),
    minute  = interval %% 100,
    daytime = hm(paste(hour, minute))
  )
```

The data table `stepsData` consists of varibles `steps, date, interval, hour, minute, daytime` for about `17600` observations. As mentioned in the description, these are the measurements for an anonymous individual, call him Mr. Anon, during two months in 5 minute intervals.

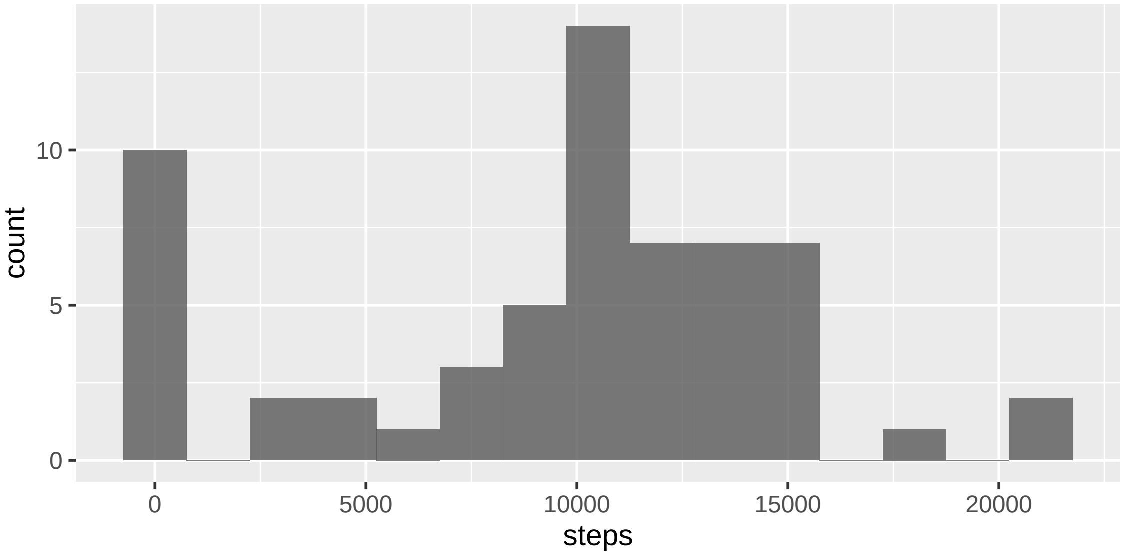
The last four variables are closely related; in particular `interval` and `daytime` differ only in their class representation. While `daytime` captures the whole meaning of such variable, it presented issues in the computations. Thus we remove it, and instead create a simple funtion to display it when necessary.

```
stepsData %<>% select(-c(hour, minute, daytime))

int2time <- function(int) {
  hour   <- floor(int/100)
  minute <- int %% 100
  str    <- paste(hour, minute, sep=":")
  return(str)
}
```

Steps per Day

Continue by totaling the steps taken by Mr. Anon each day and examine their overall measurements. The reader might recognise the use of *dplyr* methods to achieve just this.

```
byDay <- group_by(stepsData, date) %>%
  summarise(steps = sum(steps, na.rm=T))

ggplot(byDay, aes(steps)) +
  geom_histogram(binwidth=1500, alpha=0.8)
```

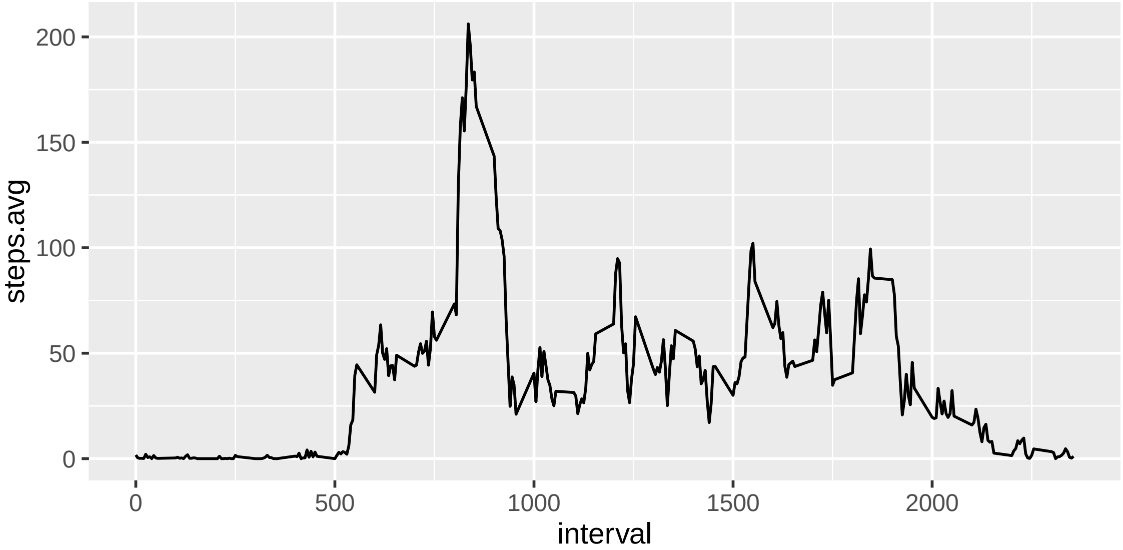


We are asked to compute the mean and median from the steps taken by Mr. Anon, which are approximately 9350 and 10400 respectively.
Observe that there are many days with a total count less than 1500. This is highly unlikely given the rest of the distribution for this particular graph.

Daily Activity Pattern

This section asks about Mr. Anon's average daily pattern of steps. As before, observations can be grouped with *dplyr* package, and we plot the resulting averages through the day.

```
alongDay <- group_by(stepsData, interval) %>%
  summarise(steps.avg = mean(steps, na.rm=T))

ggplot(alongDay, aes(interval, steps.avg)) + geom_line()
```



Moreover, the interval where Mr. Anon took the most steps on average is 8:35, which accounted to 206 of them. That whole period between, say 8:00 and 9:00 is one highly stepped hour of the days of Mr. Anon.

A hypothesis for this time being the highest is that Mr. Anon exercises regularly at this time; and this is more viable than him walking to work because it is not compensated by a regular time to come back from work.

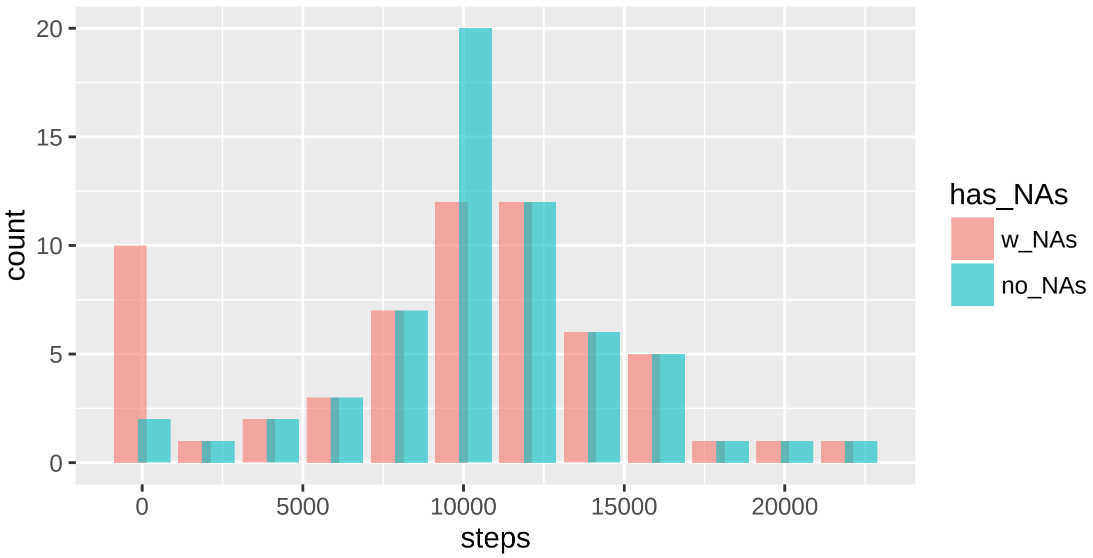
Missing Values

It is noted that there are several missing values NAs in the data. More specifically 13.1 % of the observations are missing, which account to 13.1 % of days which totaled zero.

Having these two percentages match suggests that the missing values were taken during whole days; a glance of the first dates in the table byDay suggests that once a week -probably Sunday- Mr. Anon was not recording his steps.

In any case, the missing values are supplied with the average for the corresponding interval, which is itself imported from the variable alongDay that was calculated earlier. The following code does that, and place histograms side by side.

```
stepsData %<>% left_join(alongDay, "interval") %>%
  mutate(steps_ = ifelse(is.na(steps), steps.avg, steps))

byDay_mod <- group_by(stepsData, date) %>%
  summarise(w_NAs = as.double(sum(steps, na.rm=T)),
            no_NAs = sum(steps_)) %>%
  melt(id="date", variable="has_NAs", value="steps") %>%
  mutate(jitter = 0.2*(has_NAs=="w_NAs"))

ggplot(byDay_mod, aes(steps, fill=has_NAs)) +
  geom_histogram(binwidth=2000, alpha=0.6,
    position=position_dodge(width=1500))
```



The comparisson between the histograms shows a difference only in days with 0-2 thousand steps and 10-12 thousand. I suspect this happens because the ones with NAs were first counted as having 0 steps and when averaged they all fell in the same bin between 10-12 thousand steps.

The means and medians can be displayed from the following code.

```
byDay_mod %>%
  group_by(has_NAs) %>%
  summarise_each(funs(mean, median), vars=steps)
```

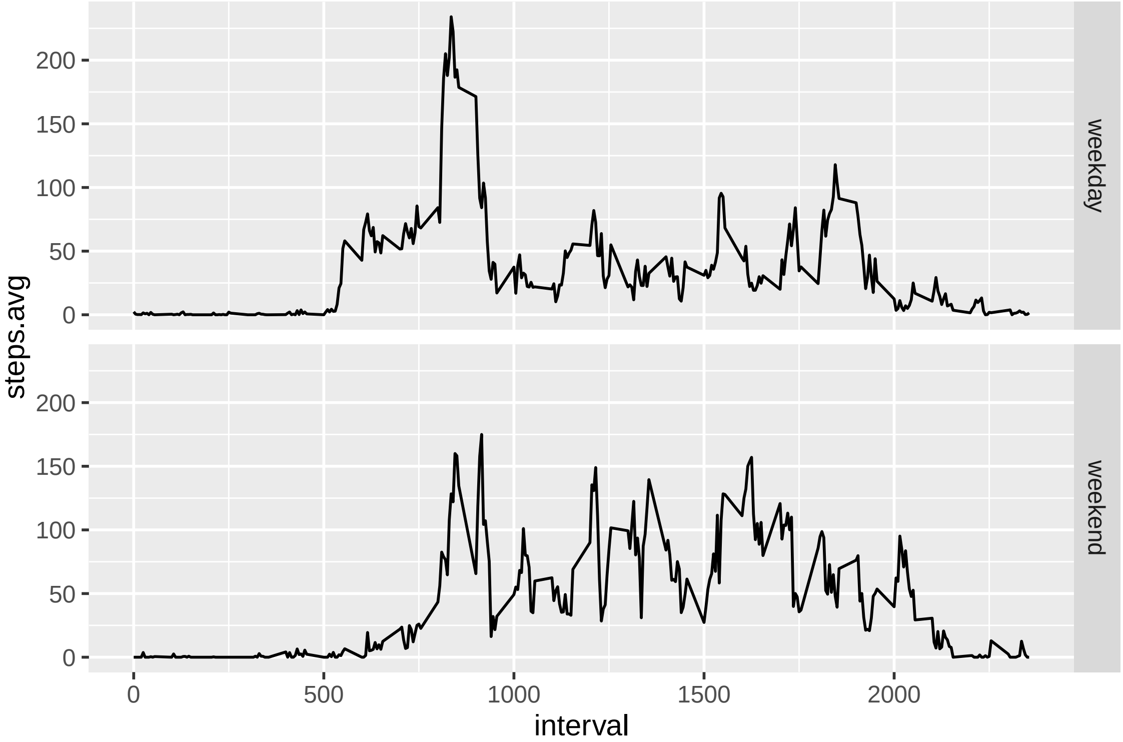
```
## Source: local data table [2 x 3]
##
##   has_NAs  mean median
##    (fctr) (dbl)  (dbl)
## 1   w_NAs  9354  10395
## 2  no_NAs 10766  10766
```

The change in mean is expected because of the shift of the days count that was observed in the histogram. However the change in median is not necessarily so; the fact that it did change says that the resulting placement of the NA observations -that is their average's average- is larger than the original median.

Weekend Patterns

Next, we'll predict the weekend patterns by including a variable for the date being in a weekend. Since we suspect that Mr. Anon didn't measure his steps during the weekend, we'll consider our original measurements.

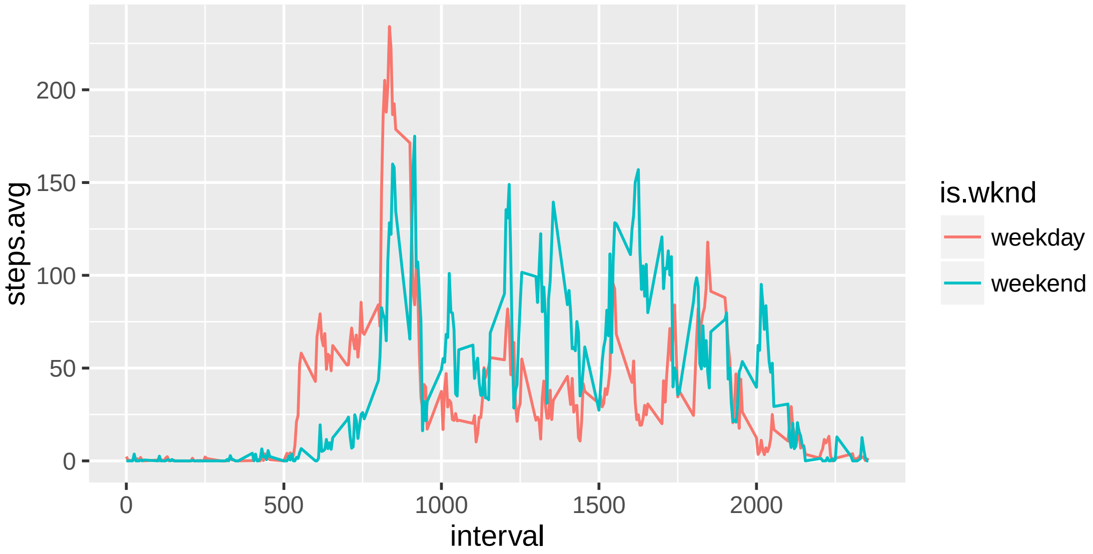
```
stepsData %<>% mutate(is.wknd = abs(wday(date)-4)==3,
  is.wknd = factor(is.wknd, labels=c("weekday", "weekend")))

along_wknd <- group_by(stepsData, is.wknd, interval) %>%
  summarise(steps.avg = mean(steps, na.rm=T))

ggplot(along_wknd, aes(interval, steps.avg)) + geom_line() +
  facet_grid(is.wknd ~ .)
```



This panel plot is not the most illustrative of plots, because it doesn't show whether Mr. Anon walks more during the weekdays. But this is the plot that was requested in the project, so we'll just stick with it.

A bonus plot is one that shows more clearly the different steps patterns during the day.

```
ggplot(along_wknd, aes(interval, steps.avg, colour=is.wknd)) + geom_line()
```



Slightly more clear, you can see the shift of steps taken during that `8:00-9:00` period distributed more uniformly along the whole day.