Reproducible Research Peer Assessment 1

M Hamersma

December 2019

# Movement Activity Monitoring

## 1. Loading and preprocessing data

The script will run correctly if the file “activity.csv” has been downloaded and unzipped to the working directory. Library the dplyr and ggplot packages.

setwd("~/R5")  
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)  
activity <- read.csv("activity.csv", stringsAsFactors = FALSE)  
activity$date <- as.Date(activity$date)  
names(activity)

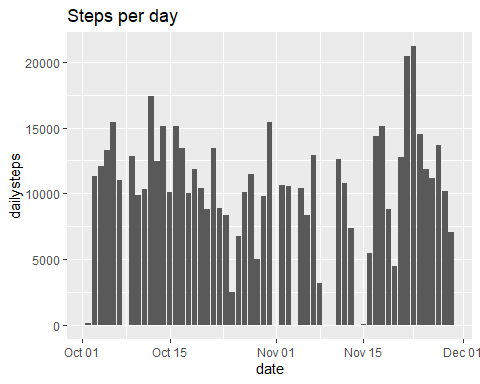
## [1] "steps" "date" "interval"

## 2. Calculate total number of steps per day and plot histogram

Convert to a data table using dplyr, group the table by date, then create a new dataframe called **stps** containing the total number of steps per day, called **dailysteps**.

act <- tbl\_df(activity)  
stps <- act %>% group\_by(date) %>% summarise(dailysteps = sum(steps))  
g1 <- ggplot(stps, aes(x = date, y = dailysteps)) + geom\_bar(stat = "identity") + ggtitle("Steps per day")  
print(g1)

## Warning: Removed 8 rows containing missing values (position\_stack).



## 3. Mean and Median Steps per day

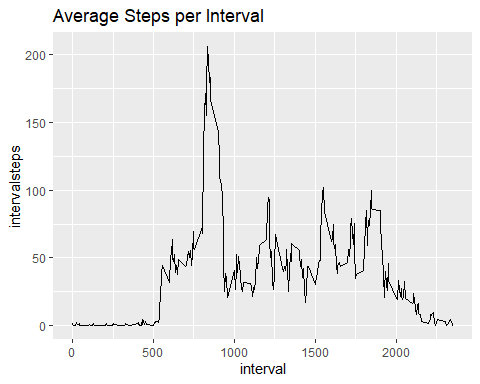
meanstps <- as.integer(mean(stps$dailysteps, na.rm=TRUE))  
medianstps <- as.integer(median(stps$dailysteps, na.rm=TRUE))

The Mean number of steps per day is **10766** and the median is **10765**. Reported in Integer, as decimals on such a large number is meaningless.

## 4. Average Daily Activity Pattern

Calculate the average steps per interval into a new dataframe called **stps2**, then plot the **steps (y)** against the **interval (x)**.

stps2 <- act %>% group\_by(interval) %>% summarise(intervalsteps = mean(steps, na.rm=TRUE))  
g2 <- ggplot(stps2, aes(x = interval, y = intervalsteps)) + geom\_line() + ggtitle("Average Steps per Interval")  
print(g2)



## 5. Interval with Maximum Steps

Determine the maximum of the averages, then identify the interval number with that value.

max\_interval <- max(stps2$intervalsteps)  
stps3 <- stps2 %>% filter(intervalsteps == max\_interval)  
max\_interval\_no <- as.integer(stps3[1,2])  
max\_interval <- as.integer(stps3[1,1])

The Interval with the maximum average steps is number **206**, averaging **835** steps.

## 6. Count the records with missing values

is\_complete <- complete.cases(act)  
missing\_val <- sum(!is\_complete)

There are **2304** records with missing values.

## 7. 8. and 9. Replace (impute) the missing values with Averages for that Interval

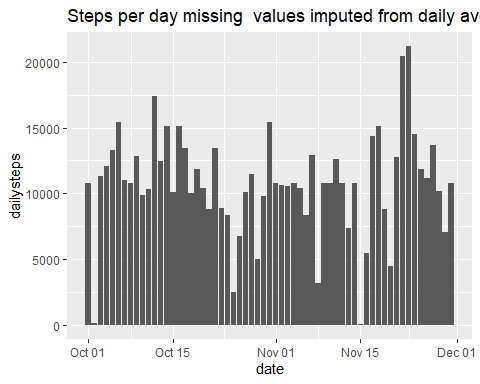
Find the records in the original dataframe **act** where steps = NA, then replace with it with the average for that interval from the stps2 data frame, saving in new imputed dataframe **act\_imputed**

interval\_avg <- act %>%  
 group\_by(interval) %>%  
 summarise(avg\_steps = mean(steps, na.rm = TRUE))  
  
act\_imputed <- act  
  
for (i in 1:nrow(act\_imputed))   
{  
 if (is.na(act\_imputed[i, "steps"]) == TRUE)   
 {  
 intervalno <- as.integer(act\_imputed[i, "interval"])  
 imputed\_value <- interval\_avg[interval\_avg$interval == intervalno, "avg\_steps"]  
 act\_imputed[i, "steps"] <- imputed\_value  
 }   
 else   
 {  
 act\_imputed[i, "steps"] <- act[i, "steps"]  
 }  
}

## 9. Histogram of steps with NA repalce with daily average

Convert to a data table using dplyr, group the table by date, then create a new dataframe called **stps\_imputed** containing the total number of steps per day, called **dailysteps**, and calculate the mean and median

stps\_imputed <- act\_imputed %>% group\_by(date) %>% summarise(dailysteps = sum(steps))  
g3 <- ggplot(stps\_imputed, aes(x = date, y = dailysteps)) + geom\_bar(stat = "identity") + ggtitle("Steps per day missing values imputed from daily average")  
print(g3)



meanstps\_imputed<- as.integer(mean(stps\_imputed$dailysteps, na.rm=TRUE))  
medianstps\_imputed <- as.integer(median(stps\_imputed$dailysteps, na.rm=TRUE))

The Mean number of steps per day when imputing missing values is **10766** vs 10766 and the median is **10766** vs 10765.

## 10. Compare the steps on weekdays versus weekends

First define if the date falls in a weekend or not, then double-plot with facets wrap on weekday indicator.

act\_imputed$date <- as.Date(act\_imputed$date)  
act\_imputed$weekday\_indicator <- ifelse(weekdays(act\_imputed$date) %in% c("Saturday", "Sunday"), "Weekend", "Weekday")  
  
stps4 <- act\_imputed %>% group\_by(interval, weekday\_indicator) %>% summarise(interval\_steps = mean(steps, na.rm = TRUE))  
  
ggplot(stps4, aes(x = interval, y = interval\_steps)) +  
 geom\_line(color = "blue") +  
 facet\_wrap(~weekday\_indicator)

