# Reproducible Research: Peer Assessment 1

Andrew Dieterich

2022-05-21

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

## Loading and preprocessing the data

```
## Are there differences in activity patterns between weekdays and weekends?
## reading in the data
```

## viewing the data table from CSV file and its dimensions, and a preview:

```
data <- read.csv("activity.csv", header = TRUE)
View(data)
  str(data)
```

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

```
  head(data)
```

```
##   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
```

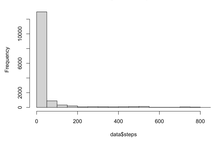
```
  dim(data)
```

```
## [1] 17568     3
```

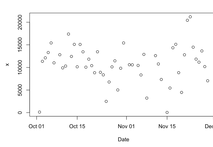
## What is mean total number of steps taken per day?

# histogram of steps taken

```
  hist(dataSsteps, main = "Histogram of Steps")
```




```
  # find steps taken by day
  ## first change the date column with 'as.Date()'
  dataSdate <- as.Date(as.character(dataSdate, "%Y-%m-%d"));
  daily_sum <- aggregate(dataSsteps, by=list(Date=dataSdate), FUN=sum)
  View(daily_sum)
  plot(daily_sum)
```




```
  ## summary of steps per day including Median and Mean
  table <- summary(daily_sum$x)

  ## Median and Mean steps taken per day:
  print(table[3:4])
```

```
## Median   Mean
## 10765.0 10766.19
```

## What is the average daily activity pattern?

```
  # 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)

  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
```

```
  AVG <- data %>%
    group_by(interval) %>%
    summarize(avg = mean(steps, na.rm = TRUE))
  AVG
```

```
## # A tibble: 288 × 2
##    interval     avg
##       <int>   <dbl>
##  1        0  1.72
##  2        5  0.340
##  3       10  0.132
##  4       15  0.151
##  5       20  0.0755
##  6       25  2.09
##  7       30  0.528
##  8       35  0.868
##  9       40  0
## 10       45  1.47
## # … with 278 more rows
```

```
  View(AVG)
  ## checking that the average vector of means is equal to the
  ## length of the number of time intervals in the data file:

  dim(AVG)
```

```
## [1] 288   2
```

```
  length(unique(data$interval))
```

```
## [1] 288
```

```
  # plotting a line graph with average daily steps divided by interval:
  plot(AVG, type = 'l', main = "AVG daily steps by interval")
```


```
  print("This is the interval with the maximum average steps:")
```

```
## [1] "This is the interval with the maximum average steps:"
```

```
  AVG$interval[which.max(AVG$avg)]
```

```
## [1] 835
```

## Imputing missing values

```
  print("Total number of missing values in the data set:")
```

```
## [1] "Total number of missing values in the data set:"
```

```
  sum(is.na(dataSsteps))
```

```
## [1] 2304
```

```
  # filling in missing values and re-plotting the graph of daily averages:
  library(tidyr)
  library(dplyr)

  ## making a copy of the data set
  data2 <- data
  View(data2)

  ## imputing missing values with the overall mean:
  data2$steps[is.na(data2$steps)] <- mean(data2$steps, na.rm=TRUE)
```

## re-doing the previous steps with missing values now added

## first the histogram:

```
  hist(data2$steps, main = "Histogram of Steps with missing values filled in")

  # total steps taken by day
  ## first change the date column with 'as.Date()'
  data2$date <- as.Date(as.character(data2$date, "%Y-%m-%d"));
  daily_sum2 <- aggregate(data2$steps, by=list(Date=data2$date), FUN=sum)
  View(daily_sum2)
  plot(daily_sum2)

  ## summary of steps per day including Median and Mean
  table2 <- summary(daily_sum2$x)

  ## Median and Mean steps taken per day:
  print("Median and Mean with missing values imputed:")
  print(table2[3:4])

  # 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)

  library(dplyr)
  AVG2 <- data %>%
    group_by(interval) %>%
    summarize(avg = mean(steps, na.rm = TRUE))
  AVG2
  View(AVG2)
  ## checking that the average vector of means is equal to the
  ## length of the number of time intervals in the data file:

  dim(AVG2)
  length(unique(data2$interval))

  # plotting a line graph with average daily steps divided by interval:
  plot(AVG2, type = 'l', main = "AVG daily steps by interval, missing values filled in")
```

## Changing days to either weekday or weekend

## To then make a two-graph plot with average steps divided by interval

## For both weekdays, and weekends in order to examine any differences

```
  library(dplyr)
  data2$weekdays <- weekdays(data2$date)
  weekends <- c("Saturday", "Sunday")
    data2 <- filter(data2, weekdays %in% weekends)

    weekday_list <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday")
    data2 <- filter(data2, weekdays %in% weekday_list)

    ## checking the number of rows equals the original data set
    nrow(data2) - nrow(data2) == nrow(data2)
```

```
## [1] 0
```

```
    data2$weekdays <- c("weekend")
    data2$weekdays <- c("weekday")

    rbind(data2, data2) -> data3
    View(data3)
    nrow(data3)
```

```
## [1] 17568
```

```
  # grouping
  AVG3 <- data3 %>%
    group_by(weekdays, interval) %>%
    summarize(avg = mean(steps, na.rm = TRUE))

## `summarise()` has grouped output by 'weekdays'. You can override using the
## `.groups` argument.
```

```
  AVG3
```

```
## # A tibble: 576 × 3
## # Groups:   weekdays [2]
##    weekdays interval   avg
##    <chr>       <int> <dbl>
##  1 weekday         0  7.01
##  2 weekday         5  5.38
##  3 weekday        10  5.14
##  4 weekday        15  5.16
##  5 weekday        20  5.07
##  6 weekday        25  6.49
##  7 weekday        30  5.61
##  8 weekday        35  6.01
##  9 weekday        40  4.98
## 10 weekday        45  6.58
## # … with 566 more rows
```

```
  View(AVG3)
  ## checking that the average vector of means is equal to the
  ## length of the number of time intervals in the data file:

  dim(AVG3)
```

```
## [1] 576   3
```

```
  length(unique(data3$interval))
```

```
## [1] 288
```

## Including the final plot

## plotting a line graph with average daily steps divided by interval:

```
  library(ggplot2)

  ggplot(AVG3, aes(x = interval, y = avg, color = weekdays))+
    geom_line()+facet_grid(rows = vars(weekdays))
```