title: “Reproducible Research”

Peer Assessment 1

author: “Henrik Gjerning” date: “10 August 2015” output: html_document

1. Preparing setup and environment:

```
setwd("~/G-ART/artData/Coursera/ReproducibleResearch")
if(!file.exists("./Assignment1")){dir.create("./Assignment1")}
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 3.2.1
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.1
```

```
library(RCurl)
```

```
## Warning: package 'RCurl' was built under R version 3.2.1
```

```
## Loading required package: bitops
```

```
#knit2html = function(input, output = NULL, ...){
#  out = knit(input, output)
#  owd = setwd(dirname(out)); on.exit(setwd(owd))
#  markdown::markdownToHTML(basename(out), ...)
#}
#knit2html("./Assignment1/PA1_Template.md", "./Assignment1/PA1_Template.html")
echo = TRUE
```

2. Download of data:

```
if(!file.exists("./Assignment1/activity.csv")) {
setInternet2(use = TRUE)
fileUrl <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(fileUrl,destfile="./Assignment1/Data.zip")
unzip(zipfile="./Assignment1/Data.zip",exdir="./Assignment1")
}
```

3. Reading the data:

```
na.class <- c("NA", "<NA>")
variable.class <-  c("numeric","Date","numeric")
personal.activity = read.csv("./Assignment1/activity.csv",
header=TRUE, na.strings = na.class, colClasses=variable.class)
```

A. Assignment

What is mean total number of steps taken per day?

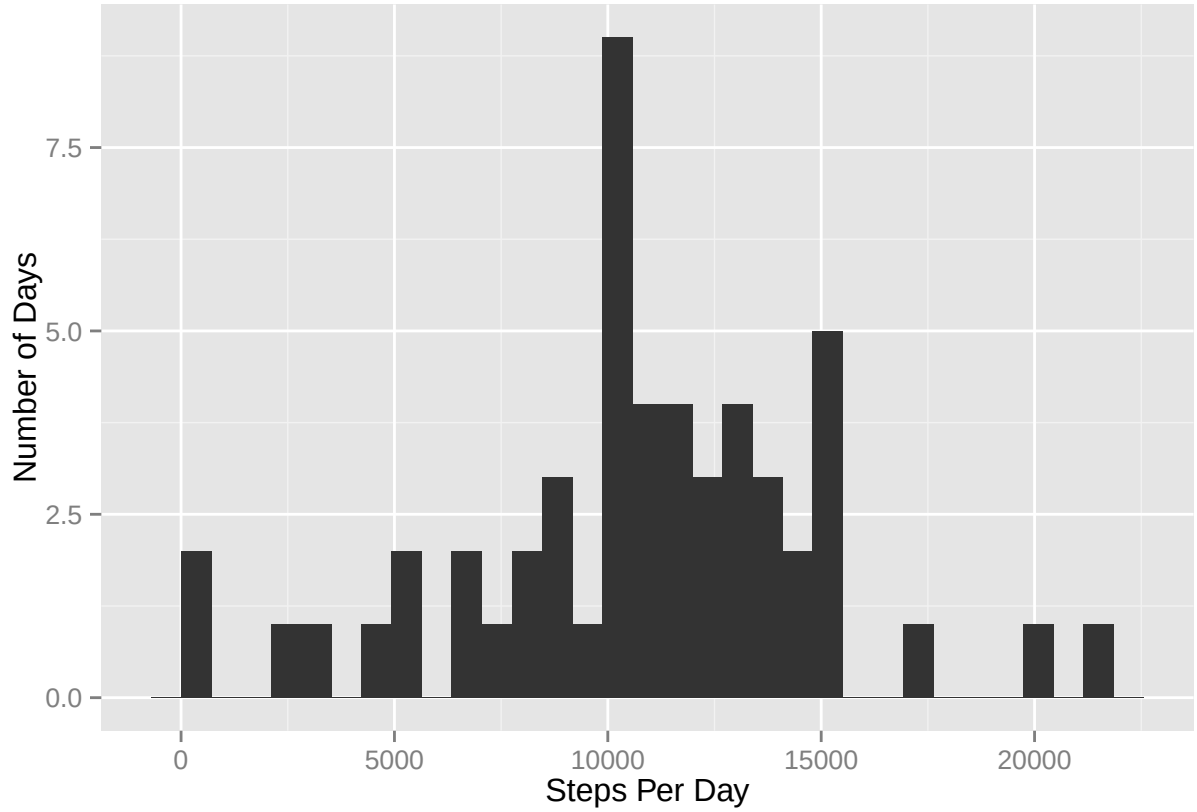
For this part of the assignment, you can ignore the missing values in the dataset.

1.Calculate the total number of steps taken per day

2.If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

```
total_steps_pr_day <- aggregate(steps ~ date, data = personal.activity, sum, na.rm = TRUE)
qplot(total_steps_pr_day$steps, xlab="Steps Per Day", ylab="Number of Days")
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



3.Calculate and report the mean and median of the total number of steps taken per day

```
x <- mean(total_steps_pr_day$steps , na.rm=TRUE)
y <- median(total_steps_pr_day$steps , na.rm=TRUE)
cat("the mean total steps pr day is:",x) + cat(" and the median total steps pr. day is:",y)
```

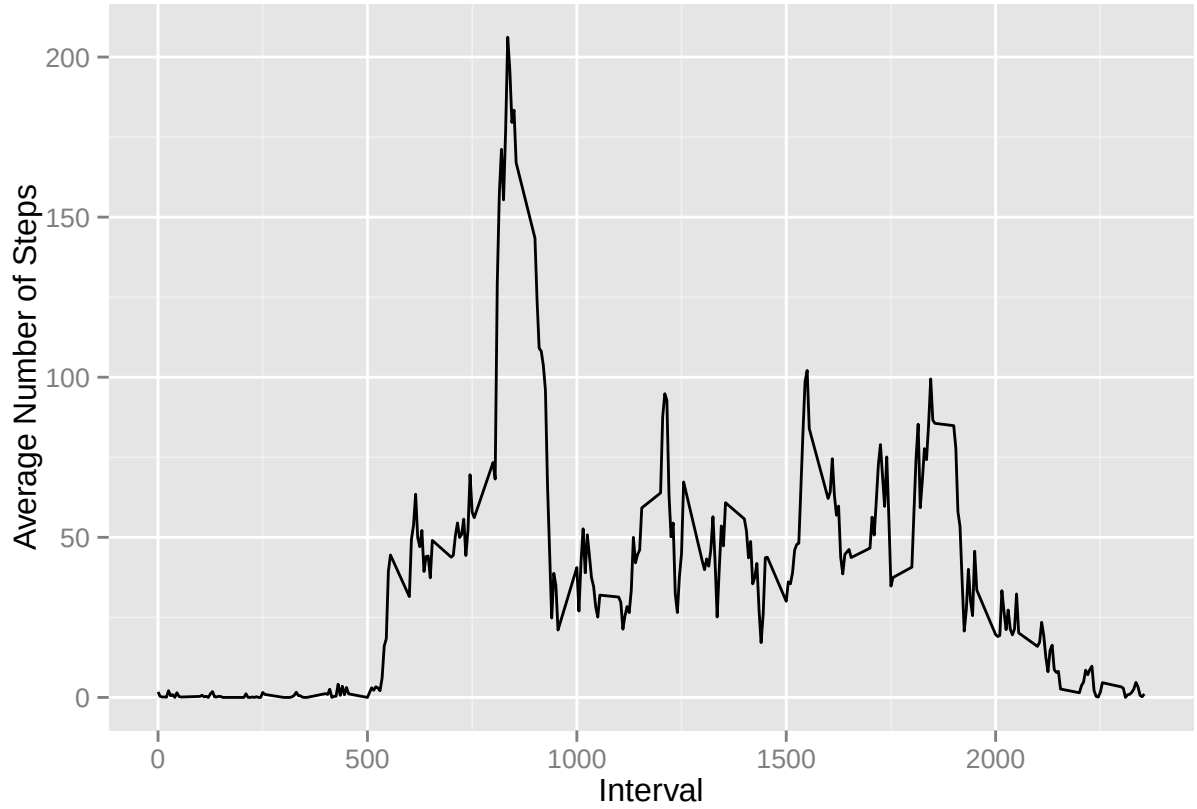
```
## the mean total steps pr day is: 10766.19 and the median total steps pr. day is: 10765
```

```
## numeric(0)
```

B. What is the average daily activity pattern?

1.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)

```
steps_per_interval <- aggregate(x=list(steps=personal.activity$steps), by=list(interval=personal.activi
ggplot(data=steps_per_interval, aes(x=interval, y=steps)) +
    geom_line() + xlab("Interval") + ylab("Average Number of Steps")
```



2.Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
z <- steps_per_interval[which.max(steps_per_interval$steps),]
cat("Time interval", z$interval) + cat(" has the maximum average number of steps,", z$steps)
```

```
## Time interval 835 has the maximum average number of steps, 206.1698
```

```
## numeric(0)
```

C. Imputing missing values

Note that there are a number of days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

1.Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
missing_values <- is.na(personal.activity$steps)
table(missing_values)
```

```
## missing_values
## FALSE  TRUE
## 15264  2304
```

```
cat("Total number of missing values are: ", sum(missing_values))
```

```
## Total number of missing values are:  2304
```

2.Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc.

```
interval2steps <- function(interval) {
    steps_per_interval[steps_per_interval$interval == interval, ]$steps
}
```

The above strategy is to replace NA's with the mean for that 5-minute interval

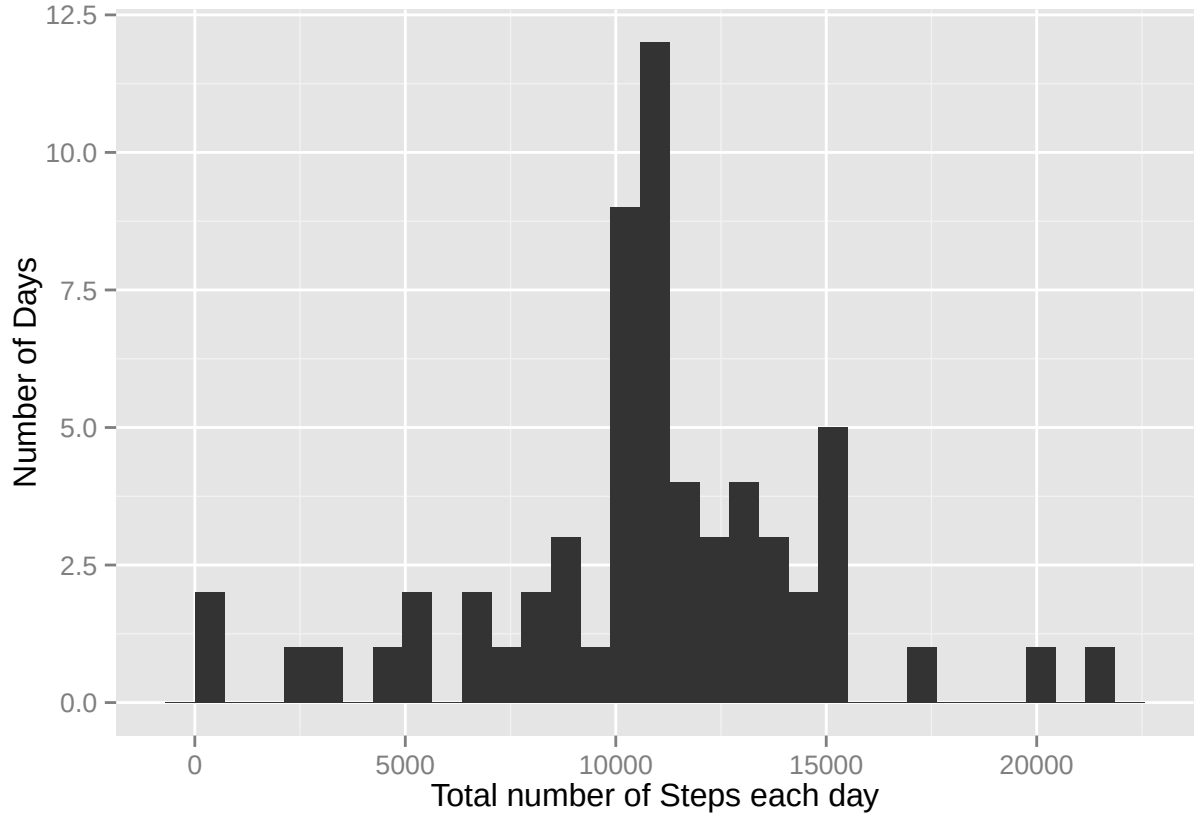
3.Create a new dataset that is equal to the original dataset but with the missing data filled in.

```
personal.activity.filled <- personal.activity
        for (i in 1:nrow(personal.activity.filled)) {
                if (is.na(personal.activity.filled[i, ]$steps)) {
                personal.activity.filled[i, ]$steps <-
                interval2steps(personal.activity.filled[i, ]$interval)
                }
}
```

4.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.

```
imputed_steps_each_day <- tapply(personal.activity.filled$steps, personal.activity.filled$date, FUN=sum
qplot(imputed_steps_each_day, xlab="Total number of Steps each day", ylab="Number of Days")
```

```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



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?

```
mean(imputed_steps_each_day)
```

```
## [1] 10766.19
```

```
median(imputed_steps_each_day)
```

```
## [1] 10766.19
```

```
meandiff <- mean(imputed_steps_each_day, na.rm=TRUE) - mean(total_steps_pr_day$steps, na.rm=TRUE)
mediandiff <- median(imputed_steps_each_day, na.rm=TRUE) - median(total_steps_pr_day$steps, na.rm=TRUE)
cat("The change in mean value is:",meandiff) + cat(" and the change in median value is: ", mediandiff)
```

```
## The change in mean value is: 0 and the change in median value is:  1.188679
```

```
## numeric(0)
```

D. Are there differences in activity patterns between weekdays and weekends?

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

1.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.

```
isWeekdayOrWeekend <- function(date) {
    day <- weekdays(date)
    if (day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))
        return("Weekday")
    else if (day %in% c("Saturday", "Sunday"))
        return("Weekend")
    else
        return("date not classified ...")
}
personal.activity.filled$date <- as.Date(personal.activity.filled$date)
personal.activity.filled$dayofweek <- sapply(personal.activity.filled$date, FUN=isWeekdayOrWeekend)
```

2.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). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

```
steps_by_weekday_or_weekend <- aggregate(steps ~ interval + dayofweek, data=personal.activity.filled, m
ggplot(steps_by_weekday_or_weekend, aes(interval, steps)) + geom_line() + facet_grid(dayofweek ~ .) +
    xlab("Interval") + ylab("Steps")
```