Reproducible Research

GitHub Documents

This is an R Markdown format used for publishing markdown documents to GitHub. When you click the **Knit** button all R code chunks are run and a markdown file (.md) suitable for publishing to GitHub is generated.

Including Code

You can include R code in the document as follows:

Including Plots

Reproducible Research: Assessment 1

Githud repo with RMarkdown Source

Introduction

** It is now possible to collect a large amount of data about personal movement using activity monitoring devices such as a Fitbit, Nike Fuelband, or Jawbone Up. These type of devices are part of the "quantified self" movement a group of enthusiasts who take measurements about themselves regularly to improve their health, to find patterns in their behavior, or because they are tech geeks. But these data remain under-utilized both because the raw data are hard to obtain and there is a lack of statistical methods and software for processing and interpreting the data.

This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.**

Data Source

** The data for this assignment can be downloaded from the course web site:

Dataset:Activity monitoring data[52K] The variables included in this dataset are:

steps: Number of steps taking in a 5-minute interval (missing values are coded as NA)

date: The date on which the measurement data was taken in YYYY-MM-DD format

interval: Identifier for the 5-minute interval in which measurement was taken

The dataset is stored in a comma-separated-value (CSV) file and there are a total of 17,568 observations in this dataset.**

Data Packages Sources ** Loading "data.table" "knitr", "gridExtra", "ggplot2", "plyr" and "dplyr" packages. And set "echo", "results" and "tidy" as global options for knitr.**

```r
library(data.table)
library(knitr)
library(gridExtra)
library(ggplot2)
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following objects are masked from 'package:data.table':
##
##     between, first, last

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
opts_chunk$set(echo = TRUE, results = 'hold', tidy = TRUE)
```

Loading and Preprocessing the Data

** This assignment makes use of data from a personal activity monitoring device. This device collects data at 5 minute intervals through out the day. The data consists of two months of data from an anonymous individual collected during the months of October and November, 2012 and include the number of steps taken in 5 minute intervals each day.**

- Check if the zip file exists
- Unzip the zip file
- Load the data(i.e. read.csv())
- Assign the result to the variable "rdata"

```r
read_data <- function() {
    file_name = "activity.zip"
    Url = "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
    if (!file.exists(file_name)) {
        download.file(Url, destfile = file_name)
    }
    csv_file <- unz(file_name, "activity.csv")
    rdata <- read.csv(csv_file, header = T, colClasses = c("numeric", "character",
        "numeric"))
    rdata$interval <- factor(rdata$interval)
    rdata$date <- as.Date(rdata$date, format = "%Y-%m-%d")
    rdata
}
rdata <- suppressWarnings(read_data())
```

Assigment Questions

- What is mean total number of steps taken per day?
- We will describe with the following 3 variables
- Steps: The numbers of steps for each interval.
- Date: The day, month and year the data was taken.
- Interval: The 5- minute interval of the day.
- Build the Histogram of the total numbers of steps using ggpot plotted with - - bin of intervals of 1000 steps.

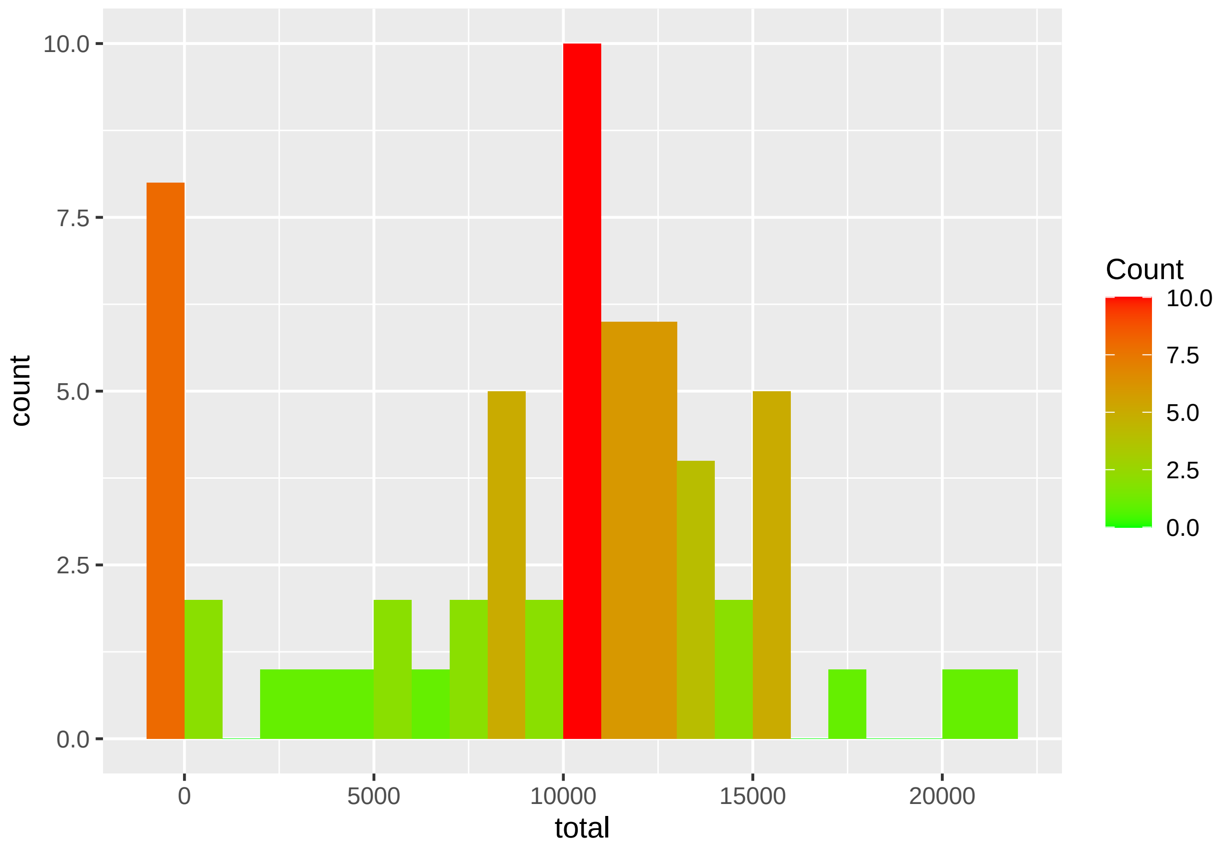
```
day_global <- rdata %>%
    group_by(date) %>%
    summarise(total = sum(steps, na.rm = T))
```

Compute the histogram of the total number of steps each day

```
ggplot(day_global, aes(x = total)) + geom_histogram(aes(fill = ..count..), origin = 0.1,
    binwidth = 1000) + scale_fill_gradient("Count", low = "green", high = "red")
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: The `origin` argument of `stat_bin()` is deprecated as of ggplot2 2.1.0.
## i Please use the `boundary` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Compute Mean and Median without NA's

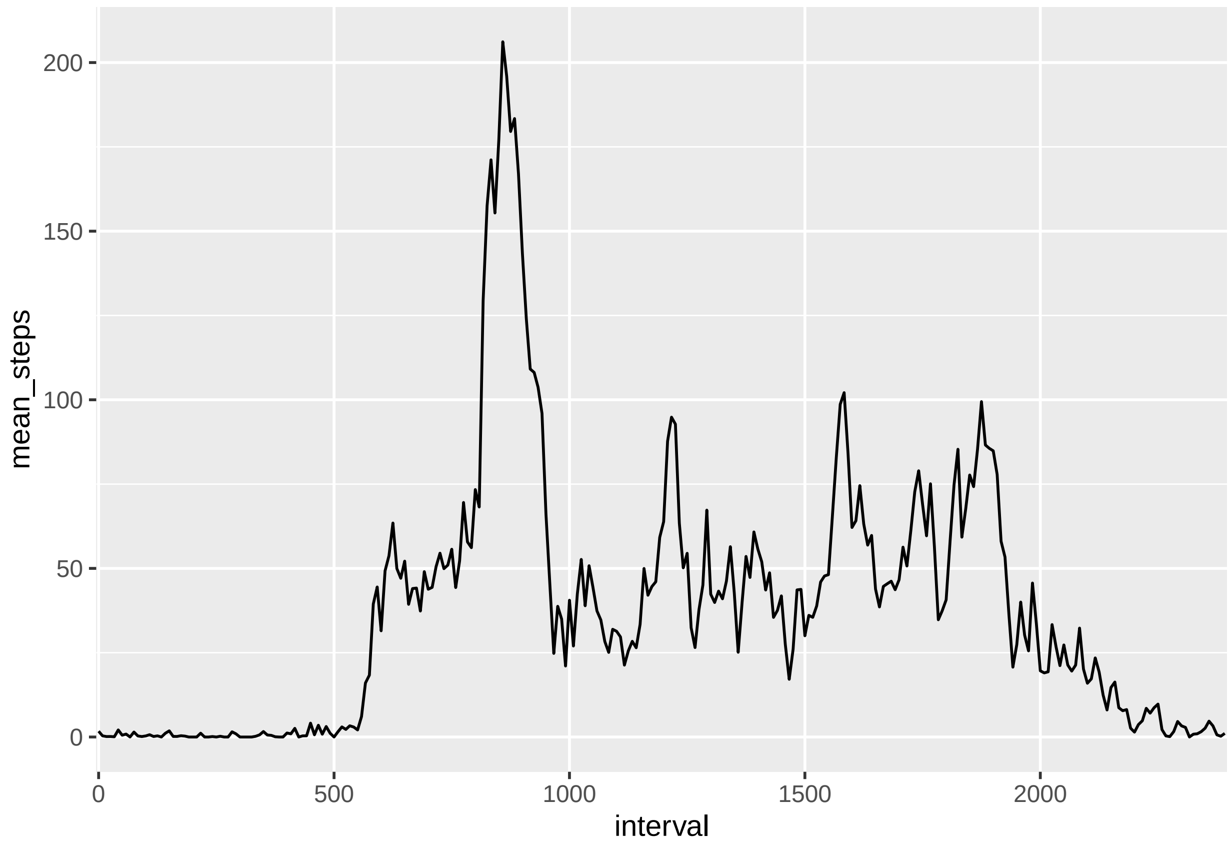
"'{ r average, echo = TRUE} day_global[day_global == 0] <- NA

mean_data <- function(x) { mean_fun <- c(mean = mean, median = median) lapply(mean_fun, function(f) f(x, na.rm = T)) } mean_data(day_global$total)

~~~
### $mean
###
### [1] 10766.19

### $median
###
### [1] 10765

** The Mean is 10766.19 and the Median is 10765 **

### Finding the Average of Daily Activity Pattern.

*** Make the plot with ggplot for the 5-minute interval (x-axis) and the average number of steps during


```r
daily_avg <- rdata %>%
    group_by(interval) %>%
    summarise(mean_steps = mean(steps, na.rm = T))
~~~

ggplot of the average steps taken

```
ggplot(daily_avg, aes(x = interval, y = mean_steps, group = 1)) + geom_line() + scale_x_discrete(breaks
    2500, 500))
```



### The 5-Minutes Interval which contains the Maximum Number of Steps.

```
daily_avg[which.max(daily_avg$mean_steps), ]
```

```
## # A tibble: 1 x 2
##   interval mean_steps
##   <fct>         <dbl>
## 1 835            206.
```

*** Interval means_steps ***

[1] 835 206.1698

The 835th 5-minute interval contains the maximum number of steps.

Imputation of missing values

Compute and report the total of missing data.

```
sum(is.na(rdata))
```

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

[1] 2304

There are 2304 NA's

*** The next process consists of replacing NA's found in an interval with its mean. We do so by first binding a column with the means of steps par 5-minutes interval and then subset the 3 original columns.***

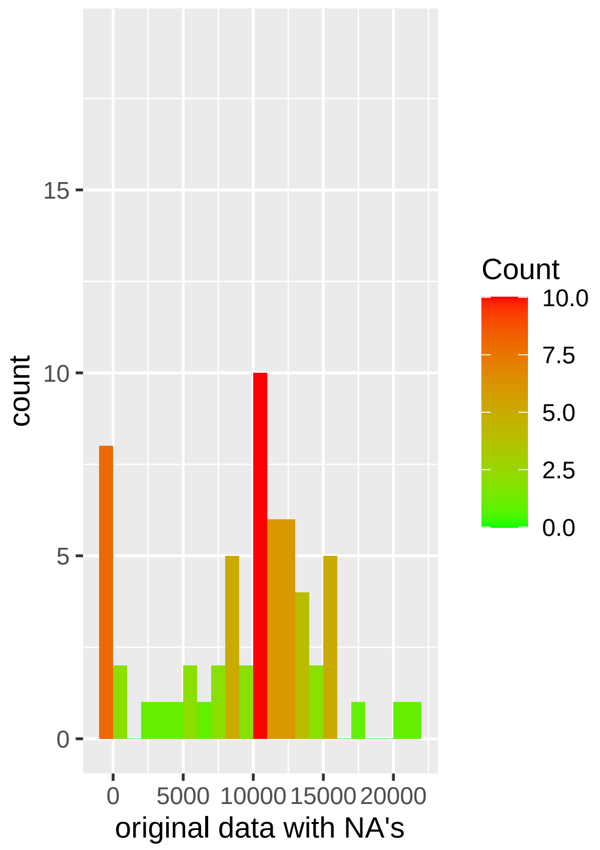
```
rdata_noNA <- cbind(rdata, daily_avg)
rdata_noNA$steps[is.na(rdata_noNA$steps)] <- rdata_noNA$mean_steps[is.na(rdata_noNA$steps)]
rdata_noNA <- rdata_noNA[, 1:3]
```

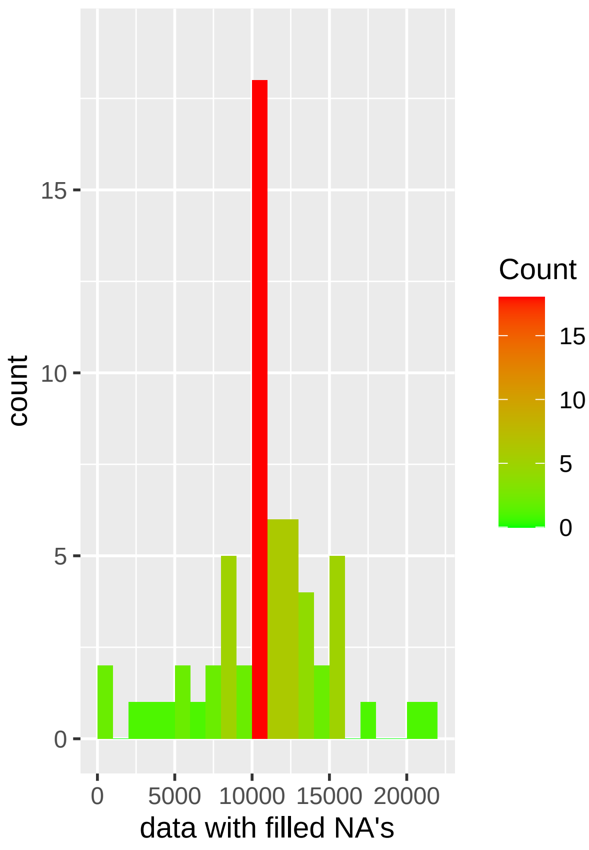
Create a histogram with ggplot of the total number of steps taken each day with the new NA's, and aggregate steps by date using dplyr. Plotted a bin interval 1000 steps.

```
daily_totalnoNA <- rdata_noNA %>%
    group_by(date) %>%
    summarise(total = sum(steps, na.rm = T))
```

Create histogram with original NA's and another histogram with filled NA's, and compare the two.

```
g1 <- ggplot(day_global, aes(x = total)) + geom_histogram(aes(fill = ..count..),
    boundary = 0.5, binwidth = 1000) + ylim(0, 19) + xlab("original data with NA's") +
    scale_fill_gradient("Count", low = "green", high = "red")

g2 <- ggplot(daily_totalnoNA, aes(x = total)) + geom_histogram(aes(fill = ..count..),
    boundary = 0.5, binwidth = 1000) + ylim(0, 19) + xlab("data with filled NA's") +
    scale_fill_gradient("Count", low = "green", high = "red")

grid.arrange(g1, g2, ncol = 2)
```





Compute and report the mean and median of the total number of steps taken daily

```
resume_totalnoNA <- function(x) {
    resum <- c(mean = mean, median = median)
    lapply(resum, function(f) f(x, na.rm = T))
}
resume_totalnoNA(daily_totalnoNA$total)
```

```
## $mean
## [1] 10766.19
##
## $median
## [1] 10766.19
```

$mean

[1] **10766.19**

$median

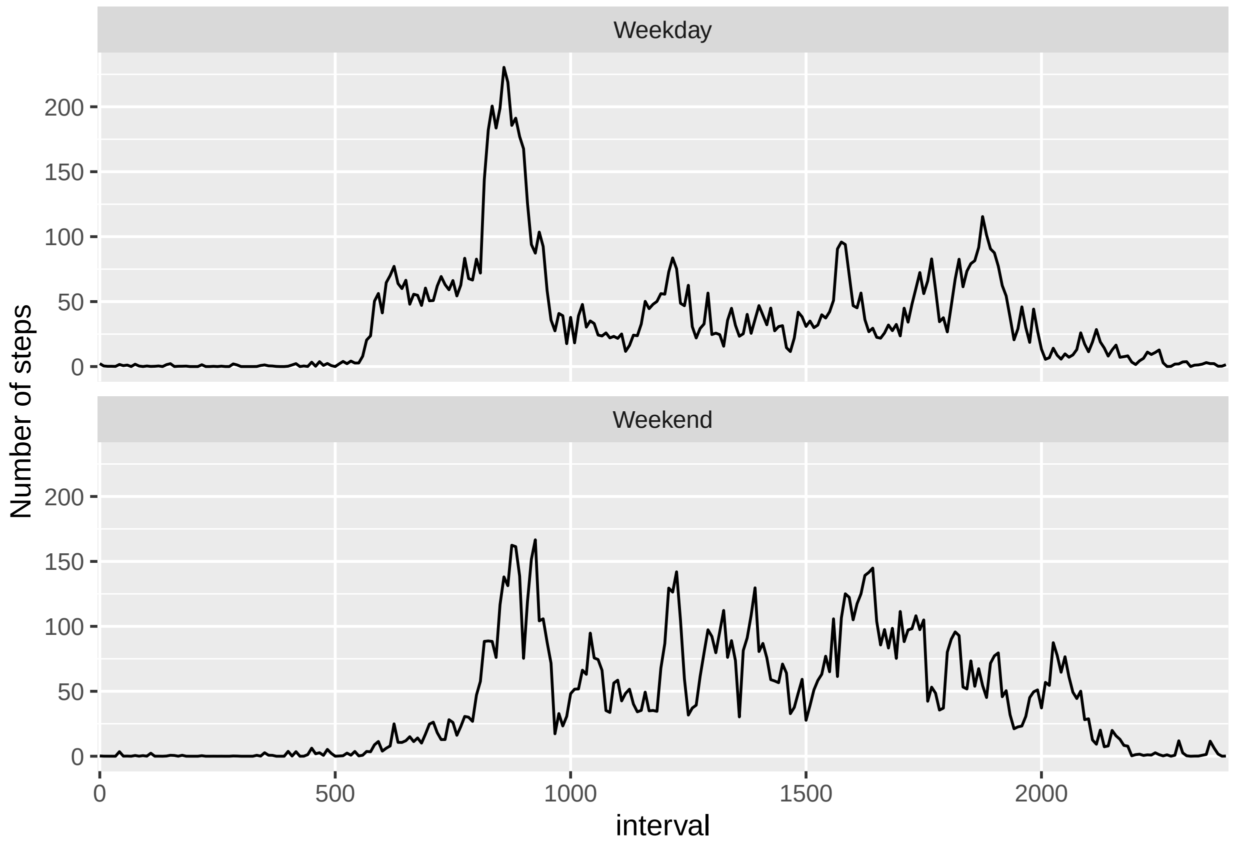
[1] **10766.19**

*** The median and the mean values of populated missing data are all equals to 10766.19.*** - The mean value remains unchanged. - The median value now matches the mean. - Conclusion: The impact of imputation of NA’s doesn’t influence the prédiction. - Comparison of Weekdays and Weekends Activities. - Create a new factor variable with two levels: 1. weekday 2. weekends -Subset by each factor. -Aggregate the mean. -Bind both sets.

```r
rdata_days <- rdata_noNA %>%
    mutate(type_of_day = as.factor(format(date, "%a")))

rdata_days$type_of_day <- revalue(rdata_days$type_of_day, c(Mon = "Weekday", Tue = "Weekday",
    Wed = "Weekday", Thu = "Weekday", Fri = "Weekday", Sat = "Weekend", Sun = "Weekend"))
rdata_weekday <- subset(rdata_days, type_of_day == "Weekday")
rdata_weekend <- subset(rdata_days, type_of_day == "Weekend")

weekday_means <- aggregate(rdata_weekday$steps, list(rdata_weekday$interval, rdata_weekday$type_of_day)
    mean)
weekend_means <- aggregate(rdata_weekend$steps, list(rdata_weekend$interval, rdata_weekend$type_of_day)
    mean)

colnames(weekday_means) <- c("interval", "type_of_day", "avg_steps")
colnames(weekend_means) <- c("interval", "type_of_day", "avg_steps")

weekday_data <- rbind(weekday_means, weekend_means)
```

plot the 5- minute interval(x-axis) and the average number of steps taken by weekdays and weekends(y-axis).

```r
ggplot(weekday_data, aes(x = interval, y = avg_steps, group = 1)) + geom_line() +
    scale_x_discrete(breaks = seq(0, 2500, 500)) + facet_wrap(~type_of_day, nrow = 2) +
    ylab("Number of steps")
```



### Conclusion *** The weekday has the greatest peak from all the steps due to the fact that weekdays activities may follow a certain sport routine denoted by the peak over 200 steps. On the other hands, 'weekends' days are more related to routine weekend activities with more peaks on the average distributed compare to weekdays.***