Reproducible Research

Load some necessary R packages

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.2.5
```

```
##
## 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(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.2.5
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.2.5
```

Loading and preprocessing the data

```
data <- read.csv("activity.csv", header = TRUE, sep = ',', colClasses = c("numeric", "character","integer"))

data$date <- ymd(data$date)


str(data)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Date, format: "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
```

What is mean total number of steps taken per day?

For this part of the assignment the missing values can be ignored.

1.Calculate the total number of steps taken per day.

2.Make a histogram of the total number of steps taken each day.

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

Methods and Results

1. Calculate the total number of steps per day using dplyr and group by date:

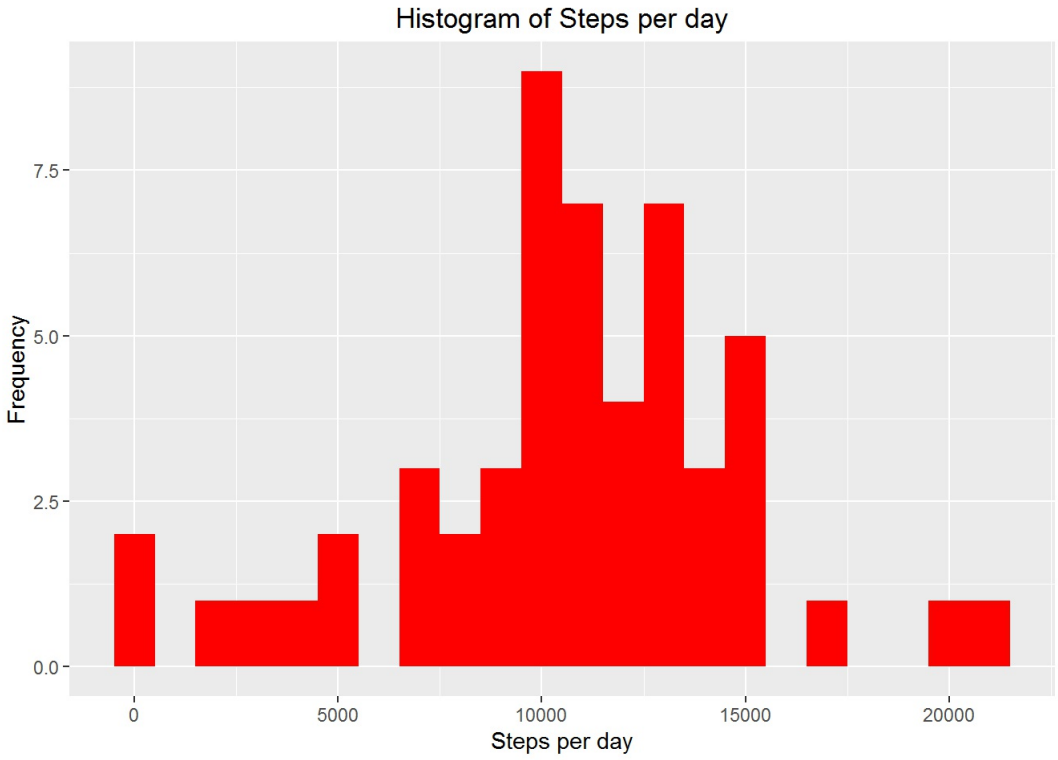
```
steps <- data %>%
filter(!is.na(steps)) %>%
group_by(date) %>%
summarize(steps = sum(steps)) %>%

print
```

```
## Source: local data frame [53 x 2]
##
##          date steps
##        (date) (dbl)
## 1  2012-10-02   126
## 2  2012-10-03 11352
## 3  2012-10-04 12116
## 4  2012-10-05 13294
## 5  2012-10-06 15420
## 6  2012-10-07 11015
## 7  2012-10-09 12811
## 8  2012-10-10  9900
## 9  2012-10-11 10304
## 10 2012-10-12 17382
## ..        ...   ...
```

2.Make histogram using ggplot

```
ggplot(steps, aes(x = steps)) +
 geom_histogram(fill = "red", binwidth = 1000) +
 labs(title = "Histogram of Steps per day", x = "Steps per day", y = "Frequency")
```



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

```
meanofsteps <- mean(steps$steps, na.rm = TRUE)

medianofsteps <- median(steps$steps, na.rm = TRUE)

meanofsteps
```

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

```
medianofsteps
```

```
## [1] 10765
```

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)

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

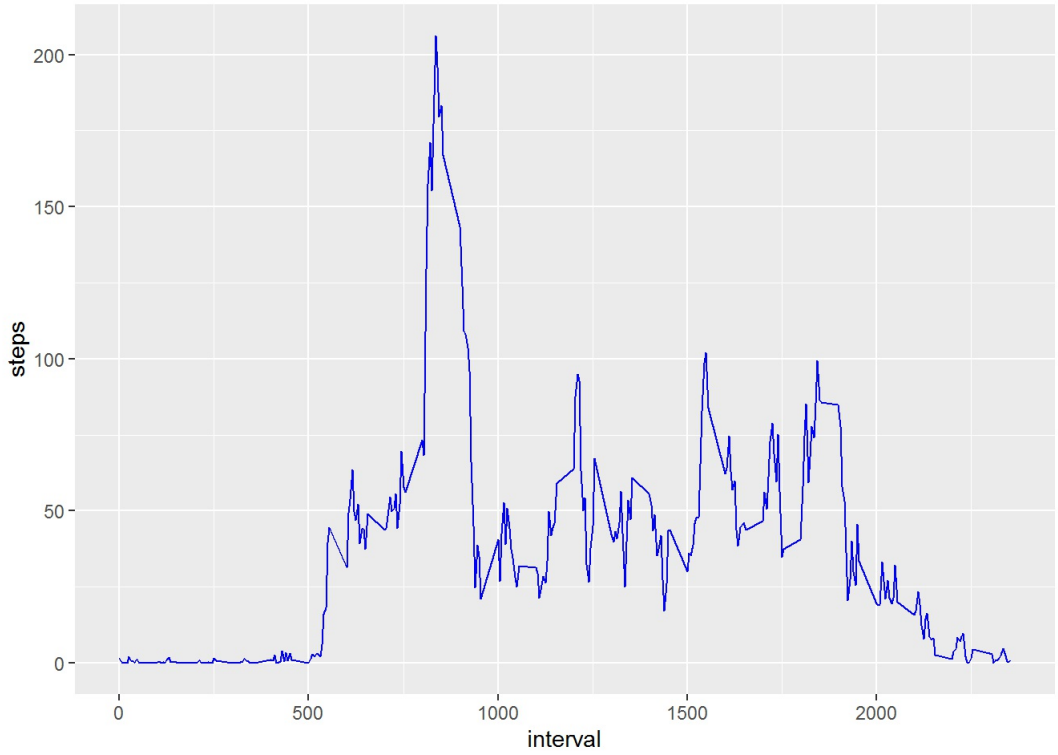
Methods and the Results

1.Calculate the average number of steps taken in each 5-minute interval per day using dplyr and group by interval:

```
interval <- data %>%
  filter(!is.na(steps)) %>%
  group_by(interval) %>%
  summarize(steps = mean(steps))
```

Making the time series of the 5-minute interval and average steps taken using the ggplot:

```
ggplot(interval, aes(x=interval, y=steps)) +
  geom_line(color = "Blue")
```



2.To find out the maximum steps, on average, across all the days use which.max()

```
interval[which.max(interval$steps),]
```

```
## Source: local data frame [1 x 2]
##
##   interval    steps
##      (int)    (dbl)
## 1      835 206.1698
```

Thus, the interval 835 on average across all the days has the maximum steps.

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)

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.

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

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. 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?

Methods followed and results

1.Find the number of missing values

```
sum(is.na(data$steps))
```

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

Thus, the total missing values are 2304

2.Fill in a missing NA with the average number of steps in the same 5-min interval.

3.Create a new dataset as the original

```
newdata<- data
nas <- is.na(newdata$steps)
avg_interval <- tapply(newdata$steps, newdata$interval, mean, na.rm=TRUE, simplify=TRUE)
newdata$steps[nas] <- avg_interval[as.character(newdata$interval[nas])]
```

Checking if there are missing values

```
sum(is.na(newdata$steps))
```

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

There are no more missing values.

4.To make a historgram ,first Calculate the number of steps taken in each 5-minute interval per day using dplyr and group by interval.

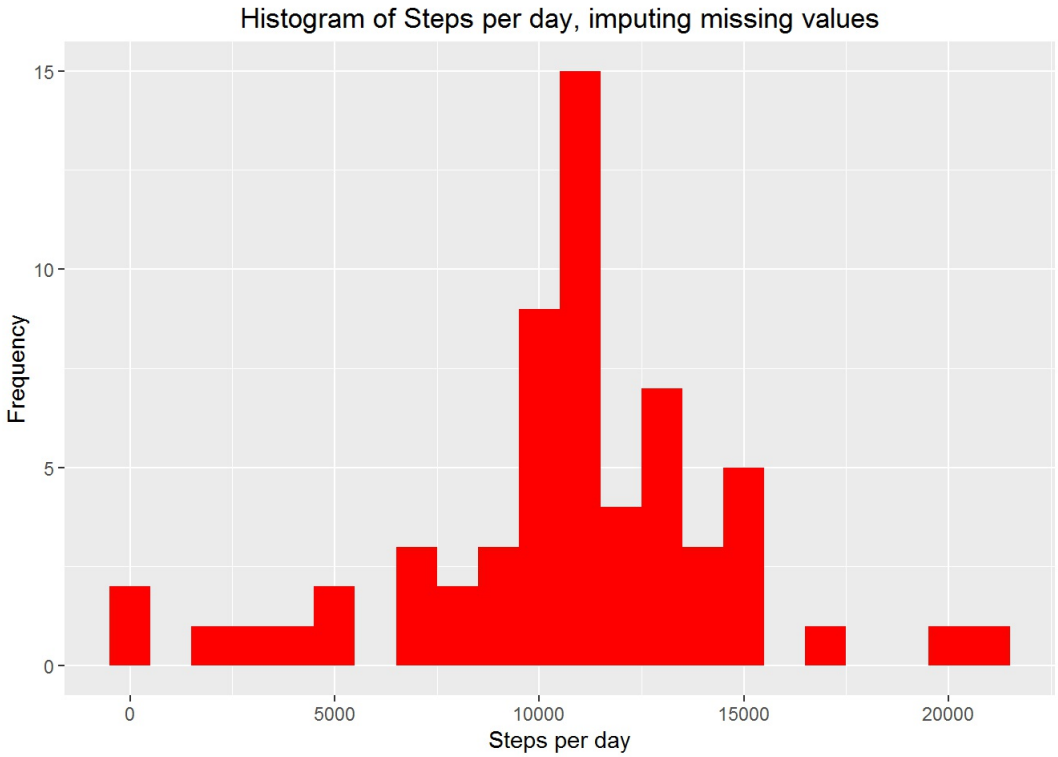
```
full_steps <- newdata %>%

  filter(!is.na(steps)) %>%
  group_by(date) %>%
  summarize(steps = sum(steps)) %>%
  print
```

```
## Source: local data frame [61 x 2]
##
##          date    steps
##        (date)    (dbl)
## 1  2012-10-01 10766.19
## 2  2012-10-02   126.00
## 3  2012-10-03 11352.00
## 4  2012-10-04 12116.00
## 5  2012-10-05 13294.00
## 6  2012-10-06 15420.00
## 7  2012-10-07 11015.00
## 8  2012-10-08 10766.19
## 9  2012-10-09 12811.00
## 10 2012-10-10  9900.00
## ..        ...      ...
```

Make a histogram using ggplot

```
ggplot(full_steps, aes(x = steps)) +
  geom_histogram(fill = "Red", binwidth = 1000) +
  labs(title = "Histogram of Steps per day, imputing missing values", x = "Steps per day", y = "Frequency")
```



Calculate the mean and median steps with the filled in values:

```
meanoffull_steps <- mean(full_steps$steps, na.rm = TRUE)

medianoffull_steps <- median(full_steps$steps, na.rm = TRUE)

meanoffull_steps
```

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

```
medianoffull_steps
```

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

After the missing values are imputed the mean and the median are equal

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.

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.

Methods and Result

create a new column,i.e weektype, and apply whether the day is weekend or weekday:

```
library(dplyr)
```

```
newdata<- mutate(newdata, weektype = ifelse(weekdays(newdata$date) == "Saturday" | weekdays(newdata$date) == "S
unday", "weekend", "weekday"))

newdata$weektype <- as.factor(newdata$weektype)

head(newdata)
```

```
##       steps       date interval weektype
## 1 1.7169811 2012-10-01        0  weekday
## 2 0.3396226 2012-10-01        5  weekday
## 3 0.1320755 2012-10-01       10  weekday
## 4 0.1509434 2012-10-01       15  weekday
## 5 0.0754717 2012-10-01       20  weekday
## 6 2.0943396 2012-10-01       25  weekday
```

2.Calculate the average steps in the 5-minute interval and use ggplot for making the time series of the 5-minute interval for weekday and weekend, and compare the average steps:

```
fullinterval <- newdata%>%
  group_by(interval, weektype) %>%
  summarise(steps = mean(steps))

graph <- ggplot(fullinterval, aes(x=interval, y=steps, color = weektype)) +
  geom_line() +
  facet_wrap(~weektype, ncol = 1, nrow=2)

print(graph)
```