Reproducible Research

Week 2: Assignment 1

==

```
# Load the data
activity = read.csv("/Users/Sam/Downloads/activity.csv", header=TRUE)
# Remove NAs
data = na.omit(activity)
# Convert date variable from factor to date
data$date = as.Date(data$date)
```

What is mean total number of steps taken per day?

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

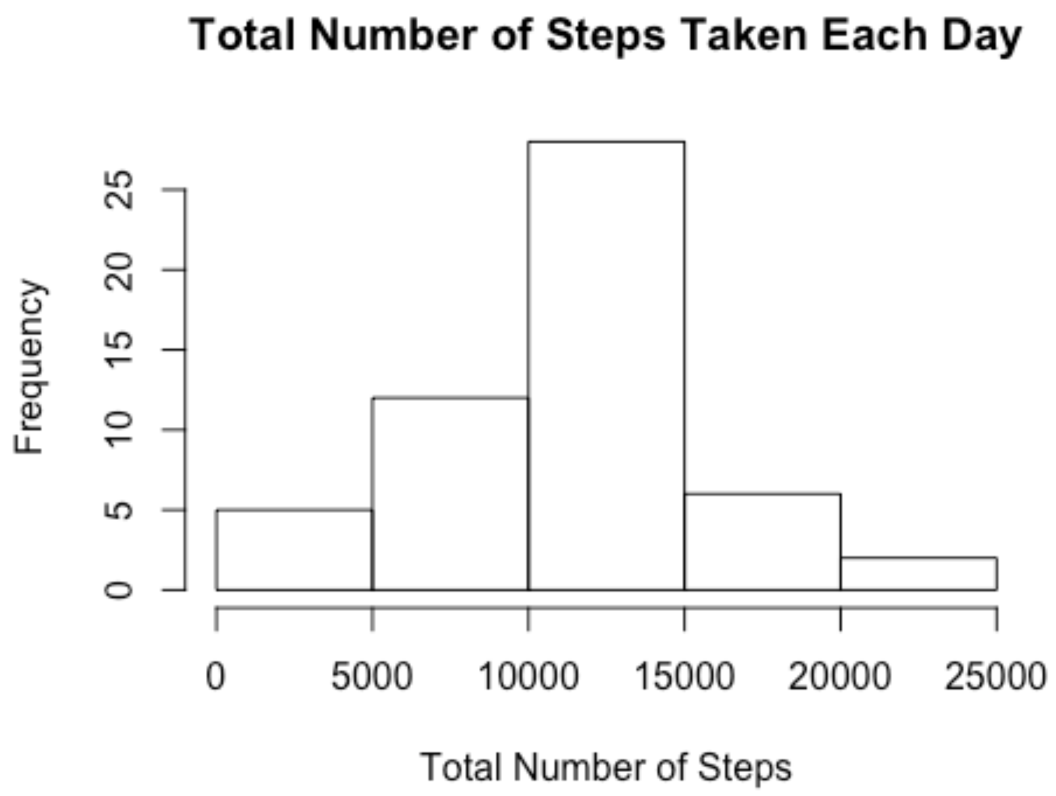
```
# Group data by date and summarize the sum of steps: option 1
sum_steps_per_day1 = aggregate(activity$steps,
by=list(Category=activity$date), FUN=sum, na.rm=TRUE)
head(sum_steps_per_day1)

##     Category     x
## 1 2012-10-01     0
## 2 2012-10-02   126
## 3 2012-10-03 11352
## 4 2012-10-04 12116
## 5 2012-10-05 13294
## 6 2012-10-06 15420

# Group data by date and summarize the sum of steps: option 2
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

activity %>%
  group_by(date) %>%
  summarize(Frequency = sum(steps))
```

```
## # A tibble: 61 × 2
##          date Frequency
##        <fctr>     <int>
## 1  2012-10-01        NA
## 2  2012-10-02       126
## 3  2012-10-03     11352
## 4  2012-10-04     12116
## 5  2012-10-05     13294
## 6  2012-10-06     15420
## 7  2012-10-07     11015
## 8  2012-10-08        NA
## 9  2012-10-09     12811
## 10 2012-10-10      9900
## # ... with 51 more rows
```

```
sum_steps_per_day2 = data %>%
  group_by(date) %>%
  summarize(TotalSteps=sum(steps))
```

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.

```
# Histogram of total number of steps taken per day
hist(sum_steps_per_day2$TotalSteps,
  xlab="Total Number of Steps",
  ylab="Frequency",
  main="Total Number of Steps Taken Each Day")
```



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

The mean total number of steps taken per day is 10766.19. The median total number of steps taken per day is 10765.

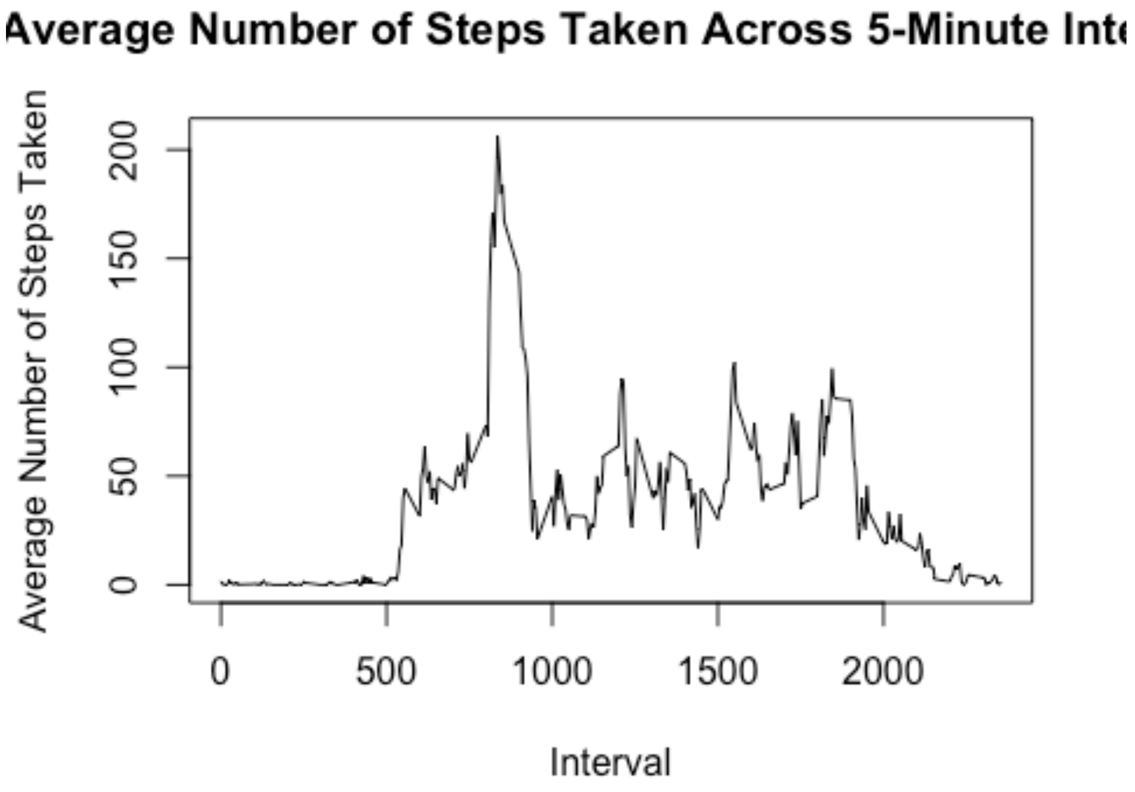
```
mean(sum_steps_per_day2$TotalSteps)

## [1] 10766.19

median(sum_steps_per_day2$TotalSteps)

## [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).

```
# Group data by 5-minute interval and summarize the average number of
steps taken across all days
five_min_average = data %>%
  group_by(interval) %>%
  summarize(average_steps=mean(steps))
```

```r
plot(five_min_average$interval, five_min_average$average_steps,
  type="l",
  xlab="Interval",
  ylab="Average Number of Steps Taken",
  main="Average Number of Steps Taken Across 5-Minute Interval")
```



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

Interval 835 contains the maximum number of steps.

```r
max_steps =
five_min_average$interval[which.max(five_min_average$average_steps)]
max_steps

## [1] 835
```

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).

There are 2,304 NAs.

```
# Calculate the number of missing values
missing = sum(is.na(activity$steps))
missing

## [1] 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.

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

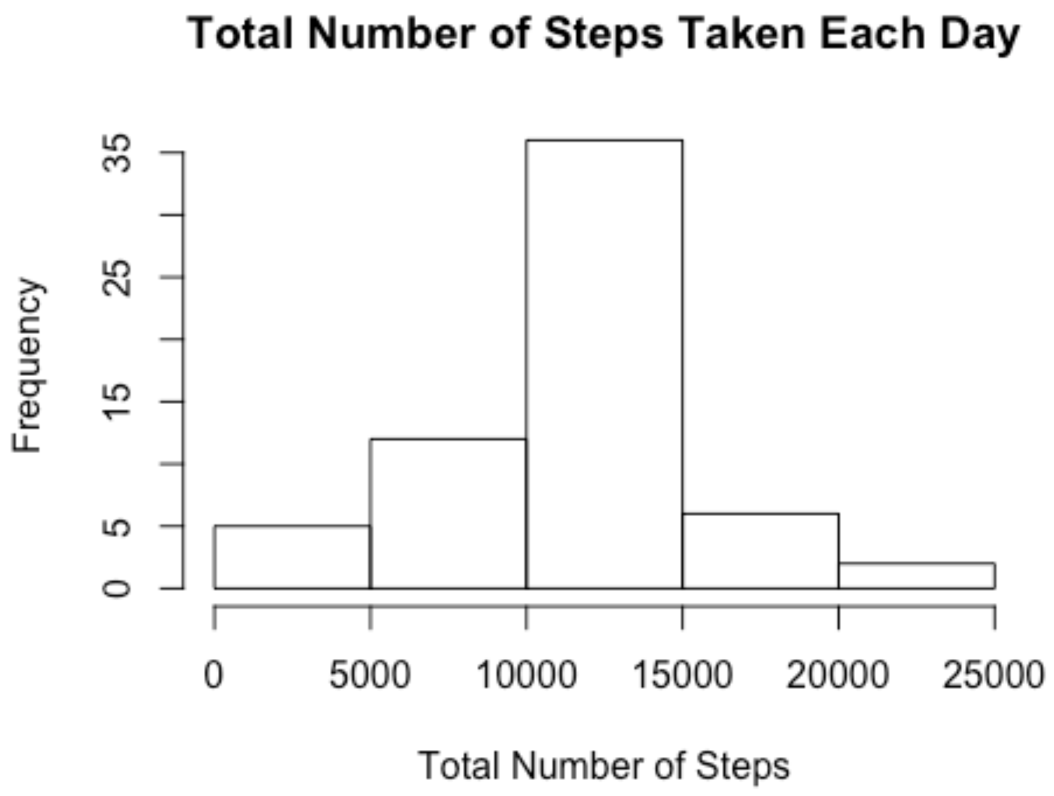
```
# Create a filled in dataset by assigning the average value for that
time interval if a missing value is found.
imputed_data = activity
for (i in 1:nrow(imputed_data)) {
  if (is.na(imputed_data$steps[i])) {
    # Find the index value for when the interval matches the average
    index_value = which(imputed_data$interval[i] ==
five_min_average$interval)
    # Assign the average value to replace the missing value
    imputed_data$steps[i] =
five_min_average[index_value,]$average_steps
    }
}

# Convert date variable from factor to date
imputed_data$date = as.Date(imputed_data$date)
```

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?

The mean total number of steps taken per day with the imputed data is 10766.19, which is the same mean from the original dataset. The median is

10766.19, which is the same as the mean. This makes sense because I filled in the 2,304 missing values with the mean.

```r
# Group data by date and summarize the sum of steps
sum_steps_per_day3 = imputed_data %>%
  group_by(date) %>%
  summarize(TotalSteps=sum(steps))

# Histogram of total number of steps taken per day
hist(sum_steps_per_day3$TotalSteps,
  xlab="Total Number of Steps",
  ylab="Frequency",
  main="Total Number of Steps Taken Each Day")
```



```r
mean(sum_steps_per_day3$TotalSteps)
```

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

```r
median(sum_steps_per_day3$TotalSteps)
```

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

Are there differences in activity patterns between weekdays and weekends?

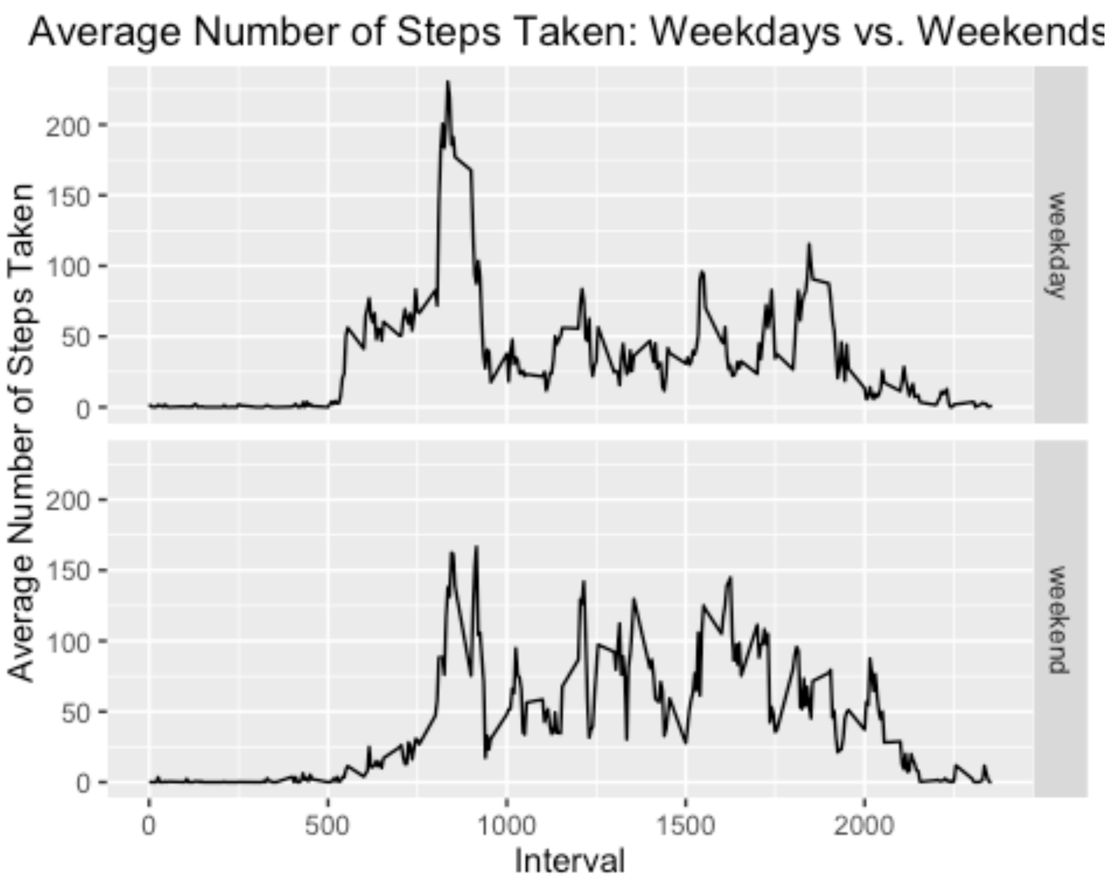
For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

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.

```
# Create weekday variable
imputed_data$day = weekdays(imputed_data$date)
# Set all days as weekdays
imputed_data$daytype = "weekday"
# Set Saturday and Sunday as weekends
imputed_data$daytype[imputed_data$day %in% c("Saturday", "Sunday")] =
"weekend"
head(imputed_data)

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

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.

```
# Group data by 5-minute interval and summarize the average number of
steps taken across all days
five_min_average2 = imputed_data %>%
  group_by(daytype, interval) %>%
  summarize(average_steps=mean(steps))

library(ggplot2)

qplot(interval, average_steps, data=five_min_average2,
  geom="line",
  xlab="Interval",
  ylab="Average Number of Steps Taken",
  main="Average Number of Steps Taken: Weekdays vs. Weekends",
  facets=daytype ~ .)
```


Average Number of Steps Taken: Weekdays vs. Weekends
Average Number of Steps Taken
Interval
weekday
weekend
0
50
100
150
200
0
500
1000
1500
2000