# Activity Analysis

*Luiz*

*April 10, 2016*

First you need set the directory that have file activity.csv and after you can execute this program

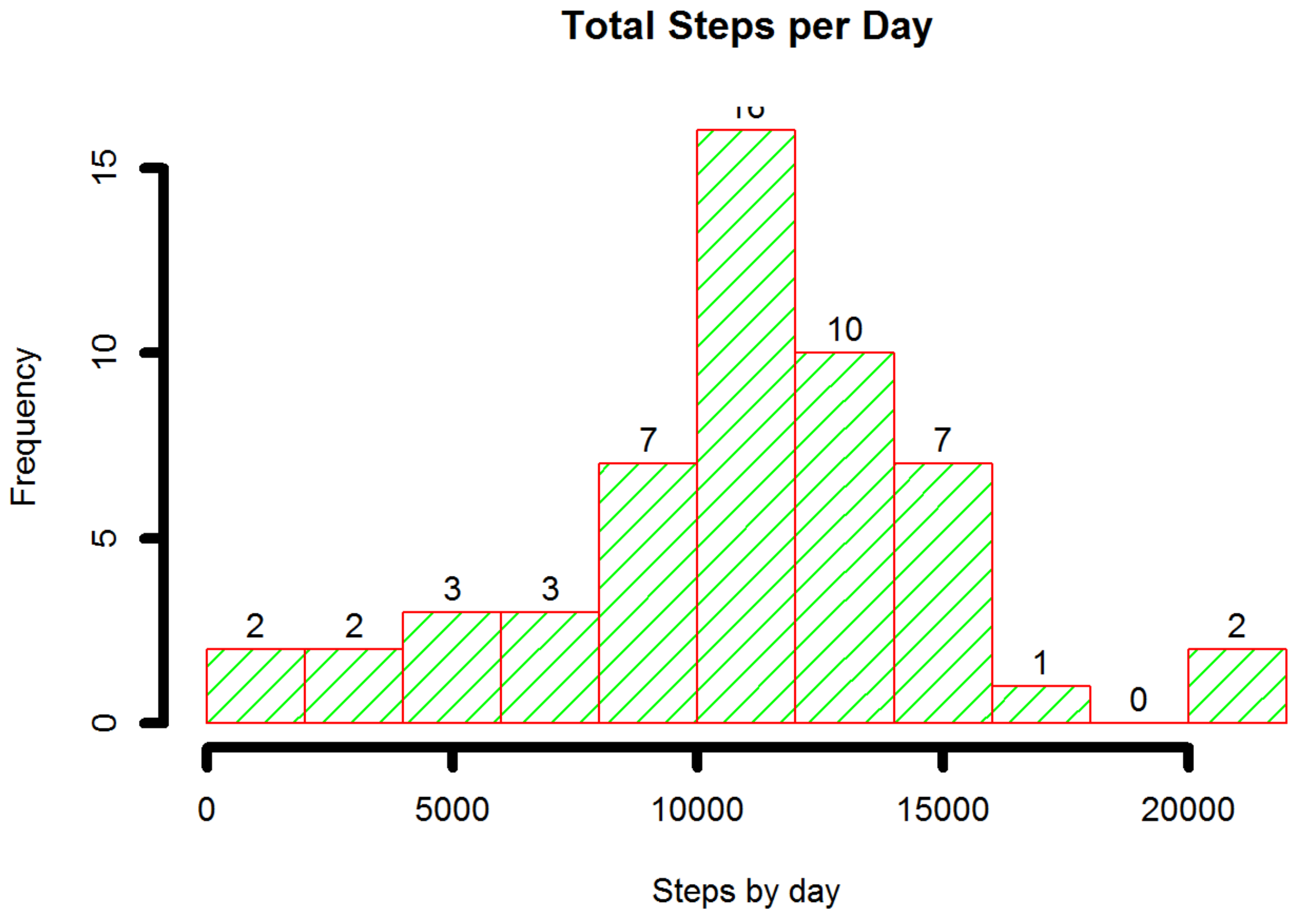
```
unzip("activity.zip")

data_act <- read.csv("activity.csv", header = TRUE)
```

## Calculate the total number of steps taken per day

Total Steps per Day

```
par(mfrow=c(1,1))
check_act <- aggregate(steps ~ date , data = data_act, sum, na.rm = TRUE)
hist(check_act$steps, breaks = 10, main="Total Steps per Day", xlab = "Steps by day",
     col = "green", density = 10, border = 10, lwd = 5, labels = TRUE)
```



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

```
mean(check_act$steps, na.rm=TRUE)
```

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

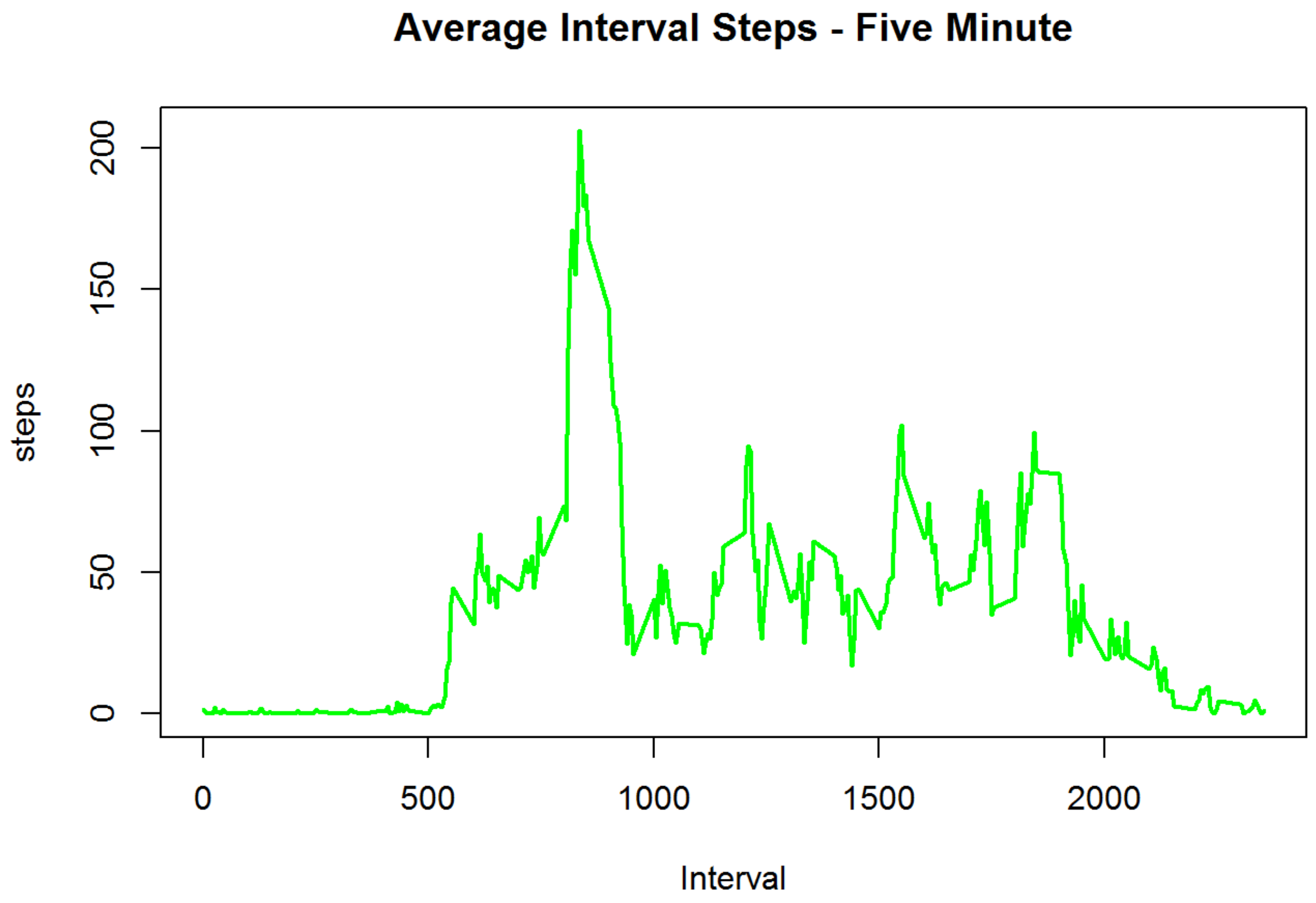
```
median(check_act$steps, na.rm=TRUE)
```

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

# 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)

```
check_interval <- aggregate(steps ~ interval , data = data_act, mean, na.rm = TRUE)
plot(check_interval$interval, check_interval$steps, type="l", col = "green", lwd = 2,
     main="Average Interval Steps - Five Minute", xlab="Interval", ylab="steps")
```



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

```
max(check_interval$steps)
```

```
## [1] 206.1698
```

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

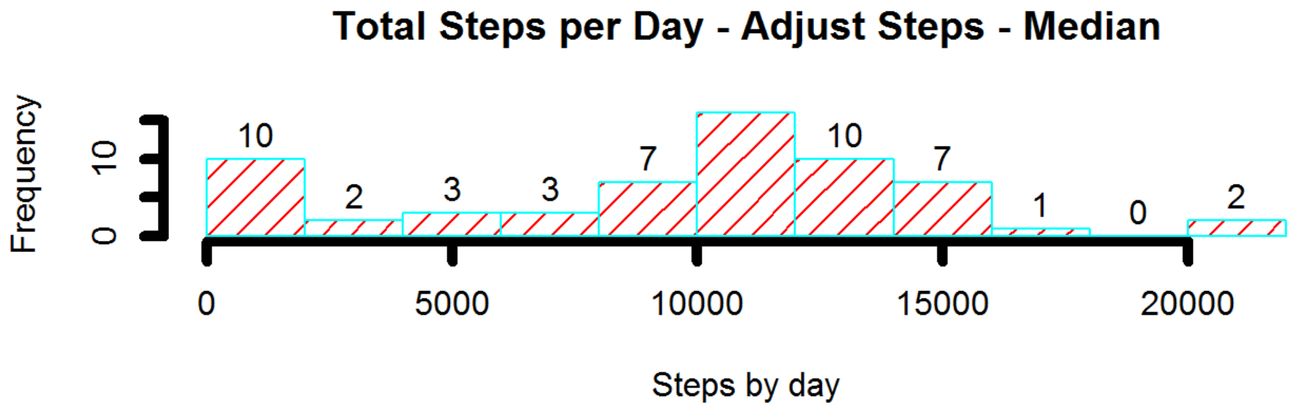
```
sum(is.na(check_interval$steps))
```

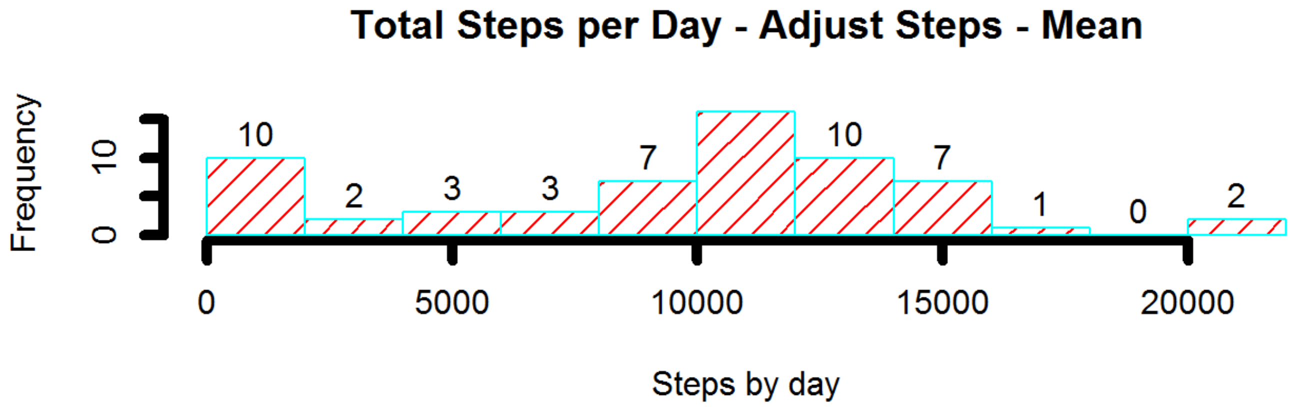
```
## [1] 0
```

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

```
data_act_full <- data_act
data_act_full$steps[is.na(data_act_full$steps)] <- median(data_act$steps, na.rm=TRUE)
check_act_full <- aggregate(steps ~ date, data=data_act_full, sum, na.rm=TRUE)
par(mfrow=c(2,1))
hist(check_act_full$steps, breaks = 10, main="Total Steps per Day - Adjust Steps - Media
n", xlab = "Steps by day",
     col = "red", density = 10, border = 5, lwd = 5, labels = TRUE)
data_act_full$steps[is.na(data_act_full$steps)] <- mean(data_act$steps, na.rm=TRUE)
check_act_full <- aggregate(steps ~ date, data=data_act_full, sum, na.rm=TRUE)
hist(check_act_full$steps, breaks = 10, main="Total Steps per Day - Adjust Steps - Mean",
xlab = "Steps by day",
     col = "red", density = 10, border = 5, lwd = 5, labels = TRUE)
```





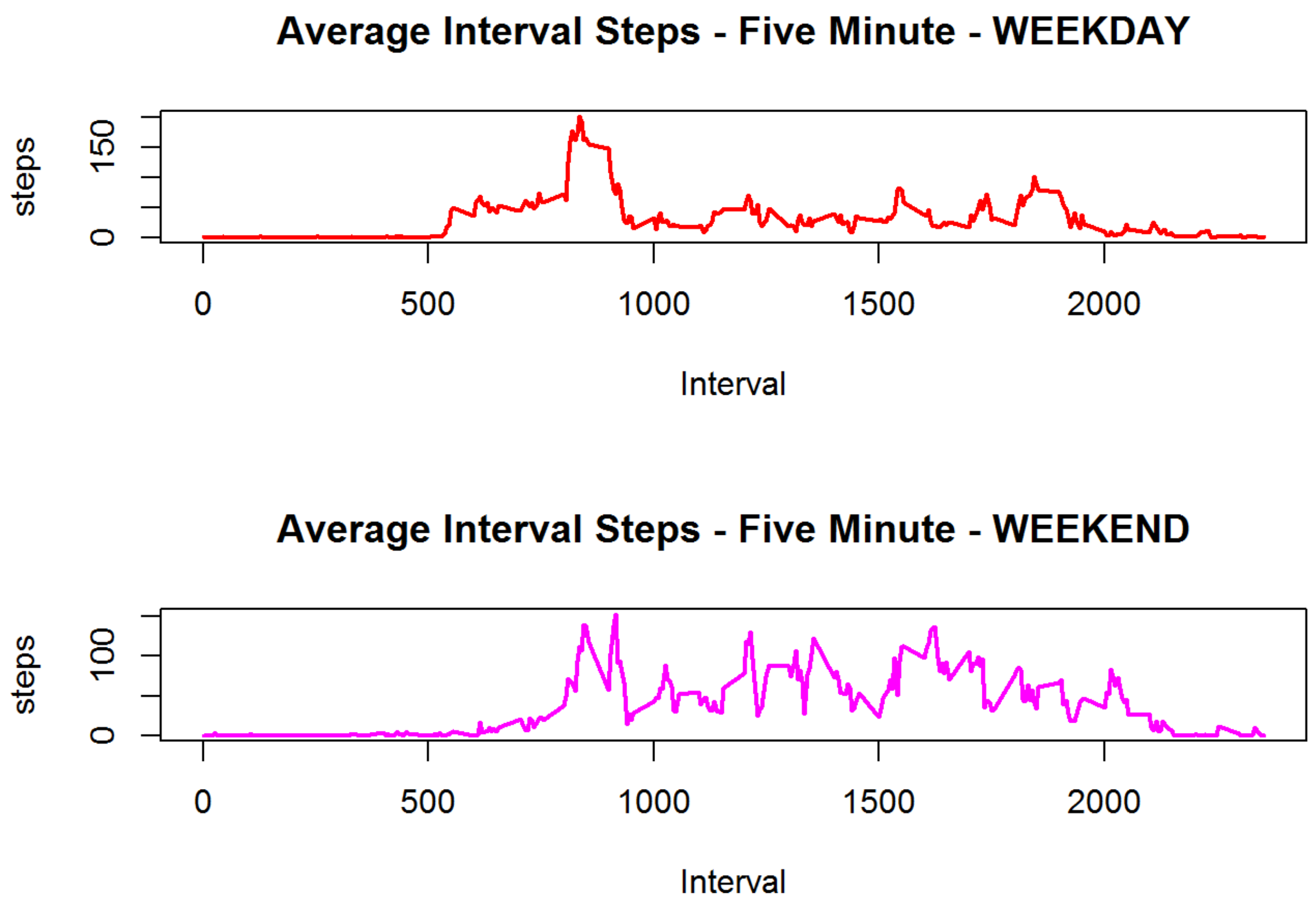
###Create a new factor variable in the dataset with two levels - “weekday” and “weekend” indicating ###whether a given date is a weekday or weekend day.

```
data_act_week <- data_act_full
data_act_week$date <- as.Date(data_act_week$date)
data_act_week$week <- weekdays(data_act_week$date)
data_act_week$week <- gsub("Monday",    "Weekday", data_act_week$week )
data_act_week$week <- gsub("Tuesday",   "Weekday", data_act_week$week )
data_act_week$week <- gsub("Wednesday", "Weekday", data_act_week$week )
data_act_week$week <- gsub("Thursday",  "Weekday", data_act_week$week )
data_act_week$week <- gsub("Friday",    "Weekday", data_act_week$week )
data_act_week$week <- gsub("Saturday",  "Weekend", data_act_week$week )
data_act_week$week <- gsub("Sunday",    "Weekend", data_act_week$week )
```

## 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)

```
par(mfrow=c(2,1))
data_week = subset(data_act_week, week == "Weekday")
print_week <- aggregate(steps ~ interval , data = data_week, mean, na.rm = TRUE)
plot(print_week$interval, print_week$steps, type="l", col = 26, lwd = 2,
     main="Average Interval Steps - Five Minute - WEEKDAY", xlab="Interval", ylab="step
s")
data_week = subset(data_act_week, week == "Weekend")
print_week <- aggregate(steps ~ interval , data = data_week, mean, na.rm = TRUE)
plot(print_week$interval, print_week$steps, type="l", col = 30, lwd = 2,
     main="Average Interval Steps - Five Minute - WEEKEND", xlab="Interval", ylab="step
s")
```


Average Interval Steps - Five Minute - WEEKDAY
steps
Interval
Average Interval Steps - Five Minute - WEEKEND
steps
Interval