

Markdown

Me

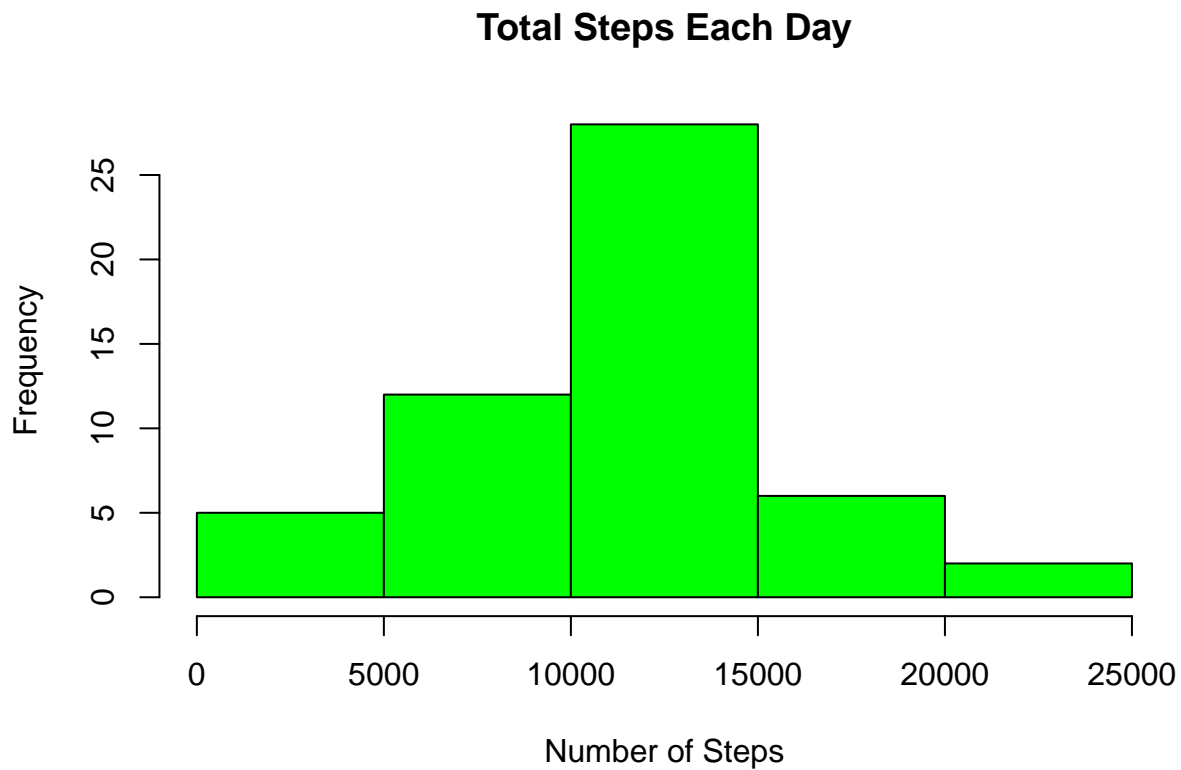
31/03/2022

```
data <- read.csv("activity.csv")
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
```

Histogram of the total number of steps taken each day

```
steps_by_day <- aggregate(steps ~ date, data, sum)
hist(steps_by_day$steps, main = paste("Total Steps Each Day"), col="green", xlab="Number of Steps")
```



Mean and median number of steps taken each day

```
steps_by_day <- aggregate(steps ~ date, data, sum)
rmean <- mean(steps_by_day$steps)
rmedian <- median(steps_by_day$steps)
sprintf("MEAN of steps taken each day = %.3f", rmean)
```

```
## [1] "MEAN of steps taken each day = 10766.189"
```

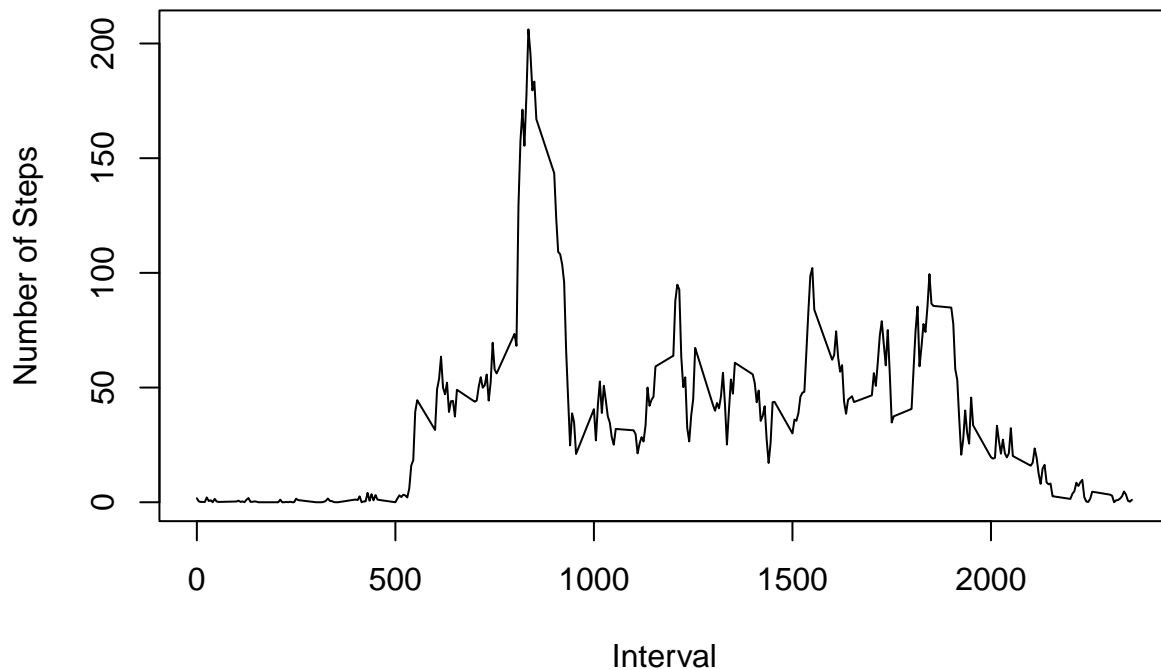
```
sprintf("MEDIAN of steps taken each day = %.3f", rmedian)
```

```
## [1] "MEDIAN of steps taken each day = 10765.000"
```

Time series plot of the average number of steps taken

```
steps_by_interval <- aggregate(steps ~ interval, data, mean)
plot(steps_by_interval$interval, steps_by_interval$steps, type="l", xlab="Interval", ylab="Number of Steps")
```

Average Number of Steps per Day by Interval



The 5-minute interval that, on average, contains the maximum number of steps

```
max_interval <- steps_by_interval[which.max(steps_by_interval$steps),1]
sprintf("MAXIMUM number of steps in 5-minute interval = %.0f", max_interval)
```

```
## [1] "MAXIMUM number of steps in 5-minute interval = 835"
```

Code to describe and show a strategy for imputing missing data

```
incomplete_data <- sum(!complete.cases(data))
sprintf("MISSING data = %.0f", incomplete_data)
```

```
## [1] "MISSING data = 2304"
```

The strategy is use the average value for that missing interval. For example, at “date”==“2012-10-01” the value 0 is NA, so the value 1.7169811 will be used as steps. This value came from the data set `steps_by_interval`.

```
imputed_data <- transform(data, steps = ifelse(is.na(data$steps), steps_by_interval$steps[match(data$date, steps_by_interval$date)], data$steps))
```

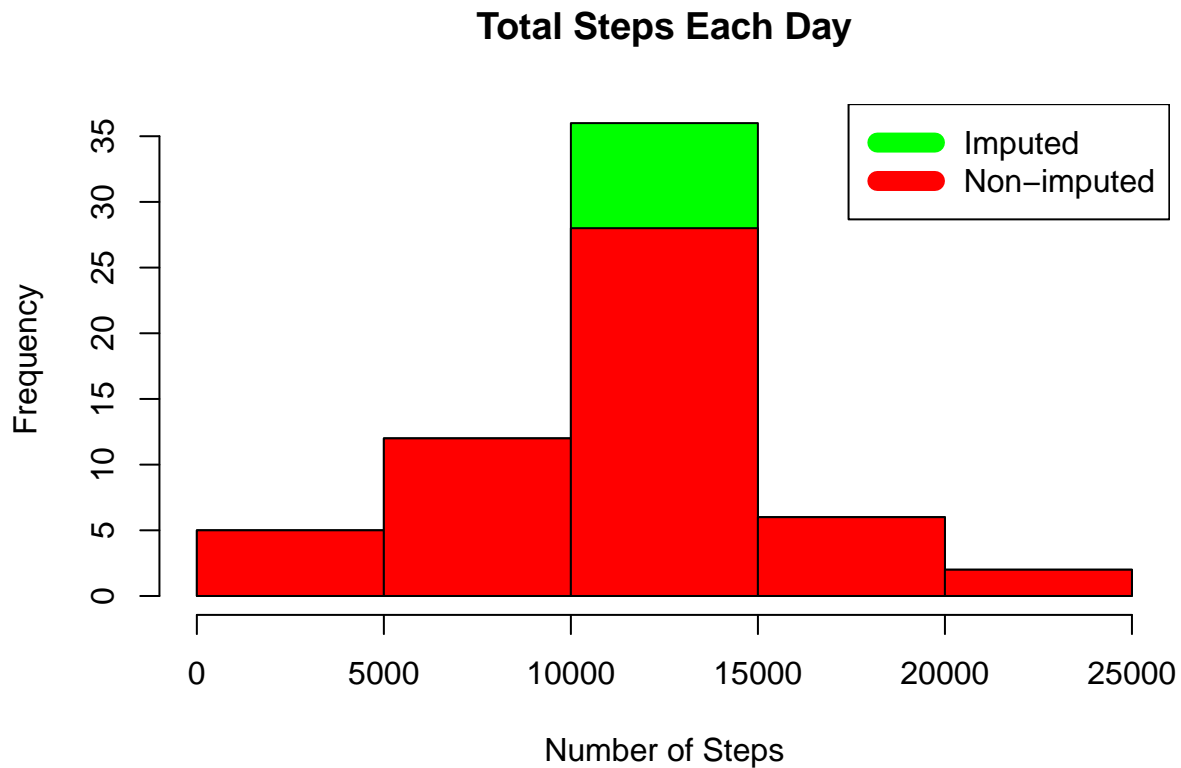
Histogram of the total number of steps taken each day after missing values are imputed

```

imputed_steps_by_day <- aggregate(steps ~ date, imputed_data, sum)
hist(imputed_steps_by_day$steps, main = paste("Total Steps Each Day"), col="green", xlab="Number of Steps", add=T)

hist(steps_by_day$steps, main = paste("Total Steps Each Day"), col="red", xlab="Number of Steps", add=T)
legend("topright", c("Imputed", "Non-imputed"), col=c("green", "red"), lwd=10)

```



MEAN difference

```

rmean <- mean(steps_by_day$steps)
imputed_rmean <- mean(imputed_steps_by_day$steps)
sprintf("MEAN of steps taken each day = %.3f", rmean)

```

```
## [1] "MEAN of steps taken each day = 10766.189"
```

```
sprintf("MEAN of steps taken each day with IMPUTED data = %.3f", imputed_rmean)
```

```
## [1] "MEAN of steps taken each day with IMPUTED data = 10766.189"
```

```
sprintf("The difference is %.3f ", imputed_rmean-rmean)
```

```
## [1] "The difference is 0.000 "
```

MEDIAN difference

```
rmedian <- median(steps_by_day$steps)
imputed_rmedian <- median(imputed_steps_by_day$steps)
sprintf("MEDIAN of steps taken each day = %.3f", rmedian)
```

```
## [1] "MEDIAN of steps taken each day = 10765.000"
```

```
sprintf("MEDIAN of steps taken each day with IMPUTED data = %.3f", imputed_rmedian)
```

```
## [1] "MEDIAN of steps taken each day with IMPUTED data = 10766.189"
```

```
sprintf("The difference is %.3f ", imputed_rmedian-rmedian)
```

```
## [1] "The difference is 1.189 "
```

TOTAL steps difference

```
total <- sum(steps_by_day$steps)
imputed_total <- sum(imputed_steps_by_day$steps)
sprintf("TOTAL of steps = %.3f", total)
```

```
## [1] "TOTAL of steps = 570608.000"
```

```
sprintf("TOTAL of steps with IMPUTED data = %.3f", imputed_total)
```

```
## [1] "TOTAL of steps with IMPUTED data = 656737.509"
```

```
sprintf("The difference is %.3f ", imputed_total-total)
```

```
## [1] "The difference is 86129.509 "
```

Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
weekend_days_set <- c("Saturday", "Sunday")
imputed_data$dow = as.factor(ifelse(is.element(weekdays(as.Date(imputed_data$date))), weekend_days_set),
imputed_steps_by_interval <- aggregate(steps ~ interval + dow, imputed_data, mean)

library(lattice)

xyplot(imputed_steps_by_interval$steps ~ imputed_steps_by_interval$interval|imputed_steps_by_interval$dow,
```

Average Steps per Day by Interval

