Reproducible Research Course Project 1

Troy Huffman

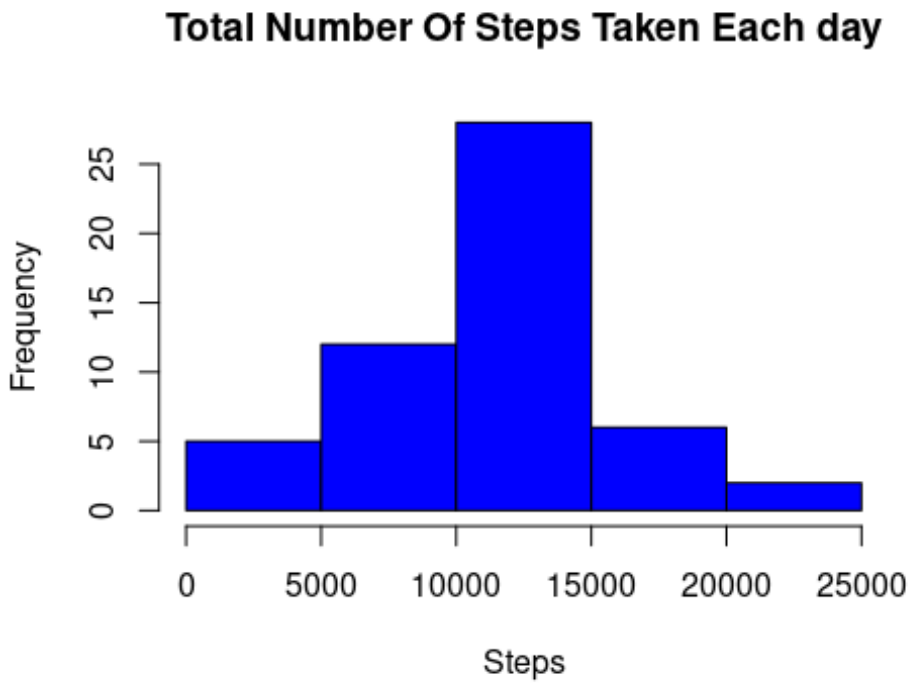
2/9/2022

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

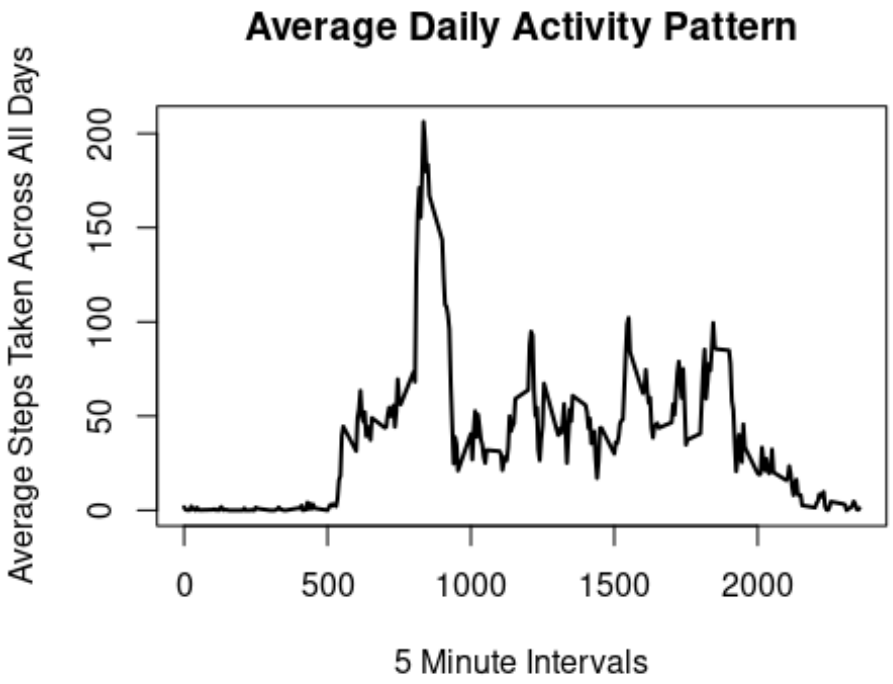
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

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

tot_step_day <- aggregate(steps ~ date, data = activity, FUN = sum, na.rm =
TRUE)
head(tot_step_day)

##         date steps
## 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

hist(tot_step_day$steps,
     col="blue",
     xlab = "Steps",
     ylab = "Frequency",
     main = "Total Number Of Steps Taken Each day")
```




```
print_mean_step <-"Mean steps (no missing values replaced):"
print_mean_step

## [1] "Mean steps (no missing values replaced):"

tot_step_day_mean <- mean(tot_step_day$steps)
tot_step_day_mean

## [1] 10766.19

print_median_step <-"Median steps (no missing values replaced):"
print_median_step

## [1] "Median steps (no missing values replaced):"

tot_step_day_median <- median(tot_step_day$steps)
tot_step_day_median

## [1] 10765

#total steps over 5 minutes
tot_int <- aggregate(steps ~ interval, data = activity, FUN = mean, na.rm =
TRUE)

plot(tot_int$interval, tot_int$steps,
     type = "l", lwd = 2,
     xlab = "5 Minute Intervals",
```

```
     ylab = "Average Steps Taken Across All Days",
     main = "Average Daily Activity Pattern")
```




```
max_step <- tot_int$interval[which.max(tot_int$steps)]
print_max_step <-"Interval that contains the maximum number of steps:"
print_max_step
```

```
## [1] "Interval that contains the maximum number of steps:"
```

```
max_step
```

```
## [1] 835
```

```
print_missing <- "TRUE represents the total number of NA values"
print_missing
```

```
## [1] "TRUE represents the total number of NA values"
```

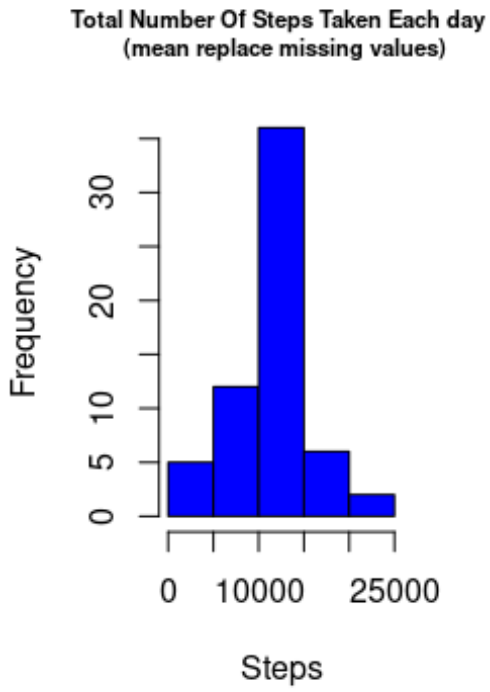
```
table(is.na(activity))
```

```
##
## FALSE  TRUE
## 50400  2304
```

```
activity2 <- activity
nas <- is.na(activity2$steps)
avg_interval <- tapply(activity2$steps, activity2$interval, mean, na.rm=TRUE,
simplify = TRUE)
activity2$steps[nas] <- avg_interval[as.character(activity2$interval[nas])]
```

```r
tot_steps_day_new<- aggregate(steps ~ date, activity2, FUN=sum)

par(mfrow=c(1,2))

#Histogram after imputing mean for NA values
hist(tot_steps_day_new$steps,
     col="blue",
     xlab = "Steps",
     ylab = "Frequency",
     main = "Total Number Of Steps Taken Each day \n (mean replace missing
values)",
     cex.main = 0.7)

print_mean_step_new <-"Mean steps (mean replace missing values):"
print_mean_step_new

## [1] "Mean steps (mean replace missing values):"

tot_step_day_mean_new <- mean(tot_steps_day_new$steps)
tot_step_day_mean_new

## [1] 10766.19

print_mean_step

## [1] "Mean steps (no missing values replaced):"

tot_step_day_mean

## [1] 10766.19

diff_mean_step <-"Difference in mean steps:"
diff_mean_step_val <- tot_step_day_mean_new - tot_step_day_mean
diff_mean_step

## [1] "Difference in mean steps:"

diff_mean_step_val

## [1] 0

print_median_step_new <-"Median steps (mean replace missing values):"
print_median_step_new

## [1] "Median steps (mean replace missing values):"

tot_step_day_median_new <- median(tot_steps_day_new$steps)
tot_step_day_median_new

## [1] 10766.19

print_median_step
```

```
## [1] "Median steps (no missing values replaced):"

tot_step_day_median

## [1] 10765

diff_median_step <-"Difference in median steps:"
diff_median_step_val <- tot_step_day_median_new - tot_step_day_median
diff_median_step

## [1] "Difference in median steps:"

diff_median_step_val

## [1] 1.188679
```




```
library(chron)
library(ggplot2)

activity2$dayofweek <- ifelse(is.weekend(tot_steps_day_new$date), "weekend",
"weekday")

meaninterval_new<- aggregate(steps ~ interval + dayofweek, activity2,
FUN=mean)

ggplot(meaninterval_new, aes(x=interval, y=steps)) +
  geom_line(color="blue", size=1) +
```

```
  facet_wrap(~dayofweek, nrow=2) +
  labs(x="\nInterval", y="\nNumber of steps")
```