Steps Activity

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 (http://rmarkdown.rstudio.com).

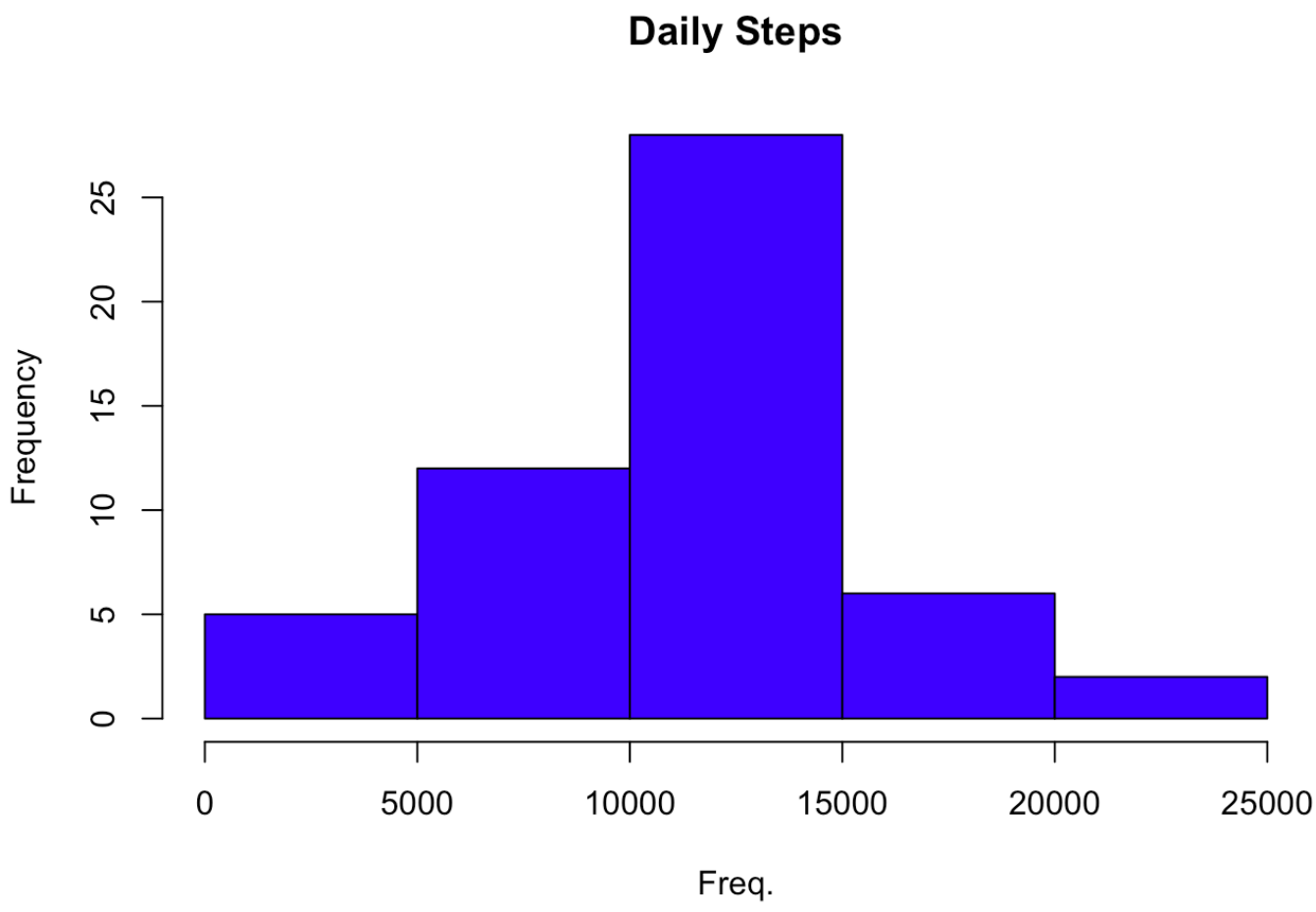
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
getwd()
```

```
## [1] "/Users/cbeltis/Documents"
```

```
setwd("/Users/cbeltis/Documents")

activity_data <- read.csv("activity.csv")
activity_data$date <- as.Date(activity_data$date, "%Y-%m-%d")
Steps <- aggregate(steps ~ date, data = activity_data, sum, na.rm = TRUE)
hist(Steps$steps, main = "Daily Steps", xlab = "Freq.", col = "blue")
```



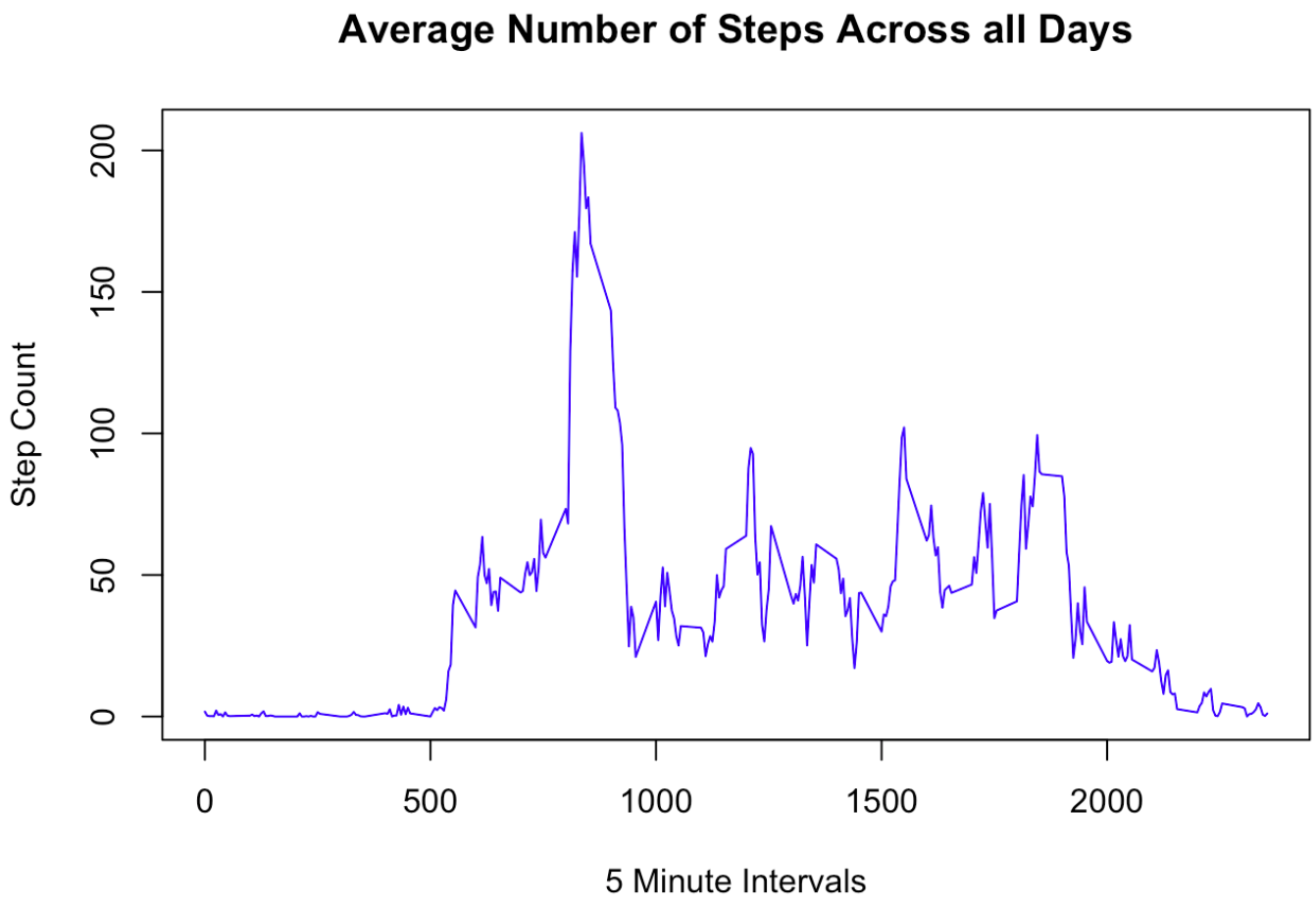
```
mean(Steps$steps)
```

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

```
median(Steps$steps)
```

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

```
step_intervals <- tapply(activity_data$steps, activity_data$interval, mean, na.rm
= TRUE)
plot(row.names(step_intervals), step_intervals, type = "l", xlab = "5 Minute Inter
vals",
     ylab = "Step Count", main = "Average Number of Steps Across all Days",
     col = "blue")
```




```
max_int <- which.max(step_intervals)
names(max_int)
```

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

```
NA_count <- sum(is.na(activity_data))
NA_count
```

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

```
NA_values <- which(is.na(activity_data$steps))
updated_vals <- rep(mean(activity_data$steps, na.rm=TRUE), times=length(NA_value
s))
activity_data[NA_values, "steps"] <- updated_vals
agg <- aggregate(activity_data$steps, by=list(activity_data$date), FUN=sum)
hist(agg$x, col="red", xlab="Number of steps",  main="Total Steps by Day")
mean(agg$x)
```

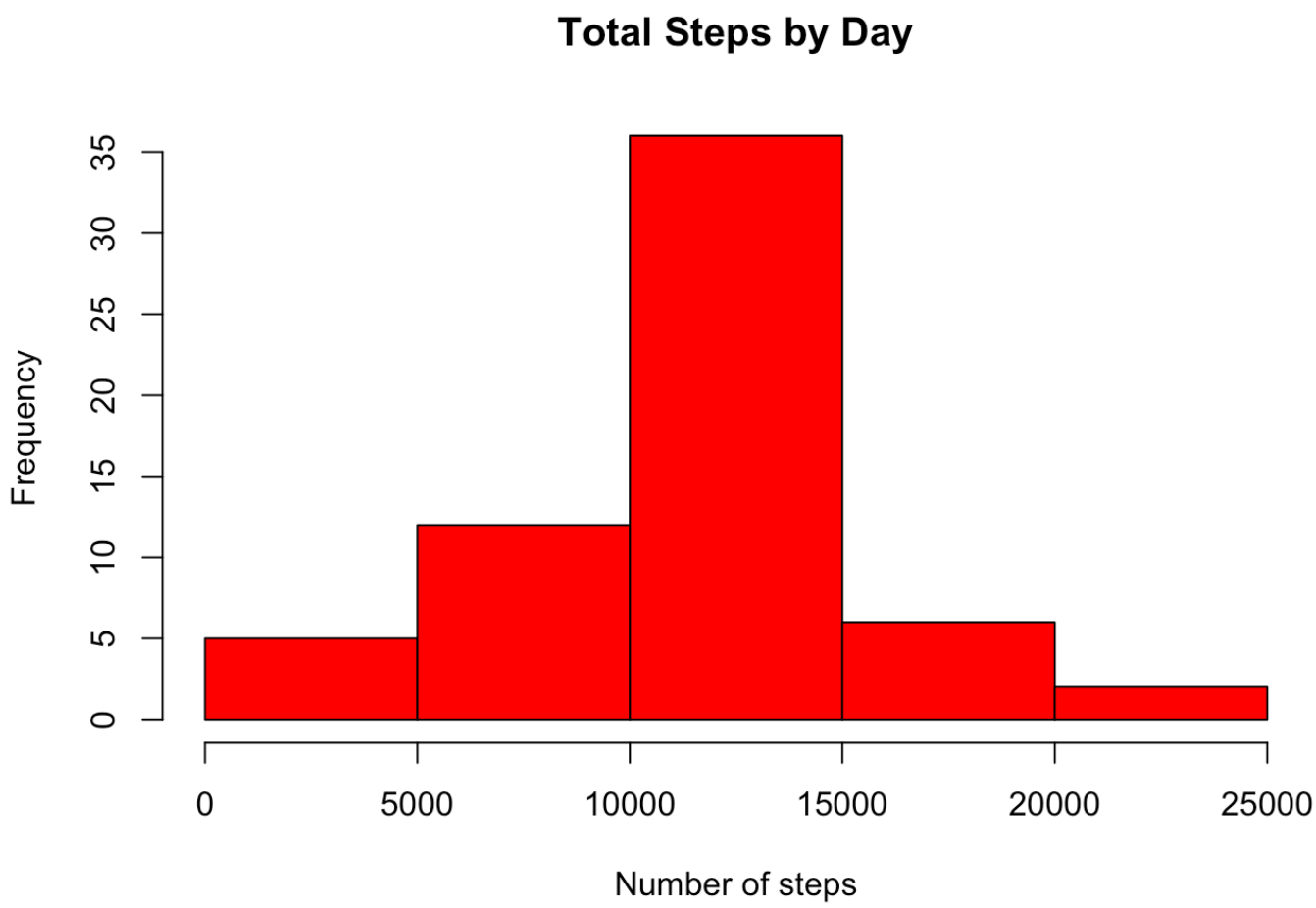
```
## [1] 10766.19
```

```
median(agg$x)
```

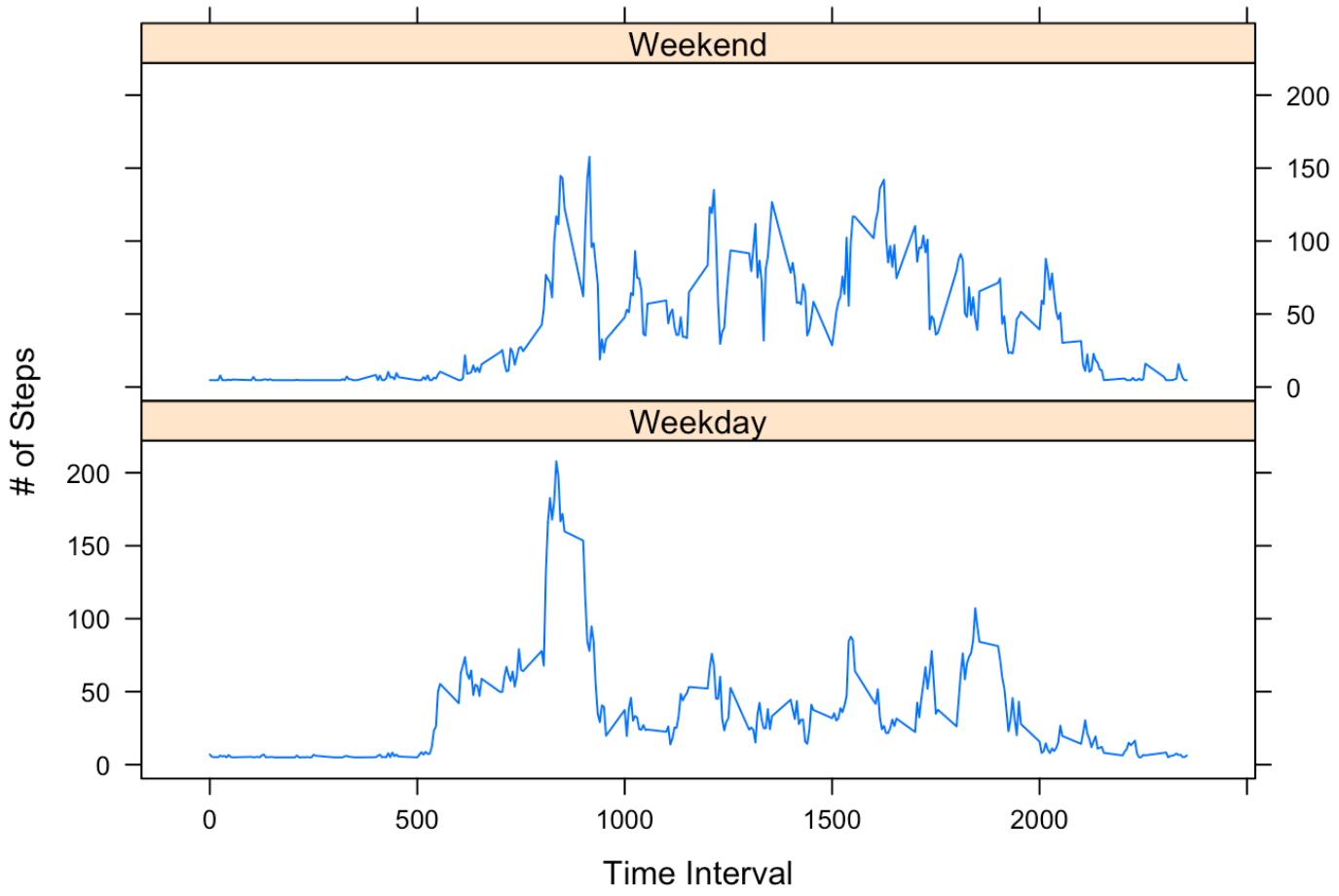
```
## [1] 10766.19
```

```
days <- weekdays(activity_data$date)
day_type <- vector()
for (i in 1:nrow(activity_data)) {
  if (days[i] == "Saturday") {day_type[i] <- "Weekend"}
  else if (days[i] == "Sunday") {day_type[i] <- "Weekend"}
  else {day_type[i] <- "Weekday"}}
activity_data$day_type <- day_type
activity_data$day_type <- factor(activity_data$day_type)

stepsByDay <- aggregate(steps ~ interval + day_type, data = activity_data, mean)
names(stepsByDay) <- c("interval", "day_type", "steps")
library(lattice)
```




```
xyplot(steps ~ interval | day_type, stepsByDay, type = "l", layout = c(1, 2),
       xlab = "Time Interval", ylab = "# of Steps")
```


Weekend
Weekday
# of Steps
Time Interval
0
50
100
150
200
0
500
1000
1500
2000