Reproducible Research Week 2 Assignment

Wei Hao Khoong

26 May 2018

1: Code for reading in the dataset and/or processing the data

```
setwd("C:/Users/khoongwh/Desktop")
activity <- read.csv("activity.csv")
```

Exploring the basics of this data

```
dim(activity)
```

```
## [1] 17568     3
```

```
names(activity)
```

```
## [1] "steps"    "date"     "interval"
```

```
head(activity)
```

```
##   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
```

```
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

```
sum(is.na(activity$steps))/dim(activity)[[1]] #Total number of missing data
```

```
## [1] 0.1311475
```

```
library(lubridate) #Transforming the date column into date format
```

```
## Warning: package 'lubridate' was built under R version 3.3.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following object is masked from 'package:base':
##
##     date
```

```
activity$date <- ymd(activity$date)
length(unique(activity$date))
```

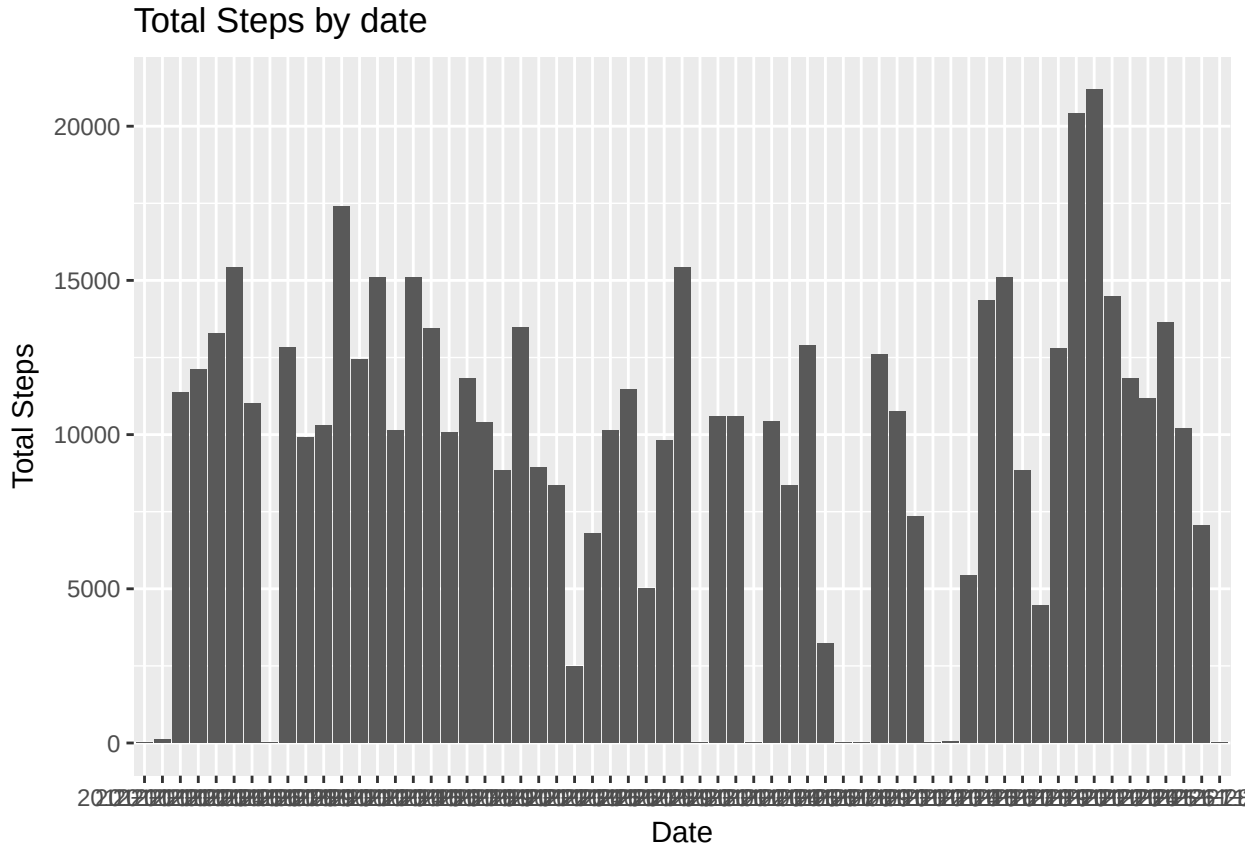
```
## [1] 61
```

2: Histogram of the total number of steps taken each day

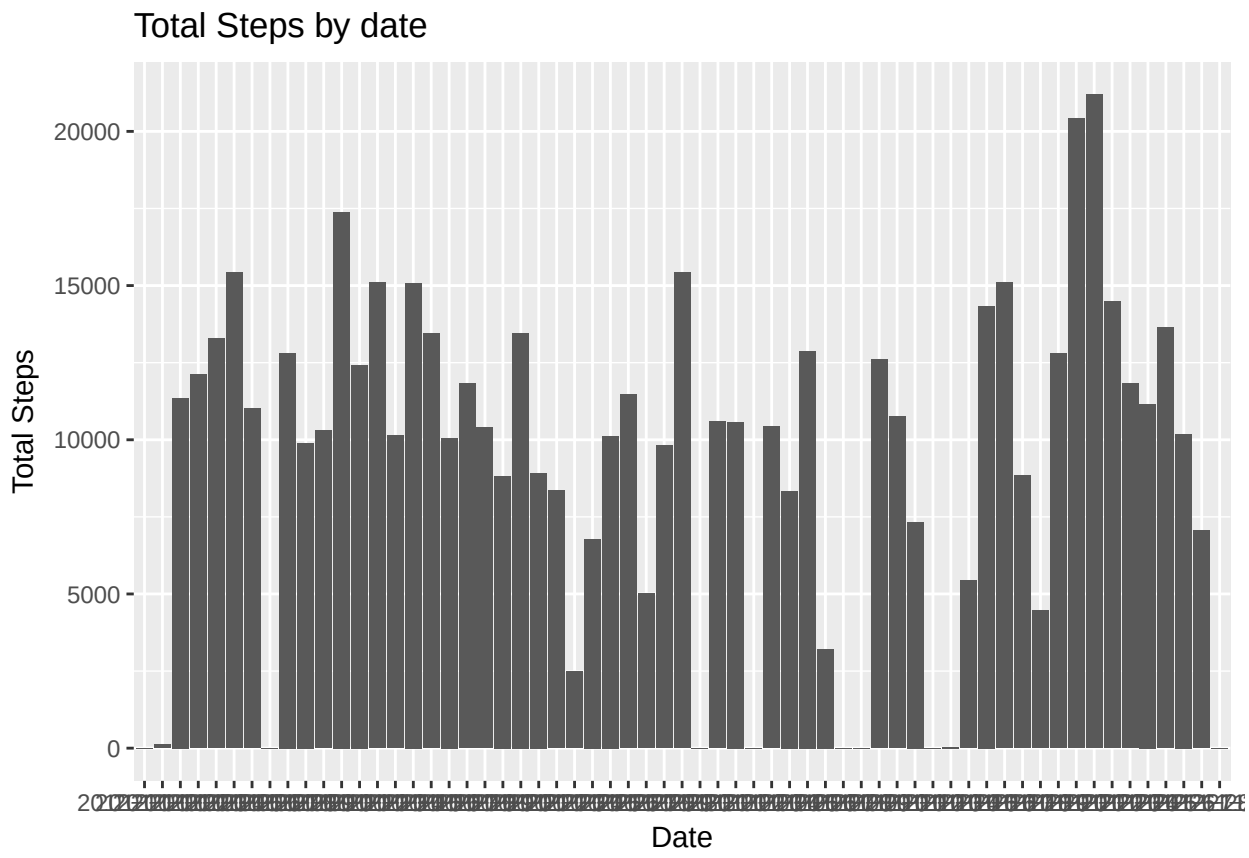
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
q2 <- data.frame(tapply(activity$steps,activity$date,sum,na.rm=TRUE))
q2$date <- rownames(q2)
rownames(q2)<-NULL
names(q2)[[1]] <- "Total Steps"

#Total Steps by date bar chart
ggplot(q2,aes(y=q2$`Total Steps`,x=q2$date))+geom_bar(stat="identity") + ylab("Total Steps")+xlab("Date
```

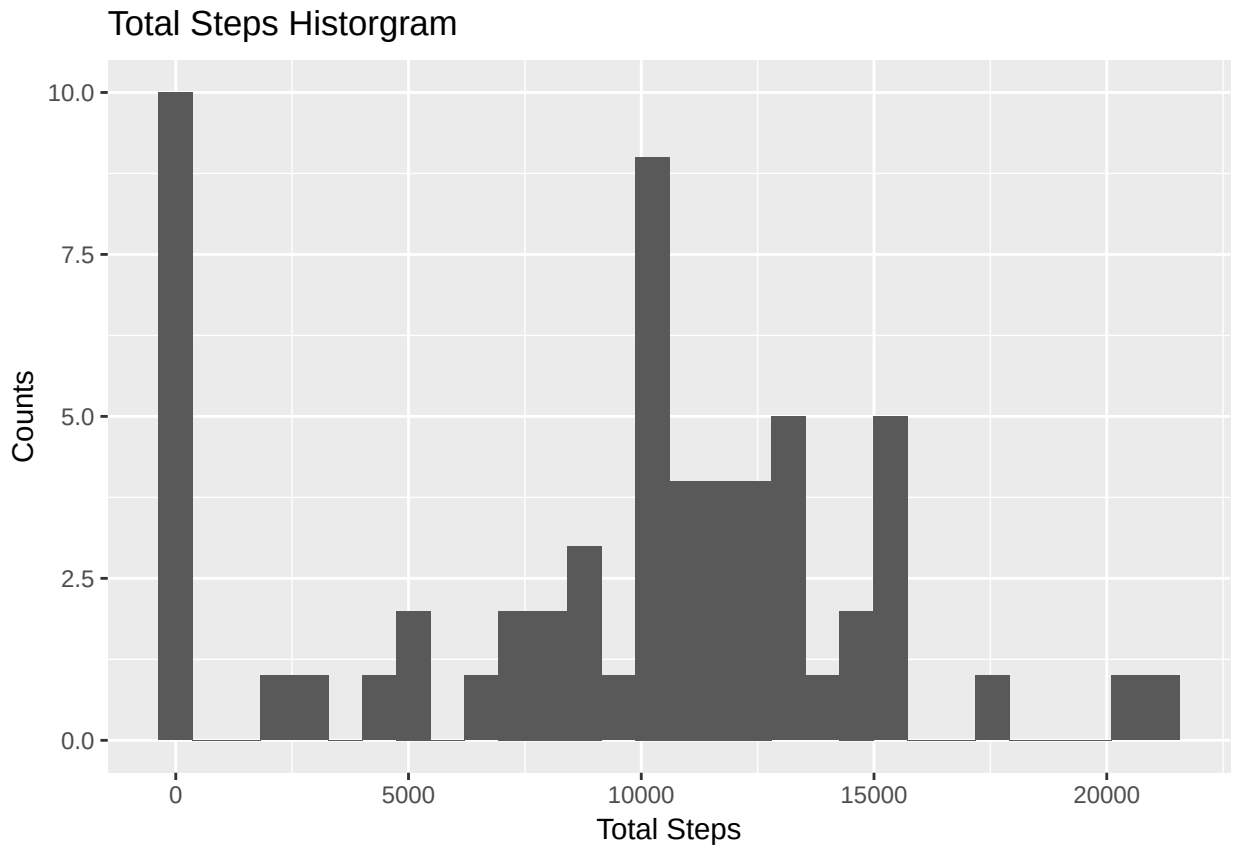



```
ggplot(q2,aes(y=q2$`Total Steps`,x=q2$date))+geom_bar(stat="identity") + ylab("Total Steps")+xlab("Date
```



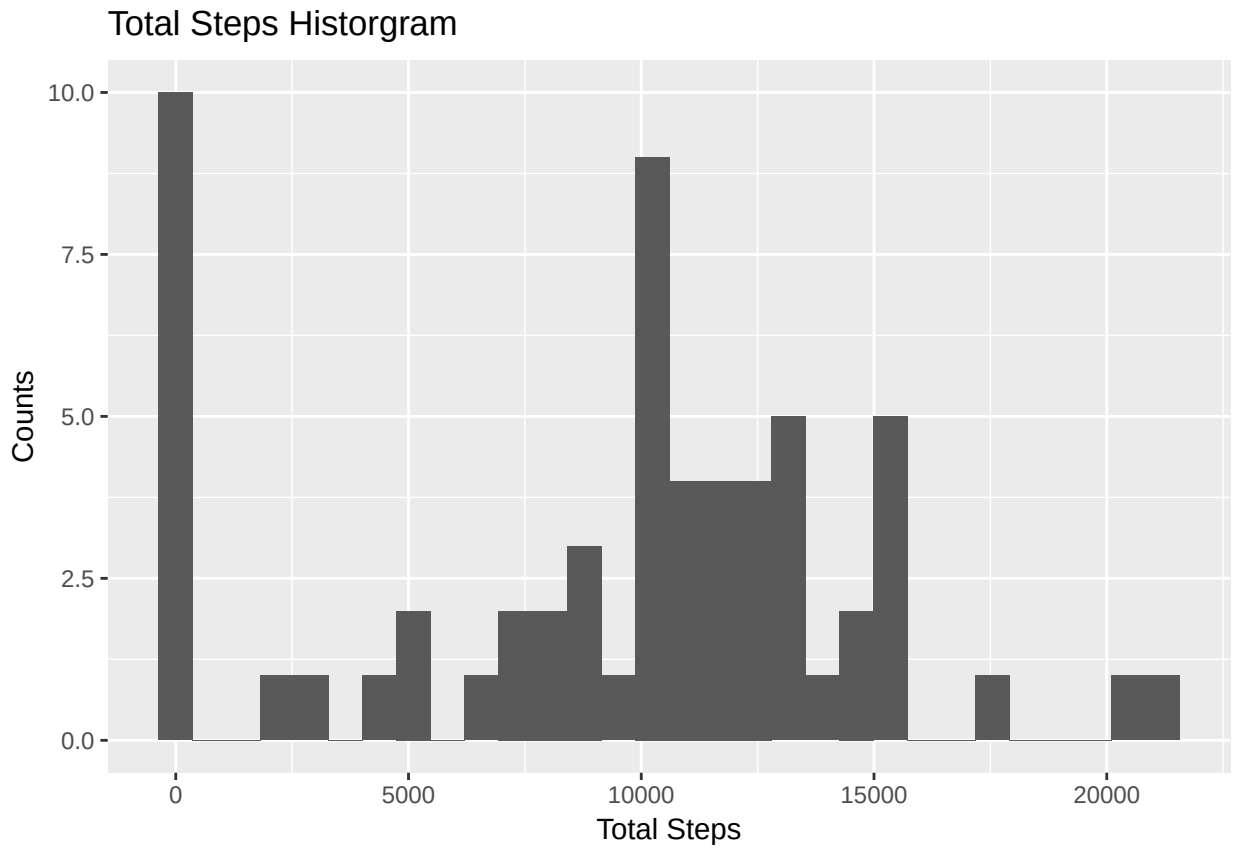

```
#Histogram of total steps
qplot(q2$`Total Steps`,geom="histogram",xlab="Total Steps",ylab="Counts",main="Total Steps Historgram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qplot(q2$`Total Steps`,geom="histogram",xlab="Total Steps",ylab="Counts",main="Total Steps Historgram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



##3: Mean and median number of steps taken each day

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
q3 <- data.frame(round(tapply(activity$steps,activity$date,mean,na.rm=TRUE),2))
q3$date <- rownames(q3)
rownames(q3) <- NULL
names(q3)[[1]] <- "Mean Steps"
temp<-activity%>%select(date,steps) %>% group_by(date) %>% summarise(median(steps))
```

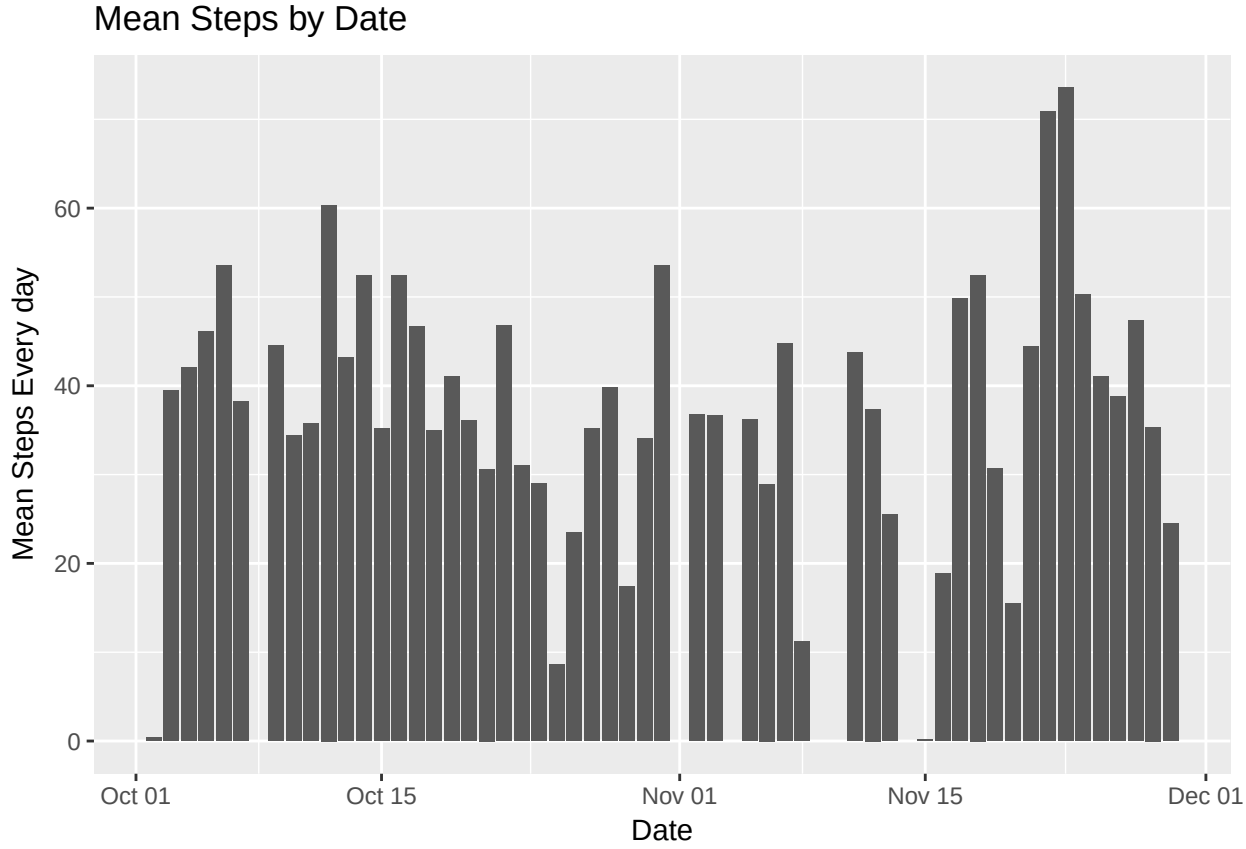
```
## Warning: package 'bindrcpp' was built under R version 3.3.3
```

```
names(temp)[[2]] <- "Median Steps"
q3$median <- temp$`Median Steps`
q3 <- q3 %>% select(date,`Mean Steps`,median)
```

4: Time series plot of the average number of steps taken

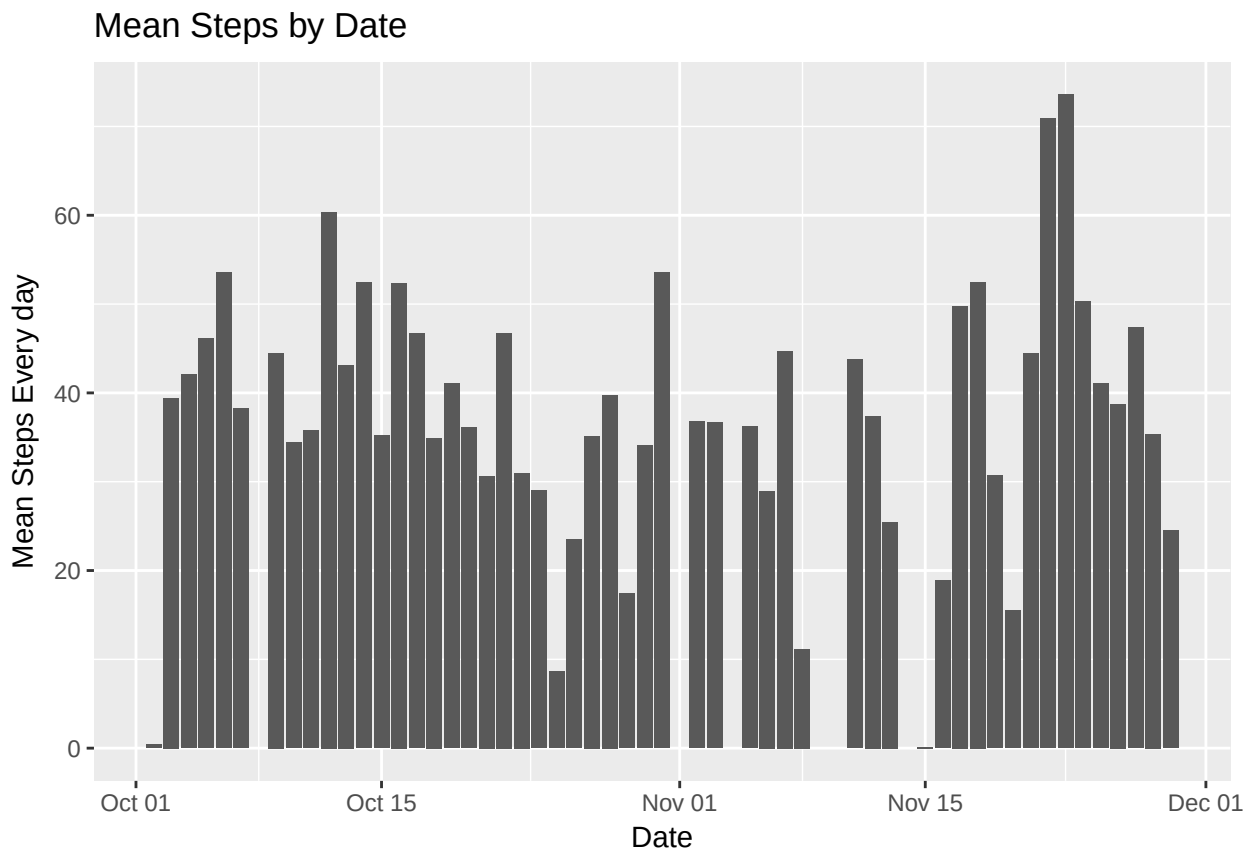
```
q4 <- q3
q4$date <- as.Date(q4$date,format="%Y-%m-%d")
ggplot(q4,aes(x=q4$date,y=q4$`Mean Steps`))+geom_bar(stat="identity")+scale_x_date()+ylab("Mean Steps E
```

```
## Warning: Removed 8 rows containing missing values (position_stack).
```




```
ggplot(q4,aes(x=q4$date,y=q4$`Mean Steps`))+geom_bar(stat="identity")+scale_x_date()+ylab("Mean Steps E
```

```
## Warning: Removed 8 rows containing missing values (position_stack).
```




```
dev.off()
```

```
## null device
##           1
```

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

```
#This is assuming that the words on average means averaging steps by date and interval
activity$interval <- factor(activity$interval)
q5 <- aggregate(data=activity,steps~date+interval,FUN="mean")
q5 <- aggregate(data=q5,steps~interval,FUN="max")
```

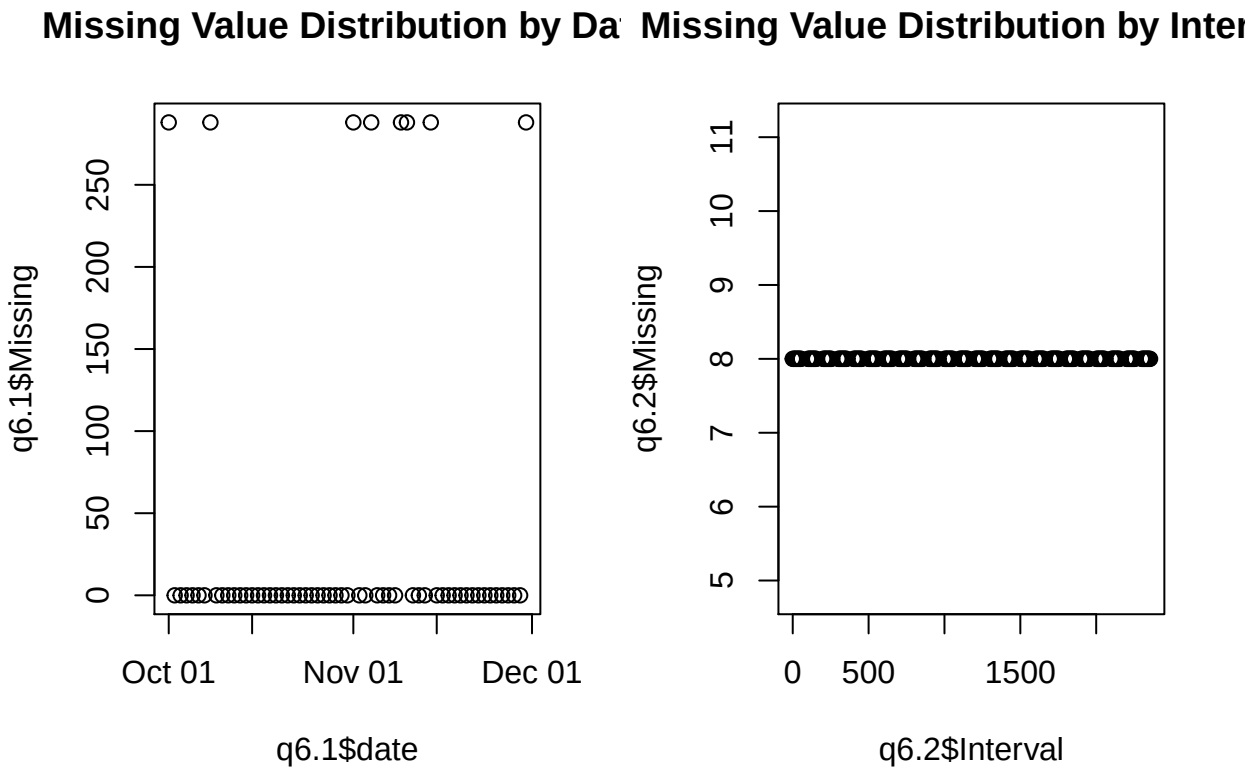
6: Code to describe and show a strategy for imputing missing data

There are multiple strategies to deal with multiple value imputations. These include:

1. Constant value imputations
2. Mean/mode value substitutions
3. Regression model value imputations

For the purpose of this question, the mean/modal value substitution will be implemented to impute missing values. This means using the mean values to substitute the missing values in the original data set. Furthermore, before any sort of imputation, we first try to understand what are the distributions of missing values by date and interval:

```r
q6 <- activity
q6$Missing <- is.na(q6$steps)
q6 <- aggregate(data=q6,Missing~date+interval,FUN="sum")
q6.1<-data.frame(tapply(q6$Missing,q6$date,sum))
q6.1$date<-rownames(q6.1)
rownames(q6.1) <- NULL
names(q6.1) <- c("Missing","date")
q6.1$date <- as.Date(q6.1$date,format="%Y-%m-%d")

q6.2 <- data.frame(tapply(q6$Missing,q6$interval,sum))
q6.2$date <- rownames(q6.2)
rownames(q6.2) <- NULL
names(q6.2) <- c("Missing","Interval")

par(mfrow=c(1,2))
plot(y=q6.1$Missing,x=q6.1$date,main="Missing Value Distribution by Date")
plot(y=q6.2$Missing,x=q6.2$Interval,main="Missing Value Distribution by Interval")
```




```r
table(activity$date)
```

```
##
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##        288        288        288        288        288        288
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12
##        288        288        288        288        288        288
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18
```

```
##        288        288        288        288        288        288
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24
##        288        288        288        288        288        288
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30
##        288        288        288        288        288        288
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05
##        288        288        288        288        288        288
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
##        288        288        288        288        288        288
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17
##        288        288        288        288        288        288
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23
##        288        288        288        288        288        288
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29
##        288        288        288        288        288        288
## 2012-11-30
##        288
```

From the plot, we can observe that the missing values have a disctinct pattern. For every interval, there are consistantly 8 missing values. And for the date, there are consistantly 288 missing values. In total, there are 8 dates that have missing values. Thus, we can say that the mean value imputation is appropriate.

In particular, every date has 288 data points. This implies that the 8 dates have no data points at all. We can refine our analysis by focusing on these missing values, depending on their Weekday and interval parameters to matach with the average:

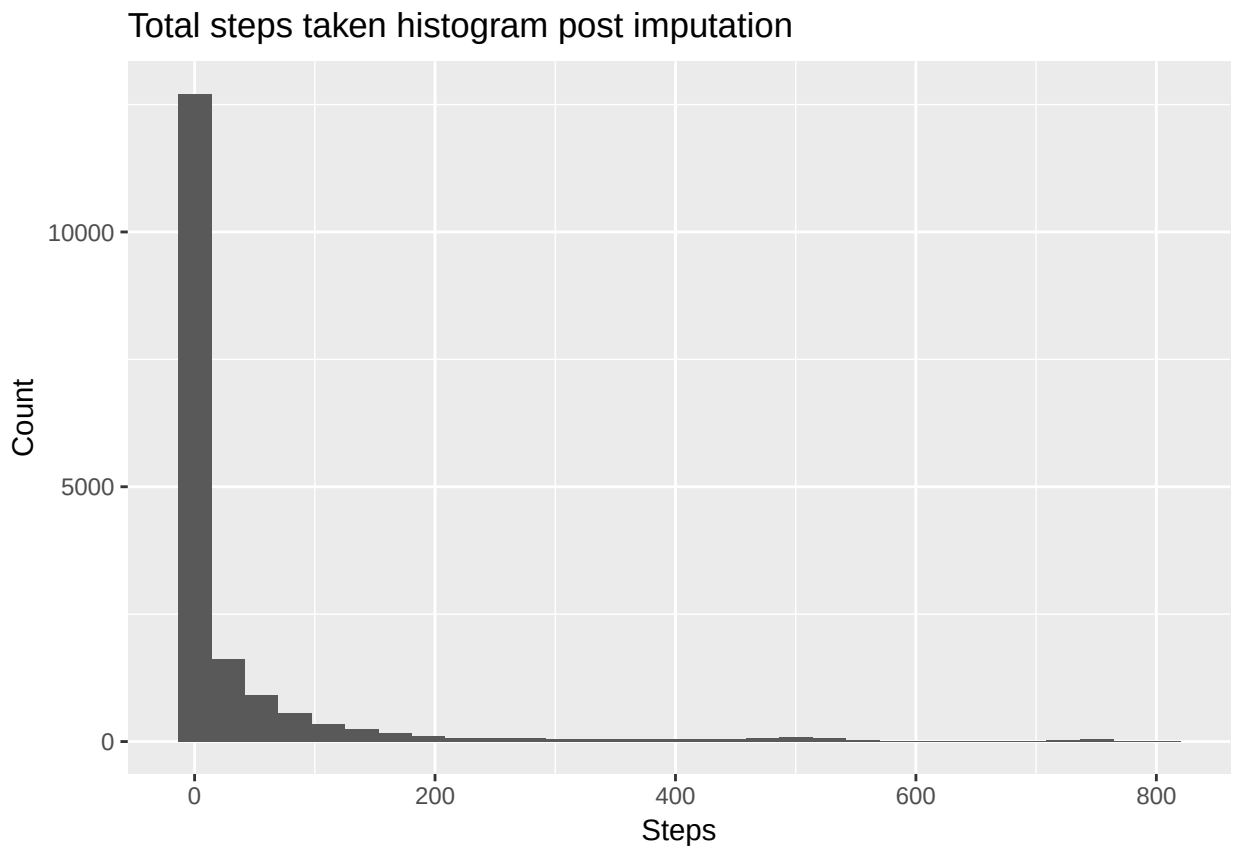
```
#Dates that have missing values
library(lubridate)
q6.3 <- as.data.frame(q6.1) %>% select(date,Missing) %>% arrange(desc(Missing))
q6.3 <- q6.3[which(q6.3$Missing!=0),]
q6.3$Weekday<-wday(q6.3$date,label=TRUE)
q6.4 <- activity
q6.4$weekday <- wday(q6.4$date,label=TRUE)

#To find the mean of steps every monday, and every interval
q6.5 <- aggregate(data=q6.4,steps~interval+weekday,FUN="mean",na.rm=TRUE)

#Now merge the pre-imputation table q6.4 table with the average table q6.5
q6.6<-merge(x=q6.4,y=q6.5,by.x=c("interval","weekday"),by.y=c("interval","weekday"),all.x=TRUE)

#Replace the steps.x column NA value with the values from steps.y column value
q6.6$Steps.Updated<-0
for (i in 1:dim(q6.6)[[1]]){
  if(is.na(q6.6[i,3])){q6.6[i,6]=q6.6[i,5]}
  else {q6.6[i,6]=q6.6[i,3]}
}

#Now simplify the imputed analytical data frame
q6.6 <-q6.6  %>% select(date,weekday,interval,Steps.Updated)
names(q6.6)[[4]]<-"Steps"
```

Step 7

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

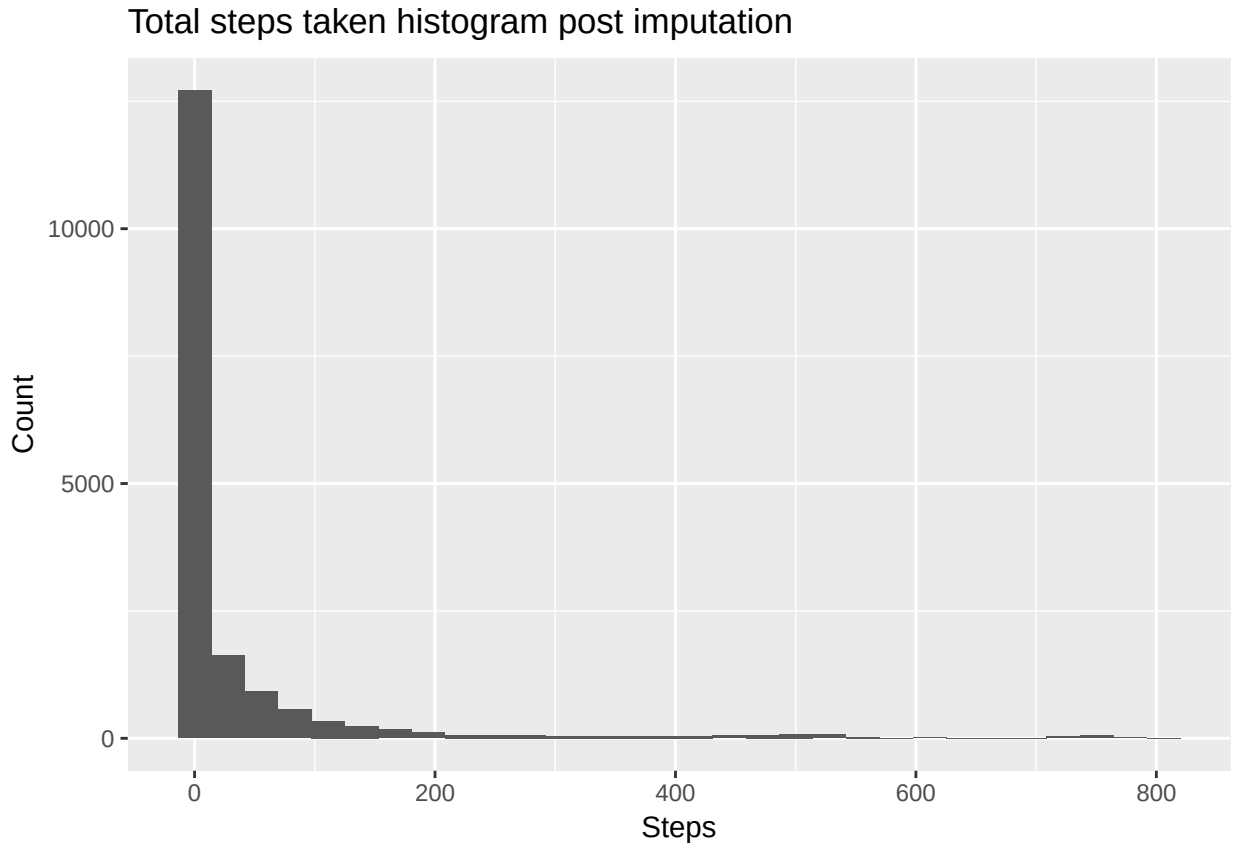
```
qplot(q6.6$Steps,geom="histogram",main="Total steps taken histogram post imputation",xlab="Steps",ylab=
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



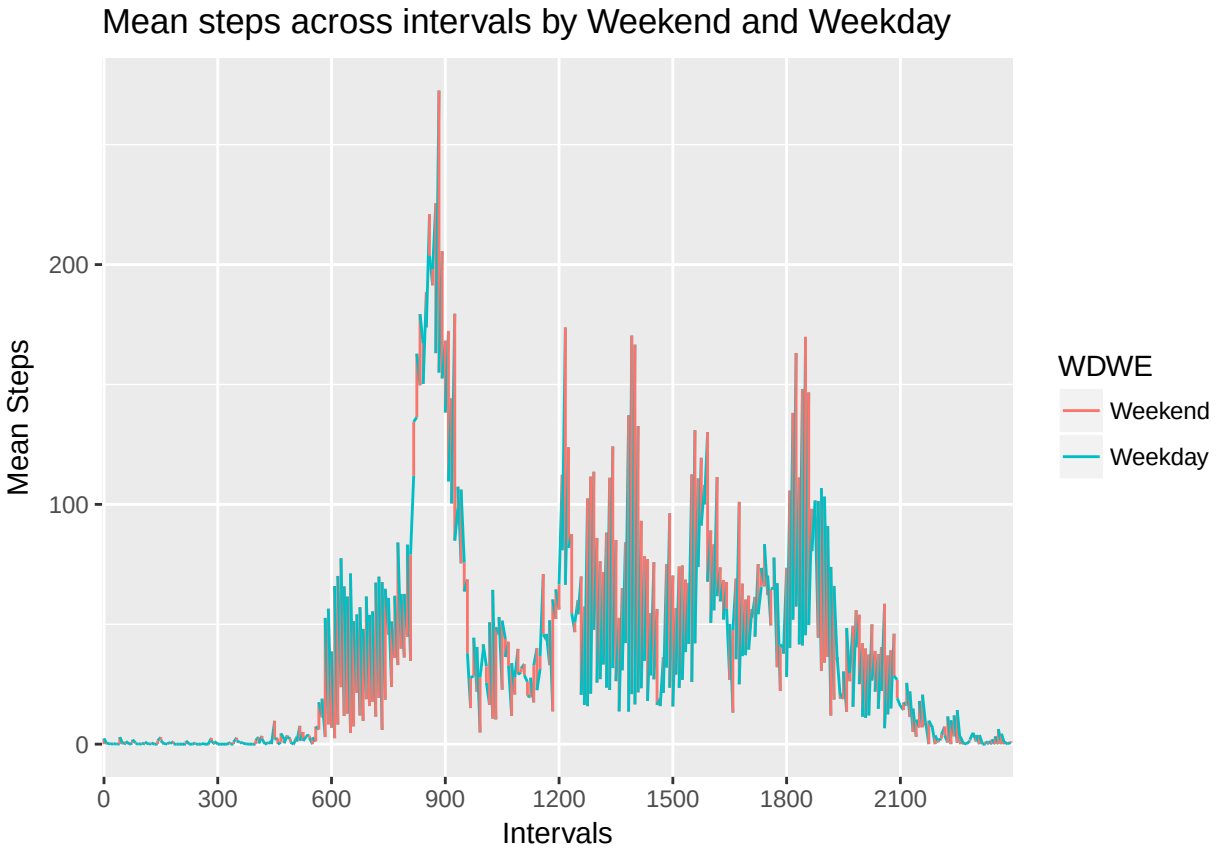

```
qplot(q6.6$Steps,geom="histogram",main="Total steps taken histogram post imputation",xlab="Steps",ylab=
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

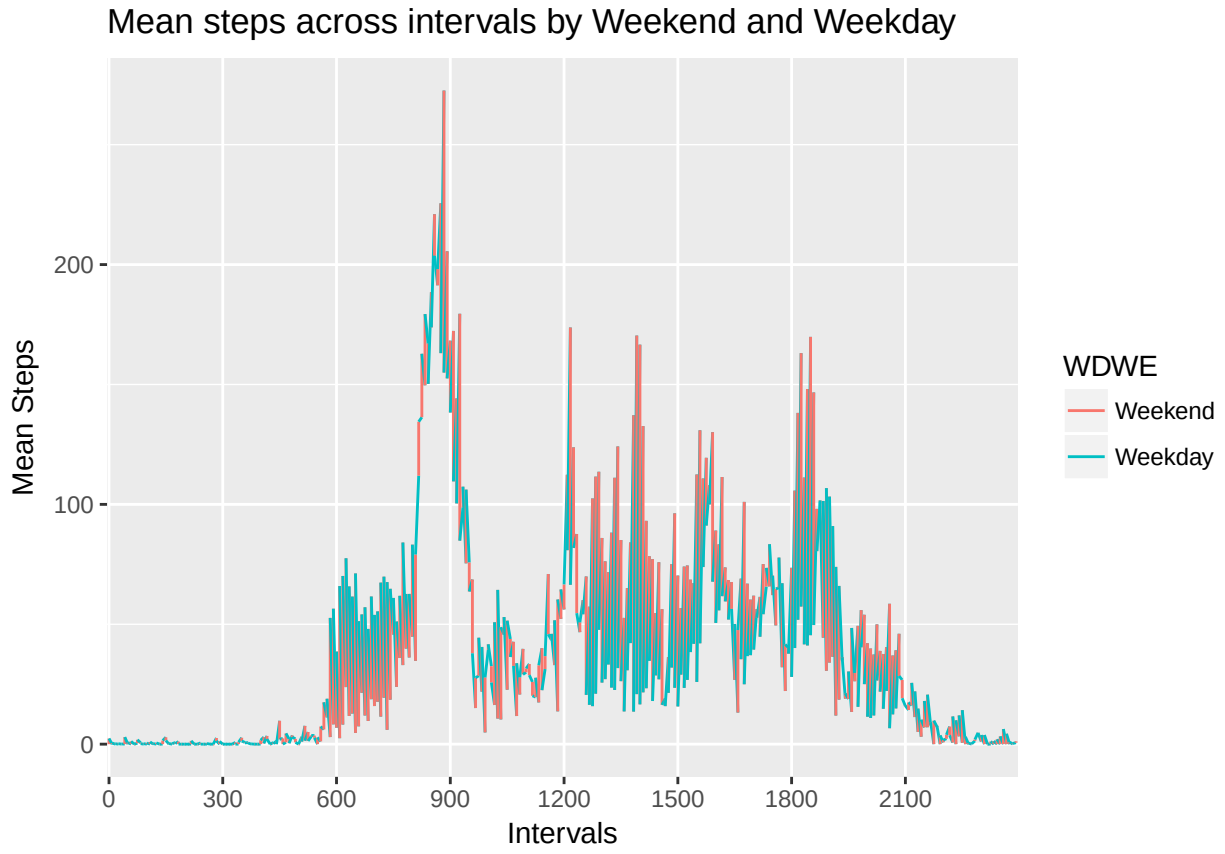


Step 8 Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

```
q8 <- q6.6
levels(q8$weekday) <- c(1,2,3,4,5,6,7)
q8$WDWE <- q8$weekday %in% c(1,2,3,4,5)
q8.1 <- aggregate(data=q8,Steps~interval+WDWE,mean,na.rm=TRUE)
q8.1$WDWE <- as.factor(q8.1$WDWE)
levels(q8.1$WDWE) <- c("Weekend","Weekday")

ggplot(data=q8.1,aes(y=Steps,x=interval,group=1,color=WDWE))+geom_line() +scale_x_discrete(breaks = seq
```

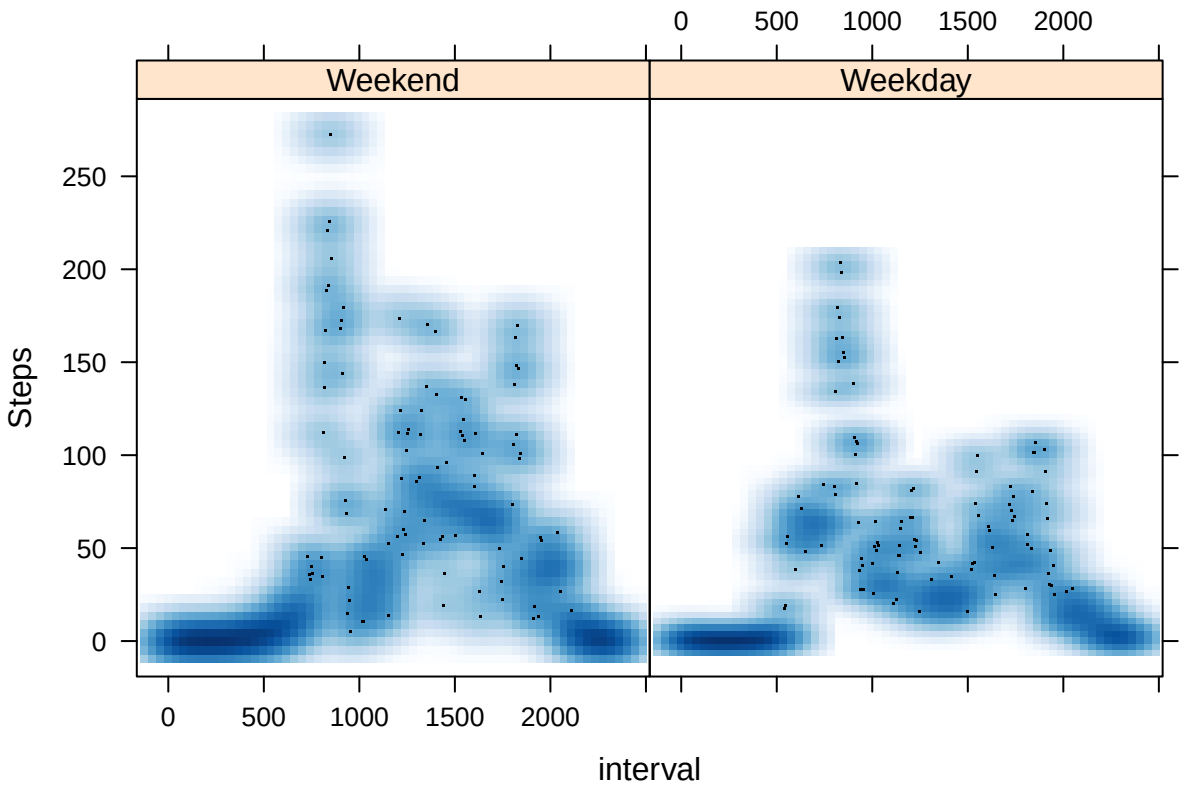



```
ggplot(data=q8.1,aes(y=Steps,x=interval,group=1,color=WDWE))+geom_line() +scale_x_discrete(breaks = seq
```




```
#Producing the panel plot
q8.1$interval<-as.numeric(as.character(q8.1$interval))
library(lattice)
xyplot(data=q8.1,Steps~interval|WDWE, grid = TRUE, type = c("p", "smooth"), lwd = 4,panel = panel.smoothl
```

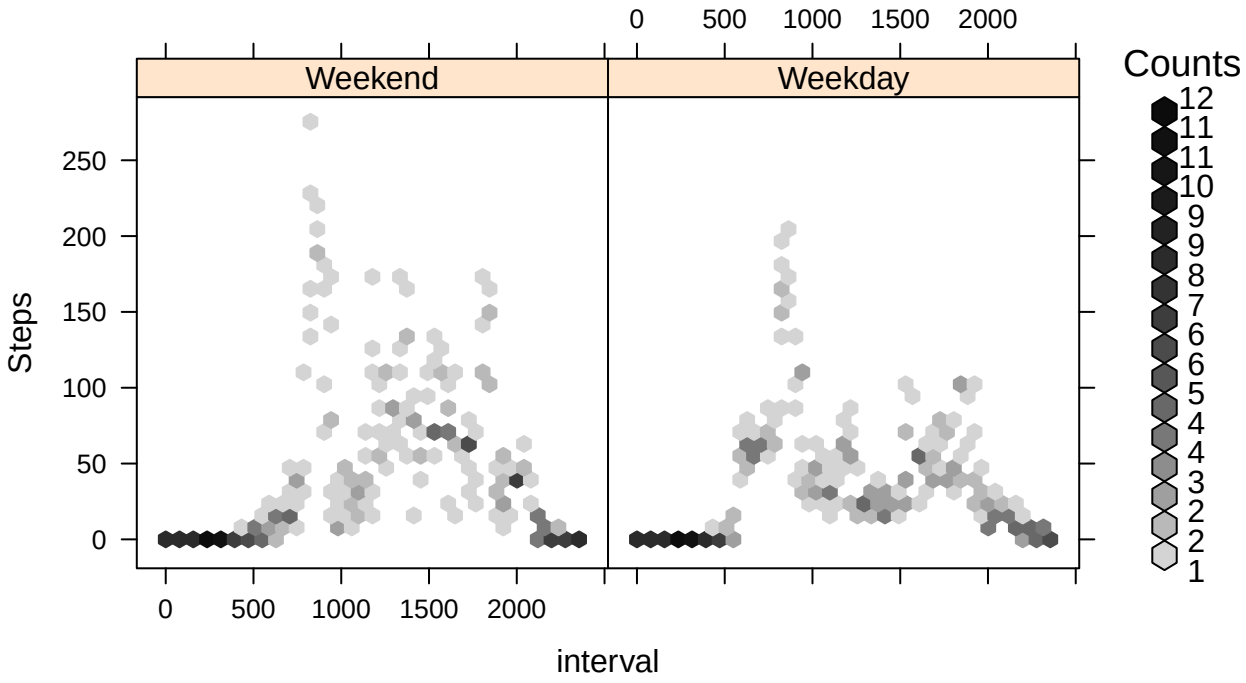
```
## (loaded the KernSmooth namespace)
```



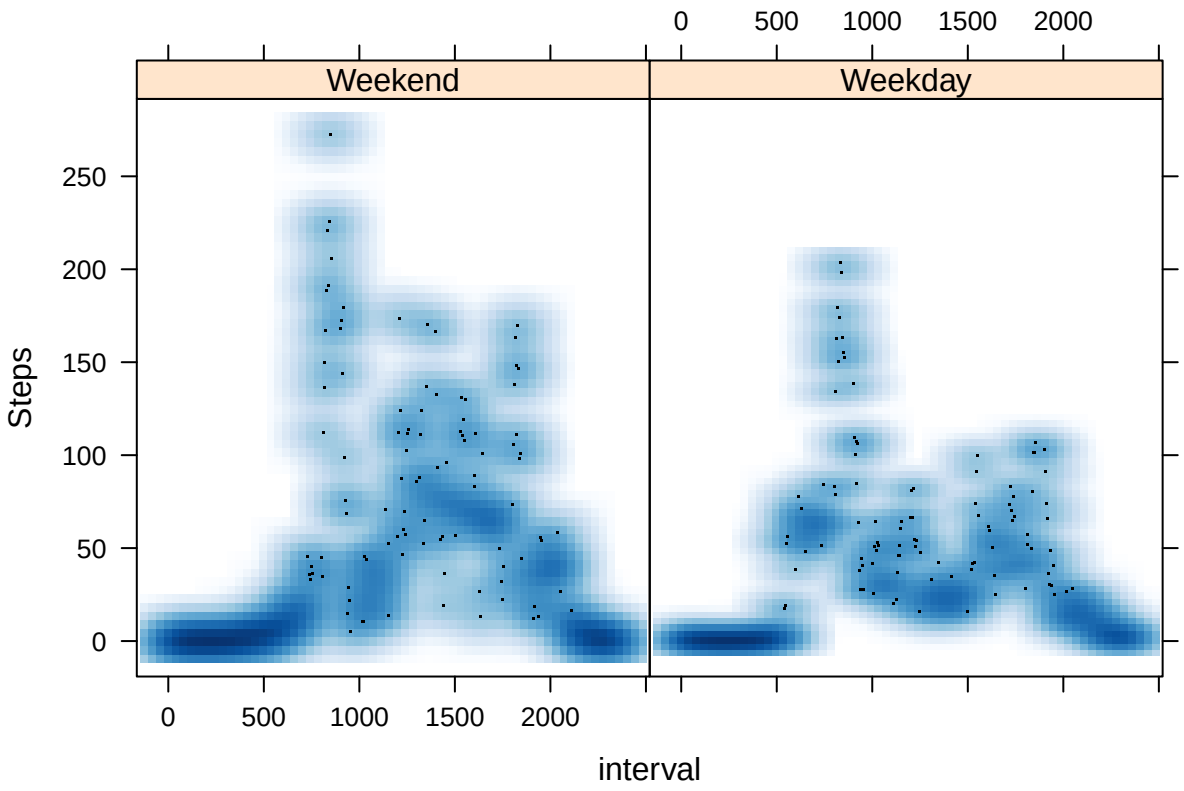

```
library(hexbin)
```

```
## Warning: package 'hexbin' was built under R version 3.3.3
```

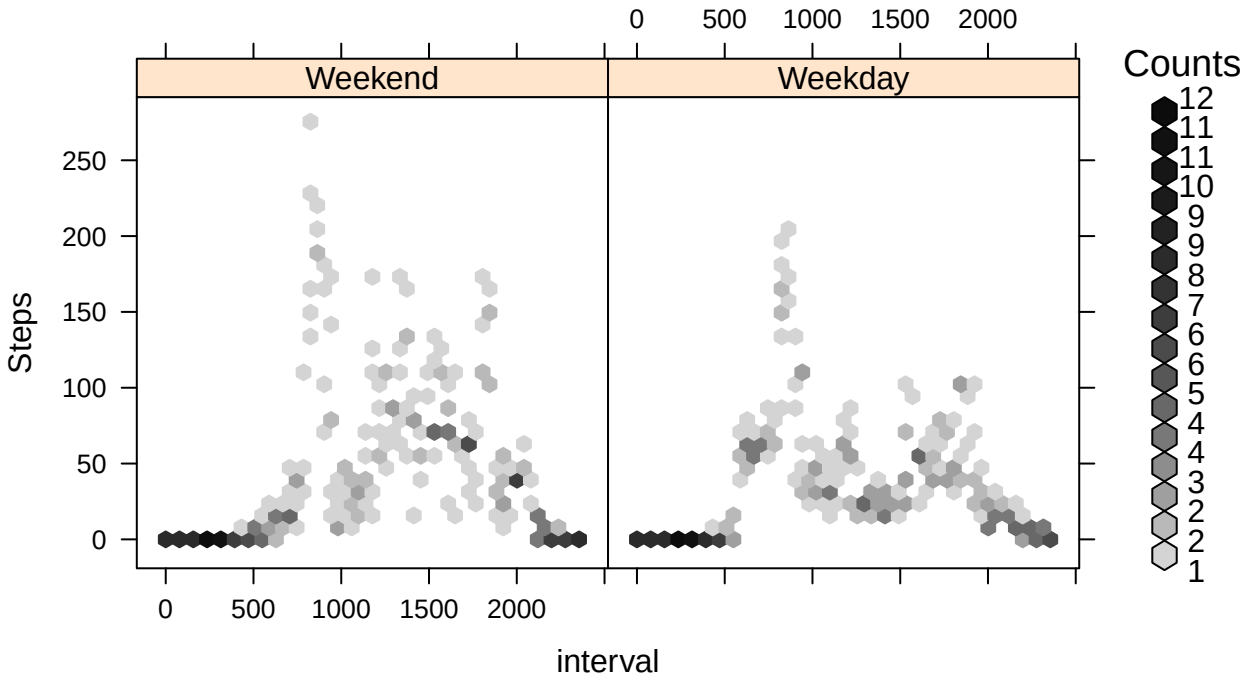
```
hexbinplot(data=q8.1,Steps~interval|WDWE, aspect = 1, bins=50)
```




```
xyplot(data=q8.1,Steps~interval|WDWE, grid = TRUE, type = c("p", "smooth"), lwd = 4,panel = panel.smoothl
```




```
hexbinplot(data=q8.1,Steps~interval|WDWE, aspect = 1, bins=50)
```


Weekend
Weekday
Steps
interval
Counts
0
500
1000
1500
2000
250
200
150
100
50
0
12
11
11
10
9
9
8
7
6
6
5
4
4
3
2
2
1