Course Project Reproducible Research

alvarorp22

24/8/2020

##Assignment Instructions 1.Code for reading in the dataset and/or processing the data 2.Histogram of the total number of steps taken each day 3.Mean and median number of steps taken each day 4.Time series plot of the average number of steps taken 5.The 5-minute interval that, on average, contains the maximum number of steps 6.Code to describe and show a strategy for imputing missing data 7.Histogram of the total number of steps taken each day after missing values are imputed 8.Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends 9.All of the R code needed to reproduce the results (numbers, plots, etc.) in the report

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

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

Exploring the basics of this data

```r
dim(activity)
```

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

```r
names(activity)
```

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

```r
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
```

```r
str(activity)
```

```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : chr  "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

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

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

```
#transforming the date column into date format using lubridate
library(lubridate)
```

```
##
## Attaching package: ’lubridate’
```

```
## The following objects are masked from ’package:base’:
##
##     date, intersect, setdiff, union
```

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

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

##Step 2 ##Histogram of the total number of steps taken each day

```
library(ggplot2)
Q2<-data.frame(tapply(activity$steps,activity$date,sum,na.rm=TRUE))
Q2$date<-rownames(Q2)
rownames(Q2)<-NULL
names(Q2)[[1]]<-"Total Steps"
png("plot1.png")
#Total Steps by date bar chart
ggplot(Q2,aes(y=Q2$‘Total Steps‘,x=Q2$date))+geom_bar(stat="identity") + ylab("Total Steps")+xlab("Date
```

```
## Warning: Use of ‘Q2$date‘ is discouraged. Use ‘date‘ instead.
```

```
## Warning: Use of ‘Q2$‘Total Steps‘‘ is discouraged. Use ‘Total Steps‘ instead.
```

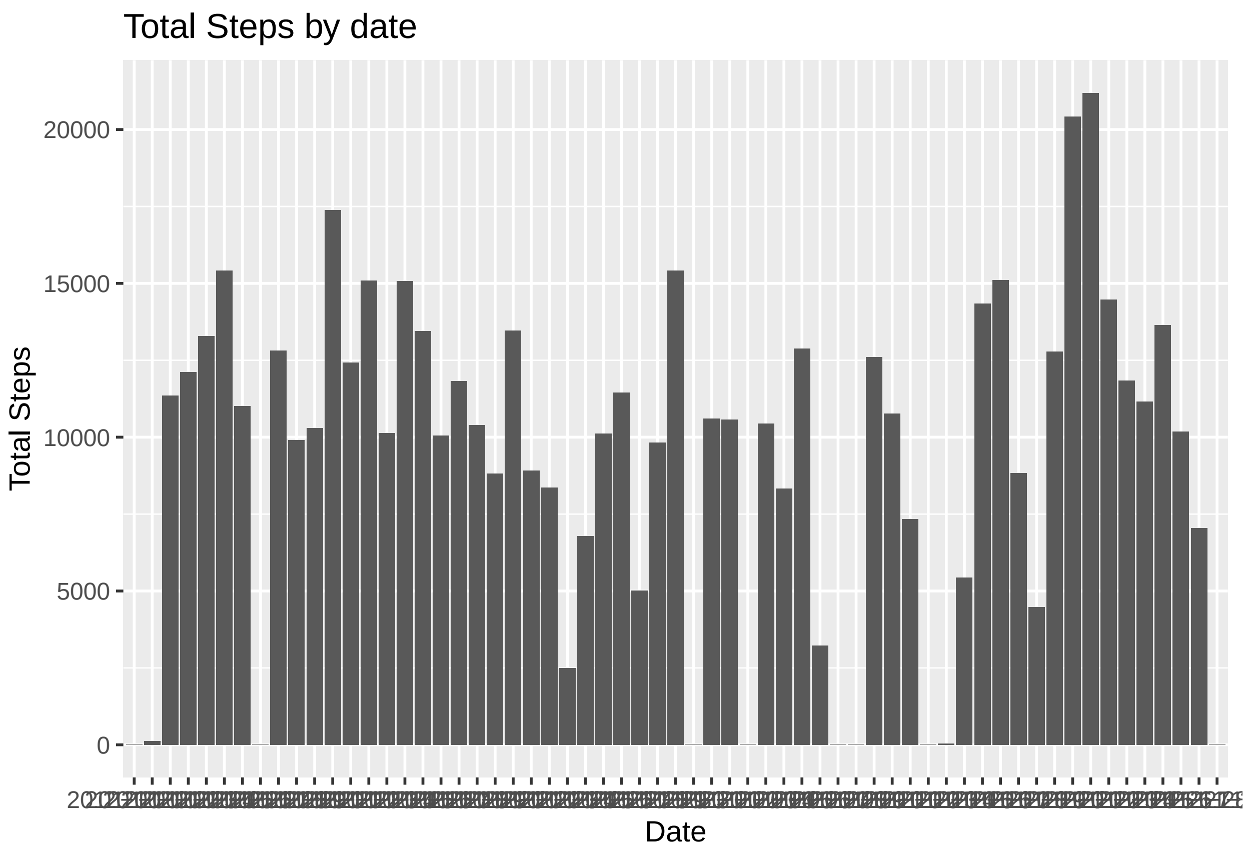
```
dev.off()
```

```
## pdf
##   2
```

```
ggplot(Q2,aes(y=Q2$‘Total Steps‘,x=Q2$date))+geom_bar(stat="identity") + ylab("Total Steps")+xlab("Date
```

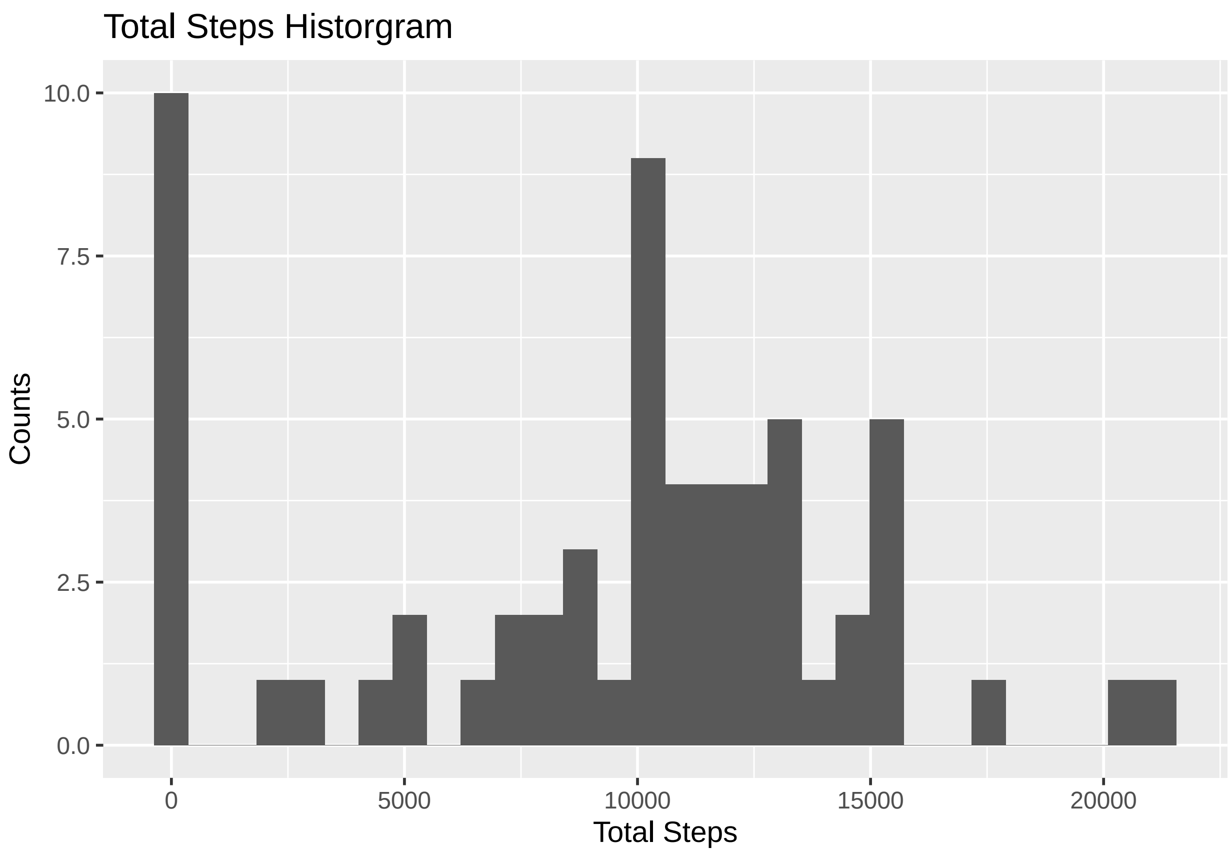
```
## Warning: Use of ‘Q2$date‘ is discouraged. Use ‘date‘ instead.
```

```
## Warning: Use of ‘Q2$‘Total Steps‘‘ is discouraged. Use ‘Total Steps‘ instead.
```



```
#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`.
```



```
png("plot1.1.png")
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`.
```

```
dev.off()
```

```
## pdf
##   2
```

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

```
library(dplyr)
```

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

```
## 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))
```

```
## ‘summarise()‘ ungrouping output (override with ‘.groups‘ argument)
```

```
names(temp)[[2]]<-"Median Steps"
Q3$median<-temp$‘Median Steps‘
Q3<-Q3 %>% select(date,‘Mean Steps‘,median)
```

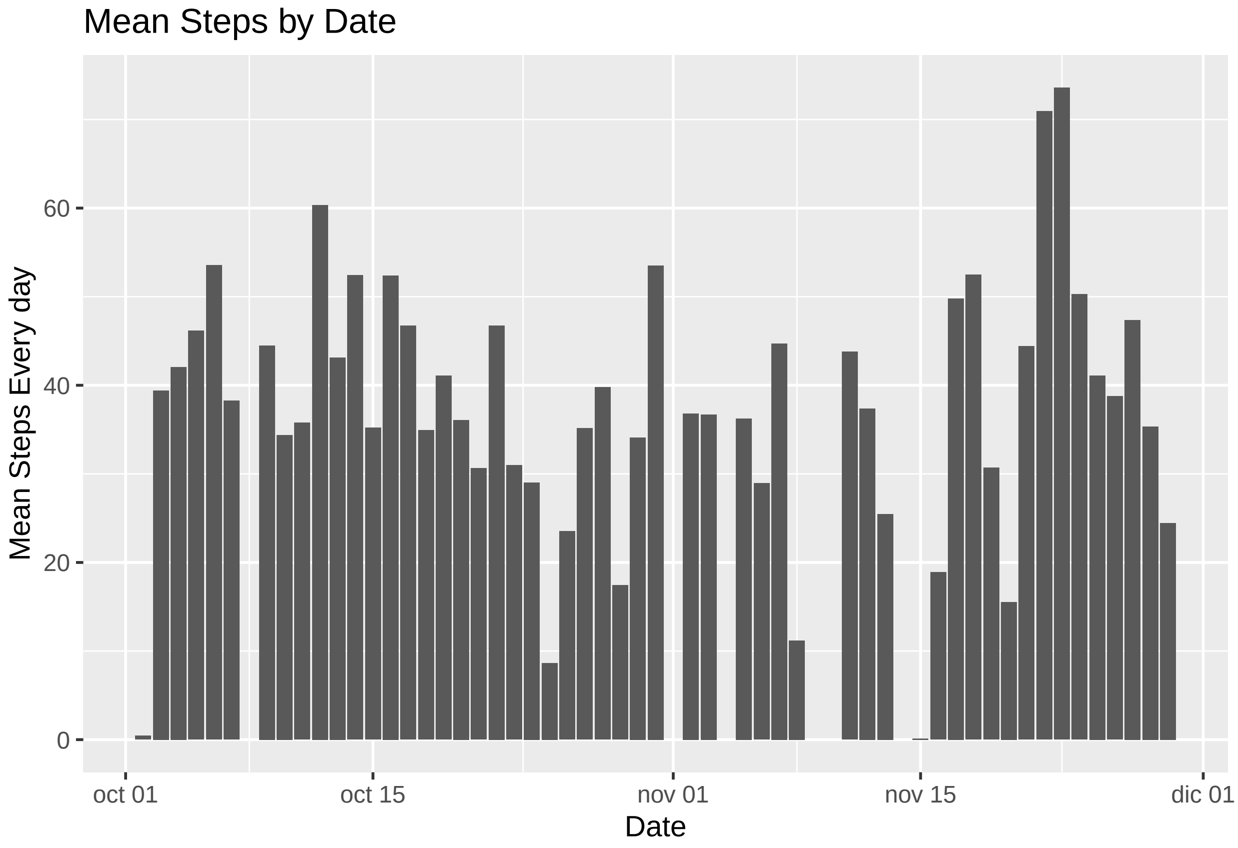
##Step 4 ##Time series plot of the average number of steps taken

```
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: Use of ‘Q4$date‘ is discouraged. Use ‘date‘ instead.

## Warning: Use of ‘Q4$‘Mean Steps‘‘ is discouraged. Use ‘Mean Steps‘ instead.

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



```
png("plot4.png")
ggplot(Q4,aes(x=Q4$date,y=Q4$‘Mean Steps‘))+geom_bar(stat="identity")+scale_x_date()+ylab("Mean Steps E
```

```
## Warning: Use of ‘Q4$date‘ is discouraged. Use ‘date‘ instead.
```

```
## Warning: Use of ‘Q4$‘Mean Steps‘‘ is discouraged. Use ‘Mean Steps‘ instead.
```

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

```
dev.off()
```

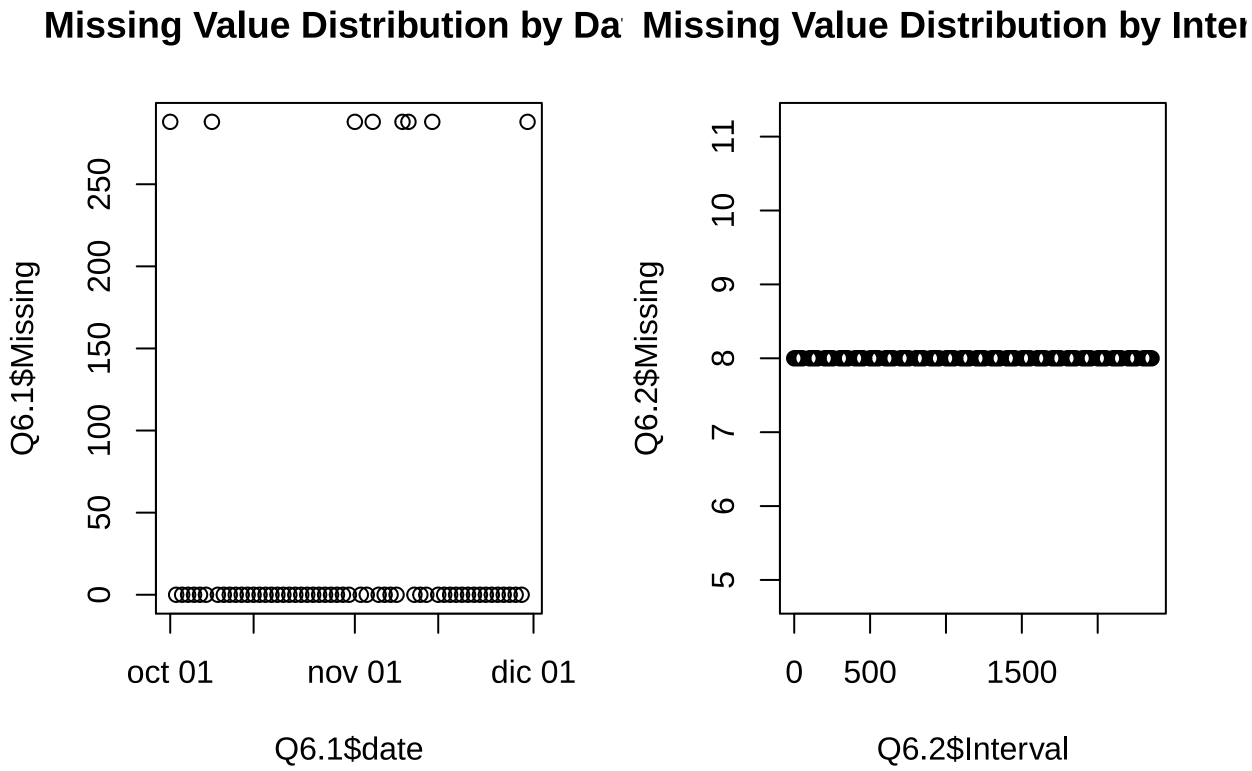
```
## pdf
##   2
```

##Step 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")
```

##Step 6 Code to describe and show a strategy for imputing missing data There are multiple strategies to deal with multiple value imputations. The common strategies include: 1. Constant value imputations 2. Regression model value imputations 3. Mean/mode value substitutions For the purpose of simplicity, in this question, I will use the mean/mode value substitution strategy to impute missing values. That is, using the mean values to substitute out the missing values in the original data set Before doing any sort of imputation, it is helpful to understand what are the distributions of missing values by date and interval

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

By this point, from the plot, that the missing values have a very disctinct pattern. For every interval, there are consistantly 8 missing values. For the date, there are consistantly 288 missing values. And in total, there are 8 dates that have missing value. We don't exactly know the cause for these missing values but there's a pattern. For that matter, we can see that the mean value imputation is appropriate.

We can see that every date has 288 data points. It means that the 8 dates have no data points at all what so ever. We can refine the analysis by looking at 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$date,label=TRUE)
#Finding the mean of steps every monday, and every interval
Q6.5<-aggregate(data=Q6.4,steps~interval+weekday,FUN="mean",na.rm=TRUE)
#Merge the pre-imputation table Q6.4 table with the average table Q6.5
Q6.6<-merge(x=Q6.4,y=Q6.5,by.x=c("interval","weekday"),by.y=c("interval","weekday"),all.x=TRUE)
#Conditionally replacing the steps.x column NA value with the values from steps.y column value
Q6.6$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

```
png("plot7.png")
qplot(Q6.6$Steps,geom="histogram",main="Total steps taken histogram post imputation",xlab="Steps",ylab=
```

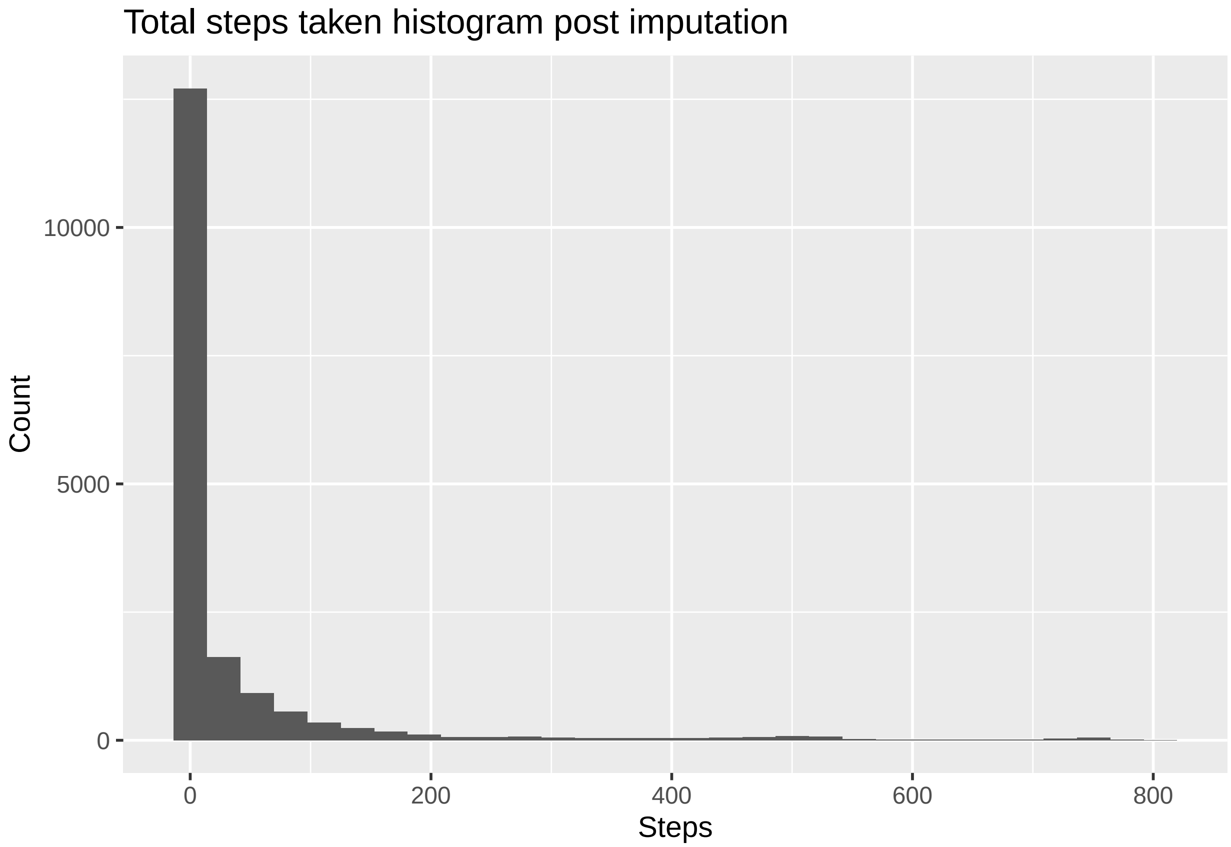
```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```

```
dev.off()
```

```
## pdf
##   2
```

```
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‘.
```



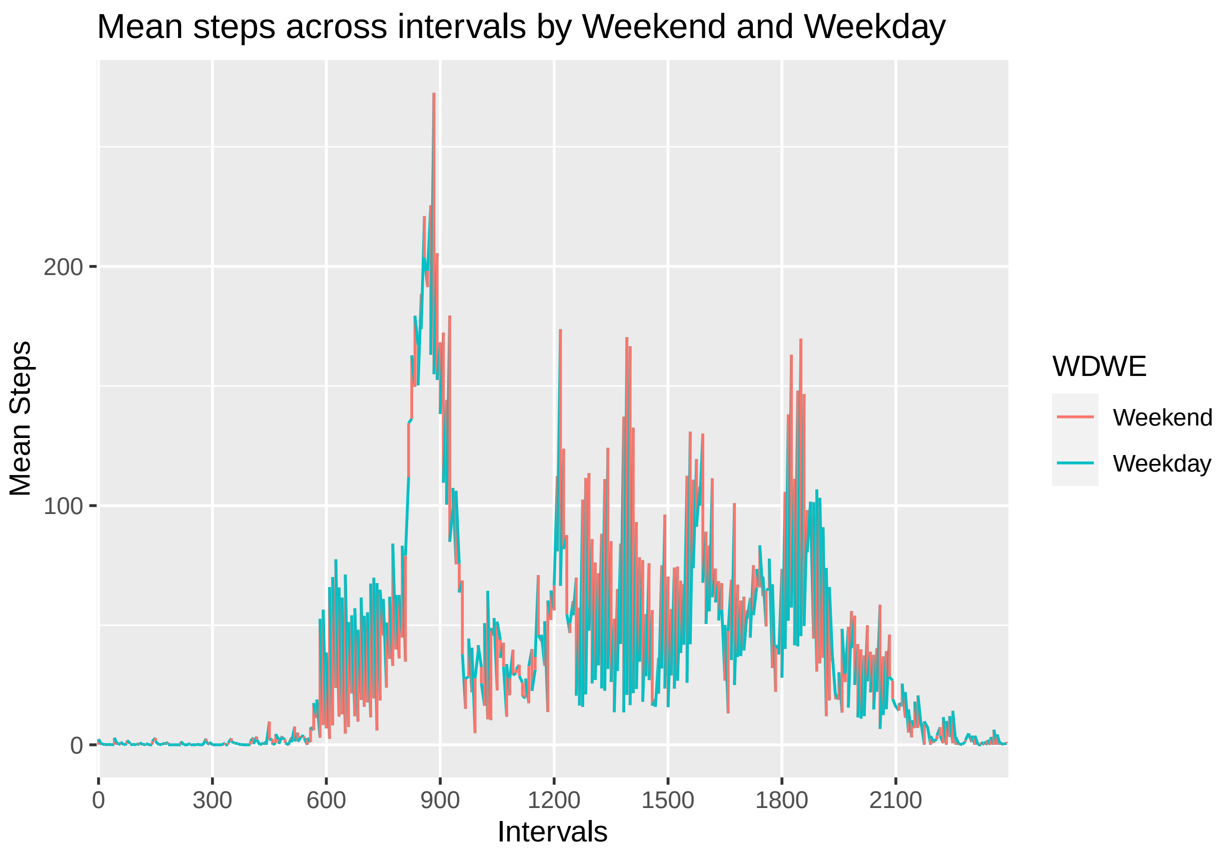
Step 8

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")
png("plot8.png")
ggplot(data=Q8.1,aes(y=Steps,x=interval,group=1,color=WDWE))+geom_line() +scale_x_discrete(breaks = seq
dev.off()
```

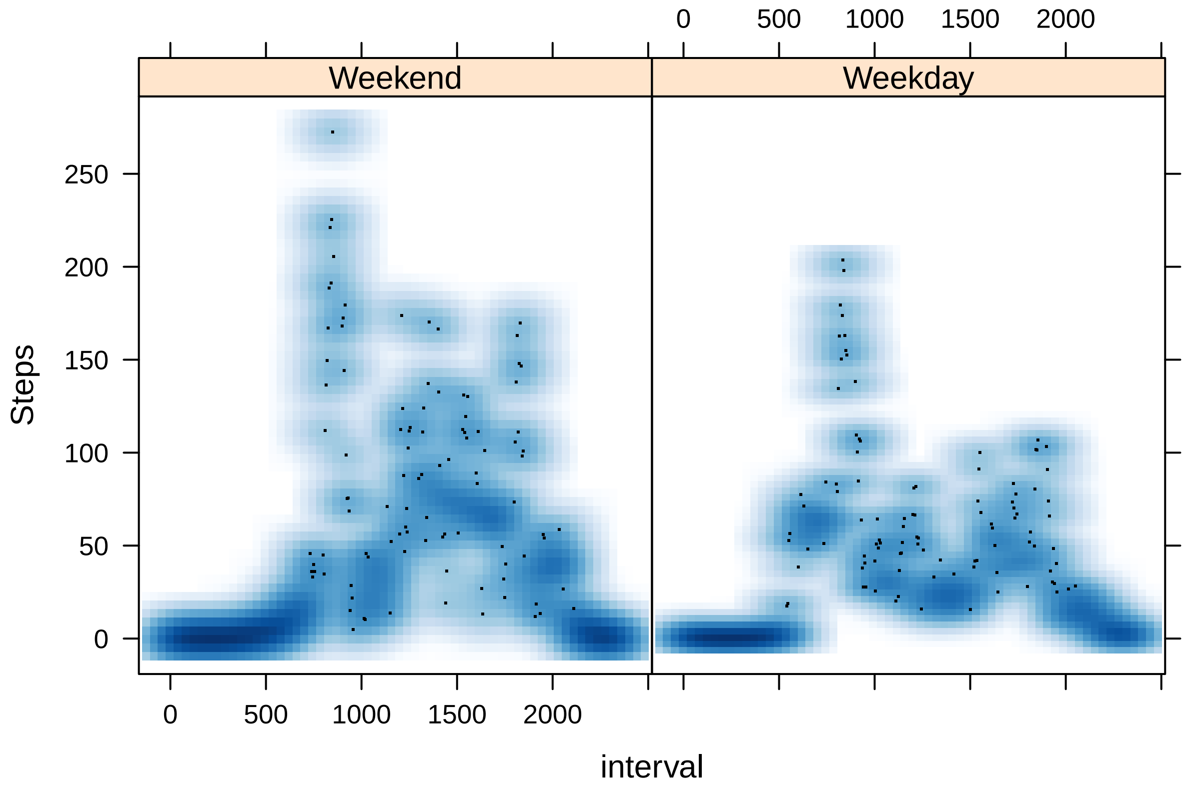
```
## pdf
##   2
```

```
ggplot(data=Q8.1,aes(y=Steps,x=interval,group=1,color=WDWE))+geom_line() +scale_x_discrete(breaks = seq
```

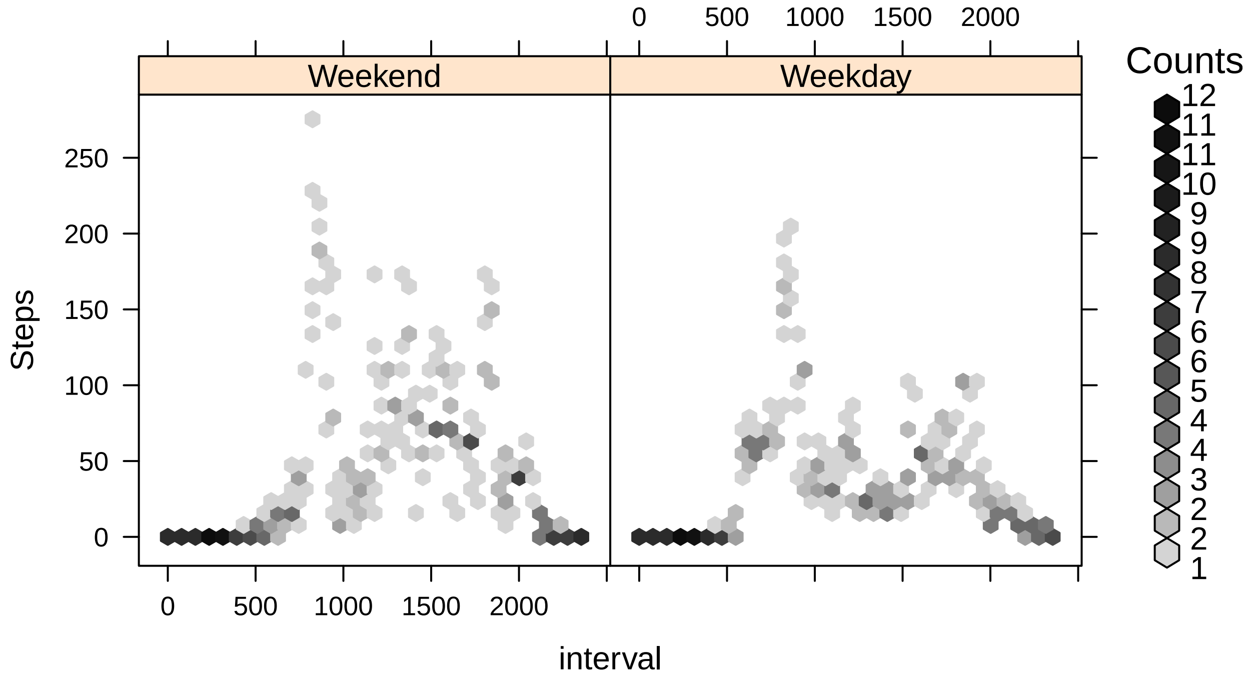


```
#Producing the panel plot
Q8.1$interval<-as.numeric(as.character(Q8.1$interval))
library(lattice)
xyplot(data=Q8.1,Steps~interval|WDWE, grid = TRUE, type = c("p", "smooth"), lwd = 4,panel = panel.smoot
```

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



```
library(hexbin)
hexbinplot(data=Q8.1,Steps~interval|WDWE, aspect = 1, bins=50)
```



```
png("plott8.1.png")
xyplot(data=Q8.1,Steps~interval|WDWE, grid = TRUE, type = c("p", "smooth"), lwd = 4,panel = panel.smoot
dev.off()
```

```
## pdf
##   2
```

```
png("plot8.2.png")
hexbinplot(data=Q8.1,Steps~interval|WDWE, aspect = 1, bins=50)
dev.off()
```

```
## pdf
##   2
```