Project1

Tong Li

July 17, 2016

Download and read in data

```
fileurl<-"https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
download.file(fileurl, "assign.zip")
unzip("assign.zip")
data<-read.csv("activity.csv", header=TRUE)
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
```

```
summary(data)
```

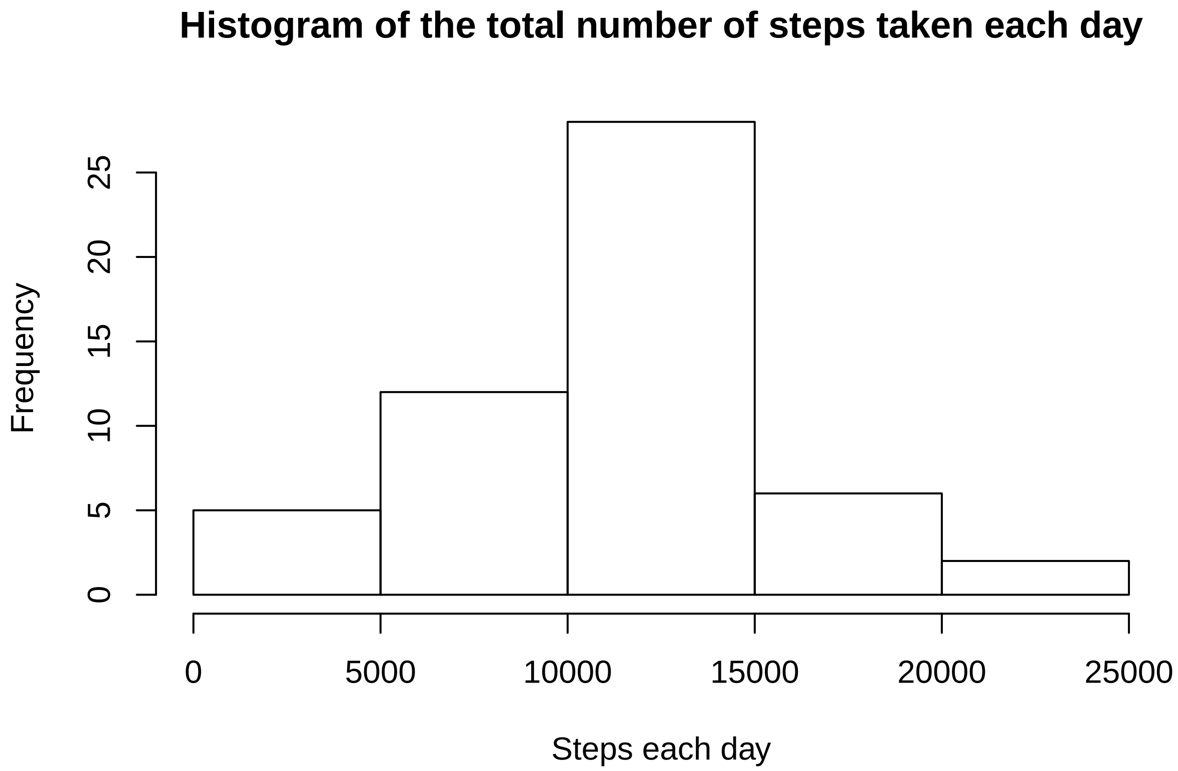
```
##      steps                 date          interval
##  Min.   :  0.00   2012-10-01:  288   Min.   :   0.0
##  1st Qu.:  0.00   2012-10-02:  288   1st Qu.: 588.8
##  Median :  0.00   2012-10-03:  288   Median :1177.5
##  Mean   : 37.38   2012-10-04:  288   Mean   :1177.5
##  3rd Qu.: 12.00   2012-10-05:  288   3rd Qu.:1766.2
##  Max.   :806.00   2012-10-06:  288   Max.   :2355.0
##  NA's   :2304     (Other)   :15840
```

Use aggregate to summarize no. of steps each day

```
newdata<-aggregate(data$steps, list(data$date), sum)
names(newdata)[2] <- "steps"
names(newdata)[1] <- "date"
```

Histogram of the total number of steps taken each day

```
hist(newdata$steps, xlab='Steps each day',
     main='Histogram of the total number of steps taken each day')
```



Calculate mean and median number of steps taken each day

```r
meansteps<-mean(newdata$steps, na.rm=TRUE)
mediansteps<-median(newdata$steps, na.rm=TRUE)

print(paste("meansteps =", meansteps))
```

```
## [1] "meansteps = 10766.1886792453"
```

```r
print(paste("mediansteps =", mediansteps))
```

```
## [1] "mediansteps = 10765"
```

Time series plot of the average number of steps taken

```r
length(unique(data$interval))
```

```
## [1] 288
```

```
newdata2<-aggregate(data$steps, list(data$interval), mean, na.rm=TRUE)
head(newdata2)
```

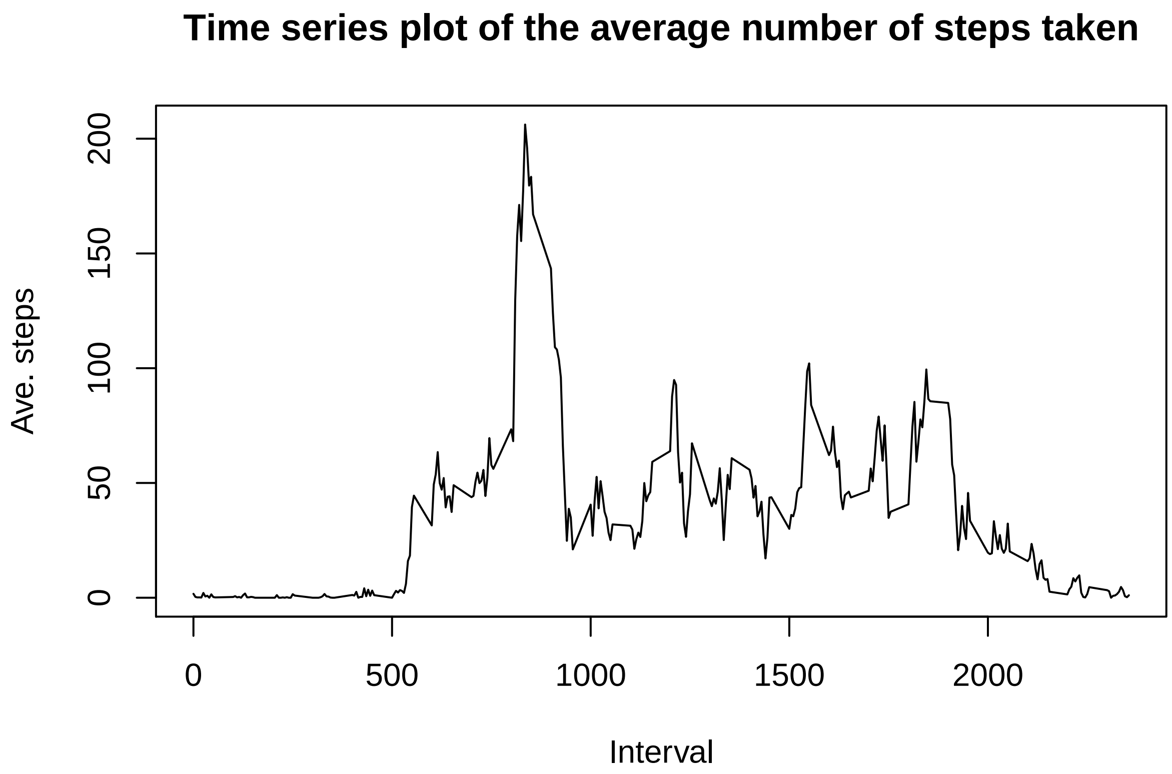
```
##   Group.1         x
## 1       0 1.7169811
## 2       5 0.3396226
## 3      10 0.1320755
## 4      15 0.1509434
## 5      20 0.0754717
## 6      25 2.0943396
```

```
str(newdata2)
```

```
## 'data.frame':    288 obs. of  2 variables:
##  $ Group.1: int  0 5 10 15 20 25 30 35 40 45 ...
##  $ x      : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
```

```
names(newdata2)[2] <- "steps"
names(newdata2)[1] <- "interval"

plot(newdata2$interval, newdata2$steps, xlab='Interval',
     ylab='Ave. steps', type='l',
     main='Time series plot of the average number of steps taken')
```



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

```
maxstep<-max(newdata2$steps)
max_interval<-newdata2$interval[which(newdata2$steps==maxstep)]
print(paste("The 5-minute interval that contains the max No. of steps is", max_interval))
```

```
## [1] "The 5-minute interval that contains the max No. of steps is 835"
```

Code to describe and show a strategy for imputing missing data

total numbers of missing values

```
missing<-sum(is.na(data$steps))
print(paste("Total No. of missing values is", missing))
```

```
## [1] "Total No. of missing values is 2304"
```

replace NAs with interval mean of steps

Use ave function to replace NAs with interval mean

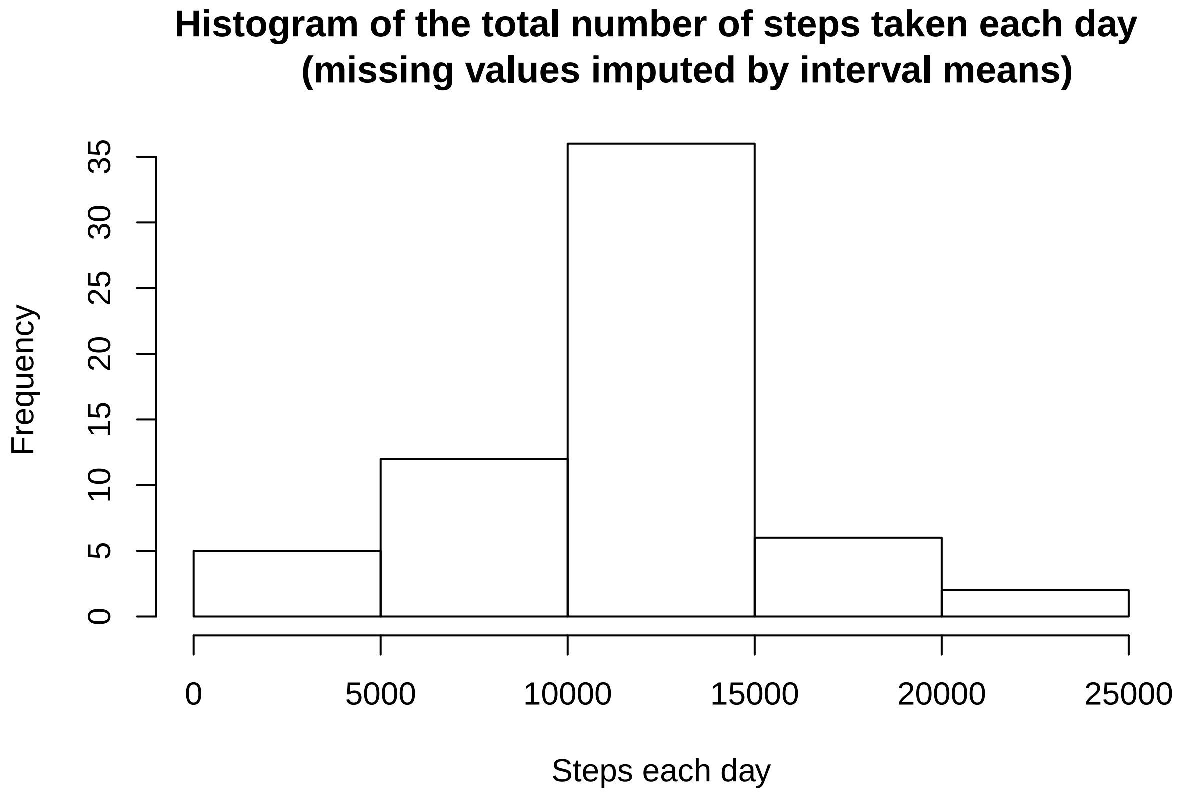
```
data2<-data
data2$steps[is.na(data2$steps)] <- ave(data2$steps, data2$interval,
                FUN=function(x) mean(x,na.rm = T))[is.na(data2$steps)]

newdata3<-aggregate(data2$steps, list(data2$date), sum)

names(newdata3)[2] <- "steps"
names(newdata3)[1] <- "date"
```

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

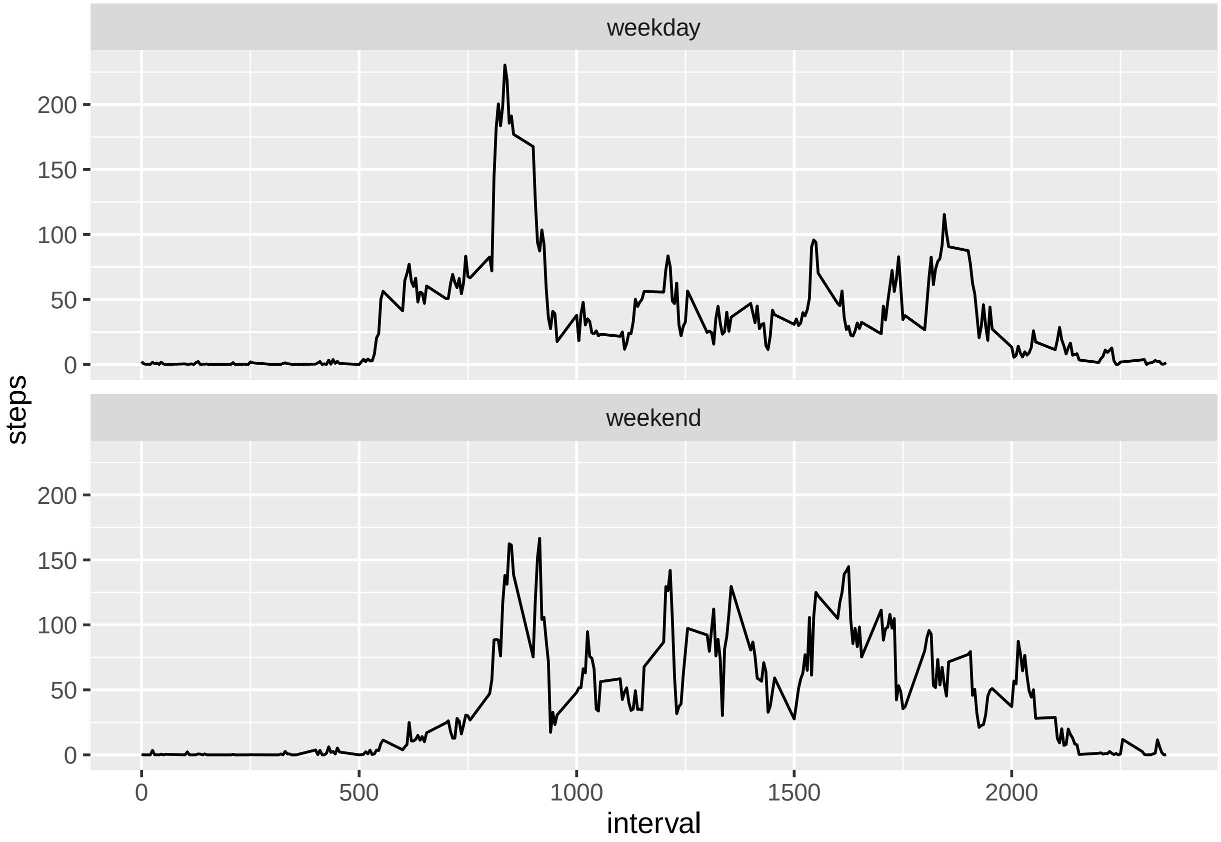
```
hist(newdata3$steps, xlab='Steps each day',
     main='Histogram of the total number of steps taken each day
     (missing values imputed by interval means)')
```



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

```
library(ggplot2)
data2$day <- weekdays(as.Date(data2$date))
data2["daytype"]<-"weekday"
data2$daytype[which(data2$day=="Saturday" | data2$day=="Sunday")]<-"weekend"

new<-aggregate(steps~daytype + interval, data2, mean)
ggplot(new, aes(x=interval, y=steps))+geom_line()+facet_wrap(~daytype, nrow=2)
```


weekday
weekend
steps
200
150
100
50
0
200
150
100
50
0
0
500
1000
1500
2000
interval