PeerAssessment1

Loading the data

- Load the data

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

- Process/transform the data(if necessary) into a format suitable for your analysis

```
totalSteps<-aggregate(steps~date,data=activity,sum,na.rm=TRUE)
```

The mean total number of steps took per day.

- Make a histogram of the total number of steps taken each day

```
hist(totalSteps$steps)
```

- Calculate and report the **mean** and **median** total number of steps taken per day

```
mean(totalSteps$steps)
```

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

```
median(totalSteps$steps)
```

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

- The **mean** total number of steps taken per day is 1.0766189 $\times$ 104 steps.
- The **median** total number of steps taken per day is 10765 steps.

The average daily activity pattern

- Make a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis)

```
stepsInterval <- aggregate(steps ~ interval, data = activity, mean, na.rm=TRUE)
plot(steps ~ interval, data = stepsInterval, type="l")
```

- Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps?

```
stepsInterval[which.max(stepsInterval$steps),]$interval
```

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

- Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated.

: I used a strategy for filing in all of the missing values with the mean for that 5-minute interval. First of all, I made a function **"interval2steps"** to get the mean steps for particular 5-minute interval.

```
interval2steps<-function(interval){
    stepsInterval[stepsInterval$interval==interval,]$steps
}
```

- Create a new dataset that is equal to the original dataset but with the missing data filled in.

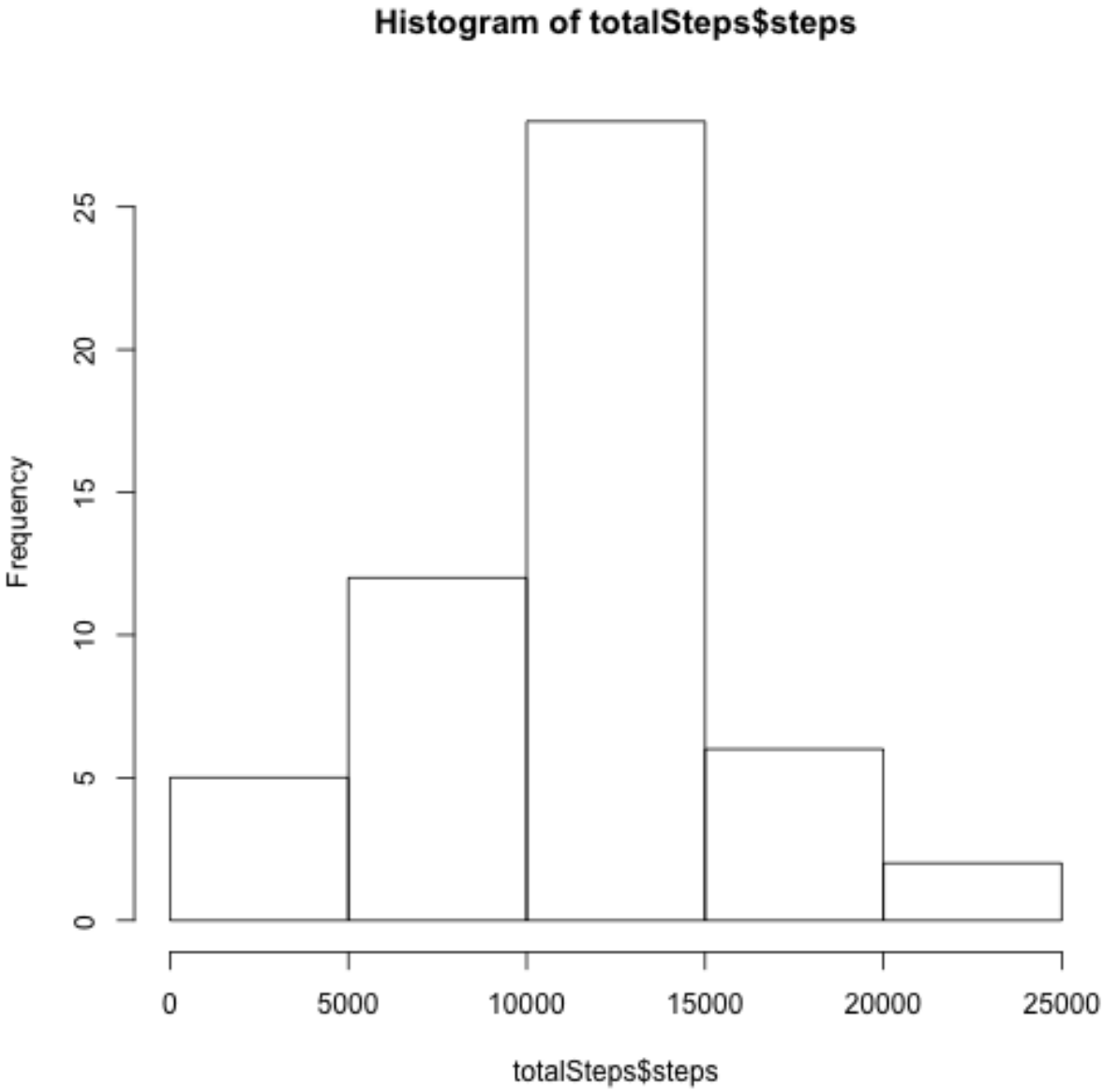

Figure 1: plot of chunk unnamed-chunk-8

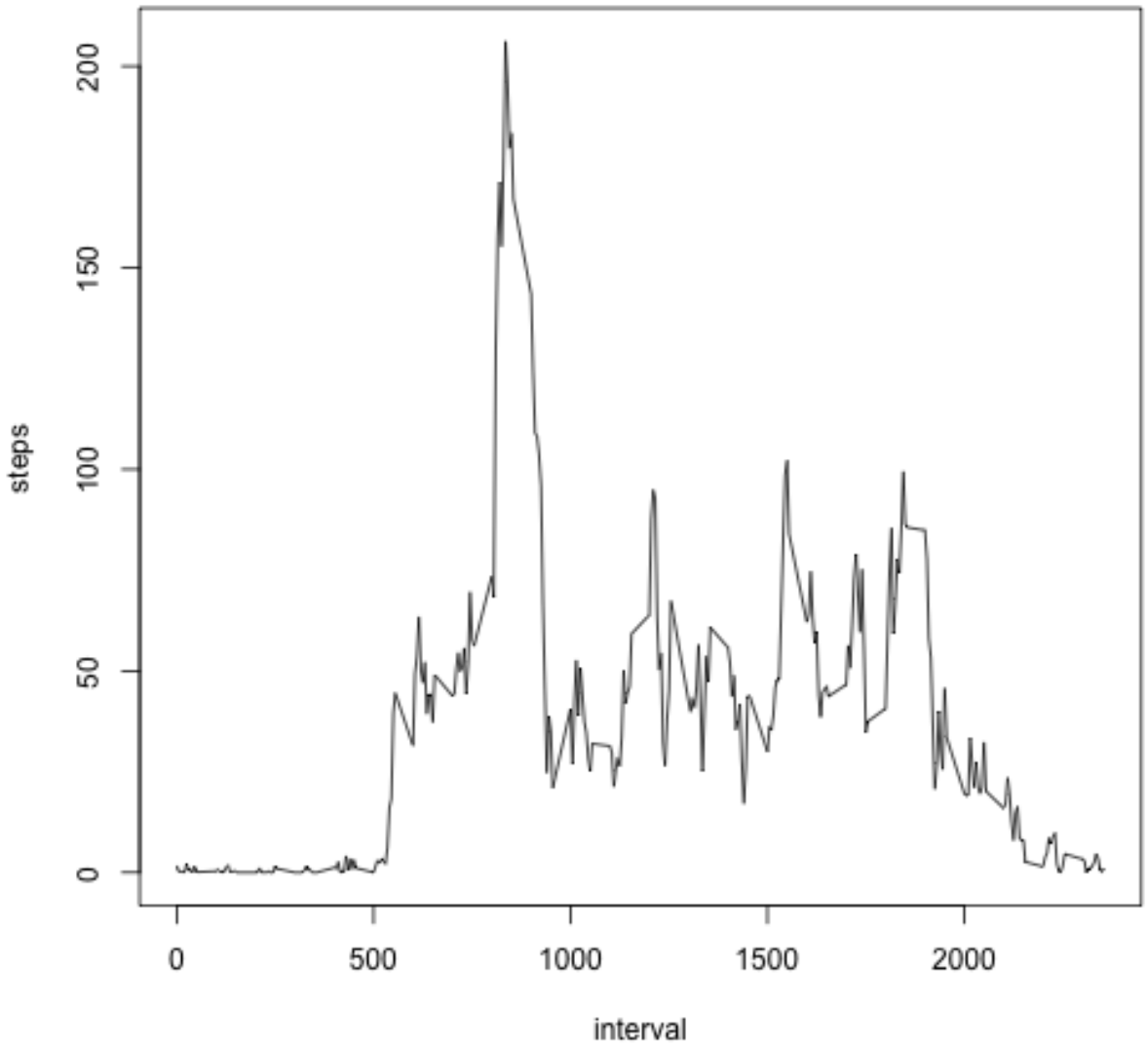

Figure 2: plot of chunk unnamed-chunk-10

```
activityFilled<-activity    # Make a new dataset with the original data
count=0           # Count the number of data filled in
for(i in 1:nrow(activityFilled)){
    if(is.na(activityFilled[i,]$steps)){
        activityFilled[i,]$steps<-interval2steps(activityFilled[i,]$interval)
        count=count+1
    }
}
cat("Total ",count, "NA values were filled.\n\r")
```

```
## Total  2304 NA values were filled.
##
```

- Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day.

```
totalSteps2<-aggregate(steps~date,data=activityFilled,sum)
hist(totalSteps2$steps)
```

```
mean(totalSteps2$steps)
```

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

```
median(totalSteps2$steps)
```

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

- The **mean** total number of steps taken per day is 1.0766189 × 104 steps.
- The **median** total number of steps taken per day is 1.0766189 × 104 steps.
- Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps?

: The **mean** value is the **same** as the value before imputing missing data because we put the mean value for that particular 5-min interval. The median value shows **a little** difference : but it depends on **where the missing values are**.

Are there differences in activity patterns between weekdays and weekends?

- Create a new factor variable in the dataset with two levels – "weekday" and "weekend" indicating whether a given date is a weekday or weekend day.

```
activityFilled$day=ifelse(as.POSIXlt(as.Date(activityFilled$date))$wday%%6==0,
                          "weekend","weekday")
# For Sunday and Saturday : weekend, Other days : weekday
activityFilled$day=factor(activityFilled$day,levels=c("weekday","weekend"))
```

- Make a panel plot containing a time series plot (i.e. type = "l") of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis). The plot should look something like the following, which was creating using simulated data:

```
stepsInterval2=aggregate(steps~interval+day,activityFilled,mean)
library(lattice)
xyplot(steps~interval|factor(day),data=stepsInterval2,aspect=1/2,type="l")
```

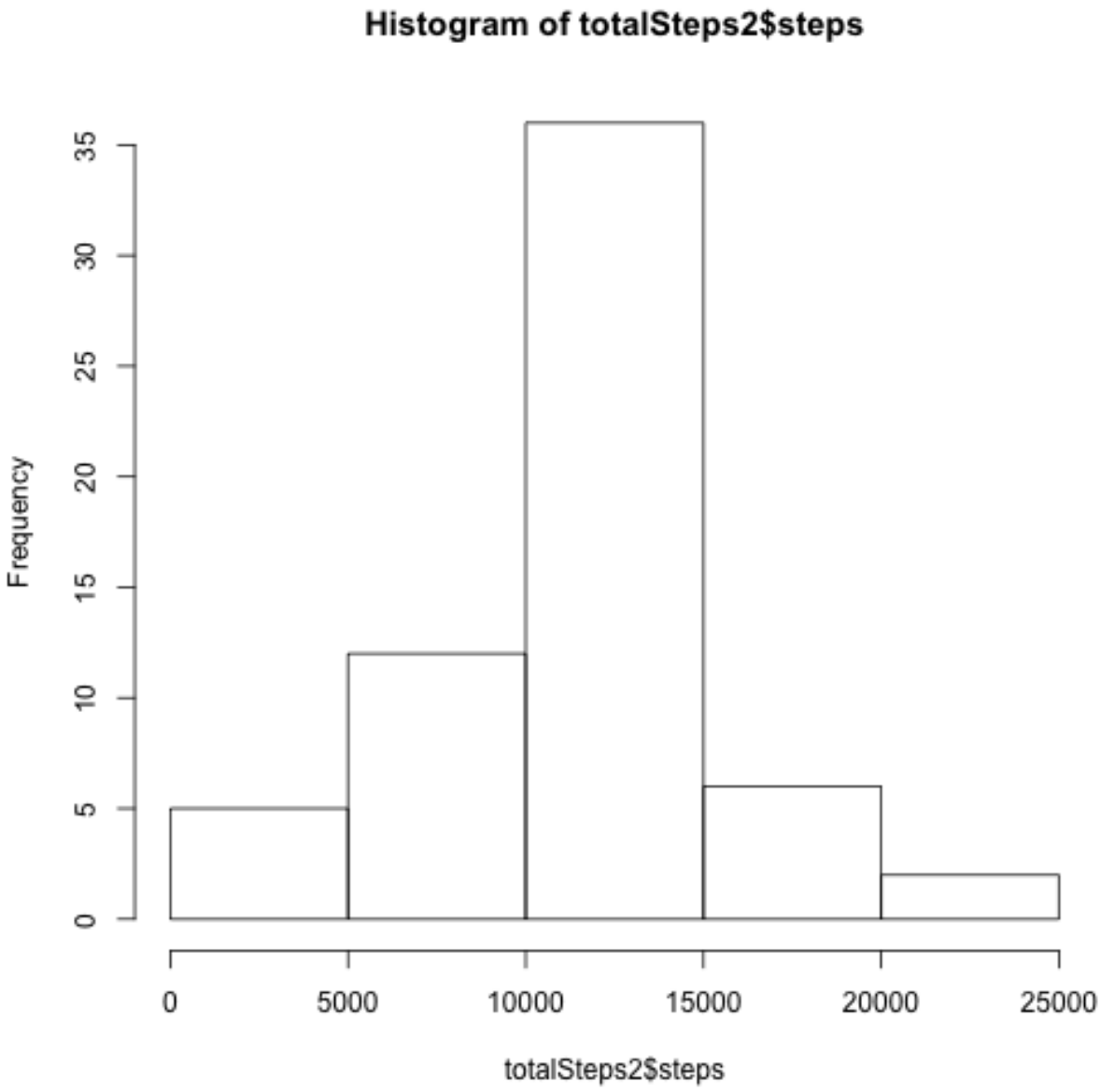

Figure 3: plot of chunk unnamed-chunk-14

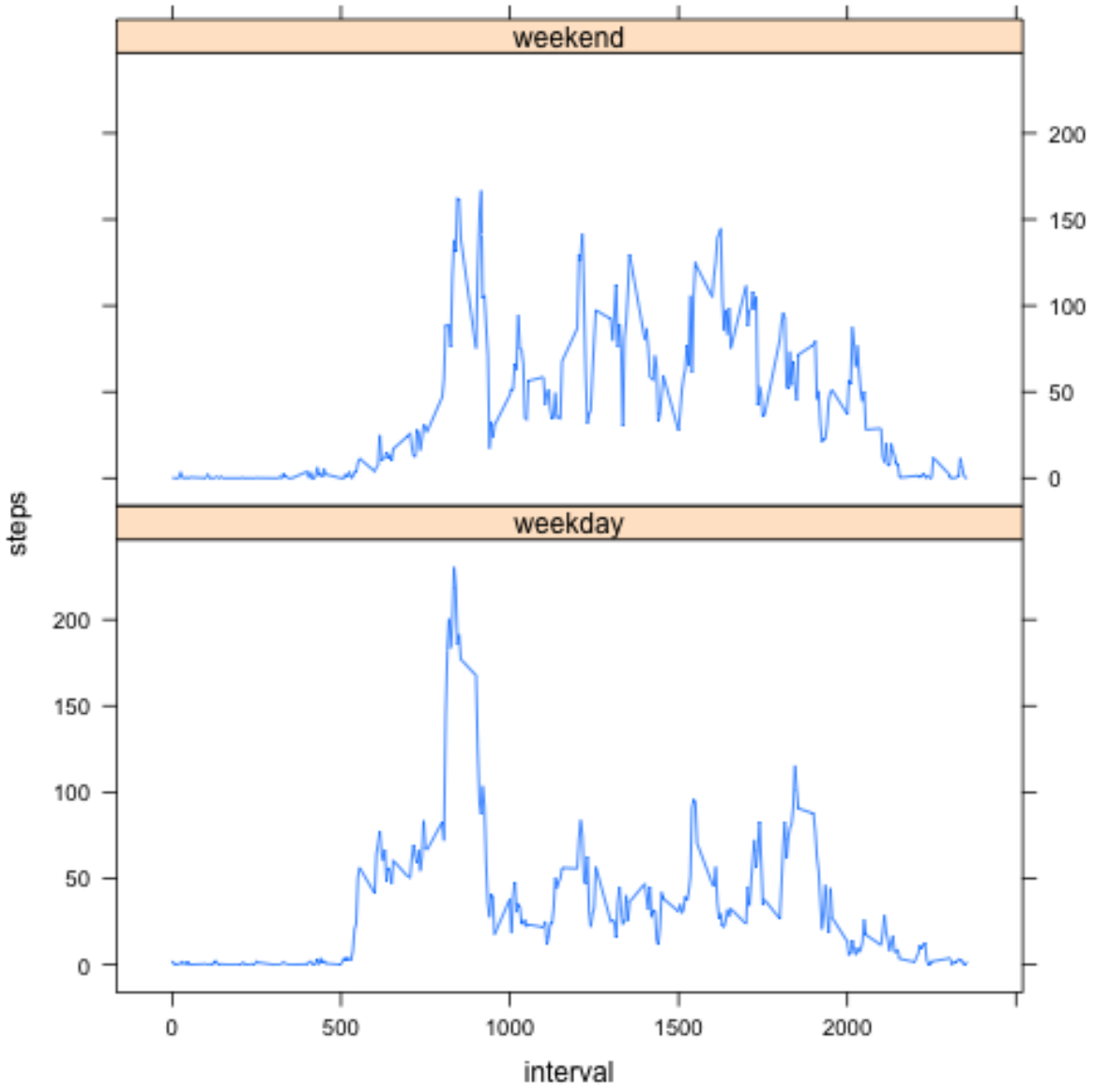

Figure 4: plot of chunk unnamed-chunk-16

Imputing missing values

- Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)

```
sum(is.na(activity$steps))
```

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