Assignment

Abhijeet Sharma

07/12/2020

```
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
```

```
library(ggplot2)
doc <- read.csv("activity.csv")
act.complete <- na.omit(doc)
```

###Total Number of Steps Taken Per Day

```
act.day <- group_by(act.complete, date)
act.day <- summarize(act.day, steps=sum(steps))
```

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

```
summary(act.day)
```

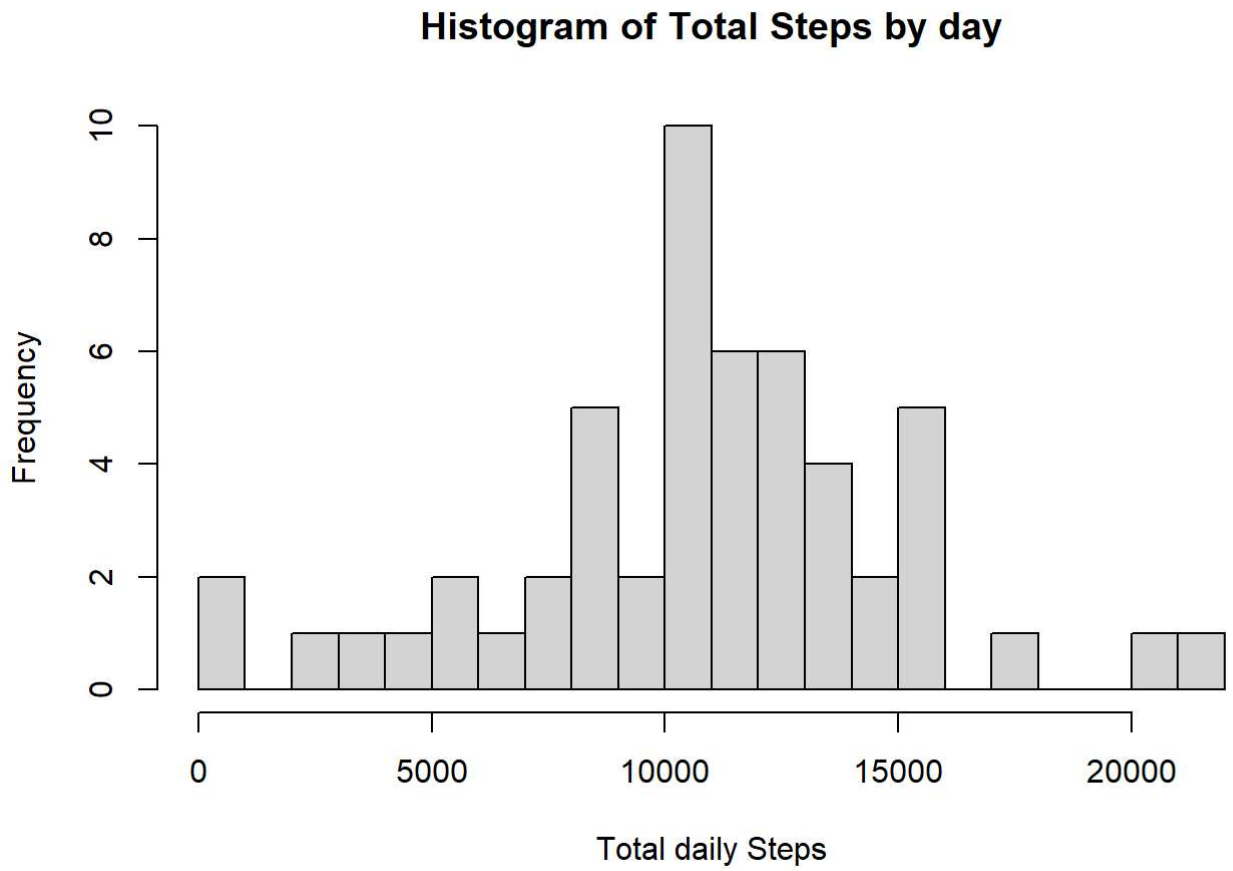
```
##      date               steps
##  Length:53          Min.   :   41
##  Class :character   1st Qu.: 8841
##  Mode  :character   Median :10765
##                     Mean   :10766
##                     3rd Qu.:13294
##                     Max.   :21194
```

```
head(act.day)
```

```
## # A tibble: 6 x 2
##   date       steps
##   <chr>      <int>
## 1 2012-10-02   126
## 2 2012-10-03 11352
## 3 2012-10-04 12116
## 4 2012-10-05 13294
## 5 2012-10-06 15420
## 6 2012-10-07 11015
```

####Histogram

```
hist(act.day$steps, xlab = "Total daily Steps",main="Histogram of Total Steps by day", breaks = 20)
```



####Mean and Median of the total number of steps taken per day

```
mean(act.day$steps)
```

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

```
median(act.day$steps)
```

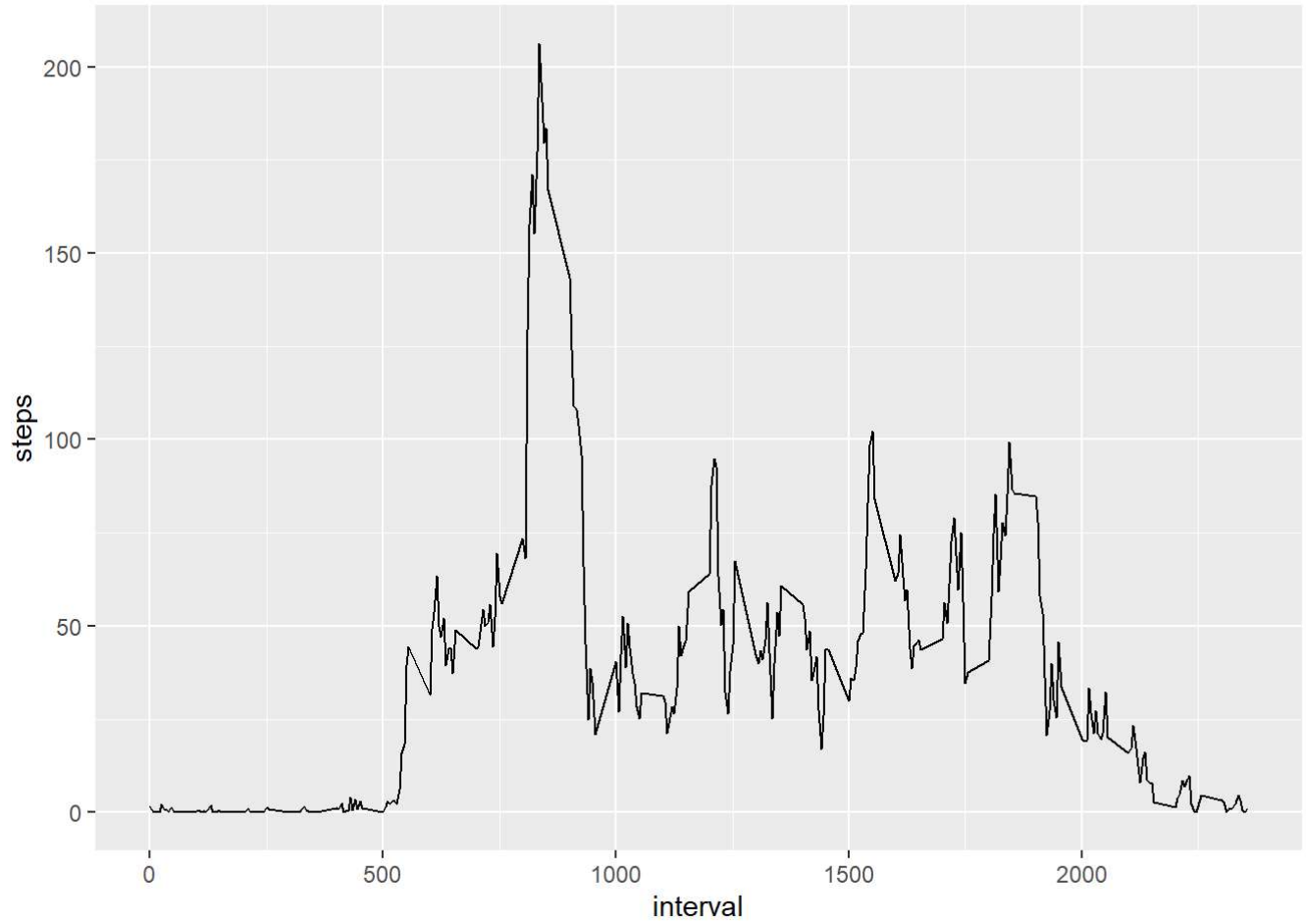
```
## [1] 10765
```

####Average Daily Activity Pattern

```
act.int <- group_by(act.complete, interval)
act.int <- summarize(act.int, steps=mean(steps))
```

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

```
ggplot(act.int, aes(interval, steps)) + geom_line()
```




```
act.int[act.int$steps==max(act.int$steps),]
```

```
## # A tibble: 1 x 2
##   interval steps
##      <int> <dbl>
## 1      835  206.
```

###Imputing Missing Values

```
nrow(doc)-nrow(act.complete)
```

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

```
names(act.int)[2] <- "mean.steps"
act.impute <- merge(doc, act.int)
```

####New dataset that is equal to the original dataset but with the missing data filled in.

```
act.impute$steps[is.na(act.impute$steps)] <- act.impute$mean.steps[is.na(act.impute$steps)]
```

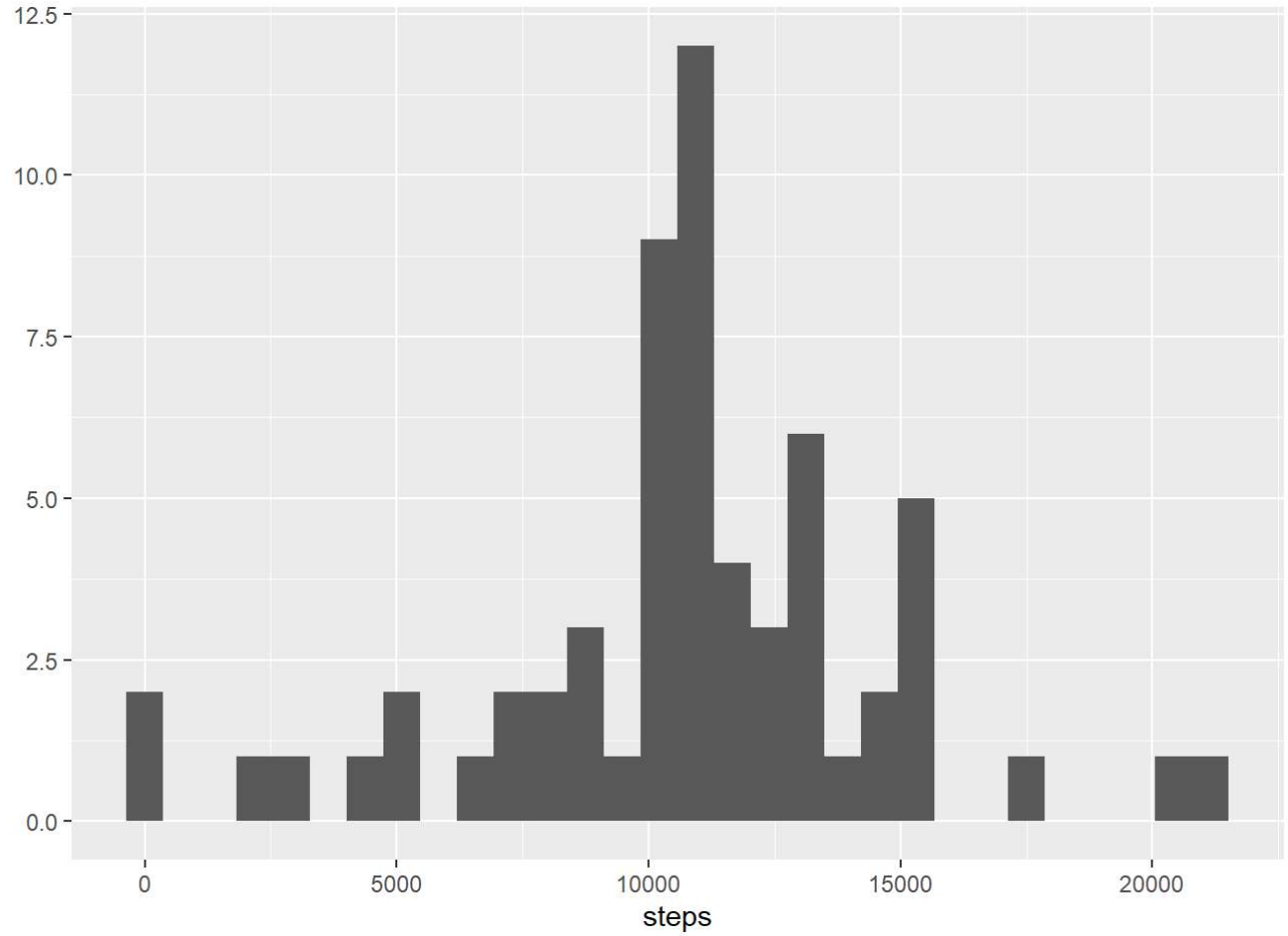
####Histogram of the total number of steps taken each day

```
act.day.imp <- group_by(act.impute, date)
act.day.imp <- summarize(act.day.imp, steps=sum(steps))
```

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

```
qplot(steps, data=act.day.imp)
```

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



###Differences in activity patterns between weekdays and weekends ####To 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.

```
act.impute$dayofweek <- weekdays(as.Date(act.impute$date))
act.impute$weekend <-as.factor(act.impute$dayofweek=="Saturday"|act.impute$dayofweek=="Sunday")
levels(act.impute$weekend) <- c("Weekday", "Weekend")
```

####Panel plot containing a time series plot First I create separate data frames for weekends and weekdays:

```
act.weekday <- act.impute[act.impute$weekend=="Weekday",]
act.weekend <- act.impute[act.impute$weekend=="Weekend",]
```

Then for each one, I find the mean number of steps across days for each 5 minute interval:

```
act.int.weekday <- group_by(act.weekday, interval)
act.int.weekday <- summarize(act.int.weekday, steps=mean(steps))
```

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

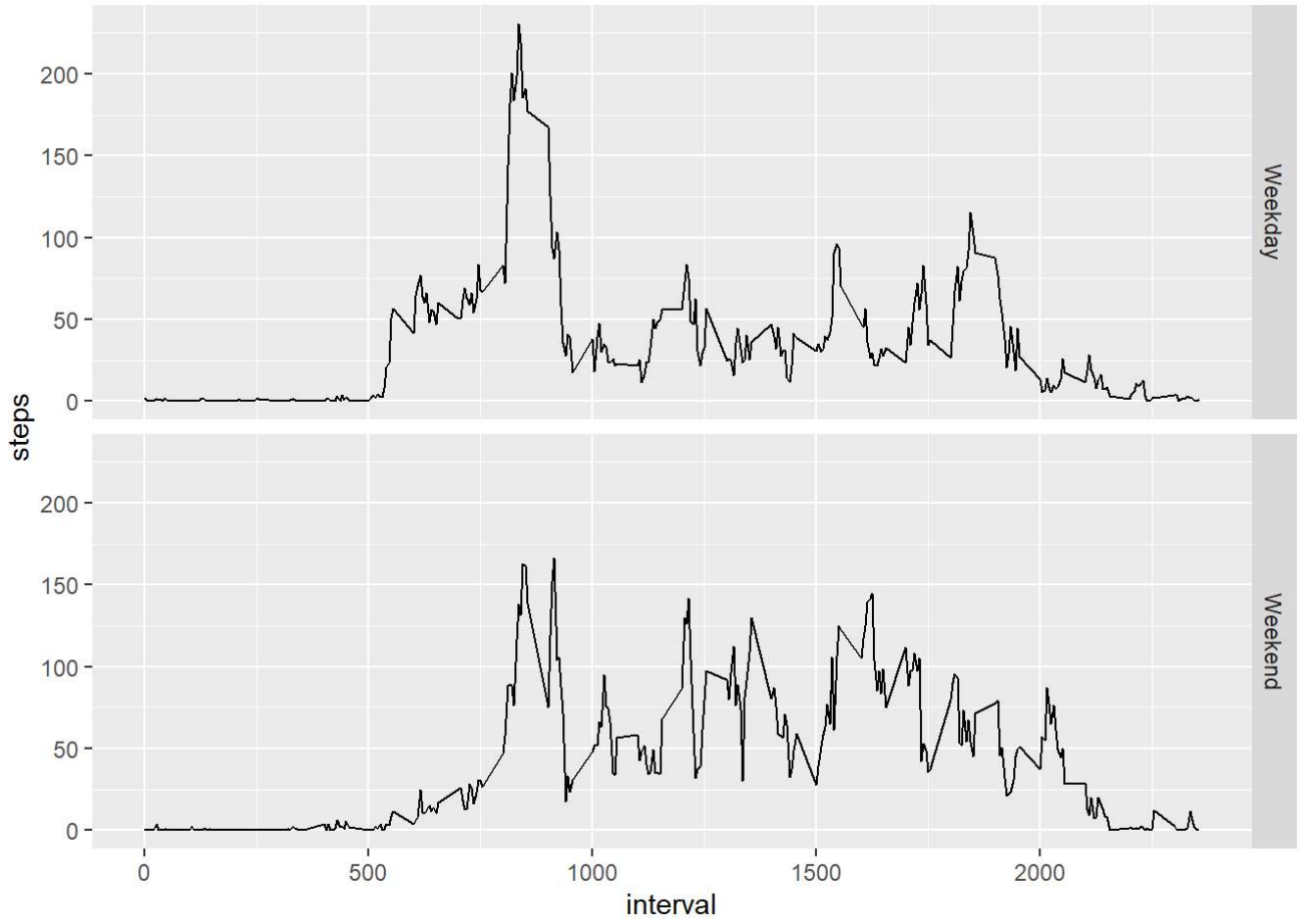
```
act.int.weekday$weekend <- "Weekday"
act.int.weekend <- group_by(act.weekend, interval)
act.int.weekend <- summarize(act.int.weekend, steps=mean(steps))
```

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

```
act.int.weekend$weekend <- "Weekend"
```

I append the two data frames together, and I make the two time series plots:

```
act.int <- rbind(act.int.weekday, act.int.weekend)
act.int$weekend <- as.factor(act.int$weekend)
ggplot(act.int, aes(interval, steps)) + geom_line() + facet_grid(weekend ~ .)
```


steps
200
150
100
50
0
Weekday
Weekend
0
500
1000
1500
2000
interval