Research_Project1

Reproducible Research: Peer Assessment 1

```
library(ggplot2)
```

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

```
library(scales)
```

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

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 3.3.2
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 3.3.2
```

```
## Loading required package: Formula
```

```
## 
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
## 
##     format.pval, round.POSIXt, trunc.POSIXt, units
```

```
library(knitr)
library(ggplot2)
library(dplyr)
```

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

```
## 
## Attaching package: 'dplyr'
## The following objects are masked from 'package:Hmisc':
## 
##     combine, src, summarize
```

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

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

```
library(plyr)
```

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

```
## -------------------------------------------------------------------------
```

```
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
```

```
## -------------------------------------------------------------------------
```

```
## 
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:dplyr':
## 
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:Hmisc':
## 
##     is.discrete, summarize
```

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

```
activity_ds <- read.csv('file:///C:/Users/emili/OneDrive/Documents/datacience specialization/Rep
 research/repdata_Fdata_Factivity/activity.csv')
head(activity_ds)
```

```
##   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
```

```
dim(activity_ds)
```

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

```
str(activity_ds)
```

```
## '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 ...
```

Process/transform the data

```
library(lubridate)
```

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

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

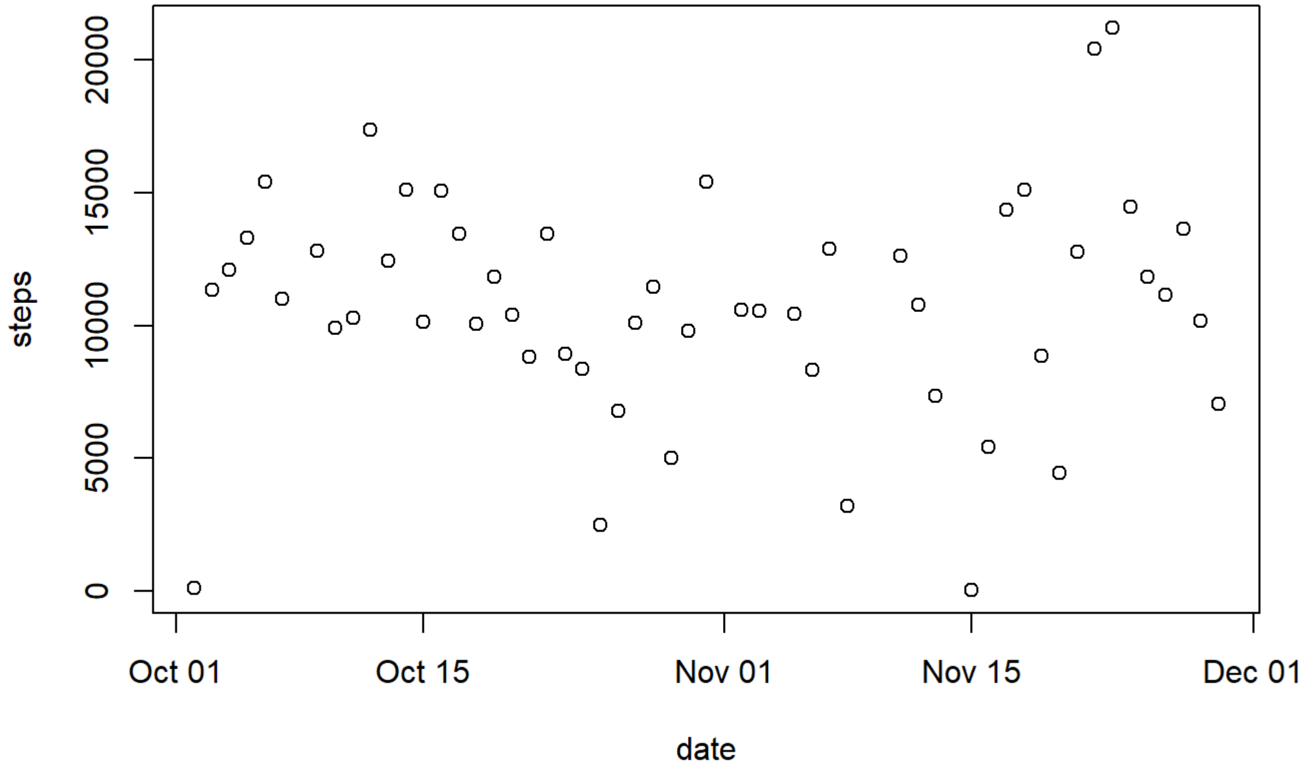
```
## The following object is masked from 'package:plyr':
## 
##     here
```

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

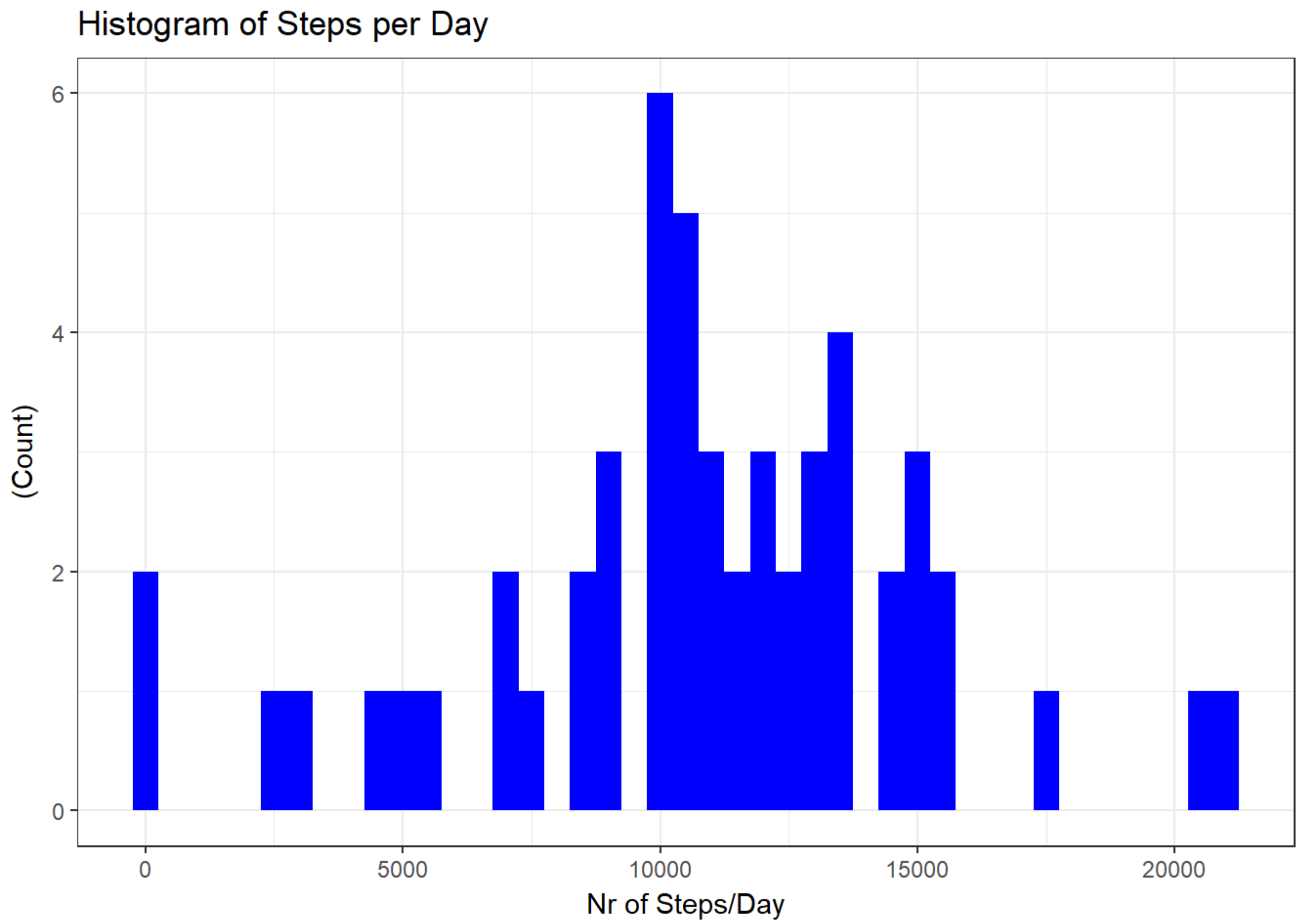
```
activity_ds$date <- ymd(activity_ds$date)

#activityData$interval <- strptime(gsub("([0-9]{1,2})([0-9]{2})", "\\1:\\2", activityData$interv
al), format='%H:%M')
```

2 Histogram of the total number of steps taken each day

```
steps_day <- aggregate(steps ~ date, data = activity_ds, FUN = sum, na.rm = TRUE)
plot(steps_day)
```



```
ggplot(steps_day, aes(x = steps)) +
  geom_histogram(fill = "blue", binwidth = 500) +
  labs(title="Histogram of Steps per Day",
       x = "Nr of Steps/Day", y = "(Count)") + theme_bw()
```



3. Calculate and report the mean and median total number of steps taken per day

```
steps_day_mean <- mean(steps_day$steps)
steps_day_median <- median(steps_day$steps)
steps_day_mean
```

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

```
#9354.23
steps_day_median
```

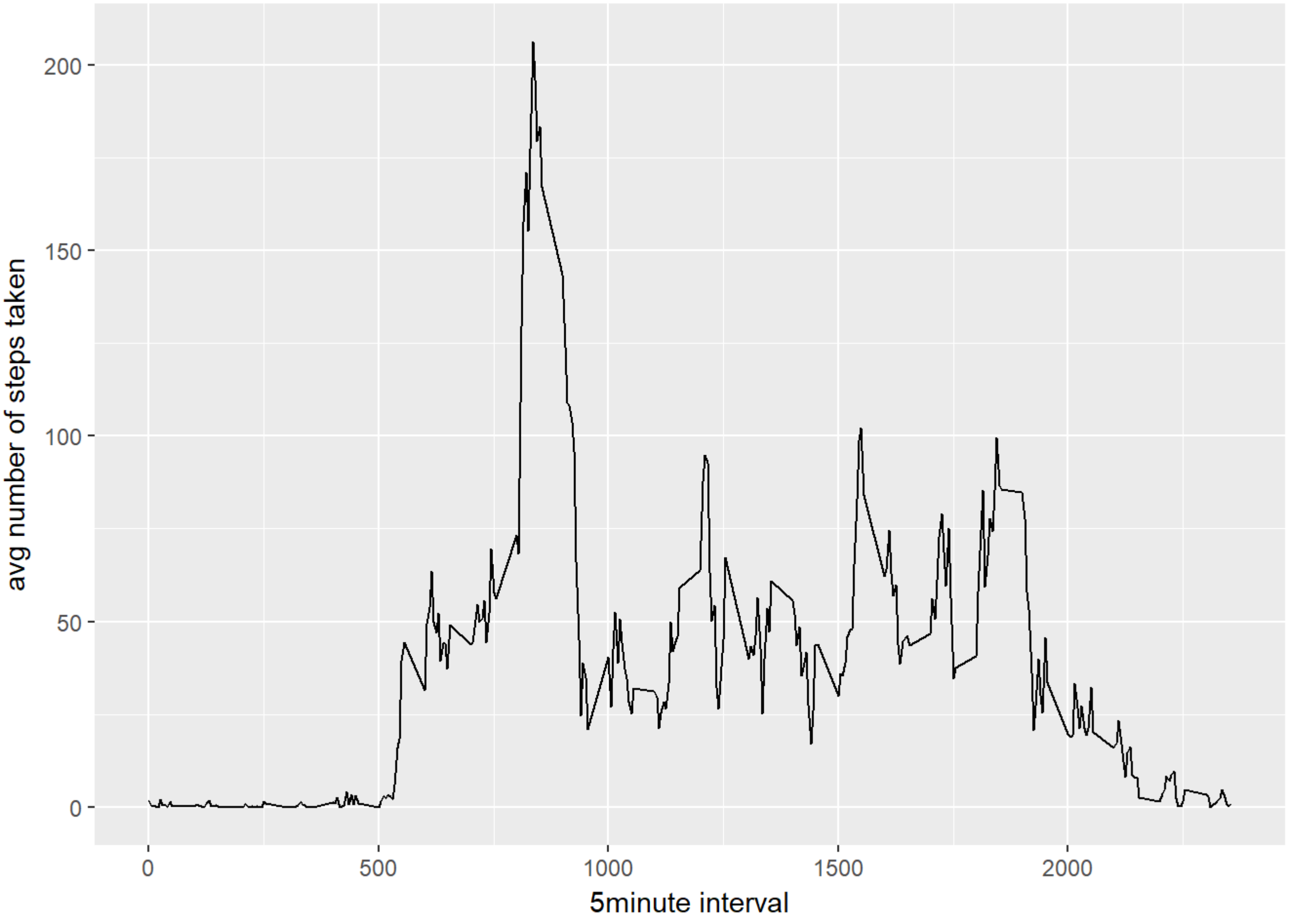
```
## [1] 10765
```

```
# 10395
```

4 Time series plot of the average number of steps taken

```
average_ap<- aggregate(steps ~ interval, data = activity_ds, FUN = mean, na.rm = TRUE)
```

```
ggplot(data=average_ap, aes(x=interval, y=steps)) +
  geom_line() +
  xlab("5minute interval") +
  ylab("avg number of steps taken")
```



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

```
maxsteps <- average_ap$interval[which.max(average_ap$steps)]
maxsteps
```

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

835th 5-min interval

Imputing missing values # 6 Code to describe and show a strategy for imputing missing data

Calculate and reportthe total number of rows with NAs)

```
missing <- length(which(is.na(activity_ds$steps)))
missing
```

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

2304 missing values

```
new_activity <- activity_ds
na <- is.na(new_activity$steps)
avg_data<- tapply(new_activity$steps, new_activity$interval, mean, na.rm=TRUE, simplify = TRUE)
new_activity$steps[na] <- avg_data[as.character(new_activity$interval[na])]
names(new_activity)
```

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

```
sum(is.na(new_activity))
```

```
## [1] 0
```

```
#no missing values in new dataset
summary(new_activity)
```

```
##      steps             date               interval
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8
##  Median :  0.00   Median :2012-10-31   Median :1177.5
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 27.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
```

without NA

```
new_activity2 <- aggregate(steps ~ date, data = new_activity, FUN = sum, na.rm = TRUE)
new_activity2
```

```
##          date    steps
## 1  2012-10-01 10766.19
## 2  2012-10-02   126.00
## 3  2012-10-03 11352.00
## 4  2012-10-04 12116.00
## 5  2012-10-05 13294.00
## 6  2012-10-06 15420.00
## 7  2012-10-07 11015.00
## 8  2012-10-08 10766.19
## 9  2012-10-09 12811.00
## 10 2012-10-10  9900.00
## 11 2012-10-11 10304.00
## 12 2012-10-12 17382.00
## 13 2012-10-13 12426.00
## 14 2012-10-14 15098.00
## 15 2012-10-15 10139.00
## 16 2012-10-16 15084.00
## 17 2012-10-17 13452.00
## 18 2012-10-18 10056.00
## 19 2012-10-19 11829.00
## 20 2012-10-20 10395.00
## 21 2012-10-21  8821.00
## 22 2012-10-22 13460.00
## 23 2012-10-23  8918.00
## 24 2012-10-24  8355.00
## 25 2012-10-25  2492.00
## 26 2012-10-26  6778.00
## 27 2012-10-27 10119.00
## 28 2012-10-28 11458.00
## 29 2012-10-29  5018.00
## 30 2012-10-30  9819.00
## 31 2012-10-31 15414.00
## 32 2012-11-01 10766.19
## 33 2012-11-02 10600.00
## 34 2012-11-03 10571.00
## 35 2012-11-04 10766.19
## 36 2012-11-05 10439.00
## 37 2012-11-06  8334.00
## 38 2012-11-07 12883.00
## 39 2012-11-08  3219.00
## 40 2012-11-09 10766.19
## 41 2012-11-10 10766.19
## 42 2012-11-11 12608.00
## 43 2012-11-12 10765.00
## 44 2012-11-13  7336.00
## 45 2012-11-14 10766.19
## 46 2012-11-15    41.00
## 47 2012-11-16  5441.00
## 48 2012-11-17 14339.00
## 49 2012-11-18 15110.00
## 50 2012-11-19  8841.00
## 51 2012-11-20  4472.00
## 52 2012-11-21 12787.00
## 53 2012-11-22 20427.00
## 54 2012-11-23 21194.00
## 55 2012-11-24 14478.00
## 56 2012-11-25 11834.00
## 57 2012-11-26 11162.00
## 58 2012-11-27 13646.00
## 59 2012-11-28 10183.00
## 60 2012-11-29  7047.00
## 61 2012-11-30 10766.19
```

```
head(new_activity2)
```

```
##         date    steps
## 1 2012-10-01 10766.19
## 2 2012-10-02   126.00
## 3 2012-10-03 11352.00
## 4 2012-10-04 12116.00
## 5 2012-10-05 13294.00
## 6 2012-10-06 15420.00
```

```
#compare
summary(new_activity)
```

```
##      steps             date               interval     
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0  
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8  
##  Median :  0.00   Median :2012-10-31   Median :1177.5  
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5  
##  3rd Qu.: 27.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2  
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0  
```

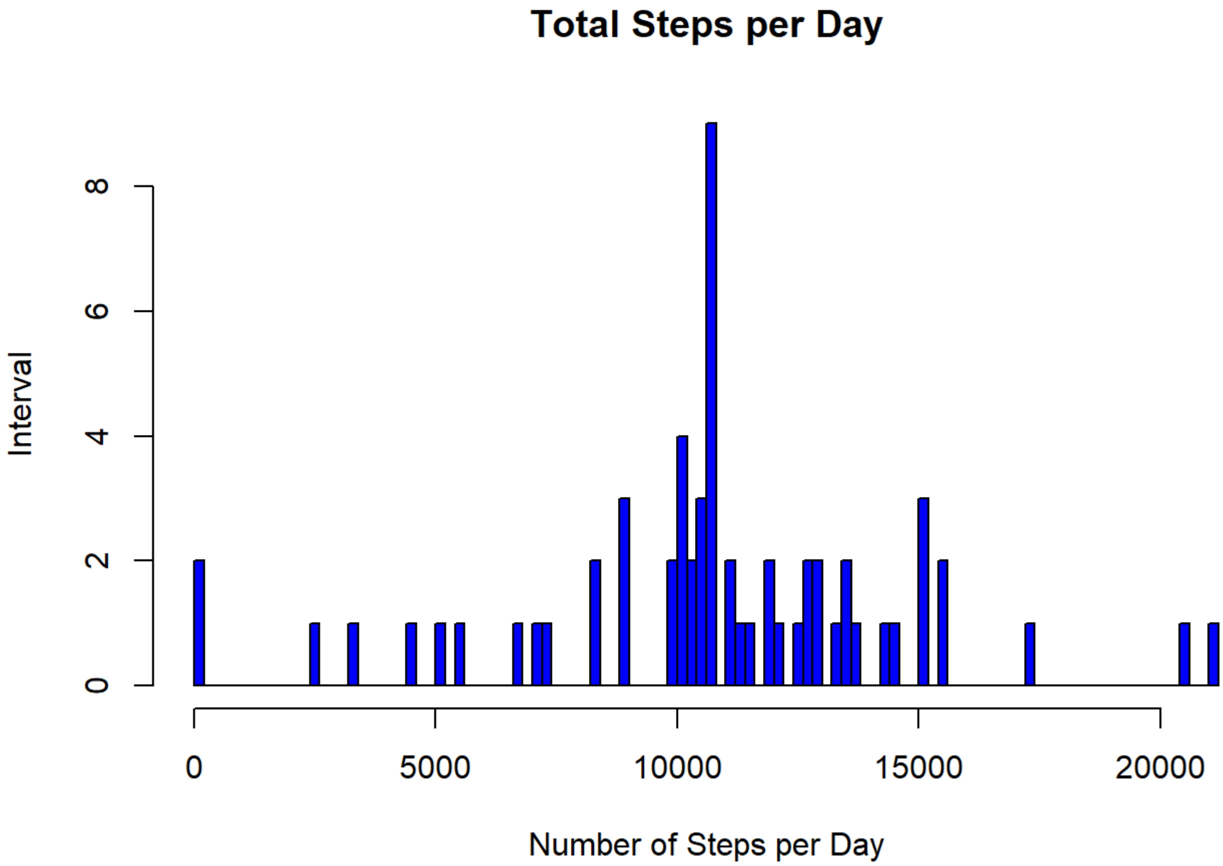
```
summary(new_activity2)
```

```
##       date                steps      
##  Min.   :2012-10-01   Min.   :   41  
##  1st Qu.:2012-10-16   1st Qu.: 9819  
##  Median :2012-10-31   Median :10766  
##  Mean   :2012-10-31   Mean   :10766  
##  3rd Qu.:2012-11-15   3rd Qu.:12811  
##  Max.   :2012-11-30   Max.   :21194  
```

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

Histogram without the NA values

```
hist(new_activity2$steps,
     main = "Total Steps per Day",
     xlab = "Number of Steps per Day",
     ylab = "Interval",
     col="blue",
     breaks=100)
```



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

New factor variable >> two levels – “weekday” | “weekend”

method1

```
new_activity<- new_activity %>%
mutate(typeofday= ifelse(weekdays(new_activity$date)=="Saturday" |
weekdays(new_activity$date)=="Sunday", "Weekend", "Weekday"))
head(new_activity)
```

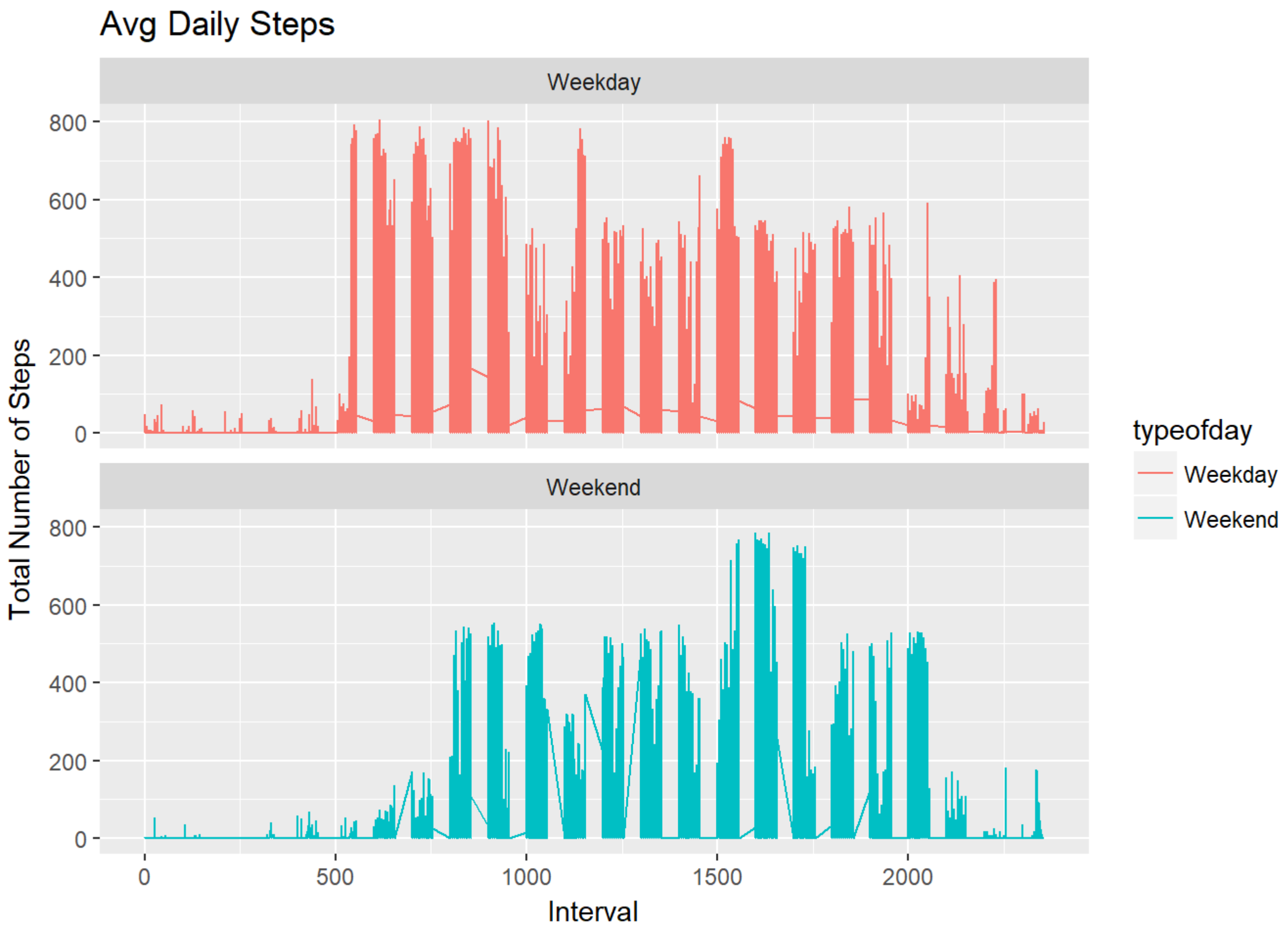
```
##       steps       date interval typeofday
## 1 1.7169811 2012-10-01        0   Weekday
## 2 0.3396226 2012-10-01        5   Weekday
## 3 0.1320755 2012-10-01       10   Weekday
## 4 0.1509434 2012-10-01       15   Weekday
## 5 0.0754717 2012-10-01       20   Weekday
## 6 2.0943396 2012-10-01       25   Weekday
```

Plot1

```
fivemin<- aggregate(steps ~ interval, data = new_activity, FUN = mean, na.rm = TRUE)
head(fivemin)
```

```
##   interval     steps
## 1        0 1.7169811
## 2        5 0.3396226
## 3       10 0.1320755
## 4       15 0.1509434
## 5       20 0.0754717
## 6       25 2.0943396
```

```
ggplot(new_activity, aes(x =interval , y=steps, color=typeofday)) +
  geom_line() +
  labs(title = "Avg Daily Steps", x = "Interval", y = "Total Number of Steps") +
  facet_wrap(~ typeofday, ncol = 1, nrow=2)
```


Avg Daily Steps
Weekday
Weekend
Total Number of Steps
800
600
400
200
0
0
500
1000
1500
2000
Interval
typeofday
Weekday
Weekend