Reproducible Research: Peer Assessment 1 (Elisa Du)

Load some packages.

```
library(knitr)
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)
```

First, set global options.

```
opts_chunk$set(echo=TRUE,cache=FALSE)
```

Loading and preprocessing the data

```
df<-read.csv('activity.csv',sep=',',na.strings = 'NA')
head(df)
```

```
##   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
```

What is mean total number of steps taken per day?

To make a histogram of total number of steps taken each day, first sort data by date.

```
df2<-df[!is.na(df$steps),] # subset all rows with non-NA values
head(df2)
```

```
##     steps       date interval
## 289     0 2012-10-02        0
## 290     0 2012-10-02        5
## 291     0 2012-10-02       10
## 292     0 2012-10-02       15
## 293     0 2012-10-02       20
## 294     0 2012-10-02       25
```

```
df_GroupByDate<-group_by(df2,date) # group by 'date' as factor
head(df_GroupByDate)
```

```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   steps date       interval
##   <int> <fct>         <int>
## 1     0 2012-10-02        0
## 2     0 2012-10-02        5
## 3     0 2012-10-02       10
## 4     0 2012-10-02       15
## 5     0 2012-10-02       20
## 6     0 2012-10-02       25
```

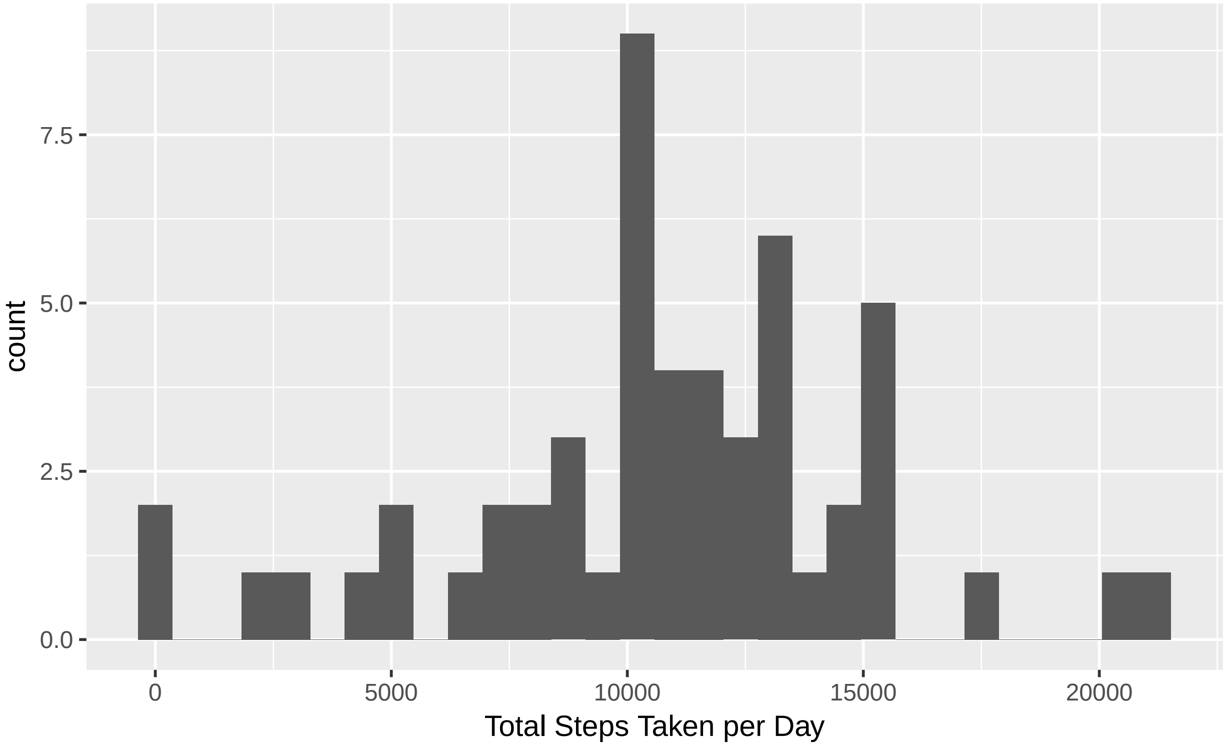
```
df_stepsPerDay<-summarise(df_GroupByDate,TotalSteps=sum(steps)) # create new data frame
head(df_stepsPerDay)
```

```
## # A tibble: 6 x 2
##   date       TotalSteps
##   <fct>           <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
```

Now we can plot the histogram.

```
plot1<-qplot(TotalSteps,data=df_stepsPerDay,geom='histogram')
plot1+labs(x='Total Steps Taken per Day')
```

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



Now we find the mean total number of steps taken for each day.

```
meanTotalSteps<- mean(df_stepsPerDay$TotalSteps)
meanTotalSteps
```

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

```
meanTotalSteps<-as.numeric(format(meanTotalSteps,digits=5))
meanTotalSteps # round to integer
```

```
## [1] 10766
```

Next we find the median total number of steps taken for each day.

```
medianTotalSteps<- median(df_stepsPerDay$TotalSteps)
meanTotalSteps<-as.numeric(format(medianTotalSteps,digits=5))
medianTotalSteps # round to integer
```

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

The mean total number of steps taken per day is 10766 (rounded to integer).
The median total number of steps taken per day is 10765.

What is the average daily activity pattern?

To plot the average number of steps per 5-minute interval:

```
df_GroupByInterval<-group_by(df2,interval)
tail(df_GroupByInterval)
```

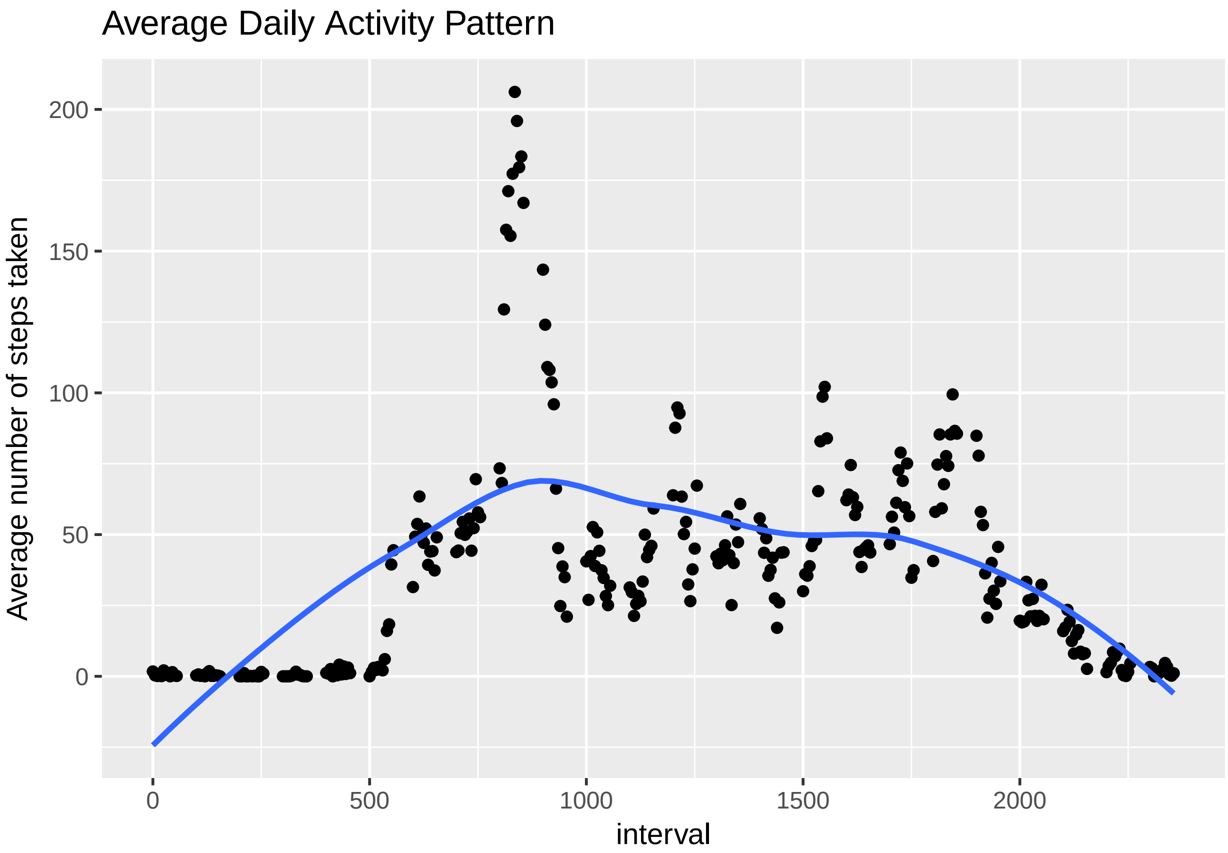
```
## # A tibble: 6 x 3
## # Groups:   interval [6]
##   steps date       interval
##   <int> <fct>         <int>
## 1     0 2012-11-29     2330
## 2     0 2012-11-29     2335
## 3     0 2012-11-29     2340
## 4     0 2012-11-29     2345
## 5     0 2012-11-29     2350
## 6     0 2012-11-29     2355
```

```
df_AveStepsInterval <- summarise(df_GroupByInterval,MeanSteps = mean(steps))
head(df_AveStepsInterval)
```

```
## # A tibble: 6 x 2
##   interval MeanSteps
##      <int>     <dbl>
## 1        0    1.72
## 2        5    0.340
## 3       10    0.132
## 4       15    0.151
## 5       20    0.0755
## 6       25    2.09
```

```
plot2<-qplot(interval,MeanSteps,data=df_AveStepsInterval,geom='point')
plot2 + geom_smooth(se=FALSE) + labs(x='interval',y='Average number of steps taken',
        title='Average Daily Activity Pattern')
```

```
## `geom_smooth()` using method = 'loess'
```

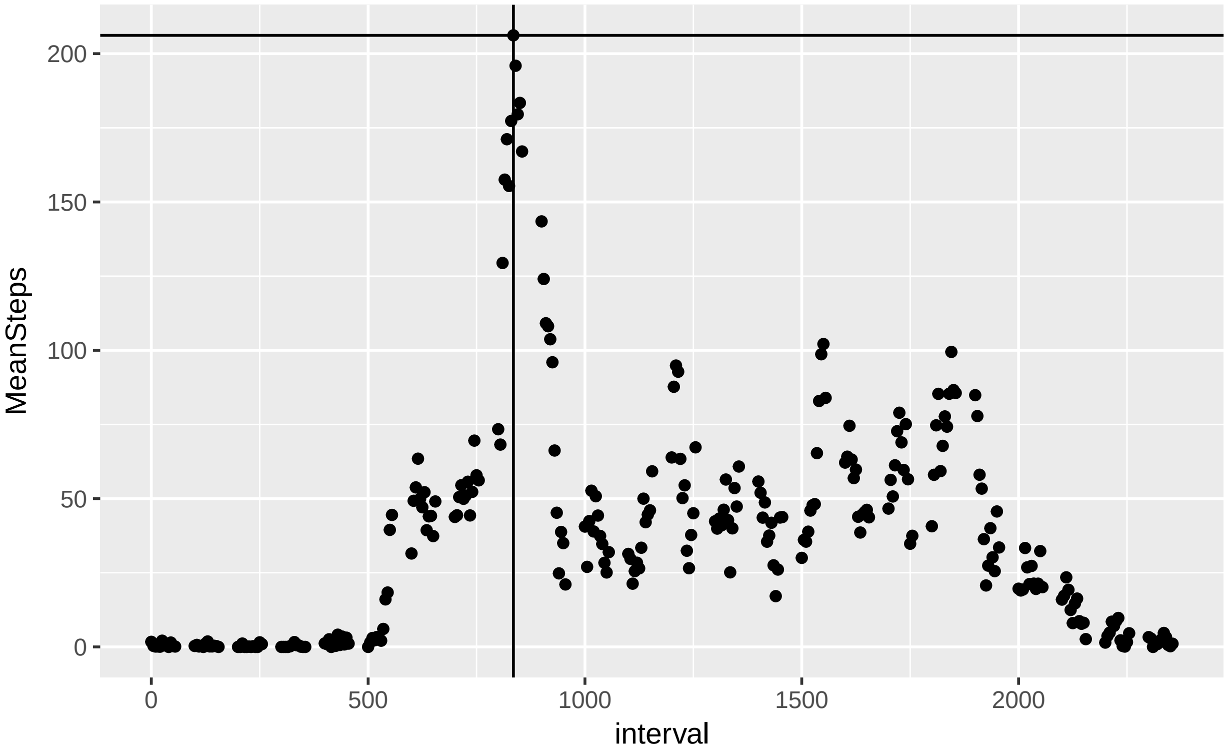


To find the 5-minute interval where the average number of steps taken is at a maximum:

```
max <- max(df_AveStepsInterval$MeanSteps)
# interval at which max avg steps occur
maxInt<-df_AveStepsInterval[df_AveStepsInterval$MeanSteps==max,]$interval
maxInt
```

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

```
# plot where max average step occurs
plot2 + geom_hline(yintercept = max) + geom_vline(xintercept = maxInt)
```



Therefore the interval 835 contains the maximum average number of steps.

Imputing missing values

```
df_NA<-df[is.na(df$steps),]
nrow(df_NA)
```

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

There are 2304 missing values in the dataset.

Next, fill in all missing values as mean of the corresponding 5-min interval.

'impdata' is the new dataset with the missing values filled in.

```
impData<-cbind(df,MeanSteps=df_AveStepsInterval$MeanSteps)
impData$steps[is.na(impData$steps)] <- impData[is.na(impData$steps),]$MeanSteps
head(impData)
```

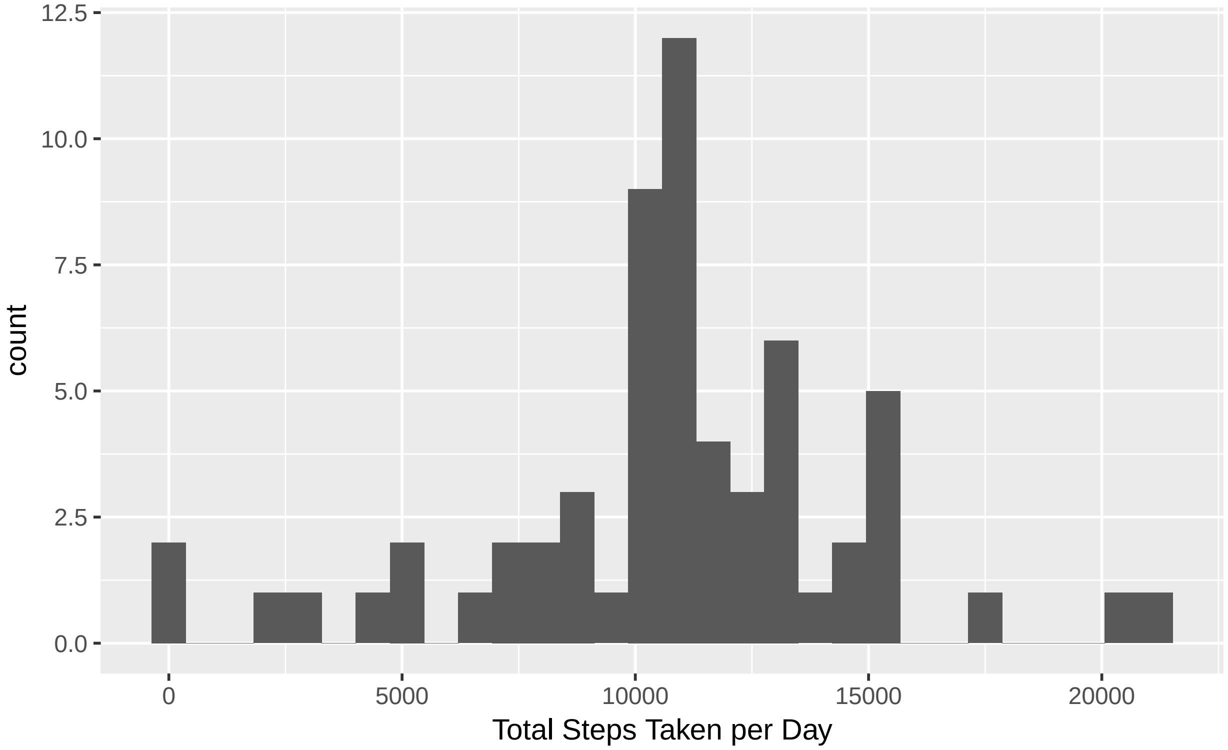
```
##       steps       date interval MeanSteps
## 1 1.7169811 2012-10-01        0 1.7169811
## 2 0.3396226 2012-10-01        5 0.3396226
## 3 0.1320755 2012-10-01       10 0.1320755
## 4 0.1509434 2012-10-01       15 0.1509434
## 5 0.0754717 2012-10-01       20 0.0754717
## 6 2.0943396 2012-10-01 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

Plot histogram of total number of steps taken each day, accounting for filled-in missing values.

```
Newdf_GroupByDate<-group_by(impData,date) # group by 'date' as factor
Newdf_stepsPerDay<-summarise(Newdf_GroupByDate,TotalSteps=sum(steps)) # create new data frame
```

```r
plot3<-qplot(TotalSteps,data=Newdf_stepsPerDay,geom='histogram')
plot3+labs(x='Total Steps Taken per Day')
```

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



Find new mean and median.

```r
NewMeanTot<- mean(Newdf_stepsPerDay$TotalSteps)
NewMeanTot<-as.numeric(format(meanTotalSteps,digits=5))
NewMeanTot # round to integer
```

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

```r
NewMedTot<- median(Newdf_stepsPerDay$TotalSteps)
NewMedTot<-as.numeric(format(meanTotalSteps,digits=5))
NewMedTot
```

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

Both the new mean and median steps taken are 10765. The mean remains the same as before missing values are filled in, and the median is increased by 1. Imputing estimates of total daily steps as the mean of corresponding 5-min interval slightly overestimates the median, but does not affect the mean in this scenario.

Are there differences in activity patterns between weekdays and weekends?

Create new factor variable comprised of the levels 'weekday' and 'weekend'. Incorporate into dataset with filled-in missing vlaues to differentiate date as weekday or weekend.

```r
impData$date<-as.Date(impData$date)
WeekDays<-c('Monday','Tuesday','Wednesday','Thursday','Friday')
impData$DayType<-factor((weekdays(impData$date) %in% WeekDays), levels=c(FALSE,TRUE),labels=c('weekend'
df_WeekDay <- impData[impData$DayType == 'weekDay', ]
df_Weekend<- impData[impData$DayType == 'weekend', ]
```

We can now plot the average steps taken for both weekends and weekdays.

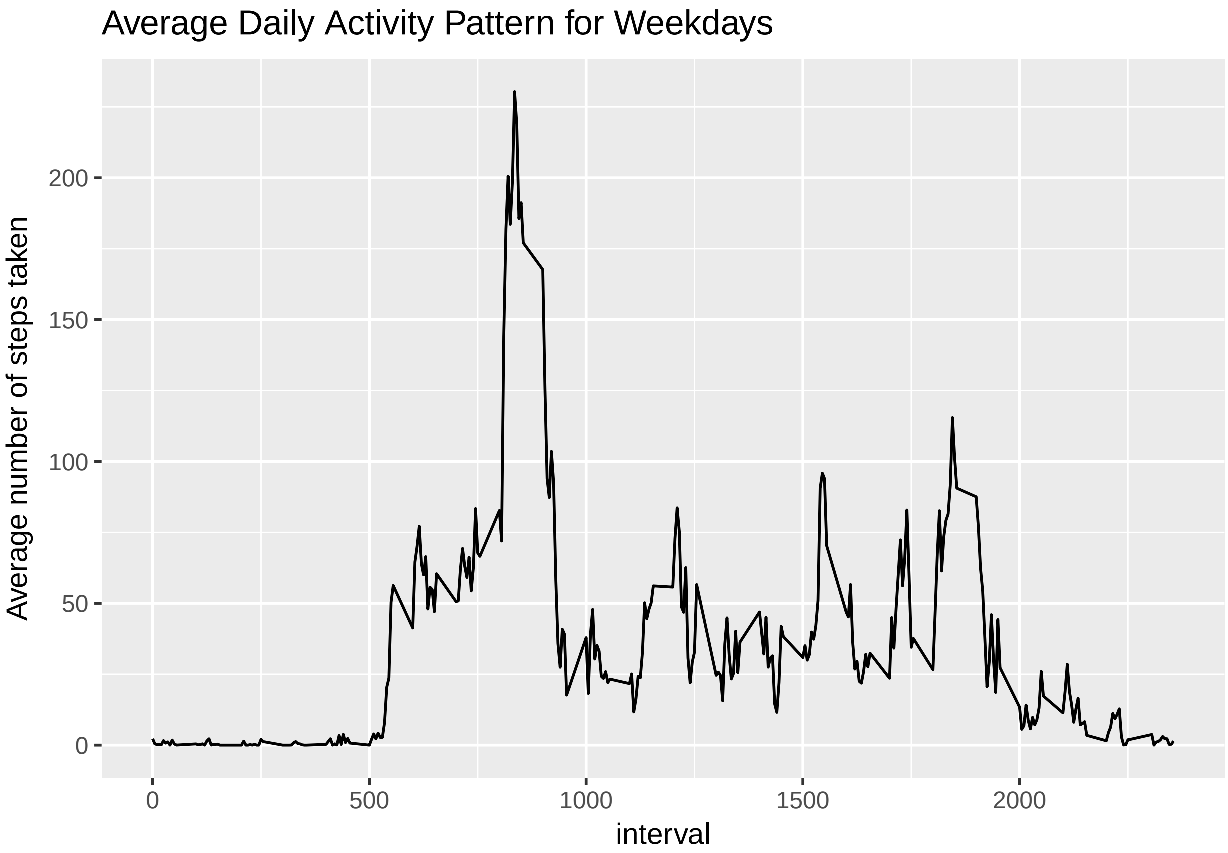
```
df_WeekDayInt<-group_by(df_WeekDay,interval)
head (df_WeekDayInt)
```

```
## # A tibble: 6 x 5
## # Groups:   interval [6]
##    steps date       interval MeanSteps DayType
##    <dbl> <date>        <int>     <dbl> <fct>
## 1 1.72   2012-10-01        0    1.72   weekDay
## 2 0.340  2012-10-01        5    0.340  weekDay
## 3 0.132  2012-10-01       10    0.132  weekDay
## 4 0.151  2012-10-01       15    0.151  weekDay
## 5 0.0755 2012-10-01       20    0.0755 weekDay
## 6 2.09   2012-10-01       25    2.09   weekDay
```

```
df_WeekdayAve <- summarise(df_WeekDayInt,MeanSteps = mean(steps))
head(df_WeekdayAve)
```

```
## # A tibble: 6 x 2
##   interval MeanSteps
##      <int>     <dbl>
## 1        0    2.25
## 2        5    0.445
## 3       10    0.173
## 4       15    0.198
## 5       20    0.0990
## 6       25    1.59
```

```
plot4 <- qplot(interval,MeanSteps,data=df_WeekdayAve,geom = 'line' )
plot4 + labs(x='interval',y='Average number of steps taken',
             title='Average Daily Activity Pattern for Weekdays')
```



```
df_WeekEndInt <- group_by(df_Weekend,interval)
tail(df_WeekEndInt)
```

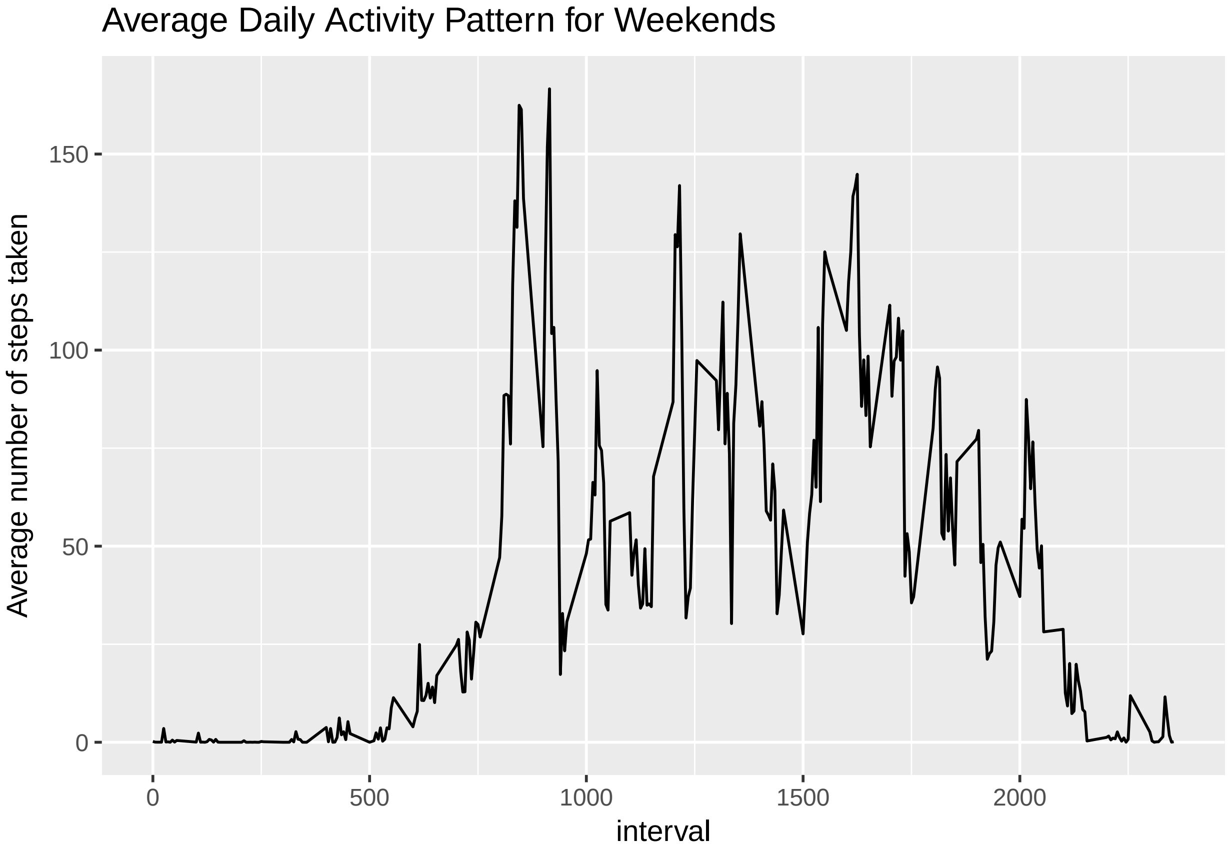
```
## # A tibble: 6 x 5
## # Groups:   interval [6]
##   steps date       interval MeanSteps DayType
##   <dbl> <date>        <int>     <dbl> <fct>
## 1    17 2012-11-25     2330     2.60  weekend
## 2   176 2012-11-25     2335     4.70  weekend
## 3    94 2012-11-25     2340     3.30  weekend
## 4    26 2012-11-25     2345     0.642 weekend
## 5     0 2012-11-25     2350     0.226 weekend
## 6     0 2012-11-25     2355     1.08  weekend
```

```
df_WeekendAve <- summarise(df_WeekEndInt,MeanSteps = mean(steps))
head(df_WeekendAve)
```

```
## # A tibble: 6 x 2
##   interval MeanSteps
##      <int>     <dbl>
## 1        0   0.215
## 2        5   0.0425
## 3       10   0.0165
## 4       15   0.0189
## 5       20   0.00943
## 6       25   3.51
```

```
plot5 <- qplot(interval,MeanSteps,data=df_WeekendAve,geom = 'line' )
```

```
plot5 + labs(x='interval',y='Average number of steps taken',
          title='Average Daily Activity Pattern for Weekends')
```


Average Daily Activity Pattern for Weekends
Average number of steps taken
interval