Reproducible Research: Peer Assessment 1

Loading required libraries

```
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)
library(lubridate)
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, mday, month, quarter, wday, week, yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, last
```

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 3.2.4

##
## Attaching package: 'reshape2'

## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```
#library(zoo)
```

Loading and preprocessing the data

```
#setwd("C:/Users/R/OneDrive/coursera/reproducibleResearch/week1Assignment")

if(!exists("activity")){
  unzip("activity.zip")
  activity<- read.csv("activity.csv")
}

activity$date<-ymd(as.character(activity$date))
```

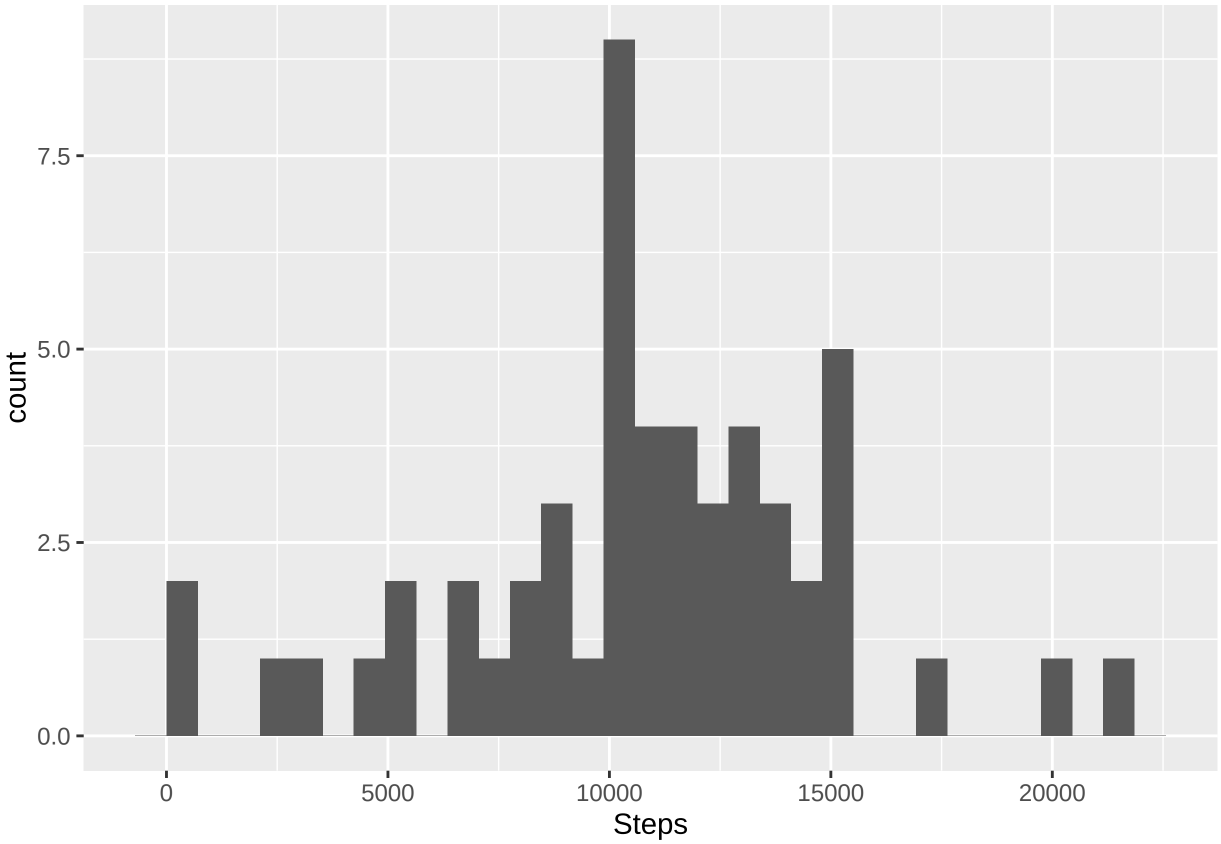
What is mean total number of steps taken per day?

```
activityByDay<- group_by(activity[,c(1:2)],date) %>%summarise(sum(steps))
names(activityByDay)<-c("date","totalSteps")

totalStepsaDay <- ggplot(activityByDay, aes(x=totalSteps)) +geom_histogram()  +labs(x="Steps")
print(totalStepsaDay)
```

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

```
## Warning: Removed 8 rows containing non-finite values (stat_bin).
```



```
meanTotalDays <- mean(activityByDay$totalSteps,na.rm = TRUE)
medianTotalDays <- median(activityByDay$totalSteps,na.rm=TRUE)
```

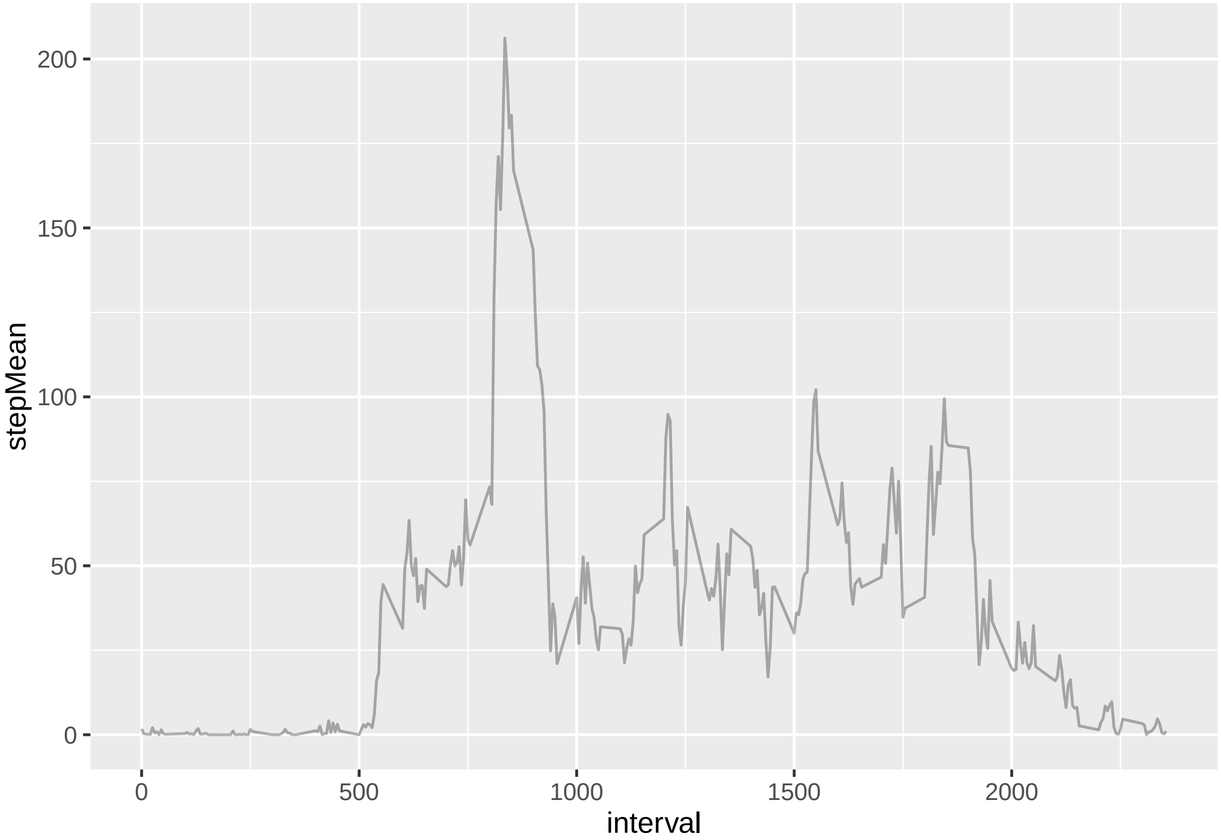
Mean Total steps taken a day: 10766 Median Total steps taken a day: 10765

What is the average daily activity pattern?

See below the time serie of the interval averages over all days.

```
activityByInterval<- group_by(activity[,-2],interval) %>%summarise(mean(steps,na.rm=TRUE))
names(activityByInterval)<- c("interval", "stepMean")

IntervalMean <- ggplot(data=activityByInterval,aes(x=interval,y=stepMean)) + geom_line(alpha=.3)
print(IntervalMean)
```



```
maxInterval <- filter(activityByInterval,stepMean==max(activityByInterval$stepMean, na.rm = TRUE))
```

The 5 minute inteval of the day with the most steps is: 835, 206.1698113

Imputing missing values

First I tried for replace Na's with the function "na.locf" of the zoo package, it replaces a na with a nearest value, which most of the time is a 0, so not good, finally I used the suggested solution from the instruction, replace a NA in a day for a mean of that day. I had some inspiration from slashdot (link to slashdot resource).

Eyeballing the histogram the frequence for some is now a bit higher but when calculating the means of the totals of the data frame with NA and from the data frame without NA I could hardly see a difference.

```
NumberOfNA <- sum(!complete.cases(activity))
NumberOfNA
```

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

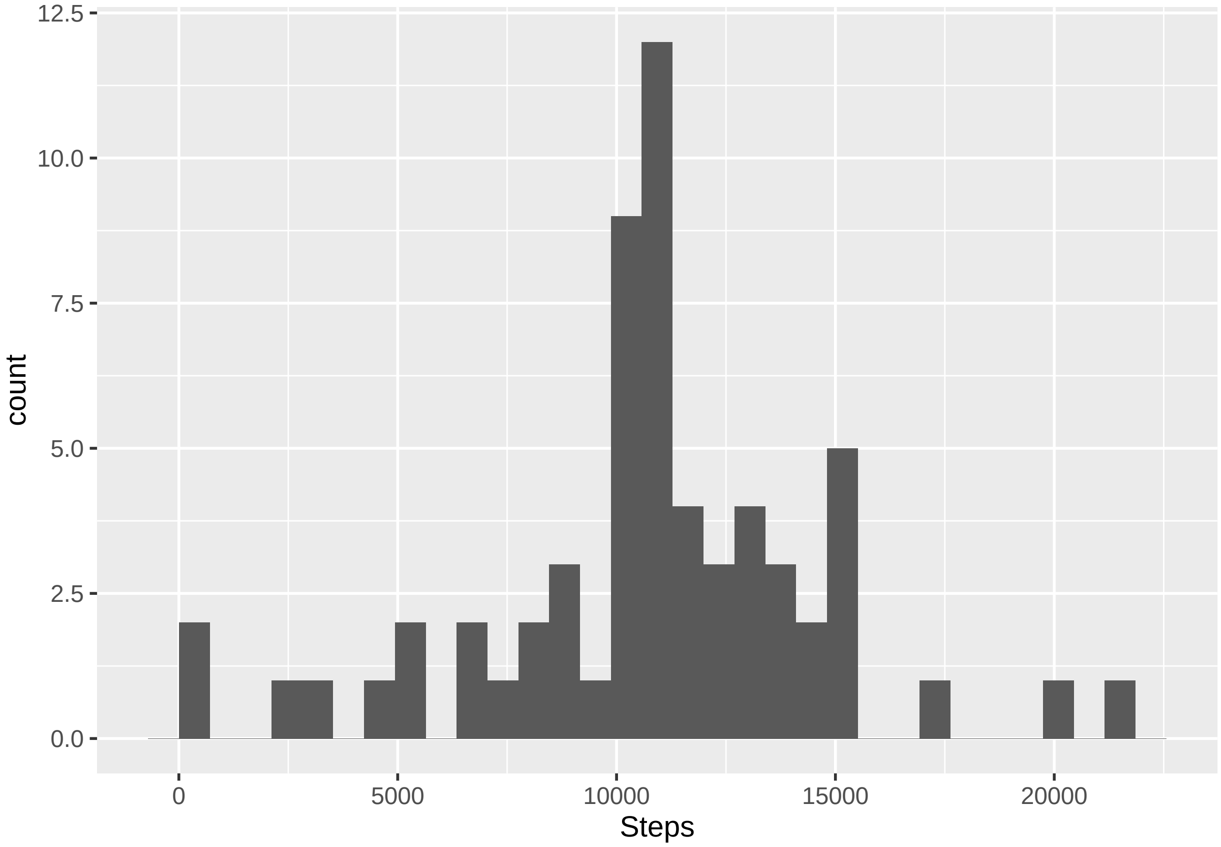
```
impute.mean <- function(x) replace(x, is.na(x), mean(x,na.rm=TRUE))
activity<- group_by(activity,interval) %>%mutate(steps =impute.mean(steps))

NumberOfNA_afterReplacement <- sum(!complete.cases(activity))
NumberOfNA_afterReplacement
```

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

```
activityByDayNoNA<- group_by(activity,date) %>%summarise(sum(steps))
names(activityByDayNoNA)<-c("date","totalSteps")
totalStepsaDay <- ggplot(activityByDayNoNA, aes(x=totalSteps)) +geom_histogram()  +labs(x="Steps")
print(totalStepsaDay)
```

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



```
#
 meanNoNATotalDays <- mean(activityByDayNoNA$totalSteps)
 medianNoNATotalDays <- median(activityByDayNoNA$totalSteps)
```

Mean Total steps taken a day: 10766 Median Total steps taken a day: 10766

Are there differences in activity patterns between weekdays and weekends?

See below the two time series of steps during weekdays and steps during weekends,
there is more activity(more steps) early in the day during weekdays than during weekends.

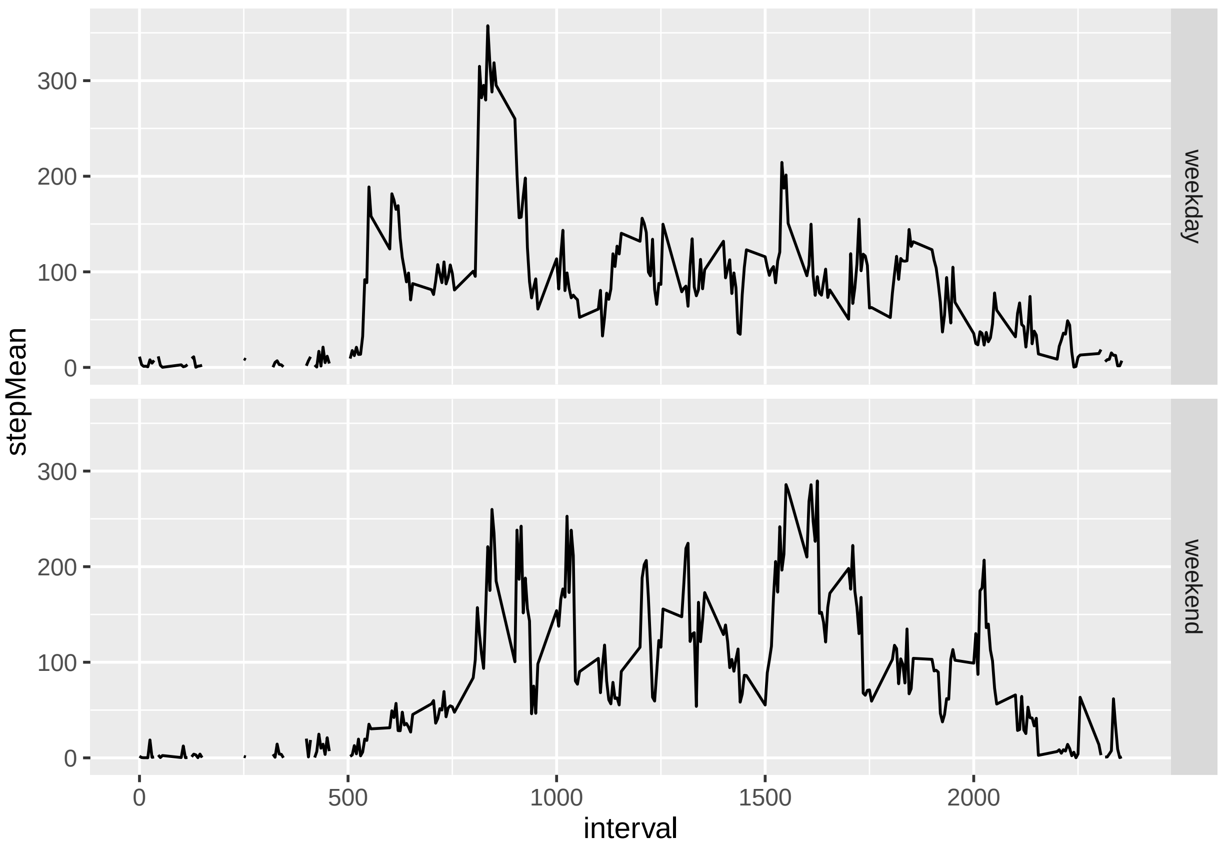
```
#add factor
isweekend<-ifelse(wday(activity$date) %in% c(7,1),1,0)
activity$kindOfDay<-factor(isweekend,levels = c(0,1),labels=c("weekday","weekend"))

#create data frame interval over weekdays
weekdayActivityByInterval <- filter(activity,kindOfDay=="weekday")
weekdayActivityByInterval<- group_by(weekdayActivityByInterval[,-2],interval) %>%summarise(mean(steps[s
names(weekdayActivityByInterval)<- c("interval", "stepMean","kindOfDay")

#create data frame interval over weekends
weekendActivityByInterval <- filter(activity,kindOfDay=="weekend")
weekendActivityByInterval<- group_by(weekendActivityByInterval[,-2],interval) %>%summarise(mean(steps[s
names(weekendActivityByInterval)<- c("interval", "stepMean","kindOfDay")

#merge weekdays and weekends data frames
weekActivityByInterval <-rbind(weekendActivityByInterval,weekdayActivityByInterval)

#plotting
dayIntervalMean <- ggplot(data=weekActivityByInterval,aes(x=interval,y=stepMean)) + geom_line()  +
                  facet_grid(kindOfDay ~ .)
print(dayIntervalMean)
```


stepMean
300
200
100
0
300
200
100
0
weekday
weekend
0
500
1000
1500
2000
interval