# This is My Markdown File

Siddhesh Dalavi

## Reproducible search, week2 assignment

library(knitr)  
opts\_chunk$set(echo=TRUE,results="asis")

Load the data.

Activity\_data<-read.csv("activity.csv")  
summary(Activity\_data)

steps date interval

Min. : 0.00 2012-10-01: 288 Min. : 0.0  
1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8  
Median : 0.00 2012-10-03: 288 Median :1177.5  
Mean : 37.38 2012-10-04: 288 Mean :1177.5  
3rd Qu.: 12.00 2012-10-05: 288 3rd Qu.:1766.2  
Max. :806.00 2012-10-06: 288 Max. :2355.0  
NA's :2304 (Other) :15840

head(Activity\_data)

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

Remove missing values

act.complete<-na.omit(Activity\_data)  
head(act.complete)

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

Total Number of steps taken

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

act.day<-group\_by(act.complete,date)  
act.day<-summarize(act.day,steps=sum(steps))  
head(act.day)

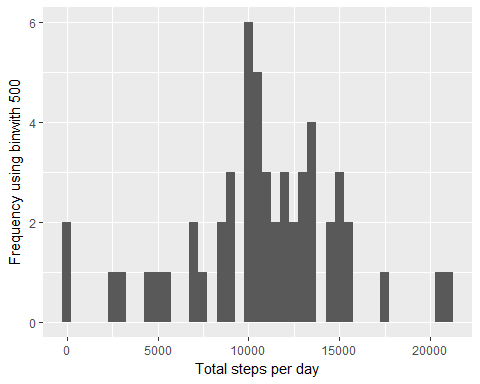
# A tibble: 6 x 2

date steps  
 <fctr> <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

Plot the histogram

library(ggplot2)  
qplot(steps,data=act.day,xlab='Total steps per day', ylab='Frequency using binwith 500', binwidth=500)



calculating the mean and median of steps taken per day

mean(act.day$steps)

[1] 10766.19

median(act.day$steps)

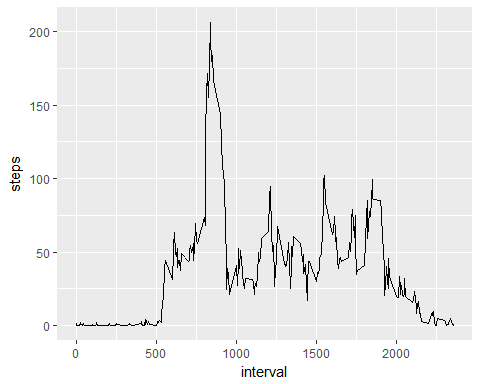
[1] 10765

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)

act.int<-group\_by(act.complete,interval)  
act.int<-summarize(act.int,steps=mean(steps))

plot the graph

ggplot(act.int,aes(interval,steps))+geom\_line()



We find the row in the interval data frame for which steps is equal to the maximum number of steps, then we look at the interval of that row

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

# A tibble: 1 x 2

interval steps 1 835 206.1698

total number of NA values

nrow(Activity\_data)-nrow(act.complete)

[1] 2304

filling up all missing values

names(act.int)[2] <- "mean.steps"  
act.impute <- merge(Activity\_data, act.int)

create new dataset with filled values

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

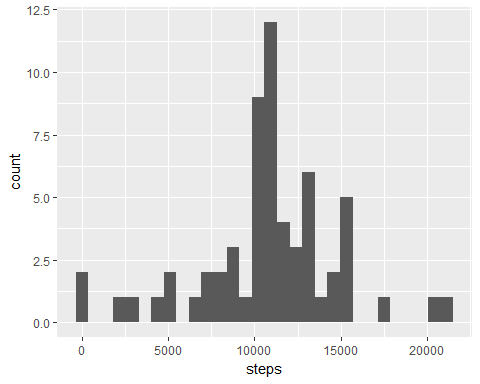
create a dataset with imputed values

act.day.imp <- group\_by(act.impute, date)  
act.day.imp <- summarize(act.day.imp, steps=sum(steps))

plot the graph

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

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



mean(act.day.imp$steps)

[1] 10766.19

median(act.day.imp$steps)

[1] 10766.19

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.

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")

Create seperate dataframes for weekday and weekend, also take mean for steps

act.weekday <- act.impute[act.impute$weekend=="Weekday",]  
act.weekend <- act.impute[act.impute$weekend=="Weekend",]  
act.int.weekday <- group\_by(act.weekday, interval)  
act.int.weekday <- summarize(act.int.weekday, steps=mean(steps))  
act.int.weekday$weekend <- "Weekday"  
act.int.weekend <- group\_by(act.weekend, interval)  
act.int.weekend <- summarize(act.int.weekend, steps=mean(steps))  
act.int.weekend$weekend <- "Weekend"

now append the 2 dataframes and plot the time series

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 ~ .)

