Reproducible Research: Peer Assessment 1

## Load packages

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.2.5

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.2.5

## Loading and preprocessing the data

activity <- read.csv("activity.csv", stringsAsFactors=FALSE)  
str(activity)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : chr "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

### Summarizing steps per day

activity\_steps\_day <- na.omit(activity) %>%   
 group\_by(date) %>%  
 summarise(steps\_day=sum(steps))  
  
head(activity\_steps\_day)

## # A tibble: 6 × 2  
## date steps\_day  
## <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

tail(activity\_steps\_day)

## # A tibble: 6 × 2  
## date steps\_day  
## <chr> <int>  
## 1 2012-11-24 14478  
## 2 2012-11-25 11834  
## 3 2012-11-26 11162  
## 4 2012-11-27 13646  
## 5 2012-11-28 10183  
## 6 2012-11-29 7047

### Summarizing steps on average through the 24 hours

activity\_steps\_hour <- na.omit(activity) %>%   
 group\_by(interval) %>%  
 summarise(steps\_interval=mean(steps))  
  
activity\_steps\_hour <- activity\_steps\_hour %>% mutate(hour = trunc(interval/100))

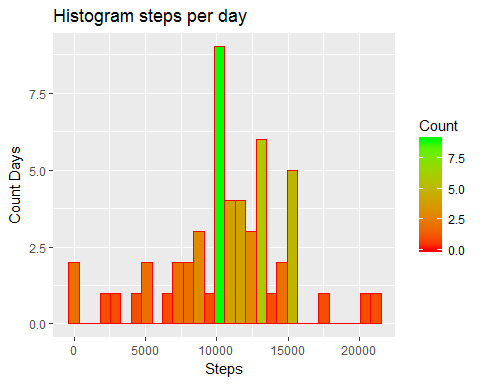
## What is the mean total number of steps taken per day?

For this part of the assignment the missing values in the dataset are ignore.

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

ggplot(data=activity\_steps\_day, aes(activity\_steps\_day$steps\_day)) +   
 geom\_histogram(col="red",   
 aes(fill=..count..)) +  
 scale\_fill\_gradient("Count", low = "red", high = "green") +  
 labs(title="Histogram steps per day") +  
 labs(x="Steps", y="Count Days")

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



### Mean and median of total number of steps taken per day

activity\_steps\_day %>%   
 summarise(steps\_median=median(steps\_day))

## # A tibble: 1 × 1  
## steps\_median  
## <int>  
## 1 10765

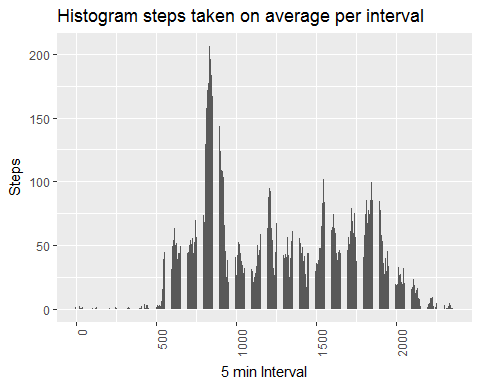
activity\_steps\_day %>%   
 summarise(steps\_mean=mean(steps\_day))

## # A tibble: 1 × 1  
## steps\_mean  
## <dbl>  
## 1 10766.19

## Average daily activity pattern

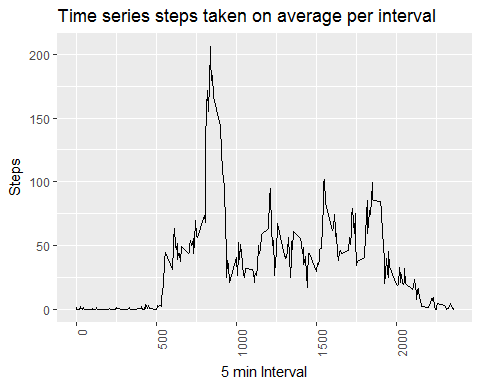
### Time series plot 5 minute interval

ggplot(data=activity\_steps\_hour, aes(x=interval, y=steps\_interval)) +   
 geom\_bar(stat="identity") +   
 labs(title="Histogram steps taken on average per interval") +  
 labs(x="5 min Interval", y="Steps") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



\*\*From the histogram we can observe that the most active hour is 8AM.

ggplot(data=activity\_steps\_hour, aes(x=interval, y=steps\_interval)) + geom\_line() +  
 labs(title="Time series steps taken on average per interval") +  
 labs(x="5 min Interval", y="Steps") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



### 5-minute interval which contains on average maximum number of steps across all the days in the dataset

activity\_steps\_hour %>%   
 summarise(steps\_max=max(steps\_interval))

## # A tibble: 1 × 1  
## steps\_max  
## <dbl>  
## 1 206.1698

### The 5-minute interval with more steps on average is at 8:35am. I could find the max average but I was not able to display in the same result which time specific interval was it. I look for it manually in the data.

## Imputing missing values

### Total number of missing values in the dataset

sum(is.na(activity$steps))

## [1] 2304

Total number of rows with missing values in the data set is 2304.

### strategy for filling missing values is assiging the 5-minute interval mean:

activity\_steps\_hour %>%   
 summarise(steps\_mean=mean(steps\_interval))

## # A tibble: 1 × 1  
## steps\_mean  
## <dbl>  
## 1 37.3826

activity2 <- activity  
activity2$steps[is.na(activity2$steps)] <- 37  
  
sum(is.na(activity2$steps))

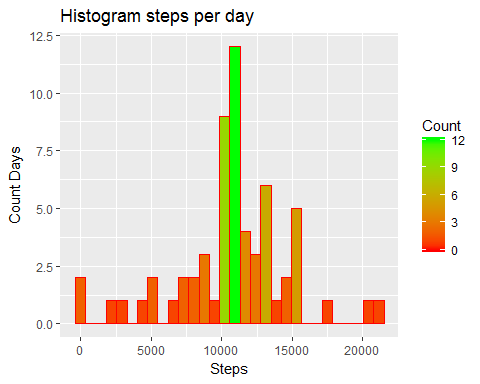
## [1] 0

### Histogram of the total number of steps taken each day (Missing values filled)

activity\_steps\_day <- activity2 %>%   
 group\_by(date) %>%  
 summarise(steps\_day=sum(steps))

ggplot(data=activity\_steps\_day, aes(activity\_steps\_day$steps\_day)) +   
 geom\_histogram(col="red",   
 aes(fill=..count..)) +  
 scale\_fill\_gradient("Count", low = "red", high = "green") +  
 labs(title="Histogram steps per day") +  
 labs(x="Steps", y="Count Days")

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



### Mean and median of total number of steps taken per day (Missing values filled)

activity\_steps\_day %>%   
 summarise(steps\_median=median(steps\_day))

## # A tibble: 1 × 1  
## steps\_median  
## <dbl>  
## 1 10656

activity\_steps\_day %>%   
 summarise(steps\_mean=mean(steps\_day))

## # A tibble: 1 × 1  
## steps\_mean  
## <dbl>  
## 1 10751.74

**Before filling missing values the mean and the median were almost equal (median=10765, mean=70766) After filling the missing values the Mean and median has changed, the mean is now greater than median by small difference.**

## Pattern differences in activity between weekdays and weekends

### define day of the week in the dataset and recreate the average 5-minute interval dataset

activity2$day <- weekdays(as.Date(activity$date))  
  
activity2$wday <- "weekday"  
activity2$wday[activity2$day=="Saturday"] <- "weekend"  
activity2$wday[activity2$day=="Sunday"] <- "weekend"  
  
  
activity\_steps\_hour <- activity2 %>%   
 group\_by(interval, wday) %>%  
 summarise(steps\_interval=mean(steps))

### panel plot containing a time series plot of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all weekday days or weekend days (y-axis).

ggplot(data=activity\_steps\_hour, aes(x=interval, y=steps\_interval)) + geom\_line() +  
 labs(title="Time series steps taken on average per interval") +  
 labs(x="5 min Interval", y="Steps") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 facet\_wrap(~ wday)

