Coursera\_repdata\_project1

# Reproducible Research: Peer Assessment 1

Loading and preprocessing the data

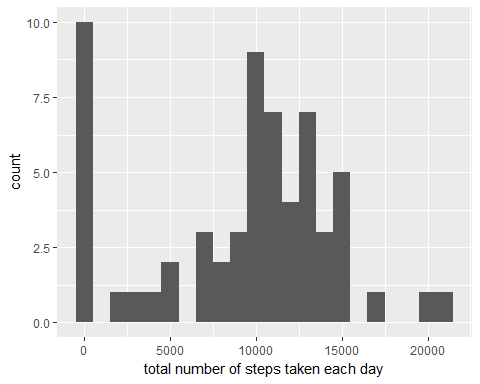
setwd("E:/Anuja/Study/DS\_Specialization/Course 5")  
data <- read.csv("activity.csv")

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

library(ggplot2)

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

total.steps <- tapply(data$steps, data$date, FUN=sum, na.rm=TRUE)  
qplot(total.steps, binwidth=1000, xlab="total number of steps taken each day")



mean(total.steps, na.rm=TRUE)

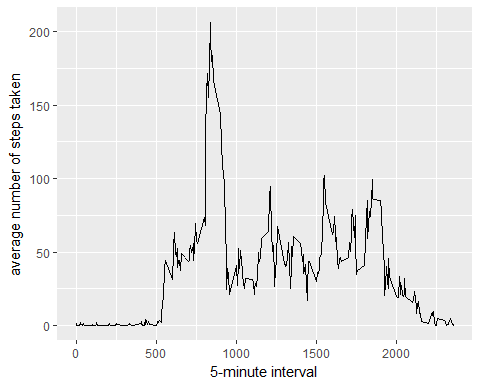
## [1] 9354.23

median(total.steps, na.rm=TRUE)

## [1] 10395

# What is the average daily activity pattern?

library(ggplot2)  
averages <- aggregate(x=list(steps=data$steps), by=list(interval=data$interval),  
 FUN=mean, na.rm=TRUE)  
ggplot(data=averages, aes(x=interval, y=steps)) +  
 geom\_line() +  
 xlab("5-minute interval") +  
 ylab("average number of steps taken")



# On average across all the days in the dataset, the 5-minute interval contains the maximum number of steps?

averages[which.max(averages$steps),]

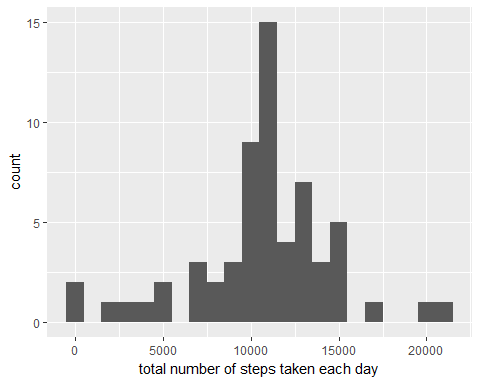
# Imputing missing values

missing <- is.na(data$steps)  
# How many missing  
table(missing)

## missing  
## FALSE TRUE   
## 15264 2304

# Replace each missing value with the mean value of its 5-minute interval

fill.value <- function(steps, interval) {  
 filled <- NA  
 if (!is.na(steps))  
 filled <- c(steps)  
 else  
 filled <- (averages[averages$interval==interval, "steps"])  
 return(filled)  
}  
filled.data <- data  
filled.data$steps <- mapply(fill.value, filled.data$steps, filled.data$interval)  
  
total.steps <- tapply(filled.data$steps, filled.data$date, FUN=sum)  
qplot(total.steps, binwidth=1000, xlab="total number of steps taken each day")



mean(total.steps)

## [1] 10766.19

median(total.steps)

## [1] 10766.19

# Are there differences in activity patterns between weekdays and weekends?

weekday.or.weekend <- function(date) {  
 day <- weekdays(date)  
 if (day %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))  
 return("weekday")  
 else if (day %in% c("Saturday", "Sunday"))  
 return("weekend")  
 else  
 stop("invalid date")  
}  
filled.data$date <- as.Date(filled.data$date)  
filled.data$day <- sapply(filled.data$date, FUN=weekday.or.weekend)  
averages <- aggregate(steps ~ interval + day, data=filled.data, mean)  
ggplot(averages, aes(interval, steps)) + geom\_line() + facet\_grid(day ~ .) +  
 xlab("5-minute interval") + ylab("Number of steps")

