Assignment 1

Duncan

Sunday, February 15, 2015

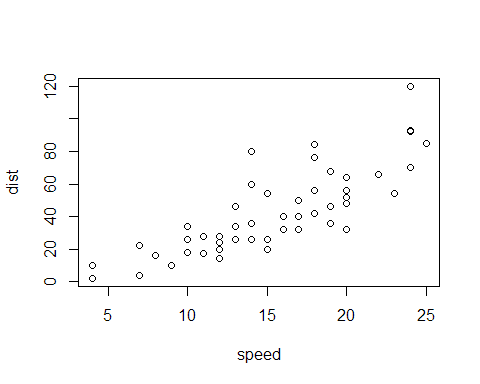
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

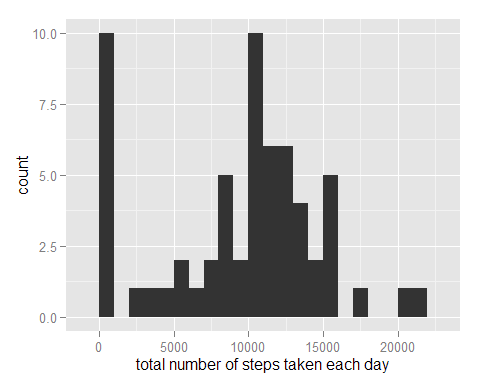
# Reproducible Research: Peer Assessment 1

## Loading and preprocessing the data

data <- read.csv("activity.csv")

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

library(ggplot2)  
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)

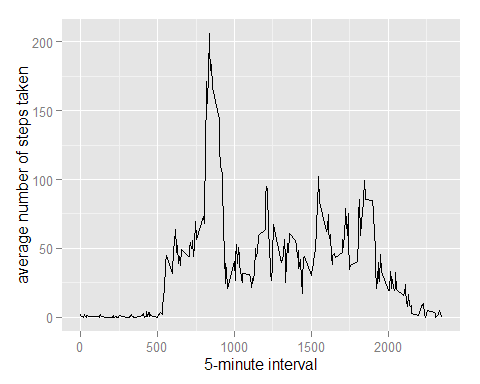
## [1] 9354.23

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),]

## interval steps  
## 104 835 206.1698

## Imputing missing values

There are many days/intervals where there are missing values (coded as NA). The presence of missing days may introduce bias into some calculations or summaries of the data.

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

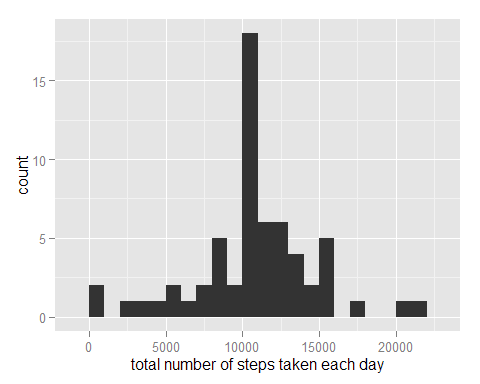
## missing  
## FALSE TRUE   
## 15264 2304

All of the missing values are filled in with mean value for that 5-minute interval.

# 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)

Now, using the filled data set, let's make a histogram of the total number of steps taken each day and calculate the mean and median total number of steps.

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

Mean and median values are higher after imputing missing data. The reason is that in the original data, there are some days with steps values NA for any interval. The total number of steps taken in such days are set to 0s by default. However, after replacing missing steps values with the mean steps of associated interval value, these 0 values are removed from the histogram of total number of steps taken each day.

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

First, let's find the day of the week for each measurement in the dataset. In this part, we use the dataset with the filled-in values.

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)

Now, let's make a panel plot containing plots of average number of steps taken on weekdays and weekends.

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")

