Loading and preprocessing the data

==================================

Show any code that is needed to

# 1.Load the data (i.e. read.csv())

To load the data I use the read.csv() command.

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

To see what the data look like, I use the str(), summary(), and head() commands:

str(activity)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

summary(activity)

## 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)

## 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

# 2.Process/transform the data (if necessary) into a format suitable for your analysis

For the following tasks, we will need to remove missing values. So we create a second version of the data without missing values.

act.complete <- na.omit(activity)

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

For this part of the assignment, you can ignore the missing values in the dataset.

1.Calculate the total number of steps taken per day

I use the dplyr library to collpase the data by day, creating the sum of steps.

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))  
summary(act.day)

## date steps   
## 2012-10-02: 1 Min. : 41   
## 2012-10-03: 1 1st Qu.: 8841   
## 2012-10-04: 1 Median :10765   
## 2012-10-05: 1 Mean :10766   
## 2012-10-06: 1 3rd Qu.:13294   
## 2012-10-07: 1 Max. :21194   
## (Other) :47

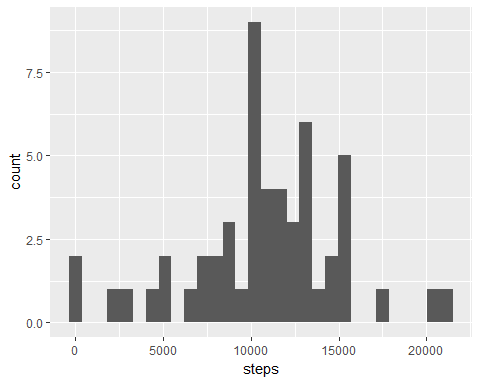
# 2. If you do not understand the difference between a histogram and a barplot, research the difference between them. Make a histogram of the total number of steps taken each day

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 3.3.2

qplot(steps, data=act.day)

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



3. Calculate and report the mean and median of the total number of steps taken per day  
======================================================================================

I use the mean() and median() functions.

mean(act.day$steps)

## [1] 10766.19

median(act.day$steps)

## [1] 10765

# What is the average daily activity pattern?

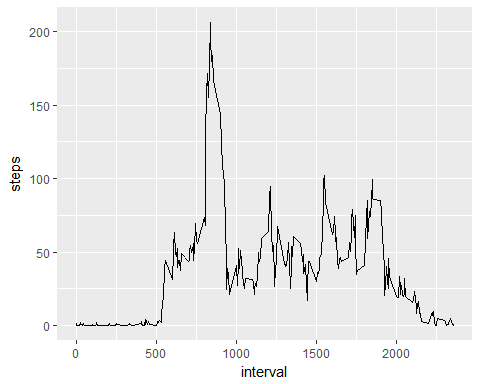
1. Make a time series plot (i.e. type = “l”) of the 5-minute interval (x-axis) and the average number of steps taken, averaged across all days (y-axis) ======================================================================================================

First we create a data frame in which steps are aggregated into averages within each 5 minute interval:

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

Next we plot the average daily steps against the intervals:

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



2.Which 5-minute interval, on average across all the days in the dataset, contains the maximum number of steps? ======================================================================================================

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  
## <int> <dbl>  
## 1 835 206.1698

# Imputing missing values

# Note that there are a number of 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.

1. Calculate and report the total number of missing values in the dataset (i.e. the total number of rows with NAs)  
=================================================================================================

The total number of rows with NAs is equal to the difference between the number of rows in the raw data and the number of rows in the data with only complete cases:

nrow(activity)-nrow(act.complete)

## [1] 2304

2.Devise a strategy for filling in all of the missing values in the dataset. The strategy does not need to be sophisticated. For example, you could use the mean/median for that day, or the mean for that 5-minute interval, etc. ======================================================================================================

Some days have no data, so it is not feasible to replace missing values with the day’s mean. Instead, I replace missing values with the mean number of steps for each interval across all of the days. The act.int data frame contains these means. I start by merging the act.int data with the raw data:

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

3.Create a new dataset that is equal to the original dataset but with the missing data filled in. ===================================================================================================

If steps is NA, I replace the value with the mean number of steps for the interval:

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

1. Make a histogram of the total number of steps taken each day and Calculate and report the mean and median total number of steps taken per day. Do these values differ from the estimates from the first part of the assignment? What is the impact of imputing missing data on the estimates of the total daily number of steps? ====================================================================================================

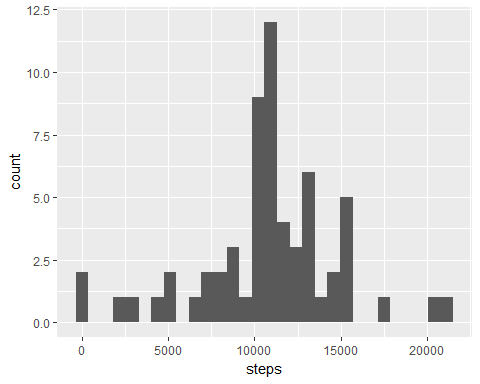
I first create a dataset with the total number of steps per day using the imputed data:

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

Then I generate the histogram and summary statistics:

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

The mean appears to be unaffected by this simple data imputation. The median is smaller.

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

For this part the weekdays() function may be of some help here. Use the dataset with the filled-in missing values for this part.

1. 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.

======================================================================================================= I convert the date variable to the date class, then use the weekdays() function to generate the day of the week of each date. I create a binary factor to indicate the two weekend days:

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")

2.Make a panel plot containing 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 weekday days or weekend days (y-axis). See the README file in the GitHub repository to see an example of what this plot should look like using simulated data.

First I create separate data frames for weekends and weekdays:

act.weekday <- act.impute[act.impute$weekend=="Weekday",]  
act.weekend <- act.impute[act.impute$weekend=="Weekend",]

Then for each one, I find the mean number of steps across days for each 5 minute interval:

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"

I append the two data frames together, and I make the two time series plots:

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 ~ .)

