# RepData peer Assesssment 1

## Loading and preprocessing the data

1. Load the Data

holder = read.csv("activity.csv") ##having unzipped the assignment dat into your working directory.

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

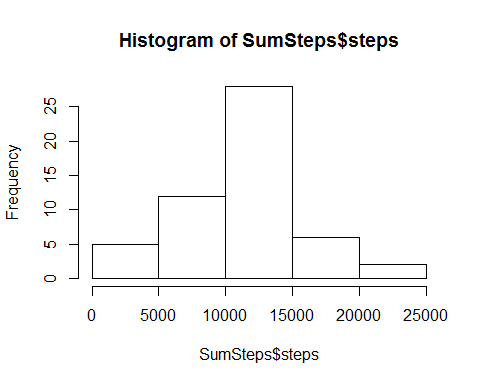
## What is mean total number of steps taken per day?

1. Calculate the total number of steps taken per day, ignoring the missing values in the dataset

SumSteps <- aggregate(steps ~ date, data = holder, sum, na.rm = TRUE)

1. Make a histogram of the total number of steps taken each day

hist(SumSteps$steps)



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

mean(SumSteps$steps)

## [1] 10766.19

median(SumSteps$steps)

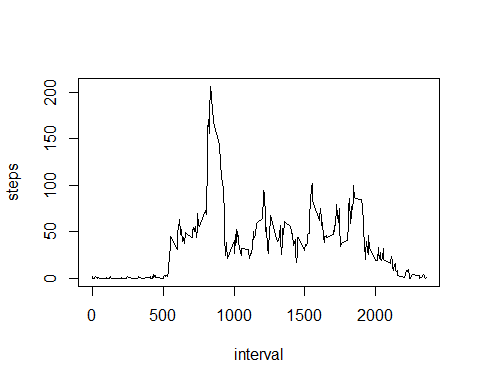
## [1] 10765

The mean total number of steps taken per day is 10766.19 steps. The median total number of steps taken per day is 10765 steps.

## What is the average daily activity pattern?

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

StepsInterval <- aggregate(steps ~ interval, data = holder, mean, na.rm = TRUE)  
plot(steps ~ interval, data = StepsInterval, type = "l")



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

StepsInterval[which.max(StepsInterval$steps), ]$interval

## [1] 835

Answer: The 835th interval.

## Imputing missing values

Note that there are a number of days/intervals where there are missing values. 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

sum(is.na(holder$steps))

## [1] 2304

Answer: 2304 rows are missing.

1. 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.

Method: Used the Mean, created function to calculate mean steps for particular 5-minute interval

StepsInterval2 <- function(interval) {  
 StepsInterval[StepsInterval$interval == interval, ]$steps  
}

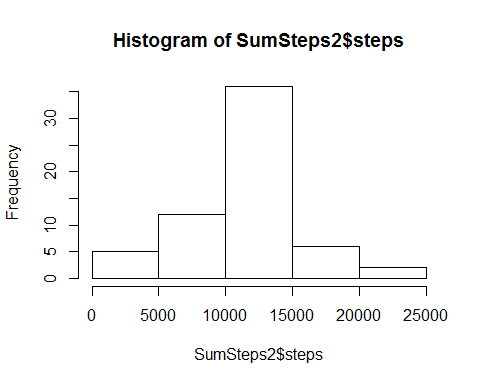
1. Create a new dataset from the original data set and filling the missing values with the mean

Filled <- holder # Make a new dataset with the original data  
count = 0 # Count the number of data filled in  
for (i in 1:nrow(Filled)) {  
 if (is.na(Filled[i, ]$steps)) {  
 Filled[i, ]$steps <- StepsInterval2(Filled[i, ]$interval)  
 count = count + 1  
 }  
}  
cat("Total ", count, "NA values were filled.")

## Total 2304 NA values were filled.

4.a. 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.

SumSteps2 <- aggregate(steps ~ date, data = Filled, sum)  
hist(SumSteps2$steps)



mean(SumSteps2$steps)

## [1] 10766.19

median(SumSteps2$steps)

## [1] 10766.19

The mean total number of steps taken per day is 10766 steps. The median total number of steps taken per day is 10766 steps.

4.b. 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?

Answer: The mean value is the same as the value before imputing missing data because we put the mean value for that particular 5-min interval. The median value is now the same as the mean since we have now evened out the spread with inserting the Mean into the missing values.

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

Using weekdays() function and the "Filled" dataset.

1. Create a new factor variable in the dataset with two levels - "weekday" and "weedkend" indicating whether a given date is a weekday or weekend day.

# convert date from string to Date class  
Filled$date <- as.Date(Filled$date, "%Y-%m-%d")  
# add a new column indicating day of the week   
Filled$day <- weekdays(Filled$date)  
# add a new column called day type and initialize to weekday  
Filled$day\_type <- c("weekday")  
# If day is Saturday or Sunday, make day\_type as weekend  
for (i in 1:nrow(Filled)){  
 if (Filled$day[i] == "Saturday" || Filled$day[i] == "Sunday"){  
 Filled$day\_type[i] <- "weekend"  
 }  
}  
# convert day\_time from character to factor  
Filled$day\_type <- as.factor(Filled$day\_type)  
# aggregate steps as interval to get average number of steps in an interval across all days  
TableFilledSteps <- aggregate(steps ~ interval+day\_type, Filled, mean)

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

library(ggplot2)  
qplot(interval, steps, data=TableFilledSteps, geom=c("line"), xlab="Interval",   
 ylab="Number of steps", main="") + facet\_wrap(~ day\_type, ncol=1)

