# Reproducible Research: Peer Assessment 1

## Loading libraries

library(lattice)  
library(ggplot2)

## Loading and preprocessing the data

1. Loading the data

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

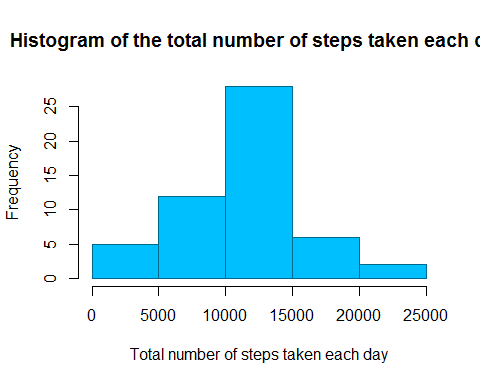
1. Preprocessing the data

# convert date from string to date type  
df$date <- as.Date(df$date, format = "%Y-%m-%d")   
  
# create dataframe with total steps per day  
df.day <- aggregate(df$steps, by = list(df$date), sum)  
names(df.day)[1] <- "day"  
names(df.day)[2] <- "steps"  
  
# create dataframe with total steps per interval  
df.interval <- aggregate(df$steps, by = list(df$interval), sum, na.rm = TRUE, na.action = NULL)  
names(df.interval)[1] <- "interval"  
names(df.interval)[2] <- "steps"  
  
# create dataframe with mean steps per interval  
df.mean.interval <- aggregate(df$steps, by = list(df$interval), mean, na.rm = TRUE, na.action = NULL)  
names(df.mean.interval)[1] <- "interval"  
names(df.mean.interval)[2] <- "mean.steps"

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

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

title = "Histogram of the total number of steps taken each day"  
xlab = "Total number of steps taken each day"  
hist(df.day$steps, main = title, xlab = xlab, col = 'deepskyblue', border = 'deepskyblue4')



1. The mean and median total number of steps taken per day

#Mean number of steps per day  
mean(df.day$steps, na.rm = TRUE)

## [1] 10766

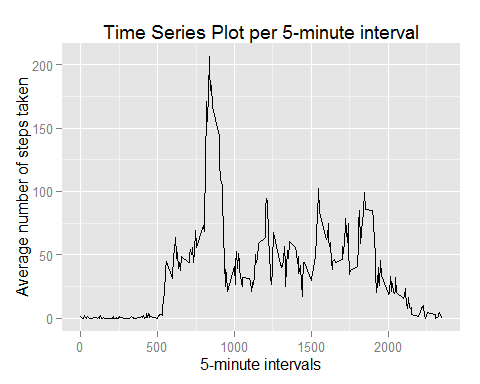
#Median number of steps per day  
median(df.day$steps, na.rm = TRUE)

## [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)

title="Time Series Plot per 5-minute interval"  
xlab="5-minute intervals"  
ylab="Average number of steps taken"  
qplot(interval, mean.steps, data=df.mean.interval, geom="line", main=title, xlab=xlab, ylab=ylab)



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

df.mean.interval[which.max(df.mean.interval$mean.steps), 1]

## [1] 835

## Inputing missing values

1. The total number of missing values

#Total number of missing values in the dataset  
sum(is.na(df$steps))

## [1] 2304

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.

Strategy: Use the mean for the interval to fill missing values.

# merge df and df.mean.interval dataframes  
df.missing <- merge(df, df.mean.interval, by = "interval", sort = FALSE)  
# sort on date and interval  
df.missing <- df.missing[with(df.missing, order(date, interval)), ]   
# replace in steps column NA with value in mean.steps column  
df.missing$steps[is.na(df.missing$steps)] <- df.missing$mean.steps[is.na(df.missing$steps)]  
# remove the column with the mean  
df.missing$mean.steps <- NULL   
# report  
head(df.missing)

## interval steps date  
## 1 0 1.71698 2012-10-01  
## 63 5 0.33962 2012-10-01  
## 128 10 0.13208 2012-10-01  
## 205 15 0.15094 2012-10-01  
## 264 20 0.07547 2012-10-01  
## 327 25 2.09434 2012-10-01

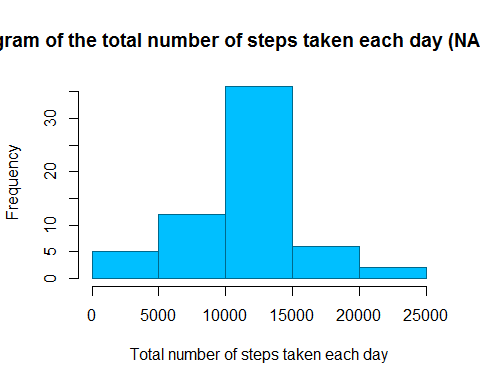
# round up  
df.missing$steps <- round(df.missing$steps, digits = 0)

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

df.new <- df.missing[, c(2, 3, 1)]

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?

# create dataframe with total steps per day  
df.day.new <- aggregate(df.new$steps, by = list(df.new$date), sum)  
names(df.day.new)[1] <- "day"  
names(df.day.new)[2] <- "steps"  
  
#Histogram of the total number of steps taken each day  
title="Histogram of the total number of steps taken each day (NA replaced)"  
xlab = "Total number of steps taken each day"  
hist(df.day.new$steps, main = title, xlab = xlab, col='deepskyblue', border='deepskyblue4')



#The mean and median total number of steps taken per day  
mean(df.day.new$steps, na.rm = TRUE)

## [1] 10766

# Median number of steps per day  
median(df.day.new$steps, na.rm = TRUE)

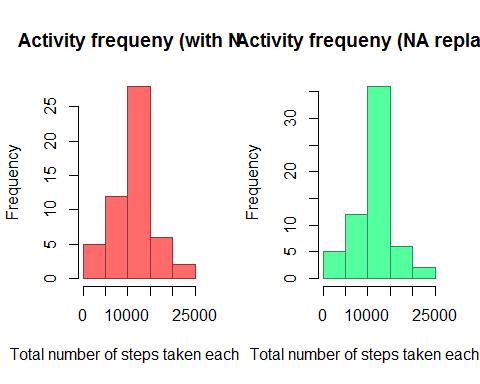
## [1] 10762

The Mean is equal to the estimates from the first part of the assignment.

The Median is slightly lower when compared to the first part of the assignment.

The histogram shows a similar shape as before with overall higher frequencies due to the NA being replaced in the new histogram. See also this side by side plot:

par(mfrow = c(1, 2))  
xlab = "Total number of steps taken each day"  
hist(df.day$steps, main = "Activity frequeny (with NA)", xlab = xlab, col='indianred1', border='indianred4')  
hist(df.day.new$steps, main = "Activity frequeny (NA replaced)", xlab = xlab, col='seagreen1', border='seagreen4')



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

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

# create copy of the dataframe  
df.new.2 <- df.new  
# make sure we use English date names  
Sys.setlocale("LC\_TIME", "English")

## [1] "English\_United States.1252"

# create a factor with the names of the days for all dates  
df.new.2$weekdays <- factor(format(df.new.2$date, "%A"))  
# the day names fe  
levels(df.new.2$weekdays)

## [1] "Friday" "Monday" "Saturday" "Sunday" "Thursday" "Tuesday"   
## [7] "Wednesday"

# replace the levels  
levels(df.new.2$weekdays) <- list(weekday = c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"),   
 weekend = c("Saturday", "Sunday"))

1. 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). The plot should look something like the following, which was creating using simulated data:

df.new.2.mean.interval <- aggregate(df.new.2$steps, by = list(df.new.2$weekdays, df.new.2$interval), mean, na.rm = TRUE, na.action = NULL)  
names(df.new.2.mean.interval)[1] <- "weekday"  
names(df.new.2.mean.interval)[2] <- "interval"  
names(df.new.2.mean.interval)[3] <- "mean.steps"  
xyplot(df.new.2.mean.interval$mean.steps ~ df.new.2.mean.interval$interval | df.new.2.mean.interval$weekday,  
 layout = c(1, 2), type = "l", xlab = "Interval", ylab = "Number of steps")

