# Coursera Reproducible research Assignment 1

### Import data from local drive as CSV format and do a quick discovery

df <- read.csv('activity.csv', header = TRUE)  
  
#data discovery   
str(df)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : chr "2012-10-01" "2012-10-01" "2012-10-01" "2012-10-01" ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

min(df$date)

## [1] "2012-10-01"

max(df$date)

## [1] "2012-11-30"

max(df$interval)

## [1] 2355

head (df)

## 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

# number of NA rows out of the 17568 obs.  
sum( as.numeric(as.logical(is.na(df$steps))))

## [1] 2304

### 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.

Calculate the total number of steps taken per day

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

we will use library(dplyr)

library(dplyr)

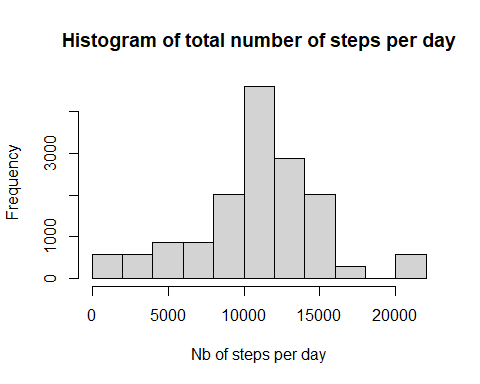
## Warning: package 'dplyr' was built under R version 4.0.3

##   
## 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

# create new dataset by ignoring null values and adding the daily sum of steps  
df2 <- df %>% subset(steps != 'NA') %>% group\_by(date) %>% mutate (steps\_p\_day = sum(steps))  
  
# generate related histogram  
hist (df2$steps\_p\_day  
 , main = "Histogram of total number of steps per day"  
 , xlab = 'Nb of steps per day')



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

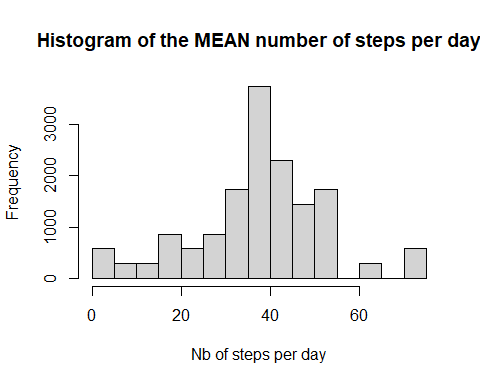
(df2$mean\_steps\_p\_day\_tot <- mean(df2$steps\_p\_day))

## [1] 10766.19

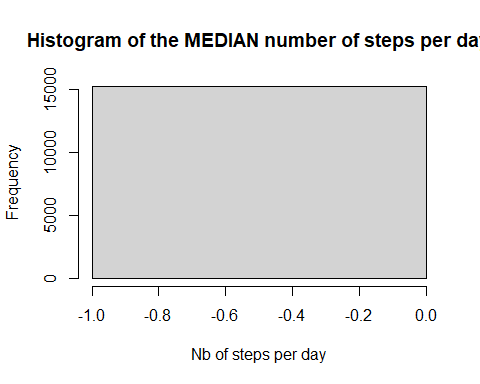
(df2$med\_steps\_p\_day\_tot <- median(df2$steps\_p\_day))

## [1] 10765

# generate the histogram of the mean and median of the total steps per day  
df2 <- df2 %>% group\_by(date) %>% mutate (mean\_steps\_p\_day = mean(steps), med\_steps\_p\_day = median(steps))  
  
hist (df2$mean\_steps\_p\_day  
 , main = "Histogram of the MEAN number of steps per day"  
 , xlab = 'Nb of steps per day')



hist (df2$med\_steps\_p\_day  
 , main = "Histogram of the MEDIAN number of steps per day"  
 , xlab = 'Nb of steps per day')

 we can see from above charts that the mode of the average number of steps per day is between 35 and 40 when the median is at 0

### What is the average daily activity pattern?

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)

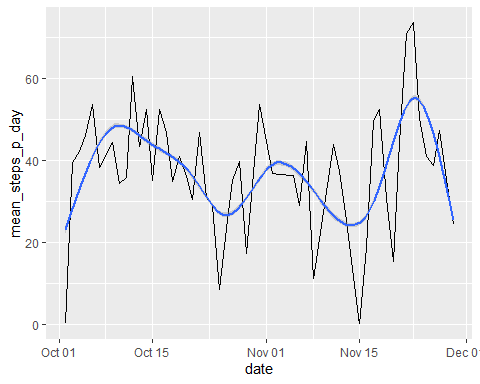
we will use library ggplot2

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.3

# change date field into date type and identify x axis range  
df2$date <- as.Date(df2$date, "%Y-%m-%d")  
mx\_dt <- max(df2$date)  
mn\_dt <- min(df2$date)  
  
# graph steps per days  
ggplot(df2, aes(x = date, y = mean\_steps\_p\_day)) +  
 xlim (mn\_dt, mx\_dt) +  
 geom\_line () +  
 geom\_smooth ()

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



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

df2 <- df2 %>% group\_by(interval) %>% mutate (steps\_p\_interv\_mean = mean(steps))  
unique(df2[df2$steps\_p\_interv\_mean == max(df2$steps\_p\_interv\_mean), 3])

## # A tibble: 1 x 1  
## # Groups: interval [1]  
## interval  
## <int>  
## 1 835

therefore the interval with the maximum steps is 835

### Imputing missing Values

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

# number of missing values  
sum (as.numeric(as.logical(is.na(df$steps))))

## [1] 2304

# Percentage of steps with value = 0 over non null steps values  
sum (as.numeric(as.logical(!is.na(df$steps) & df$steps == 0)))/sum (as.numeric(as.logical(!is.na(df$steps))))

## [1] 0.7215671

Devise a strategy for filling in all of the missing values in the dataset - we will use the mean steps per interval to fill in the gaps (NA)

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

# start by simplifying the df2 with the mean data   
df\_mean <- df2 %>% select (interval,steps\_p\_interv\_mean) %>% distinct (interval,steps\_p\_interv\_mean)  
  
# create joined (imputed) dataset by joining df with df\_mean on interval to include the mean per interval on missing data  
df\_joined <- inner\_join(df,df\_mean, by = "interval")  
  
# fill in the gaps with 'mean steps per interval' in df\_joined   
df\_joined$steps [which (is.na(df\_joined$steps))] <- df\_joined$steps\_p\_interv\_mean [which (is.na(df\_joined$steps))]

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

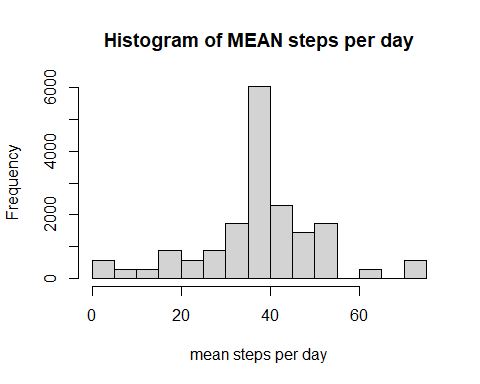
# create new columns with sum, mean, med steps for the imputed dataset  
df\_joined <- df\_joined %>%   
 group\_by(date) %>%   
 mutate (steps\_p\_day\_sum = sum(steps),   
 steps\_p\_day\_mean = mean(steps),  
 steps\_p\_day\_med = median(steps))  
  
# Percentage of steps with value = 0 over non null steps values  
sum (as.numeric(as.logical(!is.na(df$steps) & df\_joined$steps == 0)))/sum (as.numeric(as.logical(!is.na(df\_joined$steps))))

## [1] 0.6269353

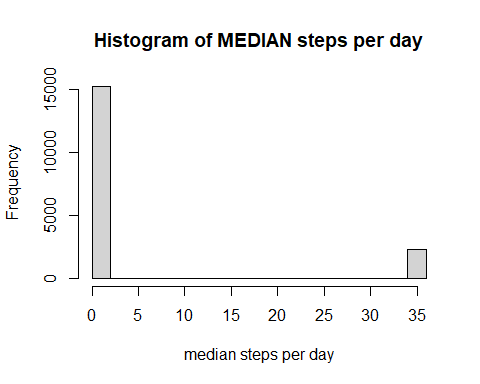
df\_joined %>%   
 select (date,steps\_p\_day\_sum,steps\_p\_day\_mean,steps\_p\_day\_med) %>%   
 distinct (date,steps\_p\_day\_sum,steps\_p\_day\_mean,steps\_p\_day\_med)

## # A tibble: 61 x 4  
## # Groups: date [61]  
## date steps\_p\_day\_sum steps\_p\_day\_mean steps\_p\_day\_med  
## <chr> <dbl> <dbl> <dbl>  
## 1 2012-10-01 10766. 37.4 34.1  
## 2 2012-10-02 126 0.438 0   
## 3 2012-10-03 11352 39.4 0   
## 4 2012-10-04 12116 42.1 0   
## 5 2012-10-05 13294 46.2 0   
## 6 2012-10-06 15420 53.5 0   
## 7 2012-10-07 11015 38.2 0   
## 8 2012-10-08 10766. 37.4 34.1  
## 9 2012-10-09 12811 44.5 0   
## 10 2012-10-10 9900 34.4 0   
## # ... with 51 more rows

hist (df\_joined$steps\_p\_day\_mean  
 , main = "Histogram of MEAN steps per day"  
 , xlab = 'mean steps per day')



hist (df\_joined$steps\_p\_day\_med  
 , main = "Histogram of MEDIAN steps per day"  
 , xlab = 'median steps per day')



Do these values differ from the estimates from the first part of the assignment?

# sum steps per day prior to imputation  
prior <- sum(df2$steps\_p\_day)  
# sum steps per day after imputation  
after <- sum(df\_joined$steps\_p\_day\_sum)  
# increase in steps per day after vs prior imputation (in pct)  
(after/prior-1) \* 100

## [1] 15.09434

# median steps per day after imputation  
(df\_joined$med\_steps\_p\_day\_tot <- median(df\_joined$steps\_p\_day\_sum))

## [1] 10766.19

In order to impute the missing data, we have used the mean steps per interval - regarding the mean histogram, the overall shape is very similar to the one without the imputation, except that the mode is way higher. - regarding the median data, the first histogram is only 0 as more than 50% of the data which is not null have 0 values - regarding the sum of daily steps, we can see an increase of 15.09% due to the imputation

What is the impact of imputing missing data on the estimates of the total daily number of steps? sum of steps per day has drastically increased has null values have been replaced by the interval mean usually higher than 0

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

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.

# change date field into date data type   
df\_joined$date <- as.Date(df\_joined$date, "%Y-%m-%d")  
# add day name field  
df\_joined$wd\_nm <- weekdays(df\_joined$date)  
# add week day flag with 'wd' for week day as default  
df\_joined$wd\_flag <- 'wd'  
# correct flag default with proper value 'we' for week end days  
df\_joined$wd\_flag[df\_joined$wd\_nm == 'Saturday' | df\_joined$wd\_nm == 'Sunday'] <- 'we'

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)

df\_joined <- df\_joined %>% group\_by (wd\_flag) %>% mutate (mean\_wd = mean(steps))  
  
mx\_int <- max(df\_joined$interval)  
lab\_char\_vect <- c (wd = 'weekday', we = 'weekend')  
  
ggplot(df\_joined, aes(x = interval, y = steps)) +  
 xlim (0, mx\_int) +  
 geom\_line (color = 'steelblue') +  
 theme(strip.background = element\_rect(fill="antiquewhite"),  
 panel.background = element\_rect(fill= 'white')) +  
 facet\_grid (wd\_flag ~., labeller = labeller (wd\_flag = lab\_char\_vect)) +  
 labs (title = 'average number of steps taken, across weekday or weekend days', x = 'Interval', y = 'Number of steps')

