# Loading and preprocessing the data

Following code uncompresses the data file and stores it in a R dataset.

filename <- unzip("activity.zip")  
activity\_ds <- read.csv(filename, stringsAsFactors = FALSE)  
str(activity\_ds)

## '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 ...

Dates are not formatted to a Date class. Following code converts the dates into proper format.

activity\_ds$date <- as.Date(activity\_ds$date)  
str(activity\_ds)

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

Column ‘Steps’ has some missing values. Following code counts the number of missing values.

sum(is.na(activity\_ds$steps))

## [1] 2304

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

There are 2304 missing values. These values will be excluded for this part of the assignment.

Dataset ‘activity\_1’ is created for this purpose.

activity\_1<-activity\_ds[which(!is.na(activity\_ds$steps)),]

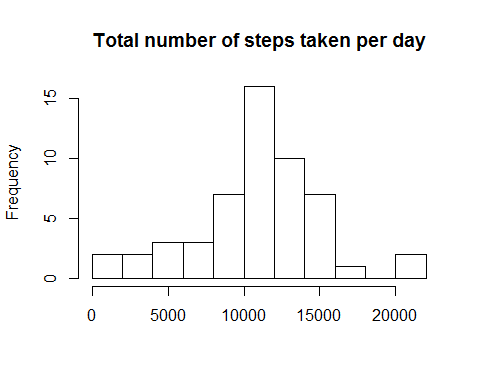
Number of steps taken is measured in 5-minute intervals. To caluculate the total number of steps taken for each day, data will be aggregated by day.

steps\_perday1<-tapply(activity\_1$steps, activity\_1$date, sum)

steps\_perday1 dataset contains total number of steps taken for each day in October and November (total of 53 days)

Following code makes a histogram of the total number of steps taken each day.

hist(steps\_perday1,10, main = "Total number of steps taken per day", xlab = "")



plot of Total number of steps taken per day

The mean total number of steps in one day is 1.0766 × 104. The median of the total steps is 10765.

mean(steps\_perday1)

## [1] 10766.19

median(steps\_perday1)

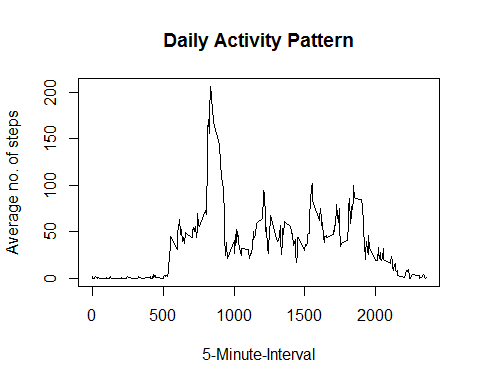
## [1] 10765

# What is the average daily activity pattern?

To explore the data throughout the day, dataset needs to be aggregated by intervals. A per interval array is created to do this and a time series plot will also be created.

The x-axis labels are names of the intervals in the dataset. The coding of the interval names is as follows: 1000 will be considered as 10:00, 1500 will be considered as 15:00. x-axis is a 24-hour-day starting from 12 am to 11:59:59 pm.

daily\_activity<-tapply(activity\_1$steps, activity\_1$interval, mean)  
plot(y = daily\_activity, x = names(daily\_activity), type = "l", xlab = "5-Minute-Interval",   
 main = "Daily Activity Pattern", ylab = "Average no. of steps")

 plot of daily average activity

Interval with maximum average number of steps throughout the day is 835 with steps.

daily\_activity[daily\_activity==max(daily\_activity)]

## 835   
## 206.1698

# Imputing missing values

As mentioned before there are many os days/intervals where there are missing values (these are coded as NA). Missing days can affect the summary data calculations.

sum(is.na(activity\_ds$steps))

## [1] 2304

sum(is.na(activity\_ds))

## [1] 2304

The total number of missing values for the column ‘steps’ is the same as the total number missing in the whole dataset. This implies intervals and the dates do not have any missing values.

Following code creates a new dataset with the missing values filled in.

activity\_2<-activity\_ds  
activity\_2[which(is.na(activity\_2$steps)),1]<-  
 daily\_activity[as.character(activity\_2[which(is.na(activity\_2$steps)),3])]

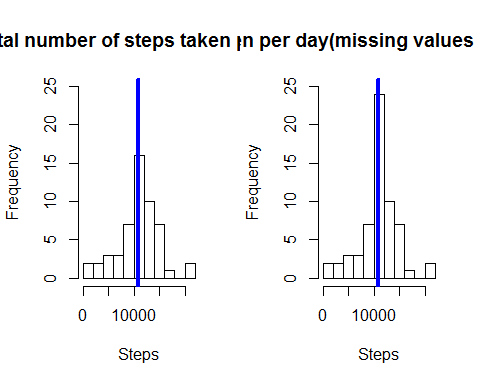
There are no missing values in the new dataset.

sum(is.na(activity\_2))

## [1] 0

Following code creates a histogram using the new dataset.

steps\_perday2<-tapply(activity\_2$steps, activity\_2$date, sum)  
par(mfrow=c(1,2))  
hist(steps\_perday1,10, main = "Total number of steps taken per day", xlab = "Steps", ylim =c(0, 25))  
abline(v = median(steps\_perday1), col = 4, lwd = 4)  
hist(steps\_perday2,10, main = "Total number of steps taken per day(missing values replaced with mean of interval)", xlab = "Steps",ylim =c(0, 25))  
abline(v = median(steps\_perday2), col = 4, lwd = 4)



plot of chunk histogram both

Median and the mean of the filled in dataset:

mean(steps\_perday2)

## [1] 10766.19

median(steps\_perday2)

## [1] 10766.19

Imputting missing data has a minimal impact. The median seems to be changing but by just over one step.

mean(steps\_perday2)-mean(steps\_perday1)

## [1] 0

median(steps\_perday2)-median(steps\_perday1)

## [1] 1.188679

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

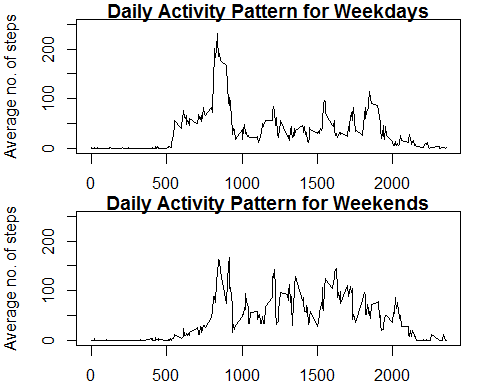
In the code below variables “week\_day” and “week\_end” denote if the day is weekday or weekend.

activity\_2$wd<-weekdays(activity\_2$date)  
activity\_2$fwd<- as.factor(c("week\_end", "week\_day"))  
activity\_2[activity\_2$wd == "Sunday" | activity\_2$wd == "Saturday" ,5]<- factor("week\_end")  
activity\_2[!(activity\_2$wd == "Sunday" | activity\_2$wd == "Saturday"),5 ]<- factor("week\_day")

Following code creates two aggregated arrays for the total number of steps taken per 5 minute time interval for weekdays and weekends and creates a graph to compare if there are any differences.

Please note that the plot has been created in the base system.

activity\_2\_we <- subset(activity\_2, fwd == "week\_end")   
activity\_2\_wd <- subset(activity\_2, fwd == "week\_day")   
daily\_activity\_we<-tapply(activity\_2\_we$steps, activity\_2\_we$interval, mean)  
daily\_activity\_wd<-tapply(activity\_2\_wd$steps, activity\_2\_wd$interval, mean)  
par(mfrow=c(2,1), mar=c(2,4,1,1))  
plot(y = daily\_activity\_wd, x = names(daily\_activity\_wd), type = "l", xlab = "5 Minute Interval",main = "Daily Activity Pattern for Weekdays", ylab = "Average no. of steps",ylim =c(0, 250))  
plot(y = daily\_activity\_we, x = names(daily\_activity\_we), type = "l", xlab = "5 Minute Interval",main = "Daily Activity Pattern for Weekends", ylab = "Average no. of steps",ylim =c(0, 250))



From the two graphs, we can infer that the distribution throughout the day is very different. The person for whom the data was collected seems to wake up at least an hour later during the weekends. Another fact is that there is a huge amount of steps taken on weekdays which is not the case on Saturdays or Sundays. The whole weekend seems to be evenly distributed with no big deviations during hours when a normal person is expected to be awake. There are more steps on weekend day than on a working day. From this we can infer that the person studied is working and on the days he works goes on a walk.

Please note that the x-axis point labels are the names of the intervals in the dataset. The interval names are coded as follows: e.g. 1000 should be considered as 10:00 and 1500 as 15:00. So x-axis is 24-hour-day starting from 12:00 am and ending at 11:59:59 pm.