### Unzip the data

unzip("repdata-data-activity.zip", overwrite = T)

### Loading the data into R

activity <- read.csv("activity.csv", header = T, sep = ",")

### assigning Null to the variables created below

echo = TRUE df\_summary <- NULL su2 <- NULL su <- NULL mn\_int <- NULL activity2 <- NULL mean\_su2 <- NULL median\_su2 <- NULL activity2\_weekend <- NULL activity2\_weekday <- NULL mean\_activity2\_weekday <- NULL mean\_activity2\_weekend <- NULL

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

### First, the total (sum) of steps is determined for every single date.

echo = TRUE

su <- tapply(activitydate, sum, na.rm=T)

## Hereafter is presented a histogram of the total number of steps taken each day.

echo = TRUE

hist(su, xlab = "sum of steps per day", main = "histogram of steps per day")

### The mean and the median total number of steps taken per day are reported :

echo = TRUE

mean\_su <- round(mean(su))

median\_su <- round(median(su))

print(c("The mean:",mean\_su))

### [1] "The mean:" "9354"

print(c("The median:",median\_su))

### [1] "The median:" "10395"

### what is the average daily activity pattern ?

### A time series plot of the 5-minute interval and the average number of steps taken (averaged across all days) is shown below:

echo = TRUE

mn\_int <- tapply(activityinterval, mean, na.rm=T)

plot(mn\_int ~ unique(activity$interval), type="l", xlab = "5-min interval")

### The 5-minute interval that contains the maximum number of steps is the following

echo = TRUE

mn\_int[which.max(mn\_int)]

### 835

### 206.1698

### Assigning 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.First, in order to visualize in which variable the NAs are:

echo = TRUE

table(is.na(activity) == TRUE)

### FALSE TRUE

### 50400 2304

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

### A total of 2304 NA's are present

### Strategy for filling in all of the missing values in the dataset

### The following strategy is chosen: for any NA is the step variable, the mean (of steps) of the corresponding interval is taken as the replacing value.

### The 'mn\_int' contains the mean for each single interval calculated over the 61 days. The right value coming from 'mn\_int' is going to be used to replace the NA at the same interval.

echo = TRUE ### creation of the dataset that will have no more NAs

activity2 <- activity

for (i in 1:nrow(activity)){ if(is.na(activity$steps[i])){ activity2$steps[i]<- mn\_int[[as.character(activity[i, "interval"])]] } }

### Below is a histogram of the total number of steps taken each day. The mean and median total number of steps taken per day are reported.

echo = TRUE

su2 <- tapply(activity2date, sum, na.rm=T)

hist(su2, xlab = "sum of steps per day", main = "histogram of steps per day")

mean\_su2 <- round(mean(su2))

median\_su2 <- round(median(su2))

### The new values are :

echo = TRUE

print(c("The mean:",mean\_su2))

### [1] "The mean:" "10766"

print(c("The median:",median\_su2))

### [1] "The median:" "10766"

### In order to compare the new values with the “old” values:

echo = TRUE

df\_summary <- rbind(df\_summary, data.frame(mean = c(mean\_su, mean\_su2), median = c(median\_su, median\_su2)))

rownames(df\_summary) <- c("with NA's", "without NA's")

print(df\_summary)

### mean median

### with NA's 9354 10395

### without NA's 10766 10766

### For comparison with NA's and without (see earlier):

echo = TRUE

summary(activity2) ## 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.: 27.00 2012-10-05: 288 3rd Qu.:1766.2  
## Max. :806.00 2012-10-06: 288 Max. :2355.0 ## (Other) :15840

### There are No NA's now

### Do we find any difference in activity patterns between weekdays and weekends?

### A new column is added to the dataframe, this column will contain the factor “weekday days”“ or "weekend days”.

echo = TRUE activity2$weekday <- c("weekday")

activity2[weekdays(as.Date(activity2[, 2])) %in% c("Saturday", "Sunday", "samedi", "dimanche", "saturday", "sunday", "Samedi", "Dimanche"), ][4] <- c("weekend")

table(activity2$weekday == "weekend")

### FALSE TRUE

### 12960 4608

activity2weekday)

### In order to visualize the difference bewteen weekends and days of the week, a new dataframe is created to be usable by the lattice package. First, the data are calculated:

echo = TRUE activity2\_weekend <- subset(activity2, activity2$weekday == "weekend") activity2\_weekday <- subset(activity2, activity2$weekday == "weekday")

mean\_activity2\_weekday <- tapply(activity2\_weekdayinterval, mean) mean\_activity2\_weekend <- tapply(activity2\_weekendinterval, mean)

### Afterwards the dataframe is prepared and the plot is… plotted !

echo = TRUE library(lattice) df\_weekday <- NULL df\_weekend <- NULL df\_final <- NULL df\_weekday <- data.frame(interval = unique(activity2\_weekday$interval), avg = as.numeric(mean\_activity2\_weekday), day = rep("weekday", length(mean\_activity2\_weekday))) df\_weekend <- data.frame(interval = unique(activity2\_weekend$interval), avg = as.numeric(mean\_activity2\_weekend), day = rep("weekend", length(mean\_activity2\_weekend))) df\_final <- rbind(df\_weekday, df\_weekend)

xyplot(avg ~ interval | day, data = df\_final, layout = c(1, 2), type = "l", ylab = "Number of steps")

### we can notice a minor change between the period.