Project 1

# Reproducible Research: Course Project 1

Introduction

This project is an assignment of the Reproducible Research course. It loads a data set, performs some processing, and produces some charts. The data was related with the number of steps taken as recorded by a Fitbit or other similar device.

setwd("~/Cursos/5 - Reproducible Research/Week 1 - Concepts, Ideas, & Structure/Assignment - Course Project 1/Repro\_Research\_-Project\_1")

Item 1 - Code for reading in the dataset and/or processing the data Loading and preprocessing the data. The data was in a file named "activity.csv".

actv<-read.csv("activity.csv", sep = ",")  
str(actv)

## 'data.frame': 17568 obs. of 3 variables:  
## $ steps : int NA NA NA NA NA NA NA NA NA NA ...  
## $ date : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ interval: int 0 5 10 15 20 25 30 35 40 45 ...

summary(actv)

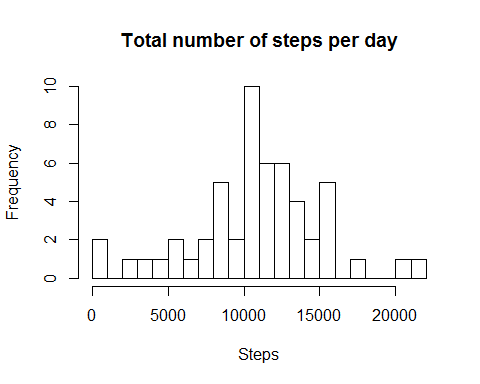
## steps date interval   
## Min. : 0.00 2012-10-01: 288 Min. : 0.0   
## 1st Qu.: 0.00 2012-10-02: 288 1st Qu.: 588.8   
## Median : 0.00 2012-10-03: 288 Median :1177.5   
## Mean : 37.38 2012-10-04: 288 Mean :1177.5   
## 3rd Qu.: 12.00 2012-10-05: 288 3rd Qu.:1766.2   
## Max. :806.00 2012-10-06: 288 Max. :2355.0   
## NA's :2304 (Other) :15840

head(actv)

## 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

Item 2 - Histogram of the total number of steps taken each day The data is presented as a histogram exploring its distribution.

steps\_day <- aggregate(steps ~ date, data = actv, sum, na.rm = T)  
hist(steps\_day$steps, breaks = 20, main = "Total number of steps per day", xlab = "Steps", ylab = "Frequency")



The histogram sugests the distribution of steps in a day. However, estimating visually is not always accurate enough.

Item 3 - Mean and median number of steps taken each day What is mean total number of steps taken per day? Calculating the mean and median of the aggregated steps per day is straightforward.

mean(steps\_day$steps)

## [1] 10766.19

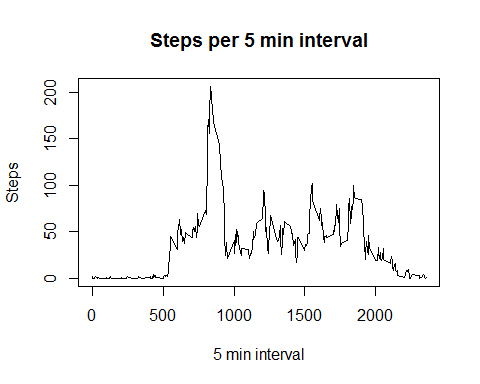
median(steps\_day$steps)

## [1] 10765

Item 4 - Time series plot of the average number of steps taken

To find the average daily activy pattern the data was aggregated the data on mean steps and was then presented as a time series plot.

steps\_interval <- aggregate(steps ~ interval, data = actv, mean, na.rm = T)  
plot( steps\_interval$interval, steps\_interval$steps,type = "l", main = "Steps per 5 min interval", xlab = "5 min interval", ylab = "Steps")



Item 5 - The 5-minute interval that, on average, contains the maximum number of steps

which.max(steps\_interval$steps)

## [1] 104

steps\_interval[104,]

## interval steps  
## 104 835 206.1698

Item 6 - Code to describe and show a strategy for imputing missing data The "Nas" was replaced by the median value in the data set.

sum(is.na(actv$steps))

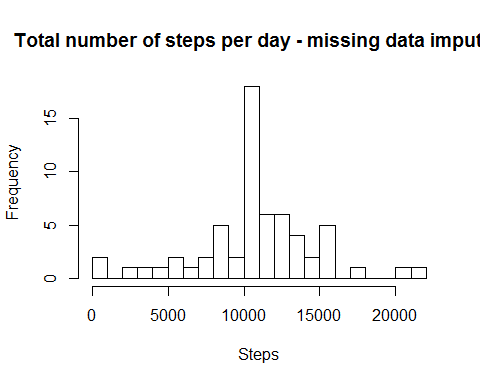
## [1] 2304

actv\_i <- actv  
actv\_i$steps[is.na(actv\_i$steps)] <- mean(actv\_i$steps, na.rm = T)  
sum(is.na(actv\_i$steps))

## [1] 0

Item 7 - Histogram of the total number of steps taken each day after missing values are imputed

steps\_day\_i <- aggregate(steps ~ date, data = actv\_i, sum, na.rm = T)  
hist(steps\_day\_i$steps, breaks = 20, main = "Total number of steps per day - missing data imputed", xlab = "Steps", ylab = "Frequency")



mean(steps\_day\_i$steps)

## [1] 10766.19

median(steps\_day\_i$steps)

## [1] 10766.19

These results for mean and median are not surprising as the median of the uncorrected set is 0, so replacing NAs with zeros unsurprisingly a) reduces the mean and b) has no impact on the median.

Item 8 - Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends Analysis of the differences in activity patterns between weekdays and weekends

actv\_i$date <- as.Date(actv\_i$date)  
actv\_i$dayname <- weekdays(actv\_i$date)  
actv\_i$weekend <- as.factor(ifelse(actv\_i$dayname == "sábado" | actv\_i$dayname == "domingo", "weekend", "weekday"))  
library(lattice)  
plotdata <- aggregate(steps ~ interval + weekend, actv\_i, mean)  
xyplot(steps ~ interval | factor(weekend), layout = c(1, 2), data=plotdata, aspect=1/3, type="l")

