Reproducible Research: Peer Assessment 1

Enzo Alda

Sunday, January 18, 2015

Load libraries

```
library(lattice);
library(psych);
library(sfsmisc);
```

Load data and augment columns

```
# Read the data (already unzipped after download)
data <- read.csv("activity.csv");
# Convert *date* field to proper Date
data$date <- as.Date(data$date);
# Add weekday column
data$weekday <- weekdays(data$date);
# Add the 'minute' column, showing the number of minutes from day start
data$minute <- as.integer(data$interval / 100) * 60 + (data$interval %% 100)
# Add a proper timestamp column
time.string <- paste(as.character(as.integer(data$interval / 100)), as.character(data$interval %% 100),
data$time <- strptime(paste(data$date, time.string), format = "%Y-%m-%d %H:%M");
```

Check data types

```
# Check data types
str(data);
```

```
## 'data.frame':    17568 obs. of  6 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 ...
##  $ weekday : chr  "Monday" "Monday" "Monday" "Monday" ...
##  $ minute  : num  0 5 10 15 20 25 30 35 40 45 ...
##  $ time    : POSIXlt, format: "2012-10-01 00:00:00" "2012-10-01 00:05:00" ...
```

```
# show summary
summary(data);
```

```
##      steps             date               interval        weekday
##  Min.   :  0.00   Min.   :2012-10-01   Min.   :   0.0   Length:17568
##  1st Qu.:  0.00   1st Qu.:2012-10-16   1st Qu.: 588.8   Class :character
##  Median :  0.00   Median :2012-10-31   Median :1177.5   Mode  :character
##  Mean   : 37.38   Mean   :2012-10-31   Mean   :1177.5
##  3rd Qu.: 12.00   3rd Qu.:2012-11-15   3rd Qu.:1766.2
##  Max.   :806.00   Max.   :2012-11-30   Max.   :2355.0
##  NA's   :2304
```

```
##      minute            time
##  Min.   :   0.0   Min.   :2012-10-01 00:00:00
##  1st Qu.: 358.8   1st Qu.:2012-10-16 05:58:45
##  Median : 717.5   Median :2012-10-31 11:57:30
##  Mean   : 717.5   Mean   :2012-10-31 12:23:59
##  3rd Qu.:1076.2   3rd Qu.:2012-11-15 17:56:15
##  Max.   :1435.0   Max.   :2012-11-30 23:55:00
##
```

Compute mean and median for the total number of steps taken per day

```
steps <- tapply(data$steps, data$date, sum);
mn <- mean(steps, na.rm = T)
md <- median(steps, na.rm = T)
```

Show mean and median

```
mn
```

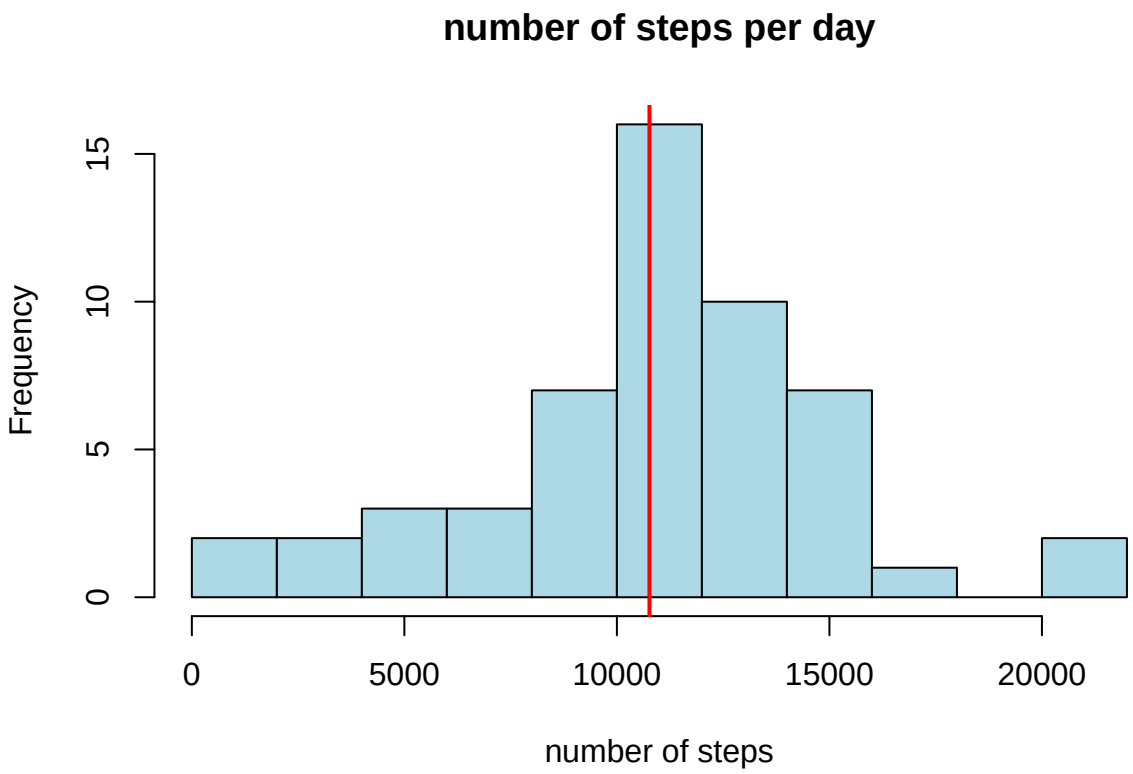
```
## [1] 10766.19
```

```
md
```

```
## [1] 10765
```

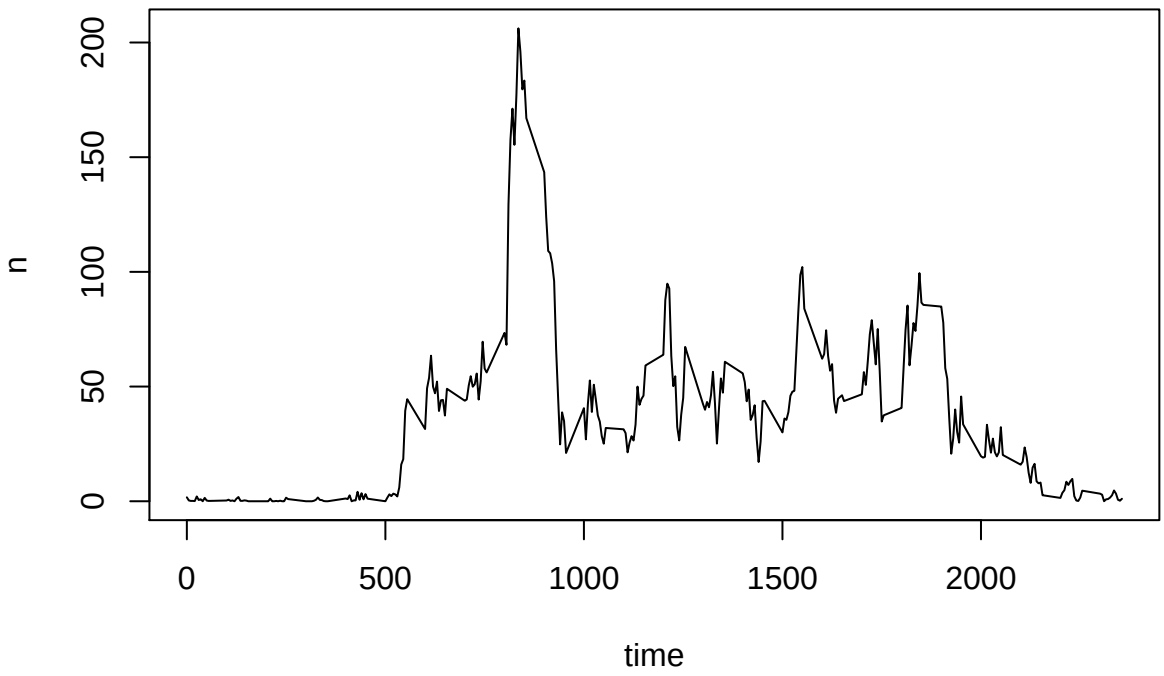
Plot histogram

```
hist(steps, breaks=9, main="number of steps per day", xlab="number of steps", col="lightblue");
abline(v = mn, col="red", lwd=2);
```



Display average daily activity pattern

```
n <- empty.dimnames(tapply(data$steps, data$interval, mean, na.rm = T));
t <- unique(data$interval);
steps <- data.frame(t=t, n=n);
plot(steps, type="l", xlab="time");
```



Locate maximum

```
max.steps <- max(steps$n);
index <- match(max.steps, steps$n);
steps$t[index]
```

```
## [1] 835
```

The interval 835 corresponds to the time interval 8:35-8:40 am

Calculate and report the total number of missing values in the dataset

```
sum(is.na(data$interval))
```

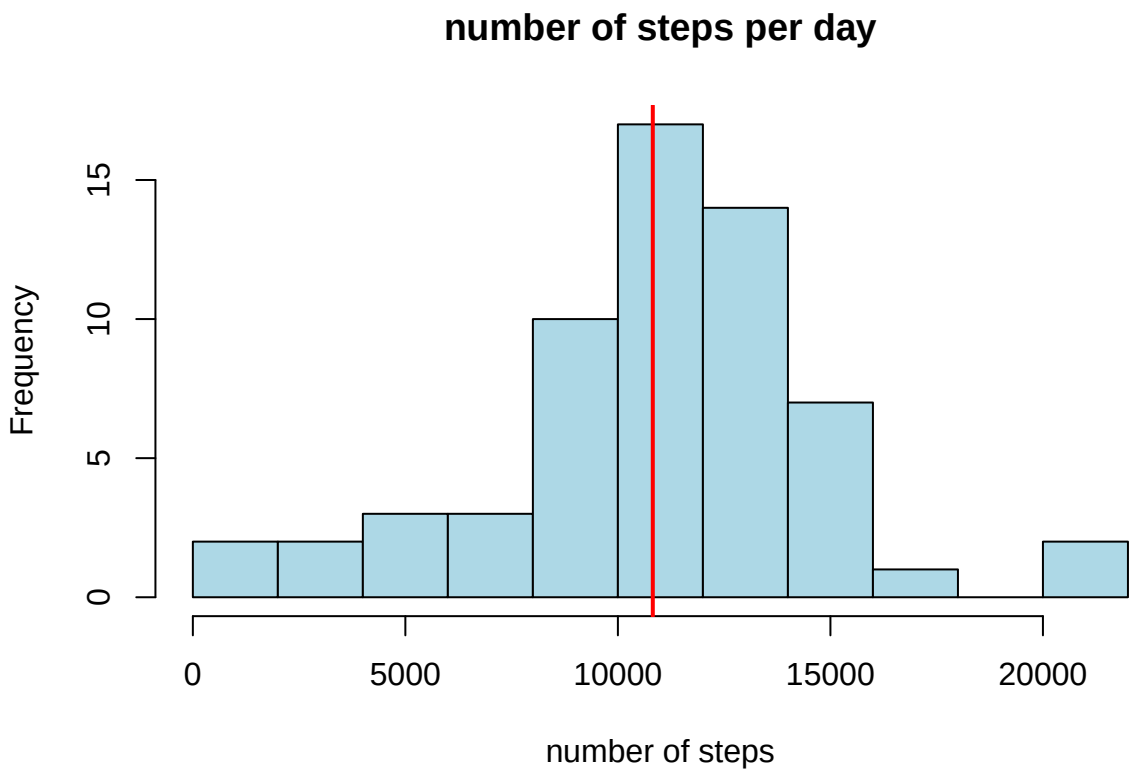
```
## [1] 0
```

We now create a new data set with no missing values. Our imputing strategy is to replace missing values (NA steps) with the mean number of steps (rounded to the nearest integer) for the corresponding interval in the corresponding weekday.

```
# function 'round.mean' returns the rounded value of the mean of a vector
round.mean <- function(v) {
    round(mean(v, na.rm=T));
};
```

```
# Overall estimates
estimates <- aggregate(steps ~ weekday + interval, data=data, FUN=round.mean);

# Function that returns the number of imputed steps for a given weekday/interval
imputed.steps <- function(weekday, interval) {
    estimates[estimates$weekday==weekday &  estimates$interval==interval,]$steps
}

# Prepare new data set
pdata <- data;

# Get number of rows
N <- dim(pdata)[1];

# Impute missing values
for (i in 1:N) {
    if (is.na(pdata$steps[i])) {
        pdata$steps[i] <- imputed.steps(pdata$weekday[i], pdata$interval[i]);
    }
}
```

```
# hist mean median
steps <- tapply(pdata$steps, pdata$date, sum);
mn <- mean(steps, na.rm = T)
md <- median(steps, na.rm = T)
hist(steps, breaks=9, main="number of steps per day", xlab="number of steps", col="lightblue");
abline(v = mn, col="red", lwd=2);
```




```
mn
```

```
## [1] 10821.1
```

```
md
```

```
## [1] 11015
```

```
#
pdata$wday <- NULL;
pdata$wday[  pdata$weekday %in% c("Saturday", "Sunday") ] <- "weekend";
pdata$wday[!(pdata$weekday %in% c("Saturday", "Sunday"))] <- "workday";
pdata$wday <- as.factor(pdata$wday);

pdata.workday <- pdata[pdata$wday == "workday", ];
pdata.weekend <- pdata[pdata$wday == "weekend", ];

par(mfcol=c(2, 1));

n <- empty.dimnames(tapply(pdata.workday$steps, pdata.workday$interval, mean, na.rm = T));
t <- unique(pdata.workday$interval);
steps <- data.frame(t=t, n=n);
plot(
    steps,
    type="l",
```

```
    ylab = "Average Steps",
    xlab = "Time in Hours",
    col="red"
);
legend("topleft", c("Weekdays"), lty=c(1,1), col=c("red"))

n <- empty.dimnames(tapply(pdata.weekend$steps, pdata.weekend$interval, mean, na.rm = T));
t <- unique(pdata.weekend$interval);
steps <- data.frame(t=t, n=n);
plot(
    steps,
    type="l",
    ylab = "Average Steps",
    xlab = "Time in Hours",
    col="green"
);
legend("topleft", c("Weekends"), lty=c(1,1), col=c("green"))
```