Reproducible Research: Project 1

Loading and preprocessing the data

Loading

```
dat = read.csv("activity.csv", header = TRUE)
nasidx = which(!complete.cases(dat[,1 ]))
nasdays = dat[nasidx,2]
summary(nasdays)
```

```
## 2012-10-01 2012-10-02 2012-10-03 2012-10-04 2012-10-05 2012-10-06
##        288          0          0          0          0          0
## 2012-10-07 2012-10-08 2012-10-09 2012-10-10 2012-10-11 2012-10-12
##          0        288          0          0          0          0
## 2012-10-13 2012-10-14 2012-10-15 2012-10-16 2012-10-17 2012-10-18
##          0          0          0          0          0          0
## 2012-10-19 2012-10-20 2012-10-21 2012-10-22 2012-10-23 2012-10-24
##          0          0          0          0          0          0
## 2012-10-25 2012-10-26 2012-10-27 2012-10-28 2012-10-29 2012-10-30
##          0          0          0          0          0          0
## 2012-10-31 2012-11-01 2012-11-02 2012-11-03 2012-11-04 2012-11-05
##          0        288          0          0        288          0
## 2012-11-06 2012-11-07 2012-11-08 2012-11-09 2012-11-10 2012-11-11
##          0          0          0        288        288          0
## 2012-11-12 2012-11-13 2012-11-14 2012-11-15 2012-11-16 2012-11-17
##          0          0        288          0          0          0
## 2012-11-18 2012-11-19 2012-11-20 2012-11-21 2012-11-22 2012-11-23
##          0          0          0          0          0          0
## 2012-11-24 2012-11-25 2012-11-26 2012-11-27 2012-11-28 2012-11-29
##          0          0          0          0          0          0
## 2012-11-30
##        288
```

Processing

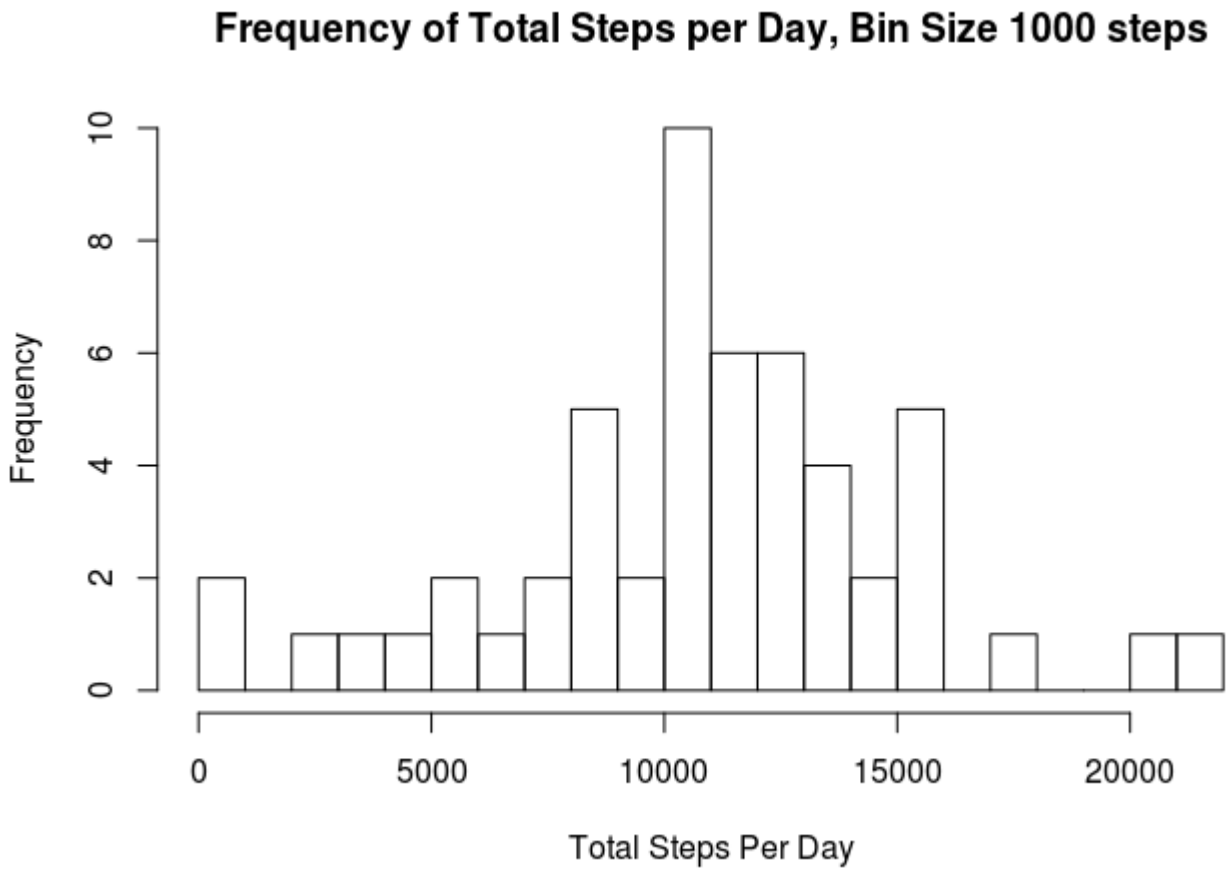
From the summary of the NA values, it is clear that NAs are simply days during which no data were collected.
Adding a columun with the weekday names

```
dat[, 4] = weekdays(as.Date(dat[, 2]))
 colnames(dat)[4] <- "weekdays"
```

Mean total number of steps taken per day

Steps Per Day

```
library(dplyr)
TotStepsEachDay = group_by(dat, date) %>% summarize(steps_per_day = sum(steps))
hist(TotStepsEachDay$steps_per_day, 25, main = "Frequency of Total Steps per Day, Bin Si
ze 1000 steps", xlab = "Total Steps Per Day")
```



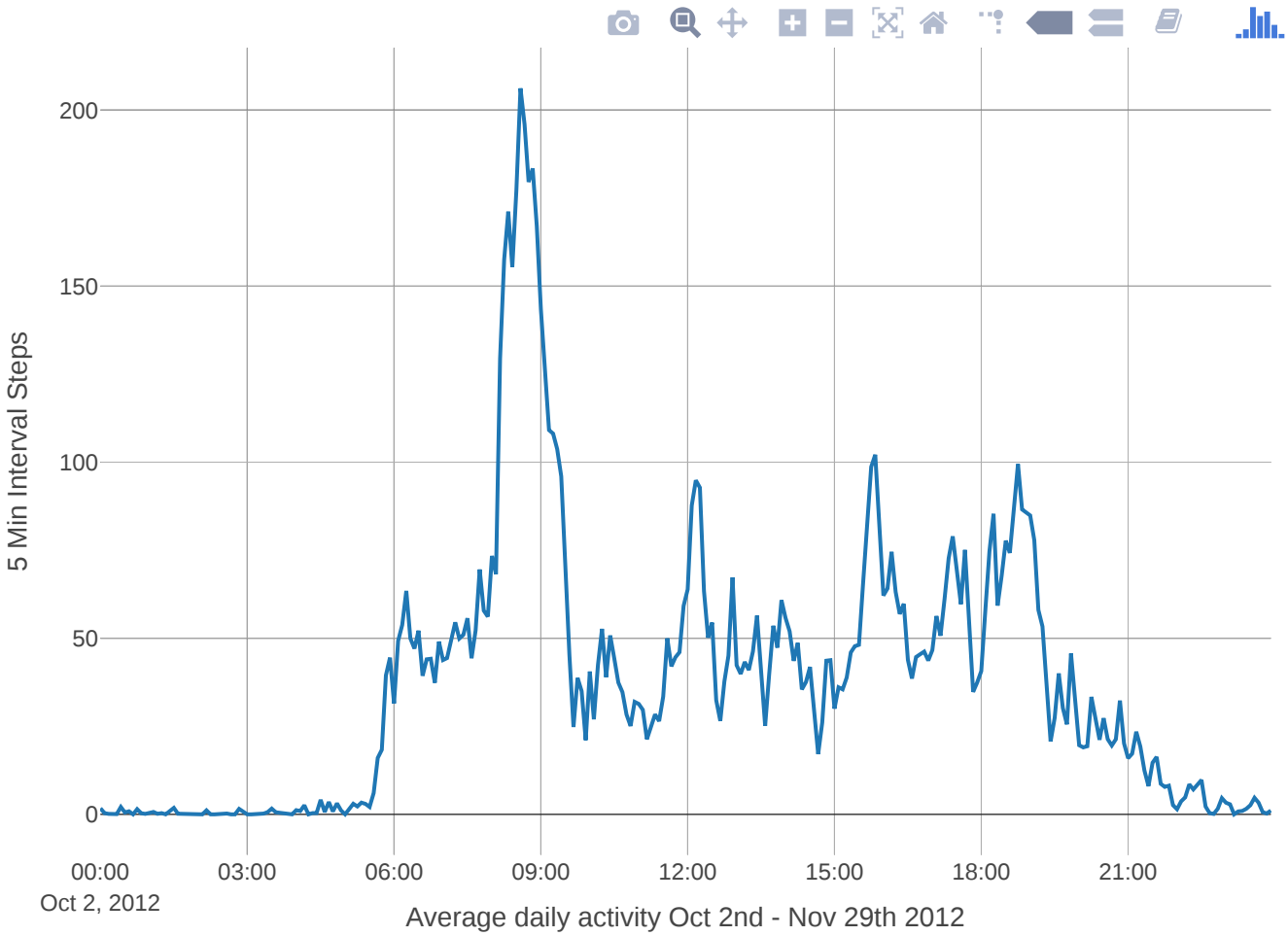
Mean and Median of Total Steps Per Day

```
summary(TotStepsEachDay)
```

```
##          date    steps_per_day
##  2012-10-01: 1   Min.   :   41
##  2012-10-02: 1   1st Qu.: 8841
##  2012-10-03: 1   Median :10765
##  2012-10-04: 1   Mean   :10766
##  2012-10-05: 1   3rd Qu.:13294
##  2012-10-06: 1   Max.   :21194
##  (Other)   :55   NA's   :8
```

What is the average daily activity pattern?

```
library(plotly)
clean_dat = dat[which(complete.cases(dat[,1 ])), ]
missing_dat = dat[which(!complete.cases(dat[,1 ])), ]
tm <- seq(0, 299*288, by = 300)
now <- as.Date(clean_dat[1, 2])
now <- as.POSIXct(now)
now <- now-(9*60*60)
x <- now+tm
five_min_avg = group_by(clean_dat, interval) %>% summarize(five_min_mean = mean(steps))
y <- unlist(five_min_avg[ ,2], use.names = FALSE)
p <- plot_ly(x = ~x, y = ~y, mode = 'lines')
ax <- list(title = "Average daily activity Oct 2nd - Nov 29th 2012")
ay <- list(title = "5 Min Interval Steps")
layout(p, xaxis= ax, yaxis = ay)
```



Time interval with largest average numbe of step

```
five_min_avg[which(y==max(y)), 1]
```

```
## # A tibble: 1 x 1
##   interval
##      <int>
## 1      835
```

Imputing missing values

Total number of raws with a missing value

```
length(which(!complete.cases(dat[,1 ])))
```

```
## [1] 2304
```

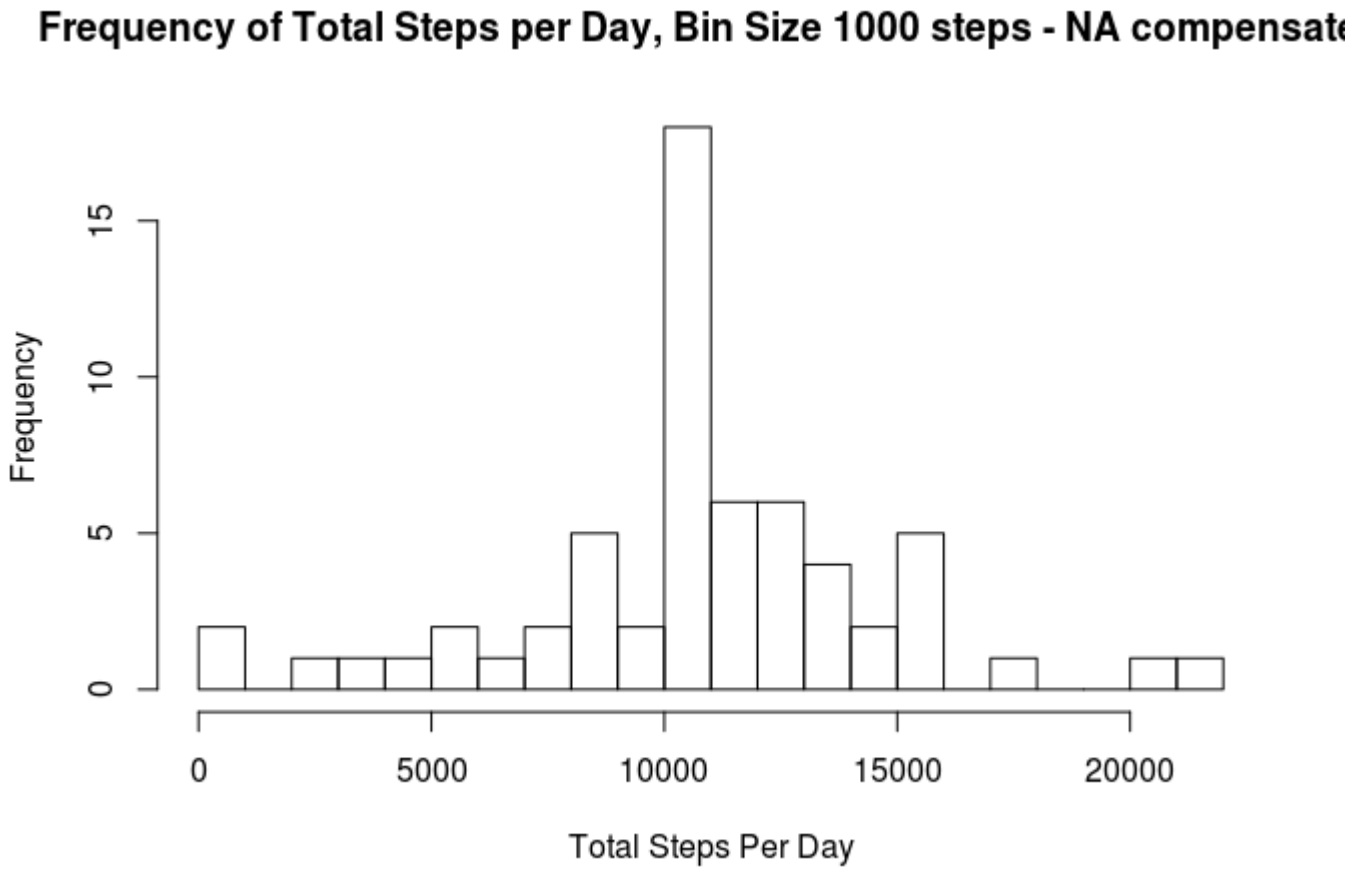
From the pre-processing phase of the analsys, we know the missing values correspond to whole days, when no data were collected. A simple strategy to imputing the missing data is simply replacing the missing days with the average activty day. However, it might be useful to verify if the missing days display some other property that could be used to more accurately imput the missing data.
An easy way to do this is by checking what days of the week are missing. If for instance only Mondays were missing, and Mondays were to have much less steps than other days of the week, it would be more accurate to replace the missing data, with the average of the avaibale data collected on Mondays, rather than any other day.
Days where data are missing:

```
 levels(factor(clean_dat$weekdays))
```

```
## [1] "Friday"    "Monday"    "Saturday"  "Sunday"    "Thursday"  "Tuesday"
## [7] "Wednesday"
```

Every single day of the week is inlcuded in the 8 days of missing data, indicating that there is no particular pattern linked with the missing data days, therefore we will simply replace them with the average day steps distribution shown above.

```
for (i in seq(from=0, to=288*8-1, by=1)) {
     missing_dat[1+i, 1] = five_min_avg[1+i%%288, 2]
     }
full_dat = bind_rows(missing_dat, clean_dat)
TotStepsEachDay = group_by(full_dat, date) %>% summarize(steps_per_day = sum(steps))
hist(TotStepsEachDay$steps_per_day, 25, main = "Frequency of Total Steps per Day, Bin Si
ze 1000 steps - NA compensated", xlab = "Total Steps Per Day")
```



Are there differences in activity patterns between weekdays and weekends?

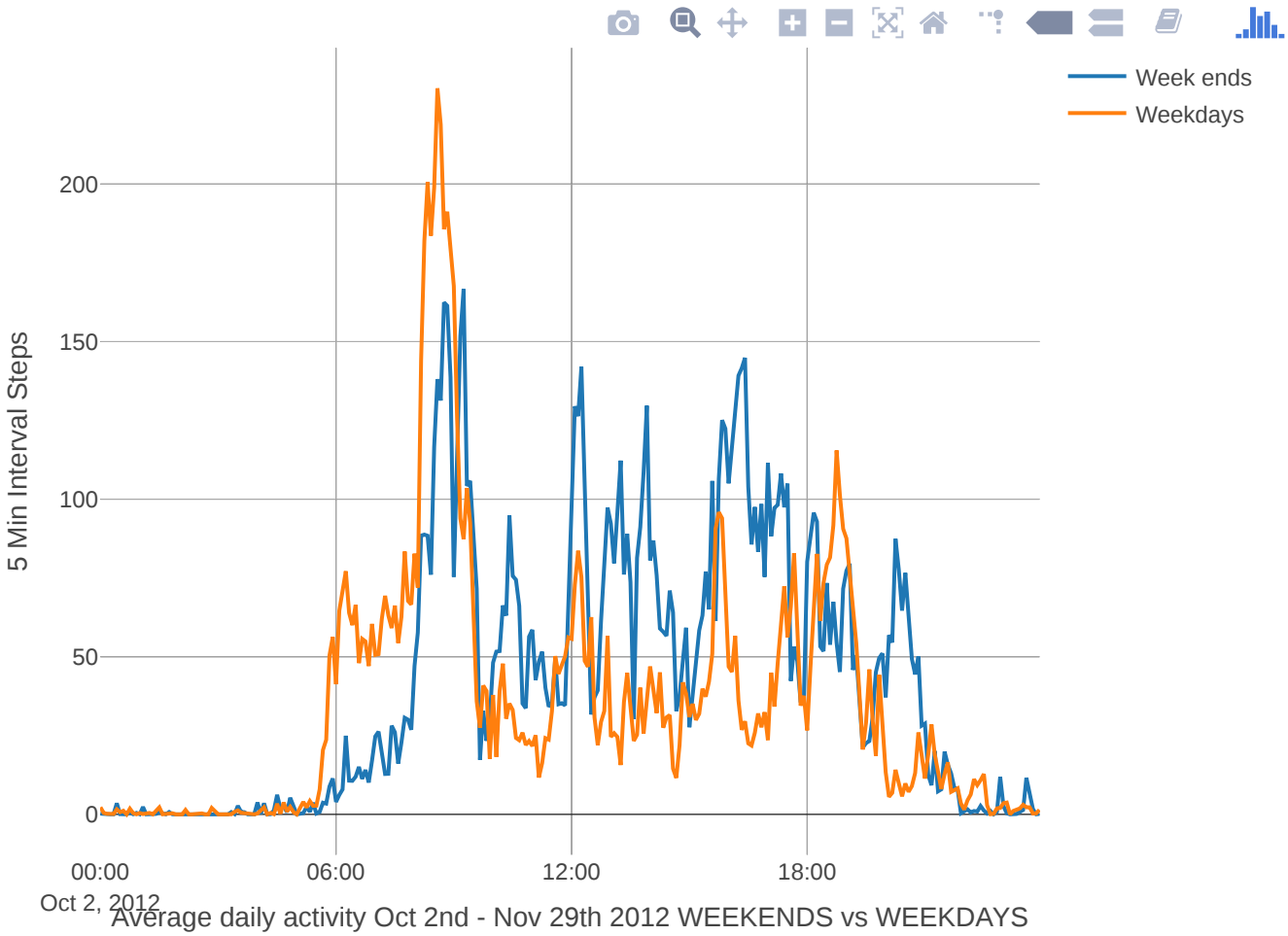
```
# Separate Weekdays from Weekend days
weekends = full_dat[full_dat$weekdays == "Sunday" | full_dat$weekdays == "Saturday" , ]
week = full_dat[full_dat$weekdays != "Sunday" & full_dat$weekdays != "Saturday" , ]

# weekdays and weekend means plots

five_min_avg_wkend = group_by(weekends, interval) %>% summarize(five_min_mean = mean(ste
ps))
y1 <- unlist(five_min_avg_wkend[ ,2], use.names = FALSE)
five_min_avg_wk = group_by(week, interval) %>% summarize(five_min_mean = mean(steps))
y2 <- unlist(five_min_avg_wk[ ,2], use.names = FALSE)

ax1 <- list(title = "Average daily activity Oct 2nd - Nov 29th 2012 WEEKENDS vs WEEKDAY
S")
ay1 <- list(title = "5 Min Interval Steps")
data <- data.frame(x, y1, y2)

p <- plot_ly(data, x = ~x, y = ~y1, name = 'Week ends', type = 'scatter', mode = 'lines'
) %>%
  add_trace(y = ~y2, name = 'Weekdays', mode = 'lines')
layout(p, xaxis= ax1, yaxis = ay1)
```


Week ends
Weekdays
200
150
100
50
0
5 Min Interval Steps
00:00
06:00
12:00
18:00
Oct 2, 2012
Average daily activity Oct 2nd - Nov 29th 2012 WEEKENDS vs WEEKDAYS