Reproducible Research: Course Project 1

Alejandro Osorio

August 2018

Reading and Visualizing Data

Reading

```
dataoriginal <- read_csv("activity.csv")
```

```
## Parsed with column specification:
## cols(
##   steps = col_integer(),
##   date = col_date(format = ""),
##   interval = col_integer()
## )
```

Visualizing

General structure:

```
head(dataoriginal)
```

```
## # A tibble: 6 x 3
##   steps date       interval
##   <int> <date>        <int>
## 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
```

So three variables (steps, date and interval), in correct formats, and indeed with some NAs to be taken care of (in Second Part: Handling NAs).

Dimensioning data structure:

```
dim(dataoriginal)
```

```
## [1] 17568     3
```

First Part: Ignoring NAs

Creating dataset without rows that contain NAs

Rows to ignore:

```
rowignore <- which(rowSums(is.na(dataoriginal)) != 0)
length(rowignore)
```

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

Dataset without rows that contain NAs:

```
datanonas <- dataoriginal[-rowignore,]
head(datanonas)
```

```
## # A tibble: 6 x 3
##   steps date       interval
##   <int> <date>        <int>
## 1     0 2012-10-02        0
## 2     0 2012-10-02        5
## 3     0 2012-10-02       10
## 4     0 2012-10-02       15
## 5     0 2012-10-02       20
## 6     0 2012-10-02       25
```

Total number of steps taken per day

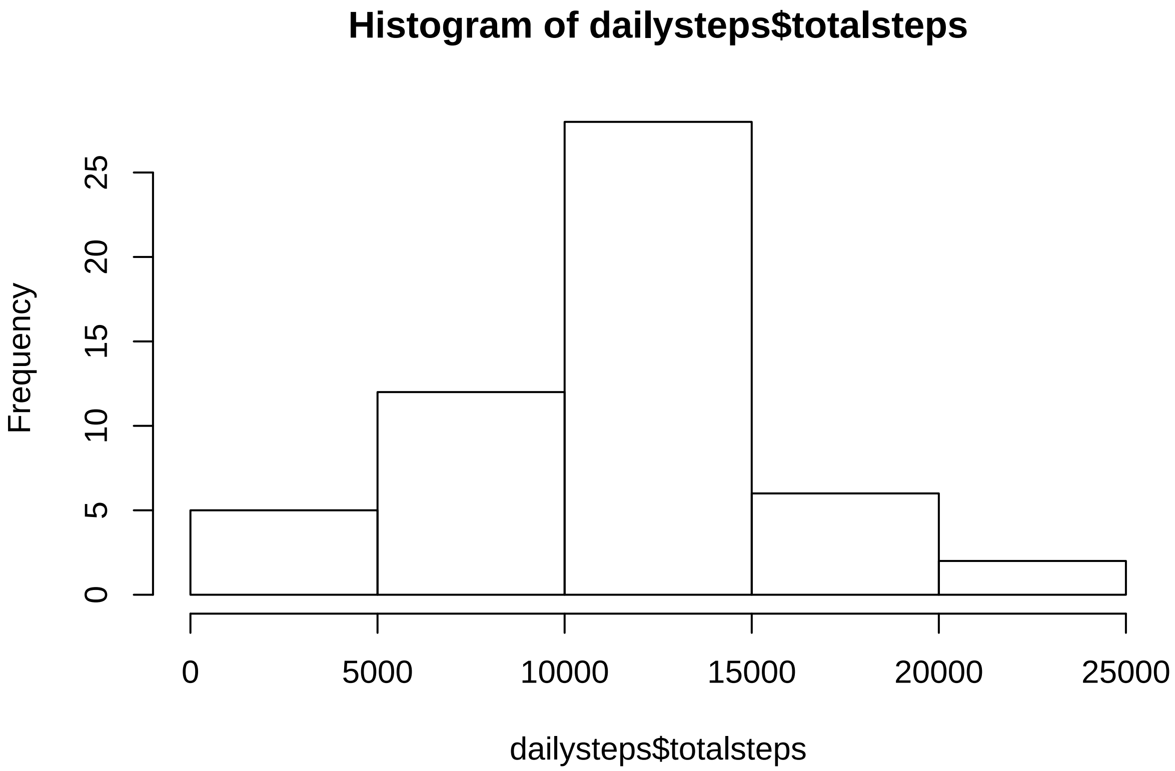
Dataset:

```
dailysteps <- datanonas %>% group_by(date) %>% summarise(totalsteps = sum(steps))
summary(dailysteps)
```

```
##       date              totalsteps
##  Min.   :2012-10-02   Min.   :   41
##  1st Qu.:2012-10-16   1st Qu.: 8841
##  Median :2012-10-29   Median :10765
##  Mean   :2012-10-30   Mean   :10766
##  3rd Qu.:2012-11-16   3rd Qu.:13294
##  Max.   :2012-11-29   Max.   :21194
```

Histogram:

```
hist(dailysteps$totalsteps)
```



Mean and Median number of steps taken per day

Dataset:

```
dailystats <- datanonas %>% group_by(date) %>% summarise(meansteps = mean(steps), mediansteps = median(
summary(dailystats)
```

```
##       date              meansteps        mediansteps
##  Min.   :2012-10-02   Min.   : 0.1424   Min.   :0
##  1st Qu.:2012-10-16   1st Qu.:30.6979   1st Qu.:0
##  Median :2012-10-29   Median :37.3785   Median :0
##  Mean   :2012-10-30   Mean   :37.3826   Mean   :0
##  3rd Qu.:2012-11-16   3rd Qu.:46.1597   3rd Qu.:0
##  Max.   :2012-11-29   Max.   :73.5903   Max.   :0
```

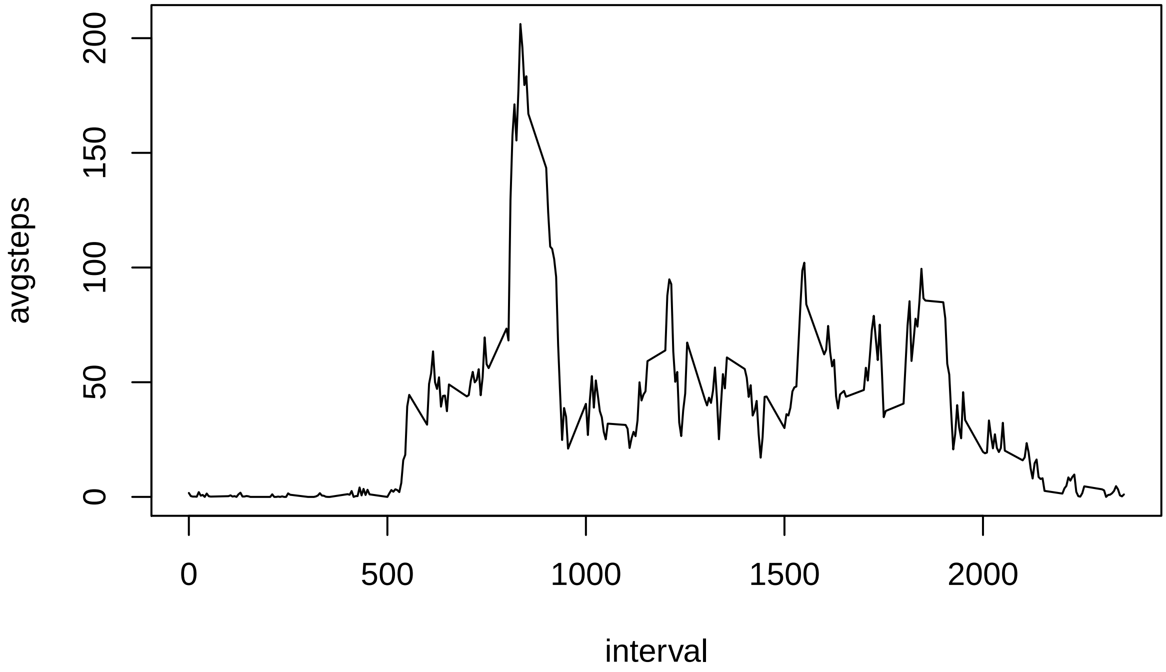
Average daily activity pattern

Dataset:

```
dailyactivity <- datanonas %>% group_by(interval) %>% summarise(avgsteps = mean(steps))
summary(dailyactivity)
```

```
##     interval          avgsteps
##  Min.   :   0.0   Min.   :  0.000
##  1st Qu.: 588.8   1st Qu.:  2.486
##  Median :1177.5   Median : 34.113
##  Mean   :1177.5   Mean   : 37.383
##  3rd Qu.:1766.2   3rd Qu.: 52.835
##  Max.   :2355.0   Max.   :206.170
```

```
with(dailyactivity, plot(x = interval, y = avgsteps, type = "l", xlab = "interval"))
```



Interval with maximum avg number of steps:

```
dailyactivity %>% filter(avgsteps == max(avgsteps))
```

```
## # A tibble: 1 x 2
##   interval avgsteps
##      <int>    <dbl>
## 1      835     206.
```

Second Part: Handling NAs

Dimensioning NAs

Total number of rows with NAs:

```
colSums((is.na(dataoriginal)))
```

```
##    steps     date interval
##     2304        0        0
```

So all NAs are concentrated within the first variable, with a total number of rows equal to the total number of NAs.

Figuring how NAs are distributed, by date:

```
table(is.na(dataoriginal$steps), dataoriginal$date)
```

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

So dates with NAs were all 100% NAs.

Strategy for handling NAs

Given that dates with NAs were all 100% NAs, the days ignored in previous analysis, were:

```
missingdays <- unique(dataoriginal[rowignore,]$date)
missingdays
```

```
## [1] "2012-10-01" "2012-10-08" "2012-11-01" "2012-11-04" "2012-11-09"
## [6] "2012-11-10" "2012-11-14" "2012-11-30"
```

The strategy consists on replacing NAs with their total mean value, for each interval (values already available, from "Average daily activity pattern").

Creating dataset that handles NAs

First, replacing NAs for mean values obtained previously:

```
datanas <- data.frame("steps" = integer(0), "date" = integer(0), "interval" = integer(0))
class(datanas$date) <- "Date"
for (day in 1:length(missingdays)) {
        datanas <- bind_rows(datanas, data.frame("steps" = dailyactivity$avgsteps, "date" = missingdays
}
```

Checking number of rows per date:

```
table(datanas$date)
```

```
##
## 2012-10-01 2012-10-08 2012-11-01 2012-11-04 2012-11-09 2012-11-10
##        288        288        288        288        288        288
## 2012-11-14 2012-11-30
##        288        288
```

Checking structure:

```
str(datanas)
```

```
## 'data.frame':    2304 obs. of  3 variables:
##  $ steps   : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

So each date that had 100% NAs for the 'steps' variable, had them replaced for the total means. Finally, binding them with the data that had NAs filtered, and arranging the result by date:

```
datanew <- bind_rows(datanonas, datanas)
datanew <- arrange(datanew, date)
```

Checking final structure:

```
str(datanew)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : num  1.717 0.3396 0.1321 0.1509 0.0755 ...
##  $ date    : Date, format: "2012-10-01" "2012-10-01" ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

Which has same dimensions and structure than the original dataset.

Total number of steps taken per day

Dataset:

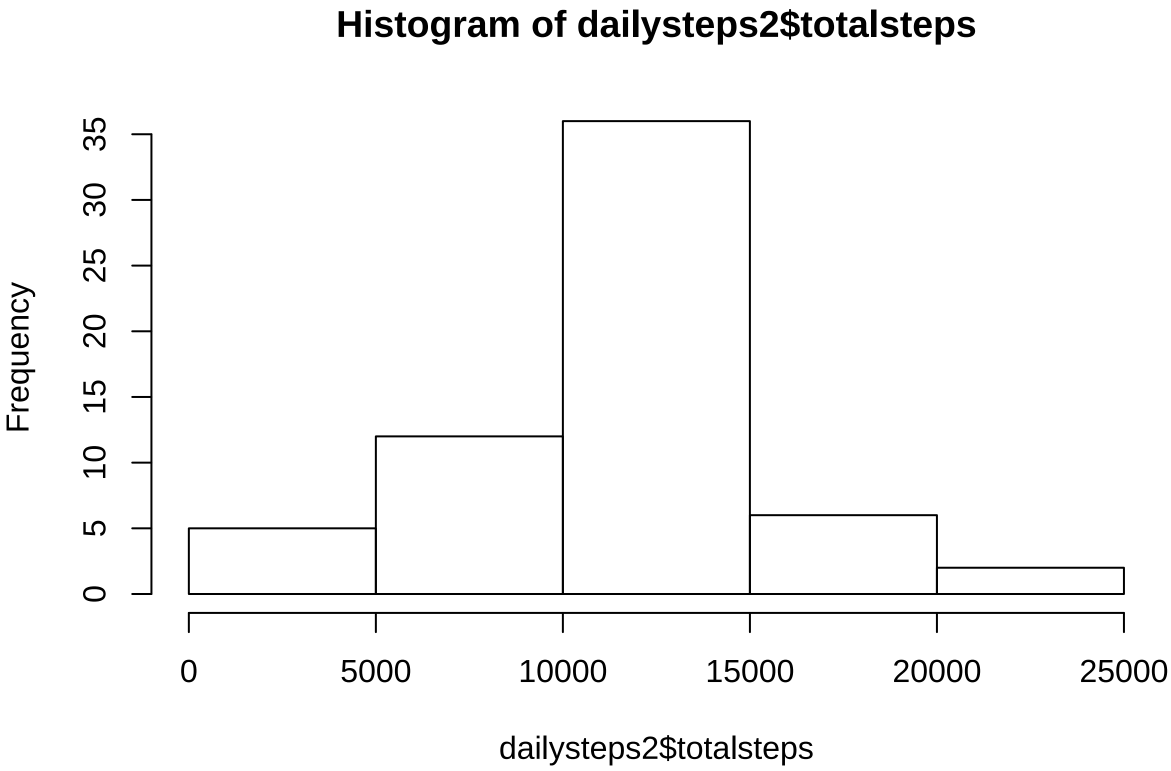
```
dailysteps2 <- datanew %>% group_by(date) %>% summarise(totalsteps = sum(steps))
summary(dailysteps2)
```

```
##       date              totalsteps   
##  Min.   :2012-10-01   Min.   :   41  
##  1st Qu.:2012-10-16   1st Qu.: 9819  
##  Median :2012-10-31   Median :10766  
##  Mean   :2012-10-31   Mean   :10766  
##  3rd Qu.:2012-11-15   3rd Qu.:12811  
##  Max.   :2012-11-30   Max.   :21194  
```

Which shows differences with the dataset without NAs, regarding 1st and 3rd quarters of the total number of steps.

Histogram:

```
hist(dailysteps2$totalsteps)
```



Which looks pretty close to the one without NAs.

Mean and Median number of steps taken per day

Dataset:

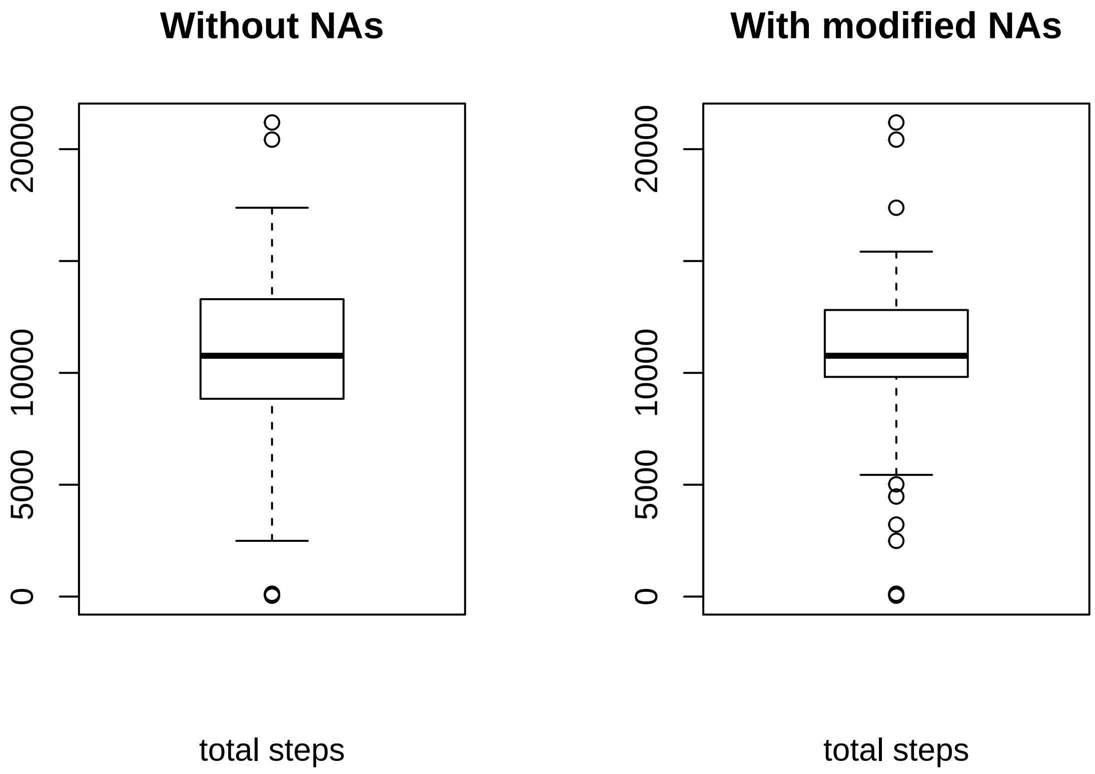
```
dailystats2 <- datanew %>% group_by(date) %>% summarise(meansteps = mean(steps), mediansteps = median(s
summary(dailystats2)
```

```
##       date              meansteps         mediansteps
##  Min.   :2012-10-01   Min.   : 0.1424   Min.   : 0.000
##  1st Qu.:2012-10-16   1st Qu.:34.0938   1st Qu.: 0.000
##  Median :2012-10-31   Median :37.3826   Median : 0.000
##  Mean   :2012-10-31   Mean   :37.3826   Mean   : 4.474
##  3rd Qu.:2012-11-15   3rd Qu.:44.4826   3rd Qu.: 0.000
##  Max.   :2012-11-30   Max.   :73.5903   Max.   :34.113
```

Comparison between results without NAs and with replaced NAs

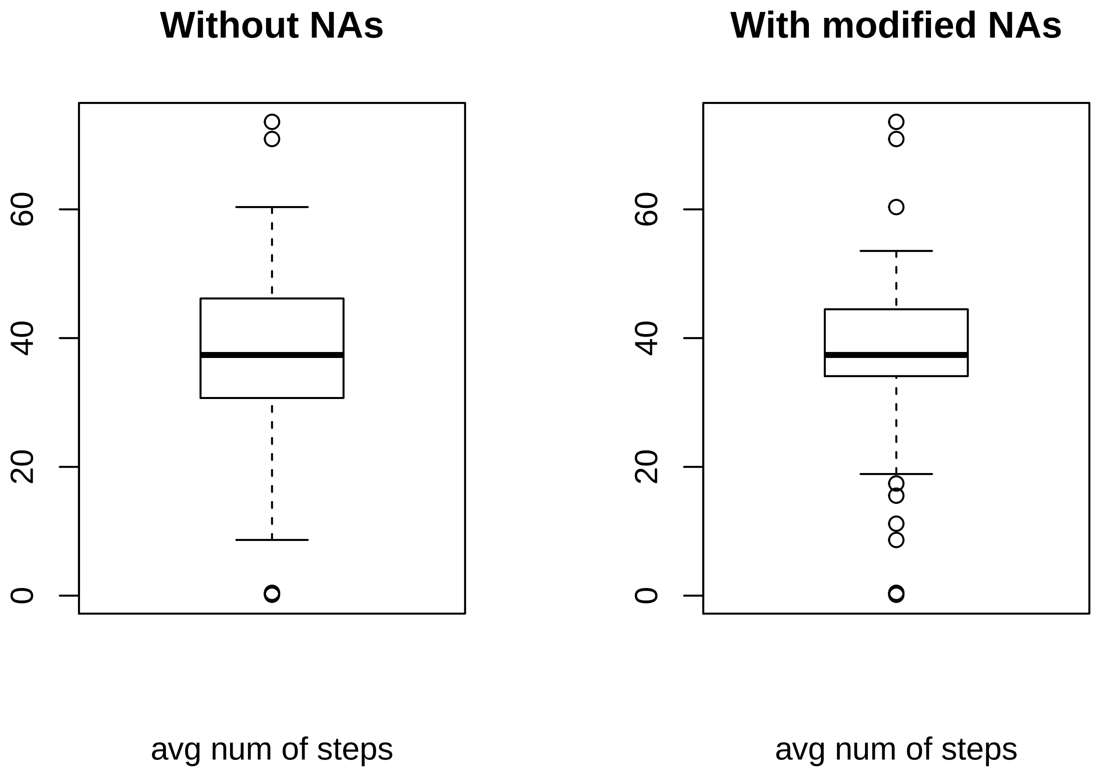
Comparison between total number of steps:

```
par(mfrow = c(1,2))
boxplot(dailysteps$totalsteps, xlab = "total steps", main = "Without NAs")
boxplot(dailysteps2$totalsteps, xlab = "total steps", main = "With modified NAs")
```



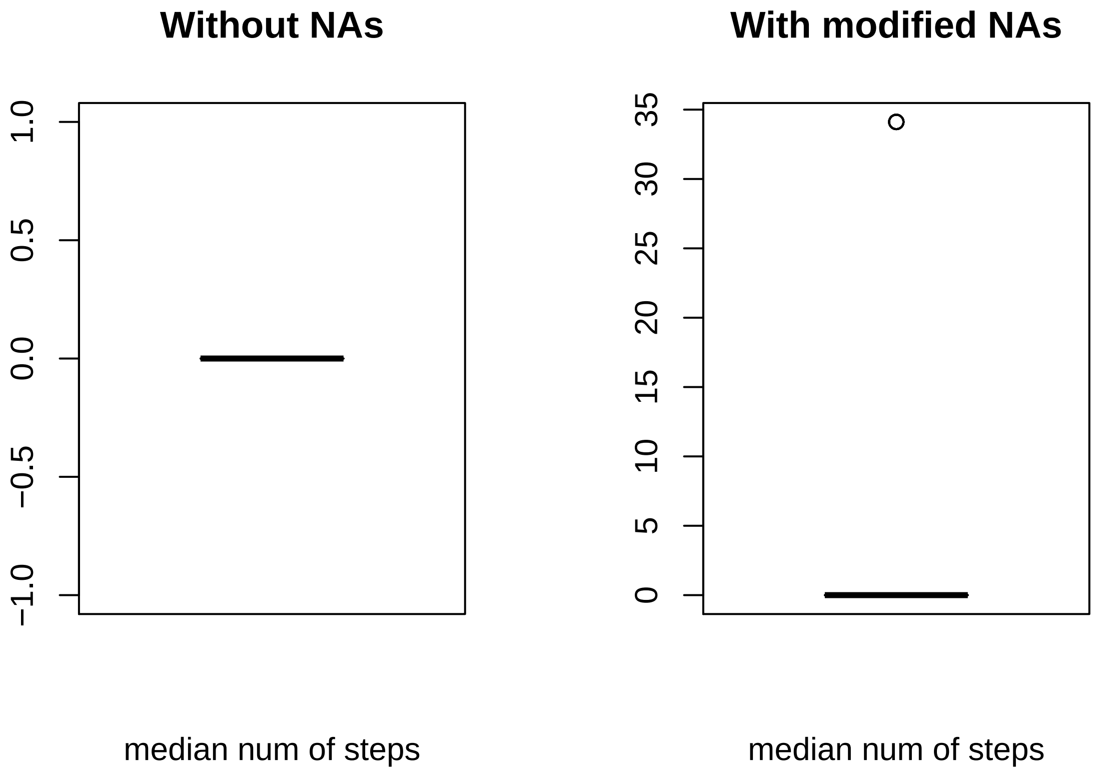
Comparison between average number of steps:

```
par(mfrow = c(1,2))
boxplot(dailystats$meansteps, xlab = "avg num of steps", main = "Without NAs")
boxplot(dailystats2$meansteps, xlab = "avg num of steps", main = "With modified NAs")
```



Comparison between median number of steps:

```r
par(mfrow = c(1,2))
boxplot(dailystats$mediansteps, xlab = "median num of steps", main = "Without NAs")
boxplot(dailystats2$mediansteps, xlab = "median num of steps", main = "With modified NAs")
```



Conclusions:

As it can be seen from the plots, the values obtained after replacing NAs, differ from the estimates from the first part of the assignment. Specifically, the impact of imputing missing data on the estimates of the total daily number of steps, shrunk the variability of the results.

Third Part: Activity patterns between weekdays and weekends

Creating dataset with factors

Adding each date's corresponding days, to the dataset with filled-in NAs :

```
datanew <- datanew %>% mutate(day =  factor(weekdays(date)))
head(datanew)
```

```
## # A tibble: 6 x 4
##    steps date       interval day
##    <dbl> <date>        <int> <fct>
## 1 1.72   2012-10-01        0 lunes
## 2 0.340  2012-10-01        5 lunes
## 3 0.132  2012-10-01       10 lunes
## 4 0.151  2012-10-01       15 lunes
## 5 0.0755 2012-10-01       20 lunes
## 6 2.09   2012-10-01       25 lunes
```

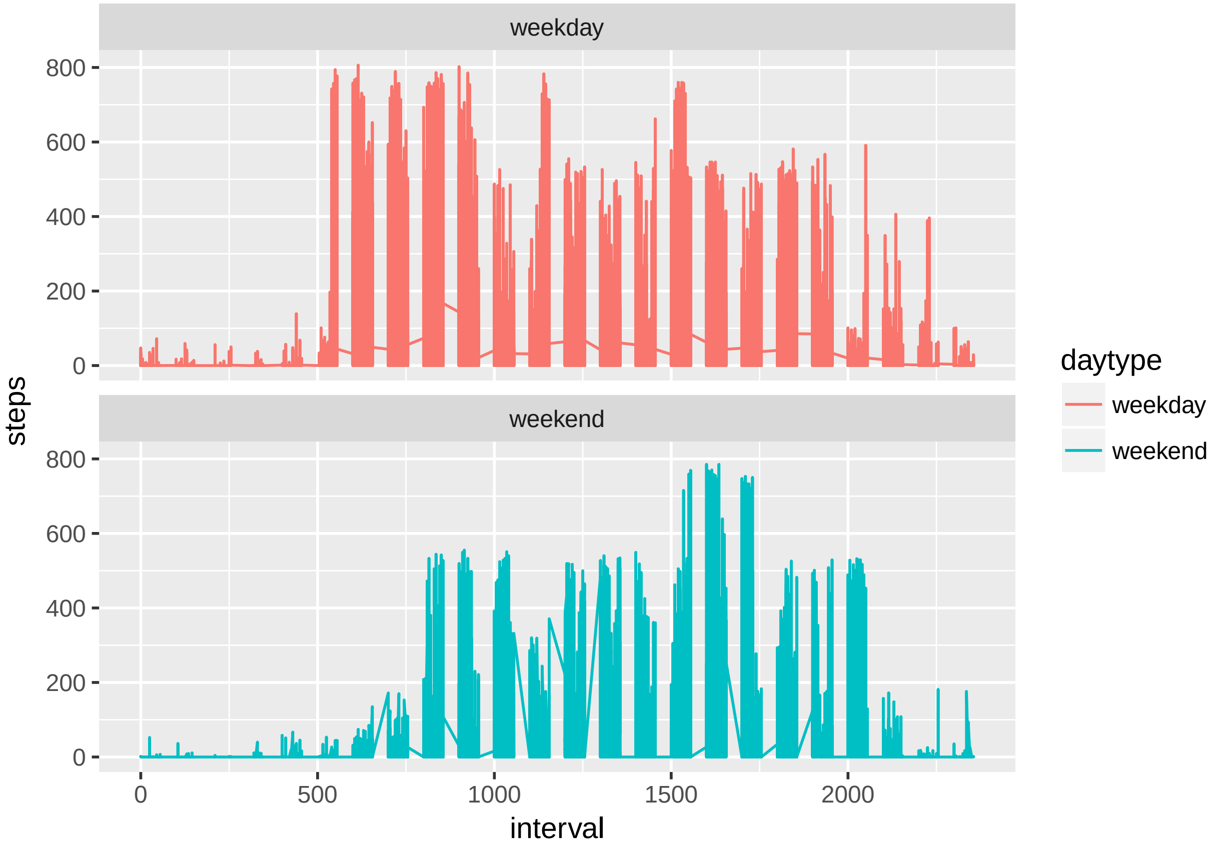
Adding factor variable (weekday / weekend) to previous dataset:

```
datanew <- datanew %>% mutate(daytype = factor(ifelse(day %in% c("sábado", "domingo"), "weekend", "weekd
head(datanew)
```

```
## # A tibble: 6 x 5
##    steps date       interval day   daytype
##    <dbl> <date>        <int> <fct> <fct>
## 1 1.72   2012-10-01        0 lunes weekday
## 2 0.340  2012-10-01        5 lunes weekday
## 3 0.132  2012-10-01       10 lunes weekday
## 4 0.151  2012-10-01       15 lunes weekday
## 5 0.0755 2012-10-01       20 lunes weekday
## 6 2.09   2012-10-01       25 lunes weekday
```

Plotting average number of steps taken, across weekdays or weekends

```
ggplot(data = datanew) +
        geom_line(mapping = aes(x = interval, y = steps, color = daytype)) +
        facet_wrap(~ daytype, nrow = 2)
```


weekday
weekend
steps
interval
daytype
weekday
weekend