Reproducible Research - Week 2 - Course Project

Sourav Datta

6/19/2020

1. Getting and cleaning data

Download the data from the given URL and verify it is in proper format to continue analysis.

```
if (!file.exists('./data')) {
  dir.create('data')
}

download.file('https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip',
              method = 'curl',
              mode = 'wb',
              destfile = 'data/activitydata.zip')

unzip('data/activitydata.zip', overwrite = TRUE, exdir = 'data')

stepdata <- read.csv('data/activity.csv', header = TRUE)
str(stepdata)
```

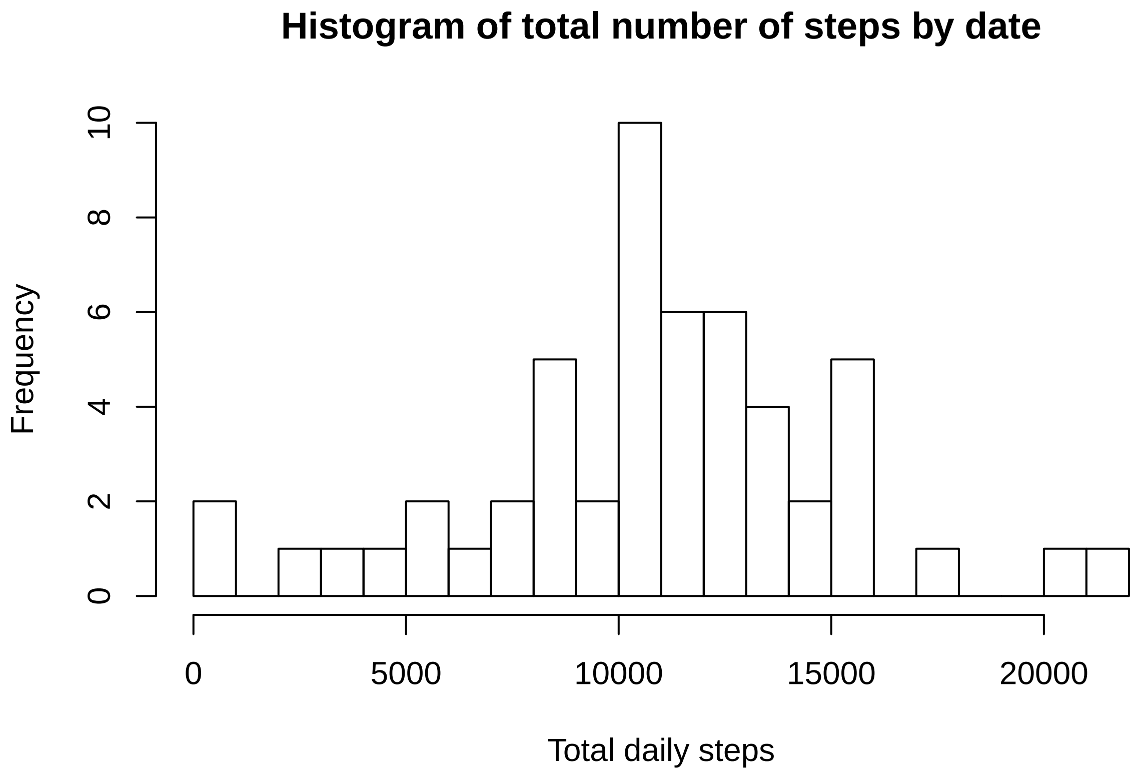
```
## 'data.frame':    17568 obs. of  3 variables:
##  $ steps   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int  0 5 10 15 20 25 30 35 40 45 ...
```

2. Total number of steps taken each day

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
stepsbydate <- stepdata %>%
  select(steps, date) %>%
  na.omit() %>%
  group_by(date) %>%
  summarise(nsteps = sum(steps))

hist(stepsbydate$nsteps, xlab = 'Total daily steps',
     main = 'Histogram of total number of steps by date',
     breaks = 20)
```



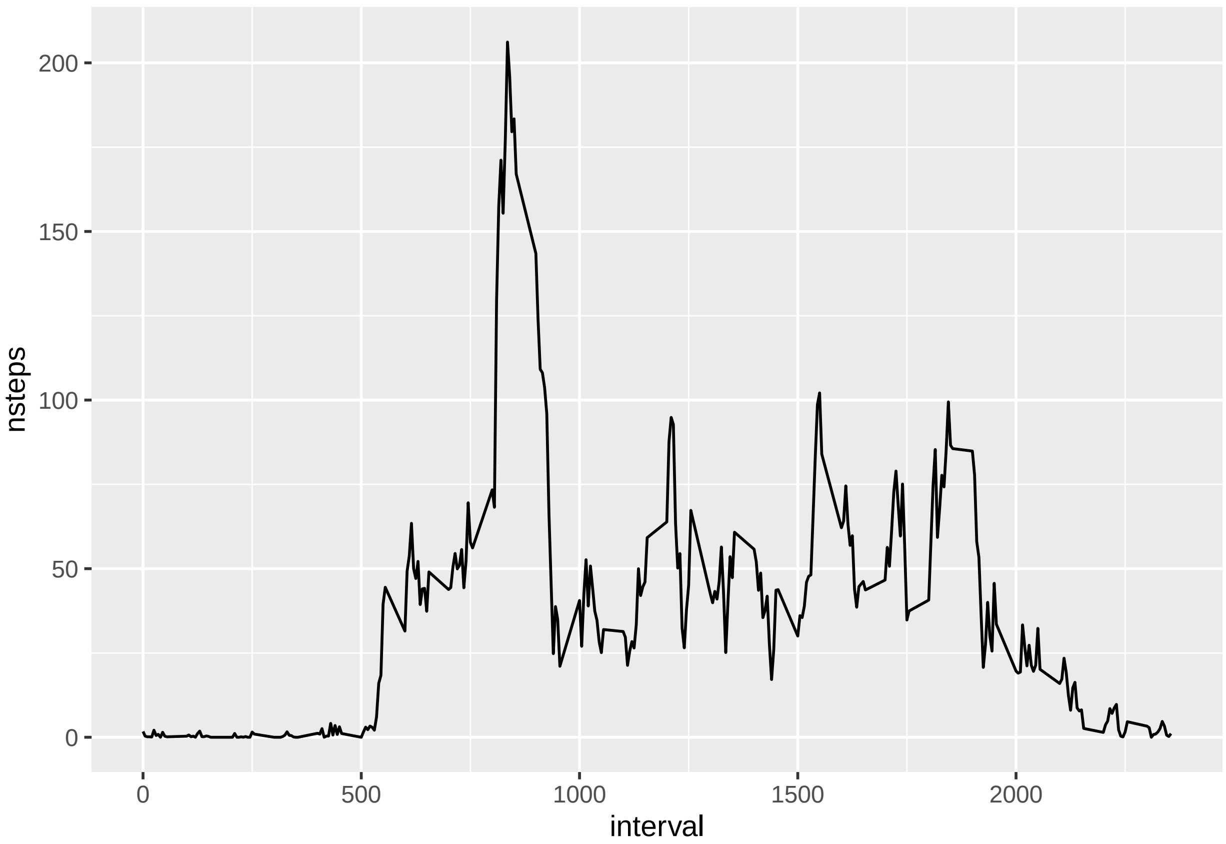
3. Mean and median of the total number steps taken per day

```
mean1 <- mean(stepsbydate$nsteps)
```

```
median1 <- median(stepsbydate$nsteps)
```

4. Time series plot of the average number of steps taken

```
library(ggplot2)

stepsbyinterval <- stepdata %>%
  select(steps, interval) %>%
  na.omit() %>%
  group_by(interval) %>%
  summarise(nsteps = mean(steps))

ggplot(stepsbyinterval, aes(x = interval, y = nsteps)) + geom_line()
```



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

```
stepsbyinterval[which(stepsbyinterval$nsteps == max(stepsbyinterval$nsteps)), ]
```

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

6. Imputing missing data

Missing values (i.e. `NA`) can be replaced by the mean of that interval.

```
replace.missing <- function (x) {
  replace(x, is.na(x), mean(x, na.rm = TRUE))
}
```

We can now replace the missing values for each group of interval by the above function.

```
replaced.stepdata <- stepdata %>%
  group_by(interval) %>%
  mutate(steps = replace.missing(steps)) %>%
  ungroup()

str(replaced.stepdata)
```

```
## tibble [17,568 x 3] (S3: tbl_df/tbl/data.frame)
##  $ steps   : num [1:17568] 1.717 0.3396 0.1321 0.1509 0.0755 ...
##  $ date    : Factor w/ 61 levels "2012-10-01","2012-10-02",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ interval: int [1:17568] 0 5 10 15 20 25 30 35 40 45 ...
```

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

Now we recalculate the number of steps each day with replaced data.

```
stepsbydate2 <- replaced.stepdata %>%
  select(steps, date) %>%
  group_by(date) %>%
  summarise(nsteps = sum(steps))

head(stepsbydate2)
```

```
## # A tibble: 6 x 2
##   date       nsteps
##   <fct>       <dbl>
## 1 2012-10-01 10766.
## 2 2012-10-02   126
## 3 2012-10-03 11352
## 4 2012-10-04 12116
## 5 2012-10-05 13294
## 6 2012-10-06 15420
```

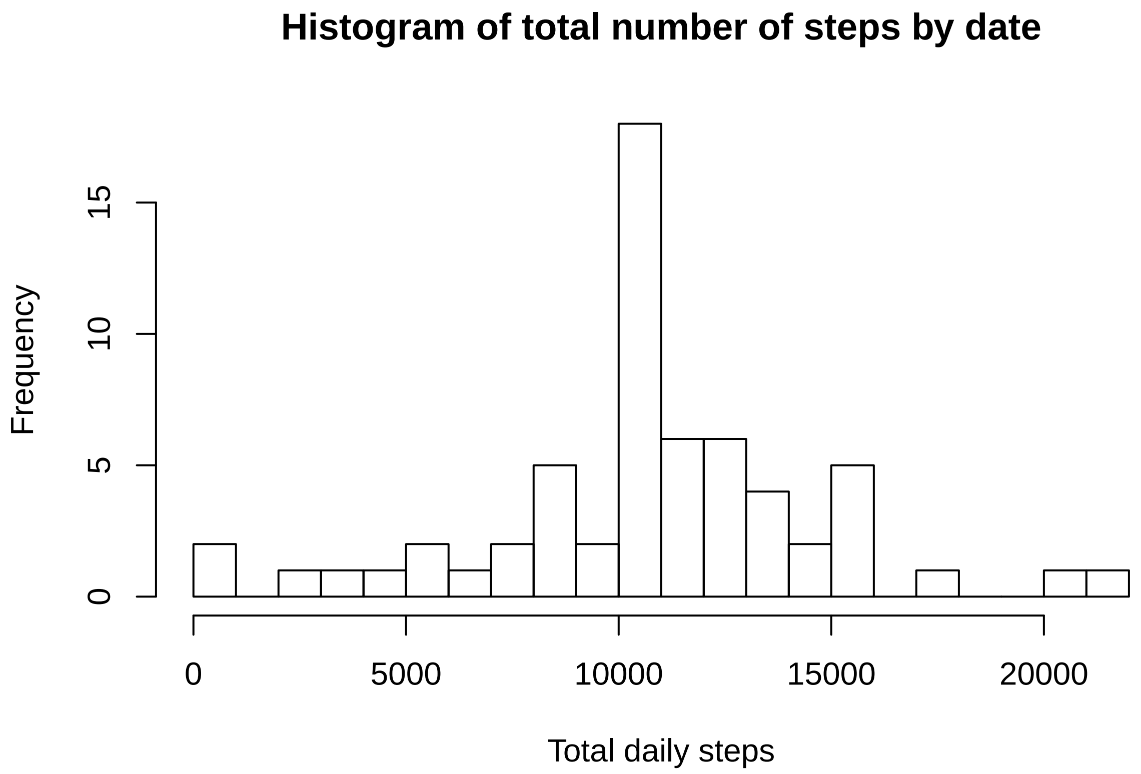
Summary of new dataset

```
summary(stepsbydate2)
```

```
##          date        nsteps
##  2012-10-01: 1   Min.   :   41
##  2012-10-02: 1   1st Qu.: 9819
##  2012-10-03: 1   Median :10766
##  2012-10-04: 1   Mean   :10766
##  2012-10-05: 1   3rd Qu.:12811
##  2012-10-06: 1   Max.   :21194
##  (Other)   :55
```

Draw histogram with new data

```
hist(stepsbydate2$nsteps,
     xlab = 'Total daily steps',
     main = 'Histogram of total number of steps by date',
     breaks = 20)
```



Also, recalculate mean and median of the total number steps taken per day.

```
mean2 <- mean(stepsbydate2$nsteps)
```

```
median2 <- median(stepsbydate2$nsteps)
```

Do, the old and new values of mean and median differ?

```
data.frame(mean_diff = mean2 - mean1, median_diff = median2 - median1)
```

```
##   mean_diff median_diff
## 1         0    1.188679
```

8. Panel plot comparing the average number of steps taken per 5-minute interval across weekdays and weekends

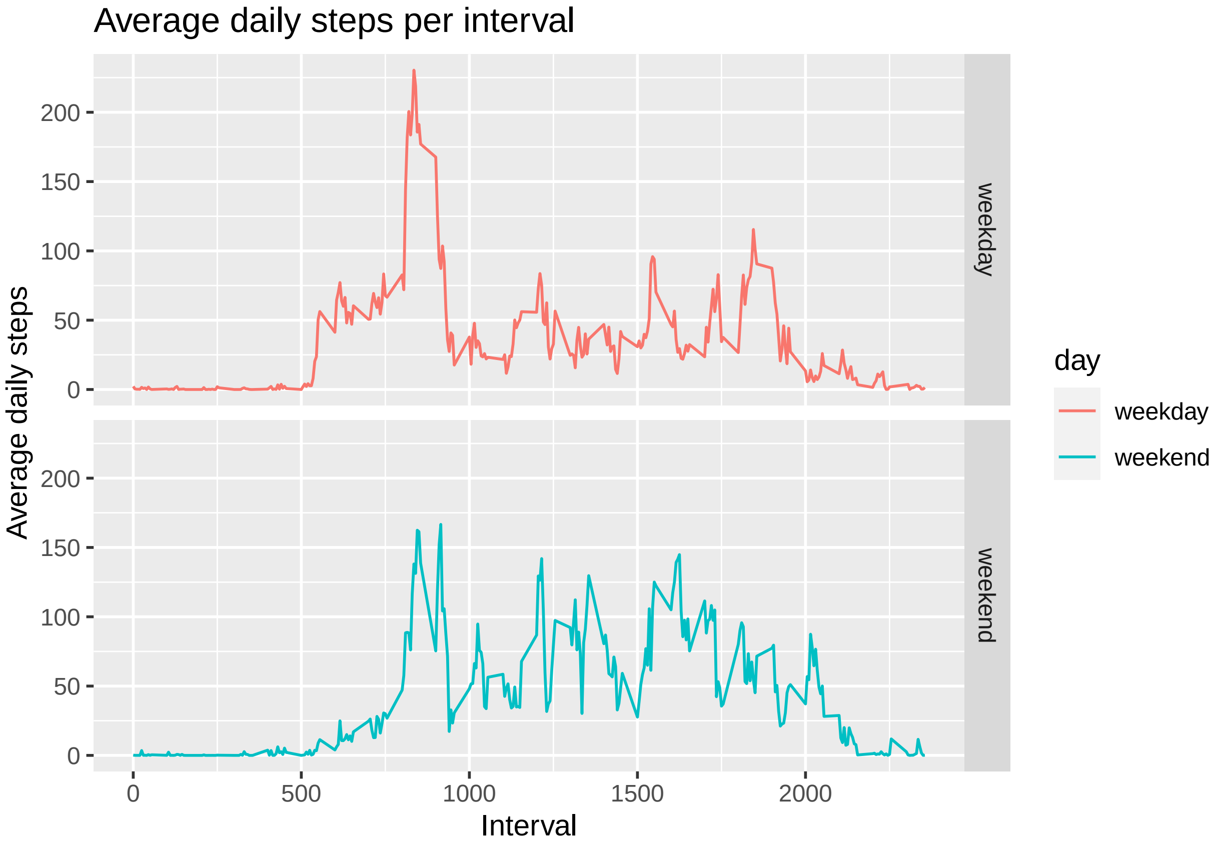
First, we add a new column which indicates the day is `weekend` or `weekday`.

```
stepsdatawithday <- replaced.stepdata %>%
  mutate(day = weekdays(as.Date(date))) %>%
  mutate(day = ifelse(day == 'Saturday' | day == 'Sunday', 'weekend', 'weekday'))

table(stepsdatawithday$day)
```

```
##
## weekday weekend
##   12960    4608
```

Then, we calculate the average number of steps grouped by interval

```
plotdata <- stepsdatawithday %>%
  select(steps, interval, day) %>%
  group_by(day, interval) %>%
  summarise(nsteps = mean(steps))

ggplot(plotdata, aes(x = interval, y = nsteps, color = day)) + geom_line() +
  facet_grid(day ~ .) + xlab('Interval') + ylab('Average daily steps') +
  ggtitle('Average daily steps per interval')
```


Average daily steps per interval
Average daily steps
weekday
weekend
Interval
day
weekday
weekend