Course Project 1

Ignacio Ojea

July 8, 2016

Part 0: Loading and preprocessing the data

The first step consists in (a) downloading the file, (b) unzipping it, and (c) read the .cvs into the data table.

```
temp <- tempfile()
download.file("https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip", temp)
data <- read.csv(unz(temp, "activity.csv"))
```

Let me now format the data a little bit:

```
data$date <- as.POSIXct(data$date, format="%Y-%m-%d")
data$steps <- as.numeric(data$steps)
```

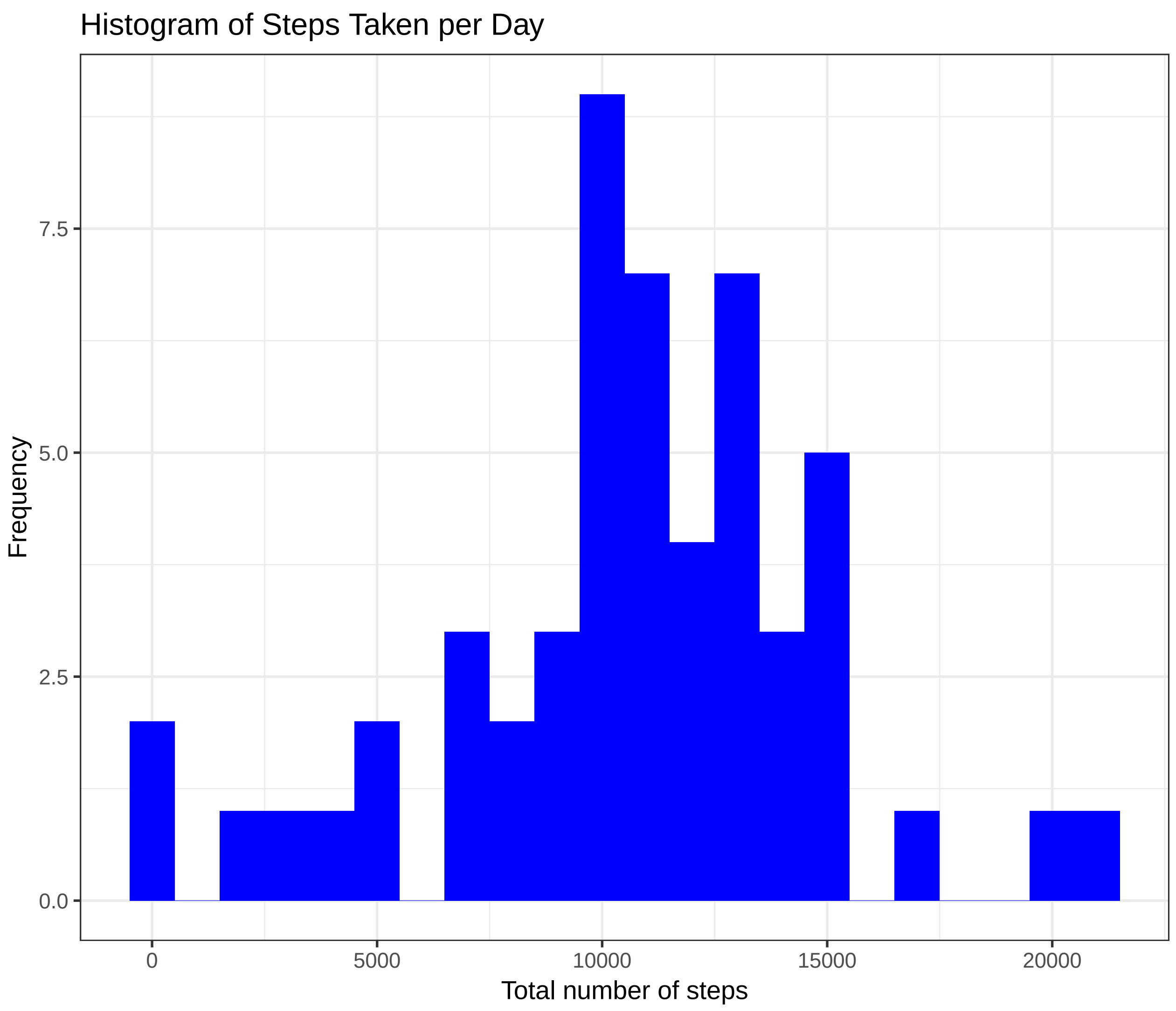
Part 1: What is mean total number of steps taken per day?

Compute the total number of steps per day:

```
stepsperday <- aggregate(steps ~ date, data, sum)
names(stepsperday) <- c("date", "total.steps")
```

Lets plot this into a histogram.

```
library(ggplot2)
ggplot(stepsperday, aes(x = total.steps)) +
       geom_histogram(fill = "blue", binwidth = 1000) +
        labs(title="Histogram of Steps Taken per Day",
             x = "Total number of steps", y = "Frequency") + theme_bw()
```



Now for the mean and median per day:

```
mean(stepsperday$total.steps)
```

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

```
median(stepsperday$total.steps)
```

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

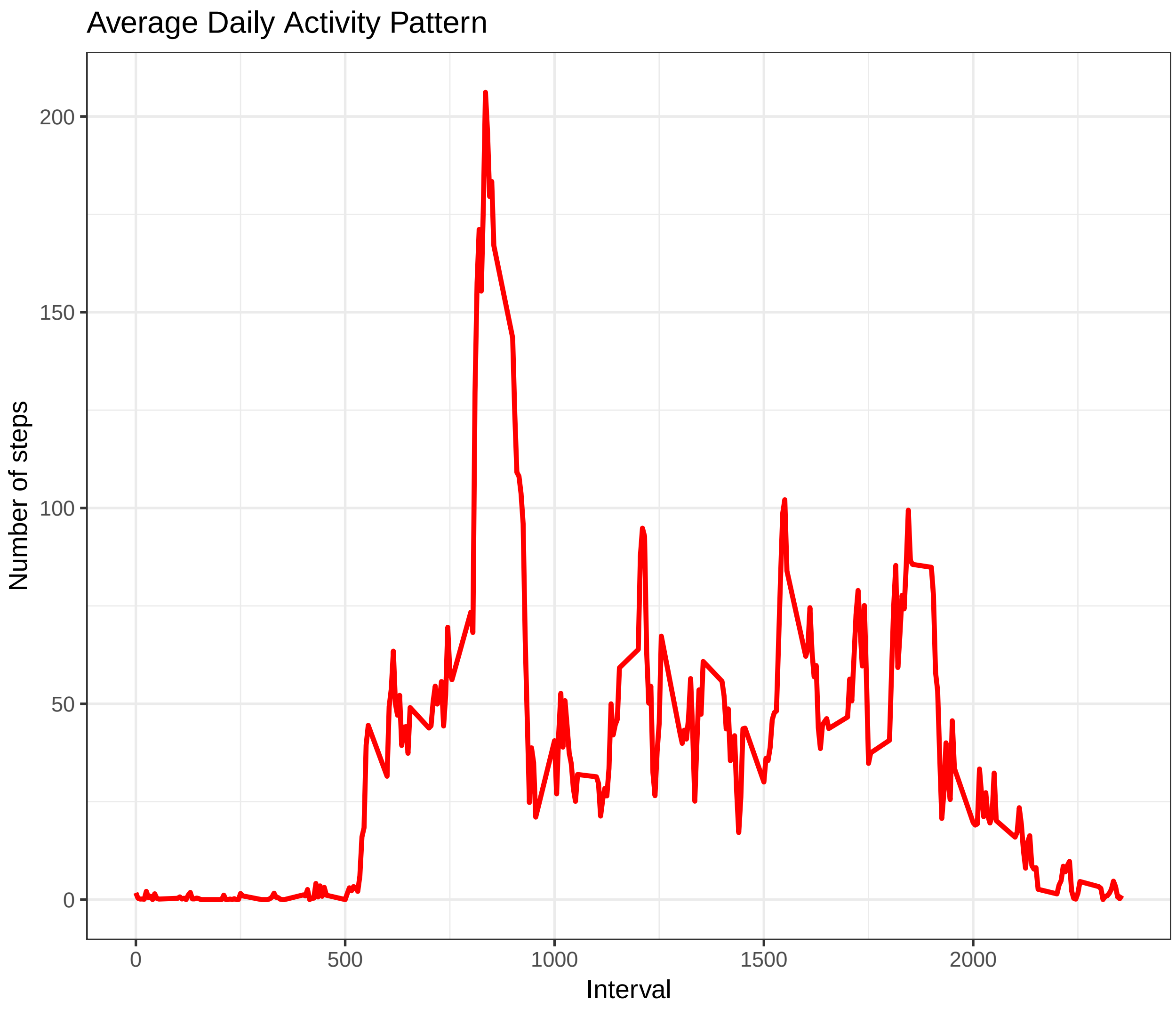
Part 2: What is the average daily activity pattern?

Compute the total number of steps per interval:

```
stepsperinterval <- aggregate(steps ~ interval, data, mean)
names(stepsperinterval) <- c("interval", "mean.steps")
```

Plot the data

```
ggplot(stepsperinterval, aes(x=interval, y=mean.steps)) +
        geom_line(color="red", size=1) +
        labs(title="Average Daily Activity Pattern", x="Interval", y="Number of steps") +
        theme_bw()
```



Find the maximum:

```
stepsperinterval[which.max(stepsperinterval$mean.steps),]
```

```
##     interval mean.steps
## 104      835   206.1698
```

Part 3: Imputing missing values

Number of missing values:

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

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

Strategy for filling in all of the missing values in the dataset, by the mean that interval accross all days.

```
filled.data <- data
missing.values <- is.na(filled.data$steps)
```

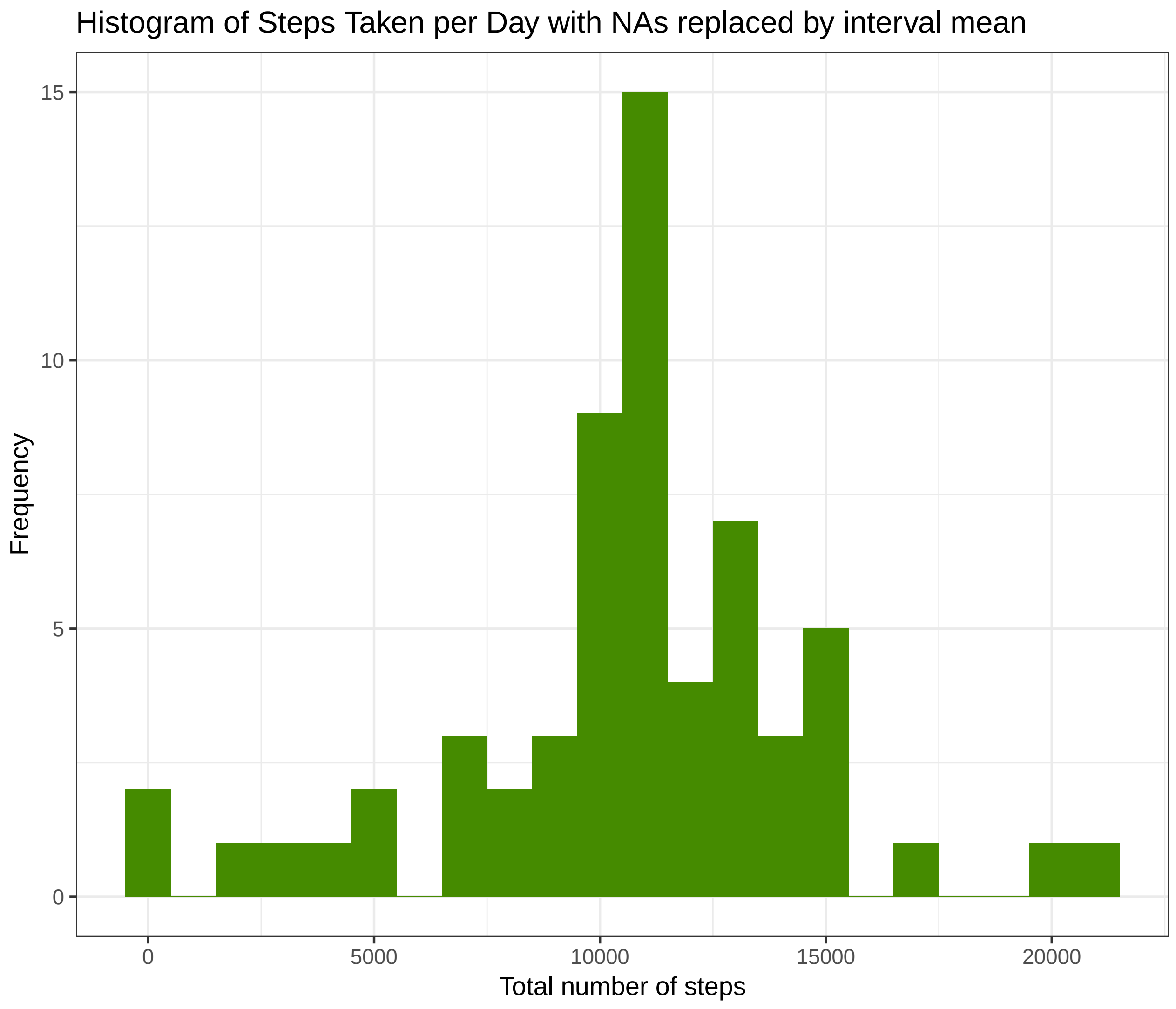
```r
avg.interval <- tapply(filled.data$steps, filled.data$interval, mean, na.rm=TRUE)
filled.data$steps[missing.values] <- avg.interval[as.character(filled.data$interval[missing.values])]
filled.data$steps <- as.numeric(filled.data$steps)
```

Compute the total number of steps per day in the new filled data:

```r
filled.stepsperday <- aggregate(steps ~ date, filled.data, sum)
names(filled.stepsperday) <- c("date", "total.steps")
```

Lets plot this into a histogram.

```r
library(ggplot2)
ggplot(filled.stepsperday, aes(x = total.steps)) +
       geom_histogram(fill = "chartreuse4", binwidth = 1000) +
        labs(title="Histogram of Steps Taken per Day with NAs replaced by interval mean",
             x = "Total number of steps", y = "Frequency") + theme_bw()
```



Now for the mean and median per day:

```r
mean(filled.stepsperday$total.steps)
```

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

```r
median(filled.stepsperday$total.steps)
```

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

The main impact of replacing the missing data with the corresponding interval average accross days is that the mean and median have the same value, namely 10766.

Part 4: Are there differences in activity patterns between weekdays and weekends?

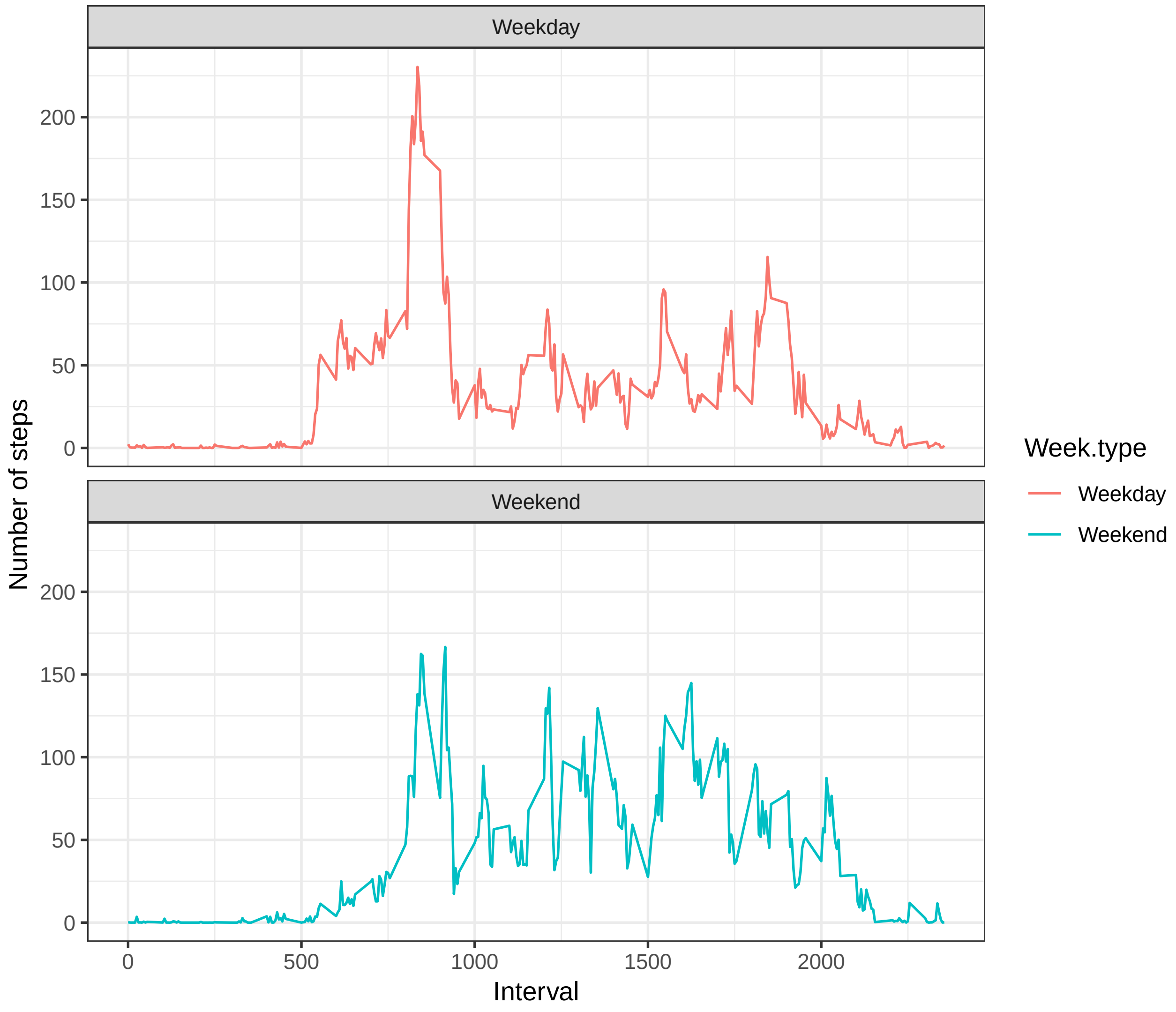
Creating a new factor variable for weekday or weekend, by defining a function:

```r
filled.data$Day.of.week <- weekdays(filled.data$date)
wdfunction <- function(date) {
    weekday <- weekdays(date)
    if (weekday %in% c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday"))
        return("Weekday")
    else if (weekday %in% c("Saturday", "Sunday"))
        return("Weekend")
    else
        stop("invalid date")
}
filled.data$Week.type <- sapply(filled.data$date, wdfunction)
```

Now for the plotting:

```r
average.data <- aggregate(steps ~ interval + Week.type, filled.data, mean)

ggplot(average.data, aes(x=interval, y=steps, color = Week.type)) +
        geom_line() +
        facet_wrap(~ Week.type, nrow=2, ncol=1) +
        labs(x="Interval", y="Number of steps") +
        theme_bw()
```



Weekday
Weekend
Number of steps
0
50
100
150
200
0
500
1000
1500
2000
Interval
Week.type
Weekday
Weekend