Activity Data Analysis

Sebastian Ruecker

28 Februar 2017

In the following document we will process and analyze a small data set containing the number of steps taken per time interval. The steps were measured via a personal mobile device and are all from the same subject.

Data Preparation

First we load the required packages and load the data. We generate the weekday and a variable denoting the weekend. We then take a look at the first couple of rows.

```
rm(list = ls())
Sys.setenv(LANG = "en")
Sys.setlocale("LC_TIME", "English")
```

```
## [1] "English_United States.1252"
```

```
wd <- "C:/Users/Sebastian/Documents/Coursera/Johns Hopkins - Data Science/5 Reproducible Research"
setwd(wd)

require("downloader")
```

```
## Loading required package: downloader
```

```
require("ggplot2")
```

```
## Loading required package: ggplot2
```

```
require("dplyr")
```

```
## Loading required package: 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
```

```
url <- "https://d396qusza40orc.cloudfront.net/repdata%2Fdata%2Factivity.zip"
#download(url, dest="./data/dataset.zip", mode="wb")
#unzip ("./data/dataset.zip", exdir = "./data")
file.remove("./data/dataset.zip")
```

```
## Warning in file.remove("./data/dataset.zip"): cannot remove file './data/
## dataset.zip', reason 'No such file or directory'
```

```
## [1] FALSE
```

```
activity <- read.csv("./data/activity.csv")
activity$date <- as.Date(activity$date)
activity$weekday <- weekdays(activity$date)
activity$weekend <- activity$weekday %in% c("Sunday", "Saturday")
summary(activity)
```

```
##      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
##   weekend
##  Mode :logical
##  FALSE:12960
##  TRUE :4608
##  NA's :0
##
##
##
```

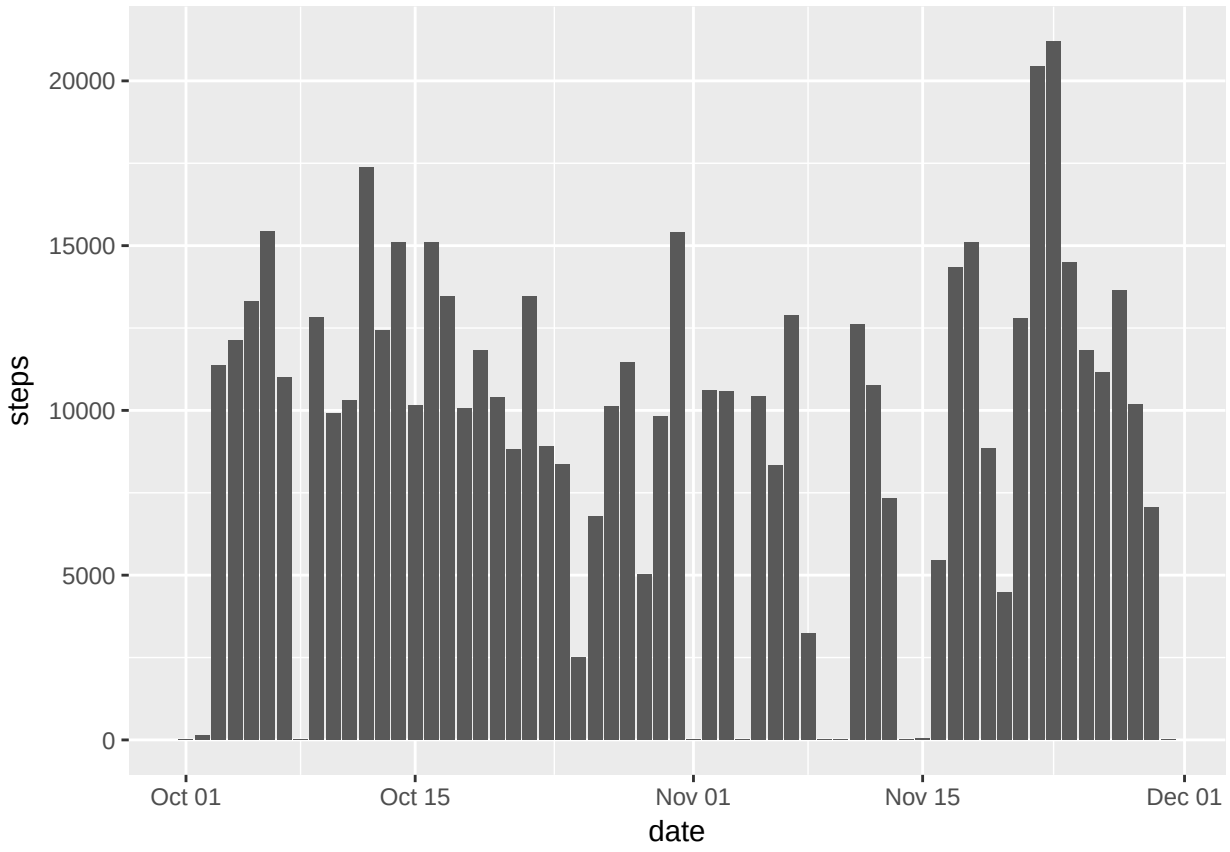
```
head(activity)
```

```
##   steps       date interval weekday weekend
## 1    NA 2012-10-01        0  Monday   FALSE
## 2    NA 2012-10-01        5  Monday   FALSE
## 3    NA 2012-10-01       10  Monday   FALSE
## 4    NA 2012-10-01       15  Monday   FALSE
## 5    NA 2012-10-01       20  Monday   FALSE
## 6    NA 2012-10-01       25  Monday   FALSE
```

Total number of steps taken per day

We see from the histogram, that the steps per day lie mostly between 5000 and 1500.

```
activity_by_day <- group_by(activity, date) %>% summarize(steps=sum(steps, na.rm=TRUE), avg_steps=mean(s
ggplot(data=activity_by_day, mapping=aes(x = date, y=steps)) + geom_bar(stat="identity")
```



Mean and median number of steps per day

Next we will have a look at the mean and median number of steps taken each day.

```
summary(activity_by_day[,2])
```

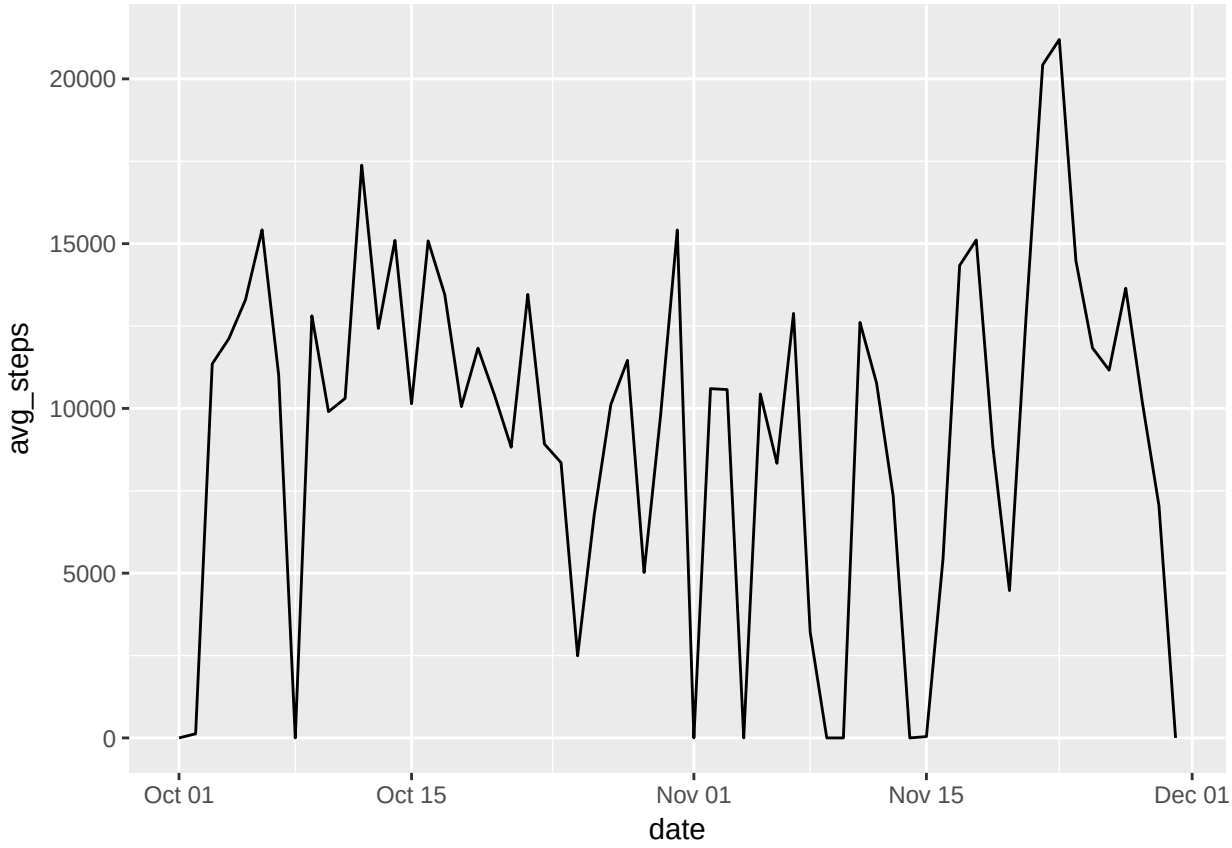
```
##      steps
##  Min.   :    0
##  1st Qu.: 6778
##  Median :10395
##  Mean   : 9354
##  3rd Qu.:12811
##  Max.   :21194
```

Some days appear to have no activity. This may be due to missing values or extreme laziness on the part of the subject.

Average number of steps taken

When we look at the average number of steps per day, we see that the number varies quite a lot. It is unclear, how much of the downward outliers are due to missing values.

```
ggplot(data=activity_by_day, mapping=aes(x = date, y=avg_steps)) + geom_line()
```



5-minute interval with maximum average number of steps

Next we identify the 5-minute interval in which the average number of steps over all days was greatest.

```
activity_by_interval <- group_by(activity, interval) %>% summarize(avg_steps=mean(steps, na.rm=TRUE))
activity_by_interval$interval[which.max(activity_by_interval$avg_steps)]
```

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

Missing data treatment

As mentioned above, missing data may be skewing our plots. Lets have a look which columns contain missing values and how many such values there are.

```
apply(activity, 2, FUN=function(x) any(is.na(x)))
```

```
##    steps     date interval  weekday  weekend
##     TRUE    FALSE    FALSE    FALSE    FALSE
```

```
mean(is.na(activity$steps))
```

```
## [1] 0.1311475
```

As we see, a good 13% of steps are missing. We will impute these values by the following method:

1. group the data by weekday and interval

2. take median over grouped data

3. fill missing values by using the median of the matching group

This way, we use the most commonly occurring number of steps per weekday and time interval as an estimation for the missing data points. We hope that this method is more robust than using the mean and more exact than using the median over the whole set.

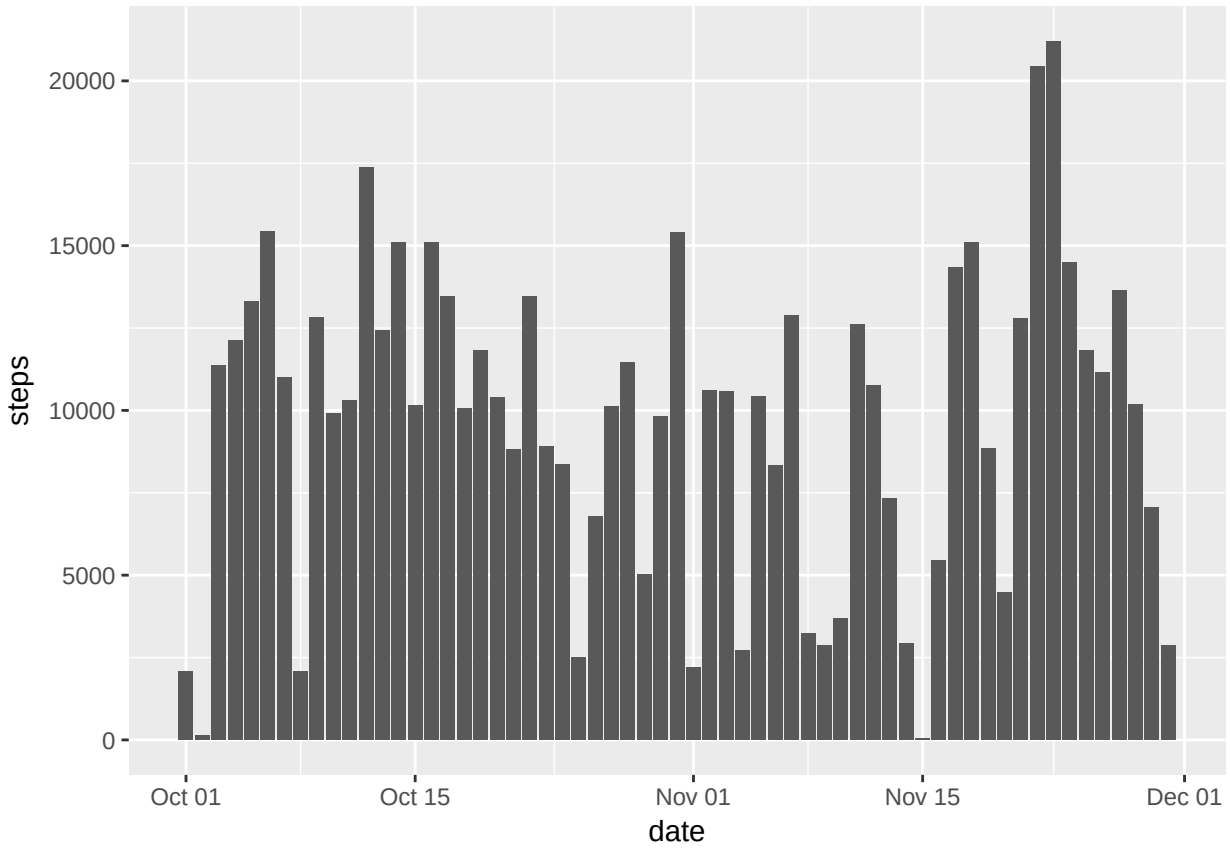
```
activity_by_day_interval <- group_by(activity, weekday, interval) %>%
  summarize(med_steps=median(steps, na.rm=TRUE)) %>%
  mutate(day_interval = paste(weekday, interval, sep=""))
na_idx <- which(is.na(activity$steps))
median_values <- activity_by_day_interval$med_steps
names(median_values) <- activity_by_day_interval$day_interval
activity_padded <- activity
activity_padded$steps[na_idx] <- median_values[paste(activity$weekday[na_idx], activity$interval[na_idx]
apply(activity_padded, 2, FUN=function(x) any(is.na(x)))
```

```
##    steps     date interval  weekday  weekend
##    FALSE    FALSE    FALSE    FALSE    FALSE
```

Total number of steps after missing values are imputed

Next, we will have a look at the data after missing values have been treated. As we can see, the missing values made the steps seem artificially low in some cases.

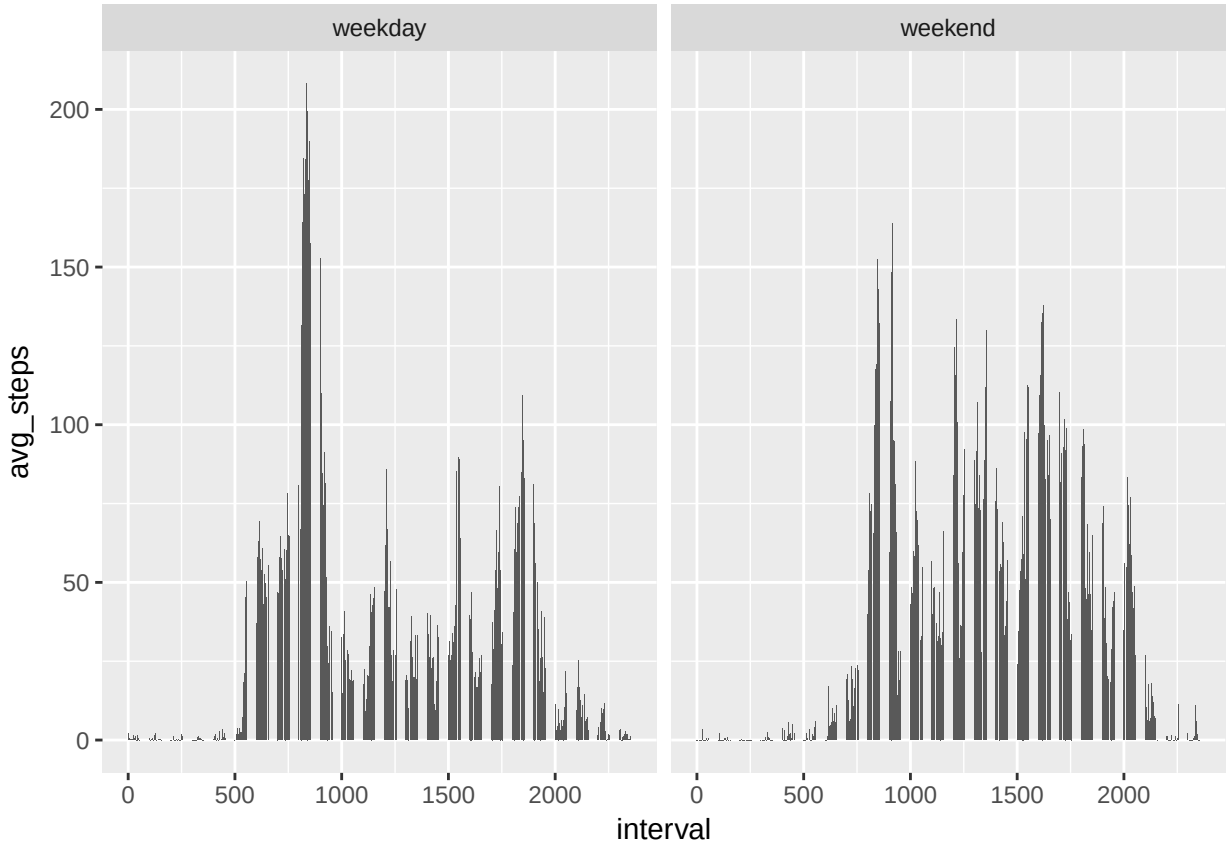
```
activity_by_day_padded <- group_by(activity_padded, date) %>% summarize(steps=sum(steps, na.rm=TRUE), a
ggplot(data=activity_by_day_padded, mapping=aes(x = date, y=steps)) + geom_bar(stat="identity")
```



Comparing the average number of steps taken per 5-minute interval across weekdays and weekends

Finally, we compare the average steps taken per interval during the week with those during the weekend. The weekend less spikes in steps. One might hypothesize, that the measured individual is lazier during the weekends, but more research would have to be conducted in order to be sure.

```
activity_by_intervall_weekend <- group_by(activity_padded, interval, weekend) %>% summarize(avg_steps=me
activity_by_intervall_weekend$weekend <- ifelse(activity_by_intervall_weekend$weekend, "weekend", "week
ggplot(data=activity_by_intervall_weekend, mapping=aes(x = interval, y=avg_steps)) +
  geom_bar(stat="identity") +
  facet_wrap(~ weekend)
```


weekday
weekend
avg_steps
200
150
100
50
0
0
500
1000
1500
2000
interval