Managing and Understanding Data

Laura Julià Melis, Víctor Navarro Garcés i Marta Piñol Palau

18 de February, 2018

Contents

- Primera toma de contacto con un informe din?mico donde se muestra algunas de sus caracteristicas.
- Son diferentes trozos de un libro.
- Se lee un archivo csv

2018-02-18

By the end of this notes, you will understand:

- The basic R data structures and how to use them to store and extract data
- How to get data into R from a variety of source formats
- Common methods for understanding and visualizing complex data

R Markdown

R Markdown is an easy-to-use system that enables students to combine statistical computing in an environment of their choosing and written analysis in one document. At a high-level, it renders a well-annotated R script into a self-contained HTML file, replete with graphics, commands, and stylized text.

Like LATEX or HTML, R Markdown relies on a source file and output file paradigm. Text, with simple rules for creating styles, is typed into an R Markdown source file, which has the `.Rmd` extension. R commands are typed directly into this file, set off in `chunks`. The `knitr` rendering engine then parses the `.Rmd` file. It first executes each of the R commands in the `chunks` and processes the output from those commands. This generates an intermediate Markdown file (with a `.md` extension) which is of no immediate interest. Next, it renders this Markdown file into a single HTML file with embedded graphics.

This information is from (Baumer et al. 2014)

R data structures

The R data structures used most frequently in machine learning are *vectors*, *lists*, and *data frames*.

Vectors

The fundamental R data structure is the **vector**, which stores an ordered set of values called **elements**. A vector can contain any number of elements. However, all the elements must be of the same type; for instance, a vector cannot contain both numbers and text.

There are several vector types commonly used in machine learning:`integer`(numbers without decimals), `numeric` (numbers with decimals), `character` (text data), or `logical` (`TRUE` or `FALSE` values). There are also two special values: `NULL`, which is used to indicate the absence of any value, and `NA`, which indicates a missing value.

More information (Lantz 2015, 28–30)

Data Frames

Data frames are the preferred way for organizing data sets that are of modest size. Fpr now, think of data grames as a rectangurlar row by column layout, where the rows are observations and the columns are variables.

This information is from (Maindonald and Braun 2006, 10:4) For More information (Maindonald and Braun 2006 , section: 1.4 subsection: 1.2)

Lists

Like vectors, lists are ordered sequences of elements, but unlike vectors, the elements of a list can themselces have more than one element. Thus we can have lists of vectors, lists of data frames, or lists containing a mixture of numbers, strings, vectors, data frames and other lists.

Lists are created with the `list()` function.

This information is from (Baayen 2008, 16)

Create vectors of data for three medical patients:

```
# create vectors of data for three medical patients
subject_name <- c("John Doe", "Jane Doe", "Steve Graves")
temperature <- c(98.1, 98.6, 101.4)
flu_status <- c(FALSE, FALSE, TRUE)
```

Access the second element in body temperature vector:

```
# access the second element in body temperature vector
temperature[2]
```

```
## [1] 98.6
```

Examples of accessing items in vector include items in the range 2 to 3.

```
## examples of accessing items in vector
# include items in the range 2 to 3
temperature[2:3]
```

```
## [1]  98.6 101.4
```

Exclude item 2 using the minus sign

```
# exclude item 2 using the minus sign
temperature[-2]
```

```
## [1]  98.1 101.4
```

Use a vector to indicate whether to include item

```
# use a vector to indicate whether to include item
temperature[c(TRUE, TRUE, FALSE)]
```

```
## [1] 98.1 98.6
```

Exploring and understanding data

After collecting data and loading it into R data structures, the next step in the machine learning process involves examining the data in detail. It is during this step that you will begin to explore the data's features and examples, and realize the peculiarities that make your data unique. The better you understand your data, the better you will be able to match a machine learning model to your learning problem. The best way to understand the process of data exploration is by example. In this section, we will explore the *usedcars.csv* dataset, which contains actual data about used cars recently advertised for sale on a popular U.S. website.

...

...

...

Since the dataset is stored in CSV form, we can use the `read.csv()` function to load the data into an R data frame:

```
##### Exploring and understanding data --------------------

## data exploration example using used car data
usedcars <- read.csv(file1, stringsAsFactors = FALSE)
```

Exploring the structure of data

One of the first questions to ask in your investigation should be about how data is organized. If you are fortunate, your source will provide a **data dictionary**, a document that describes the data's features. In our case, the used car data does not come with this documentation, so we'll need to create our own.

```
# get structure of used car data
str(usedcars)
```

```
## 'data.frame':    150 obs. of  6 variables:
##  $ year        : int  2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...
##  $ model       : chr  "SEL" "SEL" "SEL" "SEL" ...
##  $ price       : int  21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...
##  $ mileage     : int  7413 10926 7351 11613 8367 25125 27393 21026 32655 36116 ...
##  $ color       : chr  "Yellow" "Gray" "Silver" "Gray" ...
##  $ transmission: chr  "AUTO" "AUTO" "AUTO" "AUTO" ...
```

Show some registers

```
# Table of 6 first registers
kable(head(usedcars))
```

year	model	price	mileage	color	transmission
2011	SEL	21992	7413	Yellow	AUTO
2011	SEL	20995	10926	Gray	AUTO
2011	SEL	19995	7351	Silver	AUTO
2011	SEL	17809	11613	Gray	AUTO
2012	SE	17500	8367	White	AUTO
2010	SEL	17495	25125	Silver	AUTO

Exploring numeric variables

```
## Exploring numeric variables -----

# summarize numeric variables
summary(usedcars$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2000    2008    2009    2009    2010    2012
```

```
summary(usedcars[c("price", "mileage")])
```

```
##      price          mileage
##  Min.   : 3800   Min.   :  4867
##  1st Qu.:10995   1st Qu.: 27200
##  Median :13592   Median : 36385
##  Mean   :12962   Mean   : 44261
##  3rd Qu.:14904   3rd Qu.: 55124
##  Max.   :21992   Max.   :151479
```

```
# calculate the mean income
(36000 + 44000 + 56000) / 3
```

```
## [1] 45333.33
```

```
mean(c(36000, 44000, 56000))
```

```
## [1] 45333.33
```

```
# the median income
median(c(36000, 44000, 56000))
```

```
## [1] 44000
```

```
# the min/max of used car prices
range(usedcars$price)
```

```
## [1]  3800 21992
```

```
# the difference of the range
diff(range(usedcars$price))
```

```
## [1] 18192
```

```r
# IQR for used car prices
IQR(usedcars$price)
```

```
## [1] 3909.5
```

```r
# use quantile to calculate five-number summary
quantile(usedcars$price)
```

```
##      0%     25%     50%     75%    100%
##  3800.0 10995.0 13591.5 14904.5 21992.0
```

```r
# the 99th percentile
quantile(usedcars$price, probs = c(0.01, 0.99))
```

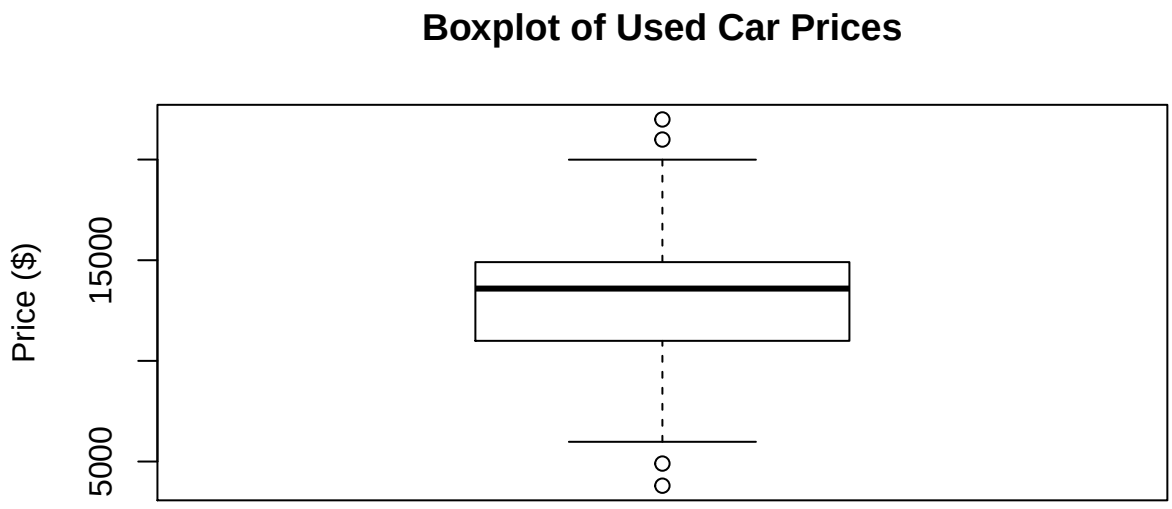
```
##       1%      99%
##  5428.69 20505.00
```

```r
# quintiles
quantile(usedcars$price, seq(from = 0, to = 1, by = 0.20))
```

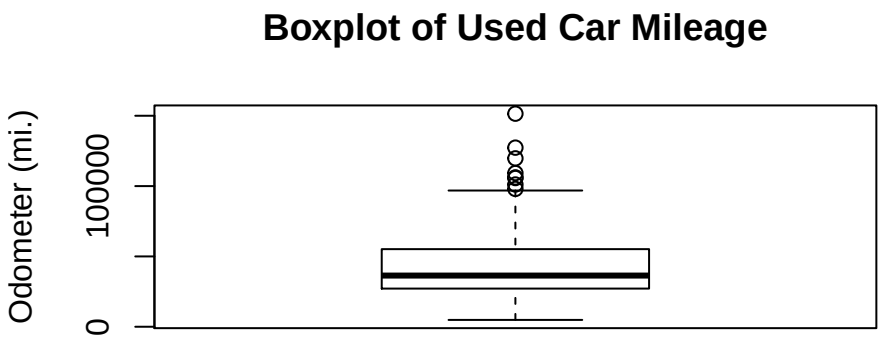
```
##      0%     20%     40%     60%     80%    100%
##  3800.0 10759.4 12993.8 13992.0 14999.0 21992.0
```

Visualizing numeric variables - boxplots

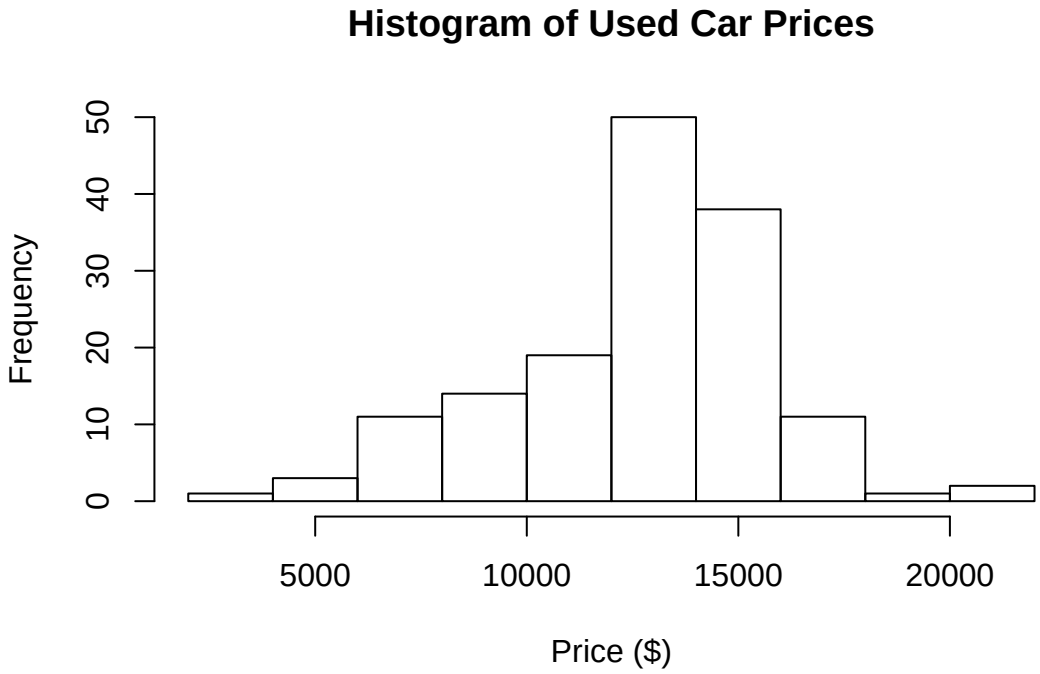
```r
# boxplot of used car prices and mileage
boxplot(usedcars$price, main="Boxplot of Used Car Prices",ylab="Price ($)")
```



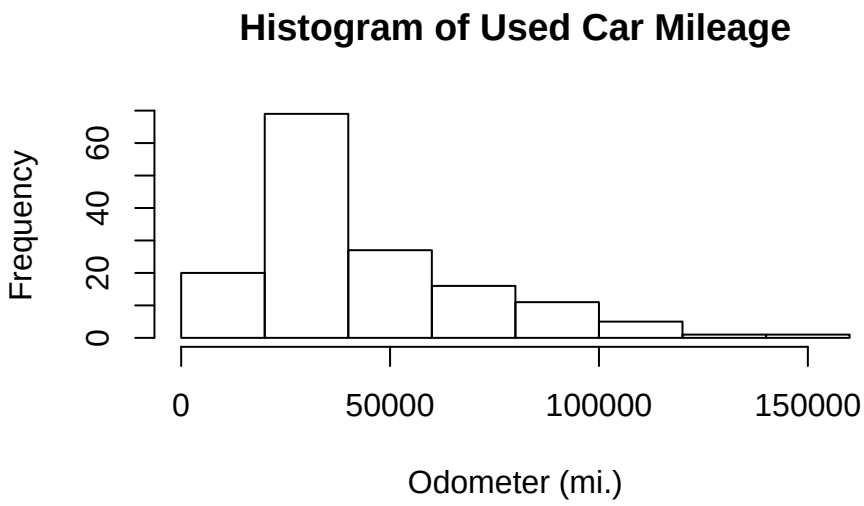

```r
boxplot(usedcars$mileage, main="Boxplot of Used Car Mileage",
      ylab="Odometer (mi.)")
```




```
# histograms of used car prices and mileage
hist(usedcars$price, main = "Histogram of Used Car Prices",
     xlab = "Price ($)")
```




```
hist(usedcars$mileage, main = "Histogram of Used Car Mileage",
     xlab = "Odometer (mi.)")
```




```
# variance and standard deviation of the used car data
var(usedcars$price)
```

```
## [1] 9749892
```

```
sd(usedcars$price)
```

```
## [1] 3122.482
```

```
var(usedcars$mileage)
```

```
## [1] 728033954
```

```
sd(usedcars$mileage)
```

```
## [1] 26982.1
```

Measuring spread - quartiles and the five-number summary

The **five-number summary** is a set of five statistics that roughly depict the spread of a dataset. All five of the statistics are included in the output of the `summary()` function. Written in order, they are:

1. Minimum (`Min.`)
2. First quartile, or Q1 (`1st Qu.`)
3. Median, or Q2 (`Median`)
4. Third quartile, or Q3 (`3rd Qu.`)
5. Maximum (`Max.`)

Measuring spread - variance and standard deviation

In order to calculate the standard deviation, we must first obtain the **variance**, which is defined as the average of the squared differences between each value and the mean value. In mathematical notation, the variance of a set of `n` values of `x` is defined by the following formula. The Greek letter `mu` (μ) (similar in appearance to an m) denotes the mean of the values, and the variance itself is denoted by the Greek letter `sigma` (σ) squared (similar to a b turned sideways):

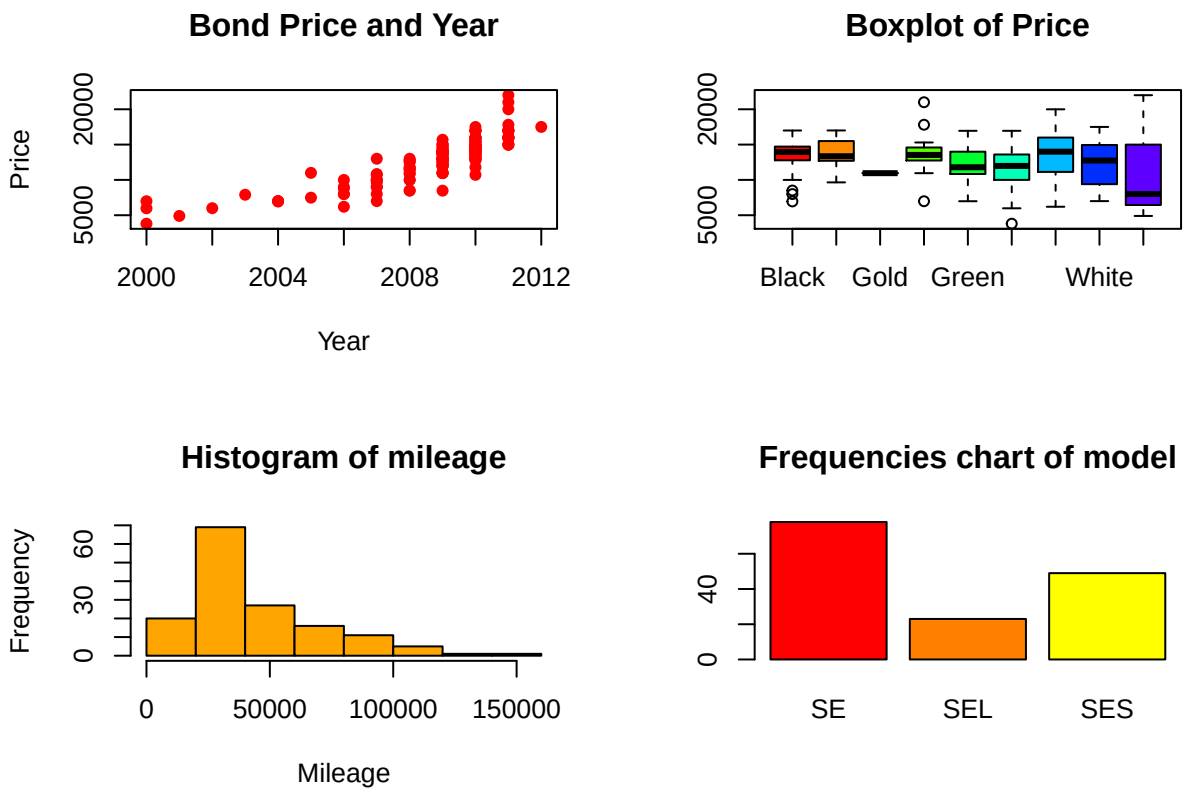
$$Var(X) = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

The standard deviation is the square root of the variance, and is denoted by `sigma` as shown in the following formula:

$$StdDev(X) = \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

Chunk that shows several graphics together.

```
par(mfrow=c(2,2))
plot(usedcars$year,usedcars$price,xlab='Year', ylab='Price', main= 'Bond Price and Year', pch=16, col='
boxplot(price~color,data=usedcars, main= 'Boxplot of Price', col=rainbow(11))
hist(usedcars$mileage, col='orange', main='Histogram of mileage', xlab='Mileage')
barplot(table(usedcars$model), col=heat.colors(3), main='Frequencies chart of model')
```



Chunk that shows a data table

```
kable(cor(usedcars[,c(1,3,4)]), caption='A data table: Correlation between Year, Price and Mileage.', al
```

Table 2: A data table: Correlation between Year, Price and Mileage.

	year	price	mileage
year	1.0000000	0.8450041	-0.7603127
price	0.8450041	1.0000000	-0.8061494
mileage	-0.7603127	-0.8061494	1.0000000

Note. For more details on using mathematical expressions in Latex (R Markdown) see https://es.sharelatex.com/learn/Mathematical_expressions.

Baayen, R Harald. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*. Cambridge University Press.

Baumer, Ben, Mine Cetinkaya-Rundel, Andrew Bray, Linda Loi, and Nicholas J Horton. 2014. "R Markdown: Integrating a Reproducible Analysis Tool into Introductory Statistics." *arXiv Preprint arXiv:1402.1894*.

Lantz, Brett. 2015. *Machine Learning with R*. Packt Publishing Ltd.

Maindonald, John, and John Braun. 2006. *Data Analysis and Graphics Using R: An Example-Based Approach*. Vol. 10. Cambridge University Press.