Data Science

Exploratory Data Analysis

Using R

- Satishkuar L. Varma

Data Science

Module 1: Introduction to Data Science

Module 2 : Predictive and Descriptive Models

Module 3 : Evaluation and Methodology of Data Science

Module 4: Text Analytics and Recommendation system (RS)

Module 5 : Data Communication and Information Visualization

Module 6 : Scaling with Big Data

Data Science: Introduction to Data Science

- Data science process
  - Defining goal
  - Retrieving data
  - Pre-processing data
  - **Exploratory data analysis**
  - Model building and data visualization
  - Ethical issues in data science
- Probability
  - Review of probability theory
  - Normal distribution
- Gaussian discriminant analysis
  - Linear discriminant analysis (LDA)
- Logistic regression
  - Bayesian logistic regression

Introduction to Data Science: Exploratory data analysis

- Exploratory Data Analysis Using R covers
  - Ways to summarize and visualize important characteristics of a data set.
- **Useful R Functions for Exploring a Data Frame**
  - Data in R are often stored in data frames,
    - because they can store multiple types of data.
  - In R, data frames are more general than matrices,
    - because matrices can only store one type of data.
  - Some common functions in R are used to explore a data frame before conducting any statistical analysis.
  - Let us use the built-in data set "InsectSprays"
    - to illustrate these functions,
    - because it contains categorical (character) and continuous (numeric) data, and
    - that allows us to show different ways of exploring these 2 types of data.

Useful R Functions for Exploring a Data Frame

dim() used to obtain the dimensions of the data frame (number of rows and number of columns). The output is a vector.

```
> dim(InsectSprays)
[1] 72  2
```

nrow() and ncol() are used to get the number of rows and number of columns, respectively. Also, get the same info by extracting first and second element of output vector from dim().

```
> nrow(InsectSprays)
# same as dim(InsectSprays)[1]
[1] 72
> ncol(InsectSprays)
# same as dim(InsectSprays)[2]
[1] 2
```

Useful R Functions for Exploring a Data Frame

head() used to obtain the first n observations and

tail() to obtain the last n observations; by default, n = 6.

Good commands for obtaining an intuitive idea of what the data look like without revealing the entire data set, which could have millions of rows and thousands of columns.

```
> head(InsectSprays, n = 5)
   count spray
1     10     A
2      7     A
3     20     A
4     14     A
5     14     A
```

Useful R Functions for Exploring a Data Frame

Let s be no. of observations. If we use a negative number for the "n" option in head(), we will obtain the first s+n observations.

Example: since s = 72 and s = -62, the following command will return the first 10 observations; the calculation is s+n = 72 + (-62) = 10.

```
> head(InsectSprays, n = -62)
   count spray
1     10     A
2      7     A
3     20     A
4     14     A
5     14     A
6     12     A
7     10     A
8     23     A
9     17     A
10    20     A
```

Useful R Functions for Exploring a Data Frame

Analogously, if we use a negative number for the "n" option in tail(),

We will get the last s+n observations.

Example, the following command will return the last 10 observations.

```
> tail(InsectSprays, n = -62)
   count spray
63    15     F
64    22     F
65    15     F
66    16     F
67    13     F
68    10     F
69    26     F
70    26     F
71    24     F
72    13     F
```

Useful R Functions for Exploring a Data Frame

names() function will return the column headers.

```
> names(InsectSprays)
[1] "count" "spray"
```

str() function returns many useful pieces of information, including the above useful outputs and the types of data for each column.

Example: "num" denotes that the variable "count" is numeric (continuous), and "Factor" denotes that the variable "spray" is categorical with 6 categories or levels.

```
> str(InsectSprays)
'data.frame': 72 obs. of 2 variables:
$ count: num 10 7 20 14 14 12 10 23 17 20 ...
$ spray: Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 1
  1 1 1 1 1 ...
```

- **Median:** It is the value that has just as many values above it as below it.
- **Mode:**
  - It is a value that occurs with the greatest frequency.
  - Depending on the data, there can be more than one mode.
  - When there are two modes, a distribution of values is referred to as bimodal.
- **Range:** The simplest measure of dispersion, the range is the difference between the highest and lowest values.
- **Midrange:** A variant of the range, is avg. of lowest data value and highest data value.
- **Quantiles:** After values are arranged from smallest to largest, quartiles are calculated similarly to the median. It may be necessary to interpolate (calculate a position between) two values to identify the data position corresponding to the quartile.

For N values arranged from lowest to highest:

$$\text{First quartile, } Q_1 = \text{Data value at position } \frac{(N+1)}{4}$$

$$\text{Second quartile (the median), } Q_2 = \text{Data value at position } \frac{2(N+1)}{4}$$

$$\text{Third quartile, } Q_3 = \text{Data value at position } \frac{3(N+1)}{4}$$

(Use N if data represent a population, n for a sample.)

Useful R Functions for Exploring a Data Frame

To obtain all of the categories or levels of a categorical variable, use the levels() function.

```
> levels(InsectSprays$spray)
[1] "A" "B" "C" "D" "E" "F"
```

For a data frame, the summary() function is essentially applied to each column, and the results for all columns are shown together.

For a continuous (numeric) variable like "count", it returns the 5-number summary.

Learn how fivenum() and summary() return different 5-number summaries.

If there are any missing values (denoted by "NA" for a particular datum), it would also provide a count for them.

Example: there are no missing values for "count", so there is no display for # NA's.

For a categorical variable like "spray", it returns the levels and # data in each level.

```
> summary(InsectSprays)
     count         spray
 Min.   : 0.00   A:12
 1st Qu.: 3.00   B:12
 Median : 7.00   C:12
 Mean   : 9.50   D:12
 3rd Qu.:14.25   E:12
 Max.   :26.00   F:12
```

Introduction to Data Science: Exploratory data analysis

- R has three basic indexing operators, with syntax displayed by the following examples

```
x[i]
x[i, j]
x[[i]]
x[[i, j]]
x$a
x$"a"
```

- For vectors and matrices the [[forms are rarely used, although they have some slight semantic differences from the [form (e.g. it drops any names or dimnames attribute, and that partial matching is used for character indices).
- When indexing multi-dimensional structures with a single index, x[[i]] or x[i] will return the ith sequential element of x.
- For lists, one generally uses [[to select any single element, whereas [returns a list of the selected elements.
- The [[form allows only a single element to be selected using integer or character indices, whereas [allows indexing by vectors. Note though that for a list, the index can be a vector and each element of the vector is applied in turn to the list, the selected component, the selected component of that

- a
- b
- c
- d

Data Science: Introduction to Data Science

- Data science process
  - Defining goal
  - Retrieving data
  - Pre-processing data
  - **Exploratory data analysis**
  - Model building and data visualization
  - Ethical issues in data science
- Probability
  - Review of probability theory
  - Normal distribution
- Gaussian discriminant analysis
  - Linear discriminant analysis (LDA)
- Logistic regression
  - Bayesian logistic regression

Module 1: Introduction to Data Science

Module 2 : Predictive and Descriptive Models

Module 3 : Evaluation and Methodology of Data Science

Module 4: Text Analytics and Recommendation system (RS)

Module 5 : Data Communication and Information Visualization

Module 6 : Scaling with Big Data

References

Reference Books:

1. Davy Cielen,Meysman,Mohamed Ali, “Introducing Data Science”, Dreamtech Press
2. Kevin P. Murphy, “Machine Learning a Probabilistic Perspective”, The MIT Press
3. Paul C. Zikopoulos, Chris Eaton, Dirk deRoos, Thomas Deutsch and George Lapis, “Understanding Big Data: Analytics for Enterprise Class Hadoop and streaming Data”, The McGraw Hill Companies, 2012 "Big Data: The next frontier for innovation, competition, and productivity". Rapporto McKinsey & Company, 2012.
4. Dean Abbott, “Applied Predictive Analytics: Principles and Techniques for the Professional Data Analyst”, Wiley, 2014
5. Noel Cressie, Christopher K. Wikle , “Statistics for Spatio-Temporal Data, Wiley
6. Seema Acharya and Subhashini Chellappan, “Big Data and Analytics”, Wiley
7. Rachel Schutt and Cathy O’Neil, “Doing Data Science”, O’Reilly Media
8. Joel Grus, Data Science from Scratch: First Principles with Python, O'Reilly Media
9. EMC Education Services,”Data Science and Big Data Analytics”, Wiley
10. DT Editorial Services, “Big Data Black Book”, Dreamtech Press