Do Data Science in 10 Hours

Gangmin Li

2020-09-29

To Shao Yong (邵雍),
for sharing a secret joy with simple words;

月到天心处，风来水面时。
一般清意味，料得少人知。

and

To Hongzhi Zhengjue (宏智禅师),
for sharing the peace of an ending life with simple words.

梦幻空华，六十七年；
白鸟淹没，秋水连天。

Contents

"Dont't touch it, It's hot!",

...

You now know it is hot. don't you?

How?

Because, you'v touched it!

Preface

0.1 Why Read this book

In this book, we will introduce an interesting method.

0.2 Structure of the Book

0.3 What Can This Course Offer You?

This short course is specifically designed for students who has no background of much Algorithms, programming and even computing. The basic requirements are desire to learn, attitude of humble and diligence of working. All that mean you need to get your hand dirty, spent time and do more practices. At the end, it cannot guaranty you become a data scientist but it will help you find the way to wards doing data science and be confident to start doing data science projects. It is cetainty that if you persist on this road, you will no doubt becomes a future data scientist.

This course will bring you:

1. Understand the data analysis procedure
2. Familiar with R language
3. Using RStudio to do your prototype of data project
4. Data preprocess - manipulate data for further analysis
5. Manage basic methods like Descriptive, Exploratory and Predictive data analysis
6. Have basic skills of visualize data results
7. Basic interpretation of you analyzing results and communicate with others
8. Report your results
9. enter the data science community

Okay, let us set off now.

0.4 Schedule

Setup Download files required for the lesson 00:00 1. Analyzing Patient Data How do I read data into R? How do I assign variables? What is a data frame? How do I access subsets of a data frame? How do I calculate simple statistics like mean and median? Where can I get help? How can I plot my data? 00:45 2. Creating Functions How do I make a function? How can I test my functions? How should I document my code? 01:15 3. Analyzing Multiple Data Sets How can I do the same thing to multiple data sets? How do I write a for loop? 01:45 4. Making Choices How do I make choices using if and else statements? How do I compare values? How do I save my plots to a PDF file? 02:15 5. Command-Line Programs How do I write a command-line script? How do I read in arguments from the command-line? 02:45 6. Best Practices for Writing R Code How can I write R code that other people can understand and use? 02:55 7. Dynamic Reports with knitr How can I put my text, code, and results all in one document? How do I use knitr? How do I write in Markdown? 03:15 8. Making Packages in R How do I collect my code together so I can reuse it and share it? How do I make my own packages? 03:45 9. Introduction to RStudio How do I use the RStudio graphical user interface? 04:00 10. Addressing Data What are the different methods for accessing parts of a data frame? 04:20 11. Reading and Writing CSV Files How do I read data from a CSV file into R? How do I write data to a CSV file? 04:50 12. Understanding Factors How is categorical data represented in R? How do I work with factors? 05:10 13. Data Types and Structures What are the different data types in R? What are the different data structures in R? How do I access data within the various data structures? 05:55 14. The Call Stack What is the call stack, and how does R know what order to do things in? How does scope work in R? 06:10 15. Loops in R How can I do the same thing multiple times more efficiently in R? What is vectorization? Should I use a loop or an apply statement? 06:40 Finish

0.5 Notes

Our goal is not to teach you R, but to teach you the basic process of doing a data science project that many other programming language like Java and Python can do. We use R in our lessons because:

- we have to use something for examples;
- it's free, well-documented, and runs almost everywhere;
- it has a large (and growing) user base among scientists; and
- it has a large library of external packages available for performing diverse tasks.

But the two most important things are to use whatever language your colleagues are using, so you can share your work with them easily, and to use that language well. apparently. R is the most used language in Data Science

0.6 Convention

info

R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see `http://www.gnu.org/licenses/`.

Instruction

R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see `http://www.gnu.org/licenses/`.

Todo

R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see `http://www.gnu.org/licenses/`.

hints

R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under the terms of the GNU General Public License versions 2 or 3. For more information about these matters see `http://www.gnu.org/licenses/`.

Acknowledgements

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): `https://yihui.name/tinytex/`.

> Some text for this block.

Introduction

> "What profession did Harvard call the Sexiest Job of the 21st Century?"
>
> ...
>
> That's right...
>
> The data scientist.

Ah yes, the ever mysterious data scientist. So what exactly is the data scientist's secret sauce, and what does this "sexy" person actually do at work every day? How so they do it?

0.7 What is Data Science?

Data science is a multidisciplinary filed. It blends of data mining, data analysis, statistics, algorithm development, machine learning and advanced computing and software technology together in order to solve analytically complex problems. Its ultimate goal is to reveal insight of data and get the data value for business.

0.7.1 Data science as Discovery of Data Insight

This aspect of data science is all about uncovering hidden patterns from data. Diving in at a granular level to mine and understand complex patterns, trends,

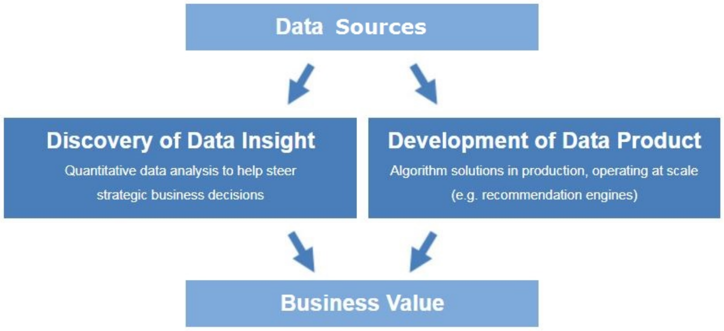

Data Sources
Discovery of Data Insight
Quantitative data analysis to help steer strategic business decisions
Development of Data Product
Algorithm solutions in production, operating at scale (e.g. recommendation engines)
Business Value

Figure 1: Concept of Data Science

and relations. It's about surfacing hidden insight that can help and enable companies to make smarter business decisions and take appropriate actions to gain competitive advantages in the market. For example:

- Amazon build recommendation system to provide users suggestion on purchase based on the user's shopping histroy.
- Netflix data mines movie viewing patterns to understand what drives user interest, and uses that to make decisions on which Netflix original series to produce.
- Target identifies what are major customer segments within it's base and the unique shopping behaviors within those segments, which helps to guide messaging to different market audiences.
- Proctor & Gamble utilizes time series models to more clearly understand future demand, which help plan for production levels more optimally. How do data scientists mine out insights? It starts with data exploration. When given a challenging question, data scientists become detectives. They investigate leads and try to understand pattern or characteristics within the data. This requires a big dose of analytical creativity.

How do data scientists mine data insights? there is a procedure to follow. It generally starts with data description it is called Described data analysis (DDA) to get first sight on the data sets available. DDS will help data scientist to grasp the quantity and quality of the data. so they can decide how to deal with the data. it then generally followed by data cleaning, manipulation, transform and attributes engineering etc, together called preprocess. Data preprocess is also generally combined with exploratory data analysis (EDA). When given a challenging question, data scientists normally become detectives. They investigate all the information available and follow any possible leads and try to understand pattern or characteristics within the data. This not only requires huge amount tools and techniques but also demand analytical creativity .

Then as needed, data scientists may apply quantitative technique in order to get a level deeper – e.g. statistical methods, projections, inferential models, segmentation analysis, time series forecasting, synthetic control experiments, etc. The intent is to scientifically piece together a forensic view of what the data is really saying.

This data-driven insight is central to providing strategic guidance. In this sense, data scientists act as consultants, information provider help business stakeholders on how to act on findings.

0.7.2 Data science as Development of Data Product

A "data product" is a technical asset that:

1. utilizes data as input, and
2. processes that data to return algorithmically-generated results.

A typical example is users' scoring system. It takes users profile or/and behavior data as input and with a complex scoring engine, it produces a credit score of the users for business decision making. Another example of a data product is a recommendation engine, which ingests user data, and makes personalized recommendations based on that data. Here are some examples of data products:

- Amazon's recommendation engines suggest items for you to buy, determined by their algorithms.
- Netflix recommends movies to you. Spotify recommends music to you.
- Gmail's spam filter is data product – an algorithm behind the scenes processes incoming mail and determines if a message is junk or not.
- Computer vision used for self-driving cars is also data product – machine learning algorithms are able to recognize traffic lights, other cars on the road, pedestrians, etc.

This is different from the "data insights" section above, where the outcome to that is to perhaps provide advice to an executive to make a smarter business decision. In contrast, a data product is technical functionality that encapsulates an algorithm, and is designed to integrate directly into core applications. Respective examples of applications that incorporate data product behind the scenes: Amazon's homepage, Gmail's inbox, and autonomous driving software.

Data scientists play a central role in developing data product. This involves building out algorithms, as well as testing, refinement, and technical deployment into production systems. In this sense, data scientists serve as technical developers, building assets that can be leveraged at wide scale.

0.8 What is Data Scientist?

Data scientists are a new breed of analytical data expert who have the technical skills to solve complex problems – and the curiosity to explore what problems need to be solved. They are part mathematician, part computer scientist and part business trend-spotter. They straddle in both the business and IT worlds with mathematical and programming weaponry.

0.8.1 The Requisite Skill Set

Data scientist needs a blend of skills in three major areas:

1. Mathematics
2. Computing and Software Engineering
3. Business

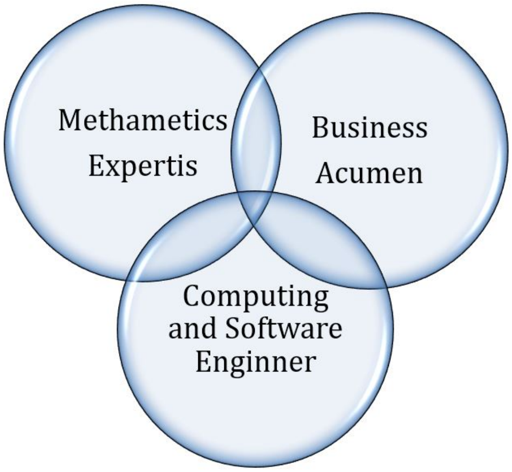


Figure 2: Quality of Data Scientists

0.8.1.1 Mathematics Narrator

At the heart of mining data insight and building data product is the ability to view the data through a quantitative lens. There are textures, dimensions, and correlations in data that can be expressed mathematically. Finding solutions utilizing data becomes a brain teaser of heuristics and quantitative technique. Solutions to many business problems involve building analytic models grounded in the hard math, where being able to understand the underlying mechanics of those models is key to success in building them.

Also, a misconception is that data science all about **statistics**. While statistics is important, it is not the only type of math utilized. First, there are two branches of statistics – classical statistics and Bayesian statistics. When most people refer to stats they are generally referring to classical statistics, but knowledge of both types is helpful. Furthermore, many inferential techniques and machine learning algorithms lean on knowledge of **linear algebra**. For example, a popular method to discover hidden characteristics in a data set is SVD, which is grounded in matrix math and has much less to do with classical stats. Overall, it is helpful for data scientists to have breadth and depth in their knowledge of mathematics.

0.8.1.2 Computing and Software Engineer Skills

Data is now collected, stored and processed with computer. With the increasing of data quantity, as termed as, we are enter the big data era. The conventional way of processing data facing unprecedented challenge. The personal computer may be not adequate to handle big data. Distributed storage, clouds computing and computer clusters become commonly-used platforms for data access and controls. Basic computing environment configuration and settings are common skills need to handle data.

The data processing tools and languages like R or Python, and a database querying language like SQL are the common used languages in data process and data analyzing. It is also important to have a strong software engineering knowledge so it can be comfortable to handle a large amount of data logging, and to develop data-driven products.

Data scientists need utilizing new technology in order to wrangle enormous data sets and work with complex algorithms, and to code or prototype quick solutions, as well as interact and integrate with complex data systems. Core languages associated with data science include SQL, Python, R, and SAS. On the periphery are Java, Scala, Julia, and others. But it is not just knowing language fundamentals. A data scientist is a technical ninja, able to creatively navigate their way through technical challenges in order to make their code work.

Along these lines, a data science is a solid algorithmic thinker, having the ability to break down messy problems and recompose them in ways that are solvable. This is critical because data scientists operate within a lot of algorithmic complexity. They need to have a strong mental comprehension of high-dimensional data and tricky data control flows. Full clarity on how all the pieces come together to form a cohesive solution.

0.8.1.3 Strong Business Acumen

It is important for a data scientist to be a tactical business consultant, an operation narrator and story teller. Working so closely with data, data scientists are positioned to learn from data in ways no one else can. They can understand the language the data speak and listen the story the data tells. That creates the responsibility to translate observations, discovery to shared knowledge, and contribute to strategy on how to solve core business problems. This means a core competency of data science is using data to cogently tell a story. No data present a cohesive narrative of problem and solution, using data insights as supporting pillars, that lead to guidance.

Having this business acumen is just as important as having acumen for technology and math and algorithms. There needs to be clear alignment between data science projects and business goals. Ultimately, the value doesn't come from data, math, and tech itself. It comes from leveraging all of the above to build valuable capabilities and have strong business influence.

0.8.2 How to Become a Data Scientist?

Many people start to Position themselves for a career in data science. Not only for good job opportunities, but also for excitement of work in the technology field with freedom for experimentation and creativity. To get to this position you need solid foundations.

A conventional way of becoming a data scientist is Choosing a university that offers a data science degree. Or register yourself for courses that in data science and analytics fields. If you cannot do these, the option left to you is to learn by yourself.

The knowledge and skills you should have are:

- **Statistics and machine learning**. A good understanding of statistics is vital as a data scientist. You should be familiar with statistical tests, distributions, maximum likelihood estimators, etc. Statistics knowledge will also help you understand when different techniques are (or aren't) a valid approach. Machine learning (ML) is a good weapon when you involve a big data project. Algorithms is the core of machine learning, although many implementations with R or Python libraries do exist and convenient to use, It is still needed a thorough understand how the algorithms works and when when it is appropriate to use different ones.
- **Coding languages such as R or Python**. It is essential, a data scientist is competent with a number of computing and data querying languages like R, Python and SQL.
- **Databases such as MySQL and Postgres**. Data is generally stored in a Database. it is important to have necessary skills for data access

and control from a DBMS systems. The most commonly used DBMS systems are MySql (https://www.mysql.com/) and Postgres (https://www.postgresql.org/) in addition to ACCESS and EXCEL.
- **Visualization and reporting technologies**. Visualizing and communicating data is incredibly important, especially with companies that are making data-driven decisions, or companies where data scientists are viewed as people who help others make data-driven decisions. When it comes to communicating, this means describing your findings, or the way techniques work to audiences, both technical and non-technical. Visualization can be immensely helpful. Therefore familiar with data visualization tools like matplotlib, ggplot, or d3.js. Tableau and dashboarding have become a popular data visualization tools. It is important to not just be familiar with the tools necessary to visualize data, but also the principles behind visually encoding data and communicating information.
- **Big data platforms like Hadoop**.(https://hadoop.apache.org/) and **Spark** (https://spark.apache.org/). Although a lot of Data Science project can be tried, or at least prototyped on PC or workstations, it is reality that most large data analyzing is done on advanced computing platforms like distributed infrastructure or computer clusters. these advanced platform mostly deploy Hadoop ecosystems.

If you don't want to learn these skills on your own, take an online course or enroll in a bootcamp. Like what you do now. It not only provides you opportunity to gain knowledge quickly but also provides you chance of networking with other people who has the similar situation like you do. Connect with other people can lead you into an online community. They all will help you gain fine gran and insider knowledge of solving problems.

0.9 Process of Doing Data Science

Understand what data science is about is just a start of becoming a data scientist. Once the goal is set. The next task is to select a correct path and work hard to to reach your destination. The path is important which can be shorter or longer, or direct and smooth, or curvy and bumpy. It is vital to follow a short and smooth path. This path is the data science project process. Figure **??** is the 6 steps process, which is inspired by the CRISP (Cross Industry Standard Process for Data Mining) (**?**), (**?**) and KDD (knowledge discovery in data bases) process (**?**).

Step 1: Understand the Problem - Define Objectives

Any data analysis must begin with business issues. With business issues a number of questions should be asked. These questions have to be the right questions

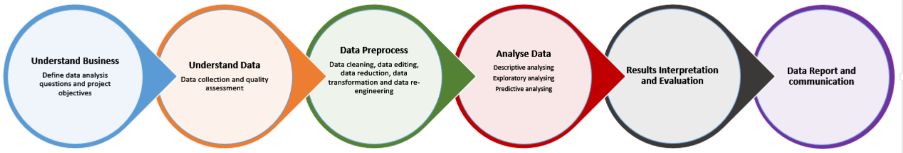

Understand Business
Define data analysis questions and project objectives
Understand Data
Data collection and quality assessment
Data Preprocess
Data cleaning, data editing, data reduction, data transformation and data re-engineering
Analyse Data
Descriptive analysing
Exploratory analysing
Predictive analysing
Results Interpretation and Evaluation
Data Report and communication

Figure 3: Process of doing Data Science

and measurable, clear and concise. Define analysis question is regarded as define a Data Requirements engineering and to get a a data project specification. It starts from a business issue and asking relevant questions, only after you fully understand the problem and the issues you may be able to turn practical problem into analytical questions.

For example, start with a business issue: A contractor is experiencing rising costs and is no longer able to submit competitive contract proposals. One of many questions to solve this business problem might include: Can the company reduce its staff without compromising quality? Or, can company find alternative suppler on the production chain?

Once you have questions, you can start to thinking about data required for analysis. The data required for analysis is based on questions. The data necessary as inputs to the analysis can then be identified (e.g., Staff skills and performance). Specific variables regarding a staff member (e.g., Age and Income) may be specified and obtained. Data type can be defined as numerical or categorical.

After you defined you analytical questions. It is important to Set Clear evaluation of your project to measurement how success of your project.

This generally breaks down into two sub-steps: A) Decide what to measure, and B) Decide how to measure it.

A) What To Measure?

Using the contractor example, consider what kind of data you'd need to answer your key question. In this case, you would need to know the number and cost of current staff and the percentage of time they spend on necessary business functions. This is what is called in business as KPI - Key performance indicators. In answering this question, you likely need to answer many sub-questions (e.g., Are staff currently under-utilized? If so, what process improvements would help?). Finally, in your decision on what to measure, be sure to include any reasonable objections any stakeholders might have (e.g., If staff are reduced, how would the company respond to surges in demand?).

B) How To Measure?

Thinking about how you measure the success fo your data science project, the deep end is to measure some key performance indicators. They are the data you have chosen to use in the previous step. So measure your data is just as important, especially before the data collection phase, because your measuring process either backs up or discredits your project later on. Key questions to ask for this step include:

- What is your time frame? (e.g., annual versus quarterly costs)
- What is your unit of measure? (e.g., USD versus Euro)
- What factors should be included? (e.g., just annual salary versus annual salary plus cost of staff benefits)

Step 2: Undertand Data - Collect Data and Data Validation

The second step is understand data. It includes **Data collection** and **Data Validation**. With problem understood and analytical questions defined and your validation criteria and measurements set, It is time to collect data.

Data Collection

Before collect data, the data source has to be determined based on the relevance. veriaty of data source may be assessed and accessed to get relevant data. These data source may include an existing databases, or organization's file system, or a third party service or even open web sources. They could provide redundant, or complementary, sometimes conflict data. it has to be cautious to select right data source from the very beginning. sometimes you need gather data via observation or interviews, then develop an interview template ahead of time to ensure consistency. it is a good idea to Keep your collected data organized in a log with collection dates and add any source notes as you go (including any data normalization performed). This practice validates your data and any conclusions down the road.

Data Collection is the actual process of gathering data on targeted variables identified as data requirements. The emphasis is on ensuring correct and accurate data collection, which means correct procedure was taken and appropriate measurements were adopted. the maximum efforts were spent to ensure the data quality. Remember that data Collection provides both a baseline to measure and a target to improve for a successful data science project.

Data Validation

Data validation id the process to Assess data quality. It is to ensure the collected data have reached quality requirements identified in the step 1, that is, they are correct and useful. data validation can include:

- Data type validation
- Range and constraint validation
- Code and cross-reference validation
- Structured validation
- Consistency validation

Step 3: Data Preprocess

Data preprocess is step that takes data processing method and technique to transforms raw data into a formatted and understandable form and ready for analyzing. Real world data is often incomplete, inconsistent, and is likely to contain many errors. Data preprocess is a proven method of resolving such issues. Tasks of data preprocess may include:

- **Data cleaning**. The process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database. It normally includes identifying incomplete, incorrect, inaccurate or irrelevant data and then replacing, modifying, or deleting the dirty or coarse data. After cleansing, a data set should be consistent with other data sets.

- **Data editing**. The process involves changing and adjusting of collected data. The purpose is to ensure the quality of the collected data. Data editing should be done by fully understand the data collected and the data requirement specification. Editing data without them can be disastrous.

- **Data reduction**. The process and methods used to reduce data quantity to fit for analyzing. Raw data set collected or selected for analysis can be huge, then it could drastically slow down the analysis process. Reducing the size of the data set without jeopardizing the data analysis results is often desired. It includes records' number reduction and data attributes reduction. Methods used for reduce data records size includes **Sampling** and **Modelings** (e.g., regression or log-linear models or histograms, clusters, etc). Methods used for attributes reduction include **Feature selection** and **Dimension reduction**. Feature selection means removal of irrelevant and redundant features. such operation should not lose information the data set has. Data analysis algorithms work better if the dimensionality, which is the number of attributes in a data object is low. Data compression techniques (e.g., wavelet transforms and principal components analysis), attribute subset selection (e.g., removing irrelevant attributes as discussed in previous paragraph), and attribute construction (e.g., where a small set of more useful attributes is derived from the large numbers of attributes in the original data set) are useful techniques.

- **Data transformation** sometimes referred to as **data munging** or **data wrangling**. It is the process of transforming and mapping data from one data form into another format with the intent of making it more appropriate and valuable for downstream analytics. It is often that data analysis method requires data to be analyzed have certain format or possesses certain attributes. For example, classification algorithms require that the data be in the form of categorical (nominal) attributes; algorithms that find association patterns require that the data be in the form of binary attributes. Thus, it is often necessary to transform a continuous attribute into a categorical attribute, which is called **Discretization**, and both continuous and discrete attributes may need to be transformed into one or more binary attributes, whci iscalled **Banalization**. Other methods include **SCaling** and **normalization**.Scaling changes the bounds of the data, and can be useful, for example, when you are working with data in different units. Normalization scales data sets to a smaller range such as [0.0, 1.0].

- **Data re-engineering** Re-engineering data is necessary when raw data come from many different data sources and in different format. Data re-engineering similar with data transformation can be done in both record level and in attributes level. Record level re-engineering is also called data **Integration**, which integrates variety of data into one file or place and in one format for analysis. for predictive analysis with a model, data re-enginering is also including split a given data set into two subsets called “Training” and “Test” Set.

Step 4: Analyze Data

After your collected data being preprocessed and suitable for analysis. Now you can drill down and attempt to answer your question from Step 1 with the actions called Data Analyzing. It is the core activity in data science project process by writing, executing, and refining computer programs that utilize some analytical methods and algorithms to obtain insights from data sets. The methods in data analysis can be categorized into three major groups: **Descriptive data analysis (DDA)**, **Exploratory data analysis (EDA)** and **Predictive data analysis (PDA)**. DDA and EDA uses quantitative and statistical methods on data sets and data attributes measurements and their value distributions while DDA focus on numeric summary but EDA emphasis on graphical (plot) means. PDA on other hand may involve modeling and machine learning. Data analyzing is generally starting from Descriptive analysis, and goes further with Exploratory analysis. The most advanced methods are predictive analysis and machine learning. The later is built based on the results form the former methods. Some times mixed methods work better.

Descriptive data analysis

It is the simplest type of analysis. It describes and summarizes a data set quantitatively. Descriptive analysis generally starts with an univariate analysis, meaning describing a single variable (can also be called attribute, column or field) of the data. The appropriate depends on the level of measurement. For nominal variables, a frequency table and a listing of the modes are sufficient. For ordinal variables the median can be calculated as a measure of central tendency and the range (and variations of it) as a measure of dispersion. For interval level variables, the arithmetic mean (average) and standard deviation are added to the toolbox and, for ratio level variables, we could add the geometric mean and harmonic mean as measures of central tendency and the coefficient of variation as a measure of dispersion. However, there are many other possible statistics which covers areas such as location ("middle" of the data), dispersion (range or spread of data) and shape of the distribution. Moving up to two variables, descriptive analysis can involve measures of association such as computing a correlation coefficient or covariance. Descriptive analysis' goal is to describe the key features of the sample numerically. It should shed light on the key numbers that summarize distributions within the data, may describe or show the relationships among variables with metrics that describe association, or by tables that cross tabulation counts. Descriptive analysis is typically the first step on the data analysis ladder, which only tries to get a sense of the data.

Explorative data analysis

Descriptive analysis is very important. However, numerical summaries can only get you so far. One problem is that it can only converting a large number of values down to a few summary numbers. Unsurprisingly, different samples with different distributions, shapes, and properties can result in the same summary statistics. This will cause problems. When you are looking a simple single summary statistic, the mean of a single variable, there can be a lot of possible "solutions" or samples. The typical example is Anscombe's quartet (**?**), it comprises four datasets that have nearly identical simple statistical properties, yet appear very different when graphed. Most kinds of statistical calculations rest on assumptions about the behavior of the data. Those assumptions may be false, and then the calculations may be misleading. We ought always to try and check whether the assumptions are reasonably correct; and if they are wrong we ought to be able to perceive in what ways they are wrong. Graphs are very valuable for these purposes.

EDA allows us to challenge or confirm our assumptions about the data. It is a good tool to be used in data prerpocess. We often have pretty good expectations of what unclean data might look like, such as outliers, missing data, and other anomalies, perhaps more so than our expectations of what clean data might look like. With the more we understood data, we could develop our intuition of what factors and possible relations at are play. EDA, with its broad suite of ways to view the data points and relationships, provides us a range of lenses with which to study story that data is telling us. That in turn, helps us to come up with new hypotheses of what might be happening. Further, if we understood which variables we can control, which levers we have to work within a system to drive the metrics such as business revenue or customer conversion in the desired direction. EDA can also highlight gaps in our knowledge and which experiments might make sense to run to fill in those gaps.

The basic tools of EDA are plots, graphs and summary statistics. Generally speaking, it's a method of systematically going through the data, plotting distributions of all variables (using box plots), plotting time series of data, transforming variables, looking at all pairwise relationships between variables using scatterplot matrices, and generating summary statistics for all of them or identifying outliers.

Predictive data analysis

Predictive analysis builds upon **inferential analysis**, which is to learn about relationships among variables from an existing training data set and develop a model that can predict values of attributes for new, incomplete, or future data points. Inferential analysis is a type of analysis that from a dataset sample in hand infer some information, which might be parameters, distributions, or relationships about the broader population from which the sample came. We typically infer metrics about the population from a sample because data collection is too expensive, impractical, or even impossible to obtain all data. The typical process of inferential analysis includes testing hypothesis and deriving estimates. There are a whole slew of approaches and tools in predictive analysis. **Regression** is the broadest family of tools. Within that, however, are a number of variants (lasso, ridge, robust etc.) to deal with different characteristics of the data. Of particular interest and power is **Logistic Regression** that can be used to predict classes. For instance, spam/not spam used to be mostly predicted with a **Naïve Bayes predictor** but nowadays logistic regression is more common. Other techniques and what come under the term **Machine Learning** include neural networks, tree-based approaches such as classification and regression trees, random forests, support vector machines (SVM), and k-nearest neighbours.

Step 5: Results Interpretation and Evaluation

After analyzing your data and get some answers about your original questions, it is possible that you need conduct further research and more analysis. Let us suppose that you are happy with the analysis results you have. It is finally time to interpret your results. As you interpret your analysis, keep in mind that you cannot ever prove a hypothesis true: rather, you can only fail to reject the hypothesis. Meaning that no matter how much data you collect, chance could always interfere with your results. Interpreting the results of analysis, you should thinking of how close the results address the original problems by asking yourself these key questions:

- Does the data answer your original question? How?
- Does the data help you defend against any objections? How?

- Are there any limitation on your conclusions, any angles you haven't considered?

If your interpretation of the data holds up under all of these questions and considerations, then you likely have come to a productive conclusion. However, there could be a chance that you may find you might need to revise your original question or collect more data and you may need to roll the ball from the starting line. Again. Either way, this initial analysis of trends, correlations, variations and outliers are not completely wasted. They help you focus your data analysis on better answering your question and any objections others might have. That is the next step report and communication.

Step 6: Data Report and Communication)

Whereas the analysis phase involves programming and run programs on different computer platforms, the reporting involves narrative the results of analysis, thinking how close the results address the original problems and communicating about the outputs of analyses with interesting parties in many cases in visual formats. During this step, data analysis tools and software are helpful but visual tools are intuitive and worth a lot of words. Visio, tableau (`https://www.tableau.com/`), Minitab (`https://www.minitab.com/`) and Stata are all good software packages for advanced statistical data analysis. There are also plenty of open source data visualization tools available.

It is important to note that the above 6 steps process is not a linear process. Any discovery of useful relationships and valuable patterns are enabled by a set of iterative activities. Iteration can occur in a single step or in a few steps in any point in the process.

0.10 Tools used in Doing a Data Science Project

Data Scientists use traditional statistical methodologies that form the core backbone of Machine Learning algorithms. They also use Deep Learning algorithms to generate robust predictions. Data Scientists use the following tools and programming languages:

R

R (`https://www.r-project.org/`) is a scripting language that is specifically tailored for statistical computing and data. It is widely used for data analysis, statistical modeling, time-series forecasting, clustering etc. R is mostly used for statistical operations. It also possesses the features of an object-oriented programming language. R is an interpreter based language and is widely popular across multiple industries particularly for doing data Science projects.

Python

Like R, Python (https://www.python.org/) is an interpreter based high-level programming language. Python is a versatile language. It is mostly used for Data Science and Software Development. Python has gained popularity due to its ease of use and code readability. As a result, Python is widely used for Data Analysis, Natural Language Processing, and Computer Vision. Python comes with various graphical and statistical packages like Matplotlib, Numpy, SciPy and more advanced packages for Deep Learning such as TensorFlow, PyTorch, Keras etc. For the purpose of data mining, wrangling, visualizations and developing predictive models, we utilize Python. This makes Python a very flexible programming language.

SQL

SQL stands for Structured Query Language. Data Scientists use SQL for managing and querying data stored in databases. Being able to extract data from databases is the first step towards analyzing the data. Relational Databases are a collection of data organized in tables. We use SQL for extracting, managing and manipulating the data. For example, A Data Scientist working in the banking industry uses SQL for extracting information of customers. While Relational Databases use SQL, **NoSQL** is a popular choice for non-relational or distributed databases. Recently NoSQL has been gaining popularity due to its flexible scalability, dynamic design, and open source nature. MongoDB, Redis, and Cassandra are some of the popular NoSQL databases.

Hadoop

Big data is another trending term that deals with management and storage of huge amount of data. Data is either structured or unstructured. A Data Scientist must have a familiarity with complex data and must know tools that regulate the storage of massive datasets. One such tool is Hadoop (https://hadoop.apache.org/). While being open-source software, Hadoop utilizes a distributed storage system using a model called **MapReduce**. There are several other packages in Hadoop together formed a Apache ecosystem, such as Apache Pig, Hive, HBase etc. Due to its ability to process colossal data quickly, its scalable architecture and low-cost deployment, Hadoop has grown to become the most popular software for Big Data.

Tableau

Tableau (https://www.tableau.com/) is a Data Visualization software specializing in graphical analysis of data. It allows its users to create interactive visualizations and dashboards. This makes Tableau an ideal choice for showing various trends and insights of the data in the form of interactable charts such as Treemaps, Histograms, Box plots etc. An important feature of Tableau is its ability to connect with spreadsheets, relational databases, and cloud platforms. This allows Tableau to process data directly, making it easier for the users.

Weka

For Data Scientists looking forward to getting familiar with Machine Learning in action, Weka (`https://www.cs.waikato.ac.nz/ml/weka/`) is, can be, an ideal option. Weka is generally used for Data Mining but also consists of various tools required for Machine Learning operations. It is completely open-source software that uses GUI Interface making it easier for users to interact with, without requiring any line of code.

0.11 Applications of Data Science

Data Science has created a strong foothold in several industries such as Government and education, Healthcare and medicine, banking and commerce, manufacturing and transportation etc. It has immense applications and has variety of uses. Some of the applications of Data Science are listed below:

Data Science in Healthcare

Data Science has been playing a pivotal role in the Healthcare Industry. With the help of classification algorithms, doctors are able to detect cancer and tumors at an early stage using Image Recognition software. Genetic Industries use Data Science for analyzing and classifying patterns of genomic sequences. Various virtual assistants are also helping patients to resolve their physical and mental ailments.

Data Science in E-commerce

Amazon uses a recommendation system that recommends users various products based on their historical purchase. Data Scientists have developed recommendation systems predict user preferences using Machine Learning.

Data Science in Manufacturing

Industrial robots have made taken over mundane and repetitive roles required in the manufacturing unit. These industrial robots are autonomous in nature and use Data Science technologies such as Reinforcement Learning and Image Recognition.

Data Science as Conversational Agents

Amazon's Alexa and Siri by Apple use Speech Recognition to understand users. Data Scientists develop this speech recognition system, that converts human speech into textual data. Also, it uses various Machine Learning algorithms to classify user queries and provide an appropriate response.

Data Science in Transport

Self Driving Cars use autonomous agents that utilize Reinforcement Learning and Detection algorithms. Self-Driving Cars are no longer fiction due to advancements in Data Science.

Summary

While Data Science is a vast subject, being an aggregate of several technologies and disciplines, it is possible to acquire these skills with the right approach. In the end, Data Science is a very robust field that best fits people who have a knack for experimentation and problem-solving. With a large number of applications, Data Science has become the most versatile career.

Excerse

1. Explain Data Science in your own term. What is rela
2.

Get Your Tools Ready

Since this book is “Do Data Science”. It means learn data science by doing. First of all we need to get our weaponry or tools ready.

We already knew that there are a list of tools used by data scientists. Apart from the personal preference, the most used tool is R. This book will use R as the tools to do a complete data science project. However this is not a R language book, it will not teach you about R language and how to use it. It will simply demonstrate a data science project completion step by step, which is completed with R language.

By doing, I mean that you can simply mimic what I have done and follow along by typing or copy past my code into your working space, observe the effects and the results of each line of code execution. Thinking of why I have to do this and what results can I expect along the line of data science project's process. monitoring the issue raised and the methods used to resolve the issues. It is a hope that at some points you can have your own thoughts, perhaps your own code, methods and experiments. Once that is achieved. the goals are reached.

0.12 Brief introductiuon about R and RStudio

R is one of the most widely used programming languages for statistical modeling. It has become the lingua franca of Data Science. Being open-source, R enjoys community support of avid developers who work on releasing new packages, updating R and making it a steady and fast programming package for Data Science.

0.12.1 Features of R Programming

R Programming has the following features:

- R is a comprehensive programming language that provides support for procedural programming involving functions as well as object-oriented programming with generic functions.

- R can be extended easily. There are over 10,000 packages in the repository of R programming. With these packages, one can make use of extended functions to facilitate easier programming.
- Being an interpreter based language, R produces a machine-independent code that is portable in nature. Furthermore, it facilitates easy debugging of the code.
- R supports complex operations with vectors, arrays, data frames as well as other data objects that have varying sizes.
- R can be easily integrated with many other technologies and frameworks like Hadoop and Spark. It can also integrate with other programming languages like C, C++, Python, Java, FORTRAN, and JavaScript.
- R provides robust facilities for data handling and storage. As discussed in the above section, R has extensive community support that provides technical assistance, seminars and several boot camps to get you started with R.
- R is cross-platform compatible. R packages can be installed and used on any OS in any software environment without any changes.

0.12.2 R Scripts

R is the primary statistical programming language for performing modeling and graphical tasks. so it can run in command line as an interpreting languages. However, With its extensive support for performing increasingly complex computations such as manipulations on matrix and dataframes, R is now mostly running in script for a variety of tasks that involve complex datasets with complex operations.

There is plenty of editing tools which perform interactions with the native R console. With any one of them you can edit and run R script. You can also simply import extra packages and use the provided functions to achieve results with minimal number lines of code. There are several editors and IDEs that facilitate GUI features for authoring and executing R scripts. Some of the useful editors that support the R programming language are: RGui (R Graphical User Interface) and RStudio, a integrated R script development environment.

This book will NOT teach you how to code in R. Learning R and to code in R language is not so hard. It just requires a lot of trials and time-spending. You can always going online and searching on Google, Baidu or stackoverflow[1]. There are also plenty of examples and code. The chances are if you're trying to figure out how to do something in R, other people have tried as well, so rather than banging your head against the wall, look online. There are also some books available to help you out on this front as well. I suggest looking other people's code and run it to see the results. R manual is always handy and is available in here[2] .

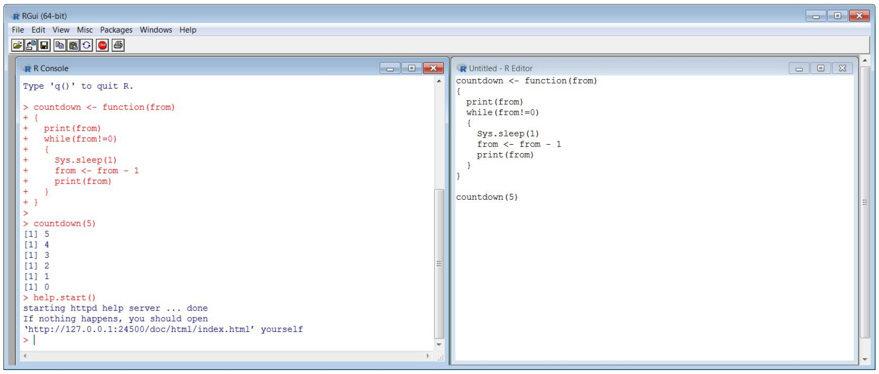
[1] https://stackoverflow.com/

[2] https://cran.r-project.org/manuals.html

If you want learn R systematically, there are many sources online providing good tutorials. You can try to learn more R language from R tutorials. Tutorialspoint (`http://www.tutorialspoint.com/r/index.htm`), codecademy (`https://www.codecademy.com/`). If you prefer an online interactive environment to learn R, this free R tutorial by DataCamp (`https://www.datacamp.com/courses/free-introduction-to-r`) is a great way to get started.

0.12.3 R Graphical User Interface (RGui)

RGui is a standard GUI (Graphic User Interface) platform comes with a R release. By default it provides two windows: R Console (on the left) and R Editor (on the right). See: Figure **??**



```
Type 'q()' to quit R.

> countdown <- function(from)
+ {
+   print(from)
+   while(from!=0)
+   {
+     Sys.sleep(1)
+     from <- from - 1
+     print(from)
+   }
+ }
>
> countdown(5)
[1] 5
[1] 4
[1] 3
[1] 2
[1] 1
[1] 0
> help.start()
starting httpd help server ... done
If nothing happens, you should open
'http://127.0.0.1:24500/doc/html/index.html' yourself
> |
```

```
countdown <- function(from)
{
  print(from)
  while(from!=0)
  {
    Sys.sleep(1)
    from <- from - 1
    print(from)
  }
}

countdown(5)
```

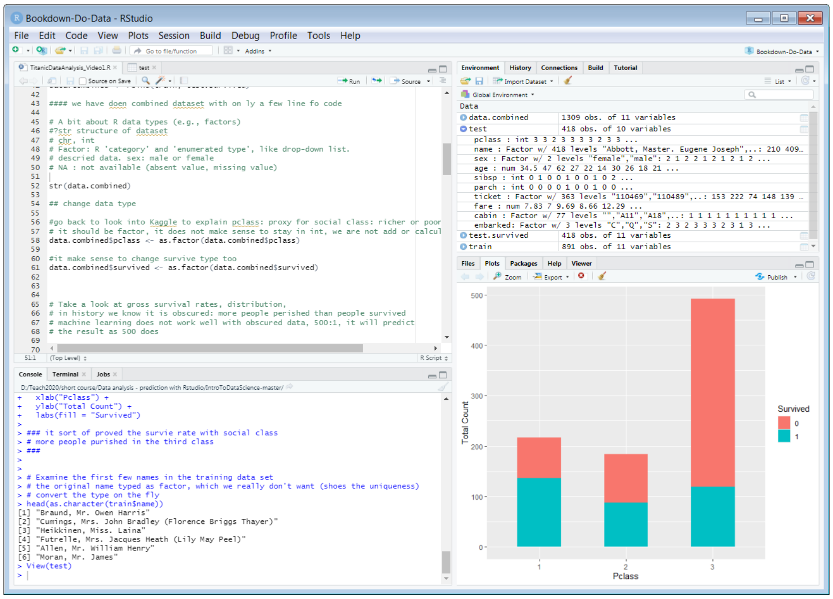
Figure 4: Screen capture of RGui: where Console i son the left and Editor is on teh right

R Console is an essential part of the RGui. In this window, we input various instructions, commands and scripts for different operations. The results of any operation or instruction execution are displayed at the console window including warning and error messages. Console window utilizes several other useful tools embedded to facilitate and ease of various of operations. The console window appears whenever you access the RGui.

R Editor is an simple build-in text editor. Where you can create new R script, edit, test and debug the script and save it into a file. To lunch R Editor, in the main panel of RGui, go to the “File” menu and select the “New Script” option. This will lunch R Editor and allow you create a new script in R. R Editor has a function of “Run line or selection”. It means you can debug your code by line or selection. It is very convenient tool for debugging.

0.12.4 RStudio

RStudio (`https://rstudio.com/products/rstudio/`) is an integrated and comprehensive Integrated Development Environment (IDE) for R. It facilitates extensive code editing, debugging and development. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. Figure **??** is a screen shot of the RStudio.



```
#### we have doen combined dataset with on ly a few line fo code

# A bit about R data types (e.g., factors)
#?str structure of dataset
# chr, int
# Factor: R 'category' and 'enumerated type', like drop-down list.
# descried data. sex: male or female
# NA : not available (absent value, missing value)

str(data.combined)

## change data type

#go back to look into Kaggle to explain pclass: proxy for social class: richer or poor
# it should be factor, it does not make sense to stay in int, we are not add or calcul
data.combined$pclass <- as.factor(data.combined$pclass)

#it make sense to change survive type too
data.combined$survived <- as.factor(data.combined$survived)



# Take a look at gross survival rates, distribution,
# in history we know it is obscured: more people perished than people survived
# machine learning does not work well with obscured data, 500:1, it will predict
# the result as 500 does
```

```
Data
data.combined    1309 obs. of 11 variables
test             418 obs. of 10 variables
   pclass : int 3 3 2 3 3 3 3 2 3 3 ...
   name : Factor w/ 418 levels "Abbott, Master. Eugene Joseph",..: 210 409…
   sex : Factor w/ 2 levels "female","male": 2 1 2 2 1 2 1 2 1 2 ...
   age : num 34.5 47 62 27 22 14 30 26 18 21 ...
   sibsp : int 0 1 0 0 1 0 0 1 0 2 ...
   parch : int 0 0 0 0 1 0 0 1 0 0 ...
   ticket : Factor w/ 363 levels "110469","110489",..: 153 222 74 148 139 …
   fare : num 7.83 7 9.69 8.66 12.29 ...
   cabin : Factor w/ 77 levels "","A11","A18",..: 1 1 1 1 1 1 1 1 1 1 ...
   embarked: Factor w/ 3 levels "C","Q","S": 2 3 2 3 3 3 2 3 1 3 ...
test.survived    418 obs. of 11 variables
train            891 obs. of 11 variables
```

```
D:/Teach2020/short course/Data analysis - prediction with Rstudio/IntroToDataScience-master/
+   xlab("Pclass") +
+   ylab("Total Count") +
+   labs(fill = "Survived")
>
> ### it sort of proved the survie rate with social class
> # more people purished in the third class
> ###
>
>
> # Examine the first few names in the training data set
> # the original name typed as factor, which we really don't want (shoes the uniqueness)
> # convert the type on the fly
> head(as.character(train$name))
[1] "Braund, Mr. Owen Harris"
[2] "Cumings, Mrs. John Bradley (Florence Briggs Thayer)"
[3] "Heikkinen, Miss. Laina"
[4] "Futrelle, Mrs. Jacques Heath (Lily May Peel)"
[5] "Allen, Mr. William Henry"
[6] "Moran, Mr. James"
> View(test)
>
```

Figure 5: Screen capture of RStudio with integrated R code developemtn environment

Here are some distinctive features provided by the RStudio:

- **An IDE that was built just for R**. With Syntax highlighting, code completion, and smart indentation. It can execute R code directly from the source editor. it can quickly jump to function definitions
- **Bring your workflow together**. Integrated R help and documentation with easily manage multiple working directories using projects and Workspace browser and data viewer
- **Powerful authoring & Debugging**. Interactive debugger to diagnose and fix errors quickly and extensive package development tools can authoring with Sweave and R Markdown

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Server Pro.