Defining Data Science

A Case Study in Australia

Xinrui WANG and Tsai-Chun TSOU

28 October 2022

Table of contents

Abstract

Universities are increasingly offering degrees that specialise in the so-called "Data Science" but what is Data Science and what skill are students expected to master in these degrees? The Australian Mathematical Sciences Institute (AMSI) and the Statistical Society of Australia (SSA) are conducting a review of role of of Data Science in Australia universities using surveys and focus groups. Our research attempts to tackle this topic by using data available from public resources. More specifically, we collected unit description and learning objectives/outcomes in Master of Data Science in the Group of Eight (Go8) Universities and job description of data scientist roles from seek.com.au available on from Kaggle. We used the data to decompose the core disciplines involved in the degree as well the type of skill sets that may be required. To expand on the initial exploratory data analysis, we also build Latent Dirichlet Allocation models to construct our own text corpus.

From the exploratory data analysis, we observe a lack of homogeneity of the composition of unit structure across the Go8 universities. The inconsistent data metrics made it difficult to draw direct comparison between the employer data and university data. Nonetheless we were able to conclude that computational components are more prominent in data science for both the university and employer perspectives.

Chapter 1

Introduction

Data Science has ranked as one of the most in-demand jobs in Australia in recent consecutive years. As demands steadily grows, students are also increasingly interested in Data Science degrees, yet recruiters still seem to struggle to fill up data science positions. This leads to our main question: what is Data Science? Is there a shared structure or skill set of Data Science courses offered at Australian universities? Are students and employers' perception of data science similar? To answer these questions, we looked at data from both University and Employer perspectives.

There is no readily available data from Australian universities, so we had to collect our own data set through web scraping. The initial target was to collect data from all universities in Australia including both undergraduate and postgraduate courses, however, due to time constrain, the data collected for this project only contains Master of Data Science courses from the Group of Eight (Go8) universities. The data on job description (which we refer to as "employer data") was retrieved from Data Scientist Job Listings on Kaggle.

By exploring the current situation and potentially a definition of Data Science in Australia from both the university and employer perspectives, the findings would help students and recruiters have a clearer picture of what to expect, as well as raising attentions and awareness to potential gaps between employer demands and university offerings.

The project was conducted in three main phases. The first phase is data collection (Chapter 2 & Chapter 3). Phase two is exploratory data analysis on the university data (Chapter 5) and on employer data (Chapter 6). Details of some pre-processing procedures are included in Chapter 4. Phase three is topic modelling (Chapter 7 & Chapter 8) where we built our own text corpus and group words from the data sets in attempt to find more meaningful results.

Our concluding summary and thoughts on the future direction of this project is in Chapter 9.

Part I

Data Collection

Chapter 2

University Data

2.1 Web Scraping

In order to explore the Data Science degrees in Australian Universities, we compiled a list of universities in Australia and the Data Science or related degrees they offered, then web scraped required information from each university's website using the R programming language (R Core Team 2021). In total, we collected 298 units from eight postgraduate courses in Data Science across all Group of Eight (Go8) universities.

To start off the project, Dr. Tanaka provided sample code for data scraping using Monash Handbook as an example. Libraries `rvest` and `RSelenium` are two of the main tools. Initially, we studied her code and then tried to replicate her code to be applied to other university's websites.

The flow of the data scraping is as follow (example code from `uom-master-datasci.qmd`):

1. Identify the main page (url) where the degree information is contained, which usually is the most updated version of the handbook.

```
remDr$navigate("https://handbook.unimelb.edu.au/2022/courses/mc-datasc/course-structure")
sub_list <- read_html(remDr$getPageSource()[[1]])
```

2. Use functions from `rvest` to retrieve all the course unit code (or course unit url). Retrieve the degree code and formal degree name and save it for later.

```
curriculum <- sub_list %>%
      html_element("#top") %>%
      html_element(".mobile-wrap") %>%
      html_elements("table") %>%
      html_elements("a") %>%
      html_attr("href")
```

3. Use `RSelenium` functions and course unit information, to direct R to the unit information page.

4. Retrieve the following information from the page using rvest functions:

   - Unit Name
   - Unit Code
   - Unit Overview
   - Unit Learning Outcome
   - Unit Prohibition/ Pre-requisite/ Co-requisite

5. Repeat step 3 & 4 with loop function.

```
for(unit in curriculum) {
      remDr$navigate(glue("{baseurl}{unit}"))
      wait_time()
      unit_html <- read_html(remDr$getPageSource()[[1]])

      # unit name
      subject_text <- unit_html %>%
        html_element("h1") %>%
        html_text()
        ...
```

6. Compile all the retrieved data from the University into a single data table and export it as a csv file.

```
data <- data %>%
        bind_rows(tibble(!!!c(list(Course = title,
                                   #Course_code = "MC-DATASC",
                                   Course_overview = paste0(coverview, collapse = " "),
                                   #Unit_code = cunit,
                                   Unit = subject_text,
                                   Overview = overview,
                                   Prerequisite = paste0(pre, collapse = ", "),
                                   Corequisite = co,
                                   Prohibition = paste0(pro, collapse = ", "),
                                   Outcomes =lo
                                   ))))
```

Despite the process being similar for each University, we soon realized the process was going to be more challenging than expected.

2.2 Inconsistent Information

Monash University's student handbook on Degrees and Courses is a spectacular website for data scraping. Its html code is clearly labelled and anything you need to know about the degree or course can be found on the website. The same cannot be said about other universities.

The course descriptions on the handbook and universities' website page are usually structured in a different manner, since the majority of the data is collected from universities' handbooks, course descriptions are also extracted from handbooks for consistency purposes.

In addition, the required unit information listed above is not all available at the targeted universities. The handbook from University of New South Wales contains extremely limited information: unit overview is brief, unit requisites are only available for a few units, and unit outcome is not provided at all.

2.3 Difficulty in webscraping

Each university website is unique. Sometimes the information is not straightforward. An example of this is University of Adelaide's course website. The main website for the degree does contain the list of units that go into the degree.

However, instead of having just one page with all the unit information, the link takes you to a page with different unit information depending on when the unit is offered and on what campus.

Tina tried bypassing the pages by directly looking at the url of the final unit information page I want to be on. Unfortunately, the url is not designed or structured in a way which she was able to predict the url based on the current unit code. With that said, her only option was to code the function to jump from pages to pages before landing on the right unit information page.

It is also often found that the unit overview and learning outcomes for each unit within the same university could vary slightly in format. For example, unit overview may appears before or after campus location at University of Western Australia, empty spaces could be found after section title at the University of Melbourne, which would break the chain of extracting corresponding information.

2.4 Collected Data

The collected data contains **298 units** from 8 universities, and **8 variables** including School, Course, Course_code, Unit, Unit_code, Outcomes, Overview and Description.

Here is an example of the data.

School	Course	Course code	Unit	Unit code	Outcomes	Overview
monash	Master of Data Science	C6004	FIT9132 - Introduction to databases	FIT9132	Apply the theories of the relational database model| Develop a sound relational database design| Implement a relational database based on a sound database design| Manage data that meets user requirements, including queries and transactions| Contrast the differences between non-relational database models and the relational database model.	This unit will introduce the concept of data management in an organisation through relational database technology. Theoretical foundation of relational model, analysis and design, implementation of relational database using SQL will be covered.
monash	Master of Data Science	C6004	FIT9136 - Algorithms and programming foundations in Python	FIT9136	Apply best practice Python programming constructs for solving computational problems| Restructure a computational program into manageable units of modules and classes using the object-oriented methodology| Demonstrate Input/Output strategies in a Python application and apply appropriate testing and exception handling techniques| Investigate useful Python packages for scientific computing and data analysis| Experiment with data manipulation, analysis, and visualisation technique to formulate business insight.	This unit introduces the Python programming and the basics of data structures and algorithms including their design, analysis and implementation in Python. Students will experience working with Python implementation of data structures and algorithms widely used in modern programming language to solve simple problems. Topics covered in this unit are: ..For more content click the Read more button below.

Chapter 3

Employer Data

For employer's perception of Data Science, we decided to look at the job postings for Data Science relevant positions. We would have scraped career websites given more time. However, due to the circumstances, we found readily available data from Exploring 2 years' of Data Scientist Job Listings.

3.1 Data Science Job Postings

The data was scraped from Seek.com by Steve Condylios. The collected data contains **2,857** job posts and **52 variables**.

Exploratory data analysis was conducted exploring the salary and breakdown of Go8 employers. However they did not yield interesting results and thus put aside. For the purpose of this report, only 29 variables were looked at, including jobId, jobTitle, jobClassification, mobileAdTemplate, 25 programming languages. Of all the job posts, 535 are for senior or managerial positions, 25 for graduate positions, and the rest not specified in the job title.

Condylios also included Data Analyst job posting in the data set. 92 jobs are for Data Analyst and 82 jobs are labeled Data Analyst/Data Scientist. This is another interesting topic for comparison between Data Analyst and Data Scientist but for the scope of the project, we do not delve deeper into the data collection decisions.

Part II

Text Analysis

To find out what is data science from universities' and employers' perspectives, whether there is a shared structure or common skill set of Master of Data Science degrees offered at Go8, what employers expect from a data scientist in the workplace, an exploratory data analysis, in particular text analysis has been conducted.

For universities, the main purpose is to identify shared skills or concepts offered by Master of Data Science degrees through exploring faculty of the units, detailed teaching contents from unit overviews and learning outcomes. Whereas for the employer data, the focus is to extract information regarding skills and programming languages in demand.

Chapter 4

Text Pre-Processing

Before any analysis can be done with the text, the target content needs to go through a series of pre-processing. For this project, we relied on functions in libraries `tidytext`, `pluralize`, and `SnowballC` to automate the process. The work flow is as follow:

1. Tokenize raw text into words
2. Remove stop words (eg. “the”, “is”, “a”)
3. Remove any “words” that are simply numbers
4. Singularize or stem the words
5. Limit the number of times words from the same unit or job posting is counted

Tokenization takes the original text and breaks it up into words. Of all the words, many of them will just be stop words which offers no insight. Hence, it is important to remove those stop words. We inspect the words after this step and realized that many of the “words” that were tokenized were actually just numbers. Therefore we filter the data to remove them. If we stop at this stage, we will see that many words are being under counted. For example “student” and “students” will be seen as two different words an counted separately, when in reality they are the same. The `singularize()` function singularizes words to fix this problem. However, we also noticed that this is not enough. Thus we used`wordStem()`to stem the words in attempt to get the root form. This does not necessarily mean to reduce the word into the dictionary root. Instead, we want to stem it only so much to remove the tenses of the word. For example “work”, “works”, and “worked” will al be stemmed to “work.”

Each unit and job description has raw text of varying lengths. If we just take the count of words directly, sometimes a word can be falsely inflated to occur a number of times due to the topic under which it is discussed. Therefore, we only allow the unique words of each unit /job description to included in the final count.

We check the words after each processes and sometimes manually remove words when we find it bring more noise than actual insight. Here is an example code of the pre-processing for unit outcomes.

```
singlewords <- unidata %>%
  unnest_tokens(word, Outcomes)%>%
  filter(!(word1 %in% stop_words$word)) %>%
  subset(!grepl("[0-9]", word1)) %>%
  mutate(word = wordStem(word, language = "english")) %>%
  distinct(School, Unit, word)
```

Chapter 5

Unit Text Analysis

5.1 Faculty Unit Code Analysis

To explore the teaching contents of Master of Data Science at Go8, an analysis based on faculty of units offered is conducted to see what components are included in this degree.

Unfortunately faculty information is not directly available on the unit handbooks, in this case, unit code is taken as a surrogate identification. As shown in the sample data below, unit code is a combination of letters and numbers, the first few characters such as FIT, MAT, usually represents the faculty this unit belongs to, we could then make relatively educated assumptions on the content of the unit.

The grouping was performed manually using the code listed below. We made certain choices though it should be noted that the grouping is not 100% accurate. For example, the code 'DATA' from the University of Sydney is all classified under IT, however, some of the units that start with DATA are taught by the faculty in the School of Mathematics and Statistics (based on personal knowledge), which means 'DATA' belongs to multiple departments. Although there would be misclassified units, the results could still provide a meaningful guidance regarding the teaching components of Master of Data Science at Go8 universities.

School	Course	Unit	Unit_code
monash	Master of Data Science	FIT9132 - Introduction to databases	FIT9132
monash	Master of Data Science	FIT9136 - Algorithms and programming foundations in Python	FIT9136
monash	Master of Data Science	FIT9137 - Introduction to computer architecture and networks	FIT9137
monash	Master of Data Science	MAT9004 - Mathematical foundations for data science and AI	MAT9004
monash	Master of Data Science	FIT5125 - IT research methods	FIT5125

```r
math <- c("STAT", "MATH", "MATHS", "STATS", "MAT", "MAST", "ACTL", "QBUS")
it <- c("COMP", "FIT", "CITS", "INFS", "COSC", "CSSE", "CSYS", "EDPC", "INMT", "PHIL", "P
commerce <- c("ECON", "FINS", "MARK", "ACCT", "FINM", "MGMT", "MKTG")
spatial <- c("GEOM", "ITLS")
science <- c("EDUC", "SCIE", "SOCR")
health <- c("BINF", "BMS", "HTIN", "PUBH")
```

It is clear from Figure **??** that IT and Stat/Math are the two dominating components in the Master of Data Science degrees at Go8. Most units (165 out of 298) fall under the IT faculty, followed by Math and Stats, which has 79 units.

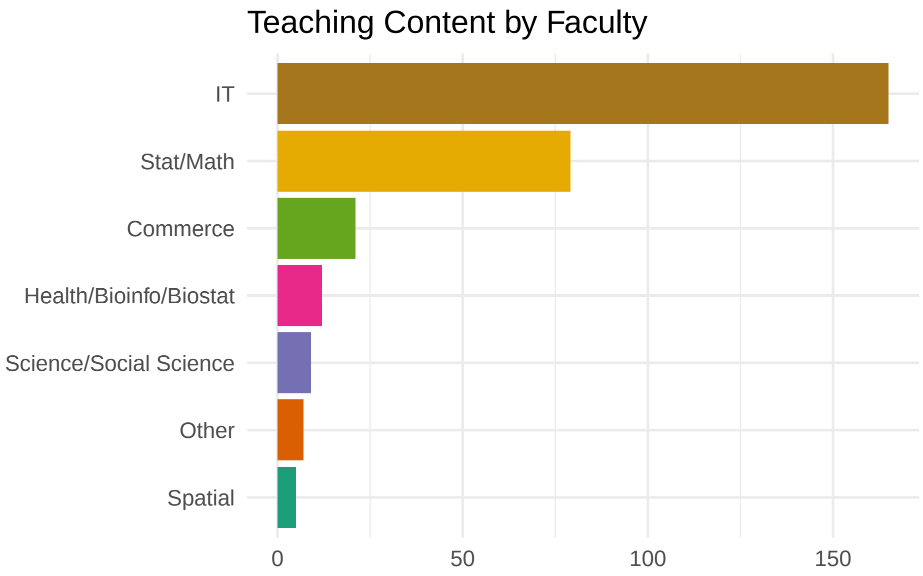


Figure 5.1: Teaching Content by Faculty

Similar findings could be observed from some but not all Go8 universities. Figure **??** shows a heat map of the faculty breakdown by university. Since the total number of units offered by each university is different, instead of showing the actual number, proportions are plotted to make better comparisons across universities.

Lighter colour represents higher proportion, it is obvious that at Monash University (monash), University of Adelaide (Uuade), University of Sydney (usyd) and University of Western Australia (uwa), units from IT faculty occupies more than 50% of the total units offered, especially at Monash University, the proportion of IT units nearly reaches 87%.

On the other hand, University of Melbourne (unimelb) and University of Queensland (uq) offers relatively higher proportion of statistical and mathematical (Stat/Math) units, almost the same percentage as IT units. Whereas units of-

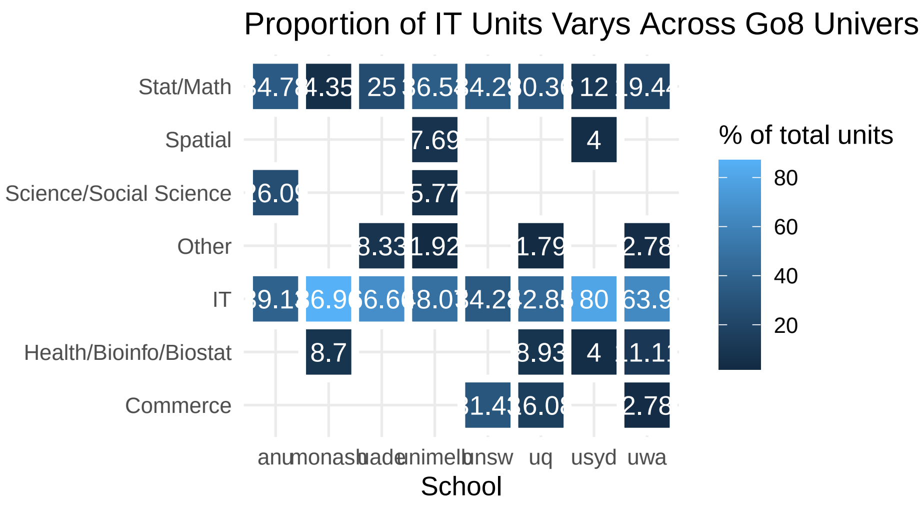


Figure 5.2: Proportion of IT Units Varys Across Go8 Universities

fered at the Australian National University (anu) and UNSW Sydney (unsw) are more evenly distributed across IT, Stat/Math, Science/Social Science and Commerce respectively.

In addition, it is also clear that units offered at University of Melbourne (unimelb), University of Queensland (uq) and University of Western Australia (uwa) covers five out of eight categories, which implies the Master of Data Science degrees at these three universities provide more varieties in terms of units offered.

Based on the findings above, it seems that there is a shared structure across Go8 that Master of Data Science is a IT based, computational degree, but the proportion it occupies varies by universities. Monash University tends to be heavily focused on IT and computational aspects, whereas the Master of Data Science degree at UNSW Sydney and ANU are more balanced across IT, statistics and math, as well as science and commerce.

5.2 Unit Overview and Learning Outcome - Bigram

After having a rough idea of the bigger picture, we then moved to explore what exactly are the teaching contents. We pre-processed text from learning outcome and unit overview to produce single word analysis, bigram, and tirgram. Words and terms such as ‘student’, ‘successful completion’ add more noises than values to the results, are removed in the pre-processing step.

The bigram, shown in Figure **??** below provides the most informative results among the n-gram analysis. Machine learning (machin learn) appears quite often, as well as software development (softwar develop), linear model, statistical analysis (statist analysi), spatial data. It seems that these frequently mentioned terms are associated with both computational and statistical concepts and skills, which aligns with the findings from the unit code analysis in previous section.

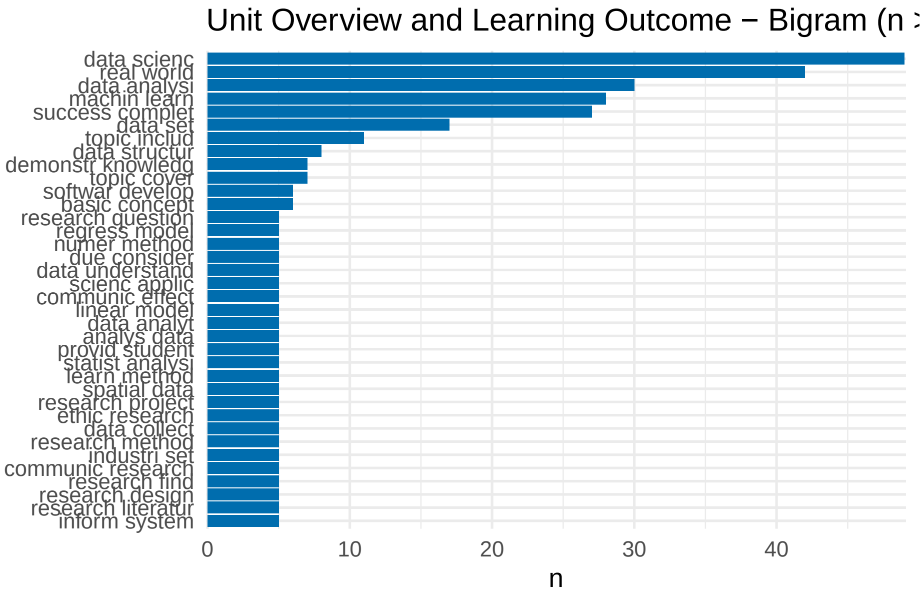


Figure 5.3: Unit Overview and Learning Outcome - Bigram (n > 4)

Unfortunately, due to the limited number of observations in the collected data set, the count for each term is too low to make meaningful interpretations or justifications. In addition, similar terms such as research findings, research designs and research literature are supposed to be grouped and counted together, but are not in the bigram. This issue is later solved by introducing the topic modeling technique for natural language processing, which will be discussed in Topic Modelling section.

Chapter 6

Job Text Analysis

To explore the skills and programming languages in demand from employers, we focused on the mobileAdTemplate and the 25 columns of programming languages. The variable mobileAdtemplate contains the job description for the position. Using this variable, we conducted text analysis and produced plots for single word frequencies, bigram, and trigram.

6.1 Word Frequency

Word frequency was calculated using the same method as Section **??**. Programming languages Python and SQL seems prominent. Potentially due to the amount of senior positions in the data, *experience* is mentioned a lot. In terms of other knowledge or skills, *statistics*, *modelling*, *analysis* are some terms that seems to be standing out. To ensure frequency is meaningful, we looked at bigram and trigram.

6.2 Bigram

For Figure **??**, we stemmed the words before joining them into bigram in attempt to avoid under-counting. However, from the figure we can still see that some terms like *data analyt* and *data analysi* are still counted separately. From this bigram, the popular skills or knowledge mentioned are *machin learn*, *predict model*, *communic skill*, *data analyt* and *data mine*. Mathematical skill, *mathemat statist* is also mentioned quite often. Some of these terms on the list are vague and can mean be grouped together. Trigram yielded similar results as the bigram. With the same problem of under-counting the n-grams due to insufficient groupings.

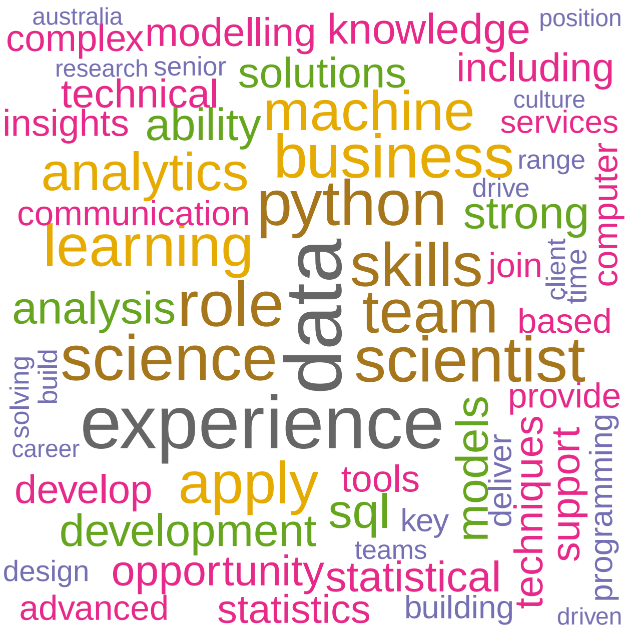


Figure 6.1: Job Word Frequency (freq >200)

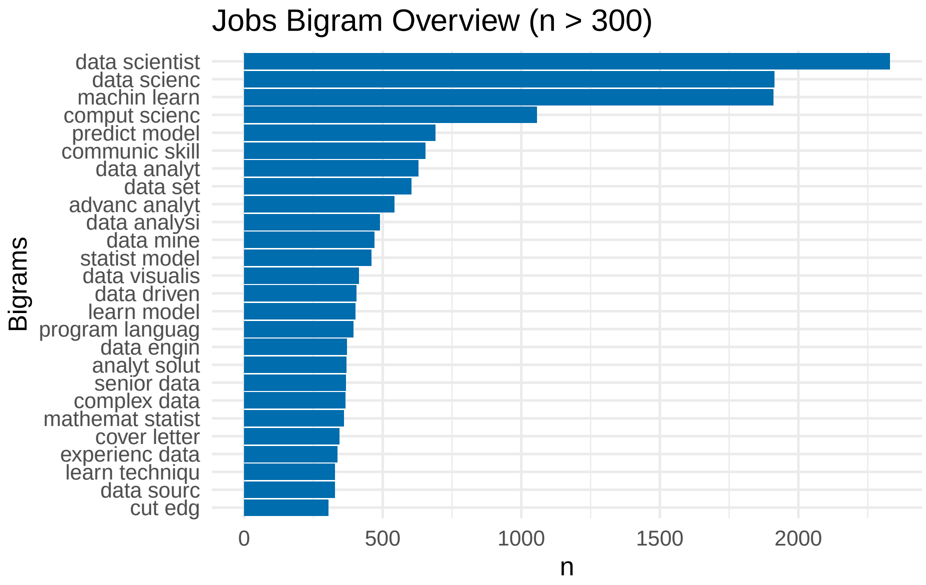


Figure 6.2: Job Description - Bigram (n > 300

6.3 Programming Languages

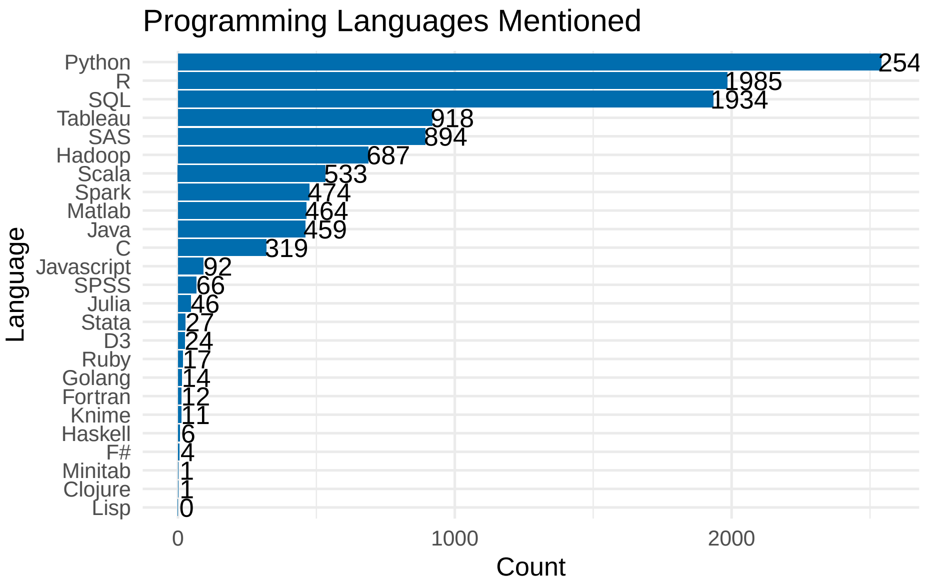


Figure 6.3: Programming Skills Mentioned by Employers

Figure **??** shows the count of each language mentioned by employers. 89% of the jobs mentioned Python, 69% mentioned R and 68% mentioned SQL. Tableau's popularity is not a surprise given the high frequency for *data visualis* in the previous section.

Part III

Topic Modelling

As mentioned in Section Section **??**, since the collected university data is relatively small, to make more educated and meaningful interpretations, similar words shall be grouped together and counted by groups. This is usually computed using text corpus, which is a language resource consisting of a large and structured set of texts, since data science is a new term waiting to be defined, there is no available text corpus on this topic. Therefore, we adopted the concept of word embedding models and tried to build our own text corpus.

There are multiple publicly available models and packages to conduct similar computations, such as `word2vec` and `text2vec`, however, each model takes hours to fit. Due to time constrains, we have only fitted the Dirichlet Allocation (LDA) model with a few parameter adjustments using the `text2vec` package with the concepts illustrated by Das (2016).

Algorithm and Model Fitting

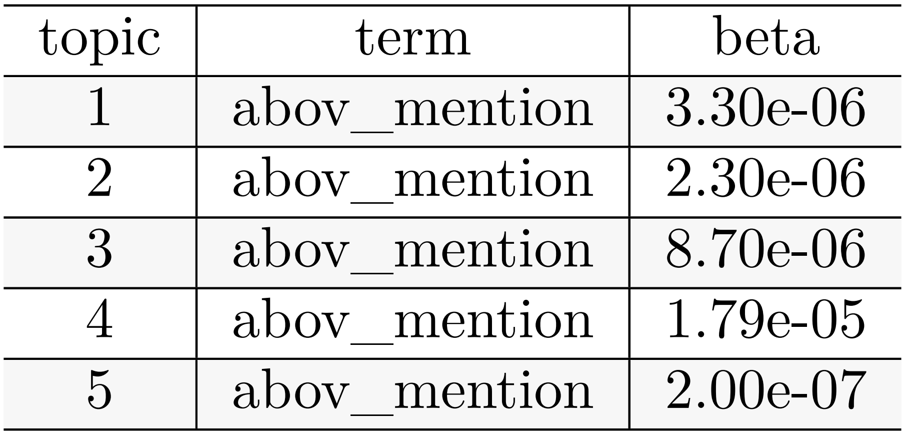
According to Das (2016), the algorithm behind the LDA model is to convert words to document-term matrix (DTM), where the rows, columns and entries correspond to documents, terms and counts respectively. LDA then fits a probabilistic model that assumes a mixture of latent topics, where each topic has a multinomial distribution for all words. The number of topics is a parameter that could be adjusted by needs.

The model must be trained before it could be used, we web scraped 4448 Wikipedia articles as training data, including 2816 articles in statistics, 1005 articles in sociology and 627 in computing. The initial codes and functions to build the LDA model was provided by Dr Tanaka, we have tested model outputs using different number of topics and tried out training the LDA models with different combinations of data.

The table below shows a glimpse of the model output of the initial LDA model, after training by data collected from all 4448 Wikipedia articles on 12 topics.

Term shows all the word extracted from the training data (Wikipedia articles), as each latent topic has a multinomial distribution for all words, beta value of a term is the score / probability computed for that particular topic. Highest beta value indicates highest probability, which means the term is most likely belongs to the corresponding topic.

Beta values for the same word would differ from model to model, and also subject to change by adjusting the number of latent topics. To acquire the most satisfactory results for our university data, we have fitted and tested multiple LDA models with different subset of training data, as well as various number of topics.



topic	term	beta
1	abov_mention	3.30e-06
2	abov_mention	2.30e-06
3	abov_mention	8.70e-06
4	abov_mention	1.79e-05
5	abov_mention	2.00e-07

Model Adjustments

After applying the fitted LDA models to our university data set, the results delivered by the models are quite different. Figure **??** compares the results produced by the four fitted models using different training data on ten topics.

For Model A, Topics A9, A10, A2 and A8 occupies relatively higher proportion compare with the others, but the order varies across universities, and their proportions are not significantly larger than the rest of other topics, makes it hard to draw meaningful interpretations. On the top right, Model B demonstrates a better picture: Topics B6, B8 and B9 are the top 3 topics across all Go8 universities, however, proportions of Topics B10, B5, B7, B3 and B4 are also obvious higher in some of the universities, brings in difficulties to make justifications.

As sociology data tends to brings in noises to the model, and is not closely relevant to the data science topic compare with statistics and computing, Model C and D are fitted using only statistics data and computing data respectively. Topic C2 is the only dominating topic in Model C, where as Topics D6 and D1 occupy significantly large proportion in Model D. Besides, cccccc together took a relatively higher proportion compare with the rest of other topics in Model D.

The table below listed the top 30 words of each topics in Model C, it turns out Topic C2 contains words like comput (computation, computational, computer), system, program, machin (machine), softwar (software), model, test, calcul (calculate, calculation) and data, which seems to be associated with mainly computational aspects.

Although Model D provides a reasonably meaningful result, there is not much interpretations could be made for the other topics, the information it offers is still not very satisfying. Compares with Model C, Model D has two domination topics D6 and D1, topics D5, D10, D2, D9 also accounts for a large proportion, which together provides more information. Therefore, Model D, which is trained by only statistics data on ten topics, is selected to use for further analysis on

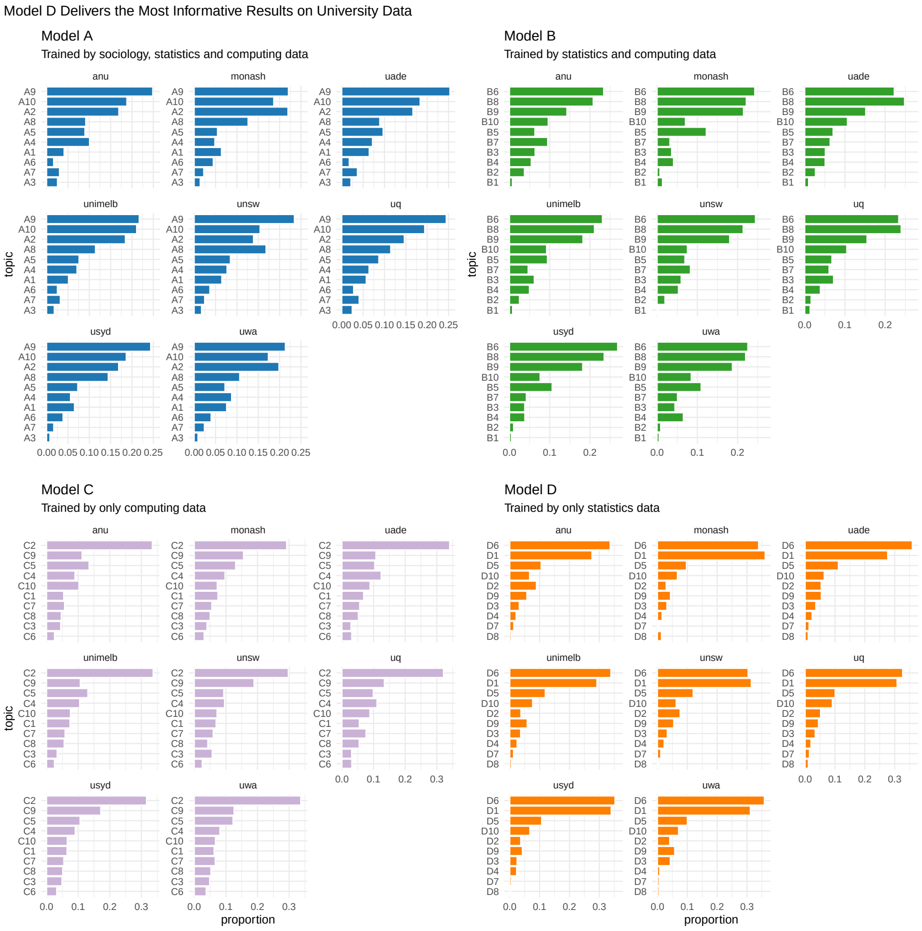


Figure 6.4: Model D Delivers the Most Informative Results on University Data

Topic C1	Topic C2	Topic C3	Topic C4	Topic C5	Topic C6	Topic C7	Topic C8
window	comput	ibm	algorithm	network	bit	format	intel
system	system	comput	can	use	instruct	use	chip
version	program	system	number	can	memori	imag	design
releas	use	disk	function	web	use	digit	processor
support	machin	drive	set	data	address	can	bit
user	design	use	state	internet	regist	video	core
oper_system	process	control	languag	secur	processor	standard	mhz
oper	develop	machin	use	access	oper	disc	use
os	inform	card	symbol	protocol	can	data	introduc
file	softwar	unit	problem	link	data	file	microprocessor
microsoft	engin	amiga	grammar	connect	system	encod	technolog
use	first	pc	rule	server	page	dvd	motorola
applic	time	model	parser	servic	mode	edit	clock
includ	research	storag	recurs	standard	architectur	pdf	power
develop	can	data	one	devic	also	code	ghz
unix	logic	commodor	input	node	set	cd	bus
mac	scienc	home	exampl	communic	one	camera	cpu
command	work	hardwar	machin	provid	code	audio	generat
interfac	model	game	ture	attack	access	compress	perform
also	oper	one	comput	may	byte	ray	support
dos	perform	oper	two	key	onli	also	seri
server	one	time	follow	also	program	includ	base
featur	test	atari	ani	client	comput	blu	product
appl	calcul	ii	pars	system	execut	blu_ray	model
run	data	tape	context	browser	cpu	graphic	first
base	univers	graphic	time	user	store	allow	cach
shell	mani	market	defin	page	point	print	market
linux	human	product	express	messag	number	media	two
manag	electron	display	string	applic	unit	store	famili
new	method	cp	onli	ethernet	perform	record	mb

our university data.

Note that it requires highly skilled linguists and huge efforts to establish a proper text corpus, the model we built is still fairly basic and could be further optimised by adjustments.