DATA SCIENCE JOBS IN AUSTRALIA

DATA7001 - GROUP 0

Ce Hou s4768730
Jisung Kim s4582833
Jaskeerat Singh s4761003
Yingqi Zhang s4658997

Executive Summary

The report presents Group O's research on data science jobs in Australia. Specifically, to understand which city is the best to live in as a data scientist and how many skills and experience are needed. Key stakeholders include data science students, job seekers, businesses, and education sectors.

The structure of the report follows the five steps of the data science process. The first part outlines the purpose and process of the research. The second part lists the datasets used for analysis. The third part identifies the issues within the dataset and how they have been resolved. The fourth part explores the datasets through visualisation to understand and gain useful insights about the data. The last part presents the results of the analysis through simple charts and map illustrations. Additionally, the appendix contains the source code written for the previous steps.

The research covered many aspects of the topic that the stakeholders might be very interested in. Gaining knowledge on the relationship between salary and cost of living, most in-demand skills and experience, and conditions for salary increase. The best city to live in is either Melbourne or Sydney. Skills and experience are recognised and offered with a higher salary. Furthermore, the only economic factor that has been applied to determine the best city is the Consumer Price Index, so further research will be required for a better determination in the future.

Table of Contents

1. Problem solving with data

1.1 Introduction

Data science jobs refer to the scientific analysis, mining, and presentation of big data based on various analytical tools to assist enterprises in making business decisions.Data science jobs cover various directions such as data development engineers, data analysts, data architects, data backend development engineers, and algorithm engineers for big data.Jobs related to data analytics require a combination of people who can have a comprehensive knowledge of mathematics, statistics, data analysis, machine learning and natural language processing.

From our report, the average salary for a data analytics job in Australia now ranges from A$1952-3092 per week. Demand for data science jobs continues to thrive and expand across all industries.The middle and high end of the data analytics talent pool will be in high demand.
And in the future, more people will choose to be employed in data analytics positions, and their salaries will improve. The big data industry has become one of the paths for more and more people to achieve their dreams of high salaries in the workplace.

In this background, our group conducted a study of data analysis work in Australia. This report details our targets, the issues to be addressed, the acquisition of data, the analysis process, and the results of the analysis. It is very informative for our stakeholders.

1.2 Research Steps

This analysis follows the five steps of the data science process. With this process(*Figure 1.1*), our research aims to address the following three questions.

- Where is the best place to do data science work in Australia?
- What are the skills needed to do data science jobs?
- What are the job prospects for data science jobs in Australia?

Figure 1.1: Five steps of the data science process

1.3 Stakeholders

Stakeholder	Interests
Data Science / Analysis Students	They can understand the job skills needed and the job trend happening in Australia.
Job Seekers	They can use our analysis report and then combined with their own needs, targeted to find their own job search direction.
Businesses	They can have a greater advantage in recruitment than similar companies based on our analysis report.
Education	Their Data Science programs can be offered in a way that is more in line with students' future development. They can acquire more sources of students while improving teaching level.

Table 1.1: Key stakeholders

Data science jobs are available in a wide range of fields, including finance, insurance, the Internet, healthcare, education, government, and more. Our relevant stakeholders are also in these fields. They can benefit from our analysis reports.

2. Getting the data we need

2.1 Data Sources

1. **Data of every Australian data science job listing from January 2019 to January 2022 on Seek.**
   This dataset contains 52 columns and 3902 rows.

2. **Data of every Australian data science job listing in August 2022 on Glassdoor.**
   This dataset contains 53 columns and 2088 rows.

3. **Data of Consumer Price Index (CPI) for every major city in Australia from 2018 to 2022.**
   This dataset contains 10 columns and 23 rows.

For the research datasets, we downloaded Dataset 1 and Dataset 2 from Kaggle. From the Data of every Australian data science job listing, we obtained data science job titles, skill requirements, earnings, and job locations, among others. After data cleaning, we filtered the data related to the research.

We downloaded the Data of Consumer Price Index (CPI) for every major city in Australia from the Australian Bureau of Statistics, through which we analysed the standard of living and consumption in major cities in Australia.

3. Is the data fit for use?

3.1 Data Quality

3.1.1 Inconsistency

After the data has been collected, we have to check and ensure that our Data is fit for use. After loading the data using python, we can see that our data is present in an inconsistent and complex format for some columns like salary, job title etc.

jobTitle
Senior Method Development Scientist (Immunoassay)
Data Engineer
Data Modeller (Erwin) Financial Services
Data Scientist - Big Data
Client Side Data Scientist - Melbourne CBD - US Tech startup
Analyst Developer
Materials Development Scientist
Senior Data Scientist - Customer/Marketing Analytics
Research Scientist - Materials Science/Engineering
Environmental or Agricultural Scientist/ Engineer
Graduate Environmental Scientist/ Engineer

salary_string
Super
$90000 - $120000 per annum
$90k - $110k p.a.
$110k - $120k p.a. + super + bonus + benefits
Attractive Salary Package
$71,509 - $90,215 per plus superannuation
$71,509 - $90,215 per plus superannuation
$134421 - $148725 p.a. + plus up to 15.4% super
$134421 - $148725 p.a. + plus up to 15.4% super
$120,000 - $159,999
Competitive remuneration
$110k Package + Benefits

mobileAdTemplate
My client is a leading Australian owned and rapidly growing full service CRO bas
Leading financial services organisation is hiring a Data Scientist to join their risk
The Company:A big brand with a reputation forÂ leadingÂ theÂ wayÂ with Big D
Responsibilities: Design and development of scorecards and models across the
â€ÇÂ Large, high profile organisationâ€ÇÂ Unique Projectâ€ÇÂ Senior roleThe R
â€ÇÂ Large, high profile organisationâ€ÇÂ Unique Projectâ€ÇÂ Senior roleThe R
A fantastic opportunity for you to showcase your passion for Data, Technology
One of Australia's leading financial service providers is seeking a Data Scientist t
A leading Australian financial institution is seeking a Senior Data Scientist to play
About the business and the role My client is a innovative and fast growing comp

Figure 3.1: Data in complex format

To get some useful insights from the data, we need to apply some pre-processing techniques to get data in a clean format. At the start, we remove duplicate and irrelevant rows of tuples from the dataset.

Regex is used to get cleaned data and create or replace columns by processing the already available data.

jobType	salary	total_skills	experience
Data Scientist	75341	2	3.18
Data Analyst	130038	2	4.75
Data Analyst	134819	3	4.75
Other	73539	1	3.18
Other	73539	1	3.18
Data Scientist	82553	6	3.18
Other	158722	8	4.75
Data Engineer	333009	6	5.01
Analytics Manager	111566	6	4.75
Data Engineer	125257	1	4.75

Figure 3.2: Data in a cleaned format

Python code:

```
#Calculating salary from salarystring
jobs_data["salary"] = None

for i in range(len(jobs_data)):
    value = str(jobs_data["salary_string"][i])

    remove_list = [',', 'per day', 'per annum', 'p.a.', 'p.d.', 'p/d', 'p/a', '.00']
    for x in remove_list:
        value = value.replace(x, "")

    value = re.sub('\d+[\.]*\d*[%]+', '', value)
    value = re.sub('[\.]*\d*[%]+', '', value)
    value = re.sub('[\.]+\d*', '', value)
    value = re.findall(r'[-]*\s*\d+\s*\w{1}', value)

    try:
        jobs_data["salary"][i] = re.sub('\D', '', value[-1].replace("k", "000").replace("K", "000"))
    except:
        jobs_data["salary"][i] = None
```

```python
#Parsing experience from online ads
jobs_data["experience"] = None

for i in range(len(jobs_data)):
    exp = None
    try:
        exp = list(filter(lambda sentence: ("year" in sentence)or("yr" in sentence)
        exp = re.findall('\d+\s*[+]*\s*[y]{1}', exp)[0]
        exp = re.sub('\D', '', exp)
    except:
        exp = None
    jobs_data["experience"][i] = exp
jobs_data["experience"] = jobs_data["experience"].astype('float64')
```

Figure 3.3: Python data to parse salary and experience

3.2 Missing Data

3.2.1 Linear Regression Imputation

After the Data has been cleaned, we move on to impute values for the missing values in numerical columns. For imputation, we fill the missing values with the mean of the column. For the salary column, we will develop and fit a linear model on total_skills, experience, city to predict the value of missing values for salaries. Here a different model is developed for each different location. This is done to not weaken the relationship between salary and location.

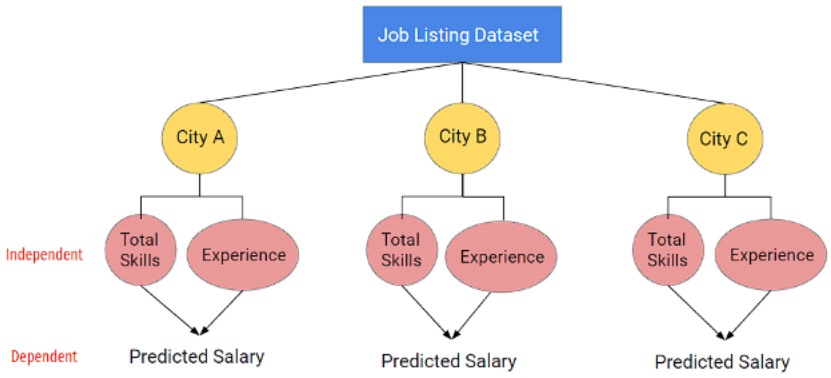

Figure 3.4: Linear Model to predict salary

A linear regression model describes the relationship between a *dependent variable or a response variable* , *y*, and one or more *independent variables*, *X*. Independent variables are also called *predictor variables*. Continuous predictor variables are also called *covariates*, and categorical predictor variables are also called *factors*.

A multiple linear regression model is defined by : $y_i = \beta_0+\beta_1 X_{i1}+\beta_2 X_{i2}+\cdots+\beta_p X_{ip}+\varepsilon_i, \quad i=1,\cdots,n$

```python
#Imputing values based on a trained Linear regression model
lrm_data = jobs_data[["state", "salary", "total_skills", "experience"]]

for i in lrm_data["state"].unique():
    lrm_train = lrm_data.loc[lrm_data['salary'].notnull(), ][lrm_data['state'] == i]

    lrm_train["salary"] = lrm_train["salary"].astype(int)
    lrm_train["salary"] = lrm_train["salary"].abs()

    lrm = linear_model.LinearRegression()
    independant = lrm_train[["total_skills", "experience"]]
    dependant = lrm_train["salary"]
    lrm.fit(independant, dependant)
    print('Model score : ', lrm.score(independant, dependant))

    for j in jobs_data[jobs_data.isnull().any(axis=1)][jobs_data["state"] == i].index:
        jobs_data["salary"][j] = round(lrm.predict([[jobs_data["total_skills"][j], jobs_data["experience"][j]]])[0], 2)

jobs_data["salary"] = jobs_data["salary"].astype(int)
```

Figure 3.5: python Linear Model to predict salary

```python
    print("R2 Score:", r2_score(y_pred,y_test))
    print("Mean Square Error:", mean_squared_error(y_test,y_pred))
    print("Root Mean Square Error:", np.sqrt(mean_squared_error(y_test,y_pred)))
    print("Mean Absolute Error:", mean_absolute_error(y_test,y_pred))
```

```
###################################### Australian Capital Territory ###################################
R2 Score: 0.9825576808531455
Mean Square Error: 15.74969218475894
Root Mean Square Error: 3.9685881853322775
Mean Absolute Error: 292.7602575387997
###################################### New South Wales #####################################
R2 Score: 0.6183768502423425
Mean Square Error: 254.25047438629804
Root Mean Square Error: 15.945233594598044
Mean Absolute Error: 923.4581212304639
###################################### Victoria ########################################
R2 Score: 0.8273713391415016
Mean Square Error: 26079.823816167176
Root Mean Square Error: 161.49248842025804
Mean Absolute Error: 6119.8420525888405
###################################### Queensland ######################################
R2 Score: 0.886630592237577
Mean Square Error: 98.57435239749299
Root Mean Square Error: 9.92846173369737
Mean Absolute Error: 1714.0253453644145
###################################### South Australia #####################################
R2 Score: 0.9999999997645055
Mean Square Error: 6.586672447767373e-07
Root Mean Square Error: 0.0008115831718171203
Mean Absolute Error: 0.1863716686718059
```

Figure 3.6: Model Statistics

3.2.2 Outliers

After the data has been imputed, outlier detection is used to remove outliers to get consistent data which will be used to build visualisations and gain insights. For outlier detection, quantile method is used, where the data between the range of max and min calculated using inter quartile range is only kept. For detection we have assumed that the data is normally distributed. The rule of data falling between the upper and lower limit of the box plot is considered, where every value outside of this range is removed from the data set.

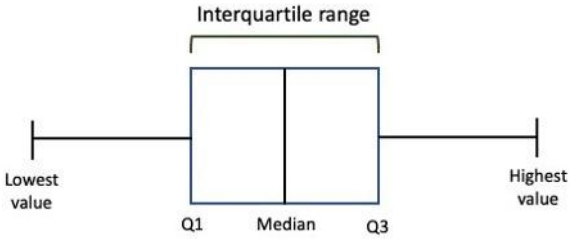

Figure 3.7: Boxplot definitions

Point	Definition
Q3	75th percentile
Q2	Middle Value (Median)
Q1	25th percentile
IQR	Q3 – Q1
Highest	Q1 + 1.5 x IQR (UPPER Whisker)
Lowest	Q1 - 1.5 x IQR (LOWER Whisker)
OUTLIERS	Value lying outside Maximum or Minimum

```
jobs_data["outlier"] = 0

for col in ["salary", "total_skills", "experience"]:

    q3, q1 = np.percentile(jobs_data[col], [75 ,25])
    iqr = q3 - q1

    upper = q3 + (1.5*iqr)
    lower = q1 - (1.5*iqr)
    jobs_data.loc[((jobs_data[col] < lower) | (jobs_data[col] > upper)), "outlier"] = 1

jobs_data = jobs_data[jobs_data["outlier"] == 0]
del jobs_data["outlier"]
```

Figure 3.8: Python code to remove outliers

Here, we will implement the quartile method to remove outliers from salary, total_skills and experience columns. Only the values between 25th and 75th percentile are kept; other values are removed from the dataset.

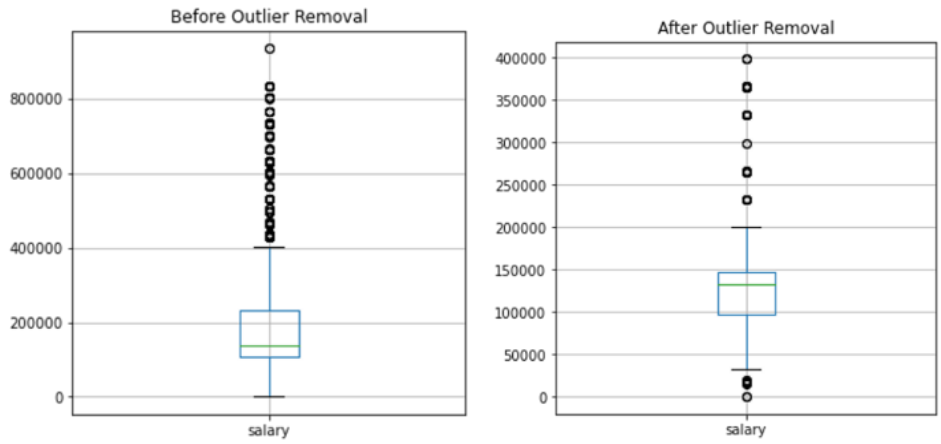

Figure 3.9: Box Plot for Before and After outlier Removal

4. Making the data confess

4.1 Exploratory Data Analysis (EDA)

Exploratory data analysis is conducted to identify patterns and anomalies and come up with hypotheses with the help of visual representations of data. The visualisations are produced with R and Tableau.

4.1.1 Number of Data Science Jobs

The number of data science jobs for every capital city in Australia is plotted in a line graph to show the trend. Sydney takes the lead with the highest count, then Melbourne takes second place. The rest are gathered significantly below the two cities, meaning there are fewer opportunities there. Most interestingly, there was a huge drop in numbers in the second quarter of 2020. This is no doubt the impact of COVID-19 on the Australian economy. However, on the bright side, the numbers can be seen increasing as pandemic restrictions are eased. We can assume that job listings will increase in the future.

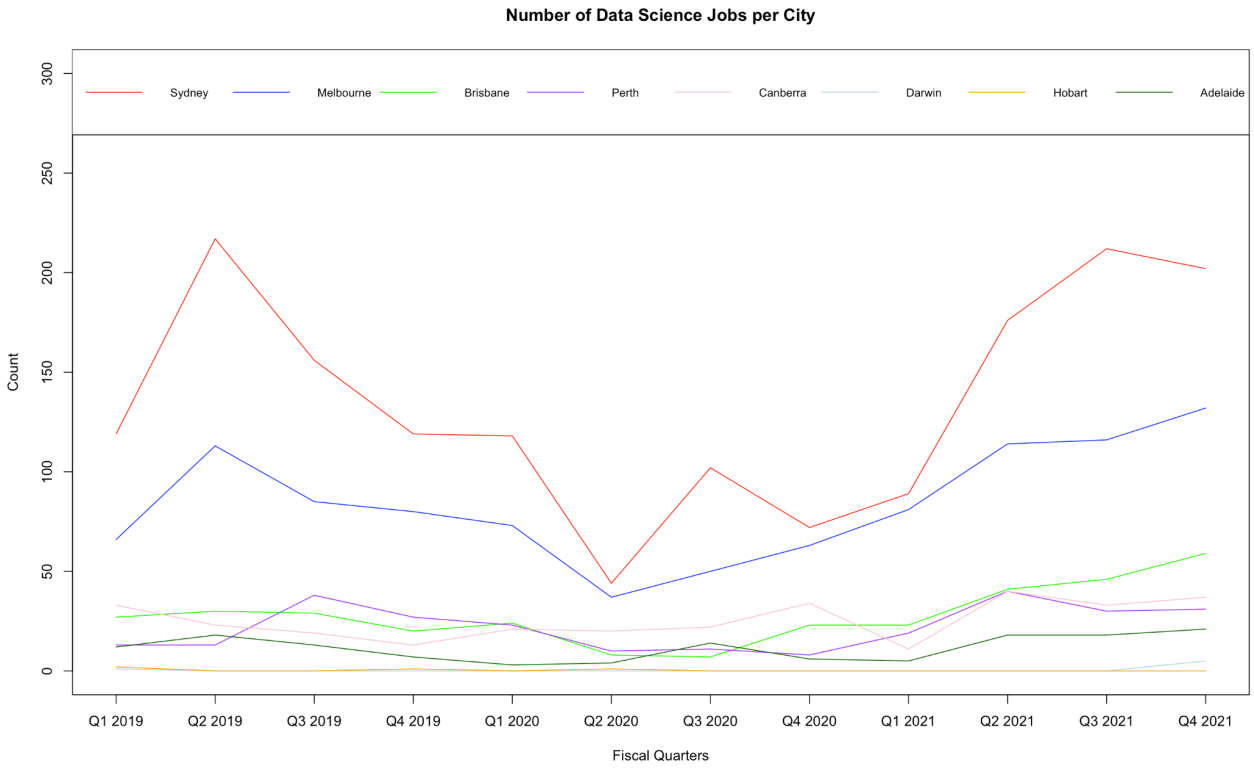

Figure 4.1: Number of Data Science Jobs Per City

4.1.2 Changes in Consumer Price Index

The change in the Consumer Price Index (CPI), often referred to as the cost of living, is presented in the line graph. Every capital city shows an increasing trend throughout the four years, but the inflation seems to get stronger in 2021 and onwards, which is a threat to a healthy economy. This is due to the current rising transport costs, new dwellings, and education. In the second quarter of 2020, a huge drop in CPI for every city is clearly visible like the number of data science jobs. This is due to COVID-19, and was incredibly effective in decreasing international travel, accommodation, and automotive fuel. Furthermore, Darwin showed a relatively slower growth in CPI. We can assume that the cost of living will continue rising rapidly for every city.

Figure 4.2: CPI changes in eight capital cities in Australia

4.2.3 Change in Salary

This is the average data science salary trend plotted as a line graph. To effectively see the trend, the line turns red during negative growth and green during positive growth. Similar to the previous graphs, in 2020, an unusual decrease in salary can be seen. Again, COVID-19 is believed to be the reason behind this notable drop in average salary. Moreover, salaries started to show a positive movement after recovering from the pandemic. Interestingly, the average salary remained at six-figures even though it was at its lowest, meaning the jobs are highly valued. The average salary is assumed to slowly increase over time, no dramatic change is expected any time soon.

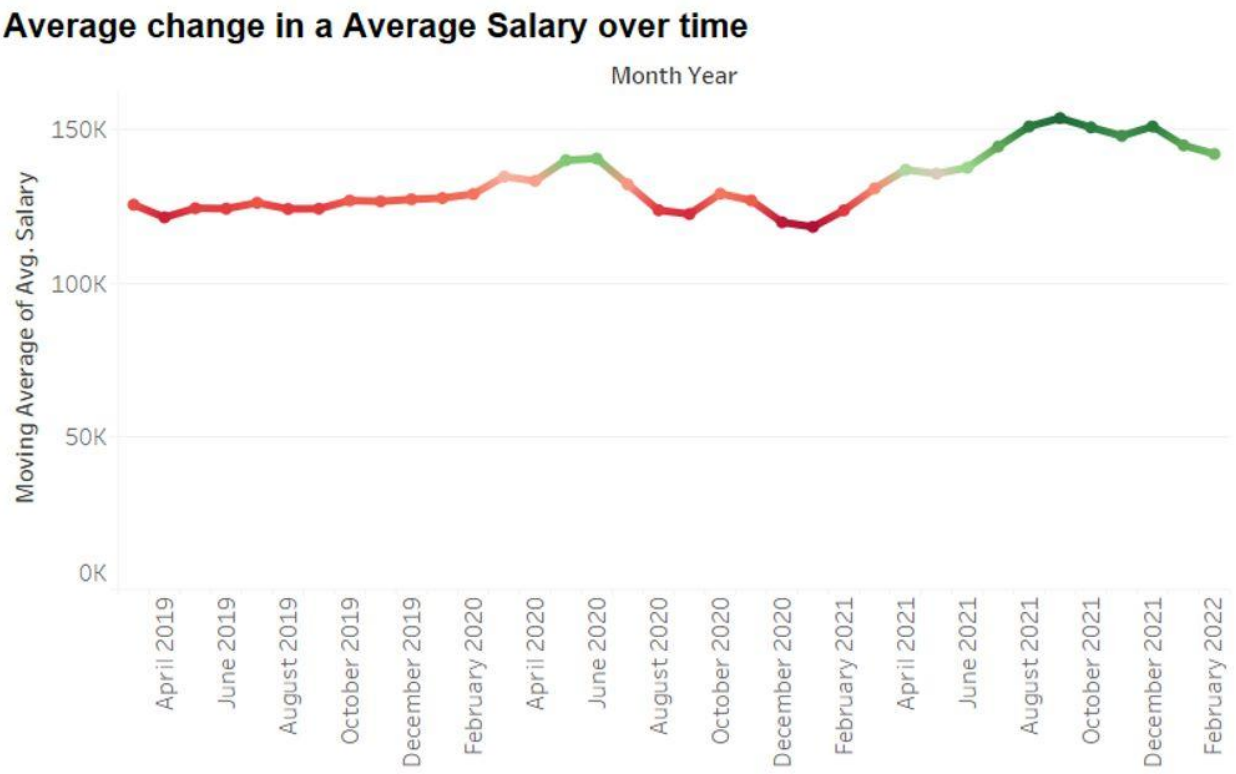

Figure 4.3: Average change in an Average Salary over time

4.2 Linear Regression Model

4.2.1 Relationship Between Salary and Total Skills

A scatter plot is used to see the distribution of salary against skills required. On top of that, a linear regression model is generated to see the change in salary as more skills are demanded. The point where the regression lines narrow down is where the frequency is the highest, which can be interpreted as the point most employers want from employees. In this case, five skills are typically asked by employers. Skills involve technological skills like programming languages, software, and many other services. Most importantly, the linear model shows a rising trend, meaning the payout grows as more skills are required for the job. This is a very good sign as years of study will be recognised.

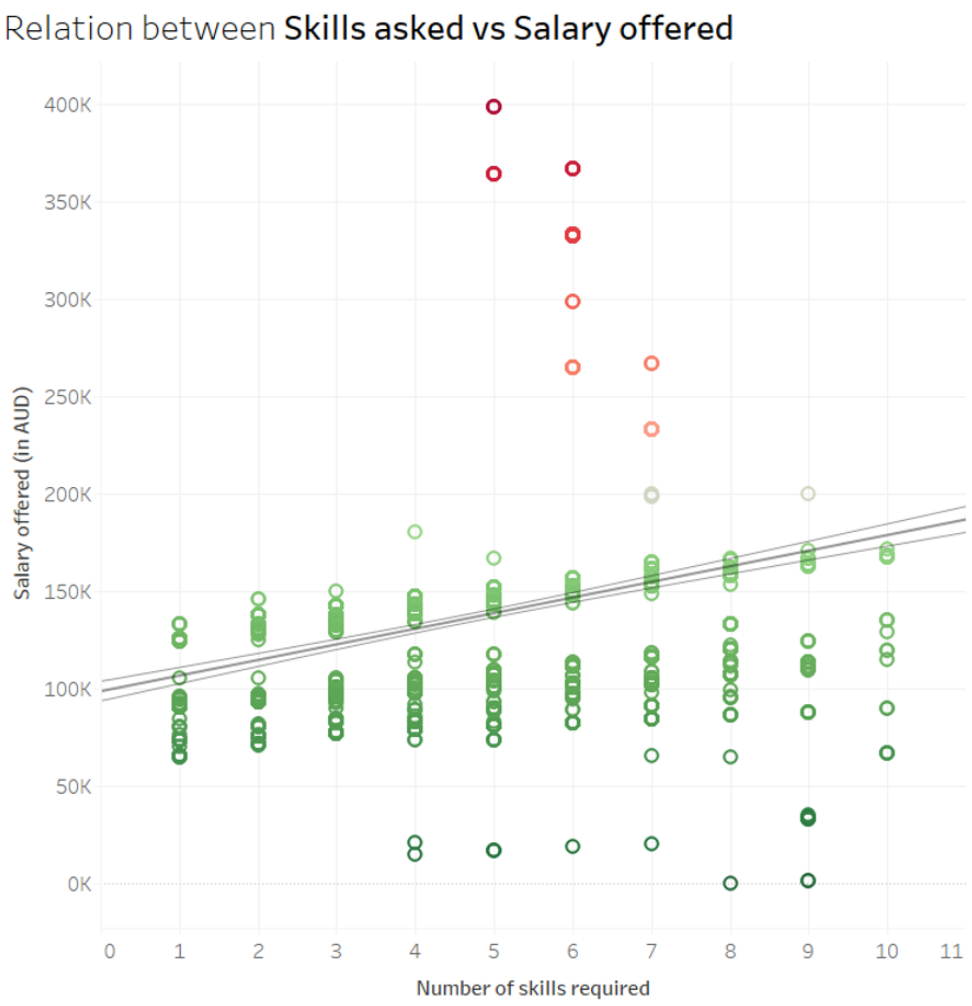

Figure 4.4: Relation between Skills asked vs Salary offered

4.2.2 Relationship Between Salary and Experience

Another job requirement that is important, other than skill, is years of experience. Again, a scatter plot and a linear regression model are displayed to explore how salary is distributed against experience and the relationship between the two. The linear lines narrow down to near 4.5 years of experience, meaning most employers ask for 4 to 5 years of experience. Interestingly, there is no data scattered below 3 years. We can tell that the dataset does not contain any entry-level or internships. Conversely, if you think, it could possibly mean that it is harder to enter the industry as a student or job seeker with minimal experience. However, if you can bear the difficulty of entering the field, you will be rewarded with a higher salary as more years of experience are accumulated. Moreover, no more than 6 years of experience is asked. It can be assumed we reach seniority in the data science field after 6 years and can apply for executive level jobs.

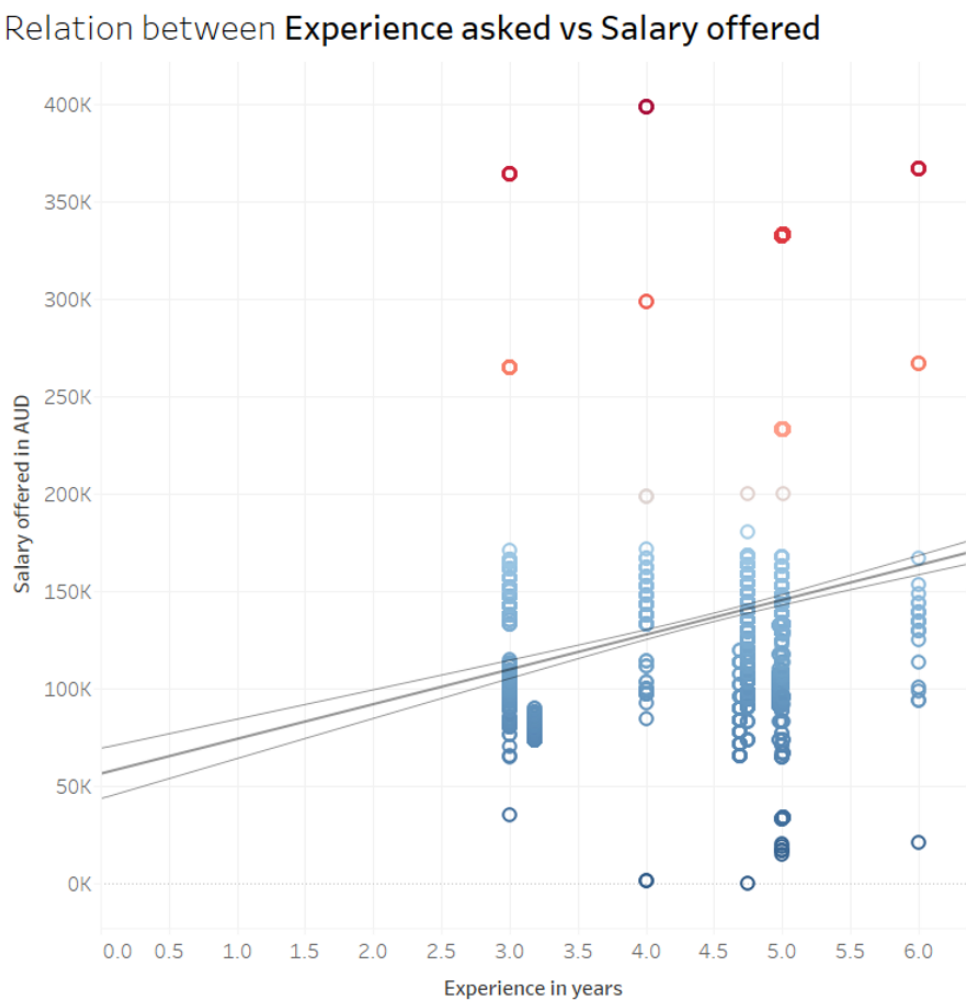

Figure 4.5: Relation between Experience asked vs Salary offered

5. Storytelling with data

5.1 Prescriptive Analysis

The visualisations are produced with R and Tableau.

5.1.1 Top Skills

The top ten skills required for Data Science Jobs are presented in the bar chart. Twenty-four skills are listed in the dataset, labelled as '0' for not required and '1' for required. Summing over these columns to get the counts for each skill appearing in the dataset. Since these skills cover a wide range of areas, some skills appear less

frequently (0 for most rows of that column) than others. To make a better plot, only the top ten skills are selected to create the bar chart. Not surprisingly, Python took 1st place, while SQL and R ranked 2nd and 3rd. Python, the most popular programming language, and R, the language for statistical computing, are the most demanded languages for data analysis. SQL, the Standard Query Language designed for interacting with databases, is a highly demanded complementary skill. Tableau is the most required software for data visualisation. Other skills, including SAS, MATLAB, Hadoop, Spark, Java and Scala, are also in the top ten rankings. However, some essential skills, such as Machine Learning, Deep Learning and AI Algorithms etc., do not appear in the dataset. To do further analysis on these additional skills, more data needs to be collected.

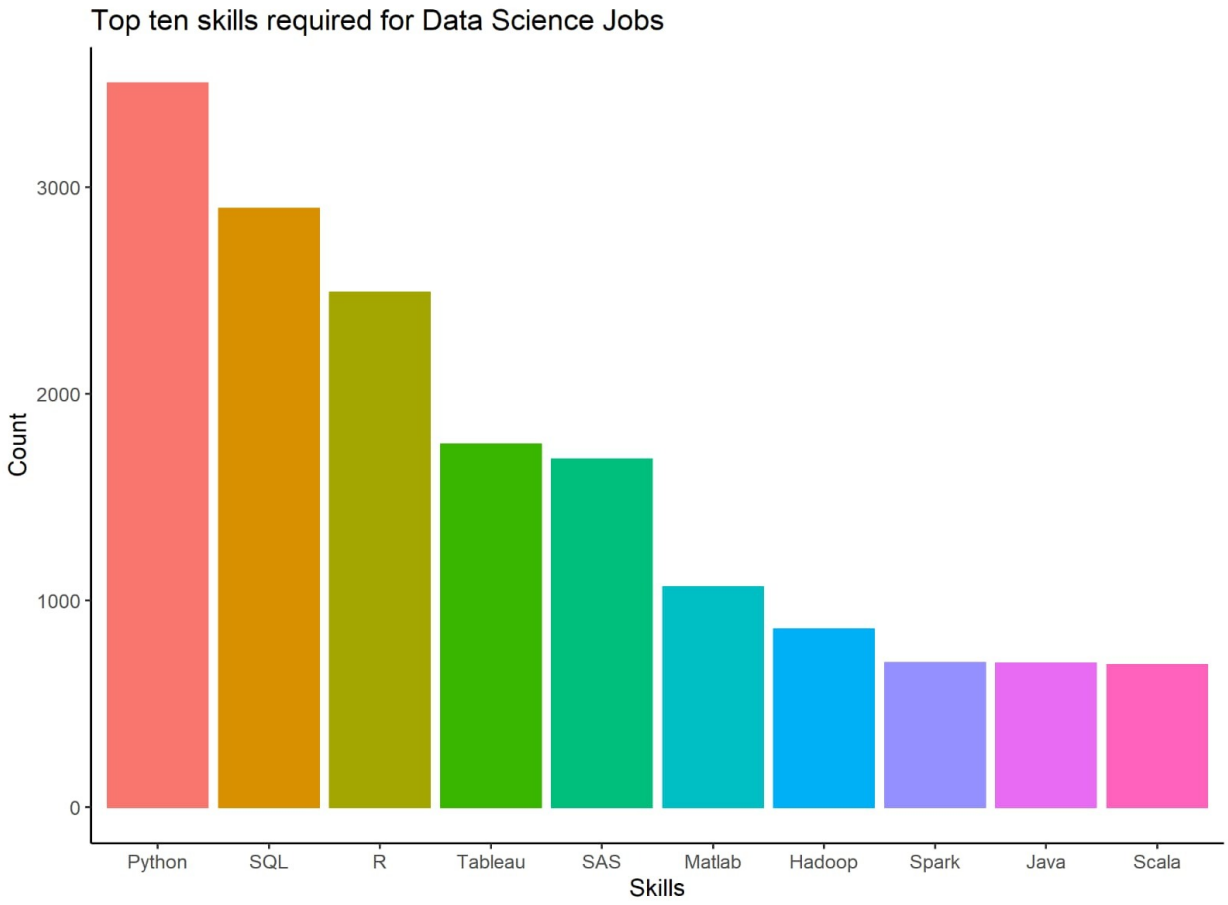

Figure 5.1: Top ten skills required for Data Science Jobs

5.1.2 Salary Against Right to Work

The average salary based on the Right to work is represented in the bar chart. Comparing the two groups, there is no significant impact on the right-to-work status except for data architects and other unknown titles. Among all the job types, Data Engineer is classified as the highest-income data science job, with an average salary of around $147,000. Other jobs like Analytics Manager, Data Scientist and Data Analyst have slightly lower salaries

ranging from approximate $132,000 to $140,000. Data Architect and other unknown job titles have the lowest salaries but the most significant difference in the right-to-work status.

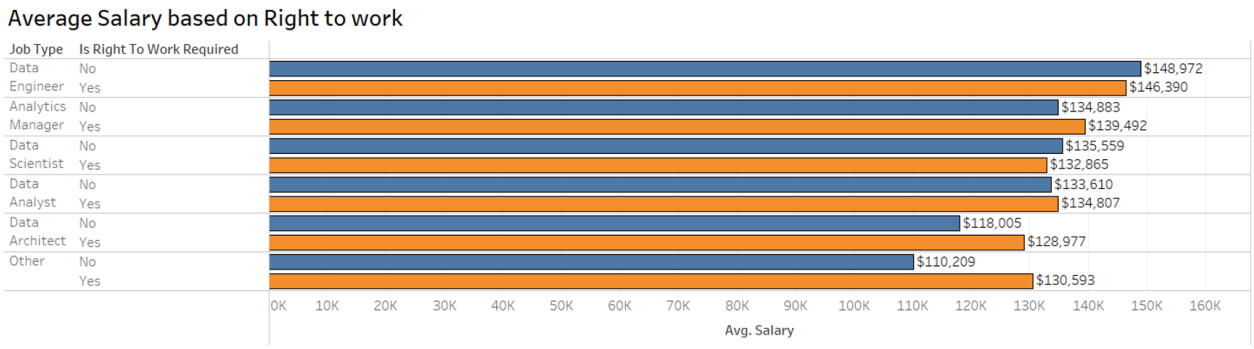

Figure 5.2: Average Salary based on RIght to work

5.1.3 Salary Against Domain

The following bar chart shows the average salary based on domain. The average salary is about $130,000, and domains are sorted according to the number of job postings. The information & Communication technology domain has the most job postings, but the salary is slightly below the average. Legal, Sales, and Sports & Recreation domains have fewer job postings but higher salaries. Government & Defence, Education & Training, Mining, Resources & Energy have more job postings but lower salaries. Salaries vary considerably between different domains, and there is no clear relationship between the average salary and the number of job postings.

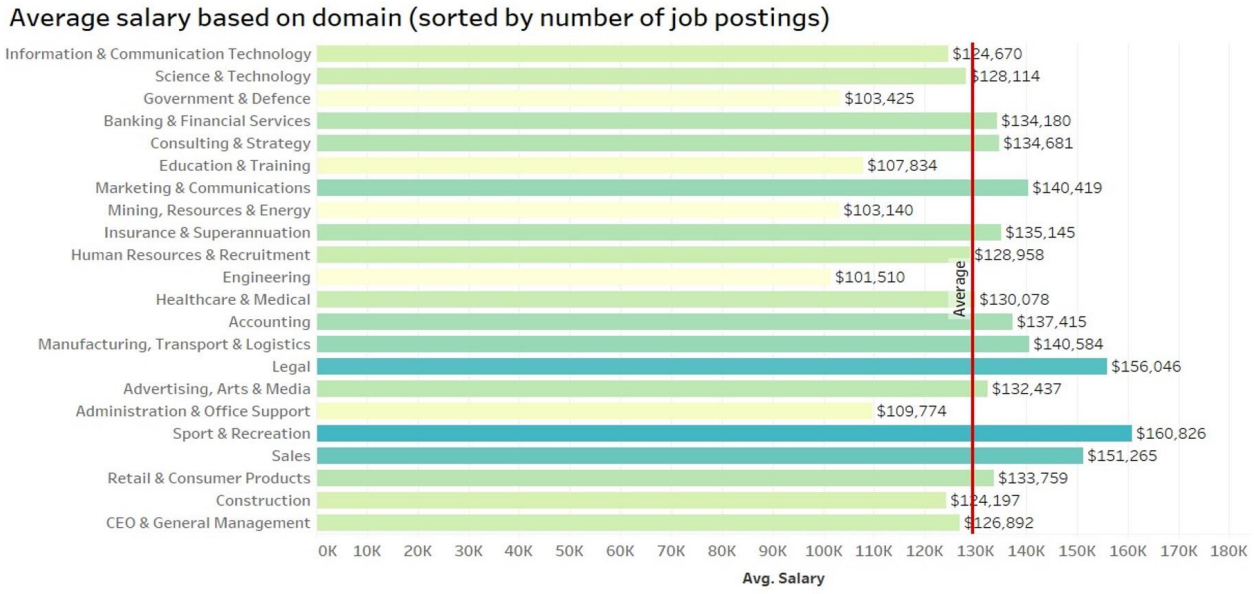

Figure 5.3: Average salary based on domain

5.1.4 Salary Against City/State

The average salaries against States in Australia is presented in the heat map. Victoria takes 1st place, with an average salary of over $154,000. New South Wales ranks 2nd, with a salary of around $140,000. The average salaries in the two states are significantly higher than in other states. Potential reasons for Victoria and New South Wales having the highest salaries are the two major cities, Melbourne and Sydney. Salaries in Queensland, Northern Territory, Western Australia and Tasmania are in the middle, ranging from approximate $95,000 to $103,000. South Australia has much lower salaries of about $85,000, while the Australian Capital Territory ranked last with a salary of around $78,000.

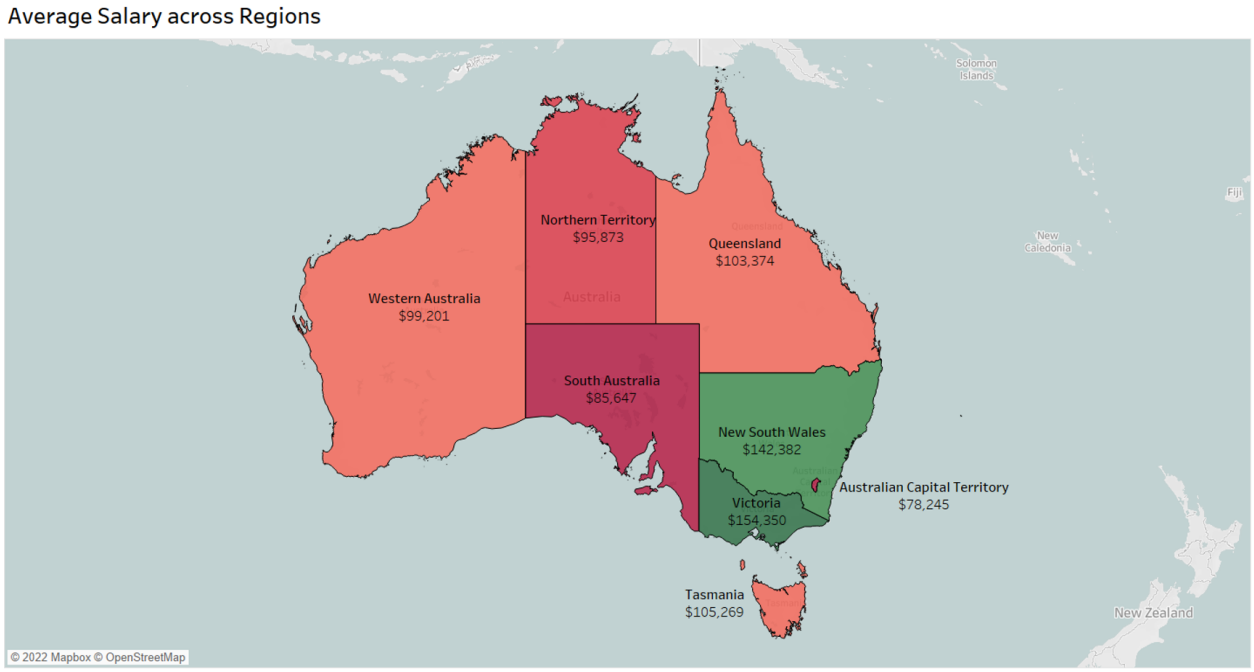

Figure 5.4: Average Salary across Regions

5.2 Regression Analysis

5.2.1 Trend of Consumer Price Index and Salary

To compare the trend of CPI and Salary, Linear regression analysis is done on both CPI and Average Salary of eight cities. Combining the two line graphs, it is clear that the cost of living increased rapidly from 2019 to 2022. Unlike the growth trend in CPI, the average salary remains relatively steady. Melbourne offers outstandingly high salaries than other cities, with Sydney offering lower salaries than Melbourne but higher than the other cities. There is no significant increase in salary. Although there is no significant increase in salary, there is no declining trend either.

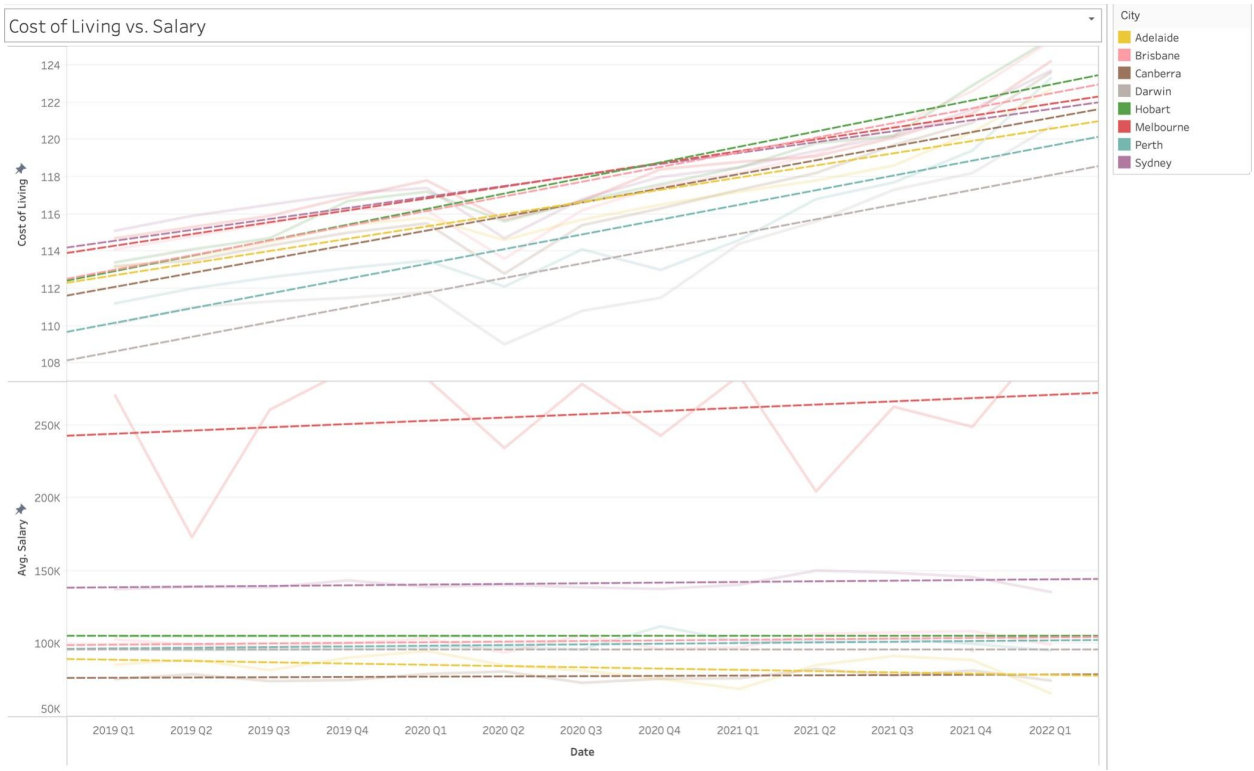

Figure 5.5: Cost of Living vs. Salary

6. Conclusion

- Cost of living increases faster than the average salary in Australia. Less savings. More expenses.
- Melbourne offers the highest average salary.
- Sydney offers more data science jobs (more opportunities) with great salaries.
- Learn Python, R, SQL, and Tableau. Highly recommended.
- 5 skills and 4 to 5 years of experience commonly desired. Rewarded with a higher salary if you have exceeding capabilities!
- Highly impacted (negative) by global disasters such as COVID-19.

7. Bibliography

ANJI BACANO (2022), Australia Data Science Jobs. Dataset retrieved from: https://www.kaggle.com/datasets/nadzmiagthomas/australia-data-science-jobs?select=AustraliaDataScienceJobs.csv

Australian Bureau of Statistics (2022), Consumer Price Index, Australia. Dataset retrieved from: https://www.abs.gov.au/statistics/economy/price-indexes-and-inflation/consumer-price-index-australia/latest-release#capital-cities-comparison

NOMILK (2022), Data Science Jobs Listings -Australia- 2019-2021. Dataset retrieved from: https://www.kaggle.com/datasets/nomilk/data-science-job-listings-australia-20192020?select=listings2019_2022.csv

What is a linear regression model? - MATLAB & Simulink. (n.d.). MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink. https://www.mathworks.com/help/stats/what-is-linear-regression.html

(n.d.). https://cdn.scribbr.com/wp-content/uploads/2020/09/iqr_boxplot.png.

8. Appendix

8.1 Datasets and Code

File	Description	Source
listings2019_2022.csv	CSV file containing every data science job listing between 2019 and 2022.	Kaggle (Seek)
All groups CPI, Index numbers(a).csv	CSV file containing CPI of Australian cities between 2018 and 2022.	Australian Bureau of Statistics
AustraliaDataScienceJobs.csv	CSV file containing every data science job listing in August 2022.	Kaggle (Glassdoor)
cleaned_withImputation_removed Outliers.csv	CSV file containing cleaned data for the project.	GROUP O
data science project.ipynb	Python (Jupyter) code used for data cleaning.	GROUP O
data_analysis.R	Code used to plot data on R.	GROUP O
CPI.R	Code used to plot changes in CPI	GROUP O
skill.R	Code used to plot top skills	GROUP O

data science project.ipynb

Importing libraries and Reading Datasets

```python
import pandas as pd
import numpy as np
import re
import warnings, random
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
warnings.filterwarnings("ignore")
pd.set_option('display.max_columns', None)

place_data = pd.read_csv(r"C:\Users\jaske\OneDrive\Desktop\studies\intro to data science\CPI.csv", skiprows = 1)
jobs_data = pd.read_csv(r"C:\Users\jaske\OneDrive\Desktop\studies\intro to data science\DS_JOBS_2019_2022.csv")
final_cleaned_data = r"C:\Users\jaske\Downloads\cleaned_withImputation_removedOutliers.csv"
```

Python

Getting only Australian cities

```python
cities = list(map(str.lower, place_data.columns[1:-1]))
cities
```

Python

```python
#Getting Jobs data for Australia and its cities
jobs_data = jobs_data[jobs_data["nation"] == "Australia"]
jobs_data["city"] = jobs_data["city"].str.lower()
jobs_data["city"].replace(to_replace = "act", value = "canberra", inplace=True )
jobs_data = jobs_data[jobs_data["city"].isin(cities)]

jobs_data.reset_index(drop=True, inplace=True)
jobs_data["city"].unique()
```

Python

```python
#Unique values in each column
unique_df = pd.DataFrame()

for i in jobs_data.columns:
    unique_df.at[0, i] = str(len(jobs_data[i].unique()))

print("Number of unique values in each column")
unique_df
```

Python

```python
#Count of nulls in each column
jobs_data.isna().sum()

null_df = pd.DataFrame()

for i in jobs_data.columns:
    null_df.at[0, i] = str(jobs_data[i].isna().sum())

print("Number of null values in each column")
null_df
```

Removing Unwanted columns

```python
jobs_data.columns
```

Python

```python
unwanted_columns = ["jobId", "advertiserId", "companyId", "teaser", "desktopAdTemplate", "companyProfileUrl", "seekJobListingUrl"]
jobs_data = jobs_data.drop(unwanted_columns, axis=1)
```

Python

Cleaning Data

```python
#Classifying on jobs to make jobtype
jobs_data["jobType"] = None

for i in range(len(jobs_data)):
    if("scientist" in jobs_data["jobTitle"][i].lower()):
        jobs_data["jobType"][i] = "Data Scientist"
    elif("analyst" in jobs_data["jobTitle"][i].lower()):
        jobs_data["jobType"][i] = "Data Analyst"
    elif("engineer" in jobs_data["jobTitle"][i].lower()):
        jobs_data["jobType"][i] = "Data Engineer"
    elif("manager" in jobs_data["jobTitle"][i].lower()):
        jobs_data["jobType"][i] = "Analytics Manager"
    elif("develop" in jobs_data["jobTitle"][i].lower()):
        jobs_data["jobType"][i] = "Data Architect"
    else:
        jobs_data["jobType"][i] = "Other"
```

Python

Python

```
#Parsing experience from online ads
jobs_data["experience"] = None

for i in range(len(jobs_data)):
    exp = None
    try:
        exp = list(filter(lambda sentence: ("year" in sentence)or("yr" in sentence) , jobs_data["mobileAdTemplate"][i].split(".")))[-1]
        exp = re.findall('\d+\s*[+]*\s*[y]{1}', exp)[0]
        exp = float(re.sub('\D', '', exp))
        jobs_data["experience"][i] = random.choice(range(5, 16)) if(exp >= 15) else exp
    except:
        jobs_data["experience"][i] = None
jobs_data["experience"] = jobs_data["experience"].astype('float64')
```

Python

Imputation using Mean Value

```
#Imputing mean values based on state level for experience column
for i in jobs_data["state"].unique():
    impute_value = round(jobs_data["experience"][jobs_data["state"] == i].mean(), 2)
    if(str(impute_value) == "nan"):
        impute_value = round(jobs_data["experience"].mean(), 2)

    print(i + " : " + str(impute_value))
    jobs_data.loc[jobs_data["state"] == i, "experience"] = jobs_data.loc[jobs_data["state"] == i, "experience"].fillna(impute_value)
```

Python

Imputation using Linear regression

```
lrm_data = jobs_data[["state", "salary", "total_skills", "experience"]]

for i in lrm_data["state"].unique():
    lrm_train = lrm_data.loc[lrm_data['salary'].notnull(), ][lrm_data['state'] == i]

    independant = lrm_train[["total_skills", "experience"]]
    dependant = lrm_train["salary"]

    lrm = LinearRegression().fit(independant, dependant)
    for j in jobs_data[jobs_data.isnull().any(axis=1)][jobs_data["state"] == i].index:
        jobs_data["salary"][j] = round(lrm.predict([[jobs_data["total_skills"][j], jobs_data["experience"][j]]])[0], 2)
```

Python

Outlier Removal

```
jobs_data.boxplot(column = "salary", figsize=(5, 5))
plt.title("Before Outlier Removal")
plt.show()
```

Python

```
jobs_data.boxplot(column = ["total_skills", "experience"], figsize=(10, 5))
plt.title("Before Outlier Removal")
plt.show()
```

Python

```
#Calculating salary from salarystring
jobs_data["salary"] = None

for i in range(len(jobs_data)):
    value = str(jobs_data["salary_string"][i])

    remove_list = [',', 'per day', 'per annum', 'p.a.', 'p.d.', 'p/d', 'p/a', '.00']
    for x in remove_list:
        value = value.replace(x, "")

    value = re.sub('\d+[\.]*\d*[%]+', '', value)
    value = re.sub('[\.]*\d*[%]+', '', value)
    value = re.sub('[\.]+\d*', '', value)
    value = re.findall(r'[-]*\s*\d+\s*\w{1}', value)

    try:
        jobs_data["salary"][i] = re.sub('\D', '', value[-1].replace("k", "000").replace("K", "000"))
    except:
        jobs_data["salary"][i] = None

jobs_data["salary"] = pd.to_numeric(jobs_data["salary"])
jobs_data["salary"] = jobs_data["salary"].abs()
```

Python

```
#Calculating total skills for each job
skill_list = ['R', 'Python', 'Matlab', 'SQL', 'Stata', 'Minitab', 'SPSS', 'Ruby', 'C', 'Scala', 'Tableau', 'Java', 'Hadoop', 'SAS', 'Julia', 'Knime', 'D3', 'Clojure', 'Haskell', 'I
jobs_data["total_skills"] = jobs_data[skill_list].sum(axis = 1)
```

Python

```
#Parsing experience from online ads
```

data_analysis.R

```
    l[[i]] <- r
  }
}

plot(l[[1]],type='l',xlab="Fiscal Quarters",ylab="Count",xaxt='n',col='red',
     ylim=c(0,300),main="Number of Data Science Jobs per City")
points(l[[2]],col='blue',type='l')
points(l[[3]],col='green',type='l')
points(l[[4]],col='purple',type='l')
points(l[[5]],col='pink',type='l')
points(l[[6]],col='lightblue',type='l')
points(l[[7]],col='orange',type='l')
points(l[[8]],col='darkgreen',type='l')
legend(x="topright",legend=o_cities,col=c('red','blue','green','purple','pink','lightblue','orange','dar
       lty=1,horiz=TRUE,cex=0.8,text.width=0.35)
axis(side=1,at=c(1:length(quarterly_trend)),labels=c("Q1 2019","Q2 2019","Q3 2019","Q4 2019",
                                                     "Q1 2020","Q2 2020","Q3 2020","Q4 2020",
                                                     "Q1 2021","Q2 2021","Q3 2021","Q4 2021"))


o_cities <- c("Sydney","Melbourne","Brisbane","Perth","Canberra","Darwin","Hobart","Adelaide")
# CPI dataset
sydney <- data.frame(x=CPI[c(1:17),'Period'],CPI=CPI[-c(18:23),'Sydney'])
melbourne <- data.frame(x=c(17:1),CPI=CPI[-c(18:23),'Melbourne'])
brisbane <- data.frame(x=c(17:1),CPI=CPI[-c(18:23),'Brisbane'])
perth <- data.frame(x=c(17:1),CPI=CPI[-c(18:23),'Perth'])
canberra <- data.frame(x=c(17:1),CPI=CPI[-c(18:23),'Canberra'])
darwin <- data.frame(x=c(17:1),CPI=CPI[-c(18:23),'Darwin'])
hobart <- data.frame(x=c(17:1),CPI=CPI[-c(18:23),'Hobart'])
adelaide <- data.frame(x=c(17:1),CPI=CPI[-c(18:23),'Adelaide'])
plot(sydney,col="red",type='l',xlab="Fiscal Quarters",xaxt='n',ylim=c(100,135))
points(melbourne,col='blue',type='l')
points(brisbane,col='green',type='l')
points(perth,col='purple',type='l')
points(canberra,col='pink',type='l')
points(darwin,col='lightblue',type='l')
points(hobart,col='orange',type='l')
points(adelaide,col='darkgreen',type='l')
legend(x="topright",legend=o_cities,col=c('red','blue','green','purple','pink','lightblue','orange','dar
       lty=1,horiz=TRUE,cex=0.75,text.width=0.6)
axis(side=1,at=c(1:17),labels=c("Q2 2018","Q3 2018","Q4 2018","Q1 2019","Q2 2019","Q3 2019"
                                ,"Q4 2019","Q1 2020","Q2 2020","Q3 2020","Q4 2020",
                                "Q1 2021","Q2 2021","Q3 2021","Q4 2021","Q1 2022","Q2 2022"))

# prediction/linear model
syd_lm <- lm(CPI~x,data=sydney)
mel_lm <- lm(CPI~x,data=melbourne)
bri_lm <- lm(CPI~x,data=brisbane)
per_lm <- lm(CPI~x,data=perth)
can_lm <- lm(CPI~x,data=canberra)
dar_lm <- lm(CPI~x,data=darwin)
hob_lm <- lm(CPI~x,data=hobart)
ade_lm <- lm(CPI~x,data=adelaide)

# root mean square error (RMSE) of prediction model
syd_rmse <- sqrt(sum(abs(predict(syd_lm,sydney)-sydney$CPI)^2)/17)
mel_rmse <- sqrt(sum(abs(predict(mel_lm,melbourne)-melbourne$CPI)^2)/17)
bri_rmse <- sqrt(sum(abs(predict(bri_lm,brisbane)-brisbane$CPI)^2)/17)
per_rmse <- sqrt(sum(abs(predict(per_lm,perth)-perth$CPI)^2)/17)
can_rmse <- sqrt(sum(abs(predict(can_lm,canberra)-canberra$CPI)^2)/17)
dar_rmse <- sqrt(sum(abs(predict(dar_lm,darwin)-darwin$CPI)^2)/17)
hob_rmse <- sqrt(sum(abs(predict(hob_lm,hobart)-hobart$CPI)^2)/17)
ade_rmse <- sqrt(sum(abs(predict(ade_lm,adelaide)-adelaide$CPI)^2)/17)

# predict future CPI for each city - Index reference period: 2011-12 = 100.0.
n <- data.frame(x=c(16:37))
plot(predict(syd_lm,n),col="red",type='l',xlab="Years",ylab="CPI",xaxt='n',ylim=c(110,160),main="Future
points(predict(mel_lm,n),col='blue',type='l')
```

```
points(predict(bri_lm,n),col='green',type='l')
points(predict(per_lm,n),col='purple',type='l')
points(predict(can_lm,n),col='pink',type='l')
points(predict(dar_lm,n),col='lightblue',type='l')
points(predict(hob_lm,n),col='orange',type='l')
points(predict(ade_lm,n),col='darkgreen',type='l')
legend(x="topright",legend=o_cities,col=c('red','blue','green','purple','pink','lightblue','orange','dark
       lty=1,horiz=TRUE,cex=0.7,text.width=0.9)
axis(side=1,at=c(1:21),labels=c("2022","","","","2023","","","","2024","",
                                "","","2025","","","","2026","","","","2027"))

CPI<-CPI[-c(15:23),]
CPI[,1]<-(c("2022-06-01 00:00:00","2022-03-01 00:00:00","2021-12-01 00:00:00","2021-09-01 00:00:00",
            "2021-06-01 00:00:00","2021-03-01 00:00:00","2020-12-01 00:00:00","2020-09-01 00:00:00",
            "2020-06-01 00:00:00","2020-03-01 00:00:00","2019-12-01 00:00:00","2019-09-01 00:00:00",
            "2019-06-01 00:00:00","2019-03-01 00:00:00"))
write.csv(CPI,"CPI_CLEAN.csv")
```

CPI.R

```
library(ggplot2)

cpi = read.csv("datasets/CPI.csv")

cpi_city <- cpi[cpi$City != "Weighted Average", ]
cpi_average <- cpi[cpi$City == "Weighted Average", ]

# CPI changes in eight cities
ggplot(data = cpi_city,
       mapping = aes(x = Period, y = CPI, group = City, colour = City)) +
  geom_line() +
  labs(x = "Period",
       y = "CPI",
       title = "CPI changes in eight capital cities in Australia") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 18))

ggsave("plots/CPI_city.png", width = 20, height = 15, units = "cm")
```

```
# Weighted Average CPI
ggplot(data = cpi_average,
       mapping = aes(x = Period, y = CPI, group = 1)) +
  geom_line(size = 1, color  ="#8968CD") +
  geom_point(size = 3, alpha = 0.3, color  ="#8968CD") +
  labs(x = "Period",
       y = "CPI",
       title = "Change in weighted average CPI for eight capital cities in Austra
  scale_y_continuous(breaks = c(112, 114, 116, 118, 120, 122, 124, 126)) +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 18))

ggsave("plots/CPI_weighted_average.png", width = 20, height = 15, units = "cm")
```

skill.R

```r
library(tidyverse)
library(forcats)
library(wordcloud)
library(RColorBrewer)

dsjobs <- read.csv("datasets/DS_JOBS_2019_2022.csv")
dsjobs <- dsjobs[, c(25:49)]

R <- sum(dsjobs$R)
Python <- sum(dsjobs$Python)
Matlab <- sum(dsjobs$Matlab)
SQL <- sum(dsjobs$SQL)
Strata <- sum(dsjobs$Stata)
Minitab <- sum(dsjobs$Minitab)
SPSS <- sum(dsjobs$SPSS)
Ruby <- sum(dsjobs$Ruby)
C <- sum(dsjobs$C)
Scala <- sum(dsjobs$Scala)
Tableau <- sum(dsjobs$Tableau)
Java <- sum(dsjobs$Java)
Hadoop <- sum(dsjobs$Hadoop)
SAS <- sum(dsjobs$SAS)
Julia <- sum(dsjobs$Julia)
Knime <- sum(dsjobs$Knime)
```

```r
D3 <- sum(dsjobs$D3)
Clojure <- sum(dsjobs$Clojure)
Spark <- sum(dsjobs$Spark)
Javascript <- sum(dsjobs$Javascript)
F. <- sum(dsjobs$F.)
Fortan <- sum(dsjobs$Fortran)


COUNT <- c(R, Python, Matlab, SQL, Strata, Minitab, SPSS, Ruby, C, Scala,
           Tableau,Hadop, SAS, Julia, Knime, D3, Clojure, Spark, Javascript,
           F.,Fortan)
skills = c("R", "Python", "Matlab", "SQL", "Strata", "Minitab",
           "SPSS", "Ruby", "C", "Scala", "Tableau", "Java",
           "Hadoop", "SAS", "Julia", "Knime", "D3", "Clojure",
           "Spark", "Javascript", "F.", "Fortan")
df <- data.frame(Skills = skills, Count = COUNT)
df <- df[order(df$Count, decreasing = TRUE), ]
df <- df[1:10, ]
```

```r
# Top ten skills barplot
df %>%
  ggplot(aes(x = reorder(Skills, -Count), y = Count,
             color = Skills, fill = Skills)) +
    geom_bar(stat = "identity") +
    labs(x = "Skills",
         y = "Count",
         title = "Top ten skills required for Data Science Jobs") +
    theme(plot.title=element_text(size=20)) +
    theme_classic() +
    theme(legend.position="none")

ggsave("plots/top_ten_skills.png", width = 20, height = 15, units = "cm")


# World cloud
set.seed(29)
wordcloud(words = skills,
          freq = COUNT,
          colors=brewer.pal(8, "Dark2"))
```

8.2 Response to Peer Reviews

In the part of "is the data fit for use", the group mentioned 61% of salary data is missing, since it's the most important feature, they should abandon this dataset and choose a better one.

We do strongly agree with this feedback. If we could go back in time, we would certainly find a completely new dataset. However, it was too late to change the dataset as the submission was not far away. If we did abandon the dataset, we would start over with a new topic because job datasets were hard to find, even on Kaggle. The data was already cleaned and several analyses were done, so we decided to continue with no other choice.

As there could be changes in taxes in each state and cost of living differs from each state and each city being considered into the analysis and more data from different city and cost of living must be plotted into the analysis for which the analysis is being conducted.

The analysis of the relationship between the city, salary, and cost of living was planned and mentioned at the end of the trial presentation. It has been analysed.

Outlier detection needs to be done before visualisation in cleaning of the data. It represents bad data quality which might not have been properly run or the data might have been coded wrongly. If an outlying point is found to be incorrect, the outlying value should be eliminated or corrected from the analysis.

We were strongly aware of this issue and it was briefly discussed in the trial presentation. It had a significant influence on the analysis as the trend was irregular and unexplainable. This has been taken care of and explained in detail under 3.2.2.

It is a good idea to have a concluding slide to reiterate the findings and to remind the audience that the presentation is coming to an end.

We have added a conclusion slide to show the results of our analysis.

Imputation should be performed to get a better understanding of the graphs shown. There are irregularities because no imputation was performed.

A linear regression imputation was used to fill in the missing salaries. Salary as the dependent variable and skills and experience as independent variables.

Linear regression imputation is a type of imputation that we don't recommend because it will strengthen the linear relationship between variables.

We believe missing salaries are not missing at random (NMAR). Would it make sense if an employee got paid below average with 10 years of experience? No

Therefore, we performed linear regression imputation with salary as the dependent variable and skills and experience as independent variables.

Another potential stakeholder could be those skilled migrants, those data science experts who want to use their strong data science skills to immigrate to Australia.

Migrants can be somewhat considered job seekers. The answers to the research questions could still be useful.