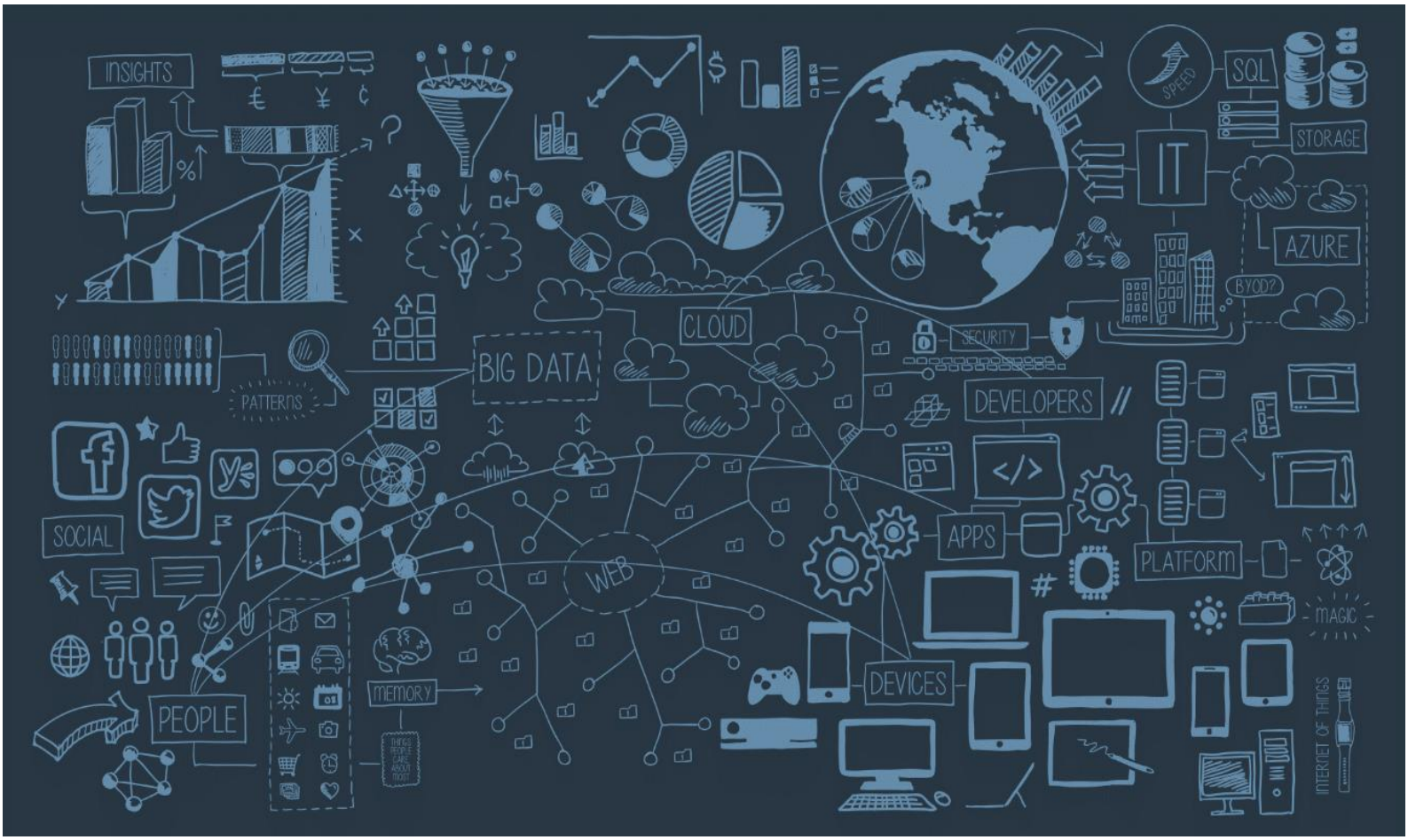
Visualisation for Data Analytics

Tarun Kurapati

Student ID: 21020862



Introduction

Here are my findings from the Data Scientist Jobs Dataset analysis and visualisation. The dataset is taken from Kaggle. In this dataset, there are 3909 rows of data and 17 columns of data (for example, ‘Salary’, ‘Revenue’, 'Job Title', 'Salary Estimate,' and so on). To make the data more understandable, I examined and displayed the dataset to extract information from it and display the information more clearly.

Due to the pandemic, many people lost their jobs; this dataset can help refine the job search so more people in need will be able to find employment. This dataset contains over 3900 job listings for data scientist roles, with attributes such as: Location, Revenue, Rating, Easy Apply, and so on.

Summary

For this project, I have used NumPy, Pandas, Matplotlib, Seaborn, TextWrap and finally WordCloud. The **NumPy** library is considered to be one of the most powerful Python libraries. It is widely used for computing arrays in the industry [1]. **Pandas** is an open-source library for the Python programming language that provides high-performance, easy-to-use data structures and data analysis tools for use with **NumPy** offers high-performance, easy-to-use data structures. In addition to this, it provides a quick way to clean and prepare data for analysis [2]. Among Python's most capable packages for plotting data, **Matplotlib** is one of the most popular. It is a cross-platform package with a variety of tools for displaying 2D plots from Python data sets in lists or arrays [3]. **Seaborn**, a Python library, is mostly used to make statistical visuals or graphs [4]. With the **Textwrap** command, plain text can be wrapped and formatted to make it look nice. A line break in the input paragraph will be automatically adjusted in this module to allow for text formatting [5]. A **WordCloud** is a data visualisation technique for displaying data from text documents, with the size of each word indicating its frequency or importance. The use of colour and volume in word clouds can highlight important textual information [6].

```
import numpy as nu
import pandas as pd
import matplotlib.pyplot as polt
import seaborn as sbn
sbn.set(style = 'whitegrid')
import textwrap
from wordcloud import WordCloud, STOPWORDS
```

WordCloud

From the graph below, it is clear that open positions aren't just available for data scientists. Thus, in the Job title columns, we remove any data that does not contain the words data scientist or data science, because we are only interested in the data scientist positions.

Data Pre-processing

It is always a good practice to process the data, to eliminate null or empty values from the dataset.

```
stat.drop(['Unnamed: 0', 'Competitors', 'Founded', 'Sector'], axis = 1, inplace = True)
stat = stat.set_index(['index'])
# Replace data with a value of -1 with a value of NaN
stat = stat.replace([-1, -1.0, '-1'], nu.nan)
print(stat.isnull().sum(axis = 0))
# Fill in the rows where there are NaN values in the column 'Easy Apply'
```

```
stat['Easy Apply'].fillna('FALSE', inplace = True)
# Rows with NaN values should be dropped
stat.dropna(axis = 0, inplace = True)
stat
```

Cleaning “Salary” Column

Applied data cleaning on the Salary column, which helped to find the attributes such as “Salary Estimates”, “Min and Max Salaries”.

```
stat['Salary Estimate']= stat['Salary Estimate'].str.replace('(', '').str.replace(')', '').str.replace('Glassdoor est.', '').str.replace('Employer est.', '')
stat['Min Salary'],stat['Max Salary']=stat['Salary Estimate'].str.split('-').str
stat['Min Salary'] = stat['Min Salary'].str.strip(' ').str.strip('$').str.strip('K').fillna(0).astype(int)
stat['Max Salary'] = stat['Max Salary'].str.replace('Per Hour','')
stat['Max Salary'] = stat['Max Salary'].str.strip(' ').str.strip('$').str.strip('K').fillna(0).astype(int)
stat[['Salary Estimate','Min Salary', 'Max Salary']]
```

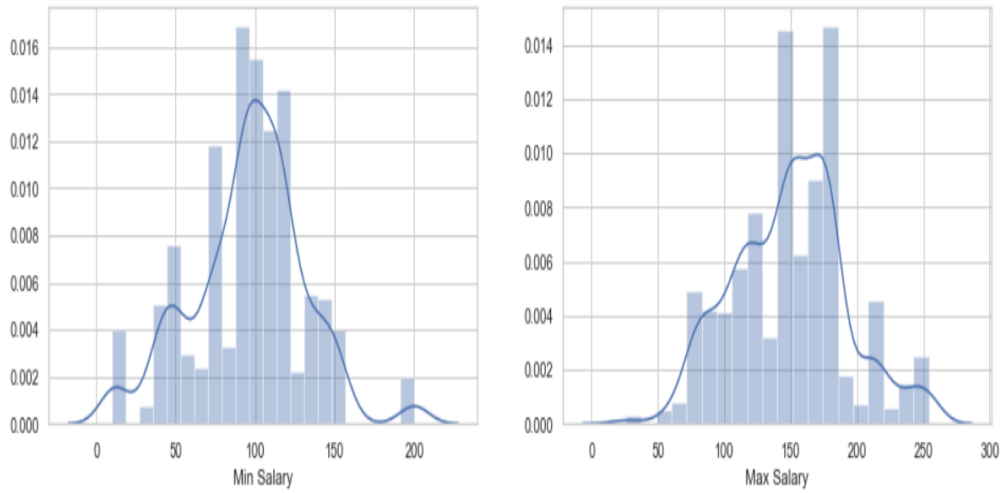
Out[7]:

index	Salary Estimate	Min Salary	Max Salary
0	$111K-181K	111	181
1	$111K-181K	111	181
3	$111K-181K	111	181
4	$111K-181K	111	181
5	$111K-181K	111	181
...	...	...	...
4375	$55K-112K	55	112
4376	$55K-112K	55	112
4377	$55K-112K	55	112
4378	$55K-112K	55	112
4379	$55K-112K	55	112

3296 rows × 3 columns

To visually represent the Min and Max Salary data, utilized seaborn library.

```
fig, ax = polt.subplots(1,2, figsize = (16,4))
sbn.distplot(ax = ax[0], a = stat['Min Salary'])
sbn.distplot( ax = ax[1], a = stat['Max Salary'])
polt.show()
```



Cleaning “Revenue” Column Value

Removed all the NaN values from the “Revenue” column.

```
stat['Revenue'].replace(['Unknown / Non-Applicable'], nu.nan, inplace = True)
stat[['Revenue']]
```

Out[10]:

index	Revenue
0	NaN
1	NaN
3	NaN
4	NaN
5	NaN
...	...
4375	10*to*25 million (USD)
4376	50*to*100 million (USD)
4377	100*to*500 million (USD)
4378	Less than $1 million (USD)
4379	2*to*5 billion (USD)

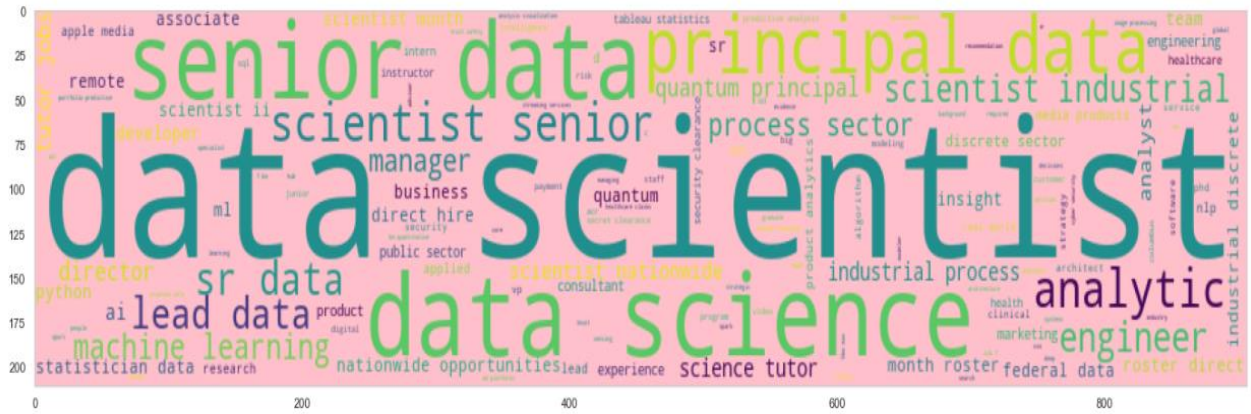
3296 rows × 1 columns

Cleaning “Job Title” Column Values

Using StopWords from WordCloud we segregate the Data Science and Data Scientist from the Job Title column.

```
# comment_words
com_words = ''
stopwords = set(STOPWORDS)

for vale in stat['Job Title']:
```

```
    vale = str(val)
    tokens = vale.split()

    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()

    com_words += " ".join(tokens)+" "

wordcloud = WordCloud(background_color = pink, height = 211, width = 907,
                      stopwords = stopwords, max_font_size = 211).generate(com_words)
fig, ax = polt.subplots(figsize = (19, 19))
ax.grid(False)
ax.imshow((wordcloud))
fig.tight_layout(pad=1)
polt.show()
```



This graph shows that data scientists are not the only roles available for hiring. In order to exclude data from Job title columns that do not contain the words data scientist or data science, we delete any data from Job title columns that do not contain this word [7].

```
stat = stat[stat['Job Title'].str.contains('Data Science|Data Scientist')]
stat
```

index	Job Title	Salary Estimate	Job Description	Rating	Company Name	Location	Headquarters	Size	Type of ownership	Industry	Revenue	Easy Apply	S
0	Senior Data Scientist	111K–181K	ABOUT HOPPER\n\nAt Hopper, we're on a mission ...	3.5	Hopper\n3.5	New York, NY	Montreal, Canada	501 to 1000 employees	Company - Private	Travel Agencies	NaN	FALSE	
1	Data Scientist, Product Analytics	111K–181K	At Noom, we use scientifically proven methods ...	4.5	Noom US\n4.5	New York, NY	New York, NY	1001 to 5000 employees	Company - Private	Health, Beauty, & Fitness	NaN	FALSE	
4	Director, Data Science	111K–181K	Director, Data Science - (200537)\nDescription...	3.4	United Entertainment Group\n3.4	New York, NY	New York, NY	51 to 200 employees	Company - Private	Advertising & Marketing	NaN	FALSE	
5	Data Scientist	111K–181K	Job Brief\n\nThe ideal candidate will have pre...	2.9	IFG Companies\n2.9	New York, NY	Hartford, CT	201 to 500 employees	Company - Private	Insurance Carriers	NaN	FALSE	
10	Data Scientist	111K–181K	Company Description:\n\nQuartet is a pioneerin...	3.9	Quartet Health\n3.9	New York, NY	New York, NY	201 to 500 employees	Company - Private	Enterprise Software & Network Solutions	NaN	True	
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4301	Principal Data Scientist, Pricing	55K–113K	We believe a large part of building an effecti...	3.8	Root Insurance\n3.8	Columbus, OH	Columbus, OH	501 to 1000 employees	Company - Private	Insurance Carriers	100to 500 million (USD)	FALSE	
4320	Data Science Practitioner	55K–113K	M LEVEL 9 OTHER The Data Science Specialist (R...	2.5	Avacend, Inc.\n2.5	Columbus, OH	Alpharetta, GA	51 to 200 employees	Company - Private	Staffing & Outsourcing	NaN	FALSE	
4349	CCB F&BM - Quantitative Modeling/Data Science ...	39K–86K	JPMorgan Chase& Co. (NYSE: JPM) is a leading g...	3.9	JPMorgan Chase Bank, N.A.\n3.9	Columbus, OH	New York, NY	10000+ employees	Company - Public	Investment Banking & Asset Management	$10+ billion (USD)	FALSE	

Visualization and Analysis

For the Analysis and visualisation section I have started from “Rating”.

```
viewdata = stat.groupby('Rating')['Job Title'].count().reset_index()
viewdata.sort_values('Job Title', ascending = False).head()
```

	Rating	Job Title
21	3.9	88
19	3.7	86
23	4.1	76
18	3.6	74
24	4.2	73

```
fig, ax = polt.subplots(figsize = (25, 10))
#sns.barplot(ax = ax, data = dataview, x = 'Rating', y = 'Job Title', order = dataview.sort_values('Job Title', ascending = False).Rating)
sbn.barplot(stat = viewdata, x = 'Rating', y = 'Job Title', palette = 'deep', ax = ax)
ax.set_ylabel('Count Jobs')
for index,viewdata in enumerate(viewdata['Job Title'].astype(int)):
    ax.text(x=index-0.1 , y =viewdata , s= f"{viewdata}" , fontdict=dict(fontsize=10))
polt.show()
```

This will show the graph where how number of jobs available for each rating, for example 3.7 rating job has 85 jobs available and 5.0 rating jobs have 17 jobs available.

To make, my visualisation much interesting I have compared Rating column with other available columns; I have compared Rating with Industry and Location.

Rating Vs Industry

This section compared the Ratings of the companies in each Industries.

```
viewdata_up = stat.groupby('Industry')['Rating'].mean().reset_index()
viewdata_up = viewdata_up.sort_values('Rating', ascending = False).head(10)

viewdata_dow = stat.groupby('Industry')['Rating'].mean().reset_index()
viewdata_dow = viewdata_dow.sort_values('Rating', ascending = True).head(10)

print(viewdata_up, '\n' )
print(viewdata_dow)
```

To represent the data virtually, I have used seaborn barplot; this give us the rating of at least 10 companies rating in each industry.

	Industry	Rating
9	Building & Personnel Services	4.900000
33	Food Production	4.600000
68	Transportation Management	4.500000
36	Grantmaking Foundations	4.200000
13	Colleges & Universities	4.107692
66	Telecommunications Services	4.100000
21	Education Training Services	4.087500
67	Transportation Equipment Manufacturing	4.050000
64	Staffing & Outsourcing	4.025455
24	Enterprise Software & Network Solutions	4.017544

	Industry	Rating
10	Cable, Internet & Telephone Providers	2.400000
52	News Outlet	2.700000
70	Utilities	2.800000
56	Pet & Pet Supplies Stores	2.900000
35	Gas Stations	2.900000
73	Wholesale	2.960000
23	Energy	3.066667
34	Gambling	3.100000
4	Audiovisual	3.100000
20	Department, Clothing, & Shoe Stores	3.130000

```
max_width = 15
ratingdata = [viewdata_up, viewdata_dow]
titledata = ['Top 10', 'Bottom 10']
fig, ax = polt.subplots(2,1, figsize = (26,14))
fig.subplots_adjust(hspace = 0.5)
for i in range(0,2):
    sbn.barplot(ax = ax[i], data = ratingdata[i], x = 'Industry', y = 'Rating', color = 'aqua', label = 'Rating')
    ax[i].set_title(titledata[i]+' Average Rating Company in Each Industry', fontsize = 20)
    ax[i].set_ylabel('Rating', fontsize = 20)
    ax[i].set_xlabel('Industry', fontsize = 20)
    ax[i].set_xticklabels(textwrap.fill(x.get_text(), max_width) for x in ax[i].get_xticklabels())
    ax[i].set_yticks(nu.arange(0, 5, step = 0.5))
    for index, ratingdata[i] in enumerate(nu.round(ratingdata[i]['Rating'], 2)):
        ax[i].text(x=index-0.1 , y =ratingdata[i] , s=f"{ratingdata[i]}" , fontdict=dict(fontsize=16))
    ax[i].tick_params(labelsize = 18)
polt.show()
```

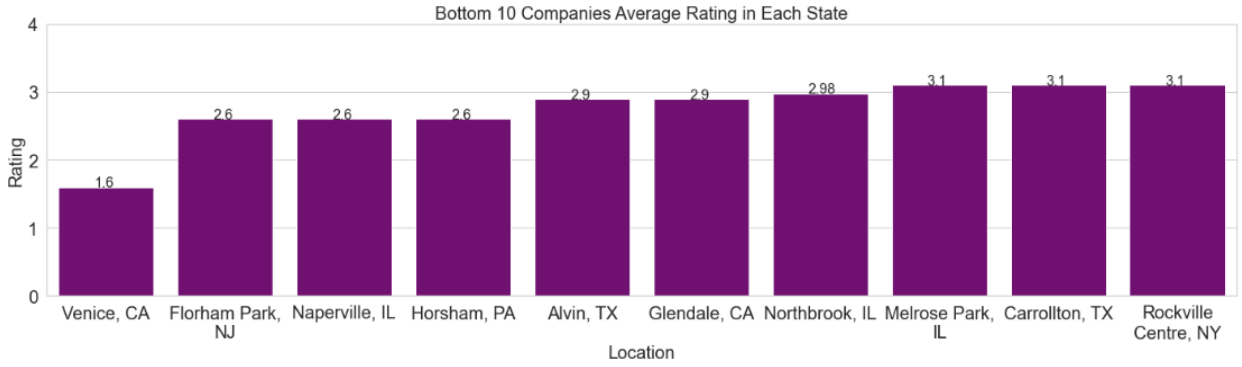
Rating Vs Location

This section compares the average Rating of at least 10 companies in each state.

```
viewdata_up = stat.groupby('Location')['Rating'].mean().reset_index()
viewdata_up = viewdata_up.sort_values('Rating', ascending = False).head(10)

viewdata_dow = stat.groupby('Location')['Rating'].mean().reset_index()
viewdata_dow = viewdata_dow.sort_values('Rating', ascending = True).head(10)

print(viewdata_up, '\n' )
print(viewdata_dow)
```

```
            Location  Rating
49      Livermore, CA    4.70
64        Newtown, PA    4.60
11         Camden, NJ    4.50
55     Menlo Park, CA    4.42
47         Lemont, IL    4.30
8         Brooklyn, NY    4.30
0          Addison, TX    4.20
56            Mesa, AZ    4.20
58  Missouri City, TX    4.20
33         Gilbert, AZ    4.20

              Location  Rating
97           Venice, CA    1.60
27      Florham Park, NJ    2.60
60        Naperville, IL    2.60
39           Horsham, PA    2.60
2              Alvin, TX    2.90
35          Glendale, CA    2.90
65        Northbrook, IL    2.98
54      Melrose Park, IL    3.10
13        Carrollton, TX    3.10
74  Rockville Centre, NY    3.10
```

As we did earlier, we have used, seaborn barplot to visually represent the data:



Easy Apply

Analysing and visualizing the effect of whether or not data scientist jobs are easy to apply for in comparison to other jobs.

```
dataview = data.groupby('Easy Apply')['Job Title'].count().reset_index()
dataview
```

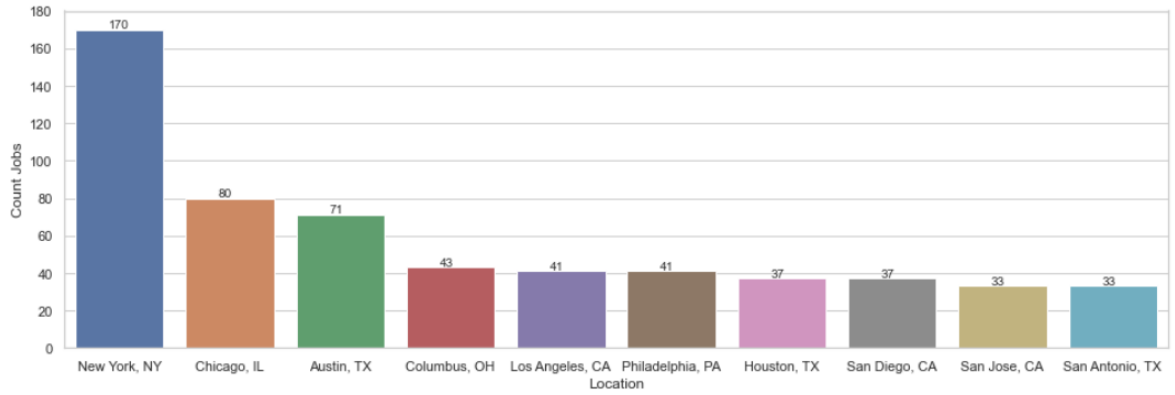
	Easy Apply	Job Title
0	FALSE	831
1	TRUE	42

```
fig, ax = plt.subplots()
ax =sns.barplot(ax = ax, data = dataview, x = 'Easy Apply', y = 'Job Title' )
ax.set_title('Easy Apply Data Science Job')
ax.set_ylabel('Counts of Jobs')
plt.show()
```

Location

List of the Top Ten Locations where you can find jobs as a data scientist/analyst/visualiser:

```
viewdata = stat.groupby('Location')['Job Title'].count().reset_index()
viewdata = viewdata.sort_values('Job Title', ascending = False).head(10)

fig, ax = polt.subplots(figsize = (16,5))
sbn.barplot(data = viewdata, x = 'Location', y = 'Job Title', ax = ax)
ax.set_ylabel('Count Jobs')
ax.set_yticks(nu.arange(0, 200, step = 20))
for index,viewdata in enumerate(viewdata['Job Title'].astype(int)):
    ax.text(x=index-0.1 , y =viewdata+1 , s=f"{viewdata}" , fontdict=dict(fontsize=10))
polt.show()
```



Revenue

Job numbers for data scientists, based on revenue analysis and visualisation.

```
stat['Revenue'].unique().tolist()
```

```
viewdata = stat.copy()
viewdata['Revenue'].replace(['Unknown / Non-Applicable'], nu.nan, inplace = True)
viewdata['Revenue'].dropna(axis = 0, inplace = True)
viewdata = viewdata.groupby('Revenue')['Job Title'].count().reset_index()
viewdata.sort_values('Job Title', ascending = False, inplace = True)
viewdata
```

	Revenue	Job Title
3	$10+ billion (USD)	261
4	100to500 million (USD)	79
7	5to10 billion (USD)	50
2	10to25 million (USD)	47
1	1to5 million (USD)	43
0	1to2 billion (USD)	40
6	25to50 million (USD)	39
5	2to5 billion (USD)	35
8	5to10 million (USD)	27
9	50to100 million (USD)	23
10	500millionto1 billion (USD)	20
11	Less than $1 million (USD)	20

```
max_width = 15
fig, ax = polt.subplots(figsize = (16,4))
sbn.barplot(ax = ax, stat = viewdata, x = 'Revenue', y = 'Job Title', palette = 'deep')
ax.set_title('Count Job Based Revenue')
ax.set_ylabel('Count Jobs')
ax.set_xticklabels(textwrap.fill(x.get_text(), max_width) for x in ax.get_xticklabels())
for index,viewdata in enumerate(viewdata['Job Title'].astype(int)):
    ax.text(x=index - 0.1 , y = viewdata + 1 , s = f"{viewdata}" , fontdict = dict(fontsize = 12))
polt.show()
```

Conclusion

To understand the meaning and implications of data, data visualisation provides a visual context through map or graph representations. It makes the data easier to understand by the human mind, and thus it makes it much easier to recognize trends, patterns, and outliers among vast data sets.

References

[1] F. Malik, “Why Should We Use NumPy?,” FinTechExplained, 31 March 2019. [Online]. Available: https://medium.com/fintechexplained/why-should-we-use-numpy-c14a4fb03ee9.

[2] M. A. Hossen, “Top Python Libraries: Numpy & Pandas,” Towards Data Science, 16 November 2019. [Online]. Available: https://towardsdatascience.com/top-python-libraries-numpy-pandas-8299b567d955.

[3] B. Kumar, “What is Matplotlib and how to use it in Python,” Python Guide, 6 August 2021. [Online]. Available: https://pythonguides.com/what-is-matplotlib/.

[4] K. Katari, “Seaborn: Python,” Towards Data Science, 11 August 2020. [Online]. Available: https://towardsdatascience.com/seaborn-python-8563c3d0ad41.

[5] GeeksforGeeks, “Textwrap – Text wrapping and filling in Python,” 31 May 2017. [Online]. Available: https://www.geeksforgeeks.org/textwrap-text-wrapping-filling-python/#:~:text=The%20textwrap%20module%20can%20be,breaks%20in%20the%20input%20paragraph..

[6] G. Geeks, “Generating Word Cloud in Python,” 5 July 2021. [Online]. Available: https://www.geeksforgeeks.org/generating-word-cloud-python/#:~:text=Word%20Cloud%20is%20a%20data,data%20from%20social%20network%20websites..

[7] R. A. Saputra, “Data Analysis and Visualization on Data Scientist Jobs,” 13 December 2020. [Online]. Available: https://github.com/rifkyahmadsaputra/Data-Analysis-and-Visualization-on-Data-Scientist-Jobs/blob/master/README.md.