Data Science Job Analysis – US

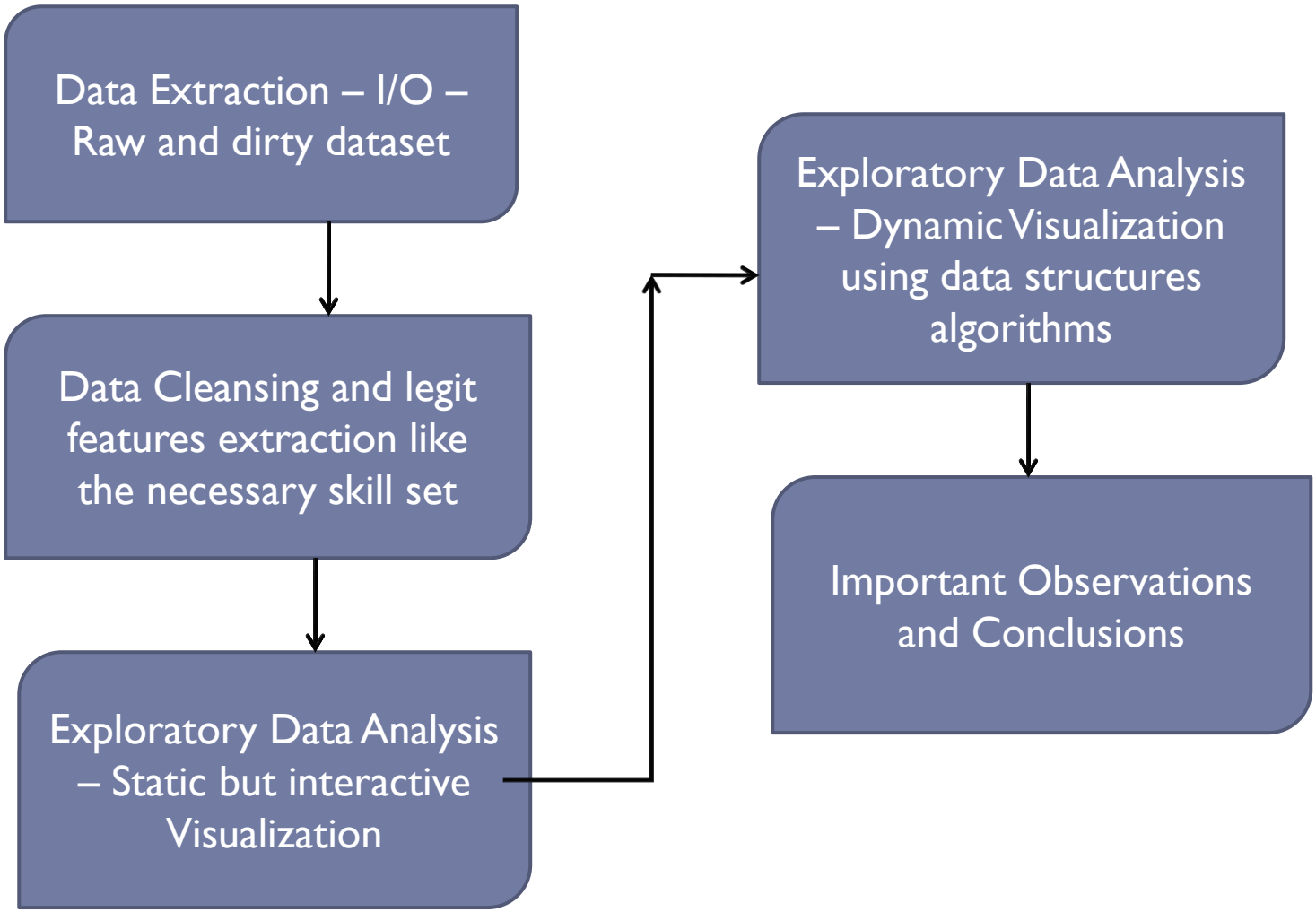
Navigating Opportunities in a Growing Field

~Pragati Mangalsing Patil
234161019

Motivation

- High Demand
- Continuous Learning
- Innovation and Impact
- Diverse Opportunities
- Of course, Competitive Salaries 😋

Block Diagram



Pseudo-Code

1. Import required libraries
2. Read Data from the .csv file – initial data very dirty
3. Cleanse the data – remove the redundant, null values, replace the anonymous characters/words by appropriate values
4. Input – choice code to explore various data visualizing techniques
5. While(input != -1):
    1. Extraction of skills(words) from the lists of lists and appending to one common list
    2. Plot interactive graphs using the libraries against necessary attributes
    3. Implement Linear and Binary Search algorithms to visualize the data dynamically
6. Output – Print observations and conclusions

Code Snippets

```
def read_dataset():

    df = pd.read_csv("jobs.csv", encoding="utf-16").drop(['ID'], axis=1).drop_duplicates()
    #Reading the csv file and dropping the duplicates while reading itself for further processing and cleaning!
    print("The dimensions of the imported dataset::",df.shape)
    return df
```

Cleaning dataset

```
def cleanse_dataset(df):
    #The dataset has too many NULL values. Dropping those especially from the very important attributes - Low, Mean and High Salaries,
    #since these are of utmost importance for data professionals like us!!
    df=df.dropna(subset=['Low_Salary','High_Salary','Mean_Salary'])
    #Mean Salary is very much to be taken care of and we, the data professionals cannot accept salaries in negative!
    #Hence, removing the least significant mean_salary values including the negatives.
    main_label = 'Mean_Salary'
    df[main_label] = df[main_label]*1e-3
    df = df[df[main_label]>0]
    df[main_label] = df[main_label]*1e+3 #Setting back to original decimal places
    #Concatenating city and state
    df['City'] = (df['City'] + ', ' + df['State'])
    #Filling nan values
    df['Profile'] = df['Profile'].fillna('None')
    df['City'] = df['City'].fillna('Remote')
    df['Remote'] = df['Remote'].fillna('On-Site')
    #The dataset seems to be a bit in spanish, so converting the calendrical spanish values to English...!
    df['Frecuency_Salary'] = df['Frecuency_Salary'].replace(['año','hora','mes', 'día', 'semana'],['annum','hour','month','day','week'])

    return df
```

Cleaning/extracting the skills from the 'skills' attribute

```python
def skills_extract(df):
    #The Skills column seems a list of lists...A very dirty dataset indeed!
    #Hereby extracting every skill value and appending to a separate list 'l2' that can be further processed.
    l1=list()
    for i in range (0,len(list(df['Skills']))):
        l1.append((list(df['Skills'])[i][2:-2].replace('\'','').split(',')))
    l2=list()
    for i in range(len(l1)):
        for j in range(len(l1[i])):
            l2.append(l1[i][j])

    return l2
```

Interactive plots using plotly and Seaborn →

```python
dfp = df['Jobs_Group'].value_counts().head(20).sort_values(ascending = True).reset_index()
dfl = df['City'].value_counts().head(20).sort_values(ascending = True).reset_index()
dfc = df['Company'].value_counts().head(20).sort_values(ascending = True).reset_index()

fig = go.Figure()
fig.add_trace(go.Bar(x = dfp['count'],
                     y = dfp['Jobs_Group'],
                     orientation='h',
                     name = 'Job_Position',
                     marker = dict(color = 'Green')))

fig.add_trace(go.Bar(x = dfl['count'],
                     y = dfl['City'],
                     orientation='h',
                     name = 'Location',
                     marker = dict(color = 'SeaGreen')))

fig.add_trace(go.Bar(x = dfc['count'],
                     y = dfc['Company'],
                     orientation='h',
                     name = 'Company',
                     marker = dict(color = 'darkgreen')))
```

Matplotlib to plot line graph(company v/s salary) and implementing **linear search visually** to highlight the salary paid by the given company name--

```
# Function to perform linear search and visualize the relationship between companies and salaries using a line graph
def linear_search_and_visualize_line_chart(df,company_name):

    print("--------------------This Function is the centre of interest for the entire Analytics-----------------------")
    df = df.head(30)
    lc = list(df['Company'])
    ls = list(df['High_Salary'])
    fig, ax = plt.subplots()

    ax.plot(lc, ls, marker='o', linestyle='-', color='lightblue')
    ax.set_xticks(lc)
    ax.set_xticklabels(lc, rotation=35, ha='right',fontsize=7)
    ax.set_ylabel('Salary')
    ax.set_title('Company vs Salary Visualization - Linear Search')
    ax.grid(True)
    found = False

    for i, name in enumerate(lc):
        if name == company_name:
            ax.plot(name, ls[i], marker='o', markersize=8, color='red', label='Found')
            ax.text(name, ls[i] + 2000, f'{ls[i]}', ha='center', va='bottom', color='black')
            ax.vlines(name, 0, ls[i], linestyles='dashed', color='orange', alpha=0.5)
            found = True
        else:
            ax.plot(name, ls[i], marker='o', linestyle='', color='blue')

        #this display.display gives the visual effect/animation to the plot
        plt.show(block=False)
        plt.pause(0.01)
```

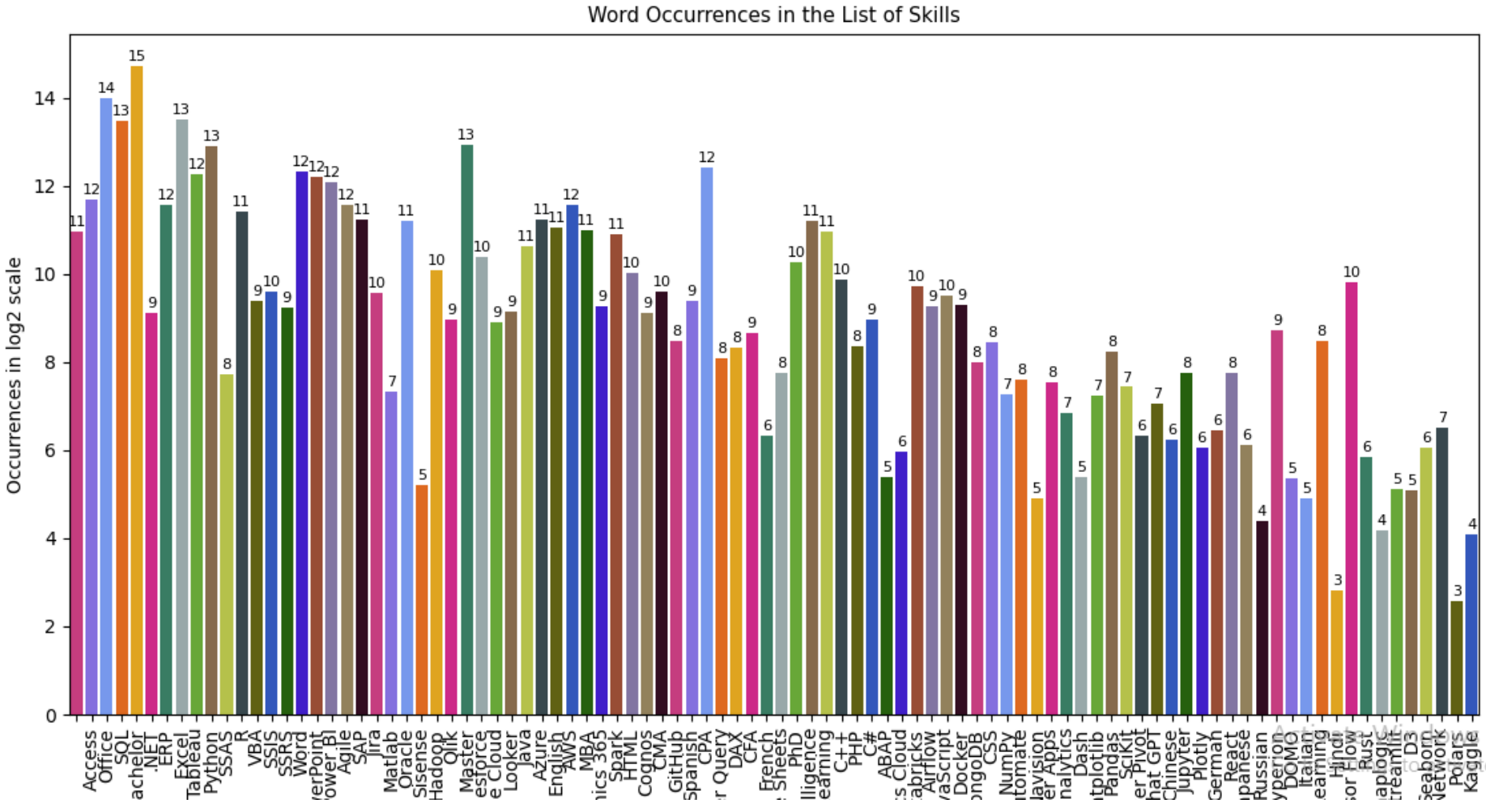
Matplotlib to plot line graph(company v/s salary) and implementing **binary search visually** to highlight the salary paid by the given company name--

```
# Function to perform binary search and visualize the relationship between companies and salaries using a line graph
def binary_search_and_visualize_line_chart(df,company_name):
```

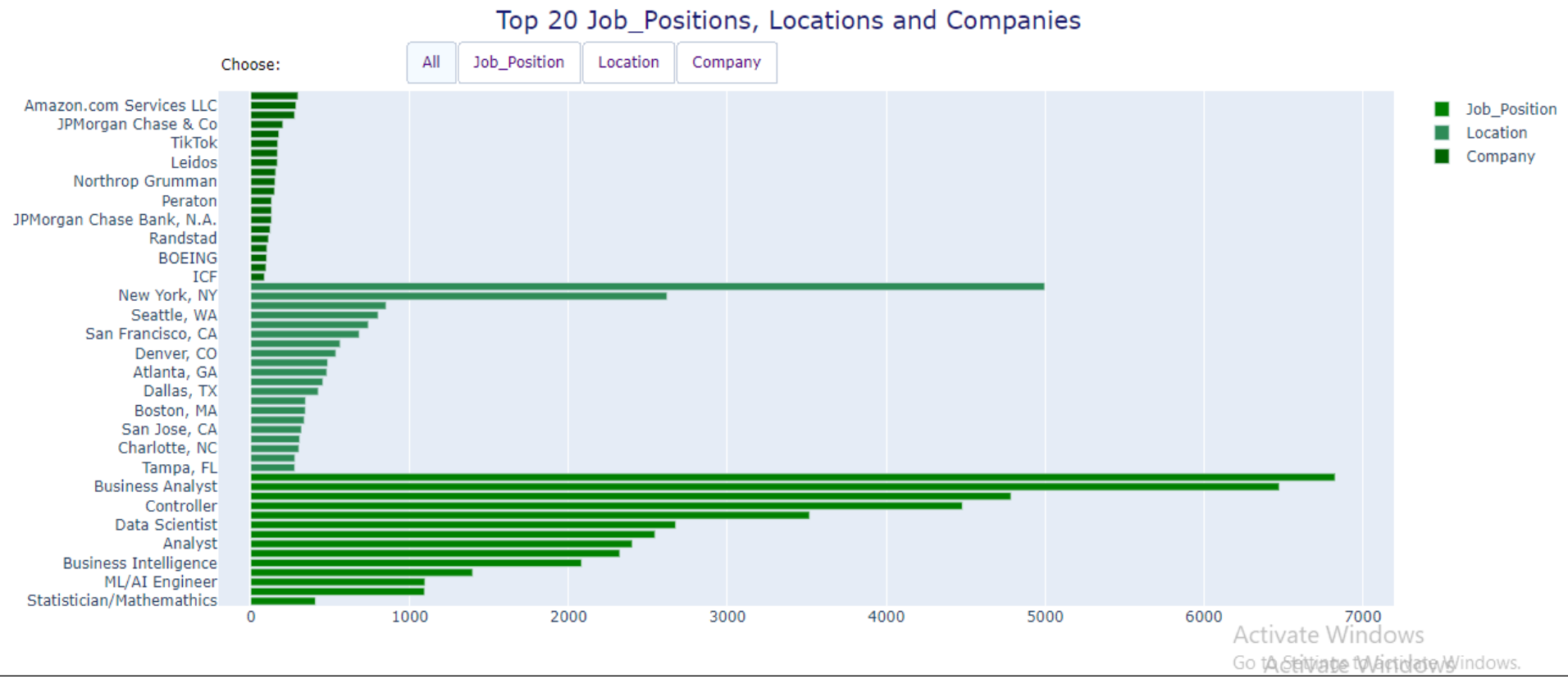
```
    left, right = 0, len(lc) - 1
    sequence_number = 1  # Initialize sequence number
    while left <= right:
        mid = (left + right) // 2
        # Highlight the current dot
        ax.plot(lc[mid], ls[mid], marker='o', markersize=10, color='green', label='Current Search')

        # Print the sequence number on top of the highlighted dot
        ax.text(lc[mid], ls[mid] + 2000, str(sequence_number), ha='center', va='top', color='white', fontsize=8)

        if lc[mid] == company_name:
            # Highlight the final result with a different color
            ax.plot(lc[mid], ls[mid], marker='D', markersize=10, color='black', label='Final Result')
            ax.text(lc[mid], ls[mid] + 2000, f'{ls[mid]}', ha='center', va='bottom', color='black')
            ax.vlines(lc[mid], 0, ls[mid], linestyles='dashed', color='darkgreen', alpha=0.5)
            found = True
            break
        elif lc[mid] < company_name:
            left = mid + 1
        else:
            right = mid - 1

        # Increment the sequence number for the next iteration
        sequence_number += 1

        # Update the plot
        plt.show(block=False)
        plt.pause(1.5)
        #display.display(fig)
        #display.clear_output(wait=True)
```

Results

Skills most required for Data Professionals like us..Excel, Power BI, Office, SQL, Tensor Flow etc..

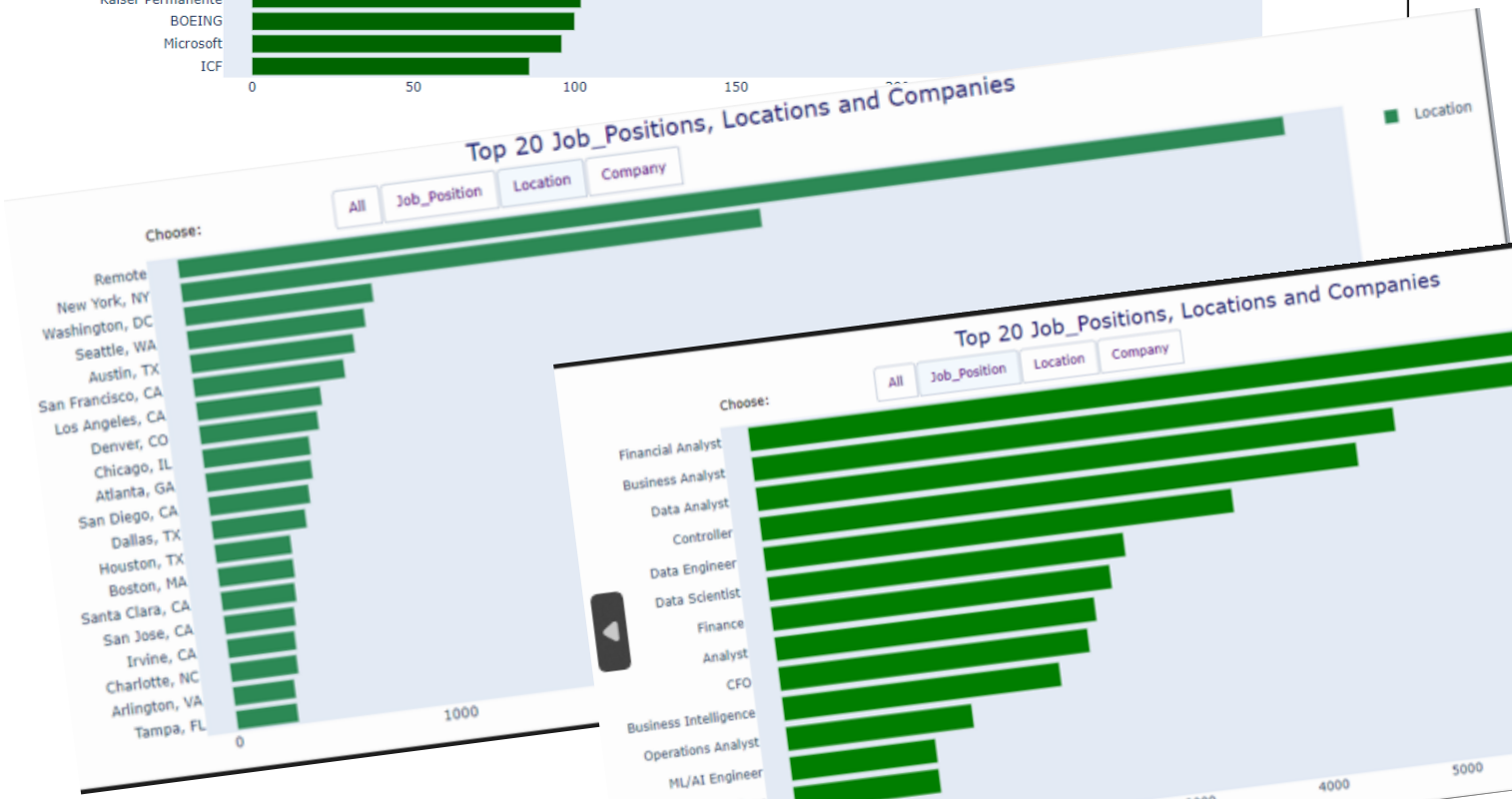


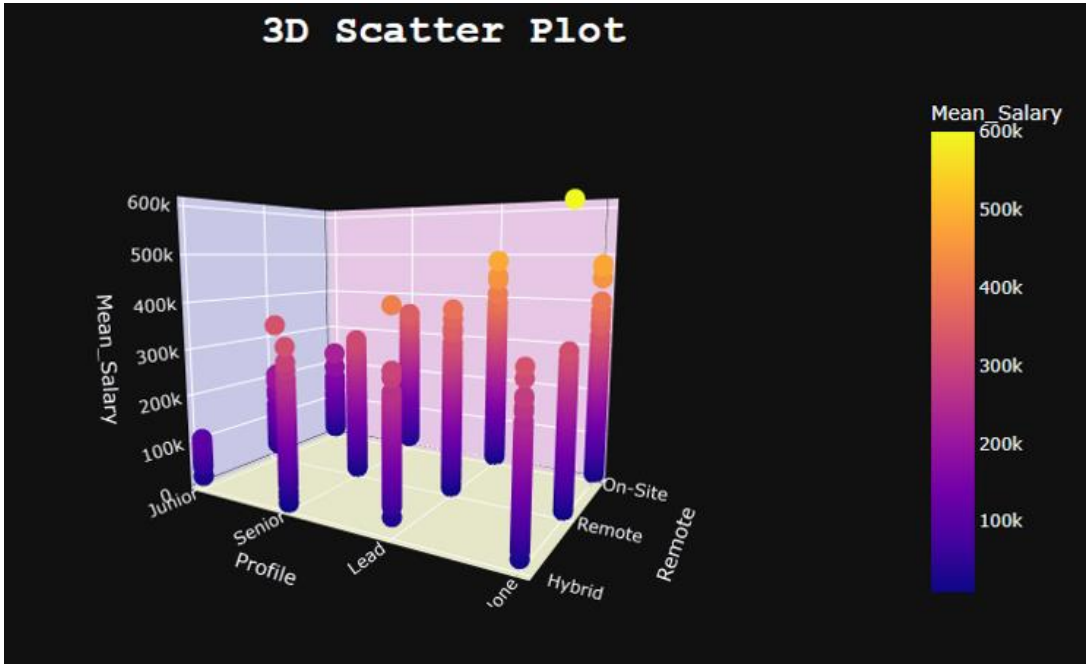
Interactive Plots



When seen closely i.e tab-wise,
"Financial Analyst, Business Analyst, Data Analyst being the TOP 3 job roles, woahhh!!"
And just look at the location---it's Remote at which the most number of data scientists work from!!
Who doesn't love working remotely, coz I do ☺

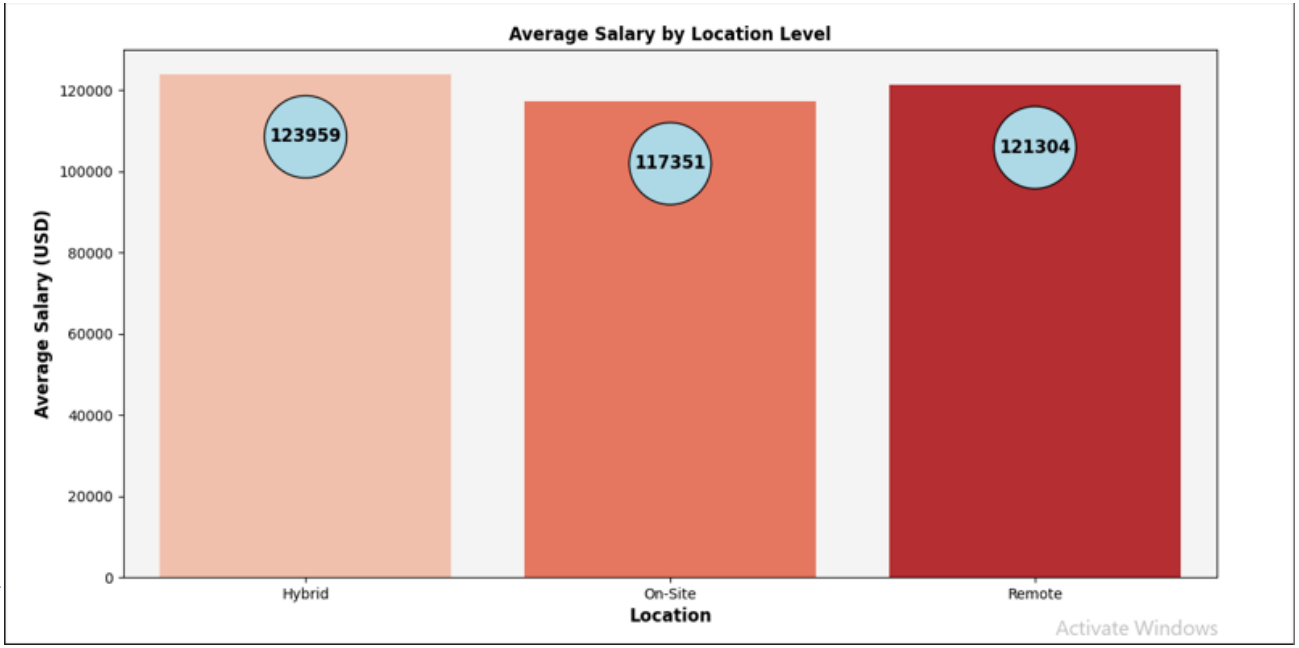
Citi and Amazon being among the top Recruiters…



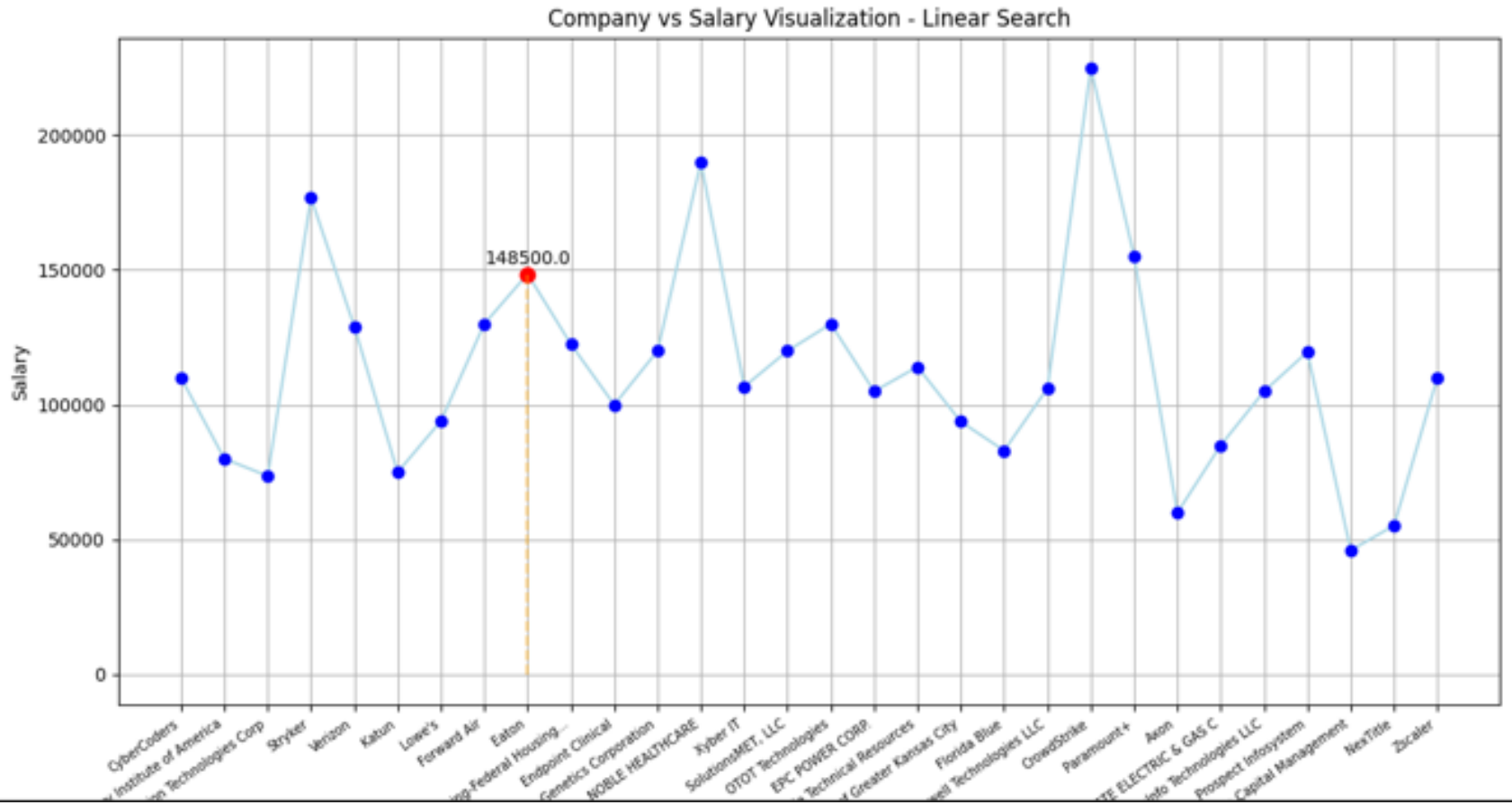


← 3D Scatter using plotly – against Mean Salaries v/s Job Levels v/s Type of mode
At remote location plus a senior level Data Professional
is expected to live a luxurious life!!

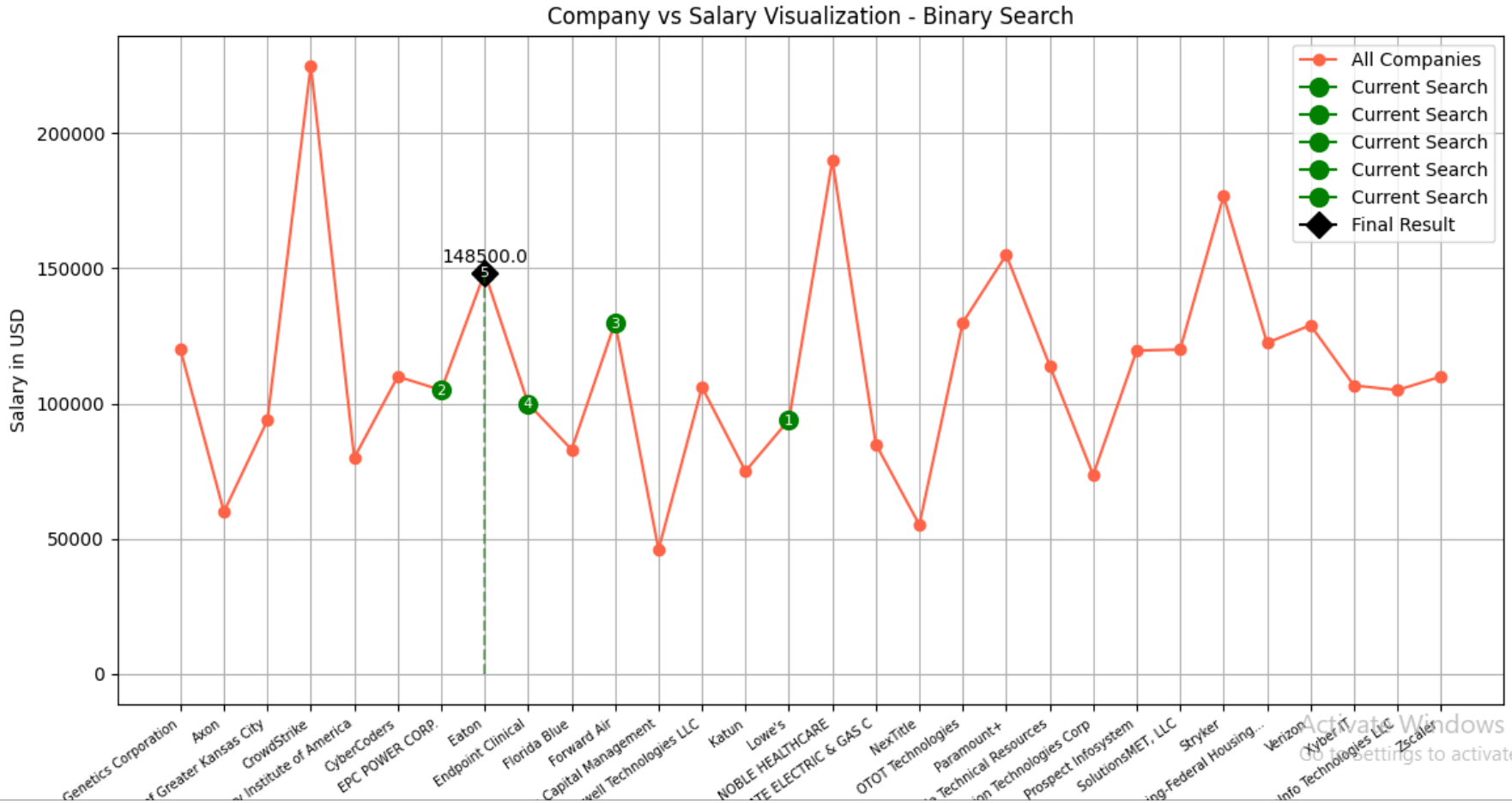
SNS Bar Plot→
"Companies prefer hybrid mode of work to get you paid more on average..."



Heart of the project – linear search implementation shown graphically
For your dream Company v/s Salary it pays



Heart of the project – Binary search implementation shown graphically
For your dream Company v/s Salary it pays



Observations::

Data Science is a great profession to pursue. US offers so many opportunities for data professionals like us with such exciting salaries.
One, who's still thinking, should now finally take the decision to enhance his career in Data Science and GROW!!!

Thank You!!