# Lab 17 Bubble Plots

April 7, 2025

## 1 Bubble Plots

Estimated time needed: **30** minutes

In this lab, you will focus on visualizing data.

The dataset will be directly loaded into pandas for analysis and visualization.

You will use various visualization techniques to explore the data and uncover key trends.

### 1.1 Objectives

In this lab, you will perform the following:

- Visualize the distribution of data.
- Visualize the relationship between two data features.
- Visualize composition of data.
- Visualize comparison of data.

**Setup: Working with the Database  Install and import the needed libraries**

[2]:
```
!pip install pandas
!pip install matplotlib
!pip install seaborn

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
Collecting pandas
  Downloading
pandas-2.2.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(89 kB)
Collecting numpy>=1.26.0 (from pandas)
  Downloading
numpy-2.2.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(62 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
```

```
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-
packages (from pandas) (2024.2)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading
pandas-2.2.3-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.7
MB)
                         12.7/12.7 MB
172.4 MB/s eta 0:00:00
Downloading
numpy-2.2.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (16.1 MB)
                         16.1/16.1 MB
190.6 MB/s eta 0:00:00
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: tzdata, numpy, pandas
Successfully installed numpy-2.2.4 pandas-2.2.3 tzdata-2025.2
Collecting matplotlib
  Downloading matplotlib-3.10.1-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (11 kB)
Collecting contourpy>=1.0.1 (from matplotlib)
  Downloading contourpy-1.3.1-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.4 kB)
Collecting cycler>=0.10 (from matplotlib)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
Collecting fonttools>=4.22.0 (from matplotlib)
  Downloading fonttools-4.57.0-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl.metadata (102 kB)
Collecting kiwisolver>=1.3.1 (from matplotlib)
  Downloading kiwisolver-1.4.8-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.2 kB)
Requirement already satisfied: numpy>=1.23 in /opt/conda/lib/python3.12/site-
packages (from matplotlib) (2.2.4)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (24.2)
Collecting pillow>=8 (from matplotlib)
  Downloading pillow-11.1.0-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (9.1
kB)
Collecting pyparsing>=2.3.1 (from matplotlib)
  Downloading pyparsing-3.2.3-py3-none-any.whl.metadata (5.0 kB)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.12/site-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
Downloading
matplotlib-3.10.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
```

```
(8.6 MB)
                           8.6/8.6 MB
179.0 MB/s eta 0:00:00
Downloading
contourpy-1.3.1-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (323
kB)
Downloading cycler-0.12.1-py3-none-any.whl (8.3 kB)
Downloading fonttools-4.57.0-cp312-cp312-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_6
4.whl (4.9 MB)
                           4.9/4.9 MB
167.3 MB/s eta 0:00:00
Downloading
kiwisolver-1.4.8-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.5
MB)
                           1.5/1.5 MB
89.6 MB/s eta 0:00:00
Downloading pillow-11.1.0-cp312-cp312-manylinux_2_28_x86_64.whl (4.5 MB)
                           4.5/4.5 MB
144.0 MB/s eta 0:00:00
Downloading pyparsing-3.2.3-py3-none-any.whl (111 kB)
Installing collected packages: pyparsing, pillow, kiwisolver, fonttools, cycler,
contourpy, matplotlib
Successfully installed contourpy-1.3.1 cycler-0.12.1 fonttools-4.57.0
kiwisolver-1.4.8 matplotlib-3.10.1 pillow-11.1.0 pyparsing-3.2.3
Collecting seaborn
  Downloading seaborn-0.13.2-py3-none-any.whl.metadata (5.4 kB)
Requirement already satisfied: numpy!=1.24.0,>=1.20 in
/opt/conda/lib/python3.12/site-packages (from seaborn) (2.2.4)
Requirement already satisfied: pandas>=1.2 in /opt/conda/lib/python3.12/site-
packages (from seaborn) (2.2.3)
Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in
/opt/conda/lib/python3.12/site-packages (from seaborn) (3.10.1)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(1.3.1)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.12/site-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(4.57.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(1.4.8)
Requirement already satisfied: packaging>=20.0 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(24.2)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.12/site-
```

```
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in
/opt/conda/lib/python3.12/site-packages (from matplotlib!=3.6.1,>=3.4->seaborn)
(2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-
packages (from pandas>=1.2->seaborn) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.12/site-
packages (from pandas>=1.2->seaborn) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.17.0)
Downloading seaborn-0.13.2-py3-none-any.whl (294 kB)
Installing collected packages: seaborn
Successfully installed seaborn-0.13.2
```

**Download and connect to the database file containing survey data.**

To start, download and load the dataset into a `pandas` DataFrame.

[3]:
```
file_path = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
↪n01PQ9pSmiRX6520flujwQ/survey-data.csv"

df = pd.read_csv(file_path)
df.tail()
```

[3]:

| | ResponseId | MainBranch | Age | \ |
|---|---|---|---|---|
| 65432 | 65433 | I am a developer by profession | 18-24 years old | |
| 65433 | 65434 | I am a developer by profession | 25-34 years old | |
| 65434 | 65435 | I am a developer by profession | 25-34 years old | |
| 65435 | 65436 | I am a developer by profession | 18-24 years old | |
| 65436 | 65437 | I code primarily as a hobby | 18-24 years old | |

| | Employment | RemoteWork | Check | \ |
|---|---|---|---|---|
| 65432 | Employed, full-time | Remote | Apples | |
| 65433 | Employed, full-time | Remote | Apples | |
| 65434 | Employed, full-time | In-person | Apples | |
| 65435 | Employed, full-time | Hybrid (some remote, some in-person) | Apples | |
| 65436 | Student, full-time | NaN | Apples | |

| | CodingActivities | \ |
|---|---|---|
| 65432 | Hobby;School or academic work | |
| 65433 | Hobby;Contribute to open-source projects | |
| 65434 | Hobby | |
| 65435 | Hobby;Contribute to open-source projects;Profe… | |
| 65436 | NaN | |

```
                                                 EdLevel  \
65432       Bachelor's degree (B.A., B.S., B.Eng., etc.)
65433                                                NaN
65434       Bachelor's degree (B.A., B.S., B.Eng., etc.)
65435  Secondary school (e.g. American high school, G…
65436                                                NaN

                                                LearnCode  \
65432  On the job training;School (i.e., University, …
65433                                                NaN
65434  Other online resources (e.g., videos, blogs, f…
65435  On the job training;Other online resources (e…
65436                                                NaN

                                      LearnCodeOnline  … JobSatPoints_6  \
65432                                             NaN  …            NaN
65433                                             NaN  …            NaN
65434  Technical documentation;Stack Overflow;Social …  …            NaN
65435  Technical documentation;Blogs;Written Tutorial…  …            0.0
65436                                             NaN  …            NaN

      JobSatPoints_7 JobSatPoints_8 JobSatPoints_9 JobSatPoints_10  \
65432            NaN            NaN            NaN             NaN
65433            NaN            NaN            NaN             NaN
65434            NaN            NaN            NaN             NaN
65435            0.0            0.0            0.0             0.0
65436            NaN            NaN            NaN             NaN

      JobSatPoints_11 SurveyLength SurveyEase ConvertedCompYearly JobSat
65432             NaN          NaN        NaN                 NaN    NaN
65433             NaN          NaN        NaN                 NaN    NaN
65434             NaN          NaN        NaN                 NaN    NaN
65435             0.0          NaN        NaN                 NaN    NaN
65436             NaN          NaN        NaN                 NaN    NaN

[5 rows x 114 columns]
```

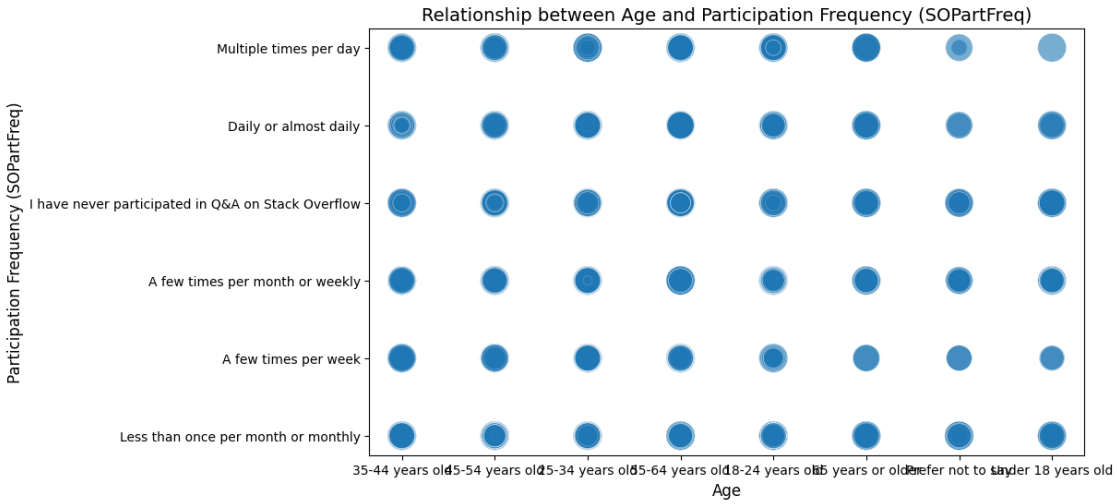
### 1.1.1 Task 1: Exploring Data Distributions Using Bubble Plots

**1. Bubble Plot for Age vs. Frequency of Participation**

- Visualize the relationship between respondents' age and their participation frequency (`SOPartFreq`) using a bubble plot.
- Use the size of the bubbles to represent their job satisfaction (`JobSat`).

[4]:
```
df['JobSat'] = pd.to_numeric(df['JobSat'], errors='coerce')
```

```
df = df.dropna(subset=['JobSat', 'Age', 'SOPartFreq'])

df = df[df['JobSat'] > 0]

plt.figure(figsize=(10, 6))
plt.scatter(df['Age'], df['SOPartFreq'], s=df['JobSat']*50, alpha=0.6,␣
 ↪edgecolors="w", linewidth=0.5)
plt.title('Relationship between Age and Participation Frequency (SOPartFreq)',␣
 ↪fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Participation Frequency (SOPartFreq)', fontsize=12)
plt.show()
```




**2. Bubble Plot for Compensation vs. Job Satisfaction** -Visualize the relationship between yearly compensation (`ConvertedCompYearly`) and job satisfaction (`JobSat`).

- Use the size of the bubbles to represent respondents' age.

[5]:
```
print(df[['ConvertedCompYearly', 'JobSat', 'Age']].isnull().sum())

df = df.dropna(subset=['ConvertedCompYearly', 'JobSat', 'Age'])

print(df[['ConvertedCompYearly', 'JobSat', 'Age']].isnull().sum())

print(df[['ConvertedCompYearly', 'JobSat', 'Age']].describe())

df['AgeScaled'] = df['Age'].apply(lambda x: max(min(x * 10, 1000), 10))
df['CompScaled'] = df['ConvertedCompYearly'].apply(lambda x: max(min(x / 1000,␣
 ↪1000), 10))
```

```
plt.figure(figsize=(10, 6))
plt.scatter(df['CompScaled'], df['JobSat'], s=df['AgeScaled'], alpha=0.6,␣
 ↪edgecolors="w", linewidth=0.5)
plt.title('Relationship between Yearly Compensation and Job Satisfaction',␣
 ↪fontsize=14)
plt.xlabel('Yearly Compensation (ConvertedCompYearly)', fontsize=12)
plt.ylabel('Job Satisfaction (JobSat)', fontsize=12)
plt.show()
```

```
ConvertedCompYearly    10342
JobSat                     0
Age                        0
dtype: int64
ConvertedCompYearly    0
JobSat                 0
Age                    0
dtype: int64
       ConvertedCompYearly        JobSat
count         1.280500e+04  12805.000000
mean          8.878664e+04      7.025849
std           1.832554e+05      1.978888
min           1.000000e+00      1.000000
25%           3.600000e+04      6.000000
50%           6.905800e+04      7.000000
75%           1.114170e+05      8.000000
max           1.381802e+07     10.000000
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[5], line 9
      5 print(df[['ConvertedCompYearly', 'JobSat', 'Age']].isnull().sum())
      7 print(df[['ConvertedCompYearly', 'JobSat', 'Age']].describe())
----> 9 df['AgeScaled'] = df['Age'].apply(lambda x: max(min(x * 10, 1000), 10))
     10 df['CompScaled'] = df['ConvertedCompYearly'].apply(lambda x: max(min(x ␣
 ↪1000, 1000), 10))
     12 plt.figure(figsize=(10, 6))

File /opt/conda/lib/python3.12/site-packages/pandas/core/series.py:4924, in␣
 ↪Series.apply(self, func, convert_dtype, args, by_row, **kwargs)
   4789 def apply(
   4790     self,
   4791     func: AggFuncType,
   (…)
   4796     **kwargs,
   4797 ) -> DataFrame | Series:
   4798     """
   4799     Invoke function on values of Series.
```

```
   4800
   (…)
   4915     dtype: float64
   4916     """
   4917     return SeriesApply(
   4918         self,
   4919         func,
   4920         convert_dtype=convert_dtype,
   4921         by_row=by_row,
   4922         args=args,
   4923         kwargs=kwargs,
-> 4924     ).apply()

File /opt/conda/lib/python3.12/site-packages/pandas/core/apply.py:1427, in␣
 ↪SeriesApply.apply(self)
   1424     return self.apply_compat()
   1426 # self.func is Callable
-> 1427 return self.apply_standard()

File /opt/conda/lib/python3.12/site-packages/pandas/core/apply.py:1507, in␣
 ↪SeriesApply.apply_standard(self)
   1501 # row-wise access
   1502 # apply doesn't have a `na_action` keyword and for backward compat␣
 ↪reasons
   1503 # we need to give `na_action="ignore"` for categorical data.
   1504 # TODO: remove the `na_action="ignore"` when that default has been␣
 ↪changed in
   1505 #  Categorical (GH51645).
   1506 action = "ignore" if isinstance(obj.dtype, CategoricalDtype) else None
-> 1507 mapped = obj._map_values(
   1508     mapper=curried, na_action=action, convert=self.convert_dtype
   1509 )
   1511 if len(mapped) and isinstance(mapped[0], ABCSeries):
   1512     # GH#43986 Need to do list(mapped) in order to get treated as neste
   1513     #  See also GH#25959 regarding EA support
   1514     return obj._constructor_expanddim(list(mapped), index=obj.index)

File /opt/conda/lib/python3.12/site-packages/pandas/core/base.py:921, in␣
 ↪IndexOpsMixin._map_values(self, mapper, na_action, convert)
    918 if isinstance(arr, ExtensionArray):
    919     return arr.map(mapper, na_action=na_action)
--> 921 return␣
 ↪algorithms.map_array(arr, mapper, na_action=na_action, convert=convert)

File /opt/conda/lib/python3.12/site-packages/pandas/core/algorithms.py:1743, in␣
 ↪map_array(arr, mapper, na_action, convert)
   1741 values = arr.astype(object, copy=False)
   1742 if na_action is None:
```

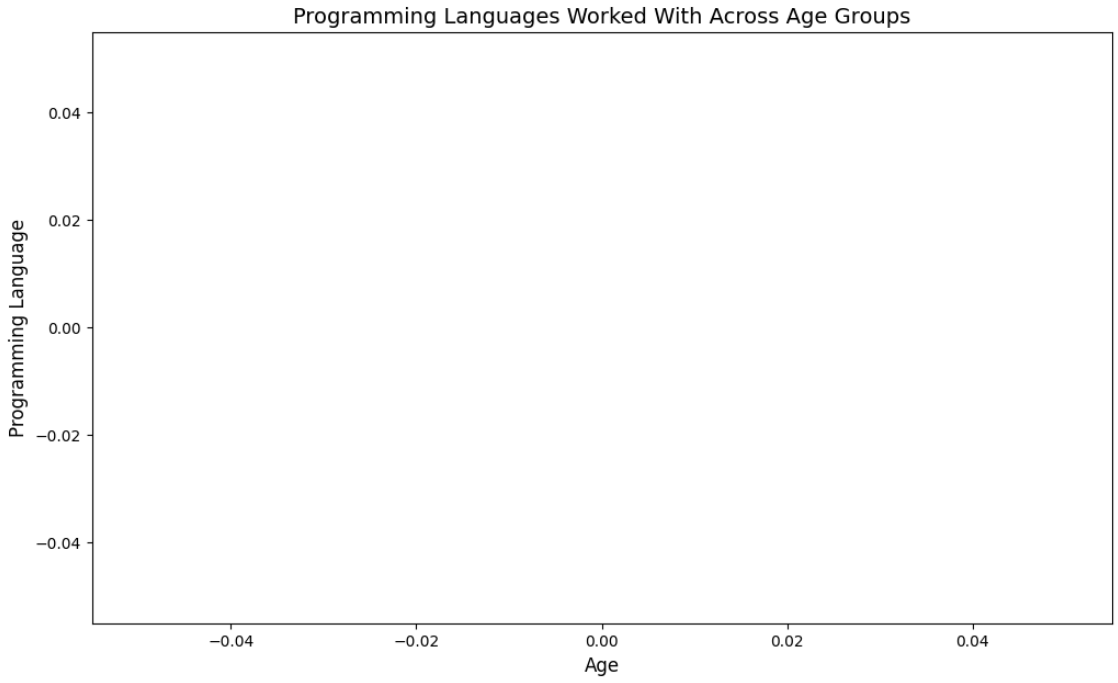
```
-> 1743     return lib.map_infer(values, mapper, convert=convert)
   1744 else:
   1745     return lib.map_infer_mask(
   1746         values, mapper, mask=isna(values).view(np.uint8), convert=convert
   1747     )

File lib.pyx:2972, in pandas._libs.lib.map_infer()

Cell In[5], line 9, in <lambda>(x)
      5 print(df[['ConvertedCompYearly', 'JobSat', 'Age']].isnull().sum())
      7 print(df[['ConvertedCompYearly', 'JobSat', 'Age']].describe())
----> 9 df['AgeScaled'] = df['Age'].apply(lambda x: max(min(x * 10, 1000), 10))
     10 df['CompScaled'] = df['ConvertedCompYearly'].apply(lambda x: max(min(x
 ↪1000, 1000), 10))
     12 plt.figure(figsize=(10, 6))

TypeError: '<' not supported between instances of 'int' and 'str'
```

### 1.1.2 Task 2: Analyzing Relationships Using Bubble Plots

#### 1. Bubble Plot of Technology Preferences by Age

- Visualize the popularity of programming languages respondents have worked with (`LanguageHaveWorkedWith`) across age groups.
- Use bubble size to represent the frequency of each language.

[6]:
```
df = df.dropna(subset=['LanguageHaveWorkedWith', 'Age'])

df['LanguageHaveWorkedWith'] = df['LanguageHaveWorkedWith'].astype(str)
df['Age'] = pd.to_numeric(df['Age'], errors='coerce')
df = df.dropna(subset=['Age'])

lang_age_pairs = df.explode('LanguageHaveWorkedWith')
lang_age_pairs['LanguageHaveWorkedWith'] =␣
 ↪lang_age_pairs['LanguageHaveWorkedWith'].str.split(';')
lang_age_pairs = lang_age_pairs.explode('LanguageHaveWorkedWith')

grouped = lang_age_pairs.groupby(['LanguageHaveWorkedWith', 'Age']).size().
 ↪reset_index(name='Count')

plt.figure(figsize=(12, 7))
plt.scatter(grouped['Age'], grouped['LanguageHaveWorkedWith'],␣
 ↪s=grouped['Count']*3, alpha=0.6, edgecolors='w', linewidth=0.5)
plt.title('Programming Languages Worked With Across Age Groups', fontsize=14)
plt.xlabel('Age', fontsize=12)
plt.ylabel('Programming Language', fontsize=12)
plt.show()
```

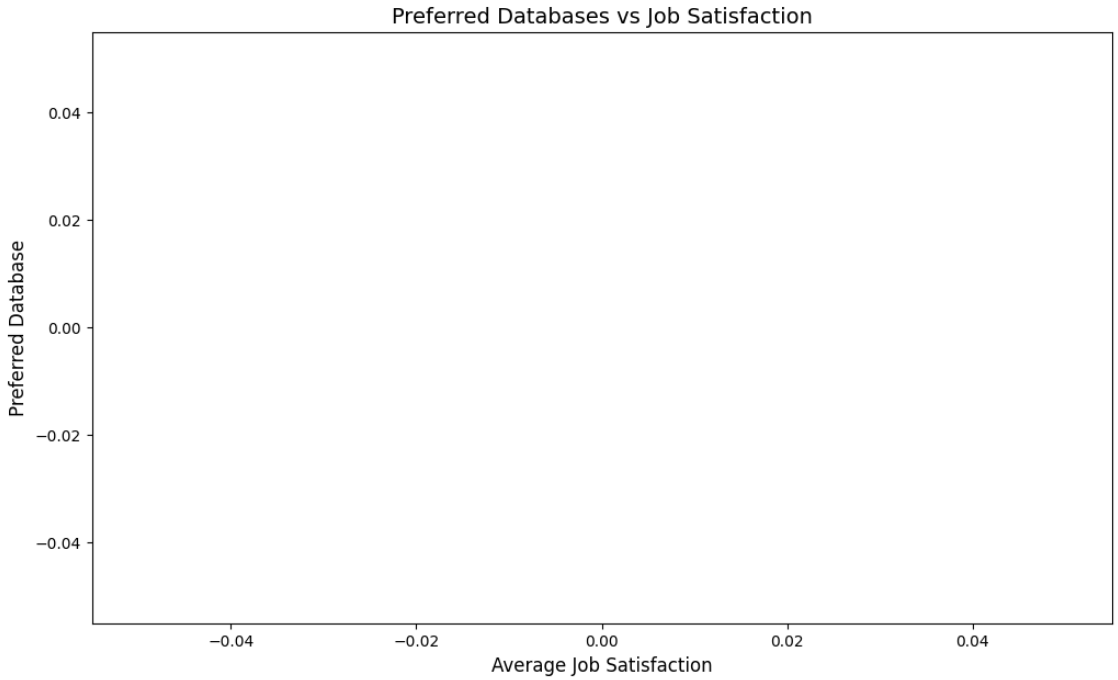



### 2. Bubble Plot for Preferred Databases vs. Job Satisfaction

- Explore the relationship between preferred databases (`DatabaseWantToWorkWith`) and job satisfaction.
- Use bubble size to indicate the number of respondents for each database.

[7]:
```
df = df.dropna(subset=['DatabaseWantToWorkWith', 'JobSat'])

df['DatabaseWantToWorkWith'] = df['DatabaseWantToWorkWith'].astype(str)
df['JobSat'] = pd.to_numeric(df['JobSat'], errors='coerce')
df = df.dropna(subset=['JobSat'])

db_sat = df.copy()
db_sat['DatabaseWantToWorkWith'] = db_sat['DatabaseWantToWorkWith'].str.split(';
 ↪')
db_sat = db_sat.explode('DatabaseWantToWorkWith')

grouped = db_sat.groupby('DatabaseWantToWorkWith').agg({'JobSat': 'mean',␣
 ↪'DatabaseWantToWorkWith': 'count'})
grouped.columns = ['AvgJobSat', 'RespondentCount']
grouped = grouped.reset_index()

plt.figure(figsize=(12, 7))
```

```
plt.scatter(grouped['AvgJobSat'], grouped['DatabaseWantToWorkWith'],␣
 ↪s=grouped['RespondentCount']*3, alpha=0.6, edgecolors='w', linewidth=0.5)
plt.title('Preferred Databases vs Job Satisfaction', fontsize=14)
plt.xlabel('Average Job Satisfaction', fontsize=12)
plt.ylabel('Preferred Database', fontsize=12)
plt.show()
```




### 1.1.3 Task 3: Comparing Data Using Bubble Plots

**1. Bubble Plot for Compensation Across Developer Roles**

- Visualize compensation (`ConvertedCompYearly`) across different developer roles (`DevType`).
- Use bubble size to represent job satisfaction.

[ ]:
```
##Write your code here
```

**2. Bubble Plot for Collaboration Tools by Age**

- Visualize the relationship between the collaboration tools used (`NEWCollabToolsHaveWorkedWith`) and age groups.
- Use bubble size to represent the frequency of tool usage.

[ ]:
```
##Write your code here
```

#### 1.1.4 Task 4: Visualizing Technology Trends Using Bubble Plots

**1. Bubble Plot for Preferred Web Frameworks vs. Job Satisfaction**

- Explore the relationship between preferred web frameworks (`WebframeWantToWorkWith`) and job satisfaction.
- Use bubble size to represent the number of respondents.

```
[ ]: ##Write your code here
```

**2. Bubble Plot for Admired Technologies Across Countries**

- Visualize the distribution of admired technologies (`LanguageAdmired`) across different countries (`Country`).
- Use bubble size to represent the frequency of admiration.

```
[ ]: ##Write your code here
```

### 1.2 Final Step: Review

After completing the lab, you will have extensively used bubble plots to gain insights into developer community preferences, demographics, compensation trends, and job satisfaction.

### 1.3 Summary

After completing this lab, you will be able to:

- Create and interpret bubble plots to analyze relationships and compositions within datasets.
- Use bubble plots to explore developer preferences, compensation trends, and satisfaction levels.
- Apply bubble plots to visualize complex relationships involving multiple dimensions effectively.

### 1.4 Authors:

Ayushi Jain

#### 1.4.1 Other Contributors:

- Rav Ahuja
- Lakshmi Holla
- Malika

<!– ## Change Log |Date (YYYY-MM-DD)|Version|Changed By|Change Description| |-|-|-|-| |2024-10-29|1.2|Madhusudhan Moole|Updated lab| |2024-10-16|1.1|Madhusudhan Moole|Updated lab| |2024-10-15|1.0|Raghul Ramesh|Created lab| –!>