assignment03-pie

October 11, 2025

Pie Chart - Jupyter Notebook When you run this notebook, it will load a CSV file containing people's email addresses, extract their email **domains**, count how many times each occurs, and display the results as a **pie chart**.

Importing the Necessary Libraries To work with the dataset, pandas will be used for handling data and matplotlib for plotting. Pandas provide convenient tools for reading, cleaning, and analyzing tabular data, while matplotlib allows enables the ability to create flexible and informative visualizations.

[75]:
```
%matplotlib inline
import pandas as pd                     # To handle csv data
import matplotlib.pyplot as plt         # To create the pie chart
```

Load the Dataset The next step is to load the file people-1000.csv, which contains the required information. This file includes a column with email addresses that will be analyzed

[76]:
```
csv_path = r"C:\Users\CAD-PC\Desktop\GitHub - Cloned␣
 ↪Repository\PFDA\Assignments\Week-3\people-1000.csv"   # full path to the CSV␣
 ↪file
data = pd.read_csv(csv_path)                      # load data into a pandas␣
 ↪DataFrame
data.head()                                       # preview the first few rows
```

[76]:
```
   Index          User Id First Name Last Name     Sex  \
0      1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1      2  3d5AD30A4cD38ed         Jo    Rivers  Female
2      3  810Ce0F276Badec     Sheryl    Lowery  Female
3      4  BF2a889C00f0cE1    Whitney    Hooper    Male
4      5  9afFEafAe1CBBB9    Lindsey      Rice  Female

                          Email                 Phone Date of birth  \
0             pwarner@example.org          857.139.8239    2014-01-27
1  fergusonkatherine@example.net       +1-950-759-8687    1931-07-26
2             fhoward@example.org          (599)782-0605    2013-11-25
3           zjohnston@example.com      +1-939-130-6258    2012-11-17
4                elin@example.net  (390)417-1635x3010    1923-04-15
```

```
                  Job Title
0         Probation officer
1                    Dancer
2                      Copy
3  Counselling psychologist
4       Biomedical engineer
```

Extracting Email Domains Here, the column that contains email addresses is found. The **domain** part of each email is extracted (everything after the `@` symbol) and store it in a new column called `domain`.

[77]:
```python
email_col = [c for c in data.columns if 'email' in c.lower()][0]  # detect email column
data['domain'] = data[email_col].apply(lambda x: str(x).split('@')[-1])  # extract domain
data.head()  # preview the updated DataFrame
```

[77]:
```
   Index          User Id First Name Last Name     Sex  \
0      1  8717bbf45cCDbEe     Shelia   Mahoney    Male
1      2  3d5AD30A4cD38ed         Jo    Rivers  Female
2      3  810Ce0F276Badec     Sheryl    Lowery  Female
3      4  BF2a889C00f0cE1    Whitney    Hooper    Male
4      5  9afFEafAe1CBBB9    Lindsey      Rice  Female

                           Email               Phone Date of birth  \
0            pwarner@example.org        857.139.8239    2014-01-27
1  fergusonkatherine@example.net     +1-950-759-8687    1931-07-26
2            fhoward@example.org       (599)782-0605    2013-11-25
3          zjohnston@example.com     +1-939-130-6258    2012-11-17
4               elin@example.net  (390)417-1635x3010    1923-04-15

                  Job Title       domain
0         Probation officer  example.org
1                    Dancer  example.net
2                      Copy  example.org
3  Counselling psychologist  example.com
4       Biomedical engineer  example.net
```

Count unique email domains Count how many times each email domain appears using the `value_counts()` function. This gives us a summary of which domains are most common in the dataset.

[78]:
```python
domain_counts = data['domain'].value_counts()   # count occurrences of each domain
domain_counts.head(10)                          # display the top 10 most common domains
```

[78]:

```
domain
example.org    341
example.com    339
example.net    320
Name: count, dtype: int64
```

Plotting the Pie Chart Visualize the distribution of email domains using a pie chart. Each slice of the pie represents a different domain, and the percentages show
how common each one is in the dataset.

[79]:

```
%matplotlib qt
plt.figure(figsize=(6, 6)) # set chart size
plt.pie(
    domain_counts,
    labels=list(domain_counts.index), # domain names
    autopct='%1.1f%%', # percentage format
    startangle=140, # rotate pie for style
    labeldistance=1.1, # push labels outward
    pctdistance=0.5, # move % inward
    textprops={'fontsize': 12, 'fontweight': 'normal', 'color': 'black'}  # %␣
 ↪text colour and style
)
plt.title("Email Domain Distribution", fontsize=18, fontweight='bold')  # bold␣
 ↪title
plt.tight_layout() # adjust spacing
plt.show() # display chart
```

Summary File This notebook read the **people-1000.csv** dataset, extracted email domain names, counted how often each appeared, and plotted the data as a pie chart using matplotlib. The chart makes it easy to see which email providers (like Gmail, Yahoo, or Outlook) are most common among the dataset's users.