Skills
Network

Pie Charts

Estimated time needed: **30** minutes

- In this lab, you will focus on visualizing data.
- The provided dataset will be loaded into pandas for analysis.
- Various pie charts will be created to:
    - Analyze developer preferences.
    - Identify technology usage trends.
- The lab aims to provide insights into key variables using visual representations.

Objectives

In this lab you will perform the following:

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

Setup: Downloading and Loading the Data

Install the libraries

In [1]:
```
!pip install pandas
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.12/site-packages (2.3.0)
Requirement already satisfied: numpy>=1.26.0 in /opt/conda/lib/python3.12/site-packages (from panda
s) (2.3.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.12/site-packages (fr
om pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-packages (from pandas)
(2024.2)
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.12/site-packages (from panda
s) (2025.2)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-packages (from python-date
util>=2.8.2->pandas) (1.17.0)
```

In [2]:
```
!pip install matplotlib
```

```
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.12/site-packages (3.10.3)
Requirement already satisfied: contourpy>=1.0.1 in /opt/conda/lib/python3.12/site-packages (from mat
plotlib) (1.3.2)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.12/site-packages (from matplot
lib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.12/site-packages (from ma
tplotlib) (4.58.4)
Requirement already satisfied: kiwisolver>=1.3.1 in /opt/conda/lib/python3.12/site-packages (from ma
tplotlib) (1.4.8)
Requirement already satisfied: numpy>=1.23 in /opt/conda/lib/python3.12/site-packages (from matplotl
ib) (2.3.0)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.12/site-packages (from matp
lotlib) (24.2)
Requirement already satisfied: pillow>=8 in /opt/conda/lib/python3.12/site-packages (from matplotli
b) (11.2.1)
Requirement already satisfied: pyparsing>=2.3.1 in /opt/conda/lib/python3.12/site-packages (from mat
plotlib) (3.2.3)
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-date
util>=2.7->matplotlib) (1.17.0)
```

In [3]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np # For NaN handling and dummy data
import functools

# Decorator to warn only once for a given message
def warn_once(func):
    warnings = set()

    @functools.wraps(func)
    def wrapper(*args, **kwargs):
        message_to_check = args[0] if args else None

        is_hashable = False
        if message_to_check is not None:
            try:
                hash(message_to_check)
                is_hashable = True
            except TypeError:
                is_hashable = False

        if is_hashable and message_to_check not in warnings:
            warnings.add(message_to_check)
            return func(*args, **kwargs)
        elif not is_hashable:
            return func(*args, **kwargs)
        else:
            pass # Already warned for this hashable message

    return wrapper

original_print = print
@warn_once
def print_once(*args, **kwargs):
    original_print(*args, **kwargs)

print = print_once # Override print for warnings within this script


# --- Setup: Download and Load the Data ---
print("--- Setup: Downloading and Loading the Data ---")

# The PDF specifies downloading 'survey-data.csv'
file_url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pSmiRX6520fluj
local_file_name = 'survey-data.csv'

try:
    df = pd.read_csv(file_url, na_values=['NA', 'N/A', 'nan', 'NaN', 'null', 'Null', '', ' ', '-'])
    print(f"Dataset loaded successfully from: {file_url}")
    print(f"Initial DataFrame shape: {df.shape}")
    print(f"Initial DataFrame columns: {df.columns.tolist()}")
except Exception as e:
    print(f"ERROR: Could not load dataset from URL: {e}")
    print(f"Creating a dummy DataFrame for demonstration purposes as '{local_file_name}' was not fo
    # Create a dummy DataFrame if download fails
    num_rows_dummy = 200
    data = {
        'ResponseId': range(1, num_rows_dummy + 1),
        'DatabaseWantToWorkWith': [';'.join(np.random.choice(['MySQL', 'PostgreSQL', 'MongoDB', 'Re
        'DevType': [';'.join(np.random.choice(['Developer, full-stack', 'Developer, back-end', 'Dev
        'OpSysProfessional use': np.random.choice(['Windows', 'macOS', 'Linux', 'BSD', 'Other'], si
        'LanguageHaveWorkedWith': [';'.join(np.random.choice(['Python', 'JavaScript', 'Java', 'C#',
        'NEWCollabToolsHaveWorkedWith': [';'.join(np.random.choice(['Git', 'Jira', 'Confluence', 'S
        'LanguageAdmired': [';'.join(np.random.choice(['Python', 'Rust', 'Go', 'TypeScript', 'C++',
        'AIToolCurrently Using': [';'.join(np.random.choice(['ChatGPT', 'GitHub Copilot', 'Bard', '
        'WebframeWantToWorkWith': [';'.join(np.random.choice(['React', 'Angular', 'Vue.js', 'Django
        'EmbeddedWantToWorkWith': [';'.join(np.random.choice(['Arduino', 'Raspberry Pi', 'ESP32', '
    }
    df = pd.DataFrame(data)
    print("Dummy DataFrame created and populated with sample data.")

# --- Data Cleaning and Preprocessing for Pie Charts ---
print("\n--- Data Cleaning and Preprocessing ---")

def clean_and_explode_column(dataframe, column_name, top_n=5):
    """
    Cleans a specified column (handles multi-valued entries), explodes it,
    and returns value counts for the top_n items.
    """
    if column_name not in dataframe.columns:
        print(f"WARNING: Column '{column_name}' not found in DataFrame. Skipping cleaning and explo
        return pd.Series(dtype=int) # Return empty series

    df_cleaned = dataframe.copy()

    # Ensure column is string type and handle common NaN representations
    df_cleaned[column_name] = df_cleaned[column_name].astype(str).str.strip()
    df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)

    # Drop rows where the specific column is NaN before splitting/exploding
    df_cleaned.dropna(subset=[column_name], inplace=True)

    if df_cleaned.empty:
        print(f"WARNING: After dropping NaNs, '{column_name}' has no data. Cannot explode or count.
        return pd.Series(dtype=int)

    # Split by semicolon (if it's a multi-valued column) and explode
    # Check if any entry contains ';' before splitting
    if df_cleaned[column_name].str.contains(';', na=False).any():
        df_cleaned[column_name] = df_cleaned[column_name].str.split(';')
        df_exploded = df_cleaned.explode(column_name)
    else:
        # If not multi-valued, just use the cleaned DataFrame
        df_exploded = df_cleaned

    df_exploded[column_name] = df_exploded[column_name].str.strip()

    # Calculate value counts and take the top N
    value_counts = df_exploded[column_name].value_counts().head(top_n)
    return value_counts


def plot_pie_chart(data_series, title):
    """
    Generates a pie chart from a pandas Series (value counts).
    """
    if data_series.empty:
        print(f"Skipping plot: No data to plot for '{title}'.")
        return

    plt.figure(figsize=(10, 8))
    # Use autopct to show percentages, and startangle for orientation
    plt.pie(data_series, labels=data_series.index, autopct='%1.1f%%', startangle=90, pctdistance=0.
    plt.title(title, fontsize=16)
    plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
    plt.tight_layout()
    plt.show()
    print(f"Pie chart for '{title}' plotted.")
```

```
--- Setup: Downloading and Loading the Data ---
Dataset loaded successfully from: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.clou
d/n01PQ9pSmiRX6520flujwQ/survey-data.csv
Initial DataFrame shape: (65437, 114)
Initial DataFrame columns: ['ResponseId', 'MainBranch', 'Age', 'Employment', 'RemoteWork', 'Check',
'CodingActivities', 'EdLevel', 'LearnCode', 'LearnCodeOnline', 'TechDoc', 'YearsCode', 'YearsCodePr
o', 'DevType', 'OrgSize', 'PurchaseInfluence', 'BuyNewTool', 'BuildvsBuy', 'TechEndorse', 'Country',
'Currency', 'CompTotal', 'LanguageHaveWorkedWith', 'LanguageWantToWorkWith', 'LanguageAdmired', 'Dat
abaseHaveWorkedWith', 'DatabaseWantToWorkWith', 'DatabaseAdmired', 'PlatformHaveWorkedWith', 'Platfo
rmWantToWorkWith', 'PlatformAdmired', 'WebframeHaveWorkedWith', 'WebframeWantToWorkWith', 'WebframeA
dmired', 'EmbeddedHaveWorkedWith', 'EmbeddedWantToWorkWith', 'EmbeddedAdmired', 'MiscTechHaveWorkedW
ith', 'MiscTechWantToWorkWith', 'MiscTechAdmired', 'ToolsTechHaveWorkedWith', 'ToolsTechWantToWorkWi
th', 'ToolsTechAdmired', 'NEWCollabToolsHaveWorkedWith', 'NEWCollabToolsWantToWorkWith', 'NEWCollabT
oolsAdmired', 'OpSysPersonal use', 'OpSysProfessional use', 'OfficeStackAsyncHaveWorkedWith', 'Offic
eStackAsyncWantToWorkWith', 'OfficeStackAsyncAdmired', 'OfficeStackSyncHaveWorkedWith', 'OfficeStack
SyncWantToWorkWith', 'OfficeStackSyncAdmired', 'AISearchDevHaveWorkedWith', 'AISearchDevWantToWorkWi
th', 'AISearchDevAdmired', 'NEWSOSites', 'SOVisitFreq', 'SOAccount', 'SOPartFreq', 'SOHow', 'SOCom
m', 'AISelect', 'AISent', 'AIBen', 'AIAcc', 'AIComplex', 'AIToolCurrently Using', 'AIToolInterested
in Using', 'AIToolNot interested in Using', 'AINextMuch more integrated', 'AINextNo change', 'AINext
More integrated', 'AINextLess integrated', 'AINextMuch less integrated', 'AIThreat', 'AIEthics', 'AI
Challenges', 'TBranch', 'ICorPM', 'WorkExp', 'Knowledge_1', 'Knowledge_2', 'Knowledge_3', 'Knowledge
_4', 'Knowledge_5', 'Knowledge_6', 'Knowledge_7', 'Knowledge_8', 'Knowledge_9', 'Frequency_1', 'Freq
uency_2', 'Frequency_3', 'TimeSearching', 'TimeAnswering', 'Frustration', 'ProfessionalTech', 'Profe
ssionalCloud', 'ProfessionalQuestion', 'Industry', 'JobSatPoints_1', 'JobSatPoints_4', 'JobSatPoints
_5', 'JobSatPoints_6', 'JobSatPoints_7', 'JobSatPoints_8', 'JobSatPoints_9', 'JobSatPoints_10', 'Job
SatPoints_11', 'SurveyLength', 'SurveyEase', 'ConvertedCompYearly', 'JobSat']

--- Data Cleaning and Preprocessing ---
```

Download and Load the Data

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

In [4]:
```
# Step 1: Download the dataset
!wget -O survey-data.csv https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9p

# Step 2: Import necessary libraries and load the dataset
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
df = pd.read_csv("survey-data.csv")

# Display the first few rows to understand the structure of the data
df.head()
```

```
--2025-06-18 17:40:42--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/n01PQ9pS
miRX6520flujwQ/survey-data.csv
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-ob
ject-storage.appdomain.cloud)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.clou
d-object-storage.appdomain.cloud)|169.63.118.104|:443... connected.
200 OKequest sent, awaiting response...
Length: 159525875 (152M) [text/csv]
Saving to: ‘survey-data.csv’

survey-data.csv     100%[===================>] 152.13M  52.9MB/s    in 2.9s

2025-06-18 17:40:45 (52.9 MB/s) - ‘survey-data.csv’ saved [159525875/159525875]
```

Out[4]:

	ResponseId	MainBranch	Age	Employment	RemoteWork	Check	CodingActivities	EdLevel	
0	1	I am a developer by profession	Under 18 years old	Employed, full-time	Remote	Apples	Hobby	Primary/elementary school	E
1	2	I am a developer by profession	35-44 years old	Employed, full-time	Remote	Apples	Hobby;Contribute to open-source projects;Other...	Bachelor's degree (B.A., B.S., B.Eng., etc.)	E medi
2	3	I am a developer by profession	45-54 years old	Employed, full-time	Remote	Apples	Hobby;Contribute to open-source projects;Other...	Master's degree (M.A., M.S., M.Eng., MBA, etc.)	E medi
3	4	I am learning to code	18-24 years old	Student, full-time	NaN	Apples	NaN	Some college/university study without earning ...	vi
4	5	I am a developer by profession	18-24 years old	Student, full-time	NaN	Apples	NaN	Secondary school (e.g. American high school, G...	vi

5 rows × 114 columns

Task 1: Visualizing Data Composition with Pie Charts

1.1 Create a Pie Chart of the Top 5 Databases Respondents Want to Work With

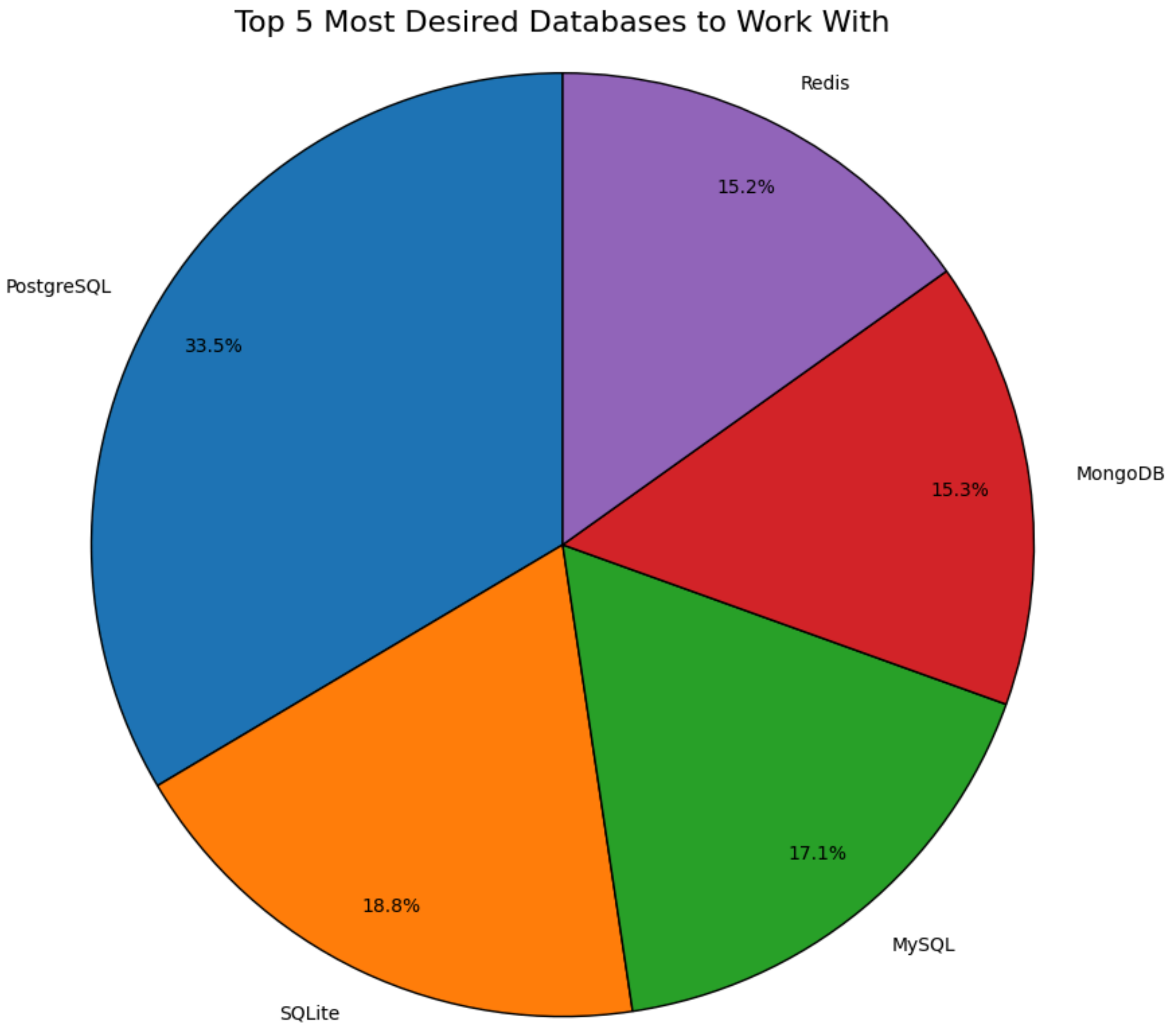
In the survey data, the `DatabaseWantToWorkWith` column lists the databases that respondents wish to work with. Let's visualize the top 5 most-desired databases in a pie chart.

In [5]:

```python
##Write your code here
# --- Task 1: Visualizing Data Composition with Pie Charts ---
print("\n--- Task 1: Visualizing Data Composition with Pie Charts ---")

# 1.1 Create a Pie Chart of the Top 5 Databases Respondents Want to Work With
databases_counts = clean_and_explode_column(df, 'DatabaseWantToWorkWith', top_n=5)
plot_pie_chart(databases_counts, 'Top 5 Most Desired Databases to Work With')
```

```
--- Task 1: Visualizing Data Composition with Pie Charts ---
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```

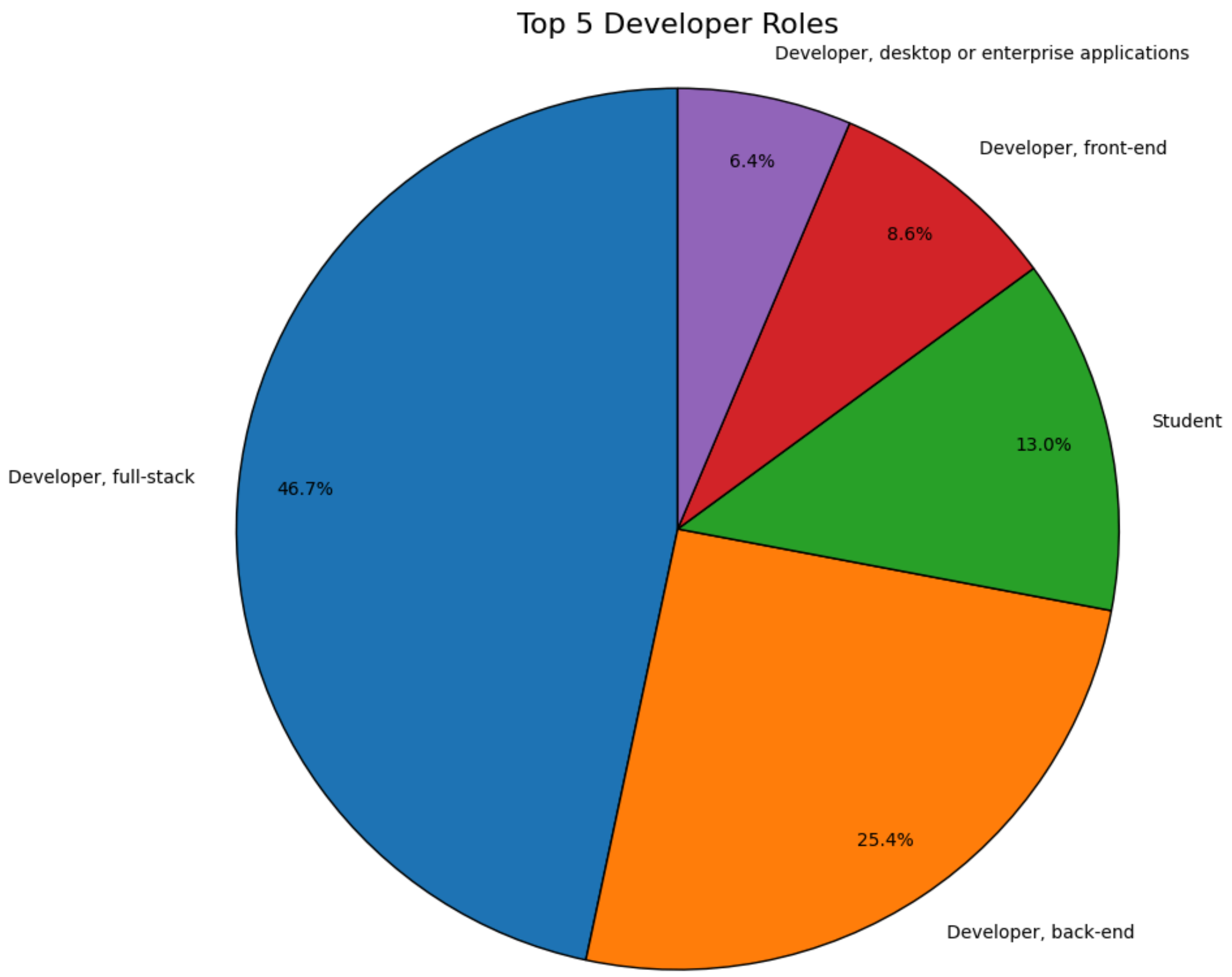


```
Pie chart for 'Top 5 Most Desired Databases to Work With' plotted.
```

The `DevType` column lists the developer types for respondents. We'll examine the distribution by showing the top 5 developer roles in a pie chart.

```
In [6]: ##Write your code here
        # 1.2 The DevType column lists the developer types for respondents. Top 5 developer roles.
        devtype_counts = clean_and_explode_column(df, 'DevType', top_n=5)
        plot_pie_chart(devtype_counts, 'Top 5 Developer Roles')
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



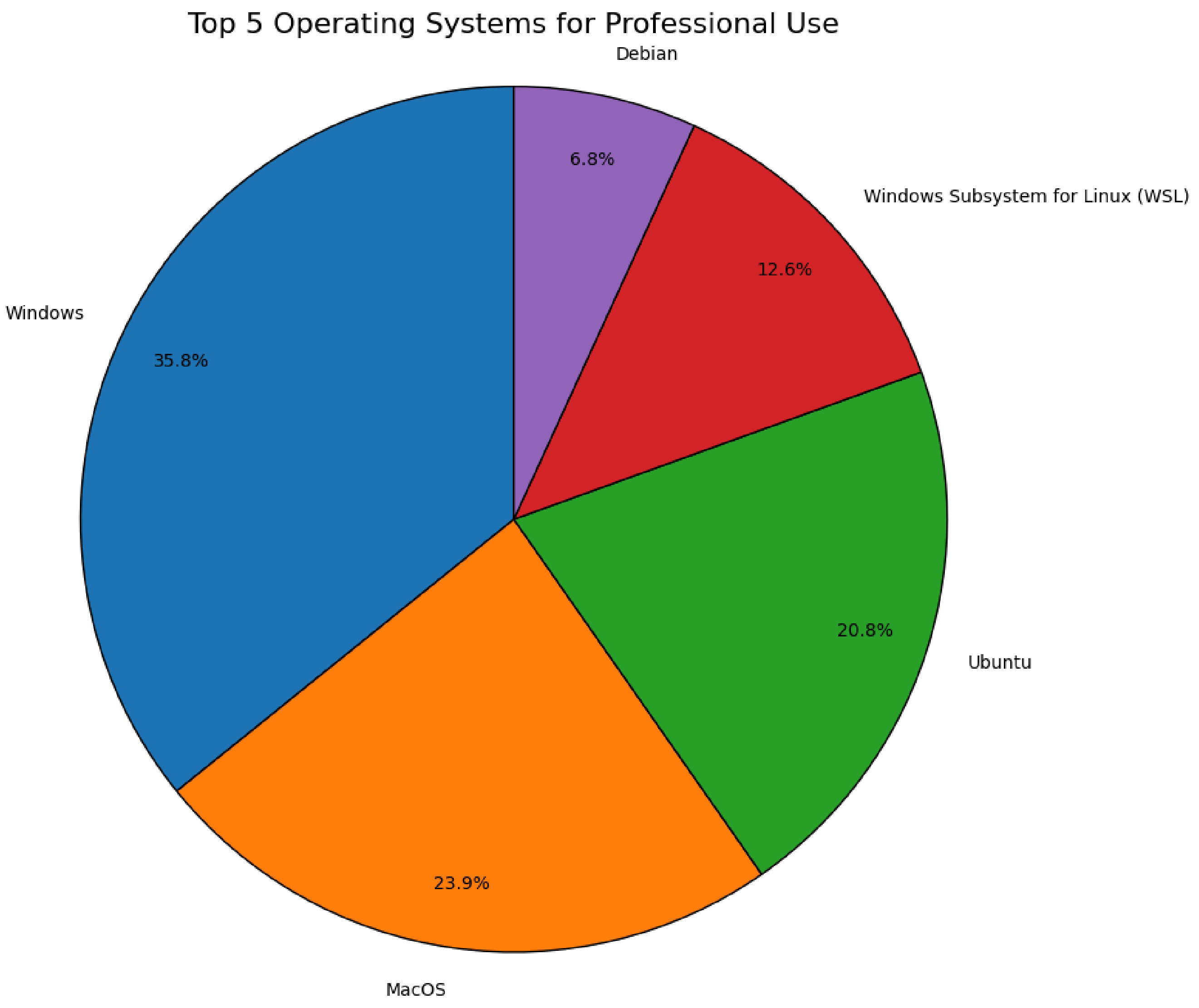
```
Pie chart for 'Top 5 Developer Roles' plotted.
```

1.3 Create a pie chart for the operating systems used by respondents for professional use

The `OpSysProfessional` use column shows the operating systems developers use professionally. Let's visualize the distribution of the top operating systems in a pie chart.

In [7]:
```
##Write your code here
# 1.3 Create a pie chart for the operating systems used by respondents for professional use
opsys_professional_counts = clean_and_explode_column(df, 'OpSysProfessional use', top_n=5)
plot_pie_chart(opsys_professional_counts, 'Top 5 Operating Systems for Professional Use')
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



```
Pie chart for 'Top 5 Operating Systems for Professional Use' plotted.
```

Task 2: Additional Visualizations and Comparisons

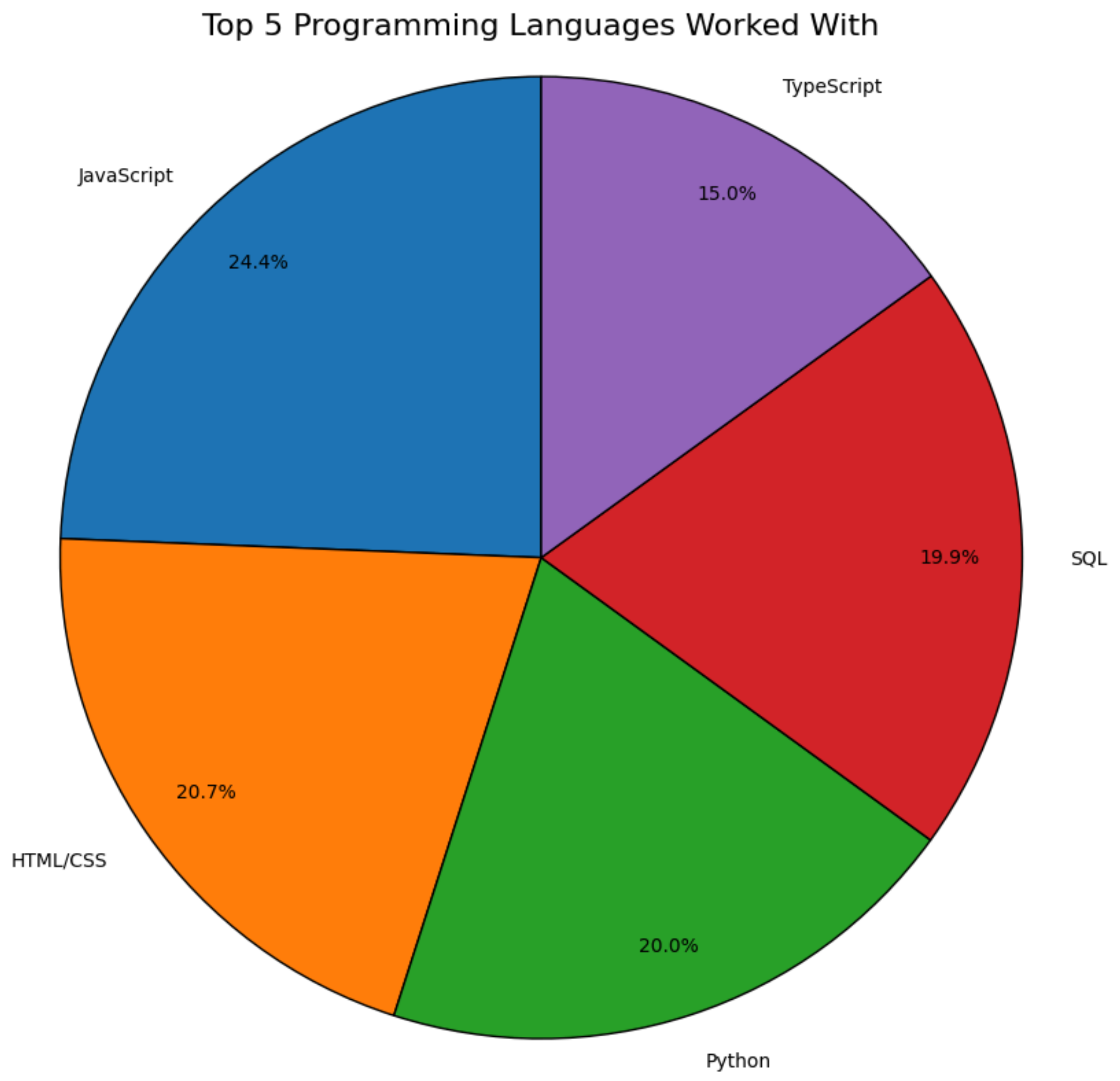
2.1 Pie Chart for Top 5 Programming Languages Respondents Have Worked With

The `LanguageHaveWorkedWith` column contains the programming languages that respondents have experience with. We'll plot a pie chart to display the composition of the top 5 languages.

In [8]:
```
##Write your code here
# --- Task 2: Additional Visualizations and Comparisons ---
print("\n--- Task 2: Additional Visualizations and Comparisons ---")

# 2.1 Pie Chart for Top 5 Programming Languages Respondents Have Worked With
languages_worked_counts = clean_and_explode_column(df, 'LanguageHaveWorkedWith', top_n=5)
plot_pie_chart(languages_worked_counts, 'Top 5 Programming Languages Worked With')
```

```
--- Task 2: Additional Visualizations and Comparisons ---
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



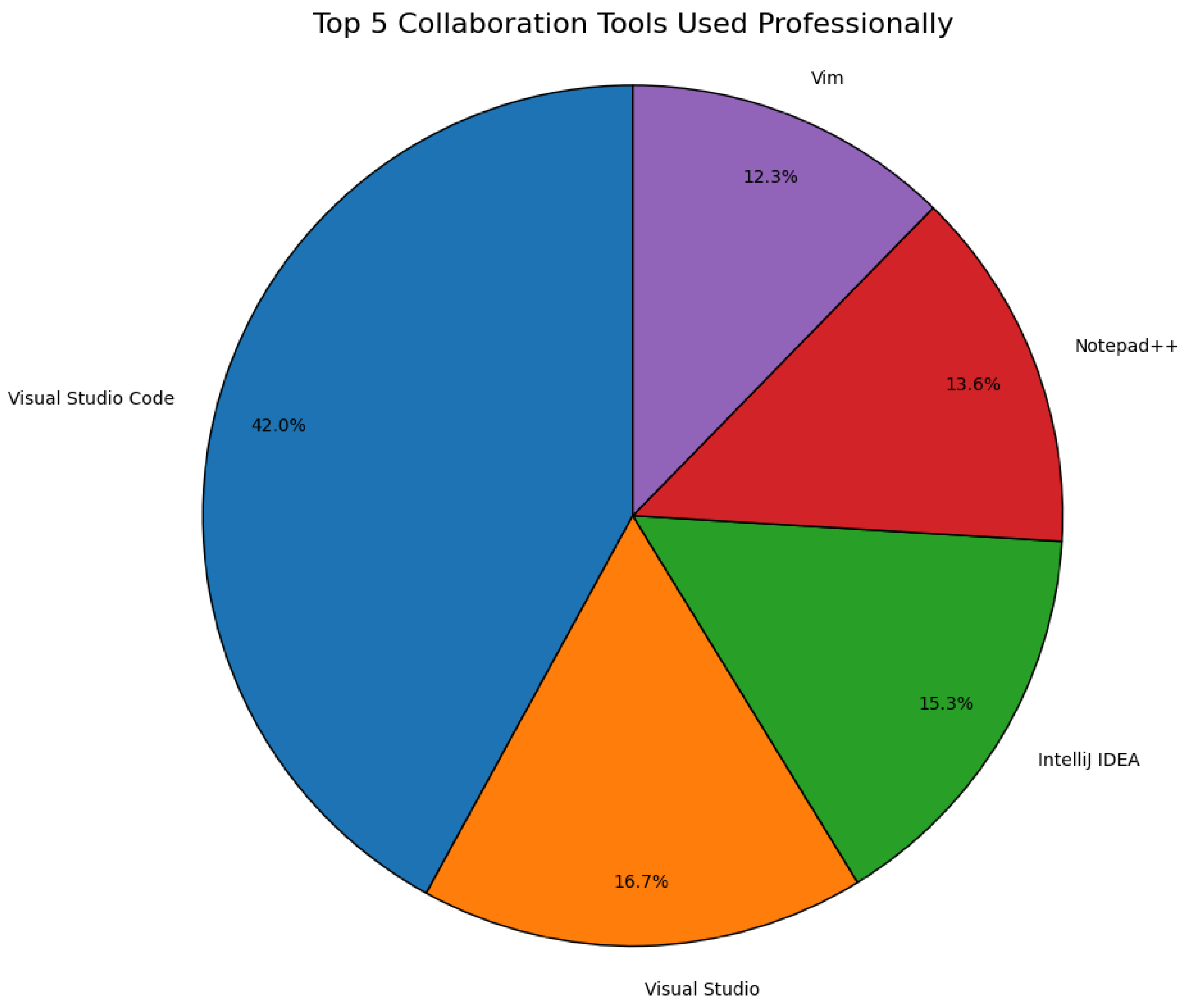
```
Pie chart for 'Top 5 Programming Languages Worked With' plotted.
```

2.2 Pie Chart for Top Collaboration Tools used in Professional Use

Using the `NEWCollabToolsHaveWorkedWith` column, we'll identify and visualize the top collaboration tools respondents use in their professional work.

In [9]:
```
##Write your code here
# 2.2 Pie Chart for Top Collaboration Tools used in Professional Use
collab_tools_counts = clean_and_explode_column(df, 'NEWCollabToolsHaveWorkedWith', top_n=5)
plot_pie_chart(collab_tools_counts, 'Top 5 Collaboration Tools Used Professionally')
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



```
Pie chart for 'Top 5 Collaboration Tools Used Professionally' plotted.
```

Task 3: Analyzing and Interpreting Composition

In this task, you will create additional pie charts to analyze specific aspects of the survey data. Use `pandas` and `matplotlib` to complete each task and interpret the findings.

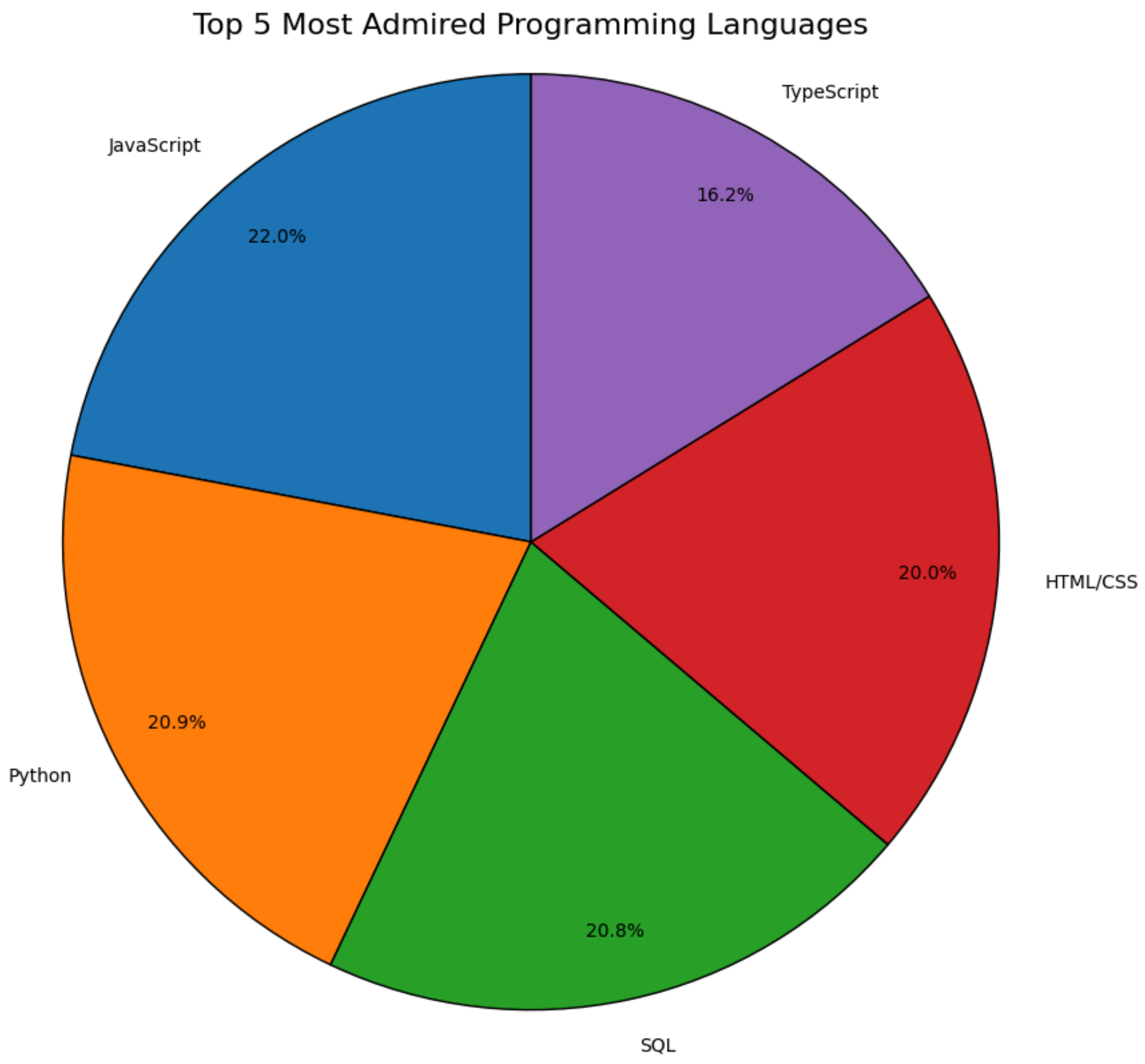
3.1 Pie Chart of `Respondents` Most Admired Programming Languages

The `LanguageAdmired` column lists the programming languages respondents admire most. Create a pie chart to visualize the top 5 admired languages.

In [10]:
```python
##Write your code here
# --- Task 3: Analyzing and Interpreting Composition ---
print("\n--- Task 3: Analyzing and Interpreting Composition ---")

# 3.1 Pie Chart of Respondents Most Admired Programming Languages
languages_admired_counts = clean_and_explode_column(df, 'LanguageAdmired', top_n=5)
plot_pie_chart(languages_admired_counts, 'Top 5 Most Admired Programming Languages')
```

```
--- Task 3: Analyzing and Interpreting Composition ---
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



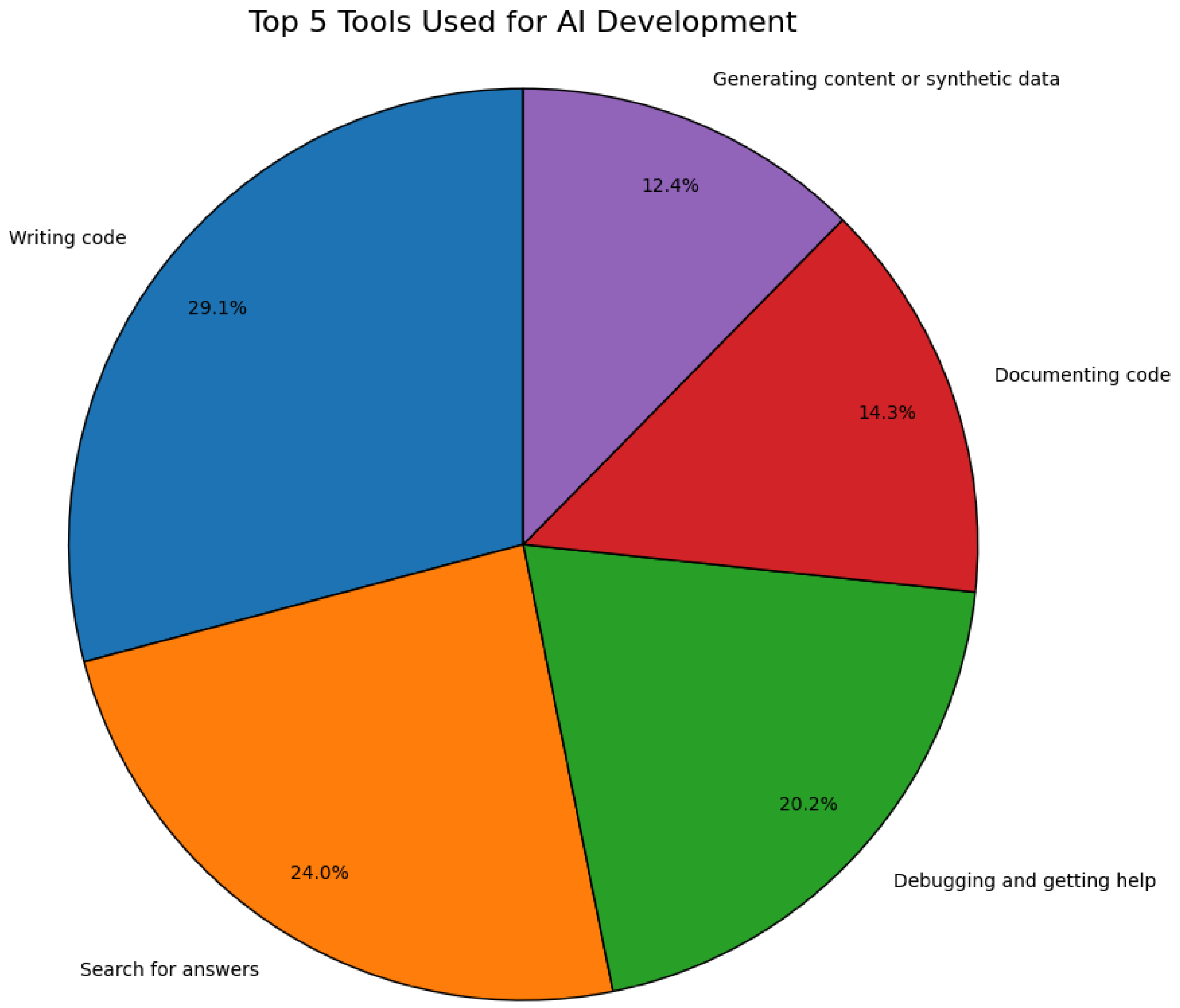
```
Pie chart for 'Top 5 Most Admired Programming Languages' plotted.
```

3.2 Pie Chart of Tools Used for AI Development

Using the `AIToolCurrently` Using column, create a pie chart to visualize the top 5 tools developers are currently using for AI development.

```
In [11]: ##Write your code here
         # 3.2 Pie Chart of Tools Used for AI Development
         ai_tools_counts = clean_and_explode_column(df, 'AIToolCurrently Using', top_n=5)
         plot_pie_chart(ai_tools_counts, 'Top 5 Tools Used for AI Development')
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



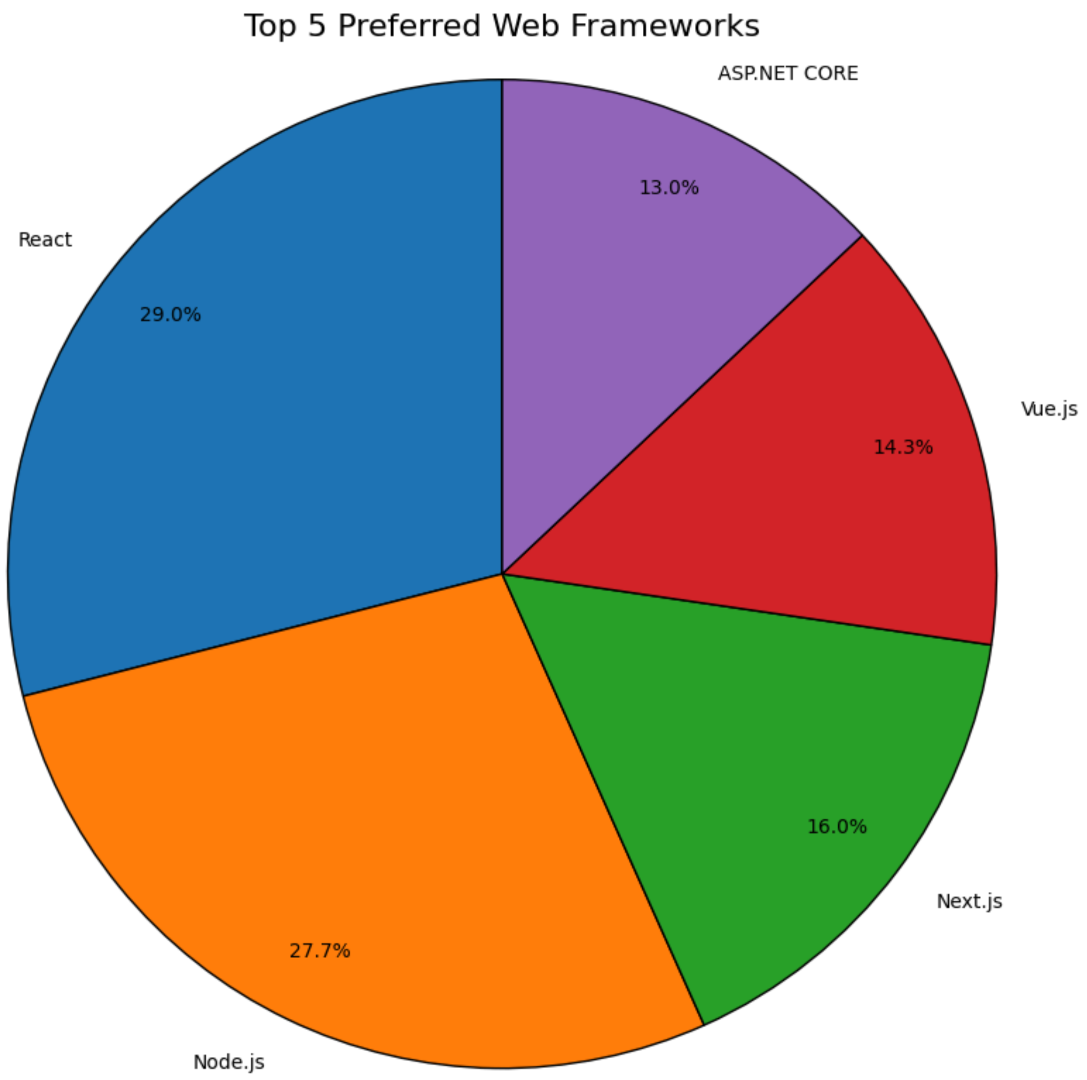
```
Pie chart for 'Top 5 Tools Used for AI Development' plotted.
```

3.3 Pie Chart for Preferred Web Frameworks

The `WebframeWantToWorkWith` column includes web frameworks that respondents are interested in working with. Visualize the top 5 frameworks in a pie chart.

In [12]:
```
##Write your code here
# 3.3 Pie Chart for Preferred Web Frameworks
web_frameworks_counts = clean_and_explode_column(df, 'WebframeWantToWorkWith', top_n=5)
plot_pie_chart(web_frameworks_counts, 'Top 5 Preferred Web Frameworks')
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



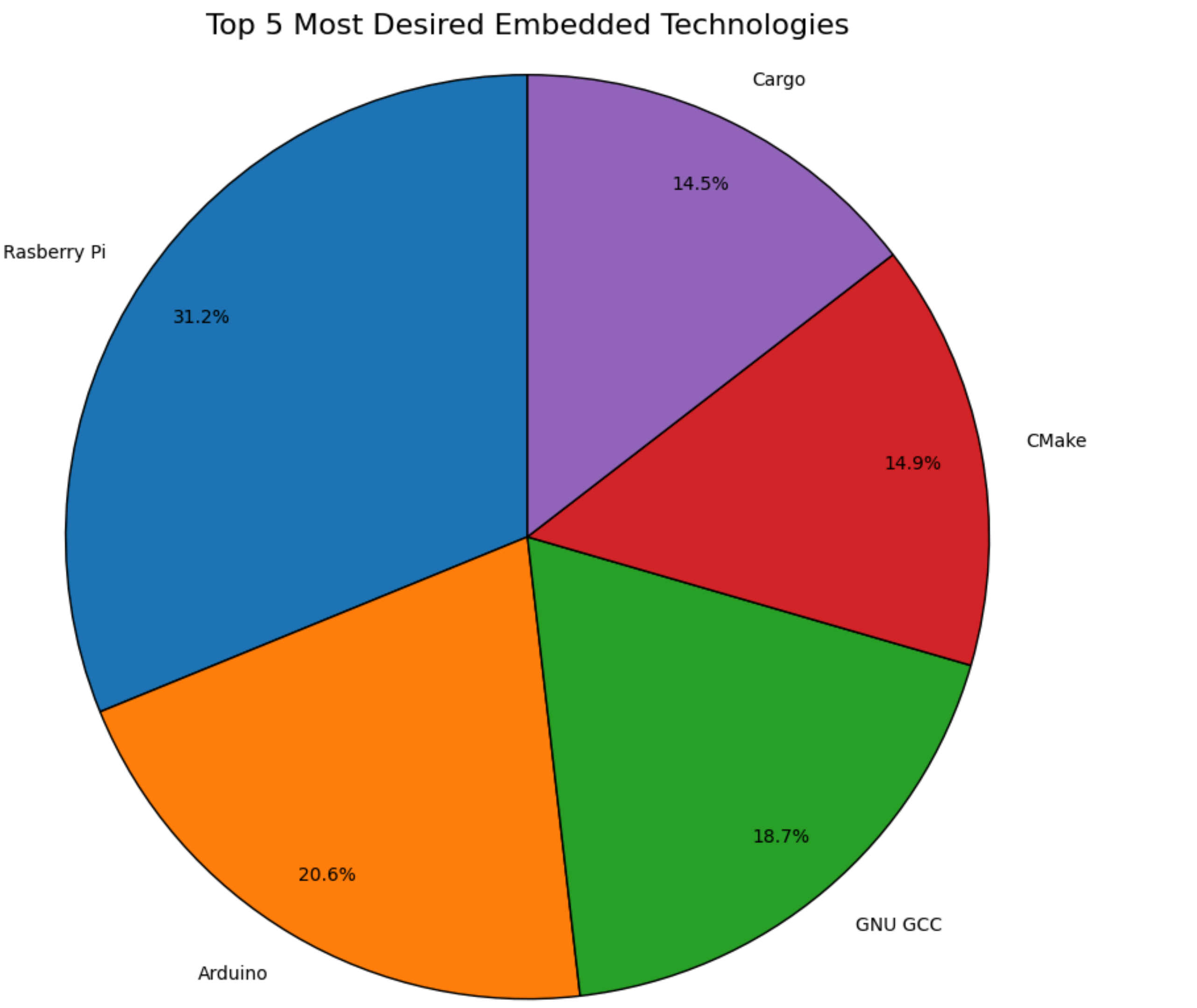
```
Pie chart for 'Top 5 Preferred Web Frameworks' plotted.
```

3.4 Pie Chart for Most Desired Embedded Technologies

Using the `EmbeddedWantToWorkWith` column, create a pie chart to show the top 5 most desired embedded technologies that respondents wish to work with.

```
In [13]: ##Write your code here
# 3.4 Pie Chart for Most Desired Embedded Technologies
embedded_tech_counts = clean_and_explode_column(df, 'EmbeddedWantToWorkWith', top_n=5)
plot_pie_chart(embedded_tech_counts, 'Top 5 Most Desired Embedded Technologies')


print("\n--- Lab Completion Summary ---")
print("All pie chart tasks have been attempted. Please review the plots and console output for any
```

```
/tmp/ipykernel_489/2755271071.py:88: FutureWarning: A value is trying to be set on a copy of a DataF
rame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate
object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, in
place=True)' or df[col] = df[col].method(value) instead, to perform the operation inplace on the ori
ginal object.


  df_cleaned[column_name].replace(['nan', 'NaN', 'N/A', 'None', '', ' '], np.nan, inplace=True)
```



```
Pie chart for 'Top 5 Most Desired Embedded Technologies' plotted.

--- Lab Completion Summary ---
All pie chart tasks have been attempted. Please review the plots and console output for any warnings
or skipped tasks.
```

Summary

After completing this lab, you will be able to:

- Create pie charts to visualize developer preferences across databases, programming languages, AI tools, and cloud platforms.
- Identify trends in technology usage, role distribution, and tool adoption through pie charts.
- Analyze and compare data composition across various categories to gain insights into developer preferences.

Authors:

Ayushi Jain

Other Contributors:

- Rav Ahuja
- Lakshmi Holla
- Malika