Data Visualization Lab

Estimated time needed: **45 to 60** minutes

In this assignment you will be focusing on the visualization of data.

The data set will be presented to you in the form of a RDBMS.

You will have to use SQL queries to extract the data.

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.

Demo: How to work with database

Download database file.

In [1]:
```
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m4_survey_data.s
```

```
--2024-02-04 18:42:26--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeDa
ta/m4_survey_data.sqlite
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.clo
ud)... 169.63.118.104, 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomai
n.cloud)|169.63.118.104|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 36679680 (35M) [application/octet-stream]
Saving to: ‘m4_survey_data.sqlite.2’

m4_survey_data.sqli 100%[===================>]  34.98M  32.4MB/s    in 1.1s

2024-02-04 18:42:28 (32.4 MB/s) - ‘m4_survey_data.sqlite.2’ saved [36679680/36679680]
```

Connect to the database.

In [2]:
```python
import sqlite3
conn = sqlite3.connect("m4_survey_data.sqlite") # open a database connection
```

Import pandas module.

In [3]:
```python
import pandas as pd
```

Demo: How to run an sql query

In [4]:
```python
# print how many rows are there in the table named 'master'
QUERY = """
SELECT COUNT(*)
FROM master
"""

# the read_sql_query runs the sql query and returns the data as a dataframe
df = pd.read_sql_query(QUERY,conn)
df.head()
```

Out[4]:

	COUNT(*)
0	11398

Demo: How to list all tables

In [5]:
```
# print all the tables names in the database
QUERY = """
SELECT name as Table_Name FROM
sqlite_master WHERE
type = 'table'
"""
# the read_sql_query runs the sql query and returns the data as a dataframe
pd.read_sql_query(QUERY,conn)
```

Out[5]:

	Table_Name
0	EduOther
1	DevType
2	LastInt
3	JobFactors
4	WorkPlan
5	WorkChallenge
6	LanguageWorkedWith
7	LanguageDesireNextYear
8	DatabaseWorkedWith
9	DatabaseDesireNextYear
10	PlatformWorkedWith
11	PlatformDesireNextYear
12	WebFrameWorkedWith
13	WebFrameDesireNextYear
14	MiscTechWorkedWith
15	MiscTechDesireNextYear
16	DevEnviron
17	Containers
18	SOVisitTo
19	SONewContent
20	Gender
21	Sexuality
22	Ethnicity
23	master

Demo: How to run a group by query

In [6]:
```
QUERY = """
SELECT Age,COUNT(*) as count
FROM master
group by age
order by age
"""
pd.read_sql_query(QUERY,conn)
```

Out[6]:

	Age	count
0	NaN	287
1	16.0	3
2	17.0	6
3	18.0	29
4	19.0	78
5	20.0	109
6	21.0	203
7	22.0	406
8	23.0	581
9	24.0	679
10	25.0	738
11	26.0	720
12	27.0	724
13	28.0	787
14	29.0	697
15	30.0	651
16	31.0	531
17	32.0	489
18	33.0	483
19	34.0	395
20	35.0	393
21	36.0	308
22	37.0	280
23	38.0	279
24	39.0	232
25	40.0	187
26	41.0	136
27	42.0	162
28	43.0	100
29	44.0	95
30	45.0	85
31	46.0	66
32	47.0	68
33	48.0	64
34	49.0	66
35	50.0	57
36	51.0	29
37	52.0	41
38	53.0	32
39	54.0	26
40	55.0	13
41	56.0	16
42	57.0	11
43	58.0	12
44	59.0	11
45	60.0	2
46	61.0	10
47	62.0	5
48	63.0	7

	Age	count
49	65.0	2
50	66.0	1
51	67.0	1
52	69.0	1
53	71.0	2
54	72.0	1
55	99.0	1

Demo: How to describe a table

In [7]:
```
table_name = 'master'  # the table you wish to describe

QUERY = """
SELECT sql FROM sqlite_master
WHERE name= '{}'
""".format(table_name)

df = pd.read_sql_query(QUERY,conn)
print(df.iat[0,0])
```

```
CREATE TABLE "master" (
"index" INTEGER,
  "Respondent" INTEGER,
  "MainBranch" TEXT,
  "Hobbyist" TEXT,
  "OpenSourcer" TEXT,
  "OpenSource" TEXT,
  "Employment" TEXT,
  "Country" TEXT,
  "Student" TEXT,
  "EdLevel" TEXT,
  "UndergradMajor" TEXT,
  "OrgSize" TEXT,
  "YearsCode" TEXT,
  "Age1stCode" TEXT,
  "YearsCodePro" TEXT,
  "CareerSat" TEXT,
  "JobSat" TEXT,
  "MgrIdiot" TEXT,
  "MgrMoney" TEXT,
  "MgrWant" TEXT,
  "JobSeek" TEXT,
  "LastHireDate" TEXT,
  "FizzBuzz" TEXT,
  "ResumeUpdate" TEXT,
  "CurrencySymbol" TEXT,
  "CurrencyDesc" TEXT,
  "CompTotal" REAL,
  "CompFreq" TEXT,
  "ConvertedComp" REAL,
  "WorkWeekHrs" REAL,
  "WorkRemote" TEXT,
  "WorkLoc" TEXT,
  "ImpSyn" TEXT,
  "CodeRev" TEXT,
  "CodeRevHrs" REAL,
  "UnitTests" TEXT,
  "PurchaseHow" TEXT,
  "PurchaseWhat" TEXT,
  "OpSys" TEXT,
  "BlockchainOrg" TEXT,
  "BlockchainIs" TEXT,
  "BetterLife" TEXT,
  "ITperson" TEXT,
  "OffOn" TEXT,
  "SocialMedia" TEXT,
  "Extraversion" TEXT,
  "ScreenName" TEXT,
  "SOVisit1st" TEXT,
  "SOVisitFreq" TEXT,
  "SOFindAnswer" TEXT,
  "SOTimeSaved" TEXT,
  "SOHowMuchTime" TEXT,
  "SOAccount" TEXT,
  "SOPartFreq" TEXT,
  "SOJobs" TEXT,
  "EntTeams" TEXT,
  "SOComm" TEXT,
  "WelcomeChange" TEXT,
  "Age" REAL,
  "Trans" TEXT,
  "Dependents" TEXT,
  "SurveyLength" TEXT,
  "SurveyEase" TEXT
)
```

Hands-on Lab

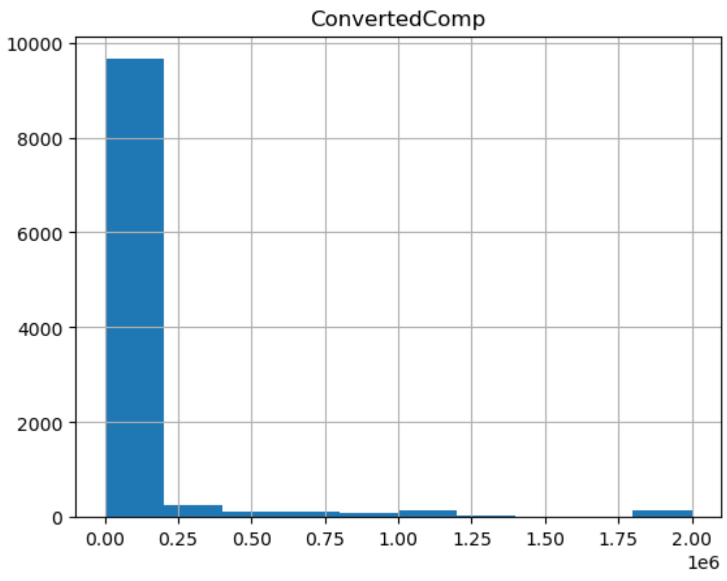
Visualizing distribution of data

Histograms

Plot a histogram of `ConvertedComp.`

In [8]:
```
QUERY = """
SELECT ConvertedComp
FROM master
"""
df = pd.read_sql_query(QUERY,conn)
df.hist(bins=10)
```

```
Out[8]: array([[<AxesSubplot:title={'center':'ConvertedComp'}>]], dtype=object)
```

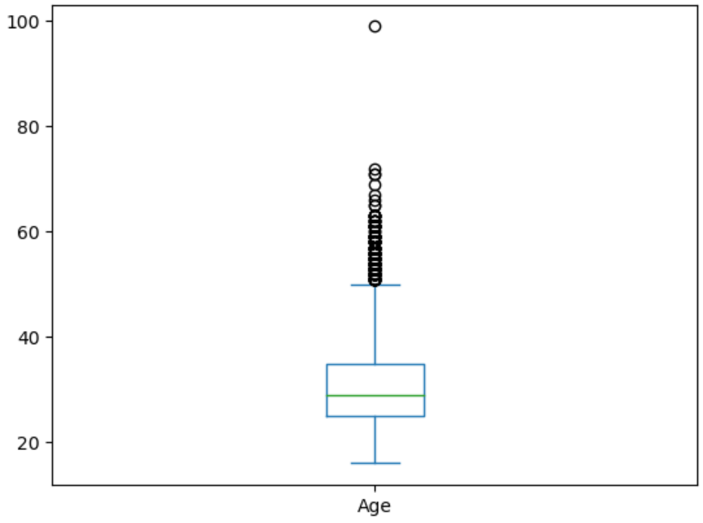


Box Plots

Plot a box plot of `Age.`

```
In [9]: QUERY = """
SELECT Age
FROM master
"""
df = pd.read_sql_query(QUERY,conn)
df.plot(kind='box')
```

```
Out[9]: <AxesSubplot:>
```



Visualizing relationships in data

Scatter Plots

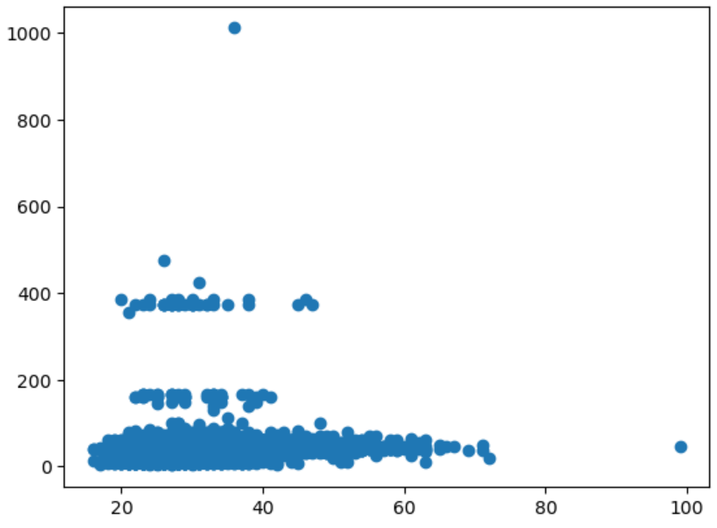
Create a scatter plot of `Age` and `WorkWeekHrs.`

```
In [10]: from matplotlib import pyplot as plt
import seaborn as sns
```

```
In [11]: QUERY = """
SELECT Age, WorkWeekHrs
FROM master
"""
df = pd.read_sql_query(QUERY,conn)
```

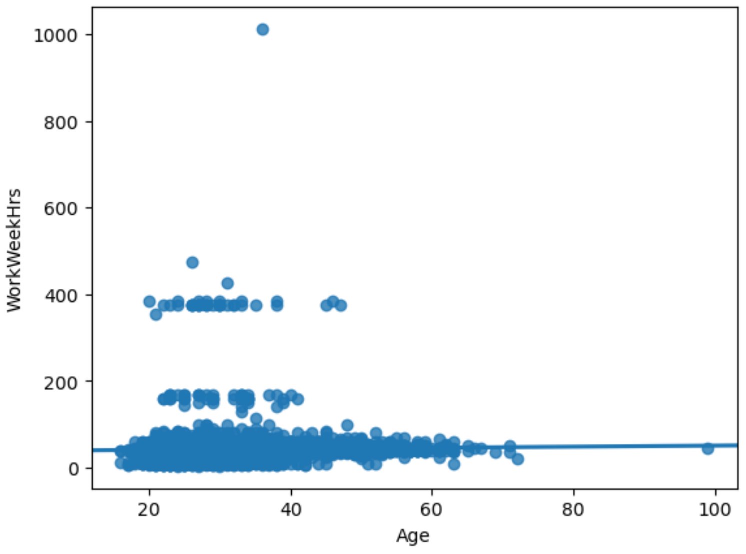
```
x=df['Age']
y=df['WorkWeekHrs']
plt.scatter(x,y)
```

Out[11]:
```
<matplotlib.collections.PathCollection at 0x7f7f6d3fa3d0>
```



In [12]:
```
#catter plot of Age and WorkWeekHrs with a regression line
sns.regplot(x=df['Age'], y=df['WorkWeekHrs'],data= df)
```

Out[12]:
```
<AxesSubplot:xlabel='Age', ylabel='WorkWeekHrs'>
```



Bubble Plots

Create a bubble plot of `WorkWeekHrs` and `CodeRevHrs`, use `Age` column as bubble size.

In [13]:
```
import plotly.express as px
```

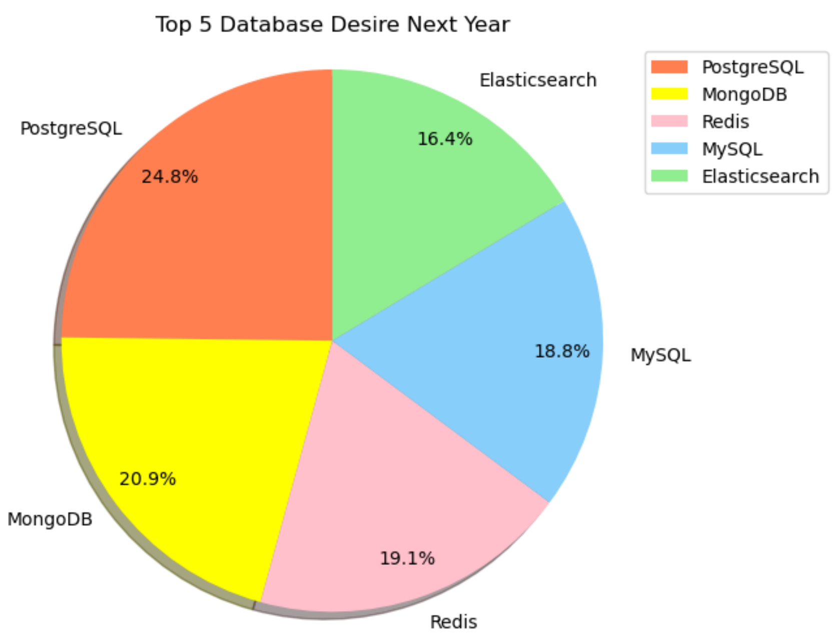
In [14]:
```
QUERY = """
SELECT WorkWeekHrs, CodeRevHrs, Age
FROM master

"""
df = pd.read_sql_query(QUERY,conn)

#drop the rows where Age value is missing.
df.dropna(subset=['Age'], inplace=True)

px.scatter(df, x='WorkWeekHrs', y='CodeRevHrs', size='Age')
```

Visualizing composition of data

Pie Charts

Create a pie chart of the top 5 databases that respondents wish to learn next year. Label the pie chart with database names. Display percentages of each database on the pie chart.

In [38]:
```
QUERY = """
SELECT DatabaseDesireNextYear, Count(*) as Count
FROM DatabaseDesireNextYear
GROUP BY DatabaseDesireNextYear
ORDER BY Count DESC
LIMIT 5
"""

df = pd.read_sql_query(QUERY, conn)

df.set_index('DatabaseDesireNextYear', inplace=True)

color_list = ['coral', 'yellow', 'pink', 'lightskyblue', 'lightgreen']
labels = df.index  # use the index as labels

# create the pie chart
plt.figure(figsize=(10, 6))
plt.pie(df['Count'], labels=labels, colors=color_list, autopct='%1.1f%%', startangle=90, shadow=True, pctdistance=0.85)
plt.title('Top 5 Database Desire Next Year')
plt.axis('equal')  # pie is drawn as a circle
plt.legend(labels, loc='upper right')

plt.show()
```



In [52]:
```
#Finf 10 most popular languages respondents wish to learn next year. What is the rank of Python?

QUERY = """
SELECT LanguageDesireNextYear, Count(*) as Count
FROM LanguageDesireNextYear
GROUP BY LanguageDesireNextYear
ORDER BY Count DESC
LIMIT 10
"""

df = pd.read_sql_query(QUERY, conn)
df.head()

#df.loc[df['LanguageDesireNextYear'] == 'Python']
```

Out[52]:

	LanguageDesireNextYear	Count
0	JavaScript	6630
1	HTML/CSS	5328
2	Python	5239
3	SQL	5012
4	TypeScript	4088

In [59]:
```
#How many respondents indicated that they currently work with 'SQL'?

QUERY = """
SELECT LanguageWorkedWith, Count(*) as Count
FROM LanguageWorkedWith
GROUP BY LanguageWorkedWith
ORDER BY Count DESC
LIMIT 10
"""

df = pd.read_sql_query(QUERY, conn)
#df.head()

df.loc[df['LanguageWorkedWith'] == 'SQL']
```

Out[59]:

	LanguageWorkedWith	Count
2	SQL	7106

In [103…
```
#How many respondents indicated that they work on 'MySQL' only?

QUERY="""
SELECT DatabaseWorkedWith, Count(Respondent) as Count
FROM DatabaseWorkedWith
group by Respondent
having count(DatabaseWorkedWith)=1 and DatabaseWorkedWith='MySQL'"""

df = pd.read_sql_query(QUERY, conn)
```

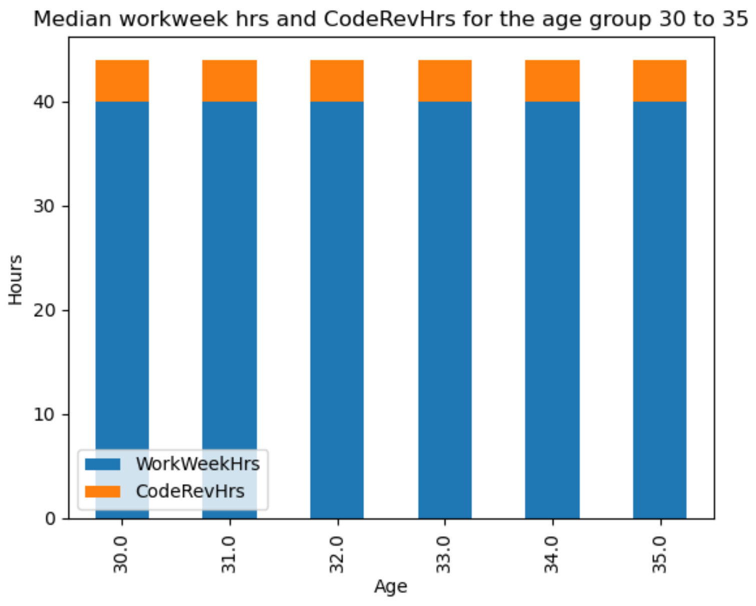
```
#df.head()
print(df['Count'].sum())
```

```
474
```

Stacked Charts

Create a stacked chart of median `WorkWeekHrs` and `CodeRevHrs` for the age group 30 to 35.

In [139…
```
QUERY = """
SELECT Age, WorkWeekHrs, CodeRevHrs
FROM master
WHERE Age BETWEEN 30 AND 35
ORDER BY Age asc
"""

df = pd.read_sql_query(QUERY, conn)

df1=df.groupby('Age').median()

df1.plot(kind='bar',stacked=True)
plt.title('Median workweek hrs and CodeRevHrs for the age group 30 to 35')
plt.xlabel('Age')
plt.ylabel('Hours')
plt.show()
```

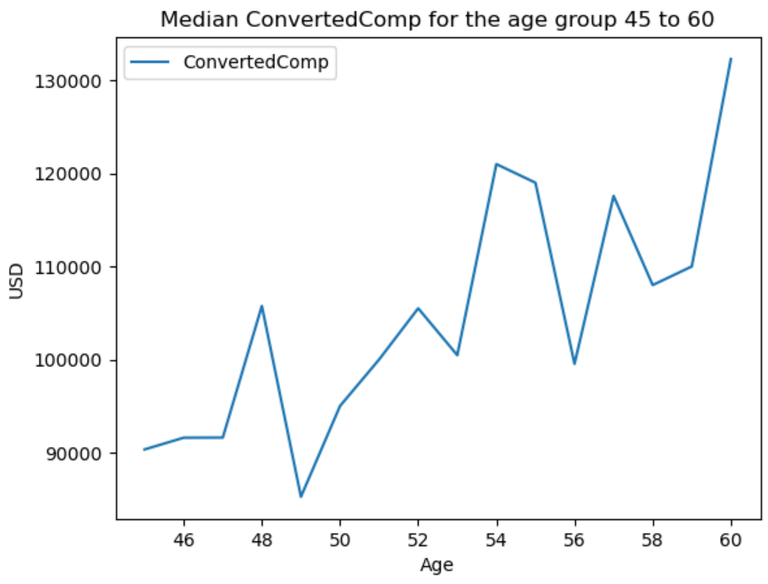


Visualizing comparison of data

Line Chart

Plot the median `ConvertedComp` for all ages from 45 to 60.

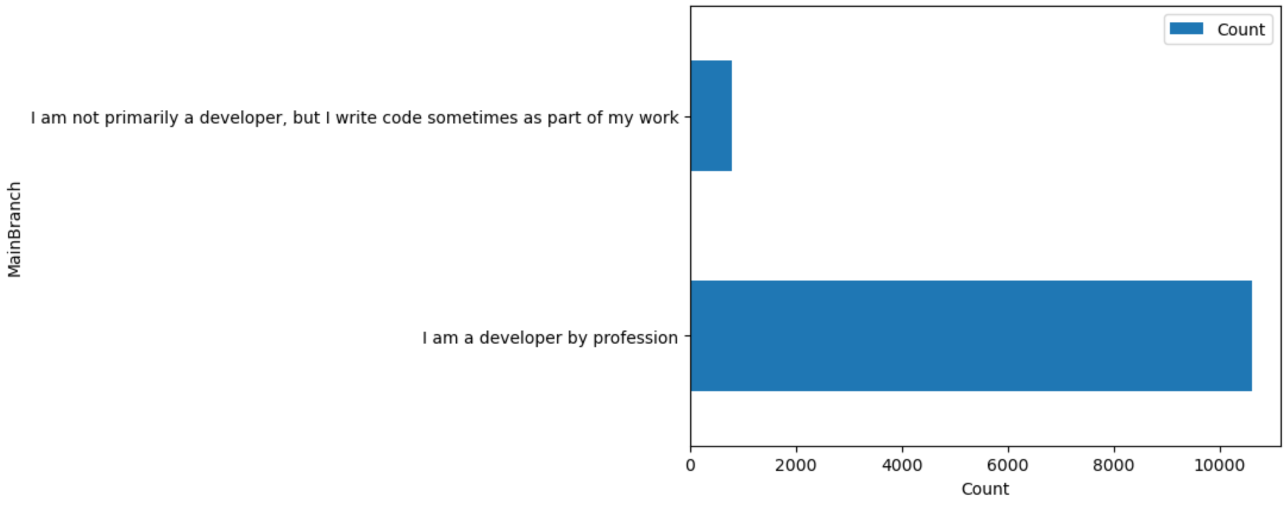
In [145…
```
QUERY = """
SELECT Age, ConvertedComp
FROM master
WHERE Age BETWEEN 45 AND 60
ORDER BY Age asc
"""

df = pd.read_sql_query(QUERY, conn)

df1=df.groupby('Age').median()
df1.head()

df1.plot(kind='line',stacked=True)
plt.title('Median ConvertedComp for the age group 45 to 60')
plt.xlabel('Age')
plt.ylabel('USD')
plt.show()
```



Bar Chart

Create a horizontal bar chart using column `MainBranch.`

```
In [156…
QUERY = """
SELECT MainBranch, Count(*) as Count
FROM master
GROUP BY MainBranch

"""

df = pd.read_sql_query(QUERY, conn)
df.set_index('MainBranch', inplace=True)
df.head()

df.plot(kind='barh')
plt.xlabel('Count')
plt.show()
```



```
In [165…
#Find the 5 top DevType (by occurence)

QUERY = """
SELECT DevType, Count(*) as Count
FROM DevType
GROUP BY DevType
ORDER BY Count DESC
LIMIT 5
"""

df = pd.read_sql_query(QUERY, conn)

df.head()
```

Out[165]:

	DevType	Count
0	Developer, full-stack	6928
1	Developer, back-end	6290
2	Developer, front-end	3920
3	Developer, desktop or enterprise applications	2575
4	Developer, mobile	1959

Close the database connection.

In [ ]:
```
conn.close()
```

Authors

Ramesh Sannareddy

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2020-10-17	0.1	Ramesh Sannareddy	Created initial version of the lab