(https://skills.network/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDA0321ENSkillsNetwork21426264-2022-01-01)

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 [22]:
```
!wget https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeData/m4_survey_data.sqli
```

```
--2023-02-02 13:04:24--  https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeDat
a/m4_survey_data.sqlite (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DA0321EN-SkillsNetwork/LargeDat
a/m4_survey_data.sqlite)
Resolving cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.clou
d)... 169.63.118.104
Connecting to cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud (cf-courses-data.s3.us.cloud-object-storage.appdomain.c
loud)|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.1’

m4_survey_data.sqli 100%[===================>]  34.98M  36.3MB/s    in 1.0s

2023-02-02 13:04:26 (36.3 MB/s) - ‘m4_survey_data.sqlite.1’ saved [36679680/36679680]
```

Connect to the database.

In [23]:
```
import sqlite3
conn = sqlite3.connect("m4_survey_data.sqlite") # open a database connection
```

Import pandas module.

In [24]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Demo: How to run an sql query

In [25]:

```
# 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[25]:

	COUNT(*)
0	11398

Demo: How to list all tables

In [26]:

```
# 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[26]:

	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

Demo: How to run a group by query

In [27]:

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

Out[27]:

	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
49	65.0	2
50	66.0	1
51	67.0	1
52	69.0	1
53	71.0	2
54	72.0	1

	Age	count
55	99.0	1

In [31]:

```
QUERY = """
SELECT DatabaseDesireNextYear, COUNT(DatabaseDesireNextYear) as count
FROM DatabaseDesireNextYear
group by DatabaseDesireNextYear
order by count desc

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

Out[31]:

	DatabaseDesireNextYear	count
0	PostgreSQL	4328
1	MongoDB	3649
2	Redis	3331
3	MySQL	3281
4	Elasticsearch	2856
5	Microsoft SQL Server	2706
6	SQLite	2437
7	Firebase	1650
8	MariaDB	1385
9	DynamoDB	1044
10	Cassandra	1003
11	Oracle	870
12	Other(s):	645
13	Couchbase	390

In [34]:

```
QUERY = """
SELECT Respondent, DatabaseWorkedWith, COUNT(DatabaseWorkedWith) as count
FROM DatabaseWorkedWith
group by Respondent
"""
# the read_sql_query runs the sql query and returns the data as a dataframe
df = pd.read_sql_query(QUERY,conn)
df[(df['DatabaseWorkedWith'] == 'SQL') & (df['count'] ==1 )]
```

Out[34]:

Respondent	DatabaseWorkedWith	count

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

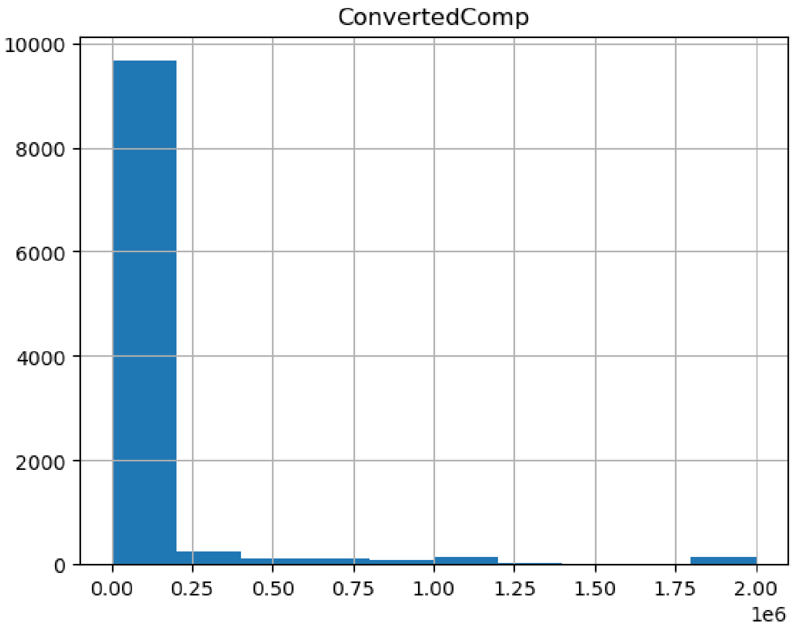
Plot a histogram of `ConvertedComp`.

In [8]:

```
# your code goes here
QUERY = """
SELECT * FROM master
"""
df = pd.read_sql_query(QUERY,conn)
df.hist(column='ConvertedComp')
```

Out[8]:

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



Box Plots

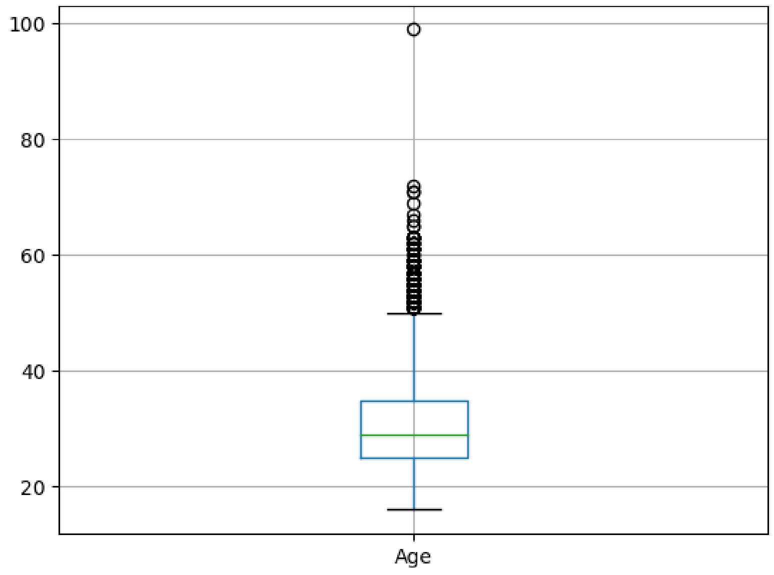
Plot a box plot of `Age`.

In [9]:

```python
# your code goes here
QUERY = """
SELECT * FROM master
"""
df = pd.read_sql_query(QUERY,conn)
df.boxplot(column='Age')
```

Out[9]:

```
<AxesSubplot:>
```



Visualizing relationships in data

Scatter Plots

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

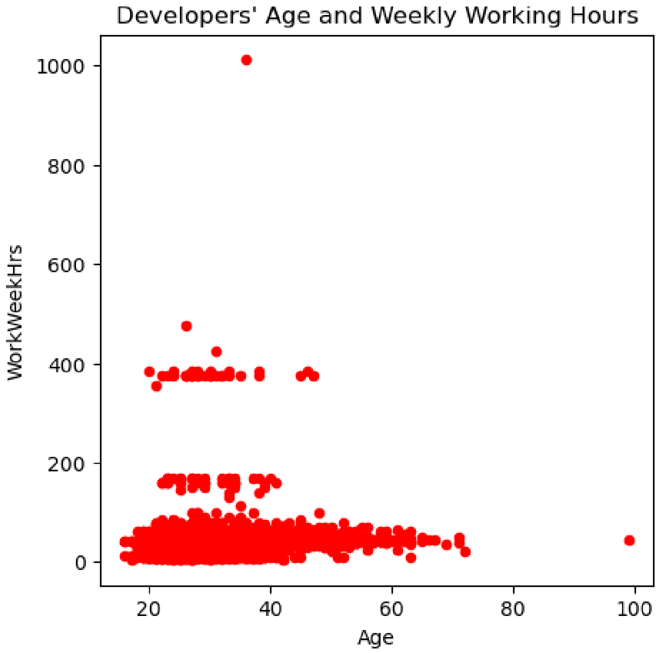
In [13]:

```python
# your code goes here

df.plot(kind='scatter', x='Age', y='WorkWeekHrs', figsize=(5, 5), color='red')

plt.title('Developers\' Age and Weekly Working Hours')
plt.xlabel('Age')
plt.ylabel('WorkWeekHrs')

plt.show()
```



Bubble Plots

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

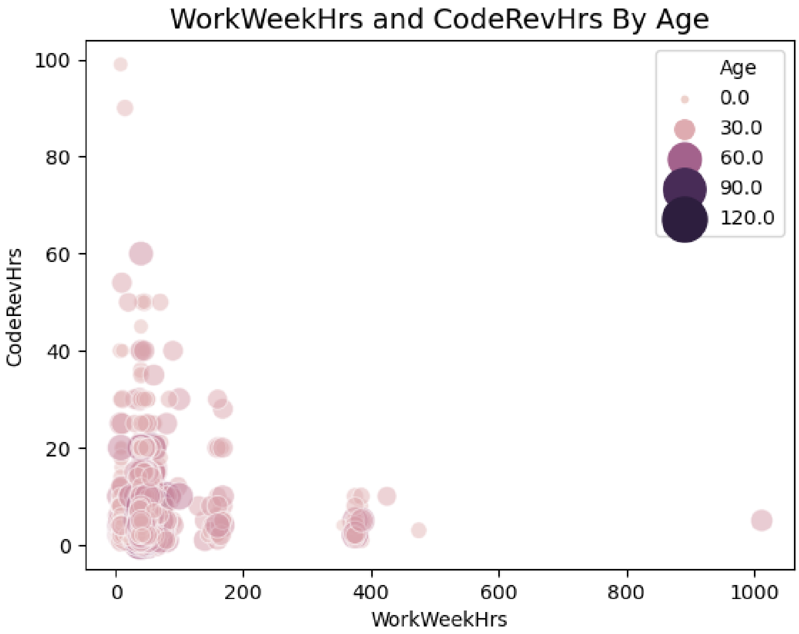
In [14]:

```
# your code goes here
QUERY = """
SELECT WorkWeekHrs, CodeRevHrs, Age FROM master
"""
df1=pd.read_sql_query(QUERY,conn)

sns.scatterplot(data=df1, x='WorkWeekHrs', y='CodeRevHrs', size='Age', hue='Age', alpha=0.5, sizes=(10, 500))

plt.title('WorkWeekHrs and CodeRevHrs By Age', size=14)
plt.xlabel('WorkWeekHrs', size=10)
plt.ylabel('CodeRevHrs', size=10)

plt.show()
```



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 [16]:

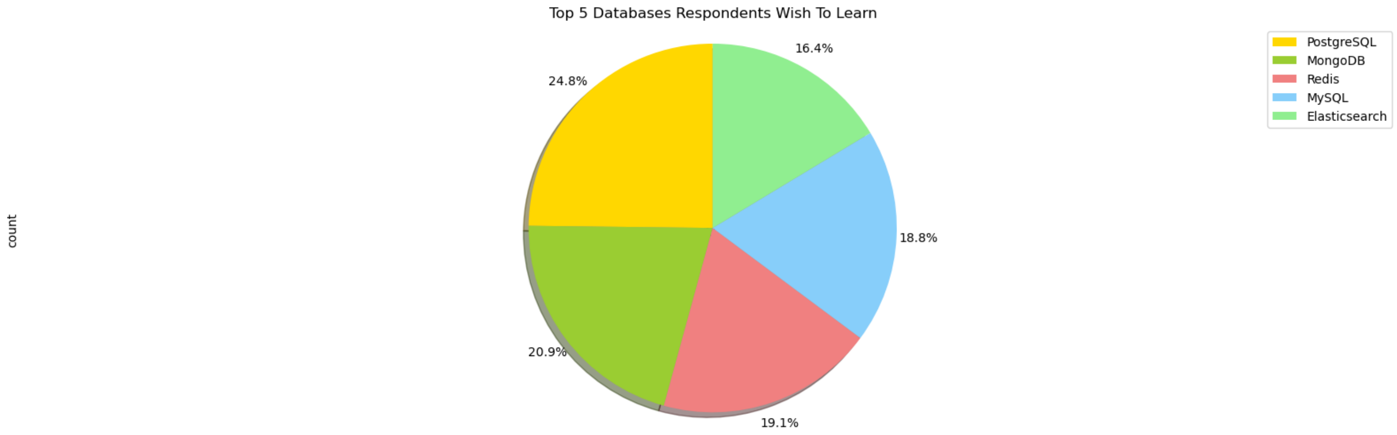
```
# your code goes here
QUERY = """
SELECT DatabaseDesireNextYear, COUNT(*) as count
from DatabaseDesireNextYear
group by DatabaseDesireNextYear
order by count(DatabaseDesireNextYear) DESC LIMIT 5
"""

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

colors_list=['gold', 'yellowgreen', 'lightcoral', 'lightskyblue', 'lightgreen', 'pink']

df['count'].plot(kind='pie', figsize=(20,6), autopct='%1.1f%%', labels=None, startangle=90, colors=colors_list, shadow=True, pctd

plt.legend(labels=df.index, loc='upper right')
plt.title('Top 5 Databases Respondents Wish To Learn')
plt.axis('equal')
plt.show()
```



Stacked Charts

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

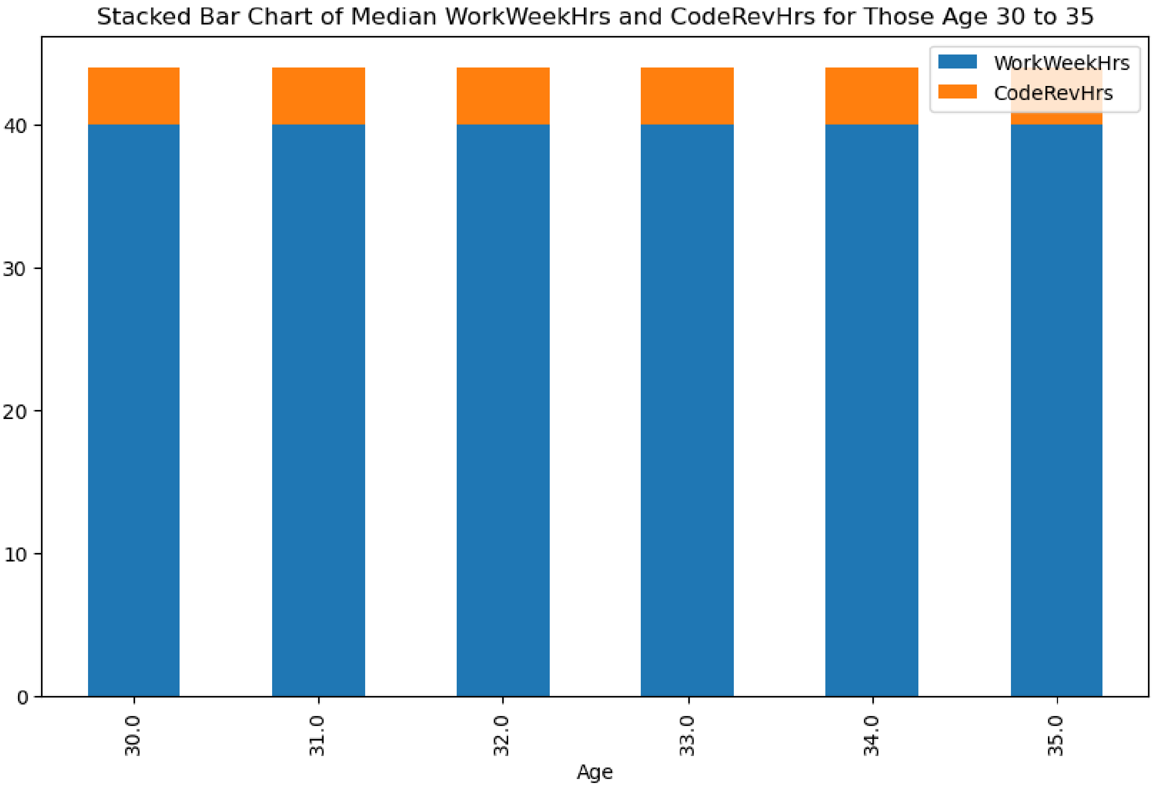
In [17]:

```
# your code goes here

QUERY = """
SELECT WorkWeekHrs, CodeRevHrs, Age FROM master
WHERE Age BETWEEN 30 AND 35
"""
df = pd.read_sql_query(QUERY,conn)
df1 = df.groupby('Age').median()

df1.plot(kind='bar', figsize=(10, 6), stacked=True)

plt.title('Stacked Bar Chart of Median WorkWeekHrs and CodeRevHrs for Those Age 30 to 35')
plt.show()
```



Visualizing comparison of data

Line Chart

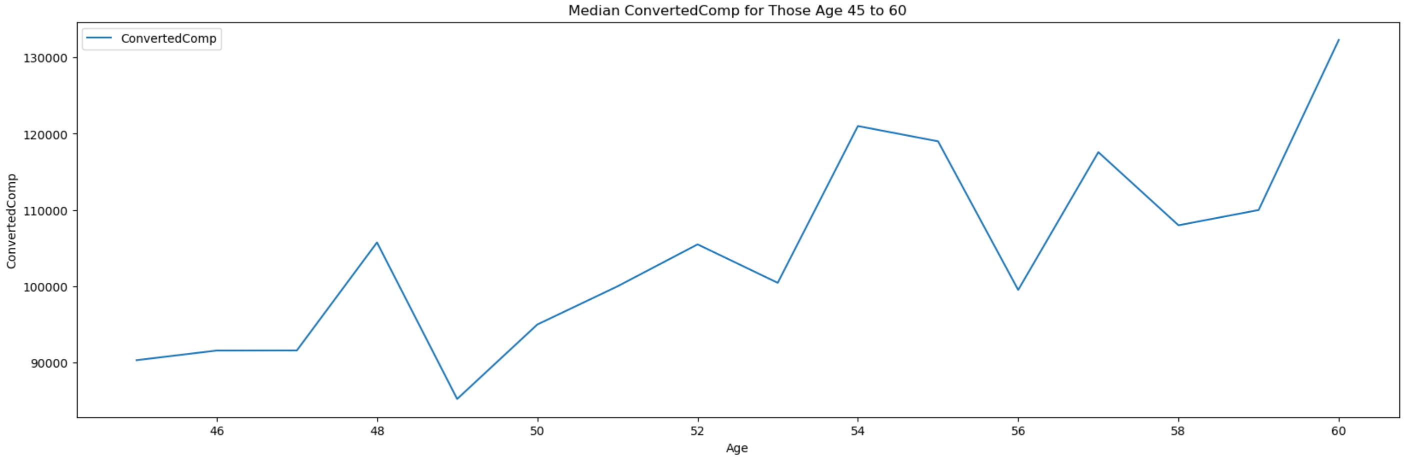
Plot the median `ConvertedComp` for all ages from 45 to 60.

In [18]:

```
# your code goes here
QUERY = """
SELECT ConvertedComp, Age FROM master
WHERE Age BETWEEN 45 AND 60
"""
df = pd.read_sql_query(QUERY,conn)
df1 = df.groupby('Age').median()

df1.plot(kind='line', figsize=(20, 6))

plt.title('Median ConvertedComp for Those Age 45 to 60')
plt.ylabel('ConvertedComp')
plt.show()
```



Bar Chart

Create a horizontal bar chart using column `MainBranch`.

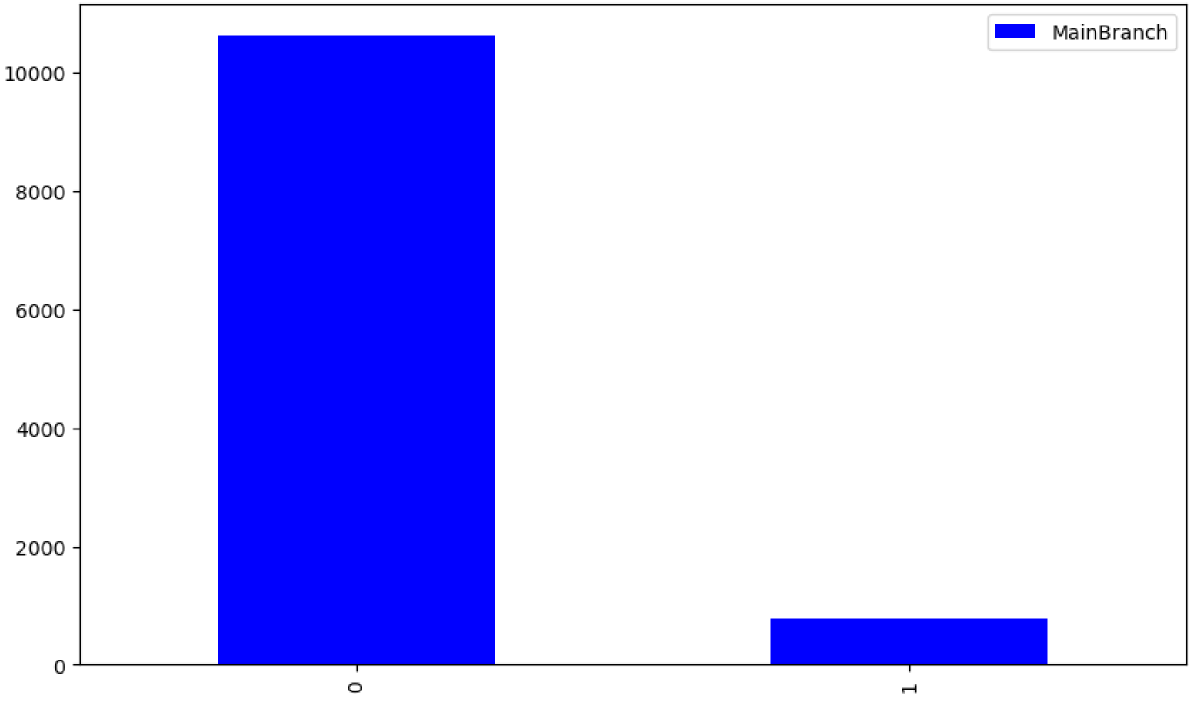
In [19]:

```
# your code goes here
QUERY = """
SELECT MainBranch, COUNT(*) as MainBranch
from master
group by MainBranch
"""

df=pd.read_sql_query(QUERY,conn)

df.plot(kind='bar', figsize=(10,6), color='blue')

plt.show()
```



Close the database connection.

In [20]:

```
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