

Case Study Overall EDA:

Objectives

1. Identify the most popular programming languages among IT professionals.
 2. Analyze average salaries and income statistics for IT professionals.
 3. Explore the age distribution among IT professionals.
 4. Provide statistical information on working hours for part-time and full-time IT professionals.
 5. Examine the relationship between income and various factors such as working hours, age, education, and other variables.
 6. Determine the most popular databases among IT professionals.
-

Tools:

Python, SQL, Postgres Libraries: numpy , pandas , seaborn , matplotlib.pyplot

Dataset:

Our data set is a survey works among IT professional , collected and published on Github in the link below. It has 11551 records and 84 columns. (11552, 85)

Columns: 'Respondent', 'MainBranch', 'Hobbyist', 'OpenSourcer', 'OpenSource', 'Employment', 'Country', 'Student', 'EdLevel', 'UndergradMajor', 'EduOther', 'OrgSize', 'DevType', 'YearsCode', 'Age1stCode', 'YearsCodePro', 'CareerSat', 'JobSat', 'MgrIdiot', 'MgrMoney', 'MgrWant', 'JobSeek', 'LastHireDate', 'LastInt', 'FizzBuzz', 'JobFactors', 'ResumeUpdate', 'CurrencySymbol', 'CurrencyDesc', 'CompTotal', 'CompFreq', 'ConvertedComp', 'WorkWeekHrs', 'WorkPlan', 'WorkChallenge', 'WorkRemote', 'WorkLoc', 'ImpSyn', 'CodeRev', 'CodeRevHrs', 'UnitTests', 'PurchaseHow', 'PurchaseWhat', 'LanguageWorkedWith', 'LanguageDesireNextYear', 'DatabaseWorkedWith', 'DatabaseDesireNextYear', 'PlatformWorkedWith', 'PlatformDesireNextYear', 'WebFrameWorkedWith', 'WebFrameDesireNextYear', 'MiscTechWorkedWith', 'MiscTechDesireNextYear', 'DevEnviron', 'OpSys', 'Containers', 'BlockchainOrg', 'BlockchainIs', 'BetterLife', 'ITperson', 'OffOn', 'SocialMedia', 'Extraversion', 'ScreenName', 'SOVisit1st', 'SOVisitFreq', 'SOVisitTo', 'SOFindAnswer', 'SOTimeSaved', 'SOHowMuchTime', 'SOAccount', 'SOPartFreq', 'SOJobs', 'EntTeams', 'SOComm', 'WelcomeChange', 'SONewContent', 'Age', 'Gender', 'Trans', 'Sexuality', 'Ethnicity', 'Dependents', 'SurveyLength', 'SurveyEase'], dtype='object')

Technical Objectives:

1. Connecting to API and import file
2. Save file into Data frame.
3. Cleaning the data (Nulls, outliers, rename, datatypes..)
4. Normalize the dataframe and create data modele where languages field moved into new data frame.

5. Transfer the new diminshinal data set into relational database or cloud. NOTE: so that include one piece of information per record each.
6. Performing EDA & Clean up.
7. Visualizing & Plotting

Extract Data:

In [416...

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
pd.set_option('display.max_columns', 100)
pd.set_option('display.max_rows', 10)
```

UsageError: unrecognized arguments: = '{:,.2f}'.format

In [417...

```
pd.options.display.float_format = '{:,.2f}'.format
```

In [307...

```
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-")
```

```
import pandas as pd from sqlalchemy import create_engine import psycopg2
```

```
db_username = "postgres" db_password ="Ghazzi4$" db_host = "localhost" db_port = "1975" db_name =
"survey" engine = create_engine(f'postgresql://{db_username}:{db_password}@{db_host}:
{db_port}/{db_name}')
```

```
table_name = 'survey_haha' df2.to_sql(table_name, engine, if_exists='append', index=False)
```

```
print("Data inserted successfully!")
```

In [308...

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11552 entries, 0 to 11551
Data columns (total 85 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Respondent                          11552 non-null  int64
1   MainBranch                          11552 non-null  object
2   Hobbyist                            11552 non-null  object
3   OpenSourcer                         11552 non-null  object
4   OpenSource                          11471 non-null  object
5   Employment                          11552 non-null  object
6   Country                             11552 non-null  object
7   Student                             11499 non-null  object
8   EdLevel                             11436 non-null  object
9   UndergradMajor                     10812 non-null  object
10  EduOther                             11388 non-null  object
11  OrgSize                             11454 non-null  object
12  DevType                             11485 non-null  object
13  YearsCode                           11543 non-null  object
14  Age1stCode                           11539 non-null  object
15  YearsCodePro                         11536 non-null  object
16  CareerSat                           11552 non-null  object
17  JobSat                              11551 non-null  object
18  MgrIdiot                            11054 non-null  object
```

19	MgrMoney	11050	non-null	object
20	MgrWant	11054	non-null	object
21	JobSeek	11552	non-null	object
22	LastHireDate	11552	non-null	object
23	LastInt	11129	non-null	object
24	FizzBuzz	11515	non-null	object
25	JobFactors	11549	non-null	object
26	ResumeUpdate	11511	non-null	object
27	CurrencySymbol	11552	non-null	object
28	CurrencyDesc	11552	non-null	object
29	CompTotal	10737	non-null	float64
30	CompFreq	11346	non-null	object
31	ConvertedComp	10730	non-null	float64
32	WorkWeekHrs	11427	non-null	float64
33	WorkPlan	11429	non-null	object
34	WorkChallenge	11384	non-null	object
35	WorkRemote	11544	non-null	object
36	WorkLoc	11520	non-null	object
37	ImpSyn	11547	non-null	object
38	CodeRev	11551	non-null	object
39	CodeRevHrs	9083	non-null	float64
40	UnitTests	11523	non-null	object
41	PurchaseHow	11354	non-null	object
42	PurchaseWhat	11514	non-null	object
43	LanguageWorkedWith	11541	non-null	object
44	LanguageDesireNextYear	11415	non-null	object
45	DatabaseWorkedWith	11096	non-null	object
46	DatabaseDesireNextYear	10497	non-null	object
47	PlatformWorkedWith	11130	non-null	object
48	PlatformDesireNextYear	10991	non-null	object
49	WebFrameWorkedWith	10139	non-null	object
50	WebFrameDesireNextYear	9918	non-null	object
51	MiscTechWorkedWith	9343	non-null	object
52	MiscTechDesireNextYear	10078	non-null	object
53	DevEnviron	11523	non-null	object
54	OpSys	11518	non-null	object
55	Containers	11470	non-null	object
56	BlockchainOrg	9198	non-null	object
57	BlockchainIs	8915	non-null	object
58	BetterLife	11452	non-null	object
59	ITperson	11517	non-null	object
60	OffOn	11514	non-null	object
61	SocialMedia	11251	non-null	object
62	Extraversion	11532	non-null	object
63	ScreenName	11039	non-null	object
64	S0Visit1st	11227	non-null	object
65	S0VisitFreq	11547	non-null	object
66	S0VisitTo	11551	non-null	object
67	S0FindAnswer	11549	non-null	object
68	S0TimeSaved	11501	non-null	object
69	S0HowMuchTime	9616	non-null	object
70	S0Account	11551	non-null	object
71	S0PartFreq	10404	non-null	object
72	S0Jobs	11546	non-null	object
73	EntTeams	11547	non-null	object
74	S0Comm	11552	non-null	object
75	WelcomeChange	11463	non-null	object
76	S0NewContent	9557	non-null	object
77	Age	11255	non-null	float64
78	Gender	11477	non-null	object
79	Trans	11429	non-null	object
80	Sexuality	11005	non-null	object
81	Ethnicity	10869	non-null	object
82	Dependents	11408	non-null	object
83	SurveyLength	11533	non-null	object
84	SurveyEase	11538	non-null	object

dtypes: float64(5), int64(1), object(79)

memory usage: 7.5+ MB

Drops unwanted columns(Filter columns)

Drop one or more columns using df.drop()

```
In [309... columns_to_drop=[ 'Hobbyist', 'OpenSourcer', 'OpenSource',  
                  'CareerSat', 'JobSat', 'MgrIdiot', 'MgrMoney',  
                  'MgrWant', 'JobSeek', 'LastHireDate', 'LastInt', 'FizzBuzz',  
                  'JobFactors', 'ResumeUpdate', 'ConvertedComp', 'WorkPlan',  
                  'WorkChallenge', 'ImpSyn', 'CodeRev',  
                  'CodeRevHrs', 'UnitTests', 'PurchaseHow', 'PurchaseWhat',  
                  'LanguageDesireNextYear',  
                  'DatabaseDesireNextYear', 'PlatformWorkedWith',  
                  'PlatformDesireNextYear', 'WebFrameWorkedWith',  
                  'WebFrameDesireNextYear', 'MiscTechWorkedWith',  
                  'MiscTechDesireNextYear', 'DevEnviron', 'OpSys', 'Containers',  
                  'BlockchainOrg', 'BlockchainIs', 'BetterLife', 'ITperson', 'OffOn',  
                  'SocialMedia', 'Extraversion', 'ScreenName', 'SOVisit1st',  
                  'SOVisitFreq', 'SOVisitTo', 'SOFindAnswer', 'SOTimeSaved',  
                  'SOHowMuchTime', 'SOAccount', 'SOPartFreq', 'SOJobs', 'EntTeams',  
                  'SOComm', 'WelcomeChange', 'SONewContent', 'Trans',  
                  'Sexuality', 'Dependents', 'SurveyLength']
```

```
In [310... df= df.drop(columns=columns_to_drop, axis=1)
```

```
In [315... # Split the 'Employment' column into two new columns ('To_delete' and 'Employment')  
df[['to_delete', 'Employment']] =df['Employment'].str.split(" ",expand=True)
```

```
In [316... df= df.drop(columns= 'to_delete', axis=1)
```

Now, it is time for cleaning the data

Remove Duplicates

```
In [312... df.duplicated().sum()
```

Out[312... 154

```
In [313... df=df.drop_duplicates()
```

```
df.set_index(df['Respondent'])
```

Average salaries and income statistics for IT professionals.

Standrize income data

```
In [419... df.CompFreq.value_counts()
```

```
Out[419... Yearly      2662  
Monthly     336  
Weekly      196  
Name: CompFreq, dtype: int64
```

```
In [393... #Create new earning column to standrize pay on annual base:
mult = {'Monthly': 12, 'Weekly': 52, 'Yearly': 1}
df['earning'] = df['CompTotal'] * df['CompFreq'].map(mult).fillna(1).astype(float)
```

Filter Data into compensations in USD only

```
In [394... df=df[df['CurrencySymbol']=='USD']
```

```
In [395... df.CurrencySymbol.value_counts()
```

```
Out[395... USD      3194
Name: CurrencySymbol, dtype: int64
```

Check/remove Nulls / Missing data

```
In [398... # remove null values from the three fields of interest
df[['CompTotal', 'CompFreq', 'Employment', 'earning']].isna().sum()
```

```
Out[398... CompTotal      0
CompFreq      0
Employment     0
earning        0
dtype: int64
```

```
In [397... df=df.dropna(how='any', subset=['CompTotal', 'CompFreq', 'Employment', 'earning'])
# Filter for rows where has zero compensation value
df = df[df['CompTotal']!=0 ]
```

```
In [399... df1=df[df['Employment']=='full-time' ]
df2=df[df['Employment']=='part-time' ]
```

Another way to standrize income data

```
earning = [] for _, row in df[['CompFreq', 'CompTotal']].iterrows(): x, y = row['CompFreq'], row['CompTotal']
if x == 'Monthly': earning.append(y*12) elif x == 'Weekly': earning.append(y*52) else: earning.append(y)
```

```
df['earning']=earning
```

Outlier / Extremes in the Compansation Data

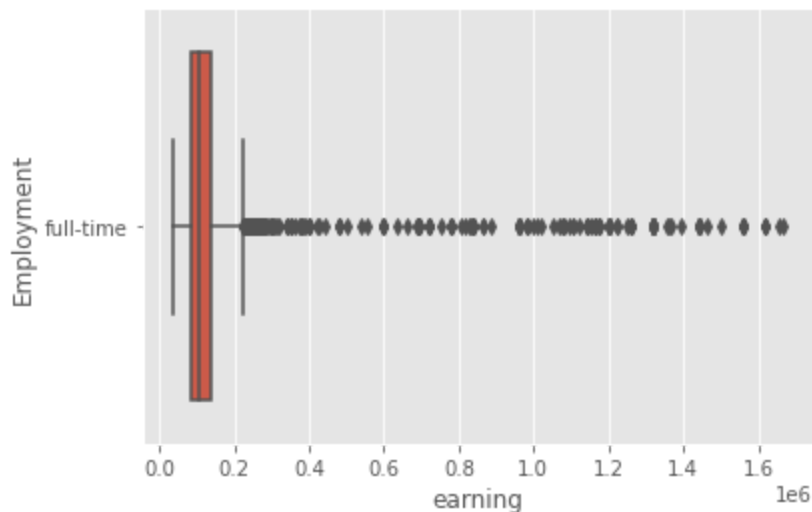
```
In [406... Number=[]
q_low = df1['earning'].quantile(0.05)
q_hi  = df1['earning'].quantile(0.95)
value= df1['earning'].median()
for x in df1['earning']:
    if q_hi > x > q_low :
        Number.append(x)
    elif df1['earning'].isnull().any():
        Number.append(q_low)
    else:
        Number.append(median_value)
df1['earning']=Number
```

```
<ipython-input-406-6d5768180d78>:12: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
df1['earning']=Number
```

```
In [418... df1['earning'].describe()
```

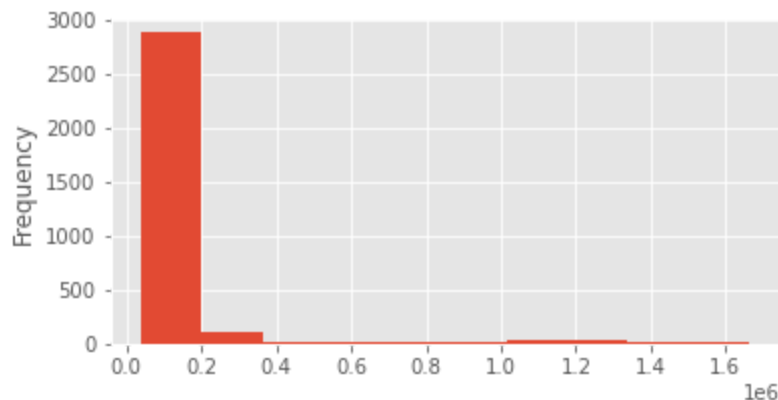
```
Out[418... count      3,125.00  
mean      152,928.26  
std       207,792.90  
min       38,272.00  
25%       84,000.00  
50%      106,000.00  
75%      140,000.00  
max      1,664,000.00  
Name: earning, dtype: float64
```

```
In [414... sns.boxplot(x='earning', y='Employment', data=df1)  
plt.show()
```



```
In [429... df1.earning.plot(kind='hist',bins=10,figsize=(6,3))
```

```
Out[429... <AxesSubplot:ylabel='Frequency'>
```

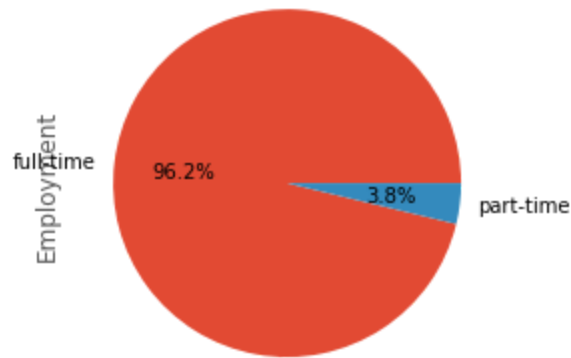


```
In [401... df1.sort_values(by='earning', ascending=False)
```

	Respondent	MainBranch	Employment	Country	Student	EdLevel	UndergradMajor	EduOther	
9452	20388	I am a developer by profession	full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	A natural science (ex. biology, chemistry, phy...	Taken an online course in programming or softw...	1 en
582	1232	I am a developer by profession	full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...	Taken an online course in programming or softw...	en
6769	14466	I am a developer by profession	full-time	United States	No	Other doctoral degree (Ph.D, Ed.D., etc.)	Computer science, computer engineering, or sof...	Taught yourself a new language, framework, or ...	1 en
6044	12920	I am a developer by profession	full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...	Taught yourself a new language, framework, or ...	1 en
3949	8358	I am a developer by profession	full-time	United States	No	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	Taken an online course in programming or softw...	en
...	...	...	...	...	...	...	...	...	
2496	5107	I am a developer by profession	full-time	United States	Yes, full-time	Master's degree (MA, MS, M.Eng., MBA, etc.)	Computer science, computer engineering, or sof...	Contributed to open source software	en
2376	4852	I am a developer by profession	full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Computer science, computer engineering, or sof...	Taken an online course in programming or softw...	en
3070	6419	I am a developer by profession	full-time	United States	Yes, full-time	Bachelor's degree (BA, BS, B.Eng., etc.)	A business discipline (ex. accounting, finance...	Participated in a full-time developer training...	en
4688	9926	I am not primarily a developer, but I write co...	full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	A business discipline (ex. accounting, finance...	Participated in a full-time developer training...	en
3660	7742	I am not primarily a developer, but I write co...	full-time	United States	No	Bachelor's degree (BA, BS, B.Eng., etc.)	Another engineering discipline (ex. civil, ele...	Taken an online course in programming or softw...	en

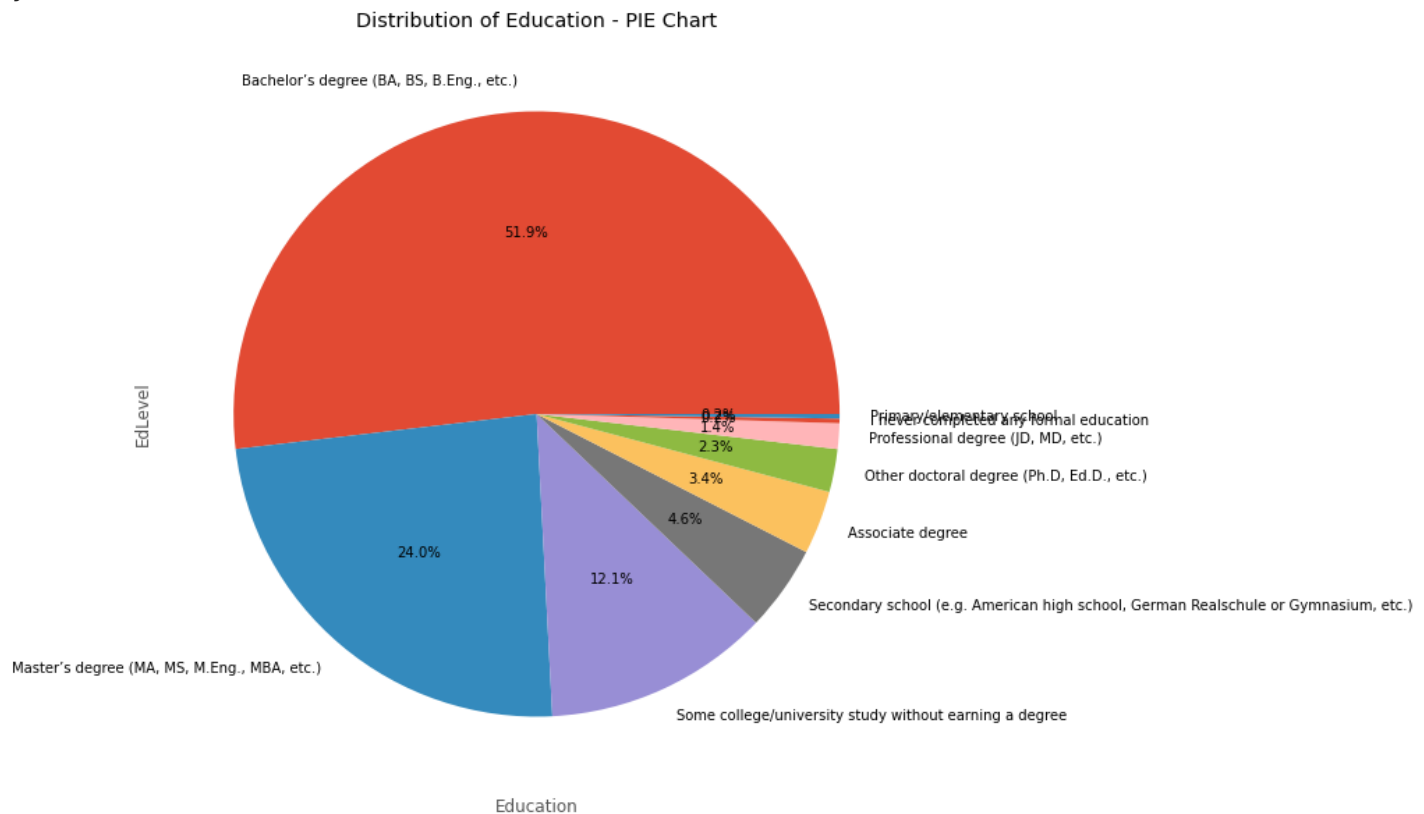
```
In [210... df.Employment.value_counts().plot(kind='pie', autopct='%1.1f%%')
```

```
Out[210... <AxesSubplot:ylabel='Employment'>
```



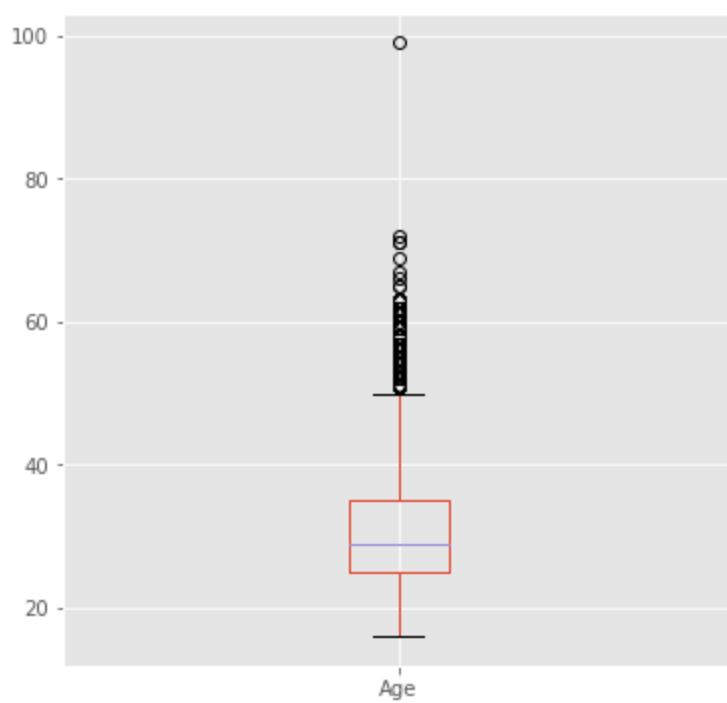
```
In [211... plt.style.use('ggplot')
plt.figure(figsize=(12, 10))
plt.title('Distribution of Education - PIE Chart')
plt.xlabel('Education')
df.EdLevel.value_counts().plot(kind='pie', legend=False, autopct='%1.1f%%')
```

```
Out[211... <AxesSubplot:title={'center':'Distribution of Education - PIE Chart'}, xlabel='Education',
ylabel='EdLevel'>
```



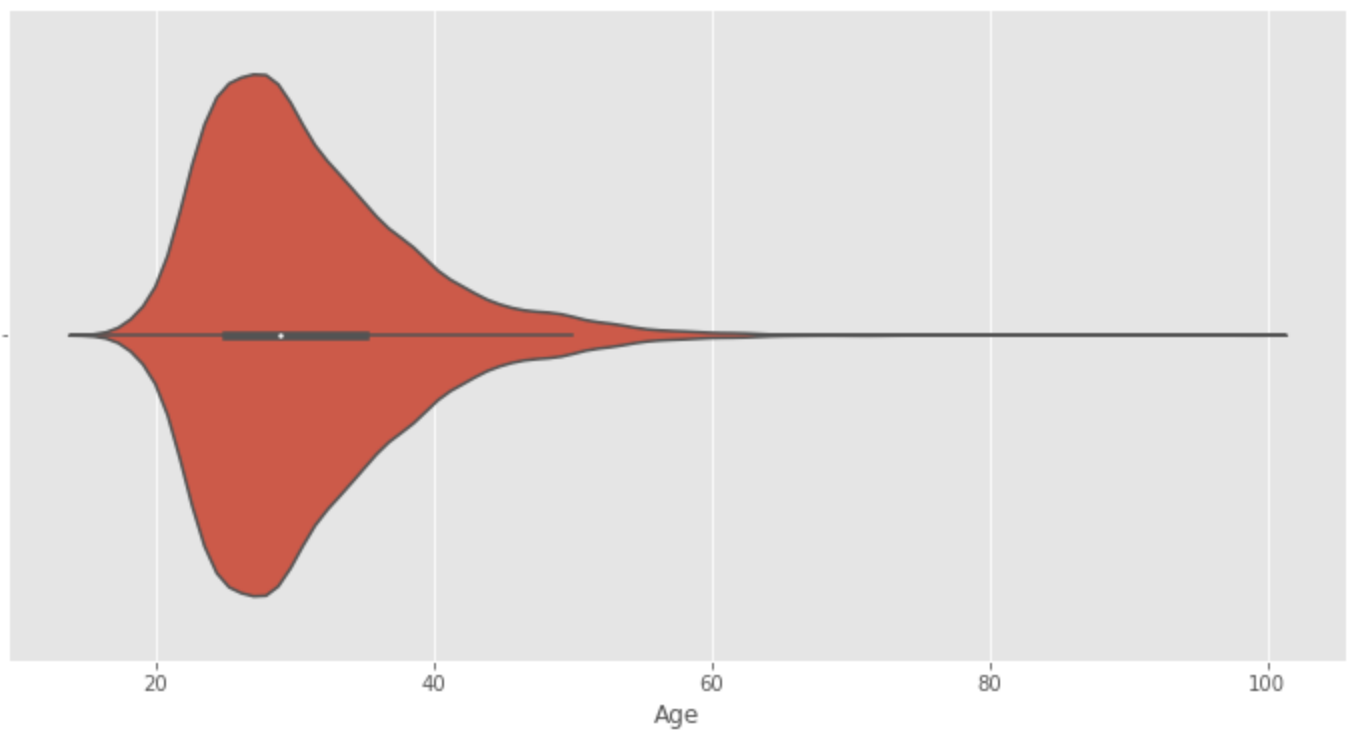
```
In [212... #plt.figure(figsize=(12, 10))
df.Age.plot(kind='box', figsize=(6,6))
```

```
Out[212... <AxesSubplot:>
```



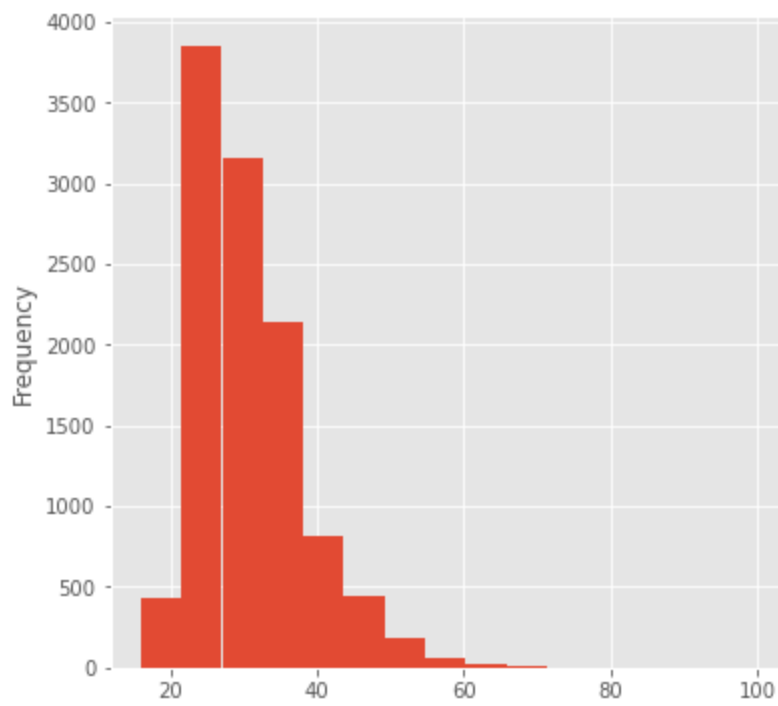
```
In [213... plt.figure(figsize=(12,6))  
sns.violinplot(x=df['Age'])
```

```
Out[213... <AxesSubplot:xlabel='Age'>
```



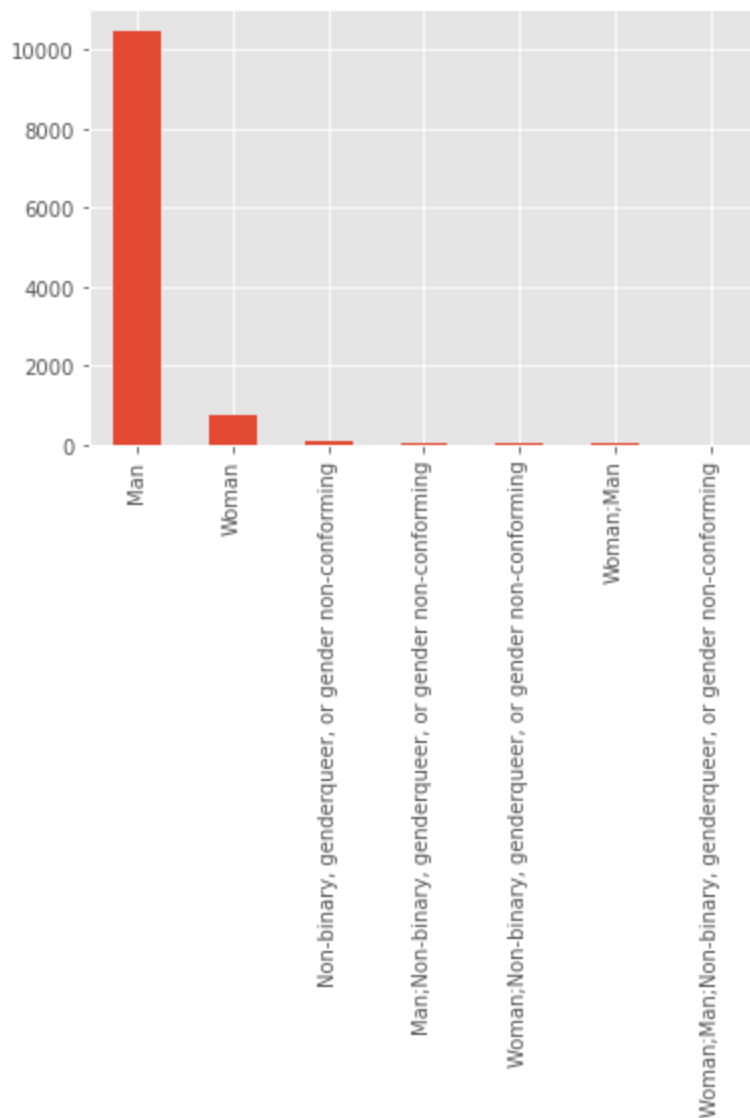
```
In [214... df.Age.plot(kind='hist',bins=15,figsize=(6,6))
```

```
Out[214... <AxesSubplot:ylabel='Frequency'>
```



In [215... `df.Gender.value_counts().plot(kind='bar')`

Out [215... `<AxesSubplot:>`



```
In [216... df.LanguageWorkedWith.value_counts().head(10)
```

```
Out[216... C#;HTML/CSS;JavaScript;SQL                274
C#;HTML/CSS;JavaScript;SQL;TypeScript      219
HTML/CSS;JavaScript;PHP;SQL                206
HTML/CSS;JavaScript;TypeScript             118
HTML/CSS;JavaScript                       107
Bash/Shell/PowerShell;C#;HTML/CSS;JavaScript;SQL;TypeScript 106
HTML/CSS;JavaScript;PHP                    104
HTML/CSS;Java;JavaScript;SQL               99
Bash/Shell/PowerShell;C#;HTML/CSS;JavaScript;SQL           97
C#;HTML/CSS;JavaScript;TypeScript           81
Name: LanguageWorkedWith, dtype: int64
```

```
In [217... df.Student.value_counts().plot(kind='pie', autopct='%1.1f%%')
```

```
Out[217... <AxesSubplot:ylabel='Student'>
```

