In [24]:

```
import pandas as pd
import numpy as np
```

In [3]:

```
people={
    "first":["Corey","Jane","John"],
    "last":["Schafer","Doe","Doe"],
    "email":["CoreyMSchafer@gmail.com","JaneDoe@gmail.com","JohnDoe@gmail.com"]
}
```

In [4]:

```
df=pd.DataFrame(people)
```

In [5]:

```
df
```

Out[5]:

	first	last	email
0	Corey	Schafer	CoreyMSchafer@gmail.com
1	Jane	Doe	JaneDoe@gmail.com
2	John	Doe	JohnDoe@gmail.com

In [6]:

```
df.columns
```

Out[6]:

```
Index(['first', 'last', 'email'], dtype='object')
```

In [9]:

```
df.columns=['first_name','last_name','email']
```

In [10]:

```
df
```

Out[10]:

	first_name	last_name	email
0	Corey	Schafer	CoreyMSchafer@gmail.com
1	Jane	Doe	JaneDoe@gmail.com
2	John	Doe	JohnDoe@gmail.com

In [11]:

```
df.columns=[x.upper() for x in df.columns]
```

In [12]:

```
df
```

Out[12]:

	FIRST_NAME	LAST_NAME	EMAIL
0	Corey	Schafer	CoreyMSchafer@gmail.com
1	Jane	Doe	JaneDoe@gmail.com
2	John	Doe	JohnDoe@gmail.com

In [13]:

```
df.columns=df.columns.str.replace('_',' ')
```

In [14]:

```
df
```

Out[14]:

```
   FIRST NAME  LAST NAME                   EMAIL
0       Corey    Schafer  CoreyMSchafer@gmail.com
1        Jane        Doe        JaneDoe@gmail.com
2        John        Doe        JohnDoe@gmail.com
```

In [15]:

```
df.columns=[x.lower() for x in df.columns]
```

In [16]:

```
df
```

Out[16]:

```
   first name  last name                   email
0       Corey    Schafer  CoreyMSchafer@gmail.com
1        Jane        Doe        JaneDoe@gmail.com
2        John        Doe        JohnDoe@gmail.com
```

In [17]:

```
df.columns=df.columns.str.replace(' ','_')
```

In [18]:

```
df
```

Out[18]:

```
   first_name  last_name                   email
0       Corey    Schafer  CoreyMSchafer@gmail.com
1        Jane        Doe        JaneDoe@gmail.com
2        John        Doe        JohnDoe@gmail.com
```

In [20]:

```
# want to change specific columns
df.rename(columns={'first_name':'first','last_name':'last'}, inplace=True)
```

In [21]:

```
df
```

Out[21]:

```
   first     last                    email
0  Corey  Schafer  CoreyMSchafer@gmail.com
1   Jane      Doe        JaneDoe@gmail.com
2   John      Doe        JohnDoe@gmail.com
```

In [22]:

```
df.loc[2]=['John','Smith','JohnDoe@email.com']
```

In [23]:

```
df
```

Out[23]:

	first	last	email
0	Corey	Schafer	CoreyMSchafer@gmail.com
1	Jane	Doe	JaneDoe@gmail.com
2	John	Smith	JohnDoe@email.com

In [24]:

```
df.loc[2,['last','email']]=['Doe','JohnDoe@gmail.com']
```

In [25]:

```
df
```

Out[25]:

	first	last	email
0	Corey	Schafer	CoreyMSchafer@gmail.com
1	Jane	Doe	JaneDoe@gmail.com
2	John	Doe	JohnDoe@gmail.com

In [26]:

```
df.loc[2,'last']='Smith'
```

In [27]:

```
df
```

Out[27]:

	first	last	email
0	Corey	Schafer	CoreyMSchafer@gmail.com
1	Jane	Doe	JaneDoe@gmail.com
2	John	Smith	JohnDoe@gmail.com

In [28]:

```
filt=(df['email']=='JohnDoe@gmail.com')
df[filt]
```

Out[28]:

	first	last	email
2	John	Smith	JohnDoe@gmail.com

In [29]:

```
df[filt]['last']='Smith'
```

```
<ipython-input-29-5c4ea8a4e6cd>:1: 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/sta
ble/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pyd
ata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-c
opy)
  df[filt]['last']='Smith'
```

In [32]:

```
df['email']=df['email'].str.lower()
```

In [33]:

```
df
```

Out[33]:

	first	last	email
0	Corey	Schafer	coreymschafer@gmail.com
1	Jane	Doe	janedoe@gmail.com
2	John	Smith	johndoe@gmail.com

In [36]:

```
#DateTime index and analysis
import pandas as pd
d_parser = lambda x: pd.datetime.strptime(x, '%Y-%m-%d %I-%p')
df = pd.read_csv(r'C:\Users\e16379\Desktop\ETH_1h.csv',parse_dates=['Date'], date
df.head()
```

```
<ipython-input-36-3e4c694d82f9>:2: FutureWarning: The pandas.datetime class is
deprecated and will be removed from pandas in a future version. Import from dat
etime module instead.
  d_parser = lambda x: pd.datetime.strptime(x, '%Y-%m-%d %I-%p')
```

Out[36]:

	Date	Symbol	Open	High	Low	Close	Volume
0	2020-03-13 20:00:00	ETHUSD	129.94	131.82	126.87	128.71	1940673.93
1	2020-03-13 19:00:00	ETHUSD	119.51	132.02	117.10	129.94	7579741.09
2	2020-03-13 18:00:00	ETHUSD	124.47	124.85	115.50	119.51	4898735.81
3	2020-03-13 17:00:00	ETHUSD	124.08	127.42	121.63	124.47	2753450.92
4	2020-03-13 16:00:00	ETHUSD	124.85	129.51	120.17	124.08	4461424.71

```
In [37]: df['Date'].dt.day_name()
```

```
Out[37]: 0          Friday
         1          Friday
         2          Friday
         3          Friday
         4          Friday
                     ...
         23669    Saturday
         23670    Saturday
         23671    Saturday
         23672    Saturday
         23673    Saturday
         Name: Date, Length: 23674, dtype: object
```

```
In [10]: df.loc[0, 'Date'].day_name()
```

```
Out[10]: 'Friday'
```

```
In [45]:  df['Date'].dt.day_name()
```

```
Out[45]: 0          Friday
         1          Friday
         2          Friday
         3          Friday
         4          Friday
                     ...
         23669    Saturday
         23670    Saturday
         23671    Saturday
         23672    Saturday
         23673    Saturday
         Name: Date, Length: 23674, dtype: object
```

```
In [11]: df['DayOfWeek']=df['Date'].dt.day_name()
```

In [12]:
```
df
```

Out[12]:

	Date	Symbol	Open	High	Low	Close	Volume	DayOfWeek
0	2020-03-13 20:00:00	ETHUSD	129.94	131.82	126.87	128.71	1940673.93	Friday
1	2020-03-13 19:00:00	ETHUSD	119.51	132.02	117.10	129.94	7579741.09	Friday
2	2020-03-13 18:00:00	ETHUSD	124.47	124.85	115.50	119.51	4898735.81	Friday
3	2020-03-13 17:00:00	ETHUSD	124.08	127.42	121.63	124.47	2753450.92	Friday
4	2020-03-13 16:00:00	ETHUSD	124.85	129.51	120.17	124.08	4461424.71	Friday
...	...	...	...	...	...	...	...	...
23669	2017-07-01 15:00:00	ETHUSD	265.74	272.74	265.00	272.57	1500282.55	Saturday
23670	2017-07-01 14:00:00	ETHUSD	268.79	269.90	265.00	265.74	1702536.85	Saturday
23671	2017-07-01 13:00:00	ETHUSD	274.83	274.93	265.00	268.79	3010787.99	Saturday
23672	2017-07-01 12:00:00	ETHUSD	275.01	275.01	271.00	274.83	824362.87	Saturday
23673	2017-07-01 11:00:00	ETHUSD	279.98	279.99	272.10	275.01	679358.87	Saturday

23674 rows × 8 columns

In [63]:
```
df['Date'].min()
```

Out[63]:
```
Timestamp('2017-07-01 11:00:00')
```

In [64]:
```
df['Date'].max()
```

Out[64]:
```
Timestamp('2020-03-13 20:00:00')
```

In [65]:
```

df['Date'].max() - df['Date'].min()
```

Out[65]:
```
Timedelta('986 days 09:00:00')
```

In [13]:
```
filt = (df['Date'] >= pd.to_datetime('2019-01-01')) & (df['Date'] < pd.to_datetim
df.loc[filt]
```

Out[13]:

	Date	Symbol	Open	High	Low	Close	Volume	DayOfWeek
1749	2019-12-31 23:00:00	ETHUSD	128.33	128.69	128.14	128.54	440678.91	Tuesday
1750	2019-12-31 22:00:00	ETHUSD	128.38	128.69	127.95	128.33	554646.02	Tuesday
1751	2019-12-31 21:00:00	ETHUSD	127.86	128.43	127.72	128.38	350155.69	Tuesday
1752	2019-12-31 20:00:00	ETHUSD	127.84	128.34	127.71	127.86	428183.38	Tuesday
1753	2019-12-31 19:00:00	ETHUSD	128.69	128.69	127.60	127.84	1169847.84	Tuesday
...	...	...	...	...	...	...	...	...
10504	2019-01-01 04:00:00	ETHUSD	130.75	133.96	130.74	131.96	2791135.37	Tuesday
10505	2019-01-01 03:00:00	ETHUSD	130.06	130.79	130.06	130.75	503732.63	Tuesday
10506	2019-01-01 02:00:00	ETHUSD	130.79	130.88	129.55	130.06	838183.43	Tuesday
10507	2019-01-01 01:00:00	ETHUSD	131.62	131.62	130.77	130.79	434917.99	Tuesday
10508	2019-01-01 00:00:00	ETHUSD	130.53	131.91	130.48	131.62	1067136.21	Tuesday

8760 rows × 8 columns

In [38]:
```
df.set_index('Date', inplace=True)
```

In [39]:
```
df
```

Out[39]:

	Symbol	Open	High	Low	Close	Volume
Date						
2020-03-13 20:00:00	ETHUSD	129.94	131.82	126.87	128.71	1940673.93
2020-03-13 19:00:00	ETHUSD	119.51	132.02	117.10	129.94	7579741.09
2020-03-13 18:00:00	ETHUSD	124.47	124.85	115.50	119.51	4898735.81
2020-03-13 17:00:00	ETHUSD	124.08	127.42	121.63	124.47	2753450.92
2020-03-13 16:00:00	ETHUSD	124.85	129.51	120.17	124.08	4461424.71
...	...	...	...	...	...	...
2017-07-01 15:00:00	ETHUSD	265.74	272.74	265.00	272.57	1500282.55
2017-07-01 14:00:00	ETHUSD	268.79	269.90	265.00	265.74	1702536.85
2017-07-01 13:00:00	ETHUSD	274.83	274.93	265.00	268.79	3010787.99
2017-07-01 12:00:00	ETHUSD	275.01	275.01	271.00	274.83	824362.87
2017-07-01 11:00:00	ETHUSD	279.98	279.99	272.10	275.01	679358.87

23674 rows × 6 columns

```
In [83]: df
```

```
Out[83]:
                     Symbol   Open    High    Low     Close       Volume
                Date
 2020-03-13 20:00:00 ETHUSD  129.94  131.82  126.87  128.71  1940673.93
 2020-03-13 19:00:00 ETHUSD  119.51  132.02  117.10  129.94  7579741.09
 2020-03-13 18:00:00 ETHUSD  124.47  124.85  115.50  119.51  4898735.81
 2020-03-13 17:00:00 ETHUSD  124.08  127.42  121.63  124.47  2753450.92
 2020-03-13 16:00:00 ETHUSD  124.85  129.51  120.17  124.08  4461424.71
                 ...    ...     ...     ...     ...     ...         ...
 2017-07-01 15:00:00 ETHUSD  265.74  272.74  265.00  272.57  1500282.55
 2017-07-01 14:00:00 ETHUSD  268.79  269.90  265.00  265.74  1702536.85
 2017-07-01 13:00:00 ETHUSD  274.83  274.93  265.00  268.79  3010787.99
 2017-07-01 12:00:00 ETHUSD  275.01  275.01  271.00  274.83   824362.87
 2017-07-01 11:00:00 ETHUSD  279.98  279.99  272.10  275.01   679358.87

23674 rows × 6 columns
```

```
In [85]: df.loc['2020']
```

```
Out[85]:
                     Symbol   Open    High    Low     Close       Volume
                Date
 2020-03-13 20:00:00 ETHUSD  129.94  131.82  126.87  128.71  1940673.93
 2020-03-13 19:00:00 ETHUSD  119.51  132.02  117.10  129.94  7579741.09
 2020-03-13 18:00:00 ETHUSD  124.47  124.85  115.50  119.51  4898735.81
 2020-03-13 17:00:00 ETHUSD  124.08  127.42  121.63  124.47  2753450.92
 2020-03-13 16:00:00 ETHUSD  124.85  129.51  120.17  124.08  4461424.71
                 ...    ...     ...     ...     ...     ...         ...
 2020-01-01 04:00:00 ETHUSD  129.57  130.00  129.50  129.56   702786.82
 2020-01-01 03:00:00 ETHUSD  130.37  130.44  129.38  129.57   496704.23
 2020-01-01 02:00:00 ETHUSD  130.14  130.50  129.91  130.37   396315.72
 2020-01-01 01:00:00 ETHUSD  128.34  130.14  128.32  130.14   635419.40
 2020-01-01 00:00:00 ETHUSD  128.54  128.54  128.12  128.34   245119.91

1749 rows × 6 columns
```

```
In [86]: df.loc['2020-01':'2020-02']
```

Out[86]:

	Symbol	Open	High	Low	Close	Volume
Date						
2020-02-29 23:00:00	ETHUSD	223.35	223.58	216.83	217.31	1927939.88
2020-02-29 22:00:00	ETHUSD	223.48	223.59	222.14	223.35	535998.57
2020-02-29 21:00:00	ETHUSD	224.63	225.14	222.74	223.48	561158.03
2020-02-29 20:00:00	ETHUSD	225.31	225.33	223.50	224.63	511648.65
2020-02-29 19:00:00	ETHUSD	225.09	225.85	223.87	225.31	1250856.20
...	...	...	...	...	...	...
2020-01-01 04:00:00	ETHUSD	129.57	130.00	129.50	129.56	702786.82
2020-01-01 03:00:00	ETHUSD	130.37	130.44	129.38	129.57	496704.23
2020-01-01 02:00:00	ETHUSD	130.14	130.50	129.91	130.37	396315.72
2020-01-01 01:00:00	ETHUSD	128.34	130.14	128.32	130.14	635419.40
2020-01-01 00:00:00	ETHUSD	128.54	128.54	128.12	128.34	245119.91

1440 rows × 6 columns

```
In [88]: df.loc['2020-01':'2020-02']['Close'].mean()

Out[88]: 195.16559027777814
```

```
In [90]: df.loc['2020-01-01']['High'].max()

Out[90]: 132.68
```

```
In [16]: highs = df['High'].resample('D').max()
In [17]: df
```

Out[17]:

	Symbol	Open	High	Low	Close	Volume	DayOfWeek
Date							
2020-03-13 20:00:00	ETHUSD	129.94	131.82	126.87	128.71	1940673.93	Friday
2020-03-13 19:00:00	ETHUSD	119.51	132.02	117.10	129.94	7579741.09	Friday
2020-03-13 18:00:00	ETHUSD	124.47	124.85	115.50	119.51	4898735.81	Friday
2020-03-13 17:00:00	ETHUSD	124.08	127.42	121.63	124.47	2753450.92	Friday
2020-03-13 16:00:00	ETHUSD	124.85	129.51	120.17	124.08	4461424.71	Friday
...	...	...	...	...	...	...	...
2017-07-01 15:00:00	ETHUSD	265.74	272.74	265.00	272.57	1500282.55	Saturday
2017-07-01 14:00:00	ETHUSD	268.79	269.90	265.00	265.74	1702536.85	Saturday
2017-07-01 13:00:00	ETHUSD	274.83	274.93	265.00	268.79	3010787.99	Saturday
2017-07-01 12:00:00	ETHUSD	275.01	275.01	271.00	274.83	824362.87	Saturday
2017-07-01 11:00:00	ETHUSD	279.98	279.99	272.10	275.01	679358.87	Saturday

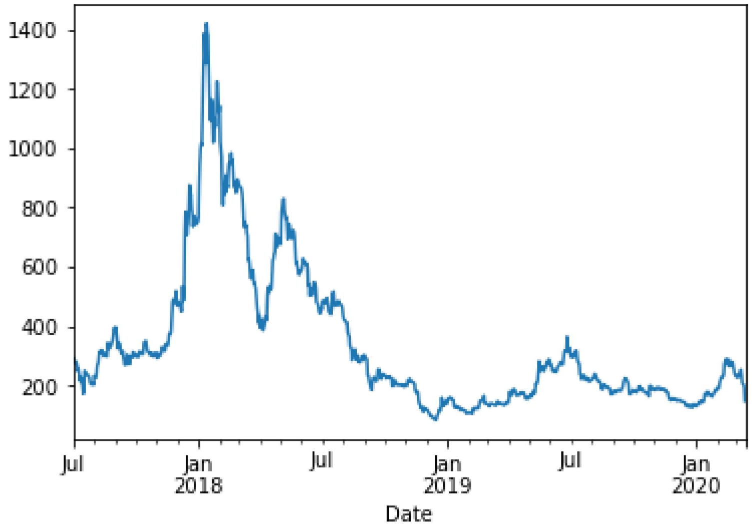
23674 rows × 7 columns

```
In [18]: highs['2020-01-01']
```

```
Out[18]: 132.68
```

```
In [19]: %matplotlib inline
         highs.plot()
```

```
Out[19]: <AxesSubplot:xlabel='Date'>
```



```
In [20]: df.resample('W').mean()
```

```
Out[20]:
                 Open        High         Low       Close        Volume
      Date
2017-07-02 268.066486  271.124595  264.819730  268.202162  2.185035e+06
2017-07-09 261.337024  262.872917  259.186190  261.062083  1.337349e+06
2017-07-16 196.193214  199.204405  192.722321  195.698393  2.986756e+06
2017-07-23 212.351429  215.779286  209.126310  212.783750  4.298593e+06
2017-07-30 203.496190  205.110357  201.714048  203.309524  1.581729e+06
       ...        ...         ...         ...         ...           ...
2020-02-16 255.021667  257.255238  252.679762  255.198452  2.329087e+06
2020-02-23 265.220833  267.263690  262.948512  265.321905  1.826094e+06
2020-03-01 236.720536  238.697500  234.208750  236.373988  2.198762e+06
2020-03-08 229.923571  231.284583  228.373810  229.817619  1.628910e+06
2020-03-15 176.937521  179.979487  172.936239  176.332821  4.259828e+06

142 rows × 5 columns
```

```
In [22]: df=df.resample('W').agg({'Close': 'mean', 'High': 'max', 'Low': 'min', 'Volume':
```

```
In [35]: df
```

```
Out[35]: <pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000021D4EED28B0>
```

```
In [40]: df.groupby('High').mean()
```

```
Out[40]:
              Open      Low    Close       Volume
     High
    82.00    81.29    81.11    81.99    323206.25
    82.13    81.92    81.80    81.89    528742.26
    82.14    81.99    81.64    81.92    228693.57
    82.32    82.32    81.00    81.29    395708.94
    82.47    82.02    81.82    82.32    227169.03
      ...      ...      ...      ...          ...
  1404.00  1395.00  1383.63  1392.24   9849624.19
  1418.33  1410.00  1365.88  1382.56  23061241.98
  1418.80  1382.56  1382.50  1418.61  15617541.24
  1418.96  1388.99  1388.99  1410.00  27959588.92
  1419.96  1418.61  1360.99  1395.00  11452917.63

16174 rows × 4 columns
```

```
In [30]: df
```

```
Out[30]:
                  Close    High     Low        Volume
        Date
  2017-07-02 268.202162  293.73  253.23  8.084631e+07
  2017-07-09 261.062083  285.00  231.25  2.246746e+08
  2017-07-16 195.698393  240.33  130.26  5.017750e+08
  2017-07-23 212.783750  249.40  153.25  7.221637e+08
  2017-07-30 203.309524  229.99  178.03  2.657305e+08
         ...        ...     ...     ...           ...
  2020-02-16 255.198452  290.00  216.31  3.912867e+08
  2020-02-23 265.321905  287.13  242.36  3.067838e+08
  2020-03-01 236.373988  278.13  209.26  3.693920e+08
  2020-03-08 229.817619  253.01  196.00  2.736569e+08
  2020-03-15 176.332821  208.65   90.00  4.983998e+08

142 rows × 4 columns
```

```
In [51]: #Aggregate columns
         df = pd.read_csv('D:/survey_results_public.csv', index_col='ResponseId')
         schema_df = pd.read_csv('D:/survey_results_schema.csv')
```

In [54]:

```
df
```

Out[54]:

Currency	CompTotal	CompFreq	LanguageHaveWorkedWith	
NaN	NaN	NaN	NaN	
CAD\tCanadian dollar	NaN	NaN	JavaScript;TypeScript	
GBP\tPound sterling	32000.0	Yearly	C#;C++;HTML/CSS;JavaScript;Python	C#;C+-
ILS\tIsraeli new shekel	60000.0	Monthly	C#;JavaScript;SQL;TypeScript	
USD\tUnited States dollar	NaN	NaN	C#;HTML/CSS;JavaScript;SQL;Swift;TypeScript	C#;Elixi
...	...	...	...	
USD\tUnited States dollar	60000.0	Yearly	Bash/Shell;Dart;JavaScript;PHP;Python;SQL;Type...	Bash/Shell;Go
USD\tUnited States dollar	107000.0	Yearly	Bash/Shell;HTML/CSS;JavaScript;Python;SQL	
USD\tUnited States dollar	NaN	NaN	HTML/CSS;JavaScript;PHP;Python;SQL	C#;HTMl
GBP\tPound sterling	58500.0	Yearly	C#;Delphi;VBA	

	Currency	CompTotal	CompFreq	LanguageHaveWorkedWith
	NaN	NaN	NaN	C#;JavaScript;Lua;PowerShell;SQL;TypeScript

```
In [46]: pd.set_option('display.max_columns', 85)
         pd.set_option('display.max_rows', 85)
```

```
In [49]: df.head()
```

Out[49]:

LearnCodeCoursesCert	YearsCode	YearsCodePro	DevType	OrgSize	PurchaseInfluence	Buy
NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	
NaN	14	5	Data scientist or machine learning specialist;...	20 to 99 employees	I have some influence	
NaN	20	17	Developer, full-stack	100 to 499 employees	I have some influence	Othe
NaN	8	3	Developer, front-end;Developer, full-stack;Dev...	20 to 99 employees	I have some influence	S c comr

```
In [56]: df['ConvertedCompYearly'].median()
```

```
Out[56]: 67845.0
```

In [57]:
```
df.median()
```

Out[57]:
```
CompTotal                       77500.0
VCHostingPersonal use               NaN
VCHostingProfessional use           NaN
WorkExp                             8.0
ConvertedCompYearly             67845.0
dtype: float64
```

In [58]:
```
df.describe()
```

Out[58]:

	CompTotal	VCHostingPersonal use	VCHostingProfessional use	WorkExp	ConvertedCompYear
count	3.842200e+04	0.0	0.0	36769.000000	3.807100e+0
mean	2.342434e+52	NaN	NaN	10.242378	1.707613e+0
std	4.591478e+54	NaN	NaN	8.706850	7.814132e+0
min	0.000000e+00	NaN	NaN	0.000000	1.000000e+0
25%	3.000000e+04	NaN	NaN	4.000000	3.583200e+0
50%	7.750000e+04	NaN	NaN	8.000000	6.784500e+0
75%	1.540000e+05	NaN	NaN	15.000000	1.200000e+0
max	9.000000e+56	NaN	NaN	50.000000	5.000000e+0

In [59]:
```
df['ConvertedCompYearly'].count()
```

Out[59]:
```
38071
```

In [60]:
```
df['Country'].value_counts()
```

Out[60]:
```
United States of America                                13543
India                                                    6639
Germany                                                  5395
United Kingdom of Great Britain and Northern Ireland     4190
Canada                                                   2490
                                                        ...  
Seychelles                                                  1
Brunei Darussalam                                           1
Solomon Islands                                             1
Monaco                                                      1
Burkina Faso                                                1
Name: Country, Length: 180, dtype: int64
```

In [61]:
```
country_grp = df.groupby(['Country'])
```

In [62]:
```
country_grp.get_group('India')
```

Out[62]:

ageHaveWorkedWith	LanguageWantToWorkWith	DatabaseHaveWorkedWith	
HP;Python;TypeScript	C;C#;C++;Elixir;Go;HTML/CSS;Java;JavaScript;Ko...	Cloud Firestore;MongoDB;Firebase Realtime Data...	
;Java;JavaScript;SQL	APL;Bash/Shell;Go;Python;TypeScript	MongoDB;MySQL	
C#	C++;JavaScript	MongoDB;MySQL	
cript;Kotlin;TypeScript	Groovy	Elasticsearch;PostgreSQL	
avaScript;Python;SQL	Bash/Shell;HTML/CSS;Java;JavaScript;Python;SQL	Oracle;PostgreSQL;Redis;SQLite	C
...	...	...	
Java;Python	Julia;Python	PostgreSQL;SQLite	
HTML/CSS;JavaScript	Go;HTML/CSS;Java;JavaScript;TypeScript	MongoDB;Firebase Realtime Database	
Script;Perl;PowerSh...	Bash/Shell;C#;HTML/CSS;JavaScript;Perl;PowerSh...	Microsoft SQL Server;MongoDB	
C;Python;SQL	C;C++	MySQL	
Go;Java;SQL	JavaScript;TypeScript	MongoDB;MySQL	

In [64]:

```
filt = df['Country'] == 'India'
df.loc[filt]['OfficeStackSyncHaveWorkedWith'].value_counts()
```

Out[64]:

```
Microsoft Teams                                                       680
Microsoft Teams;Zoom                                                  504
Slack;Zoom                                                            468
Zoom                                                                  419
Slack                                                                 373
                                                                     ... 
Wickr;Zoom                                                              1
Unify Circuit;Zoom                                                      1
Cisco Webex Teams;Mattermost;Microsoft Teams;Rocketchat;Slack;Zoom      1
Microsoft Teams;RingCentral;Slack;Zoom                                  1
Microsoft Teams;RingCentral;Symphony;Zoom                               1
Name: OfficeStackSyncHaveWorkedWith, Length: 144, dtype: int64
```

In [66]:

```
filt = df['Country'] == 'India'
df.loc[filt]['LanguageWantToWorkWith'].str.contains('Python').sum()
```

Out[66]:

```
3094
```

In [70]:

```
country_uses_python=country_grp['LanguageWantToWorkWith'].apply(lambda x: x.str.
```

In [68]:

```
country_respondents = df['Country'].value_counts()
country_respondents
```

Out[68]:

```
United States of America                                13543
India                                                    6639
Germany                                                  5395
United Kingdom of Great Britain and Northern Ireland     4190
Canada                                                   2490
                                                        ...  
Seychelles                                                  1
Brunei Darussalam                                           1
Solomon Islands                                             1
Monaco                                                      1
Burkina Faso                                                1
Name: Country, Length: 180, dtype: int64
```

In [71]:
```
python_df = pd.concat([country_respondents, country_uses_python], axis='columns',
python_df
```

Out[71]:

	Country	LanguageWantToWorkWith
United States of America	13543	5656
India	6639	3094
Germany	5395	2212
United Kingdom of Great Britain and Northern Ireland	4190	1594
Canada	2490	1000
...	...	...
Seychelles	1	0
Brunei Darussalam	1	1
Solomon Islands	1	1
Monaco	1	1
Burkina Faso	1	1

180 rows × 2 columns

In [72]:
```

python_df.rename(columns={'Country': 'NumRespondents', 'LanguageWantToWorkWith':
```

In [73]:
```
python_df
```

Out[73]:

	NumRespondents	NumKnowsPython
United States of America	13543	5656
India	6639	3094
Germany	5395	2212
United Kingdom of Great Britain and Northern Ireland	4190	1594
Canada	2490	1000
...	...	...
Seychelles	1	0
Brunei Darussalam	1	1
Solomon Islands	1	1
Monaco	1	1
Burkina Faso	1	1

180 rows × 2 columns

In [74]:

```

NumRespondents  NumKnowsPython
United States   20949   10083
India   9061    3105
Germany 5866    2451
United Kingdom  5737    2384
Canada  3395    1558
... ... ...
Dominica    1   1
Tonga   1   0
Sao Tome and Principe   1   1
Saint Kitts and Nevis   1   0
Brunei Darussalam   1   0
179 rows × 2 columns

python_df['PctKnowsPython'] = (python_df['NumKnowsPython']/python_df['NumResponde
python_df
```

```
  File "<ipython-input-74-b9056b2d7647>", line 1
    NumRespondents      NumKnowsPython
                        ^
SyntaxError: invalid syntax
```

In [75]:

```
python_df['PctKnowsPython'] = (python_df['NumKnowsPython']/python_df['NumResponde
python_df
```

Out[75]:

	NumRespondents	NumKnowsPython	PctKnowsPython
United States of America	13543	5656	41.763273
India	6639	3094	46.603404
Germany	5395	2212	41.000927
United Kingdom of Great Britain and Northern Ireland	4190	1594	38.042959
Canada	2490	1000	40.160643
...	...	...	...
Seychelles	1	0	0.000000
Brunei Darussalam	1	1	100.000000
Solomon Islands	1	1	100.000000
Monaco	1	1	100.000000
Burkina Faso	1	1	100.000000

180 rows × 3 columns

In [76]:

```
python_df.loc['Japan']
```

Out[76]:

```
NumRespondents    333.000000
NumKnowsPython    128.000000
PctKnowsPython     38.438438
Name: Japan, dtype: float64
```

In [78]:

```
#Sorting values
df.sort_values(by=['Country', 'ConvertedCompYearly'], ascending=[True, False], ir
```

In [80]:

```
df.head()
```

Out[80]:

ledge_3	Knowledge_4	Knowledge_5	Knowledge_6	Knowledge_7	Frequency_1	Frequency_2	Frequ
NaN	NaN	NaN	NaN	NaN	NaN	NaN	
NaN	NaN	NaN	NaN	NaN	NaN	NaN	
)isagree	Agree	Agree	Strongly agree	Neither agree nor disagree	1-2 times a week	1-2 times a week	1-2
er agree disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	Neither agree nor disagree	1-2 times a week	1-2 times a week	1-2
Strongly agree	Strongly agree	Strongly agree	Agree	Agree	Never	1-2 times a week	1-2

```
In [81]: df['ConvertedCompYearly'].nlargest(10)
```

```
Out[81]: ResponseId
         40305    50000000.0
         202      44790396.0
         62027    35000000.0
         70523    32500000.0
         61044    28853768.0
         18923    22500000.0
         62224    22500000.0
         66496    18000000.0
         1291     15000000.0
         24164    15000000.0
         Name: ConvertedCompYearly, dtype: float64
```

In [82]:
```python
df.nsmallest(10, 'ConvertedCompYearly')
```

Out[82]:

3	TimeSearching	TimeAnswering	Onboarding	ProfessionalTech	TrueFalse_1	TrueFalse_
a k	15-30 minutes a day	30-60 minutes a day	Very long	Innersource initiative;DevOps function;Microse...	Yes	Y
a k	60-120 minutes a day	Over 120 minutes a day	Just right	DevOps function;Microservices;Developer portal...	Yes	Y
a k	15-30 minutes a day	30-60 minutes a day	Somewhat short	DevOps function;Microservices;Developer portal...	Yes	Y
N	NaN	NaN	NaN	NaN	NaN	Na
r	30-60 minutes a day	15-30 minutes a day	Very short	None of these	Yes	Y
a k	15-30 minutes a day	15-30 minutes a day	Just right	Automated testing	Yes	N
N	NaN	NaN	NaN	NaN	NaN	Na
r	15-30 minutes a day	15-30 minutes a day	Just right	Innersource initiative;DevOps function;Microse...	Yes	Y
N	NaN	NaN	NaN	NaN	NaN	Na
r	30-60 minutes a day	Less than 15 minutes a day	Just right	Continuous integration (CI) and (more often) c...	No	N

In [83]:
```
#Add remove rows and columns
people = {
    'first': ['Corey', 'Jane', 'John'],
    'last': ['Schafer', 'Doe', 'Doe'],
    'email': ['CoreyMSchafer@gmail.com', 'JaneDoe@email.com', 'JohnDoe@email.com'
}
```

In [84]:
```
df = pd.DataFrame(people)
```

In [85]:
```
df
```

Out[85]:

	first	last	email
0	Corey	Schafer	CoreyMSchafer@gmail.com
1	Jane	Doe	JaneDoe@email.com
2	John	Doe	JohnDoe@email.com

In [86]:
```
df['first'] + ' ' + df['last']
```

Out[86]:
```
0    Corey Schafer
1         Jane Doe
2         John Doe
dtype: object
```

In [87]:
```
df['full_name'] = df['first'] + ' ' + df['last']
```

In [88]:
```
df
```

Out[88]:

	first	last	email	full_name
0	Corey	Schafer	CoreyMSchafer@gmail.com	Corey Schafer
1	Jane	Doe	JaneDoe@email.com	Jane Doe
2	John	Doe	JohnDoe@email.com	John Doe

In [89]:
```
df.drop(columns=['first', 'last'], inplace=True)
```

In [90]:
```
df
```

Out[90]:

	email	full_name
0	CoreyMSchafer@gmail.com	Corey Schafer
1	JaneDoe@email.com	Jane Doe
2	JohnDoe@email.com	John Doe

In [91]:

```
df['full_name'].str.split(' ', expand=True)
```

Out[91]:

	0	1
0	Corey	Schafer
1	Jane	Doe
2	John	Doe

In [92]:

```
df[['first', 'last']] = df['full_name'].str.split(' ', expand=True)
```

In [93]:

```
df
```

Out[93]:

	email	full_name	first	last
0	CoreyMSchafer@gmail.com	Corey Schafer	Corey	Schafer
1	JaneDoe@email.com	Jane Doe	Jane	Doe
2	JohnDoe@email.com	John Doe	John	Doe

In [94]:

```
df.append({'first': 'Tony'}, ignore_index=True)
```

Out[94]:

	email	full_name	first	last
0	CoreyMSchafer@gmail.com	Corey Schafer	Corey	Schafer
1	JaneDoe@email.com	Jane Doe	Jane	Doe
2	JohnDoe@email.com	John Doe	John	Doe
3	NaN	NaN	Tony	NaN

In [98]:

```
drop=df.drop(index=2)
```

In [99]:

```
drop
```

Out[99]:

	email	full_name	first	last
0	CoreyMSchafer@gmail.com	Corey Schafer	Corey	Schafer
1	JaneDoe@email.com	Jane Doe	Jane	Doe

In [100]:

```
df
```

Out[100]:

	email	full_name	first	last
0	CoreyMSchafer@gmail.com	Corey Schafer	Corey	Schafer
1	JaneDoe@email.com	Jane Doe	Jane	Doe
2	JohnDoe@email.com	John Doe	John	Doe

In [103]:

```python
filt=df['last']=='Doe'
df.drop(index=df[filt].index)
```

Out[103]:

	email	full_name	first	last
0	CoreyMSchafer@gmail.com	Corey Schafer	Corey	Schafer

In [106]:

```python
people = {
    'first': ['Tony', 'Steve'],
    'last': ['Stark', 'Rogers'],
    'email': ['IronMan@avenge.com', 'Cap@avenge.com']
}
df2 = pd.DataFrame(people)
df2
```

Out[106]:

	first	last	email
0	Tony	Stark	IronMan@avenge.com
1	Steve	Rogers	Cap@avenge.com

In [105]:

```python
df.append(df2, ignore_index=True, sort=False)
```

Out[105]:

	email	full_name	first	last
0	CoreyMSchafer@gmail.com	Corey Schafer	Corey	Schafer
1	JaneDoe@email.com	Jane Doe	Jane	Doe
2	JohnDoe@email.com	John Doe	John	Doe
3	IronMan@avenge.com	NaN	Tony	Stark
4	Cap@avenge.com	NaN	Steve	Rogers

In [109]:

```python
#Renaming columns
#Aggregate columns
df = pd.read_csv('D:/survey_results_public.csv', index_col='ResponseId')
schema_df = pd.read_csv('D:/survey_results_schema.csv')
```

In [110]:
```
df
```

Out[110]:

	MainBranch	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode	
ResponseId							
1	None of these	NaN	NaN	NaN	NaN	NaN	
2	I am a developer by profession	Employed, full-time	Fully remote	Hobby;Contribute to open-source projects	NaN	NaN	
3	I am not primarily a developer, but I write co...	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)	Books / Physical media;Friend or family member...	do
4	I am a developer by profession	Employed, full-time	Fully remote	I don’t code outside of work	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;School (i.e., Universit...	
5	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Other online resources (e.g., videos, blogs, f...	
...	...	...	...	...	...	...	
73264	I am a developer by profession	Employed, full-time	Fully remote	Freelance/contract work	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;Other online resources ...	
73265	I am a developer by profession	Employed, full-time	Full in-person	Hobby	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)	Other online resources (e.g., videos, blogs, f...	
73266	I am not primarily a developer, but I write co...	Employed, full-time	Hybrid (some remote, some in-person)	Hobby;School or academic work	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;Other online resources ...	
73267	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;On the job training	

	MainBranch	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode	
ResponseId							
73268	I used to be a developer by profession, but no...	Independent contractor, freelancer, or self-em...	Fully remote	Hobby;Contribute to open-source projects;Boots...	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;Friend or family member...	do

73268 rows × 78 columns

In [111]:

```
pd.set_option('display.max_columns', 85)
pd.set_option('display.max_rows', 85)
```

In [112]:

```
df.head()
```

Out[112]:

	MainBranch	Employment	RemoteWork	CodingActivities	EdLevel	LearnCode	
ResponseId							
1	None of these	NaN	NaN	NaN	NaN	NaN	
2	I am a developer by profession	Employed, full-time	Fully remote	Hobby;Contribute to open-source projects	NaN	NaN	
3	I am not primarily a developer, but I write co...	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Master’s degree (M.A., M.S., M.Eng., MBA, etc.)	Books / Physical media;Friend or family member...	doc
4	I am a developer by profession	Employed, full-time	Fully remote	I don’t code outside of work	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Books / Physical media;School (i.e., Universit...	
5	I am a developer by profession	Employed, full-time	Hybrid (some remote, some in-person)	Hobby	Bachelor’s degree (B.A., B.S., B.Eng., etc.)	Other online resources (e.g., videos, blogs, f...	

In [113]:

```
df.rename(columns={'ConvertedCompYearly': 'SalaryUSD'}, inplace=True)
```

In [116]:

```python
df['SalaryUSD']
```

Out[116]:

```
ResponseId
1             NaN
2             NaN
3         40205.0
4        215232.0
5             NaN
           ...   
73264         NaN
73265         NaN
73266         NaN
73267         NaN
73268         NaN
Name: SalaryUSD, Length: 73268, dtype: float64
```

In [ ]: