Le Ngoc Thai Phuong

18521272

I. Pandas Introduction

In [1]:
```
%matplotlib inline
import numpy as np
import pandas as pd
```

In [2]:
```
df=pd.read_csv('PastHires.csv')
df.head()
```

Out[2]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N

In [3]:
```
df.head(10)
```

Out[3]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N
5	0	N	0	PhD	Y	Y	Y
6	5	Y	2	MS	N	Y	Y
7	3	N	1	BS	N	Y	Y
8	15	Y	5	BS	N	N	Y
9	0	N	0	BS	N	N	N

In [4]:
```
1 df.tail(4)
```

Out[4]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
9	0	N	0	BS	N	N	N
10	1	N	1	PhD	Y	N	N
11	4	Y	1	BS	N	Y	Y
12	0	N	0	PhD	Y	N	Y

In [5]:
```
1 df.shape
```

Out[5]:
```
(13, 7)
```

In [6]:
```
1 df.size
```

Out[6]:
```
91
```

In [7]:
```
1 len(df)
```

Out[7]:
```
13
```

In [8]:
```
1 df.columns
```

Out[8]:
```
Index(['Years Experience', 'Employed?', 'Previous employers',
       'Level of Education', 'Top-tier school', 'Interned', 'Hired'],
      dtype='object')
```

In [9]:
```
1 df["Hired"]
```

Out[9]:
```
0     Y
1     Y
2     N
3     Y
4     N
5     Y
6     Y
7     Y
8     Y
9     N
10    N
11    Y
12    Y
Name: Hired, dtype: object
```

In [10]:
```
1 df["Hired"][:5]
```

Out[10]:
```
0    Y
1    Y
2    N
3    Y
4    N
Name: Hired, dtype: object
```

In [11]:
```
1 df[['Years Experience','Hired']]
```

Out[11]:

	Years Experience	Hired
0	10	Y
1	0	Y
2	7	N
3	2	Y
4	20	N
5	0	Y
6	5	Y
7	3	Y
8	15	Y
9	0	N
10	1	N
11	4	Y
12	0	Y

In [12]:
```
1 df[['Years Experience','Hired']][:5]
```

Out[12]:

	Years Experience	Hired
0	10	Y
1	0	Y
2	7	N
3	2	Y
4	20	N

In [13]:
```
1 df.sort_values(['Years Experience','Hired'])
```

Out[13]:

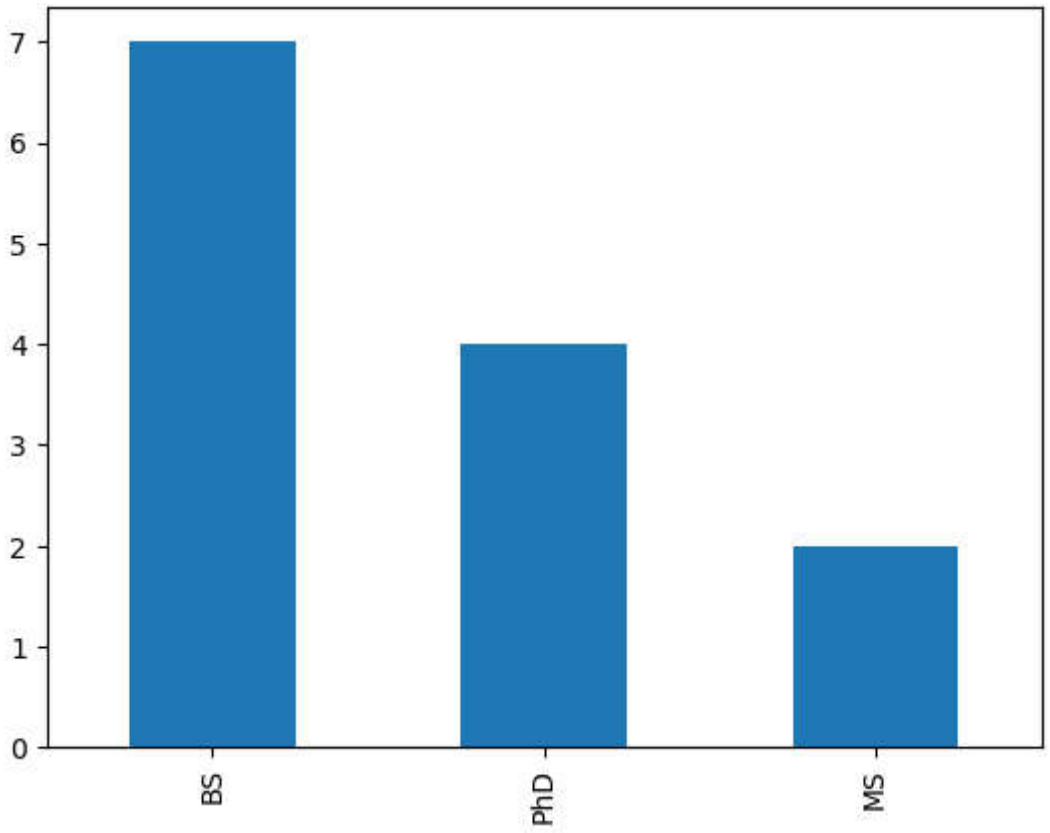
	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
9	0	N	0	BS	N	N	N
1	0	N	0	BS	Y	Y	Y
5	0	N	0	PhD	Y	Y	Y
12	0	N	0	PhD	Y	N	Y
10	1	N	1	PhD	Y	N	N
3	2	Y	1	MS	Y	N	Y
7	3	N	1	BS	N	Y	Y
11	4	Y	1	BS	N	Y	Y
6	5	Y	2	MS	N	Y	Y
2	7	N	6	BS	N	N	N
0	10	Y	4	BS	N	N	Y
8	15	Y	5	BS	N	N	Y
4	20	N	2	PhD	Y	N	N

In [14]:
```
1 degree_counts=df['Level of Education'].value_counts()
2 degree_counts
```

Out[14]:
```
BS     7
PhD    4
MS     2
Name: Level of Education, dtype: int64
```

In [15]:
```
1 degree_counts.plot(kind='bar')
```

Out[15]: <AxesSubplot:>



II. Series

In [16]:
```
1 import numpy as  np
2 import pandas as pd
```

In [17]:
```
1 labels=['a','b','c']
2 my_list=[ 10,20,30]
3 arr=np.array([ 10,20,30])
4 d={'a':10,'b':20,'c':30}
```

In [18]:
```
1 pd.Series(data=my_list)
```

Out[18]:
```
0    10
1    20
2    30
dtype: int64
```

In [19]:
```
1 pd.Series(data=my_list,index=labels)
```

Out[19]:
```
a    10
b    20
c    30
dtype: int64
```

In [20]:
```
1 pd.Series(my_list,labels)
```

Out[20]:
```
a    10
b    20
c    30
dtype: int64
```

In [21]:
```
1 pd.Series(arr)
```

Out[21]:
```
0    10
1    20
2    30
dtype: int32
```

In [22]:
```
1 pd.Series(arr,labels)
```

Out[22]:
```
a    10
b    20
c    30
dtype: int32
```

In [23]:
```
1 pd.Series(d)
```

Out[23]:
```
a    10
b    20
c    30
dtype: int64
```

In [24]:
```
1 pd.Series(data=labels)
```

Out[24]:
```
0    a
1    b
2    c
dtype: object
```

In [25]:
```
1 pd.Series([sum,print,len])
```

Out[25]:
```
0      <built-in function sum>
1    <built-in function print>
2      <built-in function len>
dtype: object
```

In [26]:
```
1 ser1=pd.Series([1,2,3,4],index=['USA',"Germany","USSR","Japan"])
```

In [27]:
```
1 ser1
```

Out[27]:
```
USA        1
Germany    2
USSR       3
Japan      4
dtype: int64
```

In [28]:
```
1 ser2=pd.Series([1,2,5,4],index=['USA','Germany','Italy','Japan'])
```

In [29]:
```
1 ser2
```

Out[29]:
```
USA        1
Germany    2
Italy      5
Japan      4
dtype: int64
```

In [30]:
```
1 ser1['USA']
```

Out[30]:
```
1
```

In [31]:
```
1 ser1+ser2
```

Out[31]:
```
Germany    4.0
Italy      NaN
Japan      8.0
USA        2.0
USSR       NaN
dtype: float64
```

III. DataFrames

In [32]:
```
1 import pandas as pd
2 import numpy as np
```

In [33]:
```
1 from numpy.random import randn
2 np.random.seed(101)
```

In [34]:
```
1 df=pd.DataFrame(randn(5,4),index='A B C D E'.split(),columns="W X Y Z".sp
```

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

Out[35]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [36]:
```
1 df['W']
```

Out[36]:
```
A    2.706850
B    0.651118
C   -2.018168
D    0.188695
E    0.190794
Name: W, dtype: float64
```

In [37]:
```
1 df[['W','Z']]
```

Out[37]:

	W	Z
A	2.706850	0.503826
B	0.651118	0.605965
C	-2.018168	-0.589001
D	0.188695	0.955057
E	0.190794	0.683509

In [38]:
```
1 df.W
```

Out[38]:
```
A    2.706850
B    0.651118
C   -2.018168
D    0.188695
E    0.190794
Name: W, dtype: float64
```

In [39]:
```
1 type(df['W'])
```

Out[39]:
```
pandas.core.series.Series
```

In [40]:
```
1 df['new']=df['W']+df['Y']
```

In [41]:
```
1 df
```

Out[41]:

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

In [42]:
```
1 df.drop("new",axis=1)
```

Out[42]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [43]:
```
1 df
```

Out[43]:

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

In [44]:
```
1 df.drop("new",axis=1,inplace=True)
```

In [45]:
```
1 df
```

Out[45]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [46]:
```
1 df.drop('E',axis=0)
```

Out[46]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057

In [47]:
```
1 df
```

Out[47]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [48]:
```
1 df.drop("E",axis=0)
```

Out[48]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057

In [49]:
```
1 df.loc['A']
```

Out[49]:
```
W    2.706850
X    0.628133
Y    0.907969
Z    0.503826
Name: A, dtype: float64
```

In [50]:
```
1 df.iloc[2]
```

Out[50]:
```
W   -2.018168
X    0.740122
Y    0.528813
Z   -0.589001
Name: C, dtype: float64
```

In [51]:
```
1 df.loc['B','Y']
```

Out[51]:
```
-0.8480769834036315
```

In [52]:
```
1 df.loc[['A','B'],['W','Y']]
```

Out[52]:

	W	Y
A	2.706850	0.907969
B	0.651118	-0.848077

In [53]:
```
1 df
```

Out[53]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [54]:
```
1 df>0
```

Out[54]:

	W	X	Y	Z
A	True	True	True	True
B	True	False	False	True
C	False	True	True	False
D	True	False	False	True
E	True	True	True	True

In [55]:
```
1 df[df>0]
```

Out[55]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	NaN	NaN	0.605965
C	NaN	0.740122	0.528813	NaN
D	0.188695	NaN	NaN	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [56]:
```
1 df[df['W']>0]
```

Out[56]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [57]:
```
1 df[df['W']>0]['Y']
```

Out[57]:
```
A    0.907969
B   -0.848077
D   -0.933237
E    2.605967
Name: Y, dtype: float64
```

In [58]:
```
1 df[df['W']>0][['Y','X']]
```

Out[58]:

	Y	X
A	0.907969	0.628133
B	-0.848077	-0.319318
D	-0.933237	-0.758872
E	2.605967	1.978757

In [59]:
```
1 (df[(df['W']>0)& (df['Y']>1)])
```

Out[59]:

	W	X	Y	Z
E	0.190794	1.978757	2.605967	0.683509

In [60]:
```
1 df
```

Out[60]:

	W	X	Y	Z
A	2.706850	0.628133	0.907969	0.503826
B	0.651118	-0.319318	-0.848077	0.605965
C	-2.018168	0.740122	0.528813	-0.589001
D	0.188695	-0.758872	-0.933237	0.955057
E	0.190794	1.978757	2.605967	0.683509

In [61]:
```
1 df.reset_index()
```

Out[61]:

	index	W	X	Y	Z
0	A	2.706850	0.628133	0.907969	0.503826
1	B	0.651118	-0.319318	-0.848077	0.605965
2	C	-2.018168	0.740122	0.528813	-0.589001
3	D	0.188695	-0.758872	-0.933237	0.955057
4	E	0.190794	1.978757	2.605967	0.683509

In [62]:
```
1 newin="CA NY WY OR CO".split()
```

In [63]:
```
1 df['States']=newin
```

In [64]:
```
1 df
```

Out[64]:

	W	X	Y	Z	States
A	2.706850	0.628133	0.907969	0.503826	CA
B	0.651118	-0.319318	-0.848077	0.605965	NY
C	-2.018168	0.740122	0.528813	-0.589001	WY
D	0.188695	-0.758872	-0.933237	0.955057	OR
E	0.190794	1.978757	2.605967	0.683509	CO

In [65]:
```
1 df.set_index('States',inplace=True)
```

In [66]:
```
1 df
```

Out[66]:

	W	X	Y	Z
States				
CA	2.706850	0.628133	0.907969	0.503826
NY	0.651118	-0.319318	-0.848077	0.605965
WY	-2.018168	0.740122	0.528813	-0.589001
OR	0.188695	-0.758872	-0.933237	0.955057
CO	0.190794	1.978757	2.605967	0.683509

In [67]:
```
1 outside=['G1','G1','G1','G2','G2','G2']
2 inside=[1,2,3,1,2,3]
3 hier_index=list(zip(outside,inside))
4 hier_index=pd.MultiIndex.from_tuples(hier_index)
```

In [68]:
```
1 hier_index
```

Out[68]:
```
MultiIndex([('G1', 1),
            ('G1', 2),
            ('G1', 3),
            ('G2', 1),
            ('G2', 2),
            ('G2', 3)],
           )
```

In [69]:
```
df=pd.DataFrame(np.random.rand(6,2),index=hier_index,columns=['A','B'])
```

In [70]:
```
df
```

Out[70]:

		A	B
G1	**1**	0.734819	0.541962
	2	0.913154	0.807920
	3	0.402998	0.357224
G2	**1**	0.952877	0.343632
	2	0.865100	0.830278
	3	0.538161	0.922469

In [71]:
```
df.loc['G1']
```

Out[71]:

	A	B
1	0.734819	0.541962
2	0.913154	0.807920
3	0.402998	0.357224

In [72]:
```
df.loc['G1'].loc[1]
```

Out[72]:
```
A    0.734819
B    0.541962
Name: 1, dtype: float64
```

In [73]:
```
df.index.names
```

Out[73]:
```
FrozenList([None, None])
```

In [74]:
```
df.index.names=['Group','Num']
```

In [75]:
```
1 df
```

Out[75]:

Group	Num	A	B
G1	1	0.734819	0.541962
	2	0.913154	0.807920
	3	0.402998	0.357224
G2	1	0.952877	0.343632
	2	0.865100	0.830278
	3	0.538161	0.922469

In [76]:
```
1 df.xs(['G1',1])
```

```
C:\Users\ACER\AppData\Local\Temp\ipykernel_14360\580597333.py:1: FutureWarnin
g: Passing lists as key for xs is deprecated and will be removed in a future
version. Pass key as a tuple instead.
  df.xs(['G1',1])
```

Out[76]:
```
A    0.734819
B    0.541962
Name: (G1, 1), dtype: float64
```

In [77]:
```
1 df.xs(1,level='Num')
```

Out[77]:

Group	A	B
G1	0.734819	0.541962
G2	0.952877	0.343632

IV. Missing Data

In [78]:
```
1 import numpy  as np
2 import pandas as pd
```

In [79]:
```
df=pd.DataFrame({
  'A':[1,2,np.nan],
  'B':[5,np.nan,np.nan],
  'C':[1,2,3]
})
df
```

Out[79]:

	A	B	C
0	1.0	5.0	1
1	2.0	NaN	2
2	NaN	NaN	3

In [80]:
```
df.dropna()
```

Out[80]:

	A	B	C
0	1.0	5.0	1

In [81]:
```
df.dropna(axis=1)
```

Out[81]:

	C
0	1
1	2
2	3

In [82]:
```
df.dropna(thresh=2)
```

Out[82]:

	A	B	C
0	1.0	5.0	1
1	2.0	NaN	2

In [83]:
```
df.fillna(value="FILL VALUE")
```

Out[83]:

	A	B	C
0	1.0	5.0	1
1	2.0	FILL VALUE	2
2	FILL VALUE	FILL VALUE	3

In [84]:
```
df['A'].fillna(value=df['A'].mean())
```

Out[84]:
```
0    1.0
1    2.0
2    1.5
Name: A, dtype: float64
```

In [85]:
```
data={'Company':["GOOG","GOOG","MSFT","MSFT",'FB','FB'],
      'Person':['Sam','Charlie','Amy','Vanessa','Car','Sarah'],
      'Sales':[200,120,340,124,234,350]
}
```

In [86]:
```
df=pd.DataFrame(data)
df
```

Out[86]:

	Company	Person	Sales
0	GOOG	Sam	200
1	GOOG	Charlie	120
2	MSFT	Amy	340
3	MSFT	Vanessa	124
4	FB	Car	234
5	FB	Sarah	350

In [87]:
```
df.groupby('Company')
```

Out[87]:
```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x000002204AB18070>
```

In [88]:
```
by_com=df.groupby('Company')
```

In [89]:
```
by_com.mean()
```

Out[89]:

	Sales
Company	
FB	292.0
GOOG	160.0
MSFT	232.0

In [90]:
```
1 df.groupby('Company').mean()
```

Out[90]:

Company	Sales
FB	292.0
GOOG	160.0
MSFT	232.0

In [91]:
```
1 by_com.std()
```

Out[91]:

Company	Sales
FB	82.024387
GOOG	56.568542
MSFT	152.735065

In [92]:
```
1 by_com.min()
```

Out[92]:

Company	Person	Sales
FB	Car	234
GOOG	Charlie	120
MSFT	Amy	124

In [93]:
```
1 by_com.max()
```

Out[93]:

Company	Person	Sales
FB	Sarah	350
GOOG	Sam	200
MSFT	Vanessa	340

In [94]:
```
1 by_com.count()
```

Out[94]:

Company	Person	Sales
FB	2	2
GOOG	2	2
MSFT	2	2

In [95]:
```
1 by_com.describe()
```

Out[95]:

	Sales							
	count	mean	std	min	25%	50%	75%	max
Company								
FB	2.0	292.0	82.024387	234.0	263.0	292.0	321.0	350.0
GOOG	2.0	160.0	56.568542	120.0	140.0	160.0	180.0	200.0
MSFT	2.0	232.0	152.735065	124.0	178.0	232.0	286.0	340.0

In [96]:
```
1 by_com.describe().transpose()
```

Out[96]:

	Company	FB	GOOG	MSFT
Sales	count	2.000000	2.000000	2.000000
	mean	292.000000	160.000000	232.000000
	std	82.024387	56.568542	152.735065
	min	234.000000	120.000000	124.000000
	25%	263.000000	140.000000	178.000000
	50%	292.000000	160.000000	232.000000
	75%	321.000000	180.000000	286.000000
	max	350.000000	200.000000	340.000000

In [97]:
```
1 by_com.describe().transpose()['GOOG']
```

Out[97]:
```
Sales  count      2.000000
       mean     160.000000
       std       56.568542
       min      120.000000
       25%      140.000000
       50%      160.000000
       75%      180.000000
       max      200.000000
Name: GOOG, dtype: float64
```

In [ ]:
```
1
```

V. Merging, Joining and Concatenating

In [98]:
```
1 import pandas as pd
```

In [99]:
```
1 df1 = pd.DataFrame({'A': ['A0','A1','A2','A3'],
2                     'B': ['B0','B1','B2','B3'],
3                     'C': ['C0','C1','C2','C3'],
4                     'D': ['D0','D1','D2','D3']},
5                    index=[0,1,2,3])
6
```

In [100]:
```
1 df2 = pd.DataFrame({'A': ['A4','A5','A6','A7'],
2                     'B': ['B4','B5','B6','B7'],
3                     'C': ['C4','C5','C6','C7'],
4                     'D': ['D4','D5','D6','D7']},
5                    index=[4,5,6,7])
```

In [101]:
```
1 df3 = pd.DataFrame({'A': ['A8','A9','A10','A11'],
2                     'B': ['B8','B9','B10','B11'],
3                     'C': ['C8','C9','C10','C11'],
4                     'D': ['D8','D9','D10','D11']},
5                    index=[8,9,10,11])
```

In [102]:
```
1 df1
```

Out[102]:

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3

In [103]:
```
1 df2
```

Out[103]:

	A	B	C	D
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7

In [104]:
```
1 df3
```

Out[104]:

	A	B	C	D
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

In [105]:
```
1 pd.concat([df1,df2,df3])
```

Out[105]:

	A	B	C	D
0	A0	B0	C0	D0
1	A1	B1	C1	D1
2	A2	B2	C2	D2
3	A3	B3	C3	D3
4	A4	B4	C4	D4
5	A5	B5	C5	D5
6	A6	B6	C6	D6
7	A7	B7	C7	D7
8	A8	B8	C8	D8
9	A9	B9	C9	D9
10	A10	B10	C10	D10
11	A11	B11	C11	D11

In [106]:
```
pd.concat([df1,df2,df3], axis=1)
```

Out[106]:

	A	B	C	D	A	B	C	D	A	B	C	D
0	A0	B0	C0	D0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	A1	B1	C1	D1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	A2	B2	C2	D2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	A3	B3	C3	D3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	A4	B4	C4	D4	NaN	NaN	NaN	NaN
5	NaN	NaN	NaN	NaN	A5	B5	C5	D5	NaN	NaN	NaN	NaN
6	NaN	NaN	NaN	NaN	A6	B6	C6	D6	NaN	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN	A7	B7	C7	D7	NaN	NaN	NaN	NaN
8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A8	B8	C8	D8
9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A9	B9	C9	D9
10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A10	B10	C10	D10
11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	A11	B11	C11	D11

In [107]:
```
left = pd.DataFrame({
                     'A': ['A0','A1','A2','A3'],
                     'B': ['B0','B1','B2','B3'],
                    'key': ['K0','K1','K2','K3']})
right = pd.DataFrame({
                     'C': ['C0','C1','C2','C3'],
                     'D': ['D0','D1','D2','D3'],
                    'key': ['K0','K1','K2','K3']})

```

In [108]:
```
left
```

Out[108]:

	A	B	key
0	A0	B0	K0
1	A1	B1	K1
2	A2	B2	K2
3	A3	B3	K3

In [109]:
```
1 right
```

Out[109]:

	C	D	key
0	C0	D0	K0
1	C1	D1	K1
2	C2	D2	K2
3	C3	D3	K3

In [110]:
```
1 pd.merge(left,right, how='inner',on='key')
```

Out[110]:

	A	B	key	C	D
0	A0	B0	K0	C0	D0
1	A1	B1	K1	C1	D1
2	A2	B2	K2	C2	D2
3	A3	B3	K3	C3	D3

In [111]:
```
1  #-------------------------bug
2  left = pd.DataFrame({
3                       'A': ['A0','A1','A2'],
4                       'B': ['B0','B1','B2'],
5                      'key1': ['K0','K0','K1'],
6                      'key2': ['K0','K1','K0']})
7  right = pd.DataFrame({
8                       'C': ['C0','C1','C2'],
9                       'D': ['D0','D1','D2'],
10                     'key2': ['K0','K1','K0'],
11                     'key1': ['K0','K0','K1']
12                     })
```

In [112]:
```
1 pd.merge(left,right, on=['key1','key2'])
```

Out[112]:

	A	B	key1	key2	C	D
0	A0	B0	K0	K0	C0	D0
1	A1	B1	K0	K1	C1	D1
2	A2	B2	K1	K0	C2	D2

In [113]:
```
pd.merge(left,right, how='outer', on=['key1','key2'])
```

Out[113]:

	A	B	key1	key2	C	D
0	A0	B0	K0	K0	C0	D0
1	A1	B1	K0	K1	C1	D1
2	A2	B2	K1	K0	C2	D2

In [114]:
```
pd.merge(left,right, how='right', on=['key1','key2'])
```

Out[114]:

	A	B	key1	key2	C	D
0	A0	B0	K0	K0	C0	D0
1	A1	B1	K0	K1	C1	D1
2	A2	B2	K1	K0	C2	D2

In [115]:
```
pd.merge(left,right, how='left', on=['key1','key2'])
```

Out[115]:

	A	B	key1	key2	C	D
0	A0	B0	K0	K0	C0	D0
1	A1	B1	K0	K1	C1	D1
2	A2	B2	K1	K0	C2	D2

In [116]:
```
left = pd.DataFrame({
                     'A': ['A0','A1','A2'],
                     'B': ['B0','B1','B2']},
                     index=['K0','K1','K2'])
right = pd.DataFrame({
                     'C': ['C0','C2','C3'],
                     'D': ['D0','D2','D3']},
                     index=['K0','K2','K3'] )
```

In [117]:
```
left.join(right)
```

Out[117]:

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2

In [118]:
```
1 left.join(right,how='outer')
```

Out[118]:

	A	B	C	D
K0	A0	B0	C0	D0
K1	A1	B1	NaN	NaN
K2	A2	B2	C2	D2
K3	NaN	NaN	C3	D3

VI. Operation

In [119]:
```
1 import numpy as np
2 import pandas as pd
```

In [120]:
```
1 df=pd.DataFrame({
2   'col1':[1,2,3,4],
3   'col2':[444,555,666,444],
4   'col3':['abc','def','ghi','xyz']
5 })
```

In [121]:
```
1 df.head()
```

Out[121]:

	col1	col2	col3
0	1	444	abc
1	2	555	def
2	3	666	ghi
3	4	444	xyz

In [122]:
```
1 df['col2'].unique()
```

Out[122]:
```
array([444, 555, 666], dtype=int64)
```

In [123]:
```
1 df['col2'].nunique()
```

Out[123]:
```
3
```

In [124]:
```
1 df['col2'].value_counts()
```

Out[124]:
```
444    2
555    1
666    1
Name: col2, dtype: int64
```

In [125]:
```
newdf = df[(df['col1']>2)&(df['col2']==444)]
newdf
```

Out[125]:

	col1	col2	col3
3	4	444	xyz

In [126]:
```
def times(x):
  return x*2
```

In [127]:
```
df['col1'].apply(times)
```

Out[127]:
```
0    2
1    4
2    6
3    8
Name: col1, dtype: int64
```

In [128]:
```
df['col3'].apply(len)
```

Out[128]:
```
0    3
1    3
2    3
3    3
Name: col3, dtype: int64
```

In [129]:
```
df['col1'].sum()
```

Out[129]:
```
10
```

In [130]:
```
del df['col1']
```

In [131]:
```
df
```

Out[131]:

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

In [132]:
```
df.columns
```

Out[132]:
```
Index(['col2', 'col3'], dtype='object')
```

In [133]:
```
df.index
```

Out[133]:
```
RangeIndex(start=0, stop=4, step=1)
```

In [134]:
```
1 df
```

Out[134]:

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

In [135]:
```
1 df.sort_values(by='col2')
```

Out[135]:

	col2	col3
0	444	abc
3	444	xyz
1	555	def
2	666	ghi

In [136]:
```
1 df.isnull()
```

Out[136]:

	col2	col3
0	False	False
1	False	False
2	False	False
3	False	False

In [137]:
```
1 df.dropna()
```

Out[137]:

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

In [138]:
```
1 df.head()
```

Out[138]:

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

In [139]:
```
1 df.isnull()
```

Out[139]:

	col2	col3
0	False	False
1	False	False
2	False	False
3	False	False

In [140]:
```
1 df.fillna('FILL')
```

Out[140]:

	col2	col3
0	444	abc
1	555	def
2	666	ghi
3	444	xyz

In [141]:
```
1 data={
2   'A':['foo','foo','foo','bar','bar','bar'],
3   'B':['one','one','two','two','one','one'],
4   'C':['x','y','x','y','x','y'],
5   'D':[1,3,2,5,4,1]
6 }
7 df=pd.DataFrame(data)
```

In [142]:
```
df
```

Out[142]:

	A	B	C	D
0	foo	one	x	1
1	foo	one	y	3
2	foo	two	x	2
3	bar	two	y	5
4	bar	one	x	4
5	bar	one	y	1

In [143]:
```
df.pivot_table(values='D',index=['A','B'],columns=['C'])
```

Out[143]:

		C	x	y
A	**B**		
bar	**one**	4.0	1.0
	two	NaN	5.0
foo	**one**	1.0	3.0
	two	2.0	NaN

In [ ]:
```

```

VII. Data Input and Output

In [144]:
```
%matplotlib inline
import numpy as np
import pandas as pd
```

In [145]:
```
df = pd.read_csv("PastHires.csv")
```

In [146]:
```
df.head()
```

Out[146]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N

In [147]:
```
1 df.head(10)
```

Out[147]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
0	10	Y	4	BS	N	N	Y
1	0	N	0	BS	Y	Y	Y
2	7	N	6	BS	N	N	N
3	2	Y	1	MS	Y	N	Y
4	20	N	2	PhD	Y	N	N
5	0	N	0	PhD	Y	Y	Y
6	5	Y	2	MS	N	Y	Y
7	3	N	1	BS	N	Y	Y
8	15	Y	5	BS	N	N	Y
9	0	N	0	BS	N	N	N

In [148]:
```
1 df.tail(4)
```

Out[148]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
9	0	N	0	BS	N	N	N
10	1	N	1	PhD	Y	N	N
11	4	Y	1	BS	N	Y	Y
12	0	N	0	PhD	Y	N	Y

In [149]:
```
1 df.shape
```

Out[149]:
```
(13, 7)
```

In [150]:
```
1 df.size
```

Out[150]:
```
91
```

In [151]:
```
1 len(df)
```

Out[151]:
```
13
```

In [152]:
```
1 df.columns
```

Out[152]:
```
Index(['Years Experience', 'Employed?', 'Previous employers',
       'Level of Education', 'Top-tier school', 'Interned', 'Hired'],
      dtype='object')
```

In [153]:
```
1 df['Hired']
```

Out[153]:
```
0     Y
1     Y
2     N
3     Y
4     N
5     Y
6     Y
7     Y
8     Y
9     N
10    N
11    Y
12    Y
Name: Hired, dtype: object
```

In [154]:
```
1 df['Hired'][:5]
```

Out[154]:
```
0    Y
1    Y
2    N
3    Y
4    N
Name: Hired, dtype: object
```

In [155]:
```
1 df['Hired'][5]
```

Out[155]:
```
'Y'
```

In [156]:
```
1 df[['Years Experience', 'Hired']]
```

Out[156]:

	Years Experience	Hired
0	10	Y
1	0	Y
2	7	N
3	2	Y
4	20	N
5	0	Y
6	5	Y
7	3	Y
8	15	Y
9	0	N
10	1	N
11	4	Y
12	0	Y

In [157]:
```
1 df[['Years Experience','Hired']][:5]
```

Out[157]:

	Years Experience	Hired
0	10	Y
1	0	Y
2	7	N
3	2	Y
4	20	N

In [158]:
```
1 df.sort_values(['Years Experience'])
```

Out[158]:

	Years Experience	Employed?	Previous employers	Level of Education	Top-tier school	Interned	Hired
1	0	N	0	BS	Y	Y	Y
5	0	N	0	PhD	Y	Y	Y
9	0	N	0	BS	N	N	N
12	0	N	0	PhD	Y	N	Y
10	1	N	1	PhD	Y	N	N
3	2	Y	1	MS	Y	N	Y
7	3	N	1	BS	N	Y	Y
11	4	Y	1	BS	N	Y	Y
6	5	Y	2	MS	N	Y	Y
2	7	N	6	BS	N	N	N
0	10	Y	4	BS	N	N	Y
8	15	Y	5	BS	N	N	Y
4	20	N	2	PhD	Y	N	N

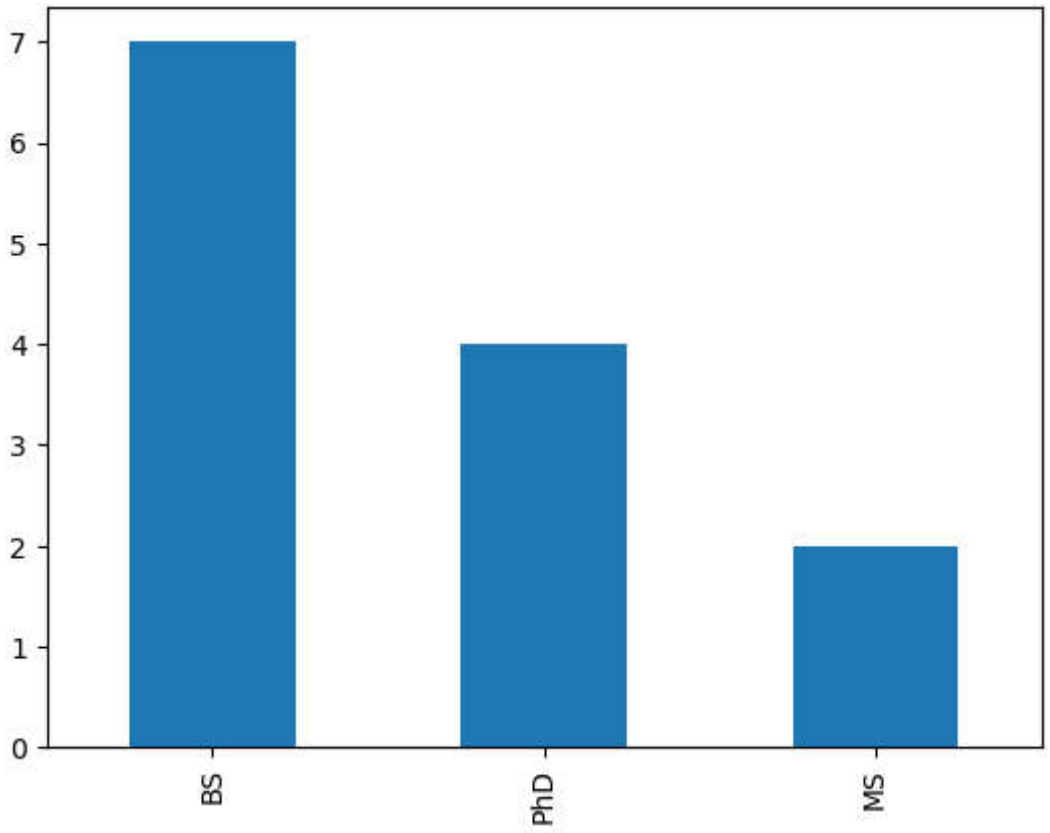
In [159]:
```
1 degree_counts = df['Level of Education'].value_counts()
```

In [160]:
```
1 degree_counts
```

Out[160]:
```
BS     7
PhD    4
MS     2
Name: Level of Education, dtype: int64
```

In [161]:
```
1 degree_counts.plot(kind='bar')
```

Out[161]: <AxesSubplot:>



In [162]:
```
1 df = pd.read_csv('example.csv')
```

In [163]:
```
1 df
```

Out[163]:

	a	b	c	d
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

In [164]:
```
1 df.to_csv('example1.csv', index = False)
2
```

In [165]:
```
1 pd.read_excel('Excel_Sample.xlsx', sheet_name='Sheet1')
```

Out[165]:

	a	b	c	d
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
3	12	13	14	15

In [166]:
```
1 from sqlalchemy import create_engine
```

In [167]:
```
1 engine = create_engine ('sqlite:///:memory:')
```

In [168]:
```
1 df.to_sql('data',engine)
```

Out[168]: 4

In [169]:
```
1 sql_df = pd.read_sql('data',con=engine)
```

In [170]:
```
1 sql_df
```

Out[170]:

	index	a	b	c	d
0	0	0	1	2	3
1	1	4	5	6	7
2	2	8	9	10	11
3	3	12	13	14	15

VIII. Exersises

In [171]:
```
1 import numpy as np
2 import pandas as pd
```

In [172]:
```
1 df1= pd.read_csv('sales.csv')
```

In [173]:
```
1 df1.head()
```

Out[173]:

	order_id	name	ordered_at	price	quantity	line_total
0	10000	"ICE CREAM" Peanut Fudge	2018-01-01 11:30:00	$3.50	3	$10.50
1	10000	"ICE CREAM" Peanut Fudge	2018-01-01 11:30:00	$3.50	1	$3.50
2	10001	"SORBET" Raspberry	2018-01-01 12:14:54	$2.50	2	$5.00
3	10001	NaN	2018-01-01 12:14:54	$1.50	1	$1.50
4	10001	"CONE" Dipped Waffle Cone	2018-01-01 12:14:54	$3.50	1	$3.50

In [174]:
```
1 df1.dtypes
```

Out[174]:
```
order_id       int64
name          object
ordered_at    object
price         object
quantity       int64
line_total    object
dtype: object
```

In [ ]:
```
1
```

In [ ]:
```
1
```

In [ ]:
```
1
```