week3-project1-1

January 2, 2024

[1]:
```
import pandas as pd
import numpy as np
```

[2]:
```
df = pd.read_csv("C:
↪\\Users\\Dipen\\Downloads\\Data-cleaning-for-beginners-using-pandas(1).csv")
```

[3]:
```
df
```

[3]:

	Index	Age	Salary	Rating	Location	Established	Easy Apply
0	0	44.0	$44k-$99k	5.4	India,In	1999	TRUE
1	1	66.0	$55k-$66k	3.5	New York,Ny	2002	TRUE
2	2	NaN	$77k-$89k	-1.0	New York,Ny	-1	-1
3	3	64.0	$44k-$99k	4.4	India In	1988	-1
4	4	25.0	$44k-$99k	6.4	Australia Aus	2002	-1
5	5	44.0	$77k-$89k	1.4	India,In	1999	TRUE
6	6	21.0	$44k-$99k	0.0	New York,Ny	-1	-1
7	7	44.0	$44k-$99k	-1.0	Australia Aus	-1	-1
8	8	35.0	$44k-$99k	5.4	New York,Ny	-1	-1
9	9	22.0	$44k-$99k	7.7	India,In	-1	TRUE
10	10	55.0	$10k-$49k	5.4	India,In	2008	TRUE
11	11	44.0	$10k-$49k	6.7	India,In	2009	-1
12	12	NaN	$44k-$99k	0.0	India,In	1999	-1
13	13	25.0	$44k-$99k	-1.0	Australia Aus	2019	TRUE
14	14	66.0	$44k-$99k	4.0	Australia Aus	2020	TRUE
15	15	44.0	$88k-$101k	3.0	Australia Aus	1999	-1
16	16	19.0	$19k-$40k	4.5	India,In	1984	-1
17	17	NaN	$44k-$99k	5.3	New York,Ny	1943	TRUE
18	18	35.0	$44k-$99k	6.7	New York,Ny	1954	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	1955	TRUE
20	20	NaN	$44k-$99k	5.7	New York,Ny	1944	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	1946	-1
22	22	19.0	$55k-$66k	7.8	New York,Ny	1988	TRUE
23	23	NaN	$44k-$99k	2.4	New York,Ny	1999	TRUE
24	24	13.0	$44k-$99k	-1.0	New York,Ny	1987	-1
25	25	55.0	$44k-$99k	0.0	Australia Aus	1980	TRUE
26	26	NaN	$55k-$66k	NaN	India,In	1934	TRUE
27	27	52.0	$44k-$99k	5.4	India,In	1935	-1

```
28     28   NaN   $39k-$88k     3.4  Australia Aus         1932         -1
```

[4]:
```
df.head() # gives first 5 values
```

[4]:
```
   Index   Age     Salary  Rating       Location  Established Easy Apply
0      0  44.0  $44k-$99k     5.4       India,In         1999       TRUE
1      1  66.0  $55k-$66k     3.5    New York,Ny         2002       TRUE
2      2   NaN  $77k-$89k    -1.0    New York,Ny           -1         -1
3      3  64.0  $44k-$99k     4.4       India In         1988         -1
4      4  25.0  $44k-$99k     6.4  Australia Aus         2002         -1
```

[5]:
```
df.tail()#give last 5 values
```

[5]:
```
    Index   Age     Salary  Rating       Location  Established Easy Apply
24     24  13.0  $44k-$99k    -1.0    New York,Ny         1987         -1
25     25  55.0  $44k-$99k     0.0  Australia Aus         1980       TRUE
26     26   NaN  $55k-$66k     NaN       India,In         1934       TRUE
27     27  52.0  $44k-$99k     5.4       India,In         1935         -1
28     28   NaN  $39k-$88k     3.4  Australia Aus         1932         -1
```

[6]:
```
df.shape
```

[6]:
```
(29, 7)
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29 entries, 0 to 28
Data columns (total 7 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Index        29 non-null     int64
 1   Age          22 non-null     float64
 2   Salary       29 non-null     object
 3   Rating       28 non-null     float64
 4   Location     29 non-null     object
 5   Established  29 non-null     int64
 6   Easy Apply   29 non-null     object
dtypes: float64(2), int64(2), object(3)
memory usage: 1.7+ KB
```

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

[8]:
```
Index(['Index', 'Age', 'Salary', 'Rating', 'Location', 'Established',
       'Easy Apply'],
      dtype='object')
```

[9]:
```
df.isnull().sum() #give null summary
```

[9]:
```
Index          0
Age            7
Salary         0
Rating         1
Location       0
Established    0
Easy Apply     0
dtype: int64
```

[ ]:

[10]:
```
df
```

[10]:

	Index	Age	Salary	Rating	Location	Established	Easy Apply
0	0	44.0	$44k-$99k	5.4	India,In	1999	TRUE
1	1	66.0	$55k-$66k	3.5	New York,Ny	2002	TRUE
2	2	NaN	$77k-$89k	-1.0	New York,Ny	-1	-1
3	3	64.0	$44k-$99k	4.4	India In	1988	-1
4	4	25.0	$44k-$99k	6.4	Australia Aus	2002	-1
5	5	44.0	$77k-$89k	1.4	India,In	1999	TRUE
6	6	21.0	$44k-$99k	0.0	New York,Ny	-1	-1
7	7	44.0	$44k-$99k	-1.0	Australia Aus	-1	-1
8	8	35.0	$44k-$99k	5.4	New York,Ny	-1	-1
9	9	22.0	$44k-$99k	7.7	India,In	-1	TRUE
10	10	55.0	$10k-$49k	5.4	India,In	2008	TRUE
11	11	44.0	$10k-$49k	6.7	India,In	2009	-1
12	12	NaN	$44k-$99k	0.0	India,In	1999	-1
13	13	25.0	$44k-$99k	-1.0	Australia Aus	2019	TRUE
14	14	66.0	$44k-$99k	4.0	Australia Aus	2020	TRUE
15	15	44.0	$88k-$101k	3.0	Australia Aus	1999	-1
16	16	19.0	$19k-$40k	4.5	India,In	1984	-1
17	17	NaN	$44k-$99k	5.3	New York,Ny	1943	TRUE
18	18	35.0	$44k-$99k	6.7	New York,Ny	1954	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	1955	TRUE
20	20	NaN	$44k-$99k	5.7	New York,Ny	1944	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	1946	-1
22	22	19.0	$55k-$66k	7.8	New York,Ny	1988	TRUE
23	23	NaN	$44k-$99k	2.4	New York,Ny	1999	TRUE
24	24	13.0	$44k-$99k	-1.0	New York,Ny	1987	-1
25	25	55.0	$44k-$99k	0.0	Australia Aus	1980	TRUE
26	26	NaN	$55k-$66k	NaN	India,In	1934	TRUE
27	27	52.0	$44k-$99k	5.4	India,In	1935	-1
28	28	NaN	$39k-$88k	3.4	Australia Aus	1932	-1

[11]:
```
text = '$40k-$100k'
text.replace('k',"000")
```

[11]:
```
'$40000-$100000'
```

[12]:
```
df.rename(columns={'Easy Apply': 'Easy_Apply'}, inplace=True)
#rename the Easy Apply column name
```

[13]:
```
df
```

[13]:

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	$44k-$99k	5.4	India,In	1999	TRUE
1	1	66.0	$55k-$66k	3.5	New York,Ny	2002	TRUE
2	2	NaN	$77k-$89k	-1.0	New York,Ny	-1	-1
3	3	64.0	$44k-$99k	4.4	India In	1988	-1
4	4	25.0	$44k-$99k	6.4	Australia Aus	2002	-1
5	5	44.0	$77k-$89k	1.4	India,In	1999	TRUE
6	6	21.0	$44k-$99k	0.0	New York,Ny	-1	-1
7	7	44.0	$44k-$99k	-1.0	Australia Aus	-1	-1
8	8	35.0	$44k-$99k	5.4	New York,Ny	-1	-1
9	9	22.0	$44k-$99k	7.7	India,In	-1	TRUE
10	10	55.0	$10k-$49k	5.4	India,In	2008	TRUE
11	11	44.0	$10k-$49k	6.7	India,In	2009	-1
12	12	NaN	$44k-$99k	0.0	India,In	1999	-1
13	13	25.0	$44k-$99k	-1.0	Australia Aus	2019	TRUE
14	14	66.0	$44k-$99k	4.0	Australia Aus	2020	TRUE
15	15	44.0	$88k-$101k	3.0	Australia Aus	1999	-1
16	16	19.0	$19k-$40k	4.5	India,In	1984	-1
17	17	NaN	$44k-$99k	5.3	New York,Ny	1943	TRUE
18	18	35.0	$44k-$99k	6.7	New York,Ny	1954	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	1955	TRUE
20	20	NaN	$44k-$99k	5.7	New York,Ny	1944	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	1946	-1
22	22	19.0	$55k-$66k	7.8	New York,Ny	1988	TRUE
23	23	NaN	$44k-$99k	2.4	New York,Ny	1999	TRUE
24	24	13.0	$44k-$99k	-1.0	New York,Ny	1987	-1
25	25	55.0	$44k-$99k	0.0	Australia Aus	1980	TRUE
26	26	NaN	$55k-$66k	NaN	India,In	1934	TRUE
27	27	52.0	$44k-$99k	5.4	India,In	1935	-1
28	28	NaN	$39k-$88k	3.4	Australia Aus	1932	-1

[ ]:
```
```

[14]:
```
df.Salary
```

[14]:
```
0      $44k-$99k
1      $55k-$66k
```

```
2      $77k-$89k
3      $44k-$99k
4      $44k-$99k
5      $77k-$89k
6      $44k-$99k
7      $44k-$99k
8      $44k-$99k
9      $44k-$99k
10     $10k-$49k
11     $10k-$49k
12     $44k-$99k
13     $44k-$99k
14     $44k-$99k
15    $88k-$101k
16     $19k-$40k
17     $44k-$99k
18     $44k-$99k
19     $44k-$99k
20     $44k-$99k
21     $44k-$99k
22     $55k-$66k
23     $44k-$99k
24     $44k-$99k
25     $44k-$99k
26     $55k-$66k
27     $44k-$99k
28     $39k-$88k
Name: Salary, dtype: object
```

[15]:
```
#1.Finding missing value
missing_values = df.isnull()
```

[16]:
```
print(missing_values)#true gives the missing value.
```

```
    Index    Age  Salary  Rating  Location  Established  Easy_Apply
0   False  False   False   False     False        False       False
1   False  False   False   False     False        False       False
2   False   True   False   False     False        False       False
3   False  False   False   False     False        False       False
4   False  False   False   False     False        False       False
5   False  False   False   False     False        False       False
6   False  False   False   False     False        False       False
7   False  False   False   False     False        False       False
8   False  False   False   False     False        False       False
9   False  False   False   False     False        False       False
10  False  False   False   False     False        False       False
11  False  False   False   False     False        False       False
```

```
12  False   True   False   False     False        False       False
13  False  False   False   False     False        False       False
14  False  False   False   False     False        False       False
15  False  False   False   False     False        False       False
16  False  False   False   False     False        False       False
17  False   True   False   False     False        False       False
18  False  False   False   False     False        False       False
19  False  False   False   False     False        False       False
20  False   True   False   False     False        False       False
21  False  False   False   False     False        False       False
22  False  False   False   False     False        False       False
23  False   True   False   False     False        False       False
24  False  False   False   False     False        False       False
25  False  False   False   False     False        False       False
26  False   True   False    True     False        False       False
27  False  False   False   False     False        False       False
28  False   True   False   False     False        False       False
```

[17]:
```
missing_values_summary = df.isnull().sum()
print(missing_values_summary)
```

```
Index          0
Age            7
Salary         0
Rating         1
Location       0
Established    0
Easy_Apply     0
dtype: int64
```

[18]:
```
#Handle missing values
df_cleaned = df.dropna()
print(df_cleaned)
```

```
    Index   Age      Salary  Rating       Location  Established Easy_Apply
0       0  44.0   $44k-$99k     5.4       India,In         1999       TRUE
1       1  66.0   $55k-$66k     3.5    New York,Ny         2002       TRUE
3       3  64.0   $44k-$99k     4.4       India In         1988         -1
4       4  25.0   $44k-$99k     6.4  Australia Aus         2002         -1
5       5  44.0   $77k-$89k     1.4       India,In         1999       TRUE
6       6  21.0   $44k-$99k     0.0    New York,Ny           -1         -1
7       7  44.0   $44k-$99k    -1.0  Australia Aus           -1         -1
8       8  35.0   $44k-$99k     5.4    New York,Ny           -1         -1
9       9  22.0   $44k-$99k     7.7       India,In           -1       TRUE
10     10  55.0   $10k-$49k     5.4       India,In         2008       TRUE
11     11  44.0   $10k-$49k     6.7       India,In         2009         -1
13     13  25.0   $44k-$99k    -1.0  Australia Aus         2019       TRUE
14     14  66.0   $44k-$99k     4.0  Australia Aus         2020       TRUE
```

15	15	44.0	$88k-$101k	3.0	Australia Aus	1999	-1
16	16	19.0	$19k-$40k	4.5	India,In	1984	-1
18	18	35.0	$44k-$99k	6.7	New York,Ny	1954	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	1955	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	1946	-1
22	22	19.0	$55k-$66k	7.8	New York,Ny	1988	TRUE
24	24	13.0	$44k-$99k	-1.0	New York,Ny	1987	-1
25	25	55.0	$44k-$99k	0.0	Australia Aus	1980	TRUE
27	27	52.0	$44k-$99k	5.4	India,In	1935	-1

[19]:
```
#replacing the negetive value with boolean value in Easy_Apply column
column_name = 'Easy_Apply'
old_value = '-1'
new_value = 'FALSE'

# Replace 'old_value' with 'new_value' in the specified column
df[column_name] = df[column_name].replace(old_value, new_value)
```

[20]:
```
df
```

[20]:

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	$44k-$99k	5.4	India,In	1999	TRUE
1	1	66.0	$55k-$66k	3.5	New York,Ny	2002	TRUE
2	2	NaN	$77k-$89k	-1.0	New York,Ny	-1	FALSE
3	3	64.0	$44k-$99k	4.4	India In	1988	FALSE
4	4	25.0	$44k-$99k	6.4	Australia Aus	2002	FALSE
5	5	44.0	$77k-$89k	1.4	India,In	1999	TRUE
6	6	21.0	$44k-$99k	0.0	New York,Ny	-1	FALSE
7	7	44.0	$44k-$99k	-1.0	Australia Aus	-1	FALSE
8	8	35.0	$44k-$99k	5.4	New York,Ny	-1	FALSE
9	9	22.0	$44k-$99k	7.7	India,In	-1	TRUE
10	10	55.0	$10k-$49k	5.4	India,In	2008	TRUE
11	11	44.0	$10k-$49k	6.7	India,In	2009	FALSE
12	12	NaN	$44k-$99k	0.0	India,In	1999	FALSE
13	13	25.0	$44k-$99k	-1.0	Australia Aus	2019	TRUE
14	14	66.0	$44k-$99k	4.0	Australia Aus	2020	TRUE
15	15	44.0	$88k-$101k	3.0	Australia Aus	1999	FALSE
16	16	19.0	$19k-$40k	4.5	India,In	1984	FALSE
17	17	NaN	$44k-$99k	5.3	New York,Ny	1943	TRUE
18	18	35.0	$44k-$99k	6.7	New York,Ny	1954	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	1955	TRUE
20	20	NaN	$44k-$99k	5.7	New York,Ny	1944	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	1946	FALSE
22	22	19.0	$55k-$66k	7.8	New York,Ny	1988	TRUE
23	23	NaN	$44k-$99k	2.4	New York,Ny	1999	TRUE
24	24	13.0	$44k-$99k	-1.0	New York,Ny	1987	FALSE
25	25	55.0	$44k-$99k	0.0	Australia Aus	1980	TRUE

```
26     26   NaN   $55k-$66k     NaN       India,In          1934       TRUE
27     27  52.0   $44k-$99k     5.4       India,In          1935      FALSE
28     28   NaN   $39k-$88k     3.4  Australia Aus          1932      FALSE
```

[21]:
```
#replacing the negetive value with a specific  Established Year in the␣
 ↪Established column
column_name = 'Established'
df[column_name] = df[column_name].apply(lambda x: max(x, 2000))
```

[22]:
```
df
```

[22]:

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	$44k-$99k	5.4	India,In	2000	TRUE
1	1	66.0	$55k-$66k	3.5	New York,Ny	2002	TRUE
2	2	NaN	$77k-$89k	-1.0	New York,Ny	2000	FALSE
3	3	64.0	$44k-$99k	4.4	India In	2000	FALSE
4	4	25.0	$44k-$99k	6.4	Australia Aus	2002	FALSE
5	5	44.0	$77k-$89k	1.4	India,In	2000	TRUE
6	6	21.0	$44k-$99k	0.0	New York,Ny	2000	FALSE
7	7	44.0	$44k-$99k	-1.0	Australia Aus	2000	FALSE
8	8	35.0	$44k-$99k	5.4	New York,Ny	2000	FALSE
9	9	22.0	$44k-$99k	7.7	India,In	2000	TRUE
10	10	55.0	$10k-$49k	5.4	India,In	2008	TRUE
11	11	44.0	$10k-$49k	6.7	India,In	2009	FALSE
12	12	NaN	$44k-$99k	0.0	India,In	2000	FALSE
13	13	25.0	$44k-$99k	-1.0	Australia Aus	2019	TRUE
14	14	66.0	$44k-$99k	4.0	Australia Aus	2020	TRUE
15	15	44.0	$88k-$101k	3.0	Australia Aus	2000	FALSE
16	16	19.0	$19k-$40k	4.5	India,In	2000	FALSE
17	17	NaN	$44k-$99k	5.3	New York,Ny	2000	TRUE
18	18	35.0	$44k-$99k	6.7	New York,Ny	2000	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	2000	TRUE
20	20	NaN	$44k-$99k	5.7	New York,Ny	2000	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	2000	FALSE
22	22	19.0	$55k-$66k	7.8	New York,Ny	2000	TRUE
23	23	NaN	$44k-$99k	2.4	New York,Ny	2000	TRUE
24	24	13.0	$44k-$99k	-1.0	New York,Ny	2000	FALSE
25	25	55.0	$44k-$99k	0.0	Australia Aus	2000	TRUE
26	26	NaN	$55k-$66k	NaN	India,In	2000	TRUE
27	27	52.0	$44k-$99k	5.4	India,In	2000	FALSE
28	28	NaN	$39k-$88k	3.4	Australia Aus	2000	FALSE

[23]:
```
#Replace negative values in 'Rating' 0.0
column_name = 'Rating'
df[column_name] = df[column_name].apply(lambda x: max(x, 0.0))
```

[24]:
```
df
```

[24]:

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	$44k-$99k	5.4	India,In	2000	TRUE
1	1	66.0	$55k-$66k	3.5	New York,Ny	2002	TRUE
2	2	NaN	$77k-$89k	0.0	New York,Ny	2000	FALSE
3	3	64.0	$44k-$99k	4.4	India In	2000	FALSE
4	4	25.0	$44k-$99k	6.4	Australia Aus	2002	FALSE
5	5	44.0	$77k-$89k	1.4	India,In	2000	TRUE
6	6	21.0	$44k-$99k	0.0	New York,Ny	2000	FALSE
7	7	44.0	$44k-$99k	0.0	Australia Aus	2000	FALSE
8	8	35.0	$44k-$99k	5.4	New York,Ny	2000	FALSE
9	9	22.0	$44k-$99k	7.7	India,In	2000	TRUE
10	10	55.0	$10k-$49k	5.4	India,In	2008	TRUE
11	11	44.0	$10k-$49k	6.7	India,In	2009	FALSE
12	12	NaN	$44k-$99k	0.0	India,In	2000	FALSE
13	13	25.0	$44k-$99k	0.0	Australia Aus	2019	TRUE
14	14	66.0	$44k-$99k	4.0	Australia Aus	2020	TRUE
15	15	44.0	$88k-$101k	3.0	Australia Aus	2000	FALSE
16	16	19.0	$19k-$40k	4.5	India,In	2000	FALSE
17	17	NaN	$44k-$99k	5.3	New York,Ny	2000	TRUE
18	18	35.0	$44k-$99k	6.7	New York,Ny	2000	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	2000	TRUE
20	20	NaN	$44k-$99k	5.7	New York,Ny	2000	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	2000	FALSE
22	22	19.0	$55k-$66k	7.8	New York,Ny	2000	TRUE
23	23	NaN	$44k-$99k	2.4	New York,Ny	2000	TRUE
24	24	13.0	$44k-$99k	0.0	New York,Ny	2000	FALSE
25	25	55.0	$44k-$99k	0.0	Australia Aus	2000	TRUE
26	26	NaN	$55k-$66k	NaN	India,In	2000	TRUE
27	27	52.0	$44k-$99k	5.4	India,In	2000	FALSE
28	28	NaN	$39k-$88k	3.4	Australia Aus	2000	FALSE

[25]:
```
#finding Data types
data_types = df.dtypes
print(data_types)
```

```
Index            int64
Age            float64
Salary          object
Rating         float64
Location        object
Established      int64
Easy_Apply      object
dtype: object
```

[26]:
```
df_cleaned = df.dropna()
print(df_cleaned) #drop the null values
```

```
    Index   Age      Salary  Rating       Location  Established Easy_Apply
```

0	0	44.0	$44k-$99k	5.4	India,In	2000	TRUE
1	1	66.0	$55k-$66k	3.5	New York,Ny	2002	TRUE
3	3	64.0	$44k-$99k	4.4	India In	2000	FALSE
4	4	25.0	$44k-$99k	6.4	Australia Aus	2002	FALSE
5	5	44.0	$77k-$89k	1.4	India,In	2000	TRUE
6	6	21.0	$44k-$99k	0.0	New York,Ny	2000	FALSE
7	7	44.0	$44k-$99k	0.0	Australia Aus	2000	FALSE
8	8	35.0	$44k-$99k	5.4	New York,Ny	2000	FALSE
9	9	22.0	$44k-$99k	7.7	India,In	2000	TRUE
10	10	55.0	$10k-$49k	5.4	India,In	2008	TRUE
11	11	44.0	$10k-$49k	6.7	India,In	2009	FALSE
13	13	25.0	$44k-$99k	0.0	Australia Aus	2019	TRUE
14	14	66.0	$44k-$99k	4.0	Australia Aus	2020	TRUE
15	15	44.0	$88k-$101k	3.0	Australia Aus	2000	FALSE
16	16	19.0	$19k-$40k	4.5	India,In	2000	FALSE
18	18	35.0	$44k-$99k	6.7	New York,Ny	2000	TRUE
19	19	32.0	$44k-$99k	3.3	New York,Ny	2000	TRUE
21	21	35.0	$44k-$99k	5.0	New York,Ny	2000	FALSE
22	22	19.0	$55k-$66k	7.8	New York,Ny	2000	TRUE
24	24	13.0	$44k-$99k	0.0	New York,Ny	2000	FALSE
25	25	55.0	$44k-$99k	0.0	Australia Aus	2000	TRUE
27	27	52.0	$44k-$99k	5.4	India,In	2000	FALSE

[ ]:

[ ]:

[ ]:

[27]:
```
#formating the Salary column
```

[28]:
```
# Display a sample of the Salary column
print(df['Salary'].head())
```

```
0    $44k-$99k
1    $55k-$66k
2    $77k-$89k
3    $44k-$99k
4    $44k-$99k
Name: Salary, dtype: object
```

[29]:
```
# Remove non-numeric characters and convert to numeric
df['Salary'] = pd.to_numeric(df['Salary'].str.replace('[^\d.]', ''),␣
 ↪errors='coerce')
```

```
C:\Users\Dipen\AppData\Local\Temp\ipykernel_12348\351522724.py:2: FutureWarning:
The default value of regex will change from True to False in a future version.
```

```
  df['Salary'] = pd.to_numeric(df['Salary'].str.replace('[^\d.]', ''),
errors='coerce')
```

[30]:
```
df
```

[30]:

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	4499	5.4	India,In	2000	TRUE
1	1	66.0	5566	3.5	New York,Ny	2002	TRUE
2	2	NaN	7789	0.0	New York,Ny	2000	FALSE
3	3	64.0	4499	4.4	India In	2000	FALSE
4	4	25.0	4499	6.4	Australia Aus	2002	FALSE
5	5	44.0	7789	1.4	India,In	2000	TRUE
6	6	21.0	4499	0.0	New York,Ny	2000	FALSE
7	7	44.0	4499	0.0	Australia Aus	2000	FALSE
8	8	35.0	4499	5.4	New York,Ny	2000	FALSE
9	9	22.0	4499	7.7	India,In	2000	TRUE
10	10	55.0	1049	5.4	India,In	2008	TRUE
11	11	44.0	1049	6.7	India,In	2009	FALSE
12	12	NaN	4499	0.0	India,In	2000	FALSE
13	13	25.0	4499	0.0	Australia Aus	2019	TRUE
14	14	66.0	4499	4.0	Australia Aus	2020	TRUE
15	15	44.0	88101	3.0	Australia Aus	2000	FALSE
16	16	19.0	1940	4.5	India,In	2000	FALSE
17	17	NaN	4499	5.3	New York,Ny	2000	TRUE
18	18	35.0	4499	6.7	New York,Ny	2000	TRUE
19	19	32.0	4499	3.3	New York,Ny	2000	TRUE
20	20	NaN	4499	5.7	New York,Ny	2000	TRUE
21	21	35.0	4499	5.0	New York,Ny	2000	FALSE
22	22	19.0	5566	7.8	New York,Ny	2000	TRUE
23	23	NaN	4499	2.4	New York,Ny	2000	TRUE
24	24	13.0	4499	0.0	New York,Ny	2000	FALSE
25	25	55.0	4499	0.0	Australia Aus	2000	TRUE
26	26	NaN	5566	NaN	India,In	2000	TRUE
27	27	52.0	4499	5.4	India,In	2000	FALSE
28	28	NaN	3988	3.4	Australia Aus	2000	FALSE

[31]:
```
# Check the data type of the Salary column
print(df['Salary'].dtype)
```

```
int64
```

[32]:
```
# Check for missing values in the Salary column
missing_values_salary = df['Salary'].isnull().sum()
print(missing_values_salary)
```

```
0
```

[33]:
```
#describe the Salary column
print(df['Salary'].describe())
```

```
count       29.000000
mean      7375.310345
std      15589.661178
min       1049.000000
25%       4499.000000
50%       4499.000000
75%       4499.000000
max      88101.000000
Name: Salary, dtype: float64
```

[34]:
```
#fomating the Location column
```

[35]:
```
# Display unique values in the Location column
print(df['Location'].unique())
```

```
['India,In' 'New York,Ny' 'India In' 'Australia Aus']
```

[36]:
```
df['Location'].value_counts()
```

[36]:
```
New York,Ny      12
India,In          9
Australia Aus     7
India In          1
Name: Location, dtype: int64
```

[37]:
```
df['Location'] = df['Location'].replace({'New York,Ny':'New York Ny',␣
↪'India,In':'India Ind','India In':'India Ind', 'Australia Aus':'Australia␣
↪Aus'})
```

[38]:
```
df['Location'].value_counts()
```

[38]:
```
New York Ny      12
India Ind        10
Australia Aus     7
Name: Location, dtype: int64
```

[39]:
```
print(df['Location'].unique())
```

```
['India Ind' 'New York Ny' 'Australia Aus']
```

[40]:
```
print(df['Location'].describe())
```

```
count              29
unique              3
top       New York Ny
```

```
freq                12
Name: Location, dtype: object
```

[41]:
```
#formation Established column
```

[42]:
```
# Describe the all values in the Established column
print(df['Established'].describe())
```

```
count      29.000000
mean     2002.068966
std         5.311248
min      2000.000000
25%      2000.000000
50%      2000.000000
75%      2000.000000
max      2020.000000
Name: Established, dtype: float64
```

[43]:
```
# Display unique values in the Established column
print(df['Established'].unique())
```

```
[2000 2002 2008 2009 2019 2020]
```

[44]:
```
# Display the minimum and maximum dates in the Established column
print("Minimum Date:", df['Established'].min())
print("Maximum Date:", df['Established'].max())
```

```
Minimum Date: 2000
Maximum Date: 2020
```

[45]:
```
#formatig the Easy_Apply column
```

[46]:
```
# Describe the all values in the Easy Apply column
print(df['Easy_Apply'].describe())
```

```
count       29
unique       2
top       TRUE
freq        15
Name: Easy_Apply, dtype: object
```

[47]:
```
# Display unique values in the Easy Apply column
print(df['Easy_Apply'].unique())
```

```
['TRUE' 'FALSE']
```

[48]:
```
# Check the data type of the Easy Apply column
print(df['Easy_Apply'].dtype)
```

```
object
```

[49]:
```
# Check unique values to identify boolean representations
print(df['Easy_Apply'].value_counts())
```

```
TRUE     15
FALSE    14
Name: Easy_Apply, dtype: int64
```

[ ]:
```

```

[50]:
```
print(df)
```

```
    Index   Age  Salary  Rating       Location  Established Easy_Apply
0       0  44.0    4499     5.4      India Ind         2000        TRUE
1       1  66.0    5566     3.5    New York Ny         2002        TRUE
2       2   NaN    7789     0.0    New York Ny         2000       FALSE
3       3  64.0    4499     4.4      India Ind         2000       FALSE
4       4  25.0    4499     6.4  Australia Aus         2002       FALSE
5       5  44.0    7789     1.4      India Ind         2000        TRUE
6       6  21.0    4499     0.0    New York Ny         2000       FALSE
7       7  44.0    4499     0.0  Australia Aus         2000       FALSE
8       8  35.0    4499     5.4    New York Ny         2000       FALSE
9       9  22.0    4499     7.7      India Ind         2000        TRUE
10     10  55.0    1049     5.4      India Ind         2008        TRUE
11     11  44.0    1049     6.7      India Ind         2009       FALSE
12     12   NaN    4499     0.0      India Ind         2000       FALSE
13     13  25.0    4499     0.0  Australia Aus         2019        TRUE
14     14  66.0    4499     4.0  Australia Aus         2020        TRUE
15     15  44.0   88101     3.0  Australia Aus         2000       FALSE
16     16  19.0    1940     4.5      India Ind         2000       FALSE
17     17   NaN    4499     5.3    New York Ny         2000        TRUE
18     18  35.0    4499     6.7    New York Ny         2000        TRUE
19     19  32.0    4499     3.3    New York Ny         2000        TRUE
20     20   NaN    4499     5.7    New York Ny         2000        TRUE
21     21  35.0    4499     5.0    New York Ny         2000       FALSE
22     22  19.0    5566     7.8    New York Ny         2000        TRUE
23     23   NaN    4499     2.4    New York Ny         2000        TRUE
24     24  13.0    4499     0.0    New York Ny         2000       FALSE
25     25  55.0    4499     0.0  Australia Aus         2000        TRUE
26     26   NaN    5566     NaN      India Ind         2000        TRUE
27     27  52.0    4499     5.4      India Ind         2000       FALSE
28     28   NaN    3988     3.4  Australia Aus         2000       FALSE
```

[51]:
```
#formating the Rating column
```

[52]:
```
# Describe the all values in the Rating column
print(df['Rating'].describe())
```

```
count    28.000000
mean      3.671429
std       2.601261
min       0.000000
25%       1.050000
50%       4.200000
75%       5.400000
max       7.800000
Name: Rating, dtype: float64
```

[53]:
```
# Display unique values in the Rating column
print(df['Rating'].unique())
```

```
[5.4 3.5 0.  4.4 6.4 1.4 7.7 6.7 4.  3.  4.5 5.3 3.3 5.7 5.  7.8 2.4 nan
 3.4]
```

[54]:
```
df_cleaned = df.dropna()
```

[55]:
```
print(df_cleaned)
```

```
    Index   Age  Salary  Rating       Location  Established Easy_Apply
0       0  44.0    4499     5.4      India Ind         2000       TRUE
1       1  66.0    5566     3.5    New York Ny         2002       TRUE
3       3  64.0    4499     4.4      India Ind         2000      FALSE
4       4  25.0    4499     6.4  Australia Aus         2002      FALSE
5       5  44.0    7789     1.4      India Ind         2000       TRUE
6       6  21.0    4499     0.0    New York Ny         2000      FALSE
7       7  44.0    4499     0.0  Australia Aus         2000      FALSE
8       8  35.0    4499     5.4    New York Ny         2000      FALSE
9       9  22.0    4499     7.7      India Ind         2000       TRUE
10     10  55.0    1049     5.4      India Ind         2008       TRUE
11     11  44.0    1049     6.7      India Ind         2009      FALSE
13     13  25.0    4499     0.0  Australia Aus         2019       TRUE
14     14  66.0    4499     4.0  Australia Aus         2020       TRUE
15     15  44.0   88101     3.0  Australia Aus         2000      FALSE
16     16  19.0    1940     4.5      India Ind         2000      FALSE
18     18  35.0    4499     6.7    New York Ny         2000       TRUE
19     19  32.0    4499     3.3    New York Ny         2000       TRUE
21     21  35.0    4499     5.0    New York Ny         2000      FALSE
22     22  19.0    5566     7.8    New York Ny         2000       TRUE
24     24  13.0    4499     0.0    New York Ny         2000      FALSE
25     25  55.0    4499     0.0  Australia Aus         2000       TRUE
27     27  52.0    4499     5.4      India Ind         2000      FALSE
```

[ ]:

[56]:
```
df
```

[56]:

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	4499	5.4	India Ind	2000	TRUE
1	1	66.0	5566	3.5	New York Ny	2002	TRUE
2	2	NaN	7789	0.0	New York Ny	2000	FALSE
3	3	64.0	4499	4.4	India Ind	2000	FALSE
4	4	25.0	4499	6.4	Australia Aus	2002	FALSE
5	5	44.0	7789	1.4	India Ind	2000	TRUE
6	6	21.0	4499	0.0	New York Ny	2000	FALSE
7	7	44.0	4499	0.0	Australia Aus	2000	FALSE
8	8	35.0	4499	5.4	New York Ny	2000	FALSE
9	9	22.0	4499	7.7	India Ind	2000	TRUE
10	10	55.0	1049	5.4	India Ind	2008	TRUE
11	11	44.0	1049	6.7	India Ind	2009	FALSE
12	12	NaN	4499	0.0	India Ind	2000	FALSE
13	13	25.0	4499	0.0	Australia Aus	2019	TRUE
14	14	66.0	4499	4.0	Australia Aus	2020	TRUE
15	15	44.0	88101	3.0	Australia Aus	2000	FALSE
16	16	19.0	1940	4.5	India Ind	2000	FALSE
17	17	NaN	4499	5.3	New York Ny	2000	TRUE
18	18	35.0	4499	6.7	New York Ny	2000	TRUE
19	19	32.0	4499	3.3	New York Ny	2000	TRUE
20	20	NaN	4499	5.7	New York Ny	2000	TRUE
21	21	35.0	4499	5.0	New York Ny	2000	FALSE
22	22	19.0	5566	7.8	New York Ny	2000	TRUE
23	23	NaN	4499	2.4	New York Ny	2000	TRUE
24	24	13.0	4499	0.0	New York Ny	2000	FALSE
25	25	55.0	4499	0.0	Australia Aus	2000	TRUE
26	26	NaN	5566	NaN	India Ind	2000	TRUE
27	27	52.0	4499	5.4	India Ind	2000	FALSE
28	28	NaN	3988	3.4	Australia Aus	2000	FALSE

[57]:
```
# Drop missing values permanently
df.dropna(subset=['Rating'], inplace=True)
df.dropna(subset=['Age'], inplace=True)
```

[58]:
```
df
```

[58]:

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	4499	5.4	India Ind	2000	TRUE
1	1	66.0	5566	3.5	New York Ny	2002	TRUE
3	3	64.0	4499	4.4	India Ind	2000	FALSE
4	4	25.0	4499	6.4	Australia Aus	2002	FALSE
5	5	44.0	7789	1.4	India Ind	2000	TRUE
6	6	21.0	4499	0.0	New York Ny	2000	FALSE
7	7	44.0	4499	0.0	Australia Aus	2000	FALSE
8	8	35.0	4499	5.4	New York Ny	2000	FALSE
9	9	22.0	4499	7.7	India Ind	2000	TRUE

```
10     10  55.0    1049     5.4      India Ind         2008       TRUE
11     11  44.0    1049     6.7      India Ind         2009      FALSE
13     13  25.0    4499     0.0  Australia Aus         2019       TRUE
14     14  66.0    4499     4.0  Australia Aus         2020       TRUE
15     15  44.0   88101     3.0  Australia Aus         2000      FALSE
16     16  19.0    1940     4.5      India Ind         2000      FALSE
18     18  35.0    4499     6.7    New York Ny         2000       TRUE
19     19  32.0    4499     3.3    New York Ny         2000       TRUE
21     21  35.0    4499     5.0    New York Ny         2000      FALSE
22     22  19.0    5566     7.8    New York Ny         2000       TRUE
24     24  13.0    4499     0.0    New York Ny         2000      FALSE
25     25  55.0    4499     0.0  Australia Aus         2000       TRUE
27     27  52.0    4499     5.4      India Ind         2000      FALSE
```

[59]:
```
#formating the Rating column
```

[60]:
```
# Describe the all values in the Rating column
print(df['Rating'].describe())
```

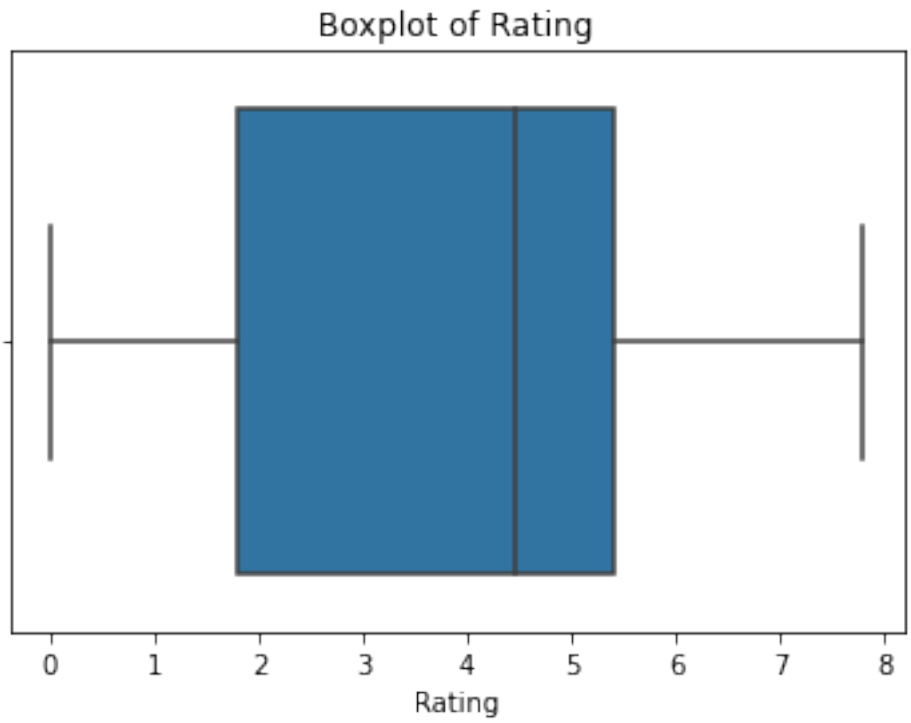
```
count    22.000000
mean      3.909091
std       2.637082
min       0.000000
25%       1.800000
50%       4.450000
75%       5.400000
max       7.800000
Name: Rating, dtype: float64
```

[ ]:
```

```

[61]:
```
import seaborn as sns
import matplotlib.pyplot as plt

sns.boxplot(x=df['Rating'])
plt.xlabel('Rating')
plt.title('Boxplot of Rating')
plt.show()
```



[62]:
```
#finding outliers in Rating column
```

[63]:
```
# Calculate the Interquartile Range (IQR) for the 'Salary' column
Q1 = df['Rating'].quantile(0.25)
Q3 = df['Rating'].quantile(0.75)
IQR = Q3 - Q1

# Define the upper and lower bounds for identifying outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Identify outliers based on the bounds
outliers_if = (df['Rating'] < lower_bound) | (df['Rating'] > upper_bound)

# Display the rows with outliers in the 'Salary' column
outliers_df = df[outliers_if]
```

[64]:
```
print(IQR)
```

```
3.6000000000000005
```

[65]:
```
print(lower_bound)
```

```
-3.6000000000000005
```

[66]:
```
print(upper_bound)
```

```
10.8
```

[67]:
```
print(outliers_df)#there is no outliers is Rating column
```

```
Empty DataFrame
Columns: [Index, Age, Salary, Rating, Location, Established, Easy_Apply]
Index: []
```

[68]:
```
# In Rating column there is no outliers.
```

[ ]:

[ ]:

[69]:
```
#formating Age column
```

[70]:
```
# Describe the all values in the Age column
print(df['Age'].describe())
```

```
count    22.000000
mean     39.045455
std      16.134781
min      13.000000
25%      25.000000
50%      39.500000
75%      50.000000
max      66.000000
Name: Age, dtype: float64
```

[ ]:

[71]:
```
df['Age'].value_counts()
```

[71]:
```
44.0    5
35.0    3
66.0    2
25.0    2
55.0    2
19.0    2
64.0    1
21.0    1
22.0    1
32.0    1
13.0    1
52.0    1
```

```
Name: Age, dtype: int64
```

[72]:
```
#Check for unusual entries in the Age column
unusual_entries = df[(df['Age'] < 0) | (df['Age'] > 120)]  # Assuming a␣
 ↪reasonable age range
print(unusual_entries)
```

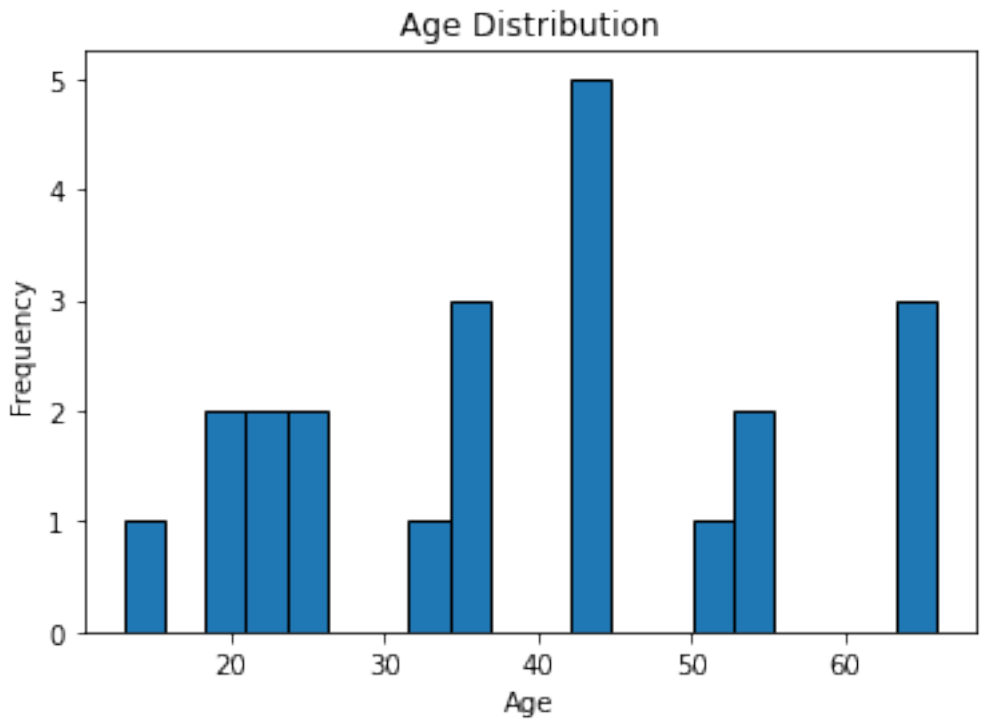
```
Empty DataFrame
Columns: [Index, Age, Salary, Rating, Location, Established, Easy_Apply]
Index: []
```

[73]:
```
#  also check for unexpected data types
unexpected_data_types = df[~df['Age'].apply(lambda x: isinstance(x, (int,␣
 ↪float)))]

# Display the rows with unexpected data types
print(unexpected_data_types)
```

```
Empty DataFrame
Columns: [Index, Age, Salary, Rating, Location, Established, Easy_Apply]
Index: []
```

[74]:
```
plt.hist(df['Age'], bins=20, edgecolor='black')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Age Distribution')
plt.show()
```



[75]:
```
# Check for inconsistencies between Established and Age columns
inconsistent_entries = df[df['Established'] != df['Age']]
print(inconsistent_entries)
```

	Index	Age	Salary	Rating	Location	Established	Easy_Apply
0	0	44.0	4499	5.4	India Ind	2000	TRUE
1	1	66.0	5566	3.5	New York Ny	2002	TRUE
3	3	64.0	4499	4.4	India Ind	2000	FALSE
4	4	25.0	4499	6.4	Australia Aus	2002	FALSE
5	5	44.0	7789	1.4	India Ind	2000	TRUE
6	6	21.0	4499	0.0	New York Ny	2000	FALSE
7	7	44.0	4499	0.0	Australia Aus	2000	FALSE
8	8	35.0	4499	5.4	New York Ny	2000	FALSE
9	9	22.0	4499	7.7	India Ind	2000	TRUE
10	10	55.0	1049	5.4	India Ind	2008	TRUE
11	11	44.0	1049	6.7	India Ind	2009	FALSE
13	13	25.0	4499	0.0	Australia Aus	2019	TRUE
14	14	66.0	4499	4.0	Australia Aus	2020	TRUE
15	15	44.0	88101	3.0	Australia Aus	2000	FALSE
16	16	19.0	1940	4.5	India Ind	2000	FALSE
18	18	35.0	4499	6.7	New York Ny	2000	TRUE
19	19	32.0	4499	3.3	New York Ny	2000	TRUE
21	21	35.0	4499	5.0	New York Ny	2000	FALSE
22	22	19.0	5566	7.8	New York Ny	2000	TRUE

```
24     24  13.0    4499     0.0    New York Ny         2000      FALSE
25     25  55.0    4499     0.0  Australia Aus         2000       TRUE
27     27  52.0    4499     5.4      India Ind         2000      FALSE
```

[76]:
```
#finding outliers in Salary column
```

[ ]:

[ ]:

[ ]:

[77]:
```
# Calculate the Interquartile Range (IQR) for the 'Salary' column
Q1 = df['Salary'].quantile(0.25)
Q3 = df['Salary'].quantile(0.75)
IQR = Q3 - Q1

# Define the upper and lower bounds for identifying outliers
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Identify outliers based on the bounds
outliers_ifs = (df['Salary'] < lower_bound) | (df['Salary'] > upper_bound)

# Display the rows with outliers in the 'Salary' column
outliers_dfs = df[outliers_ifs]
```

[78]:
```
print(lower_bound)
```

```
4499.0
```

[79]:
```
print(upper_bound)
```

```
4499.0
```

[80]:
```
print(IQR)
```

```
0.0
```

[81]:
```
print(outliers_dfs)
```

```
    Index   Age  Salary  Rating       Location  Established Easy_Apply
1       1  66.0    5566     3.5    New York Ny         2002       TRUE
5       5  44.0    7789     1.4      India Ind         2000       TRUE
10     10  55.0    1049     5.4      India Ind         2008       TRUE
11     11  44.0    1049     6.7      India Ind         2009      FALSE
15     15  44.0   88101     3.0  Australia Aus         2000      FALSE
```

```
16     16  19.0    1940     4.5       India Ind         2000      FALSE
22     22  19.0    5566     7.8    New York Ny          2000       TRUE
```

[ ]:

[ ]:

[ ]: