In [22]:
```
import pandas as pd
import numpy as np
```

In [23]:
```
features = ["Age", "Workclass", "fnlwgt", "Education", "Education-Num", "Martial Status", "Occupation", "Relationship",
            "Race", "Sex", "Capital Gain", "Capital Loss", "Hours per week", "Country", "Target"]

adult = pd.read_csv('adult.data', names=features)
adult
```

Out[23]:

	Age	Workclass	fnlwgt	Education	Education-Num	Martial Status	Occupation	Relationship	Race	Sex	Capital Gain	Capital Loss	Hours per week	Country	Target
0	39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	2174	0	40	United-States	<=50K
1	50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	13	United-States	<=50K
2	38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	0	0	40	United-States	<=50K
3	53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	0	0	40	United-States	<=50K
4	28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	0	0	40	Cuba	<=50K
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
32556	27	Private	257302	Assoc-acdm	12	Married-civ-spouse	Tech-support	Wife	White	Female	0	0	38	United-States	<=50K
32557	40	Private	154374	HS-grad	9	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	40	United-States	>50K
32558	58	Private	151910	HS-grad	9	Widowed	Adm-clerical	Unmarried	White	Female	0	0	40	United-States	<=50K
32559	22	Private	201490	HS-grad	9	Never-married	Adm-clerical	Own-child	White	Male	0	0	20	United-States	<=50K
32560	52	Self-emp-inc	287927	HS-grad	9	Married-civ-spouse	Exec-managerial	Wife	White	Female	15024	0	40	United-States	>50K

32561 rows × 15 columns

In [24]:
```
adult["Sex"].value_counts()
```

Out[24]:
```
 Male      21790
 Female    10771
Name: Sex, dtype: int64
```

In [25]:
```
adult[["Sex", "Age"]].groupby("Sex").mean()
```

Out[25]:

Sex	Age
Female	36.858230
Male	39.433547

In [26]:
```
gen_fem = adult[adult["Sex"].str.contains("Female")]
gen_fem["Age"].mean()
```

Out[26]:
```
36.85823043357163
```

In [27]:
```
percent_germany = adult[adult['Country'].str.contains('Germany')]
propo_germany = (len(percent_germany)*100)/len(adult)
print("Proportion of German Citizens:", propo_germany)
```

```
Proportion of German Citizens: 0.42074874850281013
```

In [28]:
```
age_more50k = adult[adult['Target'].str.contains('>50K')]
print("Mean value of Age who is having Target >50K:", age_more50k.Age.mean().round(2))
print("Std value of Age who is having Target >50K:", age_more50k.Age.std())
```

```
Mean value of Age who is having Target >50K: 44.25
Std value of Age who is having Target >50K: 10.51902771985177
```

In [29]:
```
age_less50k = adult[adult['Target'].str.contains('<=50K')]
print("Mean value of Age who is having Target <=50K:", age_less50k.Age.mean().round(2))
print("Std value of Age who is having Target <=50K:", age_less50k.Age.std())
```

```
Mean value of Age who is having Target <=50K: 36.78
Std value of Age who is having Target <=50K: 14.020088490824813
```

In [30]:
```
adult[adult['Target'].str.contains('>50K')]['Education'].unique()
```

Out[30]:
```
array([' HS-grad', ' Masters', ' Bachelors', ' Some-college',
       ' Assoc-voc', ' Doctorate', ' Prof-school', ' Assoc-acdm',
       ' 7th-8th', ' 12th', ' 10th', ' 11th', ' 9th', ' 5th-6th',
       ' 1st-4th'], dtype=object)
```

In [ ]:
```

```