In [15]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('attainment.csv')
data
```

Out[15]:

	Year	Sex	Min degree	Total	White	Black	Hispanic	Asian	Pacific Islander	American Indian/Alaska Native	Two or more races
0	1920	A	high school	?	22.0	6.3	?	?	?	?	?
1	1940	A	high school	38.1	41.2	12.3	?	?	?	?	?
2	1950	A	high school	52.8	56.3	23.6	?	?	?	?	?
3	1960	A	high school	60.7	63.7	38.6	?	?	?	?	?
4	1970	A	high school	75.4	77.8	58.4	?	?	?	?	?
...	...	...	...	...	...	...	...	...	...	...	...
209	2014	F	master's	9.3	11.1	5.0	3.6	20.8	?	?	7.5
210	2015	F	master's	10.4	12.0	7.2	4.1	23.2	?	?	10.2
211	2016	F	master's	11.2	12.3	6.3	6.3	28.8	?	?	8.2
212	2017	F	master's	10.5	11.8	6.8	5	25.8	?	?	5.4
213	2018	F	master's	10.7	12.6	6.2	3.8	29.9	?	?	?

214 rows × 11 columns

In [4]:
```
missing_rows = sum(df.isnull().any(axis=1))
print("Total number of rows with missing values: ",missing_rows)
```

```
Total number of rows with missing values:  0
```

In [6]:
```
print("Header:")
print(df.columns.tolist())
```

```
Header:
['Year', 'Sex', 'Min degree', 'Total', 'White', 'Black', 'Hispanic', 'Asian',
'Pacific Islander', 'American Indian/Alaska Native', 'Two or more races']
```

In [7]:
```
total_missing_values = (df == '?').sum().sum()
print("Total missing values",total_missing_values )
```

```
Total missing values 319
```

In [9]:
```
missing_values_male = (data[data['Sex'] == 'M'] == '?').sum()
missing_values_female = (data[data['Sex'] == 'F'] == '?').sum()
```

In [9]:
```
missing_values_male = (data[data['Sex'] == 'M'] == '?').sum()
missing_values_female = (data[data['Sex'] == 'F'] == '?').sum()
print("Number of '?' values for Male:")
print(missing_values_male)
print("\nNumber of '?' values for Female:")
print(missing_values_female)
```

```
Number of '?' values for Male:
Year                                0
Sex                                 0
Min degree                          0
Total                               0
White                               0
Black                               0
Hispanic                            0
Asian                              12
Pacific Islander                   39
American Indian/Alaska Native      29
Two or more races                  20
dtype: int64

Number of '?' values for Female:
Year                                0
Sex                                 0
Min degree                          0
Total                               0
White                               0
Black                               0
Hispanic                            0
Asian                              12
Pacific Islander                   44
American Indian/Alaska Native      25
Two or more races                  15
dtype: int64
```

In [11]:
```
missing_values_male = (data[data['Sex'] == 'M'] == '?').sum().sum()
missing_values_female = (data[data['Sex'] == 'F'] == '?').sum().sum()
print(f"Contains missing values for Men: {missing_values_male}")
print(f"Contains missing values for Women: {missing_values_female}")
```

```
Contains missing values for Men: 100
Contains missing values for Women: 96
```

In [12]:
```
null_values = (data == '?').sum()
print("Number of null values in individual columns:")
```

In [12]:
```
null_values = (data == '?').sum()
print("Number of null values in individual columns:")
print(null_values)
```

```
Number of null values in individual columns:
Year                                 0
Sex                                  0
Min degree                           0
Total                                2
White                                0
Black                                0
Hispanic                            10
Asian                               46
Pacific Islander                   121
American Indian/Alaska Native       83
Two or more races                   57
dtype: int64
```

In [13]:
```
missing_values_male = (data[data['Sex'] == 'M'] == '?').sum().sum()
missing_values_female = (data[data['Sex'] == 'F'] == '?').sum().sum()
result_df = pd.DataFrame({'Sex': ['Men', 'Women'], 'Total': [missing_values_male, missing_values_fema
print(result_df)
```

```
     Sex  Total
0    Men    100
1  Women     96
```

In [16]:
```
data = data.replace('?', np.nan)
print(data)
```

```
     Year Sex   Min degree Total  White  Black Hispanic Asian  \
0    1920   A  high school   NaN   22.0    6.3      NaN   NaN
1    1940   A  high school  38.1   41.2   12.3      NaN   NaN
2    1950   A  high school  52.8   56.3   23.6      NaN   NaN
3    1960   A  high school  60.7   63.7   38.6      NaN   NaN
4    1970   A  high school  75.4   77.8   58.4      NaN   NaN
..    ...  ..          ...   ...    ...    ...      ...   ...
209  2014   F     master's   9.3   11.1    5.0      3.6  20.8
210  2015   F     master's  10.4   12.0    7.2      4.1  23.2
211  2016   F     master's  11.2   12.3    6.3      6.3  28.8
212  2017   F     master's  10.5   11.8    6.8        5  25.8
213  2018   F     master's  10.7   12.6    6.2      3.8  29.9

    Pacific Islander American Indian/Alaska Native Two or more races
0                NaN                           NaN               NaN
1                NaN                           NaN               NaN
2                NaN                           NaN               NaN
3                NaN                           NaN               NaN
4                NaN                           NaN               NaN
..               ...                           ...               ...
209              NaN                           NaN               7.5
210              NaN                           NaN              10.2
211              NaN                           NaN               8.2
212              NaN                           NaN               5.4
213              NaN                           NaN               NaN

[214 rows x 11 columns]
```

In [17]:
```
numeric_columns = df.columns[3:]
df[numeric_columns] = df[numeric_columns].apply(pd.to_numeric, errors='coerce')
```

In [17]:
```
[214 rows x 11 columns]
numeric_columns = df.columns[3:]
df[numeric_columns] = df[numeric_columns].apply(pd.to_numeric, errors='coerce')
skewness = df.skew()
for column, skew in skewness.items():
    if -0.5 <= skew <= 0.5:
        df[column].fillna(df[column].mean(), inplace=True)
    elif -1 <= skew < -0.5 or 0.5 < skew <= 1:
        df[column].fillna(df[column].mean(), inplace=True)
    else:
        df[column].fillna(df[column].median(), inplace=True)
print(df.head())
```

```
   Year Sex   Min degree      Total  White  Black  Hispanic      Asian  \
0  1920   A  high school  42.771226   22.0    6.3    27.675  62.020238
1  1940   A  high school  38.100000   41.2   12.3    27.675  62.020238
2  1950   A  high school  52.800000   56.3   23.6    27.675  62.020238
3  1960   A  high school  60.700000   63.7   38.6    27.675  62.020238
4  1970   A  high school  75.400000   77.8   58.4    27.675  62.020238

   Pacific Islander  American Indian/Alaska Native  Two or more races
0         50.786022                      42.667176           44.457962
1         50.786022                      42.667176           44.457962
2         50.786022                      42.667176           44.457962
3         50.786022                      42.667176           44.457962
4         50.786022                      42.667176           44.457962

C:\Users\ADMIN\AppData\Local\Temp\ipykernel_12988\4116660737.py:12: FutureWarni
ng: The default value of numeric_only in DataFrame.skew is deprecated. In a fut
ure version, it will default to False. In addition, specifying 'numeric_only=No
ne' is deprecated. Select only valid columns or specify the value of numeric_on
ly to silence this warning.
  skewness = df.skew()
```

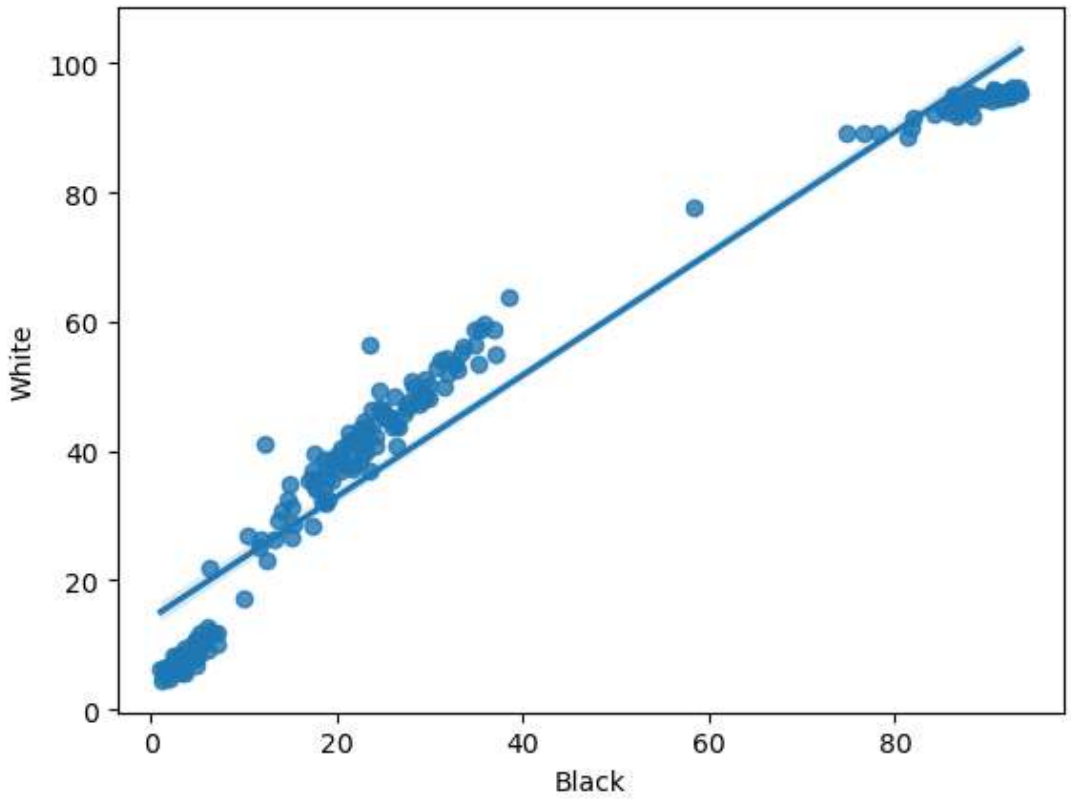
In [18]:
```
df_1980_hs = data[(data['Year'] == 1980) & (data['Min degree'] == 'high school')]
women_count = df_1980_hs[df_1980_hs['Sex'] == 'F']['Total'].values[0]
men_count = df_1980_hs[df_1980_hs['Sex'] == 'M']['Total'].values[0]

print(f"Number of women whose degree in high school in the year 1980: {women_count} ")
print(f"Number of men whose degree in high school in the year 1980: {men_count} ")
```

```
Number of women whose degree in high school in the year 1980: 85.5
Number of men whose degree in high school in the year 1980: 85.4
```

In [19]:
```
sns.regplot(x='Black', y='White', data=data)
plt.show()
```

In [19]:
```
sns.regplot(x='Black', y='White', data=data)
plt.show()
```



In [22]:
```
filtered_data = data[(data['Sex'] == 'A') & (data['Year'] >= 2000) & (data['Year'] <= 2010)]
race_means = filtered_data[['Black', 'White', 'Asian', 'Hispanic', 'American Indian/Alaska Native']].mean
```

In [22]:
```
filtered_data = data[(data['Sex'] == 'A') & (data['Year'] >= 2000) & (data['Year'] <= 2010)]
race_means = filtered_data[['Black', 'White', 'Asian', 'Hispanic', 'American Indian/Alaska Native']].mean
top_races = race_means.nlargest(2)
print(top_races)
Asian_white = filtered_data[['Asian', 'White']].sum()

bennett_email = "E22CSEU0119@bennett.edu.in"
part_1, part_2 = bennett_email.split('@')

print(f"Two most commonly awarded races: {top_races.index.values}")
print(f"Merged race values: {Asian_white.values}")
print(f"Part_1: {part_1}, Part_2: {part_2}")
```

```
White    45.821429
Black    34.367857
dtype: float64
Two most commonly awarded races: ['White' 'Black']
Merged race values: [1283.]
Part_1: E22CSEU0119, Part_2: bennett.edu.in

C:\Users\ADMIN\AppData\Local\Temp\ipykernel_12988\2149650727.py:2: FutureWarnin
g: The default value of numeric_only in DataFrame.mean is deprecated. In a futu
re version, it will default to False. In addition, specifying 'numeric_only=Non
e' is deprecated. Select only valid columns or specify the value of numeric_onl
y to silence this warning.
  race_means = filtered_data[['Black', 'White', 'Asian', 'Hispanic', 'American
Indian/Alaska Native']].mean()
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_12988\2149650727.py:5: FutureWarnin
g: The default value of numeric_only in DataFrame.sum is deprecated. In a futur
e version, it will default to False. In addition, specifying 'numeric_only=Non
e' is deprecated. Select only valid columns or specify the value of numeric_onl
y to silence this warning.
  Asian_white = filtered_data[['Asian', 'White']].sum()
```

In [36]:
```
def diff_track(s1,s2):
    if len(s1)>len(s2):
        diff=len(s1)-len(s2)
        s1[:diff]
    elif len(s2)>len(s1):
        diff=len(s2)-len(s1)
        s2[:diff]
    else:
        diff=0
    for i in range(len(s1)):
        if s1[i]!= s2[i]:
            diff+=1
    return diff
print("Distance 1:",diff_track("Asian","Indian"))
print("Distance 2:",diff_track("Min degree","Max degrees"))
```

```
Distance 1: 6
Distance 2: 3
```

In [38]:
```
df3 = pd.DataFrame(data)
df3['Sex'] = df3['Sex'].replace({'F': 0, 'M': 1, 'A': 2})
```

In [38]:
```
df3 = pd.DataFrame(data)
df3['Sex'] = df3['Sex'].replace({'F': 0, 'M': 1, 'A': 2})
df3['Min degree'] = df3['Min degree'].replace({'high school': 0, "associate's": 1, "bachelor's": 2, "maste
result_df = df3[['Sex', 'Min degree']]
print(result_df)
```

```
     Sex  Min degree
0      2           0
1      2           0
2      2           0
3      2           0
4      2           0
..   ...         ...
209    0           3
210    0           3
211    0           3
212    0           3
213    0           3

[214 rows x 2 columns]
```

In [ ]:
```

```