In [1]:
```
#importing necessary libraries
import warnings
warnings.filterwarnings("ignore")

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv("IQ_level.csv")
df
```

	rank	country	IQ	education_expenditure	avg_income	avg_temp
0	1	Hong Kong	106	1283.0	35304.0	26.2
1	2	Japan	106	1340.0	40964.0	19.2
2	3	Singapore	106	1428.0	41100.0	31.5
3	4	Taiwan	106	NaN	NaN	26.9
4	5	China	104	183.0	4654.0	19.1
...	...	...	...	...	...	...
103	104	Equatorial Guinea	56	NaN	7625.0	29.9
104	105	Gambia	55	14.0	648.0	32.9
105	106	Guatemala	55	92.0	2830.0	32.1
106	107	Sierra Leone	52	16.0	412.0	30.4
107	108	Nepal	51	22.0	595.0	24.6

108 rows × 6 columns

Our variable features

--countries : contains the country names
--iq : contains the IQ scores
--education_expenditure : contains the education expenditure
--avg_income : contains the average income
--avg_temp : contains the average temperature

Goal : Analyzing the correlation between IQ ar

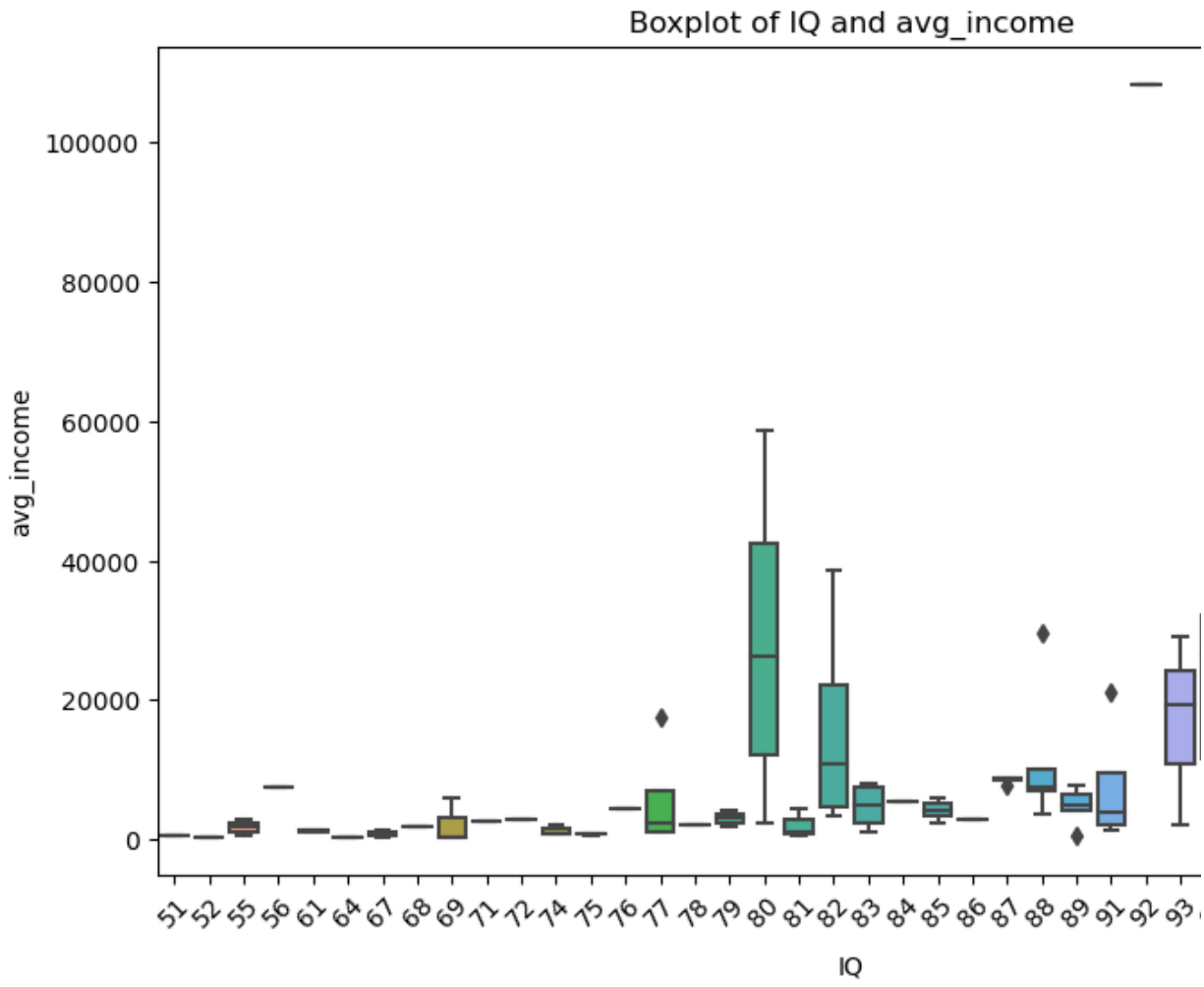
Introduction

```
>>> National IQ scores are an intriguing topic.
Many factors potentially influence IQ differences between countries.
By visualizing data, we can analyze these relationships.
The analysis reveals interesting correlations.
These insights raise questions about how countries can support intellectual
```

In [3]:
```
df.duplicated().sum()
```

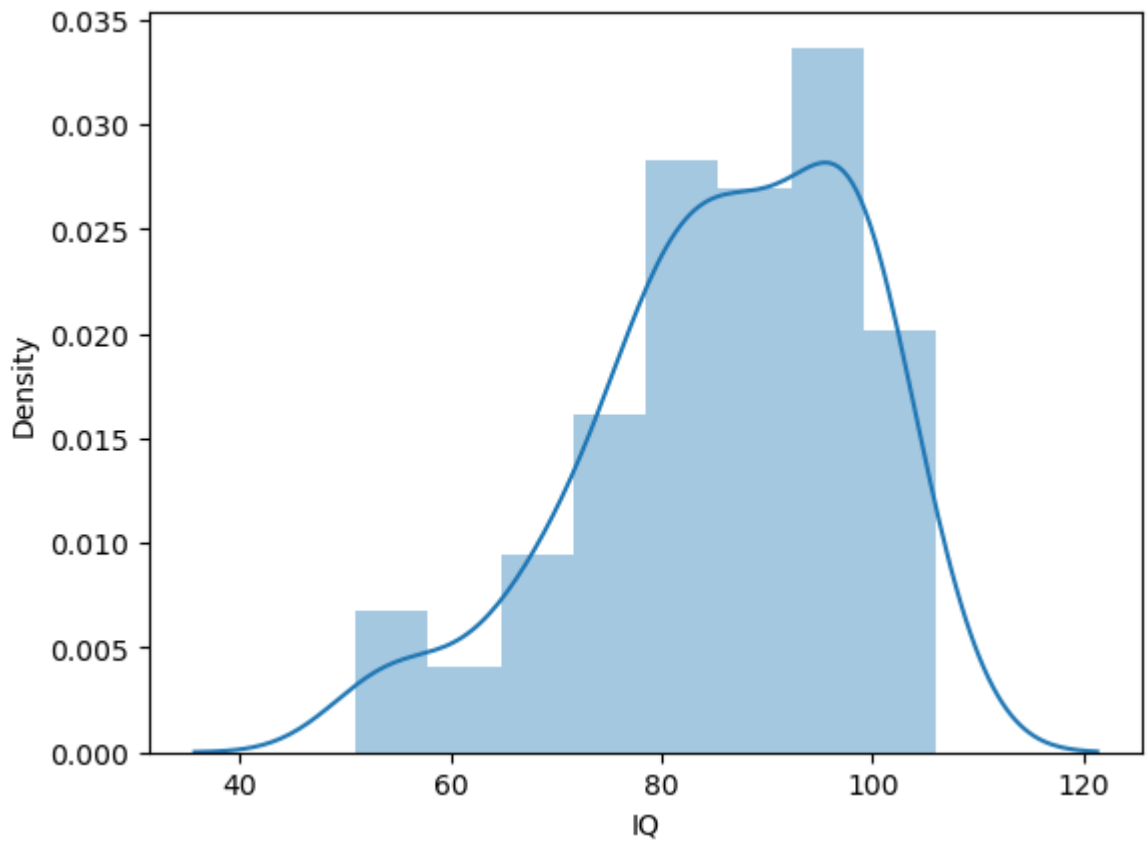
```
0
```

In [4]:
```
plt.figure(figsize=(10,6))
sns.boxplot(data=df,x="IQ",y="avg_income")
plt.xticks(rotation=45)
plt.title("Boxplot of IQ and avg_income")
plt.show()
```

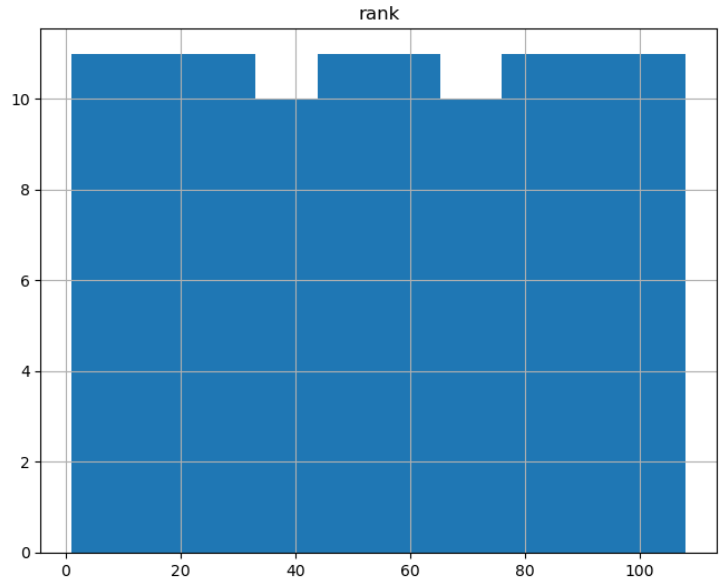


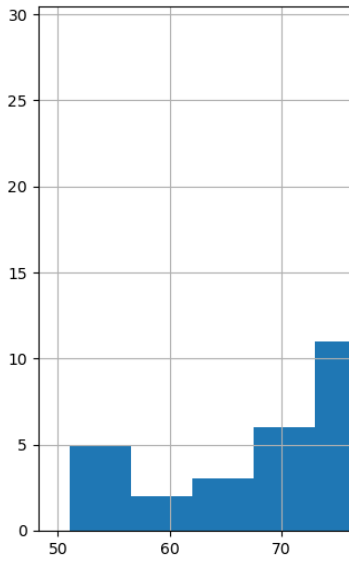
In [5]:
```
sns.distplot(df["IQ"])
```

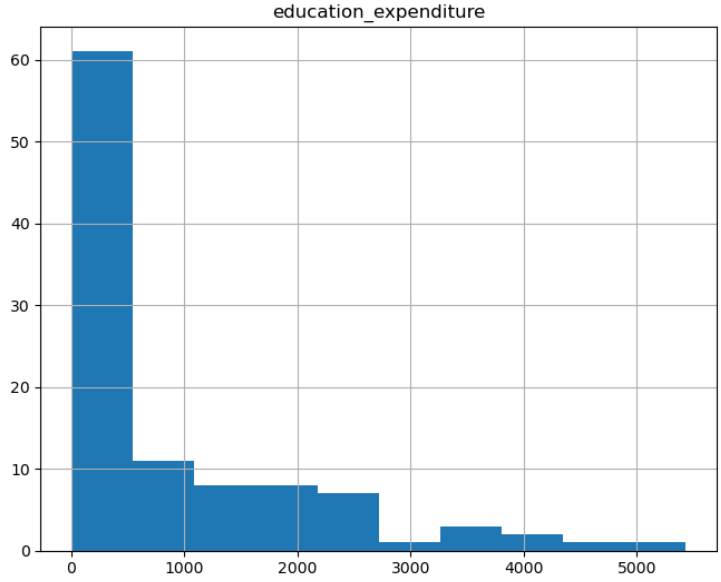
```
<Axes: xlabel='IQ', ylabel='Density'>
```

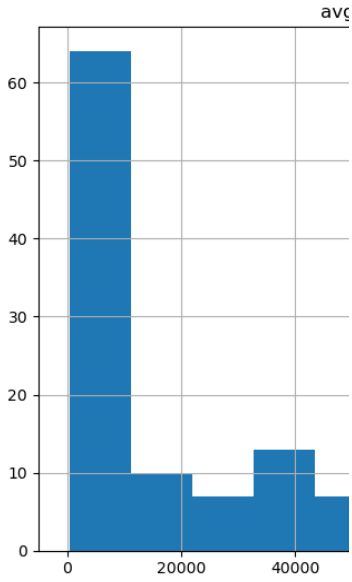


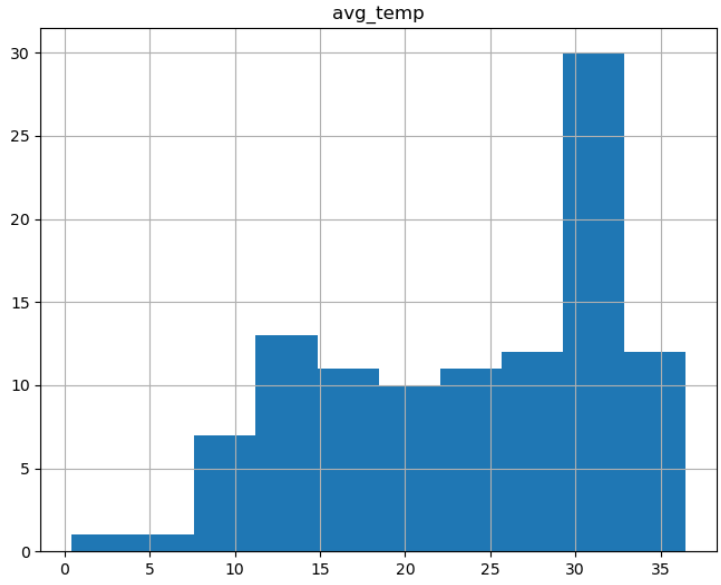
In [6]:
```
df.hist(figsize=(18,22))
plt.show()
```


rank
0
2
4
6
8
10
0
20
40
60
80
100


0
5
10
15
20
25
30
50
60
70

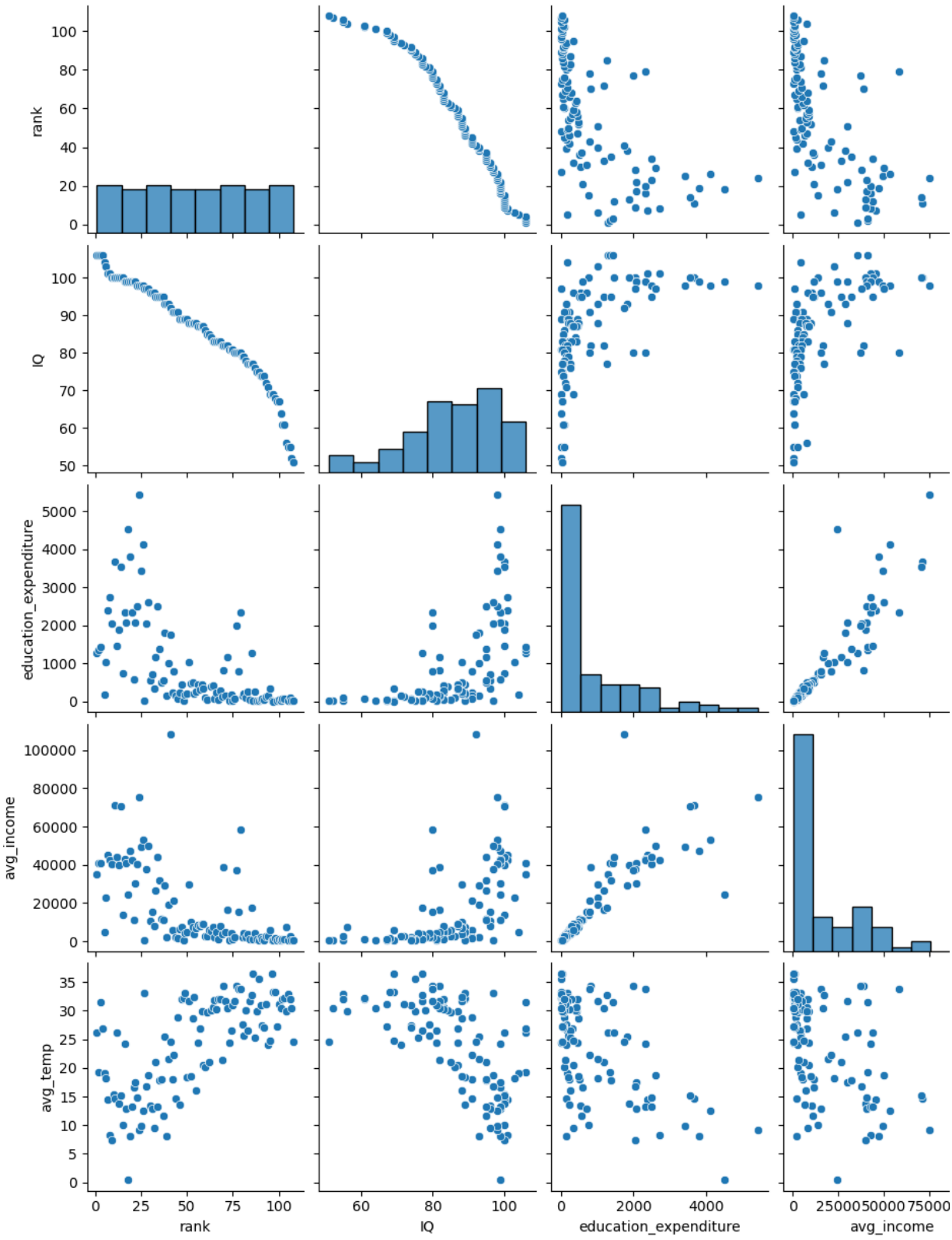

education_expenditure
0
10
20
30
40
50
60
0
1000
2000
3000
4000
5000


avg
0
10
20
30
40
50
60
0
20000
40000


avg_temp
0
5
10
15
20
25
30
0
5
10
15
20
25
30
35

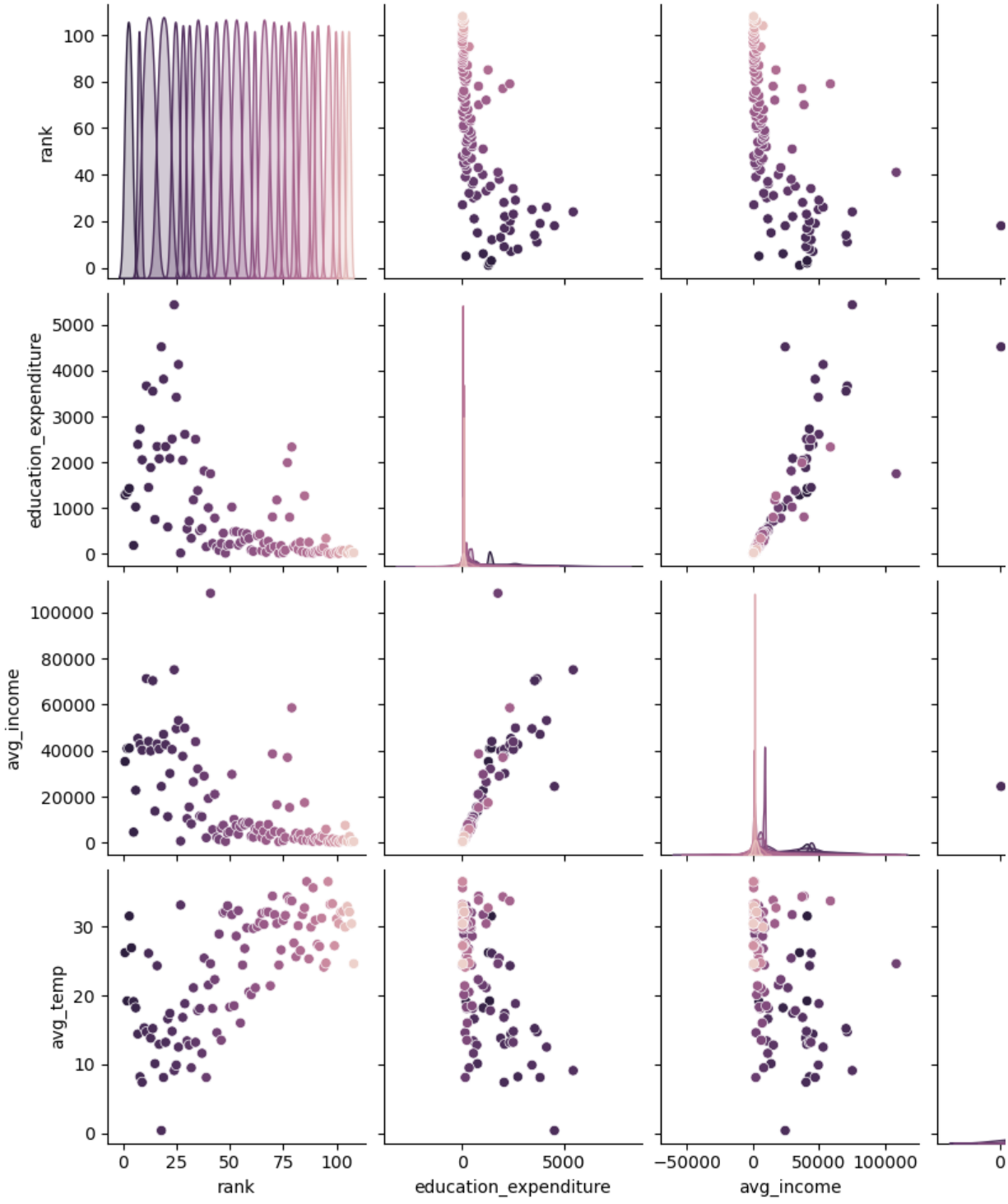
In [7]:
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x1b87736bdf0>
```



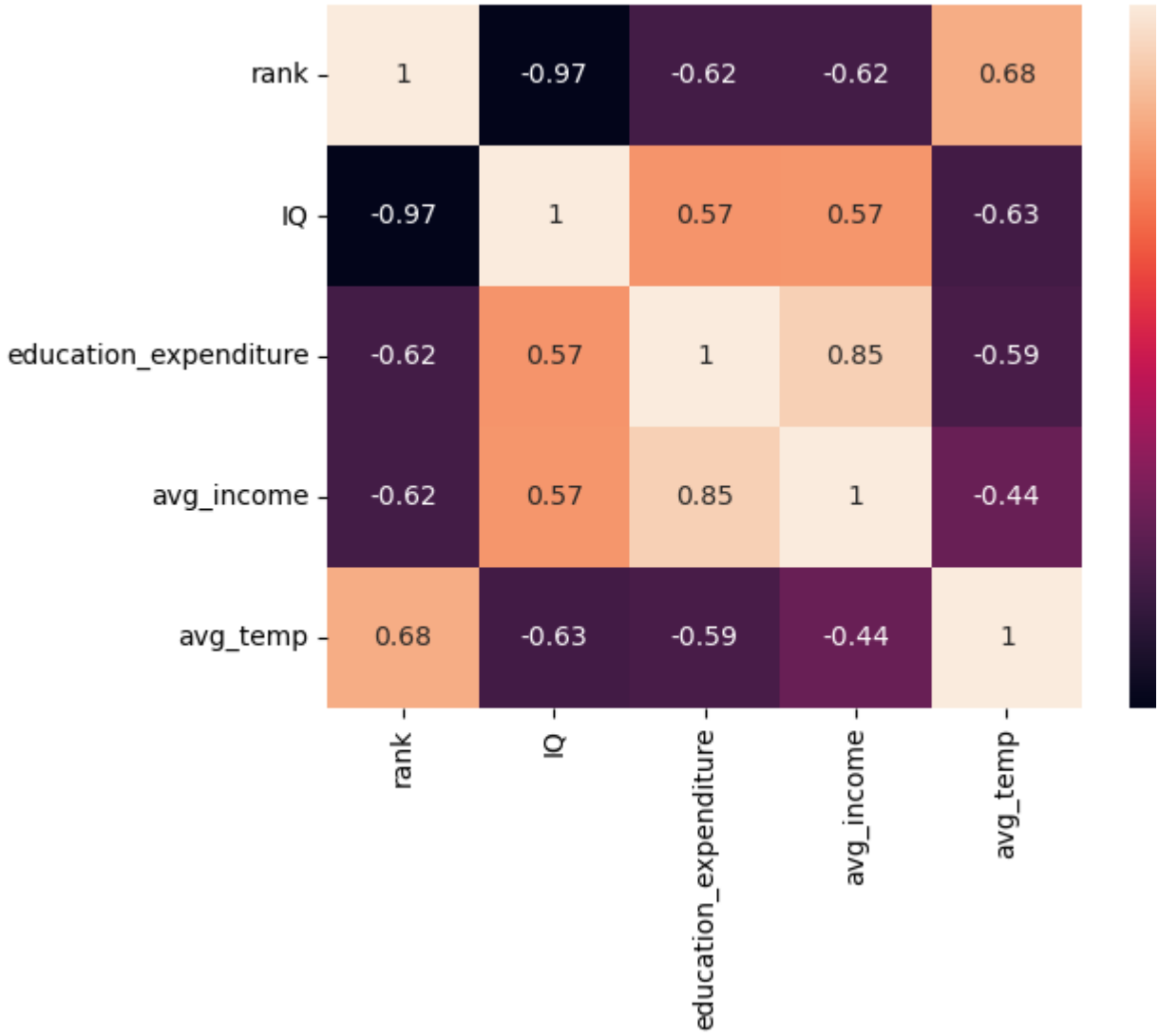
In [8]:
```
sns.pairplot(df,hue="IQ")
```

```
<seaborn.axisgrid.PairGrid at 0x1b874b37370>
```



In [9]:
```
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```



Data cleaning

In [10]:
```
df.isnull().sum()
```

```
rank                    0
country                 0
IQ                      0
education_expenditure   5
avg_income              2
avg_temp                0
dtype: int64
```

In [11]:
```
sns.heatmap(df.isnull(),yticklabels="False",cbar=False,cmap="viridis")
plt.show()
```

In [12]:
```
df.dropna(subset=df.columns[3:5],inplace=True)
```

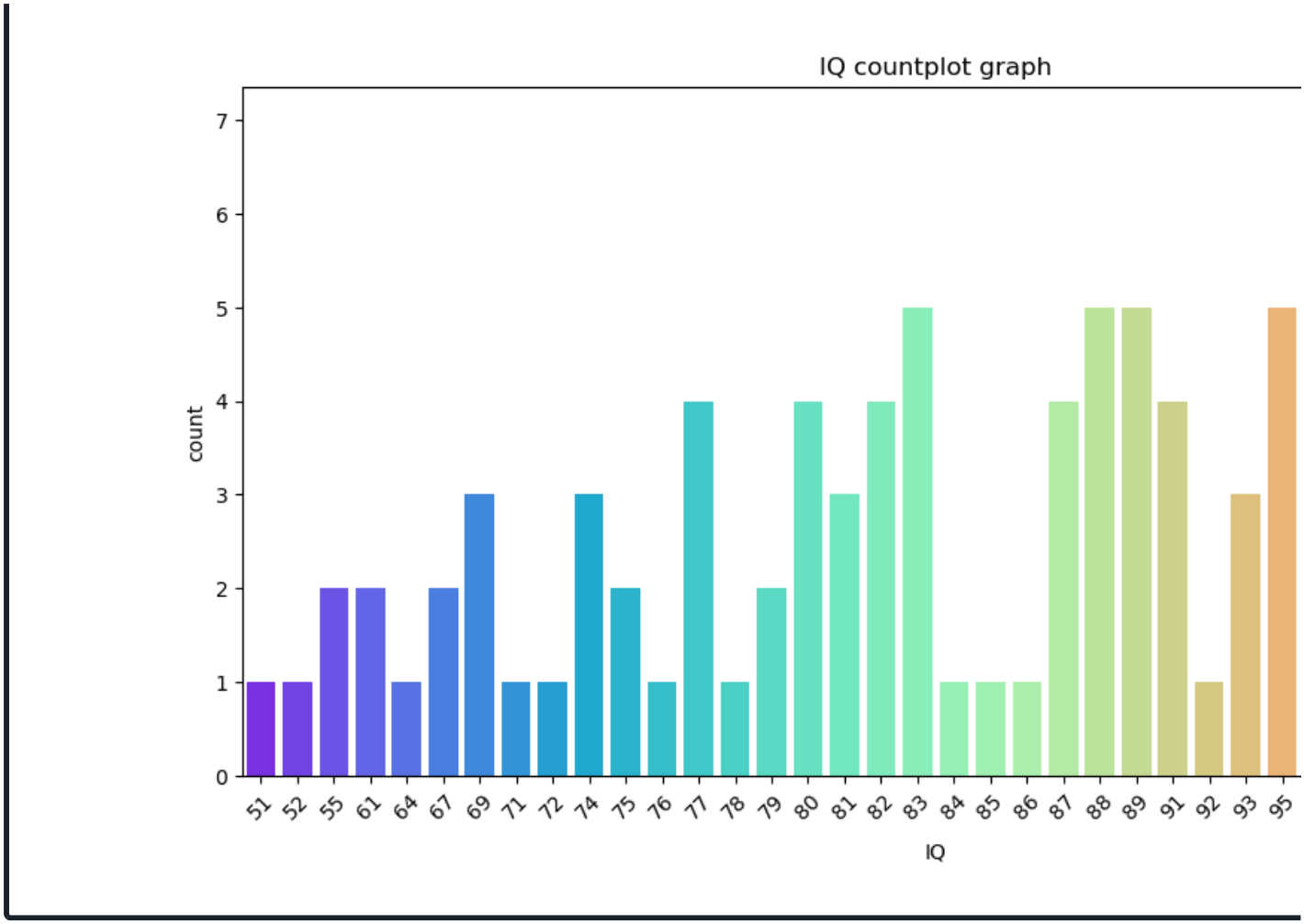
In [13]:
```
df.isnull().sum()
```

```
rank                    0
country                 0
IQ                      0
education_expenditure   0
avg_income              0
avg_temp                0
dtype: int64
```

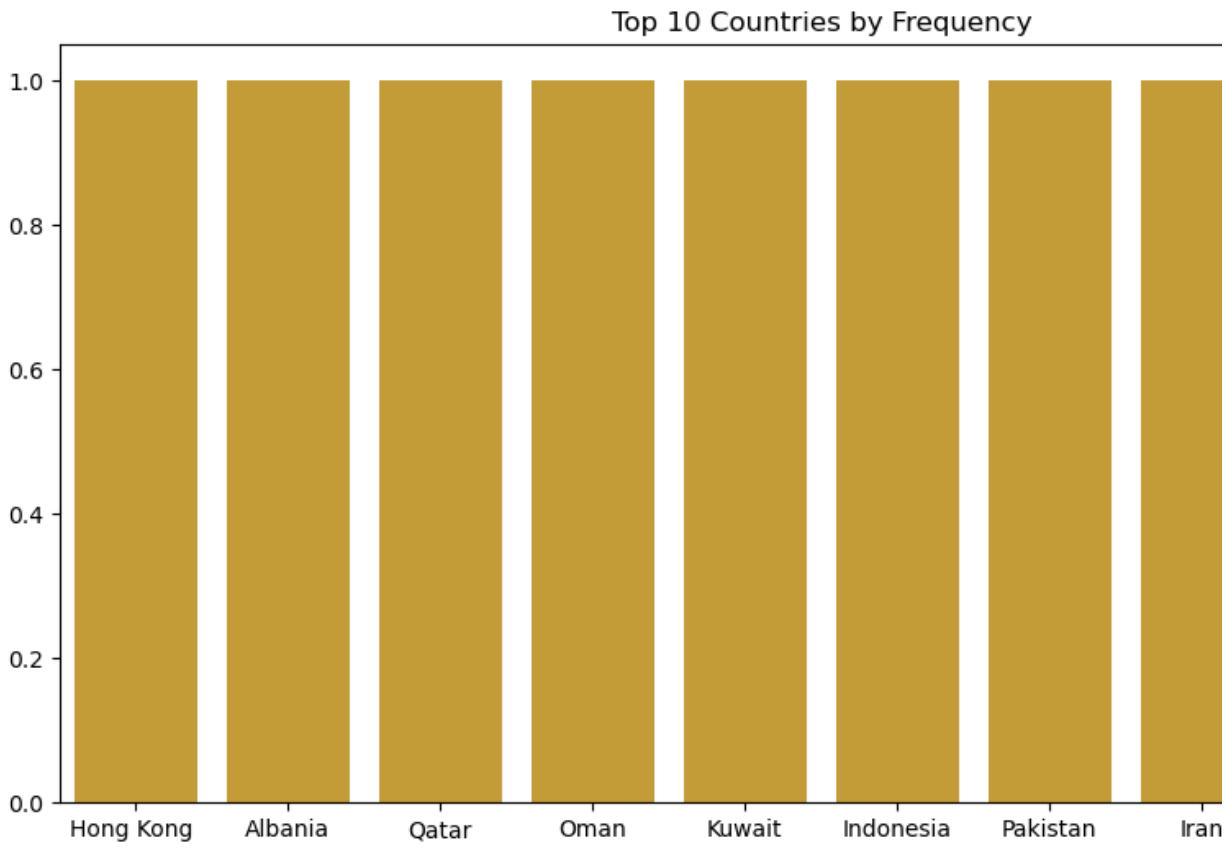
Data visualization

In [24]:
```
#countplot of IQ
plt.figure(figsize=(12,6))
sns.countplot(data=df,x="IQ",palette="rainbow")
plt.title("IQ countplot graph")
plt.xticks(rotation=45)
```

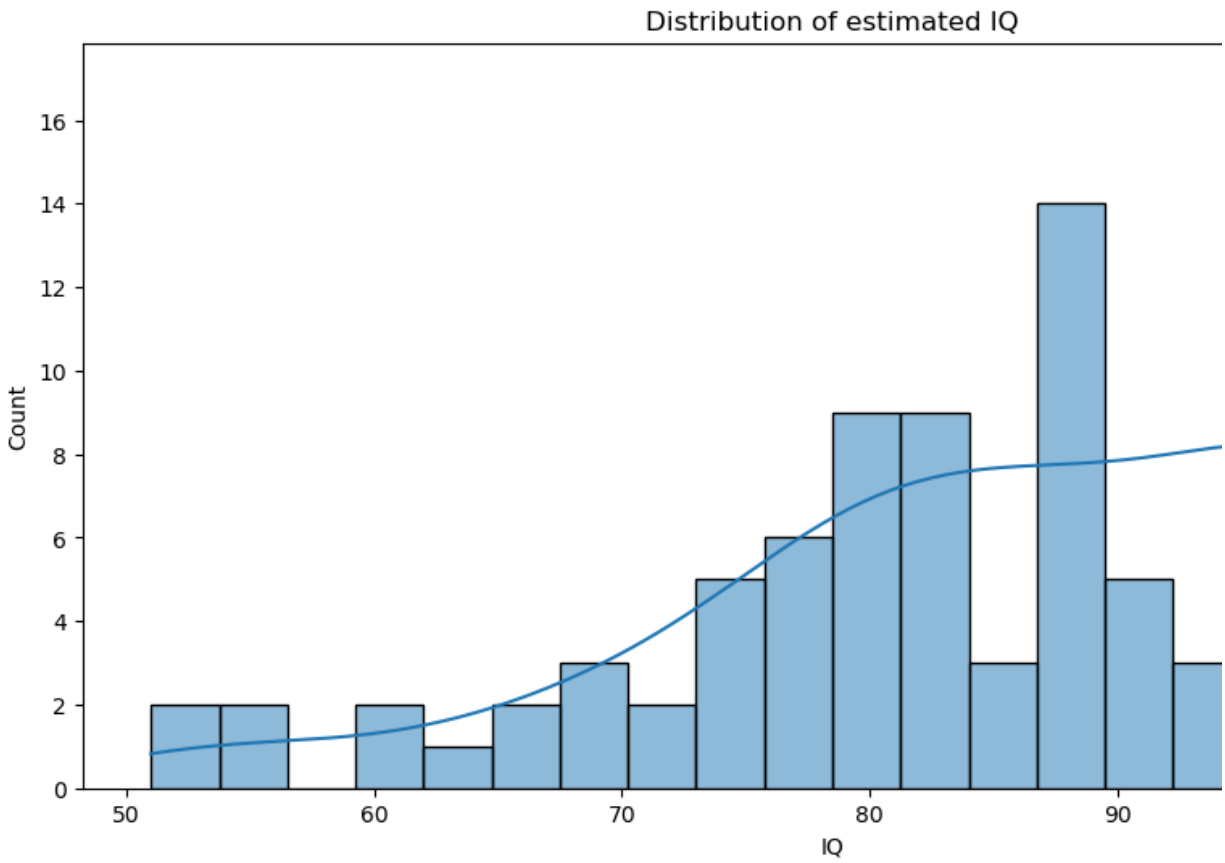
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37]),
 [Text(0, 0, '51'),
  Text(1, 0, '52'),
  Text(2, 0, '55'),
  Text(3, 0, '61'),
  Text(4, 0, '64'),
  Text(5, 0, '67'),
  Text(6, 0, '69'),
  Text(7, 0, '71'),
  Text(8, 0, '72'),
  Text(9, 0, '74'),
  Text(10, 0, '75'),
  Text(11, 0, '76'),
  Text(12, 0, '77'),
  Text(13, 0, '78'),
  Text(14, 0, '79'),
  Text(15, 0, '80'),
  Text(16, 0, '81'),
  Text(17, 0, '82'),
  Text(18, 0, '83'),
  Text(19, 0, '84'),
  Text(20, 0, '85'),
  Text(21, 0, '86'),
  Text(22, 0, '87'),
  Text(23, 0, '88'),
  Text(24, 0, '89'),
  Text(25, 0, '91'),
  Text(26, 0, '92'),
  Text(27, 0, '93'),
  Text(28, 0, '95'),
  Text(29, 0, '96'),
  Text(30, 0, '97'),
  Text(31, 0, '98'),
  Text(32, 0, '99'),
  Text(33, 0, '100'),
  Text(34, 0, '101'),
  Text(35, 0, '103'),
  Text(36, 0, '104'),
  Text(37, 0, '106')])
```


IQ countplot graph
count
0
1
2
3
4
5
6
7
51
52
55
61
64
67
69
71
72
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
91
92
93
95
IQ

In [15]:
```
top_10_country_count = df["country"].value_counts().nlargest(10)

plt.figure(figsize=(12,6))

sns.barplot(x=top_10_country_count.index,
            y=top_10_country_count.values,
            color="goldenrod")

plt.title("Top 10 Countries by Frequency")
plt.xlabel=("Country")
plt.ylabel=("Frequency")

plt.show()
```

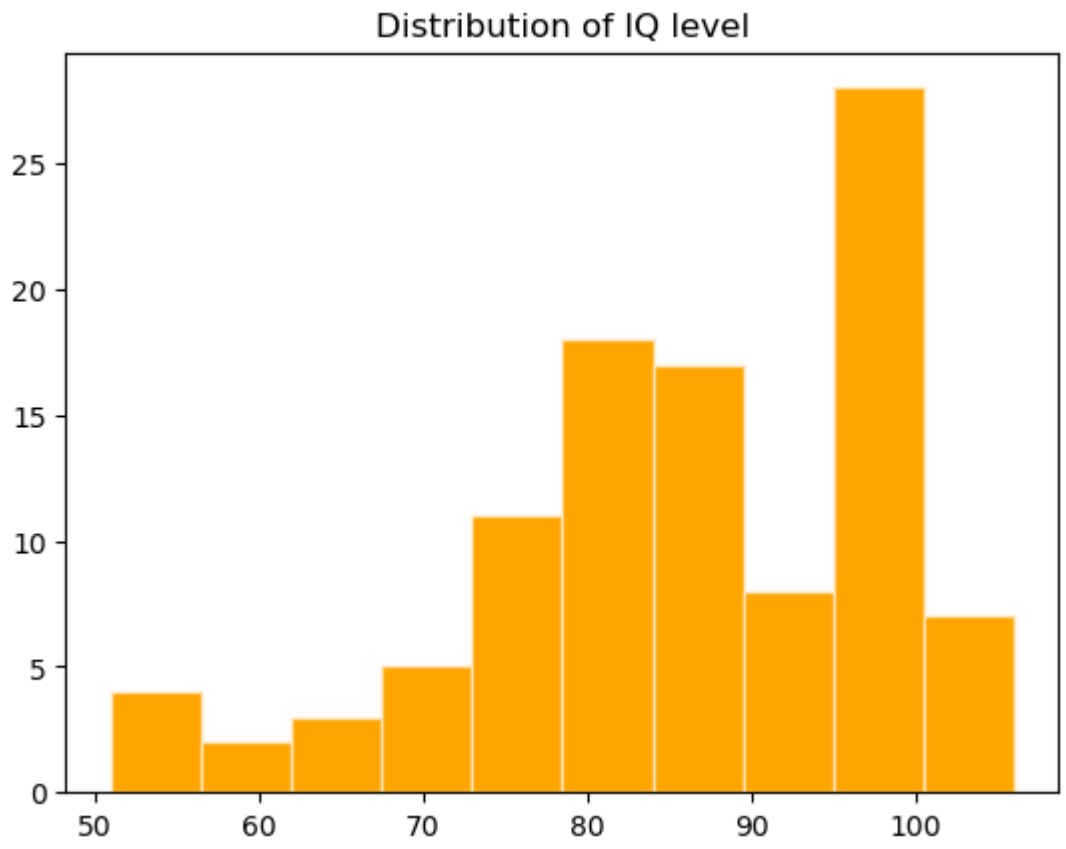


In [16]:
```
#DIstribution of estimates IQ
plt.figure(figsize=(12,6))
sns.histplot(df["IQ"],bins=20,kde=True)
plt.title("Distribution of estimated IQ")
plt.ylabel=("frequency")
plt.show()
```



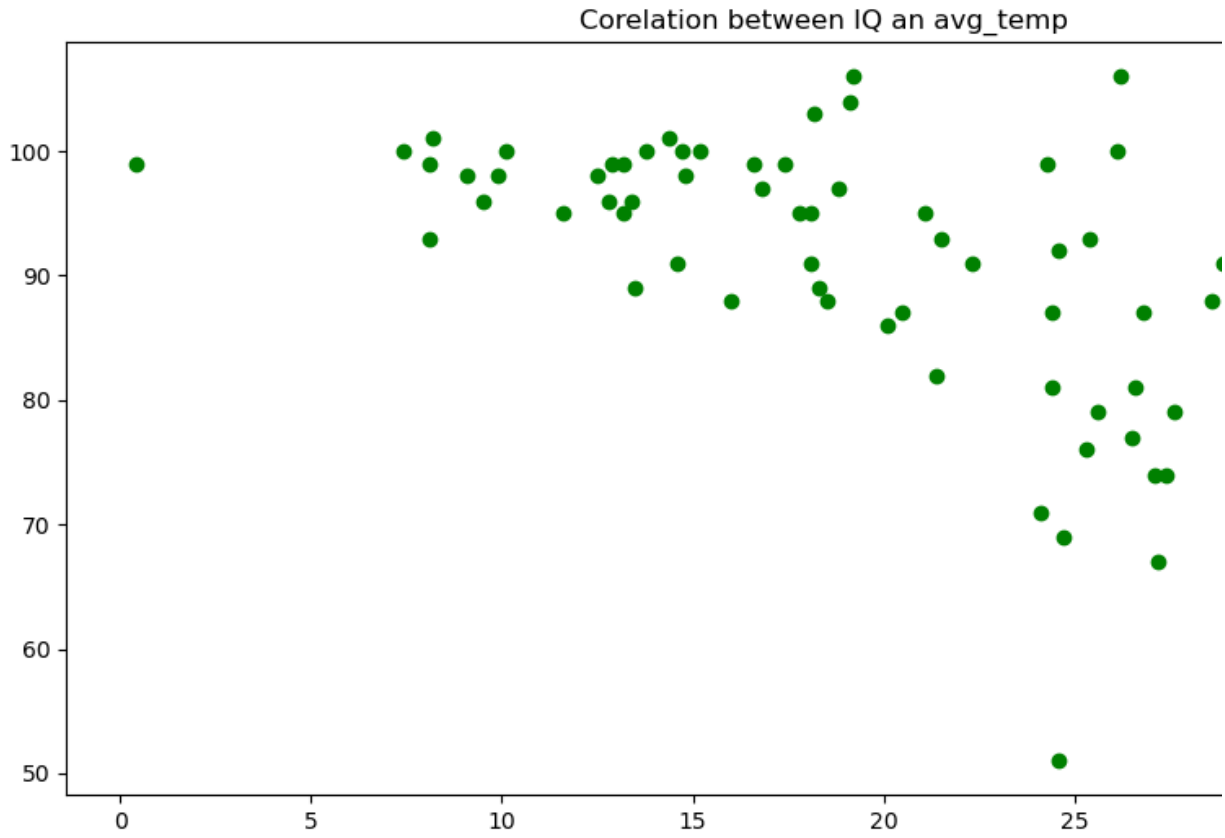
In [28]:

```
#Distribution of IQ score

plt.hist(data=df,x="IQ",color="orange",edgecolor="moccasin")
plt.title("Distribution of IQ level")
plt.xlabel=("IQ")
plt.ylabel=("count")
plt.show()
```

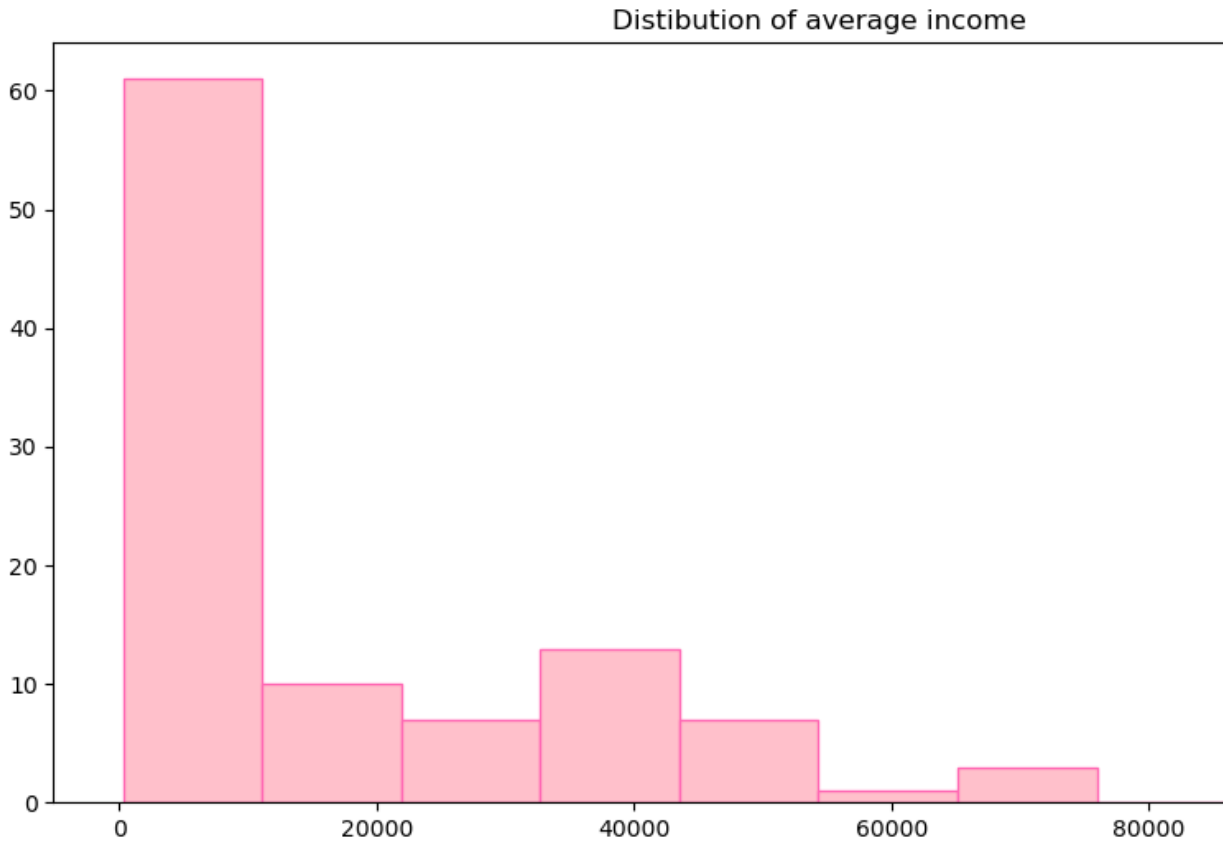


In [18]:
```
plt.figure(figsize=(12,6))
plt.scatter(data=df,x="avg_temp",y="IQ",color="green")
plt.title("Corelation between IQ an avg_temp")
```

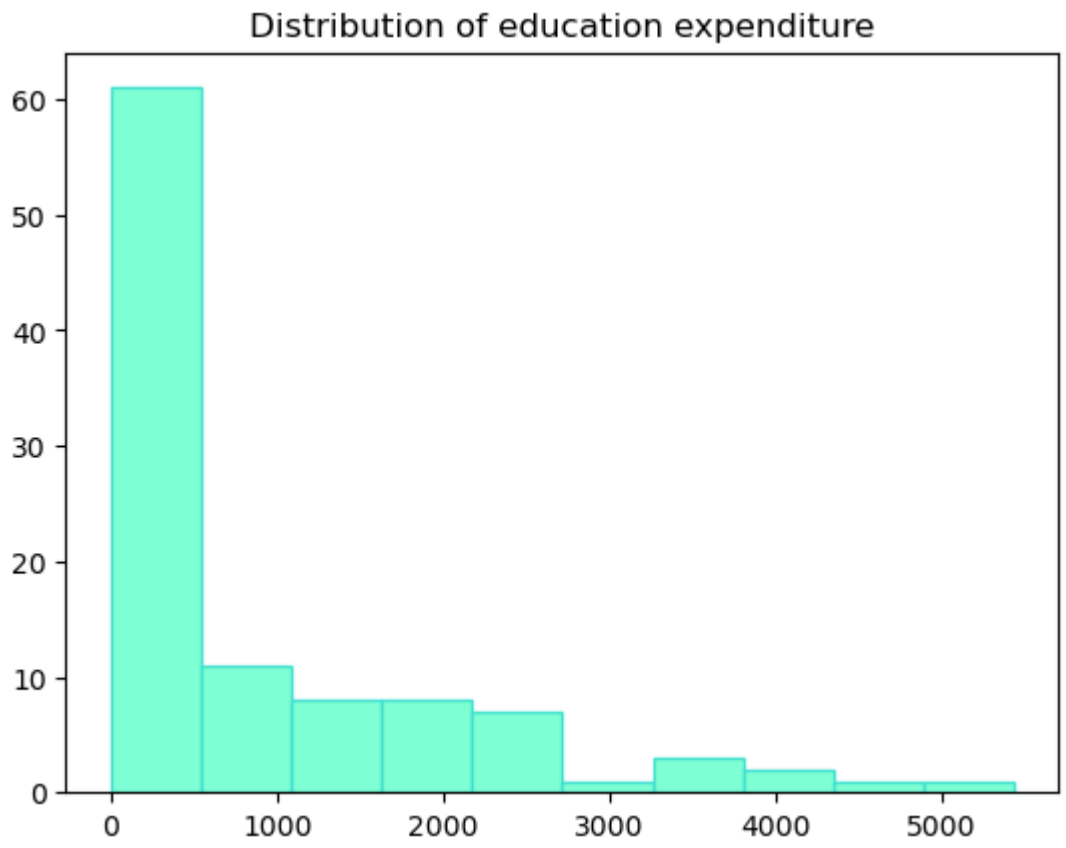
```
Text(0.5, 1.0, 'Corelation between IQ an avg_temp')
```



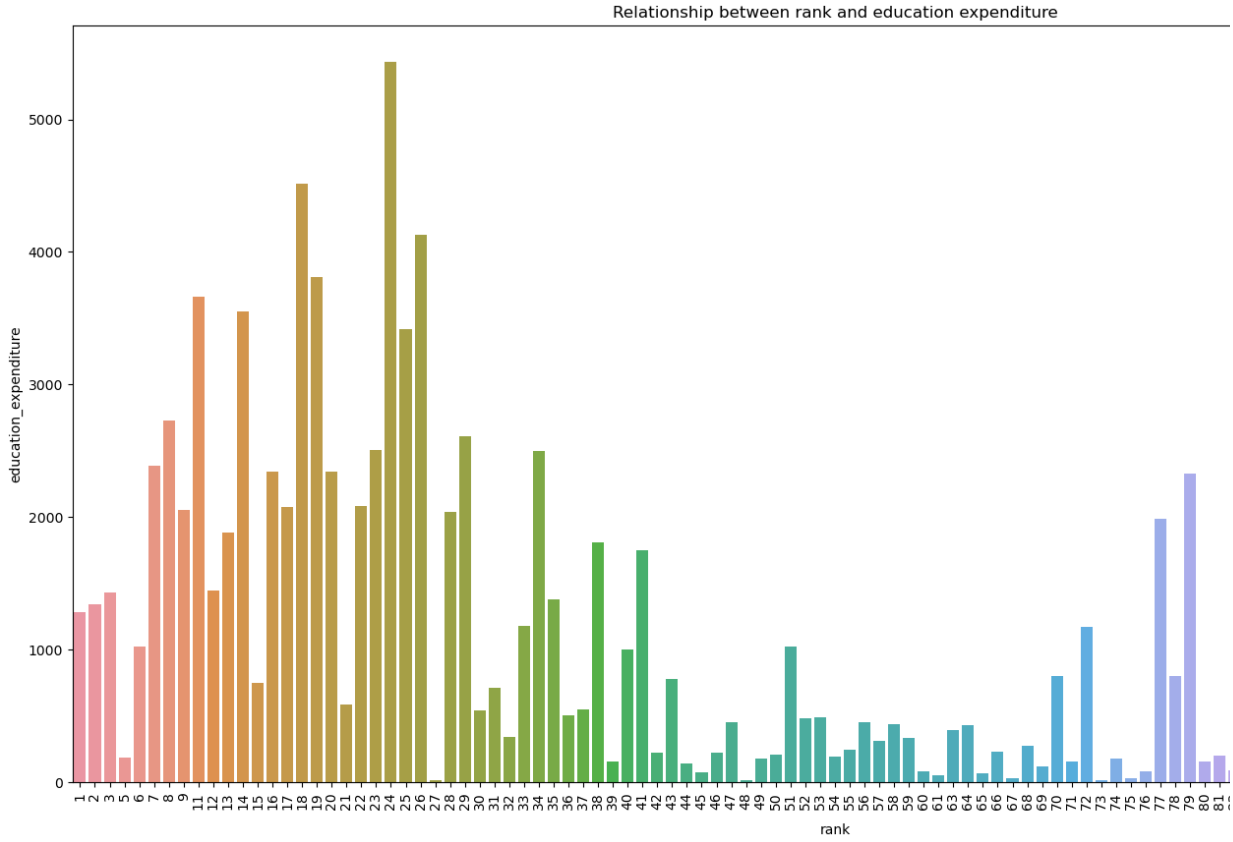
In [29]:
```
#distribution of average income

plt.figure(figsize=(12,6))
plt.hist(data=df,x="avg_income",color="pink",edgecolor="hotpink")
plt.title("Distibution of average income")
plt.show()
```



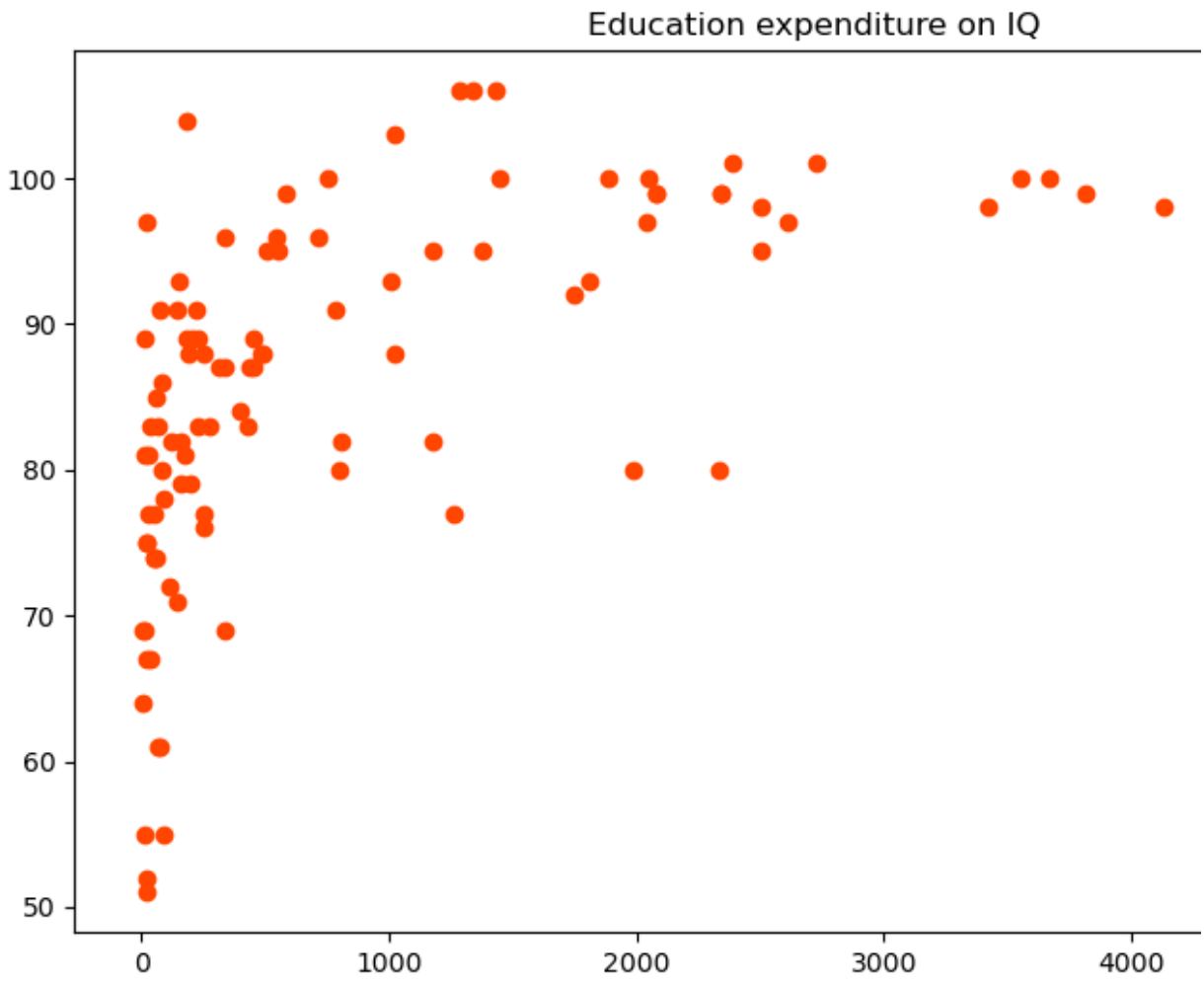
In [31]:
```
#"Distribution of education expenditure"
plt.hist(data=df,x="education_expenditure",color="aquamarine",edgecolor="turq
plt.title("Distribution of education expenditure")
plt.xlabel=("education_expenditure")
plt.ylabel=("count")
plt.show()
```



In [21]:
```
plt.figure(figsize=(20,10))
sns.barplot(data=df,x="rank",y="education_expenditure")
plt.title("Relationship between rank and education expenditure")
plt.xticks(rotation=90)
plt.show()
```



In [22]:
```
# education expenditure on IQ_

plt.figure(figsize=(10,6))
plt.scatter(data=df,y="IQ",x="education_expenditure",color="orangered")
plt.title("Education expenditure on IQ")
plt.show()
```



In [23]:
```
#Correlation between Temperature and IQ

plt.figure(figsize=(10,6))
plt.scatter(data=df,x="avg_temp",y="IQ",color="darkviolet")
plt.title("Correlation between Temperature and IQ")
plt.show()
```