1. Data Cleaning

In [1]:
```
import pandas as pd
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
# Loading the dataset
spectacles_data = pd.read_csv('Dots Potential Customer Survey Data - Sheet1.csv')
```

In [3]:
```
spectacles_data.head()
```

Out[3]:

	Unique ID	Country	Age	Annual Family Income ($)	Gender	Time spent watching videos/TV	Time spent playing indoor sports	Time spent playing outdoor sports	Total Time spent working in front of screen	Sleeping hours	...	sul
0	7319483	CAN	13	20423	Male	3	4	3	7	8	...	
1	4791965	CAN	13	5570	Female	3	3	1	10	3	...	
2	2991718	CAN	13	58706	Female	2	2	1	4	9	...	
3	4220106	CAN	13	57118	Male	6	2	4	10	11	...	
4	2263008	CAN	14	59834	Male	6	3	4	13	12	...	

5 rows × 29 columns

In [4]:
```
# Displaying basic information about the dataset

print(spectacles_data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3220 entries, 0 to 3219
Data columns (total 29 columns):
 #   Column                                                Non-Null Count  Dtype
---  ------                                                --------------  -----
 0   Unique ID                                             3220 non-null   int64
 1   Country                                               3220 non-null   object
 2   Age                                                   3220 non-null   int64
 3   Annual Family Income ($)                              3220 non-null   int64
 4   Gender                                                3220 non-null   object
 5   Time spent watching videos/TV                         3220 non-null   int64
 6   Time spent playing indoor sports                      3220 non-null   int64
 7   Time spent playing outdoor sports                     3220 non-null   int64
 8   Total Time spent working in front of screen           3220 non-null   int64
 9   Sleeping hours                                        3220 non-null   int64
 10  IQ                                                    3220 non-null   int64
 11  Whether parents have specs                            3220 non-null   int64
 12  English speaker                                       3220 non-null   int64
 13  Migrated within country                               3220 non-null   int64
 14  Migrated overseas                                     3220 non-null   int64
 15  Maritial Status (0 - Single, 1 - Married, 2 - Divorced)  3220 non-null   int64
 16  Has Diabetes                                          3220 non-null   int64
 17  Education Status                                      3220 non-null   object
 18  Has Gym Subscription                                  3220 non-null   int64
 19  Has OTT subscription                                  3220 non-null   int64
 20  Number of friends                                     3220 non-null   int64
 21  Likes spicy food                                      3220 non-null   int64
 22  Likes desserts                                        3220 non-null   int64
 23  Wants to change career                                3220 non-null   int64
 24  Has debt                                              3220 non-null   int64
 25  Has kids                                              3220 non-null   int64
 26  Drinks alcohol                                        3220 non-null   int64
 27  Smoker                                                3220 non-null   int64
 28  Wear Specs                                            3220 non-null   int64
dtypes: int64(26), object(3)
memory usage: 729.7+ KB
None
```

In [5]:
```
spectacles_data.describe()
```

Out[5]:

	Unique ID	Age	Annual Family Income ($)	Time spent watching videos/TV	Time spent playing indoor sports	Time spent playing outdoor sports	Total Time spent working in front of screen
count	3.220000e+03	3220.000000	3220.000000	3220.000000	3220.000000	3220.000000	3220.000000
mean	4.988139e+06	44.834161	20578.639130	4.347205	1.991925	1.700621	8.162733
std	2.867911e+06	19.073161	15079.429422	2.807803	1.205222	1.190910	2.759419
min	1.683000e+03	13.000000	10.000000	0.000000	0.000000	-2.000000	-2.000000
25%	2.516499e+06	29.000000	8494.000000	2.000000	1.000000	1.000000	6.000000
50%	5.041256e+06	44.000000	16982.000000	4.000000	2.000000	2.000000	8.000000
75%	7.459871e+06	60.000000	30965.250000	6.000000	3.000000	3.000000	10.000000
max	9.999011e+06	91.000000	59858.000000	10.000000	4.000000	5.000000	18.000000

8 rows × 26 columns

In [6]:
```
spectacles_data.columns
```

Out[6]:
```
Index(['Unique ID', 'Country', 'Age', 'Annual Family Income ($)', 'Gender',
       'Time spent watching videos/TV', 'Time spent playing indoor sports',
       'Time spent playing outdoor sports',
       'Total Time spent working in front of screen', 'Sleeping hours', 'IQ',
       'Whether parents have specs', 'English speaker',
       'Migrated within country', 'Migrated overseas',
       'Maritial Status (0 - Single, 1 - Married, 2 - Divorced)',
       'Has Diabetes', 'Education Status', 'Has Gym Subscription',
       'Has OTT subscription', 'Number of friends', 'Likes spicy food',
       'Likes desserts', 'Wants to change career', 'Has debt', 'Has kids',
       'Drinks alcohol', 'Smoker', 'Wear Specs'],
      dtype='object')
```

In [7]:
```
spectacles_data.isnull().sum()
```

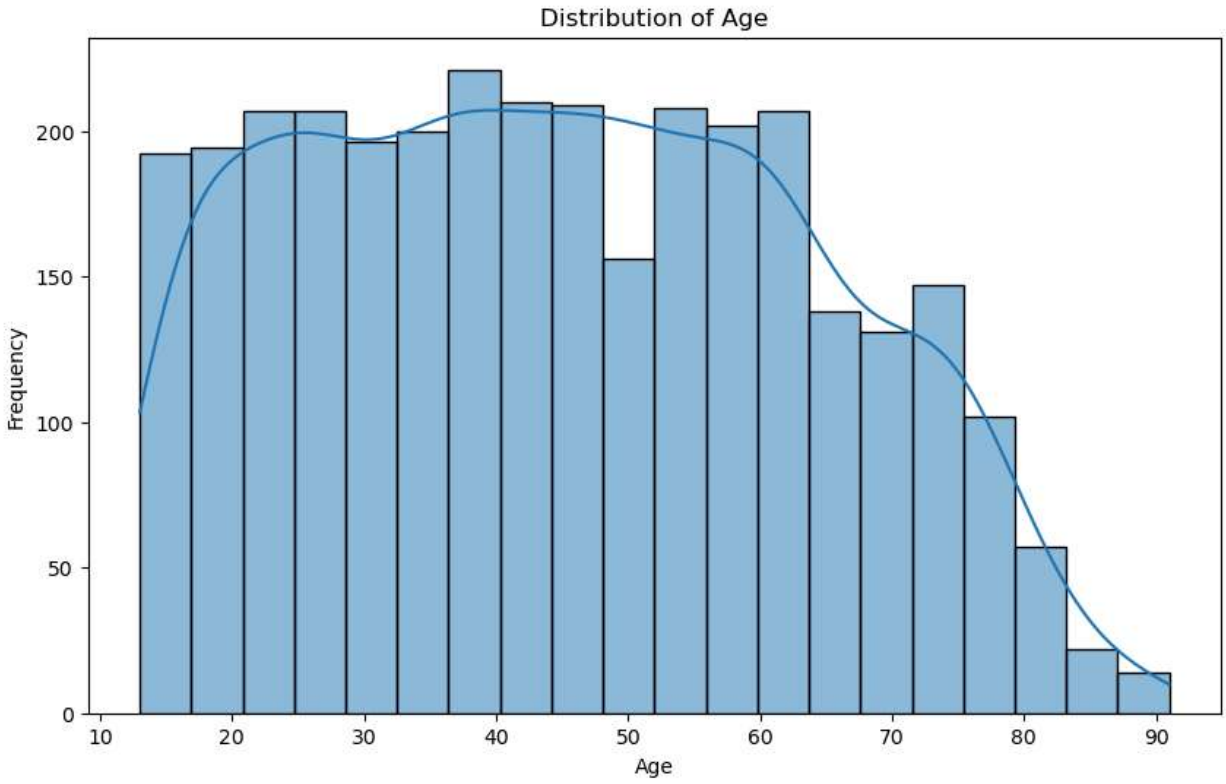
Out[7]:
```
Unique ID                                                0
Country                                                  0
Age                                                      0
Annual Family Income ($)                                 0
Gender                                                   0
Time spent watching videos/TV                            0
Time spent playing indoor sports                         0
Time spent playing outdoor sports                        0
Total Time spent working in front of screen              0
Sleeping hours                                           0
IQ                                                       0
Whether parents have specs                               0
English speaker                                          0
Migrated within country                                  0
Migrated overseas                                        0
Maritial Status (0 - Single, 1 - Married, 2 - Divorced)  0
Has Diabetes                                             0
Education Status                                         0
Has Gym Subscription                                     0
Has OTT subscription                                     0
Number of friends                                        0
Likes spicy food                                         0
Likes desserts                                           0
Wants to change career                                   0
Has debt                                                 0
Has kids                                                 0
Drinks alcohol                                           0
Smoker                                                   0
Wear Specs                                               0
dtype: int64
```

In [8]:
```
# Handling missing values

spectacles_data.dropna(inplace=True)
```

In [9]:
```
# Removing duplicate rows if any

spectacles_data.drop_duplicates(inplace=True)
```

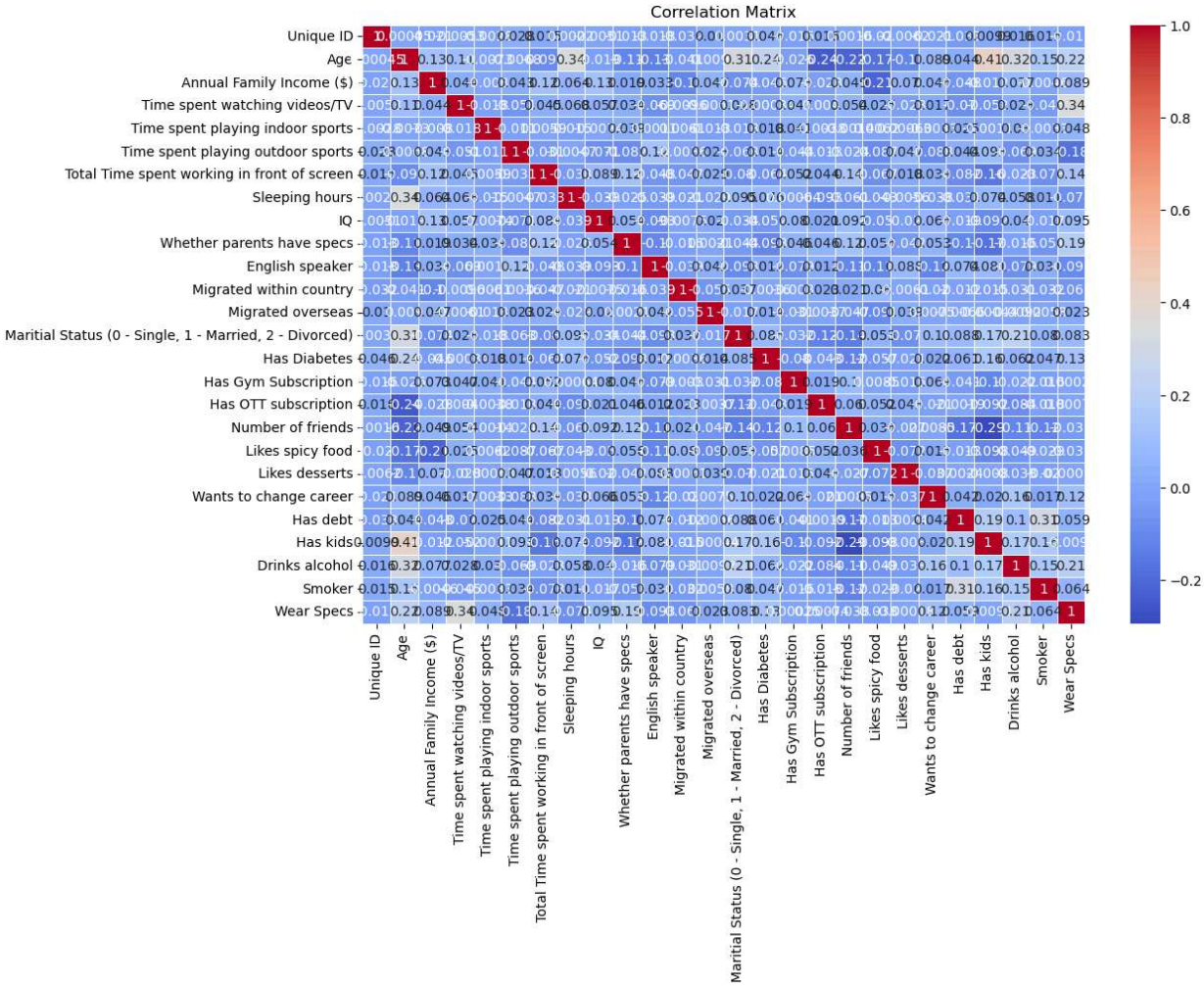
2. Exploratory Data Analysis (EDA)

In [10]:
```
# Exploring the distribution of age

plt.figure(figsize=(10, 6))
sns.histplot(spectacles_data['Age'], bins=20, kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

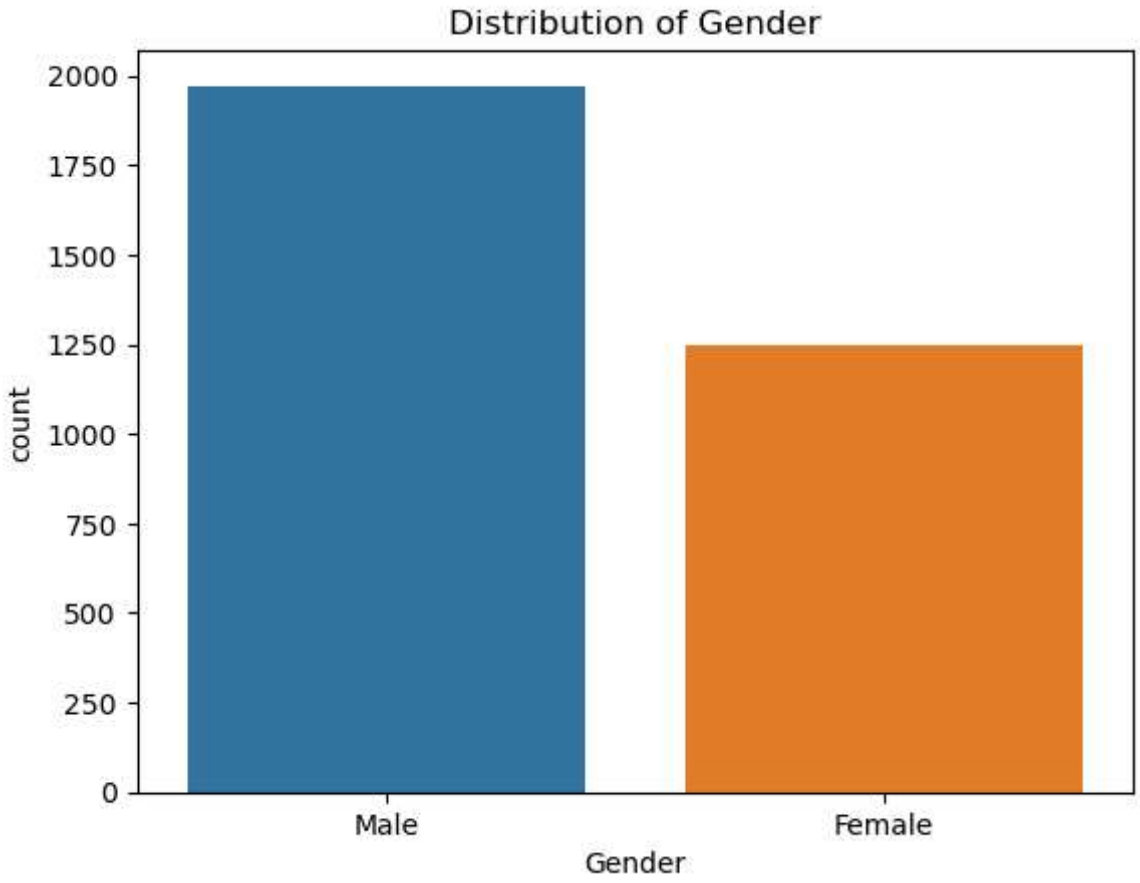


In [11]:
```
# Exploring the correlation between variables

plt.figure(figsize=(12, 8))
sns.heatmap(spectacles_data.corr(), annot=True, cmap='coolwarm', linewidths=.5)
plt.title('Correlation Matrix')
plt.show()
```

```
C:\Users\Deviare User\AppData\Local\Temp\ipykernel_7932\3879473017.py:4: FutureWarnin
g: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of num
eric_only to silence this warning.
  sns.heatmap(spectacles_data.corr(), annot=True, cmap='coolwarm', linewidths=.5)
```

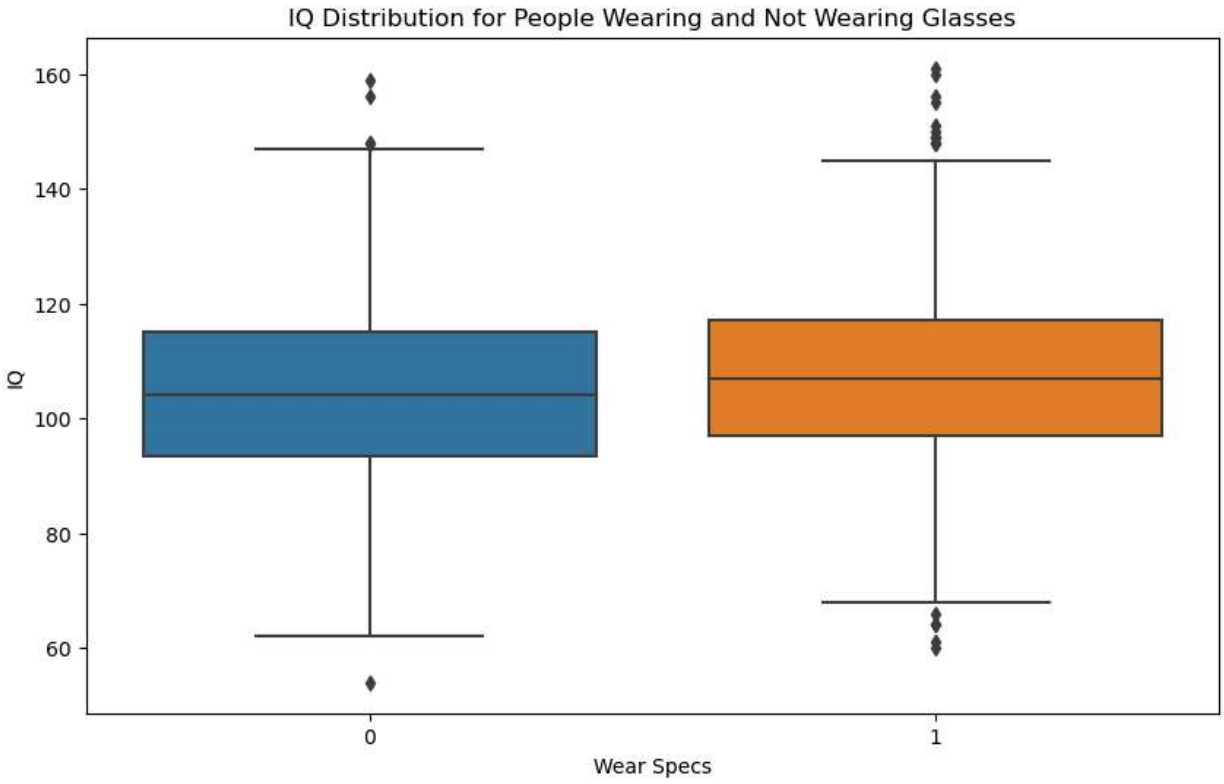


In [12]:
```
# Exploring categorical variables

sns.countplot(x='Gender', data=spectacles_data)
plt.title('Distribution of Gender')
plt.show()
```

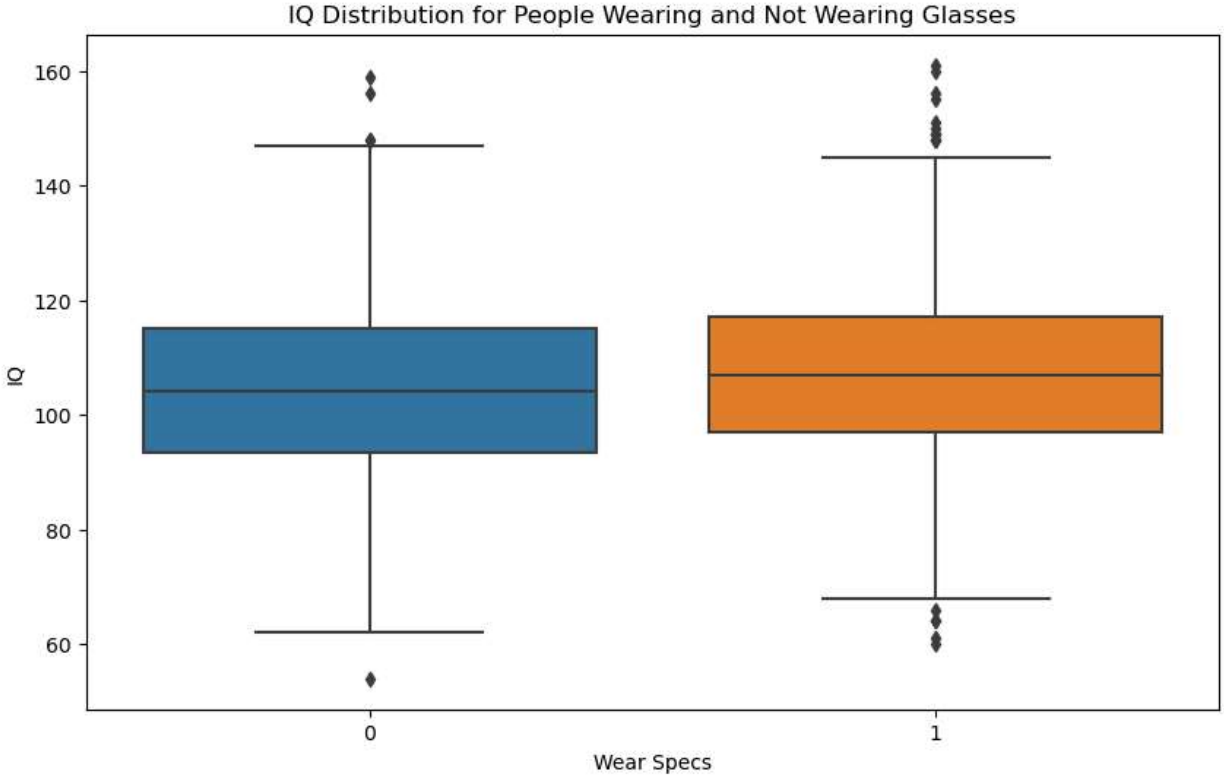


3. Plotting Data for Insights

In [13]:
```
# Exploring the distribution of health scores for people wearing glasses and not weari

plt.figure(figsize=(10, 6))
sns.boxplot(x='Wear Specs', y='IQ', data=spectacles_data)
plt.title('IQ Distribution for People Wearing and Not Wearing Glasses')
plt.xlabel('Wear Specs')
plt.ylabel('IQ')
plt.show()
```



In [14]:
```
# Exploring the distribution of health scores for people wearing glasses and not weari

plt.figure(figsize=(10, 6))
sns.boxplot(x='Wear Specs', y='IQ', data=spectacles_data)
plt.title('IQ Distribution for People Wearing and Not Wearing Glasses')
plt.xlabel('Wear Specs')
plt.ylabel('IQ')
plt.show()
```



In [15]:
```
# Calculating the correlation matrix

correlation_matrix = spectacles_data.corr()
correlation_matrix
```

```
C:\Users\Deviare User\AppData\Local\Temp\ipykernel_7932\842463942.py:3: FutureWarnin
g: The default value of numeric_only in DataFrame.corr is deprecated. In a future ver
sion, it will default to False. Select only valid columns or specify the value of num
eric_only to silence this warning.
  correlation_matrix = spectacles_data.corr()
```

Out[15]:

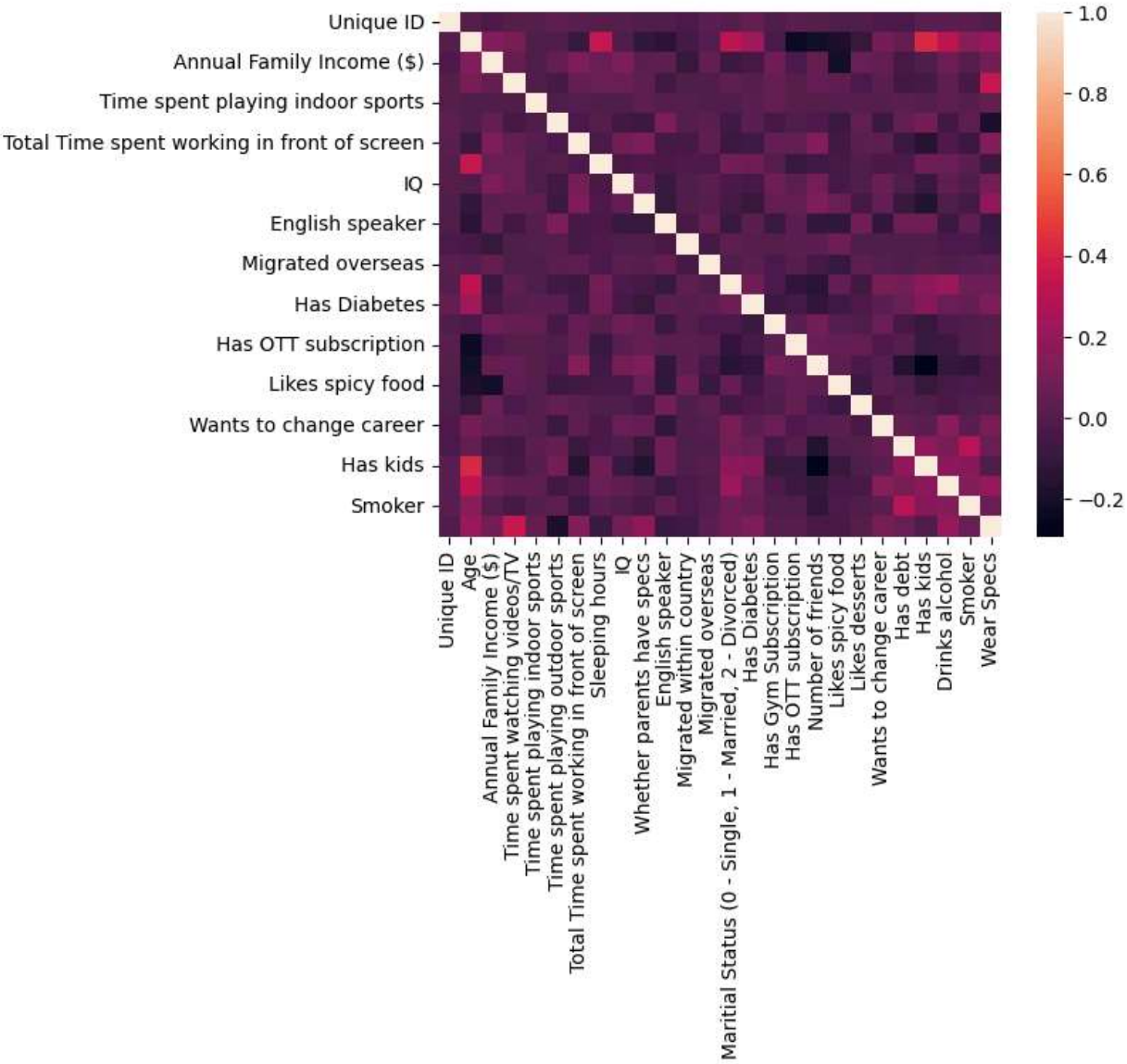
	Unique ID	Age	Annual Family Income ($)	Time spent watching videos/TV	Time spent playing indoor sports	Time spent playing outdoor sports	Total Time spent working in front of screen	Sleeping hours	
Unique ID	1.000000	0.000448	-0.021472	-0.005284	-0.002802	0.028006	0.015158	0.002242	-(
Age	0.000448	1.000000	0.131567	0.109516	-0.007342	-0.006768	-0.096354	0.340297	-(
Annual Family Income ($)	-0.021472	0.131567	1.000000	0.043633	-0.008028	0.042641	0.120710	0.063909	(
Time spent watching videos/TV	-0.005284	0.109516	0.043633	1.000000	-0.017807	-0.051310	0.045070	0.067608	(
Time spent playing indoor sports	-0.002802	-0.007342	-0.008028	-0.017807	1.000000	-0.011424	0.005906	-0.014500	-(
Time spent playing outdoor sports	0.028006	-0.006768	0.042641	-0.051310	-0.011424	1.000000	-0.031208	-0.004720	-(
Total Time spent working in front of screen	0.015158	-0.096354	0.120710	0.045070	0.005906	-0.031208	1.000000	-0.032766	(
Sleeping hours	0.002242	0.340297	0.063909	0.067608	-0.014500	-0.004720	-0.032766	1.000000	-(
IQ	-0.005143	-0.018661	0.130307	0.056773	-0.007369	-0.070595	0.088911	-0.038778	1
Whether parents have specs	-0.012921	-0.114403	0.018905	0.034391	0.039159	-0.081945	0.117013	-0.025053	(
English speaker	-0.018309	-0.127309	0.033134	-0.068643	0.001063	0.118305	-0.047727	-0.039075	-(
Migrated within country	-0.032232	-0.040960	-0.101765	-0.009612	0.006144	-0.003616	-0.047193	-0.020776	-(
Migrated overseas	0.010216	0.008985	0.046809	-0.006058	-0.013498	0.023269	0.029127	-0.029003	(
Maritial Status (0 - Single, 1 - Married, 2 - Divorced)	0.003077	0.314131	-0.073593	0.028225	-0.018203	-0.062526	-0.079758	0.095290	-(
Has Diabetes	0.046390	0.236401	-0.045684	-0.000345	0.017857	0.014095	-0.064241	0.075653	-(

	Unique ID	Age	Annual Family Income ($)	Time spent watching videos/TV	Time spent playing indoor sports	Time spent playing outdoor sports	Total Time spent working in front of screen	Sleeping hours	
Has Gym Subscription	-0.016451	-0.026155	0.073399	0.047367	0.040591	-0.043578	0.051587	0.000637	(
Has OTT subscription	0.015877	-0.238392	-0.027831	0.003997	-0.003756	-0.013291	0.044221	-0.092801	(
Number of friends	0.001641	-0.215031	0.049422	0.053961	-0.001394	-0.024055	0.138441	-0.060692	(
Likes spicy food	-0.020153	-0.172288	-0.210395	0.025041	0.006230	-0.086600	-0.066966	-0.042507	-(
Likes desserts	-0.006227	-0.102045	0.070308	-0.028014	0.000630	0.046965	0.017949	-0.005585	-(
Wants to change career	-0.020872	0.088803	0.045767	0.016636	-0.003317	-0.087651	0.038882	-0.038345	(
Has debt	-0.033077	0.043587	-0.048003	-0.069840	0.024718	0.043871	-0.082182	-0.031085	-(
Has kids	0.009948	0.409205	-0.012442	-0.051618	-0.001149	0.093462	-0.155499	0.073832	-(
Drinks alcohol	0.015785	0.315849	0.076659	0.027832	0.030476	-0.068505	-0.022999	0.057836	(
Smoker	0.014561	0.149041	-0.004574	-0.045065	-0.003295	0.034192	-0.078283	0.010958	-(
Wear Specs	-0.016292	0.221421	0.089451	0.342571	0.048289	-0.176922	0.139726	-0.074425	(

26 rows × 26 columns

In [16]:
```
# Creating the heatmap

sns.heatmap(correlation_matrix)
```

Out[16]:
```
<Axes: >
```



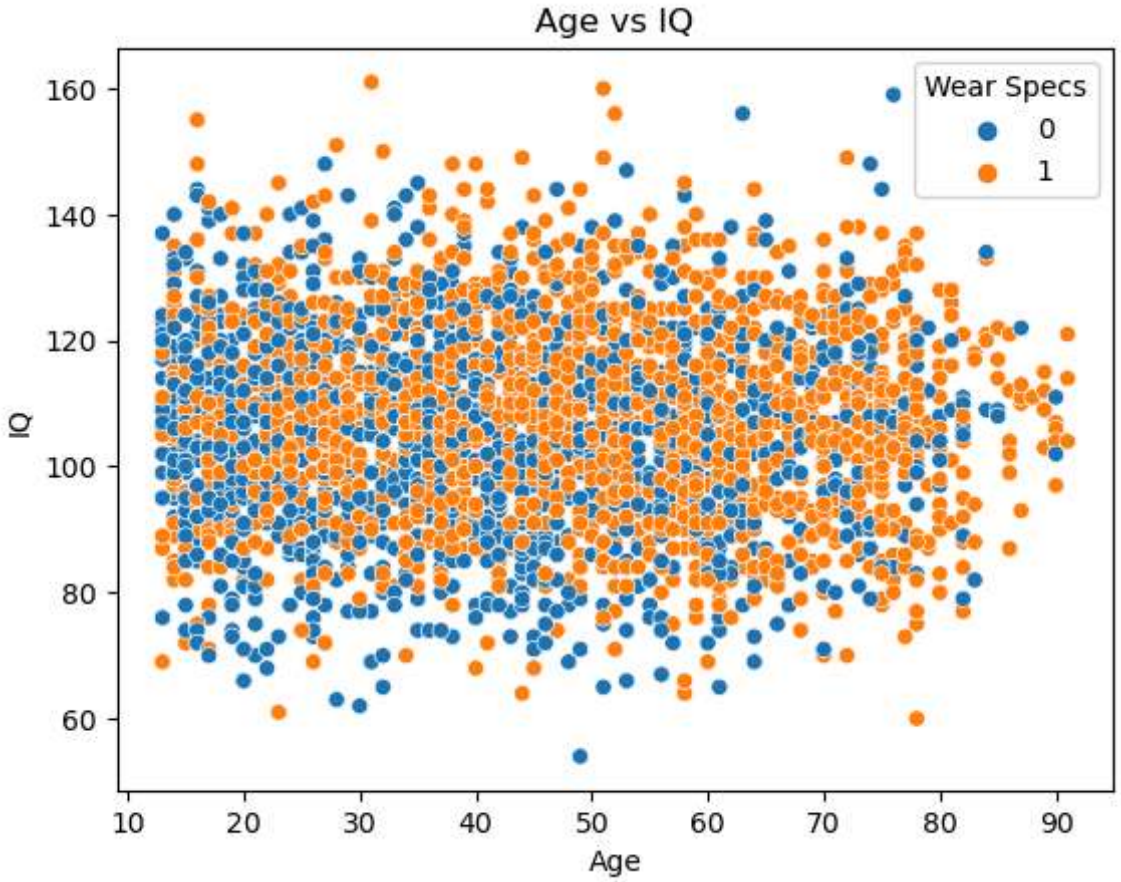
In [17]:
```
# Counting the occurrences of each Uniuque ID

track_counts = spectacles_data['Unique ID'].value_counts()
track_counts
```

Out[17]:
```
7319483    1
9298373    1
9947708    1
908689     1
3023855    1
          ..
1649371    1
6429100    1
6143237    1
3501157    1
9367449    1
Name: Unique ID, Length: 3220, dtype: int64
```

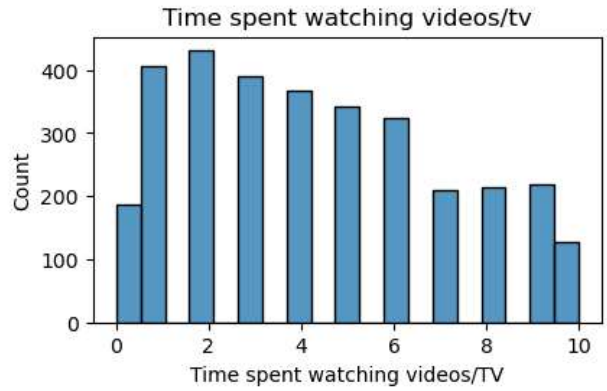
In [ ]:
```

```

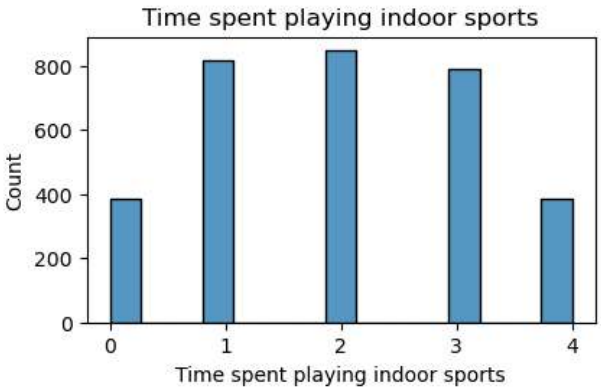
In [18]:
```
# Exploring relationships between variables

sns.scatterplot(x='Age', y='IQ', hue='Wear Specs', data=spectacles_data)
```

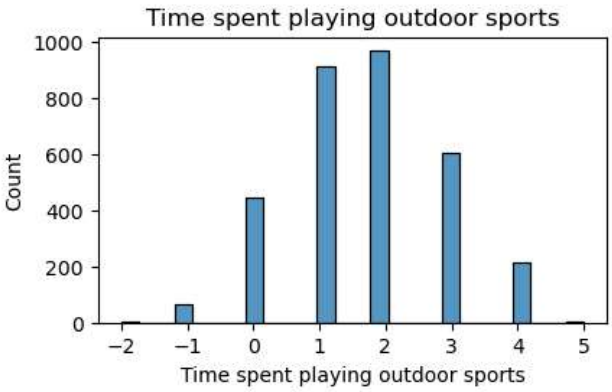
```
plt.title('Age vs IQ')
plt.xlabel('Age')
plt.ylabel('IQ')
plt.show()
```

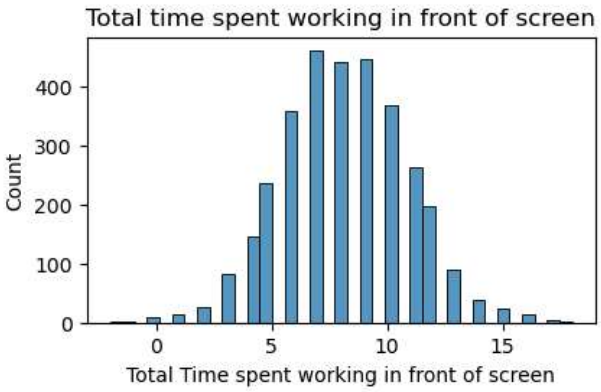


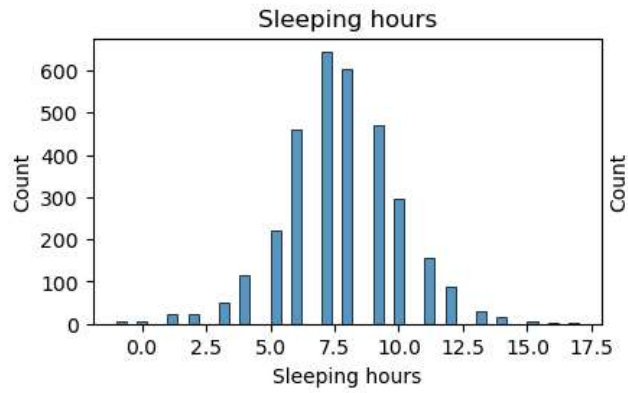
In [19]:
```
# Selecting the columns to be plotted

columns = ['Time spent watching videos/TV', 'Time spent playing indoor sports', 'Time
    'Whether parents have specs', 'Migrated within country', 'Migrated overseas', 'Mar

# Setting up the subplots using Seaborn's 'subplot' function
plt.figure(figsize=(10, 18))
plt.subplots_adjust(hspace=0.5)

for i, col in enumerate(columns, 1):
    plt.subplot(5, 2, i)
    sns.histplot(spectacles_data[col])
    plt.title(col.capitalize())

plt.show()
```

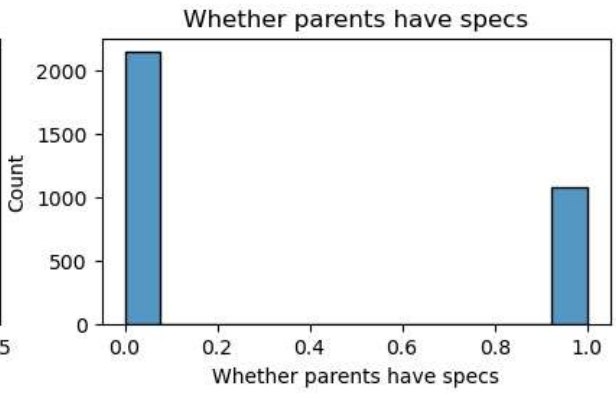

Time spent watching videos/tv
Count
Time spent watching videos/TV

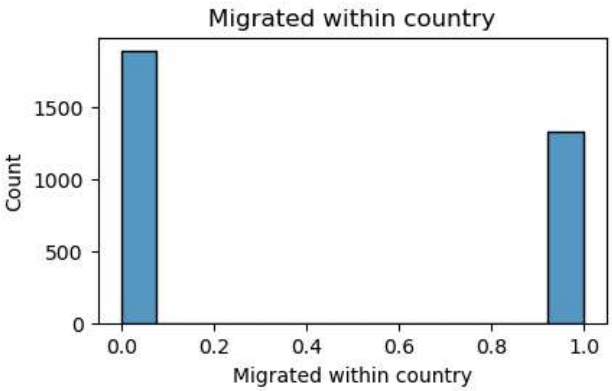

Time spent playing indoor sports
Count
Time spent playing indoor sports

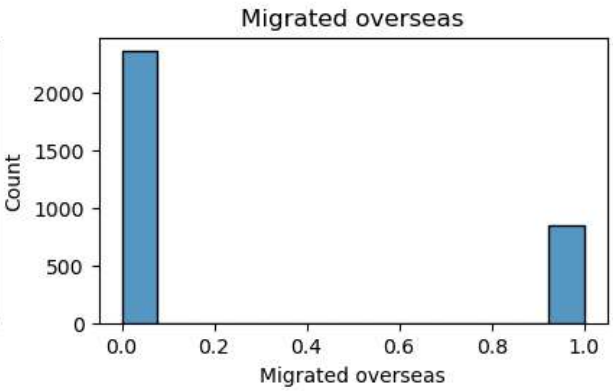

Time spent playing outdoor sports
Count
Time spent playing outdoor sports

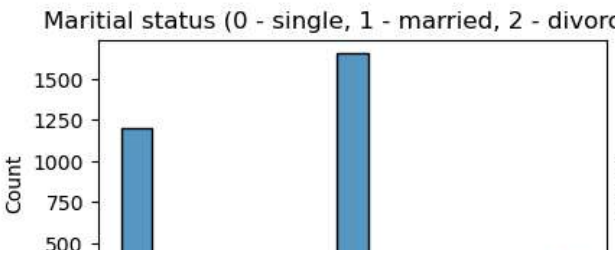

Total time spent working in front of screen
Count
Total Time spent working in front of screen


Sleeping hours
Count
Sleeping hours

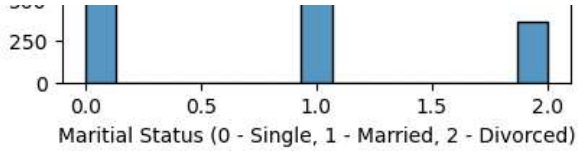

Whether parents have specs
Count
Whether parents have specs

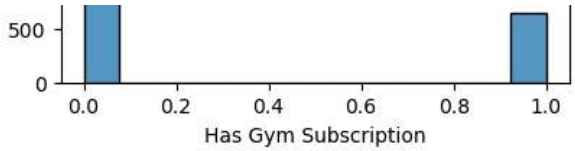

Migrated within country
Count
Migrated within country


Migrated overseas
Count
Migrated overseas


Maritial status (0 - single, 1 - married, 2 - divorced)
Count


Has gym subscription
Count


250
0
0.0
0.5
1.0
1.5
2.0
Maritial Status (0 - Single, 1 - Married, 2 - Divorced)


500
0
0.0
0.2
0.4
0.6
0.8
1.0
Has Gym Subscription

In [ ]: