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

In [2]:
```
data = pd.read_csv('./SLID.csv')
data
```

Out[2]:

	Unnamed: 0	wages	education	age	sex	language
0	1	10.56	15.0	40	Male	English
1	2	11.00	13.2	19	Male	English
2	3	NaN	16.0	49	Male	Other
3	4	17.76	14.0	46	Male	Other
4	5	NaN	8.0	71	Male	English
...	...	...	...	...	...	...
7420	7421	NaN	8.0	73	Male	Other
7421	7422	30.49	16.0	52	Male	Other
7422	7423	22.00	15.0	41	Male	Other
7423	7424	11.85	11.0	47	Female	English
7424	7425	23.00	14.0	30	Male	English

7425 rows × 6 columns

In [3]:
```
#Dimensions of dataset
data.shape
```

Out[3]:
```
(7425, 6)
```

In [4]:
```
data.columns
```

Out[4]:
```
Index(['Unnamed: 0', 'wages', 'education', 'age', 'sex', 'language'], dtype='ob
ject')
```

In [5]:
```
#Handling Null Values
data.isnull().sum()
```

Out[5]:
```
Unnamed: 0       0
wages         3278
education      249
age              0
sex              0
language       121
dtype: int64
```

In [6]:
```
data.dropna(inplace=True)
data.isnull().sum()
```

Out[6]:
```
Unnamed: 0    0
wages         0
education     0
age           0
sex           0
language      0
dtype: int64
```

In [7]:
```
#Dropping Unnecessary column
df = data.drop(['Unnamed: 0'], axis=1)
df.columns
```

Out[7]:
```
Index(['wages', 'education', 'age', 'sex', 'language'], dtype='object')
```

In [8]:
```
#After Data Pre-processing
df
```

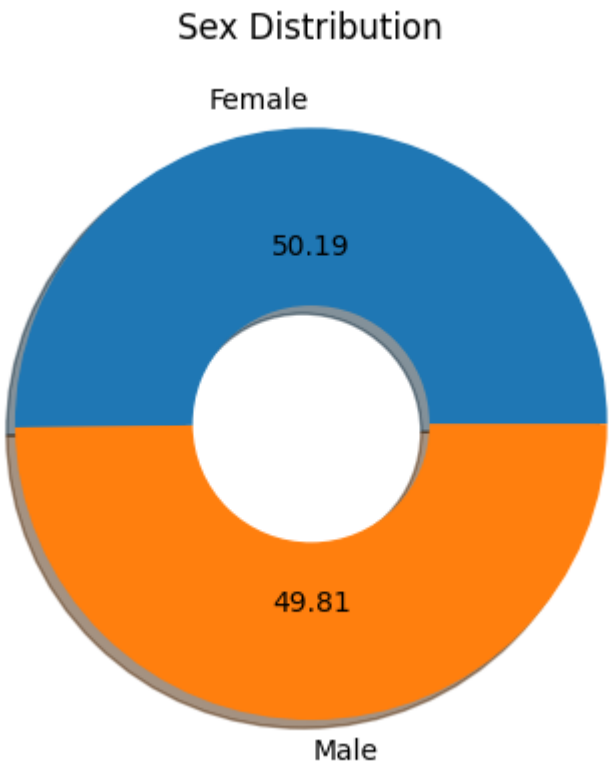
Out[8]:

	wages	education	age	sex	language
0	10.56	15.0	40	Male	English
1	11.00	13.2	19	Male	English
3	17.76	14.0	46	Male	Other
5	14.00	16.0	50	Female	English
8	8.20	15.0	31	Male	English
...	...	...	...	...	...
7417	6.80	13.1	20	Male	English
7421	30.49	16.0	52	Male	Other
7422	22.00	15.0	41	Male	Other
7423	11.85	11.0	47	Female	English
7424	23.00	14.0	30	Male	English

3987 rows × 5 columns

In [9]:
```
#Data Visualization
```

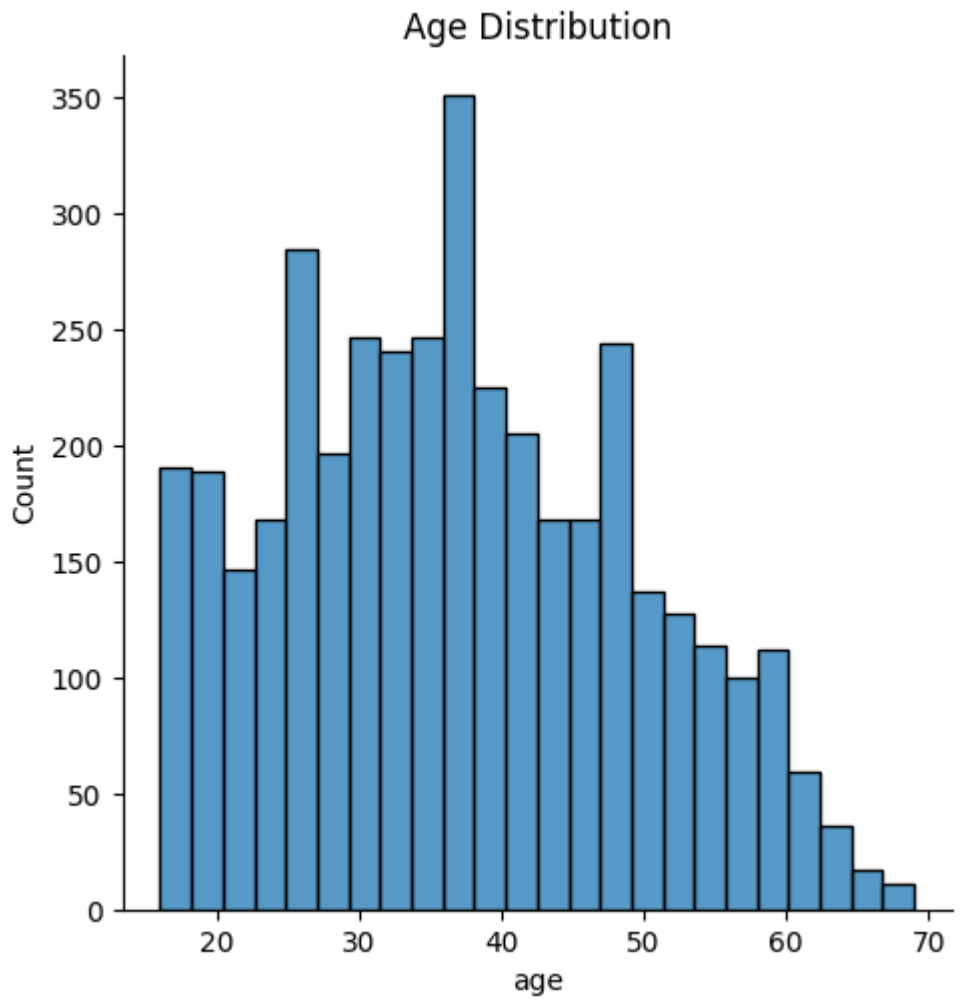
In [10]:
```
#Male vs Female
cnt = df['sex'].value_counts()
plt.pie(cnt, labels=cnt.index, shadow=True, wedgeprops=dict(width = 0.6), autopc
plt.title('Sex Distribution')
plt.show()
```



In [11]:
```
#Age Distribution
plt.figure(figsize=(9,7))
sns.displot(df['age'])
plt.title('Age Distribution')
```

Out[11]: Text(0.5, 1.0, 'Age Distribution')

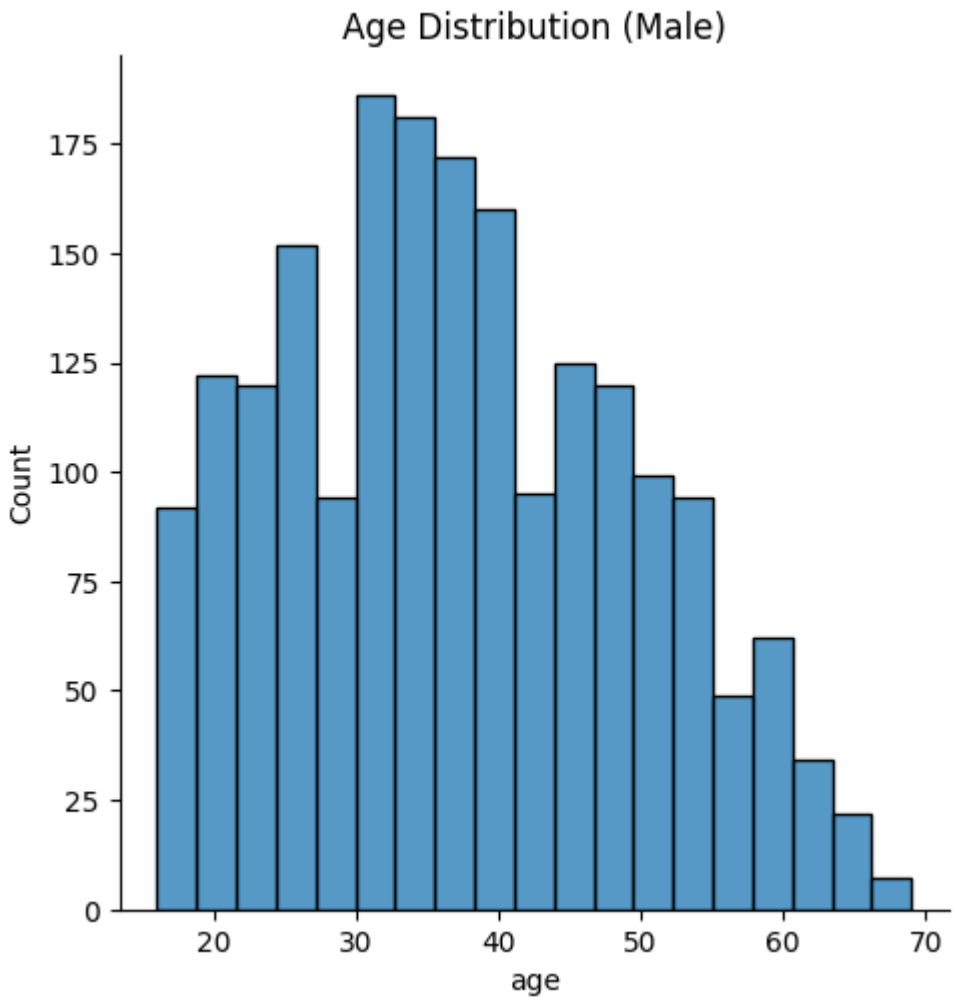
<Figure size 900x700 with 0 Axes>



In [12]:
```
plt.figure(figsize=(9,7))
dfm = df.loc[df['sex'] == 'Male']
sns.displot(dfm['age'])
plt.title('Age Distribution (Male)')
```

Out[12]: Text(0.5, 1.0, 'Age Distribution (Male)')

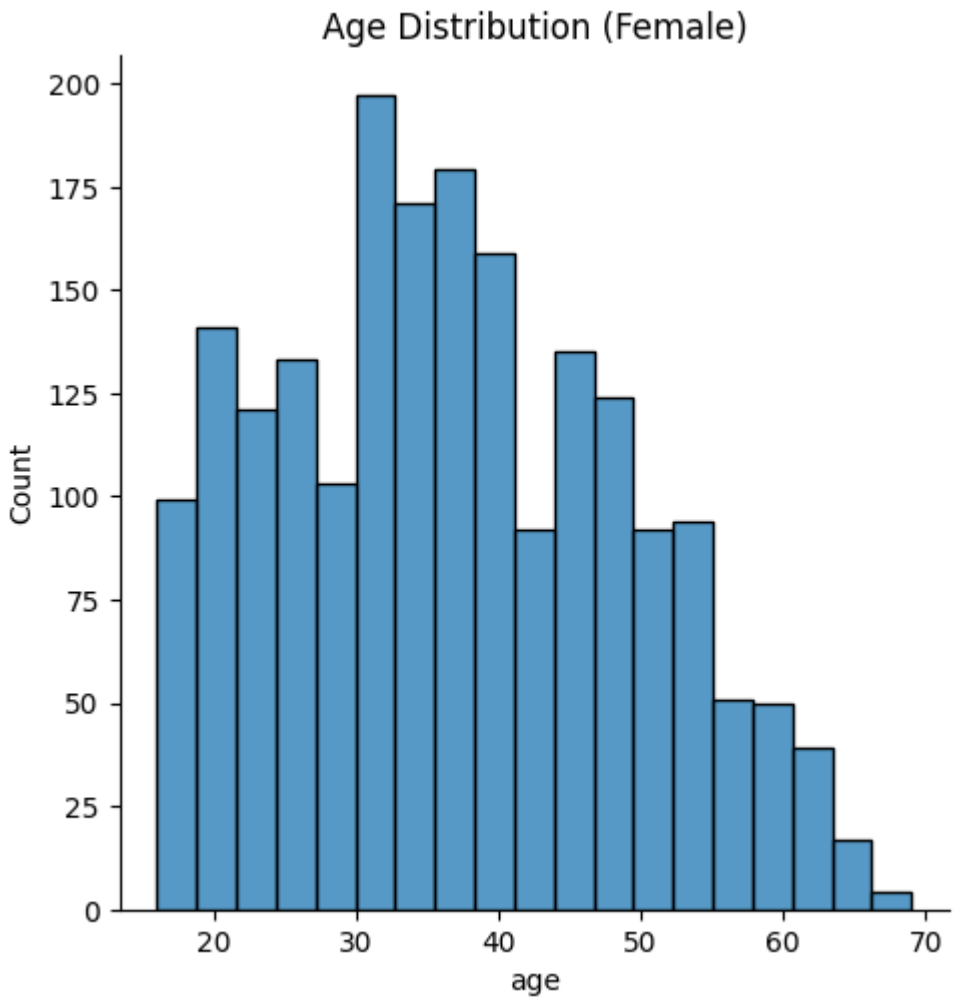
<Figure size 900x700 with 0 Axes>



In [13]:
```
plt.figure(figsize=(9,7))
dfm = df.loc[df['sex'] == 'Female']
sns.displot(dfm['age'])
plt.title('Age Distribution (Female)')
```

Out[13]: Text(0.5, 1.0, 'Age Distribution (Female)')

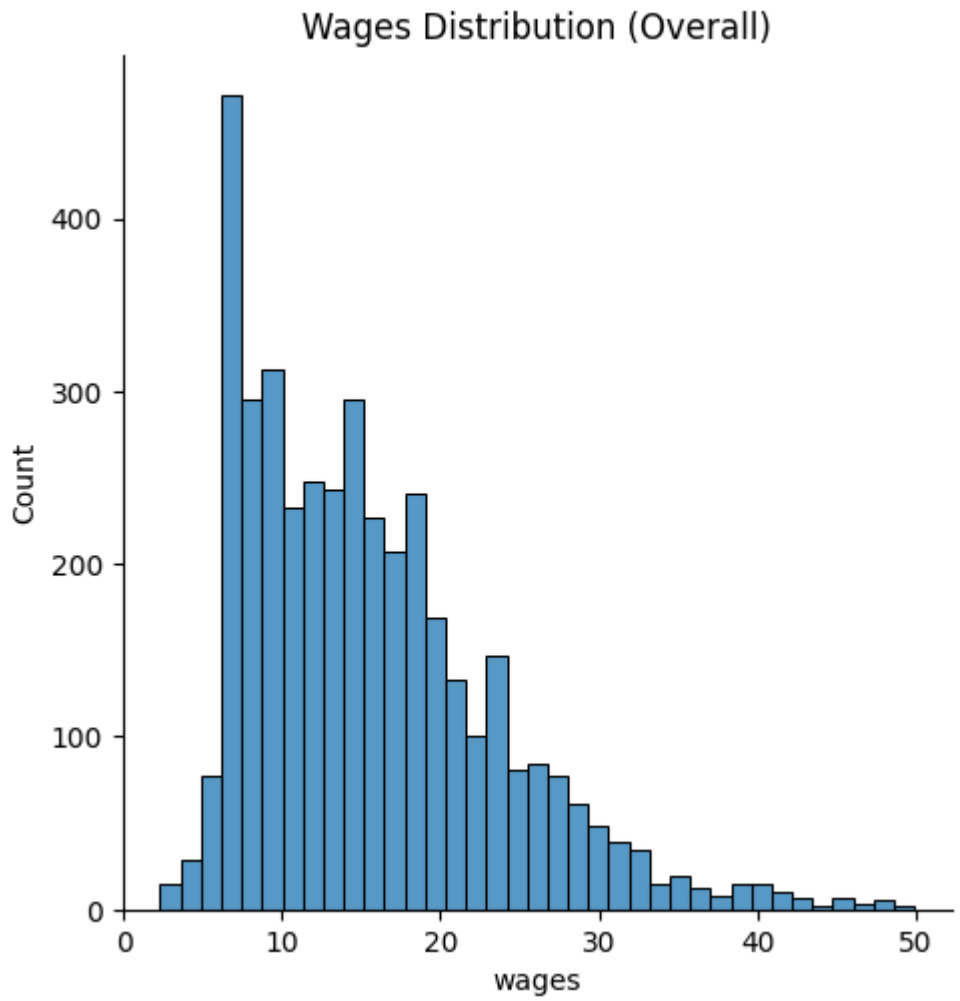
<Figure size 900x700 with 0 Axes>



In [14]:
```
#Wages Distribution
plt.figure(figsize=(9,7))
sns.displot(df['wages'])
plt.title('Wages Distribution (Overall)')
```

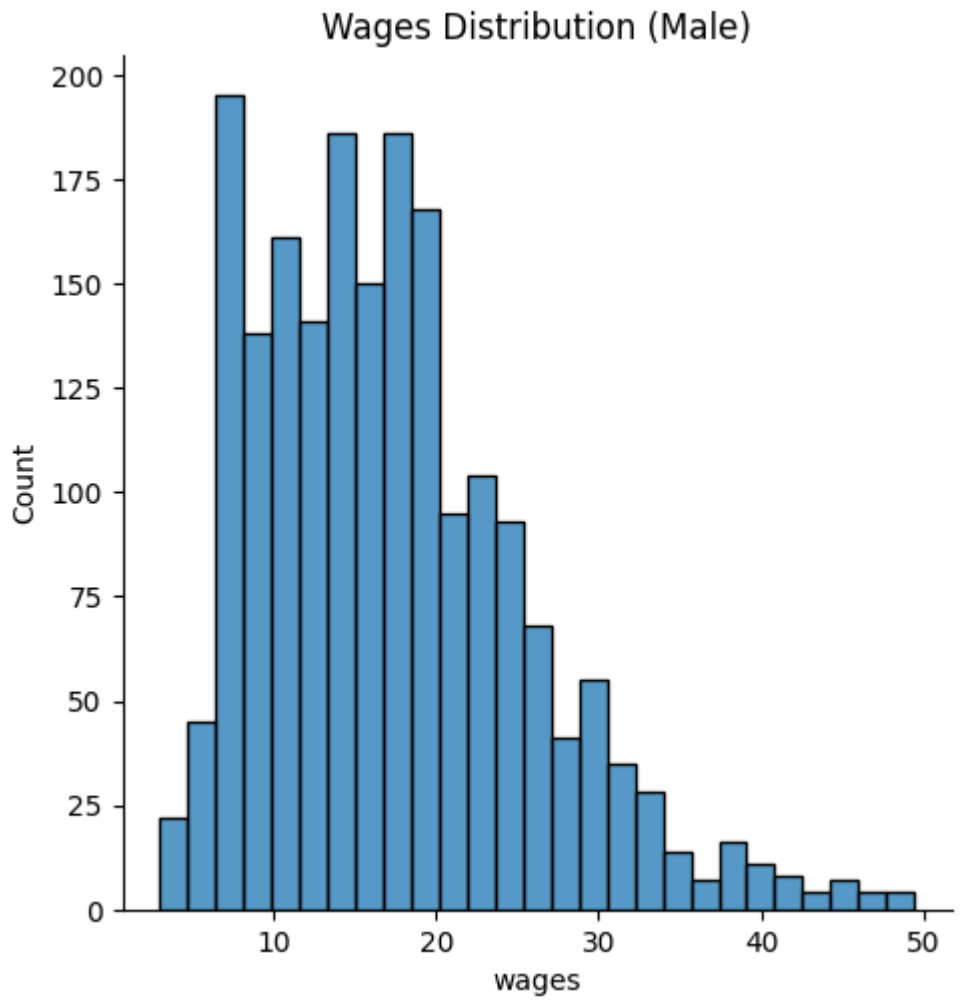
Out[14]: Text(0.5, 1.0, 'Wages Distribution (Overall)')

<Figure size 900x700 with 0 Axes>



In [15]:
```
#Wages Distrbution (Male)
plt.figure(figsize=(9,7))
dfm = df.loc[df['sex'] == 'Male']
sns.displot(dfm['wages'])
plt.title('Wages Distribution (Male)')
```

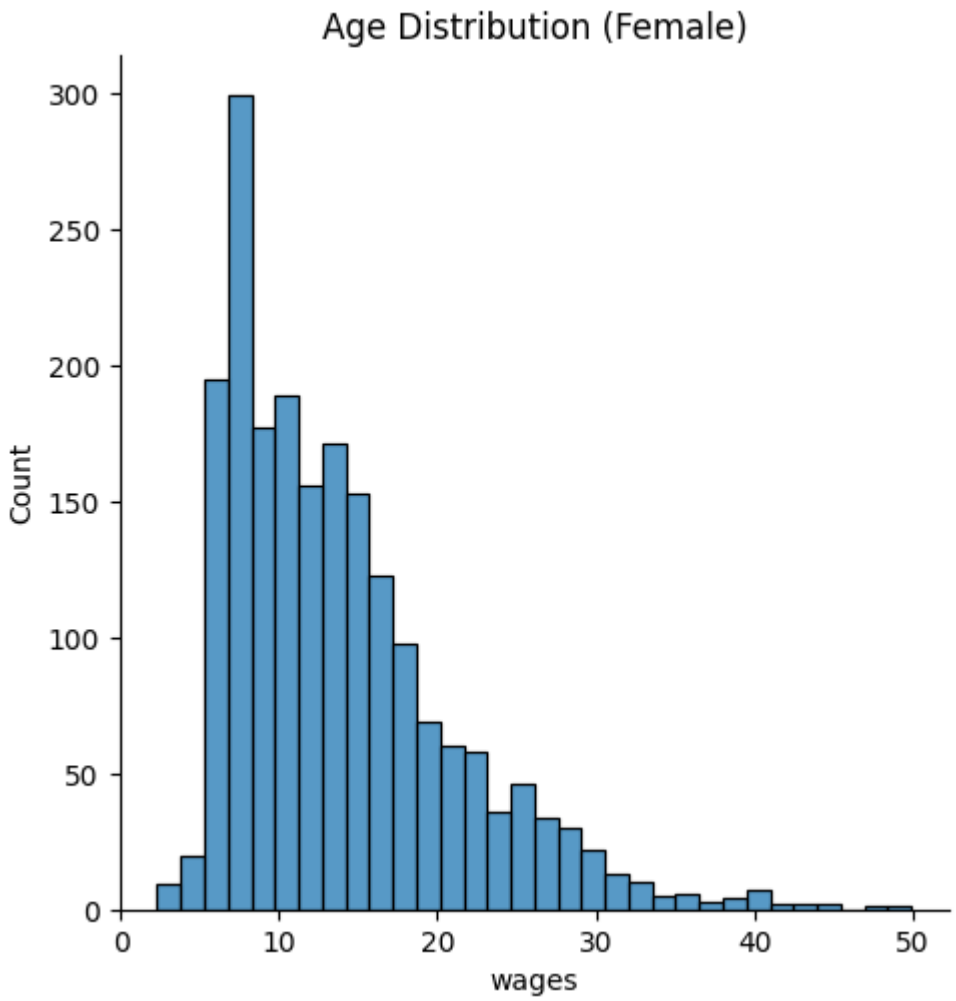
Out[15]:
```
Text(0.5, 1.0, 'Wages Distribution (Male)')
<Figure size 900x700 with 0 Axes>
```



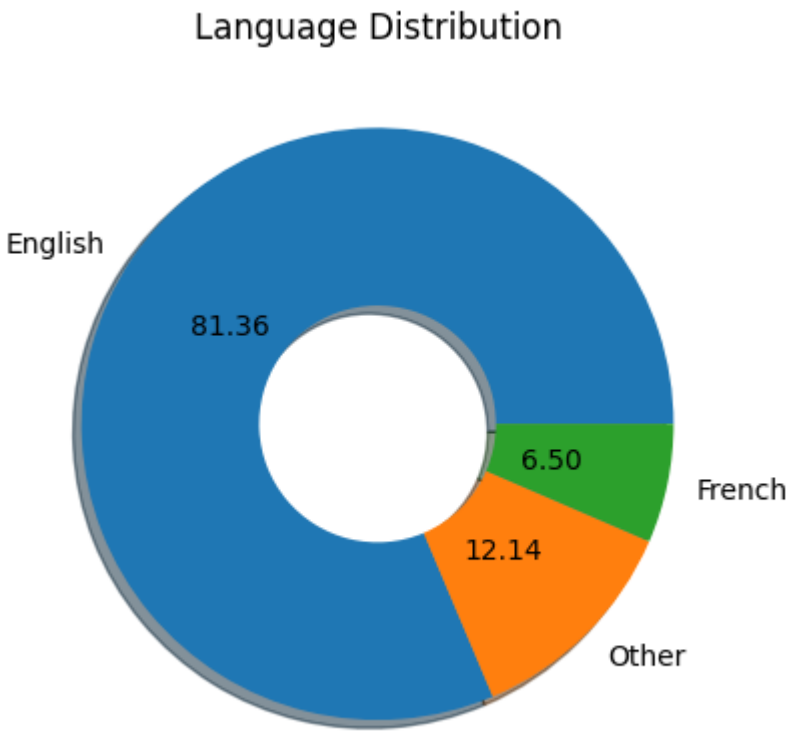
In [16]:
```
plt.figure(figsize=(9,7))
dfm = df.loc[df['sex'] == 'Female']
sns.displot(dfm['wages'])
plt.title('Age Distribution (Female)')
```

Out[16]:
```
Text(0.5, 1.0, 'Age Distribution (Female)')
```

```
<Figure size 900x700 with 0 Axes>
```

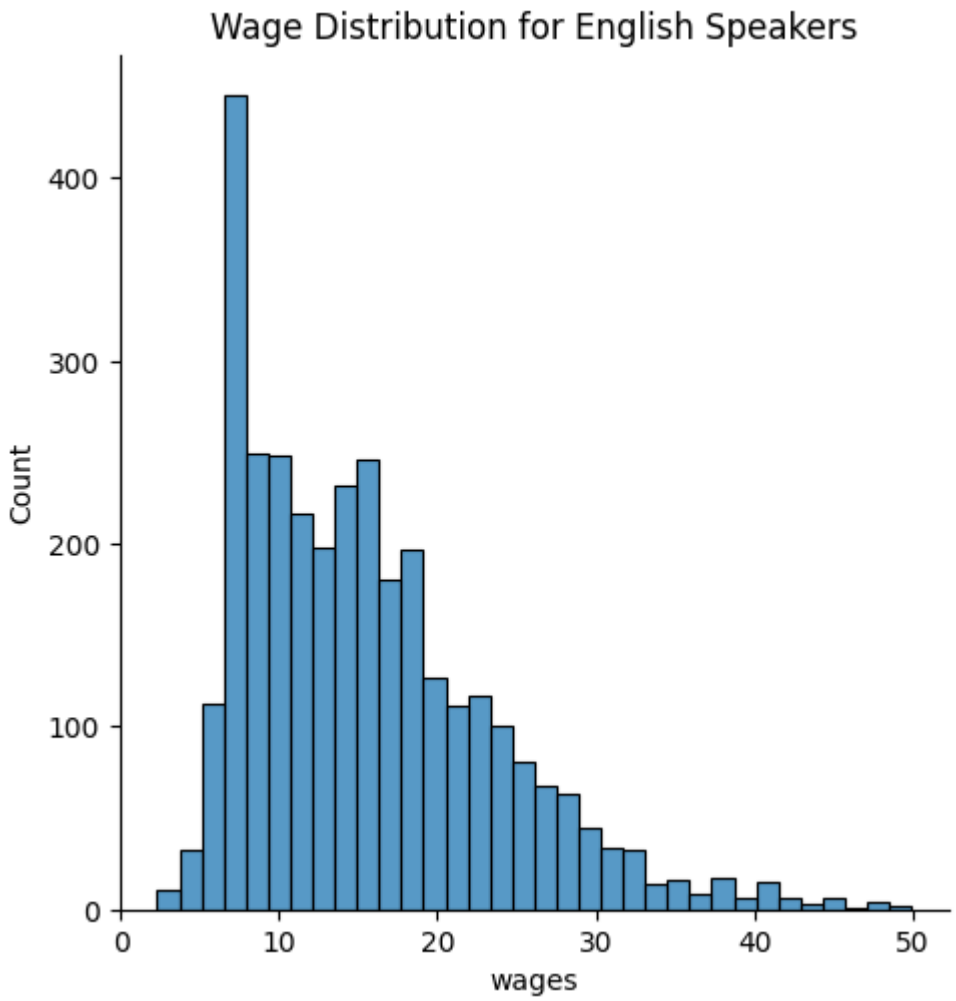


In [17]:
```
#Language Distribution
lan = df['language'].value_counts()
plt.pie(lan, labels=lan.index, shadow=True, wedgeprops=dict(width = 0.6), autopc
plt.title("Language Distribution")
plt.show()
```



In [18]:
```
#How Language Affects Wages
eng = df.loc[df['language'] == 'English']
plt.figure(figsize=(9,7))
sns.displot(eng['wages'])
plt.title("Wage Distribution for English Speakers")
```

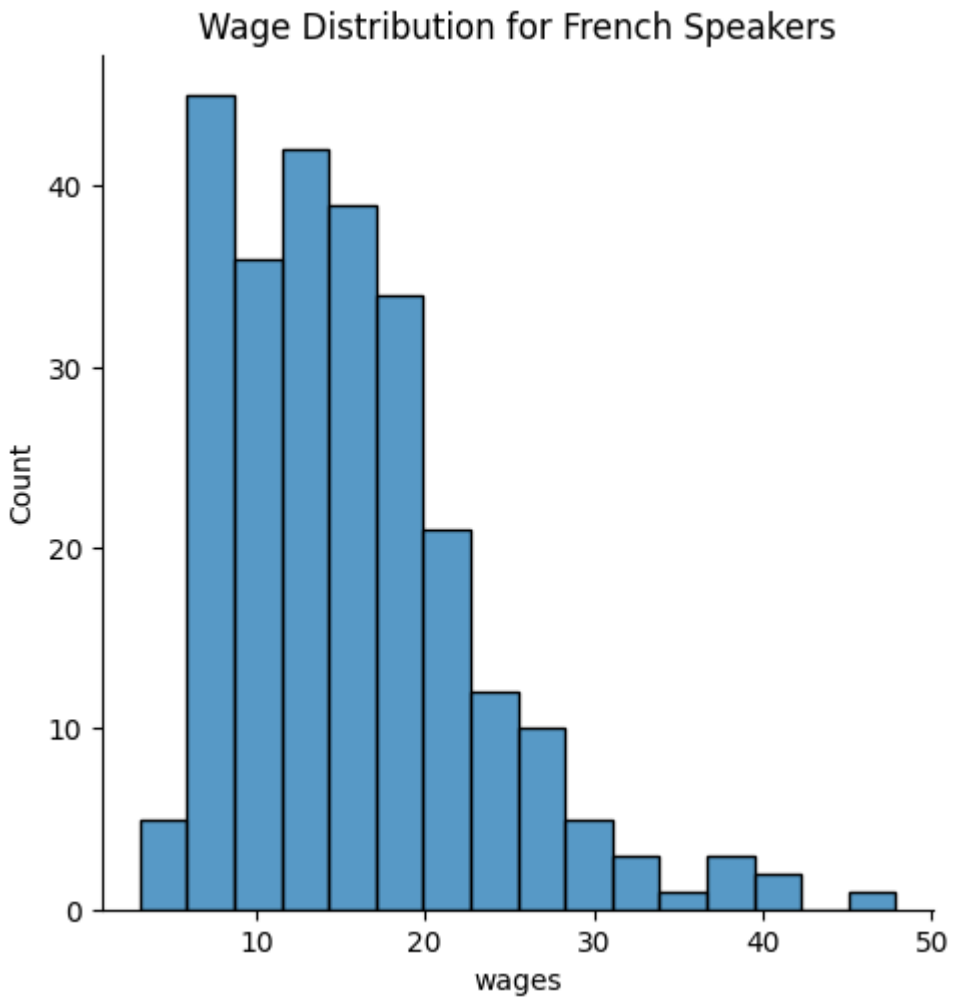
Out[18]:
```
Text(0.5, 1.0, 'Wage Distribution for English Speakers')
<Figure size 900x700 with 0 Axes>
```



In [19]:
```
frn = df.loc[df['language'] == 'French']
plt.figure(figsize=(9,7))
sns.displot(frn['wages'])
plt.title("Wage Distribution for French Speakers")
```

Out[19]: Text(0.5, 1.0, 'Wage Distribution for French Speakers')

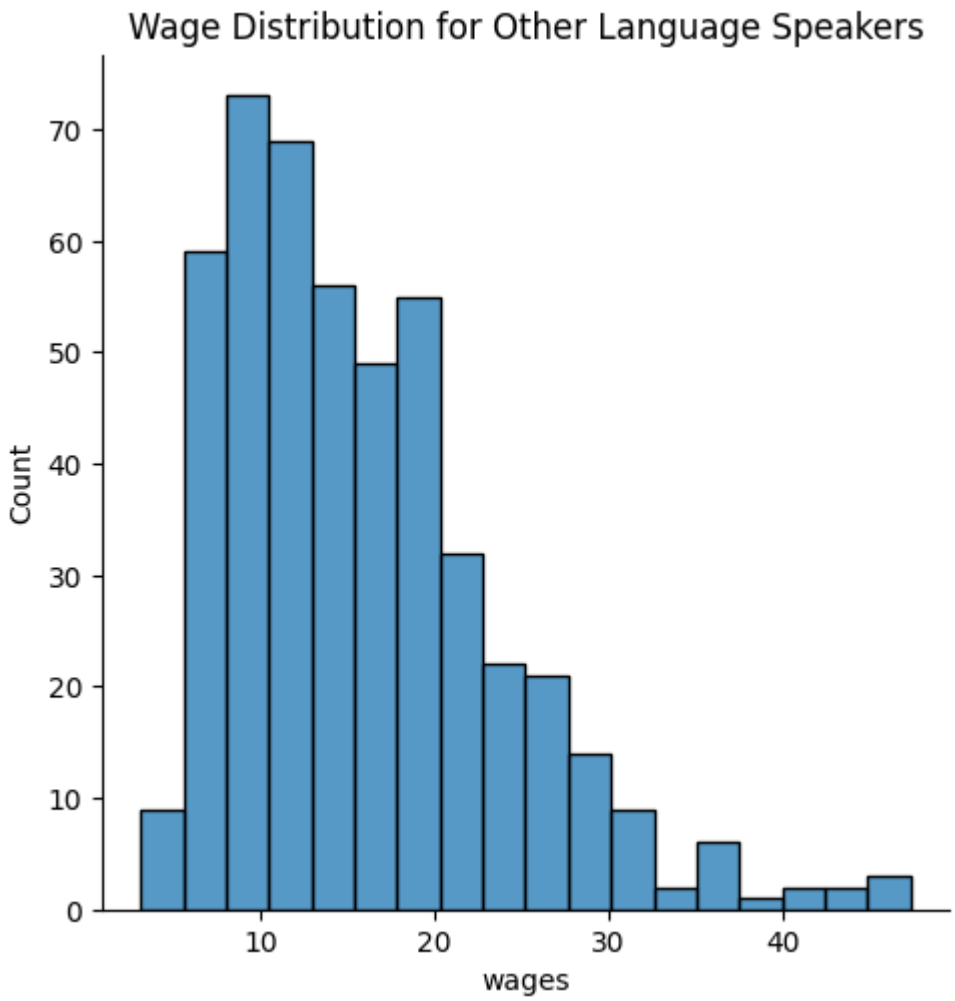
<Figure size 900x700 with 0 Axes>



In [20]:
```
oth = df.loc[df['language'] == 'Other']
plt.figure(figsize=(9,7))
sns.displot(oth['wages'])
plt.title("Wage Distribution for Other Language Speakers")
```

Out[20]: Text(0.5, 1.0, 'Wage Distribution for Other Language Speakers')

<Figure size 900x700 with 0 Axes>



In [32]:
```
#How are Wages and Education Related to Each other
x = np.array(df['education']).reshape(-1, 1)
y = np.array(df['wages'].values).reshape(-1, 1)
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
linReg = LinearRegression()
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.2, random_st
linReg.fit(xtrain, ytrain)
ypred = linReg.predict(xtest)
print(ypred)
```

```
[[14.57371053]
 [19.46194382]
 [14.98106331]
 [17.42517995]
 [12.94429944]
 [19.86929659]
 [18.81017938]
 [12.94429944]
 [17.99547383]
 [14.57371053]
 [12.94429944]
 [17.83253272]
 [12.53694667]
 [15.79576885]
 [13.75900499]
 [18.2398855 ]
 [12.94429944]
 [18.48429716]
 [17.83253272]
 [14.57371053]
 [14.57371053]
 [14.57371053]
 [14.98106331]
 [11.31488835]
 [14.57371053]
 [18.07694439]
 [21.09135491]
 [12.94429944]
 [19.46194382]
 [14.98106331]
 [17.01782718]
 [16.20312163]
 [20.60253158]
 [13.75900499]
 [12.94429944]
 [15.38841608]
 [14.57371053]
 [14.57371053]
 [17.09929773]
 [21.09135491]
 [18.48429716]
 [12.94429944]
 [14.57371053]
 [14.57371053]
 [18.64723827]
 [21.09135491]
 [16.20312163]
 [12.94429944]
 [13.75900499]
 [17.01782718]
 [15.38841608]
 [17.01782718]
 [14.57371053]
 [20.27664936]
 [14.57371053]
 [18.07694439]
 [14.57371053]
 [14.98106331]
 [14.16635776]
 [15.38841608]
```

[15.38841608]
[14.57371053]
[18.2398855]
[14.57371053]
[15.38841608]
[18.64723827]
[14.57371053]
[17.83253272]
[21.09135491]
[12.94429944]
[21.09135491]
[17.01782718]
[18.64723827]
[17.01782718]
[13.75900499]
[21.09135491]
[18.56576771]
[20.27664936]
[15.38841608]
[15.38841608]
[17.01782718]
[14.57371053]
[12.94429944]
[12.94429944]
[13.75900499]
[18.64723827]
[13.02577   ]
[13.43312277]
[14.16635776]
[15.38841608]
[21.09135491]
[17.83253272]
[12.12959389]
[14.98106331]
[18.2398855]
[13.35165221]
[13.75900499]
[14.8181222]
[11.31488835]
[14.57371053]
[14.57371053]
[17.83253272]
[14.16635776]
[15.46988664]
[15.63282775]
[17.83253272]
[16.20312163]
[14.98106331]
[12.94429944]
[17.42517995]
[14.57371053]
[14.57371053]
[18.07694439]
[14.65518109]
[17.01782718]
[13.75900499]
[18.64723827]
[16.20312163]
[19.78782603]
[16.20312163]

[14.57371053]
[16.20312163]
[21.09135491]
[17.83253272]
[12.12959389]
[17.83253272]
[15.87723941]
[14.98106331]
[12.94429944]
[13.75900499]
[17.01782718]
[17.83253272]
[10.5001828]
[19.46194382]
[13.27018166]
[13.35165221]
[15.55135719]
[12.12959389]
[14.57371053]
[14.98106331]
[14.98106331]
[17.83253272]
[14.57371053]
[16.44753329]
[12.12959389]
[15.06253386]
[14.57371053]
[15.55135719]
[12.94429944]
[16.44753329]
[15.38841608]
[14.57371053]
[14.57371053]
[14.57371053]
[14.98106331]
[14.41076943]
[11.72224112]
[17.01782718]
[17.01782718]
[12.12959389]
[17.01782718]
[16.6104744]
[14.57371053]
[14.98106331]
[14.57371053]
[17.91400328]
[16.20312163]
[12.94429944]
[15.38841608]
[15.79576885]
[15.79576885]
[15.7142983]
[13.75900499]
[13.75900499]
[20.52106103]
[14.57371053]
[14.57371053]
[19.46194382]
[14.98106331]
[15.55135719]

[18.64723827]
[12.94429944]
[12.94429944]
[18.64723827]
[18.64723827]
[13.75900499]
[16.20312163]
[21.09135491]
[14.57371053]
[14.57371053]
[18.4028266]
[13.75900499]
[18.64723827]
[16.20312163]
[15.79576885]
[14.98106331]
[19.05459104]
[16.20312163]
[12.12959389]
[17.83253272]
[16.69194496]
[19.46194382]
[13.75900499]
[12.94429944]
[15.79576885]
[18.64723827]
[4.79724397]
[15.22547497]
[17.01782718]
[13.75900499]
[11.31488835]
[18.2398855]
[16.20312163]
[17.83253272]
[17.83253272]
[12.94429944]
[12.94429944]
[20.27664936]
[17.01782718]
[14.16635776]
[14.57371053]
[14.57371053]
[14.57371053]
[14.57371053]
[14.98106331]
[14.98106331]
[13.75900499]
[16.44753329]
[14.57371053]
[21.09135491]
[15.38841608]
[17.83253272]
[14.98106331]
[13.75900499]
[12.94429944]
[9.68547725]
[21.09135491]
[14.57371053]
[19.46194382]
[13.35165221]

[17.83253272]
[12.94429944]
[18.07694439]
[14.57371053]
[18.97312049]
[9.68547725]
[15.79576885]
[15.38841608]
[21.09135491]
[13.75900499]
[18.64723827]
[15.79576885]
[19.86929659]
[14.57371053]
[11.31488835]
[18.64723827]
[17.18076828]
[17.01782718]
[14.57371053]
[18.2398855]
[11.31488835]
[14.57371053]
[16.20312163]
[17.83253272]
[15.38841608]
[12.94429944]
[17.42517995]
[17.42517995]
[12.12959389]
[18.64723827]
[17.42517995]
[17.83253272]
[15.38841608]
[17.01782718]
[13.35165221]
[12.94429944]
[16.20312163]
[13.75900499]
[12.12959389]
[12.12959389]
[19.46194382]
[14.57371053]
[17.01782718]
[14.57371053]
[15.38841608]
[18.15841494]
[12.94429944]
[15.38841608]
[17.42517995]
[16.20312163]
[20.27664936]
[14.57371053]
[17.09929773]
[12.94429944]
[15.79576885]
[13.75900499]
[16.20312163]
[17.83253272]
[14.57371053]
[14.65518109]

[14.57371053]
[17.01782718]
[15.79576885]
[14.57371053]
[16.28459218]
[12.94429944]
[5.61194952]
[12.12959389]
[18.07694439]
[17.01782718]
[14.98106331]
[17.01782718]
[14.57371053]
[16.20312163]
[14.98106331]
[18.89164993]
[16.6104744]
[18.81017938]
[16.20312163]
[18.64723827]
[17.01782718]
[21.09135491]
[19.46194382]
[18.64723827]
[18.64723827]
[15.38841608]
[12.94429944]
[15.63282775]
[19.70635548]
[17.83253272]
[19.46194382]
[14.57371053]
[14.98106331]
[15.38841608]
[12.94429944]
[19.46194382]
[17.01782718]
[14.57371053]
[12.94429944]
[12.94429944]
[16.20312163]
[17.01782718]
[13.75900499]
[12.94429944]
[16.20312163]
[14.57371053]
[17.83253272]
[19.70635548]
[16.04018052]
[16.20312163]
[14.57371053]
[15.79576885]
[18.64723827]
[17.83253272]
[18.89164993]
[14.89959275]
[19.46194382]
[15.79576885]
[16.20312163]
[15.79576885]

[15.55135719]
[17.01782718]
[15.38841608]
[13.75900499]
[14.98106331]
[14.57371053]
[16.20312163]
[15.38841608]
[15.14400442]
[14.57371053]
[14.57371053]
[16.6104744]
[14.57371053]
[14.98106331]
[16.20312163]
[18.64723827]
[11.31488835]
[16.20312163]
[14.57371053]
[15.46988664]
[18.64723827]
[20.35811992]
[15.38841608]
[17.26223884]
[15.38841608]
[14.57371053]
[18.72870882]
[14.57371053]
[17.83253272]
[14.24782832]
[12.94429944]
[13.75900499]
[13.75900499]
[15.38841608]
[19.05459104]
[15.38841608]
[20.27664936]
[17.01782718]
[14.57371053]
[14.98106331]
[19.46194382]
[21.09135491]
[14.57371053]
[19.46194382]
[14.57371053]
[18.07694439]
[21.09135491]
[12.94429944]
[14.98106331]
[17.01782718]
[12.94429944]
[17.83253272]
[16.20312163]
[15.38841608]
[12.12959389]
[14.57371053]
[14.57371053]
[11.31488835]
[16.20312163]
[18.64723827]

[14.16635776]
[17.83253272]
[12.94429944]
[14.98106331]
[14.73665164]
[17.01782718]
[14.57371053]
[11.31488835]
[19.54341437]
[14.98106331]
[17.01782718]
[15.22547497]
[17.01782718]
[14.65518109]
[5.61194952]
[18.64723827]
[11.31488835]
[14.57371053]
[14.57371053]
[21.09135491]
[14.57371053]
[14.57371053]
[18.48429716]
[16.36606274]
[14.57371053]
[11.31488835]
[14.98106331]
[15.38841608]
[14.98106331]
[16.6104744]
[15.79576885]
[13.35165221]
[13.75900499]
[14.57371053]
[19.05459104]
[14.65518109]
[14.98106331]
[13.35165221]
[16.20312163]
[8.46341893]
[20.27664936]
[14.57371053]
[17.01782718]
[11.47782946]
[14.98106331]
[14.57371053]
[14.57371053]
[14.24782832]
[18.64723827]
[15.38841608]
[14.57371053]
[17.83253272]
[12.94429944]
[15.79576885]
[15.55135719]
[16.20312163]
[15.38841608]
[18.72870882]
[14.57371053]
[15.63282775]

[17.01782718]
[15.79576885]
[12.94429944]
[17.01782718]
[14.57371053]
[19.78782603]
[10.5001828]
[21.09135491]
[15.38841608]
[14.65518109]
[15.38841608]
[20.27664936]
[14.98106331]
[16.28459218]
[15.38841608]
[16.20312163]
[14.57371053]
[14.57371053]
[15.79576885]
[18.2398855]
[15.38841608]
[13.75900499]
[16.20312163]
[14.57371053]
[14.57371053]
[12.94429944]
[14.65518109]
[16.52900385]
[16.28459218]
[20.27664936]
[14.65518109]
[15.79576885]
[16.20312163]
[16.20312163]
[14.57371053]
[17.01782718]
[11.31488835]
[13.75900499]
[17.83253272]
[17.83253272]
[12.94429944]
[14.8181222]
[16.44753329]
[12.12959389]
[20.27664936]
[12.94429944]
[15.38841608]
[15.38841608]
[14.57371053]
[17.01782718]
[18.64723827]
[12.94429944]
[17.01782718]
[12.94429944]
[18.64723827]
[13.9219461]
[16.20312163]
[16.20312163]
[16.85488607]
[14.98106331]

[11.31488835]
[16.20312163]
[14.57371053]
[17.83253272]
[12.94429944]
[21.09135491]
[12.12959389]
[14.65518109]
[14.98106331]
[16.20312163]
[14.57371053]
[15.38841608]
[12.53694667]
[13.75900499]
[14.98106331]
[16.93635662]
[16.85488607]
[16.04018052]
[17.01782718]
[16.20312163]
[16.36606274]
[12.21106445]
[12.12959389]
[12.94429944]
[17.66959161]
[14.57371053]
[12.94429944]
[19.46194382]
[14.57371053]
[14.57371053]
[14.57371053]
[14.57371053]
[16.69194496]
[14.57371053]
[15.38841608]
[18.56576771]
[12.12959389]
[15.38841608]
[17.01782718]
[15.87723941]
[18.07694439]
[15.38841608]
[15.38841608]
[17.42517995]
[14.57371053]
[14.57371053]
[14.98106331]
[13.75900499]
[16.20312163]
[18.72870882]
[18.97312049]
[17.83253272]
[12.53694667]
[15.38841608]
[21.09135491]
[14.57371053]
[14.98106331]
[14.57371053]
[9.68547725]
[16.20312163]

[15.38841608]
[14.57371053]
[17.42517995]
[15.38841608]
[15.38841608]
[12.12959389]
[14.57371053]
[20.27664936]
[17.99547383]
[13.75900499]
[15.38841608]
[13.75900499]
[15.38841608]
[14.57371053]
[13.43312277]
[15.7142983]
[17.83253272]
[14.57371053]
[19.05459104]
[11.31488835]
[20.27664936]
[13.75900499]
[10.5001828]
[15.38841608]
[17.83253272]
[14.57371053]
[17.01782718]
[12.12959389]
[17.01782718]
[15.22547497]
[14.98106331]
[19.46194382]
[8.87077171]
[18.64723827]
[21.09135491]
[15.79576885]
[15.14400442]
[14.57371053]
[15.38841608]
[17.83253272]
[15.38841608]
[13.9219461]
[12.12959389]
[13.75900499]
[14.57371053]
[15.38841608]
[18.89164993]
[17.01782718]
[15.38841608]
[20.27664936]
[12.94429944]
[18.64723827]
[14.57371053]
[14.65518109]
[15.79576885]
[17.83253272]
[14.57371053]
[14.57371053]
[19.46194382]
[14.57371053]

[18.2398855]
[14.57371053]
[12.94429944]
[14.98106331]
[18.89164993]
[21.09135491]
[15.38841608]
[15.06253386]
[12.94429944]
[15.38841608]
[16.20312163]
[17.83253272]
[12.94429944]
[16.20312163]
[16.20312163]
[20.27664936]
[16.44753329]
[17.01782718]
[14.57371053]
[16.6104744]
[16.20312163]
[17.26223884]
[14.73665164]
[20.27664936]
[16.20312163]
[12.12959389]
[13.35165221]
[13.75900499]
[14.57371053]
[14.98106331]
[19.46194382]
[13.35165221]
[12.94429944]
[14.57371053]
[18.64723827]
[13.35165221]
[15.06253386]
[14.57371053]
[16.20312163]
[19.1360616]
[10.5001828]
[14.57371053]
[16.77341551]
[12.53694667]
[19.46194382]
[17.01782718]
[14.57371053]
[7.24136061]
[16.20312163]
[17.01782718]
[13.75900499]
[21.09135491]
[19.46194382]
[12.12959389]
[17.01782718]
[19.46194382]
[16.20312163]
[16.6104744]
[17.01782718]
[17.01782718]

[16.20312163]
[14.57371053]
[16.20312163]
[14.98106331]
[15.38841608]
[16.20312163]
[14.57371053]
[14.57371053]
[15.38841608]
[12.12959389]
[14.57371053]
[11.31488835]
[16.20312163]
[15.46988664]
[18.64723827]
[15.38841608]
[13.35165221]
[15.38841608]
[13.75900499]
[19.46194382]
[12.12959389]
[16.28459218]
[17.01782718]
[14.57371053]
[14.57371053]
[15.38841608]
[11.31488835]
[14.16635776]
[16.20312163]
[13.75900499]
[14.57371053]
[14.73665164]
[12.12959389]
[14.57371053]
[14.73665164]
[13.75900499]
[15.38841608]
[18.64723827]
[15.38841608]
[17.01782718]
[18.07694439]
[16.20312163]
[21.09135491]
[14.98106331]
[14.57371053]
[17.01782718]
[14.57371053]
[14.57371053]
[13.75900499]
[14.98106331]
[17.83253272]
[12.94429944]
[14.98106331]
[17.09929773]
[13.9219461]
[15.38841608]
[15.38841608]
[14.57371053]
[21.09135491]
[21.09135491]

```
[15.79576885]
[16.6104744 ]
[15.14400442]
[14.57371053]
[13.75900499]
[14.98106331]
[12.94429944]
[16.20312163]
[19.05459104]
[20.27664936]
[18.64723827]
[17.01782718]
[14.57371053]
[15.38841608]
[16.20312163]
[14.57371053]
[13.75900499]
[13.84047554]]
```

In [33]:
```
plt.scatter(xtrain, ytrain)
plt.plot(xtrain, linReg.predict(xtrain), color = 'Black')
plt.show()
```