In [18]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [19]:

```
df=pd.read_csv(r"C:\Users\chila\Downloads\USA_Housing.csv")
df
```

Out[19]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michae 674\nLau
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johr Suite Kath
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	912 Stravenue\nD V
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnet
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raym
...	...	...	...	...	...	...	
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	USNS Willi AP 3
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 8489\nAPO
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 Tra Suite 076\nJ
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	USS Wallace
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 Geo Apt. 509\n

5000 rows × 7 columns

In [20]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Avg. Area Income             5000 non-null   float64
 1   Avg. Area House Age          5000 non-null   float64
 2   Avg. Area Number of Rooms    5000 non-null   float64
 3   Avg. Area Number of Bedrooms  5000 non-null   float64
 4   Area Population              5000 non-null   float64
 5   Price                        5000 non-null   float64
 6   Address                      5000 non-null   object 
dtypes: float64(6), object(1)
memory usage: 273.6+ KB
```

In [21]:

```
df.describe()
```

Out[21]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5.000000e+03
mean	68583.108984	5.977222	6.987792	3.981330	36163.516039	1.232073e+06
std	10657.991214	0.991456	1.005833	1.234137	9925.650114	3.531176e+05
min	17796.631190	2.644304	3.236194	2.000000	172.610686	1.593866e+04
25%	61480.562388	5.322283	6.299250	3.140000	29403.928702	9.975771e+05
50%	68804.286404	5.970429	7.002902	4.050000	36199.406689	1.232669e+06
75%	75783.338666	6.650808	7.665871	4.490000	42861.290769	1.471210e+06
max	107701.748378	9.519088	10.759588	6.500000	69621.713378	2.469066e+06

In [22]:

```
df.isna().any()
```

Out[22]:

```
Avg. Area Income               False
Avg. Area House Age            False
Avg. Area Number of Rooms      False
Avg. Area Number of Bedrooms   False
Area Population                False
Price                          False
Address                        False
dtype: bool
```

In [23]:

```
df.isnull().sum()
```

Out[23]:

```
Avg. Area Income                0
Avg. Area House Age             0
Avg. Area Number of Rooms       0
Avg. Area Number of Bedrooms    0
Area Population                 0
Price                           0
Address                         0
dtype: int64
```

In [24]:

```
df.head()
```

Out[24]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	A
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Fe 674\nLaurab
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnsor Suite 079 Kathleer
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Eli Stravenue\nDani WI 0
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nF
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymonc AE

In [25]:

```
df.tail()
```

Out[25]:

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Add
4995	60567.944140	7.830362	6.137356	3.46	22837.361035	1.060194e+06	U Williams\n AP 30153-7
4996	78491.275435	6.999135	6.576763	4.02	25616.115489	1.482618e+06	PSC 9258, 8489\nAPO 42991-3
4997	63390.686886	7.250591	4.805081	2.13	33266.145490	1.030730e+06	4215 T Garden S 076\nJoshual VA
4998	68001.331235	5.534388	7.130144	5.44	42625.620156	1.198657e+06	Wallace\n AE 73
4999	65510.581804	5.992305	6.792336	4.07	46501.283803	1.298950e+06	37778 Ge Ridges 509\nEast H NV

In [26]:

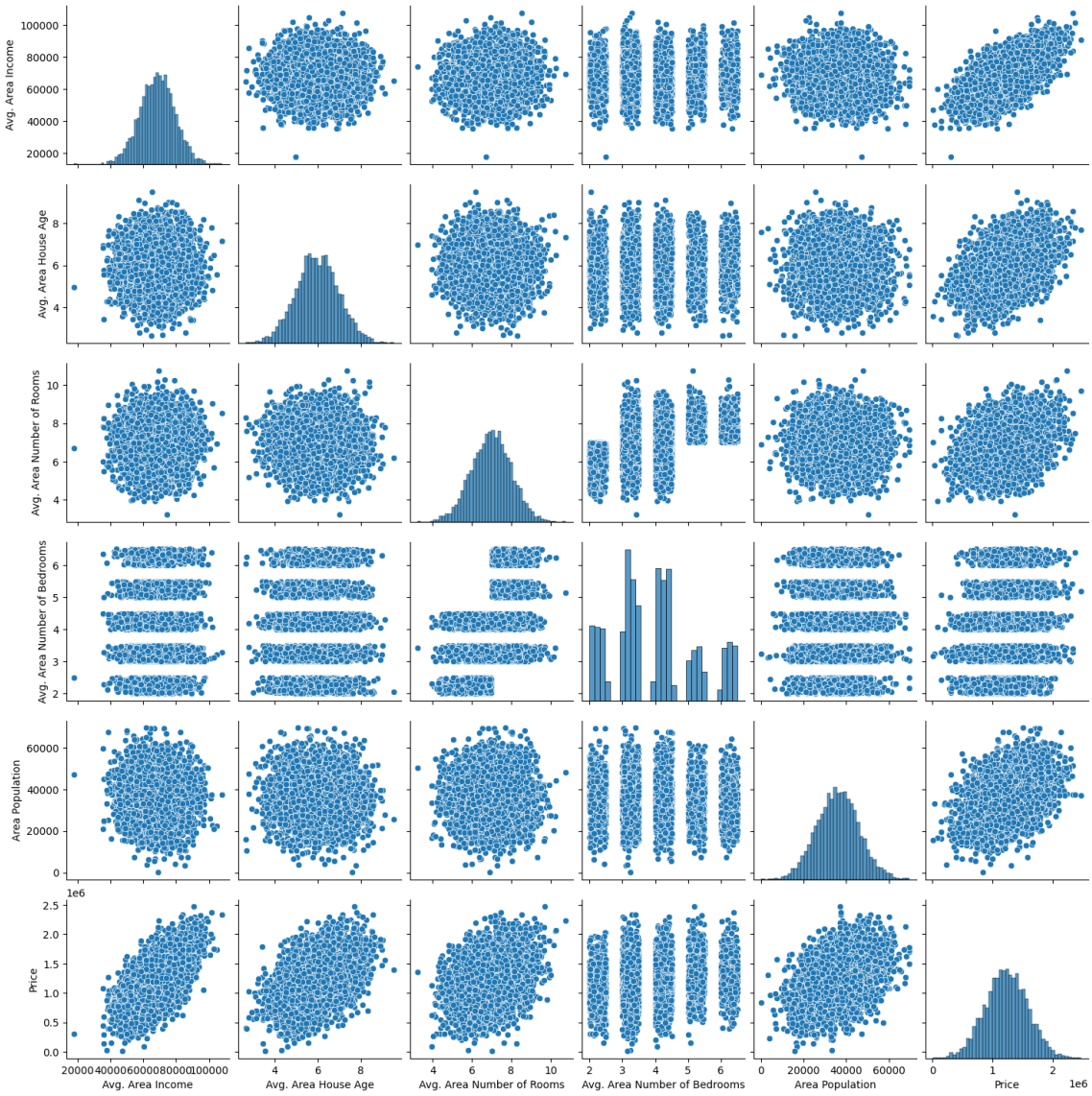
```
df.fillna(method='ffill',inplace=True)
```

In [27]:

```
sns.pairplot(df)
```

Out[27]:

```
<seaborn.axisgrid.PairGrid at 0x1d4e0e8b130>
```

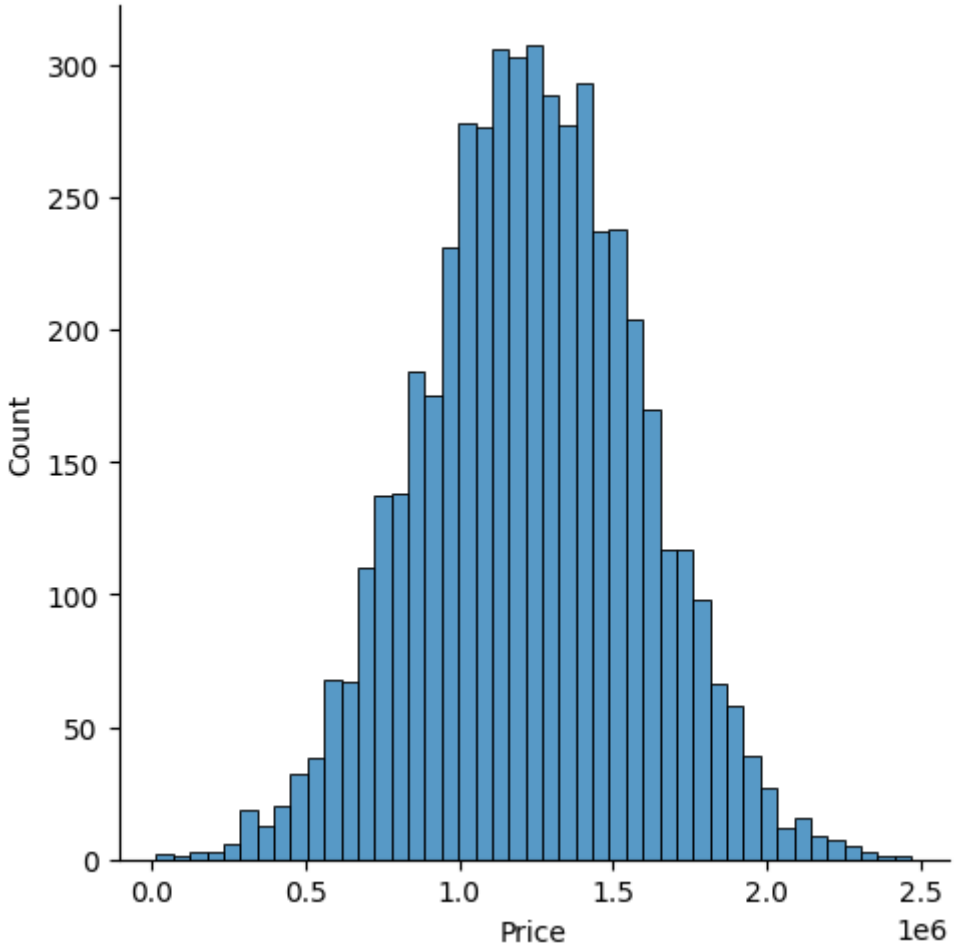


In [28]:

```
sns.displot(df['Price'])
```

Out[28]:

```
<seaborn.axisgrid.FacetGrid at 0x1d4f15df5e0>
```

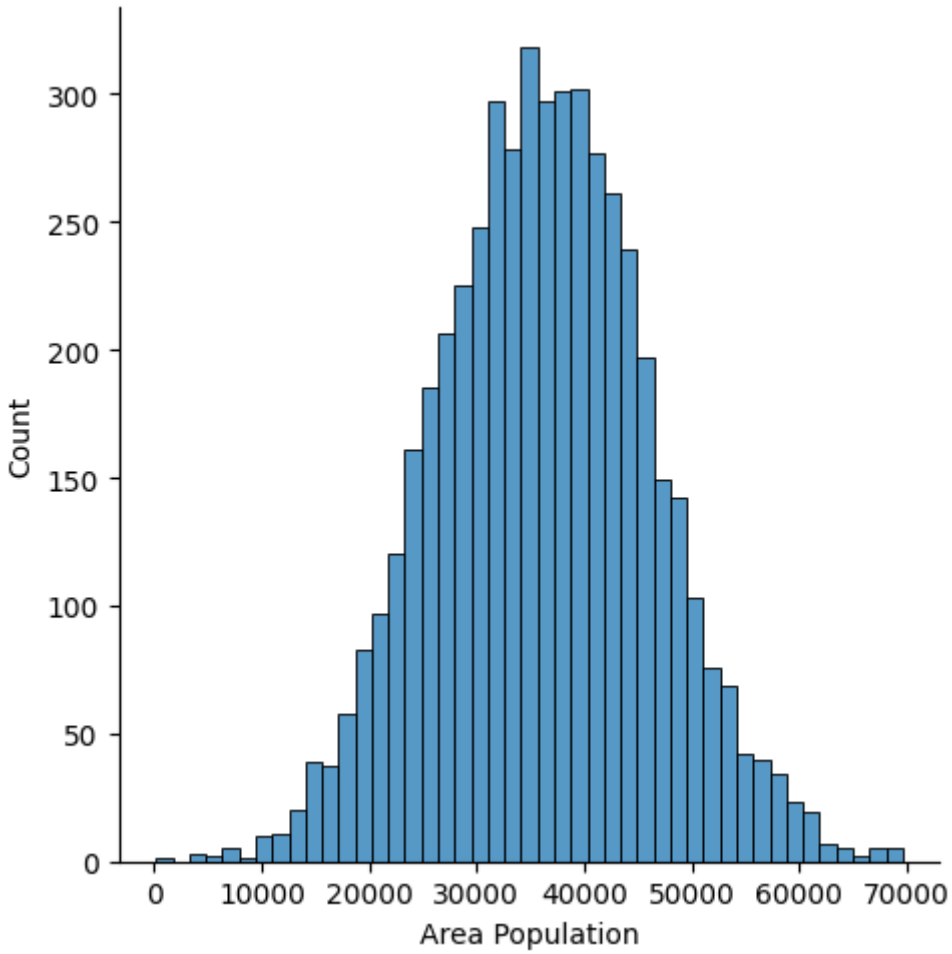


In [29]:

```
sns.displot(df['Area Population'])
```

Out[29]:

```
<seaborn.axisgrid.FacetGrid at 0x1d4f5e32a70>
```

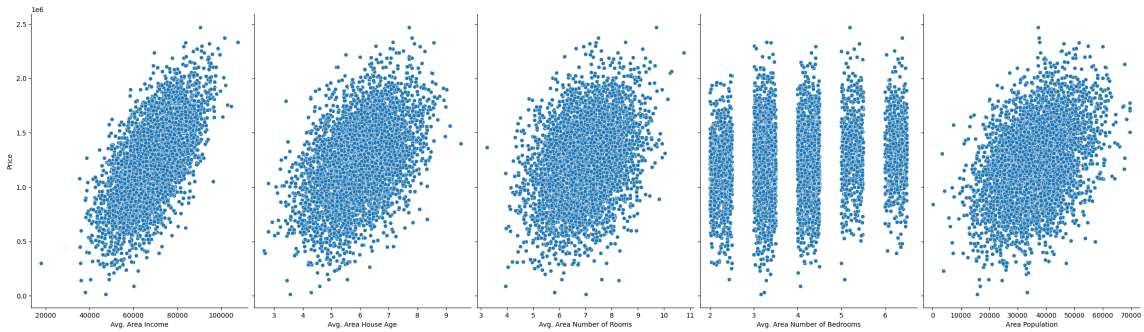


In [30]:

```
sns.pairplot(df, x_vars=['Avg. Area Income','Avg. Area House Age','Avg. Area Number of R
```

Out[30]:

```
<seaborn.axisgrid.PairGrid at 0x1d4f5eafc40>
```



In [31]:

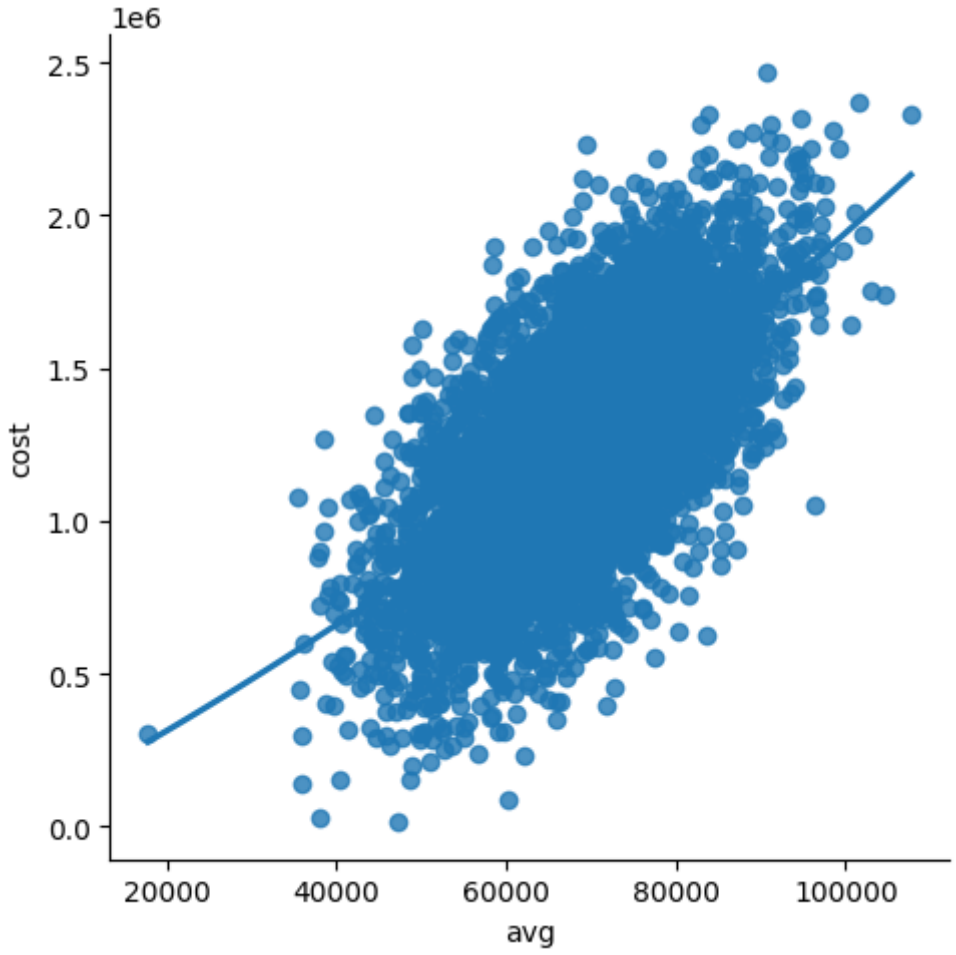
```
df=df[['Avg. Area Income','Price']]
df.columns=['avg','cost']
```

In [32]:

```
sns.lmplot(x='avg',y='cost',data=df,order=2,ci=None)
```

Out[32]:

```
<seaborn.axisgrid.FacetGrid at 0x1d4f6804550>
```



In [34]:

```
x=np.array(df['avg']).reshape(-1,1)
y=np.array(df['cost']).reshape(-1,1)
```

In [35]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print("Regression: ",regr.score(x_test,y_test))
```

```
Regression:  0.41802780615080537
```

In [36]:

```
df.dropna()
```

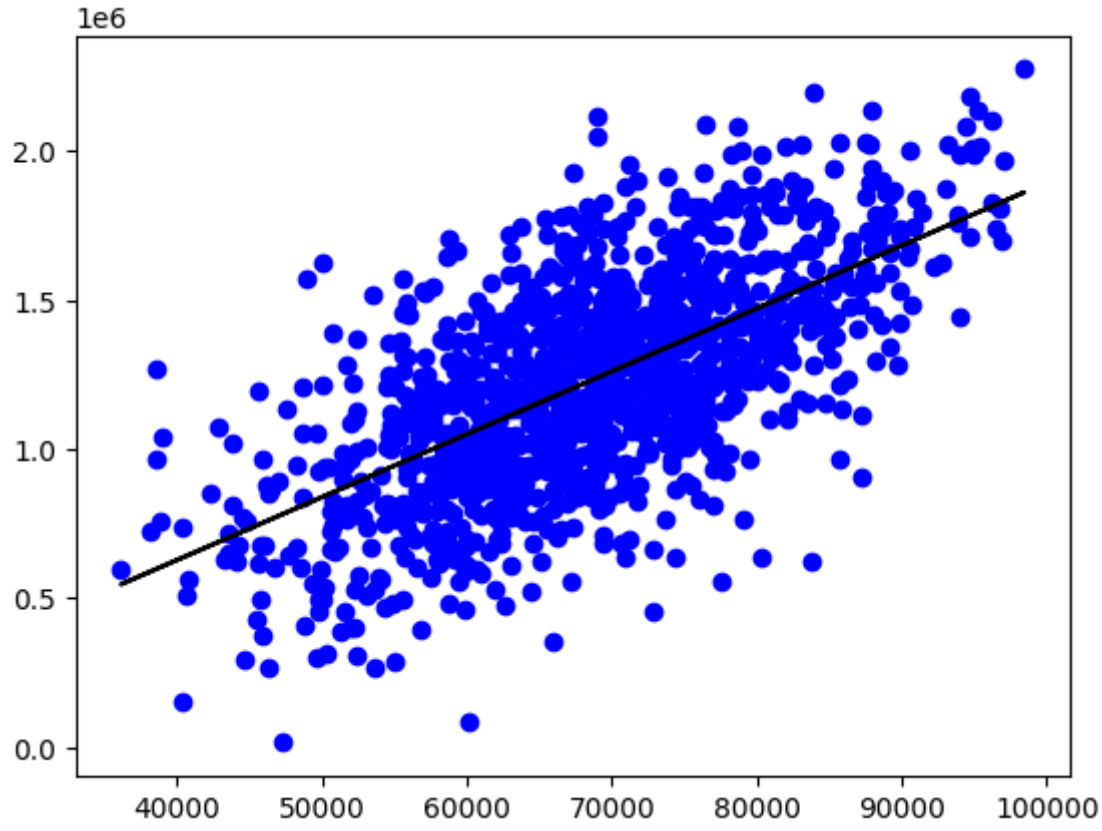
Out[36]:

	avg	cost
0	79545.458574	1.059034e+06
1	79248.642455	1.505891e+06
2	61287.067179	1.058988e+06
3	63345.240046	1.260617e+06
4	59982.197226	6.309435e+05
...	...	...
4995	60567.944140	1.060194e+06
4996	78491.275435	1.482618e+06
4997	63390.686886	1.030730e+06
4998	68001.331235	1.198657e+06
4999	65510.581804	1.298950e+06

5000 rows × 2 columns

In [37]:

```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

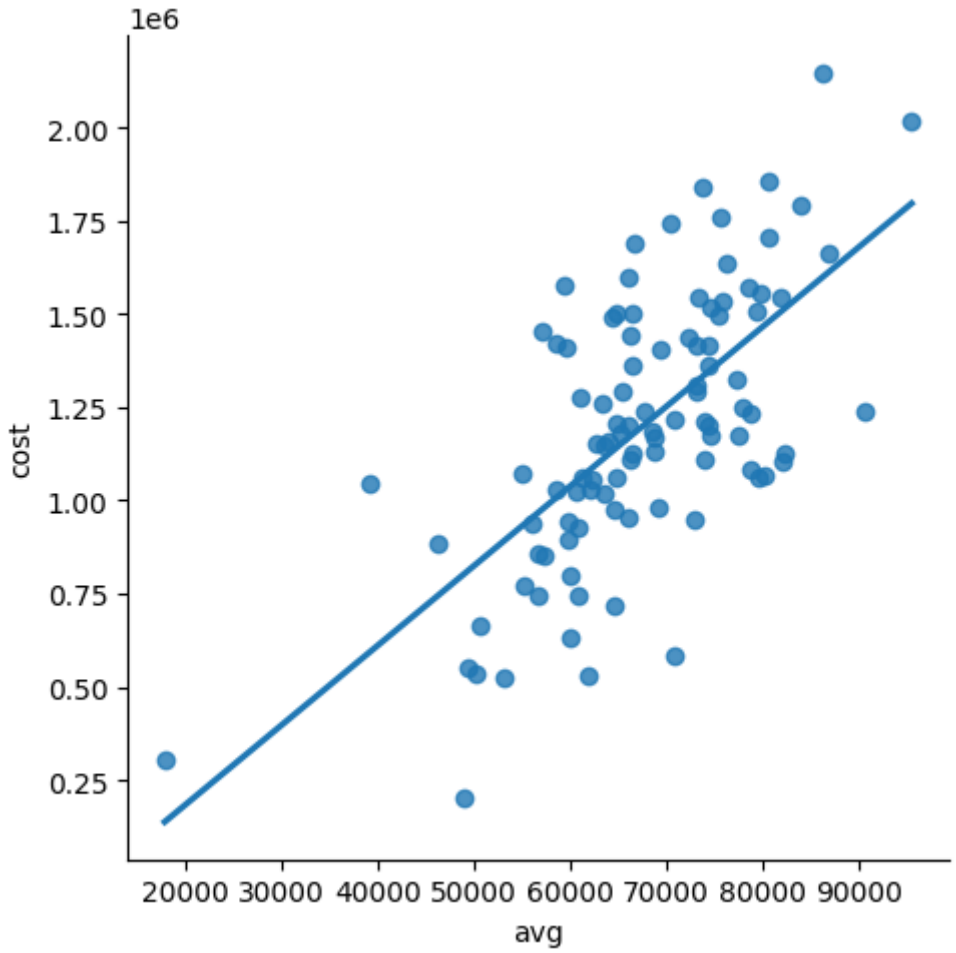


In [38]:

```
df100=df[:][:100]
sns.lmplot(x='avg',y='cost',data=df100,order=1,ci=None)
```

Out[38]:

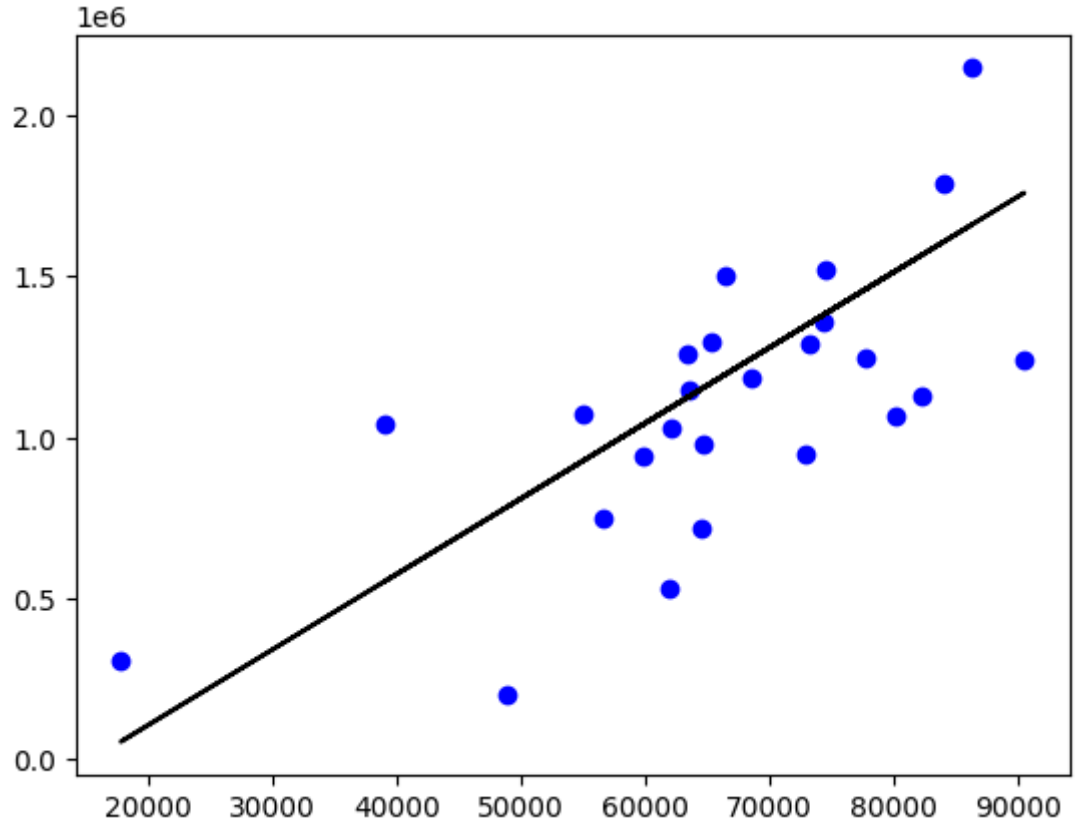
```
<seaborn.axisgrid.FacetGrid at 0x1d4f67a2f50>
```



In [39]:

```
df100.fillna(method='ffill',inplace=True)
x = np.array(df100['avg']).reshape(-1,1)
y = np.array(df100['cost']).reshape(-1,1)
df100.dropna(inplace=True)
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.25)
regr = LinearRegression()
regr.fit(x_train,y_train)
print("regression: ",regr.score(x_test,y_test))
y_pred = regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

```
regression:  0.39968917309103424
```



In [40]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model =LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2_Score: ",r2)
```

```
R2_Score:  0.39968917309103424
```

Conclusion: