In [2]:
```
import pandas as pd

file_path = './Downloads/StatsProject1Data.xlsx'
df = pd.read_excel(file_path)
print(df)
```

```
     Age  Gender Real Estate Purchases?  Value of Investments ($)  \
0     38  Female                     No                     12200
1     30    Male                     No                     12400
2     41  Female                     No                     26800
3     28  Female                    Yes                     19600
4     31  Female                    Yes                     15100
..   ...     ...                    ...                       ...
405   32  Female                     No                     27400
406   26  Female                    Yes                     14400
407   24    Male                    Yes                     36000
408   25  Female                    Yes                     30300
409   28  Female                    Yes                     30500

     Number of Transactions Broadband Access?  Household Income ($)  \
0                         4                Yes                 75200
1                         4                Yes                 70300
2                         5                Yes                 48200
3                         6                 No                 95300
4                         5                 No                 73300
..                      ...                ...                   ...
405                       7                Yes                 83500
406                       3                Yes                 28200
407                       4                Yes                103500
408                       6                 No                 35100
409                      11                Yes                 36800

    Have Children?  Unnamed: 8 Unnamed: 9  Unnamed: 10  Unnamed: 11  \
0              Yes         NaN        NaN          NaN          NaN
1              Yes         NaN        NaN          NaN          NaN
2               No         NaN        NaN          NaN          NaN
3               No         NaN        NaN          NaN          NaN
4              Yes         NaN          .          NaN          NaN
..             ...         ...        ...          ...          ...
405             No         NaN        NaN          NaN          NaN
406            Yes         NaN        NaN          NaN          NaN
407             No         NaN        NaN          NaN          NaN
408             No         NaN        NaN          NaN          NaN
409            Yes         NaN        NaN          NaN          NaN

     Unnamed: 12  Unnamed: 13
0            NaN          NaN
1            NaN          NaN
2            NaN          NaN
3            NaN          NaN
4            NaN          NaN
..           ...          ...
405          NaN          NaN
406          NaN          NaN
407          NaN          NaN
408          NaN          NaN
409          NaN          NaN

[410 rows x 14 columns]
```

In [3]:
```
#Central Tendency


#Mean for age
AgeMean = df['Age'].mean()
print("Age mean is: ", AgeMean)


#Median for age
AgeMedian = df['Age'].median()
print("Age median is: ", AgeMedian)


#Mean for household income
HouseholdIncomeMean = df['Household Income ($)'].mean()
print("HouseholdIncome mean is: ", HouseholdIncomeMean)



#Median for household income
HouseholdIncomeMedian = df['Household Income ($)'].median()
print("HouseholdIncome median is: ", HouseholdIncomeMedian)
```

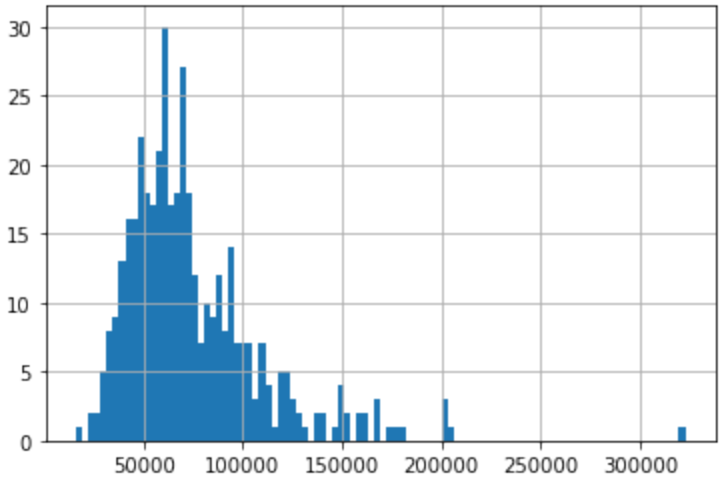
```
Age mean is:  30.11219512195122
Age median is:  30.0
HouseholdIncome mean is:  74459.51219512195
HouseholdIncome median is:  66050.0
```

In [4]:
```
#Variability
#Assess the range, variance, and standard deviation
#for age, household income, and value of investments.

AgeR = df['Age'].max() - df['Age'].min()
print("Range of Age: ", AgeR)
AgeV = df['Age'].var()
print("Variance of Age: ", AgeV)
AgeSD = df['Age'].std()
print("SD of Age: ", AgeSD)

print("")

HouseholdIncomeR = df['Household Income ($)'].max() - df['Household Incor
print("Range of HouseholdIncome: ", HouseholdIncomeR)
HouseholdIncomeV = df['Household Income ($)'].var()
print("Variance of HouseholdIncome: ", HouseholdIncomeV)
HouseholdIncomeSD = df['Household Income ($)'].std()
print("SD of HouseholdIncome: ", HouseholdIncomeSD)

print("")

ValueInvestmentsR = df['Value of Investments ($)'].max() - df['Value of I
print("Range of ValueInvestments: ", ValueInvestmentsR)
ValueInvestmentsV = df['Value of Investments ($)'].var()
print("Variance of ValueInvestments: ", ValueInvestmentsV)
ValueInvestmentsSD = df['Value of Investments ($)'].std()
print("SD of ValueInvestments: ", ValueInvestmentsSD)
```

```
Range of Age:  23
Variance of Age:  16.19276045083189
SD of Age:  4.024022918775674

Range of HouseholdIncome:  306300
Variance of HouseholdIncome:  1212307794.382491
SD of HouseholdIncome:  34818.2106717518

Range of ValueInvestments:  133400
Variance of ValueInvestments:  249982368.71608347
SD of ValueInvestments:  15810.830740858732
```

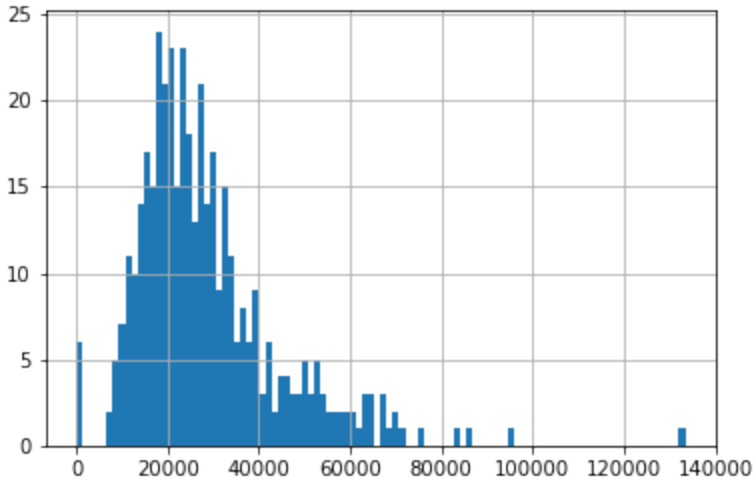
In [5]:
```
#Distributiomn Shape: Comment on skewnes or kurtosis for continuous varia


#Discrete RV are Age, Gender, Real Estate Purchases, Broadband access, Ha
#All Else are Continuous

import matplotlib.pyplot as plt
import numpy as np



df['Household Income ($)'].hist(bins=100)
```

Out[5]: <AxesSubplot:>



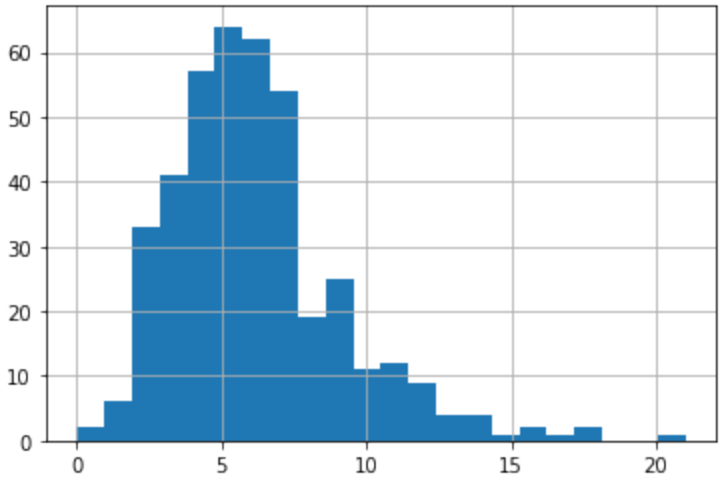
In [6]:
```
df['Value of Investments ($)'].hist(bins=100)
```

Out[6]: <AxesSubplot:>



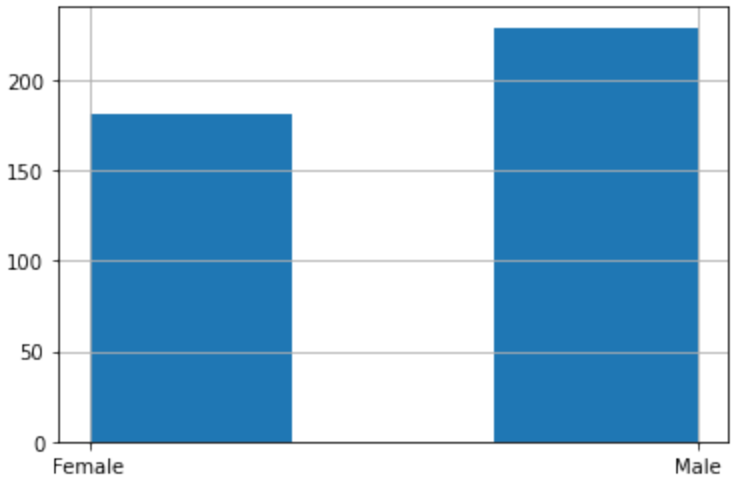
In [7]:
```
df['Number of Transactions'].hist(bins=22)

df['Number of Transactions'].mean()
#df['Number of Transactions'].median()
```

Out[7]: 5.9731707317073175



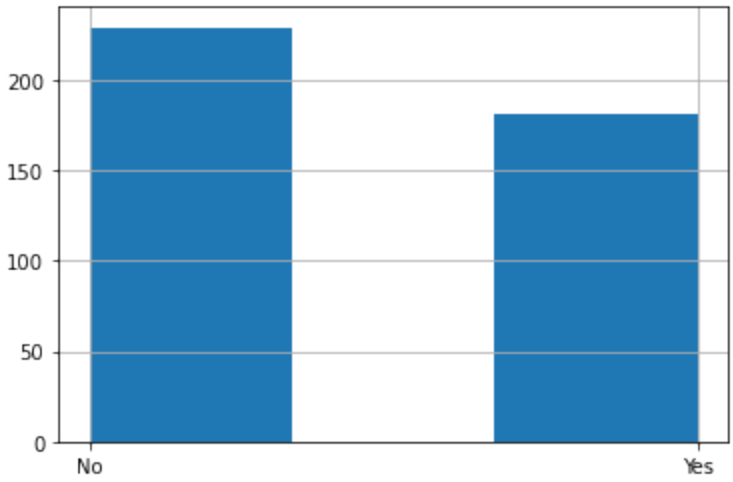
In [8]:
```
#NOW FREQUENCY DISTRIBUTIONS FOR GENDER, REAL ESTATE, BROADBAND, AND HAVI
df['Gender'].hist(bins=3)
gender_counts = df['Gender'].value_counts()
gender_percentages = gender_counts / len(df) * 100
print(gender_percentages)
```

```
Male      55.853659
Female    44.146341
Name: Gender, dtype: float64
```



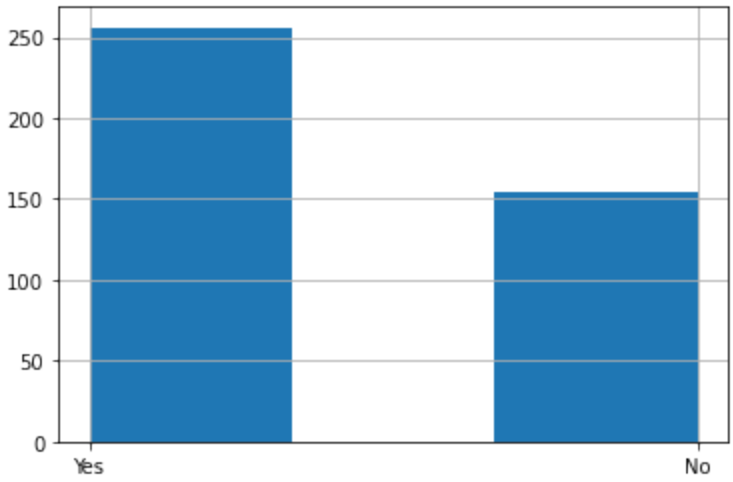
In [9]:
```
df['Real Estate Purchases?'].hist(bins=3)
RealEstate_counts = df['Real Estate Purchases?'].value_counts()
RealEstate_Percentages = RealEstate_counts / len(df) * 100
print(RealEstate_Percentages)
```

```
No     55.853659
Yes    44.146341
Name: Real Estate Purchases?, dtype: float64
```



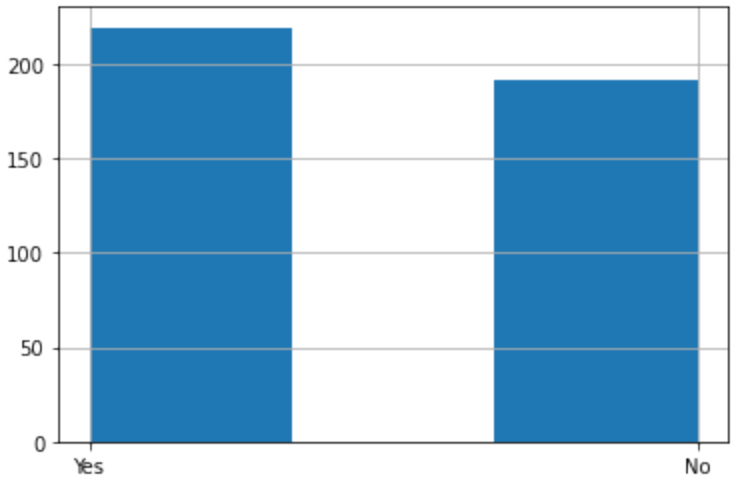
In [10]:
```
df['Broadband Access?'].hist(bins=3)
Broadband_counts = df['Broadband Access?'].value_counts()
Broadband_percentages = Broadband_counts / len(df) * 100
print(Broadband_percentages)
```

```
Yes    62.439024
No     37.560976
Name: Broadband Access?, dtype: float64
```



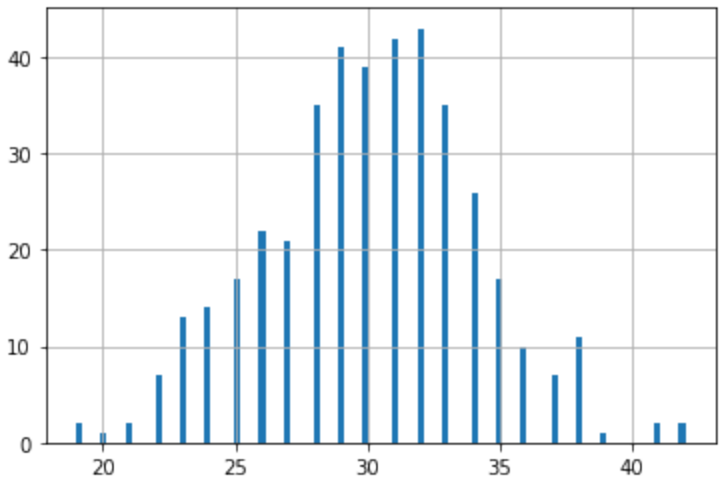
In [11]:
```
df['Have Children?'].hist(bins=3)
HaveChildren_counts = df['Have Children?'].value_counts()
HaveChildren_percentages = HaveChildren_counts / len(df) * 100
print(HaveChildren_percentages)
```

```
Yes    53.414634
No     46.585366
Name: Have Children?, dtype: float64
```



In [12]:
```
df['Age'].hist(bins=100)
```

Out[12]: <AxesSubplot:>



In [13]:
```
#Develop 95% confidence intervals for the mean age and household income (
import scipy.stats as stats
import math



sample_size = len(df)

sample_mean = df['Age'].mean()
z_critical = stats.norm.ppf(q = 0.975)
print(z_critical)
popstdev = df['Age'].std()

margin_of_error = z_critical * (popstdev/math.sqrt(sample_size))
confidence_interval = (sample_mean - margin_of_error, sample_mean + margi
print(confidence_interval)
```

```
1.959963984540054
(29.722686919009217, 30.501703324893224)
```

In [14]:
```
#HouseHold Income
sample_size = len(df)

sample_mean = df['Household Income ($)'].mean()
z_critical = stats.norm.ppf(q = 0.975)
print(z_critical)
popstdev = df['Household Income ($)'].std()

margin_of_error = z_critical * (popstdev/math.sqrt(sample_size))
confidence_interval = (sample_mean - margin_of_error, sample_mean + margi
print(confidence_interval)
```

```
1.959963984540054
(71089.25836154376, 77829.76602870013)
```

In [15]:
```
#CI Percent of subsribers with broadband access 'Yes'
from scipy.stats import norm

Broadband_Access_counts = df['Broadband Access?'].value_counts()
yesProportion = Broadband_Access_counts['Yes']/len(df)
SE = np.sqrt((yesProportion * (1 - yesProportion)) / (len(df)))
ME = norm.ppf(0.975) * SE
print(ME)
CI = (yesProportion - ME, yesProportion + ME)
print(CI)
```

```
0.046876227036195296
(0.5775140168662437, 0.6712664709386343)
```

In [16]:
```
#CI Percent of subsribers with broadband access 'No'

Broadband_Access_counts = df['Broadband Access?'].value_counts()
noProportion = Broadband_Access_counts['No']/len(df)
SE = np.sqrt((noProportion * (1 - noProportion)) / (len(df)))
ME = norm.ppf(0.975) * SE
print(ME)
CI = (noProportion - ME, noProportion + ME)
print(CI)
```

```
0.046876227036195296
(0.32873352906136566, 0.4224859831337563)
```

In [17]:
```
#CI Percent of subscribers with children 'Yes'

Children_counts = df['Have Children?'].value_counts()
yesProportion = Children_counts['Yes']/len(df)
SE = np.sqrt((yesProportion * (1 - yesProportion)) / (len(df)))
ME = norm.ppf(0.975) * SE
print(ME)
CI = (yesProportion - ME, yesProportion + ME)
print(CI)
```

```
0.0482848678338137
(0.48586147362960097, 0.5824312092972284)
```

In [18]:
```
#CI Percent of subscribers with children 'No'
Children_counts = df['Have Children?'].value_counts()
noProportion = Children_counts['No']/len(df)
SE = np.sqrt((noProportion * (1 - noProportion)) / (len(df)))
ME = norm.ppf(0.975) * SE
print(ME)
CI = (noProportion - ME, noProportion + ME)
print(CI)
```

```
0.0482848678338137
(0.4175687907027717, 0.5141385263703991)
```

In [19]:
```
print(df['Household Income ($)'].corr(df['Value of Investments ($)']))
```

```
0.003026142819767055
```

In [20]:
```
df.corr()
```

Out[20]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11
Age	1.000000	-0.029727	0.021208	0.010395	NaN	NaN	NaN
Value of Investments ($)	-0.029727	1.000000	0.085912	0.003026	NaN	NaN	NaN
Number of Transactions	0.021208	0.085912	1.000000	0.070633	NaN	NaN	NaN
Household Income ($)	0.010395	0.003026	0.070633	1.000000	NaN	NaN	NaN
Unnamed: 8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 11	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 12	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 13	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [21]:
```
df['Broadband Access?'].value_counts()
```

Out[21]:
```
Yes    256
No     154
Name: Broadband Access?, dtype: int64
```

In [22]:
```
len(df)
```

Out[22]:
```
410
```

In [23]:
```
#Investigative Part
#Would Young Professional be a good advertising outlet for online broker

#First area I want to explore is the dataset of people who have NO plans
#This is because I am making an educated guess that they have more room
#investments in online brokers, because they are not investing in real e



df2 = df[(df['Real Estate Purchases?'] == 'No')]
print("Household Income for people with NO real estate purchases: ", df2
df3 = df[(df['Real Estate Purchases?'] == 'Yes')]
print("Household Income for people WITH real estate purchases: ", df3['H

#By calculating the means of the household incomes of the two seperated
#people with no plans on purchasing Real Estate purchases actually make
#Which would imply that they could have much more moeny left over to inv

#Because there are more subscribers who DONT plan on real estate investm
#that many of them have alot of money left over without having to worry
```

```
Household Income for people with NO real estate purchases:  75695.19650
655022
Household Income for people WITH real estate purchases:  72896.13259668
509
```

In [24]:
```
df2['Value of Investments ($)'].mean()
```

Out[24]:
```
28140.174672489084
```

In [25]:
```
df3['Value of Investments ($)'].mean()
#Additionally the mean value of investments of someone without real esta
#and the mean value of investments of someone with real estate purchases
#The fact that these two means are so close, furthers our inference that
#(WHICH IS A PROPORTIONAL MAJORITY OF THE DATASET) Have more money to in
#because they both have very close mean values of investment values, sho
#and without real estate both invest signifiant amounts of money, but of
#that dont have real estate, the proportional majority of the dataset, t
#and dont pay a mortgage on real estate, leaving more available money fo
```

Out[25]:
```
29041.98895027624
```

In [26]:
```
df2['Have Children?'].value_counts()
```

Out[26]:
```
Yes    125
No     104
Name: Have Children?, dtype: int64
```

In [27]:
```
df3['Number of Transactions'].mean()
```

Out[27]:
```
6.1602209944751385
```

df2.corr()

In [28]:
```
df3.corr()
#Maybe Speak on this?? Compared to data as whole corr chart from above?
```

Out[28]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11
Age	1.000000	-0.060938	0.022316	-0.000521	NaN	NaN	NaN
Value of Investments ($)	-0.060938	1.000000	0.103161	-0.039616	NaN	NaN	NaN
Number of Transactions	0.022316	0.103161	1.000000	0.096742	NaN	NaN	NaN
Household Income ($)	-0.000521	-0.039616	0.096742	1.000000	NaN	NaN	NaN
Unnamed: 8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 11	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 12	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 13	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [29]:
```
#Now I would like to split the dataset into people with more than 6 trans
#im doing this because they have tendencies to make transactions, meaning

df4 = df[(df['Number of Transactions'] >= 6)]
df5 = df[(df['Number of Transactions'] < 6)]
```

In [30]:
```
df4.corr()
#Speak on correlation between value of investments and number of transac
#Higher with people who made more transactions, then those of subscriber
#Meaning you could easily infer that people making money on their invest
#of the value of their investments, would be willing to make more invest
#tangible return in them
```

Out[30]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11	U
Age	1.000000	0.034765	0.10781	0.048185	NaN	NaN	NaN	
Value of Investments ($)	0.034765	1.000000	0.17363	0.051299	NaN	NaN	NaN	
Number of Transactions	0.107810	0.173630	1.00000	0.087310	NaN	NaN	NaN	
Household Income ($)	0.048185	0.051299	0.08731	1.000000	NaN	NaN	NaN	
Unnamed: 8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Unnamed: 10	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Unnamed: 11	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Unnamed: 12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
Unnamed: 13	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

In [31]:
```
df5.corr()
```

Out[31]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11
Age	1.000000	-0.094214	-0.060912	-0.035542	NaN	NaN	NaN
Value of Investments ($)	-0.094214	1.000000	0.049151	-0.053889	NaN	NaN	NaN
Number of Transactions	-0.060912	0.049151	1.000000	0.058933	NaN	NaN	NaN
Household Income ($)	-0.035542	-0.053889	0.058933	1.000000	NaN	NaN	NaN
Unnamed: 8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 11	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 12	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 13	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [32]:
```
df3.describe()
```

Out[32]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11	U
count	181.000000	181.000000	181.000000	181.000000	0.0	0.0	0.0	
mean	30.215470	29041.988950	6.160221	72896.132597	NaN	NaN	NaN	
std	3.962741	15357.535408	3.062636	32716.411724	NaN	NaN	NaN	
min	19.000000	0.000000	0.000000	24300.000000	NaN	NaN	NaN	
25%	28.000000	18700.000000	4.000000	49600.000000	NaN	NaN	NaN	
50%	31.000000	24700.000000	6.000000	65700.000000	NaN	NaN	NaN	
75%	33.000000	36000.000000	8.000000	89500.000000	NaN	NaN	NaN	
max	41.000000	95200.000000	21.000000	201700.000000	NaN	NaN	NaN	

In [33]:
```
df6 = df[(df['Real Estate Purchases?'] == 'No') | (df['Number of Transac
df6
```

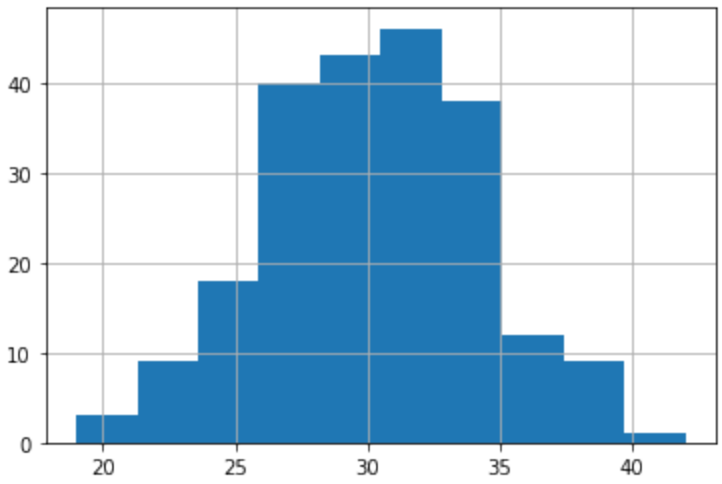
Out[33]:

	Age	Gender	Real Estate Purchases?	Value of Investments ($)	Number of Transactions	Broadband Access?	Household Income ($)	Have Children?	Unn
0	38	Female	No	12200	4	Yes	75200	Yes	
1	30	Male	No	12400	4	Yes	70300	Yes	
2	41	Female	No	26800	5	Yes	48200	No	
3	28	Female	Yes	19600	6	No	95300	No	
5	32	Male	No	39700	3	Yes	123400	Yes	
...	...	...	...	...	...	...	...	...	
400	31	Female	Yes	37100	6	Yes	77100	Yes	
402	30	Male	Yes	19300	6	No	52700	No	
405	32	Female	No	27400	7	Yes	83500	No	
408	25	Female	Yes	30300	6	No	35100	No	
409	28	Female	Yes	30500	11	Yes	36800	Yes	

330 rows × 14 columns

In [34]:
```
df7 = df[(df['Have Children?'] == 'Yes')]
df7['Age'].hist(bins=10)
```

Out[34]: <AxesSubplot:>



In [35]:
```
df8 = df7[(df7['Age'] >= 25) & (df7['Age'] <= 35) & (df7['Broadband Acce
df8
```

Out[35]:

	Age	Gender	Real Estate Purchases?	Value of Investments ($)	Number of Transactions	Broadband Access?	Household Income ($)	Have Children?	Unn
1	30	Male	No	12400	4	Yes	70300	Yes	
5	32	Male	No	39700	3	Yes	123400	Yes	
6	32	Male	Yes	21900	2	Yes	73900	Yes	
7	26	Female	Yes	41900	2	Yes	54300	Yes	
12	28	Female	No	17300	7	Yes	73600	Yes	
...	...	...	...	...	...	...	...	...	
395	32	Male	Yes	20600	12	Yes	93600	Yes	
400	31	Female	Yes	37100	6	Yes	77100	Yes	
401	32	Female	Yes	64100	4	Yes	62300	Yes	
406	26	Female	Yes	14400	3	Yes	28200	Yes	
409	28	Female	Yes	30500	11	Yes	36800	Yes	

114 rows × 14 columns

In [36]:
```
df7
```

Out[36]:

	Age	Gender	Real Estate Purchases?	Value of Investments ($)	Number of Transactions	Broadband Access?	Household Income ($)	Have Children?	Unn
0	38	Female	No	12200	4	Yes	75200	Yes	
1	30	Male	No	12400	4	Yes	70300	Yes	
4	31	Female	Yes	15100	5	No	73300	Yes	
5	32	Male	No	39700	3	Yes	123400	Yes	
6	32	Male	Yes	21900	2	Yes	73900	Yes	
...	...	...	...	...	...	...	...	...	
396	37	Male	Yes	38300	4	No	100300	Yes	
400	31	Female	Yes	37100	6	Yes	77100	Yes	
401	32	Female	Yes	64100	4	Yes	62300	Yes	
406	26	Female	Yes	14400	3	Yes	28200	Yes	
409	28	Female	Yes	30500	11	Yes	36800	Yes	

219 rows × 14 columns

In [37]:
```
from scipy.stats import norm, kurtosis


kurtosis(df['Number of Transactions'])
```

Out[37]:
```
2.4159227986976157
```

In [39]:
```
#This is the cell for other types of articles of interest to readers!




mean_income_30_and_older = df[df['Age'] >= 30]['Household Income ($)'].m

mean_income_younger_than_30 = df[df['Age'] < 30]['Household Income ($)']

print("Mean household income for people of age 30 and older:", mean_incor
print("Mean household income for people younger than 30:", mean_income_y
```

```
Mean household income for people of age 30 and older: 73621.27659574468
Mean household income for people younger than 30: 75585.14285714286
```

In [44]:
```
income_30_and_older = df[df['Age'] >= 30]
income_younger_than_30 = df[df['Age'] < 30]
```

In [49]:
```
df_men = df[df['Gender'] == 'Male']
df_women = df[df['Gender'] == 'Female']
```

In [50]:
```
df_men.describe()
```

Out[50]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11
count	229.000000	229.000000	229.000000	229.000000	0.0	0.0	0.0
mean	29.925764	27538.427948	5.921397	75734.061135	NaN	NaN	NaN
std	4.146304	15808.914018	2.950309	32320.174184	NaN	NaN	NaN
min	19.000000	0.000000	1.000000	16200.000000	NaN	NaN	NaN
25%	27.000000	18100.000000	4.000000	53300.000000	NaN	NaN	NaN
50%	30.000000	23700.000000	6.000000	68300.000000	NaN	NaN	NaN
75%	32.000000	33300.000000	7.000000	93100.000000	NaN	NaN	NaN
max	42.000000	133400.000000	18.000000	201700.000000	NaN	NaN	NaN

In [51]:
```
df_women.describe()
```

Out[51]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11
count	181.000000	181.000000	181.000000	181.000000	0.0	0.0	0.0
mean	30.348066	29803.314917	6.038674	72846.961326	NaN	NaN	NaN
std	3.862262	15766.003864	3.288641	37772.996736	NaN	NaN	NaN
min	22.000000	8200.000000	0.000000	24300.000000	NaN	NaN	NaN
25%	28.000000	18300.000000	4.000000	49200.000000	NaN	NaN	NaN
50%	31.000000	26800.000000	6.000000	63300.000000	NaN	NaN	NaN
75%	33.000000	36100.000000	7.000000	86200.000000	NaN	NaN	NaN
max	41.000000	95200.000000	21.000000	322500.000000	NaN	NaN	NaN

df_men.corr()

In [52]:
```
df_men.corr()
```

Out[52]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11
Age	1.000000	-0.049585	-0.015179	0.027459	NaN	NaN	NaN
Value of Investments ($)	-0.049585	1.000000	-0.038274	0.035188	NaN	NaN	NaN
Number of Transactions	-0.015179	-0.038274	1.000000	0.099210	NaN	NaN	NaN
Household Income ($)	0.027459	0.035188	0.099210	1.000000	NaN	NaN	NaN
Unnamed: 8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 11	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 12	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 13	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [53]:
```
df_women.corr()
```

Out[53]:

	Age	Value of Investments ($)	Number of Transactions	Household Income ($)	Unnamed: 8	Unnamed: 10	Unnamed: 11
Age	1.000000	-0.011788	0.063668	-0.004572	NaN	NaN	NaN
Value of Investments ($)	-0.011788	1.000000	0.225322	-0.025718	NaN	NaN	NaN
Number of Transactions	0.063668	0.225322	1.000000	0.044561	NaN	NaN	NaN
Household Income ($)	-0.004572	-0.025718	0.044561	1.000000	NaN	NaN	NaN
Unnamed: 8	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 10	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 11	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 12	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Unnamed: 13	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [ ]:
```
#The correlation between the number of transactions (Stocks/Bonds/Market)
#and the corresponding value of those investments in women is almost 30%
#stronger in women then men

#This technically leads us to believe that women in this study make more
#Intelligent investing decisions then men, and see greater returns
#on their investments
```