```
import pandas as pd
```

```
df = pd.read_excel('census2021firstresultsenglandwales1.xlsx', sheet_name='P01')
```

```
df.head()
```

	Area code [note 2]	Area name	All persons	Females	Males
0	K04000001	England and Wales	59597300	30420100	29177200
1	E92000001	England	56489800	28833500	27656300
2	E12000001	North East	2647100	1353800	1293300
3	E06000047	County Durham	522100	266800	255300
4	E06000005	Darlington	107800	55100	52700

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 375 entries, 0 to 374
Data columns (total 5 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Area code [note 2]  375 non-null    object
 1   Area name           375 non-null    object
 2   All persons         375 non-null    int64
 3   Females             375 non-null    int64
 4   Males               375 non-null    int64
dtypes: int64(3), object(2)
memory usage: 14.8+ KB
```

```
df_sorted = df.sort_values(by='Area name')
```

```
df_sorted
```

	Area code [note 2]	Area name	All persons	Females	Males
311	E07000223	Adur	64500	33300	31200
24	E07000026	Allerdale	96100	49100	47100
93	E07000032	Amber Valley	126200	64200	62000
312	E07000224	Arun	164800	85400	79400
118	E07000170	Ashfield	126300	64400	61900
...	...	...	...	...	...
158	E07000238	Wychavon	132500	67500	65000
53	E07000128	Wyre	111900	57500	54400
159	E07000239	Wyre Forest	101600	51800	49800
65	E06000014	York	202800	105300	97500
60	E12000003	Yorkshire and The Humber	5480800	2791800	2689000

375 rows × 5 columns

```
print("Missing values for each column:")
print(df.isnull().sum())
```

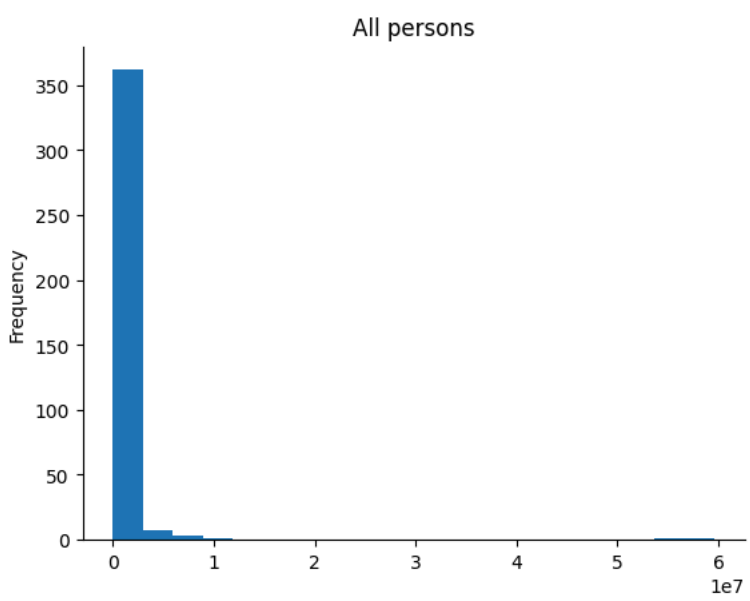
```
Missing values for each column:
Area code [note 2]    0
Area name             0
All persons           0
Females               0
Males                 0
dtype: int64
```

```
duplicates = df[df.duplicated()]
print("Duplicate rows:\n", duplicates)
```

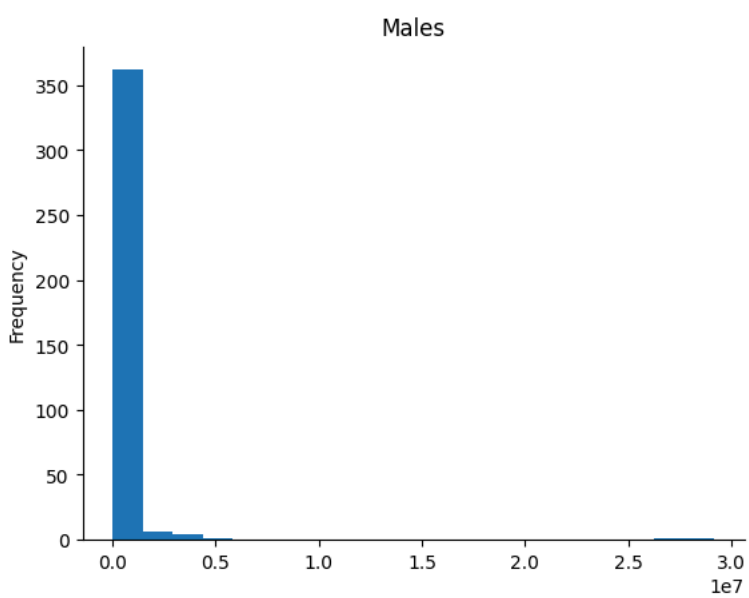
```
Duplicate rows:
 Empty DataFrame
Columns: [Area code [note 2], Area name, All persons, Females, Males]
Index: []
```

All persons

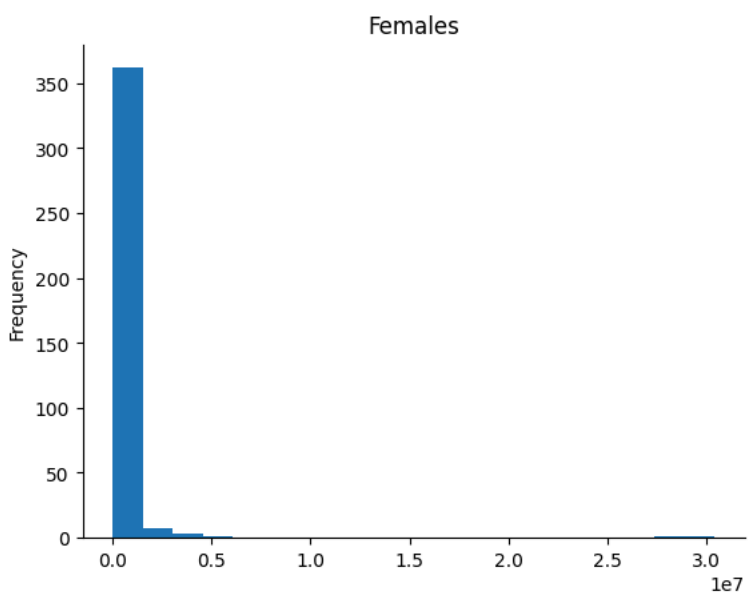
```
# @title All persons

from matplotlib import pyplot as plt
df_sorted['All persons'].plot(kind='hist', bins=20, title='All persons')
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
df_sorted['Males'].plot(kind='hist', bins=20, title='Males')
plt.gca().spines[['top', 'right',]].set_visible(False)
```

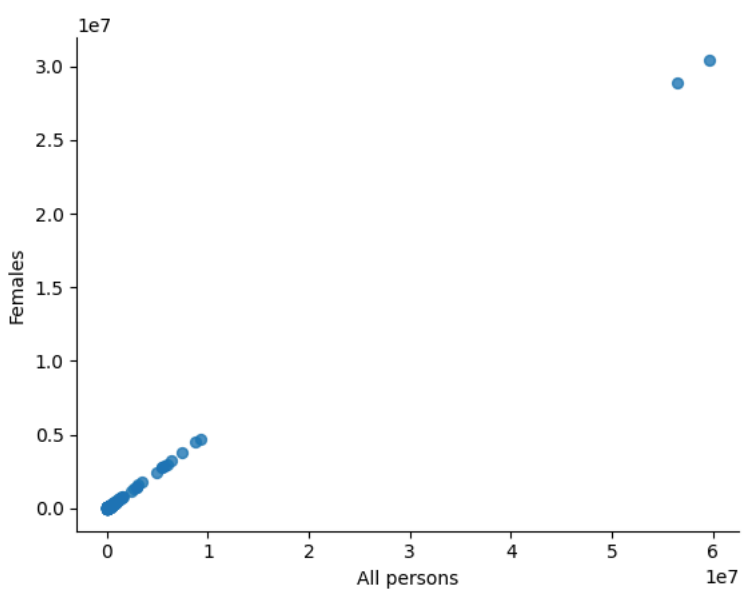



```
df_sorted['Females'].plot(kind='hist', bins=20, title='Females')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



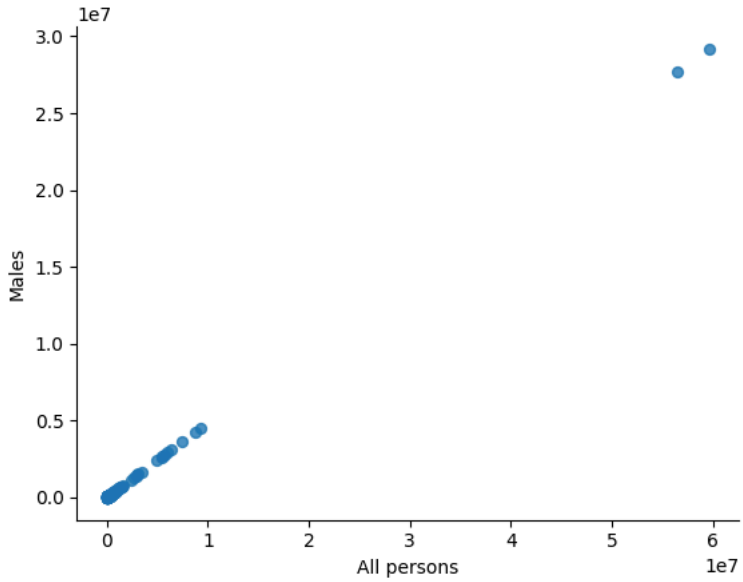
All persons vs Females

```
# @title All persons vs Females

from matplotlib import pyplot as plt
df_sorted.plot(kind='scatter', x='All persons', y='Females', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```



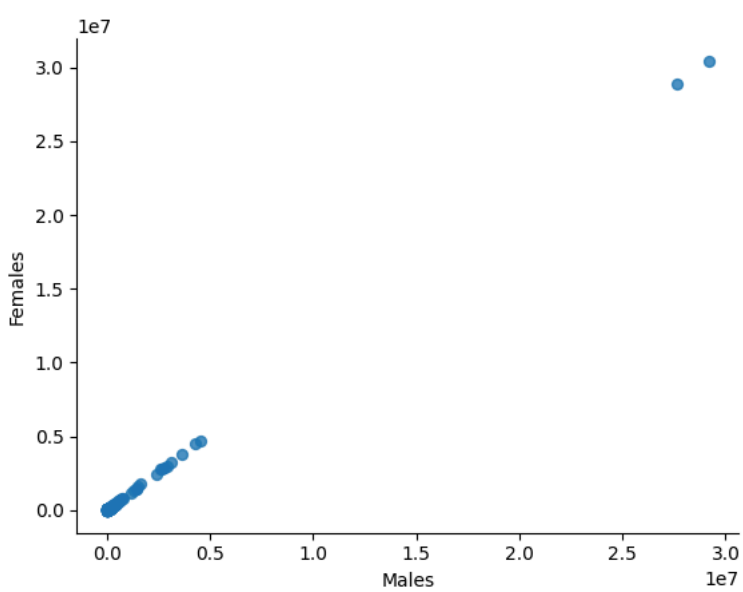
All persons vs Males

```
# @title All persons vs Males

from matplotlib import pyplot as plt
df_sorted.plot(kind='scatter', x='All persons', y='Males', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```



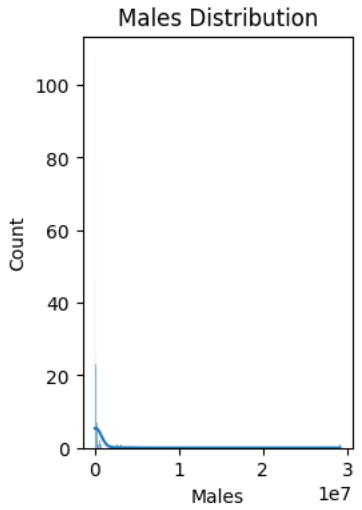
All persons vs Females

```
# @title All persons vs Females
df_sorted.plot(kind='scatter', x='Males', y='Females', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```



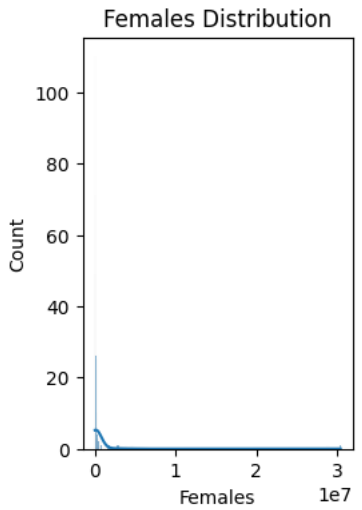

```
import seaborn as sns
# Visualizing distributions with histograms
plt.figure(figsize=(12, 4))
plt.subplot(1, 4, 1)
sns.histplot(df['Males'], kde=True)
plt.title('Males Distribution')
```

```
Text(0.5, 1.0, 'Males Distribution')
```




```
# Visualizing distributions with histograms
plt.figure(figsize=(12, 4))
plt.subplot(1, 4, 1)
sns.histplot(df['Females'], kde=True)
plt.title('Females Distribution')
```

```
Text(0.5, 1.0, 'Females Distribution')
```



Working with Sheet 2. This sheet is for the number of people per age group

```
import pandas as pd
sheet2 = pd.read_excel('census2021firstresultsenglandwales1.xlsx', sheet_name='P02', skiprows=8)
```

```
for col in sheet2.columns[9:]:
    # Only apply if column is of string type
    if sheet2[col].dtype == 'str':
        sheet[col] = sheet2[col].str[:-6]
```

```
print("Updated column names:")
print(sheet2.columns.tolist())
```

```
Updated column names:
['Area code [note 2]', 'Area name', 'All persons', 'Aged 4 years and under', 'Aged 5 to 9 years', 'Aged 10 to 14 years', 'Aged 15 to
```

```
sheet2.head()
```

	Area code [note 2]	Area name	All persons	Aged 4 years and under	Aged 5 to 9 years	Aged 10 to 14 years	Aged 15 to 19 years	Aged 20 to 24 years	Aged 25 to 29 years\n[note 12]	Aged 30 to 34 years\n[note 12]	...	Aged 45 to 49 years\n[note 12]	Aged years\
0	K04000001	England and Wales	59597300	3232100	3524600	3595900	3394700	3602100	3901800	4148800	...	3788700	4
1	E92000001	England	56489800	3077000	3348600	3413100	3218900	3414400	3715400	3952600	...	3602600	3
2	E12000001	North East	2647100	134300	150500	154400	150100	162900	160700	168000	...	159000	
3	E06000047	County Durham	522100	24800	28400	29500	31200	33100	29200	30700	...	31600	
4	E06000005	Darlington	107800	5500	6300	6600	5800	5400	6400	7000	...	6900	

5 rows × 22 columns

```
print("Missing values for each column:")
print(sheet2.isnull().sum())
```

```
Missing values for each column:
Area code [note 2]                   0
Area name                            0
All persons                          0
Aged 4 years and under               0
Aged 5 to 9 years                    0
Aged 10 to 14 years                  0
Aged 15 to 19 years                  0
Aged 20 to 24 years                  0
Aged 25 to 29 years\n[note 12]       0
Aged 30 to 34 years\n[note 12]       0
Aged 35 to 39 years\n[note 12]       0
Aged 40 to 44 years\n[note 12]       0
Aged 45 to 49 years\n[note 12]       0
Aged 50 to 54 years\n[note 12]       0
Aged 55 to 59 years\n[note 12]       0
Aged 60 to 64 years\n[note 12]       0
Aged 65 to 69 years\n[note 12]       0
Aged 70 to 74 years\n[note 12]       0
Aged 75 to 79 years\n[note 12]       0
Aged 80 to 84 years\n[note 12]       0
Aged 85 to 89 years\n[note 12]       0
Aged 90 years and over\n[note 12]    0
dtype: int64
```

```
duplicates_sheet2 = sheet2[sheet2.duplicated()]
print("Duplicate rows:\n", duplicates_sheet2)
```

```
Duplicate rows:
 Empty DataFrame
Columns: [Area code [note 2], Area name, All persons, Aged 4 years and under, Aged 5 to 9 years, Aged 10 to 14 years, Aged 15 to 19
[note 12], Aged 30 to 34 years
[note 12], Aged 35 to 39 years
[note 12], Aged 40 to 44 years
[note 12], Aged 45 to 49 years
[note 12], Aged 50 to 54 years
[note 12], Aged 55 to 59 years
[note 12], Aged 60 to 64 years
[note 12], Aged 65 to 69 years
[note 12], Aged 70 to 74 years
[note 12], Aged 75 to 79 years
[note 12], Aged 80 to 84 years
[note 12], Aged 85 to 89 years
[note 12], Aged 90 years and over
[note 12]]
Index: []

[0 rows x 22 columns]
```

```
duplicates_sheet2
```

Area code [note 2]	Area name	All persons	Aged 4 years and under	Aged 5 to 9 years	Aged 10 to 14 years	Aged 15 to 19 years	Aged 20 to 24 years	Aged 25 to 29 years\n[note 12]	Aged 30 to 34 years\n[note 12]	...	Aged 45 to 49 years\n[note 12]	Aged 50 to 54 years\n[note 12]	Aged 55 to 59 years\n[note 12]

0 rows × 22 columns

```
from matplotlib import pyplot as plt


# Remove '\n[note 12]' and strip spaces from column names
sheet2.columns = sheet2.columns.str.replace('\n[note 12]', '', regex=False).str.strip()

# Display new column names
print(sheet2.columns.tolist())
```

```
['Area code [note 2]', 'Area name', 'All persons', 'Aged 4 years and under', 'Aged 5 to 9 years', 'Aged 10 to 14 years', 'Aged 15 to
```

```
sheet2_sorted = sheet2.sort_values(by='Area name')
```

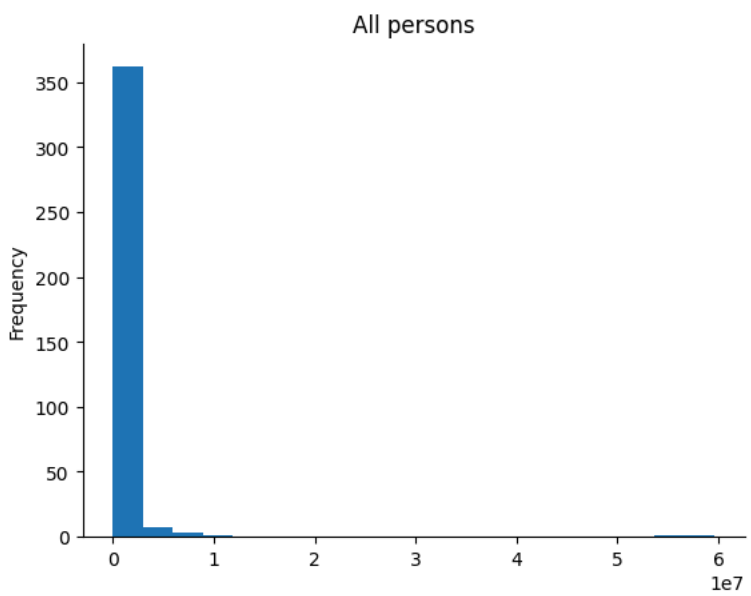
```
sheet2_sorted
```

	Area code [note 2]	Area name	All persons	Aged 4 years and under	Aged 5 to 9 years	Aged 10 to 14 years	Aged 15 to 19 years	Aged 20 to 24 years	Aged 25 to 29 years	Aged 30 to 34 years	...	Aged 45 to 49 years	Aged 50 to 54 years	Aged 55 to 59 years	Aged 60 to 64 years	Ag 65 yea
311	E07000223	Adur	64500	3200	3900	3800	3200	2700	3200	3700	...	4500	4700	4500	4000	36
24	E07000026	Allerdale	96100	4400	4900	5400	4600	4400	5200	5200	...	5800	7600	7600	7100	63
93	E07000032	Amber Valley	126200	6100	6700	7100	6000	6000	7200	7500	...	8400	9900	9800	8400	75
312	E07000224	Arun	164800	7300	8200	8300	7600	7400	8400	9300	...	9400	11400	11900	11000	107
118	E07000170	Ashfield	126300	6800	7700	7600	6500	6400	7900	8600	...	8100	9500	9200	7700	66
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
158	E07000238	Wychavon	132500	6200	6900	7100	6500	5700	6900	7600	...	8400	10100	10300	9000	84
53	E07000128	Wyre	111900	4900	5600	6000	5700	4800	5500	5800	...	6400	8000	8800	8400	76
159	E07000239	Wyre Forest	101600	4800	5500	5600	5000	4700	5400	5900	...	6400	7700	7500	6400	63
65	E06000014	York	202800	8600	9900	10300	15200	20400	13000	12900	...	11900	13300	13000	11300	98
60	E12000003	Yorkshire and The Humber	5480800	298600	326100	332600	323100	347700	350900	367400	...	338000	380600	373900	327500	2807

375 rows × 22 columns

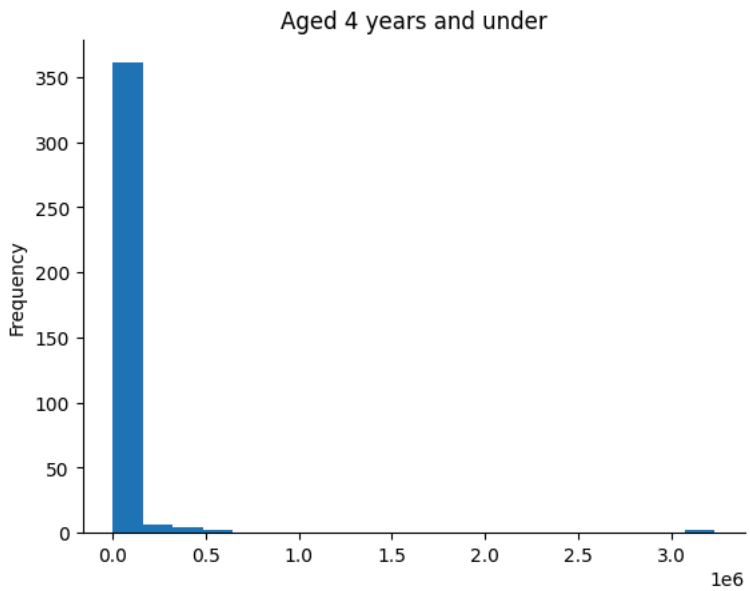
All persons

```
# @title All persons

from matplotlib import pyplot as plt
sheet2_sorted['All persons'].plot(kind='hist', bins=20, title='All persons')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



All persons

```
# @title All persons

from matplotlib import pyplot as plt
sheet2_sorted['Aged 4 years and under'].plot(kind='hist', bins=20, title='Aged 4 years and under')
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
# Remove '\n[note 12]' and strip spaces from column names
sheet2.columns = sheet2.columns.str.replace('\n[note 12]', '', regex=False).str.strip()

# Display new column names
print(sheet2.columns.tolist())
```

```
['Area code [note 2]', 'Area name', 'All persons', 'Aged 4 years and under', 'Aged 5 to 9 years', 'Aged 10 to 14 years', 'Aged 15 to
```

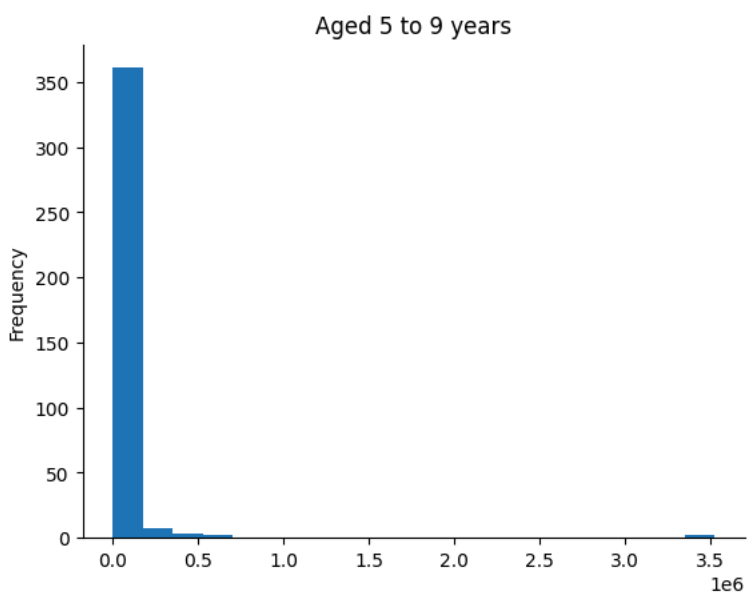
```
sheet2.head()
```

	Area code [note 2]	Area name	All persons	Aged 4 years and under	Aged 5 to 9 years	Aged 10 to 14 years	Aged 15 to 19 years	Aged 20 to 24 years	Aged 25 to 29 years	Aged 30 to 34 years	...	Aged 45 to 49 years	Aged 50 to 54 years	Aged 55 to 59 years	Age
0	K04000001	England and Wales	59597300	3232100	3524600	3595900	3394700	3602100	3901800	4148800	...	3788700	4123400	4029000	345
1	E92000001	England	56489800	3077000	3348600	3413100	3218900	3414400	3715400	3952600	...	3602600	3907700	3806300	325
2	E12000001	North East	2647100	134300	150500	154400	150100	162900	160700	168000	...	159000	183400	192300	17
3	E06000047	County Durham	522100	24800	28400	29500	31200	33100	29200	30700	...	31600	37900	39500	3
4	E06000005	Darlington	107800	5500	6300	6600	5800	5400	6400	7000	...	6900	7800	7900	

5 rows × 22 columns

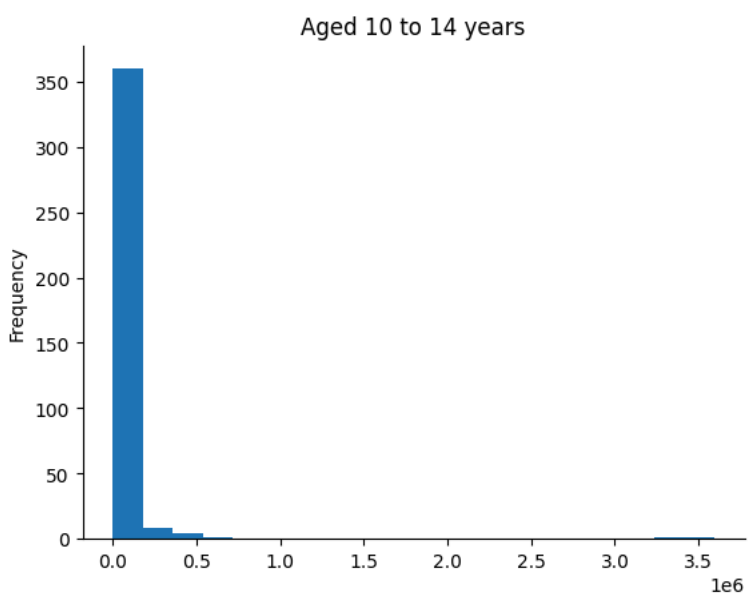
Aged 5 to 9 years

```
# @title Aged 5 to 9 years

sheet2_sorted['Aged 5 to 9 years'].plot(kind='hist', bins=20, title='Aged 5 to 9 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



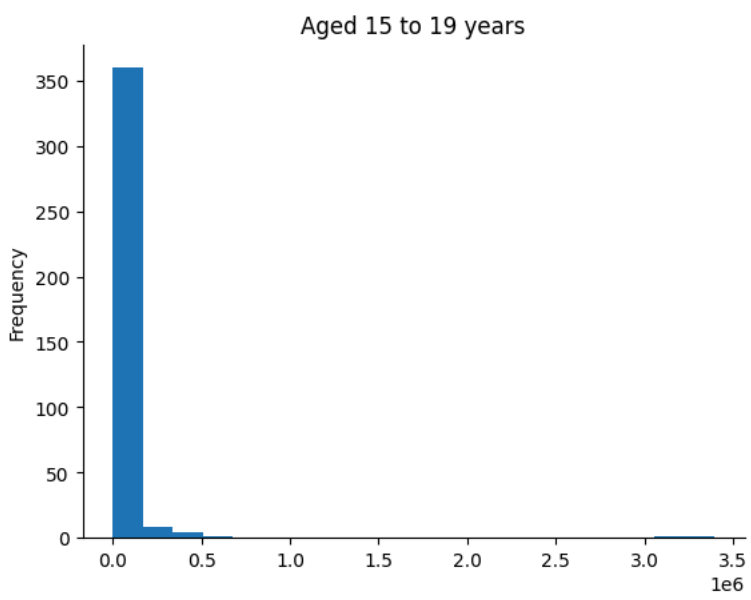
Aged 10 to 14 years

```
# @title Aged 10 to 14 years

sheet2_sorted['Aged 10 to 14 years'].plot(kind='hist', bins=20, title='Aged 10 to 14 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



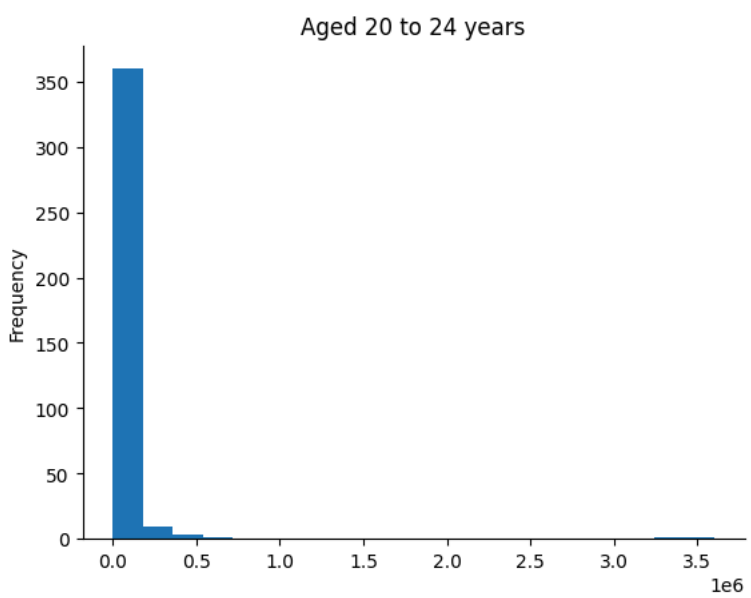
Aged 15 to 19 years

```
# @title Aged 15 to 19 years

sheet2_sorted['Aged 15 to 19 years'].plot(kind='hist', bins=20, title='Aged 15 to 19 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



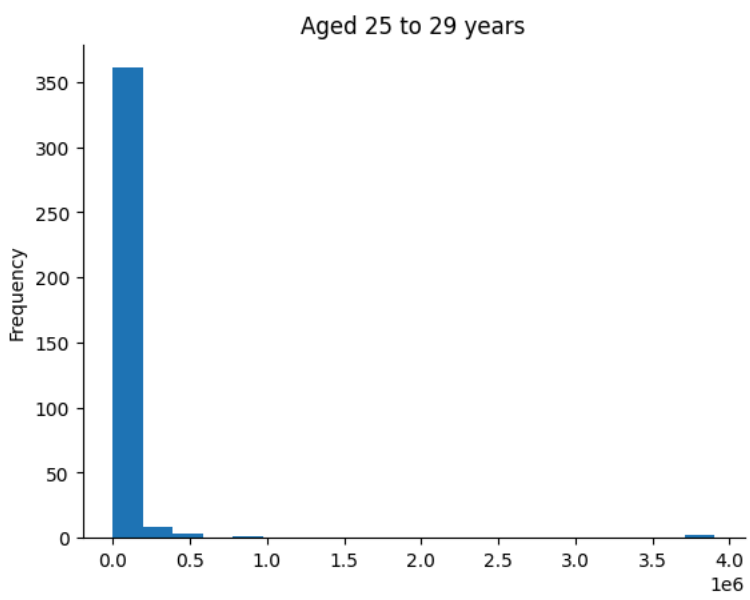
Aged 20 to 24 years

```
# @title Aged 20 to 24 years

sheet2_sorted['Aged 20 to 24 years'].plot(kind='hist', bins=20, title='Aged 20 to 24 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



Aged 25 to 29 years

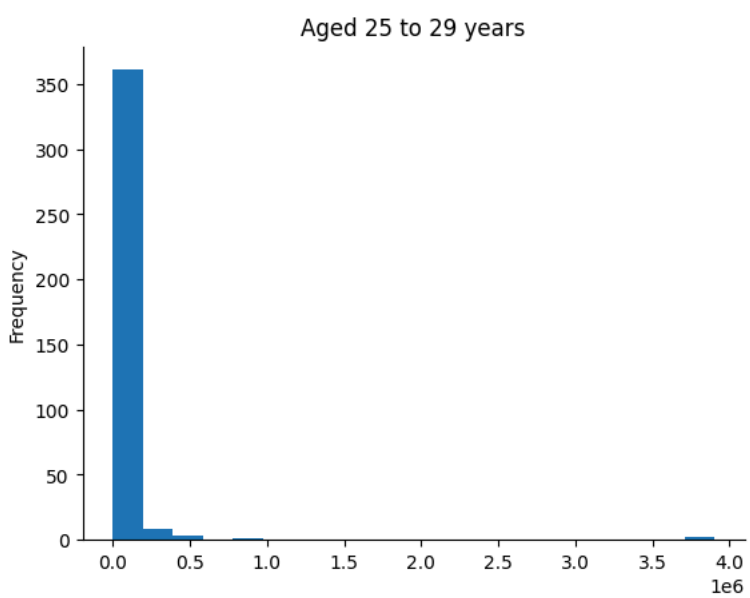
```
# @title Aged 25 to 29 years

sheet2_sorted['Aged 25 to 29 years'].plot(kind='hist', bins=20, title='Aged 25 to 29 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
sheet2_sorted = sheet2.sort_values(by='Area name')
```

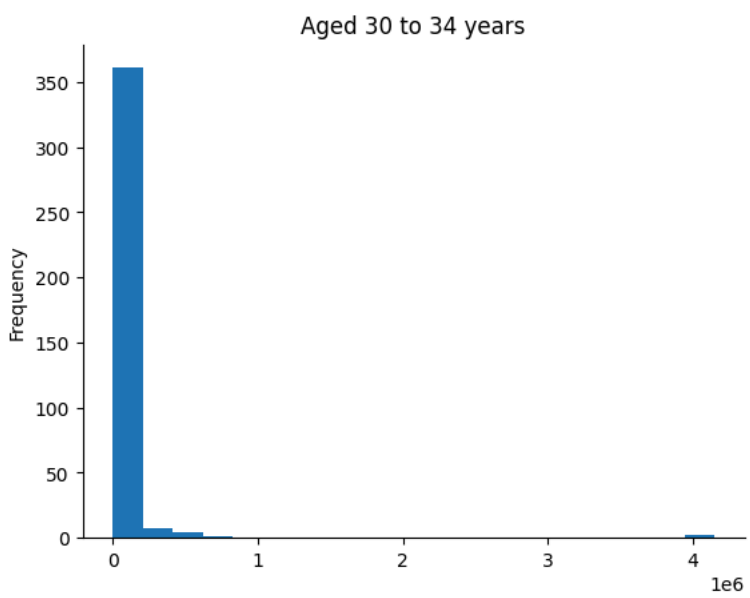
Aged 25 to 29 years

```
# @title Aged 25 to 29 years

sheet2_sorted['Aged 25 to 29 years'].plot(kind='hist', bins=20, title='Aged 25 to 29 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



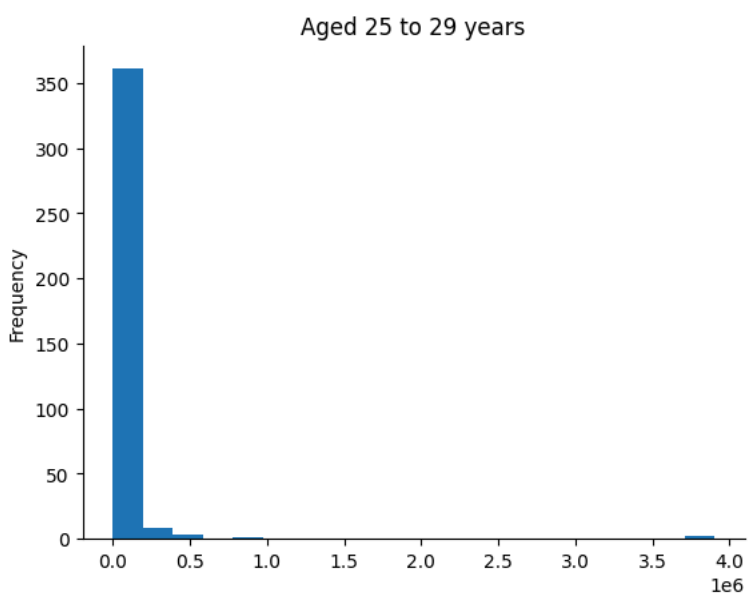
Aged 30 to 34 years

```
# @title Aged 30 to 34 years

sheet2_sorted['Aged 30 to 34 years'].plot(kind='hist', bins=20, title='Aged 30 to 34 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



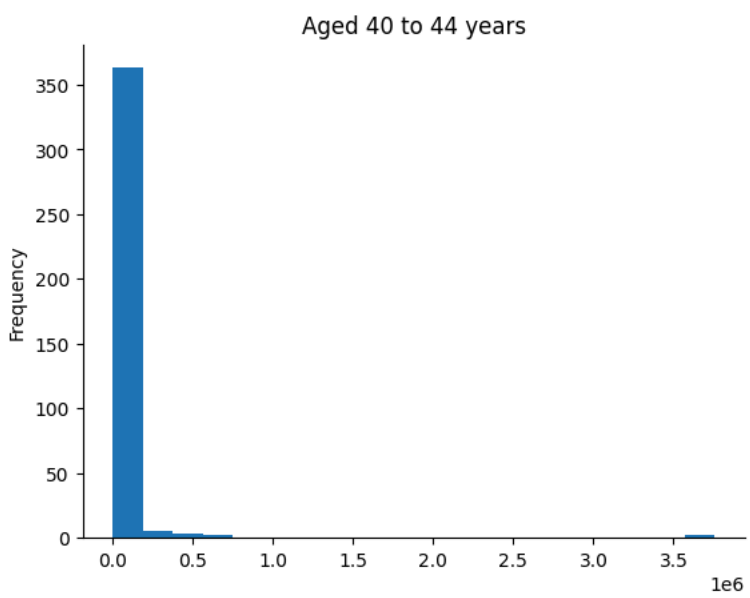
Aged 35 to 39 years

```
# @title Aged 35 to 39 years

sheet2_sorted['Aged 25 to 29 years'].plot(kind='hist', bins=20, title='Aged 25 to 29 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



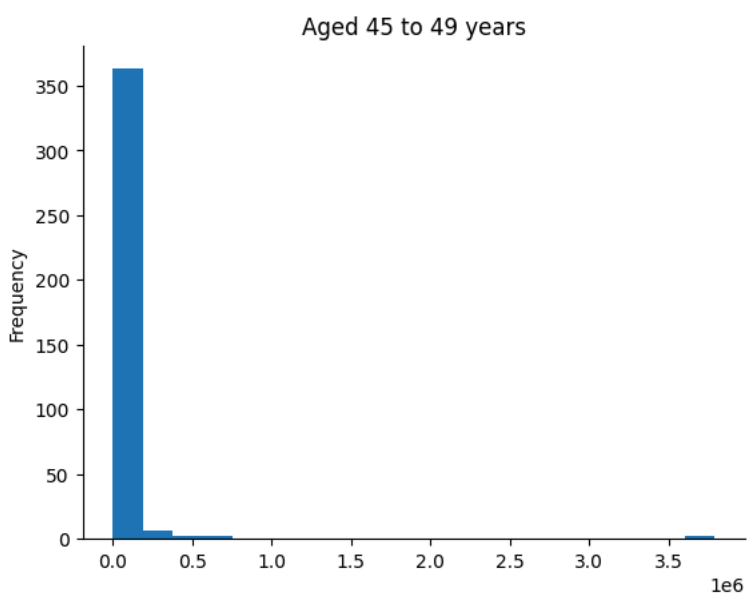
Aged 40 to 44 years

```
# @title Aged 40 to 44 years

sheet2_sorted['Aged 40 to 44 years'].plot(kind='hist', bins=20, title='Aged 40 to 44 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



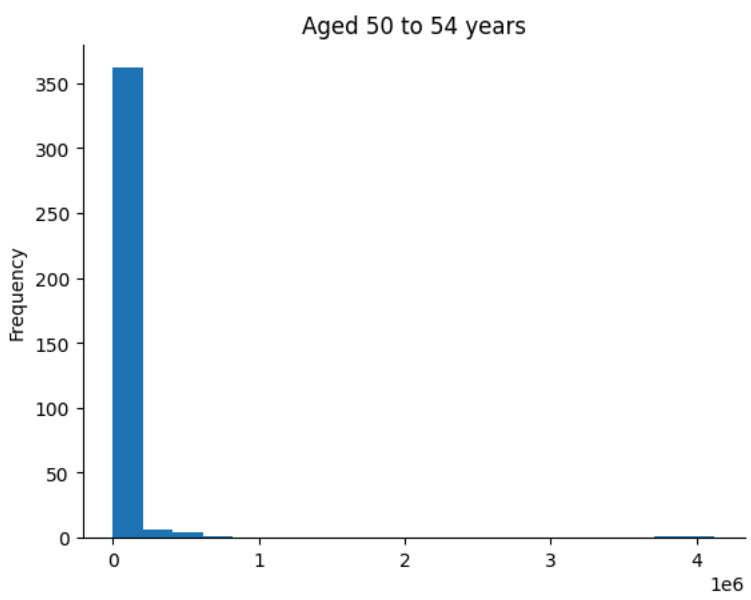
Aged 45 to 49 years

```
# @title Aged 45 to 49 years

sheet2_sorted['Aged 45 to 49 years'].plot(kind='hist', bins=20, title='Aged 45 to 49 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



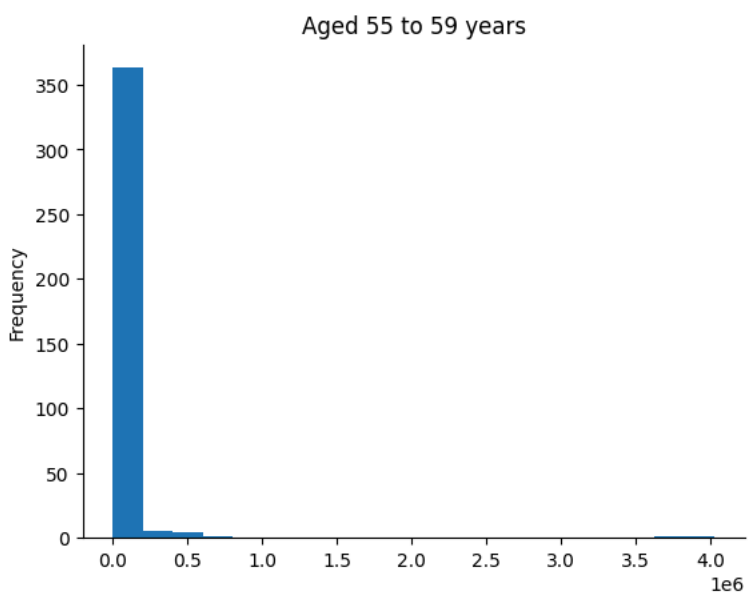
Aged 50 to 54 years

```
# @title Aged 50 to 54 years

sheet2_sorted['Aged 50 to 54 years'].plot(kind='hist', bins=20, title='Aged 50 to 54 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



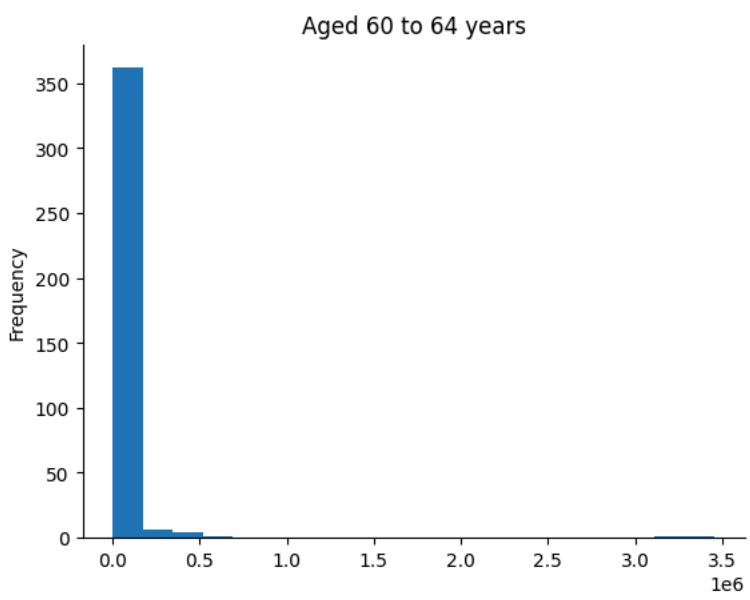
Aged 55 to 59 years

```
# @title Aged 55 to 59 years

sheet2_sorted['Aged 55 to 59 years'].plot(kind='hist', bins=20, title='Aged 55 to 59 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



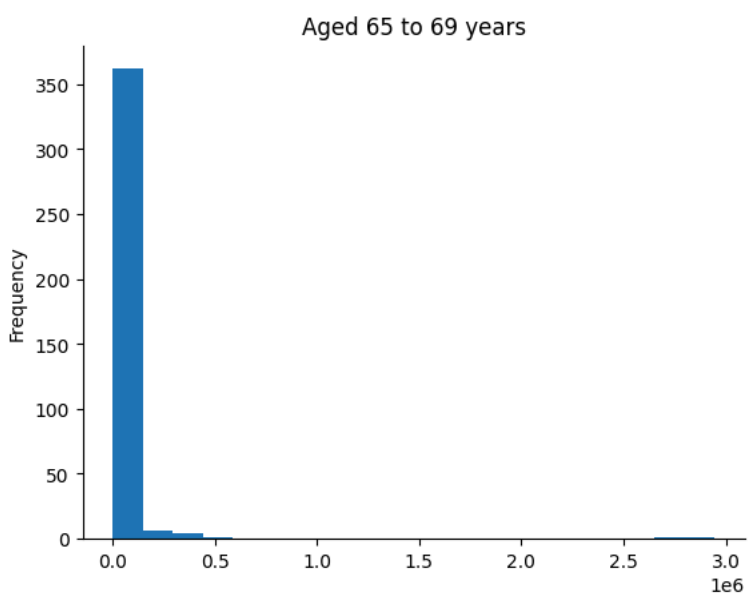
Aged 60 to 64 years

```
# @title Aged 60 to 64 years

sheet2_sorted['Aged 60 to 64 years'].plot(kind='hist', bins=20, title='Aged 60 to 64 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



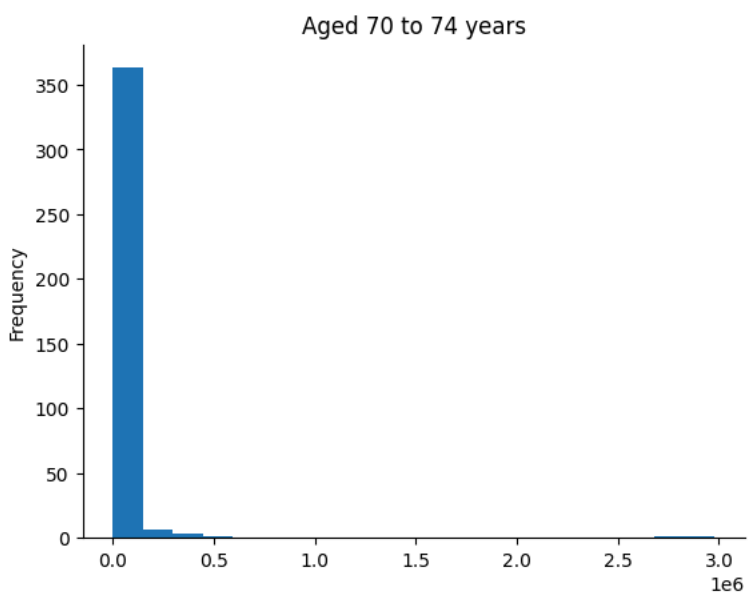
Aged 65 to 69 years

```
# @title Aged 65 to 69 years

sheet2_sorted['Aged 65 to 69 years'].plot(kind='hist', bins=20, title='Aged 65 to 69 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



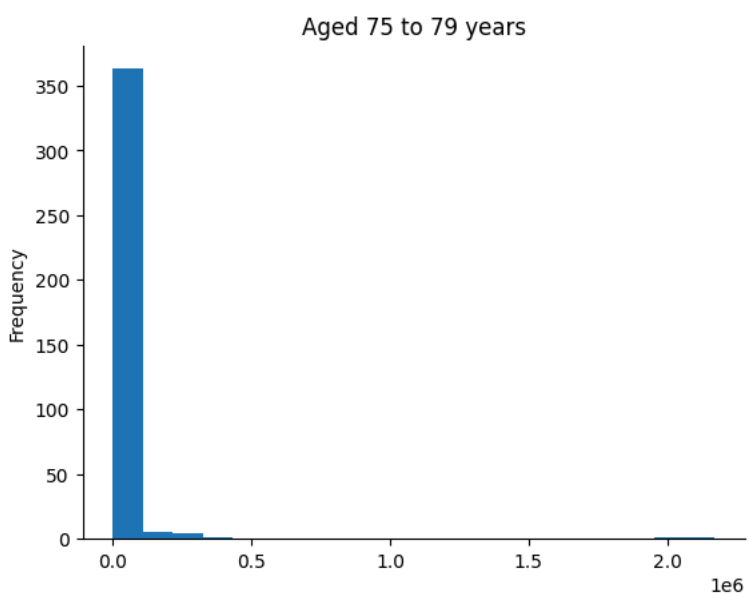
Aged 70 to 74 years

```
# @title Aged 70 to 74 years

sheet2_sorted['Aged 70 to 74 years'].plot(kind='hist', bins=20, title='Aged 70 to 74 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



Aged 75 to 79 years

```
# @title Aged 75 to 79 years

sheet2_sorted['Aged 75 to 79 years'].plot(kind='hist', bins=20, title='Aged 75 to 79 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



Aged 80 to 84 years

```
# @title Aged 80 to 84 years

sheet2_sorted['Aged 80 to 84 years'].plot(kind='hist', bins=20, title='Aged 80 to 84 years')
plt.gca().spines[['top', 'right',]].set_visible(False)
```


Aged 80 to 84 years
350
300
250