1. Librerías a utilizar

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

2. Importar csv

```
In [ ]: sales = pd.read_csv('sales_data.csv')
```

3. Visualización simple de las primeras líneas

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

Out[ ]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Country
0	2013-11-26	26	November	2013	19	Youth (<25)	M	Canada
1	2015-11-26	26	November	2015	19	Youth (<25)	M	Canada
2	2014-03-23	23	March	2014	49	Adults (35-64)	M	Australia

4. Cantidad de filas y columnas

```
In [ ]: sales.shape
```

```
Out[ ]: (113036, 18)
```

5. Información de las columnas

```
In [ ]: sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 113036 entries, 0 to 113035
Data columns (total 18 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   Date              113036 non-null  object
 1   Day               113036 non-null  int64 
 2   Month             113036 non-null  object
 3   Year              113036 non-null  int64 
 4   Customer_Age      113036 non-null  int64 
 5   Age_Group         113036 non-null  object
 6   Customer_Gender   113036 non-null  object
 7   Country           113036 non-null  object
 8   State             113036 non-null  object
 9   Product_Category  113036 non-null  object
 10  Sub_Category      113036 non-null  object
 11  Product           113036 non-null  object
 12  Order_Quantity    113036 non-null  int64 
 13  Unit_Cost         113036 non-null  int64 
 14  Unit_Price        113036 non-null  int64 
 15  Profit            113036 non-null  int64 
 16  Cost              113036 non-null  int64 
 17  Revenue           113036 non-null  int64 
dtypes: int64(9), object(9)
memory usage: 15.5+ MB
```

6. Datos estadísticos de las columnas con numeros (int - float)

In [ ]:
```
sales.describe()
```

Out[ ]:

	Day	Year	Customer_Age	Order_Quantity	Unit_Cost	
count	113036.000000	113036.000000	113036.000000	113036.000000	113036.000000	11
mean	15.665753	2014.401739	35.919212	11.901660	267.296366	
std	8.781567	1.272510	11.021936	9.561857	549.835483	
min	1.000000	2011.000000	17.000000	1.000000	1.000000	
25%	8.000000	2013.000000	28.000000	2.000000	2.000000	
50%	16.000000	2014.000000	35.000000	10.000000	9.000000	
75%	23.000000	2016.000000	43.000000	20.000000	42.000000	
max	31.000000	2016.000000	87.000000	32.000000	2171.000000	

7 Análisis por columnas

a) Columna 'Unit_Cost'

In [ ]:
```
sales['Unit_Cost'].describe()
```

```
Out[ ]: count    113036.000000
        mean        267.296366
        std         549.835483
        min           1.000000
        25%           2.000000
        50%           9.000000
        75%          42.000000
        max        2171.000000
        Name: Unit_Cost, dtype: float64
```

```
In [ ]: sales['Unit_Cost'].mean()
```

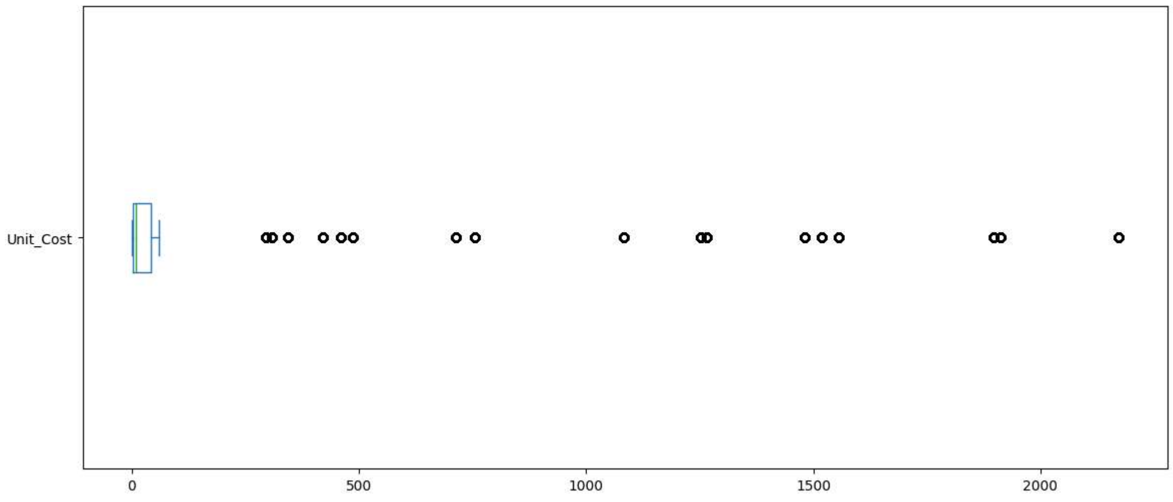
```
Out[ ]: 267.296365759581
```

```
In [ ]: sales['Unit_Cost'].median()
```

```
Out[ ]: 9.0
```

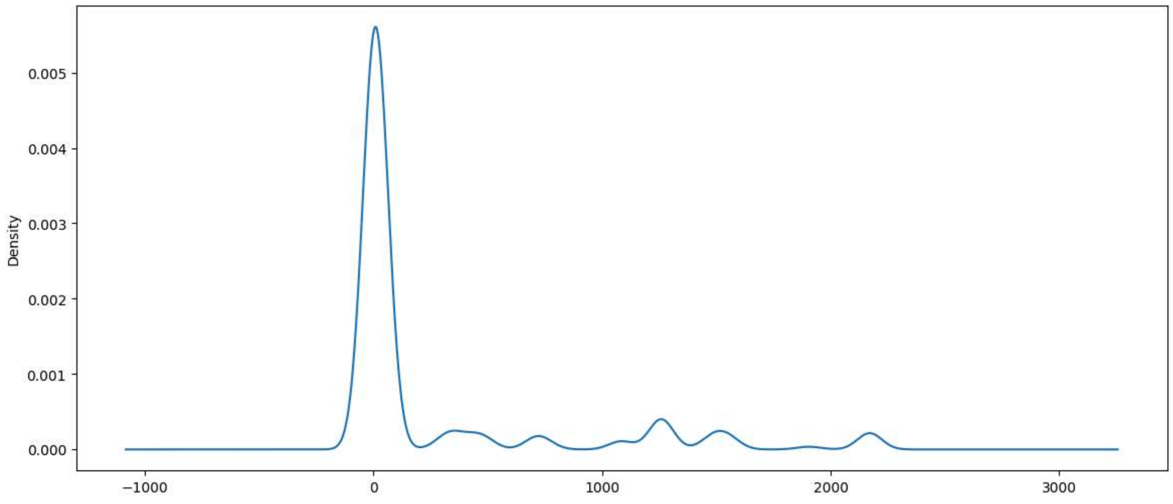
```
In [ ]: sales['Unit_Cost'].plot(kind="box", vert=False, figsize=(14,6))
```

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



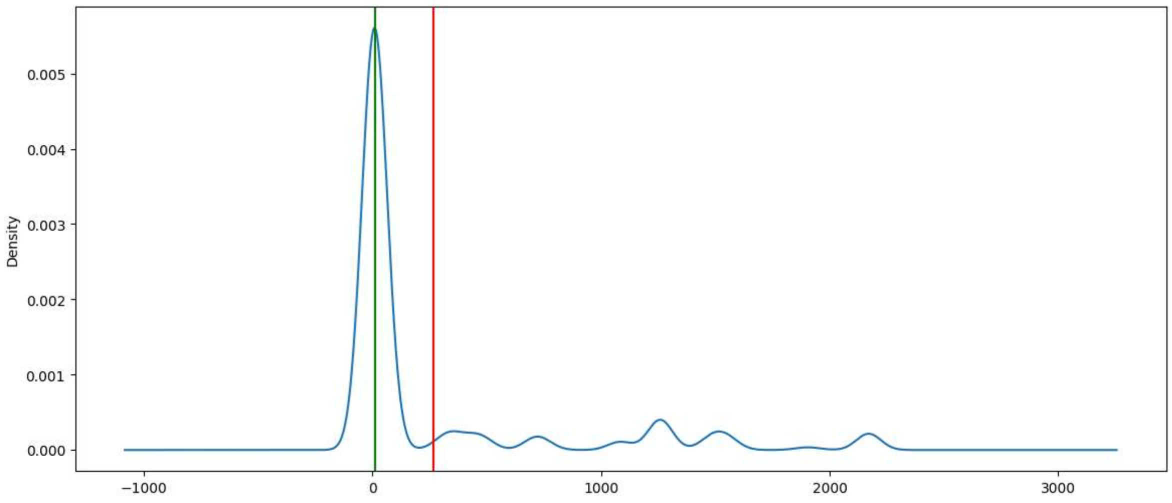
```
In [ ]: sales['Unit_Cost'].plot(kind="density", figsize=(14,6))
```

```
Out[ ]: <Axes: ylabel='Density'>
```



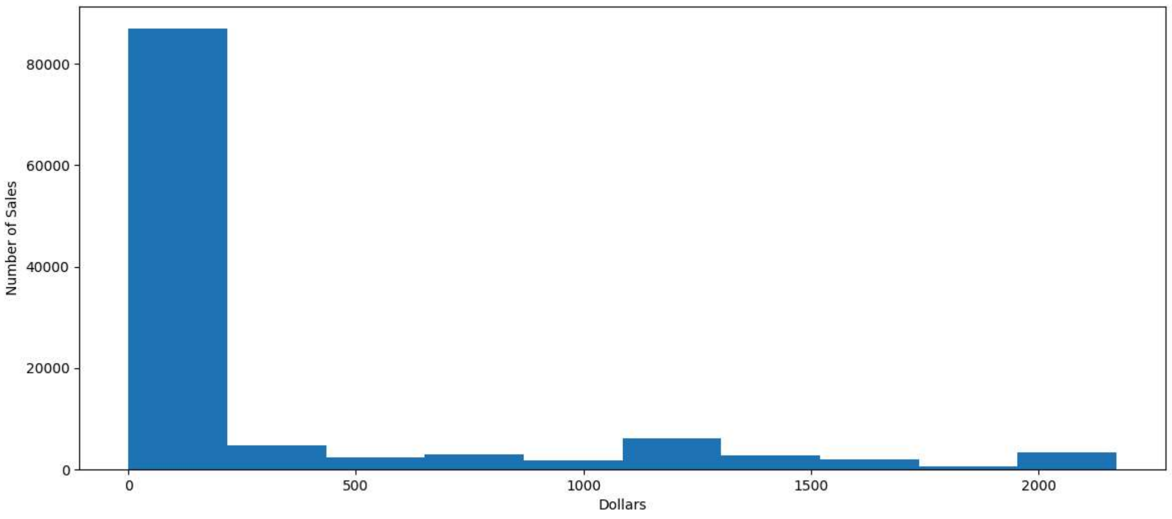
```
In [ ]: ax = sales['Unit_Cost'].plot(kind='density', figsize=(14,6))
        ax.axvline(sales['Unit_Cost'].mean(), color='red')
ax.axvline(sales['Unit_Cost'].median(), color='green')
```

Out[ ]:
```
<matplotlib.lines.Line2D at 0x23605ca68d0>
```



In [ ]:
```
ax = sales['Unit_Cost'].plot(kind='hist', figsize=(14,6))
ax.set_ylabel('Number of Sales')
ax.set_xlabel('Dollars')
```

Out[ ]:
```
Text(0.5, 0, 'Dollars')
```



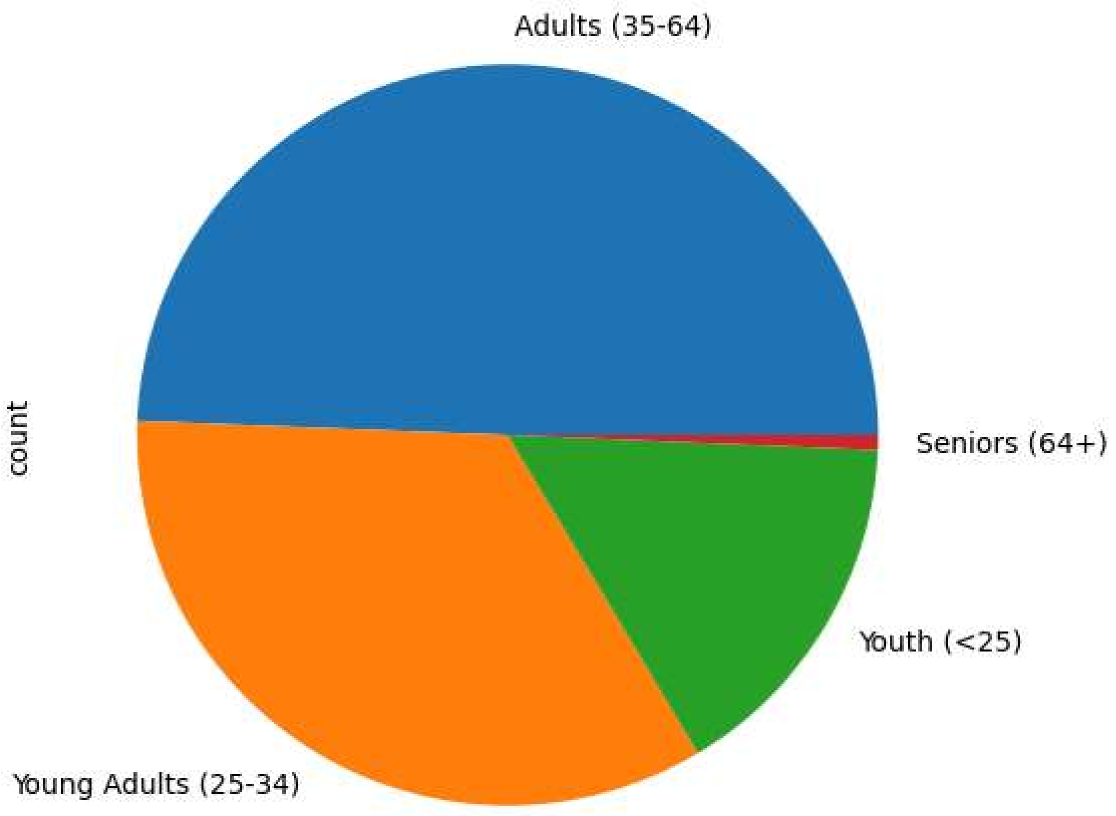
b) Columna 'Age_Group'

In [ ]:
```
sales['Age_Group'].value_counts()
```

Out[ ]:
```
Age_Group
Adults (35-64)          55824
Young Adults (25-34)    38654
Youth (<25)             17828
Seniors (64+)             730
Name: count, dtype: int64
```

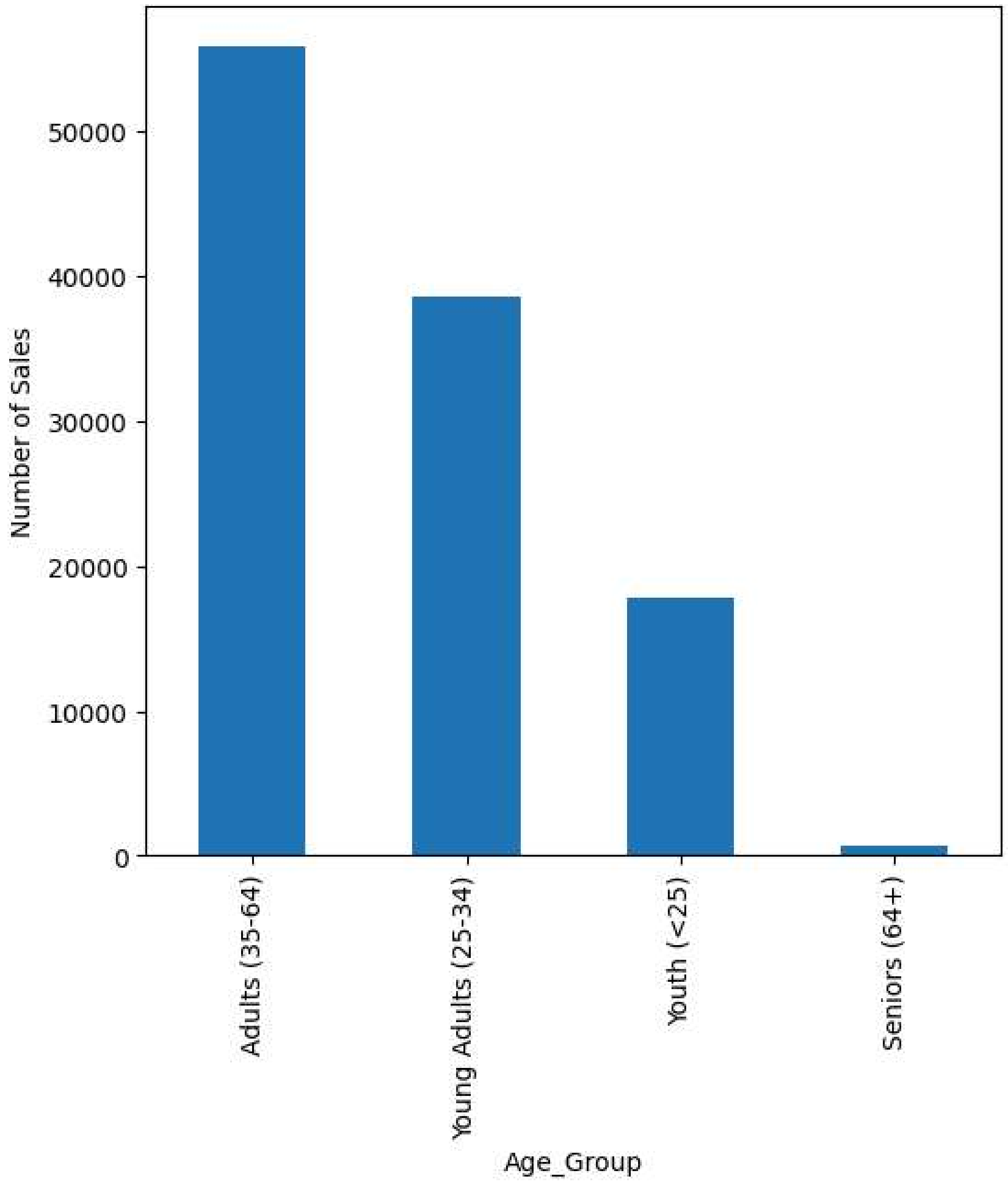
In [ ]:
```
sales['Age_Group'].value_counts().plot(kind='pie', figsize=(6,6))
```

Out[ ]:
```
<Axes: ylabel='count'>
```



```
In [ ]: ax = sales['Age_Group'].value_counts().plot(kind='bar', figsize=(6,6))
        ax.set_ylabel('Number of Sales')
```

```
Out[ ]: Text(0, 0.5, 'Number of Sales')
```



Correlación entre columnas

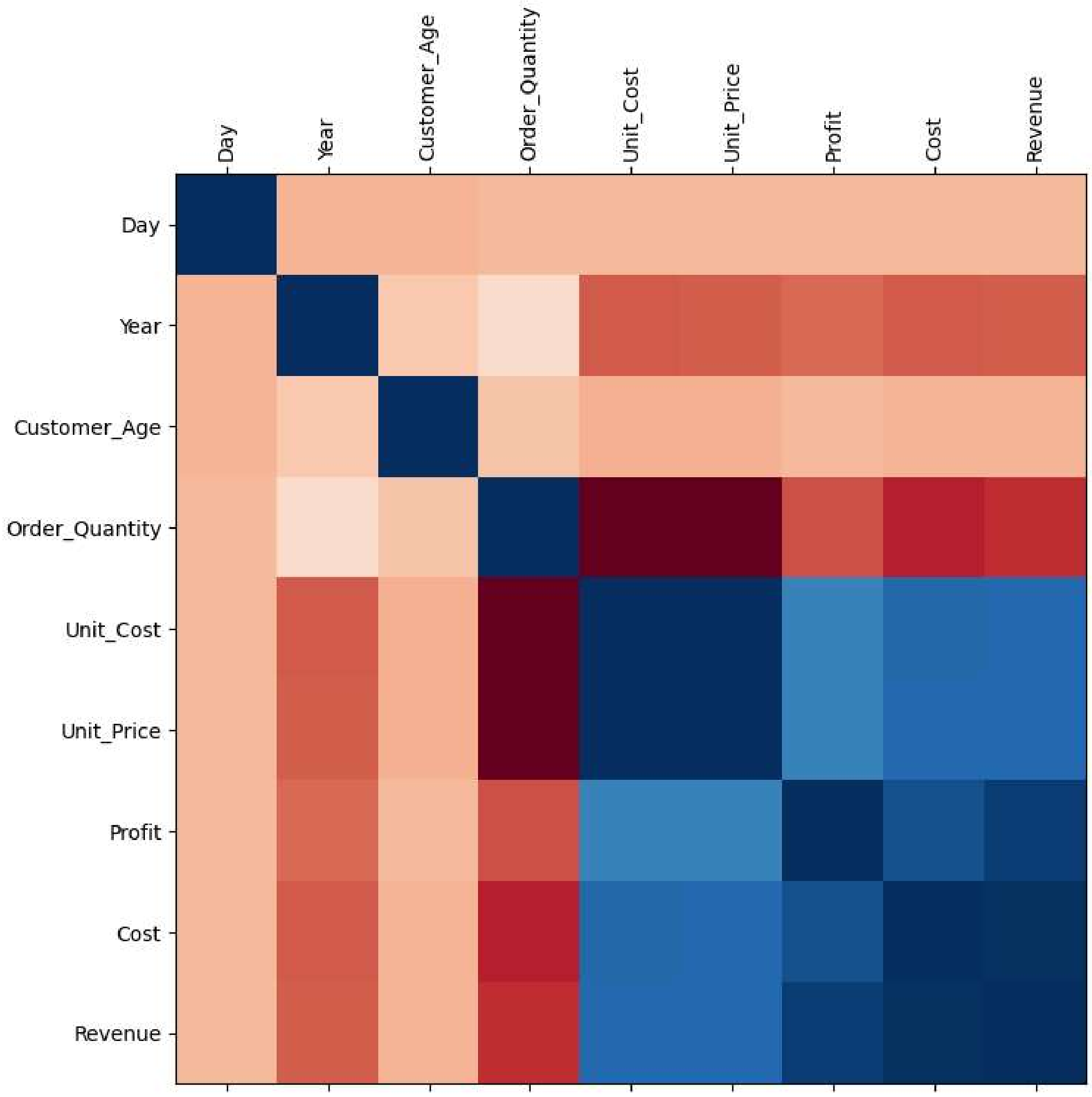
```
In [ ]: sales_num = sales.select_dtypes(include=[np.number])
        corr = sales_num.corr()
        corr
```

Out[ ]:

	Day	Year	Customer_Age	Order_Quantity	Unit_Cost	Unit_Pr
Day	1.000000	-0.007635	-0.014296	-0.002412	0.003133	0.0032
Year	-0.007635	1.000000	0.040994	0.123169	-0.217575	-0.213
Customer_Age	-0.014296	0.040994	1.000000	0.026887	-0.021374	-0.020
Order_Quantity	-0.002412	0.123169	0.026887	1.000000	-0.515835	-0.515
Unit_Cost	0.003133	-0.217575	-0.021374	-0.515835	1.000000	0.997
Unit_Price	0.003207	-0.213673	-0.020262	-0.515925	0.997894	1.000
Profit	0.004623	-0.181525	0.004319	-0.238863	0.741020	0.749
Cost	0.003329	-0.215604	-0.016013	-0.340382	0.829869	0.826
Revenue	0.003853	-0.208673	-0.009326	-0.312895	0.817865	0.818

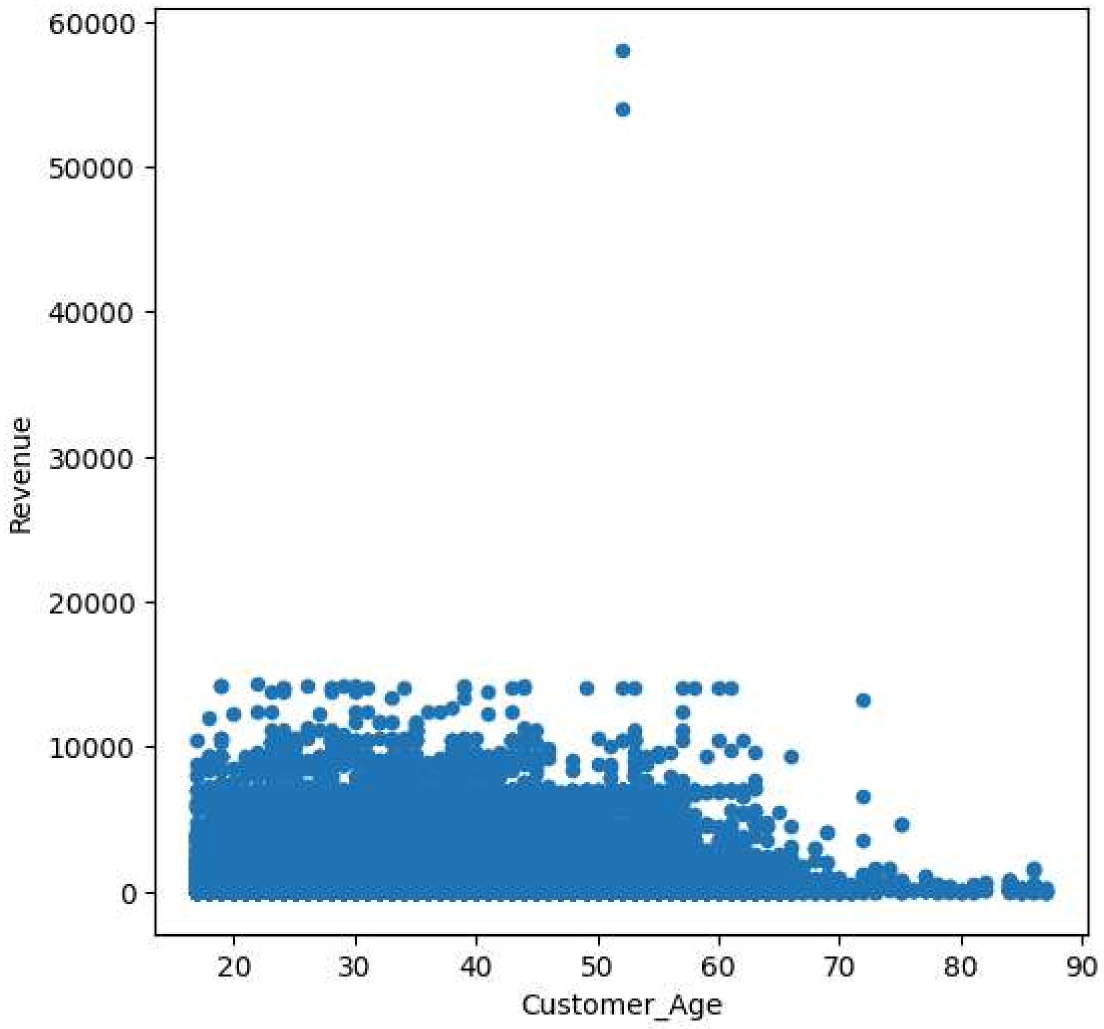
In [ ]:

```python
fig = plt.figure(figsize=(8,8))
plt.matshow(corr, cmap='RdBu', fignum=fig.number)
plt.xticks(range(len(corr.columns)), corr.columns, rotation='vertical')
plt.yticks(range(len(corr.columns)), corr.columns);
```



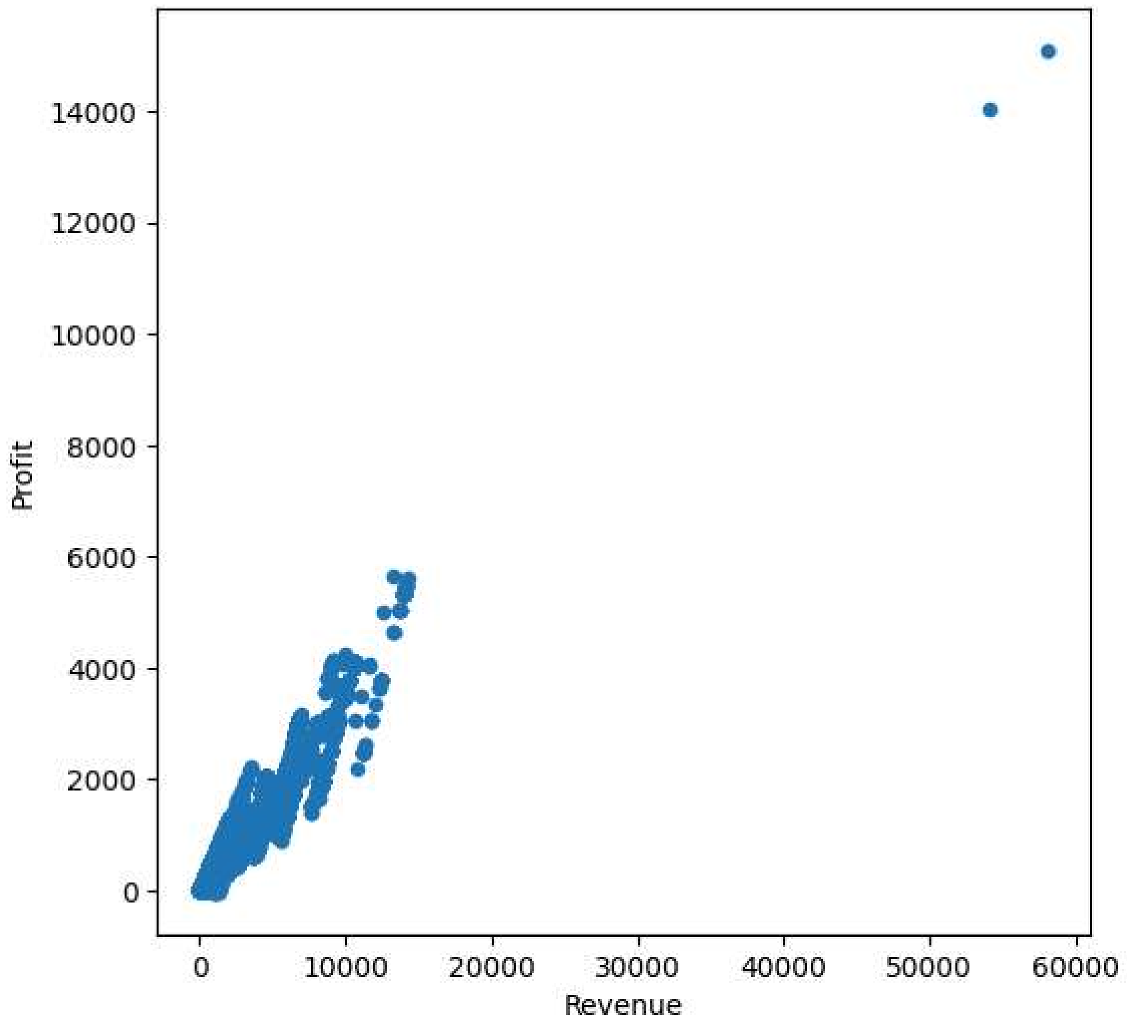
```
In [ ]: sales_num.plot(kind='scatter', x='Customer_Age', y='Revenue', figsize=(6,6))
```

```
Out[ ]: <Axes: xlabel='Customer_Age', ylabel='Revenue'>
```



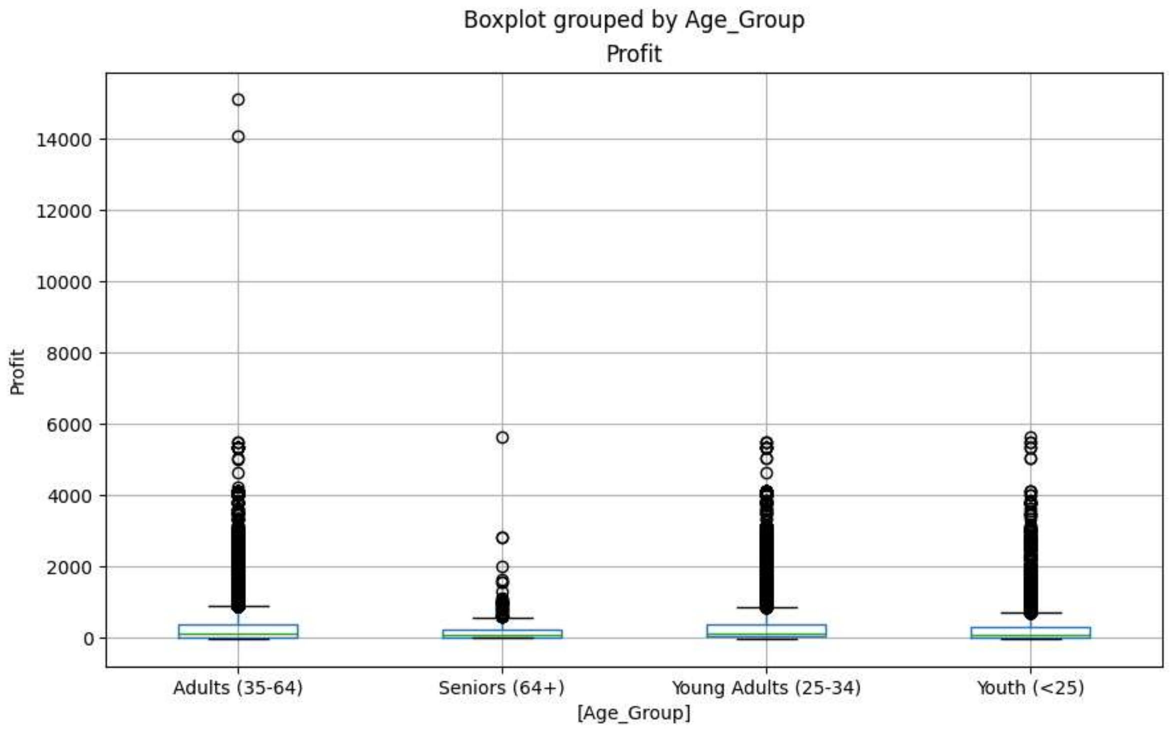
```
In [ ]: sales_num.plot(kind='scatter', x='Revenue', y='Profit', figsize=(6,6))
```

```
Out[ ]: <Axes: xlabel='Revenue', ylabel='Profit'>
```



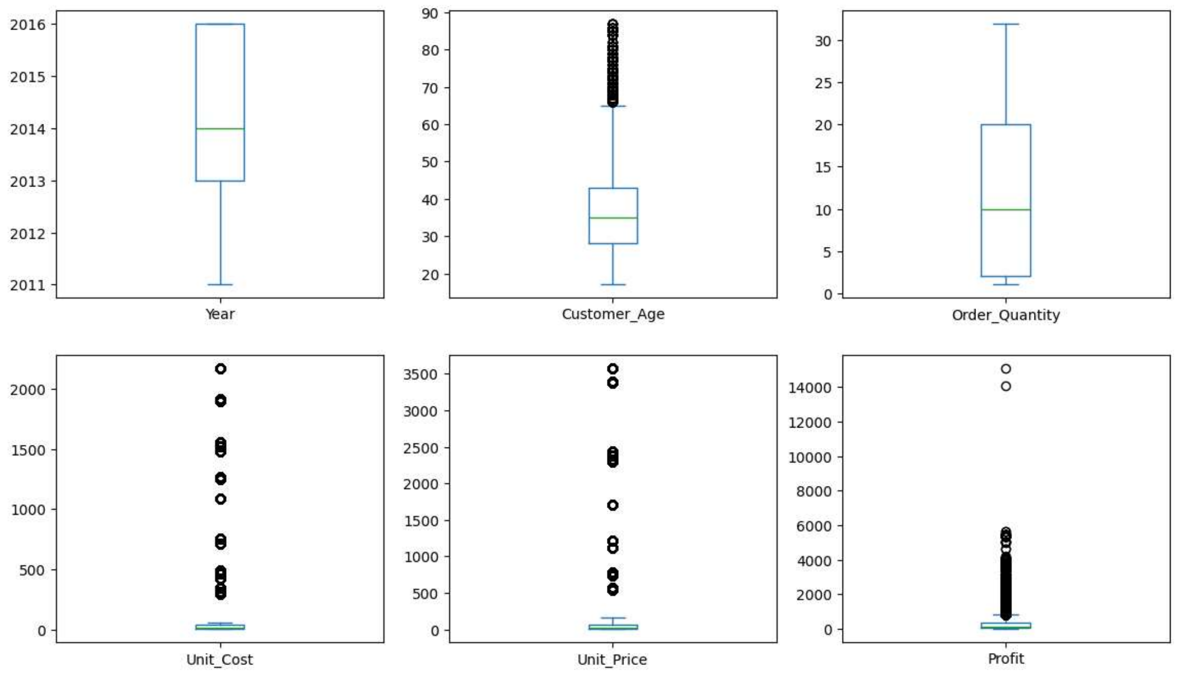
```
In [ ]: ax = sales[['Profit', 'Age_Group']].boxplot(by='Age_Group', figsize=(10,6))
        ax.set_ylabel('Profit')
```

```
Out[ ]: Text(0, 0.5, 'Profit')
```



```
In [ ]: boxplot_cols = ['Year', 'Customer_Age', 'Order_Quantity', 'Unit_Cost', 'Unit_Pri
        sales[boxplot_cols].plot(kind="box", subplots=True, layout=(2,3), figsize=(14,8)
```

```
Out[ ]: Year               Axes(0.125,0.53;0.227941x0.35)
        Customer_Age    Axes(0.398529,0.53;0.227941x0.35)
        Order_Quantity  Axes(0.672059,0.53;0.227941x0.35)
        Unit_Cost          Axes(0.125,0.11;0.227941x0.35)
        Unit_Price      Axes(0.398529,0.11;0.227941x0.35)
        Profit          Axes(0.672059,0.11;0.227941x0.35)
        dtype: object
```



Agregar nuevas columnas

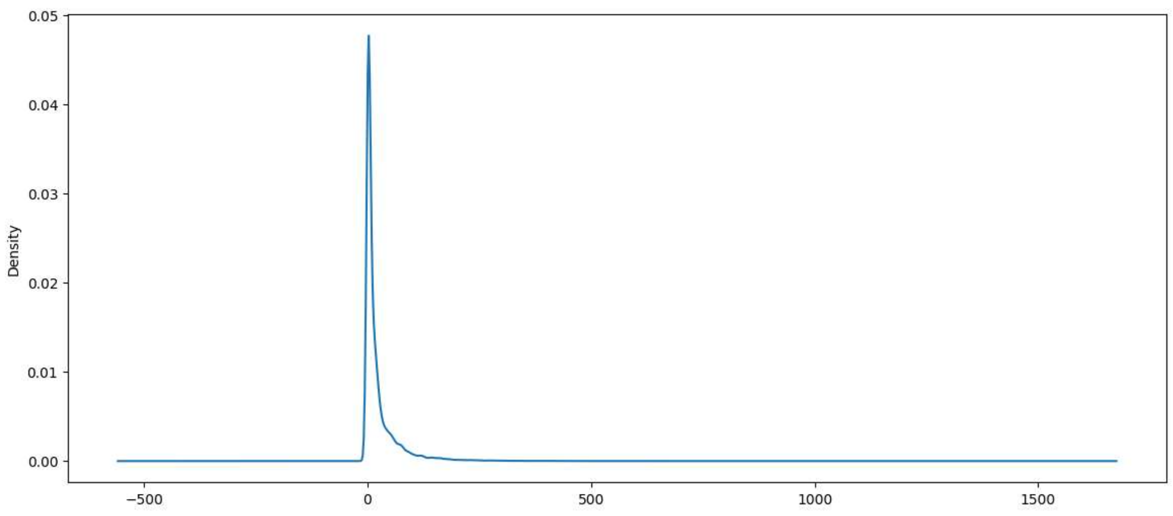
Columna nueva 'Revenue_per_age'

```
In [ ]: sales['Revenue_per_Age'] = sales['Revenue']/sales['Customer_Age']
        sales['Revenue_per_Age'].head()
```

```
Out[ ]: 0    50.000000
        1    50.000000
        2    49.000000
        3    42.612245
        4     8.893617
        Name: Revenue_per_Age, dtype: float64
```

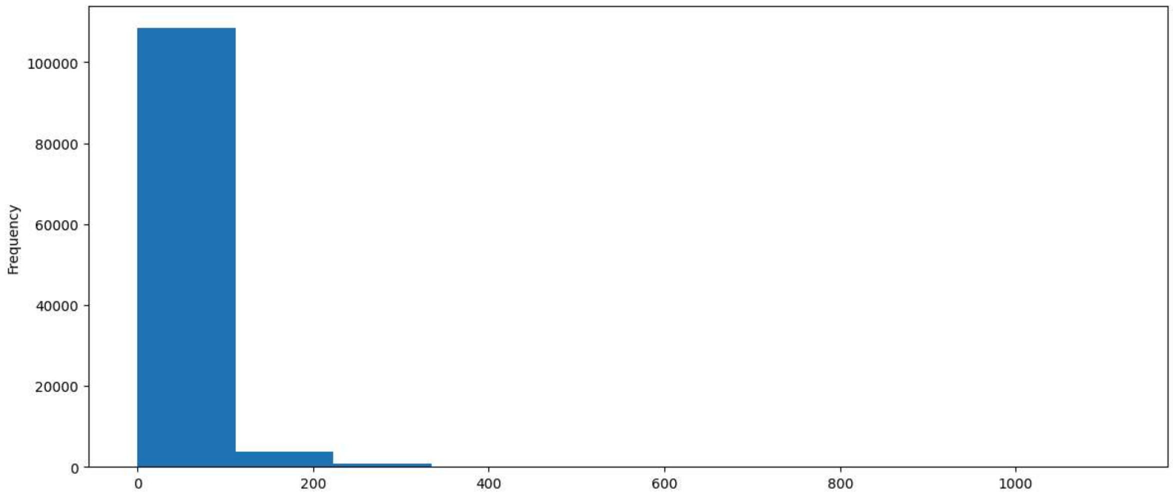
```
In [ ]: sales['Revenue_per_Age'].plot(kind='density', figsize=(14,6))
```

```
Out[ ]: <Axes: ylabel='Density'>
```



```
In [ ]: sales['Revenue_per_Age'].plot(kind='hist', figsize=(14,6))
```

```
Out[ ]: <Axes: ylabel='Frequency'>
```



Columna 'Calculated_Cost'

```
In [ ]: sales['Calculated_Cost'] = sales['Order_Quantity'] * sales['Unit_Cost']
        sales['Calculated_Cost'].head()
```

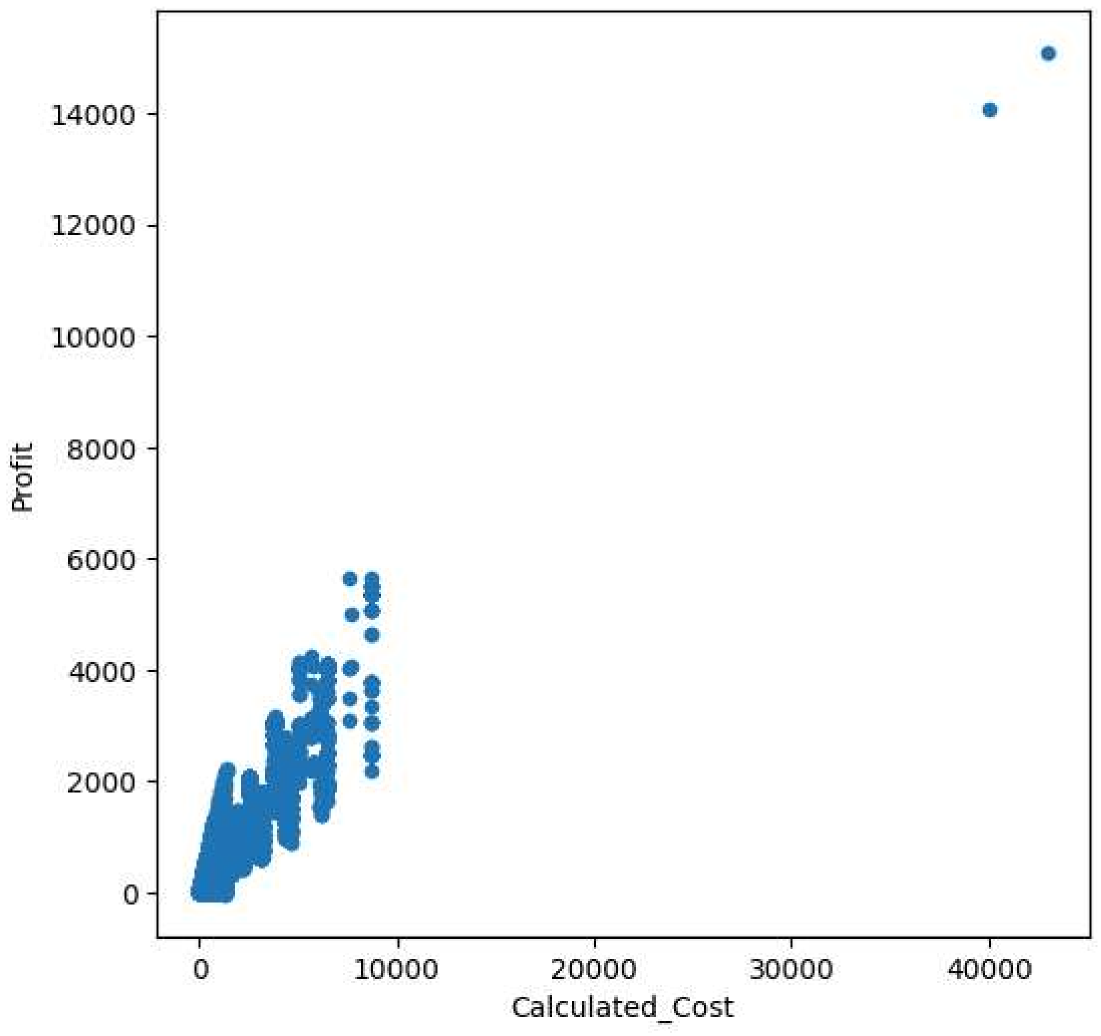
```
Out[ ]: 0     360
        1     360
        2    1035
        3     900
        4     180
        Name: Calculated_Cost, dtype: int64
```

```
In [ ]: (sales['Calculated_Cost'] != sales['Cost']).sum()
```

```
Out[ ]: 0
```

```
In [ ]: sales.plot(kind='scatter', x='Calculated_Cost', y='Profit', figsize=(6,6))
```

```
Out[ ]: <Axes: xlabel='Calculated_Cost', ylabel='Profit'>
```



Columna 'Calculated_Revenue'

```
In [ ]: sales['Calculated_Revenue'] = sales['Cost'] + sales['Profit']
        sales['Calculated_Revenue'].head()
```

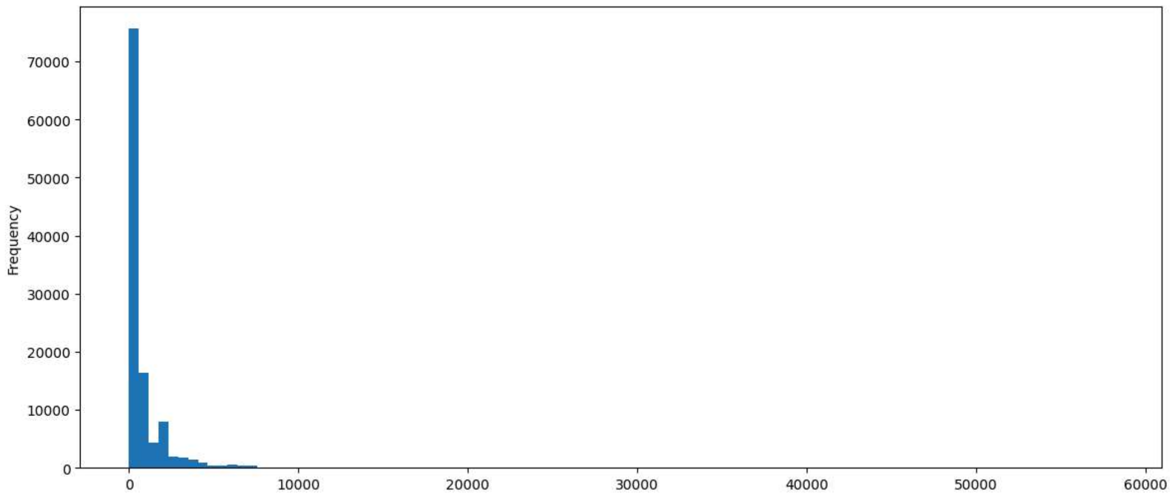
```
Out[ ]: 0     950
        1     950
        2    2401
        3    2088
        4     418
        Name: Calculated_Revenue, dtype: int64
```

```
In [ ]: (sales['Calculated_Revenue'] != sales['Revenue']).sum()
```

```
Out[ ]: 0
```

```
In [ ]: sales['Revenue'].plot(kind='hist', bins=100, figsize=(14,6))
```

```
Out[ ]: <Axes: ylabel='Frequency'>
```



Modificar todos los precios de 'Unit_Price', agregando 3% de taxes a estos

In [ ]:
```
sales['Unit_Price'] *= 1.03
sales['Unit_Price'].head()
```

Out[ ]:
```
0    123.6
1    123.6
2    123.6
3    123.6
4    123.6
Name: Unit_Price, dtype: float64
```

Obtener todas las ventas realizadas en el estado de Kentucky

In [ ]:
```
sales.loc[sales['State'] == 'Kentucky']
```

Out[ ]:

	Date	Day	Month	Year	Customer_Age	Age_Group	Customer_Gender	Co
156	2013-11-04	4	November	2013	40	Adults (35-64)	M	U S
157	2015-11-04	4	November	2015	40	Adults (35-64)	M	U S
23826	2014-04-16	16	April	2014	40	Adults (35-64)	M	U S
23827	2016-04-16	16	April	2016	40	Adults (35-64)	M	U S
31446	2014-04-16	16	April	2014	40	Adults (35-64)	M	U S
31447	2016-04-16	16	April	2016	40	Adults (35-64)	M	U S
79670	2014-04-16	16	April	2014	40	Adults (35-64)	M	U S
79671	2014-04-16	16	April	2014	40	Adults (35-64)	M	U S
79672	2016-04-16	16	April	2016	40	Adults (35-64)	M	U S
79673	2016-04-16	16	April	2016	40	Adults (35-64)	M	U S

10 rows × 21 columns

Obtener el Revenue promedio del grupo Adultos (Adults 35-64)

In [ ]:
```
sales.loc[sales['Age_Group'] == 'Adults (35-64)', 'Revenue'].mean()
```

Out[ ]:
```
762.8287654055604
```

Cuantos records pertenecen al grupo 'Youth (<25)' o 'Adults (35-64)'

In [ ]:
```
sales.loc[(sales['Age_Group'] == 'Youth (<25)') | (sales['Age_Group'] == 'Adults
```

Out[ ]:
```
73652
```

Obtener el Revenue promedio del grupo 'Adults (35-64)' en Estados Unidos

```
In [ ]: sales.loc[(sales['Age_Group'] == 'Adults (35-64)') & (sales['Country'] == 'Unite
```

```
Out[ ]: 726.7260473588342
```

Incrementar el Revenue en 10% por cada venta hecha en Francia

```
In [ ]: sales.loc[sales['Country'] == 'France','Revenue']*=1.1
```

```
C:\Users\lancenterstore\AppData\Local\Temp\ipykernel_11184\3980652493.py:1: Futur
eWarning: Setting an item of incompatible dtype is deprecated and will raise in a
future error of pandas. Value '[ 865.7  865.7 3252.7 ...  473.  1386.  1327.7]' h
as dtype incompatible with int64, please explicitly cast to a compatible dtype fi
rst.
  sales.loc[sales['Country'] == 'France','Revenue']*=1.1
```

```
In [ ]: sales.loc[sales['Country'] == 'France','Revenue'].head()
```

```
Out[ ]: 50     865.7
        51     865.7
        52    3252.7
        53    3136.1
        60     688.6
        Name: Revenue, dtype: float64
```