In [1]:
```
import pandas as pd
```

In [2]:
```
df=pd.read_csv('marketing_data.csv')
df.head()
```

Out[2]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWir
0	1826	1970	Graduation	Divorced	$84,835.00	0	0	6/16/14	0	1
1	1	1961	Graduation	Single	$57,091.00	0	0	6/15/14	0	4
2	10476	1958	Graduation	Married	$67,267.00	0	1	5/13/14	0	1
3	1386	1967	Graduation	Together	$32,474.00	1	1	5/11/14	0	
4	5371	1989	Graduation	Single	$21,474.00	1	0	4/8/14	0	

5 rows × 28 columns

df properties

shape
dtype
columns

df methods

head()
info()
describe()

In [3]:
```
df.columns
```

Out[3]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', ' Income ',
       'Kidhome', 'Teenhome', 'Dt_Customer', 'Recency', 'MntWines',
       'MntFruits', 'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Response', 'Complain', 'Country'],
      dtype='object')
```

In [44]:
```
df.shape
```

Out[44]:
```
(2240, 28)
```

In [ ]:
```
#df.dtypes
```

In [ ]:
```
#df.info()
-Can check the missing values
-DataType issue
```

In [4]:
```
df.columns=df.columns.str.replace(' ','')
```

In [5]:
```
df.columns
```
Out[5]:
```
Index(['ID', 'Year_Birth', 'Education', 'Marital_Status', 'Income', 'Kidhome',
       'Teenhome', 'Dt_Customer', 'Recency', 'MntWines', 'MntFruits',
       'MntMeatProducts', 'MntFishProducts', 'MntSweetProducts',
       'MntGoldProds', 'NumDealsPurchases', 'NumWebPurchases',
       'NumCatalogPurchases', 'NumStorePurchases', 'NumWebVisitsMonth',
       'AcceptedCmp3', 'AcceptedCmp4', 'AcceptedCmp5', 'AcceptedCmp1',
       'AcceptedCmp2', 'Response', 'Complain', 'Country'],
      dtype='object')
```
In [6]:
```
df['Income']=df['Income'].str.replace('$','')
df['Income']=df['Income'].str.replace(',','')
df['Income']=df['Income'].str.replace(' ','')
```
```
C:\Users\Pintoo\AppData\Local\Temp\ipykernel_27608\1892822497.py:1: FutureWarning: The d
efault value of regex will change from True to False in a future version. In addition, s
ingle character regular expressions will *not* be treated as literal strings when regex=
True.
  df['Income']=df['Income'].str.replace('$','')
```
In [7]:
```
df.head()
```
Out[7]:

	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome	Dt_Customer	Recency	MntWine
0	1826	1970	Graduation	Divorced	84835.00	0	0	6/16/14	0	18
1	1	1961	Graduation	Single	57091.00	0	0	6/15/14	0	46
2	10476	1958	Graduation	Married	67267.00	0	1	5/13/14	0	13
3	1386	1967	Graduation	Together	32474.00	1	1	5/11/14	0	1
4	5371	1989	Graduation	Single	21474.00	1	0	4/8/14	0	

5 rows × 28 columns

In [8]:
```
df['Income']
```
Out[8]:
```
0       84835.00
1       57091.00
2       67267.00
3       32474.00
4       21474.00
          ...
2235    66476.00
2236    31056.00
2237    46310.00
2238    65819.00
2239    94871.00
Name: Income, Length: 2240, dtype: object
```
In [9]:
```
df['Income']=df['Income'].astype('float')
```

Checking misssing values

1) fixing the column names

2) fixing the dtypes of each column

3) check missing values

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
ID                      0
Year_Birth              0
Education               0
Marital_Status          0
Income                 24
Kidhome                 0
Teenhome                0
Dt_Customer             0
Recency                 0
MntWines                0
MntFruits               0
MntMeatProducts         0
MntFishProducts         0
MntSweetProducts        0
MntGoldProds            0
NumDealsPurchases       0
NumWebPurchases         0
NumCatalogPurchases     0
NumStorePurchases       0
NumWebVisitsMonth       0
AcceptedCmp3            0
AcceptedCmp4            0
AcceptedCmp5            0
AcceptedCmp1            0
AcceptedCmp2            0
Response                0
Complain                0
Country                 0
dtype: int64
```

In [11]:
```
# Analaysing Income Feature
```

NON VIZ

MIN
MAX
CENTRAL TENDENCY
DISPERSION
PERCENTILE

VIZ

PLOT OF DISTRIBUTION(DIST)
PLOT OF BOXPLOT(OUTLIER)

In [12]:
```
df['Income']
```

Out[12]:
```
0       84835.0
1       57091.0
2       67267.0
3       32474.0
4       21474.0
          ...
2235    66476.0
2236    31056.0
```

```
2237    46310.0
2238    65819.0
2239    94871.0
Name: Income, Length: 2240, dtype: float64
```

In [13]:
```
df['Income'].min()
```

Out[13]:
```
1730.0
```

In [14]:
```
df['Income'].max()
```

Out[14]:
```
666666.0
```

In [15]:
```
df['Income'].mean()
```

Out[15]:
```
52247.25135379061
```

In [16]:
```
df['Income'].median()
```

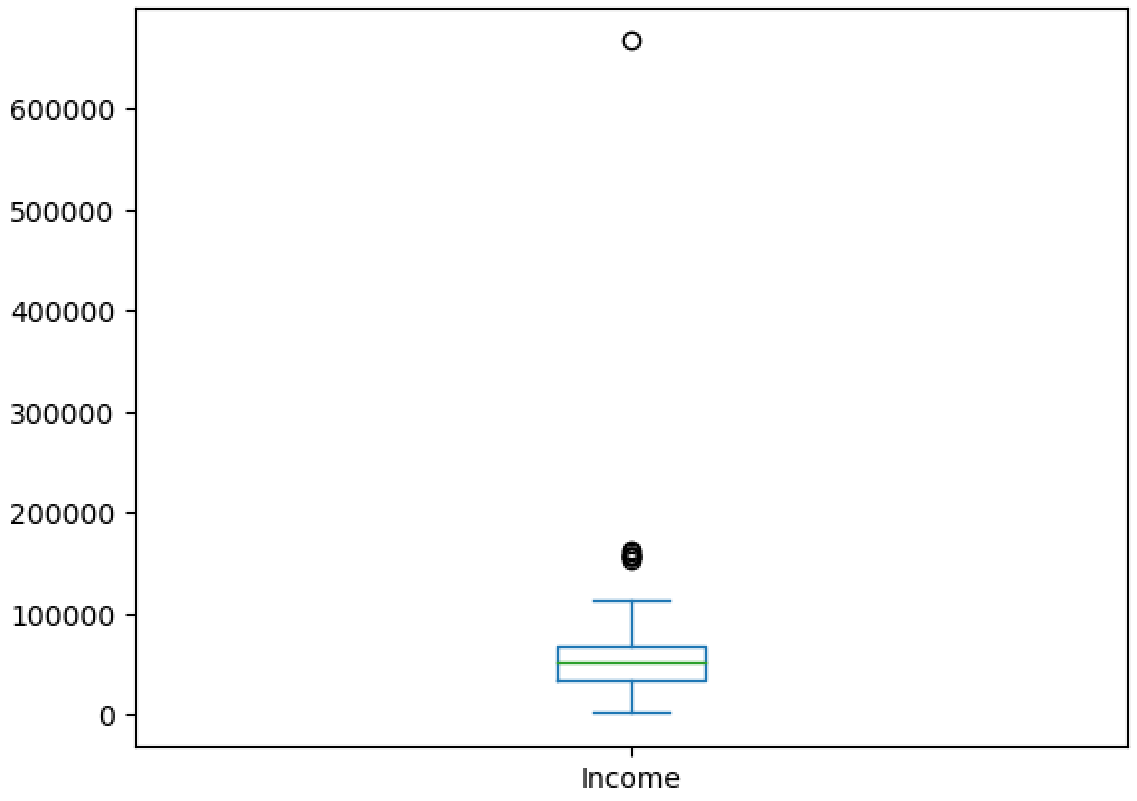
Out[16]:
```
51381.5
```

In [17]:
```
df['Income'].std()
```

Out[17]:
```
25173.0766609014
```

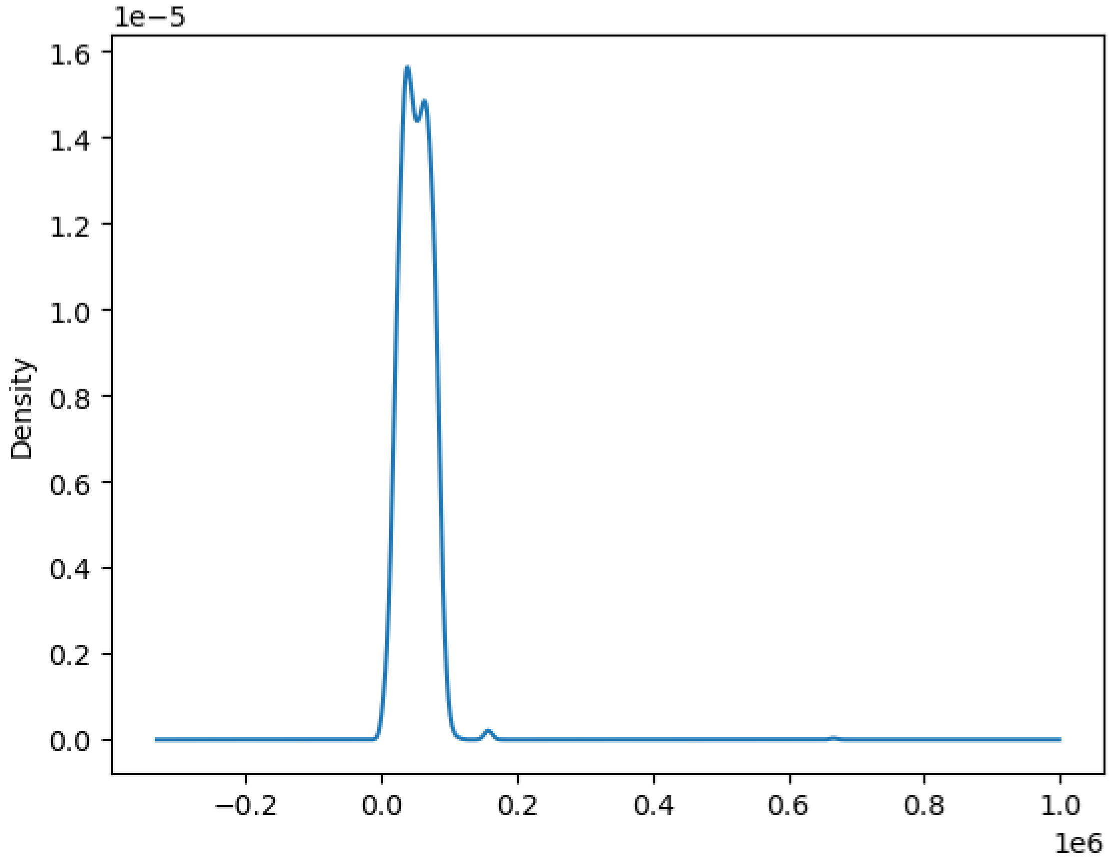
In [18]:
```
df['Income'].plot(kind='box')
```

Out[18]:
```
<AxesSubplot:>
```



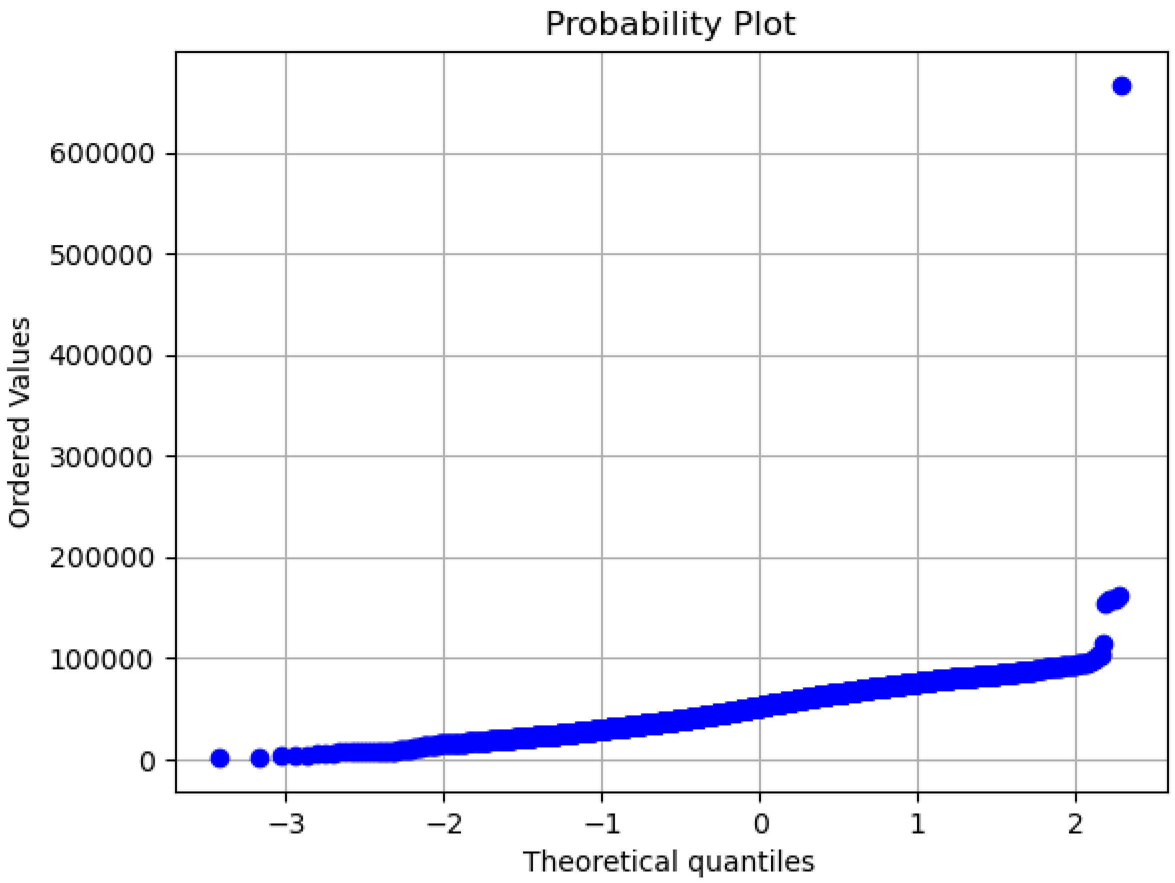
In [19]:
```
df['Income'].plot(kind='kde')
```

Out[19]:
```
<AxesSubplot:ylabel='Density'>
```



```
In [20]: from scipy import stats
         import matplotlib.pyplot as plt
```

```
In [21]: stats.probplot(df['Income'],dist='norm',plot=plt)
         plt.grid()
```



```
In [22]: #Not a normal dist
```

In [23]:
```
df['Income']=df['Income'].fillna(df['Income'].median())
```

In [24]:
```
df['Income'].fillna(df['Income'].median()).isnull().sum()
```

Out[24]:
```
0
```

In [25]:
```
clean_df = df[df['Income'] < 500000]
```

In [26]:
```
print(clean_df.shape)
print(df.shape)
```

```
(2239, 28)
(2240, 28)
```

In [28]:
```
clean_df.dtypes
df.dtypes
```

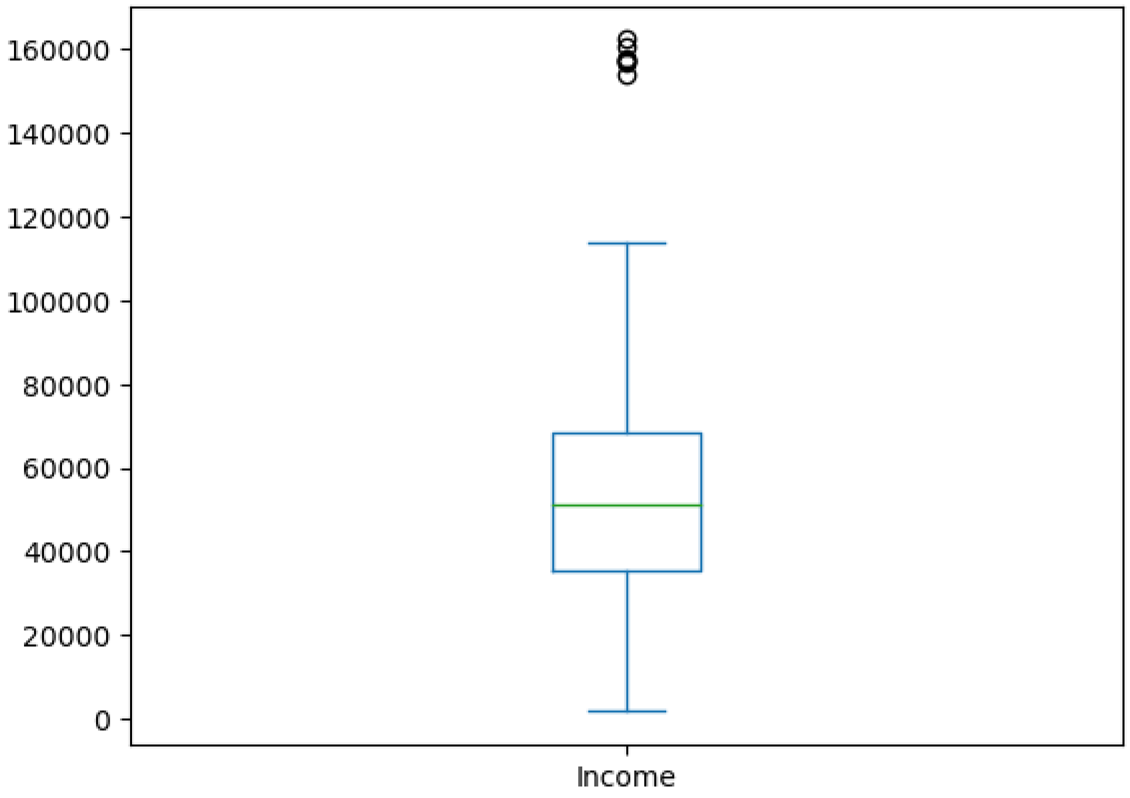
Out[28]:
```
ID                        int64
Year_Birth                int64
Education                object
Marital_Status           object
Income                  float64
Kidhome                   int64
Teenhome                  int64
Dt_Customer              object
Recency                   int64
MntWines                  int64
MntFruits                 int64
MntMeatProducts           int64
MntFishProducts           int64
MntSweetProducts          int64
MntGoldProds              int64
NumDealsPurchases         int64
NumWebPurchases           int64
NumCatalogPurchases       int64
NumStorePurchases         int64
NumWebVisitsMonth         int64
AcceptedCmp3              int64
AcceptedCmp4              int64
AcceptedCmp5              int64
AcceptedCmp1              int64
AcceptedCmp2              int64
Response                  int64
Complain                  int64
Country                  object
dtype: object
```

In [29]:
```
print(clean_df['Income'].min())
print(clean_df['Income'].max())
print(clean_df['Income'].mean())
print(clean_df['Income'].median())
print(clean_df['Income'].std())
```

```
1730.0
162397.0
51963.55471192497
51381.5
21410.672115542126
```

In [30]:
```
clean_df['Income'].plot(kind='box')
```

Out[30]:
```
<AxesSubplot:>
```



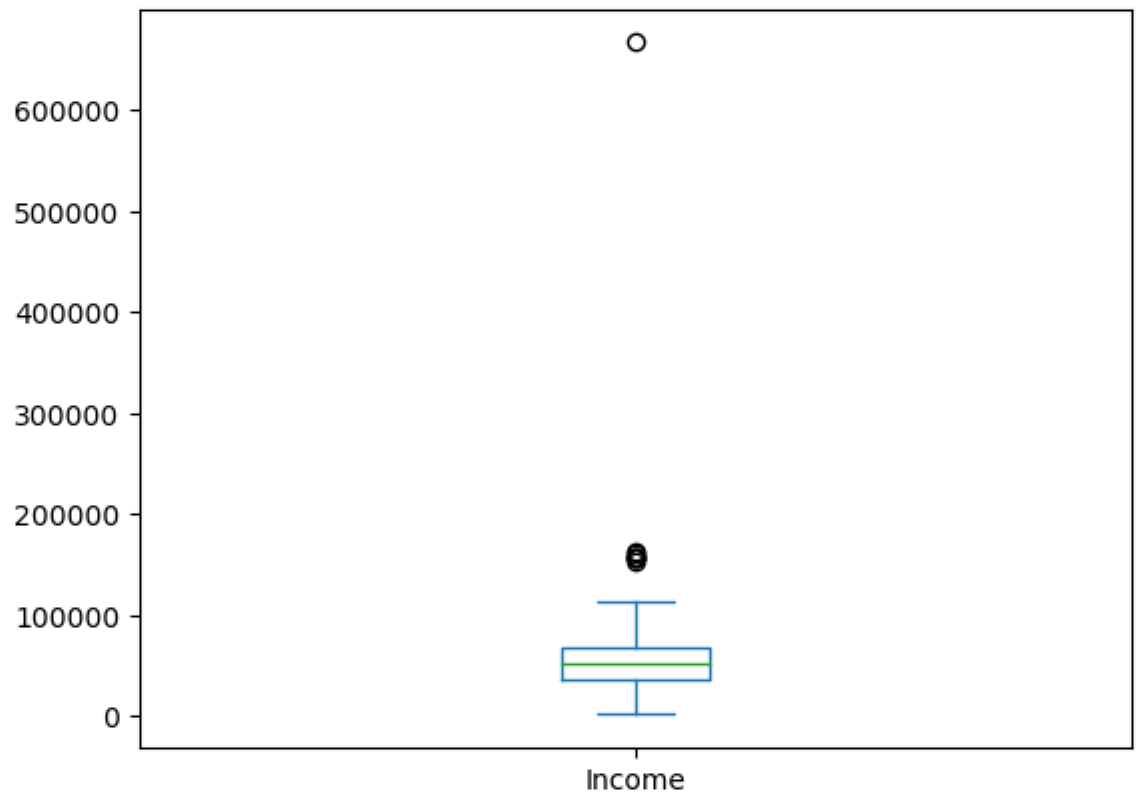
```
In [31]: q1=df['Income'].quantile(0.25)
         q3=df['Income'].quantile(0.75)
         IQR=q3-q1
```

```
In [32]: Income_lower_boundary=q1-1.5*IQR
         Income_upper_boundary=q3+1.5*IQR
```

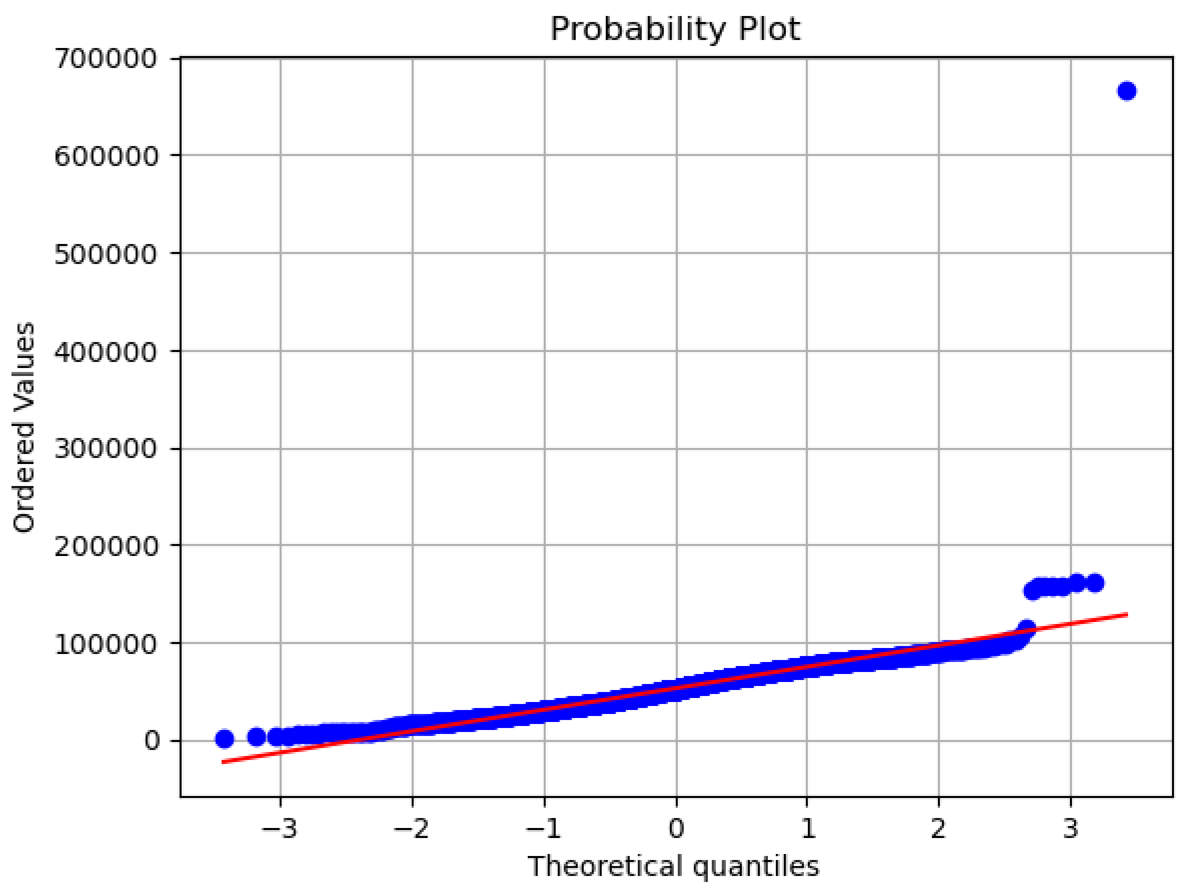
```
In [33]: clean_df=df[(df['Income']>Income_lower_boundary) & df['Income']<Income_upper_boundary]
```

```
In [34]: clean_df['Income'].plot(kind='box')
```

```
Out[34]: <AxesSubplot:>
```



In [35]:
```
stats.probplot(clean_df['Income'],dist='norm',plot=plt)
plt.grid()
```



In [36]:
```
# Categorical Univariate
```

In [37]:
```
clean_df['Education'].mode()
```

Out[37]:
```
0    Graduation
Name: Education, dtype: object
```

```
In [38]: clean_df['Education'].value_counts()
```

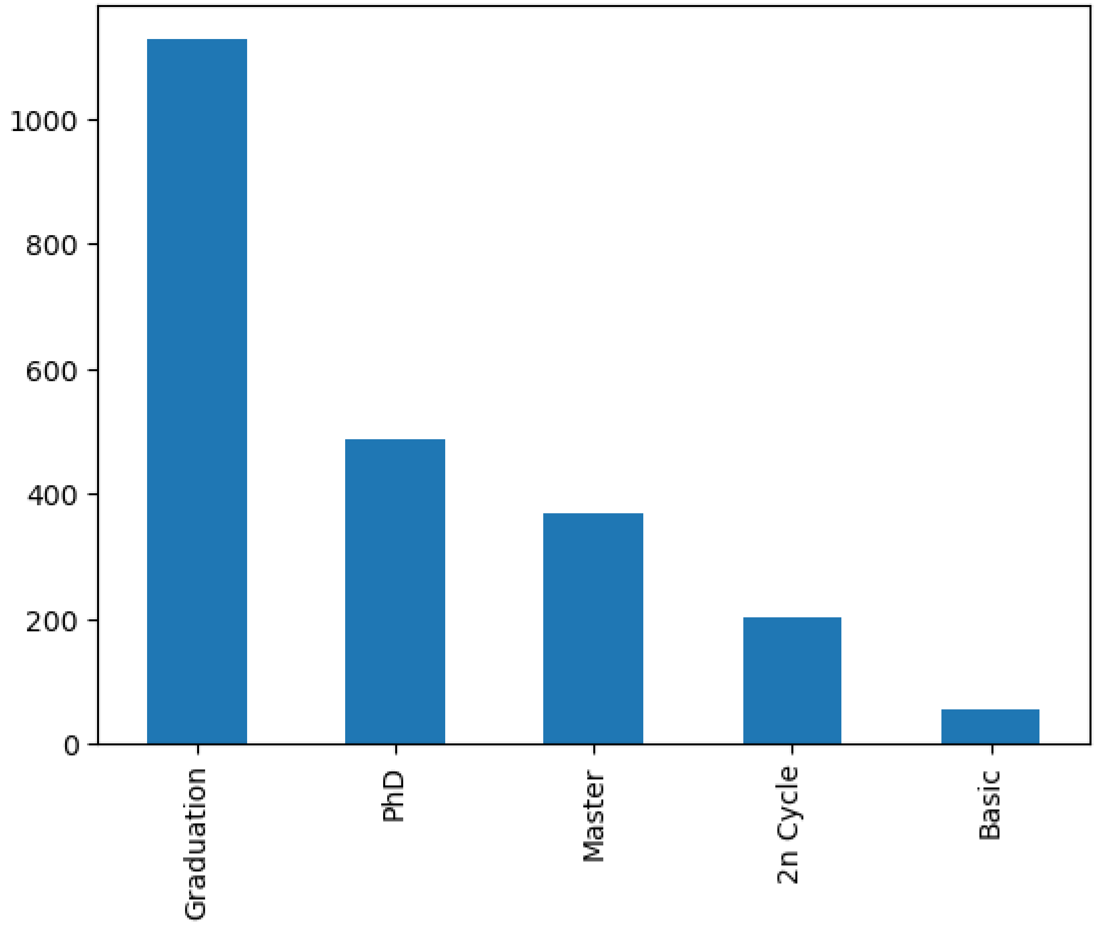
```
Out[38]: Graduation    1127
         PhD            486
         Master         370
         2n Cycle       203
         Basic           54
         Name: Education, dtype: int64
```

```
In [39]: clean_df['Education'].value_counts(normalize=True)
```

```
Out[39]: Graduation    0.503125
         PhD           0.216964
         Master        0.165179
         2n Cycle      0.090625
         Basic         0.024107
         Name: Education, dtype: float64
```

```
In [40]: clean_df['Education'].value_counts().plot(kind='bar')
```

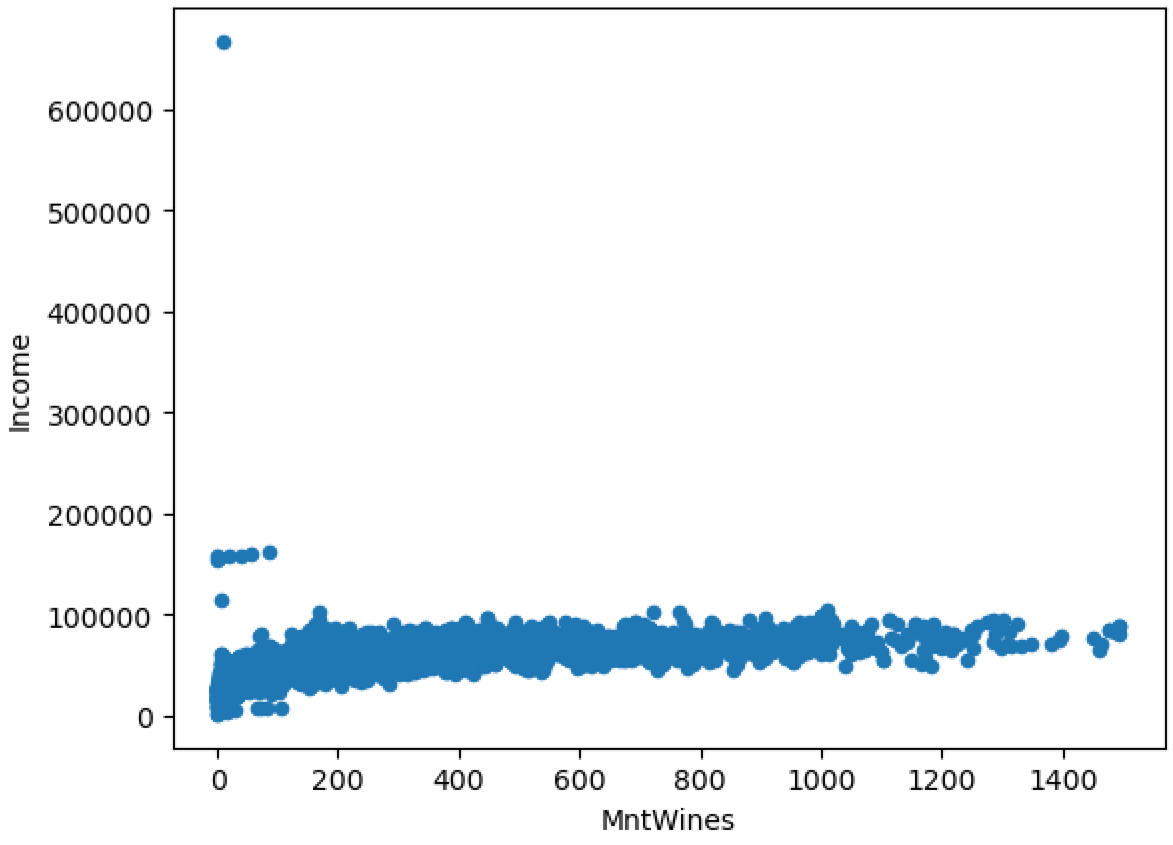
```
Out[40]: <AxesSubplot:>
```



Bivariate

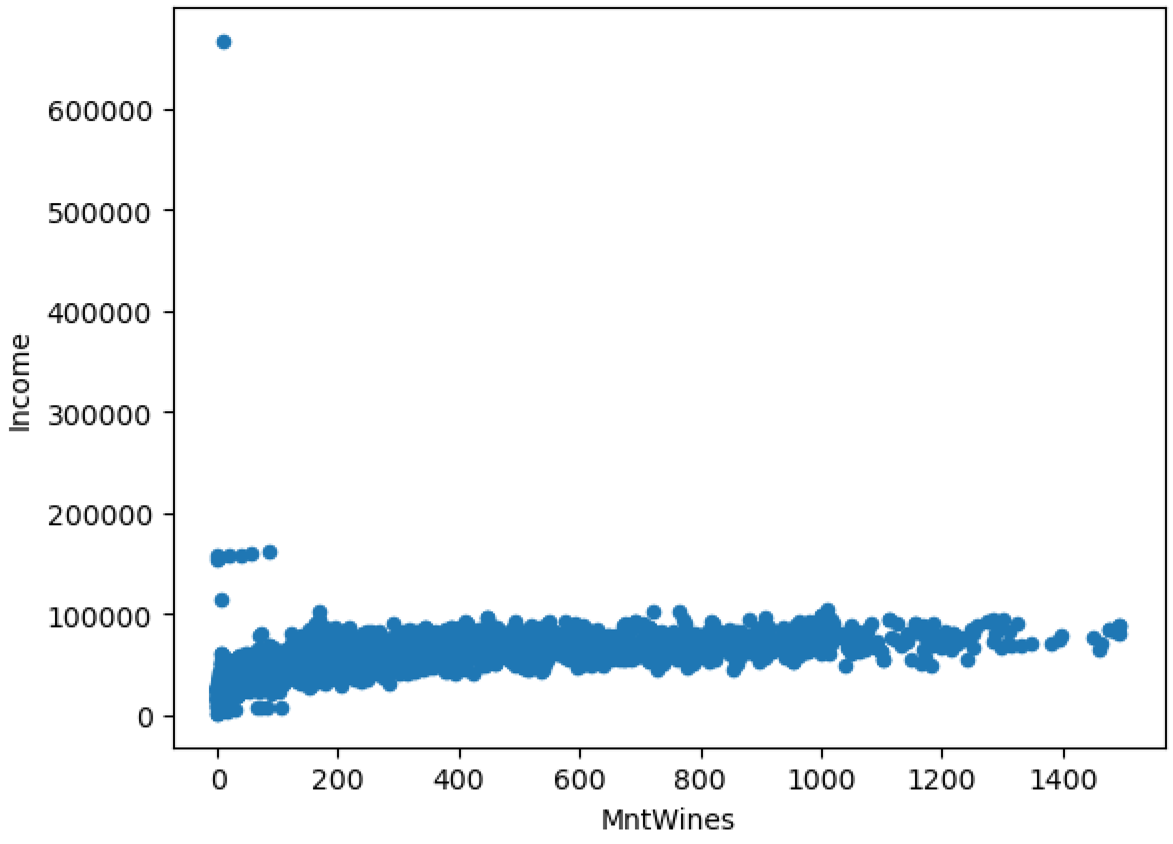
```
In [42]: # NUM VS NUM
         df.plot(kind='scatter',x='MntWines',y='Income')
```

```
Out[42]: <AxesSubplot:xlabel='MntWines', ylabel='Income'>
```



```
In [43]: clean_df.plot(kind='scatter',x='MntWines',y='Income')
```

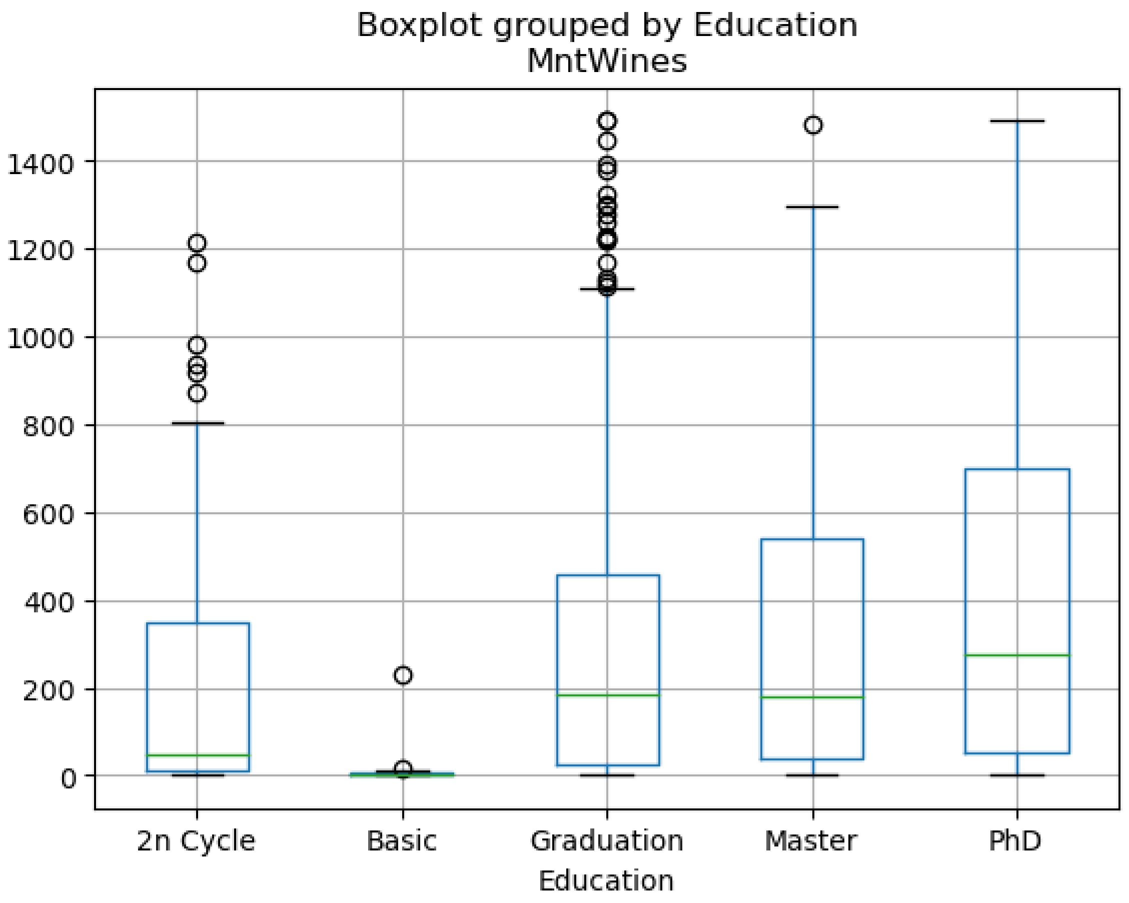
```
Out[43]: <AxesSubplot:xlabel='MntWines', ylabel='Income'>
```



```
In [46]: #NUM VS CAT
```

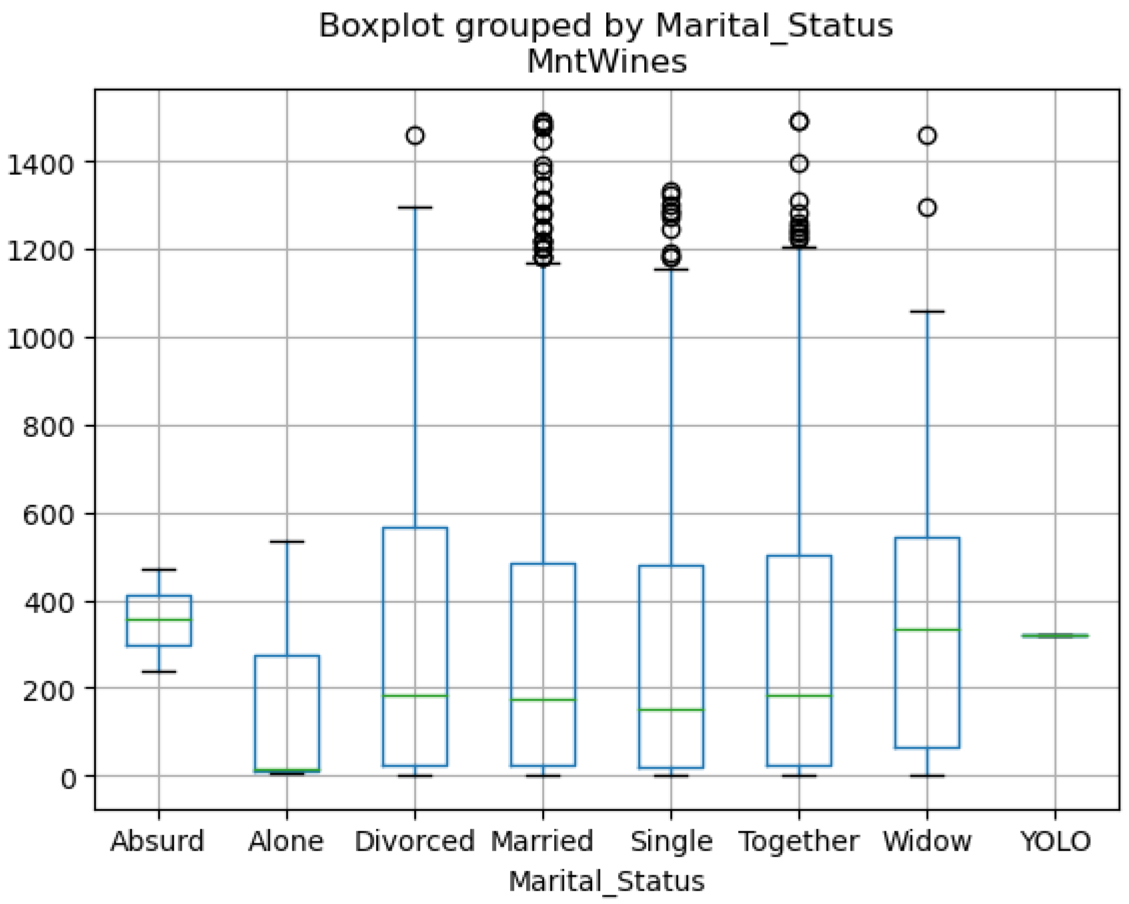
```
In [47]: clean_df.boxplot(by='Education',column='MntWines')
```

```
Out[47]: <AxesSubplot:title={'center':'MntWines'}, xlabel='Education'>
```



```
In [51]: clean_df.boxplot(by='Marital_Status',column='MntWines')
```

```
Out[51]: <AxesSubplot:title={'center':'MntWines'}, xlabel='Marital_Status'>
```



```
In [ ]:
```