```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
import pandas as pd
df = pd.read_csv('bankloan.csv')
df.head()
```

	ID	Age	Experience	Income	ZIP.Code	Family	CCAvg	Education	Mortgage	Personal.Loan	Securities.Account	CD.Account	Online	Cred
0	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	
1	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	
2	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	
3	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	
4	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	

```
# Show first 5 rows
print(df.head(15))
```

```
    ID  Age  Experience  Income  ZIP.Code  Family  CCAvg  Education  Mortgage  \
0    1   25           1      49     91107       4    1.6          1         0
1    2   45          19      34     90089       3    1.5          1         0
2    3   39          15      11     94720       1    1.0          1         0
3    4   35           9     100     94112       1    2.7          2         0
4    5   35           8      45     91330       4    1.0          2         0
5    6   37          13      29     92121       4    0.4          2       155
6    7   53          27      72     91711       2    1.5          2         0
7    8   50          24      22     93943       1    0.3          3         0
8    9   35          10      81     90089       3    0.6          2       104
9   10   34           9     180     93023       1    8.9          3         0
10  11   65          39     105     94710       4    2.4          3         0
11  12   29           5      45     90277       3    0.1          2         0
12  13   48          23     114     93106       2    3.8          3         0
13  14   59          32      40     94920       4    2.5          2         0
14  15   67          41     112     91741       1    2.0          1         0

    Personal.Loan  Securities.Account  CD.Account  Online  CreditCard
0               0                   1           0       0           0
1               0                   1           0       0           0
2               0                   0           0       0           0
3               0                   0           0       0           0
4               0                   0           0       0           1
5               0                   0           0       1           0
6               0                   0           0       1           0
7               0                   0           0       0           1
8               0                   0           0       1           0
9               1                   0           0       0           0
10              0                   0           0       0           0
11              0                   0           0       1           0
12              0                   1           0       0           0
13              0                   0           0       1           0
14              0                   1           0       0           0
```

```
# basic info
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ID                  5000 non-null   int64
 1   Age                 5000 non-null   int64
 2   Experience          5000 non-null   int64
 3   Income              5000 non-null   int64
 4   ZIP.Code            5000 non-null   int64
 5   Family              5000 non-null   int64
 6   CCAvg               5000 non-null   float64
 7   Education           5000 non-null   int64
 8   Mortgage            5000 non-null   int64
 9   Personal.Loan       5000 non-null   int64
 10  Securities.Account  5000 non-null   int64
```

```
 11  CD.Account          5000 non-null   int64
 12  Online              5000 non-null   int64
 13  CreditCard          5000 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 547.0 KB
None
```

```
#  null values
print(df.isnull().sum())
```

```
ID                   0
Age                  0
Experience           0
Income               0
ZIP.Code             0
Family               0
CCAvg                0
Education            0
Mortgage             0
Personal.Loan        0
Securities.Account   0
CD.Account           0
Online               0
CreditCard           0
dtype: int64
```

```
#Chek column names

print(df.columns)
```

```
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP.Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal.Loan', 'Securities.Account',
       'CD.Account', 'Online', 'CreditCard'],
      dtype='object')
```

```
df. shape
```

```
(5000, 14)
```

```
df. tail()
```

	ID	Age	Experience	Income	ZIP.Code	Family	CCAvg	Education	Mortgage	Personal.Loan	Securities.Account	CD.Account	Online
4995	4996	29	3	40	92697	1	1.9	3	0	0	0	0	1
4996	4997	30	4	15	92037	4	0.4	1	85	0	0	0	1
4997	4998	63	39	24	93023	2	0.3	3	0	0	0	0	0
4998	4999	65	40	49	90034	3	0.5	2	0	0	0	0	1
4999	5000	28	4	83	92612	3	0.8	1	0	0	0	0	1

```
df. info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 14 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   ID                  5000 non-null   int64
 1   Age                 5000 non-null   int64
 2   Experience          5000 non-null   int64
 3   Income              5000 non-null   int64
 4   ZIP.Code            5000 non-null   int64
 5   Family              5000 non-null   int64
 6   CCAvg               5000 non-null   float64
 7   Education           5000 non-null   int64
 8   Mortgage            5000 non-null   int64
 9   Personal.Loan       5000 non-null   int64
 10  Securities.Account  5000 non-null   int64
 11  CD.Account          5000 non-null   int64
 12  Online              5000 non-null   int64
 13  CreditCard          5000 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 547.0 KB
```

```
df. describe ()
```

	ID	Age	Experience	Income	ZIP.Code	Family	CCAvg	Education	Mortgage	Personal.Loan
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	2500.500000	45.338400	20.104600	73.774200	93152.503000	2.396400	1.937938	1.881000	56.498800	0.096000
std	1443.520003	11.463166	11.467954	46.033729	2121.852197	1.147663	1.747659	0.839869	101.713802	0.294621
min	1.000000	23.000000	-3.000000	8.000000	9307.000000	1.000000	0.000000	1.000000	0.000000	0.000000
25%	1250.750000	35.000000	10.000000	39.000000	91911.000000	1.000000	0.700000	1.000000	0.000000	0.000000
50%	2500.500000	45.000000	20.000000	64.000000	93437.000000	2.000000	1.500000	2.000000	0.000000	0.000000
75%	3750.250000	55.000000	30.000000	98.000000	94608.000000	3.000000	2.500000	3.000000	101.000000	0.000000
max	5000.000000	67.000000	43.000000	224.000000	96651.000000	4.000000	10.000000	3.000000	635.000000	1.000000

```
# Get value counts of a specific column
print(df['Personal.Loan'].value_counts())
```

```
Personal.Loan
0    4520
1     480
Name: count, dtype: int64
```

```
# Filter data
loan_yes = df[df['Personal.Loan'] == 'yes']
print(loan_yes.head())
```

```
Empty DataFrame
Columns: [ID, Age, Experience, Income, ZIP.Code, Family, CCAvg, Education, Mortgage, Personal.Loan, Securities.Account, CD.Account, Onli
Index: []
```

```
df.columns
```

```
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP.Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal.Loan', 'Securities.Account',
       'CD.Account', 'Online', 'CreditCard'],
      dtype='object')
```

```
# Drop a column
df.drop('ID', axis=1, inplace=True)
```

```
#
```

```
# Sort values by a column
sorted_df = df.sort_values(by='Income', ascending=False)
print(sorted_df.head())
```

```
         ID  Age  Experience  Income  ZIP.Code  Family  CCAvg  Education  \
3896  3897   48          24     224     93940       2   6.67          1
4993  4994   45          21     218     91801       2   6.67          1
526    527   26           2     205     93106       1   6.33          1
2988  2989   46          21     205     95762       2   8.80          1
2278  2279   30           4     204     91107       2   4.50          1

      Mortgage  Personal.Loan  Securities.Account  CD.Account  Online  \
3896         0              0                   0           1       1
4993         0              0                   0           0       1
526        271              0                   0           0       0
2988       181              0                   1           0       1
2278         0              0                   0           0       1

      CreditCard
3896           1
4993           0
526            1
2988           0
2278           0
```

```
# Shape of the DataFrame (rows, columns)
print(df.shape)
```

```
(5000, 14)
```

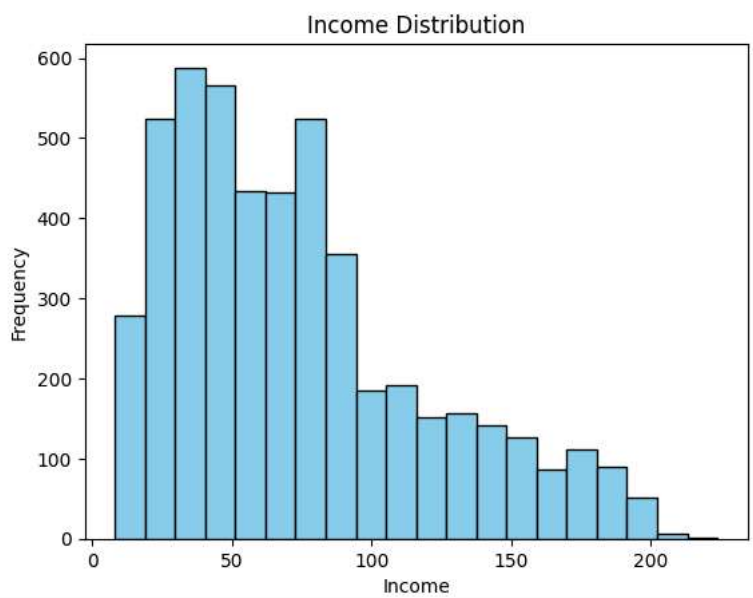
```
# Data types of each column
print(df.dtypes)
```

```
ID                     int64
Age                    int64
Experience             int64
Income                 int64
ZIP.Code               int64
Family                 int64
CCAvg                float64
Education              int64
Mortgage               int64
Personal.Loan          int64
Securities.Account     int64
CD.Account             int64
Online                 int64
CreditCard             int64
dtype: object
```

```
# Rename a column
df.rename(columns={'personal_loan': 'loan_status'}, inplace=True)
```

```
import matplotlib.pyplot as plt

# Histogram for a numeric column
plt.hist(df['Income'], bins=20, color='skyblue', edgecolor='black')
plt.title('Income Distribution')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.show()
```

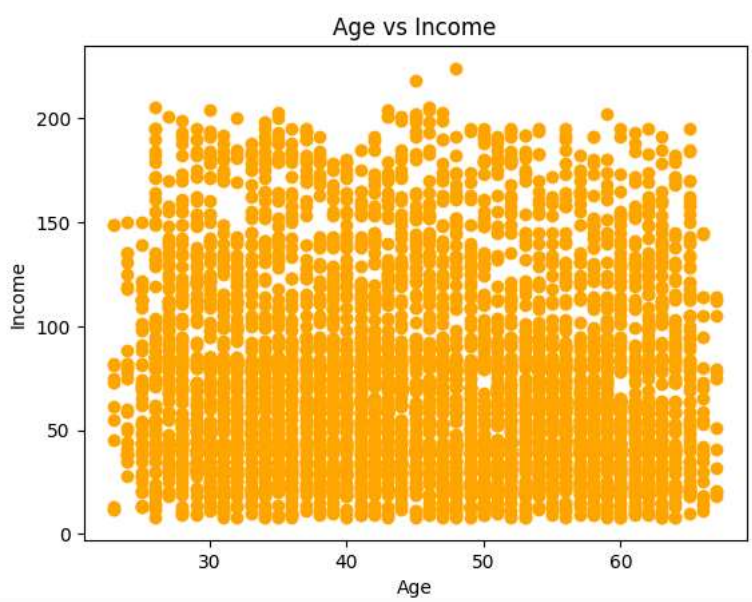



```
# Group by a column and calculate mean
print(df.groupby('Personal.Loan')['Income'].mean())
```

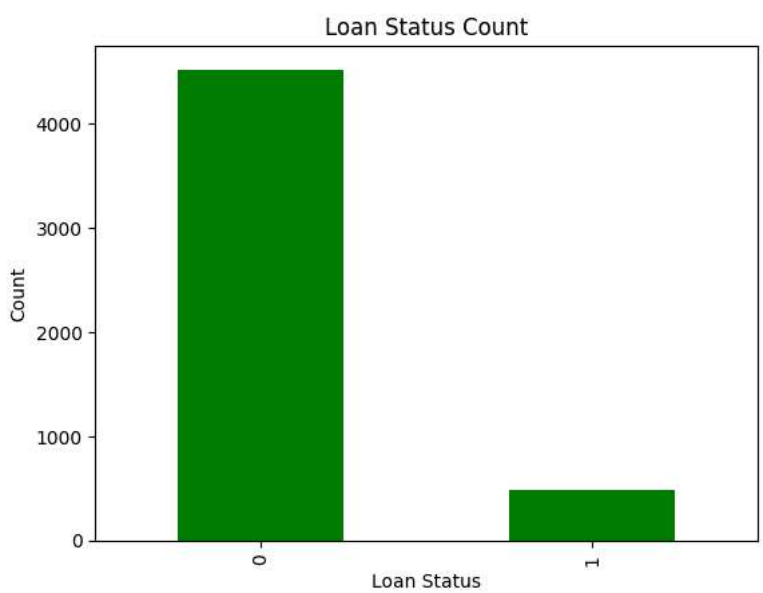
```
Personal.Loan
0     66.237389
1    144.745833
Name: Income, dtype: float64
```

```
# Scatter plot (e.g., Income vs Age)
plt.scatter(df['Age'], df['Income'], color='orange')
plt.title('Age vs Income')
```

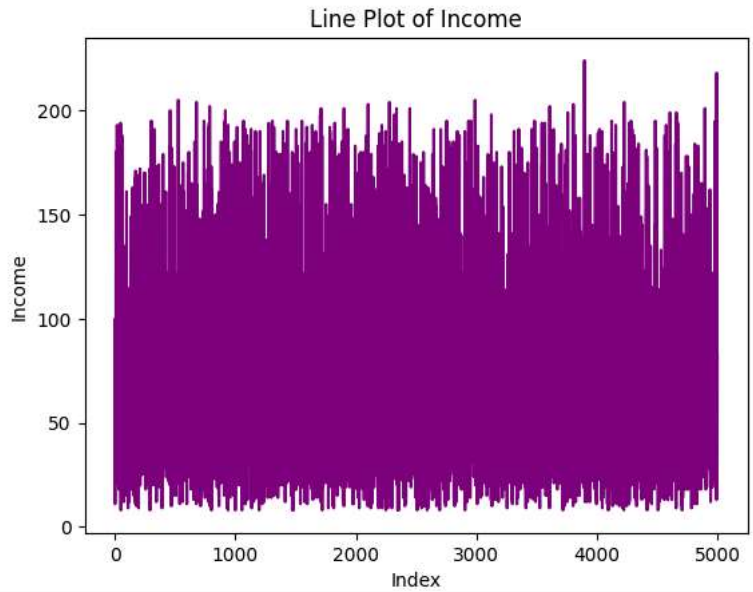
```
plt.xlabel('Age')
plt.ylabel('Income')
plt.show()
```




```
# Bar chart for categorical data
Loan_counts = df['Personal.Loan'].value_counts()
Loan_counts.plot(kind='bar', color='green')
plt.title('Loan Status Count')
plt.xlabel('Loan Status')
plt.ylabel('Count')
plt.show()
```




```
# Line plot (e.g., income vs index)
plt.plot(df['Income'], color='purple')
plt.title('Line Plot of Income')
plt.xlabel('Index')
plt.ylabel('Income')
plt.show()
```




```
# Box plot
import seaborn as sns
import matplotlib.pyplot as plt

# Replace 'personal_loan' with your actual column name if different
# Corrected column name from 'personal_loan' to 'Personal.Loan'
sns.boxplot(x='Personal.Loan', y='Income', data=df)

plt.title('Income Distribution by Loan Status')
plt.xlabel('Personal Loan (0 = No, 1 = Yes)')
plt.ylabel('Income')
plt.show()
```