In [1]:
```
# read the data
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
bank_data=pd.read_csv(r"F:\FSDS\Data Files\bank.csv", sep=';')
```

head

In [5]:
```
bank_data.head()
```

Out[5]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	du
0	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	

tail

In [6]:
```
bank_data.tail()
```

Out[6]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month
4516	33	services	married	secondary	no	-333	yes	no	cellular	30	jul
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	9	may
4518	57	technician	married	secondary	no	295	no	no	cellular	19	aug
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	6	feb
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	3	apr

shape

In [7]:
```
bank_data.shape
```

Out[7]:
```
(4521, 17)
```

In [8]:
```
print("the number of rows:", bank_data.shape[0])
print("the number of columns:", bank_data.shape[1])
```

```
the number of rows: 4521
the number of columns: 17
```

size

In [9]:
```
bank_data.size
```

Out[9]:
```
76857
```

In [10]:
```
bank_data.shape[0]*bank_data.shape[1]
```

Out[10]:
```
76857
```

columns

In [11]:
```
bank_data.columns
```

Out[11]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'y'],
      dtype='object')
```

In [12]:
```
type(bank_data.columns)
```

Out[12]:
```
pandas.core.indexes.base.Index
```

In [13]:
```
list(bank_data.columns)
```

Out[13]:
```
['age',
 'job',
 'marital',
 'education',
 'default',
 'balance',
 'housing',
 'loan',
 'contact',
 'day',
 'month',
 'duration',
 'campaign',
 'pdays',
 'previous',
 'poutcome',
 'y']
```

In [14]:
```
len(bank_data.columns)
```

Out[14]:
```
17
```

data-types

In [15]:
```
bank_data.dtypes
```

Out[15]:
```
age           int64
job          object
marital      object
education    object
default      object
balance       int64
housing      object
loan         object
contact      object
day           int64
month        object
duration      int64
campaign      int64
pdays         int64
previous      int64
poutcome     object
y            object
dtype: object
```

In [16]:
```
dtypes=bank_data.dtypes
dtypes.keys()
```

Out[16]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'y'],
      dtype='object')
```

In [17]:
```
bank_data.columns
```

Out[17]:
```
Index(['age', 'job', 'marital', 'education', 'default', 'balance', 'housing',
       'loan', 'contact', 'day', 'month', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'y'],
      dtype='object')
```

In [18]:
```
dtypes.values
```

Out[18]:
```
array([dtype('int64'), dtype('O'), dtype('O'), dtype('O'), dtype('O'),
       dtype('int64'), dtype('O'), dtype('O'), dtype('O'), dtype('int64'),
       dtype('O'), dtype('int64'), dtype('int64'), dtype('int64'),
       dtype('int64'), dtype('O'), dtype('O')], dtype=object)
```

task

In [20]:
```
for i, j in dtypes.items():
    if j=='object':
        print(i)
```

```
job
marital
education
default
housing
loan
contact
month
poutcome
y
```

In [21]:
```
cat=[i for i,j in dtypes.items() if j=='object']
num=[i for i,j in dtypes.items() if j!='object']
```

In [22]:
```
cat
```

Out[22]:
```
['job',
 'marital',
 'education',
 'default',
 'housing',
 'loan',
 'contact',
 'month',
 'poutcome',
 'y']
```

In [23]:
```
num
```

Out[23]:
```
['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous']
```

selectdtypes

In [25]:
```
bank_data.select_dtypes(include='object').columns
```

Out[25]:
```
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome', 'y'],
      dtype='object')
```

In [26]:
```
bank_data.select_dtypes(exclude='object').columns
```

Out[26]:
```
Index(['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous'], dtype
='object')
```

isnull

In [27]:
```
bank_data.isnull()
```

Out[27]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	durati
0	False	False	False	False	False	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	False	Fa
...	...	...	...	...	...	...	...	...	...	...	...	
4516	False	False	False	False	False	False	False	False	False	False	False	Fa
4517	False	False	False	False	False	False	False	False	False	False	False	Fa
4518	False	False	False	False	False	False	False	False	False	False	False	Fa
4519	False	False	False	False	False	False	False	False	False	False	False	Fa
4520	False	False	False	False	False	False	False	False	False	False	False	Fa

4521 rows × 17 columns

In [29]:
```
bank_data.isnull().sum()
```

Out[29]:
```
age          0
job          0
marital      0
education    0
default      0
balance      0
housing      0
loan         0
contact      0
day          0
month        0
duration     0
campaign     0
pdays        0
previous     0
poutcome     0
y            0
dtype: int64
```

In [30]:
```
bank_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4521 entries, 0 to 4520
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   age        4521 non-null   int64 
 1   job        4521 non-null   object
 2   marital    4521 non-null   object
 3   education  4521 non-null   object
 4   default    4521 non-null   object
 5   balance    4521 non-null   int64 
 6   housing    4521 non-null   object
 7   loan       4521 non-null   object
 8   contact    4521 non-null   object
 9   day        4521 non-null   int64 
 10  month      4521 non-null   object
 11  duration   4521 non-null   int64 
 12  campaign   4521 non-null   int64 
 13  pdays      4521 non-null   int64 
 14  previous   4521 non-null   int64 
 15  poutcome   4521 non-null   object
 16  y          4521 non-null   object
dtypes: int64(7), object(10)
memory usage: 600.6+ KB
```

take

In [31]:
```
l=[100,200,300]
bank_data.take(l)
```

Out[31]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	du
100	36	blue-collar	married	secondary	no	0	yes	no	unknown	6	may	
200	34	technician	single	tertiary	no	992	yes	no	cellular	4	may	
300	70	retired	divorced	primary	no	4531	no	no	cellular	18	may	

In [33]:
```
l=[1,2,3]
bank_data.take(l,axis=1)
```

Out[33]:

	job	marital	education
0	unemployed	married	primary
1	services	married	secondary
2	management	single	tertiary
3	management	married	tertiary
4	blue-collar	married	secondary
...	...	...	...
4516	services	married	secondary
4517	self-employed	married	tertiary
4518	technician	married	secondary
4519	blue-collar	married	secondary
4520	entrepreneur	single	tertiary

4521 rows × 3 columns

In [34]:
```
l=[1,2,3]
bank_data.take(l, axis=0)
```

Out[34]:

	age	job	marital	education	default	balance	housing	loan	contact	day	month	du
1	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	
2	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	

iloc-loc

In [35]:
```
bank_data.iloc[1:2, 3:7]
# when to use loc and iloc
```

Out[35]:

	education	default	balance	housing
1	secondary	no	4789	yes

In [37]:
```
bank_data['job']
# it is in series
```

Out[37]:
```
0          unemployed
1            services
2          management
3          management
4         blue-collar
            ...
4516         services
4517    self-employed
4518       technician
4519      blue-collar
4520     entrepreneur
Name: job, Length: 4521, dtype: object
```

In [38]:
```
bank_data[['job']]
# it is in dataframe
```

Out[38]:

	job
0	unemployed
1	services
2	management
3	management
4	blue-collar
...	...
4516	services
4517	self-employed
4518	technician
4519	blue-collar
4520	entrepreneur

4521 rows × 1 columns

In [ ]:
```

```