

```
import pandas as pd
df=pd.DataFrame(
    {"a":[4,5,6],
     "b":[7,8,9],
     "c":[10,11,12]},
    index=[1,2,3])
df
```

	a	b	c
1	4	7	10
2	5	8	11
3	6	9	12

```
import pandas as pd
df=pd.DataFrame(
    [[4,5,6],
     [7,8,9],
     [10,11,12]],
    index=[1,2,3],
    columns=['a','b','c'])
df
```

	a	b	c
1	4	5	6
2	7	8	9
3	10	11	12

```
import pandas as pd
df=pd.DataFrame()
print(df)
```

```
Empty DataFrame
Columns: []
Index: []
```

```
import pandas as pd
data=[1,2,3,4]
df=pd.DataFrame(data)
print(df)
```

```
0
0 1
1 2
2 3
3 4
```

```
import pandas as pd
data=[['Alex',10],['Bob',12]]
df=pd.DataFrame(data,columns=['Name','Age'])
print(df)
```

```
Name  Age
0  Alex   10
1  Bob    12
```

```
import pandas as pd
data={'Name':['Tom','Jack','steve','ricky'],'Age':[28,34,29,42]}
df=pd.DataFrame(data,index=['a','b','c','d'])
print(df)
```

	Name	Age
a	Tom	28
b	Jack	34
c	steve	29
d	ricky	42

```
import pandas as pd
mydataset={
    'cars':["BMW","Volvo","Ford"],
    'passings':[3,7,2]
}
myvar=pd.DataFrame(mydataset)
myvar
```

	cars	passings
0	BMW	3
1	Volvo	7
2	Ford	2

```
import pandas as pd
data=[{'a':1,'b':2},{'a':5,'b':10,'c':20}]
df=pd.DataFrame(data)
df
```

	a	b	c
0	1	2	NaN
1	5	10	20.0

```
import pandas as pd
data=[{'a':1,'b':2},{'a':5,'b':10,'c':20}]
df=pd.DataFrame(data,index=['first','second'])
df
```

	a	b	c
first	1	2	NaN
second	5	10	20.0

```
import pandas as pd
data=[{'a':1,'b':2},{'a':5,'b':10,'c':20}]
df=pd.DataFrame(data,index=['first','second'],columns=['a','b'])
df
```

	a	b
first	1	2
second	5	10

```
import pandas as pd
data={'Name':['ash','serina','naruto'],
      'Age':[18,19,20],
      'Address':['Palettown','leafvillage','konoha'],
      'Qualification':['battler','battler','shinobi']}
df=pd.DataFrame(data)
print(df[['Name','Address','Qualification']])
```

	Name	Address	Qualification
0	ash	Palettown	battler
1	serina	leafvillage	battler
2	naruto	konoha	shinobi

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'],'Height':[5.1,6.2,5.1,5.2],'Qualification':['Msc','MA','MCA','PhD']}
df=pd.DataFrame(data)
address=['Delhi','Kanpur','Allahabad','Kannauj']
df['Address']=address
print(df)
```

	Name	Height	Qualification	Address
0	Jai	5.1	Msc	Delhi
1	Princi	6.2	MA	Kanpur
2	Gaurav	5.1	MCA	Allahabad
3	Anuj	5.2	PhD	Kannauj

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'],'Address':['Delhi','Kanpur','Allahabad','Kannauj'],'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
del df['Address']
print(df)
```

	Name	Height	Qualification
0	Jai	5.1	Msc
1	Princi	6.2	MA
2	Gaurav	5.1	MCA
3	Anuj	5.2	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'],'Address':['Delhi','Kanpur','Allahabad','Kannauj'],'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
df.drop(['Address'],axis=1,inplace=False)
print(df)
```

	Name	Address	Height	Qualification
0	Jai	Delhi	5.1	Msc
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'],'Address':['Delhi','Kanpur','Allahabad','Kannauj'],'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
df.drop(['Address'],axis=1,inplace=True)
print(df)
```

	Name	Height	Qualification
0	Jai	5.1	Msc
1	Princi	6.2	MA
2	Gaurav	5.1	MCA
3	Anuj	5.2	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
df.drop(['Address'],axis=1)
print(df)
```

	Name	Address	Height	Qualification
0	Jai	Delhi	5.1	Msc
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
df.pop('Height')
print(df)
```

	Name	Address	Qualification
0	Jai	Delhi	Msc
1	Princi	Kanpur	MA
2	Gaurav	Allahabad	MCA
3	Anuj	Kannauj	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
print(df)
df.rename(columns={'Address':'Place'},inplace=True)
df
```

	Name	Address	Height	Qualification
0	Jai	Delhi	5.1	Msc
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

	Name	Place	Height	Qualification
0	Jai	Delhi	5.1	Msc
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
print(df)
df.columns=['A','B','C','D']
df
```

	Name	Address	Height	Qualification
0	Jai	Delhi	5.1	Msc
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

	A	B	C	D
0	Jai	Delhi	5.1	Msc
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

```
import pandas as pd
df=pd.DataFrame([[1,2],[3,4]],columns=['a','b'])
df2=pd.DataFrame([[5,6],[7,8]],columns=['a','b'])
df=df.append(df2)
df
```

```
<ipython-input-19-2463f2a19c68>:4: FutureWarning: The frame.append method
df=df.append(df2)
```

	a	b
0	1	2
1	3	4
0	5	6
1	7	8

```
import pandas as pd
df=pd.DataFrame([[1,2],[3,4]],columns=['a','b'])
df2=pd.DataFrame([[5,6],[7,8]],columns=['a','b'])
df=df.append(df2,ignore_index=True)
df
```

```
<ipython-input-3-d4e2f1da207e>:4: FutureWarning: The frame.append method
df=df.append(df2,ignore_index=True)
```

	a	b
0	1	2
1	3	4
2	5	6
3	7	8

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'],'Address':['Delhi','Kanpur','Allahabad','Kannauj'],'Height':[5.1,6.2,5.1,5.2]}
df=pd.DataFrame(data)
df
df.drop(0,axis=0,inplace=True)
df
```

	Name	Address	Height	Qualification
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'],'Address':['Delhi','Kanpur','Allahabad','Kannauj'],'Height':[5.1,6.2,5.5,5.9]}
df=pd.DataFrame(data)
df
df.drop(1, axis=0, inplace=True)
df
```

	Name	Address	Height	Qualification
0	Jai	Delhi	5.1	Msc
2	Gaurav	Allahabad	5.1	MCA
3	Anuj	Kannauj	5.2	PhD

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df=df[['name']]
print(df)
```

```
0      Jai
1    Princi
2    Gaurav
3     Anuj
Name: name, dtype: object
```

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df=df[['name']]
print(df)
```

```
      name
0      Jai
1    Princi
2    Gaurav
3     Anuj
```

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df=df[['age']]
print(df)
```

```
0    25
1    32
2    18
3    47
Name: age, dtype: int64
```

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df.filter(items=['name','age'])
```

	name	age
0	Jai	25
1	Princi	32
2	Gaurav	18
3	Anuj	47

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df.filter(like='eigh')
```

	height
0	1.62
1	1.78
2	1.65
3	1.83

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
#select only the column that contain the string end
df.filter(like='end')
```

	gender
0	F
1	M
2	M
3	M

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df.filter(regex='e|a',axis=1)
```

	name	age	gender	height
0	Jai	25	F	1.62
1	Princi	32	M	1.78
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df.filter(regex='ei',axis=1)
```

	height
0	1.62
1	1.78
2	1.65
3	1.83

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df.filter(regex='e',axis=1)
```

	name	age	gender	height
0	Jai	25	F	1.62
1	Princi	32	M	1.78
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df.filter(regex='e|i',axis=1)
```

	name	age	gender	height
0	Jai	25	F	1.62
1	Princi	32	M	1.78
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df.filter(regex='i',axis=1)
```

	height
0	1.62
1	1.78
2	1.65
3	1.83

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates()
df
```

	name	age	gender	height
0	Jai	25	F	1.62
1	Princi	32	M	1.78
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Jai','Gaurav','Anuj'],
      'age':[25,25,18,47],
      'gender':['F','F','M','M'],
      'height':[1.62,1.62,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates()
df
```

	name	age	gender	height
0	Jai	25	F	1.62
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Princi','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates(subset=['name', 'age'])
df
```

	name	age	gender	height
<b>0</b>	Jai	25	F	1.62
<b>1</b>	Princi	32	M	1.78
<b>2</b>	Gaurav	18	M	1.65
<b>3</b>	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Jai','Gaurav','Anuj'],
      'age':[25,25,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates(subset=['name', 'age'])
df
```

	name	age	gender	height
<b>0</b>	Jai	25	F	1.62
<b>2</b>	Gaurav	18	M	1.65
<b>3</b>	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Jai','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates(subset=['name', 'age'])
df
```

	name	age	gender	height
<b>0</b>	Jai	25	F	1.62
<b>1</b>	Jai	32	M	1.78
<b>2</b>	Gaurav	18	M	1.65
<b>3</b>	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Jai','Gaurav','Anuj'],
      'age':[25,32,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates()
df
```

	name	age	gender	height
0	Jai	25	F	1.62
1	Jai	32	M	1.78
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Jai','Gaurav','Anuj'],
      'age':[25,25,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates(subset=['name', 'age'],keep='last')
df
```

	name	age	gender	height
1	Jai	25	M	1.78
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Jai','Gaurav','Anuj'],
      'age':[25,25,18,47],
      'gender':['F','F','M','M'],
      'height':[1.62,1.62,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates(keep='last')
df
```

	name	age	gender	height
1	Jai	25	F	1.62
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'name':['Jai','Jai','Gaurav','Anuj'],
      'age':[25,25,18,47],
      'gender':['F','M','M','M'],
      'height':[1.62,1.78,1.65,1.83]}
df=pd.DataFrame(data)
df = df.drop_duplicates(subset=['name', 'age'],keep='first')
df
```

	name	age	gender	height
0	Jai	25	F	1.62
2	Gaurav	18	M	1.65
3	Anuj	47	M	1.83

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.3,5.5]}
df=pd.DataFrame(data)
df_sample=df.sample(n=2)
print(df_sample)
```

	Name	Address	Height	Qualification
3	Anuj	Kannauj	5.2	PhD
1	Princi	Kanpur	6.2	MA

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.3,5.5]}
df=pd.DataFrame(data)
df_sample=df.sample(n=3)
print(df_sample)
```

	Name	Address	Height	Qualification
3	Anuj	Kannauj	5.2	PhD
0	Jai	Delhi	5.1	Msc
2	Gaurav	Allahabad	5.1	MCA

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.3,5.5]}
df=pd.DataFrame(data)
df.sample(n=2)
```

	Name	Address	Height	Qualification
1	Princi	Kanpur	6.2	MA
0	Jai	Delhi	5.1	Msc

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.3,5.5]}
df=pd.DataFrame(data)
df_sample=df.sample(n=2)
print(df_sample)
```

	Name	Address	Height	Qualification
1	Princi	Kanpur	6.2	MA
2	Gaurav	Allahabad	5.1	MCA

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.3,5.5]}
df=pd.DataFrame(data)
df_sample=df.sample(frac=0.5)
print(df_sample)
```

	Name	Address	Height	Qualification
1	Princi	Kanpur	6.2	MA
3	Anuj	Kannauj	5.2	PhD

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.3,5.5]}
df=pd.DataFrame(data)
df_sample=df.sample(frac=0.5)
print(df_sample)
```

	Name	Address	Height	Qualification
3	Anuj	Kannauj	5.2	PhD
1	Princi	Kanpur	6.2	MA

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.5,5.9], 'Qualification':['MSc','MA','MCA','PhD']}
df=pd.DataFrame(data)
df_sample=df.sample(frac=0.5)
print(df_sample)
```

	Name	Address	Height	Qualification
2	Gaurav	Allahabad	5.1	MCA
1	Princi	Kanpur	6.2	MA

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.5,5.9], 'Qualification':['MSc','MA','MCA','PhD']}
df=pd.DataFrame(data)
df_sample=df.sample(frac=0.4)
print(df_sample)
```

	Name	Address	Height	Qualification
3	Anuj	Kannauj	5.2	PhD
0	Jai	Delhi	5.1	Msc

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.5,5.9], 'Qualification':['MSc','MA','MCA','PhD']}
df=pd.DataFrame(data)
df_sample=df.sample(frac=0.1)
print(df_sample)
```

Empty DataFrame  
Columns: [Name, Address, Height, Qualification]  
Index: []

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.5,5.9], 'Qualification':['MSc','MA','MCA','PhD']}
df=pd.DataFrame(data)
df_sample=df.sample(frac=0.2)
print(df_sample)
```

	Name	Address	Height	Qualification
0	Jai	Delhi	5.1	Msc

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.5,5.9], 'Qualification':['MSc','MA','MCA','PhD']}
df=pd.DataFrame(data)
df_sample=df.sample(frac=0.5)
print(df_sample)
```

	Name	Address	Height
2	Gaurav	Allahabad	5.1
3	Anuj	Kannauj	5.2

```
import pandas as pd
data={'Name':['Jai','Princi','Gaurav','Anuj'], 'Address':['Delhi','Kanpur','Allahabad','Kannauj'], 'Height':[5.1,6.2,5.5,5.9], 'Qualification':['MSc','MA','MCA','PhD']}
df=pd.DataFrame(data)
df_sample=df.sample(n=2, axis=1)
print(df_sample)
```

	Name	Qualification
0	Jai	Msc
1	Princi	MA
2	Gaurav	MCA
3	Anuj	PhD

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],
        'age': [25, 30, 35, 40, 45],
        'salary': [50000, 60000, 70000, 80000, 90000]}
df = pd.DataFrame(data)
# Get the 2 rows with the largest salary
top_salaries = df.nlargest(2, columns='salary')
print(top_salaries)
```

	name	age	salary
4	Emily	45	90000
3	David	40	80000

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],
        'age': [25, 30, 35, 40, 45],
        'salary': [50000, 60000, 70000, 80000, 90000]}
df = pd.DataFrame(data)
top_salaries = df.nsmallest(2, columns='salary')
print(top_salaries)
```

	name	age	salary
0	Alice	25	50000
1	Bob	30	60000

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'David', 'Emily'],
        'age': [25, 30, 35, 40, 45],
        'salary': [50000, 60000, 70000, 80000, 90000]}
df = pd.DataFrame(data)
df.query('age >= 30')
```

	name	age	salary
1	Bob	30	60000
2	Charlie	35	70000
3	David	40	80000
4	Emily	45	90000

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M'],
        'height': [1.62, 1.78, 1.65, 1.83]}
df = pd.DataFrame(data)
df.query('name.str.contains("a") and height>1.7')
```

	name	age	gender	height
3	Dave	47	M	1.83

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M'],
        'height': [1.62, 1.78, 1.65, 1.83]}
df = pd.DataFrame(data)
df.query('gender==["F","M"] and height<=1.65')
```

	name	age	gender	height
0	Alice	25	F	1.62
2	Charlie	18	M	1.65

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M'],
        'height': [1.62, 1.78, 1.65, 1.83]}
df = pd.DataFrame(data)
df.query('height<=1.65')
```

	name	age	gender	height
0	Alice	25	F	1.62
2	Charlie	18	M	1.65

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M'],
        'height': [1.62, 1.78, 1.65, 1.83]}
df = pd.DataFrame(data)
df.query('age==[25,32] and height<=1.65')
```

	name	age	gender	height
0	Alice	25	F	1.62

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M'],
        'height': [1.62, 1.78, 1.65, 1.83]}
df = pd.DataFrame(data)
df.query('name==["Alice", "Bob"] and height<=1.65')
```

	name	age	gender	height
0	Alice	25	F	1.62

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M']}
df = pd.DataFrame(data)
df.loc[:, 'age']
```

```
0    25
1    32
2    18
3    47
Name: age, dtype: int64
```

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M']}
df = pd.DataFrame(data)
df.loc[:,['age']]
```

```
age
0    25
1    32
2    18
3    47
```

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M']}
df = pd.DataFrame(data)
df.loc[:,['name','age']]
```

	name	age
0	Alice	25
1	Bob	32
2	Charlie	18
3	Dave	47

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M']}
df = pd.DataFrame(data)
df.iloc[:,2]
```

```
0    F
1    M
2    M
3    M
Name: gender, dtype: object
```

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M']}
df = pd.DataFrame(data)
df.iloc[:,0]
```

```
0      Alice
1      Bob
2    Charlie
3     Dave
Name: name, dtype: object
```

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M']}
df = pd.DataFrame(data)
df.iloc[:,[0]]
```

	name
0	Alice
1	Bob
2	Charlie
3	Dave

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M']}
df_filtered=df[df['age']>30]
print(df_filtered)
```

	name	age	gender
1	Bob	32	M
3	Dave	47	M

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M'],
        'height': [1.62, 1.78, 1.65, 1.83]}
df=pd.DataFrame(data)
df_filtered=df[(df['gender'] == 'M') & (df['height'] > 1.7)]
print(df_filtered)
```

	name	age	gender	height
1	Bob	32	M	1.78
3	Dave	47	M	1.83

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 32, 18, 47],
        'gender': ['F', 'M', 'M', 'M'],
        'height': [1.62, 1.78, 1.65, 1.83]}
df=pd.DataFrame(data)
df_filtered = df[df['name'].str.startswith(('A', 'C'))]
print(df_filtered)
```

	name	age	gender	height
0	Alice	25	F	1.62
2	Charlie	18	M	1.65

```
import pandas as pd
data = {'Name': ['John', 'Sarah', 'Mike', 'Emily', 'David'],
        'Age': [25, 31, 29, 35, 27],
        'Gender': ['M', 'F', 'M', 'F', 'M'],
        'Salary': [50000, 70000, 60000, 80000, 55000]}
df = pd.DataFrame(data)
print(df.head(3))
```

```
Name  Age Gender  Salary
0   John    25      M    50000
1  Sarah    31      F    70000
2   Mike    29      M    60000
```

```
import pandas as pd
data = {'Name': ['John', 'Sarah', 'Mike', 'Emily', 'David'],
        'Age': [25, 31, 29, 35, 27],
        'Gender': ['M', 'F', 'M', 'F', 'M'],
        'Salary': [50000, 70000, 60000, 80000, 55000]}
df = pd.DataFrame(data)
print(df.tail(2))
```

```
Name  Age Gender  Salary
3  Emily    35      F    80000
4  David    27      M    55000
```

```
import pandas as pd
data = {'Name': ['John', 'Sarah', 'Mike', 'Emily', 'David'],
        'Age': [25, 31, 29, 35, 27],
        'Gender': ['M', 'F', 'M', 'F', 'M'],
        'Salary': [50000, 70000, 60000, 80000, 55000]}
df = pd.DataFrame(data)
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype  
--- 
 0   Name     5 non-null    object  
 1   Age      5 non-null    int64  
 2   Gender   5 non-null    object  
 3   Salary   5 non-null    int64  
dtypes: int64(2), object(2)
memory usage: 288.0+ bytes
None
```

```
import pandas as pd
data = {'Name': ['John', 'Sarah', 'Mike', 'Emily', 'David'],
        'Age': [25, 31, 29, 35, 27],
        'Gender': ['M', 'F', 'M', 'F', 'M'],
        'Salary': [50000, 70000, 60000, 80000, 55000]}
df = pd.DataFrame(data)
print(df.describe())
```

	Age	Salary
count	5.000000	5.000000
mean	29.400000	63000.000000
std	3.847077	12041.594579
min	25.000000	50000.000000
25%	27.000000	55000.000000
50%	29.000000	60000.000000
75%	31.000000	70000.000000
max	35.000000	80000.000000

```
import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 30, 35, 40],
        'score': [90, 80, 85, 95]}
df = pd.DataFrame(data)
df_sorted = df.sort_values(by='age', ascending=False)
print(df_sorted)
```

```

      name  age  score
3     Dave   40    95
2  Charlie   35    85
1     Bob   30    80
0    Alice   25    90

```

```

import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave'],
        'age': [25, 30, 35, 40],
        'score': [90, 80, 85, 95]}
df = pd.DataFrame(data)
df_sorted = df.sort_values(by='age', ascending=True)
print(df_sorted)

```

```

      name  age  score
0    Alice   25    90
1     Bob   30    80
2  Charlie   35    85
3     Dave   40    95

```

```

import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Emily', 'Frank'],
        'gender': ['F', 'M', 'M', 'M', 'F', 'M'],
        'age': [25, 35, 40, 28, 30, 45],
        'salary': [50000, 70000, 60000, 80000, 65000, 90000]}
df = pd.DataFrame(data)
grouped = df.groupby('gender').mean()['salary']
print(grouped)

```

```

gender
F    57500.0
M    75000.0
Name: salary, dtype: float64
<ipython-input-68-b1981a841f>:7: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is
grouped = df.groupby('gender').mean()['salary']

```

```

import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Emily', 'Frank'],
        'gender': ['F', 'M', 'M', 'M', 'F', 'M'],
        'age': [25, 35, 40, 28, 30, 45],
        'salary': [50000, 70000, 60000, 80000, 65000, 90000]}
df = pd.DataFrame(data)
grouped = df.mean()['salary']
print(grouped)

```

```

69166.66666666667
<ipython-input-73-d600711b3d02>:7: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated
grouped = df.mean()['salary']

```

```

import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Emily', 'Frank'],
        'gender': ['F', 'M', 'M', 'M', 'F', 'M'],
        'age': [25, 35, 40, 28, 30, 45],
        'salary': [50000, 70000, 60000, 80000, 65000, 90000]}
df = pd.DataFrame(data)
grouped = df.groupby('age').mean()['salary']
print(grouped)

```

```

age
25    50000.0
28    80000.0

```

```

30    65000.0
35    70000.0
40    60000.0
45    90000.0
Name: salary, dtype: float64
<ipython-input-74-be935822177a>:7: FutureWarning: The default value of numeric_only in DataFrameGroupBy.mean is
grouped = df.groupby('age').mean()['salary']

```

```

import pandas as pd
data = {'name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Emily', 'Frank'],
        'gender': ['F', 'M', 'M', 'M', 'F', 'M'],
        'age': [25, 35, 40, 28, 30, 45],
        'salary': [50000, 70000, 60000, 80000, 65000, 90000]}
df = pd.DataFrame(data)
grouped = df.groupby('gender').count()
print(grouped)

```

	name	age	salary
gender			
F	2	2	2
M	4	4	4

```

import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, 32, None, 41, 28],
        'Salary': [50000, None, 70000, 90000, 60000]}
df = pd.DataFrame(data)
df_cleaned = df.dropna(subset=['Salary'])
print(df_cleaned)

```

	Name	Age	Salary
0	Alice	25.0	50000.0
2	Charlie	NaN	70000.0
3	Dave	41.0	90000.0
4	Eve	28.0	60000.0

```

import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, 32, None, 41, 28],
        'Salary': [50000, None, 70000, 90000, 60000]}
df = pd.DataFrame(data)
df_cleaned_all = df.dropna(how='all')
print(df_cleaned_all)

```

	Name	Age	Salary
0	Alice	25.0	50000.0
1	Bob	32.0	NaN
2	Charlie	NaN	70000.0
3	Dave	41.0	90000.0
4	Eve	28.0	60000.0

```

import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, 32, None, 41, 28],
        'Salary': [50000, None, 70000, 90000, 60000]}
df = pd.DataFrame(data)
df_cleaned_thresh = df.dropna(thresh=2)
print(df_cleaned_thresh)

```

	Name	Age	Salary
0	Alice	25.0	50000.0
1	Bob	32.0	NaN
2	Charlie	NaN	70000.0

```
3      Dave  41.0  90000.0
4      Eve   28.0  60000.0
```

```
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, 32, None, 41, 28],
        'Salary': [50000, None, 70000, 90000, 60000]}
df = pd.DataFrame(data)
df_cleaned_thresh = df.dropna(thresh=3)
print(df_cleaned_thresh)
```

	Name	Age	Salary
0	Alice	25.0	50000.0
3	Dave	41.0	90000.0
4	Eve	28.0	60000.0

```
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, 32, None, 41, 28],
        'Salary': [50000, None, 70000, 90000, 60000]}
df = pd.DataFrame(data)
df.dropna()
```

	Name	Age	Salary
0	Alice	25.0	50000.0
3	Dave	41.0	90000.0
4	Eve	28.0	60000.0

```
import pandas as pd
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, 32, None, 41, 28],
        'Salary': [50000, None, 70000, 90000, 60000]}
df = pd.DataFrame(data)
df.dropna(inplace=True)
print(df)
```

	Name	Age	Salary
0	Alice	25.0	50000.0
3	Dave	41.0	90000.0
4	Eve	28.0	60000.0

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, np.nan, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000]}
df = pd.DataFrame(data)
df_filled = df.fillna(0)
print(df_filled)
```

	Name	Age	Salary
0	Alice	25.0	50000.0
1	Bob	0.0	0.0
2	Charlie	35.0	70000.0
3	Dave	41.0	0.0
4	Eve	0.0	60000.0

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, np.nan, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000]}
df = pd.DataFrame(data)
df_ffilled = df.fillna(method='ffill')
print(df_ffilled)
```

	Name	Age	Salary
0	Alice	25.0	50000.0
1	Bob	25.0	50000.0
2	Charlie	35.0	70000.0
3	Dave	41.0	70000.0
4	Eve	41.0	60000.0

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [np.nan, 25, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000]}
df = pd.DataFrame(data)
df_ffilled = df.fillna(method='ffill')
print(df_ffilled)
```

	Name	Age	Salary
0	Alice	NaN	50000.0
1	Bob	25.0	50000.0
2	Charlie	35.0	70000.0
3	Dave	41.0	70000.0
4	Eve	41.0	60000.0

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, np.nan, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000]}
df = pd.DataFrame(data)
df_bfilled = df.fillna(method='bfill')
print(df_bfilled)
```

	Name	Age	Salary
0	Alice	25.0	50000.0
1	Bob	35.0	70000.0
2	Charlie	35.0	70000.0
3	Dave	41.0	60000.0
4	Eve	NaN	60000.0

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, np.nan, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, 60000, np.nan]}
df = pd.DataFrame(data)
df_bfilled = df.fillna(method='bfill')
print(df_bfilled)
```

	Name	Age	Salary
0	Alice	25.0	50000.0
1	Bob	35.0	70000.0
2	Charlie	35.0	70000.0
3	Dave	41.0	60000.0
4	Eve	NaN	NaN

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, np.nan, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000]}
df = pd.DataFrame(data)
df_mean = df.fillna(df.mean())
print(df_mean)
```

```
Name      Age   Salary
0   Alice  25.000000 50000.0
1     Bob  33.666667 60000.0
2  Charlie 35.000000 70000.0
3    Dave  41.000000 60000.0
4     Eve  33.666667 60000.0
<ipython-input-20-5507d3e63723>:7: FutureWarning: The default value of numeric_only in DataFrame.mean is deprecated
df_mean = df.fillna(df.mean())
```

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, np.nan, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000]}
df = pd.DataFrame(data)
df.fillna(method='ffill', inplace=True)
print(df)
```

```
Name      Age   Salary
0   Alice  25.0 50000.0
1     Bob  25.0 50000.0
2  Charlie 35.0 70000.0
3    Dave  41.0 70000.0
4     Eve  41.0 60000.0
```

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve'],
        'Age': [25, np.nan, 35, 41, np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000]}
df = pd.DataFrame(data)
df=df[df.duplicated()]
print(df)
```

```
Empty DataFrame
Columns: [Name, Age, Salary]
Index: []
```

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve','Eve'],
        'Age': [25, np.nan, 35, 41, np.nan,np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000,60000]}
df = pd.DataFrame(data)
df=df[df.duplicated()]
print(df)
```

```
Name  Age   Salary
5   Eve  NaN  60000.0
```

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve','Eve'],
        'Age': [25, np.nan, 35, 41, np.nan,np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000,60000]}
df = pd.DataFrame(data)
df.duplicated()
```

```
0    False
1    False
2    False
3    False
4    False
5    True
dtype: bool
```

```
import pandas as pd
import numpy as np
data = {'Name': ['Alice', 'Bob', 'Charlie', 'Dave', 'Eve','Eve'],
        'Age': [25, np.nan, 35, 41, np.nan,np.nan],
        'Salary': [50000, np.nan, 70000, np.nan, 60000,60000]}
df = pd.DataFrame(data)
~df.duplicated()
```

```
0    True
1    True
2    True
3    True
4    True
5   False
dtype: bool
```

```
import pandas as pd
df=pd.read_csv("data1.csv")
df
```

	Unnamed: 0	Name	Age	Salary
0	0	Alice	20.0	10000.0
1	1	Bob	NaN	20000.0
2	2	Charlie	16.0	NaN

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

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	previous
0	76	retired	married	secondary	no	2302.0	no							
1	66	retired	divorced	unknown	no	53.0	no							
2	51	management	married	tertiary	no	2455.0	yes							
3	41	blue-collar	married	secondary	no	356.0	yes							
4	51	technician	married	secondary	no	-1944.0	yes							
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4461	33	management	married	tertiary	no	133.0	yes							
4462	39	services	divorced	secondary	no	687.0	yes							
4463	40	admin.	single	secondary	no	2040.0	yes							
4464	31	technician	single	secondary	no	628.0	yes							
4465	70	retired	divorced	primary	no	383.0	no							

4466 rows × 17 columns

1. Select the rows where clients with primary education have subscribed to a deposit?

```
df.loc[(df['education']=='primary') & (df['deposit']=='yes')]
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	previous
29	39	blue-collar	divorced	primary	no	1317.0	yes	no	cellular	20	nov	543		
39	31	unemployed	single	primary	no	163.0	no	no	cellular	30	jan	707		
56	49	blue-collar	single	primary	no	566.0	yes	no	cellular	25	jul	979		
66	53	blue-collar	married	primary	yes	-462.0	no	no	cellular	29	jan	470		
103	42	blue-collar	single	primary	no	4930.0	no	no	unknown	18	jun	973		
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4411	55	housemaid	married	primary	no	0.0	yes	no	cellular	17	jul	1303		
4422	80	retired	married	primary	no	1468.0	no	no	cellular	13	jan	330		
4451	41	blue-collar	married	primary	no	143.0	yes	yes	unknown	2	jun	659		
4452	53	blue-collar	married	primary	no	421.0	yes	no	cellular	20	nov	677		
4458	32	blue-collar	married	primary	no	-454.0	yes	yes	cellular	18	may	801		

243 rows × 17 columns

2. Select the rows where clients who have not subscribed to a deposit?

```
df.loc[(df['deposit']=='no')]
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	camp
0	76	retired	married	secondary	no	2302.0	no	no	telephone	5	feb	110	
3	41	blue-collar	married	secondary	no	356.0	yes	no	cellular	14	may	90	
6	59	retired	married	secondary	no	136.0	no	no	cellular	6	aug	301	
7	34	blue-collar	married	primary	no	5299.0	yes	no	unknown	26	jun	75	
9	44	blue-collar	married	secondary	no	879.0	yes	no	cellular	3	apr	383	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4457	43	management	married	tertiary	no	1336.0	yes	yes	cellular	27	may	82	
4460	54	retired	married	secondary	no	522.0	no	yes	cellular	14	jul	81	
4461	33	management	married	tertiary	no	133.0	yes	no	unknown	26	may	308	
4464	31	technician	single	secondary	no	628.0	yes	no	unknown	12	may	1083	
4465	70	retired	divorced	primary	no	383.0	no	no	cellular	28	apr	50	

2354 rows × 17 columns

3. Select the rows where clients who have subscribed to a deposit either have a housing or a personal loan?

df.loc[(df['deposit']=='yes') &amp; ((df['housing']=='yes') | (df['loan']=='yes'))]

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	camp
2	51	management	married	tertiary	no	2455.0	yes	no	cellular	21	jul	553	
4	51	technician	married	secondary	no	-1944.0	yes	no	cellular	7	may	623	
15	37	management	single	tertiary	no	455.0	yes	no	cellular	13	aug	904	
17	24	admin.	single	tertiary	no	0.0	yes	no	cellular	27	may	122	
21	33	admin.	married	tertiary	no	79.0	yes	no	cellular	5	may	389	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4454	30	blue-collar	single	secondary	no	155.0	yes	yes	cellular	9	jul	1426	
4458	32	blue-collar	married	primary	no	-454.0	yes	yes	cellular	18	may	801	
4459	37	technician	single	secondary	no	3326.0	yes	no	unknown	21	may	799	
4462	39	services	divorced	secondary	no	687.0	yes	no	cellular	9	jul	869	
4463	40	admin.	single	secondary	no	2040.0	yes	no	cellular	18	may	906	

893 rows × 17 columns

4. Select the rows where clients with secondary education who have not subscribed to a deposit?

df.loc[(df['education']=='secondary') &amp; (df['deposit']=='no')]

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign
0	76	retired	married	secondary	no	2302.0	no	no	telephone	5	feb	110	
3	41	blue-collar	married	secondary	no	356.0	yes	no	cellular	14	may	90	
6	59	retired	married	secondary	no	136.0	no	no	cellular	6	aug	301	
9	44	blue-collar	married	secondary	no	879.0	yes	no	cellular	3	apr	383	
10	34	services	married	secondary	no	1637.0	yes	no	cellular	21	nov	107	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4446	35	services	married	secondary	no	0.0	yes	no	cellular	24	jul	810	
4453	31	services	married	secondary	no	505.0	no	no	cellular	11	jul	773	
4456	54	blue-collar	married	secondary	no	-102.0	yes	no	cellular	27	aug	164	
4460	54	retired	married	secondary	no	522.0	no	yes	cellular	14	jul	81	

5. Select the rows where clients who have subscribed to a term deposit as an outcome of the successful marketing campaign?

```
df.loc[(df['poutcome']=='success') & (df['deposit']=='yes')]
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campai
19	76	self-employed	married	unknown	no	4984.0	no	no	telephone	28	apr	403	
21	33	admin.	married	tertiary	no	79.0	yes	no	cellular	5	may	389	
45	71	retired	divorced	secondary	no	0.0	no	no	cellular	26	feb	771	
51	68	retired	married	secondary	no	1146.0	no	no	cellular	13	may	356	
52	46	management	married	tertiary	no	273.0	yes	no	cellular	18	mar	910	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4338	38	admin.	divorced	secondary	no	19.0	yes	no	cellular	5	feb	1130	
4372	20	student	single	secondary	no	215.0	no	no	cellular	24	feb	175	
4376	42	technician	married	secondary	no	994.0	yes	no	cellular	12	nov	227	
4408	29	housemaid	single	tertiary	no	19.0	no	no	cellular	4	may	268	
4448	27	blue-collar	single	secondary	no	535.0	no	no	cellular	16	aug	265	

6. Select the rows where unemployed clients who have not subscribed to deposit?

```
df.loc[(df['job']=='unemployed') & (df['deposit']=='no')]
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign
74	37	unemployed	single	secondary	no	48.0	no	no	cellular	31	jul	40	
304	48	unemployed	married	secondary	no	855.0	yes	no	unknown	9	may	241	
404	57	unemployed	married	primary	no	0.0	yes	no	cellular	20	nov	231	
464	47	unemployed	divorced	secondary	no	947.0	no	no	cellular	5	feb	124	
494	45	unemployed	married	tertiary	no	1148.0	no	no	cellular	19	nov	782	
550	55	unemployed	married	primary	no	8585.0	no	no	telephone	28	jan	201	
644	35	unemployed	single	secondary	no	2116.0	yes	no	cellular	13	may	409	
690	31	unemployed	single	unknown	no	167.0	no	no	cellular	20	nov	316	
705	42	unemployed	divorced	secondary	no	759.0	no	no	cellular	28	jan	97	
811	38	unemployed	married	secondary	no	995.0	no	no	unknown	26	jul	84	
821	43	unemployed	married	secondary	no	1943.0	yes	no	cellular	18	nov	202	
827	52	unemployed	married	secondary	no	1639.0	no	no	cellular	8	aug	93	
856	31	unemployed	married	secondary	no	20.0	no	no	telephone	18	nov	52	
966	42	unemployed	married	secondary	no	-165.0	yes	yes	unknown	27	may	179	
967	57	unemployed	married	secondary	no	1350.0	no	no	cellular	25	jun	131	
1003	35	unemployed	married	secondary	no	2080.0	yes	no	telephone	14	apr	266	
1123	50	unemployed	married	secondary	no	3478.0	yes	no	unknown	13	may	342	
1236	43	unemployed	divorced	secondary	no	1854.0	no	no	cellular	13	nov	131	
1351	38	unemployed	divorced	secondary	no	189.0	yes	yes	cellular	21	jul	105	