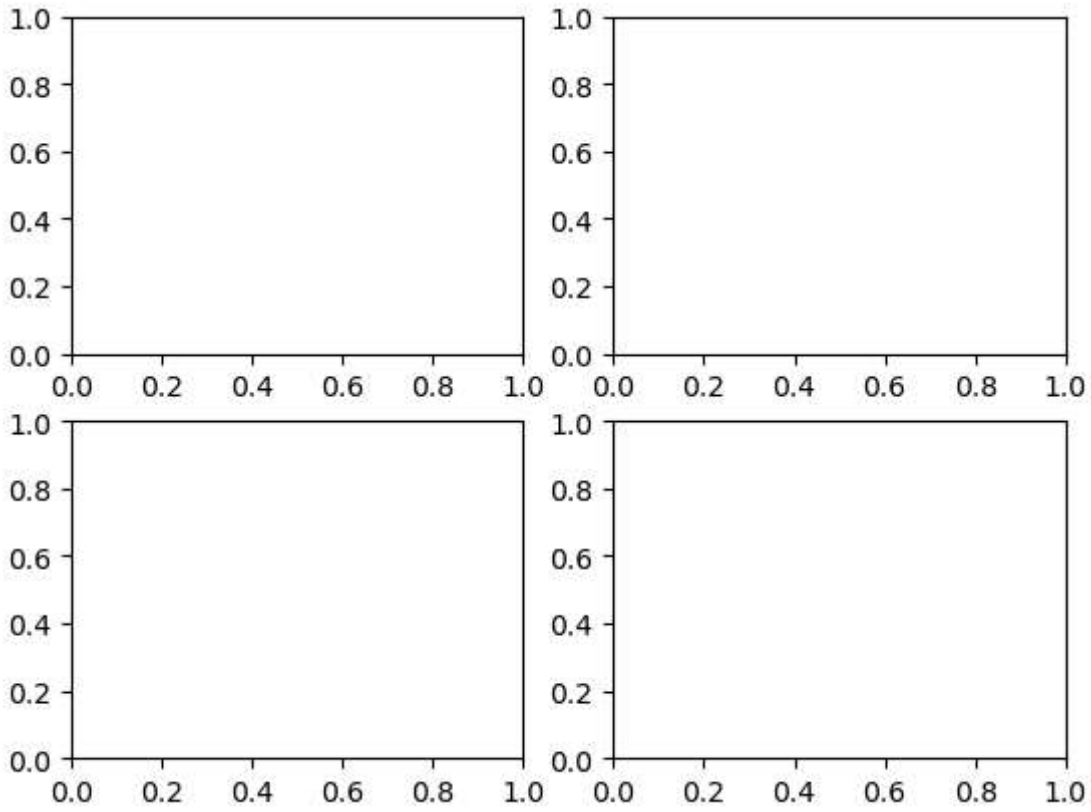
In [1]:
```
import pandas as pd                    #import packages
import matplotlib.pyplot as plt
import seaborn as sns
```

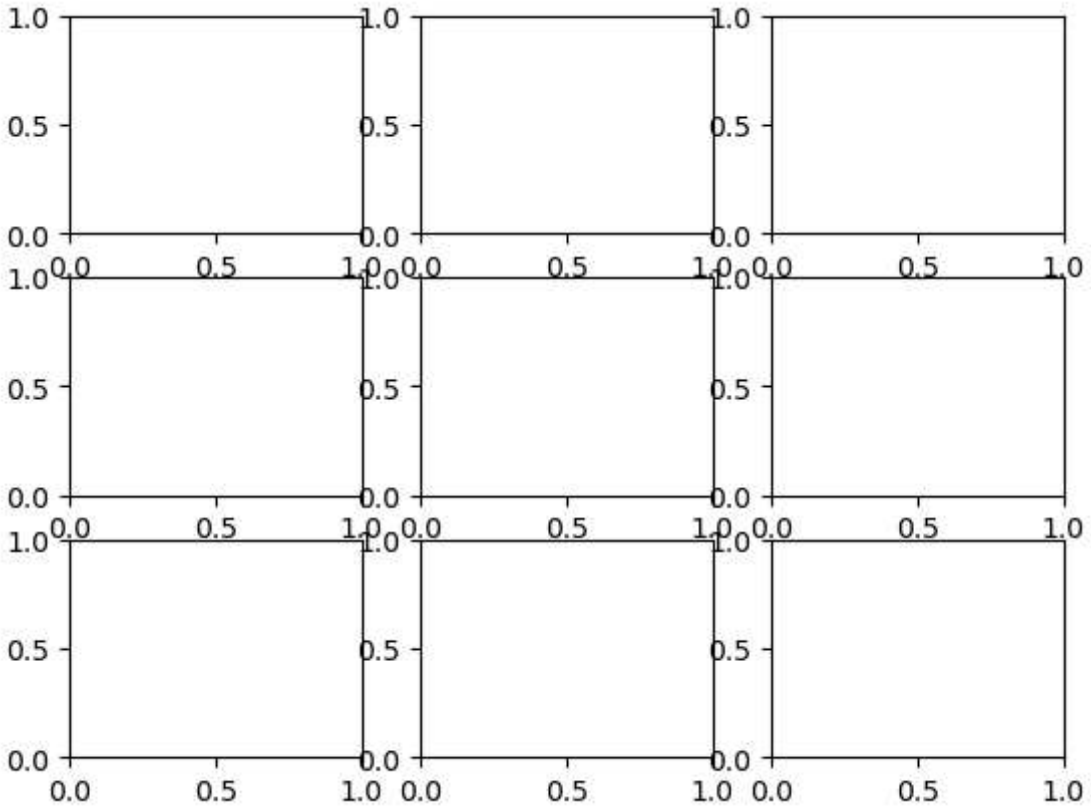
In [245…
```
#subplot 2x2

plt.subplot(2,2,1)   #subplot frist
plt.subplot(2,2,2)   #subplot second
plt.subplot(2,2,3)   #subplot third
plt.subplot(2,2,4)   #subplot forth
```

Out[245… <Axes: >



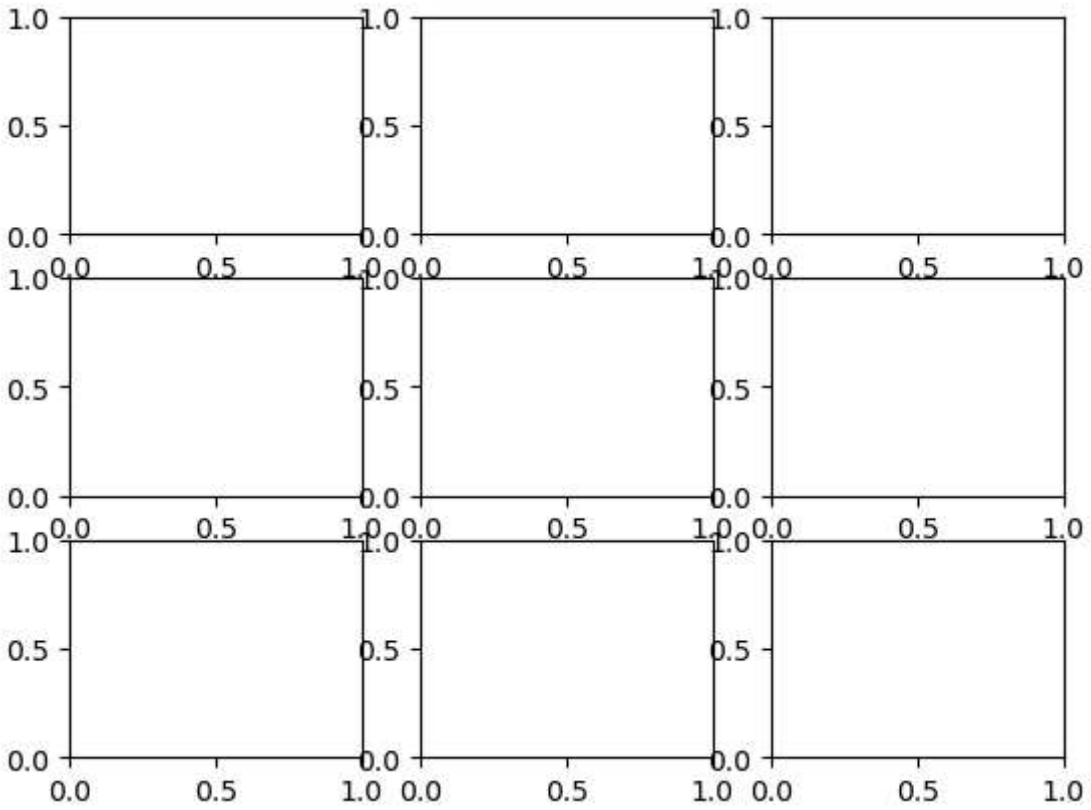
In [159…
```
#subplot 3x3

plt.subplot(3,3,1)
plt.subplot(3,3,2)
plt.subplot(3,3,3)
plt.subplot(3,3,4)
plt.subplot(3,3,5)
plt.subplot(3,3,6)
plt.subplot(3,3,7)
plt.subplot(3,3,8)
plt.subplot(3,3,9)
```

Out[159… <Axes: >



In [111…

```
# subplot using loop

for i in range(1,10):
    plt.subplot(3,3,i)
```



In [13]:
```
import os
os.getcwd()     #check directory
```

Out[13]:
```
'C:\\Users\\Mrityunjay\\Desktop\\Data science naresh it\\Self practice'
```

In [19]:
```
bank_df=pd.read_csv("bank.csv",sep=";")   #read datasets
bank_df
```

Out[19]:

	age	job	marital	education	default	balance	housing	loan	contact	da
0	30	unemployed	married	primary	no	1787	no	no	cellular	
1	33	services	married	secondary	no	4789	yes	yes	cellular	
2	35	management	single	tertiary	no	1350	yes	no	cellular	
3	30	management	married	tertiary	no	1476	yes	yes	unknown	
4	59	blue-collar	married	secondary	no	0	yes	no	unknown	
...	...	...	...	...	...	...	...	...	...	
4516	33	services	married	secondary	no	-333	yes	no	cellular	
4517	57	self-employed	married	tertiary	yes	-3313	yes	yes	unknown	
4518	57	technician	married	secondary	no	295	no	no	cellular	
4519	28	blue-collar	married	secondary	no	1137	no	no	cellular	
4520	44	entrepreneur	single	tertiary	no	1136	yes	yes	cellular	

4521 rows × 17 columns

In [21]:
```
bank_df.dtypes  #check datatypes
```

Out[21]:
```
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 [27]:
```
#make different list for categorical and numerical

cat_col=[keys for keys,value in dict(bank_df.dtypes).items() if value=="object"]
num_col=[keys for keys,value in dict(bank_df.dtypes).items() if value!="object"]
num_col
```

Out[27]:
```
['age', 'balance', 'day', 'duration', 'campaign', 'pdays', 'previous']
```

In [25]:
```
cat_col
```

Out[25]:
```
['job',
 'marital',
 'education',
 'default',
 'housing',
 'loan',
 'contact',
 'month',
 'poutcome',
 'y']
```

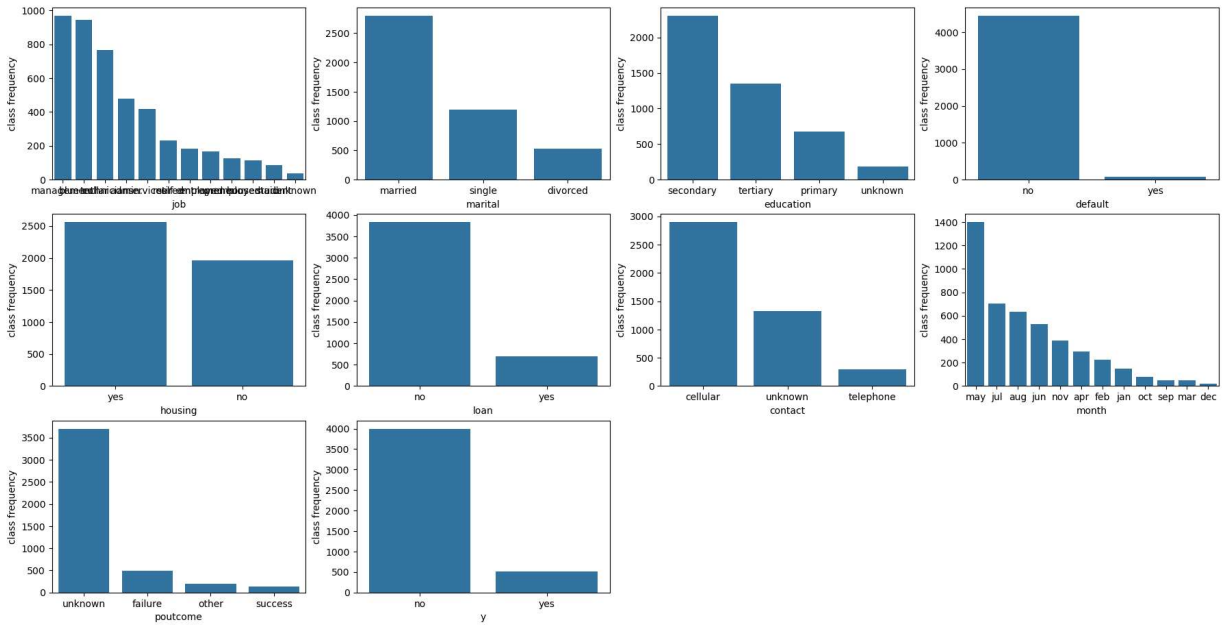
In [29]:
```
len(cat_col)
```

Out[29]:
```
10
```

In [81]:
```
bank_df["job"].value_counts().keys()    #
```

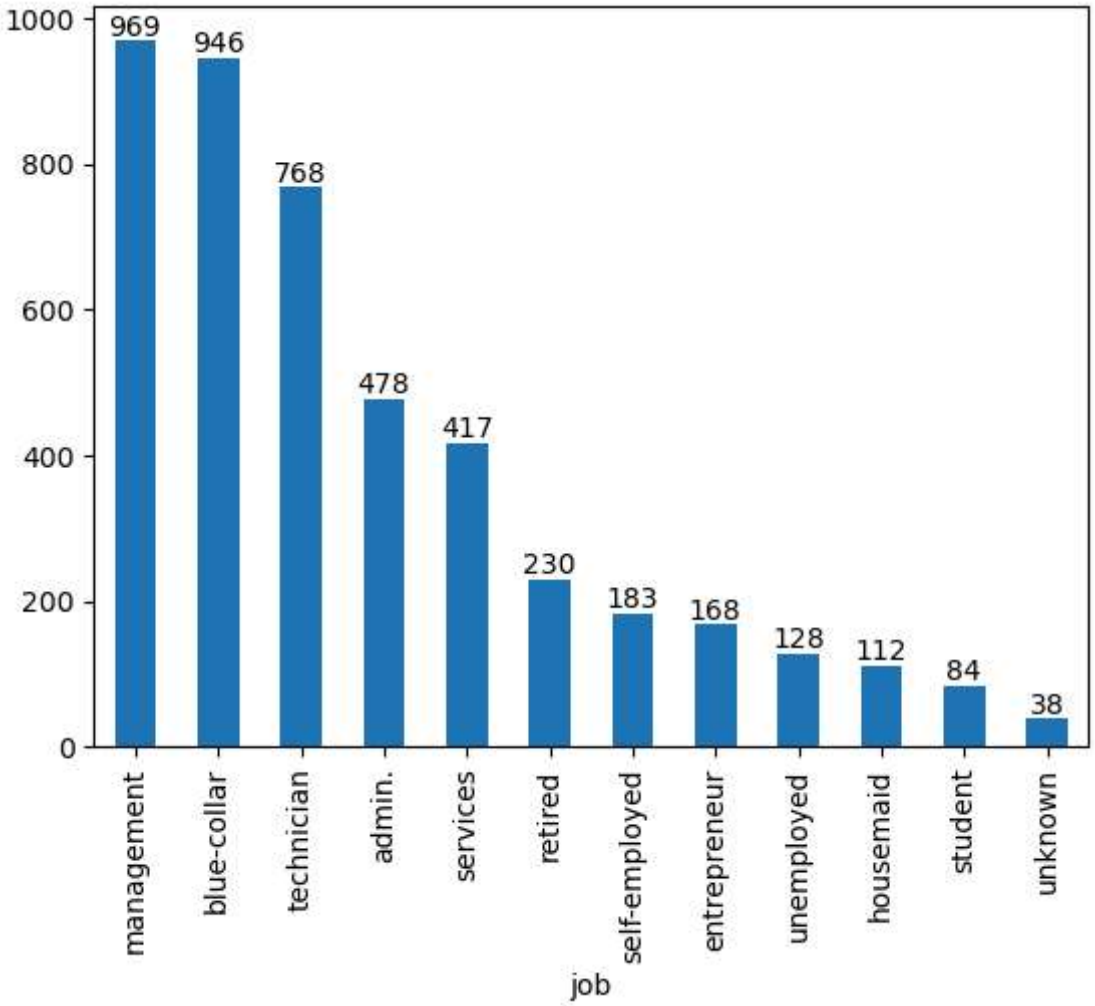
Out[81]:
```
Index(['management', 'blue-collar', 'technician', 'admin.', 'services',
       'retired', 'self-employed', 'entrepreneur', 'unemployed', 'housemaid',
       'student', 'unknown'],
      dtype='object', name='job')
```

In [163…
```
#plot bank categorical columns using subplot

plt.figure(figsize=(22,15))
for i in range(len(cat_col)):
    plt.subplot(4,4,i+1)
    sns.countplot(data=bank_df,
                  x=cat_col[i],
                  order=bank_df[cat_col[i]].value_counts().keys())
    plt.xlabel(f"{cat_col[i]}")
    plt.ylabel("class frequency")
```



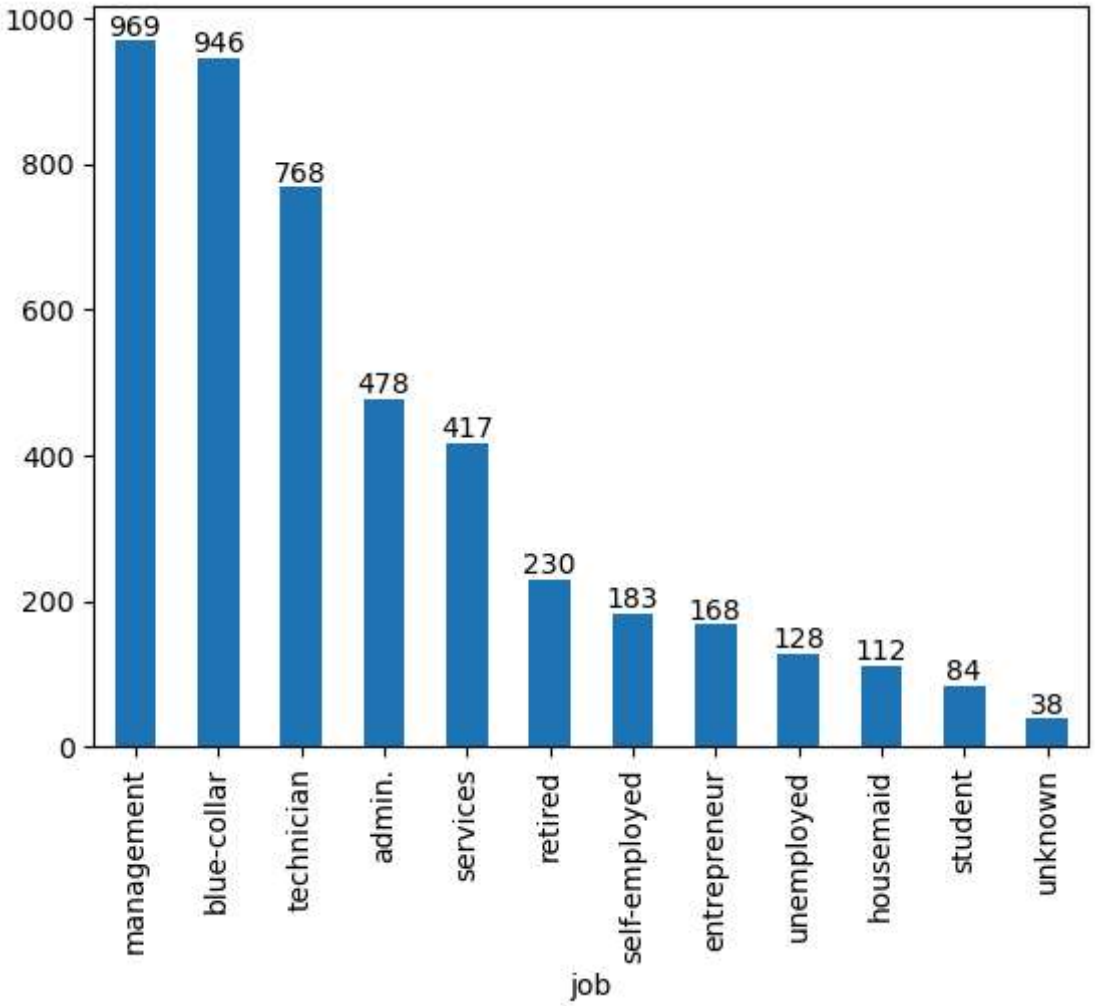
In [195…

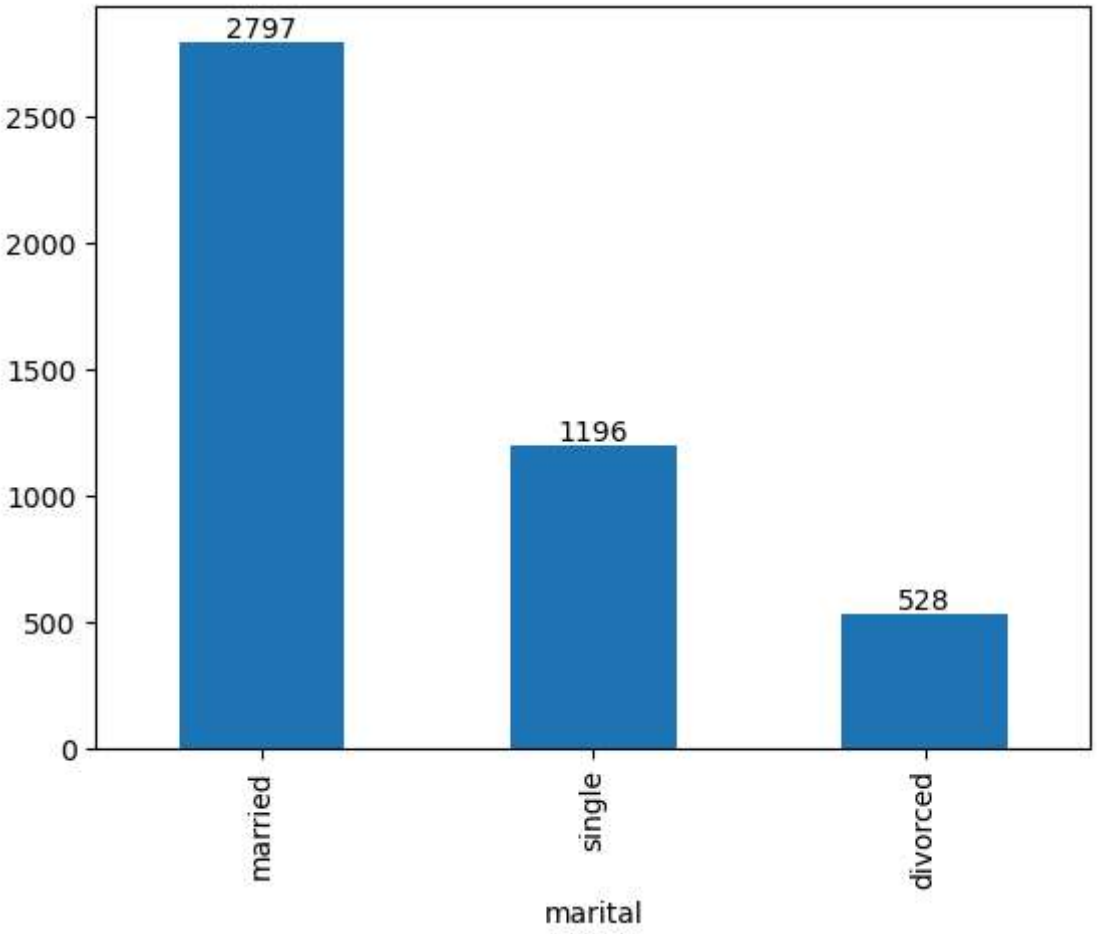
```
# plot bar chart

dnf=bank_df["job"].value_counts()
ax=dnf.plot(kind="bar")
ax.bar_label(ax.containers[0])
plt.show()
```

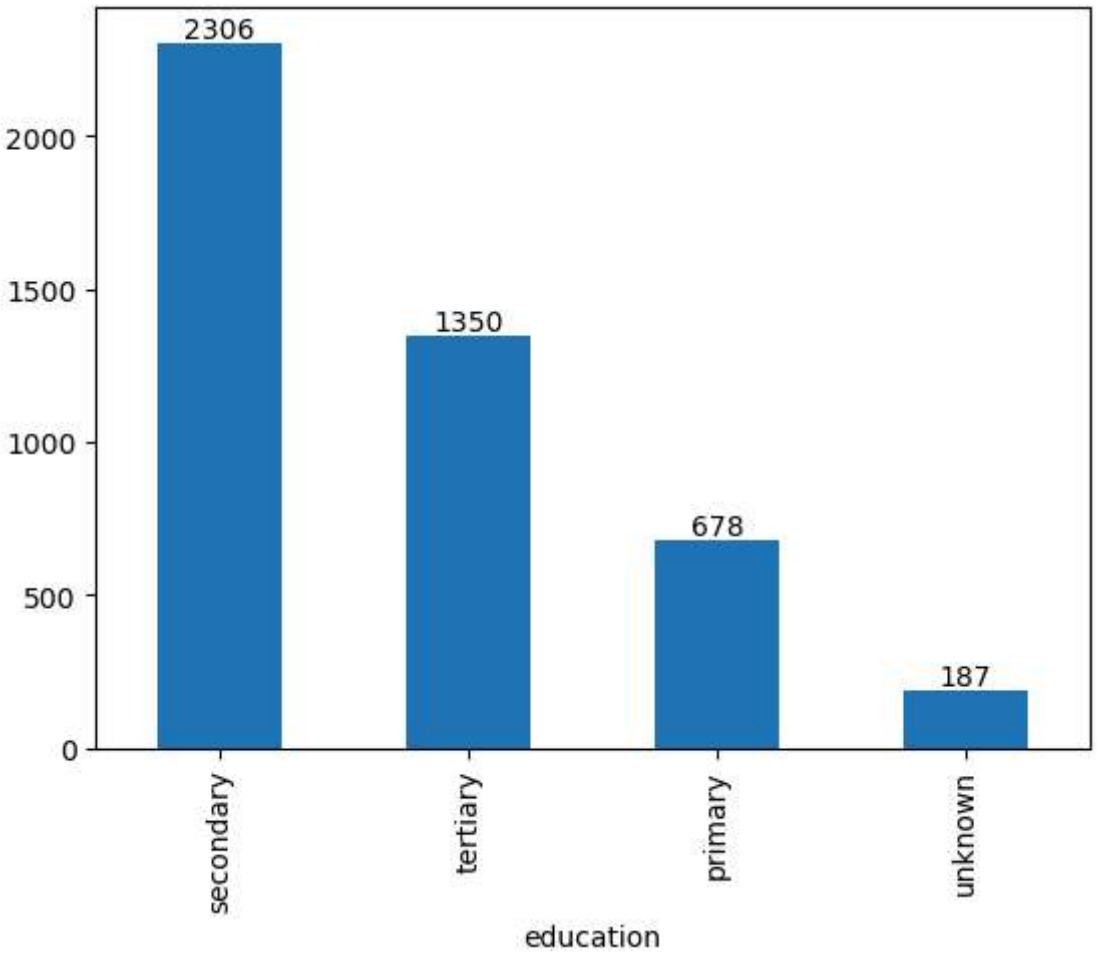


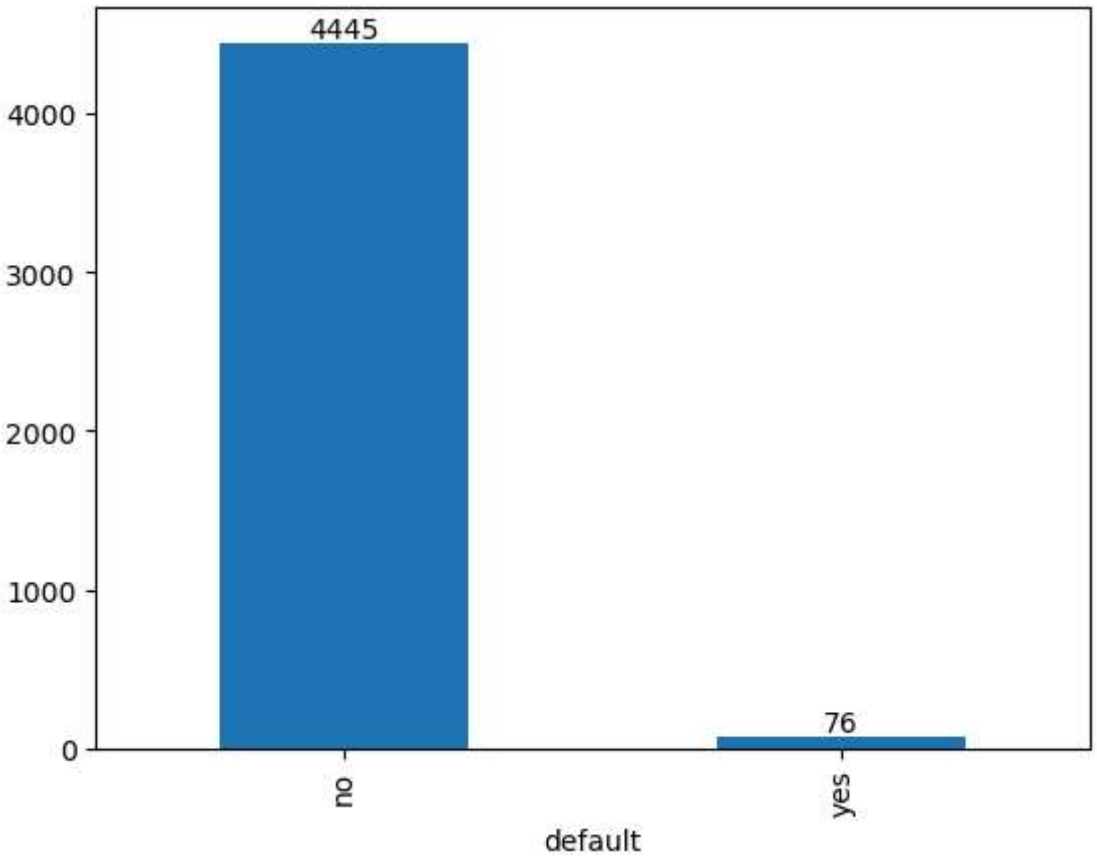
In [247…

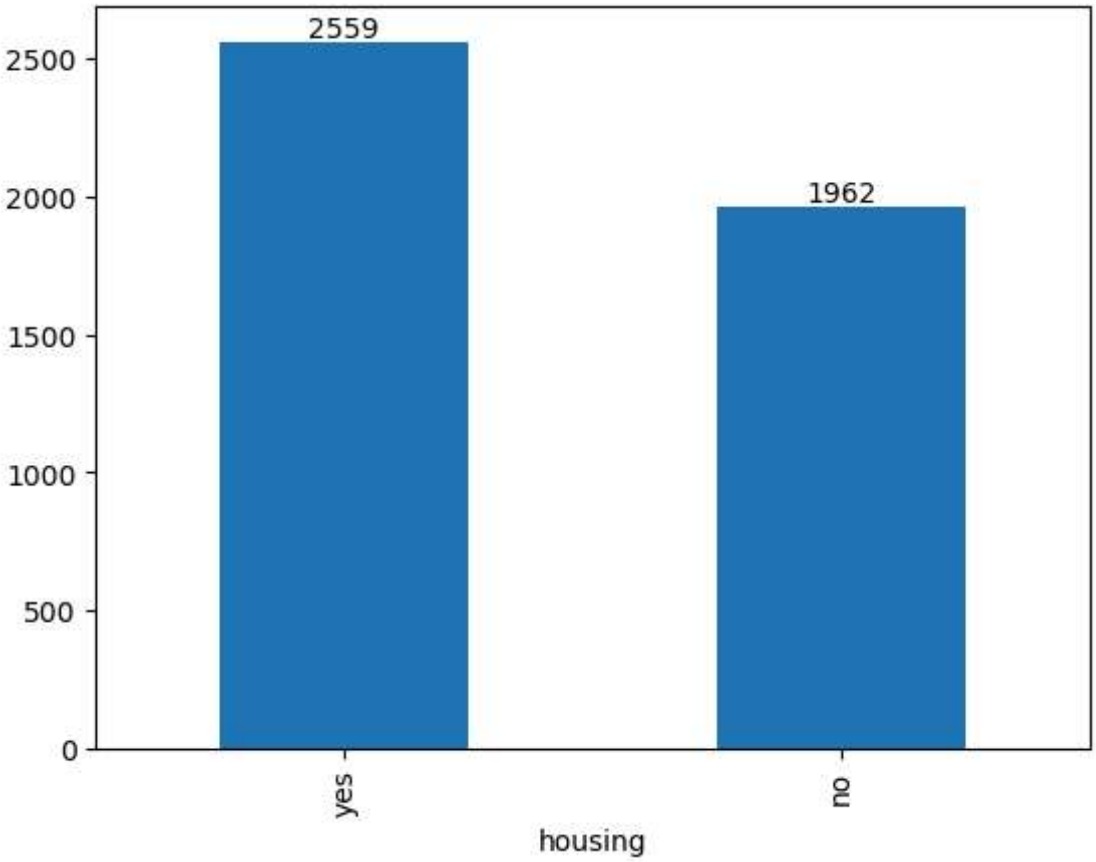
```
# plot bar charts

for i in cat_col:
    dnf=bank_df[i].value_counts()
    ax=dnf.plot(kind="bar")
    ax.bar_label(ax.containers[0])
    plt.show()
```

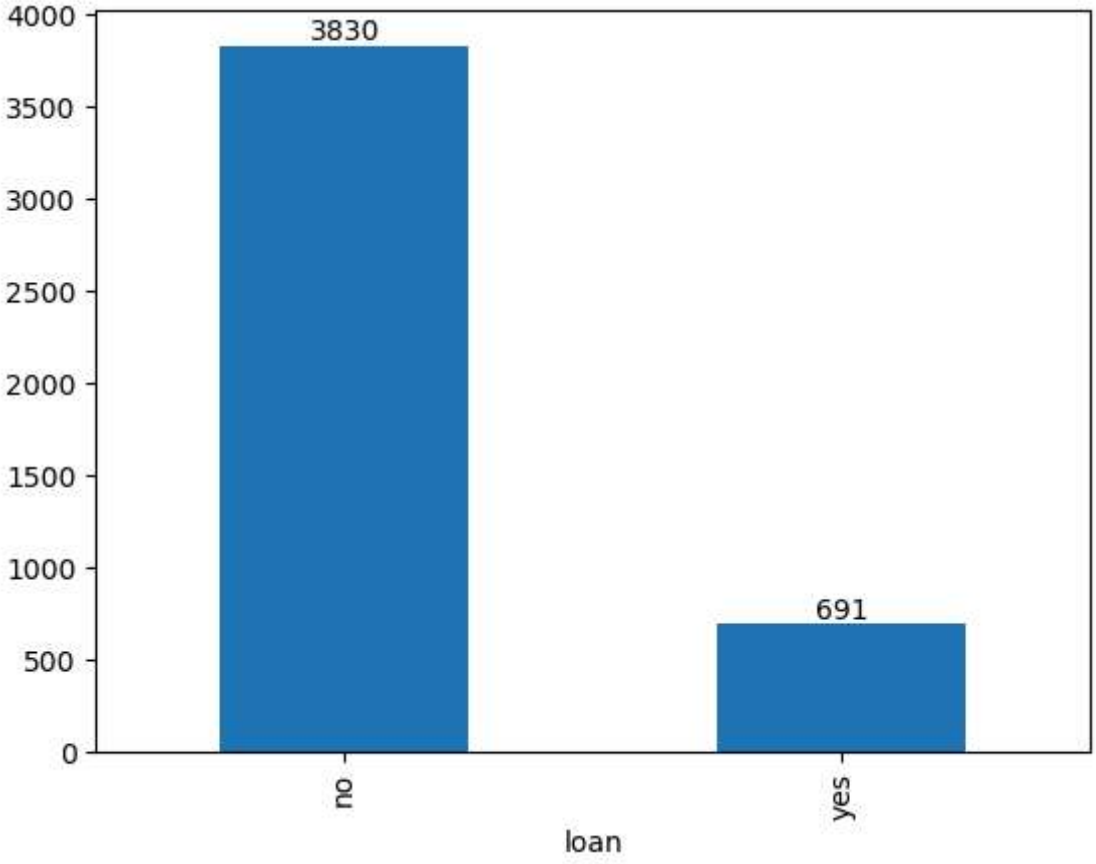

1000
800
600
400
200
0
969
946
768
478
417
230
183
168
128
112
84
38
management
blue-collar
technician
admin.
services
retired
self-employed
entrepreneur
unemployed
housemaid
student
unknown
job

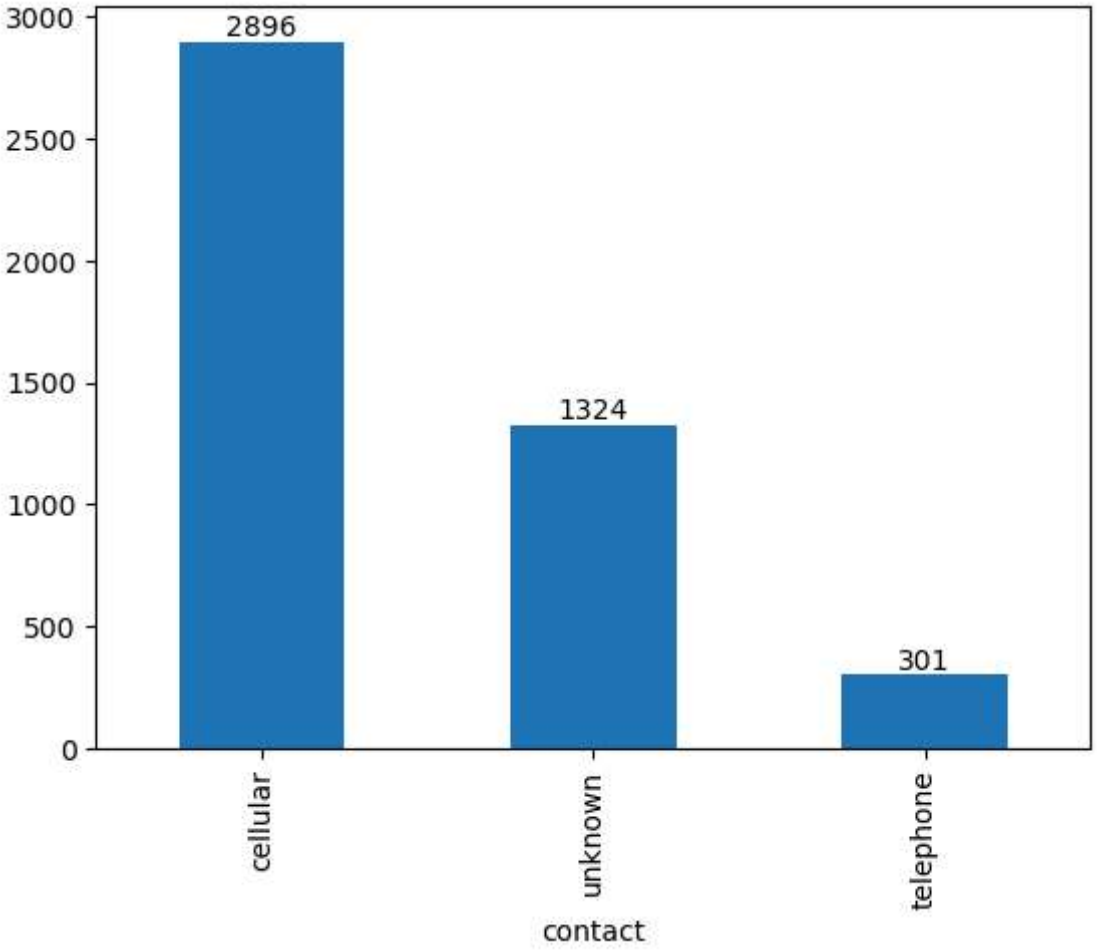

2797
1196
528
2500
2000
1500
1000
500
0
married
single
divorced
marital

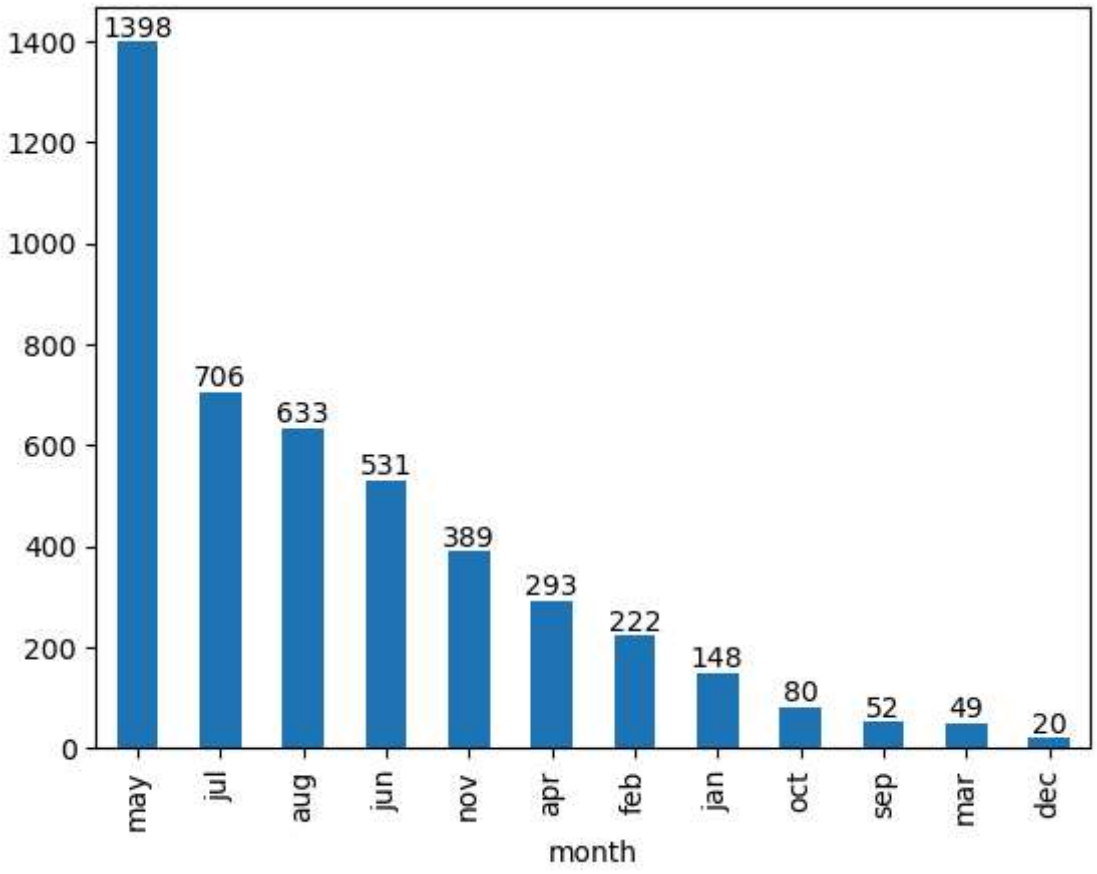

2306
1350
678
187
2000
1500
1000
500
0
secondary
tertiary
primary
unknown
education

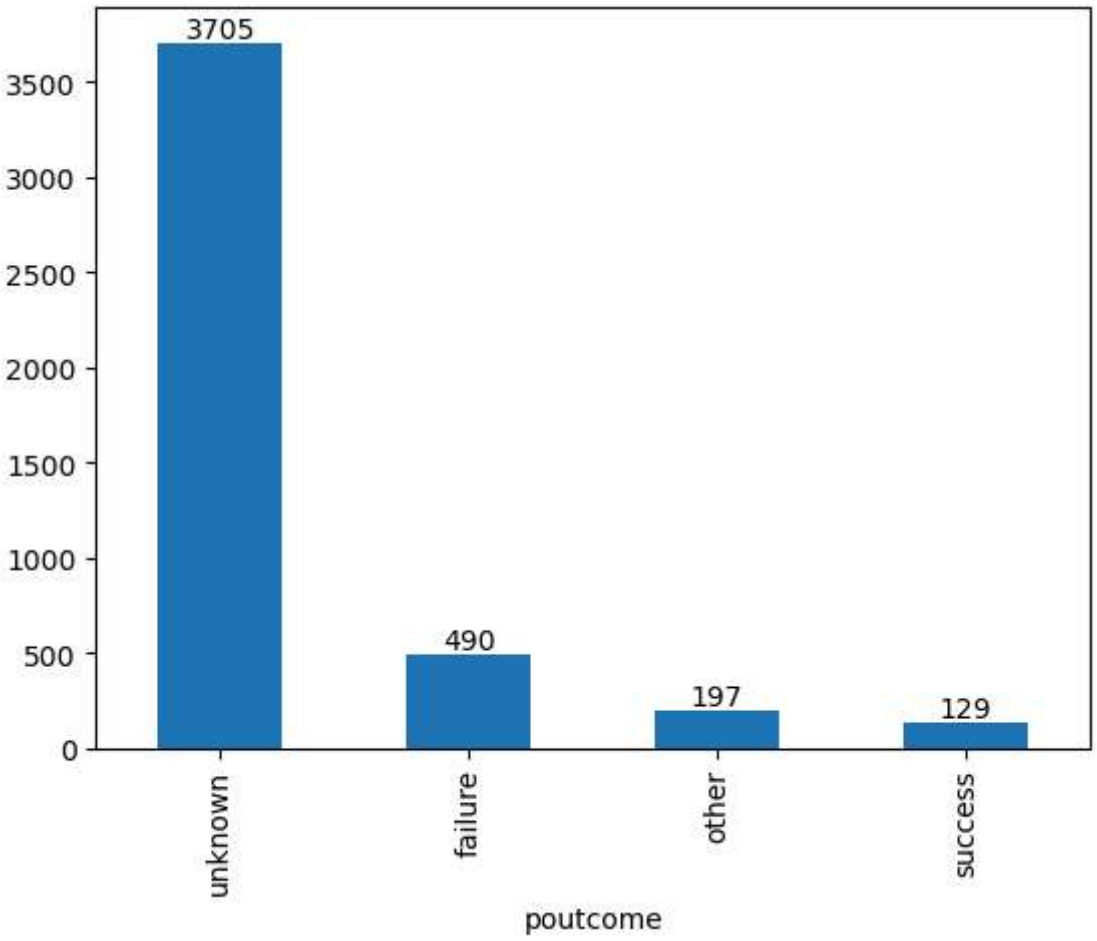

4445
76
4000
3000
2000
1000
0
no
yes
default

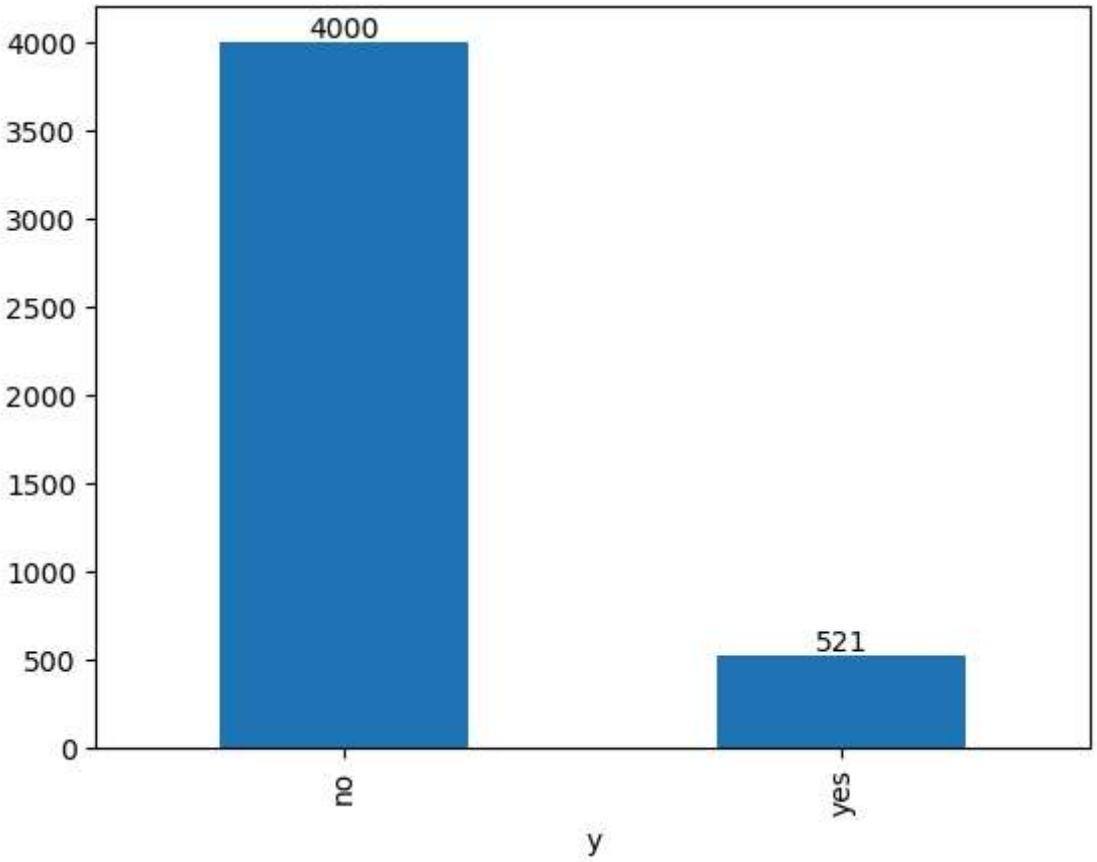

2559
1962
2500
2000
1500
1000
500
0
yes
no
housing


3830
691
4000
3500
3000
2500
2000
1500
1000
500
0
no
yes
loan


3000
2500
2000
1500
1000
500
0
2896
1324
301
cellular
unknown
telephone
contact


1400
1200
1000
800
600
400
200
0
1398
706
633
531
389
293
222
148
80
52
49
20
may
jul
aug
jun
nov
apr
feb
jan
oct
sep
mar
dec
month

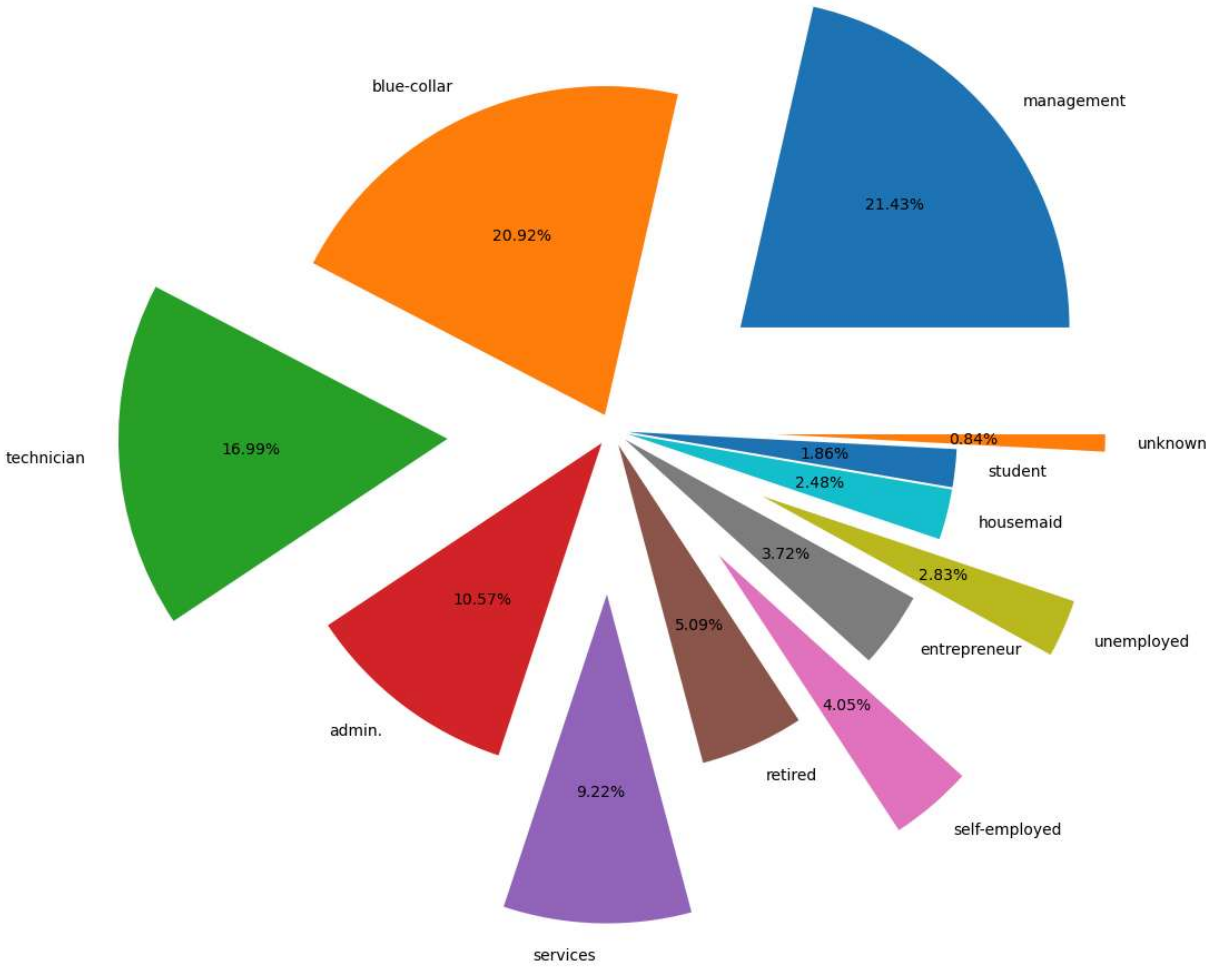

3705
490
197
129
3500
3000
2500
2000
1500
1000
500
0
unknown
failure
other
success
poutcome


4000
521
4000
3500
3000
2500
2000
1500
1000
500
0
no
yes
y

In [223…
```
bank_df["job"].value_counts()
```

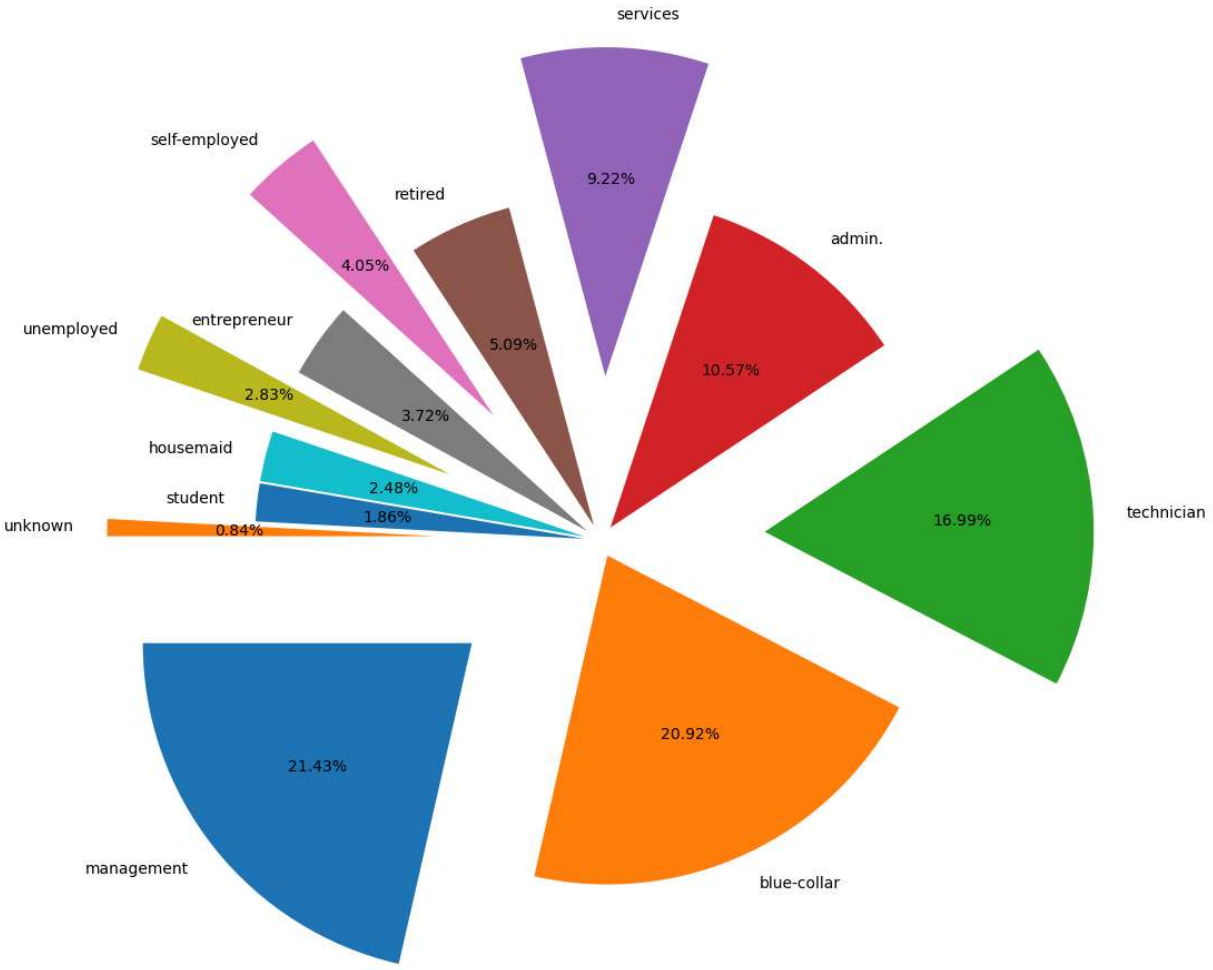
Out[223…
```
job
management       969
blue-collar      946
technician       768
admin.           478
services         417
retired          230
self-employed    183
entrepreneur     168
unemployed       128
housemaid        112
student           84
unknown           38
Name: count, dtype: int64
```

In [241…
```
#plot pie charts
dnf=bank_df["job"].value_counts()
keys=dnf.keys()
value=dnf.values
plt.pie(x=value,
        labels=keys,
       autopct="%0.2f%%",
       radius=2,
       explode=[1,0.1,1,0.1,1,0.1,1,0.1,1,0.1,0.1,1])
plt.show()
```



In [243…

```
#pie
dnf=bank_df["job"].value_counts()
keys=dnf.keys()
value=dnf.values
plt.pie(x=value,
        labels=keys,
       autopct="%0.2f%%",
       radius=2,
       explode=[1,0.1,1,0.1,1,0.1,1,0.1,1,0.1,0.1,1],startangle=180
       )
plt.show()
```


services
9.22%
admin.
10.57%
technician
16.99%
blue-collar
20.92%
management
21.43%
unknown
0.84%
student
1.86%
housemaid
2.48%
unemployed
2.83%
entrepreneur
3.72%
self-employed
4.05%
retired
5.09%

In [ ]: