Bank Telemarketing

In [1]:

```
#import data by kaggle
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
```

In [2]:

```
!kaggle datasets download -d gobert/bank-telemarketing-moro-et-al
```

```
Warning: Your Kaggle API key is readable by other users on this sy
stem! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.jso
n'
Downloading bank-telemarketing-moro-et-al.zip to /content
  0% 0.00/58.1k [00:00<?, ?B/s]
100% 58.1k/58.1k [00:00<00:00, 29.1MB/s]
```

In [3]:

```
#file unzip
import zipfile
zip_ref = zipfile.ZipFile('/content/bank-telemarketing-moro-et-al.z
zip_ref.extractall('/content')
zip_ref.close()
```

In [62]:

```
#import labriby
import numpy as np
import pandas as pd
import seaborn as sns
from matplotlib import rcParams
import matplotlib.pyplot as plt
from sklearn import preprocessing
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
```

In [5]:

```
#upload dataset in pandas dataframe
data = pd.read_csv('/content/bank-additional.csv')
```

In [6]:
```
#check first five rows of the dataset
data.head()
```

Out[6]:

	age	job	marital	education	default	housing	loan	contact	month	da
0	30	blue-collar	married	basic.9y	no	yes	no	cellular	may	
1	39	services	single	high.school	no	no	no	telephone	may	
2	25	services	married	high.school	no	yes	no	telephone	jun	
3	38	services	married	basic.9y	no	unknown	unknown	telephone	jun	
4	47	admin.	married	university.degree	no	yes	no	cellular	nov	

5 rows × 21 columns

In [7]:
```
#check last five rows of the dataset
data.tail()
```

Out[7]:

	age	job	marital	education	default	housing	loan	contact	month	day
4114	30	admin.	married	basic.6y	no	yes	yes	cellular	jul	
4115	39	admin.	married	high.school	no	yes	no	telephone	jul	
4116	27	student	single	high.school	no	no	no	cellular	may	
4117	58	admin.	married	high.school	no	no	no	cellular	aug	
4118	34	management	single	high.school	no	yes	no	cellular	nov	

5 rows × 21 columns

In [8]:
```
#check shape of the dataset
data.shape
```

Out[8]:
```
(4119, 21)
```

In [9]:
```
#check more infomation of the dataset
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4119 entries, 0 to 4118
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             4119 non-null   int64  
 1   job             4119 non-null   object 
 2   marital         4119 non-null   object 
 3   education       4119 non-null   object 
 4   default         4119 non-null   object 
 5   housing         4119 non-null   object 
 6   loan            4119 non-null   object 
 7   contact         4119 non-null   object 
 8   month           4119 non-null   object 
 9   day_of_week     4119 non-null   object 
 10  duration        4119 non-null   int64  
 11  campaign        4119 non-null   int64  
 12  pdays           4119 non-null   int64  
 13  previous        4119 non-null   int64  
 14  poutcome        4119 non-null   object 
 15  emp.var.rate    4119 non-null   float64
 16  cons.price.idx  4119 non-null   float64
 17  cons.conf.idx   4119 non-null   float64
 18  euribor3m       4119 non-null   float64
 19  nr.employed     4119 non-null   float64
 20  y               4119 non-null   object 
dtypes: float64(5), int64(5), object(11)
memory usage: 675.9+ KB
```

In [10]:
```
#check mathamatic relation of the dataset
data.describe()
```

Out[10]:

	age	duration	campaign	pdays	previous	emp.var.rate	con
count	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	41
mean	40.113620	256.788055	2.537266	960.422190	0.190337	0.084972	
std	10.313362	254.703736	2.568159	191.922786	0.541788	1.563114	
min	18.000000	0.000000	1.000000	0.000000	0.000000	-3.400000	
25%	32.000000	103.000000	1.000000	999.000000	0.000000	-1.800000	
50%	38.000000	181.000000	2.000000	999.000000	0.000000	1.100000	
75%	47.000000	317.000000	3.000000	999.000000	0.000000	1.400000	
max	88.000000	3643.000000	35.000000	999.000000	6.000000	1.400000	

In [11]:
```
#check corr realtion of the dataset
data.corr()
```

Out[11]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.pric
age	1.000000	0.041299	-0.014169	-0.043425	0.050931	-0.019192	-0.00
duration	0.041299	1.000000	-0.085348	-0.046998	0.025724	-0.028848	0.0
campaign	-0.014169	-0.085348	1.000000	0.058742	-0.091490	0.176079	0.1
pdays	-0.043425	-0.046998	0.058742	1.000000	-0.587941	0.270684	0.0
previous	0.050931	0.025724	-0.091490	-0.587941	1.000000	-0.415238	-0.1
emp.var.rate	-0.019192	-0.028848	0.176079	0.270684	-0.415238	1.000000	0.7
cons.price.idx	-0.000482	0.016672	0.145021	0.058472	-0.164922	0.755155	1.0
cons.conf.idx	0.098135	-0.034745	0.007882	-0.092090	-0.051420	0.195022	0.0
euribor3m	-0.015033	-0.032329	0.159435	0.301478	-0.458851	0.970308	0.6
nr.employed	-0.041936	-0.044218	0.161037	0.381983	-0.514853	0.897173	0.4

In [12]:
```
#check missing value of the dataset
data.isnull().sum()
```

Out[12]:
```
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome          0
emp.var.rate      0
cons.price.idx    0
cons.conf.idx     0
euribor3m         0
nr.employed       0
y                 0
dtype: int64
```

In [13]:
```
#check duplicated value in dataset
data.duplicated().sum()
```

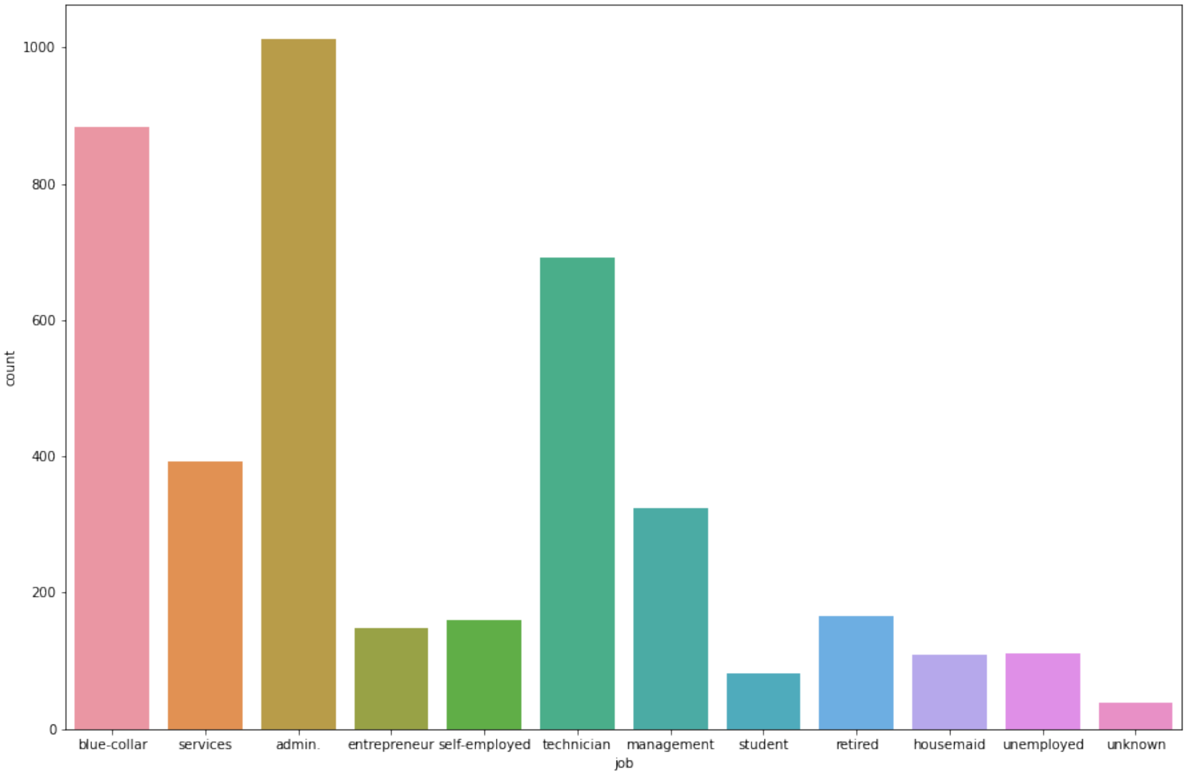
Out[13]:
```
0
```

EDA OF the Dataset

In [14]:
```
#count the value of job
rcParams['figure.figsize'] = 15,10
sns.countplot(data['job'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

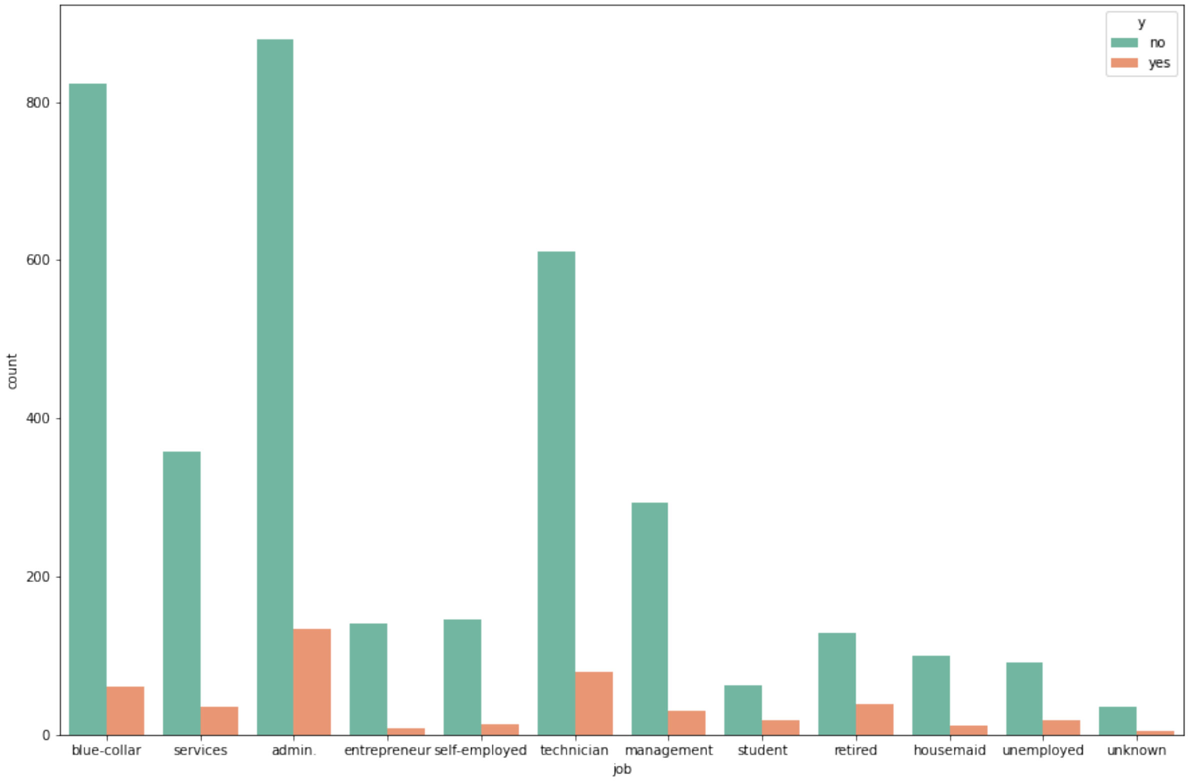
Out[14]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc8729310>
```



In [15]:
```
#count the value of job
rcParams['figure.figsize'] = 15,10
sns.countplot(data['job'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

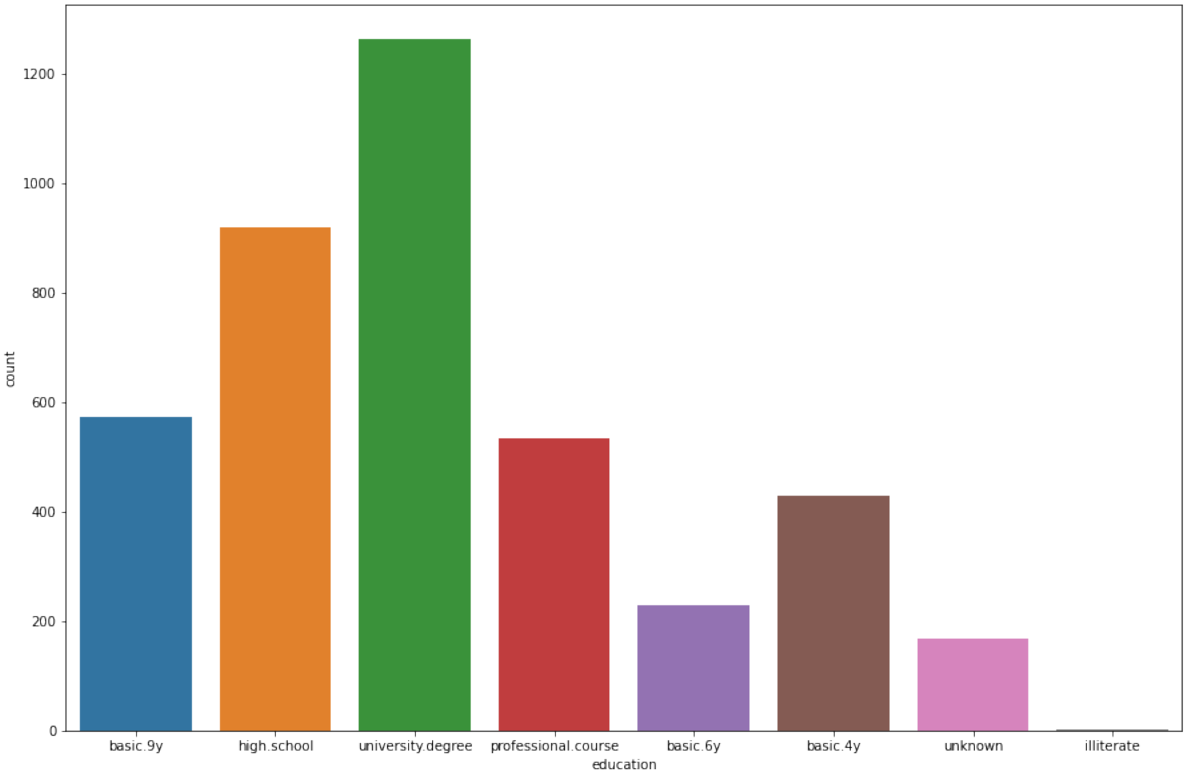
Out[15]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc87dc510>
```



In [16]:
```
#This infers that admin and technician are mostly taking the bank d
rcParams['figure.figsize'] = 15,10
sns.countplot(data['education'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

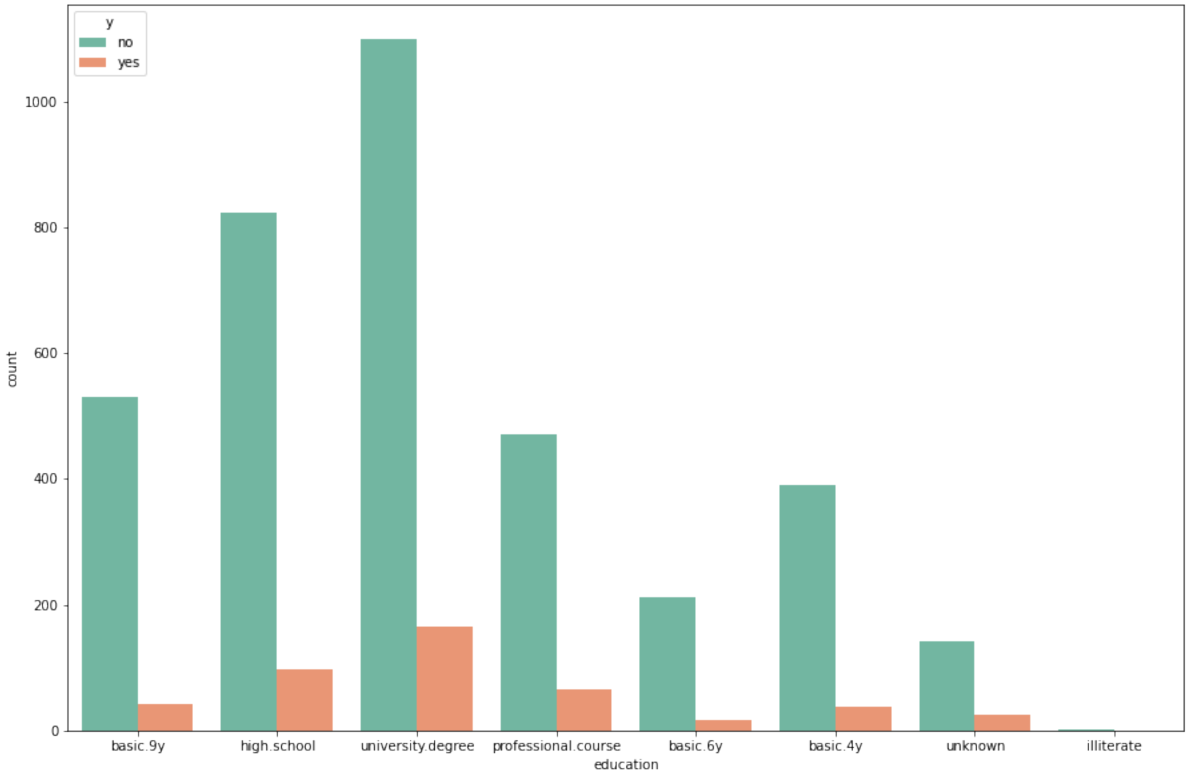
Out[16]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc80e1410>
```



In [17]:
```
#count the value education
rcParams['figure.figsize'] = 15,10
sns.countplot(data['education'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc807be50>
```

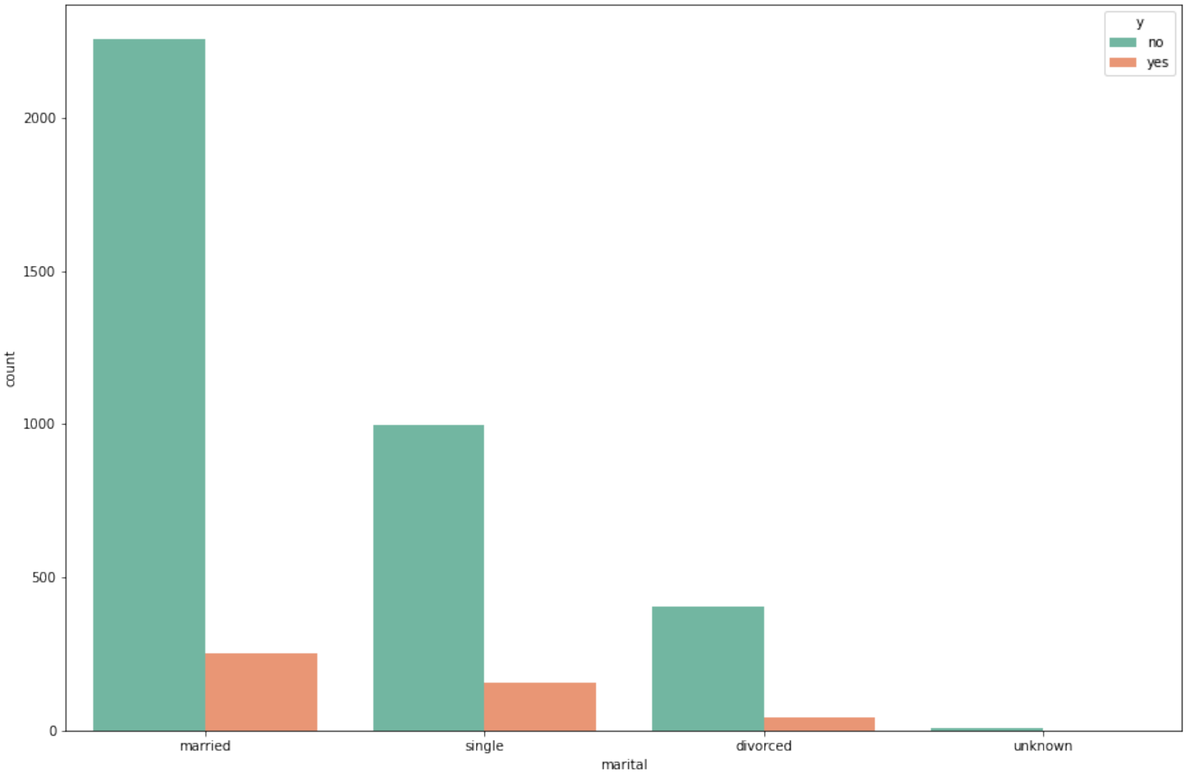


persons who have university degree and high school are getting the bank deposit

In [18]:
```
#count the value maritak
rcParams['figure.figsize'] = 15,10
sns.countplot(data['marital'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

Out[18]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7fc61d0>
```



Married and single people are accepting the bank deposit

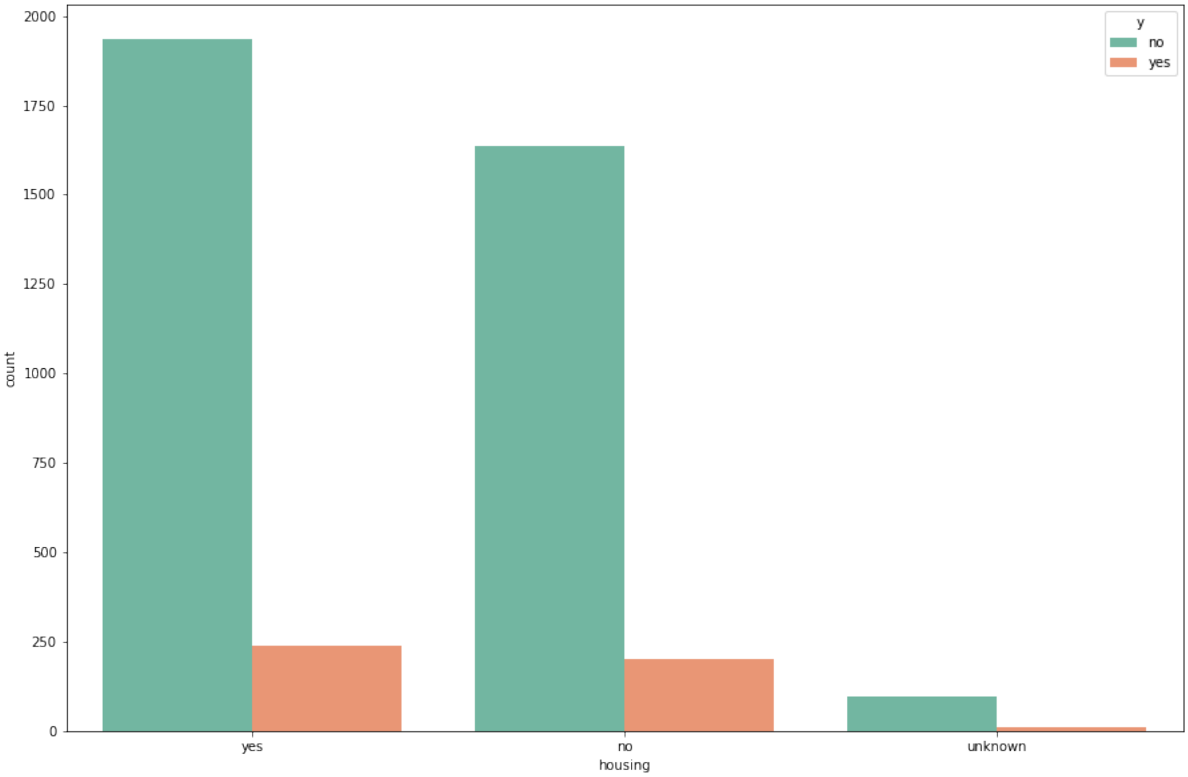
In [19]:

```
#count the values housing
rcParams['figure.figsize'] = 15,10
sns.countplot(data['housing'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

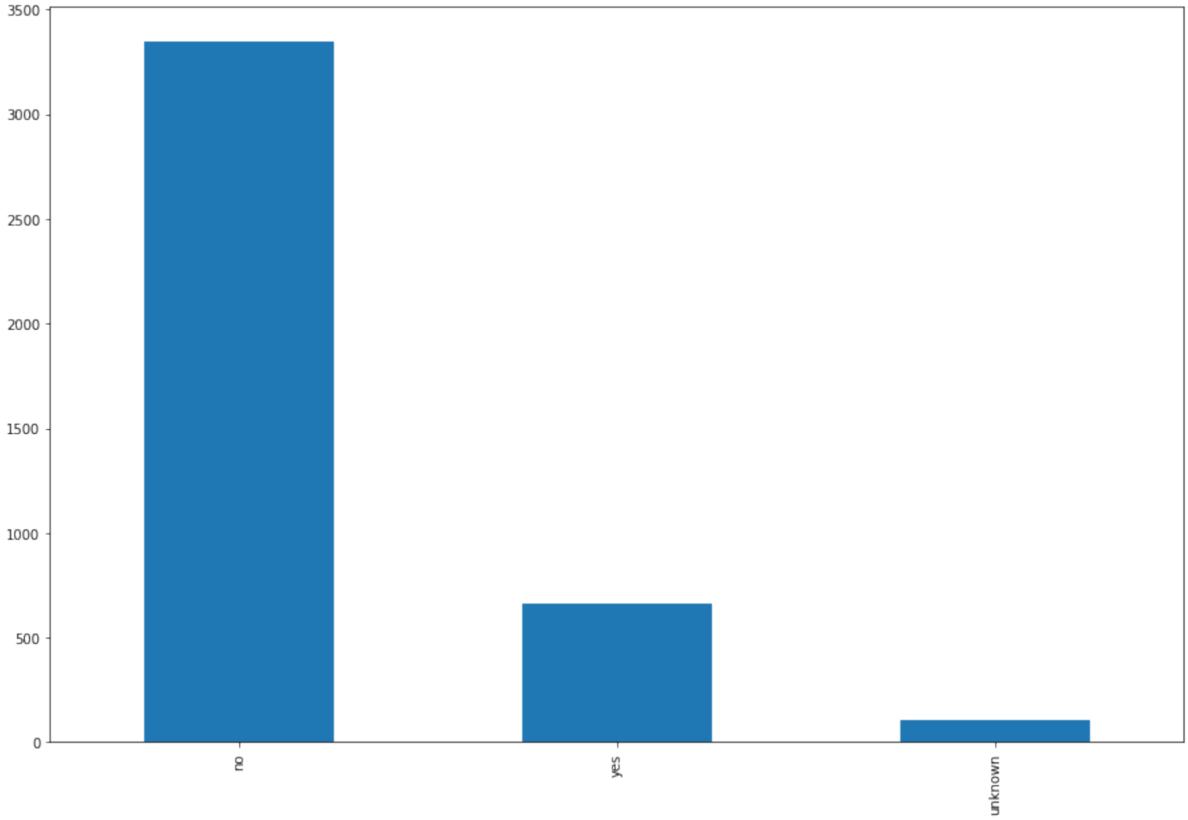
Out[19]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7f3fed0>
```



```
In [20]: #It shows those who have hosuing loan are more tend to accept the b
         data['loan'].value_counts().plot(kind="bar")
```

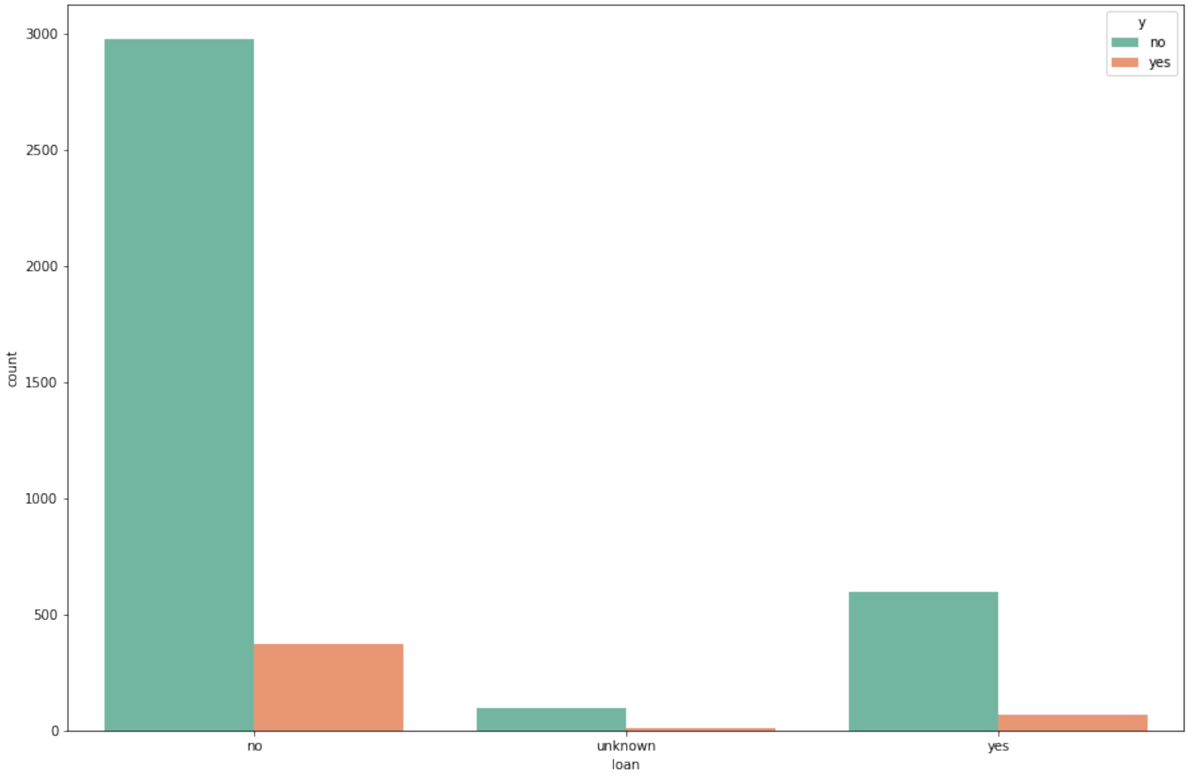
```
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x7efdc7e14210>
```



In [21]:
```
#cout the loan value
rcParams['figure.figsize'] = 15,10
sns.countplot(data['loan'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

Out[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7e0e650>
```



Personal Loan

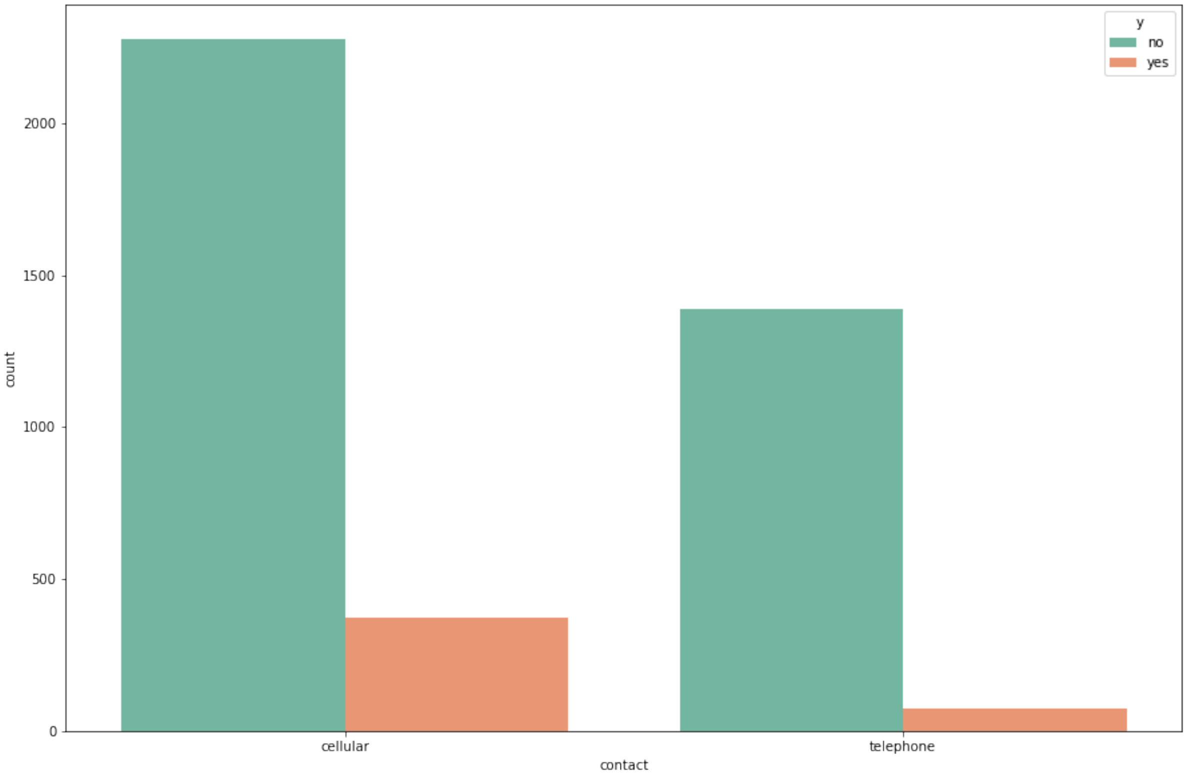
In [22]:

```
#The person who has no personal loan will subscribe the bank deposi
rcParams['figure.figsize'] = 15,10
sns.countplot(data['contact'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

Out[22]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7d82190>
```



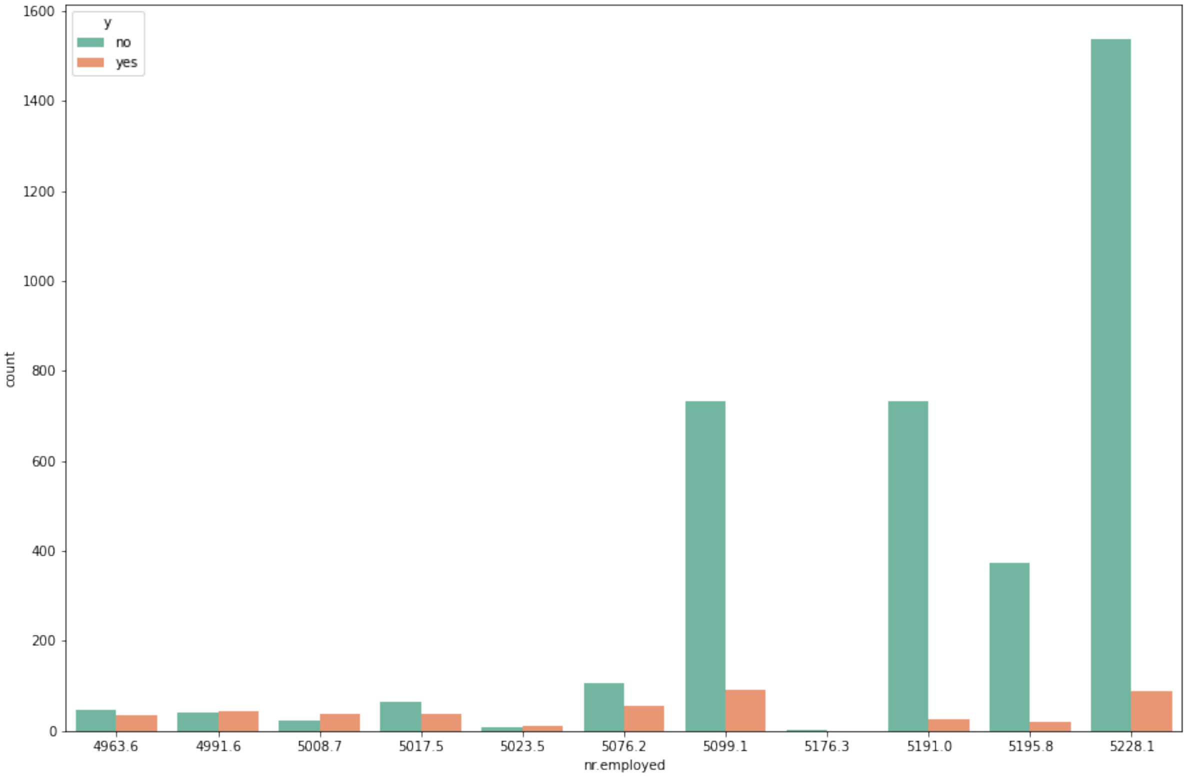
In [23]:

```
#count the value nr.employed
rcParams['figure.figsize'] = 15,10
sns.countplot(data['nr.employed'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

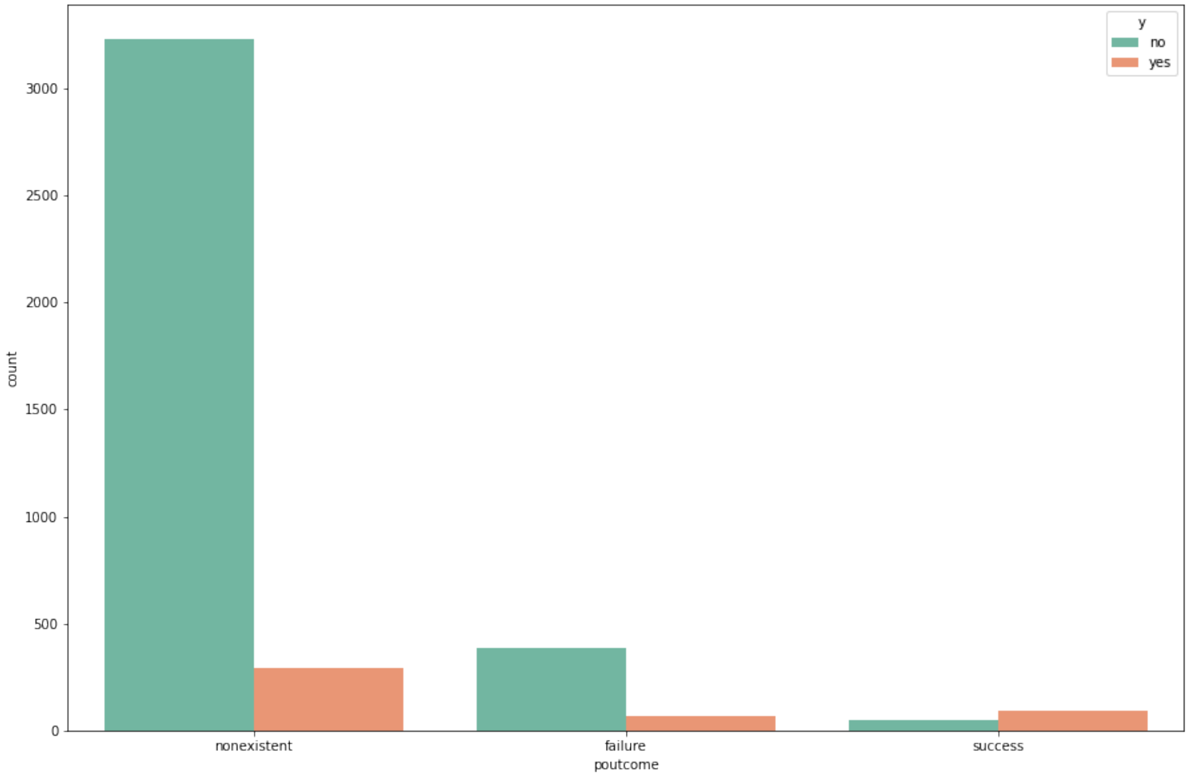
Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7d01790>
```

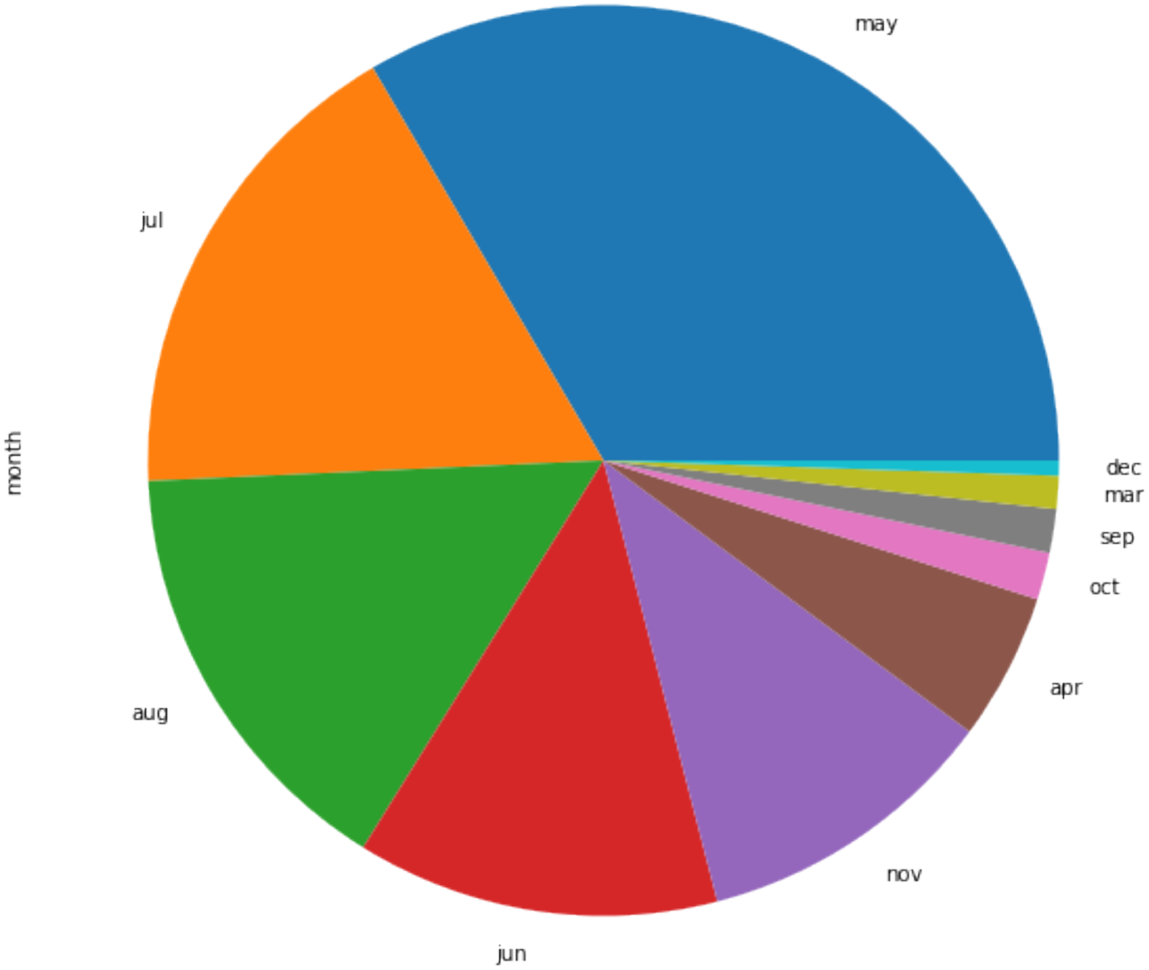


In [24]:
```
#count the value poutcome
rcParams['figure.figsize'] = 15,10
sns.countplot(x=data['poutcome'],hue=data['y'],palette="Set2")
```

Out[24]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7cf7190>
```



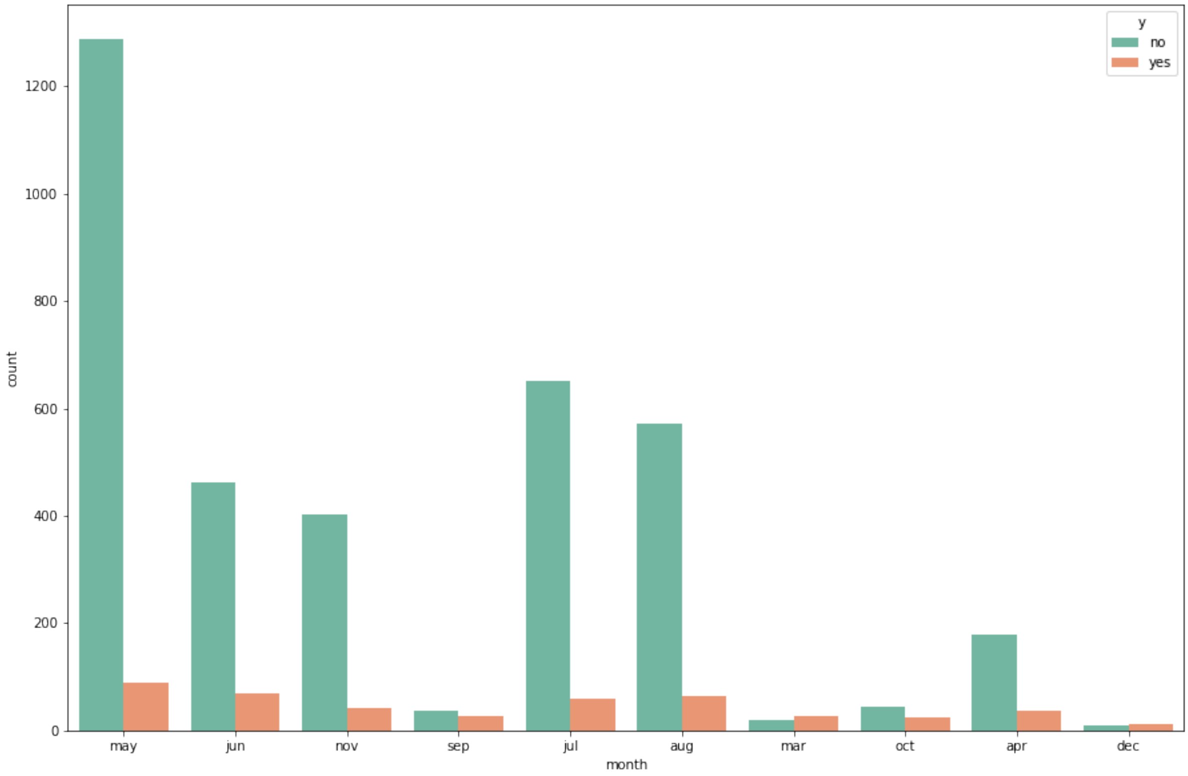
```
In [25]: #nonexistant people are more exposed for the subscripton of the ban
         data['month'].value_counts().plot(kind="pie")

Out[25]: <matplotlib.axes._subplots.AxesSubplot at 0x7efdc7ba4350>
```



In [26]:
```
#count the values month
rcParams['figure.figsize'] = 15,10
sns.countplot(data['month'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

Out[26]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc80853d0>
```

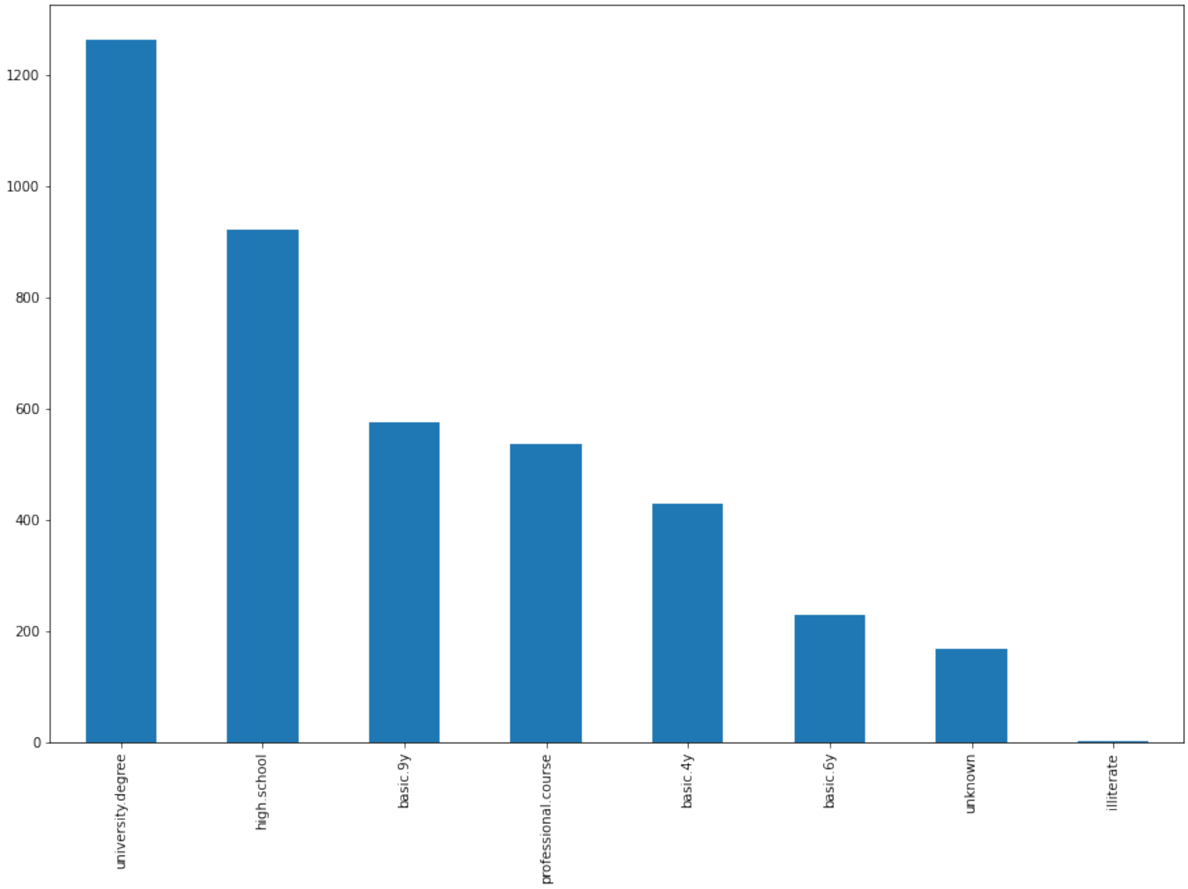


In [27]:
```
#check loan status
data["loan"].value_counts()
```

Out[27]:
```
no         3349
yes         665
unknown     105
Name: loan, dtype: int64
```

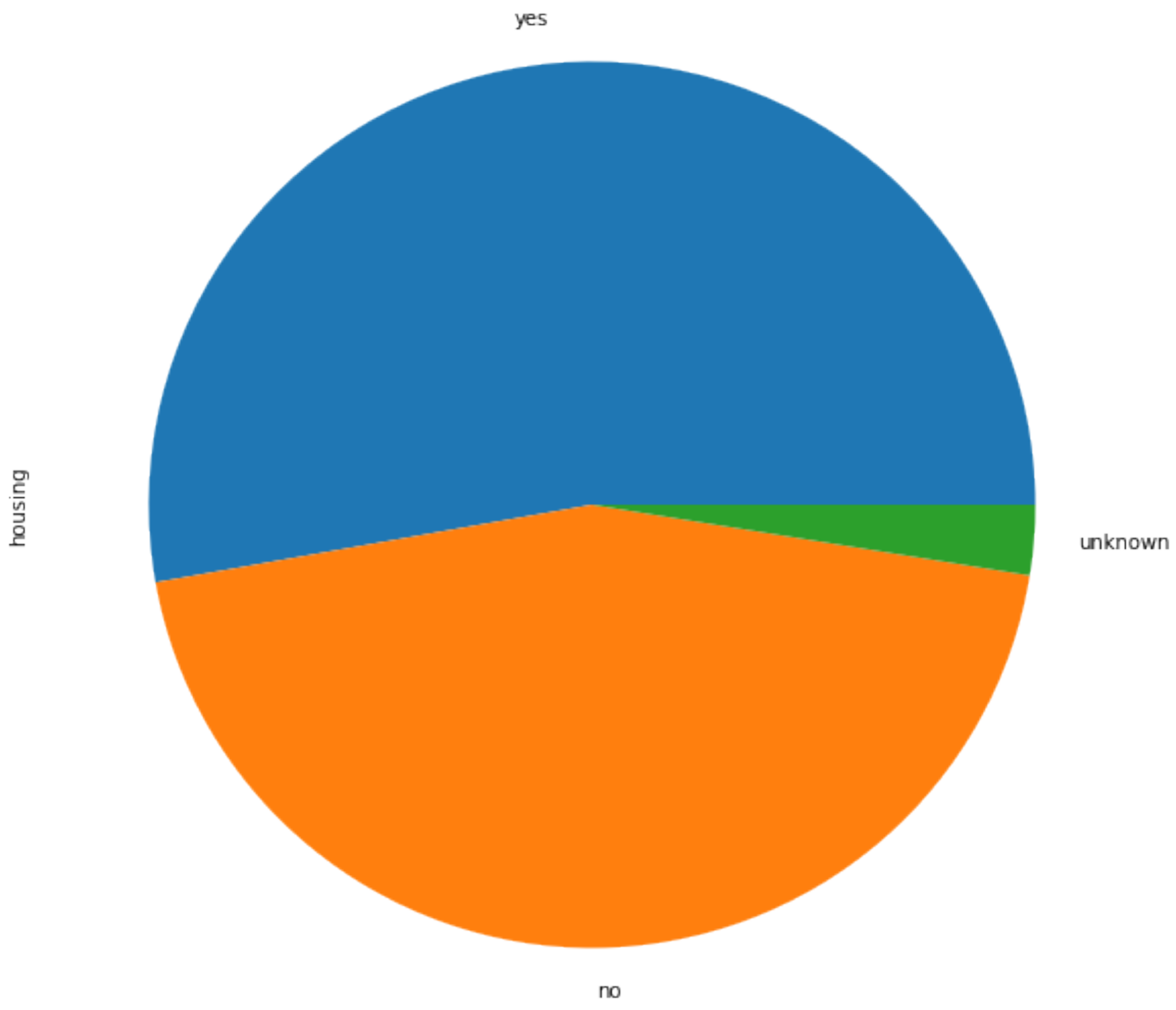
In [28]:
```
#count the value eduction
data["education"].value_counts().plot(kind="bar")
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7c6e290>
```

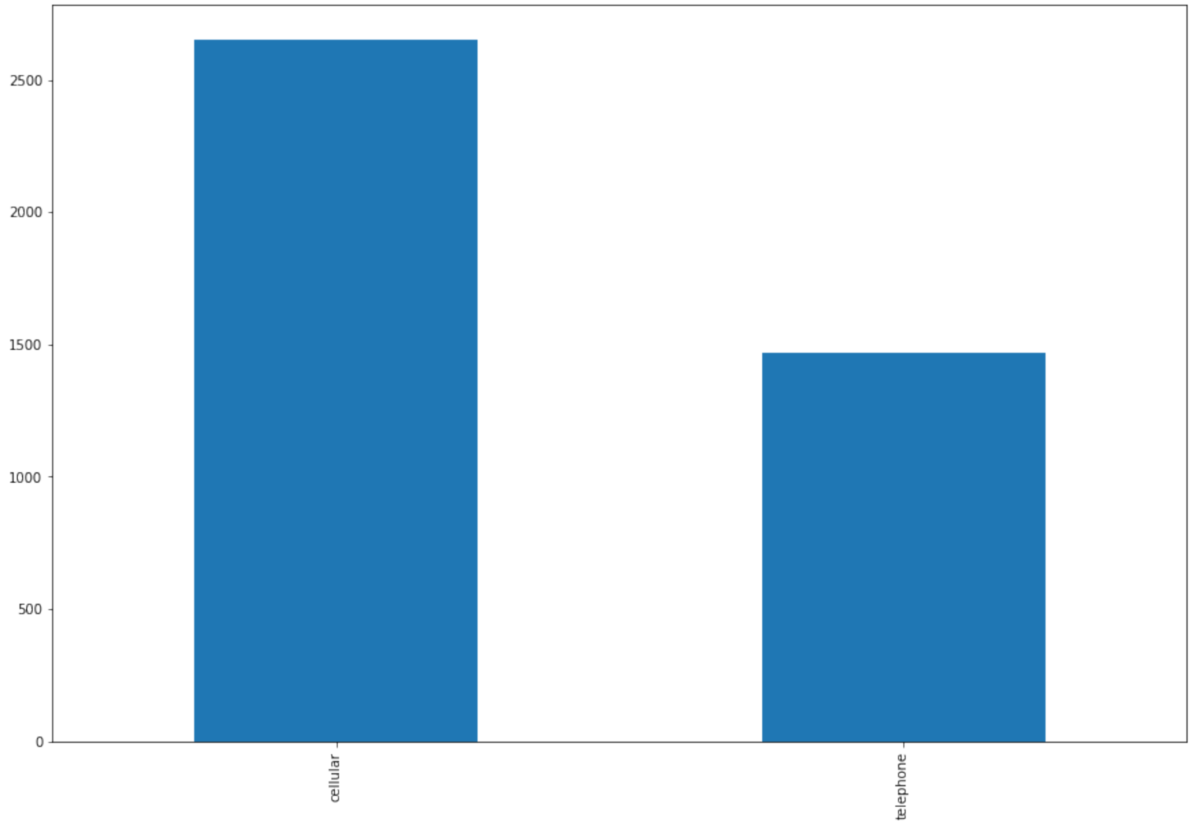


In [29]:
```
data["housing"].value_counts().plot(kind="pie")
```

Out[29]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7a2db90>
```

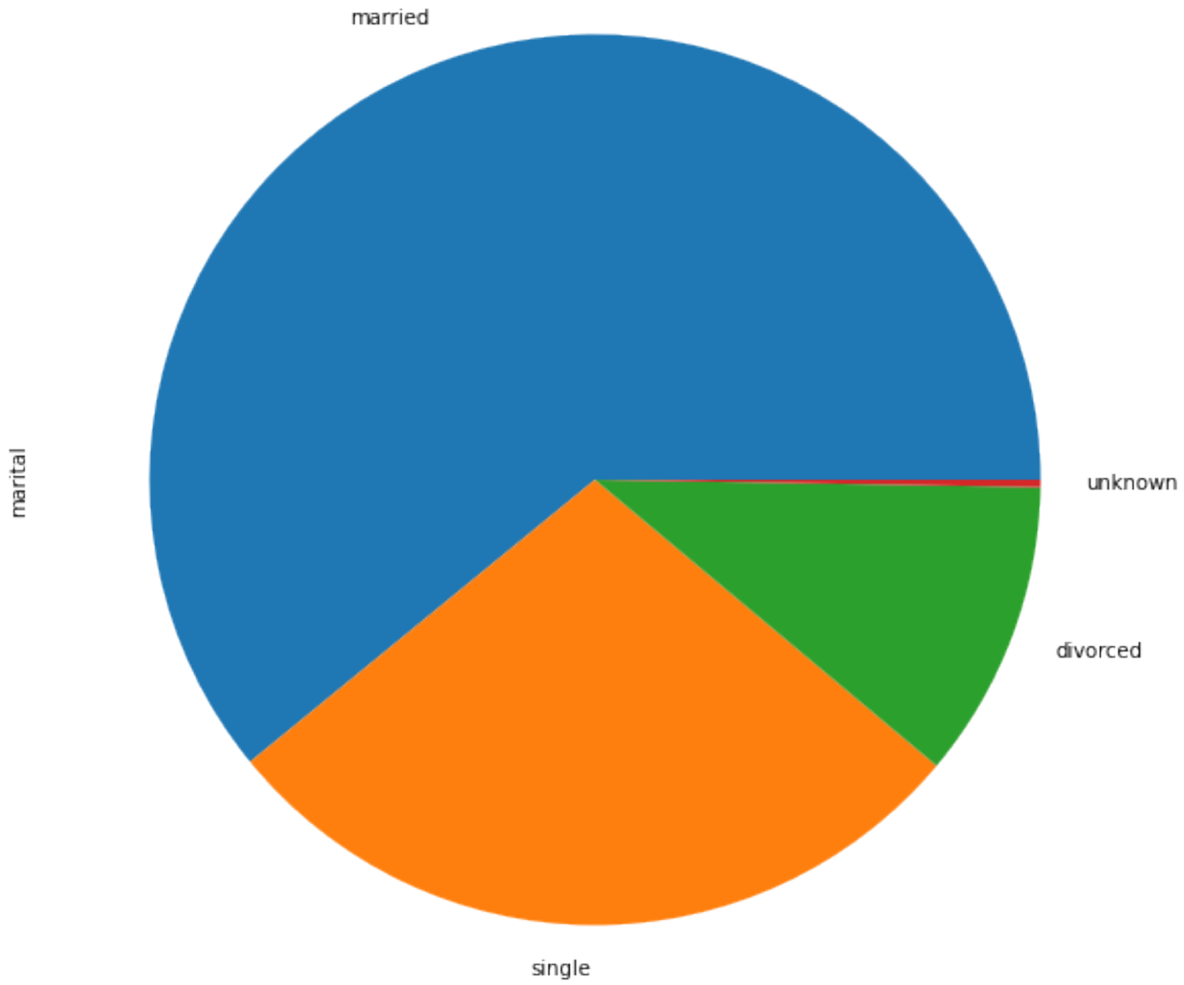


```
In [30]: #contacct value count
         data["contact"].value_counts().plot(kind="bar")

Out[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7efdc7a02710>
```



```
In [31]: data['marital'].value_counts().plot(kind="pie")
```

```
Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x7efdc79fe710>
```

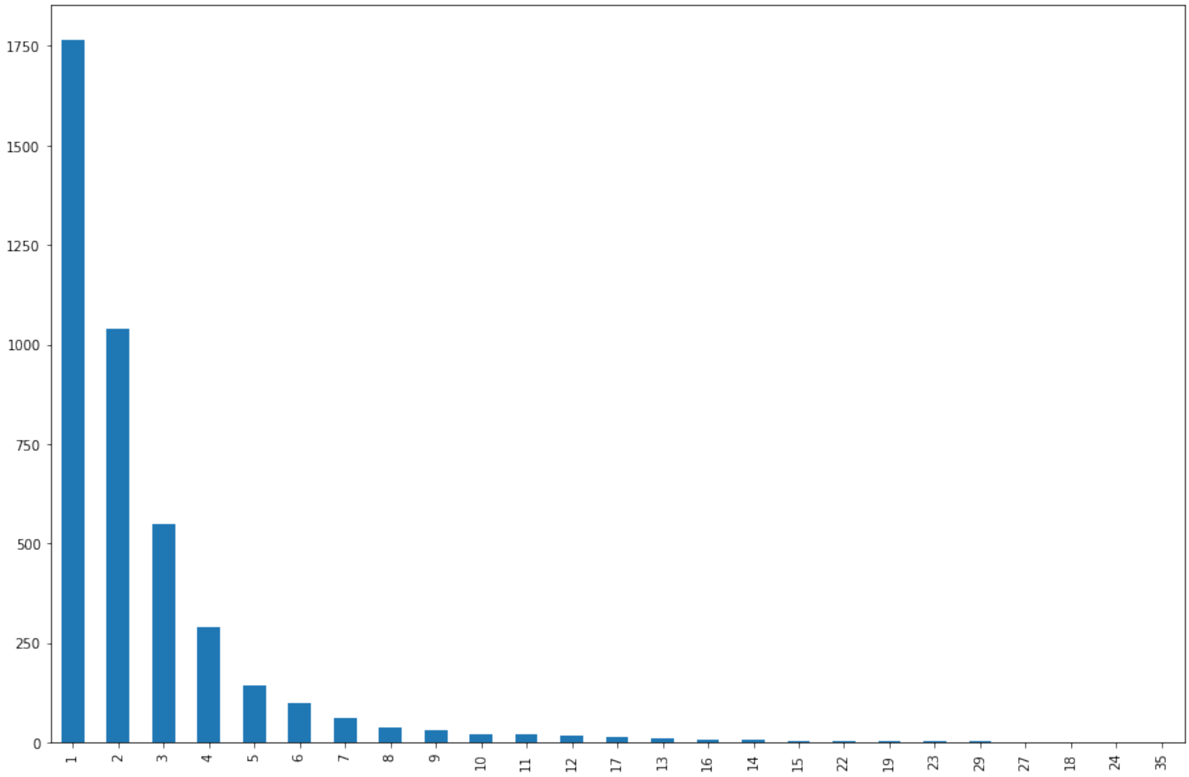


In [32]:
```
data['campaign'].value_counts()
```

Out[32]:
```
1     1764
2     1039
3      549
4      291
5      142
6       99
7       60
8       36
9       32
10      20
11      19
12      16
17      14
13      11
16       7
14       6
15       2
22       2
19       2
23       2
29       2
27       1
18       1
24       1
35       1
Name: campaign, dtype: int64
```

```
In [33]: data['campaign'].value_counts().plot(kind="bar")
```

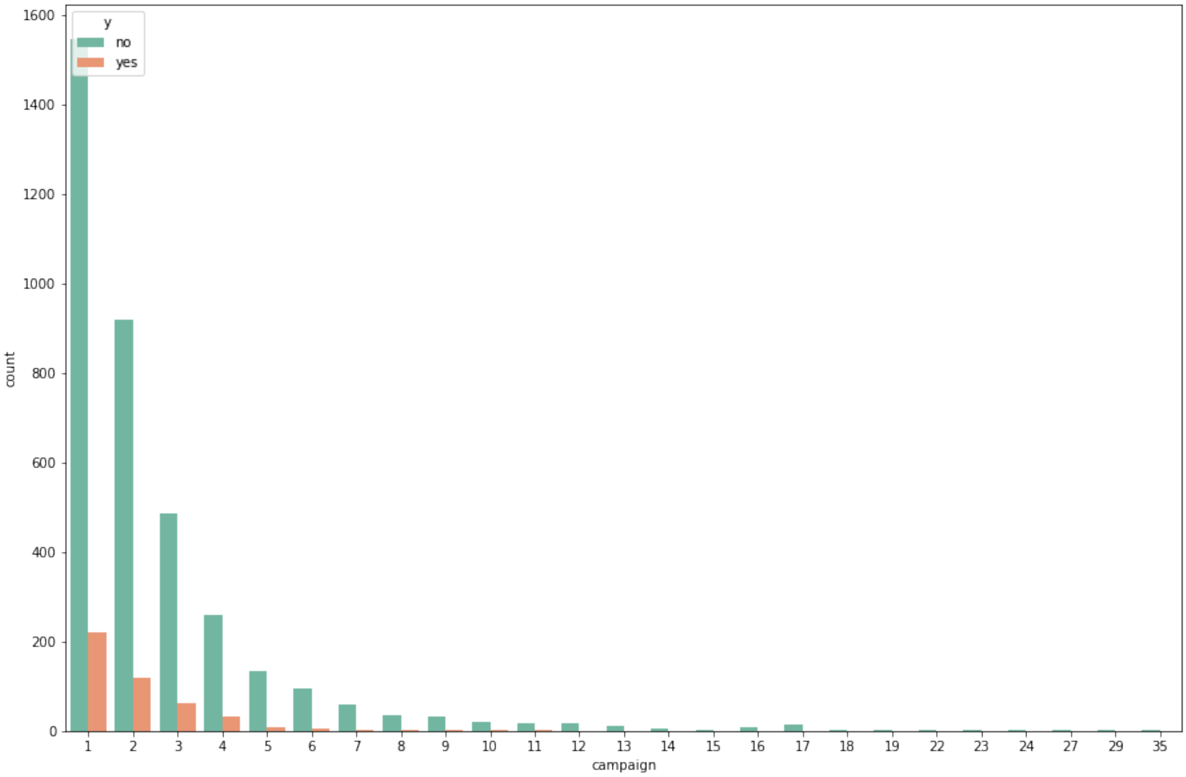
```
Out[33]: <matplotlib.axes._subplots.AxesSubplot at 0x7efdc7949390>
```



In [34]:
```
rcParams['figure.figsize'] = 15,10
sns.countplot(data['campaign'],hue=data['y'],palette="Set2")
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

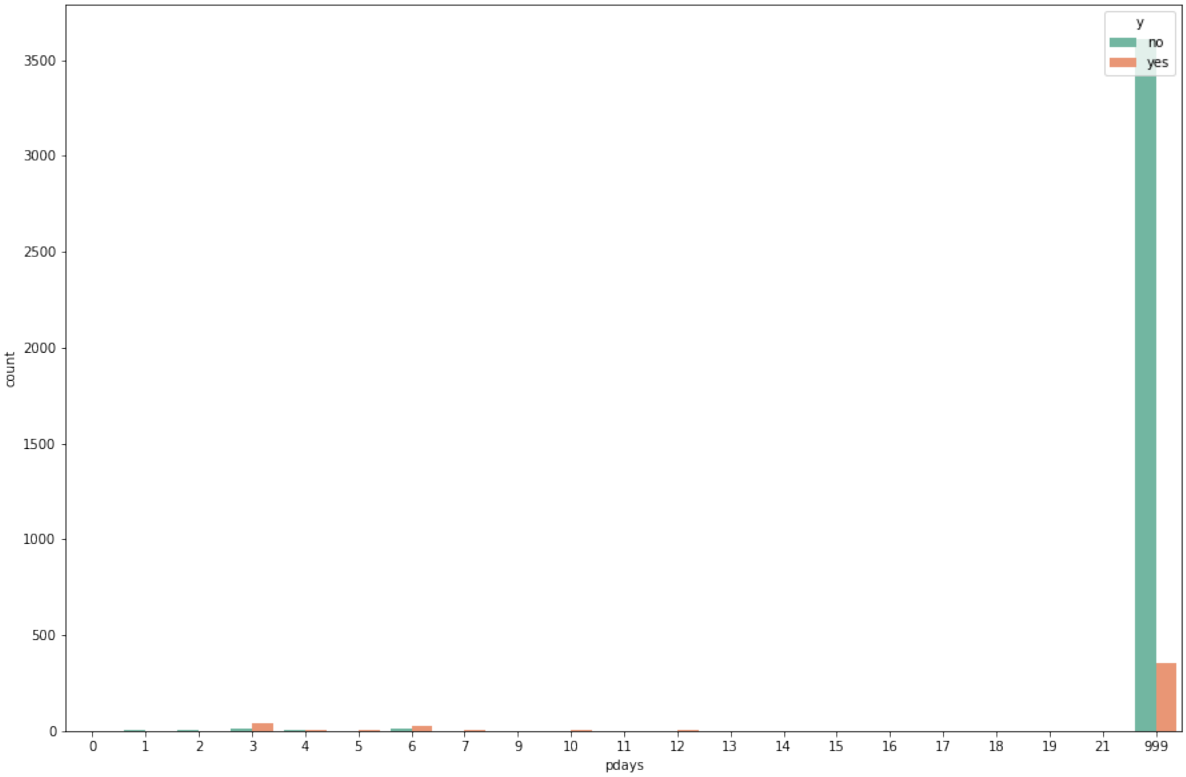
Out[34]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7851450>
```



In [35]:
```
rcParams['figure.figsize'] = 15,10
sns.countplot(data['pdays'],hue=data['y'],palette="Set2")
```

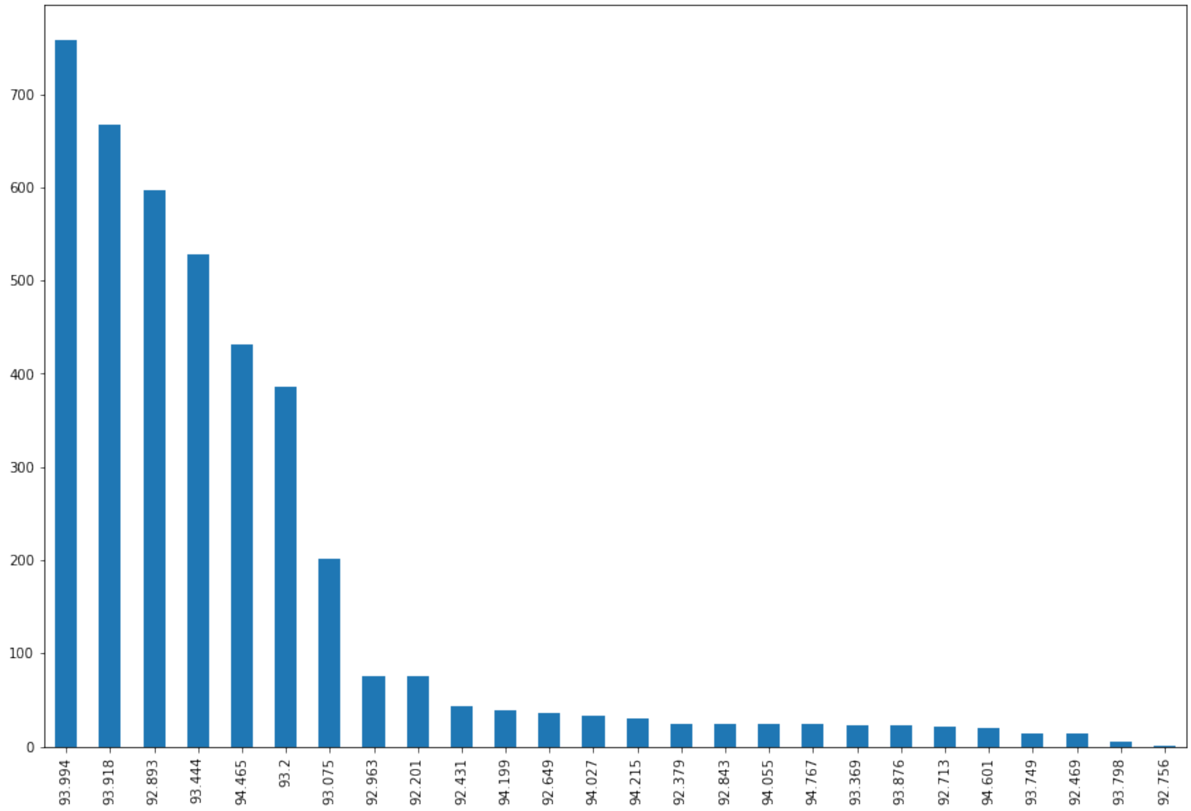
```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43:
FutureWarning: Pass the following variable as a keyword arg: x. Fr
om version 0.12, the only valid positional argument will be `data`
, and passing other arguments without an explicit keyword will res
ult in an error or misinterpretation.
  FutureWarning
```

Out[35]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc782a550>
```



```
In [36]: data['cons.price.idx'].value_counts().plot(kind="bar")
```

```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7efdc76098d0>
```



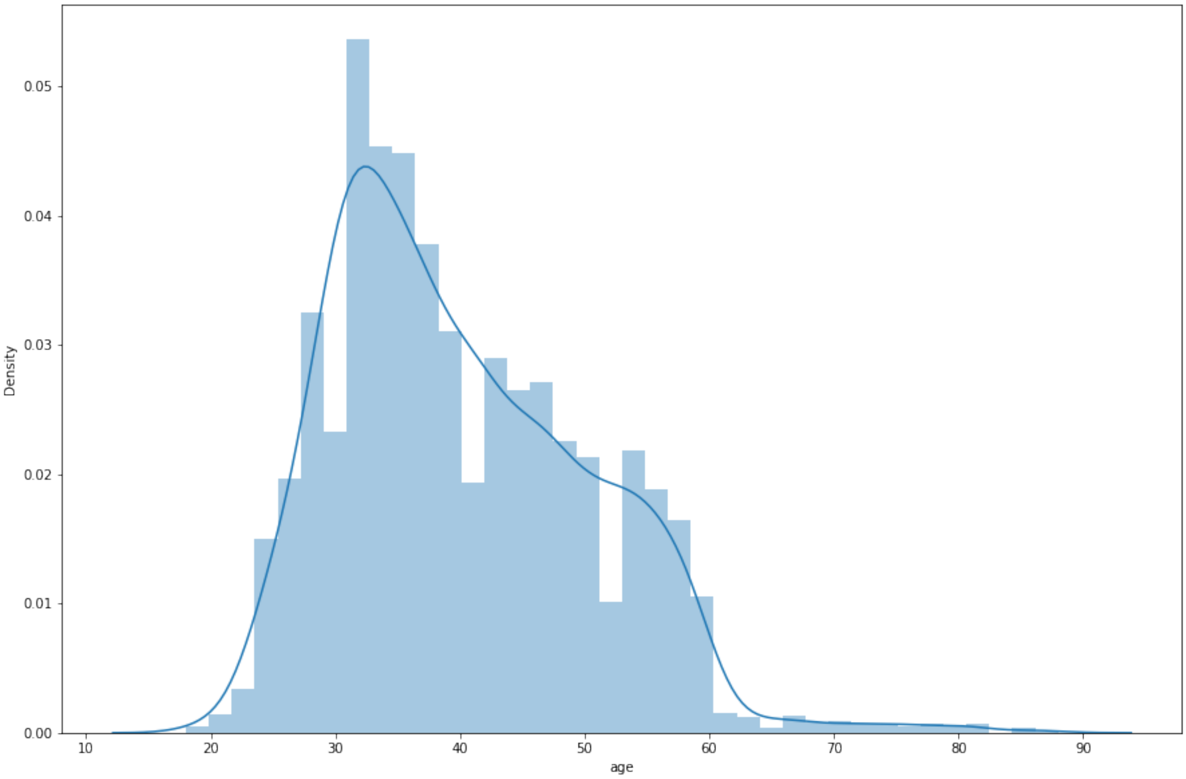
In [38]:

```
#check dataset distubution
sns.distplot(data['age'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

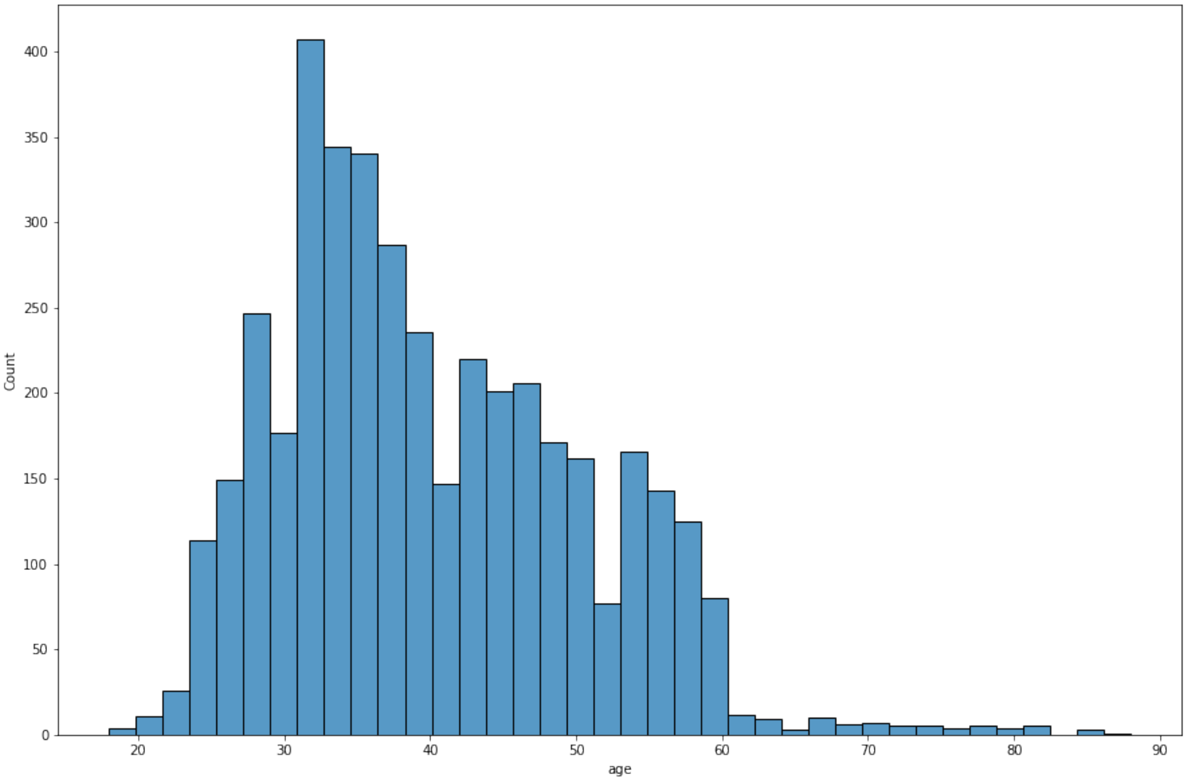
Out[38]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc74c6c90>
```



In [39]:
```
#check histplot for age
sns.histplot(data['age'])
```

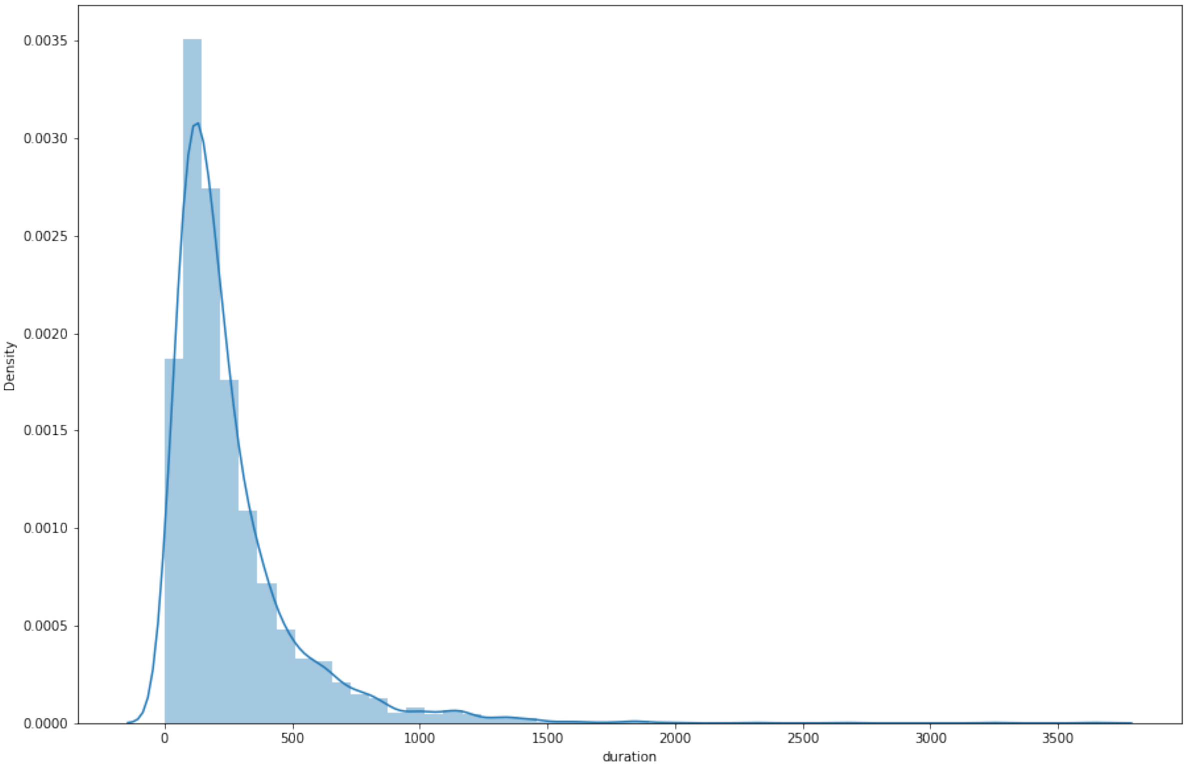
Out[39]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc7437750>
```



In [40]:
```
#check dataset distubution
sns.distplot(data['duration'])
```

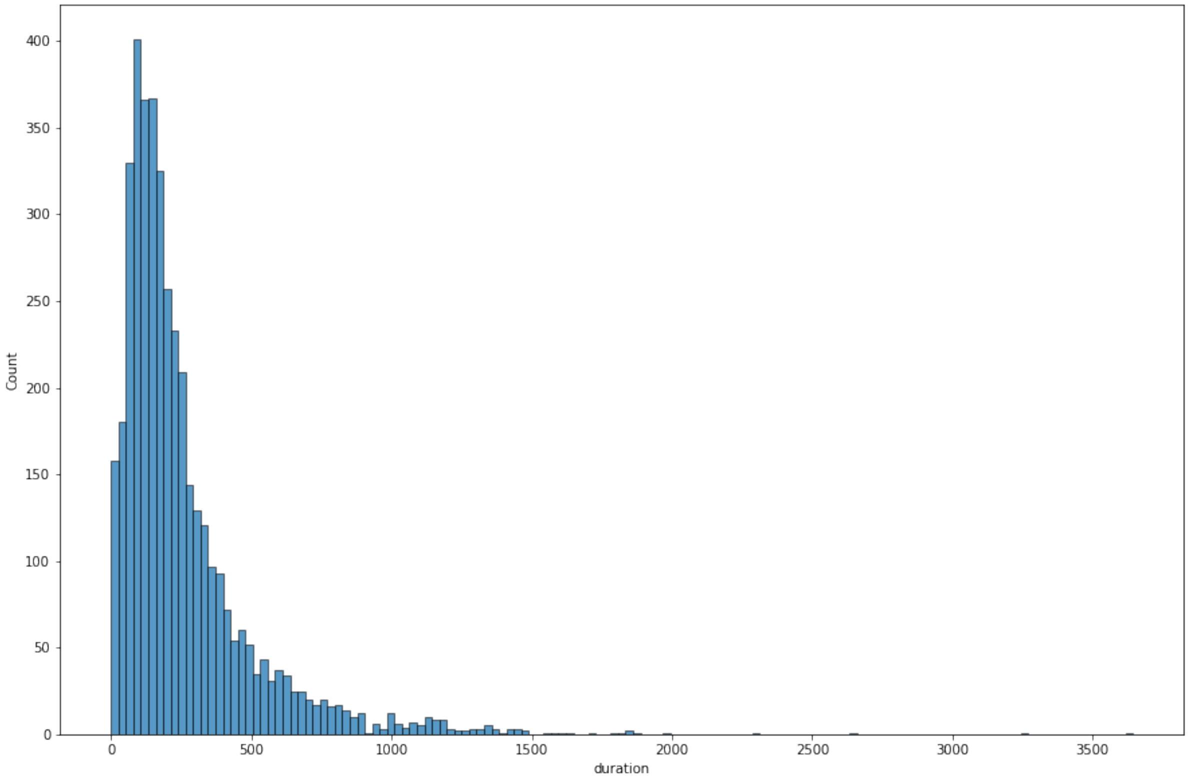
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[40]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc78400d0>
```



In [41]:
```
#check histplot for duration
sns.histplot(data['duration'])
```

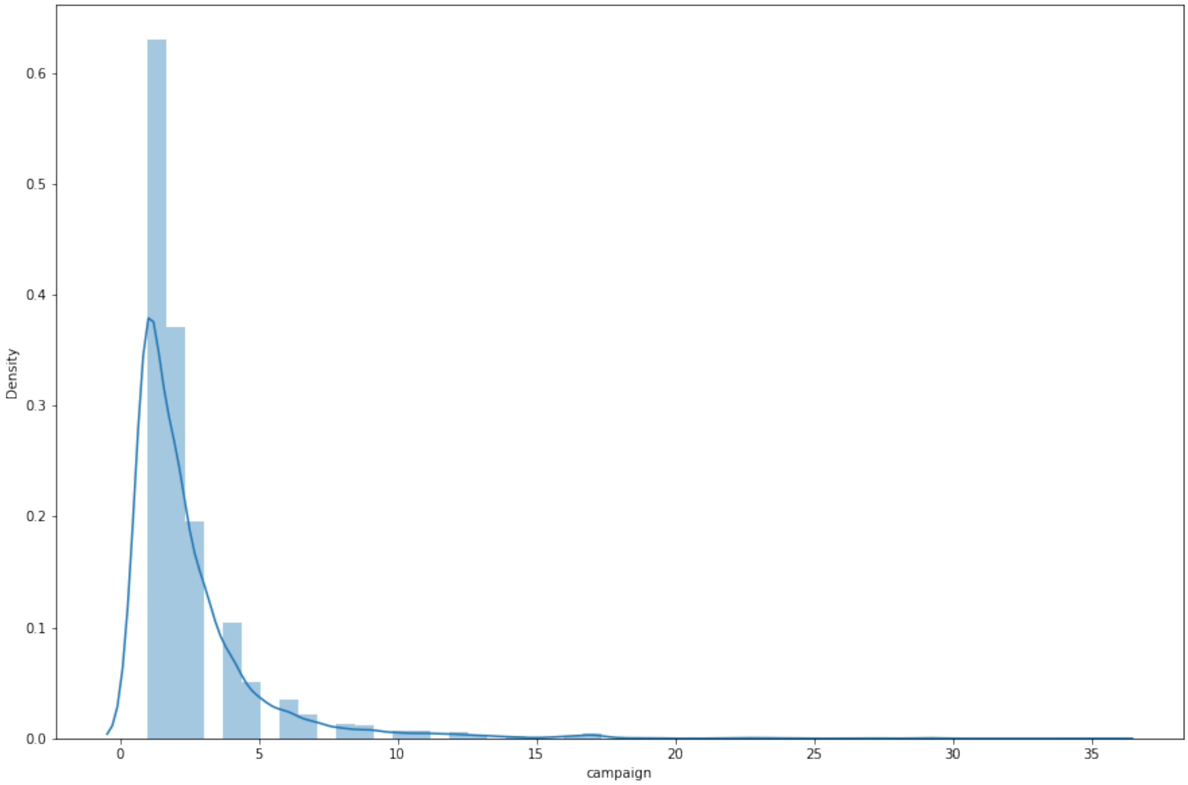
Out[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc720ce90>
```



In [42]:
```
#check dataset distubution
sns.distplot(data['campaign'])
```

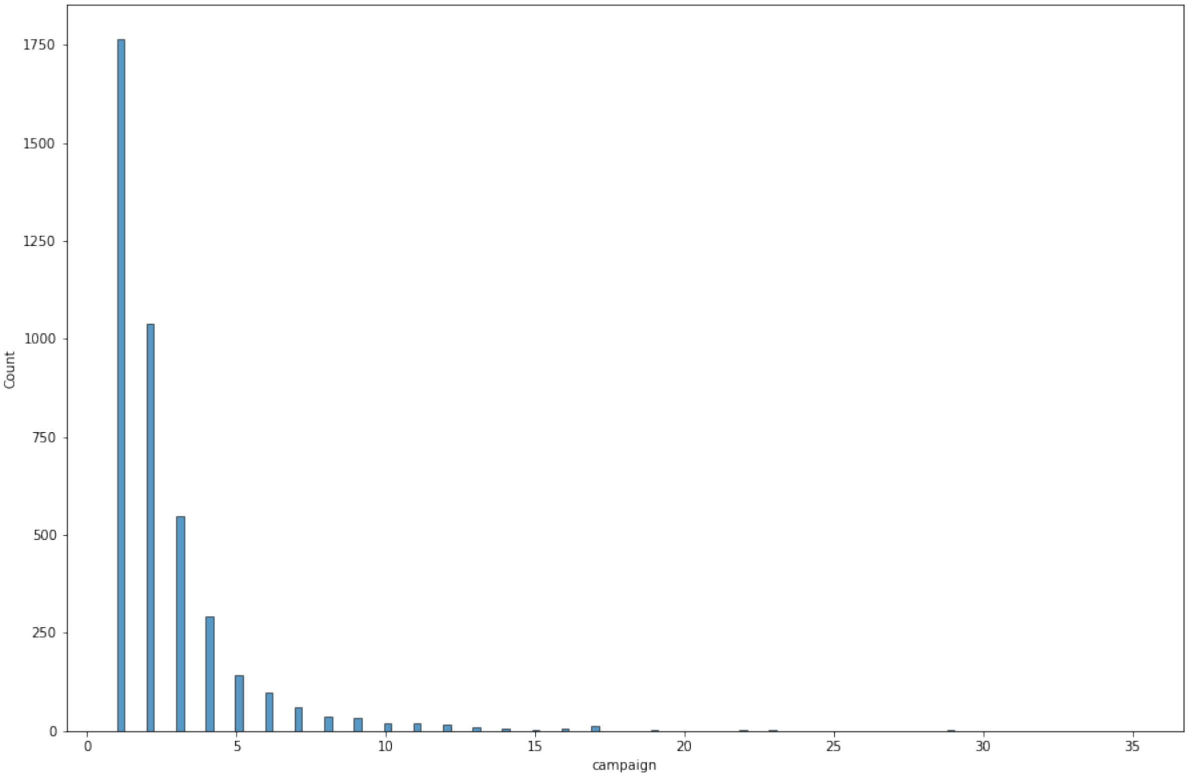
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[42]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc6fb56d0>
```



```
In [43]: #check histplot for campaign
         sns.histplot(data['campaign'])
```

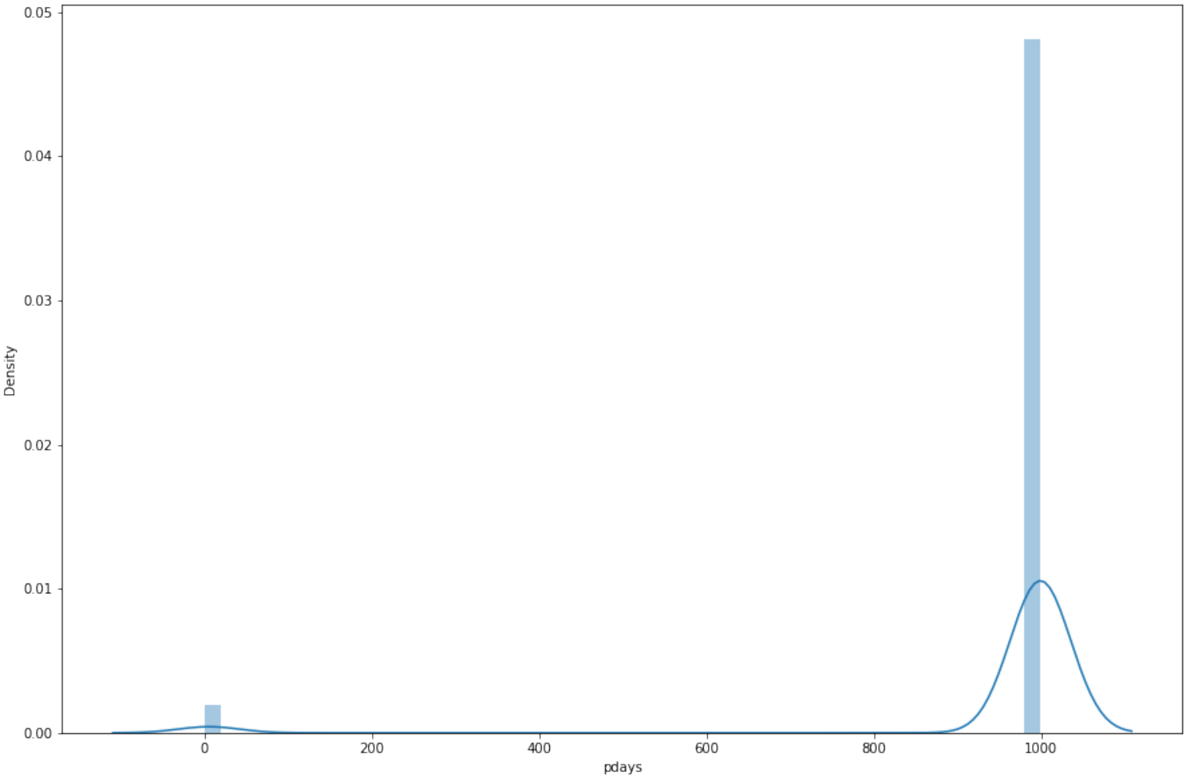
```
Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7efdc6eb7d90>
```



In [44]:
```
#check dataset distubution
sns.distplot(data['pdays'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[44]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc6efa050>
```

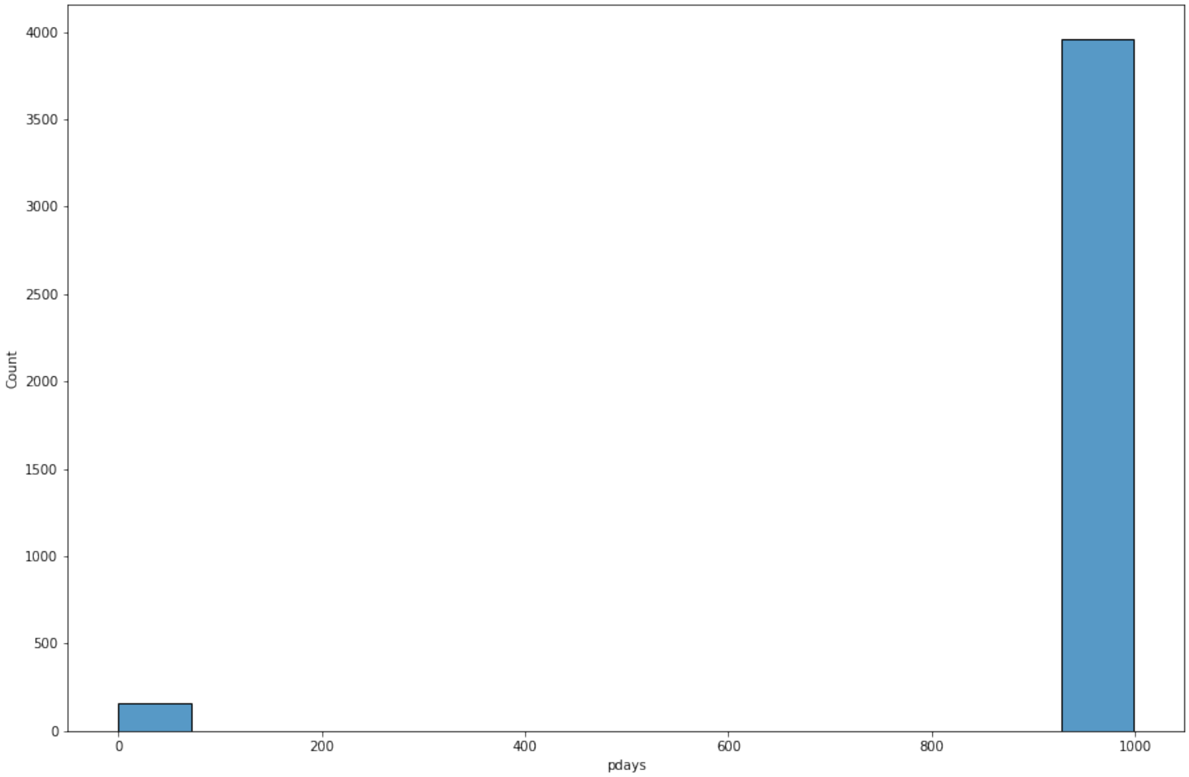


In [45]:

```
#check histplot for pdays
sns.histplot(data['pdays'])
```

Out[45]:

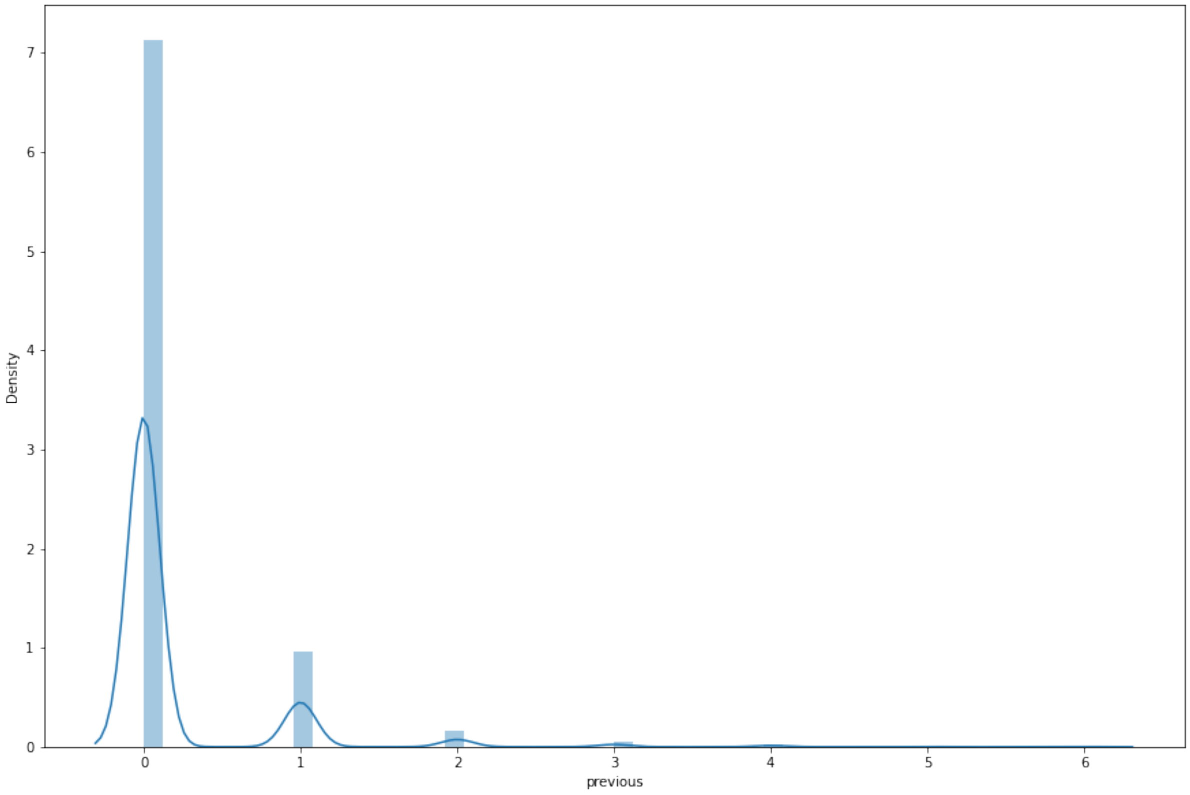
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc6b531d0>
```



In [46]:
```
#check dataset distubution
sns.distplot(data['previous'])
```

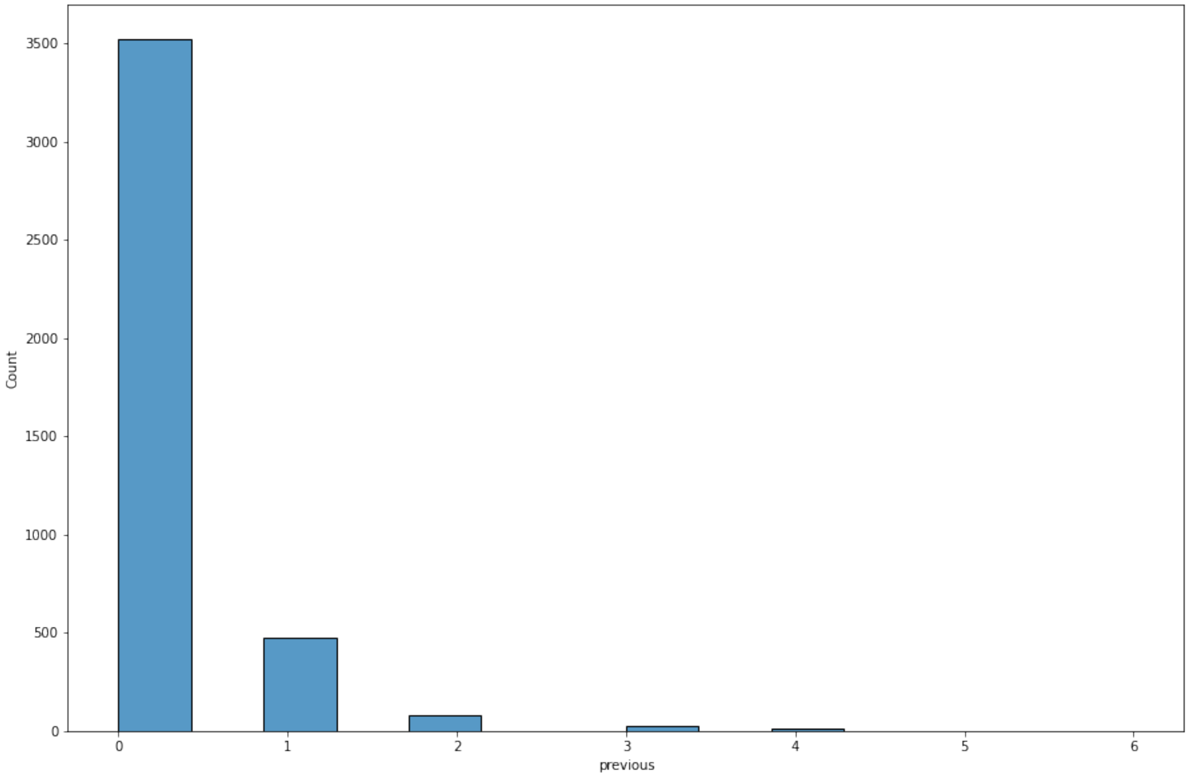
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[46]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc6add910>
```



In [47]:
```
#check histplot for previous
sns.histplot(data['previous'])
```

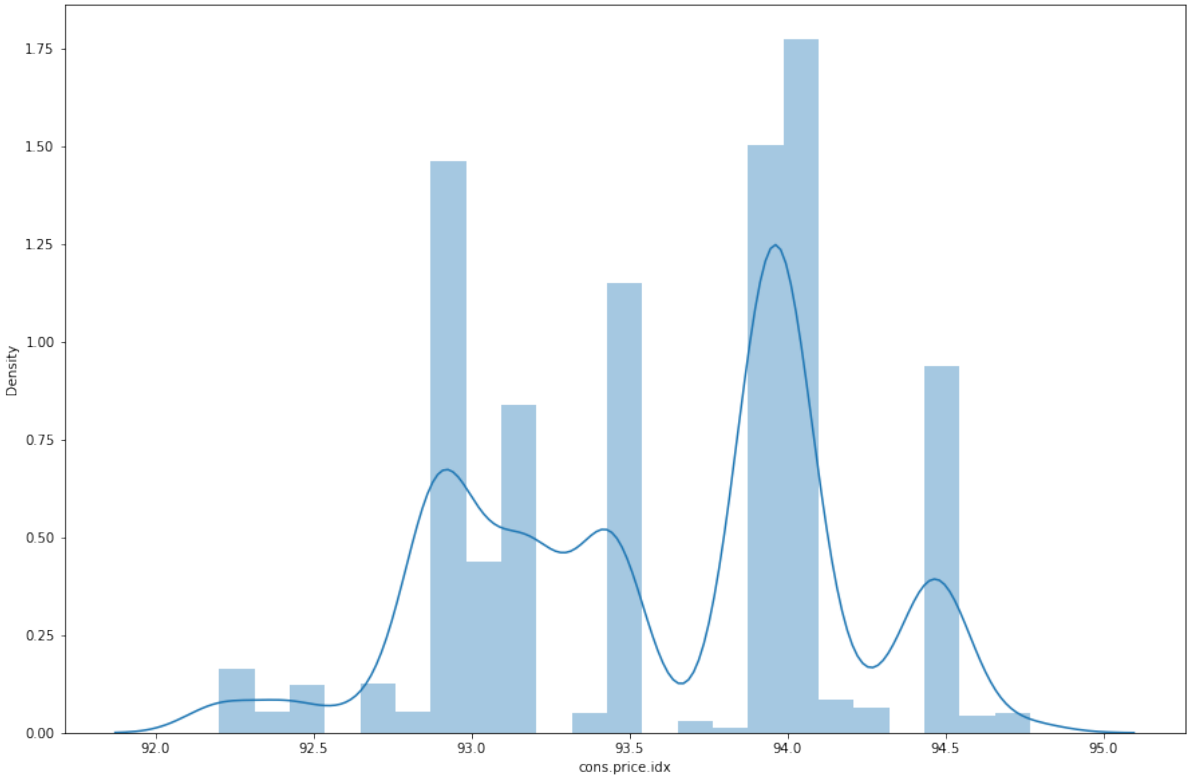
Out[47]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc69e0390>
```



In [48]:
```
#check dataset distubution
sns.distplot(data['cons.price.idx'])
```

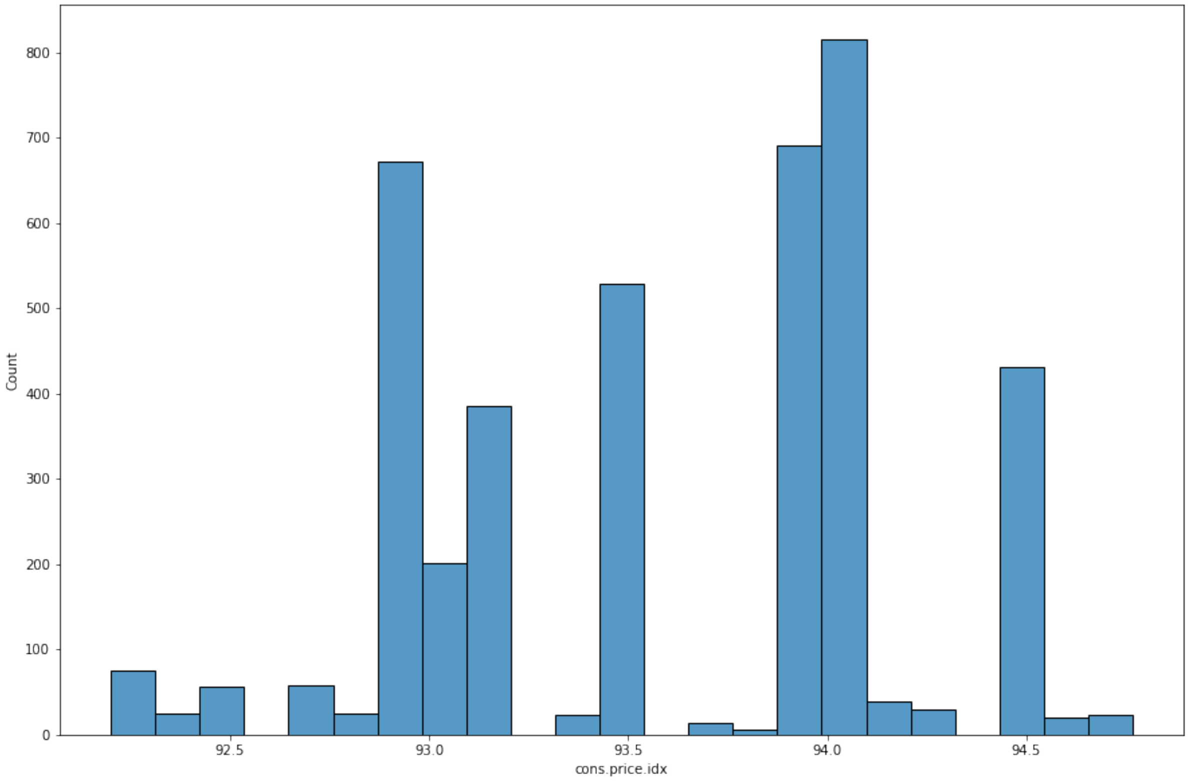
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[48]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc697c0d0>
```



In [49]:
```
#check histplot for cons.price.idx
sns.histplot(data['cons.price.idx'])
```

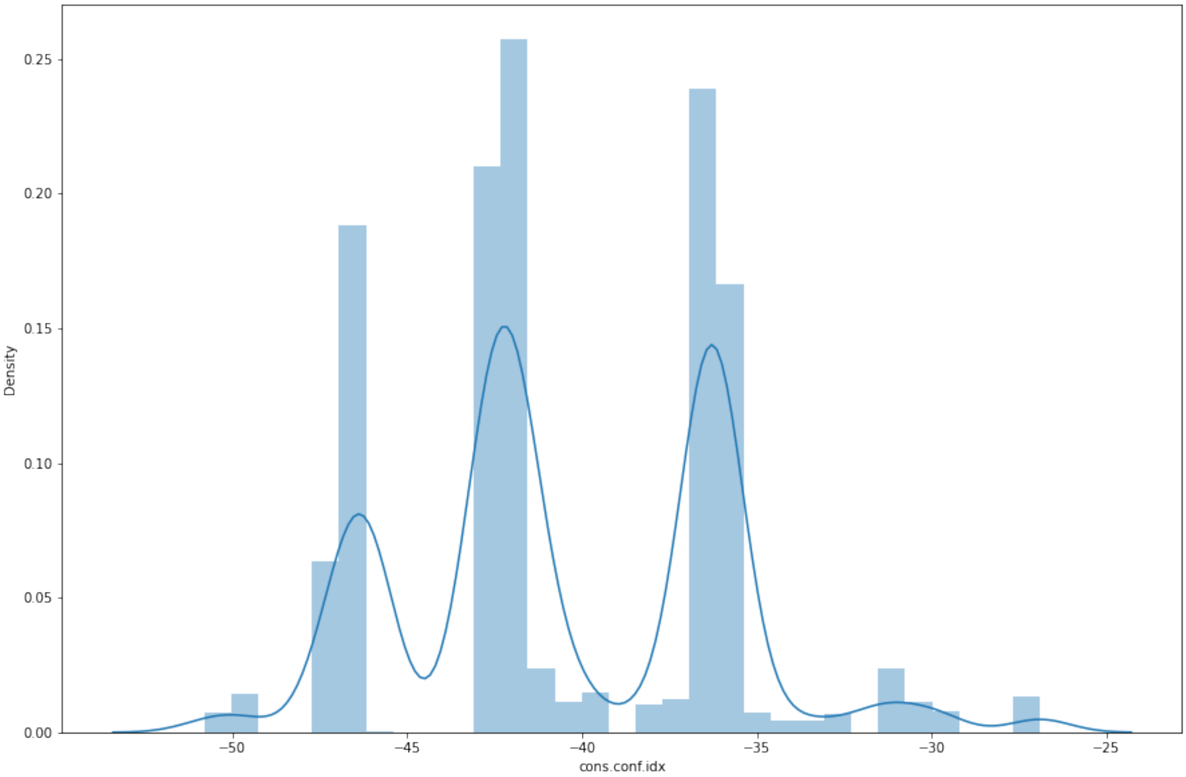
Out[49]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc724a0d0>
```



In [50]:
```
#check dataset distubution
sns.distplot(data['cons.conf.idx'])
```

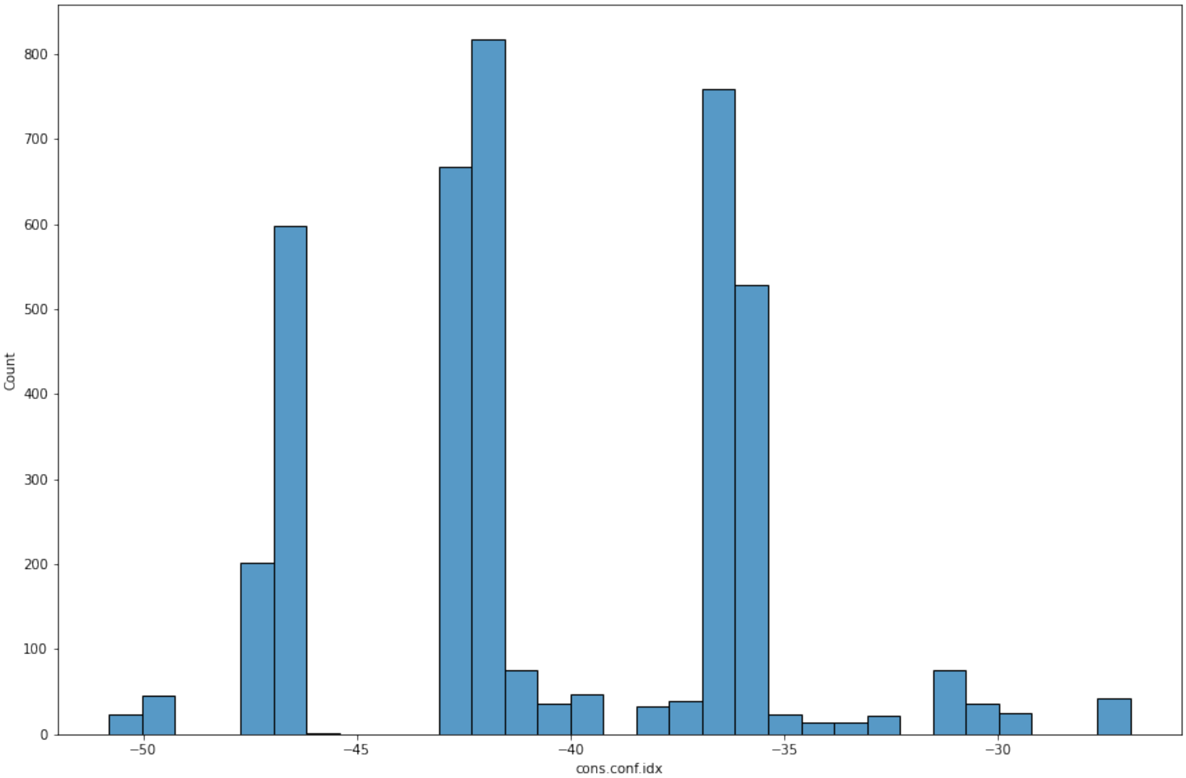
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[50]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc682a350>
```



In [51]:
```
#check histplot for cons.conf.idx
sns.histplot(data['cons.conf.idx'])
```

Out[51]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc67b10d0>
```



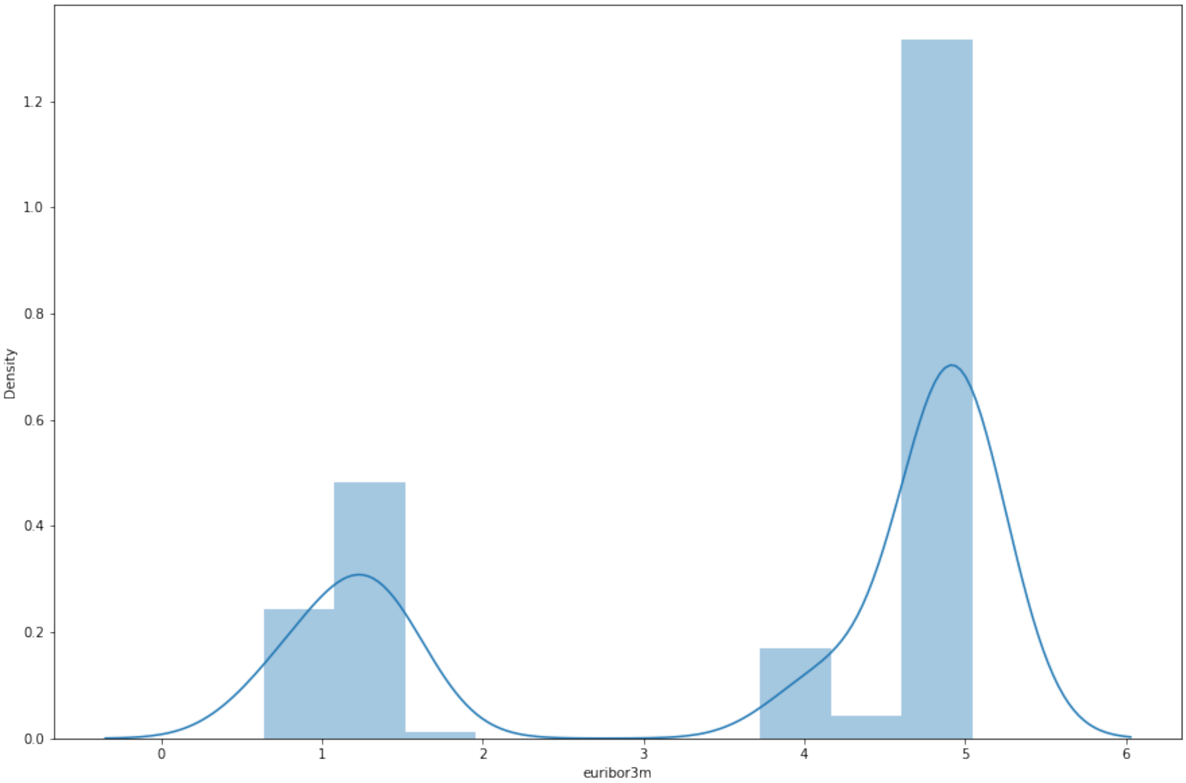
In [52]:

```
#check dataset distubution
sns.distplot(data['euribor3m'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[52]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc66becd0>
```

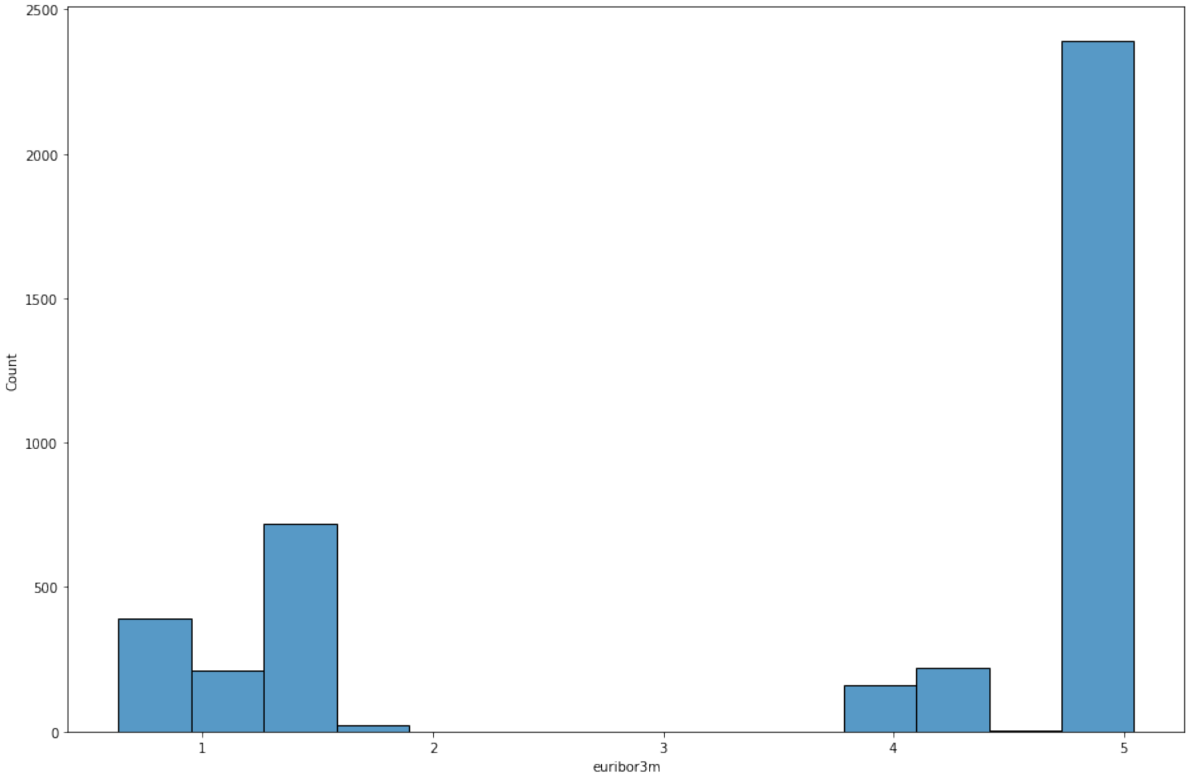


In [53]:

```
#check histplot for euribor3m
sns.histplot(data['euribor3m'])
```

Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc663df90>
```



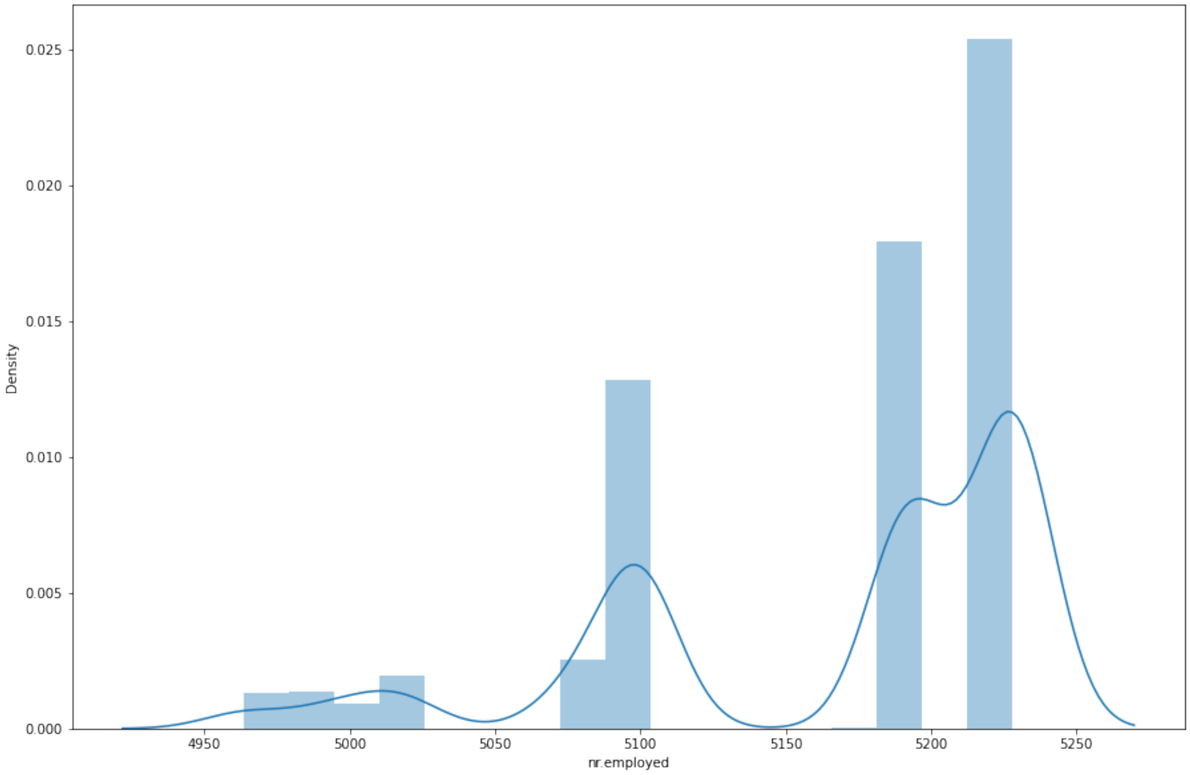
In [54]:

```
#check dataset distubution
sns.distplot(data['nr.employed'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:26
19: FutureWarning: `distplot` is a deprecated function and will be
removed in a future version. Please adapt your code to use either
`displot` (a figure-level function with similar flexibility) or `h
istplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

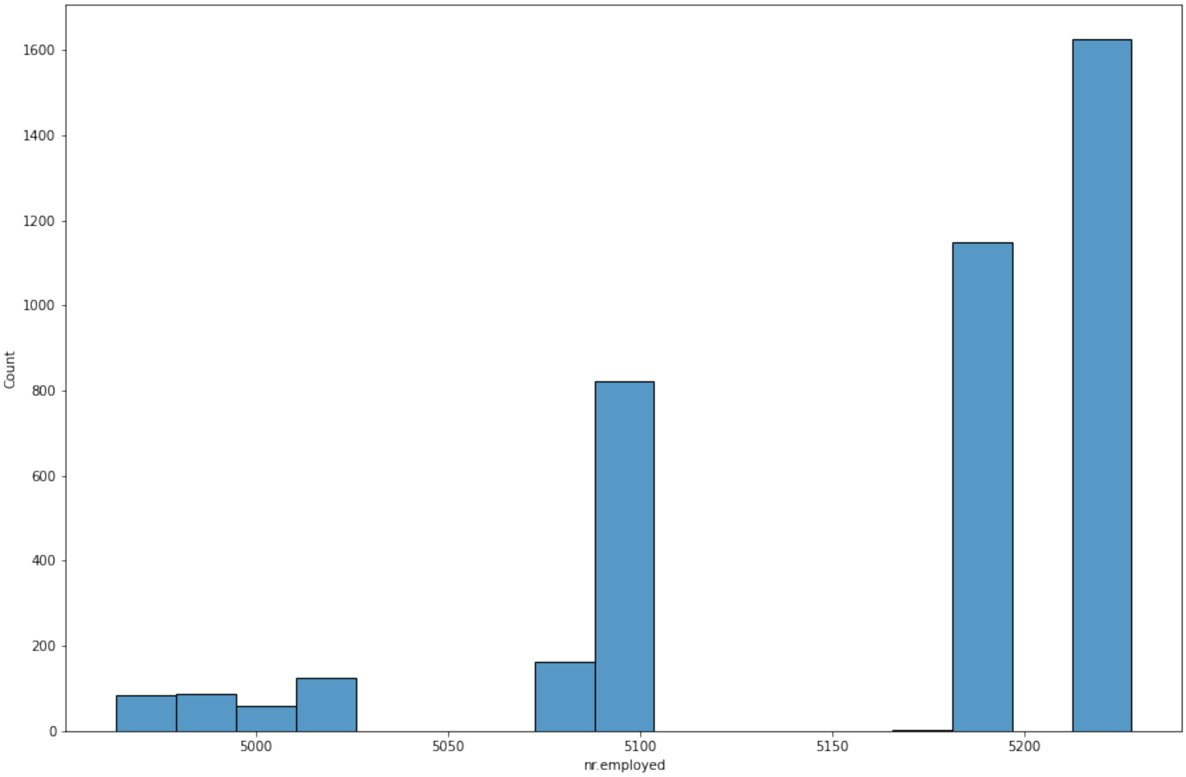
Out[54]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc655d510>
```



In [55]:
```
#check histplot for nr.employed
sns.histplot(data['nr.employed'])
```

Out[55]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc6503e50>
```

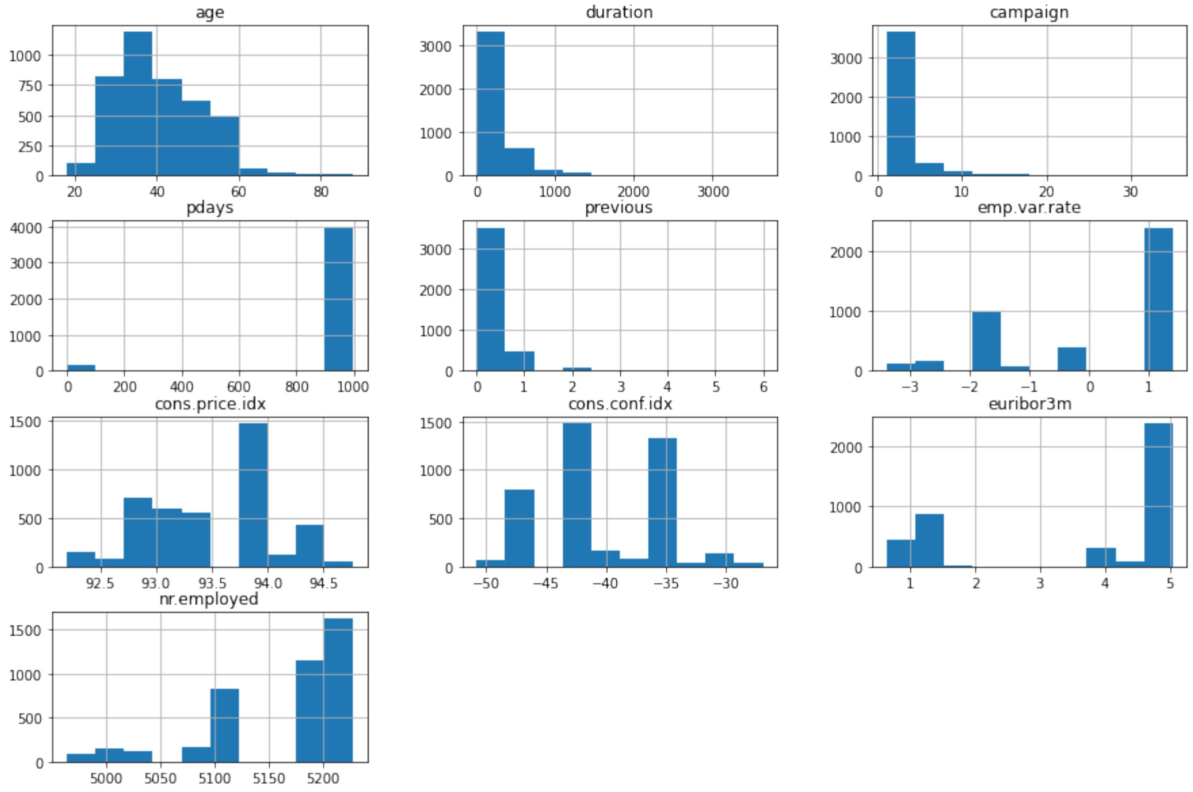


In [56]:

```
data.hist()
```

Out[56]:

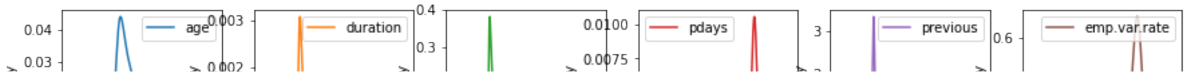
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc64
2e150>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc64
1db90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc64
0c490>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc63
c1990>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc63
78e90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc63
3c3d0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc62
f19d0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc62
a6e10>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc62
a6e50>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc62
69490>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc61
d7d90>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc61
9b2d0>]],
      dtype=object)
```

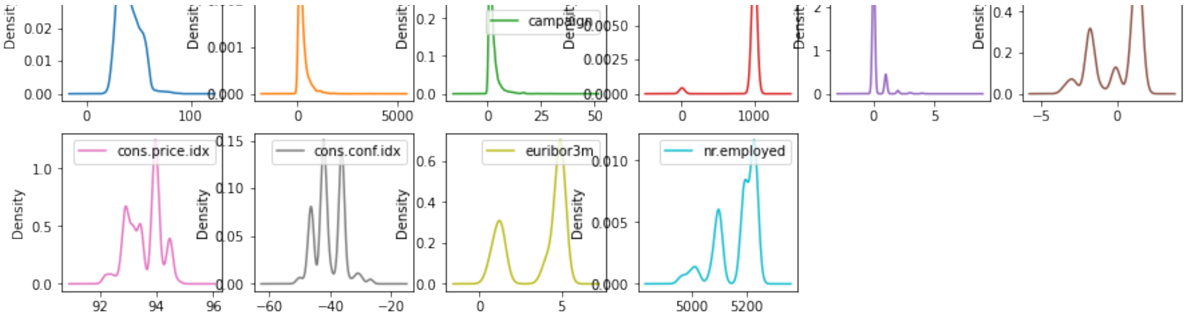


In [57]:

```
data.plot(kind='density', subplots=True, layout=(4,6), sharex=False
```

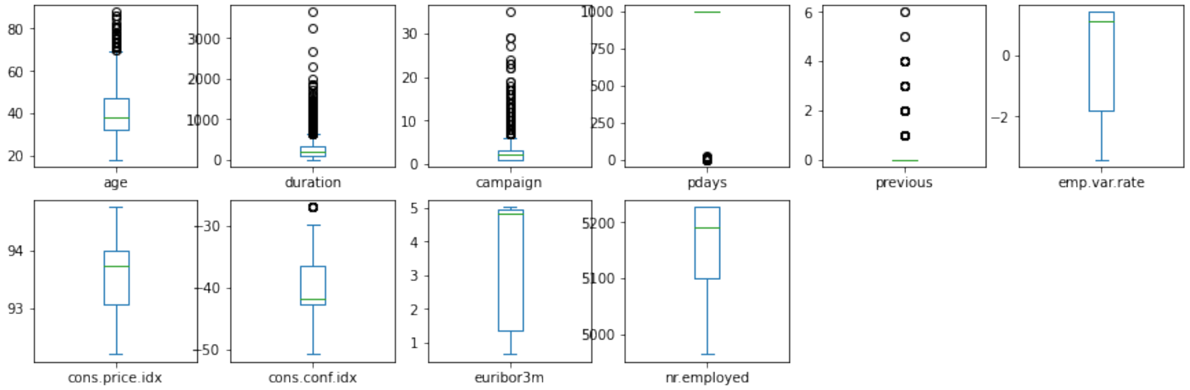
```
Out[57]: array([[<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc87
47050>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc71
ff6d0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc67
4fc50>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc65
56310>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc68
171d0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc60
8e0d0>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5d
f9750>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5d
b0c90>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5d
b0cd0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5d
73410>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5c
e2f10>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5c
a5550>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5c
5ab50>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5c
1f190>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5b
d8790>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5c
0cd90>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5b
523d0>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5b
889d0>],
                [<matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5b
2ba90>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5b
03610>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5a
bac10>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5a
7f250>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc5a
35850>,
                 <matplotlib.axes._subplots.AxesSubplot object at 0x7efdc59
eae50>]],
      dtype=object)
```





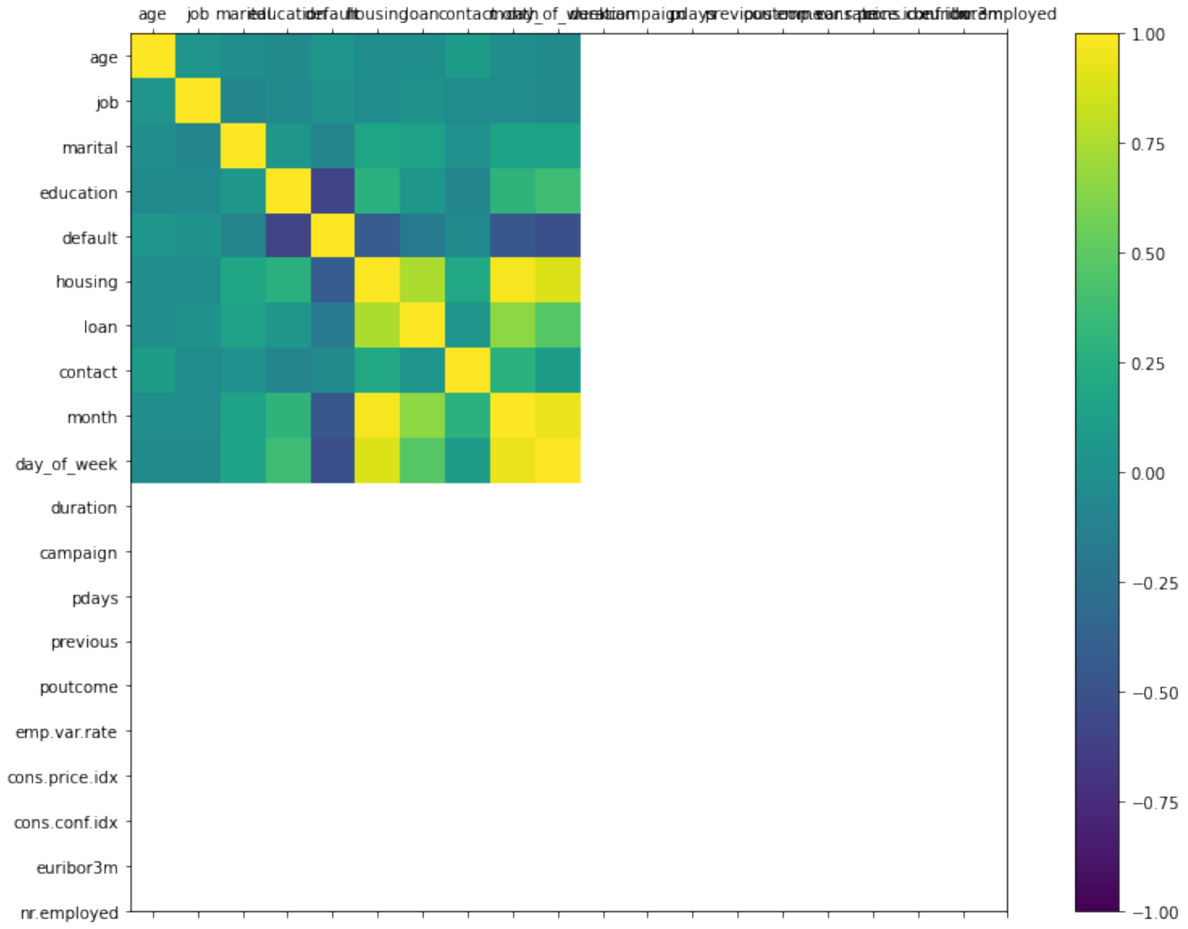
In [58]:
```
data.plot(kind='box', subplots=True, layout=(4,6), sharex=False, sh
```

Out[58]:
```
age                   AxesSubplot(0.125,0.71587;0.110714x0.16413)
duration           AxesSubplot(0.257857,0.71587;0.110714x0.16413)
campaign           AxesSubplot(0.390714,0.71587;0.110714x0.16413)
pdays              AxesSubplot(0.523571,0.71587;0.110714x0.16413)
previous           AxesSubplot(0.656429,0.71587;0.110714x0.16413)
emp.var.rate       AxesSubplot(0.789286,0.71587;0.110714x0.16413)
cons.price.idx       AxesSubplot(0.125,0.518913;0.110714x0.16413)
cons.conf.idx     AxesSubplot(0.257857,0.518913;0.110714x0.16413)
euribor3m         AxesSubplot(0.390714,0.518913;0.110714x0.16413)
nr.employed       AxesSubplot(0.523571,0.518913;0.110714x0.16413)
dtype: object
```



In [59]:
```
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(data.corr(), vmin=-1, vmax=1)
fig.colorbar(cax)
ticks = np.arange(0,20,1)
ax.set_xticks(ticks)
ax.set_yticks(ticks)
ax.set_xticklabels(data.columns)
ax.set_yticklabels(data.columns)
```

Out[59]:
```
[Text(0, 0, 'age'),
 Text(0, 0, 'job'),
 Text(0, 0, 'marital'),
 Text(0, 0, 'education'),
 Text(0, 0, 'default'),
 Text(0, 0, 'housing'),
 Text(0, 0, 'loan'),
 Text(0, 0, 'contact'),
 Text(0, 0, 'month'),
 Text(0, 0, 'day_of_week'),
 Text(0, 0, 'duration'),
 Text(0, 0, 'campaign'),
 Text(0, 0, 'pdays'),
 Text(0, 0, 'previous'),
 Text(0, 0, 'poutcome'),
 Text(0, 0, 'emp.var.rate'),
 Text(0, 0, 'cons.price.idx'),
 Text(0, 0, 'cons.conf.idx'),
 Text(0, 0, 'euribor3m'),
 Text(0, 0, 'nr.employed')]
```



```
In [60]: #copy data
         new_df = data.copy(deep=True)
```

In [63]:

```
le = preprocessing.LabelEncoder()

# job
le.fit(new_df['job'])
new_df['job'] = le.transform(new_df['job'])

# maritial feature
le.fit(new_df['marital'])
new_df['marital'] = le.transform(new_df['marital'])

# education_feature
le.fit(new_df['education'])
new_df['education'] = le.transform(new_df['education'])

# housing_feature
le.fit(new_df['housing'])
new_df['housing'] = le.transform(new_df['housing'])

# loan_feature
le.fit(new_df['loan'])
new_df['loan'] = le.transform(new_df['loan'])

# contact_feature
le.fit(new_df['contact'])
new_df['contact'] = le.transform(new_df['contact'])

# Month_feature
le.fit(new_df['month'])
new_df['month'] = le.transform(new_df['month'])

# day of week_feature
le.fit(new_df['day_of_week'])
new_df['day_of_week'] = le.transform(new_df['day_of_week'])

# poutcome_feature
le.fit(new_df['poutcome'])
new_df['poutcome'] = le.transform(new_df['poutcome'])

# default_feature
le.fit(new_df['default'])
new_df['default'] = le.transform(new_df['default'])



# Target_feature
le.fit(new_df['y'])
new_df['y'] = le.transform(new_df['y'])
```

In [64]:
```
new_df.head()
```

Out[64]:

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...
0	30	1	1	2	0	2	0	0	6	0	...
1	39	7	2	3	0	0	0	1	6	0	...
2	25	7	1	3	0	2	0	1	4	4	...
3	38	7	1	2	0	1	1	1	4	0	...
4	47	0	1	6	0	2	0	0	7	1	...

5 rows × 21 columns

In [66]:
```
#check data type after
new_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4119 entries, 0 to 4118
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   age             4119 non-null   int64  
 1   job             4119 non-null   int64  
 2   marital         4119 non-null   int64  
 3   education       4119 non-null   int64  
 4   default         4119 non-null   int64  
 5   housing         4119 non-null   int64  
 6   loan            4119 non-null   int64  
 7   contact         4119 non-null   int64  
 8   month           4119 non-null   int64  
 9   day_of_week     4119 non-null   int64  
 10  duration        4119 non-null   int64  
 11  campaign        4119 non-null   int64  
 12  pdays           4119 non-null   int64  
 13  previous        4119 non-null   int64  
 14  poutcome        4119 non-null   int64  
 15  emp.var.rate    4119 non-null   float64
 16  cons.price.idx  4119 non-null   float64
 17  cons.conf.idx   4119 non-null   float64
 18  euribor3m       4119 non-null   float64
 19  nr.employed     4119 non-null   float64
 20  y               4119 non-null   int64  
dtypes: float64(5), int64(16)
memory usage: 675.9 KB
```

In [67]:
```
#groupby dataset mean
new_df.groupby(data['y']).mean()
```

Out[67]:

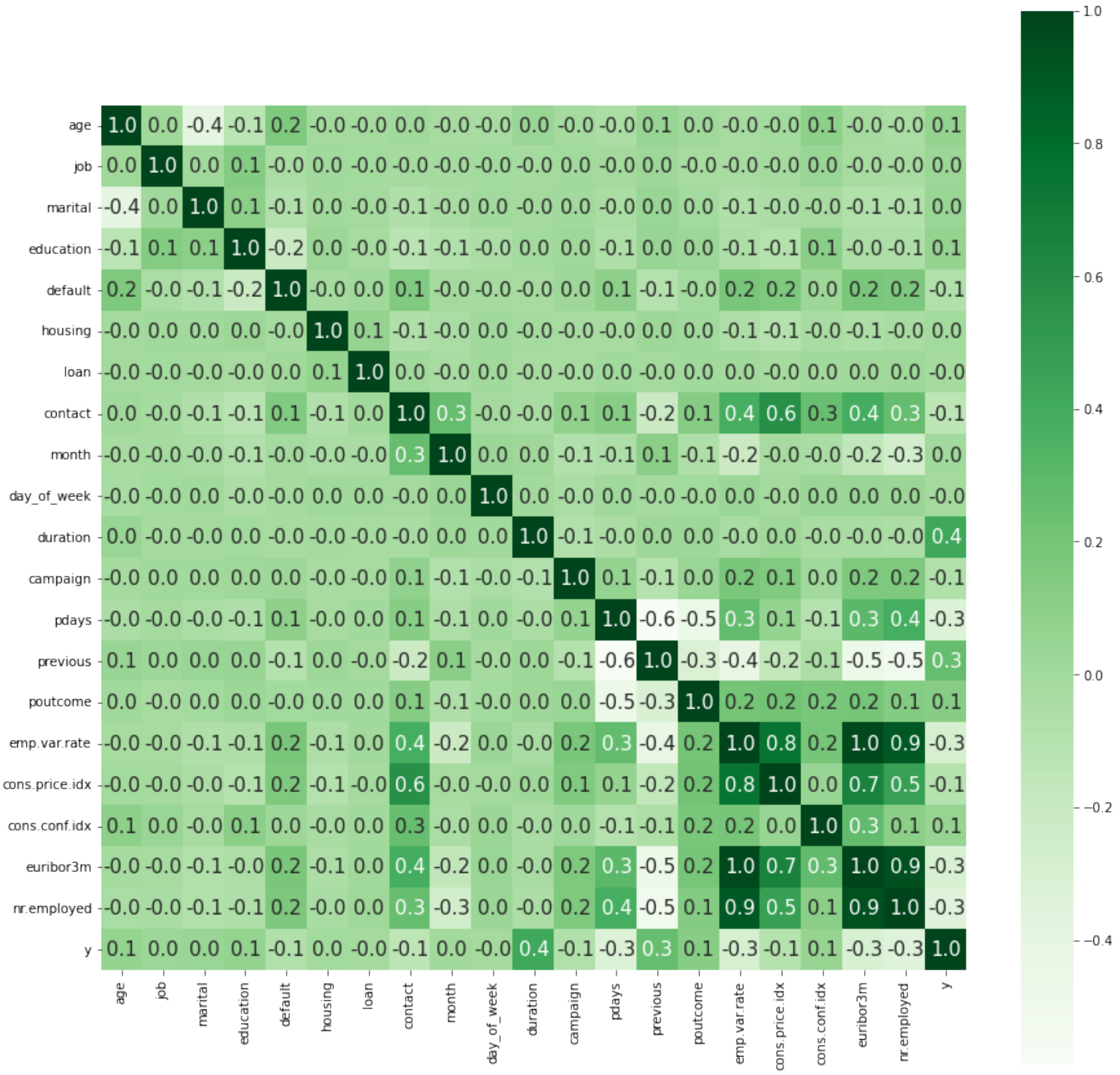
	age	job	marital	education	default	housing	loan	contact	n
y									
no	39.895311	3.791167	1.167666	3.729553	0.206107	1.081243	0.351690	0.379226	4.2
yes	41.889135	4.099778	1.252772	4.192905	0.108647	1.084257	0.321508	0.168514	4.3

2 rows × 21 columns

In [68]:
```
#check correlation of the dataset
correleation_matrix = new_df.corr()
```

In [69]:
```
rcParams['figure.figsize'] = 15,15
sns.heatmap(correleation_matrix, cbar=True, square= True,fmt='.1f',
```

Out[69]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7efdc2946c50>
```



	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration	campaign	pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	y
age	1.0	0.0	-0.4	-0.1	0.2	-0.0	-0.0	0.0	-0.0	-0.0	0.0	-0.0	-0.0	0.1	0.0	-0.0	-0.0	0.1	-0.0	-0.0	0.1
job	0.0	1.0	0.0	0.1	-0.0	0.0	-0.0	-0.0	-0.0	-0.0	-0.0	0.0	-0.0	0.0	-0.0	-0.0	-0.0	0.0	-0.0	-0.0	0.0
marital	-0.4	0.0	1.0	0.1	-0.1	0.0	-0.0	-0.1	-0.0	0.0	-0.0	0.0	-0.0	0.0	0.0	-0.1	-0.0	-0.0	-0.1	-0.1	0.0
education	-0.1	0.1	0.1	1.0	-0.2	0.0	-0.0	-0.1	-0.1	-0.0	0.0	0.0	-0.1	0.0	0.0	-0.1	-0.1	0.1	-0.0	-0.1	0.1
default	0.2	-0.0	-0.1	-0.2	1.0	-0.0	0.0	0.1	-0.0	-0.0	-0.0	0.0	0.1	-0.1	-0.0	0.2	0.2	0.0	0.2	0.2	-0.1
housing	-0.0	0.0	0.0	0.0	-0.0	1.0	0.1	-0.1	-0.0	0.0	-0.0	-0.0	-0.0	0.0	0.0	-0.1	-0.1	-0.0	-0.1	-0.0	0.0
loan	-0.0	-0.0	-0.0	-0.0	0.0	0.1	1.0	0.0	-0.0	0.0	-0.0	-0.0	0.0	-0.0	0.0	0.0	-0.0	0.0	0.0	0.0	-0.0
contact	0.0	-0.0	-0.1	-0.1	0.1	-0.1	0.0	1.0	0.3	-0.0	-0.0	0.1	0.1	-0.2	0.1	0.4	0.6	0.3	0.4	0.3	-0.1
month	-0.0	-0.0	-0.0	-0.1	-0.0	-0.0	-0.0	0.3	1.0	0.0	0.0	-0.1	-0.1	0.1	-0.1	-0.2	-0.0	-0.0	-0.2	-0.3	0.0
day_of_week	-0.0	-0.0	0.0	-0.0	-0.0	0.0	0.0	-0.0	0.0	1.0	0.0	-0.0	0.0	-0.0	-0.0	0.0	-0.0	0.0	0.0	0.0	-0.0
duration	0.0	-0.0	-0.0	0.0	-0.0	-0.0	-0.0	-0.0	0.0	0.0	1.0	-0.1	-0.0	0.0	0.0	-0.0	0.0	-0.0	-0.0	-0.0	0.4
campaign	-0.0	0.0	0.0	0.0	0.0	-0.0	-0.0	0.1	-0.1	-0.0	-0.1	1.0	0.1	-0.1	0.0	0.2	0.1	0.0	0.2	0.2	-0.1
pdays	-0.0	-0.0	-0.0	-0.1	0.1	-0.0	0.0	0.1	-0.1	0.0	-0.0	0.1	1.0	-0.6	-0.5	0.3	0.1	-0.1	0.3	0.4	-0.3
previous	0.1	0.0	0.0	0.0	-0.1	0.0	-0.0	-0.2	0.1	-0.0	0.0	-0.1	-0.6	1.0	-0.3	-0.4	-0.2	-0.1	-0.5	-0.5	0.3
poutcome	0.0	-0.0	0.0	0.0	-0.0	0.0	0.0	0.1	-0.1	-0.0	0.0	0.0	-0.5	-0.3	1.0	0.2	0.2	0.2	0.2	0.1	0.1
emp.var.rate	-0.0	-0.0	-0.1	-0.1	0.2	-0.1	0.0	0.4	-0.2	0.0	-0.0	0.2	0.3	-0.4	0.2	1.0	0.8	0.2	1.0	0.9	-0.3
cons.price.idx	-0.0	-0.0	-0.0	-0.1	0.2	-0.1	-0.0	0.6	-0.0	-0.0	0.0	0.1	0.1	-0.2	0.2	0.8	1.0	0.0	0.7	0.5	-0.1
cons.conf.idx	0.1	0.0	-0.0	0.1	0.0	-0.0	0.0	0.3	-0.0	0.0	-0.0	0.0	-0.1	-0.1	0.2	0.2	0.0	1.0	0.3	0.1	0.1
euribor3m	-0.0	-0.0	-0.1	-0.0	0.2	-0.1	0.0	0.4	-0.2	0.0	-0.0	0.2	0.3	-0.5	0.2	1.0	0.7	0.3	1.0	0.9	-0.3
nr.employed	-0.0	-0.0	-0.1	-0.1	0.2	-0.0	0.0	0.3	-0.3	0.0	-0.0	0.2	0.4	-0.5	0.1	0.9	0.5	0.1	0.9	1.0	-0.3
y	0.1	0.0	0.0	0.1	-0.1	0.0	-0.0	-0.1	0.0	-0.0	0.4	-0.1	-0.3	0.3	0.1	-0.3	-0.1	0.1	-0.3	-0.3	1.0

In [88]:
```
#spliting the dataset in X and Y
X = new_df.drop(['y'],axis=1)
Y = new_df['y']
```

In [89]:
```
print(X)
print(Y)
```

```
       age  job  marital  education  default  housing  loan  contac
t  month  \
0       30    1        1          2        0        2     0
0       6
1       39    7        2          3        0        0     0
1       6
2       25    7        1          3        0        2     0
1       4
```

```
3      38    7        1          2        0        1     1
1      4
4      47    0        1          6        0        2     0
0      7
...   ...  ...      ...        ...      ...      ...   ...      ..
.    ...
4114   30    0        1          1        0        2     2
0      3
4115   39    0        1          3        0        2     0
1      3
4116   27    8        2          3        0        0     0
0      6
4117   58    0        1          3        0        0     0
0      1
4118   34    4        2          3        0        2     0
0      7

      day_of_week  duration  campaign  pdays  previous  poutcome  
\
0               0       487         2    999         0         1
1               0       346         4    999         0         1
2               4       227         1    999         0         1
3               0        17         3    999         0         1
4               1        58         1    999         0         1
...           ...       ...       ...    ...       ...       ...
4114            2        53         1    999         0         1
4115            0       219         1    999         0         1
4116            1        64         2    999         1         0
4117            0       528         1    999         0         1
4118            4       175         1    999         0         1

      emp.var.rate  cons.price.idx  cons.conf.idx  euribor3m  nr.e
mployed
0             -1.8          92.893          -46.2      1.313      
5099.1
1              1.1          93.994          -36.4      4.855      
5191.0
2              1.4          94.465          -41.8      4.962      
5228.1
3              1.4          94.465          -41.8      4.959      
5228.1
4             -0.1          93.200          -42.0      4.191      
5195.8
...            ...             ...            ...        ...      
...
4114           1.4          93.918          -42.7      4.958      
5228.1
4115           1.4          93.918          -42.7      4.959      
5228.1
4116          -1.8          92.893          -46.2      1.354      
5099.1
4117           1.4          93.444          -36.1      4.966      
5228.1
```

```
4118          -0.1          93.200          -42.0      4.120
5195.8

[4119 rows x 20 columns]
0       0
1       0
2       0
3       0
4       0
       ..
4114    0
4115    0
4116    0
4117    0
4118    0
Name: y, Length: 4119, dtype: int64
```

In [90]:

```
#spliting the dataset in train and test
X_train,X_test,Y_train,Y_test = model_selection.train_test_split(X,
```

In [91]:

```
#print shape of X_train and Y_train
print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape)
```

```
(3707, 20) (412, 20) (3707,) (412,)
```

Loading the Model

In [92]:

```
model = LogisticRegression()
```

In [93]:

```
model.fit(X_train,Y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logis
tic.py:818: ConvergenceWarning: lbfgs failed to converge (status=1
):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as
shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
(https://scikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver opti
ons:
    https://scikit-learn.org/stable/modules/linear_model.html#logi
stic-regression (https://scikit-learn.org/stable/modules/linear_mo
del.html#logistic-regression)
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

Out[93]:

```
LogisticRegression()
```

In [95]:

```
y_pred = model.predict(X_test)
```

In [97]:

```
from sklearn import metrics
f1 = metrics.f1_score(y_true=Y_test,y_pred=y_pred)
acc = metrics.accuracy_score(y_true=Y_test,y_pred=y_pred)
pres = metrics.precision_score(y_true=Y_test,y_pred=y_pred)
recall = metrics.recall_score(y_true=Y_test,y_pred=y_pred)
```

In [98]:

```
print("The accuracy of the model Logistic Regression Model",acc)
print("The F1 Score of the model Logistic Regression Model",f1)
print("The Precision of the model Logistic Regression Model",pres)
print("The recall of the model Logistic Regression Model",recall)
```

```
The accuracy of the model Logistic Regression Model 0.910194174757
2816
The F1 Score of the model Logistic Regression Model 0.478873239436
6197
The Precision of the model Logistic Regression Model 0.58620689655
17241
The recall of the model Logistic Regression Model 0.40476190476190
477
```

using ensemble learning

In [99]:

```
from sklearn import ensemble
```

In [100]:

```
#using RandomForestClassifier()
RFC = ensemble.RandomForestClassifier()
```

In [101]:

```
RFC.fit(X_train,Y_train)
```

Out[101]:

```
RandomForestClassifier()
```

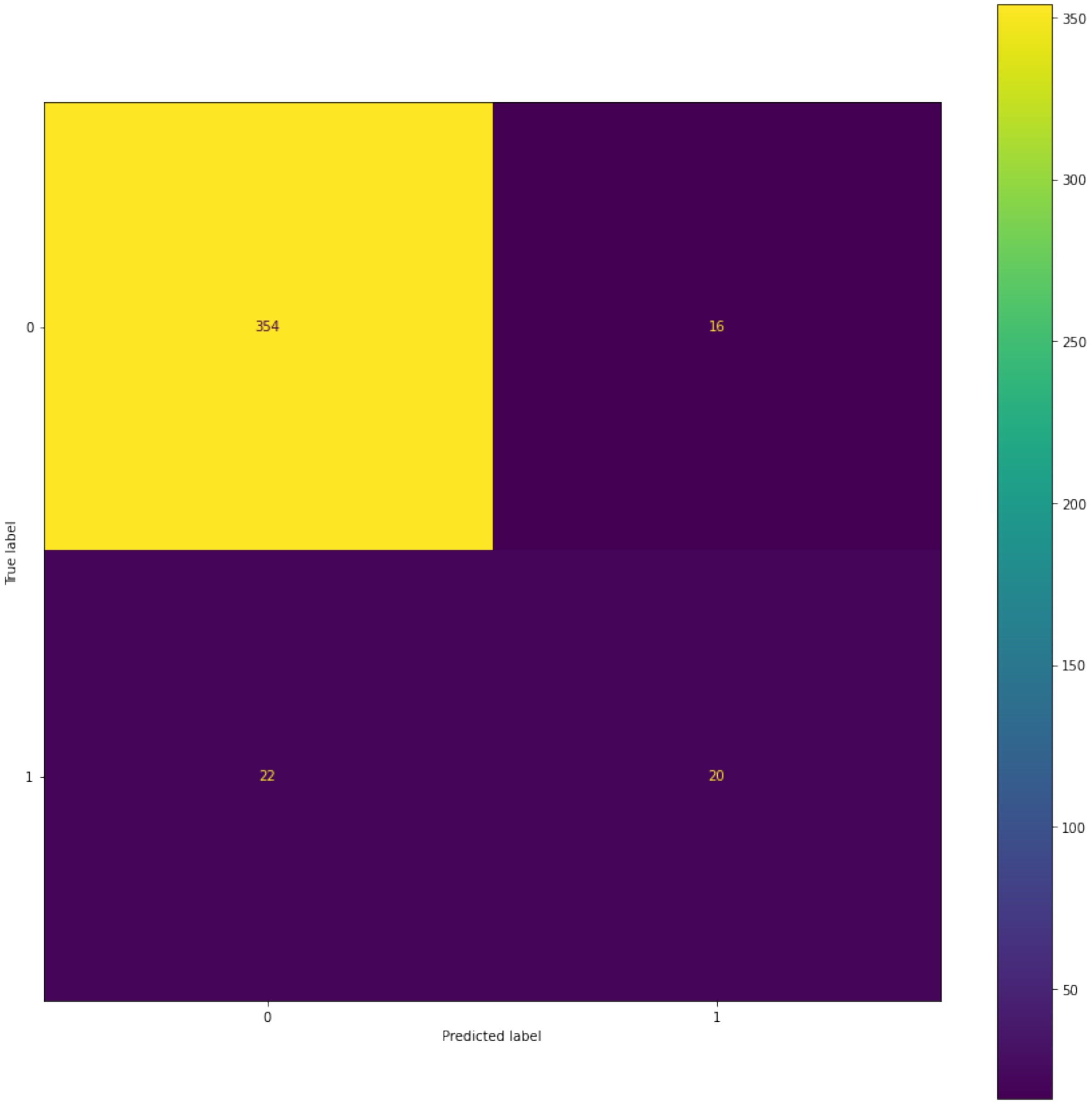
In [102]:

```
y_pred_rfc = RFC.predict(X_test)
```

In [103]:

```
f1_rfc = metrics.f1_score(y_true=Y_test,y_pred=y_pred_rfc)
acc_rfc = metrics.accuracy_score(y_true=Y_test,y_pred=y_pred_rfc)
pres_rfc = metrics.precision_score(y_true=Y_test,y_pred=y_pred_rfc)
recall_rfc = metrics.recall_score(y_true=Y_test,y_pred=y_pred_rfc)
cfn_matrix = metrics.plot_confusion_matrix(RFC,X_test,Y_test)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.p
y:87: FutureWarning: Function plot_confusion_matrix is deprecated;
Function `plot_confusion_matrix` is deprecated in 1.0 and will be
removed in 1.2. Use one of the class methods: ConfusionMatrixDispl
ay.from_predictions or ConfusionMatrixDisplay.from_estimator.
  warnings.warn(msg, category=FutureWarning)
```



In [104]:

```
#check cfn_matrix
cfn_matrix
```

Out[104]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at
0x7efdbfc9c990>
```

In [105]:

```
print("The accuracy of the model RandomForestClassifier Model",acc_
print("The F1 Score of the model RandomForestClassifier Model",f1_r
print("The Precision of the model RandomForestClassifier Model",pre
print("The recall of the model RandomForestClassifier",recall_rfc)
```

```
The accuracy of the model RandomForestClassifier Model 0.907766990
2912622
The F1 Score of the model RandomForestClassifier Model 0.512820512
8205129
The Precision of the model RandomForestClassifier Model 0.55555555
55555556
The recall of the model RandomForestClassifier 0.47619047619047616
```

In [106]:

```
#save model
import pickle
filename = 'random_forest.pkl'
pickle.dump(RFC, open(filename, 'wb'))
```

In [ ]: