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

In [2]:
```
df = pd.read_csv('C:\\Users\\anitt\\Downloads\\bank-additional.csv',delimit
df.head()
```

Out[2]:

	age	job	marital	education	default	housing	loan	contact	month	day_
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 [6]:
```
df.tail()
```

Out[6]:

	age	job	marital	education	default	housing	loan	contact	month	day_c
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 [7]:
```
df.shape
```

Out[7]:
```
(4119, 21)
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['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'],
      dtype='object')
```

In [9]:
```
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   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]:
```
df.describe()
```

Out[10]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.p
count	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119
mean	40.113620	256.788055	2.537266	960.422190	0.190337	0.084972	93
std	10.313362	254.703736	2.568159	191.922786	0.541788	1.563114	0
min	18.000000	0.000000	1.000000	0.000000	0.000000	-3.400000	92
25%	32.000000	103.000000	1.000000	999.000000	0.000000	-1.800000	93
50%	38.000000	181.000000	2.000000	999.000000	0.000000	1.100000	93
75%	47.000000	317.000000	3.000000	999.000000	0.000000	1.400000	93
max	88.000000	3643.000000	35.000000	999.000000	6.000000	1.400000	94

In [11]:

```
df.isnull().sum()
```

Out[11]:

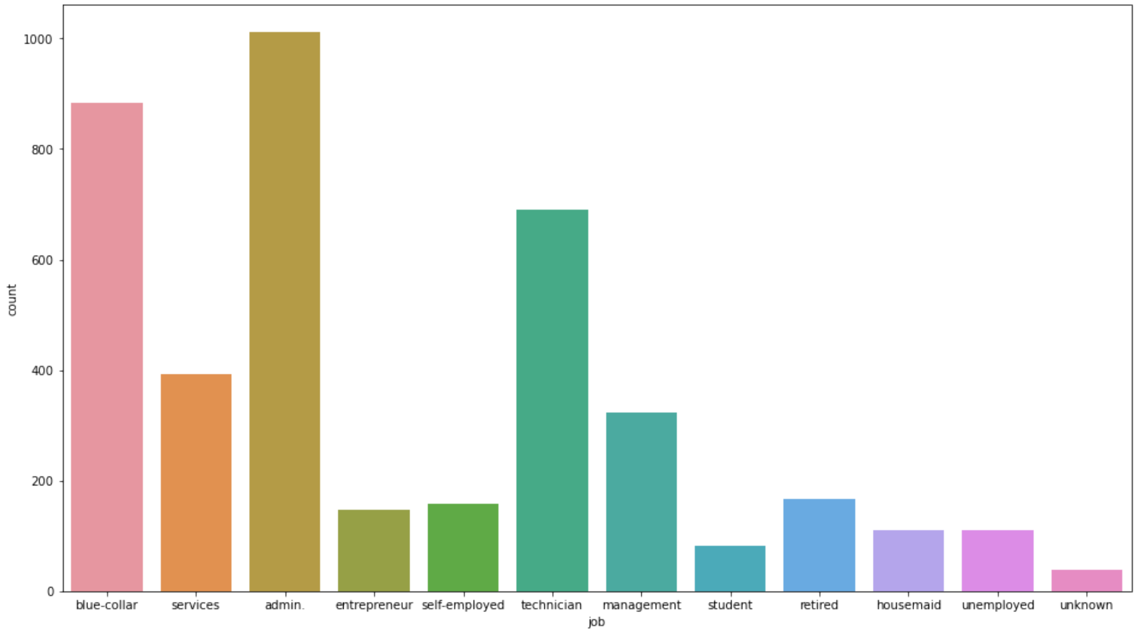
```
age               0
job               0
marital           0
education         0
default           0
housing           0
loan              0
contact           0
month             0
day_of_week       0
duration          0
campaign          0
pdays             0
previous          0
poutcome        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

In [12]:

```
plt.figure(figsize = (16,9))
sns.countplot(x = "job",data = df)
```

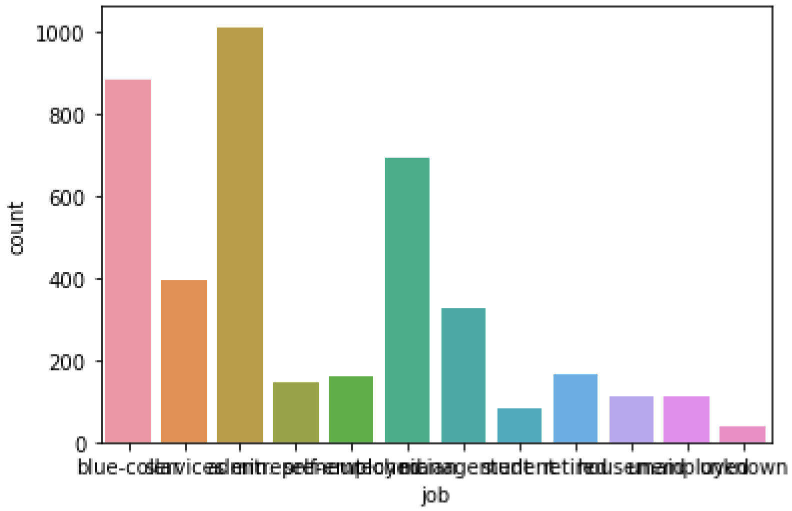
Out[12]:

```
<AxesSubplot:xlabel='job', ylabel='count'>
```



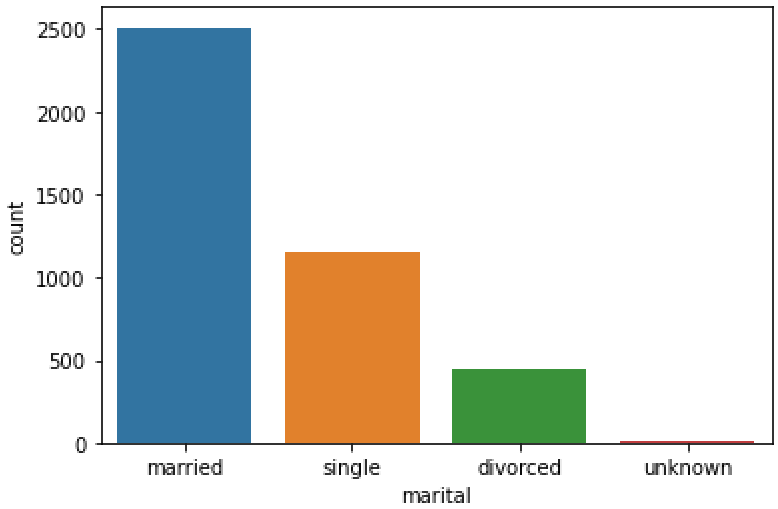
```
In [13]: sns.countplot(x = "job",data = df)
```

```
Out[13]: <AxesSubplot:xlabel='job', ylabel='count'>
```



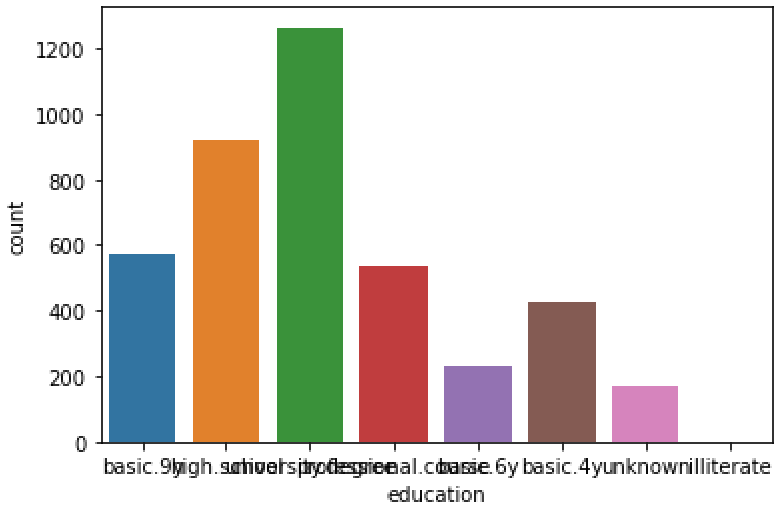
```
In [14]: sns.countplot(x = "marital",data = df)
```

```
Out[14]: <AxesSubplot:xlabel='marital', ylabel='count'>
```



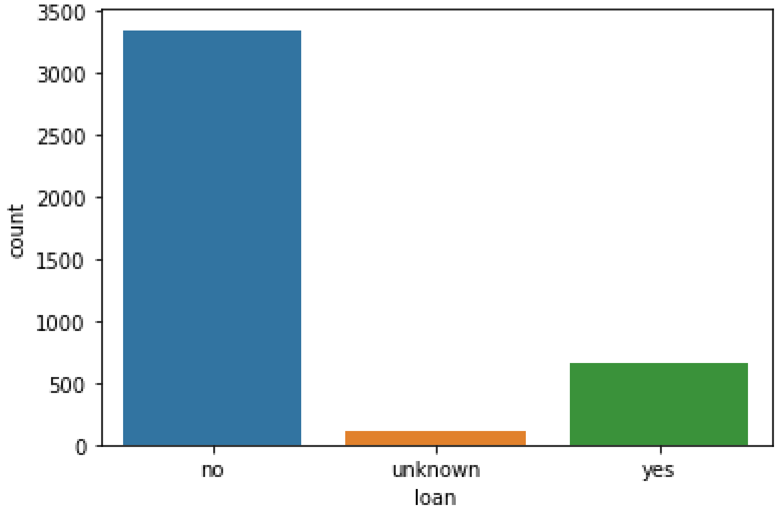
```
In [15]: sns.countplot(x = "education",data = df)
```

```
Out[15]: <AxesSubplot:xlabel='education', ylabel='count'>
```



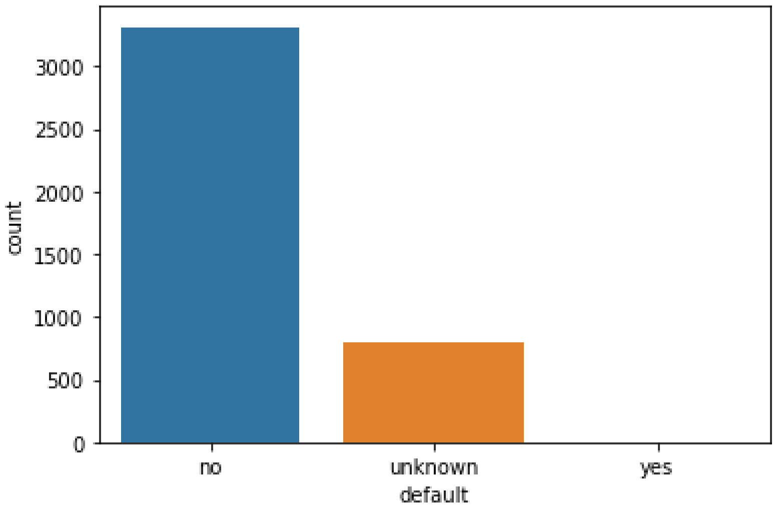
```
In [16]: sns.countplot(x = "loan",data = df)
```

```
Out[16]: <AxesSubplot:xlabel='loan', ylabel='count'>
```



In [17]:
```
sns.countplot(x = "default",data = df)
```

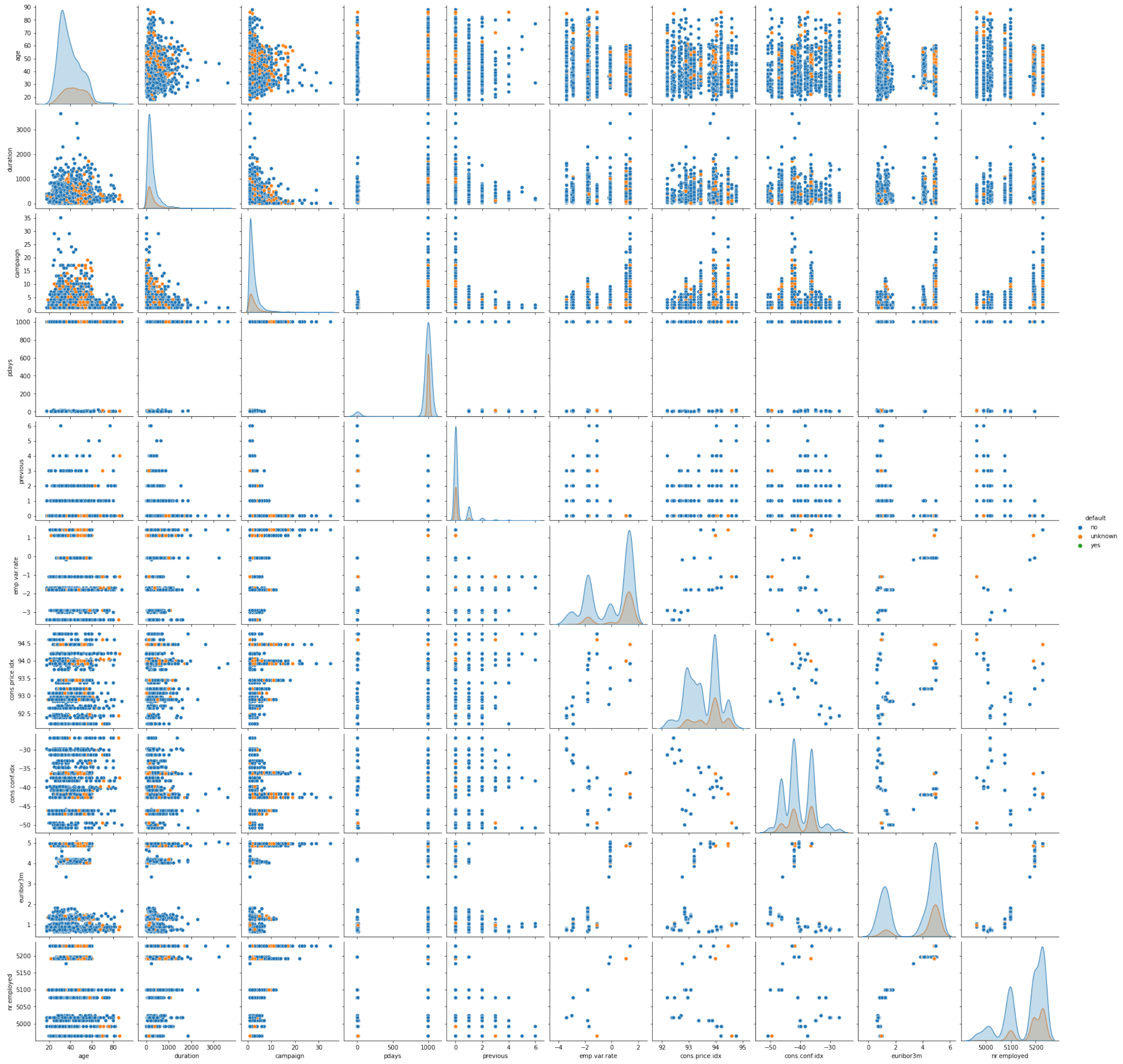
Out[17]:
```
<AxesSubplot:xlabel='default', ylabel='count'>
```



In [3]:
```
plt.figure(figsize = (16,9))
sns.pairplot(data = df,hue = "default")
```

Out[3]:
```
<seaborn.axisgrid.PairGrid at 0x20d2ced0310>

<Figure size 1152x648 with 0 Axes>
```



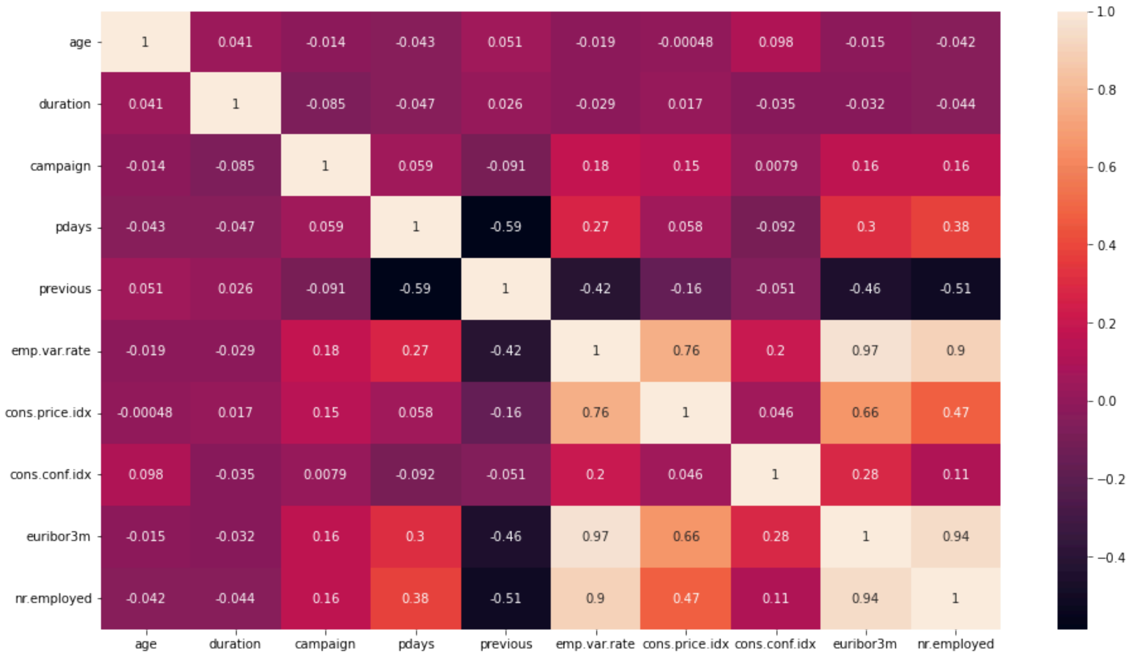
In [4]:
```
my_df=df.select_dtypes(exclude=[object])
my_df.corr()
```

Out[4]:

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.
age	1.000000	0.041299	-0.014169	-0.043425	0.050931	-0.019192	-0.0004
duration	0.041299	1.000000	-0.085348	-0.046998	0.025724	-0.028848	0.0166
campaign	-0.014169	-0.085348	1.000000	0.058742	-0.091490	0.176079	0.1450
pdays	-0.043425	-0.046998	0.058742	1.000000	-0.587941	0.270684	0.0584
previous	0.050931	0.025724	-0.091490	-0.587941	1.000000	-0.415238	-0.1649
emp.var.rate	-0.019192	-0.028848	0.176079	0.270684	-0.415238	1.000000	0.7551
cons.price.idx	-0.000482	0.016672	0.145021	0.058472	-0.164922	0.755155	1.0000
cons.conf.idx	0.098135	-0.034745	0.007882	-0.092090	-0.051420	0.195022	0.0458
euribor3m	-0.015033	-0.032329	0.159435	0.301478	-0.458851	0.970308	0.6571
nr.employed	-0.041936	-0.044218	0.161037	0.381983	-0.514853	0.897173	0.4725

In [5]:
```
plt.figure(figsize = (16,9))
sns.heatmap(my_df.corr(),annot = True)
```

Out[5]:
```
<AxesSubplot:>
```



In [6]:
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
```

In [7]:

```
df["job"] = le.fit_transform(df["job"])
df["marital"] = le.fit_transform(df["marital"])
df["education"] = le.fit_transform(df["education"])
df["default"] = le.fit_transform(df["default"])
df["loan"] = le.fit_transform(df["loan"])
df["contact"] = le.fit_transform(df["contact"])
df["poutcome"] = le.fit_transform(df["poutcome"])
df["housing"] = le.fit_transform(df["housing"])
df["month"] = le.fit_transform(df["month"])
```

In [8]:

```
df.head()
```

Out[8]:

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

5 rows × 21 columns

In [9]:

```
df.drop(["pdays","previous","poutcome"],axis = 1)
df.head()
```

Out[9]:

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

5 rows × 21 columns

In [ ]: