INPUT

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

INPUT

```
df=pd.read_csv("/content/bank-additional.csv",delimiter=';')
df.rename(columns={'y':'deposit'},inplace=True)
print("DISPLAY HEAD DATA")
print('-------------------')
df.head()
```

OUTPUT

DISPLAY HEAD DATA
----

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

5 rows × 21 columns

pdays	previous	poutcome	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	deposit
999	0	nonexistent	-1.8	92.893	-46.2	1.313	5099.1	no
999	0	nonexistent	1.1	93.994	-36.4	4.855	5191.0	no
999	0	nonexistent	1.4	94.465	-41.8	4.962	5228.1	no
999	0	nonexistent	1.4	94.465	-41.8	4.959	5228.1	no
999	0	nonexistent	-0.1	93.200	-42.0	4.191	5195.8	no

INPUT

```
print('DISPLAY DATA INFO')
print('--------------------')
df.info()
```

OUTPUT

DISPLAY 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        2727 non-null   float64
 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  deposit         4119 non-null   object
```

dtypes: float64(6), int64(4), object(11)

memory usage: 675.9+ KB

INPUT

```
print('DISPLAY TAIL DATA')
print('-------------------')
df.tail()
```

OUTPUT

```
DISPLAY TAIL DATA
-------------------
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome
4114	30	admin.	married	basic.6y	no	yes	yes	cellular	jul	thu	...	1	999	0	nonexistent
4115	39	admin.	married	high.school	no	yes	no	telephone	jul	fri	...	1	999	0	nonexistent
4116	27	student	single	high.school	no	no	no	cellular	may	mon	...	2	999	1	failure
4117	58	admin.	married	high.school	no	no	no	cellular	aug	fri	...	1	999	0	nonexistent
4118	34	management	single	high.school	no	yes	no	cellular	nov	wed	...	1	999	0	nonexistent

5 rows × 21 columns

emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	deposit
1.4	93.918	-42.7	4.958	5228.1	no
1.4	93.918	-42.7	4.959	5228.1	no
-1.8	92.893	-46.2	1.354	5099.1	no
1.4	93.444	-36.1	4.966	5228.1	no
-0.1	93.200	-42.0	4.120	5195.8	no

INPUT

```
print('DISPLAYING SHAPE')
df.shape
```

OUTPUT

```
DISPLAYING SHAPE
(4119, 21)
```

INPUT

```
df.columns
```

OUTPUT

```
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', 'deposit'],
      dtype='object')
```

INPUT

```
print('DISPLAY DATA TYPE')
print('--------------------')
df.dtypes
```

OUTPUT

```
DISPLAY DATA TYPE
--------------------
```

	0
age	int64
job	object
marital	object
education	object
default	object
housing	object
loan	object
contact	object
month	object
day_of_week	object
duration	float64
campaign	int64
pdays	int64
previous	int64
poutcome	object
emp.var.rate	float64
cons.price.idx	float64
cons.conf.idx	float64
euribor3m	float64
nr.employed	float64
deposit	object

dtype: object

INPUT

```
df.dtypes.value_counts()
```

OUTPUT

	count
object	11
int64	5
float64	5

dtype: int64

INPUT

```
print('DUPLICATED VALUE')
df.duplicated().sum()
```

OUTPUT

DUPLICATED VALUE

1

INPUT

df.isna().sum()

OUTPUT

	0
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
deposit	0

INPUT

```
print('DISPLAYING COLUMNS')
print('--------------------')
cat_cols=df.select_dtypes(include='object').columns
print(cat_cols)
num_cols=df.select_dtypes(exclude='object').columns
print(num_cols)
```

OUTPUT

```
DISPLAYING COLUMNS
--------------------
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'day_of_week', 'poutcome', 'deposit'],
      dtype='object')
Index(['age', 'duration', 'campaign', 'pdays', 'previous', 'emp.var.rate',
       'cons.price.idx', 'cons.conf.idx', 'euribor3m', 'nr.employed'],
      dtype='object')
```

INPUT

```
print('DISPLAYING DATA')
print('-----------------')
df.describe()
```

OUTPUT

```
DISPLAYING DATA
-----------------
```

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	4119.000000	2727.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000	4119.000000
mean	40.113620	129.639897	2.537266	960.422190	0.190337	0.084972	93.579704	-40.499102	3.621356	5166.481695
std	10.313362	63.753556	2.568159	191.922786	0.541788	1.563114	0.579349	4.594578	1.733591	73.667904
min	18.000000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.635000	4963.600000
25%	32.000000	81.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.334000	5099.100000
50%	38.000000	128.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	47.000000	180.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	88.000000	252.000000	35.000000	999.000000	6.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

INPUT

```
df.describe(include='object')
```

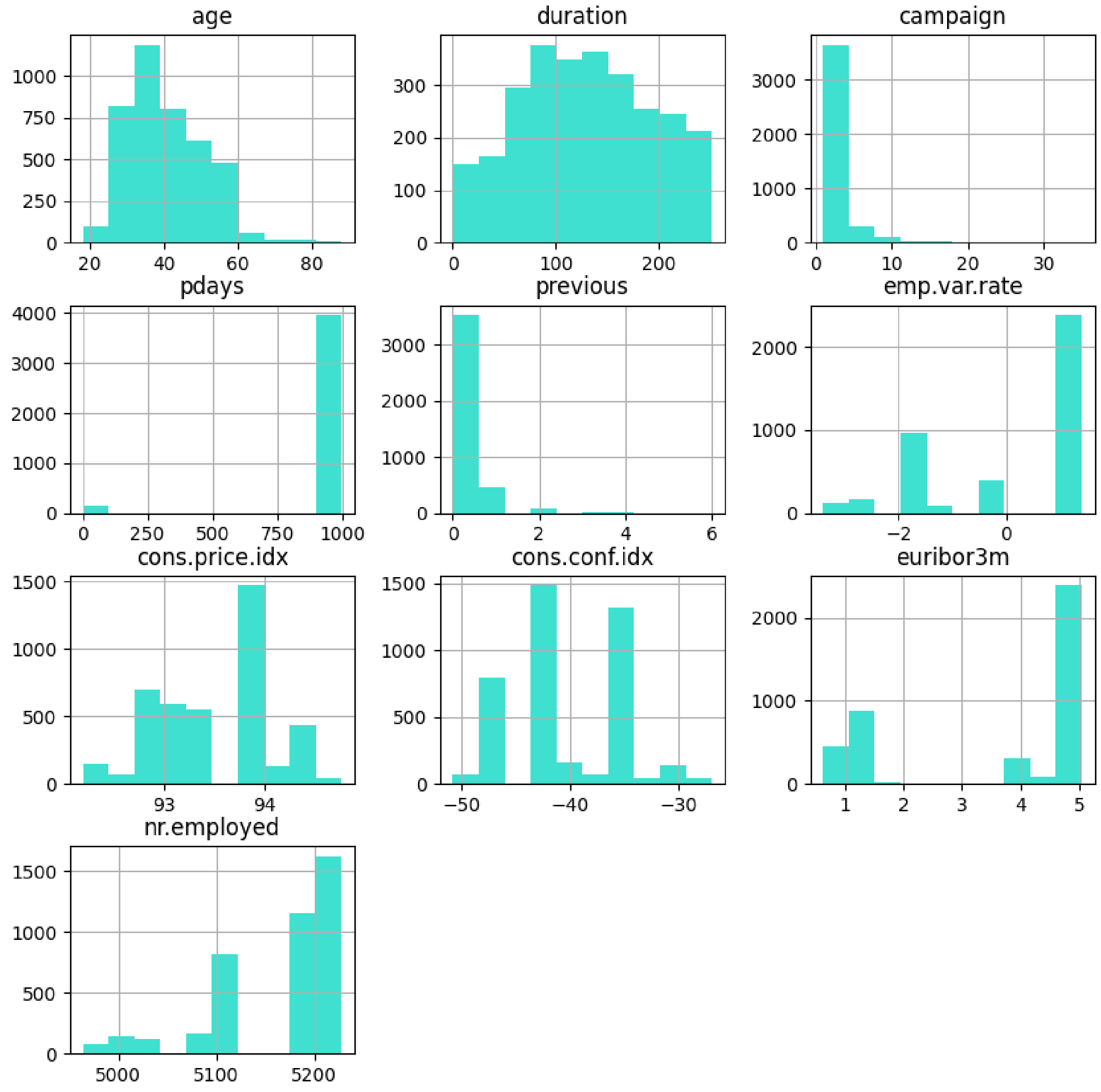
OUTPUT

	job	marital	education	default	housing	loan	contact	month	day_of_week	poutcome	deposit
count	4119	4119	4119	4119	4119	4119	4119	4119	4119	4119	4119
unique	12	4	8	3	3	3	2	10	5	3	2
top	admin.	married	university.degree	no	yes	no	cellular	may	thu	nonexistent	no
freq	1012	2509	1264	3315	2175	3349	2652	1378	860	3523	3668

INPUT

```
print('CONSTRUCTING HISTOGRAMS')

print('----------------------')

df.hist(figsize=(10,10),color='#40E0D0')

plt.show()
```

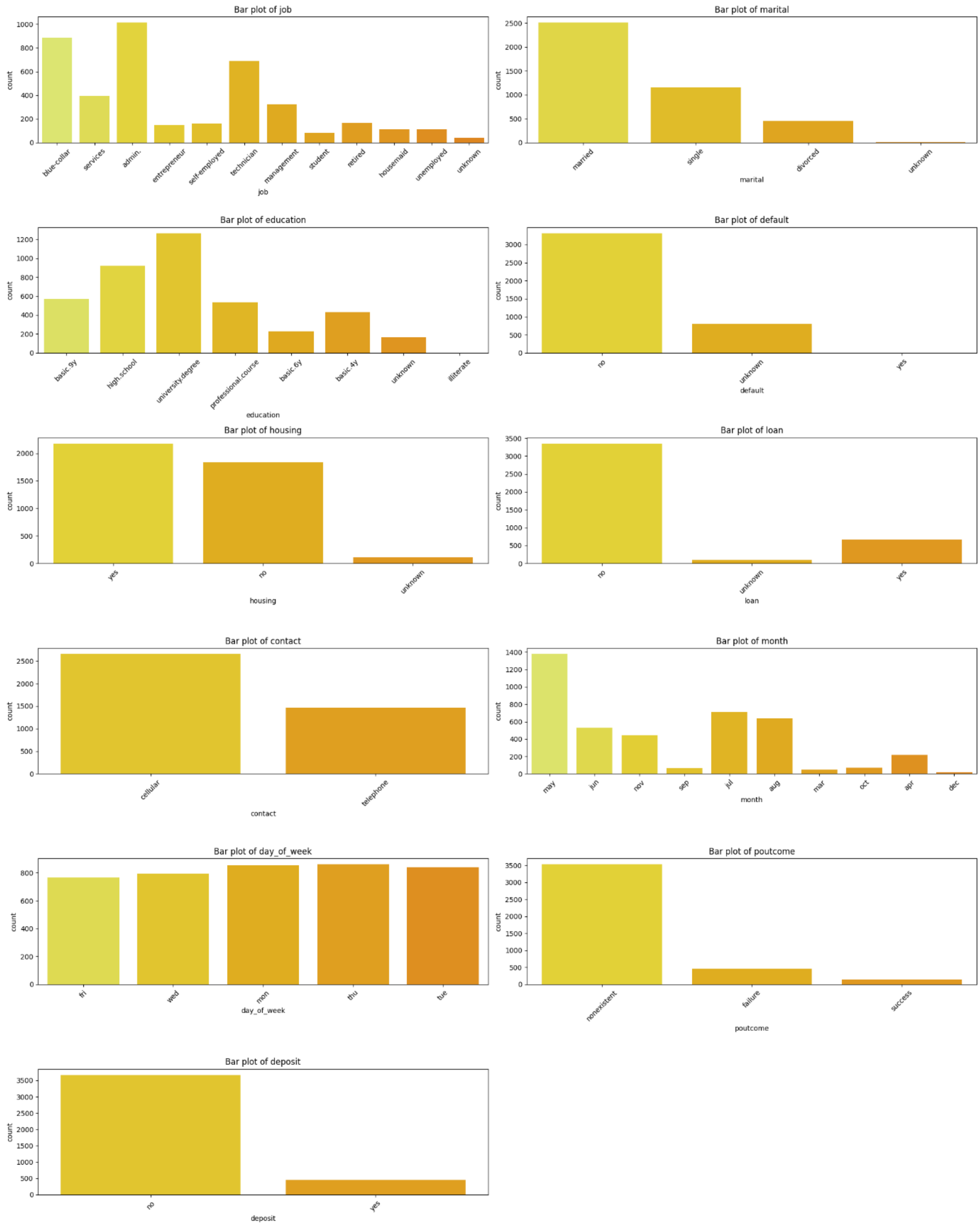
OUTPUT



INPUT

```
print('CONSTRUCTING BARPLOT')
print('----------------------')
num_plots=len(cat_cols)
num_rows=(num_plots+1)//2
num_cols=2
plt.figure(figsize=(20,25))
for i,feature in enumerate(cat_cols,1):
  plt.subplot(num_rows,num_cols,i)
  sns.countplot(x=feature,data=df,palette='Wistia')
  plt.title(f'Bar plot of {feature}')
  plt.xlabel(feature)
  plt.ylabel('count')
  plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

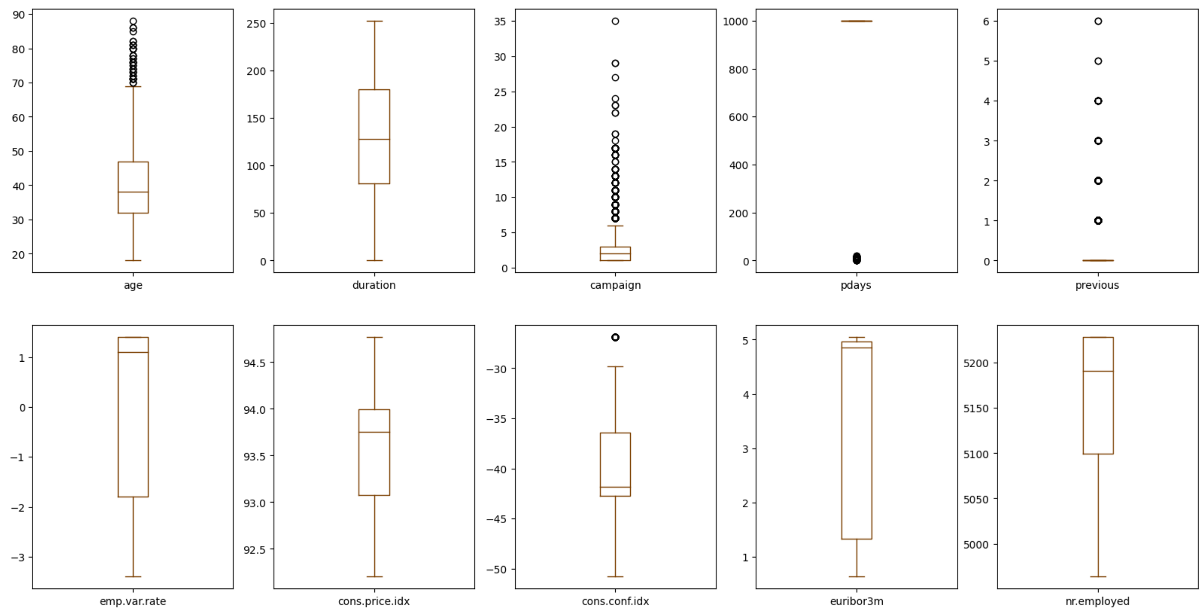
OUTPUT


Bar plot of job
count
job
blue-collar
services
admin.
entrepreneur
self-employed
technician
management
student
retired
housemaid
unemployed
unknown
Bar plot of marital
marital
married
single
divorced
unknown
Bar plot of education
education
basic.9y
high.school
university.degree
professional.course
basic.6y
basic.4y
unknown
illiterate
Bar plot of default
default
no
unknown
yes
Bar plot of housing
housing
yes
no
unknown
Bar plot of loan
loan
no
unknown
yes
Bar plot of contact
contact
cellular
telephone
Bar plot of month
month
may
jun
nov
sep
jul
aug
mar
oct
apr
dec
Bar plot of day_of_week
day_of_week
fri
wed
mon
thu
tue
Bar plot of poutcome
poutcome
nonexistent
failure
success
Bar plot of deposit
deposit
no
yes

INPUT

```
print("CONSTRUCT CHART 1")

print('--------------------')

df.plot(kind='box',subplots=True,layout=(2,5),figsize=(20,10),color='#7b3f00')

plt.show()
```

OUTPUT



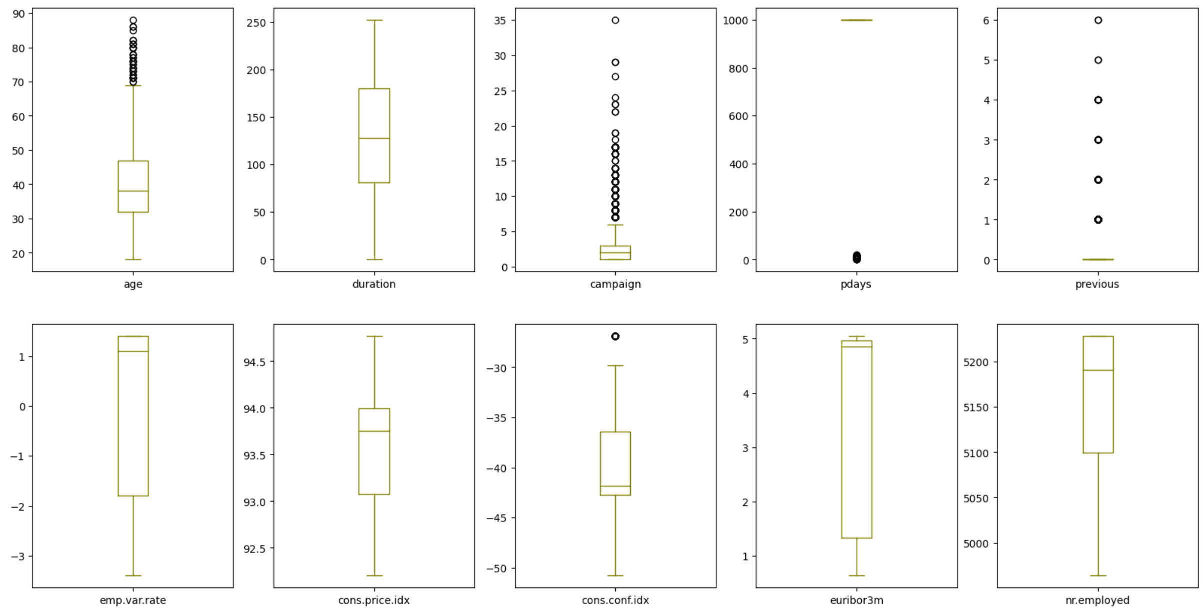
INPUT

```
column=df[['age','campaign','duration']]

q1=np.percentile(column,25)

q3=np.percentile(column,75)

iqr=q3-q1

lowerbound=q1- 1.5 *iqr

upperbound=q3+ 1.5 *iqr

df[['age','campaign','duration']]=column[(column>lowerbound)&(column<upperbound)]
```

INPUT

```
print('CONSTRUCT CHART 2')

print('--------------------')

df.plot(kind='box',subplots=True,layout=(2,5),figsize=(20,10),color='#808000')

plt.show()
```

OUTPUT



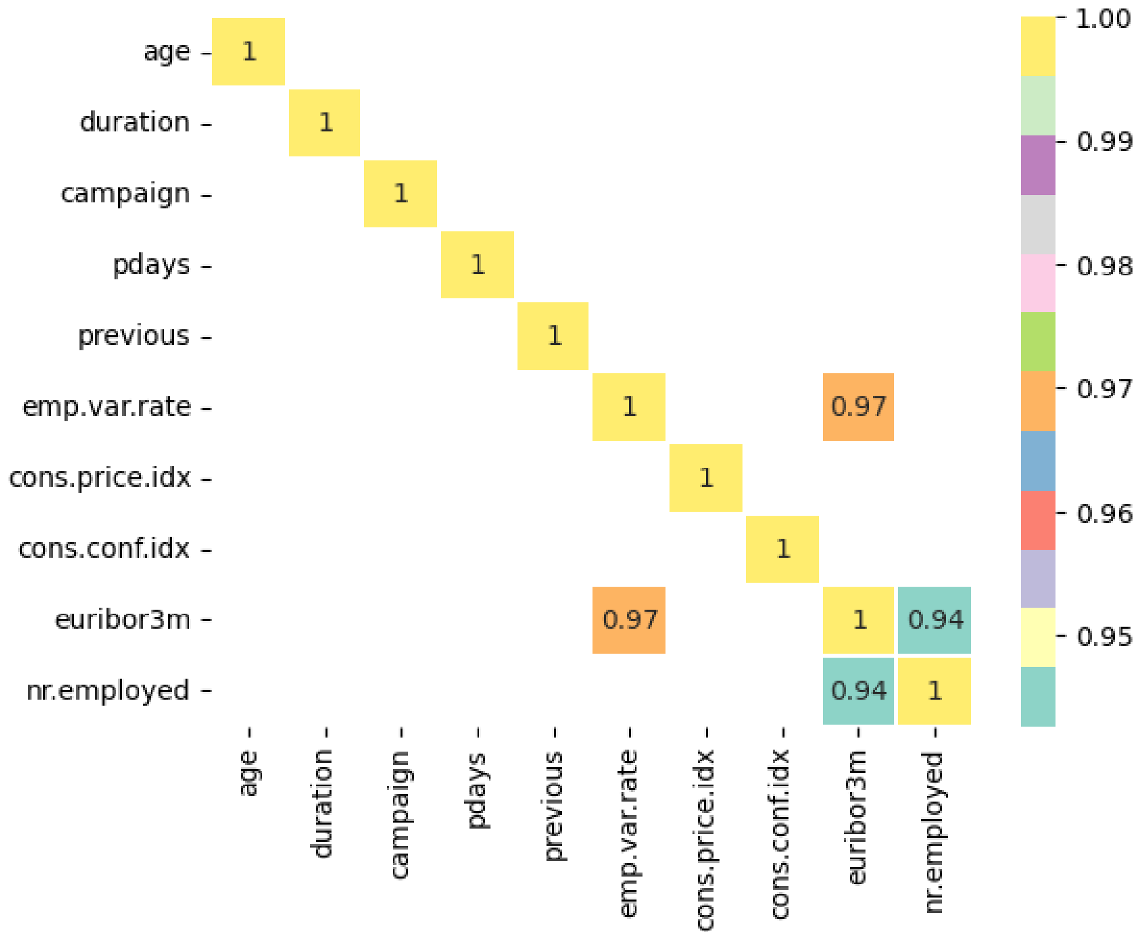
INPUT

```
numeric_df = df.drop(columns=cat_cols)

corr = numeric_df.corr()

print(corr)

corr=corr[abs(corr) >=0.90]

print('\n')

print('CONSTRUCT HEATMAP')

print('------------------')

print('\n')

sns.heatmap(corr, annot=True, cmap='Set3', linewidths=0.2)

plt.show()
```

OUTPUT

age	-0.019192	-0.000482	0.098135	-0.015033
duration	-0.063870	-0.013338	0.045889	-0.067815
campaign	0.176079	0.145021	0.007882	0.159435
pdays	0.270684	0.058472	-0.092090	0.301478
previous	-0.415238	-0.164922	-0.051420	-0.458851
emp.var.rate	1.000000	0.755155	0.195022	0.970308
cons.price.idx	0.755155	1.000000	0.045835	0.657159
cons.conf.idx	0.195022	0.045835	1.000000	0.276595
euribor3m	0.970308	0.657159	0.276595	1.000000
nr.employed	0.897173	0.472560	0.107054	0.942589

	nr.employed
age	-0.041936
duration	-0.097339
campaign	0.161037
pdays	0.381983
previous	-0.514853
emp.var.rate	0.897173
cons.price.idx	0.472560
cons.conf.idx	0.107054
euribor3m	0.942589
nr.employed	1.000000



INPUT

high_corr_cols=['emp.var.rate','euribor3m','nr.employed']

INPUT

df1=df.copy()

df1.columns

OUTPUT

```
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', 'deposit'],
      dtype='object')
```

INPUT

```
df1.drop(high_corr_cols,inplace=True,axis=1)
df1.columns
```

OUTPUT

```
Index(['age', 'job', 'marital', 'education', 'default', 'housing', 'loan',
       'contact', 'month', 'day_of_week', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'cons.price.idx', 'cons.conf.idx', 'deposit'],
      dtype='object')
```

INPUT

```
df1.shape
```

OUTPUT

```
(4119, 18)
```

INPUT

from sklearn.preprocessing import LabelEncoder

lb=LabelEncoder()

df_encoded=df1.apply(lb.fit_transform)

df_encoded

OUTPUT

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	duration
0	12	1	1	2	0	2	0	0	6	0	250
1	21	7	2	3	0	0	0	1	6	0	250
2	7	7	1	3	0	2	0	1	4	4	224
3	20	7	1	2	0	1	1	1	4	0	14
4	29	0	1	6	0	2	0	0	7	1	55
...	...	...	...	...	...	...	...	...	...	...	...
4114	12	0	1	1	0	2	2	0	3	2	50
4115	21	0	1	3	0	2	0	1	3	0	216
4116	9	8	2	3	0	0	0	0	6	1	61
4117	40	0	1	3	0	0	0	0	1	0	250
4118	16	4	2	3	0	2	0	0	7	4	172

4119 rows × 18 columns

campaign	pdays	previous	poutcome	cons.price.idx	cons.conf.idx	deposit
1	20	0	1	8	4	0
3	20	0	1	18	16	0
0	20	0	1	23	8	0
2	20	0	1	23	8	0
0	20	0	1	11	7	0
...	...	...	...	...	...	...
0	20	0	1	17	6	0
0	20	0	1	17	6	0
1	20	1	0	8	4	0
0	20	0	1	13	17	0
0	20	0	1	11	7	0

INPUT

```
df_encoded['deposit'].value_counts()
```

OUTPUT

	count
deposit	
0	3668
1	451

dtype: int64

INPUT

```
print('DISPLAYING SHAPE AND TYPE')
x=df_encoded.drop('deposit',axis=1)
y=df_encoded['deposit']
print(x.shape)
print(y.shape)
print(type(x))
print(type(y))
```

OUTPUT

```
DISPLAYING SHAPE AND TYPE
(4119, 17)
(4119,)
<class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.series.Series'>
```

INPUT

```
from sklearn.model_selection import train_test_split


x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=1)
print(x_train.shape)
print(y_train.shape)
print(y_train.shape)
print(y_test.shape)
```

OUTPUT

```
(3089, 17)
(3089,)
(3089,)
(1030,)
```

INPUT

```
from sklearn.metrics import
confusion_matrix,classification_report,accuracy_score
def eval_model(y_test,y_pred):
  acc=accuracy_score(y_test,y_pred)
  print('Accuracy_Score',acc)
  cm=confusion_matrix(y_test,y_pred)
  print('confusion matrix\n',cm)
  print('classification report\n',classification_report(y_test,y_pred))

def mscore(model):
    train_score=model.score(x_train,y_train)
    test_score=model.score(x_test,y_test)
    print('training score',train_score)
    print('testing score',test_score)
```

INPUT

```
from sklearn.tree import DecisionTreeClassifier

dt=DecisionTreeClassifier(criterion='gini',max_depth=5,min_samples
_split=10)
dt.fit(x_train,y_train)
```

OUTPUT

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(max_depth=5, min_samples_split=10)
```

INPUT

mscore(dt)

OUTPUT

training score 0.9148591777274199

testing score 0.8990291262135922

INPUT

ypred_dt=dt.predict(x_test)

print(ypred_dt)

OUTPUT

[0 0 1 ... 0 0 0]

INPUT

eval_model(y_test,ypred_dt)

OUTPUT

```
Accuracy_Score 0.8990291262135922
confusion matrix
 [[905  25]
 [ 79  21]]
classification report
              precision    recall  f1-score   support

           0       0.92      0.97      0.95       930
           1       0.46      0.21      0.29       100

    accuracy                           0.90      1030
   macro avg       0.69      0.59      0.62      1030
weighted avg       0.87      0.90      0.88      1030
```

INPUT

```
from sklearn.tree import plot_tree
```

INPUT

```
cn=['no','yes']
fn=x_train.columns
print(fn)
print(cn)
```

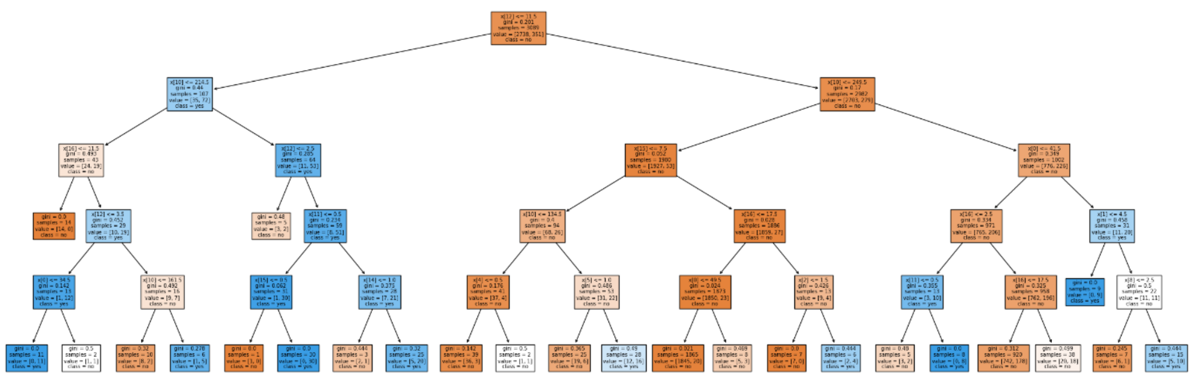
OUTPUT

```
Index(['age', 'job', 'marital', 'education', 'default', 'housing', 'loan',
       'contact', 'month', 'day_of_week', 'duration', 'campaign', 'pdays',
       'previous', 'poutcome', 'cons.price.idx', 'cons.conf.idx'],
      dtype='object')
['no', 'yes']
```

INPUT

```
print('DISPLAYING TREE PLOT 1')
print('-----------------------')
plt.figure(figsize=(30,10))
plot_tree(dt,class_names=cn,filled=True)
plt.show()
```

OUTPUT



INPUT

dt1=DecisionTreeClassifier(criterion='entropy',max_depth=4,min_samples_split=15)

dt1.fit(x_train,y_train)

OUTPUT

```
▾ DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', max_depth=4, min_samples_split=15)
```

INPUT

mscore(dt)

OUTPUT

training score 0.9148591777274199

testing score 0.8990291262135922

INPUT

ypred_dt1=dt1.predict(x_test)

INPUT

eval_model(y_test,ypred_dt1)

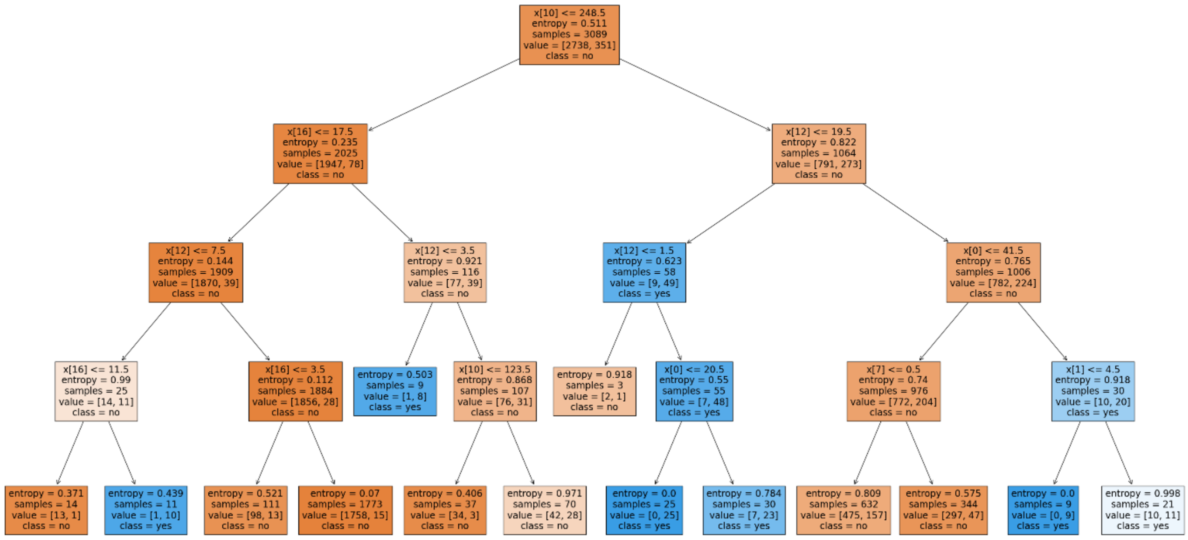
OUTPUT

```
Accuracy_Score 0.9048543689320389
confusion matrix
 [[915  15]
 [ 83  17]]
classification report
              precision    recall  f1-score   support

           0       0.92      0.98      0.95       930
           1       0.53      0.17      0.26       100

    accuracy                           0.90      1030
   macro avg       0.72      0.58      0.60      1030
weighted avg       0.88      0.90      0.88      1030
```

INPUT

```
print('DISPLAYING TREE PLOT 2')
print('-----------------------')
plt.figure(figsize=(40,20))
plot_tree(dt1,class_names=cn,filled=True)
plt.show()
```

OUTPUT


x[10] <= 248.5
entropy = 0.511
samples = 3089
value = [2738, 351]
class = no

x[16] <= 17.5
entropy = 0.235
samples = 2025
value = [1947, 78]
class = no

x[12] <= 19.5
entropy = 0.822
samples = 1064
value = [791, 273]
class = no

x[12] <= 7.5
entropy = 0.144
samples = 1909
value = [1870, 39]
class = no

x[12] <= 3.5
entropy = 0.921
samples = 116
value = [77, 39]
class = no

x[12] <= 1.5
entropy = 0.623
samples = 58
value = [9, 49]
class = yes

x[0] <= 41.5
entropy = 0.765
samples = 1006
value = [782, 224]
class = no

x[16] <= 11.5
entropy = 0.99
samples = 25
value = [14, 11]
class = no

x[16] <= 3.5
entropy = 0.112
samples = 1884
value = [1856, 28]
class = no

entropy = 0.503
samples = 9
value = [1, 8]
class = yes

x[10] <= 123.5
entropy = 0.868
samples = 107
value = [76, 31]
class = no

entropy = 0.918
samples = 3
value = [2, 1]
class = no

x[0] <= 20.5
entropy = 0.55
samples = 55
value = [7, 48]
class = yes

x[7] <= 0.5
entropy = 0.74
samples = 976
value = [772, 204]
class = no

x[1] <= 4.5
entropy = 0.918
samples = 30
value = [10, 20]
class = yes

entropy = 0.371
samples = 14
value = [13, 1]
class = no

entropy = 0.439
samples = 11
value = [1, 10]
class = yes

entropy = 0.521
samples = 111
value = [98, 13]
class = no

entropy = 0.07
samples = 1773
value = [1758, 15]
class = no

entropy = 0.406
samples = 37
value = [34, 3]
class = no

entropy = 0.971
samples = 70
value = [42, 28]
class = no

entropy = 0.0
samples = 25
value = [0, 25]
class = yes

entropy = 0.784
samples = 30
value = [7, 23]
class = yes

entropy = 0.809
samples = 632
value = [475, 157]
class = no

entropy = 0.575
samples = 344
value = [297, 47]
class = no

entropy = 0.0
samples = 9
value = [0, 9]
class = yes

entropy = 0.998
samples = 21
value = [10, 11]
class = yes