```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data=pd.read_csv("/content/sample_data/bank-additional-full-1 (1) (1).csv",sep=";")
```

```
data.head()
```

	age	job	marital	education	default	housing	loan	contact	month	day_of_week	...	campaign	pdays	previous	poutcome
0	56.0	housemaid	married	basic.4y	no	no	no	telephone	may	mon	...	1.0	999.0	0.0	nonexistent
1	57.0	services	married	high.school	unknown	no	no	telephone	may	mon	...	1.0	999.0	0.0	nonexistent
2	37.0	services	married	high.school	no	yes	no	telephone	may	mon	...	1.0	999.0	0.0	nonexistent
3	40.0	admin.	married	basic.6y	no	no	no	telephone	may	mon	...	1.0	999.0	0.0	nonexistent
4	56.0	services	married	high.school	no	no	yes	telephone	may	mon	...	1.0	999.0	0.0	nonexistent

5 rows × 21 columns

```
data.shape
```

```
(41199, 21)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41199 entries, 0 to 41198
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             41195 non-null  float64
 1   job             41194 non-null  object
 2   marital         41194 non-null  object
 3   education       41194 non-null  object
 4   default         41195 non-null  object
 5   housing         41196 non-null  object
 6   loan            41195 non-null  object
 7   contact         41195 non-null  object
 8   month           41196 non-null  object
 9   day_of_week     41196 non-null  object
 10  duration        41196 non-null  float64
 11  campaign        41196 non-null  float64
 12  pdays           41196 non-null  float64
 13  previous        41196 non-null  float64
 14  poutcome        41194 non-null  object
 15  emp.var.rate    41196 non-null  float64
 16  cons.price.idx  41195 non-null  float64
 17  cons.conf.idx   41196 non-null  float64
 18  euribor3m       41196 non-null  float64
 19  nr.employed     41196 non-null  float64
 20  y               41196 non-null  object
dtypes: float64(10), object(11)
memory usage: 6.6+ MB
```

```
data.isna().sum().sum()
```

```
76
```

```
data.isna().mean()*100
```

	0
age	0.009709
job	0.012136
marital	0.012136
education	0.012136
default	0.009709
housing	0.007282
loan	0.009709
contact	0.009709
month	0.007282
day_of_week	0.007282
duration	0.007282
campaign	0.007282
pdays	0.007282
previous	0.007282
poutcome	0.012136
emp.var.rate	0.007282
cons.price.idx	0.009709
cons.conf.idx	0.007282
euribor3m	0.007282
nr.employed	0.007282
y	0.007282

```
data=data.dropna()
```

```
data.isna().sum()
```

	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
y	0

```
data.describe()
```

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.
count	41190.000000	41190.000000	41190.000000	41190.000000	41190.000000	41190.000000	41190.000000	41190.000000	41190.000000	411
mean	40.026196	258.284074	2.567613	962.477227	0.173003	0.081828	93.575722	-40.503100	3.621165	51
std	10.425734	259.272989	2.769948	186.906543	0.494923	1.570943	0.578886	4.628642	1.734499	7
min	17.000000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	49
25%	32.000000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	50
50%	38.000000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	51
75%	47.000000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	52
max	98.000000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	52

```
data.job.value_counts()
```

job	count
admin.	10422
blue-collar	9254
technician	6743
services	3969
management	2924
retired	1722
entrepreneur	1456
self-employed	1421
housemaid	1060
unemployed	1014
student	875
unknown	330

```
data.y.value_counts()
```

y	count
no	36550
yes	4640

```
data.y=np.where(data['y']=='yes',1,0)
```

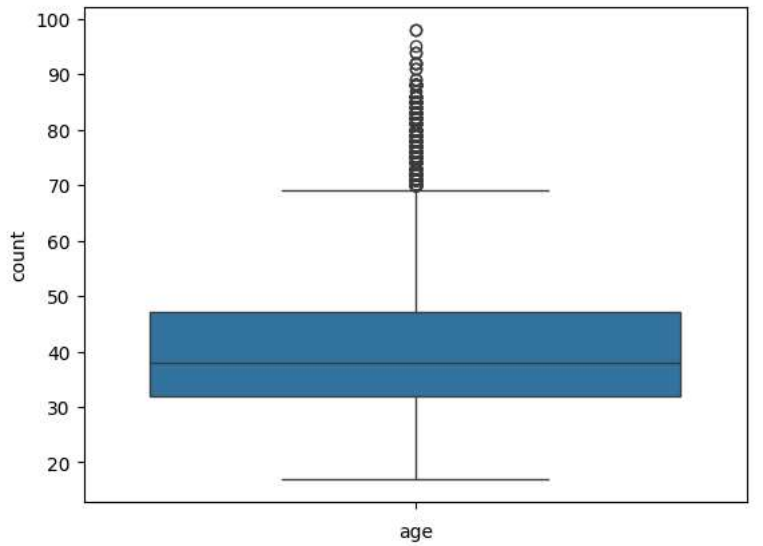
```
data.y.head()
```

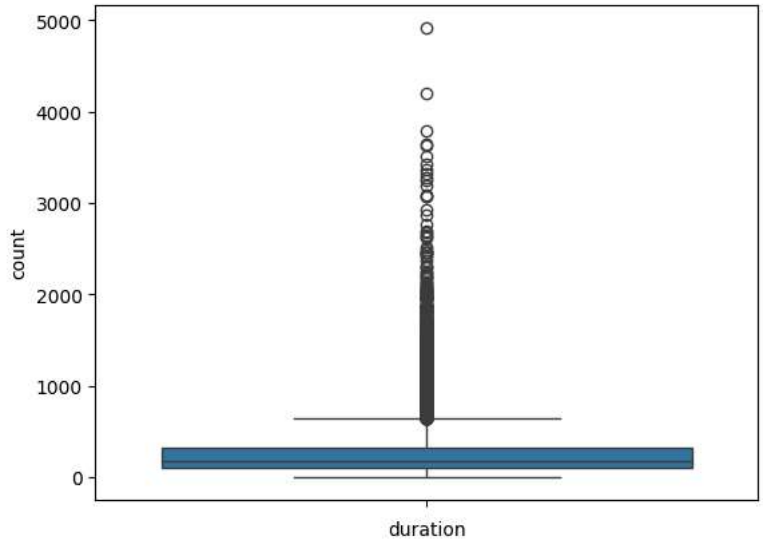
	y
0	0
1	0
2	0
3	0
4	0

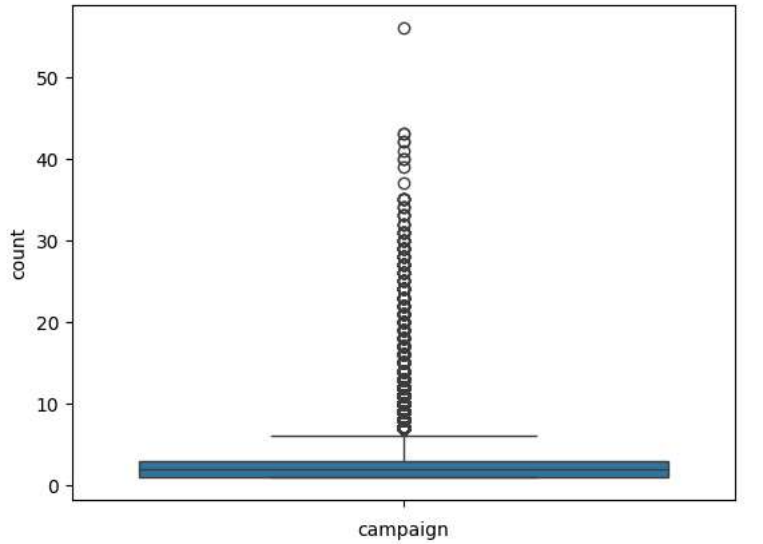
```
data['y']=data['y'].astype('str')
```

```
for i in data.columns:
  if (data[i].dtypes=='int64' or data[i].dtypes=='float64'):
    sns.boxplot(data[i])
    plt.xlabel(i)
    plt.ylabel('count')
    plt.show()
```

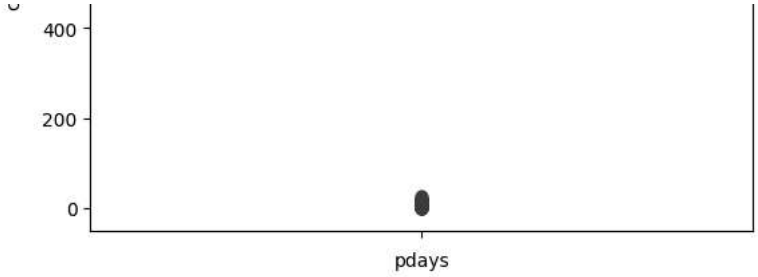
[4]

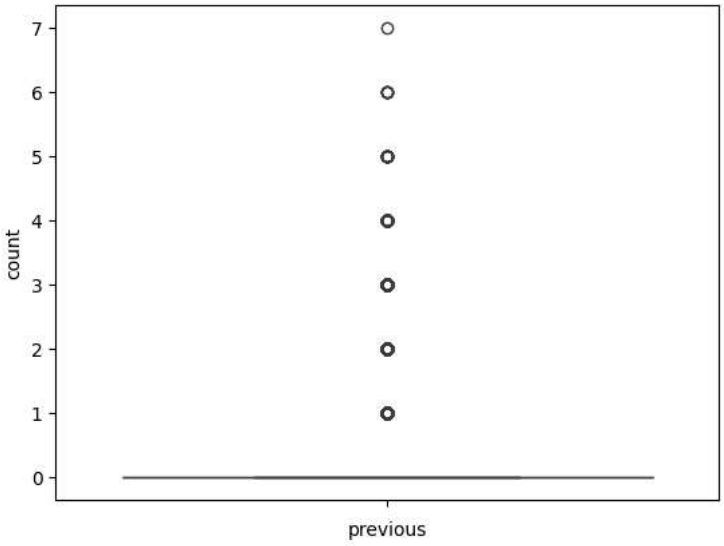

100
90
80
70
60
50
40
30
20
count
age

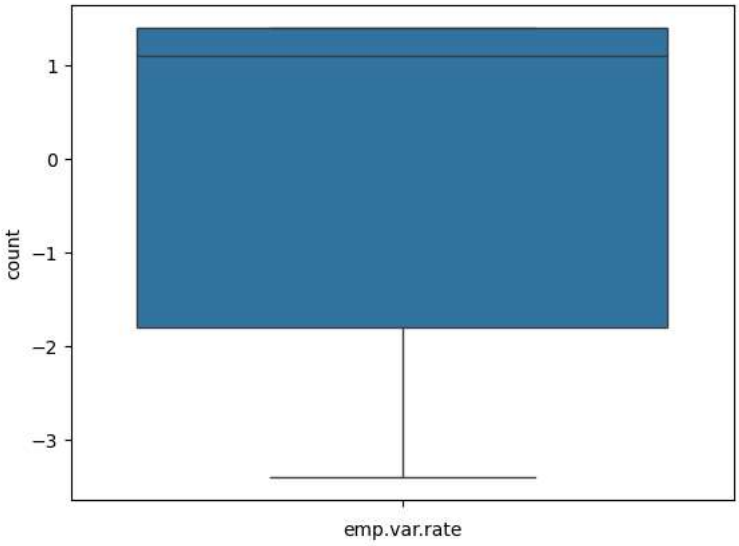

5000
4000
3000
2000
1000
0
count
duration

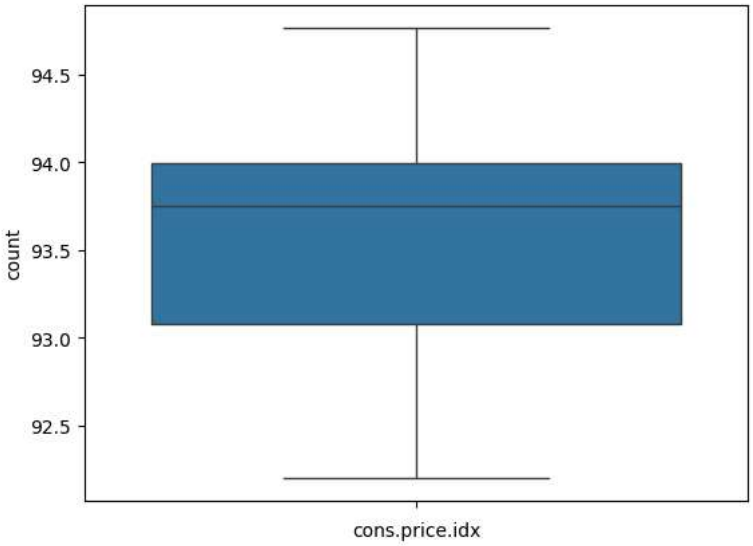

50
40
30
20
10
0
count
campaign

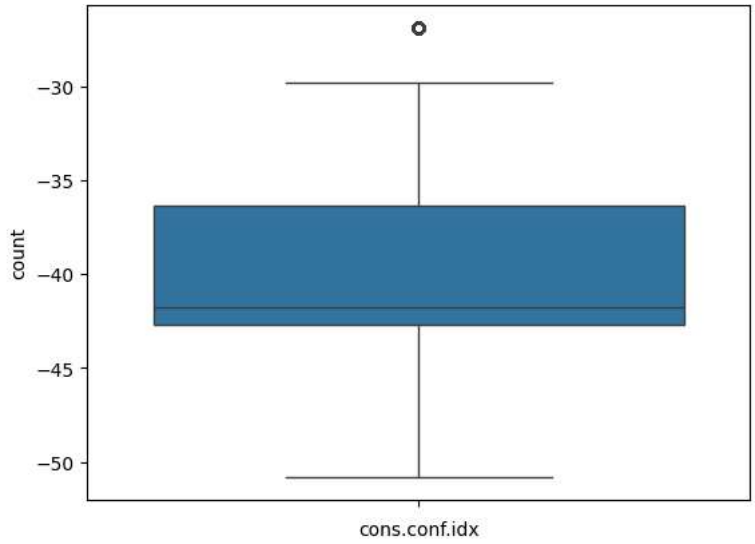

1000
800
600
count

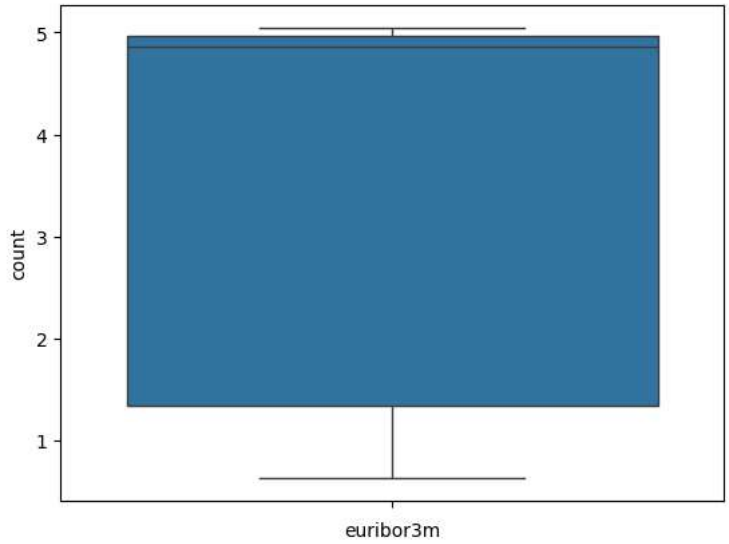

400
200
0
pdays

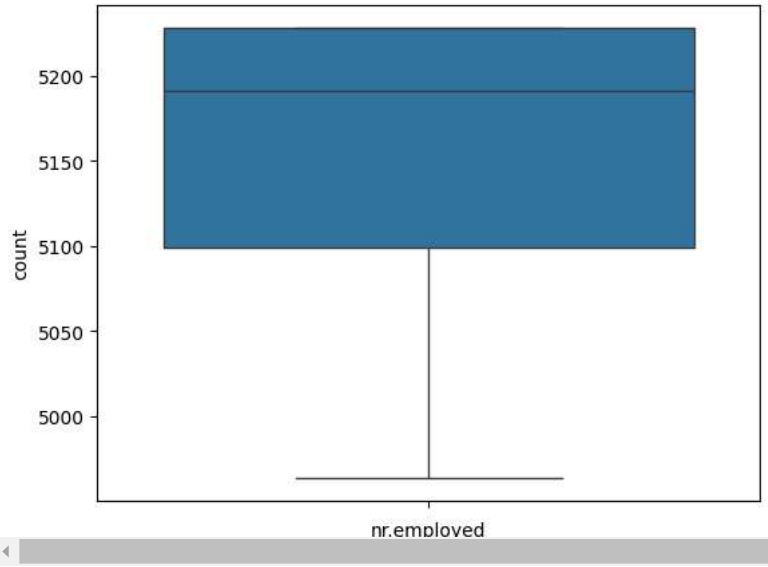

7
6
5
4
3
2
1
0
count
previous


1
0
−1
−2
−3
count
emp.var.rate


94.5
94.0
93.5
93.0
92.5
count
cons.price.idx


−30
−35
−40
−45
−50
count
cons.conf.idx

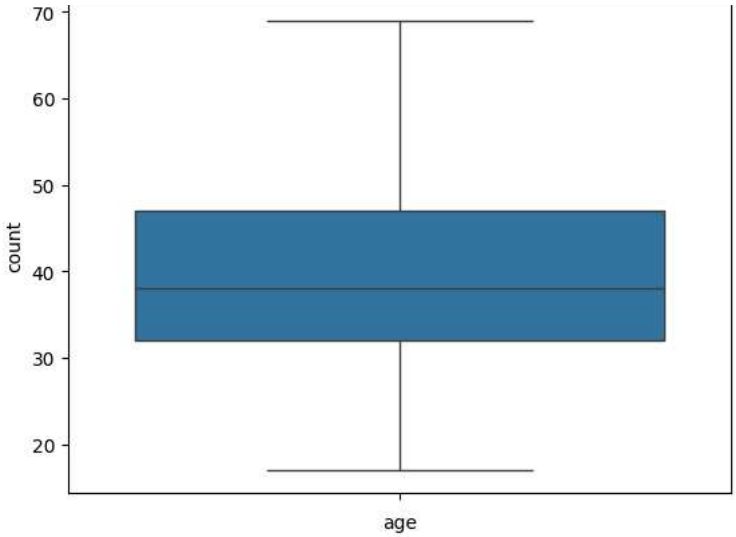

5
4
3
2
1
count
euribor3m

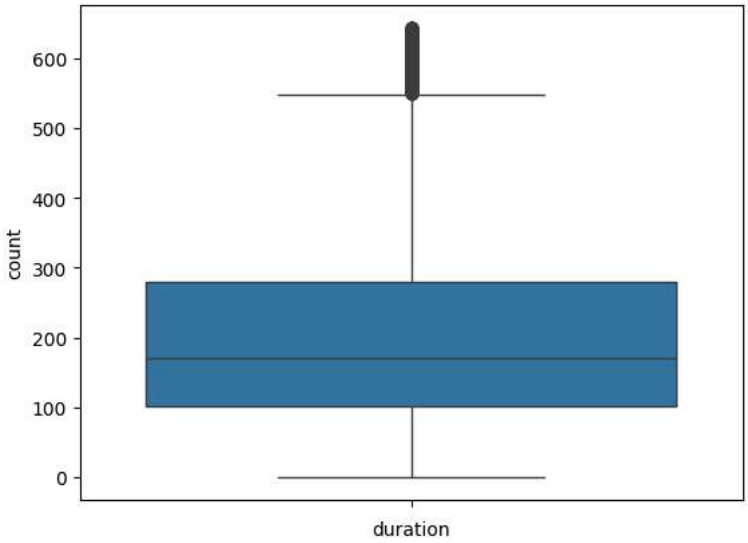

5200
5150
5100
5050
5000
count
nr.employed

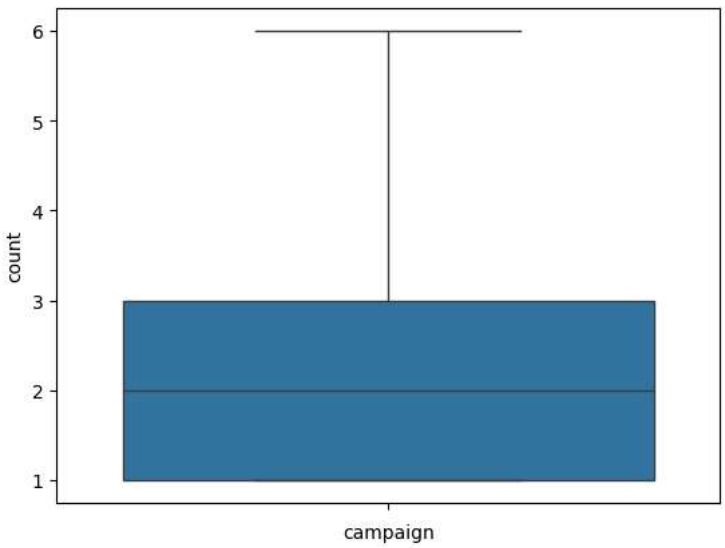
```
q1=data['age'].quantile(0.25)
q3=data['age'].quantile(0.75)
iqr=q3-q1
lower=q1-1.5*iqr
upper=q3+1.5*iqr
data=data[(data['age']>=lower)&(data['age']<=upper)]

q1=data['duration'].quantile(0.25)
q3=data['duration'].quantile(0.75)
iqr=q3-q1
lower=q1-1.5*iqr
upper=q3+1.5*iqr
data=data[(data['duration']>=lower)&(data['duration']<=upper)]


q1=data['campaign'].quantile(0.25)
q3=data['campaign'].quantile(0.75)
iqr=q3-q1
lower=q1-1.5*iqr
upper=q3+1.5*iqr
data=data[(data['campaign']>=lower)&(data['campaign']<=upper)]



q1=data['cons.conf.idx'].quantile(0.25)
q3=data['cons.conf.idx'].quantile(0.75)
iqr=q3-q1
lower=q1-1.5*iqr
upper=q3+1.5*iqr
data=data[(data['cons.conf.idx']>=lower)&(data['cons.conf.idx']<=upper)]
```

```
for i in data.columns:
  if (data[i].dtypes=='int64' or data[i].dtypes=='float64'):
    sns.boxplot(data[i])
    plt.xlabel(i)
    plt.ylabel('count')
    plt.show()
```

[4]


70
60
50
40
30
20
count
age


600
500
400
300
200
100
0
count
duration


6
5
4
3
2
1
count
campaign


1000
800
600
count