```
# IMPORTING THE REQUIRED LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files Default_Fin.csv

- **Default_Fin.csv**(application/vnd.ms-excel) - 275118 bytes, last modified: 7/30/2021 - 100% done

```
Saving Default_Fin.csv to Default_Fin (1).csv
```

```
uploaded.keys()
```

```
dict_keys(['Default_Fin.csv'])
```

```
# importing dataset
default=pd.read_csv("Default_Fin.csv")
default.head()
```

	Index	Employed	Bank Balance	Annual Salary	Defaulted?
0	1	1	8754.36	532339.56	0
1	2	0	9806.16	145273.56	0
2	3	1	12882.60	381205.68	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

```
default.shape
```

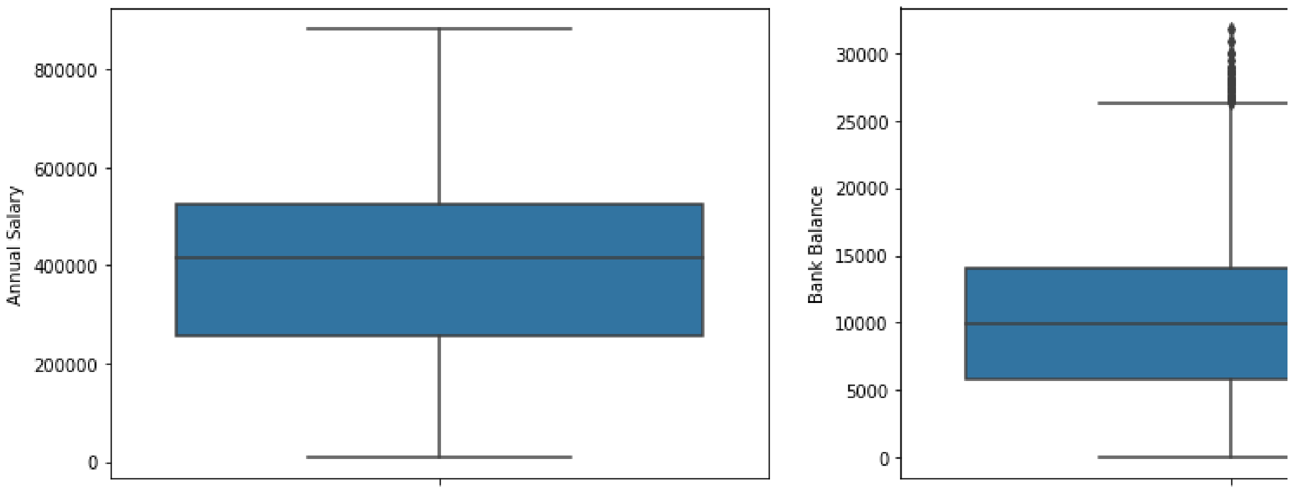
```
(10000, 5)
```

```
default.describe()
```

	Index	Employed	Bank Balance	Annual Salary	Defaulted?
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000
mean	5000.50000	0.705600	10024.498524	402203.782224	0.033300
std	2886.89568	0.455795	5804.579486	160039.674988	0.179428

```
#boxplot
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.boxplot(y=default["Annual Salary"])

plt.subplot(1,2,2)
sns.boxplot(y=default["Bank Balance"])
plt.show()
```



To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

```
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.countplot(default["Employed"])

plt.subplot(1,2,2)
sns.countplot(default["Defaulted?"])
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
```

7000
6000
5000
4000
10000
8000
6000
0
1
0

```
#value count
default["Employed"].value_counts()
```

```
1    7056
0    2944
Name: Employed, dtype: int64
```

```
default["Defaulted?"].value_counts()
```

```
0    9667
1     333
Name: Defaulted?, dtype: int64
```

```
# value count in the form of percentage
default["Employed"].value_counts(normalize=True)
```

```
1    0.7056
0    0.2944
Name: Employed, dtype: float64
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

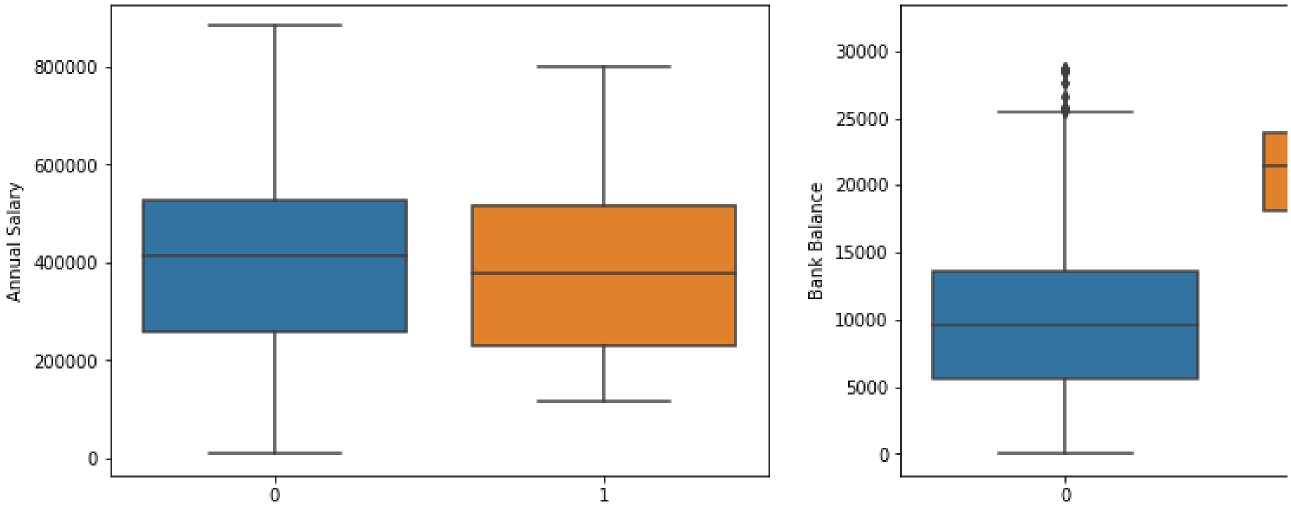
```
0    0.9667
1    0.0333
Name: Defaulted?, dtype: float64
```

bivariate analysis

```
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
sns.boxplot(default["Defaulted?"],default["Annual Salary"])

plt.subplot(1,2,2)
sns.boxplot(default["Defaulted?"],default["Bank Balance"])
plt.show()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pas
  FutureWarning
```

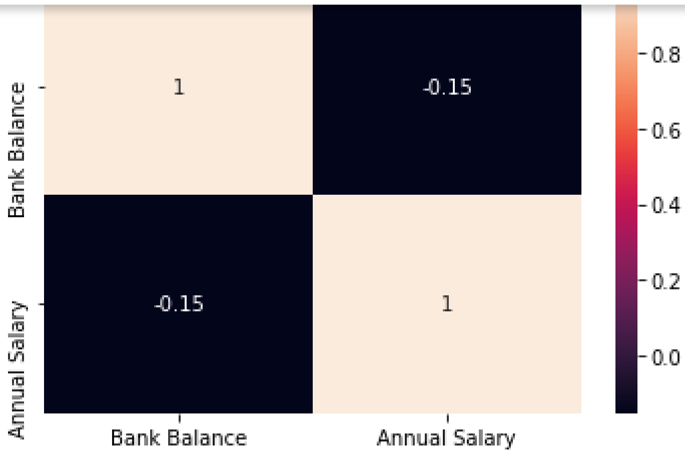


```
# cross tab
pd.crosstab(default["Employed"],default["Defaulted?"],normalize="index").round(2)
```

Defaulted?	0	1
Employed		
0	0.96	0.04
1	0.97	0.03

```
#heatmap to check correlation
sns.heatmap(default[["Bank Balance","Annual Salary"]].corr(),annot=True)
plt.show()
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ✕

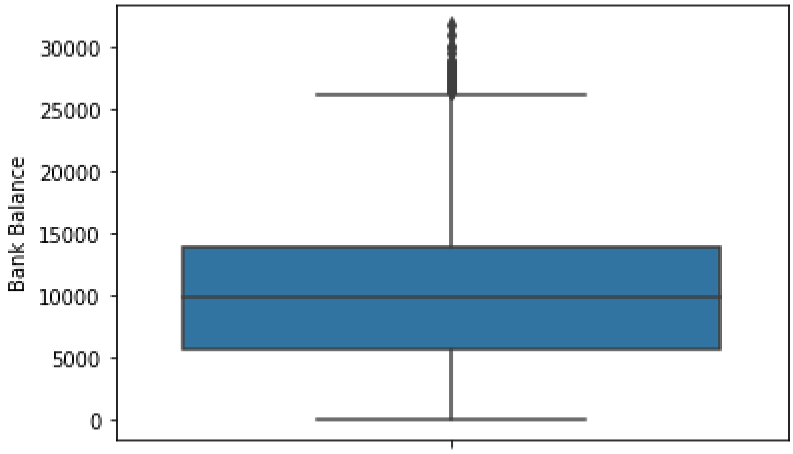


```
# to check null values
default.isnull().sum()
```

```
Index            0
Employed         0
Bank Balance     0
Annual Salary    0
```

```
    Defaulted?       0
    dtype: int64
```

```
sns.boxplot(y=default["Bank Balance"])
plt.show()
```



```
q1,q2=default["Bank Balance"].quantile([0.25,0.75])
```

```
# inter quantile range
iqr=q2-q1
iqr
```

```
    8214.87
```

```
# lower limit
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

```
# upper limit
ul=q2+(1.5*iqr)
print(ul)
```

```
    -6541.515000000001
    26317.965
```

```
# values in database which are greater than upper limit
df=default[default["Bank Balance"]>ul]
df.head()
```

```
df["Bank Balance"].count()
```

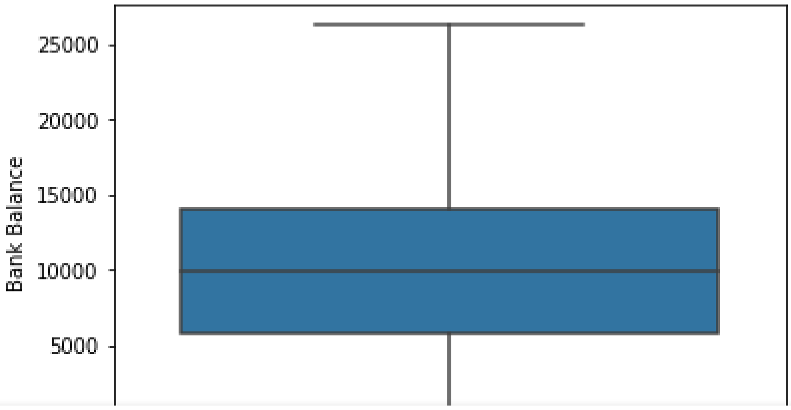
```
31
```

```
#how many defaulters are from outlier
df["Defaulted?"].value_counts()
```

```
1    26
0     5
Name: Defaulted?, dtype: int64
```

```
# setting valus which are greater than upper limit to the value of upper limit
default["Bank Balance"]=np.where(default["Bank Balance"]>ul,ul,default["Bank Balance"])
```

```
sns.boxplot(y=default["Bank Balance"])
plt.show()
```



To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

```
default[default["Bank Balance"]==ul].count()
```

```
Index           31
Employed        31
Bank Balance    31
Annual Salary   31
Defaulted?      31
dtype: int64
```

```
default=default.drop("Index",axis=1)
```

```
default.columns=["employed","balance","income","default"]
default.head()
```

	employed	balance	income	default
0	1	8754.36	532339.56	0
1	0	9806.16	145273.56	0

```
#applying logistic regresion
from sklearn.model_selection import train_test_split
```

```
x=default.drop("default",axis=1)
y=default["default"]
```

```
x.head()
```

	employed	balance	income
0	1	8754.36	532339.56
1	0	9806.16	145273.56
2	1	12882.60	381205.68
3	1	6351.00	428453.88
4	1	9427.92	461562.00

```
0x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=21,stratify
```

```
print(x_train.shape)
print(y_train.shape)
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu ×

```
print(y_train.value_counts(normalize=True).round(2))
print(" ")
print(y_test.value_counts(normalize=True).round(2))
```

```
0    0.97
1    0.03
Name: default, dtype: float64

0    0.97
1    0.03
Name: default, dtype: float64
```

```
!pip install imblearn
```

```
Requirement already satisfied: imblearn in /usr/local/lib/python3.7/dist-packages (0
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: numpy>=1.8.2 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: scikit-learn>=0.20 in /usr/local/lib/python3.7/dist-p
Requirement already satisfied: scipy>=0.13.3 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-package
```

```
# SMOTE (synthetic minority over sampling technique)
import imblearn
from imblearn.over_sampling import SMOTE
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/externals/six.py:31: FutureWarning: T
  "(https://pypi.org/project/six/).", FutureWarning)
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:144: FutureWarni
  warnings.warn(message, FutureWarning)
```

```
sm=SMOTE(random_state=33, sampling_strategy=0.75)
```

```
x_res,y_res=sm.fit_sample(x_train,y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/utils/deprecation.py:87: FutureWarnin
  warnings.warn(msg, category=FutureWarning)
```

```
from sklearn.linear_model import LogisticRegression
```

```
lr=LogisticRegression()
```

```
lr.fit(x_res,y_res)
```

To undo cell deletion use Ctrl+M Z or the Undo option in the Edit menu

```
                                                fit_intercept=True,
                                                x_iter=100,
                                                ty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
y_pred=lr.predict(x_test)
```

```
from sklearn.metrics import confusion_matrix
```

```
confusion_matrix(y_test,y_pred)
```

```
array([[2103,  797],
       [  28,   72]])
```

```
#calculating accuracy
(2103+72)/(2103+797+28+72)
```

```
0.725
```