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

```python
credit_df = pd.read_csv("/content/CreditRisk.csv")
```

```python
credit_df.shape
```

```
(614, 13)
```

```python
credit_df.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	Co
0	LP001002	Male	No	0	Graduate	No	5849	
1	LP001003	Male	Yes	1	Graduate	No	4583	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	
4	LP001008	Male	No	0	Graduate	No	6000	

```python
credit_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 614 entries, 0 to 613
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Loan_ID            614 non-null    object
 1   Gender             601 non-null    object
 2   Married            611 non-null    object
 3   Dependents         599 non-null    object
 4   Education          614 non-null    object
 5   Self_Employed      582 non-null    object
 6   ApplicantIncome    614 non-null    int64
 7   CoapplicantIncome  614 non-null    float64
 8   LoanAmount         614 non-null    int64
 9   Loan_Amount_Term   600 non-null    float64
 10  Credit_History     564 non-null    float64
 11  Property_Area      614 non-null    object
 12  Loan_Status        614 non-null    int64
dtypes: float64(3), int64(3), object(7)
memory usage: 62.5+ KB
```

```
credit_df.tail(20)
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome
594	LP002938	Male	Yes	0	Graduate	Yes	16120
595	LP002940	Male	No	0	Not Graduate	No	3833
596	LP002941	Male	Yes	2	Not Graduate	Yes	6383
597	LP002943	Male	No	NaN	Graduate	No	2987
598	LP002945	Male	Yes	0	Graduate	Yes	9963
599	LP002948	Male	Yes	2	Graduate	No	5780
600	LP002949	Female	No	3+	Graduate	NaN	416
601	LP002950	Male	Yes	0	Not Graduate	NaN	2894
602	LP002953	Male	Yes	3+	Graduate	No	5703
603	LP002958	Male	No	0	Graduate	No	3676
604	LP002959	Female	Yes	1	Graduate	No	12000
605	LP002960	Male	Yes	0	Not Graduate	No	2400
606	LP002961	Male	Yes	1	Graduate	No	3400
607	LP002964	Male	Yes	2	Not Graduate	No	3987
608	LP002974	Male	Yes	0	Graduate	No	3232
609	LP002978	Female	No	0	Graduate	No	2900
610	LP002979	Male	Yes	3+	Graduate	No	4106
611	LP002983	Male	Yes	1	Graduate	No	8072
612	LP002984	Male	Yes	2	Graduate	No	7583
613	LP002990	Female	No	0	Graduate	Yes	4583

```
credit_df.describe()
```

	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
count	614.000000	614.000000	614.000000	600.00000	564.000000
mean	5403.459283	1621.245798	141.166124	342.00000	0.842199
std	6109.041673	2926.248369	88.340630	65.12041	0.364878
min	150.000000	0.000000	0.000000	12.00000	0.000000
25%	2877.500000	0.000000	98.000000	360.00000	1.000000

```
credit_df.Loan_Status.value_counts()
```

```
1    422
0    192
Name: Loan_Status, dtype: int64
```

```
credit_df.iloc[:,12]
```

```
0      1
1      0
2      1
3      1
4      1
      ..
609    1
610    1
611    1
612    1
613    0
Name: Loan_Status, Length: 614, dtype: int64
```

```
credit_df.groupby(['Education','Loan_Status']).Education.count()
```

```
Education      Loan_Status
Graduate       0              140
               1              340
Not Graduate   0               52
               1               82
Name: Education, dtype: int64
```
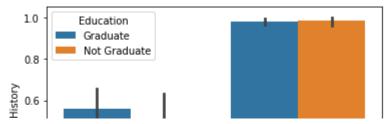
```
sns.barplot(y='Credit_History', x='Loan_Status', hue='Education', data=credit_df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8625a34c10>
```


▾ Fill Null Values

```
100 * credit_df.isnull().sum() / credit_df.shape[0]
```

```
Loan_ID               0.000000
Gender                2.117264
Married               0.488599
Dependents            2.442997
Education             0.000000
Self_Employed         5.211726
ApplicantIncome       0.000000
CoapplicantIncome     0.000000
LoanAmount            0.000000
Loan_Amount_Term      2.280130
Credit_History        8.143322
Property_Area         0.000000
Loan_Status           0.000000
dtype: float64
```

```
object_columns = credit_df.select_dtypes(include=['object']).columns
numeric_columns = credit_df.select_dtypes(exclude=['object']).columns
```