```
from google.colab import files
uploaded = files.upload()
import pandas as pd
df = pd.read_csv("Train.csv")
df = df.dropna(subset=["Gender","Ever_Married","Age","Graduated","Profession","Spending_Score","Family_Size","Var_1","Segmen
print(df.shape)
df.head()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Train.csv to Train (3).csv
(7376, 11)
```

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Var_1	Segmentat
0	462809	Male	No	22	No	Healthcare	1.0	Low	4.0	Cat_4	
1	462643	Female	Yes	38	Yes	Engineer	NaN	Average	3.0	Cat_4	
2	466315	Female	Yes	67	Yes	Engineer	1.0	Low	1.0	Cat_6	
3	461735	Male	Yes	67	Yes	Lawyer	0.0	High	2.0	Cat_6	
4	462669	Female	Yes	40	Yes	Entertainment	NaN	High	6.0	Cat_6	

```
from sklearn.preprocessing import LabelEncoder

categorical_cols = ["Gender","Ever_Married","Graduated","Profession","Spending_Score","Var_1","Segmentation"]

le = LabelEncoder()
for col in categorical_cols:
    df[col] = le.fit_transform(df[col].astype(str))

df.head()
```

	ID	Gender	Ever_Married	Age	Graduated	Profession	Work_Experience	Spending_Score	Family_Size	Var_1	Segmentatic
0	462809	1	0	22	0	5	1.0	2	4.0	3	
1	462643	0	1	38	1	2	NaN	0	3.0	3	
2	466315	0	1	67	1	2	1.0	2	1.0	5	
3	461735	1	1	67	1	7	0.0	1	2.0	5	
4	462669	0	1	40	1	3	NaN	1	6.0	5	

```
df["Recency"] = df["Age"]
df["Frequency"] = df["Family_Size"]
df["Monetary"] = df["Spending_Score"].map({"Low":1,"Average":2,"High":3})
df[["Recency","Frequency","Monetary"]].head()
```

	Recency	Frequency	Monetary
0	22	4.0	NaN
1	38	3.0	NaN
2	67	1.0	NaN
3	67	2.0	NaN
4	40	6.0	NaN

```
score_map = {"Low":1, "Average":2, "High":3}

df["Monetary"] = df["Spending_Score"].map(score_map)
print(df["Spending_Score"].unique())

df["Monetary"] = df["Spending_Score"]
print(df[["Recency","Frequency","Monetary"]].head(10))
```

```
[2 0 1]
   Recency  Frequency  Monetary
0       22        4.0         2
1       38        3.0         0
2       67        1.0         2
3       67        2.0         1
4       40        6.0         1
```

```
5       56        2.0         0
6       32        3.0         2
7       33        3.0         2
8       61        3.0         2
9       55        4.0         0
```

```
Start coding or generate with AI.
```

```
   Recency_norm  Frequency_norm  Monetary_norm
0      0.056338           0.375            1.0
1      0.281690           0.250            0.0
2      0.690141           0.000            1.0
3      0.690141           0.125            0.5
4      0.309859           0.625            0.5
5      0.535211           0.125            0.0
6      0.197183           0.250            1.0
7      0.211268           0.250            1.0
8      0.605634           0.250            1.0
9      0.521127           0.375            0.0
```

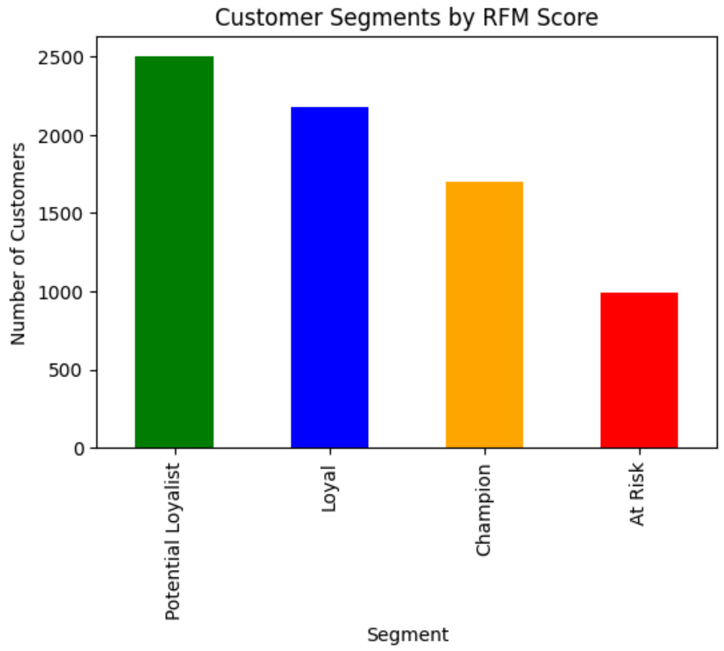
```
df["R_score"] = pd.qcut(df["Recency"], 4, labels=[4,3,2,1])
df["F_score"] = pd.qcut(df["Frequency"], 4, labels=[1,2,3,4])
df["M_score"] = pd.cut(df["Monetary"], bins=3, labels=[1,2,3])
df["R_score"] = pd.qcut(df["Recency"], 4, labels=[4,3,2,1])
df["F_score"] = pd.qcut(df["Frequency"], 4, labels=[1,2,3,4])
df["RFM_Score"] = df["R_score"].astype(int) + df["F_score"].astype(int) + df["M_score"].astype(int)

print(df[["Recency","Frequency","Monetary","R_score","F_score","M_score","RFM_Score"]].head(10))
```

```
   Recency  Frequency  Monetary R_score F_score M_score  RFM_Score
0       22        4.0         2       4       3       3         10
1       38        3.0         0       3       2       1          6
2       67        1.0         2       1       1       3          5
3       67        2.0         1       1       1       2          4
4       40        6.0         1       3       4       2          9
5       56        2.0         0       1       1       1          3
6       32        3.0         2       3       2       3          8
7       33        3.0         2       3       2       3          8
8       61        3.0         2       1       2       3          6
9       55        4.0         0       1       3       1          5
```

```
def segment_customer(score):
    if score >= 9:
        return "Champion"
    elif score >= 7:
        return "Loyal"
    elif score >= 5:
        return "Potential Loyalist"
    else:
        return "At Risk"

df["Segment"] = df["RFM_Score"].apply(segment_customer)

print(df[["Recency","Frequency","Monetary","RFM_Score","Segment"]].head(10))
```

```
   Recency  Frequency  Monetary  RFM_Score             Segment
0       22        4.0         2         10            Champion
1       38        3.0         0          6  Potential Loyalist
2       67        1.0         2          5  Potential Loyalist
3       67        2.0         1          4             At Risk
4       40        6.0         1          9            Champion
5       56        2.0         0          3             At Risk
6       32        3.0         2          8               Loyal
7       33        3.0         2          8               Loyal
8       61        3.0         2          6  Potential Loyalist
9       55        4.0         0          5  Potential Loyalist
```

```
import matplotlib.pyplot as plt
segment_counts = df["Segment"].value_counts()

plt.figure(figsize=(6,4))
segment_counts.plot(kind="bar", color=["green","blue","orange","red"])
plt.title("Customer Segments by RFM Score")
plt.xlabel("Segment")
plt.ylabel("Number of Customers")
plt.show()
```



```
df.to_csv("customer_segmentation.csv", index=False)

print("Segmentation table exported successfully!")
from google.colab import files
files.download("customer_segmentation.csv")
```

```
Segmentation table exported successfully!
```