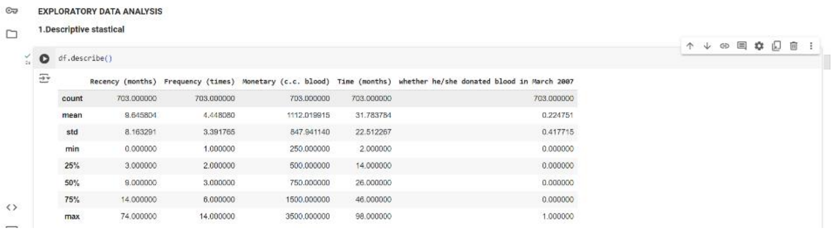
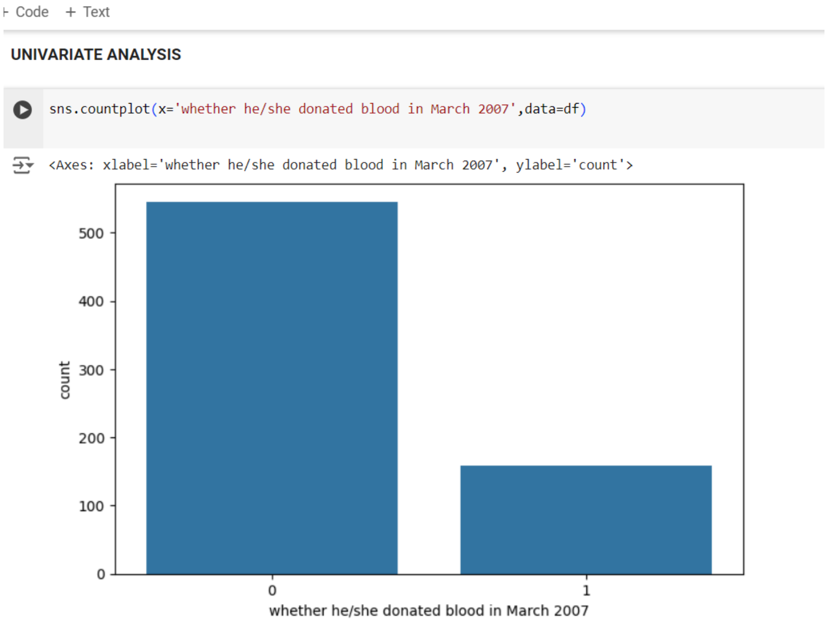
Data Collection and Preprocessing Phase

Date	14 July 2024
Team ID	739872
Project Title	Blood Donation Prediction
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	**Description**
Data Overview	
Univariate Analysis	

EXPLORATORY DATA ANALYSIS

1.Descriptive stastical

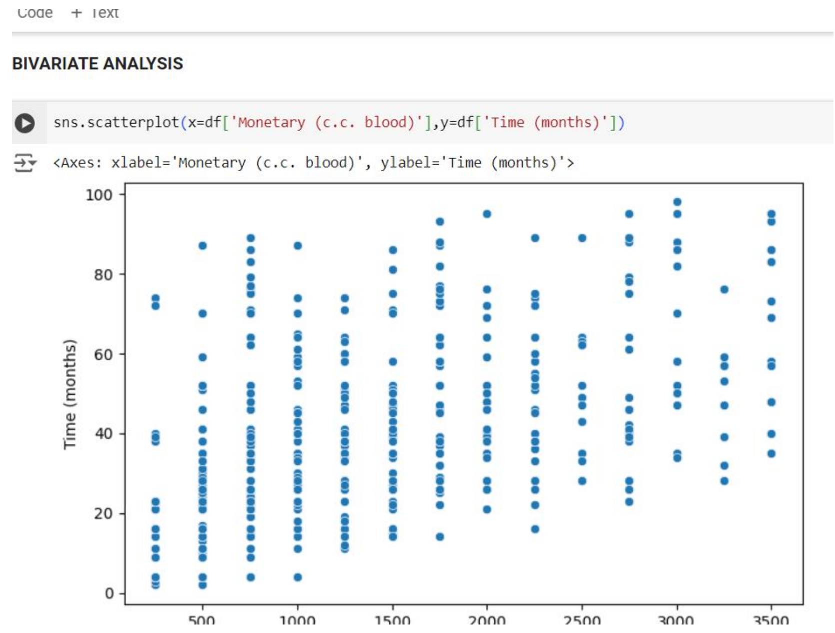
```
df.describe()
```

	Recency (months)	Frequency (times)	Monetary (c.c. blood)	Time (months)	whether he/she donated blood in March 2007
count	703.000000	703.000000	703.000000	703.000000	703.000000
mean	9.645804	4.448080	1112.019915	31.783784	0.224751
std	8.163291	3.391765	847.941140	22.512267	0.417715
min	0.000000	1.000000	250.000000	2.000000	0.000000
25%	3.000000	2.000000	500.000000	14.000000	0.000000
50%	9.000000	3.000000	750.000000	26.000000	0.000000
75%	14.000000	6.000000	1500.000000	46.000000	0.000000
max	74.000000	14.000000	3500.000000	98.000000	1.000000

UNIVARIATE ANALYSIS

```
sns.countplot(x='whether he/she donated blood in March 2007',data=df)
```

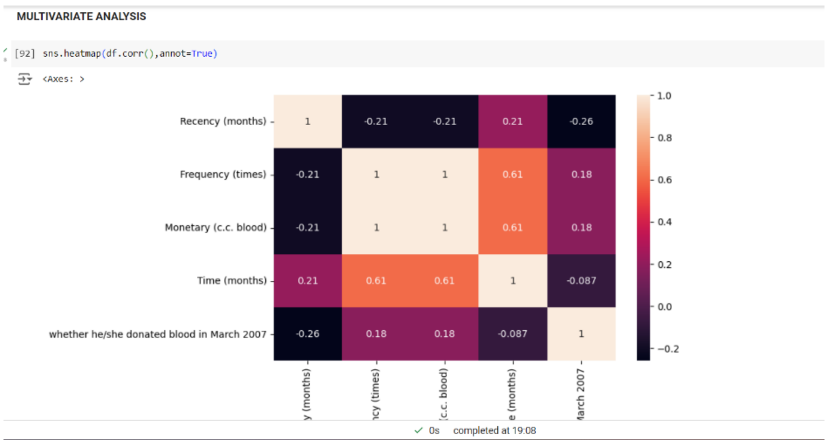
```
<Axes: xlabel='whether he/she donated blood in March 2007', ylabel='count'>
```

<table>
<tr>
<td>Bivariate Analysis</td>
<td>


BIVARIATE ANALYSIS

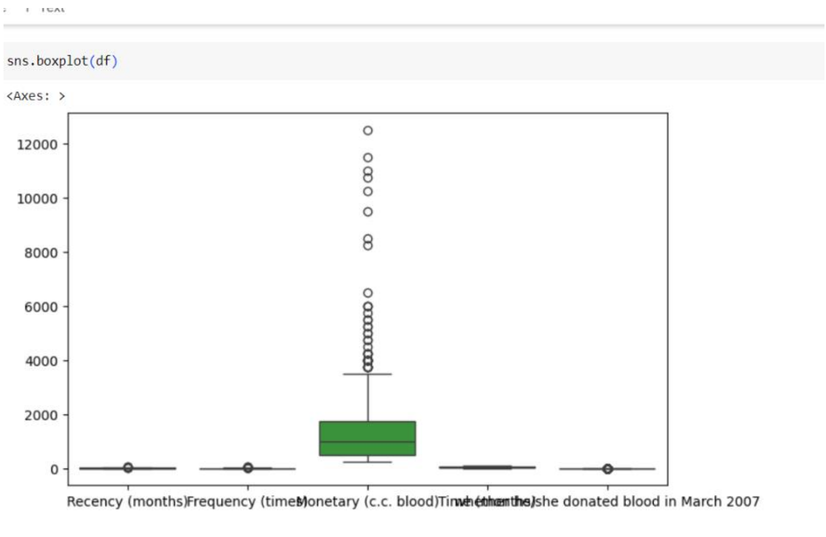
```
sns.scatterplot(x=df['Monetary (c.c. blood)'],y=df['Time (months)'])
```

```
<Axes: xlabel='Monetary (c.c. blood)', ylabel='Time (months)'>
```

</td>
</tr>
<tr>
<td>Multivariate Analysis</td>
<td>

MULTIVARIATE ANALYSIS

```
[92] sns.heatmap(df.corr(),annot=True)
```

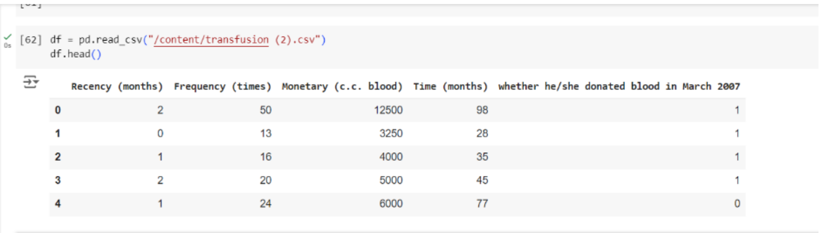
```
<Axes: >
```

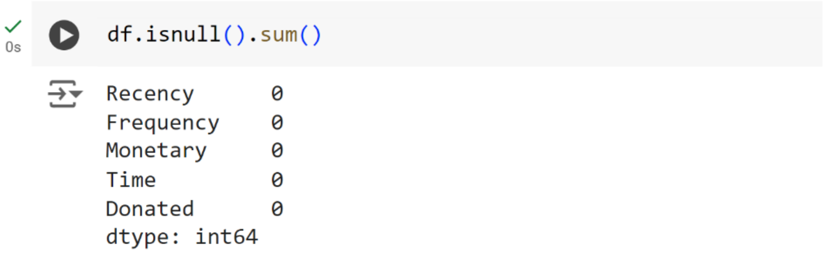
✓ 0s completed at 19:08
</td>
</tr>
<tr>
<td>Outliers and Anomalies</td>
<td>


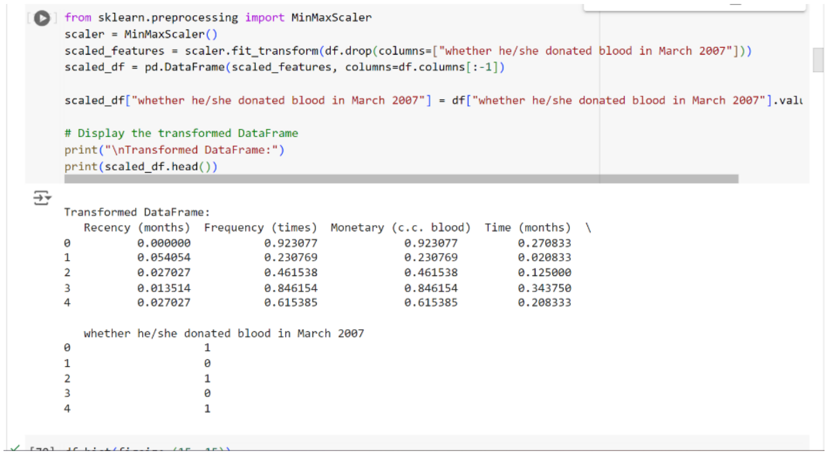
```
sns.boxplot(df)
```

```
<Axes: >
```

</td>
</tr>
</table>

<table>
<tr><th colspan="2">Data Preprocessing Code Screenshots</th></tr>
<tr><td>Loading Data</td><td>
<pre>
[62] df = pd.read_csv("/content/transfusion (2).csv")
     df.head()

       Recency (months) Frequency (times) Monetary (c.c. blood) Time (months) whether he/she donated blood in March 2007
    0   2   50   12500   98   1
    1   0   13   3250   28   1
    2   1   16   4000   35   1
    3   2   20   5000   45   1
    4   1   24   6000   77   0
</pre></td></tr>
<tr><td>Handling Missing Data</td><td>
<pre>
0s
df.isnull().sum()

Recency   0
Frequency   0
Monetary   0
Time   0
Donated   0
dtype: int64
</pre></td></tr>
<tr><td>Data Transformation</td><td>
<pre>
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaled_features = scaler.fit_transform(df.drop(columns=["whether he/she donated blood in March 2007"]))
scaled_df = pd.DataFrame(scaled_features, columns=df.columns[:-1])

scaled_df["whether he/she donated blood in March 2007"] = df["whether he/she donated blood in March 2007"].valu

# Display the transformed DataFrame
print("\nTransformed DataFrame:")
print(scaled_df.head())

Transformed DataFrame:
   Recency (months) Frequency (times) Monetary (c.c. blood) Time (months) \
0   0.000000   0.923077   0.923077   0.270833
1   0.054054   0.230769   0.230769   0.020833
2   0.027027   0.461538   0.461538   0.125000
3   0.013514   0.846154   0.846154   0.343750
4   0.027027   0.615385   0.615385   0.208333

   whether he/she donated blood in March 2007
0   1
1   0
2   1
3   0
4   1
</pre></td></tr>
<tr><td>Feature Engineering</td><td>Attached the codes in final submission.</td></tr>
<tr><td>Save Processed Data</td><td>-</td></tr>
</table>