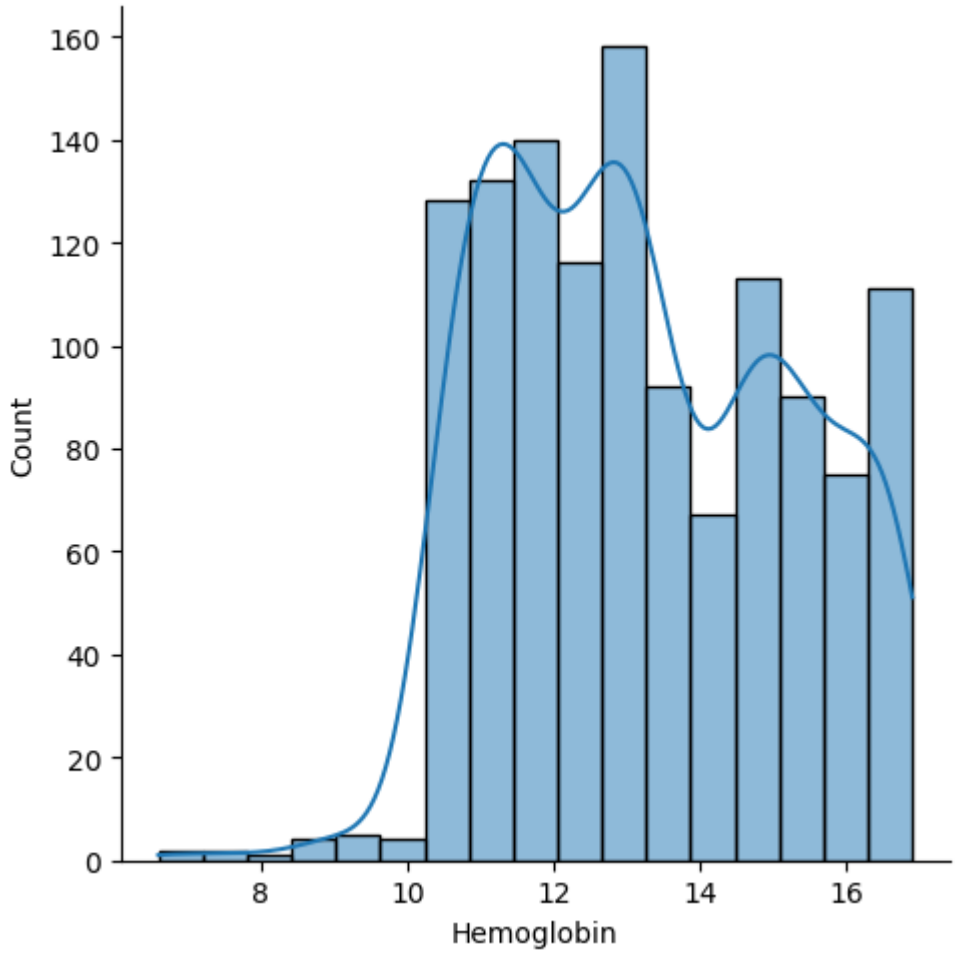
Data Collection and Preprocessing Phase

Data Exploration and Preprocessing report

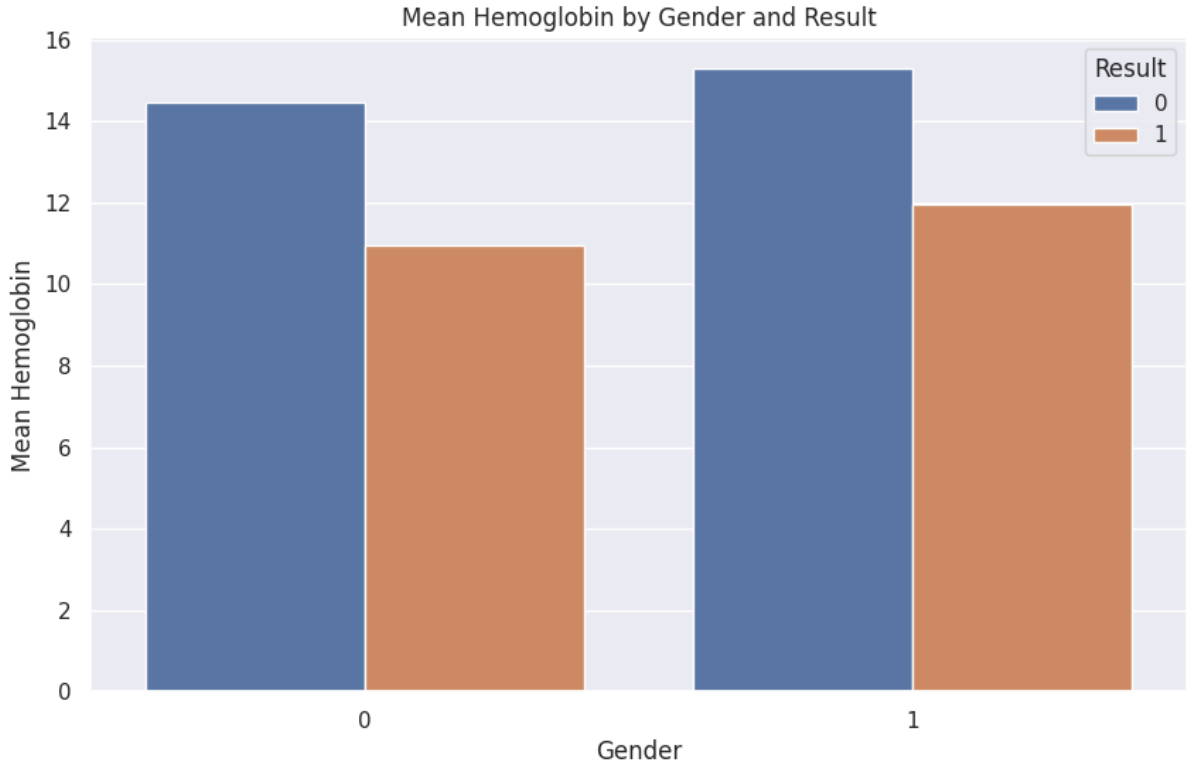
Data Overview-

	Gender	Hemoglobin	MCH	MCHC	MCV	Result
count	1421.000000	1421.000000	1421.000000	1421.000000	1421.000000	1421.000000
mean	0.520760	13.412738	22.905630	30.251232	85.523786	0.436312
std	0.499745	1.974546	3.969375	1.400898	9.636701	0.496102
min	0.000000	6.600000	16.000000	27.800000	69.400000	0.000000
25%	0.000000	11.700000	19.400000	29.000000	77.300000	0.000000
50%	1.000000	13.200000	22.700000	30.400000	85.300000	0.000000
75%	1.000000	15.000000	26.200000	31.400000	94.200000	1.000000
max	1.000000	16.900000	30.000000	32.500000	101.600000	1.000000

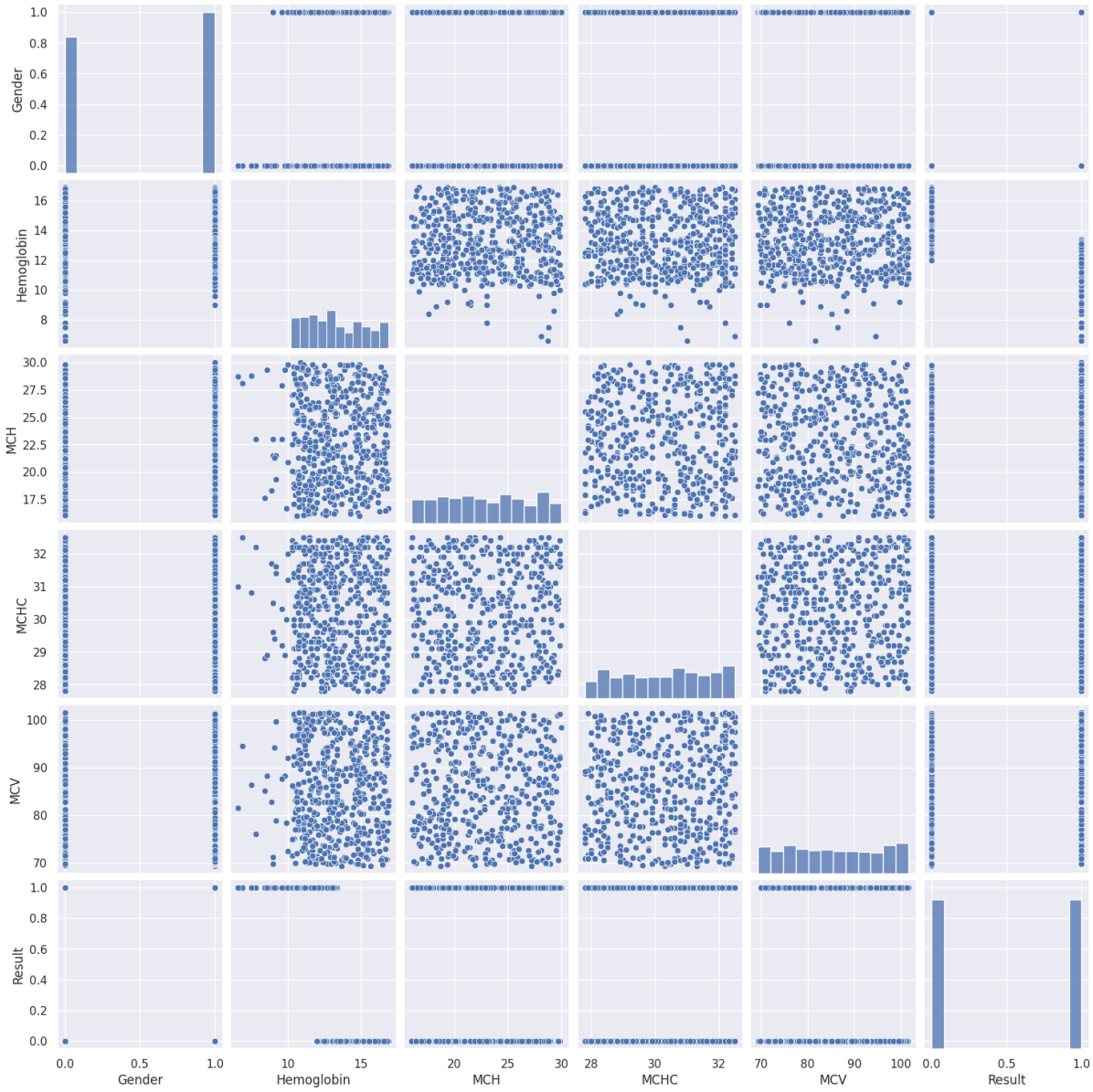
Univarient Analysis-



Bivarient Analysis-



Multivarient Analysis-



Data Preprocessing Code Screenshots

Loding data-

```
df=pd.read_csv('anemia.csv')
df.head()
```

	Gender	Hemoglobin	MCH	MCHC	MCV	Result
0	1	14.9	22.7	29.1	83.7	0
1	0	15.9	25.4	28.3	72.0	0
2	0	9.0	21.5	29.6	71.2	1
3	0	14.9	16.0	31.4	87.5	0
4	1	14.7	22.0	28.2	99.5	0

Haldling Missing Data-

```
df.info()
```
✓ 0.0s

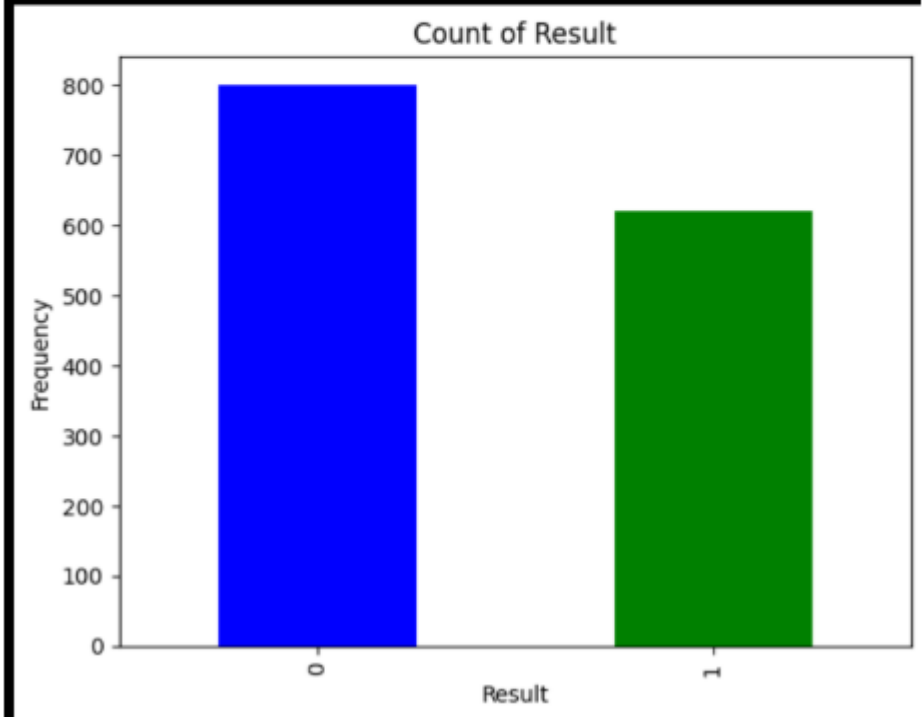
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1421 entries, 0 to 1420
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   Gender      1421 non-null   int64  
 1   Hemoglobin  1421 non-null   float64
 2   MCH         1421 non-null   float64
 3   MCHC        1421 non-null   float64
 4   MCV         1421 non-null   float64
 5   Result      1421 non-null   int64  
dtypes: float64(4), int64(2)
memory usage: 66.7 KB
```

Handling Imbalance Value-

```python
#0-not anemic 1-anemic
#checking for the count of anemia and not anemia

results = df['Result'].value_counts()
results.plot(kind = 'bar', color=['blue', 'green'])
plt.xlabel('Result')
plt.ylabel('Frequency')
plt.title('Count of Result')
plt.show()
```

✓ 0.6s




```python
#we can see that the female count is more than the male so,
# we can balance it using the undersampling

from sklearn.utils import resample
majorclass = df[df['Result'] == 0]
minorclass = df[df['Result'] == 1]

major_downsample = resample(majorclass, replace=False, n_samples=len(minorclass)
                            ,random_state=42)

df = pd.concat([major_downsample, minorclass])

print(df['Result'].value_counts())
```

✓ 0.3s Python

```
Result
0    620
1    620
Name: count, dtype: int64
```

```python
# Plot the balanced gender counts
result_balanced = df['Result'].value_counts()
result_balanced.plot(kind='bar', color=['blue', 'green'])
plt.xlabel('Result')
plt.ylabel('Frequency')
plt.title('Count of Result (Balanced)')
plt.show()
```

✓ 0.4s