Data Collection and Preprocessing Phase

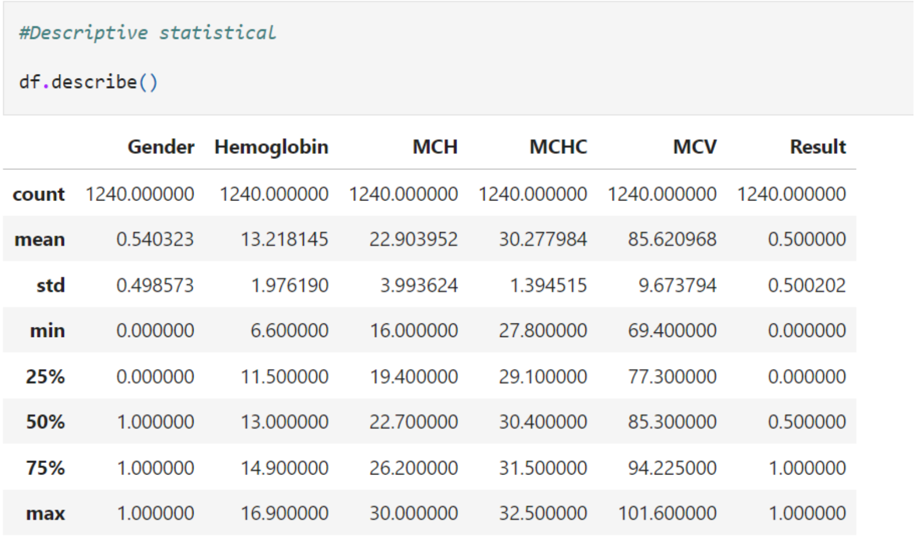
Date	4 June 2024
Team ID	SWTID1720076203
Project Title	Anemia Sense: Leveraging Machine Learning for Precise Anemia Recognitions
Maximum Marks	6 Marks

Data Exploration and Preprocessing

Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions

<table>
<tr><th>Section</th><th>Description</th></tr>
<tr><td>Data Overview</td><td>

Dimension:
1421 rows x 6 columns

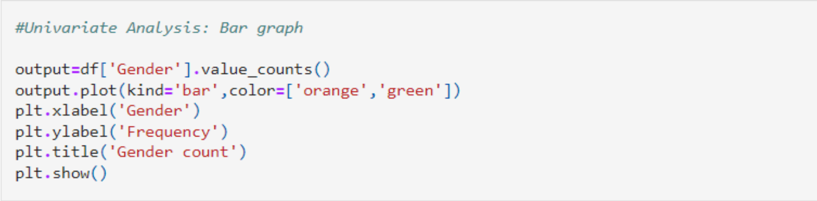


```
#Descriptive statistical

df.describe()
```

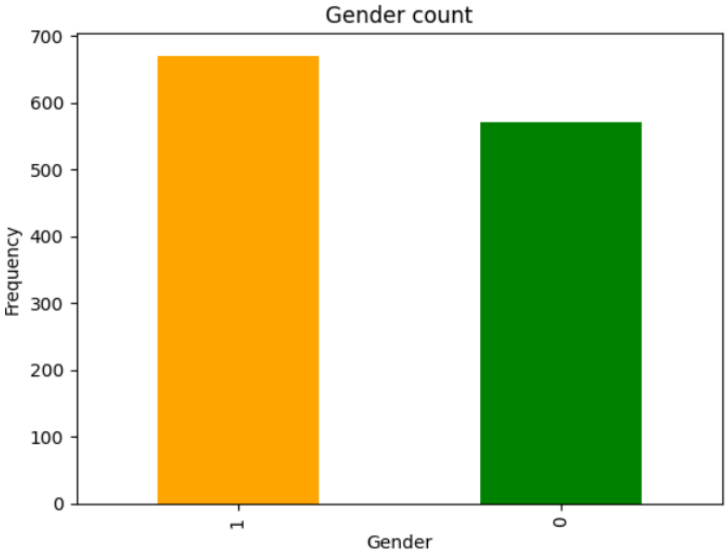
```
            Gender   Hemoglobin          MCH         MCHC          MCV       Result
count  1240.000000  1240.000000  1240.000000  1240.000000  1240.000000  1240.000000
 mean     0.540323    13.218145    22.903952    30.277984    85.620968     0.500000
  std     0.498573     1.976190     3.993624     1.394515     9.673794     0.500202
  min     0.000000     6.600000    16.000000    27.800000    69.400000     0.000000
  25%     0.000000    11.500000    19.400000    29.100000    77.300000     0.000000
  50%     1.000000    13.000000    22.700000    30.400000    85.300000     0.500000
  75%     1.000000    14.900000    26.200000    31.500000    94.225000     1.000000
  max     1.000000    16.900000    30.000000    32.500000   101.600000     1.000000
```

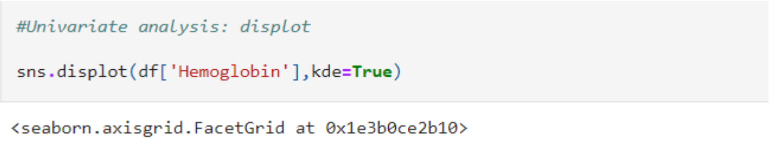
</td></tr>
</table>

<table>
<tr>
<td>Univariate Analysis</td>
<td>



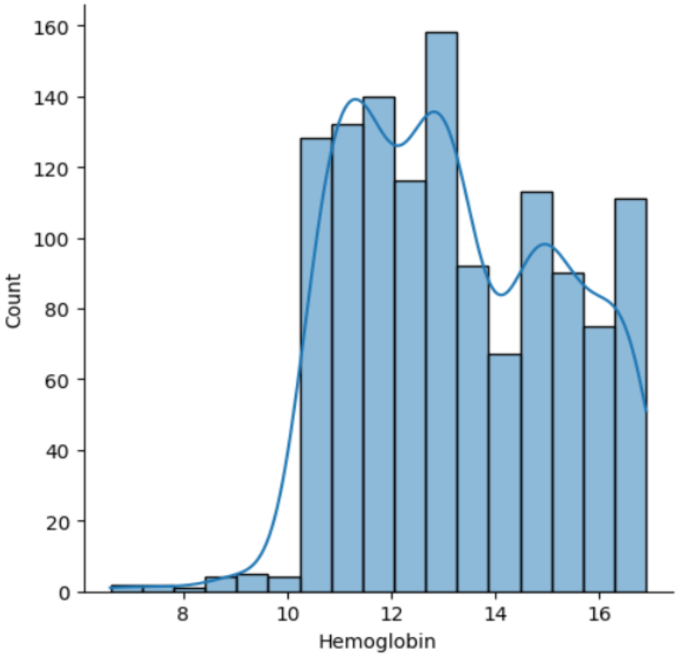
```
#Univariate Analysis: Bar graph

output=df['Gender'].value_counts()
output.plot(kind='bar',color=['orange','green'])
plt.xlabel('Gender')
plt.ylabel('Frequency')
plt.title('Gender count')
plt.show()
```





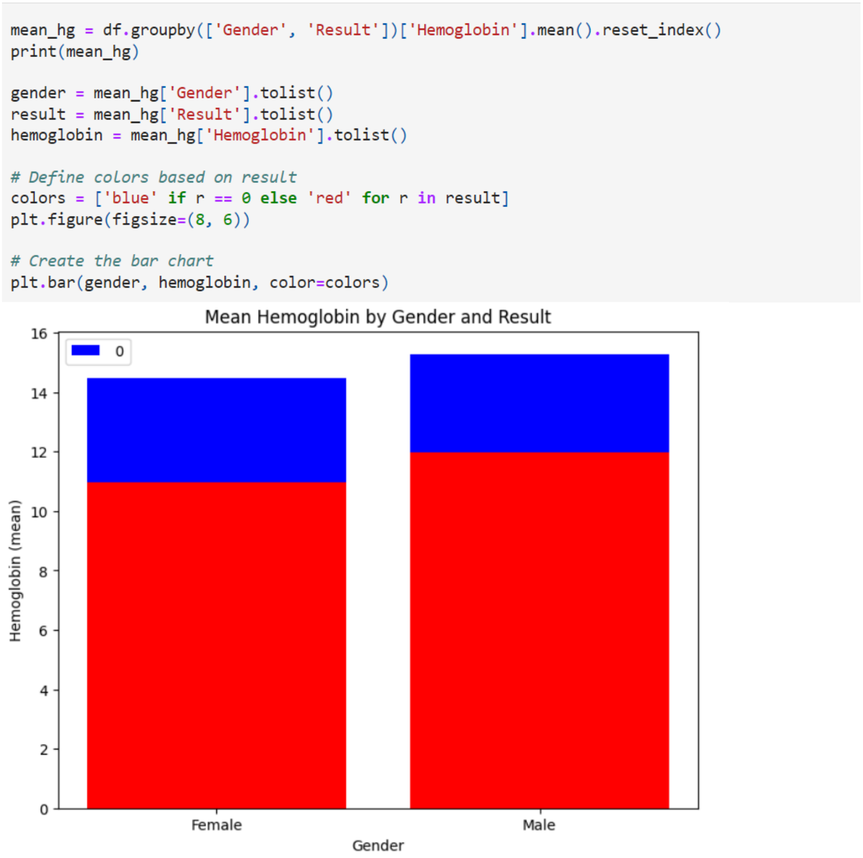
```
#Univariate analysis: displot

sns.displot(df['Hemoglobin'],kde=True)
```

```
<seaborn.axisgrid.FacetGrid at 0x1e3b0ce2b10>
```



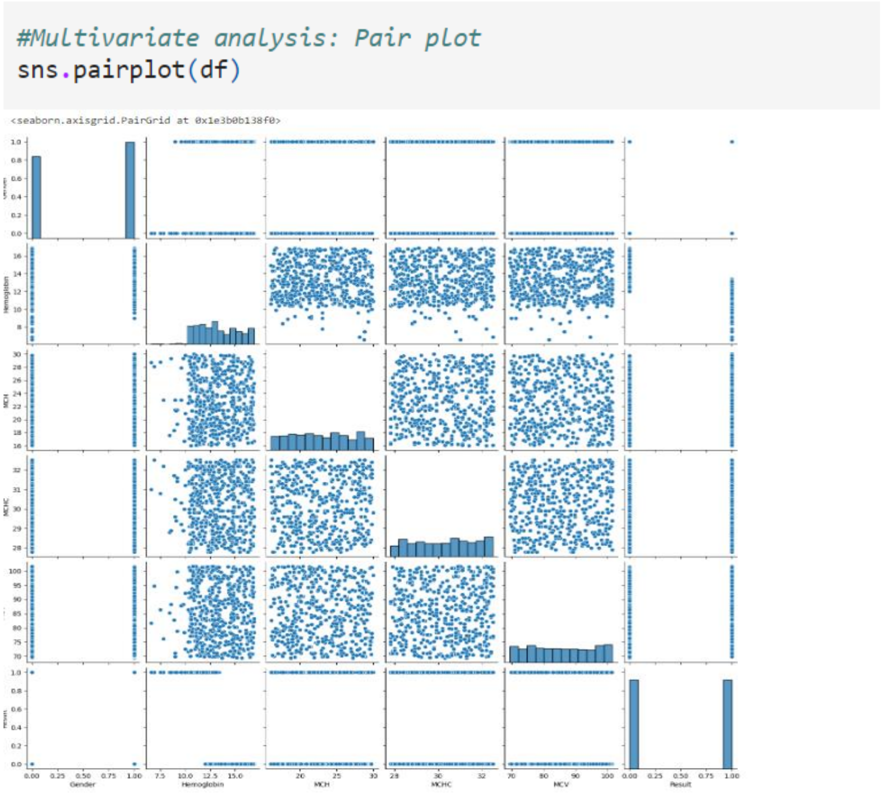
</td>
</tr>
</table>

Bivariate Analysis



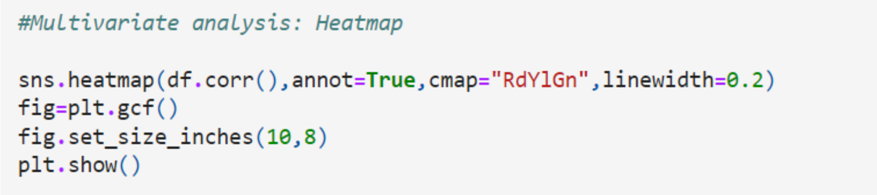
```
mean_hg = df.groupby(['Gender', 'Result'])['Hemoglobin'].mean().reset_index()
print(mean_hg)

gender = mean_hg['Gender'].tolist()
result = mean_hg['Result'].tolist()
hemoglobin = mean_hg['Hemoglobin'].tolist()

# Define colors based on result
colors = ['blue' if r == 0 else 'red' for r in result]
plt.figure(figsize=(8, 6))

# Create the bar chart
plt.bar(gender, hemoglobin, color=colors)
```

Multivariate Analysis

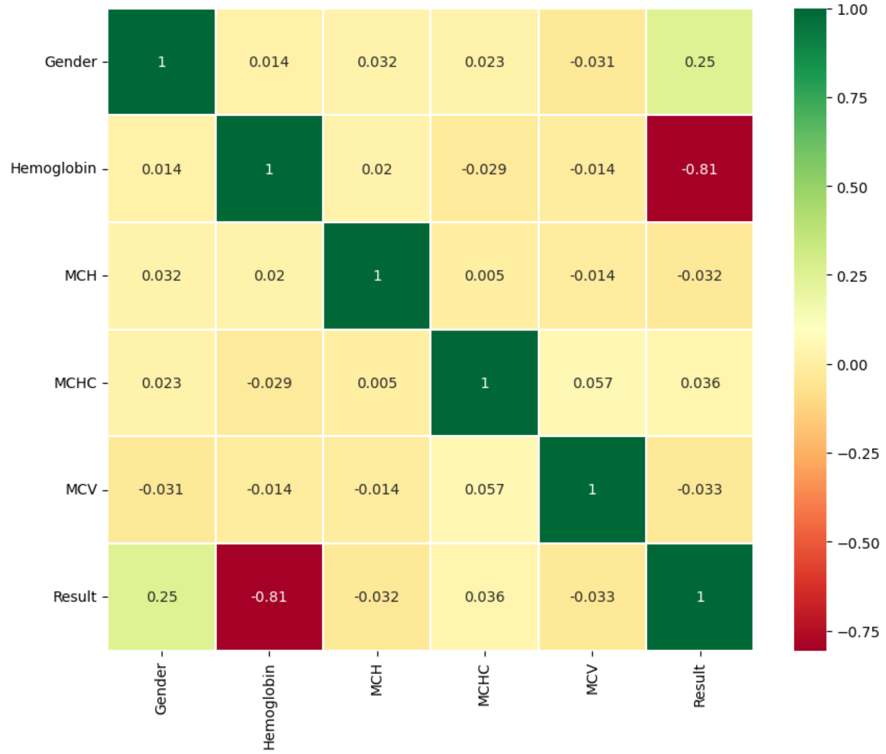


```
#Multivariate analysis: Pair plot
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x1e3b0b138f0>
```



```
#Multivariate analysis: Heatmap

sns.heatmap(df.corr(),annot=True,cmap="RdYlGn",linewidth=0.2)
fig=plt.gcf()
fig.set_size_inches(10,8)
plt.show()
```

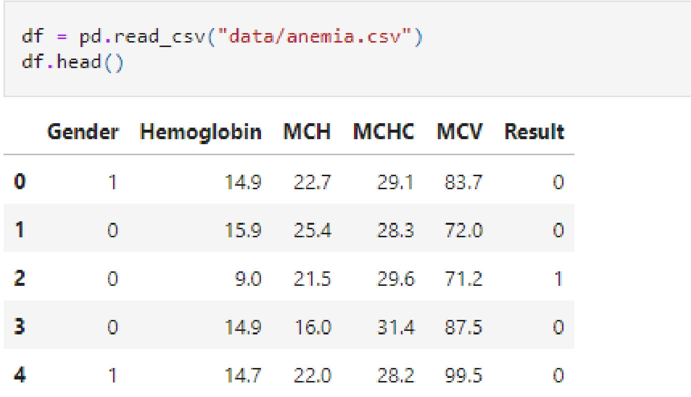


Outliers and Anomalies

-

Data Preprocessing Code Screenshots

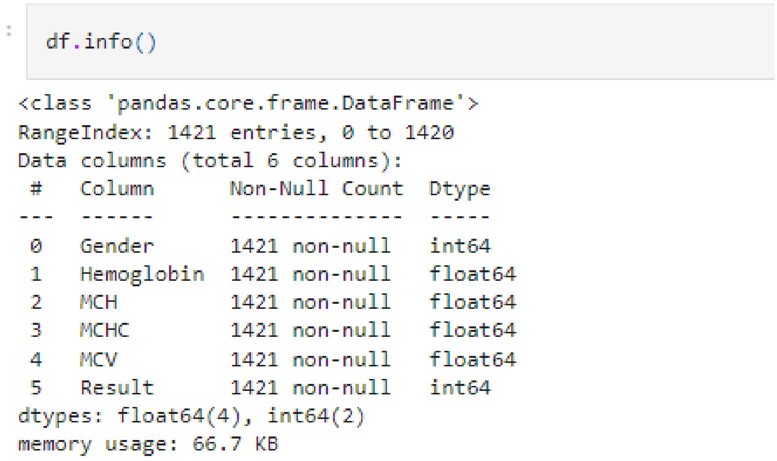
Loading Data



```
df = pd.read_csv("data/anemia.csv")
df.head()
```

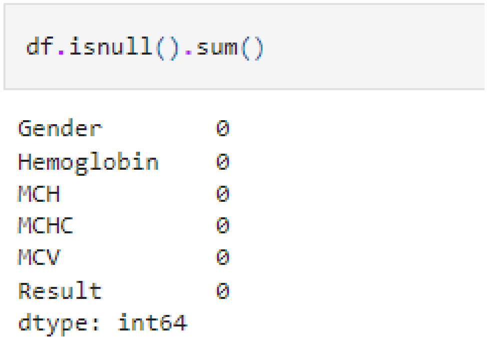
	Gender	Hemoglobin	MCH	MCHC	MCV	Result
0	1	14.9	22.7	29.1	83.7	0
1	0	15.9	25.4	28.3	72.0	0
2	0	9.0	21.5	29.6	71.2	1
3	0	14.9	16.0	31.4	87.5	0
4	1	14.7	22.0	28.2	99.5	0

<table>
<tr>
<td>Handling Missing Data</td>
<td>



```
df.info()
```

```
<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
```

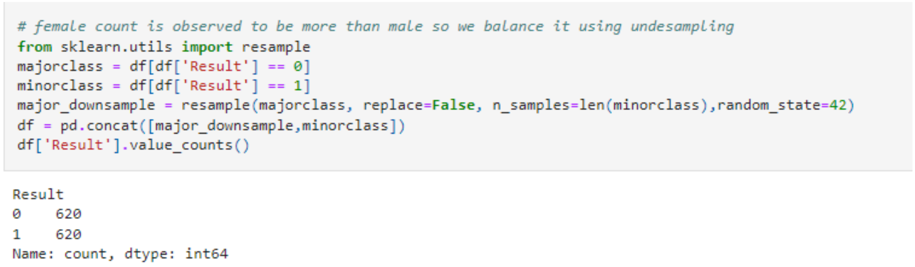


```
df.isnull().sum()
```

```
Gender        0
Hemoglobin    0
MCH           0
MCHC          0
MCV           0
Result        0
dtype: int64
```

There are no missing values in the dataset

</td>
</tr>
<tr>
<td>Data Transformation</td>
<td>



```
# female count is observed to be more than male so we balance it using undesampling
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])
df['Result'].value_counts()
```

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

</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>