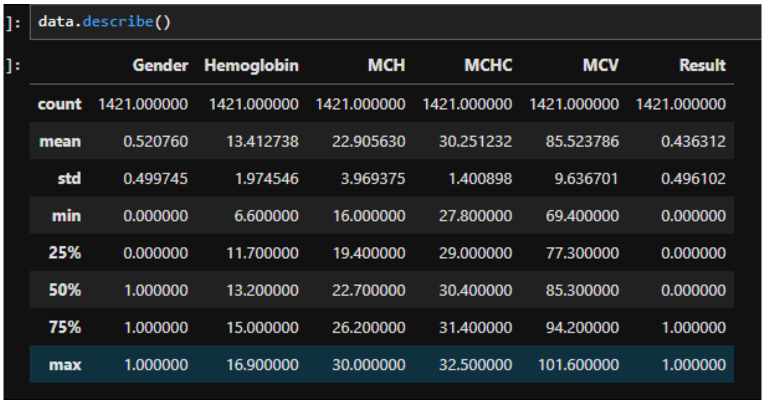
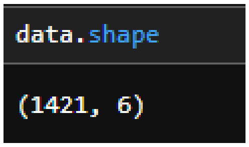
Data Collection and Preprocessing Phase

Date	10 July 2024
Team ID	SWTID1720078683
Project Title	Anemia Sense: Leveraging Machine Learning for Precise Anemia Recognitions
Maximum Marks	6 Marks

Data Exploration and Preprocessing

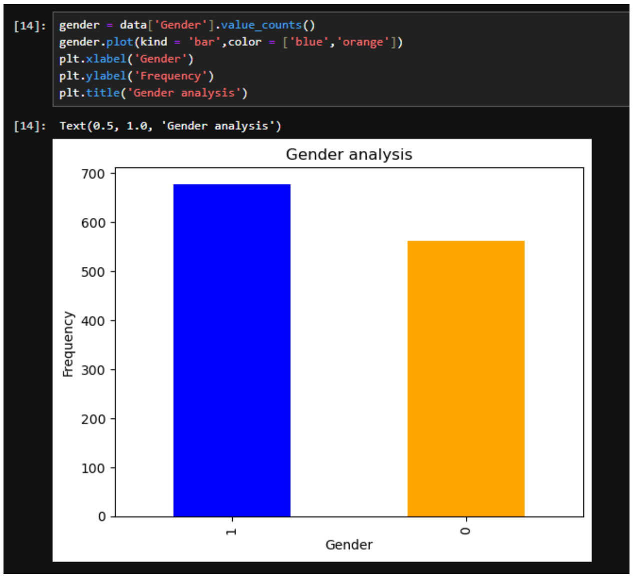
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	**Description**
Data Overview	 

```
]: data.describe()

]:           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
```

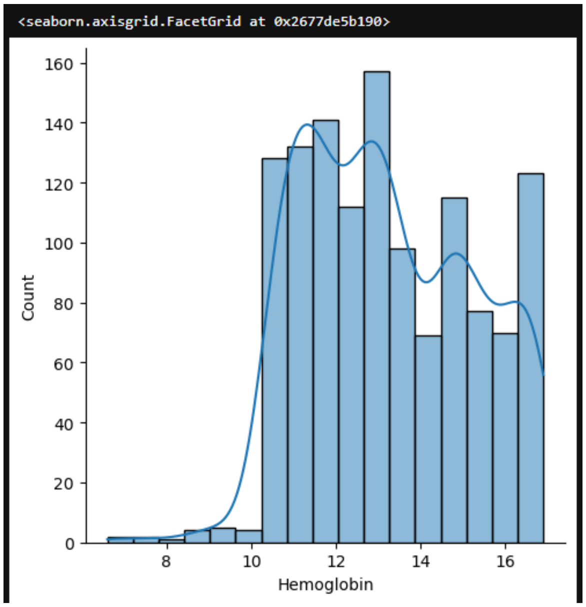
```
data.shape

(1421, 6)
```

Univariate Analysis



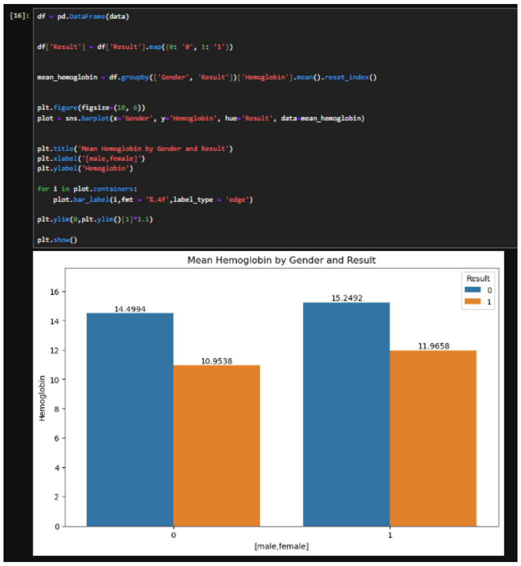
```
[14]: gender = data['Gender'].value_counts()
      gender.plot(kind = 'bar',color = ['blue','orange'])
      plt.xlabel('Gender')
      plt.ylabel('Frequency')
      plt.title('Gender analysis')

[14]: Text(0.5, 1.0, 'Gender analysis')
```

```
[15]: sns.displot(data['Hemoglobin'],kde = True)
```



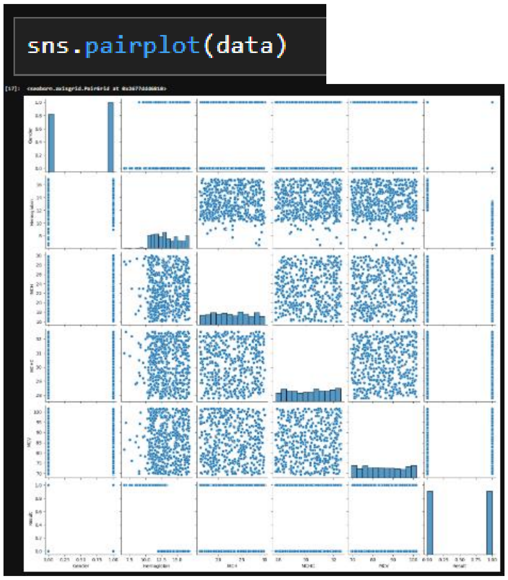
```
<seaborn.axisgrid.FacetGrid at 0x2677de5b190>
```

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



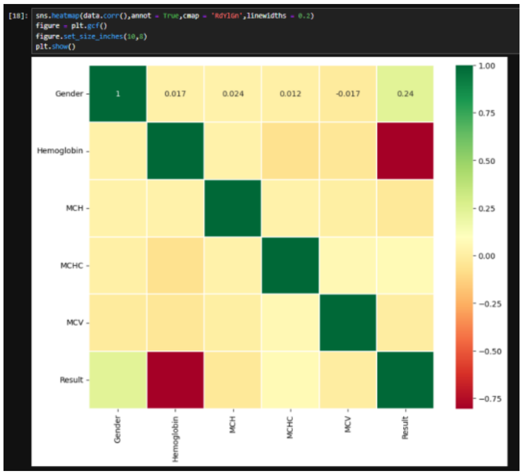
```
[16]: df = pd.DataFrame(data)


      df['Result'] = df['Result'].map({0: '0', 1: '1'})


      mean_hemoglobin = df.groupby(['Gender', 'Result'])['Hemoglobin'].mean().reset_index()


      plt.figure(figsize=(10, 6))
      plot = sns.barplot(x='Gender', y='Hemoglobin', hue='Result', data=mean_hemoglobin)


      plt.title('Mean Hemoglobin by Gender and Result')
      plt.xlabel('[male,female]')
      plt.ylabel('Hemoglobin')

      for i in plot.containers:
          plot.bar_label(i,fmt = '%.4f',label_type = 'edge')

      plt.ylim(0,plt.ylim()[1]*1.1)

      plt.show()
```

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



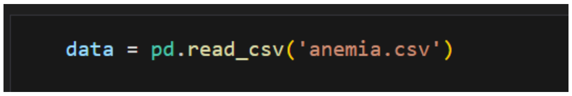
```
sns.pairplot(data)
```

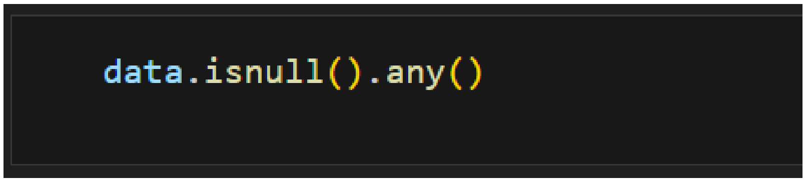
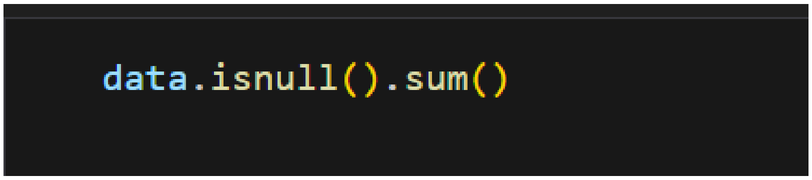
```
[17]: <seaborn.axisgrid.PairGrid at 0x2677dd36810>
```

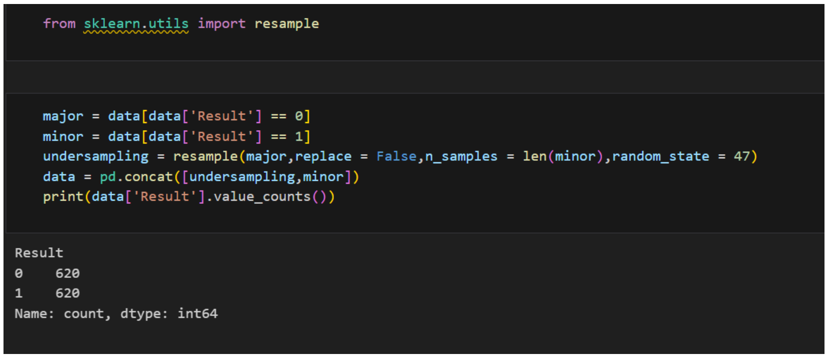


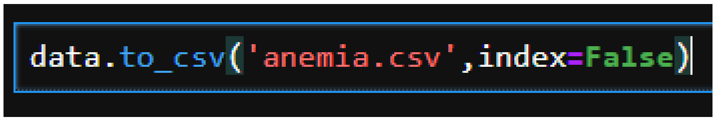
```
[18]: sns.heatmap(data.corr(),annot = True,cmap = 'RdYlGn',linewidths = 0.2)
      figure = plt.gcf()
      figure.set_size_inches(10,8)
      plt.show()
```

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

Data Preprocessing Code Screenshots	
Loading Data	 `data = pd.read_csv('anemia.csv')`

Handling Missing Data	
	`data.isnull().any()`
	
	`data.isnull().sum()`

Data Transformation	
	`from sklearn.utils import resample`
	`major = data[data['Result'] == 0]`
	`minor = data[data['Result'] == 1]`
	`undersampling = resample(major,replace = False,n_samples = len(minor),random_state = 47)`
	`data = pd.concat([undersampling,minor])`
	`print(data['Result'].value_counts())`
	`Result`
	`0    620`
	`1    620`
	`Name: count, dtype: int64`

Save Processed Data	 `data.to_csv('anemia.csv',index=False)`