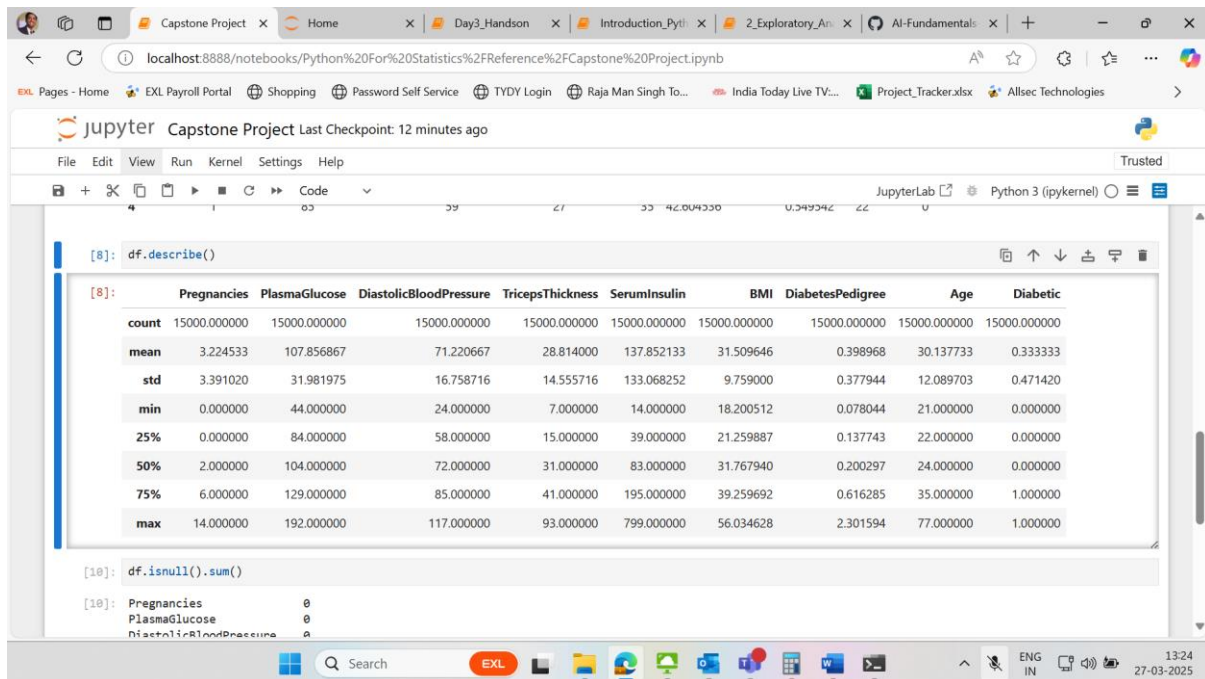


## 1. Data Understanding & Cleaning

- Load the dataset and examine its structure.
  - Identify missing or inconsistent data and suggest appropriate handling techniques.
  - Analyze the distribution of each variable using summary statistics.
  - Discuss potential data transformations (e.g., normalization, scaling).
- 

### Solution:

- ID REMOVED since it is not required
- Data is numerical
- Missing value handling:
  - No Null values found
- Since Measures of central tendencies are different, we **normalize** the data



Capstone Project Last Checkpoint: 56 seconds ago

```
[28]: df_normalized
```

	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic
0	0.000000	0.858108	0.602151	0.313953	0.011465	0.668952	0.510511	0.000000	0.0
1	0.571429	0.324324	0.741935	0.465116	0.028025	0.080352	0.036123	0.035714	0.0
2	0.500000	0.479730	0.247312	0.523256	0.026752	0.616137	0.000438	0.035714	0.0
3	0.642857	0.398649	0.580645	0.209302	0.369427	0.300831	0.541848	0.392857	1.0
4	0.071429	0.277027	0.376344	0.232558	0.026752	0.645027	0.212047		
...	...	...	...	...	...	...	...		
14995	0.714286	0.141892	0.387097	0.453488	0.207643	0.404713	0.031608		
14996	0.142857	0.195946	0.451613	0.232558	0.196178	0.315380	0.352683		
14997	0.000000	0.331081	0.698925	0.418605	0.054777	0.012956	0.156959		
14998	0.000000	0.594595	0.795699	0.127907	0.187261	0.042056	0.100836	0.035714	0.0
14999	0.214286	0.472973	0.440860	0.465116	0.634395	0.476156	0.031175	0.232143	1.0

15000 rows x 9 columns

Microsoft Teams  
'Statistics using Python - A Practical Ap...  
Monika: Please email. Will cascade this error with partners.  
Send a quick reply

## 2. Descriptive Statistics

Compute central tendency measures (mean, median, mode) for all numerical variables.

Calculate dispersion metrics (variance, standard deviation, interquartile range).

Create frequency distributions for categorical variables.

Identify outliers using boxplots and discuss their potential impact.

Capstone Project Last Checkpoint: 12 minutes ago

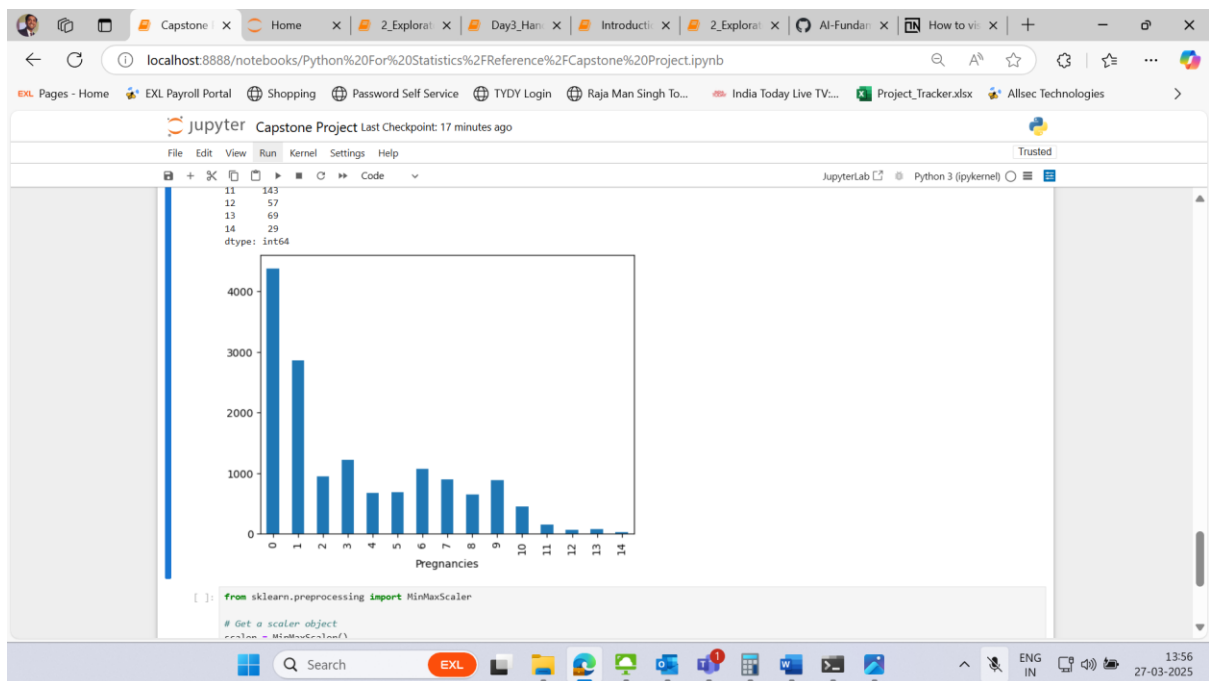
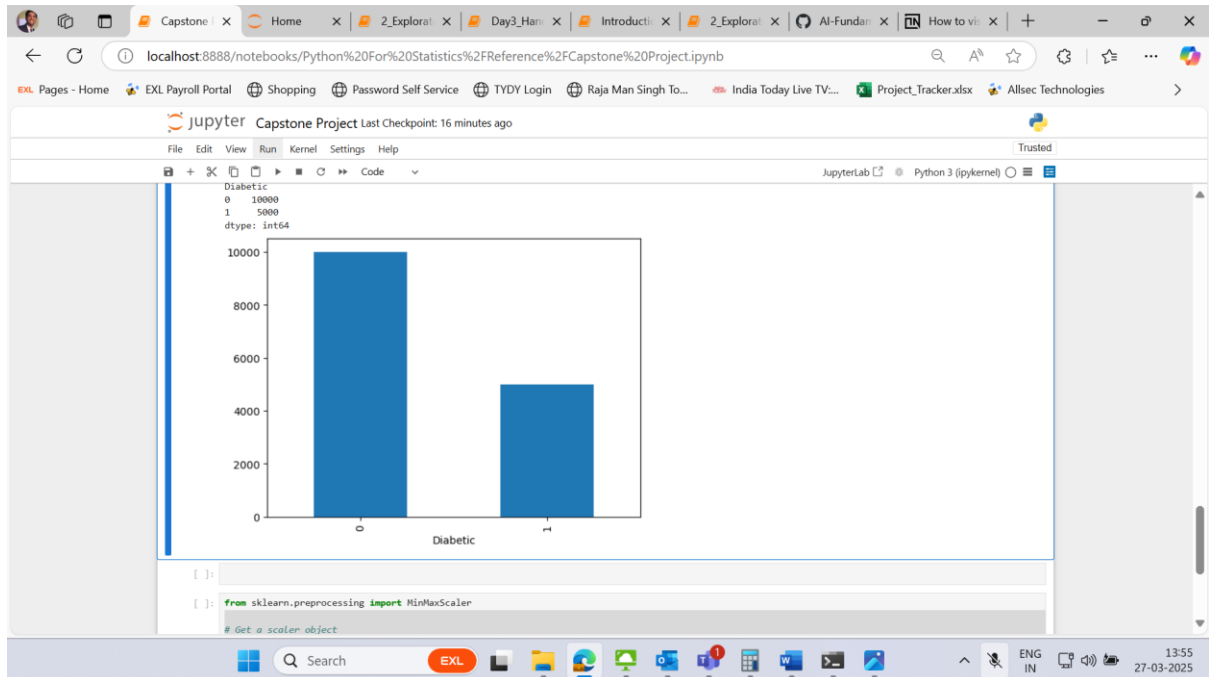
```
[8]: df.describe()
```

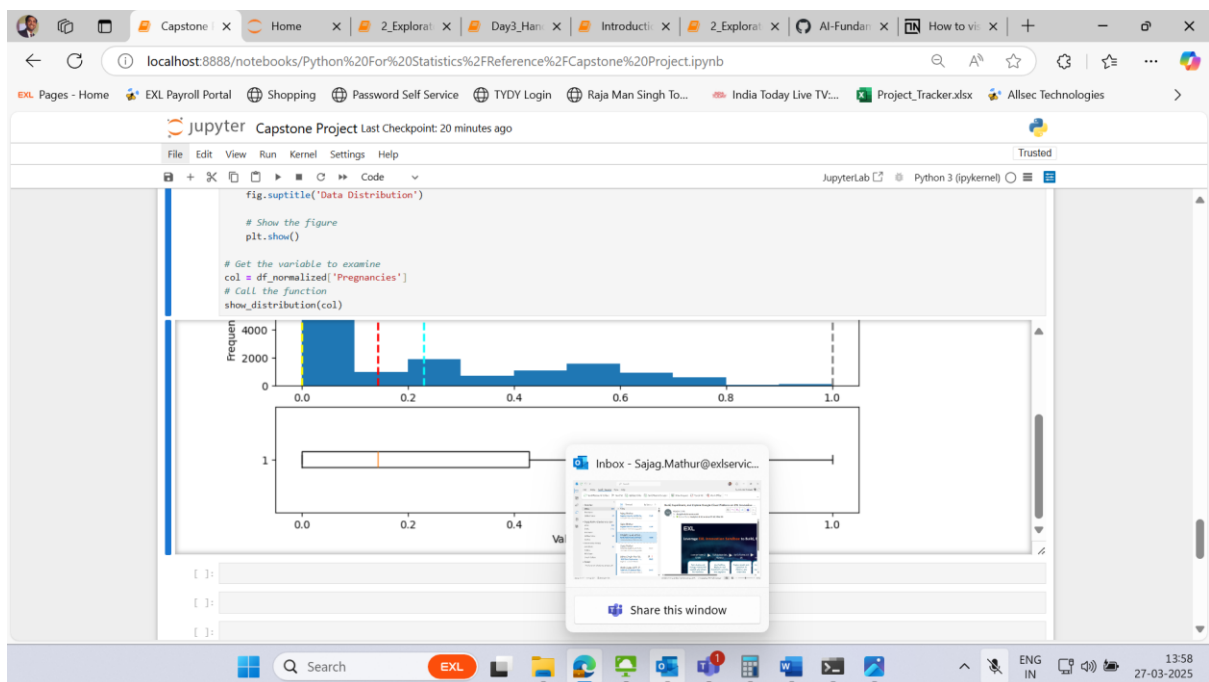
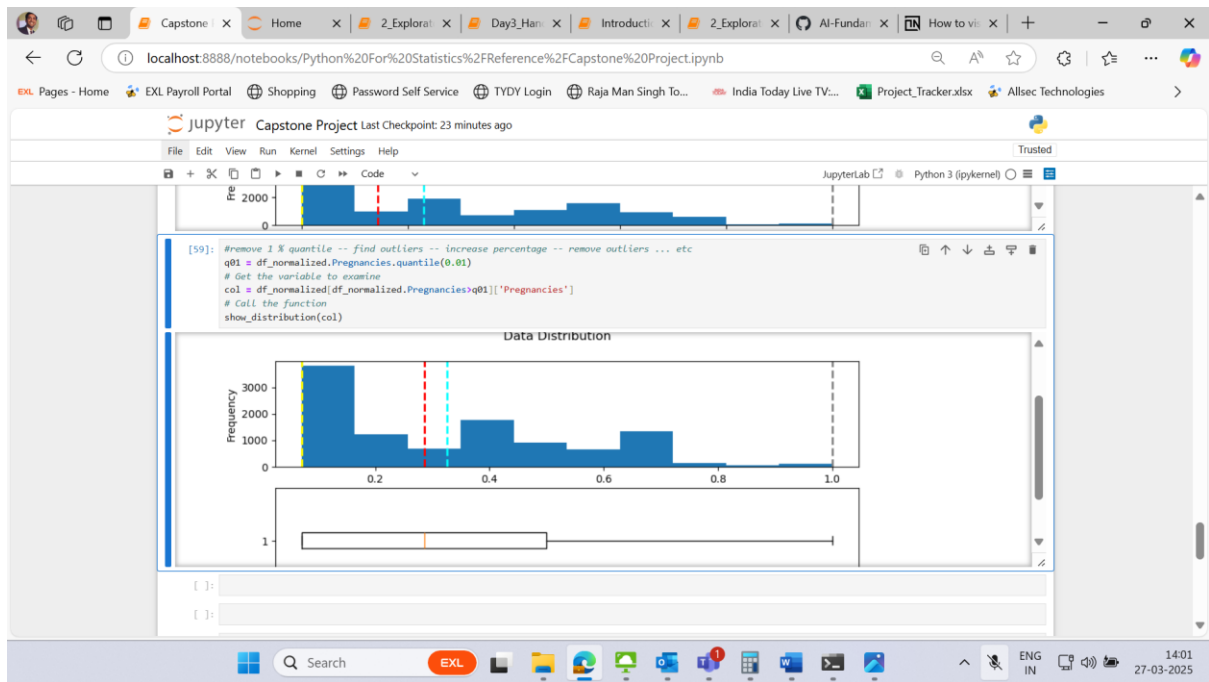
	Pregnancies	PlasmaGlucose	DiastolicBloodPressure	TricepsThickness	SerumInsulin	BMI	DiabetesPedigree	Age	Diabetic
count	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000	15000.000000
mean	3.224533	107.856867	71.220667	28.814000	137.852133	31.509646	0.398968	30.137733	0.333333
std	3.391020	31.981975	16.758716	14.555716	133.068252	9.759000	0.377944	12.089703	0.471420
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200512	0.078044	21.000000	0.000000
25%	0.000000	84.000000	58.000000	15.000000	39.000000	21.259887	0.137743	22.000000	0.000000
50%	2.000000	104.000000	72.000000	31.000000	83.000000	31.767940	0.200297	24.000000	0.000000
75%	6.000000	129.000000	85.000000	41.000000	195.000000	39.259692	0.616285	35.000000	1.000000
max	14.000000	192.000000	117.000000	93.000000	799.000000	56.034628	2.301594	77.000000	1.000000

```
[10]: df.isnull().sum()
```

```
[10]: Pregnancies      0
      PlasmaGlucose    0
      DiastolicBloodPressure  0
```

## Creating pregnancies and diabetic frequency distribution:





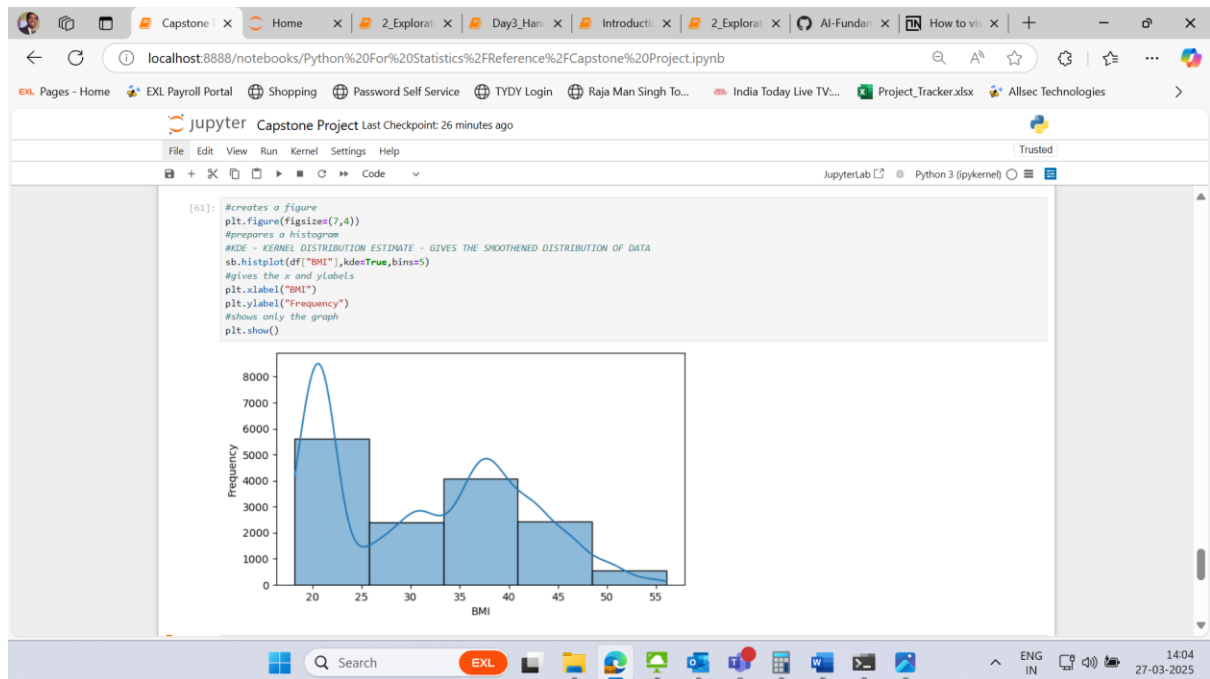
Outliers removed using box plot.

### 3. Exploratory Data Analysis (EDA)

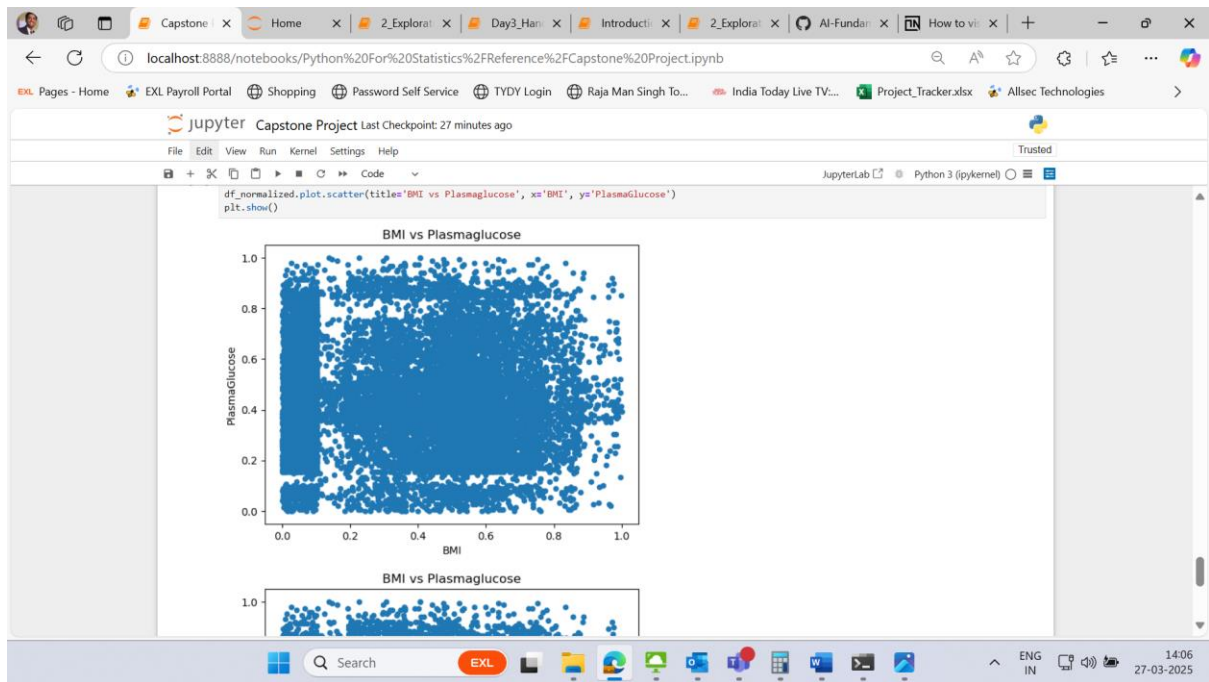
- Generate visualizations (histograms, scatter plots, heatmaps) to explore data distribution and correlations.
- Examine relationships between features such as **BMI vs. PlasmaGlucose**, **Age vs. Diabetes Pedigree**, etc.

- Analyze the correlation matrix to determine the strength of relationships between variables.
  - Perform segmentation analysis by dividing patients into age groups and evaluating diabetes risk.
- 

## Histogram:

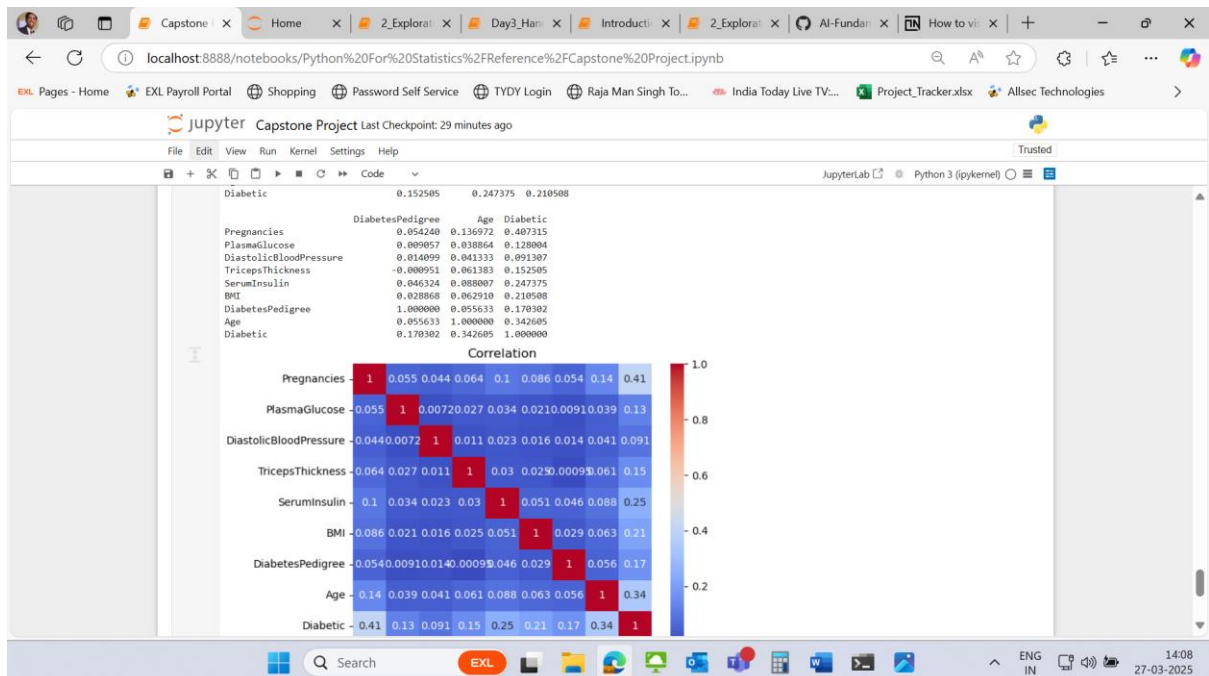


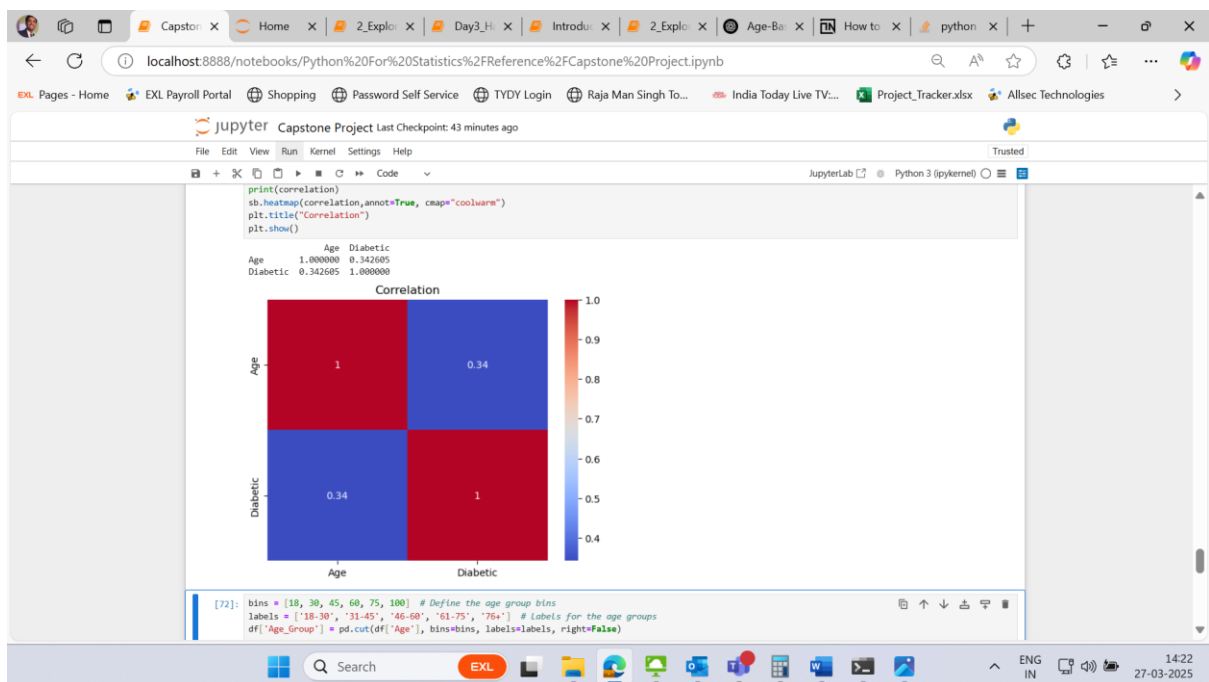
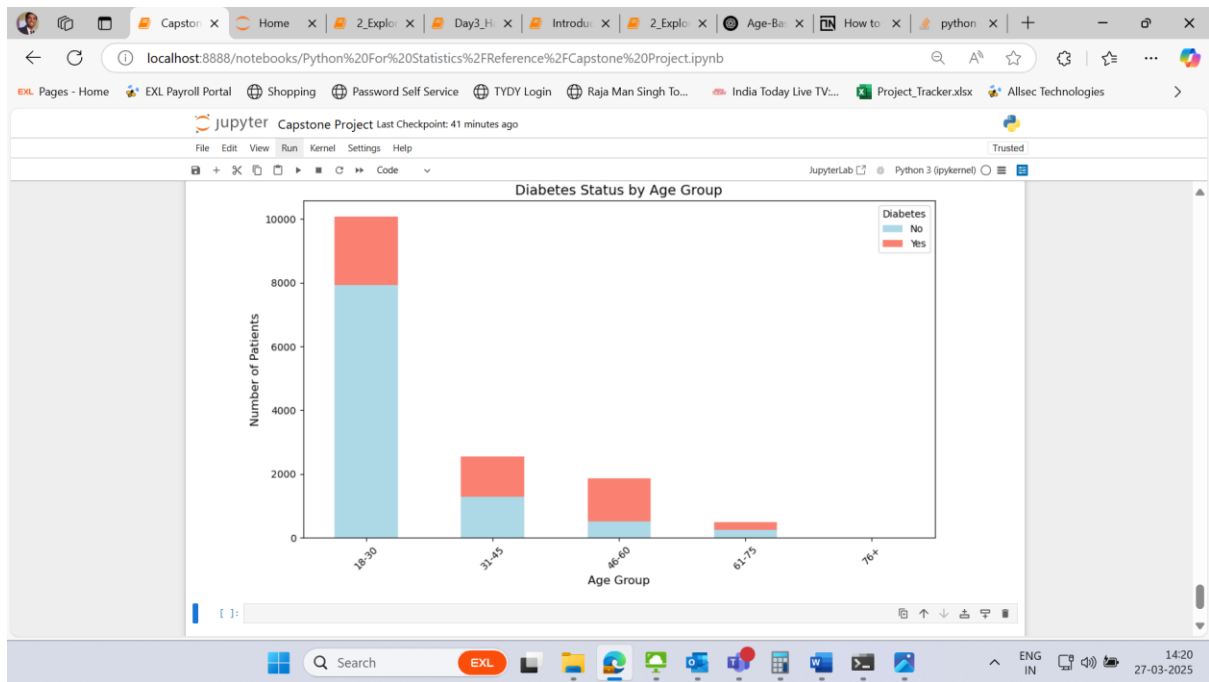
## BMI Vs Plasma Glucose – Scatterplot



No clear relationship between the two.

## Correlation Matrix:





## 4. Inferential Statistics

- Conduct hypothesis testing:
  - Compare the mean BMI of diabetic vs. non-diabetic patients (t-test).
  - Assess the relationship between **Pregnancies** and **Diabetes** using a chi-square test.



- Perform an ANOVA test to compare **PlasmaGlucose** levels across different age groups.
- Interpret p-values and confidence intervals to draw meaningful conclusions.

```

JupyterLab Capstone Project Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

15000 rows x 9 columns

[88]: df2 = df.groupby(['Age_Group'])

C:\Users\sajag177350\AppData\Local\Temp\ipykernel_13948\3402247037.py:1: FutureWarning: The default of observed=False is deprecated and will be changed to True in a future version of pandas. Pass observed=False to retain current behavior or observed=True to adopt the future default and silence this warning.
  df2 = df.groupby(['Age_Group'])

[92]: diabetic_bmi = df[df['Diabetic'] == 1]['BMI']
nondiabetic_bmi = df[df['Diabetic'] == 0]['BMI']
t_statistic, p_value = stats.ttest_ind(diabetic_bmi, nondiabetic_bmi, nan_policy='omit')

[93]: print(f'T-Statistic: {t_stat}, P-Value: {p_value}')
T-Statistic: -390.80395641719724, P-Value: 7.504788727506502e-150

[ ]:

```

**Difference between BMI of Diabetic vs non diabetic patients is statistically significant.**

- Perform an ANOVA test to compare **PlasmaGlucose** levels across different age groups.

```

JupyterLab Capstone Project Last Checkpoint: 1 hour ago

File Edit View Run Kernel Settings Help

[92]: diabetic_bmi = df[df['Diabetic'] == 1]['BMI']
nondiabetic_bmi = df[df['Diabetic'] == 0]['BMI']
t_statistic, p_value = stats.ttest_ind(diabetic_bmi, nondiabetic_bmi, nan_policy='omit')

[93]: print(f'T-Statistic: {t_stat}, P-Value: {p_value}')
T-Statistic: -390.80395641719724, P-Value: 7.504788727506502e-150

[98]: df['Age_Group'].unique()
['18-30', '31-45', '46-60', '76+', '61-75']
Categories (5, object): ['18-30' < '31-45' < '46-60' < '61-75' < '76+']

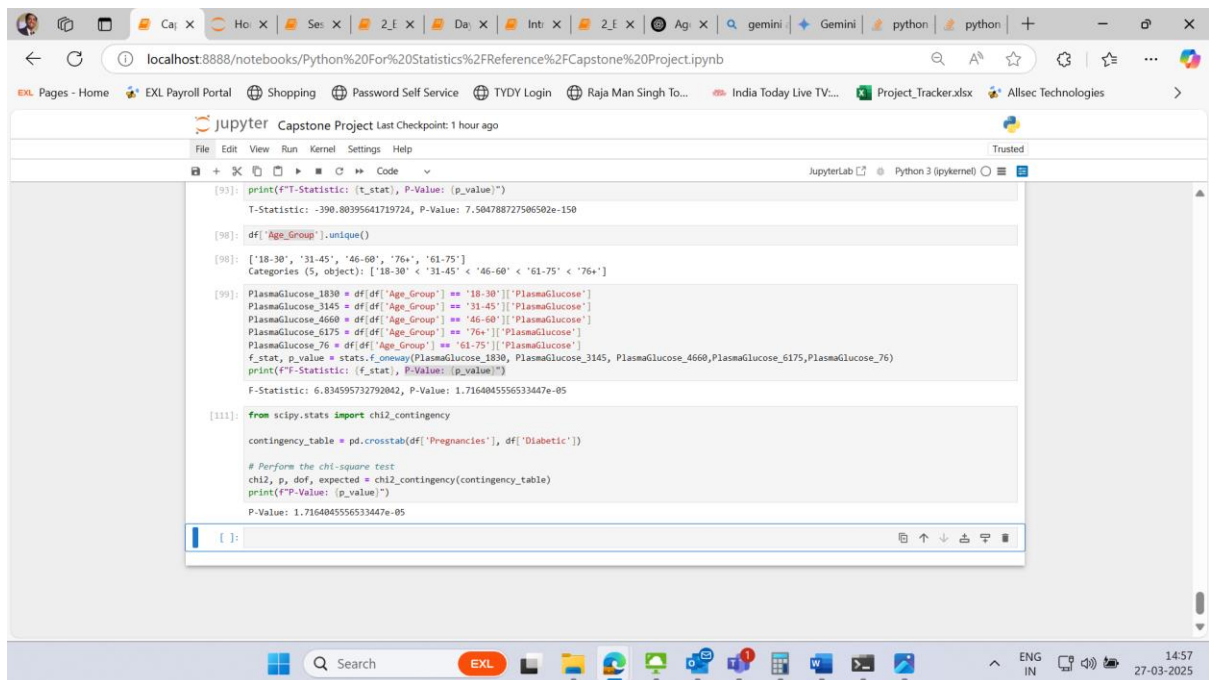
[99]: PlasmaGlucose_1830 = df[df['Age_Group'] == '18-30']['PlasmaGlucose']
PlasmaGlucose_3145 = df[df['Age_Group'] == '31-45']['PlasmaGlucose']
PlasmaGlucose_4660 = df[df['Age_Group'] == '46-60']['PlasmaGlucose']
PlasmaGlucose_6175 = df[df['Age_Group'] == '76+']['PlasmaGlucose']
PlasmaGlucose_76 = df[df['Age_Group'] == '61-75']['PlasmaGlucose']
f_stat, p_value = stats.f_oneway(PlasmaGlucose_1830, PlasmaGlucose_3145, PlasmaGlucose_4660, PlasmaGlucose_6175, PlasmaGlucose_76)
print(f'F-Statistic: {f_stat}, P-Value: {p_value}')
F-Statistic: 6.834595732792042, P-Value: 1.716404556533447e-05

[ ]:

```



**PlasmaGlucose is statistically different across agegroups.**



The screenshot shows a JupyterLab notebook interface with a code editor and a console output. The code performs a t-test to compare PlasmaGlucose levels across different age groups. The output shows a T-Statistic of -390.80395641719724 and a P-Value of 7.504788727506502e-150, indicating a statistically significant difference.

```
[97]: print(f'T-Statistic: {t_stat}, P-Value: {p_value}')
T-Statistic: -390.80395641719724, P-Value: 7.504788727506502e-150

[98]: df['Age_Group'].unique()

[99]: ['18-30', '31-45', '46-60', '76+', '61-75']
Categories (5, object): ['18-30' < '31-45' < '46-60' < '61-75' < '76+']

[100]: PlasmaGlucose_1830 = df[df['Age_Group'] == '18-30']['PlasmaGlucose']
PlasmaGlucose_3145 = df[df['Age_Group'] == '31-45']['PlasmaGlucose']
PlasmaGlucose_4660 = df[df['Age_Group'] == '46-60']['PlasmaGlucose']
PlasmaGlucose_6175 = df[df['Age_Group'] == '61-75']['PlasmaGlucose']
PlasmaGlucose_76 = df[df['Age_Group'] == '76+']['PlasmaGlucose']
f_stat, p_value = stats.f_oneway(PlasmaGlucose_1830, PlasmaGlucose_3145, PlasmaGlucose_4660, PlasmaGlucose_6175, PlasmaGlucose_76)
print(f'T-Statistic: {f_stat}, P-Value: {p_value}')

[101]: F-Statistic: 6.834595732792042, P-Value: 1.7164045556533447e-05

[102]: from scipy.stats import chi2_contingency

contingency_table = pd.crosstab(df['Pregnancies'], df['Diabetic'])

# Perform the chi-square test
chi2, p, dof, expected = chi2_contingency(contingency_table)
print(f'P-Value: {p_value}')

[103]: P-Value: 1.7164045556533447e-05
```

**P Value is statistically significant. Hence, there is a relationship between pregnancies and diabetes.**

## Conclusions and Recommendations:

Pregnancies, age and seruminsulin impact Diabetese positively. Hence, close attention should be given on these.