Analyzing Diabetes Progression Using Python-Based Clinical Data Analytics

Group 5: Anthony Cid, Staci Lobosco, Robert Shea, Amanda Wilson, Dana Wiwczar

ISM6644 - Python Fundamentals for Business Analytics
Dr. Mirzaei
October 16, 2025

Project Description

- This project applies Python-based data analytics to a clinical dataset related to diabetes progression.

- Objectives:
  - Prepare and clean the dataset for analysis
  - Explore the relationships between variables
  - Create clear and informative charts using Matplotlib and Seaborn
  - Generate insights for clinical decision making

Data

The dataset used for this project simulates clinical data collected from a group of patients living with diabetes.

Dataset: Diabetes_data.xlsx

Key Variables:

- Age
- Sex
- Body mass index
- Blood pressure
- Blood serum measurements
- Target
- Smoking status
- Insurance
- Hypertension

Objective 1: Clean and Prepare the Data for Analysis

```
#Obj 1: Rename columns
df.rename(columns={'s1': 'total_cholesterol',
                   's2': 'ldl',
                   's3': 'hdl',
                   's4': 'tch_hdl_ratio',
                   's5': 'log_serum_triglycerides',
                   's6': 'blood_sugar_level'}, inplace=True)
```

Rename columns

Replace missing values from numeric columns with their mean values

```python
#Obj 1: Identify the columns with missing data
df.isnull().sum()
```

```
PID                       0
age                       1
sex                       0
bmi                       2
bp                        0
total_cholesterol         0
ldl                       1
hdl                       1
tch_hdl_ratio             0
log_serum_triglycerides   0
blood_sugar_level         0
target                    0
smoking_status            0
insurance                 0
hypertension              0
dtype: int64
```

```python
#Obj 1: Make a dataframe copy so original data stays untouched
df_clean = df.copy()

# Replace missing numeric columns with their mean
for col in df_clean.select_dtypes(include='number').columns:
    if df_clean[col].isna().any():
        df_clean[col] = df_clean[col].fillna(df_clean[col].mean())

# Replace missing categorical columns with 'Unknown'
for col in df_clean.select_dtypes(exclude='number').columns:
    if df_clean[col].isna().any():
        df_clean[col] = df_clean[col].fillna("Unknown")
```

```python
#Obj 1:  Identify outliers

from scipy import stats

# Identify outlier counts per numeric column to show me which variables have extreme values
numeric_cols = df_clean.select_dtypes(include='number').columns
outlier_counts = {}
for col in numeric_cols:
    z = np.abs(stats.zscore(df_clean[col]))
    outlier_counts[col] = (z > 3).sum()

print("Outlier counts per column:")
print(outlier_counts)
```

Identify outliers

```
Outlier counts per column:
{'PID': 0, 'age': 1, 'bmi': 2, 'bp': 0, 'total_cholesterol': 2, 'ldl': 2, 'hdl': 5, 'tch_hdl_ratio': 4, 'log_serum_triglycerides': 1, 'blood_sugar_level': 1,
'target': 0}
```

```python
#Obj 1: Remove outliers using Z-score filter
df_no_outliers = df_clean.copy()

for col in numeric_cols:
    z = np.abs(stats.zscore(df_no_outliers[col]))
    df_no_outliers = df_no_outliers[z < 3]  # drop only those beyond ±3 for that column
```

Remove outliers

```python
#Obj 1: Check dataframe without outliers
print(f"Before: {df_clean.shape[0]} rows")
print(f"After:  {df_no_outliers.shape[0]} rows")
```

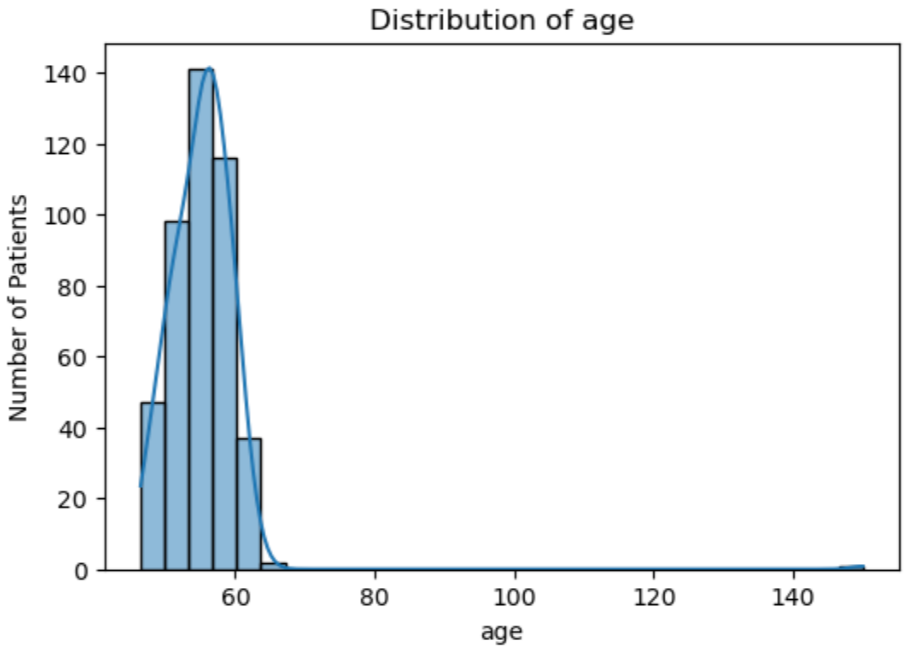
```
Before: 442 rows
After:  427 rows
```

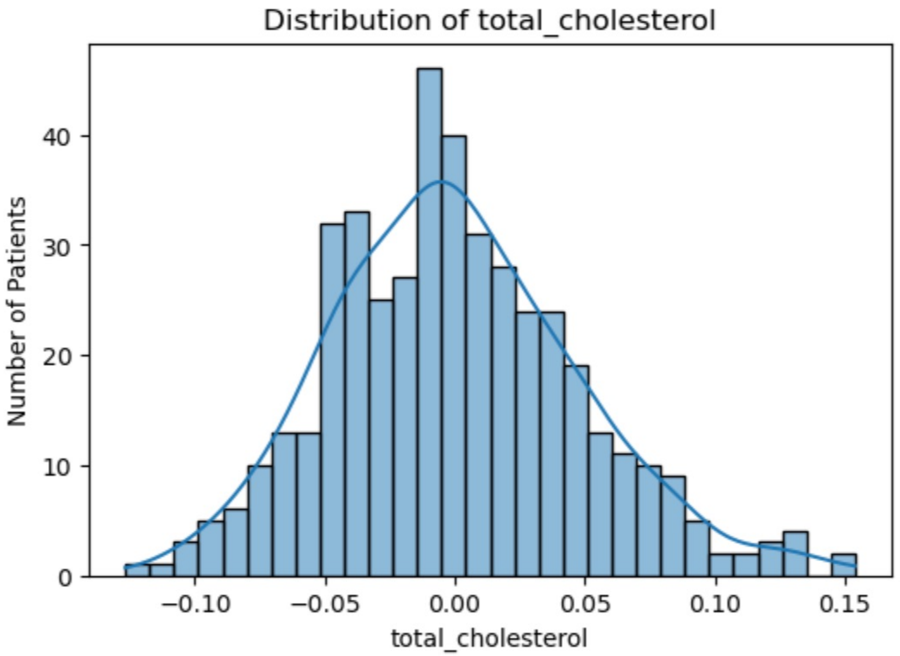
Objective 2: Explore Patterns and Relationships

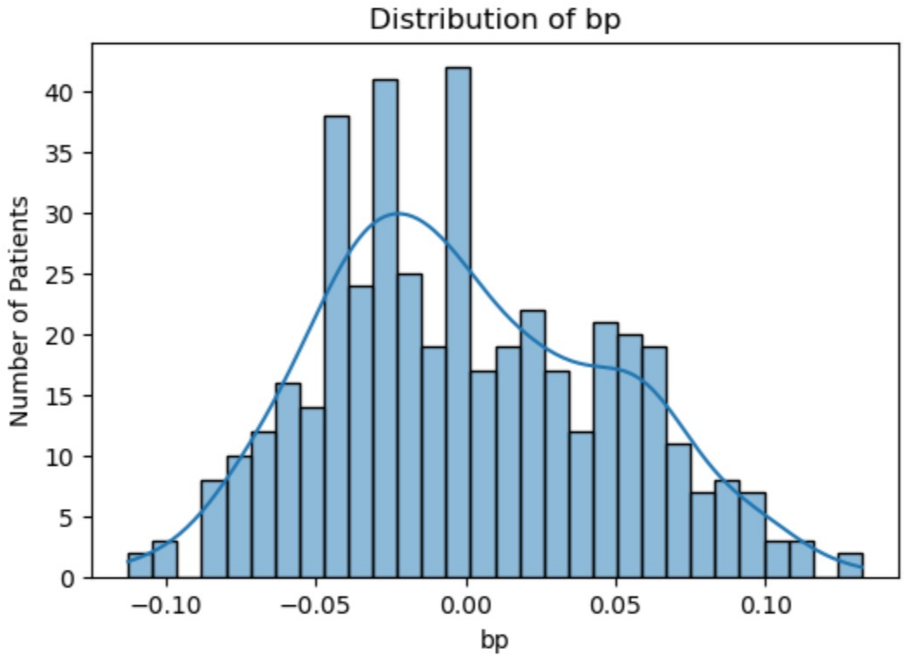
```python
#Obj 2: Overview of numerical variables
df.describe()

# Check categorical distributions
for col in ['sex', 'smoking_status', 'insurance', 'hypertension']:
    print(f"\nValue counts for {col}:")
    print(df[col].value_counts())
# Fairly balanced sex distribution
# Majority never-smokers, many current/former
# Mixed insurance coverage, medicare / medicaid plans most common
# Approx 1/4 patients are hypertensive
```

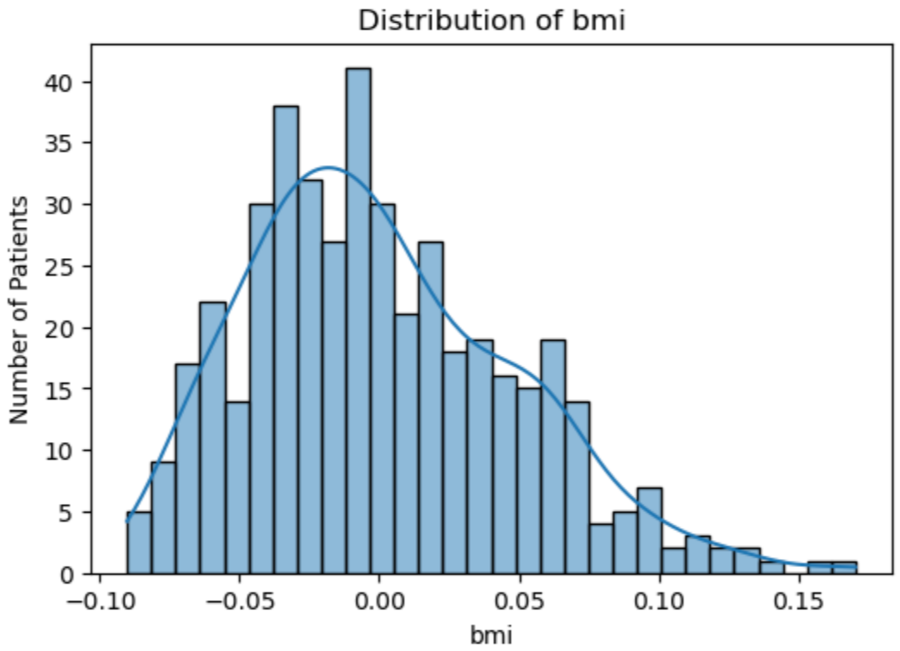
```
Value counts for sex:
sex
female    235
male      207
Name: count, dtype: int64

Value counts for smoking_status:
smoking_status
never      164
current    149
former     129
Name: count, dtype: int64

Value counts for insurance:
insurance
Medicare     124
Medicaid     116
Uninsured    114
Private       88
Name: count, dtype: int64

Value counts for hypertension:
hypertension
no     321
yes    121
Name: count, dtype: int64
```

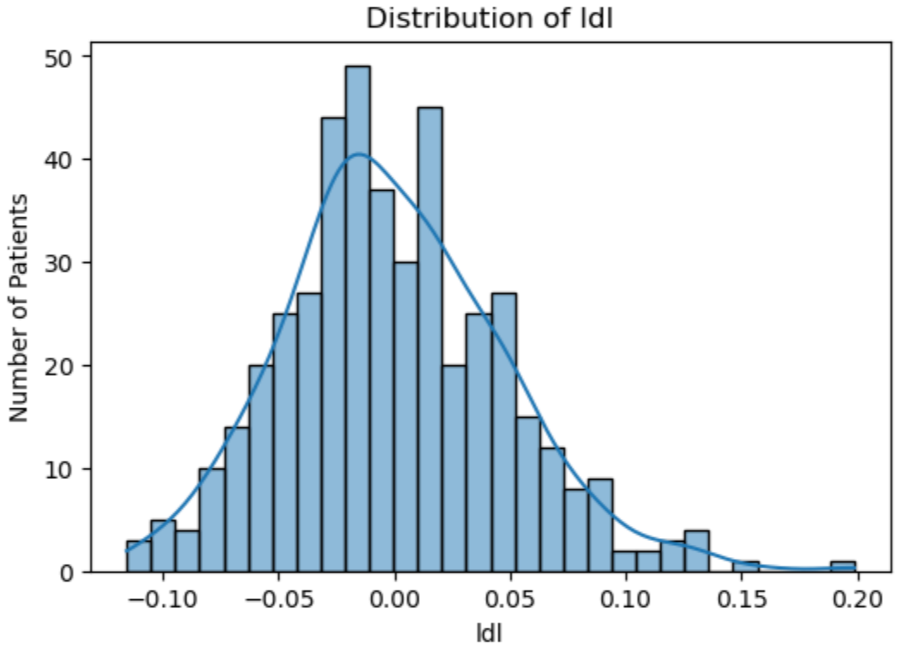
```python
#Obj 2/3: Distribution of continuous variables
continuous = [
    'age','bmi','bp','total_cholesterol','ldl','hdl',
    'tch_hdl_ratio','log_serum_triglycerides','blood_sugar_level','target'
]

for col in continuous:
    plt.figure(figsize=(6,4))
    sns.histplot(df_clean[col], kde=True, bins=30)
    plt.title(f'Distribution of {col}')
    plt.xlabel(col)
    plt.ylabel('Number of Patients')
    plt.show()

# Most continuous variables look roughly normal from the plots.
# Will check skewness to confirm which ones lean higher or lower.
```

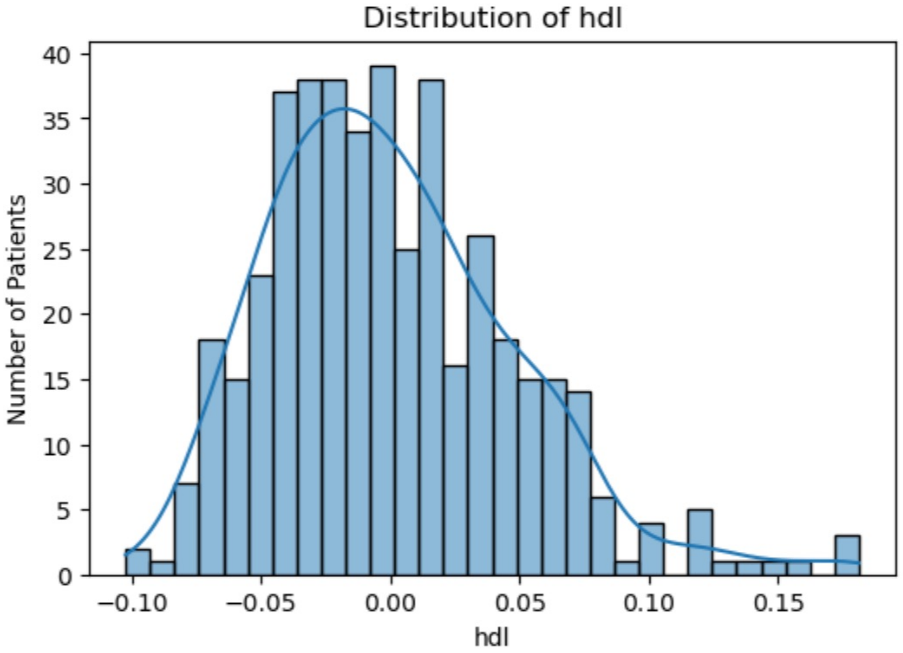

Distribution of age
Number of Patients
0
20
40
60
80
100
120
140
60
80
100
120
140
age

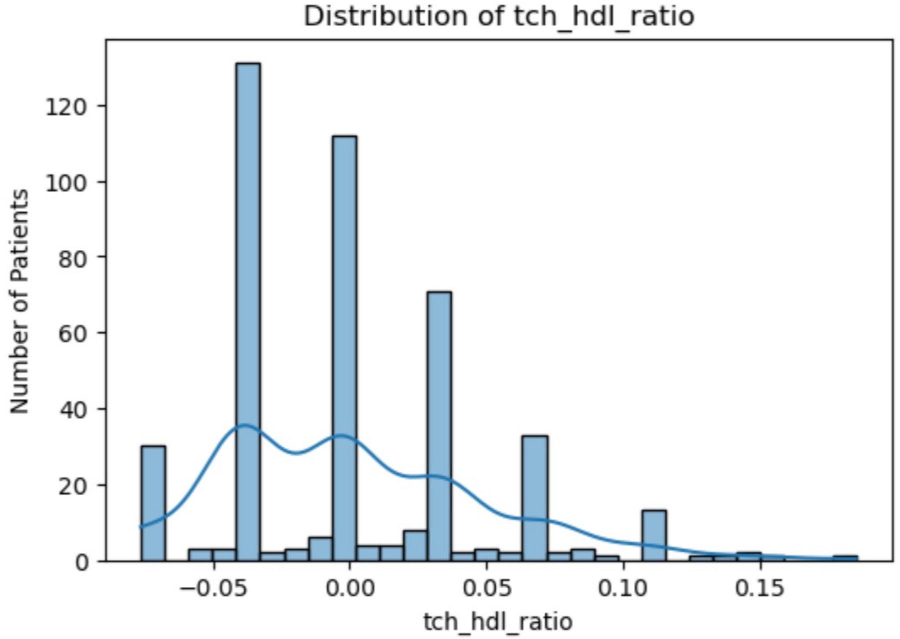

Distribution of total_cholesterol
Number of Patients
0
10
20
30
40
−0.10
−0.05
0.00
0.05
0.10
0.15
total_cholesterol

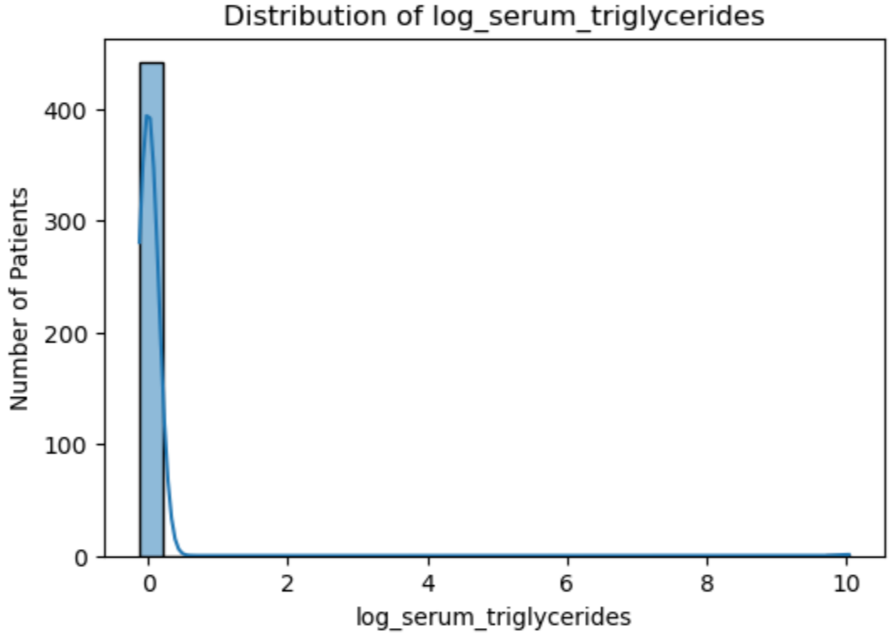

Distribution of bp
Number of Patients
0
5
10
15
20
25
30
35
40
−0.10
−0.05
0.00
0.05
0.10
bp

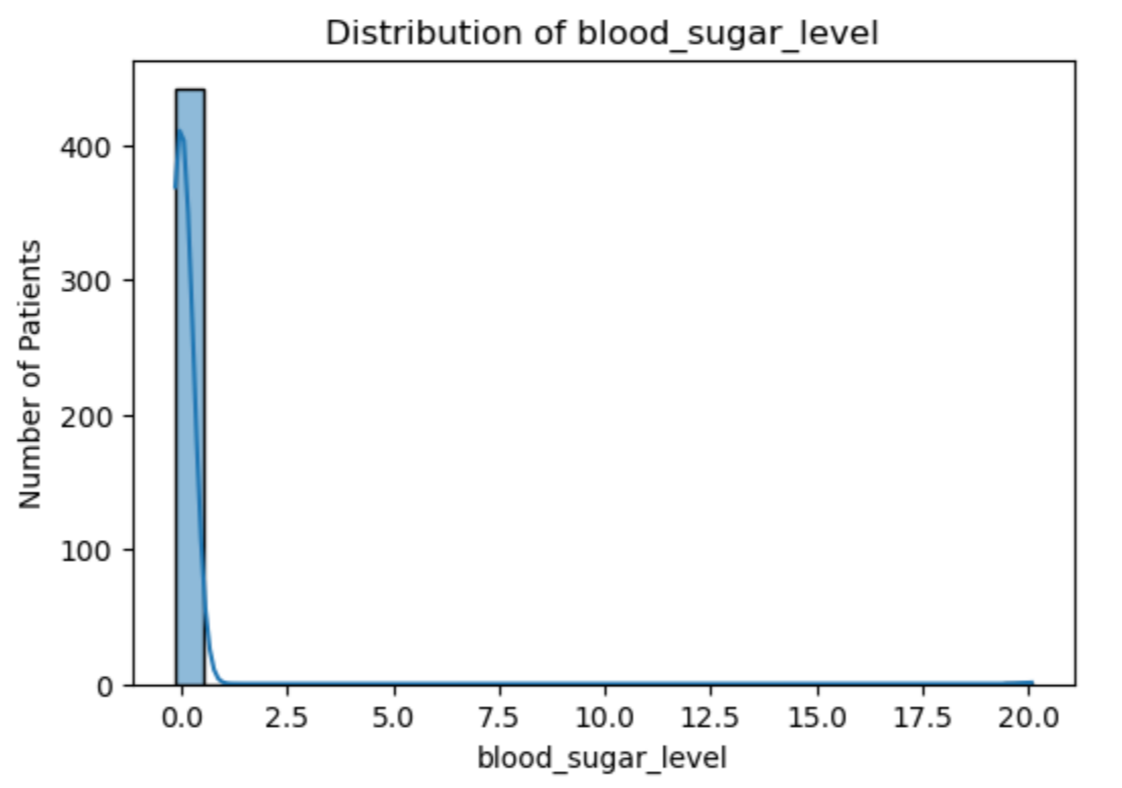

Distribution of bmi
Number of Patients
0
5
10
15
20
25
30
35
40
−0.10
−0.05
0.00
0.05
0.10
0.15
bmi

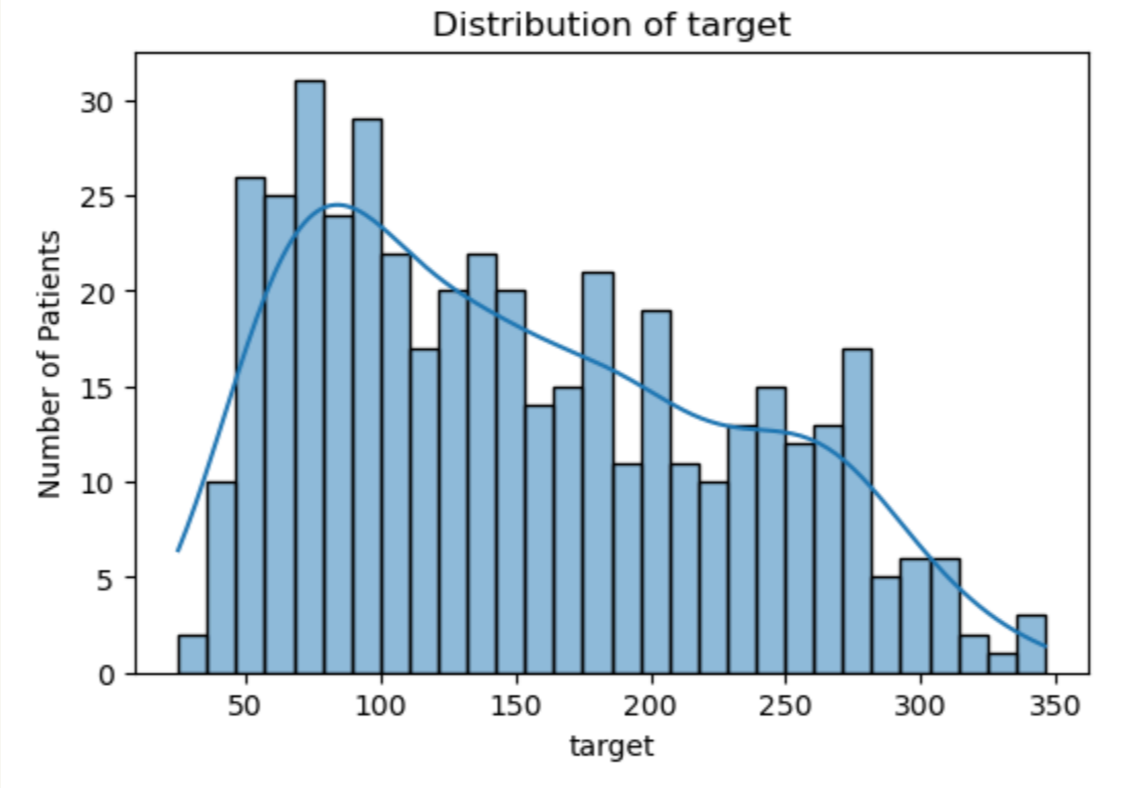

Distribution of ldl
Number of Patients
ldl
0
10
20
30
40
50
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20


Distribution of hdl
Number of Patients
hdl
0
5
10
15
20
25
30
35
40
−0.10
−0.05
0.00
0.05
0.10
0.15


Distribution of tch_hdl_ratio
Number of Patients
tch_hdl_ratio
0
20
40
60
80
100
120
−0.05
0.00
0.05
0.10
0.15


Distribution of log_serum_triglycerides
Number of Patients
log_serum_triglycerides
0
100
200
300
400
0
2
4
6
8
10





Clinical Application:

- These distributions show how the continuous variables (age, BMI, blood pressure, and lipid measures) are spread across the patient population.
- Most variables appear roughly normal, meaning the dataset represents a balanced range of patients rather than being dominated by extreme outliers.
- This is clinically useful because it indicates that most patients fall within expected healthy-to-moderate ranges, allowing comparisons across risk factors to be meaningful.

Clinical Application Continued…

Key observations:

- BMI and blood pressure distributions have slightly higher tails, reflecting a subgroup of patients who are overweight or hypertensive - these individuals are typically at higher metabolic risk.
- Lipid measures such as total cholesterol and the TCH/HDL ratio show moderate variation, aligning with real-world differences in diet, medication use, and genetics.
- Clinically, understanding these variable distributions helps healthcare professionals target interventions toward patients at the higher end of these ranges, who may need earlier monitoring or treatment adjustments.

Skewness

```
#Obj 2: Check skewness for all continuous variables
num_cols = ['age','bmi','bp','total_cholesterol','ldl','hdl',
            'tch_hdl_ratio','log_serum_triglycerides','blood_sugar_level','target']

df_clean[num_cols].skew().sort_values(ascending=False)

# blood_sugar_level and log_serum_triglycerides show strong right skew,
# BMI, HDL, and ratio values show mild skew.
# LDL, target, cholesterol, and BP look normal.
# Overall, skewness appears realistic for clinical data no extreme outliers remain.
```

```
blood_sugar_level          20.945667
log_serum_triglycerides    20.713140
age                         9.313314
hdl                         0.796625
tch_hdl_ratio               0.735374
bmi                         0.603112
ldl                         0.442341
target                      0.440563
total_cholesterol           0.378108
bp                          0.290658
dtype: float64
```

- blood_sugar_level and log_serum_triglycerides show strong right skew
- BMI, HDL, and ratio values show mild skew.
- LDL, target, cholesterol, and BP look normal.
- Overall, skewness appears realistic for clinical data no extreme outliers remain.

Correlation Analysis

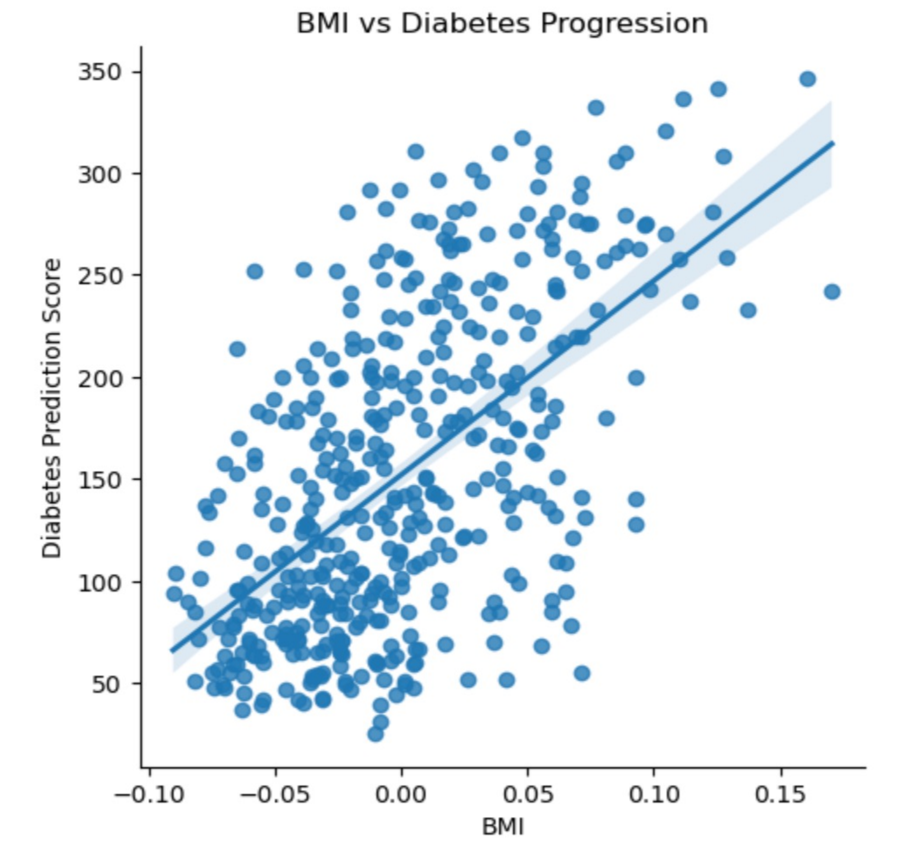
```
#Obj 2: Correlation Analysis
# Looking at relationship of numeric variable to diabetes progression (target)
corr = df.select_dtypes(include='number').corr()

# Sort by strongest to weakest correlation with target
target_corr = corr['target'].sort_values(ascending=False)
print("Correlation of variables with target:")
print(target_corr)

# Higher positive numbers indicate a stronger correlation with disease progression
# Negative numbers indicate the variable may go the opposite way (better outcomes)
```

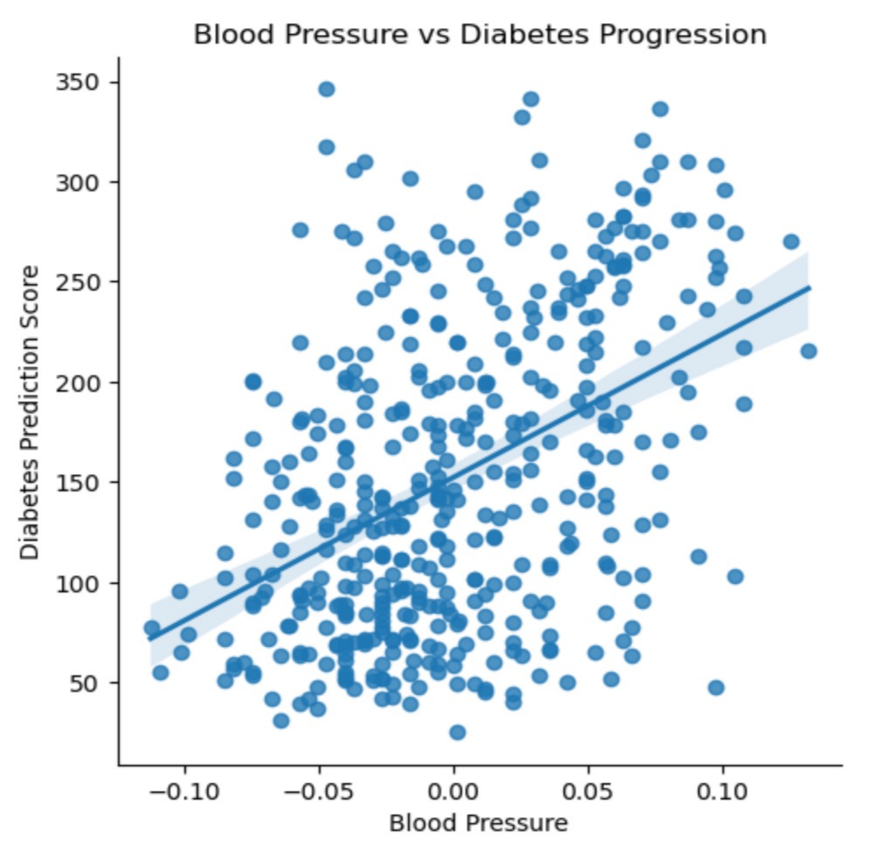
```
Correlation of variables with target:
target                     1.000000
bmi                        0.586049
bp                         0.441482
tch_hdl_ratio              0.430453
total_cholesterol          0.212022
ldl                        0.174888
age                        0.092485
log_serum_triglycerides    0.067737
PID                        0.059912
blood_sugar_level         -0.022871
hdl                       -0.396161
Name: target, dtype: float64
```

- Higher positive numbers indicate a stronger correlation with disease progression
- Negative numbers indicate the variable may go the opposite way (better outcomes)

Objective 3: Create Meaningful Visualization Using Matplotlib and Seaborn

```
#Obj 2/3: Relationship Exploration & Data Visualization
# Visualized how BMI, BP, and blood sugar relate to diabetes progression using scatter plots with a regression line
#Higher progression values = stronger correlation

import warnings
warnings.filterwarnings("ignore") #applied this because the system recognized it was a lot of data on a small chart and wanted to make it all fit in the chart

sns.lmplot(data=df_clean, x='bmi', y='target') #has regress
plt.title('BMI vs Diabetes Progression')
plt.xlabel('BMI')
plt.ylabel('Diabetes Prediction Score')
plt.show()

sns.lmplot(data=df_clean, x='bp', y='target')
plt.title('Blood Pressure vs Diabetes Progression')
plt.xlabel('Blood Pressure')
plt.ylabel('Diabetes Prediction Score')
plt.show()

#The below chart appears incorrect, but it is not. Blood sugar levels are in the dataset as standardized values, all near zero  so the regression line appears
sns.lmplot(data=df, x='blood_sugar_level', y='target')
plt.title('Blood Sugar Level vs Diabetes Progression')
plt.xlabel('Blood Sugar Level')
plt.ylabel('Diabetes Prediction Score')
plt.show()


# Higher BMI correlates with higher diabetes incidence, this is the strongest linear correlation of all the variables described
# Higher blood pressure correlates with a higher diabetes incidence, in a less concentrated manner, showing a moderate association between the two
# Blood sugar doesn’t show a clear pattern or trend
```



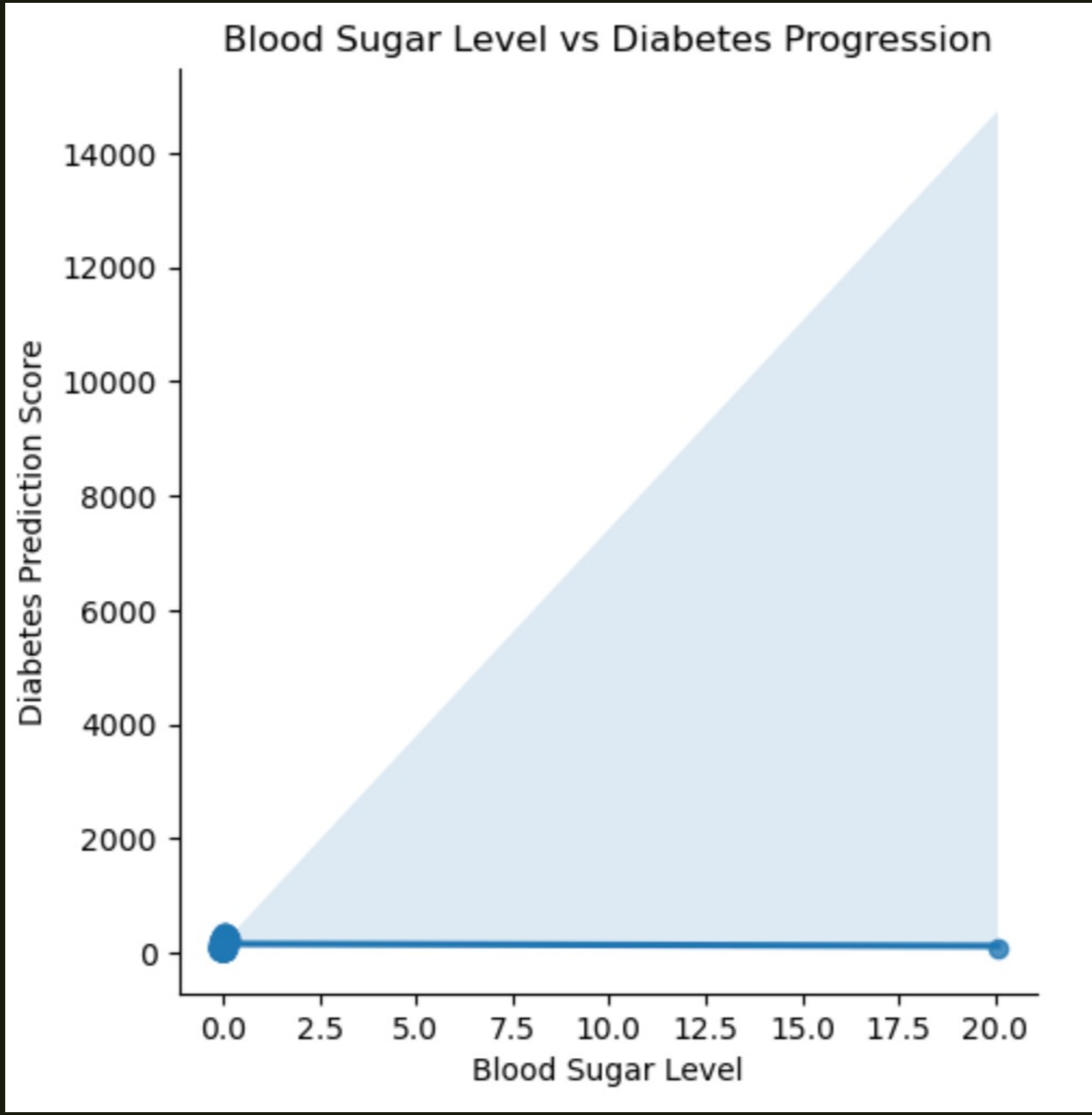


BMI vs Diabetes Progression:

- The plot shows that patients with higher BMI values tend to have higher diabetes-progression scores.
- Excess body fat contributes to insulin resistance and poor glycemic control, so early weight management and nutrition counseling are key to slowing disease progression.

Blood Pressure vs Diabetes Progression:

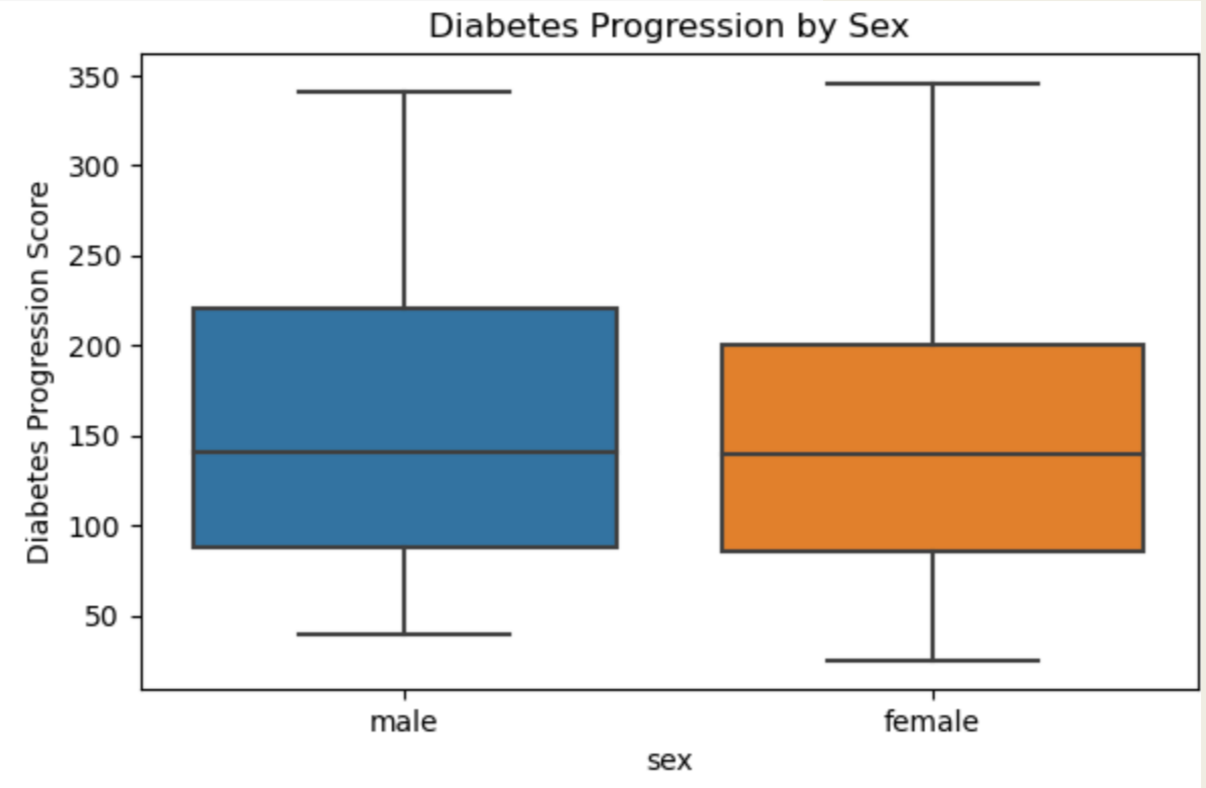
- A moderate upward trend indicates that patients with higher blood pressure also experience faster diabetes progression.
- Elevated blood pressure adds cardiovascular strain and metabolic stress, reinforcing the need for regular monitoring and antihypertensive therapy in diabetic care plans.



Blood Sugar Level vs Diabetes Progression:

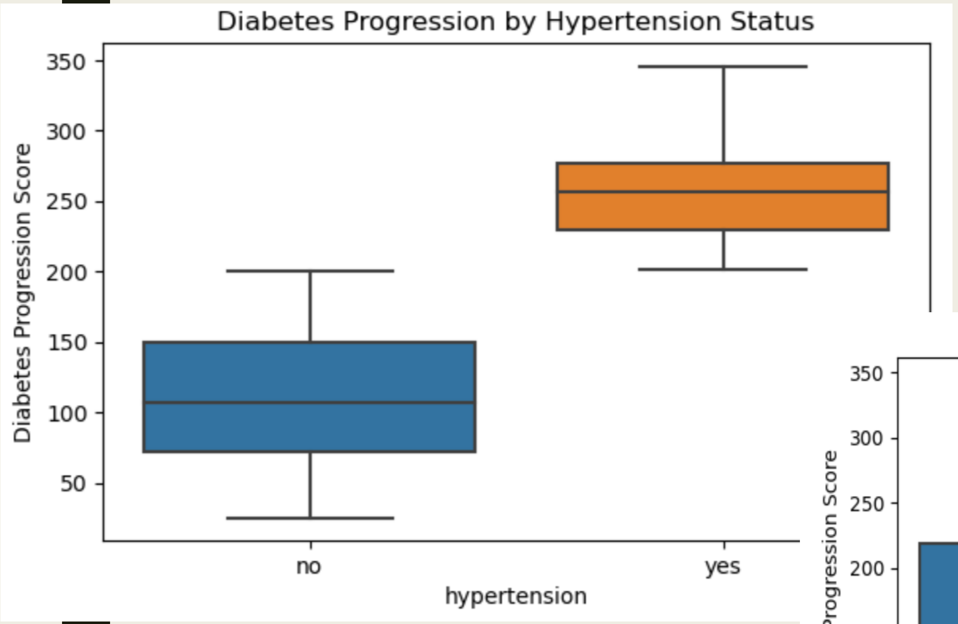
- Although the regression line appears flat, this is due to the standardized blood sugar values being centered near zero.
- In a clinical context, maintaining blood glucose within a stable range is crucial, and variations outside this normalized range would likely show a stronger relationship with disease outcomes.

```python
#Obj 2/3: Group Comparisons & Data Visualization
#Created box plots that show diabetes progression scores in correlation to Sex, Hypertension, Smoking status, and Insurance type


fig, axes = plt.subplots(2, 2, figsize=(12,8))

sns.boxplot(x='sex', y='target', data=df_clean, ax=axes[0,0])
axes[0,0].set_title('Diabetes Progression by Sex')
axes[0,0].set_ylabel('Diabetes Progression Score')

sns.boxplot(x='hypertension', y='target', data=df_clean, ax=axes[0,1])
axes[0,1].set_title('Diabetes Progression by Hypertension Status')
axes[0,1].set_ylabel('Diabetes Progression Score')

sns.boxplot(x='smoking_status', y='target', data=df_clean, ax=axes[1,0])
axes[1,0].set_title('Diabetes Progression by Smoking Status')
axes[1,0].set_ylabel('Diabetes Progression Score')

sns.boxplot(x='insurance', y='target', data=df_clean, ax=axes[1,1])
axes[1,1].set_title('Diabetes Progression by Insurance Type')
axes[1,1].set_ylabel('Diabetes Progression Score')

plt.tight_layout()
plt.show()
```



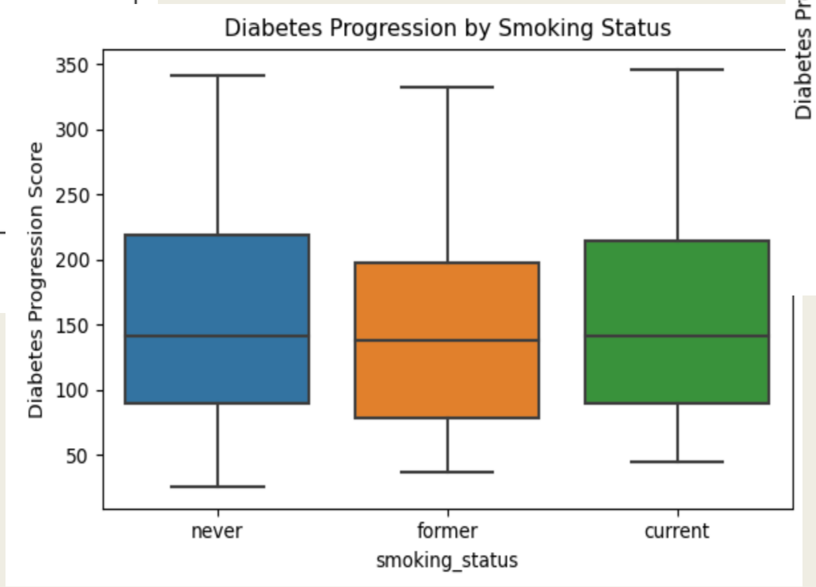
Sex:

- There is minimal difference in diabetes progression between males and females, suggesting that sex alone is not a major determinant.
- However, treatment plans should still consider sex-specific health differences in cardiovascular and hormonal factors.



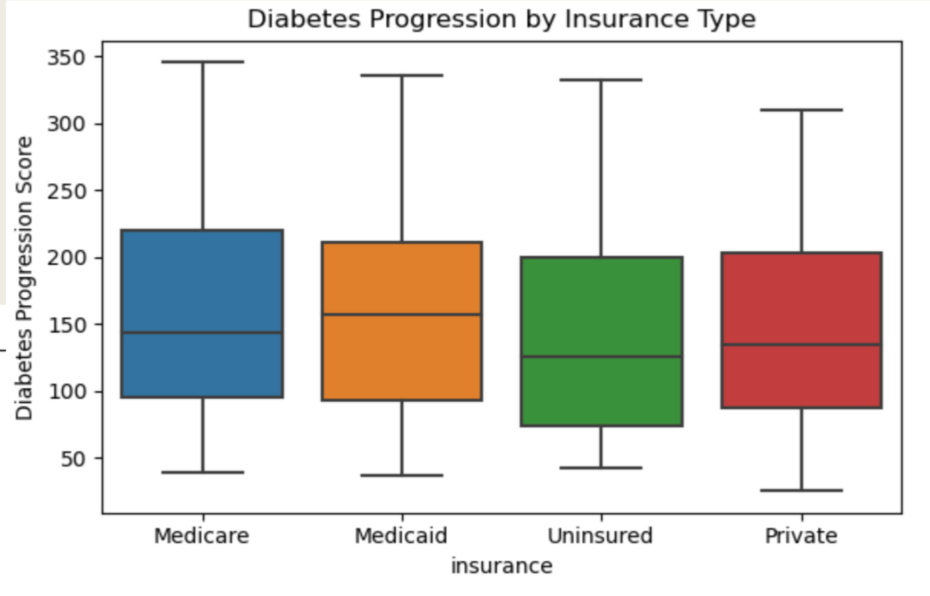
Hypertension:

- Patients with hypertension show higher median progression scores than those without.
- This supports the strong link between cardiovascular strain and worsening diabetes outcomes, emphasizing the importance of blood-pressure control and monitoring.



Smoking Status:

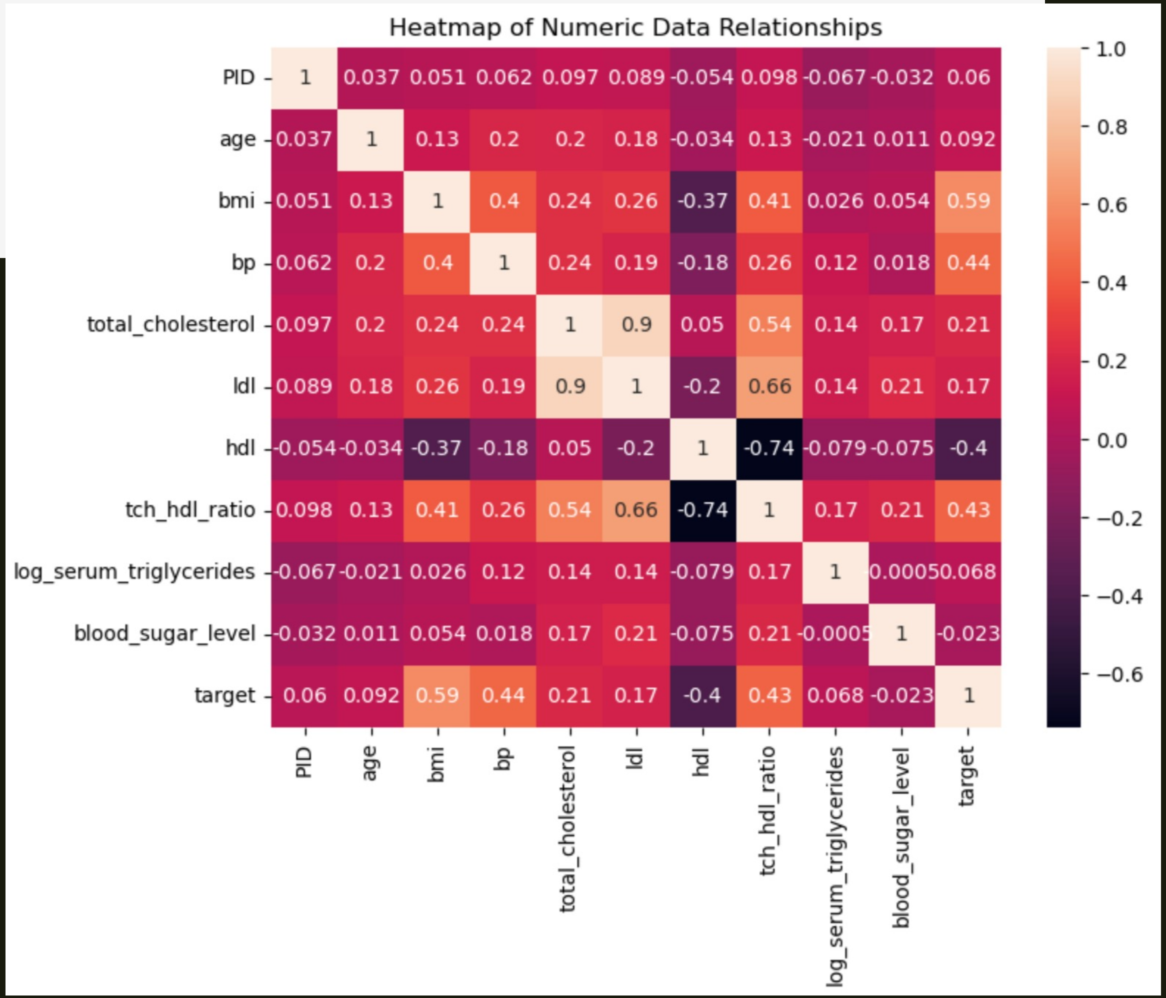
- Current and former smokers display slightly higher diabetes-progression scores than non-smokers.
- Smoking contributes to vascular and metabolic stress, so smoking-cessation counseling remains critical in diabetic care.



Insurance Type:

- Slight variations appear across insurance groups, which may reflect differences in healthcare access, socioeconomic status, or preventive care utilization.
- Improving equitable access to chronic-disease management programs can reduce disparities in outcomes.

```python
#Obj 3: Created a heat map using Seaborn
#Displayed the correlation of all variables being measured in a colored heat map
#The higher the number, the stronger the correlation. A negative number indicates a low correlation & healthier preferred outcome
import seaborn as sns
import matplotlib.pyplot as plt
correlation = df_clean.select_dtypes(include='number').corr()
plt.figure(figsize=(8,6))
sns.heatmap(correlation, annot=True)
plt.title("Heatmap of Numeric Data Relationships")
plt.show()
```



Clinical Application:

- The heatmap displays the strength and direction of relationships between all numeric variables.
- Larger positive values indicate stronger associations with higher diabetes-progression scores, while negative values suggest protective or healthier patterns.

Clinical Application Continued…

Key Observations:

- BMI, blood pressure, and cholesterol ratio (TCH/HDL) show the highest positive correlations with the target, highlighting them as major metabolic risk factors.

- Some lipid measures (HDL) show mild negative correlations, consistent with their protective cardiovascular role.

- Understanding these correlations helps clinicians prioritize interventions on variables that most strongly drive disease progression.

- Clinically, the heatmap provides a quick visual summary of which physiological measures have the greatest influence on diabetes outcomes and can guide preventive and therapeutic focus.

Objective 4: Generate Insights for Clinical Decision Making

The exploratory analysis reveals how multiple clinical and lifestyle factors jointly influence diabetes progression.

Key Findings:

- BMI, blood pressure, and the total-to-HDL cholesterol ratio show the strongest positive correlations with the progression score, confirming that excess body fat, elevated BP, and poor lipid balance accelerate disease severity.
- Hypertensive and smoking patients display higher median progression levels, highlighting the vascular and metabolic stress caused by these risk factors.
- Age shows a mild upward trend, suggesting cumulative metabolic wear over time.

Clinical Application Conclusion

Prioritize

- Prioritize weight, cholesterol, and blood-pressure management for high-risk patients.

Emphasize

- Emphasize smoking-cessation programs and preventive care for middle-aged adults to slow disease advancement.

Use

- Use these insights to support targeted interventions and personalized care planning.