In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import plotly.express as px
```

In [2]:
```
diabetes_data = load_diabetes()
diabetes_data
```

Out[2]:

```
{'data': array([[ 0.03807591,  0.05068012,  0.06169621, ..., -0.00259226,
          0.01990749, -0.01764613],
        [-0.00188202, -0.04464164, -0.05147406, ..., -0.03949338,
         -0.06833155, -0.09220405],
        [ 0.08529891,  0.05068012,  0.04445121, ..., -0.00259226,
          0.00286131, -0.02593034],
        ...,
        [ 0.04170844,  0.05068012, -0.01590626, ..., -0.01107952,
         -0.04688253,  0.01549073],
        [-0.04547248, -0.04464164,  0.03906215, ...,  0.02655962,
          0.04452873, -0.02593034],
        [-0.04547248, -0.04464164, -0.0730303 , ..., -0.03949338,
         -0.00422151,  0.00306441]]),
 'target': array([151.,  75., 141., 206., 135.,  97., 138.,  63., 110., 31
0., 101.,
         69., 179., 185., 118., 171., 166., 144.,  97., 168.,  68.,  49.,
         68., 245., 184., 202., 137.,  85., 131., 283., 129.,  59., 341.,
         87.,  65., 102., 265., 276., 252.,  90., 100.,  55.,  61.,  92.,
        259.,  53., 190., 142.,  75., 142., 155., 225.,  59., 104., 182.,
        128.,  52.,  37., 170., 170.,  61., 144.,  52., 128.,  71., 163.,
        150.,  97., 160., 178.,  48., 270., 202., 111.,  85.,  42., 170.,
        200., 252., 113., 143.,  51.,  52., 210.,  65., 141.,  55., 134.,
         42., 111.,  98., 164.,  48.,  96.,  90., 162., 150., 279.,  92.,
         83., 128., 102., 302., 198.,  95.,  53., 134., 144., 232.,  81.,
        104.,  59., 246., 297., 258., 229., 275., 281., 179., 200., 200.,
        173., 180.,  84., 121., 161.,  99., 109., 115., 268., 274., 158.,
        107.,  83., 103., 272.,  85., 280., 336., 281., 118., 317., 235.,
         60., 174., 259., 178., 128.,  96., 126., 288.,  88., 292.,  71.,
        197., 186.,  25.,  84.,  96., 195.,  53., 217., 172., 131., 214.,
         59.,  70., 220., 268., 152.,  47.,  74., 295., 101., 151., 127.,
        237., 225.,  81., 151., 107.,  64., 138., 185., 265., 101., 137.,
        143., 141.,  79., 292., 178.,  91., 116.,  86., 122.,  72., 129.,
        142.,  90., 158.,  39., 196., 222., 277.,  99., 196., 202., 155.,
         77., 191.,  70.,  73.,  49.,  65., 263., 248., 296., 214., 185.,
         78.,  93., 252., 150.,  77., 208.,  77., 108., 160.,  53., 220.,
        154., 259.,  90., 246., 124.,  67.,  72., 257., 262., 275., 177.,
         71.,  47., 187., 125.,  78.,  51., 258., 215., 303., 243.,  91.,
        150., 310., 153., 346.,  63.,  89.,  50.,  39., 103., 308., 116.,
        145.,  74.,  45., 115., 264.,  87., 202., 127., 182., 241.,  66.,
         94., 283.,  64., 102., 200., 265.,  94., 230., 181., 156., 233.,
         60., 219.,  80.,  68., 332., 248.,  84., 200.,  55.,  85.,  89.,
         31., 129.,  83., 275.,  65., 198., 236., 253., 124.,  44., 172.,
        114., 142., 109., 180., 144., 163., 147.,  97., 220., 190., 109.,
        191., 122., 230., 242., 248., 249., 192., 131., 237.,  78., 135.,
        244., 199., 270., 164.,  72.,  96., 306.,  91., 214.,  95., 216.,
        263., 178., 113., 200., 139., 139.,  88., 148.,  88., 243.,  71.,
         77., 109., 272.,  60.,  54., 221.,  90., 311., 281., 182., 321.,
         58., 262., 206., 233., 242., 123., 167.,  63., 197.,  71., 168.,
        140., 217., 121., 235., 245.,  40.,  52., 104., 132.,  88.,  69.,
        219.,  72., 201., 110.,  51., 277.,  63., 118.,  69., 273., 258.,
         43., 198., 242., 232., 175.,  93., 168., 275., 293., 281.,  72.,
        140., 189., 181., 209., 136., 261., 113., 131., 174., 257.,  55.,
         84.,  42., 146., 212., 233.,  91., 111., 152., 120.,  67., 310.,
         94., 183.,  66., 173.,  72.,  49.,  64.,  48., 178., 104., 132.,
        220.,  57.]),
 'frame': None,
 'DESCR': '.. _diabetes_dataset:\n\nDiabetes dataset\n----------------\n\n
Ten baseline variables, age, sex, body mass index, average blood\npressur
e, and six blood serum measurements were obtained for each of n =\n442 dia
betes patients, as well as the response of interest, a\nquantitative measu
re of disease progression one year after baseline.\n\n**Data Set Character
```

```
istics:**\n\n  :Number of Instances: 442\n\n  :Number of Attributes: First
10 columns are numeric predictive values\n\n  :Target: Column 11 is a quan
titative measure of disease progression one year after baseline\n\n  :Attr
ibute Information:\n      - age     age in years\n      - sex\n      - bmi
body mass index\n      - bp      average blood pressure\n      - s1      t
c, total serum cholesterol\n      - s2      ldl, low-density lipoproteins
\n      - s3      hdl, high-density lipoproteins\n      - s4      tch, tot
al cholesterol / HDL\n      - s5      ltg, possibly log of serum triglycer
ides level\n      - s6      glu, blood sugar level\n\nNote: Each of these
10 feature variables have been mean centered and scaled by the standard de
viation times the square root of `n_samples` (i.e. the sum of squares of e
ach column totals 1).\n\nSource URL:\nhttps://www4.stat.ncsu.edu/~boos/va
r.select/diabetes.html\n\nFor more information see:\nBradley Efron, Trevor
Hastie, Iain Johnstone and Robert Tibshirani (2004) "Least Angle Regressio
n," Annals of Statistics (with discussion), 407-499.\n(https://web.stanfor
d.edu/~hastie/Papers/LARS/LeastAngle_2002.pdf)\n',
 'feature_names': ['age',
  'sex',
  'bmi',
  'bp',
  's1',
  's2',
  's3',
  's4',
  's5',
  's6'],
 'data_filename': 'diabetes_data_raw.csv.gz',
 'target_filename': 'diabetes_target.csv.gz',
 'data_module': 'sklearn.datasets.data'}
```

In [3]:
```
X = pd.DataFrame(diabetes_data.data, columns=diabetes_data.feature_names)
X
```

Out[3]:

	age	sex	bmi	bp	s1	s2	s3	s4	
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.
...	...	...	...	...	...	...	...	...	
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.

442 rows × 10 columns

In [4]:
```
y = pd.Series(diabetes_data.target, name='Diabetes Progression')
y
```

Out[4]:
```
0      151.0
1       75.0
2      141.0
3      206.0
4      135.0
       ...
437    178.0
438    104.0
439    132.0
440    220.0
441     57.0
Name: Diabetes Progression, Length: 442, dtype: float64
```

In [5]:
```
df = pd.concat([X, y], axis=1)
df
```

Out[5]:

	age	sex	bmi	bp	s1	s2	s3	s4	
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.
...	...	...	...	...	...	...	...	...	
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.

442 rows × 11 columns

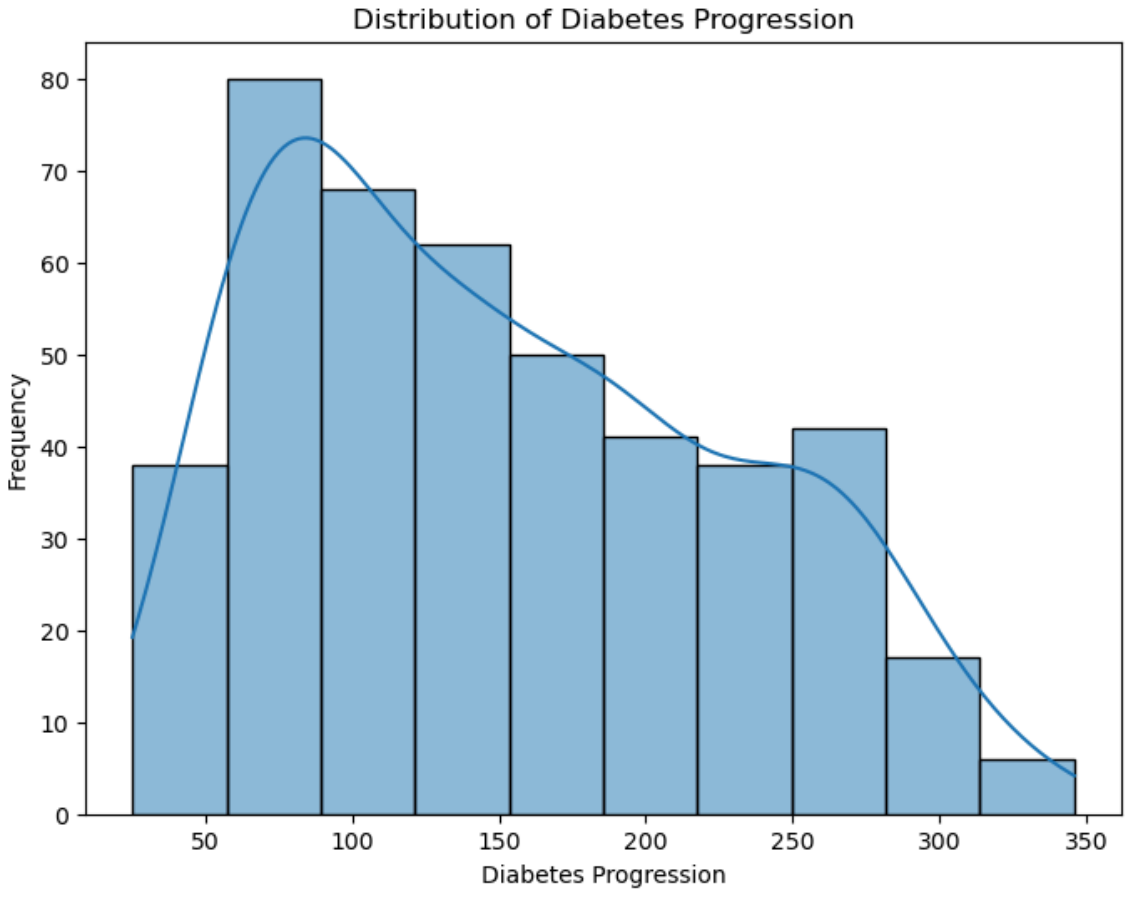
In [6]:
```
print(df.describe())
```

```
                age           sex           bmi            bp            s
1  \
count  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+0
2
mean  -2.511817e-19  1.230790e-17 -2.245564e-16 -4.797570e-17 -1.381499e-1
7
std    4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-0
2
min   -1.072256e-01 -4.464164e-02 -9.027530e-02 -1.123988e-01 -1.267807e-0
1
25%   -3.729927e-02 -4.464164e-02 -3.422907e-02 -3.665608e-02 -3.424784e-0
2
50%    5.383060e-03 -4.464164e-02 -7.283766e-03 -5.670422e-03 -4.320866e-0
3
75%    3.807591e-02  5.068012e-02  3.124802e-02  3.564379e-02  2.835801e-0
2
max    1.107267e-01  5.068012e-02  1.705552e-01  1.320436e-01  1.539137e-0
1

                s2            s3            s4            s5            s
6  \
count  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+0
2
mean   3.918434e-17 -5.777179e-18 -9.042540e-18  9.293722e-17  1.130318e-1
7
std    4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-0
2
min   -1.156131e-01 -1.023071e-01 -7.639450e-02 -1.260971e-01 -1.377672e-0
1
25%   -3.035840e-02 -3.511716e-02 -3.949338e-02 -3.324559e-02 -3.317903e-0
2
50%   -3.819065e-03 -6.584468e-03 -2.592262e-03 -1.947171e-03 -1.077698e-0
3
75%    2.984439e-02  2.931150e-02  3.430886e-02  3.243232e-02  2.791705e-0
2
max    1.987880e-01  1.811791e-01  1.852344e-01  1.335973e-01  1.356118e-0
1

       Diabetes Progression
count            442.000000
mean             152.133484
std               77.093005
min               25.000000
25%               87.000000
50%              140.500000
75%              211.500000
max              346.000000
```

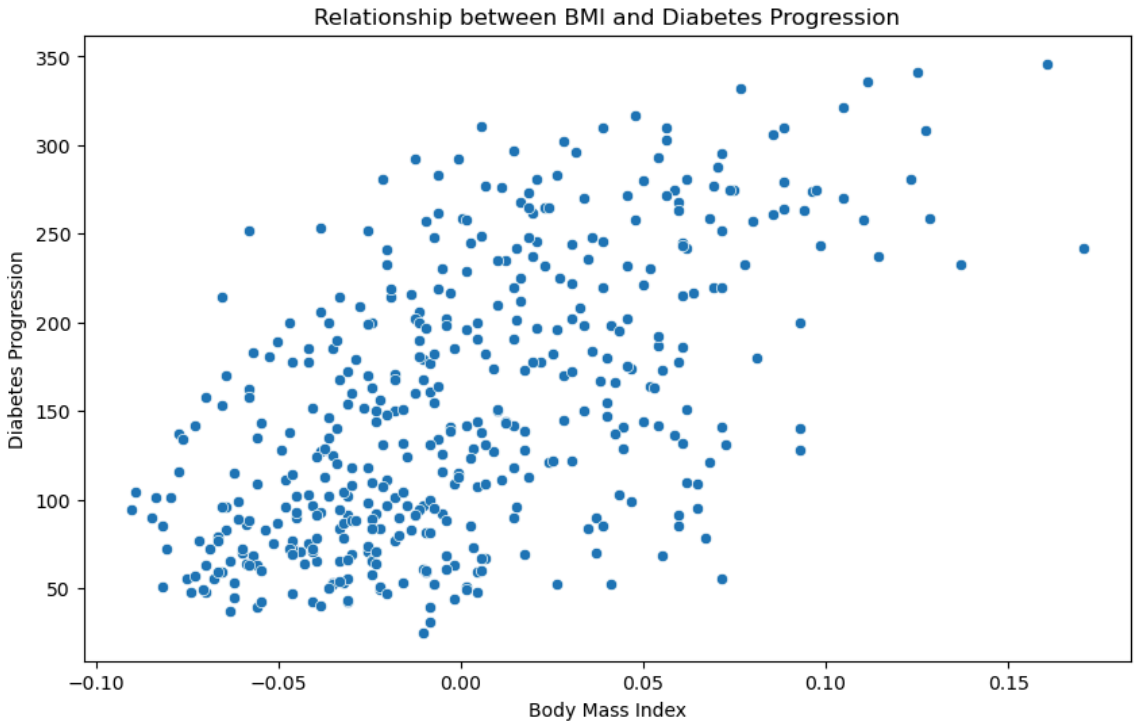
In [7]:
```
print(df.isnull().sum())
```

```
age                      0
sex                      0
bmi                      0
bp                       0
s1                       0
s2                       0
s3                       0
s4                       0
s5                       0
s6                       0
Diabetes Progression     0
dtype: int64
```

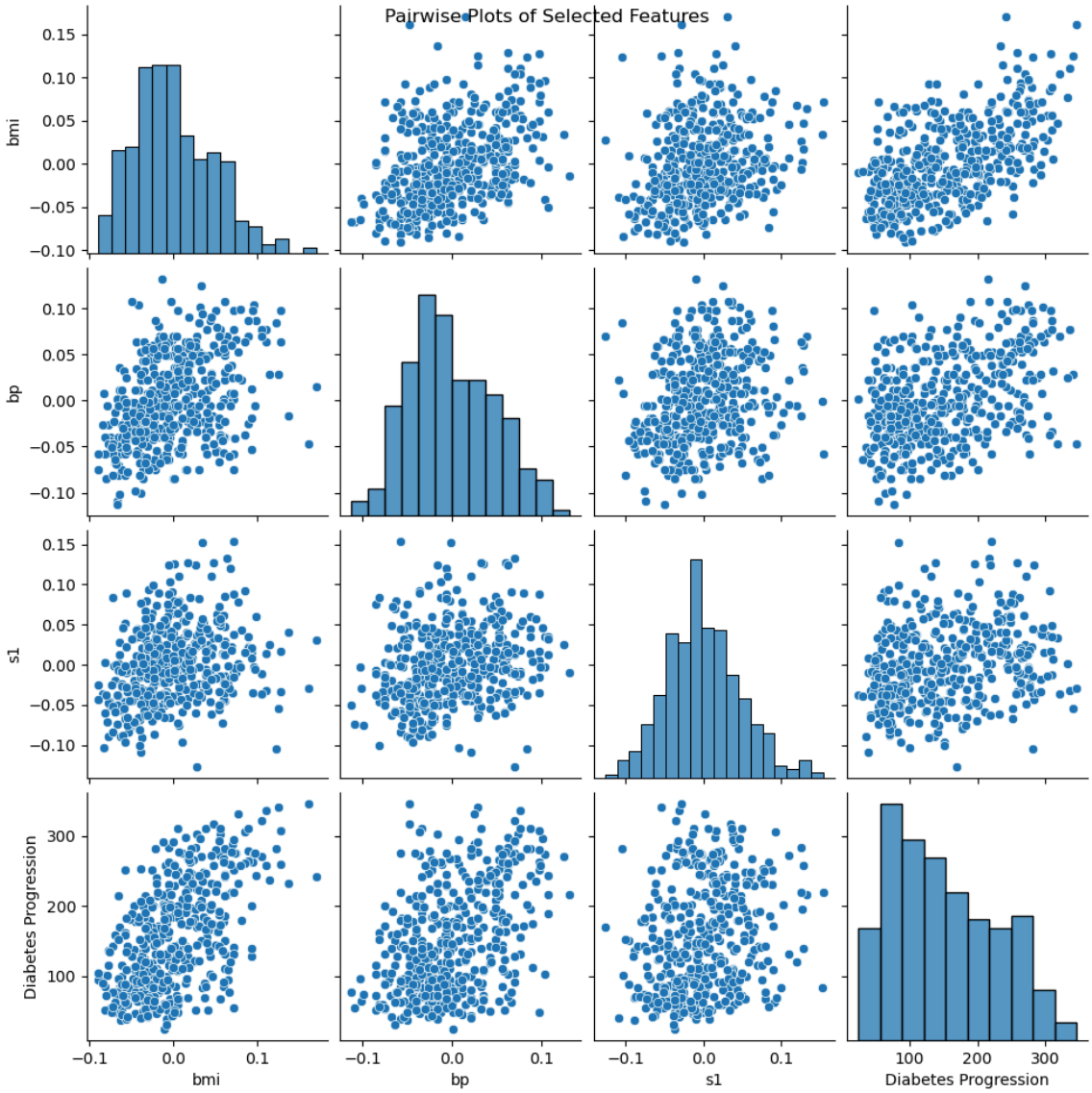
In [8]:
```
plt.figure(figsize=(8, 6))
sns.histplot(df['Diabetes Progression'], kde=True)
plt.title('Distribution of Diabetes Progression')
plt.xlabel('Diabetes Progression')
plt.ylabel('Frequency')
plt.show()
```



In [9]:
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='bmi', y='Diabetes Progression', data=df)
plt.title('Relationship between BMI and Diabetes Progression')
plt.xlabel('Body Mass Index')
plt.ylabel('Diabetes Progression')
plt.show()
```

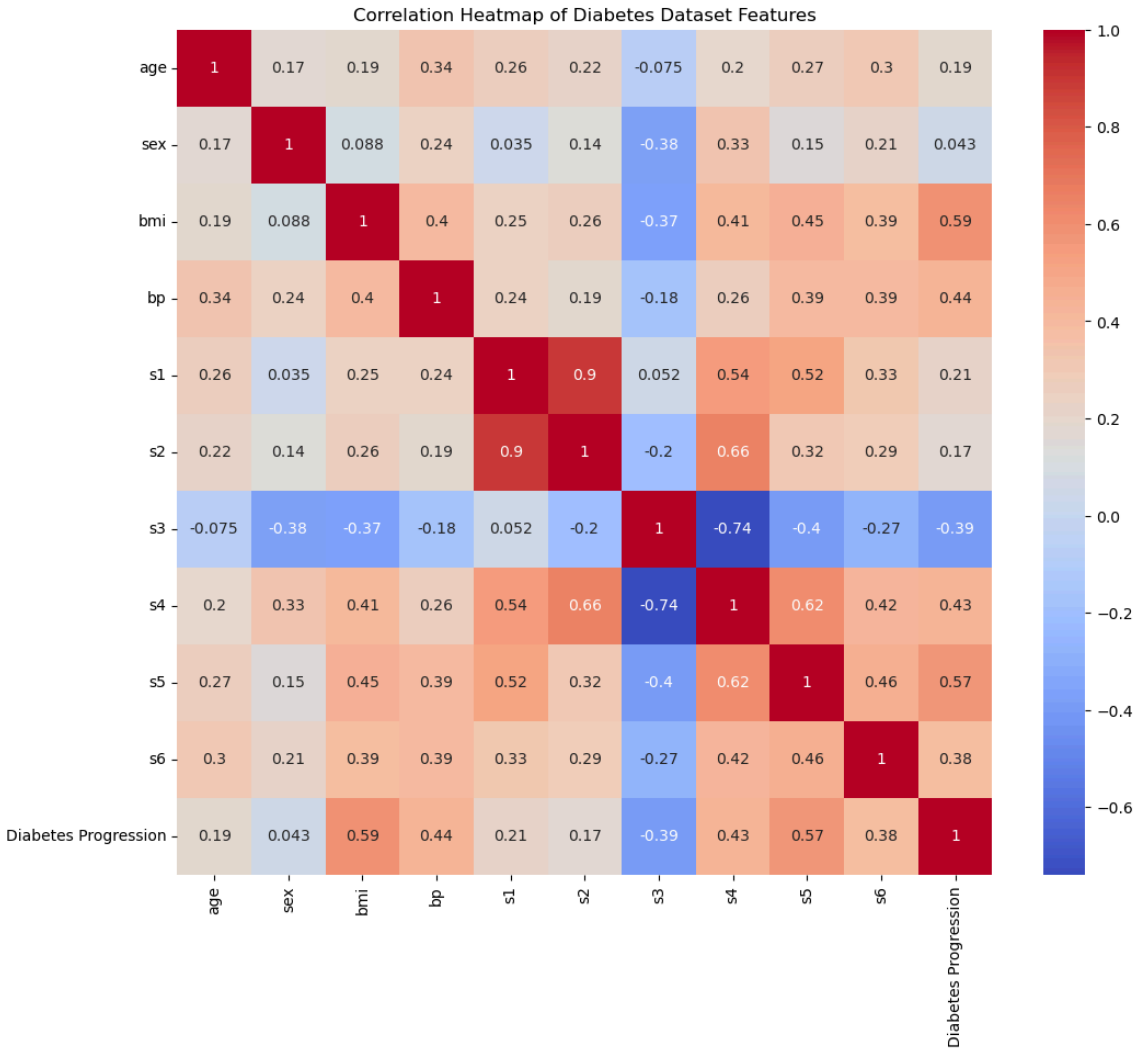


In [10]:
```
sns.pairplot(df[['bmi', 'bp', 's1', 'Diabetes Progression']])
plt.suptitle('Pairwise Plots of Selected Features')
plt.show()
```



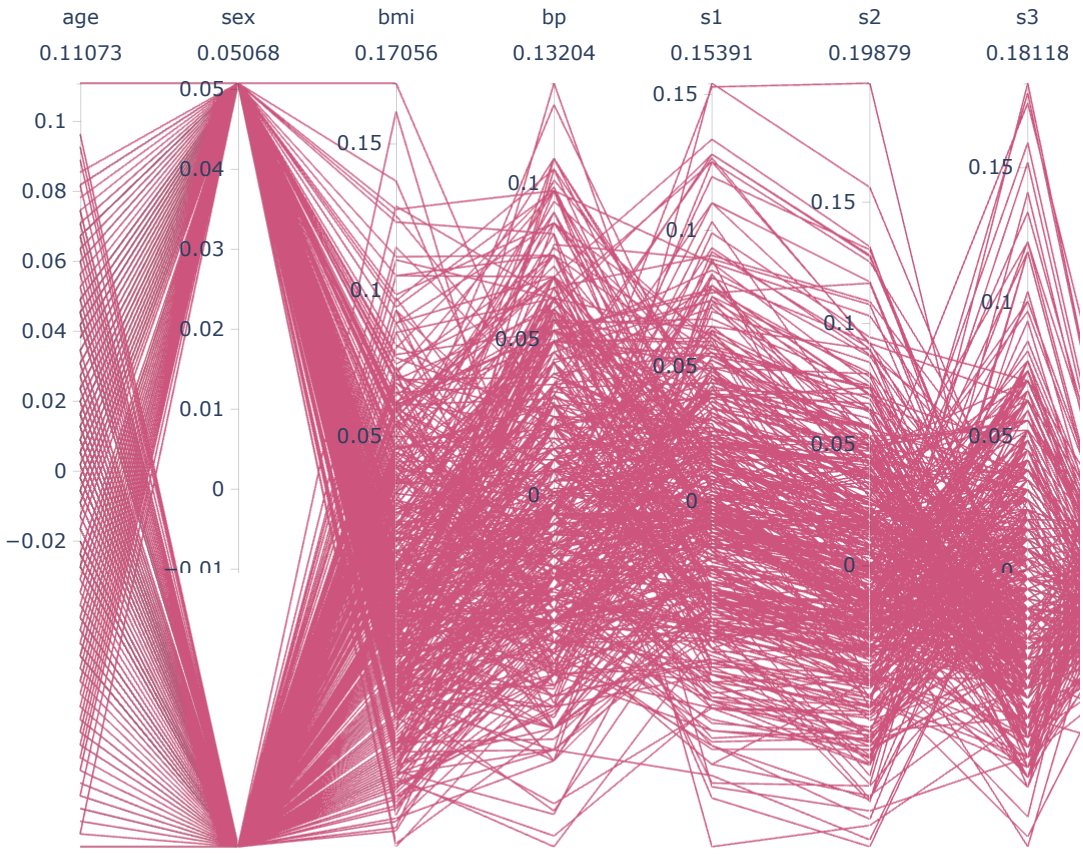
In [11]:

```
plt.figure(figsize=(12, 10))
correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap of Diabetes Dataset Features')
plt.show()
```



In [12]:
```
fig = px.parallel_coordinates(df, color='Diabetes Progression',
                              dimensions=['age', 'sex', 'bmi', 'bp', 's1',
                              color_continuous_scale=px.colors.diverging.Te
                              color_continuous_midpoint=np.average(df['Diab
fig.show()
```

```
C:\Users\Aditya Kudva\anaconda3\Lib\site-packages\plotly\express\_core.py:
279: FutureWarning: iteritems is deprecated and will be removed in a futur
e version. Use .items instead.
  dims = [
```



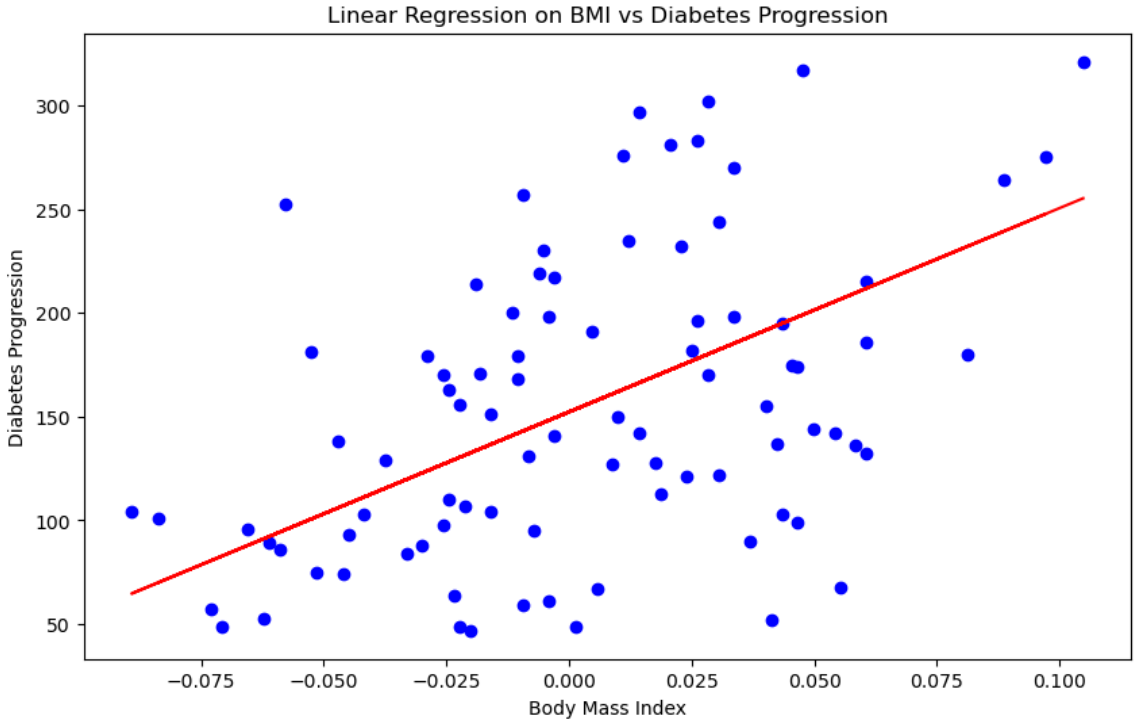
In [13]:
```
X_train, X_test, y_train, y_test = train_test_split(df[['bmi']], df['Diabet
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[13]:
```
▾ LinearRegression
LinearRegression()
```

In [14]:
```
y_pred = model.predict(X_test)
print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
print('Coefficient of Determination:', r2_score(y_test, y_pred))
```

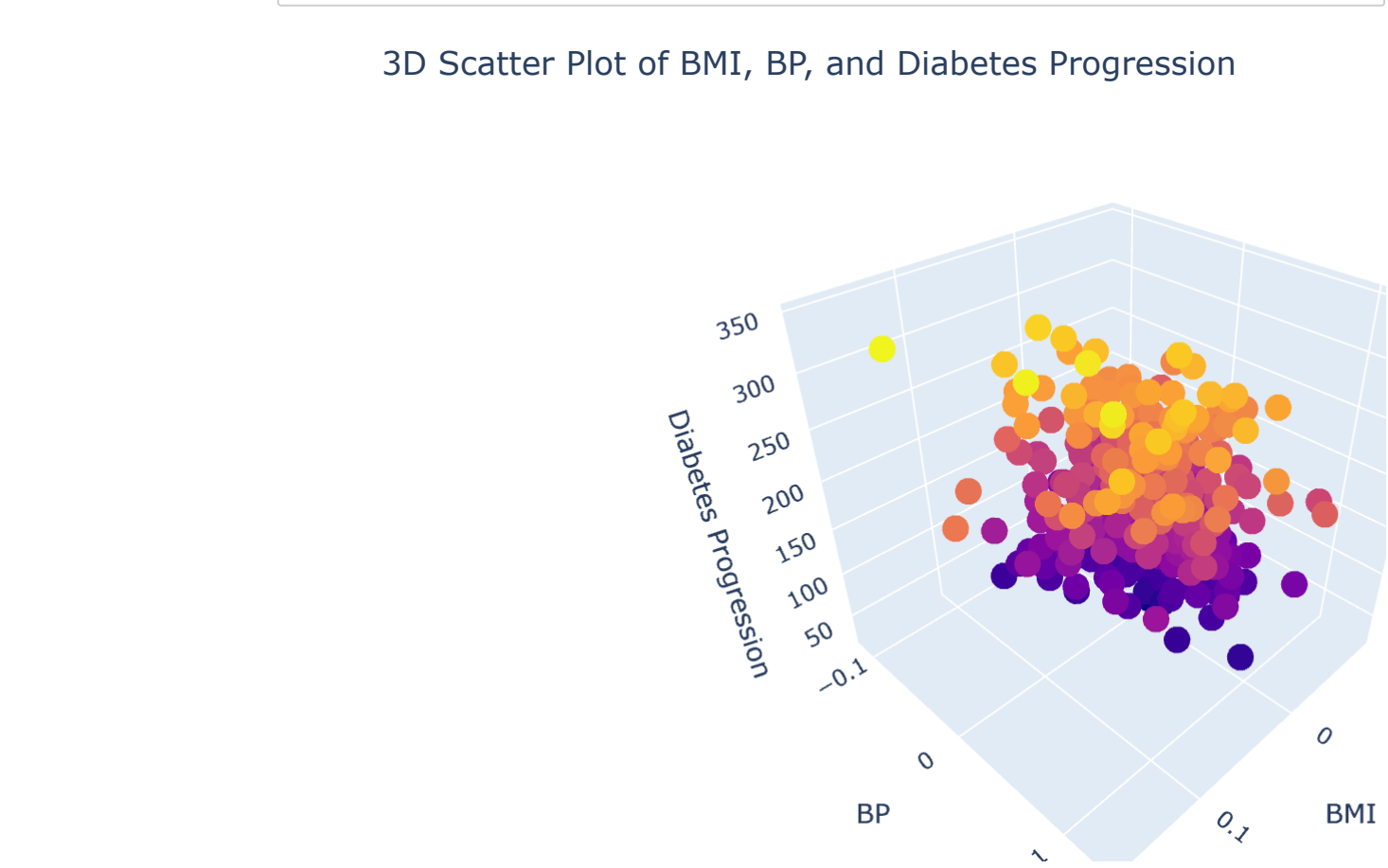
```
Mean Squared Error: 4150.6801893299835
Coefficient of Determination: 0.19057346847560142
```

In [15]:
```
plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='blue')
plt.plot(X_test, y_pred, color='red')
plt.title('Linear Regression on BMI vs Diabetes Progression')
plt.xlabel('Body Mass Index')
plt.ylabel('Diabetes Progression')
plt.show()
```

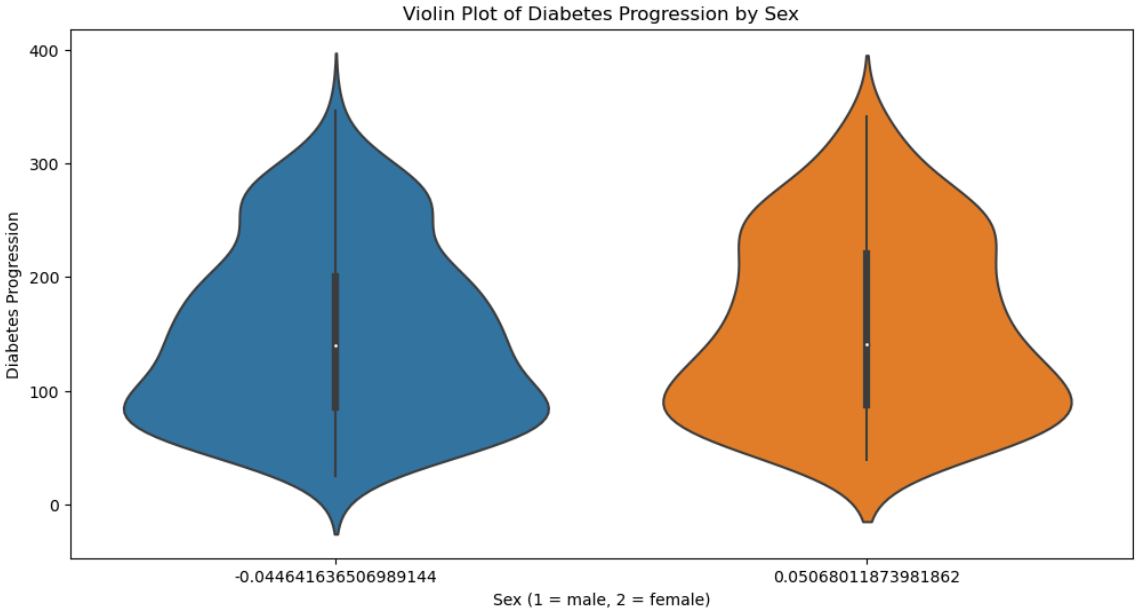


In [16]:
```
import plotly.express as px
```

In [17]:
```
fig = px.scatter_3d(df, x='bmi', y='bp', z='Diabetes Progression', color='D
fig.update_layout(title='3D Scatter Plot of BMI, BP, and Diabetes Progressi
                  scene=dict(xaxis_title='BMI',
                             yaxis_title='BP',
                             zaxis_title='Diabetes Progression'))
fig.show()
```



In [18]:
```
plt.figure(figsize=(12, 6))
sns.violinplot(x='sex', y='Diabetes Progression', data=df)
plt.title('Violin Plot of Diabetes Progression by Sex')
plt.xlabel('Sex (1 = male, 2 = female)')
plt.ylabel('Diabetes Progression')
plt.show()
```

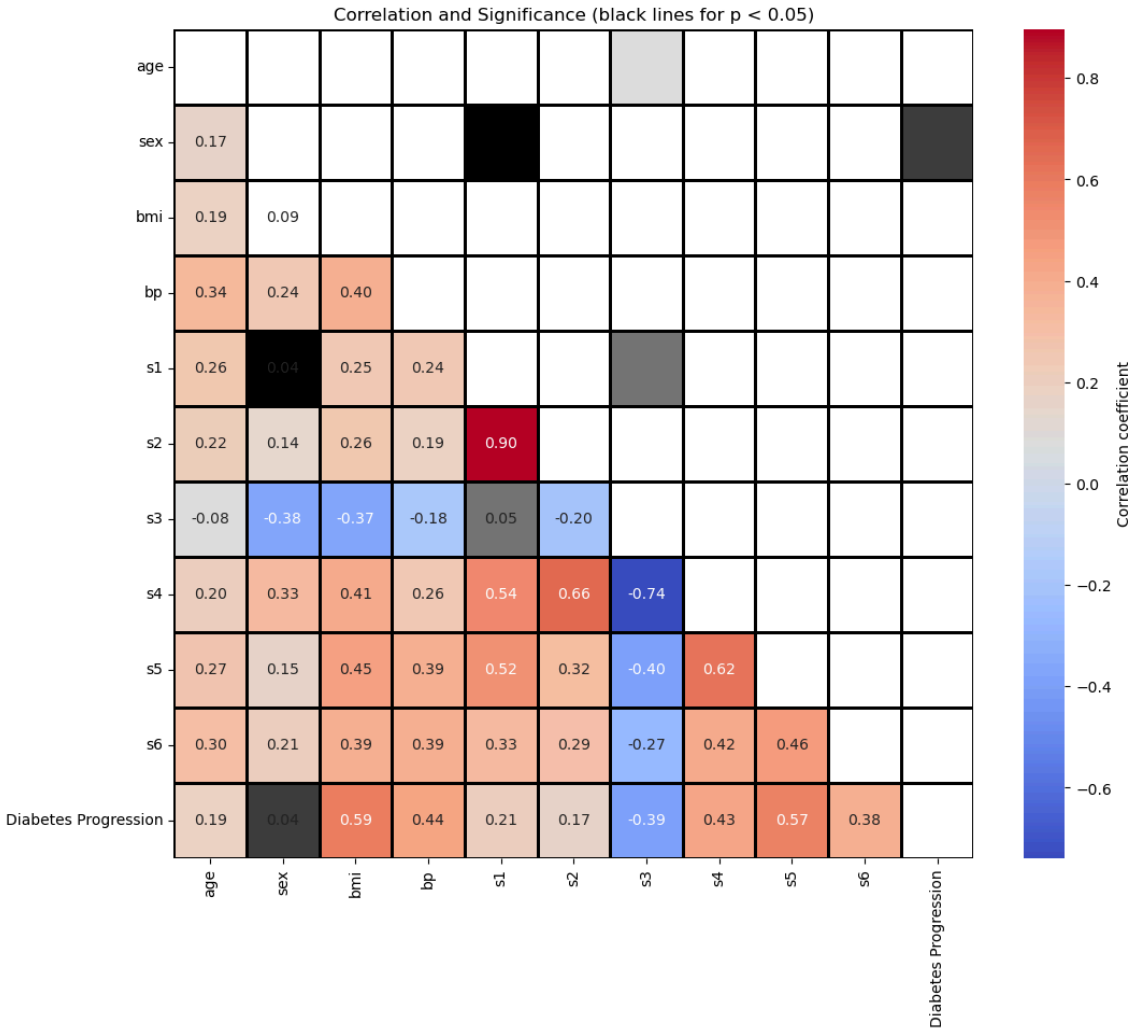


In [19]:
```
from scipy.stats import pearsonr
```

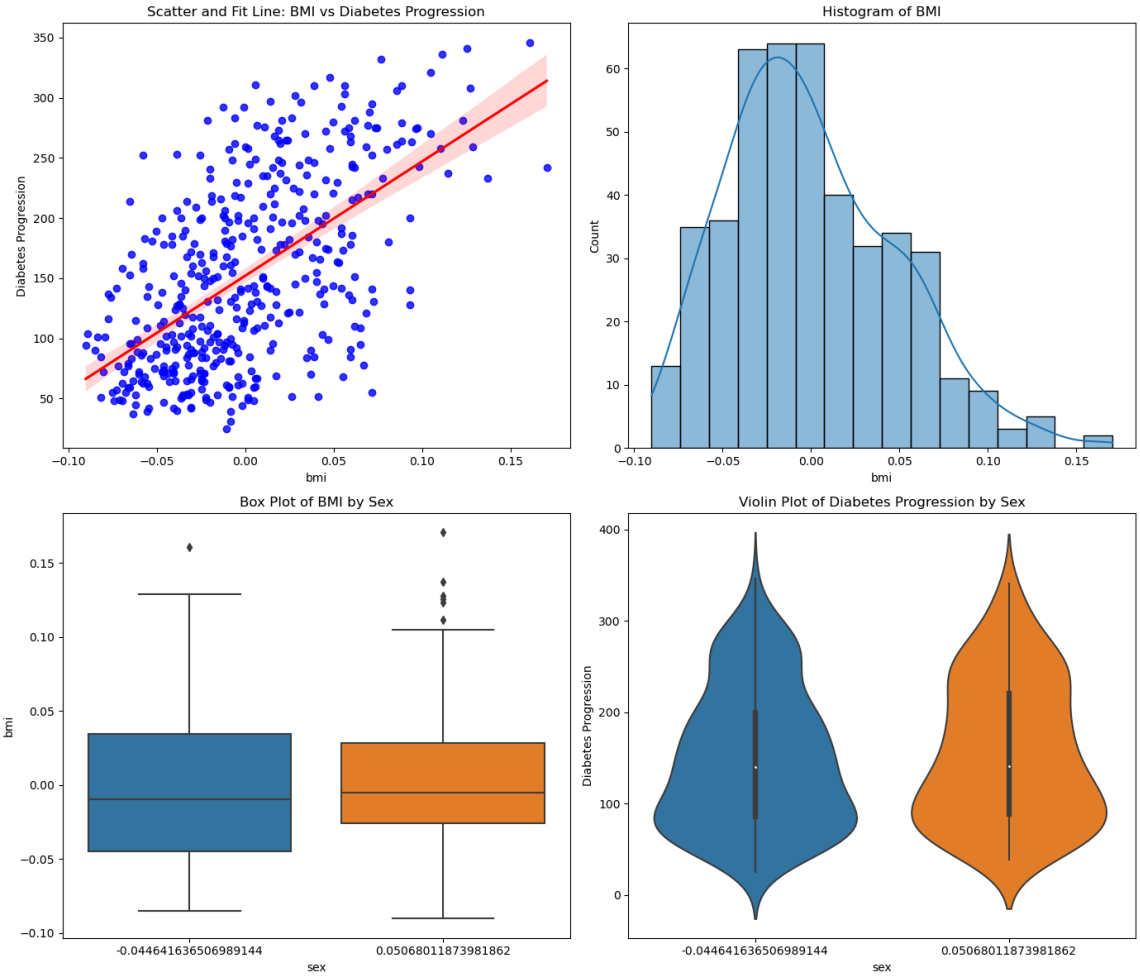
In [20]:
```
corr = df.corr()
pval = pd.DataFrame([[pearsonr(df[col], df[col2])[1] for col2 in df.columns
```

In [21]:
```
mask = np.triu(np.ones_like(corr, dtype=bool))
corr[mask] = np.nan
```

In [22]:
```
plt.figure(figsize=(12, 10))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm', cbar_kws={'label'
sns.heatmap(pval, mask=pval < 0.05, annot=False, cbar=False, cmap='binary',
plt.title('Correlation and Significance (black lines for p < 0.05)')
plt.show()
```



In [28]:
```
fig, axes = plt.subplots(2, 2, figsize=(14, 12))
sns.regplot(ax=axes[0, 0], x='bmi', y='Diabetes Progression', data=df, colo
axes[0, 0].set_title('Scatter and Fit Line: BMI vs Diabetes Progression')

sns.histplot(ax=axes[0, 1], data=df, x='bmi', kde=True)
axes[0, 1].set_title('Histogram of BMI')

sns.boxplot(ax=axes[1, 0], x='sex', y='bmi', data=df)
axes[1, 0].set_title('Box Plot of BMI by Sex')

sns.violinplot(ax=axes[1, 1], x='sex', y='Diabetes Progression', data=df)
axes[1, 1].set_title('Violin Plot of Diabetes Progression by Sex')
plt.tight_layout()
plt.show()
```



In [ ]: