In [1]:
```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes
df=load_diabetes(as_frame=True).frame
sns.set_theme(style="whitegrid")
```

In [2]:
```
df
```

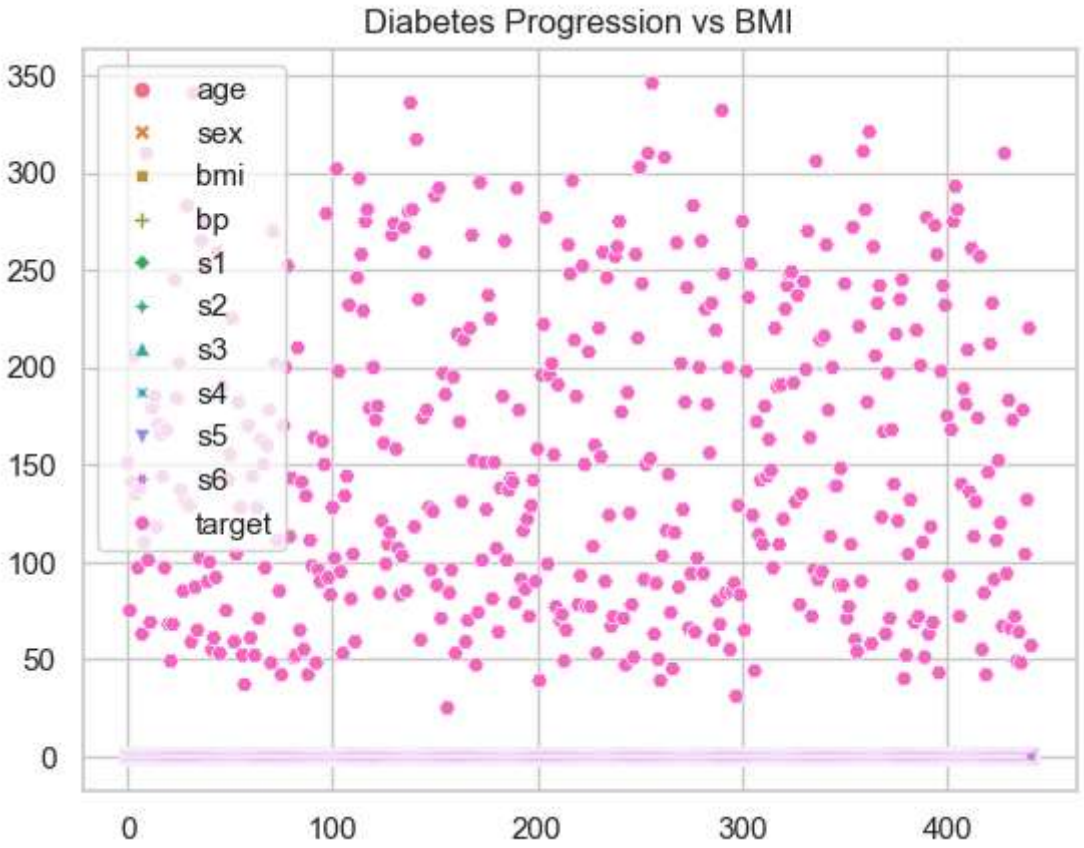
Out[2]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0
...	...	...	...	...	...	...	...	...	...	...	...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018114	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.046883	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044529	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.004222	0.003064	57.0

442 rows × 11 columns

In [3]:
```
sns.scatterplot(data=df)
plt.title("Diabetes Progression vs BMI")
```

Out[3]:
```
Text(0.5, 1.0, 'Diabetes Progression vs BMI')
```



In [4]:
```
X = df.drop(columns=["target"])
y = df["target"]
```

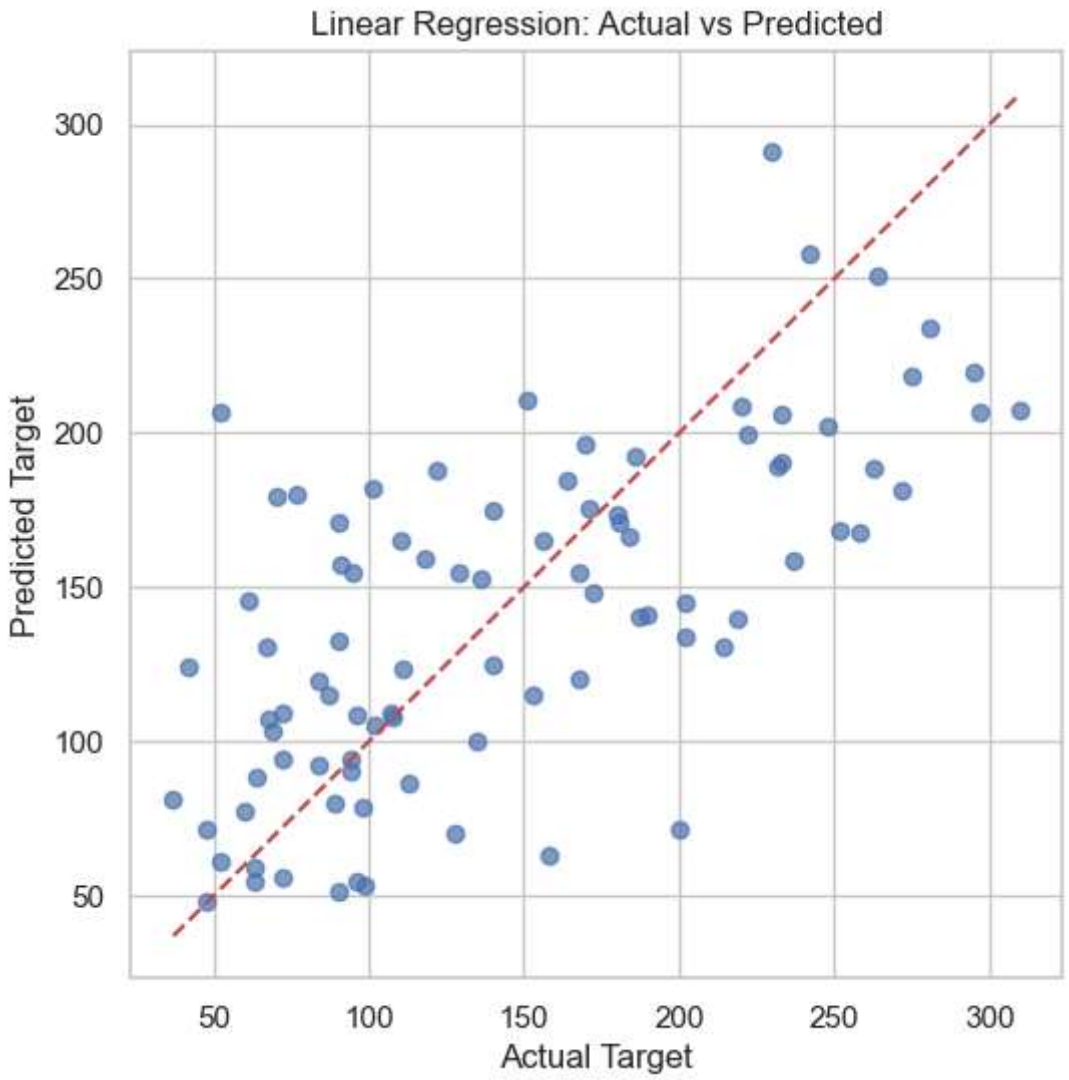
In [5]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Fit a linear regression model
lr = LinearRegression()
lr.fit(X_train, y_train)

# Predict on the test set
y_pred_lr = lr.predict(X_test)
```

```
y_pred_train = lr.predict(X_train)
plt.figure(figsize=(6,6))
plt.scatter(y_test, y_pred_lr, alpha=0.7)
plt.xlabel("Actual Target")
plt.ylabel("Predicted Target")
plt.title("Linear Regression: Actual vs Predicted")
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.show()
```



In [6]:
```
X_train
```

Out[6]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
17	0.070769	0.050680	0.012117	0.056301	0.034206	0.049416	-0.039719	0.034309	0.027364	-0.001078
66	-0.009147	0.050680	-0.018062	-0.033213	-0.020832	0.012152	-0.072854	0.071210	0.000272	0.019633
137	0.005383	-0.044642	0.049840	0.097615	-0.015328	-0.016345	-0.006584	-0.002592	0.017036	-0.013504
245	-0.027310	-0.044642	-0.035307	-0.029770	-0.056607	-0.058620	0.030232	-0.039493	-0.049872	-0.129483
31	-0.023677	-0.044642	-0.065486	-0.081413	-0.038720	-0.053610	0.059685	-0.076395	-0.037129	-0.042499
...	...	...	...	...	...	...	...	...	...	...
106	-0.096328	-0.044642	-0.076264	-0.043542	-0.045599	-0.034821	0.008142	-0.039493	-0.059471	-0.083920
270	0.005383	0.050680	0.030440	0.083844	-0.037344	-0.047347	0.015505	-0.039493	0.008641	0.015491
348	0.030811	-0.044642	-0.020218	-0.005670	-0.004321	-0.029497	0.078093	-0.039493	-0.010903	-0.001078
435	-0.012780	-0.044642	-0.023451	-0.040099	-0.016704	0.004636	-0.017629	-0.002592	-0.038460	-0.038357
102	-0.092695	-0.044642	0.028284	-0.015999	0.036958	0.024991	0.056003	-0.039493	-0.005142	-0.001078

353 rows × 10 columns

In [7]:
```
X_test
```

Out[7]:

	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6
287	0.045341	-0.044642	-0.006206	-0.015999	0.125019	0.125198	0.019187	0.034309	0.032432	-0.005220
211	0.092564	-0.044642	0.036907	0.021872	-0.024960	-0.016658	0.000779	-0.039493	-0.022517	-0.021788
72	0.063504	0.050680	-0.004050	-0.012556	0.103003	0.048790	0.056003	-0.002592	0.084492	-0.017646
321	0.096197	-0.044642	0.051996	0.079265	0.054845	0.036577	-0.076536	0.141322	0.098648	0.061054
73	0.012648	0.050680	-0.020218	-0.002228	0.038334	0.053174	-0.006584	0.034309	-0.005142	-0.009362
...	...	...	...	...	...	...	...	...	...	...
255	0.001751	-0.044642	-0.065486	-0.005670	-0.007073	-0.019476	0.041277	-0.039493	-0.003301	0.007207
90	0.012648	-0.044642	-0.025607	-0.040099	-0.030464	-0.045155	0.078093	-0.076395	-0.072133	0.011349
57	-0.027310	-0.044642	-0.063330	-0.050427	-0.089630	-0.104340	0.052322	-0.076395	-0.056153	-0.067351
391	-0.023677	-0.044642	-0.069797	-0.064199	-0.059359	-0.050478	0.019187	-0.039493	-0.089133	-0.050783
24	-0.063635	-0.044642	0.035829	-0.022885	-0.030464	-0.018850	-0.006584	-0.002592	-0.025953	-0.054925

89 rows × 10 columns

In [8]:
```
y_train
```

Out[8]:
```
17     144.0
66     150.0
137    280.0
245    125.0
31      59.0
       ...
106    134.0
270    202.0
348    148.0
435     64.0
102    302.0
Name: target, Length: 353, dtype: float64
```

In [9]:
```
y_test
```

Out[9]:
```
287    219.0
211     70.0
72     202.0
321    230.0
73     111.0
       ...
255    153.0
90      98.0
57      37.0
391     63.0
24     184.0
Name: target, Length: 89, dtype: float64
```

In [10]:
```
y_pred_lr
```

Out[10]:
```
array([139.5475584 , 179.51720835, 134.03875572, 291.41702925,
       123.78965872,  92.1723465 , 258.23238899, 181.33732057,
        90.22411311, 108.63375858,  94.13865744, 168.43486358,
        53.5047888 , 206.63081659, 100.12925869, 130.66657085,
       219.53071499, 250.7803234 , 196.3688346 , 218.57511815,
       207.35050182,  88.48340941,  70.43285917, 188.95914235,
       154.8868162 , 159.36170122, 188.31263363, 180.39094033,
        47.99046561, 108.97453871, 174.77897633,  86.36406656,
       132.95761215, 184.53819483, 173.83220911, 190.35858492,
       124.4156176 , 119.65110656, 147.95168682,  59.05405241,
        71.62331856, 107.68284704, 165.45365458, 155.00975931,
       171.04799096,  61.45761356,  71.66672581, 114.96732206,
        51.57975523, 167.57599528, 152.52291955,  62.95568515,
       103.49741722, 109.20751489, 175.64118426, 154.60296242,
        94.41704366, 210.74209145, 120.2566205 ,  77.61585399,
       187.93203995, 206.49337474, 140.63167076, 105.59678023,
       130.70432536, 202.18534537, 171.13039501, 164.91423047,
       124.72472569, 144.81030894, 181.99635452, 199.41369642,
       234.21436188, 145.95665512,  79.86703276, 157.36941275,
       192.74412541, 208.89814032, 158.58722555, 206.02195855,
       107.47971675, 140.93598906,  54.82129332,  55.92573195,
       115.01180018,  78.95584188,  81.56087285,  54.37997256,
       166.2543518 ])
```

In [11]:
```
y_pred_train
```

Out[11]:
```
array([184.69998932, 151.94648006, 239.23281831, 105.01861779,
        65.03298695,  54.03772648, 197.74802871, 250.61211643,
       184.57999461, 104.66214316,  64.07814485, 193.99383833,
       102.92339577, 295.23784291,  95.42096533, 150.77793462,
       103.69669722, 127.49424297, 123.3788916 , 172.05561705,
       180.24879522, 112.11717223, 207.94276506, 203.78762568,
       212.10045329, 197.01042374, 161.13144688, 147.35024317,
       132.99202695, 112.27831332, 166.52382748, 170.26047458,
        93.75445178,  42.82093557, 165.66549653, 124.76847628,
       184.30282961, 173.08902349, 109.50485097, 199.76313049,
        92.34075672, 118.03501124,  84.46019399, 234.9206077 ,
       234.56256175, 236.69930416, 160.06590015, 290.90186967,
       159.39044745,  74.37415917, 228.85324369,  85.40170112,
       169.90434447, 223.16412242,  95.53607689, 230.76321477,
       210.13539006, 157.79420917, 210.74864257, 135.76171129,
       145.55684501, 169.73653238, 243.09036664, 122.15442383,
       201.6977569 ,  83.75269578, 179.6962925 , 151.21141743,
        90.51118293, 122.74128733, 146.06524912, 152.00250989,
        75.56288804,  57.27797675, 148.44648665, 113.77394655,
       165.97775707, 190.78584816,  81.22046258, 121.26602825,
        94.06267555, 156.58828816,  70.49812787, 119.27445469,
       205.41280513, 256.39780185, 215.10147915, 229.02916335,
       242.06391297, 265.00296509, 130.11643736, 120.34592963,
       185.98410633, 152.40619966,  71.78241048, 183.05957367,
       138.28133819,  81.93161087, 214.12158653, 224.60574814,
       200.53638736,  93.94582351, 147.02815853, 160.15126346,
       123.16650191, 238.76362852, 113.91122229, 263.25814842,
       163.18113869, 193.29011944, 143.68771746, 130.42540742,
       224.41531598, 206.90600558, 150.80035302, 102.15585569,
       124.41132508, 200.30197844,  76.99736974, 225.3585334 ,
       184.05964366, 191.39748854, 140.74000616, 124.59796685,
       104.86045976, 174.5899331 , 171.58241469, 210.13355926,
       115.54877081,  86.04175597, 168.85442819, 125.26665709,
       212.18177425, 151.32011115, 117.02634699, 248.88214255,
       155.20839777, 182.19909117, 221.87606761, 202.7405191 ,
       158.06034559, 173.7069955 , 124.91826403, 230.28876888,
       240.63167058, 267.16400526, 204.91461635, 104.54419151,
       120.24409477, 229.18373641, 184.61749465, 189.24242034,
       117.31331265,  31.36368872, 121.39834677, 115.28967557,
        41.58801343, 190.13408341, 142.41100256, 208.95397997,
       257.77550565, 191.74085383, 185.94529919,  79.27232802,
       158.6421542 , 118.58604398, 175.47685383, 141.0480234 ,
```

```
166.33765238, 147.79865815, 162.85169942, 168.95402202,
127.46797614, 224.18319047, 109.40386589, 162.45065423,
147.37072022, 176.00931551, 146.54774958,  52.6586183 ,
194.12716048, 121.52590858, 156.61906201, 186.11816789,
157.99950135,  73.22954369,  93.51007662,  98.2983188 ,
106.9413408 , 158.0759558 , 160.66899498,  93.19395765,
 69.72881283, 283.47862324, 228.66922192,  49.03387099,
190.26879892,  84.30745774, 169.5162796 ,  85.5706787 ,
196.32979324, 234.19710309, 114.68482082,  82.58360969,
129.45297331,  48.46744384, 166.28086909,  63.01671673,
204.8324854 , 138.07409793, 120.02635892, 175.54790863,
240.65679875, 113.18590115, 247.62689683,  99.17192404,
 69.99393919, 141.05455113, 160.24269554, 137.72314427,
251.71290817, 139.66049175, 129.4324169 , 270.97704204,
 61.71608548, 162.68126871,  73.49318321, 174.2705011 ,
 93.99482918,  87.88350448, 121.81254876,  95.02705257,
265.06160932, 182.72287534, 107.14770792, 276.43453558,
134.07020103,  91.56130255, 257.90974845, 178.17821766,
112.64888886, 154.15135402,  59.33291068,  81.37650887,
142.9654618 , 180.43555317, 259.62047692,  60.38891311,
241.22847129,  87.85690013, 160.53586083, 158.27440612,
116.9856396 , 216.50000881, 104.35638379, 146.14473367,
148.38501902, 295.69543639, 200.37666801, 229.3504969 ,
118.33354037, 243.84710261, 178.53021584, 184.17004149,
207.84996826, 150.40200043, 146.16627478, 248.29904572,
274.27085642, 126.5795494 , 201.76225333, 229.71733101,
245.0880614 ,  85.33977397,  64.65217704, 223.53534   ,
 71.77973494, 221.67678829, 198.53204157, 129.31456454,
142.38033718, 117.00362196, 157.37976123,  69.43671116,
159.71905127, 191.34189865, 173.71023213, 227.52530879,
 80.55190897,  40.71485587, 251.5950736 , 184.24825534,
138.27758136, 128.86724533, 126.71133702, 127.74392637,
117.78610944, 163.37767035, 104.36744998,  75.20391982,
 94.28800854, 188.66574257,  61.34285855, 188.48415471,
129.19313991, 137.31565635, 127.3071938 , 172.57664836,
193.50076601, 104.51001942, 134.57489622, 164.87113236,
159.82129154, 166.36445482, 115.74360663,  95.36791596,
 66.80634536, 212.76614307, 105.43879585,  74.67830709,
136.15406241, 183.41094157, 181.96307761,  82.95531169,
216.4841103 ,  78.46053352, 180.78448524, 127.05103485,
144.33762513, 177.29697474, 115.84012326, 156.92704922,
102.39936531,  90.88912594, 178.44895738, 243.94752494,
```

```
       143.98944582, 176.85593277, 138.30485426, 108.28467284,
       148.75088581, 184.82620244, 103.04383528, 204.07177882,
        43.47908283, 115.60754303, 150.59838224, 234.81321832,
        69.0151309 , 193.98351146, 132.92972967, 119.18604409,
       141.37461393])
```

In [12]:
```
df.fillna = df.fillna(df.mean())
# Check for missing values
missing_values = df.isnull().sum()
print("Missing values in each column:\n", missing_values)
```

```
Missing values in each column:
 age       0
sex       0
bmi       0
bp        0
s1        0
s2        0
s3        0
s4        0
s5        0
s6        0
target    0
dtype: int64
```

In [13]:
```
df["target"].describe()
```

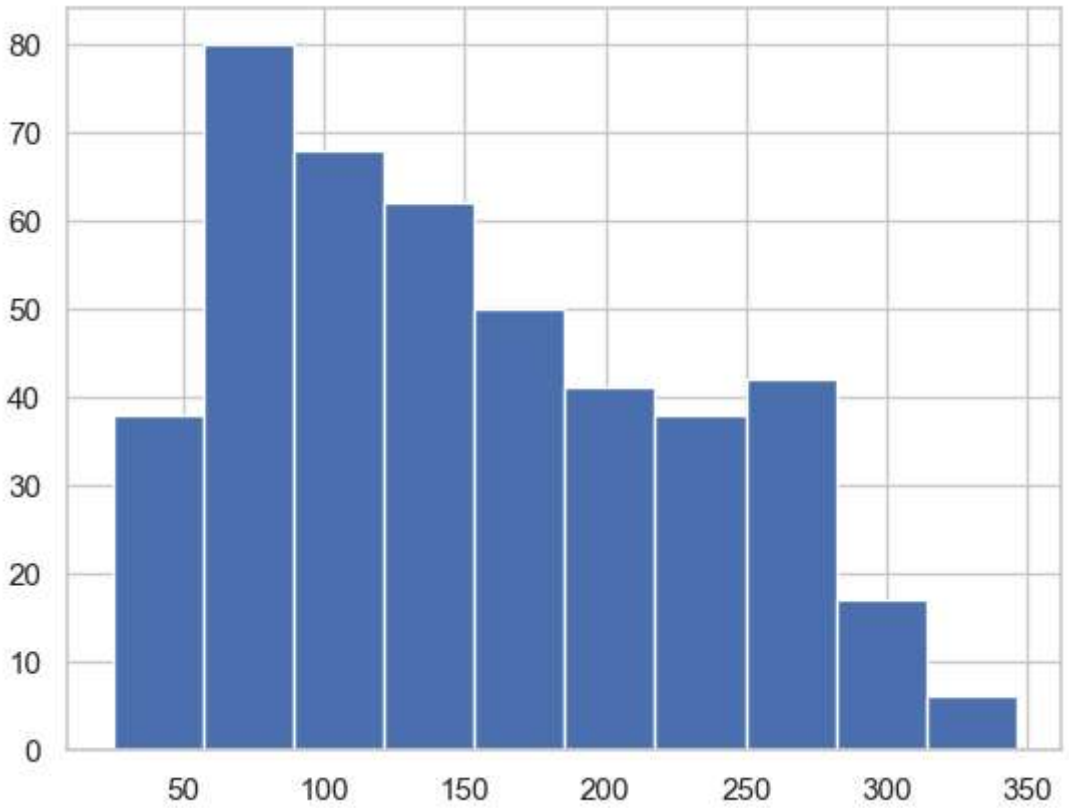
Out[13]:
```
count    442.000000
mean     152.133484
std       77.093005
min       25.000000
25%       87.000000
50%      140.500000
75%      211.500000
max      346.000000
Name: target, dtype: float64
```

In [14]:
```
df.nunique()
```

Out[14]:
```
age        58
sex         2
bmi       163
bp        100
s1        141
s2        302
s3         63
s4         66
s5        184
s6         56
target    214
dtype: int64
```

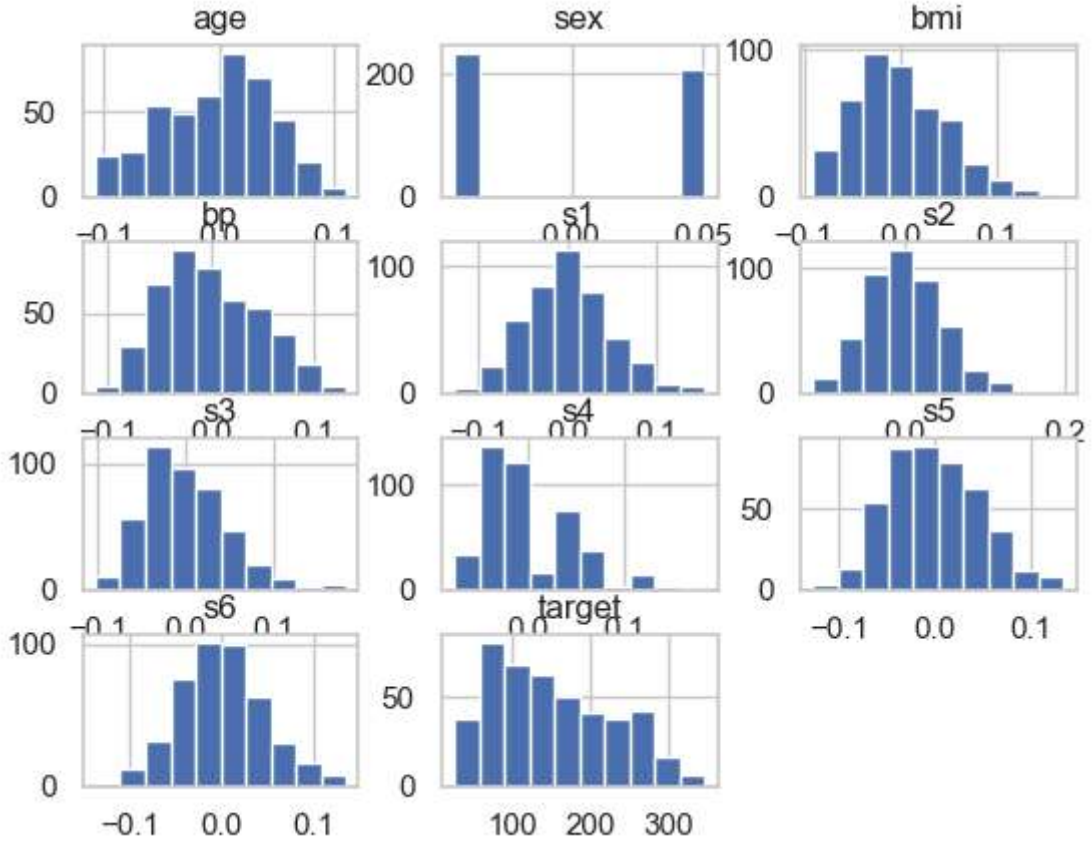
In [15]:
```
df['target'].hist()
```

Out[15]: <Axes: >



In [16]:
```
df.hist()
```

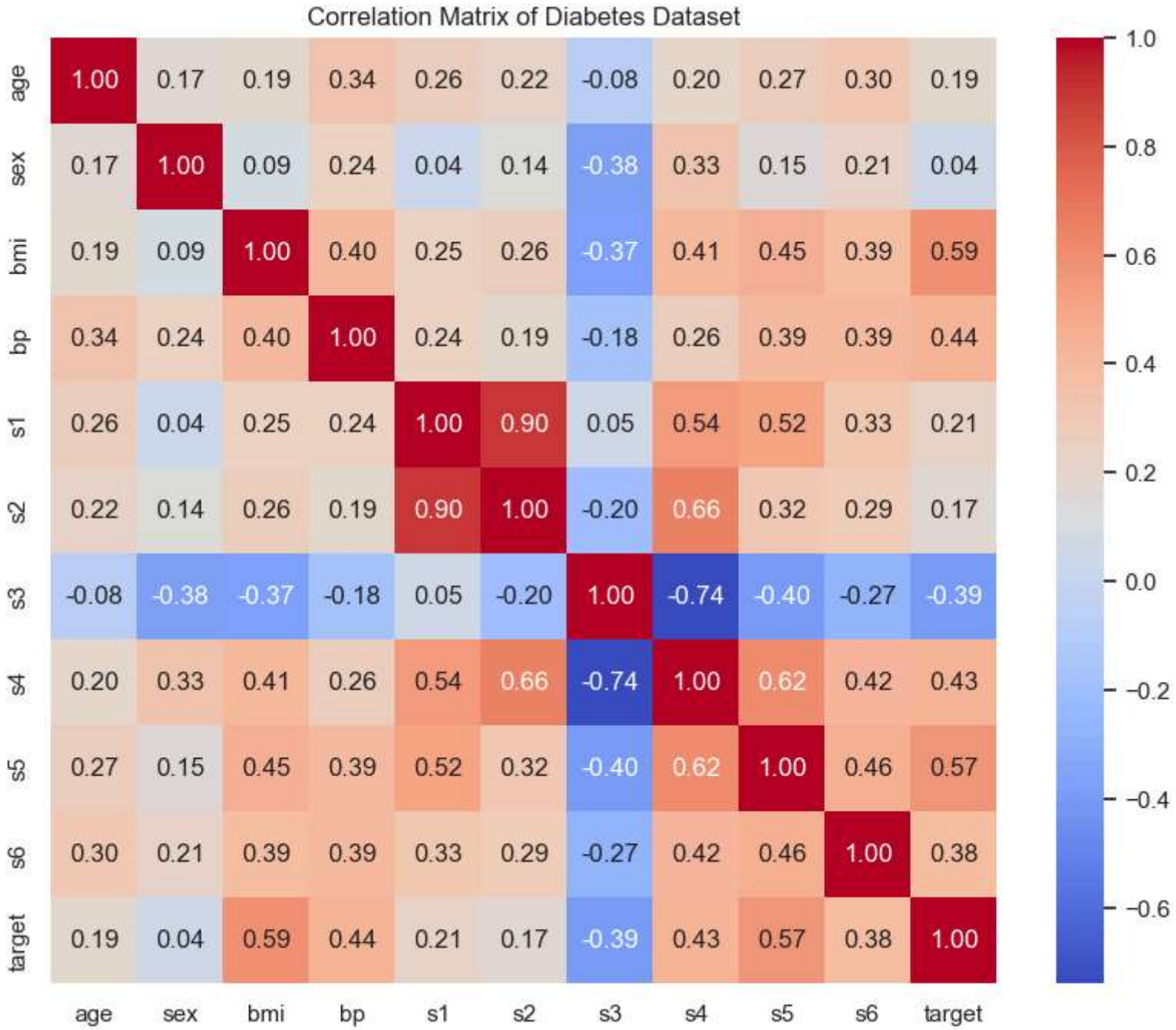
Out[16]:
```
array([[<Axes: title={'center': 'age'}>, <Axes: title={'center': 'sex'}>,
        <Axes: title={'center': 'bmi'}>],
       [<Axes: title={'center': 'bp'}>, <Axes: title={'center': 's1'}>,
        <Axes: title={'center': 's2'}>],
       [<Axes: title={'center': 's3'}>, <Axes: title={'center': 's4'}>,
        <Axes: title={'center': 's5'}>],
       [<Axes: title={'center': 's6'}>,
        <Axes: title={'center': 'target'}>, <Axes: >]], dtype=object)
```



In [17]:
```
df.corr()
# Visualize the correlation matrix
plt.figure(figsize=(10, 8))
```

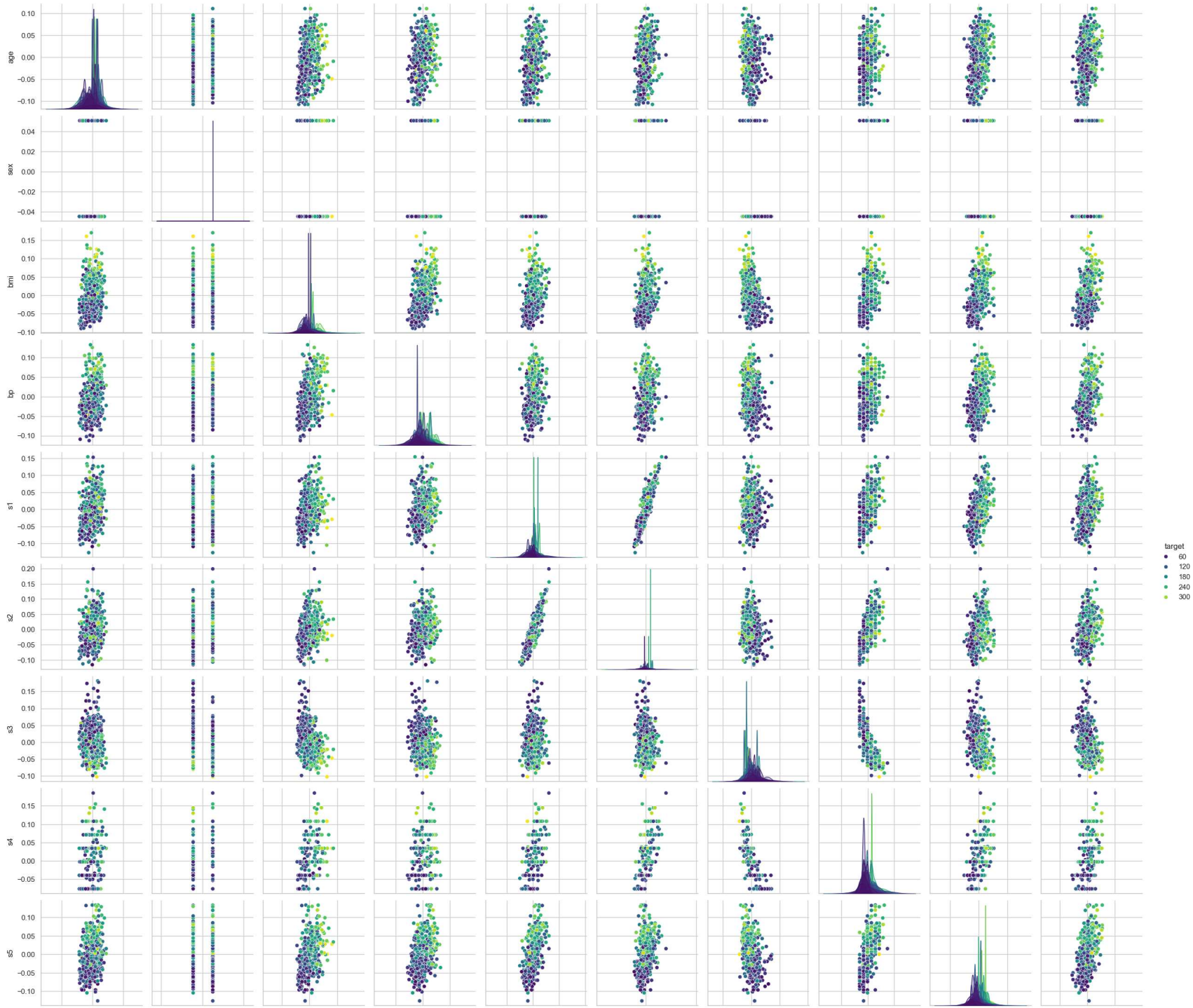
```
sns.heatmap(df.corr(), annot=True, fmt=".2f", cmap="coolwarm", square=True)
plt.title("Correlation Matrix of Diabetes Dataset")
```

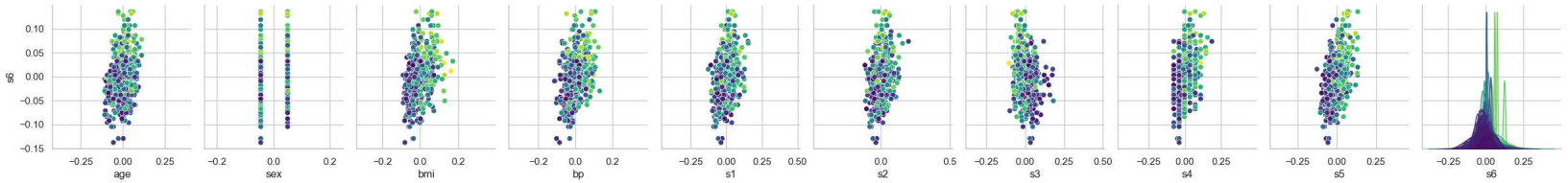
Out[17]: Text(0.5, 1.0, 'Correlation Matrix of Diabetes Dataset')

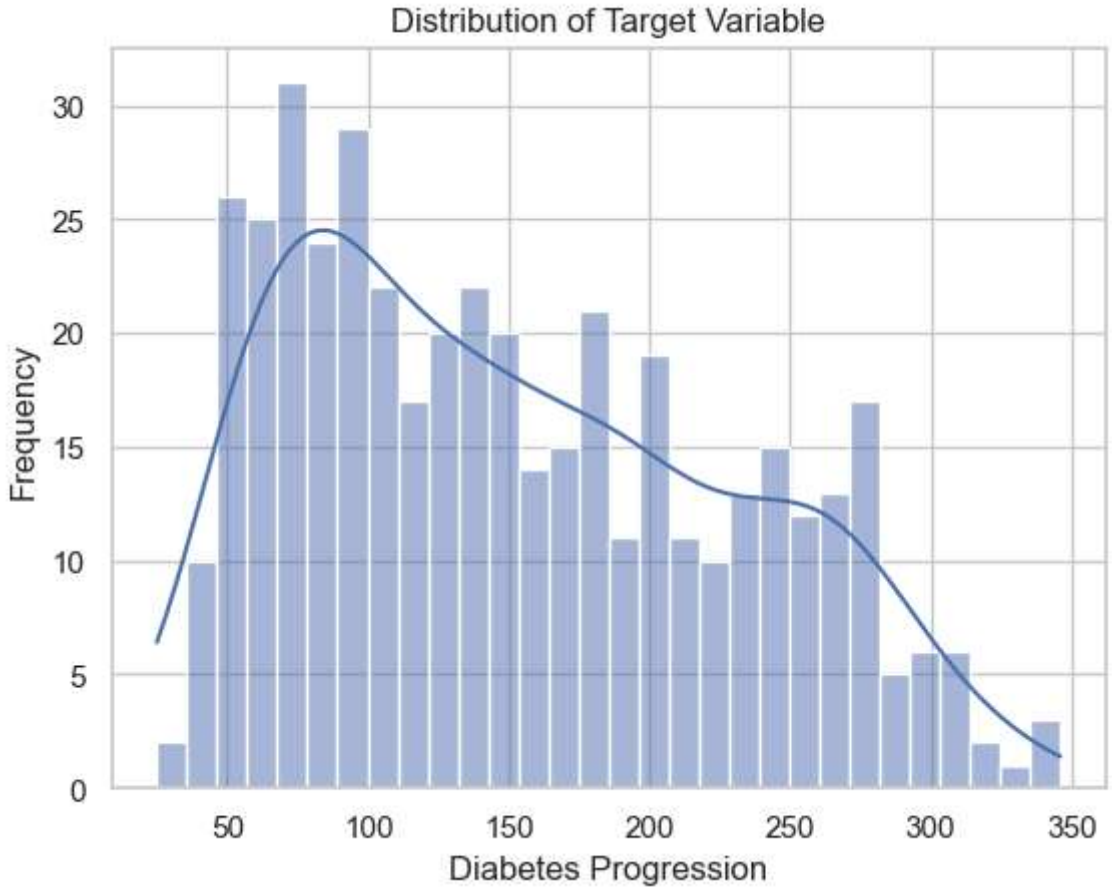

Correlation Matrix of Diabetes Dataset
age 1.00 0.17 0.19 0.34 0.26 0.22 -0.08 0.20 0.27 0.30 0.19
sex 0.17 1.00 0.09 0.24 0.04 0.14 -0.38 0.33 0.15 0.21 0.04
bmi 0.19 0.09 1.00 0.40 0.25 0.26 -0.37 0.41 0.45 0.39 0.59
bp 0.34 0.24 0.40 1.00 0.24 0.19 -0.18 0.26 0.39 0.39 0.44
s1 0.26 0.04 0.25 0.24 1.00 0.90 0.05 0.54 0.52 0.33 0.21
s2 0.22 0.14 0.26 0.19 0.90 1.00 -0.20 0.66 0.32 0.29 0.17
s3 -0.08 -0.38 -0.37 -0.18 0.05 -0.20 1.00 -0.74 -0.40 -0.27 -0.39
s4 0.20 0.33 0.41 0.26 0.54 0.66 -0.74 1.00 0.62 0.42 0.43
s5 0.27 0.15 0.45 0.39 0.52 0.32 -0.40 0.62 1.00 0.46 0.57
s6 0.30 0.21 0.39 0.39 0.33 0.29 -0.27 0.42 0.46 1.00 0.38
target 0.19 0.04 0.59 0.44 0.21 0.17 -0.39 0.43 0.57 0.38 1.00
age sex bmi bp s1 s2 s3 s4 s5 s6 target
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

In [19]:

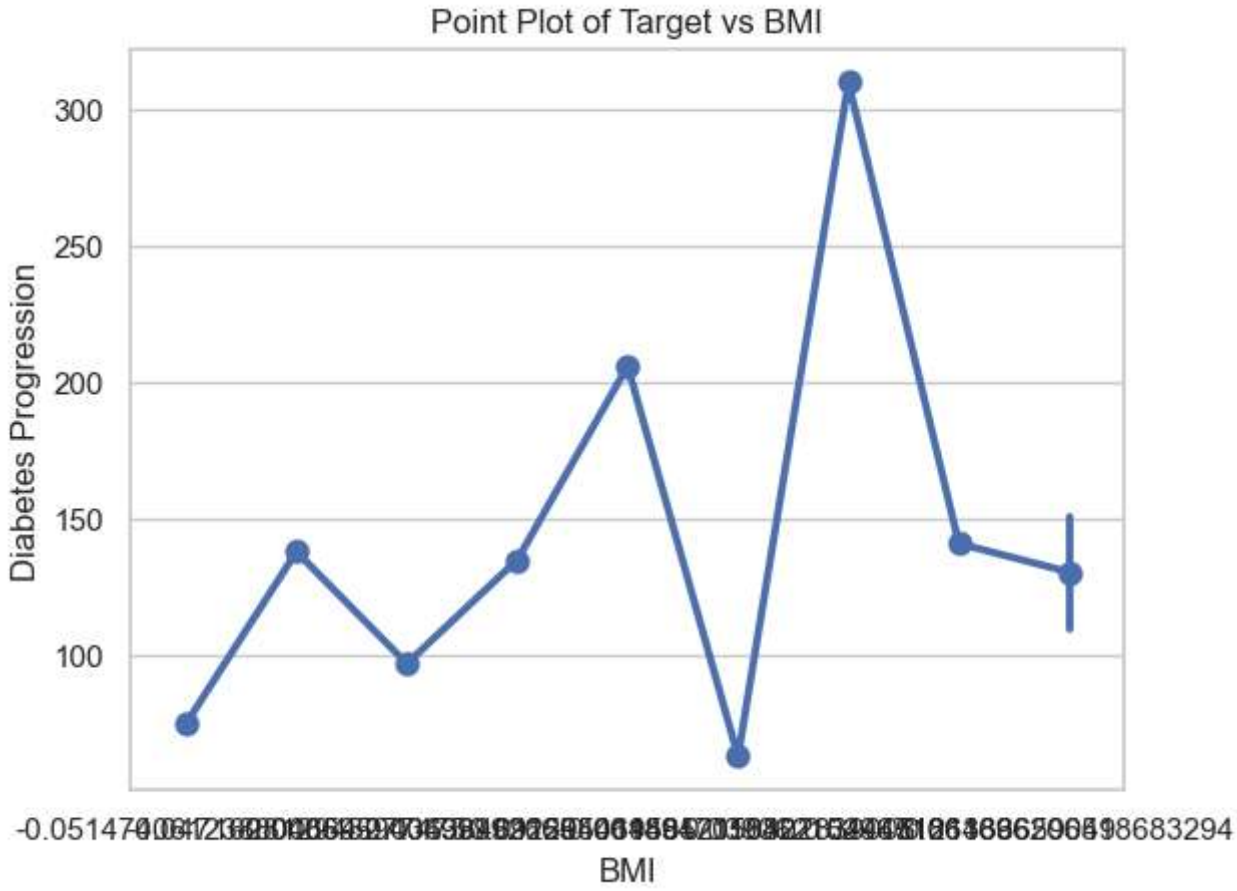
```
sns.pairplot(df, diag_kind='kde', markers='o', hue='target', palette='viridis')
# Show the pairplot
plt.show()
# Visualize the distribution of the target variable
sns.histplot(df['target'], kde=True, bins=30)
# Show the distribution plotplt.title("Distribution of Target Variable")
plt.title("Distribution of Target Variable")
plt.xlabel("Diabetes Progression")
plt.ylabel("Frequency")
plt.show()
```


age
sex
bmi
bp
s1
s2
s3
s4
s5
target
60
120
180
240
300





In [20]:
```
sns.pointplot(data=df.head(10), x='bmi', y='target')
# Show the point plot
plt.title("Point Plot of Target vs BMI")
plt.xlabel("BMI")
plt.ylabel("Diabetes Progression")
plt.show()
```



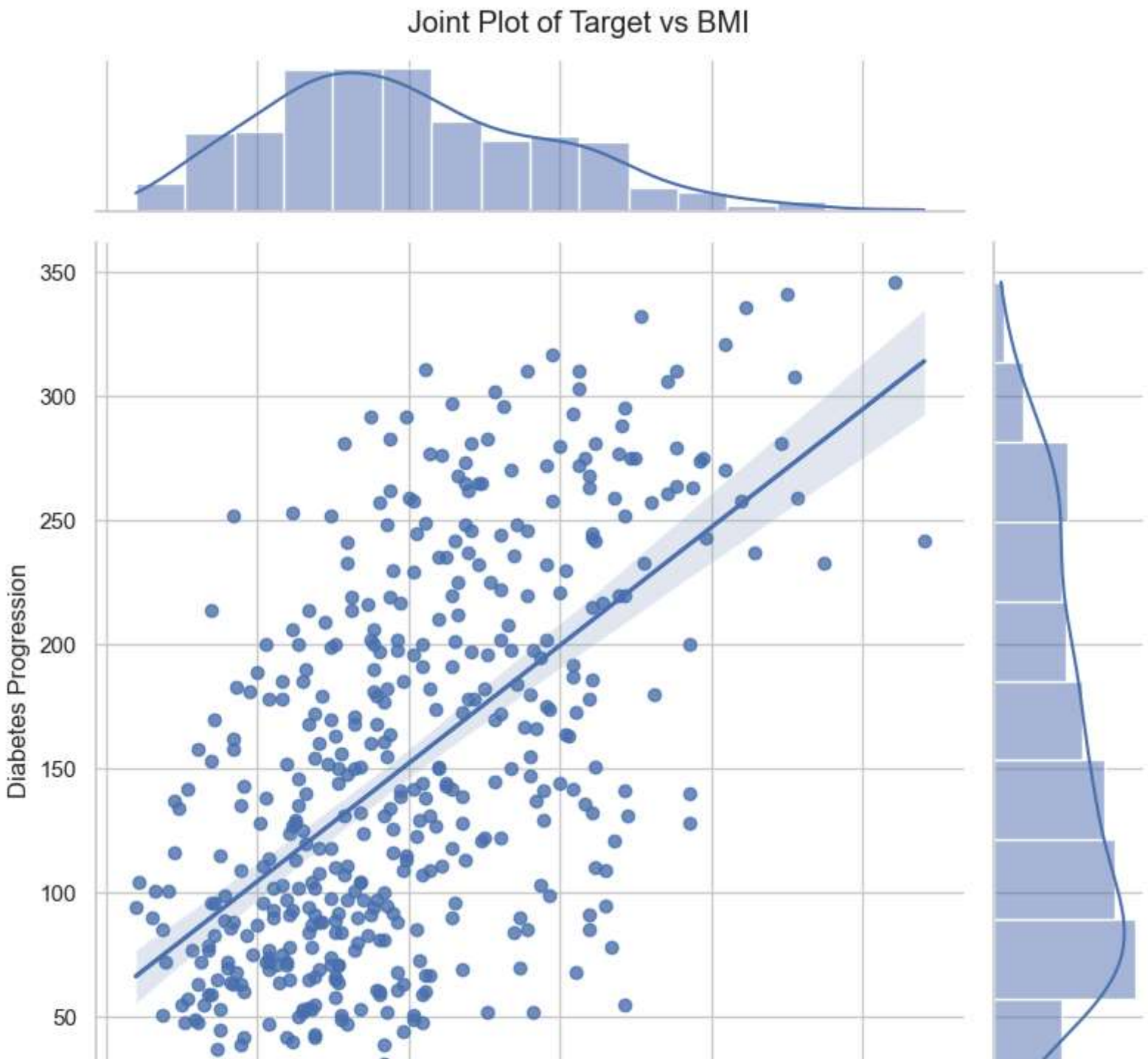
In [21]:
```
df["target"].describe()
```

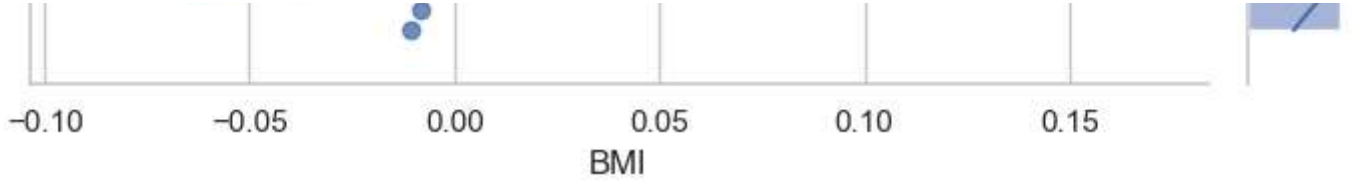
Out[21]:
```
count    442.000000
mean     152.133484
std       77.093005
min       25.000000
25%       87.000000
50%      140.500000
75%      211.500000
max      346.000000
Name: target, dtype: float64
```

In [22]:
```
df.skew()
```

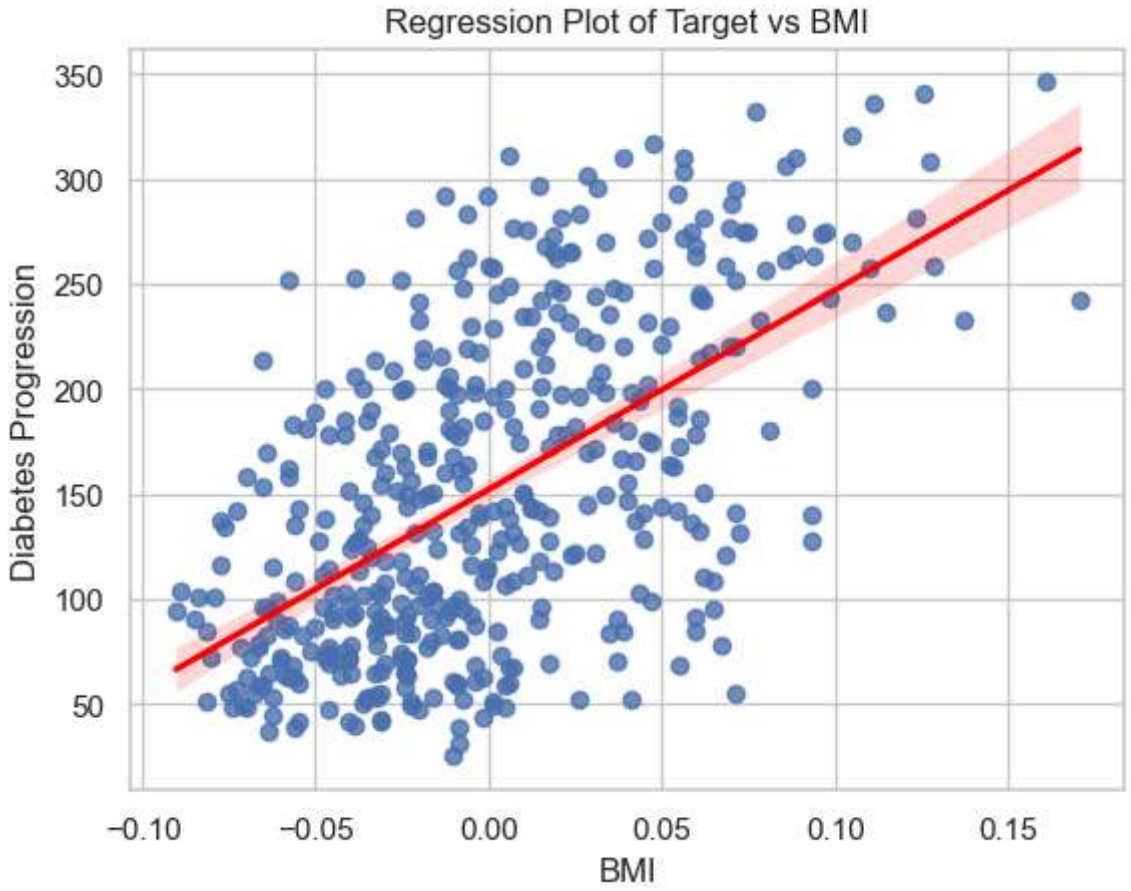
Out[22]:
```
age      -0.231382
sex       0.127385
bmi       0.598148
bp        0.290658
s1        0.378108
s2        0.436592
s3        0.799255
s4        0.735374
s5        0.291754
s6        0.207917
target    0.440563
dtype: float64
```

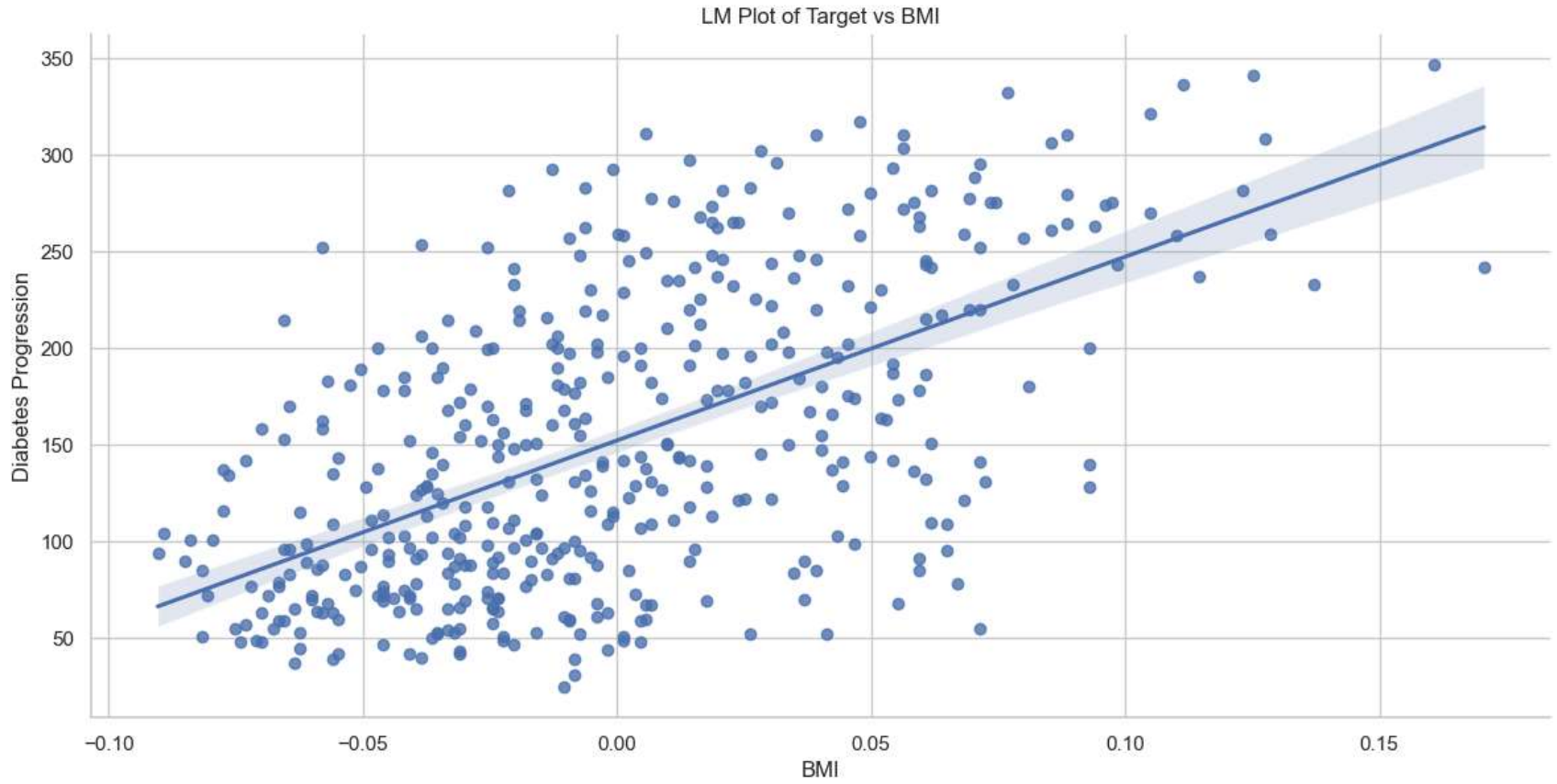
In [23]:
```
sns.jointplot(data=df, x='bmi', y='target', kind='reg', height=8)
plt.suptitle("Joint Plot of Target vs BMI", y=1.02)
plt.xlabel("BMI")
plt.ylabel("Diabetes Progression")
plt.show()
```


Joint Plot of Target vs BMI
Diabetes Progression
350
300
250
200
150
100
50



In [24]:
```
sns.regplot(data=df, x='bmi', y='target', line_kws={'color': 'red'})
plt.title("Regression Plot of Target vs BMI")
plt.xlabel("BMI")
plt.ylabel("Diabetes Progression")
plt.show()
sns.lmplot(data=df, x='bmi', y='target', aspect=2, height=6)
plt.title("LM Plot of Target vs BMI")
plt.xlabel("BMI")
plt.ylabel("Diabetes Progression")
plt.show()
```


Regression Plot of Target vs BMI
Diabetes Progression
350
300
250
200
150
100
50
−0.10
−0.05
0.00
0.05
0.10
0.15
BMI

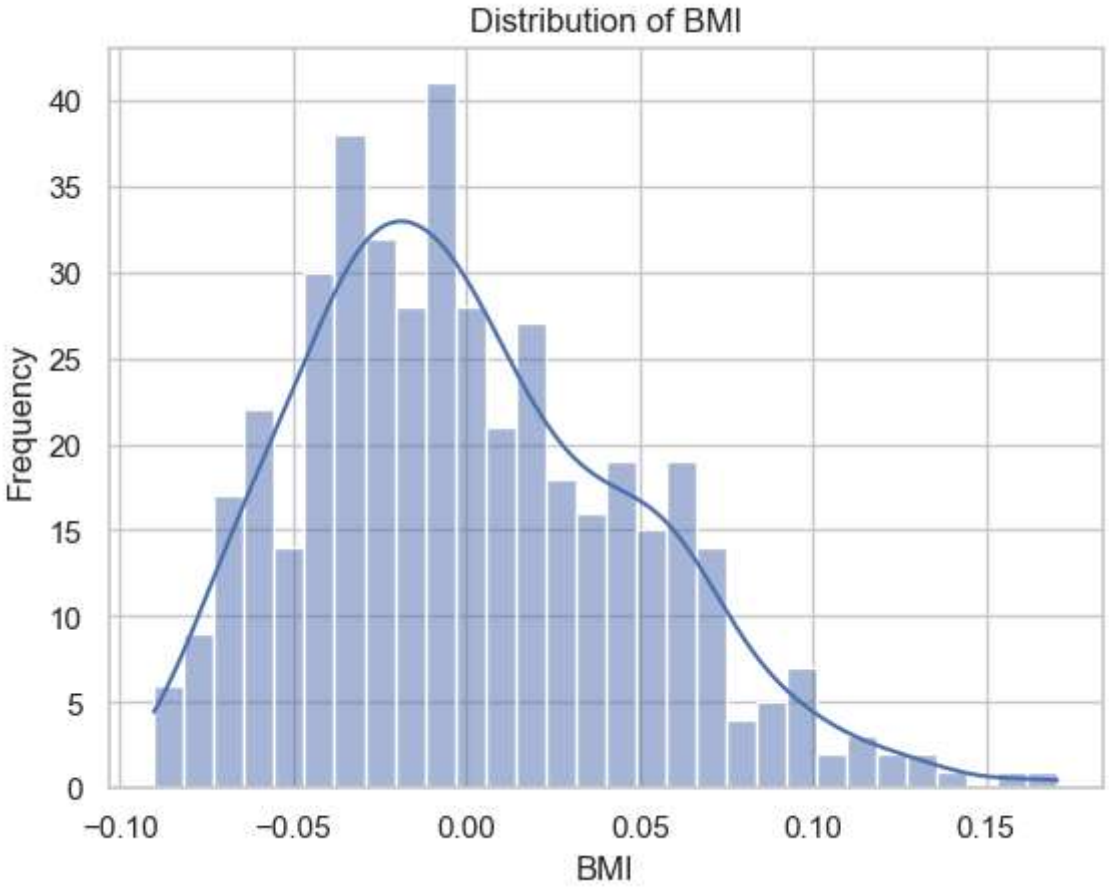


In [ ]:

```
from scipy import stats
# Calculate the Pearson correlation coefficient
pearson_corr, p_value = stats.pearsonr(df['bmi'], df['target'])
print(f"Pearson correlation coefficient: {pearson_corr:.2f}, p-value: {p_value:.2e}")
# Calculate the Spearman correlation coefficient
spearman_corr, p_value_spearman = stats.spearmanr(df['bmi'], df['target'])
print(f"Spearman correlation coefficient: {spearman_corr:.2f}, p-value: {p_value_spearman:.2e}")
# Calculate the Kendall correlation coefficient
kendall_corr, p_value_kendall = stats.kendalltau(df['bmi'], df['target'])
print(f"Kendall correlation coefficient: {kendall_corr:.2f}, p-value: {p_value_kendall:.2e}")
# Visualize the distribution of BMI
sns.histplot(df['bmi'], kde=True, bins=30)
plt.title("Distribution of BMI")
plt.xlabel("BMI")
```

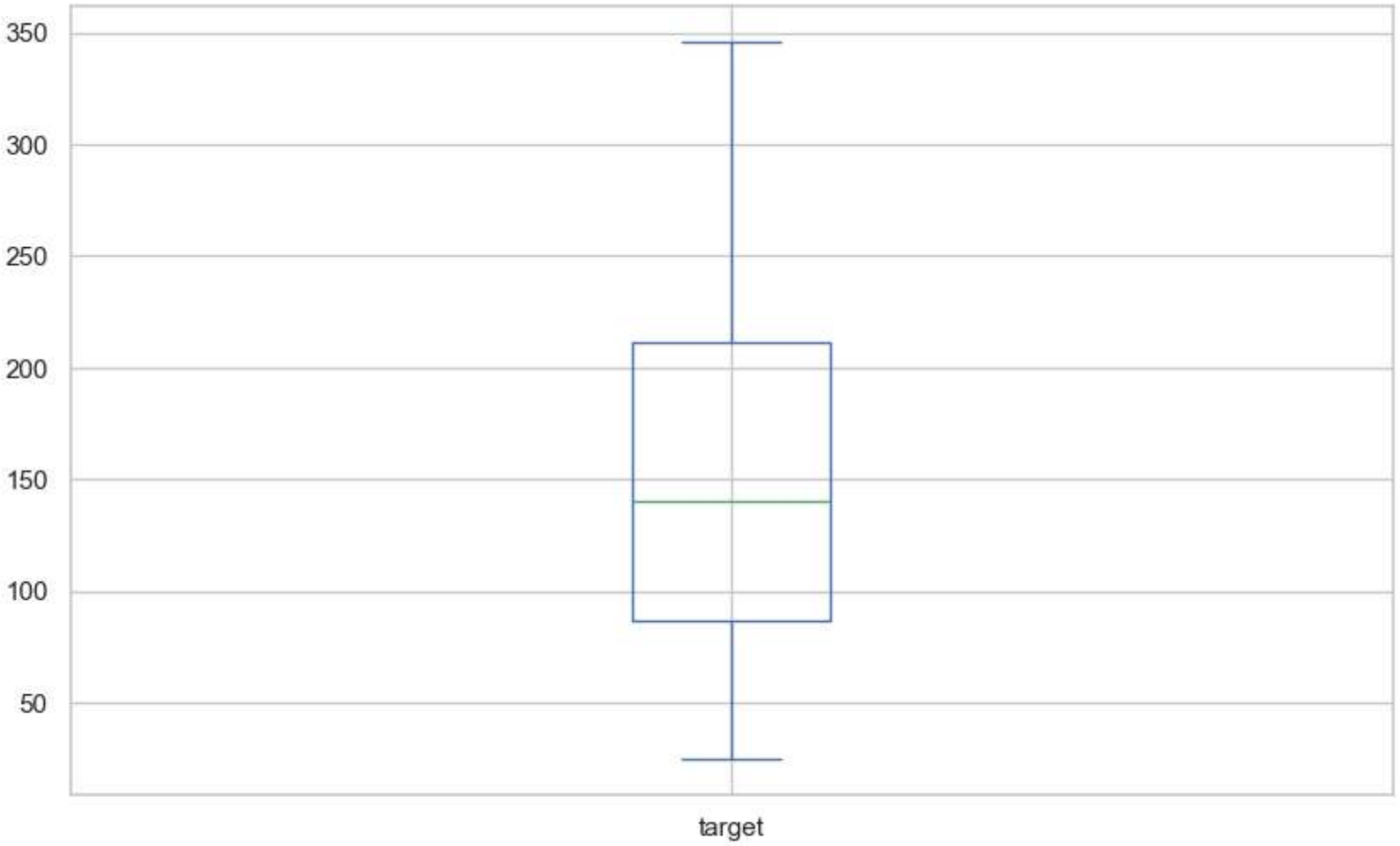
```
plt.ylabel("Frequency")
plt.show()
```

```
Pearson correlation coefficient: 0.59, p-value: 3.47e-42
Spearman correlation coefficient: 0.56, p-value: 4.57e-38
Kendall correlation coefficient: 0.39, p-value: 2.24e-34
```



In [26]:
```
df['target'].plot(kind='box', figsize=(10, 6))
```

Out[26]: <Axes: >



In [27]:
```
y_pred_lr
```

Out[27]:
```
array([139.5475584 , 179.51720835, 134.03875572, 291.41702925,
       123.78965872,  92.1723465 , 258.23238899, 181.33732057,
        90.22411311, 108.63375858,  94.13865744, 168.43486358,
        53.5047888 , 206.63081659, 100.12925869, 130.66657085,
       219.53071499, 250.7803234 , 196.3688346 , 218.57511815,
       207.35050182,  88.48340941,  70.43285917, 188.95914235,
       154.8868162 , 159.36170122, 188.31263363, 180.39094033,
        47.99046561, 108.97453871, 174.77897633,  86.36406656,
       132.95761215, 184.53819483, 173.83220911, 190.35858492,
       124.4156176 , 119.65110656, 147.95168682,  59.05405241,
        71.62331856, 107.68284704, 165.45365458, 155.00975931,
       171.04799096,  61.45761356,  71.66672581, 114.96732206,
        51.57975523, 167.57599528, 152.52291955,  62.95568515,
       103.49741722, 109.20751489, 175.64118426, 154.60296242,
        94.41704366, 210.74209145, 120.2566205 ,  77.61585399,
       187.93203995, 206.49337474, 140.63167076, 105.59678023,
       130.70432536, 202.18534537, 171.13039501, 164.91423047,
       124.72472569, 144.81030894, 181.99635452, 199.41369642,
       234.21436188, 145.95665512,  79.86703276, 157.36941275,
       192.74412541, 208.89814032, 158.58722555, 206.02195855,
       107.47971675, 140.93598906,  54.82129332,  55.92573195,
       115.01180018,  78.95584188,  81.56087285,  54.37997256,
       166.2543518 ])
```

In [28]:
```
y_pred_lr.std()
```

Out[28]:
```
np.float64(54.040341022653124)
```