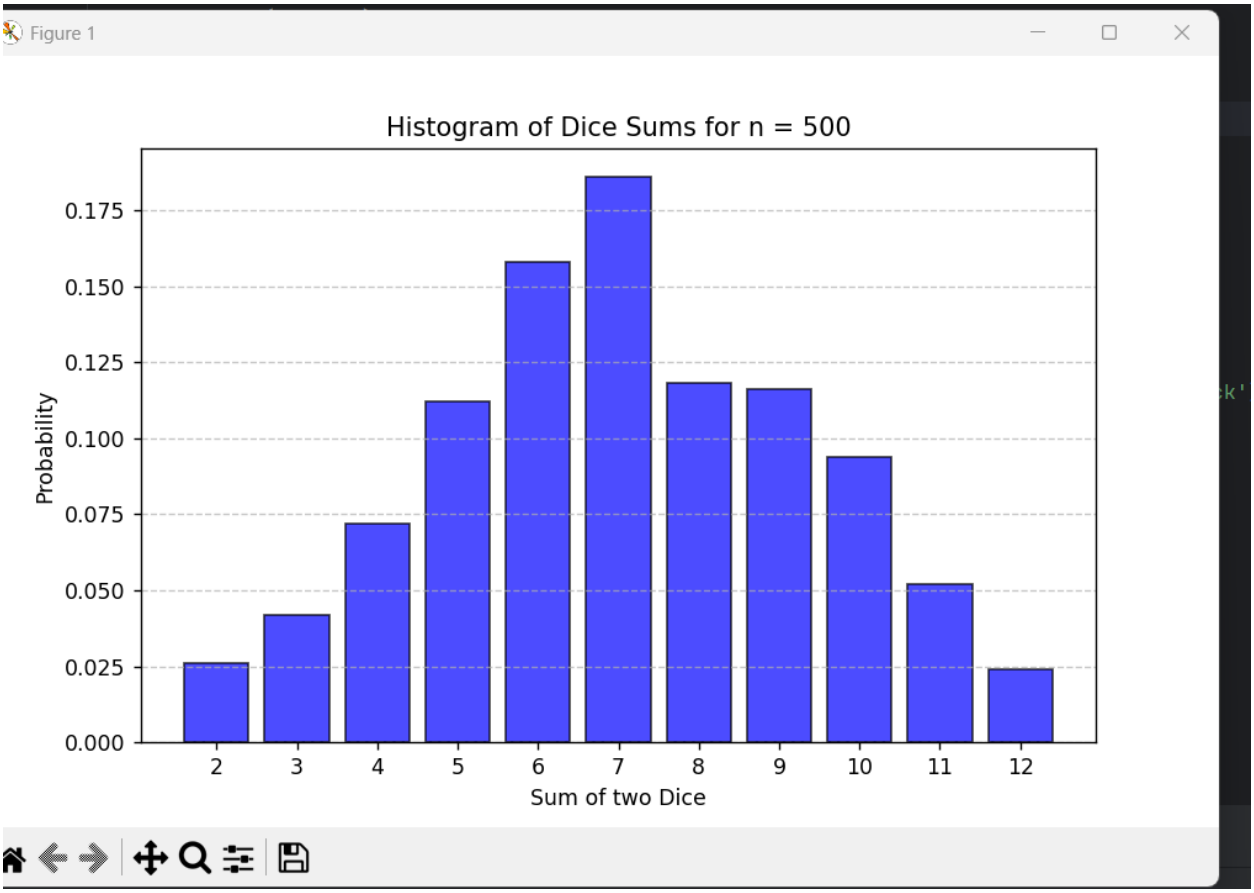
Assignment 4

Ex.1

Solution,

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('TkAgg')
sample_sizes = [500, 1000, 2000, 5000, 10000, 15000, 20000, 50000,
100000]
for n in sample_sizes:
    dice1 = np.random.randint(1, 7, size=n)
    dice2 = np.random.randint(1, 7, size=n)
    sums = dice1 + dice2
    h, h_edges = np.histogram(sums, bins=range(2, 14))

    probabilities = h / n

    plt.figure(figsize=(8, 5))
    plt.bar(range(2, 13), probabilities, width=0.8, color='blue', alpha=0.7,
edgecolor='black')
    plt.title(f"Histogram of Dice Sums for n = {n}")
    plt.xlabel("Sum of two Dice")
    plt.ylabel("Probability")
    plt.xticks(range(2, 13))
    plt.grid(axis='y', linestyle='--', alpha=0.7)
    plt.show()
```

Output:



4.

I observed that this histogram is bellshaped and centred at 7, the most common sum. For small values of n there is more dispersion, and the graph is not quite as smooth; As n gets bigger the graph smooths out and approximates the theoretical probabilities more and more closely. Larger n is a good example of the Law of Large Numbers.

5.

The frequencies of the sums observed converge towards the theoretical probabilities as n increases, with 7 as the most probable, which demonstrates regression to the mean: extreme variations in small samples are averaged out, and results more closely approximate expected averages for larger samples.

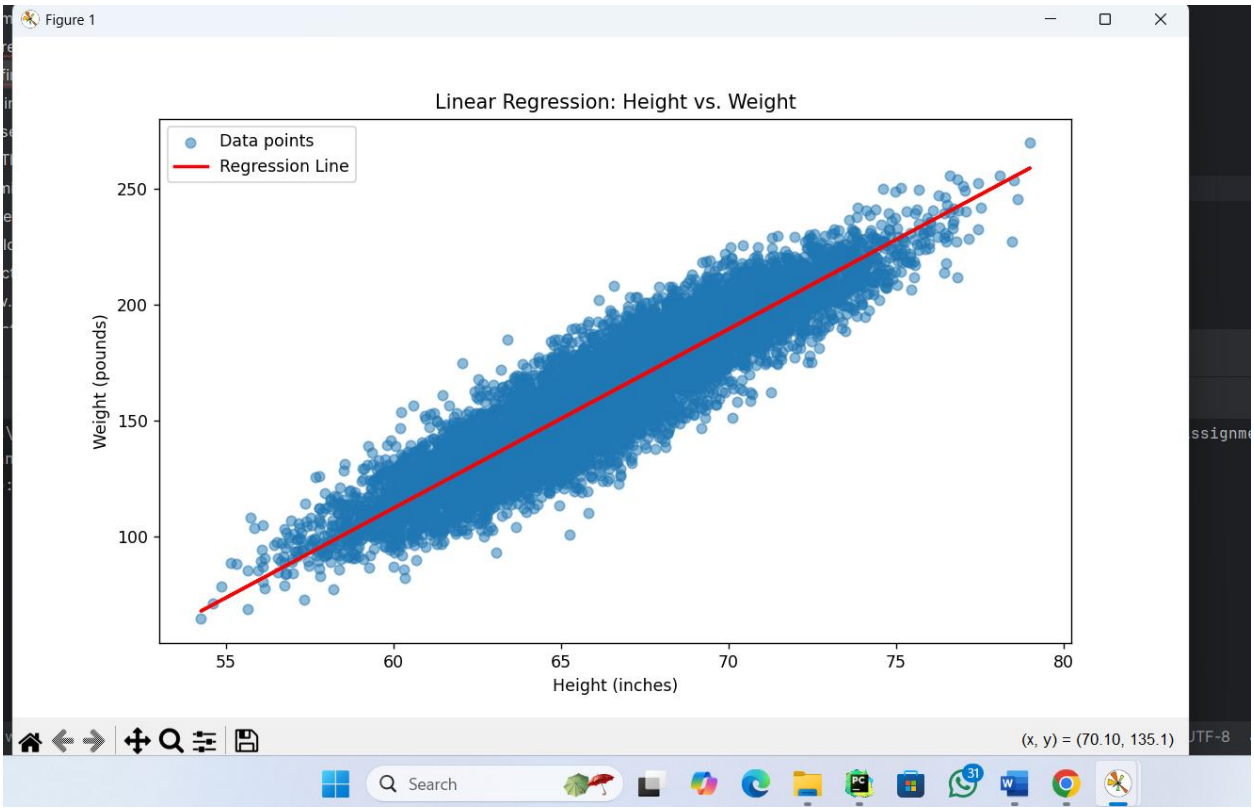
Ex.2

Solution,

```
import pandas as pd
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
matplotlib.use('TkAgg')
from sklearn import linear_model
from sklearn import metrics

data = pd.read_csv('weight-height.csv')

X = data["Height"].values.reshape(-1, 1)
y = data["Weight"].values

model = linear_model.LinearRegression()

model.fit(X, y)

y_pred = model.predict(X)

rmse = np.sqrt(metrics.mean_squared_error(y, y_pred))
r2 = metrics.r2_score(y, y_pred)  # R² score

print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R² Value: {r2:.2f}")

plt.figure(figsize=(10, 6))
plt.scatter(X, y, alpha=0.5, label="Data points")
plt.plot(X, y_pred, color="red", linewidth=2, label="Regression Line")
```

```
plt.title("Linear Regression: Height vs. Weight")
plt.xlabel("Height (inches)")
plt.ylabel("Weight (pounds)")
plt.legend()
plt.show()
```

Output:



6.

If the scatter plot is such that data points closely lie on the regression line, the fit is good. The wider the spread of the points, the poorer the fit is. RMSE will tell you how far away your predictions are from the actual values. The lower the value of RMSE, the better is the accuracy.

The value of R^2 ranges from 1 to 0; the nearer R^2 is to 1, the better the fit; the nearer it is to 0, the poorer the fit. A model fits well if it has a low RMSE with a high R^2.