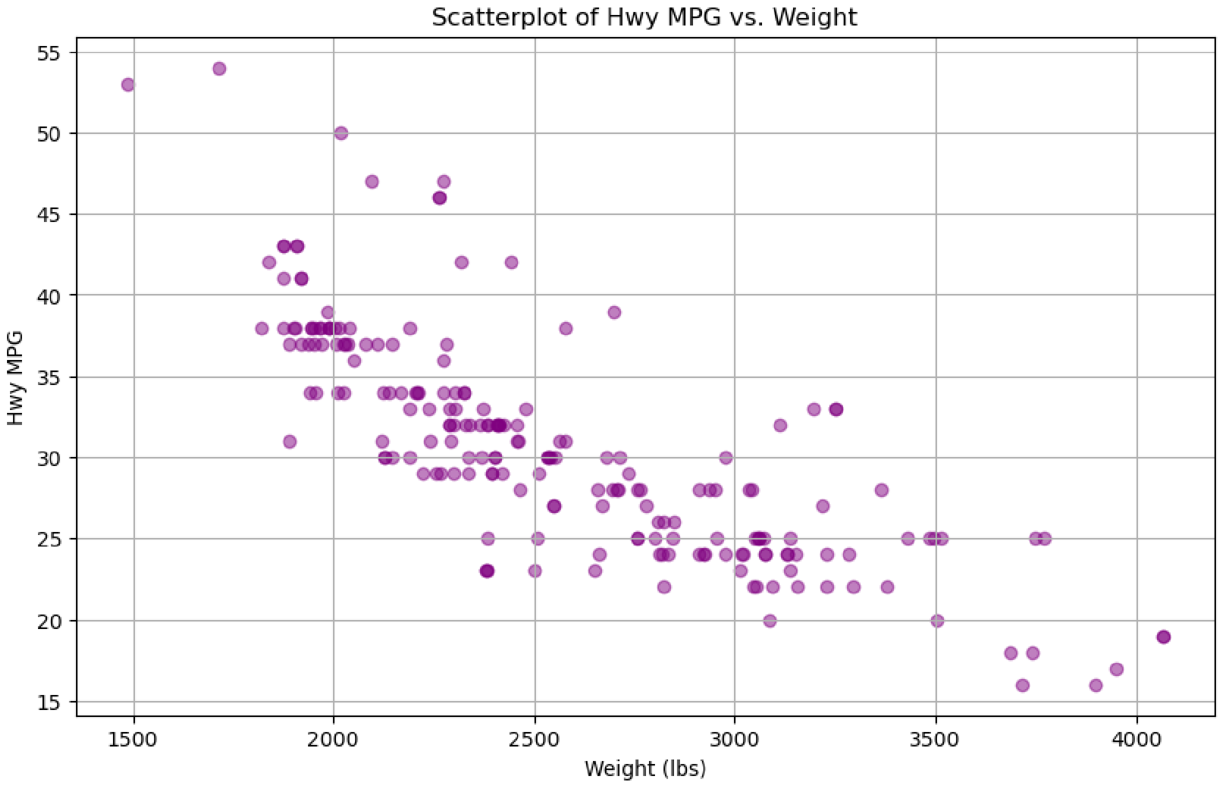
In [10]:

```
#make a scatterplot of the highway miles per gallon (y-axis) versus the weight (x-axis

#import pd not np
import pandas as pd
import matplotlib.pyplot as plt

#csv data excel transfer
#car data csv.csv
data = pd.read_csv("car data csv.csv")

#csv colums weight nd hwy_mpg
weight = data["weight"]
hwy_mpg = data["hwy_mpg"]

#chart details
plt.figure(figsize=(10, 6))
plt.scatter(weight, hwy_mpg, alpha=0.5, color="purple")
plt.title("Scatterplot of Hwy MPG vs. Weight")
plt.xlabel("Weight (lbs)")
plt.ylabel("Hwy MPG")
plt.grid(True)
plt.show()
```



Based on your plot, what is the general trend of how highway miles per gallon varies with the weight?

- The heavier the vehicle, the lower mpg they get.

If you were to build a linear model using this data to predict highway miles per gallon from weight,

would you expect the slope to be positive or negative? Explain

-If building a linear model, I would expect the slope to be negative. Given that there is already a well established notion in the automotive physics sector about the inverse relationship between vehicle weight and fuel efficiency. The heavier a vehicle is, the bigger engine it will have, which in turn will consume more fuel to propel it forward.

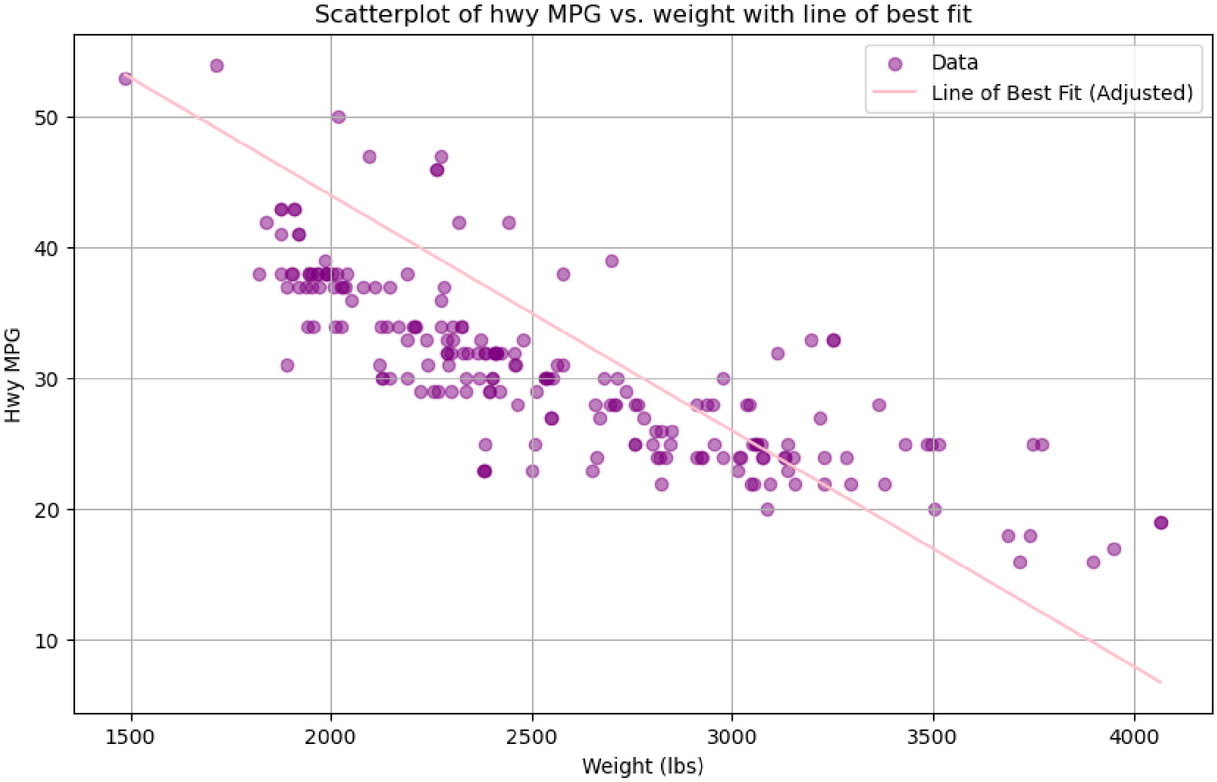
if the slope of a linear model predicting highway miles per gallon from the weight, interpret the meaning

of the slope being −0.05.

-the meaning of the slope being -0.5 is that for every unit increase in weight , we should be able to expect a coorelated decrease of 0.05 units in highway MPG.

In [24]:

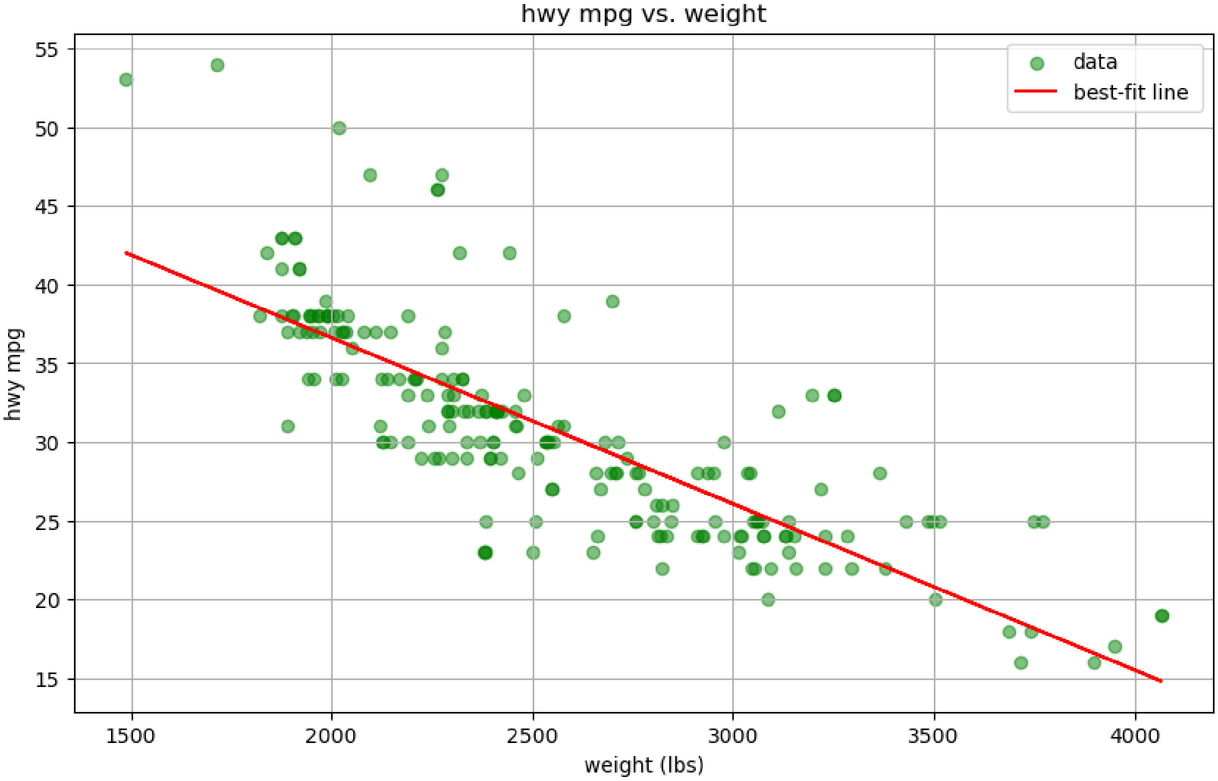
```
#write code to add a line to the graph you made in problem (1). Adjust the slope and y

#pandas,matplot, and numpy
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np


#csv data excel transfer
#car data csv.csv
data = pd.read_csv("car data csv.csv")
#columns in csv are weight and hwy_mpg
weight = data["weight"]
hwy_mpg = data["hwy_mpg"]

#plot data
plt.figure(figsize=(10, 6))
plt.scatter(weight, highway_mpg, alpha=0.5, color="purple", label="Data")
#slope adjust?
slope = -0.018
#y intercept adjust?
intercept = 80
x_range = np.linspace(weight.min(), weight.max(), 100)
y_fit = slope * x_range + intercept


plt.plot(x_range, y_fit, color="pink", label="Line of Best Fit (Adjusted)")

plt.title("Scatterplot of hwy MPG vs. weight with line of best fit")
plt.xlabel("Weight (lbs)")
plt.ylabel("Hwy MPG")
plt.grid(True)
plt.legend()
plt.show()
```



Final values slope: -0.018 y-intercept: 80

In [28]:

```
#use Python to find the best-fit line. The Scikit-Learn package is a good choice to us

import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

#csv data excel transfer
#car data csv.csv
data = pd.read_csv("car data csv.csv")
#csv weight and hwy_mpg
X = data[["weight"]]
y = data["hwy_mpg"]  #

#linear regression
model = LinearRegression()
model.fit(X, y)
#slope and y intercept
slope = model.coef_[0]
intercept = model.intercept_

#scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(X, y, alpha=0.5, color="green", label="data")
y_fit = slope * X + intercept
plt.plot(X, y_fit, color="red", label=f"best-fit line ")
plt.title("hwy mpg vs. weight")
plt.xlabel("weight (lbs)")
plt.ylabel("hwy mpg")
plt.grid(True)
plt.legend()
plt.show()

#slope and y intercept
print(f"slope: {slope:.4f}")
print(f"intercept: {intercept:.4f}")
```



```
slope: -0.0105
intercept: 57.7052
```

In [45]:

```
#use the best-fit line in problem (5) to predict the highway mpg of a car that weighs

#my slope or python slope?
#my predicted slope and y intercept
slope_best_fit = -0.018
intercept_best_fit = 80
weight = 3200
predicted_mpg = slope_best_fit * weight + intercept_best_fit

print(f"highway mpg of a car that weighs 3200 pounds: {predicted_mpg:.2f}")
```

```
highway mpg of a car that weighs 3200 pounds: 22.40
```