```
================================================

Title: 2.2 Exercises

Author: Chad Wood

Date: 8 Dec 2021

Modified By: Chad Wood

Description: This program demonstrates the use of scatterplots and linear regression lines

================================================
```

Scatterplots and Linear Regression

For this assignment, you will be using the data set car data.csv. This data set is a modified version of the data set imports 85.data from https://archive.ics.uci.edu/ml/datasets/automobile. There are two columns in this data set. One is the weight of the vehicle in pounds, and the other is the highway miles per gallon.

Setup

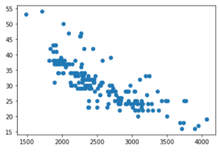
In [12]:
```
import pandas as pd
import matplotlib.pyplot as plt

cars = pd.read_csv('week2data\car_data.csv')
houses = pd.read_csv('week2data\housing_data.csv')
```

1. Make a scatterplot of the highway miles per gallon (y-axis) versus the weight (x-axis)

In [14]:
```
# plt.scatter(x, y)
plt.scatter(cars.weight, cars.hwy_mpg)
```

Out[14]:
```
<matplotlib.collections.PathCollection at 0x1a43f0d8a00>
```



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

The plot shows a linear negative slope indicating that as weight goes up, mpg goes down.

3. 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.

I would expect the slope to be negative because it descends as the explanatory variable decreases.

4. If the slope of a linear model predicting highway miles per gallon from the weight, interpret the meaning of the slope being -0.05.

Slope is the gradient of a regression line, and a slope of -0.05 would equate to a change in y of -0.05 for every change in x of 1

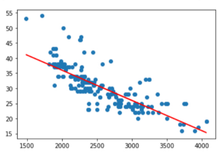
5. Write code to add a line to the graph you made in problem (1). Adjust the slope and y-intercept of this line until you think you have found the line that best fits the data. Record the slope and y-intercept.

In [31]:
```
x, y = cars.weight, cars.hwy_mpg

plt.scatter(x,y)

# Adds regression line, m: -0.01; b: 56
plt.plot(x, -0.01*x + 56, color='red')
```

Out[31]:
```
[<matplotlib.lines.Line2D at 0x1a4403a0df0>]
```



6. Use Python to find the best-fit line. The Scikit-learn package is a good choice to use for this.

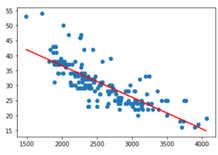
In [33]:
```
import numpy as np

x, y = cars.weight, cars.hwy_mpg
# Determines least squares polynomial fit for slope and y-intercept
m, b = np.polyfit(x, y, 1)

plt.scatter(x,y)

# Adds regression line, m: -0.010547157168719583; b: 57.705174225744706
plt.plot(x, m*x + b, color='red')
```

Out[33]:
```
[<matplotlib.lines.Line2D at 0x1a440429af0>]
```



7. Find the root mean squared error (RMSE) of the prediction line you found in problem (4) and the actual best-fit line found in problem (5). How do these compare?

In [37]:
```
def rmse(actual, predicted):

    # Squares each element in list
    def square(list):
        return [i ** 2 for i in list]

    # Equivilant to RMSE formula
    sqrdDiffs = square(predicted-actual)
    return (1/len(actual) * sum(sqrdDiffs)) ** 0.5
```

In [38]:
```
# From problem 5:
yhat = -0.01*x + 56

rmse(y, yhat)
```

Out[38]:
```
4.165945387977351
```

In [39]:
```
# From problem 6:
m, b = np.polyfit(x, y, 1)
yhat = m*x + b

rmse(y, yhat)
```

Out[39]:
```
4.144895442072009
```

8. Use the best-fit line in problem (5) to predict the highway mpg of a car that weighs 3200 pounds.

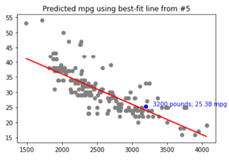
In [63]:
```
# From problem 5: Gets yhat values and predicts y
yhat = -0.01*x + 56
predicY = np.interp(3200, x, yhat)

# From problem 5: Generates plot
plt.scatter(x, y, color='grey')
plt.plot(x, yhat, color='red')

# Adds label text and marker
stagger=100
plt.text(3200+stagger, predicY, f'3200 pounds; {predicY} mpg', color='blue')
plt.plot(3200, predicY, 'bo')

plt.title('Predicted mpg using best-fit line from #5')
```

Out[63]:
```
Text(0.5, 1.0, 'Predicted mpg using best-fit line from #5')
```



In [65]:
```
# From problem 6: Gets yhat values and predicts y
m, b = np.polyfit(x, y, 1)
yhat = m*x + b
predicY = np.interp(3200, x, yhat)

# From problem 6: Generates plot
plt.scatter(x,y, color='grey')
plt.plot(x, yhat, color='red')

# Adds label text and marker
stagger=100
plt.text(3200+stagger, predicY, f'3200 pounds; {predicY} mpg', color='blue')
plt.plot(3200, predicY, 'bo')

plt.title('Predicted mpg using best-fit line from #6')
```

Out[65]:
```
Text(0.5, 1.0, 'Predicted mpg using best-fit line from #6')
```