MODULE 3

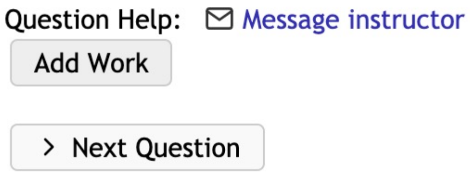
QUIZ REVIEW

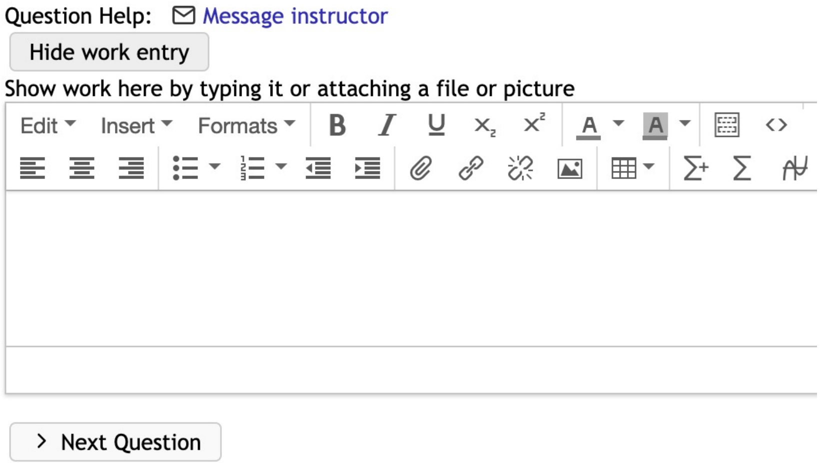
MODULE 3 QUIZ

- One attempt per question – no time limit
- Can leave and return throughout the week

Caution: A new version of unanswered problems may be generated

- Add work option available





- Correct answers and scores show immediately
- Based on Modules 1 & 2

MODULE 1 – LINEAR FUNCTIONS

- Linear equations
  - Slope-intercept form: $y = mx + b$
  - Standard form: $Ax + By = C$, where A and B are not both zero
- Linear functions
  - $f(x) = mx + b$
  - Graphs are non-vertical lines
- Least squares calculations
  - Uses method of least squares to determine least squares line
  - Technology is recommended for least squares line and correlation coefficient

MODULE 2 – LINEAR SYSTEMS

- Methods to solve
  - Graphing method
  - Substitution
  - Elimination
  - Inverse matrices
  - Cramer's Rule
- Matrices
  - Operations – addition, subtraction, multiplication
  - Inverses

LINEAR FUNCTIONS

U.S. imports from China have grown from about $364 billion in 2010 to $452 billion in 2019. This growth has been approximately linear.

(a) Determine a linear equation that approximates the growth in imports in terms of t, where t represents the number of years since 2000. Round to two decimal places as needed.

Use $y = mx + b$

Two things needed: a point on the line and the slope of the line

Use the points $(10,364)$ and $(19,452)$ to determine the slope

$$m = \frac{452 - 364}{19 - 10} \approx 9.78$$

Use either point to find b

$$364 = 9.78(10) + b$$
$$b \approx 266.2$$

We can also write this as a function of t:

$$f(t) = 9.78t + 266.2$$

(b) Interpret the slope and y-intercept

Based on the linear function, U.S. imports from China were about $\$266$ billion in 2000 and are increasing at a rate of about $\$9.78$ billion per year

(c) Use the equation in part (a) to predict the amount in imports from China in 2025.

Use $t = 25$:

$$f(25) = 9.78(25) + 266.2 = 510.7$$

It's predicted that U.S. imports from China will be about $\$511$ billion in the year 2025

(d) If this trend continues, in what year will imports from China be at least $\$600$ billion?

$$9.78t + 266.2 = 600$$
$$t \approx 34.13$$

Since $t = 0$ corresponds to 2000, U.S. imports from China will be at least $\$600$ billion in the year 2035

BREAK-EVEN ANALYSIS

The linear cost function of the form $C(x) = mx + b$ represents the cost to produce x units, where m is the marginal cost and b is the fixed cost.

Let p be the selling price per unit and let x be the number of units sold.

The revenue, $R(x)$, is the amount of money that a company receives for the sale of x units:

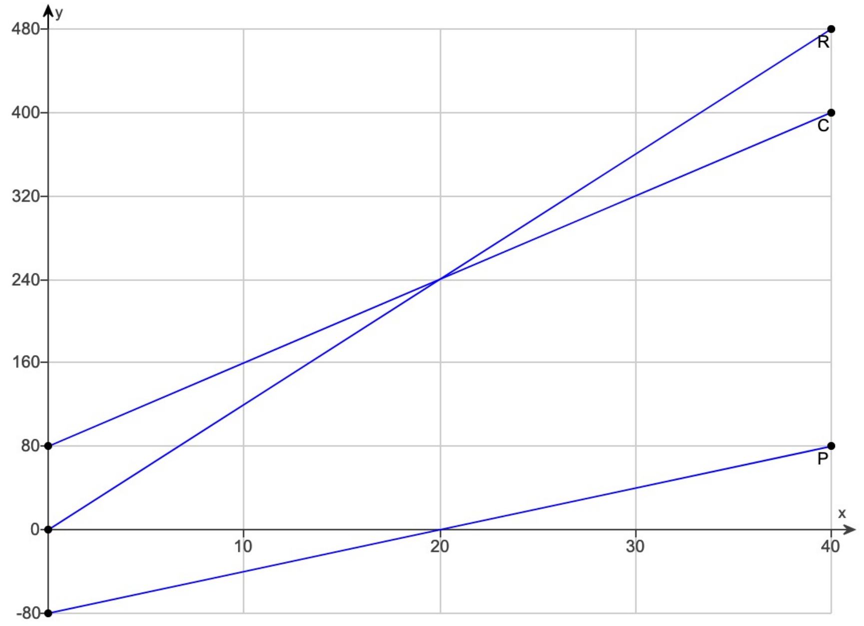
$$R(x) = px$$

The profit, $P(x)$, is the money a company makes from selling x units after paying its costs:

$$P(x) = R(x) - C(x)$$

The number of units, x, at which revenue equals cost is the break-even quantity.

Solve $R(x) = C(x)$ for x or $P(x) = 0$ for x to find the break-even quantity



If the product must be sold as a whole unit and the break-even quantity is not a whole number, always round up.

LEAST SQUARES CALCULATIONS

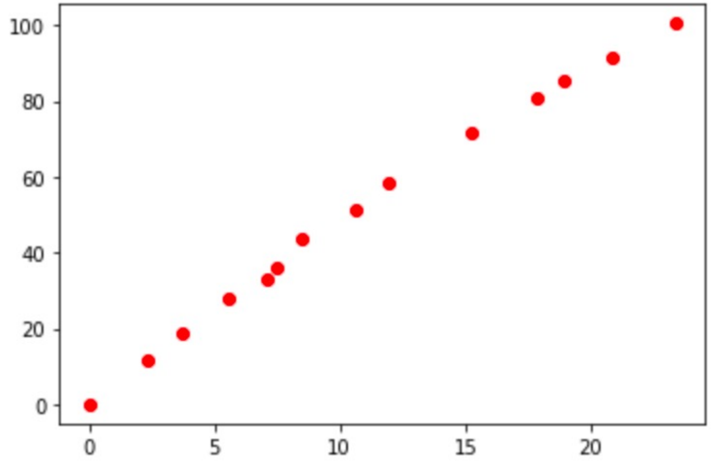
Consider the Hardrock 100, a 100.5 mile running race held in southwestern Colorado. In 2008, Kyle Skaggs was the winner of that race. The table below lists the times that he arrived at various mileage points along the way.

Time (hours)	Miles
0	0
2.317	11.5
3.72	18.9
5.6	27.8
7.08	32.8
7.5	36.0
8.5	43.9
10.6	51.5
11.93	58.4
15.23	71.8
17.82	80.9
18.97	85.2
20.83	91.3
23.38	100.5

(a) What was his average speed?

$\dfrac{100.5}{23.38} \approx 4.298$ mph

(b) Graph the data, plotting time on the x-axis and distance on the y-axis. Do the data appear to lie approximately on a straight line?



(c) Find the equation for the least squares line with distance as a function of time. Round to 3 decimal places as needed.

Use technology $y = 4.318x + 3.418$

(d) Calculate the correlation coefficient. Round to 4 decimal places as needed. Does it indicate a good fit of the least squares line to the data?

Use technology $r = 0.9971$

(e) Based on calculations so far, what is a good value for his average speed and why?

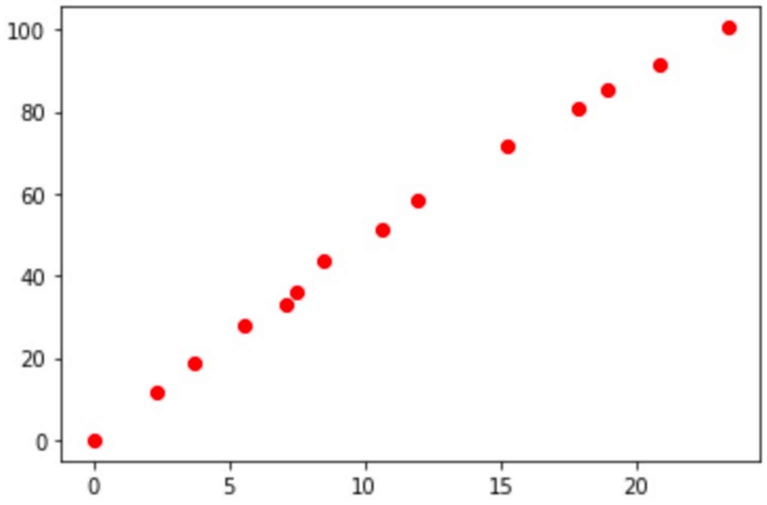
Due to a high correlation from part (d), the line in part (c) is a good fit to the data. So, 4.318 mph is a better value than part (a)

```python
import matplotlib.pyplot as plt
import numpy as np

x = np.array([0,2.317,3.72,5.6,7.08,7.5,8.5,10.6,11.93,15.23,17.82,18.97,20.83,23.38])
y = np.array([0,11.5,18.9,27.8,32.8,36.0,43.9,51.5,58.4,71.8,80.9,85.2,91.3,100.5])

plt.scatter(x, y, marker = 'o', color = 'r')
```

```
<matplotlib.collections.PathCollection at 0x7fa7a7c5f990>
```




```python
from scipy import stats

slope, intercept, r_value, p_value, std_err = stats.linregress(x, y)

print('The slope of the line is', round(slope,3))
print('The y-intercept is', round(intercept,3))
print('The correlation coefficient is', round(r_value,4))
```

```
The slope of the line is 4.318
The y-intercept is 3.418
The correlation coefficient is 0.9971
```

METHODS FOR SOLVING SYSTEMS OF LINEAR EQUATIONS

- Graphing
  - Only works with 2 variables
  - Best with integer solutions
  - Use intercepts to graph
  - Quick way to see if no solution or infinite solutions
- Substitution
  - Quick algebraic method to simplify system
  - Always recommended when possible
- Elimination
  - Eliminate fractions/decimals first
  - Eliminate common factors in any equation first
  - Works nicely if system is not too large
- Inverse coefficient matrix
  - Only works for square coefficient matrix
  - Eliminate fractions/decimals first
  - Eliminate common factors in any row first
  - Can be tedious
  - Consider checking with technology
- Cramer's Rule
  - Only works for square coefficient matrix
  - Can be tedious
  - Recommended in conjunction with technology
- Python
  - Very efficient with square coefficient matrix
  - Always recommended when possible

SYSTEMS OF LINEAR EQUATIONS

A lake is stocked each spring with three species of fish, A, B, and C. Three foods, I, II, and III, are available in the lake. Each fish of species A requires an average of 1.31 units of food I, 2.9 units of food II, and 1.75 units of food III each day. Species B fish each require 2.1 units of food I, 0.95 units of food II, and 0.6 units of food III daily. Species C fish require 0.86, 1.52, and 2.01 units of foods I, II, and III, respectively. If 490 units of food I, 897 units of food II, and 653 units of food III are available daily, how many of each species should be stocked?

Let A = the number of species A fish, B = the number of species B fish, and C = the number of species C fish to be stocked

Equation for units of food I: $1.31A + 2.1B + 0.86C = 490$

Equation for units of food II: $2.9A + 0.95B + 1.52C = 897$

Equation for units of food III: $1.75A + 0.6B + 2.01C = 653$

Graphing and substitution methods cannot be used

The elimination method is not recommended

Finding the inverse of the coefficient matrix by hand is not recommended

That leaves Cramer's Rule and technology or a combination of the two

CRAMER'S RULE

$$1.31A + 2.1B + 0.86C = 490$$
$$2.9A + 0.95B + 1.52C = 897$$
$$1.75A + 0.6B + 2.01C = 653$$

$$D = \begin{vmatrix} 1.31 & 2.1 & 0.86 \\ 2.9 & 0.95 & 1.52 \\ 1.75 & 0.6 & 2.01 \end{vmatrix} \approx -5.2815$$

$$D_x = \begin{vmatrix} 490 & 2.1 & 0.86 \\ 897 & 0.95 & 1.52 \\ 653 & 0.6 & 2.01 \end{vmatrix} \approx -1283.735$$

$$D_y = \begin{vmatrix} 1.31 & 490 & 0.86 \\ 2.9 & 897 & 1.52 \\ 1.75 & 653 & 2.01 \end{vmatrix} \approx -212.5759$$

$$D_z = \begin{vmatrix} 1.31 & 2.1 & 490 \\ 2.9 & 0.95 & 897 \\ 1.75 & 0.6 & 653 \end{vmatrix} \approx -534.7035$$

The lake should be stocked with 243 fish of species A, 40 fish of species B, and 101 fish of species C.

$$A = \frac{D_x}{D} \approx 243$$

$$B = \frac{D_y}{D} \approx 40$$

$$C = \frac{D_z}{D} \approx 101$$

Reminder: Python could be used for these calculations

```
import numpy as np

a = np.array([[1.31,2.1,0.86],[2.9,0.95,1.52],[1.75,0.6,2.01]])
a1 = np.array([[490,2.1,0.86],[897,0.95,1.52],[653,0.6,2.01]])
a2 = np.array([[1.31,490,0.86],[2.9,897,1.52],[1.75,653,2.01]])
a3 = np.array([[1.31,2.1,490],[2.9,0.95,897],[1.75,0.6,653]])

D = np.linalg.det(a)
Dx = np.linalg.det(a1)
Dy = np.linalg.det(a2)
Dz = np.linalg.det(a3)

A = Dx/D
B = Dy/D
C = Dz/D

print('A =', A)
print('B =', B)
print('C =', C)
```

```
A = 243.06142638726502
B = 40.24896218421756
C = 101.24036144863464
```

PYTHON

$$1.31A + 2.1B + 0.86C = 490$$
$$2.9A + 0.95B + 1.52C = 897$$
$$1.75A + 0.6B + 2.01C = 653$$

```
import numpy as np
from scipy import linalg

a = np.array([[1.31,2.1,0.86],[2.9,0.95,1.52],[1.75,0.6,2.01]])
b = np.array([490,897,653])

np.dot(linalg.inv(a),b)
```

```
array([243.06142639,  40.24896218, 101.24036145])
```

Note: Only works for square coefficient matrices

QUESTIONS?