Math for Data Science

Variables and Functions

Joanna Bieri DATA100

Today's Goals:

- Deeper Thinking: What is a function? What is a variable?
- Overview: Types of Functions **No BS Guide to Math and Physics** Functions Reference (p.63)
- Introduction to Empirical Modeling.
- Least Squares Regression

What is a function?

Seriously... can you describe this idea.

What is a function?

Give a specific example of a mathematical formula for a function.

What is a function?

Can you give a general definition for a function?

What is a function?

Definition: Function, Domain, and Range

Let A and B be two sets. A function f from A to B is a relation between A and B such that for each a in A there is one and only one associated b in B. The set A is called the domain of the function, B is called its range.

Often a function is denoted as $y = f(x)$ or simply $f(x)$.

Common Function Types:

LINEAR

- Line

$$y = f(x) = mx + b$$

Common Function Types:

NON-LINEAR

- Square
$$y = f(x) = ax^2 + b$$
- Square Root
$$y = f(x) = a\sqrt{x} + b$$
- Absolute Value
$$y = f(x) = a|x| + b$$
- Polynomials higher than degree 1

$$y = f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + ...$$

Common Function Types:

NON-LINEAR

- Sine
$$y = f(x) = a\sin(bx) + c$$
- Cosine
$$y = f(x) = a\cos(bx) + c$$
- Exponential
$$y = f(x) = ae^{bx} + c$$
- Natural Logarithm
$$y = f(x) = a\ln bx + c$$

Common Function Types:

Each of these basic types of functions allows for transformations. As we choose different coefficients a, b, c, we can change the location and some of the shape of the function. More complicated functions can be build out of these basic function times. For example:

- Rational Functions

$$y = f(x) = \frac{P(x)}{Q(x)}$$

  where $P(x)$ and $Q(x)$ are polynomials.

Functions can have more than one variable!

In the examples above we see that our *dependent* variable y is a function of only one independent variable x. Sometimes we want our functions to allow for more variables.

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + ...$$

In this example β_n are all coefficients and X_n are all variables. We can have as many variables as we want!

Why do we care?!?

Having a good instinct for functions is important when choosing models in data science. We will explore these functions in the context of *Empirical Modeling*. Here are just a few areas that rely on knowing your functions:

- Predictive Modeling
- Feature Engineering
- Machine Learning - Cost Functions
- Data Visualization

Empirical Modeling

Empirical Modeling is the are of establishing how one variable depends on another using data. Sometimes, the goal of this type of modeling is to fit a line or a curve though your data that will allow you to predict results for instances where you do not have data. Other times, the goal is just to establish that there is a dependence (correlation) between two variables.

TODAY How do we actually fit a line through our data?

Empirical Modeling

Last class we used the command

```
coefficients = np.polyfit(x, y, 1)
```

to get the coefficients of a degree one polynomial through our data (line $y = ax + b$). It is fine to take advantage of the power of Python, but even better is to truly understand what these commands are doing.

A discussion of Linear Dependence

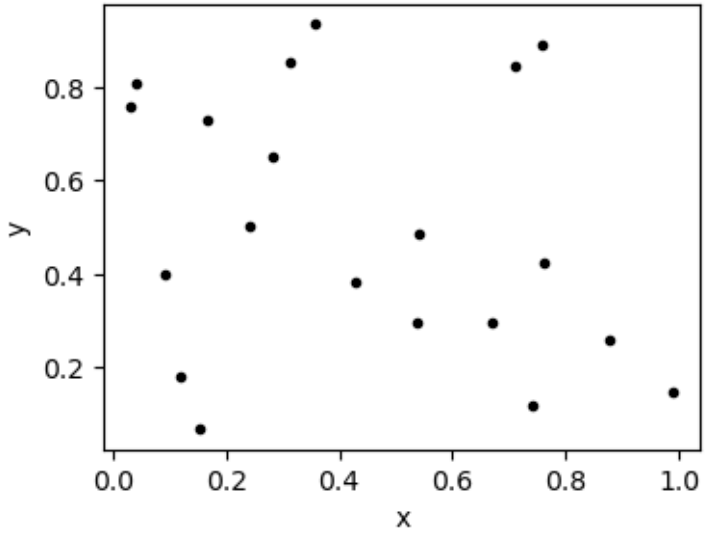
What does it mean if we say data contains a linear dependence?

Consider the following three graphs

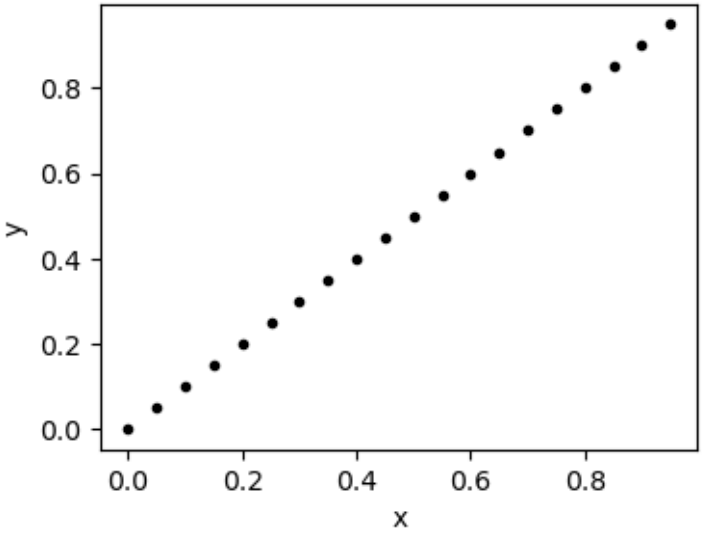
In which example would we say that y is highly dependent on x? Why?

In which example would we say that y does not seem to depend on x at all? Why?

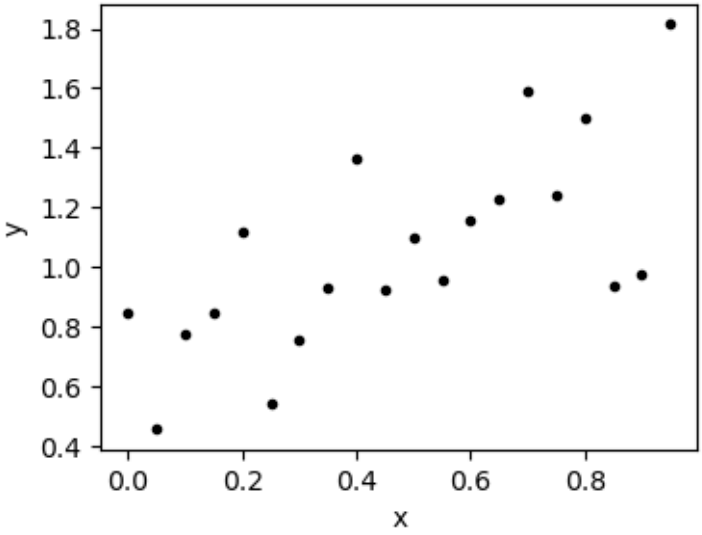
A discussion of Linear Dependence



A discussion of Linear Dependence



A discussion of Linear Dependence



How can we MEASURE the degree of linear dependence between two variables.

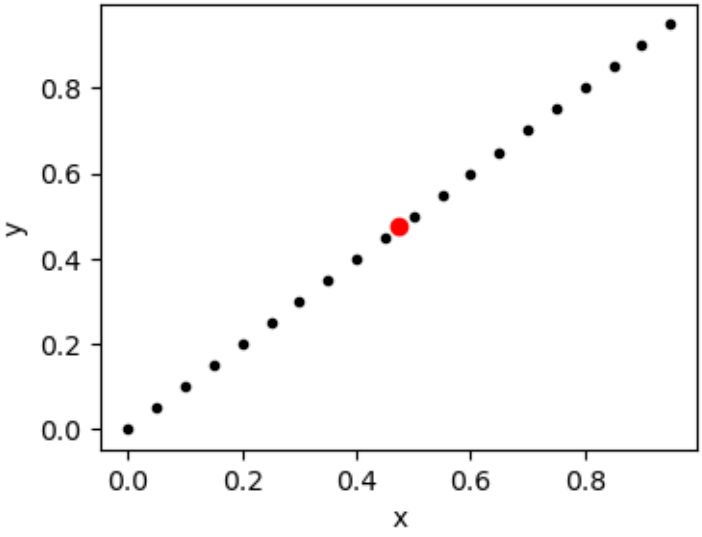
- It is fine to look at a graph and say - I SEE DEPENDENCE!
- it is a more convincing statement if you can measure that dependence.

Covariance

- Covariance measures how much two variables vary together.
- We will choose the average of the data $(\bar{x}, \bar{y})$ as our reference.
- Then for each point in the data we can ask, how far away are you in each direction from this average.

$$(x_i - \bar{x})$$
$$(y_i - \bar{y})$$

Covariance - Example 2 - Very Linear



Covariance - Example 2 - Very Linear

	x	y	x-bar(x)	y-bar(x)	mult
0	0.00	0.00	-0.475	-0.475	0.225625
1	0.05	0.05	-0.425	-0.425	0.180625
2	0.10	0.10	-0.375	-0.375	0.140625
3	0.15	0.15	-0.325	-0.325	0.105625
4	0.20	0.20	-0.275	-0.275	0.075625
5	0.25	0.25	-0.225	-0.225	0.050625
6	0.30	0.30	-0.175	-0.175	0.030625
7	0.35	0.35	-0.125	-0.125	0.015625
8	0.40	0.40	-0.075	-0.075	0.005625
9	0.45	0.45	-0.025	-0.025	0.000625
10	0.50	0.50	0.025	0.025	0.000625
11	0.55	0.55	0.075	0.075	0.005625
12	0.60	0.60	0.125	0.125	0.015625
13	0.65	0.65	0.175	0.175	0.030625
14	0.70	0.70	0.225	0.225	0.050625

Covariance - Example 2 - Very Linear

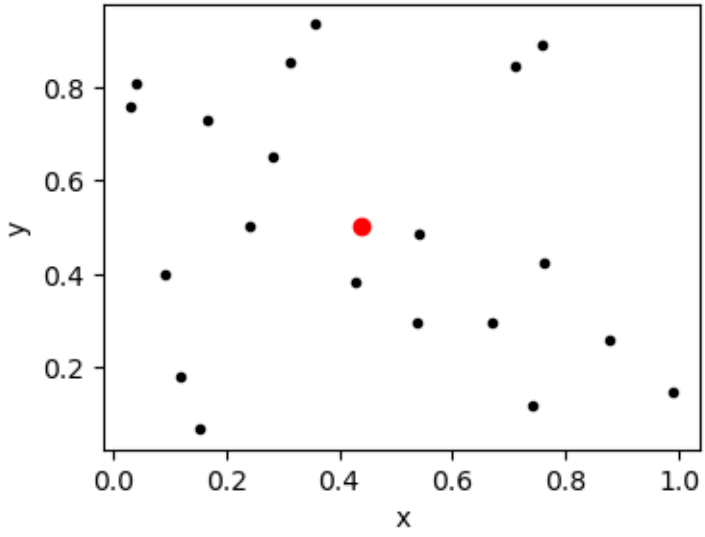
When we multiply together

$$(x_i - \bar{x})(y_i - \bar{y})$$

we see we get all positive values and if we added these up we would get an even more positive value!

```
1.6625
```

Covariance - Example 1 - Random Data



Covariance - Example 1 - Random Data

	x	y	x-bar(x)	y-bar(x)	mult
0	0.742329	0.117742	0.301962	-0.383808	-0.115895
1	0.426917	0.384317	-0.013450	-0.117233	0.001577
2	0.040778	0.809669	-0.399589	0.308119	-0.123121
3	0.670480	0.295140	0.230113	-0.206410	-0.047498
4	0.537662	0.296514	0.097294	-0.205036	-0.019949
5	0.165276	0.728022	-0.275091	0.226472	-0.062300
6	0.539733	0.485549	0.099366	-0.016001	-0.001590
7	0.312388	0.852854	-0.127979	0.351304	-0.044960
8	0.282735	0.649146	-0.157632	0.147595	-0.023266
9	0.708921	0.846366	0.268554	0.344815	0.092602
10	0.761814	0.425291	0.321447	-0.076259	-0.024513
11	0.090757	0.399341	-0.349610	-0.102209	0.035733
12	0.153400	0.068035	-0.286967	-0.433516	0.124405
13	0.877155	0.258577	0.436788	-0.242973	-0.106128
14	0.032674	0.757723	-0.407694	0.256173	-0.104440

Covariance - Example 1 - Random Data

Now when we multiply together

$$(x_i - \bar{x})(y_i - \bar{y})$$

we see we get some positive and some negative values. These will cancel each other out somewhat when we add them up.

```
-0.42350550332160514
```

YOU TRY

Redo this experiment but now with a FOURTH example $y = -x$ to see what happens when our straight line is decreasing slope.

Sample Covariance Equation

$$\hat{C}ov(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{(n-1)}$$

where n is the sample size. We divide by n so that we are not biased by the number of samples. (EG if we had a lot of positive samples this might add up to a really big number only because of the number of samples.)

Sample Covariance Equation

- $\hat{C}ov(x, y)$ is positive then as x increases in general y increases.
- $\hat{C}ov(x, y)$ is negative then as x increases in general y decreases.
- $\hat{C}ov(x, y)$ is zero then there is no indication of a linear dependence.

Sample Covariance Equation

Here are the results for our three examples:

Random

```
-0.022289763332716053
```

Linear

```
0.08750000000000001
```

Linear with some randomness

```
0.06882914052871991
```

YOU TRY

Calculate the sample covariance for your FOURTH example $y = -x$

Sample Correlation Coefficient r

In the examples here we see that all the numbers are similar because all of the data for the experiments was between zero and one. If we are comparing data sets with very different magnitudes, then we would have to be careful.

Sample Correlation Coefficient r

$$r = \frac{\hat{C}ov(x, y)}{s_x s_y}$$

where s_x and s_y are the standard deviations for our data in x and y

Sample Correlation Coefficient r

Random

```
-0.2583898341930045
```

Linear

```
1.0
```

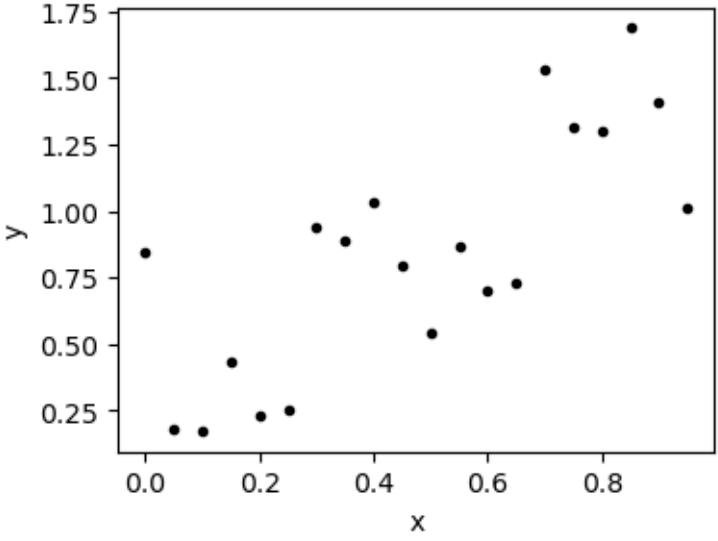
Linear with some Randomness

```
0.6876172920827351
```

YOU TRY

Calculate the sample correlation for your FOURTH example $y = -x$

Fitting a line to data – using least-squares.



Fitting a line to data – using least-squares.

The goal is to **minimize** the **sum** of **square distances** between the line $\widehat{y} = \beta_0 + \beta_1 x$ and the data point values y. Lets look at this is parts:

1. What is the distance between y and $\widehat{y}$ for each point in the data?

$$y_i - \widehat{y}_i = y_i - (\beta_0 + \beta_1 x_i)$$

This is also called the *residual* or *error* for point i in our data.

Fitting a line to data – using least-squares.

2. What is the square distance?

$$(y_i - (\beta_0 + \beta_1 x_i))^2$$

We just square the residual or error for point i in our data.

Fitting a line to data – using least-squares.

3. How do we add these up?

Using the summation notation:

$$\sum_i (y_i - (\beta_0 + \beta_1 x_i))^2$$

Now we are adding up the square error for all of the points in the data.

Fitting a line to data – using least-squares.

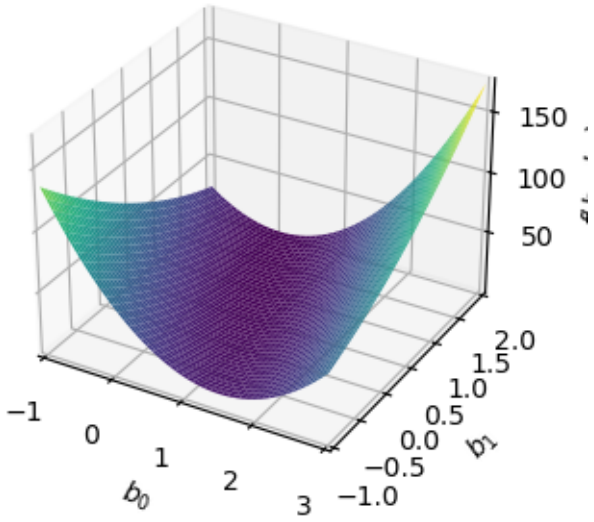
4. How can we minimize this?

Well, we might have to wait until we get some calculus under our belt before we can really see what the calculation does here. But this is a quadratic equation for β_0 and β_1. We look for points on this surface that minimize the sum of square errors.

Fitting a line to data – using least-squares.

$$
\begin{aligned}
&(0.152648555418422 - b_0)^2 + (-b_0 - 0.6b_1 + 0.959221163487778)^2 + \\
&(-b_0 - 0.5b_1 + 0.608535349889668)^2 + \\
&(-b_0 - 0.45b_1 + 0.614627161095876)^2 + \\
&(-b_0 - 0.35b_1 + 0.701728447520536)^2 + \\
&(-b_0 - 0.3b_1 + 0.916268826390829)^2 + \\
&(-b_0 - 0.15b_1 + 0.340109892676699)^2 + \\
&(-b_0 - 0.1b_1 + 0.185947932256662)^2 + \\
&(-b_0 - 0.05b_1 + 0.375015214863821)^2 + \\
&1.201219662406\,(-0.912407368198695b_0 - 0.501824052509282b_1 + 1)^2 + \\
&1.21462887098235\,(-0.907357012650559b_0 - 0.181471402530112b_1 + 1)^2 - \\
&1.50924349272702\,(-0.813992388674826b_0 - 0.203498097168706b_1 + 1)^2 - \\
&1.51495278163206\,(-0.812457124823964b_0 - 0.324982849929585b_1 + 1)^2 - \\
&1.54669992933502\,(-0.804075754933938b_0 - 0.562853028453757b_1 + 1)^2 - \\
&1.88875389054787\,(-0.727632877428953b_0 - 0.582106301943162b_1 + 1)^2 -
\end{aligned}
$$

Fitting a line to data – using least-squares.



Fitting a line to data – using least-squares.

If we solve (using calculus) we find that

$$\beta_1 = \frac{n \sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n \sum_i x_i^2 - (\sum_i x_i)^2)}$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x}$$

where $\bar{x}$ and $\bar{y}$ are the averages of the x_i and y_i data points respectively. WHAT A MESS :)

Python is Amazing – np.polyfit()

Luckily for us, we don't have to do this calculation by hand. We can us polyfit to get the line. Below we will have python calculate the coefficients for us and then we will plot the resulting line.

BEWARE - np.polyfit() returns the coefficients in reverse order [beta1,beta0]

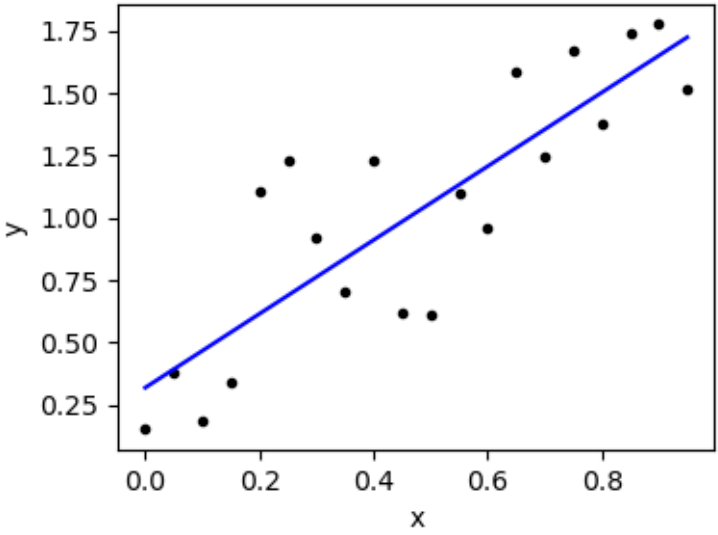
Python is Amazing – np.polyfit()

```
betas = np.polyfit(xdata,ydata,1)

xfit = xdata
yfit = betas[0]*xfit+betas[1]

plt.plot(xdata,ydata,'k.')
plt.plot(xfit,yfit,'b-')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

Python is Amazing – np.polyfit()

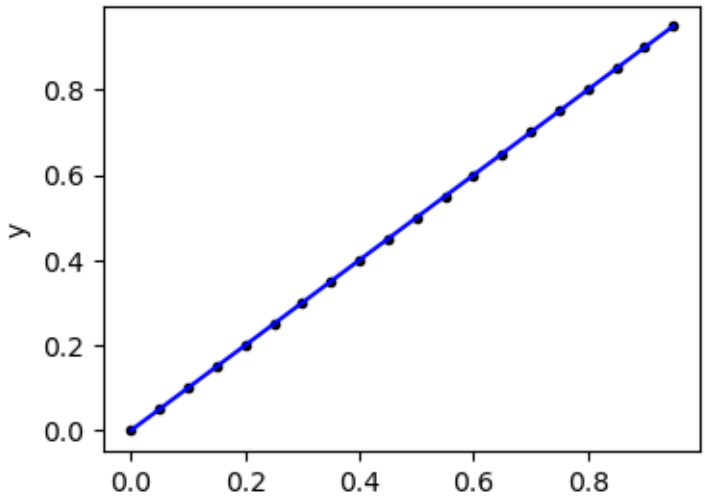


PAUSE - What do we have so far?

- Covariance np.cov() will measure how much the points in a sample data set vary together.
- Correlation np.corcoeff() gives us a coefficient to tell us how correlated the data is - "normalizes" for the magnitude of the numbers.
- We can find a line of best fit (Least Squares) np.polyfit(x,y,1)

BUT How good is the fit?

Consider the picture below. Is the linear fit better in this case? Why?



R^2 A measure of fit.

R^2 measures the amount of variation in the data that is explained by the model. We can compare the model data (the line) and the sample data (the points). Here are some things we might consider:

How much of the variation in the data IS NOT described by the model?

We can look at the distance between the line and each of the sample points to see the error in the model or residual - aka Residual sum of squares

$$SS_{res} = \sum_i (y_i - \hat{y}_i)^2$$

How much of the variation in the data IS described by the model?

We can look at the difference between the line and the average (or mean) of the data.

$$SS_{reg} = \sum_i (\hat{y}_i - \bar{y})^2$$

R^2 is defined as

$$R^2 = \frac{SS_{reg}}{SS_{reg} + SS_{res}} = \frac{SS_{reg}}{SS_{total}}$$

so R^2 measures the ratio of the variation that is explained by the model to the total variation in the data.

- if $R^2 = 0$ then none of the data's variation is explained by the model.
- if $R^2 = 1$ the all of the data's variation is explained by the model.

R^2 in Python

```
from sklearn.metrics import r2_score

# R^2 for the "messy" data
r2_score(ydata,yfit)
```

```
0.7095804639917491
```

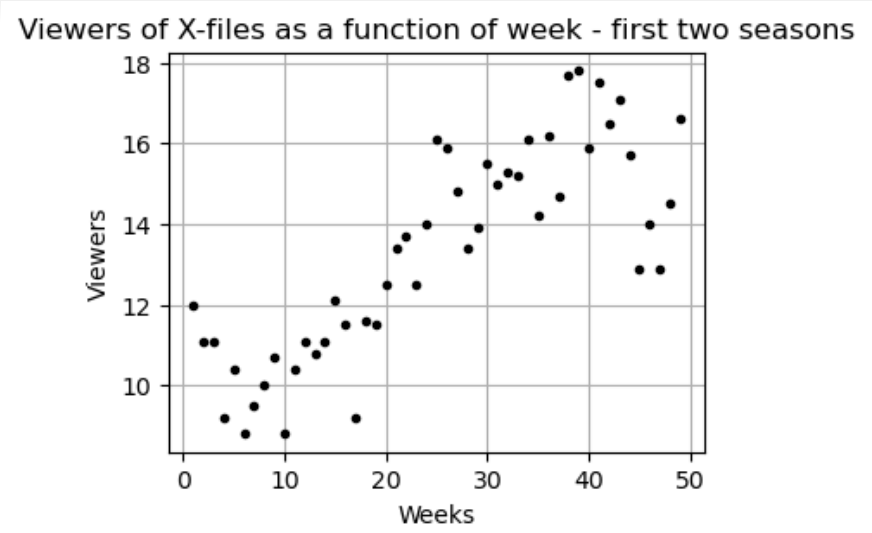
```
# R^2 for the linear data
r2_score(y2,yfit2)
```

```
1.0
```

Your Homework - Predict Viewership for the Show X-files

Below is some data for viewership of the TV show The X-files. It was SUPER popular in the 90's. The first few cells below gather the data and plot it. You can just run these cells.

https://en.wikipedia.org/wiki/List_of_The_X-Files_episodes


Viewers of X-files as a function of week - first two seasons
Viewers
Weeks
18
16
14
12
10
0
10
20
30
40
50

Please answer the following questions:

1. Does there seem to be a relationship between the week number and the viewership?
2. What do the sample covariance and sample correlation tell you about this data? Explain why your answer make sense.
3. Find a straight line that fits this data (Linear Regression).
   - Write down the equation you found.
   - Plot the linear fit and the original data on the same plot.
   - How does it look?
   - Based on your line what should viewership in the first week of the third season be?
4. Calculate the R^2 value and talk about what this means in terms of your data and your linear fit.

EXTRA - Below I load data for season three of the series. How well does your linear fit match season three. If you did a linear fit just on season three would you have the same line or a different line? WHat does the difference in these lines mean?f