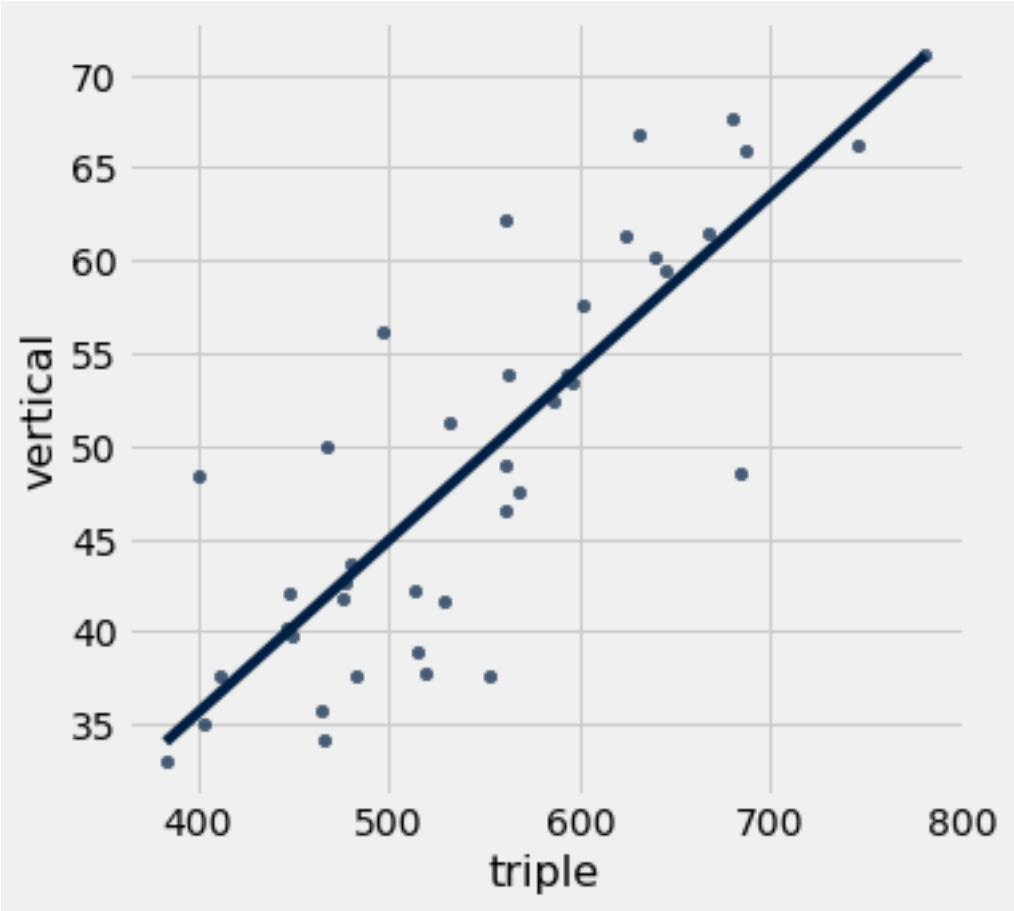
Question 1.3. Before running a regression, it's important to see what the data looks like, because our eyes are good at picking out unusual patterns in data. Draw a scatter plot, **that includes the regression line**, with the triple jump distances on the horizontal axis and the vertical jump heights on vertical axis. **(5 points)**

See the documentation on `scatter` here for instructions on how to have Python draw the regression line automatically.

Hint: The `fit_line` argument may be useful here!

```
In [11]: jumps.scatter('triple', 'vertical', fit_line=True) #SOLUTION
```



Question 1.4. Does the correlation coefficient r look closest to 0, .5, or -.5? Explain. **(5 points)**

Type your answer here, replacing this text.

SOLUTION: It definitely looks closest to .5. The two variables are positively associated, so it's not -.5. The data roughly follow a line, so the correlation is probably closer to .5 than to 0.

Question 1.8. Do you think it makes sense to use this line to predict Edwards' vertical jump? **(5 points)**

Hint: Compare Edwards' triple jump distance to the triple jump distances in `jumps`. Is it relatively similar to the rest of the data (shown in Question 1.3)?

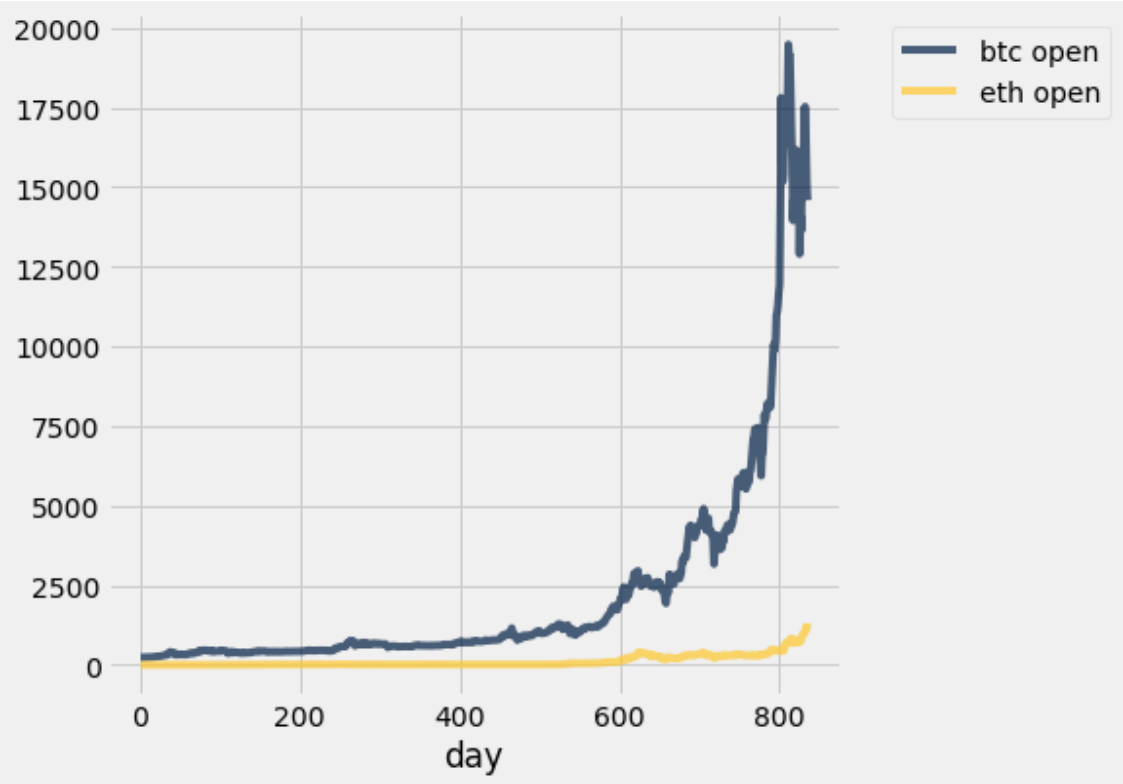
Type your answer here, replacing this text.

SOLUTION: No; we have absolutely no information on the triple jump distances in any remote region near 18.29 meters, so it's not smart to make an estimate for it based on this data that is outside our observed range. In fact, this is around 7 cm higher than the current world record according to Guinness. That's not totally implausible, but it seems unlikely.

Question 2.1. In the cell below, create an overlaid line plot that visualizes the BTC and ETH open prices as a function of the day. Both BTC and ETH open prices should be plotted on the same graph. **(5 points)**

Hint: Section 7.3 in the textbook might be helpful!

```
In [27]: # BEGIN SOLUTION NO PROMPT
         btc_only = btc.select('day', 'open').relabeled('open', 'btc open')
         both = btc_only.with_column('eth open', eth.column('open'))
         both.plot('day')
         # END SOLUTION
         """ # BEGIN PROMPT
         # Create a line plot of btc and eth open prices as a function of time
         ...
         """; # END PROMPT
```

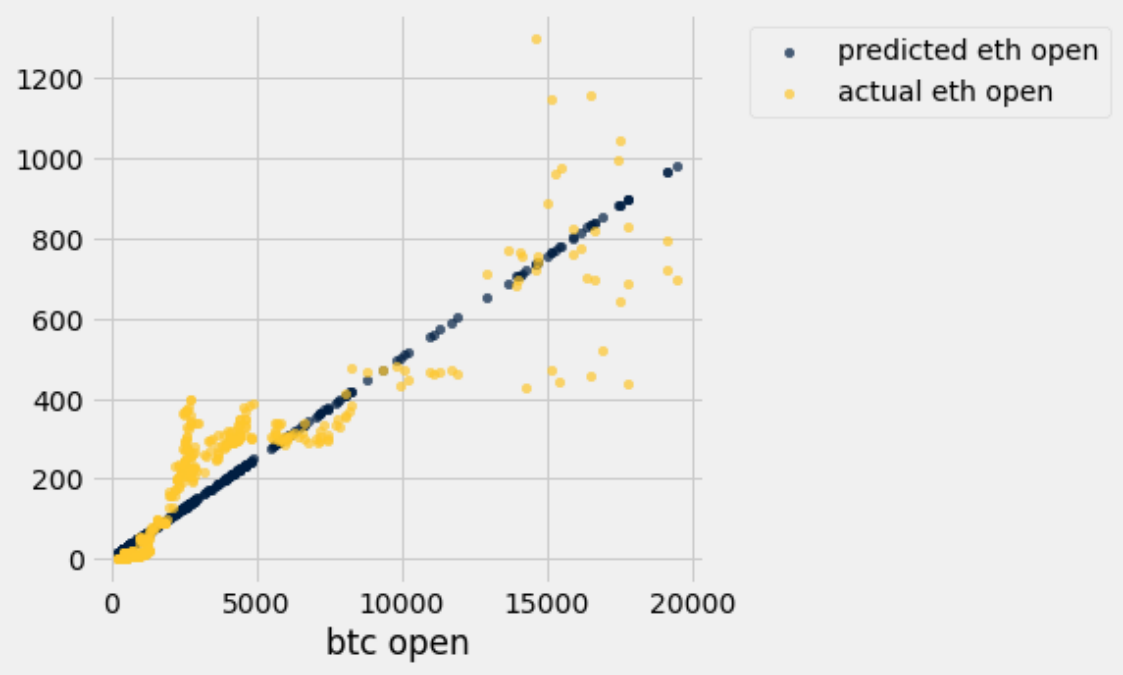


Question 2.4. Now, using the `eth_predictor` function you just defined, make a scatter plot with BTC prices along the x-axis and both real and predicted ETH prices along the y-axis. The color of the dots for the real ETH prices should be different from the color for the predicted ETH prices. **(5 points)**

Hint 1: An example of such a scatter plot is generated can be found here.

Hint 2: Think about the table that must be produced and used to generate this scatter plot. What data should the columns represent? Based on the data that you need, how many columns should be present in this table? Also, what should each row represent? Constructing the table will be the main part of this question; once you have this table, generating the scatter plot should be straightforward as usual.

```
In [35]: btc_open = btc.select('open') # SOLUTION
         eth_pred = btc_open.with_column("predicted eth open", btc.apply(eth_predictor, "open")) # SOLU
         eth_pred_actual = eth_pred.with_column("actual eth open", eth.column("open")) # SOLUTION
         eth_pred_actual.relabeled("open", "btc open").scatter('btc open') # SOLUTION
```



Question 2.5. Considering the shape of the scatter plot of the true data, is the model we used reasonable? If so, what features or characteristics make this model reasonable? If not, what features or characteristics make it unreasonable? **(5 points)**

Type your answer here, replacing this text.

SOLUTION: This is not a great model for this particular data, as the true data are not even close to being linear (we can see the actual ETH open prices may follow a higher-order pattern). We have produced the line of best fit, but that doesn't mean much when the data points that the line is best fit for is a bad set of non-linear trending data; in other words, there will always be a line of best fit for any data, but that does not mean that the data itself is best fit by a linear model.

Question 3.4. Suppose that we create another model that simply predicts the average outcome regardless of the value for spread. Does this new model minimize the least squared error? Why or why not? **(5 points)**

Type your answer here, replacing this text.

SOLUTION: The new predictor is a horizontal **line** that passes through the average value for outcome. Therefore. it does not minimize least squared error, as only the regression line is the unique straight line that minimizes least squared error among all straight lines.

Question 3.9. The slope and intercept pair you found in Question 3.8 should be very similar to the values that you found in Question 3.3. Why were we able to minimize RMSE to find the same slope and intercept from the previous formulas? **(3 points)**

Type your answer here, replacing this text.

SOLUTION: The regression line is the unique straight line (in other words, the unique slope/intercept pair) that minimizes RMSE. Therefore, we can also find the regression line by finding the slope and intercept values that minimize RMSE.