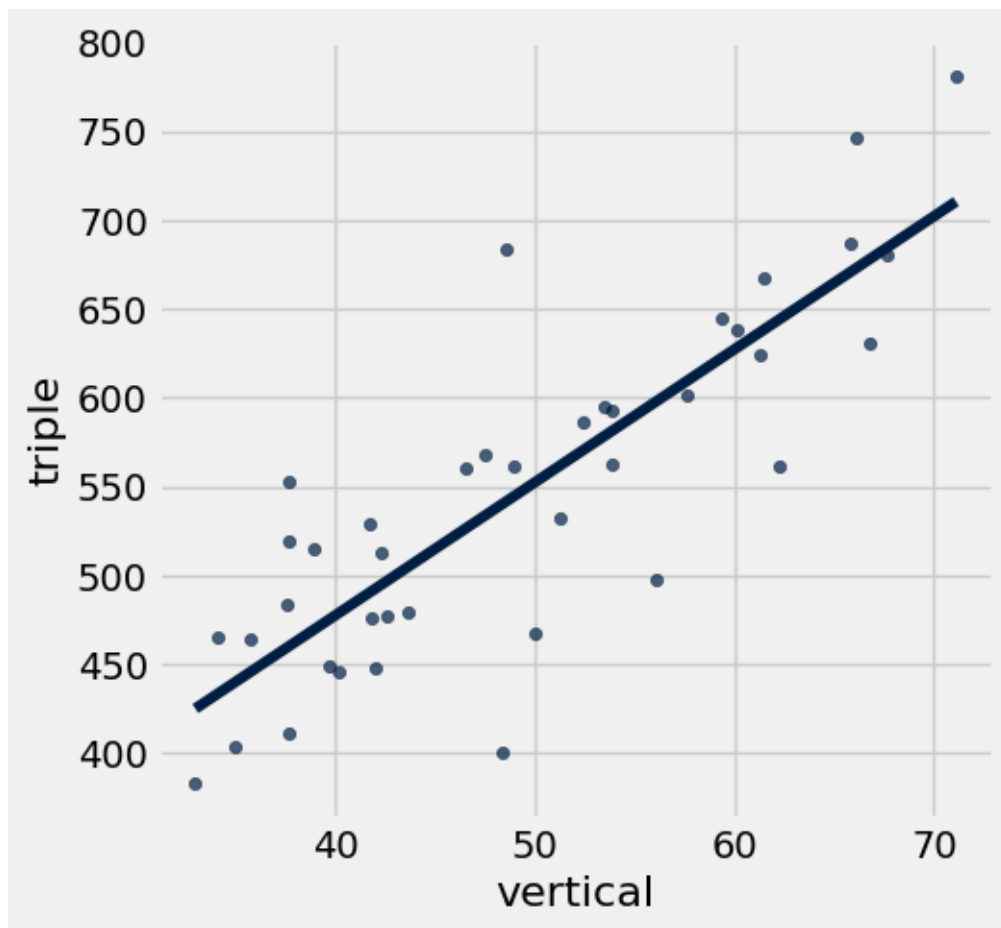


Question 1.3. Before running a regression, it's important to see what the data looks like, because our eyes are good at picking out unusual patterns in data. Draw a scatter plot, **that includes the regression line**, with the triple jump distances on the horizontal axis and the vertical jump heights 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 [8]: jumps.scatter('vertical', 'triple', fit_line = True)
```



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.

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.

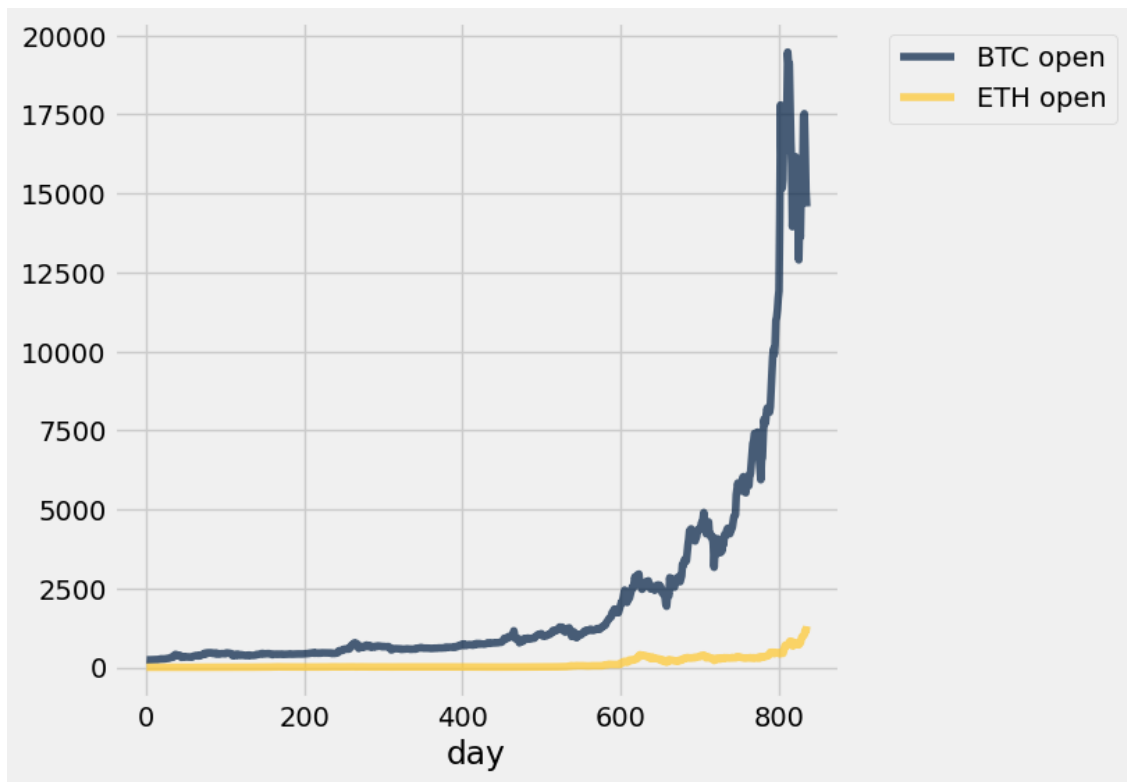
Question 2.1. In the cell below, create an overlaid line plot that visualizes the BTC and ETH open prices as a function of the day. Both BTC and ETH open prices should be plotted on the same graph. **(5 points)**

Hint: [Section 7.3](#) in the textbook might be helpful!

```
In [18]: btc2 = btc.select('day', 'open').relabel('open', 'BTC open')
eth2 = eth.select('day', 'open').relabel('open', 'ETH open')
btc_and_eth = btc2.join('day', eth2 )
btc_and_eth
```

```
Out[18]: day | BTC open | ETH open
1      | 239.02   | 0.579414
2      | 236.64   | 0.661192
3      | 236      | 0.734307
4      | 237.26   | 0.683732
5      | 237.2    | 0.678783
6      | 238.53   | 0.686343
7      | 238.15   | 0.666784
8      | 240.36   | 0.622218
9      | 246.17   | 0.650515
10     | 243.07   | 0.609501
... (825 rows omitted)
```

```
In [19]: # Create a line plot of btc and eth open prices as a function of time
btc_and_eth.plot('day')
```



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)**

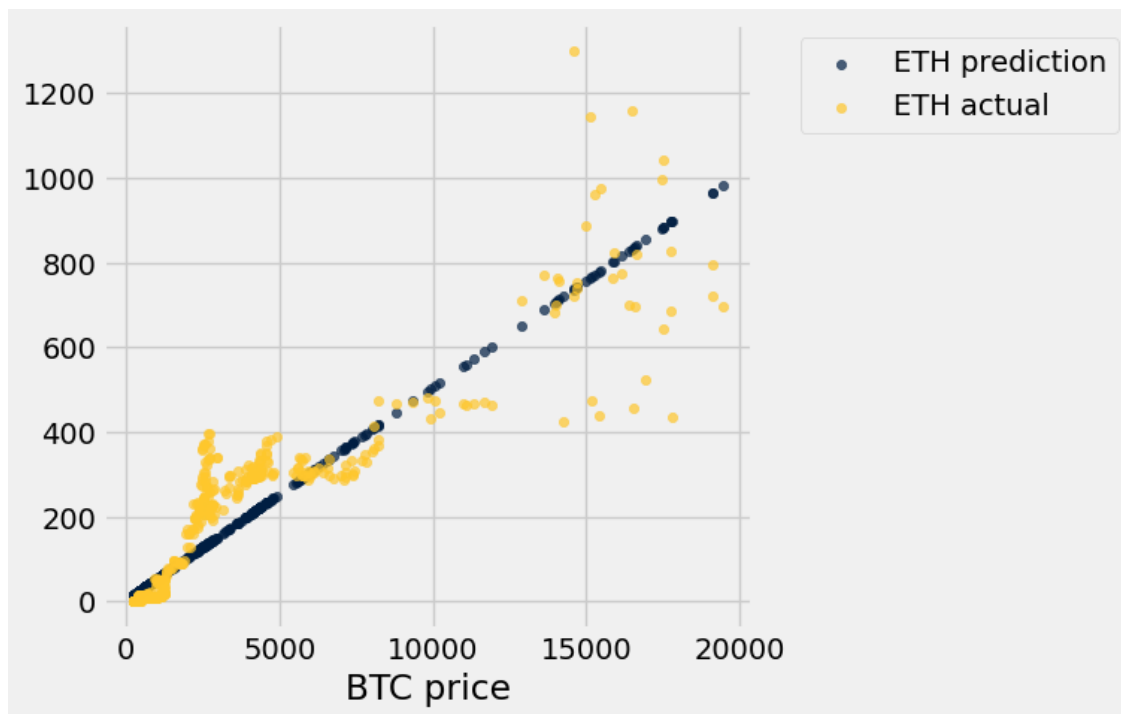
Hint 1: An example of such a scatter plot is generated can be found [here](#).

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 [30]: btc_open = btc_and_eth.column('BTC open')
eth_pred = btc_and_eth.apply(eth_predictor, 'BTC open')
eth_pred_actual = btc_and_eth.column('ETH open')

plot_tbl = Table().with_columns(
    'BTC price', btc_open,
    'ETH prediction', eth_pred,
    'ETH actual', eth_pred_actual
)

plot_tbl.scatter('BTC price', ['ETH prediction', 'ETH actual'])
```



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.

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.

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.

