Motivation: Cryptocurrencies are all the hype now. But with global semiconductor shortage, graphics cards (GraphicalProcessingUnits) are a popular traded commodity in 2020-2021. As a brief background, GPUs are used to “mine” cryptocurrencies. When prices of crypto are high, it is profitable for some to buy GPUs and use them to mine crypto and make profit. Here we take a look at historical data from 2013-2018 of a few crypto coin prices and GPU prices for exploration.

Before making a model, it’s always good to have a first impression of the data you have. You can plot the price of whichever cryptocurrency and/or GPUs over time and note any trends, seasonality, cycles or shocks you may observe.

We’ll apply the concepts we learned in linear regression to time series data. Previously we had Y = b0 + b1 \* X + e as a basic linear regression model. With time series data, we are assuming that the time series of interest y has a linear relationship with some other time series x when we use Yt = b0 + b1 \* Xt + et with the assumptions that the error term has 0 mean, constant variance, no correlation with X, no autocorrelation, and that it’s normally distributed. However typically in time series data there is often autocorrelation (some data point at time t is correlated with some past or future datapoint).

1. Assume the data we use meet the assumptions, build a linear regression model of the price of GeForce Titan X regressed on the historical price of Bitcoin. There are a few skus of the GPU, choose one that gives you a reasonable amount of data points to fit a model. (I used Id=1975) Report how many data points your model used and what time period it used. Plot the price of BTC and the GPU over the time period you used and plot residuals vs fitted values.
2. Choose any other GPU from the given data and build a model with the prices of all mineable coins as independent variables in a similar fashion as the above. What can or can’t this model tell us? What are the shortcomings of using this approach to model time series data?

Why might standardizing your time series data be helpful in our type of modeling?

Given the type of trend you see, did our assumptions for the linear model hold? How might using a model such as log(yt) - log(yt-1) = log(yt / yt-1) be a better choice?

Autoregression: one method of modeling time series data is using autoregression. In other words, lagging the data by some time step and using that as our independent variable. Yt = b0 + b1 \* Yt-n + et . Using n=1, that means our prediction of the current value is dependent on the previous value (and recursively until the first data point Y0). The concept can be extended to multiple ns. (If you’re interested you can read more about AR(p) models, acf, and pacf)

1. As an exercise, build an autoregressive model for the price of any cryptocurrency using a year’s worth of data from 01-01 to 12-31 with 5 lag variables (Yt-1, Yt-2, Yt-3, Yt-4, Yt-5). Note, this means you may need data from before 2015-01-01 if treating that as Y0, you’d need Y0-5 more data points.
2. Now, try using 2 year’s worth of data from 2014-01-01 to 2015-12-31 with 30 lag variables and compare the model’s prediction for 2016-01-01 to 2017-12-31 with the true prices of that time period

A more advanced model built on the concepts of the autoregressive model is the ARIMA model where a difference term and moving average are considered in the model. Though that model depends on the data being stationary. A stationary time series data is one where

1. Mean of series not a function of time
2. Variance of series not a function of time
3. Covariance of ith term and i+m term should not be function of time

Connecting back to the models you build in 3, 4, why might differencing the data before building the model be a better approach? How do you determine what degree you want to difference your data? You can (optionally) replicate your steps in 3 or 4 by differencing the data using the diff function in r. How do you interpret a model built on differenced data?

Extra Cred:

1. We’ll combine data from GPUs and crypto coins. Aggregate the data from daily to monthly for the mineable crypto coins. Make a AR model with GPU\_Pricet = b0 + b1 \* BTC\_Pricet-1 + b2 \* ETH\_Pricet-1 + b3 \* XMR\_Pricet-1 + b4 \* DOGE\_Pricet-1 + b5 \* LTC\_Pricet-1 over ¾ of the data’s overlapping time period. (For instance, there’s data for BTC since 2013 but ETH from late 2015. Find their mutual overlapping time, use the first ¾ as your training and last ¼ as testing)

Further reading:

In practice, many types of time-series specific models have been invented that work much better than what we did here. ARIMA model is one that uses the autoregression concept along with differencing and moving average. There are also multivariate (Vector Autoregression) models to explore. [Suggested book](https://otexts.com/fpp3/dynamic.html) to follow for more information