



CRYPTO WEB

TOKEN ANALYSIS

UOFT SCS FinTech Boot Camp | Toronto
Amira Ali | Nadeem Hassan | Jeff Zhang
April 18th, 2022

TOKENS ANALYZED

- Solana
- Tron
- Cronos
- Fantom
- Ethereum
- Binance
- Poly
- XRP



SUMMARY

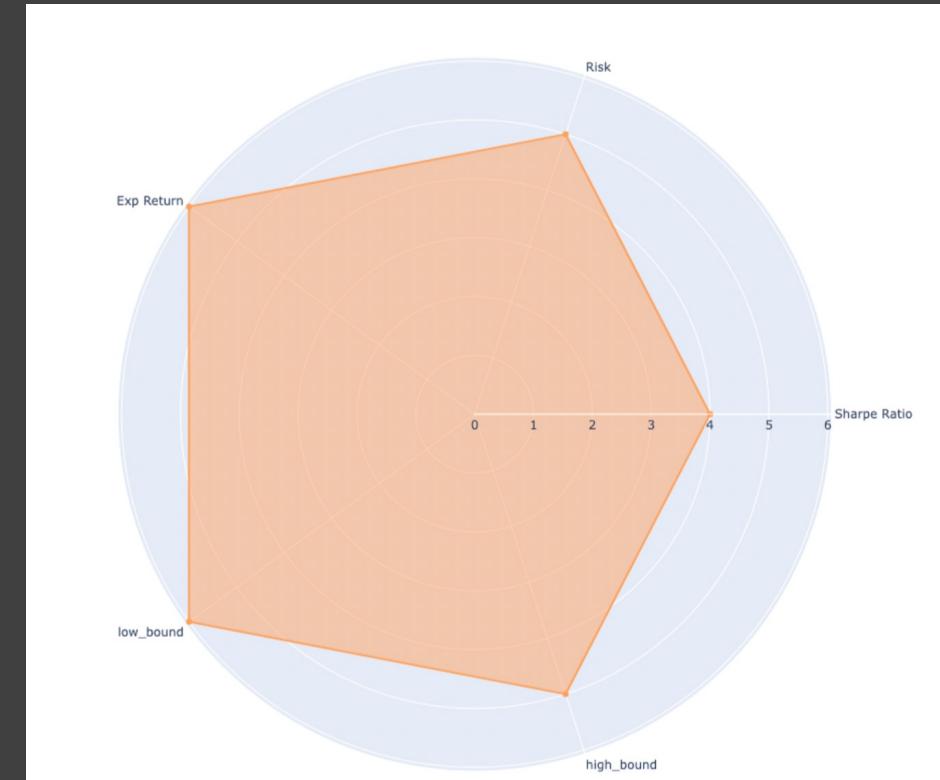
OBJECTIVE

To help investors understand which blockchain best suits their needs based on the performance of their most popular tokens.

QUESTIONS TO ANSWER

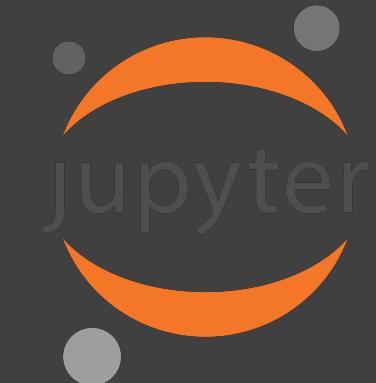
- Which token provides the highest returns?
- Which token is the least / most risky?
- Which token has better future projections?
- How do these compare to a benchmark?

PRODUCT DEMO



TECHNOLOGIES USED

CoinGecko API



Jupyter Notebook

Python



plotly



Plotly

QUESTIONS

How do we make this fair?

- Ensure there is data available for each token from the desired start and end date.
- Date range :April 2020 – April 2022

Which token provides the highest returns?

- Closing prices for each token
- Calculate returns on each
- CoinGecko API

Which token is the least/most risky?

- Beta (mean, std, variance, covariance)
- Plot risk vs return

QUESTIONS cont.

Which token has better future projections?

- Monte Carlo Simulation
- Need to decide an appropriate # of simulations

How do these compare to a benchmark?

- As a measure of comparison
- Create a balanced ETF – take an investment amount and deploy an even amount to each token.

DATA COLLECTION

In [1]:

```
pip install pycoingecko
```

In [2]:

```
from pycoingecko import CoinGeckoAPI
cg = CoinGeckoAPI()
```

In [7]:

```
def get_historical_data(cryptocurrency, fiat_currency, number_of_days):
    historic_price = cg.get_coin_market_chart_by_id(cryptocurrency, fiat_currency, number_of_days)
    prices = [price[1] for price in historic_price['prices']]
    return prices
```

DATA COLLECTION cont.

```
Out[3]: [{}{'id': 'binance',
    'name': 'Binance',
    'year_established': 2017,
    'country': 'Cayman Islands',
    'description': '',
    'url': 'https://www.binance.com/',
    'image': 'https://assets.coingecko.com/markets/images/52/small/binance.jpg?1519353250',
    'has_trading_incentive': False,
    'trust_score': 10,
    'trust_score_rank': 1,
    'trade_volume_24h_btc': 319257.20564049354,
    'trade_volume_24h_btc_normalized': 319257.20564049354},
    {'id': 'okex',
    'name': 'OKX',
    'year_established': 2013,
    'country': 'Belize',
    'description': '',
    'url': 'https://www.okx.com',
    'image': 'https://assets.coingecko.com/markets/images/96/small/WeChat_Image_20220117220452.png?1642428
377',
    'has_trading_incentive': False,
    'trust_score': 10,
    'trust_score_rank': 2,
    'trade_volume_24h_btc': 85569.98612513798,
    'trade_volume_24h_btc_normalized': 85569.98612513798},
    {'id': 'crypto_com',
    'name': 'Crypto.com Exchange',
    'year_established': 2019,
    'country': 'Cayman Islands',
    'description': 'Crypto.com Exchange is the best place to trade crypto, with deep liquidity, low fees and best execution prices, users can trade major cryptocurrencies like Bitcoin, Ethereum, and many more and receive great CRO-powered rewards',
    'url': 'https://crypto.com/exchange',
    'image': 'https://assets.coingecko.com/markets/images/589/small/Crypto.jpg?1629861084',
    'has_trading_incentive': False,
    'trust_score': 10,
    'trust_score_rank': 3,
    'trade_volume_24h_btc': 67495.30052510073,
    'trade_volume_24h_btc_normalized': 67495.30052510073},
    {'id': 'gdax',
    'name': 'GDAX',
    'year_established': 2015,
    'country': 'United States'}
```

```
In [32... historic_pricetron = cg.get_coin_market_chart_by_id("tron", "usd", 730)
tronprice=pd.DataFrame(historic_pricetron['prices'], columns=['Time', 'Tron Price'])
tronprice
```

```
Out[32...           Time  Tron Price
0  1586476800000  0.013435
1  1586563200000  0.012504
2  1586649600000  0.012500
3  1586736000000  0.012708
4  1586822400000  0.012576
...
726 1649203200000  0.069441
727 1649289600000  0.063205
728 1649376000000  0.064453
729 1649462400000  0.062606
730 1649519815000  0.062542
731 rows × 2 columns
```

```
In [33... tronprice.to_csv("tron.csv")
```

DATA COLLECTION cont.

```
In [27...]  
all_vol_df = pd.concat([solana_vol_df, bnb_vol_df, cron_vol_df, eth_vol_df, fantom_vol_df, poly_vol_df,  
all_vol_df.dropna(inplace = True)  
all_vol_df
```

	Solana Volume	Binance Volume	Cronos Volume	Ethereum Volume	Fantom Volume	Polymath Volume	Tron Volume	XRP Volume
Time								
2020-04-11	4.945869e+07	3.790711e+08	6.679442e+06	1.112899e+10	2.339508e+06	1.217371e+06	1.101730e+09	1.781002e+09
2020-04-12	3.782101e+07	4.049799e+08	6.662411e+06	1.300280e+10	1.684559e+06	1.555734e+06	1.092810e+09	2.018968e+09
2020-04-13	1.957921e+07	4.153753e+08	5.896746e+06	1.328648e+10	1.955905e+06	1.867479e+06	1.297979e+09	2.228365e+09
2020-04-14	1.752353e+07	4.319139e+08	5.457073e+06	1.146843e+10	1.698655e+06	2.018466e+06	1.297749e+09	1.862120e+09
2020-04-15	1.265954e+07	3.933861e+08	5.023388e+06	8.578827e+09	1.657415e+06	1.645990e+06	1.278331e+09	1.937689e+09
...
2022-04-05	2.136267e+09	2.111671e+09	7.151299e+07	1.511406e+10	5.501345e+08	1.582625e+07	1.400060e+09	2.017747e+09
2022-04-06	3.198016e+09	2.227804e+09	8.810043e+07	2.610976e+10	9.302908e+08	2.655821e+07	1.448266e+09	3.043359e+09
2022-04-07	2.381088e+09	1.902419e+09	6.680808e+07	1.614978e+10	5.494603e+08	1.649691e+07	9.209929e+08	2.580151e+09
2022-04-08	2.314899e+09	1.895965e+09	6.642291e+07	1.752394e+10	4.576184e+08	1.403973e+07	8.567597e+08	2.728014e+09
2022-04-09	1.670543e+09	1.539115e+09	4.898937e+07	1.250607e+10	3.897682e+08	1.332954e+07	9.913204e+08	2.248539e+09

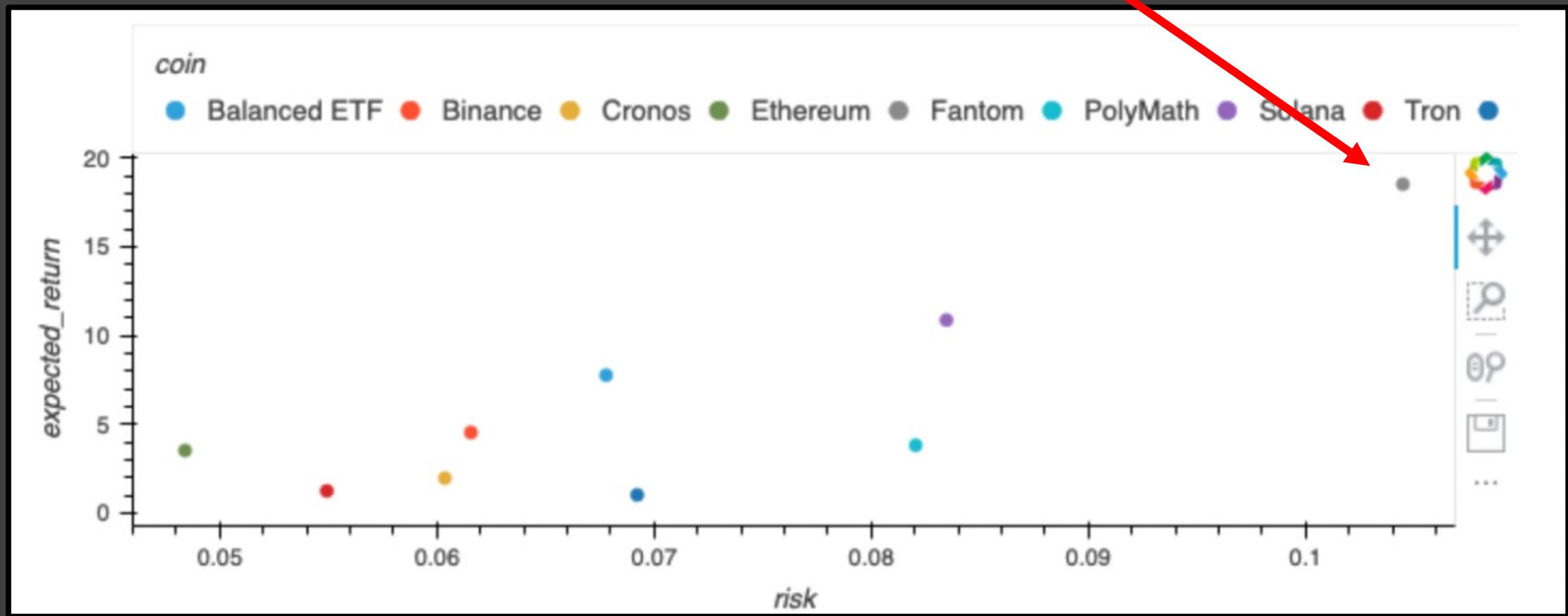
729 rows x 8 columns

- Concatenate the data frames
- Prices, Volume, Market Capitalization
- All converted to csv files for further analysis

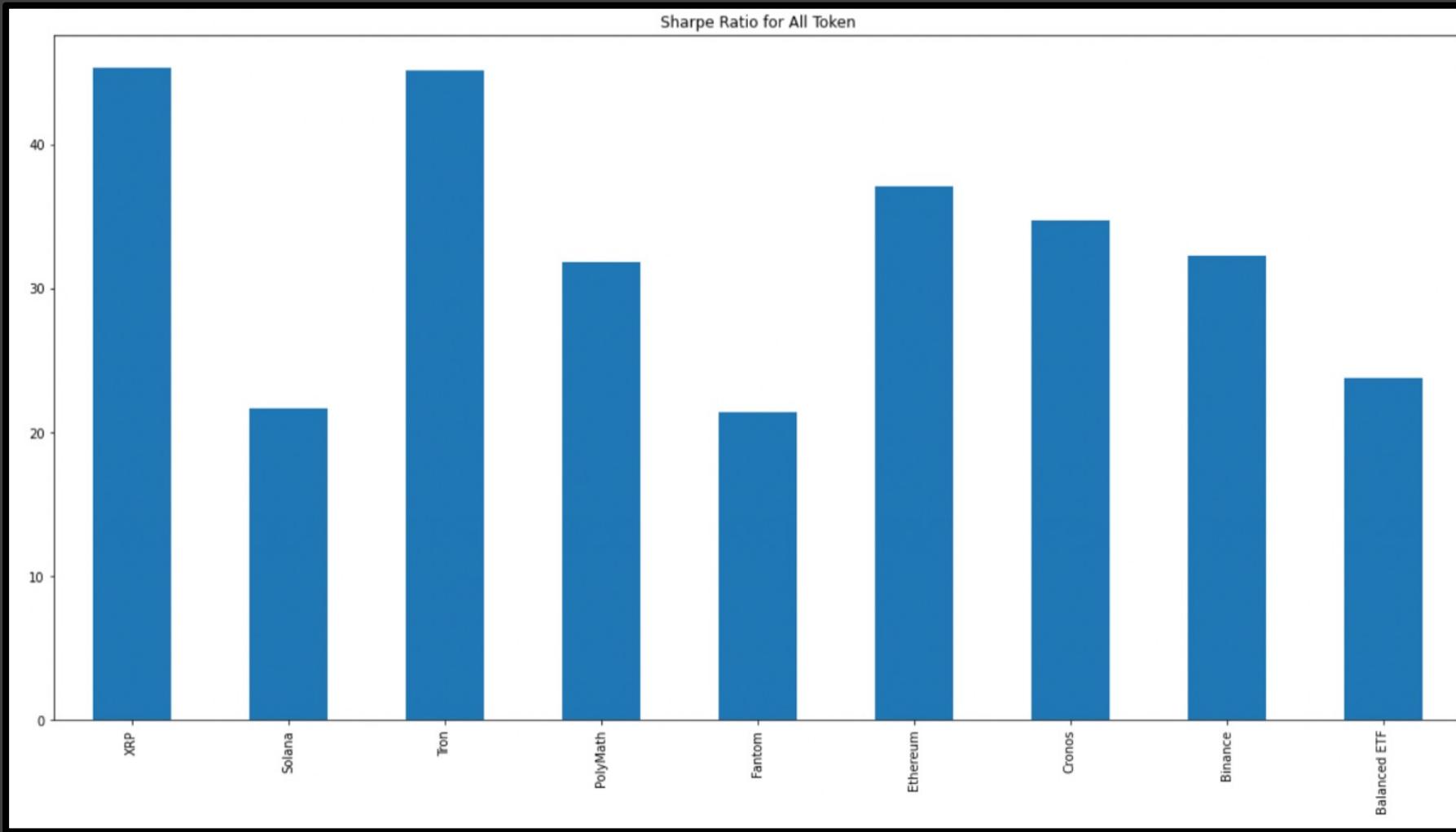
DATA COLLECTION Limitations

- **Limited at 50 calls per minute**
 - Slowed down our data collection
 - Reduced efficiency
- **Limited by time frame**
 - Adjust models to account for smaller date range
 - Monte Carlo simulation – 30 day projections (~5% of data points)
- **Ethereum skewed data upwards**
 - Ethereum has considerably higher prices than the other tokens
 - No benchmark evaluation
- **We started with AVAL but don't have 2 years worth of data**
- **Creative way to do the Markowitz Bullet**

DATA ANALYSIS I

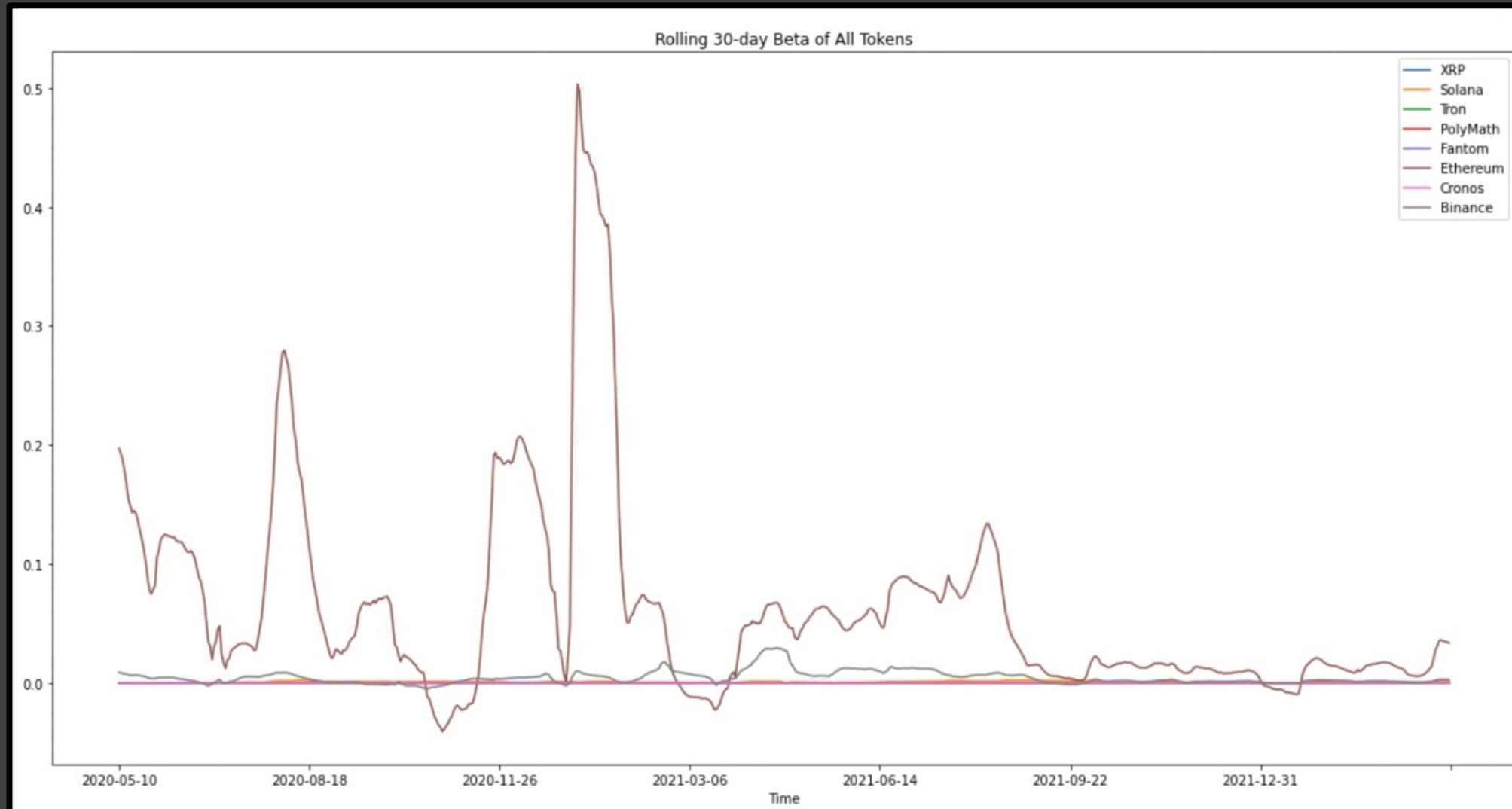


DATA ANALYSIS II



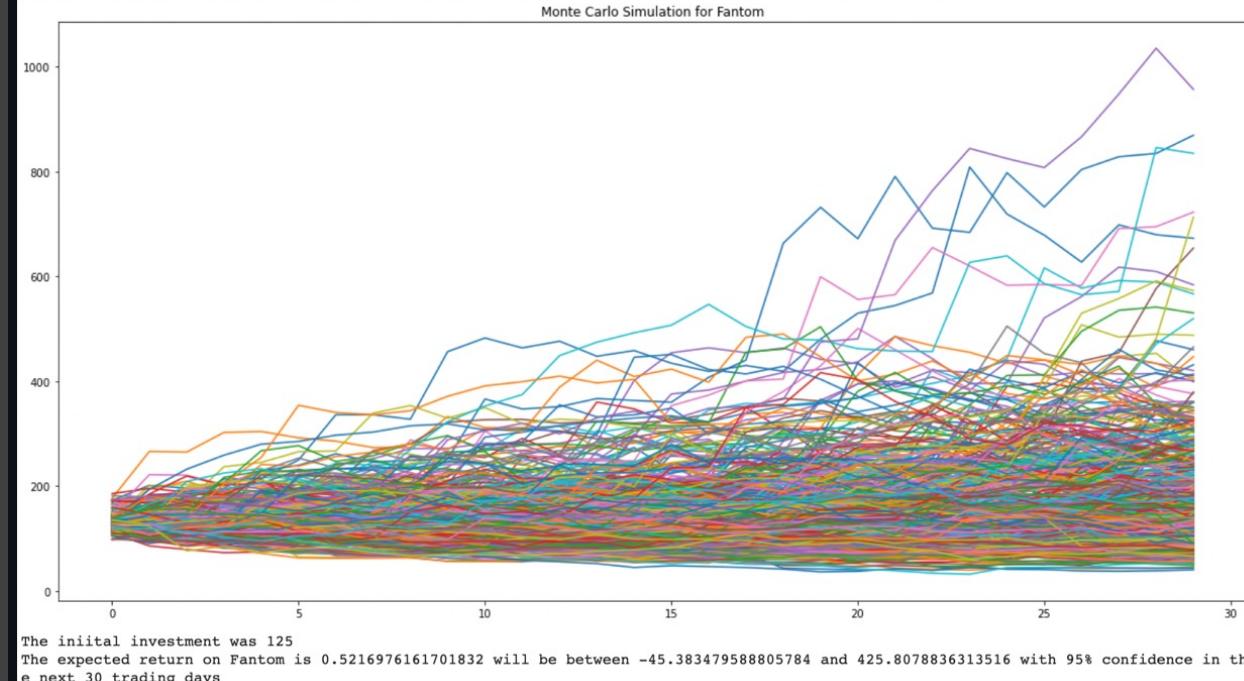
```
(<AxesSubplot:title={'center':  
    XRP      45.371852  
    Solana   21.657771  
    Tron     45.183341  
    PolyMath 31.844010  
    Fantom   21.377277  
    Ethereum 37.109490  
    Cronos   34.750156  
    Binance  32.273410  
    Balanced ETF 23.774779  
    dtype: float64})
```

DATA ANALYSIS III

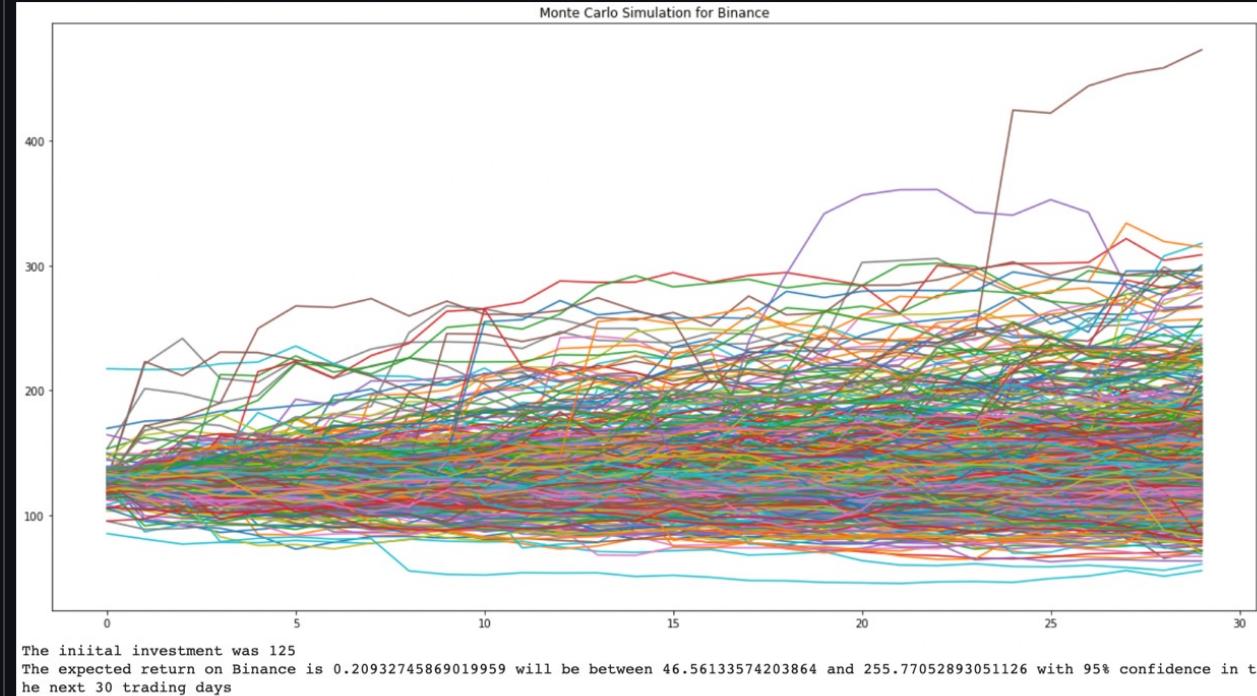


DATA ANALYSIS IV

Fantom



Binance

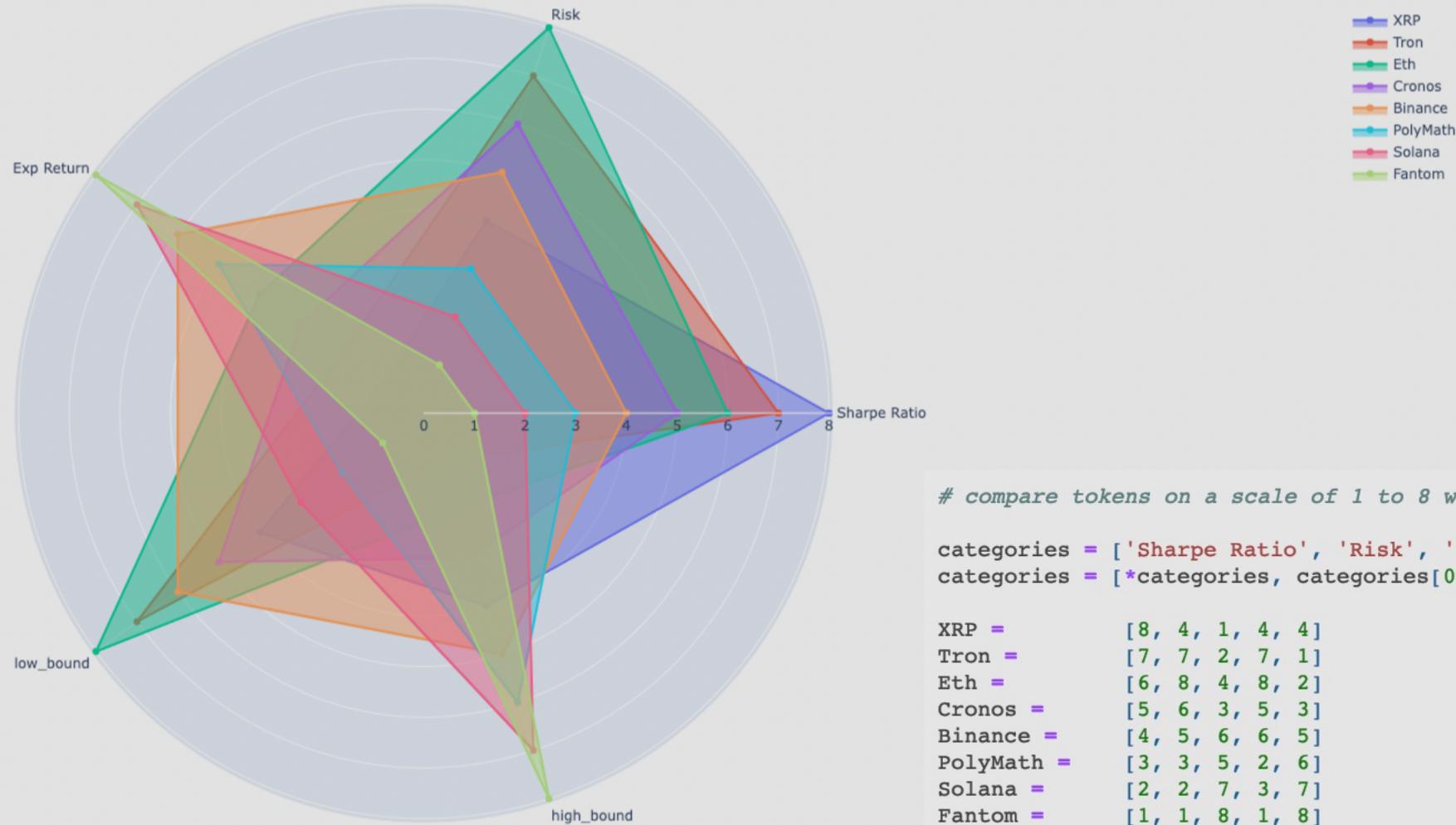


DATA ANALYSIS V

DATA TO SORT & RANK

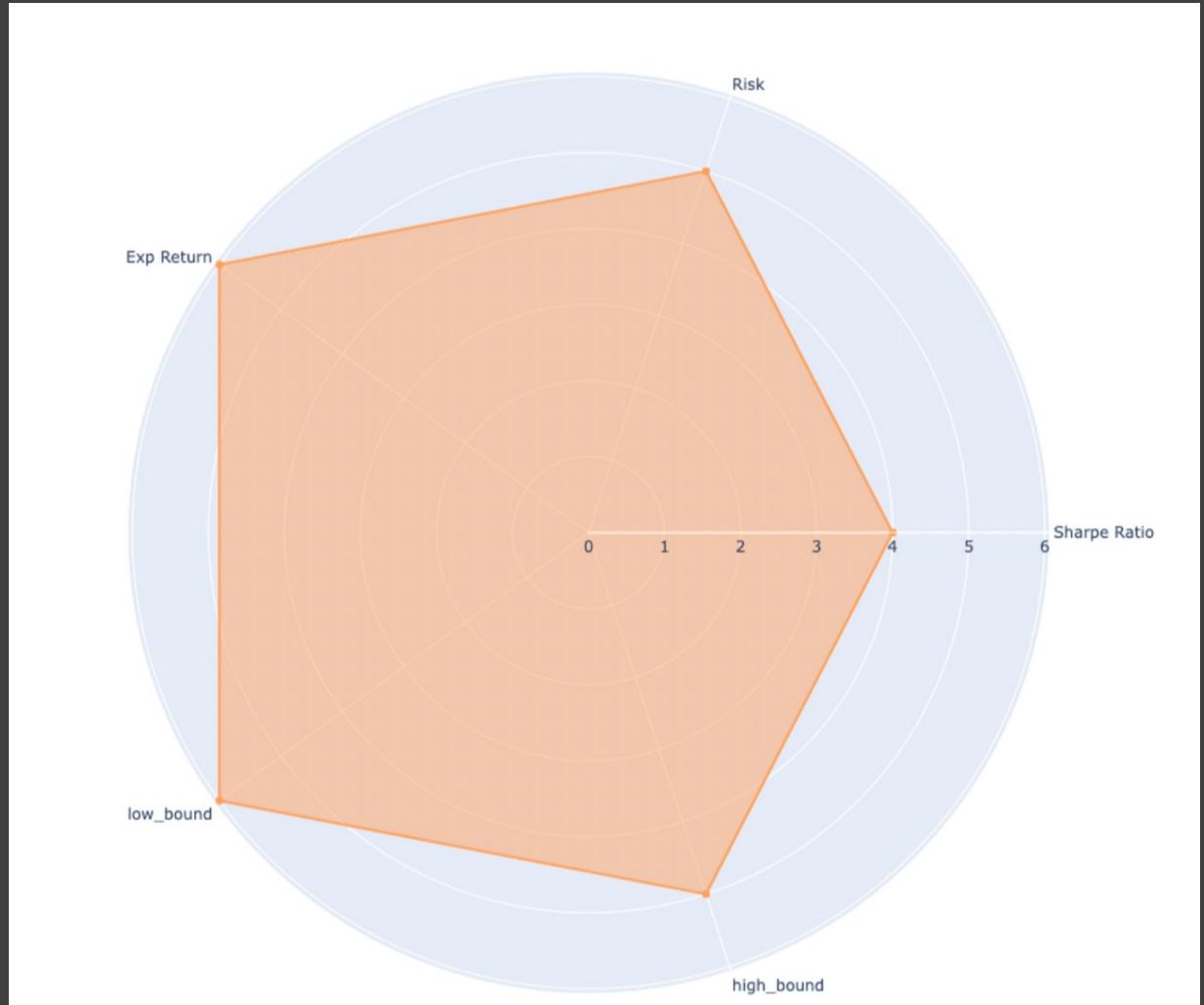
ticker	sharpe_ratio	sharpe_ratio rank	low_bound	low_bound rank	high_bound	high_bound rank	risk	risk rank	expected_return	expected_return rank
XRP	45.371852	8	34.583045	4	252.770422	4	0.069221	4	1.012270	1
Solana	21.657771	2	4.800165	3	331.727586	7	0.083462	2	10.873725	7
Tron	45.183341	7	51.426254	7	232.821847	1	0.054925	7	1.236821	2
PolyMath	31.844010	3	14.896136	2	299.345353	6	0.082045	3	3.812742	5
Fantom	21.377277	1	-45.603645	1	416.230926	8	0.104483	1	18.537073	8
Ethereum	37.109490	6	71.521669	8	219.718008	2	0.048396	8	3.514322	4
Cronos	34.750156	5	53.362272	5	230.433409	3	0.060364	6	1.957541	3
Binance	32.273410	4	54.955086	6	244.282062	5	0.061563	5	4.533797	6

TOOL



EXPECTATIONS VS REALITY

- We expected Ethereum to be the most "balanced" token / blockchain given that they had a relatively low risk and modest returns
- Based on our findings, Binance seems to be the most balanced. Every other token is skewed to one or more factors
 - *** Each investor should use this tool based on their own personal objectives



POSTMORTEM

- User count

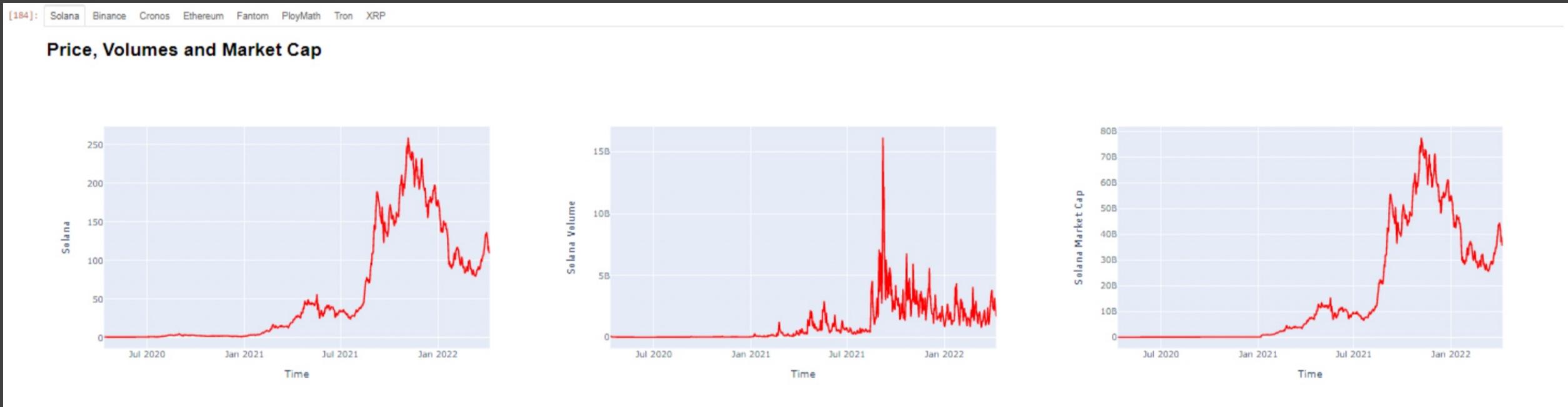
```
from datetime import date, timedelta
import datetime
start_date = date(2022, 2, 17)
end_date = date(2022, 4, 8)
sub = []
delta = timedelta(days=1)
while start_date <= end_date:
    start_date += delta
    sd = start_date.strftime("%d-%m-%Y")
    subcountxrp = cg.get_coin_history_by_id(id='ripple', date=sd, localization='false')
    subscribersxrp = subcountxrp['developer_data']['subscribers']
    sub.append(subscribersxrp)
print (sub)

[505, None, 505, 505, 505, 505, None, 505, None, 505, None, 505, 505, 505, None, None, 505, None, 505, None, 505, 505, 505, 505, 504, None, 504, 504, None, 504, None, None, 505, None, 506, None, 506, None, 506, None, 506, None, 505, 505, None]
```

- How to make the chart more user friendly
- Twitter Followers – Sentiment Analysis (non-financial data)

POSTMORTEM

- Market Cap & Volume Analysis – more in depth instead of visualizations



QUESTIONS?

