# W207 Exploratory Visualization - 16th June 2021 Summer

Team Members : Jeffrey Adams, Pow Chung, Sneha Bhattacharya, Matt White

## Introduction

Our research is focused on the cryptocurrency market since 2016 to the present. Although Bitcoin, the leading cryptocurrency, has been in circulation since 2009, in the last year there has been a paradigm shift where companies, governments, financial institutions, and individuals are beginning to take it seriously as a viable alternative to traditional currency and as a means of storing value. Cryptocurrency markets are still very immature and are subject to extreme volatility and speculation. Our research seeks to understand what factors drive the market, whether the behavior of the various cryptocurrencies is similar to that of traditional trading markets, and how ETFs and commodities, like Cryptocurrency, will fare going forward. Hopefully, our research will provide insight into the behavior of cryptocurrencies so that individuals and institutions can make informed trading decisions, across ETFs and commodities. Our Cryptocurrency

## Hypothesis 2: Alt coins are highly correlated with and follow Bitcoin price

The view shows that Alt coins are on par with Bitcoin in terms of price percentage performance, there is a great opportunity to invest in alt coins.

```
[In]: alt.Chart(df[df.Symbol.isin(['BTC-USD', 'ETH-USD'])].mark_line().encode(
          y = alt.Y('Value:Q',
          scale = alt.Scale(type='log'),
          axis = alt.Axis(grid=False),
          title = 'Crypto Price (USD)'),
          color = 'Symbol:N',
          x = alt.X('Date:T', title = 'Date')
      ).properties(width = 700, title='BTC-USD Price VS ETH-USD')
```

The view shows that BTC and Alt Coin are very much aligned in the daily returns, this is an insight that alt coin follows the momentum of BTC as they fluctuate with the highly similar pattern when we overlay their trend lines.

```
[In]: alt.Chart(df[df.Symbol.isin(['BTC-USD', 'ETH-USD'])].mark_line().encode(
          y = alt.Q('Value:Q',
          axis = alt.Axis(grid=False),
          title = 'Percent(%)'),
          color = 'Symbol:N',
          x = alt.X('Date:T', title='BTC and ETH Returns by Date Overlaid')
      ).properties(width = 700, title='BTC and ETH Returns by Date Overlaid')
```

```
[In]: alt.Chart(df[df.Symbol.isin(['BTC-USD', 'ETH-USD', 'LTC-USD', 'ADA-USD', 'XRP-USD', 'DOGE-USD', 'BCH-USD', 'ETC-USD', 'MATIC-USD', 'XLM-USD', 'ALGO-USD', 'VET-USD', 'TRX-USD', 'EOS-USD', 'XMR-USD', 'XTZ-USD', 'BNB-USD', 'USDT-USD', 'USDC-USD', 'BUSD-USD', 'DAI-USD', 'HEX-USD', 'CRO-USD', 'LEO-USD', 'OKB-USD', 'UNI-USD', 'SOL-USD', 'AVAX-USD', 'LUNA-USD', 'ICP-USD', 'FTT-USD', 'HBAR-USD', 'MANA-USD', 'SHIB-USD', 'CRV-USD', 'KLAY-USD', 'XEC-USD', 'ZEC-USD', 'DASH-USD', 'NEAR-USD', 'FTM-USD', 'ZIL-USD', 'ENJ-USD', 'GRT-USD', 'BAT-USD', 'COMP-USD', 'YFI-USD', 'SNX-USD', 'MKR-USD', 'AAVE-USD', 'SUSHI-USD', 'UMA-USD', 'REN-USD', 'KSM-USD', 'ZRX-USD', 'OMG-USD', 'LRC-USD', 'QTUM-USD', 'BTM-USD', 'ICX-USD', 'ONT-USD', 'SC-USD', 'BTG-USD', 'NEO-USD', 'XVG-USD', 'RVN-USD', 'DCR-USD', 'WAVES-USD', 'ZEN-USD', 'SXP-USD', 'BAL-USD', 'BNT-USD', 'CELR-USD', 'CHZ-USD', 'CKB-USD', 'COTI-USD', 'DGB-USD', 'FET-USD', 'FLOW-USD', 'IOST-USD', 'IOTA-USD', 'KAVA-USD', 'KNC-USD', 'MIR-USD', 'NMR-USD', 'OCEAN-USD', 'ONE-USD', 'PERP-USD', 'QNT-USD', 'RAY-USD', 'REEF-USD', 'RLC-USD', 'SKL-USD', 'STORJ-USD', 'TOMO-USD', 'UMEE-USD', 'WAXP-USD', 'WRX-USD', 'XEM-USD', 'XNO-USD', 'XYO-USD', 'YGG-USD', 'ZIL-USD', 'ZRX-USD'])].pivot_table(index='Symbol', columns='Date', values='Value').
```

| | JPY | BTC-USD | ETH-USD | AAPL | TSLA | MSFT | MSGS | EQ | PYPL | MSTR | *USD | *USJPY | DEUR | DJUP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 82.000002 | 220.782903 | 2.777135 | 27.584469 | 44.497004 | 46.400001 | 20.204999 | 13.070000 | 36.700999 | 162.399994 | 47.595.000000 | 1765.537012 | 2008.699902 | 1183.000024 | 13.179000 |
| 1 | 82.400002 | 221.353903 | 2.753000 | 27.555000 | 43.863000 | 46.700999 | 20.120999 | 12.820000 | 36.600999 | 161.199994 | 47.595.000000 | 1765.537012 | 2008.699902 | 1183.000024 | 13.179000 |
| 2 | 83.020999 | 219.222012 | 2.757997 | 28.562500 | 43.219002 | 46.330002 | 19.795001 | 12.120000 | 36.700001 | 162.899994 | 47.595.000000 | 1765.537012 | 2020.879902 | 1023.000024 | 13.040001 |
| 3 | 83.000000 | 220.000000 | 2.739948 | 28.090001 | 42.230000 | 45.830002 | 19.350001 | 12.100000 | 36.700001 | 162.899994 | 47.595.000000 | 1765.537012 | 2020.879902 | 1023.000024 | 13.040001 |
| 4 | 83.270002 | 284.165007 | 2.967060 | 28.670000 | 42.189999 | 46.020000 | 19.139999 | 11.800000 | 36.439999 | 158.000000 | 48.650.470215 | 1768.619531 | 2025.800024 | 1012.099976 | 48.650002 |
| ... | | | | | | | | | | | | | | | |
| 1395 | 36.760004 | 0.676.750000 | 272.235.000 | 36.527499 | 330.167999 | 232.880000 | 435.050998 | 214.700991 | 247.250000 | 116.339996 | 106.503.984375 | 2977.040039 | 3224.139990 | 1960.000000 | 40.270000 |
| 1396 | 37.229002 | 0.680.239993 | 252.927000 | 36.397499 | 329.003998 | 233.820000 | 437.600998 | 215.000991 | 250.250000 | 117.019996 | 107.051.984375 | 2989.040039 | 3237.639990 | 2001.799992 | 40.000000 |
| 1397 | 38.410004 | 0.686.390137 | 241.541.000 | 37.050000 | 333.027997 | 236.750000 | 443.140998 | 217.099991 | 248.410999 | 121.040000 | 108.063.953125 | 2984.620000 | 3257.050049 | 2012.099976 | 41.700001 |
| 1398 | 38.600002 | 0.692.828993 | 237.019.000 | 37.272500 | 335.600000 | 239.460000 | 451.750998 | 217.600991 | 252.520000 | 121.099996 | 107.708.953125 | 2799.620004 | 3279.020000 | 2031.000000 | 40.000000 |
| 1399 | 38.080003 | 0.704.379200 | 248.874.000 | 37.480000 | 337.607998 | 241.020000 | 455.160998 | 218.000991 | 247.750000 | 119.760000 | 108.451.953125 | 2800.020000 | 3235.020000 | 2012.099976 | 41.000001 |

1400 rows × 15 columns

```
[In]: df = pd.concat([prices, metals], axis=1).pivot_table()
```

```
[In]: var_data = ndata.melt().stack().reset_index(level=0).reset_index(drop=False).rename(columns={0: 'variable', 'level_0': 'variable', 'level_1': 'variable2'})

[In]: var_data['correlation_label'] = var_data['correlation'].map('{:.2f}'.format)

[In]: base = alt.Chart(var_data).encode(
          y= alt.X('variable2:N', title = 'Tickers'),
          x= alt.X('variable:N', title = 'Tickers'
      ).
      transform_filter(
          alt.datum.correlation < 0.5,
          alt.datum.'Value' 1,
          alt.datum.'Value' -1
      ).
      ).

# The correlation heatmap layer
[In]: var_plot = base.mark_rect().encode(
          color='correlation:Q'
      ).properties(height = 470, width = 810, title = 'NFC Heatmap')

[In]: var_plot + base # overlay the text and plot
```

## Hypothesis 3: Cryptocurrency market is highly influenced by non-market factors that don't affect traditional markets (ie. tweets, corporate investment, covid?)

Cryptocurrency market is highly influenced by non-market factors that don't affect traditional markets (ie. tweets, corporate investment, individual opinions, google searches). What we are trying to discover here is whether or not the crypto market, being different regulation, is susceptible to factors that don't affect conventional markets like NASDAQ and NYSE. Tweets and comments made by Elon Musk have affected the overall crypto market, as well as the price of Bitcoin and the price and demand for Doge. Here we want to see just what factors have the biggest impact on movements in the crypto market, whether positive or negative.

```
[In]: plot_range = 365

[In]: alt.Chart(df[df.Symbol == 'BTC-USD'].tail(365).mark_bar().encode(
          y = alt.Q('Value:Q',
          y = alt.Q('Returns:Q',
          axis = alt.Axis(grid=False),
          color = alt.condition(
              alt.datum.Value > 0,
              alt.value('steelblue'),
              alt.value('orange'
          ).properties(width = 700, title='BTC Daily Returns Trend')

[In]: historic_news = pd.DataFrame([
          {
              'start': '2021-01-15',
              'end': '2021-01-21',
              'news': 'Jim Bidder'
          },
          {
              'start': '2020-11-22',
              'end': '2021-01-21',
              'news': 'Elon Musk'
          },
          {
              'start': '2020-11-04',
              'end': '2020-11-10',
              'news': 'Trump-19'
          },
      ])

[In]: events = alt.Chart(historic_news).mark_rule(
          color='lightgray',
          strokeWidth=20,
          opacity = 0.5,
      ).encode(
          x = alt.X(historic_news.date.year == 'Elon Musk'
      )

events2 = alt.Chart(historic_news).mark_rule(
          color='lightgray',
          strokeWidth=20,
          opacity = 0.5,
      ).encode(
          x = alt.X(historic_news.date.year == 'Trump-19'
      )

events3 = alt.Chart(historic_news).mark_rule(
          color='lightgray',
          strokeWidth=20,
          opacity = 0.5,
      ).encode(
          x = alt.X(historic_news.date.year == 'Jim Bidder'
      )

text = alt.Chart(historic_news).mark_text(
          align='left',
          baseline='middle',
          dx=7,
          dy=-5,
          angle=270,
          color='black'
          ).encode(
          x= alt.X('start:T',
          y= -400:Q
          text='news:N',
          color=alt.value('#000000')
      ).

[In]: events + events2 + events3 + text
```

### What could be improved about this view:

We use heatmap to prepare any specific event that might cause the spike in Bitcoin price movement.

```
[In]: alt.Chart(df[df.Symbol == 'BTC-USD'].mark_rect().encode(
          y = alt.O('monthdate(d):T',
          x = alt.O('month(date):T',
          color = alt.Q('Returns:Q',
      ).properties(title = 'ATR Heatmap by Day and Month')

[Out]:
```

```
[In]: alt.Chart(df[df.Symbol == 'BTC-USD'].mark_rect().encode(
          y = alt.O('monthdate(d):T',
          x = alt.O('month(date):T',
          color = alt.Q('Returns:Q',
      ).properties(title = 'Returns Heatmap by Hours and Day of the Week')

[In]: alt.Chart(df[df.Symbol == 'BTC-USD'].mark_rect().encode(
          y = alt.O('monthdate(d):T',
          x = alt.O('month(date):T',
          color = alt.Q('Returns:Q',
      ).properties(title = 'BTC Returns Heatmap by Day and Month')

[Out]:
```

```
[In]: alt.Chart(df[df.Symbol == 'BTC-USD'].mark_rect().encode(
          y = alt.O('monthdate(d):T',
          x = alt.O('month(date):T',
          color = alt.Q('Returns:Q',
      ).properties(title = 'Returns Heatmap by Hours and Day of the Week')

[Out]:
```

```
[In]: alt.Chart(df[df.Symbol.isin(['BTC-USD', 'LTC-USD'])].mark_circle().encode(
          y = alt.Q('Returns:Q',
          axis = alt.Axis(grid=False),
          color = 'Symbol:N',
      ).properties(width = 700, title='BTC VS Down Jone (In Terms of Return VS Volatility)')
```

### What could be improved about this view:

The shortest candlestick chart is potential to compare Bitcoin Versus Gold market as both of the commodities are highly driven by events such as government policy. As shown in the charts below, both of them are moving in the opposite direction.

```
[In]: # Candlestick plot for short term price movement analysis

      plot_range = 100

      open_close_color = alt.condition('datum.Open >= datum.Close',
                                       alt.value('#FF9854'),
                                       alt.value('#4C78A8'))

      base = alt.Chart(df[df.Symbol == 'BTC-USD'].tail(plot_range).encode(
          x = alt.X('
              timeUnit='date',
              axis=alt.Axis(
              format='%a%d',
              labelAngle=-45,
              title='Date'
          ),
          ),
          color=open_close_color
      ).properties(title = 'BTC 3-month Candlestick Chart', width = 700)

      rule = base.mark_rule().encode(
          y = alt.Y(
              scale=alt.Scale(zero=False),
              title='Price'
          ),
          y2 = alt.Y('High')
      )

      bar = base.mark_bar(tooltip=True).encode(
          y = alt.Y('Open:Q',
          y2 = alt.Y('Close:Q'
      )

      MFC = rule + bar + line1

[In]: # Candlestick plot for short term price movement analysis

      plot_range = 100

      open_close_color = alt.condition('datum.Open >= datum.Close',
                                       alt.value('#FF9854'),
                                       alt.value('#4C78A8'))

      base = alt.Chart(df[df.Symbol == 'GC=F'].tail(plot_range).encode(
          x = alt.X('
              timeUnit='date',
              axis=alt.Axis(
              format='%a%d',
              labelAngle=-45,
              title='Date'
          ),
          ),
          color=open_close_color
      ).properties(title = 'GOLD 3-month Candlestick Chart', width = 700)

      rule = base.mark_rule().encode(
          y = alt.Y(
              scale=alt.Scale(zero=False),
              title='Price'
          ),
          y2 = alt.Y('High')
      )

      bar = base.mark_bar(tooltip=True).encode(
          y = alt.Y('Open:Q',
          y2 = alt.Y('Close:Q'
      )

      MFC = rule + bar + line1

[Out]:
```

### What could be improved about this view:

Correlation matrix is created to show the hashtags of the combination among all the assets

```
[In]: corrMatrix = df.corr()

[In]: ndata = df.dropna(axis=0).groupby('Symbol').get_group('JPY').['Close'].reset_index(drop=False)
      for Sym in tickers:
          if Sym != 'JPY':
              temp = df.dropna(axis=0).groupby('Symbol', axis=0).get_group(Sym).['Close'].reset_index(drop=False).rename(columns={'Close': 'Close'})
              ndata = ndata.merge(temp, ticks=True, axis=1, joins='inner')

[In]: ndata
```

| | JPY | BTC-USD | ETH-USD | AAPL | TSLA | MSFT | MSGS | EQ | PYPL | MSTR | *USD | *USJPY | DEUR | DJUP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 82.000002 | 220.782903 | 2.777135 | 27.584469 | 44.497004 | 46.400001 | 20.204999 | 13.070000 | 36.700999 | 162.399994 | 47.595.000000 | 1765.537012 | 2008.699902 | 1183.000024 | 13.179000 |
| 1 | 82.400002 | 221.353903 | 2.753000 | 27.555000 | 43.863000 | 46.700999 | 20.120999 | 12.820000 | 36.600999 | 161.199994 | 47.595.000000 | 1765.537012 | 2008.699902 | 1183.000024 | 13.179000 |
| 2 | 83.020999 | 219.222012 | 2.757997 | 28.562500 | 43.219002 | 46.330002 | 19.795001 | 12.120000 | 36.700001 | 162.899994 | 47.595.000000 | 1765.537012 | 2020.879902 | 1023.000024 | 13.040001 |
| 3 | 83.000000 | 220.000000 | 2.739948 | 28.090001 | 42.230000 | 45.830002 | 19.350001 | 12.100000 | 36.700001 | 162.899994 | 47.595.000000 | 1765.537012 | 2020.879902 | 1023.000024 | 13.040001 |
| 4 | 83.270002 | 284.165007 | 2.967060 | 28.670000 | 42.189999 | 46.020000 | 19.139999 | 11.800000 | 36.439999 | 158.000000 | 48.650.470215 | 1768.619531 | 2025.800024 | 1012.099976 | 48.650002 |
| ... | | | | | | | | | | | | | | | |
| 1395 | 36.760004 | 0.676.750000 | 272.235.000 | 36.527499 | 330.167999 | 232.880000 | 435.050998 | 214.700991 | 247.250000 | 116.339996 | 106.503.984375 | 2977.040039 | 3224.139990 | 1960.000000 | 40.270000 |
| 1396 | 37.229002 | 0.680.239993 | 252.927000 | 36.397499 | 329.003998 | 233.820000 | 437.600998 | 215.000991 | 250.250000 | 117.019996 | 107.051.984375 | 2989.040039 | 3237.639990 | 2001.799992 | 40.000000 |
| 1397 | 38.410004 | 0.686.390137 | 241.541.000 | 37.050000 | 333.027997 | 236.750000 | 443.140998 | 217.099991 | 248.410999 | 121.040000 | 108.063.953125 | 2984.620000 | 3257.050049 | 2012.099976 | 41.700001 |
| 1398 | 38.600002 | 0.692.828993 | 237.019.000 | 37.272500 | 335.600000 | 239.460000 | 451.750998 | 217.600991 | 252.520000 | 121.099996 | 107.708.953125 | 2799.620004 | 3279.020000 | 2031.000000 | 40.000000 |
| 1399 | 38.080003 | 0.704.379200 | 248.874.000 | 37.480000 | 337.607998 | 241.020000 | 455.160998 | 218.000991 | 247.750000 | 119.760000 | 108.451.953125 | 2800.020000 | 3235.020000 | 2012.099976 | 41.000001 |

1400 rows × 15 columns

```
[In]: df = pd.concat([prices, metals], axis=1).pivot_table()

[In]: var_data = ndata.melt().stack().reset_index(level=0).reset_index(drop=False).rename(columns={0: 'variable', 'level_0': 'variable', 'level_1': 'variable2'})

[In]: var_data['correlation_label'] = var_data['correlation'].map('{:.2f}'.format)

[In]: base = alt.Chart(var_data).encode(
          y= alt.X('variable2:N', title = 'Tickers'),
          x= alt.X('variable:N', title = 'Tickers'
      ).
      transform_filter(
          alt.datum.correlation < 0.5,
          alt.datum.'Value' 1,
          alt.datum.'Value' -1
      ).
      ).

# The correlation heatmap layer
[In]: var_plot = base.mark_rect().encode(
          color='correlation:Q'
      ).properties(height = 470, width = 810, title = 'NFC Heatmap')

[In]: var_plot + base # overlay the text and plot
```




### Conclusion:

As can be seen from the charts above looking at Apple and S&P 500 to BTC. The returns on a daily basis do mirror each other. When Apple has negative returns often BTC as well. It would be helpful to view this plot with other tickers as well. What also needs to be determined is the magnitude of these returns over a long period of time.

### Challenges:

The data consists of multiple assets spanning across 5 years of data which created a huge volume of data entries to plot a chart. The memory crunch is a serious issue, so we have modified the dataset to slice the data for each of the chart and tried many methods to avoid generating huge memory size of notebook.

## End of Presentation