Composite view to see result of Sentiment data on Bitcoin price

In this file we are performing composite view with sentiment analysis on the ElonMusk tweet data and Bitcoin price against year 2021. The data's are already prepared in sepearte notebooks named titter.ipynb and bitcoin.ipynb

In [1]:
```
# Initial imports
import matplotlib.pyplot as plt
%matplotlib inline
import panel as pn
pn.extension('plotly')
import plotly.express as px
import pandas as pd
import hvplot.pandas

import os
from pathlib import Path
from dotenv import load_dotenv

from panel.interact import interact

from bokeh.models.renderers import GlyphRenderer
from bokeh.models import Range1d, LinearAxis
```

In [2]:
```
### Import the data for Plotting
```

In [3]:
```
### sentitment data
tweet_path = "Resources/Sentiment_Tweets.csv"
master_tweet_df = pd.read_csv( tweet_path, index_col="date", infer_datetime_format=True, parse_
master_tweet_df.head()
```

Out[3]:

date	name	tweet	replies_count	retweets_count	likes_count	Classification	Confidence	period	se
2021-02-04	Elon Musk	Dogecoin is the peopleâ s crypto	18465	97994	533684	Positive	0.474	February	
2021-02-07	Elon Musk	@itsALLrisky Itâ s the most fun crypto!	807	3537	22013	Positive	0.418	February	
2021-02-09	Elon Musk	@CryptoShrikar @CoinDesk @Tesla @Dan_Z_Palmer ...	240	158	3167	Positive	0.594	February	
2021-02-10	Elon Musk	@freewalletorg Any crypto wallet that wonâ t ...	2012	4449	28205	Positive	0.508	February	

date	name	tweet	replies_count	retweets_count	likes_count	Classification	Confidence	period	se
2021-02-19	Elon Musk	@business Teslaâ s action is not directly ref...	922	3228	26473	Positive	0.623	February	

In [4]:
```
tweet_df = master_tweet_df[['Classification','sentiment_numeric','negative','positive','neutral
tweet_df.head()
```

Out[4]:

date	Classification	sentiment_numeric	negative	positive	neutral
2021-02-04	Positive	1	0.0	1.0	0.0
2021-02-07	Positive	1	0.0	1.0	0.0
2021-02-09	Positive	1	0.0	1.0	0.0
2021-02-10	Positive	1	0.0	1.0	0.0
2021-02-19	Positive	1	0.0	1.0	0.0

In [5]:
```
### Bitcoin data
bitcoin_path = "Resources/combined_bitcoin_data.csv"
bitcoin_df = pd.read_csv( bitcoin_path, index_col="Date", infer_datetime_format=True, parse_dat
bitcoin_df = bitcoin_df.reset_index()
bitcoin_df = bitcoin_df.rename(columns = {'Date':'date'}).set_index('date')

bitcoin_df
```

Out[5]:

date	Close	Daily Returns
2021-01-02	32127.27	0.09373
2021-01-03	32782.02	0.02038
2021-01-04	31971.91	-0.02471
2021-01-05	33992.43	0.06320
2021-01-06	36824.36	0.08331
...	...	...
2021-12-27	50640.42	-0.00333
2021-12-28	47588.85	-0.06026
2021-12-29	46444.71	-0.02404
2021-12-30	47178.13	0.01579
2021-12-31	46306.45	-0.01848

364 rows × 2 columns

In [6]:
```
# Joining Bitcoin and Sentiment by date index

bitcoin_df = bitcoin_df.join(tweet_df)
bitcoin_df
```

Out[6]:

date	Close	Daily Returns	Classification	sentiment_numeric	negative	positive	neutral
2021-01-02	32127.27	0.09373	NaN	NaN	NaN	NaN	NaN
2021-01-03	32782.02	0.02038	NaN	NaN	NaN	NaN	NaN
2021-01-04	31971.91	-0.02471	NaN	NaN	NaN	NaN	NaN
2021-01-05	33992.43	0.06320	NaN	NaN	NaN	NaN	NaN
2021-01-06	36824.36	0.08331	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...
2021-12-27	50640.42	-0.00333	NaN	NaN	NaN	NaN	NaN
2021-12-28	47588.85	-0.06026	NaN	NaN	NaN	NaN	NaN
2021-12-29	46444.71	-0.02404	NaN	NaN	NaN	NaN	NaN
2021-12-30	47178.13	0.01579	NaN	NaN	NaN	NaN	NaN
2021-12-31	46306.45	-0.01848	NaN	NaN	NaN	NaN	NaN

367 rows × 7 columns

Cleaning the data for null values with "0" and "Neutral"

In [7]:

```
bitcoin_df['Classification'] = bitcoin_df['Classification'].fillna('Neutral')
bitcoin_df['sentiment_numeric'] = bitcoin_df['sentiment_numeric'].fillna(0)
bitcoin_df['negative'] = bitcoin_df['negative'].fillna(0)
bitcoin_df['positive'] = bitcoin_df['positive'].fillna(0)
bitcoin_df['neutral'] = bitcoin_df['neutral'].fillna(0)

bitcoin_df
```

Out[7]:

date	Close	Daily Returns	Classification	sentiment_numeric	negative	positive	neutral
2021-01-02	32127.27	0.09373	Neutral	0.0	0.0	0.0	0.0
2021-01-03	32782.02	0.02038	Neutral	0.0	0.0	0.0	0.0
2021-01-04	31971.91	-0.02471	Neutral	0.0	0.0	0.0	0.0
2021-01-05	33992.43	0.06320	Neutral	0.0	0.0	0.0	0.0
2021-01-06	36824.36	0.08331	Neutral	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...
2021-12-27	50640.42	-0.00333	Neutral	0.0	0.0	0.0	0.0
2021-12-28	47588.85	-0.06026	Neutral	0.0	0.0	0.0	0.0
2021-12-29	46444.71	-0.02404	Neutral	0.0	0.0	0.0	0.0
2021-12-30	47178.13	0.01579	Neutral	0.0	0.0	0.0	0.0
2021-12-31	46306.45	-0.01848	Neutral	0.0	0.0	0.0	0.0

367 rows × 7 columns

In [8]:

```
bitcoin_df.isnull().sum()
```

Out[8]:

```
Close                0
```

```
Daily Returns        0
Classification       0
sentiment_numeric    0
negative             0
positive             0
neutral              0
dtype: int64
```

Aggregate the sentiments by period:month

In [9]:

```
bitcoin_df = bitcoin_df.reset_index()
bitcoin_df['period'] = bitcoin_df['date'].dt.month_name(locale = 'English')
bitcoin_df = bitcoin_df.set_index('date')

# aggregate the sentiments by period:month

# this single line replaced my old implemenation using groupby by the below 4 liners.
# Thanks to our Tutor, Swaraj for explaining agg{} in resample
groupby_period= bitcoin_df[["Close","negative","positive","neutral"]].resample('w').agg({'Close
                                                                                          'negat
                                                                                          'posit
                                                                                          'neutr


#groupby_period = bitcoin_df[["Close", "Daily Returns","negative","positive","neutral"]].groupb

# Sort by month index
#sort_order = ['January', 'February', 'March', 'April', 'May', 'June', 'July','August','Septemb

#groupby_period.index = pd.CategoricalIndex(groupby_period['period'], categories=sort_order, or
#groupby_period =groupby_period .sort_index()
groupby_period.head()
```

Out[9]:

date	Close	negative	positive	neutral
2021-01-03	32454.645000	0.0	0.0	0.0
2021-01-10	37366.905714	0.0	0.0	0.0
2021-01-17	36398.300000	0.0	0.0	0.0
2021-01-24	33776.588571	0.0	0.0	0.0
2021-01-31	32933.594286	0.0	0.0	0.0

Helper functions

In [10]:

```
def create_line_chart(data, title, xlabel, ylabel, size):
    """
    Create a line chart based in the data argument.
    """
    fig = plt.figure(constrained_layout=True, figsize=(6,5))
    linechart = data.plot.line(figsize = size, title=title, legend=True )
    linechart.set_xlabel(xlabel)
    linechart.set_ylabel(ylabel)

    return fig

# Resuable function for creating bar chart
def create_bar_chart(data, title, xlabel, ylabel,  size):

    """
    Create a barplot based in the data argument.
    """

    fig = plt.figure(constrained_layout=True, figsize=(6,6))
```

```
    barchart = data.plot.bar(figsize=size, title=title, x=xlabel )
    barchart.set_xlabel(xlabel)
    barchart.set_ylabel(ylabel)

    return fig


def overall_crypto_sentiment(data):
    fig = px.sunburst(data, path=[ 'Classification', 'date'], title="Overall Crypto Sentiment -
    return fig

def px_bar(data, title, xlabel, ylabel,  size):
    fig =  px.bar(
        data,
        x=xlabel,

        title=title
        )
    return fig

# Use the secondary y axis for sentiment data and primary y axis for bit coin data.
# Refered the solution to achieve the twiny plot https://github.com/holoviz/holoviews/issues/39
def apply_positive_formatter(plot, element):
    p = plot.state

    # create secondary range and axis
    p.extra_y_ranges = {"twiny": Range1d(start=0, end=6)}
    p.add_layout(LinearAxis(y_range_name="twiny"), 'left')

    # set glyph y_range_name to the one we've just created
    glyph = p.select(dict(type=GlyphRenderer))[0]
    glyph.y_range_name = 'twiny'

bit_plot = groupby_period['Close'].hvplot.line(yaxis="right" ).opts(
    yformatter="%.0f"
)
```

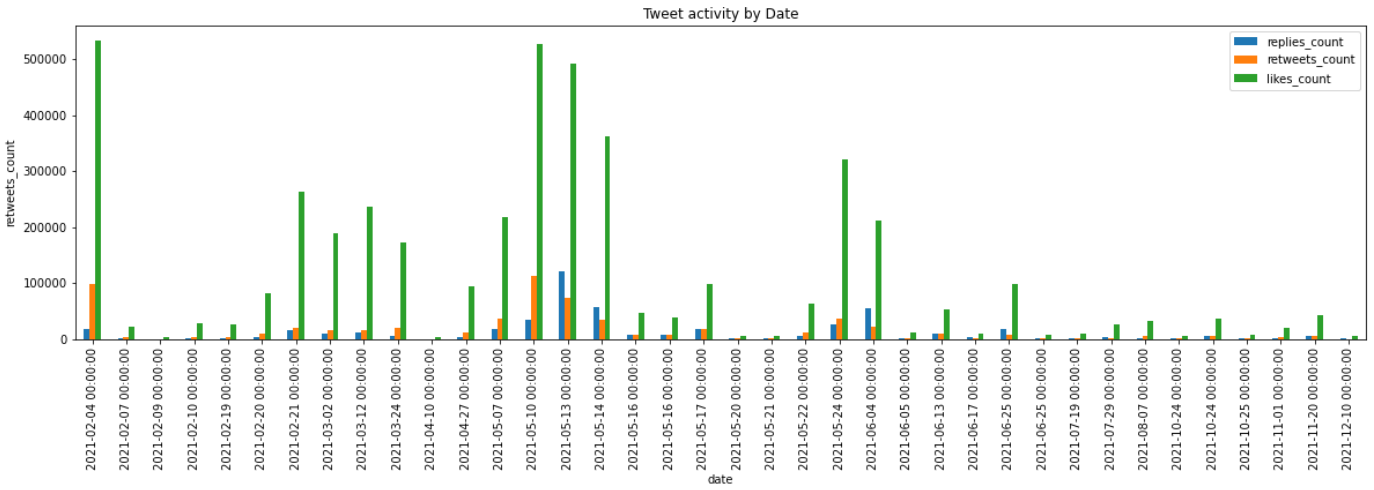
In [11]:

```
# plot the tweet activities
master_tweet_df = master_tweet_df.reset_index()

tweet_activity_date = create_bar_chart(master_tweet_df[["date","replies_count","retweets_count"
tweet_activity_date
```

Out[11]:

```
<Figure size 432x432 with 0 Axes>
```


```
<Figure size 432x432 with 0 Axes>
```

In [12]:

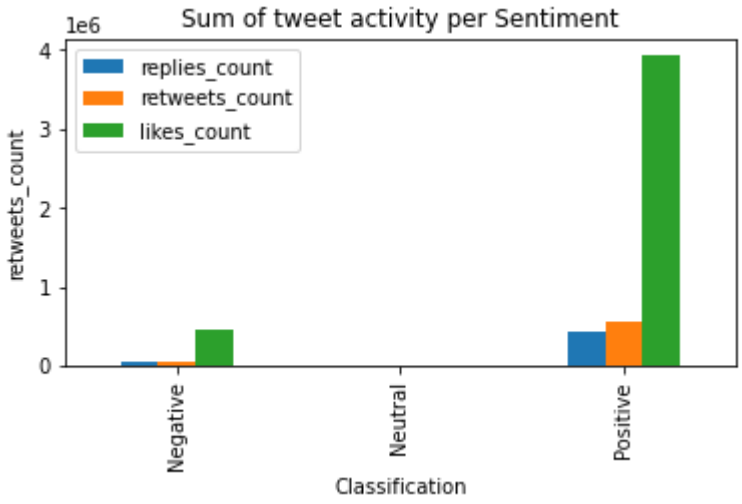
```
# groupby the tweet columns by classification

groupby_classification = master_tweet_df[["replies_count","retweets_count","likes_count"]].grou
```

```
sum_tweet_activity = create_bar_chart(groupby_classification, "Sum of tweet activity per Sentim

sum_tweet_activity
```

Out[12]:

```
<Figure size 432x432 with 0 Axes>
```

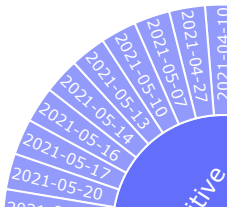
```
<Figure size 432x432 with 0 Axes>
```



In [13]:

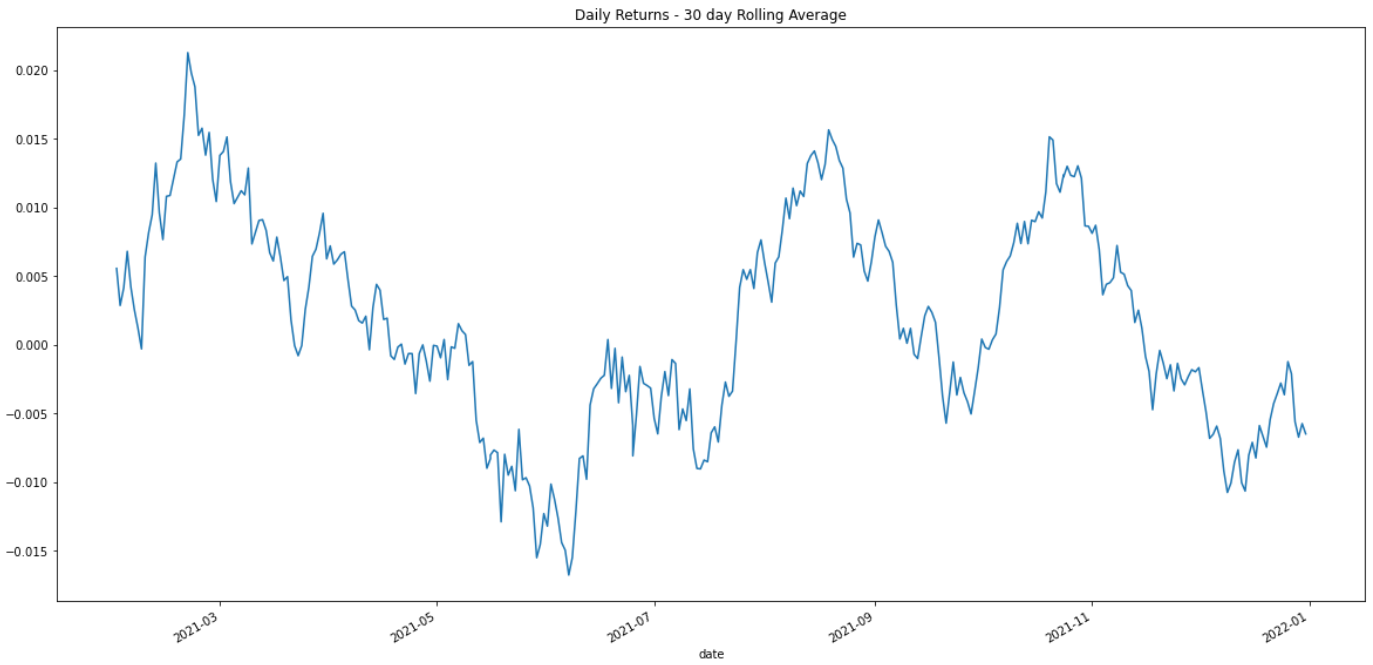
```
# Sentiment view

df = tweet_df.reset_index()
overall_c_sentiment = overall_crypto_sentiment(df)
overall_c_sentiment
```

Overall Crypto Sentiment - 2021



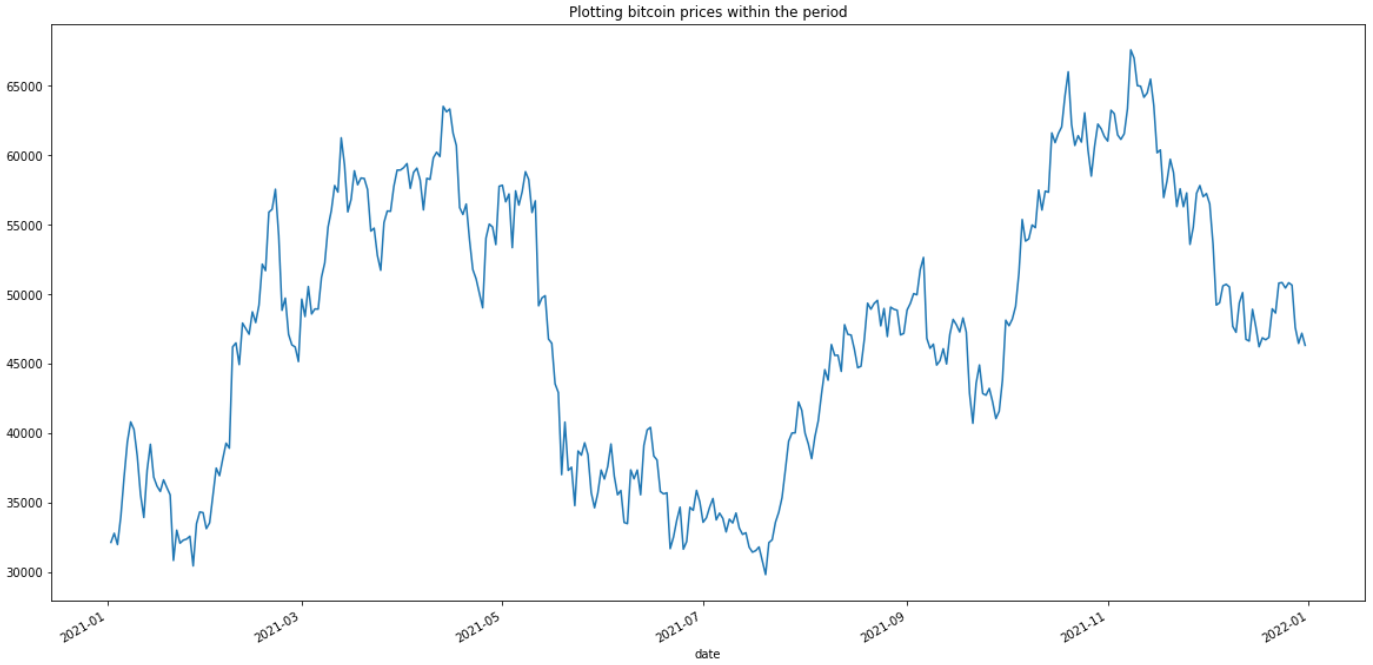
In [14]:

```
MA_btc_daily_ret = bitcoin_df['Daily Returns'].rolling(window = 30).mean()

daily_returns = MA_btc_daily_ret.plot(figsize = (20,10), title = 'Daily Returns - 30 day Rollir
```



In [15]:

```
close_plot =bitcoin_df['Close'].plot(figsize = (20,10), title="Plotting bitcoin prices within t
```



Composite Plotting of sentitment with bitcoin

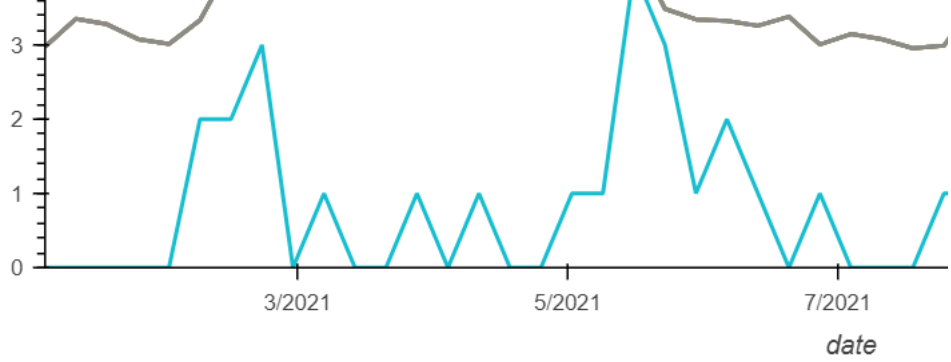
In [16]:

```
positive_plot = groupby_period[["positive"]].hvplot.line(yaxis="left").opts(hooks=[apply_positi

negative_plot = groupby_period[["negative"]].hvplot.line(yaxis="left").opts(hooks=[apply_positi
```

In [24]:

```
positive_plot = (bit_plot * positive_plot ).opts(
    title="ElonMusk Postive tweet and Bitcoin Prices aggregated over period in 2021", width=900
)

positive_plot
```

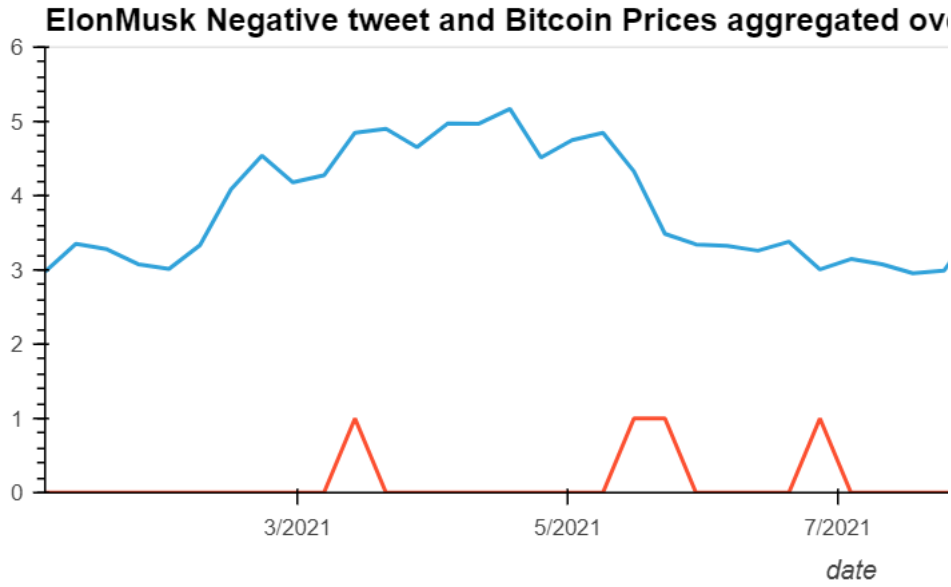
Out[24]:



In [18]:

```
negative_plot = (bit_plot * negative_plot).opts(
    title="ElonMusk Negative tweet and Bitcoin Prices aggregated over period in 2021", width=90
)

negative_plot
```

Out[18]:



Dashboard panel creation for the above visualisation

In [26]:

```
# Create a Title for the Dashboard
title = "## Elon Musk tweets sentiment effects on Bitcoin Price"

# Define a welcome text
tweet_text = "#### The visualization of historical tweets and Bitcoin prices of year 2021"


overall_sentiment  = pn.Row(tweet_text,overall_c_sentiment, sum_tweet_activity)
tweet_activity = pn.Row(sum_tweet_activity)
tweet_act_date = pn.Row(tweet_activity_date)

welcome_column = pn.Column( overall_sentiment)
sentiment_column = pn.Column(tweet_activity ,tweet_act_date)
bitcoin_column = pn.Column(pn.Row(daily_returns), pn.Row(close_plot))
composite_column = pn.Column(pn.Row(positive_plot), pn.Row(negative_plot))

# Create the main dashboard
to_sentiment_dashboard_tabs = pn.Tabs(
    (
        "Welcome",
        welcome_column
    ),
    (
```

```
        "Sentiment",
        sentiment_column
    ),
    (
        "Bitcoin",
        bitcoin_column
    ),
    (
        "Bitcoin & Tweet sentiment composit plot",
        composite_column
    )

)

to_sentiment_dashboard = pn.Column(title, to_sentiment_dashboard_tabs)
to_sentiment_dashboard
```

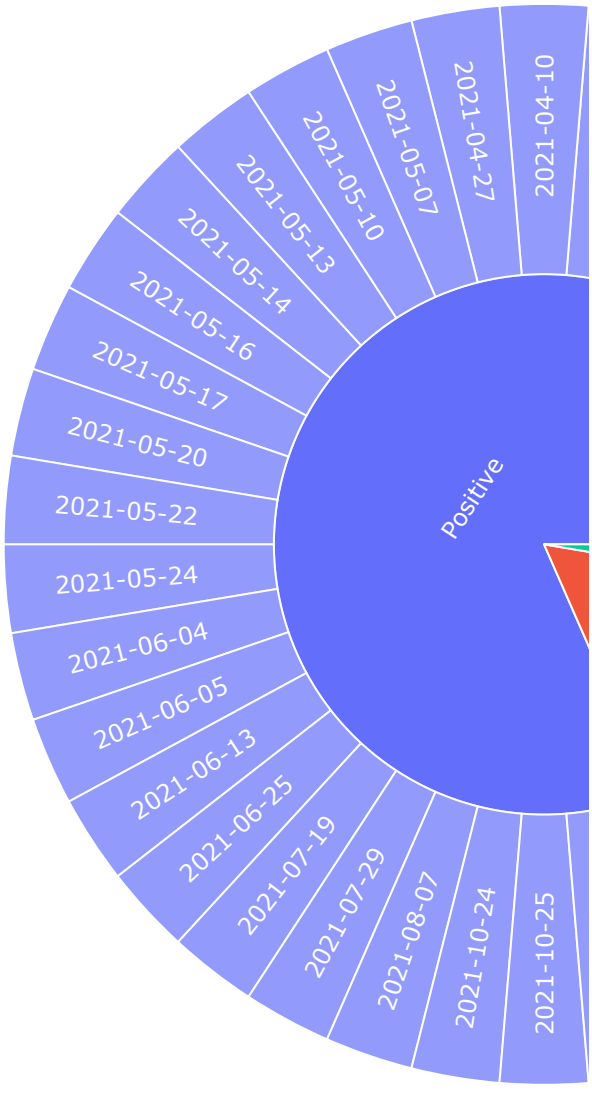
Out[26]:

Elon Musk tweets sentiment effects on Bitcoin Price

Welcome | Sentiment | Bitcoin | Bitcoin & Tweet sentiment composit plot

The visualization of historical tweets and Bitcoin prices of year 2021

Overall Crypto Sentiment - 2021



In [20]:

```
to_sentiment_dashboard.servable()
```

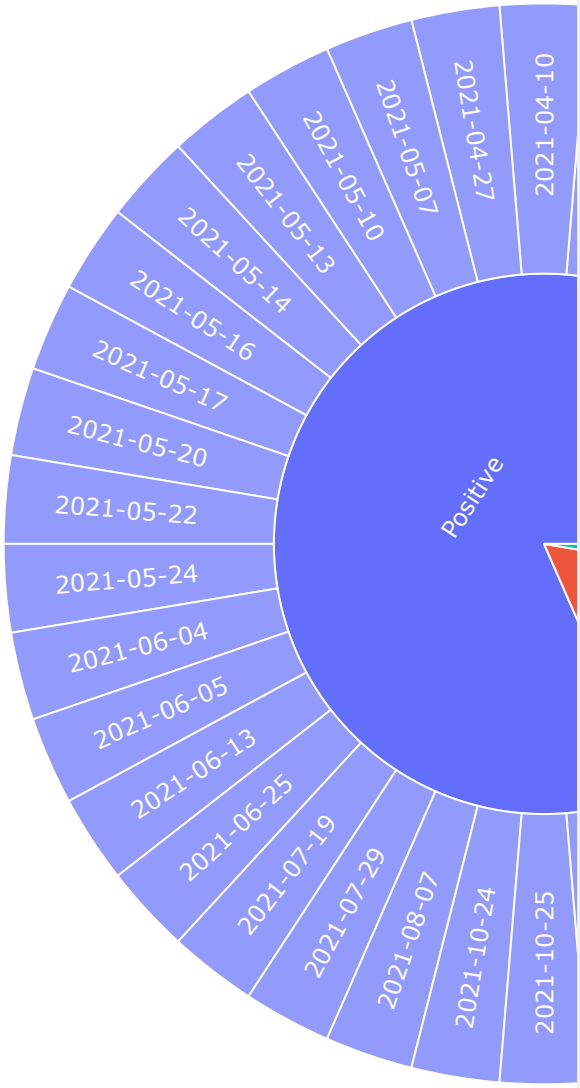
Out[20]:

Elon Musk tweets sentiment effects on Bitcoin Price

Welcome | Sentiment | Bitcoin | Bitcoin & Tweet sentiment composit plot

The visualization of historical tweets of year 2021

Overall Crypto Sentiment - 2021



In [ ]: