A Crypto Arbitrage Opportunity: analysis

Use case: Traders use arbitrage to exploit price differences between markets. If there are significant price differences, a trader could buy at a lower price from one exchange and sell at a higher price to another exchange. In this analysis, I will harvest data using Quandl API and load it in Jupyter notebook for further analysis with Python.

Objective: assess if there is a price difference between exchanges and which exchanges offer the best arbitrage opportunity.

Limitations: In this demonstration, I will focus only on the price of bitcoin (BTC) from 3 different exchanges: Kraken, Bitstamp, Bitflyer for the year 2020.

1. Import required dependencies

In [1]:
```
%pip install quandl
import os
import numpy as np
import pandas as pd

import quandl
from datetime import datetime

import matplotlib.pyplot as plt

%pip install plotly==4.14.3
import plotly.offline as py
import plotly.graph_objs as go
import plotly.express as px
import plotly.figure_factory as ff
py.init_notebook_mode(connected=True)
```

```
Requirement already satisfied: quandl in /opt/anaconda3/lib/python3.8/site-packages (3.6.1)
Requirement already satisfied: inflection>=0.3.1 in /opt/anaconda3/lib/python3.8/site-packages
(from quandl) (0.5.1)
Requirement already satisfied: pandas>=0.14 in /opt/anaconda3/lib/python3.8/site-packages (from
quandl) (1.0.5)
Requirement already satisfied: numpy>=1.8 in /opt/anaconda3/lib/python3.8/site-packages (from q
uandl) (1.19.5)
Requirement already satisfied: requests>=2.7.0 in /opt/anaconda3/lib/python3.8/site-packages (f
rom quandl) (2.24.0)
Requirement already satisfied: more-itertools in /opt/anaconda3/lib/python3.8/site-packages (fr
om quandl) (8.4.0)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.8/site-packages (from quandl)
(1.15.0)
Requirement already satisfied: python-dateutil in /opt/anaconda3/lib/python3.8/site-packages (f
rom quandl) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /opt/anaconda3/lib/python3.8/site-packages (from
pandas>=0.14->quandl) (2020.1)
Requirement already satisfied: chardet<4,>=3.0.2 in /opt/anaconda3/lib/python3.8/site-packages
(from requests>=2.7.0->quandl) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /opt/anaconda3/lib/py
thon3.8/site-packages (from requests>=2.7.0->quandl) (1.25.9)
Requirement already satisfied: certifi>=2017.4.17 in /opt/anaconda3/lib/python3.8/site-packages
(from requests>=2.7.0->quandl) (2020.6.20)
Requirement already satisfied: idna<3,>=2.5 in /opt/anaconda3/lib/python3.8/site-packages (from
requests>=2.7.0->quandl) (2.10)
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: plotly==4.14.3 in /opt/anaconda3/lib/python3.8/site-packages (4.
14.3)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.8/site-packages (from plotly==
4.14.3) (1.15.0)
Requirement already satisfied: retrying>=1.3.3 in /opt/anaconda3/lib/python3.8/site-packages (f
rom plotly==4.14.3) (1.3.3)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
quandl.ApiConfig.api_key ='UCNJ8ur3X5XcEVeyiDXo'
```

1. Harvest bitcoin data. Set exchange preferance and date range.
   - Get Bitcoin pricing data using [Quandl's free Bitcoin API (https://blog.quandl.com/api-for-bitcoin-data)](https://blog.quandl.com/api-for-bitcoin-data).

In [3]:
```
btc_usd_price_kraken= quandl.get("BCHARTS/KRAKENUSD", start_date="2020-01-01", end_date="2020-12-31")
btc_usd_price_kraken.head()
```

Out[3]:

Date	Open	High	Low	Close	Volume (BTC)	Volume (Currency)	Weighted Price
2020-01-01	7168.3	7235.0	7150.0	7174.4	1827.634894	1.315264e+07	7196.538417
2020-01-02	7174.4	7185.8	6915.0	6942.3	4057.331546	2.850185e+07	7024.778058
2020-01-03	6943.4	7397.3	6860.0	7334.8	8120.491405	5.873937e+07	7233.475126
2020-01-04	7333.2	7396.4	7260.0	7350.2	2912.373131	2.131265e+07	7317.966324
2020-01-05	7350.2	7493.3	7301.1	7346.9	2904.093494	2.153416e+07	7415.106167

1. Create an interactive visualization of bitcoin 2020 price evolution from Kraken Exchange. Hover over and select timestamp of interest. The "Weighted Price" column was used as a referance instead of closing price.

In [4]:
```
btc_trace = go.Scatter(x=btc_usd_price_kraken.index, y=btc_usd_price_kraken['Weighted Price'])
py.iplot([btc_trace])
```

1. Pull data from two additional exchanges.

In [5]:
```
exchanges = ['BITSTAMP','BITFLYER']

exchange_data = {}

exchange_data['KRAKEN'] = btc_usd_price_kraken

for exchange in exchanges:
    exchange_code = 'BCHARTS/{}USD'.format(exchange)
    btc_exchange_df = quandl.get(exchange_code, start_date="2020-01-01", end_date="2020-12-31")
    exchange_data[exchange] = btc_exchange_df
```

1. Merge all price data into one data frame on their "Weighted Price" column.

In [6]:
```
def merge_dfs_on_column(dataframes, labels, col):

    series_dict = {}
    for index in range(len(dataframes)):
        series_dict[labels[index]] = dataframes[index][col]

    return pd.DataFrame(series_dict)
```

In [7]:
```
btc_usd_datasets = merge_dfs_on_column(list(exchange_data.values()), list(exchange_data.keys()),
'Weighted Price')
```

In [8]:
```
btc_usd_datasets.head(5)
```

Out[8]:

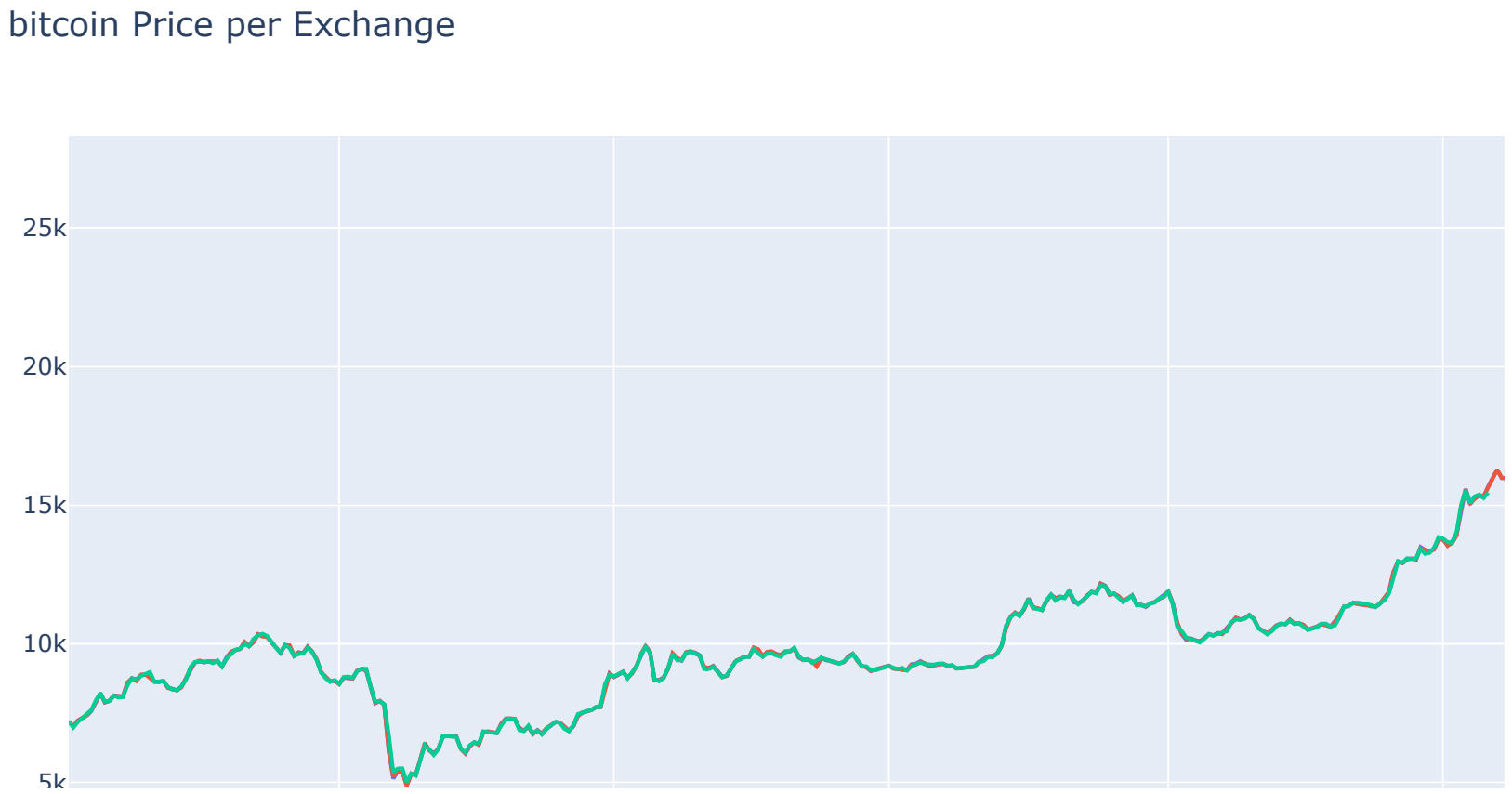
Date	KRAKEN	BITSTAMP	BITFLYER
2020-01-01	7196.538417	7195.247241	7207.406297
2020-01-02	7024.778058	7030.211788	6982.137692
2020-01-03	7233.475126	7230.186000	7185.435769
2020-01-04	7317.966324	7315.738130	7329.849319
2020-01-05	7415.106167	7427.569042	7410.826062

1. Create a function that will generate a scatter plot for the entire dataframe.

In [9]:
```
def df_scatter(df, title, seperate_y_axis=False, y_axis_label='', scale='linear', initial_hide=False):

    label_arr = list(df)
    series_arr = list(map(lambda col: df[col], label_arr))

    layout = go.Layout(
        title=title,
        legend=dict(orientation="h"),
        xaxis=dict(type='date'),
        yaxis=dict(
            title=y_axis_label,
            showticklabels= not seperate_y_axis,
            type=scale
        )
    )
```

```
    y_axis_config = dict(
        overlaying='y',
        showticklabels=False,
        type=scale )

    visibility = True
    if initial_hide:
        visibility = True

    # Form Trace For Each Series
    trace_arr = []
    for index, series in enumerate(series_arr):
        trace = go.Scatter(
            x=series.index,
            y=series,
            name=label_arr[index],
            visible=visibility
        )

        # Add seperate axis for the series
        if seperate_y_axis:
            trace['yaxis'] = 'y{}'.format(index + 1)
            layout['yaxis{}'.format(index + 1)] = y_axis_config
        trace_arr.append(trace)

    fig = go.Figure(data=trace_arr, layout=layout)
    py.iplot(fig)
```

In [10]:
```
# Remove "0" values
btc_usd_datasets.replace(0, np.nan, inplace=True)
df_scatter(btc_usd_datasets, "bitcoin Price per Exchange")
```



1. As we can see the price range of bitcoin from all three exchanges is close, however if we zoom we can observe price discrepencies at a certain time stamp. The next step is to quanitfy the price differences between exchanges. Let's take Kraken and Bitstamp for example.

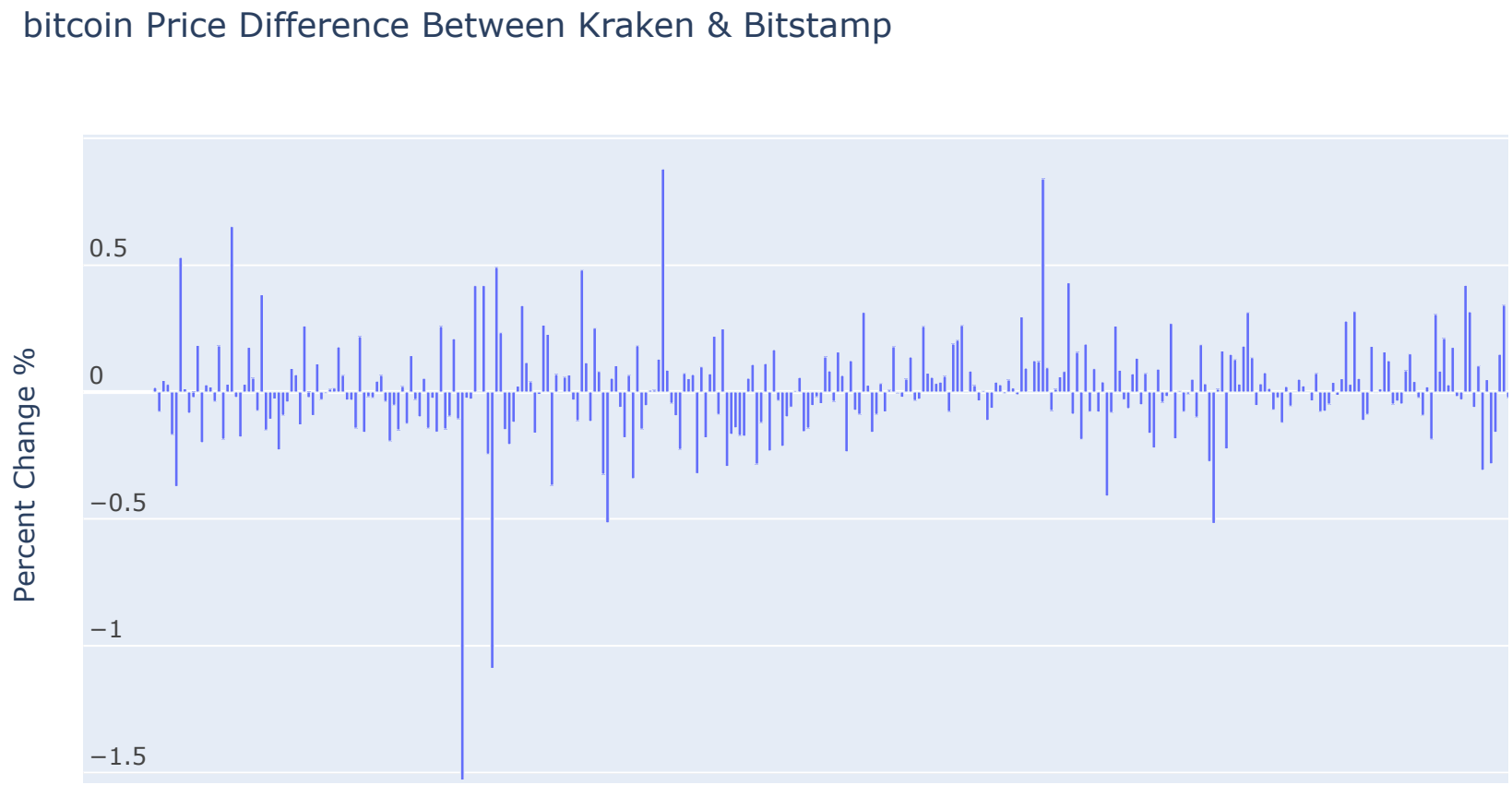
In [19]:
```
df = btc_usd_datasets[['KRAKEN','BITSTAMP']]
df['price_dif']=df.KRAKEN - df.BITSTAMP
df["percent_change"] = (df.price_dif / df.KRAKEN)*100

df.replace(0, np.nan, inplace=True)
df.head(10)
```

Out[19]:

Date	KRAKEN	BITSTAMP	price_dif	percent_change
2020-01-01	7196.538417	7195.247241	1.291177	0.017942
2020-01-02	7024.778058	7030.211788	-5.433729	-0.077351
2020-01-03	7233.475126	7230.186000	3.289126	0.045471
2020-01-04	7317.966324	7315.738130	2.228194	0.030448
2020-01-05	7415.106167	7427.569042	-12.462875	-0.168074
2020-01-06	7573.789593	7601.994206	-28.204614	-0.372398
2020-01-07	7967.765502	7925.503937	42.261564	0.530407
2020-01-08	8221.295753	8220.231471	1.064283	0.012945
2020-01-09	7886.044126	7892.620464	-6.576338	-0.083392
2020-01-10	7941.385199	7943.120177	-1.734978	-0.021847

1. Visualize and assess the percent difference in bitcoin price between Bitstamp and Kraken exchanges.

In [132]:
```
fig = px.bar(df, x=df.index, y="percent_change", title="bitcoin Price Difference Between Kraken & Bitstamp")
fig.update_yaxes(ticklabelposition="inside top", title="Percent Change %")
fig.show()
```



1. As we can see the price difference rarely passes 1%. Let's see how many times is passed 0.5% price difference.

In [21]:
```
# Frequency of dicrepency above 0.5%
df.loc[df.percent_change > 0.5, 'percent_change'].count()
```

Out[21]: 6

In [23]:
```
# Frequency of discrepncy below -0.5%
df.loc[df.percent_change < -0.5, 'percent_change'].count()
```

Out[23]: 4

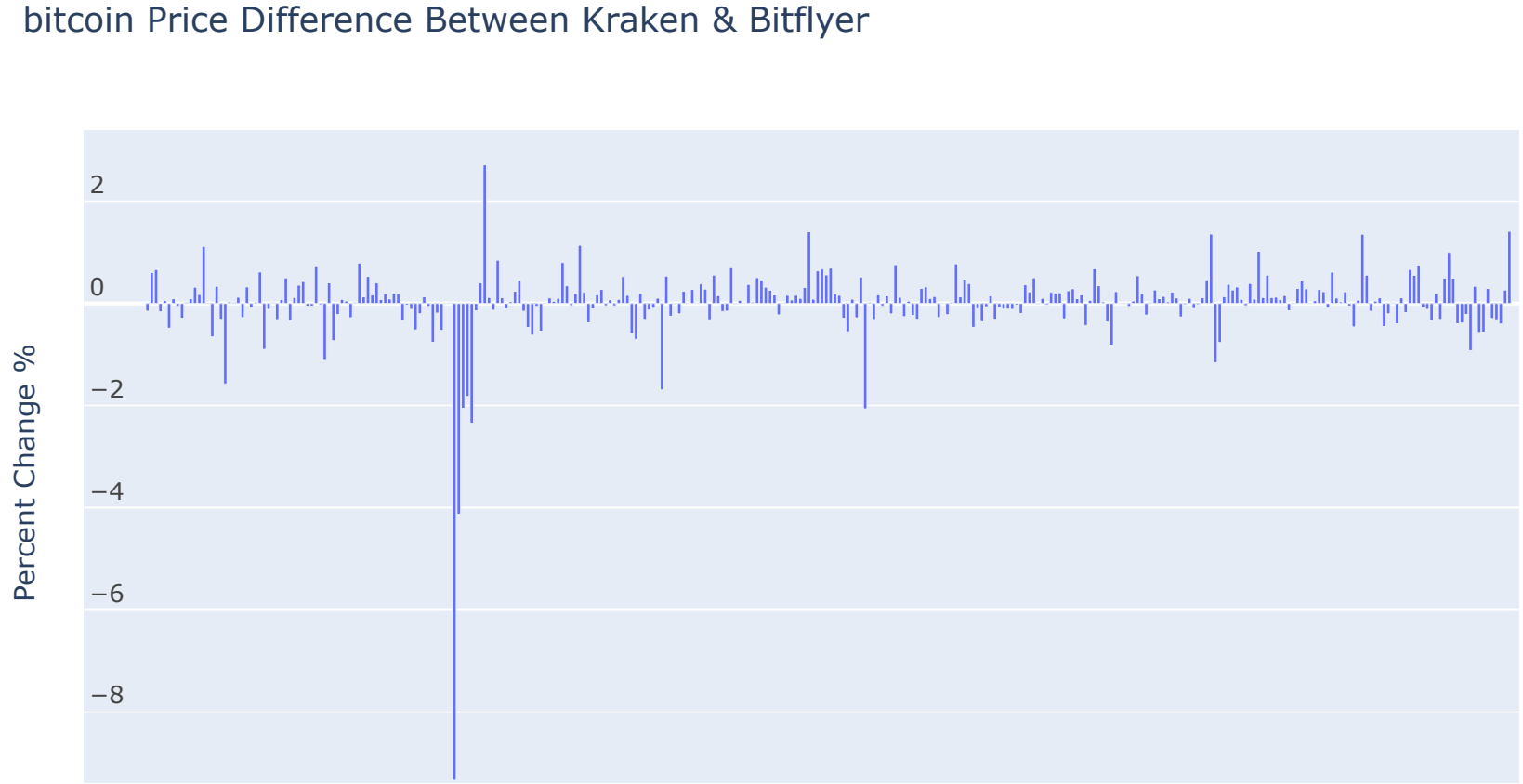
1. As we can see only 10 days in 2020 had a price difference of 0.5% between Kraken and Bitstamp. Let's see if the other exchange show more opportunity.

In [36]:
```
df2 = btc_usd_datasets[['KRAKEN','BITFLYER']]
df2['price_dif']=df2.KRAKEN - df2.BITFLYER
df2["percent_change"] = (df2.price_dif / df2.KRAKEN)*100

df2.replace(0, np.nan, inplace=True)
df2.head(10)
```

Out[36]:

Date	KRAKEN	BITFLYER	price_dif	percent_change
2020-01-01	7196.538417	7207.406297	-10.867880	-0.151015
2020-01-02	7024.778058	6982.137692	42.640367	0.606999
2020-01-03	7233.475126	7185.435769	48.039357	0.664126
2020-01-04	7317.966324	7329.849319	-11.882995	-0.162381
2020-01-05	7415.106167	7410.826062	4.280106	0.057721
2020-01-06	7573.789593	7610.576632	-36.787040	-0.485715
2020-01-07	7967.765502	7960.304573	7.460929	0.093639
2020-01-08	8221.295753	8225.233853	-3.938100	-0.047901
2020-01-09	7886.044126	7908.924641	-22.880515	-0.290139
2020-01-10	7941.385199	7939.579177	1.806022	0.022742

In [37]:
```
fig = px.bar(df2, x=df2.index, y="percent_change", title="bitcoin Price Difference Between Kraken & Bitflyer")
fig.update_yaxes(ticklabelposition="inside top", title="Percent Change %")
fig.show()
```



In [38]:
```
# Frequency of dicrepency above 0.5%
df2.loc[df2.percent_change > 0.5, 'percent_change'].count()
```

Out[38]: 39

In [39]:
```
# Frequency of discrepncy below -0.5%
df2.loc[df2.percent_change < -0.5, 'percent_change'].count()
```

Out[39]: 26

1. There were 65 days in 2020 when the price difference passed 0.5% threshold betweeen Kraken and Bitflyer.

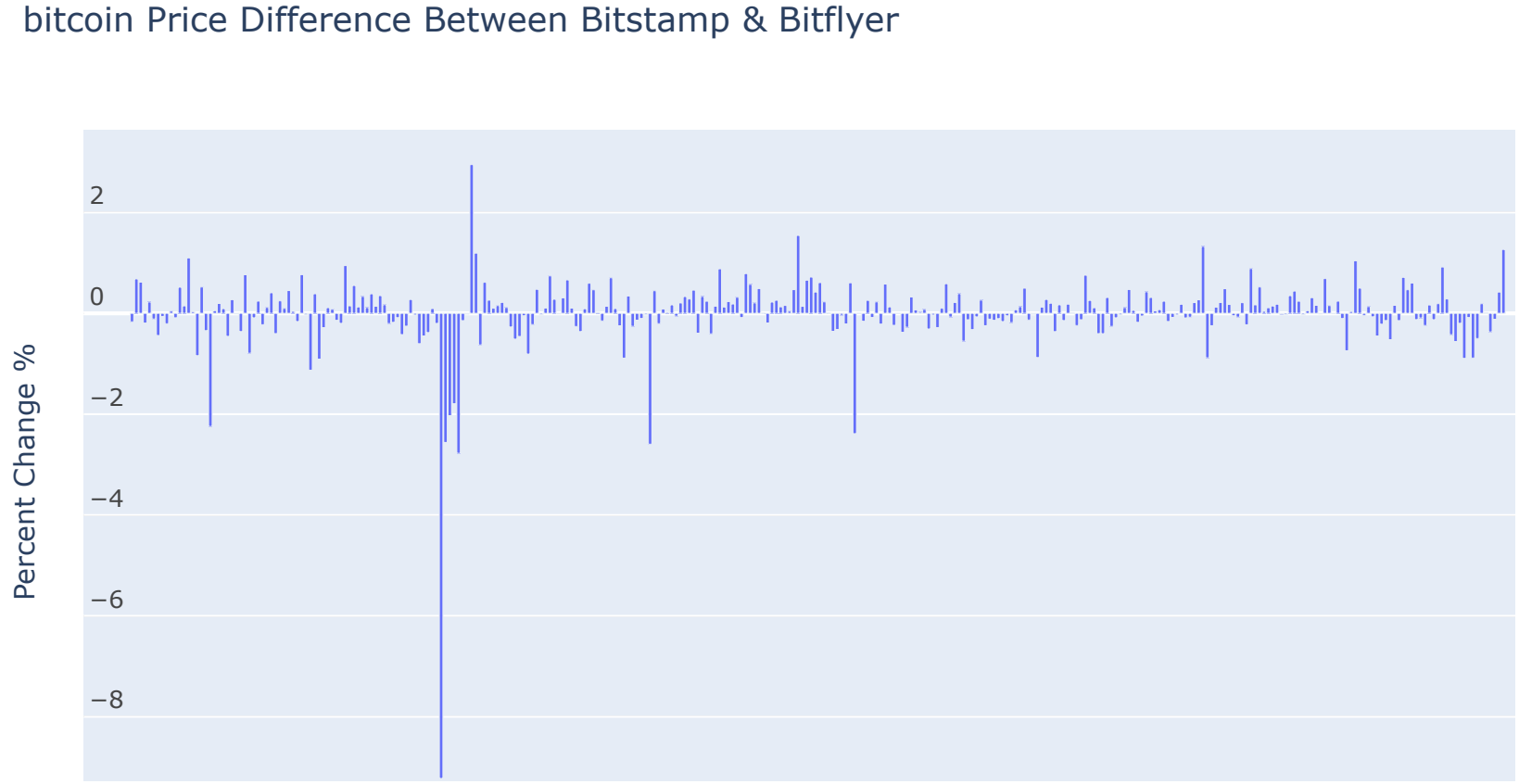
1. Finally, let's see the difference of prices between Bitstamp and Bitflyer.

In [40]:
```
df3 = btc_usd_datasets[['BITSTAMP','BITFLYER']]
df3['price_dif']=df3.BITSTAMP - df3.BITFLYER
df3["percent_change"] = (df3.price_dif / df3.BITSTAMP)*100

df3.replace(0, np.nan, inplace=True)
df3.head(10)
```

Out[40]:

Date	BITSTAMP	BITFLYER	price_dif	percent_change
2020-01-01	7195.247241	7207.406297	-12.159057	-0.168987
2020-01-02	7030.211788	6982.137692	48.074096	0.683821
2020-01-03	7230.186000	7185.435769	44.750231	0.618936
2020-01-04	7315.738130	7329.849319	-14.111190	-0.192888
2020-01-05	7427.569042	7410.826062	16.742980	0.225417
2020-01-06	7601.994206	7610.576632	-8.582426	-0.112897
2020-01-07	7925.503937	7960.304573	-34.800636	-0.439097
2020-01-08	8220.231471	8225.233853	-5.002383	-0.060855
2020-01-09	7892.620464	7908.924641	-16.304177	-0.206575
2020-01-10	7943.120177	7939.579177	3.541000	0.044579

In [41]:
```
fig = px.bar(df3, x=df3.index, y="percent_change", title="bitcoin Price Difference Between Bitsta
mp & Bitflyer")
fig.update_yaxes(ticklabelposition="inside top", title="Percent Change %")
fig.show()
```



In [46]:
```
# Frequency of dicrepency above 0.5%
df3.loc[df3.percent_change > 0.5, 'percent_change'].count()
```

Out[46]: 38

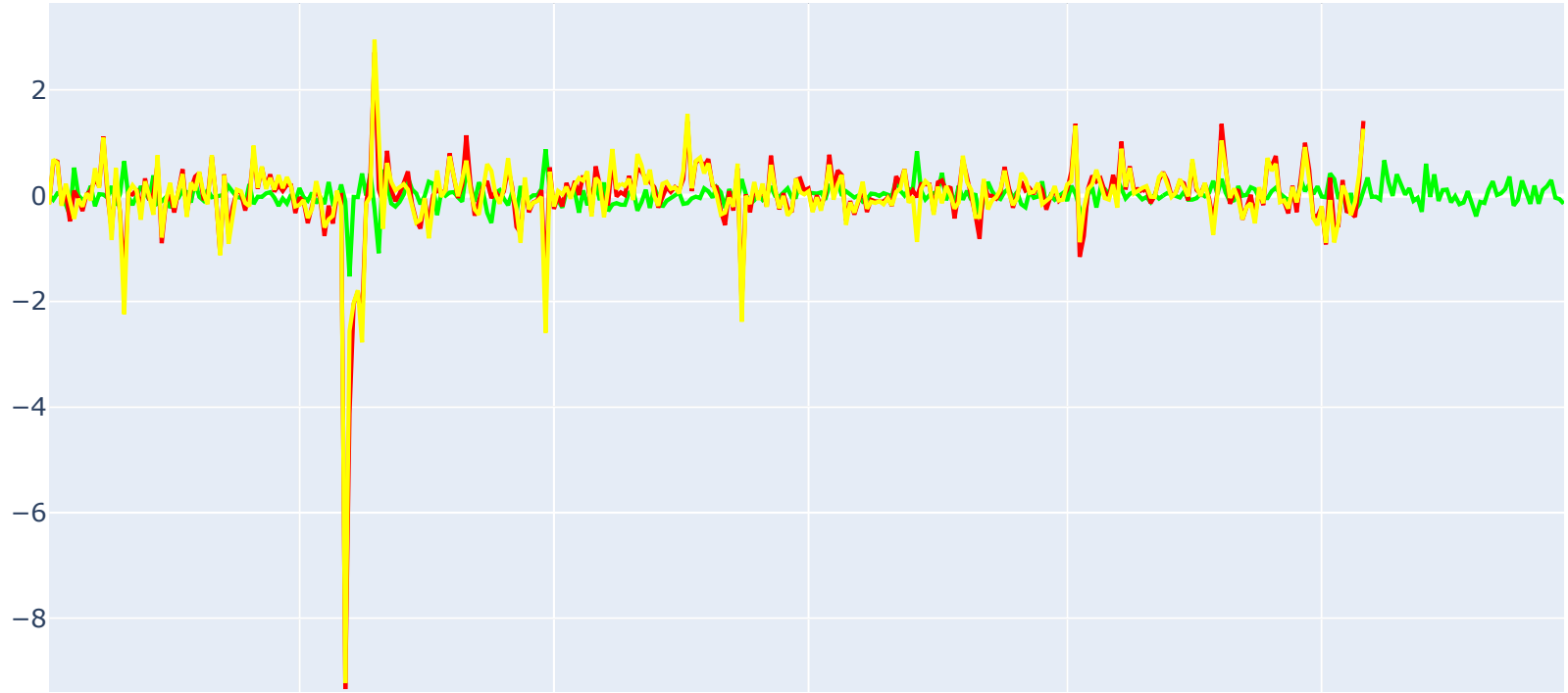
In [47]:
```
# Frequency of discrepency below -0.5%
df3.loc[df3.percent_change < -0.5, 'percent_change'].count()
```

Out[47]: 26

There are 64 days in 2020 when the price difference passed 0.5% threshold betweeen Bitstamp and Bitflyer. Let's visualize the percent change of all 3 exchanges.

In [51]:
```
trace0 = go.Scatter(x=df.index, y=df.percent_change, name='Kraken_Bitstamp', line=dict(color='lim
e'))
trace1 = go.Scatter(x=df2.index, y=df2.percent_change, name='Kraken_Bitflyer', line=dict(color='r
ed'))
trace2 = go.Scatter(x=df3.index, y=df3.percent_change, name='Bitstamp_Bitflyer', line=dict(color=
'yellow'))

data = [trace0, trace1, trace2]
layout = dict(
            title='Percent Price Difference: Kraken, Bitflyer, Bitstamp',
              xaxis = dict(
        range = ['2020-01-01','2020-12-31']
              )
             )

fig = dict(data=data, layout=layout)
py.iplot(fig)
```

Percent Price Difference: Kraken, Bitflyer, Bitstamp



Conclusion:

- Based on this data, it appears there are price differences between exchanges.
- The price difference are small, and seldom pass the 1% threshold.
- Arbitrage opportunity exists, however with low margin available.
- Best Arbitrage opportunity takes place between Bitflyer and Kraken, following closely Bistamp and Bitflyer.

In [ ]: