SIT220/731 2022.T1: Task 2P

Working with numpy Vectors (Unidimensional Data)

[redacted]

Undergradute (SIT220)

Importing Libraries:

The purpose of this code is to import libraries. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer.

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
```

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices.

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.

Loading files in notebook.

Note: If you get strange characters at the first element, make sure to use (encoding="utf-8-sig") because your computer is using a different encoder to open up the file.

Data from 01/01/2021 - 08/03/2022

The purpose of this code is to read external files such as csv, txt, etc.

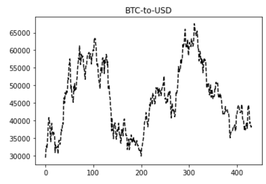
In [4]:
```
rates = np.loadtxt('Data.csv', encoding="utf-8-sig")
```

By using np.loadtxt from the libary numpy, we are able to read the file 'Data.csv'.

Plotting Data:

The purpose of this code is to generate a visual graph of 'Data.csv' file

In [5]:
```
x = np.arange(0,len(rates))
y = rates
plt.plot(x, y, linestyle = 'dashed', color = 'black')
plt.title('BTC-to-USD')
plt.show()
```



By using plt.plot from the libary matplotlib, a line graph is about to be created taking in data from 'Data.csv'

Printing arithmetic mean, median, minimum, maximum, standard deviation, and interquartile range of data:

The purpose of this code is to print the mean, median, minimum, maximum, standard deviation, and interquartile range of data.

In [6]:
```
mean = np.mean(rates)

median = np.median(rates)

minimum = np.min(rates)

maximum = np.max(rates)

std = np.std(rates)

iqr = np.subtract(*np.percentile(rates, [75, 25]))

#printed to 2 decimal place
print('Mean: ' + str(round(mean, 2)) + '\n' + 'Median: ' + str(round(median, 2)) + '\n' + 'Minimum: ' + str(round(minimum, 2)) + '\n' + 'Maximum: ' + str(round(maximum, 2)) + '\n'
+ 'Standard Deviation: ' + str(round(std, 2)) + '\n' + 'Interquartile Range: ' + str(round(iqr, 2)))
```

```
Mean: 46423.36
Median: 46378.41
Minimum: 29374.15
Maximum: 67566.83
Standard Deviation: 9363.83
Interquartile Range: 16496.37
```

Instead of manually calculating the mean, median, minimum, maximum, standard deviation, and interquartile range, numpy has in-bulit function where it automatically calculate the mean, median, etc.

Print the day with the lowest and highest observed prices:

The purpose of this code is to print out the corresponding day where price was the lowest and highest.

In [7]:
```
highestDay = np.argmax(y)
lowestDay = np.argmin(y)

print('The day where price was the highest for BTC is:  ' + str(x[highestDay]) + '\n' +
      'The day where price was the lowest for BTC is:  ' + str(x[lowestDay]))
```

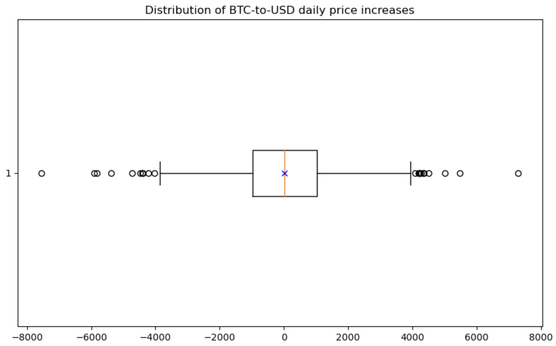
```
The day where price was the highest for BTC is:  311
The day where price was the lowest for BTC is:  0
```

As a result, between 01/01/2021 - 08/03/2022 day 311 was the day where price where the highest and day 0 was the day where price was the lowest.

Plotting Box Plot:

The purpose is this code is to create a boxplot from the data 'Data.csv'

In [8]:
```
mean_shape = dict(markerfacecolor='blue', marker='x', markeredgecolor='blue') #changing the mean shape icon
plt.figure(figsize = (10, 6), dpi = 100)
plt.boxplot(np.diff(y), vert = 0, showmeans=True, meanprops=mean_shape)
plt.title('Distribution of BTC-to-USD daily price increases')
plt.show()
```



Once again, by using functions within matplotlib libary, a boxplot was constructed which shows the daily price increase/decrease from 01/01/2021 - 08/03/2022 of BTC/USD.

The purpose of this code is to show the five number summary of the daily price increase/decrease of BTC/USD between 01/01/2021 - 08/03/2022.

In [9]:
```
print('Five Number Summary Of Boxplot Data:')

data1 = np.diff(y)

median1 = np.median(data1)

minimum1 = np.min(data1)

maximum1 = np.max(data1)

quarter1 = np.percentile(data1, 25)
quarter3 = np.percentile(data1, 75)

iqr1 = np.subtract(*np.percentile(data1, [75, 25]))

#printed to 2 decimal place
print('Median: ' + str(round(median1, 2)) + '\n' + 'Minimum: ' + str(round(minimum1, 2)) + '\n' + 'Maximum: ' + str(round(maximum1, 2)) + '\n'
+ 'First Quartile: ' + str(round(quarter1, 2)) + '\n' + 'Thrid Quartile: ' + str(round(quarter3, 2)) + '\n' + 'Interquartile Range: ' + str(round(iqr1, 2)))
```

```
Five Number Summary Of Boxplot Data:
Median: 12.52
Minimum: -7554.04
Maximum: 7293.02
First Quartile: -979.79
Thrid Quartile: 1023.7
Interquartile Range: 2003.49
```

A boxplot is a graph that represents the locality, spread and skewness of numerical data. With the data above, it generates a five number summary:

- Median: 12.52 meaning that 50% of daily price increae/decrease is lower than the median and 50% of daily price is greater than the median.
- Minimum: The minimum is the lowest number in the data. In this case, -7554.04 is the lowest daily price increase/decrease.
- Maximum: The Maximum is the lowest number in the data. In this case, 7293.02 is the highest daily price increase/decrease.
- First Quartile: The first quartile represents one quarter the data. In other words, 25% of daily price increase/decrease is less than 979.79 and 75% of daily price increase/decrease is higher than 979.79.
- Thrid Quartile: The third quartile represents three quarter the data In other words, 25% of daily price increase/decrease is higher than 1023.7 and 75% of daily price increase/decrease is less than 1023.7.
- Interquartile Range: In additional, the IQR serve as the data between Q1 and Q3. Therefore, we can conclude that the range between Q1 and Q3 of BTC/USD daily price increase/decrease is 2003.49.

Outliers:

The purpose of this code is to print out the number of outliers of the daily price increase/decrease data.

In [125…
```
data = np.diff(y)
q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)
iqr = q3 - q1

count = ((data < (q1 - 1.5 * iqr)) | (data > (q3 + 1.5 * iqr))).sum()

print('The number of outlier that ocurr in the boxplot is: ', count)
```

```
The number of outlier that ocurr in the boxplot is:  21
```

By calculating the lower and upper fence, data that is lower than the lower fence and data that is higher than the upper fence are considered outliers. In this case, using the sum() function, it counts all outliers which is 21 for the daily price increase/decrease of BTC/USD

An outlier is data point that lies an abnormal distance from the rest of the data. From the graph above, all the circle or points that are outside the whiskers of the boxplot are considered an outlier. This can affect the mean of the data as it can push the mean out of its usaul position. For example, let's say some data of (1,2,3), this would have the mean of 2, however if an outlier of 94 was added to the data set, then the mean would be 25. As a result, the mean moves towards the outlier which can skew the results.