SIT220: 2P Working with numpy Vectors (Unidimensional Data)

George David Nicholson (undergraduate (SIT220))

Student ID: 218403172

Email: gnicholson@deakin.edu.au

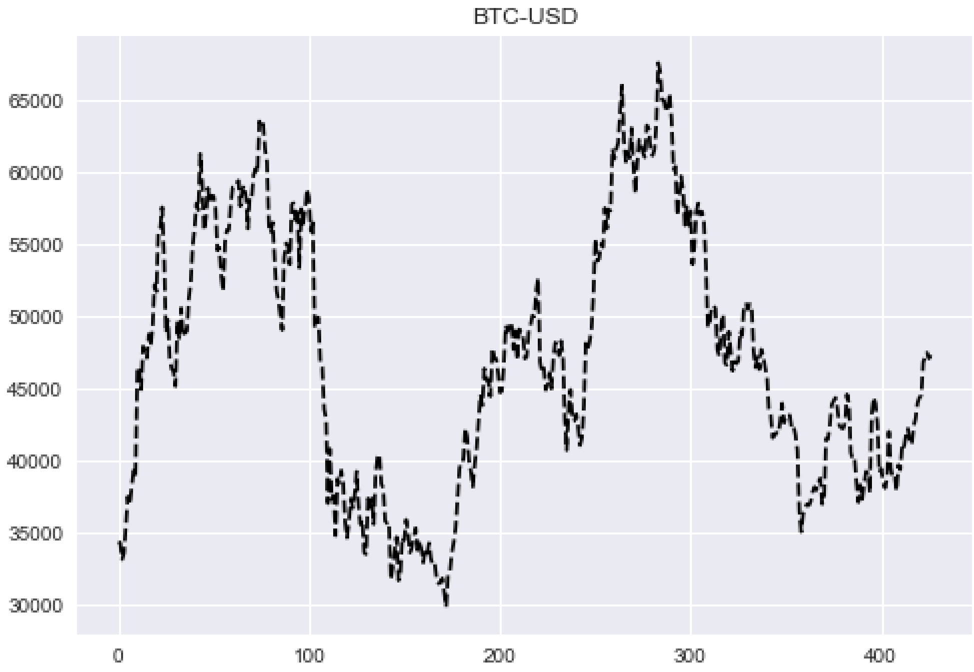
In [173…

```
import numpy as np
# import the csv using numpy
close = np.loadtxt("close.csv")
import matplotlib.pyplot as plt
import seaborn as sns
#declare plot style using seaborn
plt.style.use("seaborn")
```

below i have declared a title for the plotgraph and its line style before showing it

In [174…

```
plt.title("BTC-USD")
plt.plot(close, color='black', linestyle='dashed')
plt.show()
```



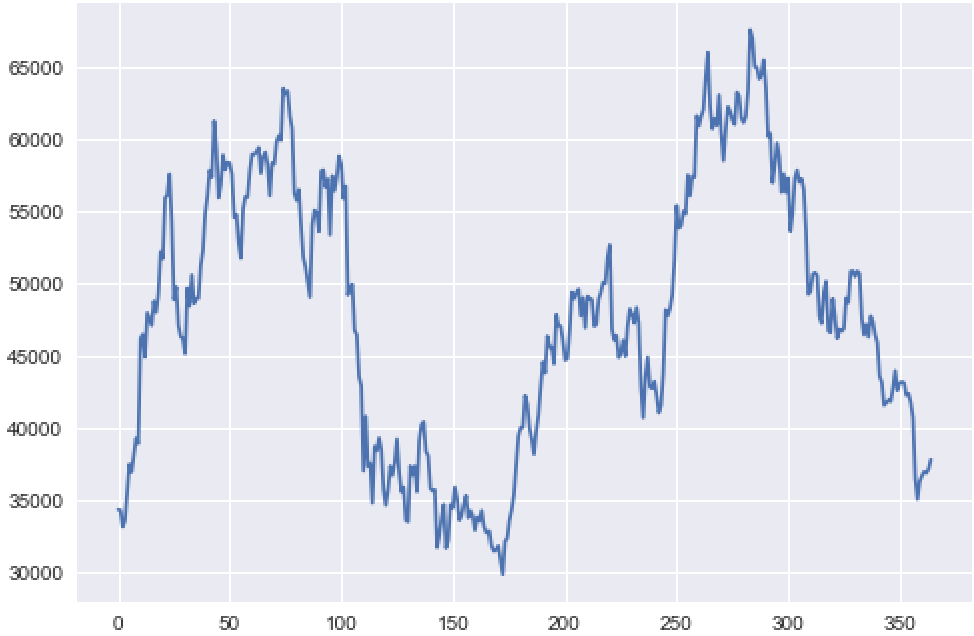
In [175…

```
# created a 2021 daily close vector to show 365 days of close into one vector called "_

_year21 = close[0:365]
plt.plot(_year21)
```

```
# used len to confirm that i had the right amount of daily closed index and charted for
r = str(len(_year21))
print(r + " days")
```

```
365 days
```



In [176…

```
#the arithmatic mean:
# another example for len and mean ==>   _year21.shape[0] # len of vector sorted...

# Used numpy to calculate mean with .mean
_mean = str(np.mean(_year21))

    # np.sum(_year21)/r
    # np.sum(_year21)/_year21.shape[0]


# median:
# Used numpy to calculate mean with .median
_median = str(np.median(_year21))

    # n_year21 = _year21.shape[0]


# minimum
# Used numpy to calculate mean with .min
_min = str(np.min(_year21))

# maximum
# Used numpy to calculate mean with .max
_max = str(np.max(_year21))


# Measures of dispersion
# saved the quantiles, could be useful later on working with the data and displaying it

_1qrt = np.quantile(_year21, 0.25)
_3qrt = np.quantile(_year21, 0.75)
```

```
    #use of QUA(N)Tile
   # declare the quantiles found in one line from one array. easy and nice trick
                     #min,   1st quartile, median, 3rd quartile, max
np.quantile(_year21, [0.0,    0.25,        0.5,      0.75,      1.0])

# interQUA(R)Tile range (IQR)
# calculated using 1st and 3rd quantile converting to quartile
_2021IQR = str((_3qrt - _1qrt)) # interquarterile range

#Standard deviation

#   np.std    (numpy function)

_std = str(np.std(_year21, ddof=1))

#all results below
print("The mean is " + _mean)
print("The median is " + _median)
print("The minimum is " + _min)
print("The maximum is "+ _max)
print("The interquartile range is " +_2021IQR)
print("The standard deviation is " + _std )
```

```
The mean is 47941.30820928767
The median is 47783.35938
The minimum is 29807.34766
The maximum is 67566.82813
The interquartile range is 15997.707029999998
The standard deviation is 9292.841874486065
```

In [177…

```
np.quantile(_year21, 0.25)
np.quantile(_year21, 0.75)
# used to visualise the data gathered from using np.quantile on the array
np.quantile(_year21, [0.0, 0.25, 0.5, 0.75, 1.0])
```

Out[177…

```
array([29807.34766, 40218.47656, 47783.35938, 56216.18359, 67566.82813])
```

In [178…

```
#find max element with amax
#find the index of max element with argmax

maxElement = str(np.amax(_year21))
maxIndex = str(np.argmax(_year21))

#find min element with amin
#find the index of max element with argmin

minElement = str(np.amin(_year21))
minIndex = str(np.argmin(_year21))

#present the data nicely by printing.
print("Highest Close: " + maxElement)
print("day: " + maxIndex + "/365")
print("---------------------------")
print("lowest Close: " + minElement)
print("day: " + minIndex + "/365")
```

```
Highest Close: 67566.82813
day: 283/365
---------------------------
lowest Close: 29807.34766
day: 172/365
```

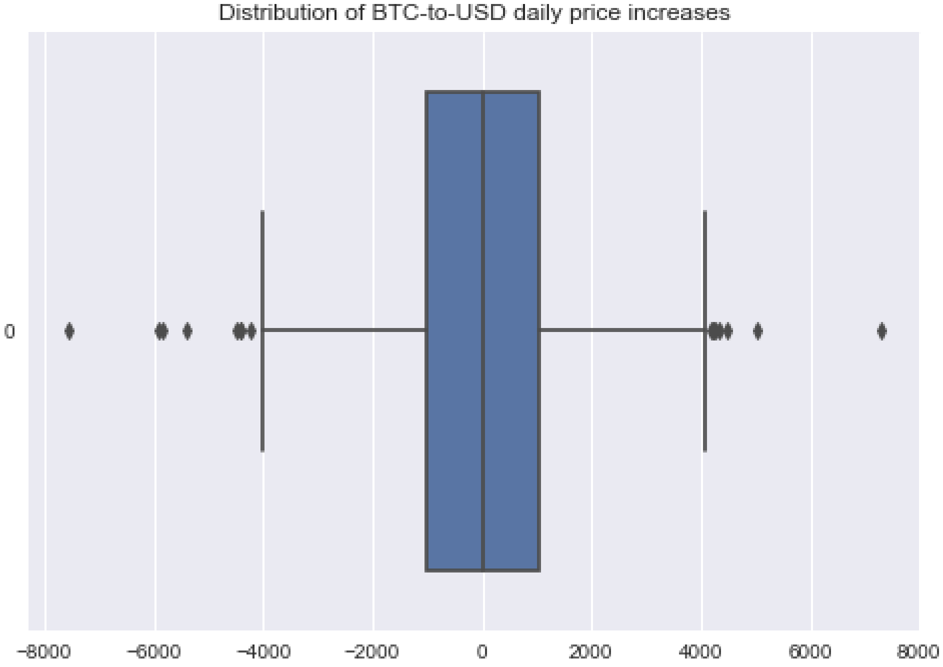
Box and Whisker plot aka boxplot

In [179…

```
# calculate the daily increase / decrease via np.diff
daily = np.diff(_year21)


sns.boxplot(data=daily, orient="h")
plt.title("Distribution of BTC-to-USD daily price increases")

plt.show()
```



from the boxplot above it is clear that the above data shows us that the daily price change ranged from an average increase of 4000 and decrease of 4000 as shown within the Interquartile range. this being said the median having the median at 0 and the Quartile 1 and 3 range show a 25% and 75% average of the price movement to be 1000 dollars either side.

with the additions of the outliers it is clear that bitcoin price ranges also had exceptions of breakouts above and below the 4000dollar min and max. this tells us that there will be days in which bitcoin will close drastically below and above the average determined. thus looking promising for investors looking for days with high returns, this could be further analysed to find these best days, or the days it took place

with this chart looking as it is, it shows an average of pricemovement and the unreliability of price movement either direction depending on the previous day close and next day close price