Crypto Currency Data Analysis Project

Contents

1. Introduction

It is quite common to visualize financial data for analysis and decision making. You will build a web page that displays crypto currency price trend against US Dollar from 2018. Through this project, you will learn how to :

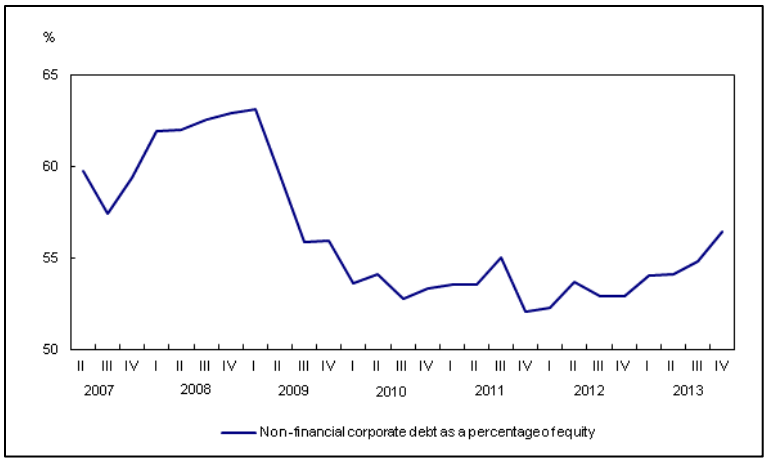
- Get real word from internet automatically.
- Transform data into visualization.
- Identify financial information through the chart.
- Use Django, Bokeh and AlphaVantage API.

We will start with Bitcoin price trend. After we finish it, we can easily extend it to display other crypto currencies.

Once you get the chart, try to think about:

1. What is the common price trending of major crypto currencies?
2. How is crypto currency price trending different than stocks?
3. What happened around the time when certain crypto currency price change sharply?

The result will be a line chart. This is sample line chart. It does not reflect any crypto currency price.



2. Design Consideration

2.1 Identify Data Source

There are many free or paid financial datasets on the internet. Instead of manually download dataset, We can get them programmatically through Application Program Interface, aka API. In this project, we can use AlphaVantage API, which provides free Real-Time data about stock, forex and crypto markets.

You will need to get a free API key in order use it.

https://www.alphavantage.co/support/#api-key

2.2 Understand Data Structure

You can view the sample dataset in json format at
https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_WEEKLY&symbol=BTC&market=CNY&apikey=demo

Sample data:


```
{
    "Meta Data": {
        "1. Information": "Weekly Prices and Volumes for Digital Currency",
        "2. Digital Currency Code": "BTC",
        "3. Digital Currency Name": "Bitcoin",
        "4. Market Code": "CNY",
        "5. Market Name": "Chinese Yuan",
        "6. Last Refreshed": "2021-04-23 00:00:00",
        "7. Time Zone": "UTC"
    },
    "Time Series (Digital Currency Weekly)": {
        "2021-04-23": {
            "1a. open (CNY)": "364436.02490400",
            "1b. open (USD)": "56150.01000000",
            "2a. high (CNY)": "373372.00762400",
            "2b. high (USD)": "57526.81000000",
            "3a. low (CNY)": "327765.20000000",
            "3b. low (USD)": "50500.00000000",
            "4a. close (CNY)": "335732.36071200",
```

```
            "4b. close (USD)": "51727.53000000",
            "5. volume": "325919.86460100",
            "6. market cap (USD)": "325919.86460100"
        },
        "2021-04-18": {
            "1a. open (CNY)": "389416.21152000",
            "1b. open (USD)": "59998.80000000",
            "2a. high (CNY)": "420928.40160000",
            "2b. high (USD)": "64854.00000000",
            "3a. low (CNY)": "330564.50952000",
            "3b. low (USD)": "50931.30000000",
            "4a. close (CNY)": "364436.02490400",
            "4b. close (USD)": "56150.01000000",
            "5. volume": "549048.29803200",
            "6. market cap (USD)": "549048.29803200"
        }
    }
}
```


The data you will need is within key `"Time Series (Digital Currency Weekly)"`. Within this object, you can find the date and prices in different currencies.

To visualize the data, you will use date as x axis, `"2b. high (USD)"` as y axis.

3. Development

Step 1: Set up Environment

We need to create a virtual Python environment.

```
mkdir RealCrypto
cd RealCrypto
```

Now, we'll install the packages and create the Django app.

```
pipenv install --python 3.9
pipenv install django pandas requests
pipenv install bokeh==2.3.1

pipenv shell
django-admin startproject config .
django-admin startapp crypto
```

Don't forget to add the app to your project settings.

```
#config/settings.py

# Application definition

INSTALLED_APPS = [
    ...
    'django.contrib.staticfiles',

    'crypto', # new app
]
```

Start the server. You should see Django sample site.

Step 2 : Utilities functions

First, we will get weekly Bitcoin price data. We need to make an API call to AlphaVantage API. Then, we are going to convert the JSON response into dataframe pandas format that we'll use with Bokeh in your view.

As you see, it's already a lot of things and one function can't handle that. So in crypto app folder, create a file utils.py. Let's add the first function.

```
# crypto/dataservice.py
import requests
import pandas as pd

# Crypto Currency
def get_cypto_data(cypto_currency, market,API_KEY):
    # Get Crypto Currency
    url =
'https://www.alphavantage.co/query?function=DIGITAL_CURRENCY_WEEKLY&interval=
5min&symbol=' + cypto_currency + '&market=' + market + '&apikey=' + API_KEY
    print(url)
    r = requests.get(url)
    dataIntraday = r.json()
    #print(dataIntraday)

    return dataIntraday['Time Series (Digital Currency Weekly)']
```

The second function will use pandas to convert the dictionary returned (`dataIntraday[Time Series (Digital Currency Weekly)']`) of get_data() in dataframe pandas format.

```
def convert_to_df_crypto(data):
    """Convert the result JSON in pandas dataframe"""
    df = pd.DataFrame.from_dict(data, orient='index')
    df = df.reset_index()

    #Rename columns
    df = df.rename(index=str, columns={"index": "date", "1b. open (USD)":
"open",
    "2b. high (USD)": "high", "3b. low (USD)": "low", "4b. close (USD)":
"close"})

    #Change to datetime
    df['date'] = pd.to_datetime(df['date'])

    #Sort data according to date
    df = df.sort_values(by=['date'])

    #Change the datatype
    df.open = df.open.astype(float)
    df.close = df.close.astype(float)
    df.high = df.high.astype(float)
```

```
    df.low = df.low.astype(float)

    #Checks
    df.head()
    df.info()

    return df
```

Step 3: Setting up the View

In view of the application, we can now use the power of Bokeh to visualize data.

```
# crypto/views.py
from django.shortcuts import render
from bokeh.plotting import figure, output_file, show
from bokeh.embed import components
import pandas as pd
from math import pi
import datetime
from .dataservice import get_cypto_data,convert_to_df_crypto

def homepage(request):
    crypto = 'BTC'

    market = 'USD'
    result = get_cypto_data(crypto,market, '1ZITU2X3HTCLIVY3') #repalce with
your own APIKEY

    source = convert_to_df_crypto(result)
```

First, we use the methods of our dataservice.py file. We then convert the result into dataframe.

```
    TOOLS = "pan, box_zoom, reset, save"

    title = f'{crypto} to {market} chart'

    p = figure(x_axis_type="datetime", tools=TOOLS, plot_width=700,
plot_height=500, title = title)
    p.xaxis.major_label_orientation = pi / 4
    p.line(x='date', y='high', line_width=2, source=source)

    script, div = components(p)

    return render(request,' crypto/base.html',{'script':script, 'div':div })
```

As we are dealing with time-series data, it's desirable to have an axis that can display labels that's are appropriate to different dates and time scales.

Step 3: Setup Template

In `crypto` application folder, create a folder templates. Then inside this folder, create another folder `crypto` (same name as your app), create HTML file base.html. Feel free to style it the way you want.

```
<!-- crypto/templates/crypto/base.html -->
<html>

<head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
    <title>Financial Data| by PwA </title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.4.0/css/bulma.css" />
</head>

<body>
    <nav class="navbar hero is-dark is-bold has-text-centered"
role="navigation" aria-label="main navigation">
        <div class="navbar-brand">
            <a class="navbar-item">
                <h2 class="title has-text-weight-normal is-1 has-text-white">
                    Crypto Currency Price - PwA
                </h2>
            </a>
        </div>
    </nav>
    <section class="container">
        <div class="columns">
            <div class="column">

            </div>
            <div class="column is-6">
                {{ div | safe }}
            </div>
            <div class="column">

            </div>
        </div>
    </section>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/bokeh/2.3.1/bokeh.min.js"></scrip
t>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/bokeh/2.3.1/bokeh-
widgets.min.js"></script>
    {{ script | safe }}
</body>

</html>
```

We need the HTML file to contain a div where the visualization will be displayed and the Bokeh dependencies in the file.

CSS files in the head, and JS files at the end of the code. For the sake of performance.

Step 5: Setup URL

Create urls.py in `crypto` folder and add these lines.

```
#crypto/urls.py
from django.urls import path
from . import views

urlpatterns = [
    path('', views.homepage, name='homepage')
]
```

Because we separated the URLs file, we should notify it in the project configu/urls.py file.

```
#config/urls.py

from django.contrib import admin
from django.urls import path, include
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('crypto.urls')),
]
```

Good job! Let's see what you get.

4. Add Tooltips

With the basic chart you can easily see the trending of the bitcoin price over 2 years. Next, we will add tooltips to the chart so we can see price and date when we hover over the line.

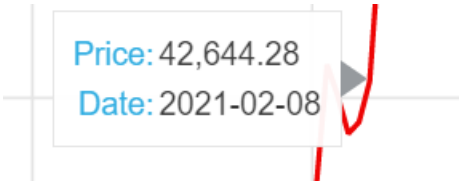
Add the following code to crypto/views.py.

```
     line = p.line(x='date', y='high', line_width=2,line_color="#ed0505",
source=source)

    #setup tooltip
    hover_tool = HoverTool(tooltips=[
                    ('Price', '$y{0,0.00}'),
                    ('Date', '$x{%F}'),],
                    formatters={'$x': 'datetime'},

        )
    p.tools.append(hover_tool)
```

After you launch the page, you can move your mouse over the line, you should see the price and date at that point.



You are most done!

5. Add More Crypto Currencies

You've done a lot of heavy lifting to display bitcoin price chart. With couple of lines of code, you can enable a user to select different crypto currency from a dropdown list. It's up to you if you want to add this feature.

Next, add a dropdown list in crypto/templates/crypto/base.html.

Find

```
    <section class="container">
            <div class="columns">
                <div class="column">
```

Add

```
     <form method="POST">
        {% csrf_token %}
        <select name="crypto" id="currency">
           <option value="">Please select</option>
           <option value="BTC">Bitcoin</option>
           <option value="ETH">Ethereum</option>
        </select>
           <br><br>
        <input type="submit" value="Submit">
     </form>
```

You can add your favorite crypto currencies. You can find crypto currency at
`https://coinmarketcap.com/`

You can add more options, such as
`<option value="DOGE">Dogecoin</option>`

In crypto/views.py, after

```
def homepage(request):
    crypto = 'BTC'
```

add

```
    if request.method == 'POST':
        crypto = request.POST['crypto']
```

This will take user's selection.

It's time to launch your website. You should be able to select different crypto currency and see the price chart.

Now you've done data visualization for crypto currency by using Django, bokeh and pandas.

6. Food to thought

Use the same technology, you could extent this project to

1. Visualize stock data.
2. Visualize foreign currency exchange rate from any currency to any currency.
3. Plot multiple line charts in one figure.
4. Plot different type of charts, such as vbar etc.

Wow, well done!