

Analyzing the best ETF investments





What are ETFs?

Exchange-Traded Funds (ETFs) are a type of investment fund traded on stock exchanges. They typically consist of multiple assets.

Advantages:

- ❖ Diversification
 - Lower risk
 - ETFs have varied investments, so it spreads out risk from just investing in one stock
- ❖ Transparency
- ❖ Dividends
- ❖ Lower Costs



Testing the best ETF investment

- ❖ Which sector has performed the best over time?
- ❖ How stable has a fund's performance been?
- ❖ How many dividends have been received by investors?



Analyzing Data

- 1) Creating a Python Flask-powered App
- 2) Creating a database in MongoDB based on an API
- 3) Connecting MongoDB to Flask app and jsonifying data
- 4) Creating an HTML and JavaScript frontend



Mongodb

Creating a database in Mongodb

Connecting the Database to JupyterNotebook

Adding info about the ETFs to the database

Querying the database

alphaVantage_db.data

10 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter Type a query: { field: 'value' } Explain Reset Find Options

ADD DATA EXPORT DATA

1 - 10 of 10

Object
<pre>{ "_id": "Object('650f1bb7f60231f4e05fd443')", "Meta Data": Object, "Weekly Adjusted Time Series": Object }</pre>
Object
<pre>{ "_id": "Object('650f22e0f60231f4e05fd444')", "Meta Data": Object, "Weekly Adjusted Time Series": Object }</pre>
Object
<pre>{ "_id": "Object('650f22e0f60231f4e05fd445')", "Meta Data": Object, "Weekly Adjusted Time Series": Object }</pre>



Python Flask-powered App

Import modules

Connect to MongoDB database

Loop through database and enter data into dictionary

Jsonify dictionary and return

Create route to Json and HTML file

```
#####  
# Flask Setup  
#####  
app = Flask(__name__)  
  
#####  
# Database Setup  
#####  
  
#Connecting to Mongo  
mongo = MongoClient(port=27017)  
db = mongo['alphaVantage_db']  
data = db['data']
```

```
    Add route to return dataset  
app.route("/api/stocks")  
def stocks():  
  
    #query data for all stocks  
    allData = data.find({})  
  
    #create dictionary  
    results = {}  
  
    # loop through stocks and add to dictionary with ticker as k  
    for d in allData:  
        d.pop("_id")  
  
        print(d["Meta Data"]["2 Symbol"])  
        ticker = d["Meta Data"]["2 Symbol"]  
  
        results[ticker] = d
```



JavaScript and HTML

```
let url = "http://127.0.0.1:5000/api/stocks"
var dataSet;

function init() {
  d3.json(url).then(function(data) {
    dataSet = data;
    bar("AGG", data);

    metadataDisplay("AGG", data);

    lineChart("AGG", data);
    CandlestickChart("AGG", data);
    // Populate the dropdown menu
    var titles = Object.keys(data);

    let dropdownMenu = d3.select("#selDataset");
    titles.forEach(function(name) {
      dropdownMenu.append("option").text(name).property("value", name);
    });
  });
};
```

In JavaScript, get data from Flask route

Using Plotly create line chart with dividend data, Candlestick chart with price movement data, and

Use new library Chart.js to create bar chart of trading volumes

Create metadata display to show what each ETF is tracking and what sector it primarily deals with

Create HTML frontend with proper imported modules to display visualizations

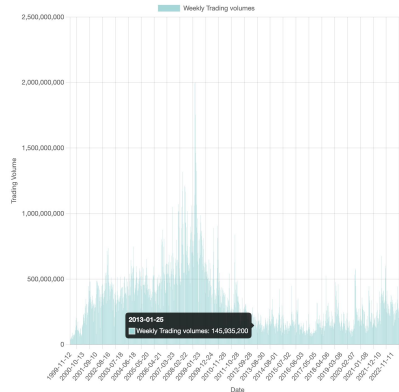
Website Visualizations

Stock ID #:

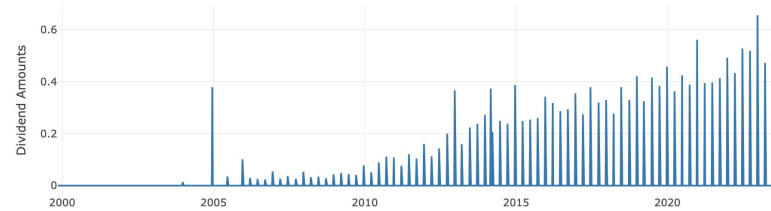
QQQ ▾

Stock Info

QQQ tracks the
NASDAQ-100 Index
Sector: Technology



QQQ ETF Dividend Amounts



Candlestick Chart for QQQ ETF Weekly Price Movements





Results

Sectors such as real estate, bonds, commodities, VNQ, are very affected by market conditions — 2008 financial crisis

Technology sector shows consistent growth but since it is still based on the status of a specific field it's status could change quickly. This is indicated by the other ETFs that are focused on only one sector. A major market or societal change will be very impactful on ETF performance.

ETFs that track various sectors are more protected from other factors: SPY has holdings in various sectors and it shows the most consistent growth in dividends and stock prices over time