Cryptocurrency Investment Analysis Report

Assessing Risk-Return Profiles and Simulated Outcomes

Abstract:
My project leverages historical data from 23 major cryptocurrencies to analyze and compute key risk-return metrics; annualized returns, volatility, and Sharpe ratios. I constructed a data-analytics pipeline that included an exploratory data analysis, investment simulation, and visualization of results through informative graphs and heatmaps via MATPLOTLIB & SEABORN. My findings provided actionable insights into which coins may be optimal for long-term investment.

Background:
Cryptocurrencies are essentially “digital money” that uses encryption to perform secure transactions. They operate on decentralized networks (blockchains), meaning no single bank or government controls them. This allows for fast, global, and transparent transfers of value.

Objectives:
- Quantify risk and return by calculating annualized returns, volatility, and Sharpe ratios for each cryptocurrency.
- Identify coins with favorable risk-adjusted performance for long-term investment.
- Develop a reproducible pipeline using Python, Jupyter Notebook, and related libraries (Pandas, NumPy, Matplotlib, Seaborn, etc.).

Tools:
- Python - Jupyter Notebook - NumPy - Pandas - Matplotlib - Seaborn - SciPy

Setup & Environment:

This code imports all required libraries and configures settings to ensure consistent visualizations throughout the project.

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
import glob
import os

# Set a random seed for reproducibility
np.random.seed(42)

# Configure visualization settings for Seaborn and Matplotlib
sns.set(style="darkgrid", palette="deep", font="Arial")
plt.style.use("dark_background")

# Configure pandas display options for better readability
pd.set_option("display.max_columns", None)
pd.set_option("display.width", 1200)

# Display confirmation message
print("Environment setup complete! All libraries imported and configurations set.")
```

Edit

Data Loading & Preprocessing:

This code reads the CSV files, converts date strings to datetime objects, sorts the data by date, and calculates daily returns for each cryptocurrency.

```
# Dictionary to store the processed DataFrame for each coin
coin_data = {}

# Process each coin CSV file using the provided folder path
for coin, filename in file_names.items():
    path = os.path.join(data_dir, filename)
    try:
        print(f"Processing {coin} from {path}...")
        df = pd.read_csv(path)

        # Convert the 'Date' column to datetime format
        df["Date"] = pd.to_datetime(df["Date"])

        # Sort the data by date in ascending order
        df = df.sort_values("Date")

        # Calculate daily returns using the 'Close' price
        df["Daily Return"] = df["Close"].pct_change()

        # Store the processed DataFrame in the dictionary
        coin_data[coin] = df

        print(f"Processed {coin}: {len(df)} records.")
        display(df.head(3))
    except Exception as e:
        print(f"Error processing {coin} from {path}: {e}")
```

Exploratory Data Analysis (EDA):

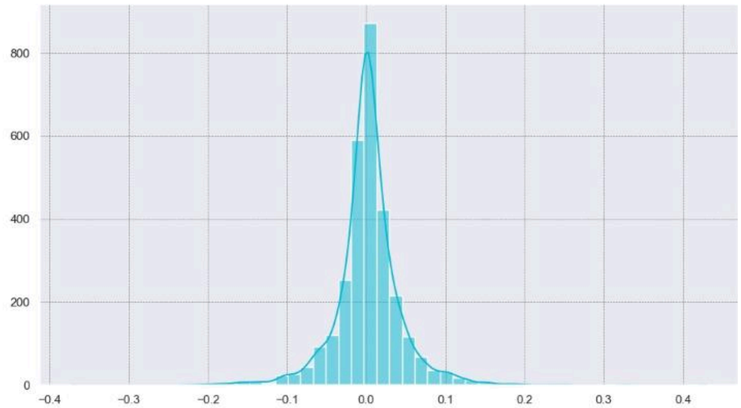
In the EDA phase, I analyzed statistics and visualized the distribution and temporal trends of daily returns for each coin. By creating histograms with KDE overlays, time series plots, and heat maps we develop a comprehensive understanding of our datas' behavior.

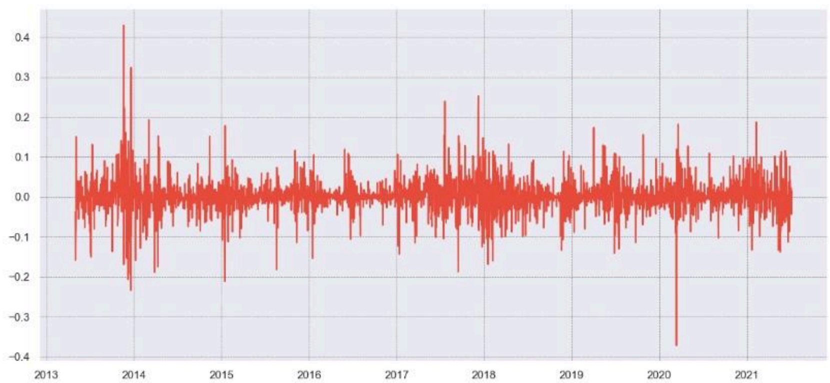
```python
import matplotlib.pyplot as plt
import seaborn as sns

# Set dark theme styles for consistency
plt.style.use("dark_background")
sns.set_style("dark")

# Loop through each coin's DataFrame in coin_data to perform EDA
for coin, df in coin_data.items():
    print(f"\n--- Exploratory Data Analysis for {coin} ---")

    # Compute descriptive statistics for daily returns (excluding NaN)
    daily_returns = df["Daily Return"].dropna()
    desc_stats = daily_returns.describe()
    print("Descriptive Statistics for Daily Return:")
    print(desc_stats)

    # Plot histogram with KDE for daily returns
    plt.figure(figsize=(10, 6))
    sns.histplot(daily_returns, bins=50, kde=True, color="#00BCD4")
    plt.title(f"{coin} Daily Returns Distribution", fontsize=16, fontweight="bold", color="#FFFFFF")
    plt.xlabel("Daily Return", fontsize=14, color="#FFFFFF")
    plt.ylabel("Frequency", fontsize=14, color="#FFFFFF")
    plt.grid(color="#555555", linestyle="--", linewidth=0.5, alpha=0.7)
    plt.tight_layout()
    plt.show()

    # Plot time series of daily returns
    plt.figure(figsize=(12, 6))
    plt.plot(df["Date"], df["Daily Return"], color="#E74C3C", linewidth=1.5)
    plt.title(f"{coin} Daily Returns Over Time", fontsize=16, fontweight="bold", color="#FFFFFF")
    plt.xlabel("Date", fontsize=14, color="#FFFFFF")
    plt.ylabel("Daily Return", fontsize=14, color="#FFFFFF")
    plt.grid(color="#555555", linestyle="--", linewidth=0.5, alpha=0.7)
    plt.tight_layout()
    plt.show()
```

Across our 23 cryptocurrencies that were analyzed, our data reveals a market that is characterized by high volatility and wide-ranging daily returns and fluctuations. Most currencies exhibited modest daily gains, though these small returns are accompanied by some significant variability, as indicated by large standard deviations. In most cases, the majority of daily returns had clustered within a narrow band, but there was a presence of extreme outliers both on the positive and negative side of things. While more established coins such as Bitcoin and Ethereum tended to show relatively consistent, modest gains, other less known coins displayed more erratic behavior, highlighting a diverse risk profile across the crypto landscape. These findings underscore that despite the potential for steady growth, investing in crypto as a whole inherently comes with some considerable risk.

EDA - Continued:

Because there are 23 different coins below is an example output for Bitcoin



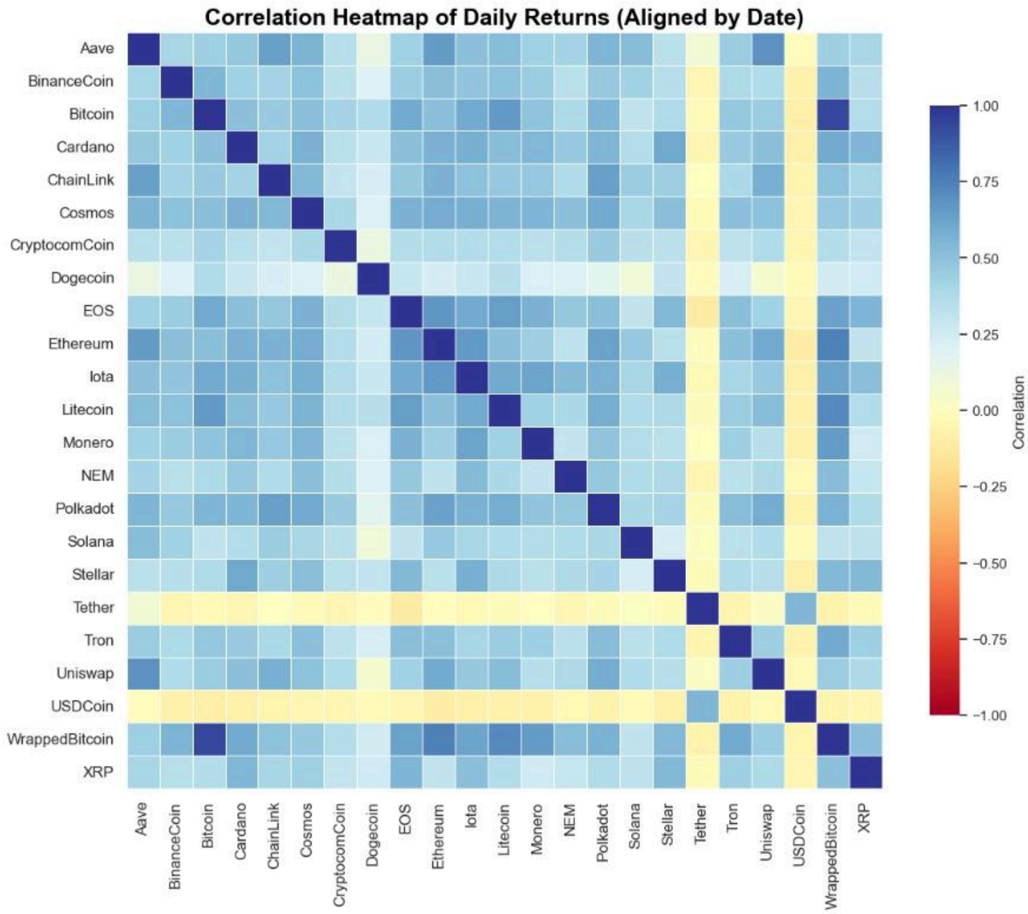



```
--- Exploratory Data Analysis for Bitcoin ---
Descriptive Statistics for Daily Return:
count    2990.000000
mean        0.002741
std         0.042639
min        -0.371695
25%        -0.013007
50%         0.001883
75%         0.018669
max         0.429680
Name: Daily Return, dtype: float64
```

Bitcoin's daily returns, averaged over 2,990 observations, are modestly positive at around 0.27%, but the high standard deviation (4.26%) shows significant daily fluctuations, with typical returns between -1.3% and +1.87% and occasional extreme movements from -37% to +43%.

This analysis was then performed on our other 22 currencies (pictured in Jupyter).

EDA - Continued:



- Dark Blue: Indicates a strong positive correlation (values near 1). This means the coins move very similarly; for example, when a coin is compared with itself, the correlation is 1, so it appears dark blue.
- Dark Red: Indicates a strong negative correlation (values near -1), meaning the coins tend to move in opposite directions.
- Lighter/Neutral Colors: Represent correlations near 0, suggesting little to no linear relationship.

The heat map above highlights how closely each cryptocurrency's daily move in tandem. For the most part, coins show moderate to strong correlation with the exception of a few outliers. The idea is that when the market sentiment shifts many assets tend to follow suit.

EDA - Continued:

Due to the outcome of the heat map I came to three conclusions:

- Limited Diversification: If you hold multiple coins that are highly correlated, you may not significantly reduce your portfolio's overall risk.
- Market-Wide Trends: Strong correlations suggest that external factors (like macroeconomic news or regulatory changes) can move the entire crypto market together.
- Identifying Outliers: Coins with weaker correlations to the rest of the market could offer better diversification benefits.

Bitcoin Time Series Decomposition Analysis - (EDA)

- Close: Shows a long period of low prices before a rapid climb beginning around 2017, with major spikes in 2020–2021.
- Trend: Remains flat until 2017, then surges upward, reflecting Bitcoin's overall long-term growth.
- Seasonal: Exhibits a repeating wave, suggesting some cyclical behavior, though it may be exaggerated by Bitcoin's volatility.
- Residual: Contains large, unpredictable price movements driven by market news and sentiment, emphasizing Bitcoin's high volatility.

EDA - Continued:

The above plot underscores Bitcoin's strong upward trend in recent years, occasional wave-like seasonal artifacts and significant unexplained volatility in our residual. This goes to highlight that Bitcoin (and crypto in general) are a high-reward and high-risk option of investing.

I singled out Bitcoin to perform this analysis on for its popularity and reputation

Key Takeaways:

- High Volatility: Cryptocurrencies show small average daily gains but experience significant price swings.
- Market-Wide Trends: Many coins tend to move together, indicating that overall market sentiment drives their performance.
- Growth & Cycles: A strong long-term upward trend exists with recurring cycles and unpredictable market shocks.

Overall our EDA findings emphasize the idea that while the crypto market offers significant growth opportunities, it also carries the substantial risk and market-wide volatility–insights that are crucial for developing effective, risk-managed investment strategies.

Risk-Return Analysis:

In this piece I assessed the risk and reward characteristics of each one of the 23 cryptocurrencies by calculating key metrics. Those being annualized volatility, returns, and Sharpe ratio. The model provides a quantitative basis to understand how much return the asset generates relative to their volatility, helping to identify which coins will offer the most attractive balance for our long-term investments.

Sharpe Ratio: A measure of risk-adjusted return. Calculated by taking the difference between the investment's return and the risk-free rate, then dividing by our investments standard deviation or in other words volatility. The higher Sharpe ratio indicates that an investment is generating more return per unit of risk.

```python
import numpy as np

# Number of trading days in a year
trading_days = 252

# List to collect risk-return metrics for each coin
metrics_list = []

for coin, df in coin_data.items():
    # Drop NaN values from daily returns
    daily_returns = df["Daily Return"].dropna()

    # Compute the mean and standard deviation of daily returns
    mean_daily = daily_returns.mean()
    std_daily = daily_returns.std()

    # Annualize the returns and volatility
    annual_return = mean_daily * trading_days
    annual_volatility = std_daily * np.sqrt(trading_days)

    # Calculate Sharpe Ratio (assuming risk-free rate = 0)
    sharpe_ratio = annual_return / annual_volatility if annual_volatility != 0 else np.nan

    metrics_list.append({
        "Coin": coin,
        "Annual Return": annual_return,
        "Annual Volatility": annual_volatility,
        "Sharpe Ratio": sharpe_ratio,
        "Records": len(df)
    })

# Create a summary DataFrame with risk-return metrics for all coins
risk_return_df = pd.DataFrame(metrics_list)

# Sort the DataFrame by Sharpe Ratio in descending order for easier interpretation
risk_return_df = risk_return_df.sort_values("Sharpe Ratio", ascending=False)
print("Risk-Return Analysis Summary:")
display(risk_return_df)
```

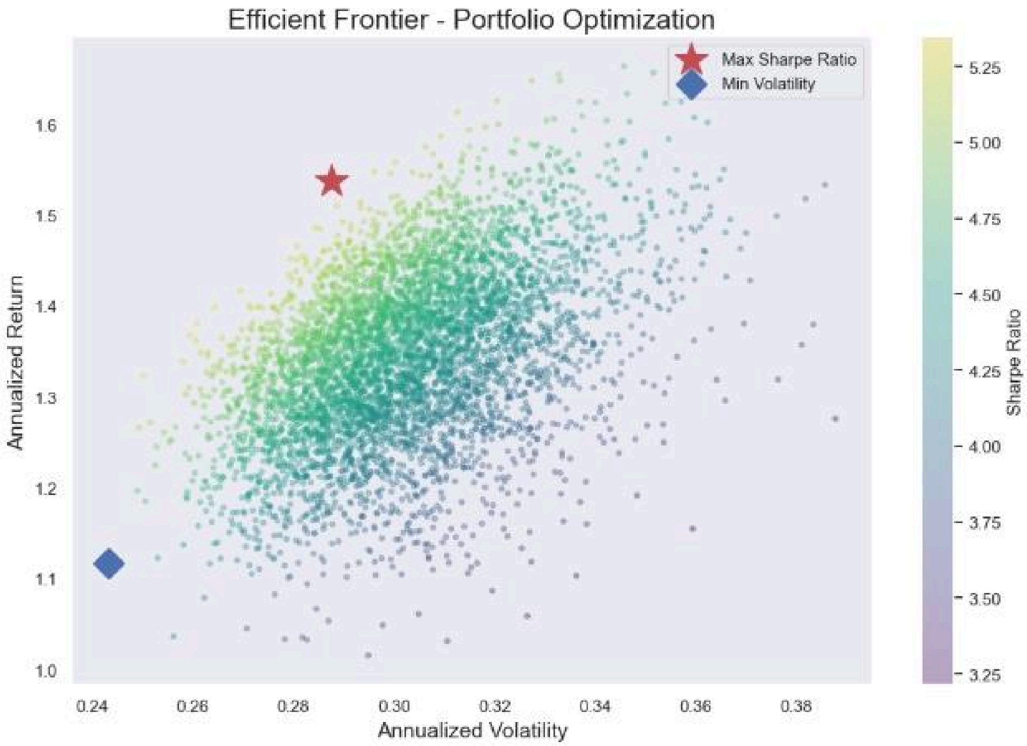
The above code snippet calculates each coin's average daily return and volatility, we then take these metrics and implement them into our Sharpe ratio to measure the risk-adjusted performance. It then compiles the results and displays them in a sorted DataFrame.

Risk-Return Analysis - Continued:

	Coin	Annual Return	Annual Volatility	Sharpe Ratio	Records
15	Solana	3.224374	1.500256	2.149217	452
0	Aave	2.581229	1.373899	1.878762	275
1	BinanceCoin	2.140179	1.270752	1.684183	1442
14	Polkadot	2.272454	1.383948	1.642008	320
9	Ethereum	1.428717	1.000659	1.427775	2160
19	Uniswap	2.023868	1.449326	1.396420	292
21	WrappedBitcoin	0.885093	0.680340	1.300957	888
4	ChainLink	1.660273	1.276387	1.300760	1385
3	Cardano	1.480840	1.327077	1.115866	1374
18	Tron	1.653802	1.512271	1.093588	1392
13	NEM	1.487687	1.379369	1.078527	2288
2	Bitcoin	0.690636	0.676877	1.020327	2991
12	Monero	1.043324	1.108635	0.941089	2602
6	CryptocomCoin	1.210821	1.293483	0.936093	935
7	Dogecoin	1.668655	1.801096	0.926467	2760
16	Stellar	1.186855	1.294257	0.917016	2527
22	XRP	1.128528	1.294819	0.871572	2893
11	Litecoin	0.822026	1.087915	0.755598	2991
5	Cosmos	0.836330	1.142848	0.731795	845
10	Iota	0.756095	1.167613	0.647556	1484
8	EOS	0.757062	1.197870	0.632007	1466
17	Tether	0.019788	0.281545	0.070283	2318
20	USDCoin	0.000944	0.072900	0.012951	1002

The figure above displays our coins sorted by their Sharpe ratio, meaning that coins at the top demonstrate a historically higher return per unit of volatility. Coins at the top are the most appealing to investors on a risk-adjusted basis. On the other hand, coins at the bottom of the totem tend to show lower returns paired with lower volatility, these are considered stablecoins.

Risk-Return Analysis - Continued:



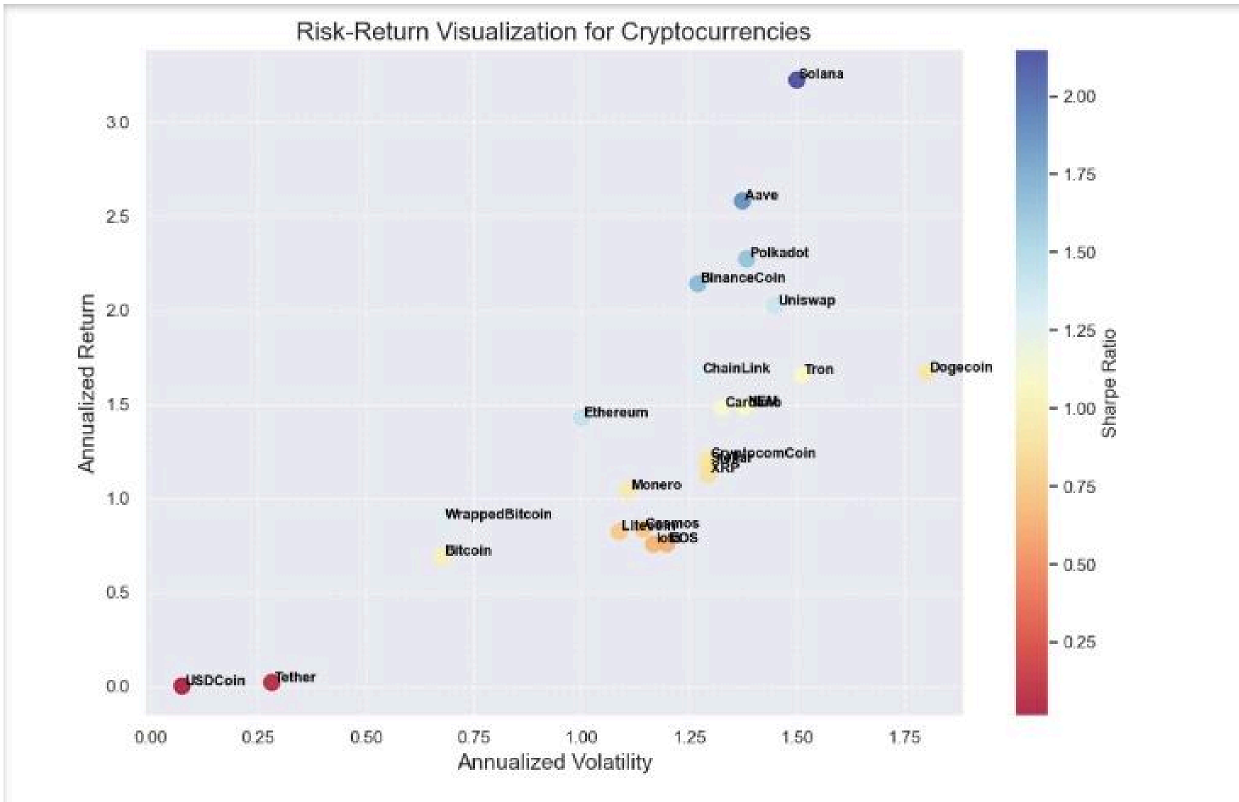
The figure above depicts how different blends of our 23 coins can achieve various combinations of risk and reward. Each point represents a randomly generated portfolio with unique sets of weights across our assets. The star represents the dataset that will provide the max sharpe ratio. Below is both the code for the figure along with the best portfolio for maximizing our Sharpe ratio.

```
Aave: 6.95%
BinanceCoin: 7.82%
Bitcoin: 7.05%
Cardano: 5.55%
ChainLink: 1.64%
Cosmos: 0.60%
CryptocomCoin: 0.39%
Dogecoin: 5.02%
EOS: 0.30%
Ethereum: 6.06%
Iota: 5.84%
Litecoin: 0.65%
Monero: 7.43%
NEM: 4.28%
Polkadot: 7.41%
Solana: 7.39%
Stellar: 1.61%
Tether: 3.12%
Tron: 1.10%
Uniswap: 6.35%
USDCoin: 4.13%
WrappedBitcoin: 7.61%
XRP: 1.71%
```

Risk-Return Analysis - Continued:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Ensure that coin_data is already defined from the Data Loading & Preprocessing step

# Build a DataFrame of daily returns from coin_data
daily_returns_df = pd.DataFrame({coin: df["Daily Return"] for coin, df in coin_data.items()})

# Number of trading days per year
trading_days = 252

# Calculate expected annual returns and annual covariance matrix from daily returns
expected_daily_returns = daily_returns_df.mean()
expected_annual_returns = expected_daily_returns * trading_days
cov_daily = daily_returns_df.cov()
cov_annual = cov_daily * trading_days

# Number of random portfolios to simulate
num_portfolios = 5000
num_assets = len(daily_returns_df.columns)

# Arrays to store simulation results: volatility, return, Sharpe ratio
results = np.zeros((3, num_portfolios))
weights_record = []

# Simulate random portfolios
for i in range(num_portfolios):
    # Generate random weights and normalize them so that they sum to 1
    weights = np.random.random(num_assets)
    weights /= np.sum(weights)
    weights_record.append(weights)

    # Calculate portfolio expected return and volatility
    portfolio_return = np.dot(weights, expected_annual_returns)
    portfolio_volatility = np.sqrt(np.dot(weights.T, np.dot(cov_annual, weights)))

    # Calculate Sharpe Ratio (assume risk-free rate = 0)
    sharpe_ratio = portfolio_return / portfolio_volatility if portfolio_volatility != 0 else 0

    # Save the results
    results[0, i] = portfolio_volatility
    results[1, i] = portfolio_return
    results[2, i] = sharpe_ratio

# Identify portfolio with highest Sharpe Ratio and minimum volatility
max_sharpe_idx = np.argmax(results[2])
max_sharpe_vol = results[0, max_sharpe_idx]
max_sharpe_ret = results[1, max_sharpe_idx]

min_vol_idx = np.argmin(results[0])
min_vol_vol = results[0, min_vol_idx]
min_vol_ret = results[1, min_vol_idx]

# Plot the efficient frontier
plt.figure(figsize=(10, 7))
plt.scatter(results[0, :], results[1, :], c=results[2, :], cmap='viridis', marker='o', s=10, alpha=0.3)
plt.colorbar(label='Sharpe Ratio')
plt.scatter(max_sharpe_vol, max_sharpe_ret, marker='*', color='r', s=500, label='Max Sharpe Ratio')
plt.scatter(min_vol_vol, min_vol_ret, marker='D', color='b', s=200, label='Min Volatility')
plt.title('Efficient Frontier - Portfolio Optimization', fontsize=18)
plt.xlabel('Annualized Volatility', fontsize=14)
plt.ylabel('Annualized Return', fontsize=14)
plt.legend()
plt.tight_layout()
plt.show()
```

Visualization of Results:



The scatter plot compares our coins annualized return against their annualized volatility with the color indicating the coin's corresponding Sharpe ratio. Heres what we can take away from the figure.

- Risk V Reward: Coins towards the right side have higher volatility meaning that their prices can swing widely, meanwhile those higher up on the chart have delivered greater annualized returns.
- Sharpe Ratio: Brighter colors indicate a higher Sharpe ratio, these coins historically offer stronger returns relative to their volatility. Cooler colors show lower risk-adjusted returns.
- Cluster & Outliers: Solana and Aave combine strong gains with more dramatic price fluctuations. Eth and ChainLink found a balance between potential returns and volatility. Stablecoins like USDC showed close to zero returns and minimum volatility (hence the name stablecoin).

Investment Simulations:

```
investment = 100000.0  # Total investment in USD

# Define the investment date as January 1, 2021
invest_date = pd.Timestamp("2021-01-01")

# Dictionaries to store our computed values
portfolio_investment = {}  # Money allocated to each coin initially
units_purchased = {}       # Units bought per coin
initial_values = {}        # Closing price on the investment date for each coin
current_values = {}        # Latest price (today) for each coin

# For each coin, determine the closing price closest to the investment date
for coin, weight in optimized_weights.items():
    df = coin_data[coin]
    # Find the row where the date is closest to invest_date
    idx = (df["Date"] - invest_date).abs().idxmin()
    invest_row = df.loc[idx]
    invest_price = invest_row["Close"]
    initial_values[coin] = invest_price
    portfolio_investment[coin] = investment * weight
    units_purchased[coin] = portfolio_investment[coin] / invest_price

# Initialize CoinGecko API client and fetch the current prices in USD for all coins
cg = CoinGeckoAPI()
current_prices_data = cg.get_price(ids=list(coin_gecko_ids.values()), vs_currencies='usd')
for coin in optimized_weights.keys():
    cg_id = coin_gecko_ids[coin]
    current_price = current_prices_data[cg_id]['usd']
    current_values[coin] = current_price

# Calculate the current value of each coin and sum up the total portfolio value
portfolio_current_value = 0
print("Optimized Portfolio Performance (from 2021-01-01 to today):")
print("----------------------------------------------------------------")
for coin in optimized_weights.keys():
    current_val = units_purchased[coin] * current_values[coin]
    portfolio_current_value += current_val
    print(f"{coin}:")
    print(f"  Invested: ${portfolio_investment[coin]:.2f} at ${initial_values[coin]:.2f} per unit on {invest_date.date()}")
    print(f"  Today's Price: ${current_values[coin]:.2f} -> Current Value: ${current_val:.2f}\n")

print(f"Total Portfolio Value Today: ${portfolio_current_value:.2f}")
print(f"Net Gain/Loss: ${portfolio_current_value - investment:.2f}")
```

Above is the code to calculate what would've happened if we invested $100,000 diversified according to our portfolio optimization piece that we looked into in our risk-return analysis. The money was invested on the first day of 2021 and then compared to current day numbers. Results are posted on the following page. We witnessed net gains of just under a million dollars or in other words a 10x return on our investments.

Investment Simulations - Continued:

```
Optimized Portfolio Performance (from 2021-01-01 to today):
--------------------------------------------------------------
Aave:
  Invested: $6950.00 at $87.53 per unit on 2021-01-01
  Today's Price: $173.09 -> Current Value: $13743.97

BinanceCoin:
  Invested: $7820.00 at $37.38 per unit on 2021-01-01
  Today's Price: $619.11 -> Current Value: $129533.28

Bitcoin:
  Invested: $7050.00 at $29001.72 per unit on 2021-01-01
  Today's Price: $84030.00 -> Current Value: $20426.77

Cardano:
  Invested: $5550.00 at $0.18 per unit on 2021-01-01
  Today's Price: $0.72 -> Current Value: $22038.29

ChainLink:
  Invested: $1640.00 at $11.27 per unit on 2021-01-01
  Today's Price: $14.20 -> Current Value: $2066.27

Cosmos:
  Invested: $600.00 at $6.49 per unit on 2021-01-01
  Today's Price: $4.73 -> Current Value: $437.07

CryptocomCoin:
  Invested: $390.00 at $0.06 per unit on 2021-01-01
  Today's Price: $0.08 -> Current Value: $526.87

Dogecoin:
  Invested: $5020.00 at $0.00 per unit on 2021-01-01
  Today's Price: $0.17 -> Current Value: $180625.65

EOS:
  Invested: $300.00 at $2.60 per unit on 2021-01-01
  Today's Price: $0.55 -> Current Value: $63.88

Ethereum:
  Invested: $6060.00 at $737.80 per unit on 2021-01-01
  Today's Price: $1957.72 -> Current Value: $16079.87

Iota:
  Invested: $5840.00 at $0.30 per unit on 2021-01-01
  Today's Price: $0.18 -> Current Value: $3597.87

Litecoin:
  Invested: $650.00 at $124.69 per unit on 2021-01-01
  Today's Price: $91.02 -> Current Value: $474.48

Monero:
  Invested: $7430.00 at $156.57 per unit on 2021-01-01
  Today's Price: $208.62 -> Current Value: $9899.88

NEM:
  Invested: $4280.00 at $0.20 per unit on 2021-01-01
  Today's Price: $0.02 -> Current Value: $404.87

Polkadot:
  Invested: $7410.00 at $9.29 per unit on 2021-01-01
  Today's Price: $4.37 -> Current Value: $3484.71

Solana:
  Invested: $7390.00 at $1.51 per unit on 2021-01-01
  Today's Price: $128.13 -> Current Value: $626700.63

Stellar:
  Invested: $1610.00 at $0.13 per unit on 2021-01-01
  Today's Price: $0.28 -> Current Value: $3509.17

Tether:
  Invested: $3120.00 at $1.00 per unit on 2021-01-01
  Today's Price: $1.00 -> Current Value: $3116.81

Tron:
  Invested: $1100.00 at $0.03 per unit on 2021-01-01
  Today's Price: $0.23 -> Current Value: $9448.22

Uniswap:
  Invested: $6350.00 at $5.17 per unit on 2021-01-01
  Today's Price: $6.68 -> Current Value: $8207.28

USDCoin:
  Invested: $4130.00 at $1.00 per unit on 2021-01-01
  Today's Price: $1.00 -> Current Value: $4129.96

WrappedBitcoin:
  Invested: $7610.00 at $28977.74 per unit on 2021-01-01
  Today's Price: $83910.00 -> Current Value: $22036.05

XRP:
  Invested: $1710.00 at $0.22 per unit on 2021-01-01
  Today's Price: $2.42 -> Current Value: $18823.21

Total Portfolio Value Today: $1099375.08
Net Gain/Loss: $999375.08
```

The above figure represents the results from simulating our diversified portfolio running it from 2021-01-01 to today's values.

Investing in all 23 of the coins with optimized weights led to substantial gains. The simulation shows that while individual coins experienced different levels of volatility that the portfolio grew significantly.

Investment Simulations - Continued:

```
# Dictionaries to store simulation data for the top 5 coins
portfolio_investment_top5 = {}  # Dollar amount invested per coin
units_purchased_top5 = {}       # Number of units bought per coin
initial_values_top5 = {}        # Closing price on 2021-01-01 for each coin
current_values_top5 = {}        # Latest price (today) for each coin

# For each coin, find the closing price closest to January 1, 2021 and calculate units purchased
for coin in top5_sharpe:
    df = coin_data[coin]  # Assumes coin_data has been populated from previous preprocessing
    # Find the row with the date closest to invest_date
    idx = (df["Date"] - invest_date).abs().idxmin()
    invest_row = df.loc[idx]
    invest_price = invest_row["Close"]
    initial_values_top5[coin] = invest_price
    portfolio_investment_top5[coin] = investment_top5
    units_purchased_top5[coin] = investment_top5 / invest_price

# Initialize the CoinGecko API client and fetch current prices for the top 5 coins
cg = CoinGeckoAPI()
# Assume coin_gecko_ids mapping is defined correctly, for example:
# {
#   "Solana": "solana", "Aave": "aave", "BinanceCoin": "binancecoin",
#   "Polkadot": "polkadot", "Ethereum": "ethereum", ...
# }
top5_ids = [coin_gecko_ids[coin] for coin in top5_sharpe]
current_prices_data_top5 = cg.get_price(ids=top5_ids, vs_currencies='usd')

for coin in top5_sharpe:
    cg_id = coin_gecko_ids[coin]
    current_price = current_prices_data_top5[cg_id]['usd']
    current_values_top5[coin] = current_price

# Calculate the current value of each coin and the total portfolio value
portfolio_current_value_top5 = 0
print("Optimized Portfolio Performance for Top 5 Coins (Invested $20,000 each on 2021-01-01):")
print("-----------------------------------------------------------------------------")
for coin in top5_sharpe:
    current_val = units_purchased_top5[coin] * current_values_top5[coin]
    portfolio_current_value_top5 += current_val
    print(f"{coin}:")
    print(f"  Invested: ${portfolio_investment_top5[coin]:.2f} at ${initial_values_top5[coin]:.2f} per unit on {invest_date.date()}")
    print(f"  Today's Price: ${current_values_top5[coin]:.2f} -> Current Value: ${current_val:.2f}\n")

total_investment_top5 = investment_top5 * len(top5_sharpe)
print(f"Total Portfolio Value Today: ${portfolio_current_value_top5:.2f}")
print(f"Net Gain/Loss: ${portfolio_current_value_top5 - total_investment_top5:.2f}")
```

Above is our second simulation that we performed where we took our top 5 cryptos with the best Sharpe ratio and invested $20,000 in each for a total of $100,000 instead of diversifying the $100,000 into all 23 of our cryptos like we did in our first figure.

Investment Simulations - Continued:

```
Optimized Portfolio Performance for Top 5 Coins (Invested $20,000 each on 2021-01-01):
--------------------------------------------------------------------------------
Solana:
  Invested: $20000.00 at $1.51 per unit on 2021-01-01
  Today's Price: $127.37 -> Current Value: $1686017.23

Aave:
  Invested: $20000.00 at $87.53 per unit on 2021-01-01
  Today's Price: $172.34 -> Current Value: $39379.63

BinanceCoin:
  Invested: $20000.00 at $37.38 per unit on 2021-01-01
  Today's Price: $617.43 -> Current Value: $330388.19

Polkadot:
  Invested: $20000.00 at $9.29 per unit on 2021-01-01
  Today's Price: $4.36 -> Current Value: $9383.90

Ethereum:
  Invested: $20000.00 at $737.80 per unit on 2021-01-01
  Today's Price: $1953.30 -> Current Value: $52949.06

Total Portfolio Value Today: $2118118.03
Net Gain/Loss: $2018118.03
```

Above we focused on our top 5 coins based on our risk-return analysis, we witnessed very impressive gains. The portfolio's value today sits at just over two million dollars, well above our initial $100,000 investment. This data-driven approach to single out the best Sharpe ratios proved to be a better investment strategy over the diversification strategy.

Conclusion:

The project aimed to investigate the historical performance of 23 major cryptocurrencies through a data analytics pipeline. Our EDA revealed that while the coins can deliver sizable gains, they also exhibit high volatility and frequently move in tandem, limiting our diversification benefits. Additionally by looking into decomposing time series, we were able to further highlight the idea that the market's cyclical tendencies and the vulnerability of news-driven price swings.

Our Risk-Return Analysis calculated annualized returns, volatility, and Sharpe ratios. Providing insight into how different coins balance risk and reward. The metric proved immensely useful in identifying strong performers relative to their volatility, meanwhile stablecoins provided negligible returns.

The Investment Simulations confirmed the findings, the diversified portfolio of all 23 coins witnessed substantial growth over the past few years. Our more selective approach of investing solely in the top five coins via Sharpe ratio produced an even greater return. These findings underscore the potential benefits of focusing on our higher risk-adjusted assets. Additionally an emphasis is put on the inherent risk in our often volatile market.

Key Takeaways

- Volatility and Correlation: Defining features of the crypto market, with coins often moving in sync with one another.
- Risk-Adjusted Metrics: Metrics like the Sharpe ratio can provide strong insight into long-term investment strategies. Identifying coins that deliver strong returns for each unit of risk sacrificed.
- Selective Allocation: The higher performing assets may significantly boost our returns, but they also increase our exposure to market swings.

The provided insights highlight the importance of robust data analysis and risk management in the ever evolving cryptocurrency world. Past performance cannot guarantee future success, though, a methodical approach such as mine can provide valuable guidance for investors who seek to navigate a high-risk, high-reward market.