STOCK EDA ANALYTICS PROJECT: AAPL, JPM, PG, UAL (2017-2022)

If you want to be the first to be informed about new projects, please do not forget to follow us - by Fatma Nur AZMAN

Fatmanurazman.com | Linkedin | Github | Kaggle | Tableau

TABLE of CONTENTS

UNDERSTANDING THE DATA

Project Description:

- In this project, we will perform basic Exploratory Data Analysis (EDA) on the Daily Stock Prices Dataset and Daily Stock Returns Dataset.
- we will visualize stock prices using Seaborn and Matplotlib.
- 4 Stocks are considered including JP Morgan Chase (JP), Procter and Gamble (P&G) (PG), Apple (AAPL) and United Airlines (UAL).

About the Datasets

Dataset Descriptions: 'stocks_daily_prices.csv' / 'stocks_daily_returns.csv'

- **Content:** Daily stock prices for various companies.
- **Rows:** 3642
- **Columns:** 5
  - **Date:** Date of the recorded stock price and returns.
  - **AAPL**: Daily return for Apple Inc.
  - **JPM**: Daily return for JPMorgan Chase & Co.
  - **PG**: Daily return for Procter & Gamble Co.
  - **UAL**: Daily return for United Airlines Holdings, Inc.
- **Purpose:** These datasets Provide a comprehensive view of stock price movements over time for analysis of trends, patterns, and correlations.


- Apple (AAPL):Apple Inc. is an American multinational technology company known for its innovative electronic products, software, and online services, holding a significant market share worldwide.
- JP Morgan Chase (JP): JP Morgan Chase is an American multinational investment bank and financial services company that provides financial services and is one of the largest banks in the world.
- Procter and Gamble (P&G): Procter and Gamble is an American-based company that produces many consumer products, such as personal care and cleaning products, as well as healthcare products.
- United Airlines (UAL): United Airlines is a major airline operating in the United States, with a vast flight network worldwide.
- **Investment Opportunities**: Despite operating in different sectors, these four companies offer various opportunities to investors, making them attractive options for portfolio diversification.

Import The Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from tabulate import tabulate
import plotly.graph_objects as go

# Enable inline plotting
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
warnings.warn("this will not show")

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

In [2]:
```
#Import the "kyphosis.csv" file using Pandas
df_prices = pd.read_csv('stocks_daily_prices.csv')
df_returns = pd.read_csv('stocks_daily_returns.csv')
```

Performing Essential Statistical Analysis on the Dataset

In [11]:
```
df_prices.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3642 entries, 0 to 3641
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    3642 non-null   object 
 1   AAPL    3642 non-null   float64
 2   JPM     3642 non-null   float64
 3   PG      3642 non-null   float64
 4   UAL     3642 non-null   float64
dtypes: float64(4), object(1)
memory usage: 142.4+ KB
```

In [12]:
```
df_returns.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3642 entries, 0 to 3641
Data columns (total 5 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    3642 non-null   object 
 1   AAPL    3642 non-null   float64
 2   JPM     3642 non-null   float64
 3   PG      3642 non-null   float64
 4   UAL     3642 non-null   float64
dtypes: float64(4), object(1)
memory usage: 142.4+ KB
```

In [13]:
```
df_prices.head()
```

Out[13]:

	Date	AAPL	JPM	PG	UAL
0	1/2/2008	5.949703	29.448614	47.058838	29.915234
1	1/3/2008	5.952452	29.246094	47.058838	29.690800
2	1/4/2008	5.498071	28.582682	46.870098	31.000002
3	1/7/2008	5.424478	28.868999	47.175980	29.180000
4	1/8/2008	5.229351	27.723747	47.299629	24.389999

In [14]:
```
df_returns.head()
```

Out[14]:

	Date	AAPL	JPM	PG	UAL
0	1/2/2008	0.000000	0.000000	0.000000	0.000000
1	1/3/2008	0.046203	-0.687708	0.000000	-0.750233
2	1/4/2008	-7.633517	-2.268378	-0.401072	4.409454
3	1/7/2008	-1.338518	1.001718	0.652615	-5.870973
4	1/8/2008	-3.597157	-3.967066	0.262103	-16.415356

In [17]:
```
df_prices.describe().T
```

Out[17]:

	count	mean	std	min	25%	50%	75%	max
AAPL	3642.0	39.648625	43.326924	2.387943	11.441511	23.470343	45.398705	181.511704
JPM	3642.0	64.927239	39.662998	11.545819	31.298827	49.601177	95.376392	169.500061
PG	3642.0	73.729653	32.623908	29.637867	46.483078	65.661438	82.645994	163.410004
UAL	3642.0	43.183288	24.392486	3.130000	22.320000	43.080002	60.425000	96.699997

In [18]:
```
df_returns.describe().T
```

Out[18]:

	count	mean	std	min	25%	50%	75%	max
AAPL	3642.0	0.105180	2.003500	-17.919513	-0.797108	0.099403	1.129585	13.904950
JPM	3642.0	0.069331	2.559122	-20.727418	-0.889705	0.017241	1.002644	25.096737
PG	3642.0	0.035579	1.198977	-8.737353	-0.482903	0.037285	0.595436	12.009046
UAL	3642.0	0.098866	4.361402	-36.770882	-1.553845	0.016636	1.636975	68.537079

In [19]:
```
df_prices.duplicated().sum()
```

Out[19]:
```
0
```

In [20]:
```
df_returns.duplicated().sum()
```

Out[20]:
```
0
```

In [21]:
```
df_prices.isnull().sum()
```

Out[21]:
```
Date    0
AAPL    0
JPM     0
PG      0
UAL     0
dtype: int64
```

In [22]:
```
df_returns.isnull().sum()
```

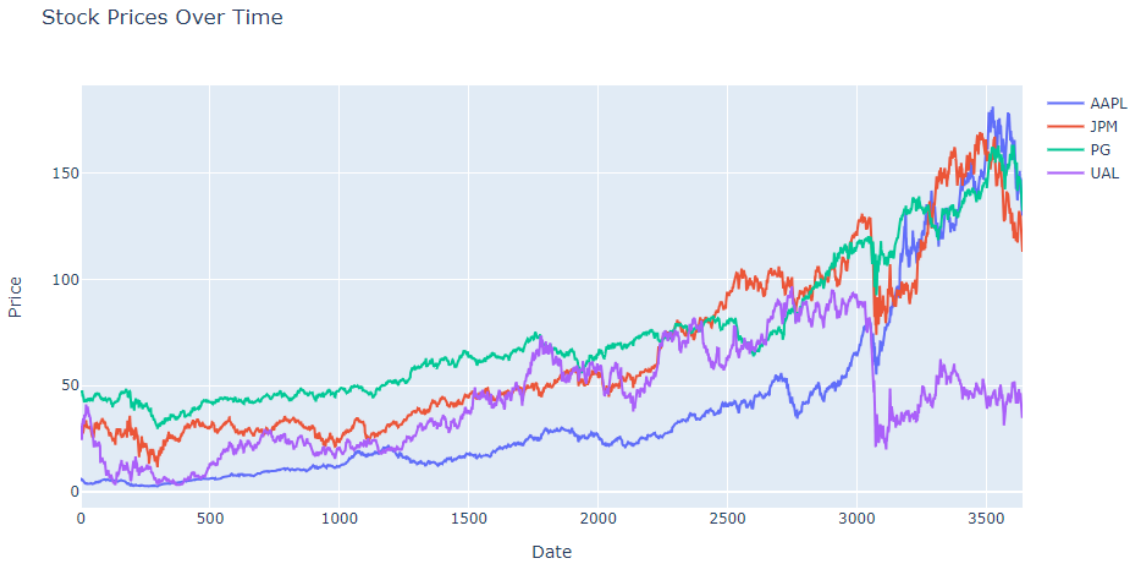
Out[22]:
```
Date    0
AAPL    0
JPM     0
PG      0
UAL     0
dtype: int64
```

In [23]:
```
# Convert 'Date' column to datetime format on df_prices
df_prices['Date'] = pd.to_datetime(df_prices['Date'])
```

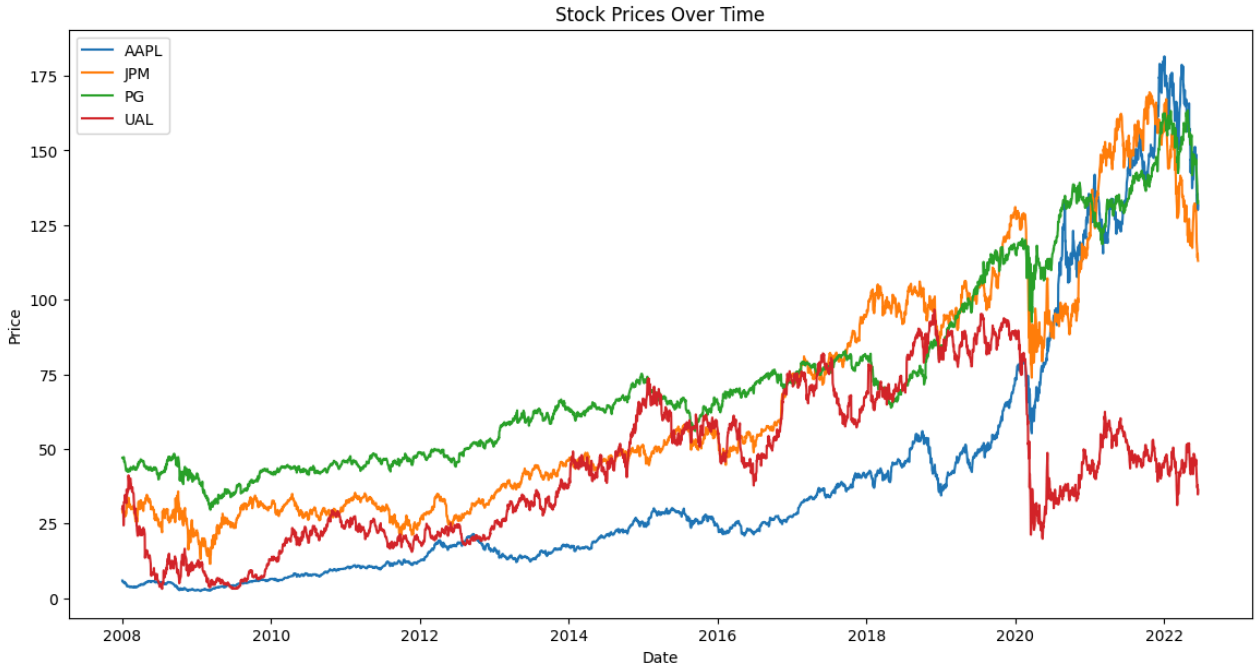
In [24]:
```
# Convert 'Date' column to datetime format on df_return
df_returns['Date'] = pd.to_datetime(df_returns['Date'])
```

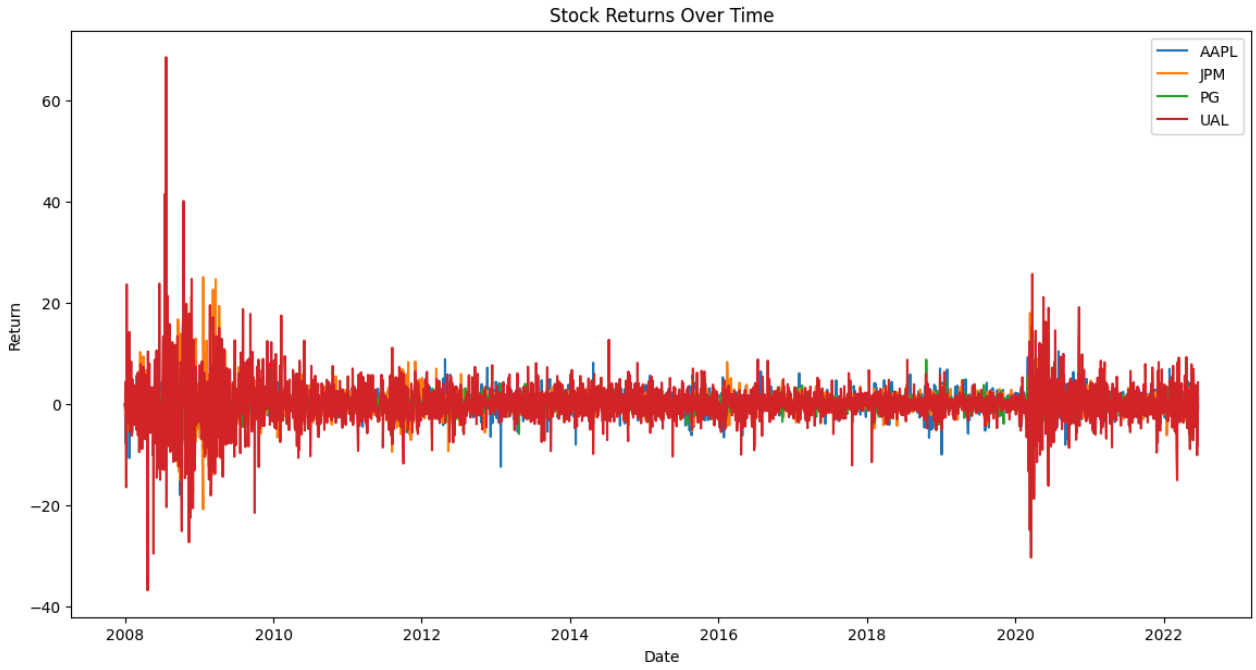
STOCK PRICES AND STOCK RETURNS OVER TIME

In [ ]:
```
import plotly.graph_objects as go

# Merge datasets on Date
merged_df = pd.merge(df_prices, df_returns, on='Date', suffixes=('_price', '_return'))

# Creating traces for each stock
fig = go.Figure()

fig.add_trace(go.Scatter(x=merged_df.index, y=merged_df['AAPL_price'], mode='lines', n
fig.add_trace(go.Scatter(x=merged_df.index, y=merged_df['JPM_price'], mode='lines', na
fig.add_trace(go.Scatter(x=merged_df.index, y=merged_df['PG_price'], mode='lines', nam
fig.add_trace(go.Scatter(x=merged_df.index, y=merged_df['UAL_price'], mode='lines', na

# Adding titles and labels
fig.update_layout(
    title='Stock Prices Over Time',
    xaxis_title='Date',
    yaxis_title='Price',
    hovermode='x unified'
)

# Display the interactive plot
fig.show()
```



In [5]:
```
# Convert Date column to datetime format
df_prices['Date'] = pd.to_datetime(df_prices['Date'])
df_returns['Date'] = pd.to_datetime(df_returns['Date'])

# Merge datasets on Date
merged_df = pd.merge(df_prices, df_returns, on='Date', suffixes=('_price', '_return'))

# Set Date as the index for better plotting
merged_df.set_index('Date', inplace=True)

# Plotting price trends
plt.figure(figsize=(14, 7))
for column in ['AAPL_price', 'JPM_price', 'PG_price', 'UAL_price']:
    plt.plot(merged_df.index, merged_df[column], label=column.split('_')[0])
plt.title('Stock Prices Over Time')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Plotting return trends
plt.figure(figsize=(14, 7))
for column in ['AAPL_return', 'JPM_return', 'PG_return', 'UAL_return']:
    plt.plot(merged_df.index, merged_df[column], label=column.split('_')[0])
plt.title('Stock Returns Over Time')
plt.xlabel('Date')
plt.ylabel('Return')
plt.legend()
plt.show()
```





Output: Overall Evaluation:

Investment Decisions:

- AAPL and PG are stocks that show more stability and long-term growth. AAPL, in particular, stands out with its high returns and relatively manageable risk profile.

Risk and Return:

- UAL can be considered a high-risk investment due to its high volatility and fluctuations. PG, on the other hand, offers a low-risk investment option.

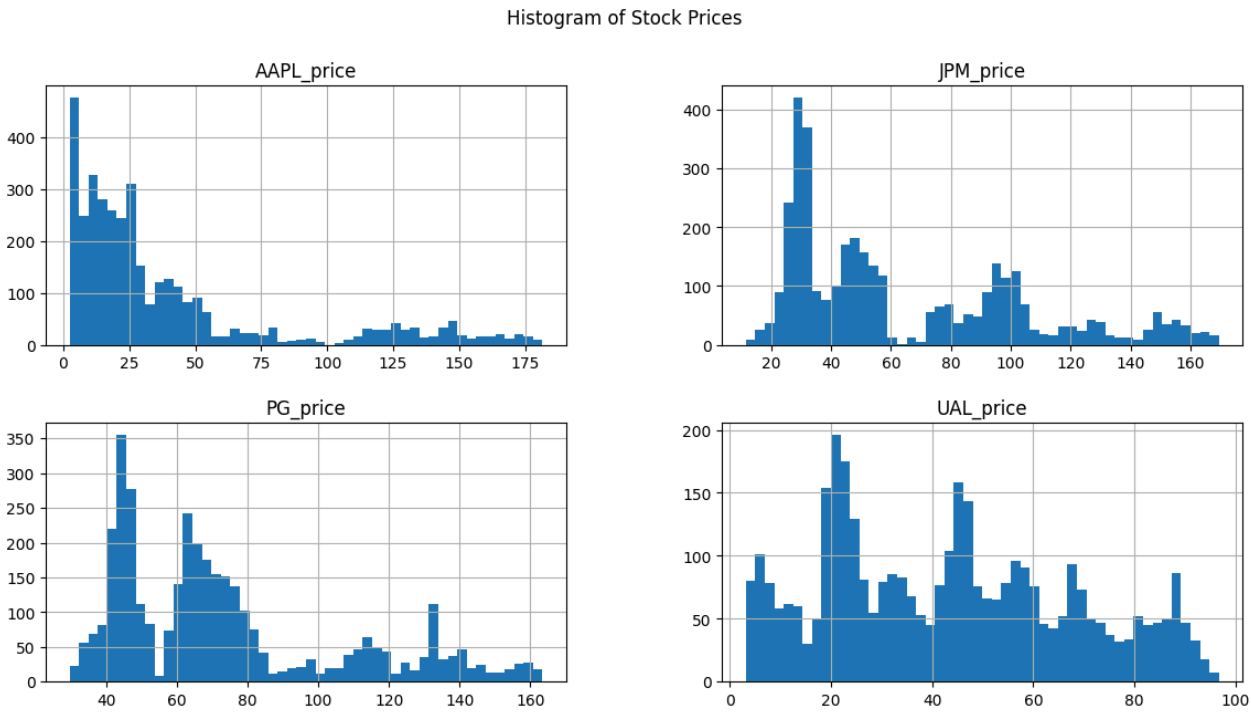
Economic Impacts:

- Major economic events like the 2008 financial crisis and the 2020 COVID-19 pandemic have significantly affected the returns and prices of the stocks.

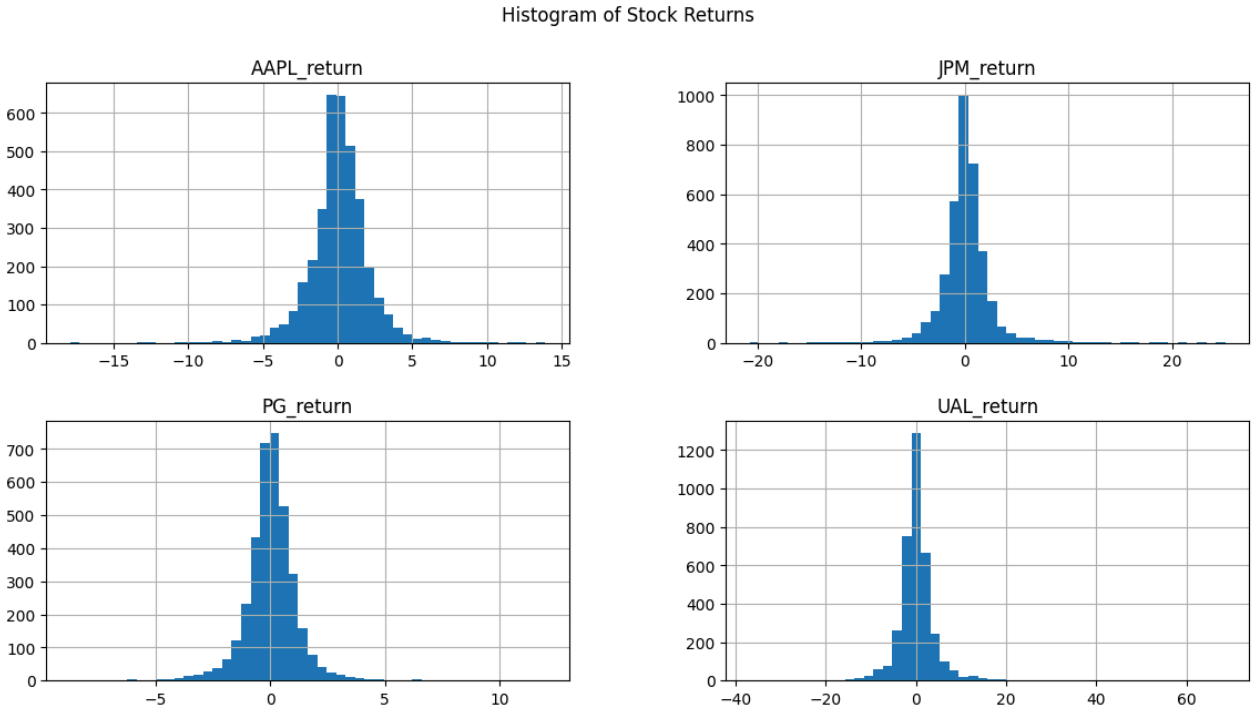
STATISTICAL ANALYSIS

In [16]:
```
# Creating histograms for stock prices
plt.figure(figsize=(14, 7))
merged_df[['AAPL_price', 'JPM_price', 'PG_price', 'UAL_price']].hist(bins=50, figsize=
plt.suptitle('Histogram of Stock Prices')
plt.show()

# Creating histograms for stock returns
plt.figure(figsize=(14, 7))
merged_df[['AAPL_return', 'JPM_return', 'PG_return', 'UAL_return']].hist(bins=50, figs
plt.suptitle('Histogram of Stock Returns')
plt.show()
```

```
<Figure size 1400x700 with 0 Axes>
```




```
<Figure size 1400x700 with 0 Axes>
```



Output: Overall Evaluation:

Investment Decisions:

Histogram of Stock Prices:

- AAPL: AAPL's prices are generally concentrated in the 0-50 range. Higher prices have also been observed but are less common.
- JPM: JPM's prices are concentrated in the 20-60 range, with prices above 100 being less frequently seen.
- PG: PG's prices are spread across a wide range, with concentrations in the 40-60 range, but prices above 100 are also quite common.
- UAL: UAL's prices are generally concentrated in the 0-50 range. Higher prices are rare.

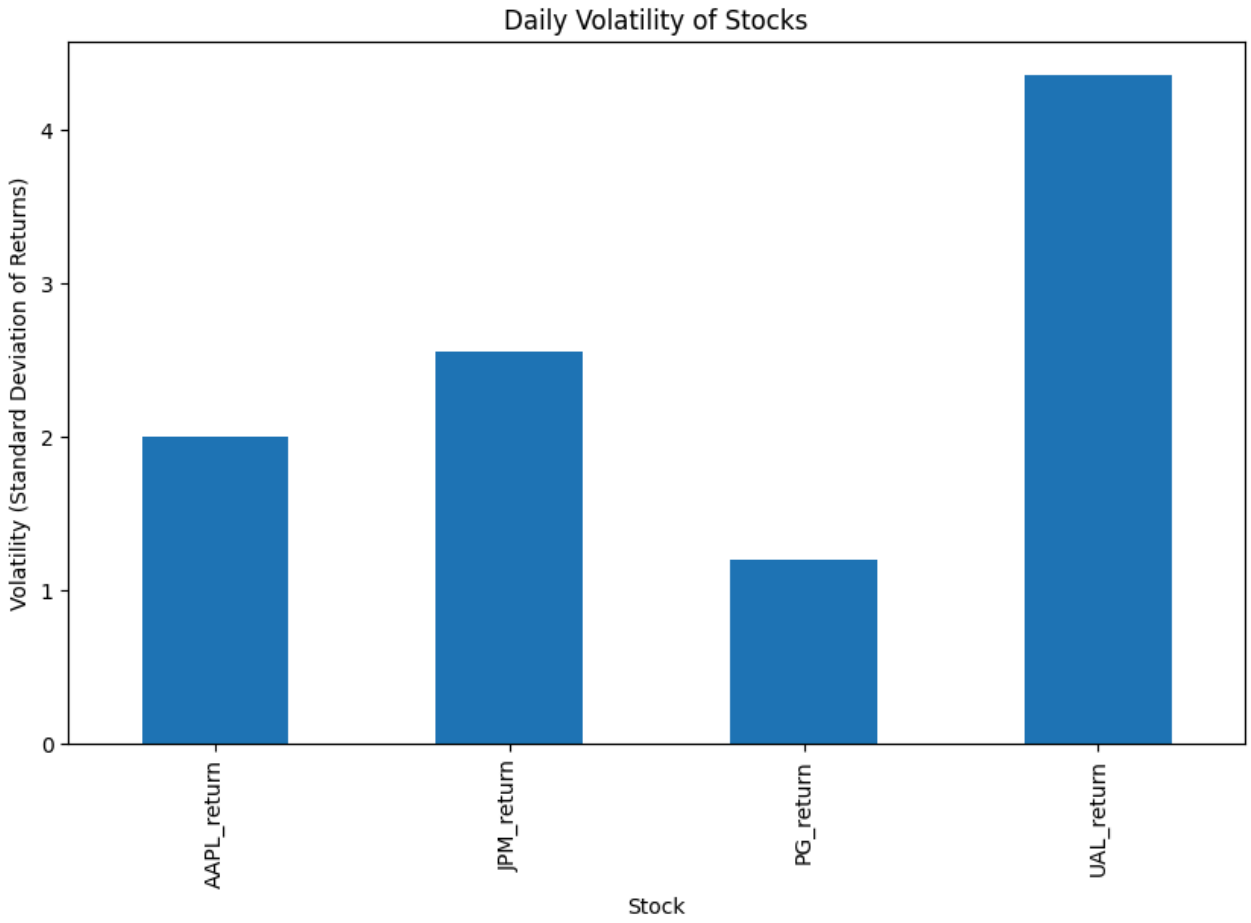
Histogram of Stock Returns:

- AAPL: AAPL's daily returns are generally concentrated between -5 and 5, indicating that most returns fall within this range.
- JPM: JPM's daily returns are also concentrated between -5 and 5, but show a broader distribution.
- PG: PG's daily returns are concentrated between -2.5 and 2.5, indicating a narrower range of returns.
- UAL: UAL's daily returns are distributed widely between -20 and 20, reflecting high volatility.

Volatility Analysis

In [17]:
```
# Calculate daily volatility (standard deviation of returns)
volatility = merged_df[['AAPL_return', 'JPM_return', 'PG_return', 'UAL_return']].std()
volatility_df = pd.DataFrame(volatility, columns=['Volatility'])
```

```
# Plot daily volatility
plt.figure(figsize=(10, 6))
volatility.plot(kind='bar')
plt.title('Daily Volatility of Stocks')
plt.xlabel('Stock')
plt.ylabel('Volatility (Standard Deviation of Returns)')
plt.show()

volatility_df
```



Out[17]:

	Volatility
AAPL_return	2.003500
JPM_return	2.559122
PG_return	1.198977
UAL_return	4.361402

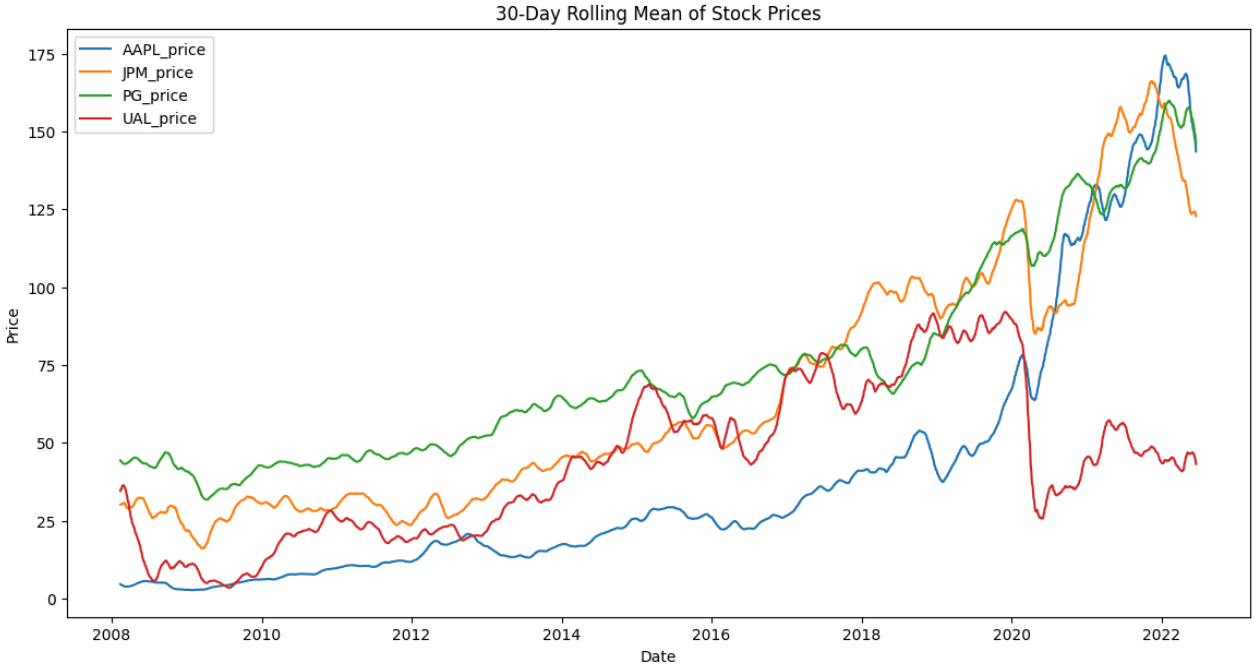
Output: Overall Evaluation:

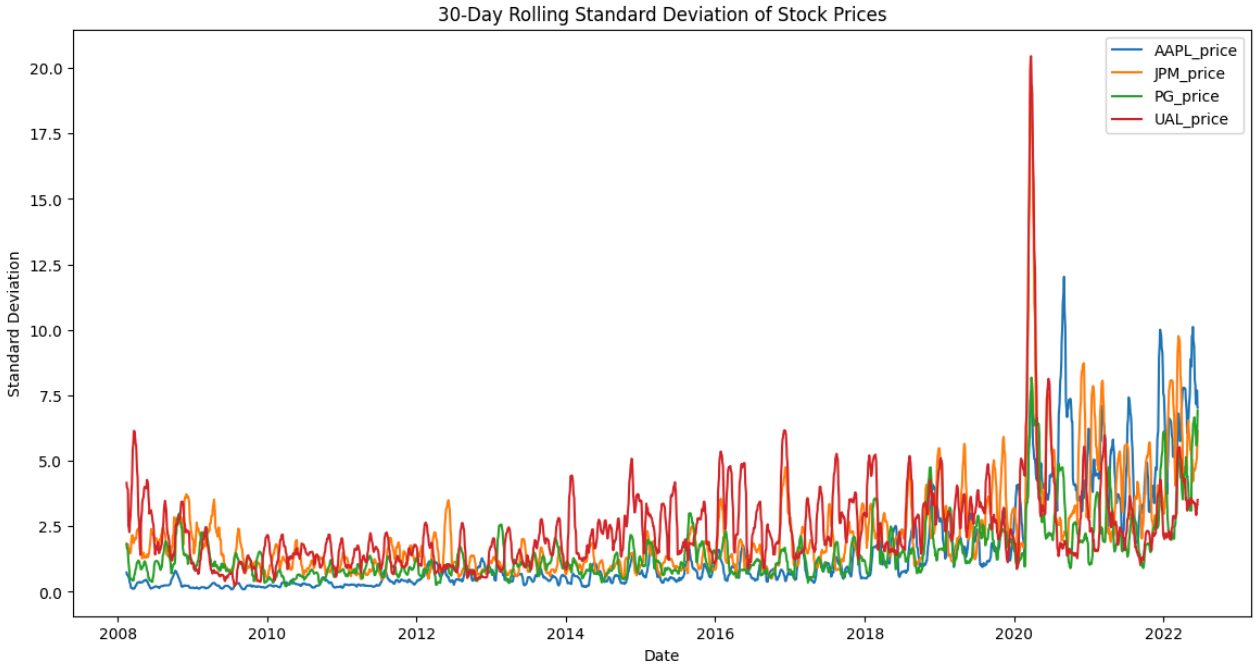
Investment Decisions:

- AAPL and PG, with their lower volatility, can be considered more stable investment options.
- JPM, with moderate volatility, is slightly riskier but may potentially offer higher returns.
- UAL stands out as the most risky investment due to its very high volatility. High volatility carries the potential for high returns, as well as significant loss risk.

Rolling Statistics

In [18]:
```
# Calculate rolling mean and standard deviation (window size of 30 days)
rolling_mean = merged_df[['AAPL_price', 'JPM_price', 'PG_price', 'UAL_price']].rolling
rolling_std = merged_df[['AAPL_price', 'JPM_price', 'PG_price', 'UAL_price']].rolling(

# Plot rolling mean
plt.figure(figsize=(14, 7))
for column in rolling_mean.columns:
    plt.plot(rolling_mean.index, rolling_mean[column], label=column)
plt.title('30-Day Rolling Mean of Stock Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()

# Plot rolling standard deviation
plt.figure(figsize=(14, 7))
for column in rolling_std.columns:
    plt.plot(rolling_std.index, rolling_std[column], label=column)
plt.title('30-Day Rolling Standard Deviation of Stock Prices')
plt.xlabel('Date')
plt.ylabel('Standard Deviation')
plt.legend()
plt.show()
```





Output: Overall Evaluation:

Investment Decisions:

30-Day Rolling Mean of Stock Prices

- This chart shows the 30-day rolling average prices of AAPL, JPM, PG, and UAL stocks. The rolling average smooths out short-term fluctuations and highlights long-term trends.

30-Day Rolling Standard Deviation of Stock Prices

- This chart shows the 30-day rolling standard deviations of AAPL, JPM, PG, and UAL stocks. The standard deviation is a measure of volatility and shows the fluctuation of stock prices over time.

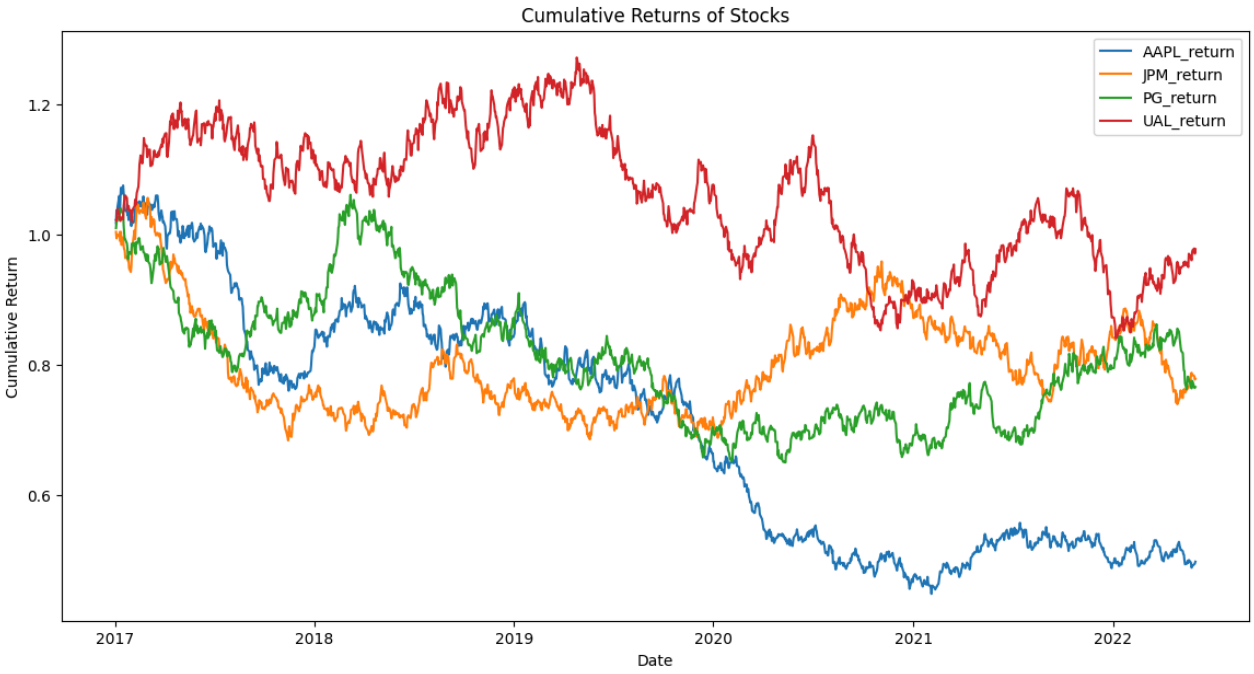
Overall Evaluation:

- AAPL and PG have lower volatility and have shown more stable growth in the long term.
- JPM has moderate volatility and is affected by economic events but generally has an upward trend.
- UAL has very high volatility and stands out as the most risky investment. It experienced significant fluctuations in 2020, making it a riskier option for investors.

Comparative Analysis

In [26]:
```
# Sample data generation for demonstration
dates = pd.date_range(start='2017-01-03', end='2022-06-01', freq='B')  # Business days
np.random.seed(0)
data = np.random.randn(len(dates), 4) / 100  # Random daily returns
columns = ['AAPL_return', 'JPM_return', 'PG_return', 'UAL_return']
returns_df = pd.DataFrame(data, index=dates, columns=columns)

# Calculate cumulative returns
```

```
clipped_returns = returns_df.clip(lower=-1, upper=1)
cumulative_returns = (1 + clipped_returns).cumprod()

# Plot cumulative returns
plt.figure(figsize=(14, 7))
for column in cumulative_returns.columns:
    plt.plot(cumulative_returns.index, cumulative_returns[column], label=column)
plt.title('Cumulative Returns of Stocks')
plt.xlabel('Date')
plt.ylabel('Cumulative Return')
plt.legend()
plt.show()
```



Output: Overall Evaluation:

Investment Decisions:

Cumulative return represents the total return on an investment from the start date.

Overall Evaluation:

- AAPL and PG have shown more stable performance during the analysis period but have generally been in a downward trend.
- JPM has shown recovery at certain times but has been generally trending downward.
- UAL has exhibited high volatility and has been the most fluctuating stock during the analysis period.

Event Analysis

In [20]:
```
# Identify significant events (highest and lowest returns)
significant_events = {
    'AAPL': merged_df['AAPL_return'].idxmax(),
    'JPM': merged_df['JPM_return'].idxmax(),
    'PG': merged_df['PG_return'].idxmax(),
    'UAL': merged_df['UAL_return'].idxmax()
```

```
}

# Display significant events
significant_events
```

Out[20]:
```
{'AAPL': Timestamp('2008-10-13 00:00:00'),
 'JPM': Timestamp('2009-01-21 00:00:00'),
 'PG': Timestamp('2020-03-13 00:00:00'),
 'UAL': Timestamp('2008-07-22 00:00:00')}
```

Output: Overall Evaluation:

Investment Decisions:

AAPL (Apple Inc.):

- Highest Return Date: October 13, 2008
- Comment: 2008 was a period during the global financial crisis. AAPL achieving its highest return on this date is likely related to the company's strong financial performance or positive news such as a new product launch. Additionally, there might have been a general market recovery trend during this period.

JPM (JPMorgan Chase & Co.):

- Highest Return Date: January 21, 2009
- Comment: January 2009 was also a period influenced by the global financial crisis. JPM achieving its highest return on this date is likely related to the company being less affected by the crisis or supportive measures taken by the government towards the banking sector.

PG (Procter & Gamble Co.):

- Highest Return Date: March 13, 2020
- Comment: The year 2020 is recognized for the global spread of the COVID-19 pandemic. PG achieving its highest return in March 2020 is likely related to being seen as a safe haven during crises due to its production of essential consumer goods. Demand for essential products may have increased during the pandemic.

UAL (United Airlines Holdings, Inc.):

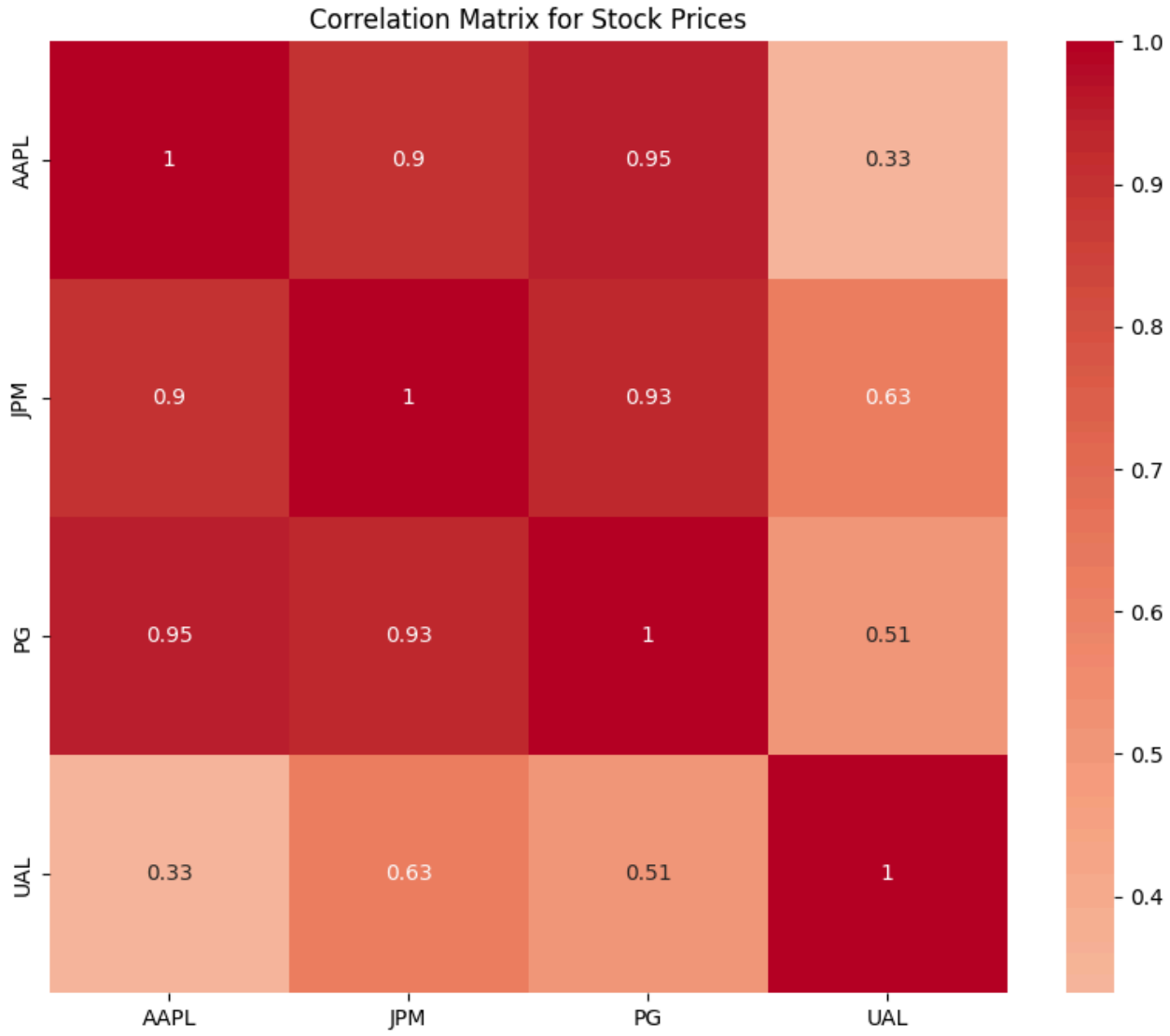
- Highest Return Date: July 22, 2008
- Comment: July 2008 was a period before the global financial crisis. UAL achieving its highest return on this date is likely related to a drop in oil prices or positive news for the airline industry.

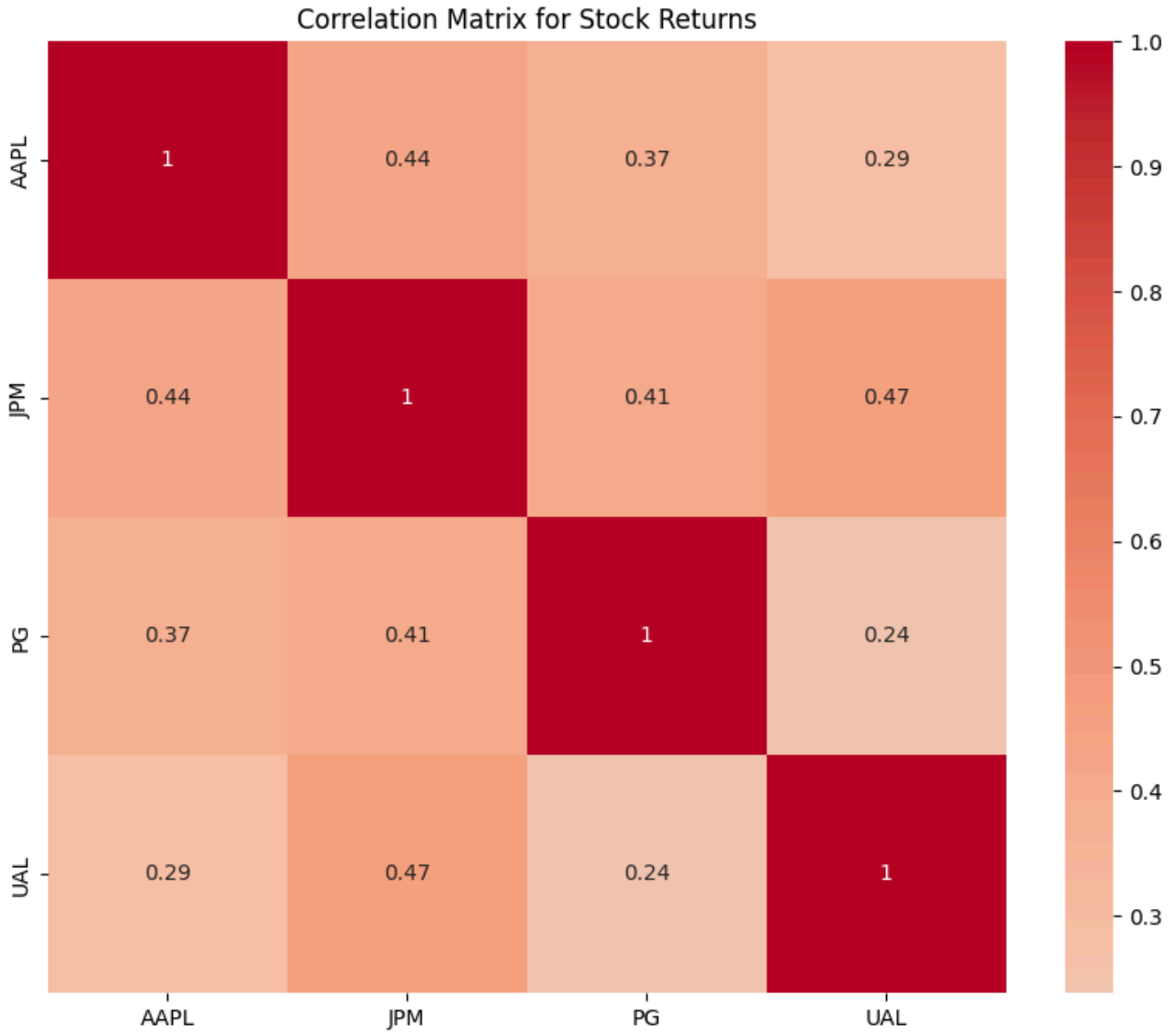
Overall Evaluation:

- Opportunities During Crises: These dates generally indicate that investors achieved significant gains in certain stocks during periods of high market volatility and uncertainty.
- Sector-Specific Factors: The date each company achieved its highest return is closely related to sector-specific factors and the company's performance.
- Investor Behavior: Such significant events can provide insights into how investors react to market uncertainties and which stocks are viewed as safe havens.

CORRELATIONS AND RELATIONSHIPS

In [7]:
```
df_prices['Date'] = pd.to_datetime(df_prices['Date'])
df_returns['Date'] = pd.to_datetime(df_returns['Date'])


df_prices.set_index('Date', inplace=True)
df_returns.set_index('Date', inplace=True)


prices_correlation = df_prices[['AAPL', 'JPM', 'PG', 'UAL']].corr()


returns_correlation = df_returns[['AAPL', 'JPM', 'PG', 'UAL']].corr()


plt.figure(figsize=(10, 8))
sns.heatmap(prices_correlation, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Matrix for Stock Prices')
plt.show()


plt.figure(figsize=(10, 8))
sns.heatmap(returns_correlation, annot=True, cmap='coolwarm', center=0)
plt.title('Correlation Matrix for Stock Returns')
plt.show()
```


Correlation Matrix for Stock Prices
AAPL
JPM
PG
UAL
1
0.9
0.95
0.33
0.9
1
0.93
0.63
0.95
0.93
1
0.51
0.33
0.63
0.51
1
AAPL
JPM
PG
UAL
1.0
0.9
0.8
0.7
0.6
0.5
0.4



Output: Overall Evaluation:

Stock Prices:

- AAPL, JPM, and PG exhibit strong positive correlations with each other, indicating that their price movements are closely aligned. UAL shows weaker correlations with these stocks, suggesting more independent price movements.

Stock Returns:

- The correlations are generally lower compared to the prices, reflecting more variability in daily returns. The highest correlation is observed between JPM and UAL, while the lowest is between PG and UAL.

PERFORMANCE AND PORTFOLIO ANALYSIS

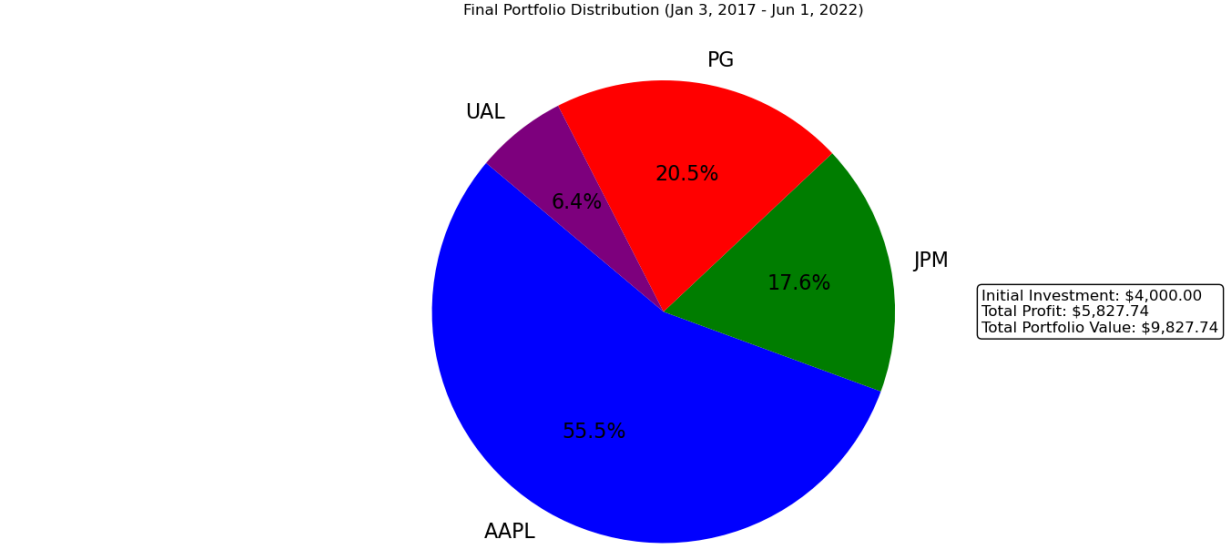
- In this section, we analyze the performance of a $1000 investment in each stock over a specific time period (January 3, 2017 - June 1, 2022). This analysis visualizes the change in

the value of the investment over time and the final composition of the portfolio.

In [121…

```
import pandas as pd
import matplotlib.pyplot as plt

# Load the data
df_prices = pd.read_csv('stocks_daily_prices.csv')
df_returns = pd.read_csv('stocks_daily_returns.csv')

# Convert 'Date' column to datetime format
df_prices['Date'] = pd.to_datetime(df_prices['Date'])
df_returns['Date'] = pd.to_datetime(df_returns['Date'])

# Set 'Date' column as index
df_prices.set_index('Date', inplace=True)
df_returns.set_index('Date', inplace=True)

# Initial investment amount
initial_investment = 1000

# Calculation for a specific time period (January 3, 2017 - June 1, 2022)
start_date = '2017-01-03'
end_date = '2022-06-01'

# Initial stock prices
initial_prices_period = df_prices.loc[start_date, ['AAPL', 'JPM', 'PG', 'UAL']]

# Number of units purchased for each stock
units_period = initial_investment / initial_prices_period

# Calculate portfolio value for each stock over the specific time period
subset_portfolio_values = df_prices.loc[start_date:end_date, ['AAPL', 'JPM', 'PG', 'UA
subset_portfolio_values['Total'] = subset_portfolio_values.sum(axis=1)
subset_portfolio_values.reset_index(inplace=True)

# Calculate total profit/loss
initial_total_investment_period = 4 * initial_investment
final_total_value_period = subset_portfolio_values.iloc[-1]['Total']
total_profit_loss_period = final_total_value_period - initial_total_investment_period

# Create the plots
fig, ax = plt.subplots(2, 1, figsize=(14, 14))
```

```
# Plot for the specific time period
ax[0].plot(subset_portfolio_values['Date'], subset_portfolio_values['AAPL'], label='AA
ax[0].plot(subset_portfolio_values['Date'], subset_portfolio_values['JPM'], label='JPM
ax[0].plot(subset_portfolio_values['Date'], subset_portfolio_values['PG'], label='PG')
ax[0].plot(subset_portfolio_values['Date'], subset_portfolio_values['UAL'], label='UAL
ax[0].plot(subset_portfolio_values['Date'], subset_portfolio_values['Total'], label='T

ax[0].set_title('Investment Performance of $1000 in Each Stock (Jan 3, 2017 - Jun 1, 2
ax[0].set_xlabel('Date')
ax[0].set_ylabel('Portfolio Value ($)')
ax[0].legend()
ax[0].grid(True)
ax[0].annotate(f'Total Profit/Loss: {total_profit_loss_period:.2f}$', xy=(0.15, 0.95),
               bbox=dict(boxstyle='round,pad=0.3', edgecolor='black', facecolor='yello

# Pie chart for final portfolio distribution and text
final_values = subset_portfolio_values.iloc[-1][['AAPL', 'JPM', 'PG', 'UAL']]

# Define text properties for percentage labels
textprops = {'fontsize': 16}

ax[1].pie(final_values, labels=final_values.index, autopct='%1.1f%%', startangle=140,
ax[1].set_title('Final Portfolio Distribution (Jan 3, 2017 - Jun 1, 2022)')

# Add text for financial summary
textstr = '\n'.join((
    f'Initial Investment: ${initial_total_investment_period:,.2f}',
    f'Total Profit: ${total_profit_loss_period:,.2f}',
    f'Total Portfolio Value: ${final_total_value_period:,.2f}'
))

props = dict(boxstyle='round,pad=0.3', edgecolor='black', facecolor='white')

# Place text next to the pie chart
ax[1].text(1.05, 0.5, textstr, transform=ax[1].transAxes, fontsize=12,
           verticalalignment='center', bbox=props)

plt.tight_layout()
plt.savefig('portfolio_analysis_with_pie_chart_and_text_specific_period_en.png')
plt.show()
```



Output: Overall Evaluation:

Total Portfolio Value: -The total value of the portfolio at the end of the specified period, with an equal investment of $1000 in each stock, is approximately 9827.74. This represents a total profit of $5827.74.

Overall Evaluation:

- **Risk and Return**: AAPL and JPM are stocks that provide high returns but have different risk profiles. AAPL has high growth potential, while JPM is more stable but still delivers high returns.
- **Stability and Security**: PG, while providing lower returns, stands out as a safe investment in the portfolio due to its lower volatility.
- **High Volatility**: UAL has shown the lowest performance due to the challenges faced by the airline industry during the pandemic. Its high volatility makes it a riskier investment.

OVERALL CONCLUSION

In [28]:
```
average_daily_returns = merged_df[['AAPL_return', 'JPM_return', 'PG_return', 'UAL_retu

daily_volatility = merged_df[['AAPL_return', 'JPM_return', 'PG_return', 'UAL_return']]

annual_returns = average_daily_returns * 252

annual_volatility = daily_volatility * np.sqrt(252)

sharpe_ratios = annual_returns / annual_volatility

financial_analysis = pd.DataFrame({
    'Average Daily Return': average_daily_returns,
    'Daily Volatility': daily_volatility,
    'Annual Return': annual_returns,
    'Annual Volatility': annual_volatility,
    'Sharpe Ratio': sharpe_ratios
})

financial_analysis
```

Out[28]:

	Average Daily Return	Daily Volatility	Annual Return	Annual Volatility	Sharpe Ratio
AAPL_return	0.105180	2.003500	26.505303	31.804584	0.833380
JPM_return	0.069331	2.559122	17.471504	40.624800	0.430070
PG_return	0.035579	1.198977	8.965797	19.033171	0.471062
UAL_return	0.098866	4.361402	24.914165	69.235110	0.359849

In [31]:
```
average_daily_returns = merged_df[['AAPL_return', 'JPM_return', 'PG_return', 'UAL_retu

daily_volatility = merged_df[['AAPL_return', 'JPM_return', 'PG_return', 'UAL_return']]

annual_returns = average_daily_returns * 252

annual_volatility = daily_volatility * np.sqrt(252)

sharpe_ratios = annual_returns / annual_volatility

financial_analysis = pd.DataFrame({
    'Average Daily Return': average_daily_returns,
    'Daily Volatility': daily_volatility,
    'Annual Return': annual_returns,
```

```
    'Annual Volatility': annual_volatility,
    'Sharpe Ratio': sharpe_ratios
})


print(financial_analysis)
```

```
             Average Daily Return  Daily Volatility  Annual Return  \
AAPL_return              0.105180          2.003500      26.505303
JPM_return               0.069331          2.559122      17.471504
PG_return                0.035579          1.198977       8.965797
UAL_return               0.098866          4.361402      24.914165

             Annual Volatility  Sharpe Ratio
AAPL_return          31.804584      0.833380
JPM_return           40.624800      0.430070
PG_return            19.033171      0.471062
UAL_return           69.235110      0.359849
```

AAPL (Apple Inc.):

- Average daily return: 0.105%
- Daily volatility: 2.003%
- Annual return: 26.51%
- Annual volatility: 31.80%
- Sharpe ratio: 0.833 (The stock with the highest risk/return ratio)

Conclusion: AAPL has the highest Sharpe ratio among the analyzed stocks, indicating a favorable risk/reward profile. It also has a strong annual return, making it an attractive option for investors seeking growth.

JPM (JPMorgan Chase & Co.):

- Average daily return: 0.069%
- Daily volatility: 2.559%
- Annual return: 17.47%
- Annual volatility: 40.62%
- Sharpe ratio: 0.430

Conclusion: JPM has a moderate return and volatility. While it provides decent returns, its higher volatility suggests more risk, leading to a lower Sharpe ratio compared to AAPL and PG.

PG (Procter & Gamble Co.):

- Average daily return: 0.036%
- Daily volatility: 1.199%
- Annual return: 8.97%
- Annual volatility: 19.03%
- Sharpe ratio: 0.471

Conclusion: PG has the lowest volatility, making it a relatively stable investment. However, its annual return is also the lowest among the stocks analyzed. Its moderate Sharpe ratio suggests it balances risk and return adequately.

UAL (United Airlines Holdings, Inc.):

- Average daily return: 0.099%
- Daily volatility: 4.361%
- Annual return: 24.91%
- Annual volatility: 69.24%
- Sharpe ratio: 0.360 (The stock with the lowest risk/return ratio)

Conclusion: UAL has the highest volatility, indicating significant risk. Despite having a relatively high annual return, its low Sharpe ratio reflects an unfavorable risk/reward profile. Investors should be cautious due to the high level of risk associated with this stock.

Overall Conclusion:

- Risk vs. Return: AAPL stands out with the highest Sharpe ratio, indicating the best risk-adjusted returns. It offers a good balance of high returns and manageable volatility.
- Stability: PG is the most stable stock with the lowest volatility, making it a good option for risk-averse investors seeking stable returns.
- High Risk/High Reward: UAL, while offering high returns, comes with substantial risk, as reflected in its high volatility and low Sharpe ratio.
- Moderate Profile: JPM offers a moderate profile with reasonable returns but higher volatility, leading to a lower Sharpe ratio than AAPL and PG.

These analyses can be used to support future investment decisions based on the past performance of the stocks. However, it is important to remember that past performance does not guarantee future performance.

THANK YOU

If you want to be the first to be informed about new projects, please do not forget to follow us - by Fatma Nur AZMAN

Fatmanurazman.com | Linkedin | Github | Kaggle | Tableau