FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
Willytom Tanguy Rigaud	France	willytomrig@gmail.com	
Fengyuexin Huang	Canada	fengyuexinhuang@gmail.com	
Babatunde Falola	USA	bfalola1@binghamton.edu	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

Team member 1	**Willytom Rigaud**
Team member 2	**Fengyuexin Huang**
Team member 3	**Babatunde Falola**

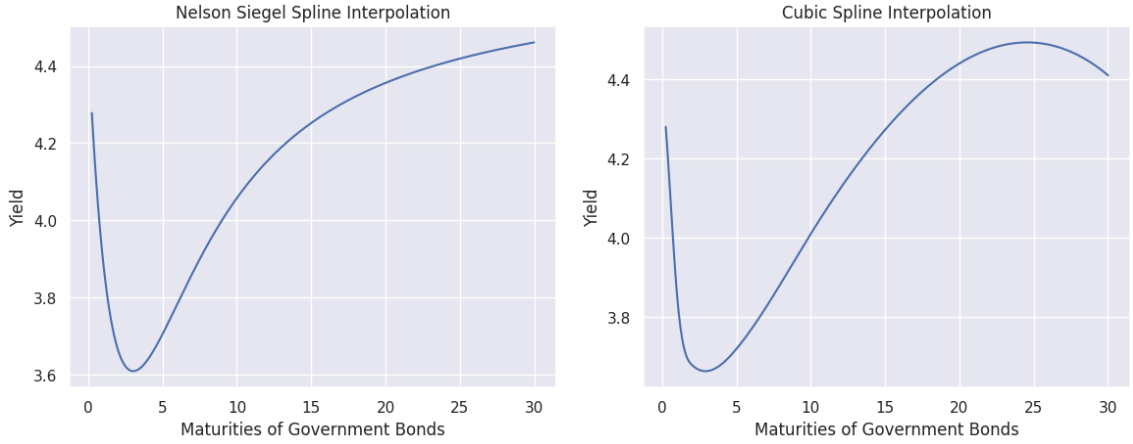
Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
Note: You may be required to provide proof of your outreach to non-contributing members upon request.

N/A

1. **Data quality**
   a. Regarding poorly unstructured financial data, I believe it is hard to come across because any published dataset is done by professional data scientists who would not let poor-quality data be published. Therefore, I tried to find data that would, though not poor quality, require some transformations prior to using them. I have seen the global dataset downloadable from: https://data.worldbank.org/indicator/CM.MKT.TRAD.GD.ZS?locations=FR,
   b. I would say that this is suboptimal quality because, as we have seen in lesson 1 of this module, rows should reflect observations time and columns variables. In this case, however, the columns are the different years, and the rows are the different countries. Furthermore, the data is quite incomplete. Some countries having data from previous years can lack observations, making for
   c. This earning call transcript from Dollar Tree is an example of unstructured poor quality unstructured data: https://www.fool.com/earnings/call-transcripts/2025/03/26/dollar-tree-dltr-q4-2024-earnings-call transcript/
   d. This is poor quality because the format is not consistent: if we were to scrape the Motley Fool website for all of the earnings call transcripts they provide, we would struggle because, given how it's produced from a speech, each company is free to shape the speech in whatever way they like. There are lots of numbers and percentages thrown around, and it would be complex to use them without an advanced natural language processing algorithm.

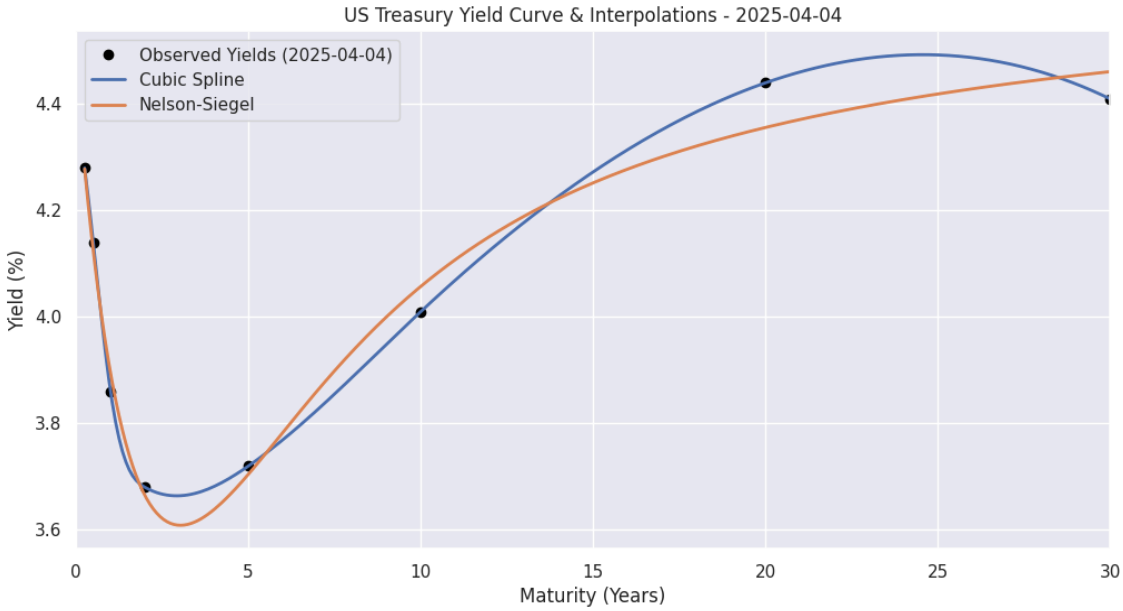
2. **Yield Curve Modeling**

   a. We are using government Bonds from the US government because Babatunde Falola is living there.
   b. We have picked the following maturities to get an accurate depiction of the yield curve: 3 Months, 6 Months, 1 Year, 2 Years, 5 Years, 10 Years, 20 Years, 30 Years.
   c. We fit the Nelson Siegel and Cubic Spline Models in the Jupyter Notebook and get these graph:



   d. See above.

e. To compare the fit of both models, we plot them on the same graph, along with the discrete yields from our original dataset :



We can see that the Cubic Spline Model fits every observable data point perfectly, which is not the case of the Nelson Siegel Model. This is due to the essence of the interpolation in the Cubic Spline mβodel. Since we are solving equations with said points as references, the curve will always pass through those. The models are comparable for the short-end of the curve : up to 2 years maturity, but we can then see the curves diverge. This could either be a good thing or a bad thing. A perfect fit locally could mean that the cubic spline model has a tendency to overfit and therefore miss the bigger picture. The Nelson siegel model is therefore more stable.

f. The model parameters for Nelson Spiegel are found by using print(curve) in the Jupyter Notebook : β_0 =4.67 describes the level of the yield curve, which is the value when t goes to infinity, β_1 = -0.2 is the short-term component of the yield curve, which would basically be the short term slope, β_2 = -3.238 reflects the magnitude of the medium term dip we see in the graph, around 4-5 years, τ = 1.83.

g. Smoothing is unethical when it is presented as true data to market performance. By smoothing, and therefore reducing volatility, we increase the risk adjusted return of a fund or security for example. However, this is not the purpose of the Nelson Siegel model. It is clearly presented as an approximation, and the parameters make it transparent to understand. Good usage of the Nelson Siegel model by staying within its intended use, and not using it to hide interest rate volatility is ethical. Crafting strategies on interest rates using it with omitting the stochastic nature of volatility would be unethical.

Q3. Exploiting Correlation

Financial Data is meant not only to process data but to understand how meaningful factors can be used to summarize or represent the data.

a. Generate 5 uncorrelated Gaussian random variables that simulate yield changes (they can be positive or negative with a mean close to 0 and a standard deviation that is small).

Code:

```
np.random.seed(42)

num_samples = 100

mean = 0

std_dev = 0.05

uncorrelated_data = np.random.normal(mean, std_dev, size=(num_samples, 5))

uncorrelated_df = pd.DataFrame(uncorrelated_data, columns=[f'Yield_{i+1}' for i in range(5)])
```

b. Run a Principal Components using EITHER the correlation OR covariance matrix.

Code:

```
# Step b: Perform PCA using the covariance matrix

pca = PCA()

pca.fit(uncorrelated_df)

# Explained variance ratio

explained_variance_ratio = pca.explained_variance_ratio_
```

c. Write a paragraph explaining how the variances of each component compare with each other. In this paragraph, you will address the following question: how much variance is explained by Component 1, Component 2, and Component 3?

Component	**Proportion of Variance**	**Percentage Explained**
Component 1	0.2626	26.26
Component 2	0.2158	21.58
Component 3	0.2022	20.22
Component 4	0.1813	18.13

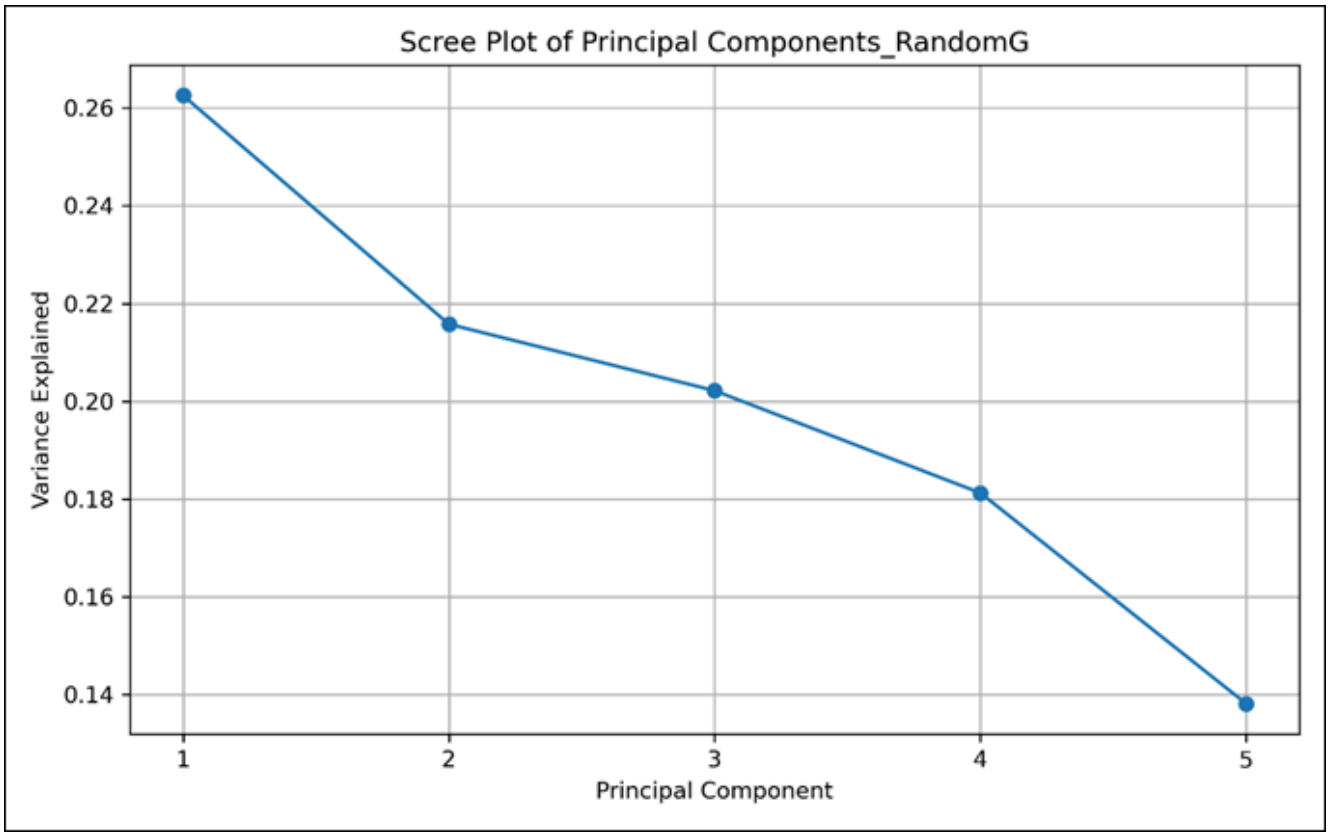
The first task was to generate the 5 uncorrelated Gaussian random variables and then applied Principal Component Analysis (PCA) using the covariance matrix. Principal Component Analysis (PCA) has

evolved significantly and is widely applied in dimensionality reduction and data exploration across scientific disciplines (Jolliffe & Cadima, 2021). We analyzed how much variance each principal component explains, as shown in Table 1 above: **Component 1** explains approximately **26.26%** of the total variance. **Component 2** explains about **21.58%,** and **Component 3** accounts for **20.22%** of the variance. The distribution of the total variance by the principal components are relatively even since the original variables are uncorrelated and have similar variances

d. Produce a screeplot (see https://en.wikipedia.org/wiki/Scree_plot) of the variance explained for each component.

The scree plot illustrates how much variance is captured by each principal component (Abdi and Williams, 2010). In this case, the plot shows a gradual and steady decrease in the variance explained by each successive component rather than a steep decline after the first one. This pattern indicates that no single component largely dominates the data structure.

Statistically, if there had been a sharp drop-off after the first component, it would suggest that most of the variability in the data can be captured by just one dominant factor. However, the gradual decline indicates that the variance is more evenly distributed across components, meaning each component contributes a significant, though smaller, amount of unique information. This kind of distribution is typical for uncorrelated or synthetic datasets, where no strong underlying relationship ties the variables together, as opposed to real-world financial data, which often has highly correlated features leading to one dominant principal component.



Now let's work with real data:

e. Collect the daily closing yields for 5 government securities, say over 6 months.: I collected for 13 months. See file attached.

Data Source: U.S. Department of the Treasury. *Interest Rate Statistics - CSV Archive*. U.S. Department of the Treasury, https://home.treasury.gov/interest-rates-data-csv-archive. Accessed 13 Apr. 2025.

f. Be sure to compute the daily yield changes!

```
# Step 1: Load Excel File

file_path = "par-real-yield-curve-rates-2003-2024.xlsx"

df = pd.read_excel(file_path, sheet_name='par-real-yield-curve-rates-2003')

# Step 2: Convert 'Date' column to datetime

df['Date'] = pd.to_datetime(df['Date'])

# Step 3: Filter by date range

start_date = "2023-10-01"

end_date = "2024-10-30"

df_filtered = df[(df['Date'] >= start_date) & (df['Date'] <= end_date)].sort_values('Date')

# Step 4: Select relevant maturity columns

selected_columns = ['5 YR', '7 YR', '10 YR', '20 YR', '30 YR']

df_yields = df_filtered[selected_columns]

# Step 5: Compute daily yield changes

df_yield_changes = df_yields.diff().dropna()
```

g. Re-run the Principal Components using EITHER the correlation or covariance matrix.

```
# Step 6: Perform PCA using the covariance matrix

pca = PCA()

pca.fit(df_yield_changes)

explained_variance_ratio = pca.explained_variance_ratio_
```

```
# Step 7: Print explained variance for each component

print("Explained Variance Ratio by Component:")

for i, ratio in enumerate(explained_variance_ratio, 1):

    print(f"Component {i}: {ratio:.2%}")
```

h. How do the variances of each component compare? In other words, how much variance is explained by Component 1, Component 2, Component 3, etc.?

PCA Analysis of Real Yield Changes (10/01/2023 – 10/30/2024)

After processing the daily yield changes for 5 government securities (5YR, 7YR, 10YR, 20YR, 30YR), we performed Principal Component Analysis (PCA) using the covariance matrix (U.S. Department of the Treasury, 2025). Here's how the variance is explained by each principal component:

- **Component 1:** 92.88% of the total variance
- **Component 2:** 6.36%
- **Component 3:** 0.49%
- **Component 4:** 0.16%
- **Component 5:** 0.11%

This result shows that the first principal component alone explains nearly all the variance in the yield changes. This is typical in interest rate data, where movements across different maturities are highly correlated, often representing a parallel shift of the yield curve.

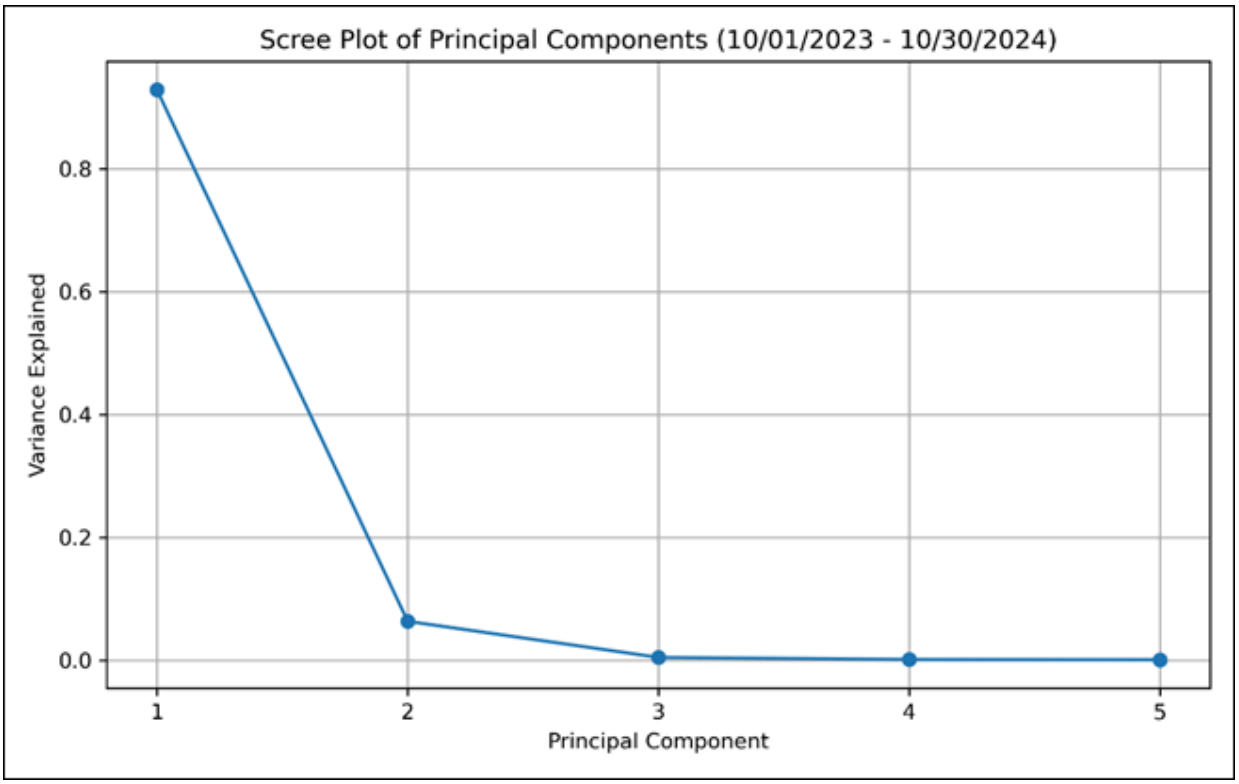
i. Produce a screeplot of the variance explained for each component.

```
# Step 8: Plot Scree Plot

plt.figure(figsize=(8, 5))

plt.plot(range(1, len(explained_variance_ratio) + 1), explained_variance_ratio, marker='o')

plt.title('Scree Plot of Principal Components (10/01/2023 - 10/30/2024)')

plt.xlabel('Principal Component')

plt.ylabel('Variance Explained')

plt.grid(True)

plt.xticks(range(1, len(explained_variance_ratio) + 1))
```

plt.tight_layout()

plt.savefig("my_plot2.png", dpi=300)

plt.show()



j. How does the screeplot from the uncorrelated data compare with the screeplot from the government data?

Scree Plot Comparison

Compared to the scree plot for uncorrelated synthetic data, where the variance was more evenly distributed across components, the real yield data shows a dominant Component 1, followed by a steep drop. This difference highlights:

- **Uncorrelated Data:** No dominant factor; variance spread nearly evenly.
- **Government Yield Data:** Strong shared trend captured in the first component; very little additional variation in others.

The scree plot derived from the PCA of U.S. Treasury yield changes between October 1, 2023, and October 30, 2024, reveals that the first principal component alone accounts for approximately 92.88% of the total variance. This dominant component reflects a strong common trend across yield changes for different maturities (5-year, 7-year, 10-year, 20-year, and 30-year bonds), indicating that movements in interest rates across the yield curve are highly correlated.

This is a significant finding because it confirms a well-established financial principle: yield changes across different maturities often shift together due to shared macroeconomic factors such as inflation expectations, monetary policy, or market sentiment. In other words, most of the variability in long-term yields can be explained by a single underlying factor, typically interpreted as a parallel shift in the yield curve (Diebold & Li, 2006). In contrast to synthetic or uncorrelated data, where variance is distributed more evenly across components, the real-world U.S. Treasury yield data displays a **sharp drop** after the first component in the scree plot. This validates the use of PCA in fixed-income analysis, where reducing dimensionality to a few components (often three: level, slope, and curvature) is both analytically sound and practically useful in forecasting, portfolio management, and risk analysis.

References:

1. U.S. Department of the Treasury. *Interest Rate Statistics - CSV Archive*. U.S. Department of the Treasury, https://home.treasury.gov/interest-rates-data-csv-archive. Accessed 13 Apr. 2025.
2. Jolliffe, I. T., and Jorge Cadima.**"Principal component analysis: a review and recent developments."** *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2191, 2021, https://doi.org/10.1098/rsta.2020.0161.
3. Litterman, Robert, and José Scheinkman.**"Common factors affecting bond returns."** *The Journal of Fixed Income*, vol. 1, no. 1, 1991, pp. 54–61.
4. Abdi, Hervé, and Lynne J. Williams. **"Principal component analysis."** *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, 2010, pp. 433–459. https://doi.org/10.1002/wics.101.
5. Diebold, Francis X., and Canlin Li. **"Forecasting the term structure of government bond yields."** *Journal of Econometrics*, vol. 130, no. 2, 2006, pp. 337–364. https://doi.org/10.1016/j.jeconom.2005.03.005.

4. Empirical Analysis of ETFs

a. Find the 30 Largest Holdings of the XLK ETF:

For the empirical analysis of the XLK ETF, which focuses on the technology sector, we identified the top 30 holdings using the financial data platform TipRanks. This platform provides detailed and up-to-date information on ETF compositions. The list of holdings was accessed directly from TipRanks, ensuring accuracy and reliability in our analysis. Below is the list of the 30 largest holdings in the XLK ETF:

1. Apple Inc. (AAPL)
2. Microsoft Corp. (MSFT)
3. NVIDIA Corp. (NVDA)
4. Broadcom Inc. (AVGO)
5. Salesforce Inc. (CRM)
6. Cisco Systems Inc. (CSCO)
7. Oracle Corp. (ORCL)
8. International Business Machines Corp. (IBM)
9. Palantir Technologies Inc. (PLTR)
10. Accenture Plc (ACN)
11. ServiceNow Inc. (NOW)

12. Intuit Inc. (INTU)
13. Adobe Inc. (ADBE)
14. Qualcomm Inc. (QCOM)
15. Advanced Micro Devices Inc. (AMD)
16. Texas Instruments Inc. (TXN)
17. Applied Materials Inc. (AMAT)
18. Palo Alto Networks Inc. (PANW)
19. Analog Devices Inc. (ADI)
20. KLA Corp. (KLAC)
21. CrowdStrike Holdings Inc. (CRWD)
22. Intel Corp. (INTC)
23. Lam Research Corp. (LRCX)
24. Micron Technology Inc. (MU)
25. Amphenol Corp. (APH)
26. Arista Networks Inc. (ANET)
27. Cadence Design Systems Inc. (CDNS)
28. Motorola Solutions Inc. (MSI)
29. Synopsys Inc. (SNPS)
30. Fortinet Inc. (FTNT)

b. Get at Least 6 Months of Data:

To perform a comprehensive analysis of the XLK ETF, we collected historical price data for its top 30 holdings over a one-year period, from January 1, 2023, to January 1, 2024. When we look at the XLK ETF, we can see how the daily returns of its top 30 holdings are interconnected. The data was sourced using the yfinance library in Python, which allows for efficient retrieval of financial data from Yahoo Finance. This approach ensures that the data is both accurate and up-to-date, forming a solid foundation for subsequent analyses.

c. Compute the Daily Returns:

Daily returns were calculated for the top 30 holdings of the XLK ETF by determining the percentage change in their closing prices from one day to the next. This analysis provides insights into the volatility and trends of the selected stocks, aiding in risk assessment and investment decision-making. By examining the daily returns, we can better understand the behavior of these stocks within the ETF. A sample visualization of holdings like Apple Inc. (AAPL), Microsoft Corp. (MSFT), and NVIDIA Corp. (NVDA) helps observe the volatility and performance of these stocks over time.

d. Compute the Covariance Matrix:

The covariance matrix is a statistical tool that measures how two variables move together. In the case of the XLK ETF, it emphasizes how the daily returns of its top 30 holdings connect with one another. It helps assess risk by indicating how stocks move in relation to each other. A positive covariance means stocks move in the same direction, while a negative covariance means they move in opposite directions. By understanding these relationships, investors can diversify their portfolios by selecting stocks that are less correlated, thereby reducing risk. The matrix is derived from the daily returns, with diagonal elements

representing the variance of each stock and off-diagonal elements representing the covariance between pairs of stocks.

e. Compute the PCA (Principal Component Analysis):

PCA is a technique used to reduce the dimensionality of a dataset while preserving as much variance as possible. For the XLK ETF, it helps identify the main factors driving the variability in the daily returns of the top 30 holdings. PCA involves finding the eigenvectors and eigenvalues of the covariance matrix of the returns. The eigenvectors represent the directions of maximum variance, and the eigenvalues indicate the magnitude of variance in those directions. PCA makes it easier to understand the dataset by emphasizing the key components that really matter. The first principal component (PC1) accounts for 43.78% of the variance, mainly driven by stocks like AMAT, KLAC, and SNPS.

f. Compute the SVD (Singular Value Decomposition):

SVD is a technique used to decompose a dataset into components that reveal its structure. For the XLK ETF, it helps analyze the daily returns data. SVD breaks down the data into simpler parts for easier analysis and dimensionality reduction, retaining essential information. Singular values indicate the importance of each component in capturing variance. SVD provides a comprehensive view of the data's structure, with singular values analogous to the eigenvalues in PCA, showing the importance of each component.

Detailed Analysis:

Daily returns are a fundamental aspect of financial analysis as they provide a snapshot of how an asset's price changes from one day to the next. This info is really important for getting a grip on how unstable the ETF’s holdings might be and how they perform.

By looking at the daily returns, investors can understand the risks that come with each stock, spot trends, and make smarter investment choices. Watching the daily ups and downs of big stocks like AAPL, MSFT, and NVDA can really show us when things are shaky and when they calm down. This info is super helpful for figuring out risks and planning our next moves.

The covariance matrix is another critical tool in this analysis. It measures how the daily returns of the top 30 holdings relate to each other, providing insights into the relationships between stocks. A positive covariance indicates that stocks move together, potentially increasing portfolio risk, while a negative covariance suggests they move in opposite directions, offering diversification benefits. Understanding these relationships helps investors optimize their portfolios by selecting less correlated stocks, thereby reducing risk.

Principal Component Analysis (PCA) is employed to reduce the dimensionality of the dataset while preserving as much variance as possible. PCA identifies the main factors driving variability in daily returns. When it comes to the XLK ETF, it emphasizes how the daily returns of its top 30 holdings are connected to one another. Eigenvectors represent the directions of maximum variance, while eigenvalues indicate the magnitude of variance in those directions. Positive and negative weights in PCA components show the direction of influence on the component. PCA helps break down the dataset by emphasizing the most important parts. This makes it much simpler to spot patterns and understand what really influences the ETF's performance.

Singular Value Decomposition (SVD) is another technique used to decompose the dataset into components that reveal its structure. SVD provides a comprehensive view of the data's structure, with singular values indicating the importance of each component. SVD is analogous to PCA but offers a more general decomposition, capturing the data's essential patterns and relationships. Singular values in SVD are similar to eigenvalues in PCA, showing the importance of each component. SVD breaks down the data into simpler parts for easier analysis and dimensionality reduction, retaining essential information.

While both PCA and SVD are used for dimensionality reduction, they serve different purposes. PCA focuses on variance, identifying the principal components that capture the most variance in the data. In contrast, SVD provides a more general decomposition, revealing the data's structure through singular values and vectors. PCA is a specific application of SVD when applied to the covariance matrix. Both techniques reveal that a few components capture most of the variance, as seen in the explained variance plots. This suggests that the dataset can be effectively reduced without losing important information.

The top PCA components reveal which stocks have the most impact on the ETF's returns. For example, PC2, with an explained variance of 7.05%, is heavily influenced by PANW and NOW, indicating their important role in the ETF's dynamics. Understanding these components helps identify key drivers and potential risks within the portfolio. By focusing on the most important components, we can better understand the underlying factors affecting the ETF and optimize strategies accordingly.

Overall, daily returns, PCA, and SVD collectively provide a comprehensive view of the ETF's performance. This means we can shrink the dataset without losing important details. This analysis not only aids in risk management but also enhances portfolio optimization by identifying key performance drivers. Using these statistical tools can help investors really get what’s going on with the ETF and come up with better investment strategies.