**GROUP WORK PROJECT # 1 M5** MScFE 600: FINANCIAL DATA
**GROUP NUMBER:** 10069

| FULL LEGAL NAME | LOCATION (COUNTRY) | EMAIL ADDRESS | MARK X FOR ANY NON-CONTRIBUTING MEMBER |
|---|---|---|---|
| ZULFIQAR ALI MIR | Pakistan | manager.equity.finance@gmail.com | |
| Darlington Okereke | Nigeria | darlingtonokereke@gmail.com | |
| Sunil Patra | | | X |

**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** | ZULFIQAR ALI MIR |
| **Team member 2** | Darlington Okereke |
| **Team member 3** | |

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.

We put messages in the Discussion Group, and I also tried multiple times to reach out on the WhatsApp Group, but there was no response from Sunil Patra till now on July 6, 2025.

Please check his/her attendance.

## Task 1 Data Quality

### Poor Quality Structure Data Example

An Excel spreadsheet that has daily exchange rates can be put an example in this case. For example, this file contains (IBM):

- Missing entries for certain dates,
- Data format inconsistencies like "01/07/25", "July 1, 2025", and "2025-07-01",
- Alphabetic characters in numeric fields

As this data is inconsistent and incomplete, it creates a problem. It creates parsing errors or is deleted from the analysis. Therefore, it leads to unreliable results and a high chance of wrong decision-making in financial analysis.

### Poor Quality Unstructured Data Example

Let's find tweets discussing bond yield: (Sanger and Warin)

- "Bond yields are cray rn 😳📉"
- "IDK why 10y treasury down lol"
- "buy $TLT if u know what's good 📈🔥"

Though these are unstructured data, they contain slang, abbreviations, and emojis.
As it does not follow a consistent structure, it is difficult to use it in analysis. The use of slang, abbreviations, and emojis needs to be interpreted. For such data, we have to use Natural Language Processing. Even then, accuracy is not certain, causing difficulty in decision-making for financial insights. (WQU Course Notes)

### References

- IBM. *Structured vs. Unstructured Data*. IBM Cloud Learn Hub, https://www.ibm.com/cloud/blog/structured-vs-unstructured-data July 5, 2025.
- Sanger, William, and Thierry Warin. "High Frequency and Unstructured Data in Finance: An Exploratory Study of Twitter." *Journal of Global Research in Computer Science*, vol. 7, no. 4, 2016, pp. 6–16, https://www.rroij.com/open-access/high-frequency-and-unstructured-data-in-finance-an-exploratory-study-oftwitter-.php?aid=70514.
- World Quant University. *MScFE 600 Course Materials and Lecture Notes*. 2025.

## Task 2: Yield Curve Modelling

### Collecting Yields Data

U.S. Treasury yields can be found from FRED via Fred API for maturities including 6-month, 1-year, 2-year, 3-year, 5-year, 7-year, 10-year, 20-year, 30-year, covering timeframe 2020-2025.

### Risk-Free Interest Rates

U.S. Treasury yields are taken as "risk-free" as the U.S. government is unlikely to default.

These rates are the benchmark for almost all asset pricing.

So these are also used as a baseline in Task 3, Yield Curve Modeling

### Credit Risk

If a borrower is unable to pay back the loan, then it is called Credit Risk. Whereas Treasuries are considered risk-free.

Extra yield, investors want is called credit spread.

Credit Spread = Corporate Bond Yield - Treasury Yield (same maturity)

Change in yield over time is called Volatility. It affects investment in bonds, financial risk and portfolio.

### Fitting the Nelson–Siegel Model

The Nelson-Siegel function is given below:

$$y(\tau) = \beta_0 + \beta_1 \cdot \backslash frac\{1 - e^{\{-\lambda \backslash tau\}}\}\{\lambda \backslash tau\} + \beta_2 \cdot \left(\backslash frac\{1 - e^{\{-\lambda \backslash tau\}}\}\{\lambda \backslash tau\} - e^{\{-\lambda \backslash tau\}}\right)$$

### FRED API for Yield Data

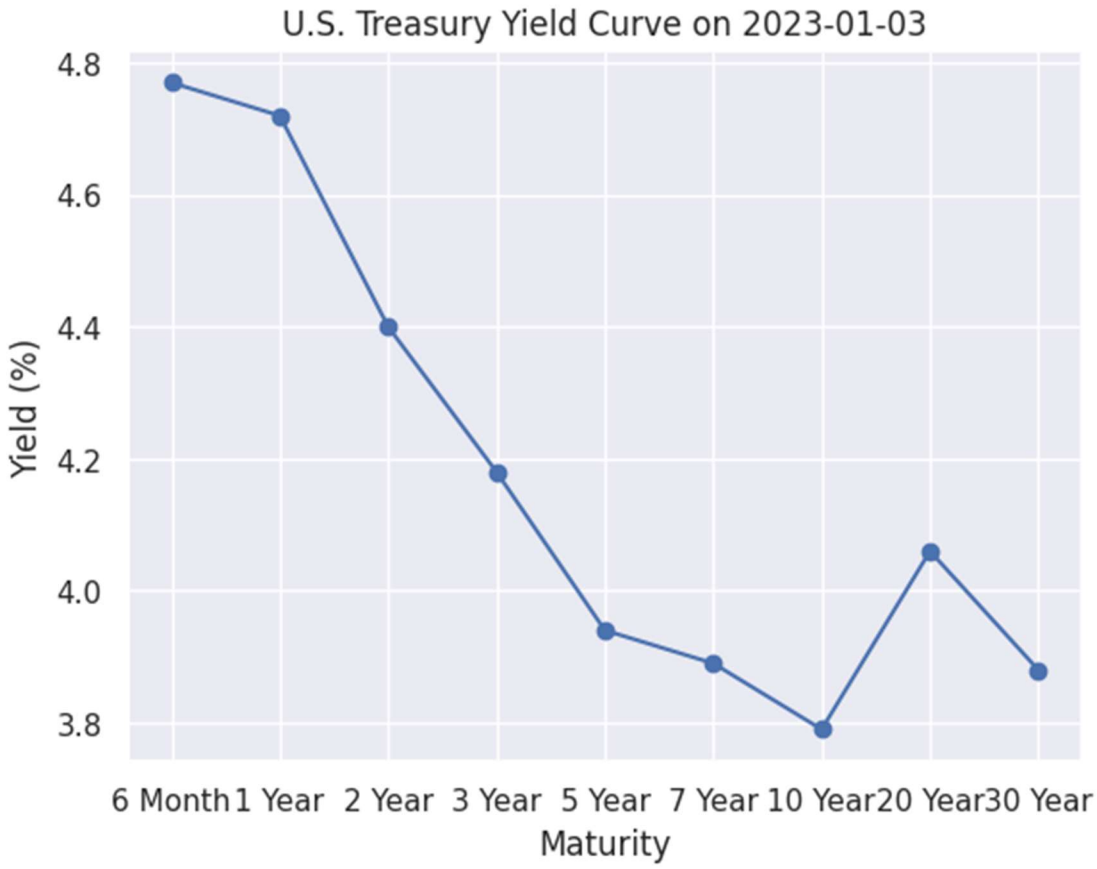
Using Python and FRED (Federal Reserve Economic Data) API, yield data has been pulled. It is real time government bond yield data for 9 different maturities from 6 months to 30 years (e.g., DGS6MO, DGS1, ..., DGS30). It covered time from January 1, 2020, to July 1, 2025.

Using this structured dataset, analyzation of yield curve was done using polynomial fitting and the Nelson Siegel model.

Top five rows of dataset are as follow:

| | 6 Month | 1 Year | 2 Year | 3 Year | 5 Year | 7 Year | 10 Year | 20 Year | 30 Year |
|---|---|---|---|---|---|---|---|---|---|
| **2020-01-01** | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| **2020-01-02** | 1.57 | 1.56 | 1.58 | 1.59 | 1.67 | 1.79 | 1.88 | 2.19 | 2.33 |
| **2020-01-03** | 1.55 | 1.55 | 1.53 | 1.54 | 1.59 | 1.71 | 1.80 | 2.11 | 2.26 |
| **2020-01-06** | 1.56 | 1.54 | 1.54 | 1.56 | 1.61 | 1.72 | 1.81 | 2.13 | 2.28 |
| **2020-01-07** | 1.56 | 1.53 | 1.54 | 1.55 | 1.62 | 1.74 | 1.83 | 2.16 | 2.31 |

The plot of sample yield curve is given below



Below is the line graph of all yield values of all maturities over time

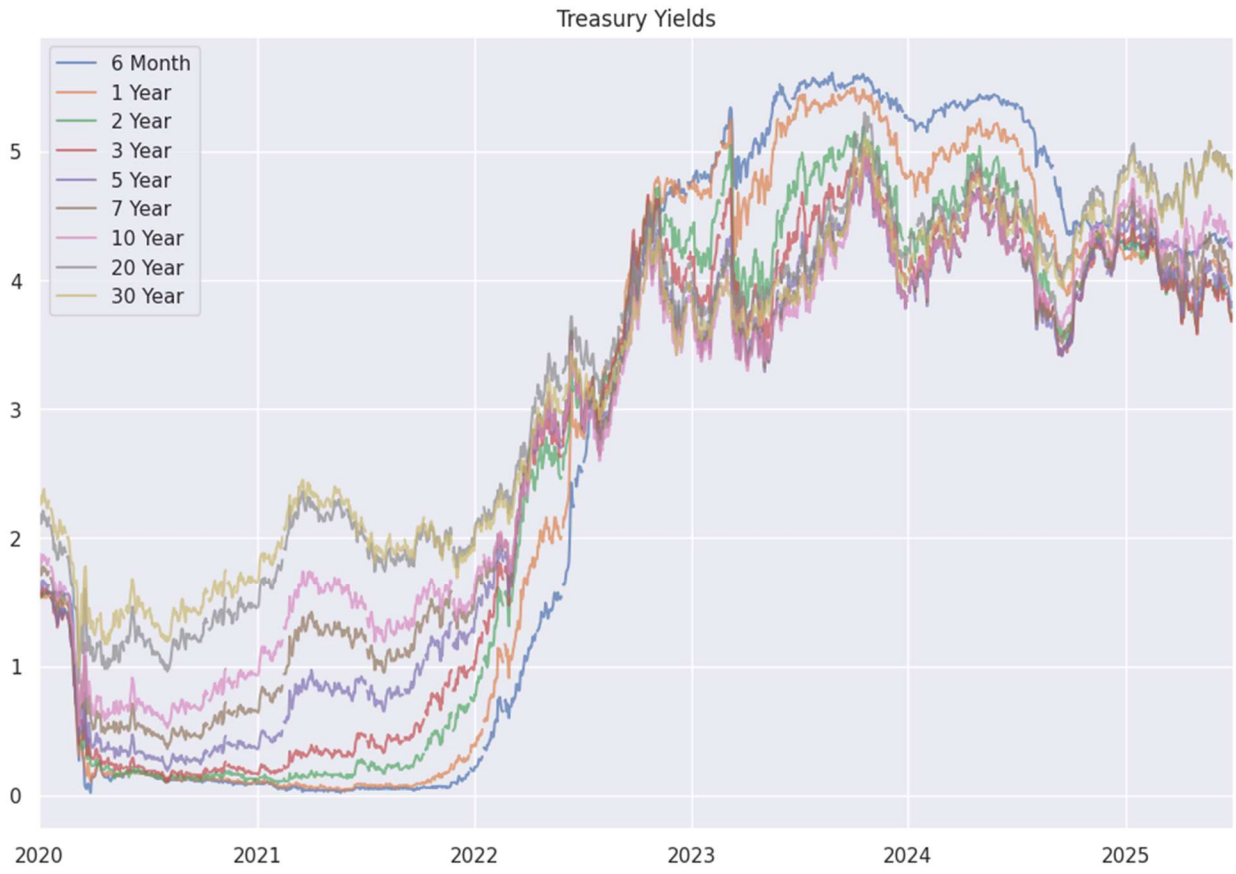



Figure 1 shows the evolution of long and short-term treasury yields and the effect of market conditions or monetary policy shifts.

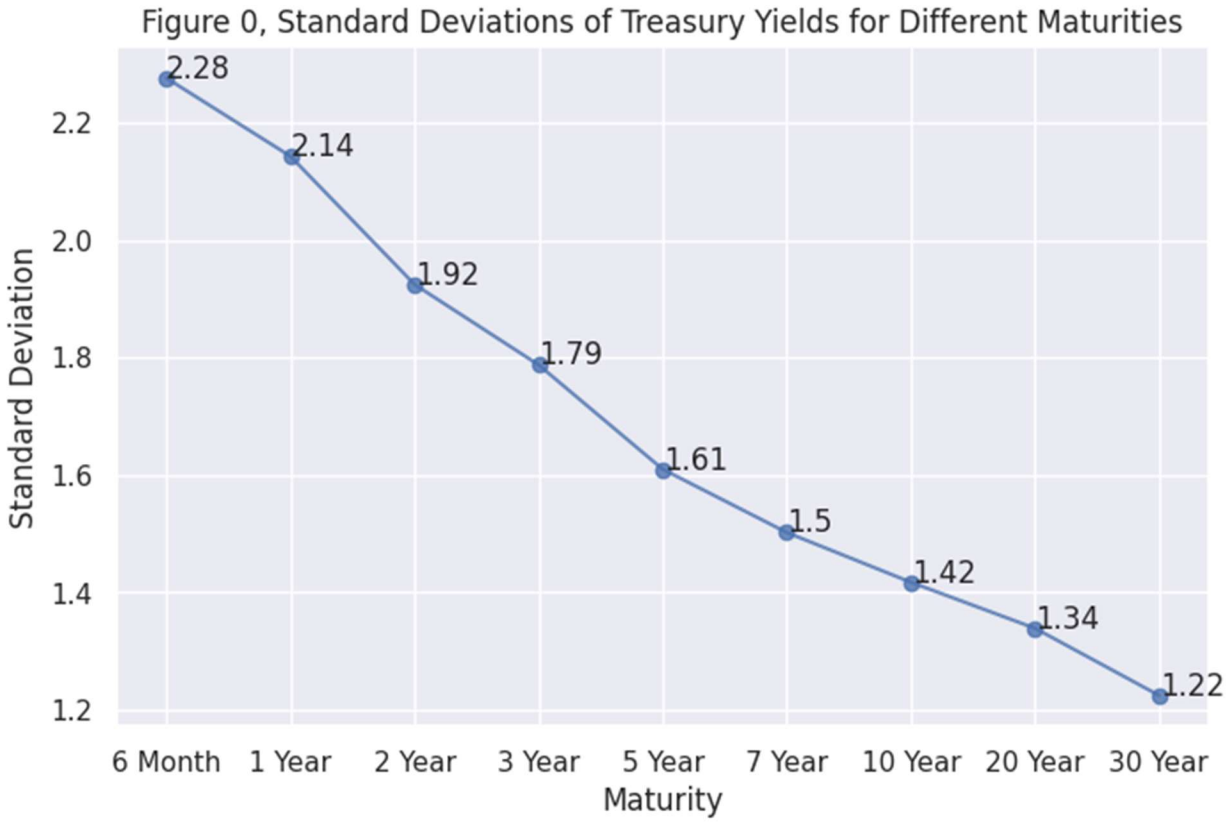
From 2020 to 2022, yield continued near a historical low due to accommodating monetary policy during the COVID-19 pandemic.

From 2022, all maturities yield curve surged sharply, reflecting violent interest rate hikes by the FED to fight inflation. Inversion, when short-term rates cross above long-term, signals of potential recession expectations and also steepening and flattening, can be seen on the yield curves.

Yields continued to swell from 2023 to 2024, but the gap between short-term and long-term rates lessened, depicting shifting market expectations about future interest rates.

## The Standard Deviations

Using yield data, standard deviations are calculated




Short-term maturities (6M, 1Y) display higher standard deviations (2.28%, 2.14%), that is they are more volatile.

Long-term maturities (20Y, 30Y) are less volatile with falling standard deviations to 1.34% and 1.22%, respectively.

This settles that short-term rates are sensitive to economic and policy shifts, whereas long-term rates are stable, frequently showing long-run expectations like inflation and growth.

## Treasury Bond Price Yield Curve

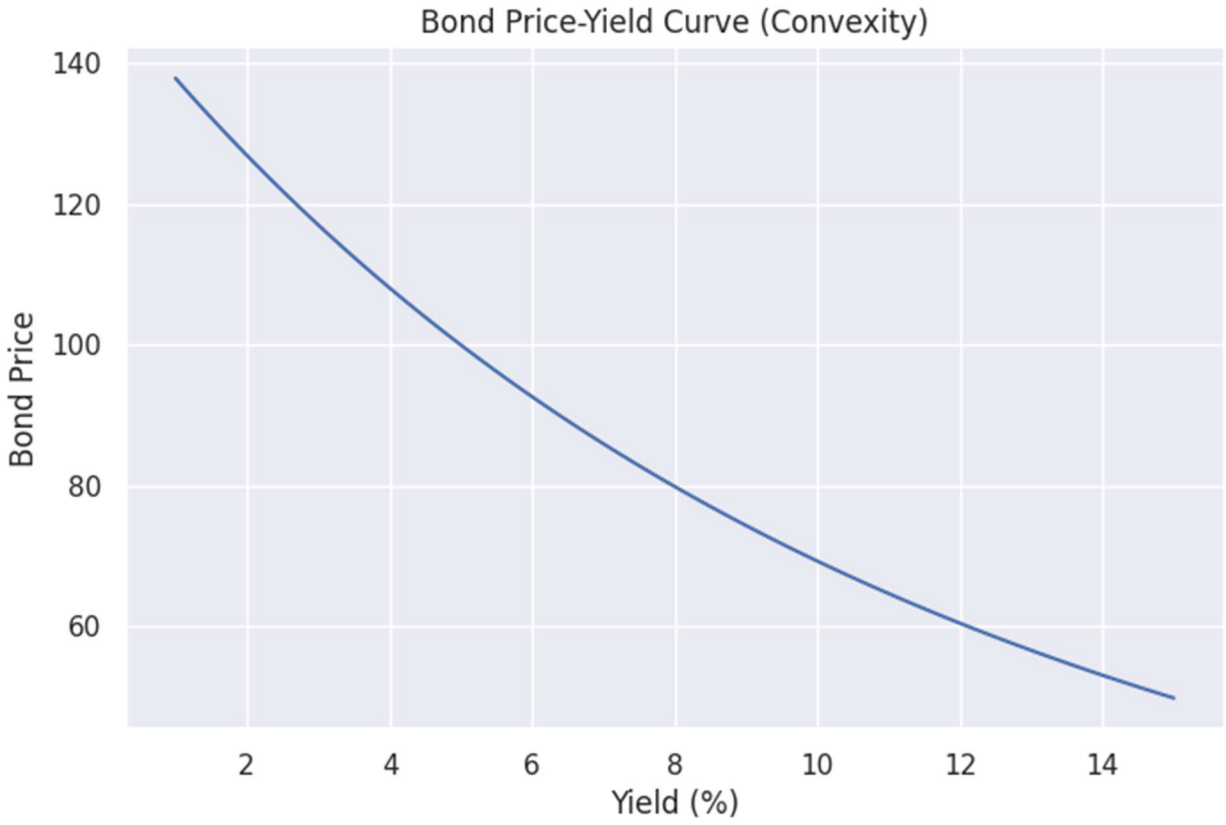
- Bond prices and bond yields follow an inverse and nonlinear pattern.
- When yields increase, the present value of future cash flows declines, so bond prices drop.
- When yields decrease, future payments are more valued, so bond prices increase.

But it's not linear – it's curved (convex). This is as:

- As yields cut, bond prices surge at a quicker rate.
- As yields surge, bond prices cut at a gentler rate.

This curvature is called "bond convexity" and is vital for duration risk understanding.

It helps capture, level, slope, and curvature of the curve, which is helpful for understanding yield analysis. (Rebonato). 90% of the volatility is captured through level and slope features. (Oprea)




## Curvature or Convexity in Bond Prices

The curved nature of the yield curve is called convexity. It helps understand the relationship.

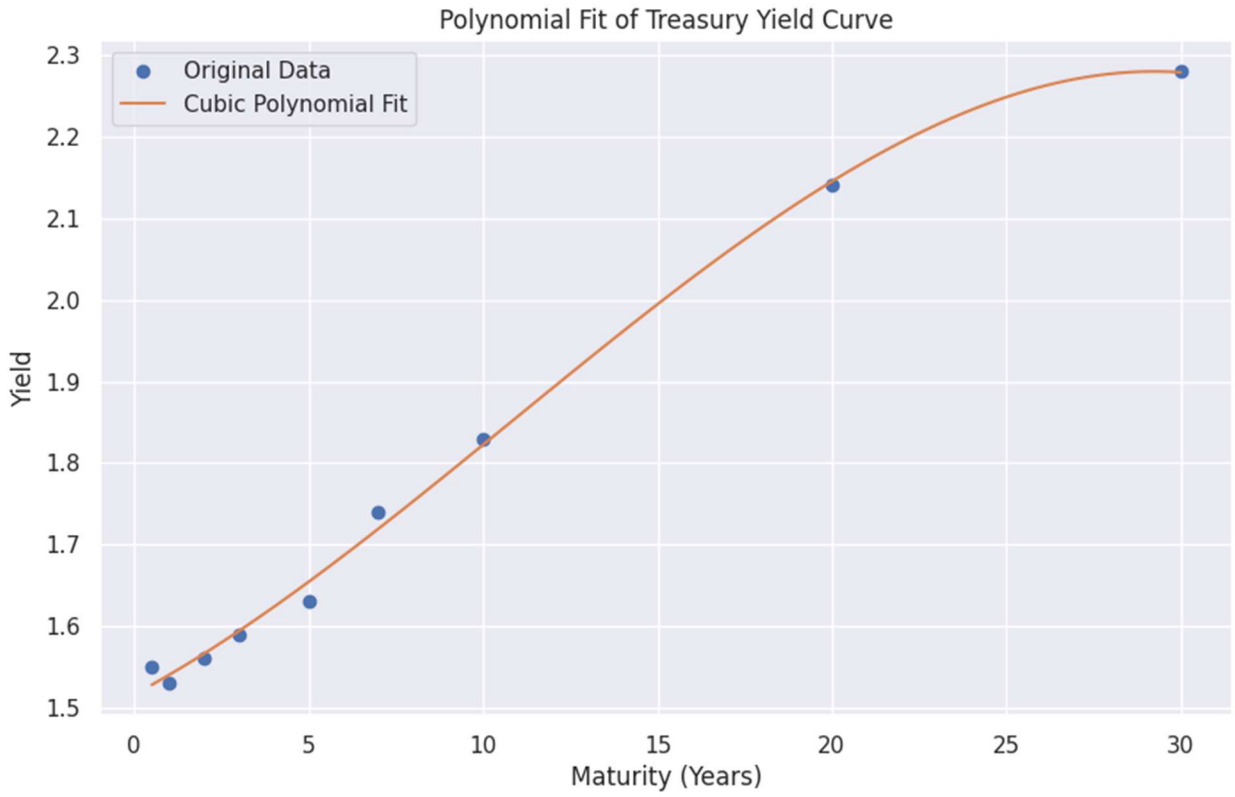
Bond prices are very sensitive to yield change and vice versa. It means the bond price does not change by the same amount for a unit change.

This behaviour helps investors to manage risk, evaluating bond prices.

## Polynomial Fitting for Treasury Yield Curve

The relationship between yield and time to maturity is called the Term Structure of Interest Rates or yield curve.

Real data is scattered in nature, but polynomial fitting using math function creates a smooth curve. For example, Nelson-Siegel and Cubic Spline.




There can be an overfitting problem with higher-degree polynomials.

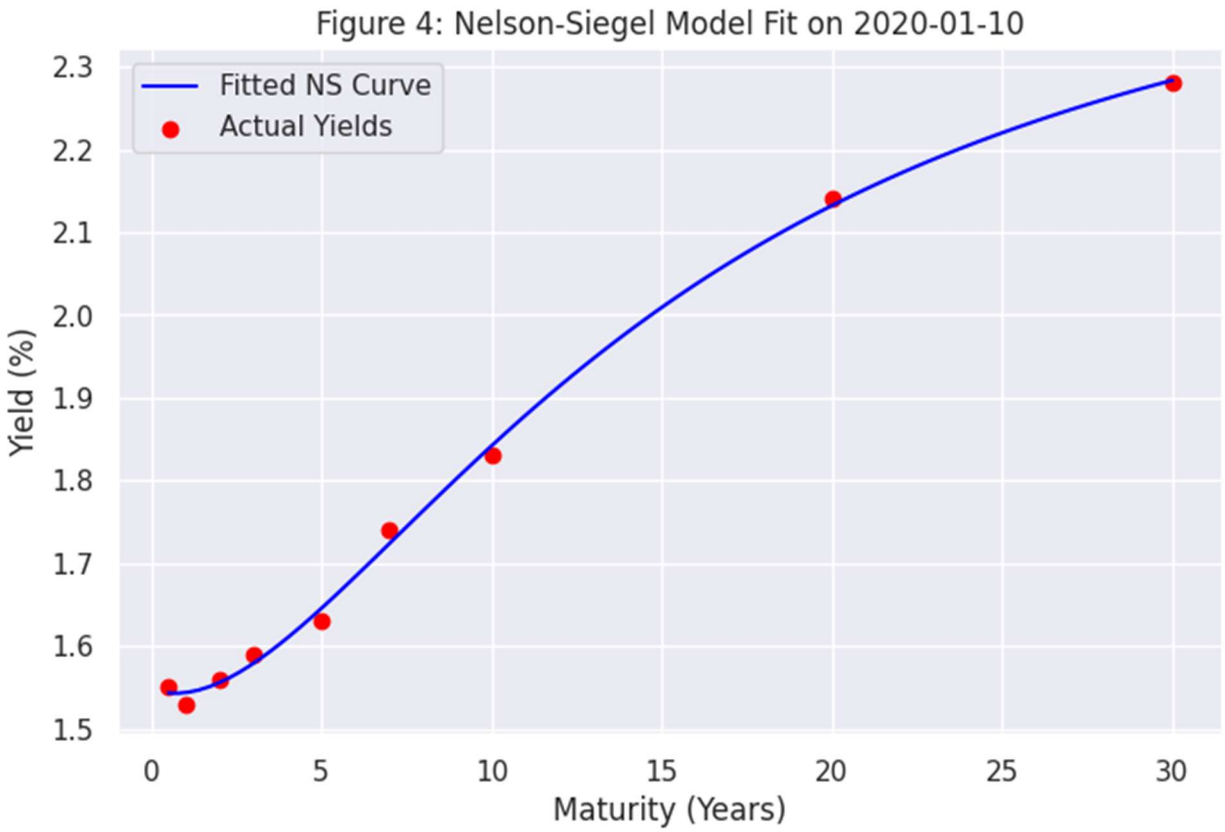
## The Nelson-Siegel Model

It is a parametric stable model that fits the yield curve, having four parameters capturing the full curve. Its parameters have economic meaning. Therefore, central banks, traders, and economists use it.

$$y(\tau) = \beta_0 + \beta_1 \cdot \backslash frac\{1 - e^{\{-\lambda\tau\}}\}\{\lambda\tau\} + \beta_2 \cdot \left(\backslash frac\{1 - e^{\{-\lambda\tau\}}\}\{\lambda\tau\} - e^{\{-\lambda\ \}}\right)$$

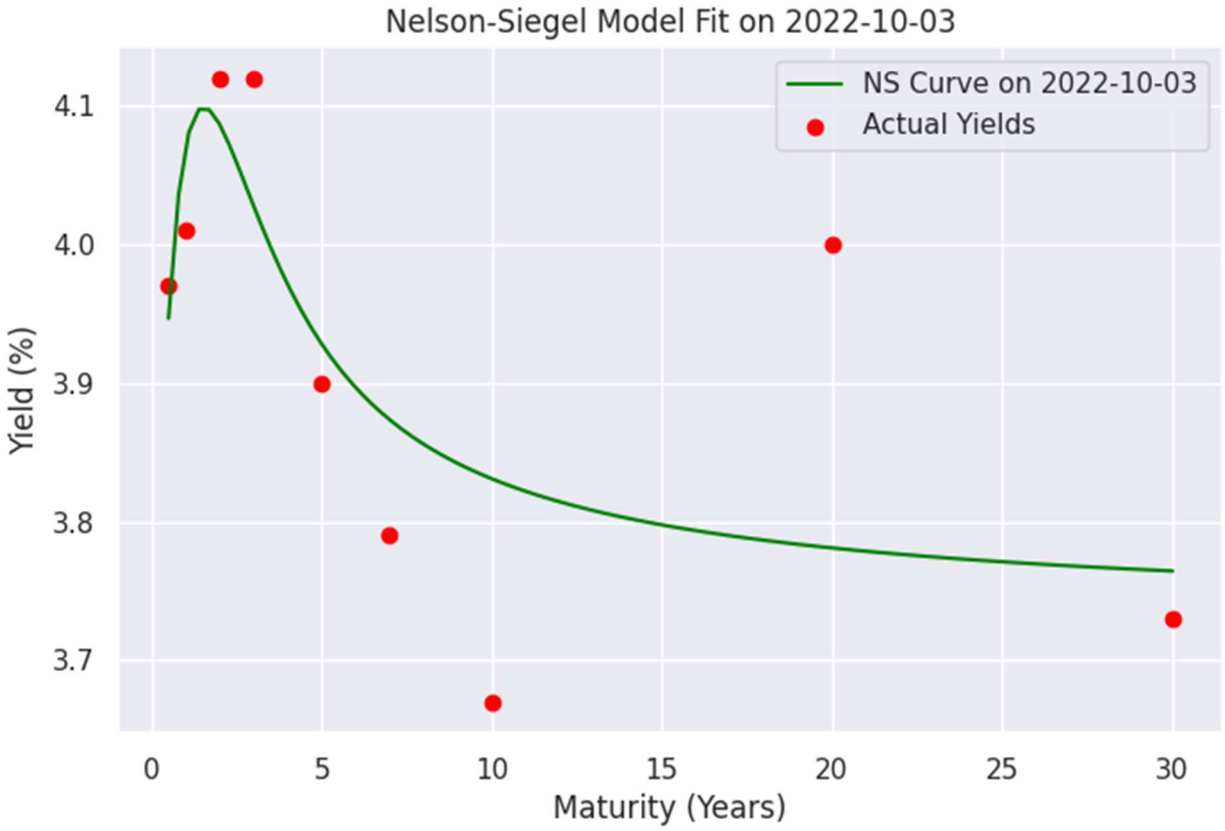
Using calibrate_ns_ols function from the nelson_siegel_svensson Python package, on 2020-01-10, at $\tau_0$ = 0.5, the model effectively captures the term structure of interest rates with only a few parameters.

**$\beta_0$ ≈ 2.62**: Signifies the **long-term yield level**, which shows the yield curve at distant maturities.
**$\beta_1$ ≈ -1.07**: Imprisonments the **slope or short-term effect,** a negative value shows a **sharp drop in yields** at shorter maturities.
**$\beta_2$ ≈ -1.19**: Labels the **curvature;** a negative value suggests a **dip in yields** at medium-term maturities.
**$\tau$ ≈ 4.47**: Controls the location on the maturity axis where this bump or curvature effect arises.




With the Nelson-Siegel Model, we can predict future interest rate moves. (Pape)

With new date at 2022-10-03



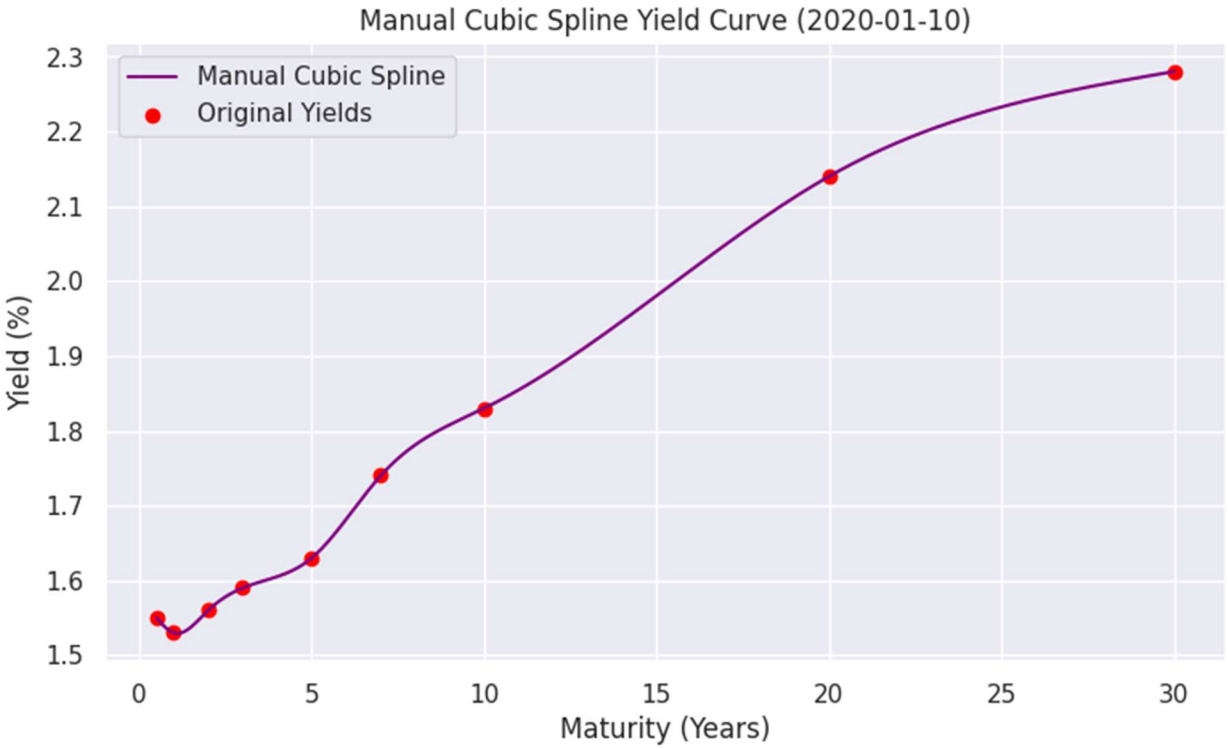

The model picked well the curve shift and structure. Green NS model picked humped shape.

## Cubic Spline Yield Curve Fitting

It joins a series of cubic polynomials that creates a smooth curve, passing through all the data. The first and second derivatives are continuous at the knots. It fits better than the NS model.

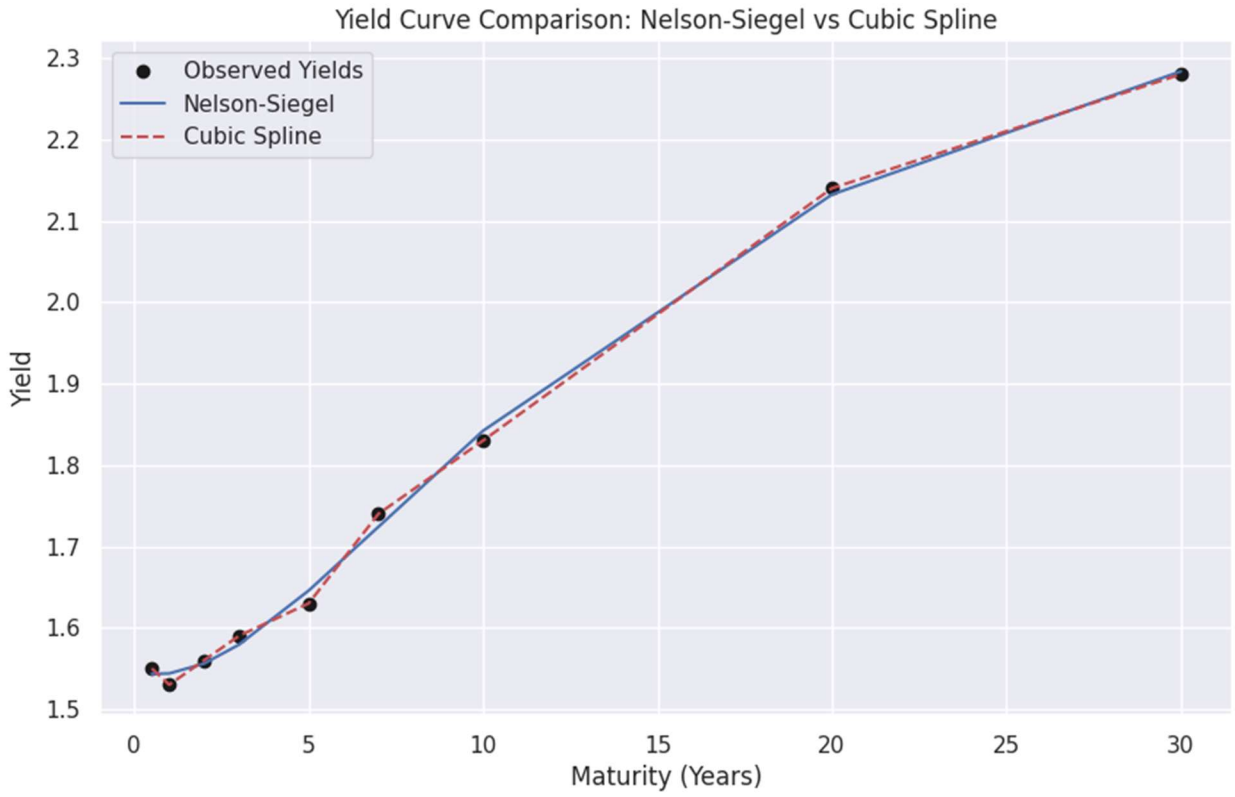
If data is trustworthy, it improves interpretation as it does not follow any economic theory.

A spline curve provides a realistic and smooth curve that fits bond yields well as these change gradually, just like a Spline curve.




## Fit Comparison: Cubic Spline vs. Nelson-Siegel

Nelson-Siegel is a parametric, theory-driven model, whereas Cubic Spline is a non-parametric, data-driven model.




Both models performed a good job for long-term maturities, that is, (10–30 years), but for short-term (1–10 years), Cubic-Spline shows wiggles, a sign of overfitting. On the other hand, Nelson-Siegel is smooth and stable, which is preferable for economic interpretations like level, slope, and curvature factors.

**Ethics of Smoothing Data in Financial Reporting**

In financial data analysis, smoothing is used to show data less volatile. It is beneficial in context of academic study or forecasting, but it causes trouble if it is used to hide or mislead investors by hiding the real picture by not disclosing the whole information. (Le Marois)

**Why Nelson-Siegel Is Ethical Yield Curve Smoothing**

Smoothing produced by Nelson-Siegel is ethical because it is transparent, mathematically defined, and openly disclosed in academic and financial literature. Whereas, it provides a continuous curve with a small number of parameters. It captures important information like level, slope, and curvature, essential for understanding financial analysis for risk management, portfolio management, and volatility.

Ethical practices depend on how and why smoothing is applied. These are not simple techniques.

**References**


- Rebonato, Riccardo. *Bond Pricing and Yield Curve Modeling: A Structural Approach*. Cambridge University Press, 2018.
- Oprea, Andreea, *The Use of Principal Component Analysis (PCA) in Building Yield Curve Scenarios and Identifying Relative-Value Trading Opportunities on the Romanian Government Bond Market*, Journal of Risk and Financial Management, 2022, 15(6), 247
- Le Marois, Oliver, and Raphael Douady. "Return Smoothing Practices." *The Hedge Fund Journal*, no. 30, 2007, https://thehedgefundjournal.com/return-smoothing-practices/.
- * Pape. "Understanding the Nelson-Siegel-Svensson (NSS) Model for Bond Yield Curve Analysis." Medium, 2024 May 12. https://medium.com/@pape14/understanding-the-nelson-siegel-svensson-nss-model-for-bond-yield-curve-analysis-2a23202cbf6b.
    - Svensson, Lars. "Estimating and Interpreting Forward Interest Rates: Sweden 1992-1994." *NBER Working Paper Series*, no. 4871, 1994.

## Task 3 Exploiting Correlation

## 1. Synthetic Data Part

**Generating 5 Uncorrelated Gaussian Random Variables**

Using NumPy, 1,000,000 synthetic observations with 5 uncorrelated features were generated from a standard normal distribution (mean = 0, standard deviation = 1).

To improve readability, array were converted to DataFrame with column name Var1 through Var5. Below is the first five rows of the synthetic data for rapid review.

| | Var1 | Var2 | Var3 | Var4 | Var5 |
|---|---|---|---|---|---|
| **0** | 1.764052 | 0.400157 | 0.978738 | 2.240893 | 1.867558 |
| **1** | -0.977278 | 0.950088 | -0.151357 | -0.103219 | 0.410599 |
| **2** | 0.144044 | 1.454274 | 0.761038 | 0.121675 | 0.443863 |
| **3** | 0.333674 | 1.494079 | -0.205158 | 0.313068 | -0.854096 |
| **4** | -2.552990 | 0.653619 | 0.864436 | -0.742165 | 2.269755 |

This data is saved and its downloaded file is saved in zip folder for review. It can also be downloaded any time from Jupiter Notebook.

**Principal Component Analysis (PCA) on Synthetic Data**

We applied Principal Component Analysis (PCA) using sci

kit-learn library to comprehend the structure and variance within our synthetic dataset.

Basically, PCA is a dimensionality reduction technique.

It transforms original variables into a new set of uncorrelated variables called principal components. (Oprea)

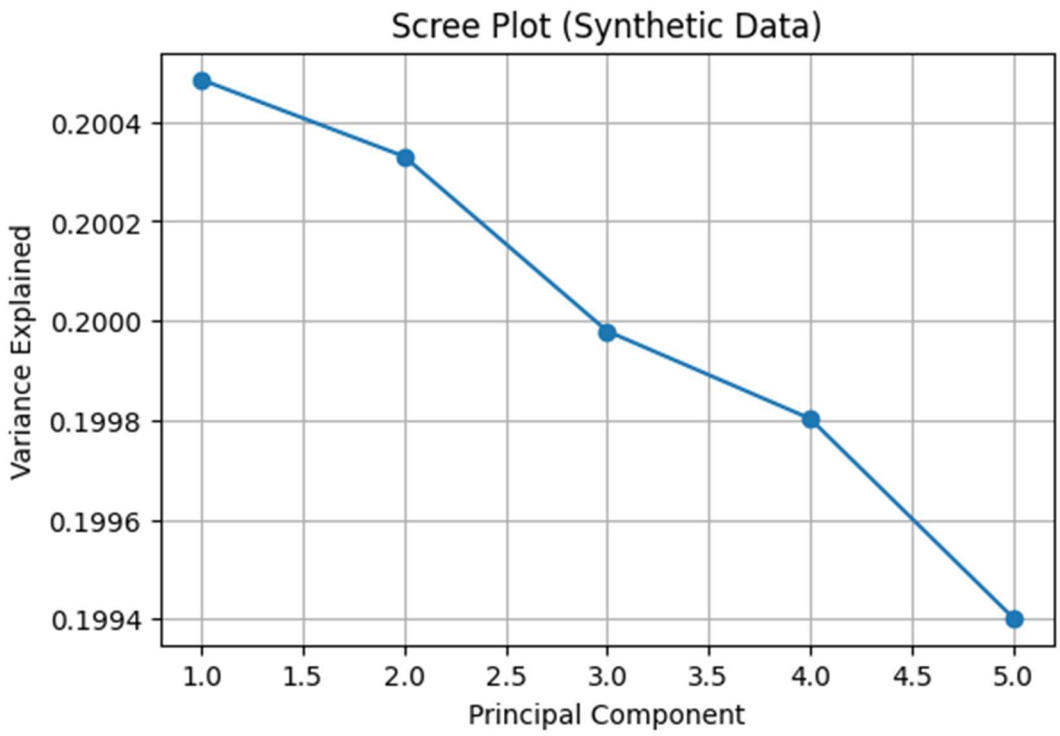
Then, it arranges them based on the variance (information) explained, captured from the data.

The proportion of variance explained by each principal component is given below:

[0.2004849, 0.20033032, 0.19997914, 0.19980312, 0.19940253]

This shows that each component explains approximately 20% of the variance. It is as per expectation, because:

We generated the uncorrelated data which have equal variance using the standard normal distribution, and it is confirmed using PCA. It reinforces that the features in our synthetic dataset are equally important and uncorrelated.




The above screen plot confirms the results of PCA. It shows no single component dominate while explain the variance. Here each component explains about 20-25% of the total variance.

Because of its independence of synthetic variables, it sets the benchmark for real-world financial data comparison.

## 2. Real Data Part

We again used the FRED API to fetch U.S. Treasury Yield Data for analysis of interest rate trends.

U.S. Treasury yields can be found from FRED via Fred API for maturities including 6-month, 1-year, 2-year, 3-year, 5-year, 7-year, 10-year, 20-year, 30-year, covering timeframe 2020-2025.

We assigned human-readable labels to each series of maturities 1 Year, 2 Year, 5 Year, 10 Year and 30 Year from January 1, 2020, to July 1, 2025.

After computing the daily changes, we ran PCA on yield changes and found the following results.

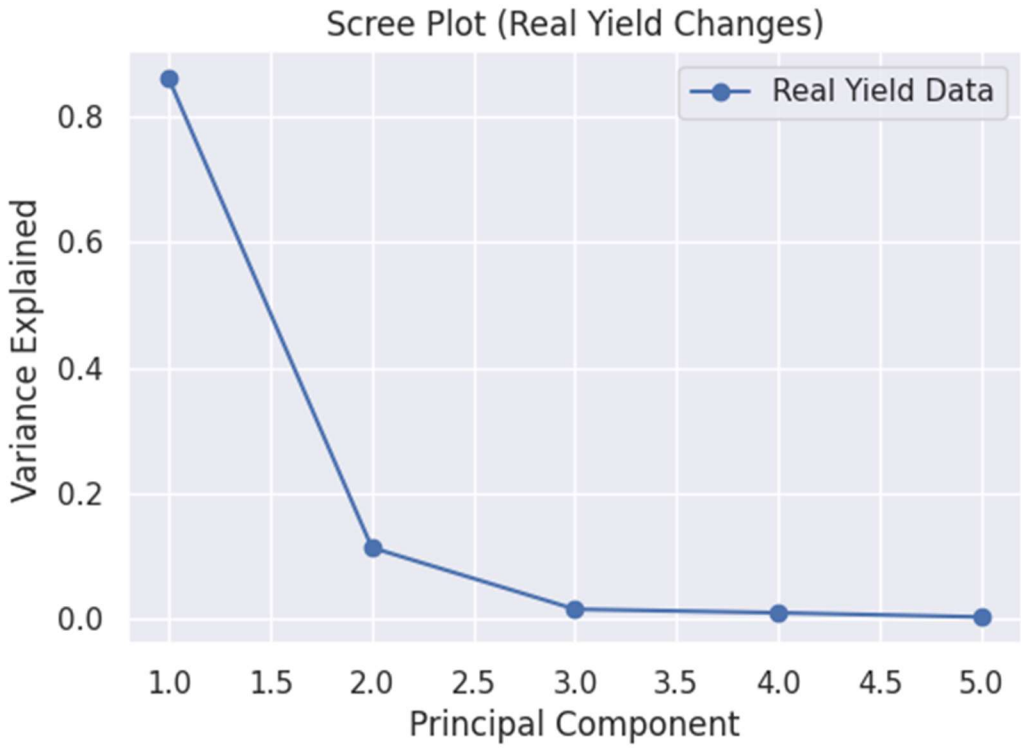
[0.86092879 0.11285782 0.01477498 0.0090822 0.00235621]

This uncovered the patterns in real U.S. Treasury yield changes from PCA. PCA transformed correlated yield movements into principal components.

Results show that the first component explains about 86% of the total variance (information), which indicates a strong underlying factor, often called the Level Factor in Fixed Income Analysis.

So this PCA revealed that most yield curve information is explained by just the first 1 to 2 components which can be called level, slope, and curvature in yield curve analysis.

For better understanding, let's draw a yield curve. (Bjerring)




Above screen plot depicts explained variance from PCA on U.S. Treasury yield changes. This shows:

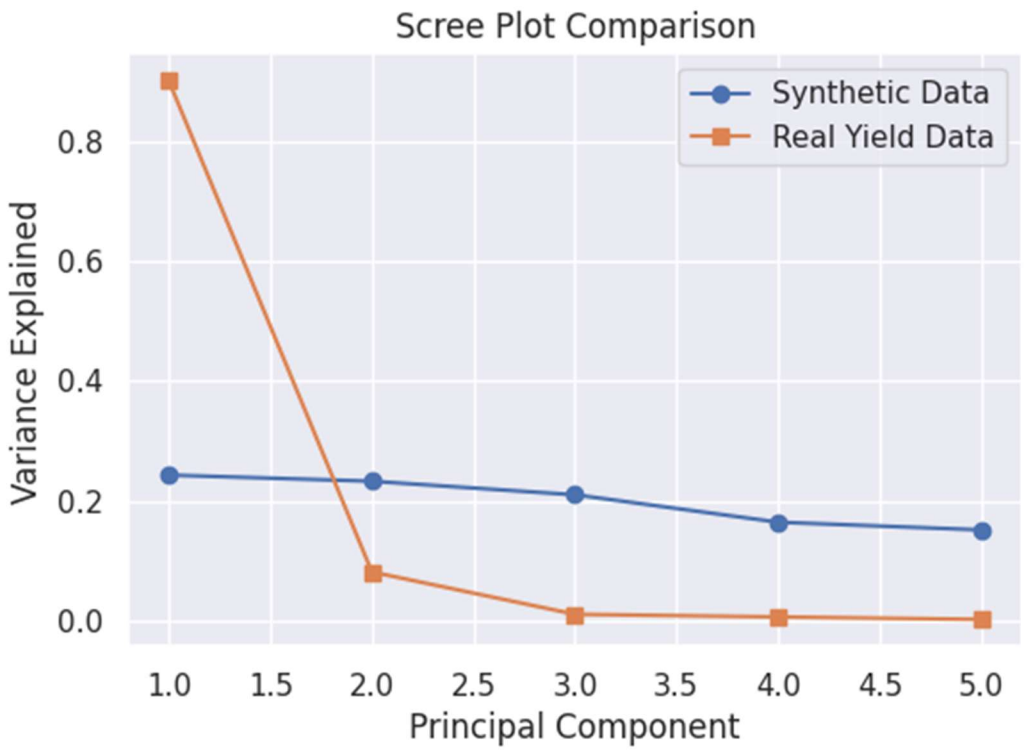
- 90% of total variance is explained by First Component, which is like Level factor of interest rates.
- 10% variation by the second component, usually interpreted as Slope of the yield curve.
- Remaining are just acting like a noise are explaining very little variances.

It confirms the structured behaviour of U.S. Treasury yield changes. It shows that first one or two principal components can explain the most of the behaviour. This also supports the concept of real-world yield curves are mostly shift in parallel and there is high correlation among many components. First three PCs which normally explained most of information are called as follow:

PC1: Level (general shift in yield curve)

PC2: Slope (difference between short and long rates)

PC3: Curvature (hump shape)




**Comparison of Synthetic vs. Real Screen Plots**

We draw overlay plots of both datasets.

Therefore, we can conclude that, the synthetic screen plot illustrates evenly distributed variance across components which is consistent with uncorrelated random variables.

In contrast, first component captures about 90% of information which reveals high correlation among maturities. It validates the use of factor models like Nelson-Siegel.

It is recommended to use NS model due its fit with real-world yield curve information capturing.

So the above comparison between synthetic vs. real-world data helped understand that real-world data is more structured which confirms the real market forces.

**References**

- Bjerring, Thomas T. "The Yield Curve and Its Components." *Github*, 16 October 2019, https://bjerring.github.io/bonds/2019/10/16/the-yield-curve-and-its-components.html.

- Oprea, Andreea. "The Use of Principal Component Analysis (PCA) in Building Yield Curve Scenarios and Identifying Relative-Value Trading Opportunities on the Romanian Government Bond Market." *Journal of Risk and Financial Management*, vol. 15, no. 6, 2022. https://www.mdpi.com/1911-8074/15/6/247.

## Task 4: Empirical Analysis of ETFs

We picked the XLF ETF and took its data for top 30 largest holdings, e.g., JPM, BAC, GS, MS, etc. Data is downloaded from January 1, 2024, to July 1, 2024 (6 months-180 days) using the yfinance package. (Yahoo Finance)

We extracted only the 'Adj Close' prices to account for dividends and splits.

["BRK-B","JPM","V","MA","BAC","WFC","GS","AXP","MS","SPGI",

"C","SCHW","BLK","PGR","COF","BX","MMC","CB","ICE","CME",

"FISV","KKR","AJG","PNC","MCO","AON","COIN","PYPL","USB","BK"]

In this data, one ticker (FISV) failed to download due to being possibly delisted.

Below is the table of first five rows

| Ticker | AJG | AON | AXP | BAC | BK | BLK | BRK-B | BX | C | CB | ... | MMC | MS | PGR | PNC | PYPL | SCHW | SPGI | USB | V | WFC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Date | | | | | | | | | | | | | | | | | | | | | |
| 2024-01-02 | 222.151123 | 285.621185 | 184.565155 | 32.658970 | 50.309341 | 773.296021 | 362.459991 | 123.142075 | 50.381668 | 223.651642 | ... | 186.109558 | 89.284668 | 157.455460 | 147.862411 | 61.459999 | 67.648895 | 431.435944 | 41.019020 | 256.049530 | 47.562691 |
| 2024-01-03 | 222.170883 | 288.467346 | 182.614761 | 32.302517 | 49.752266 | 757.691040 | 366.750000 | 117.458733 | 50.951595 | 222.726929 | ... | 187.235001 | 87.392487 | 158.955643 | 143.289536 | 58.630001 | 65.641960 | 424.980042 | 39.935390 | 255.169174 | 46.935970 |
| 2024-01-04 | 223.039719 | 288.318542 | 184.010986 | 32.562634 | 49.953968 | 759.430298 | 363.679993 | 117.986755 | 51.075085 | 223.602448 | ... | 187.401398 | 87.620697 | 158.790054 | 144.111511 | 58.450001 | 65.485313 | 424.792236 | 40.365105 | 256.781433 | 47.514473 |
| 2024-01-05 | 222.427597 | 287.178162 | 185.898895 | 33.169563 | 50.318939 | 756.415466 | 365.589996 | 117.324333 | 51.607018 | 223.995956 | ... | 186.765244 | 88.657112 | 158.507568 | 147.607300 | 60.119999 | 65.818169 | 423.071930 | 40.925602 | 256.860596 | 48.131542 |
| 2024-01-08 | 225.942505 | 293.078613 | 186.046371 | 32.909451 | 50.616688 | 770.291016 | 368.179993 | 117.449135 | 51.303051 | 222.520370 | ... | 187.509033 | 88.913841 | 161.098831 | 148.826111 | 61.740002 | 66.376205 | 427.263855 | 41.065723 | 259.679504 | 48.131542 |

5 rows × 30 columns

Then we computed the daily returns, and top five rows are presented below.

| Ticker | AJG | AON | AXP | BAC | BK | BLK | BRK-B | BX | C | CB | ... | MMC | MS | PGR | PNC | PYPL | SCHW | SPGI | USB | V | WFC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Date | | | | | | | | | | | | | | | | | | | | | |
| 2024-01-03 | 0.000089 | 0.009965 | -0.010568 | -0.010914 | -0.011073 | -0.020180 | 0.011836 | -0.046153 | 0.011312 | -0.004135 | ... | 0.006047 | -0.021193 | 0.009528 | -0.030927 | -0.046046 | -0.029667 | -0.014964 | -0.026418 | -0.003438 | -0.013177 |
| 2024-01-04 | 0.003911 | -0.000516 | 0.007646 | 0.008053 | 0.004054 | 0.002295 | -0.008371 | 0.004495 | 0.002424 | 0.003931 | ... | 0.000889 | 0.002611 | -0.001042 | 0.005736 | -0.003070 | -0.002386 | -0.000442 | 0.010760 | 0.006318 | 0.012325 |
| 2024-01-05 | -0.002744 | -0.003955 | 0.010260 | 0.018639 | 0.007306 | -0.003970 | 0.005252 | -0.005614 | 0.010415 | 0.001760 | ... | -0.003395 | 0.011828 | -0.001779 | 0.024258 | 0.028571 | 0.005083 | -0.004050 | 0.013886 | 0.000308 | 0.012987 |
| 2024-01-08 | 0.015802 | 0.020546 | 0.000793 | -0.007842 | 0.005917 | 0.018344 | 0.007084 | 0.001064 | -0.005890 | -0.006588 | ... | 0.003982 | 0.002896 | 0.016348 | 0.008257 | 0.026946 | 0.008478 | 0.009908 | 0.003424 | 0.010974 | 0.000000 |
| 2024-01-09 | -0.011799 | -0.005279 | -0.012896 | -0.015515 | 0.004744 | -0.003349 | -0.003477 | -0.010953 | -0.009813 | -0.003935 | ... | -0.002662 | -0.015506 | 0.007861 | -0.019109 | -0.011176 | -0.015487 | -0.004628 | -0.011374 | 0.003009 | -0.012620 |

5 rows × 29 columns

FISV was dropped due to data issues.

### Covariance and Correlation Matrices

Covariance and correlation matrix of daily returns of top 30 holdings of XLF ETF help us find co-movements of stock returns for financial analysis.

Covariance Matrix measures the co-movement of stock returns (in actual units). Whereas a correlation matrix measures the strength and direction of the relationship between stock returns (scaled between -1 and 1)

We also confirmed the symmetry of Matrices.

- 1 means, two stock moves similarly
- 0 means no linear relationship
- -1 means, opposite movement.

So, the covariance matrix provides raw co-movement in returns, whereas, correlation matrix shows standardized. Both are used for PCA to do financial analysis.
Below is are matrix of Covariance and Correlation.

| Ticker | AJG | AON | AXP | BAC | BK | BLK | BRK-B | BX | C | CB | ... | MMC | MS | PGR | PNC | PYPL | SCHW | SPGI | USB | V | WFC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ticker** | | | | | | | | | | | | | | | | | | | | | |
| **AJG** | 0.000103 | 0.000073 | 0.000020 | 0.000039 | 0.000030 | 0.000041 | 0.000034 | 0.000032 | 0.000028 | 0.000052 | ... | 0.000058 | 0.000017 | 0.000036 | 0.000023 | 0.000014 | 0.000021 | 0.000034 | 0.000036 | 0.000022 | 0.000035 |
| **AON** | 0.000073 | 0.000152 | 0.000027 | 0.000030 | 0.000014 | 0.000038 | 0.000028 | 0.000025 | 0.000005 | 0.000027 | ... | 0.000062 | 0.000013 | 0.000032 | 0.000015 | 0.000008 | 0.000021 | 0.000016 | 0.000023 | 0.000018 | 0.000023 |
| **AXP** | 0.000020 | 0.000027 | 0.000181 | 0.000085 | 0.000051 | 0.000059 | 0.000046 | 0.000083 | 0.000077 | 0.000034 | ... | 0.000024 | 0.000060 | 0.000046 | 0.000078 | 0.000062 | 0.000068 | 0.000015 | 0.000086 | 0.000018 | 0.000084 |
| **BAC** | 0.000039 | 0.000030 | 0.000085 | 0.000167 | 0.000086 | 0.000080 | 0.000048 | 0.000114 | 0.000130 | 0.000037 | ... | 0.000030 | 0.000103 | 0.000016 | 0.000140 | 0.000064 | 0.000084 | 0.000040 | 0.000141 | 0.000021 | 0.000123 |
| **BK** | 0.000030 | 0.000014 | 0.000051 | 0.000086 | 0.000107 | 0.000064 | 0.000036 | 0.000082 | 0.000078 | 0.000033 | ... | 0.000023 | 0.000054 | 0.000015 | 0.000088 | 0.000069 | 0.000065 | 0.000036 | 0.000091 | 0.000023 | 0.000059 |

5 rows × 29 columns

| Ticker | AJG | AON | AXP | BAC | BK | BLK | BRK-B | BX | C | CB | ... | MMC | MS | PGR | PNC | PYPL | SCHW | SPGI | USB | V | WFC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Ticker** | | | | | | | | | | | | | | | | | | | | | |
| **AJG** | 1.000000 | 0.583035 | 0.147244 | 0.296422 | 0.289740 | 0.360668 | 0.451945 | 0.177712 | 0.201944 | 0.468779 | ... | 0.658648 | 0.119854 | 0.306908 | 0.154391 | 0.062582 | 0.158846 | 0.332982 | 0.239731 | 0.250079 | 0.253969 |
| **AON** | 0.583035 | 1.000000 | 0.161102 | 0.186531 | 0.109213 | 0.275219 | 0.305696 | 0.112057 | 0.027364 | 0.198232 | ... | 0.578375 | 0.079169 | 0.225419 | 0.080860 | 0.029591 | 0.131532 | 0.129412 | 0.127073 | 0.168218 | 0.136025 |
| **AXP** | 0.147244 | 0.161102 | 1.000000 | 0.488779 | 0.370244 | 0.391015 | 0.453462 | 0.347760 | 0.419935 | 0.227077 | ... | 0.206721 | 0.327368 | 0.293099 | 0.395833 | 0.207744 | 0.384917 | 0.113694 | 0.428107 | 0.153636 | 0.463340 |
| **BAC** | 0.296422 | 0.186531 | 0.488779 | 1.000000 | 0.646994 | 0.555572 | 0.500869 | 0.493564 | 0.736273 | 0.261865 | ... | 0.269118 | 0.583731 | 0.103631 | 0.736828 | 0.223269 | 0.493611 | 0.311808 | 0.733252 | 0.184192 | 0.705055 |
| **BK** | 0.289740 | 0.109213 | 0.370244 | 0.646994 | 1.000000 | 0.557010 | 0.465147 | 0.444617 | 0.555709 | 0.291553 | ... | 0.258356 | 0.387311 | 0.120978 | 0.583768 | 0.299626 | 0.478409 | 0.346028 | 0.589741 | 0.256445 | 0.426438 |

5 rows × 29 columns

### Run PCA

We applied Principal Component Analysis (PCA) on stock returns to find key patterns that drive stock price movements. PCA provided us with uncorrelated linear combinations of the original variables.

From the extracted explained variance ratio, we found how much of the total variance each principal component explains.

The decomposition was performed using Scikit-learn's PCA class (Pedregosa et al.).

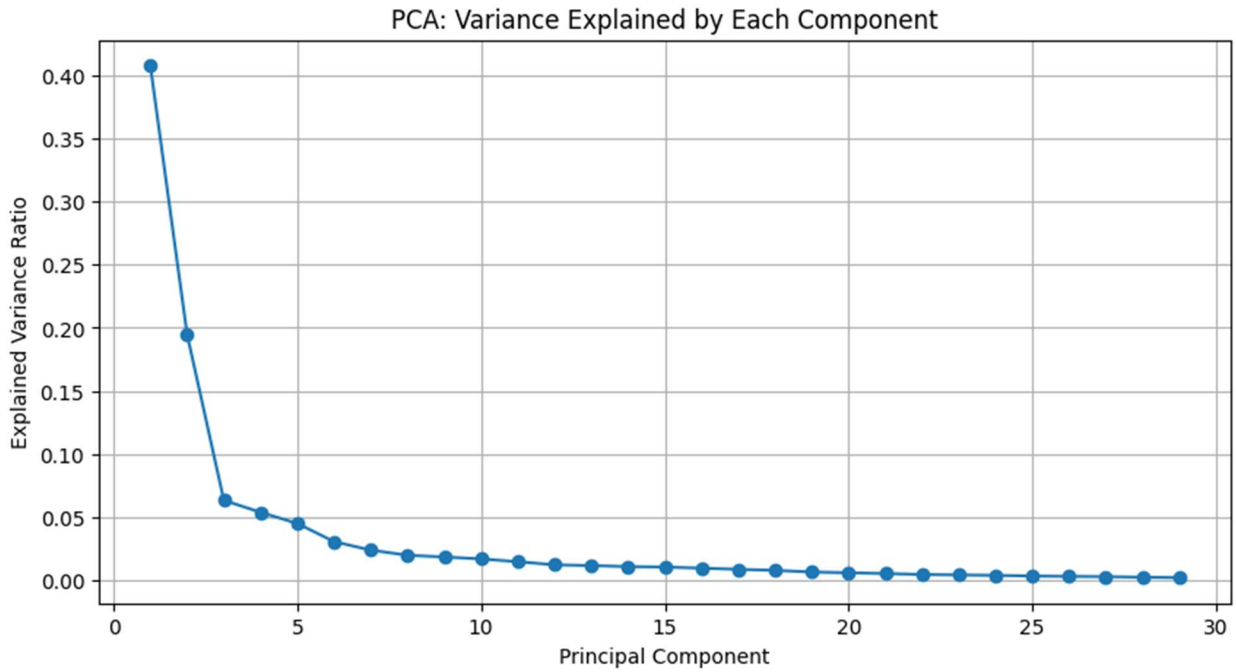
Below are the first top 10 components results.

[0.40780624, 0.19477112, 0.06319253, 0.05369445, 0.04484421,

0.03043447, 0.02376662, 0.01964991, 0.01827548, 0.01676436]

From here, we can see that the first PC alone explains about 41% of the variance in returns and the first three combined explain over 66%.

PCA reduces the dimensionality by transforming variables into principal components (Shlens). This reveals the importance of a small number of components that capture most of the market behaviour, a common phenomenon in financial data due to strong underlying market trends




The above chart shows how much of the total information in stock returns is captured by each PC.

It confirms the results we discussed earlier. The first 3 to 5 PC account for the majority of the variance, whereas from about the 6th onward, the curve flattens, showing diminishing returns in explanatory power.

**Singular Value Decomposition (SVD)**

It is a mathematical way similar to PCA. In PCA, we use eigenvalues of the covariance matrix for explaining variance, whereas in SVD, we decompose the original data matrix into matrices that show patterns, structure, and rank.

First, we standardize the return and then apply SVD to get U, S, and $V^T$.

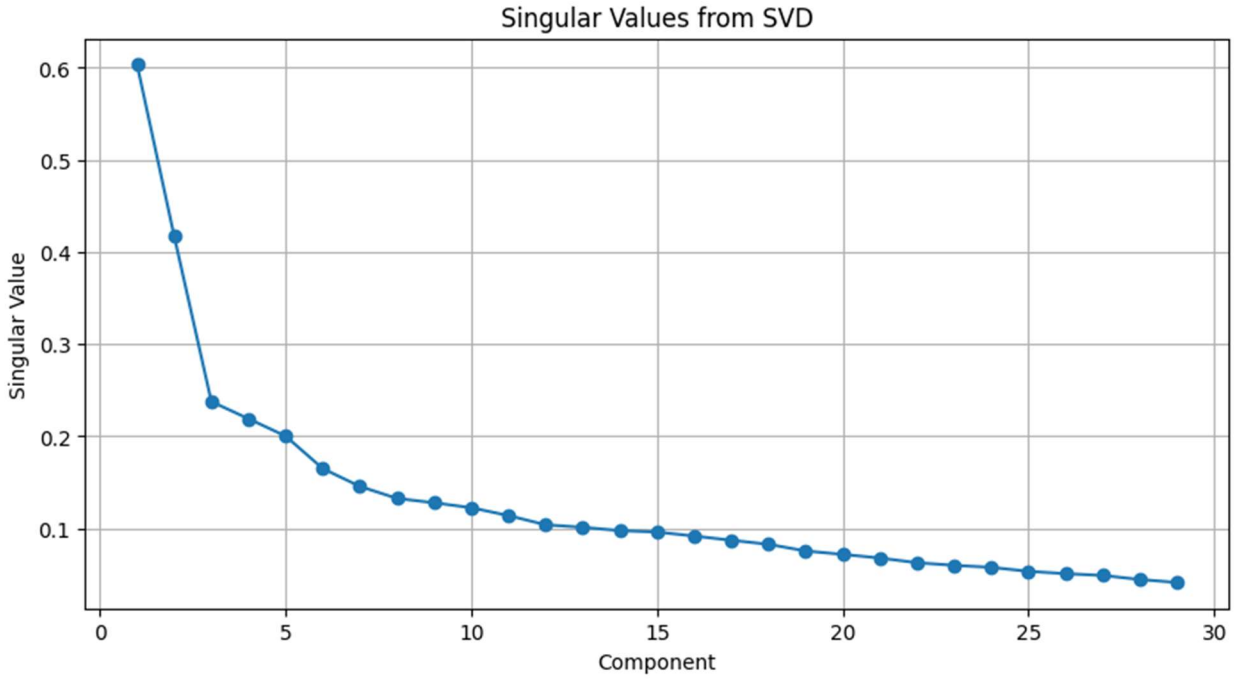
- U: Left singular vectors show stock patterns.
- S: Singular values related to the importance of each component.
- $V^T$ : Right singular vector for time-based patterns.

Following the output singular values:

[0.60362717 0.41716141 0.23761547 0.21903149 0.20016818 0.16490154
 0.1455381 0.13250191 0.12778395 0.122387  0.11390134 0.10386492
 0.10116691 0.09755001 0.09601941 0.09165305 0.08711353 0.08271381
 0.07547451 0.07174633 0.06779769 0.06270092 0.05986009 0.05767782
 0.05335419 0.05084449 0.04886925 0.04451151 0.04124854]

These are magnitudes of variance captured by each component. Larger values show their dominance just like in PCA.

The importance of SVD is that it is more stable and it is used under the hood by PCA. It works great with big data or when covariance matrices are not well-conditioned.




The above plot confirms what we have discussed. The first value is significantly important, and from the 6th or 7th onwards, information is less important.

**The 500-Word Explanation**

Prices only show the worth of a stock, while returns reveal change in their value, which is used to compare the performance of an investment and help in risk management. Return standardized comparisons regardless of their prices. For example, a S2 rise on a $20 stock is a 10% return, but it is 1% for a stock of $200. This ease of comparison using returns helps us evaluate performance, risk estimation, volatility, and Sharpe ratios. Returns are used to perform statistical analyses like covariance, correlation, Principal Component Analysis (PCA), and Singular Value Decomposition (SVD). (Pedregosa et al.)

PCA, or Principal Component Analysis, is based on the eigenvalue decomposition of the covariance (or correlation) matrix of the data. They are called principal components, orthogonal in directions, and maximize the information captured in the shape of variance in the dataset. Whereas, Singular Value Decomposition (SVD) does not require a covariance matrix, instead it runs on the original data matrix. Singular Value Decomposition (SVD) factorizes a data matrix into three factor matrices called U, S, and $V^T$. S consists of singular values, closely linked with the square root of the eigenvalues from PCA, which is also based on SVD itself. (3Blue1Brown).

Understanding the procedure of both PCA and SVD, it is vital to grasp the meaning of eigenvalues, eigenvectors, and singular values. In PCA, eigenvectors signify the directions (or axes) in which the data varies the most. These are the principal components, and each is orthogonal to the others, meaning they show independent patterns. Eigenvector shows the variance, whereas eigenvalues mention the scale of these. If a component has a large eigenvalue, then it is very important and contains a lot of information. So only a few reveal almost full information, which helps to reduce dimensionality while keeping much information.

In Singular Value Decomposition (SVD), the concept is similar to that in Principal Component Analysis (PCA), but is stated through singular values. These singular values, found in matrix S, measure the strength or importance of each component resulting from the data. The square of each singular value, divided by the sum of all squared singular values, specifies the proportion of total variance explained, much like the role of eigenvalues in PCA. Moreover, the columns of matrices U and V in SVD are similar to eigenvectors. Exactly, U corresponds to the "left singular vectors," which signify patterns across rows (such as stocks), while $V^T$ agrees to the "right singular vectors," revealing patterns across columns (such as time points).

In conclusion, understanding returns is vital for effective financial analysis, as they offer a reliable and similar method for measuring asset movements. Two influential tools, Principal Component Analysis (PCA) and Singular Value Decomposition (SVD), play a critical part in simplifying and interpreting the intricacies of return data by uncovering central patterns. While PCA is widely preferred in the finance industry for its emphasis on variance through covariance matrices, SVD delivers a multipurpose approach that shines in computational contexts. So, the concepts like eigenvalues, eigenvectors, and singular values are all smart techniques and fundamental to understanding the underlying trends and structures in financial analysis, where data is high-dimensional. So having a grip on such skill boosts our performance in the field of data science of finance (Shlens; Pedregosa et al.).

**References**


- Axler, Sheldon. *Linear Algebra Done Right*. 4th edition, Springer, 2024.
- . “Financial Select Sector SPDR Fund (XLF) Holdings.” *Yahoo Finance*, https://finance.yahoo.com/quote/XLF/holdings.
- Jonathon. *A Tutorial on Principal Component Analysis*. arXiv, 25 Apr. 2014, https://arxiv.org/abs/1404.1100.
- Shlens, Jonathon. *A Tutorial on Principal Component Analysis*. arXiv, 25 Apr. 2014, https://arxiv.org/abs/1404.1100.
- Pedregosa, Fabian, et al. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research*, vol. 12, 2011, pp. 2825–2830, https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.
- 3Blue1Brown. “Eigenvectors and Eigenvalues.” *YouTube*, 17 Dec. 2016, https://www.youtube.com/watch?v=PFDu9oVAE-g.