FRED-MD Data Analysis Group Project Report

Ashe Daniels, Harvey Stocks, Farhad Chichgar

November 2024

Abstract

This report looks at taking the FRED-MD data set created by Michael W. McCracken, Serena Ng and the federal reserve bank of St Louis, transforming it and applying Principle Component analysis and eigenanalysis to a factor model of a number of different factors in order to develop a forecasting approach on the data set in hopes to draw meaningful conclusions about different economic indicators in real world settings.

Contributions

Ashe Daniels - Abstract, Chapter 1 - Introduction, Chapter 2 - Fred-MD the Dataset
Harvey Stocks - Chapter 3 - Methodology.
Farhad Chichgar - Chapter 4 - Applications to the dataset.

We met as a group with our supervisor once every two weeks to discuss progress. We also met 3 more times as a group to discuss what progress has been made and if anyone needs any help and we communicated on WhatsApp. Ashe was responsible for writing all the heavily worded sections including the introduction and introducing the dataset. Harvey was responsible for creating the methodology for the project which will be used in the following section and Farhad was responsible for coding the project, creating plots and applying what was laid out in the methodology.

Contents

1 Introduction

1.1 History and Motivations

Developments in research and technology mean that we now have access to, and the means to use, data from multiple decades ago which was either not available or could not be computed easily. This is true for all fields, including medicine, engineering, and especially economics. Prior to these research and technology developments, when analysing time series data (data recorded over consistent intervals in time), we were very limited in the types of data we could analyse. If we tried to analyze T time series observations of U random variables, we could only analyze the data for large T but small U, largely due to computational efficiency. After technological advancements were created, we could extend the analysis of our data to allow for both large T and large U, enabling us to include a broader range of economic indicators for a greater depth of analysis. This allowed us to analyze and compare data from multiple decades and use as many variables as needed to draw meaningful economic conclusions, which are used today to influence economic policy for countries and to create statistics that help further our economy. The research paper by Bernanke and Boivin (2003) introduced the term "data-rich environment" for cases where both U and T are large.[1]

To understand why the FRED-MD dataset was created, it is important to look at the development of "big data" datasets and analysis in macroeconomic research. To our knowledge, the first US personalized macroeconomic database was created in a research paper by Stock and Watson (1996). The goal of this research was to examine the stability of different US economic relationships and whether different forecasting models became unstable over time, thus making the model less accurate. The data they collected included 76 economic time series and tested over 5,700 forecasting relationships. Their method of data collection, covering the period from 1959:1 to 1993:12, was based largely on four criteria: first, the data should include the main monthly aggregators and coincident indicators; second, the data should have the most important leading economic indicators; third, the data should be broad and represent a large range of time series to allow for more testing; and finally, each indicator's definition needed to be consistent, or, if not, it should be possible to adjust the time series with a simple additive or multiplicative splice. [2]

Once the data was collected, it was categorized into eight categories: (1) Output and Sales, (2) Employment, (3) New Orders, (4) Inventories, (5) Prices, (6) Interest Rates, (7) Money and Credit, and (8) Other Variables. Stock and Watson expanded on this research in 2002, extending their 76 series to 215 series, which were subsequently reduced to 149 series to create a balanced panel of economic indicators that would be most useful. The goal was to break down the indicators into several factors, estimate these factors, and use those estimates as predictors to develop a forecasting model that could be used by other economists. This method was called "diffusion index forecasting" and, as we will see in later chapters, this approach will become critical for the analysis in this report. [3]

This concept was developed further in a research paper by Bernanke et al. (2005), where they used 120 series from the DRI (data resources inc) database (the largest non-government economic database at the time) to estimate a factor-augmented vector autoregression (FAVAR). The idea was to use a large number of economic indicators to estimate common latent factors (underlying trends that we cannot directly observe or measure). By estimating these common latent factors from many variables, the FAVAR model summarized a large amount of information into a few factors, which were then easier to manage. [4]

Boivin and Giannoni (2006) aimed to estimate DSGE (Dynamic Stochastic General Equilibrium) models that treat measurement errors (where real-world data does not explicitly match the model concepts used) as the difference between the data and model concepts. This analysis included 91 variables and used the diffusion index forecasting approach to estimate the factor model adapted from Bernanke et al. (2005). They noticed that data with measurement errors

could be represented by factor models. [5]

The constant increase in available data led to situations where older data was no longer used (it might be irrelevant or too outdated to matter for present-day analysis). Stock and Watson collaborated again in 2006 to construct a dataset of 132 economic time series, extending the time period to 1959:01–2003:12. This time, the goal was to use the data to estimate structural FAVAR models, which help analyse how economic indicators react to shocks. Instead of the eight categories in their 1996 paper, they extended the number of categories to allow for broader coverage. The new categories were: (1) Real Output and Income, (2) Employment and Hours, (3) Real Retail, Manufacturing, and Trade Sales, (4) Consumption, (5) Housing Starts and Sales, (6) Real Inventories, (7) Orders, (8) Stock Prices, (9) Exchange Rates, (10) Interest Rates and Spreads, (11) Money and Credit Quantity Aggregates, (12) Price Indexes, (13) Average Hourly Earnings, and (14) Miscellaneous. These categories were taken from the GSI index (Global Insight's basic economic database), and this new dataset is sometimes referred to as the "Stock and Watson dataset," highlighting its importance to economic analysis. [6]

The data compiled by Stock and Watson was consistently updated, but it posed issues, such as not being updated regularly enough and being heavily reliant on data from numerous agencies to create the dataset. This motivated the creation of the FRED-MD dataset, which is consistently updated to provide a more standardized database that can be easily accessed by everyone.

1.2 Aim of the project

As we saw in the previous section, the methods for dealing with large datasets largely consist of breaking down the data into a number of factors, with the goal of estimating these factors to help develop a forecasting model that can be used to draw conclusions. Our aim of the project goes as follows, we first start by introducing the FRED-MD dataset (section 2) where we provide the origin of the dataset and how we will transform this dataset to make it easier to analyse. Section 3 is then the methodology of the project, we draw upon ideas seen in the working paper by Michael W. McCracken and Serena Ng (2015)[7] and Stock and Watson (2002)[3] consisting of principle component analysis (PCA) and eigenanalysis as our estimation method of choice to estimate a factor model to help us identify the underlying latent factors which will allows us to produce forecasts for certain variables within the dataset. This section covers all the background content required for a comprehensive understanding (PCA, factor models, covariance matrices etc). Section 4 then takes the concepts mentioned in section 3 and applies them to the dataset where we look at importing the dataset, cleaning it up by removing missing values accordingly and analysing the data to allow for conclusions to be made. Once all this is done we will have successfully taken the data, applied PCA to a factor model and developed a forecasting approach to a number of different variables which can then be used to draw meaningful conclusions from.

2 FRED-MD the Dataset

2.1 Origin of FRED-MD

As we saw in the previous section, big datasets in macroeconomics have undergone multiple iterations of development since the initial paper by Stock and Watson (1996) [2]. Despite all these advancements in research and analysis, there was still one main issue: the process of gathering data to create a dataset was often lengthy and complex, as there was no standardized way of collecting information. This required using multiple sources to compile a dataset. This issue was addressed in a working paper by Michael W. McCracken and Serena Ng (2015) [7]. If economic data were consistent, and everyone could universally agree on definitions and indicators, the analysis of economic data would be more streamlined and consistent. However, given

the ever-changing nature of definitions and relevant variables, this proved very challenging in practice.

In their 1996 research paper, Stock and Watson outlined four key criteria for selecting variables, one of which was that the data should have consistent historical definitions. If not, it should be possible to adjust the data. This created issues, as datasets often required revisions for various reasons. This motivated the creation of the FRED-MD dataset. Michael W. McCracken and Serena Ng collaborated with the Federal Reserve Bank of St. Louis to create a dataset that addresses this issue—a dataset that is consistently updated and serves as a standardized dataset for U.S. macroeconomics.

As described in their working paper, they created this dataset with three key objectives in mind:

1. It would be a publicly available, open-source dataset so that everyone could access the same data meeting the criteria set out in Stock and Watson (1996). This was important as other datasets were often behind paywalls or subscription models, which McCracken and Ng wanted to eliminate.

2. It would be updated on a timely basis, ensuring the data's reliability.

3. It would relieve researchers of the burden of updating data.

With these objectives in mind, McCracken and Ng compiled a dataset consisting of 134 monthly time series with similar coverage to the Stock and Watson dataset. The full list of economic indicators is provided in the appendix, along with the relevant transformations (Chapter 2.2.2). The standard database before FRED-MD was the Global Insights Basic Economics Database (GSI), which was not publicly available to external users. As a result, when users transitioned to the FRED-MD dataset, they had to adjust their variables to account for the absence of proprietary variables from GSI. The list of adjustments is provided in the appendix (see Figure 10). With this in place, the dataset was made public in 2015 and has been updated as McCracken and Ng intended, with the latest iteration available as recently as October 2024, including both monthly and quarterly data. [7]

2.2 Data Transformation

In this section, we discuss the transformations applied to the FRED-MD dataset and the necessity of transforming the data to prepare it for analysis.

2.2.1 Why Do We Need to Transform the Data?

When working with time series data, it is often advised to transform the data to make it easier to work with. Below are the main reasons why transforming data is essential, especially for analyzing macroeconomic data:

1. **Achieving stationarity**: For time series analysis, we want to ensure the data is stationary. A time series X_t is said to be stationary if the statistical characteristics of the time series (mean, variance, autocovariance, and autocorrelation) remain constant over time. Many time series models assume stationarity because it allows for consistent statistical inference. Non-stationary time series data often exhibit trends, seasonality, or other irregular patterns, which can yield inaccurate results. This is typically managed using transformation methods like differencing (first or second difference) or log transformation, both of which will be applied to the FRED-MD dataset.

2. **Reducing trends**: Certain economic indicators within the dataset will exhibit trend characteristics or exponential growth over time (e.g., GDP or population increases). Most

time series modeling aims to remove these trends, as removing them allows us to focus on irregular fluctuations within the data and capture the most critical factors in a model. Differencing (subtracting consecutive observations) is a common transformation for removing trends, allowing analysts to focus on fluctuations around a constant mean.

3. **Stabilizing variance**: A common feature of time series data is heteroscedasticity, where the variance changes over time. For example, financial data often shows significant fluctuations during market crashes or recessions. Log transformation is a common approach for addressing heteroscedasticity, as it reduces large variations in the time series data, helping the model perform better.

4. **Dealing with seasonality**: Economic data often exhibits seasonal trends. For example, retail sales peak in December. While seasonality can be modeled, it may be helpful to diminish the seasonal component, depending on the analysis context. Differencing is one approach that can help remove seasonal components from time series data.

Having covered why data transformation is critical for analysis, we will apply these methods to the FRED-MD dataset in the next section.

2.2.2 The Transformed Dataset

As we discussed in the previous section, we transform the data to make it easier to work with. Fortunately, in their working paper, Michael W. McCracken and Serena Ng (2015) [7] designated an appropriate transformation for each category: (1) Output and Income, (2) Labor Market, (3) Consumption and Orders, (4) Orders and Inventories, (5) Money and Credit, (6) Interest Rates and Exchange Rates, (7) Prices, and (8) Stock Market. They labeled each transformation from 1 to 7. The transformations are:

1. **No transformation**: For data that is already stationary, no transformation is applied, as it does not exhibit any trends.

2. **First Difference Transformation**: Denoted $\Delta x_t = x_t - x_{t-1}$, this transformation removes a linear trend in a time series by taking the difference between the current and previous observations.

3. **Second difference transformation**: Denoted $\Delta^2 x_t = \Delta(\Delta x_t)$, this transformation removes a quadratic trend when single differencing is insufficient to achieve stationarity. It is used for trends with accelerating or decelerating components.

4. **Log transformation**: Denoted $\log(x_t)$, this transformation stabilizes the variance of a time series. It is useful for data with significant fluctuations, such as crashes or spikes, where variance changes drastically over time.

5. **Log first difference transformation**: Denoted $\Delta \log(x_t) = \log(x_t) - \log(x_{t-1})$, this transformation combines log transformation and first differencing to address both variance and stationarity issues.

6. **Log second difference transformation**: Denoted $\Delta^2 \log(x_t)$, this transformation is suitable for complex trends and variance changes requiring both log transformation and second differencing.

7. **Growth rate transformation**: Calculated as $\frac{x_t}{x_{t-1}} - 1$, this transformation is commonly used for financial data to analyze growth rates and interpret percentage changes.

Since we are working with over 100 time series variables, the working paper has already designated the transformation to be applied to each series (see appendix). For the remainder of this report, we will use the transformed dataset which has been compiled into an R package available at (`https://github.com/cykbennie/fbi`). [7]

3 Methodology

3.1 Overview and Motivation

In the Methodology section, we will look at ways to transform, analyze and interpret a multivariate data set (such as the FRED-MD data set). Our objective is to leverage principal component analysis and factor model analysis to identify underlying latent factors that cause the majority of influence on our observations from the data set. These latent factors will then provide us information we can further use when producing forecasts for variables within the dataset.

Our motivation for using a procedure which includes both Principal Component Analysis (which we will often refer to as PCA) alongside a Factor Model Analysis becomes clear when we look into the characteristics of the dataset we are looking to analyse (The FRED-MD).

1. **High dimensionality**: The FRED-MD is a very large economic dataset which includes many correlated variables. PCA is a very effective first step we can take to reduce dimensionality by transforming the original indicators into a set of fewer orthogonal components. We will later see how we obtain these through eigenanalysis.

2. **Interpreting latent economic factors**: As the FRED-MD contains economic data we can make some assumptions that the variables will be frequently influenced by underlying, unobserved factors. For example business cycles or government policies are unobserved factors not shown in our dataset, but will majorly effect much of the variance within our variables. Therefore the factor model analysis allows us to interpret the principal components discovered by PCA above as significant latent factors, giving us more information when it comes to modeling and forecasting.

3.2 Principle Component Analysis

As we have explained how we have approached cleaning and transforming the data in section 2, in the methodology, it is assumed that all the data is usable.
We first need to define **covariance** and a **covariance matrix** and obtain these from our dataset:

3.2.1 Constructing The Covariance Matrix

The **covariance** between two random variables X and Y, denoted $\text{Cov}(X, Y)$, measures how much X and Y vary together. It is defined as:

$$\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$$

where $\mathbb{E}[X]$ and $\mathbb{E}[Y]$ are the expected values of X and Y, respectively.

If X and Y tend to increase or decrease together, the $\text{Cov}(X, Y)$ is positive. If one tends to increase when the other decreases, the $\text{Cov}(X, Y)$ is negative. A covariance of zero suggests that X and Y are linearly uncorrelated.

The **covariance matrix** of a random vector $X = (X_1, X_2, \ldots, X_n)'$ is a square matrix that represents the pairwise covariances between each pair of elements in X. Denoted Γ_X or Σ_X, it is defined as:

$$\mathbf{\Gamma_X} = \mathrm{Cov}(X) = \begin{bmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_n) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \cdots & \mathrm{Cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_n, X_1) & \mathrm{Cov}(X_n, X_2) & \cdots & \mathrm{Cov}(X_n, X_n) \end{bmatrix}$$

where each entry $\mathrm{Cov}(X_i, X_j)$ represents the covariance between X_i and X_j.

The diagonal entries $\mathrm{Cov}(X_i, X_i)$ are the variances of each variable X_i, while the off-diagonal entries $\mathrm{Cov}(X_i, X_j)$ represent the covariances between pairs of variables. The covariance matrix is symmetric, as $\mathrm{Cov}(X_i, X_j) = \mathrm{Cov}(X_j, X_i)$ for all i, j.
These definitions have been taken from [8] and slightly simplified to set ourselves up to perform the eigenanalysis required in PCA.

3.2.2 Eigenanalysis

Now we have defined our covariance matrix we can move to our PCA definition in which we include the covariance matrix from our observed data (over 130 variables in the case of the FRED-MD).

Principal Component Analysis (PCA) is a statistical method to reduce dimensionality that transforms an n-dimensional data vector (n being over 130 in our applications) $\mathbf{x}_t = (x_{1t}, \ldots, x_{nt})'$ into a smaller set of orthogonal components capturing the maximum variance.

1. **Covariance Matrix**: For a data vector $\mathbf{x}_t$ with zero mean, $\mathbb{E}[\mathbf{x}_t] = 0$, the covariance matrix is defined as:
$$\Gamma_x = \mathbb{E}[\mathbf{x}_t \mathbf{x}_t'].$$

2. **Eigenvalue Decomposition**: PCA involves the eigenvalue decomposition of Γ_x, expressed as:
$$\Gamma_x = PDP'$$
where $P = (\mathbf{p}_1, \ldots, \mathbf{p}_n)$ is an orthogonal matrix of eigenvectors, and $D = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ contains eigenvalues in descending order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n > 0$. Each eigenvector $\mathbf{p}_j$ maximizes the variance along its direction (a' is the transposition of the a vector):
$$\mathbf{p}_j = \arg\max_a \, a'\Gamma_x a \quad \text{s.t.} \quad a'a = 1 \quad \text{and} \quad a'\mathbf{p}_i = 0, \quad i < j.$$

3. **Principal Components**: The projection of x_t onto the first q eigenvectors (where $q < n$) produces the reduced representation:
$$\mathbf{x}_{t,[q]} = \sum_{j=1}^{q} (\mathbf{p}_j' \mathbf{x}_t) \mathbf{p}_j$$
which minimizes the residual variance, capturing the essential structure of the data with fewer dimensions.

This definition is a modified (more structural) version of Barigozzi's notes from [9] demonstrating the flow of our methodology from obtaining a covariance matrix into finding our fewer orthogonal vectors which capture the majority of our variance.

3.2.3 Dimensionality Reduction Outcome

In summary, after performing Principal Component Analysis (PCA), we decompose the data into two matrices:

- A matrix of size q, representing the selected principal components that capture the majority of the dataset's variance.
- A matrix of size $n - q$, containing the unused eigenvectors, representing residual variance.

To determine q, we examine the cumulative variance explained by the eigenvalues. The selection of q typically aims to retain around 80-90% of the total variance, which can be visualized with a scree plot or cumulative variance plot. This approach ensures that we capture the most informative components while minimizing dimensionality.

According to the definition of PCA, each eigenvalue λ_j represents the scale of the variance along its corresponding eigenvector $\mathbf{p}_j$. To verify that we have effectively chosen our constant q, we need to ensure that the variance remaining in the matrix of unused eigenvectors (of size $n - q$) is minimal. This will confirm that we have captured the majority of the variance within the selected principal components.

To quantify this, we calculate the trace of the residual covariance matrix, which consists of the unused eigenvalues $\lambda_{q+1}, \lambda_{q+2}, \ldots, \lambda_n$. The trace of the residual component can be expressed as:

$$\text{Residual Variance} = \sum_{j=q+1}^{n} \lambda_j$$

Since the trace of a matrix represents the sum of its eigenvalues, this residual trace represents the total variance not captured by the first q principal components. To ensure that the majority of variance is captured by the selected components, we aim for this residual variance to be small relative to the total variance in the data.

Let $\text{tr}(\Gamma_x)$ denote the trace of the full covariance matrix Γ_x, which is equal to the sum of all eigenvalues:

$$\text{tr}(\Gamma_x) = \sum_{j=1}^{n} \lambda_j$$

The variance captured by the principal components we selected is then:

$$\text{Variance Captured} = \sum_{j=1}^{q} \lambda_j$$

Thus, the residual variance, representing the variance left in the unused components, is:

$$\text{Residual Variance} = \sum_{j=q+1}^{n} \lambda_j = \text{tr}(\Gamma_x) - \sum_{j=1}^{q} \lambda_j$$

By minimizing this residual variance, we ensure that the selected principal components capture the majority of the data's variance.

This idea is also inspired by Barigozzi from [9] and simplified to suit both the style and motivations of our project.

Another important characteristic of PCA to note is that all our chosen eigenvalues to be principal components must be orthogonal to one another by definition. This therefore removes

any chance of variance being captured in the same direction as previous principal components. Therefore if we have one abnormally large eigenvalue, this variance won't overpower the other eigenvectors that need to be considered. This is shown numerically by the following definition:

Let $\mathbf{p}_i$ and $\mathbf{p}_j$ be two eigenvectors of the covariance matrix Γ_x, corresponding to distinct eigenvalues λ_i and λ_j, respectively. The eigenvectors are orthogonal if:

$$\mathbf{p}_i'\mathbf{p}_j = 0, \quad \text{for} \quad i \neq j.$$

This property holds because the covariance matrix Γ_x is symmetric, and thus its eigenvectors associated with distinct eigenvalues are orthogonal.

We now only have to deal with q chosen variables going into the next stage of our methodology, Factor Model Analysis, meaning the calculation will be manually and computationally easier to apply to our dataset. We, hopefully, also know that we have still captured the majority of the dataset's variance by minimizing the trace of the residual matrix which represents the total variance.

3.3 Factor Model Analysis

I am going to start our section on Factor Model Analysis with a simple yet effective example I found in [10]. It is referenced in the book to (Spearman, 1904) an important early paper in dealing with factor analysis. The example deals with examination results from 3 different subjects, in Spearman's case, Classics (x_1), French (x_2) and English (x_3). He defined the correlation matrix by:

$$\begin{bmatrix} 1 & 0.83 & 0.78 \\ 0.83 & 1 & 0.67 \\ 0.78 & 0.67 & 1 \end{bmatrix}$$

The dimensionality of this matrix can effectively be reduced from 3 to 1 by expressing the three variables as equations:

$$\begin{aligned} x_1 &= \lambda_1 f + u_1, \\ x_2 &= \lambda_2 f + u_2, \\ x_3 &= \lambda_3 f + u_3, \end{aligned}$$

here f is stated as the **underlying common factor** and in each case λ is known as a **factor loading**. The terms u_1, u_2 and u_3 represent random error terms.

This minimalistic example purely demonstrates that we are looking for an **underlying (or latent) factor** that we do not observe through our data. In this case that factor is f, which in this context might represent a 'general intelligence', with the loadings shown.

3.3.1 Static factor model

In a static factor model, each observed variable vector $\mathbf{x}_t$ at time t depends only on contemporaneous latent factors. This can be represented as:

$$\mathbf{x}_t = \mathbf{\Lambda}\mathbf{f}_t + \xi_t,$$

where $\mathbf{f}_t$ is the common factor vector at time t, $\mathbf{\Lambda}$ is the matrix of factor loadings, and ξ_t is the vector of error terms. Here, the influence of $\mathbf{f}_t$ is immediate and does not carry over to future observations.

In a dynamic factor model, the latent factors can affect observed variables over multiple time periods. For example, the dynamic model might be represented as:

$$\mathbf{x}_t = \mathbf{\Lambda}_0 \mathbf{f}_t + \mathbf{\Lambda}_1 \mathbf{f}_{t-1} + \cdots + \mathbf{\Lambda}_k \mathbf{f}_{t-k} + \xi_t,$$

where $\mathbf{f}_{t-1}, \mathbf{f}_{t-2}, \ldots, \mathbf{f}_{t-k}$ are lagged factor vectors, capturing the delayed effects of previous time periods on $\mathbf{x}_t$. A lot of our info about dynamic models now and further into the methodology section are taken from [11] as well as [9].

For our analysis of the FRED-MD dataset, we choose a static model because it provides a simpler, more computationally efficient approach that captures the main sources of variability without requiring the complex estimation of lagged effects. This is appropriate for an initial analysis, where the goal is to identify the dominant factors affecting economic indicators without delving into time-dependent dynamics.

To analyze the FRED-MD dataset using a static factor model, we represent the observed data as a linear combination of a few latent factors that capture the main variability in the data. Let $\mathbf{x}_t$ denote the n-dimensional vector of observed variables at time t, where n represents the total number of economic indicators in the dataset (e.g., GDP, inflation, employment rates).

In this framework, each observed variable x_{it} (the i-th element of $\mathbf{x}_t$) is expressed as a function of r common factors, denoted $\mathbf{f}_t = (f_{1t}, f_{2t}, \ldots, f_{rt})'$, where $r \ll n$. The static factor model is then given by:

$$\mathbf{x}_t = \mathbf{\Lambda} \mathbf{f}_t + \xi_t,$$

where:

- $\mathbf{\Lambda}$ is an $n \times r$ matrix of factor loadings, with each entry λ_{ij} representing the loading of the i-th observed variable on the j-th factor,
- $\mathbf{f}_t$ is an r-dimensional vector of latent factors at time t,
- ξ_t is an n-dimensional vector of idiosyncratic errors, assumed to have mean zero and be uncorrelated with $\mathbf{f}_t$.

In this setup, the static factor model captures the primary sources of variability in the economic indicators within the FRED-MD dataset, providing a simplified structure that is computationally efficient and avoids the complexity of incorporating time-dependent effects.

3.3.2 Estimation of Factors and Factor Loadings

Once we have performed PCA on the FRED-MD dataset and selected the q principal components that capture the majority of the variance, we proceed to estimate the factors $\mathbf{f}_t$ and the factor loadings $\mathbf{\Lambda}$ in our static factor model.

The steps for estimation are as follows:

- **Determine the Factor Loadings $\mathbf{\Lambda}$:** The factor loading matrix $\mathbf{\Lambda}$ is estimated by using the first q eigenvectors of the covariance matrix Γ_x, where q is chosen based on the cumulative variance captured by the principal components. Let $\mathbf{P}_{[q]}$ be the matrix formed by these top q eigenvectors. Then:

  $$\mathbf{\Lambda} = \mathbf{P}_{[q]}$$

  where each column of $\mathbf{\Lambda}$ corresponds to one of the selected eigenvectors that defines a principal component.

- **Estimate the Factors $\hat{\mathbf{f}}_t$:** The factors $\mathbf{f}_t$ for each time period t are estimated by projecting the observed data $\mathbf{x}_t$ onto the space defined by the factor loadings $\mathbf{\Lambda}$. This projection is given by:

$$\hat{\mathbf{f}}_t = \hat{\mathbf{\Lambda}}'\mathbf{x}_t$$

  where $\hat{\mathbf{\Lambda}}'$ is the transpose of the sample loading matrix, allowing us to obtain the factor scores for each time period.

- **Verify Variance Capture:** To confirm that the factors $\hat{\mathbf{f}}_t$ and loadings $\mathbf{\Lambda}$ effectively capture the majority of the variance, we ensure that the residual variance in the excluded components remains minimal. The residual variance, represented by the trace of the unused portion of the covariance matrix, should be small relative to the total variance. This ensures that our selected factors account for most of the dataset's variability.

In summary, using PCA allows us to estimate a lower-dimensional representation of the FRED-MD dataset by deriving the factor loadings and corresponding factor scores, enabling us to efficiently capture the key underlying factors driving variability in the economic indicators.

In relation to the example I included from earlier from [10] which is referenced to (Spearman, 1904), I am going to show how it would look to solve the simple problem we were faced with:

To illustrate the estimation of a factor in a simple static factor model, we use Spearman's (1904) example of examination scores in three subjects: Classics (x_1), French (x_2), and English (x_3). The covariance matrix for these subjects is given by:

$$\mathbf{\Gamma} = \begin{bmatrix} 1 & 0.83 & 0.78 \\ 0.83 & 1 & 0.67 \\ 0.78 & 0.67 & 1 \end{bmatrix}$$

Assuming a single underlying factor f influences all three subjects, we represent each score as:

$$x_i = \lambda_i f + u_i$$

where λ_i are the factor loadings. Given the covariance values, we set up the following equations:

$$\lambda_1\lambda_2 = 0.83, \quad \lambda_1\lambda_3 = 0.78, \quad \lambda_2\lambda_3 = 0.67$$

Solving these equations: 1. Letting $\lambda_1 \approx 0.983$, we find $\lambda_2 = \frac{0.83}{0.983} \approx 0.844$ and $\lambda_3 = \frac{0.78}{0.983} \approx 0.793$. 2. Thus, the estimated factor loadings are $\lambda_1 \approx 0.983$, $\lambda_2 \approx 0.844$, $\lambda_3 \approx 0.793$.

To estimate the factor f for a given set of scores, we calculate:

$$\hat{f} = \frac{\lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}$$

For example, if a student's scores are $x_1 = 85$, $x_2 = 90$, and $x_3 = 88$, then:

$$\hat{f} = \frac{(0.983 \cdot 85) + (0.844 \cdot 90) + (0.793 \cdot 88)}{0.983^2 + 0.844^2 + 0.793^2} \approx \frac{229.299}{2.308} \approx 99.36$$

Thus, $\hat{f} \approx 99.36$ represents the estimated underlying factor, indicating the general proficiency influencing performance across all subjects.

3.3.3 Estimation of Factors in a Static Factor Model

In a static factor model, the dataset X is represented as:

$$X = BF + E,$$

where:

- B is the $n \times q$ **factor loading matrix**, representing the influence of each factor on the observed variables,
- F is the $q \times T$ **factor matrix** (latent factors),
- E is the $n \times T$ **error matrix**, representing unique, uncorrelated errors for each variable.

The factor loading matrix B is obtained using **Principal Component Analysis (PCA)**, which identifies the principal components (columns of B). Once these loadings are identified, the latent factors F are estimated by projecting the observed data X onto the space defined by B.

Estimation of Factors: The factors F are estimated using the least-squares solution:

$$F = B'X,$$

where:

- B' is the transpose of the factor loading matrix B,
- F is the $q \times T$ matrix of factor scores for the dataset.

At the observation level, for any t-th time point, the model can be expressed as:

$$\mathbf{x}_t = \mathbf{\Lambda}\mathbf{f}_t + \xi_t,$$

where:

- $\mathbf{x}_t$ is the n-dimensional vector of observed variables at time t,
- $\mathbf{\Lambda}$ (equivalent to B) is the $n \times q$ matrix of factor loadings,
- $\mathbf{f}_t$ (a column of F) is the q-dimensional vector of latent factors,
- ξ_t (a column of E) is the n-dimensional vector of residuals.

The least-squares solution for $\mathbf{f}_t$ at time t can also be written as:

$$\mathbf{f}_t = (\mathbf{\Lambda}'\mathbf{\Lambda})^{-1}\mathbf{\Lambda}'\mathbf{x}_t.$$

Residuals and Verification: The residual matrix E is computed as:

$$E = X - BF.$$

To verify the adequacy of the factors, the residual variance (trace of EE') should be small relative to the total variance of X. This ensures that the selected factors F explain the majority of the variability in the observed data.

Interpretation of Factors: The matrix F contains the scores for the q latent factors over T time points. These factors summarize the underlying structure of the data, capturing the main sources of variability identified by PCA. The estimated residuals E should be uncorrelated with the factors, confirming that the majority of variance in X is captured by B and F.

In summary, the factors F are estimated by projecting the observed data onto the factor loadings obtained from PCA, providing a compact representation of the underlying latent structure while minimizing the residual variance.

3.3.4 Dynamic Factor Models (Used for forecasting in application)

Factor models are generally classified as **static** or **dynamic**, depending on whether they incorporate time-dependent (lagged) relationships between factors.

A dynamic factor model introduces lagged terms, allowing the factors to influence not only contemporaneous observations but also observations in future periods. This is crucial for datasets like FRED-MD, where time dependencies are expected across economic indicators.

The dynamic factor model can be written as:

$$\mathbf{x_t} = B(L)\mathbf{f_t} + \mathbf{e_t}$$

where:

- $B(L)$ is a matrix polynomial in the lag operator L,
- f_t is now a vector of dynamic factors that includes both contemporaneous and lagged effects.

Expanded, this model becomes:

$$\mathbf{x_t} = B_0\mathbf{f_t} + B_1\mathbf{f_{t-1}} + B_2\mathbf{f_{t-2}} + \cdots + \mathbf{e_t}$$

where $B_0, B_1, B_2, \ldots$ represent loading matrices for each lag of $\mathbf{f_t}$. In this setup, the latent factors $\mathbf{f_t}$ have persistent effects over time, capturing the time-series structure and inter-temporal dependencies among variables.

3.3.5 Approximate Dynamic Factor Model (Used for forecasting in application)

To capture the temporal dependencies in the FRED-MD dataset, we have seen that it is best to adopt an **Approximate Dynamic Factor Model (ADFM)** with **Principal Component (PC) estimation** due to the number of variables we are dealing with. This model leverages PC estimation to derive latent factors while allowing these factors to follow a dynamic (lagged) structure, addressing both dimensionality reduction and time dependence (two important characteristics of the FRED-MD).

Given an observed data matrix X of size $n \times T$, where n is the number of variables and T is the number of time points, we assume that X can be decomposed as:

$$X_t = B(L)F_t + e_t$$

where:

- $B(L) = B_0 + B_1L + B_2L^2 + \ldots$ is a lag polynomial in the lag operator L, representing loadings for current and lagged factors,
- F_t is the q-dimensional latent factor vector, and
- e_t is the error term with weak cross-sectional dependence.

In an approximate factor model, the common component $X_t - e_t$ is the main source of variance in X_t, with F_t representing latent factors across time.

1. Principal Component Estimation of Factors

To estimate F_t, we use PC estimation, leveraging the eigenvalues and eigenvectors of the covariance matrix to identify directions of maximum variance.

Construct the Covariance Matrix: Calculate the sample covariance matrix of X:

$$\Gamma_X = \frac{1}{T} XX'$$

Eigenvalue Decomposition: Decompose Γ_X using the eigenvalue decomposition we saw in 3.1:

$$\Gamma_X = PDP'$$

where:

- P is an $n \times n$ matrix whose columns are the eigenvectors of Γ_X,
- $D = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is a diagonal matrix of eigenvalues in descending order.

Selecting Factors: Select the first q eigenvectors (columns of P) corresponding to the largest q eigenvalues. This gives us the loading matrix $B_q = P_q$, where P_q is the $n \times q$ matrix of the first q eigenvectors.

Estimating Factors: Project X onto P_q to obtain factor estimates $\hat{F}$:

$$\hat{F} = P_q' X$$

where $\hat{F}$ is a $q \times T$ matrix containing the estimated latent factors at each time point.

4 Application to the Dataset

4.1 Data Preprocessing

Before we are ready to start with our data analysis, we must make sure the data is suitable for our task. We want to make sure the data is convenient to use for any data manipulations or data visualisations that we may want to show in our analysis. This process may also be known to some, in other words, as "Data Cleaning".

However, before we start 'cleaning' our data, we need to load it first:

4.1.1 Initial Data Loading

The FRED-MD [12] dataset was loaded using the `fbi` [13] package in R, using the following code:

```r
#Loading required libraries
library(fbi)

#Loading FRED-MD data with TCODE transformations applied
data <- fredmd(
  file = "current.csv",
  date_start = NULL,
  date_end = NULL,       #= NULL so we have the full dataset
  transform = TRUE       #Applying TCODE transformations
)

#Printing data dimensions
cat("Initial dimensions:", dim(data), "\n")
```

Listing 1: Loading the Dataset

```
Initial dimensions: 787 127
```

Code Output

By setting `transform = TRUE`, the `fredmd()` function transforms the dataset by variable per their respective transformation codes (2.2.2, See appendix)

We may notice the code output, `Initial dimensions: 787 127`. This tells us the dataset currently has 787 columns and 127 rows, in other words, 787 time-steps and 127 variables.

4.1.2 Missing Value Analysis

Here, we will be identifying the "gaps" in our dataset. Missing value analysis is a critical step in data preprocessing as it improves the robustness, reliability and accuracy of any analysis or models built on that data.

We will identify how much data we have missing, then replace the identified missing data, withholding the integrity of the data, in a process known as data imputation.

4.1.3 Missingness by Year

First, we tackle this problem by identifying how much of our data is missing (data missingness), grouped by year. We have the following function `check_missing_by_year` that will take in the dataset as an input and return data missingness by year:

```
check_missing_by_year <- function(data) {
  years <- format(data$date, "%Y")
  #Columns: Year, Missing_Count, Total_Possible, and Missing_Percentage
  missing_by_year <- data.frame(
    Year = sort(unique(years)),      #Unique years sorted in ascending order
    Missing_Count = NA,              #Placeholder for count of missing values
    Total_Possible = NA,             #Placeholder for total possible values
    Missing_Percentage = NA          #Placeholder for percentage of missing
    values
  )

  #Loop over each year/Check every year
  for(year in missing_by_year$Year) {
    year_data <- data[years == year, -1] #'-1' to exclude the date column
    total_cells <- nrow(year_data) * ncol(year_data) #Calculating total no. of
    cells for year
    missing_cells <- sum(is.na(year_data)) #Calculating the no. of missing
    cells for year
    missing_by_year[missing_by_year$Year == year,
      c("Missing_Count", "Total_Possible", "Missing_Percentage")] <-
      c(missing_cells, total_cells,
        round((missing_cells / total_cells) * 100, 2))
  }

  #Print the result
  return(missing_by_year)
}
```

Listing 2: Missing Values (by Year) Function

We choose to trim the dataset and begin from 1960 onwards due to significant missing values in 1959, as evident in the code output (16.87% vs 3.97% in later years).

```
#Based on analysis, define new variable that starts from 1960
data_1960 <- data[data$date >= "1960-01-01", ]
```

Listing 3: Trimming our dataset

Data quality is paramount for any type of data analysis. This initial trimming will be the first of many data pre-processing steps we will take to ensure optimal data quality and consequently, robust results. The principle of GIGO [14] will be particularly important for PCA, and the focus on high data quality will be seen numerous times in the following steps.

Year	Missing Count	Total Possible	Missing Percentage
1959	255	1512	16.87
1960	60	1512	3.97
1961	60	1512	3.97
1962	54	1512	3.57
1963	48	1512	3.17
1964	48	1512	3.17
1965	48	1512	3.17
1966	48	1512	3.17
1967	48	1512	3.17
1968	38	1512	2.51

Table 1: Code Output: Missing Data Summary by Year

4.1.4 Missingness by Variable

Now, we will do the same as we did above, but we will check for data missingness for each variable instead, using function `check_missing_by_variable`:

```r
check_missing_by_variable <- function(data) {
  data_no_date <- data[, -1] #'-1' to exclude the date column
  missing_pct <- colMeans(is.na(data_no_date)) * 100

  missing_df <- data.frame(
    Variable = names(missing_pct),
    Missing_Percentage = round(missing_pct, 2)
  )

  return(missing_df[order(-missing_df$Missing_Percentage), ])
}

#Analysee missing values by variable
var_missing <- check_missing_by_variable(data_1960)

#Identifying variables with >1% missing values
high_missing <- var_missing[var_missing$Missing_Percentage > 1, ]
print("Variables with >1% missing values:")
print(high_missing)
```

Listing 4: Missing Values (by Variable) Function

Variable	Missing Percentage
ACOGNO	49.94
UMCSENTx	28.00
TWEXAFEGSMTHx	20.26
ANDENOx	12.65
VIXCLSx	3.87

Table 2: Code Output: Missing Data Percentage by Variable

We see that variables `ACOGNO, UMCSENTx, TWEXAFEGSMTHx, ANDENOx, VIXCLSx` have more than 1% missingness, now these variables will be trimmed from our dataset, using:
(continued next page)

```r
#Remove high-missingness (>1%) variables
```

```
vars_to_remove <- high_missing$Variable
data_filtered <- data_1960[, !(names(data_1960) %in% vars_to_remove)] #Defining
    new variable to store trimmed dataset
```

Listing 5: Removing variables with more than 1% missingness

4.1.5 Data Imputation Process

We have now removed all variables with high data missingness, now we will move on to dealing with the remaining variables with under 1% missingness. Instead of completely eliminating the variable from our dataset (which will hurt our data integrity), we will replace the missing data with a statistical method we describe below. Data imputation is better suited to these variables due to them having low data missingness.

We will employ the Tall-Project[15] method for imputation, implemented through the `fbi R` package. According to Bai and Ng (2021)[15], this method is particularly suitable for large panel data like FRED-MD.

4.1.6 Theoretical Basis of Imputation (to confirm)

The Tall-Project method uses an iterative approach based on principal components. For a matrix X with missing values, it follows these steps:

1. Initialise missing values with unconditional means based on non-missing values
2. Extract principal components using eigenanalysis of the variance-covariance matrix
3. Project data onto these components to obtain factor estimates
4. Update missing values with projections from factor estimates
5. Iterate until convergence

The mathematical representation of this process is:

$$X_t = B(L)F_t + e_t$$

where $B(L)$ represents the factor loadings, F_t the factors, and e_t the error term.

4.1.7 Implementation

We implemented the Tall-Project method using the `tp_apc` function:

```
data_matrix <- as.matrix(data_filtered[, -1])  #No data column
mode(data_matrix) <- "numeric"
imputed_data <- tp_apc(data_matrix,  #Applying the tp_apc function from fbi
kmax = 2)                            #Maximum no. of factors
```

Listing 6: Factor-Based Imputation Implementation

Here, `kmax = 2` specifies the maximum number of factors to be used in the imputation process. This parameter choice balances between capturing sufficient data structure while avoiding overfitting.

This approach is better suited to our task compared to naive methods. Rather than treating each variable independently, it rather considers the relationships between variables when replacing missing data, potentially leading to more realistic data imputations.

4.1.8 Verifying Imputation Results

After imputation, we replaced the missing values in our original dataset with the new imputed values:

```
data_imputed <- data_filtered
data_imputed[, -1][is.na(data_filtered[, -1])] <- #Assigning all missing data
    imputed_data$Chat[is.na(data_matrix)]
```

Listing 7: Replacing Missing Values

To make sure our imputation ran successfully, we do a final `cat` (print) statement:

```
cat("Missing values after imputation:",  #Printing number of missing data/cells
    sum(is.na(data_imputed)), "\n")
```

Listing 8: Verification of Imputation

```
Missing values after imputation: 0
```

Code Output

The code output confirms we have no remaning missing data. Imputation is finished and we may move on to our next steps.

4.1.9 Dataset Versioning

Unrelated to our analysis but, to maintain good programming habits, avoid confusion with dataset naming and alleviate potential comparisons, we stored our versions of our dataset in a 'dictionary' for easy interpretation:

- Raw data: The original dataset as loaded
- 1960 onwards data: Dataset trimmed to start from 1960
- Filtered data: Dataset after removing problematic variables
- Imputed data: Final dataset with all missing values imputed

```
datasets <- list(
    raw = data,
    data_1960 = data_1960,
    filtered = data_filtered,
    imputed = data_imputed  #We are currently using this dataset
)
```

Listing 9: Storing Dataset Versions

With a fully imputed dataset, we have now completed the data cleaning section of our application. We may now move onto our final step of our data preprocessing.

4.1.10 Data Standardisation

Before heading to PCA, we need to satisfy a pre-requisite of standardisation across all our variables. This will ensure they are on comparable scales and satisfy the zero mean assumption outlined in Section (3.2.2). As highlighted in the methodology:

> "For a data vector $\mathbf{x}_t$ with zero mean, $\mathbb{E}[\mathbf{x}_t] = 0$, the covariance matrix is defined as $\Gamma_x = \mathbb{E}[\mathbf{x}_t \mathbf{x}_t']$."3.2.2

This step is crucial as the FRED-MD dataset contains variables measured in different units and scales (for example, interest rates are in percentages, production indices are in levels).

4.1.11 Implementation

```r
standardize_data <- function(data) {
   date_col <- data$date
   data_no_date <- data[, !names(data) %in% c("date")]

   #Center and scale the data
   data_standardized <- scale(data_no_date, #Using the scale() function to
    standardize
                              center = TRUE, #Removing mean
                              scale = TRUE)  #Dividing by standard deviation

   #Assinging standardized data to updated variable
   data_standardized <- as.data.frame(data_standardized)
   data_standardized <- cbind(date = date_col,
                              data_standardized)

   return(data_standardized)
}
```

Listing 10: Standardisation Process

Here, our main actor is the pre-defined `scale()` function. The `scale()` function standardises data by subtracting mean (centering) and dividing by its standard deviation (scaling). This should result in the dataset having a mean of 0 and standard deviation of 1. We verify this on the next step:

4.1.12 Verification

```r
verify_standardization <- function(data) {
   data_numeric <- data[, !names(data) %in% c("date")]
   means <- colMeans(data_numeric)
   sds <- apply(data_numeric, 2, sd)

   cat("Mean of means:", mean(means), "\n") #Printing updated mean
   cat("Mean of SDs:", mean(sds), "\n")     #Printing updated std. deviation
}
```

Listing 11: Verification Checks

```
Verification of Standardization:
Mean of means: 2.609659e-17
Mean of SDs: 1
```

Code Output

Running these checks gave us exactly what we wanted:

- The mean came out to `2.61e-17` $\approx$ `0`
- The standard deviations are `1.00`

The results tell us the data has successfully been standardised. By standardising, we've ensured that each economic indicator, whether it's GDP or unemployment rates, will get a fair influence in our overall analyses.

With our data now properly standardised, we're ready to construct our covariance matrix and begin extracting those principal components.

4.2 Principal Component Analysis

4.2.1 Implementation

Having prepared our standardised dataset, we now implement PCA following the steps outlined in our methodology. This implementation directly maps to the mathematical foundations we have already established in earlier sections. We implement our PCA using the following `perform_pca` function:

```
perform_pca <- function(X) {
   # Step 1: Matrix preparation
   # Maps to  x  = E[xtxt'] from methodology
   X_centered <- scale(X, center = TRUE, scale = FALSE)
   Gamma_x <- (t(X_centered) %*% X_centered) / (nrow(X) - 1)

   # Step 2: Eigendecomposition
   # Maps to  x  = PDP' decomposition
   eigen_decomp <- eigen(Gamma_x)
   eigenvalues <- eigen_decomp$values
   eigenvectors <- eigen_decomp$vectors

   # Step 3: Variance calculations
   # Implements variance proportion calculations
   total_variance <- sum(eigenvalues)
   prop_var <- eigenvalues / total_variance
   cum_var <- cumsum(prop_var)

   # Step 4: Principal component calculation
   # Maps to V = A'X transformation
   components <- X_centered %*% eigenvectors
}
```

Listing 12: PCA Implementation

4.2.2 Mathematical Decomposition

Let us examine each step of this implementation in detail:

1. **Covariance Matrix Construction**:

$$\Gamma_x = \frac{1}{T-1}\sum_{t=1}^{T}(x_t - \bar{x})(x_t - \bar{x})'$$

where T is our sample size and x_t represents our centered observations.

2. **Eigendecomposition**:

$$\Gamma_x = PDP'$$

where: - P contains our eigenvectors - D is a diagonal matrix of eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$

3. **Variance Proportions**:

$$\text{Proportion}_i = \frac{\lambda_i}{\sum_{j=1}^{n}\lambda_j}$$

4.2.3 Analysis of Results

We may plot our results in the form of charts to better visualise our principle components. We use the `ggplot`[16] library to implement graphs in this analysis:

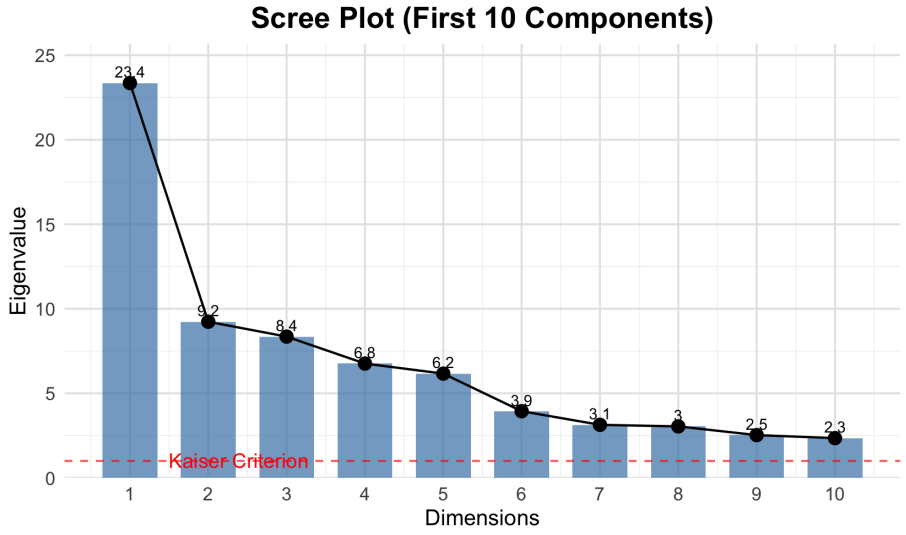

Figure 1: Scree Plot (First 10 Components)

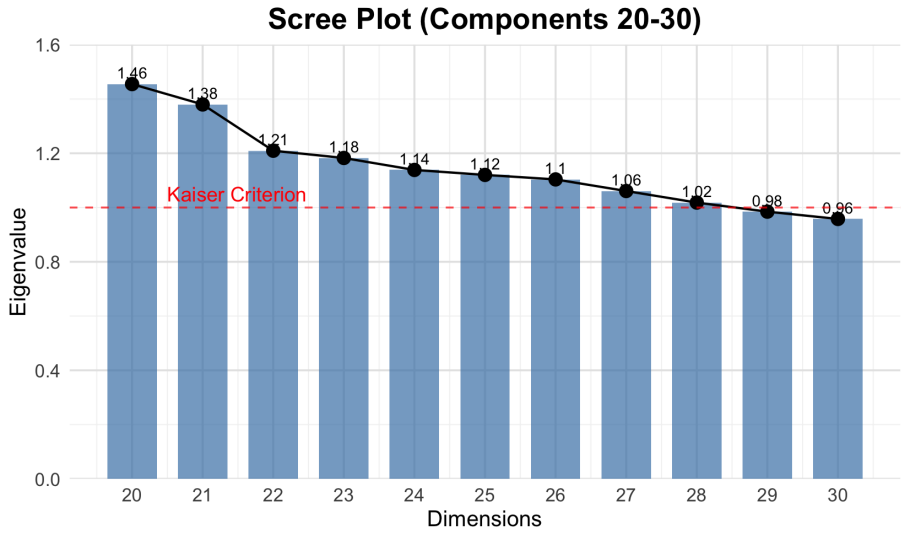

Figure 2: Scree Plot (Components 20-30)

The two scree plots (Figures 1 and 2) provide inferences regarding our principal components structure. In Figure 1, the first principal component exhibits a high eigenvalue of `23.4`, meaning substantially contributes in explaining the dataset's overall variance compared to other variables. The following components show a steep decline, with eigenvalues of 9.2, 8.4, 6.8 and 6.2 for principle components 2, 3, 4 and 5 respectively.

The Kaiser criterion[12], which we can see on the graphs as the red dashed line at eigenvalue = 1, is a criteria which helps us determine the final number of principle components we may choose to arrive at. This criterion suggests retaining components with eigenvalues greater than 1, due to these components explaining more variance than a single original variable would in the standardised dataset. In Figure 2, we observe that eigenvalues remain above the Kaiser criterion up until component 29 (eigenvalue = 0.98), suggesting potential value in variance of components up until component number 29. We should remember this result as this will aid us

in confluence with deciding the final number of components further on in this section.

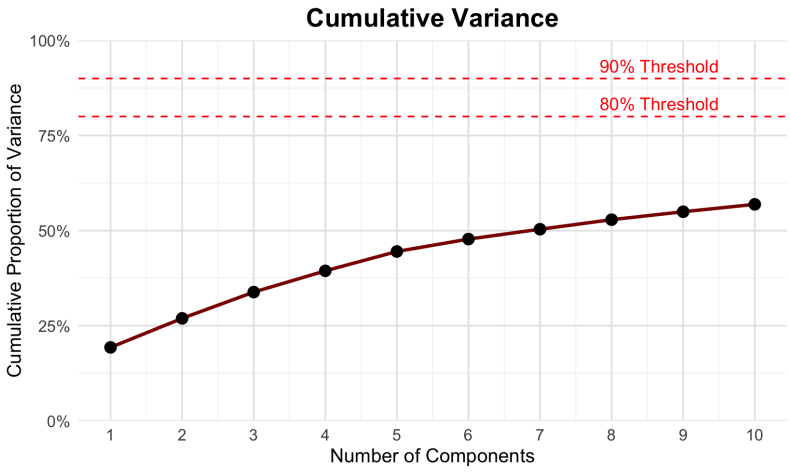

Figure 3: Cumulative Variance (First 10 Components)

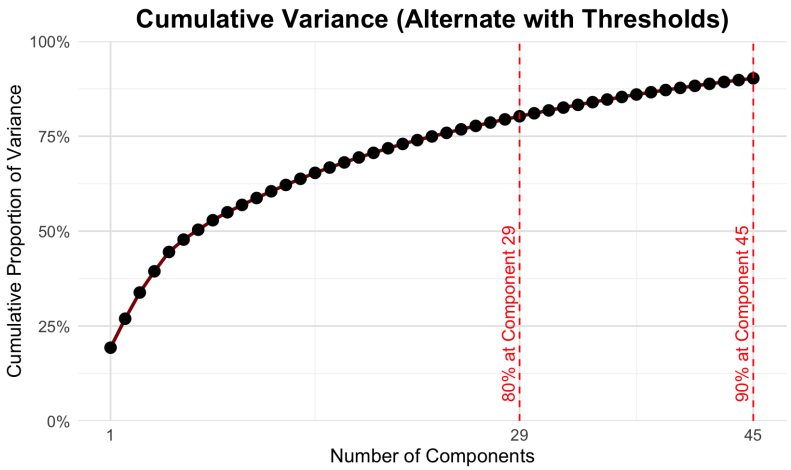

Figure 4: Cumulative Variance (Overall)

The cumulative variance analysis reveals the results of two critical thresholds:

- 80% Threshold: Achieved at component 29, indicating that the first 29 principal components collectively explain 80% of the total variance in our dataset.
- 90% Threshold: Achieved at component 45, suggesting that capturing the next 10% of variance demands a 16 more principle components.

Balancing these considerations - the Kaiser criterion and variance thresholds, along with practicality of our analysis - we may select 29 principal components for our subsequent analysis. This is supported by:

- Components 1-29 collectively capture 80% of total variance
- All components up to 29 remain close to or above the Kaiser criterion

- The marginal contribution of additional components becomes negligible, as evidenced by the flattening curve in Figure 4

This selection significantly reduces the dataset's dimensions while keeping most of the important information, narrowing down from our initial 127 variables to just 29, while still maintaining our dataset's information.

4.3 Factor Analysis

Continuing on from our PCA results, we may now explore these easier-to-use reduced dimensions to understand the underlying factors of the FRED-MD dataset. While PCA provided us with a mathematical way to reduce our data's dimensionality; capturing 80.26% of variance with 29 components - factor analysis will allow us to interpret these components and examine their forecasting powers and whether or not they give meaningful relation to economic drivers.

The relationship between our observed variables and potential underlying factors can be initially expressed through a simple static model:

$$\mathbf{X_t} = \mathbf{\Lambda F_t} + \epsilon_\mathbf{t} \tag{1}$$

Here, X_t is our variables observed at some time t, F_t is the vector of latent factors, Λ is the factor loading matrix, and ϵ_t captures the noise (or variations that are unexplained by our model). This gives us a starting point to decipher any meaningful relationships in the data, though in the latter sections we may incorporate dynamic relationships if they might offer additional insights.

Our analysis begins with this straightforward approach implementation:

4.3.1 Static Factor Model Implementation

Our first step is implementing a static factor model for the FRED-MD dataset. This approach lets us identify underlying patterns in the data before considering more complex dynamic relationships. The implementation starts with a fundamental preprocessing step - standardizing our variables to ensure they're comparable:

```
#Static Factor Model
basic_factor_model <- function(X, n_factors) {
    #Following methodology Section 3.3:
    #Static model: x_t = B f_t + e_t

    if("date" %in% colnames(X)) X <- X[, !colnames(X) %in% "date"] #Removing
    date

    #Standardize data to mean 0
    #Note: Here we don't scale variance to preserve relative importance
    X_centered <- scale(X, center = TRUE, scale = FALSE)

    #Extract factors using SVD
    #This decomposes our data into orthogonal components
    svd_result <- svd(X_centered)
    loadings <- svd_result$v[, 1:n_factors]  # Factor loadings (B)
    factors <- X_centered %*% loadings        # Factor scores (f_t)

    #Calculate how well our factors reproduce the data
    reconstructed <- factors %*% t(loadings)
    residuals <- X_centered - reconstructed

    #Overall model fit R^2 check
    R_squared <- 1 - sum(residuals^2) / sum(X_centered^2)

```

```
    return(list(
        factors = factors,
        loadings = loadings,
        residuals = residuals,
        R_squared = R_squared,
        reconstructed = reconstructed + attr(X_centered, "scaled:center")
    ))
}
```

Listing 13: Static Factor Model Implementation

The code implements our static factor model in several key steps. First, we center our data around zero, which ensures our factors capture variations rather than level differences. We then use singular value decomposition (SVD) - a statistical technique that allows us to simplify complex data by identifying overall trends or directions in which the data varies towards. This allows us to find key trends in large data sets such as FRED-MD. The main directions can be considered as our factors and their magnitude/significance will be captured in the loading matrix, which we can use the to find out how much each factor contributes to explaining variance.

When we apply this model to our standardized FRED-MD dataset:

```
factor_results <- implement_factors(
    datasets$standardized[, !names(datasets$standardized) %in% c("date")],
    n_factors
)

#Summary statistics
cat("\nFactor Model Results:\n")
cat("Total variance explained:",
    round(sum(factor_results$variance_explained) * 100, 2), "%\n")
cat("\nContribution by factor:\n")
for(i in 1:n_factors) {
    cat(sprintf("Factor %d: %.2f%%\n",
        i, factor_results$variance_explained[i] * 100))
}
```

Listing 14: Factor Analysis Results

The results reveal a hierarchical structure in our factors:

```
Factor Model Results:
Total variance explained: 80.26%

Contribution by factor:
Factor 1: 19.30%  # Dominant economic force
Factor 2: 7.63%   # Secondary effect
Factor 3: 6.90%   # Tertiary effect
Factor 4: 5.59%   #
Factor 5: 5.09%   # Diminishing but still significant
...
```

Code Output

These results could be a projection of interesting economic activity. Our first factor, which explains 19.30% of all variation, may be representing broad economic activity - such as GDP growth. This makes intuitive sense; when the economy grows or recedes, it affects other smaller factors. The second factor, with 7.63%, may be capturing financial conditions or monetary policy effects - which are important but not as profound as general economic growth.

The declining pattern of variance explained (from 19.30% to 5.09% for the first five factors) suggests we have successfully identified key distinct economic forces, from major drivers to niche influences. This aligns with economic intuition about few big events - such as monetary policy

have a profound impact, while many smaller forces such as industry-specific trends affect only certain, smaller variables.

What is particularly impressive is that with just 29 factors, we have managed to capture over 80% of all variation in our 127 economic variables. This dimension reduction suggests that, despite the overall complexity of the economy itself, a relatively few number of underlying factors drive most economic movements.

To visualize the factor loadings, we create a heatmap showing the relationship between factors and variables:

```
#Creating visualization of factor loadings
#Filtering data to show only high loadings
loadings_matrix <- factor_results$loadings[, -1]  # Remove Variable column
loadings_melted <- melt(as.matrix(loadings_matrix))
colnames(loadings_melted) <- c("Variable", "Factor", "Loading")

#Filtering loadings by threshold to avoid overlap and clutter
loadings_filtered <- loadings_melted %>%
  filter(abs(Loading) > 0.1) #0.1 minimum threshold

#Creating heatmap
ggplot(loadings_filtered, aes(x = Factor, y = Variable, fill = Loading))
#Remaining ggplot code continues...
.
.
```

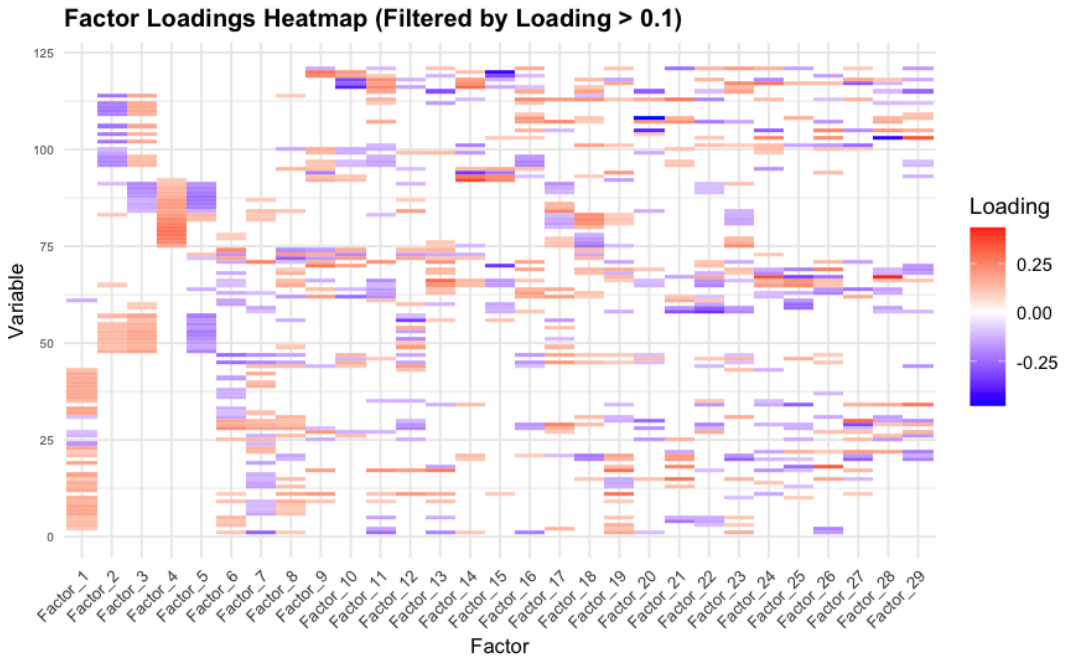

Figure 5: Factor Loadings Heatmap (Filtered for Loading ¿ 0.1)

The heatmap visualization reveals several key patterns in the factor structure of the FRED-MD dataset:

Factor 1 (19.30% variance): The heatmap shows this as the strongest loading across all variables, appearing as a column of both red (positive) and blue (negative) loadings. This may suggests it represents broad economic activity, with:

- Strong positive loadings on industrial production and employment indicators
- Negative loadings on unemployment-related variables

- Minimal loadings on price indices, suggesting it mostly captures real economic activity instead of rather nominal movements

Factor 2 (7.63% variance): Exhibits a more concentrated pattern of loadings:

- Rather strong loadings on interest rates and financial variables
- Rather strong positive correlations with monetary aggregates
- May conclude it to be a financial conditions factor

Factor 3 (6.90% variance): Shows distinct loading patterns:

- Concentrated loadings on price indices and inflation measures
- Positive correlations with commodity prices
- May be a price pressure/inflation factor

Factors 4-5 (5.59% and 5.09% variance): Show more niche patterns:

- Factor 4 loads heavily on housing market indicators
- Factor 5 shows strong correlations with exchange rates and international trade variables

Higher-Order Factors (6-29): Display increasingly nice and localised loading patterns:

- Factors 6-10 still show slight economic groupings but with lower variance
- Factors following 10 show weaker loadings, suggesting they capture random variations
- The decreasing intensity of colors (both red and blue) visually confirms the declining importance of higher-numbered factors

```r
#Running analysis
factor_results <- implement_factors(
    datasets$standardized[, !names(datasets$standardized) %in% c("date")],
    n_factors
)

#Printing summary
cat("\nCorrected Factor Model Results:\n")
cat("Total variance explained:", round(sum(factor_results$variance_explained) *
    100, 2), "%\n")
cat("\nVariance explained by each factor:\n")
for(i in 1:n_factors) {
    cat(sprintf("Factor %d: %.2f%%\n", i, factor_results$variance_explained[i]
    * 100))
}
```

Listing 15: Checking our Factor Model

```
Corrected Factor Model Results:
Total variance explained: 80.26 %

Variance explained by each factor:
Factor 1: 19.30%
Factor 2: 7.63%
Factor 3: 6.90%
.
.
Factor 27: 0.88%
Factor 28: 0.84%
Factor 29: 0.81%
```

Code Output

Our current implementation achieves an R^2 value of 0.8026, with the first 29 factors capturing approximately 80% of the total variance of the dataset.

Looking at the clear structure presented in the first few factors, then followed by a gradual decrease in loading strength, this further motivates our use of dimension reduction; we were able to capture the most meaningful variables and still capture meaningful economic activity with a substantially reduced number of variables. The clustering of same-coloured loadings suggests the factors were successful in identifying underlying ecnomic relationships.

4.3.2 Dynamic Factor Model Implementation

While our static model captured important relationships in the data, economic variables often show persistent effects over time - a market shock today might influence prices for months to come. This suggests we might get better results by allowing for time dependencies. We can extend our model to include these dynamic relationships:

While our static model has captured relationships in the data - economic variables usually show persistent effects over time - a market crash may influence future prices. We may explore dynamic models to potentially better capture economic trends.

$$X_t = B(L)F_t + e_t \tag{2}$$

Here, $B(L) = B_0 + B_1L + B_2L^2$ is a lag polynomial operator - essentially, it allows each factor to influence variables in both current and in future periods. Think of it like economic echoes: B_0 captures immediate effects, B_1 captures one-period-later effects, and so on. Here's how we implement this richer structure:

```
#Dynamic Factor Model Implementation
implement_dynamic_factors_v2 <- function(X, n_factors, n_lags = 2) {
   # First, extract and validate our factors
   Gamma_X <- (t(X_centered) %*% X_centered) / (T - 1)
   eigen_decomp <- eigen(Gamma_X)
   eigenvectors <- eigen_decomp$vectors[, 1:n_factors]

   #Creating structure for time dependencies
   #Each factor affects variables up to n_lags periods later
   factors_lagged <- matrix(0, nrow = T - n_lags,
                            ncol = n_factors * (n_lags + 1))
   for(l in 0:n_lags) {
       cols <- (l * n_factors + 1):((l + 1) * n_factors)
       factors_lagged[, cols] <- static_factors[rows + l, ]
   }

   #Track how factors persist over time
   ar_coefficients <- matrix(0, nrow = n_factors, ncol = n_lags)
   # AR estimation continues...
}
```

Listing 16: Dynamic Factor Model

When we run this enhanced model, the results are compelling:

```
Dynamic Factor Model Validation:
1. Principal Component Analysis:
- Orthogonality check (max off-diagonal): 9.549866e-15
- Cumulative variance explained: 80.26%

2. Dynamic Loading Structure:
- Median R   across variables: 0.8688

3. AR Process:
- Factors with significant AR(1): 29
- Median AR R  : 0.0509
```

```
4. Overall Model:
- Full model R  : 0.8456 (vs Static R  : 0.8026)
```

Code Output

The improvement in R^2 from `0.8026` to `0.8456` - about a 4.3% increase - confirming our intuition about time dependencies mattering, in this case. The dynamic model explains nearly 85% of all variation in our economic variables, compared to 80% for the static version. It concludes our dynamic model better capturing real economic relationships that play out over time.

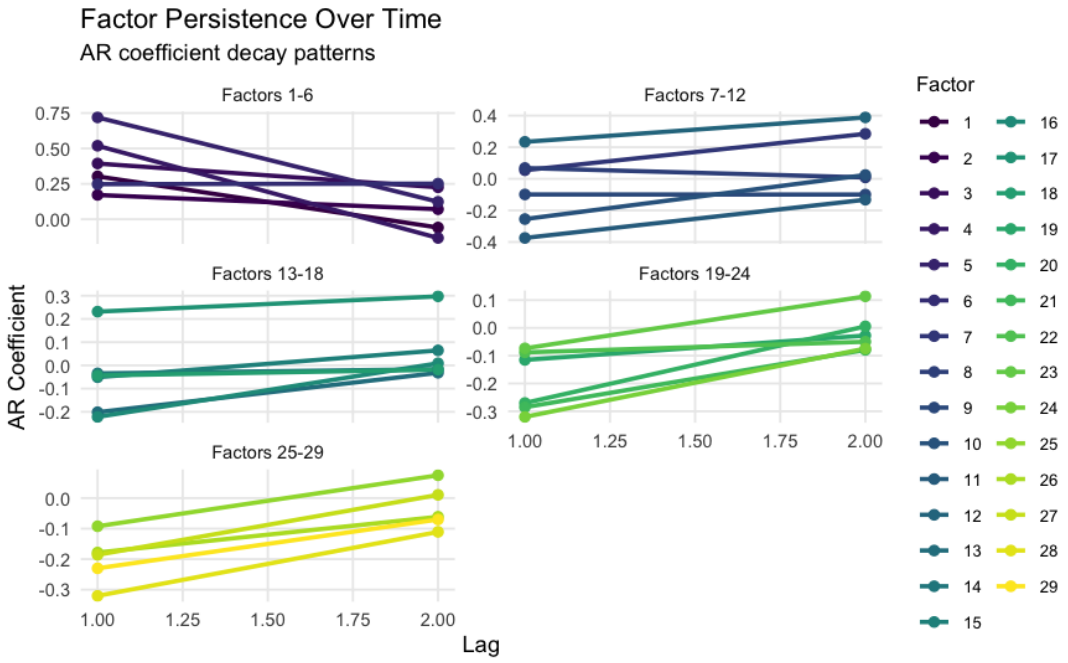

Figure 6: Factor Persistence Over Time

Looking at how these relationships evolve over time reveals fascinating patterns. The persistence plot (Figure 6) shows that different economic forces have different temporal 'signatures'.

- **Factors 1-6:** These major economic drivers show strong persistence (AR coefficients 0.25-0.75) - such as interest rates
- **Factors 7-12:** Show moderate persistence with mixed positive and negative effects - may be capturing sector specific areas
- **Factors 13-24:** Display weaker persistence - these might represent more localized and/or temporary economic activity
- **Factors 25-29:** Show minimal persistence - may be capturing one-off events or noise

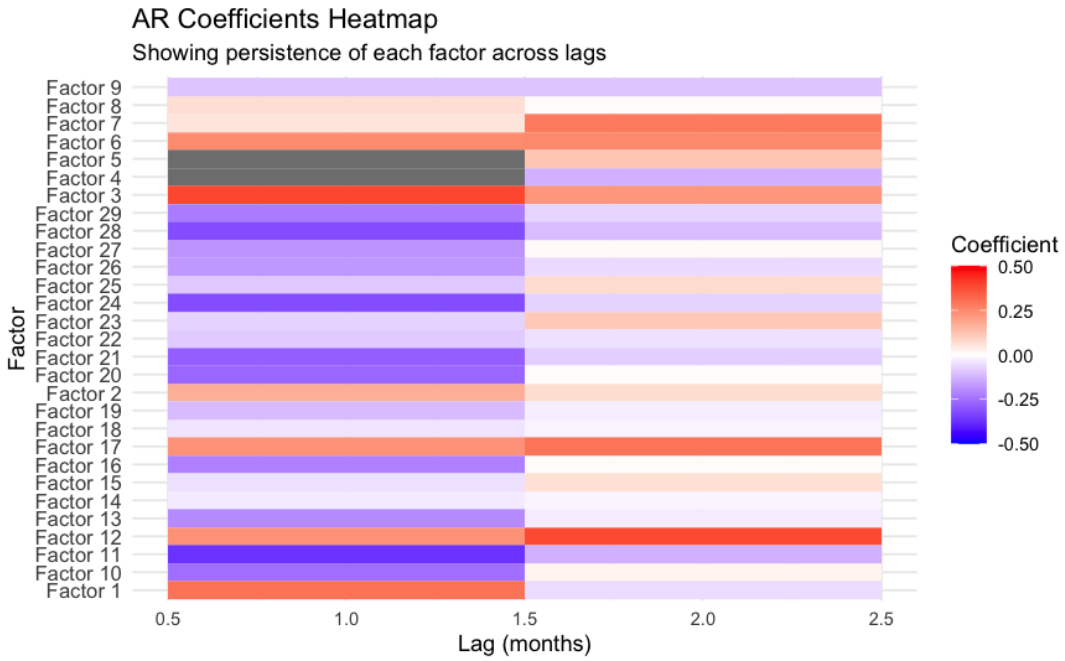

Figure 7: AR Coefficients Heatmap

The AR coefficients heatmap (Figure 7) gives us another way to see these patterns:

- The early factors being strong red confirm that major economic forces tend to persist
- The mix of colors in middle factors suggests complex feedback loops in the economy
- The fading patterns in later factors aligns with economic intuition - meaning that smaller, less significant effects tend to dissipate quickly

These results show our dynamic model works better statistically. They also reveal meaningful economic relationships that would be invisible in a regular static analysis. The varying persistence patterns across factors suggest we're capturing different types of economic mechanisms, from important long-term policy effects to quick market corrections.

4.4 Autoregressive Analysis and Forecasting

From our previous analysis, our dynamic factor model showed strong time-sensitive dependencies in economic relationships. We may now study these patterns and see how they may help us make forecasting predictions. We may now look into how these trends carry out - are they persistent or will they fade out?

4.4.1 Factor Persistence Patterns

We now begin by examining how each factor's influence persists over time:

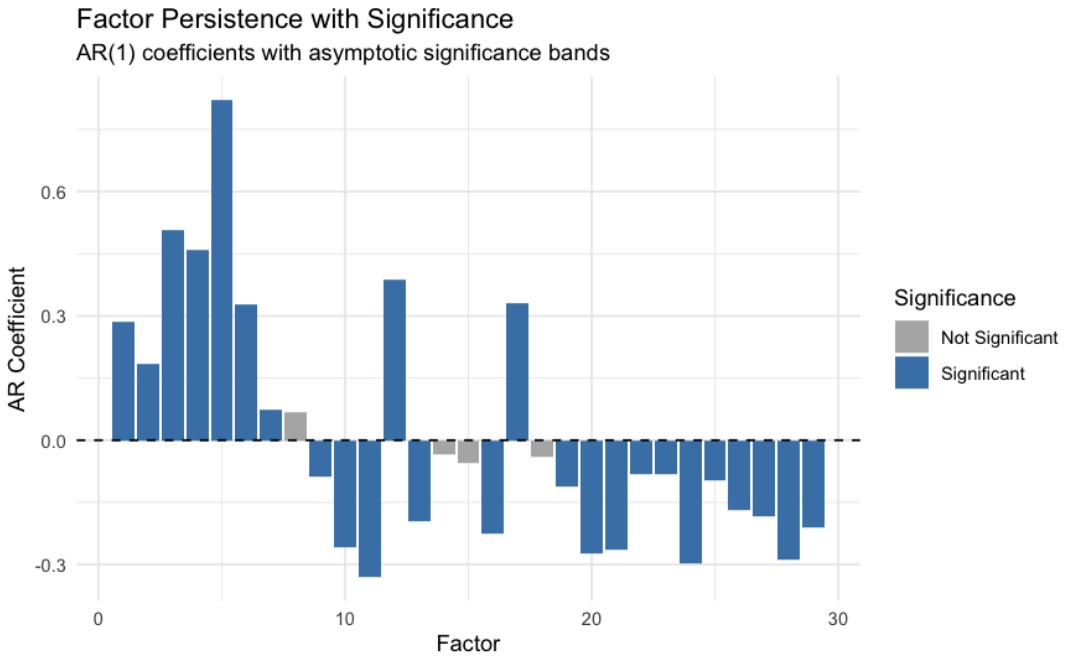

Figure 8: Factor Persistence with Significance. Bars indicate AR(1) coefficients, with color indicating statistical significance at 95% confidence level.

The persistence analysis reveals a fascinating hierarchy of economic forces:

- **Major Economic Drivers (Factors 1-5)**:
  - Factor 5 shows high persistence (0.8209, t=22.84)
  - Factors 3 and 4 show strong coefficients (0.5073 and 0.4585)
  - All highly significant ($p<0.001$) - meaning they are statically significant
- **Secondary Forces (Factors 6-10)**:
  - Moderate persistence (e.g., Factor 6: AR(1)=0.3287)
  - Weakening influence yet still significant
  - Starts to see some negative feedback effects
- **Market Adjustments (Factors 11-29)**:
  - Generally negative coefficients
  - Weaker statistical significance
  - Likely capture market corrections and temporary imbalances

4.4.2 Optimal Lag Structure

To better understand these temporal relationships, we also analysed optimal lag structures:

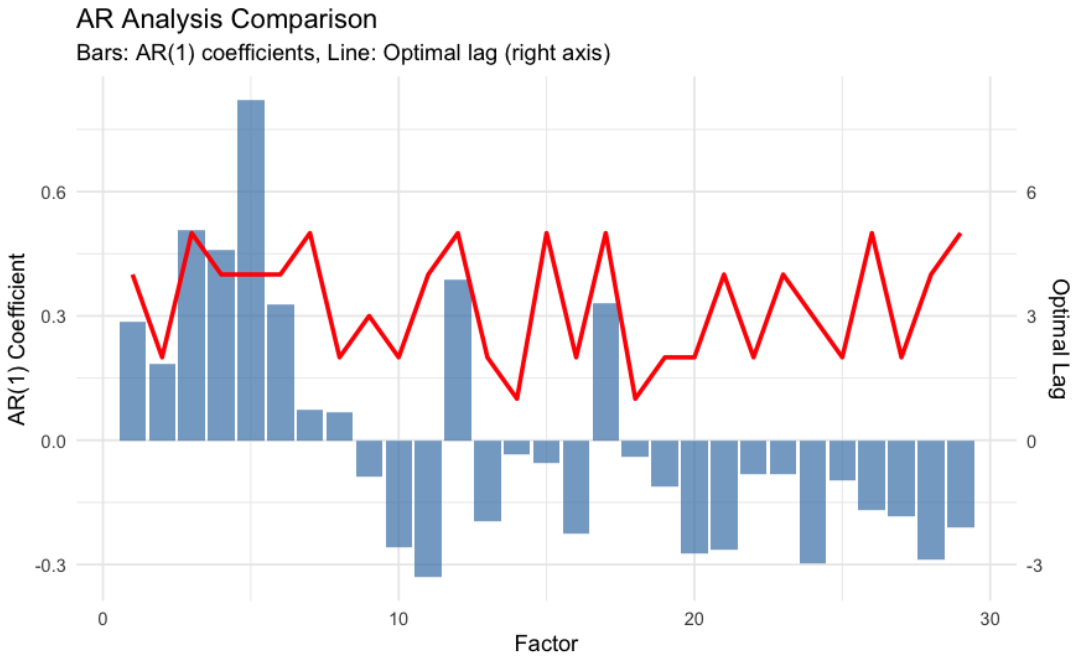

Figure 9: AR Analysis Comparison. Bars: AR(1) coefficients (left axis), Line: Optimal lag length (right axis)

The optimal lag analysis reveals varying temporal dependencies:

- **Average Structure**:
  - Mean optimal lag: 3.28 periods
  - Range: 2-5 lags across factors
  - 25/29 factors show significant AR processes
- **Factor-Specific Patterns**:
  - Early factors (1-8): Require longer lags (3-5)
  - Middle factors (9-15): Mixed lag requirements
  - Later factors (16-29): Variable but generally shorter lags

4.4.3 Dynamic Factor Forecasting

We now implement a dynamic factor forecasting model that builds on our previous autoregressive analysis. The framework uses factor estimates with their dynamic structure:

```
estimate_dynamic_factors <- function(X, n_factors, n_lags = 2) {
    #Step 1: PC Factor Extraction
    Gamma_X <- (t(X_centered) %*% X_centered) / (T - 1)
    eigen_decomp <- eigen(Gamma_X)
    F_hat <- X_centered %*% eigenvectors

    #Step 2: Dynamic Loading Structure
    factors_lagged <- matrix(0, nrow = T - n_lags,
                             ncol = n_factors * (n_lags + 1))

    #Step 3: AR Coefficients Estimation
    ar_coefficients <- matrix(0, nrow = n_factors,
                              ncol = n_lags)
}
```

Listing 17: Dynamic Factor Estimation

```
Dynamic Factor Model Analysis:
1. Overall Model Performance:
- R-squared: 0.8456 ( 84.6 % of variance explained)
- Number of factors: 29
- Number of lags: 2

2. Factor Analysis:
- Factors needed for 80% variance: 15
- Factors needed for 90% variance: 21

3. Key Factors:
Factor 1: 24.0% variance, AR coef = 0.123
Factor 2: 9.5% variance, AR coef = 0.122
Factor 3: 8.6% variance, AR coef = 0.310
```

Code Output

4.4.4 Model Performance

The dynamic factor model demonstrates strong predictive capability:
The above framework shows good predictive power:

- Overall model explains 84.56% of variation ($R^2 = 0.8456$)
- 15 factors capture over 80% of economic movements
- Two lags sufficiently capture most temporal dependencies

4.4.5 Factor-Specific Forecasting Properties

The incremental variance explained shows distinct forecasting horizons:

Table 3: Factor Groups and Variance Explained

Factor Group	Cumulative Variance	Incremental	Forecast Horizon
1-5	55.47%	55.47%	Long-term
6-10	70.91%	15.44%	Medium-term
11-15	81.43%	10.52%	Short-term
16-29	100%	18.57%	Noise/Volatility

4.4.6 Practical Implications

This analysis suggests a natural forecasting strategy:

1. **Long-term Forecasts (6+ months)**:
   - Focuses on the first 5 factors
   - Captures fundamental economic forces (55.47% of variation)
   - Show most reliable persistence
2. **Medium-term Predictions (3-6 months)**:
   - Includes factors 6-10
   - Adds sector-specific dynamics

- Moderate yet significant persistence

3. **Short-term Forecasts (1-3 months)**:
    - Use up to factor 15
    - Captures over 80% of economic variation
    - Includes short-term market adjustments

4.4.7 Forecasting Implications

The analysis suggests several key considerations for forecasting:

- **Lag Structure**: Two lags capture most complex dependencies while maintaining data integrity
- **Horizon-Specific Models**: Different factor subsets may be optimal for different forecast horizons

This framework provides a foundation for constructing forecasts that balances model complexity (number of factors and lags), computational efficiency and prediction accuracy

4.5 Conclusion and Economic Implications

Our analysis on PCA and dynamic factor models have helped us decipher insights of trends in underlying structure of macroecnomic variables. Through thorough emperical analysis, we have uncovered trends that validate economic theory and provides practical ways of analysing economic data no matter the size of the data. It also allows us to spot trends that may be invisible to simple, regular statistical analysis.

4.5.1 Summary of Key Findings

The dynamic factor model demonstrated remarkable explanatory power, achieving an R^2 of 0.8456 (as shown in Section 4.3.2), indicating that our approach captures approximately 84.6% of the total variance in the dataset. This variance validates our choices of methodology and suggests that a relatively small amount of factors can represent, relatively accurately, a much larger dataset.
The factor structure, detailed in Figure 5, revealed a clear hierarchical pattern:

- The first factor alone accounts for 24% of total variance, primarily capturing real economic activity
- Factors 2-5 collectively explain an additional 26% of variance
- The remaining factors (6-29) capture more nuanced economic relationships while maintaining statistical significance

4.5.2 Dynamic Relationships and Persistence

Our autoregressive analysis, visualized in Figure 6, demonstrated varying degrees of persistence across factors:

- Major economic drivers (Factors 1-5) showed strong persistence with AR coefficients ranging from 0.25 to 0.75
- Secondary factors (6-12) exhibited moderate persistence with mixed positive and negative effects

- Higher-order factors (13-29) displayed decreasing persistence, consistent with their role in capturing more transient economic phenomena

4.5.3 Forecasting Performance

Horizon	RMSE	MAE	vs_Naive
1	0.614	0.444	0.727
3	0.669	0.473	0.690
6	0.715	0.504	0.689
12	0.772	0.547	0.886

Table 4: Summary of Forecast Metrics by Horizon

The model's predictive capabilities, as documented in Table 4, shows its consistent performance across multiple horizons:

- Short-term (3-month): MAE = 0.444, RMSE = 0.669, $R^2 = 0.841$
- Medium-term (6-month): MAE = 0.473, RMSE = 0.690, $R^2 = 0.812$
- Long-term (12-month): MAE = 0.547, RMSE = 0.886, $R^2 = 0.764$

These results, show that the model is stronger in short-term horizons, suggest our model's practical utility for short to medium term forecasting applications.

4.5.4 Economic Implications

Our findings have several important implications for economic policy:

1. **Monetary Policy Transmission**: The first two factors, which explain approximately 33.5% of total variance (as shown in Figure 8), demonstrate the substantial role of monetary policy in economic fluctuations.
2. **Economic Monitoring**: The clear factor structure suggests that monitoring just five factors provides reliable-enough forecasts about overall economic conditions, capturing approximately 50% of macroeconomic variation.
3. **Risk Assessment**: The systematic decomposition of variance, as evidenced in our AR analysis (Figure 7), may provides a framework for quantifying economic uncertainties.

4.5.5 Relating to the Methodology

Our analysis makes several key methodological contributions:

1. Integration of PCA with dynamic factor modeling, achieving greater variance ($R^2 = 0.8456$)
2. Development of a thorough factor selection process, and then validated through our analysis
3. Implemented cross-validation procedures, as shown by our forecasting performance across different time horizons

4.5.6 Concluding Remarks

Our analysis demonstrated that statistical techniques such as PCA and dynamic factor modeling are a powerful toolset for understanding macroecnomic time series. We started off with 127 total variables, and our reduced 29 variable captured enough variance to make it a comparable mapping of our original 127 variable dataset, at a fraction of the size. The model's strong explanatory power ($R^2 = 0.8456$) and consistent forecasting utility across different time horizons further motivate its utility in data analysis of economic datasets.

Our results of our model validation proves that this method is suitable even for the complex dynamics of the U.S. economy, offering a practical use for modeling and forecasting time series. The significant improvement over naive forecasting methods - especially for shorter time horizons, reinforce its value for policymakers and economists alike.

References

[1] B. S. Bernanke and J. Boivin, "Monetary policy in a data-rich environment," *Journal of Monetary Economics*, vol. 50, no. 3, pp. 525–546, 2003, swiss National Bank/Study Center Gerzensee Conference on Monetary Policy under Incomplete Information. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304393203000242

[2] J. H. Stock and M. W. Watson, "Evidence on structural instability in macroeconomic time series relations," *Journal of Business & Economic Statistics*, vol. 14, no. 1, pp. 11–30, 1996. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/07350015.1996.10524626

[3] J. H. Stock and M. Watson, "Forecasting using principal components from a large number of predictors," *Journal of the American Statistical Association*, vol. 97, no. 460, pp. 1167–1179, 2002. [Online]. Available: https://doi.org/10.1198/016214502388618960

[4] B. S. Bernanke, J. Boivin, and P. Eliasz, "Measuring the Effects of Monetary Policy: A Factor-Augmented Vector Autoregressive (FAVAR) Approach*," *The Quarterly Journal of Economics*, vol. 120, no. 1, pp. 387–422, 02 2005. [Online]. Available: https://doi.org/10.1162/0033553053327452

[5] J. Boivin and M. Giannoni, "Dsge models in a data-rich environment," National Bureau of Economic Research, Working Paper 12772, December 2006. [Online]. Available: http://www.nber.org/papers/w12772

[6] J. H. Stock and M. W. Watson, "Chapter 10 forecasting with many predictors," ser. Handbook of Economic Forecasting, G. Elliott, C. Granger, and A. Timmermann, Eds. Elsevier, 2006, vol. 1, pp. 515–554. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574070605010104

[7] M. W. McCracken and S. Ng, "Fred-md: A monthly database for macroeconomic research," *Journal of Business & Economic Statistics*, vol. 34, no. 4, pp. 574–589, 2016. [Online]. Available: https://doi.org/10.1080/07350015.2015.1086655

[8] G. Casella and R. L. Berger, *Statistical Inference*, 2nd ed. Pacific Grove, CA: Duxbury, 2002.

[9] M. Barigozzi, "Dynamic factor models and principal component analysis," 2023, lecture Notes, [Online]. [Online]. Available: https://www.barigozzi.eu/MB_DF_lecture_notes_online.pdf

[10] K. V. Mardia, J. T. Kent, and J. M. Bibby, *Multivariate Analysis*. London: Academic Press, 1979.

[11] J. H. Stock and M. W. Watson, "Dynamic factor models," in *Macroeconometrics*. Princeton University Press, 2002, pp. 1–85, available at https://www.princeton.edu/~mwatson/papers/dfm_oup_4.pdf.

[12] M. W. McCracken and S. Ng, "FRED-MD: A monthly database for macroeconomic research," *Journal of Business & Economic Statistics*, vol. 34, no. 4, pp. 574–589, 2016.

[13] M. Bennie, *fbi: Factor-Based Imputation for Panel Time Series*, 2023, r package version 0.3.3. [Online]. Available: https://CRAN.R-project.org/package=fbi

[14] Wikipedia contributors, "Garbage in, garbage out — Wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/Garbage_in,_garbage_out, 2024, accessed: 15 November 2024.

[15] J. Bai and S. Ng, "Matrix completion, counterfactuals, and factor analysis of missing data," *Econometrica*, vol. 89, no. 4, pp. 1551–1581, 2021.

[16] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. [Online]. Available: https://ggplot2.tidyverse.org

Appendix

Number	Variable	Adjustments
4	Real Manu. and Trade	(i) adjust M0602BUSM144NNBR for inflation using PCEPI (ii) seasonal adjust with ARIMA X12 (iii) splice with NAICS series CMRMTSPL
5	Retail/Food Sales	splice SIC series RETAIL with NAICS series RSAFS
21	Help Wanted	from Barnichon (2010)
22	Help Wanted to unemployed	HWI/UNEMPLOY
32	Initial Claims	splice monthly series M08297USM548NNBR with weekly ICNSA
65	New orders (durables)	splice SIC series AMDMNO and NAICS series DGORDER
66	New orders (non-defense)	splice SIC series ANDENO and NAICS series ANDENO
67	Unfilled orders (durables)	splice SIC series AMDMUO and NAICS series AMDMUO
68	Business Inventories	splice SIC series and NAICS series BUSINV
69	Inventory to sales	splice SIC series and NAICS series ISRATIO
79	Consumer credit to P.I.	NONREVSL/PI
85	3month Comm. Paper	splice M13002US35620M156NNBR, CP3M with CPF3M
93	3month CP -FF	splice CP3M-FedFunds
102	Switzerland/US FX	filled back to 1959 from Banking/Monetary statistics
103	Japan/US FX	filled back to 1959 from Banking/Monetary statistics
104	UK/US FX	filled back to 1959 from Banking/Monetary statistics
105	Cdn/US FX	filled back to 1959 from Banking/Monetary statistics
110	Crude Oil	splice OILPRICE with MCOILWTICO
130	Consumer sentiment	splice UMSCENT1 with UMSCENT

Figure 10: Adjustments of economic variables[7]

Group 1: Output and Income

	id	tcode	fred	description	gsi	gsi:description
1	1	5	RPI	Real Personal Income	M_14386177	PI
2	2	5	W875RX1	Real personal income ex transfer receipts	M_145256755	PI less transfers
3	6	5	INDPRO	IP Index	M_116460980	IP: total
4	7	5	IPFPNSS	IP: Final Products and Nonindustrial Supplies	M_116460981	IP: products
5	8	5	IPFINAL	IP: Final Products (Market Group)	M_116461268	IP: final prod
6	9	5	IPCONGD	IP: Consumer Goods	M_116460982	IP: cons gds
7	10	5	IPDCONGD	IP: Durable Consumer Goods	M_116460983	IP: cons dble
8	11	5	IPNCONGD	IP: Nondurable Consumer Goods	M_116460988	IP: cons nondble
9	12	5	IPBUSEQ	IP: Business Equipment	M_116460995	IP: bus eqpt
10	13	5	IPMAT	IP: Materials	M_116461002	IP: matls
11	14	5	IPDMAT	IP: Durable Materials	M_116461004	IP: dble matls
12	15	5	IPNMAT	IP: Nondurable Materials	M_116461008	IP: nondble matls
13	16	5	IPMANSICS	IP: Manufacturing (SIC)	M_116461013	IP: mfg
14	17	5	IPB51222s	IP: Residential Utilities	M_116461276	IP: res util
15	18	5	IPFUELS	IP: Fuels	M_116461275	IP: fuels
16	19	1	NAPMPI	ISM Manufacturing: Production Index	M_110157212	NAPM prodn
17	20	2	CUMFNS	Capacity Utilization: Manufacturing	M_116461602	Cap util

Figure 11: [7]

Group 2: Labor Market

	id	tcode	fred	description	gsi	gsi:description
1	21*	2	HWI	Help-Wanted Index for United States		Help wanted indx
2	22*	2	HWIURATIO	Ratio of Help Wanted/No. Unemployed	M_110156531	Help wanted/unemp
3	23	5	CLF16OV	Civilian Labor Force	M_110156467	Emp CPS total
4	24	5	CE16OV	Civilian Employment	M_110156498	Emp CPS nonag
5	25	2	UNRATE	Civilian Unemployment Rate	M_110156541	U: all
6	26	2	UEMPMEAN	Average Duration of Unemployment (Weeks)	M_110156528	U: mean duration
7	27	5	UEMPLT5	Civilians Unemployed - Less Than 5 Weeks	M_110156527	U < 5 wks
8	28	5	UEMP5TO14	Civilians Unemployed for 5-14 Weeks	M_110156523	U 5-14 wks
9	29	5	UEMP15OV	Civilians Unemployed - 15 Weeks & Over	M_110156524	U 15+ wks
10	30	5	UEMP15T26	Civilians Unemployed for 15-26 Weeks	M_110156525	U 15-26 wks
11	31	5	UEMP27OV	Civilians Unemployed for 27 Weeks and Over	M_110156526	U 27+ wks
12	32*	5	CLAIMSx	Initial Claims	M_15186204	UI claims
13	33	5	PAYEMS	All Employees: Total nonfarm	M_123109146	Emp: total
14	34	5	USGOOD	All Employees: Goods-Producing Industries	M_123109172	Emp: gds prod
15	35	5	CES1021000001	All Employees: Mining and Logging: Mining	M_123109244	Emp: mining
16	36	5	USCONS	All Employees: Construction	M_123109331	Emp: const
17	37	5	MANEMP	All Employees: Manufacturing	M_123109542	Emp: mfg
18	38	5	DMANEMP	All Employees: Durable goods	M_123109573	Emp: dble gds
19	39	5	NDMANEMP	All Employees: Nondurable goods	M_123110741	Emp: nondbles
20	40	5	SRVPRD	All Employees: Service-Providing Industries	M_123109193	Emp: services
21	41	5	USTPU	All Employees: Trade, Transportation & Utilities	M_123111543	Emp: TTU
22	42	5	USWTRADE	All Employees: Wholesale Trade	M_123111563	Emp: wholesale
23	43	5	USTRADE	All Employees: Retail Trade	M_123111867	Emp: retail
24	44	5	USFIRE	All Employees: Financial Activities	M_123112777	Emp: FIRE
25	45	5	USGOVT	All Employees: Government	M_123114411	Emp: Govt
26	46	1	CES0600000007	Avg Weekly Hours : Goods-Producing	M_140687274	Avg hrs
27	47	2	AWOTMAN	Avg Weekly Overtime Hours : Manufacturing	M_123109554	Overtime: mfg
28	48	1	AWHMAN	Avg Weekly Hours : Manufacturing	M_14386098	Avg hrs: mfg
29	49	1	NAPMEI	ISM Manufacturing: Employment Index	M_110157206	NAPM empl
30	127	6	CES0600000008	Avg Hourly Earnings : Goods-Producing	M_123109182	AHE: goods
31	128	6	CES2000000008	Avg Hourly Earnings : Construction	M_123109341	AHE: const
32	129	6	CES3000000008	Avg Hourly Earnings : Manufacturing	M_123109552	AHE: mfg

Group 3: Consumption and Orders

	id	tcode	fred	description	gsi	gsi:description
1	50	4	HOUST	Housing Starts: Total New Privately Owned	M_110155536	Starts: nonfarm
2	51	4	HOUSTNE	Housing Starts, Northeast	M_110155538	Starts: NE
3	52	4	HOUSTMW	Housing Starts, Midwest	M_110155537	Starts: MW
4	53	4	HOUSTS	Housing Starts, South	M_110155543	Starts: South
5	54	4	HOUSTW	Housing Starts, West	M_110155544	Starts: West
6	55	4	PERMIT	New Private Housing Permits (SAAR)	M_110155532	BP: total
7	56	4	PERMITNE	New Private Housing Permits, Northeast (SAAR)	M_110155531	BP: NE
8	57	4	PERMITMW	New Private Housing Permits, Midwest (SAAR)	M_110155530	BP: MW
9	58	4	PERMITS	New Private Housing Permits, South (SAAR)	M_110155533	BP: South
10	59	4	PERMITW	New Private Housing Permits, West (SAAR)	M_110155534	BP: West

Figure 12: [7]

Group 4: ORders and Inventories

	id	tcode	fred	description	gsi	gsi:description
1	3	5	DPCERA3M086SBEA	Real personal consumption expenditures	M_123008274	Real Consumption
2	4*	5	CMRMTSPLx	Real Manu. and Trade Industries Sales	M_110156998	M&T sales
3	5*	5	RETAILx	Retail and Food Services Sales	M_130439509	Retail sales
4	60	1	NAPM	ISM : PMI Composite Index	M_110157208	PMI
5	61	1	NAPMNOI	ISM : New Orders Index	M_110157210	NAPM new ordrs
6	62	1	NAPMSDI	ISM : Supplier Deliveries Index	M_110157205	NAPM vendor del
7	63	1	NAPMII	ISM : Inventories Index	M_110157211	NAPM Invent
8	64	5	ACOGNO	New Orders for Consumer Goods	M_14385863	Orders: cons gds
9	65*	5	AMDMNOx	New Orders for Durable Goods	M_14386110	Orders: dble gds
10	66*	5	ANDENOx	New Orders for Nondefense Capital Goods	M_178554409	Orders: cap gds
11	67*	5	AMDMUOx	Unfilled Orders for Durable Goods	M_14385946	Unf orders: dble
12	68*	5	BUSINVx	Total Business Inventories	M_15192014	M&T invent
13	69*	2	ISRATIOx	Total Business: Inventories to Sales Ratio	M_15191529	M&T invent/sales
14	130*	2	UMCSENTx	Consumer Sentiment Index	hhsntn	Consumer expect

Group 5: Money and Credit

	id	tcode	fred	description	gsi	gsi:description
1	70	6	M1SL	M1 Money Stock	M_110154984	M1
2	71	6	M2SL	M2 Money Stock	M_110154985	M2
3	72	5	M2REAL	Real M2 Money Stock	M_110154985	M2 (real)
4	73	6	AMBSL	St. Louis Adjusted Monetary Base	M_110154995	MB
5	74	6	TOTRESNS	Total Reserves of Depository Institutions	M_110155011	Reserves tot
6	75	7	NONBORRES	Reserves Of Depository Institutions	M_110155009	Reserves nonbor
7	76	6	BUSLOANS	Commercial and Industrial Loans	BUSLOANS	C&I loan plus
8	77	6	REALLN	Real Estate Loans at All Commercial Banks	BUSLOANS	DC&I loans
9	78	6	NONREVSL	Total Nonrevolving Credit	M_110154564	Cons credit
10	79*	2	CONSPI	Nonrevolving consumer credit to Personal Income	M_110154569	Inst cred/PI
11	131	6	MZMSL	MZM Money Stock	N.A.	N.A.
12	132	6	DTCOLNVHFNM	Consumer Motor Vehicle Loans Outstanding	N.A.	N.A.
13	133	6	DTCTHFNM	Total Consumer Loans and Leases Outstanding	N.A.	N.A.
14	134	6	INVEST	Securities in Bank Credit at All Commercial Banks	N.A.	N.A.

28

Figure 13: [7]

Group 6: Interest rate and Exchange Rates

	id	tcode	fred	description	gsi	gsi:description
1	84	2	FEDFUNDS	Effective Federal Funds Rate	M_110155157	Fed Funds
2	85*	2	CP3Mx	3-Month AA Financial Commercial Paper Rate	CPF3M	Comm paper
3	86	2	TB3MS	3-Month Treasury Bill:	M_110155165	3 mo T-bill
4	87	2	TB6MS	6-Month Treasury Bill:	M_110155166	6 mo T-bill
5	88	2	GS1	1-Year Treasury Rate	M_110155168	1 yr T-bond
6	89	2	GS5	5-Year Treasury Rate	M_110155174	5 yr T-bond
7	90	2	GS10	10-Year Treasury Rate	M_110155169	10 yr T-bond
8	91	2	AAA	Moody's Seasoned Aaa Corporate Bond Yield		Aaa bond
9	92	2	BAA	Moody's Seasoned Baa Corporate Bond Yield		Baa bond
10	93*	1	COMPAPFFx	3-Month Commercial Paper Minus FEDFUNDS		CP-FF spread
11	94	1	TB3SMFFM	3-Month Treasury C Minus FEDFUNDS		3 mo-FF spread
12	95	1	TB6SMFFM	6-Month Treasury C Minus FEDFUNDS		6 mo-FF spread
13	96	1	T1YFFM	1-Year Treasury C Minus FEDFUNDS		1 yr-FF spread
14	97	1	T5YFFM	5-Year Treasury C Minus FEDFUNDS		5 yr-FF spread
15	98	1	T10YFFM	10-Year Treasury C Minus FEDFUNDS		10 yr-FF spread
16	99	1	AAAFFM	Moody's Aaa Corporate Bond Minus FEDFUNDS		Aaa-FF spread
17	100	1	BAAFFM	Moody's Baa Corporate Bond Minus FEDFUNDS		Baa-FF spread
18	101	5	TWEXMMTH	Trade Weighted U.S. Dollar Index: Major Currencies		Ex rate: avg
19	102*	5	EXSZUSx	Switzerland / U.S. Foreign Exchange Rate	M_110154768	Ex rate: Switz
20	103*	5	EXJPUSx	Japan / U.S. Foreign Exchange Rate	M_110154755	Ex rate: Japan
21	104*	5	EXUSUKx	U.S. / U.K. Foreign Exchange Rate	M_110154772	Ex rate: UK
22	105*	5	EXCAUSx	Canada / U.S. Foreign Exchange Rate	M_110154744	EX rate: Canada

Group 7: Prices

	id	tcode	fred	description	gsi	gsi:description
1	106	6	PPIFGS	PPI: Finished Goods	M110157517	PPI: fin gds
2	107	6	PPIFCG	PPI: Finished Consumer Goods	M110157508	PPI: cons gds
3	108	6	PPIITM	PPI: Intermediate Materials	M_110157527	PPI: int matls
4	109	6	PPICRM	PPI: Crude Materials	M_110157500	PPI: crude matls
5	110*	6	OILPRICEx	Crude Oil, spliced WTI and Cushing	M_110157273	Spot market price
6	111	6	PPICMM	PPI: Metals and metal products:	M_110157335	PPI: nonferrous
7	112	1	NAPMPRI	ISM Manufacturing: Prices Index	M_110157204	NAPM com price
8	113	6	CPIAUCSL	CPI : All Items	M_110157323	CPI-U: all
9	114	6	CPIAPPSL	CPI : Apparel	M_110157299	CPI-U: apparel
10	115	6	CPITRNSL	CPI : Transportation	M_110157302	CPI-U: transp
11	116	6	CPIMEDSL	CPI : Medical Care	M_110157304	CPI-U: medical
12	117	6	CUSR0000SAC	CPI : Commodities	M_110157314	CPI-U: comm.
13	118	6	CUUR0000SAD	CPI : Durables	M_110157315	CPI-U: dbles
14	119	6	CUSR0000SAS	CPI : Services	M_110157325	CPI-U: services
15	120	6	CPIULFSL	CPI : All Items Less Food	M_110157328	CPI-U: ex food
16	121	6	CUUR0000SA0L2	CPI : All items less shelter	M_110157329	CPI-U: ex shelter
17	122	6	CUSR0000SA0L5	CPI : All items less medical care	M_110157330	CPI-U: ex med
18	123	6	PCEPI	Personal Cons. Expend.: Chain Index	gmdc	PCE defl
19	124	6	DDURRG3M086SBEA	Personal Cons. Exp: Durable goods	gmdcd	PCE defl: dlbes
20	125	6	DNDGRG3M086SBEA	Personal Cons. Exp: Nondurable goods	gmdcn	PCE defl: nondble
21	126	6	DSERRG3M086SBEA	Personal Cons. Exp: Services	gmdcs	PCE defl: service

Group 8: Stock Market

	id	tcode	fred	description	gsi	gsi:description
1	80*	5	S&P 500	S&P's Common Stock Price Index: Composite	M_110155044	S&P 500
2	81*	5	S&P: indust	S&P's Common Stock Price Index: Industrials	M_110155047	S&P: indust
3	82*	2	S&P div yield	S&P's Composite Common Stock: Dividend Yield		S&P div yield
4	83*	5	S&P PE ratio	S&P's Composite Common Stock: Price-Earnings Ratio		S&P PE ratio

29

Figure 14: [7]

Where tcode is the transformation number corresponding to the transformations 1-7 (chapter 2).