The Advantages of Vector Autoregression Over Independent Component Modelling (A Study on US Macroeconomic Data)

Matthew Seguin

Introduction

Economies are an essential part of any nation. They are what determine a nation's access to goods and services. Without a strong economy the population struggles. There has been much work done to measure the strength of an economy. One of the most popular ways to measure the strength of an economy is through the use of macroeconomic indicators. Some examples of macroeconomic indicators include GDP, unemployment rate, inflation rate, and interest rates. The indicators I will be using in this project are the following:

- Money Supply (M2)
- Consumer Price Index (CPI)
- Producer Price Index (PPI)

The money supply is the total amount of money in circulation or in existence in a country. The consumer price index is a measure of the average change over time in the prices paid by urban consumers for a market basket of consumer goods and services. The producer price index is a measure of the average change over time in the selling prices received by domestic producers for their output. These three indicators are important because they can help us understand how the economy is performing and how it may perform in the future. Additionally, they are all interrelated. For example, an increase in the money supply puts more money in the hands of the population while the demand for goods and services stays the same. This can lead to an increase in the consumer price index and subsequently increase in the producer price index. This interrelation is what makes vector autoregression (VAR) a useful tool for modeling these indicators. VAR is a statistical model similar to Autoregression that attempts to predict future points of a time series based on the past points, the key distinction here is that VAR uses vectors (with possible dimension greater than 1) as the observations instead of numbers. The question then arises: is VAR better than fitting a model for each component of a multivariate time series (i.e. independent component modeling)?

The goal of this project is to assess the performance of vector autoregressive models (VAR) in comparison to fitting independent ARMA models for each component of a multivariate time series. I will implement VAR from scratch and compare it with the predicitons from the independent modeling for each component time series. The data I will use in this project comes from three monthly datasets on FRED:

- Money Supply: https://fred.stlouisfed.org/series/M2REAL
- Consumer Price Index: https://fred.stlouisfed.org/series/CPIAUCSL
- Producer Price Index: https://fred.stlouisfed.org/series/PPIACO

In order to assess the performance of VAR in a usual setting I will disregard the data starting from 2020 which is when the COVID-19 pandemic started. The reason for this is because COVID caused a huge unexpected impact on the US economy so any predictions will be very far from the true results. So I will restrict the analysis in this project to working with data that is more predictable to compare VAR with independent component modeling. The data I will use is from 1959 to 2019 with a total of 732 observations. I will reserve a little over 15% of the data for testing (125 points) and a little under 85% for training (607 points).

Background

Stationarity

A time series y_t is called stationary if it is constant in mean $\mathbb{E}[y_t] = \mu$ for all t and $\text{Cov}(y_{t_1}, y_{t_2})$ only depends on $|t_1 - t_2|$. We then call $\gamma(h) = \text{Cov}(y_t, y_{t+h})$ the autocovariance function of the time series y_t. The autocorrelation function (ACF) of y_t is then $\rho(h) = \frac{\gamma(h)}{\gamma(0)}$ which is used to express how linearly related the time series is with its past values.

Autoregressive Models

An Autoregressive model is a statistical model used to predict future values of a time series based solely on the past values of that time series.

For a given set of starting points $y_1, \dots, y_p$ an AR(p) model assumes that each $\epsilon_t \overset{\text{iid}}{\sim} N(0, \sigma^2)$ and that for $t > p$ the future points follow the model

$$y_t = \phi_0 + \phi_1 y_{t-1} + \dots + \phi_p y_{t-p} + \epsilon_t$$

An important tool for choosing the order p of an Autoregressive model is the partial autocorrelation function denoted by $\text{PACF}(h) = \widehat{\phi}_h$ for a lag h is the value of the estimated coefficient $\widehat{\phi}_h$ on lag h in the model above when an AR(h) model is fit. For a true AR(p) model the PACF will drop to zero after lag p so by looking at the PACF we can tell what reasonable values of p are.

ARMA Models

Autoregressive Moving Average models are also statistical models used to predict future values of a time series based solely on the past values of that time series. An ARMA(p, q) model is represented by the following equation

$$(y_t - \mu) - \phi_1(y_{t-1} - \mu) - \dots - \phi_p(y_{t-p} - \mu) = \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q}$$

VAR Models

A Vector Autoregressive model is an abstraction of the univariate Autoregressive model. Instead of y_t being a single number in $\mathbb{R}$ it is instead $y_t = (y_{1t}, \dots, y_{kt})^T$, a vector in $\mathbb{R}^k$. The equation for a VAR(p) model is then

$$y_t = v + A_1 y_{t-1} + \dots + A_p y_{t-p} + u_t \text{ where } v = (v_1, \dots, v_k)^T \in \mathbb{R}^k,\ A_j \in \mathbb{R}^{k \times k} \text{ for each } j \in \{1, \dots, p\}$$
$$\text{and } u_t = (u_{1t}, \dots, u_{kt})^T \in \mathbb{R}^k \text{ is a residual such that } \mathbb{E}[u_t] = 0 \text{ and } \text{Cov}(u_t) = \mathbb{E}[u_t u_t^T] = \Sigma_u$$

A VAR(p) model also requires the same stationarity conditions except $\mathbb{E}[y_t] = \mu \in \mathbb{R}^k$ for all t and instead of $\gamma(h) : \mathbb{Z} \to \mathbb{R}$ as the autocovariance function it is $\Gamma(h) : \mathbb{Z} \to \mathbb{R}^{k \times k}$

VAR Forecasting

Forecasting for a VAR model is rather straightforward from the equation above. If $v, A_1, \dots, A_p$ are known then we simply use the expectation of y_t and our previous p forecasts (or starting data points).

$$\widehat{y}_t = \mathbb{E}[y_t] = v + A_1 \widehat{y}_{t-1} + \dots + A_p \widehat{y}_{t-p} \text{ where } \widehat{y}_s = y_s \text{ if it was one of the starting data points}$$

If v, A_1, ..., A_p are unknown then we simply use the above formula substituting in our estimates $\widehat{v}, \widehat{A}_1, \dots, \widehat{A}_p$.

VAR Estimation

In order to estimate the parameters $v, A_1, \dots, A_p$ I will use the multivariate least squares estimator. In order to define this we must first define a few terms.

$$\text{Let } Y = (y_1, \dots, y_T) \in \mathbb{R}^{k\times T}, \text{ then } B = (v, A_1, \dots, A_p) \in \mathbb{R}^{k\times kp+1}$$

$$\text{then } Z_t = \begin{pmatrix} 1 \\ y_t \\ \vdots \\ y_{t-p+1} \end{pmatrix} \in \mathbb{R}^{kp+1} \text{ and } Z = (Z_0, \dots, Z_{T-1}) \in \mathbb{R}^{kp+1\times T}, \text{ and finally } U = (u_1, \dots, u_T) \in \mathbb{R}^{k\times T}$$

Where we are assuming we have starting values $y_{-p+1}, \dots, y_0$. Then the multivariate least squares estimator is given by

$$\hat{B} = YZ^T(ZZ^T)^{-1}$$

From which we can extract $\hat{v}, \hat{A}_1, \dots, \hat{A}_p$ and perform forecasting.

VAR Order Selection

There are several criteria that can be used for VAR order selection. The first of which is the commonly used Akaike's Information Criterion (AIC) which for VAR is given by

$$AIC(m) = \log|\tilde{\Sigma}_u(m)| + \frac{2mk^2}{T} \text{ where } \tilde{\Sigma}_u(m) = \frac{1}{T}Y(I_T - Z^T(ZZ^T)^{-1}Z)Y^T$$

is the estimated covariance matrix from estimating a VAR(m) model

The chosen order $\hat{p}(AIC)$ is such that this criteria is minimized. In fact AIC is not consistent for finding the order of a true VAR process but it still provides a good balance between model complexity and precision so this will be my choice of criteria for model selection here. If a consistent estimate of the order is desired then once can use the Hannan & Quinn criteria (frequentist) or the Schwarz criteria (Bayesian) both of which are described in Lutkepohl [1].

Exploratory Data Analysis

I first begin by importing the necessary packages, loading the data, and making the test-train split.

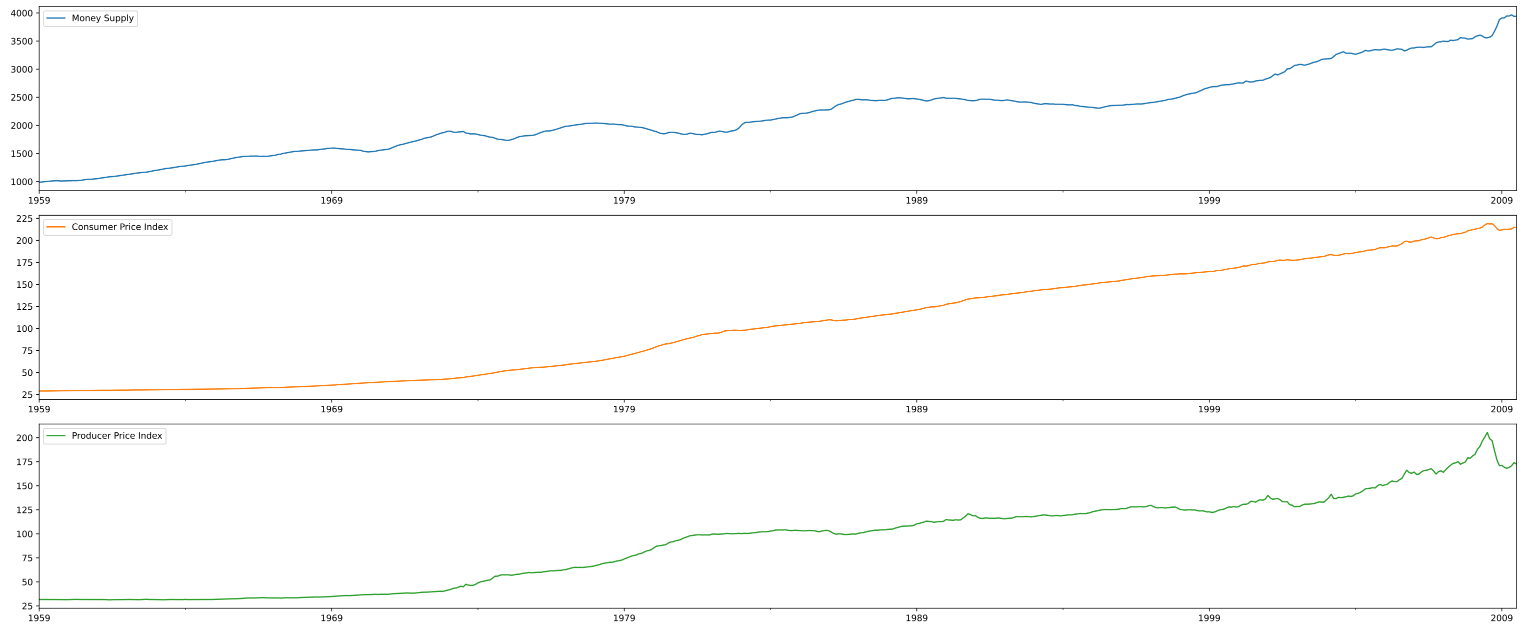
```python
# Import packages
import warnings
import numpy as np
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA

# Read individual series
M2 = pd.read_csv("M2REAL.csv")
CPI = pd.read_csv("CPIAUCSL.csv")
PPI = pd.read_csv("PPIACO.csv")

# Initialize full dataframe
data = M2
```

```python
# Which series to add
add_series = [
  "CPI",
  "PPI"
]

# Merge series
for series in add_series:
  data = pd.merge(data, globals()[series], on = "observation_date")

# Convert to datetime, rename columns, and set index
data["observation_date"] = pd.to_datetime(data["observation_date"])
data = data.rename(columns =
  {
    "observation_date": "Date",
    "M2REAL": "Money Supply",
    "CPIAUCSL": "Consumer Price Index",
    "PPIACO": "Producer Price Index"
  }
).set_index("Date")

# Remove the data after the COVID date
COVID_Date = pd.to_datetime("2020-01-01")
data = data[data.index < COVID_Date]

# Get the number of observations and number of series
# Print important information
n, K = data.shape
names = data.columns.to_list()
print("Data shape:", (n, K))
print("Data columns:")
for name in names:
    print(" - ", name)

print()
n_test = 125
n_train = n - n_test
train_df = data.head(n_train).copy()
test_df = data.tail(n_test).copy()
print("Train data shape:", train_df.shape)
print("Test data shape:", test_df.shape)
```

```
Data shape: (732, 3)
Data columns:
 -  Money Supply
 -  Consumer Price Index
 -  Producer Price Index

Train data shape: (607, 3)
Test data shape: (125, 3)
```

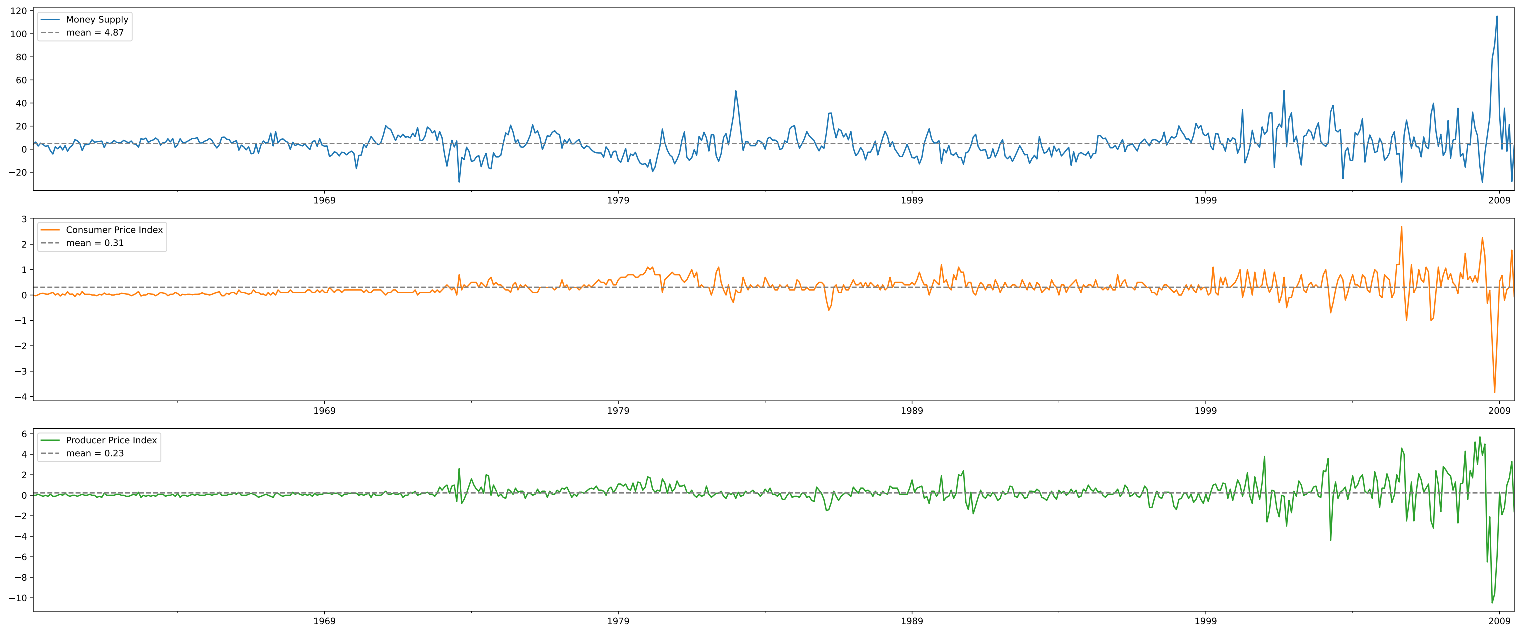
I will now plot the training data to get a sense of what we are looking at.

```
fig, axes = plt.subplots(K, figsize = (24, 10), sharex = True)
train_df.plot(ax = axes, subplots = True)
for axis in axes:
    axis.legend(loc = "upper left")
    axis.tick_params(labelbottom = True)
plt.xlabel("")
plt.tight_layout()
plt.show()
```



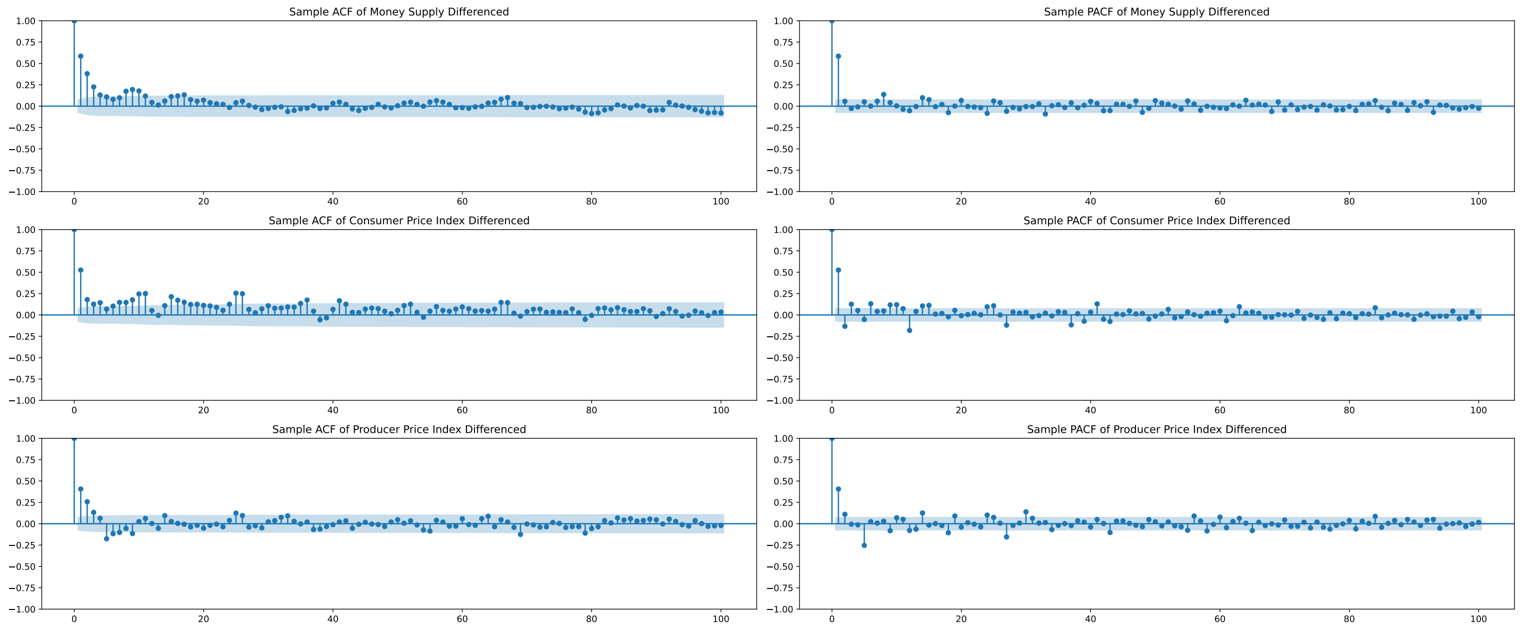
There is a clear, increasing trend in each of the time series, none of them are stationary. So first I will difference each series then check whether additional differencing is needed.

```
train_diff = train_df.diff(1).dropna().copy()

fig, axes = plt.subplots(K, figsize = (24, 10), sharex = True)
train_diff.plot(ax = axes, subplots = True)
for name, axis in zip(names, axes):
    mean_val = train_diff[name].mean()
    axis.axhline(mean_val,
                 linestyle = "--",
                 color = "gray",
                 label = f"mean = {mean_val:.2f}")
    axis.legend(loc="upper left")
    axis.tick_params(labelbottom = True)
plt.xlabel("")
plt.tight_layout()
plt.show()
```



The trend is mostly gone now but it looks like there is still a clear trend in consumer price index and some potential trends in money supply. Before continuing I will look at the sample ACF and PACF of each time series to check whether they need additional differencing.

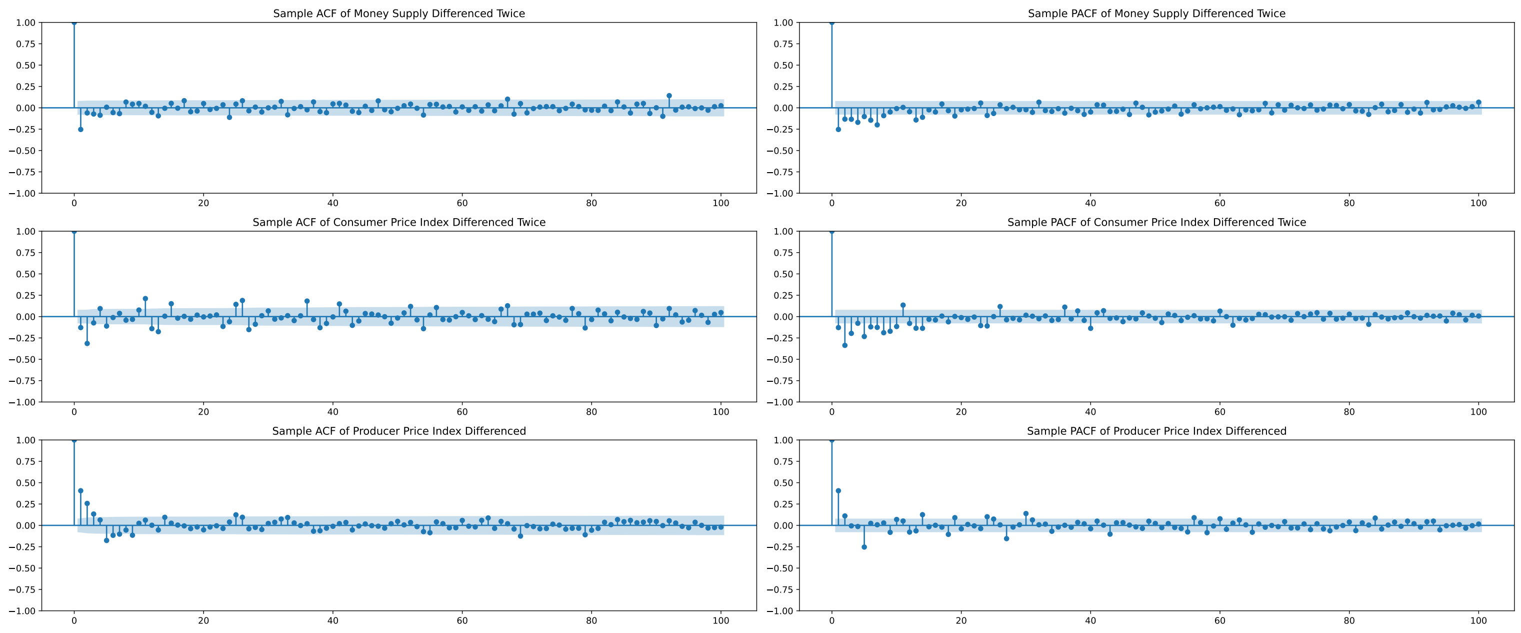
```python
fig, axes = plt.subplots(nrows = K, ncols = 2, figsize = (24, 10))
for i, name in enumerate(names):
    # Plot the ACF in left column
    sm.graphics.tsa.plot_acf(train_diff[name], lags = 100, ax = axes[i, 0])
    axes[i, 0].set_title(f"Sample ACF of {name} Differenced")
    # Plot the PACF in right column
    sm.graphics.tsa.plot_pacf(train_diff[name], lags = 100, ax = axes[i, 1])
    axes[i, 1].set_title(f"Sample PACF of {name} Differenced")
plt.tight_layout()
plt.show()
```



There is a large spike at lag 1 in the sample PACF of money supply with a semi-slow decay in the sample ACF. This suggests that money supply needs additional differencing. Similarly there is a large spike at lag 1 in the sample PACF of consumer price index with a much slower decay in the sample ACF. This suggests that consumer price index definitely needs additional differencing. Producer price index does not need any additional differencing as the sample ACF and

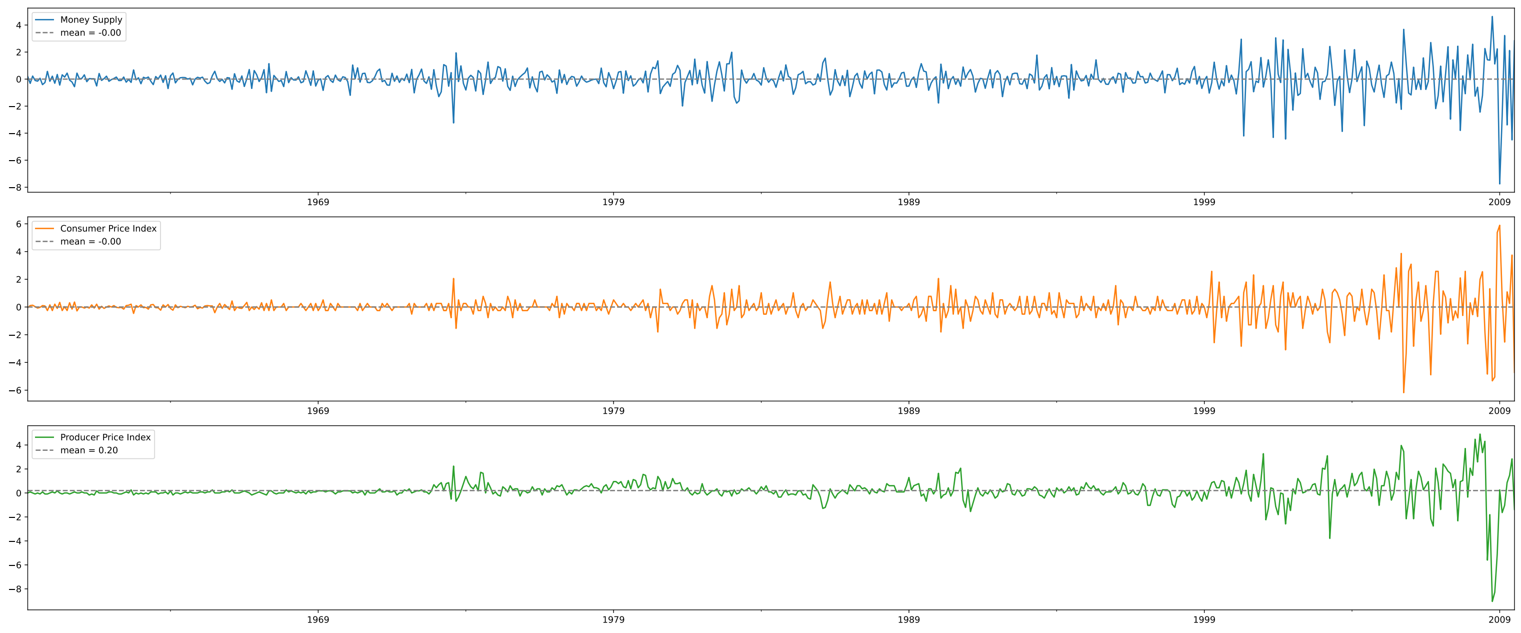
PACF both decay quickly. So I will difference money supply and consumer price index one more time then check the sample ACF and PACF again to be sure I have not overdifferenced. We also know from plotting the data before that the scales of all of these series are different. So I will standardize the data before fitting any models.

```python
# Which columns need another difference
need_double_diff = [
  "Money Supply",
  "Consumer Price Index"
]

# Difference data that needs it
train = train_diff.copy()
for name in need_double_diff:
  train[name] = train[name].diff(1)
train = train.dropna()

# Standardize data
sds = {}
for name in names:
  sds[name] = np.std(train[name])
  train[name] /= sds[name]

fig, axes = plt.subplots(nrows = K, ncols = 2, figsize = (24, 10))
for i, name in enumerate(names):
    # Plot the ACF in left column
    sm.graphics.tsa.plot_acf(train[name], lags = 100, ax = axes[i, 0])
    if name in need_double_diff:
        axes[i, 0].set_title(f"Sample ACF of {name} Differenced Twice")
    else:
        axes[i, 0].set_title(f"Sample ACF of {name} Differenced")
    # Plot the PACF in right column
    sm.graphics.tsa.plot_pacf(train[name], lags = 100, ax = axes[i, 1])
    if name in need_double_diff:
        axes[i, 1].set_title(f"Sample PACF of {name} Differenced Twice")
    else:
        axes[i, 1].set_title(f"Sample PACF of {name} Differenced")
plt.tight_layout()
plt.show()
```



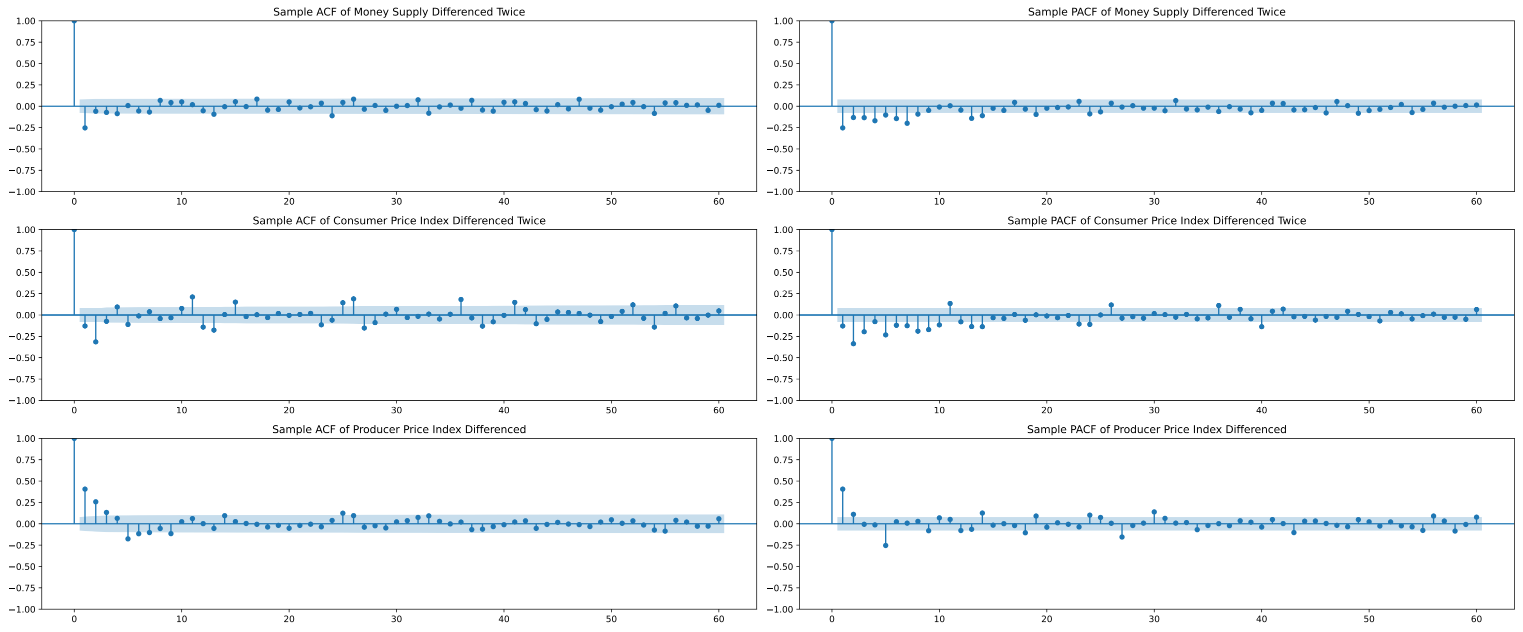
These look much better now. There doesn't seem to be overdifferencing for money supply or consumer price index since there is no very large negative spike in the sample ACF or PACF at lag 1. Now I will take a look at the fully differenced data and check that the trends are gone.

```python
fig, axes = plt.subplots(K, figsize = (24, 10), sharex = True)
train.plot(ax = axes, subplots = True)
for name, axis in zip(names, axes):
    mean_val = train[name].mean()
    axis.axhline(mean_val,
                 linestyle = "--",
                 color = "gray",
                 label = f"mean = {mean_val:.2f}")
    axis.legend(loc="upper left")
    axis.tick_params(labelbottom = True)
plt.xlabel("")
plt.tight_layout()
plt.show()
```



Clearly the trends have been removed, these time series are now stationary. Additionally, they are all on the same scale now. This is an ideal scenario to now fit a model according to the Box-Jenkins methodology. When looking at the sample ACF and PACF before most of the significant lags are rather small with a few significant lags later on, but none really greater 60. First I will take a closer look at the sample ACF and PACF for lags up to 60.

```
fig, axes = plt.subplots(nrows = K, ncols = 2, figsize = (24, 10))
for i, name in enumerate(names):
    # Plot the ACF in left column
    sm.graphics.tsa.plot_acf(train[name], lags = 60, ax = axes[i, 0])
    if name in need_double_diff:
        axes[i, 0].set_title(f"Sample ACF of {name} Differenced Twice")
    else:
        axes[i, 0].set_title(f"Sample ACF of {name} Differenced")
    # Plot the PACF in right column
    sm.graphics.tsa.plot_pacf(train[name], lags = 60, ax = axes[i, 1])
    if name in need_double_diff:
        axes[i, 1].set_title(f"Sample PACF of {name} Differenced Twice")
    else:
        axes[i, 1].set_title(f"Sample PACF of {name} Differenced")
plt.tight_layout()
plt.show()
```



For money supply I will clearly want q to be no larger than 1 and it is clear I won't want p to be larger than 13 since all of the lags after are negligible. For consumer price index most of the significant lags are between 1 and 13 but there are a few lags further out that exceed the significance region (for example lag 36 in the sample ACF and lag 40 in the sample PACF). For producer price index most of the significant lags are less than 10 but here there are also a few lags further out that exceed the significance region (for example lag 27 in the sample PACF). We can use these observations to choose a set of potential models and use a model selection criteria to select parameters.

Fitting ARMA Models

I will use AIC to for model selection here since it aims to provide a balance between modelcomplexity and goodness of fit, ideal for forecasting.

Before continuing I write a function for undifferencing the data when we want to make predictions so that the scale of the predictions is the same as the original data.

```
def undifference(name, where, m):
  # Check that the time series is valid
  assert name in names

  # If it needs double differencing apply
  # a double undifference transformation
  if name in need_double_diff:
    undiff = (where[name].cumsum() + \
              np.repeat(train_diff[name].values[-1], m)).cumsum() + \
              np.repeat(train_df[name].values[-1], m)
  # Otherwise just apply a single
  # undifference transformation
  else:
    undiff = where[name].cumsum() + \
              np.repeat(train_df[name].values[-1], m)
  return(undiff)
```

Now I move on to fit the ARMA models.

```
# Parameters from EDA
ARMA_params = {
  "Money Supply": [
    (p, 0, q)
    for p in [0, 1, 4, 7, 13]
    for q in [0, 1]
  ],
  "Consumer Price Index": [
    (p, 0, q)
    for p in [0, 2, 5, 9, 14, 24, 40]
    for q in [0, 2, 11, 13, 26, 36]
  ],
  "Producer Price Index": [
    (p, 0, q)
    for p in [0, 1, 5, 27]
    for q in [0, 2, 5]
  ]
}

# Initialize model dictionaries
ARMA_models = {}
fcasts_ARMA = {}
MSEs_ARMA = {}

for name in names:
  # Get series and parameters
  series = train[name].values
  best_model = {"AIC": np.inf}
  params = ARMA_params[name]

  with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    for param in params:
      try:
        # Fit the ARMA(p,q) model
        model = ARIMA(series,
                      order = param
                      ).fit()
        # Perform Model Selection
        # Criteria with AIC
        AIC = model.aic
        if AIC < best_model["AIC"]:
          best_model["AIC"] = AIC
          best_model["param"] = param
          best_model["Model"] = model
      # If an error occurs print the
      # statement and continue
      except Exception as e:
        print(f"{name} ARMA({param}) failed: {e}")
        continue
    # Get forecasts and MSE for best models
    ARMA_models[name] = best_model
    fcast_standard = best_model["Model"].get_forecast(n_test).predicted_mean
    fcasts_ARMA[name] = fcast_standard*sds[name]
    fcasts_ARMA[name] = undifference(name, fcasts_ARMA, n_test)
    MSEs_ARMA[name] = np.mean((fcasts_ARMA[name] - test_df[name].values)**2)
  del best_model["Model"]
  print(f"{name} Chosen parameters: ", ARMA_models[name]["param"])
```

```
Money Supply Chosen parameters:  (1, 0, 1)
Consumer Price Index Chosen parameters:  (14, 0, 13)
Producer Price Index Chosen parameters:  (27, 0, 5)
```

So the best ARMA model for the double differenced money supply data according to AIC is ARMA(1, 1), the best ARMA model for the double differenced consumer price index data according to AIC is ARMA(14, 13), and the best ARMA model for the once differenced producer price index data according to AIC is ARMA(27, 5). I will now move on to fitting the VAR model.

Fitting VAR Model

I will use the same model selection criteria as before, AIC. First I implement the multivariate least squares estimator, forecasting, and AIC from scratch. I also implemented the Hannan & Quinn (frequentist) and Schwarz (Bayesian) criterions which are consistent for finding the true order of a VAR(p) process while AIC is not. However, AIC still performs well when balancing model complexity and accuracy so I will stick with AIC for my selection criteria.

```
def fit_VAR(Y, p):
  # Get dimensions of data
  K, T = Y.shape
  # Raise error if not enough
  # data to fit model
  if T <= p:
    raise ValueError("Lag order p should be less than time series length T")
  # Since we need points
  # y_t-p+1 up to y_0
  # we can't use the starting p
  # points of our data
  useable = T - p
  # Initialize Z matrix
  Z = np.ones((K*p + 1, useable))

  for t in range(p, T):
    # Get the Zt vectors
    Zt = []
    for j in range(1, p + 1):
      Zt.append(Y[:,t-j])
    # Create Z matrix
    Z[1:,t-p] = np.concatenate(Zt)
  # Find multivariate least
  # squares estimator
  y_useable = Y[:,p:]
  B = y_useable @ Z.T @ np.linalg.inv(Z @ Z.T)
  return(Z, B)

def extract_VAR_coeffs(B, K, p):
  # Get v (intercept)
  v = B[:,0]
  # Get coefficient
  # matrices
  A = []
  for j in range(p):
    start = 1 + j*K
    end = start + K
    A.append(B[:,start:end])
  return(v, A)

def predict_next_VAR(lags, v, A):
  p = len(A)
  y_next = v.copy()
  # Implement one step
  # forecasting
  for j in range(p):
    y_next += A[j] @ lags[-1-j]
  return(y_next)

def get_predictions_VAR(y0, v, A, m):
  K = v.shape[0]
  lags = y0
  # Initialize the predictions
  pred = np.zeros((K, m))
  # Repeatedly get next prediction
  for t in range(m):
    y_pred = predict_next_VAR(lags, v, A)
    pred[:,t] = y_pred
    lags.append(y_pred)
    lags.pop(0)
  return(pred)

# Get covariance matrix
# and different criterion
def get_covmat(Z, Y, p):
  y_useable = Y[:,p:]
```

```python
  _, T = y_useable.shape
  Sigma = (y_useable @ (np.identity(T) - Z.T @ np.linalg.inv(Z @ Z.T) @ Z) @ y_useable.T)/T
  return(Sigma)

def get_AIC(Sigma, p, Y):
  y_useable = Y[:,p:]
  K, T = y_useable.shape
  AIC = np.log(np.linalg.det(Sigma)) + (2*p*(K**2))/T
  return(AIC)

def get_HQ(Sigma, p, Y):
  y_useable = Y[:,p:]
  K, T = y_useable.shape
  HQ = np.log(np.linalg.det(Sigma)) + ((2*np.log(np.log(T)))*p*(K**2))/T
  return(HQ)

def get_SC(Sigma, p, Y):
  y_useable = Y[:,p:]
  K, T = y_useable.shape
  SC = np.log(np.linalg.det(Sigma)) + ((np.log(T))*p*(K**2))/T
  return(SC)
```

I will use 40 as the maximum lag allowed for the VAR(p) model since this was the highest value of p allowed to be chosen in the ARMA modeling. This is so that I don't give additional power to VAR simply by using a larger order p.

```python
# Create Y matrix
# Note that I am staying
# consistent with my notation
# before by making it a kxT matrix
Y = train.to_numpy().T
pvals = range(1, 41)
best_model = {"AIC": np.inf}

import warnings
with warnings.catch_warnings():
  warnings.simplefilter("ignore")
  for p in pvals:
    try:
      # Fit the VAR(p) model
      Z, B = fit_VAR(Y, p)
      v, A = extract_VAR_coeffs(B, Y.shape[0], p)
      Sigma = get_covmat(Z, Y, p)
      # Perform model selection
      # with AIC
      AIC = get_AIC(Sigma, p, Y)
      if AIC < best_model["AIC"]:
        best_model["p"] = p
        best_model["AIC"] = AIC
        best_model["Z"] = Z
        best_model["B"] = B
        best_model["v"] = v
        best_model["A"] = A
        best_model["Sigma"] = Sigma
    # If something went wrong
    # print the error and continue
    except Exception as e:
      print(f"VAR({p}) failed: {e}")
      continue

print(f"Best model found: VAR({best_model["p"]})")
print("AIC:", best_model["AIC"])
print("\nEstimated Intercept v:")
print( v)
print("\nEstimated Coefficient Matrices:")
for j in range(len(A)):
  print(f"A[{j+1}]")
  print(A[j])
```

```
Best model found: VAR(40)
AIC: -2.5480833696459007

Estimated Intercept v:
[-0.06537982  0.06976205  0.0893726 ]

Estimated Coefficient Matrices:
A[1]
[[-0.53801146 -0.01294283 -0.14659832]
 [-0.10632013 -0.76924583  0.29124425]
 [-0.0777183   0.03650318  0.37969598]]
A[2]
[[-0.35817095  0.17165142 -0.16602347]
 [-0.15500819 -1.04743517  0.1761996 ]
 [-0.05471487 -0.02593942  0.17741368]]
A[3]
[[-0.31697552  0.27084699  0.10347613]
 [-0.11860949 -1.01088648  0.01936044]
 [-0.1140664  -0.09451689  0.04775203]]
A[4]
[[-0.4019932   0.05457774  0.02707366]
 [-0.19970617 -0.90051025 -0.0700718 ]
 [-0.12974531  0.22089321 -0.10421952]]
A[5]
[[-0.32114409  0.03985709  0.17487004]
 [-0.05876626 -0.95222469 -0.08938273]
 [-0.02330029  0.01773203 -0.19604015]]
A[6]
[[-0.31057402 -0.0092662   0.02824497]
 [-0.03353301 -0.69831557 -0.0179019 ]
 [-0.0788055   0.19793662 -0.04210967]]
A[7]
[[-0.38287072 -0.07320378  0.03902169]
 [-0.02272725 -0.50934761 -0.0672831 ]
 [-0.08831215  0.37221709 -0.11276247]]
A[8]
[[-0.17032223 -0.06623974  0.17778961]
 [-0.17741722 -0.57163218 -0.13345522]
 [-0.16064255  0.34001948 -0.03585934]]
A[9]
[[-0.04314898 -0.0525632   0.02561272]
```

```
 [-0.14215943 -0.38069808 -0.13578138]
 [-0.1423564   0.41708349 -0.16348675]]
A[10]
[[-0.09749791 -0.22236293  0.09580379]
 [-0.04882797 -0.18935249 -0.10829654]
 [-0.09819124  0.45079144 -0.03494108]]
A[11]
[[-0.22199399 -0.36712279 -0.02963738]
 [ 0.0229137   0.07551942  0.02904218]
 [ 0.00495355  0.53115228  0.13405021]]
A[12]
[[-0.13006829 -0.1539809   0.09074659]
 [-0.13711895 -0.15352876 -0.11000133]
 [-0.1752058   0.23779427  0.04836646]]
A[13]
[[-0.20776615 -0.16200689 -0.0962046 ]
 [-0.06257706 -0.22394632 -0.00159799]
 [-0.06682231  0.07665672  0.05621125]]
A[14]
[[-0.15344157 -0.12235331  0.03025677]
 [-0.0531649  -0.24808294  0.03118448]
 [-0.19812172 -0.00341587  0.07730361]]
A[15]
[[ 0.01275196  0.06825763 -0.04226317]
 [-0.14372489 -0.28103085  0.08927892]
 [-0.17757833 -0.15048324 -0.00772264]]
A[16]
[[ 0.08236864  0.28726538 -0.09689111]
 [-0.21278013 -0.33535484 -0.02401467]
 [-0.26137872 -0.18712594  0.06140949]]
A[17]
[[-0.03027579  0.1430364  -0.16466841]
 [-0.02712566 -0.22770672  0.15562451]
 [-0.05956654 -0.18852734  0.19593715]]
A[18]
[[ 0.04231886  0.24337226  0.12997969]
 [-0.07495284 -0.25221195 -0.15656514]
 [-0.12970246 -0.23432783 -0.14597237]]
A[19]
[[ 0.05200221  0.4260211  -0.23395926]
 [-0.10252994 -0.21101601  0.15515824]
 [-0.14916202 -0.25820865  0.21614869]]
A[20]
[[ 0.11239983  0.46015254 -0.00655367]
 [-0.16249734 -0.12804531 -0.11054905]
 [-0.14249547 -0.24276109 -0.08897442]]
A[21]
[[ 0.10379992  0.56948882 -0.0888014 ]
 [-0.16764495 -0.23424649 -0.0012261 ]
 [-0.1556099  -0.14089615 -0.00328294]]
A[22]
[[ 0.13128723  0.33197789  0.18533531]
 [-0.15282828 -0.15520516 -0.12427579]
 [-0.15024179 -0.06888712 -0.18093101]]
A[23]
```

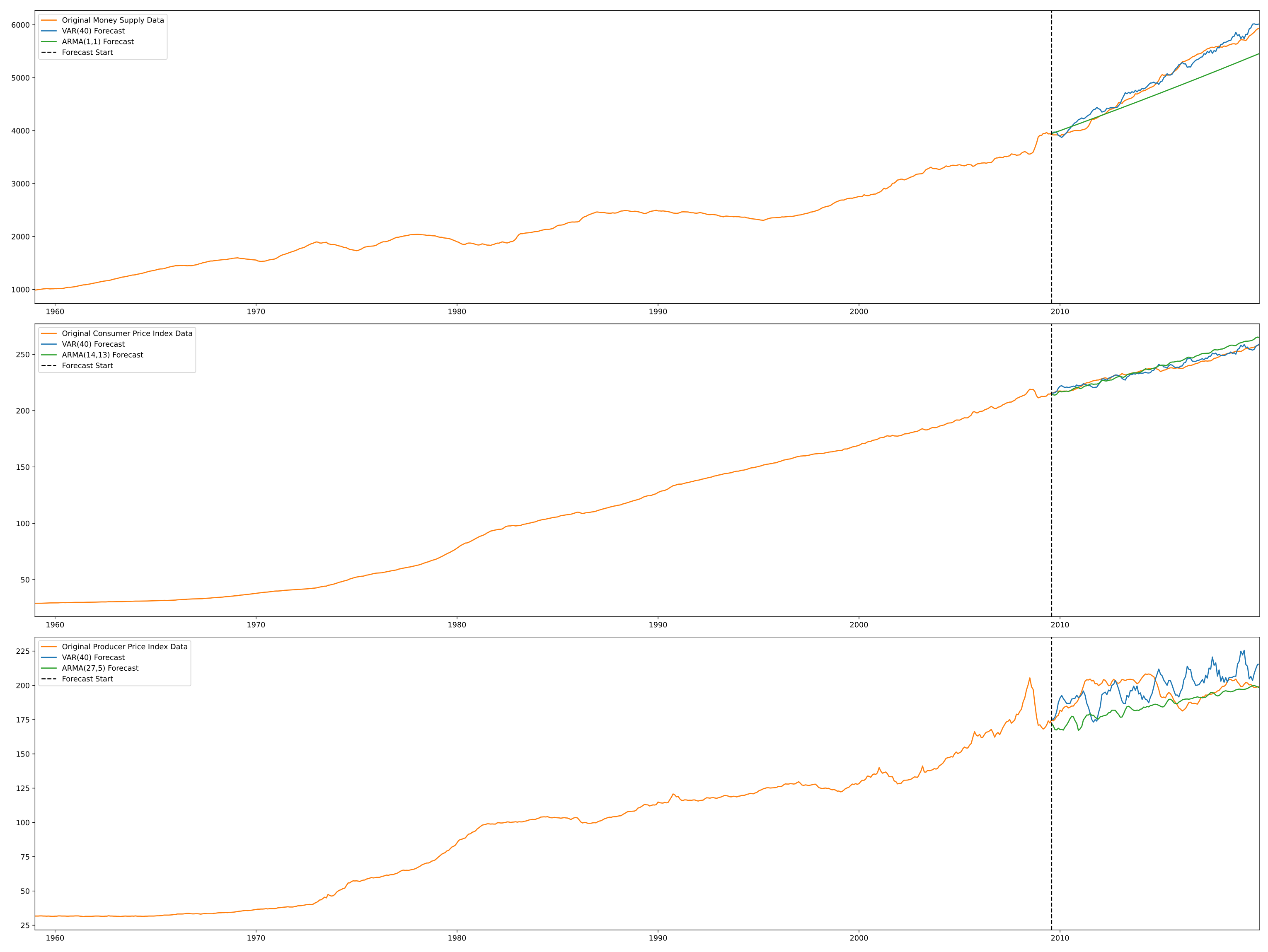
```
[[ 0.10034061  0.4952017  -0.0976893 ]
 [-0.04775567 -0.11636945  0.07382614]
 [ 0.06177472  0.00756155 -0.05386411]]
A[24]
[[-1.33975610e-01  1.58969624e-01 -1.29513085e-04]
 [-6.69840484e-02 -1.58362216e-02 -2.89737836e-02]
 [-5.30221558e-02  8.83694327e-02  3.30902776e-02]]
A[25]
[[-0.24924874 -0.15802751  0.13276271]
 [-0.00034884  0.07379944  0.03685364]
 [-0.07538441  0.13944864  0.04370261]]
A[26]
[[-0.24594207 -0.11979316 -0.10502044]
 [ 0.02194187  0.0258493   0.12910577]
 [-0.04996424  0.17261572  0.12152341]]
A[27]
[[-0.07410362  0.0193707   0.11112717]
 [-0.12751594 -0.32090791 -0.06219658]
 [-0.11955326 -0.11192617 -0.13956136]]
A[28]
[[-0.12374596 -0.18198623  0.15658213]
 [-0.09376939 -0.14114296 -0.00034571]
 [-0.14511782 -0.033212    0.02428649]]
A[29]
[[-0.15266115 -0.22606831  0.21703957]
 [ 0.020646    0.01292545 -0.29904421]
 [-0.0023136  -0.00853163 -0.13725851]]
A[30]
[[-0.16200497 -0.29988067 -0.07005634]
 [ 0.00839909 -0.02520973  0.33536051]
 [ 0.06010879  0.15826135  0.25967462]]
A[31]
[[-0.11142766 -0.13138314 -0.00177914]
 [-0.06266198 -0.100924   -0.08653726]
 [-0.00906921  0.01110952  0.1108194 ]]
A[32]
[[-0.07885563 -0.07499133 -0.17148554]
 [-0.06764266 -0.19204104  0.19141408]
 [-0.06359219  0.03526155  0.08961643]]
A[33]
[[-0.20385787 -0.12313218 -0.00874625]
 [ 0.00510481 -0.06203137 -0.05525613]
 [ 0.07100758  0.07915726  0.19259784]]
A[34]
[[-0.10638904 -0.04869983 -0.02287984]
 [-0.01254668 -0.03858664 -0.11975657]
 [ 0.06963919  0.24947264 -0.23330263]]
A[35]
[[-0.11153854  0.08671488  0.05207978]
 [ 0.05535855 -0.1118117  -0.03094744]
 [ 0.17999604  0.15548466 -0.00137924]]
A[36]
[[-0.26963822 -0.26032013  0.06831644]
 [ 0.12309508  0.22356823 -0.06344621]
 [ 0.13864807  0.39227954 -0.09179176]]
```

```
A[37]
[[-0.18069304 -0.03543735 -0.07789856]
 [ 0.02281971 -0.03639211 -0.08529198]
 [-0.04366036 -0.09850276  0.08050917]]
A[38]
[[-0.18049139 -0.18129413  0.05164811]
 [ 0.09865283  0.09763022  0.04602756]
 [ 0.08338277  0.12672127 -0.03682198]]
A[39]
[[ 0.04081164  0.23282963 -0.12911591]
 [-0.00066728 -0.18981554  0.05051502]
 [-0.09028233 -0.15102491  0.13686934]]
A[40]
[[-0.00211167  0.10923984  0.19933415]
 [-0.03857954 -0.29260929 -0.06394886]
 [ 0.00515232 -0.11950883 -0.04245543]]
```

So the best VAR model for the multivariate time series of double differenced money supply, double differenced consumer price index, and once differenced producer price index is VAR(40).

```python
# Get parameters
p = best_model["p"]
Z = best_model["Z"]
B = best_model["B"]
v = best_model["v"]
A = best_model["A"]
# Get the most recent values
lags = [Y[:,-t] for t in range(1, p + 1)][::-1]
# Make predictions
fcast_standard = get_predictions_VAR(lags, v, A, n_test)
fcasts_VAR = {}
MSEs_VAR = {}

# Put predictions in the scale of the original data
# and plot predictions and get MSEs
fig, axs = plt.subplots(K, 1, figsize=(24, 18), sharex = True)
for j in range(len(names)):
  name = names[j]
  fcasts_VAR[name] = fcast_standard[j,:]*sds[name]
  fcasts_VAR[name] = undifference(name, fcasts_VAR, n_test)

  fcast_VAR = fcasts_VAR[name]
  fcast_ARMA = fcasts_ARMA[name]

  MSEs_VAR[name] = np.mean((fcast_VAR - test_df[name].values)**2)

  ax = axs[j]
  ax.plot(data.index, data[name].values, label = f"Original {name} Data", color = "C1")
  ax.plot(data.index[-n_test:], fcast_VAR, label = f"VAR({p}) Forecast", color = "C0")
  ax.plot(data.index[-n_test:], fcast_ARMA, label = f"ARMA({ARMA_models[name]["param"][0]},{ARMA_models
  ax.axvline(x = data.index[-n_test], color = "black", linestyle = "--", label = "Forecast Start")
  ax.set_xlim(data.index[0], data.index[-1])
  ax.tick_params(labelbottom = True)
  ax.legend(loc = "upper left")
```

```
plt.tight_layout()
plt.show()
```



The benefit which VAR is providing over independent ARMA modeling is very clear, especially in the money supply time series. Now I will show the mean squared prediction errors for each time series and the overall mean squared prediction error.

```
print(f"Independant ARMA Model MSEs:")
for name in names:
    print(f"MSE for {name} {MSEs_ARMA[name]:.2f}")
print(f"\nIndependant Model Overall MSE: {np.mean(list(MSEs_ARMA.values())):.2f}")

print(f"\n\nVAR MSEs:")
for name in names:
    print(f"MSE for {name} {MSEs_VAR[name]:.2f}")
print(f"\nVAR Overall MSE: {np.mean(list(MSEs_VAR.values())):.2f}")
```

```
Independant ARMA Model MSEs:
MSE for Money Supply 86848.55
MSE for Consumer Price Index 21.60
MSE for Producer Price Index 229.68
```

```
Independant Model Overall MSE: 29033.28


VAR MSEs:
MSE for Money Supply 10369.81
MSE for Consumer Price Index 8.28
MSE for Producer Price Index 183.16

VAR Overall MSE: 3520.42
```

VAR clearly does much better in every time series. ARMA does the worst in the money supply time series so I will show what the overall mean squared prediction error is if we exclude that series.

```python
print(f"Independant Model Overall MSE Excluding Money Supply: {np.mean([MSEs_ARMA[name] for name in nam
print(f"VAR Overall MSE Excluding Money Supply: {np.mean([MSEs_VAR[name] for name in names if name != "
```

```
Independant Model Overall MSE Excluding Money Supply: 125.64
VAR Overall MSE Excluding Money Supply: 95.72
```

Even ignoring the worst performing series for ARMA this is a very significant improvement by VAR.

Validating That My Implementation Worked Properly

I will now compare my VAR results with those from the statsmodels package.

```python
from statsmodels.tsa.api import VAR

with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    sm_VAR = VAR(train).fit(p)

print(sm_VAR.summary())
```

```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Mon, 07, Jul, 2025
Time:                     17:05:03
--------------------------------------------------------------------
No. of Equations:         3.00000    BIC:                   0.248851
Nobs:                     565.000    HQIC:                  -1.44991
Log likelihood:          -1325.27    FPE:                  0.0806803
AIC:                     -2.53746    Det(Omega_mle):       0.0450755
--------------------------------------------------------------------
Results for equation Money Supply
===========================================================================================
                              coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------------------
const                           -0.065380         0.047002           -1.391           0.164
L1.Money Supply                 -0.538011         0.061180           -8.794           0.000
L1.Consumer Price Index         -0.012943         0.082665           -0.157           0.876
```

```
L1.Producer Price Index        -0.146598         0.064852          -2.261          0.024
L2.Money Supply                -0.358171         0.072319          -4.953          0.000
L2.Consumer Price Index         0.171651         0.105564           1.626          0.104
L2.Producer Price Index        -0.166023         0.067106          -2.474          0.013
L3.Money Supply                -0.316976         0.077913          -4.068          0.000
L3.Consumer Price Index         0.270847         0.133459           2.029          0.042
L3.Producer Price Index         0.103476         0.067778           1.527          0.127
L4.Money Supply                -0.401993         0.081619          -4.925          0.000
L4.Consumer Price Index         0.054578         0.152232           0.359          0.720
L4.Producer Price Index         0.027074         0.067935           0.399          0.690
L5.Money Supply                -0.321144         0.088420          -3.632          0.000
L5.Consumer Price Index         0.039857         0.173368           0.230          0.818
L5.Producer Price Index         0.174870         0.068105           2.568          0.010
L6.Money Supply                -0.310574         0.091087          -3.410          0.001
L6.Consumer Price Index        -0.009266         0.191252          -0.048          0.961
L6.Producer Price Index         0.028245         0.069188           0.408          0.683
L7.Money Supply                -0.382871         0.092931          -4.120          0.000
L7.Consumer Price Index        -0.073204         0.203639          -0.359          0.719
L7.Producer Price Index         0.039022         0.069367           0.563          0.574
L8.Money Supply                -0.170322         0.095088          -1.791          0.073
L8.Consumer Price Index        -0.066240         0.212272          -0.312          0.755
L8.Producer Price Index         0.177790         0.067739           2.625          0.009
L9.Money Supply                -0.043149         0.096835          -0.446          0.656
L9.Consumer Price Index        -0.052563         0.219567          -0.239          0.811
L9.Producer Price Index         0.025613         0.068146           0.376          0.707
L10.Money Supply               -0.097498         0.096707          -1.008          0.313
L10.Consumer Price Index       -0.222363         0.222275          -1.000          0.317
L10.Producer Price Index        0.095804         0.069091           1.387          0.166
L11.Money Supply               -0.221994         0.096635          -2.297          0.022
L11.Consumer Price Index       -0.367123         0.225856          -1.625          0.104
L11.Producer Price Index       -0.029637         0.069332          -0.427          0.669
L12.Money Supply               -0.130068         0.097373          -1.336          0.182
L12.Consumer Price Index       -0.153981         0.225739          -0.682          0.495
L12.Producer Price Index        0.090747         0.070643           1.285          0.199
L13.Money Supply               -0.207766         0.098368          -2.112          0.035
L13.Consumer Price Index       -0.162007         0.225240          -0.719          0.472
L13.Producer Price Index       -0.096205         0.074011          -1.300          0.194
L14.Money Supply               -0.153442         0.099074          -1.549          0.121
L14.Consumer Price Index       -0.122353         0.225657          -0.542          0.588
L14.Producer Price Index        0.030257         0.073998           0.409          0.683
L15.Money Supply                0.012752         0.100209           0.127          0.899
L15.Consumer Price Index        0.068258         0.225573           0.303          0.762
L15.Producer Price Index       -0.042263         0.075819          -0.557          0.577
L16.Money Supply                0.082369         0.100121           0.823          0.411
L16.Consumer Price Index        0.287265         0.225038           1.277          0.202
L16.Producer Price Index       -0.096891         0.075694          -1.280          0.201
L17.Money Supply               -0.030276         0.100256          -0.302          0.763
L17.Consumer Price Index        0.143036         0.224976           0.636          0.525
L17.Producer Price Index       -0.164668         0.079389          -2.074          0.038
L18.Money Supply                0.042319         0.099963           0.423          0.672
L18.Consumer Price Index        0.243372         0.224163           1.086          0.278
L18.Producer Price Index        0.129980         0.079308           1.639          0.101
L19.Money Supply                0.052002         0.099856           0.521          0.603
L19.Consumer Price Index        0.426021         0.223584           1.905          0.057
L19.Producer Price Index       -0.233959         0.078942          -2.964          0.003
```

```
L20.Money Supply                 0.112400         0.099460            1.130             0.258
L20.Consumer Price Index         0.460153         0.222943            2.064             0.039
L20.Producer Price Index        -0.006554         0.080998           -0.081             0.936
L21.Money Supply                 0.103800         0.099750            1.041             0.298
L21.Consumer Price Index         0.569489         0.221156            2.575             0.010
L21.Producer Price Index        -0.088801         0.081649           -1.088             0.277
L22.Money Supply                 0.131287         0.099992            1.313             0.189
L22.Consumer Price Index         0.331978         0.220096            1.508             0.131
L22.Producer Price Index         0.185335         0.081818            2.265             0.023
L23.Money Supply                 0.100341         0.100116            1.002             0.316
L23.Consumer Price Index         0.495202         0.219722            2.254             0.024
L23.Producer Price Index        -0.097689         0.082125           -1.190             0.234
L24.Money Supply                -0.133976         0.100712           -1.330             0.183
L24.Consumer Price Index         0.158970         0.220872            0.720             0.472
L24.Producer Price Index        -0.000130         0.082807           -0.002             0.999
L25.Money Supply                -0.249249         0.101175           -2.464             0.014
L25.Consumer Price Index        -0.158028         0.221112           -0.715             0.475
L25.Producer Price Index         0.132763         0.082728            1.605             0.109
L26.Money Supply                -0.245942         0.102122           -2.408             0.016
L26.Consumer Price Index        -0.119793         0.221261           -0.541             0.588
L26.Producer Price Index        -0.105020         0.081822           -1.284             0.199
L27.Money Supply                -0.074104         0.101962           -0.727             0.467
L27.Consumer Price Index         0.019371         0.221618            0.087             0.930
L27.Producer Price Index         0.111127         0.081068            1.371             0.170
L28.Money Supply                -0.123746         0.100941           -1.226             0.220
L28.Consumer Price Index        -0.181986         0.222468           -0.818             0.413
L28.Producer Price Index         0.156582         0.080992            1.933             0.053
L29.Money Supply                -0.152661         0.100765           -1.515             0.130
L29.Consumer Price Index        -0.226068         0.223050           -1.014             0.311
L29.Producer Price Index         0.217040         0.080511            2.696             0.007
L30.Money Supply                -0.162005         0.100169           -1.617             0.106
L30.Consumer Price Index        -0.299881         0.222762           -1.346             0.178
L30.Producer Price Index        -0.070056         0.081775           -0.857             0.392
L31.Money Supply                -0.111428         0.099811           -1.116             0.264
L31.Consumer Price Index        -0.131383         0.221297           -0.594             0.553
L31.Producer Price Index        -0.001779         0.083417           -0.021             0.983
L32.Money Supply                -0.078856         0.099607           -0.792             0.429
L32.Consumer Price Index        -0.074991         0.216195           -0.347             0.729
L32.Producer Price Index        -0.171486         0.084232           -2.036             0.042
L33.Money Supply                -0.203858         0.097718           -2.086             0.037
L33.Consumer Price Index        -0.123132         0.210133           -0.586             0.558
L33.Producer Price Index        -0.008746         0.084348           -0.104             0.917
L34.Money Supply                -0.106389         0.095616           -1.113             0.266
L34.Consumer Price Index        -0.048700         0.199762           -0.244             0.807
L34.Producer Price Index        -0.022880         0.085104           -0.269             0.788
L35.Money Supply                -0.111539         0.092742           -1.203             0.229
L35.Consumer Price Index         0.086715         0.187502            0.462             0.644
L35.Producer Price Index         0.052080         0.085975            0.606             0.545
L36.Money Supply                -0.269638         0.089882           -3.000             0.003
L36.Consumer Price Index        -0.260320         0.173928           -1.497             0.134
L36.Producer Price Index         0.068316         0.085723            0.797             0.425
L37.Money Supply                -0.180693         0.084818           -2.130             0.033
L37.Consumer Price Index        -0.035437         0.158227           -0.224             0.823
L37.Producer Price Index        -0.077899         0.084755           -0.919             0.358
L38.Money Supply                -0.180491         0.082494           -2.188             0.029
```

```
L38.Consumer Price Index       -0.181294         0.141822           -1.278           0.201
L38.Producer Price Index        0.051648         0.084489            0.611           0.541
L39.Money Supply                0.040812         0.076094            0.536           0.592
L39.Consumer Price Index        0.232830         0.116434            2.000           0.046
L39.Producer Price Index       -0.129116         0.083692           -1.543           0.123
L40.Money Supply               -0.002112         0.066461           -0.032           0.975
L40.Consumer Price Index        0.109240         0.089326            1.223           0.221
L40.Producer Price Index        0.199334         0.084140            2.369           0.018
===========================================================================================

Results for equation Consumer Price Index
===========================================================================================
                              coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------------------
const                           0.069762         0.042254            1.651           0.099
L1.Money Supply                -0.106320         0.054999           -1.933           0.053
L1.Consumer Price Index        -0.769246         0.074314          -10.351           0.000
L1.Producer Price Index         0.291244         0.058300            4.996           0.000
L2.Money Supply                -0.155008         0.065014           -2.384           0.017
L2.Consumer Price Index        -1.047435         0.094900          -11.037           0.000
L2.Producer Price Index         0.176200         0.060327            2.921           0.003
L3.Money Supply                -0.118609         0.070042           -1.693           0.090
L3.Consumer Price Index        -1.010886         0.119977           -8.426           0.000
L3.Producer Price Index         0.019360         0.060931            0.318           0.751
L4.Money Supply                -0.199706         0.073374           -2.722           0.006
L4.Consumer Price Index        -0.900510         0.136853           -6.580           0.000
L4.Producer Price Index        -0.070072         0.061072           -1.147           0.251
L5.Money Supply                -0.058766         0.079488           -0.739           0.460
L5.Consumer Price Index        -0.952225         0.155854           -6.110           0.000
L5.Producer Price Index        -0.089383         0.061225           -1.460           0.144
L6.Money Supply                -0.033533         0.081885           -0.410           0.682
L6.Consumer Price Index        -0.698316         0.171931           -4.062           0.000
L6.Producer Price Index        -0.017902         0.062198           -0.288           0.773
L7.Money Supply                -0.022727         0.083543           -0.272           0.786
L7.Consumer Price Index        -0.509348         0.183067           -2.782           0.005
L7.Producer Price Index        -0.067283         0.062360           -1.079           0.281
L8.Money Supply                -0.177417         0.085482           -2.075           0.038
L8.Consumer Price Index        -0.571632         0.190828           -2.996           0.003
L8.Producer Price Index        -0.133455         0.060896           -2.192           0.028
L9.Money Supply                -0.142159         0.087053           -1.633           0.102
L9.Consumer Price Index        -0.380698         0.197386           -1.929           0.054
L9.Producer Price Index        -0.135781         0.061262           -2.216           0.027
L10.Money Supply               -0.048828         0.086937           -0.562           0.574
L10.Consumer Price Index       -0.189352         0.199821           -0.948           0.343
L10.Producer Price Index       -0.108297         0.062111           -1.744           0.081
L11.Money Supply                0.022914         0.086872            0.264           0.792
L11.Consumer Price Index        0.075519         0.203040            0.372           0.710
L11.Producer Price Index        0.029042         0.062328            0.466           0.641
L12.Money Supply               -0.137119         0.087536           -1.566           0.117
L12.Consumer Price Index       -0.153529         0.202935           -0.757           0.449
L12.Producer Price Index       -0.110001         0.063507           -1.732           0.083
L13.Money Supply               -0.062577         0.088431           -0.708           0.479
L13.Consumer Price Index       -0.223946         0.202486           -1.106           0.269
L13.Producer Price Index       -0.001598         0.066534           -0.024           0.981
L14.Money Supply               -0.053165         0.089065           -0.597           0.551
```

```
L14.Consumer Price Index        -0.248083           0.202861             -1.223              0.221
L14.Producer Price Index         0.031184           0.066523              0.469              0.639
L15.Money Supply                -0.143725           0.090086             -1.595              0.111
L15.Consumer Price Index        -0.281031           0.202786             -1.386              0.166
L15.Producer Price Index         0.089279           0.068160              1.310              0.190
L16.Money Supply                -0.212780           0.090007             -2.364              0.018
L16.Consumer Price Index        -0.335355           0.202305             -1.658              0.097
L16.Producer Price Index        -0.024015           0.068047             -0.353              0.724
L17.Money Supply                -0.027126           0.090128             -0.301              0.763
L17.Consumer Price Index        -0.227707           0.202249             -1.126              0.260
L17.Producer Price Index         0.155625           0.071369              2.181              0.029
L18.Money Supply                -0.074953           0.089864             -0.834              0.404
L18.Consumer Price Index        -0.252212           0.201517             -1.252              0.211
L18.Producer Price Index        -0.156565           0.071296             -2.196              0.028
L19.Money Supply                -0.102530           0.089769             -1.142              0.253
L19.Consumer Price Index        -0.211016           0.200997             -1.050              0.294
L19.Producer Price Index         0.155158           0.070967              2.186              0.029
L20.Money Supply                -0.162497           0.089412             -1.817              0.069
L20.Consumer Price Index        -0.128045           0.200421             -0.639              0.523
L20.Producer Price Index        -0.110549           0.072816             -1.518              0.129
L21.Money Supply                -0.167645           0.089673             -1.870              0.062
L21.Consumer Price Index        -0.234246           0.198814             -1.178              0.239
L21.Producer Price Index        -0.001226           0.073400             -0.017              0.987
L22.Money Supply                -0.152828           0.089891             -1.700              0.089
L22.Consumer Price Index        -0.155205           0.197861             -0.784              0.433
L22.Producer Price Index        -0.124276           0.073552             -1.690              0.091
L23.Money Supply                -0.047756           0.090002             -0.531              0.596
L23.Consumer Price Index        -0.116369           0.197526             -0.589              0.556
L23.Producer Price Index         0.073826           0.073829              1.000              0.317
L24.Money Supply                -0.066984           0.090538             -0.740              0.459
L24.Consumer Price Index        -0.015836           0.198560             -0.080              0.936
L24.Producer Price Index        -0.028974           0.074442             -0.389              0.697
L25.Money Supply                -0.000349           0.090954             -0.004              0.997
L25.Consumer Price Index         0.073799           0.198775              0.371              0.710
L25.Producer Price Index         0.036854           0.074371              0.496              0.620
L26.Money Supply                 0.021942           0.091805              0.239              0.811
L26.Consumer Price Index         0.025849           0.198909              0.130              0.897
L26.Producer Price Index         0.129106           0.073556              1.755              0.079
L27.Money Supply                -0.127516           0.091662             -1.391              0.164
L27.Consumer Price Index        -0.320908           0.199230             -1.611              0.107
L27.Producer Price Index        -0.062197           0.072878             -0.853              0.393
L28.Money Supply                -0.093769           0.090744             -1.033              0.301
L28.Consumer Price Index        -0.141143           0.199994             -0.706              0.480
L28.Producer Price Index        -0.000346           0.072810             -0.005              0.996
L29.Money Supply                 0.020646           0.090585              0.228              0.820
L29.Consumer Price Index         0.012925           0.200517              0.064              0.949
L29.Producer Price Index        -0.299044           0.072377             -4.132              0.000
L30.Money Supply                 0.008399           0.090049              0.093              0.926
L30.Consumer Price Index        -0.025210           0.200258             -0.126              0.900
L30.Producer Price Index         0.335361           0.073514              4.562              0.000
L31.Money Supply                -0.062662           0.089728             -0.698              0.485
L31.Consumer Price Index        -0.100924           0.198941             -0.507              0.612
L31.Producer Price Index        -0.086537           0.074990             -1.154              0.249
L32.Money Supply                -0.067643           0.089545             -0.755              0.450
L32.Consumer Price Index        -0.192041           0.194355             -0.988              0.323
```

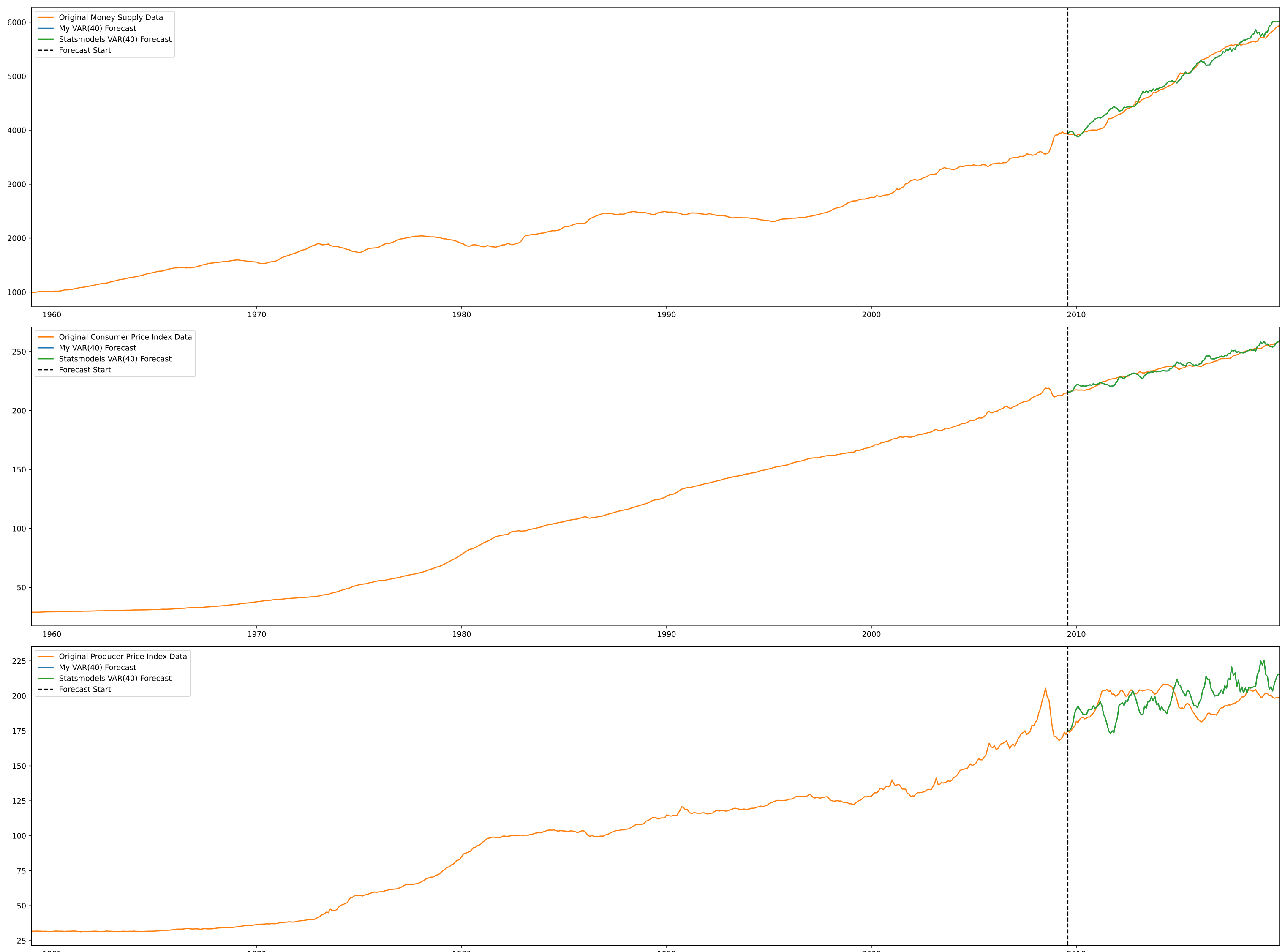
```
L32.Producer Price Index         0.191414         0.075722            2.528           0.011
L33.Money Supply                 0.005105         0.087846            0.058           0.954
L33.Consumer Price Index        -0.062031         0.188905           -0.328           0.743
L33.Producer Price Index        -0.055256         0.075827           -0.729           0.466
L34.Money Supply                -0.012547         0.085957           -0.146           0.884
L34.Consumer Price Index        -0.038587         0.179581           -0.215           0.830
L34.Producer Price Index        -0.119757         0.076507           -1.565           0.118
L35.Money Supply                 0.055359         0.083373            0.664           0.507
L35.Consumer Price Index        -0.111812         0.168560           -0.663           0.507
L35.Producer Price Index        -0.030947         0.077290           -0.400           0.689
L36.Money Supply                 0.123095         0.080802            1.523           0.128
L36.Consumer Price Index         0.223568         0.156357            1.430           0.153
L36.Producer Price Index        -0.063446         0.077063           -0.823           0.410
L37.Money Supply                 0.022820         0.076249            0.299           0.765
L37.Consumer Price Index        -0.036392         0.142243           -0.256           0.798
L37.Producer Price Index        -0.085292         0.076193           -1.119           0.263
L38.Money Supply                 0.098653         0.074160            1.330           0.183
L38.Consumer Price Index         0.097630         0.127495            0.766           0.444
L38.Producer Price Index         0.046028         0.075954            0.606           0.545
L39.Money Supply                -0.000667         0.068407           -0.010           0.992
L39.Consumer Price Index        -0.189816         0.104672           -1.813           0.070
L39.Producer Price Index         0.050515         0.075237            0.671           0.502
L40.Money Supply                -0.038580         0.059747           -0.646           0.518
L40.Consumer Price Index        -0.292609         0.080303           -3.644           0.000
L40.Producer Price Index        -0.063949         0.075640           -0.845           0.398
===========================================================================================

Results for equation Producer Price Index
===========================================================================================
                              coefficient       std. error           t-stat            prob
-------------------------------------------------------------------------------------------
const                            0.089373         0.046370            1.927           0.054
L1.Money Supply                 -0.077718         0.060357           -1.288           0.198
L1.Consumer Price Index          0.036503         0.081553            0.448           0.654
L1.Producer Price Index          0.379696         0.063979            5.935           0.000
L2.Money Supply                 -0.054715         0.071346           -0.767           0.443
L2.Consumer Price Index         -0.025939         0.104144           -0.249           0.803
L2.Producer Price Index          0.177414         0.066204            2.680           0.007
L3.Money Supply                 -0.114066         0.076865           -1.484           0.138
L3.Consumer Price Index         -0.094517         0.131663           -0.718           0.473
L3.Producer Price Index          0.047752         0.066866            0.714           0.475
L4.Money Supply                 -0.129745         0.080521           -1.611           0.107
L4.Consumer Price Index          0.220893         0.150184            1.471           0.141
L4.Producer Price Index         -0.104220         0.067021           -1.555           0.120
L5.Money Supply                 -0.023300         0.087230           -0.267           0.789
L5.Consumer Price Index          0.017732         0.171036            0.104           0.917
L5.Producer Price Index         -0.196040         0.067188           -2.918           0.004
L6.Money Supply                 -0.078806         0.089861           -0.877           0.381
L6.Consumer Price Index          0.197937         0.188679            1.049           0.294
L6.Producer Price Index         -0.042110         0.068257           -0.617           0.537
L7.Money Supply                 -0.088312         0.091681           -0.963           0.335
L7.Consumer Price Index          0.372217         0.200899            1.853           0.064
L7.Producer Price Index         -0.112762         0.068434           -1.648           0.099
L8.Money Supply                 -0.160643         0.093808           -1.712           0.087
L8.Consumer Price Index          0.340019         0.209416            1.624           0.104
```

```
L8.Producer Price Index        -0.035859         0.066828          -0.537            0.592
L9.Money Supply                -0.142356         0.095532          -1.490            0.136
L9.Consumer Price Index         0.417083         0.216613           1.925            0.054
L9.Producer Price Index        -0.163487         0.067229          -2.432            0.015
L10.Money Supply               -0.098191         0.095406          -1.029            0.303
L10.Consumer Price Index        0.450791         0.219285           2.056            0.040
L10.Producer Price Index       -0.034941         0.068161          -0.513            0.608
L11.Money Supply                0.004954         0.095335           0.052            0.959
L11.Consumer Price Index        0.531152         0.222818           2.384            0.017
L11.Producer Price Index        0.134050         0.068399           1.960            0.050
L12.Money Supply               -0.175206         0.096063          -1.824            0.068
L12.Consumer Price Index        0.237794         0.222703           1.068            0.286
L12.Producer Price Index        0.048366         0.069693           0.694            0.488
L13.Money Supply               -0.066822         0.097045          -0.689            0.491
L13.Consumer Price Index        0.076657         0.222210           0.345            0.730
L13.Producer Price Index        0.056211         0.073015           0.770            0.441
L14.Money Supply               -0.198122         0.097741          -2.027            0.043
L14.Consumer Price Index       -0.003416         0.222621          -0.015            0.988
L14.Producer Price Index        0.077304         0.073003           1.059            0.290
L15.Money Supply               -0.177578         0.098861          -1.796            0.072
L15.Consumer Price Index       -0.150483         0.222539          -0.676            0.499
L15.Producer Price Index       -0.007723         0.074799          -0.103            0.918
L16.Money Supply               -0.261379         0.098774          -2.646            0.008
L16.Consumer Price Index       -0.187126         0.222011          -0.843            0.399
L16.Producer Price Index        0.061409         0.074676           0.822            0.411
L17.Money Supply               -0.059567         0.098907          -0.602            0.547
L17.Consumer Price Index       -0.188527         0.221950          -0.849            0.396
L17.Producer Price Index        0.195937         0.078321           2.502            0.012
L18.Money Supply               -0.129702         0.098618          -1.315            0.188
L18.Consumer Price Index       -0.234328         0.221147          -1.060            0.289
L18.Producer Price Index       -0.145972         0.078241          -1.866            0.062
L19.Money Supply               -0.149162         0.098513          -1.514            0.130
L19.Consumer Price Index       -0.258209         0.220576          -1.171            0.242
L19.Producer Price Index        0.216149         0.077880           2.775            0.006
L20.Money Supply               -0.142495         0.098122          -1.452            0.146
L20.Consumer Price Index       -0.242761         0.219944          -1.104            0.270
L20.Producer Price Index       -0.088974         0.079909          -1.113            0.266
L21.Money Supply               -0.155610         0.098408          -1.581            0.114
L21.Consumer Price Index       -0.140896         0.218181          -0.646            0.518
L21.Producer Price Index       -0.003283         0.080550          -0.041            0.967
L22.Money Supply               -0.150242         0.098647          -1.523            0.128
L22.Consumer Price Index       -0.068887         0.217135          -0.317            0.751
L22.Producer Price Index       -0.180931         0.080717          -2.242            0.025
L23.Money Supply                0.061775         0.098769           0.625            0.532
L23.Consumer Price Index        0.007562         0.216766           0.035            0.972
L23.Producer Price Index       -0.053864         0.081021          -0.665            0.506
L24.Money Supply               -0.053022         0.099357          -0.534            0.594
L24.Consumer Price Index        0.088369         0.217901           0.406            0.685
L24.Producer Price Index        0.033090         0.081693           0.405            0.685
L25.Money Supply               -0.075384         0.099814          -0.755            0.450
L25.Consumer Price Index        0.139449         0.218137           0.639            0.523
L25.Producer Price Index        0.043703         0.081615           0.535            0.592
L26.Money Supply               -0.049964         0.100748          -0.496            0.620
L26.Consumer Price Index        0.172616         0.218285           0.791            0.429
L26.Producer Price Index        0.121523         0.080721           1.505            0.132
```

```
L27.Money Supply               -0.119553         0.100591           -1.189            0.235
L27.Consumer Price Index       -0.111926         0.218637           -0.512            0.609
L27.Producer Price Index       -0.139561         0.079977           -1.745            0.081
L28.Money Supply               -0.145118         0.099583           -1.457            0.145
L28.Consumer Price Index       -0.033212         0.219475           -0.151            0.880
L28.Producer Price Index        0.024286         0.079902            0.304            0.761
L29.Money Supply               -0.002314         0.099409           -0.023            0.981
L29.Consumer Price Index       -0.008532         0.220049           -0.039            0.969
L29.Producer Price Index       -0.137259         0.079427           -1.728            0.084
L30.Money Supply                0.060109         0.098821            0.608            0.543
L30.Consumer Price Index        0.158261         0.219765            0.720            0.471
L30.Producer Price Index        0.259675         0.080675            3.219            0.001
L31.Money Supply               -0.009069         0.098468           -0.092            0.927
L31.Consumer Price Index        0.011110         0.218320            0.051            0.959
L31.Producer Price Index        0.110819         0.082295            1.347            0.178
L32.Money Supply               -0.063592         0.098267           -0.647            0.518
L32.Consumer Price Index        0.035262         0.213286            0.165            0.869
L32.Producer Price Index        0.089616         0.083098            1.078            0.281
L33.Money Supply                0.071008         0.096403            0.737            0.461
L33.Consumer Price Index        0.079157         0.207306            0.382            0.703
L33.Producer Price Index        0.192598         0.083213            2.315            0.021
L34.Money Supply                0.069639         0.094330            0.738            0.460
L34.Consumer Price Index        0.249473         0.197074            1.266            0.206
L34.Producer Price Index       -0.233303         0.083959           -2.779            0.005
L35.Money Supply                0.179996         0.091494            1.967            0.049
L35.Consumer Price Index        0.155485         0.184979            0.841            0.401
L35.Producer Price Index       -0.001379         0.084819           -0.016            0.987
L36.Money Supply                0.138648         0.088672            1.564            0.118
L36.Consumer Price Index        0.392280         0.171588            2.286            0.022
L36.Producer Price Index       -0.091792         0.084570           -1.085            0.278
L37.Money Supply               -0.043660         0.083677           -0.522            0.602
L37.Consumer Price Index       -0.098503         0.156098           -0.631            0.528
L37.Producer Price Index        0.080509         0.083614            0.963            0.336
L38.Money Supply                0.083383         0.081384            1.025            0.306
L38.Consumer Price Index        0.126721         0.139914            0.906            0.365
L38.Producer Price Index       -0.036822         0.083353           -0.442            0.659
L39.Money Supply               -0.090282         0.075070           -1.203            0.229
L39.Consumer Price Index       -0.151025         0.114868           -1.315            0.189
L39.Producer Price Index        0.136869         0.082566            1.658            0.097
L40.Money Supply                0.005152         0.065567            0.079            0.937
L40.Consumer Price Index       -0.119509         0.088125           -1.356            0.175
L40.Producer Price Index       -0.042455         0.083008           -0.511            0.609
===========================================================================================

Correlation matrix of residuals
                        Money Supply  Consumer Price Index  Producer Price Index
Money Supply                1.000000             -0.636107             -0.408936
Consumer Price Index       -0.636107              1.000000              0.668839
Producer Price Index       -0.408936              0.668839              1.000000
```

There are a lot of things to look at here but checking a few values for example the constant on each series we see that the estimates from statsmodels are the same as mine. Now I will show the predictions from mine and the predictions from statsmodels on the same graph.

```python
fcasts_standard = sm_VAR.forecast(train[-p:].values, n_test).T
fcasts_sm_VAR = {}
MSEs_sm_VAR = {}

fig, axs = plt.subplots(K, 1, figsize=(24, 18), sharex = True)
for j in range(len(names)):
  name = names[j]
  fcasts_sm_VAR[name] = fcasts_standard[j,:]*sds[name]
  fcasts_sm_VAR[name] = undifference(name, fcasts_sm_VAR, n_test)

  fcast_sm_VAR = fcasts_sm_VAR[name]
  fcast_VAR = fcasts_VAR[name]

  MSEs_sm_VAR[name] = np.mean((fcast_sm_VAR - test_df[name].values)**2)

  ax = axs[j]
  ax.plot(data.index, data[name].values, label = f"Original {name} Data", color = "C1")
  ax.plot(data.index[-n_test:], fcast_VAR, label = f"My VAR({p}) Forecast", color = "C0")
  ax.plot(data.index[-n_test:], fcast_sm_VAR, label = f"Statsmodels VAR({p}) Forecast", color = "C2")
  ax.axvline(x = data.index[-n_test], color = "black", linestyle = "--", label = "Forecast Start")
  ax.set_xlim(data.index[0], data.index[-1])
  ax.tick_params(labelbottom = True)
  ax.legend(loc = "upper left")
plt.tight_layout()
plt.show()
```



Clearly they are perfectly overlapping. There is no difference between the predictions which shows I have successfully implemented VAR from scratch. It is not necessary since I have just shown that the forecasts are the same but I have also provided the mean squared prediction errors below as well.

```
print(f"My VAR MSEs:")
for name in names:
    print(f"MSE for {name} {MSEs_VAR[name]:.2f}")
print(f"\nMy VAR Overall MSE: {np.mean(list(MSEs_VAR.values())):.2f}")

print(f"\n\nStatsmodels VAR MSEs:")
for name in names:
    print(f"MSE for {name} {MSEs_sm_VAR[name]:.2f}")
print(f"\nStatsmodels VAR Overall MSE: {np.mean(list(MSEs_sm_VAR.values())):.2f}")
```

```
My VAR MSEs:
MSE for Money Supply 10369.81
MSE for Consumer Price Index 8.28
MSE for Producer Price Index 183.16

My VAR Overall MSE: 3520.42
```

```
Statsmodels VAR MSEs:
MSE for Money Supply 10369.81
MSE for Consumer Price Index 8.28
MSE for Producer Price Index 183.16

Statsmodels VAR Overall MSE: 3520.42
```

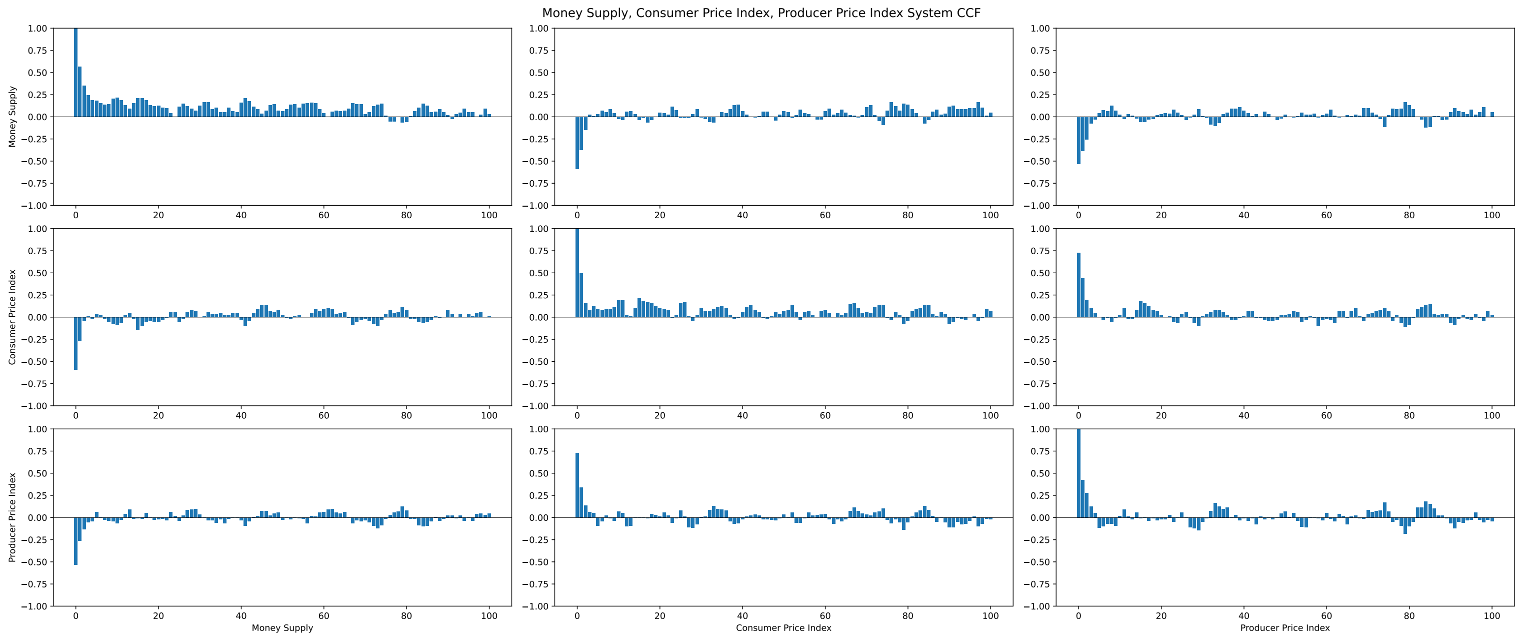
Again, they are exactly the same. I have properly implemented VAR.

Why VAR is Performing Better

The reason VAR performs so much better is precisely because of the interrelation of these time series. Looking at the plots of the time series you can clearly see that a change in one time series corresponds with a change in the others. When using independent component modeling with ARMA this important interrelation is not captured since we are only ever looking at one time series. However, VAR estimates all time series jointly and is able to capture these interrelations and improve its performance. An important tool for examining the structure of the system is the cross correlation function (CCF).

The CCF matrix $R(h)$ has its entry at i, j defined to be $\rho_{ij}(h) = \mathrm{Corr}(y_{i,t}, y_{j,t-h})$ similar to the ACF for univariate series. In fact if $i = j$ then the CCF is just the ACF. Below I implement the sample CCF from scratch and plot the CCF for the differenced data in the macroeconomic system I have been using.

```python
def ccf(x, y, lags):
    # Standardize
    x = (x - np.mean(x))/np.std(x)
    y = (y - np.mean(y))/np.std(y)

    ccf_vals = []
    lags = range(0, lags + 1)
    for lag in lags:
        # Compute cross correlation
        if lag == 0:
            corr = np.corrcoef(x, y)[0, 1]
        else:
            corr = np.corrcoef(x[lag:], y[:-lag])[0, 1]
        ccf_vals.append(corr)
    return(np.array(ccf_vals), lags)


fig, axes = plt.subplots(K, K, figsize = (24, 10), constrained_layout = True)
for i, name1 in enumerate(names):
    for j, name2 in enumerate(names):
        ax = axes[i, j]
        corr, h = ccf(data.diff(1).dropna()[name1].values, data.diff(1).dropna()[name2].values, lags =
        ax.bar(h, corr)
        ax.axhline(0, color = "black", linewidth = 0.5)
        ax.set_ylim(-1, 1)
        if i == K - 1:
            ax.set_xlabel(name2, fontsize = 10)
        else:
            ax.set_xlabel("")
        if j == 0:
            ax.set_ylabel(name1, fontsize = 10)
        else:
            ax.set_ylabel("")
fig.suptitle("Money Supply, Consumer Price Index, Producer Price Index System CCF", fontsize = 14)
plt.show()
```

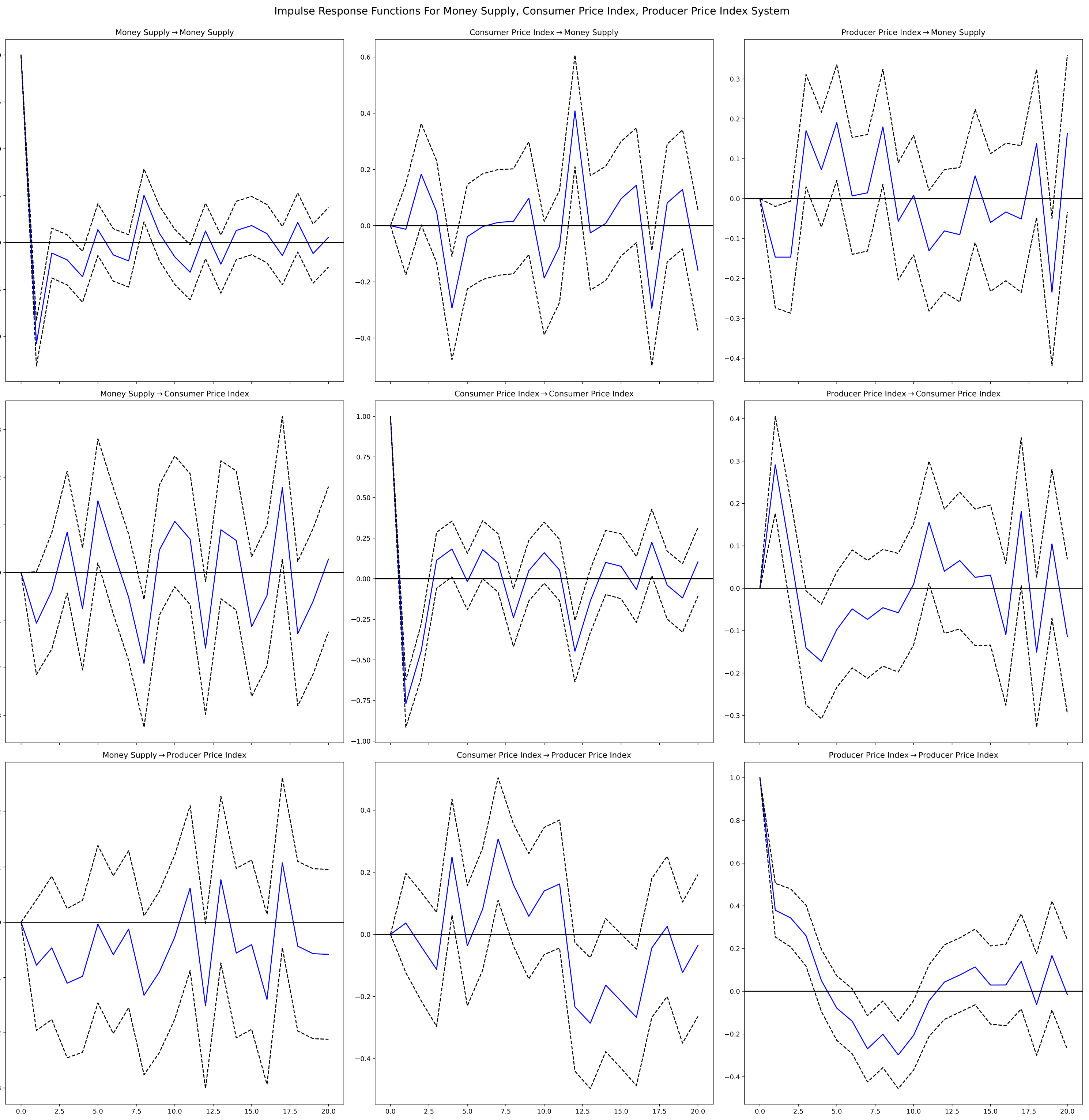


The plots on the diagonals are indeed the ACF for each time series but importantly notice that the cross correlation for each of the series is rather significant for the low lags. It is important to note again that these are the CCFs for the once differenced data so these plots show how changes in one variable are correlated with changes in the others. This is different from the standard interpretation which is usually how the level of one variable is correlated with the level of the others.

Still, this confirms the fact that these time series are highly interrelated and explains why VAR performs much better compared to independent component modeling with ARMA. By examining the plots we see that changes in money supply are negatively correlated with changes in consumer price index in the short run but are somewhat positively correlated with changes in consumer price index in the long run. Also it seems that changes in money supply are negatively correlated with changes in producer price index in the short run but slightly positively correlated with changes in producer price index in the long run. Finally, it seems like changes in producer price index are positively correlated with changes in consumer price index in the short run and slightly positively correlated with changes in consumer price index in the long run.

Another useful tool for examining VAR models are impulse response functions (IRFs) which show how a shock from one variable affects the others over time. I will not implement IRF from scratch but will show the results from the statsmodels package.

```
IRF = sm_VAR.irf(20)
fig = IRF.plot(orth = False)
fig.set_size_inches(24, 24)
plt.tight_layout()
fig.suptitle("Impulse Response Functions For Money Supply, Consumer Price Index, Producer Price Index Sy
plt.show()
```



When looking at IRFs we are primarily interested in the plots not on the main diagonal since those show the interaction of the time series. Note that the VAR model was fit to twice differenced money supply data, twice differenced consumer price index data, and once differenced producer price index data. These plots do not have much meaningful interpretation in terms of the undifferenced variables so I will not go very into depth about them. The reason they have no meaningful interpretation for each of the variables is that a sharp spike in one of the double differenced data represents a sharp spike in the acceleration of the undifferenced variable and a sharp spike in single differenced data represents a sharp spike in the rate of change of the undifferenced variable. It is not really possible to invert the differencing effects to obtain meaningful effects without simply fitting a VAR model to the undifferenced data which would be ill-advised since the undifferenced data is non-stationary. The main takeaways in this post analysis are the results from the CCF, the IRF is not very relevant.

The result for this specific dataset generalizes to other datasets where the component time series are highly interrelated, VAR is able to outperform independent component ARMA modeling since it can capture those interrelations and improve forecasts. In datasets with very low or no interrelation I do not forsee VAR performing better, in fact it might perform worse since it needs to estimate more parameters than each ARMA model does on its own which could lead to overfitting.

References

[1] Lutkepohl, H. (2005). New introduction to multiple time series analysis. Springer Science & Business Media.