2024-04-11

##2 Discuss the series you picked, describe the series descriptions, and what part of the economy is describes.

```
The sectors of Finance and Insurance, Real Estate, Professional Services, and Information repres
ent pivotal elements of the economic landscape, each offering distinct perspectives on sectoral
health and trends.

Finance and Insurance: This crucial economic pillar supports transactions and risk management th
rough banking, investments, and insurance services. It mirrors the financial system's robustness
and its capacity to back economic endeavors.

Real Estate: This sector, encompassing the trade and leasing of properties, acts as an economic
barometer, reflecting broader economic trends through property market dynamics. Fluctuations in
real estate prices and leasing rates often precede shifts in the economic cycle.

Professional Services: Encompassing specialized services like legal, accounting, architectural,
and consulting, this sector bolsters other businesses' operational efficiency and innovation. It
s growth signals a buoyant economy where firms are keen to enhance competitiveness.

Information: Covering the dissemination and processing of information and cultural products, thi
s sector is at the forefront of the knowledge economy, underscored by digitalization and technol
ogical advances. It is instrumental in fostering innovation and economic expansion.

Summary: These sectors are integral to decoding the U.S. economy's fabric, each serving critical
roles—from underpinning economic transactions and managing risks in Finance and Insurance, actin
g as an economic health gauge in Real Estate, indicating business vibrancy and forward investmen
t in Professional Services, to driving the shift towards a digital, knowledge-intensive economy
in the Information sector. Together, they offer a comprehensive view of economic vitality and pr
ospective directions, essential for informed decision-making and policy formulation.
```

Time Series Homework 4

```r
# Load necessary libraries
library(readr)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(vars)
```

```
## Loading required package: MASS
```

```
## Loading required package: strucchange
```

```
## Loading required package: sandwich
```

```
## Loading required package: urca
```

```
## Loading required package: lmtest
```

```r
library(BigVAR)
```

```
## Loading required package: lattice
```

```r
library(tseries)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
# Read the data
data <- read_csv("HW4_data.csv")
```

```
## Rows: 246 Columns: 5
```

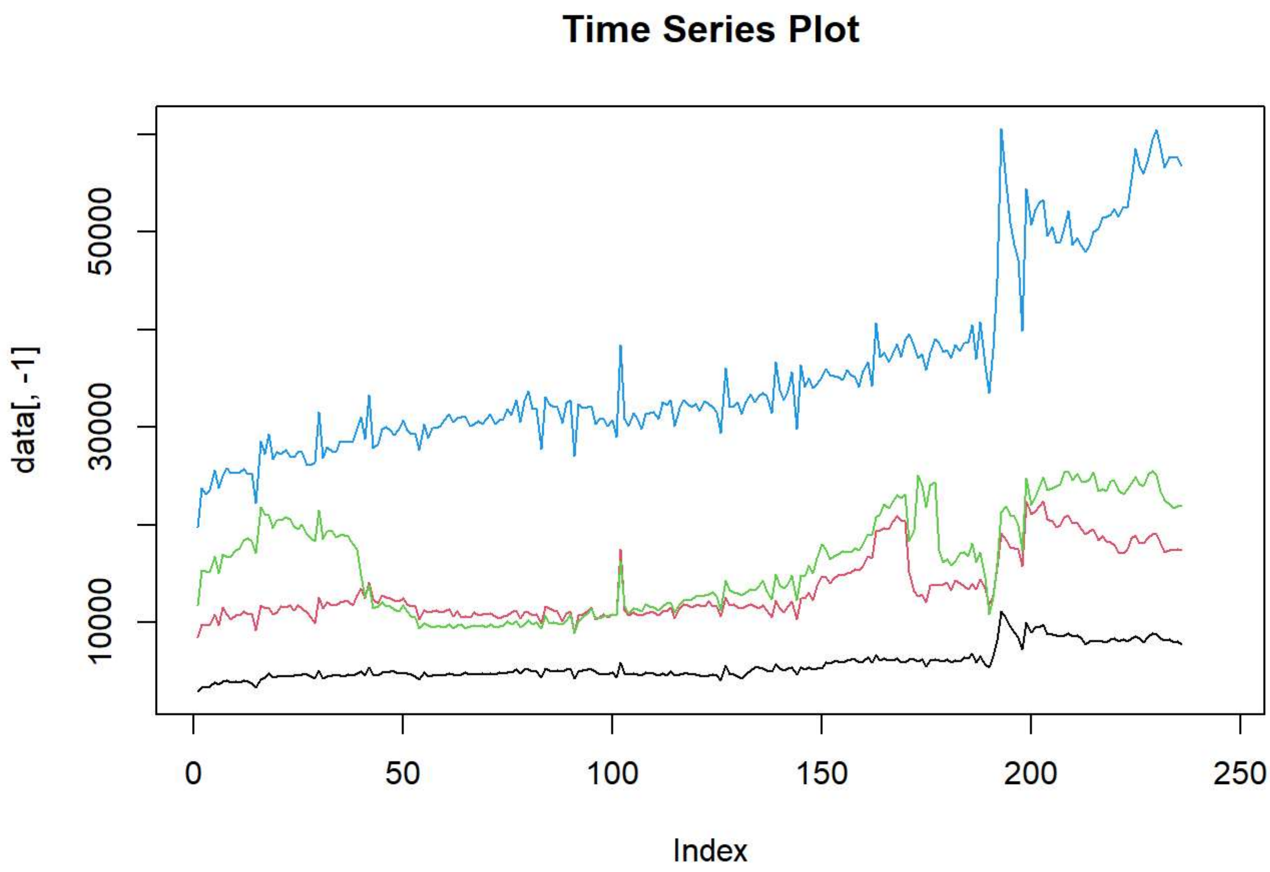
```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): Period
## num (4): IUST, FAIUST, REUST, PSUST
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

3. Empirical Analysis

```
# Convert 'Period' to a year-month format and set it as index
data$Period <- as.yearmon(data$Period, "%b-%Y")

# Convert numeric columns to numeric type (they might be read as factors or characters)
data$IUST <- as.numeric(gsub(",", "", data$IUST))
data$FAIUST <- as.numeric(gsub(",", "", data$FAIUST))
data$REUST <- as.numeric(gsub(",", "", data$REUST))
data$PSUST <- as.numeric(gsub(",", "", data$PSUST))

# Perform empirical analysis
summary(data)
```

```
##      Period          IUST           FAIUST          REUST           PSUST
##  Min.   :2004   Min.   : 2944   Min.   : 8534   Min.   : 8923   Min.   :19729
##  1st Qu.:2009   1st Qu.: 4669   1st Qu.:11074   1st Qu.:11435   1st Qu.:30062
##  Median :2014   Median : 5045   Median :11835   Median :16146   Median :32403
##  Mean   :2014   Mean   : 5709   Mean   :13487   Mean   :16201   Mean   :35711
##  3rd Qu.:2019   3rd Qu.: 6197   3rd Qu.:15236   3rd Qu.:20760   3rd Qu.:37909
##  Max.   :2024   Max.   :11144   Max.   :22445   Max.   :25547   Max.   :60523
##  NA's   :10     NA's   :10      NA's   :10      NA's   :10      NA's   :10
```

```
plot.zoo(data[, -1], plot.type = "single", col = 1:4, main = "Time Series Plot")
```




```
# Stationarity check function for each series in the dataframe
check_stationarity <- function(data, column_name) {
  series <- na.omit(data[[column_name]])  # Pre-process to omit NA values

  # Perform ADF test
  adf_test <- adf.test(series)

  # Print the ADF test results
  cat("ADF Test Results for", column_name, ":\n")
  cat("Test Statistic:", adf_test$statistic, "\n")
  cat("P-value:", adf_test$p.value, "\n")
  cat("Critical Values:\n")
  for (cv in names(adf_test$critical)) {
    cat(cv, ":", adf_test$critical[cv], "\n")
  }

  # Interpretation based on p-value
  cat("Result: ")
  if (adf_test$p.value < 0.05) {
    cat("Series is stationary because p value is less than 0.05\n\n")
  } else {
    cat("Series is non-stationary\n\n")
  }
}
```

```
# Assuming 'data' is loaded and prepared as per the initial script
columns_to_check <- c("IUST", "FAIUST", "REUST", "PSUST")

# Apply the stationarity check to each column
for (col in columns_to_check) {
  check_stationarity(data, col)
}
```

```
## ADF Test Results for IUST :
## Test Statistic: -1.983107
## P-value: 0.5829817
## Critical Values:
## Result: Series is non-stationary
##
## ADF Test Results for FAIUST :
## Test Statistic: -2.746734
## P-value: 0.2619614
## Critical Values:
## Result: Series is non-stationary
##
## ADF Test Results for REUST :
## Test Statistic: -1.66694
## P-value: 0.715895
## Critical Values:
## Result: Series is non-stationary
##
## ADF Test Results for PSUST :
## Test Statistic: -0.8035347
## P-value: 0.9602201
## Critical Values:
## Result: Series is non-stationary
```

4(a). Fit a VAR(1) model

```
# Check for NAs in each column
colSums(is.na(data))
```

```
## Period   IUST FAIUST  REUST  PSUST
##     10     10     10     10     10
```

```
data <- na.omit(data)

# Impute missing values
for(i in 2:ncol(data)) {
  data[is.na(data[,i]), i] <- mean(data[,i], na.rm = TRUE)
}
```

```
## Warning in mean.default(data[, i], na.rm = TRUE): argument is not numeric or
## logical: returning NA
## Warning in mean.default(data[, i], na.rm = TRUE): argument is not numeric or
## logical: returning NA
## Warning in mean.default(data[, i], na.rm = TRUE): argument is not numeric or
## logical: returning NA
## Warning in mean.default(data[, i], na.rm = TRUE): argument is not numeric or
## logical: returning NA
```

```
# Time series conversion and VAR model fitting
data.ts <- ts(data[,-1], start=c(2004, 7), frequency=12) # Adjust start period as necessary
fit_var1 <- VAR(data.ts, p=1, type="both")
summary(fit_var1)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: IUST, FAIUST, REUST, PSUST
## Deterministic variables: both
## Sample size: 235
## Log Likelihood: -7479.327
## Roots of the characteristic polynomial:
## 0.9644 0.9644 0.8559 0.794
## Call:
## VAR(y = data.ts, p = 1, type = "both")
##
##
## Estimation results for equation IUST:
## =====================================
## IUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##             Estimate Std. Error t value Pr(>|t|)
## IUST.l1     0.977284   0.079843  12.240  < 2e-16 ***
## FAIUST.l1  -0.021680   0.024002  -0.903   0.3673
## REUST.l1    0.006702   0.009744   0.688   0.4923
## PSUST.l1   -0.025333   0.013799  -1.836   0.0677 .
## const     773.913280 175.568460   4.408 1.60e-05 ***
## trend       3.881926   0.944661   4.109 5.53e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 461.3 on 229 degrees of freedom
## Multiple R-Squared: 0.9171,  Adjusted R-squared: 0.9153
## F-statistic: 506.6 on 5 and 229 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation FAIUST:
## =======================================
## FAIUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##             Estimate Std. Error t value Pr(>|t|)
## IUST.l1      0.51658    0.19014   2.717 0.007093 **
## FAIUST.l1    0.79655    0.05716  13.936  < 2e-16 ***
## REUST.l1     0.03560    0.02321   1.534 0.126389
## PSUST.l1    -0.11150    0.03286  -3.393 0.000814 ***
## const     2150.43702  418.09648   5.143 5.79e-07 ***
## trend        9.06812    2.24960   4.031 7.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1098 on 229 degrees of freedom
## Multiple R-Squared: 0.8956,  Adjusted R-squared: 0.8934
## F-statistic:   393 on 5 and 229 DF,  p-value: < 2.2e-16
##
```

```
##
## Estimation results for equation REUST:
## ======================================
## REUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##              Estimate Std. Error t value Pr(>|t|)
## IUST.l1       0.49532    0.24583   2.015 0.045085 *
## FAIUST.l1    -0.12745    0.07390  -1.725 0.085939 .
## REUST.l1      0.99424    0.03000  33.139  < 2e-16 ***
## PSUST.l1     -0.11643    0.04249  -2.740 0.006621 **
## const      2095.01108  540.56040   3.876 0.000139 ***
## trend         9.09348    2.90853   3.126 0.001998 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1420 on 229 degrees of freedom
## Multiple R-Squared: 0.9257,  Adjusted R-squared: 0.924
## F-statistic: 570.3 on 5 and 229 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation PSUST:
## ======================================
## PSUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##              Estimate Std. Error t value Pr(>|t|)
## IUST.l1       0.23694    0.43577   0.544    0.587
## FAIUST.l1    -0.16501    0.13100  -1.260    0.209
## REUST.l1      0.05420    0.05318   1.019    0.309
## PSUST.l1      0.81034    0.07531  10.760  < 2e-16 ***
## const      4271.46874  958.22134   4.458 1.30e-05 ***
## trend        22.16281    5.15579   4.299 2.54e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 2518 on 229 degrees of freedom
## Multiple R-Squared: 0.9254,  Adjusted R-squared: 0.9238
## F-statistic: 568.5 on 5 and 229 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##           IUST  FAIUST   REUST   PSUST
## IUST    212766  381036  462019 1074195
## FAIUST  381036 1206597 1206085 2258371
## REUST   462019 1206085 2016961 2632585
## PSUST  1074195 2258371 2632585 6337832
##
## Correlation matrix of residuals:
##          IUST FAIUST  REUST  PSUST
## IUST   1.0000 0.7520 0.7053 0.9250
```

```
## FAIUST 0.7520 1.0000 0.7731 0.8167
## REUST  0.7053 0.7731 1.0000 0.7363
## PSUST  0.9250 0.8167 0.7363 1.0000
```

4(b). VAR(p) model with p > 1.

```
fit_varp <- VAR(data.ts, p=3, type="both")
summary(fit_varp)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: IUST, FAIUST, REUST, PSUST
## Deterministic variables: both
## Sample size: 233
## Log Likelihood: -7352.23
## Roots of the characteristic polynomial:
## 0.9868 0.9868 0.8306 0.8306 0.5372 0.4863 0.4863 0.4826 0.4826 0.4579 0.4579 0.3364
## Call:
## VAR(y = data.ts, p = 3, type = "both")
##
##
## Estimation results for equation IUST:
## =====================================
## IUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l2
+ IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
##
##             Estimate Std. Error t value Pr(>|t|)
## IUST.l1     0.935303   0.178019   5.254 3.52e-07 ***
## FAIUST.l1  -0.033031   0.054045  -0.611  0.54171
## REUST.l1    0.009579   0.036424   0.263  0.79281
## PSUST.l1   -0.048009   0.037525  -1.279  0.20211
## IUST.l2     0.249435   0.223238   1.117  0.26507
## FAIUST.l2   0.052582   0.073652   0.714  0.47603
## REUST.l2   -0.084234   0.052140  -1.616  0.10764
## PSUST.l2    0.017135   0.045856   0.374  0.70901
## IUST.l3    -0.216274   0.186304  -1.161  0.24696
## FAIUST.l3  -0.031524   0.052869  -0.596  0.55162
## REUST.l3    0.080939   0.035134   2.304  0.02218 *
## PSUST.l3    0.012086   0.039510   0.306  0.75998
## const     571.589359 193.682165   2.951  0.00351 **
## trend       3.034787   0.991707   3.060  0.00249 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 451.8 on 219 degrees of freedom
## Multiple R-Squared: 0.9224,  Adjusted R-squared: 0.9178
## F-statistic: 200.3 on 13 and 219 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation FAIUST:
## =======================================
## FAIUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l
2 + IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
##
##             Estimate Std. Error t value Pr(>|t|)
## IUST.l1      0.38847    0.41755   0.930   0.3532
## FAIUST.l1    0.82883    0.12676   6.538 4.32e-10 ***
## REUST.l1     0.03395    0.08543   0.397   0.6915
## PSUST.l1    -0.20633    0.08802  -2.344   0.0200 *
```

```
## IUST.l2      0.79101    0.52361   1.511   0.1323
## FAIUST.l2    0.14021    0.17275   0.812   0.4179
## REUST.l2    -0.16911    0.12230  -1.383   0.1681
## PSUST.l2    -0.01708    0.10756  -0.159   0.8739
## IUST.l3     -0.68824    0.43698  -1.575   0.1167
## FAIUST.l3   -0.15752    0.12401  -1.270   0.2053
## REUST.l3     0.16842    0.08241   2.044   0.0422 *
## PSUST.l3     0.14214    0.09267   1.534   0.1265
## const     1443.29663  454.28909   3.177   0.0017 **
## trend        6.00037    2.32609   2.580   0.0105 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1060 on 219 degrees of freedom
## Multiple R-Squared: 0.9061,  Adjusted R-squared: 0.9005
## F-statistic: 162.5 on 13 and 219 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation REUST:
## ======================================
## REUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l2
+ IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
##
##              Estimate Std. Error t value Pr(>|t|)
## IUST.l1      0.78449    0.52054   1.507   0.1332
## FAIUST.l1   -0.31369    0.15803  -1.985   0.0484 *
## REUST.l1     1.05270    0.10651   9.884   <2e-16 ***
## PSUST.l1    -0.23555    0.10972  -2.147   0.0329 *
## IUST.l2     -0.04034    0.65276  -0.062   0.9508
## FAIUST.l2   -0.01575    0.21536  -0.073   0.9418
## REUST.l2    -0.33015    0.15246  -2.165   0.0314 *
## PSUST.l2     0.26649    0.13408   1.987   0.0481 *
## IUST.l3     -0.31095    0.54476  -0.571   0.5687
## FAIUST.l3    0.30352    0.15459   1.963   0.0509 .
## REUST.l3     0.25374    0.10273   2.470   0.0143 *
## PSUST.l3    -0.12964    0.11553  -1.122   0.2630
## const     1157.42638  566.33752   2.044   0.0422 *
## trend        5.57785    2.89981   1.924   0.0557 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1321 on 219 degrees of freedom
## Multiple R-Squared: 0.9385,  Adjusted R-squared: 0.9348
## F-statistic: 256.9 on 13 and 219 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation PSUST:
## ======================================
## PSUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l2
+ IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
```

```
##
##             Estimate Std. Error t value Pr(>|t|)
## IUST.l1      1.00390    0.95057   1.056  0.29208
## FAIUST.l1   -0.01687    0.28858  -0.058  0.95344
## REUST.l1     0.00384    0.19449   0.020  0.98427
## PSUST.l1     0.38682    0.20037   1.931  0.05483 .
## IUST.l2      0.68127    1.19203   0.572  0.56823
## FAIUST.l2    0.04781    0.39328   0.122  0.90335
## REUST.l2    -0.23856    0.27841  -0.857  0.39245
## PSUST.l2     0.23308    0.24486   0.952  0.34219
## IUST.l3     -1.76823    0.99481  -1.777  0.07688 .
## FAIUST.l3   -0.08437    0.28231  -0.299  0.76532
## REUST.l3     0.27106    0.18761   1.445  0.14993
## PSUST.l3     0.29489    0.21097   1.398  0.16359
## const     2191.26417 1034.20572   2.119  0.03523 *
## trend       13.99335    5.29542   2.643  0.00882 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 2413 on 219 degrees of freedom
## Multiple R-Squared: 0.9335,  Adjusted R-squared: 0.9295
## F-statistic: 236.3 on 13 and 219 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##           IUST  FAIUST   REUST   PSUST
## IUST    204162  352141  419066 1013600
## FAIUST  352141 1123207 1110324 2046257
## REUST   419066 1110324 1745605 2331650
## PSUST  1013600 2046257 2331650 5821156
##
## Correlation matrix of residuals:
##          IUST FAIUST  REUST  PSUST
## IUST   1.0000 0.7354 0.7020 0.9298
## FAIUST 0.7354 1.0000 0.7930 0.8002
## REUST  0.7020 0.7930 1.0000 0.7315
```

5. Compare the two fits and decide with is better.

```
To evaluate and contrast the performances of the VAR(1) model and a VAR(p) model with p>1, we ca
n consider various metrics and features of these models:

Log Likelihood: A higher log likelihood indicates a model that more closely fits the observed da
ta, although this doesn't take into account the complexity of the model.

AIC and BIC: The Akaike Information Criterion and Bayesian Information Criterion adjust for mode
l complexity by penalizing the inclusion of additional parameters. Lower scores on these criteri
a suggest a more efficient balance between model fit and complexity. The AIC leans towards model
fit, whereas the BIC prefers simplicity, especially as the data size grows.

F-statistic and p-value: An elevated F-statistic alongside a p-value below 0.05 typically points
to a model's statistical significance, implying that the predictors explain a significant portio
n of the variability in the response variable.

Adjusted R-squared: This metric adjusts for the number of predictors in the model, with higher v
alues indicating a better fit to the data.

Residual Analysis: Evaluating the residuals for randomness can provide insight into the appropri
ateness of the model. This involves checking for autocorrelation, heteroskedasticity, and whethe
r the residuals follow a normal distribution.

### VAR(1) Summary:

- Log Likelihood: -7479.327
- AIC and BIC are not directly provided in the summary, but lower AIC and BIC are generally pref
erred.
- The model is statistically significant based on F-statistics and p-values.
- Adjusted R-squared values are quite high for all equations, suggesting a good fit.

### VAR(p) Summary with \(p>1\):

- Log Likelihood: -7489.036
- AIC(n) suggests a lag of 3 might be optimal, but for simplicity, you've likely chosen p=1 for
the comparison or p=3 as suggested by AIC. It's essential to ensure consistency in your comparis
on.
- Like the VAR(1), the VAR(p) model shows significant F-statistics and p-values.
- Adjusted R-squared values are also high, indicating a good fit.

### Decision:

Without the direct AIC and BIC values for each model in the summaries provided, it's hard to mak
e a definitive comparison based solely on these results. However, you typically would prefer:

- The model with the **lower AIC and BIC** for a balance of goodness of fit and simplicity.
- The model with **higher log likelihood**, provided it doesn't overfit (which AIC and BIC help
address).
- The model that provides **better diagnostics on residuals**, indicating that the assumptions o
f the VAR model are better met.
```

```
If the AIC and BIC are lower for the VAR(p) model (with (p>1) as suggested by the AIC(n) resul
t), and the residuals do not violate model assumptions, that model might be preferable for forec
asting and interpretation purposes. It's also crucial to ensure that the increased complexity of
a VAR(p) model with (p>1) is justified by a significantly better fit, as indicated by these crit
eria.

For a more detailed analysis, we should directly compare the AIC and BIC values, perform residua
l diagnostics, and consider the practical significance of the variables and lags included in the
model.
```

6. Produce a one month a head forecast of the series.

```
print("Forecast values from VAR(1):")
```

```
## [1] "Forecast values from VAR(1):"
```

```
print(fit_var1)
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation IUST:
## =========================================
## Call:
## IUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##       IUST.l1     FAIUST.l1      REUST.l1      PSUST.l1         const
##   0.977284041  -0.021679748   0.006701793  -0.025333004 773.913280467
##         trend
##   3.881925753
##
##
## Estimated coefficients for equation FAIUST:
## ===========================================
## Call:
## FAIUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##       IUST.l1     FAIUST.l1      REUST.l1      PSUST.l1         const
##    0.51658112    0.79654581    0.03559884   -0.11149638 2150.43701675
##         trend
##    9.06811613
##
##
## Estimated coefficients for equation REUST:
## ==========================================
## Call:
## REUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##      IUST.l1    FAIUST.l1     REUST.l1     PSUST.l1        const        trend
##    0.4953193   -0.1274496    0.9942426   -0.1164256 2095.0110758    9.0934781
##
##
## Estimated coefficients for equation PSUST:
## ==========================================
## Call:
## PSUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + const + trend
##
##       IUST.l1     FAIUST.l1      REUST.l1      PSUST.l1         const
##    0.23693539   -0.16500533    0.05420333    0.81033903 4271.46873972
##         trend
##   22.16281293
```

```
print("Forecast values from VAR(p):")
```

```
## [1] "Forecast values from VAR(p):"
```

```
print(fit_varp)
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation IUST:
## =========================================
## Call:
## IUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l2
+ IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
##
##       IUST.l1     FAIUST.l1      REUST.l1      PSUST.l1       IUST.l2
##   0.935303181  -0.033031343   0.009579056  -0.048008907   0.249435490
##     FAIUST.l2      REUST.l2      PSUST.l2       IUST.l3     FAIUST.l3
##   0.052582407  -0.084233810   0.017135010  -0.216274314  -0.031523511
##      REUST.l3      PSUST.l3         const         trend
##   0.080938873   0.012085644 571.589359247   3.034787343
##
##
## Estimated coefficients for equation FAIUST:
## ===========================================
## Call:
## FAIUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l
2 + IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
##
##       IUST.l1     FAIUST.l1      REUST.l1      PSUST.l1       IUST.l2
##    0.38846527    0.82882548    0.03395219   -0.20633392    0.79100771
##     FAIUST.l2      REUST.l2      PSUST.l2       IUST.l3     FAIUST.l3
##    0.14021298   -0.16910975   -0.01708443   -0.68823651   -0.15751869
##      REUST.l3      PSUST.l3         const         trend
##    0.16842112    0.14213526 1443.29663119    6.00036764
##
##
## Estimated coefficients for equation REUST:
## ==========================================
## Call:
## REUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l2
+ IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
##
##       IUST.l1     FAIUST.l1      REUST.l1      PSUST.l1       IUST.l2
##    0.78448769   -0.31369472    1.05270079   -0.23555409   -0.04033799
##     FAIUST.l2      REUST.l2      PSUST.l2       IUST.l3     FAIUST.l3
##   -0.01574587   -0.33015067    0.26649262   -0.31094645    0.30351582
##      REUST.l3      PSUST.l3         const         trend
##    0.25373702   -0.12964122 1157.42637501    5.57784701
##
##
## Estimated coefficients for equation PSUST:
## ==========================================
## Call:
## PSUST = IUST.l1 + FAIUST.l1 + REUST.l1 + PSUST.l1 + IUST.l2 + FAIUST.l2 + REUST.l2 + PSUST.l2
+ IUST.l3 + FAIUST.l3 + REUST.l3 + PSUST.l3 + const + trend
##
```

```
##       IUST.l1     FAIUST.l1      REUST.l1      PSUST.l1       IUST.l2
##    1.00390127   -0.01686731    0.00383979    0.38682235    0.68127046
##     FAIUST.l2      REUST.l2      PSUST.l2       IUST.l3     FAIUST.l3
##    0.04781045   -0.23856352    0.23308357   -1.76822565   -0.08437332
##      REUST.l3      PSUST.l3         const         trend
```

7. Use your fitted model and discuss the Granger causality between the series.

```
fit_var1 <- VAR(data.ts, p=1, type="both")

fit_varp <- VAR(data.ts, p=3, type="both")

print(class(fit_var1))
```

```
## [1] "varest"
```

```
print(class(fit_varp))
```

```
## [1] "varest"
```

```
# For VAR(1)
single_causality_result_var1 <- causality(fit_var1, cause="IUST")
print("Single Variable Granger Causality Results for VAR(1):")
```

```
## [1] "Single Variable Granger Causality Results for VAR(1):"
```

```
print(single_causality_result_var1)
```

```
## $Granger
##
##  Granger causality H0: IUST do not Granger-cause FAIUST REUST PSUST
##
## data:  VAR object fit_var1
## F-Test = 5.4452, df1 = 3, df2 = 916, p-value = 0.001032
##
##
## $Instant
##
##  H0: No instantaneous causality between: IUST and FAIUST REUST PSUST
##
## data:  VAR object fit_var1
## Chi-squared = 108.49, df = 3, p-value < 2.2e-16
```

```
# Testing Granger causality in VAR(1) model
causality_results_var1 <- causality(fit_var1)
```

```
## Warning in causality(fit_var1):
## Argument 'cause' has not been specified;
## using first variable in 'x$y' (IUST) as cause variable.
```

```
print("Granger Causality Results for VAR(1):")
```

```
## [1] "Granger Causality Results for VAR(1):"
```

```
print(causality_results_var1)
```

```
## $Granger
##
##  Granger causality H0: IUST do not Granger-cause FAIUST REUST PSUST
##
## data:  VAR object fit_var1
## F-Test = 5.4452, df1 = 3, df2 = 916, p-value = 0.001032
##
##
## $Instant
##
##  H0: No instantaneous causality between: IUST and FAIUST REUST PSUST
##
## data:  VAR object fit_var1
## Chi-squared = 108.49, df = 3, p-value < 2.2e-16
```

```
# Testing Granger causality in VAR(p) model
causality_results_varp <- causality(fit_varp)
```

```
## Warning in causality(fit_varp):
## Argument 'cause' has not been specified;
## using first variable in 'x$y' (IUST) as cause variable.
```

```
print("Granger Causality Results for VAR(p):")
```

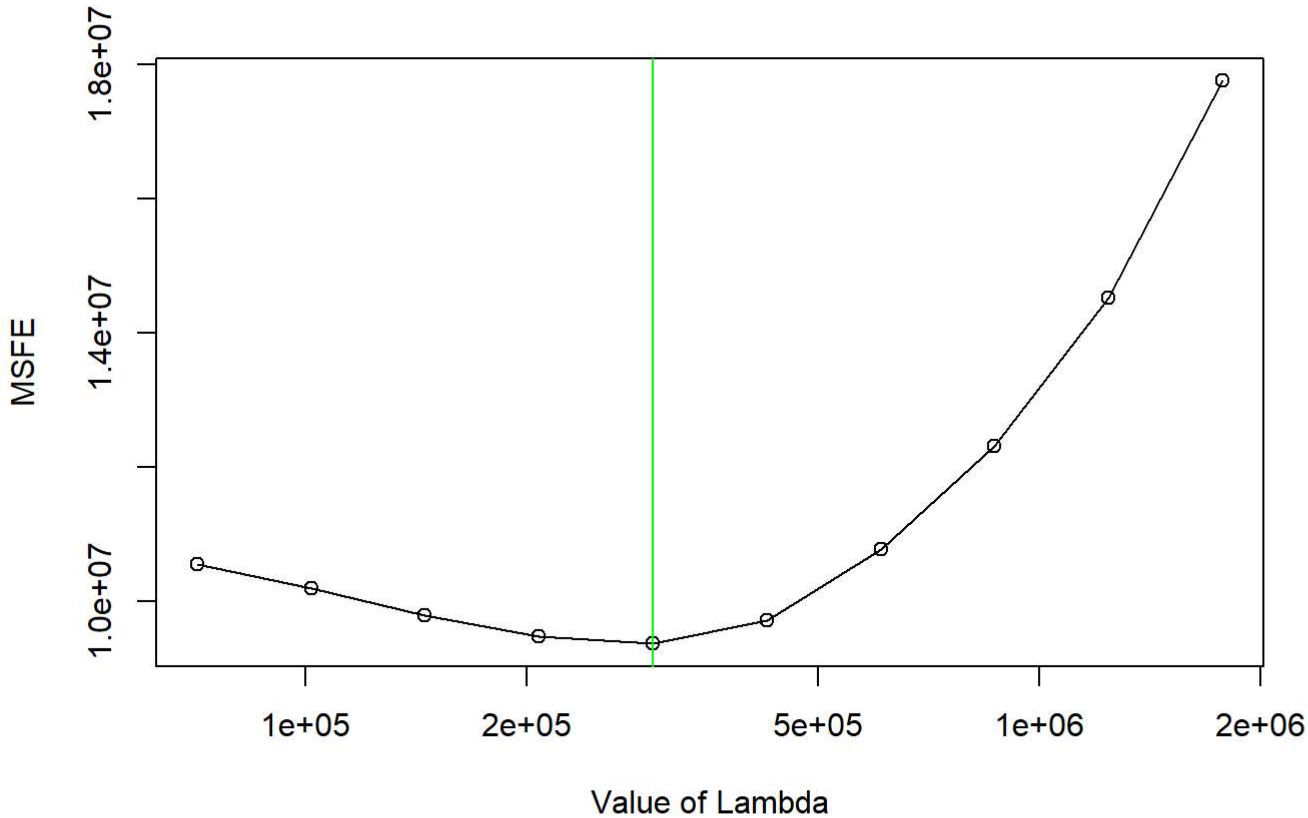
```
## [1] "Granger Causality Results for VAR(p):"
```

```
print(causality_results_varp)
```

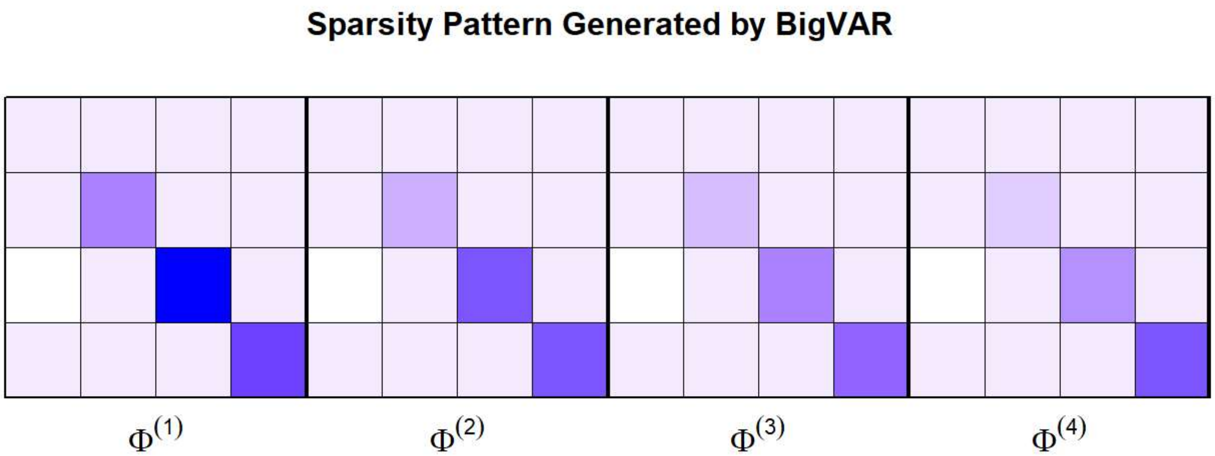
```
## $Granger
##
##  Granger causality H0: IUST do not Granger-cause FAIUST REUST PSUST
##
## data:  VAR object fit_varp
## F-Test = 3.1625, df1 = 9, df2 = 876, p-value = 0.0009104
##
##
## $Instant
##
##  H0: No instantaneous causality between: IUST and FAIUST REUST PSUST
##
## data:  VAR object fit_varp
## Chi-squared = 108.19, df = 3, p-value < 2.2e-16
```

8. Now use the BigVAR package to fit a sparse VAR model. Describe which sparsity structure you picked and what the results tell you.

```
library(BigVAR)

bigvar_model_p <- constructModel(as.matrix(data.ts), p = 4, struct = "SparseOO", gran = c(25,1
0), verbose = FALSE, h = 5, IC = TRUE)

# Cross-validation to select optimal hyperparameters
results_bigvar <- cv.BigVAR(bigvar_model_p)

# Plotting the results of cross-validation and sparsity
plot(results_bigvar)
```




```
SparsityPlot.BigVAR.results(results_bigvar)
```




```
# Forecasting with the BigVAR model
forecast_bigvar <- predict(results_bigvar, n.ahead = 1)
print(forecast_bigvar)
```

```
##            [,1]
## [1,]  6648.995
## [2,] 16273.434
## [3,] 21289.440
## [4,] 55383.900
```