ISYE 6402 Homework 8 Solutions

Background

We have explored how various U.S. economic indicators are related to each other, which is a classic application for the VAR modeling. In this problem, we will study the inter-dependence and Granger causality between various economic indicators.

Below is the data we will be using for this analysis:

GDP- Gross Domestic Product (Quarterly) IMPGSC- Real Imports of Goods and Services (Quarterly) EXPGS- Exports of Goods and Services (Quarterly) UNRATE- Unemployment Rate (Monthly) PCE- Personal Consumption Expenditures (Monthly)

Instructions on reading the data

To read the data in R, save the file in your working directory (make sure you have changed the directory if different from the R working directory) and read the data using the R function `read.csv()`

```
# Read the monthly and quarterly data
fname <- file.choose()
data <- read.csv(fname)
fname2 <- file.choose()
data2 <- read.csv(fname2)
date.quarter <- as.Date(data[,1],"%m/%d/%Y")
date.month <- as.Date(data2[,1],"%m/%d/%Y")
```

Here are the libraries you will need:

```
library(data.table)
library(vars)
library(xts)
library(mgcv)
library(stats)
library(tseries)
library(aod)
```

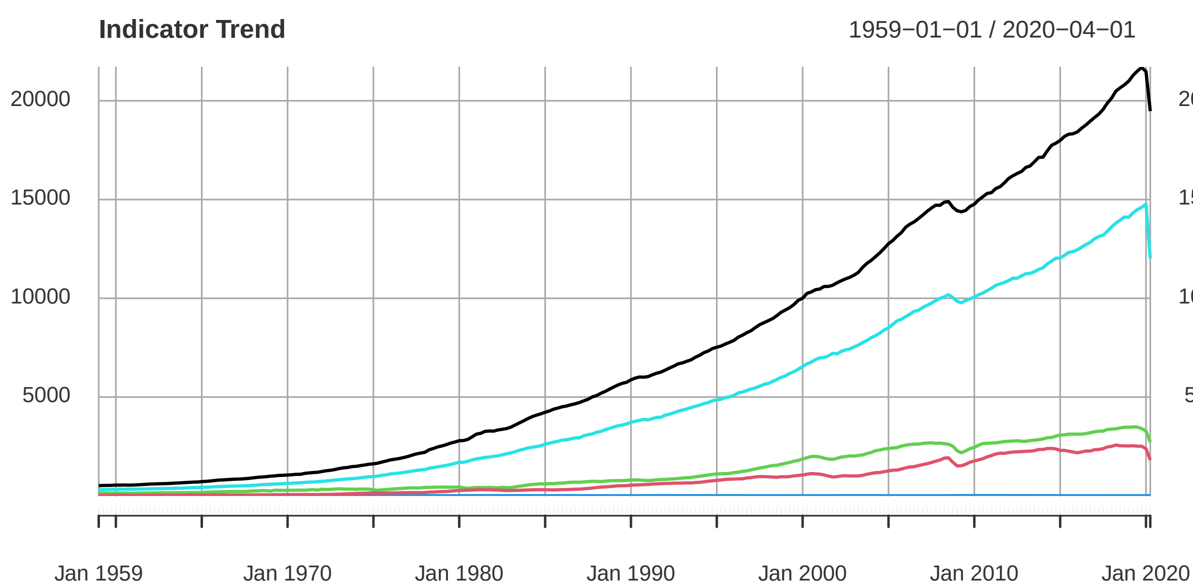
Question 1: Univariate Analysis

Question 1a

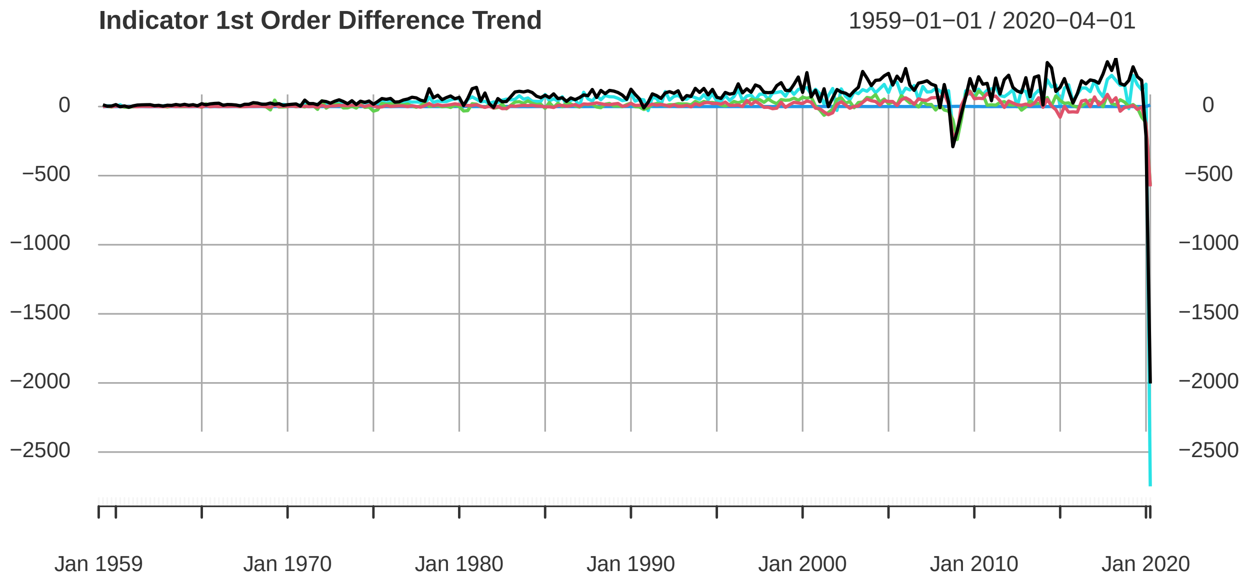
Plot the time series of all indicators for comparison and discuss whether you find any similarities in terms of trend or other features. Plot also the 1st order difference plots and the corresponding ACF plots. Interpret in terms of stationarity and volatility.

Keep in mind, 2 variables have monthly data, while three have quarterly data. You will need to standardise all into quarterly data time series in order to effectively answer all questions below.

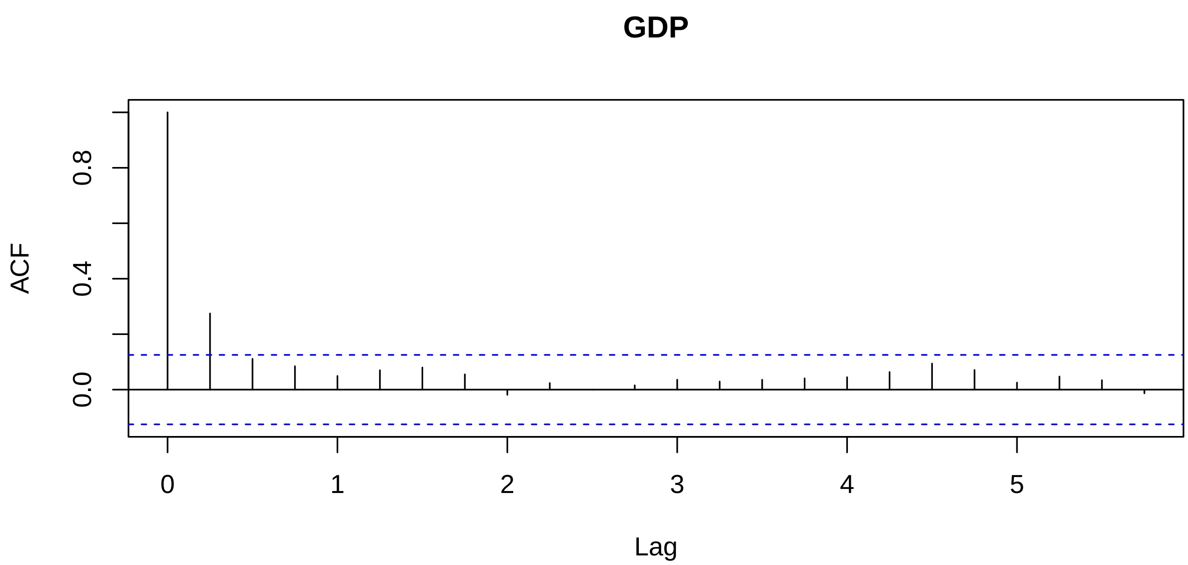
```
# transform into time series
gdp.ts <- xts(data$GDP, date.quarter)
expgs.ts <- xts(data$EXPGS, date.quarter)
imgpgsc.ts <- xts(data$IMPGSC1, date.quarter)
unrate.ts <- xts(data2$UNRATE, date.month)
pce.ts <- xts(data2$PCE, date.month)

# merge into multivariate time series
ts.merge <- merge(gdp.ts, expgs.ts, join = 'inner')
ts.merge <- merge(ts.merge, imgpgsc.ts, join = 'inner')
ts.merge <- merge(ts.merge, unrate.ts, join = 'inner')
ts.merge <- merge(ts.merge, pce.ts, join = 'inner')
colnames(ts.merge) <- c("tsgdp", "tsexpgs", "tsimgpsc", "tsunrate", "tspce")

# plot time series and its first order difference
plot(ts.merge, main = "Indicator Trend")
```



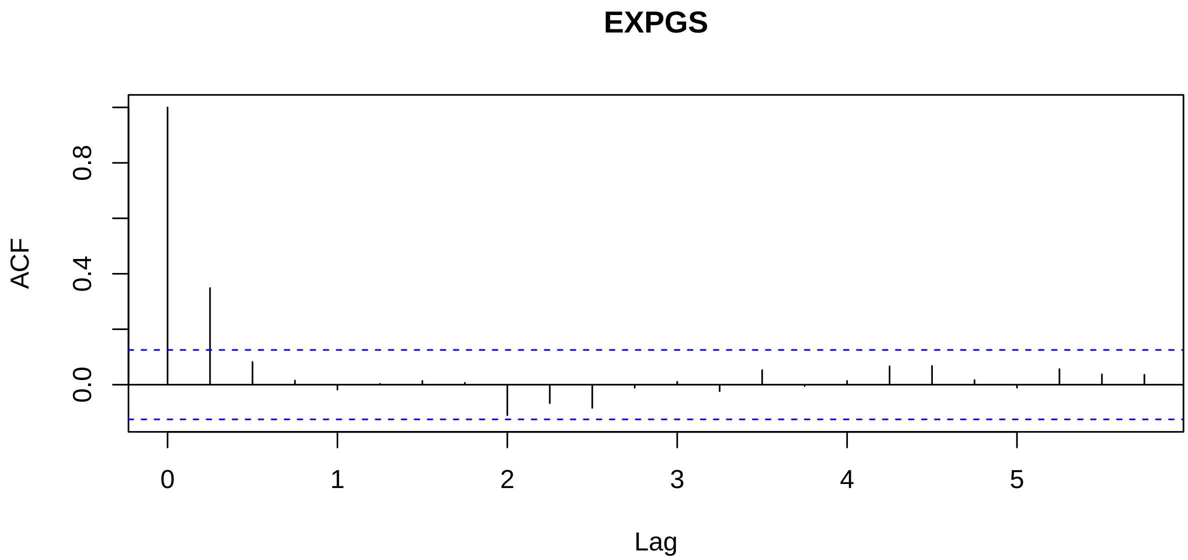

```
ts.merge.diff <- diff(ts.merge)
plot(ts.merge.diff, main = "Indicator 1st Order Difference Trend")
```



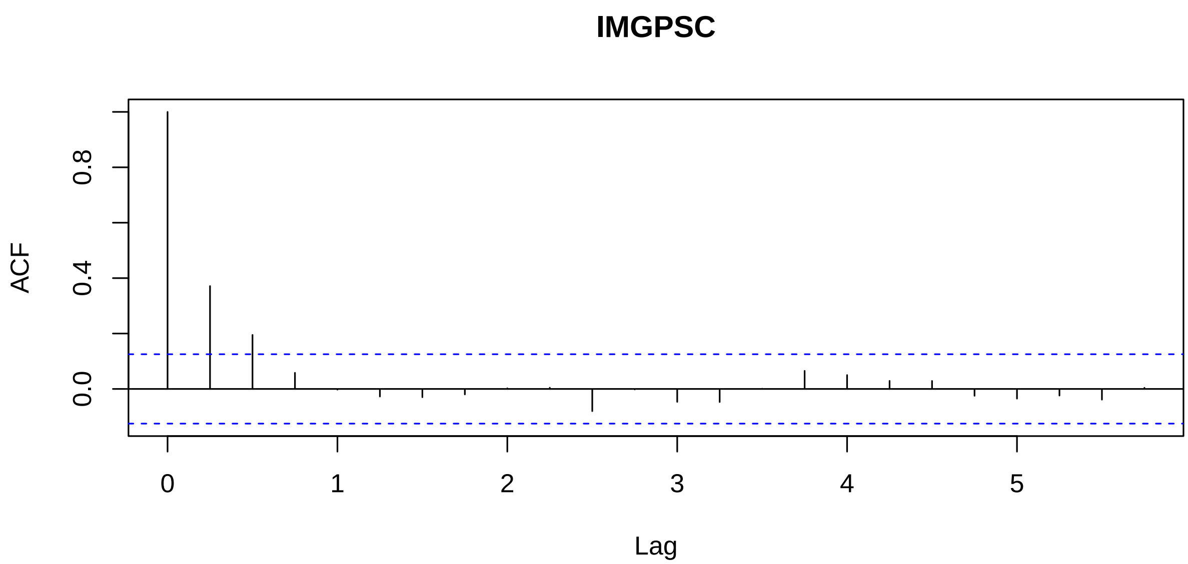

```
# plot correspoding ACF
gdp.ts <- ts(as.numeric(ts.merge$tsgdp), start = c(1959, 1), freq = 4)
gdp.diff <- diff(gdp.ts)
acf(gdp.diff, main = "GDP")
```



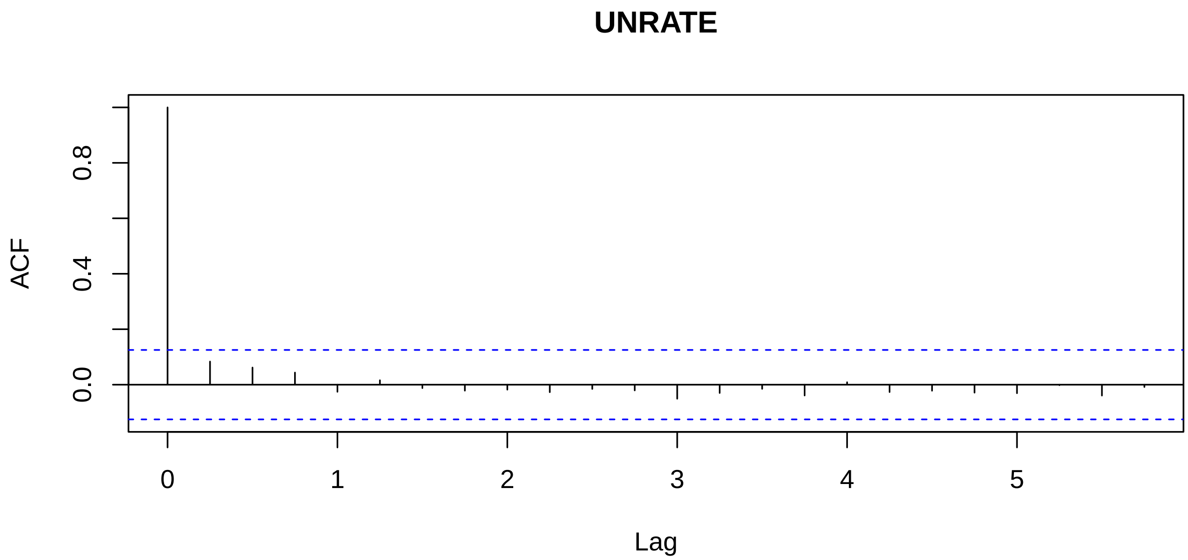

```
expgs.ts <- ts(as.numeric(ts.merge$tsexpgs), start = c(1959, 1), freq = 4)
expgs.diff <- diff(expgs.ts)
acf(expgs.diff, main = "EXPGS")
```



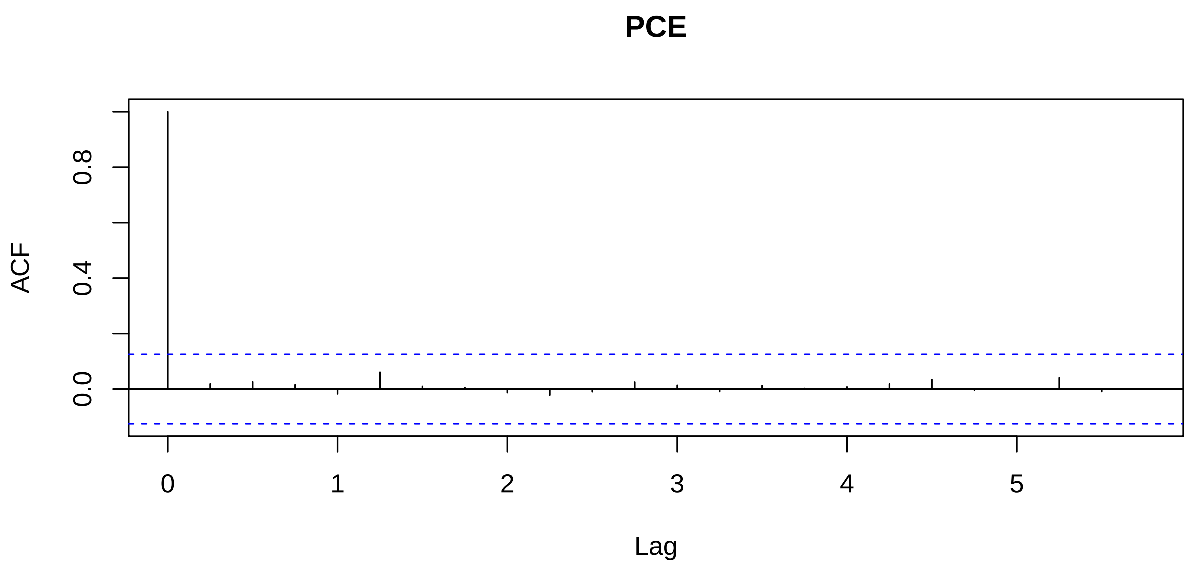

```
imgpsc.ts <- ts(as.numeric(ts.merge$tsimgpsc), start = c(1959, 1), freq = 4)
imgpsc.diff <- diff(imgpsc.ts)
acf(imgpsc.diff, main = "IMGPSC")
```




```
unrate.ts <- ts(as.numeric(ts.merge$tsunrate), start = c(1959, 1), freq = 4)
unrate.diff <- diff(unrate.ts)
acf(unrate.diff, main = "UNRATE")
```




```r
pce.ts <- ts(as.numeric(ts.merge$tspce), start = c(1959, 1), freq = 4)
pce.diff <- diff(pce.ts)
acf(pce.diff, main = "PCE")
```



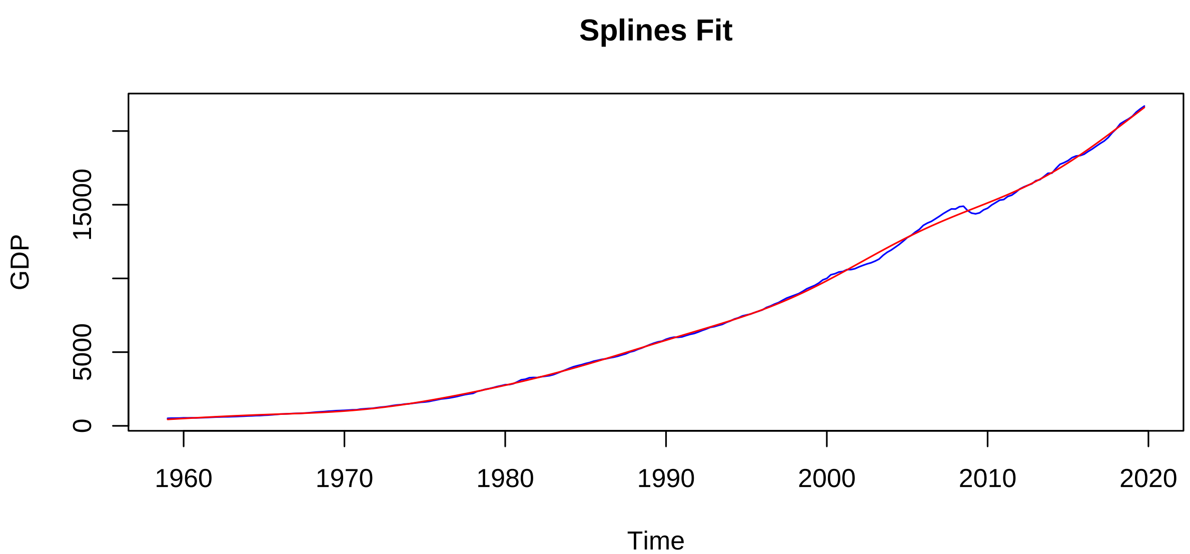
Response

All of the indicators have an increasing trend. When exploring the difference time series, we also learn that the variability increases over time, with very large volatility in 2020 as expected due to the covid19 pandemic. The ACF plots of the difference time series show no significant autocorrelation.

Question 1b

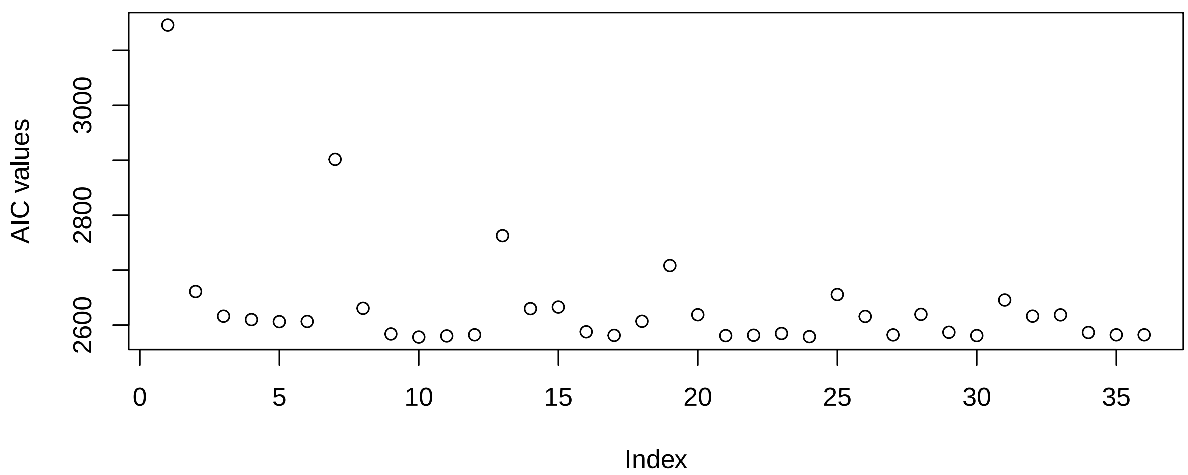
Using the original, undifferenced data, divide the GDP data into training data including the data for years 1959 to 2019 with the last two quarters being the testing data. Fit the trend using the splines regression to the GDP training time series. Then, apply ARMA to the residuals obtained from this splines fitting. Use max order of 6. Evaluate goodness of fit for the ARMA model. Forecast the first two quarters of 2020 (testing data) and compare to the observed values. Discuss why there are (or not!) significant differences between predicted vs observed. To do this, you should also evaluate the prediction intervals with a 95% confidence level.

```
gdp.train <- gdp.ts[0:(length(gdp.ts) - 2)]
gdp.test <- gdp.ts[(length(gdp.ts) - 1):length(gdp.ts)]
gdp.train.ts <- ts(gdp.train, start = 1959, frequency = 4)
gdp.test.ts <- ts(gdp.test, start = 2020, frequency = 4)
time.pts <- c(1:length(gdp.train))
time.pts <- c(time.pts - min(time.pts)) / max(time.pts)
gam.fit <- gam(gdp.train.ts ~ s(time.pts))
temp.fit.gam <- ts(fitted(gam.fit), start = 1959, frequency = 4)
ts.plot(gdp.train.ts, col = 'blue', main = 'Splines Fit', ylab = 'GDP')
lines(temp.fit.gam, col = 'red')
```

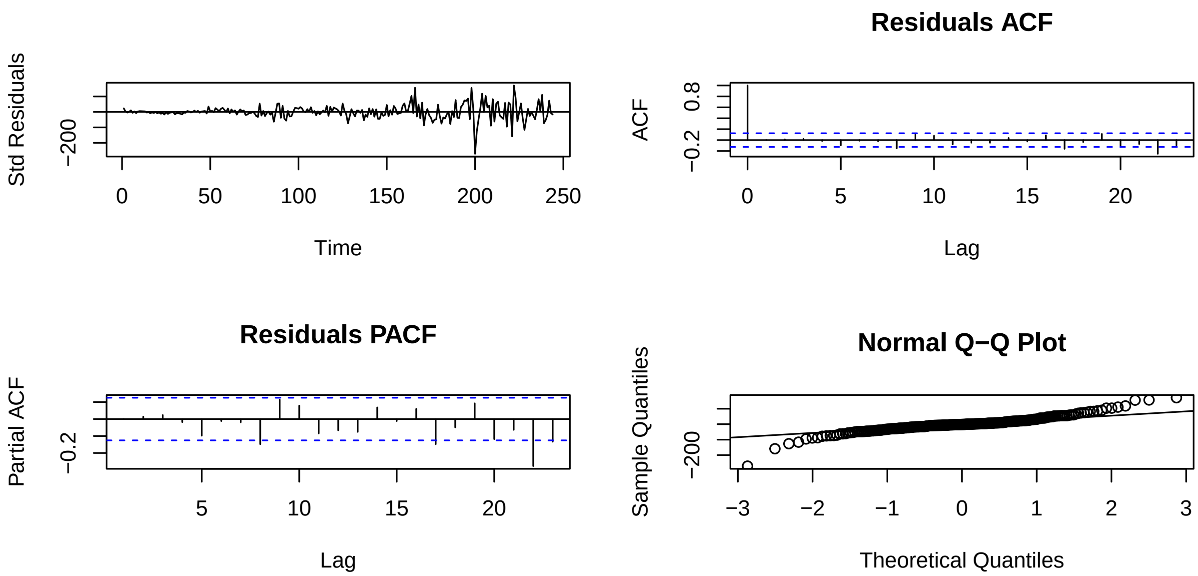



```
resid <- resid(gam.fit)
## Order selection -- AIC
n <- length(resid)
norder <- 6
p <- c(1:norder) - 1
q <- c(1:norder) - 1
aic <- matrix(0, norder, norder)
for (i in 1:norder) {
  for (j in 1:norder) {
    modij <- arima(resid, order = c(p[i], 0, q[j]), method = 'ML')
    aic[i, j] <- modij$aic - 2 * (p[i] + q[j] + 1) + 2 * (p[i] + q[j] + 1) *
    n / (n - p[i] - q[j] - 2)
```

```
  }
}
aicv <- as.vector(aic)
plot(aicv, ylab = "AIC values")
```




```
indexp <- rep(c(1:norder), norder)
indexq <- rep(c(1:norder), each = norder)
indexaic <- which(aicv == min(aicv))
porder <- indexp[indexaic] - 1
qorder <- indexq[indexaic] - 1
final_model <- arima(resid, order = c(porder, 0, qorder), method = 'ML')
par(mfrow = c(2, 2))
plot(resid(final_model), ylab = 'Std Residuals')
abline(h = 0)
acf(as.vector(resid(final_model)), main = 'Residuals ACF')
pacf(as.vector(resid(final_model)), main = 'Residuals PACF')
qqnorm(resid(final_model))
qqline(resid(final_model))
```




```r
#### Test for Uncorrelated Residuals for the final model
Box.test(
final_model$resid,
lag = (porder + qorder + 1),
type = "Box-Pierce",
fitdf = (porder + qorder)
)
```

```
##
##  Box-Pierce test
##
## data:  final_model$resid
## X-squared = 2.5789, df = 1, p-value = 0.1083
```

```r
Box.test(
final_model$resid,
lag = (porder + qorder + 1),
type = "Ljung-Box",
fitdf = (porder + qorder)
)
```

```
##
##  Box-Ljung test
##
## data:  final_model$resid
## X-squared = 2.6523, df = 1, p-value = 0.1034
```

```r
jarque.bera.test(resid(final_model))
```

```
##
##  Jarque Bera Test
```

```
##
## data:  resid(final_model)
## X-squared = 415.85, df = 2, p-value < 2.2e-16
```

```
shapiro.test(resid(final_model))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(final_model)
## W = 0.90656, p-value = 3.358e-11
```

```
final_model_pred <- arima(gdp.train.ts, order = c(porder, 0, qorder),
method = 'ML')
arima_pred <- as.vector(predict(final_model_pred, n.ahead = 2))

# Observed
gdp.test.ts
```

```
##          Qtr1     Qtr2
## 2020 21481.37 19477.44
```

```
# Predicted
arima_pred
```

```
## $pred
##          Qtr1     Qtr2
## 2020 21843.22 21984.89
##
## $se
##           Qtr1      Qtr2
## 2020  60.80108 127.53215
```

```
# 95% Prediction Intervals
arima_pred$pred[1] + c(-1, 1) * qnorm((1 - 0.05 / 2)) * arima_pred$se[1]
```

```
## [1] 21724.05 21962.38
```

```
arima_pred$pred[2] + c(-1, 1) * qnorm((1 - 0.05 / 2)) * arima_pred$se[2]
```

```
## [1] 21734.93 22234.85
```

Response

Both JB test and SW test indicate lack of normality in residuals and the Box tests indicate that we fail to reject the null hypothesis that residuals do not exhibit serial correlation or heteroscedasticity. The observed values are outside of the predicted confidence intervals, thus the quality of the prediction is not good. This is because the original data is non-stationary, and the ARMA model does not fit the data appropriately.

Question 1c

Perform a similar analysis as in (1b) but this time applying ARIMA to the GDP time series training dataset. Compare the forecast and discuss why these are different or similar from the testing data. Assume p,q belong to [0,5] and d belongs to [0,1]. Also evaluate the prediction intervals with a 95% confidence level, and compare the results of the forecast with the analysis in Question 1b.

```
test_model <- function(p, d, q) {
  mod = arima(gdp.train.ts, order = c(p, d, q), method = "ML")
  current.aic = AIC(mod)

  df = data.frame(p, d, q, current.aic)
  names(df) <- c("p", "d", "q", "AIC")
  # print(paste(p, d, q, current.aic, sep = " "))
  return(df)
}
orders <- data.frame(Inf, Inf, Inf, Inf)
names(orders) <- c("p", "d", "q", "AIC")
for (p in 0:5) {
  for (d in 0:1) {
    for (q in 0:5) {
    possibleError <- tryCatch(
    orders <- rbind(orders, test_model(p, d, q)),
    error = function(e) {}
    )
    if (inherits(possibleError, "error"))
    next
    }
  }
}
orders <- orders[order(-orders$AIC), ]
tail(orders)
```

```
##    p d q      AIC
## 56 4 1 4 2649.353
## 53 4 1 1 2648.955
## 42 3 1 1 2648.247
## 66 5 1 3 2647.510
## 33 2 1 2 2647.441
## 68 5 1 5 2646.630
```

```
arima.gdp <- arima(gdp.train.ts, order = c(5, 1, 5), method = 'ML')
arima_pred1 <- as.vector(predict(arima.gdp, n.ahead = 2))
#### Test for Uncorrelated Residuals for the final model
Box.test(
  arima.gdp$resid,
  lag = (5 + 5 + 1),
  type = "Box-Pierce",
  fitdf = (5 + 5)
)
```

```
##
##  Box-Pierce test
```

```
##
## data:  arima.gdp$resid
## X-squared = 17.124, df = 1, p-value = 3.502e-05
```

```
Box.test(
  arima.gdp$resid,
  lag = (5 + 5 + 1),
  type = "Ljung-Box",
  fitdf = (5 + 5)
)
```

```
##
##  Box-Ljung test
##
## data:  arima.gdp$resid
## X-squared = 17.807, df = 1, p-value = 2.444e-05
```

```
jarque.bera.test(resid(arima.gdp))
```

```
##
##  Jarque Bera Test
##
## data:  resid(arima.gdp)
## X-squared = 1543.2, df = 2, p-value < 2.2e-16
```

```
shapiro.test(resid(arima.gdp))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resid(arima.gdp)
## W = 0.85792, p-value = 3.17e-14
```

```
arima_pred1
```

```
## $pred
##          Qtr1     Qtr2
## 2020 21888.66 22084.99
##
## $se
##          Qtr1     Qtr2
## 2020 52.84697 93.79315
```

```
gdp.test.ts
```

```
##          Qtr1     Qtr2
## 2020 21481.37 19477.44
```

```
# 95% Prediction Intervals
arima_pred1$pred[1] + c(-1, 1) * qnorm((1 - 0.05 / 2)) * arima_pred1$se[1]
```

```
## [1] 21785.09 21992.24
```

```
arima_pred1$pred[2] + c(-1, 1) * qnorm((1 - 0.05 / 2)) * arima_pred1$se[2]
```

```
## [1] 21901.16 22268.82
```

Response

Both JB test and SW test indicate lack of normality in residuals however, the Box tests indicate residuals do exhibit serial correlation or heteroscedasticity. ARIMA fit has a better prediction accuracy than ARMA since the predictions are closer in values to the observed ones, however, the prediction intervals still do not contain the observed values.

Question 2: Multivariate Analysis using VAR modeling

For this question, divide the quarterly data into training data (excluding the first two quarters of 2020) and testing data (including the two quarters). You will apply the modeling to the training data, and we will forecast the first two quarters of 2020.

Question 2a

Apply the VAR model to the multivariate time series including all five economic indicators observed quarterly. (Note that you will apply VAR to the training data.) Identify the VAR order using both AIC and BIC and compare. If the selected order using AIC is larger than the selected order than selected using BIC, apply the Wald test to evaluate whether a smaller order than the one selected with AIC would be a better choice, meaning the smaller order model would perform similarly than the larger order model. Interpret the order selection.

This can be done by following the below substeps:- 1)Combine the variables into a multivariate dataset 2)select/display var orders and isolate models using AIC and BIC orders 3)Isolate coefficients and covariances from the AIC model 4)applying the Wald test to the values obtained from the AIC model, but would not be present in the BIC model. You can run a single Wald test for each variable, with all the lagged coefficients for that particular variable, in all resulting in 5 wald tests.

```
expgs.train <- expgs.ts[0:(length(expgs.ts) - 2)]
imgpgsc.train <- imgpgsc.ts[0:(length(imgpgsc.ts) - 2)]
unrate.train <- unrate.ts[0:(length(unrate.ts) - 2)]
pce.train <- pce.ts[0:(length(pce.ts) - 2)]


expgs.test <- expgs.ts[(length(expgs.ts) - 1):length(expgs.ts)]
imgpgsc.test <- imgpgsc.ts[(length(imgpgsc.ts) - 1):length(imgpgsc.ts)]
unrate.test <- unrate.ts[(length(unrate.ts) - 1):length(unrate.ts)]
pce.test <- pce.ts[(length(pce.ts) - 1):length(pce.ts)]


expgs.train.ts <- ts(expgs.train, start = 1959, frequency = 4)
```

```r
imgpgsc.train.ts <- ts(imgpgsc.train, start = 1959, frequency = 4)
unrate.train.ts <- ts(unrate.train, start = 1959, frequency = 4)
pce.train.ts <- ts(pce.train, start = 1959, frequency = 4)


expgs.test.ts <- ts(expgs.test, start = 2020, frequency = 4)
imgpgsc.test.ts <- ts(imgpgsc.test, start = 2020, frequency = 4)
unrate.test.ts <- ts(unrate.test, start = 2020, frequency = 4)
pce.test.ts <- ts(pce.test, start = 2020, frequency = 4)


train.ts <- cbind(gdp.train.ts,
                  expgs.train.ts,
                  imgpgsc.train.ts,
                  unrate.train.ts,
                  pce.train.ts)
test.ts <- cbind(gdp.test.ts,
                expgs.test.ts,
                imgpgsc.test.ts,
                unrate.test.ts,
                pce.test.ts)
vs.aic <- VARselect(train.ts)
vs.aic$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      7      3      2      7
```

```r
model.aic <- VAR(train.ts, p = 7)
model.bic <- VAR(train.ts, p = 2)


pord_1 = model.aic$p

## Does a smaller order fit the model equally well? Apply Wald Test
## Coefficients for orders 3 to 12
coef.gdp.3to7 <- coefficients(model.aic)$gdp.train.ts[11:(5 * pord_1), 1]
coef.expgs.3to7 <- coefficients(model.aic)$expgs.train.ts[11:(5 * pord_1), 1]
coef.imgpgsc.3to7 <- coefficients(model.aic)$imgpgsc.train.ts[11:(5 * pord_1), 1]
coef.unrate.3to7 <- coefficients(model.aic)$unrate.train.ts[11:(5 * pord_1), 1]
coef.pce.3to7 <- coefficients(model.aic)$pce.train.ts[11:(5 * pord_1), 1]

## Covariance matrix of the coefficients
index.gdp <- 12:(5 * pord_1 + 1)
var.gdp.3to7 <- vcov(model.aic)[index.gdp, index.gdp]

index.expgs <- c(((5 * pord_1) + 13):(10 * pord_1 + 2))
var.expgs.3to7 <- vcov(model.aic)[index.expgs, index.expgs]

index.imgpgsc <- c(((10 * pord_1) + 14):(15 * pord_1 + 3))
var.imgpgsc.3to7 <- vcov(model.aic)[index.imgpgsc, index.imgpgsc]

index.unrate <- c(((15 * pord_1) + 15):(20 * pord_1 + 4))
var.unrate.3to7 <- vcov(model.aic)[index.unrate, index.unrate]
```

```r
index.pce <- c(((20 * pord_1) + 16):(25 * pord_1 + 5))
var.pce.3to7 <- vcov(model.aic)[index.pce, index.pce]

## Apply Wald Test
wald.test(b = coef.gdp.3to7, var.gdp.3to7, Terms = seq(1, 5 * (pord_1 - 5)))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 14.7, df = 10, P(> X2) = 0.14
```

```r
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 44.7, df = 15, P(> X2) = 8.6e-05
wald.test(b = coef.expgs.3to7, var.expgs.3to7, Terms = seq(1, 5 * (pord_1 - 5)))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 43.9, df = 10, P(> X2) = 3.5e-06
```

```r
wald.test(b = coef.imgpgsc.3to7, var.imgpgsc.3to7, Terms = seq(1, 5 * (pord_1 - 5)))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 7.5, df = 10, P(> X2) = 0.68
```

```r
wald.test(b = coef.unrate.3to7, var.unrate.3to7, Terms = seq(1, 5 * (pord_1 - 5)))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 16.4, df = 10, P(> X2) = 0.09
```

```r
wald.test(b = coef.pce.3to7, var.pce.3to7, Terms = seq(1, 5 * (pord_1 - 5)))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 34.7, df = 10, P(> X2) = 0.00014
```

Response

The order selected using AIC is 7 and using BIC is 2. According to Wald Test applied to evaluate whether a model of order 2 performs similarly to the model of order 8, we fail to reject the null hypothesis since the p-values for gdp, imports and unemployment rate are significant.

Question 2b

Based on the analysis in 2a, select the VAR order using BIC and fit that model. Print out the model summary and comment on the statistical significance of the coefficients. Apply a model selection analysis using stepwise regression to select the models for each individual time series. What do you conclude from this model selection? Apply the restrict() command in R to restrict the model of order. How do the restricted models compare?

Follow the below steps to implement this: 1)Analyze the coefficients of the unrestricted model. 2)Then treat each of the series separately and do a stepwise regression by apply lm() and step() (in the backward direction with 3 steps) functions to each of the component time series to examine the coefficients 3)The stepwise regressions will return three separate models. Analyze if the same coefficients are significant in the overall VAR versus each of the stepwise models. Discuss. 4)Then build a model using restrict and see which predictors were significant in the restricted VAR model.

```
summary(model.bic)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gdp.train.ts, expgs.train.ts, imgpgsc.train.ts, unrate.train.ts, pce.train.ts
## Deterministic variables: const
## Sample size: 242
## Log Likelihood: -4590.213
## Roots of the characteristic polynomial:
## 1.009 0.9786 0.9786 0.9495 0.8059 0.8059 0.3523 0.3523 0.2171 0.02384
## Call:
## VAR(y = train.ts, p = 2)
##
##
## Estimation results for equation gdp.train.ts:
## ===============================================
## gdp.train.ts = gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 + pce.t
##
##                       Estimate Std. Error t value Pr(>|t|)
## gdp.train.ts.l1        1.24053    0.08407  14.756  < 2e-16 ***
## expgs.train.ts.l1     -0.05900    0.15708  -0.376   0.7075
## imgpgsc.train.ts.l1    0.61018    0.14409   4.235 3.31e-05 ***
## unrate.train.ts.l1   -13.88253   10.74571  -1.292   0.1977
## pce.train.ts.l1       -0.04535    0.12278  -0.369   0.7122
## gdp.train.ts.l2       -0.20571    0.09197  -2.237   0.0263 *
## expgs.train.ts.l2     -0.02598    0.15757  -0.165   0.8692
## imgpgsc.train.ts.l2   -0.62147    0.14569  -4.266 2.91e-05 ***
## unrate.train.ts.l2    15.98737   10.78624   1.482   0.1396
## pce.train.ts.l2        0.02166    0.11127   0.195   0.8458
## const                 -9.49036   18.67542  -0.508   0.6118
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 51.15 on 231 degrees of freedom
## Multiple R-Squared: 0.9999,  Adjusted R-squared: 0.9999
## F-statistic: 3.627e+05 on 10 and 231 DF,  p-value: < 2.2e-16
##
```

```
##
## Estimation results for equation expgs.train.ts:
## ===============================================
## expgs.train.ts = gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 + pc
##
##                     Estimate Std. Error t value Pr(>|t|)
## gdp.train.ts.l1      0.04455    0.04107   1.085 0.279161
## expgs.train.ts.l1    1.53112    0.07673  19.955  < 2e-16 ***
## imgpgsc.train.ts.l1  0.01470    0.07038   0.209 0.834735
## unrate.train.ts.l1  -5.89857    5.24892  -1.124 0.262278
## pce.train.ts.l1     -0.16377    0.05997  -2.731 0.006806 **
## gdp.train.ts.l2     -0.05734    0.04493  -1.276 0.203113
## expgs.train.ts.l2   -0.59861    0.07697  -7.777 2.44e-13 ***
## imgpgsc.train.ts.l2 -0.02221    0.07116  -0.312 0.755275
## unrate.train.ts.l2   7.48698    5.26872   1.421 0.156658
## pce.train.ts.l2      0.19837    0.05435   3.650 0.000325 ***
## const               -9.49703    9.12232  -1.041 0.298929
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 24.99 on 231 degrees of freedom
## Multiple R-Squared: 0.9991,  Adjusted R-squared: 0.999
## F-statistic: 2.489e+04 on 10 and 231 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation imgpgsc.train.ts:
## =================================================
## imgpgsc.train.ts = gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##
##                     Estimate Std. Error t value Pr(>|t|)
## gdp.train.ts.l1      0.14405    0.04059   3.549 0.000468 ***
## expgs.train.ts.l1    0.16731    0.07584   2.206 0.028358 *
## imgpgsc.train.ts.l1  1.29563    0.06956  18.625  < 2e-16 ***
## unrate.train.ts.l1  -0.91999    5.18789  -0.177 0.859402
## pce.train.ts.l1     -0.03315    0.05927  -0.559 0.576467
## gdp.train.ts.l2     -0.15856    0.04440  -3.571 0.000433 ***
## expgs.train.ts.l2   -0.21342    0.07607  -2.806 0.005452 **
## imgpgsc.train.ts.l2 -0.33258    0.07034  -4.729 3.93e-06 ***
## unrate.train.ts.l2   2.17360    5.20746   0.417 0.676773
## pce.train.ts.l2      0.07143    0.05372   1.330 0.184914
## const               -6.60988    9.01626  -0.733 0.464236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 24.7 on 231 degrees of freedom
## Multiple R-Squared: 0.9995,  Adjusted R-squared: 0.9995
## F-statistic: 4.549e+04 on 10 and 231 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation unrate.train.ts:
## ================================================
## unrate.train.ts = gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 + p
```

```
##
##                        Estimate Std. Error t value Pr(>|t|)
## gdp.train.ts.l1     -0.0014221  0.0004759  -2.988  0.00311 **
## expgs.train.ts.l1    0.0004526  0.0008893   0.509  0.61127
## imgpgsc.train.ts.l1 -0.0025885  0.0008157  -3.173  0.00171 **
## unrate.train.ts.l1   1.3092148  0.0608326  21.522  < 2e-16 ***
## pce.train.ts.l1      0.0006595  0.0006950   0.949  0.34368
## gdp.train.ts.l2      0.0015192  0.0005207   2.918  0.00387 **
## expgs.train.ts.l2   -0.0005709  0.0008920  -0.640  0.52282
## imgpgsc.train.ts.l2  0.0026413  0.0008247   3.203  0.00155 **
## unrate.train.ts.l2  -0.3371662  0.0610621  -5.522    9e-08 ***
## pce.train.ts.l2     -0.0007812  0.0006299  -1.240  0.21617
## const                0.1865111  0.1057235   1.764  0.07903 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2896 on 231 degrees of freedom
## Multiple R-Squared: 0.9685,  Adjusted R-squared: 0.9671
## F-statistic: 709.7 on 10 and 231 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation pce.train.ts:
## =============================================
## pce.train.ts = gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 + pce.t
##
##                       Estimate Std. Error t value Pr(>|t|)
## gdp.train.ts.l1        0.33000    0.05269   6.263 1.82e-09 ***
## expgs.train.ts.l1      0.41087    0.09845   4.174 4.25e-05 ***
## imgpgsc.train.ts.l1    0.13868    0.09030   1.536 0.125971
## unrate.train.ts.l1    -9.03770    6.73443  -1.342 0.180909
## pce.train.ts.l1        0.55018    0.07694   7.150 1.13e-11 ***
## gdp.train.ts.l2       -0.22317    0.05764  -3.872 0.000141 ***
## expgs.train.ts.l2     -0.45765    0.09875  -4.634 5.99e-06 ***
## imgpgsc.train.ts.l2   -0.07959    0.09130  -0.872 0.384267
## unrate.train.ts.l2     9.96183    6.75984   1.474 0.141929
## pce.train.ts.l2        0.29464    0.06973   4.225 3.44e-05 ***
## const                -22.89789   11.70406  -1.956 0.051622 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 32.06 on 231 degrees of freedom
## Multiple R-Squared: 0.9999,  Adjusted R-squared: 0.9999
## F-statistic: 4.301e+05 on 10 and 231 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                   gdp.train.ts expgs.train.ts imgpgsc.train.ts unrate.train.ts
## gdp.train.ts          2616.657        807.180          394.751        -2.14165
## expgs.train.ts         807.180        624.334          242.657        -0.47096
## imgpgsc.train.ts       394.751        242.657          609.900        -1.36627
## unrate.train.ts         -2.142         -0.471           -1.366         0.08386
```

```
## pce.train.ts          975.718        349.345          231.194        -0.97803
##                    pce.train.ts
## gdp.train.ts          975.718
## expgs.train.ts        349.345
## imgpgsc.train.ts      231.194
## unrate.train.ts        -0.978
## pce.train.ts         1027.730
##
## Correlation matrix of residuals:
##                    gdp.train.ts expgs.train.ts imgpgsc.train.ts unrate.train.ts
## gdp.train.ts           1.0000        0.63152           0.3125        -0.14458
## expgs.train.ts         0.6315        1.00000           0.3932        -0.06509
## imgpgsc.train.ts       0.3125        0.39324           1.0000        -0.19104
## unrate.train.ts       -0.1446       -0.06509          -0.1910         1.00000
## pce.train.ts           0.5950        0.43612           0.2920        -0.10535
##                    pce.train.ts
## gdp.train.ts           0.5950
## expgs.train.ts         0.4361
## imgpgsc.train.ts       0.2920
## unrate.train.ts       -0.1054
## pce.train.ts           1.0000
```

```r
model_1 <- step(lm(y~.,data = model.bic$varresult$expgs.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1568.42
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + const
##
##
## Step:  AIC=1568.42
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2
##
##                         Df Sum of Sq    RSS    AIC
## - imgpgsc.train.ts.l1  1        27 144248 1566.5
## - imgpgsc.train.ts.l2  1        61 144282 1566.5
## - gdp.train.ts.l1      1       735 144956 1567.7
## - unrate.train.ts.l1   1       788 145010 1567.7
## - gdp.train.ts.l2      1      1017 145238 1568.1
## <none>                             144221 1568.4
## - unrate.train.ts.l2   1      1261 145482 1568.5
## - pce.train.ts.l1      1      4656 148877 1574.1
## - pce.train.ts.l2      1      8317 152538 1580.0
## - expgs.train.ts.l2    1     37765 181986 1622.7
## - expgs.train.ts.l1    1    248602 392824 1808.9
##
## Step:  AIC=1566.47
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l2 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
```

```
##     unrate.train.ts.l2 + pce.train.ts.l2
##
##                        Df Sum of Sq    RSS    AIC
## - imgpgsc.train.ts.l2  1       122 144371 1564.7
## - gdp.train.ts.l1      1       774 145022 1565.8
## - unrate.train.ts.l1   1       960 145208 1566.1
## - gdp.train.ts.l2      1      1055 145304 1566.2
## <none>                             144248 1566.5
## - unrate.train.ts.l2   1      1505 145753 1567.0
## - pce.train.ts.l1      1      4652 148901 1572.2
## - pce.train.ts.l2      1      8383 152631 1578.1
## - expgs.train.ts.l2    1     42632 186880 1627.1
## - expgs.train.ts.l1    1    274692 418940 1822.5
##
## Step:  AIC=1564.67
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l2 + expgs.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2
```

```
model_2 <- step(lm(y~.,data = model.bic$varresult$imgpgsc.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1562.76
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + const
##
##
## Step:  AIC=1562.76
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2
##
##                        Df Sum of Sq    RSS    AIC
## - unrate.train.ts.l1   1        19 140906 1560.8
## - unrate.train.ts.l2   1       106 140993 1560.9
## - pce.train.ts.l1      1       191 141078 1561.1
## - pce.train.ts.l2      1      1078 141965 1562.6
## <none>                             140887 1562.8
## - expgs.train.ts.l1    1      2968 143855 1565.8
## - expgs.train.ts.l2    1      4800 145687 1568.9
## - gdp.train.ts.l1      1      7683 148569 1573.6
## - gdp.train.ts.l2      1      7777 148664 1573.8
## - imgpgsc.train.ts.l2  1     13637 154524 1583.1
## - imgpgsc.train.ts.l1  1    211564 352451 1782.7
##
## Step:  AIC=1560.79
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l2 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2
##
##                        Df Sum of Sq    RSS    AIC
```

```
## - pce.train.ts.l1      1       187 141093 1559.1
## - unrate.train.ts.l2   1       715 141621 1560.0
## - pce.train.ts.l2      1      1063 141970 1560.6
## <none>                           140906 1560.8
## - expgs.train.ts.l1    1      2955 143861 1563.8
## - expgs.train.ts.l2    1      4829 145735 1567.0
## - gdp.train.ts.l1      1      8078 148985 1572.3
## - gdp.train.ts.l2      1      8079 148985 1572.3
## - imgpgsc.train.ts.l2  1     14984 155890 1583.2
## - imgpgsc.train.ts.l1  1    232819 373725 1794.8
##
## Step:  AIC=1559.11
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     gdp.train.ts.l2 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2
```

```
model_3 <- step(lm(y~.,data = model.bic$varresult$unrate.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=-589.08
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + const
##
##
## Step:  AIC=-589.08
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2
##
##                       Df Sum of Sq    RSS     AIC
## - expgs.train.ts.l1    1     0.022 19.393 -590.81
## - expgs.train.ts.l2    1     0.034 19.406 -590.65
## - pce.train.ts.l1      1     0.076 19.447 -590.14
## - pce.train.ts.l2      1     0.129 19.500 -589.48
## <none>                           19.371 -589.08
## - gdp.train.ts.l2      1     0.714 20.085 -582.33
## - gdp.train.ts.l1      1     0.749 20.120 -581.91
## - imgpgsc.train.ts.l1  1     0.844 20.216 -580.76
## - imgpgsc.train.ts.l2  1     0.860 20.232 -580.57
## - unrate.train.ts.l2   1     2.557 21.928 -561.08
## - unrate.train.ts.l1   1    38.842 58.213 -324.81
##
## Step:  AIC=-590.81
## y ~ gdp.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l2 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2
##
##                       Df Sum of Sq    RSS     AIC
## - expgs.train.ts.l2    1     0.043 19.436 -592.27
## - pce.train.ts.l1      1     0.106 19.499 -591.50
## - pce.train.ts.l2      1     0.151 19.545 -590.93
```

```
## <none>                            19.393 -590.81
## - gdp.train.ts.l2      1     0.746 20.139 -583.68
## - gdp.train.ts.l1      1     0.756 20.149 -583.56
## - imgpgsc.train.ts.l1  1     0.841 20.234 -582.54
## - imgpgsc.train.ts.l2  1     0.859 20.252 -582.32
## - unrate.train.ts.l2   1     2.630 22.023 -562.04
## - unrate.train.ts.l1   1    39.668 59.061 -323.31
##
## Step:  AIC=-592.27
## y ~ gdp.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2
```

```r
model_4 <- step(lm(y~.,data = model.bic$varresult$pce.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1689.04
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + const
##
##
## Step:  AIC=1689.04
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2
##
##                       Df Sum of Sq    RSS    AIC
## - imgpgsc.train.ts.l2  1       781 238186 1687.8
## - unrate.train.ts.l1   1      1851 239256 1688.9
## <none>                             237406 1689.0
## - unrate.train.ts.l2   1      2232 239637 1689.3
## - imgpgsc.train.ts.l1  1      2424 239829 1689.5
## - gdp.train.ts.l2      1     15406 252811 1702.2
## - expgs.train.ts.l1    1     17902 255307 1704.6
## - pce.train.ts.l2      1     18348 255753 1705.0
## - expgs.train.ts.l2    1     22073 259479 1708.5
## - gdp.train.ts.l1      1     40318 277724 1725.0
## - pce.train.ts.l1      1     52545 289951 1735.4
##
## Step:  AIC=1687.83
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + unrate.train.ts.l2 + pce.train.ts.l2
##
##                       Df Sum of Sq    RSS    AIC
## <none>                             238186 1687.8
## - unrate.train.ts.l1   1      2746 240933 1688.6
## - unrate.train.ts.l2   1      3385 241572 1689.2
## - imgpgsc.train.ts.l1  1      7995 246181 1693.8
## - gdp.train.ts.l2      1     15815 254002 1701.4
## - pce.train.ts.l2      1     17610 255796 1703.1
```

```
## - expgs.train.ts.l1    1     22377 260563 1707.6
## - expgs.train.ts.l2    1     27458 265645 1712.2
## - gdp.train.ts.l1      1     42053 280240 1725.2
## - pce.train.ts.l1      1     53937 292124 1735.2
```

```
model_5 <- step(lm(y~.,data = model.bic$varresult$gdp.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1915.2
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + const
##
##
## Step:  AIC=1915.2
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     expgs.train.ts.l2 + imgpgsc.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2
##
##                       Df Sum of Sq     RSS    AIC
## - expgs.train.ts.l2    1        71  604519 1913.2
## - pce.train.ts.l2      1        99  604547 1913.2
## - pce.train.ts.l1      1       357  604805 1913.3
## - expgs.train.ts.l1    1       369  604817 1913.3
## - unrate.train.ts.l1   1      4367  608815 1914.9
## <none>                              604448 1915.2
## - unrate.train.ts.l2   1      5749  610196 1915.5
## - gdp.train.ts.l2      1     13089  617537 1918.4
## - imgpgsc.train.ts.l1  1     46924  651372 1931.3
## - imgpgsc.train.ts.l2  1     47616  652064 1931.5
## - gdp.train.ts.l1      1    569747 1174194 2073.9
##
## Step:  AIC=1913.23
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
##     imgpgsc.train.ts.l2 + unrate.train.ts.l2 + pce.train.ts.l2
##
##                       Df Sum of Sq     RSS    AIC
## - pce.train.ts.l2      1        73  604592 1911.3
## - pce.train.ts.l1      1       301  604820 1911.3
## - unrate.train.ts.l1   1      4301  608820 1912.9
## <none>                              604519 1913.2
## - unrate.train.ts.l2   1      5682  610201 1913.5
## - gdp.train.ts.l2      1     16782  621301 1917.8
## - expgs.train.ts.l1    1     19586  624104 1918.9
## - imgpgsc.train.ts.l1  1     53371  657889 1931.7
## - imgpgsc.train.ts.l2  1     54104  658622 1932.0
## - gdp.train.ts.l1      1    663227 1267746 2090.4
##
## Step:  AIC=1911.26
## y ~ gdp.train.ts.l1 + expgs.train.ts.l1 + imgpgsc.train.ts.l1 +
##     unrate.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l2 +
```

```
##     imgpgsc.train.ts.l2 + unrate.train.ts.l2
```

```r
model.bic.restrict <- restrict(model.bic)
```

Response

• In the equation for gdp, expgs, unrate and pce- lag 1 and 2 are statistically insignificant, which indicates that change in expgs, unrate and pce plausibly may not affect gdp. • In the equation for expgs, gdp, imgpgsc and unrate are statistically insignificant, which indicates that change in gdp, imgpgsc and unrate plausibly may not affect expgs. • In the equation for imgpgsc, only unrate and pce lag 1 and 2 are statistically insignificant, which indicates that change in unrate and pce plausibly may not affect imgpgsc. • In the equation for unrate, only expgs and pce lag 1 and 2 are statistically insignificant, which indicates that change in expgs and pce plausibly may affect unrate. • In the equation for pce, only imgpgsc and unrate lag 1 and 2 are statistically insignificant, which indicates that change in imgpgsc and unrate plausibly may not affect pce. The restricted model comparing to the original model, it does not have the pce lag 1 parameters in the equation for gdp, imgpgsc and unrate. This is reasonable, because, pce does not affect too much on these time series. This is also applicable for the unrate lag1 and lag2 coefficients as it does not affect too much on the other four time series.

Question 2c

Evaluate the goodness of fit for the restricted BIC model using the multivariate ARCH test, the Jarque-Bera test and the Portmanteau test. State which assumptions are satisfied, and which are violated. (Note: While we evaluate the residuals for the normality assumption, we do not necessarily assume normality of the data. We use the normality assumption if we use the t-test to evaluate statistical significance.)

```r
## ARCH, Residual Analysis: Constant Variance Assumption
arch.test(model.bic.restrict)
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object model.bic.restrict
## Chi-squared = 2238.6, df = 1125, p-value < 2.2e-16
```

```r
## J-B, Residual Analysis: Normality Assumption
normality.test(model.bic.restrict)
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object model.bic.restrict
## Chi-squared = 2521.4, df = 10, p-value < 2.2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object model.bic.restrict
## Chi-squared = 127.65, df = 5, p-value < 2.2e-16
```

```
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object model.bic.restrict
## Chi-squared = 2393.8, df = 5, p-value < 2.2e-16
```

```r
## Portmanteau, Residual Analysis: Uncorrelated Errors Assumption
serial.test(model.bic.restrict)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object model.bic.restrict
## Chi-squared = 575.88, df = 350, p-value = 2.864e-13
```

Response

- The Arch test shows that the residuals display heteroscedasticity. • The JB test shows the lack of normality in residuals. • The Portmanteau test shows that there exists still serial autocorrelation in residuals

Question 2d

Using the VAR model with the order selected using BIC, forecast the first two quarters of 2020 using the unrestricted and restricted VAR. Include 95% confidence intervals. Compare the predictions to the observed data. (You don't need to plot them (but can if you'd like). Using mean absolute percentage error and the precision measure, compare the predictions for GDP derived from the univariate analysis (Question 1) and this multivariate analysis. Discuss on the differences or similarities.

```r
pred_unres <- as.vector(predict(model.bic, n.ahead = 2))
pred_res <- as.vector(predict(model.bic.restrict, n.ahead = 2))
#MAPE for unrestricted VAR
preds.unres <- pred_unres$fcst$gdp.train.ts[, 1]
obs <- as.numeric(gdp.test.ts)
100 * mean(abs(preds.unres - obs) / obs)
```

```
## [1] 7.362181
```

```r
#MAPE for restricted VAR
preds.res <- pred_res$fcst$gdp.train.ts[, 1]
obs <- as.numeric(gdp.test.ts)
100 * mean(abs(preds.res - obs) / obs)
```

```
## [1] 7.435543
```

```r
#MAPE for ARIMA
preds.arima <- arima_pred1$pred
obs <- as.numeric(gdp.test.ts)
100 * mean(abs(preds.arima - obs) / obs)
```

```
## [1] 7.641779
```

```r
### Precision Measure (PM) for unrestricted VAR
sum((preds.unres - obs) ^ 2) / sum((obs - mean(obs)) ^ 2)
```

```
## [1] 3.283044
```

```r
### Precision Measure (PM) for restricted VAR
sum((preds.res - obs) ^ 2) / sum((obs - mean(obs)) ^ 2)
```

```
## [1] 3.345708
```

```r
### Precision Measure (PM) for ARIMA
sum((preds.arima - obs) ^ 2) / sum((obs - mean(obs)) ^ 2)
```

```
## [1] 3.46897
```

```r
# 95% Prediction Intervals and forcests
pred_unres$fcst$gdp.train.ts
```

```
##          fcst    lower    upper       CI
## [1,] 21840.60 21740.34 21940.86 100.2586
## [2,] 22019.65 21851.88 22187.42 167.7704
```

```r
pred_res$fcst$gdp.train.ts
```

```
##          fcst    lower    upper        CI
## [1,] 21845.76 21745.98 21945.54  99.77977
## [2,] 22043.55 21874.54 22212.56 169.01068
```

Response

The ARIMA has worse prediction results and the unrestricted and restrited VAR has similarly better results according to MAPE. Precision measure, however, gives us a different picture. All predictions seem to not capture the drastic change in the last two quarters.

Question 2e

Perform a Granger Causality analysis using Wald test to evaluate whether any of the economic indicators lead GDP. Would any of the indicators help in predicting or explaining GDP for next quarters? Provide your interpretation based on the Granger causality as well as for forecasting comparison in (2d).For this, use the unrestricted bic model from Question 2a.

```r
## Granger Causality: Wald Test
coef.gdp <- coefficients(model.bic)$gdp.train.ts[-(5 * 2 + 1), 1]
var.model <- vcov(model.bic)[2:11, 2:11]
wald.test(b = coef.gdp, var.model, Terms = seq(2, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 7.5, df = 2, P(> X2) = 0.023
```

```r
wald.test(b = coef.gdp, var.model, Terms = seq(3, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 18.4, df = 2, P(> X2) = 1e-04
```

```r
wald.test(b = coef.gdp, var.model, Terms = seq(4, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 2.6, df = 2, P(> X2) = 0.27
```

```r
wald.test(b = coef.gdp, var.model, Terms = seq(5, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 0.24, df = 2, P(> X2) = 0.89
```

Response

According to the wald test, only expgs and impgsc Granger cause gdp. However, unrate and pce do not Granger cause gdp. Since the restricted var dropped unrate and pce, the restricted var performs better than the unrestricted var.

Question 3

For this question, consider the training data to include the time values up to December 2017 and the testing data to include the first two quarters of 2018.

Question 3a

Apply the VAR modeling approach with the order selected using the BIC approach giving the unrestricted VAR model. Apply a model selection analysis using stepwise regression to select the models for each individual time series. Based on the selected models, form the restricted VAR model, much like what was presented in the Moose R example code. Compare these two models in terms of coefficients and their statistical significance with the models derived in Question 2.

Follow the below steps to implement this: 1)Analyze the coefficients of the unrestricted model. 2)Then treat each of the series separately and do a stepwise regression by apply lm() and step()(in the backward direction with 3 steps) functions to each of the component time series to examine the coefficients 3)The stepwise regressions will return three separate models. Analyze if the same coefficients are significant in the overall VAR versus each of the stepwise models. Discuss. 4)Then build a model using restrict and see which predictors were significant in the restricted VAR model.

```
expgs.test <- expgs.train[(length(expgs.train) - 7):(length(expgs.train) - 6)]
imgpgsc.test <- imgpgsc.train[(length(imgpgsc.train) - 7):(length(imgpgsc.train) - 6)]
unrate.test <- unrate.train[(length(unrate.train) - 7):(length(unrate.train) - 6)]
pce.test <- pce.train[(length(pce.train) - 7):(length(pce.train) - 6)]
gdp.test <- gdp.train[(length(gdp.train) - 7):(length(gdp.train) - 6)]


expgs.train <- expgs.train[0:(length(expgs.train) - 8)]
imgpgsc.train <- imgpgsc.train[0:(length(imgpgsc.train) - 8)]
unrate.train <- unrate.train[0:(length(unrate.train) - 8)]
pce.train <- pce.train[0:(length(pce.train) - 8)]
gdp.train <- gdp.train[0:(length(gdp.train) - 8)]


# expgs.train.ts <- ts(expgs.train, start = 1959, frequency = 4)
# imgpgsc.train.ts <- ts(imgpgsc.train, start = 1959, frequency = 4)
# unrate.train.ts <- ts(unrate.train, start = 1959, frequency = 4)
# pce.train.ts <- ts(pce.train, start = 1959, frequency = 4)
#
#
# expgs.test.ts <- ts(expgs.test, start = 2019, frequency = 4)
# imgpgsc.test.ts <- ts(imgpgsc.test, start = 2019, frequency = 4)
# unrate.test.ts <- ts(unrate.test, start = 2019, frequency = 4)
# pce.test.ts <- ts(pce.test, start = 2019, frequency = 4)
#
# train.ts <- cbind(expgs.train.ts,
#                   imgpgsc.train.ts,
#                   unrate.train.ts,
#                   pce.train.ts,
#                   gdp.train.ts)
# test.ts <- cbind(expgs.test.ts,
#                  imgpgsc.test.ts,
#                  unrate.test.ts,
#                  pce.test.ts,
#                  gdp.test.ts)


expgs.train.ts <- ts(expgs.train, start = 1959, frequency = 4)
imgpgsc.train.ts <- ts(imgpgsc.train, start = 1959, frequency = 4)
unrate.train.ts <- ts(unrate.train, start = 1959, frequency = 4)
pce.train.ts <- ts(pce.train, start = 1959, frequency = 4)
gdp.train.ts <- ts(gdp.train, start = 1959, frequency = 4)

expgs.test.ts <- ts(expgs.test, start = 2018, frequency = 4)
imgpgsc.test.ts <- ts(imgpgsc.test, start = 2018, frequency = 4)
unrate.test.ts <- ts(unrate.test, start = 2018, frequency = 4)
pce.test.ts <- ts(pce.test, start = 2018, frequency = 4)
gdp.test.ts <- ts(gdp.test, start = 2018, frequency = 4)
```

```r
train.ts <- cbind(expgs.train.ts,
                  imgpgsc.train.ts,
                  unrate.train.ts,
                  pce.train.ts,
                  gdp.train.ts)
test.ts <- cbind(expgs.test.ts,
                imgpgsc.test.ts,
                unrate.test.ts,
                pce.test.ts,
                gdp.test.ts)

vs <- VARselect(train.ts)
vs$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      5      4      2      5
```

```r
model.3a <- VAR(train.ts, p = 2)
model_1 <- step(lm(y~.,data = model.3a$varresult$expgs.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1508.12
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2 +
##     const
##
##
## Step:  AIC=1508.12
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2
##
##                        Df Sum of Sq    RSS    AIC
## - imgpgsc.train.ts.l2  1         5 134096 1506.1
## - imgpgsc.train.ts.l1  1        10 134101 1506.1
## - unrate.train.ts.l1   1       667 134758 1507.3
## - unrate.train.ts.l2   1       894 134984 1507.7
## <none>                             134091 1508.1
## - gdp.train.ts.l1      1      2126 136216 1509.8
## - gdp.train.ts.l2      1      2523 136614 1510.5
## - pce.train.ts.l1      1      4946 139037 1514.6
## - pce.train.ts.l2      1      9418 143508 1522.0
## - expgs.train.ts.l2    1     35783 169874 1561.5
## - expgs.train.ts.l1    1    231308 365399 1740.7
##
## Step:  AIC=1506.13
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + gdp.train.ts.l2
##
##                        Df Sum of Sq    RSS    AIC
```

```
## - imgpgsc.train.ts.l1  1       406 134502 1504.8
## - unrate.train.ts.l1   1       751 134846 1505.4
## - unrate.train.ts.l2   1      1021 135117 1505.9
## <none>                             134096 1506.1
## - gdp.train.ts.l1      1      2184 136279 1507.9
## - gdp.train.ts.l2      1      2550 136645 1508.5
## - pce.train.ts.l1      1      4954 139050 1512.6
## - pce.train.ts.l2      1      9579 143675 1520.3
## - expgs.train.ts.l2    1     39757 173852 1564.9
## - expgs.train.ts.l1    1    254721 388816 1753.2
##
## Step:  AIC=1504.84
## y ~ expgs.train.ts.l1 + unrate.train.ts.l1 + pce.train.ts.l1 +
##     gdp.train.ts.l1 + expgs.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + gdp.train.ts.l2
```

```
model_2 <- step(lm(y~.,data = model.3a$varresult$imgpgsc.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1491.6
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2 +
##     const
##
##
## Step:  AIC=1491.6
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq    RSS    AIC
## - unrate.train.ts.l2   1         3 124954 1489.6
## - unrate.train.ts.l1   1        24 124975 1489.7
## - pce.train.ts.l1      1       199 125150 1490.0
## - pce.train.ts.l2      1       207 125158 1490.0
## <none>                             124951 1491.6
## - expgs.train.ts.l1    1      1111 126061 1491.7
## - expgs.train.ts.l2    1      2276 127226 1493.8
## - gdp.train.ts.l1      1     10419 135370 1508.3
## - gdp.train.ts.l2      1     11894 136845 1510.9
## - imgpgsc.train.ts.l2  1     12048 136999 1511.1
## - imgpgsc.train.ts.l1  1    186590 311541 1703.4
##
## Step:  AIC=1489.61
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     pce.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq    RSS    AIC
## - unrate.train.ts.l1   1       171 125125 1487.9
## - pce.train.ts.l1      1       201 125155 1488.0
## - pce.train.ts.l2      1       206 125160 1488.0
## <none>                             124954 1489.6
```

```
## - expgs.train.ts.l1    1      1110 126064 1489.7
## - expgs.train.ts.l2    1      2298 127252 1491.9
## - gdp.train.ts.l1      1     10874 135828 1507.1
## - gdp.train.ts.l2      1     12369 137323 1509.7
## - imgpgsc.train.ts.l2  1     13320 138274 1511.3
## - imgpgsc.train.ts.l1  1    203193 328147 1713.5
##
## Step:  AIC=1487.93
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + pce.train.ts.l1 +
##     gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     pce.train.ts.l2 + gdp.train.ts.l2
```

```
model_3 <- step(lm(y~.,data = model.3a$varresult$unrate.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=-565.94
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2 +
##     const
##
##
## Step:  AIC=-565.94
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq    RSS     AIC
## - pce.train.ts.l1      1     0.003 18.971 -567.90
## - pce.train.ts.l2      1     0.018 18.986 -567.72
## - expgs.train.ts.l1    1     0.035 19.003 -567.51
## - expgs.train.ts.l2    1     0.051 19.019 -567.31
## <none>                             18.968 -565.94
## - gdp.train.ts.l2      1     0.532 19.500 -561.47
## - gdp.train.ts.l1      1     0.538 19.506 -561.40
## - imgpgsc.train.ts.l1  1     0.812 19.780 -558.14
## - imgpgsc.train.ts.l2  1     0.843 19.811 -557.76
## - unrate.train.ts.l2   1     2.585 21.553 -538.04
## - unrate.train.ts.l1   1    38.784 57.752 -307.40
##
## Step:  AIC=-567.9
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq    RSS     AIC
## - pce.train.ts.l2      1     0.042 19.013 -569.39
## - expgs.train.ts.l1    1     0.047 19.018 -569.32
## - expgs.train.ts.l2    1     0.066 19.037 -569.09
## <none>                             18.971 -567.90
## - gdp.train.ts.l2      1     0.530 19.501 -563.45
## - gdp.train.ts.l1      1     0.569 19.540 -562.99
## - imgpgsc.train.ts.l1  1     0.818 19.789 -560.03
## - imgpgsc.train.ts.l2  1     0.840 19.811 -559.76
```

```
## - unrate.train.ts.l2   1     2.582 21.553 -540.04
## - unrate.train.ts.l1   1    38.783 57.755 -309.39
##
## Step:  AIC=-569.39
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + gdp.train.ts.l2
```

```
model_4 <- step(lm(y~.,data = model.3a$varresult$pce.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1608.08
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2 +
##     const
##
##
## Step:  AIC=1608.08
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq    RSS    AIC
## - imgpgsc.train.ts.l2  1       733 206286 1606.9
## <none>                             205553 1608.1
## - unrate.train.ts.l1   1      1769 207322 1608.1
## - unrate.train.ts.l2   1      2931 208484 1609.4
## - imgpgsc.train.ts.l1  1      3048 208601 1609.5
## - gdp.train.ts.l2      1      7291 212844 1614.2
## - pce.train.ts.l2      1     15065 220618 1622.6
## - expgs.train.ts.l1    1     20187 225741 1628.0
## - expgs.train.ts.l2    1     24833 230387 1632.8
## - gdp.train.ts.l1      1     26780 232333 1634.7
## - pce.train.ts.l1      1     39853 245406 1647.5
##
## Step:  AIC=1606.92
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + unrate.train.ts.l2 +
##     pce.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq    RSS    AIC
## <none>                             206286 1606.9
## - unrate.train.ts.l1   1      2610 208896 1607.9
## - unrate.train.ts.l2   1      4224 210510 1609.7
## - gdp.train.ts.l2      1      7643 213929 1613.4
## - imgpgsc.train.ts.l1  1     12180 218465 1618.3
## - pce.train.ts.l2      1     14400 220686 1620.7
## - expgs.train.ts.l1    1     24769 231055 1631.5
## - gdp.train.ts.l1      1     28307 234593 1635.0
## - expgs.train.ts.l2    1     30352 236638 1637.0
## - pce.train.ts.l1      1     41014 247300 1647.3
```

```r
model_5 <- step(lm(y~.,data = model.3a$varresult$gdp.train.ts$model),
direction = "backward",steps = 3)
```

```
## Start:  AIC=1843.23
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2 +
##     const
##
##
## Step:  AIC=1843.23
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + pce.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq     RSS    AIC
## - pce.train.ts.l2      1         1  561492 1841.2
## - expgs.train.ts.l2    1       144  561635 1841.3
## - expgs.train.ts.l1    1       230  561721 1841.3
## - pce.train.ts.l1      1       244  561735 1841.3
## - unrate.train.ts.l1   1      4116  565606 1842.9
## <none>                              561491 1843.2
## - unrate.train.ts.l2   1      6701  568192 1844.0
## - gdp.train.ts.l2      1      6745  568236 1844.0
## - imgpgsc.train.ts.l2  1     48169  609660 1860.5
## - imgpgsc.train.ts.l1  1     50061  611552 1861.2
## - gdp.train.ts.l1      1    487762 1049253 1987.5
##
## Step:  AIC=1841.23
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + gdp.train.ts.l2
##
##                       Df Sum of Sq     RSS    AIC
## - expgs.train.ts.l2    1       158  561650 1839.3
## - expgs.train.ts.l1    1       235  561727 1839.3
## - pce.train.ts.l1      1      1642  563134 1839.9
## - unrate.train.ts.l1   1      4133  565624 1841.0
## <none>                              561492 1841.2
## - unrate.train.ts.l2   1      6718  568209 1842.0
## - gdp.train.ts.l2      1      7647  569138 1842.4
## - imgpgsc.train.ts.l2  1     49399  610891 1859.0
## - imgpgsc.train.ts.l1  1     51677  613168 1859.8
## - gdp.train.ts.l1      1    582711 1144202 2005.8
##
## Step:  AIC=1839.3
## y ~ expgs.train.ts.l1 + imgpgsc.train.ts.l1 + unrate.train.ts.l1 +
##     pce.train.ts.l1 + gdp.train.ts.l1 + imgpgsc.train.ts.l2 +
##     unrate.train.ts.l2 + gdp.train.ts.l2
```

```r
model.3a.res <- restrict(model.3a)
summary(model.3a.res)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: expgs.train.ts, imgpgsc.train.ts, unrate.train.ts, pce.train.ts, gdp.train.ts
## Deterministic variables: const
## Sample size: 234
## Log Likelihood: -4405.268
## Roots of the characteristic polynomial:
## 1.009 0.9748 0.9748 0.9469 0.7889 0.7889 0.4059 0.3839 0.2028 0.03664
## Call:
## VAR(y = train.ts, p = 2)
##
##
## Estimation results for equation expgs.train.ts:
## ===============================================
## expgs.train.ts = expgs.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + pce.tra
##
##                    Estimate Std. Error t value Pr(>|t|)
## expgs.train.ts.l1  1.55758    0.07387  21.084  < 2e-16 ***
## pce.train.ts.l1   -0.20418    0.05947  -3.433 0.000709 ***
## gdp.train.ts.l1    0.08929    0.04038   2.211 0.028020 *
## expgs.train.ts.l2 -0.61519    0.07403  -8.310 8.61e-15 ***
## pce.train.ts.l2    0.23428    0.05633   4.159 4.52e-05 ***
## gdp.train.ts.l2   -0.10158    0.04363  -2.328 0.020779 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 24.47 on 228 degrees of freedom
## Multiple R-Squared: 0.9995,  Adjusted R-squared: 0.9995
## F-statistic: 7.277e+04 on 6 and 228 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation imgpgsc.train.ts:
## =================================================
## imgpgsc.train.ts = imgpgsc.train.ts.l1 + gdp.train.ts.l1 + expgs.train.ts.l2 + imgpgsc.train.ts.l2 +
##
##                      Estimate Std. Error t value Pr(>|t|)
## imgpgsc.train.ts.l1  1.29709    0.06064  21.389  < 2e-16 ***
## gdp.train.ts.l1      0.21298    0.03281   6.492 5.24e-10 ***
## expgs.train.ts.l2   -0.04863    0.01391  -3.495 0.000569 ***
## imgpgsc.train.ts.l2 -0.35537    0.06056  -5.868 1.54e-08 ***
## pce.train.ts.l2      0.06359    0.01601   3.972 9.58e-05 ***
## gdp.train.ts.l2     -0.24090    0.03671  -6.562 3.52e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 23.59 on 228 degrees of freedom
## Multiple R-Squared: 0.9998,  Adjusted R-squared: 0.9998
## F-statistic: 1.678e+05 on 6 and 228 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation unrate.train.ts:
```

```
## ===================================================
## unrate.train.ts = imgpgsc.train.ts.l1 + unrate.train.ts.l1 + gdp.train.ts.l1 + imgpgsc.train.ts.l2 +
##
##                       Estimate Std. Error t value Pr(>|t|)
## imgpgsc.train.ts.l1 -0.0023005  0.0007758  -2.965  0.00335 **
## unrate.train.ts.l1   1.3249042  0.0594490  22.286  < 2e-16 ***
## gdp.train.ts.l1     -0.0009184  0.0003942  -2.330  0.02070 *
## imgpgsc.train.ts.l2  0.0022820  0.0007780   2.933  0.00370 **
## unrate.train.ts.l2  -0.3590535  0.0598584  -5.998 7.79e-09 ***
## gdp.train.ts.l2      0.0009301  0.0003964   2.346  0.01982 *
## const                0.2521695  0.0859909   2.933  0.00371 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.2903 on 227 degrees of freedom
## Multiple R-Squared: 0.9979,  Adjusted R-squared: 0.9978
## F-statistic: 1.539e+04 on 7 and 227 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation pce.train.ts:
## ================================================
## pce.train.ts = expgs.train.ts.l1 + imgpgsc.train.ts.l1 + pce.train.ts.l1 + gdp.train.ts.l1 + expgs.tr
##
##                      Estimate Std. Error t value Pr(>|t|)
## expgs.train.ts.l1     0.47685    0.09317   5.118 6.61e-07 ***
## imgpgsc.train.ts.l1   0.08964    0.02310   3.881 0.000137 ***
## pce.train.ts.l1       0.54379    0.08093   6.719 1.48e-10 ***
## gdp.train.ts.l1       0.30616    0.05079   6.028 6.75e-09 ***
## expgs.train.ts.l2    -0.52284    0.09309  -5.616 5.72e-08 ***
## unrate.train.ts.l2    3.16566    1.57101   2.015 0.045089 *
## pce.train.ts.l2       0.26863    0.07083   3.793 0.000191 ***
## gdp.train.ts.l2      -0.18273    0.05711  -3.200 0.001574 **
## const               -38.59127   11.70446  -3.297 0.001135 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 30.47 on 225 degrees of freedom
## Multiple R-Squared:     1,   Adjusted R-squared:     1
## F-statistic: 1.029e+06 on 9 and 225 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation gdp.train.ts:
## ================================================
## gdp.train.ts = expgs.train.ts.l1 + imgpgsc.train.ts.l1 + gdp.train.ts.l1 + imgpgsc.train.ts.l2 + gdp
##
##                     Estimate Std. Error t value Pr(>|t|)
## expgs.train.ts.l1   -0.09800    0.02297  -4.266 2.92e-05 ***
## imgpgsc.train.ts.l1  0.69062    0.12881   5.361 2.01e-07 ***
## gdp.train.ts.l1      1.25095    0.06583  19.003  < 2e-16 ***
## imgpgsc.train.ts.l2 -0.72016    0.12878  -5.592 6.36e-08 ***
## gdp.train.ts.l2     -0.22777    0.06697  -3.401 0.000792 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 50.15 on 229 degrees of freedom
## Multiple R-Squared:     1,   Adjusted R-squared:     1
## F-statistic: 1.53e+06 on 5 and 229 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                  expgs.train.ts imgpgsc.train.ts unrate.train.ts pce.train.ts
## expgs.train.ts         612.2933          242.789        -0.55152     362.2521
## imgpgsc.train.ts       242.7887          569.111        -1.41164     268.2858
## unrate.train.ts         -0.5515           -1.412         0.08581      -0.9316
## pce.train.ts           362.2521          268.286        -0.93163     936.7535
## gdp.train.ts           816.6281          443.244        -2.00431     935.6184
##                  gdp.train.ts
## expgs.train.ts        816.628
## imgpgsc.train.ts      443.244
## unrate.train.ts        -2.004
## pce.train.ts          935.618
## gdp.train.ts         2578.030
##
## Correlation matrix of residuals:
##                  expgs.train.ts imgpgsc.train.ts unrate.train.ts pce.train.ts
## expgs.train.ts          1.00000           0.4113        -0.07609       0.4783
## imgpgsc.train.ts        0.41129           1.0000        -0.20200       0.3674
## unrate.train.ts        -0.07609          -0.2020         1.00000      -0.1039
## pce.train.ts            0.47832           0.3674        -0.10391       1.0000
## gdp.train.ts            0.64998           0.3659        -0.13476       0.6021
##                  gdp.train.ts
## expgs.train.ts         0.6500
## imgpgsc.train.ts       0.3659
## unrate.train.ts       -0.1348
## pce.train.ts           0.6021
## gdp.train.ts           1.0000
```

Response The variable 'pce' has a stronger effect on the other variables than in the models in Question 2. Most variables of pce are statistically significant.However, unrate still remains insignificant in the other four variable models.

Question 3b

Forecast the first two quarters of 2019 using the unrestricted and restricted VAR models derived in (3a). Include 95% confidence intervals. Compare the predictions to the observed data using mean absolute percentage error and the precision measure for GDP. Compare the predictions to those derived in (2d). Comment on the accuracy of the predictions.

```r
pred_unres.3a <- as.vector(predict(model.3a, n.ahead = 2))
pred_res.3a <- as.vector(predict(model.3a.res, n.ahead = 2))

#MAPE for unrestricted VAR
preds.res <- pred_res.3a$fcst$gdp.train.ts[, 1]
```

```
obs <- as.numeric(gdp.test.ts)
100 * mean(abs(preds.res - obs) / obs)
```

```
## [1] 0.3129305
```

```
#MAPE for restricted VAR
preds.unres <- pred_unres.3a$fcst$gdp.train.ts[, 1]
obs <- as.numeric(gdp.test.ts)
100 * mean(abs(preds.unres - obs) / obs)
```

```
## [1] 0.4655116
```

```
pred_res.3a$fcst$gdp.train.ts
```

```
##          fcst    lower    upper        CI
## [1,] 20137.19 20038.90 20235.47  98.28408
## [2,] 20370.88 20204.09 20537.67 166.79145
```

```
pred_unres.3a$fcst$gdp.train.ts
```

```
##          fcst    lower    upper        CI
## [1,] 20114.32 20015.97 20212.67  98.34833
## [2,] 20331.61 20167.91 20495.31 163.69774
```

Response

They are both more accurate than the predictions from the models in Question 2. The restricted model is less accurate than the unrestricted one.

Question 3c

Perform a Granger Causality analysis using Wald test to evaluate whether any of the economic indicators lead GDP. Would any of the indicators help in predicting or explaining GDP for next quarters? Provide your interpretation based on the Granger causality as well as for forecasting comparison in (3b). Compare this analysis with the findings in (2e). For this question, use the unrestricted VAR model from Question 3a.

```
## Granger Causality: Wald Test
coef.gdp.3c <- coefficients(model.3a)$gdp.train.ts[-(5 * 2 + 1), 1]
var.model.3c <- vcov(model.3a)[2:11, 2:11]
wald.test(b = coef.gdp.3c, var.model.3c, Terms = seq(2, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 83.6, df = 2, P(> X2) = 0.0
```

```r
wald.test(b = coef.gdp.3c, var.model.3c, Terms = seq(3, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 17.5, df = 2, P(> X2) = 0.00016
```

```r
wald.test(b = coef.gdp.3c, var.model.3c, Terms = seq(4, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 2.7, df = 2, P(> X2) = 0.26
```

```r
wald.test(b = coef.gdp.3c, var.model.3c, Terms = seq(5, 5 * 2, 5))
```

```
## Wald test:
## ----------
##
## Chi-squared test:
## X2 = 5705.0, df = 2, P(> X2) = 0.0
```

Response

According to the Wald test, expgs, imgpgsc and pce Granger cause gdp. In 2e, pce does not. Whereas now, pce Granger causes gdp, and unemployment rate continues to not Granger cause GDP.

Question 4: Reflection

From what you encountered above and your conceptual understanding of VAR modelling, reflect on the relative strengths and weaknesses of the modelling approach. Particularly, you will need to put this analysis into the perspective of the results you found and any relevant economic events you might be potentially able to link them to.

Response

(Any logical and reasonable thoughts are good to go.)