Vector Autoregressive Models

Time Series Final | 'Flavor 2'

Marko Jurkovich & Matt Zacharski

7 June 2021

Abstract

This is our abstract, blah blah blah

Introduction

The vector autoregressive model (VAR) is a time series model that is typically used for the analysis of multivariate time series. Consequently, VAR models are particularly useful in fields where several time series may affect one another. For example, a researcher may be interested in exploring the relationship between several foreign exchange rates overtime or the trends of two countries' GDPs. While the fields of economics and finance may be initial fields that come to mind when discussing VAR models, the natural sciences also use VAR models to analysis and forecast things such as rainfall while also considering humidity, temperature, and other factors.

Like the name suggests, VAR models are an extension of the univariate autoregressive model. Similar to the univariate autoregressive model, each variable's equation includes its own lagged values. However, the VAR model goes a step further and also includes lagged values of other time series in each variable's equation. Including this extra information allows VAR models to often make more accurate predictions than the general AR model, justifying their existence and use. It is worth noting that several related models have been developed over the past few decades. Namely, structural vector autoregressive models, vector error correction models, and structural vector error correction models. While these models are outside the scope of this paper, it is useful to know they attempt to improve the VAR model by accounting for concepts such as cointegration or some type of shock.

In this paper, we will discuss the form, estimation, diagnostics, and forecasting process of VAR models. To highlight these methods, we have included a brief analysis of three cryptocurrency time series. Additionally, this paper will only deal with the selection and forecasting of stationary VAR models.

Model

(marko)

The basic structure of a VAR model is quite similar to that of an AR model, however there are two fundamental differences. The first is that there are multiple equations, one for each item in the vector of time series considered. Secondly, rather than have a time point x_t of a given time series being regressed on its own lags, it is also regressed on the lags of the other time series. The order of the model (number of lags) is typically indicated by the p in VAR(p).

There are three main forms of VAR models: reduced form, recursive, and structural. The reduced form defines each variable in the vector of time series as the function of its own lags and the lags of all other variables, with an error term. The error terms from each equation can theoretically be correlated, *but only within the same time period.* Correlation across equations across different time periods would imply autocorrelated errors within a single time series, which is not an assumption of the model.

Recursive VAR models are constructed so that the error terms for a variable are uncorrelated to the prior variables' errors. This is done by adding the preceding variables' current values to the typical VAR equation.

The structural VAR models are distinguished in that they make assumptions of the "causal structure" of the the data allow shocks to be indentified. These shocks would otherwise be incorporated into error terms in recursive and reduced models.

A reduced VAR(1) model with three variables can be modeled as such:

$$\begin{aligned} x_{t,1} &= \alpha_1 + \beta_{1,1}x_{t-1,1} + \beta_{1,2}x_{t-1,2} + \beta_{1,3}x_{t-1,3} + \epsilon_{t,1} \\ x_{t,2} &= \alpha_2 + \beta_{2,1}x_{t-1,1} + \beta_{2,2}x_{t-1,2} + \beta_{2,3}x_{t-1,3} + \epsilon_{t,2} \\ x_{t,1} &= \alpha_3 + \beta_{3,1}x_{t-1,1} + \beta_{3,2}x_{t-1,2} + \beta_{3,3}x_{t-1,3} + \epsilon_{t,3} \end{aligned} \tag{1}$$

We can generalize the previous equations for a VAR(p) model with k variables using matrix algebra:

$$\begin{bmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{k,t} \end{bmatrix} = \begin{bmatrix} \alpha_1^1 & \alpha_1^2 \\ \alpha_2 & \alpha_2^2 \\ \vdots & \vdots \\ \alpha_k^1 & \alpha_k^2 \end{bmatrix} \begin{bmatrix} 1 \\ t \end{bmatrix} + \begin{bmatrix} \beta_{1,1}^1 & \cdots & \beta_{1,k}^1 \\ \beta_{2,1}^1 & \cdots & \beta_{2,k}^1 \\ \vdots & \ddots & \vdots \\ \beta_{k,1}^1 & \cdots & \beta_{k,k}^1 \end{bmatrix} \begin{bmatrix} x_{t-1,1} \\ x_{t-1,2} \\ \vdots \\ x_{t-1,k} \end{bmatrix} + \cdots + \begin{bmatrix} \beta_{1,1}^p & \cdots & \beta_{1,k}^p \\ \beta_{2,1}^p & \cdots & \beta_{2,k}^p \\ \vdots & \ddots & \vdots \\ \beta_{k,1}^p & \cdots & \beta_{k,k}^p \end{bmatrix} \begin{bmatrix} y_{t-p,1} \\ y_{t-p,2} \\ \vdots \\ y_{t-p,k} \end{bmatrix} + \begin{bmatrix} \epsilon_{t,1} \\ \epsilon_{t,2} \\ \vdots \\ \epsilon_{t,3} \end{bmatrix} \quad (2)$$

or, abstracting the matrices

$$\mathbf{x_t} = \mathbf{Au} + \mathbf{B^1 x_{t-1}} + ... + \mathbf{B^p x_{t-p}} + \epsilon_\mathbf{t} \quad (3)$$

where $\mathbf{x_t}$ is an (k x 1) vector of time series variables, and $\mathbf{x_{t-i}}$ are the vectors of the time series variables at various lags determined by p. $\mathbf{A}$ is the coefficient matrix on the constant and trend term (both optional) represented in $\mathbf{u}$. $\mathbf{B^i x_{t-1}}$ are the (k x k) coefficient matrices; there are p of these matrices dependent on the order of the VAR(p) model. $\epsilon_\mathbf{t}$ is the vector of error terms.

Estimation

(marko)

VAR models fall into the category of Seemingly Unrealted Regression (SUR) models, which, without diving into too much detail, allows the combination multivariate systems into a conceptially simpler notation. The VAR case of SUR models in one in which all the explanatory variables (but not the coefficients, importantly) are indentical across equations. While SUR models often use Generalized Least Squares to estimate the equations, in the case of VAR models OLS is theoretically just as efficient at estimating each equation individually.

An extremely important consideration before estimating VAR models however, is parameter selection and extent. As both variables and lags are added to VAR models, the number of coefficients increases quadratically. For example, a "nine-variable, four-lag VAR has 333 unknown coefficients" (Stock 110).

- interpretation of coeffs
- asymptotically valid t-tests on individual coefficients may be constructed in the usual way

Diagnostics and Lag Length Selection

The diagnostic testing for VAR models is similar to the diagnostic testing for AR models except with a few extra tests to consider. **Idk if we want to include information criteria in the diagnostic section or not but it seems like something worth considering**

However, first we will go over the lag length selection process. The question of how many lag terms to include in a VAR model is an important because it plays a substantial role in the quality of forecasts the model will make. The common approach to select this number is to fit VAR(p) models with lag lengths from $p = 0,\ldots,p_{max}$ and choose the p with that minimizes the considered information criteria. Typically, Akaike, Bayesian, Hannan-Quinn, and forecast prediction error information criterion are considered. Model selection criteria for VAR(p) models has the form

$$IC(p) = ln|\tilde{\sum}(p)| + c_T * \varphi(n,p) \quad (4)$$

where $\tilde{\sum}(p) = T^-1 \sum_{t=1}^{T} \hat{\epsilon}_t \hat{\epsilon}_t'$ represents the residual covariance matrix without a degrees of freedom correction, c_T is a sequence indexed by the sample size T, and $\varphi(n,p)$ is a penalty function to penalize large

VAR(p) models.*neccessary to write out the individual equations for each criteria?? idk*. Each criterion has its benefits and drawbacks. However, these are not the focus on this paper but it is generally best to consider several criteria when determining the optimal lag length selection.

When determining the quality of the fit of a certain VAR model, it is useful to consider the ACF and PACFs of the residuals. Similar to evaluating an AR model, we are looking for ACF and PACF plots that show no Considering autocorrealtion within the residuals. Unlike with the univariate AR models, we must also consider cross correaltion plots. Considering the ACF plots of the residuals for each set of residuals of all of the variables in a VAR model is important in order to obsere any cross-correlations between variables indicating that your model might not be capturing the relationship between those variables and time sufficiently.

To test for heteroscedasticity within errors it is common to use an ARCH-LM test. However, this test relies on a complicated regression that is outside the scope of this paper. Similarly, the technical details of the normality test are also outside the scope of this paper but the gist of it is that researchers should use Jarque-Bera normality tests to test for normality of residuals of either each individual equation or the normality of all the equations' residuals using a variance-covariance matrix for the centered residuals.

It is also wise to test for autocorrelation within the residuals of VAR(p) models to avoid standard errors being too small or T-statistics that are too large. Do to this within a VAR(p) setting we can conduct a Portmanteau test of the form

$$Q_h = T \sum_{j=1}^{h} tr(\hat{C}_j' \hat{C}_0^{-1} \hat{C}_j \hat{C}_0^{-1}) \tag{5}$$

where $\hat{C}_i = \frac{1}{T} \sum_{t=i+1}^{T} \mathbf{\hat{u}_t \hat{u}'_{t-i}}^{\top}$ and the test statistics Q_h has an approximate $\chi^2(K^2(h - n^*))$ distribution. n^* refers to the number of non-deterministic terms in the specified VAR(p) model.

stability test

granger test for causality? not going to include this because it seems a little outside the scope...

Forecasting

(marko)

Forecasting in VAR is somewhat straightforward and similar to typical AR models. A one-step ahead forecast made at time *t+1* can be expressed as:

$$\mathbf{x_{t+1|t}} = \mathbf{Au} + \mathbf{B^1 x_t} + ... + \mathbf{B^p x_{t-p+1}} \tag{6}$$

Predicting more than one step ahead is done recursively in what can be referred to as the chain-rule of forecasting:

$$\mathbf{x_{t+1|t}} = \mathbf{Au} + \mathbf{B^1 x_t} + ... + \mathbf{B^p x_{t-p+1}} \tag{7}$$

Data Example

(matt)

1. clean code...
2. somehow highlight the modeling process in this paper? idk ....

Discussion

References

Pfaff, Bernhard. VAR, SVAR and SVEC Models: Implementation Within R Package vars. Journal of Statistical Software 27(4), 2008. https://cran.r-project.org/web/packages/vars/vignettes/vars.pdf.

PennState Eberly College of Science. Vector autoregressive models VAR(p) models.

https://online.stat.psu.edu/stat510/lesson/11/11.2.

Shumway, Robert H., and David S. Stoffer. Time Series Analysis and Its Applications. Springer Texts in Statistics, Springer International Publishing, 2017, pp. 273–279. https://link.springer.com/content/pdf/10.1007%2F978-3-319-52452-8.pdf.

Stock, James H., and Mark W. Watson. "Vector Autoregressions." Journal of Economic Perspectives, vol. 15, no. 4, American Economic Association, Nov. 2001, pp. 101–15. Crossref, doi:10.1257/jep.15.4.101.

Code Appendix

```
# reading in data

btc <- read_csv("BTC-USD.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   `Adj Close` = col_double(),
##   Volume = col_double()
## )
```

```
## Warning: 24 parsing failures.
##  row       col expected actual          file
## 1423 Open      a double   null 'BTC-USD.csv'
## 1423 High      a double   null 'BTC-USD.csv'
## 1423 Low       a double   null 'BTC-USD.csv'
## 1423 Close     a double   null 'BTC-USD.csv'
## 1423 Adj Close a double   null 'BTC-USD.csv'
## .... ......... ........ ...... ............
## See problems(...) for more details.
```

```
eth <- read_csv("ETH-USD.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
```

```
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   `Adj Close` = col_double(),
##   Volume = col_double()
## )
```

```
## Warning: 24 parsing failures.
##  row       col expected actual          file
## 1423 Open      a double   null 'ETH-USD.csv'
## 1423 High      a double   null 'ETH-USD.csv'
## 1423 Low       a double   null 'ETH-USD.csv'
## 1423 Close     a double   null 'ETH-USD.csv'
## 1423 Adj Close a double   null 'ETH-USD.csv'
## .... ......... ........ ...... ............
## See problems(...) for more details.
```

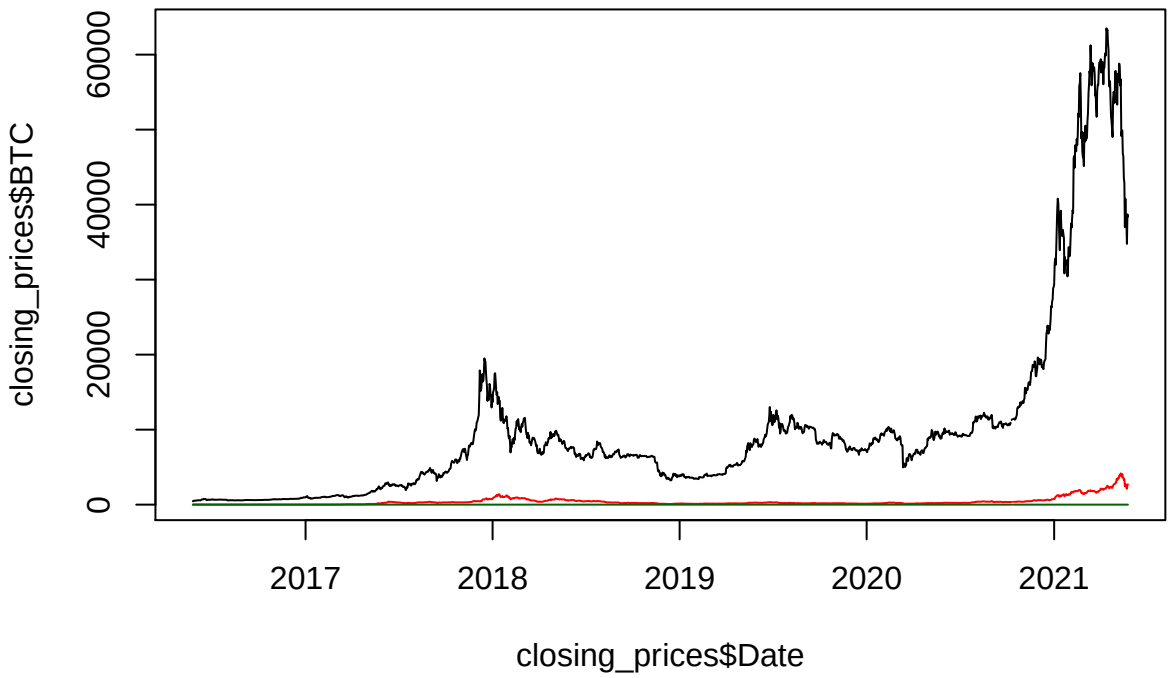
```
doge <- read_csv("DOGE-USD.csv")
```

```
## Parsed with column specification:
## cols(
##   Date = col_date(format = ""),
##   Open = col_double(),
##   High = col_double(),
##   Low = col_double(),
##   Close = col_double(),
##   `Adj Close` = col_double(),
##   Volume = col_double()
## )
```

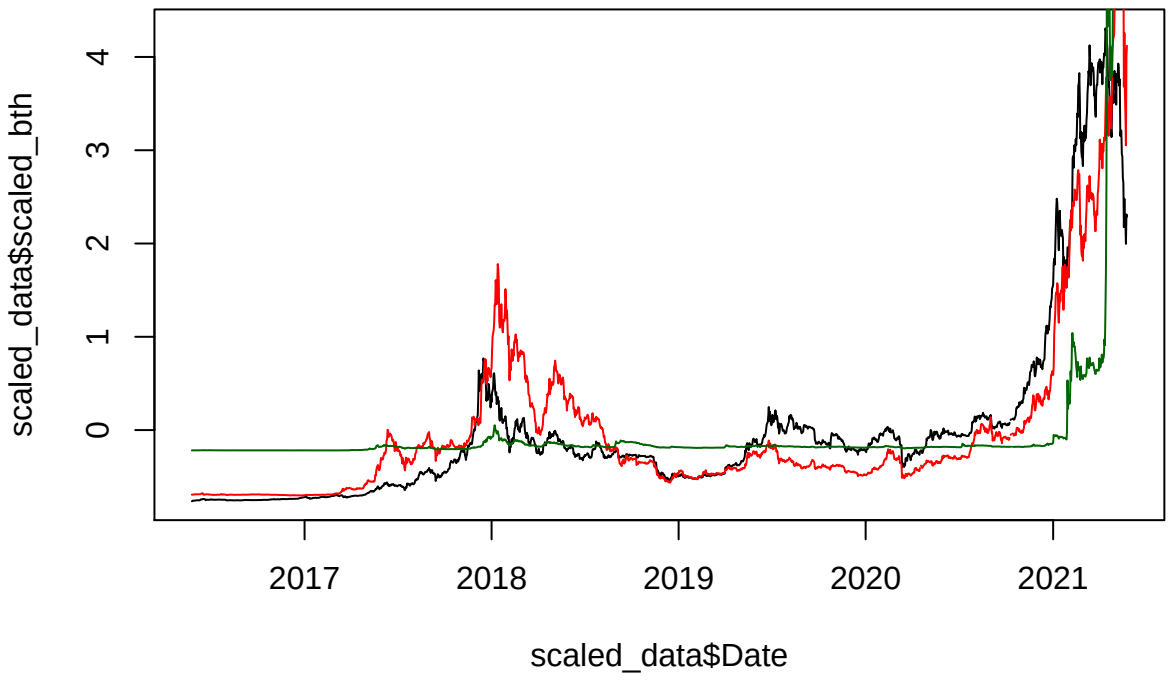
```
## Warning: 24 parsing failures.
##  row       col expected actual          file
## 1423 Open      a double   null 'DOGE-USD.csv'
## 1423 High      a double   null 'DOGE-USD.csv'
## 1423 Low       a double   null 'DOGE-USD.csv'
## 1423 Close     a double   null 'DOGE-USD.csv'
## 1423 Adj Close a double   null 'DOGE-USD.csv'
## .... ......... ........ ...... ............
## See problems(...) for more details.
```

```
eth <- eth[-1827, ] # matching length
```

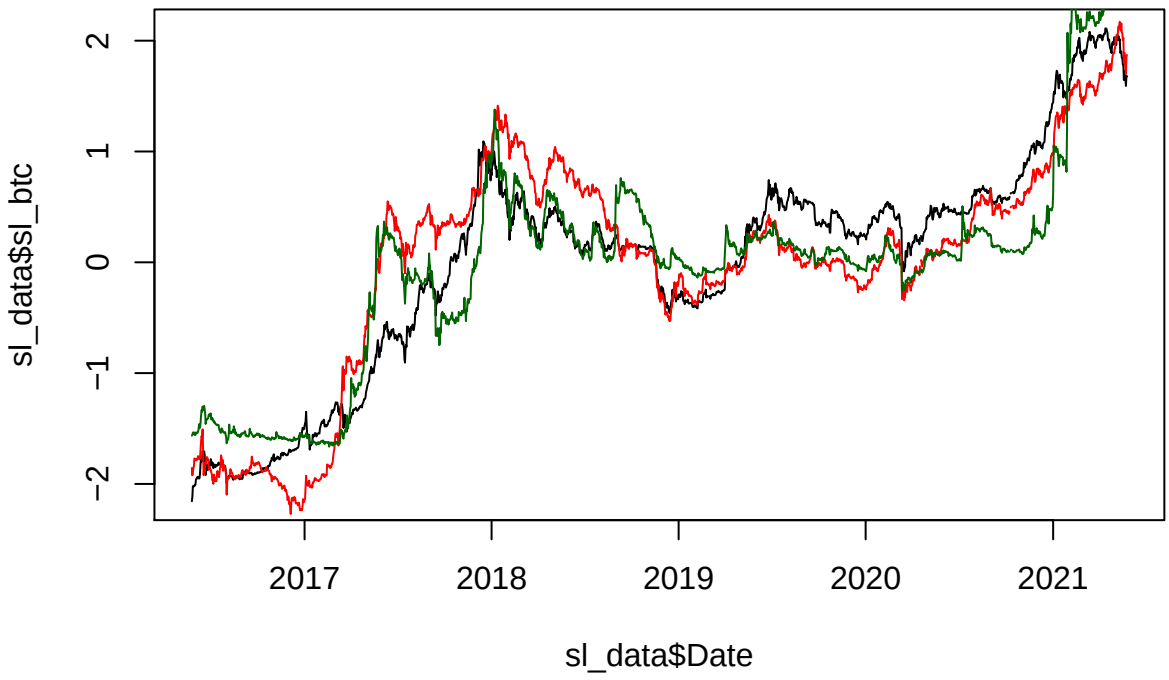
```
# creating closing price dataframe

closing_prices <- as.data.frame(cbind(btc$Close, eth$Close, doge$Close))

closing_prices$Date <- btc$Date

colnames(closing_prices) <- c("BTC", "ETH", "DOGE", "Date")

plot(closing_prices$Date, closing_prices$BTC, type = "l")
lines(closing_prices$Date, closing_prices$ETH, type = "l", col = "red")
lines(closing_prices$Date, closing_prices$DOGE, type = "l", col = "dark green")
```



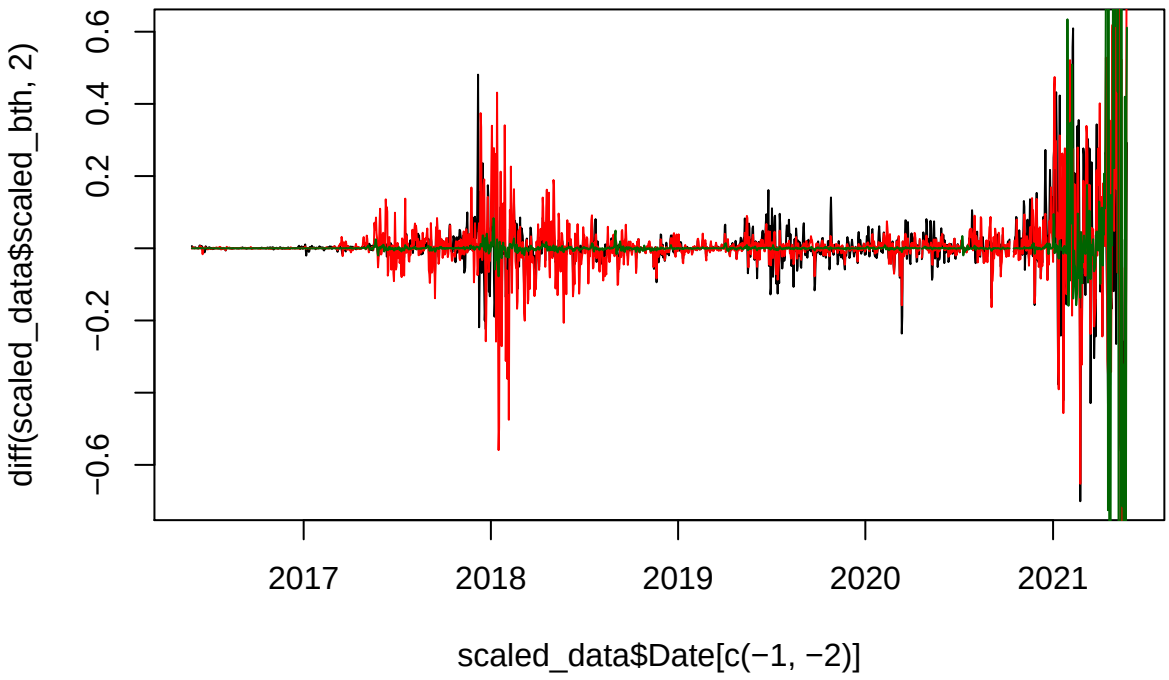

```
# creating scaled data dataframe

btc_scale <- scale(closing_prices$BTC)
doge_scale <- scale(closing_prices$DOGE)
eth_scale <- scale(closing_prices$ETH)

scaled_data <- data.frame(
  Date = closing_prices$Date,
  scaled_bth = btc_scale[, 1],
  scaled_eth = eth_scale[, 1],
  scaled_doge = doge_scale[, 1]
)

plot(scaled_data$Date, scaled_data$scaled_bth, type = "l")
lines(scaled_data$Date, scaled_data$scaled_eth, type = "l", col = "red")
lines(scaled_data$Date, scaled_data$scaled_doge, type = "l", col = "dark green")
```




```
# creating scaled log dataframe
# resolving NAs
# na2lag1(c(closing_prices$BTC,closing_prices$DOGE,closing_prices$ETH))

Date <- closing_prices$Date
sl_btc <- scale(log((closing_prices$BTC)))[, 1]
sl_doge <- scale(log(as.numeric(closing_prices$DOGE)))[, 1]
sl_eth <- scale(log(as.numeric(closing_prices$ETH)))[, 1]
sl_data <- data.frame(Date, sl_btc, sl_doge, sl_eth)

plot(sl_data$Date, sl_data$sl_btc, type = "l")
lines(sl_data$Date, sl_data$sl_eth, type = "l", col = "red")
lines(sl_data$Date, sl_data$sl_doge, type = "l", col = "dark green")
```



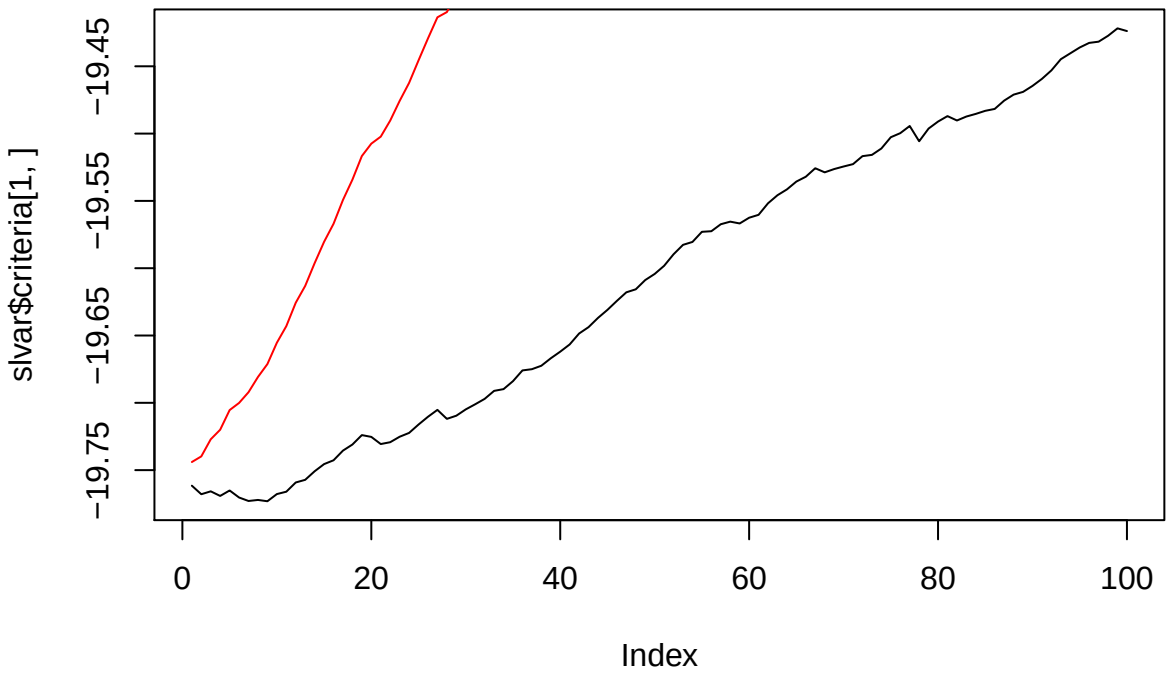

```
# differenced scaled (unlogged) data plot
plot(scaled_data$Date[c(-1, -2)], diff(scaled_data$scaled_bth, 2), type = "l")
lines(scaled_data$Date[c(-1, -2)], diff(scaled_data$scaled_eth, 2), type = "l", col = "red")
lines(scaled_data$Date[c(-1, -2)], diff(scaled_data$scaled_doge, 2), type = "l", col = "dark green")
```




```
# varselect for scaled & logged data

slvar <- VARselect(na.omit(sl_data[,-1]), lag.max = 100, type = "both")
slvar$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      9      1      1      9
```

```
plot(slvar$criteria[1,], type = 'l')
lines(slvar$criteria[2,], type = 'l', col = "red")
```



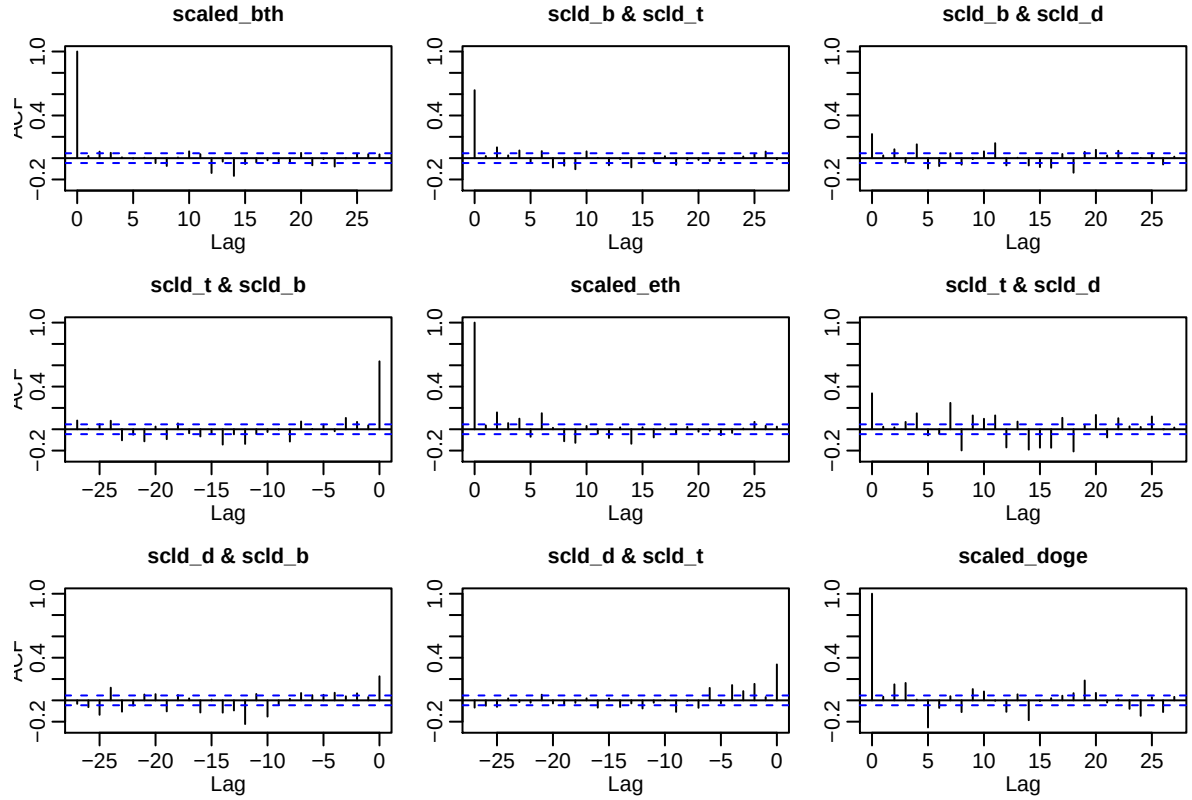

```
# prepping data for vars package
scaled_data_clean <- scaled_data[, -1] %>% na.omit()
og_data_clean <- na.omit(closing_prices)
og_data_clean <- og_data_clean[, 4]

# fitting var model
fitvar <- VAR(scaled_data_clean, p = 2, type = "both")

summary(fitvar)
```

```
##
## VAR Estimation Results:
## ========================
## Endogenous variables: scaled_bth, scaled_eth, scaled_doge
## Deterministic variables: both
## Sample size: 1820
## Log Likelihood: 5919.566
## Roots of the characteristic polynomial:
## 0.9964 0.9964 0.9671 0.1379 0.1379 0.05064
## Call:
## VAR(y = scaled_data_clean, p = 2, type = "both")
##
##
## Estimation results for equation scaled_bth:
## ===========================================
## scaled_bth = scaled_bth.l1 + scaled_eth.l1 + scaled_doge.l1 + scaled_bth.l2 + scaled_eth.l2 + scaled
```

```
##
##                   Estimate Std. Error t value Pr(>|t|)
## scaled_bth.l1   1.082e+00  3.032e-02  35.675  < 2e-16 ***
## scaled_eth.l1  -1.757e-01  2.068e-02  -8.499  < 2e-16 ***
## scaled_doge.l1 -2.315e-03  8.160e-03  -0.284   0.7767
## scaled_bth.l2  -6.694e-02  3.051e-02  -2.194   0.0283 *
## scaled_eth.l2   1.657e-01  2.055e-02   8.065 1.32e-15 ***
## scaled_doge.l2 -5.012e-03  8.246e-03  -0.608   0.5434
## const           3.502e-03  3.450e-03   1.015   0.3102
## trend          -1.704e-06  3.503e-06  -0.486   0.6267
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.05506 on 1812 degrees of freedom
## Multiple R-Squared: 0.997,   Adjusted R-squared: 0.997
## F-statistic: 8.549e+04 on 7 and 1812 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation scaled_eth:
## ===========================================
## scaled_eth = scaled_bth.l1 + scaled_eth.l1 + scaled_doge.l1 + scaled_bth.l2 + scaled_eth.l2 + scaled
##
##                   Estimate Std. Error t value Pr(>|t|)
## scaled_bth.l1   1.907e-01  4.609e-02   4.137 3.68e-05 ***
## scaled_eth.l1   7.024e-01  3.143e-02  22.351  < 2e-16 ***
## scaled_doge.l1  1.454e-02  1.240e-02   1.172  0.24126
## scaled_bth.l2  -1.512e-01  4.637e-02  -3.261  0.00113 **
## scaled_eth.l2   2.617e-01  3.123e-02   8.379  < 2e-16 ***
## scaled_doge.l2 -5.748e-03  1.253e-02  -0.459  0.64654
## const           1.633e-02  5.244e-03   3.114  0.00188 **
## trend          -1.457e-05  5.324e-06  -2.736  0.00628 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.08369 on 1812 degrees of freedom
## Multiple R-Squared: 0.993,   Adjusted R-squared: 0.993
## F-statistic: 3.686e+04 on 7 and 1812 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation scaled_doge:
## ============================================
## scaled_doge = scaled_bth.l1 + scaled_eth.l1 + scaled_doge.l1 + scaled_bth.l2 + scaled_eth.l2 + scaled
##
##                   Estimate Std. Error t value Pr(>|t|)
## scaled_bth.l1  -2.497e-01  9.083e-02  -2.749 0.006038 **
## scaled_eth.l1  -2.083e-01  6.193e-02  -3.363 0.000786 ***
## scaled_doge.l1  8.762e-01  2.444e-02  35.852  < 2e-16 ***
## scaled_bth.l2   2.872e-01  9.137e-02   3.143 0.001699 **
## scaled_eth.l2   2.088e-01  6.154e-02   3.392 0.000709 ***
## scaled_doge.l2  9.866e-02  2.470e-02   3.995 6.74e-05 ***
## const           2.294e-02  1.033e-02   2.220 0.026519 *
## trend          -2.038e-05  1.049e-05  -1.943 0.052188 .
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.1649 on 1812 degrees of freedom
## Multiple R-Squared: 0.9729,  Adjusted R-squared: 0.9728
## F-statistic:  9305 on 7 and 1812 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##             scaled_bth scaled_eth scaled_doge
## scaled_bth    0.003032   0.002936    0.002050
## scaled_eth    0.002936   0.007004    0.004646
## scaled_doge   0.002050   0.004646    0.027199
##
## Correlation matrix of residuals:
##             scaled_bth scaled_eth scaled_doge
## scaled_bth      1.0000     0.6370      0.2257
## scaled_eth      0.6370     1.0000      0.3366
## scaled_doge     0.2257     0.3366      1.0000
```

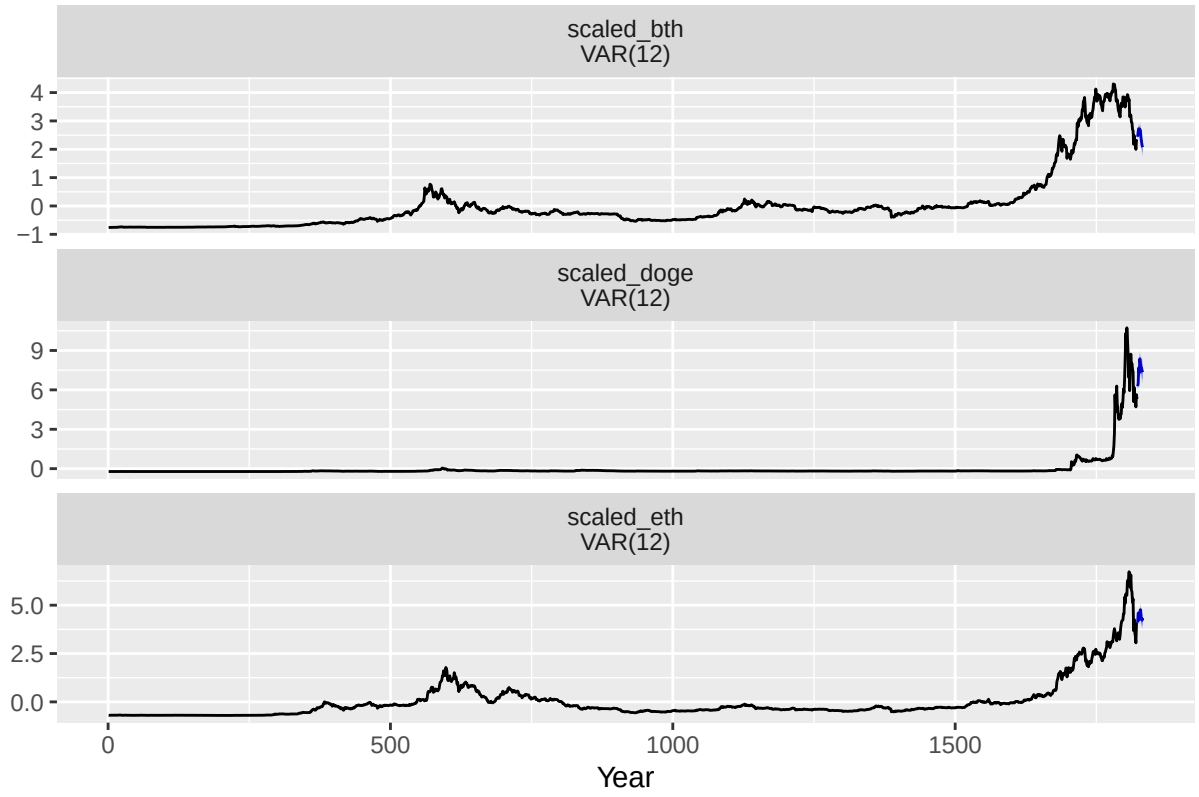
```
acf(residuals(fitvar))
```



```r
# UNSURE; TO BE LABELED
ts_obj <- as.ts(scaled_data_clean)
var12 <- VAR(ts_obj, p = 12, type = "const")
serial.test(var12, lags.pt = 12, type = "PT.asymptotic")
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object var12
## Chi-squared = 244.24, df = 0, p-value < 2.2e-16
```

```r
# forecasts

forecast(var12) %>%
  autoplot() + xlab("Year")
```




```r
# should we do an ADF test??

# granger test for causality COMMENTED OUT RN

grangertest(sl_btc ~ sl_doge, data = sl_data, order = 2)
```

```
## Granger causality test
##
```

```
## Model 1: sl_btc ~ Lags(sl_btc, 1:2) + Lags(sl_doge, 1:2)
## Model 2: sl_btc ~ Lags(sl_btc, 1:2)
##   Res.Df Df      F Pr(>F)
## 1   1815
## 2   1817 -2 1.8576 0.1563
```

```
#grangertest(BTC ~ ETH, data = og_data_clean)

  #grangertest(DOGE ~ BTC, order = 50, data = og_data_clean)
```

```
# removing NAs
sl_data$sl_btc<-na2lag1(sl_data$sl_btc)
sl_data$sl_doge<-na2lag1(sl_data$sl_doge)
sl_data$sl_eth<-na2lag1(sl_data$sl_eth)


# using results of VARselect to fit two models than compare them
fitvar_p9 <- VAR(sl_data[,-1], p = 9, type = "both")
fitvar_p1 <- VAR(sl_data[,-1], p = 1, type = "both")

summary(fitvar_p9)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: sl_btc, sl_doge, sl_eth
## Deterministic variables: both
## Sample size: 1817
## Log Likelihood: 10290.539
## Roots of the characteristic polynomial:
## 0.997 0.994 0.994 0.7428 0.7428 0.738 0.738 0.7244 0.7244 0.7173 0.7173 0.7138 0.7138 0.7036 0.7036
## Call:
## VAR(y = sl_data[, -1], p = 9, type = "both")
##
##
## Estimation results for equation sl_btc:
## =======================================
## sl_btc = sl_btc.l1 + sl_doge.l1 + sl_eth.l1 + sl_btc.l2 + sl_doge.l2 + sl_eth.l2 + sl_btc.l3 + sl_dog
##
##              Estimate Std. Error t value Pr(>|t|)
## sl_btc.l1   1.027e+00  3.213e-02  31.944  < 2e-16 ***
## sl_doge.l1  1.938e-02  1.766e-02   1.097 0.272668
## sl_eth.l1  -1.024e-01  2.808e-02  -3.648 0.000272 ***
## sl_btc.l2  -4.302e-02  4.693e-02  -0.917 0.359383
## sl_doge.l2  1.428e-02  2.513e-02   0.568 0.569856
## sl_eth.l2   1.400e-01  4.084e-02   3.428 0.000621 ***
## sl_btc.l3   2.967e-02  4.701e-02   0.631 0.528062
## sl_doge.l3 -3.598e-02  2.509e-02  -1.434 0.151776
## sl_eth.l3  -4.533e-02  4.095e-02  -1.107 0.268475
## sl_btc.l4  -7.042e-02  4.703e-02  -1.497 0.134513
## sl_doge.l4  1.452e-02  2.498e-02   0.581 0.561149
## sl_eth.l4   8.543e-02  4.085e-02   2.091 0.036627 *
## sl_btc.l5   9.714e-02  4.699e-02   2.067 0.038878 *
```

```
## sl_doge.l5  2.108e-03  2.503e-02   0.084 0.932877
## sl_eth.l5  -1.079e-01  4.087e-02  -2.640 0.008374 **
## sl_btc.l6  -1.438e-02  4.695e-02  -0.306 0.759389
## sl_doge.l6 -1.789e-02  2.496e-02  -0.717 0.473616
## sl_eth.l6   4.203e-02  4.092e-02   1.027 0.304575
## sl_btc.l7  -3.603e-02  4.685e-02  -0.769 0.441951
## sl_doge.l7 -1.629e-03  2.518e-02  -0.065 0.948421
## sl_eth.l7  -4.891e-02  4.101e-02  -1.192 0.233239
## sl_btc.l8   4.882e-02  4.672e-02   1.045 0.296242
## sl_doge.l8  2.609e-03  2.529e-02   0.103 0.917825
## sl_eth.l8  -3.192e-02  4.105e-02  -0.778 0.436846
## sl_btc.l9  -4.384e-02  3.190e-02  -1.374 0.169515
## sl_doge.l9 -2.041e-03  1.787e-02  -0.114 0.909087
## sl_eth.l9   7.474e-02  2.824e-02   2.646 0.008213 **
## const      -4.756e-03  3.680e-03  -1.292 0.196422
## trend       7.232e-06  3.899e-06   1.855 0.063766 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.03533 on 1788 degrees of freedom
## Multiple R-Squared: 0.9987,  Adjusted R-squared: 0.9987
## F-statistic: 5.098e+04 on 28 and 1788 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation sl_doge:
## ========================================
## sl_doge = sl_btc.l1 + sl_doge.l1 + sl_eth.l1 + sl_btc.l2 + sl_doge.l2 + sl_eth.l2 + sl_btc.l3 + sl_d
##
##              Estimate Std. Error t value Pr(>|t|)
## sl_btc.l1  -5.736e-02  4.889e-02  -1.173 0.240917
## sl_doge.l1  1.034e+00  2.687e-02  38.485  < 2e-16 ***
## sl_eth.l1  -2.622e-02  4.273e-02  -0.614 0.539500
## sl_btc.l2   5.784e-02  7.140e-02   0.810 0.418065
## sl_doge.l2  5.321e-03  3.824e-02   0.139 0.889338
## sl_eth.l2   5.424e-02  6.213e-02   0.873 0.382799
## sl_btc.l3  -2.513e-02  7.153e-02  -0.351 0.725374
## sl_doge.l3  2.994e-02  3.818e-02   0.784 0.433079
## sl_eth.l3   7.165e-03  6.231e-02   0.115 0.908476
## sl_btc.l4  -1.586e-02  7.157e-02  -0.222 0.824687
## sl_doge.l4 -1.036e-01  3.800e-02  -2.726 0.006479 **
## sl_eth.l4   5.564e-02  6.215e-02   0.895 0.370815
## sl_btc.l5   1.281e-01  7.150e-02   1.792 0.073333 .
## sl_doge.l5 -3.818e-02  3.808e-02  -1.002 0.316240
## sl_eth.l5  -4.973e-02  6.218e-02  -0.800 0.424021
## sl_btc.l6  -1.429e-01  7.144e-02  -2.000 0.045635 *
## sl_doge.l6  1.406e-01  3.798e-02   3.701 0.000221 ***
## sl_eth.l6   1.438e-02  6.227e-02   0.231 0.817429
## sl_btc.l7   1.337e-01  7.129e-02   1.876 0.060880 .
## sl_doge.l7 -7.247e-02  3.832e-02  -1.891 0.058769 .
## sl_eth.l7  -1.490e-01  6.240e-02  -2.388 0.017044 *
## sl_btc.l8  -1.425e-01  7.109e-02  -2.005 0.045165 *
## sl_doge.l8  3.073e-03  3.847e-02   0.080 0.936351
## sl_eth.l8   6.342e-02  6.245e-02   1.016 0.309990
```

```
## sl_btc.l9   7.253e-02  4.854e-02   1.494 0.135304
## sl_doge.l9 -3.976e-03  2.719e-02  -0.146 0.883764
## sl_eth.l9   2.890e-02  4.297e-02   0.672 0.501354
## const       4.376e-03  5.600e-03   0.781 0.434613
## trend      -2.155e-06  5.932e-06  -0.363 0.716427
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.05376 on 1788 degrees of freedom
## Multiple R-Squared: 0.9971,  Adjusted R-squared: 0.9971
## F-statistic: 2.217e+04 on 28 and 1788 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation sl_eth:
## ======================================
## sl_eth = sl_btc.l1 + sl_doge.l1 + sl_eth.l1 + sl_btc.l2 + sl_doge.l2 + sl_eth.l2 + sl_btc.l3 + sl_dog
##
##              Estimate Std. Error t value Pr(>|t|)
## sl_btc.l1  -5.852e-02  3.604e-02  -1.624   0.1046
## sl_doge.l1  3.849e-02  1.981e-02   1.943   0.0521 .
## sl_eth.l1   9.844e-01  3.149e-02  31.258   <2e-16 ***
## sl_btc.l2   9.314e-02  5.263e-02   1.770   0.0769 .
## sl_doge.l2 -2.771e-02  2.818e-02  -0.983   0.3256
## sl_eth.l2   3.852e-02  4.580e-02   0.841   0.4005
## sl_btc.l3  -5.005e-02  5.273e-02  -0.949   0.3426
## sl_doge.l3 -1.512e-02  2.814e-02  -0.537   0.5911
## sl_eth.l3   3.707e-04  4.593e-02   0.008   0.9936
## sl_btc.l4   2.991e-02  5.275e-02   0.567   0.5708
## sl_doge.l4  1.082e-02  2.801e-02   0.386   0.6993
## sl_eth.l4  -1.456e-02  4.581e-02  -0.318   0.7507
## sl_btc.l5   2.373e-02  5.270e-02   0.450   0.6525
## sl_doge.l5  2.517e-02  2.807e-02   0.897   0.3700
## sl_eth.l5  -2.930e-02  4.584e-02  -0.639   0.5227
## sl_btc.l6  -3.450e-02  5.266e-02  -0.655   0.5125
## sl_doge.l6 -4.787e-02  2.800e-02  -1.710   0.0875 .
## sl_eth.l6   7.411e-02  4.590e-02   1.615   0.1065
## sl_btc.l7  -2.740e-02  5.255e-02  -0.521   0.6021
## sl_doge.l7  5.345e-02  2.824e-02   1.892   0.0586 .
## sl_eth.l7  -7.923e-02  4.600e-02  -1.722   0.0852 .
## sl_btc.l8   5.884e-02  5.240e-02   1.123   0.2616
## sl_doge.l8 -5.500e-02  2.836e-02  -1.939   0.0526 .
## sl_eth.l8  -4.759e-03  4.603e-02  -0.103   0.9177
## sl_btc.l9  -3.139e-02  3.578e-02  -0.877   0.3803
## sl_doge.l9  1.679e-02  2.004e-02   0.838   0.4023
## sl_eth.l9   2.711e-02  3.168e-02   0.856   0.3922
## const       2.280e-03  4.127e-03   0.552   0.5807
## trend      -6.635e-07  4.372e-06  -0.152   0.8794
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.03963 on 1788 degrees of freedom
## Multiple R-Squared: 0.9984,  Adjusted R-squared: 0.9984
```

```
## F-statistic: 4.068e+04 on 28 and 1788 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##            sl_btc   sl_doge    sl_eth
## sl_btc  0.0012485 0.0008596 0.0009070
## sl_doge 0.0008596 0.0028906 0.0008759
## sl_eth  0.0009070 0.0008759 0.0015704
##
## Correlation matrix of residuals:
##         sl_btc sl_doge sl_eth
## sl_btc  1.0000  0.4525 0.6478
## sl_doge 0.4525  1.0000 0.4111
## sl_eth  0.6478  0.4111 1.0000
```

```
summary(fitvar_p1)
```

```
##
## VAR Estimation Results:
## ========================
## Endogenous variables: sl_btc, sl_doge, sl_eth
## Deterministic variables: both
## Sample size: 1825
## Log Likelihood: 10254.74
## Roots of the characteristic polynomial:
## 0.9969 0.9955 0.9955
## Call:
## VAR(y = sl_data[, -1], p = 1, type = "both")
##
##
## Estimation results for equation sl_btc:
## =======================================
## sl_btc = sl_btc.l1 + sl_doge.l1 + sl_eth.l1 + const + trend
##
##              Estimate Std. Error t value Pr(>|t|)
## sl_btc.l1   9.960e-01  4.069e-03 244.781   <2e-16 ***
## sl_doge.l1 -3.777e-03  2.103e-03  -1.796   0.0727 .
## sl_eth.l1   3.967e-03  2.995e-03   1.324   0.1855
## const      -3.149e-03  3.626e-03  -0.869   0.3852
## trend       5.738e-06  3.861e-06   1.486   0.1375
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.03564 on 1820 degrees of freedom
## Multiple R-Squared: 0.9987,  Adjusted R-squared: 0.9987
## F-statistic: 3.573e+05 on 4 and 1820 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation sl_doge:
## ========================================
## sl_doge = sl_btc.l1 + sl_doge.l1 + sl_eth.l1 + const + trend
##
```

```
##              Estimate Std. Error t value Pr(>|t|)
## sl_btc.l1   8.445e-03  6.181e-03   1.366    0.172
## sl_doge.l1  9.963e-01  3.194e-03 311.899   <2e-16 ***
## sl_eth.l1  -1.971e-03  4.551e-03  -0.433    0.665
## const       5.322e-03  5.508e-03   0.966    0.334
## trend      -2.837e-06  5.866e-06  -0.484    0.629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.05414 on 1820 degrees of freedom
## Multiple R-Squared: 0.9971,  Adjusted R-squared: 0.9971
## F-statistic: 1.547e+05 on 4 and 1820 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation sl_eth:
## =======================================
## sl_eth = sl_btc.l1 + sl_doge.l1 + sl_eth.l1 + const + trend
##
##              Estimate Std. Error t value Pr(>|t|)
## sl_btc.l1   4.834e-03  4.536e-03   1.066    0.287
## sl_doge.l1 -1.001e-05  2.344e-03  -0.004    0.997
## sl_eth.l1   9.955e-01  3.340e-03 298.076   <2e-16 ***
## const       3.875e-03  4.042e-03   0.959    0.338
## trend      -2.007e-06  4.305e-06  -0.466    0.641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.03973 on 1820 degrees of freedom
## Multiple R-Squared: 0.9984,  Adjusted R-squared: 0.9984
## F-statistic: 2.875e+05 on 4 and 1820 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##            sl_btc   sl_doge    sl_eth
## sl_btc  0.0012699 0.0008696 0.0009107
## sl_doge 0.0008696 0.0029308 0.0008757
## sl_eth  0.0009107 0.0008757 0.0015786
##
## Correlation matrix of residuals:
##         sl_btc sl_doge sl_eth
## sl_btc  1.0000  0.4508 0.6432
## sl_doge 0.4508  1.0000 0.4071
## sl_eth  0.6432  0.4071 1.0000
```

```r
# diagnostic tests

# Asymptotic Portmanteau test

# we fail to find evidence of autocorrelation in the residuals of the VAR(9) model at the default of 16
serial.test(fitvar_p9, lags.pt = 50)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object fitvar_p9
## Chi-squared = 387.05, df = 369, p-value = 0.2487
```

```
# we find evidence of autocorrelation in the residuals of the VAR(1) model
serial.test(fitvar_p1, lags.pt = 40)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object fitvar_p1
## Chi-squared = 503.99, df = 351, p-value = 1.494e-07
```

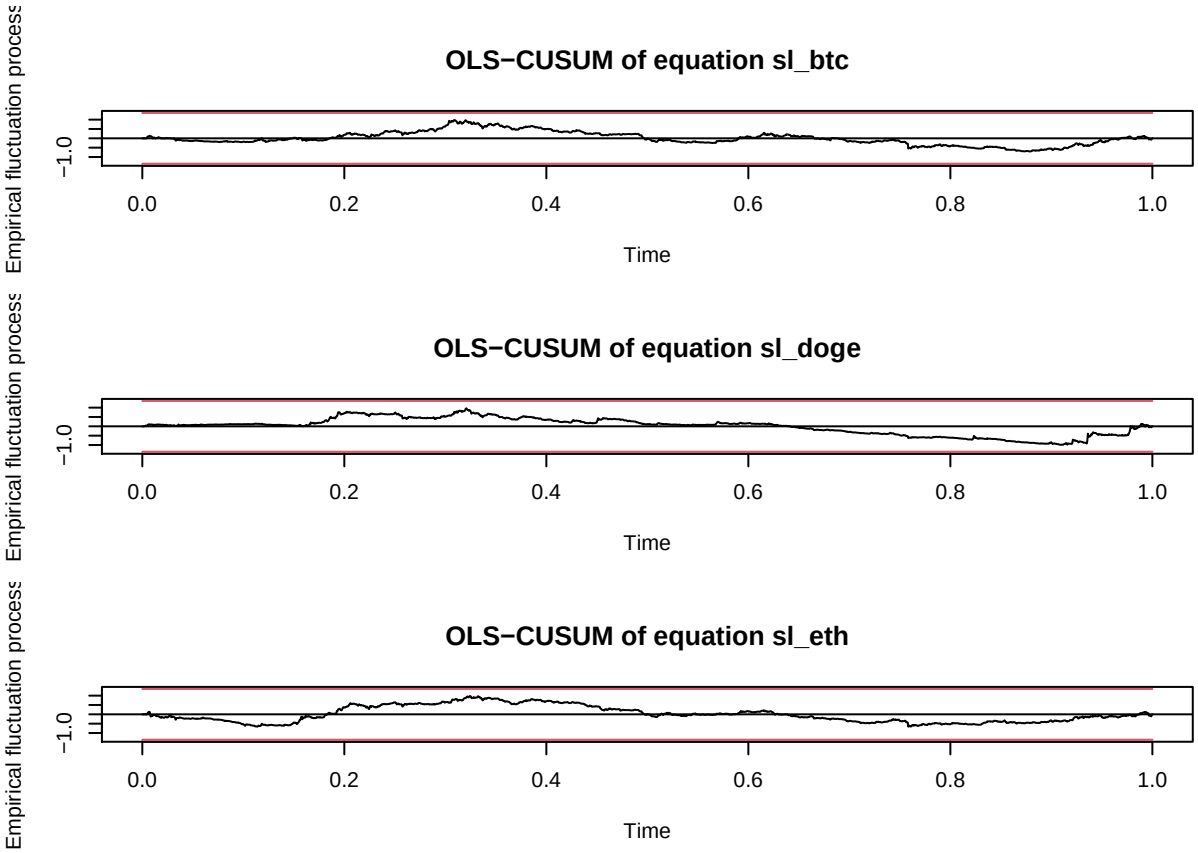
```
# Breusch-Godfrey test

# same reversial present for the BG test...
serial.test(fitvar_p9,lags.bg = 40, type = 'BG')
```

```
##
##  Breusch-Godfrey LM test
##
## data:  Residuals of VAR object fitvar_p9
## Chi-squared = 529.28, df = 360, p-value = 1.434e-08
```

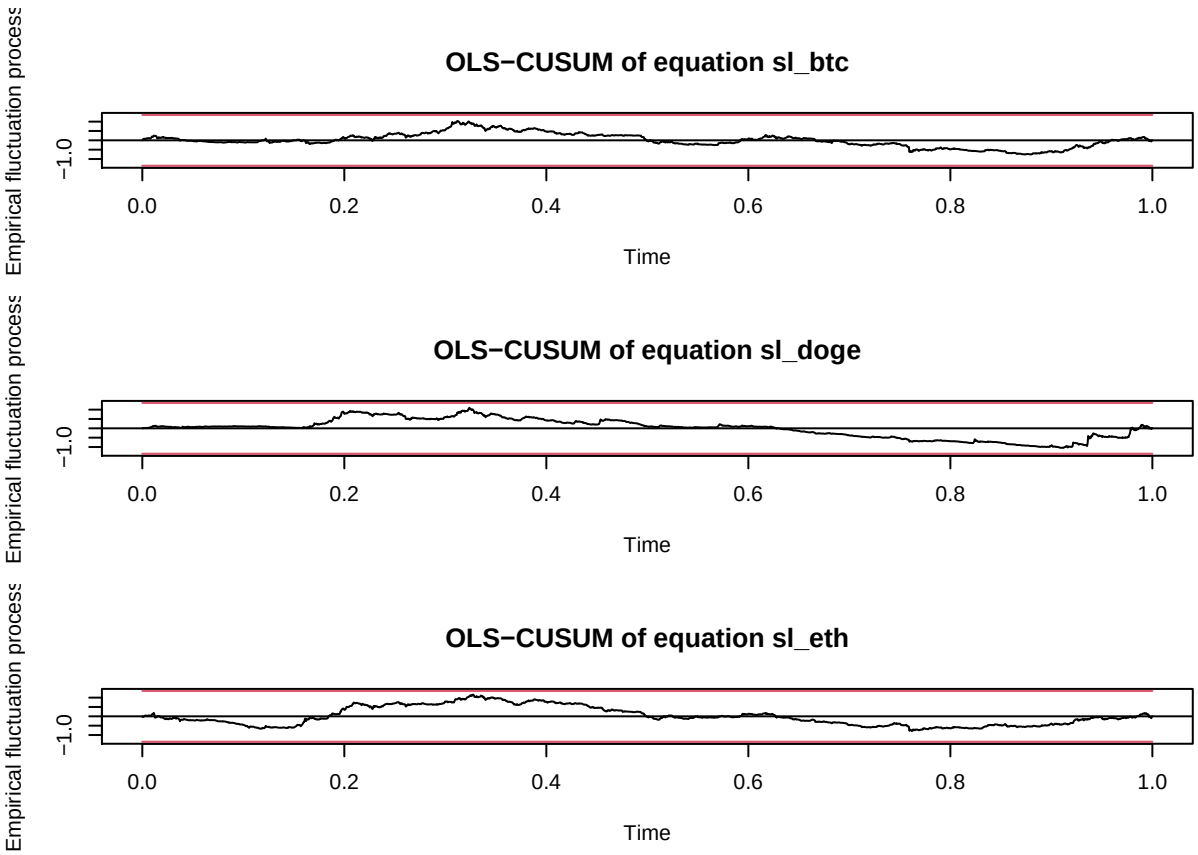
```
serial.test(fitvar_p1,lags.bg = 40, type = 'BG')
```

```
##
##  Breusch-Godfrey LM test
##
## data:  Residuals of VAR object fitvar_p1
## Chi-squared = 508.58, df = 360, p-value = 3.849e-07
```

```
# more diagnostics
stablity_test_p9 <-stability(fitvar_p9, type = c("OLS-CUSUM"))
stablity_test_p1 <-stability(fitvar_p1, type = c("OLS-CUSUM"))

plot(stablity_test_p9)
```




```
plot(stablity_test_p1)
```




```
# Jarque-Bera test is a goodness-of-fit test of whether sample data have the skewness and kurtosis matc
p9_normal <- normality.test(fitvar_p9)

#plot(p9_normal)


p1_normal <- normality.test(fitvar_p1)

#plot(p1_normal)

# Portmanteau Q and test for the null hypothesis that the residuals of a ARIMA model are homoscedastic
arch.test(fitvar_p9)
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object fitvar_p9
## Chi-squared = 900.25, df = 180, p-value < 2.2e-16
```

```
arch.test(fitvar_p1)
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object fitvar_p1
```

```
## Chi-squared = 914.61, df = 180, p-value < 2.2e-16
```

```r
#making some forecasts

#need to refit model using slightly differently formatted data
sl_data_as_ts <- as.ts(sl_data[,-1])
fitvar_p9 <- VAR(sl_data_as_ts, p = 9, type = "both")

#forecast(fitvar_p9, h = 365) %>%
  #autoplot() + xlab("Year")

fitvar_p1 <- VAR(sl_data_as_ts, p = 1, type = "both")

#forecast(fitvar_p1, h = 365) %>%
 # autoplot() + xlab("Year")

#irf stuff

#plot(irf(fitvar_p9))


#plot(fitvar_p9)
```