A TIME SERIES ANALYSIS OF NORTH AMERICAN BANKRUPTCY RATE

Model Selection

- We fit three competing models to the data to attempt to forecast the ISOL rate ahead five periods.
- ARMA(22,1,0) model
- Vector autoregression model with p = 22
- **Exponential smoothing model with additive seasonal component – this was selected for the competition!**
- Note: We also considered a model with seasonal dummies, but the dummies were not found to be significant and did not lend to white noise error terms. Additionally, this model could not be used to predict without new data.

Vector Autoregression Model

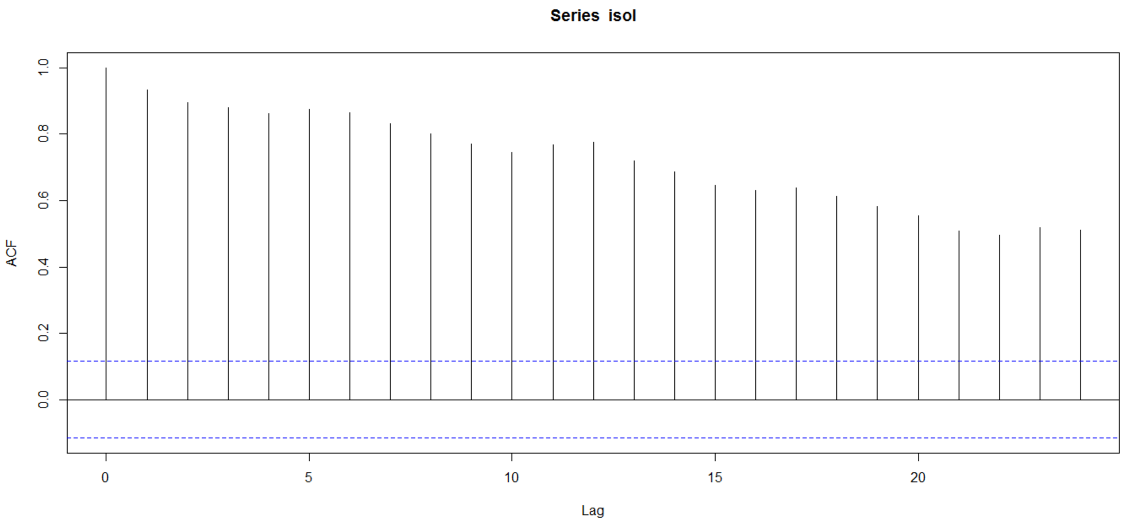
- VAR models are designed for multivariate data. They assume that the input variables are stationary, and work in a sense ‘together’ as a system of variables.
- Most of the assumption/diagnostic requirements for these models are the same as typical ARMA type models – VARs are a multivariate extension of this class of model.
- A VAR(p) model can be specified with p lags.
- VAR models are not appropriate for cointegrated series (series that move in response to one another).

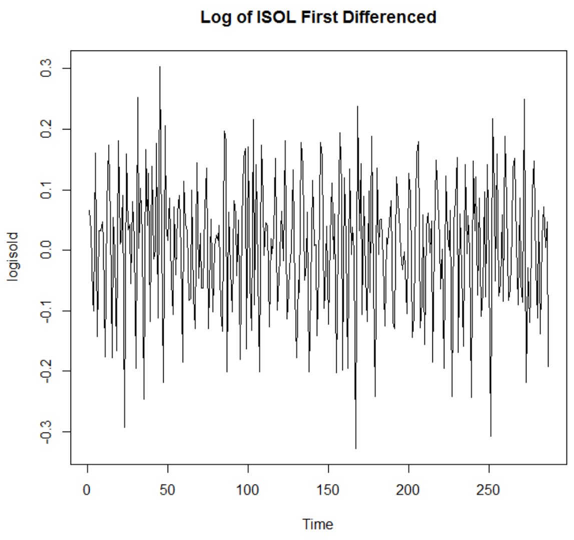
Johansen’s Cointegration Test

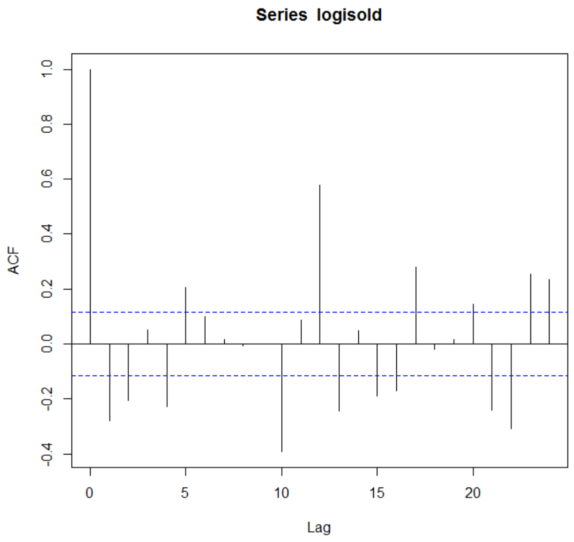
- We must first establish if the series are cointegrated to tell whether or not a VAR model is appropriate. The Johnansen’s Cointegration test can handle this task. I excluded pop from the analysis.
- Since none of the test statistics reach the critical values here, we can conclude that there is no conintegration amongst these series.
- This was done with library(urca), ca.jo(data) command

```
######################
# Johansen-Procedure #
######################

Test type: trace statistic , with linear trend

Eigenvalues (lambda):
[1] 0.088210190 0.029220444 0.003026698

Values of teststatistic and critical values of test:

          test 10pct  5pct  1pct
r <= 2 |  0.87  6.50  8.18 11.65
r <= 1 |  9.35 15.66 17.95 23.52
r = 0  | 35.76 28.71 31.52 37.22

Eigenvectors, normalised to first column:
(These are the cointegration relations)

              isol.l2        UR.l2       HPI.l2
isol.l2  1.0000000000  1.0000000000  1.000000000
UR.l2    0.0030363705 -0.0045491403 -0.024380055
HPI.l2  -0.0003413373 -0.0003479528 -0.005047043

Weights W:
(This is the loading matrix)

           isol.l2       UR.l2        HPI.l2
isol.d -0.087344 -0.03034698  0.0004173136
UR.d   -4.176374  2.25419612 -0.1302313228
HPI.d  -3.412555  2.87281143  0.1948347492
```

Preprocessing of data: ISOL

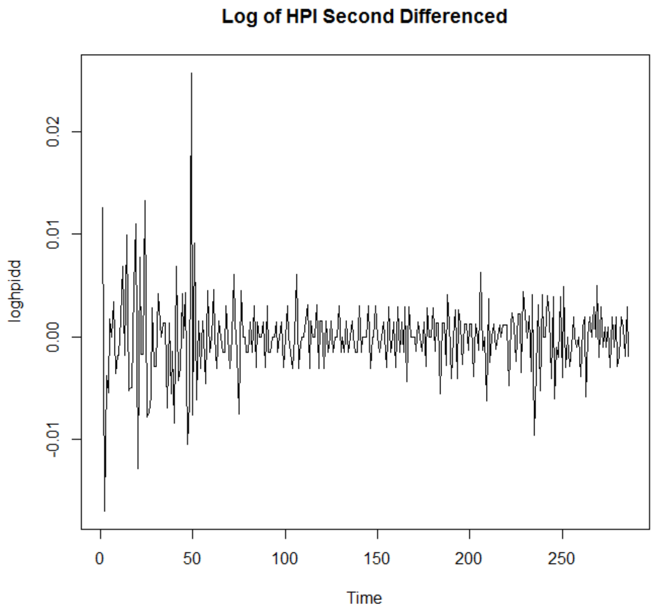


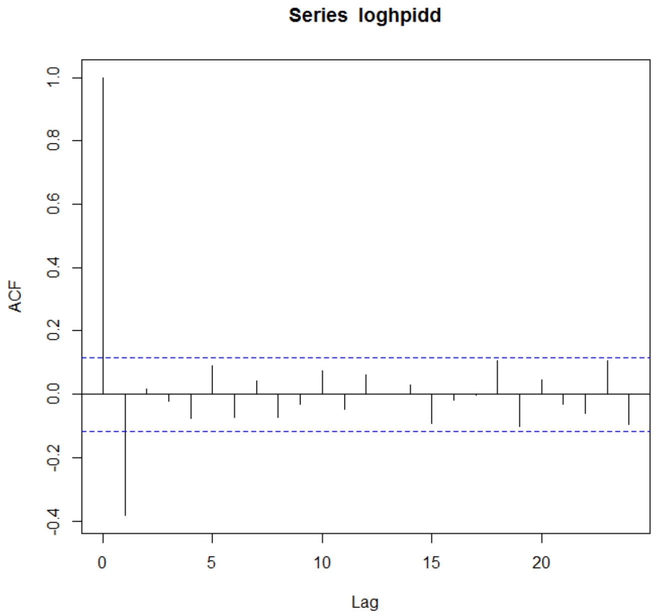


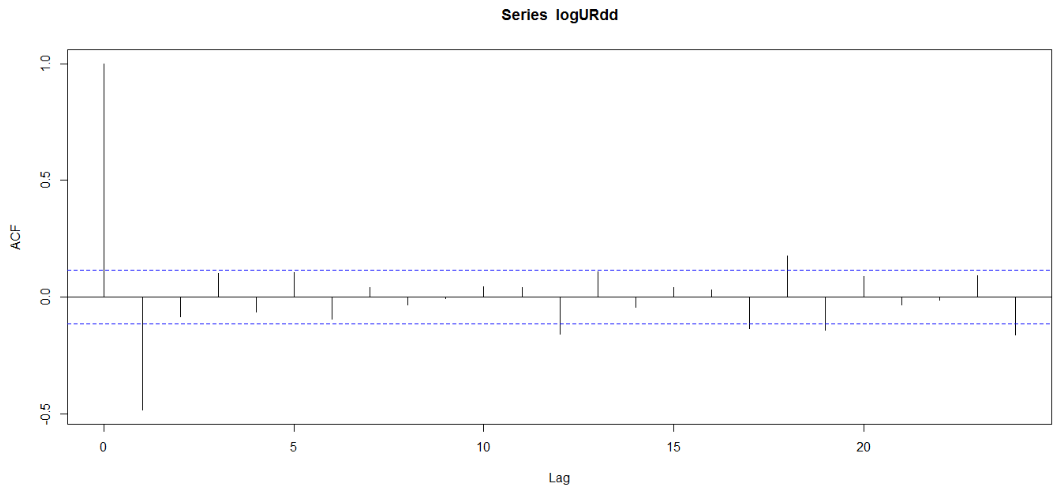


- This series was difficult to be completely stationarized. The best we could get was the first difference of the log transformation of isol.
- The ACF indicates that there is still some serial correlation (significant spikes exist in the acf).
- However, it is an improvement over the original acf pictured above. VAR models do require stationary variable inputs, but given the data this is probably the best we can get.

Preprocessing of data: HPI







- The second difference of the log of HPI still shows some autocorrelation at lag 1, but overall is better than the original series. It could still be included in the VAR with some caution.
- The second difference of UR had a very similar acf. We chose to exclude it from our final VAR model since adding an additional series that isn’t perfectly stationary wouldn’t be of any use.

Justification for Preprocessing

```
> adf.test(logisold, k=0)

        Augmented Dickey-Fuller Test

data:  logisold
Dickey-Fuller = -22.399, Lag order = 0, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(logisold, k = 0) : p-value smaller than printed p-value
> adf.test(logisol)

        Augmented Dickey-Fuller Test

data:  logisol
Dickey-Fuller = -2.0311, Lag order = 6, p-value = 0.5631
alternative hypothesis: stationary
```

While the acf for log(isol) first differenced is not the best – an augmented dickey fuller test concludes that the data may be stationary. So, we can proceed with using it in the VAR, although with some caution since the acf looks bad.

More for Preprocessing

The second difference of the log of HPI was chosen here since its p-value is the smallest.

The first difference was just on the border of significance, indicating that the first difference alone might not be enough to ensure stationary.

While the acf looks bad, we still have some evidence that we can use this data in a VAR model.

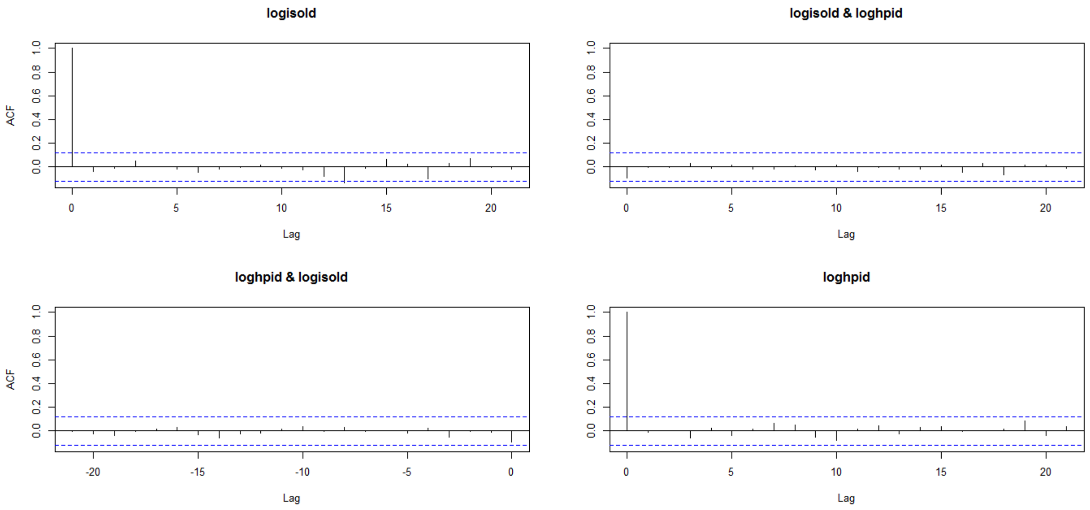
```
        Augmented Dickey-Fuller Test

data:  hpi
Dickey-Fuller = -1.4508, Lag order = 6, p-value = 0.8076
alternative hypothesis: stationary

> adf.test(loghpi)

        Augmented Dickey-Fuller Test

data:  loghpi
Dickey-Fuller = -1.7594, Lag order = 6, p-value = 0.6775
alternative hypothesis: stationary

> adf.test(loghpid)

        Augmented Dickey-Fuller Test

data:  loghpid
Dickey-Fuller = -3.3274, Lag order = 6, p-value = 0.06693
alternative hypothesis: stationary

> adf.test(loghpidd)

        Augmented Dickey-Fuller Test

data:  loghpidd
Dickey-Fuller = -7.8602, Lag order = 6, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In adf.test(loghpidd) : p-value smaller than printed p-value
```

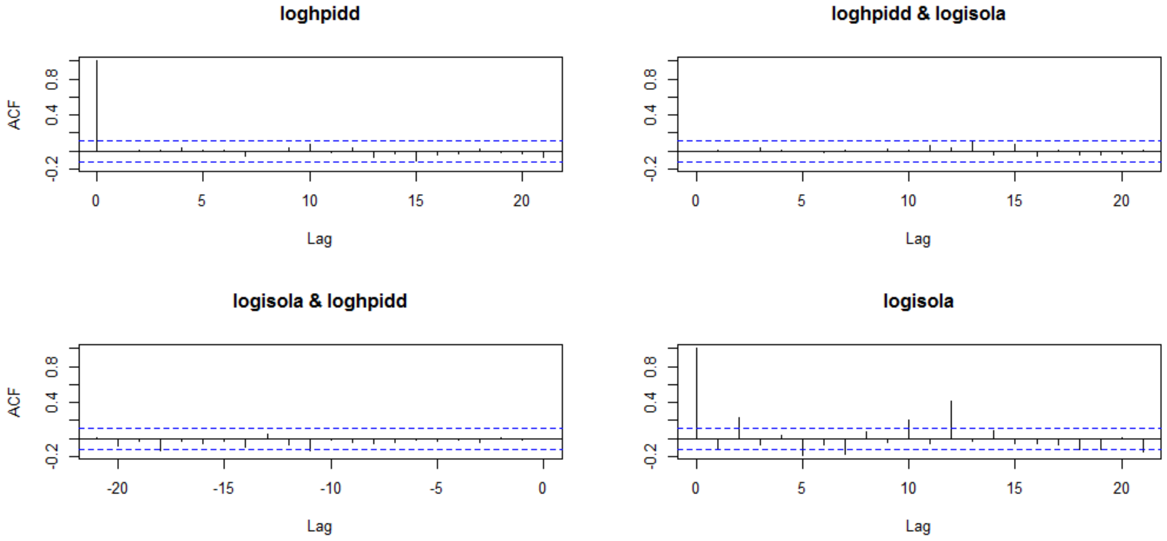
Development of VAR Model

```
AIC(n)  HQ(n)  SC(n) FPE(n) 
    10     10      4     10 

$criteria
                   1             2             3             4             5             6             7             8             9            10
AIC(n) -1.575380e+01 -1.586081e+01 -1.585227e+01 -1.601898e+01 -1.599605e+01 -1.602141e+01 -1.602693e+01 -1.603800e+01 -1.612952e+01 -1.625984e+01
HQ(n)  -1.572221e+01 -1.580817e+01 -1.577858e+01 -1.592424e+01 -1.588024e+01 -1.588455e+01 -1.586901e+01 -1.585903e+01 -1.592949e+01 -1.603876e+01
SC(n)  -1.567509e+01 -1.572964e+01 -1.566863e+01 -1.578287e+01 -1.570746e+01 -1.568036e+01 -1.563341e+01 -1.559201e+01 -1.563106e+01 -1.570891e+01
FPE(n)  1.439506e-07  1.293426e-07  1.304538e-07  1.104241e-07  1.129904e-07  1.101669e-07  1.095688e-07  1.083727e-07  9.890756e-08  8.683525e-08
```

R has a function {VARSelect} that can select the lag order p of the var model. The AIC method here selects 10 lags.

VAR Residuals ACF Output





- The model was with with R code **var <- VAR(data2, p=10 (and 20), type="const")**
- The p=10 model’s residuals (first acf to the right) look mostly stationary, except for the residuals related to log(isol).
- An improvement over these residuals can be found by specifying a VAR(22) model (first acf to the left), chosen because of the AR(22) model specified later, selected by the Yuler Walker method. Although these models are different, we just wanted to see if setting the same lag order improved the model residuals, and it did. The

VAR Normality Test On VAR(22) model

- The JB-Test for multivariate normality is concerning here, as it indicates that the normality assumption of the model is violated.
- Given the good looking residuals in terms of serial correlation, we can still proceed to use the model for prediction, although with caution since normality is violated here.

```
> normality.test(var2, multivariate.only= TRUE)
$JB

        JB-Test (multivariate)

data:  Residuals of VAR object var2
Chi-squared = 291.52, df = 4, p-value < 2.2e-16


$Skewness

        Skewness only (multivariate)

data:  Residuals of VAR object var2
Chi-squared = 11.743, df = 2, p-value = 0.002818


$Kurtosis

        Kurtosis only (multivariate)

data:  Residuals of VAR object var2
Chi-squared = 279.78, df = 2, p-value < 2.2e-16
```

VAR(22) Model Training

Obs	Predicted	Actual	Deviation
285	0.037377	0.033316	-0.00406112
286	0.031718	0.033461	0.00174259
287	0.032277	0.035049	0.00277209
288	0.027702	0.028922	0.0012204

RMSE Calculated From library(Metrics) : 0.00267872

We trained the model by removing observations 285-288, and predicting the observations with the VAR model (although with 4 observations).

The model’s performance is not bad, but there are some minor deviations from actual values.

Model Forecasting

Obs	Forecast	Lower	Upper	Length
289	0.035072	0.034565	0.035559	0.000994
290	0.035172	0.034305	0.035825	0.00152
291	0.035381	0.034047	0.036086	0.00204
292	0.035548	0.033793	0.036353	0.00256
293	0.035733	0.03354	0.03662	0.00308

The widths of the prediction intervals here are reasonably small, but again we are making a trade-off using this model since we know we are violating the normality assumption, and since our series were not completely stationary when we fed them into the model.

Since the data ranges from [.006861, .0457978] with most values being above .010, the prediction intervals are well calibrated, especially since they are close to the most recent values observed in the dataset which

Simple Exponential Smoothing Model(SES)

We apply an exponential smoothing model.

We suspect that there is some minimal seasonality in the series due to oscillations in the acf.

which must be accounted for in the exponential smoothing model. There is an R code which can detect seasonality by fitting a seasonal model, and then a non-seasonal model.

If the seasonal model is selected, then you can conclude the series does have seasonality – and this is what we saw in the ISOL series. The code for this is to the right.

```
> df <- attributes(logLik(fit1))$df - attributes(logLik(fit2))$df
> 1 - pchisq(deviance,df)
[1] 0
> pchisq(deviance,df)
[1] 1
> deviance
[1] 76.79655
> df
[1] 2
> deviance <- 2*c(logLik(fit1) - logLik(fit2))
> deviance
[1] 76.79655
> df <- attributes(logLik(fit1)$df - attributes(logLik(fit2))$df
+ )
Error in logLik(fit1)$df : $ operator is invalid for atomic vectors
> df <- attributes(logLik(fit1))$df - attributes(logLik(fit2))$df
> df
[1] 2
> 1 - pchisq(deviance, df)
[1] 0
> acf(isol)
```

Model Selection(PT2)

```
> m2 <- HoltWinters(ts(trainisol, frequency=12), beta=0)
> m3 <- HoltWinters(ts(trainisol, frequency=12), gamma=0)
> m4 <- HoltWinters(ts(trainisol, frequency=12)
+
+ m4 <- HoltWinters(ts(trainisol, frequency=12))
Error: unexpected symbol in:
"
m4"
> m5 <- HoltWinters(ts(trainisol, frequency=12), seasonal="multiplicative")
>
> m1 <- HoltWinters(ts(trainisol, frequency=12), beta=0, gamma=0)
```

```
> m1$SSE
[1] 0.001216535
> m2$SSE
[1] 0.000871944
> m3$SSE
[1] 0.001216535
> m4$SSE
[1] 0.0008911292
> m5$SSE
[1] 0.000893729
```

As confirmation for seasonality in the model found in part two, we ran a number of exponential smoothing models. The lowest model, model 2, was selected based on it having the smallest SSE. This model contains trend and an additive seasonal component.

Training of Exponential Smoothing

We trained the model by eliminating the last five observations, fitting the additive seasonal with trend exponential smoothing model on the remaining data, and then comparing the predictions against the actual results.

The model actual does quite nicely in predicting the observations, with the last observations being a bit off.

Obs	Predicted	Actual	Deviation
284	0.031181	0.03102	-0.00016
285	0.034449	0.033316	-0.00113
286	0.03343	0.033461	3.1E-05
287	0.03274	0.035049	0.002308
288	0.028544	0.028922	0.000378

RMSE Calculated from library(Metrics): 0.001299905

Prediction from SES Model

Obs	Forecast	Lower	Upper	Length
289	0.030584	0.027089	0.034079	0.00699
290	0.033731	0.029952	0.03751	0.007559
291	0.037098	0.033054	0.041142	0.008087
292	0.036273	0.031981	0.040565	0.008584
293	0.03439	0.029864	0.038917	0.009053

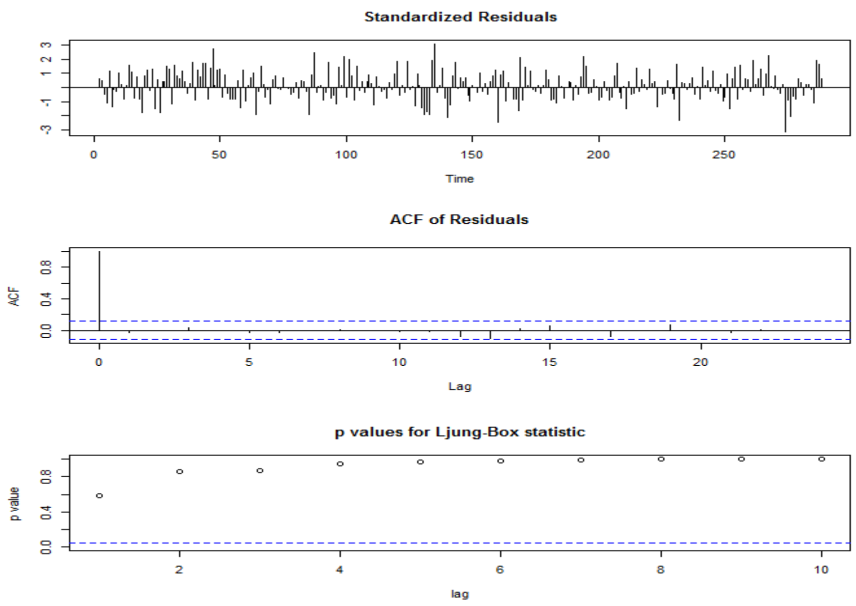
These five points are the candidates from this model for prediction. Since the data ranges from [.006861, .0457978] with most values being above .010, the prediction intervals are well calibrated, especially since they are close to the most recent values observed in the dataset which are near 0.030.

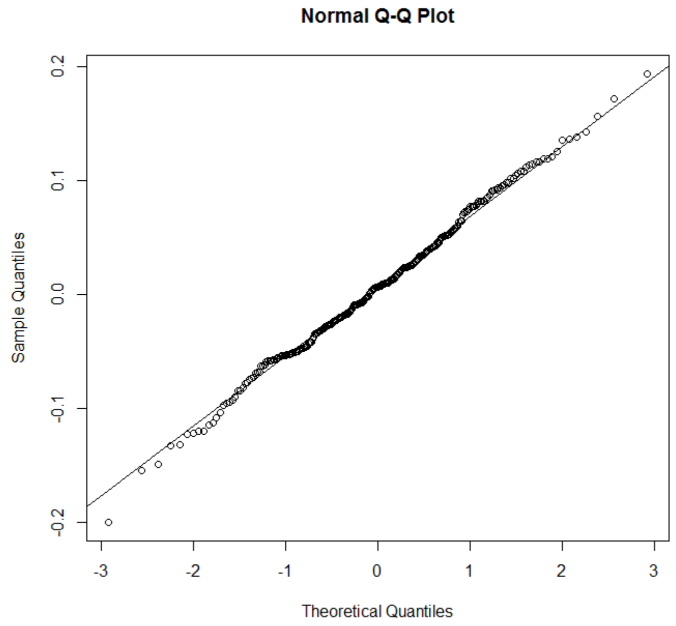
ARMA(22,1,0) Model

```
> test <- ar.yw(logisold)
> test

Call:
ar.yw.default(x = logisold)

Coefficients:
      1        2        3        4        5        6        7        8        9
-0.5320  -0.3187  -0.0412  -0.1062   0.1442   0.0352   0.0728   0.0659   0.1228
     10       11       12       13       14       15       16       17       18
-0.1091  -0.0204   0.5205   0.2656   0.1722  -0.0922  -0.0618  -0.1032  -0.0739
     19       20       21       22
-0.1194  -0.0685  -0.2469  -0.1801
```

- We ran the Yule-Walker equations in R without a lag limit on the first difference of the log of isol, and the Yule-Walker method selected 22 autoregressive terms.

Diagnostics





The model was fit with R code **model1 <- arima0(logisol, order=(22,1,0), period=12).** Although 22 AR terms seems a bit overspecified, we sacrifice simplicity here for very good residual diagnostics, meeting all model assumptions.

There is no pattern on the timeplot of the residuals that would indicate variance assumption violations, the acf looks very good, all Ljung-Box statistics are high, and we appear to have normal residuals. In fact, these are

Training of AR(22,1,0) Model

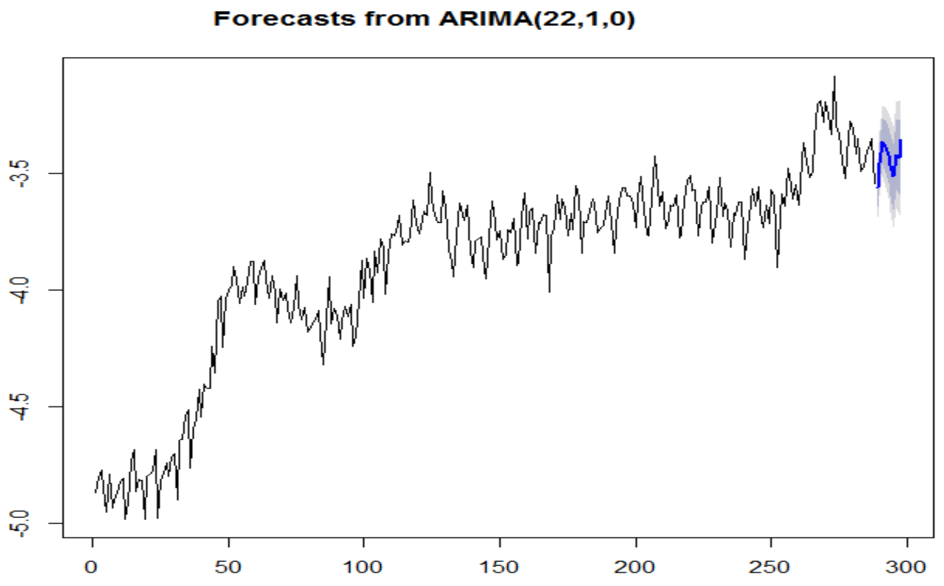
- We removed observations 284-288 from the series and used it to test the predictive power of the AR(22,1,0) model. The results are to the right
- Overall, the model doesn't do a bad job, but there are some deviations.

RMSE Calculated: 0.003421741

Obs	Predicted	Actual	Difference
284	0.031152	0.03102	-0.00013
285	0.035726	0.033316	-0.00241
286	0.030297	0.033461	0.003163
287	0.030453	0.035049	0.004596
288	0.025777	0.028922	0.003145

Prediction of AR(22,1,0) Model

Obs	Forecast	Lower	Upper	Length
289	0.028317	0.024994	0.03208	0.007086
290	0.031472	0.027439	0.036098	0.00866
291	0.034586	0.029778	0.04017	0.010392
292	0.033818	0.028516	0.040104	0.011588
293	0.032603	0.027182	0.039105	0.011924



- 95% prediction intervals for the next five observations for the AR(22) model are included to the right.
- Also, a forecast of the series in LOGS is graphed below it, to give an idea of the overall stability of the prediction vs. the actual series. The model appears to be doing a decent job in predicting since the direction of the prediction doesn't seem too far off base.

Comparison of Selected Models

Model	Avg Length	RMSE
VAR	**0.002039**	0.002678722
Exp Smth.	0.008054548	**0.00129990**
AR(22,1,0)	0.009929674	0.003421741

- ✓ The average length of periods we're trying to predict is included. Also, the RMSE are included from the value in the TRAINING sets is included. These sums begin at observation 285 and measure the deviations from actual values to predicted values.

- ✓ The VAR model appears to have the best average length of the prediction interval, however, the exponential smoothing has the smallest RMSE and tends to predict the actual training data better than any of the other models.

- ✓ We believe that because the AR(22,1,0) model and VAR(22) model are overspecified in their effort to satisfy assumptions, that it's better to use the exponential smoothing model. It's simpler, and appears to work better on the training data set when RMSE is considered.