# INDR 450/550

Spring 2022

Lecture 8: ARIMA processes (3)

March 9, 2022

**Fikri Karaesmen**

# Announcements

- Class Exercise at the end of lecture today. If you are participating online, please upload your document under Course Contents/Class Exercises

- HW 1 now available.

- The first two labs were uploaded. Please follow them.

# The Backshift Notation

The backward shift operator $B$ is a useful notational device when working with time series lags:

$$By_t = y_{t-1} \, .$$

(Some references use $L$ for "lag" instead of $B$ for "backshift".) In other words, $B$, operating on $y_t$, has the effect of shifting the data back one period. Two applications of $B$ to $y_t$ shifts the data back two periods:

$$B(By_t) = B^2 y_t = y_{t-2} \, .$$

For monthly data, if we wish to consider "the same month last year," the notation is $B^{12} y_t = y_{t-12}$.

The backward shift operator is convenient for describing the process of *differencing*. A first difference can be written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B) y_t \, .$$

Note that a first difference is represented by $(1 - B)$. Similarly, if second-order differences have to be computed, then:

$$y''_t = y_t - 2y_{t-1} + y_{t-2} = (1 - 2B + B^2) y_t = (1 - B)^2 y_t \, .$$

In general, a $d$th-order difference can be written as

$$(1 - B)^d y_t .$$

# The Backshift Formulation for ARIMA

ARIMA(p,d,q):

$$\underbrace{(1-\phi_1 B - \cdots - \phi_p B^p)}_{\substack{\uparrow \\ \text{AR}(p)}} \quad \underbrace{(1-B)^d y_t}_{\substack{\uparrow \\ d \text{ differences}}} \quad = \quad c + \underbrace{(1+\theta_1 B + \cdots + \theta_q B^q)\varepsilon_t}_{\substack{\uparrow \\ \text{MA}(q)}}$$

# ARMA: Backshift example

- ARMA: mixing AR and MA terms
- Ex: $Y_t = c + \phi_1 Y_{t-1} + \theta_4 \varepsilon_{t-4} + \varepsilon_t$

- Using the backshift notation, we can write:

$$(1 - \phi_1 B) Y_t = c + (1 + \theta_4 B^4) \epsilon_t$$

# ARIMA: Backshift example

- Consider ARIMA(1,2,1), we write

$$Y_t = c + \phi_1 Y_{t-1} + \theta_1 \varepsilon_{t-4} + \varepsilon_t$$

- Using the backshift notation, we can write:

$$(1 - \phi_{1} B)(1 - B)^2 Y_t = c + (1 + \theta_1 B)\epsilon_t$$

- It's now easy to see that $Y_t$ is related to terms upto$Y_{t\text{-}3}$.

# SARIMA: Backshift example

- Consider SARIMA(1,1,1)(1,1,1,4) we write

- Using the backshift notation, we can write:

$$(1-\phi_1 B)\,(1-\Phi_1 B^4)(1-B)(1-B^4)y_t = (1+\theta_1 B)\,(1+\Theta_1 B^4)\varepsilon_t.$$

- It’s now easy to see that $Y_t$ is related to terms upto$Y_{t\text{-}10}$.

# SARIMA Framework: Example

- SARIMA(1,0,1)(1,1,0,12) refers to a process which has an one regular AR term and was seasonally differenced once and has an AR term on the seasonal difference. The length of the season is 12.

- Using the backshift notation, we can write:

$$(1-\phi_1 B)(1-\Phi_1 B^{12})(1-B^{12})Y_t = c + (1+\theta_1 B^1)\epsilon_t$$

- We can nowe see that $Y_t$ is related to terms upto$Y_{t\text{-}25}$.

# Model Fitting Examples: (wrong) ARMA

Let's check the effect of fitting a wrong model. For instance, we might wrongfully think that MA terms are needed at lags 1 and 3.

- We can also attempt to fit a wrong (or superficial) model.For instance, we can attempt to fit:

$$Y_t = c + \phi_1 Y_{t-1} + \theta_1 \epsilon_{t-1} + \theta_3 \epsilon_{t-3} + \epsilon_t$$

- Note that the above is not exactly ARIMA(1,0,3) since it does not contain the MA-term at the second lag.
- We would need a more complete specification and use ARIMA(1,0,[1,0,1]).

# Model Fitting Examples: (wrong) ARMA

In [4]:
```python
# Fit the model
restest = modtest.fit(disp=False)
print(restest.summary());
```

```
                               SARIMAX Results
==================================================================================
Dep. Variable:                          y   No. Observations:                  399
Model:             SARIMAX(1, 0, [1, 3])   Log Likelihood               -1483.013
Date:                    Sun, 06 Mar 2022   AIC                           2976.026
Time:                            18:47:31   BIC                           2995.971
Sample:                                 0   HQIC                          2983.926
                                    - 399
Covariance Type:                      opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept    275.9916     49.422      5.584      0.000     179.127     372.857
ar.L1          0.6693      0.059     11.292      0.000       0.553       0.786
ma.L1          0.0112      0.077      0.146      0.884      -0.140       0.162
ma.L3         -0.0270      0.061     -0.442      0.658      -0.147       0.093
sigma2        98.9059      7.420     13.329      0.000      84.362     113.450
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 0.36
Prob(Q):                              0.96   Prob(JB):                         0.84
Heteroskedasticity (H):               0.92   Skew:                             0.05
Prob(H) (two-sided):                  0.64   Kurtosis:                         2.89
===================================================================================
```

# Model Fitting Examples: (wrong) ARMA

- The resulting model is very different than the theoretical model we simulated.
- MSE 1 = 99.62, MSE 2 = 99.55
- Second model has lower MSE  but it looks very suspicious because the p-values of the the two MA terms are not statistically significant.
- These are all signs of overfitting due to the additional parameters.
- We’ll do our best to avoid overfitting.

# Bias – Variance Tradeoff

- There are two types of errors when estimation is based on a sample of data using a mathematical model.
- Sampling error (variance) because the estimated model yields different results in a new sample.
- Model based error (bias) because the model that was fit is an inaccurate representation of reality.
- Unfortunately, the two errors are in conflict:
    - To reduce bias, we need a model that yields a closer fit to the training sample. This, in general, means more complicated models with a larger number of parameters.
    - But complicated models generate more sampling errors when tested out of sample. They are an excellent representation of the training sample but do not necessarily perform well in other samples from the same population. This is the problem of overfitting.
- There is a need to find the right trade-off between model complexity and variance.

# Bias – Variance Tradeoff: Information Criteria

## Information Criteria

Akaike's Information Criterion (AIC), which was useful in selecting predictors for regression, is also useful for determining the order of an ARIMA model. It can be written as

$$\text{AIC} = -2\log(L) + 2(p+q+k+1),$$

where $L$ is the likelihood of the data, $k=1$ if $c \neq 0$ and $k=0$ if $c=0$. Note that the last term in parentheses is the number of parameters in the model (including $\sigma^2$, the variance of the residuals).

For ARIMA models, the corrected AIC can be written as

$$\text{AICc} = \text{AIC} + \frac{2(p+q+k+1)(p+q+k+2)}{T-p-q-k-2},$$

and the Bayesian Information Criterion can be written as

$$\text{BIC} = \text{AIC} + [\log(T) - 2](p+q+k+1).$$

Good models are obtained by minimising the AIC, AICc or BIC. Our preference is to use the AICc.

from Hyndman and Athanasapoulos

# Bias – Variance Tradeoff: Information Criteria

- We must always keep an eye on AIC and BIC measures.
- But they are not always conclusive.
- For the synthetic AR-1 example, model 1 has slightly lower AIC than model 2.
- We will also have to validate out-of-sample (more on this later).
  - Fit the model on training data
  - Evaluate performance on a separate test set.

# Example: Australian Beer Production

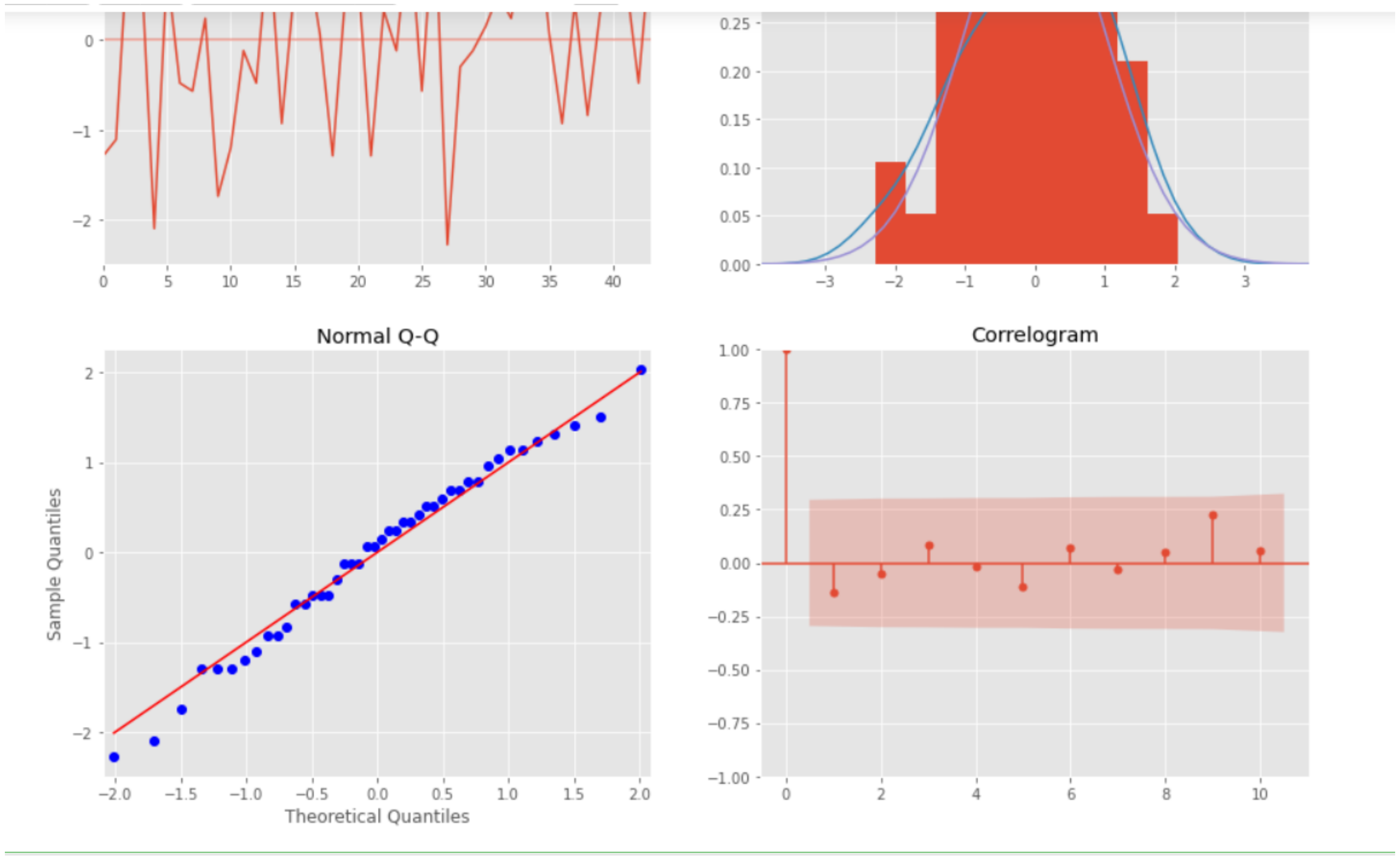
- Let’s fit a model a very simple model first: SARIMA(0,0,0)(0,1,0,12). What is this?

In [44]:
```python
# Fit the model
modbeer = sm.tsa.statespace.SARIMAX(prod,  trend='c', order=(0,0,0), seasonal_order=(0,1,0,12))
res = modbeer.fit(disp=False)
print(res.summary())
```

```
                                 SARIMAX Results
================================================================================
Dep. Variable:                        y   No. Observations:                   56
Model:             SARIMAX(0, 1, 0, 12)   Log Likelihood                -168.421
Date:                  Tue, 01 Mar 2022   AIC                            340.842
Time:                          11:43:15   BIC                            344.411
Sample:                               0   HQIC                           342.166
                                   - 56
Covariance Type:                    opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -2.6818      1.725     -1.554      0.120      -6.063       0.700
sigma2       123.6715     31.796      3.889      0.000      61.352     185.991
===================================================================================
Ljung-Box (L1) (Q):                   0.94   Jarque-Bera (JB):                 1.14
Prob(Q):                              0.33   Prob(JB):                         0.57
Heteroskedasticity (H):               0.66   Skew:                            -0.29
Prob(H) (two-sided):                  0.42   Kurtosis:                         2.46
===================================================================================
```

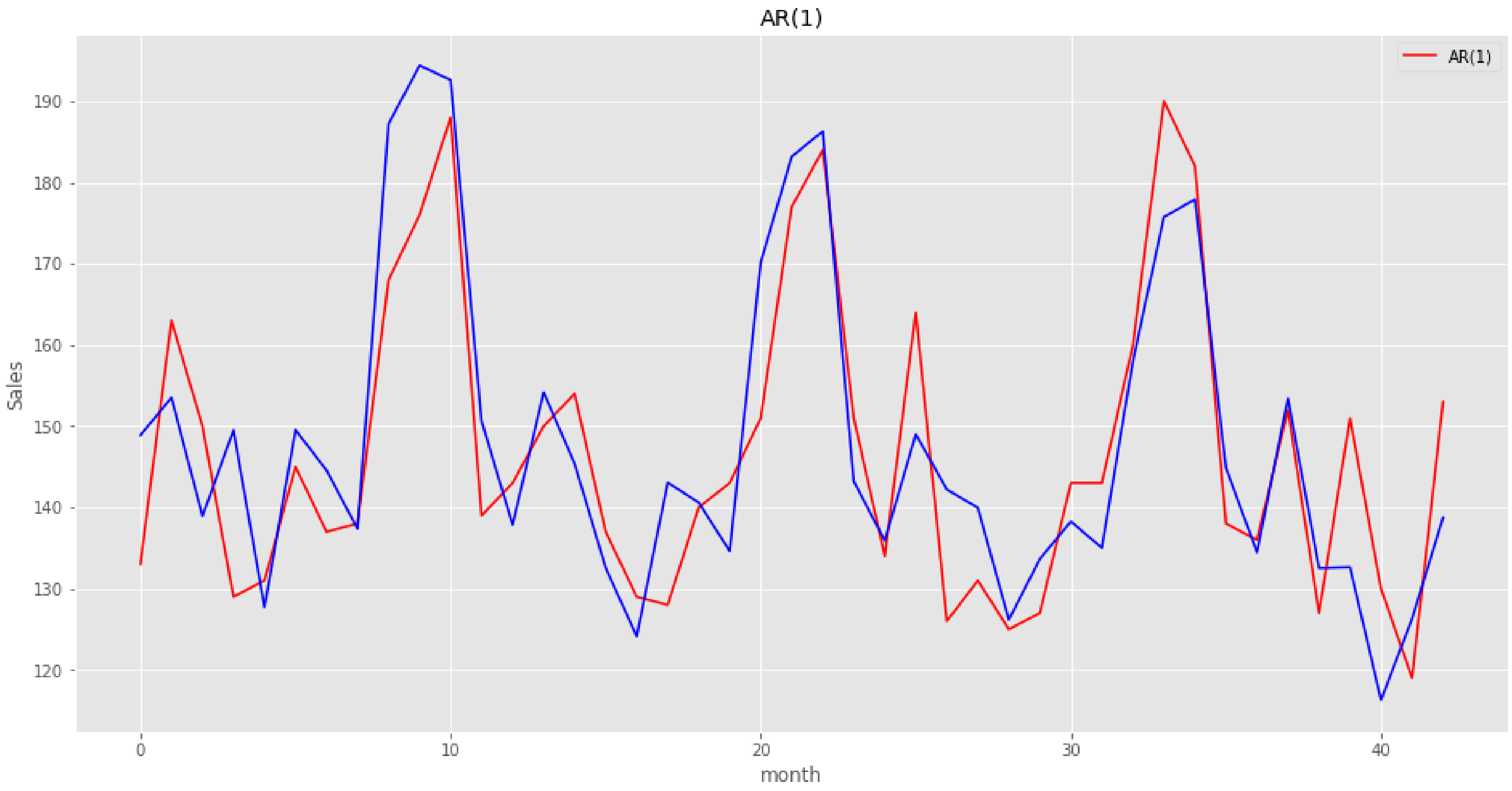
# Example: Australian Beer Production

- The residuals look fine. MSE =5742.86, $\sigma^2$ = 123.67 and AIC = 340.84



# Example: Australian Beer Production

- In-sample predictions



# Example: Australian Beer Production

- Let's fit a model a more complicated model now: SARIMA(1,0,1)(1,1,1,12).

```
In [51]: # Fit more complicated  model
         mod2beer = sm.tsa.statespace.SARIMAX(prod,  trend='c', order=(1,0,1), seasonal_order=(1,1,1,12))
         res2 = mod2beer.fit(disp=False)
         print(res2.summary())
```

```
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                  y   No. Observations:                   56
Model:             SARIMAX(1, 0, 1)x(1, 1, 1, 12)   Log Likelihood                -165.104
Date:                            Fri, 04 Mar 2022   AIC                            342.208
Time:                                    18:21:33   BIC                            352.913
Sample:                                         0   HQIC                           346.178
                                             - 56
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept     -1.8745      2.813     -0.666      0.505      -7.388       3.639
ar.L1          0.0019      0.733      0.003      0.998      -1.434       1.438
ma.L1         -0.2621      0.733     -0.358      0.721      -1.698       1.174
ar.S.L12       0.2807      0.709      0.396      0.692      -1.108       1.669
ma.S.L12      -0.9990    279.578     -0.004      0.997    -548.962     546.964
sigma2        78.4543   2.19e+04      0.004      0.997   -4.28e+04     4.3e+04
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 1.30
Prob(Q):                              0.98   Prob(JB):                         0.52
Heteroskedasticity (H):               0.74   Skew:                            -0.02
Prob(H) (two-sided):                  0.57   Kurtosis:                         2.16
===================================================================================
```

# Example: Australian Beer Production

- The residuals look fine. MSE =5726, $\sigma^2$ = 76.45 and AIC = 342.208
- MSE is significantly down but AIC is up. Several coefficients are statistically insignificant.
- The simple model is probably good enough. But the gold standard is to fit the model on a training set and compute the error performance of the fitted model on a test set.
    - This data set is too small to do that meaningfully.

# Example: Australian Beer Production

- We can further automate and check all reasonable models.

```
dit    View    Insert    Cell    Kernel    Widgets    Help
Run    Code
```

```python
            print('ARIMA{}x{}12 - MSE:{}'.format(param, param_seasonal, res
        except:
            continue
```

```
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:495.1900447210366
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - MSE:377.4257015306122
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:470.61846350607834
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - MSE:249.20605293612795
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:340.8422562077771
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - MSE:5742.864226682409
ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:339.031806512438
ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - MSE:5732.419990512659
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:448.87816765219793
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - MSE:195.9986073929865
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:453.8398530740862
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - MSE:199.22391780190267
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:339.5842638472778
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - MSE:5734.651132583305
ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:340.62545794741453
ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - MSE:5731.001823409475
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:485.13384605969276
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - MSE:303.665960812161
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:467.59381714847
```

# Example: Australian Beer Production

- But it is more sound to perform a more comprehensive check among all reasonable models rather than numerically looking at the AIC.
  - Check p-values of coefficients
  - Remove coefficients with high p-values, rerun the model with reduced parameters
  - Always test out-of sample.