INDR 422/522

Fikri Karaesmen

Spring 2023

ARIMA Forecasts-4, Regression

March 28, 2023

Reminders

- Third lab available, please take a look and work on the exercises
- Fourth lab will be available this Friday
- Participation taken. Please participate in polls.
- HW 1 (due-date March 31, 2023)
- So far, we used material from Chapters 2, 5, 8 and 9 of Athanasopoulos and Hyndman’s book: Forecasting Principles and Practice
    - Link on blackboard:
    - New edition: https://otexts.com/fpp3/

Class Exercise from last lecture

CLASS EXERCISE, March 23, 2023

1. Take the time series Y_t and let $W_t = Y_t - Y_{t-1}$ Consider an ARIMA(1,1,0) model for Y_t. This is equal to:

   Solution: ARIMA(1,1,0) uses 1 order of differencing (middle term) and an 1 AR term after differencing (first term). We therefore have:

   $$W_t = c + \phi_1 W_{t-1} + \epsilon_t$$

   Reverting the transformation, we have:

   $$Y_t = W_t + Y_{t-1} = c + \phi_1 W_{t-1} + \epsilon_t$$

   We can simplify further to:

   $$Y_t = c + Y_{t-1} + \phi_1(Y_{t-1} - Y_{t-2}) + \epsilon_t = c + (1 + \phi_1)Y_{t-1} - \phi_1 Y_{t-2} + \epsilon_t$$

2. An ARIMA (0,2,0) model for Y_t corresponds to:

   Solution: ARIMA(0,2,0) uses 2 orders of differencing (middle term) but no AR and MA terms. We have $W_t = Y_t - Y_{t-1}$ and $U_t = W_t - W_{t-1}$ and

   $$U_t = c + \epsilon_t$$

   We can then revert the transformations: $W_t = U_t + W_{t-1}$ and $Y_t = W_t + Y_{t-1}$

   $$Y_t = Y_{t-1} + U_t + W_{t-1} = Y_{t-1} + c + \epsilon_t + (Y_{t-1} - Y_{t-2})$$

   Simplifying, we obtain:

   $$Y_t = c + 2Y_{t-1} - Y_{t-2} + \epsilon_t$$

Class Exercise from last lecture

3. The backshift notation representation for ARIMA(1,3,0) is:

   Solution: The representation is:

$$(1-\phi_1 B)(1-B)^3 y_t = c + \epsilon_t$$

4. In backshift notation, $Y_t = c + 2Y_{t-1} + Y_{t-2} + \epsilon_t$ corresponds to:

   Solution:

$$Y_t = c + 2BY_t + B^2 Y_t + \epsilon_t$$

   or

$$Y_t - 2BY_t + B^2 Y_t = (1-B)^2 Y_t = c\epsilon_t$$

Summary last lectures

- Software is able to fit (i.e. estimate the coefficients of) ARIMA models
- It is easy to experiment with different models
- Overfitting is an issue

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.

Overfitting: some introduction

- ARIMA models and their software implementation enable us to test and implement alternative models with different parameters.
- Using more parameters (more AR and MA terms) increases the degrees of freedom and therefore increases the model fitting error performance (MSE etc.) on the given data.
- But by using too many parameters we might be overfitting the model to the particular (training) sample.
- We should be aware of this and take caution.

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 σ^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

Model Fitting Examples

```
7]: # Fit the model
modar = sm.tsa.statespace.SARIMAX(y_ar[100:499], trend='c', order=(1,0,0))
res = modar.fit(disp=False)
print(res.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                      y   No. Observations:                  399
Model:               SARIMAX(1, 0, 0)   Log Likelihood               -1458.647
Date:                Tue, 01 Mar 2022   AIC                           2923.295
Time:                        10:09:29   BIC                           2935.261
Sample:                             0   HQIC                          2928.034
                                - 399
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
intercept    293.9478     33.949      8.658      0.000     227.408     360.487
ar.L1          0.6470      0.041     15.866      0.000       0.567       0.727
sigma2        87.7254      6.350     13.815      0.000      75.279     100.172
===================================================================================
Ljung-Box (L1) (Q):                   0.00   Jarque-Bera (JB):                 3.89
Prob(Q):                              0.95   Prob(JB):                         0.14
Heteroskedasticity (H):               1.44   Skew:                            -0.24
Prob(H) (two-sided):                  0.04   Kurtosis:                         3.01
===================================================================================
```

The fitted model is $Y_t = 293 + 0.64\ Y_{t-1} + \varepsilon_t$

The true model was $Y_t = 250 + 0.7\ Y_{t-1} + \varepsilon_t$ and $\sigma^2 = 100$.

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]:
```
# 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
- AIC 1 = 2923, AIC 2 = 2976
    - Smaller AIC is better
- 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: 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?

Run Code

```
In [44]: # 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
===================================================================================
```

MSE Beer Seasonal = 5742.864
RMSE Beer Seasonal = 75.782
MAE Beer Seasonal = 41.713
MAPE Beer Seasonal = 0.268
AIC = 340.842

Model Fitting Examples: Aus Beer Data

```
In [73]: # let's also take regular difference and see if it improves the predictions

mod_beer2 = sm.tsa.statespace.SARIMAX(prod, trend ='c',  order=(0,1,0) , seasonal_order=(0,1,0,12))
res_beer2 = mod_beer2.fit(disp=False)
res_beer2.summary()
```

Model:	SARIMAX(0, 1, 0)x(0, 1, 0, 12)	**Log Likelihood**	-182.159
Date:	Wed, 01 Mar 2023	**AIC**	368.318
Time:	15:36:14	**BIC**	371.840
Sample:	0	**HQIC**	369.617
	- 56		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.6279	2.658	0.236	0.813	-4.581	5.837
sigma2	279.9546	72.652	3.853	0.000	137.559	422.350

Ljung-Box (L1) (Q):	13.20	**Jarque-Bera (JB):**	1.29
Prob(Q):	0.00	**Prob(JB):**	0.52
Heteroskedasticity (H):	0.47	**Skew:**	-0.34

Seasonal differences and first differences

MSE Beer Seasonal with Diff= 1071.995
RMSE Beer Seasonal with Diff = 32.741
MAE Beer Seasonal with Diff= 18.854
MAPE Beer Seasonal with Diff= 0.127
AIC = 368.318

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. AIC = 342.208.
- MSE is smaller 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

```
            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:485.1338460596927б
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 sounder 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.
- We’ll dig into this more with other methods.

Overview of ARIMA models

- An efficient class of models to capture the auto-correlation structure in the data.
- They are effective on stationary data:
  - But ARIMA framework incorporates differencing
  - And SARIMA also incorporates seasonal differencing
- Model fitting (i.e. Finding the AR and MA coefficient that best fit the sample) through Maximum Likelihood Estimation
  - More robust in larger samples
- Adding a new term always increases likelihood (more degrees of freedom in optimization)

Overview of ARIMA models

- Adding a new term always increases likelihood (more degrees of freedom in optimization) but we must be cautious of overfitting.
  - Check the statistical significance (p-value) of the fitted coefficients
  - Check AIC, BIC etc.
- Ideally, fit the model on part of the data (training) and test its error performance on a separate part (test).
  - Ensure that the error does not worsen by much on the test set.

Overview of ARIMA models

- Note that we are not looking for causality but are using the model for making predictions.
- We don’t deeply question the auto-correlation structure of the model
  - Apart from basic things like seasonality and trend etc.
- Note that we can run ARIMA models on top of other unbiased forecasts.
  - Run a double exponential smoothing forecast, check the residuals, if the residuals are auto-correlated then fit an ARIMA model to the residuals.

Overview of ARIMA models

- Recall the introductory lecture, our eventual objective is to connect predictions to prescriptions.
- The predicted mean $\hat{y}_t$ is an important part of the planning process.
- But what makes the prescriptive problem is interesting and challenging is usually the error term ϵ_t.

$$
\begin{aligned}
Y_t &= c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + \\
&\quad + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + ... + \theta_q \epsilon_{t-q} + \epsilon_t
\end{aligned}
$$

- This is why we emphasize the residuals and their distribution.

Regression for Time Series

- Linear regression is a general tool that looks for a linear relationship between a response and predictors.
- We have observations at different levels of the predictors and the corresponding response.
- The goal is to have predictions for the response that will be generated by so far unobserved levels of the predictors.

- We'll look into time series data where the data is the time series itself. The prediction is then typically a forecast for future demand, prices etc.

Regression for Time Series

- Consider the following linear model:

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \ldots + \beta_n x_{nt} + \epsilon_t$$

- y_t is the forecast and x_{kt} are the predictors.
- We are therefore looking for a linear relationship between the predictors and the response (the forecast).
- Note that in the setting of forecasting, this is somewhat different than designing a controlled experiment where we can control the levels of the predictors. The predictors that are available to us cannot be controlled in general.

Regression for Time Series

- We make the first assumption that the response is approximately a linear function of the predictors.
- We also have to assume that the errors ϵ_t:
    - have mean zero; otherwise the forecasts will be systematically biased.
    - are not autocorrelated; otherwise the forecasts will be inefficient, as there is more information in the data that can be exploited. item they are unrelated to the predictor variables; otherwise we could have an additional predictor in the model that explains the relationship between the error and the forecast
    - are hopefully normally distributed with constant variance.

Regression for Time Series

- The ordinary least squares regression finds the parameters $\beta_0, \beta_1, ..., \beta_n$ to minimize:

$$\sum_{t=1}^{T} \epsilon_t^2 = \sum_{t=1}^{T} (y_t - (\beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + ... + \beta_n x_{nt}))^2$$

- The above is an unconstrained convex optimization problem. In addition, the derivative with respect to each parameter β_k of the objective function is a linear function.
- Finding the minimizer then boils down to solving $n+1$ linear equations in $n+1$ unknowns.

Regression for Time Series

- Finding the minimizer then boils down to solving $n+1$ linear equations in $n+1$ unknowns.
- We can therefore easily find the coefficients $\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_n$ that minimize the total square error.
- To have a prediction, we can then use:

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1t} + ... + \hat{\beta}_n x_{nt}$$

Regression for Time Series: Goodness of Fit

- We measure the goodness of fit by the coefficient of determination R^2:

$$R^2 = \frac{\sum_t (y_t - \bar{y})^2 - \sum_t (y_t - \hat{y}_t)^2}{\sum_t (y_t - \bar{y})^2} = \frac{\sum_t (\hat{y}_t - \bar{y})^2}{\sum_t (y_t - \bar{y})^2}$$

- Note that $R^2 = Corr(Y, \hat{Y})^2$ the square of the correlation between the predictions and the data. The least squares optimization leads to the parameters that maximizes the correlation.
- We'll see that while R^2 is an important measure, we cannot rely on it completely without additional checks.

Regression for Time Series: Goodness of Fit

- RMSE is another measure of the goodness of fit. Since multiple parameters are estimated, we correct the RMSE for the degrees of freedom to estimate the standard deviation of the residuals:

$$\hat{\sigma}_e = \sqrt{\frac{\sum \epsilon_t^2}{T - n - 1}}$$

- We'll use $\hat{\sigma}_e$ to build confidence intervals.

Regression for Time Series: Goodness of Fit

- By design of the least squares optimization problem, linear regression yields unbiased estimators. The errors and the predictors are also uncorrelated.
- But we have seen that auto-correlation is an issue and that remains an issue for the error terms which may be auto-correlated in time.
- We should also be concerned about using as predictors other time series that have a similar pattern to the series we would like to predict.

Regression for Time Series: Spurious Correlations

- Any two data series with a similar pattern (trend/seasonality etc.) are likely to be correlated. It's very easy to find wrong (spurious) relationships.

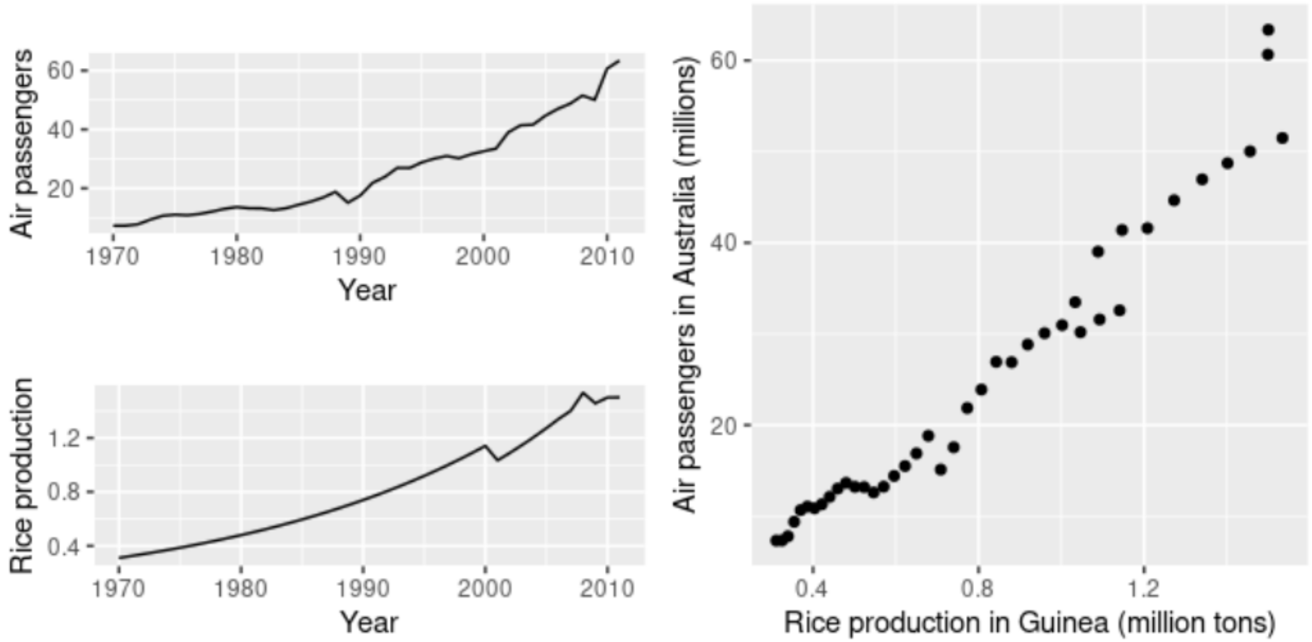

Figure 5.12: Trending time series data can appear to be related, as shown in this example where air passengers in Australia are regressed against rice production in Guinea.

Regression for Time Series: Basic Predictors

- Here are some basic predictors that can capture the patterns in the data:
- To capture simple linear trend, we can use:

$$y_t = \beta_0 + \beta_1 t + \epsilon_t$$

- We'll see that non-linear trends can also be handled, for instance:

$$y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \epsilon_t$$

Regression for Time Series: Basic Predictors: Google Share Price

- The Google Share Price Data has a strong trend.
- Let’s try a simple trend based regression.

$$y_t = \beta_0 + \beta_1 t + \epsilon_t$$

	Price	Day
1	2064.879883	1
2	2070.860107	2
3	2095.169922	3
4	2031.359985	4
5	2036.859985	5

Regression for Time Series: Basic Predictors: Google Share Price

$$y_t = \beta_0 + \beta_1 t + \epsilon_t$$

In [3]:
```python
lm = sm.OLS.from_formula('Price ~ Day', df)
result = lm.fit()
result.summary()
```

Out[3]:

OLS Regression Results

Dep. Variable:	Price	**R-squared:**	0.712
Model:	OLS	**Adj. R-squared:**	0.711
Method:	Least Squares	**F-statistic:**	619.5
Date:	Tue, 08 Mar 2022	**Prob (F-statistic):**	9.84e-70
Time:	17:39:05	**Log-Likelihood:**	-1633.4
No. Observations:	253	**AIC:**	3271.
Df Residuals:	251	**BIC:**	3278.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2197.6246	19.501	112.694	0.000	2159.218	2236.031
Day	3.3131	0.133	24.890	0.000	3.051	3.575

Omnibus:	13.705	**Durbin-Watson:**	0.068
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	15.010
Skew:	-0.590	**Prob(JB):**	0.000550
Kurtosis:	2.818	**Cond. No.**	294.

$$\hat{\beta}_0 = 2197.62, \hat{\beta}_0 = 3.31$$

Regression for Time Series: Basic Predictors: Google Share Price

- Since the execution is very easy, we are tempted to try other predictors, let us try:

$$y_t = \beta_0 + \beta_1 t + \beta_2 \sqrt{t} + \beta_3 t^2 + \epsilon_t$$

Out[2]:

	Price	Day	Sqrtd	Sqrd
1	2064.879883	1	1.000000	1
2	2070.860107	2	1.414214	4
3	2095.169922	3	1.732051	9
4	2031.359985	4	2.000000	16
5	2036.859985	5	2.236068	25

In [12]:
```
lm2 = sm.OLS.from_formula('Price ~ Day + Sqrtd+ Sqrd', df)
result2 = lm2.fit()
```

Let us call this model Model 2.

Regression for Time Series: Basic Predictors: Google Share Price

13]:

$$y_t = \beta_0 + \beta_1 t + \beta_2 \sqrt{t} + \beta_3 t^2 + \epsilon_t$$

OLS Regression Results

Dep. Variable:	Price	**R-squared:**	0.937
Model:	OLS	**Adj. R-squared:**	0.936
Method:	Least Squares	**F-statistic:**	1236.
Date:	Tue, 08 Mar 2022	**Prob (F-statistic):**	3.40e-149
Time:	12:17:45	**Log-Likelihood:**	-1440.8
No. Observations:	253	**AIC:**	2890.
Df Residuals:	249	**BIC:**	2904.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2207.3790	43.714	50.496	0.000	2121.282	2293.476
Day	19.8417	1.304	15.219	0.000	17.274	22.409
Sqrtd	-111.2141	14.968	-7.430	0.000	-140.695	-81.733
Sqrd	-0.0432	0.002	-18.802	0.000	-0.048	-0.039

Omnibus:	9.814	**Durbin-Watson:**	0.308
Prob(Omnibus):	0.007	**Jarque-Bera (JB):**	9.833
Skew:	-0.450	**Prob(JB):**	0.00733
Kurtosis:	3.352	**Cond. No.**	2.90e+05

Regression for Time Series: Basic Predictors: Google Share Price

$$y_t = \beta_0 + \beta_1 t + \beta_2 \sqrt{t} + \beta_3 t^2 + \epsilon_t$$

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2207.3790	43.714	50.496	0.000	2121.282	2293.476
Day	19.8417	1.304	15.219	0.000	17.274	22.409
Sqrtd	-111.2141	14.968	-7.430	0.000	-140.695	-81.733
Sqrd	-0.0432	0.002	-18.802	0.000	-0.048	-0.039

- Note that $\hat{\beta}_2$ and $\hat{\beta}_3$ are also statistically significant.