INDR 450/550

Spring 2022

Lecture 9: Regression for Time Series

March 14, 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 three labs were uploaded. Please follow them.
  - Next HW based on lab2

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.

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

- 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 $\beta_0, \beta_1, \ldots, \beta_n$ to minimize:

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

- 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.

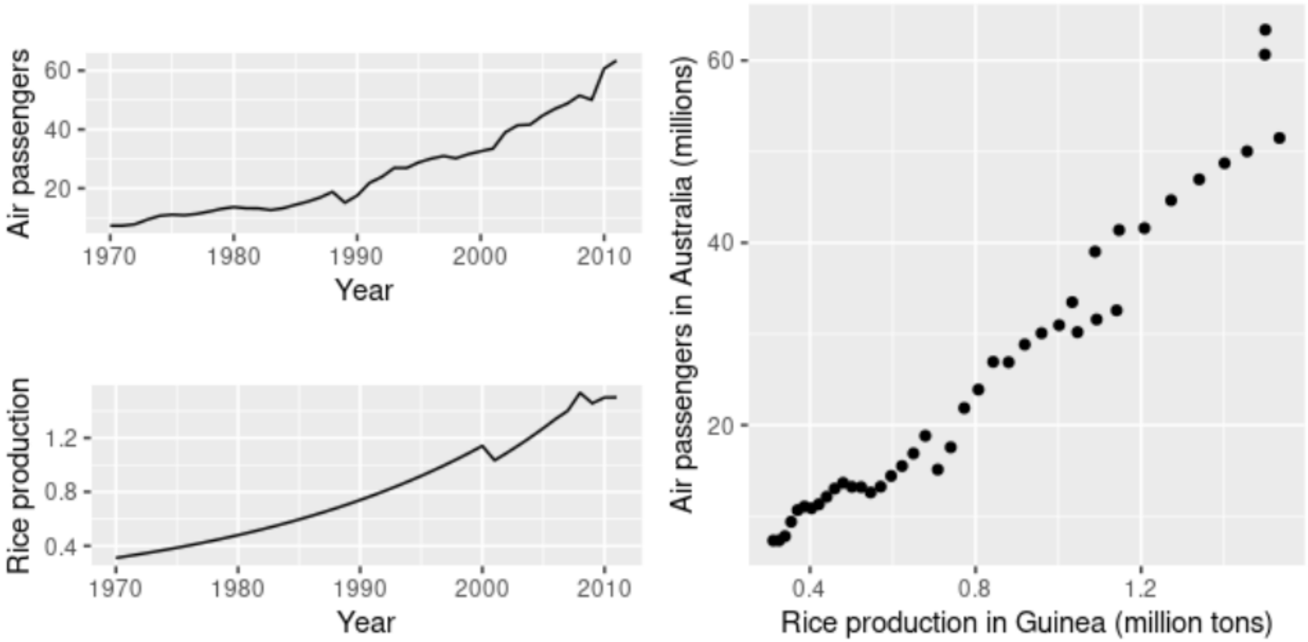

Figure 5.12: Trending time series data can appear to be related, as shown in this example where air passengers in Australia are regressed against rice production in Guinea.

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]: 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()
```

Regression for Time Series: Basic Predictors: Google Share Price

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

13]:

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