I'm not a bot

Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical test when it comes to analyzing the stationary of a series. 1. Introduction In ARIMA time series forecasting, the first step is to determine the number of differencing required to make the series stationary. Since testing the stationarity of a time series is a frequently performed activity in autoregressive models, the ADF test along with KPSS test is something that you need to be fluent in when performing time series analysis. Another point to remember is the ADF test is fundamentally a statistical significance test. That means, there is a hypothesis testing involved with a null and alternate hypothesis and as a result a test statistic is computed and p-values get reported. It is from the test statistic and the p-value, you can make an inference as to whether a given series is stationary or not. So, how exactly does the ADF test work? let's see the mathematical intuition behind the test with clear examples. Let's begin. 2. What is a Unit Root Test? The ADF test belongs to a category of tests called 'Unit Root Test', which is the proper method for testing the stationarity of a time series. So what does a 'Unit Root' mean? Unit root is a characteristic of a time series that makes it non-stationary. Technically speaking, a unit root is said to exist in a time series of the value of alpha = 1 in the below equation. where, Yt is the value of the time series at time 't' and Xe is an exogenous variable (a separate explanatory variable, which is also a time series). What does this mean to us? The presence of a unit root means the time series is non-stationary. Besides, the number of unit roots contained in the series corresponds to the number of differencing operations required to make the series stationary. Alright, let's come back to topic. 3. Dickey-Fuller Test Before going into ADF test, let's first understand what is the Dickey-Fuller test. A Dickey-Fuller test is a unit root test that tests the null hypothesis that α=1 in the following model equation. alpha is the coefficient of the first lag on Y. Null Hypothesis (H0): alpha=1 where, y(t-1) = lag 1 of time series delta Y(t-1) = first difference of the series at time (t-1) Fundamentally, it has a similar null hypothesis as the unit root test. That is, the coefficient of Y(t-1) is 1, implying the presence of a unit root. If not rejected, the series is taken to be non-stationary. The Augmented Dickey-Fuller test evolved based on the above equation and is one of the most common form of Unit Root test. 4. How does Augmented Dickey Fuller (ADF) Test work? As the name suggest, the ADF test is an 'augmented' version of the Dickey Fuller test. The ADF test expands the Dickey-Fuller test equation to include high order regressive process in the model. If you notice, we have only added more differencing terms, while the rest of the equation remains the same. This adds more thoroughness to the test. The null hypothesis however is still the same as the Dickey Fuller test. A key point to remember here is: Since the null hypothesis assumes the presence of unit root, that is α=1, the p-value obtained should be less than the significance level (say 0.05) in order to reject the null hypothesis. Thereby, inferring that the series is stationary. However, this is a very common mistake analysts commit with this test. That is, if the p-value is less than significance level, people mistakenly take the series to be non-stationary. 5. ADF Test in Python So, how to perform a Augmented Dickey-Fuller test in Python? The statsmodel package provides a reliable implementation of the ADF test via the adfuller() function in statsmodels.tsa.stattools. It returns the following outputs: The p-value The value of the test statistic Number of lags considered for the test The critical value cutoffs. When the test statistic is lower than the critical value shown, you reject the null hypothesis and infer that the time series is stationary. Alright, let's run the ADF test on the a10 dataset from the fpp package from R. This dataset counts the total monthly scripts for pharmaceutical products falling under ATC code A10. The original source of this dataset is the Australian Health Insurance Commission. As see earlier, the null hypothesis of the test is the presence of unit root, that is, the series is non-stationary. # Setup and Import data from statsmodels.tsa.stattools import adfuller import pandas as pd import numpy as np %matplotlib inline url = ' df = pd.read_csv(url, parse_dates=['date'], index_col='date') series = df.loc[:, 'value'].values df.plot(figsize=(14,8), legend=None, title='a10 - Drug Sales Series'); The packages and the data is loaded, we have everything needed to perform the test using adfuller(). An optional argument the adfuller() accepts is the number of lags you want to consider while performing the OLS regression. By default, this value is 12*(nobs/100)^{1/4}, where nobs is the number of observations in the series. But, optionally you can specify either the maximum number of lags with maxlags parameter or let the algorithm compute the optimal number iteratively. This can be done by setting the autolag='AIC'. By doing so, the adfuller will choose a the number of lags that yields the lowest AIC. This is usually a good option to follow. # ADF Test result = adfuller(series, autolag='AIC') print(f'ADF Statistic: {result[0]}') print(f'n_lags: {result[1]}') print(f'p-value: {result[1]}') for key, value in result[4].items(): print('Critial Values:') print(f' {key}, {value}') Result: ADF Statistic: 3.1451856893067296 n_lags: 1.0 p-value: 1.0 Critial Values: 1%, -3.465620397124192 Critial Values: 5%, -2.8770397560752436 Critial Values: 10%, -2.5750324547306476 The p-value is obtained is greater than significance level of 0.05 and the ADF statistic is higher than any of the critical values. Clearly, there is no reason to reject the null hypothesis. So, the time series is in fact non-stationary. 6. ADF Test on stationary series Now, let's see another example of performing the test on a series of random numbers which is usually considered as stationary. Let's use np.random.randn() to generate a randomized series. # ADF test on random numbers series = np.random.randn(100) result = adfuller(series, autolag='AIC') print(f'ADF Statistic: {result[0]}') print(f'p-value: {result[1]}') for key, value in result[4].items(): print('Critial Values:') print(f' {key}, {value}') Result: ADF Statistic: -7.4715740767231456 p-value: 5.0386184272419386e-11 Critial Values: 1%, -3.4996365338407074 Critial Values: 5%, -2.8918307730370025 Critial Values: 10%, -2.5829283377617176 The p-value is very less than the significance level of 0.05 and hence we can reject the null hypothesis and take that the series is stationary. Let's visualise the series as well to confirm. import matplotlib.pyplot as plt %matplotlib inline fig, axes = plt.subplots(figsize=(10,7)) plt.plot(series); plt.title('Random'); 7. Conclusion We saw how the Augmented Dickey Fuller Test works and how to perform it using statsmodels. Now given any time series, you should be in a position to perform the ADF Test and make a fair inference on whether the series is stationary or not. In the next one we'll see how to perform the KPSS test. This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page If a series is nonstationary, conventional hypothesis tests, confidence intervals and forecasts can be strongly misleading. The assumption of stationarity is violated if a series exhibits trends or breaks and the resulting complications in an econometric analysis depend on the specific type of the nonstationarity. This section focuses on time series that exhibit trends. A series is said to exhibit a trend if it has a persistent long-term movement. One can distinguishes between deterministic and stochastic trends as: A trend is deterministic if it is a nonrandom function of time. A trend is said to be stochastic if it is a random function of time. The figures we have produced in Chapter 14.2 reveal that many economic time series show a trending behavior that is probably best modeled by stochastic trends. This is why the book focuses on the treatment of stochastic trends. The simplest way to model a time series \(Y_t\) that has stochastic trend is the random walk \[\begin{align} Y_t = Y_{t-1} + u_t, \tag{14.6} \end{align}\] where the \(u_t\) are i.i.d. errors with \(E(u_t\vert Y_{t-1}, Y_{t-2}, \dots) = 0\). Note that \[\begin{align*} E(Y_t\vert Y_{t-1}, Y_{t-2}\dots) =& \, E(Y_{t-1}\vert Y_{t-1}, Y_{t-2}\dots) + E(u_t\vert Y_{t-1}, Y_{t-2}\dots) \\ =& \, Y_{t-1} \end{align*}\] so the best forecast for \(Y_t\) is yesterday's observation \(Y_{t-1}\). Hence the difference between \(Y_t\) and \(Y_{t-1}\) is unpredictable. The path followed by \(Y_t\) consists of random steps \(u_t\), hence it is called a random walk. Assume that \(Y_0\), the starting value of the random walk is \(0\). Another way to write (14.6) is \[\begin{align*} Y_0 =& \, 0 \\ Y_1 =& \, 0 + u_1 \\ Y_2 =& \, 0 + u_1 + u_2 \\ \vdots & \, \\ Y_t =& \, \sum_{i=1}^t u_i. \end{align*}\] Therefore we have \[\begin{align*} Var(Y_t) =& \, Var(u_1 + u_2 + \dots + u_t) \\ =& \, t \sigma_u^2. \end{align*}\] Thus the variance of a random walk depends on \(t\) which violates the assumption presented in Key Concept 14.5: a random walk is nonstationary. Obviously, (14.6) is a special case of an AR(\(1\)) model where \(\beta_1 = 1\). One can show that a time series that follows an AR(\(1\)) model is stationary if \(\lvert\beta_1\rvert < 1\). In a general AR(\(p\)) model, stationarity is linked to the roots of the polynomial \[1-\beta_1 z - \beta_2 z^2 - \beta_3 z^3 - \dots - \beta_p z^p.\] If all roots are greater than \(1\) in absolute value, the AR(\(p\)) series is stationary. If at least one root equals \(1\), the AR(\(p\)) is said to have a unit root and thus has a stochastic trend. It is straightforward to simulate random walks in R using arima.sim(). The function matplot() is convenient for simple plots of the columns of a matrix. # simulate and plot random walks starting at 0 set.seed(1) RWs |t|) #> (Intercept) -3.459488 0.3635104 -9.516889 1.354156e-15 #> L(RWs[, 3]) 1.047195 0.1450874 7.217687 1.135828e-10 The result is obviously spurious: the coefficient on \(Green_{t-1}\) is estimated to be about \(1\) and the \(p\)-value of \(1.14 \cdot 10^{-10}\) of the corresponding \(t\)-test indicates that the coefficient is highly significant while its true value is in fact zero. As an empirical example, consider the U.S. unemployment rate and the Japanese industrial production. Both series show an upward trending behavior from the mid-1960s through the early 1980s. # plot U.S. unemployment rate & Japanese industrial production plot(merge(as.zoo(USUnemp), as.zoo(JPIndProd)), plot.type = "single", col = c("darkred", "steelblue"), lwd = 2, xlab = "Date", ylab = "", main = "Spurious Regression: Macroeconomic Time series") # add a legend legend("topleft", legend = c("USUnemp", "JPIndProd"), col = c("darkred", "steelblue"), lwd = c(2, 2)) # estimate regression using data from 1962 to 1985 SR_Unemp1 #> t test of coefficients: #> #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) -2.37452 1.12041 -2.1193 0.0367 * #> ts(JPIndProd["1962::1985"]) 2.22057 0.29233 7.5961 2.227e-11 *** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 A simple regression of the U.S. unemployment rate on Japanese industrial production using data from 1962 to 1985 yields \[\begin{align} \widehat{U.S. UR}_t = -\underset{(1.12)}{2.37} + \underset{(0.29)}{2.22} \log(JapaneseIP_t). \tag{14.8} \end{align}\] This appears to be a significant relationship: the \(t\)-statistic of the coefficient on \(\log(JapaneseIP_t)\) is bigger than 7. # Estimate regression using data from 1986 to 2012 SR_Unemp2 #> t test of coefficients: #> #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) 41.7763 5.4066 7.7270 6.596e-12 *** #> ts(JPIndProd["1986::2012"]) -7.7771 1.1714 -6.6391 1.386e-09 *** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 When estimating the same model, this time with data from 1986 to 2012, we obtain \[\begin{align} \widehat{U.S. UR}_t = \underset{(5.41)}{41.78} -\underset{(1.17)}{7.78} \log(JapaneseIP)_t \tag{14.9} \end{align}\] which surprisingly is quite different. (14.8) indicates a moderate positive relationship, in contrast to the large negative coefficient in (14.9). This phenomenon can be attributed to stochastic trends in the series: since there is no economic reasoning that relates both trends, both regressions may be spurious. A formal test for a stochastic trend has been proposed by Dickey and Fuller (1979) which thus is termed the Dickey-Fuller test. As discussed above, a time series that follows an AR(\(1\)) model with \(\beta_1 = 1\) has a stochastic trend. Thus, the testing problem is \[\begin{align*} H_0: \beta_1 = 1 \ \ \ \text{vs.} \ \ \ H_1: \lvert\beta_1\rvert < 1. \end{align*}\] The null hypothesis is that the AR(\(1\)) model has a unit root and the alternative hypothesis is that it is stationary. One often rewrites the AR(\(1\)) model by subtracting \(Y_{t-1}\) on both sides: \[\begin{align} Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t \ \ \Leftrightarrow \ \ \Delta Y_t = \beta_0 + \delta Y_{t-1} + u_t, \tag{14.10} \end{align}\] where \(\delta = \beta_1 - 1\). The testing problem then becomes \[\begin{align*} H_0: \delta = 0 \ \ \ \text{vs.} \ \ \ H_1: \delta < 0, \end{align*}\] which is convenient since the corresponding test statistic is reported by many relevant R functions.11 The Dickey-Fuller test can also be applied in an AR(\(p\)) model. The Augmented Dickey-Fuller (ADF) test is summarized in Key Concept 14.8. Consider the regression \[\begin{align} \Delta Y_t = \beta_0 + \delta Y_{t-1} + \gamma_1 \Delta Y_{t-1} + \gamma_2 \Delta Y_{t-2} + \dots + \gamma_p \Delta Y_{t-p} + u_t. \tag{14.11} \end{align}\] The ADF test for a unit autoregressive root tests the hypothesis \(H_0: \delta = 0\) (stochastic trend) against the one-sided alternative \(H_1: \delta < 0\) (stationarity) using the usual OLS \(t\)-statistic. If it is assumed that \(Y_t\) is stationary around a deterministic linear time trend, the model is augmented by the regressor \(t\): \[\begin{align} \Delta Y_t = \beta_0 + at + \delta Y_{t-1} + \gamma_1 \Delta Y_{t-1} + \gamma_2 \Delta Y_{t-2} + \dots + \gamma_p \Delta Y_{t-p+1} + u_t, \tag{14.12} \end{align}\] where again \(H_0: \delta = 0\) is tested against \(H_1: \delta < 0\). The optimal lag length \(p\) can be estimated using information criteria. In (14.11), \(p=0\) (no lags of \(\Delta Y_t\) are used as regressors) corresponds to a simple AR(\(1\)). Under the null, the \(t\)-statistic corresponding to \(H_0: \delta = 0\) does not have a normal distribution. The critical values can only be obtained from simulation and differ for regressions (14.11) and (14.12) since the distribution of the ADF test statistic is sensitive to the deterministic components included in the regression. Key Concept 14.8 states that the critical values for the ADF test in the regressions (14.11) and (14.12) can only be determined using simulation. The idea of the simulation study is to simulate a large number of ADF test statistics and use them to estimate quantiles of their asymptotic distribution. This section shows how this can be done using R. First, consider the following AR(\(1\)) model with intercept \[\begin{align*} Y_t =& \, \alpha + z_t, \ \ z_t = \rho z_{t-1} + u_t. \end{align*}\] This can be written as \[\begin{align*} Y_t =& \, (1-\rho) \alpha + \rho y_{t-1} + u_t, \end{align*}\] i.e., \(Y_t\) is a random walk without drift under the null \(\rho = 1\). One can show that \(Y_t\) is a stationary process with mean \(\alpha\) for \(\lvert\rho\rvert Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 The estimation yields \[\begin{align*} \Delta\log(GDP_t) =& \underset{(0.118)}{0.28} + \underset{(0.0001)}{0.0002 t} -\underset{(0.014)}{0.033} \log(GDP_{t-1}) \\ & + \underset{(0.113)}{0.083} \Delta \log(GDP_{t-1}) + \underset{(0.071)}{0.188} \Delta \log(GDP_{t-2}) + u_t, \end{align*}\] so the ADF test statistic is \(t=-0.033/0.014 = - 2.35\). The corresponding \(5\%\) critical value from Table 14.2 is \(-3.41\) so we cannot reject the null hypothesis that \(\log(GDP)\) has a stochastic trend in favor of the alternative that it is stationary around a deterministic linear time trend. The ADF test can be done conveniently using ur.df() from the package urca. # test for unit root in GDP using 'ur.df()' from the package 'urca' summary(ur.df(LogGDP, type = "trend", lags = 2, selectlags = "Fixed")) #> #> ### #> # Augmented Dickey-Fuller Test Unit Root Test # #> ### #> #> Test regression trend #> #> #> Call: #> lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag) #> #> Residuals: #> Min 1Q Median 3Q Max #> -0.025580 -0.004109 0.000321 0.004869 0.032781 #> #> Coefficients: #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) 0.2790086 0.1180427 2.364 0.019076 * #> z.lag.1 -0.0333245 0.0144144 -2.312 0.021822 * #> tt 0.0002382 0.0001109 2.148 0.032970 * #> z.diff.lag1 0.2708136 0.0697696 3.882 0.000142 *** #> z.diff.lag2 0.1876338 0.0705557 2.659 0.008476 ** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 #> #> Residual standard error: 0.007704 on 196 degrees of freedom #> Multiple R-squared: 0.1783, Adjusted R-squared: 0.1616 #> F-statistic: 10.63 on 4 and 196 DF, p-value: 8.076e-08 #> #> #> Value of test-statistic is: -2.3119 11.2558 4.267 #> #> Critical values for test statistics: #> 1pct 5pct 10pct #> tau3 -3.99 -3.43 -3.13 #> phi2 6.22 4.75 4.07 #> phi3 8.43 6.49 5.47 The first test statistic at the bottom of the output is the one we are interested in. The number of test statistics reported depends on the test regression. For type = "trend", the second statistics corresponds to the test that there is unit root and no time trend while the third one corresponds to a test of the hypothesis that there is a unit root, no time trend and no drift term. Dickey, David A., and Wayne A. Fuller. 1979. "Distribution of the Estimators for Autoregressive Time Series with a Unit Root." Journal of the American Statistical Association 74 (366): pp. 427–431. Page 2 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page When there are discrete (at a distinct date) or gradual (over time) changes in the population regression coefficients, the series is nonstationary. These changes are called breaks. There is a variety of reasons why breaks can occur in macroeconomic time series but most often they are related to changes in economic policy or major changes in the structure of the economy. See Chapter 14.7 of the book for some examples. If breaks are not accounted for in the regression model, OLS estimates will reflect the average relationship. Since these estimates might be strongly misleading and result in poor forecast quality, we are interested in testing for breaks. One distinguishes between testing for a break when the date is known and testing for a break with an unknown break date. Let \(\tau\) denote a known break date and let \(D_t(\tau)\) be a binary variable indicating time periods before and after the break. Incorporating the break in an ADL(\(1\),\(1\)) regression model yields \[\begin{align*} Y_t =& \beta_0 + \beta_1 Y_{t-1} + \delta_1 X_{t-1} + \gamma_0 D_t(\tau) + \gamma_1\left[D_t(\tau) \cdot Y_{t-1}\right] \\ &+ \, \gamma_2\left[D_t(\tau) \cdot X_{t-1} \right] + u_t, \end{align*}\] where we allow for discrete changes in \(\beta_0\), \(\beta_1\) and \(\beta_2\) at the break date \(\tau\). The null hypothesis of no break, \[H_0: \gamma_0=\gamma_1=\gamma_2=0,\] can be tested against the alternative that at least one of the \(\gamma\)'s is not zero using an \(F\)-Test. This idea is called a Chow test after Gregory Chow (1960). When the break date is unknown the Quandt likelihood ratio (QLR) test (Quandt 1960) may be used. It is a modified version of the Chow test which uses the largest of all \(F\)-statistics obtained when applying the Chow test for all possible break dates in a predetermined range \(\left[\tau_0,\tau_1\right]\). The QLR test is summarized in Key Concept 14.9. The QLR test can be used to test for a break in the population regression function if the date of the break is unknown. The QLR test statistic is the largest (Chow) \(F(\tau)\) statistic computed over a range of eligible break dates \(\tau_0 \leq \tau \leq \tau_1\): \[\begin{align} QLR = \max\left[F(\tau_0),F(\tau_0 +1),\dots,F(\tau_1)\right]. \tag{14.14} \end{align}\] The most important properties are: The QLR test can be applied to test whether a subset of the coefficients in the population regression function breaks but the test also rejects if there is a slow evolution of the regression function. When there is a single discrete break in the population regression function that lying at a date within the range tested, the \(QLR\) test statistic is \(F(\widehat{\tau})\) and \(\widehat{\tau}/T\) is a consistent estimator of the fraction of the sample at which the break is. The large-sample distribution of \(QLR\) depends on \(q\), the number of restrictions being tested and both ratios of end points to the sample size, \(\tau_0/T, \tau_1/T\). Similar to the ADF test, the large-sample distribution of \(QLR\) is nonstandard. Critical values are presented in Table 14.5 of the book. Using the QLR statistic we may test whether there is a break in the coefficients on the lags of the term spread in (14.5), the ADL(\(2\),\(2\)) regression model of GDP growth. Following Key Concept 14.9 we modify the specification of (14.5) by adding a break dummy \(D(\tau)\) and its interactions with both lags of term spread and choose the range of break points to be tested as 1970:Q1 - 2005:Q2 (these periods are the center 70% of the sample data from 1962:Q2 - 2012:Q4). Thus, the model becomes \[\begin{align*} GDPGR_t =&\, \beta_0 + \beta_1 GDPGR_{t-1} + \beta_2 GDPGR_{t-2} \\ &+\, \beta_3 TSpread_{t-1} + \beta_4 TSpread_{t-2} \\ &+\, \gamma_1 D(\tau) + \gamma_2 (D(\tau) \cdot TSpread_{t-1}) \\ &+\, \gamma_3 (D(\tau) \cdot TSpread_{t-2}) \\ &+\, u_t. \end{align*}\] Next, we estimate the model for each break point and compute the \(F\)-statistic corresponding to the null hypothesis \(H_0: \gamma_1=\gamma_2=\gamma_3=0\). The \(QLR\)-statistic is the largest of the \(F\)-statistics obtained in this manner. # set up a range of possible break dates tau -0.6547438 The hypothesis cannot be rejected at the \(10\%\) significance level. Altogether the analysis suggests that the ADL(\(2\),\(2\)) model coefficients have been stable since the presumed break in the early 1980s. Chow, Gregory C. 1960. "Tests of Equality Between Sets of Coefficients in Two Linear Regressions." Econometrica 28 (3): 591–605. Quandt, Richard E. 1960. "Tests of the Hypothesis That a Linear Regression System Obeys Two Separate Regimes." Journal of the American Statistical Association 55 (290): 324–30. . Page 3 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page The dividend yield (the ratio of current dividends to the stock price) can be considered as an indicator of future excess returns: if a stock has a high current dividend yield, it can be considered undervalued and it can be presumed that the price of the stock goes up in the future, meaning that future excess returns go up. This presumption can be examined using ADL models of excess returns, where lags of the logarithm of the stock's dividend yield serve as additional regressors. Unfortunately, a graphical inspection of the time series of the logarithm of the dividend yield casts doubt on the assumption that the series is stationary which, as has been discussed in Chapter 14.7, is necessary to conduct standard inference in a regression analysis. # plot logarithm of dividend yield series plot(StockReturns[, 2], col = "steelblue", lwd = 2, ylab = "Logarithm", main = "Dividend Yield for CRSP Index") The Dickey-Fuller test statistic for an autoregressive unit root in an AR(\(1\)) model with drift provides further evidence that the series might be nonstationary. # test for unit root in GDP using 'ur.df()' from the package 'urca' summary(ur.df(window(StockReturns[, 2], c(1960,1), c(2002, 12)), type = "drift", lags = 0)) #> #> ### #> # Augmented Dickey-Fuller Test Unit Root Test # #> ### #> #> Test regression drift #> #> #> Call: #> lm(formula = z.diff ~ z.lag.1 + 1) #> #> Residuals: #> Min 1Q Median 3Q Max #> -14.3540 -2.9118 -0.2953 2.6375 25.5169 #> #> Coefficients: #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) -2.740964 2.080038 -1.318 0.188 #> z.lag.1 -0.007652 0.005989 -1.278 0.202 #> #> Residual standard error: 4.45 on 513 degrees of freedom #> Multiple R-squared: 0.003172, Adjusted R-squared: 0.001229 #> F-statistic: 1.633 on 1 and 513 DF, p-value: 0.2019 #> #> #> Value of test-statistic is: -1.2777 0.9339 #> #> Critical values for test statistics: #> 1pct 5pct 10pct #> tau2 -3.43 -2.86 -2.57 #> phi1 6.43 4.59 3.78 We use window() to get observations from January 1960 to December 2002 only. Since the \(t\)-value for the coefficient on the lagged logarithm of the dividend yield is \(-1.27\), the hypothesis that the true coefficient is zero cannot be rejected, even at the \(10\%\) significance level. However, it is possible to examine whether the dividend yield has predictive power for excess returns by using its differences in an ADL(\(1\),\(1\)) and an ADL(\(2\),\(2\)) model (remember that differencing a series with a unit root yields a stationary series), although these model specifications do not correspond to the economic reasoning mentioned above. Thus, we also estimate an ADL(\(1\),\(1\)) regression using the level of the logarithm of the dividend yield. That is we estimate three different specifications: \[\begin{align*} (I) \ excess \, returns_t =& \, \beta_0 + \beta_1 excess \, returns_{t-1} + \beta_3 \Delta \log(dividend yield_{t-1}) + u_t \\ \\ (II) \ excess \, returns_t =& \, \beta_0 + \beta_1 excess \, returns_{t-1} + \beta_2 excess \, returns_{t-2} \\ &+ \, \beta_3 \Delta \log(dividend yield_{t-1}) + \beta_4 \Delta \log(dividend yield_{t-2}) + u_t \\ \\ (III) \ excess \, returns_t =& \, \beta_0 + \beta_1 excess \, returns_{t-1} + \beta_5 \log(dividend yield_{t-1}) + u_t \end{align*}\] # ADL(1,1) (1st difference of log dividend yield) CRSP_ADL_1 FDD 0.471433 0.135195 3.4871 0.0005242 *** #> L(FDD, 1:6)1 0.145021 0.081557 1.7782 0.0758853 . #> L(FDD, 1:6)2 0.058364 0.058911 0.9907 0.3222318 #> L(FDD, 1:6)3 0.074166 0.047143 1.5732 0.1162007 #> L(FDD, 1:6)4 0.036304 0.029335 1.2376 0.2163670 #> L(FDD, 1:6)5 0.048756 0.031370 1.5543 0.1206535 #> L(FDD, 1:6)6 0.050246 0.045129 1.1134 0.2659919 #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 As the result we obtain \[\begin{align} \begin{split} \widehat{\%ChgOJC_t} =& -\underset{(0.21)}{0.69} + \underset{(0.14)}{0.47} FDD_t + \underset{(0.08)}{0.15} FDD_{t-1} + \underset{(0.06)}{0.06} FDD_{t-2} + \underset{(0.05)}{0.07} FDD_{t-3} \\ &+ \underset{(0.03)}{0.04} FDD_{t-4} + \underset{(0.03)}{0.05} FDD_{t-5} + \underset{(0.05)}{0.05} FDD_{t-6}, \end{split} \tag{15.1} \end{align}\] where the coefficient on \(FDD_{t-1}\) estimates the price increase in period \(t\) caused by an additional freezing degree day in the preceding month, the coefficient on \(FDD_{t-2}\) estimates the effect of an additional freezing degree day two month ago and so on. Consequently, the coefficients in (15.1) can be interpreted as price changes in current and future periods due to an unit increase in the current month's freezing degree days. Page 6 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page This section of the book describes the general idea of a dynamic causal effect and how the concept of a randomized controlled experiment can be translated to time series applications, using several examples. In general, for empirical attempts to measure a dynamic causal effect, the assumptions of stationarity (see Key Concept 14.5) and exogeneity must hold. In time series applications up until here we have assumed that the model error term has conditional mean zero given current and past values of the regressors. For estimation of a dynamic causal effect using a distributed lag model, assuming a stronger form termed strict exogeneity may be useful. Strict exogeneity states that the error term has mean zero conditional on past, present and future values of the independent variables. The two concepts of exogeneity and the distributed lag model are summarized in Key Concept 15.1. The general distributed lag model is \[\begin{align} Y_t = \beta_0 + \beta_1 X_t + \beta_2 X_{t-1} + \beta_3 X_{t-2} + \dots + \beta_{r+1} X_{t-r} + u_t, \tag{15.2} \end{align}\] where it is assumed that \(X\) is an exogenous variable, \[E(u_t\vert X_t, X_{t-1}, X_{t-2},\dots) = 0.\] a \(X_t,Y_t\) have a stationary distribution. b \((Y_t,X_t)\) and \((Y_{t-j},X_{t-j})\) become independently distributed as \(j\) gets large. Large outliers are unlikely. In particular, we need that all the variables have more than eight nonzero and finite moments — a stronger assumption than before (four finite nonzero moments) that is required for computation of the HAC covariance matrix estimator. There is no perfect multicollinearity. The distributed lag model may be extended to include contemporaneous and past values of additional regressors. On the assumption of exogeneity There is another form of exogeneity termed strict exogeneity which assumes \[E(u_t\vert \dots, X_{t+2},X_{t+1},X_t,X_{t-1},X_{t-2},\dots)=0,\] that is the error term has mean zero conditional on past, present and future values of \(X\). Strict exogeneity implies exogeneity (as defined in 1. above) but not the other way around. From this point on we will therefore distinguish between exogeneity and strict exogeneity. Exogeneity as in 1. suffices for the OLS estimators of the coefficient in distributed lag models to be consistent. However, if the the assumption of strict exogeneity is valid, more efficient estimators can be applied. Page 7 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page The following terminology regarding the coefficients in the distributed lag model (15.2) is useful for upcoming applications: The dynamic causal effect is also called the dynamic multiplier. \(\beta_{h+1}\) in (15.2) is the \(h\)-period dynamic multiplier. The contemporaneous effect of \(X\) on \(Y\), \(\beta_1\), is termed the impact effect. The \(h\)-period cumulative dynamic multiplier of a unit change in \(X\) and \(Y\) is defined as the cumulative sum of the dynamic multipliers. In particular, \(\beta_1\) is the zero-period cumulative dynamic multiplier, \(\beta_1 + \beta_2\) is the one-period cumulative dynamic multiplier and so forth. The cumulative dynamic multipliers of the distributed lag model (15.2) are the coefficients \(\delta_1,\delta_2,\dots,\delta_r,\delta_{r+1}\) in the modified regression \[\begin{align} Y_t =& \, \delta_0 + \delta_1 \Delta X_t + \delta_2 \Delta X_{t-1} + \dots + \delta_r \Delta X_{t-r+1} + \delta_{r+1} X_{t-r} + u_t, \tag{15.3} \end{align}\] and thus can be directly estimated using OLS which makes it convenient to compute their HAC standard errors. \(\delta_{r+1}\) is called the long-run cumulative dynamic multiplier. It is straightforward to compute the cumulative dynamic multipliers for (15.1), the estimated distributed lag regression of changes in orange juice concentrate prices on freezing degree days, using the corresponding model object orange_DLM and the function cumsum(). # compute cumulative multipliers cum_mult 0.4714329 0.6164542 0.6748177 0.7489835 0.7852874 0.8340436 #> 6-period CDM #> 0.8842895 Translating the distributed lag model with six lags of \(FDD\) to (15.3), we see that the OLS coefficient estimates in this model coincide with the multipliers stored in cum_mult. # estimate cumulative dynamic multipliers using the modified regression cum_mult_reg d(FDD) d(L(FDD, 1:5))1 d(L(FDD, 1:5))2 d(L(FDD, 1:5))3 d(L(FDD, 1:5))4 #> 0.4714329 0.6164542 0.6748177 0.7489835 0.7852874 #> d(L(FDD, 1:5))5 L(FDD, 6) #> 0.8340436 0.8842895 As noted above, using a model specification as in (15.3) allows to easily obtain standard errors for the estimated dynamic cumulative multipliers. # obtain coefficient summary that reports HAC standard errors coeftest(cum_mult_reg, vcov. = vcovHAC) #> #> t test of coefficients: #> #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) -0.69296 0.23668 -2.9278 0.0035431 ** #> d(FDD) 0.47143 0.13583 3.4709 0.0005562 *** #> d(L(FDD, 1:5))1 0.61645 0.13145 4.6896 3.395e-06 *** #> d(L(FDD, 1:5))2 0.67482 0.16009 4.2151 2.882e-05 *** #> d(L(FDD, 1:5))3 0.74898 0.17263 4.3387 1.682e-05 *** #> d(L(FDD, 1:5))4 0.78529 0.17351 4.5260 7.255e-06 *** #> d(L(FDD, 1:5))5 0.83404 0.18236 4.5737 5.827e-06 *** #> L(FDD, 6) 0.88429 0.19303 4.5810 5.634e-06 *** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Page 8 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page The error term \(u_t\) in the distributed lag model (15.2) may be serially correlated due to serially correlated determinants of \(Y_t\) that are not included as regressors. When these factors are not correlated with the regressors included in the model, serially correlated errors do not violate the assumption of exogeneity such that the OLS estimator remains unbiased and consistent. However, autocorrelated standard errors render the usual homoskedasticity-only and heteroskedasticity-robust standard errors invalid and may cause misleading inference. HAC errors are a remedy. Problem: If the error term \(u_t\) in the distributed lag model (15.2) is serially correlated, statistical inference that rests on usual (heteroskedasticity-robust) standard errors can be strongly misleading. Solution: Heteroskedasticity- and autocorrelation-consistent (HAC) estimators of the variance-covariance matrix circumvent this issue. There are R functions like vcovHAC() from the package sandwich which are convenient for computation of such estimators. The package sandwich also contains the function NeweyWest(), an implementation of the HAC variance-covariance estimator proposed by Newey and West (1987). Consider the distributed lag regression model with no lags and a single regressor \(X_t\) \[\begin{align*} Y_t = \beta_0 + \beta_1 X_t + u_t, \end{align*}\] with autocorrelated errors. A brief derivation of \[\begin{align} \overset{\sim}{\sigma}^2_{\widehat{\beta}_1} = \widehat{\sigma}^2_{\widehat{\beta}_1} \widehat{f}_t, \tag{15.4} \end{align}\] the so-called Newey-West variance estimator for the variance of the OLS estimator of \(\beta_1\) is presented in Chapter 15.4 of the book. \(\widehat{\sigma}^2_{\widehat{\beta}_1}\) in (15.4) is the heteroskedasticity-robust variance estimate of \(\widehat{\beta}_1\) and \[\begin{align} \widehat{f}_t = 1 + 2 \sum_{j=1}^{m-1} \left(\frac{m-j}{m}\right) \overset{\sim}{\rho}_j \tag{15.5} \end{align}\] is a correction factor that adjusts for serially correlated errors and involves estimates of \(m-1\) autocorrelation coefficients \(\overset{\sim}{\rho}_j\). As it turns out, using the sample autocorrelation as implemented in acf() to estimate the autocorrelation coefficients renders (15.4) inconsistent, see pp. 650-651 of the book for a detailed argument. Therefore, we use a somewhat different estimator. For a time series \(X\) we have \[\ \overset{\sim}{\rho}_j = \frac{\sum_{t=j+1}^T \hat v_t \hat v_{t-j}}{\sum_{t=1}^T \hat v_t^2}, \ \text{with} \ \hat v= (X_t-\overline{X}) \hat u_t. \] We implement this estimator in the function acf_c() below. \(m\) in (15.5) is a truncation parameter to be chosen. A rule of thumb for choosing \(m\) is \[\begin{align} m = \left \lceil{0.75 \cdot T^{1/3}}\right\rceil. \tag{15.6} \end{align}\] We simulate a time series that, as stated above, follows a distributed lag model with autocorrelated errors and then show how to compute the Newey-West HAC estimate of \(SE(\widehat{\beta}_1)\) using R. This is done via two separate but, as we will see, identical approaches: at first we follow the derivation presented in the book step-by-step and compute the estimate "manually". We then show that the result is exactly the estimate obtained when using the function NeweyWest(). # function that computes rho tilde acf_c Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Newey, Whitney K., and Kenneth D. West. 1987. "A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix." Econometrica 55: 703–8. Page 9 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page In general, the errors in a distributed lag model are correlated which necessitates usage of HAC standard errors for valid inference. If, however, the assumption of exogeneity (the first assumption stated in Key Concept 15.1) is replaced by strict exogeneity, that is, \[E(u_t\vert \dots, X_{t+1}, X_{t}, X_{t-1}, \dots) = 0,\] more efficient approaches than OLS estimation of the coefficients become available. For a general distributed lag model with \(r\) lags and AR(\(p\)) errors, these approaches are summarized in Key Concept 15.4. Consider the general distributed lag model with \(r\) lags and errors following an AR(\(p\)) process, \[\begin{align} Y_t =& \, \beta_0 + \beta_1 X_t + \beta_2 X_{t-1} + \dots + \beta_{r+1} X_{t-r} + u_t \tag{15.7} \\ u_t =& \, \phi_1 u_{t-1} + \phi u_{t-2} + \dots + \phi_p u_{t-p} + \tilde{u}_t. \tag{15.8} \end{align}\] Under strict exogeneity of \(X_t\), one may rewrite the above model in the ADL specification \[\begin{align*} Y_t =& \, \alpha_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} \\ &+ \, \delta_0 X_t + \delta_1 X_{t-1} + \dots + \delta_q X_{t-q} + \tilde{u}_t \end{align*}\] where \(q=r+p\) and compute estimates of the dynamic multipliers \(\beta_1, \beta_2, \dots, \beta_{r+1}\) using OLS estimates of \(\phi_1, \phi_2, \dots, \phi_p, \delta_0, \delta_1, \dots, \delta_q\). An alternative is to estimate the dynamic multipliers using feasible GLS, that is to apply the OLS estimator to a quasi-difference specification of (15.7). Under strict exogeneity, the feasible GLS approach is the BLUE estimator for the dynamic multipliers in large samples. On the one hand, as demonstrated in Chapter 15.5 of the book, OLS estimation of the ADL representation can be beneficial for estimation of the dynamic multipliers in large distributed lag models because it allows for a more parsimonious model that may be a good approximation to the large model. On the other hand, the GLS approach is more efficient than the ADL estimator if the sample size is large. We shortly review how the different representations of a small distributed lag model can be obtained and show how this specification can be estimated by OLS and GLS using R. The model is \[\begin{align} Y_t = \beta_0 + \beta_1 X_t + \beta_2 X_{t-1} + u_t, \tag{15.9} \end{align}\] so a change in \(X\) is modeled to effect \(Y\) contemporaneously (\(\beta_1\)) and in the next period (\(\beta_2\)). The error term \(u_t\) is assumed to follow an AR(\(1\)) process,\[u_t = \phi_1 u_{t-1} + \overset{\sim}{u_t},\] where \(\overset{\sim}{u_t}\) is serially uncorrelated. One can show that the ADL representation of this model is \[\begin{align} Y_t = \alpha_0 + \phi_1 Y_{t-1} + \delta_0 X_t + \delta_1 X_{t-1} + \delta_2 X_{t-2} + \tilde{u}_t, \tag{15.10} \end{align}\] with the restrictions \[\begin{align*} \beta_1 =& \, \delta_0, \\ \beta_2 =& \, \delta_1 + \phi_1 \delta_0, \end{align*}\] see p. 657 of the book. Another way of writing the ADL(\(1\),\(2\)) representation (15.10) is the quasi-difference model \[\begin{align} \tilde{Y}_t = \alpha_0 + \beta_1 \tilde{X}_t + \beta_2 \tilde{X}_{t-1} + \tilde{u}_t, \tag{15.11} \end{align}\] where \(\tilde{Y}_t = Y_t - \phi_1 Y_{t-1}\) and \(\tilde{X}_t = X_t - \phi_1 X_{t-1}\). Notice that the error term \(\tilde{u}_t\) is uncorrelated in both models and, as shown in Chapter 15.5 of the book, \[E(u_t\vert X_{t+1}, X_t, X_{t-1}, \dots) = 0,\] which is implied by the assumption of strict exogeneity. We continue by simulating a time series of \(500\) observations using the model (15.9) with \(\beta_1 = 0.1\), \(\beta_2 = 0.25\), \(\phi = 0.5\) and \(\tilde{u}_t \sim \mathcal{N}(0,1)\) and estimate the different representations, starting with the distributed lag model (15.9). # set seed for reproducibility set.seed(1) # simulate a time series with serially correlated errors obs (Intercept) 0.038340 0.073411 0.5223 0.601717 #> X 0.123661 0.046710 2.6474 0.008368 ** #> L(X) 0.247406 0.046377 5.3347 1.458e-07 *** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Next, we estimate the ADL(\(1\),\(2\)) model (15.10) using OLS. The errors are uncorrelated in this representation of the model. This statement is supported by a plot of the sample autocorrelation function of the residual series. # estimate the ADL(2,1) representation of the distributed lag model adl21_dynamic 0.1176425 0.2478484 Strict exogeneity allows for OLS estimation of the quasi-difference model (15.11). The idea of applying the OLS estimator to a model where the variables are linearly transformed, such that the model errors are uncorrelated and homoskedastic, is called generalized least squares (GLS). The OLS estimator in (15.11) is called the infeasible GLS estimator because \(\tilde{Y}\) and \(\tilde{X}\) cannot be computed without knowing \(\phi_1\), the autoregressive coefficient in the error AR(\(1\)) model, which is generally unknown in practice. Assume we knew that \(\phi = 0.5\). We then may obtain the infeasible GLS estimates of the dynamic multipliers in (15.9) by applying OLS to the transformed data. # GLS: estimate quasi-differenced specification by OLS iGLS_dynamic #> Time series regression with "ts" data: #> Start = 3, End = 500 #> #> Call: #> dynlm(formula = I(Y - 0.5 * L(Y)) ~ I(X - 0.5 * L(X)) + I(L(X) - #> 0.5 * L(X, 2))) #> #> Residuals: #> Min 1Q Median 3Q Max #> -3.0325 -0.6375 -0.0499 0.6658 3.7724 #> #> Coefficients: #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) 0.01620 0.04564 0.355 0.72273 #> I(X - 0.5 * L(X)) 0.12000 0.04237 2.832 0.00481 ** #> I(L(X) - 0.5 * L(X, 2)) 0.25266 0.04237 5.963 4.72e-09 *** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 #> #> Residual standard error: 1.017 on 495 degrees of freedom #> Multiple R-squared: 0.07035, Adjusted R-squared: 0.0666 #> F-statistic: 18.73 on 2 and 495 DF, p-value: 1.442e-08 The feasible GLS estimator uses preliminary estimation of the coefficients in the presumed error term model, computes the quasi-differenced data and then estimates the model using OLS. This idea was introduced by Cochrane and Orcutt (1949) and can be extended by continuing this process iteratively. Such a procedure is implemented in the function cochrane.orcutt() from the package orcutt. X_t #> Estimate Std. Error t value Pr(>|t|) #> (Intercept) 0.032885 0.085163 0.386 0.69956 #> X_t 0.120128 0.042534 2.824 0.00493 ** #> X_l1 0.252406 0.042538 5.934 5.572e-09 *** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 #> #> Residual standard error: 1.0165 on 495 degrees of freedom #> Multiple R-squared: 0.0704 , Adjusted R-squared: 0.0666 #> F-statistic: 18.7 on 2 and 495 DF, p-value: < 1.429e-08 #> #> Durbin-Watson statistic #> (original): 1.06907 , p-value: 1.05e-25 #> (transformed): 1.98192 , p-value: 4.246e-01 Some more sophisticated methods for GLS estimation are provided with the package nlme. The function gls() can be used to fit linear models by maximum likelihood estimation algorithms and allows to specify a correlation structure for the error term. # feasible GLS maximum likelihood estimation procedure summary(gls(Y_t ~ X_t + X_l1, correlation = corAR1())) #> Generalized least squares fit by REML #> Model: Y_t ~ X_t + X_l1 #> Data: NULL #> AIC BIC logLik #> 1451.847 1472.88 -720.9235 #> #> Correlation Structure: AR(1) #> Formula: ~1 #> Parameter estimate(s): #> Phi #> 0.4668343 #> #> Coefficients: #> Value Std.Error t-value p-value #> (Intercept) 0.03929124 0.08530544 0.460595 0.6453 #> X_t 0.11986994 0.04252270 2.818963 0.0050 #> X_l1 0.25287471 0.04252497 5.946500 0.0000 #> #> Correlation: #> (Intr) X_t #> X_t 0.039 #> X_l1 0.037 0.230 #> #> Standardized residuals: #> Min Q1 Med Q3 Max #> -3.00075518 -0.64255522 -0.05400347 0.69101814 3.28555793 #> #> Residual standard error: 1.14952 #> Degrees of freedom: 499 total; 496 residual Notice that in this example, the coefficient estimates produced by GLS are somewhat closer to their true values and that the standard errors are the smallest for the GLS estimator. Cochrane, D., and G. H Orcutt. 1949. "Application of Least Squares Regression to Relationships Containing Auto-Correlated Error Terms." Journal of the American Statistical Association 44 (245): 32–61. . Page 10 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page This section investigates the following two questions using the time series regression methods discussed here: How persistent is the effect of a single freeze on orange juice concentrate prices? Has the effect been stable over the whole time span? We start by estimating dynamic causal effects with a distributed lag model where \(\%ChgOJC_t\) is regressed on \(FDD_t\) and 18 lags. A second model specification considers a transformation of the the distributed lag model which allows to estimate the 19 cumulative dynamic multipliers using OLS. The third model, adds 11 binary variables (one for each of the months from February to December) to adjust for a possible omitted variable bias arising from correlation of \(FDD_t\) and seasons by adding season(FDD) to the right hand side of the formula of the second model. # estimate distributed lag models of frozen orange juice price changes FOJC_mod_DM Model 1: FOJC_pctc ~ L(d(FDD), 0:17) + L(FDD, 18) + season(FDD) #> Model 2: FOJC_pctc ~ L(d(FDD), 0:17) + L(FDD, 18) #> Res.Df Df F Pr(>F) #> 1 563 #> 2 574 -11 0.9683 0.4743 The \(p\)-value is \(0.47\) so we cannot reject the hypothesis that the coefficients on the monthly dummies are zero, even at the \(10\%\) level. We conclude that the seasonal fluctuations in demand for orange juice do not pose a serious threat to internal validity of the model. It is convenient to use plots of dynamic multipliers and cumulative dynamic multipliers. The following two code chunks reproduce Figures 15.2 (a) and 15.2 (b) of the book which display point estimates of dynamic and cumulative multipliers along with upper and lower bounds of their \(95\%\) confidence intervals computed using the above HAC standard errors. # 95% CI bounds point_estimates --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 We end up with the following results: \[\begin{align*} GDPGR_t =& \, \underset{(0.46)}{0.52} + \underset{(0.11)}{0.29} GDPGR_{t-1} + \underset{(0.09)}{0.22} GDPGR_{t-2} -\underset{(0.36)}{0.90} TSpread_{t-1} + \underset{(0.39)}{1.33} TSpread_{t-2}, \\ TSpread_t =& \, \underset{(0.12)}{0.46} + \underset{(0.02)}{0.01} GDPGR_{t-1} -\underset{(0.03)}{0.06} GDPGR_{t-2} + \underset{(0.10)}{1.06} TSpread_{t-1} -\underset{(0.11)}{0.22} TSpread_{t-2}. \end{align*}\] The function VAR() can be used to obtain the same coefficient estimates as presented above since it applies OLS per equation, too. # set up data for estimation using `VAR()` VAR_data VAR Estimation Results: #> ========================= #> #> Estimated coefficients for equation GDPGrowth: #> == #> Call: #> GDPGrowth = GDPGrowth.l1 + TSpread.l1 + GDPGrowth.l2 + TSpread.l2 + const #> #> GDPGrowth.l1 TSpread.l1 GDPGrowth.l2 TSpread.l2 const #> 0.2895533 -0.9025493 0.2163919 1.3298305 0.5163440 #> #> #> Estimated coefficients for equation TSpread: #> == #> Call: #> TSpread = GDPGrowth.l1 + TSpread.l1 + GDPGrowth.l2 + TSpread.l2 + const #> #> GDPGrowth.l1 TSpread.l1 GDPGrowth.l2 TSpread.l2 const #> 0.009978489 1.058227945 -0.057245123 -0.219190243 0.455773969 VAR() returns a list of lm objects which can be passed to the usual functions, for example summary() and so it is straightforward to obtain model statistics for the individual equations. # obtain the adj. R^2 from the output of 'VAR()' summary(VAR_est$varresult$GDPGrowth)$adj.r.squared #> [1] 0.2887223 summary(VAR_est$varresult$TSpread)$adj.r.squared #> [1] 0.8254311 We may use the individual model objects to conduct Granger causality tests. # Granger causality tests: # test if term spread has no power in explaining GDP growth linearHypothesis(VAR_EQ1, hypothesis.matrix = c("TSpread_t-1", "TSpread_t-2"), vcov. = sandwich) #> Linear hypothesis test #> #> Hypothesis: #> TSpread_t-1 #> TSpread_t-2 #> #> Model 1: restricted model #> Model 2: GDPGrowth ~ L(GDPGrowth, 1:2) + L(TSpread, 1:2) #> #> Note: Coefficient covariance matrix supplied. #> #> Res.Df Df F Pr(>F) #> 1 125 #> 2 123 2 5.9094 0.003544 ** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 # test if GDP growth has no power in explaining term spread linearHypothesis(VAR_EQ2, hypothesis.matrix = c("Growth_t-1", "Growth_t-2"), vcov. = sandwich) #> Linear hypothesis test #> #> Hypothesis: #> Growth_t-1 #> Growth_t-2 #> #> Model 1: restricted model #> Model 2: TSpread ~ L(GDPGrowth, 1:2) + L(TSpread, 1:2) #> #> Note: Coefficient covariance matrix supplied. #> #> Res.Df Df F Pr(>F) #> 1 125 #> 2 123 2 3.4777 0.03395 * #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Both Granger causality tests reject at the level of \(5\%\). This is evidence in favor of the conjecture that the term spread has power in explaining GDP growth and vice versa. The idea of an iterated forecast for period \(T+2\) based on observations up to period \(T\) is to use the one-period-ahead forecast as an intermediate step. That is, the forecast for period \(T+1\) is used as an observation when predicting the level of a series for period \(T+2\). This can be generalized to a \(h\)-period-ahead forecast where all intervening periods between \(T\) and \(T+h\) must be forecasted as they are used as observations in the process (see Chapter 16.2 of the book for a more detailed argument on this concept). Iterated multiperiod forecasts are summarized in Key Concept 16.2. The steps for an iterated multiperiod AR forecast are: Estimate the AR(\(p\)) model using OLS and compute the one-period-ahead forecast. Use the one-period-ahead forecast to obtain the two-period-ahead forecast. Continue by iterating to obtain forecasts farther into the future. An iterated multiperiod VAR forecast is done as follows: Estimate the VAR(\(p\)) model using OLS per equation and compute the one-period-ahead forecast for all variables in the VAR. Use the one-period-ahead forecasts to obtain the two-period-ahead forecasts. Continue by iterating to obtain forecasts of all variables in the VAR farther into the future. Since a VAR models all variables using lags of the respective other variables, we need to compute forecasts for all variables. It can be cumbersome to do so when the VAR is large but fortunately there are R functions that facilitate this. For example, the function predict() can be used to obtain iterated multivariate forecasts for VAR models estimated by the function VAR(). The following code chunk shows how to compute iterated forecasts for GDP growth and the term spread up to period 2015:Q1, that is \(h=10\), using the model object VAR_est. # compute iterated forecasts for GDP growth and term spread for the next 10 quarters forecasts $GDPGrowth #> fcst lower upper CI #> [1,] 1.738653 -3.006124 6.483430 4.744777 #> [2,] 1.692193 -3.312731 6.697118 5.004925 #> [3,] 1.911852 -3.282880 7.106583 5.194731 #> [4,] 2.137070 -3.164247 7.438386 5.301317 #> [5,] 2.329667 -3.041435 7.700769 5.371102 #> [6,] 2.496815 -2.931819 7.925449 5.428634 #> [7,] 2.631849 -2.846390 8.110088 5.478239 #> [8,] 2.734819 -2.785426 8.255064 5.520245 #> [9,] 2.808291 -2.745597 8.362180 5.553889 #> [10,] 2.856169 -2.722905 8.435243 5.579074 #> #> $TSpread #> fcst lower upper CI #> [1,] 1.676746 0.708471226 2.645021 0.9682751 #> [2,] 1.884098 0.471880228 3.296316 1.4122179 #> [3,] 1.999409 0.336348101 3.662470 1.6630609 #> [4,] 2.080836 0.242407507 3.919265 1.8384285 #> [5,] 2.131402 0.175797245 4.087008 1.9556052 #> [6,] 2.156094 0.125220562 4.186968 2.0308738 #> [7,] 2.161783 0.085037834 4.238528 2.0767452 #> [8,] 2.154170 0.051061544 4.257278 2.1031082 #> [9,] 2.138164 0.020749780 4.255578 2.1174139 #> [10,] 2.117733 -0.007139213 4.242605 2.1248722 This reveals that the two-quarter-ahead forecast of GDP growth in 2013:Q2 using data through 2012:Q4 is \(1.69\). For the same period, the iterated VAR forecast for the term spread is \(1.88\). The matrices returned by predict(VAR_est) also include \(95\%\) prediction intervals (however, the function does not adjust for autocorrelation or heteroskedasticity of the errors!). We may also plot the iterated forecasts for both variables by calling plot() on the output of predict(VAR_est). # visualize the iterated forecasts plot(forecasts) A direct multiperiod forecast uses a model where the predictors are lagged appropriately such that the available observations can be used directly to do the forecast. The idea of direct multiperiod forecasting is summarized in Key Concept 16.3. A direct multiperiod forecast that forecasts \(h\) periods into the future using a model of \(Y_t\) and an additional predictor \(X_t\) with \(p\) lags is done by first estimating \[\begin{align*} Y_t =& \, \delta_0 + \delta_1 Y_{t-h} + \dots + \delta_{p} Y_{t-p-h+1} + \delta_{p+1} X_{t-h} \\ +& \dots + \delta_{2p} Y_{t-p-h+1} + u_t, \end{align*}\] which is then used to compute the forecast of \(Y_{T+h}\) based on observations through period \(T\). For example, to obtain two-quarter-ahead forecasts of GDP growth and the term spread we first estimate the equations \[\begin{align*} GDPGR_t =& \, \beta_{10} + \beta_{11} GDPGR_{t-2} + \beta_{12} GDPGR_{t-3} + \gamma_{11} TSpread_{t-2} + \gamma_{12} TSpread_{t-3} + u_{1t}, \\ TSpread_t =& \, \beta_{20} + \beta_{21} GDPGR_{t-2} + \beta_{22} GDPGR_{t-3} + \gamma_{21} TSpread_{t-2} + \gamma_{22} TSpread_{t-3} + u_{2t} \end{align*}\] and then substitute the values of \(GDPGR_{2012:Q4}\), \(GDPGR_{2012:Q3}\), \(TSpread_{2012:Q4}\) and \(TSpread_{2012:Q3}\) into both equations. This is easily done manually. # estimate models for direct two-quarter-ahead forecasts VAR_EQ1_direct [1,] 2.439497 coef(VAR_EQ2_direct) %*% c(1, # intercept window(GDPGrowth, start = c(2012, 3), end = c(2012, 4)), window(TSpread, start = c(2012, 3), end = c(2012, 4))) #> [,1] #> [1,] 1.66578 Applied economists often use the iterated method since this forecasts are more reliable in terms of \(MSFE\), provided that the one-period-ahead model is correctly specified. If this is not the case, for example because one equation in a VAR is believed to be misspecified, it can be beneficial to use direct forecasts since the iterated method will then be biased and thus have a higher \(MSFE\) than the direct method. See Chapter 16.2 for a more detailed discussion on advantages and disadvantages of both methods. Stock, J. H., and M. W. Watson. 2015. Introduction to Econometrics, Third Update, Global Edition. Pearson Education Limited. Page 13 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page Some economic time series have smoother trends than variables that can be described by random walk models. A way to model these time series is \[\Delta Y_t = \beta_0 + \Delta Y_{t-1} + u_t,\] where \(u_t\) is a serially uncorrelated error term. This model states that the first difference of a series is a random walk. Consequently, the series of second differences of \(Y_t\) is stationary. Key Concept 16.4 summarizes the notation. When a time series \(Y_t\) has a unit autoregressive root, \(Y_t\) is integrated of order one. This is often denoted by \(Y_t \sim I(1)\). We simply say that \(Y_t\) is \(I(1)\). If \(Y_t\) is \(I(1)\), its first difference \(\Delta Y_t\) is stationary. \(Y_t\) is \(I(2)\) when \(Y_t\) needs to be differenced twice in order to obtain a stationary series. Using the notation introduced here, if \(Y_t\) is \(I(2)\), its first difference \(\Delta Y_t\) is \(I(1)\) and its second difference \(\Delta^2 Y_t\) is stationary. \(Y_t\) is \(I(d)\) when \(Y_t\) must be differenced \(d\) times to obtain a stationary series. When \(Y_t\) is stationary, it is integrated of order \(0\) so \(Y_t\) is \(I(0)\). It is fairly easy to obtain differences of time series in R. For example, the function diff() returns suitably lagged and iterated differences of numeric vectors, matrices and time series objects of the class ts. Following the book, we take the price level of the U.S. measured by the Personal Consumption Expenditures Price Index as an example. # define ts object of the U.S. PCE Price Index PCECTPI #> ### #> # Elliot, Rothenberg and Stock Unit Root Test # #> ### #> #> Test of type DF-GLS #> detrending of series with intercept and trend #> #> #> Call: #> lm(formula = dfgls.form, data = data.dfgls) #> #> Residuals: #> Min 1Q Median 3Q Max #> -0.025739 -0.004054 0.000017 0.004619 0.033620 #> #> Coefficients: #> Estimate Std. Error t value Pr(>|t|) #> yd.lag -0.01213 0.01012 -1.199 0.23207 #> yd.diff.lag1 0.28583 0.07002 4.082 6.47e-05 *** #> yd.diff.lag2 0.19320 0.07058 2.737 0.00676 ** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 #> #> Residual standard error: 0.007807 on 198 degrees of freedom #> Multiple R-squared: 0.1504, Adjusted R-squared: 0.1376 #> F-statistic: 11.69 on 3 and 198 DF, p-value: 4.392e-07 #> #> #> Value of test-statistic is: -1.1987 #> #> Critical values of DF-GLS are: #> 1pct 5pct 10pct #> critical values -3.48 -2.89 -2.57 The summary of the test shows that the test statistic is about \(-1.2\). The \(10\%\) critical value for the DF-GLS test is \(-2.57\). This is, however, not the appropriate critical value for the ADF test when an intercept and a time trend are included in the Dickey-Fuller regression: the asymptotic distributions of both test statistics differ and so do their critical values! The test is left-sided so we cannot reject the null hypothesis that U.S. inflation is nonstationary, using the DF-GLS test. Elliott, Graham, Thomas J Rothenberg, and James H Stock. 1996. "Efficient Tests for an Autoregressive Unit Root." Econometrica 64 (4): 813–36. Page 14 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page When \(X_t\) and \(Y_t\) are \(I(1)\) and if there is a \(\theta\) such that \(Y_t - \theta X_t\) is \(I(0)\), \(X_t\) and \(Y_t\) are cointegrated. Put differently, cointegration of \(X_t\) and \(Y_t\) means that \(X_t\) and \(Y_t\) have the same or a common stochastic trend and that this trend can be eliminated by taking a specific difference of the series such that the resulting series is stationary. R functions for cointegration analysis are implemented in the package urca. As an example, reconsider the relation between short-term and long-term interest rates in the example of U.S. 3-month treasury bills, U.S. 10-years treasury bonds and the spread in their interest rates which have been introduced in Chapter 14.4. The next code chunk shows how to reproduce Figure 16.2 of the book. # reproduce Figure 16.2 of the book # plot both interest series plot(merge(as.zoo(TB3MS), as.zoo(TB10YS)), plot.type = "single", lty = c(2, 1), lwd = 2, xlab = "Date", ylab = "Percent per annum", ylim = c(-5, 17), main = "Interest Rates") # add the term spread series lines(as.zoo(TSpread), col = "steelblue", lwd = 2, xlab = "Date", ylab = "Percent per annum", main = "Term Spread") # shade the term spread polygon(c(time(TB3MS), rev(time(TB3MS))), c(TB10YS, rev(TB3MS)), col = alpha("steelblue", alpha = 0.3), border = NA) # add horizontal line at 0 abline(0, 0) # add a legend legend("topright", legend = c("TB3MS", "TB10YS", "Term Spread"), col = c("black", "black", "steelblue"), lwd = c(2, 2, 2), lty = c(2, 1, 1)) The plot suggests that long-term and short-term interest rates are cointegrated: both interest series seem to have the same long-run behavior. They share a common stochastic trend. The term spread, which is obtained by taking the difference between long-term and short-term interest rates, seems to be stationary. In fact, the expectations theory of the term structure suggests the cointegrating coefficient \(\theta\) to be 1. This is consistent with the visual result. Following Key Concept 16.5, it seems natural to construct a test for cointegration of two series in the following manner: if two series \(X_t\) and \(Y_t\) are cointegrated, the series obtained by taking the difference \(Y_t - \theta X_t\) must be stationary. If the series are not cointegrated, \(Y_t - \theta X_t\) is nonstationary. This is an assumption that can be tested using a unit root test. We have to distinguish between two cases: \(\theta\) is known. Knowledge of \(\theta\) enables us to compute differences \(z_t = Y_t - \theta X_t\) so that Dickey-Fuller and DF-GLS unit root tests can be applied to \(z_t\). For these tests, the critical values are the critical values of the ADF or DF-GLS test. \(\theta\) is unknown. If \(\theta\) is unknown, it must be estimated before the unit root test can be applied. This is done by estimating the regression \[Y_t = \alpha + \theta X_t + z_t\] using OLS (this is refered to as the first-stage regression). Then, a Dickey-Fuller test is used for testing the hypothesis that \(z_t\) is a nonstationary series. This is known as the Engle-Granger Augmented Dickey-Fuller test for cointegration (or EG-ADF test) after Engle and Granger (1987). The critical values for this test are special as the associated null distribution is nonnormal and depends on the number of \(I(1)\) variables used as regressors in the first stage regression. You may look them up in Table 16.2 of the book. When there are only two presumably cointegrated variables (and thus a single \(I(1)\) variable is used in the first stage OLS regression) the critical values for the levels \(10\%\), \(5\%\) and \(1\%\) are \(-3.12\), \(-3.41\) and \(-3.96\). As has been mentioned above, the theory of the term structure suggests that long-term and short-term interest rates are cointegrated with a cointegration coefficient of \(\theta = 1\). In the previous section we have seen that there is visual evidence for this conjecture since the spread of 10-year and 3-month interest rates looks stationary. We continue by using formal tests (the ADF and the DF-GLS test) to see whether the individual interest rate series are integrated and if their difference is stationary (for now, we assume that \(\theta = 1\) is known). Both is conveniently done by using the functions ur.df() for computation of the ADF test and ur.ers for conducting the DF-GLS test. Following the book we use data from 1962:Q1 to 2012:Q4 and employ models that include a drift term. We set the maximum lag order to \(6\) and use the \(AIC\) for selection of the optimal lag length. # test for nonstationarity of 3-month treasury bills using ADF test ur.df(window(TB3MS, c(1962, 1), c(2012, 4)), lags = 6, selectlags = "AIC", type = "drift") #> #> ### #> # Augmented Dickey-Fuller Test Unit Root / Cointegration Test # #> ### #> #> The value of the test statistic is: -2.1004 2.2385 # test for nonstationarity of 10-years treasury bonds using ADF test ur.df(window(TB10YS, c(1962, 1), c(2012, 4)), lags = 6, selectlags = "AIC", type = "drift") #> #> ### #> # Augmented Dickey-Fuller Test Unit Root / Cointegration Test # #> ### #> #> The value of the test statistic is: -1.0079 0.5501 # test for nonstationarity of 3-month treasury bills using DF-GLS test ur.ers(window(TB3MS, c(1962, 1), c(2012, 4)), model = "constant", lag.max = 6) #> #>

#> # Elliot, Rothenberg and Stock Unit Root / Cointegration Test # #> ### #> #> The value of the test statistic is: -1.8042 # test for nonstationarity of 10-years treasury bonds using DF-GLS test ur.ers(window(TB10YS, c(1962, 1), c(2012, 4)), model = "constant", lag.max = 6) #> #> ### #> # Elliot, Rothenberg and Stock Unit Root / Cointegration Test # #> ### #> #> The value of the test statistic is: -0.942 The corresponding \(10\%\) critical value for both tests is \(-2.57\) so we cannot reject the null hypotheses of nonstationary for either series, even at the \(10\%\) level of significance.12 We conclude that it is plausible to model both interest rate series as \(I(1)\). Next, we apply the ADF and the DF-GLS test to test for nonstationarity of the term spread series, which means we test for non-cointegration of long- and short-term interest rates. # test if term spread is stationary (cointegration of interest rates) using ADF ur.df(window(TB10YS, c(1962, 1), c(2012, 4)) - window(TB3MS, c(1962, 1), c(2012 ,4)), lags = 6, selectlags = "AIC", type = "drift") #> #> ### #> # Augmented Dickey-Fuller Test Unit Root / Cointegration Test # #> ### #> #> The value of the test statistic is: -3.9308 7.7362 # test if term spread is stationary (cointegration of interest rates) using DF-GLS ur.ers(window(TB10YS, c(1962, 1), c(2012, 4)) - window(TB3MS, c(1962, 1),c(2012, 4)), model = "constant", lag.max = 6) #> #> ### #> # Elliot, Rothenberg and Stock Unit Root / Cointegration Test # #> ### #> #> The value of the test statistic is: -3.8576 Table 16.1 summarizes the results. Table 16.1: ADF and DF-GLS Test Statistics for Interest Rate Series TB3MS \(-2.10\) \(-1.80\) TB10YS \(-1.01\) \(-0.94\) TB10YS - TB3MS \(-3.93\) \(-3.86\) Both tests reject the hypothesis of nonstationarity of the term spread series at the \(1\%\) level of significance, which is strong evidence in favor of the hypothesis that the term spread is stationary, implying cointegration of long- and short-term interest rates. Since theory suggests that \(\theta=1\), there is no need to estimate \(\theta\) so it is not necessary to use the EG-ADF test which allows \(\theta\) to be unknown. However, since it is instructive to do so, we follow the book and compute this test statistic. The first-stage OLS regression is \[TB10YS_t = \beta_0 + \beta_1 TB3MS_t + z_t.\] # estimate first-stage regression of EG-ADF test FS_EGADF #> Time series regression with "ts" data: #> Start = 1962(1), End = 2012(4) #> #> Call: #> dynlm(formula = window(TB10YS, c(1962, 1), c(2012, 4)) ~ window(TB3MS, #> c(1962, 1), c(2012, 4))) #> #> Coefficients: #> (Intercept) window(TB3MS, c(1962, 1), c(2012, 4)) #> 2.4642 0.8147 Thus we have \[\begin{align*} \widehat{TB10YS}_t = 2.46 + 0.81 \cdot TB3MS_t, \end{align*}\] where \(\widehat{\theta} = 0.81\). Next, we take the residual series \(\{\widehat{z_t}\}\) and compute the ADF test statistic. # compute the residuals z_hat #> ### #> # Augmented Dickey-Fuller Test Unit Root / Cointegration Test # #> ### #> #> The value of the test statistic is: -3.1935 The test statistic is \(-3.19\) which is smaller than the \(10\%\) critical value but larger than the \(5\%\) critical value (see Table 16.2 of the book). Thus, the null hypothesis of no cointegration can be rejected at the \(10\%\) level but not at the \(5\%\) level. This indicates lower power of the EG-ADF test due to the estimation of \(\theta\): when \(\theta=1\) is the correct value, we expect the power of the ADF test for a unit root in the residuals series \(\widehat{z} = TB10YS - TB3MS\) to be higher than when some estimate \(\widehat{\theta}\) is used. If two \(I(1)\) time series \(X_t\) and \(Y_t\) are cointegrated, their differences are stationary and can be modeled in a VAR which is augmented by the regressor \(Y_{t-1} - \theta X_{t-1}\). This is called a vector error correction model (VECM) and \(Y_{t} - \theta X_{t}\) is called the error correction term. Lagged values of the error correction term are useful for predicting \(\Delta X_t\) and/or \(\Delta Y_t\). A VECM can be used to model the two interest rates considered in the previous sections. Following the book we specify the VECM to include two lags of both series as regressors and choose \(\theta = 1\), as theory suggests (see above). TB10YS D_TB3MS_l1 -0.0016601 0.0727060 -0.0228 0.981807 #> D_TB3MS_l2 -0.0680845 0.0435059 -1.5649 0.119216 #> D_TB10YS_l1 0.2264878 0.0957071 2.3665 0.018939 * #> D_TB10YS_l2 -0.0734486 0.0703476 -1.0441 0.297740 #> ect_l1 -0.0878871 0.0285644 -3.0768 0.002393 ** #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 coeftest(VECM_EQ2, vcov. = NeweyWest(VECM_EQ2, prewhite = F, adjust = T)) #> #> t test of coefficients: #> #> Estimate Std. Error t value Pr(>|t|) #> Intercept -0.060746 0.107937 -0.5628 0.57422 #> D_TB3MS_l1 0.240003 0.111611 2.1504 0.03276 * #> D_TB3MS_l2 -0.155883 0.153845 -1.0132 0.31220 #> D_TB10YS_l1 0.113740 0.125571 0.9058 0.36617 #> D_TB10YS_l2 -0.147519 0.112630 -1.3098 0.19182 #> ect_l1 0.031506 0.050519 0.6236 0.53359 #> --- #> Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 Thus the two estimated equations of the VECM are \[\begin{align*} \widehat{\Delta TB3MS}_t =& \, -\underset{(0.11)}{0.06} + \underset{(0.11)}{0.24} \Delta TB3MS_{t-1} -\underset{(0.15)}{0.16} \Delta TB3MS_{t-2} \\ &+ \underset{(0.13)}{0.11} \Delta TB10YS_{t-1} -\underset{(0.11)}{0.15} \Delta TB10YS_{t-2} + \underset{(0.05)}{0.03} ECT_{t-1}, and, \\ \widehat{\Delta TB10YS}_t =& \, \underset{(0.06)}{0.12} -\underset{(0.07)}{0.00} \Delta TB3MS_{t-1} -\underset{(0.04)}{0.07} \Delta TB3MS_{t-2} \\ &+ \underset{(0.10)}{0.23} \Delta TB10YS_{t-1} -\underset{(0.07)}{0.07} \Delta TB10YS_{t-2} -\underset{(0.03)}{0.09} ECT_{t-1}. \end{align*}\] The output produced by coeftest() shows that there is little evidence that lagged values of the differenced interest series are useful for prediction. This finding is more pronounced for the equation of the differenced series of the 3-month treasury bill rate, where the error correction term (the lagged term spread) is not significantly different from zero at any common level of significance. However, for the differenced 10-years treasury bonds rate the error correction term is statistically significant at \(1\%\) with an estimate of \(-0.09\). This can be interpreted as follows: although both interest rates are nonstationary, their conintegrating relationship allows to predict the change in the 10-years treasury bonds rate using the VECM. In particular, the negative estimate of the coefficient on the error correction term indicates that there will be a negative change in the next period's 10-years treasury bonds rate when the 10-years treasury bonds rate is unusually high relative to the 3-month treasury bill rate in the current period. Engle, Robert, and Clive Granger. 1987. "Co-integration and Error Correction: Representation, Estimation and Testing." Econometrica 55 (2): 251–76. Page 15 This book is in Open Review. We want your feedback to make the book better for you and other students. You may annotate some text by selecting it with the cursor and then click "Annotate" in the pop-up menu. You can also see the annotations of others: click the arrow in the upper right hand corner of the page Financial time series often exhibit a behavior that is known as volatility clustering: the volatility changes over time and its degree shows a tendency to persist, i.e., there are periods of low volatility and periods where volatility is high. Econometricians call this autoregressive conditional heteroskedasticity. Conditional heteroskedasticity is an interesting property because it can be exploited for forecasting the variance of future periods. As an example, we consider daily changes in the Whilshire 5000 stock index. The data is available for download at the Federal Reserve Economic Data Base. For consistency with the book we download data from 1989-29-12 to 2013-12-31 (choosing this somewhat larger time span is necessary since later on we will be working with daily changes of the series). The following code chunk shows how to format the data and how to reproduce Figure 16.3 of the book. # import data on the Wilshire 5000 index W5000