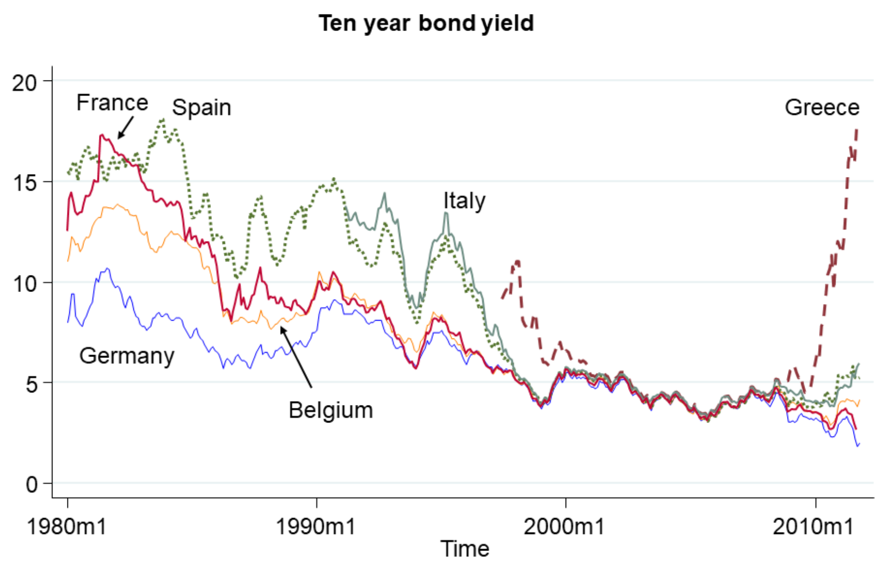
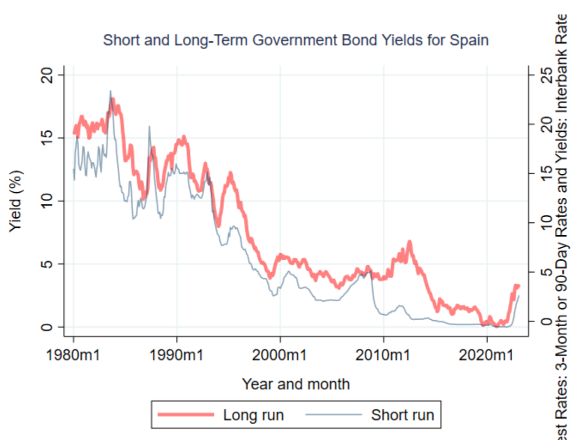
Section 11: Time Series II

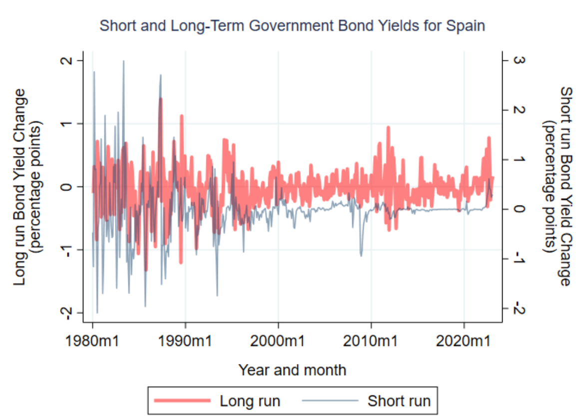
In this section, we will try to find a good model for movements in the long-term interest rate for Spanish bonds.[1] We will focus on Spain for no particular reason. Recall that all the section worksheets are revised versions of section notes that I wrote with classmates when I was in graduate school. At the time, this was one of the most popular graphs in the financial world:



The updated data through March 2023 are below. The interest rate series for Spain seems to very persistent. If we look at the change in the interest rate from one month to the next instead, there is not such a pronounced trend. Visual inspection thus suggests to use percentage point changes in the yield rather than levels in our forecasting exercise.



(a) Levels



(b) Changes (percentage points)

Figure 1: Plots of long-run and short-run yields for Spanish bonds

[1] Data were downloaded from FRED using the *freduse* command in Stata. Type *ssc install freduse* to install the *freduse* package and *freduse IRLTLT01ESM156N IR3TIB01ESM156N, clear* to download this data.

1 Model selection

Last week we talked about how to select the proper lag length of an $AR(p)$ and $ADL(p, q)$ model using the BIC. Consider the $ADL(p, q_1, q_2)$ model

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \ldots + \beta_p Y_{t-p} + \beta_{p+1} X_{t-1} + \ldots + \beta_{p+q_1} X_{t-q_1} + \beta_{p+q_1+1} W_{t-1} + \ldots + \beta_{p+q_1+q_2} W_{t-q_2} + u_t, \quad (1)$$

where the model contains p lags of the dependent variable Y and q lags of another variable X.

Q1a: How do we choose p, q_1, and q_2? Why?

```
* Basics of choosing lag length for ADL(p,q)
local bicmin= 1e+9
forvalues p = 1/6 {
forvalues q = 1/6 {

reg dspain_il L(1/`p').dspain_il L(1/`q').dspain_is if tin(1981m2,2023m3), r

* BIC and adjusted Rsquared
scalar bic = ln(e(rss)/e(N)) + e(df_m)*ln(e(N))/e(N)
scalar ar2 = e(r2_a)
dis "AR(`p') BIC =" bic
dis "AR(`p') Adjusted Rsquared= " e(r2_a)
dis "AR(`p') Rsquared= " e(r2)

if bic < `bicmin' {
local bicmin = bic
local phat = `p'
local qhat = `q'
}
}

}

. dis "BIC= " `bicmin'
BIC= -2.3972994

.
. dis "ADL BIC lengths: p = " `phat' " q = " `qhat'
ADL BIC lengths: p = 1 q = 3

.
.
.
.
.
. reg dspain_il L.dspain_il L(1/3).dspain_is if tin(1981m2,2023m3), r


Linear regression                               Number of obs     =        505
                                                F(4, 500)         =       7.85
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1284
                                                Root MSE          =     .29718

------------------------------------------------------------------------------
             |               Robust
   dspain_il | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   dspain_il |
         L1. |   .2051405   .0689265     2.98   0.003     .0697193    .3405616
             |
   dspain_is |
         L1. |   .1082347   .0544117     1.99   0.047      .001331    .2151384
         L2. |  -.0389224   .0407971    -0.95   0.341    -.1190774    .0412325
         L3. |   .1245847   .0352302     3.54   0.000     .0553672    .1938022
             |
       _cons |  -.0141025    .013519    -1.04   0.297    -.0406635    .0124585
------------------------------------------------------------------------------
```

Q1b: Should we use Newey and West (1987) standard errors or heteroskedasticity robust standard errors? Why?

1.1 Testing for structural breaks

A break is when the population regression function changes over the course of the sample, due either to a discrete change in the population regression coefficient at a distinct date or a gradual evolution of the coefficients over a longer period of time.

In this case, OLS will produce an average estimate over the entire sample even if the behavior of the series is very different over different time periods.

Q2: Why should we care about breaks? How could it bias our predictions?

Q3: Is a time series with a break stationary? Why or why not?

1.1.1 Known break date

If the break date is known, we can test whether the coefficients before and after the break date are different. Note the similarity to testing for different coefficients for different groups (e.g. men and women).

We define a dummy variable that is 0 before the break date and 1 after the break date,

$$D_t(\tau) = \begin{cases} 0, & t < \tau \\ 1, & t \geq \tau \end{cases} \quad . \tag{2}$$

Then we interact the dummy with the regressors such that our model becomes:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \gamma_0 D_t(\tau) + \gamma_1 D_t(\tau) Y_{t-1} + u_t. \tag{3}$$

Q4: What test do we want to run to test for the break in the AR(1) model?

This is an F-test, but in this particular application, it is often called a *Chow-test*.

Spain joined the European Union in January 1986. It is possible that this event changed the coefficients on the interest rate process. Let's check it out.

```
. gen d86 = 0
. replace d86 = 1 if tin(1986m1,2023m3)

. gen d86_Ldspain_il = d86*L.dspain_il
. gen d86_Ldspain_is = d86*L.dspain_is
. gen d86_L2dspain_is = d86*L2.dspain_is
. gen d86_L3dspain_is = d86*L3.dspain_is


 reg dspain_il L.dspain_il L.dspain_is d86 d86_Ldspain_il  d86_Ldspain_is d86_L2dspai
> n_is d86_L3dspain_is if tin(1981m2,2023m2), r

Linear regression                               Number of obs     =        505
                                                F(7, 497)         =       4.40
                                                Prob > F          =     0.0001
                                                R-squared         =     0.1204
                                                Root MSE          =     .29945

------------------------------------------------------------------------------
             |               Robust
   dspain_il | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   dspain_il |
         L1. |   .2892637   .1050806     2.75   0.006     .0828068    .4957206
             |
   dspain_is |
         L1. |   .0241726   .0679156     0.36   0.722    -.1092645    .1576098
             |
         d86 |   .0275983   .0566452     0.49   0.626    -.0836952    .1388918
d86_Ldspain_il |  -.1105346   .1348321    -0.82   0.413    -.3754459    .1543766
d86_Ldspain_is |   .1445446    .109595     1.32   0.188     -.070782    .3598712
d86_L2dspain_is |  -.0545094   .0680623    -0.80   0.424    -.1882347    .0792159
d86_L3dspain_is |   .1088279   .0473038     2.30   0.022     .0158879    .2017679
       _cons |  -.0396543   .0549771    -0.72   0.471    -.1476706    .0683619
------------------------------------------------------------------------------
```

```
. testparm d86 d86_Ldspain_il d86_Ldspain_is d86_L2dspain_is d86_L3dspain_is

 ( 1)  d86 = 0
 ( 2)  d86_Ldspain_il = 0
 ( 3)  d86_Ldspain_is = 0
 ( 4)  d86_L2dspain_is = 0
 ( 5)  d86_L3dspain_is = 0

       F(  5,   497) =    1.71
            Prob > F =    0.1314
```

Q5: What do we conclude from the Chow test?

1.1.2 Unknown break date

If the break date is unknown (which is often more realistic even if you think you know the break date), you can test for a structural break at different points by repeating the Chow test above. You can collect all the F-statistics from these tests (denote the F-statistic for a break at date τ by $F(\tau)$), and just look at the highest one (the intuition being that if there is a break that is where it should be). The highest F-statistic is called the *Quandt likelihood ratio (QLR) statistic*, that is, if we compute the F-statistic for all potential break dates between τ_0 and τ_1,

$$QLR = \max_{\tau \in \{\tau_0,\ldots,\tau_1\}} F(\tau). \tag{4}$$

The largest of the F statistics does not have the same distribution as the usual F statistic. Instead, it has a special distribution that depends on the number of restrictions in the null and the endpoints as a fraction of the total sample size (τ_0/T,τ_1/T). We will generally use 15% trimming where $\tau_1 = 0.85T$ and $\tau_0 = 0.15T$ (rounded to the nearest integer).[2] This means we will estimate the F statistic for every period in the middle 70% of our sample's time range. A couple of notes:

1. If there is a discrete break in $[\tau_0, \tau_1]$ the QLR will reject the null that there is no break with a high probability and will provide an estimate $\widehat{\tau}$ as the date with the largest F statistic (i.e., $F(\widehat{\tau}) = QLR$).
2. The QLR also detects other instabilities in the coefficients. $\Rightarrow$ If the QLR rejects it could mean there is (1) a single discrete break; (2) multiple discrete breaks; or (3) slow evolution of the regression function.
3. In practice, you rarely know the break date exactly. For instance, the introduction of the Euro was prepared and anticipated for quite a while. It is not clear why the break date should be the actual introduction rather than some prior date. We therefore *recommend to always use the QLR test statistic.*

The computation of the QLR test statistic follows the code in the lecture notes. *You first need to download the file* qlr.ado *from the course website and put it in the same folder as your data.* Here is how you would proceed for the Spanish interest rate:

```
. qlr dspain_il L.dspain_il L(1/3).dspain_is if tin(1981m2,2023m2), graph type(png) mlabpos(10)

     +-----------------------------------------+
     |    date          F   restrict   trimming |
     |-----------------------------------------|
  1. |  2014m9    3.15696          5        .15 |
  2. | 2014m10   3.031764          5        .15 |
  3. | 2014m11   2.932097          5        .15 |
  4. | 2014m12   2.777852          5        .15 |
  5. |  2015m7   2.763096          5        .15 |
     +-----------------------------------------+

Computed with Newey-West standard errors with 6 lags. Results saved as qlr.dta and qlr.png

. display "break date = " r(maxdate)
break date =  2014m9

. display "QLR statistic = " r(qlr)
QLR statistic = 3.1569597

. display "Number of restrictions = " r(restrict)
```

[2] See table 14.6 on page 559 of the book for critical values with 15% trimming. The critical values are larger than the $F_{q,\infty}$ distribution's critical values.

```
Number of restrictions = 5
```

We produce a graph of the F-statistic for the different breakpoint values.

Q6: Why are there five restrictions? How many would we have if we were testing an AR(3) model?

Q7: What is the critcial value for the test? What do we conclude?

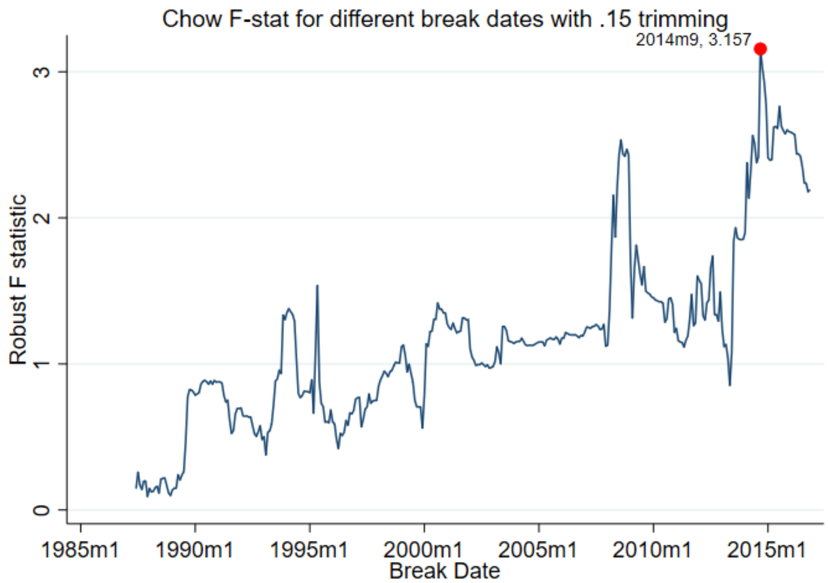


Figure 2: QLR test

2 Forecast error and forecast intervals

The *forecast error* is

$$FE_{t+1} = Y_{t+1} - \hat{Y}_{t+1|t}, \tag{5}$$

which is just the difference between the actual and the predicted value. This error can be due to *two sources.* The deviation of our estimated coefficients from the actual values (as before when we were not making predictions), and idiosyncratic errors from the future disturbance u_{t+1}. For instance, for the AR(1) model, we have

$$FE_{t+1} = u_{t+1} - \left[(\hat{\beta}_0 - \beta_0) + (\hat{\beta}_1 - \beta_1)Y_t\right]. \tag{6}$$

The *mean squared forecast error (MSFE)* is defined as

$$MSFE = E\left[(Y_{t+1} - \hat{Y}_{t+1|t})^2\right], \tag{7}$$

and its square root is the *root mean squared forecast error (RMFSE).* Under the assumption that u_{t+1} is normally distributed, we want a 68% confidence interval to be

$$\hat{Y}_{t+1} \pm 1 * RMSFE. \tag{8}$$

In period t, we do not know the actual value of Y_{t+1}, so it is hard to estimate the RMFSE. However, we can estimate the RMFSE by doing Pseudo-Out-Of-Sample prediction.

We'll start by estimating our regression using data up to one year from the end of our sample. Then, using the *predict* command in Stata we can compute the fitted values from this regression for the last year of our data. Once we have these fitted values, we can compute the RMSE within the last year of data, which should approximate the RMSFE. Here is how you would compute this for the ADL(1,3) in our example:

```
. reg dspain_il L.dspain_il L(1/3).dspain_is if tin(1981m2,2022m2), r


Linear regression                               Number of obs     =        505
                                                F(4, 500)         =       7.85
                                                Prob > F          =     0.0000
                                                R-squared         =     0.1284
                                                Root MSE          =     .29718

------------------------------------------------------------------------------
             |               Robust
   dspain_il | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
   dspain_il |
         L1. |   .2051405   .0689265     2.98   0.003     .0697193    .3405616
             |
   dspain_is |
         L1. |   .1082347   .0544117     1.99   0.047      .001331    .2151384
         L2. |  -.0389224   .0407971    -0.95   0.341    -.1190774    .0412325
         L3. |   .1245847   .0352302     3.54   0.000     .0553672    .1938022
             |
       _cons |  -.0141025    .013519    -1.04   0.297    -.0406635    .0124585
------------------------------------------------------------------------------



. tsappend, add(1)

. predict yhat_dspain_il

. *Convert prediction in changes back to levels
. gen yhat_yield = L.IRLTLT01ESM156N + yhat_dspain_il

. *Squared Error
. gen error2 = (IRLTLT01ESM156N - yhat_yield)^2
(39 missing values generated)

. *Mean error over POOS period

. sum error2 if tin(2022m3,2023m2)

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
      error2 |         12    .1266221    .1797011   .0013043   .6087307


. *Save square root of mean error
. scalar rmspe = sqrt(r(mean))

. disp rmspe
.35583998


. list time IRLTLT01ESM156N dspain_il yhat_yield yhat_dspain_il if tin(2023m1, 2023m3), noobs


  +----------------------------------------------------+
  |   time   IRLT~156N   dspain~l   yhat_y~d   yhat_d~l |
  |----------------------------------------------------|
  | 2023m1   3.2226364   .1116364   3.156785   .0457849 |
  | 2023m2      3.3891   .1664636   3.297774   .0751376 |
  | 2023m3           .          .   3.456464   .0673635 |
  +----------------------------------------------------+
```

The 68% forecast interval for March 2023 is $3.456\% \pm 1 \times 0.3558\%$.

A second method is GARCH(1,1):

```
. arch  dspain_il L.dspain_il L(1/3).dspain_is if tin(1981m2,2023m2), arch(1) garch(1)


(setting optimization to BHHH)
Iteration 0:   log likelihood =  -69.53453
Iteration 1:   log likelihood = -40.895507
Iteration 2:   log likelihood = -33.916765
```

```
Iteration 3:   log likelihood = -30.417101
Iteration 4:   log likelihood =  -28.36287
(switching optimization to BFGS)
Iteration 5:   log likelihood = -27.557575
Iteration 6:   log likelihood = -27.051332
Iteration 7:   log likelihood = -26.944057
Iteration 8:   log likelihood = -26.920412
Iteration 9:   log likelihood = -26.917149
Iteration 10:  log likelihood =  -26.91678
Iteration 11:  log likelihood = -26.916722
Iteration 12:  log likelihood = -26.916722

ARCH family regression

Sample: 1981m2 thru 2023m2                      Number of obs     =        505
                                                Wald chi2(4)      =      64.33
Log likelihood = -26.91672                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |                 OPG
   dspain_il | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
dspain_il    |
   dspain_il |
         L1. |    .270894   .0474756     5.71   0.000     .1778436    .3639444
             |
   dspain_is |
         L1. |    .064869   .0362206     1.79   0.073     -.006122    .1358601
         L2. |  -.0501699   .0375973    -1.33   0.182    -.1238591    .0235194
         L3. |   .1134379    .038616     2.94   0.003     .0377519    .1891239
             |
       _cons |  -.0127963   .0098833    -1.29   0.195    -.0321671    .0065746
-------------+----------------------------------------------------------------
ARCH         |
        arch |
         L1. |     .12026   .0224165     5.36   0.000     .0763245    .1641955
             |
       garch |
         L1. |   .8777025   .0171697    51.12   0.000     .8440505    .9113546
             |
       _cons |   .0006972   .0004468     1.56   0.119    -.0001785    .0015729
------------------------------------------------------------------------------

. predict cv, variance

. gen garch_rmsfe = sqrt(cv)

.
. *List GARCH RMSFE
. list time garch_rmsfe if tin(2023m1, 2023m3), noobs

  +--------------------+
  |   time   garch_~e |
  |--------------------|
  | 2023m1   .3471511 |
  | 2023m2   .3274736 |
  | 2023m3   .3097982 |
  +--------------------+
```

What is the 68% forecast interval for March 2023 using the GARCH(1,1)?

What is a third way to estimate the RMSFE? Which method or methods are preferred and why?