1 Forecasting and time series models

Until now, we have focused on estimating the causal effect of X on Y. In this part of the class, we will now focus on estimating the best possible prediction of our outcome variable, $\hat{Y}$, in datasets where Y is unknown. To do this, we calculate $\hat{\beta}$ using data where we know both X and Y, and apply it to a new sample where we know X but do not know Y:

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

One application of prediction methods are in time series data. Time series data are repeated measures of some variable over time. For example, annual GDP per capita of the United States in the past 50 years: $GDP_{1971}, GDP_{1970}, ..., GDP_{2023}$. Our goal will be to use previous values of a variable to make predictions about that variable in the future. We call this **forecasting**. One simple but extremely powerful way to make the prediction is to regress Y_t on Y_{t-1}. Our prediction at time period $t+1$ would be:

$$\hat{Y}_{t+1} = \hat{\beta}_0 + \hat{\beta}_1 Y_t$$

For forecasting purposes, we usually care more about the predictive power of the model than the coefficient estimates.

2 Stationarity

To make predictions about the future using data that we have now, the past must be like the present. The technical assumption is called **stationarity**. The stationary assumption means the joint distribution of the time series $\{Y_t\}$ does not change over time. One way to say this is that the series is stationary if the joint distribution of $(Y_{j+1}, \ldots, Y_{T+j})$ does not depend on j.

Q1: When might stationarity be violated?

Example: S&P 500 stock price index

We can download the S&P 500 daily price from FRED, a database of economic data put together by the St. Louis Fed. The S&P 500 data are available on the web by going to https://research.stlouisfed.org/fred2/ and searching for "S&P 500", but we can download them directly into Stata using the *freduse* command. To start, make sure you have the *freduse* command installed by typing *ssc install freduse*.

```
. freduse SP500, clear
(2,609 observations read)

. tsset daten

Time variable: daten, 08apr2013 to 06apr2023, but with gaps
        Delta: 1 day
```

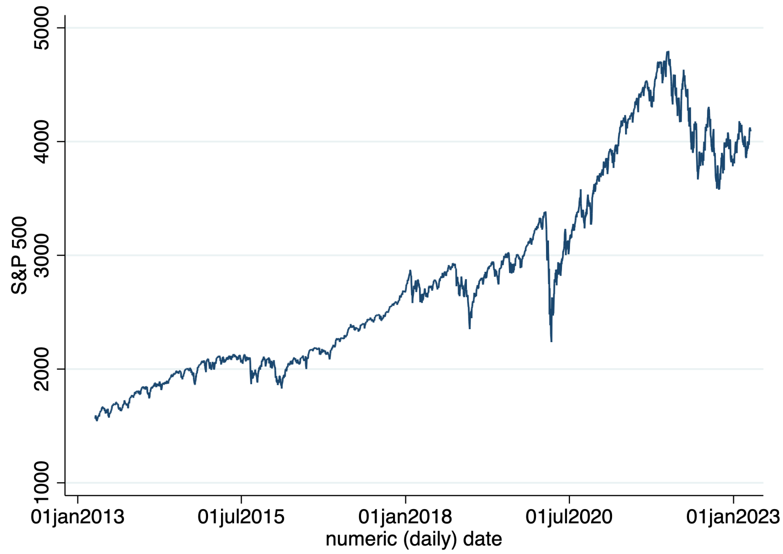

Figure 1: Level over time

The output says but with gaps. Why are there gaps?

```
. drop if SP500 == .
(91 observations deleted)
. gen tradingdate = _n
. tsset tradingdate
        time variable:  tradingdate, 1 to 2517
                delta:  1 unit
```

We can also look at the day-to-day percent change in the price:

```
. * 1-day return
. gen lnsp500 = ln(SP500)

. gen dprice = 100*(lnsp500-L.lnsp500)
(1 missing value generated)

. tsline dprice
```

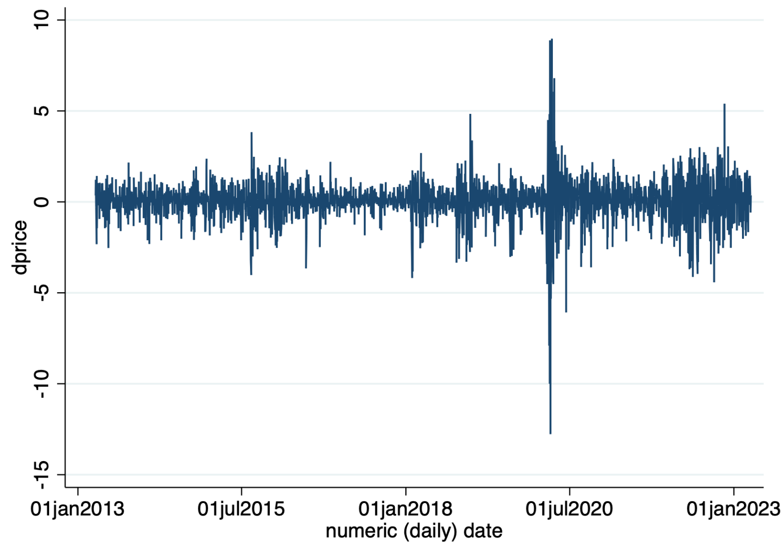

Figure 2: Difference over time

Q2: Is the stock price stationary?

Q3: Is the change in the stock price stationary?

Since we will need to work with stationary data, you may want to try a few different transformations. Differences, $Y_t - Y_{t-1}$, and differences of the logs, $\ln Y_t - \ln Y_{t-1}$, are the most common.

Q4: How do we interpret our units if we transform our data using a difference in logs?

3 Autocorrelation

For forecasting, it also must be the case that our data exhibits **autocorrelation**. That is, the value of Y in each period must be correlated with its value in previous periods.

Definition: The j^{th} autocorrelation coefficient ρ_j is the correlation between Y_t and Y_{t-j}.

$$\rho_j = \frac{Cov(Y_t, Y_{t-j})}{\sqrt{Var(Y_t) \cdot Var(Y_{t-j})}},$$

where $-1 \leq \rho_j \leq 1$. We can use Stata to find the autocorrelation of the day-to-day change in price:

```
. corrgram dprice, lags(4) noplot

 LAG       AC       PAC      Q     Prob>Q
-----------------------------------------
1       -0.1419  -0.1419   50.788  0.0000
2        0.0844   0.0655   68.733  0.0000
3       -0.0130   0.0077   69.156  0.0000
4       -0.0628  -0.0708   79.098  0.0000
```

Q5: Are these autocorrelations large or small? If we try to forecast using lagged values, will we be able to make good forecasts?

4 Autoregressive Models

One of the easiest ways to make a forecast is by regressing Y_t on its previous values $Y_{t-1}, Y_{t-2}, ..., Y_{t-j}$. However, we will need to choose the number of lags to include.

A **first order autoregressive model** AR(1) is one in which we regress Y_t on Y_{t-1}. That is, an AR(1) uses only one lag:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + u_t$$

A p**th order autoregressive model** AR(p) is one in which we regress Y_t on $Y_{t-1}, ..., Y_{t-p}$. That is, an AR(p) uses p lags:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + ... + \beta_p Y_{t-p} + u_t,$$

We might be able to improve our predictions by including other variables in the models. When we regress Y_t on previous values of itself and on values of X_t at different time periods, we call this an **autoregressive distributed lag** model. An ADL(p,q) model is given by:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + ... + \beta_p Y_{t-p} + \delta_1 X_{t-1} + ... + \delta_q X_{t-q} + u_t$$

Q6: How many "lags" would be in an AR(2) model?

Q7: Suppose we have an ADL(2,2) model - write out that regresson form.

Q8: Should we include X_t in an ADL(2,2)?

Now lets run an AR(1) model of change in stock price:

```
. reg dprice L.dprice if inrange(tradingdate, 10, 2506), r

Linear regression                               Number of obs     =      2,497
                                                F(1, 2495)        =       6.50
                                                Prob > F          =     0.0109
                                                R-squared         =     0.0202
                                                Root MSE          =     1.1095

------------------------------------------------------------------------------
             |               Robust
      dprice | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      dprice |
         L1. |  -.1420803   .0557364    -2.55   0.011    -.2513747   -.0327859
             |
       _cons |   .0429638   .0225019     1.91   0.056    -.0011604    .0870881
------------------------------------------------------------------------------
```

Alternatively, we can also run an AR(4) regression.

```
. reg dprice L(1/4).dprice if inrange(tradingdate, 10, 2506), r

Linear regression                               Number of obs     =      2,497
                                                F(4, 2492)        =       1.90
                                                Prob > F          =     0.1069
                                                R-squared         =     0.0295
                                                Root MSE          =     1.1049

------------------------------------------------------------------------------
             |               Robust
      dprice | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      dprice |
         L1. |  -.1325738   .0521997    -2.54   0.011     -.234933   -.0302145
         L2. |   .0722308   .0555435     1.30   0.194    -.0366853     .181147
         L3. |  -.0031842   .0492408    -0.06   0.948    -.0997412    .0933729
         L4. |  -.0707955   .0474745    -1.49   0.136     -.163889     .022298
             |
       _cons |   .0426205   .0239032     1.78   0.075    -.0042516    .0894926
------------------------------------------------------------------------------

. predict ar4
```

Suppose we have data up to 4/11/2022 and we would like to forecast the change in price on 4/12/2021.

Q9: How many previous observations will we need to forecast using an AR(1) model? What about an AR(4) model?

The last four observations of dprice are:

```
. list tradingdate daten dprice SP500 if tradingdate>2513, noobs

  +-------------------------------------------+
  | tradin~e       daten      dprice    SP500 |
  |-------------------------------------------|
  |     2514   30mar2023    .5699158  4050.83 |
  |     2515   31mar2023    1.433372  4109.31 |
  |     2516   03apr2023    .3691673  4124.51 |
  |     2517   04apr2023   -.5813599   4100.6 |
  |     2518   05apr2023   -.2495766  4090.38 |
  |-------------------------------------------|
  |     2519   06apr2023    .3573418  4105.02 |
  +-------------------------------------------+
```

Q10: What is our forecast of the change in price on 4/12/2022 from the AR(1) model we estimated?

Q11: What is our forecast of the change in price on 4/12/2022 in the AR(4) model?

Q12: How would we estimate the price level on 4/12/2022 using the result from the AR(4) model?

Now let's bring in a couple more predictors: the Effective Federal Funds Rate and the Economic Policy Uncertainty Index for United States. Now it is helpful to start using the predict command in Stata.

```
. reg dprice L(1/4).dprice L(2/4).USEPUINDXD L(2/4).EFFR if inrange(tradingdate, 10, 2506), r

Linear regression                               Number of obs     =      2,440
                                                F(10, 2429)       =       1.76
                                                Prob > F          =     0.0637
                                                R-squared         =     0.0419
                                                Root MSE          =     1.1023

------------------------------------------------------------------------------
             |               Robust
      dprice | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+----------------------------------------------------------------
      dprice |
         L1. |  -.1383559   .0528366    -2.62   0.009    -.2419653   -.0347465
         L2. |   .0678646   .0560157     1.21   0.226    -.0419789    .1777081
         L3. |  -.0059389   .0505018    -0.12   0.906      -.10497    .0930922
         L4. |  -.0746264   .0480658    -1.55   0.121    -.1688806    .0196279
             |
  USEPUINDXD |
         L2. |   .0011974   .0006162     1.94   0.052    -.0000109    .0024058
         L3. |  -.0004334   .0005379    -0.81   0.420    -.0014881    .0006213
         L4. |   .0003433   .0005493     0.62   0.532    -.0007338    .0014204
             |
        EFFR |
         L2. |   1.027125   .8746586     1.17   0.240    -.6880291    2.742279
         L3. |   .0691541   1.161339     0.06   0.953    -2.208164    2.346472
         L4. |  -1.109539    .859951    -1.29   0.197    -2.795852    .5767745
             |
       _cons |  -.0791377   .0528937    -1.50   0.135    -.1828591    .0245836
------------------------------------------------------------------------------

. dis "BIC = " ln(e(rss)/e(N)) + e(df_m)*ln(e(N))/e(N)
BIC = .22223365

. predict adl444
```

```
(option xb assumed; fitted values)
(62 missing values generated)

.
. list daten ar4 adl444 daten dprice SP500  if tradingdate>2513, noobs

  +-------------------------------------------------------------------+
  |     daten         ar4      adl444       daten      dprice     SP500 |
  |-------------------------------------------------------------------|
  | 30mar2023   -.1965007   -.2519794   30mar2023    .5699158   4050.83 |
  | 31mar2023    .0580273    .0114266   31mar2023    1.433372   4109.31 |
  | 03apr2023    -.099596   -.2169351   03apr2023    .3691673   4124.51 |
  | 04apr2023   -.0046811   -.0202198   04apr2023   -.5813599    4100.6 |
  | 05apr2023    .1014473    .0388367   05apr2023   -.2495766   4090.38 |
  |-------------------------------------------------------------------|
```

The same F-tests that we know and love have new names in time series forecasting, mostly for historical reasons. In lecture, we saw the Granger causality test. Does the Effective Federal Funds Rate “Granger cause” the SP500 return? This is all about prediction and not causation in the way we used it for the previous weeks in the class.

```
. testparm L(1/4).dprice

 ( 1)  L.dprice = 0
 ( 2)  L2.dprice = 0
 ( 3)  L3.dprice = 0
 ( 4)  L4.dprice = 0

       F(  4,  2419) =    2.10
            Prob > F =    0.0786

. testparm L(1/4).USEPUINDXD

 ( 1)  L.USEPUINDXD = 0
 ( 2)  L2.USEPUINDXD = 0
 ( 3)  L3.USEPUINDXD = 0
 ( 4)  L4.USEPUINDXD = 0

       F(  4,  2419) =    2.70
            Prob > F =    0.0291

. testparm L(1/4).EFFR

 ( 1)  L.EFFR = 0
 ( 2)  L2.EFFR = 0
 ( 3)  L3.EFFR = 0
 ( 4)  L4.EFFR = 0

       F(  4,  2419) =    1.85
            Prob > F =    0.1157
```

In lecture on Thursday, we discussed model selection – that is, how to choose the number of lags of each variable in a time series forecasting model. We are not going to use these F-statistics to do model selection. Instead we will use information criterion, which will be reminiscent of the penalized optimization problems for lasso and ridge regression. The difference is that we will include a penalty for the number of lags that we include, rather than the size of the coefficients themselves like in lasso or ridge.

```
. quietly reg dprice L.dprice if inrange(tradingdate, 10, 2506), r
. dis "BIC = " ln(e(rss)/e(N)) + e(df_m)*ln(e(N))/e(N)
BIC = .21020277
. dis "AIC = " ln(e(rss)/e(N)) + e(df_m)*2/e(N)
AIC = .20787083
. dis "Adjusted Squared = " e(r2_a)
Adjusted Rsquared = .01979269


. quietly reg dprice L(1/2).dprice if inrange(tradingdate, 10, 2506), r
. dis "BIC = " ln(e(rss)/e(N)) + e(df_m)*ln(e(N))/e(N)
BIC = .20888576
. dis "AIC = " ln(e(rss)/e(N)) + e(df_m)*2/e(N)
AIC = .20422189
. dis "Adjusted Rsquared = " e(r2_a)
Adjusted Rsquared = .02375354


. quietly reg dprice L(1/3).dprice if inrange(tradingdate, 10, 2506), r
. dis "BIC = " ln(e(rss)/e(N)) + e(df_m)*ln(e(N))/e(N)
BIC = .2119797
. dis "AIC = " ln(e(rss)/e(N)) + e(df_m)*2/e(N)
AIC = .20498389
. dis "Adjusted Rsquared = " e(r2_a)
Adjusted Rsquared = .02339999


. quietly reg dprice L(1/4).dprice if inrange(tradingdate, 10, 2506), r
. dis "BIC = " ln(e(rss)/e(N)) + e(df_m)*ln(e(N))/e(N)
BIC = .21008039
. dis "AIC = " ln(e(rss)/e(N)) + e(df_m)*2/e(N)
AIC = .20075264
. dis "Adjusted Rsquared = " e(r2_a)
Adjusted Rsquared = .02791218


. quietly reg dprice L(1/4).dprice L(2/4).USEPUINDXD L(2/4).EFFR if inrange(tradingdate, 10, 2506), r
. dis "BIC = " ln(e(rss)/e(N)) + e(df_m)*ln(e(N))/e(N)
BIC = .22223365
. dis "AIC = " ln(e(rss)/e(N)) + e(df_m)*2/e(N)
AIC = .19846417
. dis "Adjusted Rsquared = " e(r2_a)
Adjusted Rsquared = .0379732
```