Introduction to time-series analysis using Stata

Gustavo Sánchez

StataCorp LLC

August 23, 2022

Outline

- Stata tools
- Inputing and managing time series in Stata
  - Date formats in Stata
  - Importing data `import fred`
  - Specifying the time structure and business calendar
  - Time series operators
- Univariate time-series analysis
  - ARIMA models
  - ARCH models
- Multivariate time-series analysis
  - Vector autoregressive (VAR) model
- Forecasting systems of equations

From OLS to Time Series

- Many of the results in OLS and cross-sectional models assume that $\{y_i, x_i\}_{i=1}^n$ are independent draws from the same distribution, like rolling dice n times, i.e, they are independent and identically distributed, i.i.d.
- In time series we are working with draws that depend on each other, e.g. the exchange rate today is not independent from the exchange rate yesterday, for instance.

Stata Tools

- **Data management**
- Univariate time series
  - **ARIMA**, ARFIMA, **ARCH**
  - Prais, Newey, unobserved component model (UCM)
  - Markov-Switching and threshold regressions
  - Time-series smoothers and filters
  - Diagnostic tools
- Multivariate time series.
  - Dynamic factor (dfactor)
  - Multivariate GARCH
  - State space
  - **VAR**, SVAR, VEC models
  - Forecasting, inference, and interpretation
  - Diagnostic tools, graphs and more.
- **Forecasting models**

Managing the time series structure in Stata

Some date formats in Stata

Format	Description	Coding	
%td	daily (same as %d)	0 = 01jan1960,	1 = 02jan1960
%tw	weekly	0 = 1960w1,	1 = 1960w2
%tm	monthly	0 = 1960m1,	1 = 1960m2
%tq	quarterly	0 = 1960q1,	1 = 1960q2
%th	halfyearly	0 = 1960h1,	1 = 1960h2
%ty	yearly	1960 = 1960,	1961 = 1961

Note: times before 1960 are allowed. For instance, -1 means 31dec1959 in %td format and 1959q4 in %tq format.

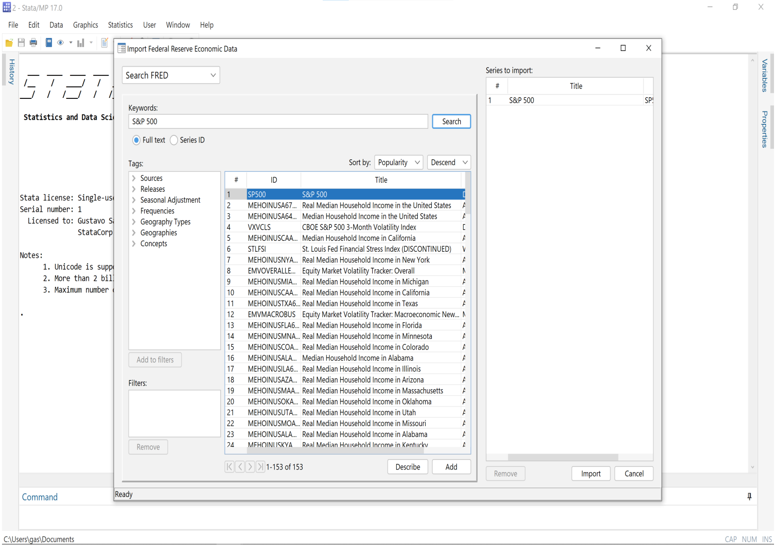
- use **generic** for unformatted time variable
- use **datetime business calendars** for business-day data or other irregular time formats

Example 1: Importing data from FRED

- We can use `import fred` to get data from the Federal Reserve Economic Data (FRED).

```
 . import fred SP500 CBBTCUSD VIXCLS, daterange(2017-01-03 2022-08-31)
. rename SP500     sp500
. rename VIXCLS    vix
. rename CBBTCUSD  bitcoin

 . describe
Contains data
 Observations:          2,057
    Variables:              5
-----------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-----------------------------------------------------------------------------
datestr         str10   %-10s                 observation date
daten           int     %td                   numeric (daily) date
sp500           float   %9.0g                 S&P 500
bitcoin         float   %9.0g                 Coinbase Bitcoin
vix             float   %9.0g                 CBOE Volatility Index: VIX
-----------------------------------------------------------------------------
Sorted by: datestr
     Note: Dataset has changed since last saved.
```

`import fred`: Dialog box



Calendars and (non)missing data

- Specify the time series structure

```
. tsset daten,daily
Time variable: daten, 03jan2017 to 21aug2022
        Delta: 1 day
```

- Some processes imply that there is no data for certain periods. For example, Weekends and holidays in daily financial data.
- For some days there is no data, but there are no missing observations on the process

```
. list daten sp500 bitcoin vix  in 1/8,sep(4)
```

	daten	sp500	bitcoin	vix
1.	03jan2017	2257.83	1020.67	12.85
2.	04jan2017	2270.75	1130.3	11.85
3.	05jan2017	2269	1007	11.67
4.	06jan2017	2276.98	895.71	11.32
5.	07jan2017	.	909	.
6.	08jan2017	.	923.33	.
7.	09jan2017	2268.9	902.66	11.56
8.	10jan2017	2268.9	907	11.49

- Use **datetime business calendars**

bcalstocks.stbcal

```
 version 17
dateformat dmy

range 03jan2017 31aug2022
centerdate 03jan2017

omit dayofweek (Sa Su)
omit date  1jan*
omit date  4jul*
omit dowinmonth +3 Mo of jan
omit dowinmonth +3 Mo of feb
omit dowinmonth -1 Mo of may
omit dowinmonth +1 Mo of sep
omit dowinmonth +4 Th of Nov
omit date 24dec2021
omit date 25dec*
omit date 14apr2017
omit date 30mar2018
omit date  5dec2018
omit date 19apr2019
omit date 10apr2020
omit date  3jul2020
omit date  2apr2021
omit date  5jul2021
omit date 15apr2022
omit date 20jun2022
```

Business Calendar for the stocks data

```
. generate bdate=bofd("bcalstocks",daten)
(639 missing values generated)
. format bdate %tbbcalstocks
. tsset bdate
Time variable: bdate, 03jan2017 to 19aug2022
        Delta: 1 day
. drop if bdate==.
(639 observations deleted)
. list daten bdate sp500 bitcoin vix  in 1/8,sep(4)
```

	daten	bdate	sp500	bitcoin	vix
1.	03jan2017	03jan2017	2257.83	1020.67	12.85
2.	04jan2017	04jan2017	2270.75	1130.3	11.85
3.	05jan2017	05jan2017	2269	1007	11.67
4.	06jan2017	06jan2017	2276.98	895.71	11.32
5.	09jan2017	09jan2017	2268.9	902.66	11.56
6.	10jan2017	10jan2017	2268.9	907	11.49
7.	11jan2017	11jan2017	2275.32	795.77	11.26
8.	12jan2017	12jan2017	2270.44	812.25	11.54

Preparing your data format with date functions

- Use the `date()` function to convert your string dates to a numerical date
- Use one of the following functions to convert your daily date to your desired frequency

`wofd(`%td_daily_date_exp`)`	returns %tw date
`mofd(`%td_daily_date_exp`)`	returns %tm date
`qofd(`%td_daily_date_exp`)`	returns %tq date
`hofd(`%td_daily_date_exp`)`	returns %th date
`yofd(`%td_daily_date_exp`)`	returns %ty date

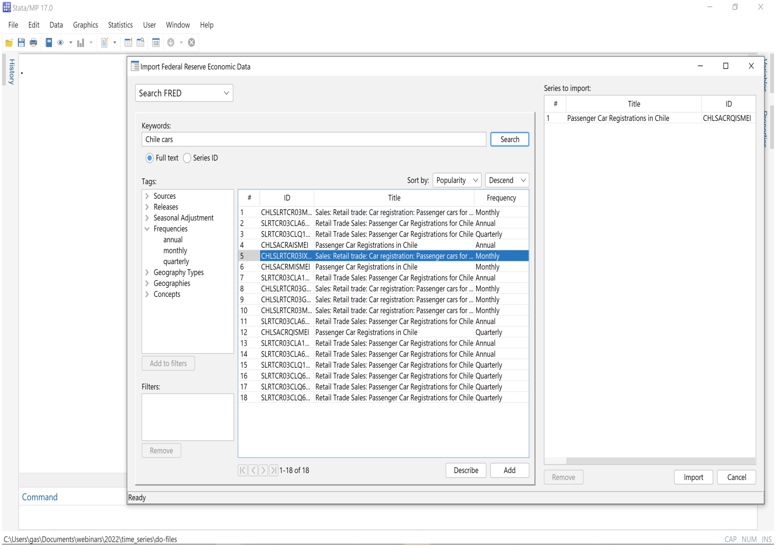
- Then use `tsset` to specify your time variable

Example 2: Passenger car registrations in Chile

- Get data from the Federal Reserve Economic Data (FRED).

```
 . import fred CHLSACRQISMEI, daterange(1990-01-01 2010-12-31) ///
>          aggregate(quarterly) clear
Summary
-------------------------------------------------------------------------
Series ID                    Nobs    Date range                Frequency
-------------------------------------------------------------------------
CHLSACRQISMEI                68      1994-01-01 to 2010-10-01  Quarterly
-------------------------------------------------------------------------
# of series imported: 1
   highest frequency: Quarterly
    lowest frequency: Quarterly

.
. rename CHLSACRQISMEI  cars_cl
. describe
Contains data
 Observations:            68
    Variables:             3
-------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
-------------------------------------------------------------------------
datestr         str10   %-10s                 observation date
daten           int     %td                   numeric (daily) date
cars_cl         float   %9.0g                 Passenger Car Registrations in
                                                Chile
-------------------------------------------------------------------------
Sorted by: datestr
     Note: Dataset has changed since last saved.
```

`import fred`: Dialog box



Example 2: Continue (Date functions)

```
. generate quarter = qofd(daten)
. format quarter %tq

.
. /* Specify time-series structure */
. tsset quarter
Time variable: quarter, 1994q1 to 2010q4
        Delta: 1 quarter

.
. list daten quarter cars_cl if tin(1994q1,1994q4)
```

	daten	quarter	cars_cl
1.	01jan1994	1994q1	39.41069
2.	01apr1994	1994q2	37.96296
3.	01jul1994	1994q3	37.82406
4.	01oct1994	1994q4	38.48565

Time-series operators

- Once you specify the time identifier variable with `tsset`, you can then use time-series operators on the variables
  - `L` is the lag operator; `L`k`.x`$=x_{t-k}$
  - `D` is the first-difference operator; `D.x` $= \Delta x_t = x_t - x_{t-1}$
    - `D`k`.x` $= \Delta^k x_t = \Delta \cdots (\Delta(\Delta x_t)) \cdots)$
  - `F` is lead operator; `F`k`.x` $= x_{t+k}$
  - `S` is the seasonal-difference operator; `S`k`.x` $= x_t - x_{t-k}$
  - `L(`*numlist*`).(`*varlist*`)`

```
. list quarter cars_cl L.cars_cl L2.cars_cl F.cars_cl D.cars_cl ///
>      if tin(1994q1,1994q4), noobs
```

quarter	cars_cl	L. cars_cl	L2. cars_cl	F. cars_cl	D. cars_cl
1994q1	39.41069	.	.	37.96296	.
1994q2	37.96296	39.41069	.	37.82406	-1.447731
1994q3	37.82406	37.96296	39.41069	38.48565	-.1388969
1994q4	38.48565	37.82406	37.96296	46.53998	.6615829

Time-series operators

- Once you specify the time identifier variable with `tsset`, you can then use time-series operators on the variables
  - `L` is the lag operator; `L`k`.x`$=x_{t-k}$
  - `D` is the first-difference operator; `D.x` $= \Delta x_t = x_t - x_{t-1}$
    - `D`k`.x` $= \Delta^k x_t = \Delta \cdots (\Delta(\Delta x_t)) \cdots)$
  - `F` is lead operator; `F`k`.x` $= x_{t+k}$
  - `S` is the seasonal-difference operator; `S`k`.x` $= x_t - x_{t-k}$
  - `L(`*numlist*`).(`*varlist*`)`

```
. list quarter cars_cl L.cars_cl L2.cars_cl F.cars_cl D.cars_cl ///
>       if tin(1994q1,1994q4), noobs
```

quarter	cars_cl	L. cars_cl	L2. cars_cl	F. cars_cl	D. cars_cl
1994q1	39.41069	.	.	37.96296	.
1994q2	37.96296	39.41069	.	37.82406	-1.447731
1994q3	37.82406	37.96296	39.41069	38.48565	-.1388969
1994q4	38.48565	37.82406	37.96296	46.53998	.6615829

Example 2: Continue (Time-series graphs)

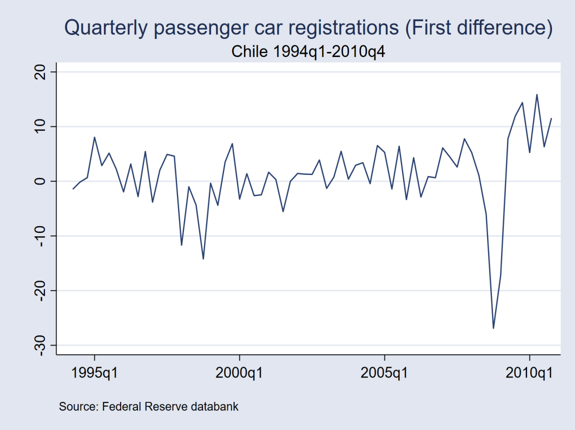
- You can also use the `tsline` command, which recognizes the time structure previously specified with `tsset`:

```
 . tsline cars_cl,                                                    ///
>    title("Quarterly total passenger car registrations")    ///
>    subtitle("Chile 1994q1-2010q4") ytitle(" ") xtitle(" ") ///
>     note(Source: Federal Reserve databank)
```

Example 1: Continue (Time-series graphs)

- Let's also plot the first difference of the series:

```
. tsline D.cars_cl,
>    title("Quarterly passenger car registrations (First difference)")
>    subtitle("Chile 1994q1-2010q4") ytitle(" ") xtitle(" ")
>     note(Source: Federal Reserve databank)
```



ARIMA models

ARMA models

- In general, ARIMA models assume that we are working with weakly stationary time series.
- These models can be understood as combinations of white noise ε_t processes (which, in short, are stationary processes with zero mean, constant variance, and covariances that don't depend on t).
- These are the models we are considering here:

$$\begin{aligned} \text{AR(1):} \quad & y_t = \phi y_{t-1} + \varepsilon_t \\ \text{MA(1):} \quad & y_t = \varepsilon_t + \theta \varepsilon_{t-1} \end{aligned}$$

$$\begin{aligned} \text{AR(p):} \quad & y_t = \phi_1 y_{t-1} + \phi_2 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t \\ \text{MA(q):} \quad & y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-1} + \cdots + \theta_p \varepsilon_{t-q} \end{aligned}$$

$$\text{ARMA(p,q):} \quad y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_p \varepsilon_{t-q}$$

ARMA models

- In general, ARIMA models assume that we are working with weakly stationary time series.
- These models can be understood as combinations of white noise ε_t processes (which, in short, are stationary processes with zero mean, constant variance, and covariances that don't depend on t).
- These are the models we are considering here:

$$\begin{aligned} \text{AR(1):} \quad & y_t = \phi y_{t-1} + \varepsilon_t \\ \text{MA(1):} \quad & y_t = \varepsilon_t + \theta \varepsilon_{t-1} \end{aligned}$$

$$\begin{aligned} \text{AR(p):} \quad & y_t = \phi_1 y_{t-1} + \phi_2 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t \\ \text{MA(q):} \quad & y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-1} + \cdots + \theta_p \varepsilon_{t-q} \end{aligned}$$

$$\text{ARMA(p,q):} \quad y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_p \varepsilon_{t-q}$$

ARMA models

- In general, ARIMA models assume that we are working with weakly stationary time series.
- These models can be understood as combinations of white noise ε_t processes (which, in short, are stationary processes with zero mean, constant variance, and covariances that don't depend on t).
- These are the models we are considering here:

$$\text{AR(1):} \qquad y_t = \phi y_{t-1} + \varepsilon_t$$
$$\text{MA(1):} \qquad y_t = \varepsilon_t + \theta \varepsilon_{t-1}$$

$$\text{AR(p):} \qquad y_t = \phi_1 y_{t-1} + \phi_2 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$
$$\text{MA(q):} \qquad y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-1} + \cdots + \theta_p \varepsilon_{t-q}$$

$$\text{ARMA(p,q):} \qquad y_t = \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_p \varepsilon_{t-q}$$

Two Important Concepts

- A time series is $\{y_t\}_{t=1}^{T}$ **weakly stationary** if $E(y_t)$, $E(y_t^2)$, and $E(y_t y_{t-j})$ do not depend on t.
- For identically distributed and not independent time series, one way to measure the dependence is the autocorrelation function:

$$
\begin{aligned}
\rho_l &= \frac{Cov(y_t y_{t-l})}{Var(y_t)} \\
\rho_l &= \frac{Cov(y_t y_{t+l})}{Var(y_t)} \\
\rho_l &\equiv \frac{\gamma_l}{\gamma_0}
\end{aligned}
$$

- Another measure is the partial autocorrelation function. It measures the correlation between y_t and y_{t-l} removing the mutual dependence that occurred in the intermediate periods.

Two Important Concepts

- A time series is $\{y_t\}_{t=1}^{T}$ **weakly stationary** if $E(y_t)$, $E(y_t^2)$, and $E(y_t y_{t-j})$ do not depend on t.
- For identically distributed and not independent time series, one way to measure the dependence is the autocorrelation function:

$$\rho_l = \frac{Cov(y_t y_{t-l})}{Var(y_t)}$$

$$\rho_l = \frac{Cov(y_t y_{t+l})}{Var(y_t)}$$

$$\rho_l \equiv \frac{\gamma_l}{\gamma_0}$$

- Another measure is the partial autocorrelation function. It measures the correlation between y_t and y_{t-l} removing the mutual dependence that occurred in the intermediate periods.

Example 2: ARIMA model for Passenger car registrations

- Let's use the `corrgram` command to look at the correlogram for D.cars_cl:

```
. corrgram D.cars_cl,lags(20)

                                          -1       0       1 -1       0       1
 LAG       AC       PAC      Q     Prob>Q  [Autocorrelation]  [Partial autocor]
-------------------------------------------------------------------------------
1       0.3851   0.4003   10.388  0.0013
2       0.1556   0.0117   12.111  0.0023
3      -0.0876  -0.1742   12.665  0.0054
4      -0.2105  -0.1701   15.916  0.0031
5      -0.1665  -0.0143   17.983  0.0030
6      -0.1755  -0.1617   20.317  0.0024
7      -0.1792  -0.2584   22.792  0.0019
8      -0.0524  -0.0521   23.007  0.0034
9       0.1178   0.3449   24.113  0.0041
10      0.1141   0.1567    25.17  0.0050
11      0.1257   0.0633   26.475  0.0055
12      0.0403  -0.0801   26.611  0.0088
13     -0.0303  -0.1055    26.69  0.0137
14     -0.0219  -0.0203   26.732  0.0209
15     -0.0351   0.0194   26.841  0.0301
16     -0.1297  -0.3395   28.367  0.0286
17      0.0119   0.1622    28.38  0.0407
18      0.0189   0.0500   28.414  0.0560
19     -0.0014  -0.2319   28.414  0.0758
20      0.0002  -0.3739   28.414  0.0999
```

Example 2: Continue (Model selection)

- Let's fit a couple of models (AR1 and MA1) for D.cars_cl:

```
. arima D.cars_cl,ar(1)
. estimates store Model_AR1
. arima D.cars_cl,ma(1)
. estimates store Model_MA1
. /* Output omitted */
```

- And we now look at a summary table or the two models:

```
. estimates table Model_AR1 Model_MA1, star stats(aic bic) b(%9.2f)
```

Variable	Model_AR1	Model_MA1
cars_cl		
_cons	1.25	1.21
ARMA		
ar		
L1.	0.39***	
ma		
L1.		0.33**
sigma		
_cons	6.07***	6.17***
Statistics		
aic	438.06	440.13
bic	444.67	446.74

Legend: * p<0.05; ** p<0.01; *** p<0.001

Example 2: Continue (Final model)

- We now fit the selected model (AR1):

```
. arima D.cars_cl,ar(1) nolog
ARIMA regression
Sample: 1994q2 thru 2010q4                      Number of obs     =         67
                                                Wald chi2(1)      =      17.27
Log likelihood = -216.0282                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |                 OPG
   D.cars_cl | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
cars_cl      |
       _cons |   1.247205   1.410933     0.88   0.377    -1.518172    4.012582
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .3948356   .0950154     4.16   0.000     .2086088    .5810624
-------------+----------------------------------------------------------------
      /sigma |   6.074494   .3603246    16.86   0.000     5.368271    6.780717
------------------------------------------------------------------------------
Note: The test of the variance against zero is one sided, and the two-sided
      confidence interval is truncated at zero.
```

Autoregressive coditional heteroskedasticity

Conditional Heteroskedasticity

- So far we have not modelled variances explicitly. In the stationary case we assumed they were constant.
- Evidently, time series experience different degrees of volatility in different periods.
- It is important not only to model the mean but the variance of a time series.
- Autoregressive conditional heteroskedasticity models capture this idea.

ARCH MODELS

- The general framework can be written as:

$$\begin{aligned} y_t &= x_t\beta + ARMA(p,q) + \varepsilon_t \\ Var\left(\sigma_t^2\right) &= \gamma_0 + A(\sigma,\varepsilon) + B(\sigma,\varepsilon)^2 \end{aligned}$$

- The original ARCH(m) model was written as:

$$\begin{aligned} y_t &= x_t\beta + \varepsilon_t \\ \sigma_t^2 &= \gamma_0 + \gamma_1\varepsilon_{t-1}^2 + \gamma_2\varepsilon_{t-2}^2 + \ldots + \gamma_m\varepsilon_{t-1}^2 \end{aligned}$$

- The original GARCH(m,k) model is given by:

$$\begin{aligned} y_t &= x_t\beta + \varepsilon_t \\ \sigma_t^2 &= \gamma_0 + \gamma_1\varepsilon_{t-1}^2 + \gamma_2\varepsilon_{t-2}^2 + \ldots + \gamma_m\varepsilon_{t-1}^2 + \delta_1\sigma_{t-1}^2 + \delta_2\sigma_{t-2}^2 + \ldots + \delta_k\sigma_{t-k}^2 \end{aligned}$$

ARCH MODELS

- The general framework can be written as:

$$\begin{aligned} y_t &= x_t\beta + ARMA(p,q) + \varepsilon_t \\ Var\left(\sigma_t^2\right) &= \gamma_0 + A(\sigma,\varepsilon) + B(\sigma,\varepsilon)^2 \end{aligned}$$

- The original ARCH(m) model was written as:

$$\begin{aligned} y_t &= x_t\beta + \varepsilon_t \\ \sigma_t^2 &= \gamma_0 + \gamma_1\varepsilon_{t-1}^2 + \gamma_2\varepsilon_{t-2}^2 + \ldots + \gamma_m\varepsilon_{t-1}^2 \end{aligned}$$

- The original GARCH(m,k) model is given by:

$$\begin{aligned} y_t &= x_t\beta + \varepsilon_t \\ \sigma_t^2 &= \gamma_0 + \gamma_1\varepsilon_{t-1}^2 + \gamma_2\varepsilon_{t-2}^2 + \ldots + \gamma_m\varepsilon_{t-1}^2 + \delta_1\sigma_{t-1}^2 + \delta_2\sigma_{t-2}^2 + \ldots + \delta_k\sigma_{t-k}^2 \end{aligned}$$

ARCH MODELS

- The general framework can be written as:

$$\begin{aligned} y_t &= x_t\beta + ARMA(p,q) + \varepsilon_t \\ Var\left(\sigma_t^2\right) &= \gamma_0 + A(\sigma,\varepsilon) + B(\sigma,\varepsilon)^2 \end{aligned}$$

- The original ARCH(m) model was written as:

$$\begin{aligned} y_t &= x_t\beta + \varepsilon_t \\ \sigma_t^2 &= \gamma_0 + \gamma_1\varepsilon_{t-1}^2 + \gamma_2\varepsilon_{t-2}^2 + \ldots + \gamma_m\varepsilon_{t-1}^2 \end{aligned}$$

- The original GARCH(m,k) model is given by:

$$\begin{aligned} y_t &= x_t\beta + \varepsilon_t \\ \sigma_t^2 &= \gamma_0 + \gamma_1\varepsilon_{t-1}^2 + \gamma_2\varepsilon_{t-2}^2 + \ldots + \gamma_m\varepsilon_{t-1}^2 + \delta_1\sigma_{t-1}^2 + \delta_2\sigma_{t-2}^2 + \ldots + \delta_k\sigma_{t-k}^2 \end{aligned}$$

Example 3: ARCH model for Mexican Pesos to U.S. Dollar

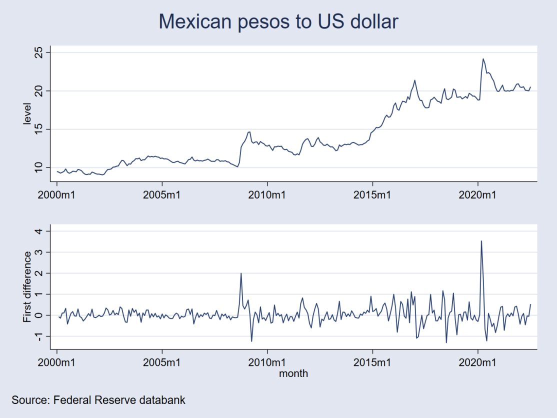
- Get data from the Federal Reserve Economic Data (FRED).

```
. import fred EXMXUS, daterange(2000-01-01 2022-08-15)
Summary
-----------------------------------------------------------------------------
Series ID                    Nobs    Date range                 Frequency
-----------------------------------------------------------------------------
EXMXUS                       271     2000-01-01 to 2022-07-01   Monthly
-----------------------------------------------------------------------------
# of series imported: 1
   highest frequency: Monthly
    lowest frequency: Monthly

.
. generate month=mofd(daten)

. label var month "numeric (monthly) date"

. rename EXMXUS exchrate_mx

.
. tsset month,monthly

Time variable: month, 2000m1 to 2022m7
        Delta: 1 month

.
. list in 1/4,abbreviate(12)

     +---------------------------------------------------+
     | datestr           daten   exchrate_mx     month   |
     |---------------------------------------------------|
  1. | 2000-01-01    01jan2000        9.4935    2000m1   |
  2. | 2000-02-01    01feb2000        9.4265    2000m2   |
  3. | 2000-03-01    01mar2000        9.2886    2000m3   |
  4. | 2000-04-01    01apr2000        9.3937    2000m4   |
     +---------------------------------------------------+
```

Example 3: Continue (Time-series graph)

- You can combine two `tsline` graphs to plot the levels and first difference of the exchange rate series:

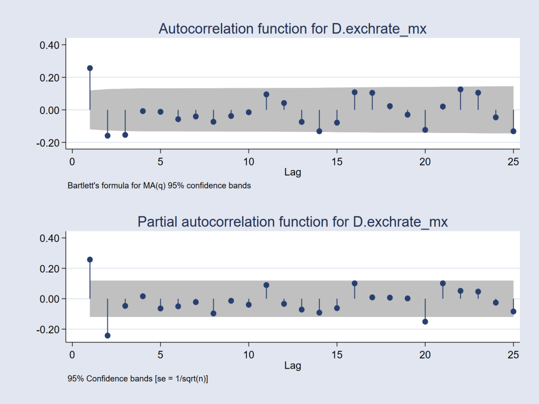
```
 . tsline exchrate_mx, name(level,replace) ytitle(level) xtitle("")
. tsline D.exchrate_mx, name(difference,replace)                  ///
>                          ytitle(First difference) xtitle("month")
. graph combine level difference,rows(2)                          ///
>      title("Mexican pesos to US dollar")                        ///
>      note(Source: Federal Reserve databank)
```



Example 3: (Continue)

- Use `ac` and `pac` to look at the autocorrelation and partial autocorrelation functions for dexchrate_mx:

```
 . ac D.exchrate_mx,lags(25) ytitle(" ")  name(ac_dexchrate_mx)   ///
>    title("Autocorrelation function for D.exchrate_mx")
. pac D.exchrate_mx,lags(25) ytitle(" ") name(pac_dexchrate_mx)   ///
>     title("Partial autocorrelation function for D.exchrate_mx")
.
. graph combine ac_dexchrate_mx pac_dexchrate_mx,rows(2)
```



Example 3: Model selection from three alternative specifications:

```
 . arch D.exchrate_mx,arch(1) garch(1) ar(1/2) nolog
. estimates store arch_ar1_2
. arch D.exchrate_mx,arch(1) garch(1) ma(1) nolog
. estimates store arch_ma1
. arch D.exchrate_mx,arch(1) garch(1) ma(1) ar(1/2) nolog
. estimates store arch_arma

 . estimates table arch_ar1_2 arch_ma1 arch_arma, ///
>    star stats(aic bic) b(%9.2f) vsquish
```

Variable	arch_ar1_2	arch_ma1	arch_arma
exchrate_mx			
_cons	0.05	0.05	0.04
ARMA			
ar			
L1.	0.37***		0.83**
L2.	-0.20*		-0.34**
ma			
L1.		0.33***	-0.48
ARCH			
arch			
L1.	0.10***	0.10***	0.11***
garch			
L1.	0.91***	0.91***	0.91***
_cons	0.00**	0.00**	0.00**
Statistics			
aic	257.92	260.33	257.25
bic	279.51	278.33	282.44

Legend: * p<0.05; ** p<0.01; *** p<0.001

Example 3: Continue (Final ARCH model)

- We now fit the selected model:

```
. arch D.exchrate_mx,arch(1) garch(1) ar(1/2) nolog vsquish
ARCH family regression -- AR disturbances
Sample: 2000m2 thru 2022m7                    Number of obs     =        270
                                              Wald chi2(2)      =      20.63
Log likelihood = -122.9579                    Prob > chi2       =     0.0000

------------------------------------------------------------------------------
D.           |                OPG
 exchrate_mx | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
exchrate_mx  |
       _cons |   .0474618   .0302807     1.57   0.117    -.0118872    .1068108
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .3713764    .087179     4.26   0.000     .2005087    .5422441
         L2. |  -.1954443   .0765553    -2.55   0.011      -.34549   -.0453987
-------------+----------------------------------------------------------------
ARCH         |
        arch |
         L1. |    .100711   .0234196     4.30   0.000     .0548095    .1466125
       garch |
         L1. |     .91257   .0113487    80.41   0.000     .8903269    .9348132
       _cons |   .0020402   .0006335     3.22   0.001     .0007986    .0032819
------------------------------------------------------------------------------
```

Vector autoregressive (VAR) model

Vector Autoregressive Models VAR

- VARs are extensions of AR(p) models for vector valued dependent variables with no structural form.
- A VAR model can be written as:

$$\mathbf{y_t} = \mathbf{v} + \mathbf{A_1 y_{t-1}} + \ldots + \mathbf{A_p y_{t-p}} + \mathbf{C_0 x_t} + \mathbf{C_1 x_{t-1}} + \ldots + \mathbf{C_s x_{t-s}} + \mathbf{u_t}$$

Where:

$\mathbf{y}_t = (y_{1t}, ..., y_{Kt})$ is a Kx1 random vector
$\mathbf{A_1}$ through $\mathbf{A_p}$ are KxK matrices of parameters.
$\mathbf{x_t}$ is an Mx1 vector of exogenous variables
$\mathbf{C_0}$ through $\mathbf{C_s}$ are KxM matrices of parameters.
$\mathbf{v}$ is a Kx1 vector of parameters
$\mathbf{u_t}$ is a vector assumed to be white noise:

$$\begin{aligned} E(\mathbf{u_t}) &= \mathbf{0} \\ E(\mathbf{u_t u_t}') &= \mathbf{\Sigma} \\ E(\mathbf{u_t u_s}') &= \mathbf{0} \;;\; t \neq s \end{aligned}$$

- The number of coefficients is quadratic to the number of dependent variables and proportional the number of lags.

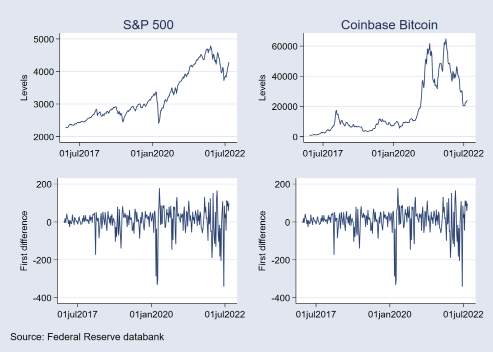
Example 4: Vector autoregressive (VAR) model for stock data

- We use `import fred` to get data from the Federal Reserve Economic Data (FRED).

```
 . import fred SP500 CBBTCUSD VIXCLS, daterange(2017-01-02 .) ///
>     aggregate(weekly,avg) clear
. rename SP500 sp500
. rename CBBTCUSD bitcoin
. rename VIXCLS vix

 . describe
Contains data
 Observations:           295
    Variables:             5
--------------------------------------------------------------------------------
Variable      Storage   Display    Value
    name         type    format    label      Variable label
--------------------------------------------------------------------------------
datestr         str10   %-10s                 observation date
daten           int     %td                   numeric (daily) date
sp500           float   %9.0g                 S&P 500
bitcoin         float   %9.0g                 Coinbase Bitcoin
vix             float   %9.0g                 CBOE Volatility Index: VIX
--------------------------------------------------------------------------------
Sorted by: datestr
     Note: Dataset has changed since last saved.

.
. tsset daten,delta(7 days)
Time variable: daten, 06jan2017 to 26aug2022
        Delta: 7 days
```

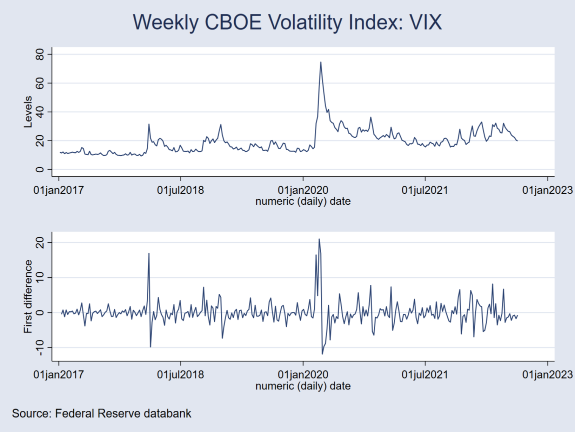
Endogenous variables for VAR model

```
 . tsline sp500,name(sp_levels,replace) ytitle("Levels")          ///
>      title("S&P 500") ylabels(#3,angle(0)) xlabels(#3) xtitle("")
. tsline D.sp500,name(sp_diff,replace) ytitle("First difference")     ///
>      ylabels(#3,angle(0)) xlabels(#3) xtitle("")
. tsline bitcoin,name(bitcoin_levels,replace) ytitle("Levels")       ///
>      title("Coinbase Bitcoin") ylabels(#3,angle(0)) xlabels(#3) xtitle("")
. tsline D.sp500,name(bitcoin_diff,replace) ytitle("First difference") ///
>      ylabels(#3,angle(0)) xlabels(#3) xtitle("")
.
. graph combine sp_levels bitcoin_levels sp_diff bitcoin_diff,rows(2)  ///
>      note(Source: Federal Reserve databank)
```



Exogenous variable for VAR model

```
 . tsline vix,name(vix_levels,replace) ytitle("Levels")
. tsline D.vix,name(vix_diff,replace) ytitle("First difference")
.
. graph combine vix_levels vix_diff,            ///
>      title("Weekly CBOE Volatility Index: VIX")      ///
>      note(Source: Federal Reserve databank) rows(2)
```

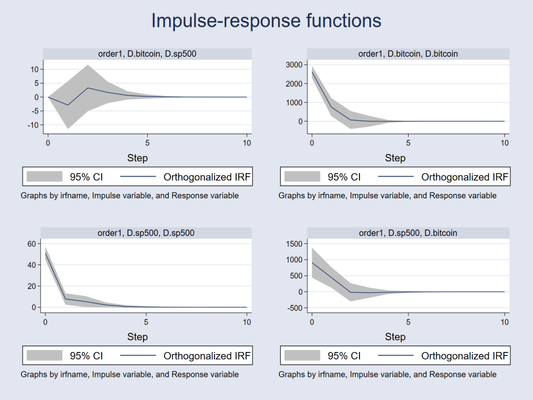


VAR model for S&P500 and Bitcoin with exogenous VIX variable

```
. var D.sp500 D.bitcoin if tin(06mar2020,19aug2022), exog(D.vix) vsquish
Vector autoregression
Sample: 06mar2020 thru 19aug2022                Number of obs     =       129
Log likelihood =  -1887.785                     AIC               =  29.45404
FPE            =   2.12e+10                     HQIC              =  29.56213
Det(Sigma_ml)  =   1.76e+10                     SBIC              =  29.72007
Equation          Parms      RMSE      R-sq      chi2     P>chi2
----------------------------------------------------------------
D_sp500             6      52.1009   0.6578   248.0081   0.0000
D_bitcoin           6      2828.93   0.1556   23.76526   0.0002
----------------------------------------------------------------

------------------------------------------------------------------------------
             |
             | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
D_sp500      |
       sp500 |
         LD. |   .1722277   .0550092     3.13   0.002     .0644116    .2800437
        L2D. |   .0567062   .0552017     1.03   0.304    -.0514871    .1648995
     bitcoin |
         LD. |  -.0011056   .0016788    -0.66   0.510     -.004396    .0021849
        L2D. |   .0017677   .0016631     1.06   0.288     -.001492    .0050275
         vix |
         D1. |  -16.62902   1.120476   -14.84   0.000    -18.82511   -14.43292
       _cons |   6.235951   4.497384     1.39   0.166    -2.578759    15.05066
-------------+----------------------------------------------------------------
D_bitcoin    |
       sp500 |
         LD. |   3.637138   2.986841     1.22   0.223    -2.216963    9.491239
        L2D. |  -2.475878   2.997291    -0.83   0.409    -8.350461    3.398705
     bitcoin |
         LD. |   .2891966   .0911558     3.17   0.002     .1105346    .4678586
        L2D. |  -.0539655   .0903042    -0.60   0.550    -.2309585    .1230275
         vix |
         D1. |  -154.3945   60.83859    -2.54   0.011    -273.6359   -35.15304
       _cons |   61.27406   244.1949     0.25   0.802    -417.3392    539.8873
------------------------------------------------------------------------------

. estimates store eq_var
```

Impulse-response functions

```
. irf create order1, step(10) set(myirf1,replace)
. irf graph oirf
. irf graph oirf,impulse(D.bitcoin) response(D.sp500) name(i_bit__r_sp)
. irf graph oirf,impulse(D.bitcoin) response(D.bitcoin) name(i_bit__r_bit)
. irf graph oirf,impulse(D.sp500) response(D.sp500) name(i_sp__r_sp)
. irf graph oirf,impulse(D.sp500) response(D.bitcoin) name(i_sp__r_bit)
.
. graph combine i_bit__r_sp i_bit__r_bit i_sp__r_sp i_sp__r_bit, ///
>       title("Impulse-response functions")
```



Solving and forecasting systems of equations

Solving models for a collection of equations

- Components
  - Stochastic equations fit using estimation commands
  - Identities
  - Coefficient vectors
- Solving the model
  - Obtain static or dynamic forecasts
  - Alternative forecast scenarios
- `forecast` command

`forecast` subcommands

- Building the model
  - `create`
  - `estimates`
  - `identity`
  - `coefvector`
  - `exogenous`
- Solving the model
  - `solve`
  - `adjust`
- Utilities
  - `describe`
  - `list`
  - `clear`
  - `drop`
  - `query`

Example 5: System of equations VAR and ARIMA

- We now combine the results for the VAR model in example 4 with an ARIMA model for the exogenous variable vix.
- Let's fit an ARIMA model for vix and store the results:

```
. arima D.vix if tin(06mar2020,19aug2022), ar(1) nolog
ARIMA regression
Sample: 06mar2020 thru 19aug2022                Number of obs     =        129
                                                Wald chi2(1)      =      28.33
Log likelihood = -359.4667                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |                 OPG
       D.vix | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
vix          |
       _cons |  -.0823354   .5566037    -0.15   0.882    -1.173259    1.008588
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .2379425   .0447051     5.32   0.000     .1503222    .3255629
-------------+----------------------------------------------------------------
      /sigma |   3.925202   .1242906    31.58   0.000     3.681597    4.168807
------------------------------------------------------------------------------
Note: The test of the variance against zero is one sided, and the two-sided
      confidence interval is truncated at zero.

.
. estimates store eq_ar1
```

Example 5: Continue

- Use the suite of `forecast` commands to solve the model.

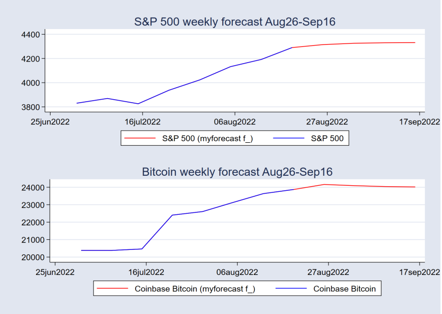
```
. tsappend, add(4)
. forecast create myforecast
  Forecast model myforecast started.
. forecast estimates eq_var,names(dsp500 dbitcoin)
  Added estimation results from var.
  Forecast model myforecast now contains 2 endogenous variables.
. forecast estimates eq_ar1,names(dvix)
  Added estimation results from arima.
  Forecast model myforecast now contains 3 endogenous variables.
. forecast identity sp500=dsp500+L.sp500
  Forecast model myforecast now contains 4 endogenous variables.
. forecast identity bitcoin=dbitcoin+L.bitcoin
  Forecast model myforecast now contains 5 endogenous variables.
. forecast identity vix=dvix+L.vix
  Forecast model myforecast now contains 6 endogenous variables.
. forecast solve, begin(td(26aug2022)) end(td(16sep2022)) prefix(f_)
Computing dynamic forecasts for model myforecast.
_______________________________________________

Starting period:  26aug2022
Ending period:    16sep2022
Forecast prefix:  f_

26aug2022:  ................
02sep2022:  ................
09sep2022:  ................
16sep2022:  ................

Forecast 6 variables spanning 4 periods.
_______________________________________
```

Example 5: Continue

- Let's look at the forecasts for the next 4 weeks.

```
. list daten sp500 f_sp500 bitcoin f_bitcoin vix f_vix ///
>      if tin(29jul2022,16sep2022), noobs
```

daten	sp500	f_sp500	bitcoin	f_bitc~n	vix	f_vix
29jul2022	4022.84	4022.84	22607.86	22607.86	22.99	22.99
05aug2022	4132.42	4132.42	23129.3	23129.3	22.26	22.26
12aug2022	4192.04	4192.04	23635.53	23635.53	20.51	20.51
19aug2022	4277.72	4277.72	23440.96	23440.96	19.94	19.94
26aug2022	.	4306.502	.	23613.29	.	19.74163
02sep2022	.	4323.848	.	23644.43	.	19.63168
09sep2022	.	4336.452	.	23710.96	.	19.54278
16sep2022	.	4347.219	.	23805.65	.	19.45888



Summing up

- Stata tools
- Inputing and managing time series in Stata
- Univariate time-series analysis
    - ARIMA models
    - ARCH models
- Multivariate time-series analysis with
    - Vector autoregressive (VAR) model
- Forecasting systems of equations

Proceedings for presentations on time series using Stata

- Technical tips on time series with Stata:

  https://www.stata.com/meeting/mexico11/materials/gsanchez.pdf
- Cointegrating VAR and probability forecasting:

  https://www.stata.com/meeting/spain12/abstracts/materials/Sanchez.pdf
- Forecasting tools in Stata

  https://www.stata.com/meeting/mexico13/abstracts/materials/mex13_sanchez_forecast.pdf
- Markov-switching regression models in Stata

  https://www.stata.com/meeting/spain15/abstracts/materials/spain15_sanchez.pdf
- Bayesian VAR models in Stata:

  https://www.stata.com/meeting/mexico21/slides/Mexico21_Sanchez.pdf