Teaching Financial Econometrics in Stata

Carlos Alberto Dorantes, Tec de Monterrey

EUSMEX 2018

Outline

The getsymbols command

The mvport ssc package

Teaching Financial Econometrics with Stata

- Descriptive statistics
- Histograms
- Linear relationships
- Market Regression model
- Writing a command for CAPM model
- Moving CAPM betas
- Illustrating diversification using portfolios
- Portfolio optimization
- Portfolio strategy based on CAPM
- Forecasting with ARIMA/SARIMA
- Volatility models

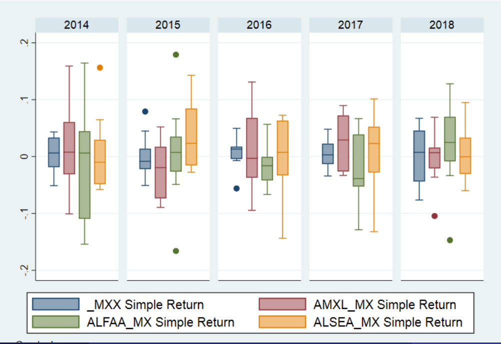
getsymbols command

getsymbols downloads data from Quandl, Yahoo Finance, and Alpha Vantage. Here an example:

```
. capture getsymbols BTC-USD, fy(2017) yahoo clear
. tsline close_BTC_USD
```

Getting data from several assets from Yahoo:

```
. capture getsymbols ^MXX AMXL.MX ALFAA.MX ALSEA.MX, fy(2014) ///
>                    freq(m) yahoo clear price(adjclose)
. *With the price option, returns are calculated
. qui gen year=yofd(dofm(period))
. graph box R_*, by(year, rows(1))
```



The mvport package

The mvport package has 11 commands:

- meanrets and varrets for estimation of expected mean returns and variance-covariance matrix
- mvport, gmvport, ovport for portfolio optimization
- efrontier identifies the efficient frontier based on a set of assets
- cmline calculates both the efficient frontier and the Capital Market Line
- backtest and cbacktest for portfolio backtesting
- holdingrets calculates holding period returns of assets
- simport simulates portfolios with random weights and estimates expected risk and return

Teaching Financial Econometrics/Progamming

- Descriptive statistics
- Histograms
- Plotting linear relationships
- Market Regression model
- Writing a command for CAPM model
- Moving CAPM betas
- Illustrating diversification
- Portfolio optimization
- Portfolio backtesting
- Portfolio strategy based on CAPM
- Forecasting with ARIMA/SARIMA
- Volatility models
- VAR models

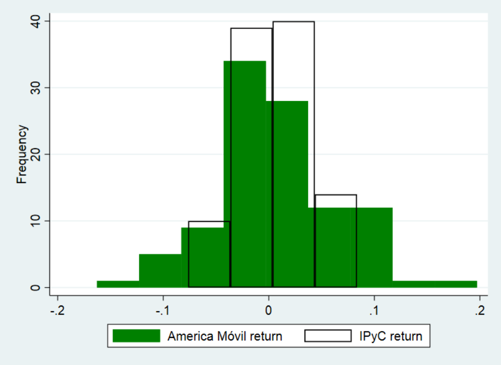
Descriptive statistics

Descriptive statistics of returns

```
. capture getsymbols ^MXX AMXL.MX, fy(2014) ///
>                    freq(m) yahoo clear price(adjclose)

. su R_AMXL R__MXX

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
    R_AMXL_MX |         55    .0058009    .0587706  -.1045638   .1591078
      R__MXX |         55    .0039337    .0324539  -.0764216   .0791098

.
```

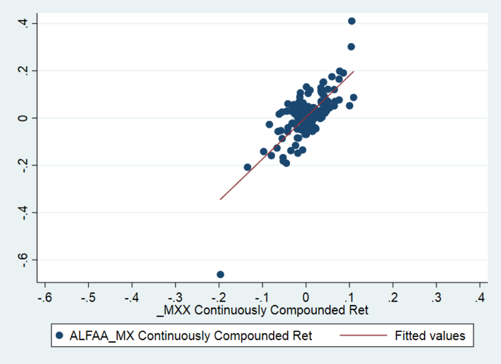
Histograms

```
. capture getsymbols ^MXX AMXL.MX, fy(2010) ///
>                       freq(m) yahoo clear price(adjclose)
. twoway (hist R_AMXL, width(0.04) freq color(green) ) ///
> (hist R__MXX, width(0.04) freq fcolor(none) lcolor(black)), ///
> legend(order(1 "America Móvil return" 2 "IPyC return" ))
```



How ALFA returns are related to the market returns:

```
.  capture getsymbols ^MXX ALFAA.MX, fy(2008) ///
>                    freq(m) yahoo clear price(adjclose)
.  twoway (scatter r_ALFAA_MX r__MXX) ///
>    (lfit r_ALFAA_MX r__MXX), xlabel(-0.60(0.10) 0.40)
```



Market regression model

I run a regression model to evaluate the relationship between market returns and Alfa returns:

```
.    capture getsymbols ^MXX ALFAA.MX, fy(2008) ///
>                   freq(m) yahoo clear price(adjclose)
.    qui reg r_ALFAA_MX r__MXX
.    _coef_table
```

r_ALFAA_MX	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
r__MXX	1.767932	.1395922	12.66	0.000	1.491662	2.044203
_cons	.0034723	.006395	0.54	0.588	-.0091841	.0161288

The market beta of Alfa is 1.77

The alpha coefficient of Alfa is 0.003472

Writing a command for CAPM model

```
. capture program drop capm

. program define capm, rclass
syntax varlist(min=3 numeric) [if]
local stockret: word 1 of `varlist´
local mktret: word 2 of `varlist´
local rfrate: word 3 of `varlist´
capture drop prem`stock´
quietly gen prem`stock´=`stockret´-`rfrate´
capture drop mktpremium
quietly gen mktpremium=`mktret´-`rfrate´
quietly reg prem`stock´ mktpremium `if´
matrix res= r(table)
local b1=res[1,1]
local b0=res[1,2]
local SEb1=res[2,1]
local SEb0=res[2,2]
local N=e(N)
dis "Market beta is " %3.2f `b1´ "; std error of beta is " %8.6f `SEb1´
dis "Alfa coeff. is " %8.6f `b0´ ", its std error is " %8.6f `SEb0´
return scalar b1=`b1´
return scalar b0=`b0´
return scalar SEb1=`SEb1´
return scalar SEb0=`SEb0´
return scalar N=`N´
end
```

...Writing a command for CAPM

Once I define the capm command I get data to run it:

The parameters of my capm command are:

1 - stock return 2 - market return 3 - risk-free return

```
.   *I have to get data for risk-free rate from the FED:
.   qui freduse INTGSTMXM193N, clear
.   * This monthly series has annual rates in %, so I create a monthly rate:
.   qui gen m_R_cetes = (INTGSTMXM193N/100)/12
.   * I calculate the continuously compounded return from the simple returns:
.   qui gen m_r_cetes = ln(1 + m_R_cetes)
.   ** I create the monthly variable :
.   qui gen period =mofd(daten)
.   format period %tm
.   * Now I indicate Stata that the time variable is period:
.   qui tsset period
.   * I save the CETES dataset as cetes:
.   qui save rfrate, replace
```

```
. * I get the stock data from Yahoo Finance:
. capture getsymbols ^MXX ALFAA.MX, fy(2008) ///
>                    freq(m) yahoo clear price(adjclose)
. * I merge it with the risk-free dataset:
. qui merge 1:1 period using rfrate, keepusing(m_r_cetes)
. qui drop if _merge!=3
. qui drop _merge
. qui save mydata1,replace
```

Now I can use my capm command:

```
.   capm r_ALFAA_MX r__MXX m_r_cetes
Market beta is 1.77; std error of beta is 0.141680
Alfa coeff. is 0.006548, its std error is 0.006508
.   return list
scalars:
                  r(N) =  124
               r(SEb0) =  .006508329271266
               r(SEb1) =  .141680230176724
                 r(b0) =  .0065479530113093
                 r(b1) =  1.765364922187618
```

Moving betas

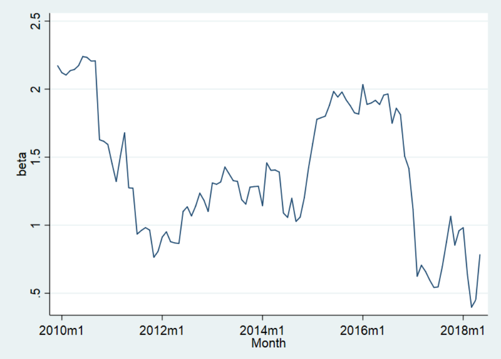
I can examine how market beta of a stock changes over time

I run my capm command using 24-month rolling windows:

```
.   rolling b1=r(b1) seb1=r(SEb1), window(24) saving(capmbetas,replace): ///
>                capm r_ALFAA_MX r__MXX m_r_cetes
(running capm on estimation sample)

Rolling replications (102)
────┼─── 1 ───┼─── 2 ───┼─── 3 ───┼─── 4 ───┼─── 5
..................................................    50
..................................................   100
..
file capmbetas.dta saved

.
```

. . . moving betas

```
.   qui use capmbetas,clear
.   label var b1 "beta"
.   label var seb1 "StdErr Beta"
.   label var end "Month"
.   qui tsset end
.   tsline b1
```



Illustrating diversification with portfolio weights

```
. capture getsymbols ALFAA.MX BIMBOA.MX, fy(2009) ///
>                    freq(m) yahoo clear price(adjclose)
. qui meanrets r_*
. matrix MEANRETS=r(meanrets)
. qui varrets r_*
. matrix COV=r(cov)
. clear
. * I create a dataset with 11 observations for 11 portfolios
. qui set obs 11
. qui egen double wa=fill(0(0.1)1)
. qui format wa %tg
. qui gen wb=1-wa
. qui format wb %tg
. *Now I computed both expected return and variance of the 11 portfolios
. qui gen ERP = wa*MEANRETS[1,1] + wb*MEANRETS[2,1]
. qui gen varP= wa^2*COV[1,1] + wb^2*COV[2,2] + 2*wa*wb*COV[2,1]
. * Now I compute the risk (standard dev) of the portfolios:
. qui gen sdP=sqrt(varP)
```

```
. twoway (scatter ERP sdP, msize(medlarge) mlabel(wa) mlabcolor(edkblue)), ///
>  ytitle(Retorno) ylabel(#5) xtitle(Riesgo) ///
>  title(Frontier using 2 stocks: Alfa and Bimbo) note(Alfa weights are doed fo
> r each portfolio)
```

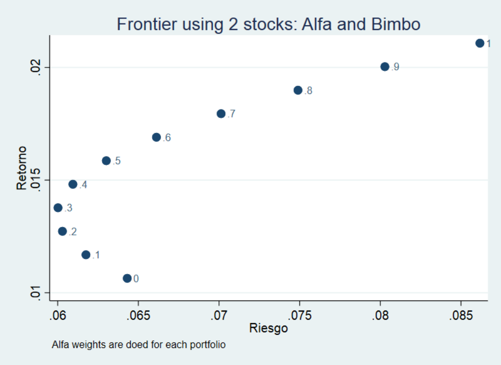


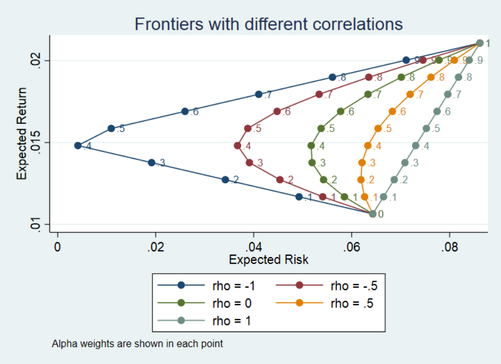
Figure 6: Diversification using 2 stocks and changing weights

Illustrating diversification with correlation

```
. local paramgraficas ""
. * I define a macro with the parameters for the twoway command
. local i=0
. * I loop over different values of correlation:
. forvalues rho1=-1(0.5)1 {
  2.    local i=`i´ + 1
  3.    * I generate the expected return of the 11 portfolios
.    qui gen ERP`i´ = wa*MEANRETS[1,1] + wb*MEANRETS[2,1]
  4.    * I assign a label for this new variable ERP`i´.
.    label variable ERP`i´ "rho = `rho1´"
  5.    * I create the variance of the 11 portfolios
.    qui gen varP`i´ = wa^2*COV[1,1] + wb^2*COV[2,2] + 2*wa*wb*`rho1´*sqrt(COV[
> 1,1])*sqrt(COV[2,2])
  6.    * Now I generate the corresponding risk variable
.    qui gen sdP`i´ = sqrt(varP`i´)
  7.    * Now I construct the parameters for the graph
.    local paramgraficas "`paramgraficas´(connected ERP`i´ sdP`i´, mlabel(wa))
> "
  8. }
```

. . . Illustrating diversification with correlation

```
. * Now I can graph all the frontiers generated according to
. * different hypothetical values of correlations:
. twoway `paramgraficas´, note(Alpha weights are shown in each point) ///
>    title(Frontiers with different correlations) ///
>    ytitle(Expected Return) xtitle(Expected Risk)
```



Portfolio optimization - efficient frontier

```
. capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX, ///
>                   fy(2014) freq(m) yahoo clear price(adjclose)

. qui efrontier r_*
```

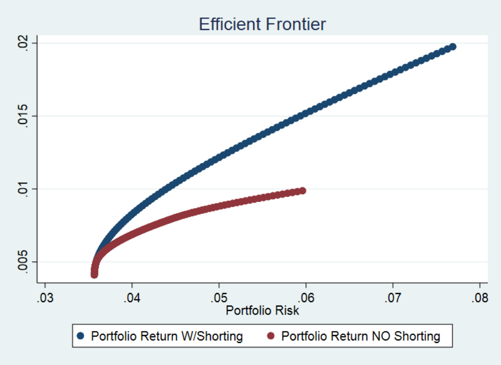


Figure 8: Efficient Frontier with 4 assets

Portfolio optimization - Capital Market Line

```
. capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX, ///
>                    fy(2014) freq(m) yahoo clear price(adjclose)
. qui cmline r_*, noshort
```

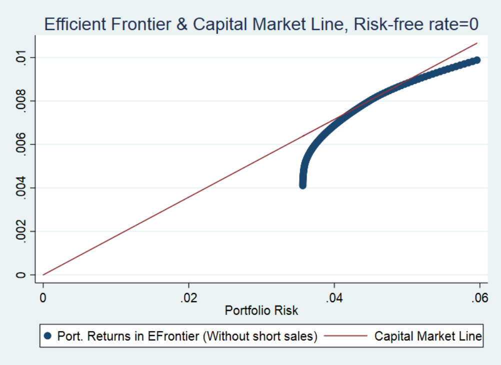


Figure 9: Capital Market Line with 4 assets

Portfolio optimization - GMV Portfolio

Finding the GMV Portfolio:

```
.  capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX ARA.MX OMAB.MX, ///
>                   fy(2014) freq(m) yahoo clear price(adjclose)
.  * Finding the GMV Portfolio without short sales:
.  gmvport r_*, noshort
Number of observations used to calculate expected returns and var-cov matrix :
> 56
The weight vector of the Global Minimum Variance Portfolio (NOT Allow Short Sal
> es) is:
               Weights
 r_ALFAA_MX  .01162011
r_BIMBOA_MX  .34878676
 r_ALSEA_MX  .14596726
  r_AMXL_MX   .2554765
   r_ARA_MX   .1966382
  r_OMAB_MX  .04151117
The return of the Global Minimum Variance Portfolio is: .00532839
The standard deviation (risk) of the Global Minimum Variance Portfolio is: .033
> 36875
```

Finding the GMV Portfolio with restrictions:

```
. capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX ARA.MX OMAB.MX, ///
>                  fy(2014) freq(m) yahoo clear price(adjclose)
. * Finding the GMV Portfolio with weights>5%:
. gmvport r_*, minw(0.05)
Number of observations used to calculate expected returns and var-cov matrix :
> 56
The weight vector of the Global Minimum Variance Portfolio (NOT Allow Short Sal
> es) is:
              Weights
 r_ALFAA_MX       .05
r_BIMBOA_MX  .33371175
 r_ALSEA_MX   .1418426
  r_AMXL_MX  .23301119
   r_ARA_MX  .19143446
  r_OMAB_MX       .05
The return of the Global Minimum Variance Portfolio is: .00504362
The standard deviation (risk) of the Global Minimum Variance Portfolio is: .033
> 48313
```

Finding the GMV Portfolio with restrictions:

```
.  capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX ARA.MX OMAB.MX, ///
>                  fy(2014) freq(m) yahoo clear price(adjclose)
.  * Finding the GMV Portfolio without short sales, and weights<25%:
.  gmvport r_*, maxw(0.25)
Number of observations used to calculate expected returns and var-cov matrix :
> 56
The weight vector of the Global Minimum Variance Portfolio (Allowing Short Sale
> s) is:
               Weights
 r_ALFAA_MX  .03060067
r_BIMBOA_MX        .25
 r_ALSEA_MX  .18964697
  r_AMXL_MX        .25
   r_ARA_MX  .21525427
  r_OMAB_MX  .06449809
The return of the Global Minimum Variance Portfolio is: .00600812
The standard deviation (risk) of the Global Minimum Variance Portfolio is: .033
> 70405
```

. . . Portfolio optimization - GMV Portfolio

Using EWMA to assign more weight to recent months:

```
. *Calculates expected returns using the EWMA method with lamda=0.94:
.  qui meanrets r_* , lew(0.94)
.  matrix mrets=r(meanrets)
. *Calculates the covariance matrix of returns using the EWMA method:
.  qui varrets r_* , lew(0.94)
.  matrix cov=r(cov)
. * Calculates the GMV portfolio using the calculated mean rets and COV:
. gmvport r_* , covm(cov) mrets(mrets)
Number of observations used to calculate expected returns and var-cov matrix :
> 56
The weight vector of the Global Minimum Variance Portfolio (Allowing Short Sale
> s) is:
               Weights
 r_ALFAA_MX  .01512741
r_BIMBOA_MX  .23834664
 r_ALSEA_MX  .21974859
  r_AMXL_MX  .24542343
   r_ARA_MX  .26241682
  r_OMAB_MX  .01893709
The return of the Global Minimum Variance Portfolio is: .00264197
The standard deviation (risk) of the Global Minimum Variance Portfolio is: .036
> 46787
```

Portfolio optimization - Optimal Portfolio

```
.  capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX ARA.MX OMAB.MX, ///
>                    fy(2014) freq(m) yahoo clear price(adjclose)
. * Finding the GMV Portfolio:
.  qui ovport r_*
.  di "The weights of the optimal portfolio are:"
The weights of the optimal portfolio are:
.  matlist r(weights),border
```

	Weights
r_ALFAA_MX	-.8202318
r_BIMBOA_MX	-.4078906
r_ALSEA_MX	.4993704
r_AMXL_MX	.521852
r_ARA_MX	-.1284891
r_OMAB_MX	1.335389

```
.  di "The expected return of the optimal portfolio is " r(rop)
The expected return of the optimal portfolio is .04467859
.  di "The expected risk of the optimal portfolio is " r(sdop)
The expected risk of the optimal portfolio is .09662549
```

```
.  capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX ARA.MX OMAB.MX, ///
>                   fy(2014) freq(m) yahoo clear price(adjclose)
. * Finding the GMV Portfolio without short sales:
.  ovport r_*, noshort
Number of observations used to calculate expected returns and var-cov matrix :
> 56
The weight vector of the Tangent Portfolio with a risk-free rate of 0 (NOT Allo
> w Short Sales) is:
               Weights
 r_ALFAA_MX          0
r_BIMBOA_MX          0
 r_ALSEA_MX  .13677267
  r_AMXL_MX   .1392179
   r_ARA_MX          0
  r_OMAB_MX  .72400943
The return of the Tangent Portfolio is: .01979399
The standard deviation (risk) of the Tangent Portfolio is: .05393653

The marginal contributions to risk of the assets in the Tangent Portfolio are:
              | Margina~k
--------------+-----------
  r_ALFAA_MX  |  .0267051
 r_BIMBOA_MX  |    .01914
  r_ALSEA_MX  |  .0268677
   r_AMXL_MX  |  .0111481
    r_ARA_MX  |   .020827
   r_OMAB_MX  |  .0672778
```

Portfolio optimization - Minimum Variance Portfolio

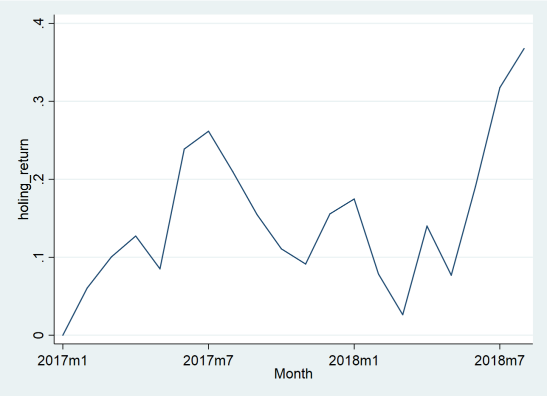
```
.  capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX ARA.MX OMAB.MX, ///
>                  fy(2014) freq(m) yahoo clear price(adjclose)
. * Finding the MV Portfolio given a specific required return:
.  mvport r_*, ret(0.02)
Portfolio weights of the portfolio:
             |    Weights
-------------+-----------
   r_ALFAA_MX |  -.2985335
  r_BIMBOA_MX |   .0666618
   r_ALSEA_MX |   .2777326
    r_AMXL_MX |   .3547938
     r_ARA_MX |   .0754154
    r_OMAB_MX |   .5239299
Number of observations used to calculate expected returns and var-covariance ma
> trix : 56
Required return of the Portfolio:  .02
Minimum standard deviation of the portfolio (Allowing for short sales): .047503
> 64
```

. . . Portfolio optimization - Minimum Variance Portfolio

```
.  capture getsymbols ALFAA.MX BIMBOA.MX ALSEA.MX AMXL.MX ARA.MX OMAB.MX, ///
>                  fy(2014) freq(m) yahoo clear price(adjclose)
. * Finding the MV Portfolio without short sales:
.  mvport r_*, ret(0.02) noshort
Portfolio weights of the portfolio:
             |    Weights
-------------+-----------
   r_ALFAA_MX |          0
  r_BIMBOA_MX |          0
   r_ALSEA_MX |   .1320823
    r_AMXL_MX |   .1325749
     r_ARA_MX |          0
    r_OMAB_MX |   .7353427
Number of observations used to calculate expected returns and var-covariance ma
> trix : 56
Required return of the Portfolio:  .02
Minimum standard deviation of the portfolio (DO NOT Allow Short Sales): .054502
> 24
```

. . . Portfolio backtesting

```
. * Backtesting using optimal no-short portfolio:
.  qui ovport r_* if period<tm(2017m1), noshort
.  matrix W=r(weights)
.  qui cbacktest p_* if period>=tm(2017m1), weights(W) gen(holing_return) timev
> ar(period)
```



Portfolio strategy based on CAPM

- Get stock data from 500 stocks (S&P500) using getsymbols in a loop
- Calculate CAPM of the 500 stocks using the capm command
- Selecting those stocks with alpha>0 and p-value<0.05
- Optimize the portfolio without the recent 12 months
- Backtest the portfolio in the recent 12 months

Forecasting with ARIMA/SARIMA

```
.   * Getting data of monthly sales of different manufacturing firms:
.   use http://www.apradie.com/ec2004/salesfabs.dta, clear
.   qui tsset month
.   * Generating the natural log of volume sales:
.   qui gen lnqfab1=ln(qfab1)
.   qui label var qfab1 "Sales Fab1"
```

. . . Forecasting with ARIMA/SARIMA

```
.   ac s12.lnqfab1
```

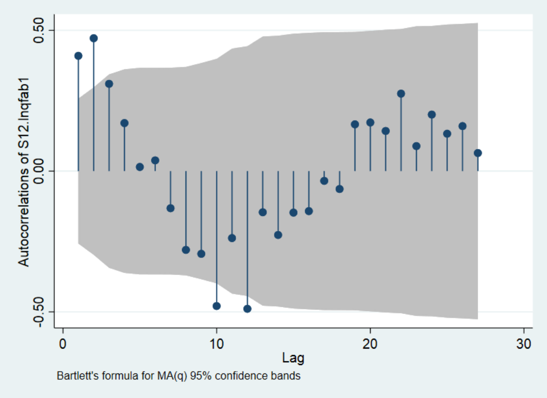


Figure 11: Autocorrelations of annual % sales growth

. . . Forecasting with ARIMA/SARIMA

```
.    pac s12.lnqfab1
```

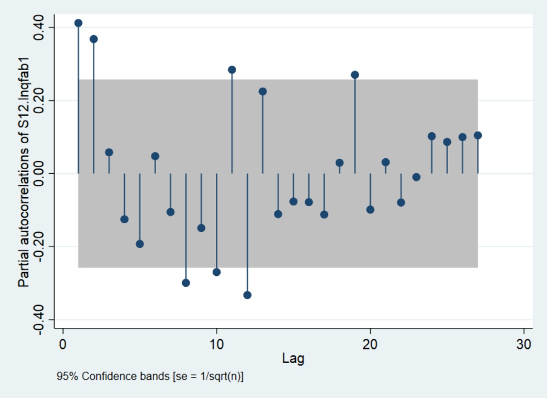


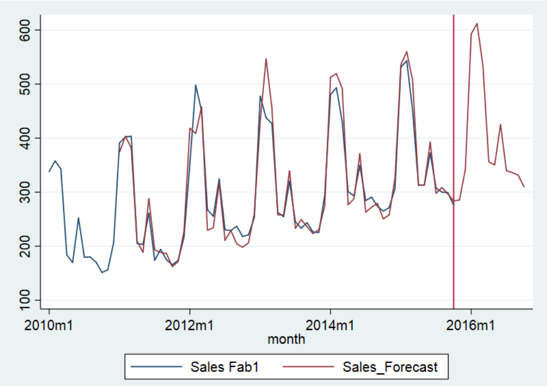
Figure 12: Partial Autocorrelations of annual % sales growth

... Forecasting with ARIMA/SARIMA

```
.   * Autocorrelation plots suggest the following model:
.   arima s12.lnqfab1, ar(1) mma(1,12) nolog

ARIMA regression

Sample:  2011m1 - 2015m10                       Number of obs     =         58
                                                Wald chi2(2)      =      23.05
Log likelihood =  66.13805                      Prob > chi2       =     0.0000

------------------------------------------------------------------------------
             |                 OPG
 S12.lnqfab1 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
lnqfab1      |
       _cons |   .1014176   .0110564     9.17   0.000     .0797475    .1230877
-------------+----------------------------------------------------------------
ARMA         |
          ar |
         L1. |   .4211392   .1073157     3.92   0.000     .2108043    .6314742
-------------+----------------------------------------------------------------
ARMA12       |
          ma |
         L1. |  -.4961238   .1464453    -3.39   0.001    -.7831514   -.2090962
-------------+----------------------------------------------------------------
      /sigma |   .0750096     .00783     9.58   0.000     .0596631    .0903562
------------------------------------------------------------------------------
Note: The test of the variance against zero is one sided, and the two-sided
      confidence interval is truncated at zero.
```

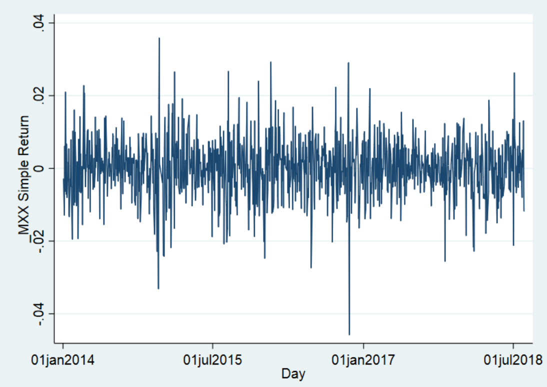
. . . Forecasting with ARIMA/SARIMA

```
.   qui tsappend, add(12)
.   qui predict lnqfab1_hat, y dynamic(tm(2015m11))
.   qui gen Sales_Forecast=exp(lnqfab1_hat)
.   local lastmonth=tm(2015m10)
.   tsline qfab1 Sales_Forecast, tline(`lastmonth´)
```



Modellilng the Mexican Market volatility

```
.    capture getsymbols ^MXX, fm(1) fd(1) fy(2014) freq(d) ///
>           price(adjclose) clear yahoo
.    tsline R__MXX
```



Running a GARCH(1,1) model

```
.   arch r__MXX, arch(1) garch(1) nolog
ARCH family regression
Sample: 1 - 1152                                Number of obs    =      1,152
Distribution: Gaussian                          Wald chi2(.)     =          .
Log likelihood =   3980.72                      Prob > chi2      =          .
```

r__MXX	Coef.	OPG Std. Err.	z	P>|z|	[95% Conf.	Interval]
r__MXX						
_cons	.0002149	.0002148	1.00	0.317	-.0002061	.0006358
ARCH						
arch L1.	.085835	.0130238	6.59	0.000	.0603089	.1113611
garch L1.	.8649778	.024098	35.89	0.000	.8177466	.9122089
_cons	3.14e-06	9.52e-07	3.30	0.001	1.28e-06	5.01e-06

We find evidence of volatility clustering (garch coef $= 0.865$)

Understanding volatility of the Market index

```
.   qui getsymbols ^MXX, fm(1) fd(1) fy(2008) lm(12) ///
>   ld(31) ly(2008) frequency(d) price(adjclose) yahoo clear
Symbol ^MXX was downloaded
.   qui drop if volume==.
.   qui gen t=_n
.   qui tsset t
.   ** I calculate moving volatility and moving average using a rolling window
> of 20 business days:
.   rolling mean=r(mean) volatility=r(sd), saving(rollingmxx, replace) window(2
> 0): su r__MXX
(running summarize on estimation sample)
Rolling replications (233)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50
..................................................   100
..................................................   150
..................................................   200
.................................
file rollingmxx.dta saved
.   qui save datamxx, replace
```

Understanding volatility of the Market index

```
.    qui use rollingmxx, clear
.    * I rename the time variable
.    ren start t
.    qui tsset t
.    label var volatility "volatility"
.    label var mean "mean returns"
.    * I merge the rolling dataset with the original
.    qui merge 1:1 t using datamxx, keepusing(period)
.    qui drop _merge
.    qui tsset period
```

. . . Volatility Models

```
.   tsline mean volatility
```

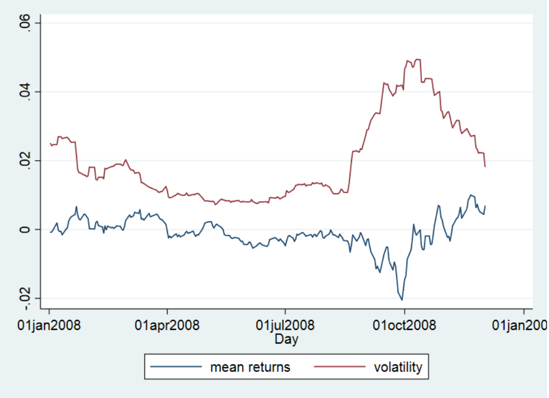


Figure 15: Market Volatility vs Return using 24-month rolling windows

Conclusion

- Unlike other leading econometrics software (e.g. R), Stata has a simple script language (do and ado files) that students can easily learn to better understand Econometrics
- New user-commands are being developed for Finance students (e.g. Dr. Dicle commands, see this ssrn paper)
- The disadvantage of Stata vs other software like R or Python is that there are much more commands/packages for finance analysis in R and Python compared with Stata
- Programming in Stata is much easier for non-programmers compared with R

Thanks! Questions?

Carlos Alberto Dorantes, Ph.D.

Profesor de tiempo completo

Departamento de Contabilidad y Finanzas

Tecnológico de Monterrey, Campus Querétaro

cdorante@itesm.mx