CAPM-based optimal portfolios

Carlos Alberto Dorantes, Tec de Monterrey

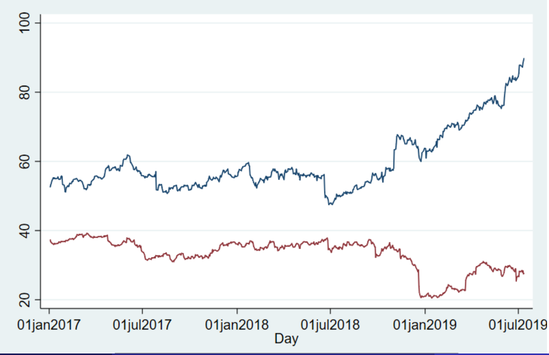
2019 Chicago Stata Conference

Outline

Data collection of financial data

getsymbols easily downloads and process data from Quandl, Yahoo Finance, and Alpha Vantage. Here an example of daily prices of two stocks:

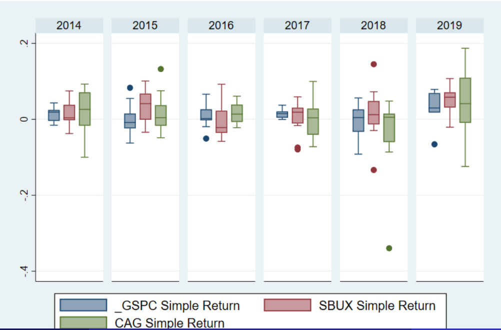
```
. * ssc install getsymbols
. cap getsymbols SBUX CAG, fy(2017) yahoo clear
. tsline adjclose_SBUX adjclose_CAG
```



. . . Data collection of financial data

Getting monthly prices and calculating returns from several instruments from Yahoo:

```
. cap getsymbols ^GSPC SBUX CAG, fy(2014) freq(m) yahoo clear price(adjclose)
. *With the price option, returns are calculated
. cap gen year=yofd(dofm(period))
. graph box R_*, by(year, rows(1))
```



The CAPM model and Portfolio Theory

Overview of Portfolio Theory

The paper "Portfolio Selection" written in 1952 by Harry Markowitz was the beginning of Modern Portfolio Theory (MPT)

Nobody before Markowitz had provided a rigorous framework to construct and select assets in a portfolio based on expected asset returns and the systematic and unsystematic risk of a portfolio.

With the mean-variance analysis, Markowitz proposes a way to measure portfolio risk based on the variance-covariance matrix of asset returns

He discovered that the relationship between risk and return is not always linear; he proposes a way to estimate the efficient frontier, and found that it is quadratic. It is possible to build portfolios that maximize expected return and also minimize risk

The CAPM model

Under the theoretical framework of MPT, in the late 1950's and early 1960's, the Capital Asset Pricing Theory was developed. The main contributors were James Tobin, John Litner and William Sharpe.

They showed that the efficient frontier is transformed from quadratic to a linear function (the Capital Market Line) when a risk-free rate is added to a portfolio of stocks

The Tangency (optimal) Portfolio is the portfolio that touches both, the CML and the efficient frontier

. . . The CAPM model

The two-fund separation theorem: any investor can optimize his/her portfolio by investing in 2 instruments: the tangent or market portfolio and the risk-free rate. The weights for each instrument depends on the investor's risk aversion.

The expected return of this 2-fund portfolio is a weighted average. From this basic idea, the CAPM was developed.

CAPM states that the expected return of a stock is given by the risk-free rate plus its market beta coefficient times the premium market return:

$$E[R_i] = R_f + \beta \left(R_M - R_f \right)$$

Relationship between CAPM and Portfolio Theory

CAPM model can be used to

1. estimate the expected return of a stock given an expected return of the market. This estimate can be used as the expected stock return, that is part if the inputs for MPT.
2. estimate the cost of equity or discount factor in the context of financial valuation
3. select stocks for a portfolio based on the Jensen's Alpha and/or market beta coefficients

Automating stock selection and portfolio optimization

CAPM estimation model

Writing a Stata command for the CAPM

Excel interface to easily change the input parameters

Stock selection based on the CAPM

Portfolio optimization

Portfolio backtesting

CAPM estimation model

To estimate the CAPM, I can run a time-series linear regression model using monthly continuously compounded returns. For this model, the dependent variable is the premium stock return (excess stock return over the risk-free rate) and the independent variable is the premium market return:

$$\left(r_{i(t)} - r_{f(t)}\right) = \alpha + \beta\left(r_{M(t)} - r_{f(t)}\right) + \varepsilon_t$$

Note that I allow the model to estimate the constant (alpha of Jensen). In theory, for a market to be in equilibrium, this constant must be zero, since there should not be a stock that systematically outperforms the market; in other words, a stock return should be determined by its market systematic risk (beta) and its unsystematic/idiosyncratic risk (regression error)

Writing a command for the CAPM model

```
. capture program drop capm

. program define capm, rclass
syntax varlist(min=2 max=2 numeric) [if], RFrate(varname)
local stockret: word 1 of `varlist´
local mktret: word 2 of `varlist´
cap drop prem`stock´
qui gen prem`stock´=`stockret´-`rfrate´
cap drop mktpremium
qui gen mktpremium=`mktret´-`rfrate´
cap reg prem`stock´ mktpremium `if´
if _rc==0 & r(N)>30  {
  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 "Alpha is " %8.6f `b0´ "; std. error of alpha 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

Code to collect risk-free data

```
.    *I get the risk-free reta from the FED:
.    qui freduse TB3MS, clear
.    * I create monthly cc rate from the annual % rate:
.    qui gen m_Rf = (TB3MS/100)/12
.    qui gen m_rf = ln(1 + m_Rf)
.    ** I create and format the monthly variable:
.    qui gen period =mofd(daten)
.    format period %tm
.    qui tsset period
.    * I save the CETES dataset:
.    qui save rfrate, replace
```

. . . Writing a command for CAPM

Now I can use my capm command using monthly data of a stock:

```
. cap getsymbols ^GSPC CAG, fy(2014) freq(m) yahoo clear price(adjclose)
. * I merge the stock data with the risk-free dataset:
. qui merge 1:1 period using rfrate, keepusing(m_rf)
. qui drop if _merge!=3
. qui drop _merge
. qui save mydata1,replace
.
.   capm r_CAG r__GSPC, rfrate(m_rf)
Market beta is 0.87; std. error of beta is 0.259952
Alpha is -0.003563; std. error of alpha is 0.008909
.   return list
scalars:
                  r(N) =  65
               r(SEb0) =  .0089092926405626
               r(SEb1) =  .259951861344863
                 r(b0) =  -.0035630903188013
                 r(b1) =  .8731915389793837
```

. . . Writing a command for CAPM

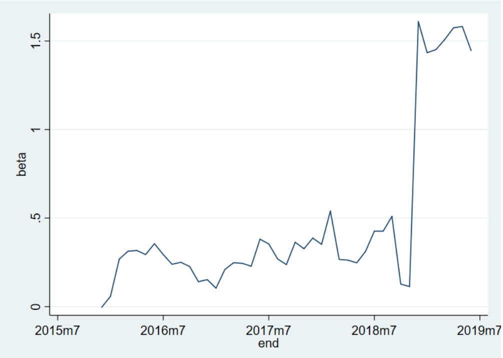
I can examine how market beta of a stock changes over time I run my capm command using 24-month rolling windows:

```
.   rolling b1=r(b1) seb1=r(SEb1), window(24) saving(capmbetas,replace): ///
>                capm r_CAG r__GSPC, rfrate(m_rf)
(running capm on estimation sample)
Rolling replications (43)
────┼─── 1 ───┼─── 2 ───┼─── 3 ───┼─── 4 ───┼─── 5
...........................................
file capmbetas.dta saved
.
```

. . . Writing a command for CAPM

Code to show how beta moves over time

```
.   qui use capmbetas,clear
.   label var b1 "beta"
.   qui tsset end
.   tsline b1
```



CAPM-GARCH estimation model using daily data

```
. capture program drop capmgarch

. program define capmgarch, rclass
syntax varlist(min=2 max=2 numeric) [if], RFrate(varname) timev(varname)
local stockret: word 1 of `varlist´
local mktret: word 2 of `varlist´
tempvar stockpremium mktpremium
tsset `timev´
qui gen `stockpremium´=`stockret´-`rfrate´
qui gen `mktpremium´=`mktret´-`rfrate´
cap arch `stockpremium´ `mktpremium´ `if´, arch(1) garch(1) ar(1)
if _rc==0 & r(N)>30  {
  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 "Alpha is " %8.6f `b0´ "; std. error of alpha 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
```

Excel interface for automation of data collection

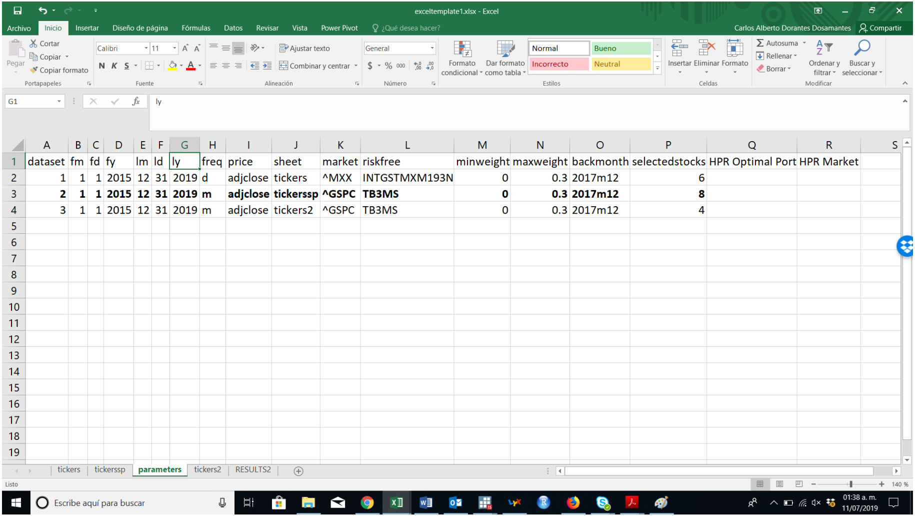


Figure 4: exceltemplate1, parameters Sheet

Excel interface for automation of data collection

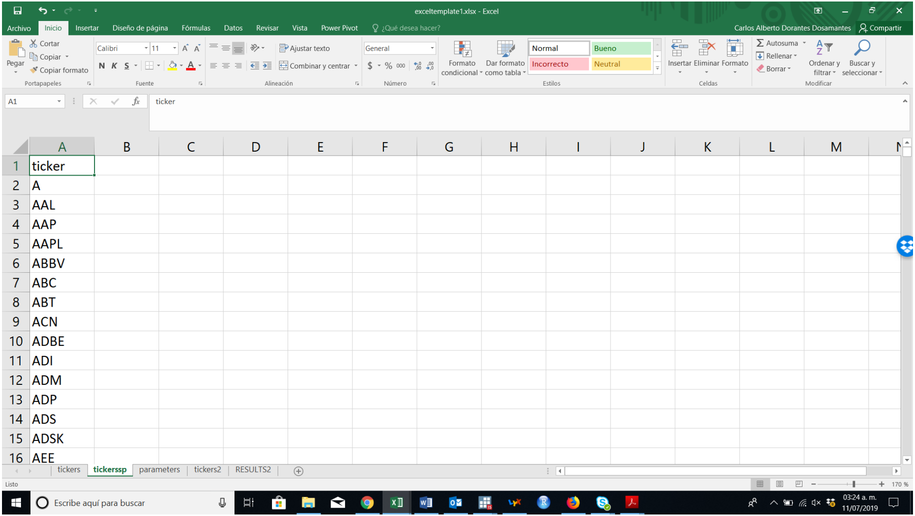


Figure 5: exceltemplate1, tickerssp Sheet

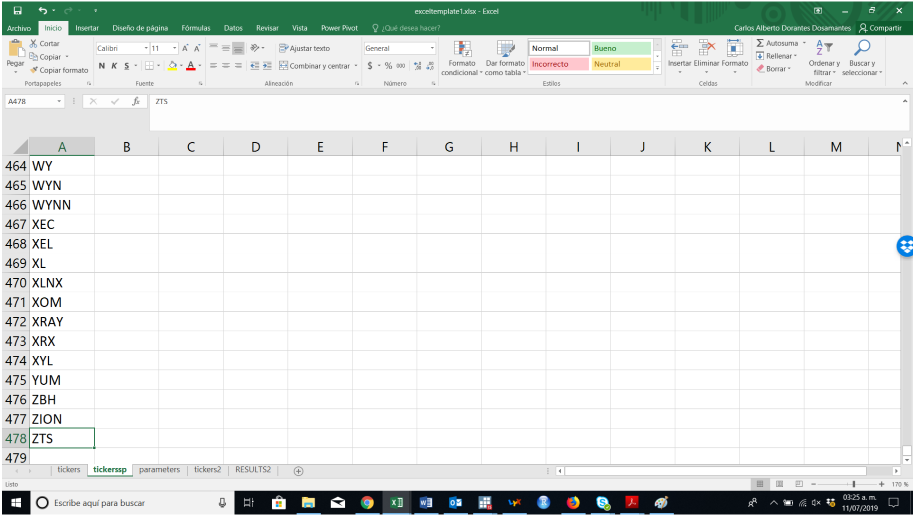


Figure 6: exceltemplate1, tickerssp Sheet

Excel interface for automation of data collection

I will use all S&P500 tickers with valid monthly price data(dataset=2)

I will use monthly historical data from Jan 2015 to Dec 2017 to estimate the CAPM models, select stocks and optimize the portfolio

I will use the period from Jan 2018 to Jun 2019 as the backtesting period for the CAPM-based investment strategy

Excel interface for automation of data collection

Code to read inputs from Excel template:

```
. clear
. import excel "exceltemplate1.xlsx", sheet("parameters") firstrow
. *I define a macro for the dataset number to be read from the Sheet:
. local dataset=2
. * I read the input parameters in global macros:
. global fm=fm[`dataset´]
. global fd=fd[`dataset´]
. global fy=fy[`dataset´]
. global lm=lm[`dataset´]
. global ld=ld[`dataset´]
. global ly=ly[`dataset´]
. global frequency=freq[`dataset´]
. global price=price[`dataset´]
. global tickersheet=sheet[`dataset´]
. global minw=minweight[`dataset´]
. global maxw=maxweight[`dataset´]
. global mktindex=market[`dataset´]
. global rfratename=riskfree[`dataset´]
. global backmonth=backmonth[`dataset´]
. global selectedstocks=selectedstocks[`dataset´]
```

Code to collect and save price and return data

```
. * Now I open the sheet where the tickers are saved:
. import excel "exceltemplate1.xlsx", sheet("$tickersheet") firstrow clear
. * I create a macro with the list of tickers from the variable:
. cap levelsof ticker, local(ltickers) clean
. global listatickers="`ltickers´"
. * I bring the price and return data of all tickers from Yahoo:
. cap getsymbols $mktindex $listatickers, ///
>  fm($fm) fd($fd) fy($fy) lm($lm) ld($ld) ly($ly) ///
>  freq($frequency) price($price) yahoo clear
. * The getsymbols command leaves the tickers that were found on Yahoo Finance
. global numtickers=r(numtickers)
. global listafinal=r(tickerlist)
. * I will create a ticker list without the market index
. global listafinal1=""
. foreach ticker of global listafinal {
  2.   if "`ticker´"!="$mktindex" {
  3. global listafinal1="$listafinal1 `ticker´"
  4.   }
  5. }
. save stockdataset, replace
file stockdataset.dta saved
```

Excel interface for automation of data collection

Code to collect and save the risk-free data

```
. clear
. *ssc install freduse
. freduse $rfratename
(1,026 observations read)
.   * I create monthly cc rate from the annual % rate:
. gen m_Rf = ($rfratename/100)/12
. * I calculate the continuously compounded return from the simple returns:
. gen m_rf = ln(1 + m_Rf)
. * I create monthly variable for the months:
. gen period =mofd(daten)
. format period %tm
. * Now I indicate Stata that the time variable is period:
. tsset period
        time variable:  period, 1934m1 to 2019m6
                delta:  1 month
. * I save the CETES dataset as cetes:
. save riskfdata, replace
file riskfdata.dta saved
```

Excel interface for automation of data collection

Code to merge the stock dataset with the risk-free dataset

```
. *Now I open the stock data and do the merge:
. use stockdataset, clear
(Source: Yahoo Finance!)

. merge 1:1 period using riskfdata, keepusing(m_rf)

    Result                           # of obs.
    -----------------------------------------
    not matched                           973
        from master                         1  (_merge==1)
        from using                        972  (_merge==2)

    matched                                54  (_merge==3)
    -----------------------------------------

. * I keep only those raws that matched (_merge==3)
. keep if _merge==3
(973 observations deleted)

. drop _merge

. * I save the the dataset with the risk-free data:
. save stockdataset, replace
file stockdataset.dta saved
```

Automating the estimation of all CAPM models

Code to create a matrix for the CAPM coefficients and std. errors

```
. * I rename the market return variable to avoid calculating
. *   a CAPM for the market variable r__MXX:
. local varmkt=strtoname("$mktindex",0)
. local varmkt="r_`varmkt´"
. rename `varmkt´ rMKT
. * I define a matrix to store the beta coefficients and the p-values:
. * The macro $numtickers has the number of valid tickers found
. set matsize 600
. matrix BETAS=J($numtickers-1,5,0)
. matrix colnames BETAS= alpha beta se_alpha se_beta N
```

Automating the estimation of all CAPM models

Code to estimate CAPM models and store coefficients in a matrix

```
. * I do a loop to run al CAPM regressions:
. local j=0
. * I define a global macro for the list of all returns that
. *  I will be using for the names of the rows for the Matrix
. global listaret=""
. foreach var of varlist r_* {
 2.   local j=`j´+1
 3.   cap capm `var´ rMKT if period<=tm($backmonth), rfrate(m_rf)
 4.   matrix BETAS[`j´,1]=r(b0)
 5.   matrix BETAS[`j´,2]=r(b1)
 6.   matrix BETAS[`j´,3]=r(SEb0)
 7.   matrix BETAS[`j´,4]=r(SEb1)
 8.   matrix BETAS[`j´,5]=r(N)
 9.   global listaret="$listaret `var´"
10. }
```

Automating the estimation of all CAPM models

Code to show results stored in the matrix

```
. * I assign names to each row according to the ticker list:
. matrix rownames BETAS=$listaret

. matlist BETAS[1..8,.]
```

	alpha	beta	se_alpha	se_beta	N
r_A	.0045485	1.541188	.0073556	.2520855	35
r_AAL	-.0060653	1.0219	.0152584	.5229235	35
r_AAP	-.0171335	.4446284	.0154508	.5295168	35
r_AAPL	.0006362	1.384231	.0094234	.3229497	35
r_ABBV	.0058071	1.291651	.0095205	.3262765	35
r_ABC	-.0085388	1.063213	.0123592	.4235645	35
r_ABT	-.0050732	1.702326	.0073894	.2532439	35
r_ACN	.0105192	1.039847	.0064168	.2199101	35

Automating the estimation of all CAPM models

Code to send results to excel

```
. * I set the Sheet where results will be sent :
. putexcel set exceltemplate1.xlsx, sheet("RESULTS`dataset´") modify
. * I save the complete matrix in cell B1
. capture putexcel B1=matrix(BETAS), names
. putexcel B1=matrix(BETAS,names)
file exceltemplate1.xlsx saved
. * I save the list of tickers in column A
. putexcel A1=("ticker")
file exceltemplate1.xlsx saved
. local j=1
. foreach ticker of global listafinal1 {
  2.     local j=`j´+1
  3.     quietly putexcel A`j´=("`ticker´")
  4. }
```

Stock selection based on CAPM coefficients

Code to create 95% C.I. of coefficients and select stocks

```
. * Importing the resulting sheet with the beta coefficients in to Stata:
. import excel using exceltemplate1, sheet("RESULTS`dataset´") firstrow clear
. * I generate the 95% confidence interval of alpha and beta:
. cap gen minalpha=alpha - abs(invttail(N,0.05)) * se_alpha
. cap gen maxalpha=alpha + abs(invttail(N,0.05)) * se_alpha
. cap gen minbeta=beta - abs(invttail(N,0.05)) * se_beta
. cap gen maxbeta=beta + abs(invttail(N,0.05)) * se_beta
. count if minalpha >=0
  31
. display "Number of stocks with SIGNIFICANT AND POSITIVE ALPHA=" r(N)
Number of stocks with SIGNIFICANT AND POSITIVE ALPHA=31

.
. keep if minalpha >=0 & minbeta>=0
(451 observations deleted)
. * Now I will sort the stocks based on Alpha:
. gsort -alpha
. * I will keep the best stocks in terms of alpha:
. capture keep in 1/$selectedstocks
. * I save the best stock tickers in a Stata dataset
. save besttickers`dataset´, replace
file besttickers2.dta saved
```

Portfolio optimization of the selected stocks

Code to bring price and return data of the selected stocks

```
. cap use besttickers`dataset´, clear
. cap levelsof ticker, local(ltickers) clean
. global besttickers="`ltickers´"
. * I bring the price and return data from Yahoo:
. cap getsymbols $besttickers, ///
>  fm($fm) fd($fd) fy($fy) lm($lm) ld($ld) ly($ly) ///
>  frequency($frequency) price($price) yahoo clear
. save beststocks`dataset´, replace
file beststocks2.dta saved
. * If delete stocks with few valid observations in the backtest period
. foreach ret of varlist r_* {
  2.    qui su `ret´ if period>tm($backmonth)
  3.    if r(N)<12 {
  4. drop `ret´
  5.    }
  6. }
```

Code to optimize the portfolio with restrictions (before backmonth)

```
. ovport r_* if period<=tm($backmonth), minw($minw) max($maxw)
Number of observations used to calculate expected returns and var-cov matrix :
> 36
The weight vector of the Tangent Portfolio with a risk-free rate of 0 (NOT Allo
> w Short Sales) is:
          Weights
r_ADBE  .01884751
r_ALGN  .07281493
r_AMZN  .07410937
r_CDNS  .05888357
r_NVDA  .16591405
 r_PGR         .3
r_TTWO  .00943057
 r_UNH         .3
The return of the Tangent Portfolio is: .03349151
The standard deviation (risk) of the Tangent Portfolio is: .03429288

The marginal contributions to risk of the assets in the Tangent Portfolio are:
             | Margina~k
-------------+-----------
      r_ADBE |  .0272682
      r_ALGN |  .0464094
      r_AMZN |  .0334565
      r_CDNS |  .0258435
      r_NVDA |  .0701327
```

Code for the holding period return of the portfolio after backmonth

```
. matrix wop=r(weights)

. backtest p_* if period>tm($backmonth), weights(wop)
It was assumed that the dataset is sorted chronologically
The holding return of the portfolio is .26143658
19 observations/periods were used for the calculation (casewise deletion was ap
> plied)
The holding return of each price variable for the specified period was:
```

Price variable	Return
p_adjclose_ADBE	.5398479
p_adjclose_ALGN	.0785878
p_adjclose_AMZN	.3792017
p_adjclose_CDNS	.6763262
p_adjclose_NVDA	-.3201532
p_adjclose_PGR	.6445434
p_adjclose_TTWO	-.0800505
p_adjclose_UNH	.1270745

```
The portfolio weights used were:
```

Asset	Weight
r_ADBE	.0188475
r_ALGN	.0728149

Estimating the return of the market portfolio

Code fo the holding period return of the market

```
. cap getsymbols $mktindex, fm($fm) fd($fd) fy($fy) lm($lm) ///
> ld($ld) ly($ly) freq($frequency) price($price) yahoo clear

. matrix w1=1

. backtest p_* if period>tm($backmonth), weights(w1)
It was assumed that the dataset is sorted chronologically
The holding return of the portfolio is .0623625
19 observations/periods were used for the calculation (casewise deletion was ap
> plied)
The holding return of each price variable for the specified period was:
```

Price variable	Return
p_adjclose__GSPC	.0623625

```
The portfolio weights used were:
```

Asset	Weight
r1	1

```
. scalar retmkt=r(retport)

. display "The HPR of the market was " retmkt
The HPR of the market was .0623625

. display "The HPR of the optimal portfolio was " retopt
```

Exporting results to the Excel template

Code to export results to the Excel template

```
. putexcel set exceltemplate1.xlsx, sheet("parameters") modify
. putexcel Q1=("HPR Optimal Port") R1=("HPR Market")
file exceltemplate1.xlsx saved
. local row=`dataset´+1
. putexcel Q`row´=(retopt)
file exceltemplate1.xlsx saved
. putexcel R`row´=(retmkt)
file exceltemplate1.xlsx saved
```

Results of the automatic stock selection

- Portfolio return of selected stocks was much higher than the benchmark (market). For the holding period of 18 months, the optimal portfolio had a return around 26%, while the S&P 500 had a holding return of around 6% (from Jan 2018 to Jun 2019)
- Better results were obtained using bootstrapping for CAPM estimation (not shown here)
- Better results were obtained using Exponential Weighted Moving Average method for the estimation of expected stock return and expected variance-covariance matrix (results not shown here)

Conclusions

- Data collection and data management can be enhanced with interfaces between Stata and Excel
- The getsymbols command along with commands from the mvport package are useful for financial data management and for constructing optimal portfolios
- Unlike other leading econometrics software, Stata has a simple script language (do and ado files) that students can easily learn to better understand Financial Econometrics

Thanks! Questions?

Carlos Alberto Dorantes, Ph.D.

Full-time professor

Accounting and Finance Department

Tecnológico de Monterrey, Querétaro Campus

cdorante@tec.mx