```
* Advanced Derivatives Assignment 1  (Jack, Zaid, Nick, Asim, Dikshant)

* Reproduction of Hull and White Results

clear all
set more off

*log using Version1, replace

cd "C:\Users\JackE\Documents\Jack\Jack's Work\UoT MFE\Semester 2\Advanced 
Derivatives\Assignment 1"
use "C:\Users\JackE\Documents\Jack\Jack's Work\UoT MFE\Semester 2\Advanced 
Derivatives\Assignment 1\Assignment1_V2.dta"

*summary stats
summarize spx_level strike_price impl_volatility

*generate variables
gen yearmonth = ym(year, month)
gen yearmonth1 = yearmonth-527

gen sqrt_T = sqrt(lifeopt/360)
gen change_S = (n_spx - spx_level)
gen change_F = (n_midmarket - midmarket)
gen delta_sq = delta^2

gen m1 = (vega/sqrt_T)
gen m2 =(change_S/spx_level)
gen multiplier = sqrt(m1*m2)

gen x1 = multiplier
gen x2 = multiplier * delta
gen x3 = multiplier * delta_sq
gen y = (change_F - (delta*change_S))

tostring date1, replace format(%20.0f)
gen date_stata = date(date1, "YMD")

gen round_delta = round(delta, 0.1)

* CALLS

gen call_x1 = .
gen call_x2 = .
gen call_x3 = .
gen yhat_call = .
gen window1 = .

*have 105 months

local j = 1
forvalues i = 1(1)105{
quietly reg y x1 x2 x3 if iscall ==1 & inrange(yearmonth1, 0+`i',35+`i'), noconstant
quietly replace call_x1 = _b[x1] if call_x1 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall
 ==1
quietly replace call_x2 = _b[x2] if call_x2 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall
 ==1
quietly replace call_x3 = _b[x3] if call_x3 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall
 ==1
quietly predict yhat
quietly replace yhat_call = yhat if yhat_call ==. & inrange(yearmonth1, 0+`i',35+`i')&
iscall ==1
quietly drop yhat
quietly replace window1 = `j' if window1 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall ==1
local j = `j'+1
}

preserve

*collpase to get a_hat, b_hat, c_hat for each regression
```

```
collapse (mean) call_x1 call_x2 call_x3 date_stata, by (window1)
gen call_minusx2 = (-1)*call_x2
format date_stata %td

gen a_call = call_x1
gen minusb_call = call_minusx2
gen c_call = call_x3

*graph
line a_call minusb_call c_call date_stata, xtitle("Date") title("Call Parameters")
graph save Call.gph, replace

restore

* PUTS

gen put_x1 = .
gen put_x2 = .
gen put_x3 = .
gen yhat_put = .
gen window2 = .

local j = 1
forvalues i = 1(1)105{
quietly reg y x1 x2 x3 if iscall ==0 & inrange(yearmonth1, 0+`i',35+`i'), noconstant
quietly replace put_x1 = _b[x1] if put_x1 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall
==0
quietly replace put_x2 = _b[x2] if put_x2 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall
==0
quietly replace put_x3 = _b[x3] if put_x3 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall
==0
quietly predict yhat
quietly replace yhat_put = yhat if yhat_put ==. & inrange(yearmonth1, 0+`i',35+`i')& iscall
==0
quietly drop yhat
quietly replace window2 = `j' if window2 ==. & inrange(yearmonth1, 0+`i',35+`i') & iscall ==0
local j = `j'+1
}

preserve

*collapse to find parameters
collapse (mean) put_x1 put_x2 put_x3 date_stata, by (window2)
format date_stata %td

gen a_put = put_x1
gen b_put = put_x2
gen c_put = put_x3

*graph
line a_put b_put c_put date_stata, xtitle("Date") title("Put Parameters")
graph save Put.gph, replace

restore

gr combine Call.gph Put.gph
graph save Combined.gph, replace

**End PART 1

*ERRORS
gen mv_error_call = (y - yhat_call)^2 if iscall ==1
gen bs_error_call = y^2 if iscall ==1

gen mv_error_put = (y - yhat_put)^2 if iscall ==0
gen bs_error_put = y^2 if iscall ==0

*calls

by round_delta, sort: egen george = total( mv_error_call) if iscall==1
by round_delta, sort: egen george2 = total( bs_error_call) if iscall==1
egen george3 = total(mv_error_call) if iscall ==1
egen george4 = total(bs_error_call) if iscall==1

*puts
by round_delta, sort: egen frank = total( mv_error_put) if iscall ==0
by round_delta, sort: egen frank2 = total( bs_error_put) if iscall ==0
egen frank3 = total(mv_error_put) if iscall ==0
egen frank4 = total(bs_error_put) if iscall ==0

*calculate gains
gen gain_call = (1-(george/george2))*100 if iscall ==1
gen gain_put = (1-(frank/frank2))*100 if iscall ==0

gen call_all = (1-(george3/george4))*100 if iscall ==1
gen put_all = (1-(frank3/frank4))*100 if iscall ==0
egen call_tot = mean(call_all) if iscall ==1
egen put_tot = mean(put_all) if iscall ==0


*put calculated GAINS in matrix
local r = 100
local column Call_Delta Gain(%) Put_Delta Gain(%)
matrix A = J(10,4,.)
matrix colnames A=`column'
matrix A[1,1] = 0.1
matrix A[2,1] = 0.2
matrix A[3,1] = 0.3
matrix A[4,1] = 0.4
matrix A[5,1] = 0.5
matrix A[6,1] = 0.6
matrix A[7,1] = 0.7
matrix A[8,1] = 0.8
matrix A[9,1] = 0.9
matrix A[10,1] = `r'

matrix A[1,3] = -0.9
matrix A[2,3] = -0.8
matrix A[3,3] = -0.7
matrix A[4,3] = -0.6
matrix A[5,3] = -0.5
matrix A[6,3] = -0.4
matrix A[7,3] = -0.3
matrix A[8,3] = -0.2
matrix A[9,3] = -0.1
matrix A[10,3] = `r'

preserve
collapse (mean) call_tot
mkmat call_tot, matrix(yay)
matrix A[10,2] = yay[1,1]
restore
preserve
collapse (mean) put_tot
mkmat put_tot, matrix(yay1)
matrix A[10,4] = yay1[1,1]
restore

preserve
bysort round_delta: keep if _n==1
mkmat gain_call, matrix(call)
mkmat gain_put, matrix(put)

local i = 2
matrix A[1,`i'] = call[10,1]
matrix A[2,`i'] = call[11,1]
matrix A[3,`i'] = call[12,1]
matrix A[4,`i'] = call[13,1]
matrix A[5,`i'] = call[14,1]
matrix A[6,`i'] = call[15,1]
matrix A[7,`i'] = call[16,1]
matrix A[8,`i'] = call[17,1]
matrix A[9,`i'] = call[18,1]



local i = 4
matrix A[1,`i'] = put[1,1]
matrix A[2,`i'] = put[2,1]
matrix A[3,`i'] = put[3,1]
matrix A[4,`i'] = put[4,1]
matrix A[5,`i'] = put[5,1]
matrix A[6,`i'] = put[6,1]
matrix A[7,`i'] = put[7,1]
matrix A[8,`i'] = put[8,1]
matrix A[9,`i'] = put[9,1]

matlist A

restore

*log close
*translate Version1.smcl Version1.pdf
```