```
/*

/// Do-file written by Carlos Goes (andregoes@gmail.com)
/// for use at Dr Prakash Loungani's Macroeconometrics course
/// at Johns Hopkins SAIS

*** This do file aims at
**** (a) practicing time-series commands in STATA
**** (b) calculating potential Output and Output Gap with the Hodrick-Prescott filter
**** (c) calculating a very simplified Taylor Rule for a Central Bank
**** (d) verify whether or not that central bank has been doveish/hawkish in respect to its own rule

ATTN:
    Make sure you have the hprescott command installed. If you dont:
     -> net search hprescott

    Make sure you have the outreg2 command installed. If you dont:
     -> net search outreg2

 */


// 1. Organize your workspace

capture log close                                                                   // closes any open logs
clear                                                                               // clears the memory
set more off                                                                        // makes sure STATA won't ask you to click "more" to continue running the code
use "https://dl.dropboxusercontent.com/u/17824416/Macroeconometrics/Stata/braziltaylor.dta"     // chooses the dataset
log using braziltaylor.log, replace                                                 // chooses logfile

gen date = ym(year,month)                                                           // generates a date variable from "year" and "month" variables; ym stands for
format date %tm                                                                     // formats data variable

** Label your variables

label var date "Date, in months"
label var y "Actual Output"
label var i "Brazil overnight interbank rate"
label var u "Unemployment, last 30 days (SP Metro Area)"
label var e "BRL Real Effective Exchange Rate"
label var ffr "US Federal Funds Rate"
label var deltapi "Annualized Inflation Rate"

tsset date, m                                                                       // sets timeseries mode on, monthly data
```

```
// 2. Calculate the output gap with monthly data (lambda = 129600)

hprescott y, stub(yhp) smooth(129600)                                   // runs HP-filter with lambda = 129000

** 2.1 Organize your output

rename yhp_y_sm_1 yhp
label var yhp "Potential Output"

gen ly = ln(y)
label var ly "Natural Log of actual output"

gen lyhp = ln(yhp)
label var lyhp "Long term trend"

gen lydiff = (ly - lyhp) / lyhp *100
label var lydiff "Output gap"

drop yhp*

** 2.2 Plot potential output and output gap

line lyhp date || line ly date, scheme(s2color) ///
  title("Potential Output and Actual Output in Brazil", position(11) margin(vsmall)) ///
  subtitle("End of Month, 2003-2012",  position(11) margin(vsmall) size(small)) ///
  caption("Source: Author's calculations, with IMF Data; Trend calculated using Hodrick-Prescott Filter", size(vsmall)) ///
  ytitle("Natural log of output", box fcolor(white)) xtitle(,box fcolor(white)) ///
  saving(output_and_potential, replace) name(output_and_potential, replace)

sort year                                                               // sorts data by year (necessary for following command)
by year: egen lydiff_ave = mean(lydiff)                                 // creates yearly averages of the Output Gap
label var lydiff_ave "Output gap annual average"

line lydiff lydiff_ave date, yline(0) title("Output Gap in Brazil", position(11) margin(vsmall)) ///
  subtitle("End of Year, 2003-2012",  position(11) margin(vsmall) size(small)) ///
  caption("Source: Author's calculations, with IMF Data; Trend calculated using Hodrick-Prescott Filter", size(vsmall)) ///
  ytitle("Deviation from trend") ttick(1998m1(24)2012m12)   ///
  saving(ygap, replace) nodraw

// 3. Add the missing data that you didn't have in the dataset

** Inflation target

qui {
gen pistar = 4.5
replace pistar = 5.5 if tin(2004m1,2004m12)
replace pistar = 4 if tin(2003m1,2003m12)
replace pistar = 3.5 if tin(2002m1,2002m12)
replace pistar = 4 if tin(2001m1,2001m12)
replace pistar = 6 if tin(2000m1,2000m12)
replace pistar = 8 if tin(1999m1,1999m12)
gen pidiff = deltapi - pistar
label var pistar "Inflation target"
label var pidiff "Inflation Differential"
}
```

```
// 4. Proceed with some pre-estimation tests

** Testing for autocorrelation

ac i, scheme(s2color) name(ac, replace) title("Autocorrelation", position(11) margin(vsmall)) ///
  subtitle("of interest rates in Brazil", position(11) margin(vsmall)) lwidth(thin) msymbol(oh) nodraw
pac i, scheme(s2color) name(pac, replace) title("Partial autocorrelation", position(11) margin(vsmall)) ///
  subtitle("of interest rates in Brazil", position(11) margin(vsmall)) lwidth(thin) msymbol(oh) ///
  ylabel(-.2(.4)1) nodraw
graph combine ac pac, rows(2) scheme(s2color) name(actable, replace)

// 5. Run different specifications for the Taylor Rule

** 5.1 Historical specification
 ** uses lagged interest rates, output gap and inflation differential

reg i l.i lydiff pidiff, robust                                                    // runs OLS regression
outreg2 l.i lydiff pidiff using table_brazil2.xls, replace cttop("Historical") cttop("OLS")     // stores the result in a XLS file

prais i l.i lydiff pidiff, corc                                                    // runs AR(1) regression
outreg2 l.i lydiff pidiff using table_brazil2.xls, cttop("Historical") cttop("AR(1)")          // stores the result in a XLS file

predict ihat                                                                       // stores predicted value in variable "ihat"
gen idiff = i - ihat                                                               // calculates the deviation of the actual interest rate to the predicted interest rate
gen mov_idiff = ( l11.idiff + l10.idiff + l9.idiff + l8.idiff + l7.idiff +  ///
    l6.idiff + l5.idiff + l4.idiff + l3.idiff + l2.idiff + l1.idiff + idiff ) / 12          // creates a 12-month moving average of the difference

predict resid_1, residual                                                          // stores the residual for the AR(1) regression
twoway lfitci resid_1 ihat if ihat < 30 || ///
    scatter resid_1 ihat if ihat < 30, name(resid_1, replace) ///
    lwidth(thin) scheme(s2color) title("Historical", margin(vsmall)) legend(off) ///
    yline(0) msymbol(oh) nodraw                                                    // plots the residual scatterplot

** 5.2 Mandate specification
 ** uses lagged interest rates and inflation differential

reg i l.i pidiff, robust                                                           // runs OLS regression
outreg2 l.i pidiff using table_brazil2.xls, cttop("Mandate") cttop("OLS")          // stores the result in a XLS file

prais i l.i pidiff, corc                                                           // runs AR(1) regression
outreg2 l.i pidiff using table_brazil2.xls, cttop("Mandate") cttop("AR(1)")        // stores the result in a XLS file

predict ihat2                                                                      // stores predicted value in variable "ihat2"
gen idiff2 = i - ihat2                                                             // calculates the deviation of the actual interest rate to the predicted interest rate
gen mov_idiff2 = ( l11.idiff2 + l10.idiff2 + l9.idiff2 + l8.idiff2 + ///
    l7.idiff2 + l6.idiff2 +  l5.idiff2 + l4.idiff2 + ///
    l3.idiff2 + l2.idiff2 + l1.idiff2 + idiff2 ) / 12                              // creates a 12-month moving average of the difference

predict resid_2, residual                                                          // stores the residual for the AR(1) regression
twoway lfitci resid_2 ihat2 if ihat2 < 30 || ///
    scatter resid_2 ihat2 if ihat2 < 30, name(resid_2, replace) ///
    lwidth(thin) scheme(s2color) title("Mandate", margin(vsmall)) legend(off) ///
    yline(0) msymbol(oh) nodraw                                                    // plots the residual scatterplot
```

```
** 5.2 Mandate specification
 ** uses lagged interest rates and inflation differential

reg i l.i pidiff, robust                                                    // runs OLS regression
outreg2 l.i pidiff using table_brazil2.xls, cttop("Mandate") cttop("OLS")   // stores the result in a XLS file

prais i l.i pidiff, corc                                                    // runs AR(1) regression
outreg2 l.i pidiff using table_brazil2.xls, cttop("Mandate") cttop("AR(1)") // stores the result in a XLS file

predict ihat2                                                               // stores predicted value in variable "ihat2"
gen idiff2 = i - ihat2                                                      // calculates the deviation of the actual interest rate to the predicted interest rate
gen mov_idiff2 = ( l11.idiff2 + l10.idiff2 + l9.idiff2 + l8.idiff2 + ///
    l7.idiff2 + l6.idiff2 +  l5.idiff2 + l4.idiff2 + ///
    l3.idiff2 + l2.idiff2 + l1.idiff2 + idiff2 ) / 12                       // creates a 12-month moving average of the difference

predict resid_2, residual                                                   // stores the residual for the AR(1) regression
twoway lfitci resid_2 ihat2 if ihat2 < 30 || ///
    scatter resid_2 ihat2 if ihat2 < 30, name(resid_2, replace) ///
    lwidth(thin) scheme(s2color) title("Mandate", margin(vsmall)) legend(off) ///
    yline(0) msymbol(oh) nodraw                                             // plots the residual scatterplot

** 5.3 XR specification
 ** uses lagged interest rates, output gap and the real effective exchange rate

reg i l.i lydiff pidiff e, robust                                           // runs OLS regression
outreg2 l.i lydiff pidiff e using table_brazil2.xls, cttop("XR") cttop("OLS")   // stores the result in a XLS file

prais i l.i lydiff pidiff e, corc                                           // runs AR(1) regression
outreg2 l.i lydiff pidiff e using table_brazil2.xls, cttop("XR") cttop("AR(1)") // stores the result in a XLS file

predict ihat3                                                               // stores predicted value in variable "ihat3"
gen idiff3 = i - ihat3                                                      // calculates the deviation of the actual interest rate to the predicted interest rate
gen mov_idiff3 = ( l11.idiff3 + l10.idiff3  + l9.idiff3  + l8.idiff3 ///
    + l7.idiff3  + l6.idiff3  + l5.idiff3  + l4.idiff3  + l3.idiff3  + l2.idiff3 ///
    + l1.idiff3  + idiff3  ) / 12                                           // creates a 12-month moving average of the difference

predict resid_3, residual                                                   // stores the residual for the AR(1) regression
twoway lfitci resid_3 ihat3 if ihat3 < 30 || ///
    scatter resid_3 ihat3 if ihat3 < 30, name(resid_3, replace) ///
    lwidth(thin) scheme(s2color) title("Exchange rate", margin(vsmall)) legend(off) ///
    yline(0) msymbol(oh) nodraw                                             // plots the residual scatterplot
```

```
** 5.4 Copom specification
 ** uses lagged interest rates, output gap and dummies for meetings of the monetary committee

reg i l.i lydiff pidiff copom, robust                                          // runs OLS regression
outreg2 l.i lydiff pidiff copom using table_brazil2.xls, cttop("Copom") cttop("OLS")   // stores the result in a XLS file

prais i l.i lydiff pidiff copom, corc                                          // runs AR(1) regression
outreg2 l.i lydiff pidiff copom using table_brazil2.xls, cttop("Copom") cttop("AR(1)")  // stores the result in a XLS file

predict ihat4                                                                  // stores predicted value in variable "ihat4"
gen idiff4 = i - ihat4                                                         // calculates the deviation of the actual interest rate to the predicted interest rate
gen mov_idiff4 = ( l11.idiff4 + l10.idiff4  + l9.idiff4  + l8.idiff4  + ///
    l7.idiff4  + l6.idiff4  + l5.idiff4  + l4.idiff4  + l3.idiff4  + ///
    l2.idiff4  + l1.idiff4  + idiff4  ) / 12                                   // creates a 12-month moving average of the difference

predict resid_4, residual                                                      // stores the residual for the AR(1) regression
twoway lfitci resid_4 ihat4 if ihat4 < 30 || ///
    scatter resid_4 ihat4 if ihat4 < 30, name(resid_4, replace) ///
    lwidth(thin) scheme(s2mono) title("Copom", margin(vsmall)) legend(off) ///
    yline(0) msymbol(oh) nodraw                                                // plots the residual scatterplot

** 5.5 Interest Rate Parity Specification
 ** uses lagged interest rates, output gap and US federal funds rate

reg i l.i lydiff pidiff ffr, robust                                            // runs OLS regression
outreg2 l.i lydiff pidiff ffr using table_brazil2.xls, cttop("IRP") cttop("OLS")       // stores the result in a XLS file

prais i l.i lydiff pidiff ffr, corc                                            // runs AR(1) regression
outreg2 l.i lydiff pidiff ffr using table_brazil2.xls, cttop("IRP") cttop("AR(1)")     // stores the result in a XLS file

predict ihat5                                                                  // calculates the deviation of the actual interest rate to the predicted interest rate
gen idiff5 = i - ihat5
gen mov_idiff5 = ( l11.idiff5 + l10.idiff5  + l9.idiff5  + l8.idiff5  + ///
    l7.idiff5  + l6.idiff5  + l5.idiff5  + l4.idiff5  + l3.idiff5  + ///
    l2.idiff5  + l1.idiff5  + idiff5  ) / 12                                   // creates a 12-month moving average of the difference

predict resid_5, residual                                                      // stores the residual for the AR(1) regression
twoway lfitci resid_5 ihat5 if ihat5 < 30 || ///
    scatter resid_5 ihat5 if ihat5 < 30, name(resid_5, replace) ///
    lwidth(thin) scheme(s2color) title("IRP", margin(vsmall)) legend(off) ///
    yline(0) msymbol(oh) nodraw                                                // plots the residual scatterplot
```

```
// 6. Plot the result graphs

** 6.1 Taylor rules graph
    ** Will show the expected interest rate for all different specifications

line ihat ihat2 ihat3 ihat4 ihat5 date if tin(2002m1,2012m12), name(actual_taylor, replace) ///
  lwidth(thin thin thin thin thin thin) scheme(s2color) ///
  title("Taylor Rules", position(11) margin(vsmall)) ///
  subtitle("Annualized rates, per month (Jan 2002 - Dec 2012)",  position(11) margin(vsmall) size(small)) ///
  caption("Source: Author's calculations, with IPEA & BCB Data", size(vsmall)) ///
  legend(label(1 "Taylor rule, historical") label(2 "Taylor rule, mandate") label(3 "Taylor rule, exchange rate") ///
  label(4 "Taylor rule, copom") label(5 "Taylor rule, international interest") label(6 "Taylor rule, full model")) ///
  tlabel(2002m1 "2002" 2003m1 "2003" 2004m1 "2004" 2005m1 "2005" 2006m1 "2006" 2007m1 "2007" 2008m1 "2008" ///
  2009m1 "2009" 2010m1 "2010" 2011m1 "2011" 2012m1 "2012" 2012m12 "2013", labstyle(small_label) ) ///
  saving(taylor2, replace)

** 6.2 Moving average IR deviation graph
    ** Will show how much actual IR are diverging from taylor rules

line mov_idiff mov_idiff2 mov_idiff3 mov_idiff4 mov_idiff5 date if tin(2002m1,2012m12), name(moving_idff, replace) ///
  lwidth(thin thin thin thin thin) scheme(s2color)  ///
  title("Deviation of Actual Interest Rate from Taylor Rule", position(11) margin(vsmall)) ///
  subtitle("Moving average of 12 previous months, annualized rates (Jan 2002 - Dec 2012)",  position(11) margin(vsmall) size(small)) ///
  caption("Source: Author's calculations, with IPEA & BCB Data", size(vsmall)) yline(0)  ///
  legend(label(1 "Taylor rule, historical") label(2 "Taylor rule, mandate") label(3 "Taylor rule, exchange rates") ///
  label(4 "Taylor rule, copom") label(5 "Taylor rule, international interest") label(6 "Taylor rule, full model")) ///
   saving(taylor_dev2_mov, replace)

** 6.3 Average IR deviation graph
    ** Will show annual average deviations

egen idiff_ave = rowmean(idiff idiff2 idiff3 idiff4 idiff5)                    // creates a new variable with the average for different specifications
sort year
by year: egen idiff_ave_y = mean(idiff_ave)                                    // creates a new variable with annual averages

egen idiff_ave_sd = sd(idiff_ave)                                              // this is just to show how you can include variables in graphs

line idiff_ave idiff_ave_y date if tin(2002m1,2012m12), name(actual_idiff, replace) ///
  lwidth(thin thick) lpattern(solid dash) scheme(s2color) ///
  title("Deviation of Actual Interest Rate from Taylor Rule", position(11) margin(vsmall)) ///
  subtitle("Annualized rates (Jan 2002 - Dez 2012)",  position(11) margin(vsmall) size(small)) ///
  caption("Source: Author's calculations, with IPEA & BCB Data", size(vsmall)) yline(0, lcolor(black))  ///
  yline(`=-1*idiff_ave_sd' `=idiff_ave_sd', lcolor(white)) tlabel(2002m1 "2002" 2004m1 "2004" 2006m1 "2006" 2008m1 "2008" 2010m1 "2010" 2012m1 "2012")  ///
  legend(label(1 "Monthly") label(2 "Annual average")) ///
   saving(taylor_dev2, replace)
```

```
** 6.4 Deviation of inflation from target graph

sort date
gen mov_pidiff = ( l11.pidiff + l10.pidiff  + l9.pidiff  + l8.pidiff + ///
    l7.pidiff  + l6.pidiff  + l5.pidiff  + l4.pidiff  + l3.pidiff  + ///
    l2.pidiff  + l1.pidiff  + pidiff  ) / 12                                    // creates moving average of the inflation differential

line mov_pidiff date if tin(2005m1,2012m12), yline(0, lcolor(black)) yline(-2 2, lpattern(dash) ///
   lcolor(black)) scheme(s2color) tlabel(2005m1 "2005" 2007m1 "2007" 2009m1 "2009" 2011m1 "2011" 2012m12 "2013") ///
   title("Deviation of Actual Inflation from Inflation Target", position(11) margin(vsmall)) ///
   subtitle("Moving average of 12 previous months, annualized rates",  position(11) margin(vsmall) size(small)) ///
   ytitle("Inflation differential") name(pidiff, replace)

** Post estimation

// Graphing residuals

graph combine resid_1 resid_2 resid_3 ///
    resid_4 resid_5, cols(2) name(residpanel, replace) scheme(s2color)          // plots a panel chart with the residuals

// Unit root tests

qui scalar a = 1
foreach x in resid_1 resid_2 resid_3 resid_4 resid_5 {
 di _newline "Unit Root Test for resid_" a ", 1 lag"
 dfuller `x', lags(1)
 di _newline "Unit Root Test for resid_" a ", 2 lags"
 dfuller `x', lags(12)
 scalar a = a + 1
}
qui scalar drop _all

log close
```