Stata Session 1. Data Input

cd p:\econ382
[Set as the main folder the folder of your choice. The folder should be created before running this command]

log using log1.txt, text replace
[Create a log file in a text format. From now on all the output will be saved in this file]

set more off
[Continue running the output is it is too long to appear on one screen]

display 2+2
[can be used to calculate any expression]

How to open data: File - Open - choose the file in a .dta format

edit
[Open Data Editor]

browse
[Open Data Browser]

clear
[Clear the dataset used]

help clear
[help command returns a help window for this command]

How to input data:
1) Open Data Editor, Copy from the source and paste into the Data Editor. Save in .dta format.
2) Manual input
input unemp
210
220
350
end

[This allows you to manually create a variable unemployment]

3) Import from the Excel. To do that, save the Excel file in a .csv format. Then go File - Import - Spreadsheet - Choose the .csv format - Choose the file - Ok.
In some version, there is an option to import directly from Excel. File - Import - Excel Spreadsheet.

Stata Session 2. Data Management

list
list year
list year unemp
[List values of variables; lists all variables if specific variables are not indicated]

rename unemp unemployment
[Rename the variable unemp, the new name is unemployment]

su
sum
summarize
su lfp
[Summarize variables, returns mean, standard deviation, numbaer of values, minimum and maximum values]

su, detail
[Returns more statistics, such as percentiles, and etc.]

help operator
[Opens the list of algebraic operators, including the relational ones:
>   greater than
<   less than
>= > or equal
<= < or equal
== equal
!= not equal
~= not equal
Relational operators are used only for conditions, for example "if var<6"]

su if year>2005
su if year==2005   [su if year=2005 is a wrong command!]
su if year!=2005

describe
describe lfp
[Describes the variable, shows the label, storage type, and etc.]

label var lfp "Labor force participation rate in the U.S."
label var unemp "Unemployment rate in the U.S."
[Changes the label of a variable]

tab year
[Shows frequency distribution for a variable]

sort lfp
sort year
[Sort the values of the variable]

gen unempsq=unemp^2

gen percentlfp=lfp/100
gen uselessvariable= lfp+ unemployment
gen smallunemp=1 if unemployment<=5
gen k=6
[Generate a new variable on the basis of the existing ones, any numbers, and any algebraic operations. Be careful, save the data that includes new important variables.]

[If a variable contains missing values, then they are denoted with .]

replace smallunemp=2 if smallunemp==.
replace uselessvariable=1000 if year<2003
[Replaces some values of the variables]

drop unempsq
drop percentlfp uselessvariable smallunemp
[Drop variables]

drop if unemployment>8
[Drop observations]

Stata Session 3 Part 1. Simple Regression

twoway (scatter Consumption Income)
[Creates a scatter plot, the first variable is dependent (Y scale), the second is independent (X scale)]

twoway (lfit Consumption Income)
[Creates a linear prediction fit plot for two variables]

twoway (lfit Consumption Income) (scatter Consumption Income)
[Creates a graph with a scatterplot and a predicted fit line]

reg Consumption Income
regress Consumption Income
[Runs a regression of the Consumtion on Income; the first variable is dependent (Y), the second is independent (X)]

reg Consumption Income if Income>=150
[Runs a regression on a restricted sample; the observation with Income<150 are not included in the analysis]

Stata Session 3 Part 2. Multiple Regression

[Use Child Mortality dataset]

reg cm flr
reg cm flr pgnp
reg cm flr pgnp tfr
[Regress variable cm on different sets of independent variables]

[Use Education Expenditure dataset]

reg educ gdp pop
[Regress variable educ on the variables gdp and pop]

[How to get the distribution values or probabilities:

Standard normal distribution:
normal( ) - left-tail area
invnormal( ) - the value for a left-tail area

t distribution:
ttail( ) - right-tail area
invttail( ) - the value for the right-tail area

Chi-square distribution:
chi2( ) - left-tail area
chi2tail( ) - right-tail area
invchi2( ) - the value for a left-tail area
invchi2tail( ) - the value for the right-tail area

F distribution:
F( ) - left-tail area
Ftail( ) - right-tail area
invF( ) - the value for a left-tail area
invFtail( ) - the value for the right-tail area]

[for example, to find P(Z<-2)]
display normal(-2)

[to find the upper 30% value of Z]
display invnormal(0.7)

[to find P(t(18 df)>1)]
display ttail(18,1)

[to find the upper 30% value of t(18 df)]
display invttail(18,0.3)

[to find P(chisq(10 df)<6)]
display chi2(10,6)
[or]
display (1-chi2tail(10,6))

[to find the upper 30% value of Chisq(10 df)]
display invchi2(10,0.7)
[or]
display invchi2tail(10,0.3)

[to find P(F(5 df, 10 df)>7)]
display (1-F(5,10,7))
[or]
display Ftail(5,10,7)

[to find the upper 30% value of F(5,10df)]
display invF(5,10,0.7)
[or]
display invFtail(5,10,0.3)

Stata Session 4. Functional Forms of Regressions

[Use *Math SAT Score* dataset]
reg y x
[regress y on x]
predict yhat
[using the results of the last regression, compute predicted values of y, or yhats; save as a new variable named yhat]
predict res, residuals
[using the results of the last regression, compute residuals; save as a new variable named res]

gen manualres=y-yhat
[Suppose we want to check if the residuals found by Stata are in fact the residuals from our model. We can generate a new variable manualres, that equals (y-yhat). If we compare the variables res and manualres, we will see that they have the same values]
drop manualres

[Now that once we have residuals, we may try to check if they are normally distributed. The most apparent way to do it is to create a histogram of values. We can do it by going
Graphics – Histogram – Choose variable res – Choose number of bins – Choose frequency on Y-axis.]

---

reg y x
[Regress y on x (linear in x model)]
gen lny=ln(y)
gen lnx=ln(x)
reg lny lnx
[Regress lny on lnx (example of double log model)]
reg y lnx
[Regress y on lnx (example of lin-log model)]
reg lny x
[Regress lny on x (example of log-lin model)]

[Use *Production Function for Mexican Economy* dataset]
gen lngdp=ln(gdp)
gen lnlabor=ln(labor)
gen lncapital=ln(capital)
reg lngdp lnlabor lncapital
[This is another example of the model where both dependent variable and independent variables are in log form (double-log or "log-linear" model)]

[Use *Population Growth* dataset]
gen lnuspopulation=ln(uspopulation)

reg lnuspopulation time
[This is another example of log-lin model]
[NOTE: Log-lin and log-linear models are not the same! Log-linear is the same as double-log.]

[Use *Inflation and Unemployment* dataset]
reg inflrate unrate
gen unrateinv=1/unrate
reg inflrate unrateinv
[This is an example of a reciprocal model; comparing it with the linear model]

[Use *Hourly Wage and Age* dataset]
gen agesq=age^2
reg wage age agesq
[This is an example of polynomial model]

[Use *Political Instability and Size Interaction* dataset]
gen sizepins=size*pins
reg lnprod size pins sizepins
[This is an example of a model with an interaction term]

[Use again *Math SAT Score* dataset]
reg y x, nocons
[This is an example of regression through the origin (no constant regression)]

---

gen xthous=x/1000
label var xthous "Annual family income, thousands of dollars"
reg y xthous
[We are changing the units of measurement for the x variable. Now we can compare this model to the original one]
reg y x

---

egen standardized_y=std(y)
egen standardized_x=std(x)
[Generate standardized variables. This is the same as if we generated variables (y-ybar)/st.dev(y) and (x-xbar)/st.dev(x), where the means and the standard deviations could be taken from the summary report]
reg standardized_y standardized_x, nocons
[Run a regression of standardized variables]

---

set mem 1g
[This increases Stata store memory so that we can use large data file]

[Open *PUMS_NYC* dataset]

[Numeric (or quantitative) variables examples:
Age (age, years),
income (total personal income, dollars),
JWMNP (travel time to work, minutes).]
[For numeric variables, summary statistics is useful]
su Age income JWMNP
[For numeric variables, frequencies are not very useful]
tab Age
tab income
tab JWMNP

[Categorical (or qualitative) variables examples:
SEX (sex; 1=male, 2=female),
boroughs (borough of the person's home; 1=the Bronx, 2=Manhattan, 3=Staten Island, 4=Brooklyn, 5=Queens),
MAR (marital status; 1=married, 2=widowed, 3=divorced, 4=separated, 5=never married).]
[For categorical variables, summary statistics makes no sense]
su SEX boroughs MAR
[For categorical variables, frequencies are useful]
tab SEX
tab boroughs
tab MAR

[Binary (or logic, or indicator, or dummy) variables examples:
africanamerican (africanamerican race; 1=yes, 0=no),
kids_under6 (the person has kids under 6 years; 1=yes, 0=no),
foreign_born (the person was born outside the US; 1=yes, 0=no).]
[For binary variables, both the summary statistics and the frequency tabulation are useful]
su africanamerican kids_under6 foreign_born
tab africanamerican
tab kids_under6
tab foreign_born

[So binary variables have the features of both quantitative and qualitative variables]

	Numeric variables	Binary Variables	Categorical variables
Summary Statistics (su)	+	+	-
Frequencies (tab)	-	+	+

[We can create dummy variables using any variables.
Consider first how to create dummies out of categorical variables. We will use SEX (male=1, female=2)]

gen male=1 if SEX==1
replace male=0 is SEX==2
(or, instead, **replace male=0 if male==.)**

[Now we will create dummies out of boroughs (1=the Bronx, 2=Manhattan, 3=Staten Island, 4=Brooklyn, 5=Queens)]

gen bronx=1 if boroughs==1
replace bronx=0 if bronx==.

gen manhattan=1 if boroughs==2
replace manhattan=0 if manhattan==.

gen statenisland=1 if boroughs==3
replace statenisland=0 if statenisland==.
[and so on.]

[There is a special command that automatically creates dummies for each category of categorical variable]
tab boroughs, gen(b)
tab MAR, gen(marst)

[Now consider how to create dummies out of numeric variables. We may be interested in using such variables in some cases. We will use the variable Age]
gen young=1 if Age<=25
replace young=0 if young==.

[We can have more groups]
gen old=1 if Age>65
replace old=0 if old==.

gen midage=1 if young==0&old==0
replace midage=0 if midage==.

[Recall that "&" means "and" (intersection), "|" means "or" (union)]

[We may create a dummy for more than one category of a qualitative variable]
gen brq=1 if boroughs==4|boroughs==5
replace brq=0 if brq==.

[Use *PUMS_NYC* dataset]

reg income male
[Regress dependent variable on one dummy]

reg income male Age
[Regress dependent variable on one dummy and one quantitative variable]

reg income male africanamerican
[Regress dependent variable on two dummies]

reg income b1-b4
[Regress dependent variable on several dummies that represent one categorical variable. The variable b5 is excluded to avoid multicollinearity.]

reg income b1 b3 b4 b5
[The same case as the previous one; now b2 is excluded]

reg income Age WKEXREL male africanamerican nativeamerican asianamerican Hispanic kids_under6 foreign_born marst1-marst4
[Regress dependent variable on many quantitative and qualitative variables]

gen formanh= foreign_born* b2
[This generates the interaction term for foreign_born and Manhattan]
reg income foreign_born b2 formanh
[Regress dependent variable on two dummies and their interaction]

gen agemale=Age*male
[This generates the interaction term for Age and male]
reg income male Age agemale
[Regress dependent variable on one quantitative variable, one dummy, and on their interaction]

[Use *Refrigerator Sales* dataset]

reg refrigerator_sales quarter2 quarter3 quarter4
[This is an example of seasonal analysis]

[Use again *PUMS_NYC* dataset]

gen manhattan=1 if boroughs==2
replace manhattan=0 if manhattan==.

gen male=1 if SEX==1
replace male=0 if male==.

tab educ_indx, gen(ed)

reg manhattan ed2-ed6 Age male
[This is an example of a linear probability model, where the dependent variable is binary. We are predicting the probability that a person lives in Manhattan given his age, gender and level of education.]

Stata Session 6. Heteroscedasticity

[Use *Los Angeles Restaurants* dataset]
reg price food service

predict pricehat
[Using the results of the last regression, compute predicted values of dependent variable, or yhats; save as a new variable named pricehat]
predict res, residuals
[Using the results of the last regression, compute residuals; save as a new variable named res]

---

gen ressq=res^2
[Generate squares of residuals; save as a new variable named ressq]

histogram res
histogram ressq
[Create histogram of residuals or squared residuals distribution]

twoway (scatter ressq food)
twoway (scatter ressq service)
[Create residual plot: squared residuals versus explanatory variable, to detect a possible heteroscedasticity]

---

reg ressq food
reg ressq service
reg ressq pricehat
[Breusch-Pagan test: regress squared residuals on the X-variable; if more than one X are correlated with residuals, then regress squared residuals on Yhat (since Yhat is a linear combination of X's.]
[If F-test is significant, then there may be heteroscedasticity.]

gen lnressq=ln(ressq)
gen lnpricehat=ln(pricehat)
reg lnressq lnpricehat
[Park test: regress log of squared residuals on X-variable (or Yhat)]
[If the coefficient on the explanatory variable used is significant, then there may be heteroscedasticity.]

gen absres=abs(res)
gen rootpricehat=sqrt(pricehat)
gen pricehatinv=1/pricehat
reg absres pricehat
reg absres rootpricehat
reg absres pricehatinv
[Glejser test in three versions: regress absolute value of residuals on X, or square root of X, or X inverse]
[If the coefficient on the explanatory variable used is significant, then there may be heteroscedasticity.]

gen foodsq=food^2
gen servicesq=service^2
gen foodservice=food*service

reg ressq food service foodsq servicesq foodservice
[White's test: regress squared residuals on X's, there squares and cross-products.]
[If n*R-squared is higher than Chi-Sq(k-1), then there may be heteroscedasticity.]

---

reg price food service, robust
[Run a regression, but estimate standard errors using White's heteroscedasticity correction.]

---

twoway scatter (ressq pricehat)
[Create a residual plot to check if there is linear or a quadratic relationship between squared residuals and X's]

[If the relationship is linear]
gen y=price/rootpricehat
gen x1=1/rootpricehat
gen x2=food/rootpricehat
gen x3=service/rootpricehat
reg y x1 x2 x3, nocons
[Perform a square root transformation of the original regression to stabilize the variance of disturbances.]
[At this point we can try to check if we got rid of heteroscedasticity, using any of the methods. We predict residuals and test them for relationship with X's.]
predict yhat
predict r, residuals
gen rsq=s^2
reg rsq yhat
[If the F-test is not significant, then we actually got rid of heteroscedasticity.]

[If the relationship is not linear]
gen y=price/pricehat
gen x1=1/pricehat
gen x2=food/pricehat
gen x3=service/pricehat
reg y x1 x2 x3, nocons
[Perform a square root transformation of the original regression to stabilize the variance of disturbances.]
[At this point we can try to check if we got rid of heteroscedasticity, using any of the methods. We predict residuals and test them for relationship with X's.]
predict yhat
predict r, residuals
gen rsq=s^2
reg rsq yhat
[If the F-test is not significant, then we actually got rid of heteroscedasticity.]

---

gen lnprice=ln(price)
gen lnfood=ln(food)
gen lnservice=ln(service)
reg lnprice lnfood lnservice
[Respecification of the original regression. May be used if the log-linear functional form is also acceptable, which is not always the case.]

Stata Session 7. Autocorrelation

[Use *Dividends and Corporate Profits* dataset.]

reg div prof
predict res, residuals
egen time=seq( )
[Create a sequence variable (1, 2, 3, 4, etc.) named *time*.]

twoway (scatter res time)
[Plot residuals against time to assess the possible autocorrelation]

tset time
[Declare data to be time series data. This step is necessary in order to run the next command.]

gen reslag=res[t-1]
[Generate a lagged variable for *res* named *reslag* (The first value of *reslag* is the second value of *res*; the second value of *reslag* is the third value of *res*, and etc.).]

twoway (scatter res reslag)
[Plot residuals against one period lagged residuals to assess the possible autocorrelation]

estat dwatson
[Perform Durbin-Watson test for autocorrelation.]

gen x=res - reslag
gen xsq=x^2
egen xx=sum(xsq)
display 70377.805/118661.179
[The same test (DW) done manually. The results are the same.]

runtest res, thresh(0)
[Perform runs test, use the variable res and set threshold level at 0. (If this option is not specified, the default threshold is the median value).]

gen m=1 if res>0
replace m=0 if res<0
tab m
[The first steps of the runs test. There is no fast way to find manually number of runs.]

[Model transformations: using different values of rho.]

gen divlag=div[t-1]
gen proflag=prof[t-1]
[Generate one period lagged variables for dependent and independent variables]

gen rho1=1

[Set the first possible estimate of rho as 1]
gen rho2=1- (.5930988/2)
[Find the second possible estimate of rho out of DW test]

reg res reslag
gen rho3=.7288331
[Find the third possible estimate of rho from the residuals regression]

[Run the first transformed regression]
gen divstar1=div- rho1* divlag
replace divstar1=div*sqrt(1-rho1^2) in 1
gen profstar1=prof-rho1*proflag
replace profstar1=prof*sqrt(1-rho1^2) in 1
reg divstar1 profstar1, nocons

[Test it for autocorrelation]
estat dwatson
predict resstar1, residuals
runtest resstar1, thresh(0)

[Run the second transformed regression]
gen divstar2=div- rho2* divlag
replace divstar2=div*sqrt(1-rho2^2) in 1
gen profstar2=prof-rho2*proflag
replace profstar2=prof*sqrt(1-rho2^2) in 1
reg divstar2 profstar2

[Test it for autocorrelation]
estat dwatson
predict resstar2, residuals
runtest resstar2, thresh(0)

[Run the third transformed regression]
gen divstar3=div- rho3* divlag
replace divstar3=div*sqrt(1-rho3^2) in 1
gen profstar3=prof-rho3*proflag
replace profstar3=prof*sqrt(1-rho3^2) in 1
reg divstar3 profstar3

[Test it for autocorrelation]
estat dwatson
predict resstar3, residuals
runtest resstar3, thresh(0)

---

newey div prof, lag(1)
[Run a regression with Newey-West HAC (heteroscedasticity and autocorrelation) corrected standard errors. Compare to the original OLS regression: standard errors were underestimated.]