EViews

29. Juni 2010

Inhaltsverzeichnis

1 Introduction 2

2 Assignment 1 5

3 Assignment 2a 6

4 Assignment 2b 7

5 Assignment 3 10

6 Assignment 3b 12

7 Assignment 4_neu (using programs) 13

8 Assignment 5b 17

9 Assingment 6 18

1 Introduction

The EViews Window

Component	Purpose
Drop Down Menus	easy-to-use tools for implementation of EViews procedures
Command Line	alternative to drop down menus for usage of command processing language at advanced level
Status Line	message area indicating default directory (Path), default database (DB), default workfile (WF) Work Area
Work Area	display of various object windows
Help Menu	provision of helpful advice

Object-orientied language and event-oriented language.

The Help System

- Help/Users Guides or Help/Command&Programming Reference: provide a complete documentation of EViews on over 2000 pages
- Help/Quick Help Reference/Function Reference: overview of functions available for easy use
- Help/Quick Help Reference/Object Reference: overview of objects available that help to organize your workfile, e.g. equation, series, etc.

Using a Workfile

Open the workfile by clicking File/Open/EViews Workfile and indicating the directory

- series: gdp (gross domestic product), m1 (money supply), pr (price level), rs (short term interest rate), resid (explained later)
- coef: c (coefficient vector used to represent parameters of equations)
- buttons: view, proc, object, etc.
- data set information: range, sample

Empty workfile (File/New/Workfile→specify data) still contains the objects: constant, residual

You always have to save the results of a regression with a proper name thus creating an object.

Working with Objects
Objects are collections of related information and operations that are bundled together into easy-to-use units that you hold in a workfile.

- creating objects: click Object/New Object, select object you desire and name it meaningfully. For taking notes: Type „Text“
- useful objects: text, series, equation, graph, group, etc.

Examining a Single Series

- spreadsheet view: double-click on series, e.g. gdp
- different sample period: click Sample and change sample range. Can de done in the main window or when the series in opened as a graph
- descriptive statistics: click View/Descriptive Statistics/... (Histogram & Stats)
- graphs: click View/Graph/Line
- saving graphs: rightclick on graph and save graph to disk: *.emf for inclusion in word and *.eps for inclusion in latex documents
- copying results into word: copy (STRG+C) and paste (STRG+V)
- freezing results: click Freeze and name object via Name as to create a new object in your workfile

Examining Several Series

- spreadsheet view: click on series1+STRG+series2, double-click in blue area and select open group
- summary statistics: click View/Descriptive Stats/Common Sample
- plotting two series: click View/Graph/Line or View/Multiple Graphs/ Line. Graph shows both series in one graph and multiple graphs shows one graph for each series
- scatter diagram: click View/Multiple Graphs/Scatter/...SCATMAT: matrix containing scatter-plots for all series

Group objects: Strg (single marks), Shift (mark tow, inbetween everything is makred). After marking click right/Open/As group/Name. If you don't name it, you get a window where you can either name or store the object

Using the Quick Menu

- changing sample: click Quick/Sample and proceed as before
- graphs: click Quick/Graph/Line etc.
- generating new series: click Quick/Generate Series and enter equation according to seriesname=function(series) etc. eg. log()

Using EViews Functions
Basic arithmetic operations and basic math functions can be used when working with series, such as in operations to generate new series (NOTE: some functions require the „@“ symbol to be identified by EViews!):

- click Quick/Generate Series and enter equation or
- use the Command Line via series seriesname=function(series)

Note that in the command line the first entry is always the object type (series, scalar, etc.) you want to create!

Descriptive statistics functions can solely be used within the Command Line.

Mean: scalar seriesname=@mean(series)
Std: scalar seriesname=@stdev(series)
→can be seen in status line
→recall: Define new object: Matrix-Vector-Coef. (insert how many values needed), then:
Vectorname(1)=@mean(series)
Vectorname(2)=@stdev(series)

Getting Data into EViews

- open *.xls and memorize number of obs, labels and ordering of variables
- close *.xls
- create empty workfile via File→New→Workfile and indicate parameters (structure type; data range)
- click Proc→Import→Read Text Lotus Excel, select data type *.xls and browse
- indicate appropriate parameters (indicate all series separated by a „Leerzeichen“), label data and save workfile

Other ways:

- Go to the folder where you saved the excel file, then drag it and drop it in the grey part of EViews→“Fertigstellen“

- Click right in the grey part of EViews→ Open→ Foreign data as workfile→ choose excel file→ Fertigstellen OR Open→ EViews Workfile→ Dateityp: .xls→Choose excel file→ Fertigstellen

2 Assignment 1

1.3
Rename an object: click right on object→rename (for workfile: save)
To change the order of the variables in a group, use „Spec“

1.4
View→Descriptive stats→Common table (for a graph)→Freeze (to save)

1.5
View→Graph→Scatter→Simple scatter→Freeze (to save)
To change graph:
Add titel: AddText
Scaling: Option→Axes/Scales→choose Axis under „Edit Axis“ and Scaling „User specified“
Symbols: Option→Line/Samples→Different options
Save to disk: Proc→Save graph to disc

2.1
Quick→Estimate equation→enter variables: food_exp c income OR food_exp=c(1)+c(2)*income and choose Method: LS→ save the output through „name“

2.2
c and resid are now filled, but are overwritten after each equation is computed

2.3
(i) c(1) insignificant on 5% significance level, c(2) significant on all conventional significant levels
(ii) If income increases by 1 unit (100$), food_exp increases by 10 units (1$)
(iii) R^2=0,385

3.1
To create predicted values after estimating:
series predictedvalues=@coefs(row of object c)+@coefs(row of object c)*x-variables

Quick way: after estimating click on „Forecast“ and enter a name to save it

3.2

- Series resid1=resid (this way the residuals are saved and don't change after the next regression)
- Series resid1=food_exp - predictedvalues

3.3
Open predictedvalues and resid1 as a group and use scatterplot.
Quick→graph→scatter→enter series names (with „Leerzeichen").
You see heteroscedasticity
Line/shade: indicate orientation and position („0")

3 Assignment 2a

2.1
File→New→Program
Mark whether a sentence is supposed to run or not: 'text prevents running

2.2
To open a workfile: wfopen"Path" → Run

2.3
To form a group: group name objects (eg. group grp1 advert price sales)
Open the descriptive statistics: name.stats (eg. grp1.stats)

2.4
Estimate a regression: equation name.typ y-var c x1-var x2-var (eg. equation reg1.ls sales c price advert)

Procs:
Compute residuals: name.makeresid name (eg. reg1.makeresid residuals)
Compute predicted values: reg1.forecast name (eg. reg1.forecast predicted_data)
Create scatterplot: graph name.scat name1 name2 (eg. graph regscat.scat predicted_data residuals)
Compute R-squared: scalar name=name.@r2 (eg. scalar firstscalar=reg1.@r2)
To save: save

Programs are superior for repetitive tasks

3.1
t-stats:
scalar name=@coefs(No)/@stderrs(No)
(Besser: scalar name=reg1.@coefs(No)/reg1.@stderrs(No))

3.2
p-value (2-seitig):
scalar name=2*(1-@ctdist(name t-stat,n-k-1)) (bei positivem t-stat)
scalar name=2*(@ctdist(name t-stat,n-k-1)) (bei negativem t-stat)

3.3
Critical value:
scalar name=@qtdist(1-α/2,n-k-1)

3.5
t-stats:
scalar name=@coefs(No)-Wert/@stderrs(No)

p-value (1-seitig):
scalar name=1-@ctdist(name t-stat,n-k-1)) (bei positivem t-stat)
scalar name=@ctdist(name t-stat,n-k-1)) (bei negativem t-stat)

Critical value:
scalar name=@qtdist(1-α,n-k-1)

Appendix
Data members of equation (are computed together with eq., so we can access them using the following comments):
@aic Akaike information criterion
@coefcov(i,j) Covariance of coefficient estimates i and j
@coefs(i) i-th coefficient value
@f F-statistic
@logl Value of the log likelihood function
@meandep Mean of the dependent variable
@ncoef Number of estimated coefficients
@r2 R-squared statistic
@regobs Number of observations in regression
@schwarz Schwarz information criterion
@sddep Standard deviation of the dependent Variable
@se Standard error of the regression
@ssr Sum of squared residuals
@stderrs(i) Standard error of coefficient i
@tstats(i) t-statistic value for coefficient i
c(i) i-th element of default coefficient vector for equation
@coefcov Covariance matrix for coefficient estimates
@coefs Coefficient vector
@stderrs Vector of standard errors for coefficients
@tstats Vector of t-statistic values for coefficients

4 Assignment 2b

1.1-2.6
Siehe Assignment 2a

3.1
Set sample range: smpl 1 n°obs

3.2
Lower bound of confidence interval: scalar name=@coefs-@stderrs*critical value
Upper bound of confidence interval: scalar name=@coefs+@stderrs*critical value

3.3
We can choose any $\overline{c}$ inside the confidence interval.

4.2
series name=@dnorm(x) (x here a series)
scalar name=@dnorm(x) (x here a scalar)

4.3
series name=@dtdist(x,df) (x here a series)
scalar name=@dtdist(x,df) (x here a scalar)

4.4
Take the series compute in 4.2 and 4.3, open as a group and than „view"→"graph"→"line"

4 using a program
Compare the distributions of a normal distribution and a t-distribution (with 4 different degrees of freedom) in a graph:

How to create a vector:
vector (No. of elements) vectorname
vectorname(1)=number or path
vectorname(2)=number or path
... (depending on number of element)

TEXT OF PROGRAM:

```
Vector(4) df
df(1)=4
df(2)=10
df(3)=100
df(4)=2000

series normal=@dnorm(x)

for!i=1 to 4
series tdist{!i}=@dtdist(x,df(!i))
group joint{!i} normal tdist{!i}
freeze(grafik{!i}) joint{!i}.line
next

freeze(grafikall) grafik1 grafik2 grafik3 grafik4
show grafikall.align(2,1,1)
```

ELEMENTE:
Vector name: df
Group name: joint

Graph name: grafik
Joint graph name: grafik all
Elements of align: align(n,h,v) with n=number of columns, h=horizontal space, v=vertical space
freeze(name of graph) to save a graph and not only show it
{!i} as a replacement

5.1
$c = z_0 = z_1 = 0$

$$R = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} * \beta = \begin{pmatrix} \mu \\ c \\ z_0 \\ z_1 \end{pmatrix} = r = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{1}$$

5.2
scalar fstat=reg1.f
scalar pval=1-@cfdist(fstat,3,2773) (allg.: scalar name=1-@cfdist(fstat,#r,n-k))

5.3
scalar critval=@qfdist(0.95,3,2773)

5.4
Equation object→View→Coefficient tests→Wald-Coefficient Restrictions→c(3)=0, c(4)=0

5.5
View→Coefficient tests→Wald-Coefficient Restrictions→c(3)=0, c(4)=0, c(2)=0.005

$$R = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} * \beta = \begin{pmatrix} \mu \\ c \\ z_0 \\ z_1 \end{pmatrix} = r = \begin{pmatrix} 0 \\ 0 \\ 0.005 \end{pmatrix} \tag{2}$$

5 using a program

Estimate the equation
equation reg1.ls delta_p c delta_q q qv

Estimate the variance-covariance matrix
c: series interc=1 (we define a constant)
X: group $\underbrace{reggr}_{\text{name for X}}$ interc delta_q q qv (create a group containing all variables)
n-k: scalar degfr=reg1.@regobs-reg1.@ncoef (calculate the degrees of freedom)

X'X: $\underbrace{matrix(4,4)}_{\text{define object}}$ $\underbrace{secmom}_{\text{name for X'X}}$ $= @$ $\underbrace{inner}_{\text{for the innter product}}$ $(reggr)$ (Alternative: X'X=@transpose(X)*X)
$(X'X)^{-1}$: matrix $\underbrace{insecmom}_{name for (X'X)^{-1}}$ =@inverse(secmom)

e'e/n-k: scalar ssqrd=reg1.@ssr/degfr (mit ssr: sum of squared residuals)

⇒ matrix estvarcov=insecmom*ssqrd

Short Way:
$\underbrace{matrix}_{\text{object type}}$ evmat=reg1.$\underbrace{@coefcov}_{\widehat{Var(b|X)}}$

Matrix of the linear restrictions
R: matrix $\underbrace{(2,4)}_{\text{dimension}}$ $\underbrace{linrest}_{\text{name}}$ (Define empty matrix)

linrest(1,3)=1 $(H_0 : z_0 = 0)$
linrest $\underbrace{(2,4)}_{\text{field of matrix}}$ =1 $(H_0 : z_1 = 0)$

Vector of the restrictions
r=Rb: vector(2) restvec=linrest* $\underbrace{reg1.@coefs}_{\text{vector with all coefficients}}$

scalar nrest=@rows(linrest) (number of restrictions)

Quadratic form
vector(1) f_ratio=@transpose(restvec)*@inverse(linrest*estvarcov*@transpose(linrest))*restvec/nrest

Testing
scalar fcritval=@qfdist(0.95,nrest,degfr) (compute the critical value for the F-distribution; in general: @qfdist(1-α,#r,n-k))

Use Conditional Statement:
if fcritval>f_ratio(1) then
%dec="The null hypothesis cannot be rejected at a 5% significance level"
else
%dec="The null hypothesis can be rejected at a 5% significance level"
end if

% used to define a string
alpha decision=%dec
show decision

5 Assignment 3

1.4
View→Distribution→Kernel Density Graphs (nonparametric method of smoothing)→Kernel (doesn't matter with large sample size= leave standard)

2.1
ls wage c educ exper (zum Speichern: equation name.ls wage c educ exper)

2.3
If you divide the dependent variable by a constant, the parameters also have to be divided by the same constant

2.6
If you multiply the independent variable by a constant, the parameters changes accordingly being divided by that constant.

2.7
Standardized series:
series wage_z=(wage-@mean(wage))/@stdev(wage)

2.8
$\widehat{\beta}_j$: @coefs(i); $\widehat{\sigma}_j$: @stdev(xi); $\widehat{\sigma}_y$: @stdev(y)
→ scalar name=@stdev(xi)/@stdev(y)*eq1.@coefs(i)

3.1
log(variable) is the function for the natural logarithm of a variable.

Appendix: Theory
Linear transformation of regressors
Let x be the nxk matrix of the regressors and A a kxk nonsingular linear transformation matrix. Postmultiplying x by A yields:
$\underbrace{xA}_{B} = x[a_1, ..., a_k] = [xa_1, ..., xa_k]$
A projection matrix looks like this: $B(B'B)^{-1}B'$
This leads to: $P_B = P_{Ax} = xA(A'x'xA)^{-1}A'x' = xAA^{-1}(x'x)^{-1}(A')^{-1}A'x' = x(x'x)^{-1}x' = P_x$
With linear transformations we only change the coefficients, but the fitted values stay the same. Since the vectors of fitted values and residuals e depend on x only through P_x and M_x, they are invariant to any nonsingular linear transformation of the columns of x.

Adding or substracting constants to an independent variable doesn't change the coefficient.

Create deviation from the mean for a series:
series educabw=educ-@mean(educ) (the sum of this series is 0)
For the regression „wage c educabw" and „wage educabw" the coefficient is the same (as c and educabw are orthogonal).

Standardization
Starting point: original OLS equation
$y_i = b_0 + b_1 x_{i1} + ... + b_k x_{ik} + e_i$
$y_i - \overline{y} = b_0 - b_0 + b_1 x_{i1} - b_1 \overline{x_1} + ... + b_k x_{ik} - b_k \overline{x_k} = b_1(x_{i1} - \overline{x_1}) + ... + b_k(x_{ik} - \overline{x_k}) + e_i$
Let $\widehat{\sigma}_y$ be the sample standard deviation of y, $\widehat{\sigma}_x$ the sample standard deviation of x.

$$\frac{(y_i - \overline{y})}{\widehat{\sigma}_y} = \frac{\widehat{\sigma}_{x_1}}{\widehat{\sigma}_y} * b_1 * \frac{(x_{i1} - \overline{x_1})}{\widehat{\sigma}_{x_1}} + ... + \frac{\widehat{\sigma}_{x_k}}{\widehat{\sigma}_y} * b_k * \frac{(x_{ik} - \overline{x_k})}{\widehat{\sigma}_{x_k}} + \frac{e_i}{\widehat{\sigma}_y} \tag{3}$$

If x_j increase by one standard deviation, then $\widehat{y}$ changes by $\widehat{\beta}_j$ standard deviations.

6 Assignment 3b

3.1
Enter for the observation: 1 200

3.2
For educ: eq.@coefs(2)$\pm$1.96*eq.@stderrs(2) (bzw. statt 1.96: -@qtdist(0.025,n-k))
$\rightarrow$ The one with the lower n is wider (as we have smaller stderrs for higher n and a different t-stat). Our aim is a narrow interval.

3.3
β_1=85 cannot be rejected.
We do not reject for all values inside the confidence interval: [64,88]

4.1
log(variable) for ln

4.3
We can reject at any conventional significance level as p-value is close to 0.

4.4
Centered R^2: 1-SSR/SST
$\rightarrow$SSR=eq6.@ssr; SST=$(eq6.@sddep)^2$*(n-1)

Adjusted R^2: 1-(n-1/n-k)SSR/SST

4.5
Adjusted R^2: (5) 0.129; (6) 0.179
Akaike: (5) 0.973; (6) 0.916
Schwarz: (5) 0.989; (6) 0.941
$\rightarrow$ all criteria point toward equation 6

4.6
AIC=log(SSR/n)+2k/n
SBC=log(SSR/n)+log(n)k/n

Maximized log-likelihood under the normality assumption:
$L = -n/2 * log(2\pi) - n/2 * log(\widetilde{\sigma}) - \frac{1}{2\widetilde{\sigma}^2}(y - X\widetilde{\beta})'(y - X\widetilde{\beta})$
$L = -n/2 * log(2\pi) - n/2 * log(\frac{e'e}{n}) - \frac{1}{2(\frac{e'e}{n})} * e'e$
$L = -n/2 * log(2\pi) - n/2 * log(\frac{SSR}{n}) - \frac{SSR}{2\frac{SSR}{n}}$
$L = -n/2 * log(2\pi) - n/2 - n/2 * log(\frac{SSR}{n})$
$L = -n/2(1 + log(2\pi) + log(\frac{SSR}{n}))$

The formula used by Eviews:
AIC=-2(L/n)+2k/n
SBC=-2(L/n)+k*log(n)/n

For the L use: eq6.@logl

4.7
Siehe 4.5

7 Assignment 4_neu (using programs)

1
Create a workfile: workfile $\underbrace{simultest}_{name}\ \underbrace{u}_{undated}\ \underbrace{11000}_{samplesize}$

1.1
Construction of the mode:
x: series $\underbrace{Regress}_{Name} = @\ \underbrace{runif}_{uniformdist}\ (\underbrace{6}_{a}, \underbrace{12}_{b})$
β-vector:
vector(2) $\underbrace{Beta}_{Name}$
Beta(1)=5
Beta(2)=0.5

→series $\underbrace{\widehat{y}}_{Name}$ =Beta(1)+Beta(2)*Regress

About distributions: q=quantil, d=density, r=random

2
2.1
Decision variables: !m=1000

vector(!m) $\underbrace{betavec}_{Name}$
smpl @first 20 (we cut the sample size down to 20)

The loop:
For !k=1 to !m
    series y_stoch=$\widehat{y} + \underbrace{@rtdist(5)}_{epsilon}$
    equation reg01.ls y_stoch c Regress
    betavec(!k)=reg01.@coefs(2)
Next

2.2
Series are dependent on the sample size, vectors aren't. But series have lots of view, that can't be used with a vector.
Change sample size to 1000 as in the vector: smpl @first !m

Transform: $\underbrace{mtos}_{matrixtoseries}$ ($\underbrace{betavec}_{vectorname}$, $\underbrace{betaser}_{seriesname}$) (where betaser is a series that is created new here)

Plot histogram: freeze($\underbrace{Kern}_{Name}$) betaser.hist (include Jarque-Bera test)

Plot empirical distribution test: freeze(Tests) betaser.edftest (4 tests to test for a normal distribution)

3
General:
{!i} as part of a name
(!i) as a variable or parameter

3.1
workfile simultest u 1 2000

Definition of the sample sizes:
vector(4) stp
stp(1)=10
stp(2)=20
stp(3)=300
stp(4)=2000

Construction of the model as in 1.1

Decision variables: !n=1000

The loops:
```
For !i=1 to 4
    !stp=stp(!i)
    smpl @first !stp (sets the sample sizes)
    vector(!n) betavec{!i} (This time we create four vectors; one for every sample size)

    For !k=1 to !n
        series y_stocj=ŷ+@rtdist(5)
        equation reg01.ls y_stoch c Regress
        betavec{!i}(!k)=reg01.@coefs(2)
    Next

Next
```

3.2
Just change the parameters of @runif(a,b)

4
4.1
Creation of series and graphics:
smpl @first !n

```
For !i=1 to 4
    mtos(betavec{!i},betaser{!i})
    freeze(Kern{!i}) betaser{!i}.hist
    freeze(Tests{!i}) betaser{!i}.edftest Next
```

(This loop could be part of the upper loops after the first next)

freeze(endgra) Kern1 Kern2 Kern3 Kern4
show endgra.align(2,3,1) (N°1: number of columns; N°2: horizontal space; N°3: vertical space)

Added task: use of either the uniform or the t-dist
Construction of the model as in 1.1; definition of the sample sizes as in 3.1

Decision variable:
!n=1000
!switch=1
!vert=1

The loop:

```
For !i=1 to 4
    !stp=stp(!i)
    smpl @first !stp
    vector(!n) betavec{!i}

    For !k=1 to !n
        If !vert=1 then
            Series y_stoch=ŷ+@runif(-2,2)
        Else
            series y_stoch=ŷ+@rtdist(5)
        End if

        equation reg01.ls y_stoch c Regress
        betavec{!i}(!k)=reg01.@coefs(2)

    Next

Next
```

5
5.1-5.4
Subroutine name(output, arguments)
...
endsub

subroutine jbt(scalar p,series y) series spow=$(y-@mean(y))^2$ series thpow=$(y-@mean(y))^3$
series fpow=$(y-@mean(y))^4$

scalar smom=@mean(spow) scalar thmom=@mean(thpow) scalar fomom=@mean(fpow)

scalar sch=thmom/($smom^(3/2)$) scalar woelb=fomom/($smom^(2)$)

scalar jbtest=(@obs(y)/6)*(sch^2+(($woelb-3)^2$)/4) scalar p=1-@cchisq(jbtest,2) end-sub

Formeln:

$$JB = \frac{n}{6}(S^2 + \frac{(K-3)^2}{4}) \tag{4}$$

mit

$$S = \frac{\frac{1}{n}\sum(y-\overline{y})^3}{(\frac{1}{n}\sum(y-\overline{y})^2)^{3/2}} \tag{5}$$

und

$$K = \frac{\frac{1}{n}\sum(y-\overline{y})^4}{(\frac{1}{n}\sum(y-\overline{y})^2)^2} \tag{6}$$

5.5
We use the programm text from the added task, then we compute:

```
smpl @first !n

For !i=1 to 4
    scalar p{!i}
    mtos(betavec{!i},betaser{!i})
Alternative computation (instead of those two lines)
    call jbt(p{!i},betaser{!i})

    if !switch=1 then
        freeze(Kern{!i}) betaser{!i}.kdensity
    else
        freeze(Kern{!i}) betaser{!i}.hist
    endif

    freeze(Tests{!i}) betaser{!i}.edftest

next
```

Composite graph:
freeze(endgra) Kern1 Kern2 Kern3 Kern4
show endgra.align(2,3,1)

5.6
Local (after subroutine, before the name): prevents the subroutine from producing everything. With local only the thing specified as output is stored and the other ones are lost after the output is created.

8 Assignment 5b

Theory CAPM

$$E[r_{jt} - r_f] = \beta_j * E[r_{mt} - r_f] \tag{7}$$

where:
r_{jt}=risky asset return on asset j (in period t)
r_{mr}=risky return of the market portfolio
r_f=riskless return (time invariant)
$\beta = \frac{Cov(r_{jt},r_{mt})}{Var(r_{mt})}$=proportionality factor

Regression model:
$v_{jt} = r_{jt} - E(r_{jt})$
$v_{mt} = r_{mt} - E(r_{mt})$

7 becomes then:
$r_{jt} - v_{jt} - r_f = \beta_j(r_{mt} - v_{mt} - r_f)$
$r_{jt} - r_f = \beta_j(r_{mt} - r_f) + \underbrace{v_{jt} - \beta_j v_{mt}}_{\epsilon_{jt}}$

It holds:
$E(\epsilon_{jt}) = E(v_{jt}) - \beta_j E(v_{mt}) = 0$

Since $\beta_j = \frac{Cov(r_{jt},r_{mt})}{Var(r_{mt})} = \frac{E[(r_{jt}-E(r_{jt}))(r_{mt}-E(r_{mt}))]}{E[(r_{mt}-E(r_{mt}))^2]} = \frac{E[v_{jt}-v_{mt}]}{E(v_{mt}^2)}$

It holds:
$E[\epsilon_{jt}(r_{mt} - r_f)] = E[(v_{jt} - \beta_j v_{mt})(r_{mt} - r_f)] = E[(v_{jt} - \beta_j v_{mt})(v_{mt} + \underbrace{E(r_{mt}) - r_f}_{\text{non-stochastic}})]$

$= E[(v_{jt} - \beta_j v_{mt}) * v_{mt}] = E[v_{jt} * v_{mt}] - \beta_j E[v_{mt}^2] = E[v_{jt} * v_{mt}] - \frac{E[v_{jt} * v_{mt}]}{E(v_{mt}^2)} * E[v_{mt}^2]=0$

1.1
Open workfile, then Proc→Structure/Resize Current Page=> Enter data structure

1.4-2.2
Use a program:

```
for !i=1 to 10
series r{!i}rf r{!i}-rf
equation eq{!i}.ls r{!i}rf c rmrf
next
```

2.3
Before loop: vector(10) tstat
As part of loop: tstat(!i)=(eq{!i}.@coefs(2)-1)/eq{!i}.@stderrs(2)

For p-value:
Before loop: vector(10) pvalue
As part of loop: pvalue(!i)=2*(1-@cnorm(@abs(tstat(!i))))

Matrix instead of vector:
matrix $\underbrace{(2,2)}_{Zeilen,Spalten}$ $\underbrace{a}_{Name}$
a(1,1)=Zahl
a(1,2)=Zahl
a(2,1)=Zahl
a(2,2)=Zahl

2.4
Before loop: vector(10) constant
As part of loop: constant(!i)=eq{!i}.@coefs(1)

3.1
for !i=1 to 10
euqation equ{!i}.ls r{!i}rf c rmrf smb hml umd
next

3.2
Use R^2_{adj} with @rbar2, Akaike with @aic and Schwarz with @schwarz. Example for R^2_{adj}:
Before loop: vector(10) radj
As part of the loop: radj(!i)=equ{!i}.@rbar2

3.3
freeze(wald{!i}) $\underbrace{Equ}_{Name}$.wald Conditions (here: c(3)=0, c(4)=0, c(5)=0)
All p-values are close to zero

9 Assingment 6

Theory
1. Homoscedasticity assumption:
$Var(\epsilon_i|x) = E(\epsilon_i|x) = \sigma^2$ for i=1,...,n

2. Heteroscedasticity:
$Var(\epsilon|x) = E(\epsilon^2|x) = \sigma_i^2$

3. Consequences of heteroscedasticity:
Under assumption 1.1-1.3 and 1.5 OLS estimator unbiased but no longer BLUE and t- and F-tests not valid
$Var(b) = (X'X)^{-1}X' \underbrace{Var(\epsilon)I} X(X'X)^{-1} = \sigma^2(X'X)^{-1}$ (under homoscedasticity)

→ is always smaller than under heteroscedasticity. To estimate without Assumption 1.4:
$\widehat{Var(b)} = (X'X)^{-1}\sum e_i^2 x_i x_i'(X'X)^{-1} = (\sum x_i x_i')^{-1}\sum e_i^2 x_i x_i'(\sum x_i x_i')^{-1} = \frac{\widehat{Avar(b)}}{n}$

4. Testing for heteroscedasticity:
Idea: start with the linear model: $y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} + \epsilon_i$
H_0: Assumption of homoscedasticity is true: $E(\epsilon_i^2|x) = \sigma^2 \forall i$
If H_0 false, then expected value of ϵ_i^2 given the independent variable can be any funktion of the x_{ij} : $E(\epsilon_i^2)$=f(x)

Breusch-Pagan-Test
f(x): linear function of x
$\epsilon_i^2 = \delta_0 + \delta_1 x_{i1} + ... + \delta_k x_{ik} + v_i$
$H_0 : \delta_1 = \delta_2 = ... = \delta_k = 0 \rightarrow$ F-test OR
$n * R^2_{\epsilon^2} \sim^a \chi^2(k)$ →Langrange Multiplier test

White-Test
f(x): linear function of x, cross product and squares of the independent variables:
$\epsilon_i^2 = \delta_0 + \delta_1 x_{i1} + ... + \delta_k x_{ik} + \delta_{k+1} x_{i1}^2 + ... + \delta_{2k} x_{ik}^2 + \delta_{2k+1} x_{i1} x_{i2} + \delta_{2k+2} x_{i1} x_{i3} + ... + v_i$
→To test more general forms of heteroscedasticity:
H_0: all $\delta_i = 0$ for $i \neq 0$
Test statistic: $R^2_{\epsilon^2} \sim^a \chi^2(k)$

1.3
series lnname=log(name) (name=Variable)

2.1
series predname=eq1.@coefs(1)+eq2.@coefs(2)*lnwage+eq3.@coefs(3)*lnoutput+eq4.@coefs(4)*lncapital

2.2
series residuals=lnlabour-predlnlabour

2.3
Residual plot: y-axis with residuals, x-axis with predicted values
Open as group→View→Graph→XY
click right on graph→Options→Axes/Scales→Zero-line

2.4
Seems to be rather heteroscedastic. From that follows:
Estimates/coefficients: don't change
Std.errors: lower than with robust test
Inference: We reject to easy
R^2: don't change

3.1
series residuals2=$residuals^2$

3.2
equation ls residuals2 c lnwage lnoutput lncapital

3.3
H_0 : we have homoscedasticity
$BP = n * R^2_{\epsilon^2} \sim \chi^2(k)$ (here: k=3)

scalar bp=n*eq2.@r2

3.4
scalar pvalue=qcchisq(bp,3)

4.1
series ln2name=$lnname^2$
series lnname1name2=lnname1*lnname2

4.2
equation ls residuals2 c lnwage lnoutput lncapital ln2wage ln2output ln2capital lnoutputwage lncapitalwage lncapitaloutput

4.3
White=$n * R^2_{\epsilon^2}$
scalar white=n*eq3.@r2

4.4
scalar whitecrit=@qchisq(0.95,9)

→White and Breusch-Pagan both rejecto n the 5% significant level

5.1
View→Estimate equation→Enter the same equation as for OLS→Options→ Heteroscedasticity-consistent→White→C