Technical Appendix: Working with IBES forecast data[1]

Nick Bloom[2], Alexander Klemm[2], Rain Newton-Smith[3], and Gertjan Vlieghe[3]

June 2004

Abstract

This note provides detailed information on the construction of UK firm level panel data sets using IBES analysts' forecasts data. It describes how to read the data into Stata and suggests a procedure for merging the forecast data with company accounts data from Datastream.

Keywords: IBES, Datastream, Investment, Panel Data, Forecast Data

Acknowledgements: This work has been financed by the Bank of England and the programme of research at the ESRC centre for the Micro-Economic Analysis of Fiscal Policy at the IFS. Financial support of the ESRC is gratefully acknowledged.

Correspondence: a.klemm@ifs.org.uk; 7 Ridgmount Street, London WC1E 7AE, UK.

[1] This paper forms a technical appendix to Bond et al (2004), 'The roles of expected profitability, Tobin's Q and cash flow in econometric models of company investment', Bank of England Working Paper no. 222.

[2] Institute for Fiscal Studies.

[3] Bank of England.

Working with IBES data[4]

IBES provide analyst forecast data on monthly basis. It is sold in a number of different ways, so that data may not exactly match the format described in this note. In our data set forecast data are provided on a monthly basis for up to five types of forecasts: A forecast for the current year and one each for the following four years, and finally a long-term growth forecast. Except for the last forecast, all forecasts are actual forecasts of earnings per share. The long-term growth forecast is a forecast of the growth rate in earnings per share over a three to five year horizon. Each month the mean, median, standard deviation and number of forecasts are reported. In the following month any new forecasts and revisions of old forecasts are incorporated. The number of forecasters who have increased or decreased their forecasts is also reported. If there are no new forecasts or revision, then the unchanged data are published.

Apart from the forecast data, a number of auxiliary data files are provided that contain among other things information on the companies, such as their names and their actual past earnings results.

Converting IBES data to Stata format

IBES data are provided on a CD-ROM. They can be transferred to Stata format using the programme "**read_ibes.do**" (see appendix). This programme reads in each of 6 files provided by IBES and saves them in Stata format. It also merges them together and saves the whole set as "ibes.dta". It should be noted that some of the files are very big, so that to read them in, a relatively large memory is needed (RAM of about 150MB). The final dataset (ibes.dta) is not very large (8.5MB), because yearly instead of monthly data are kept. So, once the data have been transferred, they can be used on smaller computers as well.

The first part of the programme is self-explanatory, as all data are simply read in, labelled and transformed into Stata format. The only complication arises during construction of the date variable, which is a string in IBES data, but numeric in Stata. At this stage nothing is dropped or created so that the Stata-files (ibes1.dta to ibes6.dta) correspond exactly to the six IBES files.

Deleting unwanted Data

Having transferred the data to Stata, the next step is to merge all the individual files into one large file containing all data of interest.

Since the datasets are so large, the order of the merging is important. The aim is to delete any unwanted data as early as possible in the programme.
The first two files that are merged are file1, the forecast data and file3, the company descriptions. File 3 contains each company's country of origin, so this immediately allows us to delete a large proportion of the data. For our purposes all non-UK firms

[4] Please note that in order to access these data via the macros provided in this paper, you must have a current subscription agreement for Thomson Financial's DataStream and IBES services. Your access to and use of the data is subject to the terms of that subscription agreement. To the fullest extent permitted by law, Thomson Financial Limited on behalf of itself and its affiliated companies disclaims all liability in relation to the use of the macros and instructions contained in this paper.

can be immediately dropped. The information in file 3 is descriptive, but it can change, e.g. a company may change its name. In that case only the most recent information is kept. Having merged the data and deleted all non-UK data, some variables that contain no information or the same information for each and every observation, (for example "measure" specifies the measure used, which is always "eps" or earnings per share) are dropped.

After saving the data, a list of UK firms (ukfirms.dta) is constructed, which includes a dummy called UK, which is always equal to 1. This list can then be used with each file to make sure that only UK firms are kept: Just merge each file with this list and keep those observations where the UK dummy equals 1.

Then the work on the previously saved data set continues. Company accounts are published with a lag. We only want to use forecasts, which do not contain any information that became available after the accounting period end date. So we drop all current year forecasts that were made after the accounting period end date. Such forecasts are potentially made with some actual results already available, and so are more like outturns than forecasts. To be consistent, we also drop all two-year ahead forecasts that were not made at least two years ahead and so on.
Furthermore only data in pound sterling are kept. This is not really necessary, but given the small number of firms that produce accounts in other currencies, this approach seems the easiest.

Defining a year variable

In order to construct time series data, and to merge the data with time series data from other sources, a year variable needs to be defined. First of all, for our company accounts data, since companies can have accounting years which do not correspond to calendar years, we need to decide which "year" a company's accounts correspond to. In the example file in the appendix, we defined a year as running from the 1st of June to the 31st of May of the following year. This is an arbitrary choice that may not be appropriate for every project. A year could also be defined as the calendar year. But in our case, if a company has an accounting year-end on May 31st 1999, the accounts for this year will be labelled 1998.

Assigning year variables to the IBES forecasts is more complicated. Our approach is to assign to each observation a year variable based on the accounting year end date to which the forecast pertains. This information is provided by IBES and denoted as the variable "periodenddate".

Our procedure is best illustrated with an example. Think of a company whose accounting years end on the 31st of June 1998. During the 12 months up to that date (from July 1st 1997 to June 31st 1998), there can be up to 12 forecasts for "current year earnings".[5] As all of these forecasts pertain to the same accounting year, they are all assigned the year of the accounting year-end date (1998 in our example). During the same 12 months (from July 1st 1997 to June 31st 1998), there will also be up to 12 forecasts for the following accounting year, i.e. the year ending on the 31st of June

[5] There may in fact be more forecasts, as IBES reports forecasts up to the date of the *publication* of accounts, rather than the accounting period end date. We have however dropped such data, as the forecasts may have been affected by information dating after the accounting period end date, e.g. if some information has leaked into the market, before the official publication of results.

1999. These forecasts will then be assigned the year of the accounting year end date less 1, i.e. 1998. This is consistent, as they were made during the same time span as the current year forecasts. These forecasts are known as the “first year ahead” forecasts. The thing to note is that the year of the *date* at which the forecast is *reported* by IBES is not used to define the year variable. Therefore a *current* year forecast made in December 1997, would nevertheless be assigned the year variable 1998, because it pertains to an accounting year which ends in 1998 (or in other words the company’s accounts for the year 1998). Year variables for longer horizons are defined in an equivalent manner. The only exception is the long- term growth forecast, because this forecast does not have an accounting period end date, as this forecast is for three to five years ahead. In this case, we base the “year” on the *current* year forecast period (e.g. the twelve months preceding the company’s accounting period end date). So in the example above, any long-term growth forecasts reported between the 1st of July 1998 and the 31st of June 1998 would be assigned the year 1998.

Choosing which forecasts to use

Having defined a year variable, we are left with the choice of which forecast within a year to keep. The aim is to keep just one (or one summary statistic), so that the company identifier (ticker) and the year variable uniquely define an observation. At this stage there can still be up 12 forecasts per year and forecast horizon. Which of these should be chosen? In order to allow some flexibility at a later stage, it is best to keep more than one possibility. In this programme, we keep the first available forecast, the forecast made six month before the end of the accounting year, and the last forecast. For each of these a variable is created: mean_f for the first available forecast, mean_h for the forecast made six months before the end of the accounting period, and mean_l for the most recent available date. The median forecasts are also kept and labelled in the same manner (i.e. median_f, median_h, median_l). The forecast is multiplied by the number of shares in order to get total earnings rather than earnings per share.

Final steps

Next, we need to merge in any file6, which is the only remaining file containing monthly rather than yearly data. After that we switch to yearly data, by dropping all observation within a year, except the first one.[6]

Then we reshape the data. After reshaping, for each ticker year we will have 6 variables of each type of forecast: e.g. for the first available forecast, we have mean_f0 to mean_f5, where the number indicates the periods forecasted, i.e. 0 is long term growth, 1 is current year, 2 is one year ahead, etc.

These were the most complicated steps. Now file 2 is merged in. This file contains actual earnings as opposed to forecast data. It is provided on a monthly basis, but changes on a yearly basis, as accounting periods are generally yearly. Therefore we define a year variable (in the same way as above) on the accounting year date and collapse the data set before merging it on the ticker and year variables.

[6] Note that we still keep the first, mean and last forecast. These are simply the same over a whole year, because they were defined as the first, mean or last across the year.

Finally file 5 which contains sector information is merged in. For instance, it provides information on which industry a company belongs to. Since this information is time invariant, it is simply merged in on the ticker (company identifier).

As a last step datastream (DS) codes which were provided by IBES, are merged in. We found them to be very unreliable, so they are not actually used and this step could be skipped, unless you want to keep them to check their accuracy. Obviously this also needs to be skipped if the file containing DS codes is missing on your data CD-ROM.

Merging the data

This section is about merging the forecast data with DS company accounts data. For details on how to obtain DS data, please refer to Bloom et al. (2002).

The data are merged using a Stata programme that we called "**q.do**", which again is provided in the appendix. Its output will be a complete panel dataset of company account and analyst forecast data, called **q.dta**.

A main feature of this programme is that it first deals with the data on a company-by-company basis and saves a separate dataset for each company. At the end all of the data are merged together. The reason for proceeding in this manner is that we merge daily and yearly data, which would lead to enormous dataset if we did this for all companies at the same time.

The programme starts by opening a list of all companies for which we have both DS and IBES data. This list of companies was obtained using the 900a DS command. The procedure is described in Bloom et al. (2002). The list is simply a Stat file containing two variables: the DS code and the identifier used by IBES, called "ticker".

In order to save time, we drop any firms from that list for which we do not have DS or IBES data. This is done by merging the list with our complete DS data set and dropping observations for which no DS data are available. The same is done for IBES data.

Using this shortened list, the programme will then merge all the data. It starts with the first company on the list. The difficulty is to merge daily data with yearly data. The way this is being dealt with is to keep daily data only at a small number of dates, which are of interest to us. These dates are the accounting year end dates and the dates on which ibes forecast are made. At the end, for an observation defined by a company identifier and a year, then will then be a number of variables (as opposed to observations) containing data from chosen dates. E.g. we keep market values at four different dates, labelled MV, MV1, MV2, MV3, where MV is the market value on the accounting year end date, MV1 is the market value on date 1 (remember that date 1 is the date when the current year forecast is made, see above), etc. Apart from keeping daily market values on these dates, we also keep averages running up to these dates. The first step is therefore to determine the date at which we wish to keep daily data. This is done by opening the IBES and DS datasets to determine the relevant dates, i.e. the accounting period end dates for each company and the IBES forecast dates.

Once the dates have been determined, the daily market value data are opened, i.e. the file downloaded with the DS **900B** command. Then three averages (over 1 month, one quarter and one year) are calculated at the relevant dates. Apart from the averages also the market value on the accounting date is kept. If there is none, e.g. because of a bank holiday, then the previous market value is kept. While the market value is also present in the yearly accounts data from the **900C** command, keeping it here allow later checks on the merging procedure.

Then these data (i.e. dates, market values and moving averages) are merged into the IBES data set. In the IBES data set we can have up to six different dates on which forecasts are made (the first available forecast, the last forecast etc., see above). We therefore need to merge the data six times, once on the date of the current year forecast, once on the date of the next year forecast, etc. Each time we keep only the market value and the moving averages of market value ending on the date on which the first IBES forecast was made. The data set thus becomes much smaller during this process.

Then the data set is merged with the **900c** data and saved.

This process is then repeated with all companies. There are about 2200 companies for which both IBES and DS data are available.

Once the merging is complete some cosmetic issues are dealt with, such as the labelling of variables.

Then some tests are performed on the merging procedure and doubtful cases are dropped. One of the criteria we use is the company name, which is provided by both IBES and DS. If they are the same, we keep the company. If they are different, we check (using the internet) whether there has been a recent name change. If not, then we conclude that the companies must be different and drop them. This affects only about 10 companies, which we have listed in the file.

Another criterion is the accounting period end date, which we have from three sources: the IBES forecast data, the IBES actual data and DS. We drop any observation for which these three dates differ by more than 30 days.

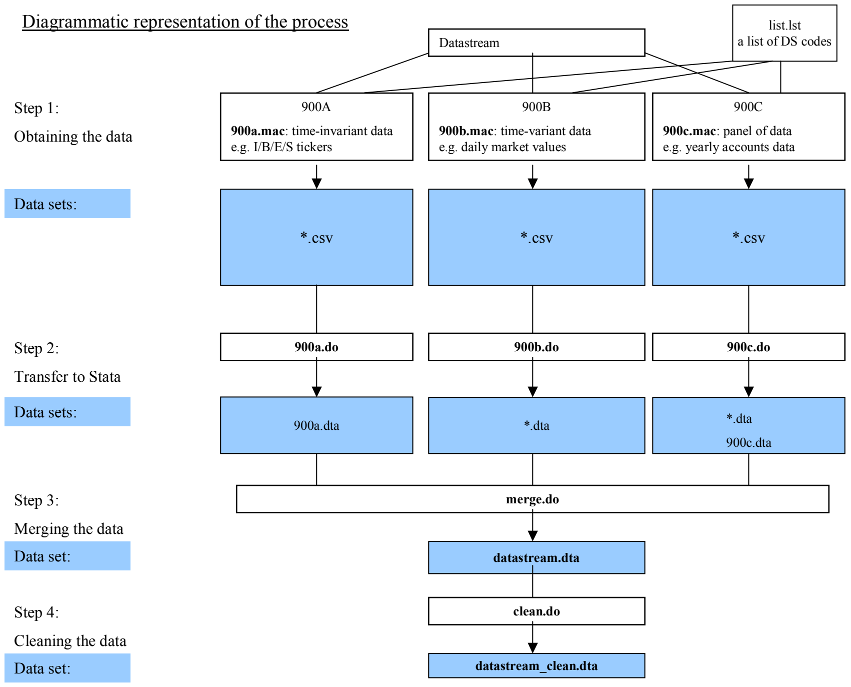
One final check of the merging procedure is to compare the market value that is provided by DS with the one from IBES data (which can be calculated by multiplying the share price by the number of shares). Cases where these two are more than 25% different are listed. Then inspection allows us to decide whether it is more likely that there are differences in the definitions or whether the two market values are from different companies. This is a subjective judgement. The general approach is to assume that if market values from one source are consistently smaller or bigger by a given percentage, then this is probably a different definition. If on the other hand no correlation can be spotted at all, then they are likely to come from different companies. In the latter case they are dropped.

Cleaning the data

We now have a complete set of forecast (IBES) and company accounts (DS) panel data (**q.dta**), which needs to be cleaned. If however you plan to merge DS data with data from other sources, then it is recommended to first merge it, and do the cleaning once the data set is complete.

For details on cleaning, please refer to Bloom et. al. (2002), where general issues and an example cleaning programme are discussed. For convenience, this programme (**clean.do**) is also provided in the appendix to this paper.

This completes the data work. A diagrammatical representation of the steps described is given in the following figure.



Appendix

read_ibes.do

```
cap log close
set more off
log using c:\read_ibes.log,replace
clear
set matsize 800
set memory 500000

cd c:\data\

******************************************************************************
* First part: Getting everything into STATA format*****************
******************************************************************************
#delimit ;
infix
str6 ticker    1-6
str8 period    8-15
str6 measure   17-22
```

```
str3 fiscal    24-26
str6 periodend 28-33
str1 ind       35-35
str1 eflag     37-37
str3 cc        39-41
     numb      43-45
     ups       47-49
     downs     51-53
     median    55-66
     mean      68-79
     se        81-92
     high      94-105
     low       107-118 using hsepssum.eur;

label var ticker    "IBES Ticker";
label var period    "Statistical period";
label var measure   "Measure" ;
label var fiscal    "Fiscal period";
label var periodend "Forecast period end data";
label var ind       "Forecast period indicator";
label var eflag     "Estimate flag";
label var cc        "Currency code";
label var numb      "Number of estimates”;
label var ups       "Number up";
label var downs     "Number down";
label var median    "Median estimate";
label var mean      "Mean estimate" ;
label var se        "Standard deviation";
label var high      "High estimate";
label var low       "Low estimate";
g str4 tempyear = substr(period,1,4);
g str2 tempmonth= substr(period,5,2);
g str2 tempday  = substr(period,7,2);
g tempryear = real(tempyear);
g temprmonth= real(tempmonth);
g temprday  = real(tempday);
g date = mdy(temprmonth, temprday, tempryear);
g str4 tempyear2 = substr(periodend,1,4);
g str2 tempmonth2= substr(periodend,5,2);
g tempryear2 = real(tempyear2);
g temprmonth2= real(tempmonth2);
g temprday2  = 30;
replace temprday2 = 31 if
temprmonth2==1|temprmonth2==3|temprmonth2==5|temprmonth2==7|temprmonth2==8|temprmonth2=
=10|temprmonth2== 12;
replace temprday2 = 28 if temprmonth2==2;
g periodenddate = mdy(temprmonth2, temprday2, tempryear2);
drop temp*;
format date %d;
format periodenddate %d;
replace se =. if se<=-9999999999;
so ticker date;
sa file1,replace;

infix
str6 ticker     1-6
str8 period     8-15
str6 measure    17-22
str1 aflag      24-24
str3 cc         26-28
str6 fy0end     30-35
     fy0eps     37-48
str6 int0d      50-55
     inteps     57-68
     epsgrowth5 70-75
     epsstab5   77-82 using hsepsact.eur,clear;
label var ticker    "IBES Ticker";
label var period     "Statistical period";
label var measure    "Measure";
label var aflag      "Actual flag";
label var cc         "currency code";
label var fy0end     "FY-0 end date";
label var fy0eps     "FY-0 actual EPS";
label var int0d      "Int-0 date";
label var inteps     "Int-0 actual EPS";
label var epsgrowth5 "5 year EPS growth";
label var epsstab5   "5 year EPS stability";
g str4 tempyear  = substr(period,1,4);
g str2 tempmonth = substr(period,5,2);
g str2 tempday   = substr(period,7,2);
g str4 tempyear2 = substr(fy0end,1,4);
g str2 tempmonth2= substr(fy0end,5,2);
g str4 tempyear3 = substr(int0d,1,4);
g str2 tempmonth3= substr(int0d,5,2);
g tempryear = real(tempyear);
g temprmonth= real(tempmonth);
g temprday  = real(tempday);
g tempryear2 = real(tempyear2);
g temprmonth2= real(tempmonth2);
g tempryear3 = real(tempyear3);
g temprmonth3= real(tempmonth3);
g temprday2  = 30;
replace temprday2 = 31 if
temprmonth2==1|temprmonth2==3|temprmonth2==5|temprmonth2==7|temprmonth2==8|temprmonth2=
=10|temprmonth2== 12;
replace temprday2 = 28 if temprmonth2==2;
g temprday3  = 30;
replace temprday3 = 31 if
temprmonth3==1|temprmonth3==3|temprmonth3==5|temprmonth3==7|temprmonth3==8|temprmonth3=
=10|temprmonth3== 12;
replace temprday3 = 28 if temprmonth3==2;
g date = mdy(temprmonth, temprday, tempryear);
g fy0enddate = mdy(temprmonth2, temprday2, tempryear2);
g int0date   = mdy(temprmonth3, temprday3, tempryear3);
format date       %d;
format fy0enddate %d;
format int0date   %d;
drop temp* ;
replace fy0eps = .     if fy0eps    <=-9999999999;
replace inteps = .     if inteps    <=-9999999999;
replace epsgrowth5 = . if epsgrowth5<=-99999;
replace epsstab5 = .   if epsstab5  <=-99999;
so ticker date;
sa file2, replace;

infix
str6  ticker     1-6
str8  cusip      8-15
str8  code       17-24
str32 company    26-57
      dilution   59-64
str1  basic_dil  66-66
str1  p_cflag    68-68
      factor1_10 70-70
str1  instflag   72-72
str1  excode     74-74
str2  cid        76-77
      sector     79-84
str8  start      86-93
str1  cflag      95-95
str24 reserved   97-120 using hsepsid.eur, clear;
label var ticker     "IBES Ticker";
label var cusip      "Cusip";
label var code       "Official ticker";
label var company    "Long company name";
label var dilution   "Dilution factor";
label var basic_dil  "Basic/diluted indicator";
label var p_cflag    "Canadian currency or P/C flag";
label var factor1_10 "1/10 factor";
label var instflag   "Instrument type flag";
label var excode     "Exchange code";
label var cid        "Country ID";
label var sector     "Sector/industry/group code";
label var start      "Start date";
label var cflag      "Company flag";
label var reserved   "Reserved";
g str4 tempyear = substr(start,1,4);
g str2 tempmonth= substr(start,5,2);
g str2 tempday  = substr(start,7,2);
g tempryear = real(tempyear);
g temprmonth= real(tempmonth);
g temprday  = real(tempday);
g startdate = mdy(temprmonth, temprday, tempryear);
drop temp* ;
format startdate %d;
so ticker;
sa file3, replace;

infix
str6 ticker 1-6
     factor 8-20
str8 split  22-29
str8 period 31-38 using hsepsadj.eur, clear;
label var ticker     "IBES Ticker";
label var factor     "Split factor";
label var split      "Split date";
label var period     "IBES Statistical period";
g str4 tempyear = substr(period,1,4);
g str2 tempmonth= substr(period,5,2);
g str2 tempday  = substr(period,7,2);
g tempryear = real(tempyear);
g temprmonth= real(tempmonth);
g temprday  = real(tempday);
g date = mdy(temprmonth, temprday, tempryear);
g str4 tempyear2 = substr(split,1,4);
g str2 tempmonth2= substr(split,5,2);
g str2 tempday2  = substr(split,7,2);
g tempryear2 = real(tempyear2);
g temprmonth2= real(tempmonth2);
g temprday2  = real(tempday2);
g splitdate = mdy(temprmonth2, temprday2, tempryear2);
drop temp*;
format date %d;
format splitdate %d;
so ticker date;
sa file4, replace;

infix
      sector    1-6
str8  secabbr   8-15
str24 secname   17-40
str8  indabbr   42-49
str24 indname   51-74
str8  groupabbr 76-83
str24 groupname 85-108 using hsepssig.eur,clear;
label var sector     "Sector/industry/group code";
label var secabbr    "Sector abbreviation";
label var secname    "Sector name";
label var indabbr    "Industry abbreviation";
label var indname    "Industry name";
label var groupabbr "Group abbreviation";
label var groupname "Group name";
so sector;
sa file5, replace;


infix
str6 ticker   1-6
str8 period   8-15
str3 cc       17-19
     price    20-32
str8 priced   34-41
     shares   43-52
     dividend 54-63 using hsepspan.eur, clear;
#delimit cr
label var ticker   "IBES Ticker"
label var period   "IBES Statistical period"
label var cc       "Currency code"
label var price    "Price"
label var priced   "Pricing date"
label var shares   "Shares outstanding (millions)"
label var dividend "Indicated annual dividend"
g str4 tempyear = substr(period,1,4)
g str2 tempmonth= substr(period,5,2)
g str2 tempday  = substr(period,7,2)
g tempryear = real(tempyear)
g temprmonth= real(tempmonth)
g temprday  = real(tempday)
g date = mdy(temprmonth, temprday, tempryear)
g str4 tempyear2 = substr(priced,1,4)
g str2 tempmonth2= substr(priced,5,2)
g str2 tempday2  = substr(priced,7,2)
g tempryear2 = real(tempyear2)
g temprmonth2= real(tempmonth2)
g temprday2  = real(tempday2)
g pricedate = mdy(temprmonth2, temprday2, tempryear2)
drop temp*
format date %d
format pricedate %d
replace price   =. if price    <=-9999999999
replace shares  =. if shares   <=-99999999
replace dividend=. if dividend <=-99999999
so ticker date
sa file6, replace

infix str6 ticker 1-6 str10 ibesdscode 10-20 in 3/24307 using ds_id.rpt,clear
g dscode = real(ibesdscode)
so ticker
sa dscode, replace

************************************************************************
*Second part: Merging the files into one big file etc**************
************************************************************************


*Start by merging files 1 and 3 (because 3 contains the country code!)
u file3
*just keep the most up to date information:
egen tempmax = max(startdate), by(ticker)
keep if startdate==tempmax
label var startdate "Date of last change"
tab cflag
drop reserved start temp* cflag cusip
so ticker
merge ticker using file1
tab _merge
keep if cid=="EX"
tab p_cflag
tab factor1_10
tab dilution
tab eflag
tab measure
tab ind
g periodind = real(ind)
label var periodind "Forecast period indicator"
label define periodind 0 "Long term growth" 1 "FY 1" 2 "FY2" 3 "FY3" 4 "FY4" 5 "FY5" 6 "Quarter 1" 7
"Quarter 2" 8 "Quarter 3" 9 "Quarter 4"
label values periodind periodind
drop cid _merge periodend period eflag measure fiscal ind factor1_10 dilution p_cflag
label var periodenddate "Forecast period end date"
so ticker date
sa temp,replace

*Aside: construct a list of UK firms
keep if ticker ~= ticker[_n-1]
keep ticker
g UK = 1
so ticker
sa ukfirms,replace

*Reshape so that ticker and year define an observation
u temp, clear
drop high low numb ups downs
keep if periodind <= 5
drop if (date > periodenddate         | date < periodenddate  -  365) & periodind==1
drop if (date > periodenddate - 365   | date < periodenddate - 2*365) & periodind==2
drop if (date > periodenddate - 2*365 | date < periodenddate - 3*365) & periodind==3
drop if (date > periodenddate - 3*365 | date < periodenddate - 4*365) & periodind==4
drop if (date > periodenddate - 4*365 | date < periodenddate - 5*365) & periodind==5

keep if cc == "BPN"
drop cc

*generate a year variable
g year = year(periodenddate) - (periodind-1)
replace year = year-1 if month(periodenddate) <= 5

*determine a year variable for the long term growth forecast (where no periodenddate exists!)
local d = 1
bys ticker periodind periodenddate: g tempD1 = 1 if _n==1 & periodind==1
egen tempD2 = count(tempD1) if periodind==1, by(ticker periodind)
egen tempD = max(tempD2)
local D = tempD[1]

g run = 1 if periodind==1
while `d'<`D' {
bys ticker periodind (periodenddate date): replace run = run[_n-1]+1 if run[_n-1]~=. &
periodenddate~=periodenddate[_n-1] & periodind==1
bys ticker periodind (periodenddate date): replace run = run[_n-1]   if run[_n-1]~=. &
periodenddate==periodenddate[_n-1] & periodind==1
local d = `d'+1 }
local d = 1
while `d'<`D' {
bys ticker: g tempdate`d' = periodenddate if run==`d'
bys ticker: g tempyear`d' = year          if run==`d'
bys ticker (tempdate`d'): replace tempdate`d' = tempdate`d'[_n-1] if tempdate`d'[_n-1]~=.
bys ticker (tempyear`d'): replace tempyear`d' = tempyear`d'[_n-1] if tempyear`d'[_n-1]~=.
so ticker periodind date
replace year = tempyear`d' if periodind==0 & date> tempdate`d'-365 & date<=tempdate`d' & year==.
local d = `d'+1 }
drop if year == .

drop temp* run
so ticker date
sa temp, replace
```

```
*merging in File 6
u file6, clear
keep ticker date shares dividend price
so ticker date
merge ticker date using temp
drop if company==""
tab _merge
drop _merge

*deal with missing data on number of shares
so ticker periodind date
g templastavail = shares
replace templastavail = templastavail[_n-1] if templastavail==. & ticker == ticker[_n-1] &
periodind==periodind[_n-1]
gsort ticker periodind - date
g temprecavail = shares
replace temprecavail  = temprecavail[_n-1]  if temprecavail ==. & ticker == ticker[_n-1] &
periodind==periodind[_n-1]
replace shares = templastavail if shares==. & temprecavail == templastavail & mean ~= .

*generating the earliest available forecast
egen tempfirstdate = min(date), by(ticker year periodind)
generat EF_f   = 10*mean*shares   if tempfirstdate == date & periodind~=0
replace EF_f   = mean/100         if tempfirstdate == date & periodind==0
egen tempmin   = min(EF_f), by(ticker year periodind)
replace EF_f   = tempmin
generat EF_fmed= 10*median*shares if tempfirstdate == date & periodind~=0
replace EF_fmed= median/100       if tempfirstdate == date & periodind==0
egen tempmin2  = min(EF_fmed), by(ticker year periodind)
replace EF_fmed= tempmin2

*the most recent available forecast
egen templastdate = max(date), by(ticker year periodind)
gen      EF_l     = 10*mean*shares   if templastdate == date & periodind~=0
replace  EF_l     = mean/100         if templastdate == date & periodind==0
egen tempminl     = min(EF_l), by(ticker year periodind)
replace EF_l      = tempminl
gen     EF_lmed   = 10*median*shares if templastdate == date & periodind~=0
replace EF_lmed   = median/100       if templastdate == date & periodind==0
egen tempminl2    = min(EF_lmed), by(ticker year periodind)
replace EF_lmed   = tempminl2

*the sixth month forecast
g tempsixdate   = date if abs(month(periodenddate)-month(date))==6 & periodind==1
egen date_h     = min(tempsixdate), by(ticker year)
g EF_h          = 10*mean*shares   if date_h == date & periodind~=0
replace EF_h    = mean/100         if date_h == date & periodind==0
egen tempmin6   = min(EF_h), by(ticker year periodind)
replace EF_h    = tempmin6
g EF_hmed       = 10*median*shares if date_h == date & periodind~=0
replace EF_hmed = median/100       if date_h == date & periodind==0
egen tempmin62  = min(EF_hmed), by(ticker year periodind)
replace EF_hmed = tempmin62

*drop the stuff that we don't need anymore:
drop median mean se temp*

*drop useless observations
bys ticker year periodind (date): g tempcount = _n
keep if tempcount==1

reshape wide date EF_f EF_fmed EF_l EF_lmed EF_h EF_hmed shares dividend price periodenddate,
i(ticker year) j(periodind)
so ticker periodenddate1

***special treatment for accounting year changes that affect the year of mean_f0
replace price0    = . if date0>periodenddate1
replace shares0   = . if date0>periodenddate1
replace dividend0 = . if date0>periodenddate1
replace date0     = . if date0>periodenddate1
replace EF_f0     = . if date0>periodenddate1
replace EF_fmed0  = . if date0>periodenddate1
replace EF_l0     = . if date0>periodenddate1
replace EF_lmed0  = . if date0>periodenddate1
sa temp, replace


*now deal with the actual data on earnings per share (FILE 2)
u file2, clear
merge ticker using ukfirms
bys UK: tab _merge
keep if UK==1
tab aflag
tab measure
collapse (mean) fy0eps epsgrowth5 epsstab5, by(ticker fy0enddate)
g periodenddate1 = fy0enddate
so ticker periodenddate1
merge ticker periodenddate1 using temp
drop if company==""
tab _merge
drop _merge
format periodenddate1 %d
so sector
sa temp, replace

*Merging in file No. 5
u file5, clear
drop indabbr secabbr groupabbr
so sector
merge sector using temp
tab _merge
drop if ticker==""
tab _merge
drop _merge
so ticker
sa temp, replace

*And finally the Datastream codes:
u dscode, clear
merge ticker using temp
drop if company==""
tab _merge
drop _merge
sa ibes, replace

erase temp.dta
```

Q.do

```
cap log close
set more off
clear
set memory 60000
set matsize 800

cd c:\
log using q.log, replace

use 900a.dta
*save time keep only companies in the list for which we have data:
so dscode
merge dscode using 900c.dta
keep if _merge==3
keep dscode ticker
```

```
bys (dscode): drop if dscode == dscode[_n-1]
so ticker
merge using ibes.dta
keep if _merge==3
keep ticker dscode
bys (ticker): drop if ticker == ticker[_n-1]
sa temp,replace


while $c<=$num {
  u temp, clear
  so dscode
  global compcode   = dscode[$c]
  global compticker = ticker[$c]

  *determine the relevant dates:
  u c:\ibes, clear
  keep if ticker == "$compticker"
  ren periodenddate1 date9
  ren date_h date6
  keep date*
  g temp=_n
  reshape long date , i(temp) j(ind)
  keep date
  so date
  keep if date~=date[_n-1] & date~=.
  so date
  sa tempdate, replace
  u 900c, clear
  keep if dscode=="$compcode"
  keep date
  so date
  merge date using tempdate
  drop _merge
  so date
  g dateno = _n
  sa tempdate, replace }

  clear
  cap u stata_files\900b\d$compcode
  cap l in 2
  if _rc == 198 { di "d$compcode - no MV (900b) data available" }
  if _rc == 0 {
  *make sure dscode is a string:
  cap g str6 tempdscode = string(dscode)
  cap g str6 tempdscode = dscode
  drop dscode
  g str6 dscode = tempdscode
  so date
  merge date using tempdate
  gsort - dscode
  replace dscode = dscode[_n-1] if dscode==""
  drop _merge
  g aMV1m = .
  g aMV3m = .
  g aMV1y = .
  egen tempmax = max(dateno)
  local m=tempmax[1]
  local f = 1
  while `f'<=`m' {
  g temp1 = date if dateno==`f'
  egen temp2 = min(temp1)
  gen tempdummyyear   = 1 if date <= temp2 & date > temp2 - 365
  gen tempdummy3month = 1 if date <= temp2 & date > temp2 - 91.5
  gen tempdummymonth  = 1 if date <= temp2 & date > temp2 - 30.5
  egen tempaMV1y = mean(MV), by(tempdummyyear)
  egen tempaMV3m = mean(MV), by(tempdummy3month)
  egen tempaMV1m = mean(MV), by(tempdummymonth)
  replace aMV1m = tempaMV1m if dateno==`f'
  replace aMV3m = tempaMV3m if dateno==`f'
  replace aMV1y = tempaMV1y if dateno==`f'
  drop temp*
  local f = `f'+1 }
  so date
  replace MV = MV[_n-1] if MV==. & date-date[_n-1]<=7
  drop dateno
  sa tempds, replace

  *Date1:
  u c:\ibes, clear
  drop dscode
  rename date_h date6
  keep if ticker == "$compticker"
  so date1
  sa tempibes, replace
  u tempds,clear
  ren MV   MV1
  ren aMV1m aMV1m1
  ren aMV3m aMV3m1
  ren aMV1y aMV1y1
  ren date  date1
  so date1
  merge date1 using tempibes
  drop _merge
  keep if company~=""
  so date2
  sa tempibes, replace
   *Date2
   u tempds,clear
   ren MV    MV2
   ren aMV1m aMV1m2
   ren aMV3m aMV3m2
   ren aMV1y aMV1y2
   ren date  date2
   so date2
   merge date2 using tempibes
   drop _merge
   keep if company~=""
   so date3
   sa tempibes, replace
    *Date3
    u tempds,clear
    ren MV    MV3
    ren aMV1m aMV1m3
    ren aMV3m aMV3m3
    ren aMV1y aMV1y3
    ren date  date3
    so date3
    merge date3 using tempibes
    drop _merge
    keep if company~=""
    so date6
    sa tempibes, replace
     *Date6
     u tempds,clear
     ren MV    MV6
     ren aMV1m aMV1m6
     ren aMV3m aMV3m6
     ren aMV1y aMV1y6
     ren date  date6
     so date6
     merge date6 using tempibes
     drop _merge
     keep if company~=""
```

```
    so date0
    sa tempibes, replace
     *Date0
     u tempds,clear
     ren MV    MV0
     ren aMV1m aMV1m0
     ren aMV3m aMV3m0
     ren aMV1y aMV1y0
     ren date  date0
     so date0
     merge date0 using tempibes
     drop _merge
     keep if company~=""
     so year
     sa temp2, replace
  cap u stata_files\900c\d$compcode
  if _rc==0 {
  *make sure dscode is a strinf:
  cap g str6 tempdscode = string(dscode)
  cap g str6 tempdscode = dscode
  drop dscode
  g str6 dscode = tempdscode
  g year = year(date)
  replace year = year-1 if month(date)<=5
  so year
  merge year using temp2
  keep if _merge==3
  drop _merge
  so date
  sa stata_files\d$compcode,replace
  u tempds,clear
  so date
  merge date using stata_files\d$compcode
  drop _merge
  keep if company~=""
  so year
  sa stata_files\d$compcode,replace
  }
  }
  global c = $c+1 }

************************************************************************************
*This puts all company data sets together
u temp, clear
ren dscode ccc
global num=[_N]
global c=1

while $c<=$num {
  gsort - ccc
  global compcode = ccc[$c]
  cap append using stata_files\d$compcode
  if _rc==0 {di $compcode}
  if _rc~=0 {di "$compcode - not available"}
  global c = $c+1 }

drop if ccc~=""
drop ccc
************************************************************************************

* now just make it a bit nicer:
label var MV        "Market value"
label var MV0       "Market value at date0"
label var MV1       "Market value at date1"
label var MV2       "Market value at date2"
label var MV3       "Market value at date3"
label var MV6       "Market value at 6months before accounting date"
```

```
cap label var MV5   "Market value at date5"
label var aMV1m     "1 month MV moving average"
label var aMV1m0    "1 month MV moving average up to date0"
label var aMV1m1    "1 month MV moving average up to date1"
label var aMV1m2    "1 month MV moving average up to date2"
label var aMV1m3    "1 month MV moving average up to date3"
label var aMV1m6    "1 month MV moving average up to 6months before accounting date"
cap label var aMV1m5    "1 month MV moving average up to date5"
label var aMV3m     "3 months MV moving average"
label var aMV3m0    "3 months MV moving average up to date0"
label var aMV3m1    "3 months MV moving average up to date1"
label var aMV3m2    "3 months MV moving average up to date2"
label var aMV3m3    "3 months MV moving average up to date3"
label var aMV3m6    "3 months MV moving average up to 6months before accounting date"
cap label var aMV3m5    "3 months MV moving average up to date5"
label var aMV1y     "1 year MV moving average"
label var aMV1y0    "1 year MV moving average up to date0"
label var aMV1y1    "1 year MV moving average up to date1"
label var aMV1y2    "1 year MV moving average up to date2"
label var aMV1y3    "1 year MV moving average up to date3"
label var aMV1y6    "1 year MV moving average up to 6months before accounting date"
cap label var aMV1y5    "1 year MV moving average up to date5"
label var price0    "Share price at date0"
label var price1    "Share price at date1"
label var price2    "Share price at date2"
label var price3    "Share price at date3"
label var price4    "Share price at date4"
label var price5    "Share price at date5"
label var shares0   "No. of shares at date0"
label var shares1   "No. of shares at date1"
label var shares2   "No. of shares at date2"
label var shares3   "No. of shares at date3"
label var shares4   "No. of shares at date4"
label var shares5   "No. of shares at date5"
label var dividend0 "Dividends at date0"
label var dividend1 "Dividends at date1"
label var dividend2 "Dividends at date2"
label var dividend3 "Dividends at date3"
label var dividend4 "Dividends at date4"
label var dividend5 "Dividends at date5"
label var EF_f0   "First LTG forecast"
label var EF_f1   "First current year forecast"
label var EF_f2   "First 1 year ahead forecast"
label var EF_f3   "First 2 year ahead forecast"
label var EF_f4   "First 3 year ahead forecast"
label var EF_f5   "First 4 year ahead forecast"
label var EF_l0   "Last LTG forecast"
label var EF_l1   "Last current year forecast"
label var EF_l2   "Last 1 year ahead forecast"
label var EF_l3   "Last 2 year ahead forecast"
label var EF_l4   "Last 3 year ahead forecast"
label var EF_l5   "Last 4 year ahead forecast"
label var EF_h0   "6 month ahead LTG forecast"
label var EF_h1   "6 month ahead current year forecast"
label var EF_h2   "6 month ahead 1 year ahead forecast"
label var EF_h3   "6 month ahead 2 year ahead forecast"
label var EF_h4   "6 month ahead 3 year ahead forecast"
label var EF_h5   "6 month ahead 4 year ahead forecast"
label var EF_fmed0 "First LTG forecast"
label var EF_fmed1 "First current year forecast"
label var EF_fmed2 "First 1 year ahead forecast"
label var EF_fmed3 "First 2 year ahead forecast"
label var EF_fmed4 "First 3 year ahead forecast"
label var EF_fmed5 "First 4 year ahead forecast"
label var EF_lmed0 "Last LTG forecast"
label var EF_lmed1 "Last current year forecast"
label var EF_lmed2 "Last 1 year ahead forecast"
label var EF_lmed3 "Last 2 year ahead forecast"
label var EF_lmed4 "Last 3 year ahead forecast"
label var EF_lmed5 "Last 4 year ahead forecast"
label var EF_hmed0 "6 month ahead LTG forecast"
label var EF_hmed1 "6 month ahead current year forecast"
label var EF_hmed2 "6 month ahead 1 year ahead forecast"
label var EF_hmed3 "6 month ahead 2 year ahead forecast"
label var EF_hmed4 "6 month ahead 3 year ahead forecast"
label var EF_hmed5 "6 month ahead 4 year ahead forecast"
label var date0     "Date of EF_f0"
label var date1     "Date of EF_f1"
label var date2     "Date of EF_f2"
label var date3     "Date of EF_f3"
label var date4     "Date of EF_f4"
label var date5     "Date of EF_f5"
label var ds104     "TOTAL SALES                "
label var ds160     "CORPORATION TAX                "
label var ds214     "EMPLOYEE REMUNERATN (DOMESTIC)"
label var ds315     "MINORITY INTERESTS            "
label var ds336     "PLANT & MACHINERY-DEPN            "
label var ds364     "TOTAL STOCK AND W.I.P.             "
label var ds423     "SALES OF FIXED ASSETS              "
label var ds1026    "NET PAYMNT FOR FIXED ASSETS          "
label var ds117     "TOTAL EMPLOYMENT COSTS           "
label var ds164     "IRRECOVERABLE A.C.T.          "
label var ds215     "TOTAL EMPLOYEE REMUNERATN              "
label var ds321     "TOTAL LOAN CAPITAL            "
label var ds337     "OTH FIXED ASSETS-DEPN        "
label var ds365     "FINISHED GOODS(O/W)          "
label var ds429     "EQUITY & PREFERENCE ISSUES    "
label var ds1035    "PAYMENTS: SUBS ETC.                    "
label var ds119     "RESEARCH AND DEVT.          "
label var ds166     "TOTAL DOMESTIC TAX           "
label var ds216     "NO. DOMESTIC EMPL. (UNITS)    "
label var ds324     "LEASED ASSETS-GROSS(O/W)       "
label var ds338     "TOT FIXED ASSETS-DEPN              "
label var ds375     "TOTAL CASH & EQUIVALENT          "
label var ds431     "FIXED ASSETS PURCHASED             "
label var ds1036    "RECEIPTS: SUBS ETC.          "
label var ds135     "PROFIT BEFORE PROVS (ADJ)       "
label var ds169     "TOTAL OVERSEAS TAX           "
label var ds219     "TOTAL NO. OF EMPL. (UNITS)      "
label var ds327     "TOTAL LAND & BLDGS-GROSS       "
label var ds339     "TOT FIXED ASSETS-NET          "
label var ds376     "TOTAL CURRENT ASSETS          "
label var ds432     "FIXED ASSETS OF NEW SUBS.     "
label var ds1037    "NET PAYMENTS: SUBS ETC.       "
label var ds136     "DEPRECIATION                       "
label var ds172     "TOTAL TAX CHARGE-ADJ               "
label var ds275     "BANK BORROWING > 1YR(O/W)      "
label var ds328     "PLANT & MACHINERY-GROSS        "
label var ds342     "RESEARCH & DEVMT.             "
label var ds387     "BANK BORROWING < 1YR(O/W)      "
label var ds435     "TOTAL NEW FIXED ASSETS             "
label var ds1038    "NET PYMNT: LOANS & ADVANCES          "
label var ds137     "OPERATING PROFIT-ADJ               "
label var ds175     "AFTER TAX PROFIT-ADJ               "
label var ds305     "EQUITY CAP. AND RESERVES       "
label var ds329     "OTH FIXED ASSETS -GROSS       "
label var ds344     "TOTAL INTANGIBLES             "
label var ds389     "TOTAL CURRENT LIABLITIES         "
label var ds436     "RESEARCH & DEV.EXP.          "
label var ds1045    "CASH INFLOW FROM FINANCING    "
label var ds143     "INTEREST INCOME                 "
label var ds181     "PREFERENCE DIVIDEND FOR PERIOD"
label var ds307     "TOT. SHARE CAPITAL & RESERVES"
label var ds330     "TOT FIXED ASSETS -GROSS         "
label var ds356    "TOTAL INVESTMENTS (INC.ASS.)           "
label var ds390    "NET CURRENT ASSETS        "
label var ds479    "FIXED ASSETS (SUBS)        "
label var dsMV        "MV"
label var ds153    "TOTAL INTEREST CHARGES"
label var ds182    "EARNED FOR ORDINARY-FULL TAX"
label var ds309    "BORROWINGS REPAYABLE < 1 YEAR"
label var ds331    "LEASED ASSETS-DEPN (O/W)"
label var ds359    "OTHER ASSETS"
label var ds406    "TOTAL EQUITY ISSUED"
label var ds666    "CONSTRUCTN IN PROGRESS"
label var ds155    "PRE-TAX PROFIT EXC ASSOCS-ADJ"
label var ds183    "NET E.P.S. FULL TAX"
label var ds312    "TOTAL DEFERRED & FUTURE TAX"
label var ds332    "LEASED ASSETS-NET (O/W)   "
label var ds360    "STOCKS"
label var ds412    "EQUITY ISSUED FOR CASH"
label var ds1024    "PAYMENTS: FIXED ASSETS"
label var ds156    "ASSOC. PRE-TAX PROFITS"
label var ds187    "ORDINARY DIVIDENDS"
label var ds313    "TOTAL LT PROVN EXC DEF TAX"
label var ds335    "TOTAL LAND & BLDG-DEPN"
label var ds361    "WORK IN PROGRESS"
label var ds418    "CHANGE IN LOAN CAPITAL"
label var ds1025    "RECEIPTS: FIXED ASSETS"
label var ds113    "WAGES AND SALARIES"
label var ds993    "OPERATING PROFIT"
label var ds981    "ADJ TO OPERATING PROFIT"
ren company name_ibes
ren name name_ds
so dscode date
cap drop temp*

*******************some checks on the merging procedure (names, MV...)
*Names
format name_ds %1s
format name_ibes %1s
g str80 tempibes_lower = lower(name_ibes)
g str80 tempds_lower = lower(name_ds)
g str3 tempibes_abr = substr(tempibes_lower,1,3)
g str3 tempds_abr = substr(tempds_lower,1,3)
so ticker
l ticker name_ibes name_ds if tempibes_abr~=tempds_abr & ticker~=ticker[_n-1]
*need to go through that list. If names really different, check e.g. using the internet.
*The following are not obviously the same:
*@1SM  TARPAN PLC  I2S                              (renamed)
*@ARP  ARLEN  Fortress Holdings                     (renamed)
*@BL2  LANGLEY & JOHNSON GROUP  Medi@Invest             (renamed)
*@BOM  BOASSE MASSIMI  BUTLERCOX                      different?
*@BPH  BREDORO  BOASE MASSIMI POLLITT                  different?
*@BVO  BEAVERCO  Headway                            (renamed)
*@CCX  MAGELLAN INDS  CELESTION INDS                  different?
*@CDG  CRATON LODGE  Princedale Group                 (renamed)
*@CUZ  CHILLINGTON CORP  Plantation & General Inv       (renamed)
*@CWQ  CARBO  COPSON,F.                               ?
*@EPR  EASTERN PRODUCE  Linton Park                   (renamed)
*@GGL  GOLD GREENLEES  The GGT Group                  (renamed)
*@GHJ  IZODIA  Infobank Intl. Holdings               (renamed)
*@HGP  HENDERSON GP.  HAMPTON TRUST PLC                different?
*@HJI  HAEMOCELL  Surgical Innovations Grp            (renamed (merged))
*@HLN  HOLMES & MARCHT  Huntsworth                    (renamed)
*@KJU  KYNOCH GROUP PLC  Bioquell                     (renamed)
*@KLB  KLEINWORT BENSON  KELT ENERGY                   different ?
*@LCO  LEE COOPER  VIVAT HOLDINGS                     (renamed)
*@LFJ  LILLEY  LINREAD                                ?
*@LOV  YJL PLC  Montpellier Group                    (renamed)
*@MVT  MICROVITEC  Ultima Networks                     (renamed)
```

```
*@MYS  MYSON  ANGLO NORDIC                                ?
*@PMG  PALMA GROUP  Pex                           (renamed)
*@PNP  PROSPECTIVE GP  PINEAPPLE GROUP                    ?
*@RJB  RJB MINING  UK Coal                        (renamed)
*@S4Q  SHORCO GROUP  Peterhouse Group                     (same (one part of the other))
*@T5M  TAMARIS PLC  World Trade Systems                   (renamed)
drop if ticker == "@BOM" | ticker=="@BPH" | ticker=="@CCX" | ticker=="@CWQ"
drop if ticker == "@HGP" | ticker=="@KLB" | ticker=="@LFJ" | ticker=="@MYS" | ticker=="@PNP"
drop temp*

*date
g temptest = abs(fy0enddate-date)
drop if temptest>31 & temptest~=.
g temptest1 = abs(periodenddate1-date)
drop if temptest1>31 & temptest1~=.
drop temptest

*MV
g MVibes = price1*shares1
g tempratiomv = 100*MV1/MVibes
egen templast = max(date), by(ticker)
egen tempnoj = count(date), by(ticker)
g temptest = 1 if [(tempratiomv<0.75 & date==templast) | (tempratiomv>1.25 & tempratiomv~=. &
date==templast)]&tempnoj>=4
egen temptest2 = min(temptest), by(ticker)
so ticker date
l ticker MVibes MV1 dsMV tempratiomv if temptest2==1, noo
*go through this list. if likely to be typo: keep. Otherwise probably different companies, so drop.
drop if ticker == "@BLX"|ticker =="@CLD"|ticker =="@DZZ"|ticker =="@TEU"
so dscode date
sa q, replace

set more on
log close

erase tempdate.dta
erase tempibes.dta
erase tempds.dta
erase temp.dta
erase temp2.dta
```

clean.do

```
cap log close
set more off
clear
set memory 60000
set matsize 800


****************Programmes that are used in this file:
* This programme generates a different series number for each new run of consecutive data
* The dscode is replaced by dscode * 10 plus series number
* Note this programme needs to be run before any command that refers to a previous date (i.e. [_n-1])
cap prog drop series
prog def series
  cap gen str6 odscode=dscode
  cap label var odscode "Original DScode"
  bys odscode (date): g tempdays = date - date[_n-1]
  ge ok = 0
  replace ok = 1 if (tempdays>395|tempdays<335)
  ge series = 1
  replace series = 2 if ok==1
  bys odscode (date): replace series = 2  if series[_n-1]==2
  local x = 3
  while `x' <= 15  {
    by odscode (date): replace series = `x' if series[_n-1]==`x'-1 & ok==1
    by odscode (date): replace series = `x' if series[_n-1]==`x'
    local x = `x'+1 }
  g tempdscode      = real(odscode)*10 +series if series<=9
  replace tempdscode= real(odscode)*100+series if series>=10
  replace dscode = string(tempdscode)
  drop series tempdays ok tempdscode
end

cd c:\
log using clean.log, replace

*specify dataset to be cleaned:
u datastream,clear
*u 900c, clear

*************determine length of accounting year
*this needs to be done before any observation is dropped, otherwise don't get correct length of
accounting years
bys dscode (date): g days = date - date[_n-1]
replace days = 365 if days==.
label var days "Days in accounting year"
drop if days>395|days<335

******************dropping now if lacking core data
drop if ds104==. | ds104==0
gen cash=(ds182+ds136)
drop if cash==.

*****************Trimming:
*the following deals with unbelievable changes (up by g% or down by -1/(1+g) %)
*use only with core variables (because each time used you lose observations)

cap prog drop trimdrop
prog def trimdrop
  qui series
  while "${_1}"~="" {
    bys dscode (date):gen temp_${_1}=(${_1}-${_1}[_n-1])/${_1}[_n-1]
    su temp_${_1}, de
    drop if (temp_${_1}>=$g/100 | temp_${_1}<=-($g/100)/(1+($g/100))) & temp_${_1}~=. & ${_1}~=0
    drop temp*
    macro shift }
end

global g=200
trimdrop ds104 ds219 ds339

********make sure we have at least four observations per firm (or virtual firm)
qui series
cap drop noj
egen noj=count(date), by(dscode)
drop if noj<4

**********dealing with the year variable
cap g year = year(date)
*if _rc==0 {replace year = year-1 if month(date)<=5}

*sometimes we have two observations in a year, because accounts change from the 31/5 to the 1/6.
generat temp = 0.03 if month(date)==5
replace temp = 10   if month(date)==6
egen tempprob = sum(temp), by(dscode)
replace year = year -1 if month(date)==6 & tempprob>10 & tempprob-int(tempprob)~=0
*test whether this worked (if nothing listed, successful):
so dscode year
l dscode year date if (year==year[_n-1] & dscode==dscode[_n-1]) | (year==year[_n+1] &
dscode==dscode[_n+1])

/* this for year from 1 january to 31 december
*sometimes we have two observations in a year, because accounts change from the 31/5 to the 1/6.
generat temp = 0.03 if month(date)==12
replace temp = 10   if month(date)==1
egen tempprob = sum(temp), by(dscode)
replace year = year -1 if month(date)==1 & tempprob>10 & tempprob-int(tempprob)~=0
*test whether this worked (if nothing listed, successful):
so dscode year
l dscode year date if (year==year[_n-1] & dscode==dscode[_n-1]) | (year==year[_n+1] &
dscode==dscode[_n+1])      */

drop temp*
sa datastream_clean, replace

log close
set more on
```