3rd set SAS assignments

Data description for the three data sets `rwetrad` (RWE), `tuitrad` (TUI) and `sietrad` (SIEMENS):

Variable	Label	Explanation
`event_d`	Date of Trading	self explanatory
`event_t`	Time of Trading	self explanatory
`event_dt`	Date and Time of Transaction	self explanatory
`vol`	Number Securities Traded	self explanatory
`price`	Security Price DM	Transaction price in Euro
`bidprice`	Last recorded Bid Price	Prevailing bid price prior to the trade
`askprice`	Last recorded Ask Price	Prevailing ask price prior to the trade
`in_class`	Buy/Sell Indicator (Quote)	Indicator variable taking the value 1 if trade was a buy and -1 if trade was a sell
`midpoint5`	Last recorded BA Midpoint 5 minutes after the trade	Prevailing Bid/Ask midpoint 5 minutes after the trade

1. Working with ultra high frequency data

i) Choose one data set and compute the Bid/Ask midpoint as $(bidprice + askprice)/2$, the effective spread ES, the realized spread RS and the price impact PI where

$$ES = \begin{cases} 2 \cdot (price - midpoint) & \text{if} \quad \text{buy} \\ 2 \cdot (midpoint - price) & \text{if} \quad \text{sell} \end{cases}$$

$$RS = \begin{cases} 2 \cdot (price - midpoint5) & \text{if} \quad \text{buy} \\ 2 \cdot (midpoint5 - price) & \text{if} \quad \text{sell} \end{cases}$$

$$PI = (ES - RS)/2$$

Provide informative labels for the newly created spread variables. Hint: Use a *if* statement in a data step (see SAS help!).

ii) The trading day starts at 9:00 and ends at 17:30. Dividing the trading day into 10 minutes intervals yields 51 intervals per day. Create with SAS two variables: an interval variable `inter1` and a count interval variable `c_int1`.

Interval	`inter1`	`c_int1`
$[9:00;9:10[$	9:00	1
$[9:10;9:20[$	9:10	2
	⋮	
$[17:10;17:20[$	17:10	50
$[17:20;17:30]$	17:20	51

Use the macro in the help file on the course homepage. Try to figure out what the macro actually does. Call the macro `xtenmin` in a data step.

iii) Write a macro (name it `datasampling`) that has the stock ticker (RWE, SIE, TUI) as argument. The content of the macro basically consists of a data step where the spread variables are computed in i) and the ten minute indicator creation in ii). Call the macro for every stock.

iv) Choose again one data set. Call a sort procedure that sorts by `inter1`. Calculate means together with 90% confidence bands for the variables ES, RS and PI for each ten minute interval of the trading day across all trading days (Use `PROC MEANS` with the option *clm* and then *lclm* and *uclm* in the output statement (SAS help!)). Write the means to a new data set, label the means and the confidence bands, and create a variable with the stock ID in this data set (e.g. ticker="`RWE`").

v) Plot the means and the confidence bands for each ten minute interval against time of day in an appealing manner. (If you put `i=join` in the goptions statement then the points will be connected by straight lines.)

vi) Write another macro (name it `plotmeans`) which sorts the data, compute means and plots the intra-day ten minute means with confidence intervals for one variable (ES, RS, PI) as done in iv) and v) with the stock, the variable and the picture output path as arguments. For keeping the axis values for the ES, RS and PI flexible you can use in the macro the if statement. An example for the effective spread.

```
%IF &var. = es %THEN %DO;
%LET name = Effective Spread;
%LET min=0;
%LET max=0.05;
%LET step=0.01;
%END;
```

which would be used in the goptions as:

⋮

```
axis2 label = (h=2 angle=90 ’Euro’) ... order=(&min. to &max. by &step.);
title h=2 ’’Diurnal Pattern of &name. (%UPCASE(&stock.))’’;
```

vii) Write another macro (name it `allvariablesforonestock`) which calls the previous two macros in the following fashion. The macro `datasampling` is called once and the macro `plotmeans` is called three times (for each variable one time). The two arguments of the macro `allvariablesforonestock` are output path and stock ticker.

/*Additional Task*/

viii) Write another macro (name it `allstocksallvariables(path,stocks,numstock);`) which calls the macro in vii) for each stock in a do loop. In order to loop over a string variable you have to use a little trick. The arguments of the macro should be the output path and a string variable containing the stock ticker symbols delimited by a space and the number of stocks.

In the macro, you loop over this string variable:

```
%DO i=1 %TO &numstock.;
%let bufstock=%trim(%scan(&stocks,&i,” ”));
```

call the macro *allvariablesforonestock* where bufstock now contains the stock ticker symbol (try to find out why)

```
%END;
```

Before you call this last macro, create a string variable:

```
%LET stocks=”rwe sie tui”; %LET numstock=”3”;
```