

PROJECT TEMPLATE LOSS DEVELOPMENT: VBA MACROS

(The attached PDF file has better formatting.)

Jacob: Why do you include VBA macros in the project templates? Isn't the student project to test if we can code the formulas ourselves?

Rachel: Just the opposite. The student projects focus on regression analysis and time series modeling. No statistical software (besides Excel) is needed.

Statisticians use statistical software packages that form the standard graphs and charts, such as residual plots for regression analysis and correlograms for time series analysis.

Excel has the basic regression built-in functions, such as *LINEST* and *TREND*, and it has many basic functions, such as *RAND* and *SUMPRODUCT*, from which we construct statistical procedures, such as Monte Carlo simulation and sample autocorrelation functions. Excel's formulas and macros suffice for all the statistical procedures in the on-line courses.

Not all candidates have the needed proficiency to use Excel's macros and formulas.

For a regression analysis student project, you should spend most of your time deciding what hypotheses to test and what the statistical results imply. The project template on loss reserving teaches you how to interpret residual plots. You should not have to spend hours forming the residual plots.

The *REGRESSION* add-in forms residual plots, but they are not the type needed to test parameter stability or data heteroscedasticity. Most statistical software packages show the needed residual plots.

We provide illustrative worksheets with cell formulas and complete documentation so you can produce residual plots. The cell formulas are general, so you are not restricted to the sample sizes in the illustrative worksheets.

Illustration: The illustrative worksheets use a 15×15 year loss triangle. The cell formulas can be used with any $N \times N$ year loss triangle.

Forming the data samples by simulation, selecting the trend parameters and the variance of the error term, ... are described in the illustrative worksheets and the associated PDF files. The Excel formulas do not take long to code if you copy and paste from the illustrative worksheets.

Candidates who are not particularly fluent in Excel may find it easier to stick with a 15×15 year array than to adapt the project template to other data samples. Some candidates will have trouble with Excel functions, such as forming the residual plots or the residual matrix.

The VBA macros automate the time-consuming, repetitive, Excel coding. They do not do the simulation or interpret the results.

USING THE VBA MACROS ON THE ILLUSTRATIVE WORKSHEETS (SUMMARIES)

The illustrative worksheets have VBA macros to handle routine data manipulation and cell formatting. The macros are fully documented so you can adapt them to your project.

The macros do not substitute for understanding the statistical techniques. They eliminate repetitive coding of cell formulas and errors in Excel syntax, but you must choose proper hypotheses, significance levels, and tests.

Take heed: Your write-up explains what your student project does, including the formatting and data processing in the macros. Be sure you understand what the macros do, so you can interpret the residual plots.

Illustration: The project templates on parameter stability and heteroscedasticity require you to form and interpret residual plots. Excel does not have a built-in function to create the residual plots needed for these student projects.

- Residual plots from the *REGRESSION* add-in do not show means or standard deviations.
- Forming the residual plots may take several hours if you are not expert in Excel.
- The student project deals with statistical techniques, not Excel expertise.
- Statistical software packages have built-in residual plots. No statistician codes these charts from scratch.
- The VBA macros in the illustrative workbook calculate means and standard deviations by each independent variable, and they form four residual plots.
- Your write-up should explain which residual plots you use, what these graphs show, and the implications of the graphs for parameter stability and heteroscedasticity.

The VBA macros are extensively documented. Candidates who use VBA may revise the macros to better suit their student projects.

Illustration: You may change the type of chart, the location of the chart on the worksheet, or the formatting of the cells by adapting the VBA statements in the macros.

Take heed: Knowledge of VBA is not needed for the student project. But a good knowledge of Excel and VBA helps all actuarial work, including the student project.

For efficient completion of the student project, first form the output yourself. You can form each illustrative worksheet using cell formulas and built-in functions.

Replicate the illustrative worksheets using the project template instructions so that you understand the relations among the cells. You learn to simulate data samples of various types, run regressions, form residual output and residual plots, and test hypotheses.

Change the values of the named cells one by one. After each change, re-calculate the worksheets and verify the changes in the cell values.

- Start with the easier parameters, such as the intercept α .
- Then vary more complex parameters, such as the change in the inflation rate.
- Finally, add another year to the regression and change the size of the residual matrix.

The preliminary work ensures that you understand what the student project does. Now decide on the type of hypothesis you want to examine, such as

- A 16 year triangle with inflation of 6% for the first 8 years and 16% for the next 8 years.
- A 20 year triangle with development period geometric decay of 20% for the first 10 years and then declining linearly to 10% over the next 10 years.

Now read the documentation for the VBA macro that forms the set of accident year by development period indices.

Copy the illustrative worksheet for stable rates to your workbook. This worksheet has a 15 × 15 loss triangle. Choose the number of years for your student project, such as 20 × 20. Use the VBA macro to replace the development period and calendar year indices.

Take heed: If you are not a heavy Excel user, the cell formulas and macros seem difficult at first. These macros are geared to you, not to Excel experts. Once you understand the gist of the project template, you can modify parameters and run simulations quickly.

FORM THE RESIDUAL MATRIX

The *REGRESSION* add-in produces a table of residuals sorted by observation. You convert the table of residuals to a matrix with dimensions of development period and calendar year.

Excel has no simple way of forming the residual matrix. One illustrative worksheet shows how to use Excel's *INDEX* and *MATCH* functions to create the residual matrix.

Take heed: The Excel built-in functions automate much of the work. But even with the illustrative worksheets as guides, you must spend time creating the matrices and graphs. The time spent on Excel coding is a dis-incentive to running more simulations with varying parameter values.

The VBA macro assumes the residual output comes from the *REGRESSION* add-in.

- It searches for the cell value RESIDUAL OUTPUT.
- It moves down two rows to the header line of the residual output.
- It forms a range of the residual output that is bounded by blank rows or columns.
 - This range has N+1 rows: N observations and one header row.
 - The range may have 3, 4, 5, or 6 columns.
 - The *REGRESSION* add-in gives 3 or 4 columns, depending on whether you ask for standardized residuals.
 - You may have added columns of development period and calendar year indices.

Take heed: VBA reads and writes to Excel worksheets using variant arrays.

- The variant array is declared *As Variant*.
- The name of the array does not have parentheses.
- After reading from the Excel worksheet, it has two dimensions (rows × columns).
- The array values are variants. The first row of this array has text (column captions), and the other rows have figures.

The macro asks if the observations are in accident year by development period order.

- If the observations are in accident year by development period order, you don't have to copy development period and calendar year indices and append them to the right of the residual output. The VBA macro that forms the simulation sample uses accident year by development period order.
- The macro checks that the number of observations is $\frac{1}{2} \times N \times (N+1)$ for an integral N . If it is not, the observations do not come from an $N \times N$ loss triangle. The macro issues a message and ends.

DETAILED INSTRUCTIONS: VBA MACRO FOR RESIDUAL PLOTS

- The VBA macro determines the size of the matrix by the number of observations.
- It assigns the residuals to calendar year by development period combinations.
- It writes a matrix of residuals several columns to the right on the same worksheet.

The procedure ResidualOutput finds the words Residual Output on the work-sheet.

The *REGRESSION* add-in begins the residual output two rows down from the cell with the words *Residual Output*. The macro descends two rows (in the same column) and selects the current region around this cell.

- The current region is the range bordered by blank rows and columns on all sides.
- The current region has the residuals in either three or four columns with a row of column headers on top. The fourth column is the standardized residuals (if you request them).

The macro reads the current region into a VBA variant array called VariantData.

- We don't know how many columns or rows the current region has.
- VariantData is declared as a variant, not as an array. It becomes an array with two dimensions, the same size as the current region.
 - The first dimension is the rows in current region.
 - The second dimension is the columns in current region.

VariantData is a one-based array, even if your macro default to zero-based arrays. If your simulation has 120 observations, VariantData has 121 values in the first dimension. The first row of values is the column headers.

- The elements of VariantData are figures or strings (column headers, residuals, etc.).
- The number of observations (RowTotal) = the upper bound of the first dimension of VariantData – 1 (subtract the row of column headers).

If the observations are observed values in an $N \times N$ loss triangle, the macro determines N by the number of observations K .

- If the number of years is Z , the number of observations is $\frac{1}{2} \times Z \times (Z+1)$.
- If the number of observations is T , the number of years is $\frac{1}{2} \times ((1 + 8T)^{\frac{1}{2}} - 1)$. (Verify that this is the solution to a quadratic equation in one unknown.)

$$\begin{aligned}\frac{1}{2} \times N \times (N+1) &= K \\ \Rightarrow N^2 + N - 2K &= 0 \\ \Rightarrow N &= (-1 + (1 + 8K)^{\frac{1}{2}}) / 2\end{aligned}$$

The number of years is assigned to the variable NoYrs.

Take heed: The simulations in the illustrative worksheets use $N \times N$ loss triangles. Actual data may be in different formats, though an $N \times N$ loss triangle is most common.

The macro verifies that the number of observations is consistent with an $N \times N$ triangle.

Illustration: If the residuals are 115 observations, they are not from an $N \times N$ triangle.

NoYrs is declared as an integer, so any fractions (decimals) are discarded.

- If the number of observations equals $\frac{1}{2} \times \text{NoYrs} \times (\text{NoYrs} + 1)$, all is well.
- If not, the macro terminates with an error message.

Jacob: I used an $N \times N$ loss triangle and I still got this error message. What happened?

Rachel: You probably left out some data points (or had too many data points) in the original simulation. If you use N years, check that you have $\frac{1}{2} \times N \times (N+1)$ observations.

RESIDUAL OUTPUT TO RESIDUAL MATRIX

The macro creates a matrix called *Residuals* to hold the residuals. If the number of years is Z , the dimensions of this matrix are (0 to $Z-1$, 0 to $Z-1$).

- Residuals is declared as a variant array with two dimensions.
- Residuals is re-dimensioned as (0 to $Z-1$, 0 to $Z-1$).
- This is a *zero based array* with Z elements in each dimension.
- We use a zero based array to be consistent with the indices for each dimension.

The procedure called ResidualArray fills this array with figures. The macro has two options:

- The observations are in accident year by development period order. Code *TRUE* as the optional parameter: *RESIDUALARRAY (TRUE)*
- The order of the observations is not specified. Code *FALSE* as the optional parameter: *RESIDUALARRAY (FALSE)*

The observations in the illustrative worksheet are in accident year by development period order. You may simulate observations in a different order; the macro works the same.

Jacob: Where do we code this parameter of *TRUE* or *FALSE*?

Rachel: Open the visual basic editor, using commands on the menu bar or Alt-F11.

- The macro runs the procedures after the declarations.
- The second procedure has this parameter.

Jacob: Why not create a dialogue (user form) asking the user if the observations are in accident year by development period order?

Rachel: We want candidates to understand how the macro works, so they can modify it for their student projects. The macros do the tedious work; you select all options.

Take heed: The macros are intended for actuarial candidates, who can use the visual basic editor. If you have never edited a macro, learn to do so.

Jacob: What does *accident year by development period order* mean?

Rachel: Let the number of years = Z and observations are in accident year by development period order.

We know the accident year, development period, and calendar year of each observation.

- The first Z observations are accident year = 0, development period = 0 to $Z-1$, and calendar year = accident year + development period (0 to $Z-1$).
- The next $Z-1$ observations are accident year = 1, development period = 0 to $Z-2$, and calendar year = accident year + development period (1 to $Z-1$).
- The next $Z-2$ observations are accident year = 2, development period = 0 to $Z-3$, and calendar year = accident year + development period (2 to $Z-1$).

The macro that writes the residual matrix uses the same pattern as the macro that forms the loss triangle. Both macros assume an accident year by development period order.

Illustration: The $\frac{1}{2} \times N \times (N+1)$ points are in accident year by development period order.

- Accident year 0; Development periods 0 to $N-1$; Calendar years 0 to $N-1$
- Accident year 1; Development periods 0 to $N-2$; Calendar years 1 to $N-1$
- Accident year 2; Development periods 0 to $N-3$; Calendar years 2 to $N-1$

The macro reads the observations one by one and assigns the proper development period and calendar year to each.

Take heed: If you use a different sequence of observations, use the alternative macro which examines the actual development period and calendar year for each point.

Jacob: What if the observations are not in accident year by development period order?

Rachel: If the observations are *not* in accident year by development period order, the macro assumes the development period and calendar year are placed in the two columns to the right of the residual output.

- If the residual output has 3 columns (perhaps A, B, C, or J, K, L), the development period and calendar year are in columns D and E (or M and N). The current region has 5 columns, which are read into VariantData.
- If the residual output has 4 columns (A, B, C, D, or Q, R, S, T), the development period and calendar year are in columns E and F (or U and V). The current region has 6 columns, which are read into VariantData.

The macro reads the development period and calendar year from the work-sheet, and it assigns the residual in Column 3 of VariantData to the appropriate cell in Residuals.

- The residuals are in Column 3 of the VariantData array.
- The residual for observation #Ct is copied to the DevPer × CalYr cell of Residuals.

Take heed: You must copy the columns of development period and calendar year indices from the simulation worksheet to the residuals output worksheet.

Jacob: Why does the VBA macro have two options? Can't we read the development period and calendar year from the columns showing these indices?

Rachel: To derive the development period and calendar year, you must copy the columns with the values of X_1 and X_2 and paste them next to the residual output. This extra step may lead to errors. But if you copy the values of X_1 and X_2 and paste them next to the residual output, you can use the cell formulas instead of the VBA macro. You can check that the macro gives the correct results, and you see the workings of the project.

Jacob: If we use different input data, such as a parallelogram instead of a triangle, can we still use this macro?

Rachel: This macro assumes the input data are a paid loss triangle. If you use other input data, make the following changes to the macro:

- Eliminate the section testing that the data are from a loss triangle.
- Change the code for the dimensions of the residuals array. The array might have 0 to 14 for the development period and 0 to 28 for the calendar year.
- Change the code to create a residual matrix on the worksheet of a different size. The residual matrix might be in a range of 15 rows by 29 columns.

The changes listed above are straight-forward. If you are not proficient at VBA coding, comment out the rest of the macro and form means, standard deviations, and charts by Excel built-in functions.

The WriteResidualsMatrix procedure writes the Residuals array to an Excel range. The upper left cell of this range is two columns down and eight columns to the right of the cell with the words *Residual Output*. The range is a square, with each side having Z cells. (Z is the number of years.)

- The macro does not test if these cells are empty.
- If the cells are not empty, the macro over-writes the previous output.

Take heed: The VBA macro over-writes any values or formulas in the output range. In most cases, you want this: you have estimated initial values with one set of parameters to verify your procedure is correct, and how you over-write both the residual output and the residual matrix with new values from another set of parameters.

WRITING FROM VBA TO EXCEL

Jacob: How does the macro write the matrix of residuals to the Excel spreadsheet?

Rachel: The macro selects a range for the matrix of residuals. The *OFFSET* method starts from the active cell. It moves two rows down, leaving room for headers, and 8 columns to the right, leaving room for columns with the values of X_1 and X_2 , and a column of row labels for the residual matrix.

Writing the array to a range is a single statement: `rngMatrix = Residuals`. But take heed:

- For the paste command, you specify the upper-left cell.
- To write to a range, you must specify the entire range.

You can choose any location for the matrix of residuals by changing the offset parameters.

COLUMN HEADINGS, ROW LABELS, MEANS, STANDARD DEVIATIONS

Creating the column headings, row labels, means, and standard deviations for the residual matrix is tedious. The macro does the work.

- The rows are development periods. The macro places the numbers 0 to Z-1 in the column to the left of the residual matrix.
- The columns are calendar years. The macro places the numbers 0 to Z-1 in the row directly above the residual matrix.

Excel has built-in functions for the mean and the standard deviation. The macro invokes these Excel built-in functions to compute the means and standard deviations.

Take heed: The Excel built-in functions *AVERAGE* and *STDEV* use cells with data. The variant array has empty cells where development period > calendar year. If you format the arrays as numeric data, the empty cells may have zeros, which will distort the means and standard deviations.

Illustration: We compute the average of three cells: A1 = 5; B1 = 8; C1 has no entry.

- If Cell C1 has numeric formatting and a value of zero, the average is 4.
- If Cell C1 is empty, the average is 6.

Take heed: Verify the output from the macro. The macro works with any size array, but it may not work correctly if you change the format of the data. Verify the computations by hand, particularly if you change the size or format of the data.

CHART

The VBA macro forms simple residual plots. Use them to work through the student project. Vary the stochasticity (σ), payment decay, or inflation, do the simulation, run the regression analysis, form residual plots, and analyze the results.

After running several simulations, you should understand how the residual plots test for stability of the regression coefficients. For your write-up, explain how the slope or curvature of the residual plot reflects the change in the regression coefficient.

- You can add the documentation in your Word write-up, << or >>
- Use highlights, call-outs, or text boxes on the chart, and copy the chart to the write-up.

VBA MACRO RESIDUAL PLOTS

The VBA macro creates residual plots of the type needed for the student projects on parameter stability and heteroscedasticity.

- Use these plots as templates: they show what is needed for the student projects.
- Add features or documentation to the residual plots as appropriate for your project.

Illustration: You might show the slope of each line segment in the residual plot.

The macro places the residual plots on the worksheet with the residual output. You can move the chart to its own chart sheet if you like (or change the macro to do so).

The macro first deletes all existing charts on the worksheet. Most commonly, your first plots had errors and you want to replace them. If you want to keep the charts on your worksheet,

- Copy the residual output to a new worksheet before running the macro, << or >>
- Comment out the lines deleting the charts on the active worksheet.

The residual matrix shows means and standard deviations of the residuals by development period and calendar year. The macro forms four residual plots:

- Means of the residuals by development period.
- Standard deviations of the residuals by development period.
- Means of the residuals by calendar year.
- Standard deviations of the residuals by calendar year.

The project template on parameter stability uses means of the residuals. The project template on heteroscedasticity uses standard deviations of the residuals.

The residual plots are placed around the residual matrix.

- The residual plots by development period are placed 50 points to the right of the residual matrix, which is 8 columns to the right of the beginning of the residual output.
- The residual plots by calendar year are placed below the residual matrix.

The residual plots have heights of 250 points and widths of 500 points. You can resize the plots and move them elsewhere if you want.

The macro uses line charts for the means and column charts for standard deviations.

The category axis for the development period or calendar year is zero based: the values range from 0 to NoYrs – 1. The values are from the range associated with the variable *rngLabels*. (If the chart is not told the values of the category axis, it would use 1 to NoYrs.)

The means of the residuals by development period are the first column in the range referenced by the variable *rngMeansDP*. (The second column is the standard deviations.) The first row in this range is column headers, so the macro

- Uses the first column: Columns(1)
- Moves down one row: Offset(1,0)
- Uses a range size of rows = NoYrs and columns = 1: Resize(NoYrs, 1)

MINIMUM AND MAXIMUM VALUES

If the regression coefficient changes, the average residual is a maximum or a minimum at that point. The macro determines the maximum and minimum mean residuals and the indices at which they occur. The chart shows the values at these points.

Take heed: Either the maximum or minimum should be an end-point, and the other should be the point where the regression parameter changes. If you do not get this result, you may have made an error. If the point is too high or low by one year, you may have indexed the values incorrectly or fluctuations may distort the expected results.

Illustration: The inflation rate changes in calendar year 8, but the maximum or minimum residual is at calendar year 7 or 9. Check that you are indexing the years properly.

Take heed: If you use a gradually changing regression parameter and a moderate or high σ , the maximum or minimum value of the mean residual may not be the turning point.

SLOPES

The slope of the residual plot reflects the difference between the actual coefficient and the ordinary least squares estimator for the coefficient.

Illustration: The ordinary least squares estimator for the inflation rate is 12%, and the actual inflation rate is 8% a certain year.

- The expected mean residual may be positive or negative that year. We must know the actual inflation rate in all years to determine if the mean residual is positive or negative.
- The actual inflation rate is 4% less than the ordinary least squares estimator, so the slope of the residual plot at that point is -4% .

Take heed: Practicing statisticians use software that produces the needed residual plots to test for parameter stability and heteroscedasticity. Your student project should explain the residual plots and use them to create a better (more accurate) fit. In past years, some candidates who were not proficient in Excel spent days forming the Excel tables and charts and had little time to interpret the results. We provide illustrative worksheets, functions, and macros, as you have in statistical software packages. The course instructor reviewing your student project focuses the inferences you draw from the statistical tests. We want to see that you understand the concepts and can draw the proper conclusions from the statistical techniques, not that you can replicate the Excel charts.

MACRO VS CELL FORMULAS

Jacob: To form the residual matrix, is it better

- To use the VBA macro or
- To copy and paste the values of X_1 and X_2 and use cell formulas?

Rachel: Candidates differ in their preferences.

- Some candidates have never used VBA, but can handle the cell formulas.
- The VBA code is most useful if you simulate several times. You can automate much routine work, from running the simulation to forming the residual plots.
- We do not provide macros for the statistical work.
- The VBA macros automate the time-consuming portions. You combine the pieces with appropriate cell formulas to show you understand the logic.

We explain cell formulas and VBA code that candidates may not be familiar with.

Illustration: New Excel uses sometimes avoid Variant arrays because Variant data take more storage and are less efficient. This was relevant many years ago, when computers were slower and processing time was critical. It is still relevant for complex simulations using hundreds of thousands of realizations.

VBA uses Variant arrays to read from and write to Excel worksheets. This is much more efficient than using numeric arrays.

Jacob: The project template explains the cell formulas for the statistical procedures. If the VBA macro does the work, do we need the cell formulas?

Rachel: Everything can be done by cell formulas: copy, paste, average, standard deviation. We recommend that you reproduce the Excel worksheets with a different simulation before using the VBA macro. Unless you understand all the pieces, it is hard to write the project.

Jacob: If I want to copy residuals from a table to a matrix, I use copy and paste commands. I don't see any copy and paste commands in the VBA macro.

Rachel: Copy and paste would be confusing. The macro must shift between the VBA and Excel environments and determine the proper cell for each copy or paste command. The better method is for the macro to read once from the worksheet, reformat the figures, and write once to the worksheet.

Jacob: Instead of 15 years, I use 20 years for my simulation. Must I verify that the residual matrix is correct?

Rachel: Definitely verify that you are properly doing each part of the student project.

- Be sure residuals are properly assigned to development periods and calendar years.
- The VBA macro writes labels for development period and calendar year, using the index values you assign. Verify that these are correct.

Jacob: What exactly should we verify?

Rachel: Choose a pair of indices, such as development period = 5 and calendar year = 7.

- The *REGRESSION* add-in gives the α and β coefficients.
- Use the index values and the coefficients to derive the estimated Y value.
- The residual is the observed (simulated) Y value minus the estimated Y value.

Illustration: The simulation parameters may be $\alpha = 10$, $\beta_1 = -0.25$, $\beta_2 = +0.08$.

- The ordinary least squares estimators may be 10.55, -0.245 , and $+0.086$.
- The simulated Y value may be 9.500 (using simulation parameters and an error term).
- The expected Y value, using the least squares estimators and no error term, is

$$10.55 - 5 \times 0.245 + 7 \times 0.086 = 9.927$$

- The residual is $9.500 - 9.927 = -0.427$

Jacob: After verifying the residuals, what else should we check?

Rachel: The VBA macro computes means and standard deviations. Use the Excel built-in functions to verify a few figures.

Jacob: What types of errors should be watch for?

Rachel: The most common error is simulating a change in a parameter and correcting it with dummy variables or a third independent variable.

Illustration: Suppose you simulate a gradually changing inflation rate from 6% in year 0 to 20% in year 14.

- Verify several simulated figures from different development periods and calendar years to be sure your formulas are correct.
- Your final regression equation has three independent variables, one of which is the square of the calendar year. Pick a few figures to be sure this regression equation gives the correct figures.