Clustering Training

TRAINING GUIDE
LAST UPDATE: APR 2, 2019

GGY AXIS Inc.
5001 Yonge Street
Suite 1300
Toronto, ON
M2N 6P6

Phone: 416-250-6777
Toll free: 1-877-GGY-AXIS
Fax: 416-250-6776
Email: axis@ggy.com
Web: **www.ggy.com**

Table of Contents

1. GENERAL STEPS for CLUSTERING

** The ”(PRERUN)” steps show the extra steps if you want one of your location variables to be based on an AXIS result

1. Build the full seriatim DataLink and DataLink Macro batch as normal
2. (PRERUN) Run model with Seriatim Report
    a. Go to View / Report / Seriatim Summary Report or Seriatim Calendar Year Report. Create a report
    b. Create a Batch “CY Projections Recalculation” or “Batch Testing” to run all policies. Add the report to export during the run. Send results to Import / Export Database
    c. Create a Remote Table to link to the table in the Import / Export Database
3. Activate feature code 272 “Seriatim Data Compression”
4. Refer to Clustering Guide in Help / AXIS Online Help / AXIS User Guide / “User Guide for Cluster Compressed Data Model
5. Create “Cross Table Formula” Batch
    a. For table A use your full seriatim Policy Information table
    b. For table B if you are doing (PRERUN) use the Remote Table from step 2, otherwise use a dummy table with the same policy IDs as table A.
    c. Link the tables by policy ID
    d. Click the Generate Formula button to get a template. In the code, it should look something like the following:

```
<EXECUTE ONCE>

=SetSimilarityMeasure (EUCLIDEAN_DISTANCE)

=SetClusterLinkageRule (IMPORTANCE)

=SetOutputLinkageRule (CLOSEST_TO_AVGLOCATION)

=SetCompressRatioAndOutputTable (40, "CLUSTER Output Table 40", "CLUSTER Report Table 40")

=StandardizeLocation(0,1)

=SetLocationVariableCount(2)

=SetLocationVariableWeight(1,1)
=SetLocationVariableWeight(2,1)

<\EXECUTE ONCE>

=SetPolicyIDAndMeasure (A.[POLID], A.[CURGUAFUN1])

=SetLocationVariable(1, A.[ISSAGE])
=SetLocationVariable(2, A.[ISSYEA])
```

    e. Note that ifthe Policy Count optional field is used, then the measure must be the adjusted for the Policy Count otherwise the answers will be wrong (since the measure will be multiplied by the policy count when you run the seriatim policy in AXIS). In this example, the measure should be changed to something like A.[CURGUAFUN1] * A.[POLCOUNT]
6. (PRERUN) In the code, also incorporate fields from your table B
7. (Optional) Create new DataLink Table “Compression Table” and “Compression Report Table”

called in the SetCompressRatioandOutputTable() function in the Cross Table Formula batch above. These can be left blank.

8. (Optional) If you want to use Compression Segments, for example if your OLD product doesn't behave well with compression but your NEW product can be compressed, you can just apply compression to the NEW product. Another benefit of this is to reduce run times to compress.
    a. Add the Compression Segment optional field to your Policy Information Table. Put logic in for this field to define segments
    b. In the Cross Table Formula, modify the SetCompressRatioAndOutputTable() code to say something like this instead:

```
IF A.[COMSEG] = "OLDPRODUCT" THEN

=SetCompressRatioAndOutputTable (100, "CLUSTER Output Table 100", "CLUSTER Report Table 100")

ELSE

=SetCompressRatioAndOutputTable (40, "CLUSTER Output Table 40", "CLUSTER Report Table 40")

ENDIF
```

    c. Similar to step 7, create the corresponding Compression Table and Compression Report Table referenced in your code
9. Add extra fields to Policy Information Table
    a. Do Not Export – set to "calculated" field, leave empty
    b. Grouping Factor – set to "calculated" field, leave empty. Pay attention to number of decimals set for the field.
10. Modify DataLink Macro batch
    a. Add "Perform Compression" step using Cross Table Formula from step 5
    b. Add / Modify "Export Seriatim" step. One should be created for each compression table in your code in steps 5, 6, and 7
        i. Activate "Specify Seriatim Data Model Name" switch
        ii. Give the model a Data Model name, this will allow you to have multiple seriatim data (e.g. "full seriatim" &"clustered seriatim") in a single model
        iii. Select the Compression Table
11. Run DataLink Macro
12. Review DataLink Tables
    a. In the Compression Table
        i. You will see the parent policies along with the grouping factor. Pay attention to number of decimals set for the field—you can change this and rerun compression if needed.
    b. In the Compression Report Table
        i. You will see the parent, included child policies, distance, parent & child measures
        ii. The grouping factor is equal to (sum child measures) / (parent measure) such that the Measure is preserved before and after grouping
13. Review Cells
    a. Seriatim policies will have the Grouping Factor, this factor is a multiplier on all results coming from the policy.
14. Switch Data Models
    a. To switch between multiple data models as set in step 9.b.ii. go to Tools / Inforce Seriatim / Set Active Model
15. (Optional) Set clustering to also set up a separate count for Expenses so per-policy expenses are preserved.
    a. In step 5, instead of =SetCompressRatioAndOutputTable (40, "CLUSTER Output Table 40", "CLUSTER Report Table 40") use =SetOutputsWithExpenseGrouping(40, "CLUSTER Output Table 40", "CLUSTER Report Table 40")

b. In step 7, append the optional field Grouping Expense Factor to the Compression Table. This field can be left empty. Pay attention to number of decimals set for the field.
c. In step 9, append the optional field Grouping Expense Factor to the Policy Information table, set to “calculated” field and leave empty. Pay attention to number of decimals set for the field.
d. In step 12.a, Grouping Expense Factor will also be populated
e. Grouping Expense Factor will appear in the final seriatim and is used as the policy count for the purpose of calculating expenses

2. Background (5 minutes)

2.1 WHOLE SESSION TAKES ABOUT 3 HOURS FOR DETAILED, 1 HOUR FOR HIGH LEVEL. THE SESSION REALLY REQUIRES COMFORTABILITY WITH ALL DATALINK FUNCTIONALITY. NEW USERS WILL NOT GET THIS EASILY.

A training example recording is available in the Recorded Training under the Clustering tab. It's called "Clustering Training Example"

2.2 Discuss Purpose of Grouping / Clustering

1. Go through Powerpoint slides until Grouping demo
    a. Try not to mix up "grouping" vs. "clustering"—we will be considering them as different topics
2. Discuss purpose of grouping and clustering
    a. Trade-off between accuracy and run-time
    b. Discuss briefly high level difference between traditional grouping and clustering
    c. Necessary to discuss this because both methods are useful for different situations

3. Traditional Grouping Example (1 hour)

3.1 Create a Traditional Grouping Model

1. If this is a more high level discussion, you can skip building the model and just review the existing sample dataset for the model setup and results
2. Discuss basic Grouping
    a. This has existed for a long time
    b. Query is the key object here, you will define the rules here to group
    c. Our example is going to group by Issue Age and Issue Year
3. Turn on Feature Code 272 SERCOM

Start with Variable Annuity Valuation cell in Sample dataset

- Show tables in DataLink
    - Look at Policy Information Variable Annuity as base table with full seriatim
        - Initialize / Load / Map and review
- Copy Policy Information Variable Annuity into a new policy information table “Policy Information Variable Annuity Grouped”
    - Add user defined field for Grouped Age (calculated field)

```
DO CASE

CASE [ISSAGE] < 30
        [GROAGE] = 30

CASE [ISSAGE] < 40
        [GROAGE] = 40

CASE [ISSAGE] < 50
        [GROAGE] = 50

CASE [ISSAGE] < 60
        [GROAGE] = 60

OTHERWISE
        [GROAGE] = 65

ENDCASE
```

        - Initialize / Load / Map and review
    - Note another option that could be used for age banding is the _BAND() function
    - Add optional fields to “grouped” table
        - Policy Count (calculated field)
    - Initialize, Load and Map to see table

    - Create a Query, group by GROUPED AGE and ISSUE YEAR
        - Average all numeric, First all non-numeric
            - User needs to decide what to do with each variable
        - For Policy Count, set to “Count” instances of Policy Count entries
        - Make a table / notice this is a user-defined table

    - Create Collect Records Batch
        - Match Fields
        - Update Issue Age with Grouped Age

        - Review the Policy Information Variable Annuity Grouped table
            - Notice it was appended, this will be fixed if we initialize during the batch

- Create a DataLink Macro batch
    - Initialize both tables, Load, Map
    - Run Query / Create Table
    - Initial table again to clear it out
    - Collect Records
    - Export to AXIS
        - Show the ability to create a different Data Model at this point

- Review Cells and Test results (try different values for policy count and review results)

4. Agglomerative Clustering (2 hours)

4.1 Create a Clustered Model

1. If this is a more high level discussion, you can skip building the model and just review the existing sample dataset for the model setup and results
2. Discuss basic Clustering
    a. This is more accurately called “Agglomerative Clustering”
    b. Link the concepts back to the grouped model for better flow and understanding
    c. Show Powerpoint for diagrams on clustering methodology

- Discuss high level clustering method.
- Compare clustering to traditional grouping. Show Powerpoint slide “Clustering Terminology”. Discuss concepts of
    - “Measure”
        - These are the values that are preserved between the original and final models
        - In the trad example, the measures were Guarantee Fund, Policy Size, etc. etc. anything that we averaged in the Query. Plus policy count.
        - In clustering we will only have 1 measure—we will use Guarantee Fund
    - “Location variables” & “distance”
        - These are the variables that we use to determine similarity.
        - In the trad example, we pre-determined the distances with Issue Age and IssueYear
        - In clustering the distance will be determined dynamically
        - Both methods allow for multiple location variables
            - For trad example, adding more location variables will automatically add more groups
    - “Representative Policy”
        - In the grouping example, we determined the representative policy using the query to get an average policy from the component policies—this policy does not really exist—how would you handle fund values, in the moneyness, etc?
        - In clustering we will have rules to determine a “centroid” based on the average policy. Then a representative policy will be determined as being closest to this centroid—do not need to make “averaging” assumptions for all the policy specific details
        - Note that if all policies are equi-distant from the centroid, the representative policy will be selected arbitrarily
- Show Powerpoint diagrams on clustering methodology. Show a single cluster and how the distance is calculated
    - Clustering is one at a time
    - This means that distances do not have to be recalculated, one point disappears every time

4.2 Show a Complete Clustering Model

1. Open the Sample dataset in a version of AXIS where the clustering model is available (after 2014.10.01) in the Annuity module
2. Notice the Variable Annuity Valuation cell has 24 policies
3. Show Tools / Seriatim / Set Active Model
    a. Set active model to Cluster Example
    b. Look at Variable Annuity Valuation cell, it now has 13 policies
4. In the seriatim listing show compression segment and grouping factor
    a. These are parameters in the clustering methodology, will show later
    b. Notice the grouping factor (try a different value and review results)

***At this point, show Clustering guide in help text

4.3 PART A – BASIC SETUP

- Go to Policy Information Variable Annuity
    - Add optional fields for Grouping Factor, and Do Not Export (both are calculated fields)
        - Explain Do Not Export field
        - Note that Do Not Export, if already used, is still active
    - Create a copy dummy table
        - Theoretically, the clustering should work on a single table, but currently we need to have two tables to allow the Cross Table Formula to function. As well this will allow us more power to use AXIS results as location variables later on
    - Load both tables
        - The tables have to be initialized before the cross table formula
    - Create a cross table formula
        - A is your main table with all the policies to be grouped
        - B in the future will be a remote table linking to AXIS results
        - “Find first matching record in table B” (not sure if this matters)
        - Link by Policy ID
    - Show formula in help text
    - Can click the Generate Formula button to give the basic code (similar to help text)
        - All code between <EXECUTE ONCE> and <\EXECUTE ONCE> is the “model specific” section and is run once per segment (to be discussed later)
        - Linkage determines which policy is kept during the clustering phase

```
<EXECUTE ONCE>

=SetSimilarityMeasure (EUCLIDEAN_DISTANCE)
=SetClusterLinkageRule (IMPORTANCE)
=SetOutputLinkageRule (CLOSEST_TO_AVGLOCATION)

=SetCompressRatioAndOutputTable (40, "Output Table 40 Percent", "")

=StandardizeLocation(0, 1)

=SetLocationVariableCount(2)
=SetLocationVariableWeight(1,1)
=SetLocationVariableWeight(2,1)

<\EXECUTE ONCE>

=SetPolicyIDAndMeasure (A.[POLID], A.[CURGUAFUN1)
=SetLocationVariable(1, A.[ISSAGE])
=SetLocationVariable(2, A.[ISSYEA])
```

    - Create a Batch to Initialize / Load / Map Policy Information table
        - Add step to run compression
    - Run and look at resulting Output Table
    - Edit Batch
        - Add step to export seriatim with compression table
    - Not that for more compression levels (80%, 60%, etc.) you need to create a different DataLink Macro for each Data Model

- Create Report Tables. Change code to:

```
=SetCompressRatioAndOutputTable (40, "Output Table 40 Percent", "Report Table 40 Percent")
```

- Review the output table

- Parents and children are all shown here
- Measures are available to help reconcile the Grouping Factor = Total Measure / Parent Measure
  - Reconcile one of the grouping factors
- Distances are the original distances between the parent and child (not distance to centroid) that were used in determining groups

4.4 PART B – SEGMENTS

1. Discuss concept of Segment
   a. All code between <EXECUTE ONCE> and <\EXECUTE ONCE> are done once per segment
   b. This is a useful feature when user has the intention to segregate and prevent clustering between major groupings of policies, such as Fixed vs. Variable Annuities, or Term Life vs. Whole Life.
   c. Cluster compression by Segment may actually reduce the run time of the compression process significantly. When the [Compression Segment] field is not specified, all policies are treated as one Segment.
2. Add Compression Segment field to Policy Info table

```
IF [ISSYEA] < 2013 THEN
    [COMSEG] = "OLDPRODUCT"
ELSE
    [COMSEG] = "NEWPRODUCT"
ENDIF
```

3. Modify Cross Table Formula to use different compression ratio for OLDBUSINESS.

```
IF A.[COMSEG] = "OLDPRODUCT" THEN
=SetCompressRatioAndOutputTable (100, "OLD Output Table 100 Percent", "OLD Report Table 100 Percent")
ELSE
=SetCompressRatioAndOutputTable (40, "Output Table 40 Percent", "Report Table 40 Percent")
ENDIF
```

4. Modify DataLink Macro and add another Export Seriatim step using the “OLD Output Table 100 Percent” – this will add the uncompressed policies to the compress policies within the same data model
   a. If you set this up as two DataLink Macros, the 2nd run would overwrite the 1st run.

4.5 PART C - RESULT-BASED LOCATION VARIABLES

1. Refer back to FR case study slides
   a. Talk about the PV amounts and how they required pre-processing
   b. Discuss a pre-run (maybe last month, or a full pre-run)
2. Create Seriatim Summary Report
   c. Fund Value
   d. Guarantee Fund 1 / 2 / 3 / 4 / 5
   e. GMDB Exposure
   f. GMWB Base
3. Create Batch to run report
   g. Use the Calendar Year Projections Recalculation batch if you are only running 1 scenario
   h. Use the Batch Testing batch if you want to run many scenarios
      i. You will need to also set up a Query to summarize the multiple scenarios into 1 result per policy (e.g. average fund value across multiple scenarios)
4. Run Batch
5. View Results in I/E database

i. Results need to show one lin

6. Create a Remote Table to link to results
7. Update Cross Table Formula
    j. Remind trainees that this table originally used a dummy table as the second table

```
=SetLocationVariableCount(4)
=SetLocationVariableWeight(1,1)
=SetLocationVariableWeight(2,1)
=SetLocationVariableWeight(3,1)
=SetLocationVariableWeight(4,1)

<\EXECUTE ONCE>
=SetPolicyIDAndMeasure (A.[POLID], A.[CURGUAFUN1])
=SetLocationVariable(1, A.[ISSAGE])
=SetLocationVariable(2, A.[ISSYEA])
=SetLocationVariable(3, B.[GMDB exposure])
=SetLocationVariable(4, B.[Annuity fund Balance ] / B.[Annuity GMWB base])
```

8. Rerun DataLink Macro Batch

Note that it's possible to do this with a seriatim CY report as well, but there are a few issues to navigate:

1. Generate your Seriatim CY report with your AXIS run. For the parameters of the report, set the report to Transpose. By doing this you put the report in a more useable format (policy IDs down the rows, reporting line across the columns). You can also ignore the initial status column if you don't need numbers from your valuation date (e.g. Dec 31, 2016):
2. Your results will go to the Import / Export database. Notice that you will have multiple rows with the same [Policy ID] field but different [Row] fields. Let's say you only want the values in [Row] = "Y2017" as a location variables.
3. Set a Remote Table to point to those results. Include the Remote Table in your Cross Table Formula batch. Now a couple of tricks for your batch:
    a. Set the For each record in table A to "Find all matching records in table B"
    b. In the formula code, you need to ignore any rows that aren't your particular date. In particular use this code to ignore any records where the [Row] <> "Y2017": For example:

```
IF B.[Row] = "Y2017" THEN
=SetPolicyIDAndMeasure (A.[POLID], A.[CURGUAFUN1])

=SetLocationVariable(1, A.[ISSAGE])
=SetLocationVariable(2, A.[ISSYEA])
=SetLocationVariable(3, B.[Gross benefits - death])
=SetLocationVariable(4, B.[Gross benefits - other])
ENDIF
```

4. So when you run, DataLink should only link 1 record by policy ID to your Policy Information table when it runs the formula—which a necessity to get this to work.