Illustration of Segmentation Visualization Before and After Clustering using t-SNE in SAS® Viya.

Segmentation is an extremely valuable and popular analytic technique that is used in Marketing, Market Research, Scientific Discovery, and in almost every industry. The method of t-SNE [1] helps understand the underlying structure of a data set visually keeping the best representation of high to low dimensionality in tact for better visualizations in two or three dimensions. In this blog I show how you can take a data set and transform the attributes, so they won't be too scattered; aka revenue and count distributions are often extremely skewed and thus need a transform such as Log, Square-root, or other data transform functions. Then I'll show the same attributes in a cluster segmentation and re-visualize the same t-SNE analysis but highlighting the cluster segments. In this fashion one can learn the underlying data structure and after clustering observe if the cluster segmentation is somewhat visually representative of the pre-clustering t-SNE depiction of the initial visual discovery.

t-SNE is a method of converting a high-dimensional data set into a matrix of pair-wise similarities that can capture most of the local structure of the high-dimensional data well and revealing the larger structure such as the presence of clusters [1]. The original method (SNE) [2] was developed by Hinton and Roweis [2] in 2002 and is later enhanced to t-SNE using a Student's T distribution which gives better properties for data mapping between high and low dimensions and in optimizing the cost functions [1]. First, the following SAS code will connect an in-memory session (CAS) and set up a stratified random sample of the data set which has about 105,000 observations; plotting that many observations would obviously be difficult to observe and take a long time in the t-SNE analysis, so a sample of 5,000 observations will be used.

Some of the benefits of using t-SNE over other dimension reduction techniques are:

- t-SNE is a non-linear technique that keeps the low-dimensional data representation of very similar datapoints close together, which is typically not possible for linear mapping techniques such as Principal Component Analysis (PCA) and Multi-dimensional Scaling (MDS).
- t-SNE keeps the local and global structure of the data in a single map compared to other techniques in PCA, MDS, or Variance Unfolding (MVU) [1].

t-SNE Visualization

Submit | Cancel | History | Add as Snippet

Code | Log

```
1  cas mySession sessopts=(caslib=casuser timeout=1800 locale="en_US" metrics='true');
2  caslib _all_ assign;
3
4  proc sort data=RACOLL.CUSTOMERS out=WORK.SORTTempTableSorted;
5      by channel public_sector us_region RFM;
6  run;
7
8  proc surveyselect data=WORK.SORTTempTableSorted out=work.RandomSample
9          method=srs sampsize=5000;
10     strata channel public_sector us_region RFM / alloc=prop;
11 run;
12
13 proc delete data=WORK.SORTTempTableSorted;
14 run;
```

The stratified attributes are the channel (0=none, 1-reseller purchase only, 2-direct purchase only, and 3-both direct and reseller purchase), public sector (0=no, 1=yes), US region (six geographic state regions), and RFM cell category (A-K unique cells). Next, I transform the attributes using Log functions.

As an example, the plot of estimated IT spend is shown below: Figure (1) shows the original data; Figure (2) shows the log transform of estimate IT spend.

Figure (1) Original IT Spend

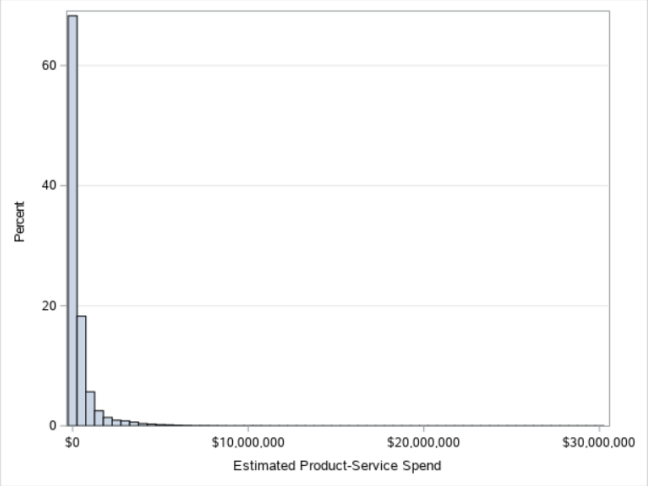
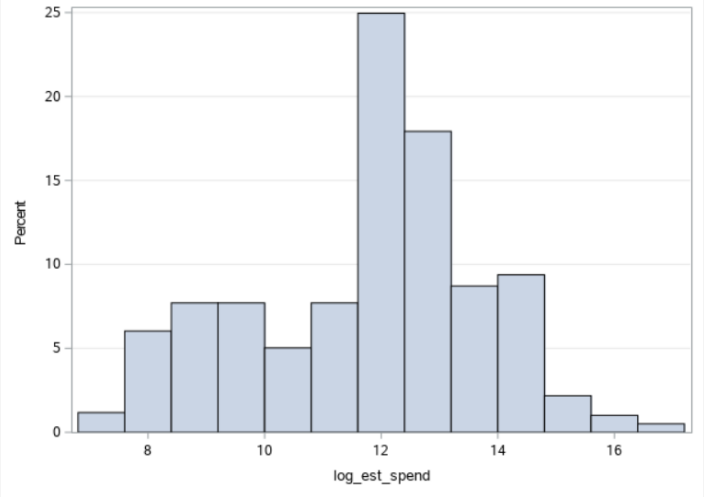


Figure (2) Log of IT Spend



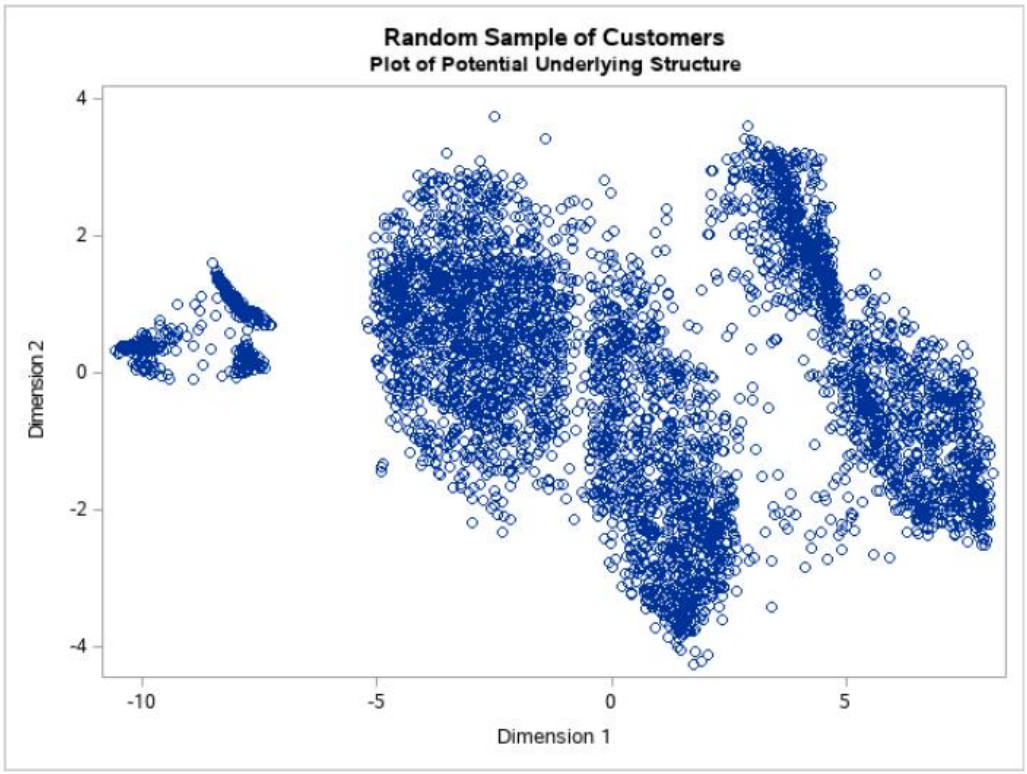
Now, the following SAS code will run the t-SNE procedure on the sampled data set that contains the transformed attributes.

```
26    title 'Random Sample of Customers';
27    proc tsne data=casuser.customers ndimensions=2 ;
28       input log_est_spend log_rev_thisyr log_corp_rev log_tot_rev channel
29    public_sector;
30    output out=casuser.tsne_output ;
31    run;
32
33    title 'Random Sample of Customers';
34    title2 'Plot of Potential Underlying Structure';
35    proc sgplot data=casuser.tsne_output;
36        scatter x=_dim_1_ y=_dim_2_ ;
37    run; title; title2;
```

The plot of the t-SNE analysis is shown in Figure (3) below. From this plot we can ascertain that there appears to be five unique general clusters and possibly three sub-clusters on the left side of the plot. While the data set has other attributes and may suggest other potential underlying structures, the six attributes used appear to have these five major clusters.

Figure (3)

Results: t-SNE Visualization



The goal now is to use the full data set and the same attributes with similar transforms and cluster them to see what cluster segments arise; then we'll plot another 5,000 random observations and highlight the cluster segment ID's and compare the before and after clustering plots.

The next step I'll use SAS VDMML graphical interface called Model Studio® to perform the transforms and clustering. The process flow forms what is called a pipeline and is shown in Figure (4) below. The table in Figure (4a) shows the automated transforms used by the Transforms node.

Figure (4a) Results of the Transform Node in Model Studio Pipeline

Transformed Variables Summary

Transformed Variable	Method	Input Variable	Formula ↑	Variable Level	Type	Variable Label
LOG_corp_rev	LOG	corp_rev	log('corp_rev'n + 1)	INTERVAL	N	Transformed Corporate Revenue last fiscal yr.
LOG_est_spend	LOG	est_spend	log('est_spend'n + 1)	INTERVAL	N	Transformed Estimated Product-Service Spend
LOG_rev_thisyr	LOG	rev_thisyr	log('rev_thisyr'n + 816363.06)	INTERVAL	N	Transformed This Years Fiscal Revenue YTD

Clustering Pipeline

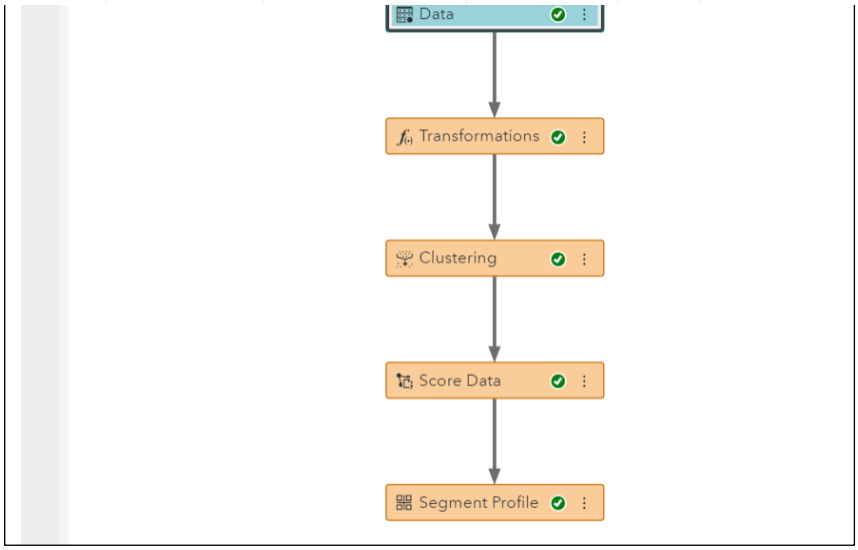


Figure (4) Solution

The clustering node settings include the cluster initialization using the Forgy algorithm as cluster seeds. The numeric attributes are using Euclidean distances, classification variables are using binary similarity distances and the number of initial clusters is set to five. The data set is scored as the initial default settings of the project included 60% random sample for training, 30% random sample for validation, and 10% is a holdout sample called test. The scoring function was automatically captured as rules from the clustering node and executed on all three partition data sets. The Segment profile node was run to observe what attributes are mostly responsible for each segment. Figures (5) and (6) show the cluster segment breakout and the worth statistics of cluster segment 1.

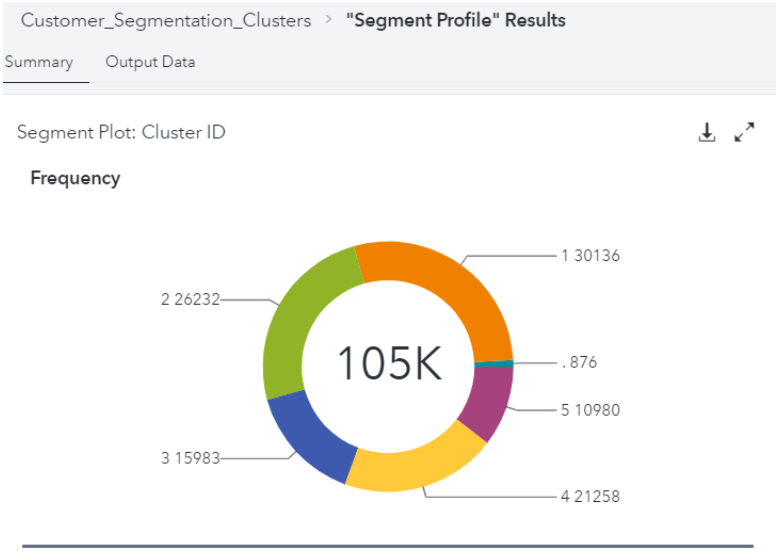
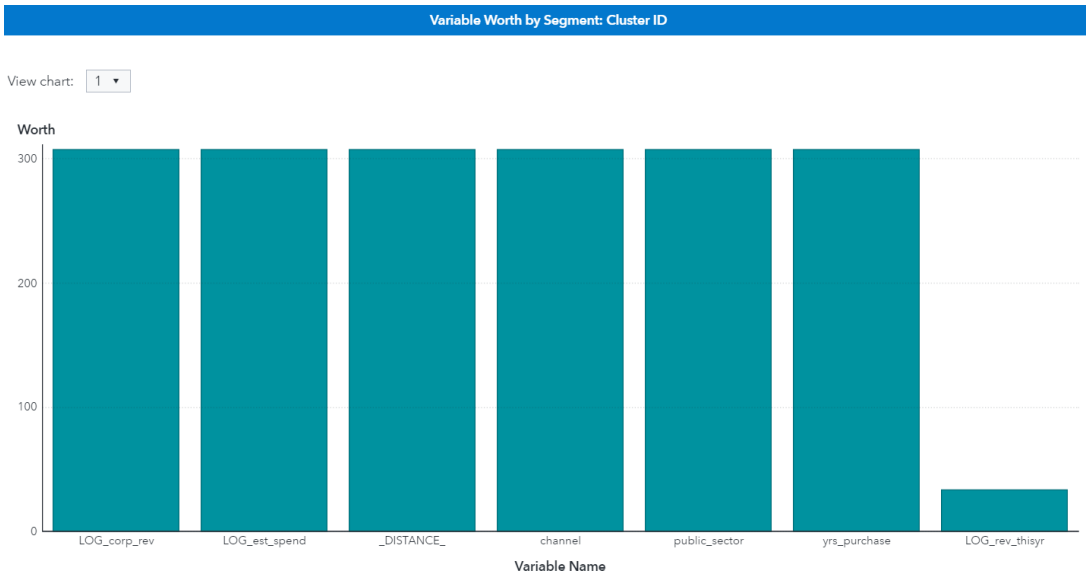


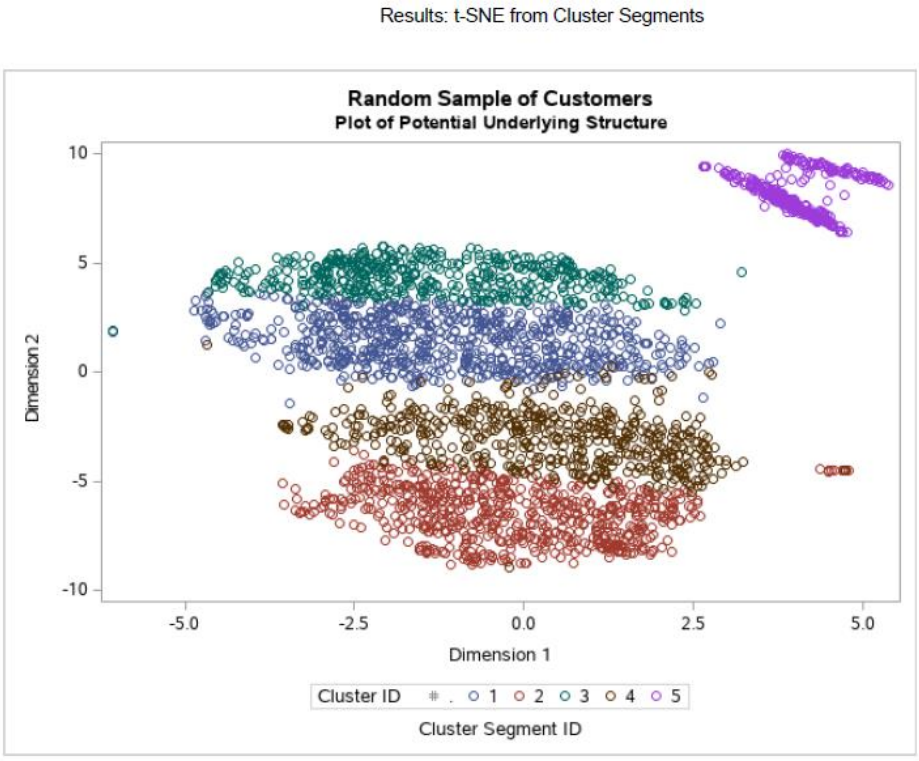
Figure (5) Cluster Segment Breakout

Figure (6) Cluster Segment 1



Now we come to the last part where after the entire data set has cluster segment ID's applied to all observations, the next step is to take the same 5,000 random samples but also overlay the cluster ID's in the t-SNE plot. Figure (7) below shows the t-SNE plot with the cluster segments identified in the random sample.

Figure (7) t-SNE with Cluster ID's Overlaid



So, what can we conclude from these analyses? While the before and after clustering t-SNE analysis plots are different in dimensions 1 and 2, it isn't a stretch to observe that the cluster ID's in Figure (7) are similar in nature to the observable groups in Figure (3)! The key phrase is *similar in nature.* They aren't identical and the initial transforms weren't either, but similar. And since I didn't use the exact random seed in the SAS procedure Proc Survey Select, the 5,000 observations weren't identical in each plot but likely had overlap. Also, this isn't a carefully controlled experiment but rather a demonstration of how a t-Distributed Stochastic Neighborhood Embedding can aid in the visualization of potential underlying structures within a data set with clustering. This method can aid in the cluster visualizations, data discovery prior to clustering and the like. I hope this brief illustration helps to understand a little about the t-SNE algorithm and its potential uses in visualizing segmentations.

References:

[1] G. E. Hinton and Laurens van der Maaten, "Visualizing Data Using t-SNE", *Journal of Machine Learning Research*, Vol. 9, 2008, pp. 2579-2605.

[2] Geoffrey Hinton and S. T. Roweis, "Stochastic Neighbor Embedding", In *Advances in Neural Information Processing Systems,* volume 15, The MIT Press, 2002, pp. 833-840.