

Modifying Insurance Rating Territories Via Clustering

Qunca Zou, New Jersey Manufacturers Insurance Company, West Trenton, NJ

Ryan Diehl, New Jersey Manufacturers Insurance Company, West Trenton, NJ

ABSTRACT

In the United States, most property and casualty lines of insurances use territories to incorporate the geographical risk into their rating structures. The geographical rating territories in theory are based on homogeneous groupings of geographical areas. Based on expected levels of loss per exposure for each grouping, a rate can be determined. For many insurance companies, territory relativities are frequently updated because of loss experience or competitive market forces; however, often territory boundaries have not changed significantly over the years.

This is a problem of cluster analysis. Several procedures in SAS can be used to create a more homogeneous territory definition which is more commensurate with the company's own experience. However, these clustering procedures do not naturally produce contiguous groupings (i.e. groupings that only include geographic units that are adjacent to each other). Since contiguous territorial boundaries are desired, a contiguity constraint needs to be added to the clustering routine. There are no readily available procedures in SAS that can conveniently apply the contiguity constraint. Therefore the author attempted to use SAS macros to combine the principle of locality and ward method, aiming at modifying the current territories. Comparison of homogeneity of the new and old territories indicates that homogeneity is significantly improved.

INTRODUCTION

In the United States, most property and casualty lines of insurances use territories to incorporate the geographical risk into its rating structures. Insurers charge varied premium based on the rating territories. The geographical rating territories in theory are based on homogeneous groupings of geographical areas. Companies typically define territories as a collection of small geographic units (e.g., zip codes, counties or census blocks) with similar expected loss costs. Territories are necessary to differentiate individual risks on top of other rating variables since geography is considered one of the primary drivers of claims experience. A rate can be determined for each grouping based on expected losses per exposure (a.k.a. pure premium).

For many insurance companies, territory relativities are updated frequently to reflect recent loss experience or competitive market forces; however, traditional territory definitions are not changed significantly over the years. This is due to the complexity of determining new territorial definitions, as well as the rate dislocation that entirely new rate boundaries may cause existing policyholders. Because of the changing environment (new housing development, changes in commuting patterns, demographic changes, etc), boundaries which may have been relevant decades ago may have lost their meanings over time. Therefore it is necessary to modify the current territory definitions so as to appropriately reflect the geographical risk.

One method for the modification is to overhaul the current territory classification using historical loss experience by applying various cluster analysis techniques. SAS procedure PROC CLUSTER is readily available to achieve this, which is promising to create more homogeneous territories that commensurate with the company's own experience. It is important to note, however, that these types of clustering routines do not naturally produce contiguous groupings (i.e., groupings that only include geographic units that are adjacent to each other).

Since contiguous territorial boundaries are desired here, a contiguity constraint needs to be added to the clustering routine. There are no readily available procedures in SAS that can conveniently apply the contiguity constraint. In this paper, the author attempted to revisit the current territory definitions by SAS macro programming. The idea of principle of locality is incorporated in the macros to create more credible pure premium for each Geographical unit. The cluster analysis carried out in this paper uses Ward's minimum variance method, which is described below.

DEFINITIONS

Several definitions are necessary before going forward. These are standard terms used in property and casualty (“P&C”) insurance, and are provided for non-insurance professionals.

Exposure – the basic unit of risk that underlies the insurance premium. For example, one house insured for one year represents one exposure for homeowners insurance. For this paper, amount of insurance purchased is used as the exposure.

Rate – the expected cost per unit of exposure. The rate must be sufficient to cover losses, expenses and a profit and contingency provision.

Premium – the final cost of the policy, which is equal to the product of exposure units and rate.

Loss cost or pure premium – the expected amount of losses per exposure unit; the goal of territorial ratemaking is to establish territorial rates that are closely correlated to expected loss costs.

DATA AND METHODOLOGY

GEOGRAPHICAL UNIT

Before any data preparation, the first step is to determine the geographical units. The unit should be refined enough to be relatively homogenous with respect to geographic differences while still having some observations in most units. The unit should also be easily accessible.

Typical units are zip codes, census blocks, counties, or some combination of these. Each of these options has practical advantages and disadvantages. For example, while zip codes have the advantage of being the most readily available, they have the disadvantage of changing over time. Counties have the advantage of being static and readily available; however, due to the large size of most counties, they tend to contain very heterogeneous risks. Census blocks are relatively static over time, but require a process to map insurance policies to the census blocks.

Postal zip codes are often used as the building blocks for territory definition mainly due to their availability. For the example presented in this paper, five years of Homeowners’ accident year data at zip code level for New Jersey including exposures and pure premium were obtained. Average annual pure premiums excluding the highest and lowest were calculated for each zip code to remove the possible outliers in the data.

Table 1 Typical Source Dataset

index	Code	X	Y	PPAOI_MID3	AOI_Exposures	Neighbors	n1	n2	...	n16
1	07001	-74.2715	40.5843	\$22.42	194238.35	7	7	32	...	
2	07002	-74.1105	40.6651	\$37.23	861389.57	1	99		...	
3	07003	-74.1870	40.8096	\$28.55	884175.03	8	11	12	...	
4	07004	-74.3043	40.8826	\$15.44	283052.53	8	6	31	...	
5	07005	-74.4204	40.9301	\$14.44	479414.77	7	30	39	...	
6	07006	-74.2812	40.8522	\$27.85	1011698.6	8	4	8	...	
7	07008	-74.2300	40.5843	\$43.05	415893.82	3	1	32	...	
8	07009	-74.2282	40.8564	\$36.83	444316.88	4	6	37	...	
...	...	...	...	...	...	...	...	...	...	...

A typical dataset used in the example is shown in table 1. There are five principal variables – “code” (zip code), “x” (longitude), “y” (latitude), “PPAOI_MID3” (pure premium), “AOI_Exposure” (exposure, adjusted for amount of insurance). In addition, to apply the contiguity constraint, the corresponding neighboring information is also retained for each zip code – “index” is a unique assigned number to each zip codes; “neighbors” represents number of neighboring zip codes for the corresponding zip code; while n1, n2, ..., n16 are the unique indexes for the neighboring zip codes. For example, zip code “07001” (index = 1) has 7 neighbors which have indexes 7 (zip

code 07008), 32 (zip code 07036, not shown in the table), etc. An adjacency table, which contains every pair of zip codes that are in contact, was used for the neighboring zip code information.

PRINCIPLE OF LOCALITY

The data can be thin for some of the zip codes shown in table 1. Generally the smaller the exposures for a zip code, the less credible it becomes. According to the principle of locality, the expected loss experience at a given location is similar to the loss experience nearest to that location. One may use similar idea to improve the estimate of any individual unit by using information from nearby units. This is sometimes referred to as spatial smoothing.

The creation of credibility-weighted pure premium for each zip code is a way of utilizing spatial smoothing techniques. The procedure starts out with a pure premium for each zip code. Then for each zip code, the latitude and longitude of the centroid of each zip is used to determine the group of zip codes whose centroid is within 5-, 10-, 15-, 20-, 25-, 50-mile radius of this zip code. A pure premium was calculated for each of the six groups. The statewide average pure premium was also calculated. For each zip code, the next step is to assign credibility to each zip code premium and the six groupings associated with that grouping. The credibility value was calculated

using earned premium and the formula $z = \frac{P}{P + K}$ where z is the credibility assigned, P is the Earned Premium and K is the credibility constant of \$2,500,000.

For the 5-mile radius grouping, the credibility assigned to the zip code was subtracted out to get the credibility assigned to this grouping's pure premium. For example if the zip code's credibility is 0.60 and credibility of the zip codes within 5-miles was 0.80, then 0.20 credibility was assigned to the 5-mile radius grouping. For the 10-mile radius grouping, the credibility previously assigned to the zip code and the 5-mile radius grouping were subtracted out of the formula credibility for the 10-mile grouping to get the credibility for the 10-mile radius grouping; and so on and so forth. If the sum of the assigned credibilities was not at 100%, then any remaining credibility was assigned to the statewide average pure premium¹. Now the credibility weighted average pure premium can be calculated as:

$$PP = z_0 \times PP_0 + z_5 \times PP_5 + z_{10} \times PP_{10} + z_{15} \times PP_{15} + z_{20} \times PP_{20} + z_{25} \times PP_{25} + z_{50} \times PP_{50} + z_{50+} \times PP_{overall}$$

where, PP is the credibility weighted pure premium, z_x (x can be 0, 5, 10, 15, 20, 25, 50, overall) is the credibility assigned to the corresponding radius group, and PP_x is the pure premium for each radius group.

The credibility formulas discussed above are just one among many choices for credibility and complement of credibility. This method is designed to pick up the information from surrounding geographical areas of the zip codes. For most zip codes in this study, almost all the credibility was assigned within a 10-mile radius.

METHODOLOGY

Several general types of cluster analysis methods exist. For each of these general types there are a number of specific methods and most of these cluster analysis methods can use a wide array of similarity or dissimilarity measures.

As in any cluster analysis problem, many different ways may be used to create the clusters. In any case, an appropriate similarity measure must be chosen first. A similarity measure is a way to tell how similar two units or two groups of units are to each other. Euclidean distance is the most often used similarity measure between two units; while centroid method (involving finding the mean vector location for each of the clusters and taking the distance between these two centroids), single linkage (the distance between the closest members of the two clusters), complete linkage (the distance between the farthest apart members), average linkage (involving looking at the distances between all pairs and averages all of these distances) are most often used to measure how close two clusters (groups of units) are to each other.

Once the measure of association as well as the method for determining the distances between clusters has been considered, one generally has two methods to proceed – agglomerative or divisive clustering.

For agglomerative clustering, one starts out with all sample units in N clusters of size 1; then at each step of the algorithm, the pair of clusters with the highest similarity is combined into a single cluster. The algorithm stops when all sample units are combined into a single cluster of size n .

For divisive clustering, one starts out with all sample units in a single cluster of size N . Then, at each step of the algorithm, clusters are partitioned into a pair of sub-clusters, selected to minimize the similarity between each sub-clusters. The algorithm stops when sample units are partitioned into N clusters of size 1.

Another alternative approach for performing cluster analysis is the Ward's method which is the method used in this paper. Basically it looks at cluster analysis as an analysis of variance problem, instead of using distance metrics or measures of association. It involves an agglomerative clustering algorithm. It will start out at the leaves and work its way to the trunk. It looks for groups of leaves that it forms into branches, the branches into limbs and eventually into the trunk. Ward's method starts out with N clusters of size 1 and continues until all the observations are included into one cluster. This method is most appropriate for quantitative variables.

For this particular method, three quantities need to be defined just as in ANOVA:

Error Sum of Squares: $ESS = \sum_i \sum_j (X_{ij} - \bar{X}_{i.})^2$. Here we are summing the squared difference

between the unit and its cluster over all of the units within each cluster. We are essentially comparing the individual observations for each variable against the cluster means for that variable. A small ESS suggests that our data are close to their cluster means, implying that the units in each cluster have high similarities.

Between Sum of Squares: $BSS = \sum_i \sum_j (X_{i.} - \bar{X}_{..})^2$. Here we are comparing the cluster mean and the

overall mean.

Total Sum of Squares: $TSS = \sum_i \sum_j (X_{ij} - \bar{X}_{..})^2$. TSS measures the total variation presented in the

data, which is the sum of ESS and BSS.

The homogeneity Measure chosen in this paper is the percentage of within variance as of the total variance (%variance), which is defined as error sum of squares / total sum of squares. So if the %variance is small, then the territories have high similarities. Let's consider two extremes:

Each individual zip code as its own cluster, i.e., if there are N zips, there will be N clusters. The ESS would be 0 since the individual zip level value is the same as its cluster mean, which is the zip code level value. So the homogeneity measure would be 0. This is the most homogeneous case.

All zips are combined into one cluster. The ESS would be the same as the TSS according the definition. So the homogeneity measure would be 1 in this case. This is the least homogeneous situation.

Therefore the percent variance is able to reasonably measure the homogeneity of the territory definition.

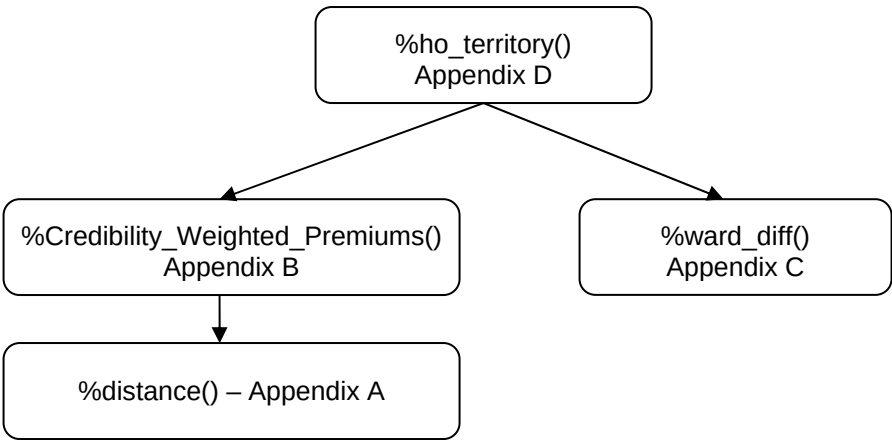
Using Ward's Method we will start out with all zip codes in N clusters of size 1 each. In the first step of the algorithm, $n - 1$ clusters are formed, one of size two and the remaining of size 1. The ESS is then computed. The pair of zip codes that yield the smallest ESS thus the smallest %Variance will form the first cluster. Then, in the second step of the algorithm, $N - 2$ clusters are formed from that $N - 1$ clusters defined in step 2. These may include two clusters of size 2, or a single cluster of size 3 including the two zip codes clustered in step 1. Again, the value of ESS is minimized. Thus, at each step of the algorithm clusters or observations are combined in such a way as to minimize the results of ESS. The algorithm stops when all zip codes are combined into a single large cluster of size N .

APPLICATION AND RESULTS

APPLICATION

Most part of the work is done in SAS via SAS macro programming. The SAS macro language is a very versatile and useful tool. It is especially useful if the same code will be run multiple times. It facilitates passing information from one procedure to another. Furthermore, it can also be used to write SAS programs that are dynamic and flexible. Using nested macros, one can better organize complicated SAS programs with SAS macro modules.

Two main jobs need to be done in this practice. First, credibility weighted pure premiums need to be calculated. Secondly, apply Ward's method on the weighted pure premiums to create contiguous grouping of zip codes. Four macro functions (shown in Appendix section) were written to do the job. The calling sequence of these macros is shown in Graph 1. The macro %distance () in Appendix A is called by and %credibility_weighted_premiums() in Appendix B to create credibility weighted pure premiums for each zip code. The macro %ward_diff () in Appendix C is designed to calculate the ward difference between two clusters. Both macros in Appendix B and C are called in the main macro %ho_territory() (Appendix D) to create homogeneous groupings of zip codes based on the Ward Method.



Graph 1 - Calling sequence of the macro functions

To calculate the credibility-weighted pure premiums, the first step is to determine the distance between zip codes. As discussed in the methodology section, the distance between any individual ZIP code and the remaining ZIPs need to be calculated to define the 6 difference radius groups. A macro function “distance” is defined in Appendix A, which requires five parameters. Long1 and lat1 are the longitude and latitude of the centroid of the first ZIP, Long2 and lat2 are the longitude and latitude of the centroid of the second ZIP, and R is the radius of the earth. By calling this macro function with the required parameters provided, one can calculate the distance between any two points on the earth.

The macro function defined in Appendix B - “credibility_weighted_premium” produces the credibility weighted pure premiums for each zip code. This macro function calls the macro in Appendix A and involves two parameters, “indata” and “outdata”. The typical source data shown in the Data and Methodology section can be used here as an input for “indata”. “outdata” can be any valid SAS dataset name. The output from calling the macro function “credibility_weighted_premium” is a SAS dataset similar to the one shown in Table 1, with one more column added, pp_wtd (the credibility weighted pure premium). The value of this variable for each ZIP code, along with the exposure and unique index, is saved in array of macro variables (resp1 – respN for the credibility weighted pure premium; weight1 – weightN for the exposure; index1 – indexN for the unique index). These values will be referred to in subsequent macro calls.

In Appendix C, another macro function is defined to calculate the Ward difference between two units. The codes in this macro will be used many times in the main macro “HO_TERRITORY” (called in do loops). By separating the codes into a macro function, one can not only reuse the same code multiple times, but also make the program

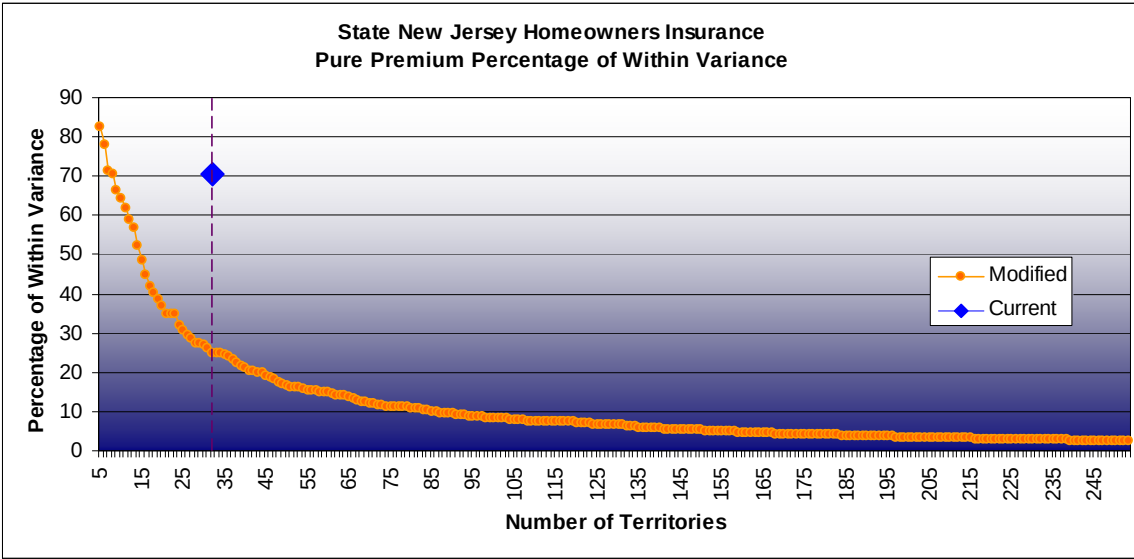
better organized and easier to read. This macro function has 4 input parameters, “n1”, “n2”, “n_index1”, “n_index2”. “n1” (“n2”) is a character string with the indexes for the zips in 1st (2nd) cluster and “n_index1” (“n_index2”) is the number of zips in the 1st (2nd) cluster.

The main macro function “HO_TERRITORY” is defined in Appendix D. It has three parameters, “indata”, “n_terr”, “limit”; “indata” is the same as that used in the macro in Appendix B. “n_terr” is the number of groupings that is desired to create. For example, if one would like to create 30 territories, then 30 should be provided for parameter “n_terr”. And extra parameter – “limit” is introduced to offer the ability to apply a constraint (maximum number of loss exposures in each grouping) when creating the groupings. One might find this useful if by chance one grouping grows too big.

RESULTS

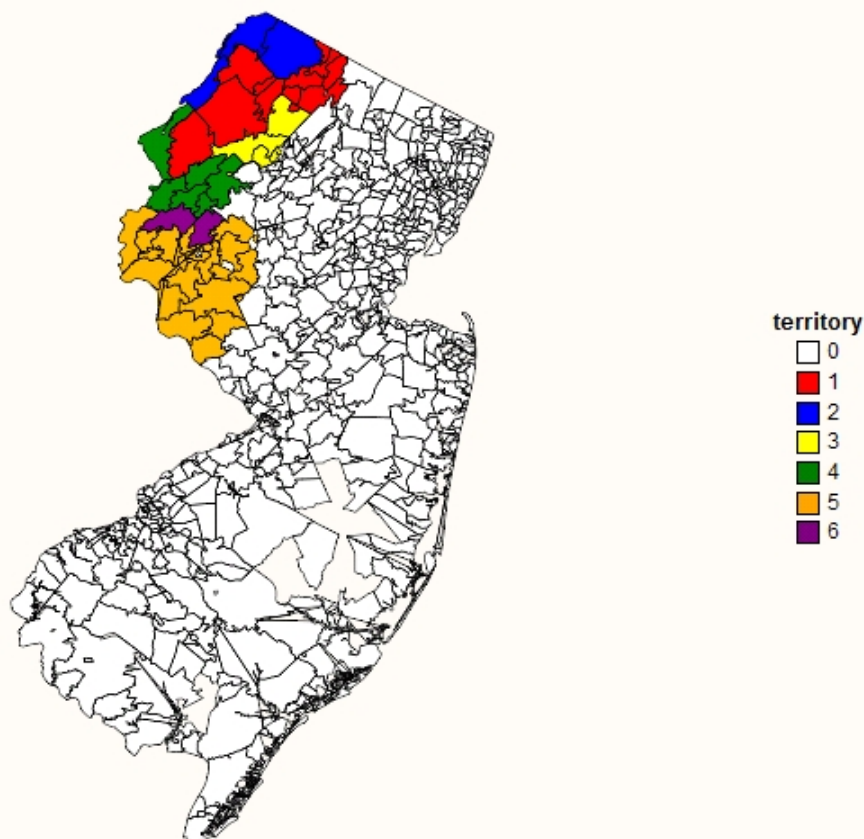
This methodology was applied to New Jersey homeowners experience. The results of this particular study are shown in Graph 1. The graph depicts the results for a current territory structure (labeled “Current”) and the modified territory structure (labeled “Modified”) using the method described in this paper. Before the modification, there are 32 territories for which the total %Variance is as high as 70%. From the graph, we realize that we are able to reduce the within cluster variance percentage to 25%, holding constant the number of territories at 32. Graph 1 also depicts the %Variance for different number of territories. As expected, by increasing the number of territories - thus increasing similarity within each territory - the %Variance falls.

The graph also helps the user to make the decision as to how many territories should be created based on the acceptable %Variance. For example, if a company would like to accept 10% of variance for its territory structure for a certain line of business, then about 80 territories should be defined. The number of territories will often depend on implementation concerns, including systems limitations, policyholder disruption, and the competitive landscape.



Graph 2 Comparison of %variance for the current and modified territory definitions

To avoid substantial rate disruption for all the policyholders, one might use this program to revisit the current territories. The program can tell the percentage of variance for each existing territory. A ranking from high to low of the percentage of variances for each territory provides useful guide for the analyst as to which territories need the most attention to modify first. The territories on the top of the list might need to be split. The program can then be run on that territory only to suggest meaningful splits. Graph 3 shows an example of split suggested by the program for one of the existing territories.



Graph 3 Example of split the current territory

CONCLUSION AND DISCUSSION

This paper has presented an application of one technique to define geographical rating territories using SAS macros. To supplement geographical units with sparse data, the principle of locality was incorporated to create the credibility weighted pure premium for each unit. Ward method is used to perform the cluster analysis. Detailed SAS codes were provided the Appendix.

The program included in this paper uses postal zip codes as the geographical units. This is not the only possibility however. The program can handle other geographical units such as county with little modification. Ward method is used to do the cluster analysis because it tends to create smaller clusters, which is a desired property here. Other cluster techniques can certainly be applied as well with appropriate adjustment.

The results shown in the paper suggest that the homogeneity measure (%Variance) could be significantly improved (from 70% to 25%). However one should keep in mind that the results were obtained under an ideal condition, i.e., no restrictions. In practice, considerations should be given to competitive concerns, rate disruption, etc. The final proposed territory boundaries could be different from the one resulted from the paper, which may lead to a reasonable increase in the homogeneity measure.

ACKNOWLEDGEMENTS

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Qunca Zou
 Research Statistician
 New Jersey Manufacturers Insurance Company
 301 Sullivan Way
 West Trenton, NJ 08628
 Work Phone: 609 883 1300 Ext. 7753
 Email: qzou@njm.com

Ryan Diehl, FCAS, MAAA
 Actuary
 New Jersey Manufacturers Insurance Company
 301 Sullivan Way
 West Trenton, NJ 08628
 Work Phone: 609 883 1300 Ext. 7758
 Email: rdiehl@njm.com

APPENDIX A – MACRO FOR CALCULATING THE DISTANCE BETWEEN THE CENTROIDS OF ANY TWO ZIP CODES

```
%macro distance(long1, lat1, long2, lat2, R);
  xx = 69.1 * (&lat2 - &lat1);
  yy = 69.1 * (&long2 - &long1) * cos(&lat1/57.3);
  distance = sqrt(xx * xx + yy * yy);
%mend distance;
```

APPENDIX B – MACRO FOR PRODUCING THE CREDIBILITY WEIGHTED PURE PREMIUMS FOR EACH ZIP CODE

```
%macro credibility_weighted_premium(indata, outdata);
data &outdata.;
```

```
set &indata.;
```

```
%do i = 1 %to &n_zip.;
  %distance(x, y, &&long&i., &&lat&i., 6371*0.621371);
  if distance <= 5 then do;
    sum_loss_5 + &&loss&i.;
    sum_exp_5 + &&exp&i.;
  end;
  if distance <= 10 then do;
    sum_loss_10 + &&loss&i.;
    sum_exp_10 + &&exp&i.;
  end;
  if distance <= 15 then do;
    sum_loss_15 + &&loss&i.;
    sum_exp_15 + &&exp&i.;
  end;
  if distance <= 20 then do;
    sum_loss_20 + &&loss&i.;
    sum_exp_20 + &&exp&i.;
  end;
  if distance <= 25 then do;
    sum_loss_25 + &&loss&i.;
    sum_exp_25 + &&exp&i.;
  end;
end;
```

```

end;
if distance <= 50 then do;
    sum_loss_50 + &&loss&i.;
    sum_exp_50 + &&exp&i.;
end;
sum_loss_50plus + &&loss&i.;
sum_exp_50plus + &&exp&i.;
%end;

pp5    = sum_loss_5 / sum_exp_5;
pp10   = sum_loss_10 / sum_exp_10;
pp15   = sum_loss_15 / sum_exp_15;
pp20   = sum_loss_20 / sum_exp_20;
pp25   = sum_loss_25 / sum_exp_25;
pp50   = sum_loss_50 / sum_exp_50;
average = sum_loss_50plus / sum_exp_50plus;

Z0 = &y.*&w. / (&y.*&w. + 2500000);
Z5 = sum_loss_5 / (sum_loss_5 + 2500000) - Z0;
Z10 = sum_loss_10 / (sum_loss_10 + 2500000) - sum_loss_5 / (sum_loss_5 + 2500000);
Z15 = sum_loss_15 / (sum_loss_15 + 2500000) - sum_loss_10 / (sum_loss_10 + 2500000);
Z20 = sum_loss_20 / (sum_loss_20 + 2500000) - sum_loss_15 / (sum_loss_15 + 2500000);
Z25 = sum_loss_25 / (sum_loss_25 + 2500000) - sum_loss_20 / (sum_loss_20 + 2500000);
Z50 = sum_loss_50 / (sum_loss_50 + 2500000) - sum_loss_25 / (sum_loss_25 + 2500000);
Z50plus = 1 - Z0 - Z5 - Z10 - Z15 - Z20 - Z25 - Z50;

&y_wtd = &y.*Z0 + pp5*Z5 + pp10*Z10 + pp15*Z15 + pp20*Z20 + pp25*Z25 + pp50*Z50 + average * Z50plus;

DROP Z: pp5 pp10 pp15 pp20 pp25 pp50 distance x y xx yy average sum_ ;
run;
%mend;

%credibility_weighted_premium (HO_TERRITORY, HO_TERRITORY_tmp);

```

APPENDIX C – MACRO FOR CALCULATING THE WARD DIFFERENCE BETWEEN TWO ZIP CODES

```

%macro ward_diff(n1,n2,n_index1, n_index2);

sum_wtd = 0;
sum_weight = 0;
diff_ward = 0;

do i = 1 to &n_index1.;
    if scan(&n1., i, "_") = "&ind1." then do;
        sum_wtd = sum_wtd + &resp1. * &weight1.;
        sum_weight = sum_weight + &weight1.;
    end;
    %do j = 2 %to &N_ZIP.;
        else if scan(&n1., i, "_") = "&&ind&j." then do;
            sum_wtd = sum_wtd + &&resp&j. * &&weight&j.;
            sum_weight = sum_weight + &&weight&j.;
        end;
    %end;
end;
do i = 1 to &n_index2.;
    if scan(&n2., i, "_") = "&ind1." then do;
        sum_wtd = sum_wtd + &resp1. * &weight1.;
        sum_weight = sum_weight + &weight1.;
    end;
    %do j = 2 %to &N_ZIP.;
        else if scan(&n2., i, "_") = "&&ind&j." then do;
            sum_wtd = sum_wtd + &&resp&j. * &&weight&j.;
            sum_weight = sum_weight + &&weight&j.;
        end;
    %end;
end;

```

```

    %end;
end;

mean_wtd = sum_wtd / sum_weight;

do i = 1 to &n_index1.;
    if scan(&n1., i, " ") = "&ind1." then diff_ward = diff_ward + (&resp1. - mean_wtd)**2 * &weight1.;
    %do j = 2 %to &N_ZIP.;
        else if scan(&n1., i, " ") = "&&ind&j." then diff_ward = diff_ward + (&&resp&j. - mean_wtd)**2 * &&weight&j.;
    %end;
end;
do i = 1 to &n_index2.;
    if scan(&n2., i, " ") = "&ind1." then diff_ward = diff_ward + (&resp1. - mean_wtd)**2 * &weight1.;
    %do j = 2 %to &N_ZIP.;
        else if scan(&n2., i, " ") = "&&ind&j." then diff_ward = diff_ward + (&&resp&j. - mean_wtd)**2 * &&weight&j.;
    %end;
end;
%mend ward_diff;

```

APPENDIX D – THE MAIN MACRO FOR CREATING CONTIGUOUS GROUPING OF ZIP CODES BASED ON THE CREDIBILITY WEIGHTED PURE PREMIUMS OF EACH ZIP CODE

```

%macro ho_territory(indata, n_terr, limit);

%let flag_stop = ;
%let temp_data = HO_TERRITORY_1;

/*create the macro variables for the macro function %credibility_weighted_premium() */
data _null_;
set &indata. end = eof;
call symput("code"||strip(_n_), code);
call symput("long"||strip(_n_), x);
call symput("lat"||strip(_n_), y);
call symput("loss"||strip(_n_), &w.*&y.);
call symput("exp"||strip(_n_), &w.);
if eof then call symput("N_ZIP", _N_);
run;

/*Call %credibility_weighted_premium() to create the credibility weighted pure premium*/
%credibility_weighted_premium(&indata., HO_TERRITORY_tmp);

/*create the macro variables for calculating the ward difference for two neighboring units*/
DATA _NULL_;
SET HO_TERRITORY_TMP;
CALL SYMPUT ("RESP"||STRIP(_n_), &y_wtd.);
CALL SYMPUT ("WEIGHT"||STRIP(_N_), &w.);
CALL SYMPUT ("IND"||STRIP(_N_), strip(index));
RUN;

/* create pairs of adjacent ZIP codes;*/
data HO_TERRITORY_tmp_1;
set HO_TERRITORY_tmp(rename = (index = index1));

array nn(*) n1 - n16;
length index2 $ 2000;

do i = 1 to dim(nn);
    if nn(i) ^= . then do; index2 = nn(i); output; end;
end;
keep index1 index2 &y_wtd. &w.;
run;

proc sort data = HO_TERRITORY_tmp_1; by index2;run;

```

```

proc sort data = HO_TERRITORY_tmp; by index; run;

data HO_TERRITORY_paired;
merge HO_TERRITORY_tmp_1(rename = (&y_wtd = resp1 &w. = weight1))
  HO_TERRITORY_tmp(rename = (&y_wtd. = resp2 index = index2 &w. = weight2));
n_index1 = 1; n_index2 = 1;
by index2;
IF resp2 > 0 and resp1 > 0;
drop resp2;
run;

/*Calculate the ward difference for each paired zip code*/
data &temp_data.;
set HO_TERRITORY_paired end = eof;
%ward_diff(index1, index2, n_index1, n_index2);
keep index1 index2 n_index1 n_index2 diff_ward weight1 weight2;
run;

/*Save the number of territories in a macro variable: tmp_n_terr - will be used in the subsequent %do loop */
proc sql noprint;
select (count(distinct(index1))) into: tmp_n_terr
from &temp_data.;
quit;

data sum_squares; run;

%do %while(&tmp_n_terr. > &n_terr and &flag_stop. ne stop);

  proc sort data = &temp_data. out = tmp_&temp_data. nodup; by diff_ward index1; run;

  /*Get the indexes of the two zip with the minimum ward difference*/
  data _NULL_;
  set tmp_&temp_data. end = eof;

  retain count 0 flag_first 0;
  length index_new $ 2000;

  if flag_first = 0 then do;
    if weight1 + weight2 <= &limit. then do;
      count + 1;
      if index1 < index2 then do;
        index_new = cats(strip(index1), "_", strip(index2));
        weight_new = weight1 + weight2;
      end;
    else do;
      index_new = cats(strip(index2), "_", strip(index1));
      weight_new = weight1 + weight2;
    end;

    call symput("index"|strip(count), strip(index1));
    call symput("index_new", strip(index_new));
    call symput("weight_new", weight_new);

    if count = 2 then flag_first = 1;
  end;
  if eof then do;
    flag_stop = "stop";
    call symput("flag_stop", flag_stop);
    stop;
  end;
end;
else do;
  if count = 0 then do;
    flag_stop = "stop";
    call symput("flag_stop", flag_stop);
  end;
end;

```

```

stop;
end;
run;

data &temp_data.;
set tmp_&temp_data.;

if strip(index1) = strip("&index1.") or strip(index1) = strip("&index2.") then do;
    index1 = strip("&index_new.");
    n_index1 = count(strip(index1), "_") + 1;
    weight1 = &weight_new.;
    %%ward_diff(index1, index2, n_index1, n_index2);
end;
if strip(index2) = strip("&index1.") or strip(index2) = strip("&index2.") then do;
    index2 = strip("&index_new.");
    n_index2 = count(strip(index2), "_") + 1;
    weight2 = &weight_new.;
    %%ward_diff(index1, index2, n_index1, n_index2);
end;

if strip(index1) = strip(index2) then delete;
keep index1 index2 n_index1 n_index2 weight1 weight2 diff_ward;
run;

proc sql noprint;
select (count(distinct(index1))) into: tmp_n_terr
from &temp_data.;
quit;

/*Get the %within variance*/
data temp;
set HO_TERRITORY_1;
do i = 1 to n_index1;
    index = scan(strip(index1), i, "_"); output;
end;
keep index index1 ;
run;

proc sql noprint;
create table temp1 as
select distinct a.*, b.code, b.&y., b.&w.
from temp as a, HO_TERRITORY_tmp as b
where a.index = b.index
order by index1;
quit;

data temp1; set temp1;
retain territory 0;
by index1;
if first.index1 then do; territory + 1; end;
run;

proc glm data = temp1;
class territory;
model &y. = territory;
weight &w.;
ods output OverallAnova = tmp_Sum_Squares(keep = Source DF SS);
run;

proc sort data = tmp_sum_squares; by descending DF; run;

data tmp_sum_squares;
set tmp_sum_squares;
retain SS_TOTAL 0;
if _n_ = 1 then SS_TOTAL = SS;
pct_variation = round(SS / SS_TOTAL * 100, 0.01);

```

```
N_Territory = &tmp_n_terr.;  
if source = "Error";  
run;  
  
data sum_squares;  
set sum_squares tmp_sum_squares;  
run;  
%end;  
%mend ho_territory;
```

REFERENCE

¹ Philip J. Jennings, Using Cluster Analysis to Define Geographical Rating Territories, Casualty Actuarial Society, 2008 Discussion Paper Program.