```
*;
*;
*     King County House Sales Cluster Analysis - Cluster Analysis;
*;
**** NOTE: Several Temporary Datasets are Reused/Replaced (Same Name) ;
*     in this analysis for the 4-Cluster and 3-Cluster Non-Hierarchical
Analysis ;
*;
*;
*ods graphics on;
*;
*options ls=80 ps=50 nodate pageno=1;
*;
ods pdf file="\\Mac\Home\Downloads\KCHouseSales_ClusterAnalysis_Final.pdf";
*;
Title 'King County House Sales Cluster Analysis';
*;
* Input King County House Sales ;
*;
proc import
datafile="\\Mac\Home\Downloads\kc_house_data_1K_new_LotPriceOutliersRemoved.c
sv"
        out=kchousesales1K
        dbms=csv
        replace;

     *getnames=no;
run;
*;
* Select Variables ID sqft_living sqft_lot Age_At_Sale;
*;
Data kchousesales1K3Var;
      Set kchousesales1K (Keep = ID sqft_living sqft_lot Age_At_Sale);
*;
*Proc Print Data = kchousesales1K3Var;
*;
* Compute Variable Means;
*;
Proc Means Data = kchousesales1K3Var;
      Var sqft_living sqft_lot Age_At_Sale;
      Output Out = Meanskchousesales1K
                   Mean(sqft_living sqft_lot Age_At_Sale) = Meansqft_living
Meansqft_lot MeanAge_At_Sale;
*;
Proc Print Data = Meanskchousesales1K;
*;
* Merge kchousesales1K Data with kchousesales1K Means ;
*;
Data Meanskchousesales1K;
      Set Meanskchousesales1K (Drop = _TYPE_ _FREQ_);
*;
Data kchousesales1KMeans;
      Retain ID sqft_living sqft_lot Age_At_Sale;
      If _N_ = 1 Then Set Meanskchousesales1K;
      Set kchousesales1K;
*;
* Compute Centered kchousesales1K Variables (Subtract Means) ;
```

```
*;
*     PriceC = Price - MeanPrice;
      sqft_livingC = sqft_living - Meansqft_living;
      sqft_lotC = sqft_lot - Meansqft_lot;
      Age_At_SaleC = Age_At_Sale - MeanAge_At_Sale;
*     X18C = X18 - MeanX18;
*;
* Compute Squared Centered kchousesales1K Variables ;
*;
*     PriceCSQR = PriceC ** 2;
      sqft_livingCSQR = sqft_livingC ** 2;
      sqft_lotCSQR = sqft_lotC ** 2;
      Age_At_SaleCSQR = Age_At_SaleC ** 2;
*     X18CSQR = X18C ** 2;
*;
* Compute Totaled Squared Centered kchousesales1K Variables ;
*;
      TotDiffSqr = Sum(sqft_livingCSQR,sqft_lotCSQR,Age_At_SaleCSQR);
*;
* Compute kchousesales1K Variables Dissimalrities (Square Root of Total);
*;
      SqrRootTot = TotDiffSqr ** 0.5;
*;
* Rank the kchousesales1K Variables Dissimalrities ;
*;
Proc Sort Data = kchousesales1KMeans;
      By Descending SqrRootTot;
*;
Proc Print Data = kchousesales1KMeans;
*;
****** Select 10 Largest HBAT Variables Dissimalrities *****;
*;
Data kchousesales1KMeans10;
      Set kchousesales1KMeans (Keep = sqft_livingC sqft_lotC Age_At_SaleC
sqft_livingCSQR sqft_lotCSQR Age_At_SaleCSQR SqrRootToT);
      If _N_ LE 10;
*;
Proc Print Data = kchousesales1KMeans10;
*;
* SAS Hierarchical Cluster Analysis ;
*;
* The PROC CLUSTER statement starts the CLUSTER procedure, specifies a
clustering method, and
      optionally specifies details for clustering methods, data sets, data
processing, and displayed output.
*;
* The METHOD = specification determines the clustering method used by the
procedure. Any one of
      the 11 methods can be specified for name;
*;
*     WARD | WAR requests Ward's minimum-variance method (error sum of
squares, trace W).
            Distance data are squared unless you specify the NOSQUARE option.
            To reduce distortion by outliers, the TRIM= option is recommended.
            See the NONORM option.;
*;
```

```
*     NONORM - prevents the distances from being normalized to unit mean or
unit root mean square with
               most methods. With METHOD=WARD, the NONORM option prevents the
between-cluster
               sum of squares from being normalized by the total sum of
squares to yield a squared semipartial
               correlation;
*;
*     SIMPLE | S - displays means, standard deviations, skewness, kurtosis,
and a coefficient of bimodality. The
                    SIMPLE option applies only to coordinate data.;
*;
*     CCC - displays the cubic clustering criterion and approximate expected
R square under the uniform
            null hypothesis. The statistics associated with the RSQUARE
option, R square
            and semipartial R square, are also displayed. The CCC option
applies only to coordinate
            data. The CCC option is not appropriate with METHOD=SINGLE
because of the method's
            tendency to chop off tails of distributions. Computation of the
CCC requires the eigenvalues
            of the covariance matrix. If the number of variables is large,
computing the eigenvalues
            requires much computer time and memory.;
*;
*     PSEUDO - displays pseudo F and t2 statistics. This option is effective
only when the data are coordinates
               or when METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD is
specified.;
*;
*     RMSSTD - displays the root mean square standard deviation of each
cluster. This option is effective only
               when the data are coordinates or when METHOD=AVERAGE,
METHOD=CENTROID, or
               METHOD=WARD is specified.;
*;
*     RSQUARE | RSQ - displays the R square and semipartial R square. This
option is effective only when the data
                     are coordinates or when METHOD=AVERAGE or
METHOD=CENTROID is specified. The
                     R square and semipartial R square statistics are always
displayed with METHOD=WARD.;
*;
*     OUTTREE = SAS-data-set
              - creates an output data set that can be used by the TREE
procedure to draw a tree diagram. You
                must give the data set a two-level name to save it. See SAS
Language Reference: Concepts
                for a discussion of permanent data sets. If you omit the
OUTTREE= option, the data set is
                named by using the DATAn convention and is not permanently
saved. If you do not want to
                create an output data set, use OUTTREE=_NULL_.;
*;
*;
```

```
*Proc Cluster Data=kchousesales1KMeans Method=Ward NoNorm Simple CCC Pseudo
RmsStd RSquare OutTree=Tree;
*     Var sqft_living sqft_lot Age_At_Sale;
*;
* Plot the Dendrogram;
*;
*Proc Tree Data=Tree;
*;
* Remove Identified Outliers: Observations 6 and 87;
*;
/*Data HBATMeans98Obs;
      Set HBATMeans;
      If ID EQ 6 Then Delete;
      If ID EQ 87 Then Delete;
*;
***** SAS Hierarchical Cluster Analysis *****;
*;
Proc Cluster Data=HBATMeans98Obs Method=Ward NoNorm Simple CCC Pseudo RmsStd
RSquare OutTree=Tree;
      Var X6 X8 X12 X15 X18;
*;
* Plot the Dendrogram;
*;
Proc Tree Data=Tree;
*;
*;
*;
***** SAS Non-Hierarchical 4-Cluster Analysis *****;
*;
* The FASTCLUS procedure performs a disjoint cluster analysis on the basis of
distances computed
               from one or more quantitative variables. The observations are
divided into clusters such that every
               observation belongs to one and only one cluster, the clusters
do not form a tree structure as they do
               in the CLUSTER procedure.;
*;
* The FASTCLUS procedure combines an effective method for finding initial
clusters with a standard
               iterative algorithm for minimizing the sum of squared
distances from the cluster means.
               The result is an efficient procedure for disjoint clustering
of large data sets.;
*;
*     RADIUS = t R=t
             - establishes the minimum distance criterion for selecting new
seeds. No observation is considered
             as a new seed unless its minimum distance to previous seeds
exceeds the value given
             by the RADIUS= option. The default value is 0. If you specify
the REPLACE=RANDOM
             option, the RADIUS= option is ignored.;
*;
*     RANDOM = n
          - specifies a positive integer as a starting value for the pseudo-
random number generator for
```

```
            use with REPLACE=RANDOM. If you do not specify the RANDOM=
option, the time of
            day is used to initialize the pseudo-random number sequence.
            REPLACE = FULL | PART | NONE | RANDOM
                                specifies how seed replacement is performed, as
follows:
                        FULL requests default seed replacement
                        PART requests seed replacement only when the distance
between the observation
                                and the closest seed is greater than the minimum
distance between seeds.
                        NONE suppresses seed replacement.
                        RANDOM selects a simple pseudo-random sample of
complete observations as initial
                                cluster seeds.;
*;
*     MAXCLUSTERS = n MAXC = n
               - specifies the maximum number of clusters permitted. If you
omit the MAXCLUSTERS=
                 option, a value of 100 is assumed.;
*;
*     MAXITER = n
          - specifies the maximum number of iterations for recomputing
cluster seeds.;
*;
*     LIST - lists all observations, giving the value of the ID variable (if
any), the number of the cluster
          to which the observation is assigned, and the distance between the
observation and the final
          cluster seed.;
*;
*     DISTANCE | DIST - computes distances between the cluster means.;
*;
*     OUT = SAS-data-set
          - creates an output data set to contain all the original data, plus
the new variables CLUSTER and
            DISTANCE.;
*;*/
*Proc FastClus Data=HBATMeans98Obs Radius=0 Replace=Random MaxClusters=4
Maxiter=20 List Distance Out=Clust;
*     Var X6 X8 X12 X15 X18;
Proc FastClus Data=kchousesales1KMeans Radius=0 Replace=Random MaxClusters=3
Maxiter=20 List Distance Out=Clust;
      Var sqft_living sqft_lot Age_At_Sale;
*;
* Plot 5-Cluster Obs Membership with X-Y Variable Scatterplots ;
*;
Proc Print Data = Clust;
*;
Proc Sgplot Data = Clust;
      Scatter X = sqft_living Y = sqft_lot / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = sqft_living Y = Age_At_Sale / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = sqft_lot Y = Age_At_Sale / Group=Cluster ;
```

```
*;
/*Proc Sgplot Data = Clust;
      Scatter X = X6 Y = X18 / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = X8 Y = X12 / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = X8 Y = X15 / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = X8 Y = X18 / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = X12 Y = X15 / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = X12 Y = X18 / Group=Cluster ;
*;
Proc Sgplot Data = Clust;
      Scatter X = X15 Y = X18 / Group=Cluster ;*/
*;
***** Validation and Profiling the 3-Clusters *****;
*;
* Merge Cluster Assignments with Original King County Data By ID (with
Outliers Removed);
*;
* Select Variables ID Price Bedrooms Bathrooms Floors sqft_living15
sqft_lot15;
*;
Data kchousesales1K3;
*     Set kchousesales1K (Keep = ID Price Bedrooms Bathrooms Floors
sqft_living15 sqft_lot15);
    Set kchousesales1K (Keep = ID Price sqft_living15 sqft_lot15);
*     If ID EQ 6 Then Delete;
*     If ID EQ 87 Then Delete;
*;
Proc Sort Data = kchousesales1K3;
      By ID;
Proc Sort Data = Clust;
      By ID;
Data kchousesales1K3Clust (Keep = ID Price sqft_living15 sqft_lot15 Cluster);
      Merge kchousesales1K3 Clust;
      By ID;
Proc Print Data = kchousesales1K3Clust;
*;
***** Assessing 3-Cluster Criterion Validity *****;
*;
* GLM MANOVA Analysis ;
*;
Proc GLM Data = kchousesales1K3Clust;
    Class Cluster;
      Model Price sqft_living15 sqft_lot15 = Cluster;
      Means Cluster / Scheffe Tukey LSD SNK Duncan;
      Means Cluster / Hovtest = Levene Hovtest = bf Hovtest = Bartlett;
      Means Cluster;
      Manova H = Cluster / MStat = Exact;
```

```
*;
***** Profiling the Final 3-Cluster Solution *****;
*;
* Merge Cluster Assignments with Original King County Data By ID (with
Outliers Removed);
*;
* Select Variables ID View Condition Grade Zipcode;
*;
Data kchousesales1K6;
      Set kchousesales1K (Keep = ID Bedrooms Bathrooms Floors View Condition
Grade);
*     If ID EQ 6 Then Delete;
*     If ID EQ 87 Then Delete;
*;
Proc Sort Data = kchousesales1K6;
      By ID;
Proc Sort Data = Clust;
      By ID;
Data kchousesales1K6Clust (Keep = ID Bedrooms Bathrooms Floors View Condition
Grade Cluster);
      Merge kchousesales1K6 Clust;
      By ID;
Proc Print Data = kchousesales1K6Clust;
*;
***** Cross-Classification of Clusters on View Condition Grade Zipcode *****;
*;
Proc Freq Data = kchousesales1K6Clust;
      Table Cluster * View;
*;
Proc Freq Data = kchousesales1K6Clust;
      Table Cluster * Condition;
*;
Proc Freq Data = kchousesales1K6Clust;
      Table Cluster * Grade;
*;
Proc Freq Data = kchousesales1K6Clust;
      Table Cluster * Bedrooms;
*;
Proc Freq Data = kchousesales1K6Clust;
      Table Cluster * Bathrooms;
*;
Proc Freq Data = kchousesales1K6Clust;
      Table Cluster * Floors;
*;
*Proc Freq Data = HBAT5Clust;
*     Table Cluster * X5;
*;
*;
Run;
Quit;
*;
ods pdf close;
```