**The University of Texas at Dallas**

**School of Management**

**Syam Menon Spring 2016**

**MIS 6334 Advanced Business Intelligence**

# Homework 03

**Group 14**

Abhimanyu Pandit -axp150330

AmeyaModak - axm152430

AmoghMarathe - axm153630

Dhanesh Shirodkar -dcs140330

Sagar Ghan - ssg150230

**Questions: Part II (PROC CLUSTER, PROC TREE)**

1. The dataset wolves was created using the following code (ignoring data), and contains measurements on skulls of wolves:

DATA wolves;

LENGTH location $2 wolf $5 sex $1;

INPUT location $ wolf $ sex $ x1-x9;

subject=\_n\_;

LABEL

X1 = 'palatal length'

X2 = 'postpalatal length'

X3 = 'zygomatic width'

X4 = 'palatal width-1'

X5 = 'palatal width-2'

X6 = 'postg foramina width'

X7 = 'interorbital width'

X8 = 'braincase width'

X9 = 'crown length';

1. **Use PROC FASTCLUS on this data set to cluster the data.**

k-means algorithm is used by PROC FASTCLUS to segment the observations into clusters. The disadvantage of k-means is that the user has to determine number of clusters without knowing the actual optimal number of cluster.

To determine the optimum number of clusters, SAS outputs a metric called CCC (cubic cluster criterion). To do this, begin with k=1 and proceed to k = 1/10th of the number of observations. The highest number of CCC will be the optimum number of clusters.

|  |  |
| --- | --- |
| K | CCC |
| 1 | 0 |
| 2 | 10.349 |
| 3 | 9.934 |
| 4 | 7.382 |
| 5 | 5.785 |

From this table, we can see that CCC increases as the number of clusters increases. Here, the highest is observed at k=2 and then it goes on decreasing. Hence, the optimum number of clusters is 2.

**Code:**

LIBNAME MIS6334 ‘D:\Study Material\Spring 16\ABI\Assignments';

K-means segments the data by calculating the distance of an observation with the centroid of every other cluster formed. Euclidian distance is used. Hence, standardizing the data gives better results.

**PROC** **STANDARD** DATA = MIS6334.WOLVES

MEAN = **0** STD = **1** OUT = STANDARD;

VAR X1-X9;

**RUN**;

**PROC** **FASTCLUS** DATA = STANDARD MAXC = **2**

MAXITER = **10** OUT = MIS6334.CLUS;

VAR X1-X9;

**RUN**;

**PROC IMPORT DATAFILE= "/folders/myfolders/catalog.xls" DBMS=xls OUT= catalogtesting REPLACE;**

**SHEET="testing";**

**GETNAMES=YES;**

**MIXED=NO;**

**RUN;**

**PROC CONTENTS DATA = catalogtesting;**

**TITLE "PROC CONTENTS";**

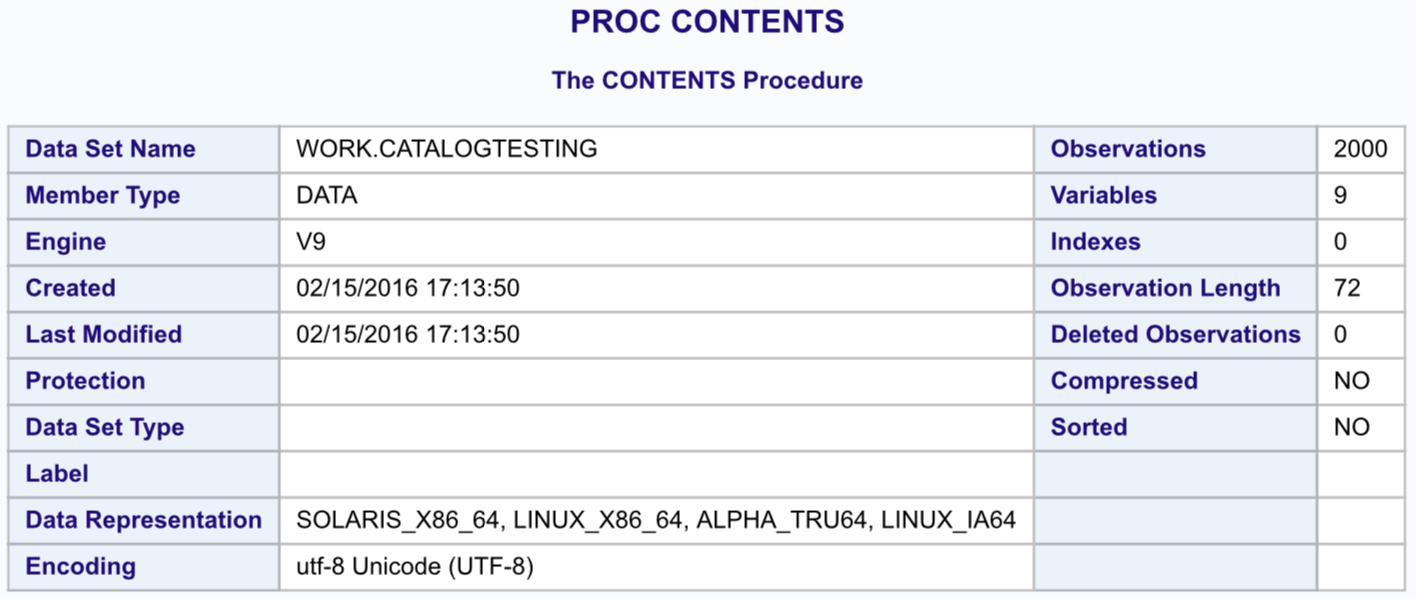
**RUN;**

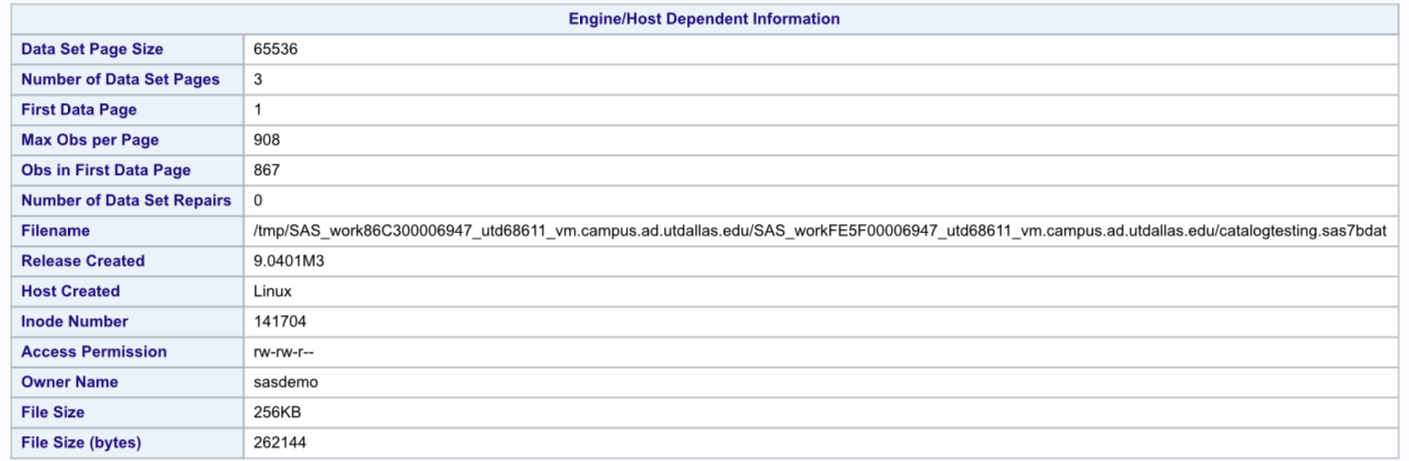
**PROC PRINT DATA = catalogtesting(obs=10) ;**

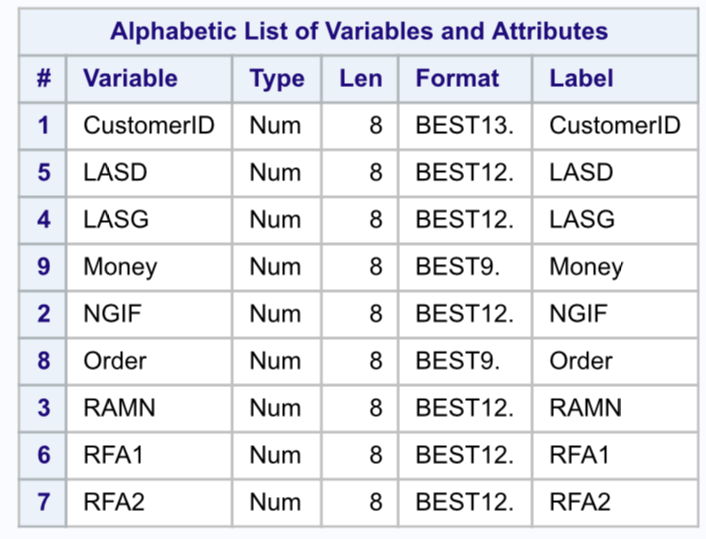
**TITLE "Dataset Imported from Excel";**

**RUN;**

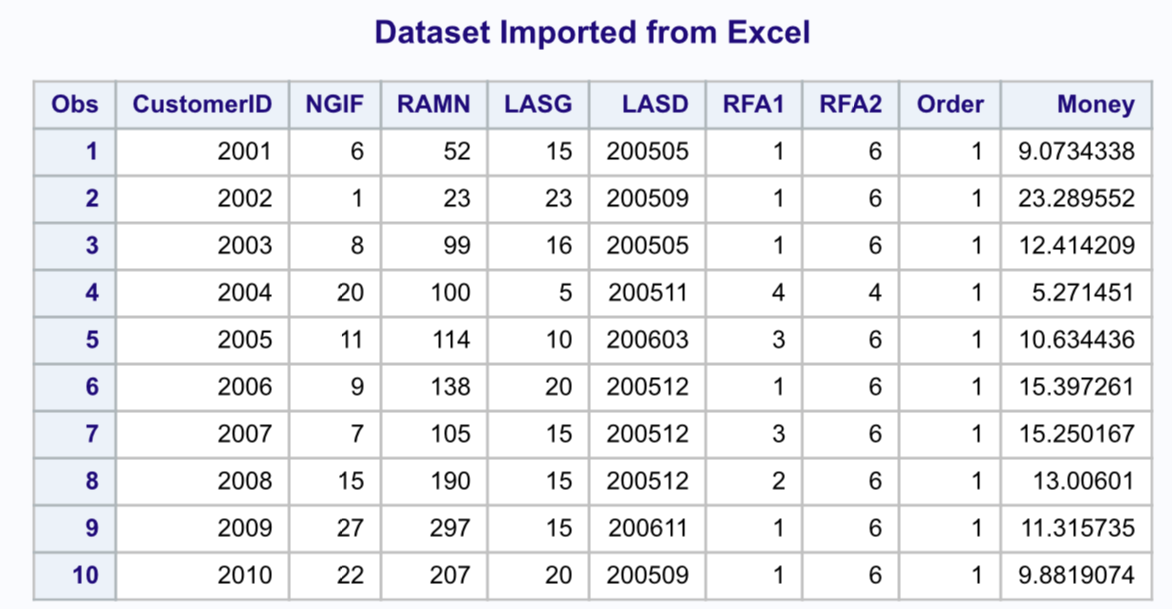
OUTPUT:







The first 10 observations:



1. Create a new variable to convert “money” into a binary variable that takes only 0 or 1 as values. Name the new variable “prospect” which indicates if the customer is potentially a heavy buyer (>20). Let the value be 1 if the customer orders more than $20 and 0 otherwise. (Note: use ‘if’ for this question).

**PROC IMPORT DATAFILE="/folders/myfolders/catalog.xls" DBMS=xlsOUT=catalogtesting REPLACE;**

**SHEET="testing";**

**GETNAMES=YES;**

**MIXED=NO;**

**RUN;**

**DATA catalogtesting2;**

**SET catalogtesting;**

**IF Money > 20 THEN**

**Prospect=1;**

**ELSE**

**Prospect=0;**

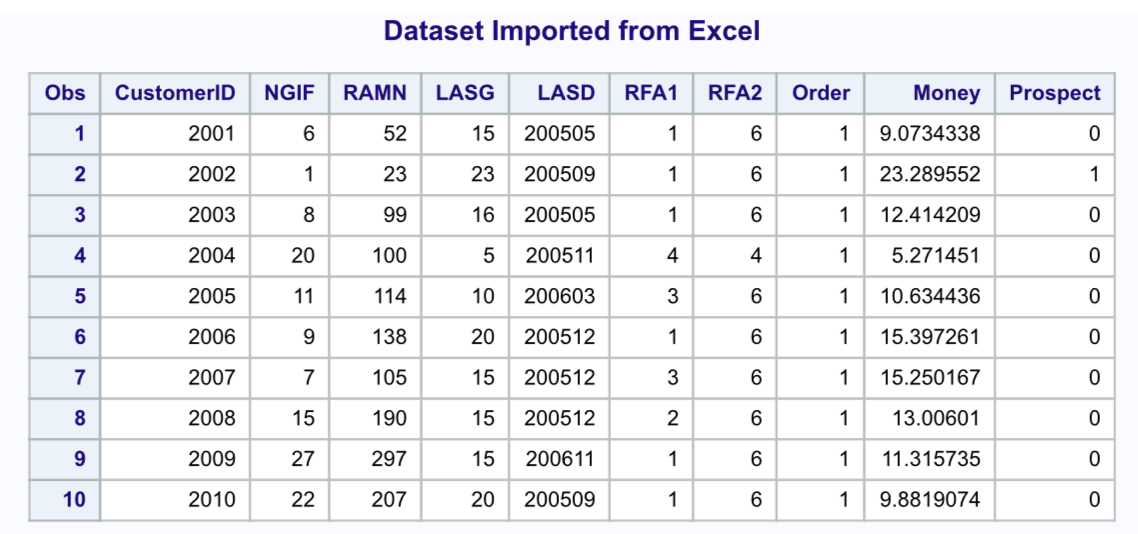
**RUN;**

**PROC PRINT DATA=catalogtesting2(obs=10);**

**TITLE "Data Imported from Excel";**

**RUN;**

OUTPUT:



1. Use ‘where’ to achieve the same as the previous question. Feel free to use multiple data steps if you can’t get it done in on data step.

**PROCIMPORT DATAFILE= "/folders/myfolders/catalog.xls" DBMS=xls OUT= catalogtesting REPLACE;**

**SHEET="testing";**

**GETNAMES=YES;**

**MIXED=NO;**

**RUN;**

**DATA catalogtesting2;**

**SET catalogtesting;**

**prospect = 0 ;**

**where Money >20;**

**prospect = 1 ;**

**RUN;**

**DATA catalogtesting3;**

**SET catalogtesting;**

**prospect = 0 ;**

**where Money <= 20;**

**prospect = 0 ;**

**RUN;**

**DATA catalogtesting4;**

**merge catalogtesting2 catalogtesting3;**

**By CustomerID;**

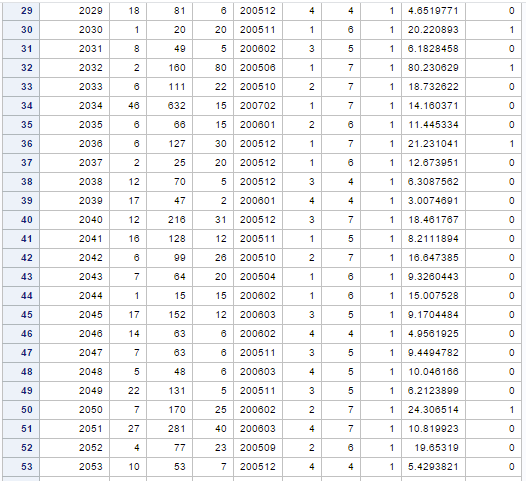
**RUN;**

**PROCPRINT DATA = catalogtesting4;**

**TITLE "Data Imported from Excel";**

**RUN;**

OUTPUT:



1. For the heavy buyers, compute the total number orders placed within the past 24 months (i.e., NGIF). Try RETAIN and SUM.

**PROC IMPORT DATAFILE="/folders/myfolders/catalog.xls" DBMS=xls OUT=catalogtesting REPLACE;**

**SHEET="testing";**

**GETNAMES=YES;**

**MIXED=NO;**

**RUN;**

**DATA catalogtesting2;**

**SET catalogtesting;**

**WHERE Money > 20;**

**RETAIN TotalOrders 0;**

**TotalOrders=SUM(TotalOrders, NGIF);**

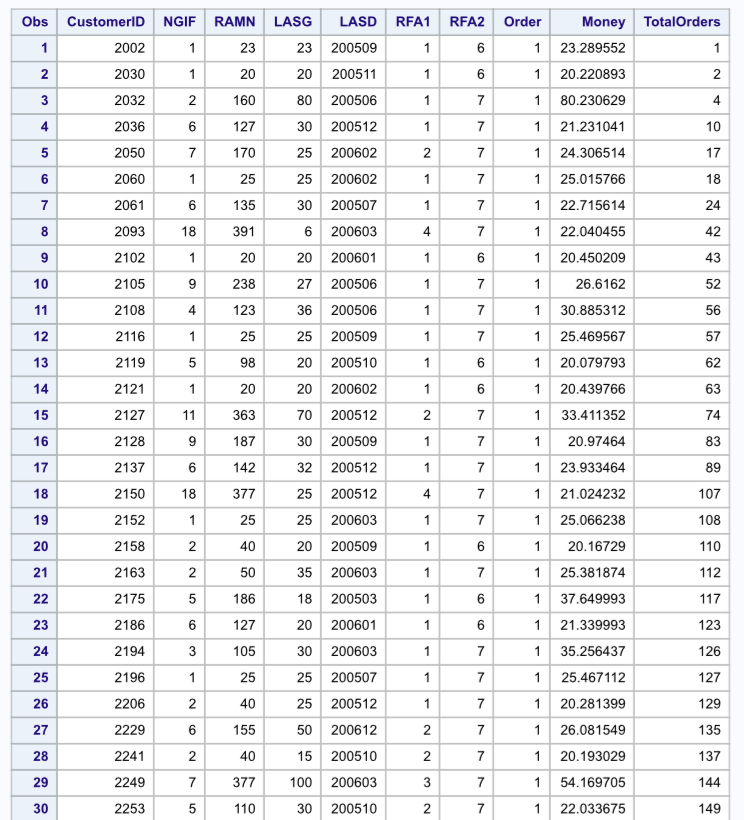
**RUN;**

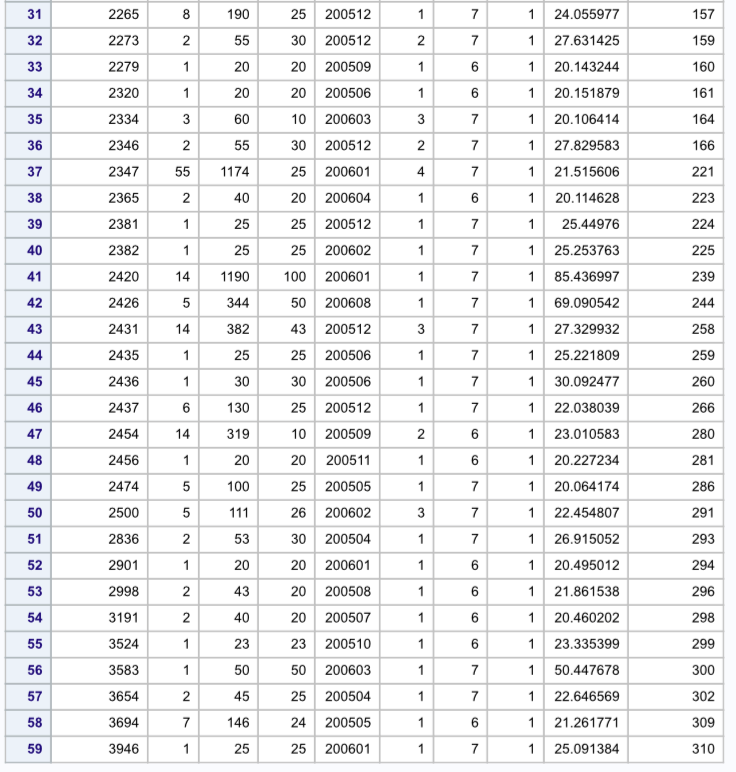
**PROC PRINT DATA=catalogtesting2;**

**TITLE "Data Imported from Excel";**

**RUN;**

OUTPUT:





1. For each RFA1 level (i.e., frequency of orders), compute the average order amount per customer (total RAMN / total number of customers at this level). Use an Array to store the value for each level of RFA1.

**PROC IMPORT DATAFILE= "/folders/myfolders/catalog.xls" DBMS=xls OUT= catalogtesting REPLACE;**

**SHEET="testing";**

**GETNAMES=YES;**

**MIXED=NO;**

**RUN;**

**DATA catalogtesting2 (DROP = i);**

**SET catalogtesting ;**

**ARRAY totalRAMN{4} totalRMAN1 totalRMAN2 totalRMAN3 totalRMAN4;**

**ARRAY countt{4} count1 count2 count3 count4;**

**ARRAY RFA{4} RFA11 RFA12 RFA13 RFA14;**

**DO i = 1 to 4;**

**RETAIN totalRAMN 0;**

**RETAIN countt 0;**

**IF RFA1 = i THEN totalRAMN{i} = totalRAMN{i} + RAMN;**

**IF RFA1 = i THEN countt{i} = countt{i} + 1;**

**RFA{i} = totalRAMN{i} / countt{i};**

**END;**

**RUN;**

**PROC PRINT DATA = catalogtesting2(FIRSTOBS = 2000);**

**TITLE 'Average order among per customer based on RFA1 Level';**

**RUN;**

OUTPUT:



1. Create a new dataset that keeps only the observations with more or equal to 2 orders in the last 24 months (NGIF). Then sort the new data on NGIF and money, output the result to a permanent Tab-delimited text file. Keep only four variables: UserID, NGIF, money and order. Try to use “libname” to create a library and use “file” to output variables to a file to the library. Print out the first 10 observations to the final data.

**PROC IMPORT DATAFILE= "/folders/myfolders/catalog.xls" DBMS=xls OUT= catalogtesting REPLACE;**

**SHEET="testing";**

**GETNAMES=YES;**

**MIXED=NO;**

**RUN;**

**LIBNAME ABI '/folders/myfolders' ;**

**DATA ABI.catalogtesting2;**

**SET catalogtesting;**

**where NGIF >= 2;**

**keepCustomerID NGIF Money Order;**

**RUN;**

**PROC SORT DATA=ABI.catalogtesting2 OUT=ABI.sortedOutput;**

**key NGIF;**

**key Money;**

**RUN;**

**DATA \_null\_ ;**

**SET ABI.sortedOutput;**

**FILE '/folders/myfolders/rawfile.txt' DLM="09"X ;**

**DBMS=tab;**

**PUT CustomerID NGIF Money Order;**

**RUN;**

**PROC PRINT DATA = ABI.sortedOutput;**

**TITLE "Data Imported from Excel";**

**RUN;**

OUTPUT:



1. Run PROC MEANS over the final data created in the previous step and report the results and discuss.

**PROC IMPORT DATAFILE= "/folders/myfolders/catalog.xls" DBMS=xls OUT= catalogtesting REPLACE;**

**SHEET="testing";**

**GETNAMES=YES;**

**MIXED=NO;**

**RUN;**

**LIBNAME ABI '/folders/myfolders/test' ;**

**DATA ABI.catalogtesting2;**

**SET catalogtesting;**

**where NGIF >= 2;**

**keepCustomerID NGIF Money Order;**

**RUN;**

**PROC SORT DATA=ABI.catalogtesting2 OUT=ABI.sortedOutput;**

**key NGIF;**

**key Money;**

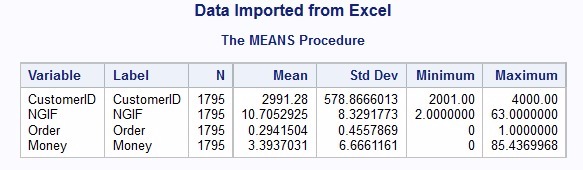
**RUN;**

**PROC MEANS DATA = ABI.sortedOutput;**

**TITLE "Data Imported from Excel";**

**RUN;**

OUTPUT:



The above table is a summary of 1795 records. They are sorted as NGIF > 2.

NGIF: number of orders in the last 24 months.

Therefore we can concur that a person who buys 11 products in 24 months spends around $3.39 on the next order for the next 24 months.

**Questions: Part II (survey.xls)**

1. Perform principal component analysis to summarize the 15 responses.

**PROCIMPORT DATAFILE = "/folders/myfolders/survey.xls"**

**DBMS = xls OUT = surveydata replace;**

**GETNAMES=yes;**

**RUN;**

**DATAsurveydatapreprocessed;**

**SET surveydata;**

**ARRAY y{15} y1-y15;**

**DO i=1 TO 15;**

**IF y{i}='.' THEN y{i}='0';**

**IF y{i}=' ' THEN y{i}='0';**

**END;**

**RUN;**

As, PCA needs numeric values, we convert character variables into numeric. All the attributes are by default considered character, whenever xls file is imported in SAS.

/\*Converting Char to Numeric\*/

**DATAsurveydatafinal;**

**SET surveydatapreprocessed;**

**ARRAY yold{15} y1-y15;**

**ARRAY ynew{15} ynew1-ynew15;**

**DO i=1 TO 15;**

**ynew{i}=input(yold{i},5.);**

**END;**

**RUN;**

**PROCGLM DATA = surveydatafinal;**

**CLASS id y1-y15;**

**MODEL id=ynew1-ynew15;**

**RUN;**

**PROCPRINCOMP DATA=surveydatafinal OUT=pcadat OUTSTAT= xyz;**

**VAR ynew1-ynew15;**

**RUN;**

The ‘pcadat’ table stores all the variables and the corresponding principle components.

No. of principle components= No. of observed variables.

The table stores the statistics(mean, std deviation and no. observed variables) and stores the correlational matrix and Eigen vectors.

OUTPUT:

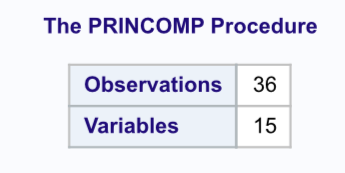


Figure 1

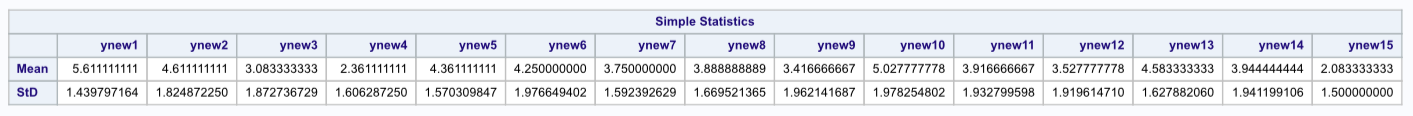


Figure 2

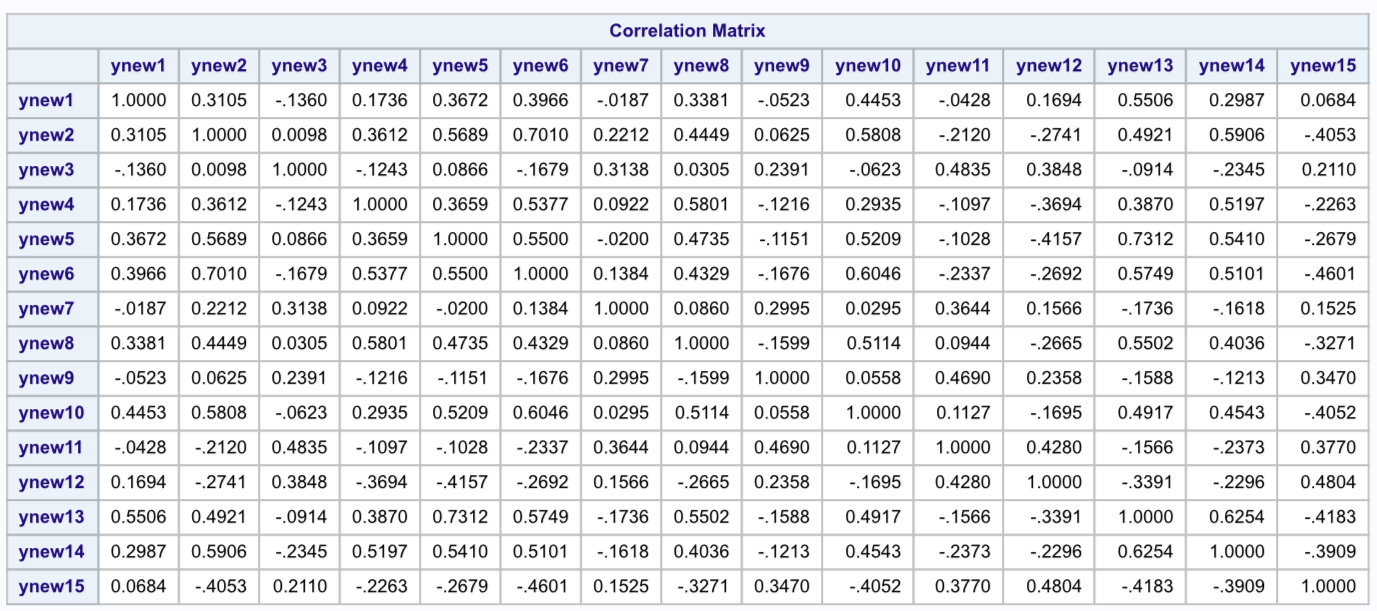


Figure 3

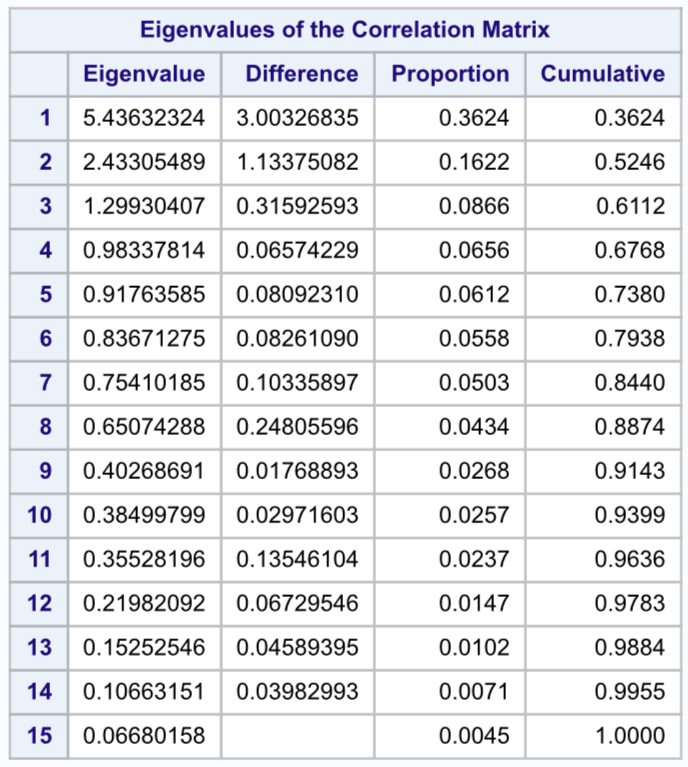


Figure 4

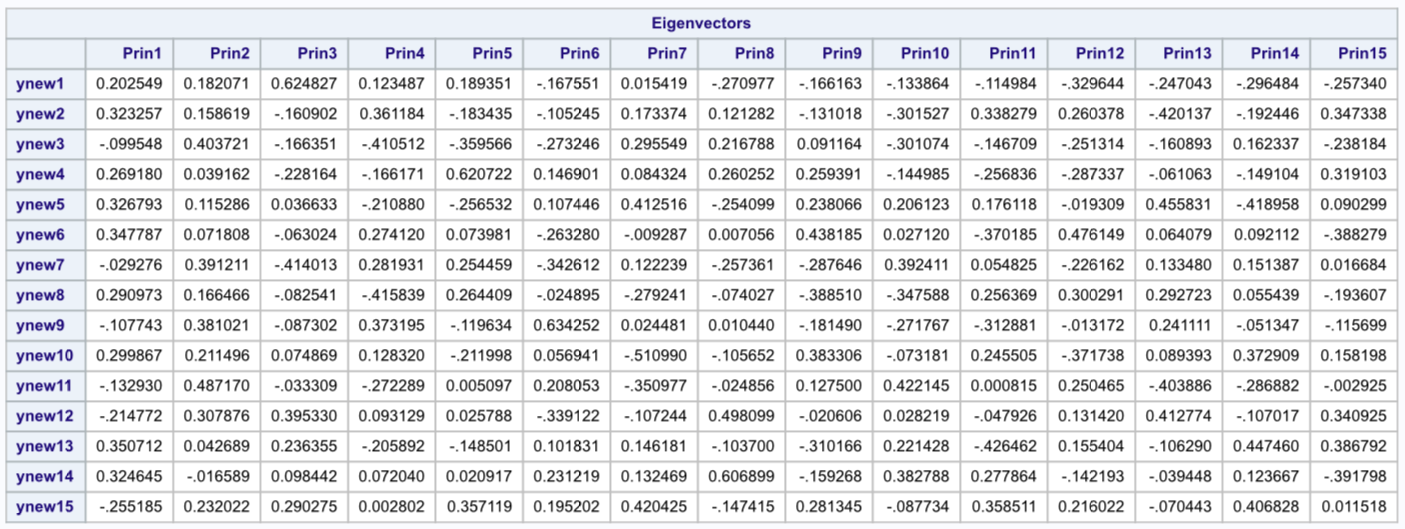


Figure 5

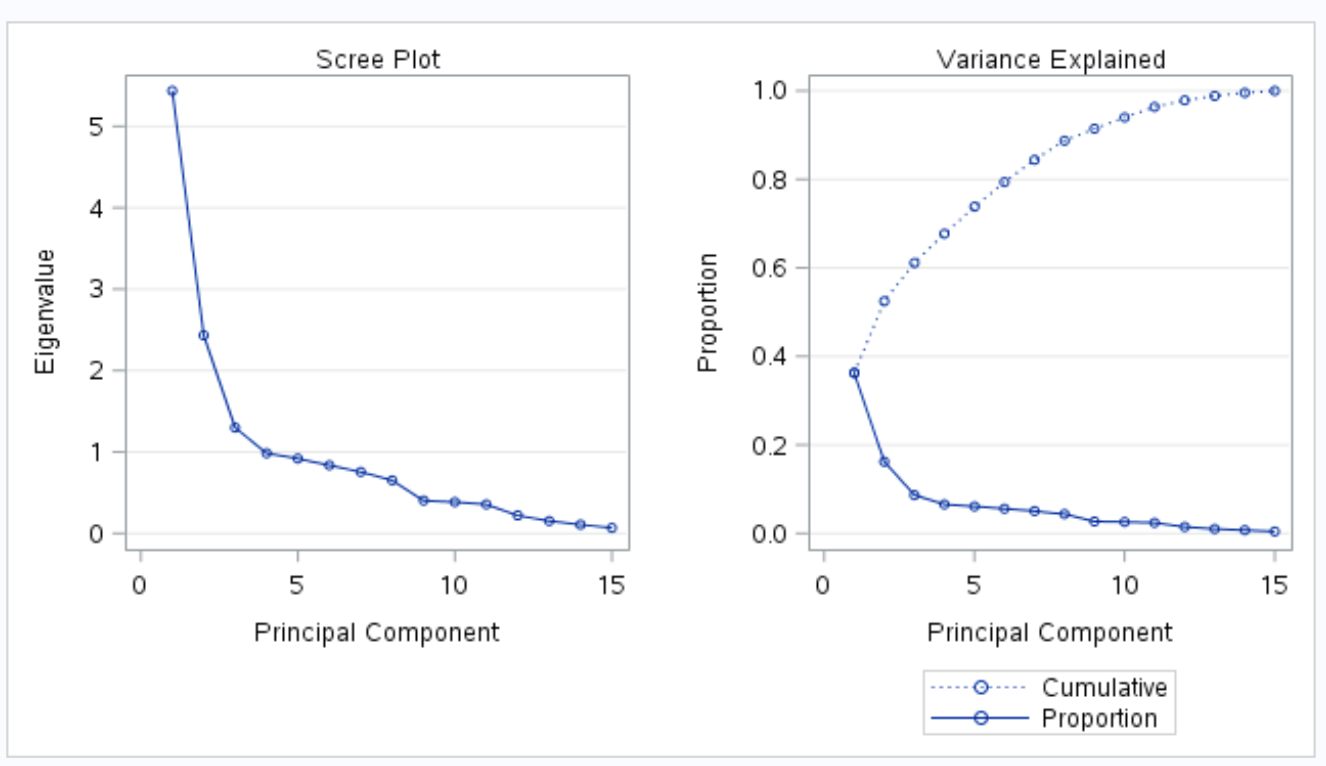


Figure 6

1. Make tables of eigenvalues and eigenvectors and discuss the results.

Ans: -

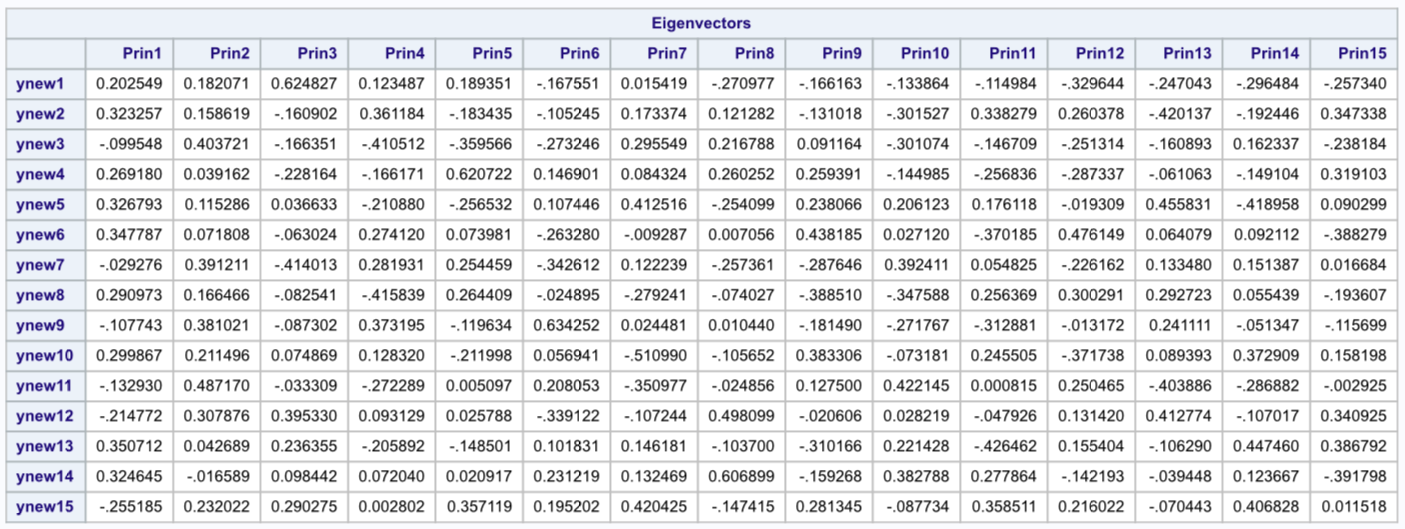


Figure 7

From the above table.

Eigen Values represent the amount of variance shown by each Principle Component.

* Eigen value for 1st principle component is 5.43 which implies to 36.24% of variation in the dataset.
* Eigen value for 2nd principle component is 2.43 which implies to 16.22% of variation in the dataset.

As, we can see the variance reduces when the number of PC increases, which means that only a few account to meaningful variability in the dataset.

C1= b1(x1) +b2(x2)….. .. .. ..+bp (xp)

C1 is the principle component and b1 through bp are the optimal weights. These weights are such that they account for the maximum variation in the dataset.

SAS computes these weights by using an equation called the eigeninan equations. The Eigen vectors are a result of this equation.

Therefore, the Eigen vectors whichcorrespond to principle component 1 and 2 represent the best set of weights to achieve optimum results.

1. How many principal components would be adequate to represent the dimensional space of the data without losing much information? Why?

Ans:-

* The first component extracted in a principal component analysis accounts for a maximal amount of total variance in the observed variables.
* The second component extracted will account for a maximal amount of variance in the data set that was not accounted for by the first component.
* The remaining components that are extracted in the analysis accounts for a maximal amount of variance in the observed variables that was not accounted for by the preceding components.

A principal component analysis proceeds with each new component accounting for progressively smaller and smaller amounts of variance (due to which only the first few components are usually retained and interpreted).

1. Any component showing an eigenvalue greater than 1.00 shows greater amount of variance than had been contributed by one variable. So, we retain such a component.
2. And the proportion also gives enough idea of the variation accounted as compared to other principle components.

Hence we can conclude that PC 1 and 2 can be considered as they have Eigen value greater that 1 (i.e.) 5.43 and 2.43 and with proportions 0.3624 and 0.1622.

The scree plot shows a break down after 2nd Principle Component as well, wherein we can understand that Principle Components after 2 do not have meaningful account of variations and so can be discarded for further analysis.

4. For the first observation, calculate (show and explain your work) the scores of the first principal component and compare it to the value obtained from SAS. Discuss any possible differences.

Ans: The Principle component scores can be printed as below

**PROC PRINT DATA = pcadat(DROP=i);**

**TITLE ' PCA score ';**

**VAR y1-y15 Prin1-Prin15;**

**RUN;**

Principle components can be calculated from the Eigen vectors using the equation below:

C1= b1(x1) +b2(x2)….. .. .. ..+bp (xp)

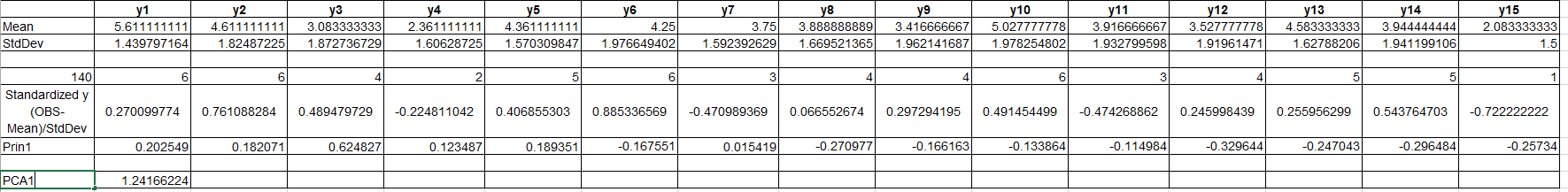


Figure 8

We can see the PCA scores from SAS in the figure below.

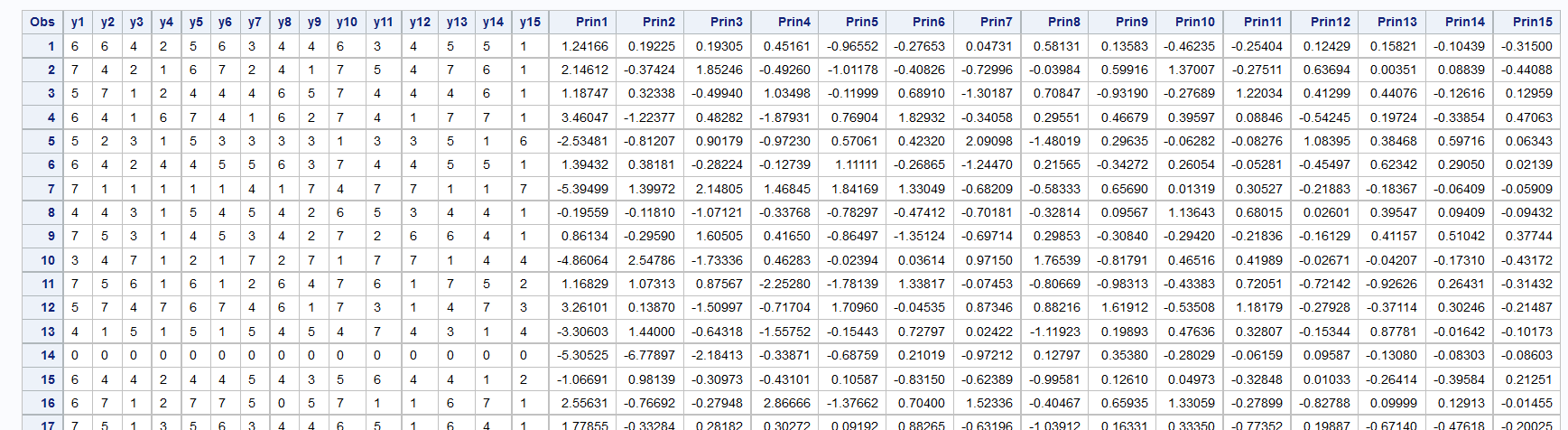


Figure 9

From PCA1 of figure 8 and Prin1 of Figure 9, we can see that the values of PCA are the same (1.24166)