Chapter 2

Preparation for Clustering

Section 2.1

Variable Clustering

Why variable clustering?

- Data sets may contain too many variables (there can be thousands or more).
- It is computationally too expensive/impractical.
- Some variables are correlated.

Objectives

- Select right variables remove collinearity.
- Introduce **PROC VARCLUS**.

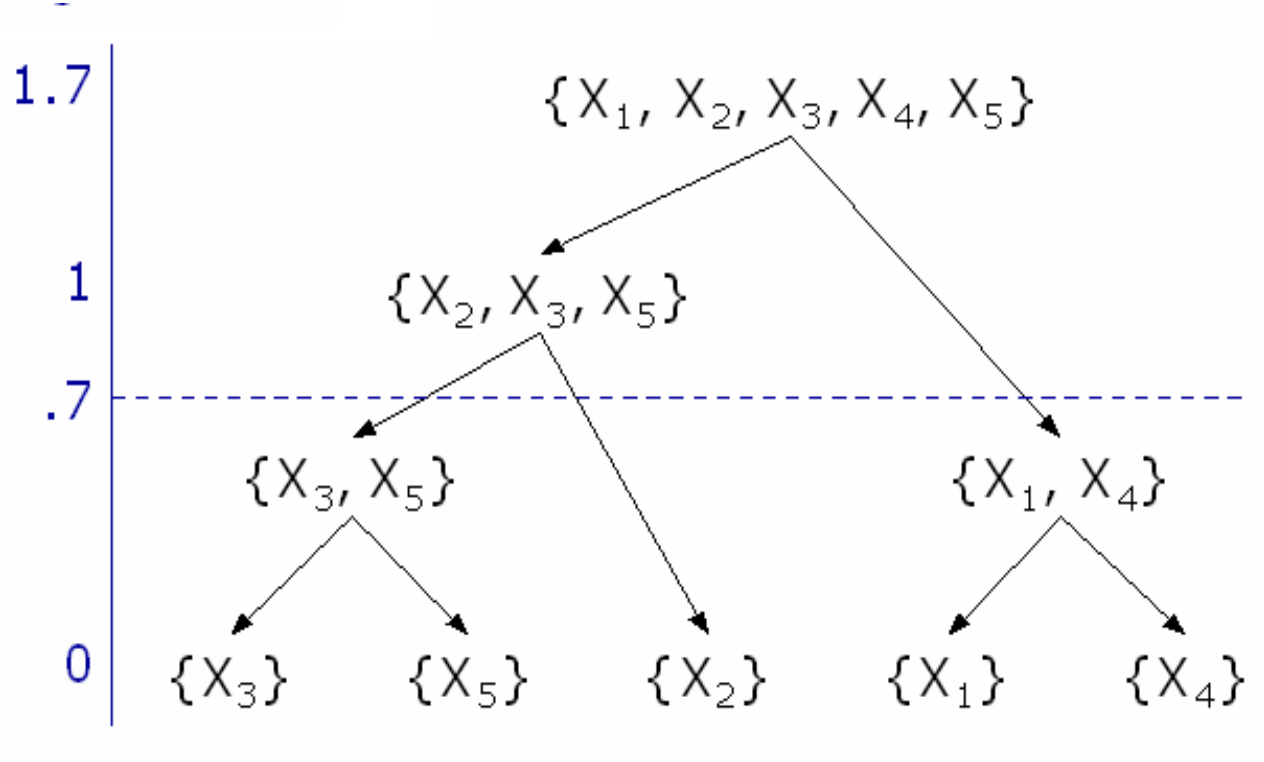
The VARCLUS procedure

General form of the VARCLUS procedure

```
PROC VARCLUS DATA=SAS-data-set <options>;
    VAR variables;
RUN;
```

- Divides a set of variables into non-overlapping clusters.
- Replace a set of variables by a single member of each cluster to act as a representative.
- Reduces the number of variables.

How does PROC VARCLUS work?



PROC VARCLUS uses divisive clustering to create variable subgroups that are as dissimilar as possible.

Variable selection

- Use your business knowledge.
- Judge by 1-R^2 ratio (R^2 = squared correlation).

$$1-R^2 \text{ ratio} = \frac{1-R^2 \text{ own cluster}}{1-R^2 \text{ next closest cluster}}$$

- Small 1–R^2 ratio values are preferred, indicating the cluster are well separated.
- The smaller the ratio, the stronger the correlation between the variable and its own cluster.

Variable Reduction

Cluster	Variable	R-squared with Own Cluster	R-squared with Next Closest	1-R**2 Ratio
Cluster 1	RedMeat	0.5350	0.2185	0.5950
	WhiteMeat	0.4544	0.3331	0.8181
	Eggs	0.7926	0.4902	0.4067
	Milk	0.5529	0.2721	0.6142
Cluster 2	Cereal	0.8255	0.4630	0.3250
	Nuts	0.8255	0.4549	0.3201
Cluster 3	Fish	0.7019	0.1365	0.3452
	Starch	0.7019	0.3075	0.4304
Cluster 4	FruitVeg	1.0000	0.0538	0.0000

Example : variable clustering using *PROC VARCLUS*

The data set: **pizza** data set, containing nutrient information on 300 samples of frozen pizza

The variables:

brand the pizza brand (used only as a class label)
id the sample analyzed
prot amount of protein per 100 grams in the sample
fat amount of fat per 100 grams
ash amount of ash per 100 grams
sodium amount of sodium per 100 grams
carb amount of carbohydrates per 100 grams
mois percentage of moisture in each sample
cal calories per gram in the sample

The SAS code

```
title 'Pizza Nutrient Data';

/* create the variable clusters */
proc varclus data=teaching.pizza proportion=.9
        outtree=tree;
  var mois prot fat ash sodium carb cal;
run;

/* print the OUTTREE dataset */
proc print data=tree;
run;

/* generate a tree diagram of the cluster structure */
proc tree data=tree horizontal;
  height  _PROPOR_;
run;
```

Some results

Oblique Principal Component Cluster Analysis

Observations	300	Proportion	0.9
Variables	7	Maxeigen	0

Cluster Summary for 4 Clusters

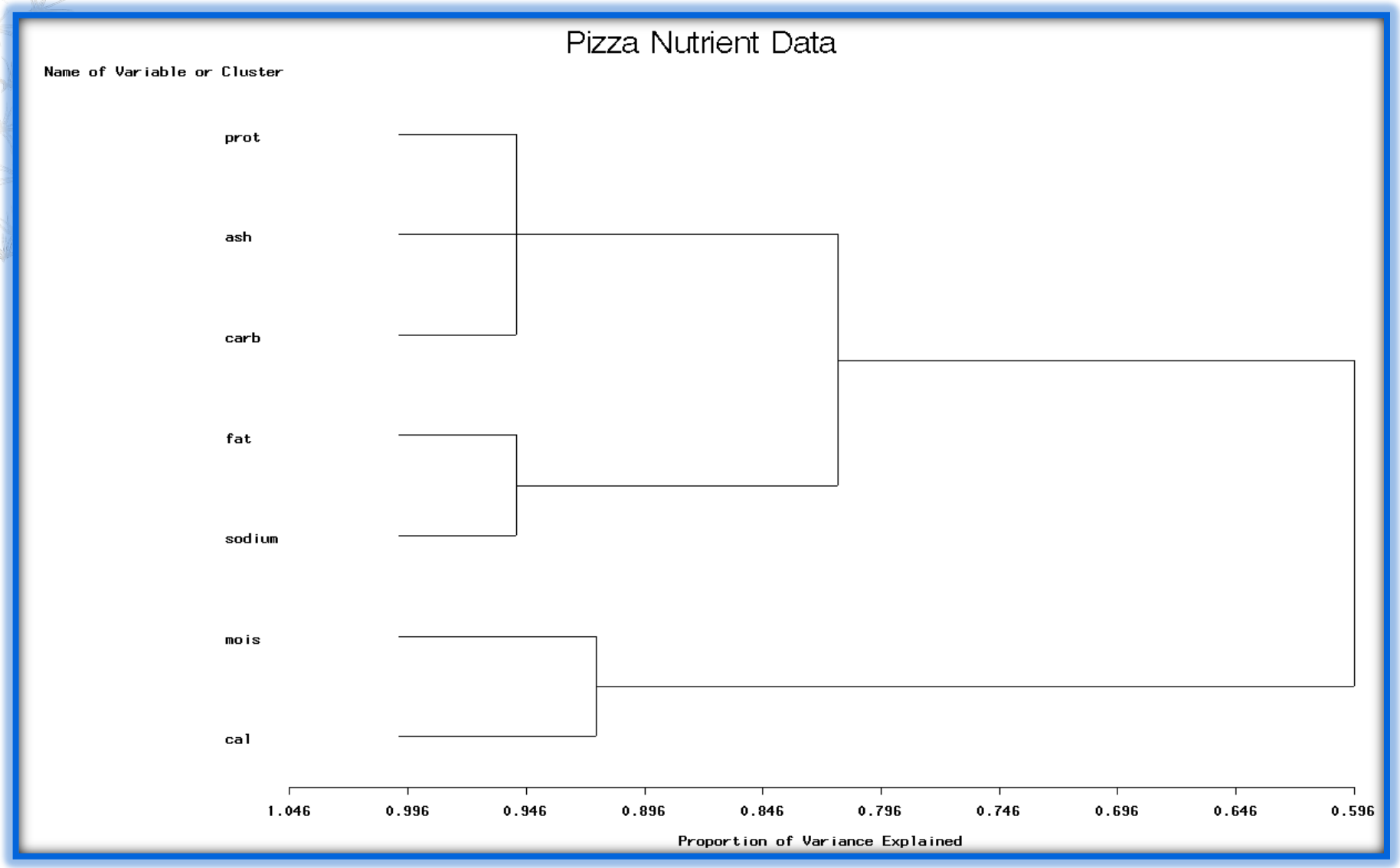
Cluster	Members	Cluster Variation	Variation Explained	Proportion Explained	Second Eigenvalue
1	3	3	2.717956	0.9060	0.1847
2	1	1	1	1.0000	
3	2	2	1.933325	0.9667	0.0667
4	1	1	1	1.0000	

Total variation explained = 6.651281 Proportion = 0.9502

Some results: the R squared table

4 Clusters		R-squared with		
Cluster	Variable	Own Cluster	Next Closest	1-R**2 Ratio
Cluster 1	prot	0.8774	0.2223	0.1576
	ash	0.9101	0.6619	0.2660
	carb	0.9305	0.4109	0.1181
Cluster 2	mois	1.0000	0.5844	0.0000
Cluster 3	fat	0.9667	0.5846	0.0802
	sodium	0.9667	0.4515	0.0608
Cluster 4	cal	1.0000	0.5844	0.0000

The tree


Pizza Nutrient Data
Name of Variable or Cluster
prot
ash
carb
fat
sodium
mois
cal
1.046
0.996
0.946
0.896
0.846
0.796
0.746
0.696
0.646
0.596
Proportion of Variance Explained

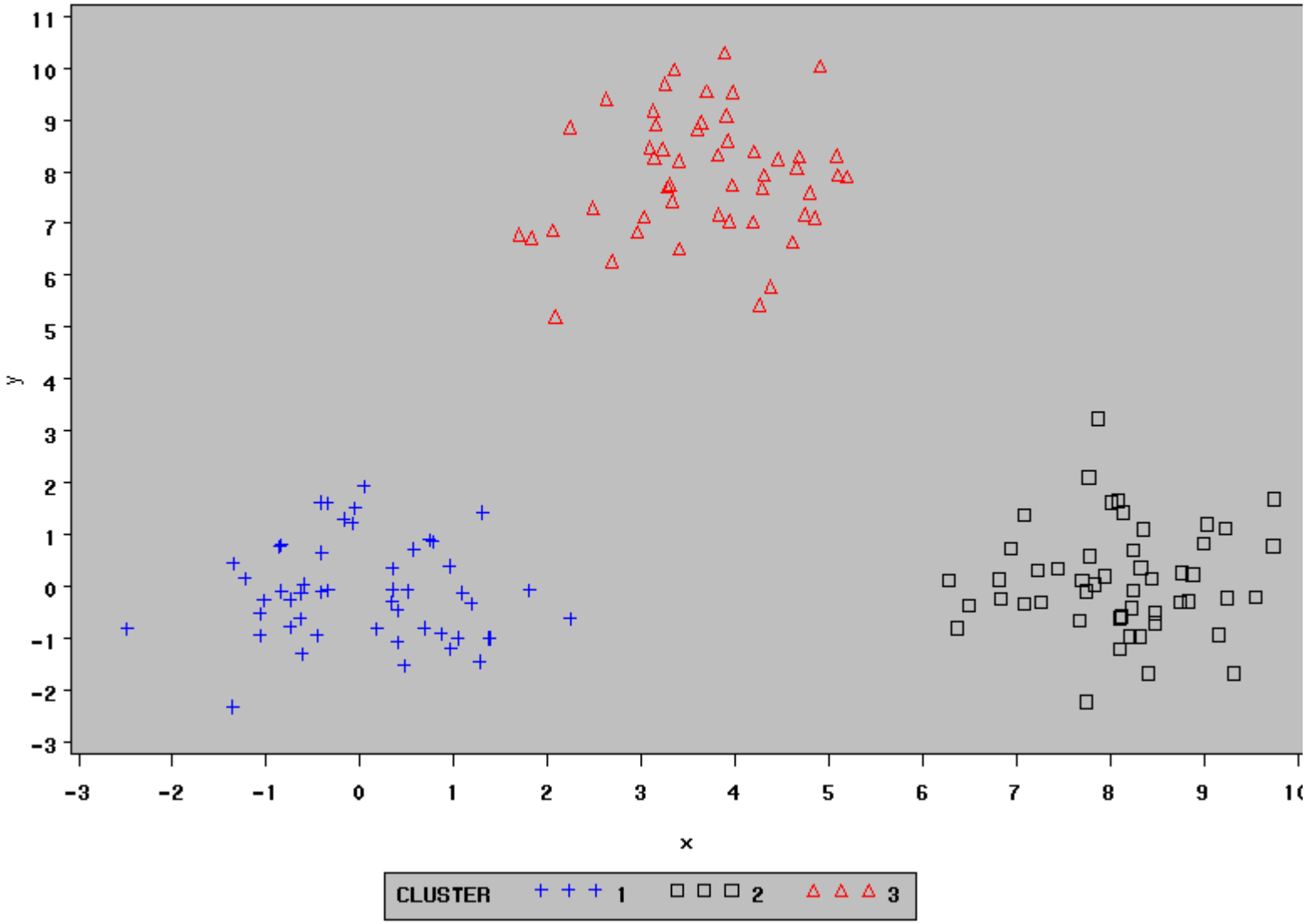
Section 2.2

Graphical Aids to Clustering

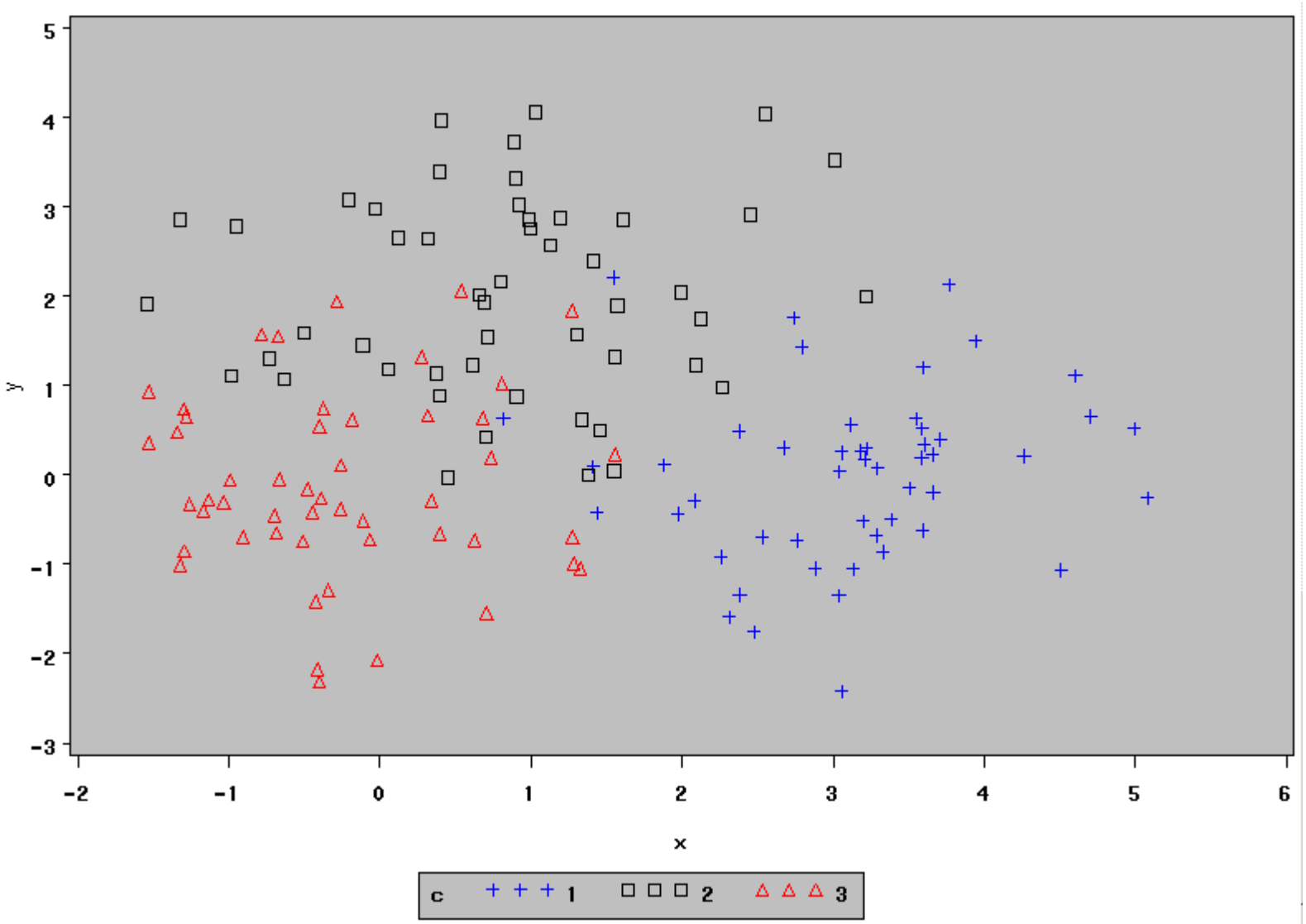
Objectives

- Use the plots (e.g., principal component plots, multidimensional scaling plots, and canonical discriminant plots) to detect cluster structure
- Introduce the PRINCOMP and GPLOT procedures

Data forming well-separated compact clusters



Data forming poorly-separated clusters



The GPLOT Procedure

for high-resolution plots

General form of the **GPLOT** procedure:

```
PROC GPLOT DATA=SAS-data-set;
        PLOT y-variable * x-variable <= class-variable>
                </ options>;
RUN;
```

Plotting more than two dimensions

How do you plot more than two dimensions?

- Principal component analysis (PRINCOMP)
- Multidimensional scaling (MDS)
- Canonical discriminant analysis (CANDISC)

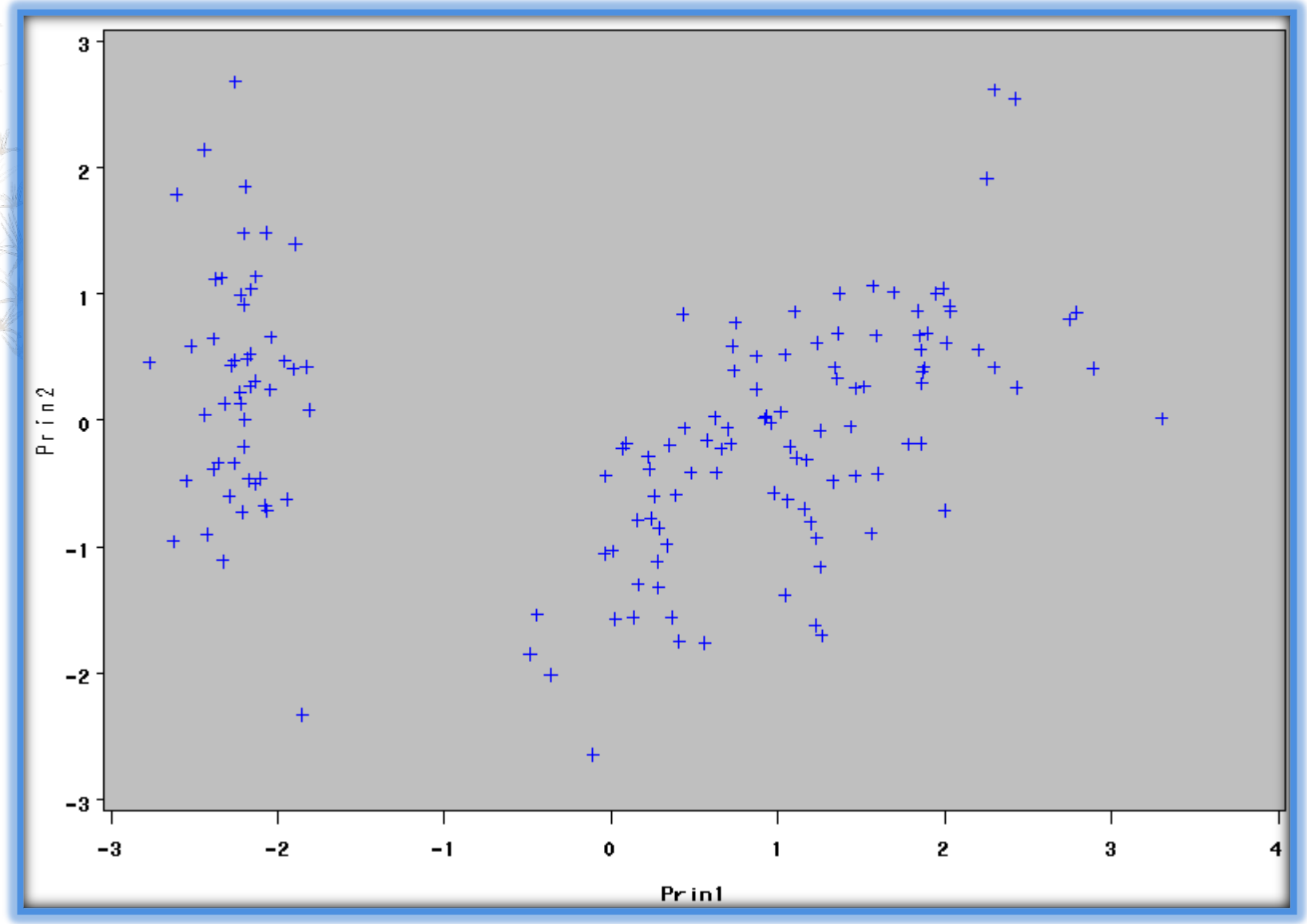
What is principal component analysis (PCA)?

- A method to reduce the dimensionality of a high-dimensional dataset while retaining the variation present in the dataset as much as possible.
- A method to create new meaningful underlying variables.
- A method to visualize data.

The principle components

- New variables that are uncorrelated.
- A linear combination of the old variables.
- Ordered such that the first few components retain most of the variation in the original variables.

Principle component analysis (PCA) plot



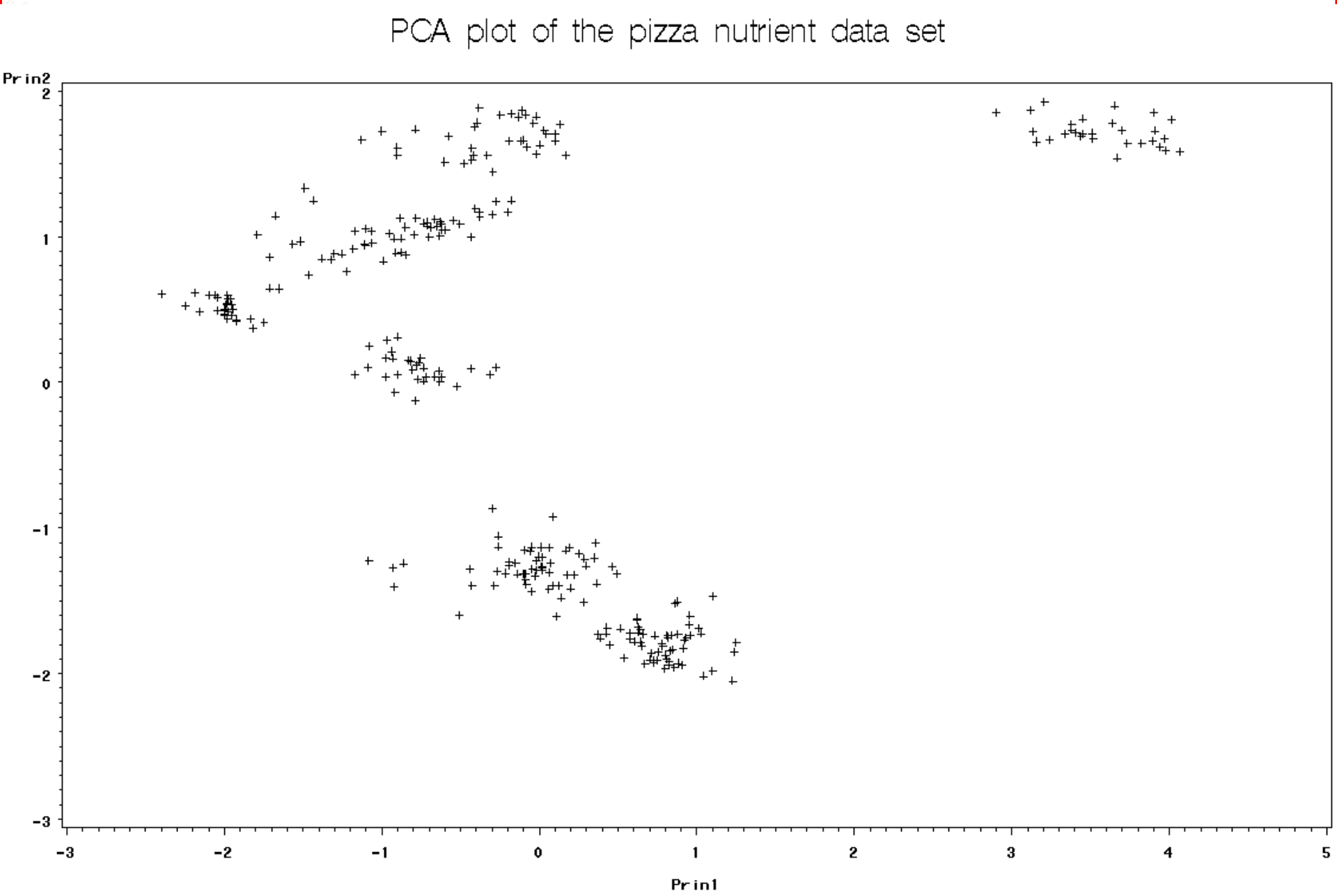
The PRINCOMP Procedure

General form:

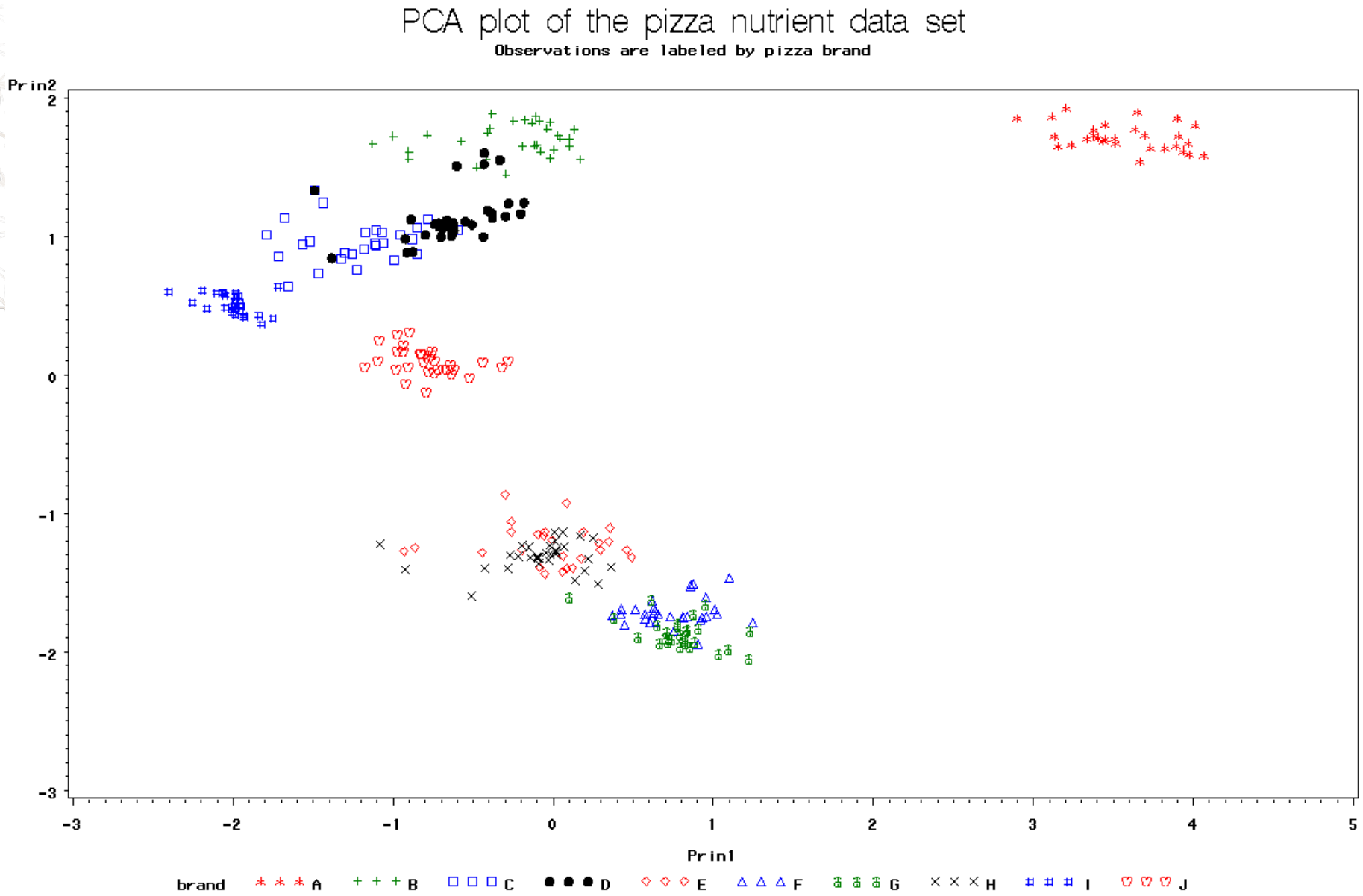
```
PROC PRINCOMP
        DATA=input dataset
        OUT=output dataset <options>;
        VAR variables;
RUN;
```

An example: PCA using SAS

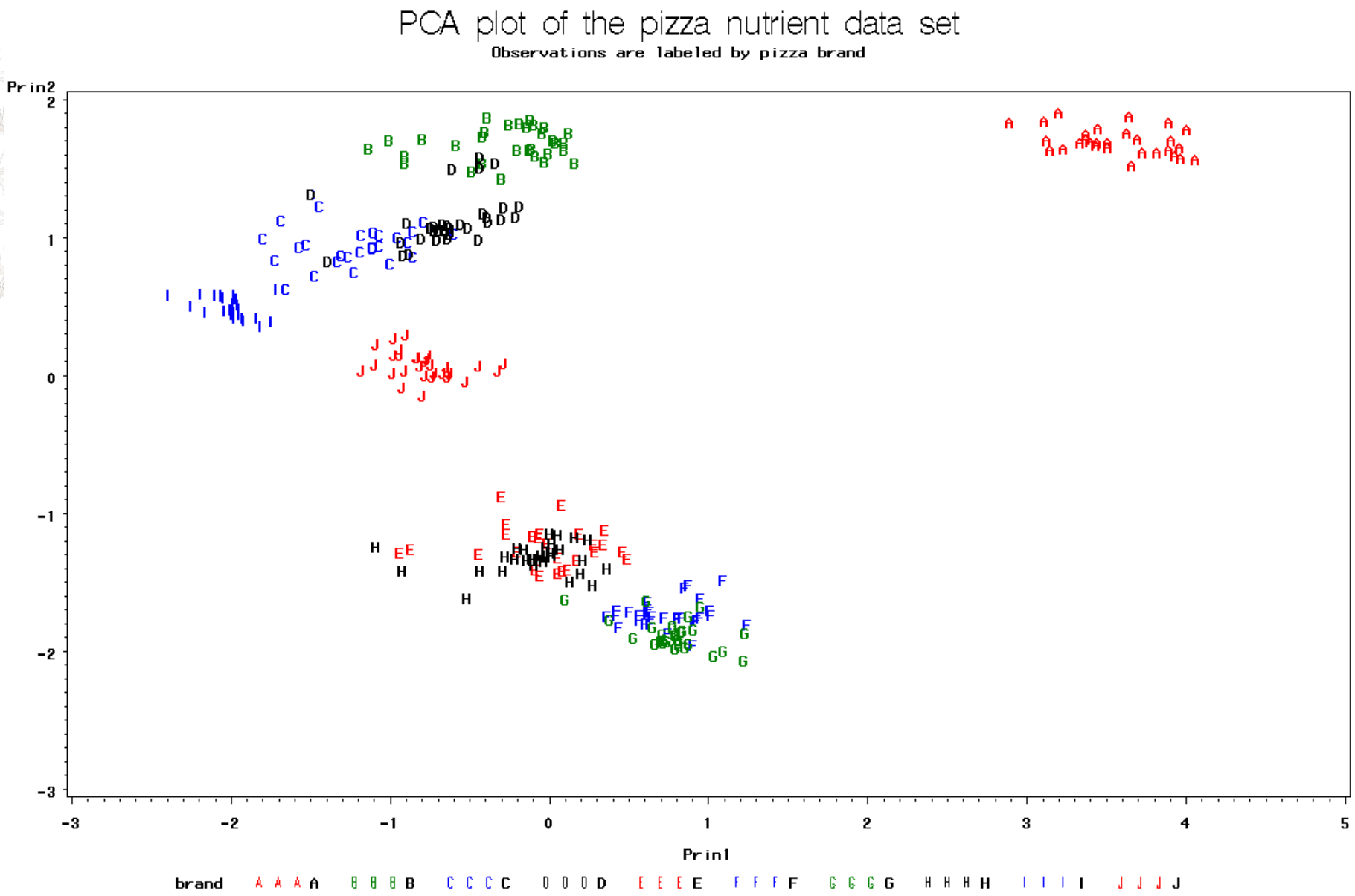
```
%let inputs = carb mois sodium cal;

title 'PCA of the pizza nutrient data set';
proc princomp data=teaching.pizza out=pcaout;
  var &inputs;
run;

title 'PCA plot of the pizza nutrient data set';
proc gplot data=pcaout;
  plot prin2*prin1;
run;

title2 'Observations are labeled by pizza brand';
proc gplot data=pcaout;
  plot prin2*prin1=brand;
run;
```


PCA plot of the pizza nutrient data set
Prin2
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
4
5
Prin1

PCA plot labeled by different symbols



PCA plot labeled by brand names



Some gplot codes

```
proc gplot data=pcaout;
  plot prin2*prin1=brand;
  symbol1 v=star color=red;
  symbol2 v=plus color=green;
  symbol3 v=square color=blue;
  symbol4 v=dot color=black;
  symbol5 v=diamond color=red;
  symbol6 v=triangle color=blue;
  symbol7 v=, color=green;
  symbol8 v=x color=black;
  symbol9 v=hash color=blue;
  symbol10 v=# color=red;

run;
```

```
proc gplot data=pcaout;
  plot prin2*prin1=brand;
  symbol1 v='A' color=red;
  symbol2 v='B' color=green;
  symbol3 v='C' color=blue;
  symbol4 v='D' color=black;
  symbol5 v='E' color=red;
  symbol6 v='F' color=blue;
  symbol7 v='G' color=green;
  symbol8 v='H' color=black;
  symbol9 v='I' color=blue;
  symbol10 v='J' color=red;

run;
```