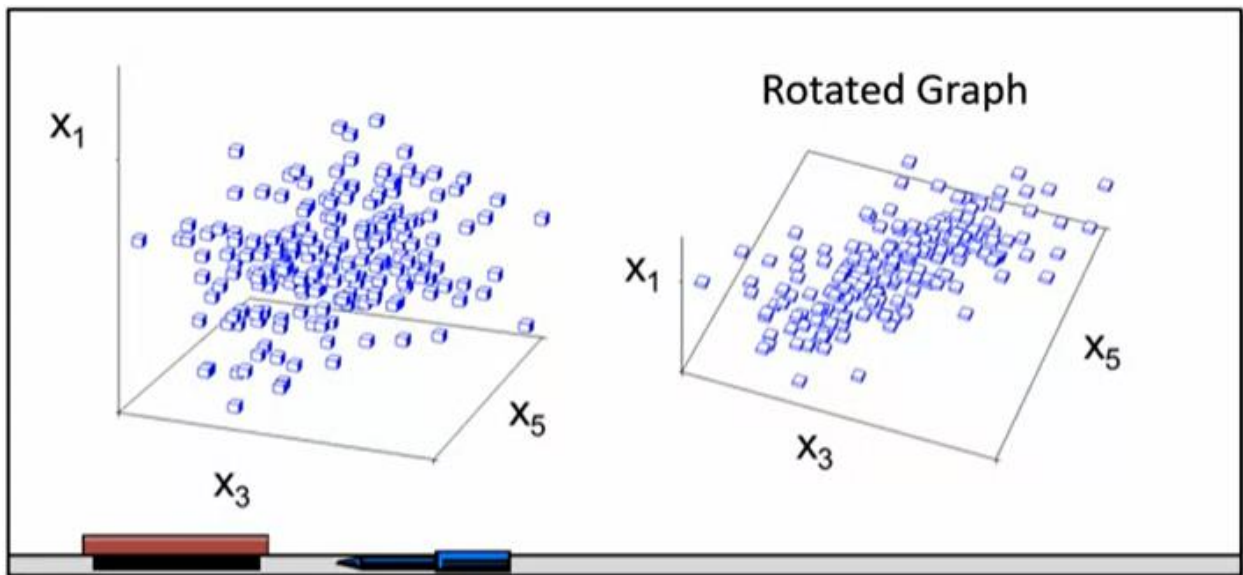
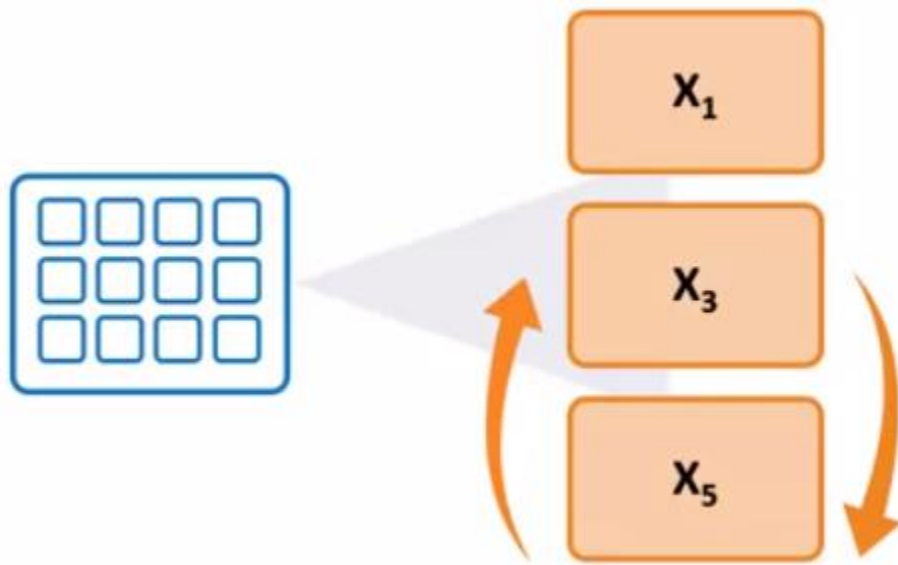
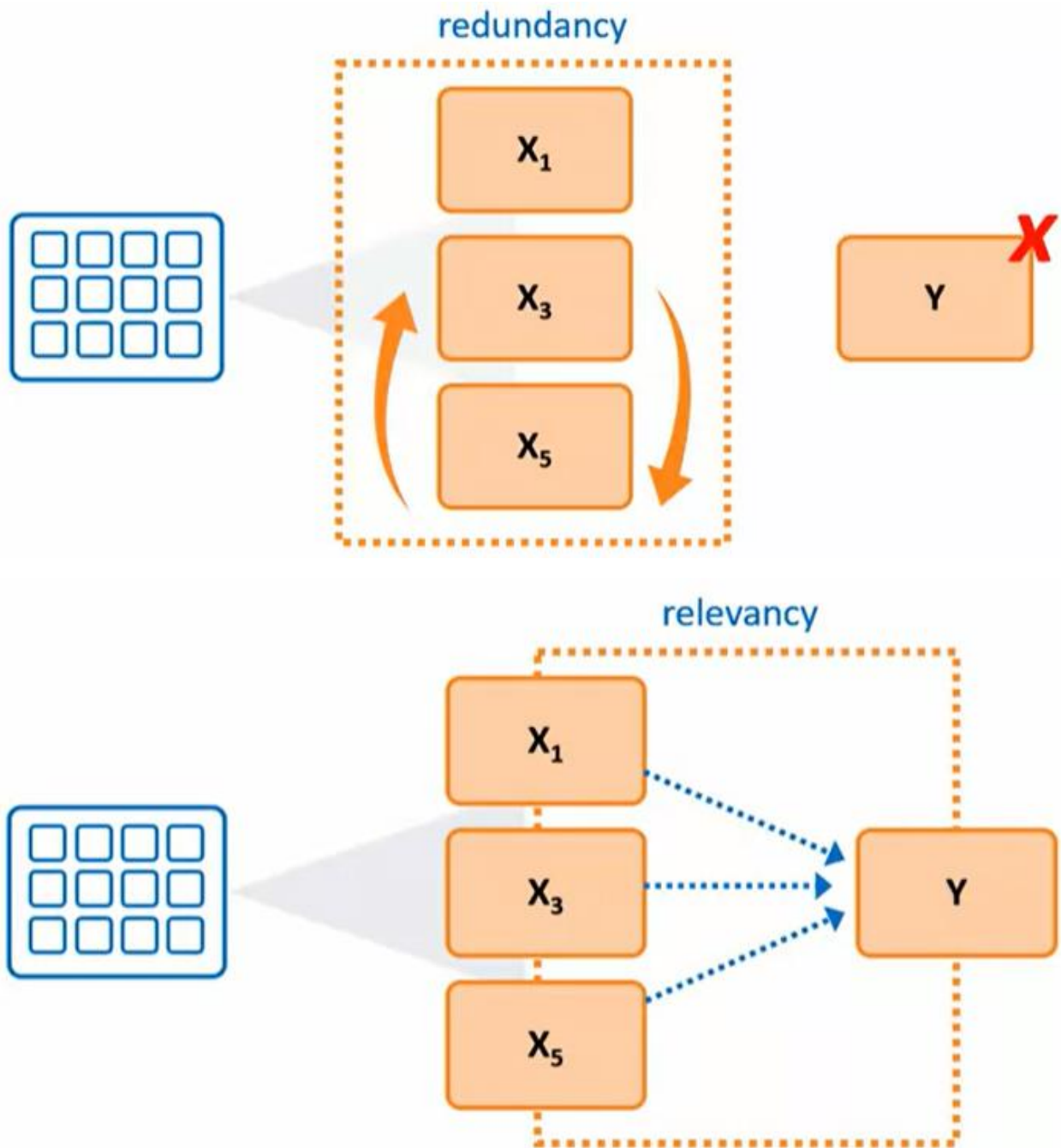


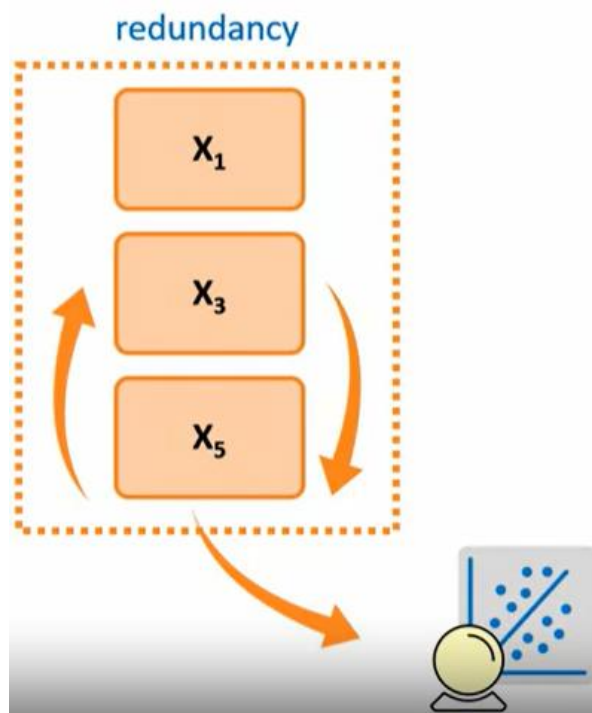
In this topic, you learn to do the following:

- identify the ways that redundant inputs can degrade your analysis
- describe the main steps of variable clustering and principal component analysis
- perform variable clustering by using the VARCLUS procedure
- list methods of selecting variables

Problem of Redundancy

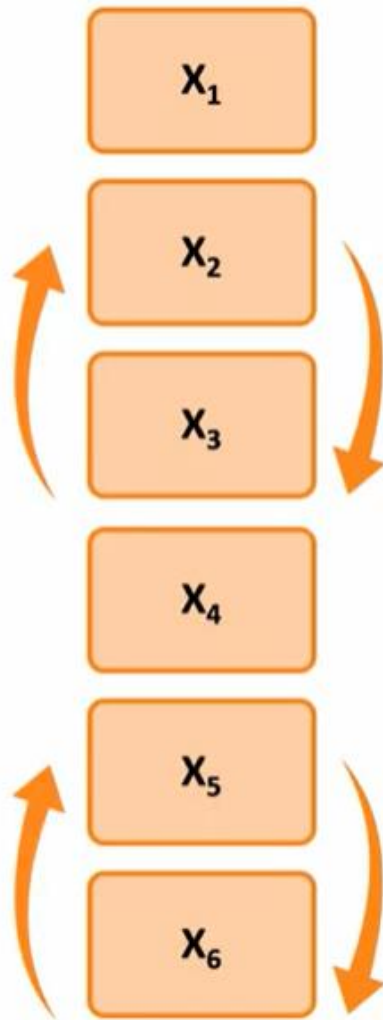






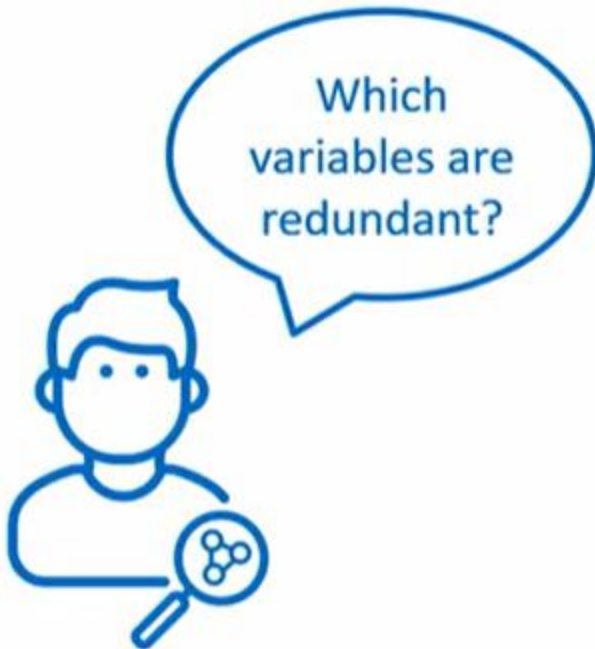
Degrade Your Analysis:

- destabilize the parameter estimates
- increase the risk of overfitting
- confound interpretation
- increase computation time
- increase scoring effort
- increase the cost of data collection and augmentation



1. Redundancy
2. Relevancy





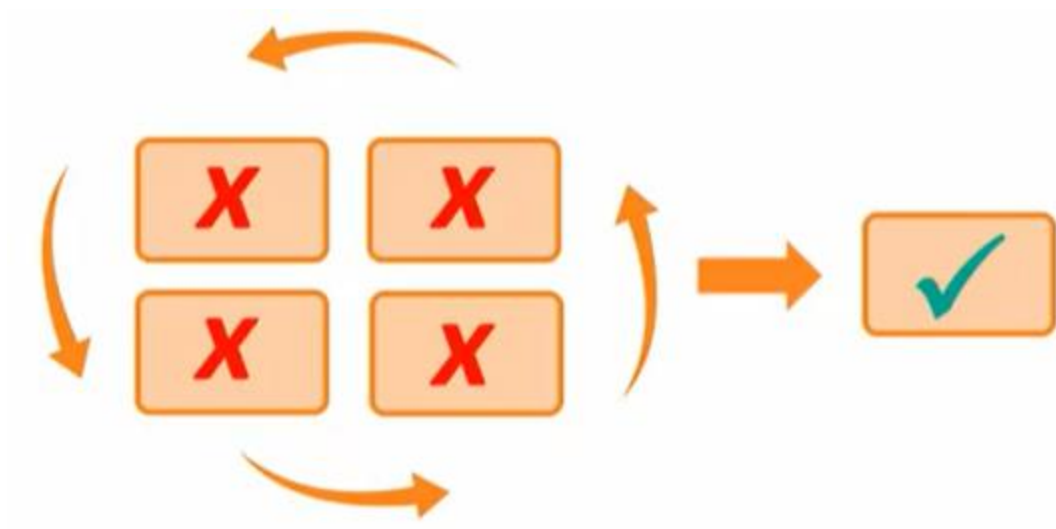
Correlation Matrix

	X_1	X_2	X_3	X_4	X_5
X_1	1	-.11	-.03	-.69	-.04
X_2	-.11	1	-.14	.07	.04
X_3	-.03	-.14	1	.04	-.73
X_4	-.69	.07	.04	1	.02
X_5	-.04	.04	-.73	.02	1

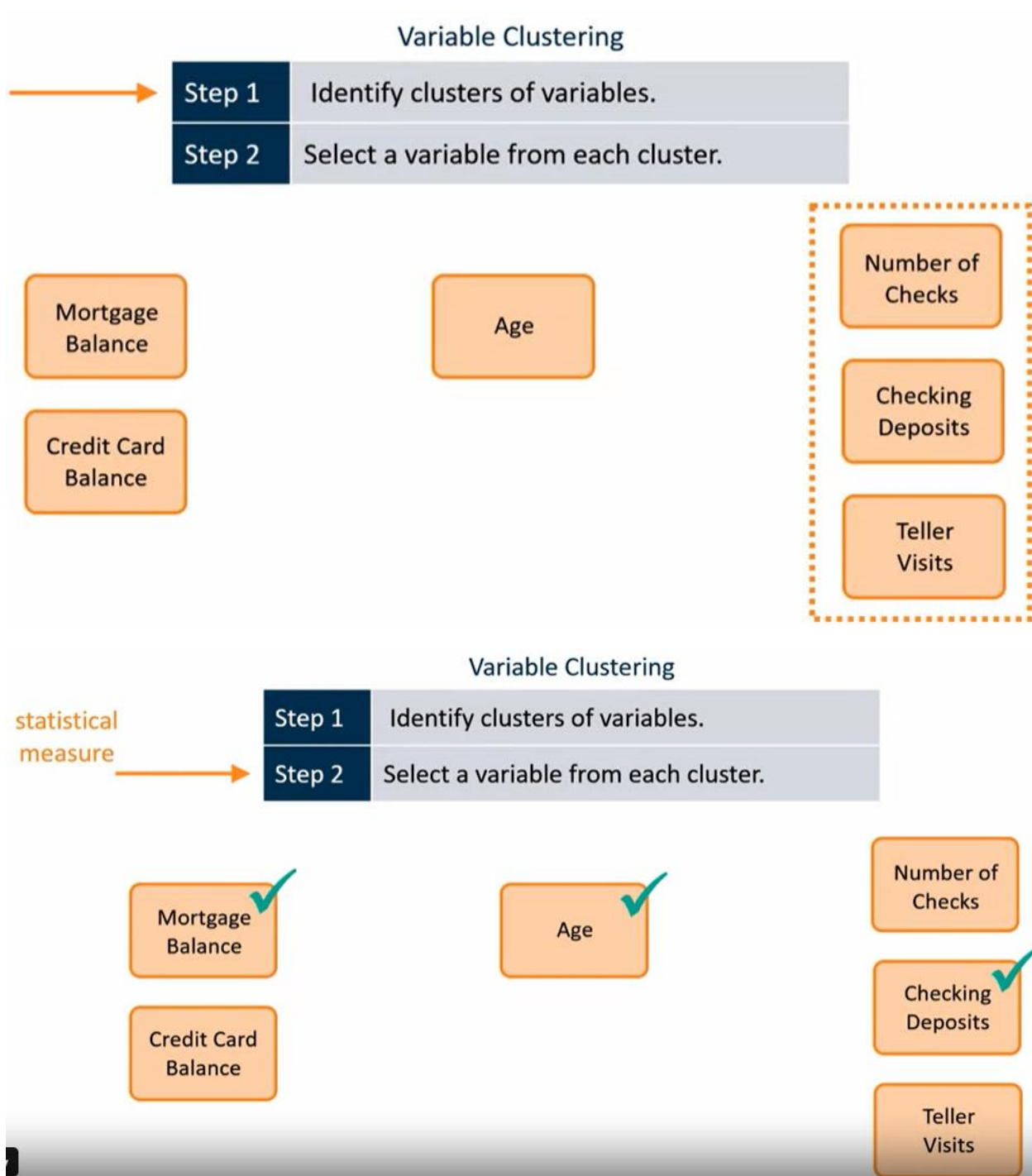
Correlation Matrix

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	...	X_{1000}
X_1	1	-.11	-.03	-.69	-.04	.25	.77	...	-.21
X_2	-.11	1	-.14	.07	.04	-.80	-.0226
X_3	-.03	-.14	1	.04	-.73	.46	-.13	...	-.17
X_4	-.69	.07	.04	1	.02	.09	.0509
X_5	-.04	.04	-.73	.02	1	.39	-.0480
X_6	.25	-.80	.46	.09	.39	1	.6653
X_7	.77	-.02	-.13	.05	-.04	.66	1	...	-.12
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮
X_{1000}	-.21	.26	-.17	.09	.80	.53	-.12	...	1





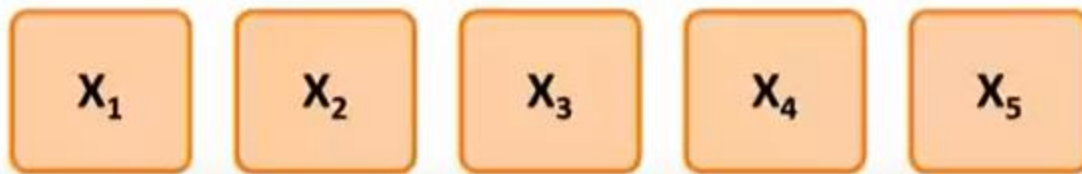
Variable Clustering Method



Understanding Principal Components

Correlation Matrix

	X_1	X_2	X_3	X_4	X_5
X_1	1	-.11	-.03	-.69	-.04
X_2	-.11	1	-.14	.07	.04
X_3	-.03	-.14	1	.04	-.73
X_4	-.69	.07	.04	1	.02
X_5	-.04	.04	-.73	.02	1



Correlation Matrix

	X_1	X_2	X_3	X_4	X_5
X_1	1	-.11	-.03	-.69	-.04
X_2	-.11	1	-.14	.07	.04
X_3	-.03	-.14	1	.04	-.73
X_4	-.69	.07	.04	1	.02
X_5	-.04	.04	-.73	.02	1

Correlation Matrix

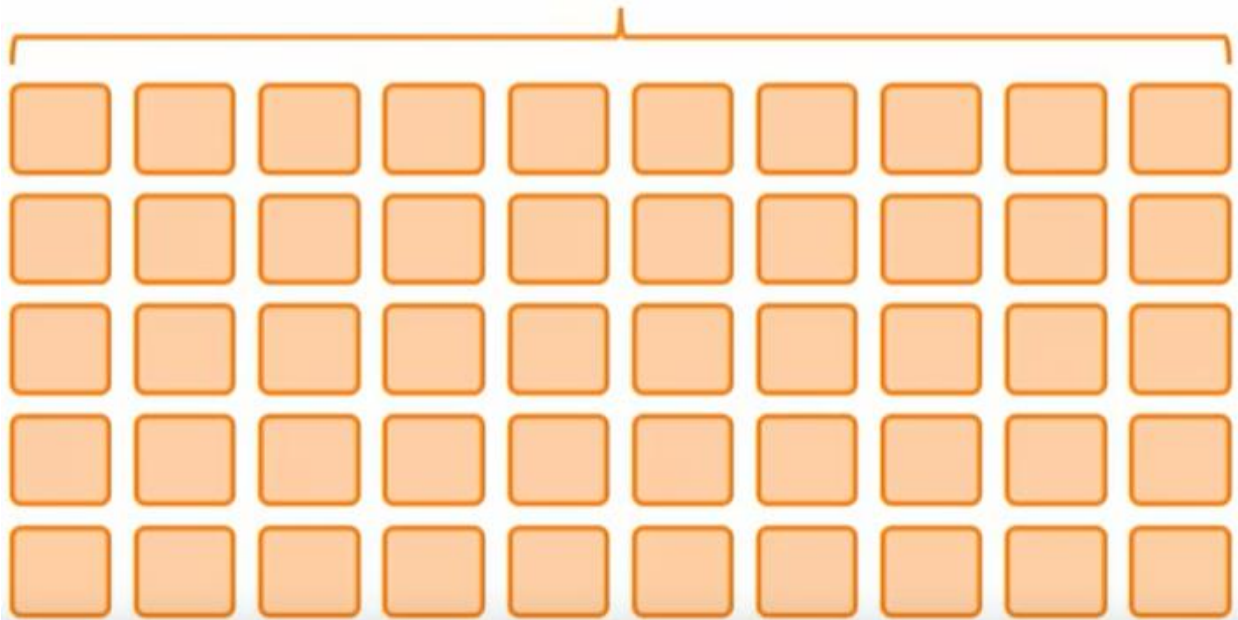
	X_1	X_2	X_3	X_4	X_5
X_1	1	-.11	-.03	-.69	-.04
X_2	-.11	1	-.14	.07	.04
X_3	-.03	-.14	1	.04	-.73
X_4	-.69	.07	.04	1	.02
X_5	-.04	.04	-.73	.02	1




PROC VARCLUS

iterative principal components analysis

k input variables



most
variation



least
variation

$$\begin{aligned}
 PC_{(1)} &= W_{(1)1}X_1 + W_{(1)2}X_2 + \dots + W_{(1)k}X_k \\
 PC_{(2)} &= W_{(2)1}X_1 + W_{(2)2}X_2 + \dots + W_{(2)k}X_k \\
 &\vdots \\
 PC_{(k)} &= W_{(k)1}X_1 + W_{(k)2}X_2 + \dots + W_{(k)k}X_k
 \end{aligned}$$

where the weights are chosen to maximize the quantity

$$\frac{\text{Variance of PC}}{\text{Total Variance}}$$

and the correlation $\text{corr}(PC_{(i)}, PC_{(j)}) = 0$ for each i is not equal to j .

Correlation Matrix

	X_1	X_2	X_3	X_4	X_5
X_1	1	-.11	-.03	-.69	-.04
X_2	-.11	1	-.14	.07	.04
X_3	-.03	-.14	1	.04	-.73
X_4	-.69	.07	.04	1	.02
X_5	-.04	.04	-.73	.02	1

Covariance Matrix

	PC_1	PC_2	PC_3	PC_4	PC_5	
1.8	0	0	0	0	0	PC_1
0	1.7	0	0	0	0	PC_2
0	0	1.0	0	0	0	PC_3
0	0	0	0.3	0	0	PC_4
0	0	0	0	0.2	0	PC_5

eigenvalue-decomposition

Covariance Matrix

PC ₁	PC ₂	PC ₃	PC ₄	PC ₅	
1.8	0	0	0	0	PC ₁
0	1.7	0	0	0	PC ₂
0	0	1.0	0	0	PC ₃
0	0	0	0.3	0	PC ₄
0	0	0	0	0.2	PC ₅

$$1.8/5 = .36$$

Covariance Matrix

PC ₁	PC ₂	PC ₃	PC ₄	PC ₅	
1.8	0	0	0	0	PC ₁
0	1.7	0	0	0	PC ₂
0	0	1.0	0	0	PC ₃
0	0	0	0.3	0	PC ₄
0	0	0	0	0.2	PC ₅

$$(1.8+1.7)/5 = .7$$

Covariance Matrix

PC ₁	PC ₂	PC ₃	PC ₄	PC ₅	
1.8	0	0	0	0	PC ₁
0	1.7	0	0	0	PC ₂
0	0	1.0	0	0	PC ₃
0	0	0	0.3	0	PC ₄
0	0	0	0	0.2	PC ₅

$$(1.8+1.7+1.0)/5 = .9$$

- X** X₁
- X** X₂
- X** X₃
- X** X₄
- X** X₅

Principal Component Coefficient Matrix

	PC ₁	PC ₂	PC ₃	PC ₄	PC ₅
X ₁	-.25	-.65	.09	.69	.15
X ₂	.21	.10	.97	.02	.11
X ₃	-.65	.28	.04	-.11	.70
X ₄	.23	.66	-.14	.70	.07
X ₅	.65	-.23	-.19	-.10	.69

- PC₁
- PC₂
- PC₃
- PC₄
- PC₅

$$PC_1 = -.25 * X_1 + .21 * X_2 + -.65 * X_3 + .23 * X_4 + .65 * X_5$$

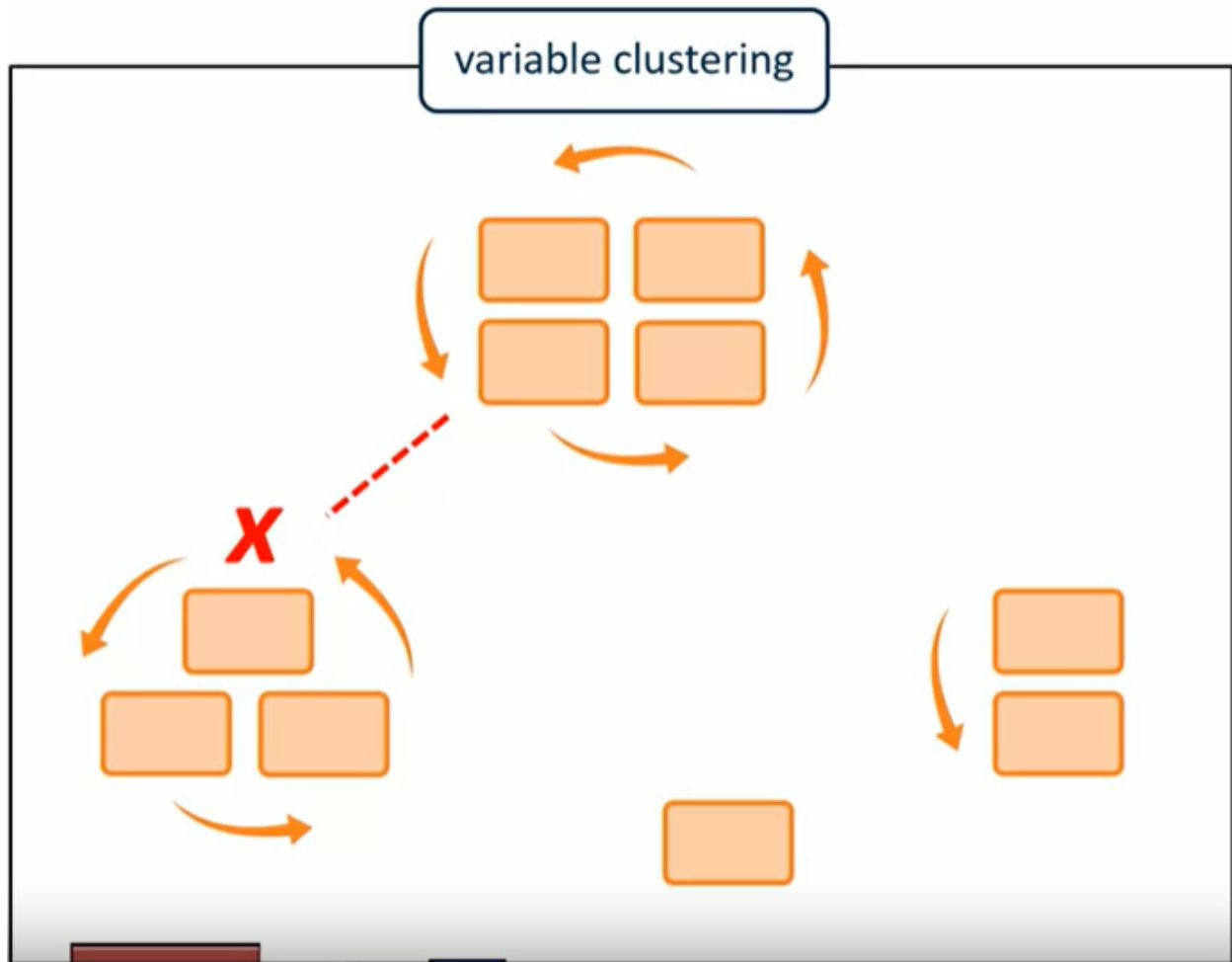
Principal Component
Coefficient Matrix

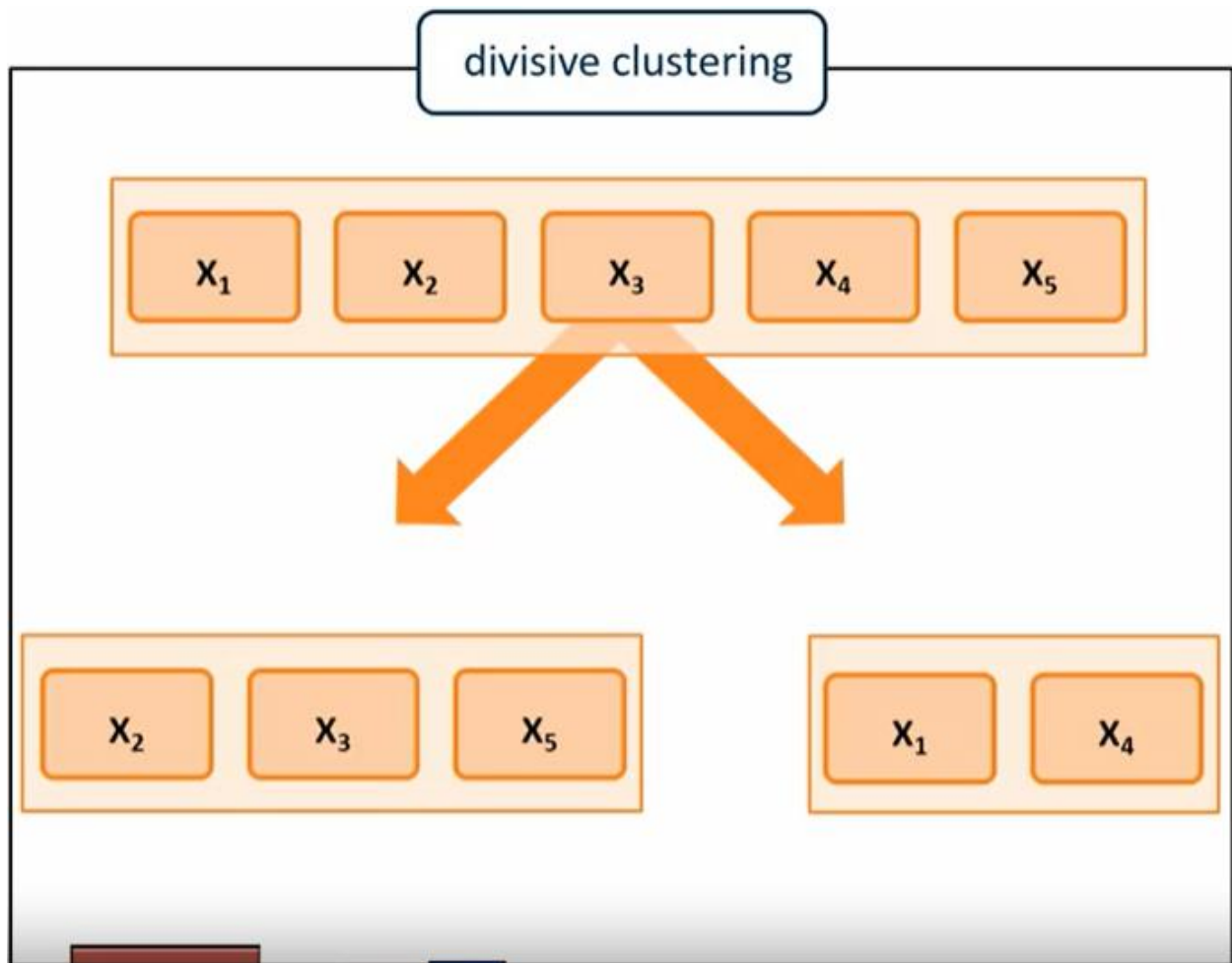
	PC ₁	PC ₂	PC ₃	PC ₄	PC ₅
X ₁	-.25	-.65	.09	.69	.15
X ₂	.21	.10	.97	.02	.11
X ₃	-.65	.28	.04	-.11	.70
X ₄	.23	.66	-.14	.70	.07
X ₅	.65	-.23	-.19	-.10	.69

variable clustering

$$PC_1 = -.25 * X_1 + .21 * X_2 + -.65 * X_3 + .23 * X_4 + .65 * X_5$$

Divisive Clustering





divisive clustering

2nd Eigenvalue

1.7

1

0



$\{X_1, X_2, X_3, X_4, X_5\}$

principal components analysis



Cutoff?

divisive clustering

2nd Eigenvalue

1.7

1

.7



0

$\{X_1, X_2, X_3, X_4, X_5\}$

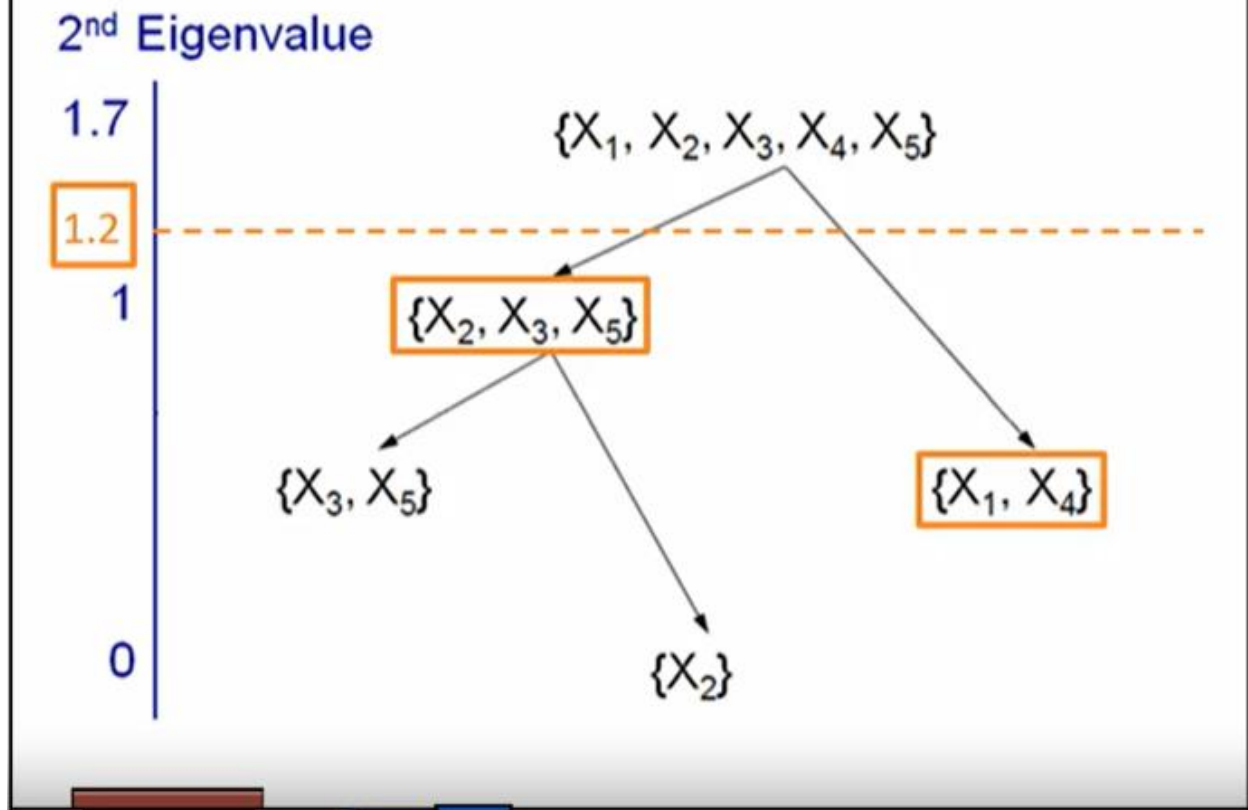
$\{X_2, X_3, X_5\}$

$\{X_3, X_5\}$

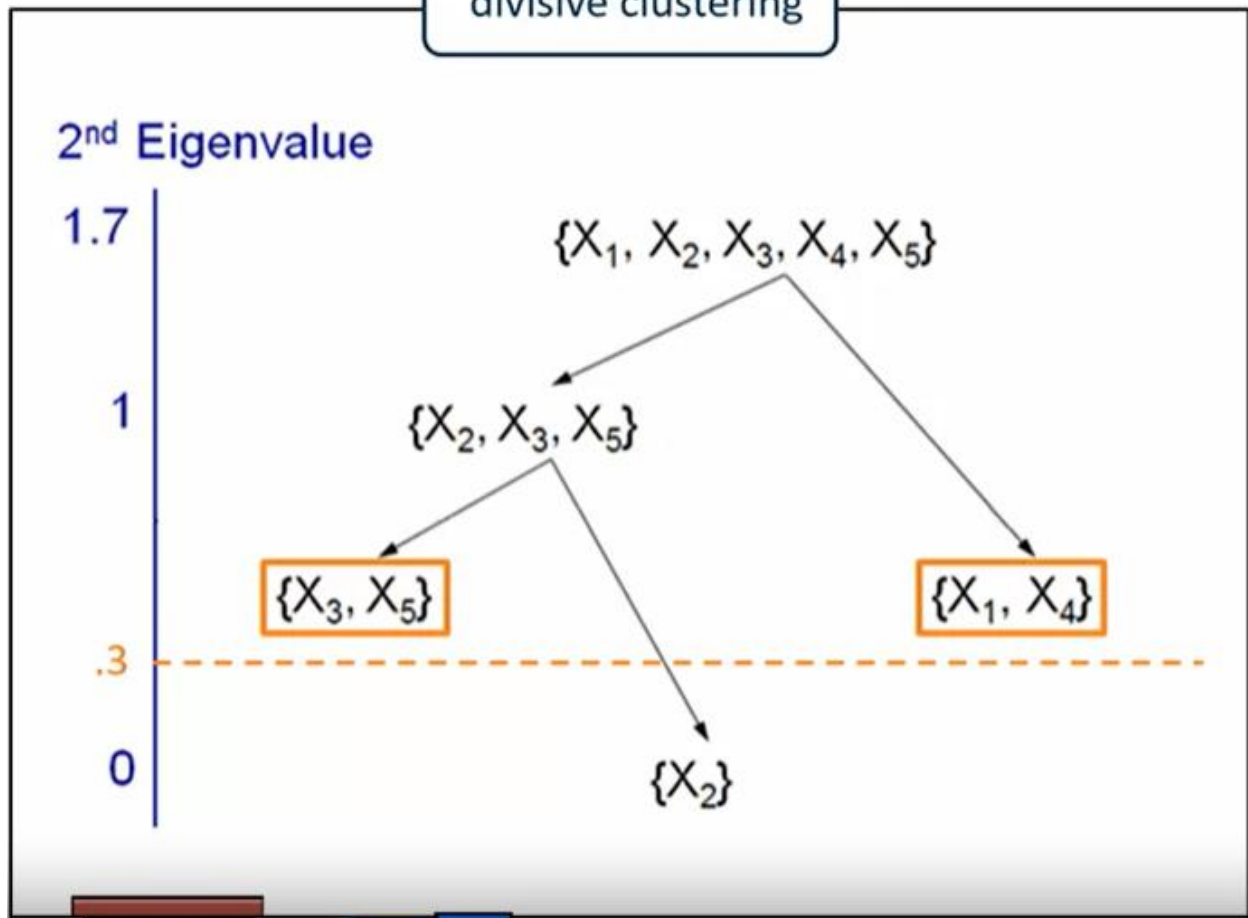
$\{X_1, X_4\}$

$\{X_2\}$

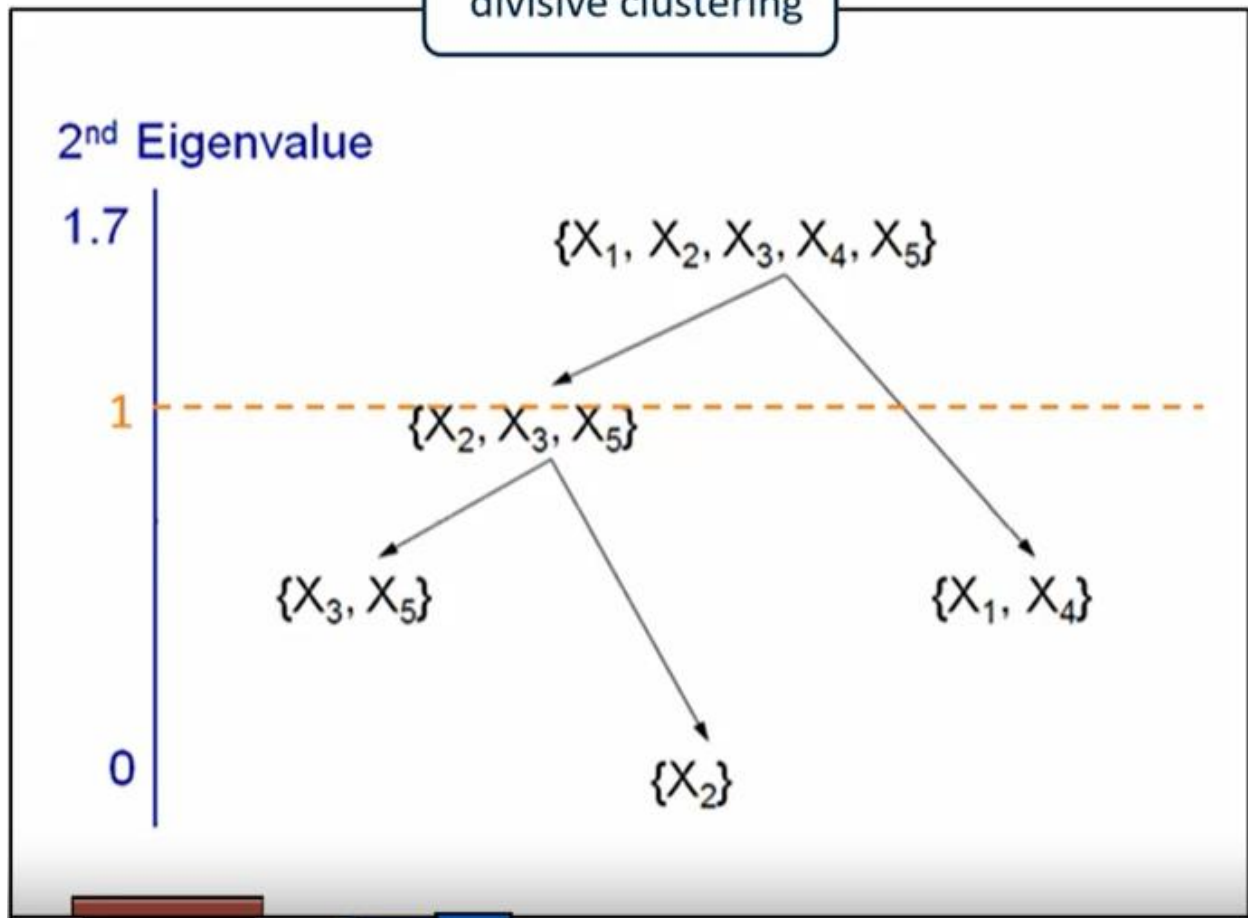
divisive clustering



divisive clustering

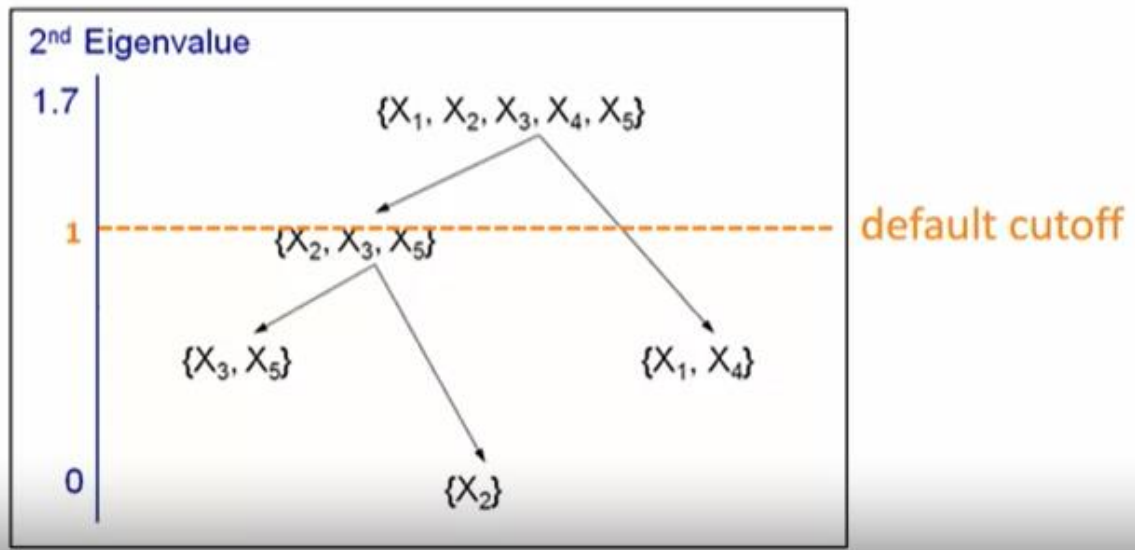


divisive clustering



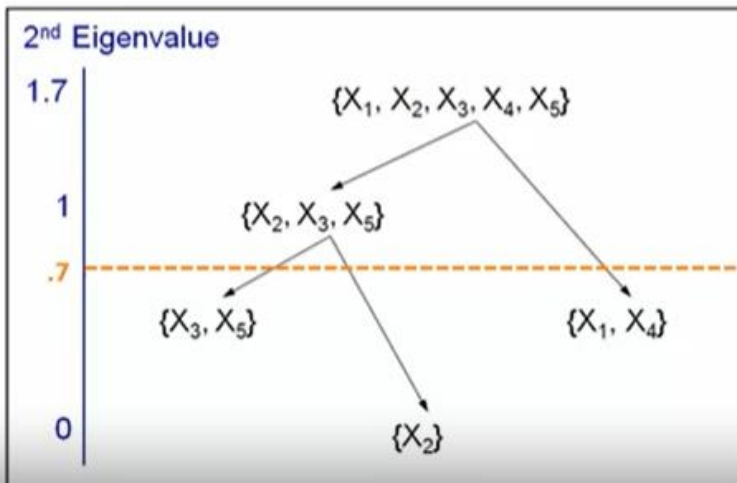
PROC VARCLUS Syntax

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



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

MAXEIGEN= n



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

SHORT



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



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

collapse the
levels

categorical



Dummy Variables

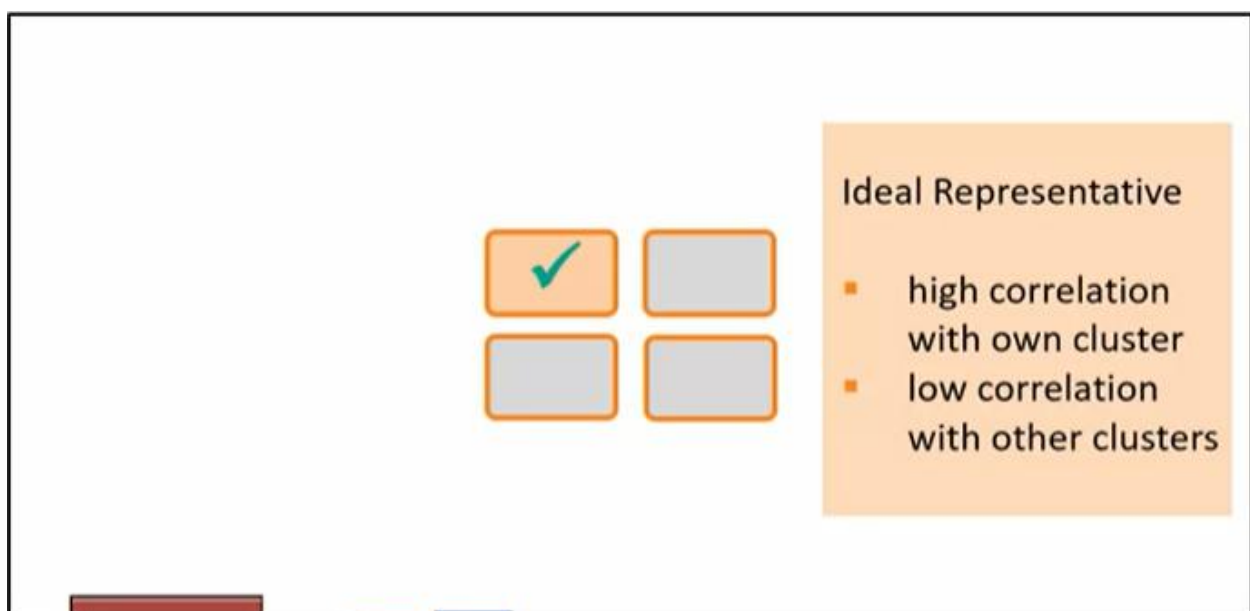
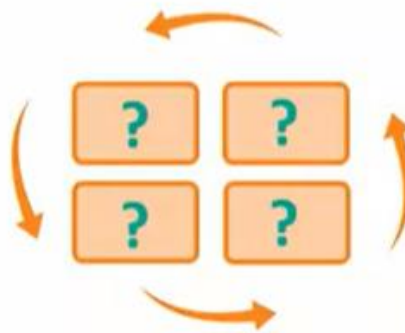
D _A	D _B	D _C
0	0	0
0	1	0
⋮	⋮	⋮

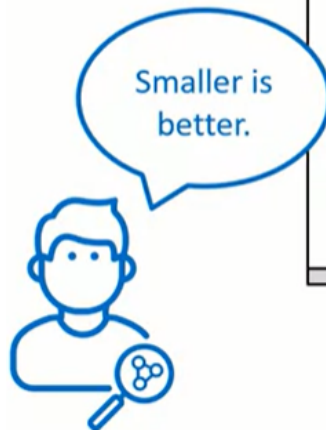
```
PROC VARCLUS DATA=SAS-data-set <options>;  
XVAR variables;  
RUN;
```

Selecting a Representative Variable from Each Cluster

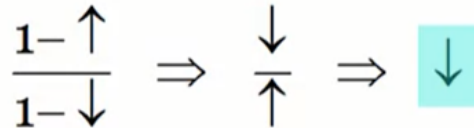
Variable Clustering

Step 1	Identify clusters of variables.
→ Step 2	Select a variable from each cluster.





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



Ideal Representative

- high correlation with own cluster
- low correlation with other clusters

Criteria for Input Selection

- $1 - R^2$ ratio
- subject-matter knowledge
- high correlation between input and target
- variables that cost the least amount of money
- variables that your peers and management think are important to control for

Demo Reducing Redundancy by Clustering Variables

```
pmlr03d04.sas
/* Use the ODS OUTPUT statement to generate data sets based on the variab.
   clustering results and the clustering summary. */

ods select none;
ods output clusterquality=work.summary
           rsquare=work.clusters;

proc varclus data=work.train_imputed_swoe maxeigen=.7 hi;
  var &inputs branch_swoe miacctag
      miphone mipos miposamt miinv
      miinvbal micc miccbal miccpure
      miincome mihmown milores mihmval
      miage micrscor;
run;
ods select all;
```

64 potential inputs

- * Cluster the numeric variables in the training data set.
- * Print the table of the R-square statistics from the last iteration of PROC VARCLUS.
- * Print the table showing the proportion of variation explained by the clusters.



```
/* Use the CALL SYMPUT function to create a macro variable:&NVAR =
   the number of of clusters. This is also the number of variables
   in the analysis, going forward. */

%global nvar;
data _null_;
  set work.summary;
  call symput('nvar',compress(NumberOfClusters));
run;

title1 "Variables by Cluster";
proc print data=work.clusters noobs label split='*';
  where NumberOfClusters=&nvar;
  var Cluster Variable RSquareRatio VariableLabel;
  label RSquareRatio="1 - RSquare*Ratio";
run;
```


Variables by Cluster

Cluster	Variable	1 - RSquare Ratio	Variable Label
Cluster 1	branch_swoe	0.4189	
	MIPhone	0.0042	
	MIPOS	0.0042	
	MIPOSAmt	0.0042	
	MIInv	0.0042	
	MIInvBal	0.0042	
	MICC	0.0042	
	MICCBal	0.0042	
	MICCPurc	0.0042	

Cluster 2	MIIncome	0.0074	
	MIHMOwn	0.0446	
	MILORes	0.0074	
	MIHMVal	0.0074	
	MIAge	0.0817	
Cluster 3	Dep	0.4122	Checking Deposits
	Checks	0.3536	Number of Checks
	Teller	0.5254	Teller Visits
Cluster 4	MTGBal	0.0344	Mortgage Balance
	CCBal	0.0322	Credit Card Balance
Cluster 5	MM	0.0926	Money Market
	MMBal	0.1068	Money Market Balance
	MMCred	0.4543	Money Market Credits

```

title1 "Variation Explained by Clusters";
proc print data=work.summary label;
run;

```

Variation Explained by Clusters

Obs	Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by Clusters	Minimum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster	Minimum R-squared for a Variable	Maximum 1-R ² Ratio for a Variable
1	1	8.839653	0.1449	0.1449	5.021094	0.0000	—
2	2	13.846715	0.2270	0.1956	3.457352	0.0000	1.0000
3	3	17.207611	0.2821	0.1373	2.625736	0.0000	1.4229
4	4	19.690396	0.3228	0.1373	2.314577	0.0001	1.4229
5	5	21.919904	0.3593	0.2239	2.059159	0.0001	1.3331
6	6	23.915604	0.3921	0.2239	1.965075	0.0003	1.2875
7	7	25.812779	0.4232	0.2239	1.607659	0.0003	1.2875
8	8	27.298020	0.4475	0.2239	1.476805	0.0003	1.3890
9	9	28.676857	0.4701	0.2239	1.410293	0.0003	1.4518
10	10	30.055834	0.4927	0.2211	1.383226	0.0003	1.4518

25	25	45.759249	0.7502	0.3616	0.966296	0.0813	1.1024
26	26	46.696723	0.7655	0.3616	0.956958	0.0813	1.1024
27	27	47.653680	0.7812	0.3616	0.947470	0.1320	1.1024
28	28	48.553991	0.7960	0.4489	0.921793	0.1320	1.1024
29	29	49.463897	0.8109	0.5234	0.882095	0.1320	1.1024
30	30	50.345236	0.8253	0.5234	0.860653	0.1896	1.1024
31	31	51.205517	0.8394	0.5234	0.843103	0.1896	1.1024
32	32	52.047720	0.8532	0.5234	0.841695	0.2686	0.8516
33	33	52.886839	0.8670	0.6108	0.778443	0.3604	0.6743
34	34	53.665282	0.8798	0.6283	0.770747	0.3604	0.6743
35	35	54.432136	0.8923	0.6283	0.743344	0.4973	0.5930
36	36	55.175480	0.9045	0.6326	0.734767	0.4973	0.5930
37	37	55.910247	0.9166	0.6442	0.693155	0.4973	0.5254

```
/* Choose a representative from each cluster. */
```

```
%global reduced;
%let reduced=branch_swoe MIINCOME Dep CCBal MM Income ILS POS NSF CD
                DDA LOC Age Inv InArea AcctAge Moved CRScore MICRScor
                IRABal MIAcctAg SavBal CashBk DDABal SDB InvBal CCPurc
                ATMAmt Sav CC Phone HMOwn DepAmt IRA MTG ATM LORes;
```

```
/* Run this code before demo l3d4 */
```

```
/* ===== */
```

```
/* Lesson 1, Section 1: l1d1.sas
```

Demonstration: Examining the Code for Generating

Descriptive Statistics and Frequency Tables */

```
/* ===== */
```

```
data work.develop;
```

```
    set pmlr.develop;
```

```
run;
```

```
%global inputs;
```

```
%let inputs=ACCTAGE DDA DDABAL DEP DEPAMT CASHBK
```

```
            CHECKS DIRDEP NSF NSFAMT PHONE TELLER
```

```
            SAV SAVBAL ATM ATMAMT POS POSAMT CD
```

```
CDBAL IRA IRABAL LOC LOCBAL INV  
INVBAL ILS ILSBAL MM MMBAL MMCRED MTG  
MTGBAL CC CCBAL CCPURC SDB INCOME  
HMOWN LORES HMVAL AGE CRSCORE MOVED  
INAREA;
```

```
proc means data=work.develop n nmiss mean min max;  
    var &inputs;  
run;
```

```
proc freq data=work.develop;  
    tables ins branch res;  
run;
```

```
/* ===== */  
/* Lesson 1, Section 2: l1d2.sas  
    Demonstration: Splitting the Data */  
/* ===== */
```

```
/* Sort the data by the target in preparation for stratified sampling. */
```

```
proc sort data=work.develop out=work.develop_sort;  
    by ins;  
run;
```

```
/* The SURVEYSELECT procedure will perform stratified sampling  
    on any variable in the STRATA statement. The OUTALL option  
    specifies that you want a flag appended to the file to
```

indicate selected records, not simply a file comprised
of the selected records. */

```
proc surveyselect noprint data=work.develop_sort
    samprate=.6667 stratumseed=restore
    out=work.develop_sample
    seed=44444 outall;

strata ins;

run;
```

```
/* Verify stratification. */
```

```
proc freq data=work.develop_sample;

    tables ins*selected;

run;
```

```
/* Create training and validation data sets. */
```

```
data work.train(drop=selected SelectionProb SamplingWeight)
    work.valid(drop=selected SelectionProb SamplingWeight);

set work.develop_sample;

if selected then output work.train;

else output work.valid;

run;
```

```
/* ===== */
```

```
/* Lesson 2, Section 1: l2d1.sas
```

Demonstration: Fitting a Basic Logistic

Regression Model, Parts 1 and 2 */

/* ===== */

title1 "Logistic Regression Model for the Variable Annuity Data Set";

proc logistic data=work.train

plots(only maxpoints=none)=(effect(clband x=(ddabal depamt checks res))

oddsratio (type=horizontalstat));

class res (param=ref ref='S') dda (param=ref ref='0');

model ins(event='1')=dda ddabal dep depamt

cashbk checks res / stb clodds=pl;

units ddabal=1000 depamt=1000 / default=1;

oddsratio 'Comparisons of Residential Classification' res / diff=all cl=pl;

effectplot slicefit(sliceby=dda x=ddabal) / noobs;

effectplot slicefit(sliceby=dda x=depamt) / noobs;

run;

title1;

/* ===== */

/* Lesson 2, Section 1: l2d2.sas

Demonstration: Scoring New Cases */

/* ===== */

/* Score a new data set with one run of the LOGISTIC procedure with the
SCORE statement. */

proc logistic data=work.train noprint;

class res (param=ref ref='S');

```
model ins(event='1')= res dda ddabal dep depamt cashbk checks;  
score data = pmlr.new out=work.scored1;  
run;
```

```
title1 "Predicted Probabilities from Scored Data Set";  
proc print data=work.scored1(obs=10);  
var p_1 dda ddabal dep depamt cashbk checks res;  
run;
```

```
title1 "Mean of Predicted Probabilities from Scored Data Set";  
proc means data=work.scored1 mean nolabels;  
var p_1;  
run;
```

```
/* Score a new data set with the OUTMODEL= amd INMODEL= options */
```

```
proc logistic data=work.train outmodel=work.scoredata noprint;  
class res (param=ref ref='S');  
model ins(event='1')= res dda ddabal dep depamt cashbk checks;  
run;
```

```
proc logistic inmodel=work.scoredata noprint;  
score data = pmlr.new out=work.scored2;  
run;
```

```
title1 "Predicted Probabilities from Scored Data Set";  
proc print data=work.scored2(obs=10);  
var p_1 dda ddabal dep depamt cashbk checks res;  
run;
```

```

/* Score a new data set with the CODE Statement */

proc logistic data=work.train noprint;
  class res (param=ref ref='S');
  model ins(event='1')= res dda ddabal dep depamt cashbk checks;
  code file="&PMLRfolder/pmlr_score.txt";
run;

data work.scored3;
  set pmlr.new;
  %include "&PMLRfolder/pmlr_score.txt";
run;

title1 "Predicted Probabilities from Scored Data Set";
proc print data=work.scored3(obs=10);
  var p_ins1 dda ddabal dep depamt cashbk checks res;
run;
title1 ;

/* ===== */
/* Lesson 2, Section 2: l2d3.sas
   Demonstration: Correcting for Oversampling */
/* ===== */

/* Specify the prior probability to correct for oversampling. */
%global pi1;
%let pi1=.02;

```



```
/* Correct predicted probabilities */
```

```
proc logistic data=work.train noprint;  
class res (param=ref ref='S');  
model ins(event='1')=dda ddabal dep depamt cashbk checks res;  
score data=pmlr.new out=work.scored4 priorevent=&pi1;  
run;
```

```
title1 "Adjusted Predicted Probabilities from Scored Data Set";  
proc print data=work.scored4(obs=10);  
var p_1 dda ddabal dep depamt cashbk checks res;  
run;
```

```
title1 "Mean of Adjusted Predicted Probabilities from Scored Data Set";  
proc means data=work.scored4 mean nolabels;  
var p_1;  
run;  
title1 ;
```

```
/* Correct probabilities in the Score Code */
```

```
proc logistic data=work.train noprint;  
class res (param=ref ref='S');  
model ins(event='1')=dda ddabal dep depamt cashbk checks res;  
/* File suffix "txt" is used so you can view the file */  
/* with a native text editor. SAS prefers "sas", but */  
/* when specified as a filename, SAS does not care. */  
code file="%PMLRfolder/pmlr_score_adj.txt";
```

```

run;

%global rho1;

proc SQL noprint;
    select mean(INS) into :rho1
    from work.train;
quit;

data new;
    set pmlr.new;
    off=log((((1-&pi1)*&rho1)/(&pi1*(1-&rho1))));
run;

data work.scored5;
    set work.new;
    %include "&PMLRfolder/pmlr_score_adj.txt";
    eta=log(p_ins1/p_ins0) - off;
    prob=1/(1+exp(-eta));
run;

title1 "Adjusted Predicted Probabilities from Scored Data Set";
proc print data=scored5(obs=10);
    var prob dda ddabal dep depamt cashbk checks res;
run;
title1 ;

/* ===== */
/* Lesson 3, Section 1: l3d1.sas

```

Demonstration: Imputing Missing Values

```
/* ===== */
```

```
title1 "Variables with Missing Values";
```

```
proc print data=work.train(obs=15);
```

```
var ccbal ccpurc income hmown;
```

```
run;
```

```
title1 ;
```

```
/* Create missing indicators */
```

```
data work.train_mi(drop=i);
```

```
set work.train;
```

```
/* name the missing indicator variables */
```

```
array mi{*} MIAcctAg MIPhone MIPOS MIPOSamt
```

```
MIInv MIInvBal MICC MICCBal
```

```
MICCPurc MIIncome MIHMOwn MILORes
```

```
MIHMVal MIAge MICRScor;
```

```
/* select variables with missing values */
```

```
array x{*} acctage phone pos posamt
```

```
inv invbal cc ccbal
```

```
ccpurc income hmown lores
```

```
hmval age crscore;
```

```
do i=1 to dim(mi);
```

```
mi{i}=(x{i}=.);
```

```
nummiss+mi{i};
```

```
end;
```

```
run;
```

```
/* Impute missing values with the median */
```

```
proc stdize data=work.train_mi reponly method=median out=work.train_imputed;
  var &inputs;
run;
```

```
title1 "Imputed Values with Missing Indicators";
proc print data=work.train_imputed(obs=12);
  var ccbal miccbal ccpurc miccpurc income miincome hmown mihmown nummiss;
run;
title1;
```

```
/* ===== */
/* Lesson 3, Section 2: l3d2a.sas
   Demonstration: Collapsing the Levels of a
   Nominal Input, Part 1          */
/* ===== */
```

```
proc means data=work.train_imputed noprint nway;
  class branch;
  var ins;
  output out=work.level mean=prop;
run;
```

```
title1 "Proportion of Events by Level";
proc print data=work.level;
run;
```

```
/* Use ODS to output the ClusterHistory output object into a data set
```

```

named "cluster." */

ods output clusterhistory=work.cluster;

proc cluster data=work.level method=ward outtree=work.fortree
    plots=(dendrogram(vertical height=rsq));
    freq_freq_;
    var prop;
    id branch;
run;

/* ===== */
/* Lesson 3, Section 2: l3d2b.sas
    Demonstration: Collapsing the Levels of a
    Nominal Input, Part 2          */
/* ===== */

/* Use the FREQ procedure to get the Pearson Chi^2 statistic of the
    full BRANCH*INS table. */

proc freq data=work.train_imputed noprint;
    tables branch*ins / chisq;
    output out=work.chi(keep=_pchi_) chisq;
run;

/* Use a one-to-many merge to put the Chi^2 statistic onto the clustering
    results. Calculate a (log) p-value for each level of clustering. */

```

```

data work.cutoff;

  if _n_=1 then set work.chi;

  set work.cluster;

  chisquare=_pchi_*rsquared;

  degfree=numberofclusters-1;

  logpvalue=logsf('CHISQ',chisquare,degfree);

run;

/* Plot the log p-values against number of clusters. */

title1 "Plot of the Log of the P-Value by Number of Clusters";

proc sgplot data=work.cutoff;

  scatter y=logpvalue x=numberofclusters

    / markerattrs=(color=blue symbol=circlefilled);

  xaxis label="Number of Clusters";

  yaxis label="Log of P-Value" min=-120 max=-85;

run;

title1 ;

/* Create a macro variable (&ncl) that contains the number of clusters
   associated with the minimum log p-value. */

proc sql;

  select NumberOfClusters into :ncl

  from work.cutoff

  having logpvalue=min(logpvalue);

quit;

```

```

proc tree data=work.fortree nclusters=&ncl out=work.clus noprint;
    id branch;
run;

proc sort data=work.clus;
    by clusname;
run;

title1 "Levels of Branch by Cluster";
proc print data=work.clus;
    by clusname;
    id clusname;
run;
title1 ;

/* The DATA Step creates the scoring code to assign the branches to a cluster. */

filename brclus "&PMLRfolder/branch_clus.sas";

data _null_;
    file brclus;
    set work.clus end=last;
    if _n_=1 then put "select (branch);";
    put "  when ('" branch +(-1) "') branch_clus = '" cluster +(-1) "'";
    if last then do;
        put "  otherwise branch_clus = 'U';" / "end;";
    end;
run;

```

```

data work.train_imputed_greenacre;

    set work.train_imputed;

    %include brclus / source2;

run;


/* ===== */
/* Lesson 3, Section 2: l3d3.sas
    Demonstration: Computing the Smoothed Weight of Evidence */
/* ===== */


/* Rho1 is the proportion of events in the training data set. */
%global rho1;

proc sql noprint;

    select mean(ins) into :rho1

    from work.train_imputed;

run;


/* The output data set from PROC MEANS will have the number of
    observations and events for each level of branch. */


proc means data=work.train_imputed sum nway noprint;

    class branch;

    var ins;

    output out=work.counts sum=events;

run;


/* The DATA Step creates the scoring code that assigns each branch to
    a value of the smoothed weight of evidence. */

```



```

filename brswoe "&PMLRfolder/swoe_branch.sas";

data _null_;
    file brswoe;

    set work.counts end=last;

    logit=log((events + &rho1*24)/(_FREQ_ - events + (1-&rho1)*24));

    if _n_=1 then put "select (branch);" ;
    put " when ('" branch +(-1) "' ) branch_swoe = " logit ";" ;

    if last then do;
        logit=log(&rho1/(1-&rho1));
        put " otherwise branch_swoe = " logit ";" / "end;";
    end;
run;

data work.train_imputed_swoe;

    set work.train_imputed;

    %include brswoe / source2;
run;

```

```

/* ===== */
/* Lesson 3, Section 3: l3d4.sas

Demonstration: Reducing Redundancy by Clustering Variables

[m643_3_i; derived from pmlr03d04.sas] */
/* ===== */

/* Use the ODS OUTPUT statement to generate data sets based on the variable
clustering results and the clustering summary. */

ods select none;

ods output clusterquality=work.summary
           rsquare=work.clusters;

proc varclus data=work.train_imputed_swoe maxeigen=.7 hi;
  var &inputs branch_swoe miacctag
      miphone mipos miposamt miinv
      miinvbal micc miccbal miccpurc
      miincome mihmown milores mihmval
      miage micrscor;
run;

ods select all;

/* Use the CALL SYMPUT function to create a macro variable:&NVAR =
the number of of clusters. This is also the number of variables
in the analysis, going forward. */

%global nvar;

data _null_;
  set work.summary;

```

```

    call symput('nvar',compress(NumberOfClusters));
run;

title1 "Variables by Cluster";
proc print data=work.clusters noobs label split='*';
    where NumberOfClusters=&nvar;
    var Cluster Variable RSquareRatio VariableLabel;
    label RSquareRatio="1 - RSquare*Ratio";
run;
title1 ;

title1 "Variation Explained by Clusters";
proc print data=work.summary label;
run;

/* Choose a representative from each cluster. */
%global reduced;
%let reduced=branch_swowe MIINCOME Dep CCBal MM Income ILS POS NSF CD
    DDA LOC Age Inv InArea AcctAge Moved CRScore MICRScor
    IRABal MIAcctAg SavBal CashBk DDABal SDB InvBal CCPurc
    ATMAmt Sav CC Phone HMOwn DepAmt IRA MTG ATM LORes;

```

```

/* ===== */
/* Lesson 1, Practice 1
Practice: Exploring the Veterans' Organization Data
Used in the Practices */
/* ===== */

```

```
data pmlr.pva(drop=control_number
               MONTHS_SINCE_LAST_PROM_RESP
               FILE_AVG_GIFT
               FILE_CARD_GIFT);

set pmlr.pva_raw_data;

STATUS_FL=REGENCY_STATUS_96NK in("F","L");
STATUS_ES=REGENCY_STATUS_96NK in("E","S");
home01=(HOME_OWNER="H");
nses1=(SES="1");
nses3=(SES="3");
nses4=(SES="4");
nses_=(SES="?");
nurbr=(URBANICITY="R");
nurbu=(URBANICITY="U");
nurbs=(URBANICITY="S");
nurbt=(URBANICITY="T");
nurb_=(URBANICITY="?");

run;

proc contents data=pmlr.pva;

run;
```

```
proc means data=pmlr.pva mean nmiss max min;
    var _numeric_;
run;
```

```
proc freq data=pmlr.pva nlevels;
    tables _character_;
run;
```

```
/* ===== */
/* Lesson 1, Practice 2
    Practice: Splitting the Data          */
/* ===== */
```

```
proc sort data=pmlr.pva out=work.pva_sort;
    by target_b;
run;
```

```
proc surveyselect noprint data=work.pva_sort
    samprate=0.5 out=pva_sample seed=27513
    outall stratumseed=restore;
    strata target_b;
run;
```

```
data pmlr.pva_train(drop=selected SelectionProb SamplingWeight)
    pmlr.pva_valid(drop=selected SelectionProb SamplingWeight);
set work.pva_sample;
if selected then output pmlr.pva_train;
```

```

else output pmlr.pva_valid;
run;

/* ===== */
/* Lesson 2, Practice 1
Practice: Fitting a Logistic Regression Model */
/* ===== */

/* Modifications for your SAS software:
-----

(Optional) To avoid a warning in the log about the
suppression of plots that have more than 5000
observations, you can add the MAXPOINTS= option
to the PROC LOGISTIC statement like this:
plots(maxpoints=none only). Omitting the
MAXPOINTS= option does not affect the results
of the practices in this course.

*/

%global ex_pi1;
%let ex_pi1=0.05;

title1 "Logistic Regression Model of the Veterans' Organization Data";
proc logistic data=pmlr.pva_train plots(only)=
    (effect(clband x=(pep_star recent_avg_gift_amt
    frequency_status_97nk)) oddsratio (type=horizontalstat));
class pep_star (param=ref ref='0');
model target_b(event='1')=pep_star recent_avg_gift_amt

```

```

        frequency_status_97nk / clodds=pl;
    effectplot slicefit(sliceby=pep_star x=recent_avg_gift_amt) / noobs;
    effectplot slicefit(sliceby=pep_star x=frequency_status_97nk) / noobs;
    score data=pmlr.pva_train out=work.scopva_train priorevent=&ex_pi1;
run;

title1 "Adjusted Predicted Probabilities of the Veteran's Organization Data";
proc print data=work.scopva_train(obs=10);
    var p_1 pep_star recent_avg_gift_amt frequency_status_97nk;

run;

title;

```

```

/* ===== */
/* Lesson 3, Practice 1
    Practice: Imputing Missing Values          */
/* ===== */

```

```

data pmlr.pva_train_mi(drop=i);
    set pmlr.pva_train;
    /* name the missing indicator variables */
    array mi{*} mi_DONOR_AGE mi_INCOME_GROUP
        mi_WEALTH_RATING;
    /* select variables with missing values */
    array x{*} DONOR_AGE INCOME_GROUP WEALTH_RATING;
    do i=1 to dim(mi);
        mi{i}=(x{i}=.);
        nummiss+mi{i};
    end;

```

```

end;

run;

proc rank data=pmlr.pva_train_mi out=work.pva_train_rank
    groups=3;
    var recent_response_prop recent_avg_gift_amt;
    ranks grp_resp grp_amt;
run;

proc sort data=work.pva_train_rank out=work.pva_train_rank_sort;
    by grp_resp grp_amt;
run;

proc stdize data=work.pva_train_rank_sort method=median
    reonly out=pmlr.pva_train_imputed;
    by grp_resp grp_amt;
    var DONOR_AGE INCOME_GROUP WEALTH_RATING;
run;

options nolabel;

proc means data=pmlr.pva_train_imputed median;
    class grp_resp grp_amt;
    var DONOR_AGE INCOME_GROUP WEALTH_RATING;
run;

options label;

/* ===== */
/* Lesson 3, Practice 2

```


Practice: Collapsing the Levels of a Nominal Input

Note: After you submit this code, a note in the log indicates that argument 3 to the LOGSDF function is invalid. You can ignore this note; it is not important for this analysis. The note pertains to the situation in which the number of clusters is 1. In this case, the degrees of freedom is 0 (degrees of freedom is equal to the number of clusters minus 1) and the mathematical operation cannot be performed in the LOGSDF function. Therefore, the log of the p-value is set to missing. */

```
/* ===== */
```

```
proc means data=pmlr.pva_train_imputed noprint nway;
  class cluster_code;
  var target_b;
  output out=work.level mean=prop;
run;
```

```
ods output clusterhistory=work.cluster;
```

```
proc cluster data=work.level method=ward
  outtree=work.fortree
  plots=(dendrogram(horizontal height=rsq));
  freq _freq_;
  var prop;
  id cluster_code;
run;
```

```

proc freq data=pmlr.pva_train_imputed noprint;

    tables cluster_code*target_b / chisq;

    output out=work.chi(keep=_pchi_) chisq;

run;


data work.cutoff;

    if _n_=1 then set work.chi;

    set cluster;

    chisquare=_pchi_*rsquared;

    degfree=numberofclusters-1;

    logpvalue=logsf('CHISQ',chisquare,degfree);

run;


title1 "Plot of the Log of the P-Value by Number of Clusters";

proc sgplot data=work.cutoff;

    scatter y=logpvalue x=numberofclusters

        / markerattrs=(color=blue symbol=circlefilled);

    xaxis label="Number of Clusters";

    yaxis label="Log of P-Value" min=-40 max=0;

run;


title1;


%global ncl;


proc sql;

    select NumberOfClusters into :ncl

    from work.cutoff

```

```

    having logpvalue=min(logpvalue);
quit;

proc tree data=work.fortree nclusters=&ncl
    out=work.clus noprint;
    id cluster_code;
run;

proc sort data=work.clus;
    by clusname;
run;

title1 "Cluster Assignments";
proc print data=work.clus;
    by clusname;
    id clusname;
run;

filename clcode "&PMLRfolder/cluster_code.sas";

data _null_;
    file clcode;
    set work.clus end=last;
    if _n_=1 then put "select (cluster_code);";
    put " when ('" cluster_code +(-1) '" )
        cluster_clus='" cluster +(-1) '"';
    if last then do;
        put " otherwise cluster_clus='U';" / "end;";
    end;
end;

```

```
run;
```

```
data pmlr.pva_train_imputed_clus;
```

```
    set pmlr.pva_train_imputed;
```

```
    %include clcode;
```

```
run;
```

```
/* ===== */
```

```
/* Lesson 3, Practice 3
```

```
    Practice: Computing the Smoothed Weight of Evidence */
```

```
/* ===== */
```

```
%global rho1_ex;
```

```
proc sql noprint;
```

```
    select mean(target_b) into :rho1_ex
```

```
    from pmlr.pva_train_imputed;
```

```
run;
```

```
proc means data=pmlr.pva_train_imputed
```

```
    sum nway noprint;
```

```
    class cluster_code;
```

```
    var target_b;
```

```
    output out=work.counts sum=events;
```

```
run;
```

```
filename clswoe "&PMLRfolder/swoe_cluster.sas";
```

```
data _null_;
```

```

file clswoe;

set work.counts end=last;

  logit=log((events + &rho1_ex*24)/
    (_FREQ_ - events + (1-&rho1_ex)*24));
if _n_=1 then put "select (cluster_code);" ;
put "  when ('" cluster_code +(-1) '" ) cluster_swoe=" logit ";" ;
if last then do;
  logit=log(&rho1_ex/(1-&rho1_ex));
  put "  otherwise cluster_swoe=" logit ";" / "end;";
end;
run;

data pmlr.pva_train_imputed_swoe;
  set pmlr.pva_train_imputed;
  %include clswoe;
run;

title;

proc print data=pmlr.pva_train_imputed_swoe(obs=1);
  where cluster_code = "01";
  var cluster_code cluster_swoe;
run;

```

/* Practice: l3p4.sas step 1 */

**%let ex_inputs= MONTHS_SINCE_ORIGIN
DONOR_AGE IN_HOUSE INCOME_GROUP PUBLISHED_PHONE
MOR_HIT_RATE WEALTH_RATING MEDIAN_HOME_VALUE
MEDIAN_HOUSEHOLD_INCOME PCT_OWNER_OCCUPIED
PER_CAPITA_INCOME PCT_MALE_MILITARY
PCT_MALE_VETERANS PCT_VIETNAM_VETERANS
PCT_WWII_VETERANS PEP_STAR RECENT_STAR_STATUS
FREQUENCY_STATUS_97NK RECENT_RESPONSE_PROP
RECENT_AVG_GIFT_AMT RECENT_CARD_RESPONSE_PROP
RECENT_AVG_CARD_GIFT_AMT RECENT_RESPONSE_COUNT
RECENT_CARD_RESPONSE_COUNT LIFETIME_CARD_PROM
LIFETIME_PROM LIFETIME_GIFT_AMOUNT
LIFETIME_GIFT_COUNT LIFETIME_AVG_GIFT_AMT
LIFETIME_GIFT_RANGE LIFETIME_MAX_GIFT_AMT
LIFETIME_MIN_GIFT_AMT LAST_GIFT_AMT
CARD_PROM_12 NUMBER_PROM_12 MONTHS_SINCE_LAST_GIFT
MONTHS_SINCE_FIRST_GIFT STATUS_FL STATUS_ES
home01 nses1 nses3 nses4 nses_ nurbr nurbu nurbs
nurbt nurb_;**

/* Solution for l3p4 */

/* step 2 */

```
%let ex_inputs= MONTHS_SINCE_ORIGIN
DONOR_AGE IN_HOUSE INCOME_GROUP PUBLISHED_PHONE
MOR_HIT_RATE WEALTH_RATING MEDIAN_HOME_VALUE
MEDIAN_HOUSEHOLD_INCOME PCT_OWNER_OCCUPIED
PER_CAPITA_INCOME PCT_MALE_MILITARY
PCT_MALE_VETERANS PCT_VIETNAM_VETERANS
PCT_WWII_VETERANS PEP_STAR RECENT_STAR_STATUS
FREQUENCY_STATUS_97NK RECENT_RESPONSE_PROP
RECENT_AVG_GIFT_AMT RECENT_CARD_RESPONSE_PROP
RECENT_AVG_CARD_GIFT_AMT RECENT_RESPONSE_COUNT
RECENT_CARD_RESPONSE_COUNT LIFETIME_CARD_PROM
LIFETIME_PROM LIFETIME_GIFT_AMOUNT
LIFETIME_GIFT_COUNT LIFETIME_AVG_GIFT_AMT
LIFETIME_GIFT_RANGE LIFETIME_MAX_GIFT_AMT
LIFETIME_MIN_GIFT_AMT LAST_GIFT_AMT
CARD_PROM_12 NUMBER_PROM_12 MONTHS_SINCE_LAST_GIFT
MONTHS_SINCE_FIRST_GIFT STATUS_FL STATUS_ES
home01 nses1 nses3 nses4 nses_ nurbr nurbu nurbs
nurbt nurb_;
```

/* step 3 */

```
ods select none;
ods output clusterquality=work.summary
```

```

rsquare=work.clusters;

proc varclus data=pmlr.pva_train_imputed_swoe
    hi maxeigen=0.70;
    var &ex_inputs mi_DONOR_AGE mi_INCOME_GROUP
        mi_WEALTH_RATING cluster_swoe;
run;

ods select all;

/* step 4 */

data _null_;
    set work.summary;
    call symput('nvar',compress(NumberOfClusters));
run;

/* step 5 */

title1 "Variables by Cluster";
proc print data=work.clusters noobs label split='*';
    where NumberOfClusters=&nvar;
    var Cluster Variable RSquareRatio;
    label RSquareRatio="1 - RSquare*Ratio";
run;

title1 "Variation Explained by Clusters";

```



```
proc print data=work.summary label;
run;
title1;
```

Variables by Cluster		
Cluster	Variable	1 - RSquare Ratio
Cluster 1	MONTHS_SINCE_ORIGIN	0.1694
	LIFETIME_CARD_PROM	0.0964
	LIFETIME_PROM	0.1097
	LIFETIME_GIFT_AMOUNT	0.6593
	LIFETIME_GIFT_COUNT	0.4943
	MONTHS_SINCE_FIRST_GIFT	0.1536
	mi_WEALTH_RATING	0.5208
Cluster 2	RECENT_AVG_GIFT_AMT	0.4247
	RECENT_AVG_CARD_GIFT_AMT	0.6359
	LIFETIME_GIFT_RANGE	0.3966
	LIFETIME_MAX_GIFT_AMT	0.1463
	LAST_GIFT_AMT	0.4065
Cluster 3	MEDIAN_HOME_VALUE	0.2932
	MEDIAN_HOUSEHOLD_INCOME	0.2241
	PER_CAPITA_INCOME	0.1872
	nses1	0.5398
Cluster 4	FREQUENCY_STATUS_97NK	0.3979
	RECENT_RESPONSE_PROP	0.2056
	RECENT_CARD_RESPONSE_PROP	0.3633
	RECENT_RESPONSE_COUNT	0.2453
	RECENT_CARD_RESPONSE_COUNT	0.2223
Cluster 5	CARD_PROM_12	0.3716
	NUMBER_PROM_12	0.2871
	MONTHS_SINCE_LAST_GIFT	0.5233

Cluster 6	nses_	0.0000
	nurb_	0.0000
Cluster 7	mi_DONOR_AGE	0.3349
	mi_INCOME_GROUP	0.4338
Cluster 8	PCT_MALE_VETERANS	0.0000
Cluster 9	PCT_VIETNAM_VETERANS	0.3168
	PCT_WWII_VETERANS	0.3527
Cluster 10	LIFETIME_AVG_GIFT_AMT	0.3681
	LIFETIME_MIN_GIFT_AMT	0.1462
Cluster 11	nses3	0.2840
	cluster_swoe	0.3086
Cluster 12	PEP_STAR	0.3905
	STATUS_ES	0.3201
Cluster 13	PCT_MALE_MILITARY	0.0000
Cluster 14	nurbu	0.0000
Cluster 15	nurbt	0.0000
Cluster 16	home01	0.0000
Cluster 17	nurbr	0.0000
Cluster 18	DONOR_AGE	0.0000
Cluster 19	STATUS_FL	0.0000
Cluster 20	MOR_HIT_RATE	0.0000
Cluster 21	nses4	0.0000
Cluster 22	INCOME_GROUP	0.0000
Cluster 23	RECENT_STAR_STATUS	0.0000
Cluster 24	IN_HOUSE	0.0000
Cluster 25	WEALTH_RATING	0.0000
Cluster 26	PUBLISHED_PHONE	0.0000

Variation Explained by Clusters							
Obs	Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by Clusters	Minimum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster	Minimum R-squared for a Variable	Maximum 1-R ² Ratio for a Variable
1	1	7.932328	0.1497	0.1497	5.826522	0.0000	—
2	2	12.645612	0.2386	0.2177	4.131285	0.0000	1.0000
3	3	16.730624	0.3157	0.2603	3.075172	0.0007	1.0041
4	4	19.665398	0.3710	0.2603	2.436935	0.0005	1.0278
5	5	21.840212	0.4121	0.3119	1.949434	0.0005	1.0616
6	6	23.775930	0.4486	0.3119	1.795110	0.0005	1.1896
7	7	25.538296	0.4819	0.3119	1.486987	0.0005	1.3666
8	8	26.833836	0.5063	0.3576	1.416451	0.0005	1.2155
9	9	28.070094	0.5296	0.3576	1.303067	0.0005	1.2155
10	10	28.813747	0.5437	0.3576	1.089054	0.0020	1.2155
11	11	29.827100	0.5628	0.3576	1.076913	0.0020	1.2155
12	12	30.722383	0.5797	0.3576	1.017719	0.0024	1.2155
13	13	31.720428	0.5985	0.3576	1.003693	0.0045	1.2155
14	14	32.715792	0.6173	0.3576	0.998028	0.0045	1.2182
15	15	33.713820	0.6361	0.3576	0.990889	0.0590	1.2182
16	16	34.664584	0.6540	0.4465	0.987679	0.0590	0.9570
17	17	35.628268	0.6722	0.4465	0.958577	0.0590	0.9570
18	18	36.579098	0.6902	0.4465	0.923508	0.1678	0.9477
19	19	37.415611	0.7060	0.4465	0.917929	0.1736	0.8674
20	20	38.286771	0.7224	0.4604	0.888606	0.1736	0.8674
21	21	39.168115	0.7390	0.5729	0.855329	0.1736	0.8674
22	22	39.964918	0.7541	0.5833	0.853670	0.1736	0.8674
23	23	40.818171	0.7702	0.5833	0.787615	0.3566	0.7397
24	24	41.488014	0.7828	0.6105	0.780909	0.3566	0.7397
25	25	42.211853	0.7965	0.6105	0.778992	0.5030	0.6593
26	26	42.990846	0.8111	0.6312	0.737511	0.5030	0.6593
27	27	43.728356	0.8251	0.6380	0.724100	0.5030	0.6593
28	28	44.452456	0.8387	0.6676	0.687113	0.5030	0.6593