Data Analytics for Materials Science

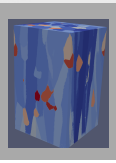
27-737

A.D. (Tony) Rollett, Amit Verma, Richard A. LeSar (Iowa State Univ.)

Dept. Materials Sci. Eng., Carnegie Mellon University

Principal Component Analysis (PCA)

Lecture 8, part 2



Revised: 3rd Mar. 2021

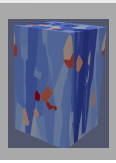
The eigensystem equation is $\mathbf{Cv} = \lambda\mathbf{v}$.

Note that letting $\mathbf{v} \rightarrow -\mathbf{v}$ does not change the equation — λ remains the same.

The signs of the eigenvectors can be different depending on what software was used to calculate them.

The basic features of the scores and loading plots (discussed soon) will be the same, but the signs of the plots may be different.

Note: solving the eigensystem equation for the covariance matrix is also known as *spectral decomposition*.

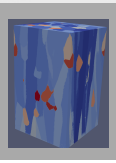


Recap:

A common type of dataset: 44 semiconductor compounds and 6 descriptors.

We have seen what regression can do to help understand the data.

	atomic number	melting pt. (C)	# valence e-	Δ in radii	electro-negativity	lattice const (Å)		atomic number	melting pt. (C)	# valence e-	Δ in radii	electro-negativity	lattice const (Å)
AlN	10	498	4	1.135	-1.21	3.11	$(ZnMg)_{0.5}$ S	18.5	598	6.5	0.78	-1.21	5.52
AlP	14	625	4	0.435	-0.68	5.47	$(SSe)_{0.5}$ Mg	18.5	682	4	-0.93	1.34	5.81
AlAs	23	1012	4	0.26	-0.63	5.66	$(SSe)_{0.5}$ Z	27.5	567	9	-0.78	1.21	5.52
AlSb	32	919	4	-0.09	-0.5	6.14	$(ZnMg)_{0.5}$ Se	27.5	651	6.5	0.595	-1.1	5.78
GaN	19	183	4	1.155	-1.15	3.16	$(ZnCd)_{0.5}$ Se	36.5	569	9	0.595	-1.1	5.86
GaP	23	310	4	0.455	-0.62	5.45	$(SeTe)_{0.5}$ Zn	36.5	650	9	-0.595	1.1	5.9
GaAs	32	545	2.5	0.28	-0.57	5.65	$(SeTe)_{0.5}$ Cd	45.5	601	9	-0.93	1.14	6.28
GaSb	41	603	4	-0.07	-0.44	6.1	$(ZnCd)_{0.5}$ Te	45.5	683	9	0.21	-0.94	6.27
InN	28	246	4	1.51	-1.22	3.54	$(AlGa)_{0.5}$ P	18.5	467	4	0.435	-0.68	5.46
InP	32	373	4	0.81	-0.69	5.87	$(PAs)_{0.5}$ Ga	27.5	503	4	-0.455	0.62	5.61
InAs	41	760	4	0.635	-0.64	6.06	$(AlGa)_{0.5}$ As	27.5	854	4	0.26	-0.63	5.65
InSb	50	667	4	0.285	-0.51	6.48	$(GaIn)_{0.5}$ P	27.5	341	4	0.455	-0.62	5.68
ZnS	23	540	9	0.78	-1.21	5.41	$(AlIn)_{0.5}$ P	23	499	4	0.435	-0.68	5.69
ZnSe	32	593	9	0.595	-1.1	5.67	$(AlG)_{0.5Zn}$As	27.5	965	4	-0.26	0.63	5.88
ZnTe	41	707	9	0.21	-0.94	6.1	$(AlZn)_{0.5}$ As	27.25	951	6.25	0.26	-0.63	5.87
CdSe	41	544	9	0.93	-1.14	6.05	$(AsSb)_{0.5}$ Ga	36.5	650	4	-0.28	0.57	5.85
CdTe	50	658	9	0.454	-0.98	6.48	$(AlGa)_{0.5}$ Sb	36.5	761	4	-0.09	-0.5	6.11
MgS	14	655	4	0.93	-1.34	5.7	$(GaIn)_{0.5}$ Sb	36.5	728	4	0.28	-0.57	5.82
MgSe	23	708	4	0.745	-1.23	5.9	$(PAs)_{0.5}$ In	36.5	566	4	-0.81	0.69	5.94
$(AlGa)_{0.5}$ N	14.5	340	4	1.135	-1.21	3.14	$(AlIn)_{0.5}$ Sb	41	793	4	-0.09	-0.5	6.27
$(AlIn)_{0.5}$ N	19	372	4	1.135	-1.21	3.32	$(GaIn)_{0.5}$ Sb	45.5	635	4	-0.07	-0.44	6.26
$(GaIn)_{0.5}$ N	23.5	214	4	1.155	-1.15	3.31	$(AsAb)_{0.5}$ In	45.5	713	4	-0.635	0.64	6.24



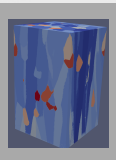
dataset-semiconductors-from-KrishnaRajan-talk-on-PCA-RALeSar.xlsx

Semiconductor compounds: Courtney of Krishna Rajan

Recap: to put all variables on the same scale and to eliminate issues with the different units in each data type, we *autoscale* the data:

$X'_{ki} = \dfrac{X_{ki} - \overline{X_i}}{\sigma_i}$, for each data entry with i = the type of data (e.g., *atomic number*).

- The vector of values of X'_{ki} will be called $\mathbf{X}'_i$ and we have: $\overline{\mathbf{X}'} = 0$ and $\sigma_{\mathbf{X}'_i} = 1$
- Create the data matrix as: $\mathbf{A}_N = \left[\mathbf{X}'_1 \quad \mathbf{X}'_2 \quad \mathbf{X}'_3 \quad \mathbf{X}'_4 \quad \mathbf{X}'_5 \quad \mathbf{X}'_6\right]$ in which each $\mathbf{X}'_i$ is a 44 component column vector of the autoscaled data type $\mathbf{X}'_i$.
- Autoscaling ensures that the variances are weighted the same for each data type and avoids complications of having various units.



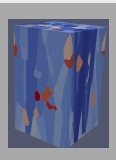
Recap: calculate the *covariance matrix* (measure of the variance between variables)

for the autoscaled data $X'_{ki} = \dfrac{X_{ki} - \overline{X_i}}{\sigma_i}$ with $\overline{X'_i} = 0$ and $\sigma_{X'_i} = 1$

$$C_{ij} = \frac{1}{N-1}\sum_{k=1}^{N} X'_{ki}X'_{kj} = \frac{1}{N-1}\sum_{k=1}^{N} \frac{\left(X_{ki} - \overline{X_i}\right)\left(X_{kj} - \overline{X_j}\right)}{\sigma_i \sigma_j}$$

$$C_{ii} = \frac{1}{N-1}\sum_{k=1}^{N} X'^{2}_{ki} = \frac{1}{N-1}\sum_{k=1}^{N} \frac{\left(X_{ki} - \overline{X_i}\right)^2}{\sigma_i^2} = \frac{S_i}{S_i} = 1$$

C is a measure of the correlation between the variables.

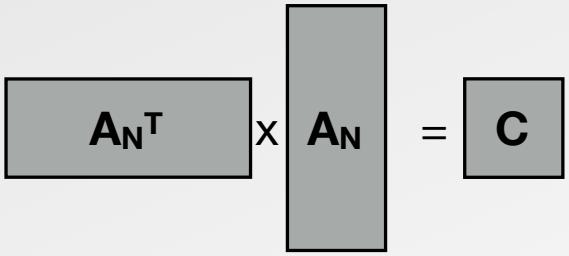


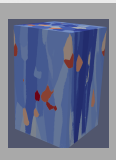
Recap: while we could sum the data as shown on the previous slide, it is actually easier to use what we've learned about matrices:

Since $\mathbf{A}_N = \begin{bmatrix} \mathbf{X}_1' & \mathbf{X}_2' & \mathbf{X}_3' & \mathbf{X}_4' & \mathbf{X}_5' & \mathbf{X}_6' \end{bmatrix}$, we can calculate $\mathbf{C}$ as

$$\mathbf{C} = \frac{\mathbf{A}_N^T \mathbf{A}_N}{N-1}$$

$\mathbf{A}_N$ is has dimensions of $p \times N$ and $\mathbf{A}_N^T$ has dimensions of N X p. The product has dimensions of p X p.





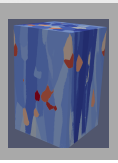
Recap: we now find the eigenvectors and eigenvalues of the covariance matrix **C**: the *principal components*, which create an orthogonal basis.

From the covariance matrix given above, we find:

$$\lambda = \begin{pmatrix} 3.06229 \\ 1.21896 \\ 0.870424 \\ 0.586331 \\ 0.220362 \\ 0.0416367 \end{pmatrix} \qquad \mathbf{P} = \begin{pmatrix} -0.393417 & -0.391968 & -0.0774367 & 0.706188 & -0.431937 & -0.0178061 \\ -0.38676 & -0.0474299 & 0.654928 & -0.451536 & -0.457014 & -0.0805578 \\ -0.170358 & -0.644791 & -0.527402 & -0.525786 & -0.0246437 & -0.00331295 \\ 0.506673 & -0.351709 & 0.203926 & 0.0689972 & -0.0348986 & -0.756319 \\ -0.411283 & 0.508746 & -0.407694 & -0.0727362 & -0.107181 & -0.623724 \\ -0.49066 & -0.214087 & 0.281282 & 0.104463 & 0.768931 & -0.179255 \end{pmatrix}$$

The eigenvectors are orthonormal: $\mathbf{P}^T\mathbf{P} = \mathbf{I}$

Note: the sum of the eigenvalues = the number of data types: $\sum_{i=1}^{p} \lambda_k = p$



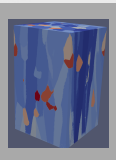
Cumulative fractional variance measures how much variance is included as new PCs are included in the description of the data:

$$\Lambda_k = \sum_{i=1}^{k} \lambda_i^f = (0.510381,\ 0.713541,\ 0.858612,\ 0.956334,\ 0.993061\ 1.)$$

71 % of the variance is contained in the first 2 principal components and 86 % in the first three principal components.

We can develop *reduced dimensional* representations of the data using only the first few principal components (PC).

The aim of PCA is find linear combinations of the variables that explain a large fraction of the data, preferably only two [principal components]!



Cumulative fractional variance

The ***scores,*** **T**, are found by *projecting* the original autoscaled data, **A**, onto the eigenvectors, **P**, since they form an orthonormal system.

$$\mathbf{T} = \mathbf{A}_N\mathbf{P}$$

Note: $\mathbf{A}_N$ is (44x6) dimensional matrix, while $\mathbf{P}$ is a (6x6) matrix: the product is the (44x6) matrix of the *principal components*.

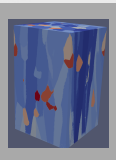
For example, the first two PCs (i.e., the 1st two columns in **P**) provides these scores:

at.no. MeltT # val. e- Δ radii elecneg latcon

$$PC1_i = -0.393417X'_{1,i} - 0.38676X'_{2,i} - 0.170358X'_{3,i} + 0.506673X'_{4,i} - 0.411283X'_{5,i} - 0.49066X'_{6,i}$$

$$PC2_i = -0.391968X'_{1,i} - 0.0474299X'_{2,i} - 0.644791X'_{3,i} - 0.351709X'_{4,i} + 0.508746X'_{5,i} - 0.214087X'_{6,i}$$

Coordinates of data point *i on the scores plot* are $(PC1_i, PC2_i)$



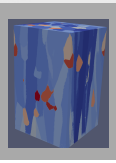
The scores depend on the principal components.

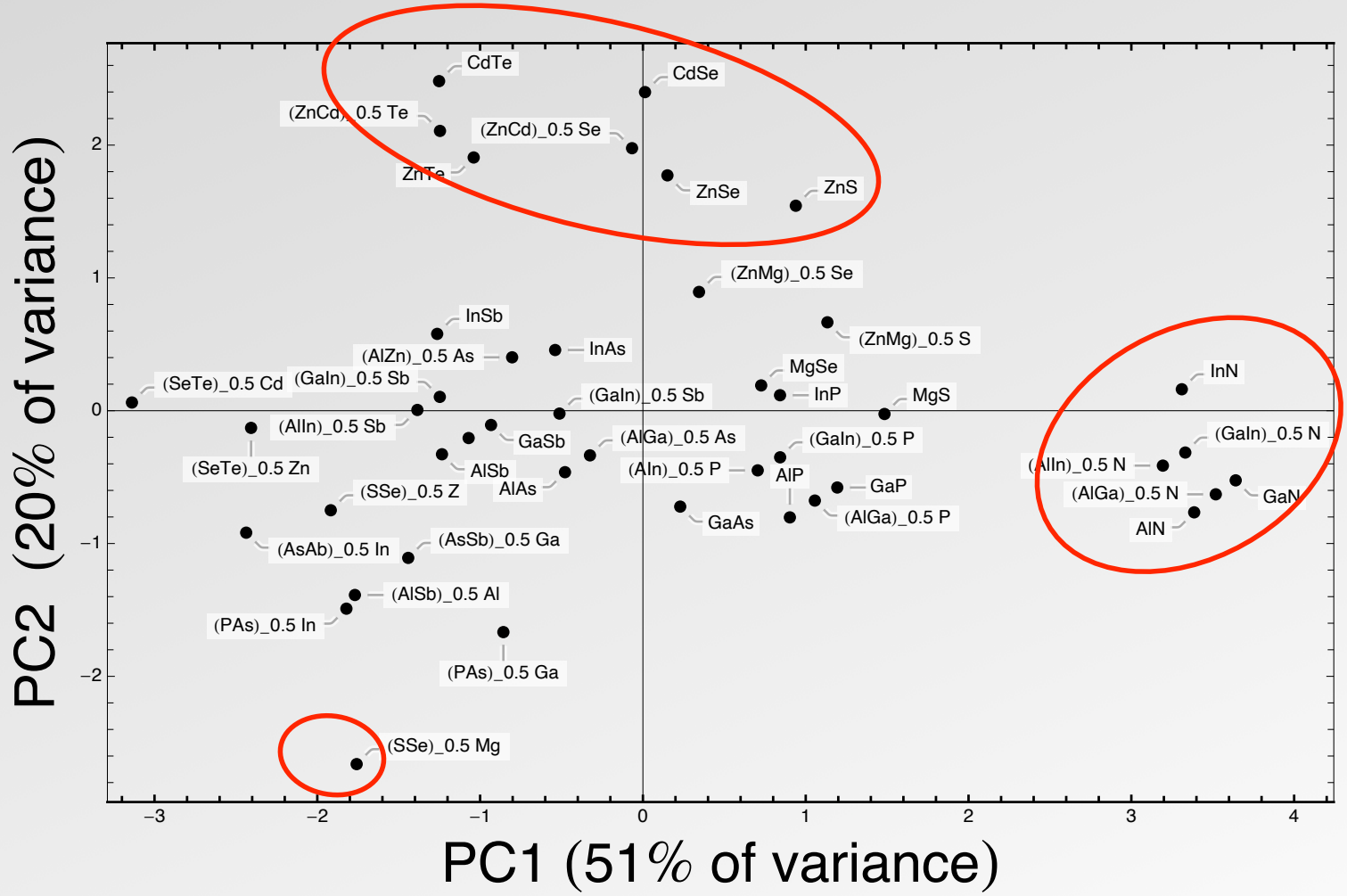
The PCs form a basis into which we can plot the scores.

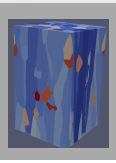
For example, the first two PCs yields 71 % of the overall variance.

We can understand the data by looking at a (2D) plot of the scores for a pair of the PCs.

PC1	PC2	PC3	PC4	PC5	PC6
3.38605	0.765178	0.0444564	−0.923304	−0.802435	0.0866611
0.902668	0.803237	0.632804	−0.821175	0.60889	0.0112951
−0.478575	0.462851	1.85174	−1.12572	−0.520492	−0.0269538
−1.23421	0.327974	1.42677	−0.302138	−0.279255	0.230554
3.641	0.524342	−1.10432	0.422339	−0.392991	0.120627
1.19379	0.578183	−0.536705	0.516768	0.961652	0.0584743
0.229129	0.722097	0.539643	0.933548	0.215606	0.0837623
−0.932761	0.107428	0.251338	1.0336	0.0573127	0.281508
3.31008	−0.161519	−0.687566	0.966179	−0.615917	−0.372175
0.842192	−0.116696	−0.108104	1.06501	0.771116	−0.441878
−0.53905	−0.457082	1.11083	0.760466	−0.358265	−0.480127
−1.26369	−0.578433	0.66809	1.57744	−0.165613	−0.211293
0.939099	−1.54346	−0.570113	−1.15694	0.389241	0.0412944
0.150621	−1.77215	−0.498859	−0.679241	0.0995653	0.0928495
−1.03943	−1.90686	−0.262708	−0.352891	−0.192736	0.291933
0.0129525	−2.39906	−0.486929	0.123092	0.14036	−0.352693
−1.25198	−2.48178	−0.280917	0.439244	−0.146783	−0.0418305
1.48393	0.0246309	1.31671	−0.748265	0.78792	−0.117394
0.726455	−0.190532	1.3702	−0.277163	0.44966	−0.0545128
3.51876	0.629499	−0.515858	−0.248215	−0.588917	0.139315
3.19368	0.413367	−0.387786	−0.00059617	−0.703294	0.0844462
3.33143	0.314974	−0.985133	0.666658	−0.531661	0.0714207
1.13312	−0.665683	0.303634	−0.971007	0.555431	0.00796419
−1.75831	2.66097	−0.6306	−0.957956	0.361215	−0.0457656
−1.91722	0.749513	−2.28023	−1.30843	−0.0182307	−0.0447058
0.344642	−0.894373	0.374888	−0.493304	0.265756	0.0595193
−0.0667437	−1.9767	−0.558973	−0.298736	0.126669	0.0596045
−2.40678	0.129375	−1.83061	−0.826709	−0.273371	−0.304606
−3.13857	−0.0618932	−2.083	−0.107022	−0.205751	0.00795682
−1.24646	−2.10689	−0.328746	0.0254132	−0.181827	0.262463
⋮	⋮	⋮	⋮	⋮	⋮







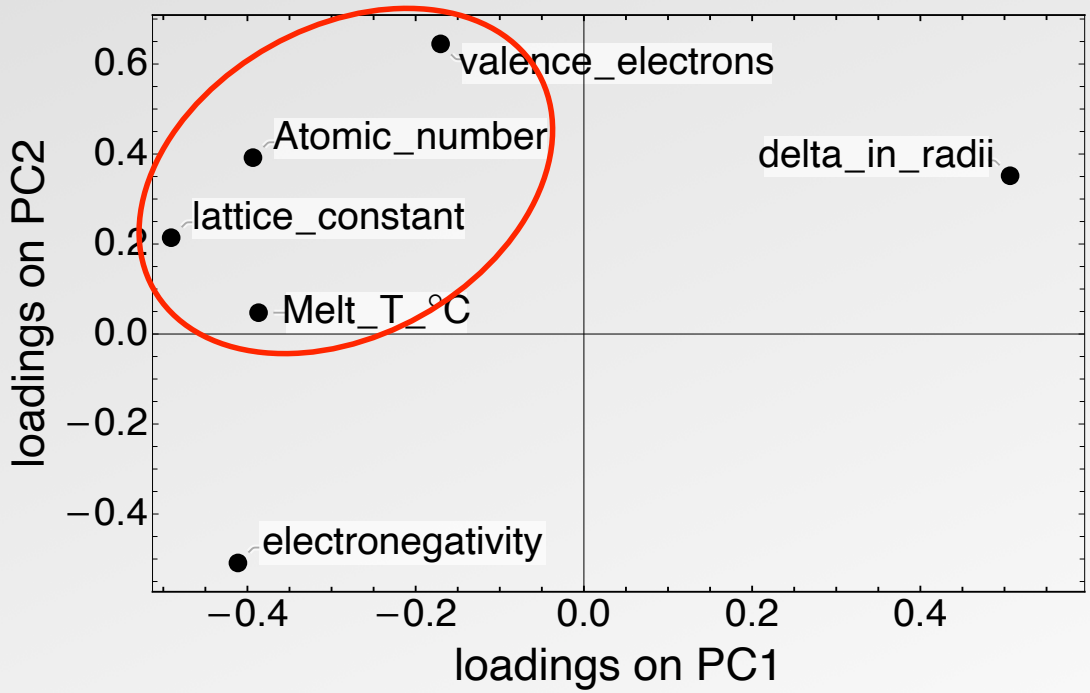
Score plot (only 71 % of variance)

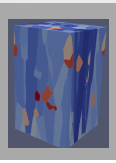
The loadings are just the location of six eigenvectors on PC1 and PC2.

$$L_{at.no.} = \{P_{11}, P_{12}\}$$

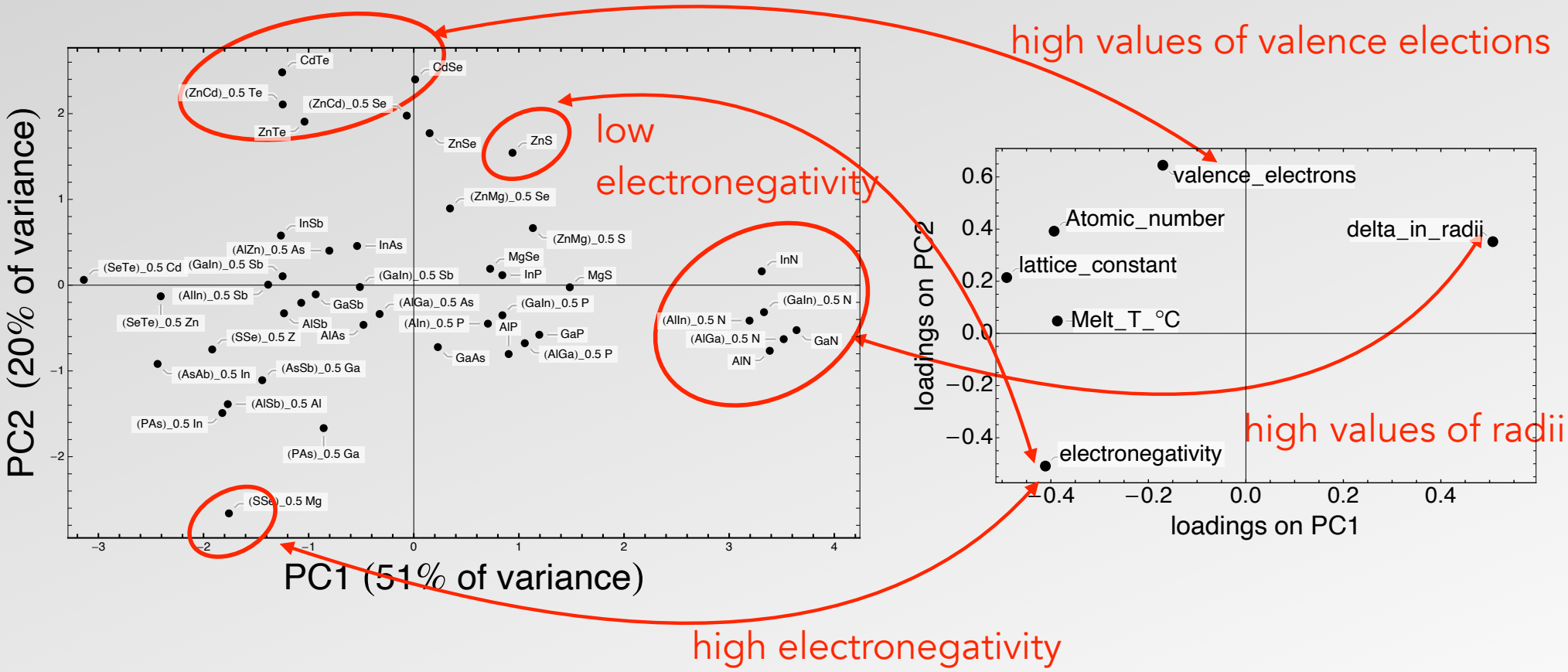
$$L_{meltingpt} = \{P_{21}, P_{22}\}$$

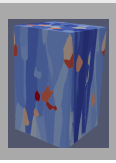
etc.



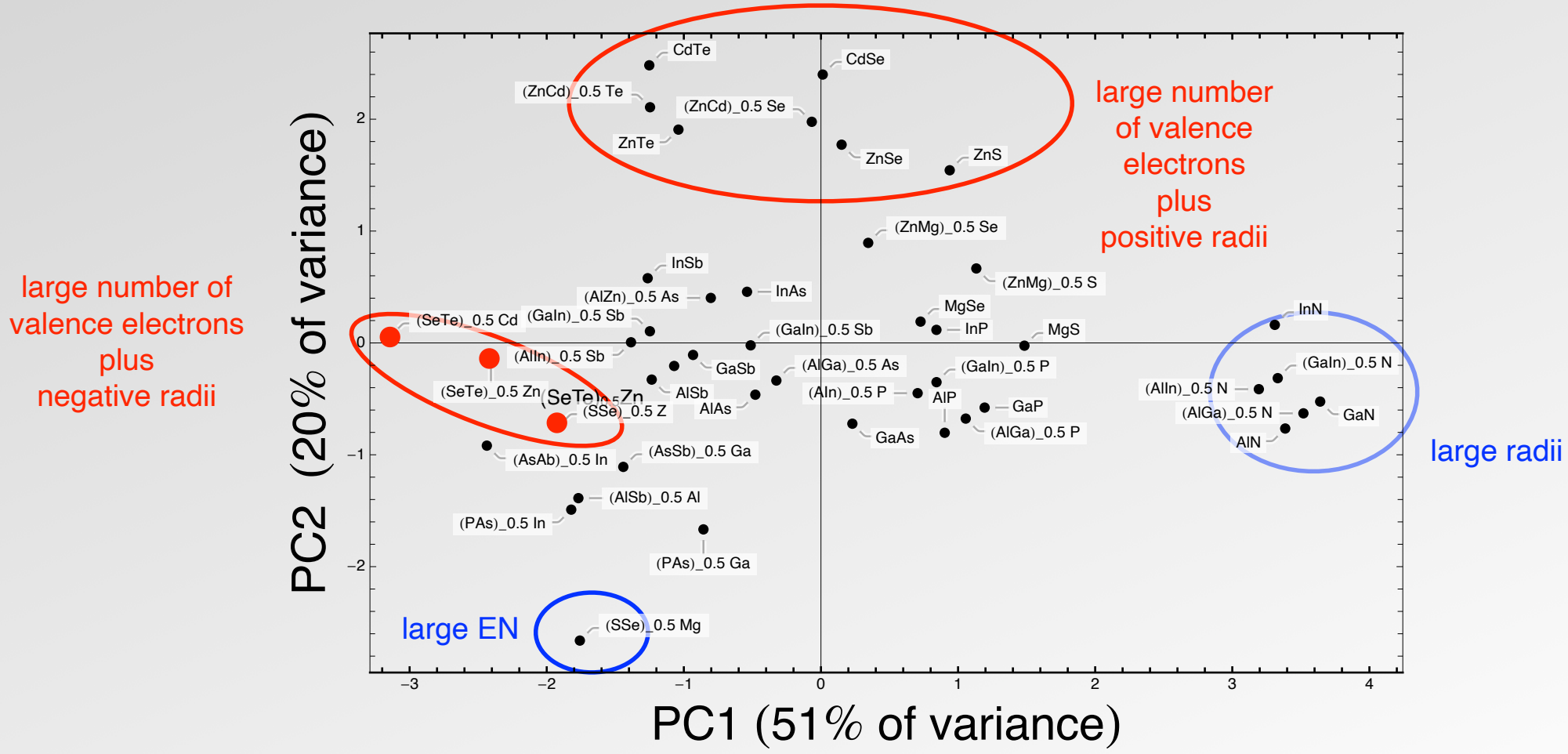


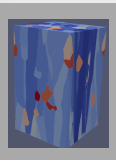
Loadings





Scores and loadings (only 71% of variance)





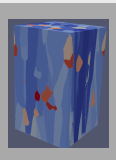
More on scores (only 71 % of variance)

Paraphrasing from Jobson:
A biplot is used to provide a 2D representation for a data matrix, **A** or **X**. We limit it to 2D for convenience. We assume that an SVD is available (but check the screeplot) and that two PCs are enough.

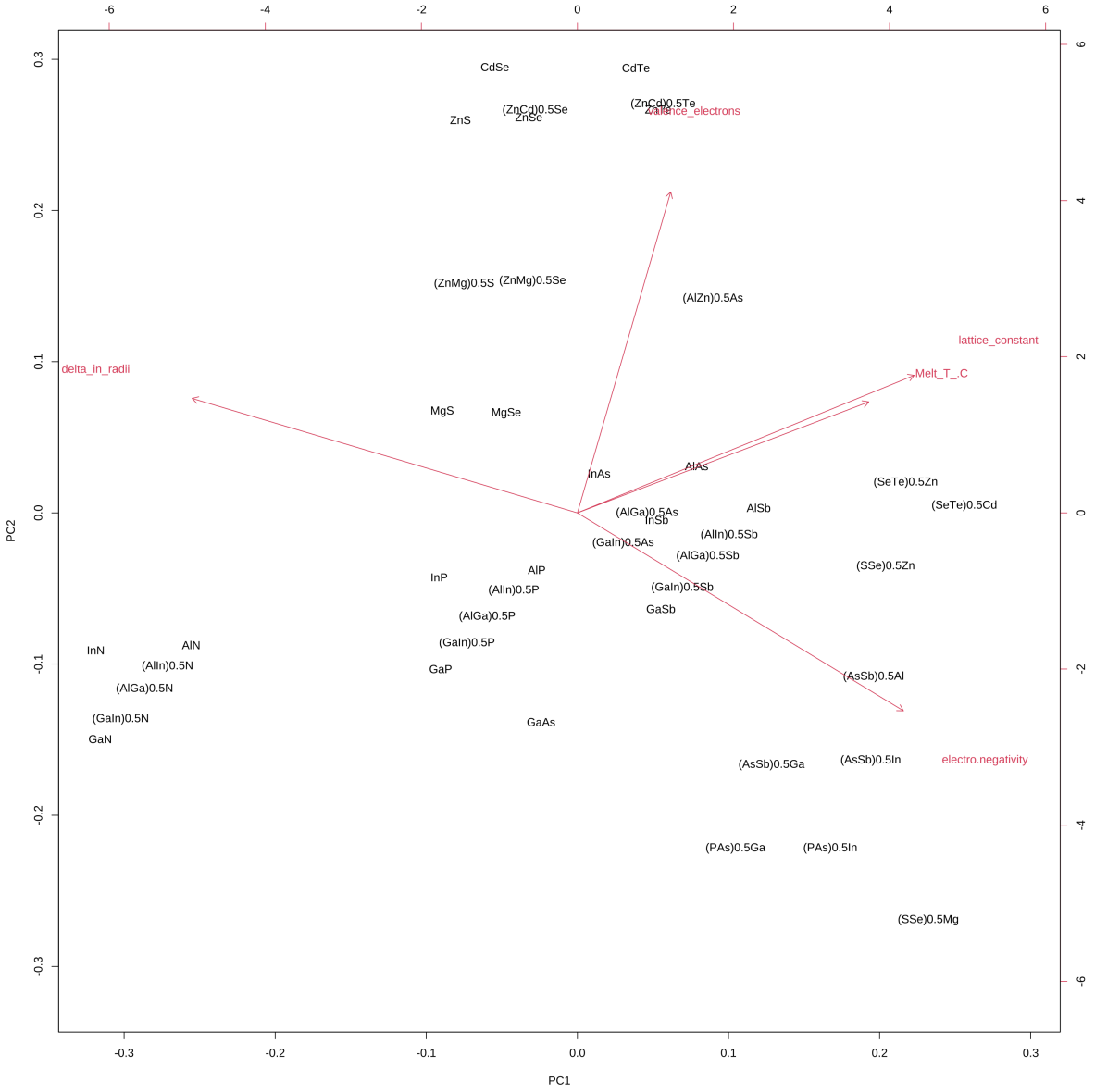
Start with the first 2 PCs, which provide the (orthogonal) axes. Plot the PC scores on the graph as points. Add the loadings of the two PCs as rays (arrows), which are the values in each eigenvector (for each variable).

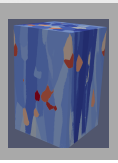
To obtain plots with other PCs, add the argument “choices”, as in:

```
biplot(PCkr, choices = 3:4, scale = 1)
```

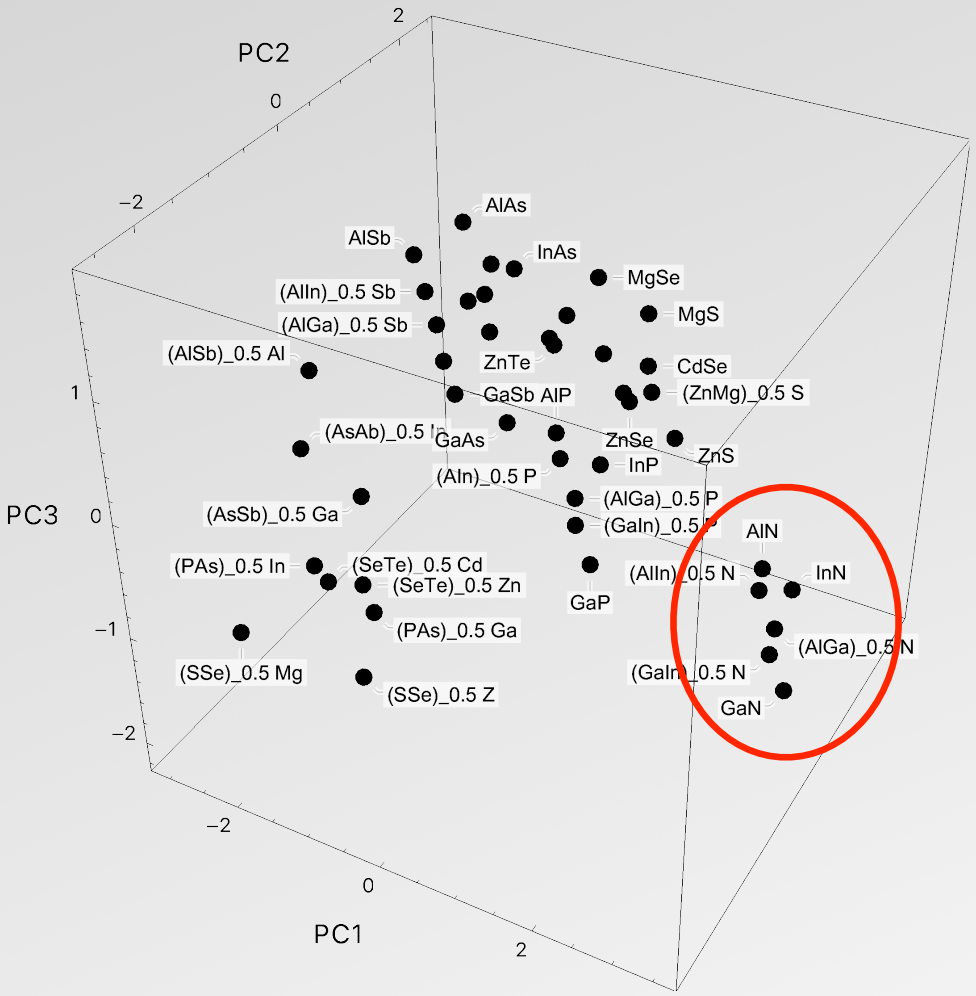


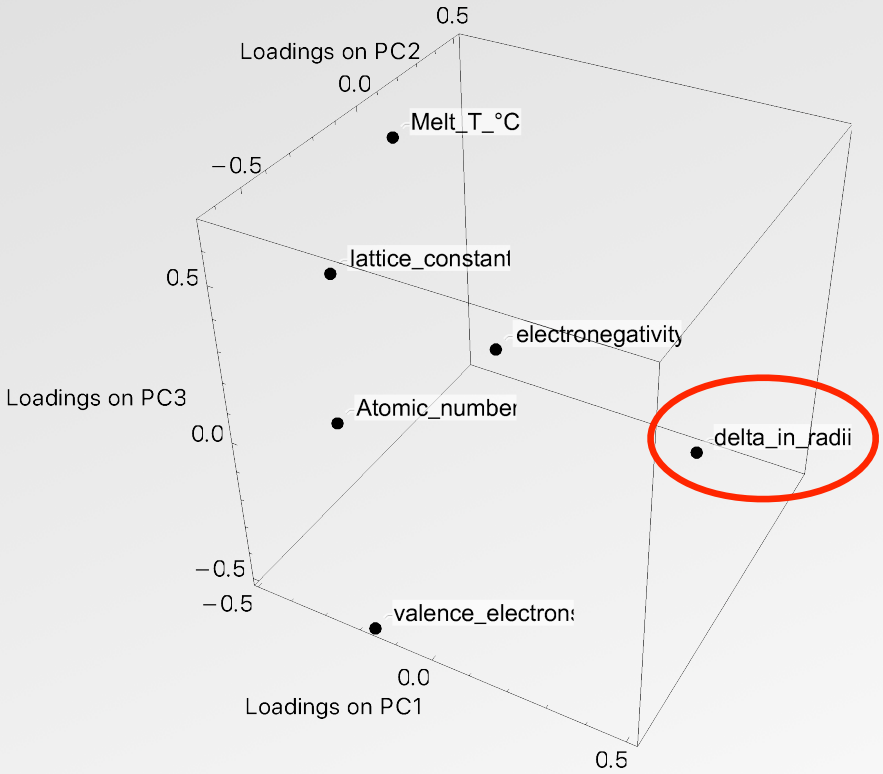
Example of a biplot

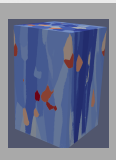




Biplot example







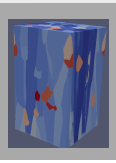
3 PCs: 86% of the variance

Applications of principal component analysis: (Rajan, DOI:10.1002/sam.10031)

- identify the strongest patterns in the data
- capture most of the variability of the data with a small fraction of the total set of dimensions
- eliminate most of the noise in the data, making it better for data mining and other data analysis algorithms

PCA can be used to determine the relationships among a group of variables in situations where it is not appropriate to make *a priori* grouping decisions, i.e., the data is not split into groups of dependent and independent variables.

It allows us to reduce the number of variables needed to represent the data by describing the data in terms of linear combinations of all the types of data.

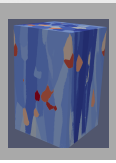


"**Average Criterion**"

We know that the total variation is given by the sum of the eigenvalues (thinking of a solution based on the covariance matrix). The average value $\bar{\lambda}$ is easily calculated. A simple criterion is to keep all the principal components that exceed the average, $\lambda_i > \bar{\lambda}$.

In the case of PCA on the correlation matrix, the sum is the number of variables, $\sum_{j=1}^{p} \lambda_i = p$, so $\bar{\lambda} = 1$. For this (common) situation, the "Average Criterion" becomes the *eigenvalue-one-criterion*. The latter is also used in *factor analysis* (not discussed in this course). Note that solving the problem with unscaled versus scaled values will generally result in a different number of components being retained.

In the case of the semiconductor analysis, we would keep only the 1st two PCs for a total of 71 % explained of the variance in the data, which is quite good.

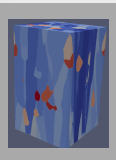


Criterion for how many PCs to keep based on $\bar{\lambda}$

In some cases, the last few PCs are associated with very small fractions of the variance. It may be possible to eliminate a variable, which decreases the total PC count. If a variable is eliminated without affecting the variance explained by the (already decided upon) retained PCs, then it can be sacrificed.

Note however that this is a complicated topic and somewhat controversial. PCA is intended to help you understand your entire dataset and its structure. It is *not* intended as a technique for regression analysis.

The next 2 slides have excerpts from Jobson …



LPOP (0.87). Thus 96% of the variation in PMEAN is explained by the first four components, whereas for NONPOOR this percentage is 77.

How Many Principal Components?

Recall that one of the objectives in principal components analysis was to replace the set of p original variables with a small subset of r principal components. The assumption was that because of the covariance relationships among the variables a small value of r would usually be sufficient to retain most of the variation. The sum of squared deviations between the original matrix $\mathbf{X}$ and the estimated values based on the first r components is given by $tr\mathbf{X}'\mathbf{X} - \sum_{j=1}^{r}\lambda_j = \sum_{j=r+1}^{p}\lambda_j$ and hence the proportion of the total sum of squares accounted for by the first r components is given by $\sum_{j=1}^{r}\lambda_j / \sum_{j=1}^{p}\lambda_j$. Some cut-off proportion, therefore, can be used to determine the number of components to retain.

Average Criterion

Since the total variation is given by $\sum_{j=1}^{p}\lambda_j$, where λ_j is the variance of Z_j, a possible rule of thumb is to retain those components whose variance exceeds the average $\bar{\lambda} = \sum_{j=1}^{p}\lambda_j/p$. In other words, retain Z_j if $\lambda_j > \bar{\lambda}$. For correlation matrices, $\sum_{j=1}^{p}\lambda_j = p$ and hence $\bar{\lambda} = 1$. This criterion becomes the *eigenvalue-one-criterion*, which is commonly used in factor analysis and will be discussed in Section 9.2.

Example

For the SSCP matrix $\mathbf{X}'\mathbf{X}$ of the example, the average eigenvalue criterion requires that eigenvalues above 833,794 be retained, and hence only the first component representing 91% of the variance would be retained. For the covariance matrix the criterion suggests that eigenvalues above 4084 correspond to factors that should be retained. Thus the first three components representing 95% of the variance should be retained. For the correlation matrix the eigenvalue-one-criterion suggests the retention of four components. The first four components account for 86% of the variation for the correlation matrix.

Geometric Mean Criterion

An alternative criterion based on the eigenvalues is the generalized variance. Since $|\mathbf{X}'\mathbf{X}| = \Pi_{j=1}^{p}\lambda_j$, we have that $|\mathbf{X}'\mathbf{X}|^{1/p} = [\Pi_{j=1}^{p}\lambda_j]^{1/p} =$ the geometric mean, $\bar{\lambda}_m$, of the eigenvalues. The *average generalized variance* is given by the geometric mean of the eigenvalues, $\bar{\lambda}_m$ and hence the criterion retain Z_j if $\lambda_j > \bar{\lambda}_m$. Recall that the geometric mean is useful for averaging a set of numbers containing a few extremes.

Example

For SSCP matrices and covariance matrices, the geometric mean provides a more useful criterion than the one based on the sum. For the SSCP matrix, the geometric mean of the eigenvalues is 23,247 and hence the first five factors should be retained. These factors jointly account for 99.7% of the variation. For the covariance matrix, the geometric mean of the eigenvalues is 347 and hence the first five components should be retained. These five factors jointly account for 98.7% of the variation. All components corresponding to the later eigenvalues are ignored.

A Test for Equality of Eigenvalues in Covariance Matrices

Since the eigenvalues decline in a geometric fashion, it can often be argued that the last $(p-r)$ eigenvalues are primarily due to "noise". In such cases it is of interest to test the null hypothesis that the last $(p-r)$ eigenvalues are equal. Under the assumption that the N observations have been sampled from a multivariate normal distribution, the test statistic is given by

$$[n - (2p+11)/6]\left[(p-r)\ln\bar{\lambda}_{p-r} - \sum_{j=r+1}^{p}\ln\lambda_j\right]$$

where λ_j, $j = 1, 2, \ldots, p$ are the eigenvalues of the covariance matrix and where $\bar{\lambda}_{p-r} = \sum_{j=r+1}^{p}\lambda_j/(p-r)$. If the null hypothesis is true, this statistic has a χ^2 distribution with $\frac{1}{2}(p-r+2)(p-r-1)$ degrees of freedom. A special case of this hypothesis was discussed as a test of sphericity in Chapter 7. Later in this chapter, in factor analysis, the scree test will also be concerned with the equality of the latter eigenvalues of the correlation matrix.

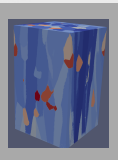
A Cross Validation Approach

For large data sets the data can be divided into g mutually exclusive subsets. A principal components solution is determined using all data excluding one of the groups. The principal components solution is used to predict the observations in the omitted group. The goodness of fit is evaluated using

$$T_j(r) = tr[\mathbf{X}(j) - \hat{\mathbf{X}}(j)]'[\mathbf{X}(j) - \hat{\mathbf{X}}(j)],$$

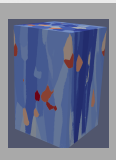
where j denotes the groups omitted. For each value r of the number of components each group is omitted once and is predicted by the remaining group. The total measure of error $T(r) = \sum_{j=1}^{g}T_j(r)$ over the g groups is determined. As the number of principal components r increases, the total error $T(r)$ decreases. When the relative change in total error as measured by $[T(r) - T(r-1)]/T(r-1)$ is considered small it is not necessary to add additional principal components.

Other approaches to cross validation based on the likelihood function will be introduced in the section on factor analysis.



From Jobson

Next week we will examine Canonical Correlation Analysis (CCA)



Questions?

We start with a data set based on 2337 grains from an unnamed superalloy, for which we have measured (or constructed) data consisting of 11 variables for each grain:

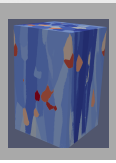
$$\left\{ b/a,\ c/a,\ a,\ b,\ c,\ x_c,\ y_c,\ z_c,\ D_{eq.},\ N_{Neighbors},\ \Omega_3 \right\}$$

- the position of each grain is $\{x_c,\ y_c,\ z_c\}$
- the size of each grain is described by $\left\{ a,\ b,\ c,\ D_{eq.}, N_{Neighbors} \right\}$
- the shape of each grain is captured by $\{b/a,\ c/a,\ \Omega_3\}$

How can we best understand this data?

- Can we use principal component analysis to reduce the dimensionality of the data, without (we hope) losing too much information?

Data from M. Groeber, AFRL from a DREAM3D data file.

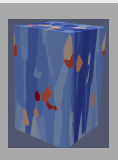


Example 2: grain data (from the USAF Materials Lab)

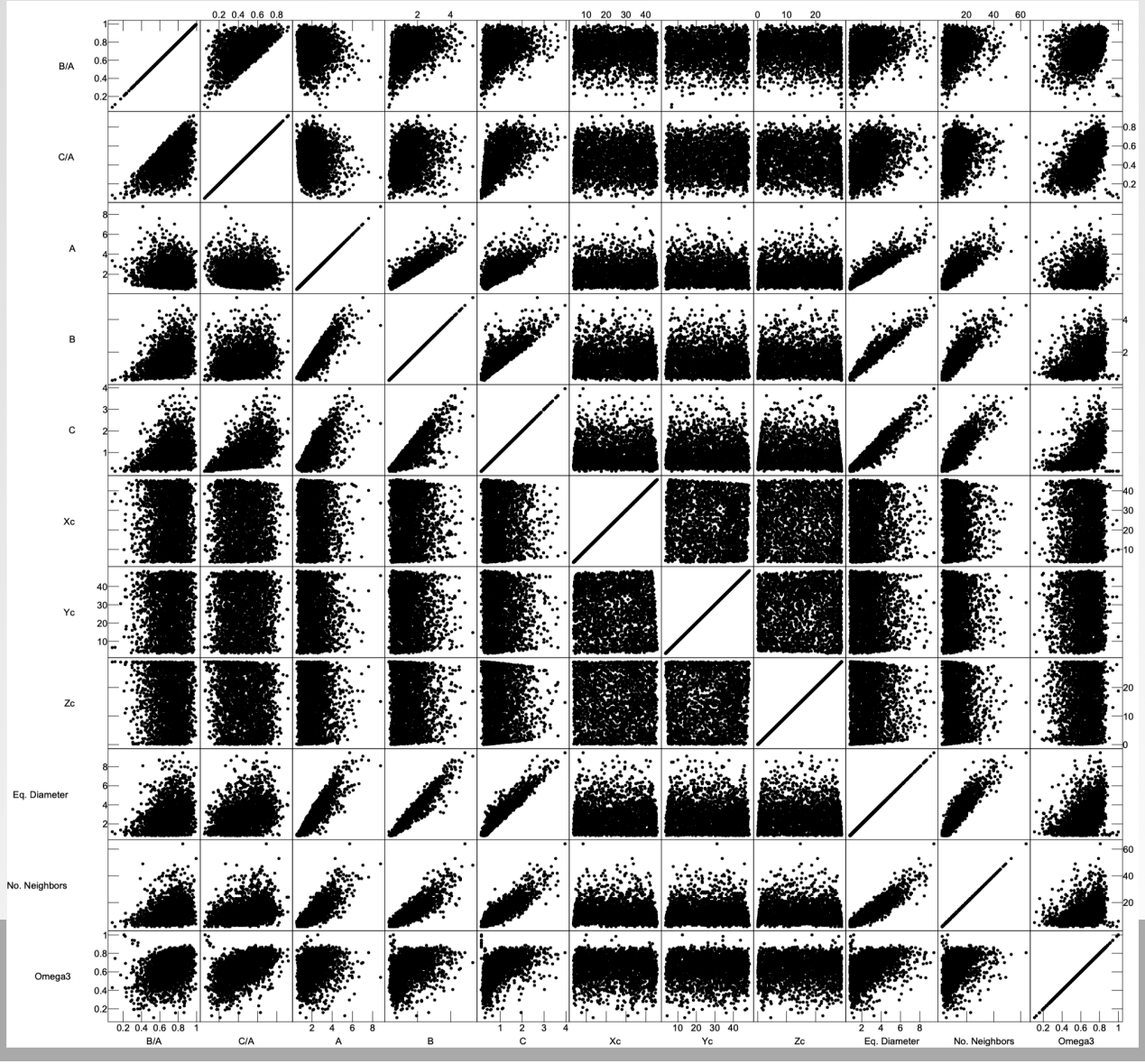
The scatterplot matrix shows correlations in data as expected: as a increases, so do $\{b, c, D_{eq.}, N_{neighbors}\}$

The positions $\{x_c, y_c, z_c\}$ fill space.

It is difficult from these plots to see any surprising correlations in the data.



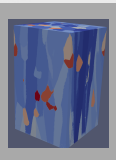
Example 2:



Steps:

- autoscale the data
- calculate the covariance matrix $\mathbf{C}$

	b/a	c/a	a	b	c	x	y	z	D	N	Ω_3
b/a	1.	0.615334	−0.0297718	0.374626	0.349329	−0.0132041	0.0120379	−0.0481651	0.27888	0.260689	0.356439
c/a	0.615334	1.	−0.009316	0.244918	0.61172	−0.028992	0.0349535	−0.0754699	0.364311	0.34937	0.547882
a	−0.0297718	−0.009316	1.	0.895273	0.735179	0.0112308	−0.0789474	0.0182463	0.903379	0.802318	0.232272
b	0.374626	0.244918	0.895273	1.	0.842408	0.00198956	−0.0653764	−0.00400199	0.96082	0.861533	0.371346
c	0.349329	0.61172	0.735179	0.842408	1.	−0.0123373	−0.032131	−0.0326387	0.941835	0.865294	0.509272
x	−0.0132041	−0.028992	0.0112308	0.00198956	−0.0123373	1.	−0.074451	−0.0345575	−0.00269399	0.00411898	−0.0185865
y	0.0120379	0.0349535	−0.0789474	−0.0653764	−0.032131	−0.074451	1.	0.0498874	−0.0576587	0.0103652	−0.0231603
z	−0.0481651	−0.0754699	0.0182463	−0.00400199	−0.0326387	−0.0345575	0.0498874	1.	−0.0124335	−0.00868138	0.0144447
D	0.27888	0.364311	0.903379	0.96082	0.941835	−0.00269399	−0.0576587	−0.0124335	1.	0.903106	0.441562
N	0.260689	0.34937	0.802318	0.861533	0.865294	0.00411898	0.0103652	−0.00868138	0.903106	1.	0.38124
Ω_3	0.356439	0.547882	0.232272	0.371346	0.509272	−0.0185865	−0.0231603	0.0144447	0.441562	0.38124	1.



Example 2: Create the correlation function

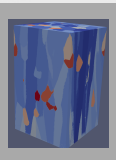
Choose:

- 3 measures of size $\left\{c,\ N_{Neighbors},\ D_{eq.}\right\}$
- 3 measures of shape $\left\{b/a,\ c/a,\ \Omega_3\right\}$

	b/a	c/a	c	D	N	Ω_3
b/a	1.	0.615334	0.349329	0.27888	0.260689	0.356439
c/a	0.615334	1.	0.61172	0.364311	0.34937	0.547882
c	0.349329	0.61172	1.	0.941835	0.865294	0.509272
D	0.27888	0.364311	0.941835	1.	0.903106	0.441562
N	0.260689	0.34937	0.865294	0.903106	1.	0.38124
Ω_3	0.356439	0.547882	0.509272	0.441562	0.38124	1.

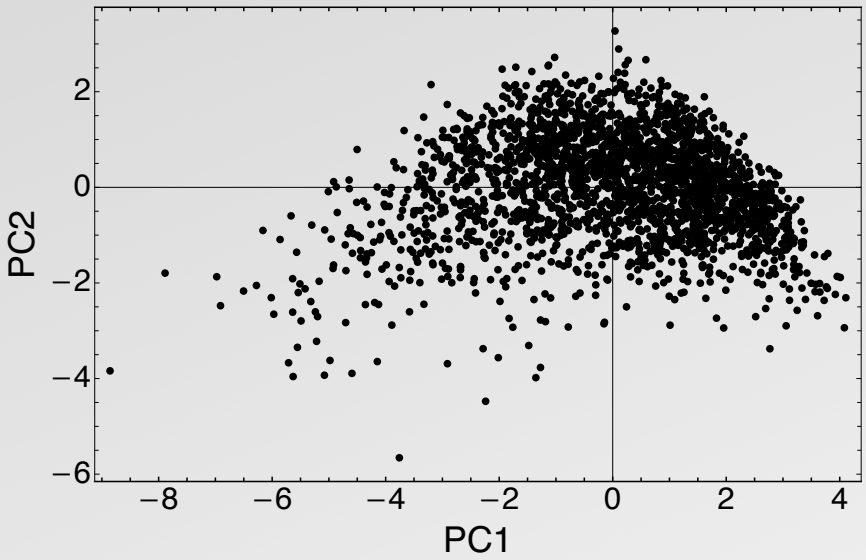
The first 2 PCs capture 82% of the variance.

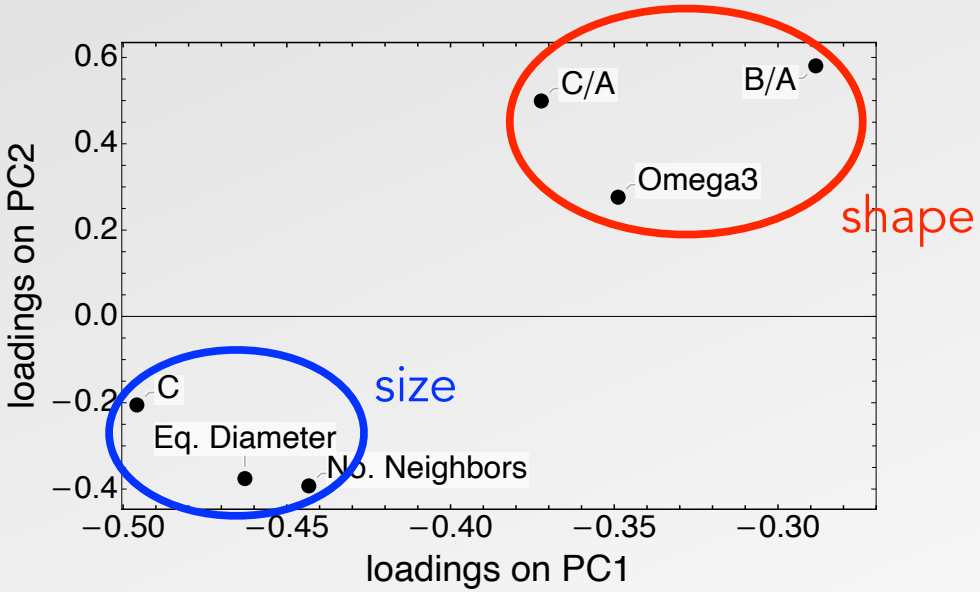
The first 3 PCs capture 92% of the variance.



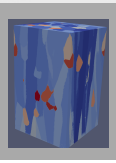
Example 2: Reduced set of variables: size and shape

82% of the variance



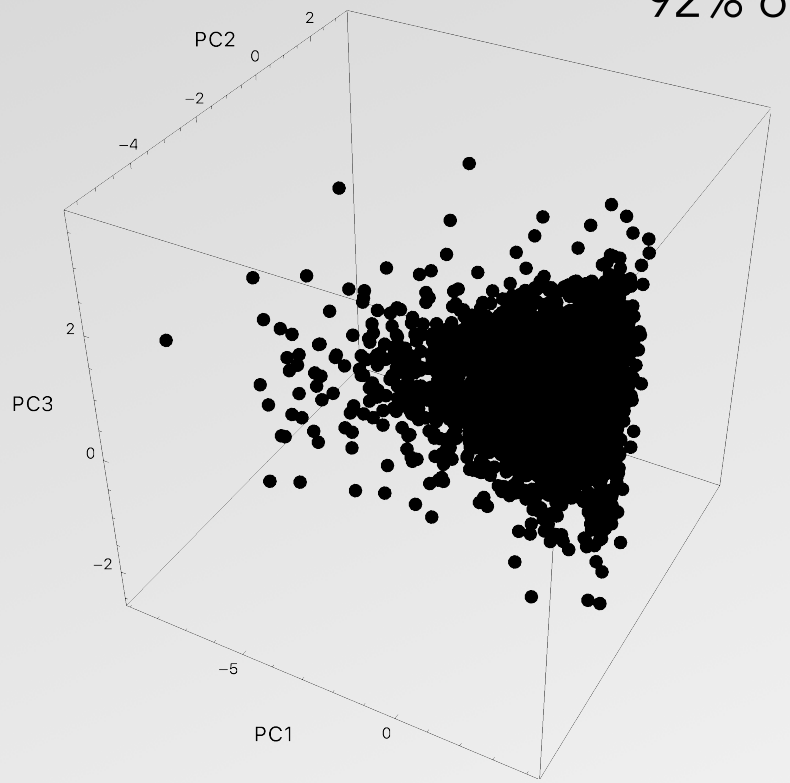


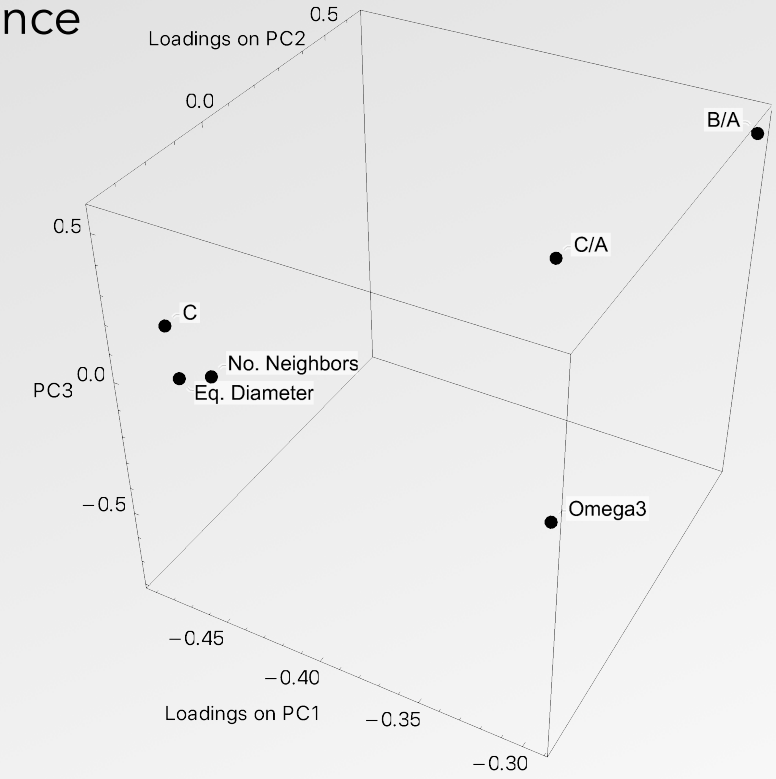
I do not see any separation of the variables in the scores.



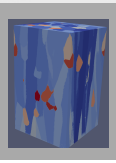
Example 2: Reduced set of variables: size and shape

92% of the variance





I do not see any separation of the variables in the scores.



Example 2: Reduced set of variables: size and shape

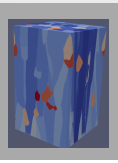
Steps:

- autoscale the data
- calculate the covariance matrix $\mathbf{C}$

	b/a	c/a	a	b	c	x	y	z	D	N	Ω_3
b/a	1.	0.615334	−0.0297718	0.374626	0.349329	−0.0132041	0.0120379	−0.0481651	0.27888	0.260689	0.356439
c/a	0.615334	1.	−0.009316	0.244918	0.61172	−0.028992	0.0349535	−0.0754699	0.364311	0.34937	0.547882
a	−0.0297718	−0.009316	1.	0.895273	0.735179	0.0112308	−0.0789474	0.0182463	0.903379	0.802318	0.232272
b	0.374626	0.244918	0.895273	1.	0.842408	0.00198956	−0.0653764	−0.00400199	0.96082	0.861533	0.371346
c	0.349329	0.61172	0.735179	0.842408	1.	−0.0123373	−0.032131	−0.0326387	0.941835	0.865294	0.509272
x	−0.0132041	−0.028992	0.0112308	0.00198956	−0.0123373	1.	−0.074451	−0.0345575	−0.00269399	0.00411898	−0.0185865
y	0.0120379	0.0349535	−0.0789474	−0.0653764	−0.032131	−0.074451	1.	0.0498874	−0.0576587	0.0103652	−0.0231603
z	−0.0481651	−0.0754699	0.0182463	−0.00400199	−0.0326387	−0.0345575	0.0498874	1.	−0.0124335	−0.00868138	0.0144447
D	0.27888	0.364311	0.903379	0.96082	0.941835	−0.00269399	−0.0576587	−0.0124335	1.	0.903106	0.441562
N	0.260689	0.34937	0.802318	0.861533	0.865294	0.00411898	0.0103652	−0.00868138	0.903106	1.	0.38124
Ω_3	0.356439	0.547882	0.232272	0.371346	0.509272	−0.0185865	−0.0231603	0.0144447	0.441562	0.38124	1.

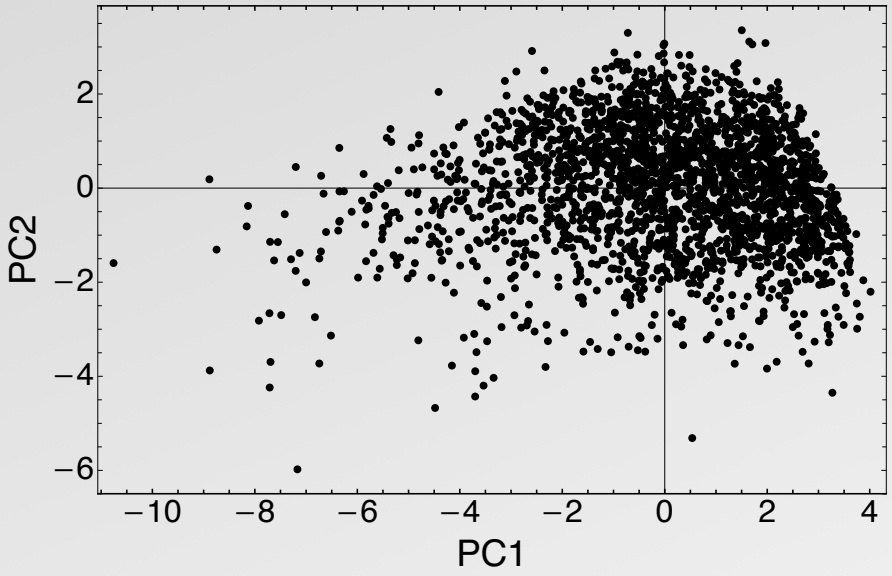
The first 2 PCs capture 61% of the variance.

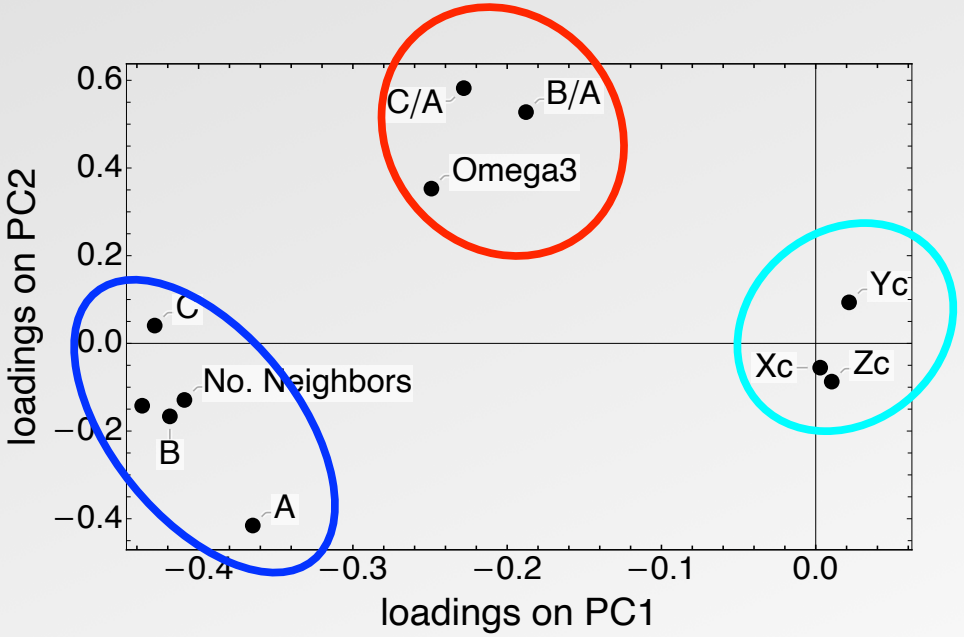
The first 3 PCs capture 71% of the variance.

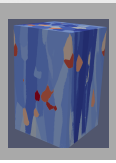


All 11 functions: the covariance matrix

61% of the variance

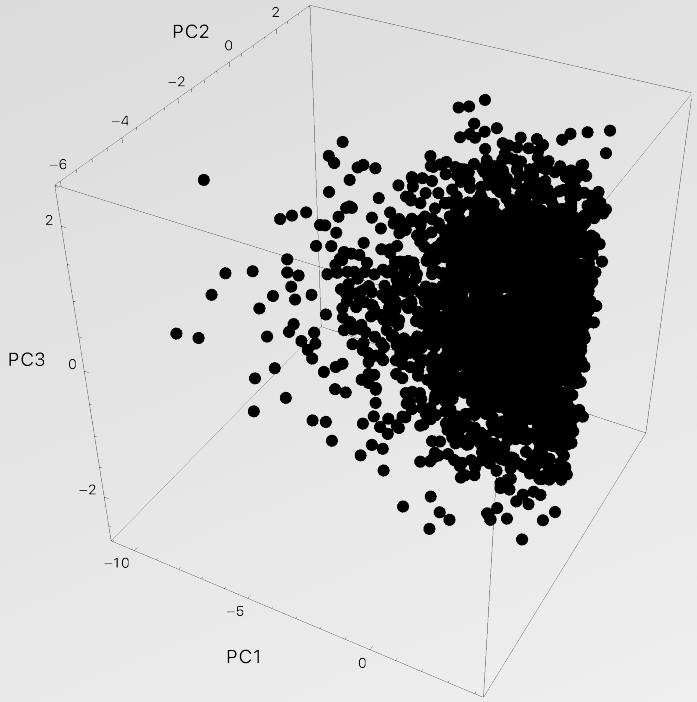


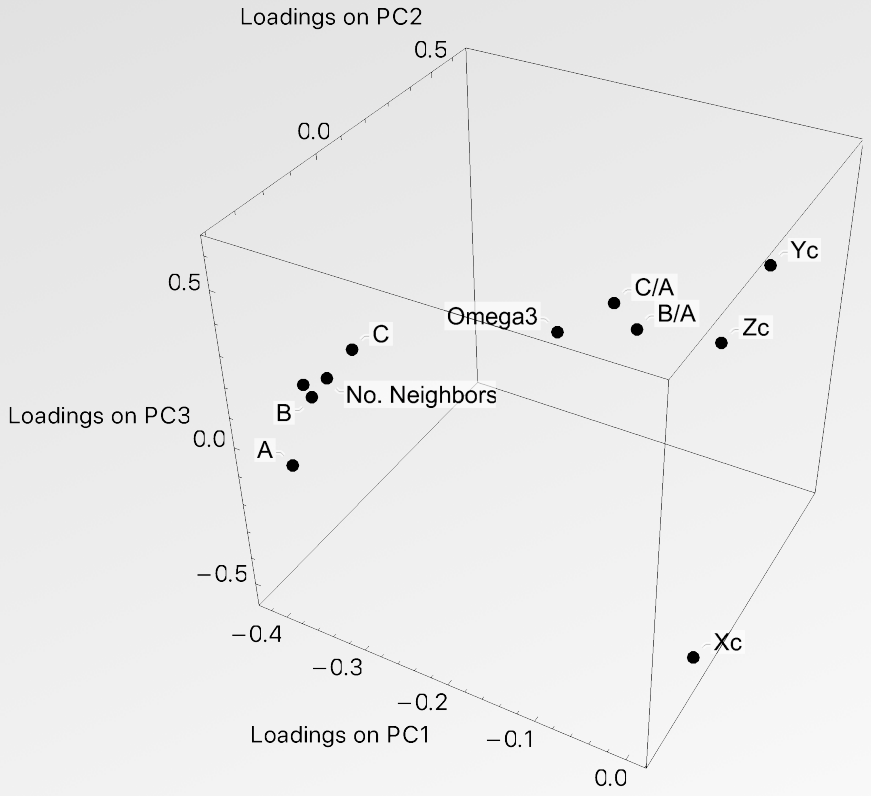


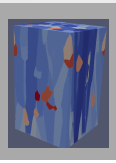


Example 2: the full set of 11 variables

71% of the variance



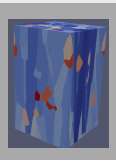




Example 2: the full set of 11 variables

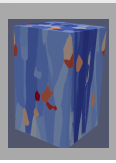
"However, care must be taken to judiciously apply PCA. Authors will readily acknowledge PCs are not necessarily simple to interpret physically especially with image data (Belianinov et al., 2015a). PCA is not guaranteed to separate clusters of data from one another (Ringnér, 2008), and overzealous projection onto PCs can actually make classes of data inseparable that were previously separable before PCA."

— Wagner and Rondinelli, Frontiers in Materials, https://www.frontiersin.org/articles/10.3389/fmats.2016.00028/full



"However, care must be taken to judiciously apply PCA. Authors will readily acknowledge PCs are not necessarily simple to interpret physically especially with image data (Belianinov et al., 2015a). **PCA is not guaranteed to separate clusters of data from one another (Ringnér, 2008),** and overzealous projection onto PCs can actually make classes of data inseparable that were previously separable before PCA."

— Wagner and Rondinelli, Frontiers in Materials, https://www.frontiersin.org/articles/10.3389/fmats.2016.00028/full



Not all methods will be useful for all datasets.