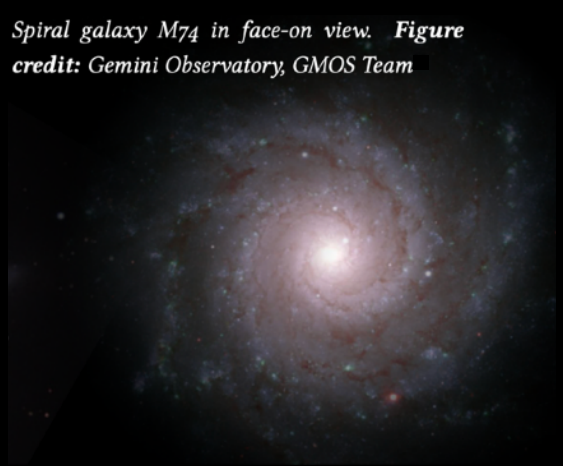
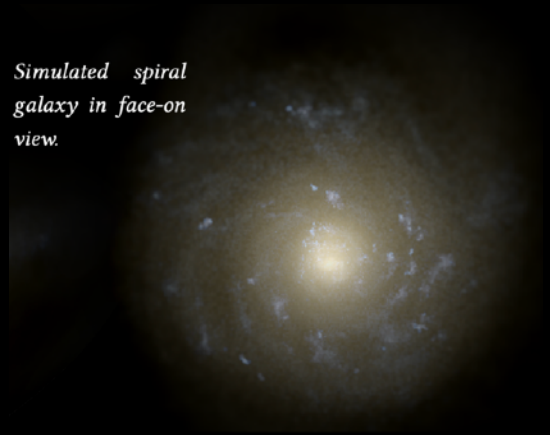
ASTR4004

COMPUTATIONAL ASTRONOMY

Week 8 https://github.com/svenbuder/astr4004_2025_week8



Spiral galaxy M74 in face-on view. ***Figure credit:*** *Gemini Observatory, GMOS Team*



Simulated spiral galaxy in face-on view.

Figure credit: *Tobias Buck*

DR SVEN BUDER | RESEARCH SCHOOL OF ASTRONOMY AND ASTROPHYSICS

Australian National University

DIMENSIONALITY REDUCTION

because high-dimensional data is hard to store, analyse, interpret, and visualise

and often highly correlated
(and therefore somewhat redundant)

Australian National University

62x47
(2914)

PCA for Image
Dimension Reduction:

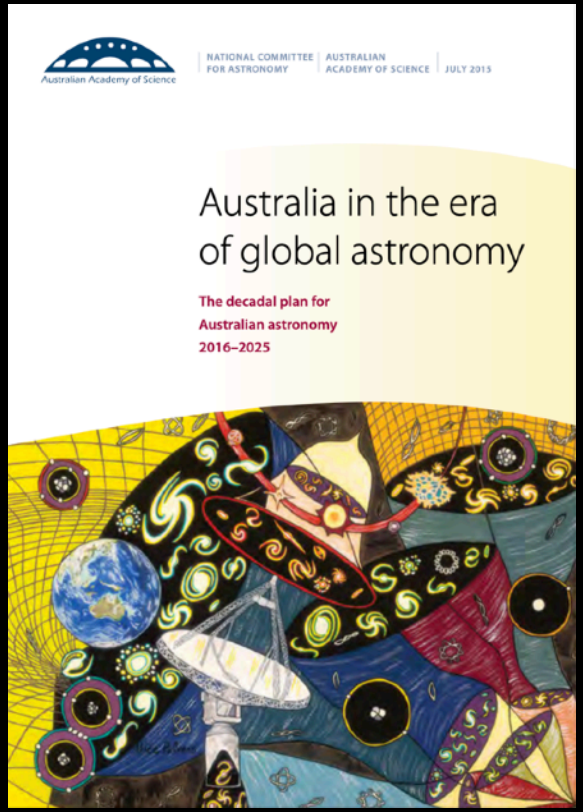
Starting from the
"average human face"

Adding more and more
detail (explain "variance")

150
each

Source: sklearn image collection and tutorials

What does that have to do with astronomy?



1. How did the first stars and galaxies transform the Universe?

2. What is the nature of dark matter and dark energy?

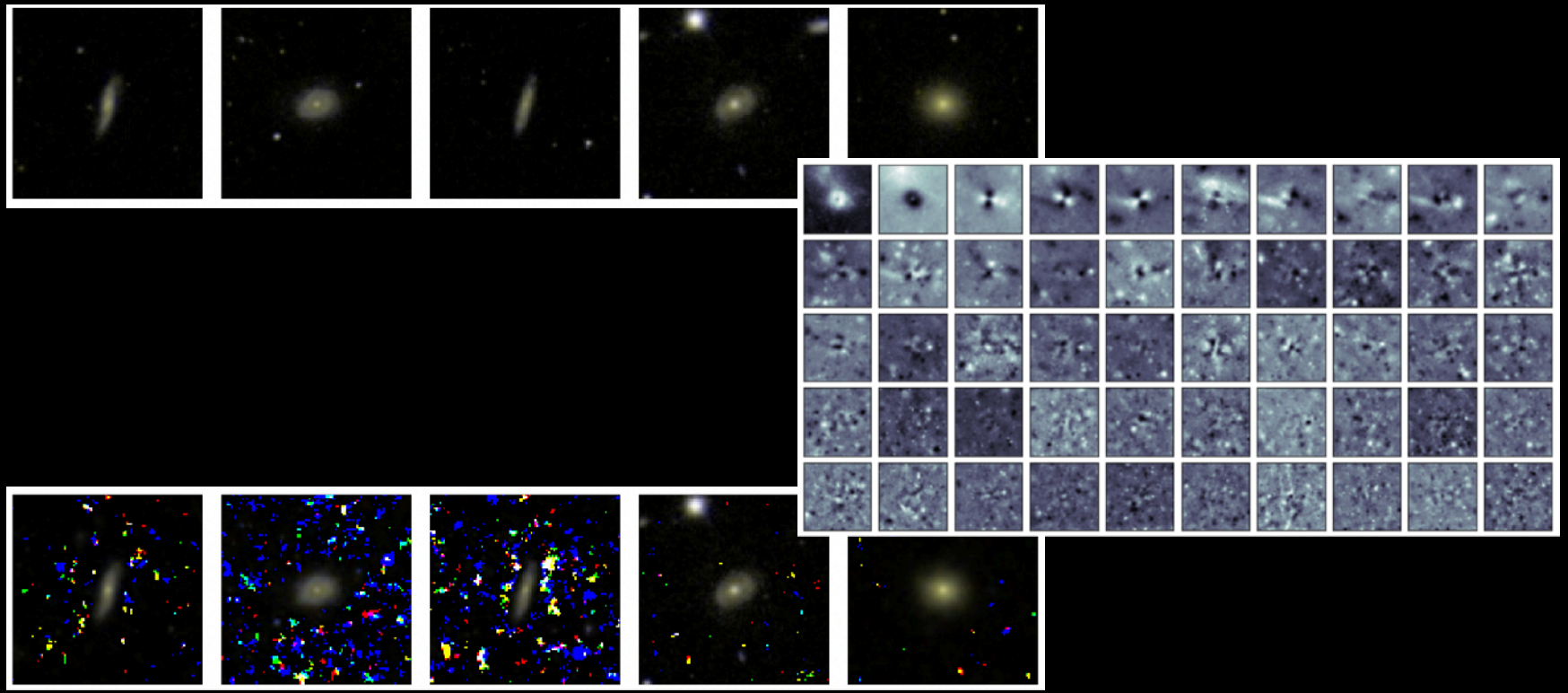
3. How do galaxies form and evolve across cosmic time?

4. How do stars and planets form?

5. How are elements produced by stars and recycled through galaxies?

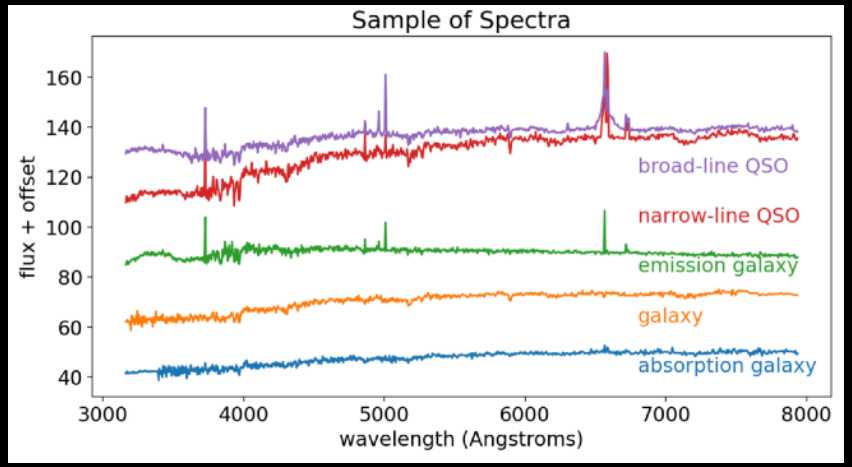
6. What is the nature of matter and gravity at extreme densities?

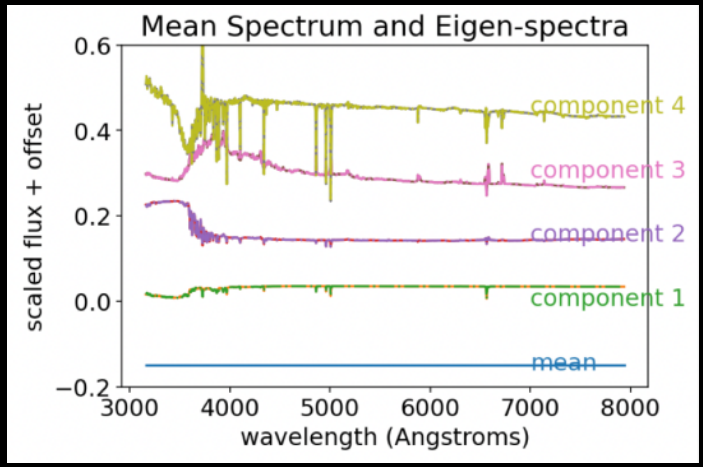
What does that have to do with astronomy?



Source: sklearn tutorials, GalaxyZoo, astroML@github

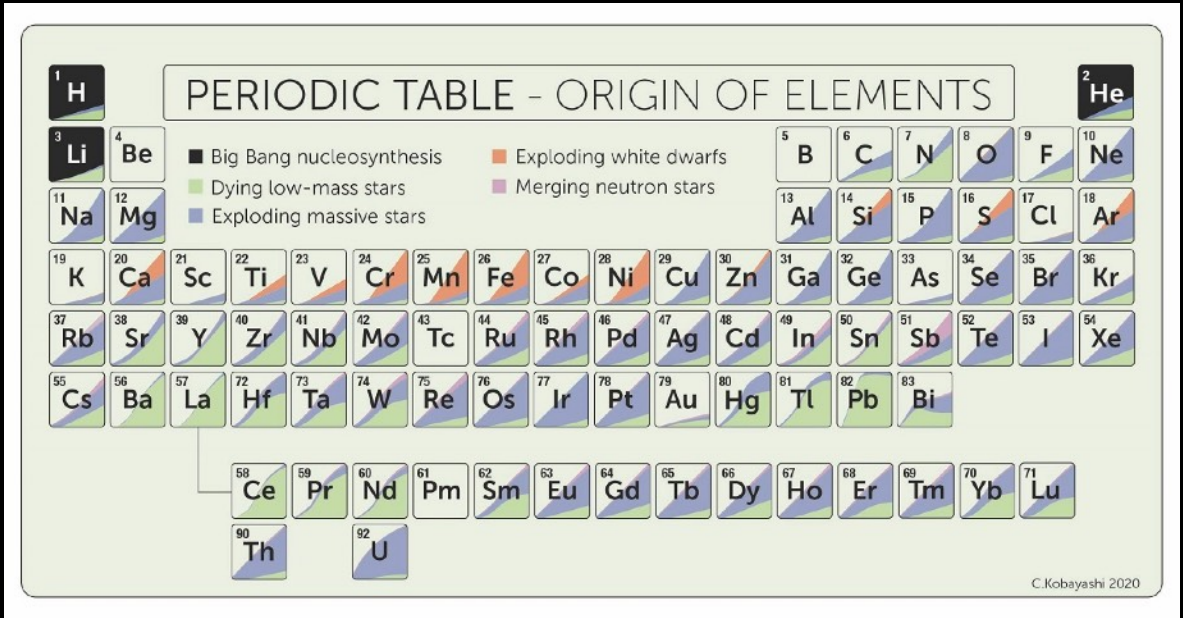
What does that have to do with astronomy?



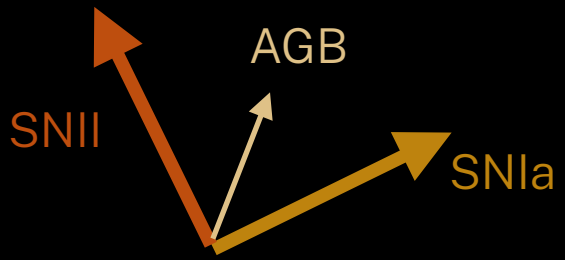


Source: sklearn tutorials, GalaxyZoo, astroML@github

What does that have to do with astronomy?



Kobayashi et al. (2020)

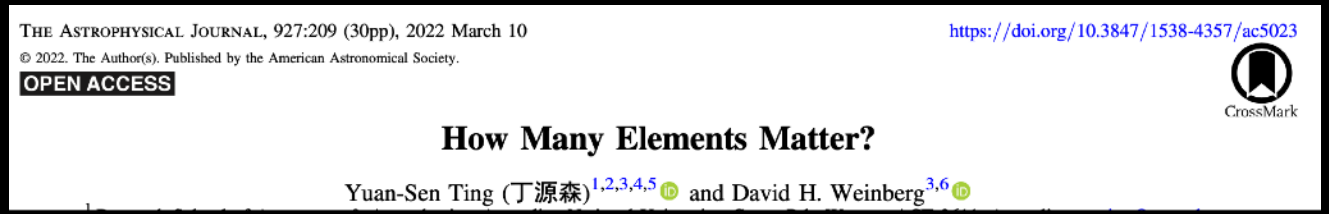


What does that have to do with astronomy?

The Astrophysical Journal, 883:177 (16pp), 2019 October 1 https://doi.org/10.3847/1538-4357/ab3e3c

In the Galactic Disk, Stellar [Fe/H] and Age Predict Orbits and Precise [X/Fe]

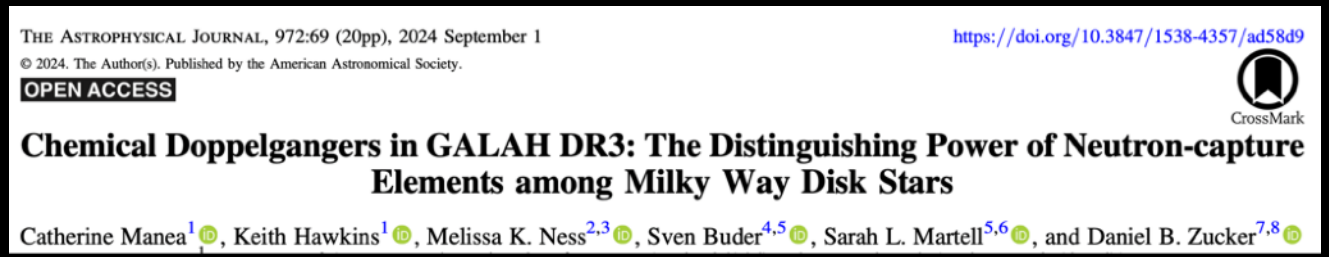
M. K. Ness[1,2], K. V. Johnston[1], K. Blancato[1], H-W. Rix[3], A. Beane[4], J. C Bird[5], and K. Hawkins[6]



The Astrophysical Journal, 927:209 (30pp), 2022 March 10 https://doi.org/10.3847/1538-4357/ac5023

How Many Elements Matter?

Yuan-Sen Ting (丁源森)[1,2,3,4,5] and David H. Weinberg[3,6]



The Astrophysical Journal, 972:69 (20pp), 2024 September 1 https://doi.org/10.3847/1538-4357/ad58d9

Chemical Doppelgangers in GALAH DR3: The Distinguishing Power of Neutron-capture Elements among Milky Way Disk Stars

Catherine Manea[1], Keith Hawkins[1], Melissa K. Ness[2,3], Sven Buder[4,5], Sarah L. Martell[5,6], and Daniel B. Zucker[7,8]

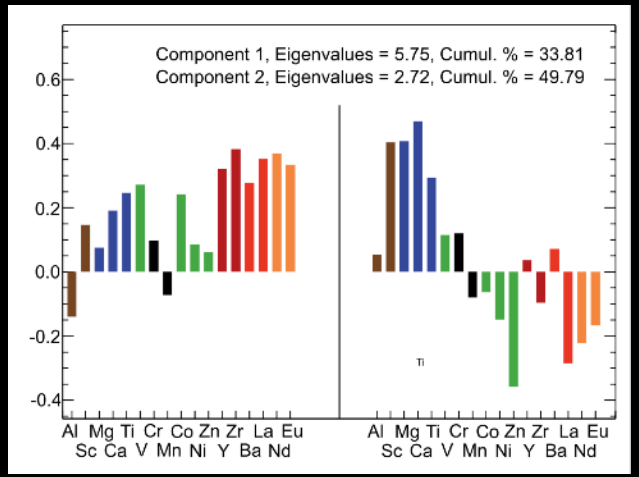
How many "chemical dimensions" do stars have?

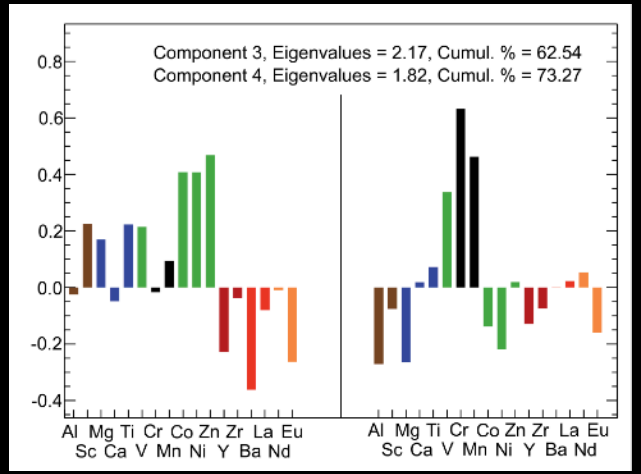
Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

Mon. Not. R. Astron. Soc. **421,** 1231–1255 (2012) doi:10.1111/j.1365-2966.2011.20387.x

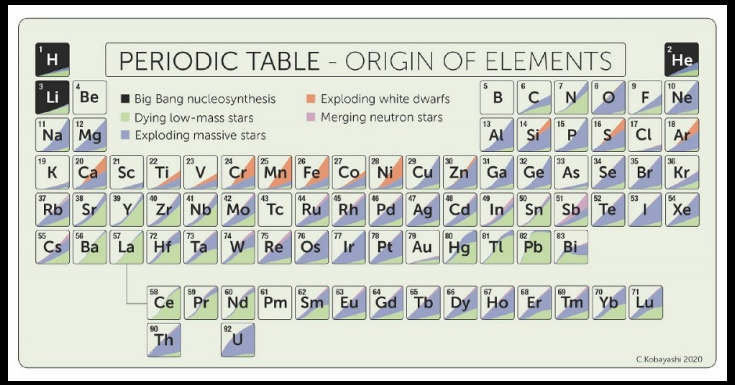
Principal component analysis on chemical abundances spaces

Y. S. Ting,[1,2★] K. C. Freeman,[1] C. Kobayashi,[1,3] G. M. De Silva[4] and J. Bland-Hawthorn[5]

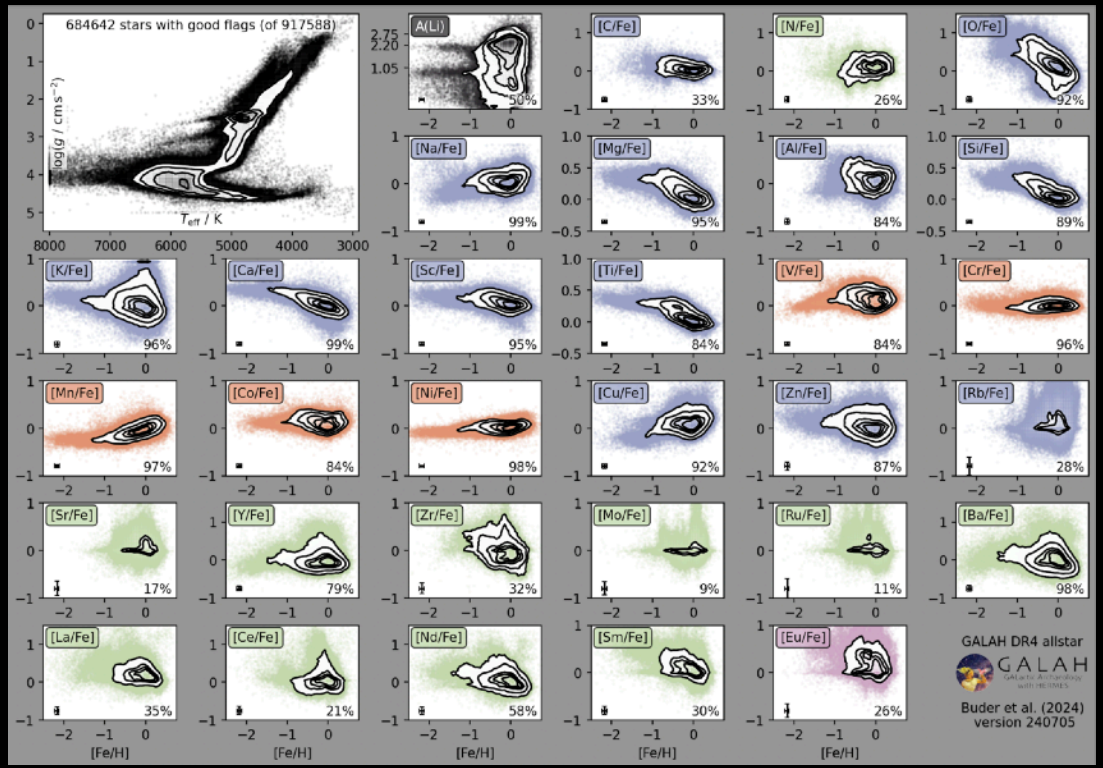




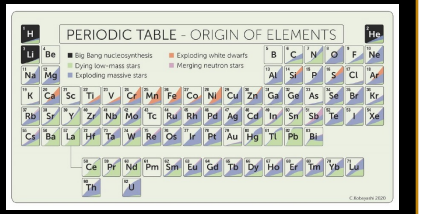
How many "chemical dimensions" do stars have?



Kobayashi et al. (2020)

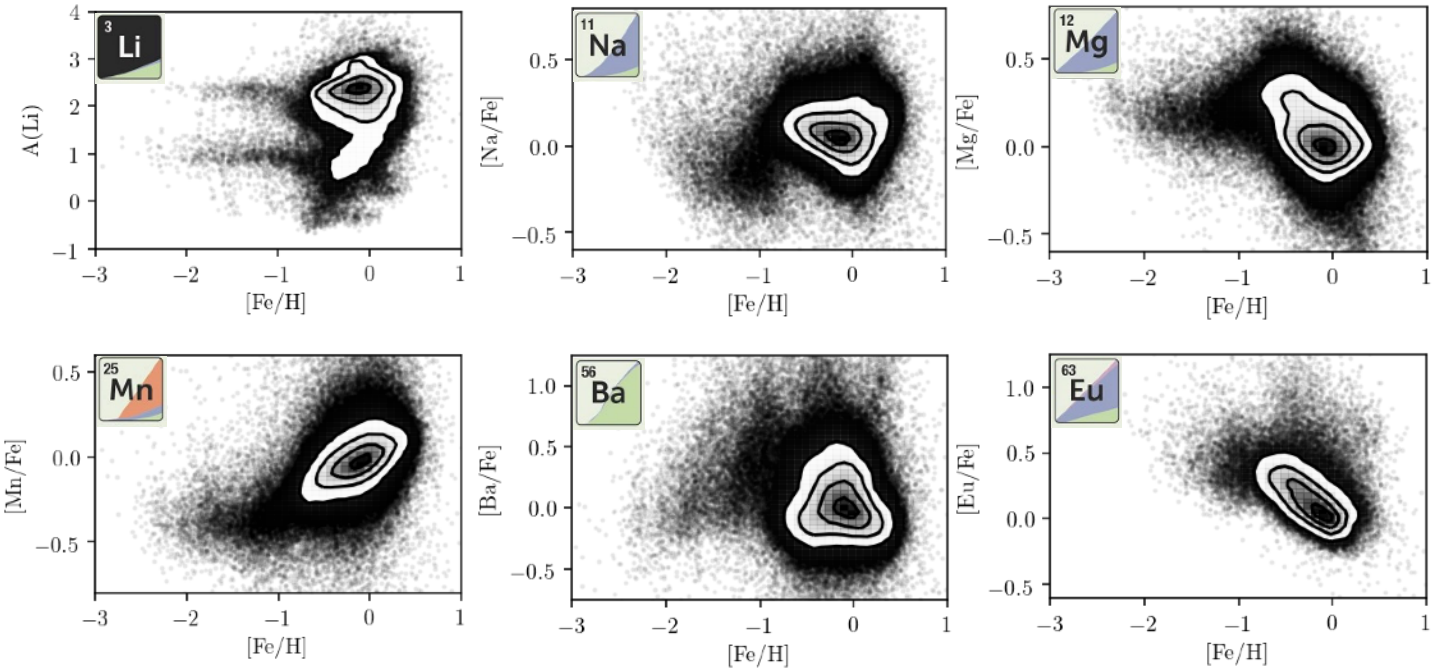


Buder et al. (2024)



Kobayashi et al. (2020)

Big Bang Nucleosynthesis | Exploding Massive Stars* | Exploding White Dwarfs | Asymptotic Giant Branch Stars | Merging Neutron Stars



*Core-Collapse Supernovae, including Supernovae II, Hypernovae, Electron-Capture Supernovae, Magneto-Rotational Supernovae

DIMENSIONALITY REDUCTION

Do we actually just have to measure ~5 elements?

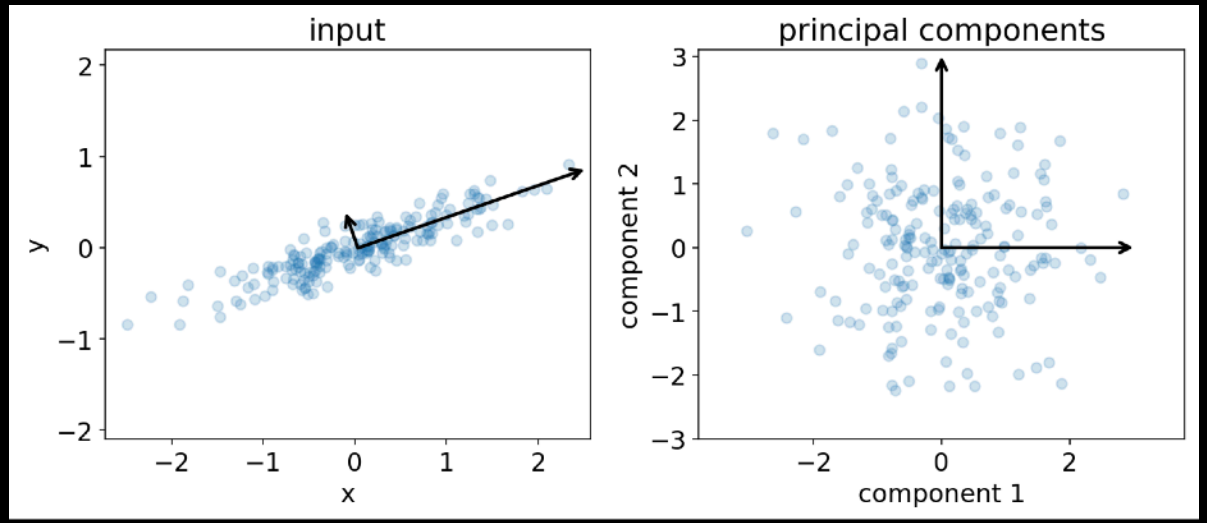
Do we just need 5 principal components?

Australian National University

Principal Component Analysis (PCA)

Dimensionality Reduction: Reduces the number of variables (features) in a dataset while maintaining most of its important information.

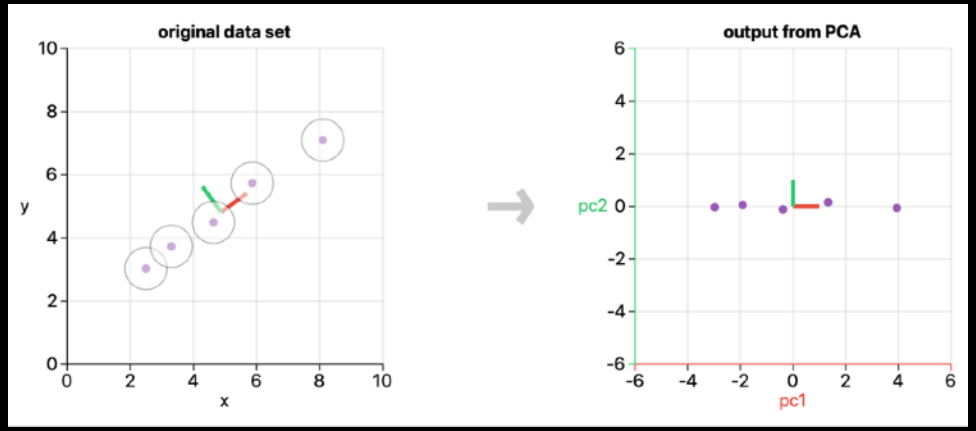
Principal Components: New variables that are linear combinations of the original ones, ordered by the amount of variance they explain.



Principal Component Analysis (PCA)

Dimensionality Reduction: Reduces the number of variables (features) in a dataset while maintaining most of its important information.

Principal Components: New variables that are linear combinations of the original ones, ordered by the amount of variance they explain.



Interactive version by Andy Casey: http://astrowizici.st/teaching/phs5000/11/

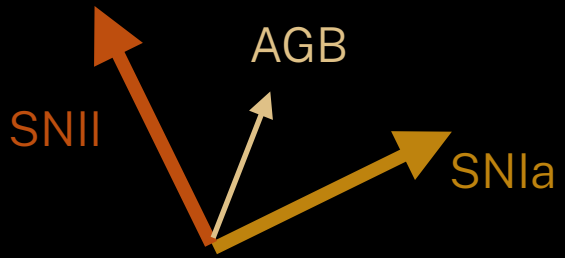
Principal Component Analysis (PCA)

Dimensionality Reduction: Reduces the number of variables (features) in a dataset while maintaining most of its important information.

Principal Components: New variables that are linear combinations of the original ones, ordered by the amount of variance they explain.

Explained Variance: Each principal component explains part of the total variance in the data.

Orthogonality: Principal components are orthogonal (uncorrelated) to each other.



Step 1: Standardise the Data

PCA is affected by the scale of the variables

Standardisation is necessary

This step ensures that each feature has a mean of 0 and a standard deviation of 1.

```python
from sklearn.datasets import load_iris
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Load Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Standardize the data
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Convert the data to a DataFrame for easier visualization
df_scaled = pd.DataFrame(X_scaled, columns=iris.feature_names)
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

$$x_{i,\text{scaled}} = \frac{x_i - \mu_i}{\sigma_i}$$

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	-0.900681	1.019004	-1.340227	-1.315444
1	-1.143017	-0.131979	-1.340227	-1.315444
2	-1.385353	0.328414	-1.397064	-1.315444
3	-1.506521	0.098217	-1.283389	-1.315444
4	-1.021849	1.249201	-1.340227	-1.315444

Step 2: Covariance Matrix

The covariance matrix shows how features for *m* samples vary together:

$$C = \frac{1}{m-1} X_{\text{scaled}}^T X_{\text{scaled}}$$

$$C \cdot v = \lambda \cdot v$$

Eigenvector (principal component): v

Eigenvalue (explained variance): λ

Again: sklearn's PCA already does that

```python
from sklearn.decomposition import PCA

# Perform PCA with 4 components (since we have 4 features in Iris)
pca = PCA(n_components=4)
pca.fit(X_scaled)

# Eigenvalues (explained variance)
explained_variance = pca.explained_variance_ratio_
print("Explained Variance Ratio:", explained_variance)

# Eigenvectors (principal components)
print("Principal Components:\n", pca.components_)
```

Step 3: Eigenvectors and Eigenvalues

The eigenvectors of the covariance matrix represent the directions (principal components) in which the data varies the most, and the larger eigenvalues λ explain more variance:

$$C \cdot v - \lambda \cdot v = 0$$

I: Identity matrix

$$(C - \lambda I) \cdot v = 0$$

$$\det(C - \lambda I) = 0$$

$$\lambda_1 = \frac{7 + \sqrt{5}}{2}, \quad \lambda_2 = \frac{7 - \sqrt{5}}{2}$$

$$C = \begin{bmatrix} 4 & 1 \\ 1 & 3 \end{bmatrix}$$

$$C - \lambda I = \begin{bmatrix} 4 - \lambda & 1 \\ 1 & 3 - \lambda \end{bmatrix}$$

$$\det(C - \lambda I) = \det \begin{bmatrix} 4 - \lambda & 1 \\ 1 & 3 - \lambda \end{bmatrix} = \lambda^2 - 7\lambda + 11$$

$$\begin{bmatrix} \frac{1-\sqrt{5}}{2} & 1 \\ 1 & \frac{-1-\sqrt{5}}{2} \end{bmatrix} \cdot \begin{bmatrix} v_{11} \\ v_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 0.851 \\ 0.526 \end{bmatrix} \quad v_2 = \begin{bmatrix} -0.526 \\ 0.851 \end{bmatrix}$$

Normalised(!) eigenvectors

Step 4: Sort Eigenvalues

The eigenvectors of the covariance matrix represent the directions (principal components) in which the data varies the most, and the larger eigenvalues λ explain more variance:

Once sorted, you can then combine the eigenvectors to the matrix of eigenvectors:

$$V = \begin{bmatrix} -0.851 & -0.526 \\ -0.526 & 0.851 \end{bmatrix}$$

Note how we can change signs of eigenvectors, because they describe an axis!

And transform the scaled data into the new principal component space:

$$X_{\text{new}} = X_{\text{scaled}} \cdot V$$

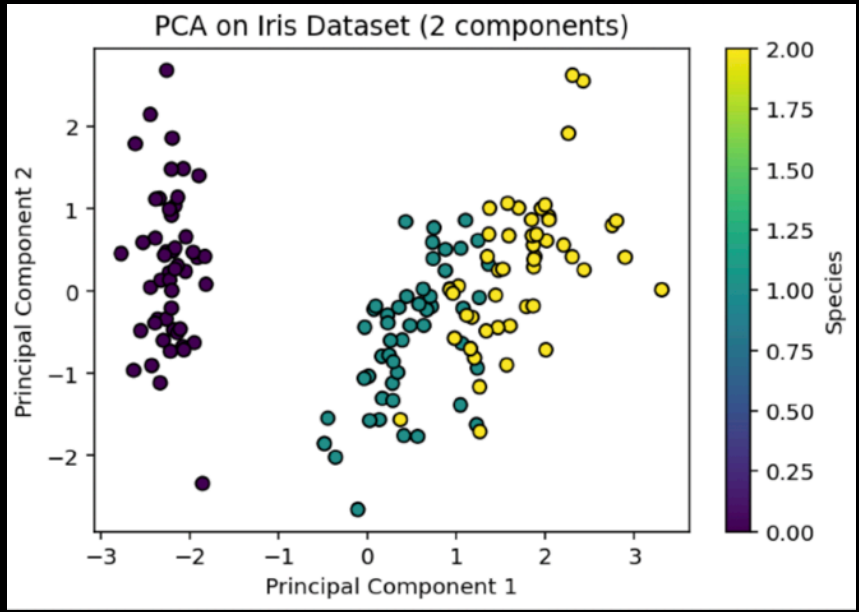
In practice?

```python
# Apply PCA and reduce to 2 components
pca_2d = PCA(n_components=2)
X_pca_2d = pca_2d.fit_transform(X_scaled)

# Create a DataFrame for visualization
df_pca = pd.DataFrame(X_pca_2d, columns=['PC1', 'PC2'])
df_pca['species'] = y

# Visualize the first two principal components
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.scatter(df_pca['PC1'], df_pca['PC2'], c=df_pca['species'],
            cmap='viridis', edgecolor='k')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA on Iris Dataset (2 components)')
plt.colorbar(label='Species')
plt.show()
```

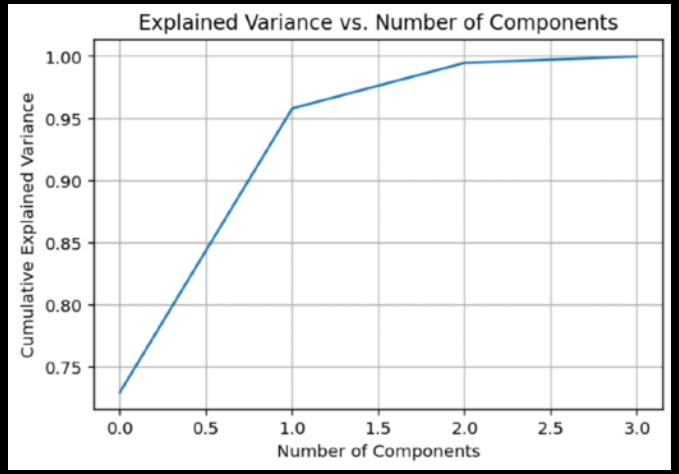


Step 5: Choose the number of components

We can determine how many principal components to retain by examining the explained variance. Often, we want to retain components that together explain 95% of the variance.

```
# Fit PCA to retain 95% of the variance
pca_full = PCA().fit(X_scaled)

# Plot the cumulative explained variance
plt.figure(figsize=(8, 6))
plt.plot(np.cumsum(pca_full.explained_variance_ratio_))
plt.xlabel('Number of Components')
plt.ylabel('Cumulative Explained Variance')
plt.title('Explained Variance vs. Number of Components')
plt.grid()
plt.show()
```



LET'S APPLY PRINCIPAL COMPONENT ANALYSIS ONTO ASTRONOMICAL DATA

Australian National University

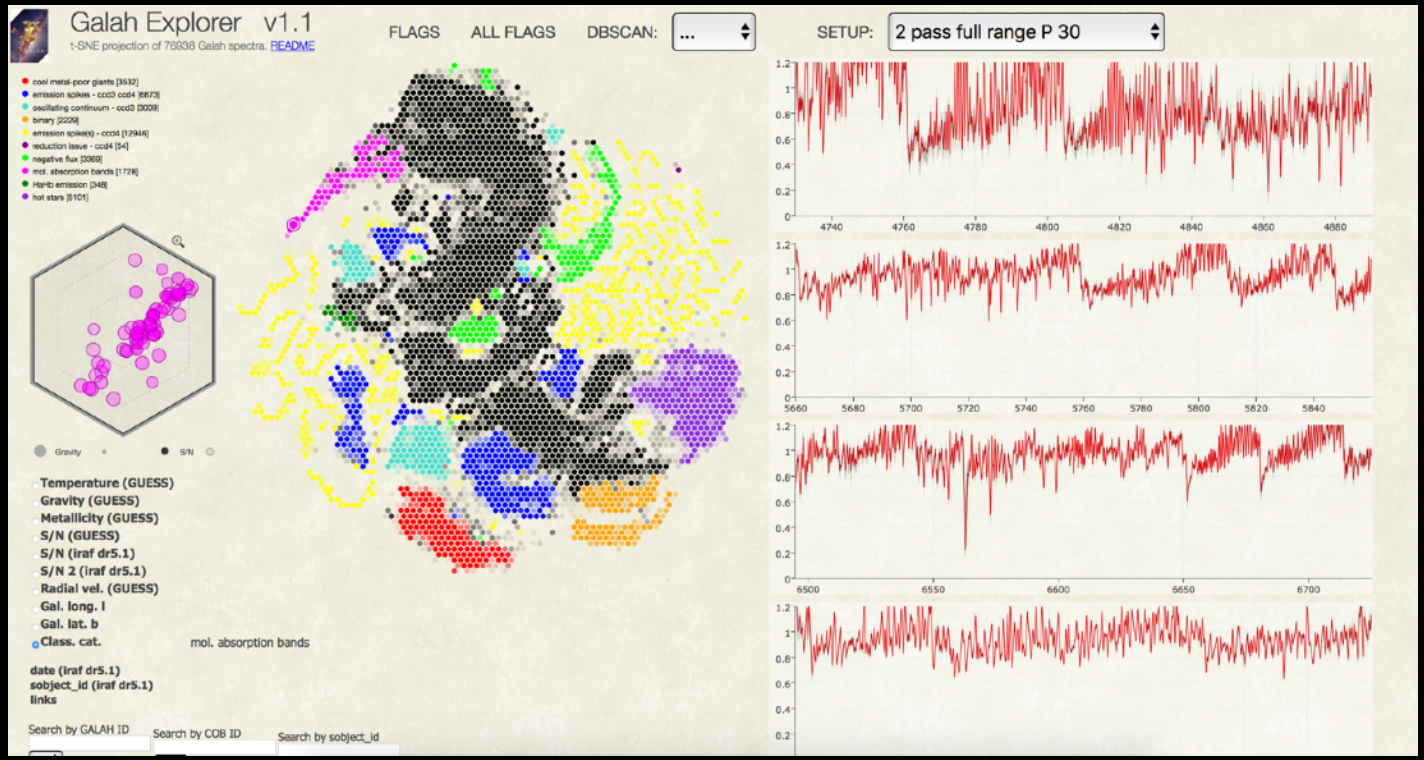
BREAK UNTIL 2PM

Australian National University

AND WHAT IF WE WANT TO REDUCE HIGH-DIMENSIONAL DATA LIKE SPECTRA?

Australian National University

tSNE is useful for exploring data



tSNE (Traven et al. 2017)

tSNE in a nutshell (just for completeness)

For each pair of data points x_i and x_j in the original high-dimensional space, t-SNE computes a similarity using a conditional probability that represents how likely point x_j would pick x_i as its neighbour.

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)}$$

The joint probability between points x_i and x_j is symmetrized for the N points:

High dimensional similarities:

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2N}$$

Low dimensional similarities:

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

Goal: Find a low dimensional representation that minimizes difference between p_{ij} and q_{ij}.

How? Kullback-Leibler divergence:

$$\mathrm{KL}(P||Q) = \sum_{i \neq j} p_{ij} \log \left(\frac{p_{ij}}{q_{ij}} \right)$$

But be careful!

tSNE is good for exploration, but there are a lot of hyperparameters that can amplify differences that are not really important (e.g. you can sort spectra based on that 1 bad column of your CCD).

There is also other dimensionality reduction methods, such as UMAP