PGE 383
Dimensionality Reduction

- Curse of Dimensionality
- Dimensionality Reduction
- Principal Component Analysis

Michael Pyrcz, The University of Texas at Austin

Motivation

- We work with highly multivariate datasets
- Projection to a lower dimension may improve intepretability, and modeling accuracy by
  - avoiding overfit and multicolinearity
- May provide opportunities for feature engineering, working with features that have more information

PGE 383
Dimensionality Reduction

- **Curse of Dimensionality**

Michael Pyrcz, The University of Texas at Austin

Multivariate

- One of the definitions of Big Data is variety
  - This suggests massively multivariate datasets
- Traditional reservoir modeling workflows were bivariate
  - Facies, then porosity in facies and permeability constrained to porosity
  - The most complicated simulation is permeability accounting for the joint porosity simulated realization
- Unconventionals, and Whole Earth Models
  - Require inclusion many more variables
  - We need to model facies, porosity, geomechanical properties, geophysical properties, total organic carbon, maturity etc.
- When working with Multivariate it is very challenging:
  - Visualize
  - Detect relationships and patterns

Curse of Dimensionality

- **Working with more features / variables is harder!**

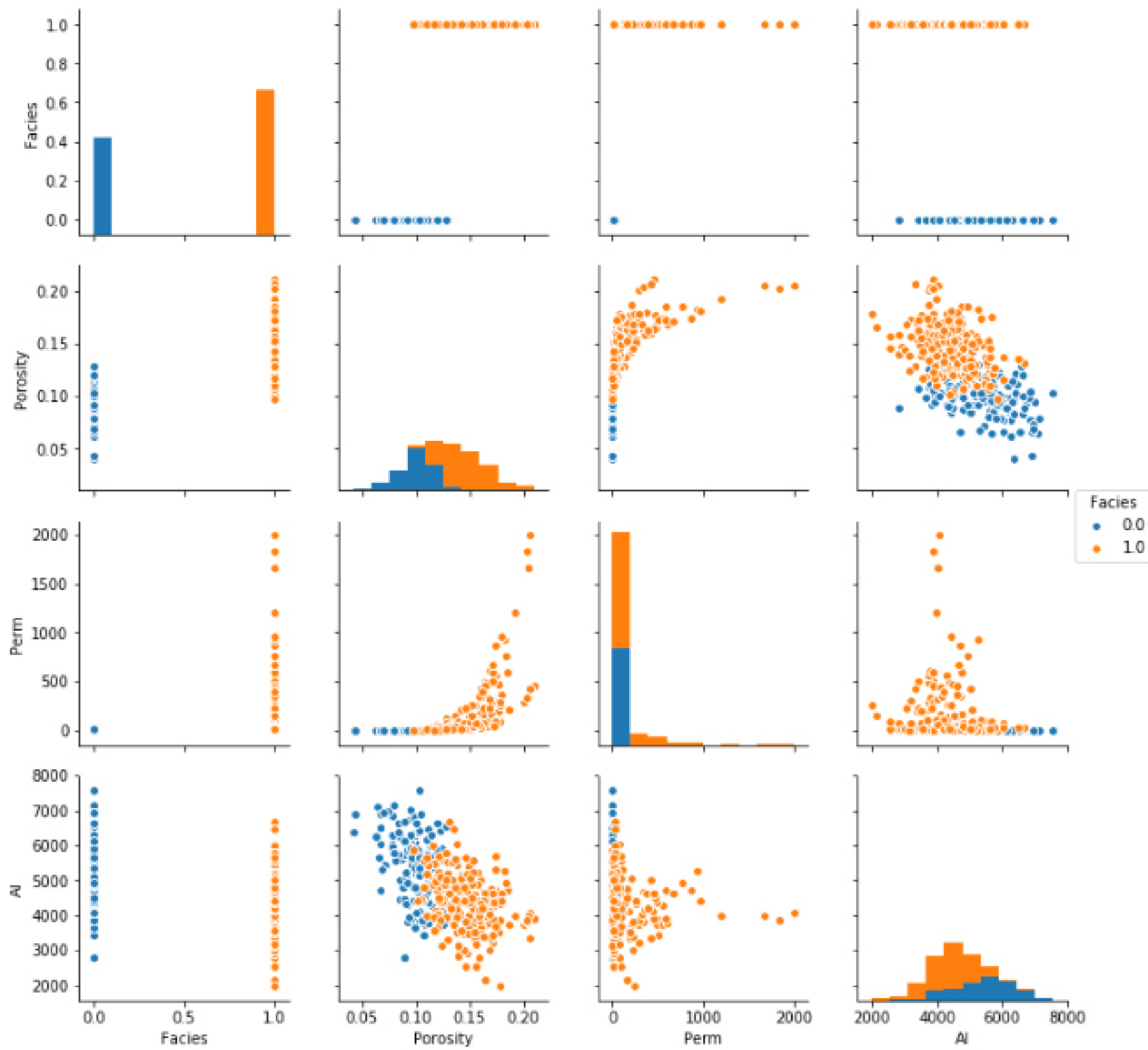
    1. More difficult to visualize
    2. More data are required to infer the joint probabilities
    3. Less coverage
    4. More difficult to interrogate / check the model
    5. More likely redundant
    6. More complicated, more likely overfit

Curse of Dimensionality

- **Consider this:**
  - 4 predictor features
  - 1 response feature (not shown)

- What are the relationships between features?

- Are there constraints?



Curse of Dimensionality

- **Consider any joint probability:**

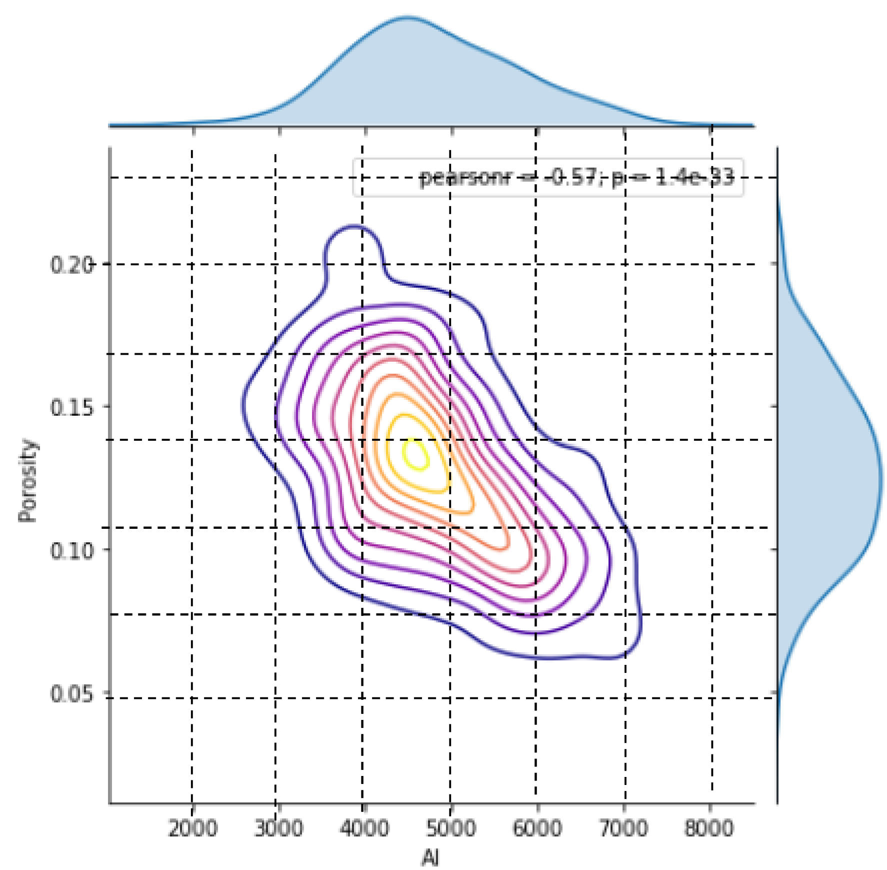
  $P(X_1 \cap, \ldots, \cap X_m)$ the joint probability of $X_1, \ldots, X_m$

- Now move to 2 features (m=2)

$$P\left(X_1^i \le X \le X_1^{i+1}, X_2^j \le X \le X_2^{j+1}\right) = \frac{n(X_1^i \le X \le X_1^{i+1}, X_2^j \le X \le X_2^{j+1})}{n}$$

$$\boldsymbol{n = Data/Bin \cdot Bins^m}$$

- This is optimistic, as it assumes uniform sampling



In each bin we are estimating a probability!
10 data in each bin = 640 data?

Curse of Dimensionality

Consider coverage:

- Now let's move to 2 features, each with 80% coverage

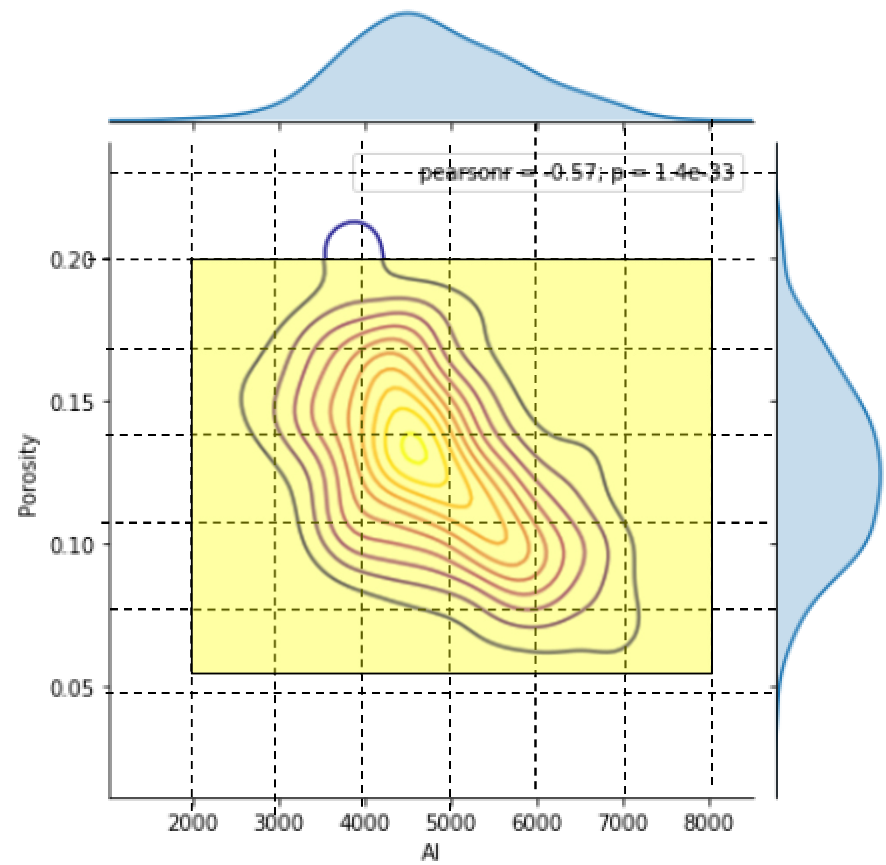
- How much of the solution space is covered?

$$0.8^D, \quad e.g.\, 0.8^2 = 0.64$$

- Even with exponential increase in number of data:

$$\boldsymbol{n = Data/Bin \cdot Bins^m}$$

coverage is decreasing as we increase the number of features!



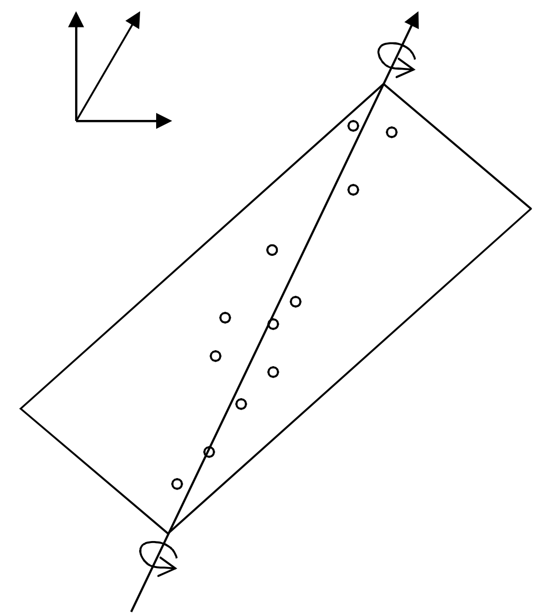
Multicollinearity
Feature Redundancy

“the existence of such a **high degree of correlation between supposedly independent variables** being used to estimate a dependent variable that the contribution of each independent variable to variation in the dependent variable cannot be determined”

- Merriam-Webster Online Dictionary

“In statistics, **multicollinearity** (also collinearity) is a phenomenon in which one predictor variable in a **multiple regression** model can be linearly predicted from the others with a substantial degree of accuracy.”

- Wikipedia



It is like fitting a plane to a line!

Motivation for Dimensionality Reduction

We get a better model with fewer, informative features than

'Throwing everything and the kitchen sink into the model!'

Fewer features for models are simpler, faster, easier to visualize and less likely overfit.

Feature Projection

- Dimensionality reduction by feature projection transforms the data to a lower dimension
- Given features, $X_1, \dots, X_m$ we would require $\binom{m}{2} = m(m-1)/2$ scatter plots to visualize just the two-dimensional scatter plots.
- Once we have 4 or more variables understanding our data gets very hard.
  - Recall the curse of dimensionality. It extends to visualization, not just sampling!

Feature Projection

- One solution, is to find a good lower dimensional, p, representation of the original dimensions m

- The Benefits:
  - Data storage / Computational Time
  - Visualization
  - Modeling with $m = 1, \ldots, M$ takes care of multicollinearity

- The Limitations:
  - It may be more difficult to understand the model
  - The new features $p = 1, \ldots, P$ are combinations of the original features $m = 1, \ldots, M$, lose their physical meaning!

Feature Projection

Wide variety of methods:

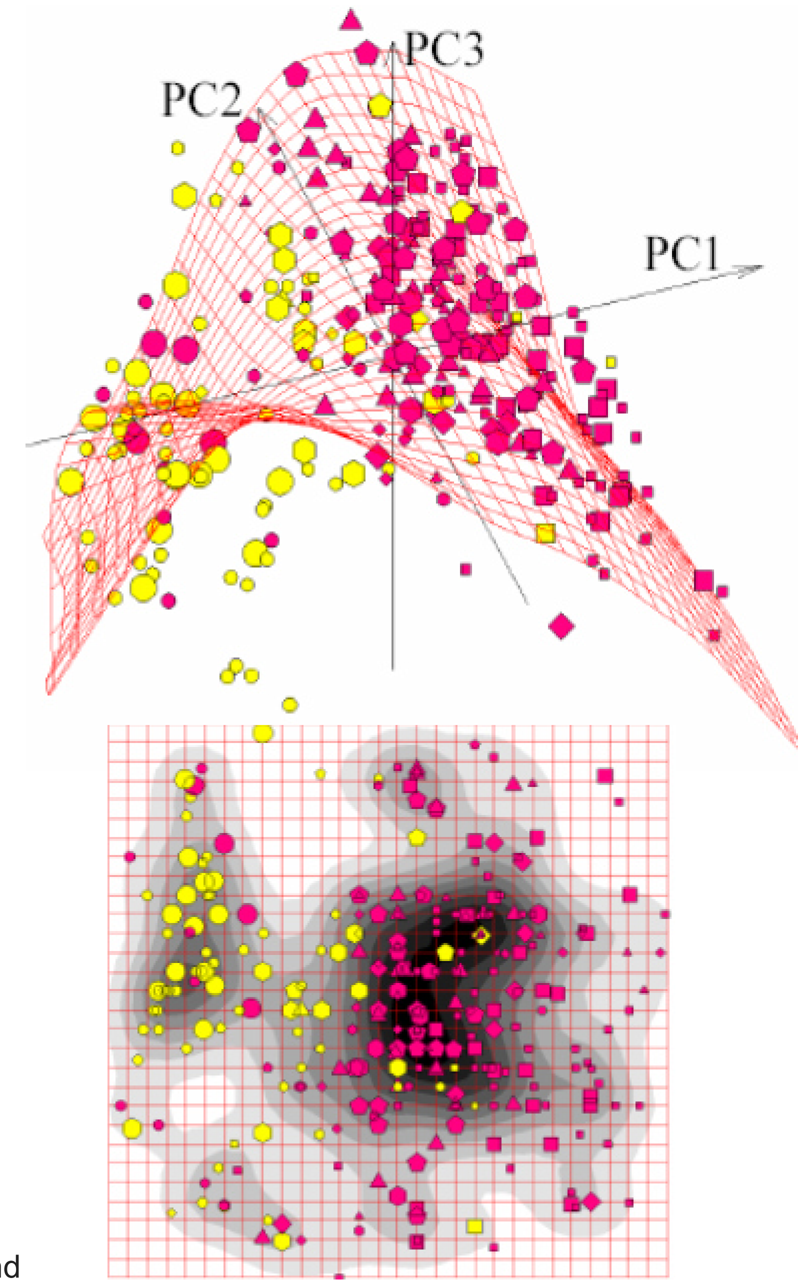
- Principal component analysis
  - Linear mapping of the data to lower dimensional space
  - Maximizes the variance explained by the reduced subset of features

- Kernel Principal component analysis
  - Nonlinear mapping of the data to lower dimensional space with the **kernel trick**
  - Kernel Trick – use of a kernel function to operate in higher dimensional feature space with only the ‘similarity’ between the data points

Feature Projection

Wide variety of methods:

- Factor Analysis
  - Like PCA, linear combinations of the features
  - Focus on inter-correlations
- Non-linear PCA
  - Form an embedded manifold for data approximation
  - Project the data onto the manifold
  - Natural geometric interpretation principal curves and manifolds

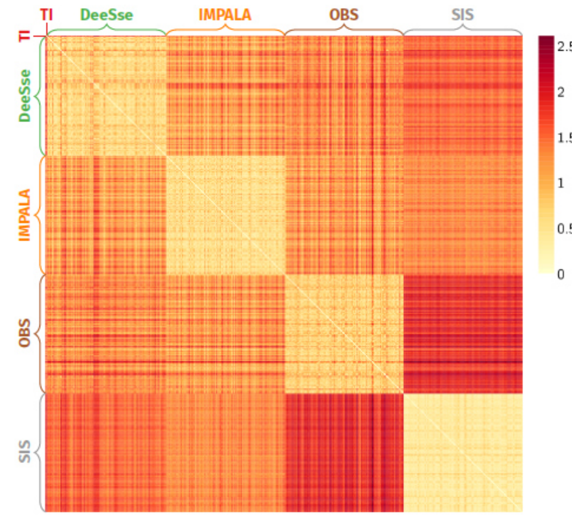


Nonlinear PCA 3D to 2D

Figure from Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques, Olivas E.S. et al Eds. Information Science Reference, IGI Global: Hershey, PA, USA, 2009. 28–59.

Feature Projection

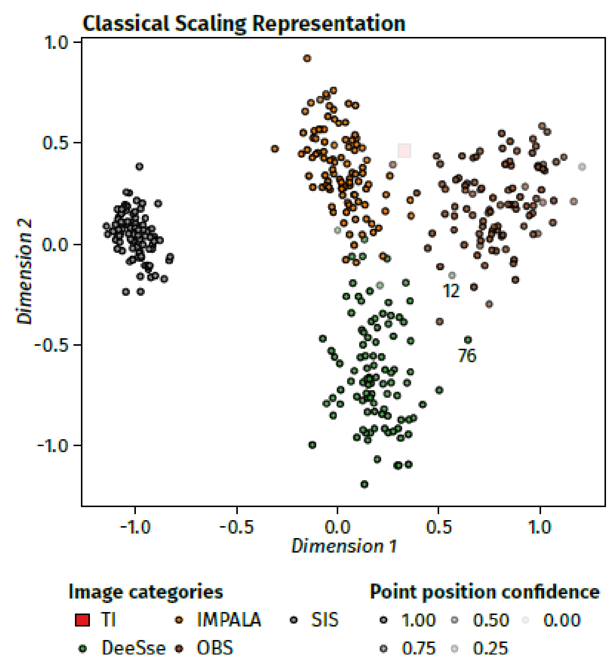
Dissimilarity based on combination of metrics: proportions, transitions, connectivity, shape, networks



Wide variety of methods:

- Multidimensional Scaling
  - Ordination technique for information visualization
  - Non-linear dimensional reduction
  - Given a matrix of pairwise distances between all data, project to lower dimensional space, P

  - such that the between sample distance is preserved as well as possible.



MDS to Visualize Model Uncertainty Space Sampled with Scenarios and Realizations

Figure from Rongier, G. Ph.D. thesis.

PGE 383
Dimensionality Reduction

- **Principal Component Analysis**

Michael Pyrcz, The University of Texas at Austin

Principal Components Analysis

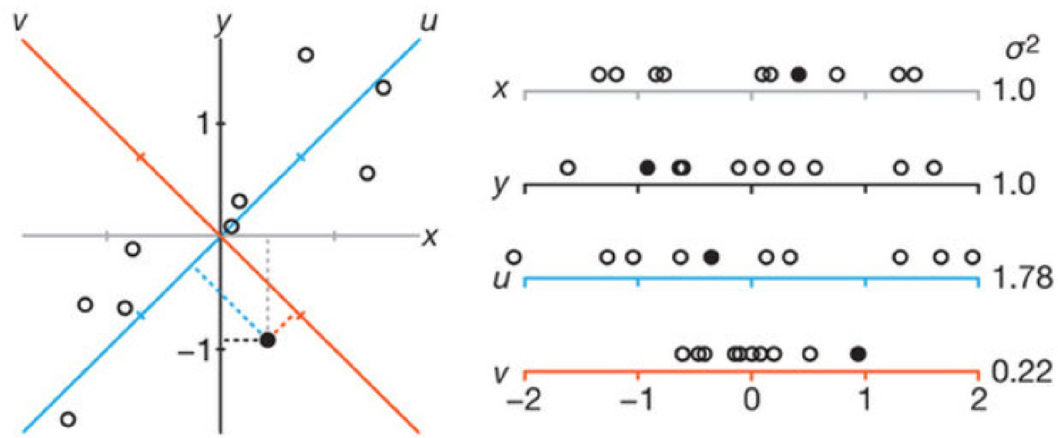
- Orthogonal Transformation
  - Convert a set of observations into a set of linearly uncorrelated variables known as principal components
- The number of principal components (p) available are $\min(n-1, m)$
  - Limited by the variables/features, m, and the number of data, n
- Components are ordered
  - First component describes the larges possible variance / accounts for as much variability as possible
  - Next component describes the largest possible remaining variance
  - Up to the maximum number of principal components

Principal Components Analysis

- **Eigen Values / Eigen Vectors**
  - The Eigen values are the variance explained for each component.
  - The Eigen vectors of the data covariance matrix are the principal components and the Eigen
  - Out of scope – just making the linkage

Principal Components Analysis

- **Finding the orthogonal projections in order of greatest variance described**
  - Start with regular 2D, data with x and y coordinates below.

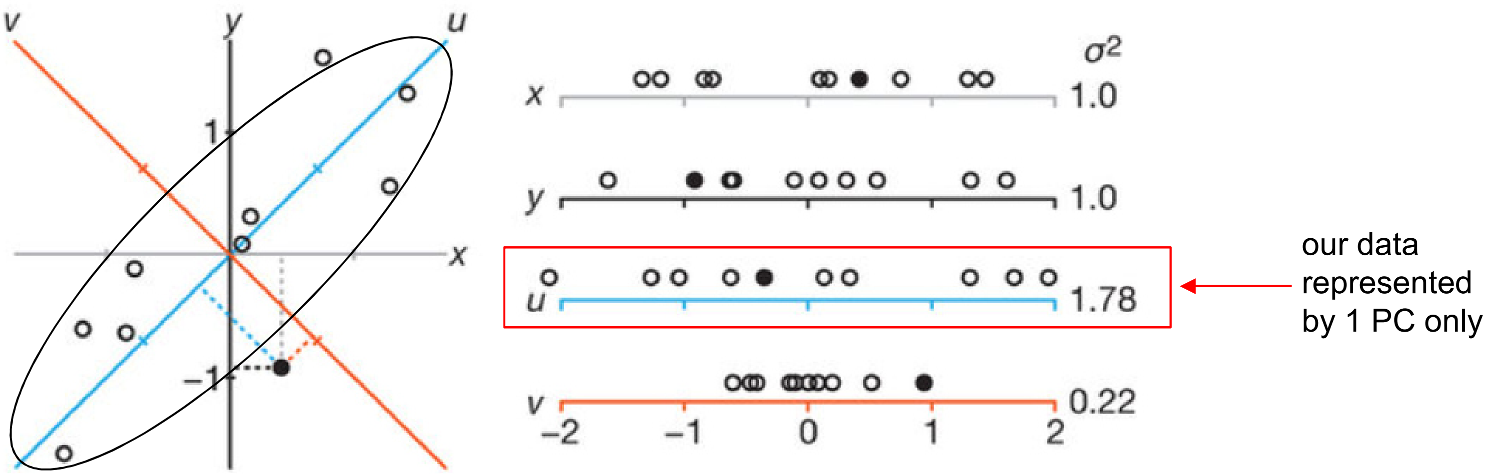


  - See the projections on to x and y axes. Note the data has equal variance in x and y. If you omitted x or y from the dataset you would lose a lot of information!
  - Find the rotation that would maximize the variance on the projection, u.
  - The 2nd axis is given as perpendicular to the first (determined since problem is 2D).

Lost image citation.

Principal Components Analysis

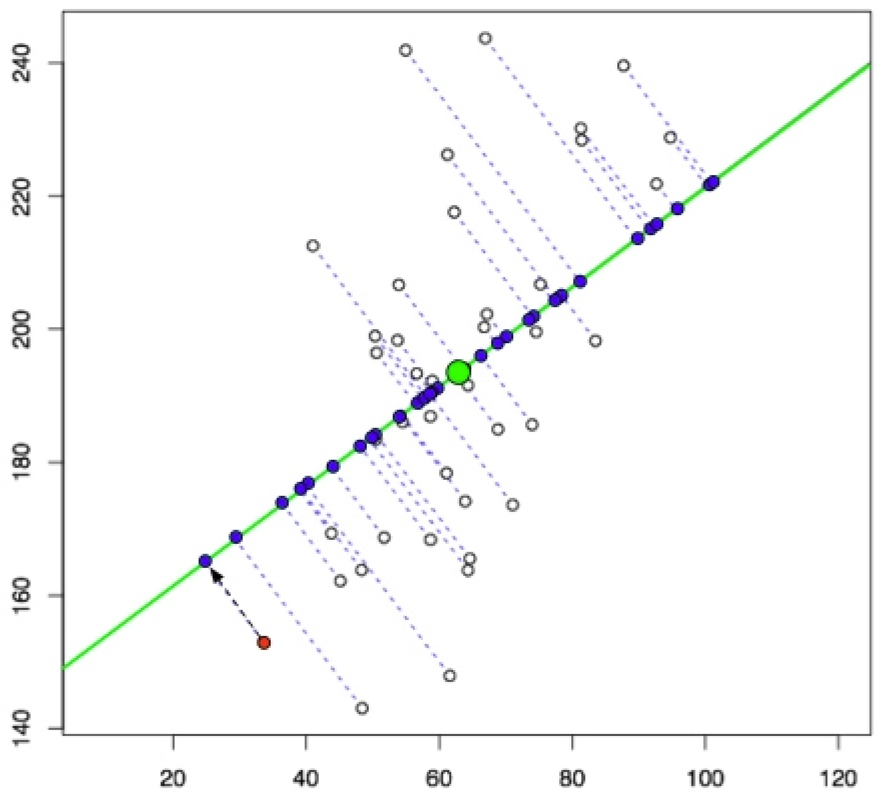
- It is fitting a m-dimensional ellipsoid to the data
  - The length of each axis indicates the amount of variance described by each component
  - Omitting that axis and the associated principal component from our representation of the dataset, we would lose information proportional to the length of the axis



Lost image citation.

Principal Components Analysis

- Graphical Representation
  - Line is the 1st principal component
  - Projection of points on line (**purple points**) are the 1st principal component scores
  - Given the problem is 2D the 2nd principal component is determined from the first (must be orthogonal)
  - If we approximated this dataset with just the 1st principal component for dimensional reduction, our approximation would be the **purple points**.
  - The first principal component maximizes the variance of the projected **purple points**.



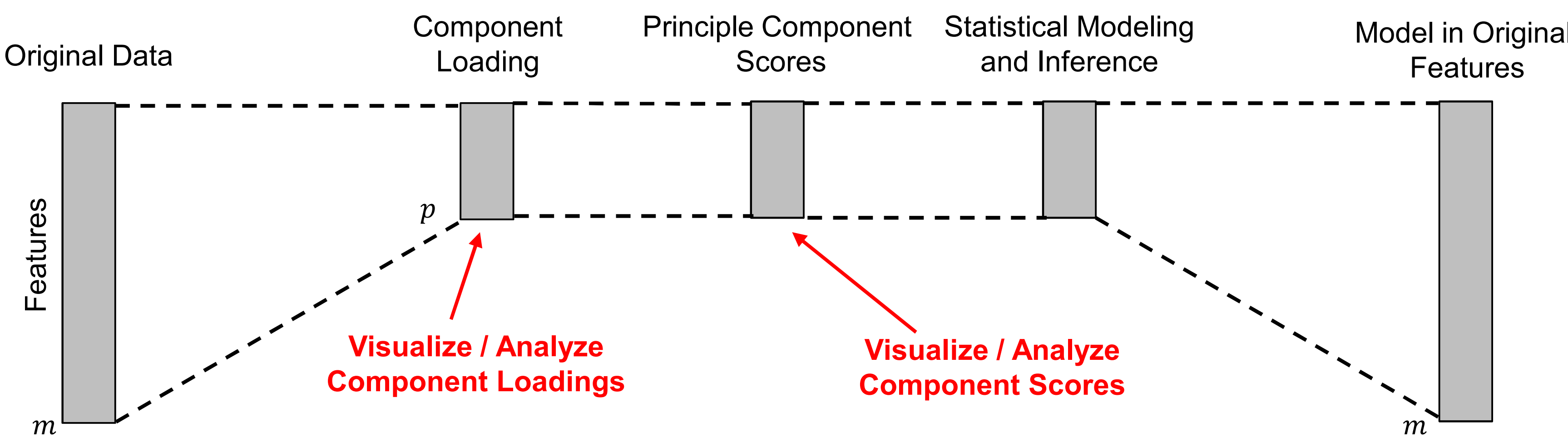
1st principal component, projects on the line are the 1st principal component scores (from https://liorpachter.wordpress.com/2014/05/26/what-is-principal-component-analysis/).

Principal Components Analysis Summary

- **Typical Workflow**

Visualize / Analyze / Model with Component Scores

Lower dimensional space, lower risk of overfit and easier to model.



Visualize / Analyze Component Loadings

The rotation that defines each independent component that describe components of the variance

Variable groupings

Important features (images) that describe the observed variability between training

Visualize / Analyze Component Scores

Set of independent Features that describe components of the variance

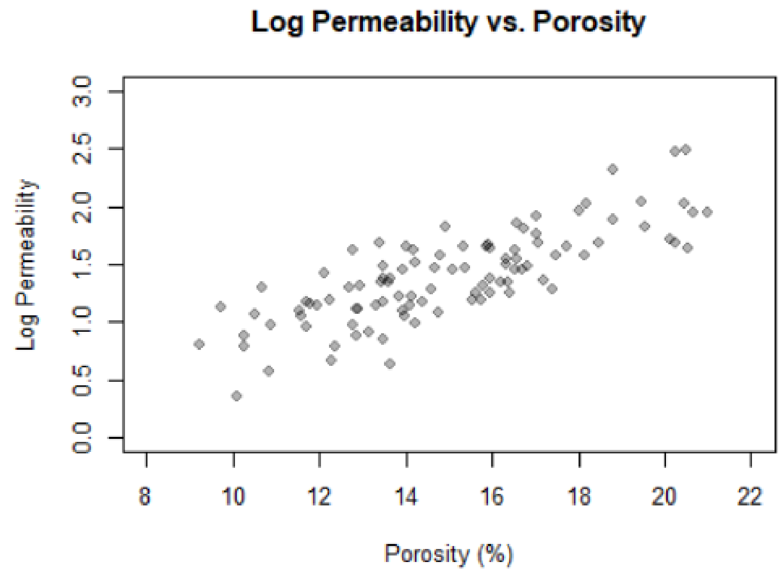
Principal Components Analysis

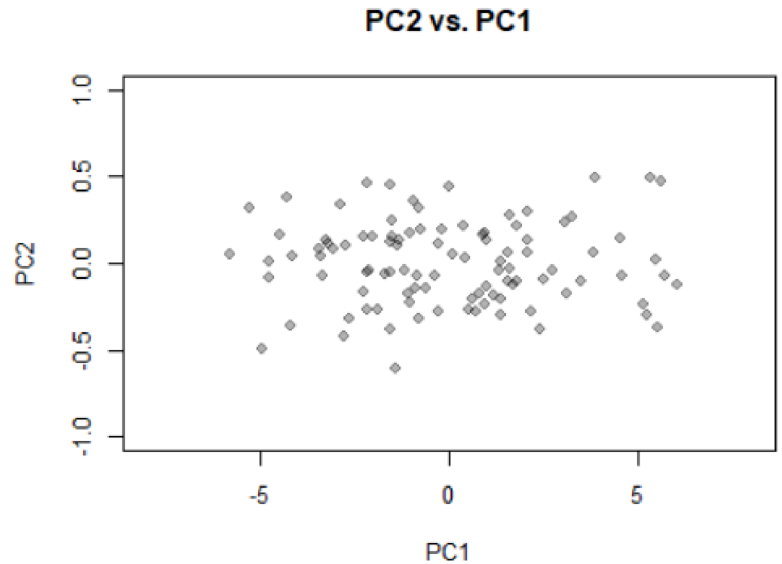
- Here we plot the original data and compare it to the plot of the principal component scores, $z_{i,1}$and $z_{i,1}$ for $i = 1, \dots, n$ data.

- We could just retain the first principal component score.

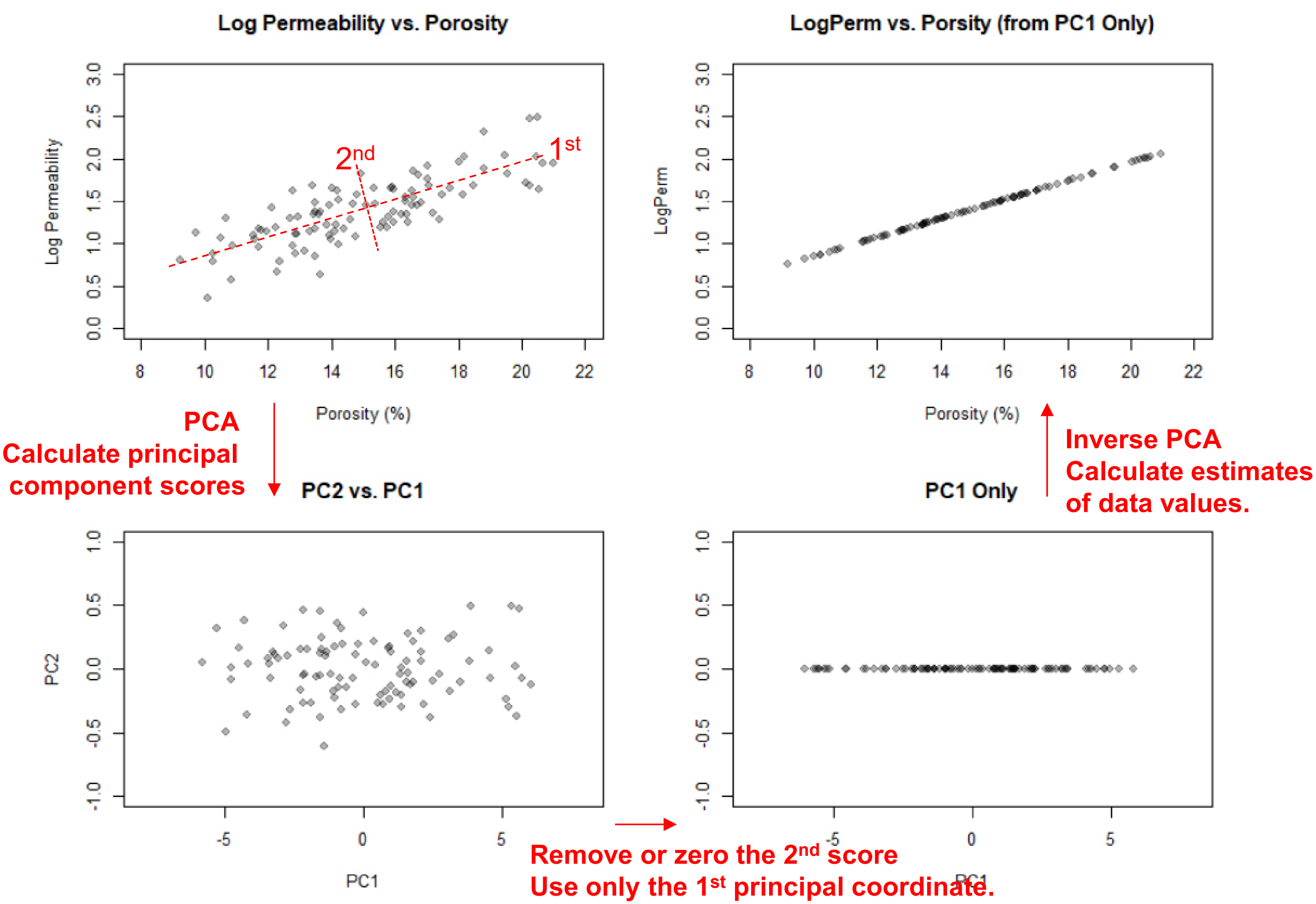
- How much information would we loose? How much of the variance would be explained?

- Let's try that.





Principal Components Analysis



Principal Components Analysis

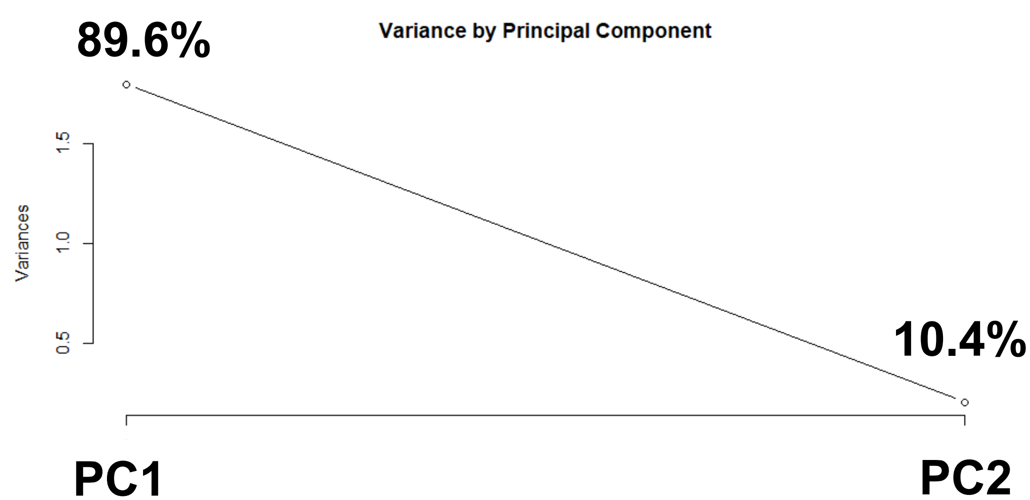
Variance Described by Each Principal Component

- So, how much variance did we capture with our components? We can calculate the proportion of variance explained by each principal component as:

$$PVE_k = \frac{1}{n}\sum_{i=1}^{n} z_{i,k}^2$$

- Should be monotonically decreasing for $k = 1, \ldots, K$.

- In our example:



Principal Components Analysis

What can you do with PCA?

Prediction:

- Reduce dimensions, build a model with the principal component scores and then restore to estimates the data values.
  - PCA regression, regression on the most important principal components

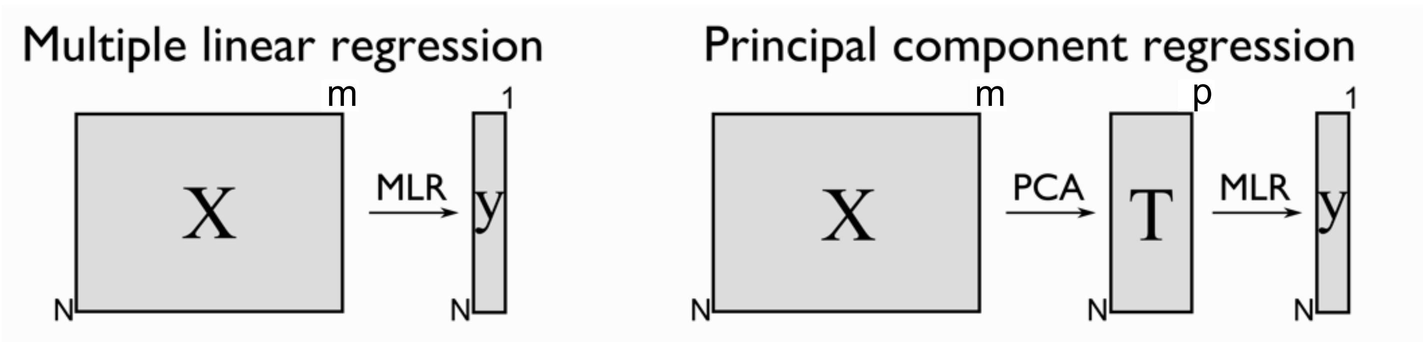


Image from: https://learnche.org/pid/latent-variable-modelling/principal-components-regression

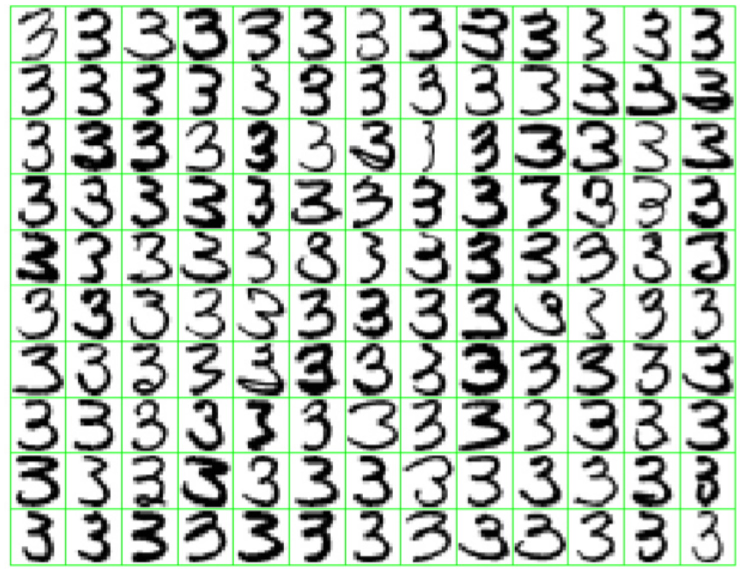
Inference:

- Understand our variables and how variance is partitioned
- Check for and mitigate multi-collinearity
  - Exclude principal components that have low variance

Principal Components Analysis Image Example

PCA with images:

- 130 examples of hand writing
- 16 x 16 grey scale images
- $m = 256$ dimensional



Comments:

- Clearly the images have commonality
- We can describe their variability with fewer than 256 features!

Workflow:

- Calculate the covariance matrix of all pixels with each other
- Results in a 256 x 256 covariance matrix
- Center by removing average of each pixel, then calculate the Eigen values and vectors (Singular value decomposition)

Figure and example from Hastie et al., 2009.

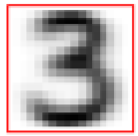
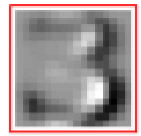
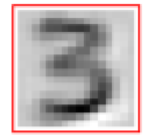
Principal Components Analysis Image Example

principal component score ($z_{i,2}$)

16x16 of loading PC #2 ($\phi_{2,k}$)

Retain the first 2 principal components:

$$\hat{f}(\lambda) = \bar{x} + \lambda_1 v_1 + \lambda_2 v_2$$

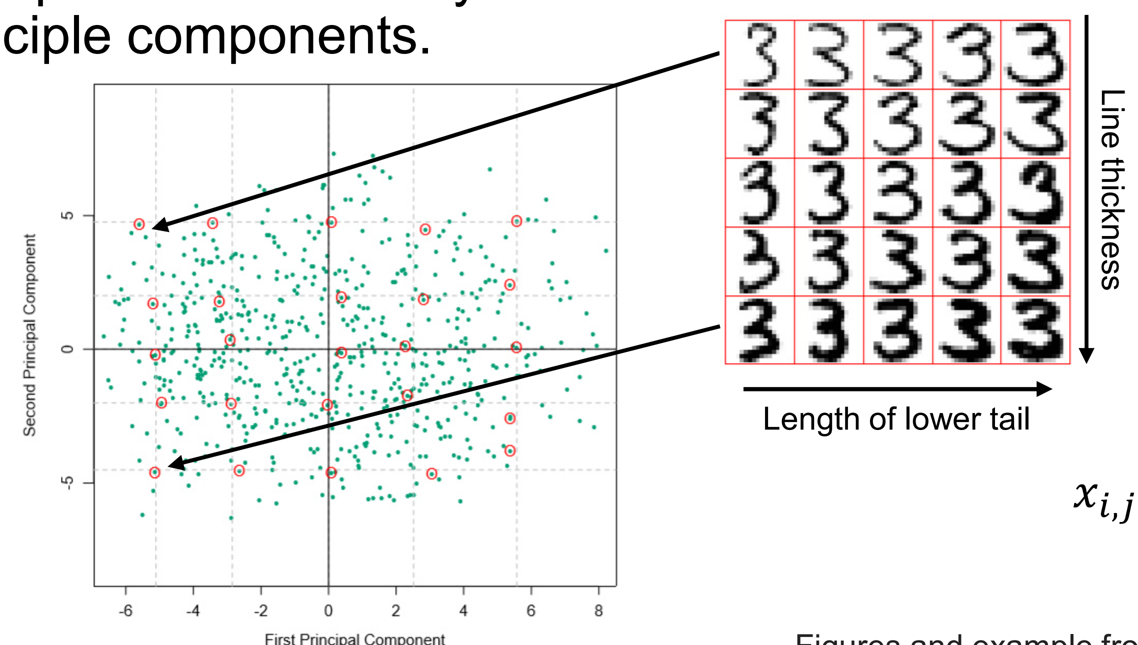
$=$  $+ \lambda_1 \cdot$  $+ \lambda_2 \cdot$  .

256x256 loadings

Comments:

- We can now explore the variability of these handwritten 3's with the first two 2 principle components.

2 dials, instead of 256 to explore the space.

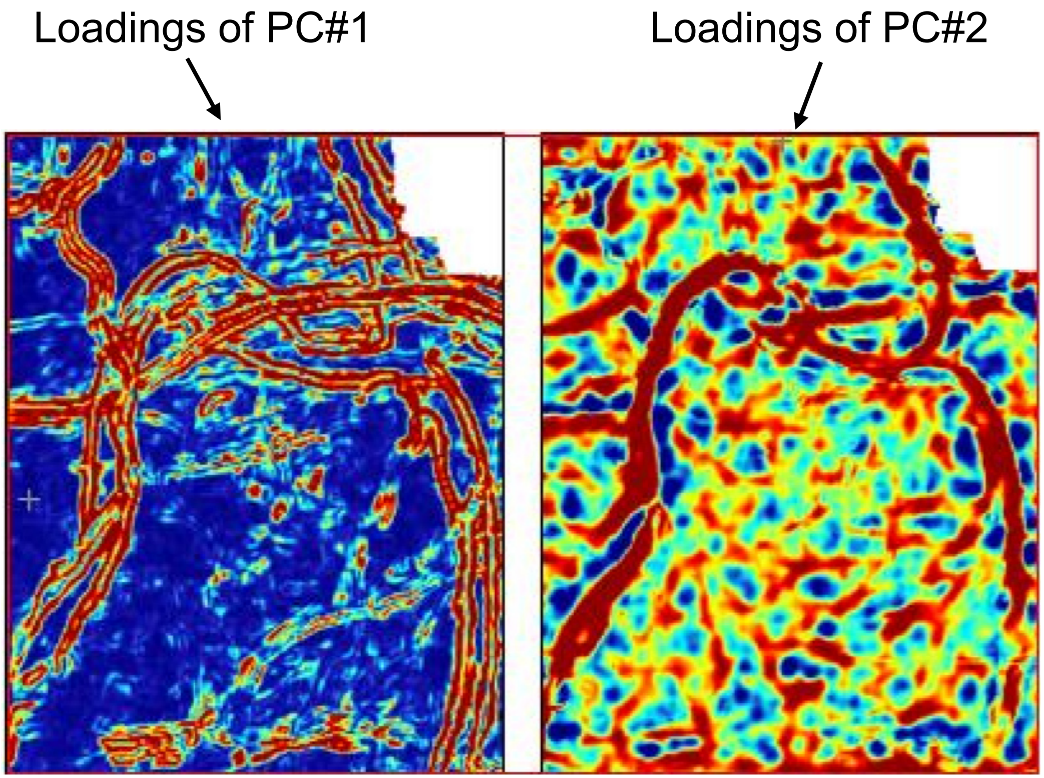


$$x_{i,j} \approx \sum_{k=1}^{p} z_{i,j} \phi_{j,k}$$

Figures and example from Hastie et al., 2009.

Principal Components Analysis Image Example

6 Seismic attributes and first 3 capture 97% of variability! Here's 2.



Each principal component describes different aspects of the multivariate seismic.

Fit models with less probability of overfit.

Figure and example from Chopra and Marfurt, 2014.

Principal Components Analysis Example

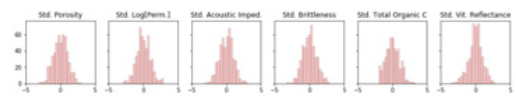
Python Demonstration:

- A well-documented Python in Jupyter Markdown html with multivariate unconventional dataset (synthetic)

Principal component analysis (PCA) is a common tool applied in machine learning workflows. It is applied widely for data analysis / exploration, dimensional reduction and directly in regression. The result of PCA is a set of orthogonal principal components and principle component scores for each data sample. These components are ordered from most variance described to least. Principal component coefficients (component loadings) reveal structures, dimensional reduction aids visualization & robust regression. Try it with a realistic dataset in a well documented **Python / Markdown Jupyter Notebook**. https://git.io/fNgRK

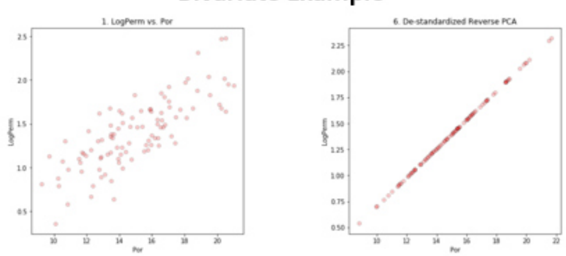
Data Checking and Cleaning

	count	mean	std	min	25%	50%	75%	max
Por	500.0	14.89936	2.985967	5.40	12.8500	14.900	17.0125	23.85
LogPerm	500.0	1.40010	0.409616	0.18	1.1475	1.380	1.6700	2.58
AI	500.0	2.99244	0.563674	1.21	2.5900	3.035	3.3725	4.70
Brittle	500.0	49.74682	15.212123	0.00	39.3125	49.595	59.2075	93.47
TOC	500.0	0.99800	0.503635	0.00	0.6400	0.960	1.3500	2.71
VR	500.0	1.99260	0.307434	0.90	1.8200	2.010	2.1725	2.84



The summary statistics and distributions look good. No obvious missing data, gaps, significant truncations, spikes or outliers. We are ready to peform principal component analysis on our 6 features.

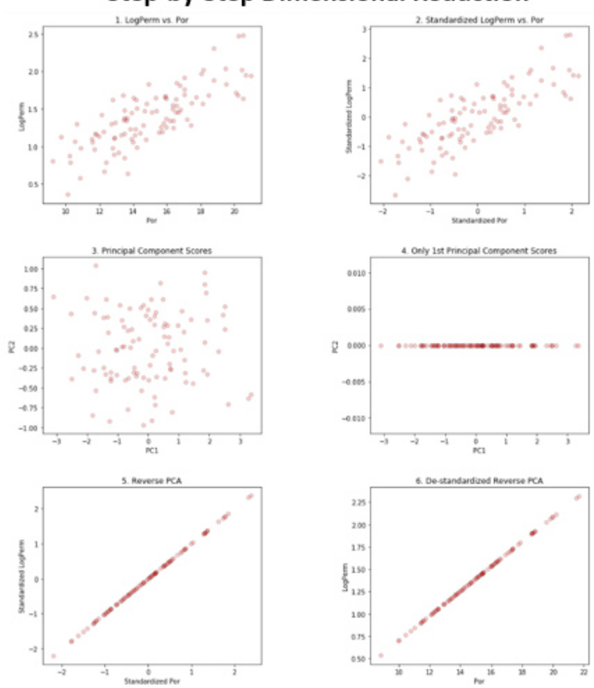
Bivariate Example



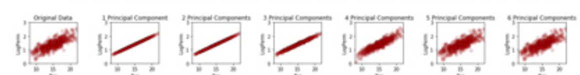

```
Variance Por = 7.89 , Variance Reduced Dimensional Por = 7.073 Fraction =  0.896
Variance LogPerm = 7.89 , Variance Reduced Dimensional LogPerm = 0.136 Fraction =  0.896
```

We got a model that explains 89.6% of the variance for both porosity and lob permeability.

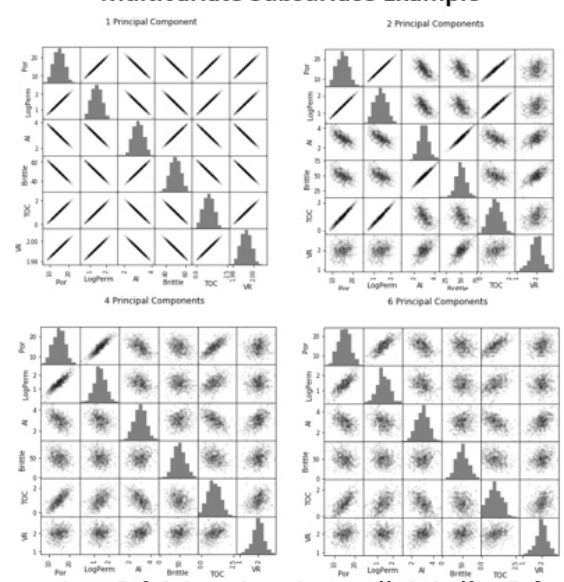
Step-by-Step Dimensional Reduction



Variable Number of Principal Components



Multivariate Subsurface Example



File **SubsurfaceDataAnalytics_PCA.ipynb** at **https://git.io/fjmRO.**

Declustering Demo

Demonstration workflow for principal compontent analysis.

Subsurface Data Analytics

Principal Component Analysis for Subsurface Data Analytics in Python

Michael Pyrcz, Associate Professor, University of Texas at Austin

Twitter | *GitHub* | *Website* | *GoogleScholar* | *Book* | *YouTube* | *LinkedIn*

PGE 383 Exercise: Principal Component Analysis for Subsurface Data Analytics in Python

Here's a simple workflow, demonstration of principal component analysis for subsurface modeling workflows. This should help you get started with building subsurface models that integrate uncertainty in the sample statistics.

Principapl Component Analysis

Principal Component Analysis one of a variety of methods for dimensional reduction:

Dimensional reduction transforms the data to a lower dimension

- Given features, $X_1, \ldots, X_m$ we would require $\binom{m}{2} = \frac{m \cdot (m-1)}{2}$ scatter plots to visualize just the two-dimensional scatter plots.
- Once we have 4 or more variables understanding our data gets very hard.
- Recall the curse of dimensionality, impact inference, modeling and visualization.

One solution, is to find a good lower dimensional, p, representation of the original dimensions m

Benefits of Working in a Reduced Dimensional Representation:

1. Data storage / Computational Time
2. Easier visualization
3. Also takes care of multicollinearity

Orthogonal Transformation

Convert a set of observations into a set of linearly uncorrelated variables known as principal components

- The number of principal components (k) available are min$(n-1, m)$
- Limited by the variables/features, m, and the number of data

Components are ordered

- First component describes the larges possible variance / accounts for as much variability as possible
- Next component describes the largest possible remaining variance
- Up to the maximum number of principal components

Eigen Values / Eigen Vectors

File SubsurfaceDataAnalytics_PCA.ipynb at https://git.io/fjmRO.

Principal Components in Reservoir Modeling

Examples of PCA in Subsurface Modeling:

- Modeling multivariate relationship while avoiding over fitting, porosity from a set of seismic attributes.

- Image analysis on seismic information, separating multiple attributes into information and noise.

- Analysis of feature grouping, redundancy

- Reducing dimensionality to support simpler workflows, e.g. bivariate, cosimulation methods

PGE 383
Dimensionality Reduction

- Curse of Dimensionality
- Dimensionality Reduction
- Principal Component Analysis

Michael Pyrcz, The University of Texas at Austin