PROFESSIONAL & CONTINUING EDUCATION

UNIVERSITY of WASHINGTON

Introduction to Machine Learning

MLEARN 510A – Lesson 10

Logistics

- Today is our last class
- Please submit your assignment (including any outstanding ones) by next Tuesday, December 15th, 2020
- Please fill End of Course Survey

Recap of Lesson 9

- Review of Probability
- Conditional Probability
- Bayes Theorem
- Application of Bayes Theorem
- Bayesian Networks
- Reasoning with BN
- Naïve Bayes

Course Outline

- 1. Introduction to Statistical Learning
- 2. Linear Regression
- 3. Classification
- 4. Model Building, Part 1
- 5. Model Building, Part 2
- 6. Resampling Methods
- 7. Linear Model Selection and Regularization
- 8. Moving Beyond Linearity
- 9. Bayesian Analysis
- **10. Dimensionality Reduction**

Outline of Lesson 10

- Dimensionality Reduction
- Issues Encountered in High Dimensional Settings
- Principal Component Analysis (PCA)
- Principal Components Regression (PCR)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)

Dimensionality Reduction

- Many modern data domains involve huge number of features
- Documents: thousands of words, millions of bigrams
- Images: thousands to millions of pixels
- Genomics: thousands of genes, millions of DNA polymorphisms
- **Dimensionality Reduction** refers to a set of techniques that reduces the dimensionality — or number of features — in your dataset

What Makes a Setting High Dimensional?

- In a *low-dimensional* setting *n*, the number of observations, is much greater than *p*, the number of features
- Data sets containing more features than observations are often referred to as *high-dimensional*
- Classical approaches such as least squares linear regression are not appropriate in this setting
- Even if the *p* is slightly smaller than *n,* considerations of the high dimensional setting still apply

Examples of High Dimensional Data

Multilingual News Stories

Brain Imaging Data (100s of MBs per scan)

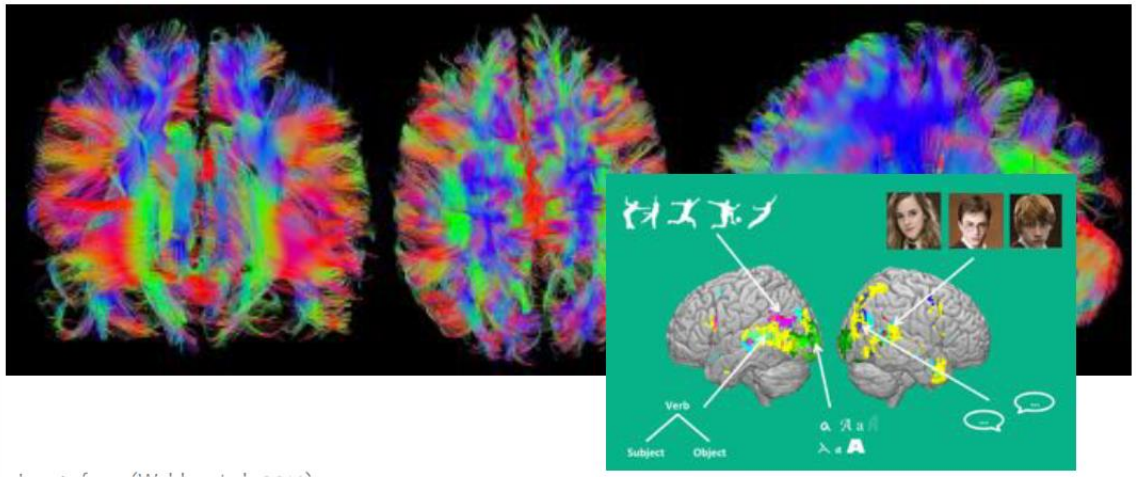


Image from https://pixabay.com/en/brain-mrt-magnetic-resonance-imaging-1728449/

What Goes Wrong in High Dimensions?

- It is important to understand what might go wrong with various analysis techniques in high dimensional settings
- Our earlier statistical algorithms may degrade once the number of features exceeds the number of observations
- Major issues in a high dimensional setting are
  - Overfitting
  - Curse of dimensionality
  - Issues with interpretation
  - Issues with Visualization

Overfitting in High Dimensions

- Take linear regression as an example (same arguments apply to logistic regression, LDA and other statistical approaches)
- When $p > n$, least squares techniques should not be applied
- Reason: regardless of the relationship between the features and the response, least squares will yield coefficient estimates that result in a perfect fit to the data, such that the residuals are zero
- When $p > n$ or $p \approx n$, a simple least squares regression line is too *flexible* and hence overfits the data

What Goes Wrong in High Dimensions?

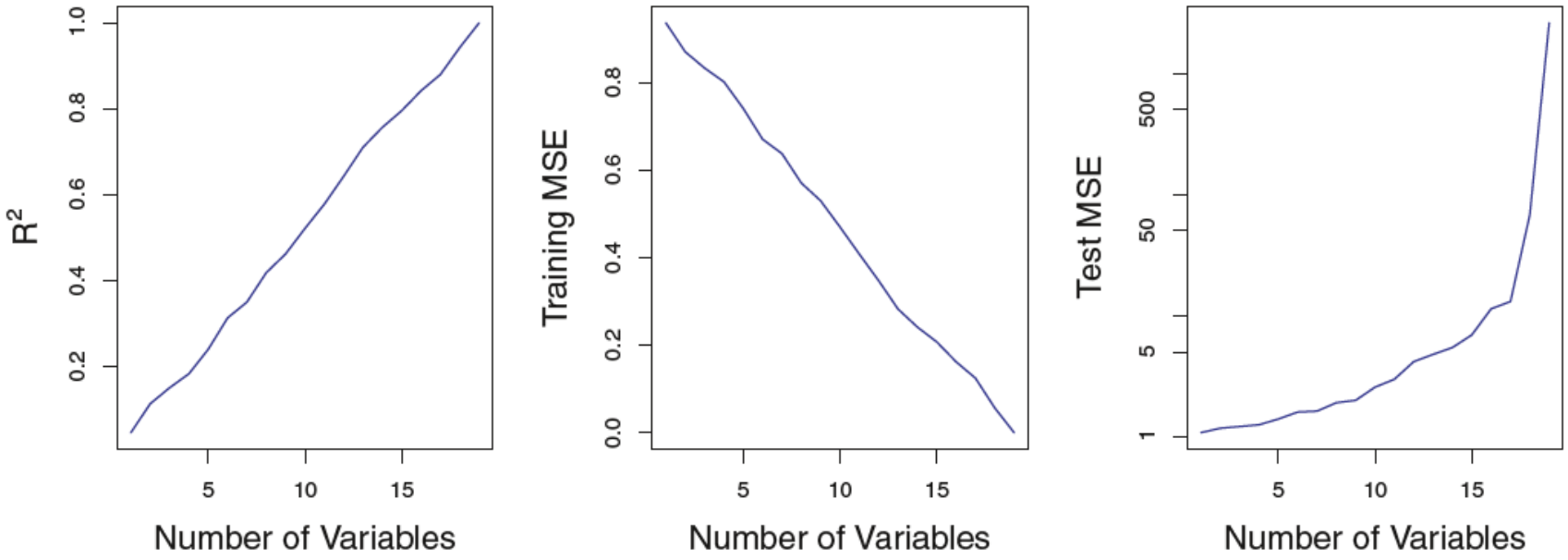
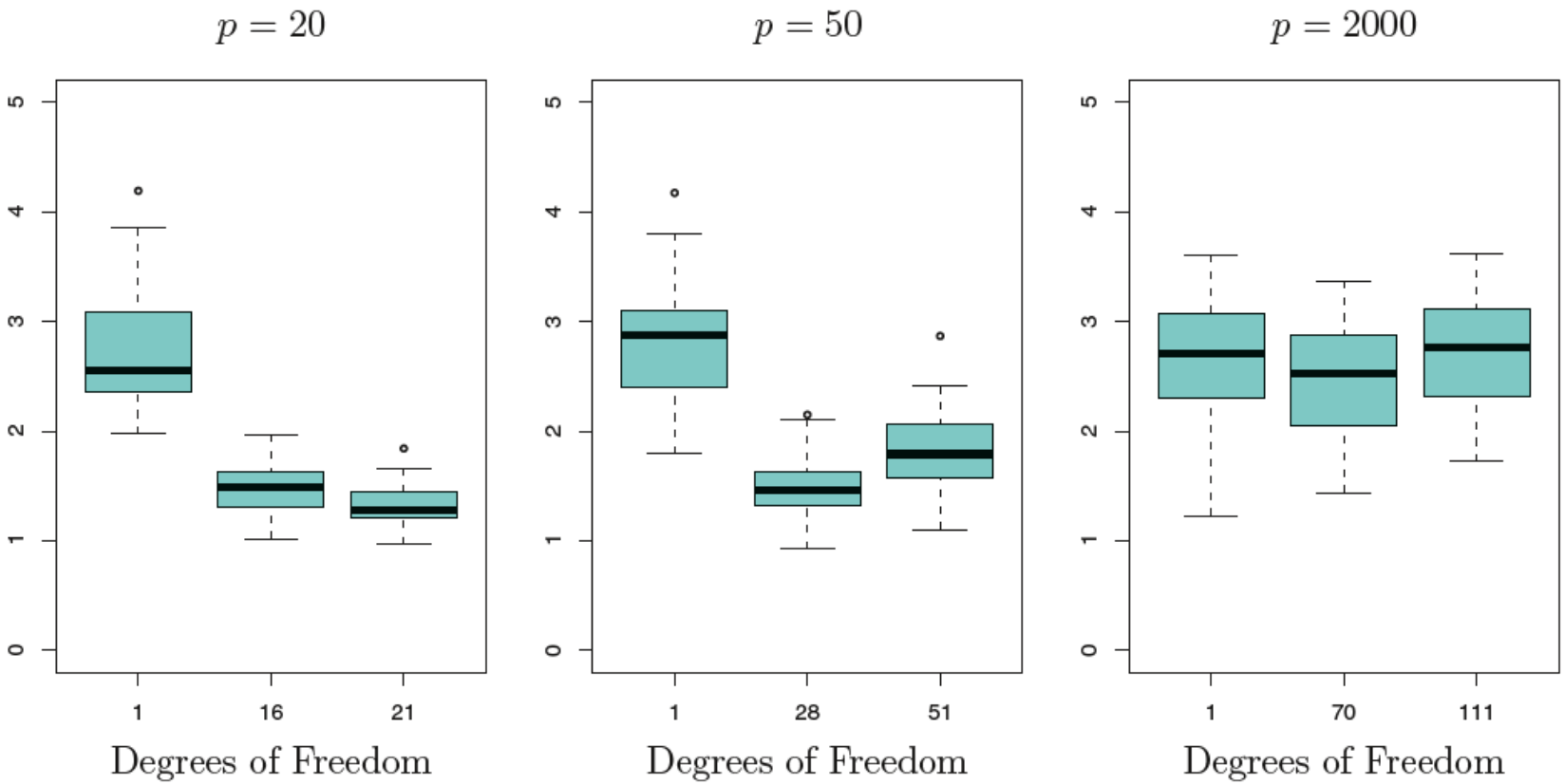


FIGURE 6.23. *On a simulated example with $n = 20$ training observations, features that are completely unrelated to the outcome are added to the model.* Left: *The R^2 increases to 1 as more features are included.* Center: *The training set MSE decreases to 0 as more features are included.* Right: *The test set MSE increases as more features are included.*

Curse of Dimensionality

- The curse of dimensionality arises in many places in computer science, and especially so in machine learning
- One might think that as the number of features used to fit a model increases, the quality of the fitted model will increase as well
- Collection thousands or millions of features may or may not lead to improvements depending on whether these features are relevant
- Even if they are relevant, the variance incurred in fitting their coefficients may outweigh the reduction in bias that they bring

Curse of Dimensionality



Another Interpretation of Curse of Dimensionality

- As the number of features increase, the number of regions that correspond to distinct values of those features increase
- We need training examples in each of these regions to make predictions on new data

Another Interpretation of Curse of Dimensionality

- In one dimensional setting (left), we need 10 regions. For 2 and 3 dimensions, there are 100 and 1000 regions, respectively requiring those many examples

- For *d* dimensions and *v* values to be distinguished along each axis we seem to need $O(v^d)$ regions and examples

Issues with Interpretation

- When $p > n$, it is easy to obtain a useless model that has zero residuals
- *Never* use sum of squared errors, p-values, R^2 statistics, or other traditional measures of model fit on the training data as evidence of a good model fit in the high-dimensional setting
- One can easily obtain a model with $R^2 = 1$ when $p > n$
- *Report results on a cross validated sample instead*

Issues with Visualization

- Suppose that we wish to visualize n observations with measurements on a set of p features, $X_1, X_2, \ldots, X_p$
- We could do examine two-dimensional scatterplots of the data. However, there are $p(p-1)/2$ such scatterplots. If p is large, then it will certainly not be possible to look at all of them
- Need a better method for visualization when p is large
- Ideally, we would like to find a low-dimensional representation of the data that captures as much of the information as possible
- We could then plot this low-dimensional (2 or 3 dimensions) representation

Approaches to Dimensionality Reduction

- Feature selection
  - Select subset of existing features (without modification)
- Model regularization
  - Lasso
  - Ridge
- Combine (map) existing features into smaller number of new features
  - Linear combination (projection) – Principal Component Analysis
  - Nonlinear combination

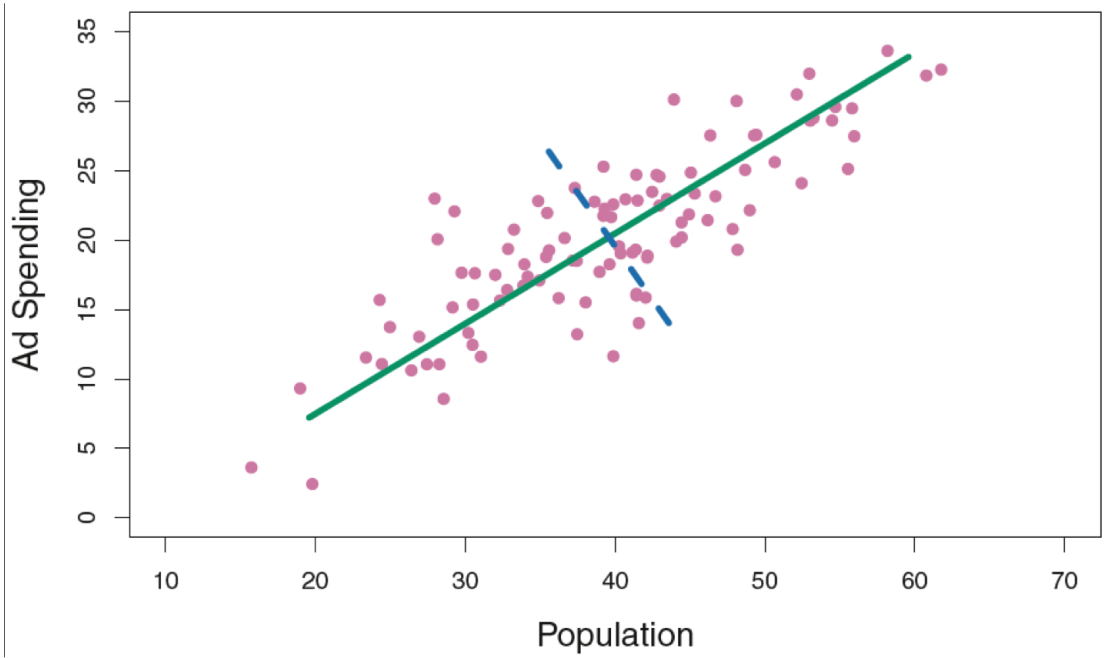
Principal Component Analysis (PCA)

- Produces a low dimensional representation of a dataset
- The first principal component represents linear combination of variables with the largest variance
- The second principal component has the largest variance, subject to being uncorrelated to the first one
- Keep going
- A large number of correlated variables can be replaced with a much smaller set of principal components

PCA – Intuition

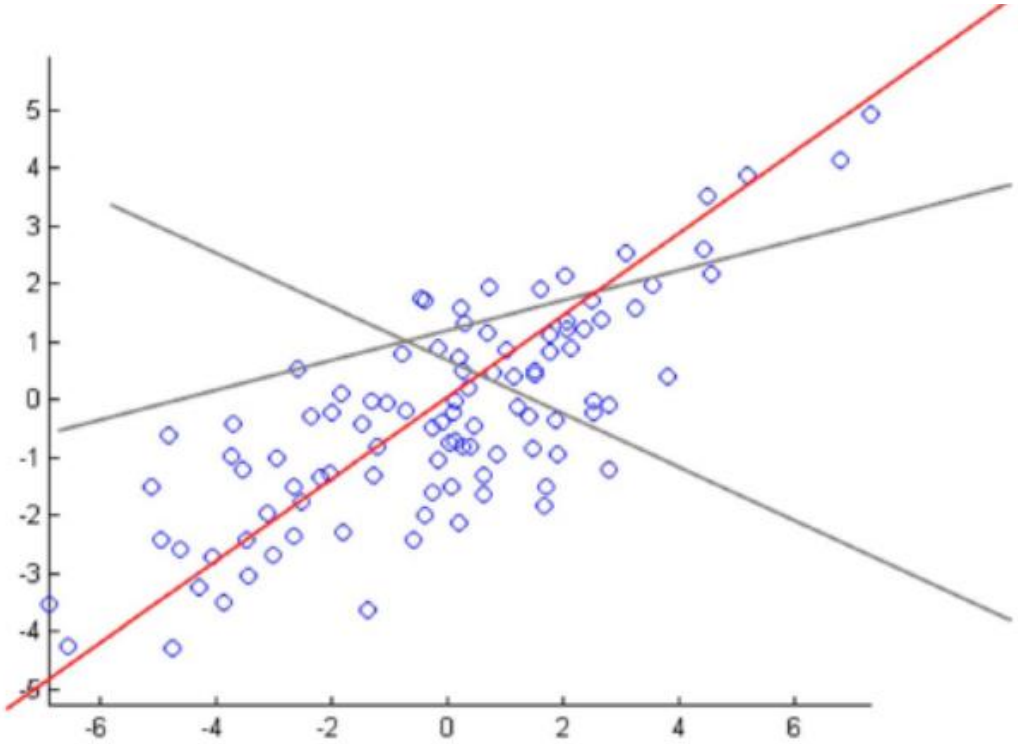
- First PC is the projection direction that maximizes the variance of the projected data

- Second PC is the projection direction that is orthogonal to the first PC and maximizes variance of the projected data



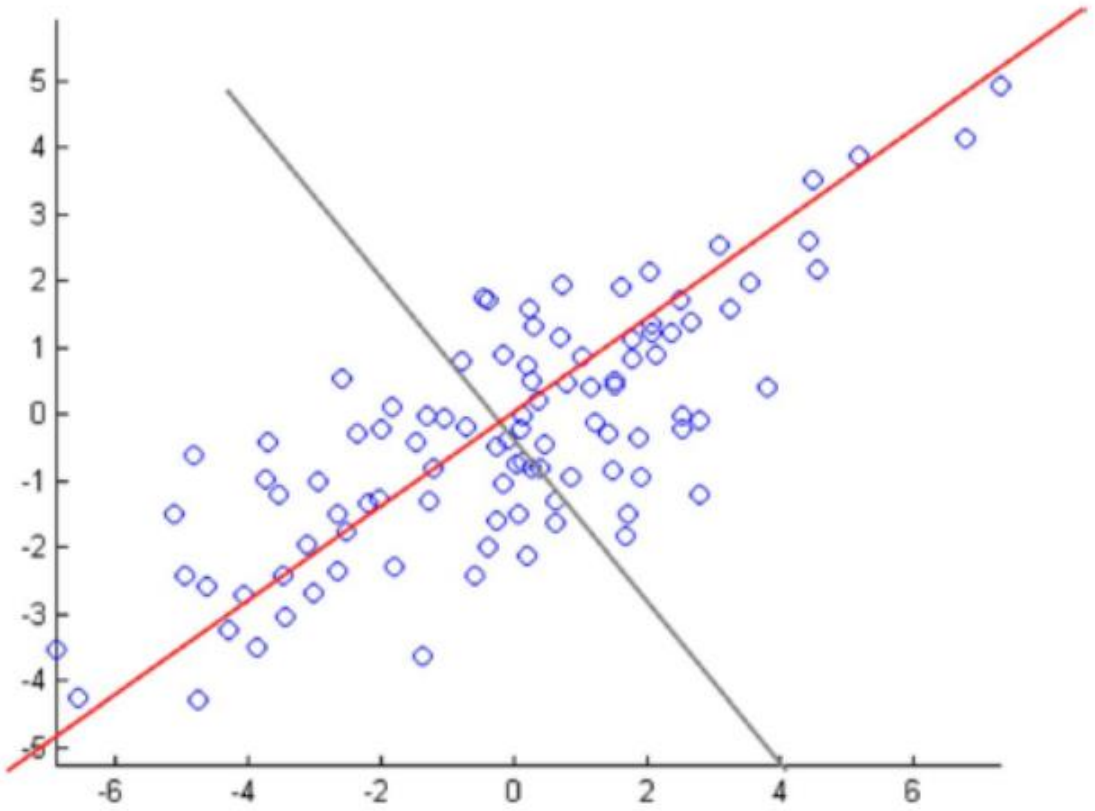
PCA – Conceptual Algorithm

- Find a line, such that when the data is projected onto that line, it has the maximum variance



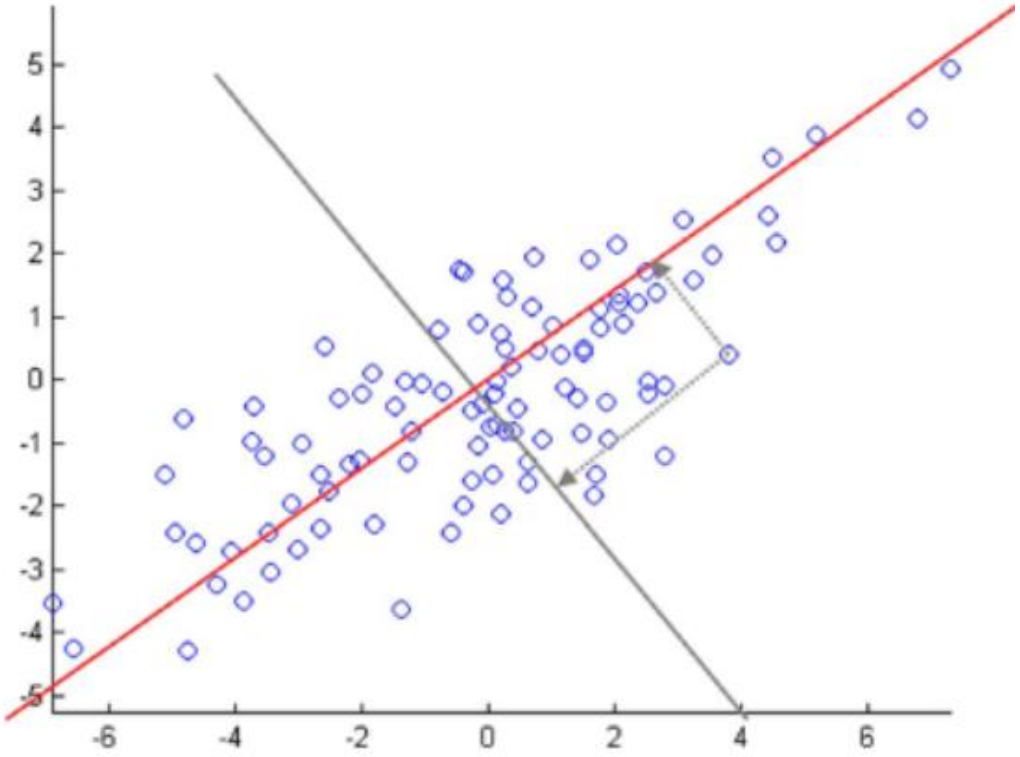
PCA – Conceptual Algorithm

- Find a second line, orthogonal to the first, such that when the data is projected onto that line, it has the maximum variance



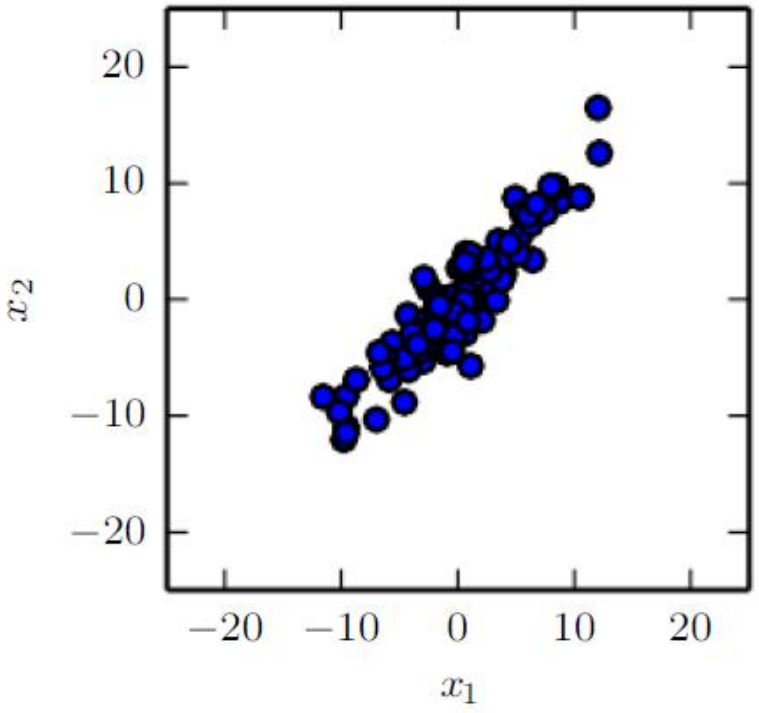
PCA – Conceptual Algorithm

- Repeat until have *k* orthogonal lines
- The projected position of a point on these lines gives the coordinates in the *k*-dimensional reduced space



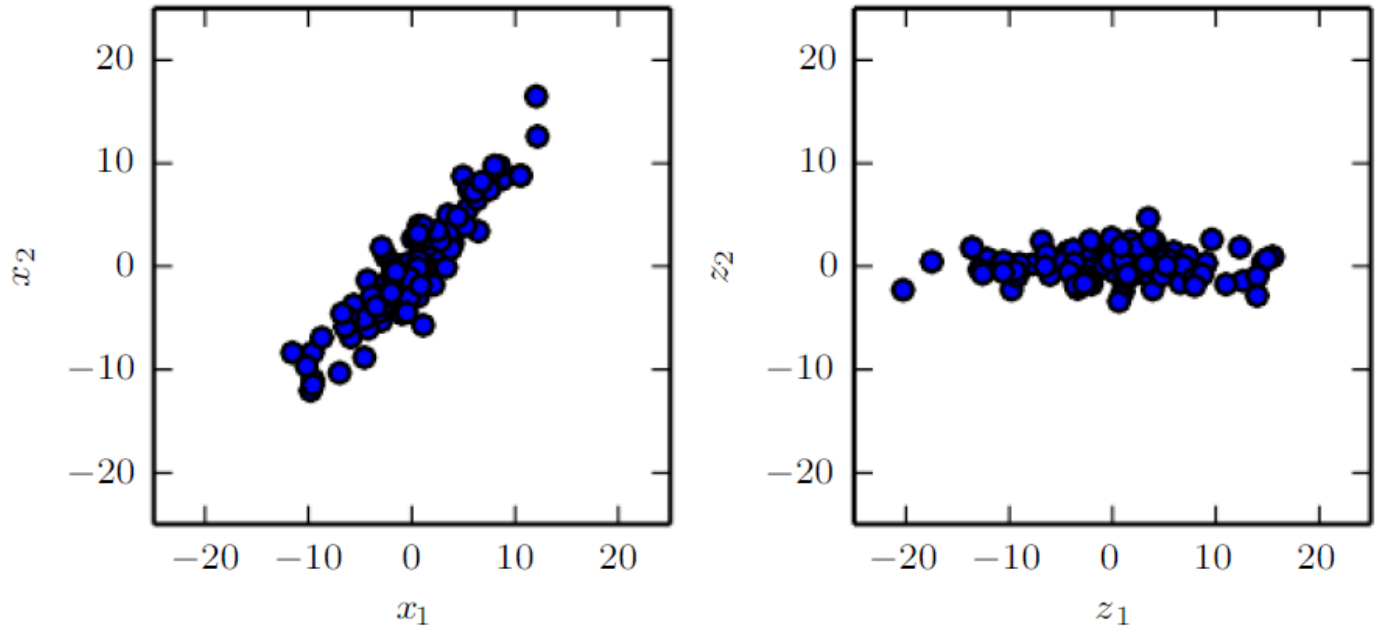
Quiz

- What will the new axes look like in terms of PCA components for the following data?



Quiz

- What will the new axes look like in terms of PCA components for the following data?



Goal of PCA

- As we keep adding principal components, we capture more and more variation of our data
- **It is hoped, in general, that most of the variation in X will be accounted for by *k* PC's where *k* << *p***
- This also suggests a stopping criterion. We can stop adding components once z% of the variation in **X** is accounted for

PCA – Details

- Let's consider a *nxp* dataset **X**, where each of the *p* features has been centered to have mean of 0

- We look for the linear combination:

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip}$$

- That has the largest variance, subject to constraint:

$$\sum_{j=1}^{p} \phi_{j1}^2 = 1$$

- This can be rewritten as this optimization problem:

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Quiz

- Why do we need the constraint?

$$\underset{\phi_{11},\ldots,\phi_{p1}}{\text{maximize}} \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j1} x_{ij} \right)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \phi_{j1}^2 = 1$$

Solving the Optimization Problem – Eigen Decomposition

- Eigen decomposition operates on the covariance matrix of data **X**
- Calculate covariance matrix **C**
- Perform eigen decomposition $\mathbf{C} = \mathbf{VLV}^T$
  - **V** is the matrix of eigenvectors and **L** is a diagonal matrix of eigenvalues sorted in decreasing order
- Eigenvectors are called *principal axes* or *principal directions*
- Projections of the data on the principal axes are called *principal components*, also known as *PC scores*; these are the new, transformed variables
- The j^{th} principal component is given by the j^{th} column of **XV**
- The coordinates of the i^{th} data point in the new PC space are given by the i^{th} row of **XV**

Solving the Optimization Problem – Singular Value Decomposition

- Operates on the actual data **X,** not its covariance matrix
- Involves decomposing **X** into its SVD $\mathbf{X = USV^T}$
- **V** are the right singular vectors
- Principal components are given by $\mathbf{XV = USV^TV = US}$
- Needs data **X** to be centered, otherwise **C** is not equal to $\mathbf{X^TX}$
- Generally, SVD is preferred due to numerical stability

Visualizing Data in Lower Dimensions

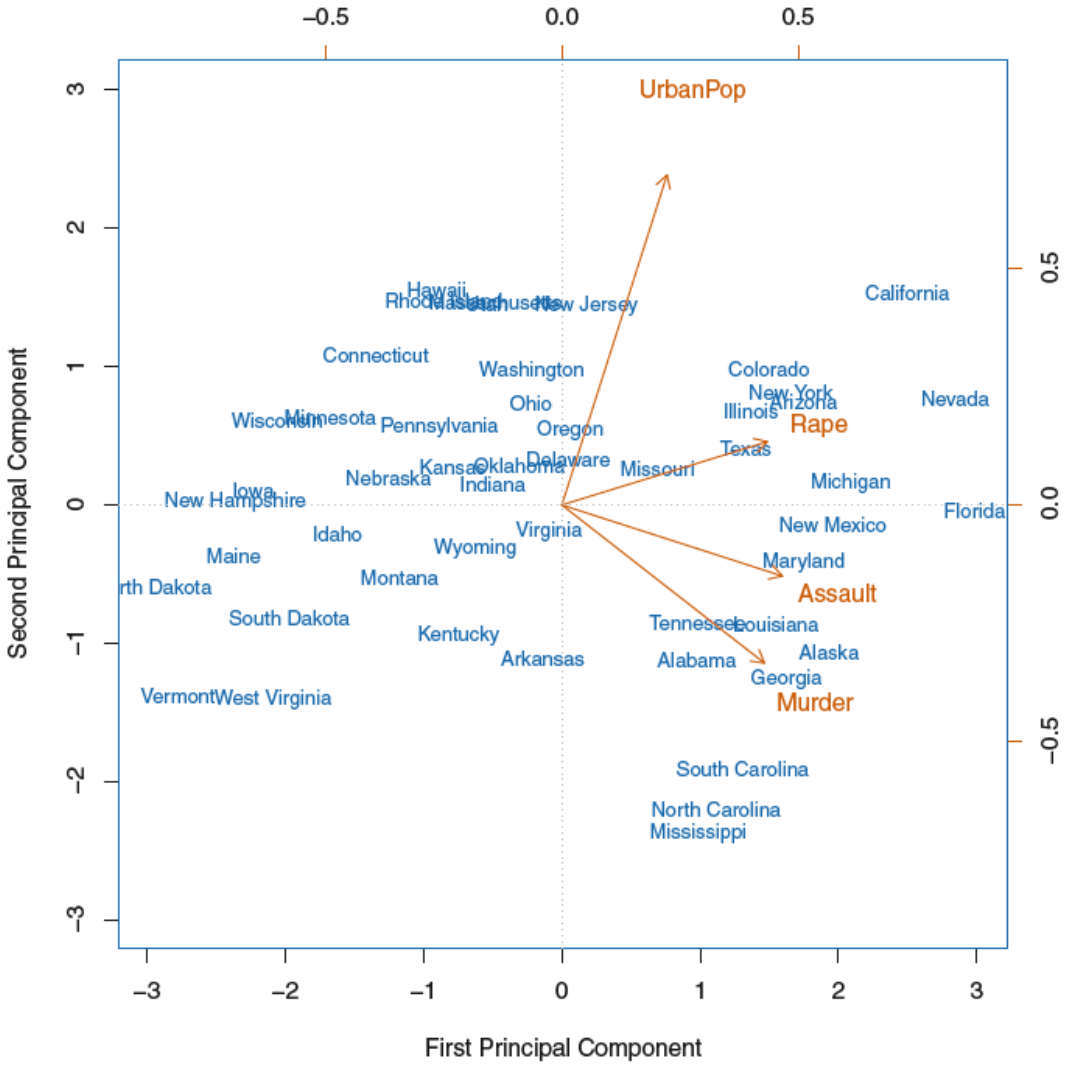
- We can plot and visualize high dimensional by projecting it onto (much fewer) principal components

- Generally, 2 or 3 principal components are used for visualization. More than 3 will be used depending on how much variation they capture

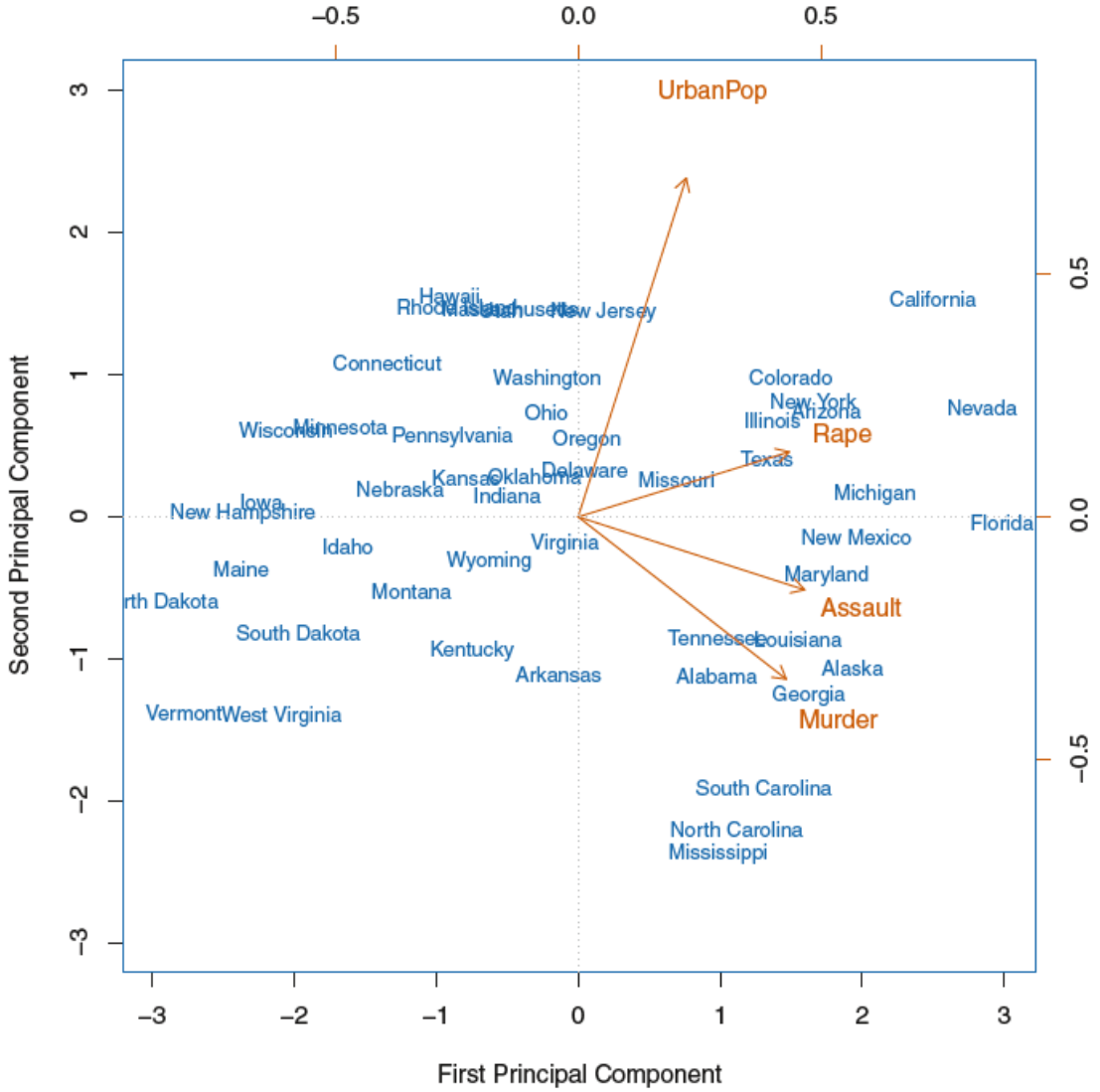
- We can plot the score vector Z_1 against Z_2, Z_1 against Z_3, Z_2 against Z_3, and so forth

PCA Example: USArrests Dataset

- Biplot is often used to p PCA results
- It shows both principal components and the loadings on those components
- Bottom axis: PC1 score
- Left axis: PC2 score.
- Top axis: loadings on PC1.
- Right axis: loadings on PC2



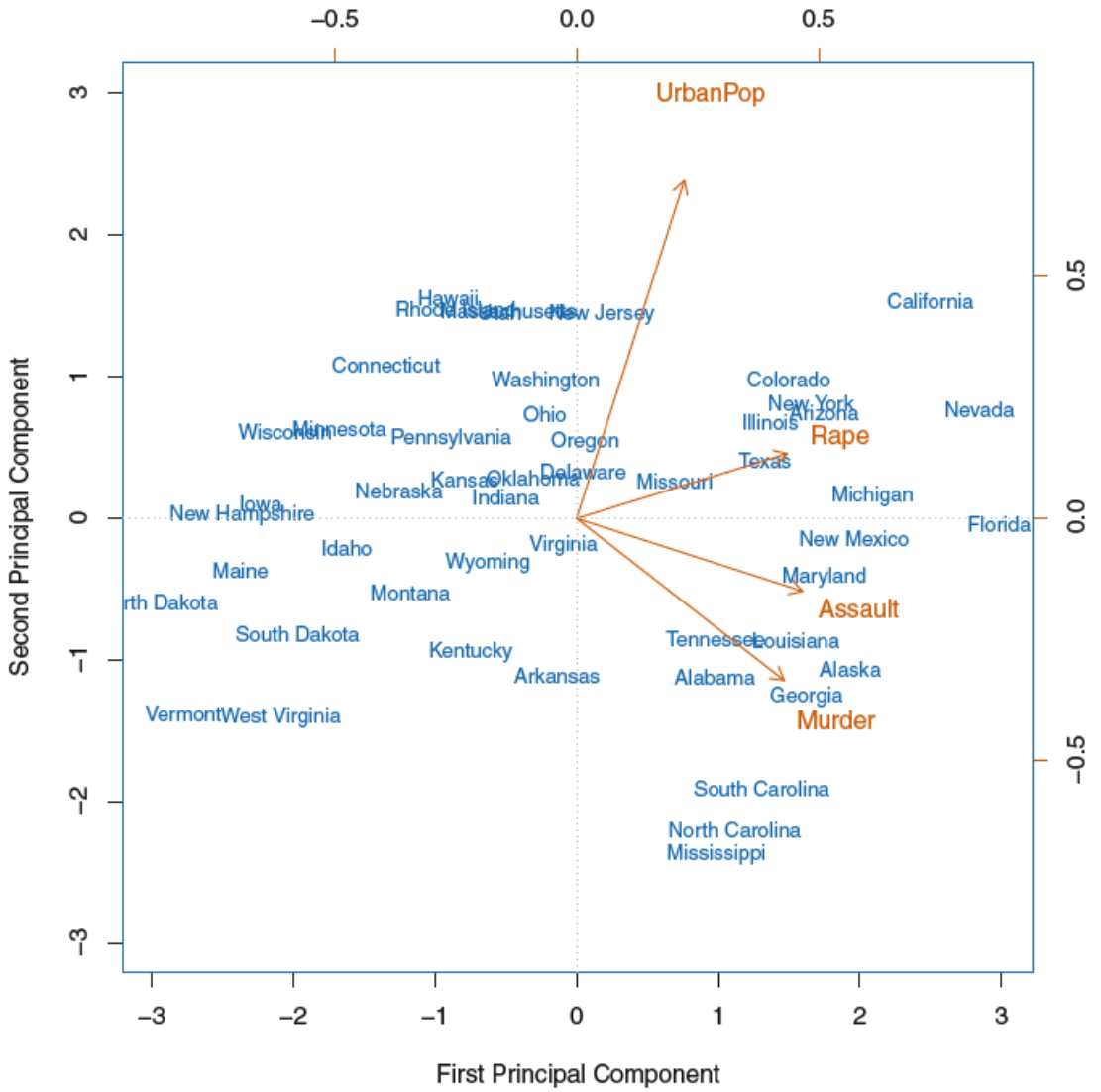
PCA Example: USArrests Dataset



PC Loadings

	PC1	PC2
Murder	0.5358995	−0.4181809
Assault	0.5831836	−0.1879856
UrbanPop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

Quiz

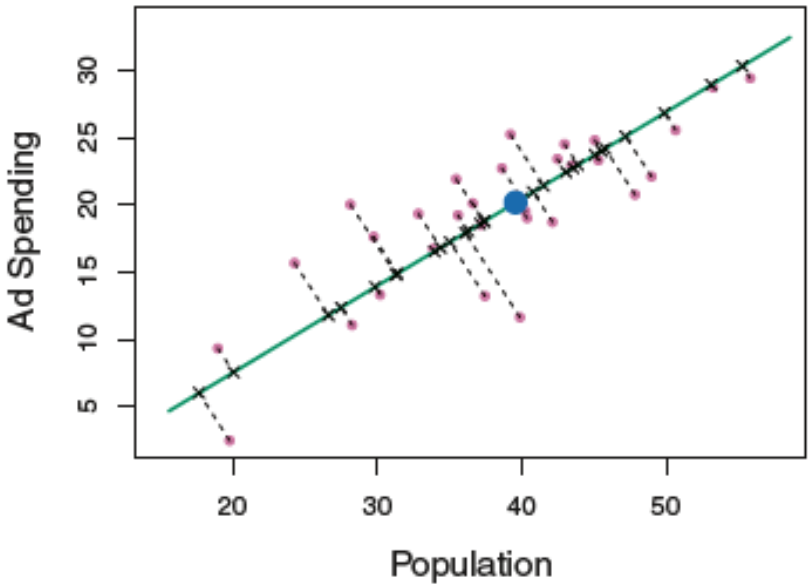


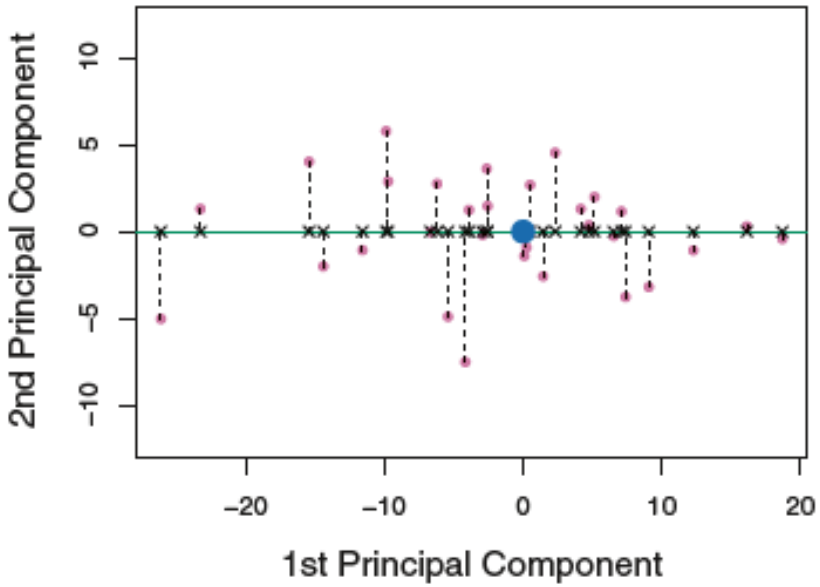
1. Which states have the high crime rates?
2. Which states have low crime rates?
3. Which states is high urbanized?
4. What can we conclude about Indiana and Virginia?

Alternative Interpretation of PCs

- They provide low-dimensional linear surfaces that are *closest* to the observations

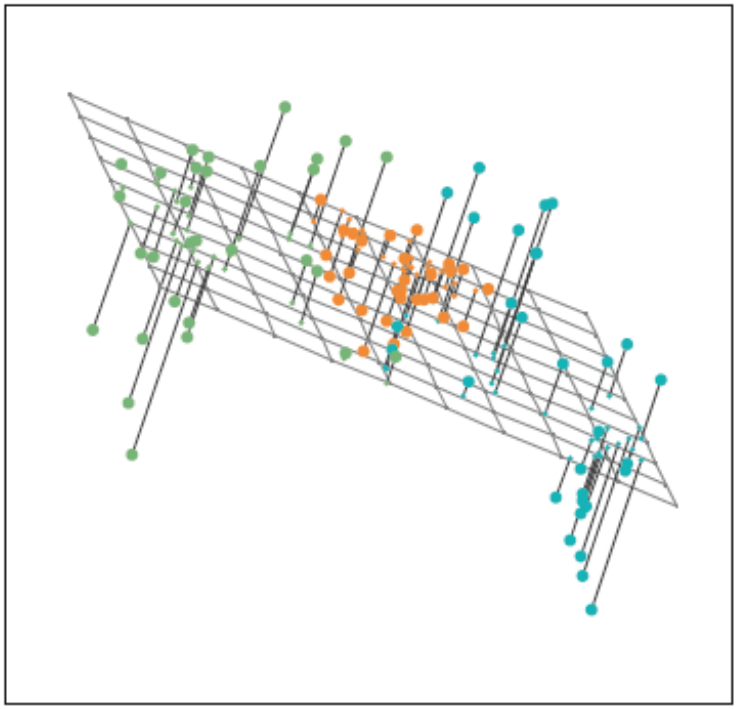
- The first principal component loading vector has a very special property: it is the line in *p*-dimensional space that is *closest* to the *n* observations

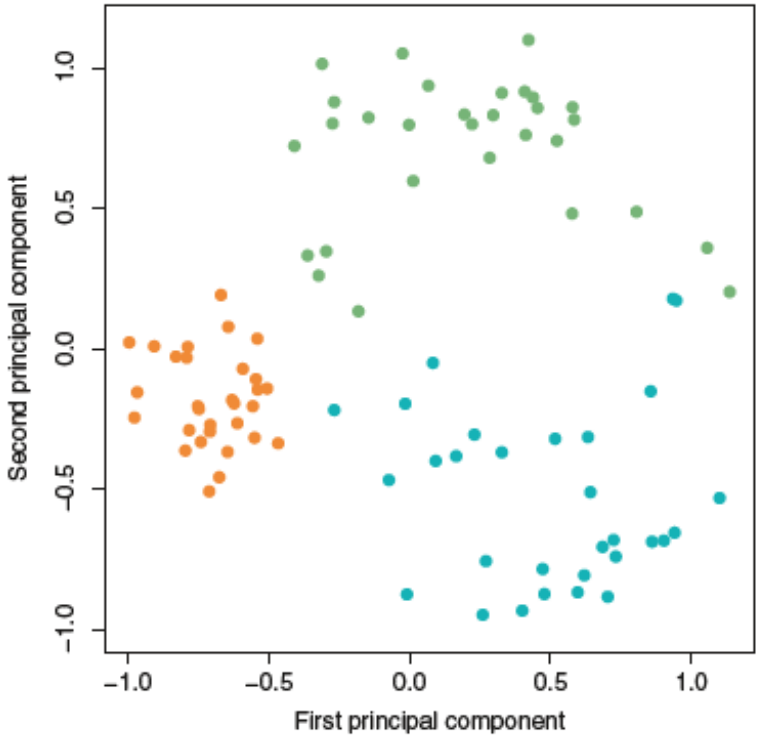




Alternative Interpretation of PCs

- The first two principal components of a data set span the plane that is closest to the *n* observations, in terms of average squared Euclidean distance





Alternative Interpretation of PCs

- Together the *M* principal component score vectors and *M* principal component loading vectors can give a good approximation to the data when *M* is sufficiently large

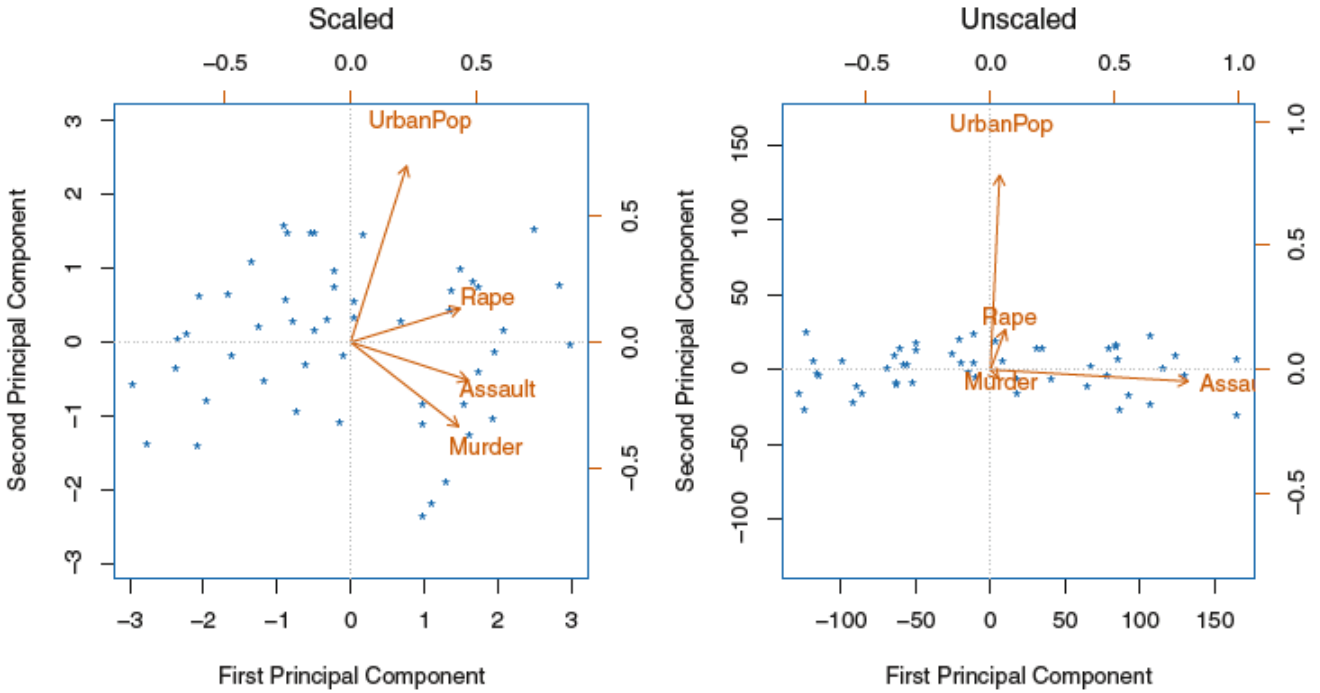
$$x_{ij} \approx \sum_{m=1}^{M} z_{im}\phi_{jm}$$

- When $M = \min(n - 1, p)$, then the representation is exact:

$$x_{ij} = \sum_{m=1}^{M} z_{im}\phi_{jm}.$$

Importance of Scaling the Variables

- Perform centering and scaling
- The results obtained when we perform PCA will also depend on whether the variables have been individually scaled
- Variance for Assault = 6945.1, Variance for Murder 18.9



Proportion of Variance Explained

- How much of the information is lost by projecting the observations onto the first few principal components?

- The *total variance* present in a data set (assuming that the variables have been centered to have mean zero) is

$$\sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2,$$

- Variance explained by the *m*th principal component is

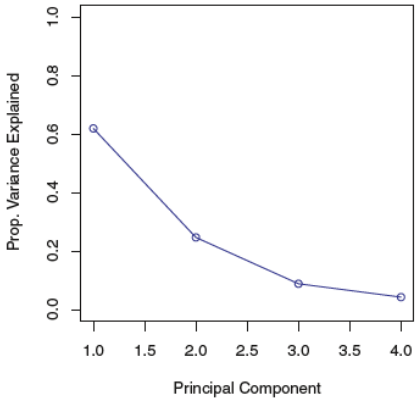
$$\frac{1}{n} \sum_{i=1}^{n} z_{im}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{jm} x_{ij} \right)^2 .$$

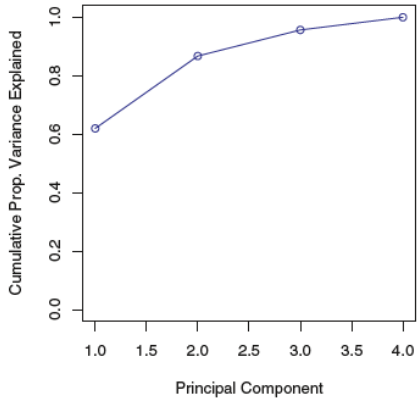
Proportion of Variance Explained

- Proportion of Variance Explained (PVE) is given by

$$\frac{\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{jm}x_{ij}\right)^2}{\sum_{j=1}^{p}\sum_{i=1}^{n}x_{ij}^2}.$$

- The PVE of each principal component is a positive quantity



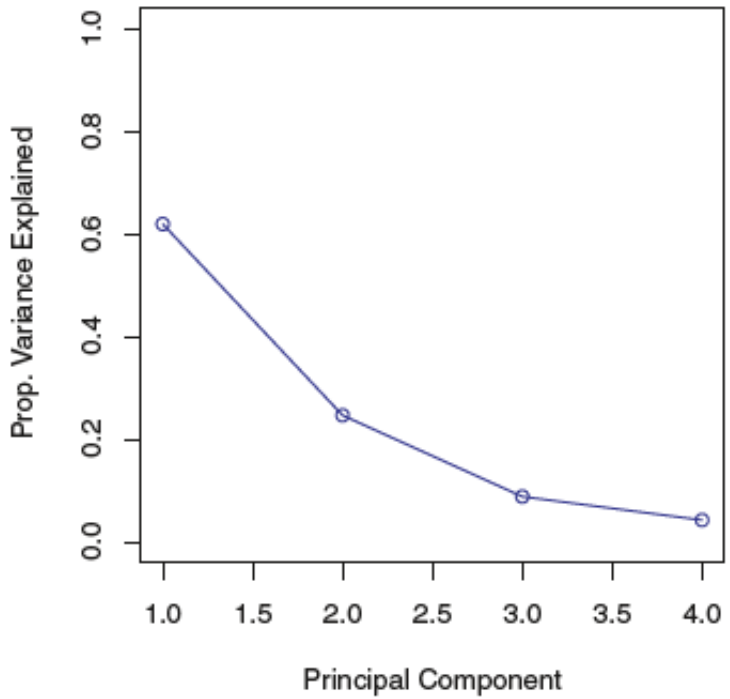


- Compute the cumulative PVE of the first *M* principal components by adding PVE of all *M* components

- PVEs sum to one

How Many PCs to Use?

- No simple answer to this question
- Look at ‘elbow’ in the *scree* plot
- Choose the smallest number of PCs that explain a sizable amount of variation in the data
- Choose the number of PCs that allow data visualization
- Unfortunately, choosing the number of PCs is qualitative



Principal Components Regression (PCR)

- Involves constructing principal components regression the first M principal components, $Z_1, \ldots, Z_M$, and then using these components as the predictors in a linear regression model that is fit using least squares

- The key idea is that often a small number of principal components suffice to explain most of the variability in the data, as well as the relationship with the response

- While this assumption is not guaranteed to be true, it often turns out to be a reasonable enough approximation to give good results

PCR vs. Ridge vs. Lasso

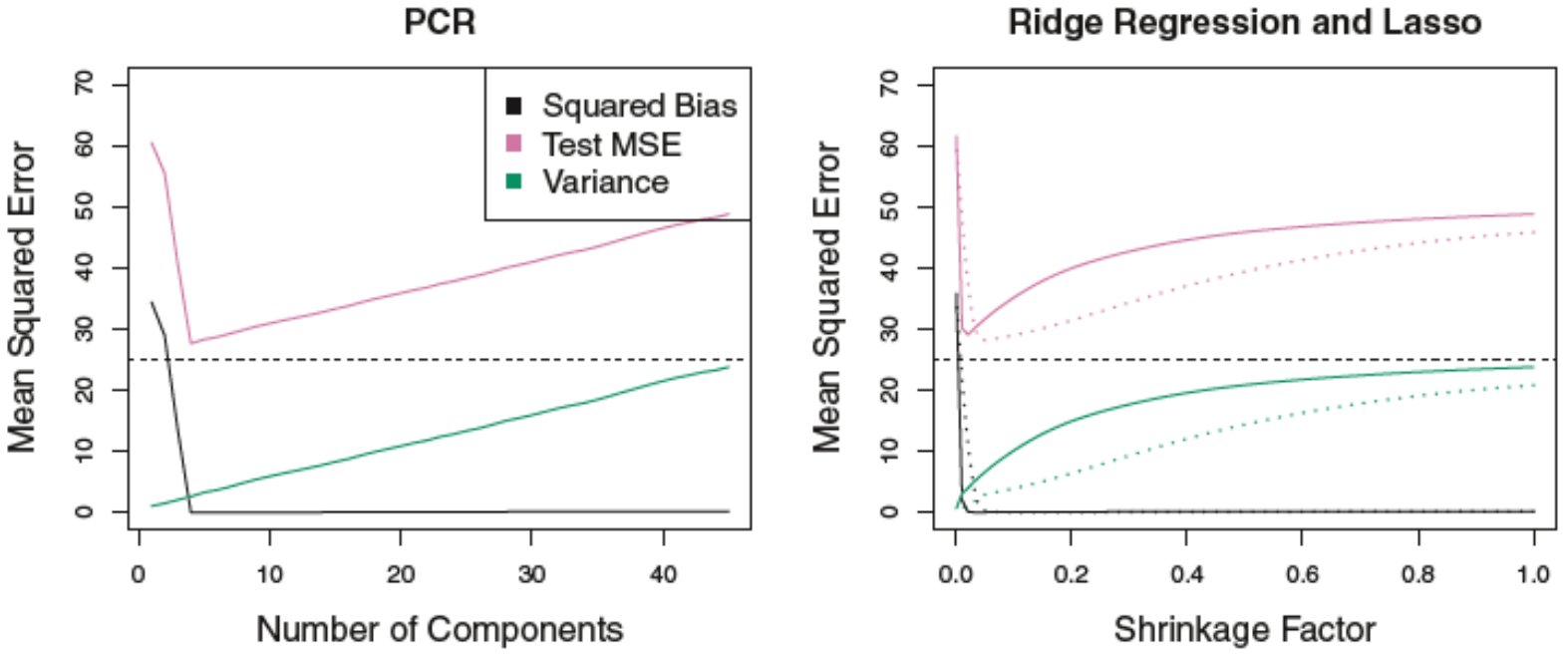
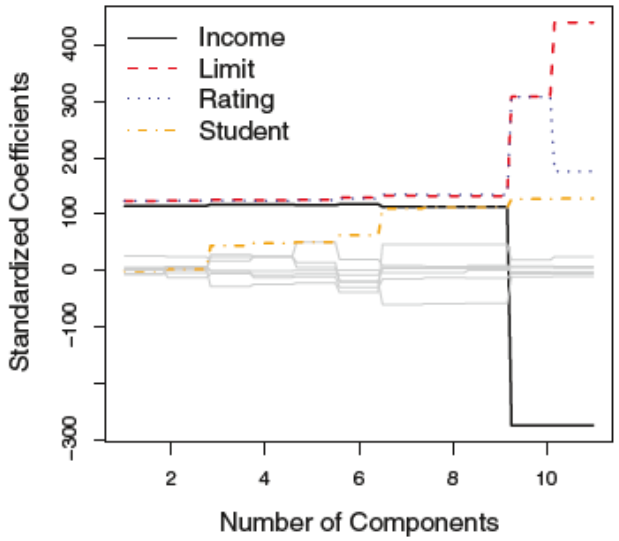
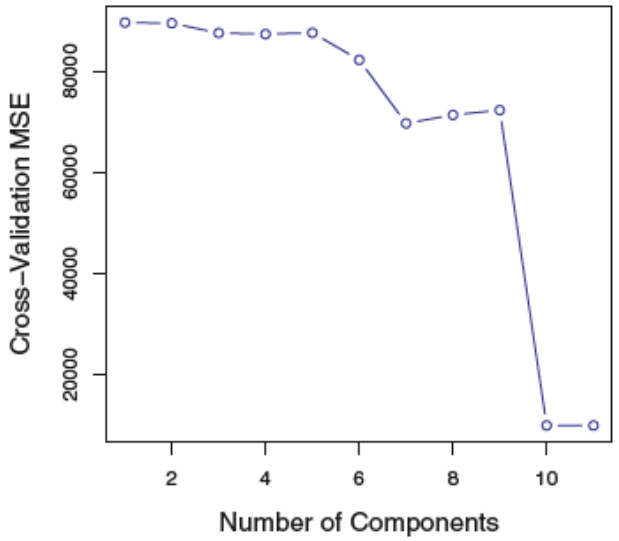


FIGURE 6.19. *PCR, ridge regression, and the lasso were applied to a simulated data set in which the first five principal components of X contain all the information about the response Y. In each panel, the irreducible error $Var(\epsilon)$ is shown as a horizontal dashed line.* Left: *Results for PCR.* Right: *Results for lasso (solid) and ridge regression (dotted). The x-axis displays the shrinkage factor of the coefficient estimates, defined as the ℓ_2 norm of the shrunken coefficient estimates divided by the ℓ_2 norm of the least squares estimate.*

Selecting *M* in PCR

- Selecting the number of principal components in PCR is well-defined (unlike selecting *M* for PCA)
- Use loss metrics on response Y to see how many PCs suffice
- Use cross-validation to find *M*





Standardizing Variables for PCR

- It is recommended to *standardize* each predictor prior to generating the principal components
- Standardization ensures that all variables are on the same scale
- In the absence of standardization, the high-variance variables will tend to play a larger role in the principal components obtained
- If the variables are all measured in the same units (say, kilograms, or inches), then there is no need to standardize them

Limitations of PCA

- Pros: Fast, simple to use and intuitive
- Relies on finding orthogonal components which are *linear* combinations of the features
- Loses interpretability. The final PCs are functions of features, not the features themselves
- PCR does not define PCs based on response (Partial Least Squares overcomes this limitation)

Bonus Content: t-Distributed Stochastic Neighbor Embedding (t-SNE)

- t-SNE, unlike PCA, is not a linear projection
- Uses the **local relationships** between points to create a low-dimensional mapping. This allows it to capture **non-linear structure**
- t-SNE creates a **probability distribution** using the **Gaussian** distribution that defines the relationships between the points in high-dimensional space

Bonus Content: t-Distributed Stochastic Neighbor Embedding (t-SNE)

- t-SNE uses the **Student t-distribution** to **recreate** the probability distribution in low-dimensional space

- This prevents the **crowding problem**, where points tend to get crowded in low-dimensional space due to the **curse of dimensionality**

- t-SNE optimizes the embeddings directly using gradient descent. The cost function is **non-convex** though, meaning there is the risk of getting stuck in local minima

- t-SNE uses multiple tricks to try to avoid this problem

How Does t-SNE Work?

- At a high level, t-SNE works in two steps
- Step 1: In the high-dimensional space, create a **probability distribution** that dictates the relationships between various neighboring points
- Step 2: **Recreate** a low dimensional space that follows that probability distribution as best as possible
- The "t" in t-SNE comes from the t-distribution, which is the distribution used in Step 2. The "S" and "N" ("stochastic" and "neighbor") come from the fact that it uses a probability distribution across neighboring points

t-SNE Transformation on New Data

- Once PCs are obtained, new incoming data can be projected on these principal components
- However, t-SNE cannot be used for new points. It merely fits the training data
- t-SNE has no parameters, and directly optimizes the embeddings
- We can train a t-SNE model on some data, but we cannot use it to reduce the dimensionality of some new data points

Limitations of t-SNE

- Stochasticity of final solution: PCA is deterministic, t-SNE is not
- Interpretability of mapping: t-SNE map only local / neighbours correctly so our insights from embeddings must be cautiously presented. Global trends are not accurately represented
- t-SNE cannot transform new data on a coordinate system like PCA
- Computationally expensive
- Requires tuning of the *perplexity* parameter

Resources

- t-SNE explained
  https://mlexplained.com/2018/09/14/paper-dissected-visualizing-data-using-t-sne-explained/

- How to use t-SNE effectively
  https://distill.pub/2016/misread-tsne/

- Look for mathematical details of deriving PCA

Jupyter Notebook

- *Case Study*

Final Thoughts

- Hopefully, you had fun learning to learn!
- Get your hands dirty with EDA and model development
- Follow AI/Engineering blogs to learn how production ML systems are built
  Amazon: https://www.amazon.science/blog
  Google: https://ai.googleblog.com/
  Uber: https://eng.uber.com/
- You can reach me at aitzaz@uw.edu

ON-BRAND STATEMENT

FOR GENERAL USE

> What defines the students and faculty of the University of Washington? Above all, it's our belief in possibility and our unshakable optimism. It's a connection to others, both near and far. It's a hunger that pushes us to tackle challenges and pursue progress. It's the conviction that together we can create a world of good. And it's our determination to Be Boundless. Join the journey at **uw.edu**.