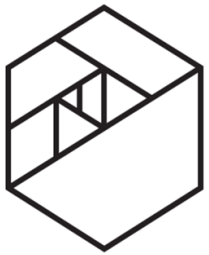


Dimensionality Reduction and Regularization

INTRODUCTION TO DATA SCIENCE – FALL 2018

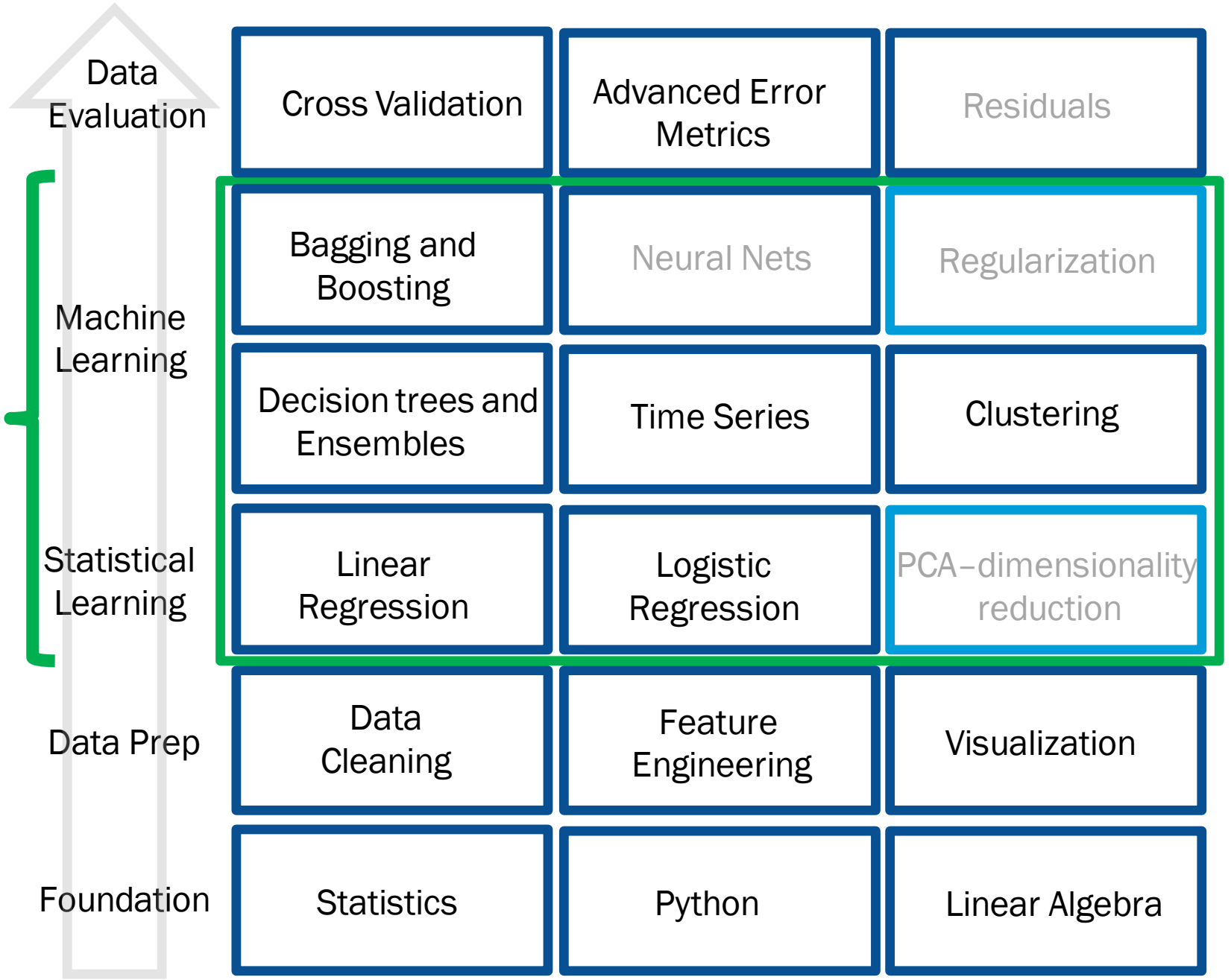
SESSION 10

AGENDA

1. Introduction
2. Principal components analysis
3. Principal components regression
4. Regularization
5. Lasso and Ridge

Introduction to Data Science

- Learning the steps in the Data Science Process
- Learning multiple model methodologies



Too many features

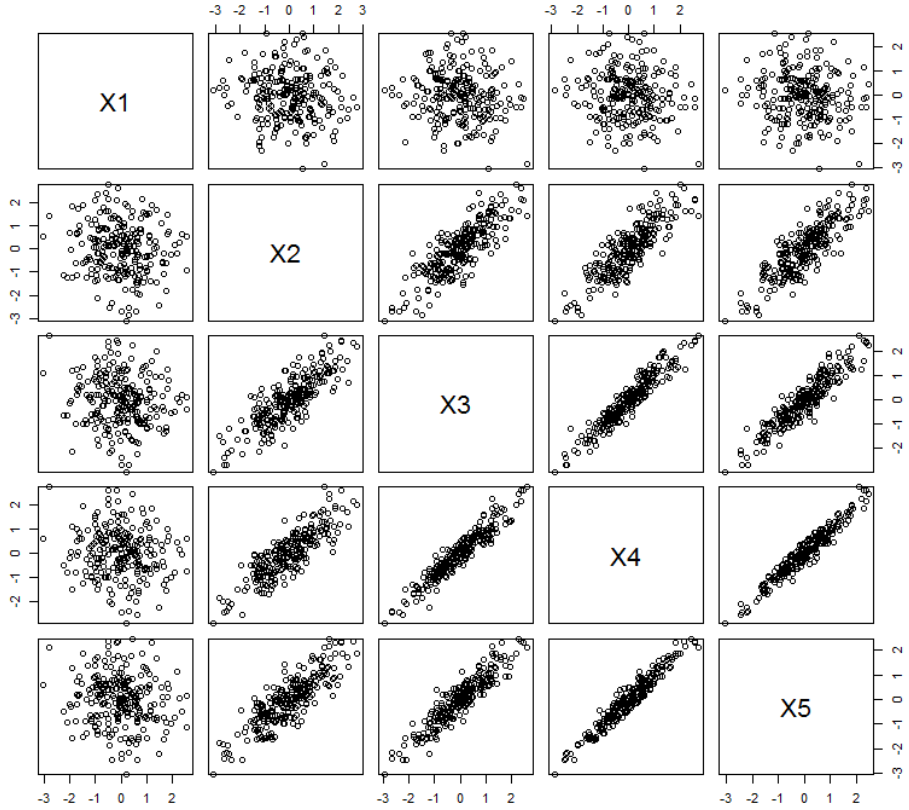
- If you have more columns than rows, you might end up creating a model with is overfit
- **How do you know which features are important?**

By the end of this lesson

- use PCA (Principal Component Analysis) to reduce the size of a given dataset's features to a strictly smaller number of features
- understand how to interpret the eigenvalues and variance explained by a certain number of PCs after performing PCA
- use Lasso/L1 regularization for both feature selection and general regression/ classification problems
- use Ridge/L2 regularization for regression and classification problems
- explain the difference between L1/L2 regularization

Not all columns are critical

- Several columns may be highly correlated with each other or with the outcome variable
- Columns may not be correlated at all with the outcome (have no "predictive ability")
- Two strategies to deal with this are
  - Dimensionality reduction
  - Regularization



Dimensionality Reduction

DEALING WITH A WIDE MATRIX

Reassigning the equation

- Previously, we considered the linear model in matrix notation as:
- Where
  - **Y** is the output
  - **X** is the matrix of predictors
  - **β** is the **unknown** coefficients

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon.$$

?

Reassigning the equation

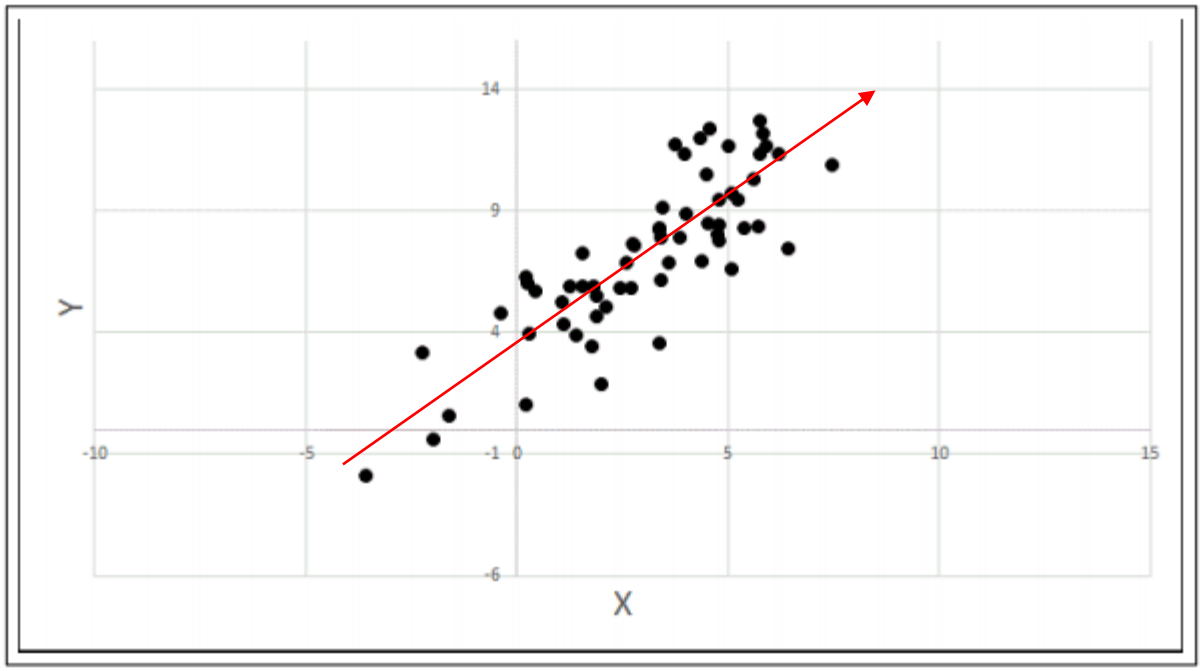
- Previously, we considered the linear model in matrix notation as:
- Where
  - **Y** is the output
  - **X** is the matrix of predictors
  - **β** is the unknown coefficients
- PCA changes the model where **Y** is known, but **X** and **β** are **unknown**
- PCA is known as a method that reduces dimensionality of the output data by accurate description of the correlation structure

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon.$$

? ?

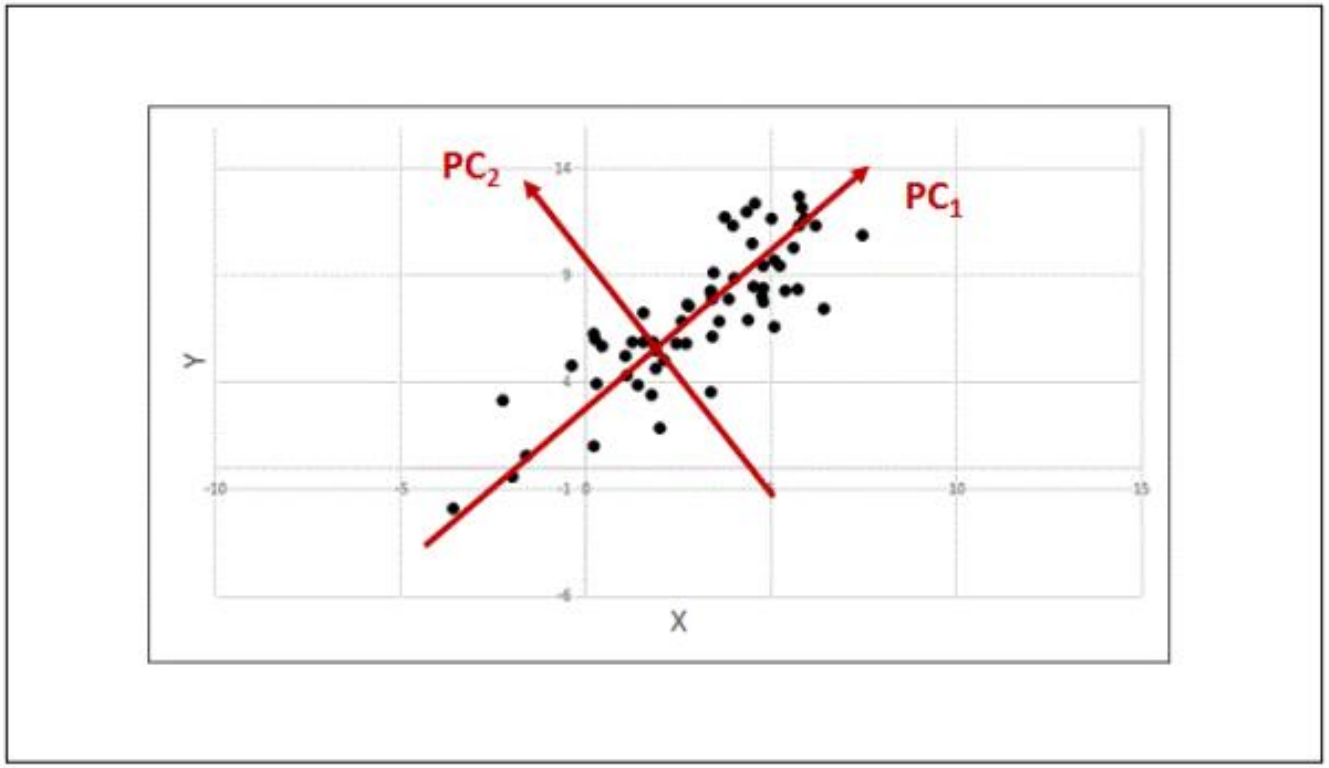
The real dimension of the data

- Here we have a cloud of observations in the Feature space, with predictors X and Y
- What is the best line to represent this data?



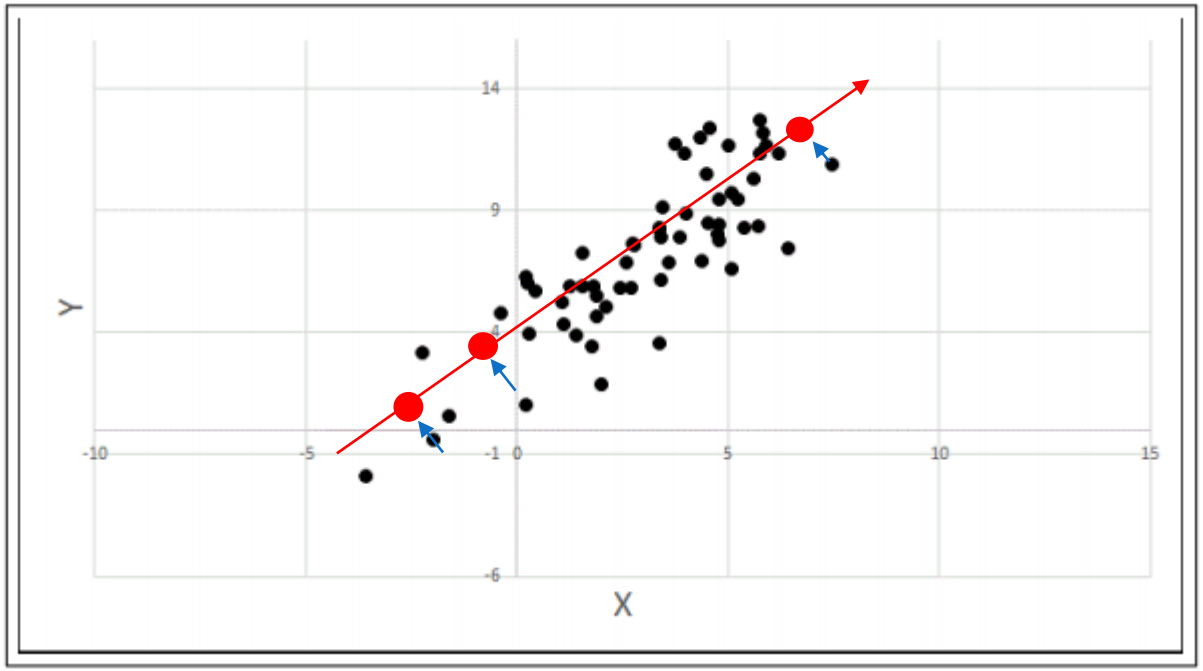
The real dimension of the data

- PCA finds a new coordinate system obtained from the old one by translation and rotation only and moves center of coordinate system to center of data.
- It moves the x axis onto the principal axis of variation, where you see the most variation relative to other data points
- It moves further axes down the road into orthogonal, less important directions of variation
- PCA finds for you these axes and also tells you how important these axes are. This allows you to use only the most important principal components when building your model.



Reduce the dimension of the data

- PCA projects the data in the original space into a reduced dimensional space
- A 1D line can represent a 2D cloud
- A 2D plane can represent a 3D cloud
- .....
- A 3D cube can represent a multi-dimensional feature space



Eigenvectors and Eigenvalues

Definition

For a given $n \times n$ matrix A a $n \times 1$ vector $\mathbf{u} \neq 0$ with unit length and a scalar λ are called **eigenvector** and **eigenvalue** of the matrix A,respectively, if

$$A\mathbf{u} = \lambda\mathbf{u}.$$

- **u** is an n x 1 vector
- An eigenvector is a vector such that after left-multiplication by a matrix it does not change direction, but gets multiplied by a scalar **λ** which is the eigenvalue

Decompositions

- Need to use these in the proof for PCA
- Matrix decomposition
- Spectral decomposition

Matrix decomposition: Allows us to represent a matrix by its eigenvalues and eigenvectors

Fact

If A is a real symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$ and eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ then

$$A = U\Lambda U^T = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n]^T,$$

Spectral decomposition (Layer Cake): Allows us to represent a matrix as a sum of **n** components, each defined by its eigenvector (Principal component direction) and weighted by its eigenvalue

Fact

If A is a real symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$ and eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ then

$$A = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T.$$

Outline of the algorithm

1. Create centered matrix

$$\begin{aligned}\mathbf{Y}_0 &= [\mathbf{y}^0_{\cdot 1}, \mathbf{y}^0_{\cdot 2}, \ldots, \mathbf{y}^0_{\cdot n}] \\ &= [\mathbf{y}_{\cdot 1} - \mathbf{1}\bar{y}_{\cdot 1}, \mathbf{y}_{\cdot 2} - \mathbf{1}\bar{y}_{\cdot 2}, \ldots, \mathbf{y}_{\cdot n} - \mathbf{1}\bar{y}_{\cdot n}]\end{aligned}$$

2. Calculate covariance matrix

$$\Theta = cov\left(\mathbf{Y}_0\right) = \left\{\mathbb{E}\left[\mathbf{y}^0_{\cdot i}\mathbf{y}^0_{\cdot j}\right]\right\}$$

3. Perform eigenvalue decomposition of Θ. Define

$$\mathbf{L} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m],$$

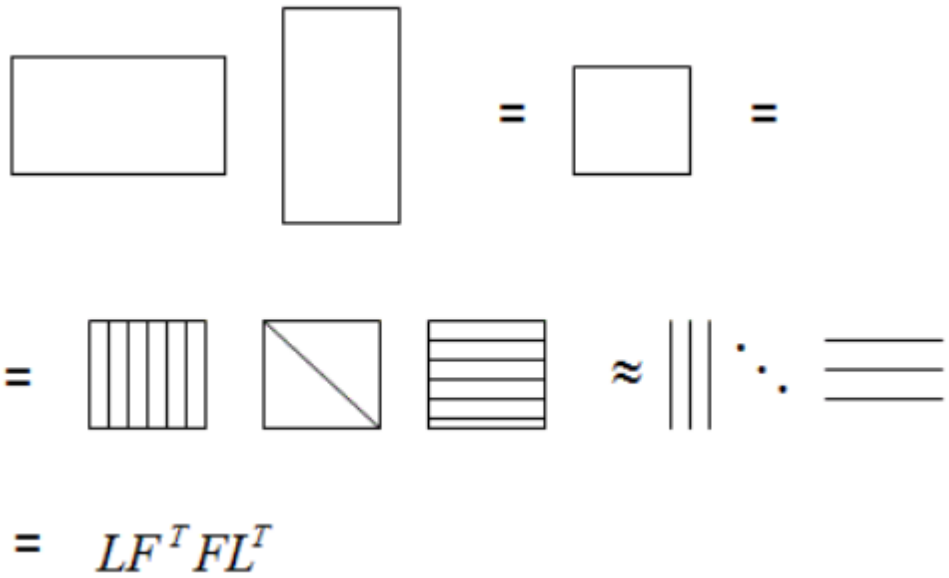
where $\mathbf{u}_i$ is the eigenvector corresponding to the i-s largest eigenvalue of Θ.

4. Define

$$\mathbf{F} = \mathbf{Y}_0\mathbf{L},$$

right-multiplying by $\mathbf{L}$ the definition of the model

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T.$$



Outline of the algorithm

1. Create centered matrix

$$\begin{aligned}\mathbf{Y}_0 &= [\mathbf{y}^0_{\cdot 1}, \mathbf{y}^0_{\cdot 2}, \ldots, \mathbf{y}^0_{\cdot n}] \\ &= [\mathbf{y}_{\cdot 1} - \mathbf{1}\bar{y}_{\cdot 1}, \mathbf{y}_{\cdot 2} - \mathbf{1}\bar{y}_{\cdot 2}, \ldots, \mathbf{y}_{\cdot n} - \mathbf{1}\bar{y}_{\cdot n}]\end{aligned}$$

2. Calculate covariance matrix

$$\Theta = cov\left(\mathbf{Y}_0\right) = \left\{\mathbb{E}\left[\mathbf{y}^0_{\cdot i}\mathbf{y}^0_{\cdot j}\right]\right\}$$

3. Perform eigenvalue decomposition of Θ. Define

$$\mathbf{L} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m],$$

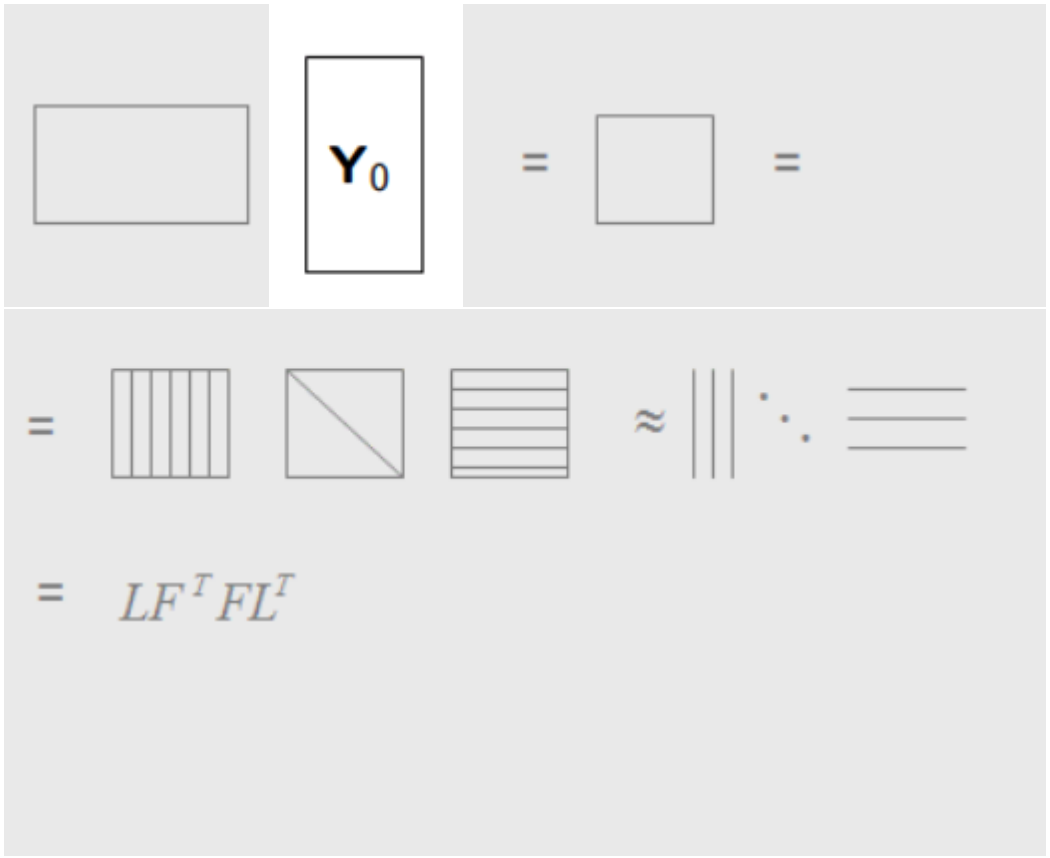
where $\mathbf{u}_i$ is the eigenvector corresponding to the i-s largest eigenvalue of Θ.

4. Define

$$\mathbf{F} = \mathbf{Y}_0\mathbf{L},$$

right-multiplying by $\mathbf{L}$ the definition of the model

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T.$$



Outline of the algorithm

1. Create centered matrix

$$\begin{aligned}\mathbf{Y}_0 &= [\mathbf{y}^0_{\cdot 1}, \mathbf{y}^0_{\cdot 2}, \ldots, \mathbf{y}^0_{\cdot n}] \\ &= [\mathbf{y}_{\cdot 1} - \mathbf{1}\bar{y}_{\cdot 1}, \mathbf{y}_{\cdot 2} - \mathbf{1}\bar{y}_{\cdot 2}, \ldots, \mathbf{y}_{\cdot n} - \mathbf{1}\bar{y}_{\cdot n}]\end{aligned}$$

2. Calculate covariance matrix

$$\Theta = cov(\mathbf{Y}_0) = \{\mathbb{E}[\mathbf{y}^0_{\cdot i}\mathbf{y}^0_{\cdot j}]\}$$

3. Perform eigenvalue decomposition of Θ. Define

$$\mathbf{L} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m],$$

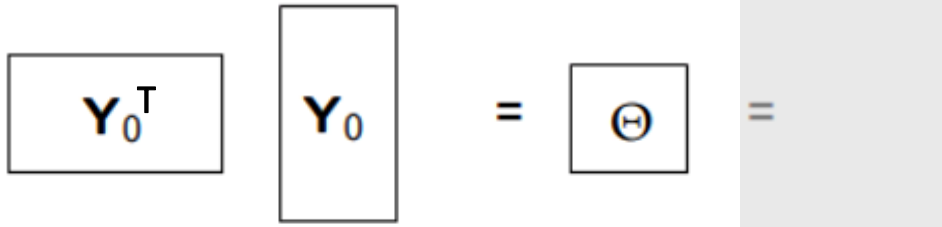
where $\mathbf{u}_i$ is the eigenvector corresponding to the i-s largest eigenvalue of Θ.

4. Define

$$\mathbf{F} = \mathbf{Y}_0\mathbf{L},$$

right-multiplying by $\mathbf{L}$ the definition of the model

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T.$$



$$= LF^TFL^T$$

Outline of the algorithm

1. Create centered matrix

$$\begin{aligned}\mathbf{Y}_0 &= [\mathbf{y}^0_{\cdot 1}, \mathbf{y}^0_{\cdot 2}, \ldots, \mathbf{y}^0_{\cdot n}] \\ &= [\mathbf{y}_{\cdot 1} - \mathbf{1}\bar{y}_{\cdot 1}, \mathbf{y}_{\cdot 2} - \mathbf{1}\bar{y}_{\cdot 2}, \ldots, \mathbf{y}_{\cdot n} - \mathbf{1}\bar{y}_{\cdot n}]\end{aligned}$$

2. Calculate covariance matrix

$$\Theta = cov(\mathbf{Y}_0) = \{\mathbb{E}[\mathbf{y}^0_{\cdot i}\mathbf{y}^0_{\cdot j}]\}$$

3. Perform eigenvalue decomposition of Θ. Define

$$\mathbf{L} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m],$$

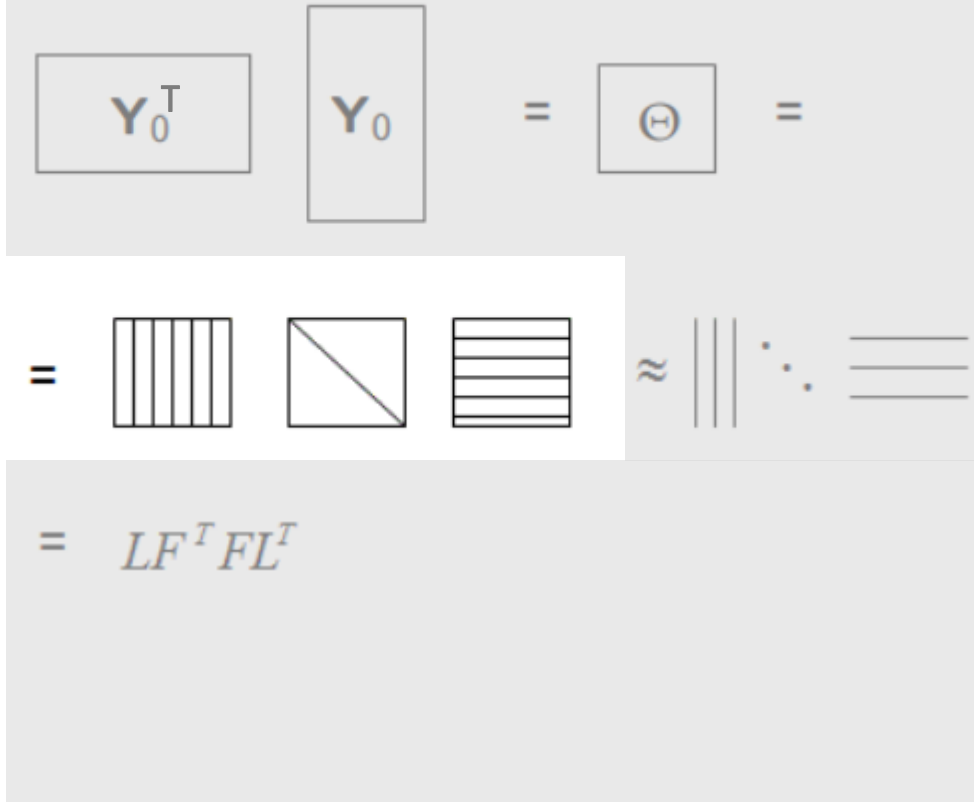
where $\mathbf{u}_i$ is the eigenvector corresponding to the i-s largest eigenvalue of Θ.

4. Define

$$\mathbf{F} = \mathbf{Y}_0\mathbf{L},$$

right-multiplying by $\mathbf{L}$ the definition of the model

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T.$$



Outline of the algorithm

1. Create centered matrix

$$\begin{aligned}\mathbf{Y}_0 &= [\mathbf{y}^0_{\cdot 1}, \mathbf{y}^0_{\cdot 2}, \ldots, \mathbf{y}^0_{\cdot n}] \\ &= [\mathbf{y}_{\cdot 1} - \mathbf{1}\bar{y}_{\cdot 1}, \mathbf{y}_{\cdot 2} - \mathbf{1}\bar{y}_{\cdot 2}, \ldots, \mathbf{y}_{\cdot n} - \mathbf{1}\bar{y}_{\cdot n}]\end{aligned}$$

2. Calculate covariance matrix

$$\Theta = cov(\mathbf{Y}_0) = \left\{\mathbb{E}\left[\mathbf{y}^0_{\cdot i}\mathbf{y}^0_{\cdot j}\right]\right\}$$

3. Perform eigenvalue decomposition of Θ. Define

$$\mathbf{L} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m],$$

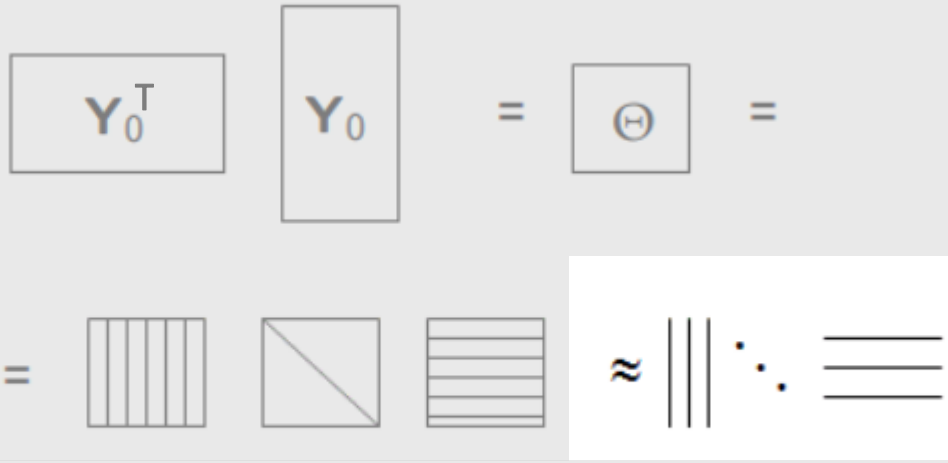
where $\mathbf{u}_i$ is the eigenvector corresponding to the i-s largest eigenvalue of Θ.

4. Define

$$\mathbf{F} = \mathbf{Y}_0\mathbf{L},$$

right-multiplying by $\mathbf{L}$ the definition of the model

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T.$$



$= LF^TFL^T$

Only need a few eigenvectors (lines) to represent matrix. Determined by largest eigenvalues (dots)

Outline of the algorithm

1. Create centered matrix

$$\begin{aligned}\mathbf{Y}_0 &= [\mathbf{y}^0_{\cdot 1}, \mathbf{y}^0_{\cdot 2}, \ldots, \mathbf{y}^0_{\cdot n}] \\ &= [\mathbf{y}_{\cdot 1} - \mathbf{1}\bar{y}_{\cdot 1}, \mathbf{y}_{\cdot 2} - \mathbf{1}\bar{y}_{\cdot 2}, \ldots, \mathbf{y}_{\cdot n} - \mathbf{1}\bar{y}_{\cdot n}]\end{aligned}$$

2. Calculate covariance matrix

$$\Theta = cov(\mathbf{Y}_0) = \{\mathbb{E}[\mathbf{y}^0_{\cdot i}\mathbf{y}^0_{\cdot j}]\}$$

3. Perform eigenvalue decomposition of Θ. Define

$$\mathbf{L} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m],$$

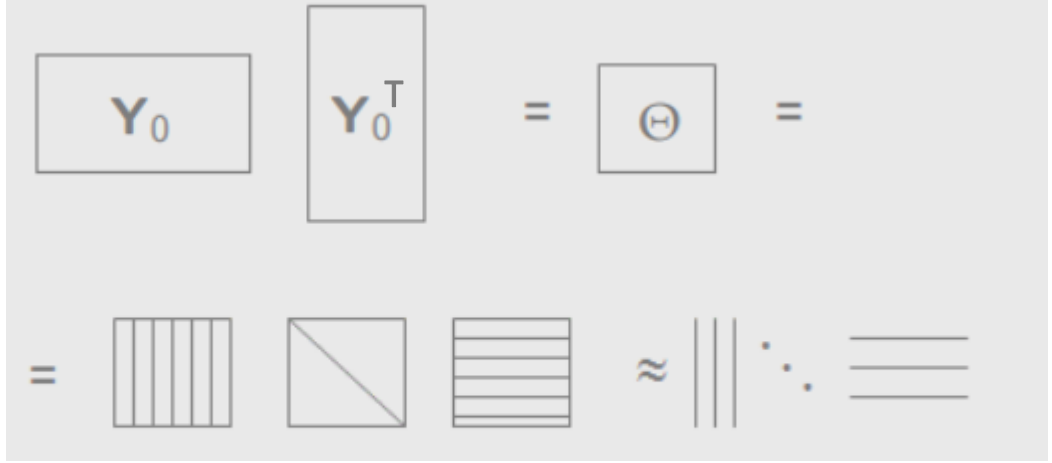
where $\mathbf{u}_i$ is the eigenvector corresponding to the i-s largest eigenvalue of Θ.

4. Define

$$\mathbf{F} = \mathbf{Y}_0\mathbf{L},$$

right-multiplying by $\mathbf{L}$ the definition of the model

$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T.$$



$$= LF^TFL^T$$

Factors and Loadings

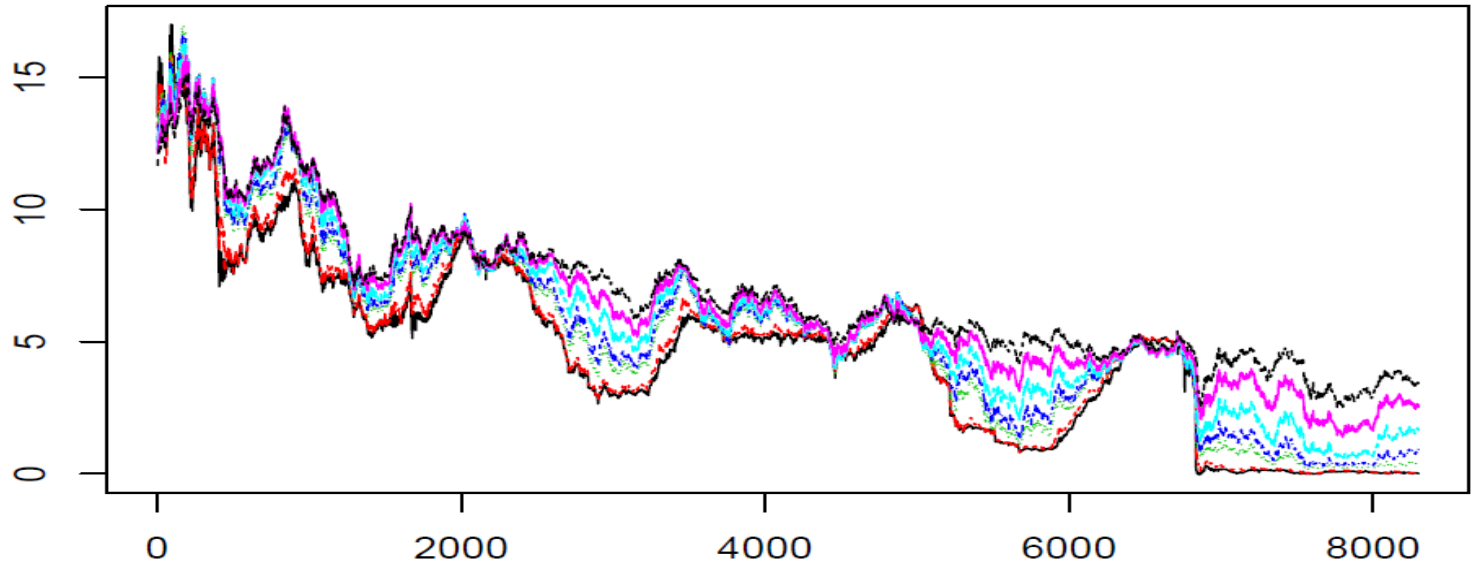
$$\mathbf{Y}_0 = \mathbf{F}\mathbf{L}^T$$

- **F** plays the role of inputs. The columns are usually called principal components, factors or factor scores
- L plays the role of slopes. The columns (rows in L^T) are called factor loadings

PCA example using Treasury data

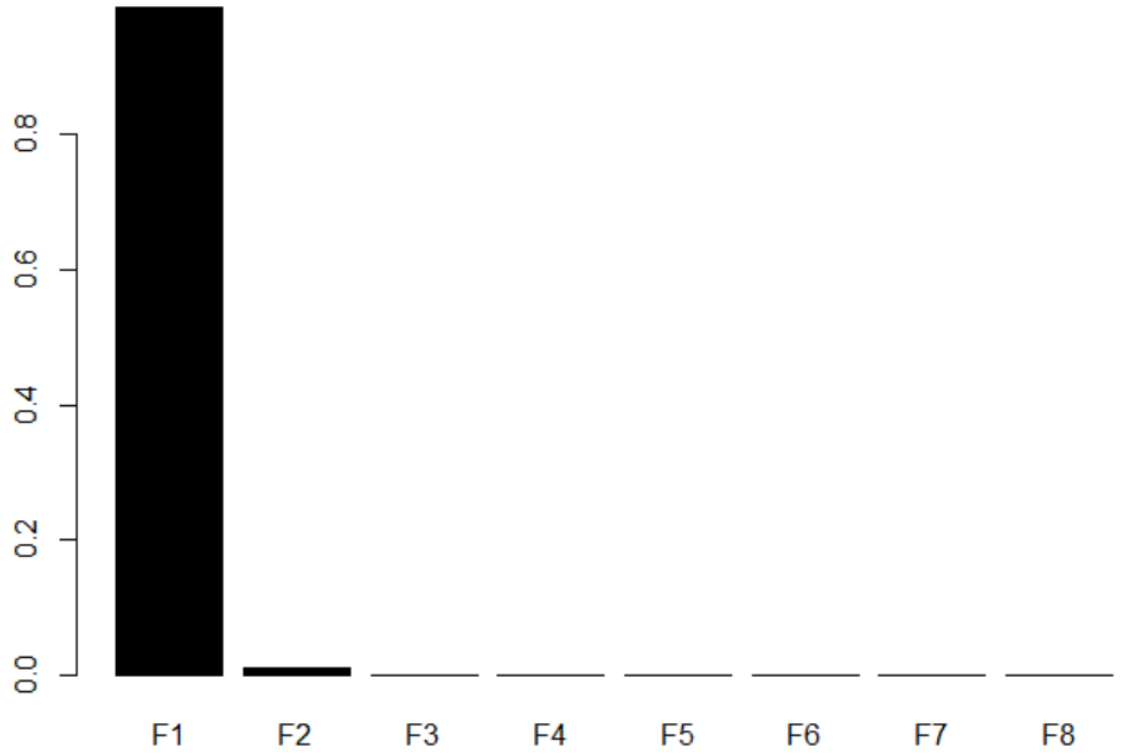
- Long data set of various treasury maturity yields
- Over 8,000 observations
- Is it possible to represent the yield curve using PCA?

	USGG3M	USGG6M	USGG2YR	USGG3YR	USGG5YR	USGG10YR	USGG30YR
1/5/1981	13.52	13.09	12.289	12.28	12.294	12.152	11.672
1/6/1981	13.58	13.16	12.429	12.31	12.214	12.112	11.672
1/7/1981	14.50	13.90	12.929	12.78	12.614	12.382	11.892
1/8/1981	14.76	14.00	13.099	12.95	12.684	12.352	11.912
1/9/1981	15.20	14.30	13.539	13.28	12.884	12.572	12.132
1/12/1981	15.22	14.23	13.179	12.94	12.714	12.452	12.082



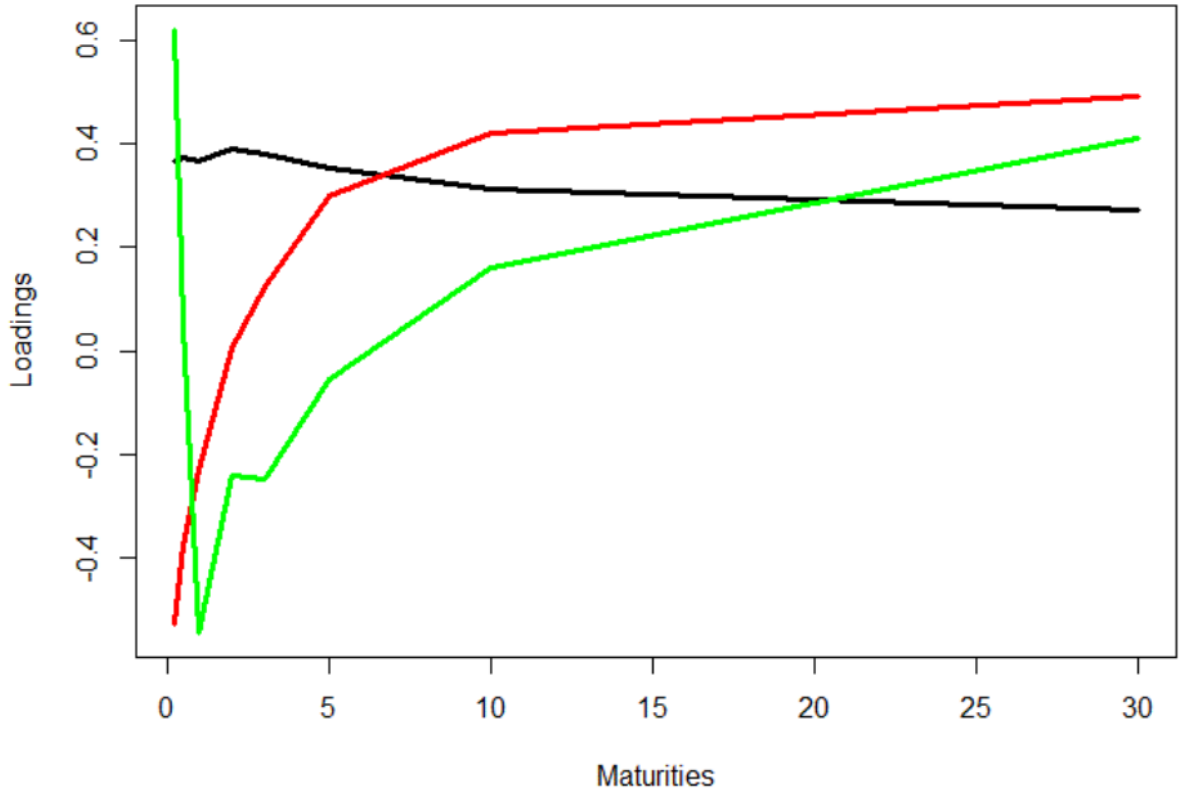
Factors of treasury data

- We only need the first three factors (principal components) to represent 99% of the relative variance in the original data

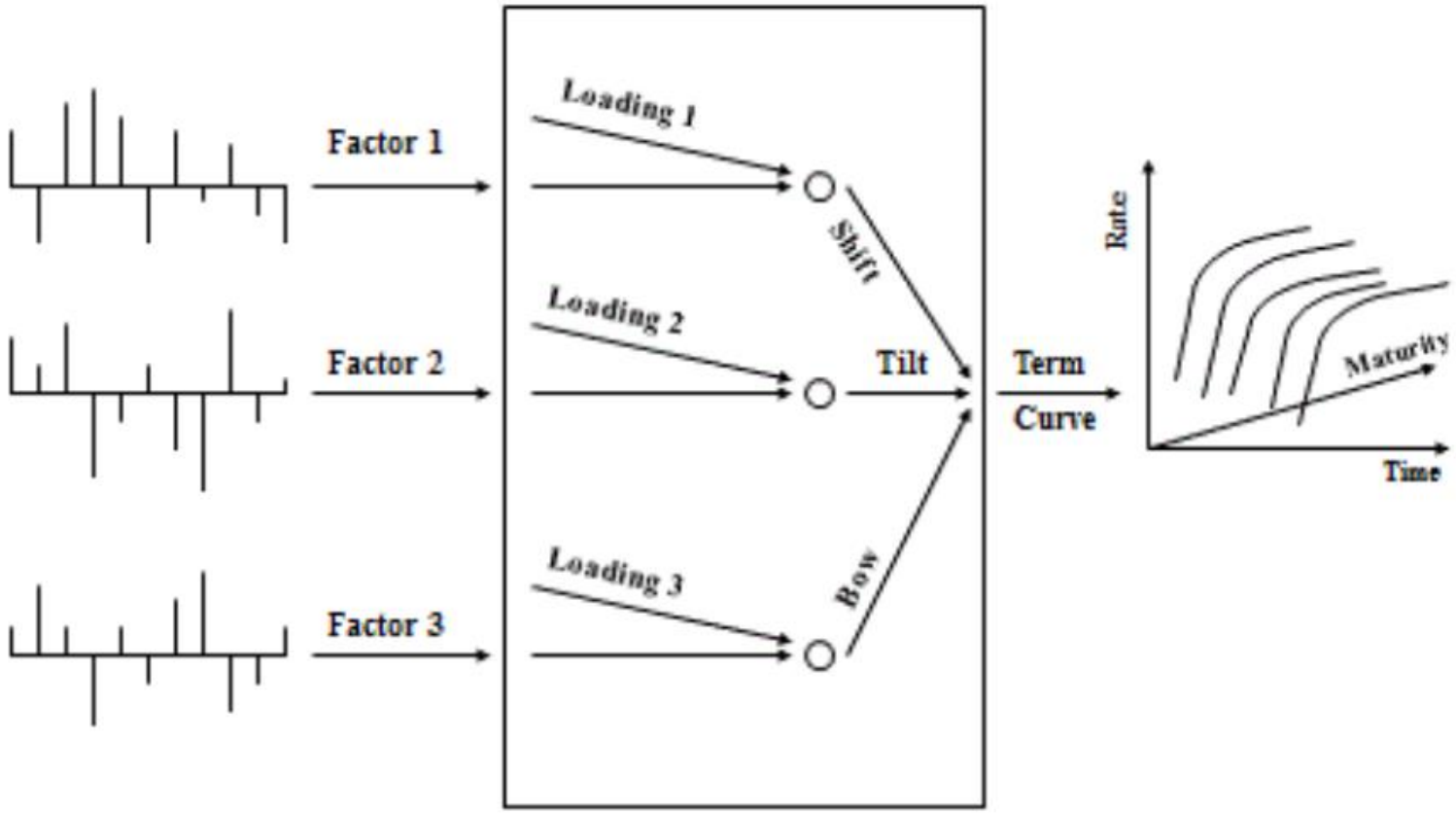


Loadings of treasury data

- If we plot the loadings, the shapes can let us interpret the data
- Black – Level
  - Parallel shifts of the curve
- Red – Slope
  - Tilting of the yield curve
- Green – Curvature
  - Flexing of the yield curve



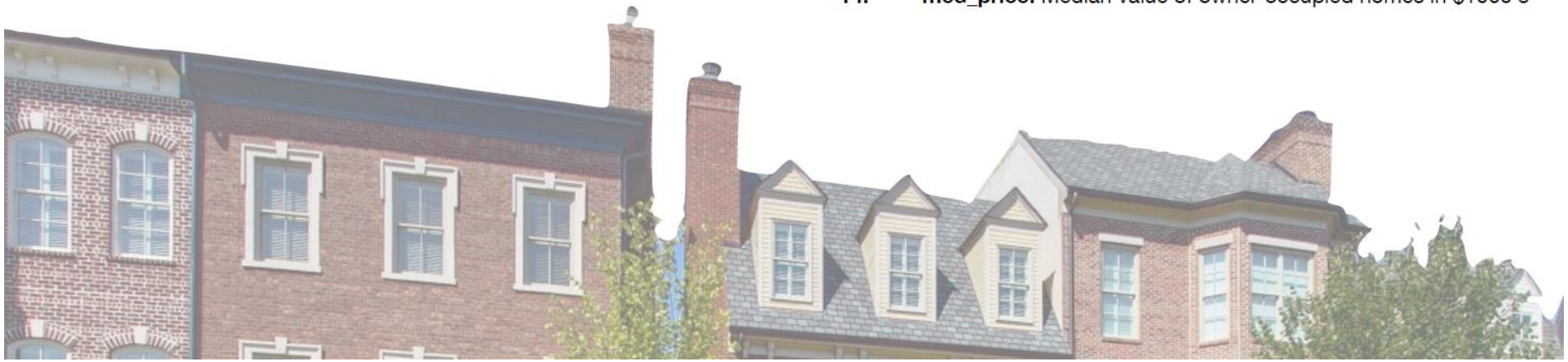
Interpreting the PCA model



PCA in the notebook

Here's the dataset schema:

1. **crime:** per capita crime rate by town
2. **zone:** proportion of residential land zoned for lots over 25,000 sq.ft.
3. **industry:** proportion of non-retail business acres per town
4. **charles:** Charles River dummy variable (1 if tract bounds river; 0 otherwise)
5. **no:** nitric oxides concentration (parts per 10 million)
6. **rooms:** average number of rooms per dwelling
7. **age:** proportion of owner-occupied units built prior to 1940
8. **distance:** weighted distances to five Boston employment centres
9. **radial:** index of accessibility to radial highways
10. **tax:** full-value property-tax rate per 10K dollars
11. **pupil:** pupil-teacher ratio by town
12. **aam:** where aam is the proportion of african americans by town
13. **lower:** percentage lower income status of the population
14. **med_price:** Median value of owner-occupied homes in $1000's



Supervised – Principal components regression

- Using least squares regression, we are given Y and X.
- Use PCA to decompose X into factors and loadings
- Use Y (response) and the most important PC/factors(inputs) in a regression model
- Decreases bias and increases variance

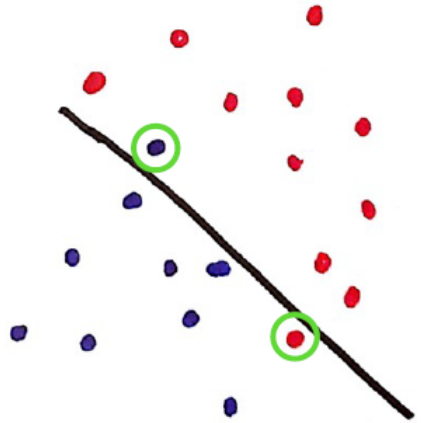
$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon.$$

Create a predictive regression model using less predictors

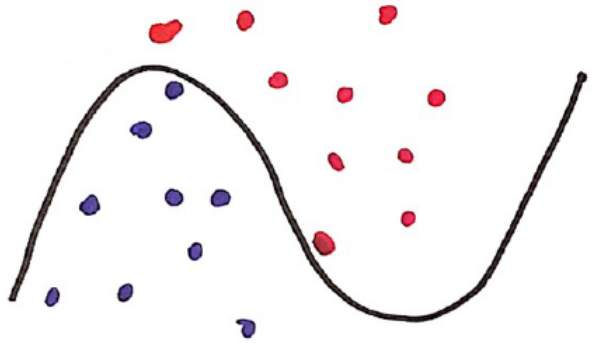
Regularization

LASSO AND RIDGE

Regularization



- Two Errors



- No errors

So is this model better?

Well, the one on the left makes a couple errors, while the model on the right has a lower error. So, if one were to choose a model based on the lowest error, the model on the right would win!

However, that model is also much more complicated!

If you take all the coefficients of the equations and add them to the error (which we'll get into shortly), you'll see that the model on the left actually has a smaller combined error.

Simple vs Complex

So, when is simple better over complex and vice versa?

We might be ok with some complexity if we're building a model that has little room for error (like a medical model or a model to send a rocket into space). Models that predict what movies you might like or who you might be friends with on Snapchat, however, have more room for experimenting and need to be simpler and faster to run on big data, so we're ok with some errors. Errors here are also not a matter of life or death.

Case 1

- requires low error
- ok if it's complex
- punishment on the complexity should be small

Case 2

- requires simplicity
- ok with errors
- punishment on the complexity should be large

Shrinkage methods

- When there are too many parameters and their number has not been reduced before fitting a model, there is a need to remove some of them in the process of fitting
- Based on constraining coefficient estimates, i.e. shrinks them to zero
- Lasso and Ridge are common shrinkage methods

Ridge regression i

- In the process of fitting linear regression model we minimize the sum of squares

$$RSS\left(\boldsymbol{\beta}; \mathbf{y}, \mathbf{x}\right) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2$$

- In order to "encourage" the coefficients to be closer to zero ridge regression replaces minimization of RSS with minimization of

$$RSS\left(\boldsymbol{\beta}; \mathbf{y}, \mathbf{x}\right) + \alpha\left(\boldsymbol{\beta}\right) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

- Function $\alpha\left(\boldsymbol{\beta}\right) = \lambda\sum_{j=1}^{p}\beta_j^2$ is called regularizator. Parameter $\lambda \geq 0$ is a tuning parameter.

Ridge regression ii

- As with least squares, ridge regression seeks coefficient estimates that fit the data, by making the RSS, the first term, small.
- The second term, called regularizator or shrinkage penalty, is small when $\beta_1, \ldots, \beta_p$ are close to zero. It forces the coefficients to be smaller in absolute value
- The objective is to eliminate small coefficients completely if they are not large enough
- The tuning parameter λ controls the relative impact of these two terms on the regression coefficient estimates
- When $\lambda = 0$, the penalty term has no effect, and ridge regression will produce the least squares estimates
- However, as $\lambda \longrightarrow \infty$, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will all approach zero
- Unlike least squares, which generates only one set of coefficient estimates, ridge regression will produce a different set of coefficient estimates for each value of λ. Selecting a good value for λ is critical

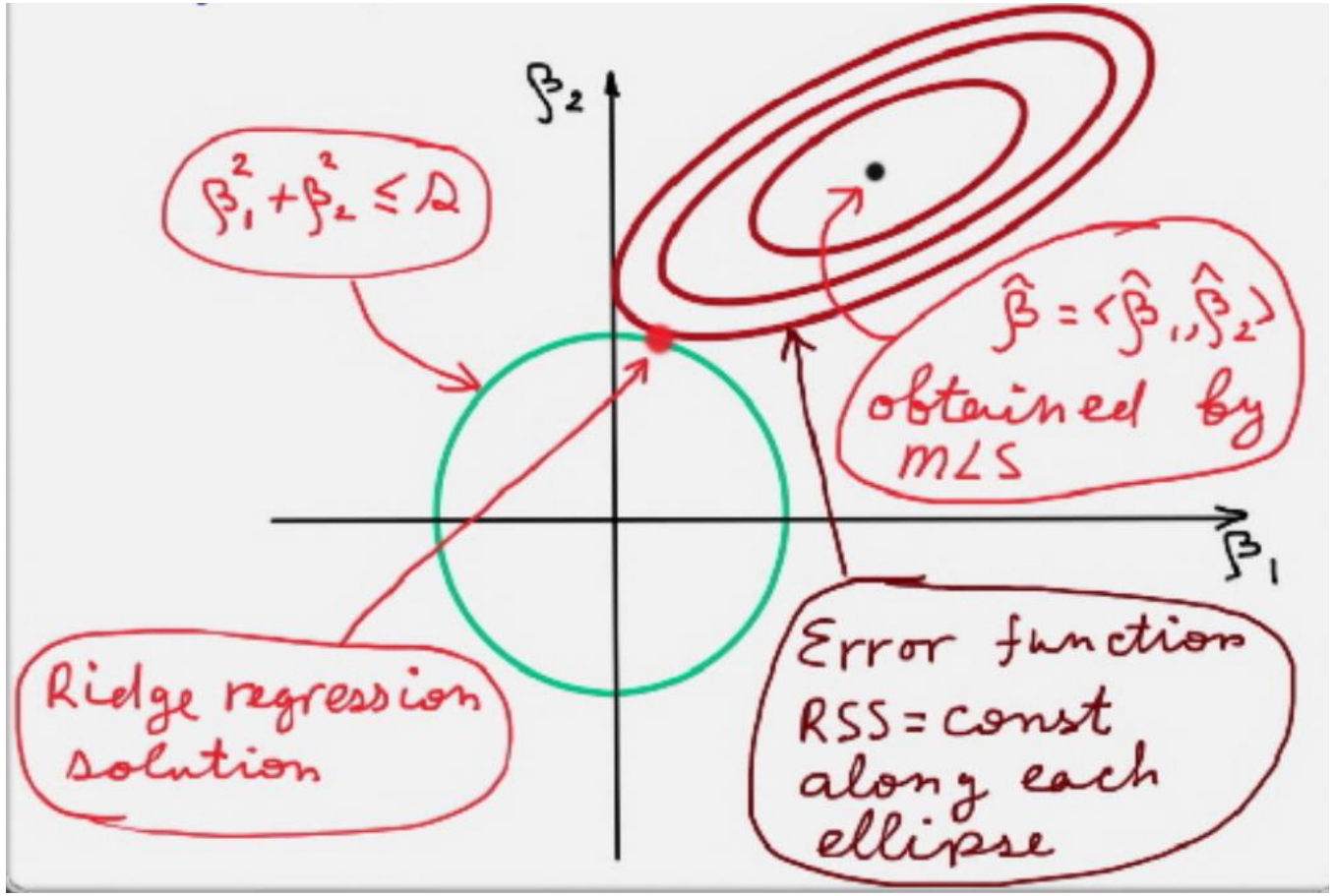
Ridge regression iii

- What ridge regression is trying to achieve is reduction in the variance of estimators
- As λ increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias for the coefficients
- This effect is called the bias-variance trade-off
- Regularization with L_2-norm regularizator which in fact is

$$\|\boldsymbol{\beta}\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j^2}$$

  also improves computational efficiency of least squares
- Effect of ridge regression can be illustrated on a simple graph

Ridge regression iv



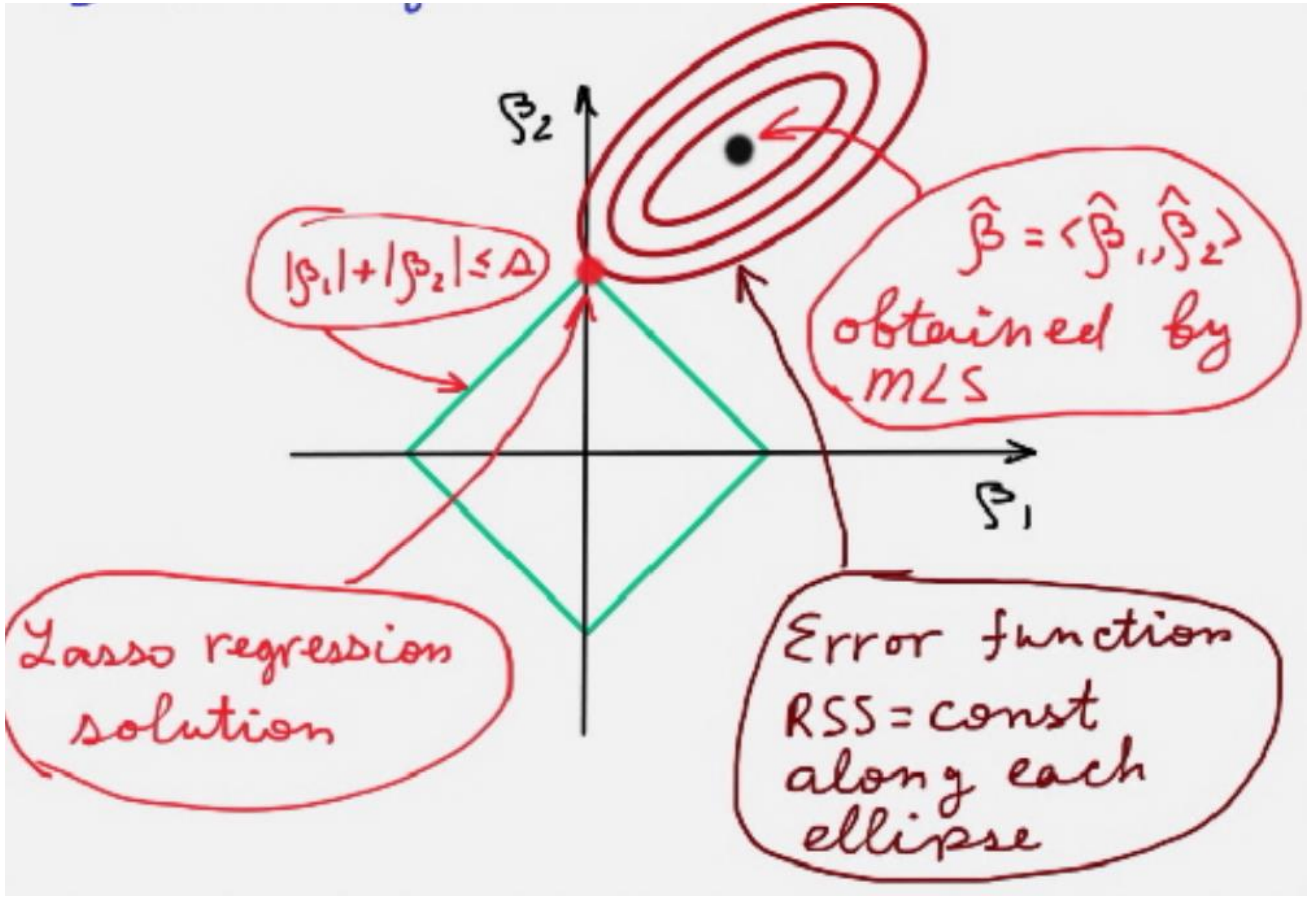
Lasso regression i

- One disadvantage of ridge regression is that it shrinks all coefficients towards zero in a pretty uniform way and it may not make them exactly zero
- Lasso regression is designed to overcome that weakness of ridge regression. It replaces the L_2-norm regularizator of ridge regression with L_1-norm regularizator

$$RSS\left(\boldsymbol{\beta}; \mathbf{y}, \mathbf{x}\right) + \alpha\left(\boldsymbol{\beta}\right) = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda\sum_{j=1}^{p}\left|\beta_j\right|$$

- As with ridge regression, the lasso shrinks the coefficient estimates towards zero. However, in the case of the lasso, the penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter λ is sufficiently large
- As in ridge regression, selecting a good value of λ for the lasso is critical and done using cross-validation

Lasso regression ii



Recap

Lasso:

- produces sparse models
- is useful for strong feature selection in order to improve model performance, or to minimize the number of explanatory variables.
- can produce non-unique solutions (when some features are very strongly correlated)
- can produce very different solutions depending on slight changes in features (because of non-uniqueness)

Ridge:

- produces stable models with smooth non-zero coefficients across features.
- is useful for data interpretation, understanding what features, even when correlated, may be used in combination to predict the response.
- may tell you something about how the data itself was generated.

You can combine both Lasso and Ridge models into a single penalized model (that uses a weighted combination of Lasso and Ridge regression). This is called the `ElasticNet`.

Appendix

Acknowledgements

Material for these slides are taken from but not limited to the following sources:

Balasanov, Yuri Statistical Analysis Session 9