LECTURE 13:

DIMENSIONALITY REDUCTION

October 25, 2017

SDS 293: Machine Learning

Outline

- Model selection: alternatives to least-squares

✓Subset selection

  ✓Best subset

  ✓Stepwise selection (forward and backward)

  ✓Estimating error using cross-validation

✓Shrinkage methods

  ✓Ridge regression and the Lasso

  – **Dimension reduction**

- Labs for each part

Recap: Ridge Regression and the Lasso

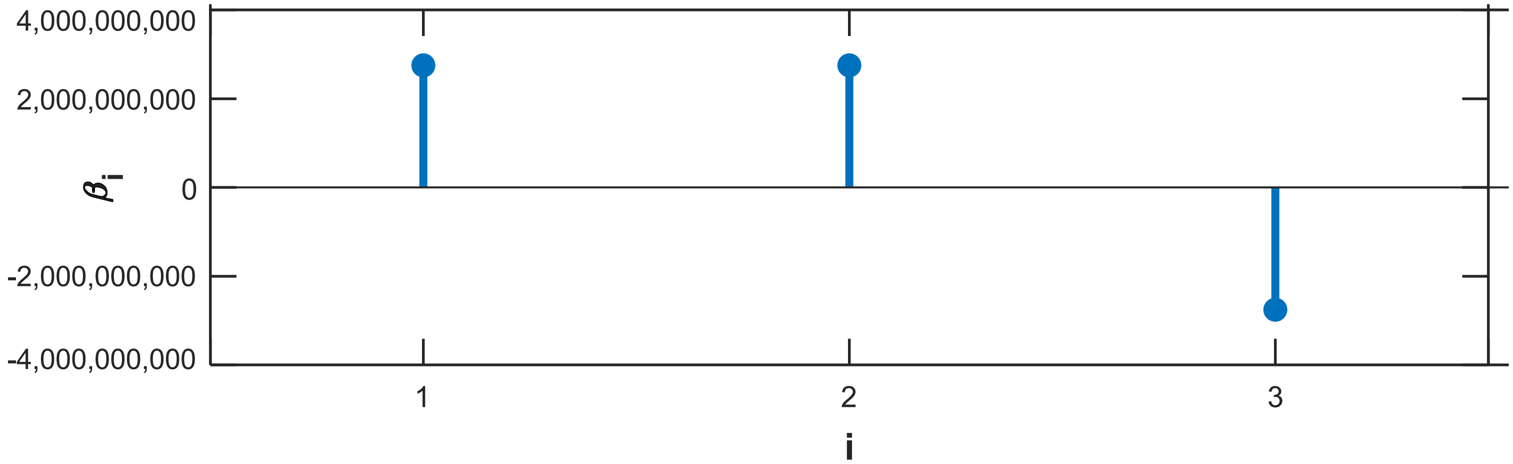
- Both are "shrinkage" methods
- Estimates for the coefficients are ***biased*** toward the origin
  - Biased = "prefers some estimates to others"
  - Does not yield the true value in expectation
- Question: why would we **want** a biased estimate?

Recap: Ridge Regression and the Lasso

- Both are "shrinkage" methods
- Estimates for the coefficients are ***biased*** toward the origin
  - Biased = "prefers some estimates to others"
  - Does not yield the true value in expectation
- Question: why would we **want** a biased estimate?

What's wrong with bias?

- What if your unbiased estimator gives you this?



May want to bias our estimate to **reduce variance**

Flashback: superheroes

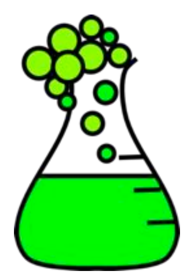
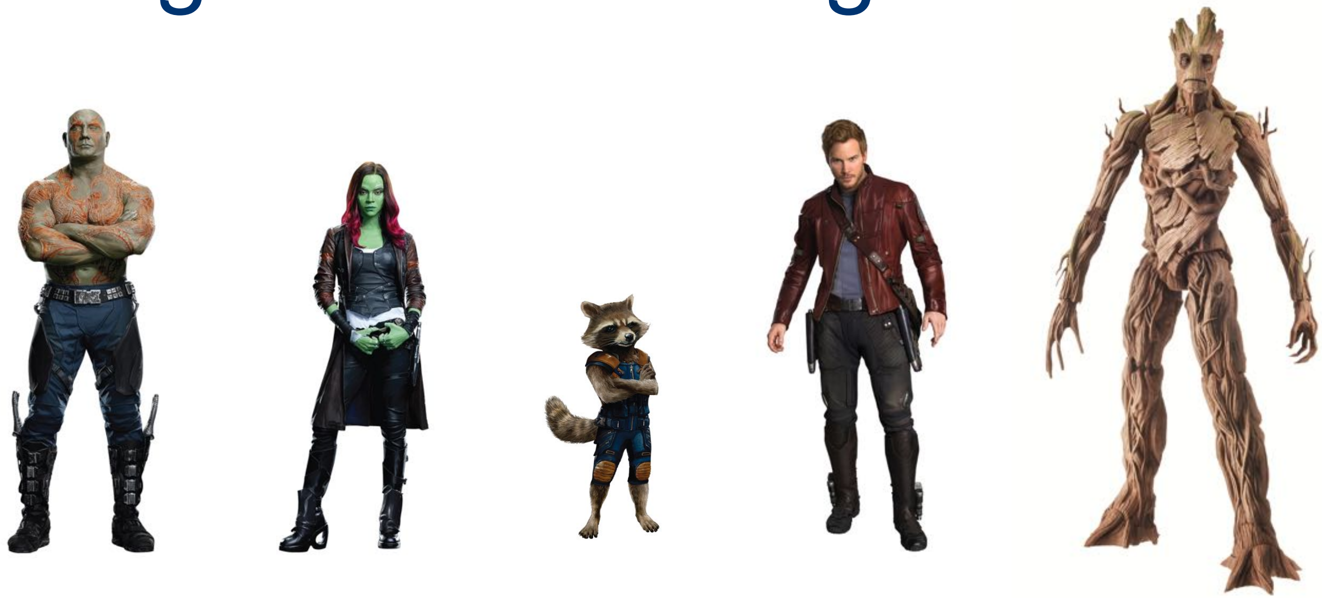
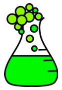
$height = \beta_1 ($  $) + \beta_2 ($  $) + \beta_3 ($  $)$

Image credit: Ming Malaykham

Estimating Guardians' Height

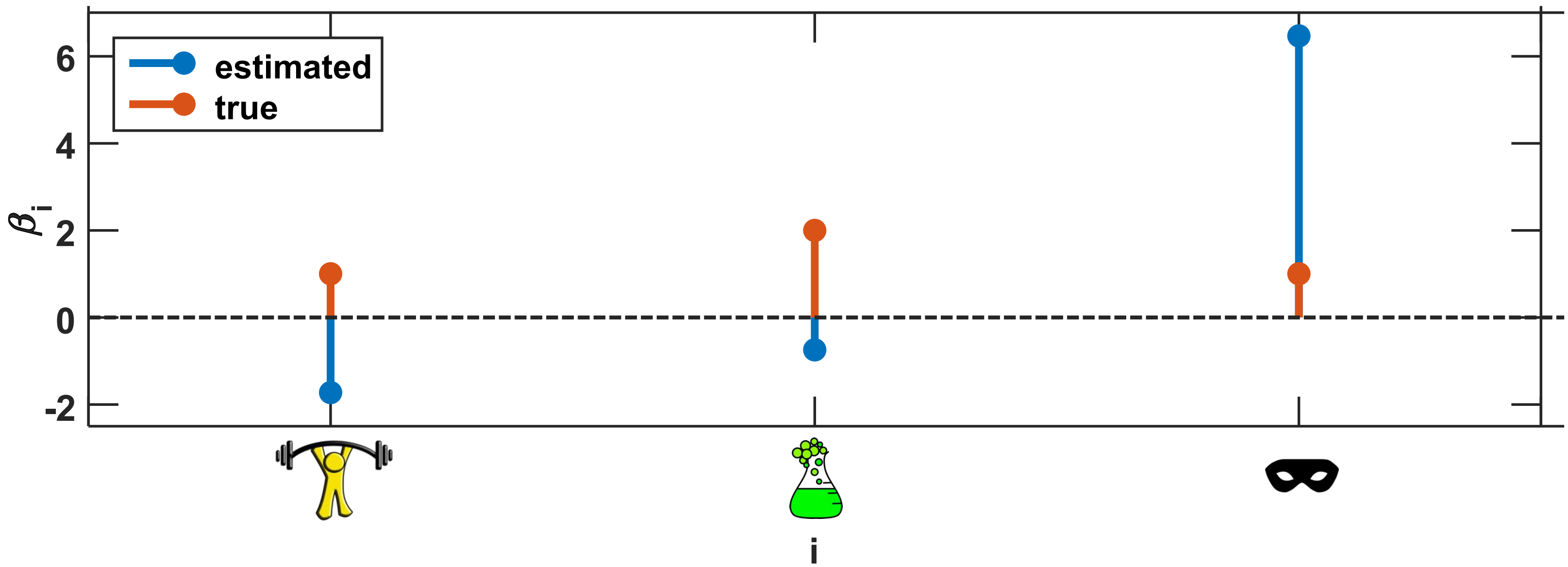


$$\begin{bmatrix} 232.03 \\ 156.29 \\ 113.82 \\ 229.07 \\ 287.72 \end{bmatrix} = 1 \begin{bmatrix} 63.9 \\ 28.9 \\ 54.3 \\ 69.8 \\ 50.4 \end{bmatrix} + 2 \begin{bmatrix} 54.0 \\ 45.1 \\ 13.3 \\ 49.5 \\ 85.4 \end{bmatrix} + 1 \begin{bmatrix} 59.1 \\ 36.9 \\ 33.7 \\ 59.7 \\ 67.9 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

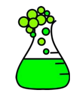
Estimate for β

- When we try to estimate using OLS, we get the following:

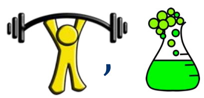


(Relatively) huge difference between actual and estimated coefficients

What's going on here?

$$\begin{bmatrix} 232.03 \\ 156.29 \\ 113.82 \\ 229.07 \\ 287.72 \end{bmatrix} = 1\begin{bmatrix} 63.9 \\ 28.9 \\ 54.3 \\ 69.8 \\ 50.4 \end{bmatrix} + 2\begin{bmatrix} 54.0 \\ 45.1 \\ 13.3 \\ 49.5 \\ 85.4 \end{bmatrix} + 1\begin{bmatrix} 59.1 \\ 36.9 \\ 33.7 \\ 59.7 \\ 67.9 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

$\approx avg\Big($  $\Big)$

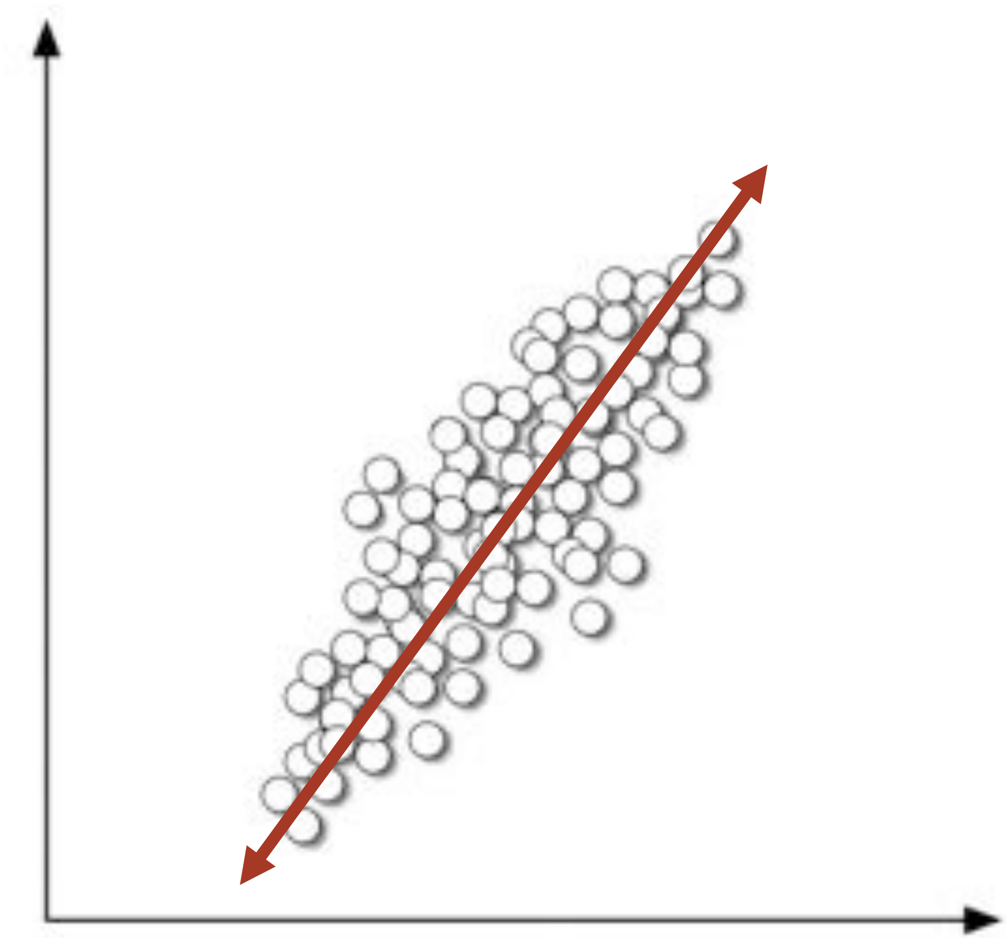
- Some dimensions are redundant
  - Little information in 3rd dimension not captured by the first two
  - In linear regression, redundancy causes noise to be **amplified**

Dimension reduction

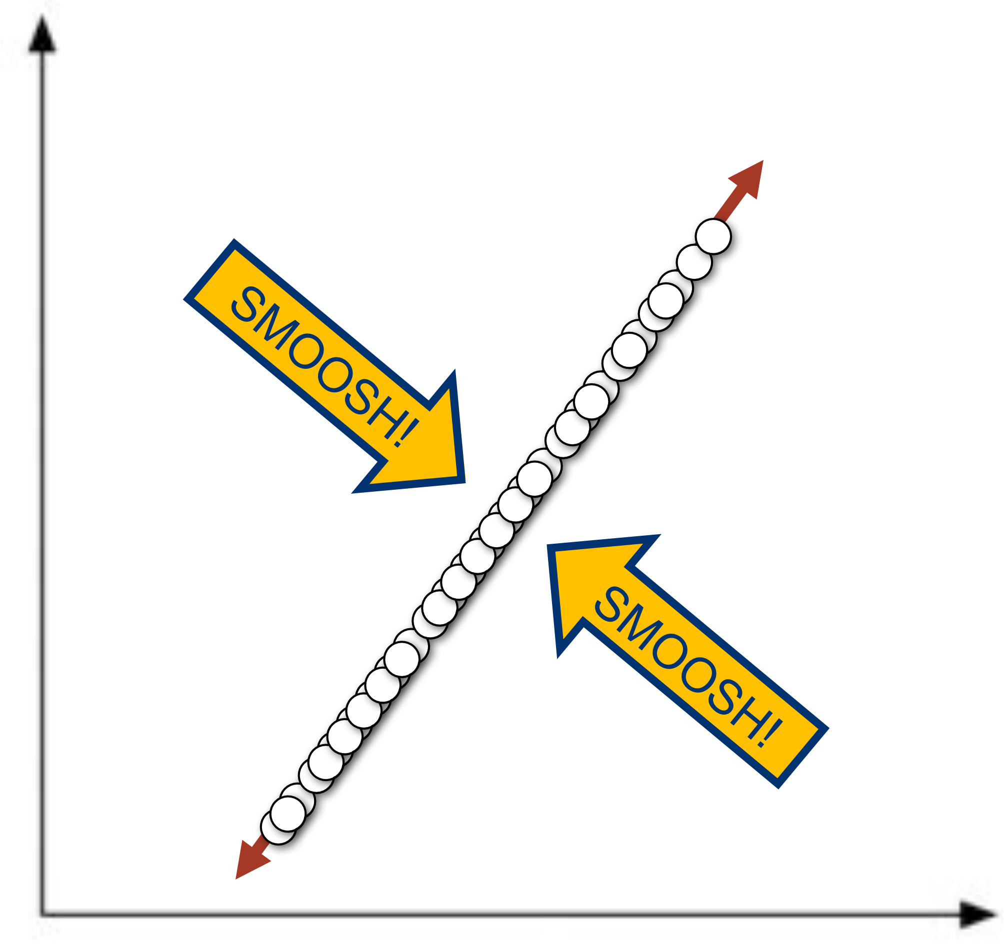
- **Current situation**: our data live in *p*-dimensional space, but not all *p* dimensions are equally useful
- **Subset selection**: throw some out
  - Pro: pretty easy to do
  - Con: lose some information
- **Alternate approach**: create **new** features that are combinations of the old ones

In other words:
Project the data into a new feature space to reduce variance in the estimate

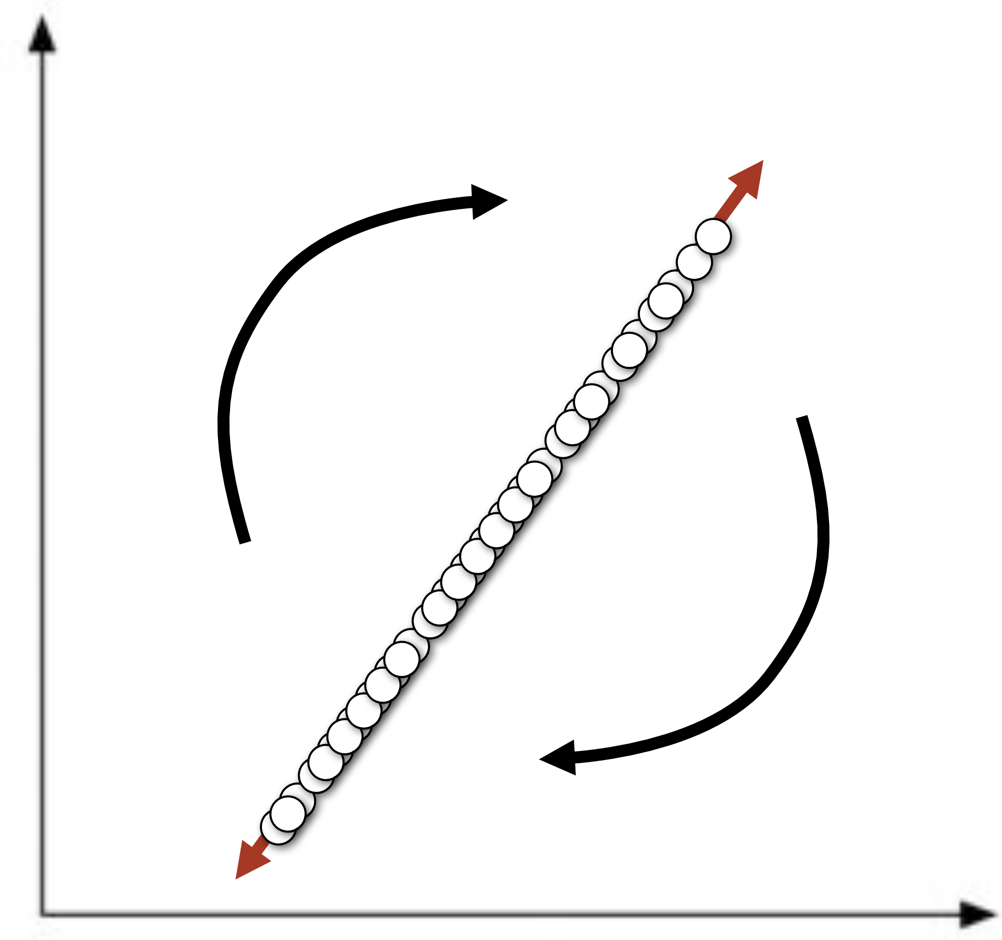
Projection



Projection



Projection



Dimension reduction via projection

- **Big idea**: *transform* the data before performing regression

$$[X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5] \mapsto [Z_1 \quad Z_2]$$

- Then instead of:

$$Y = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \varepsilon$$

we solve:

$$Y = \theta_0 + \sum_{i=1}^{m} \theta_i Z_i + \varepsilon$$

Linear projection

- New features are **linear combinations** of original data:

$$Z_j = \sum_i^m \theta_{ij} X_i$$

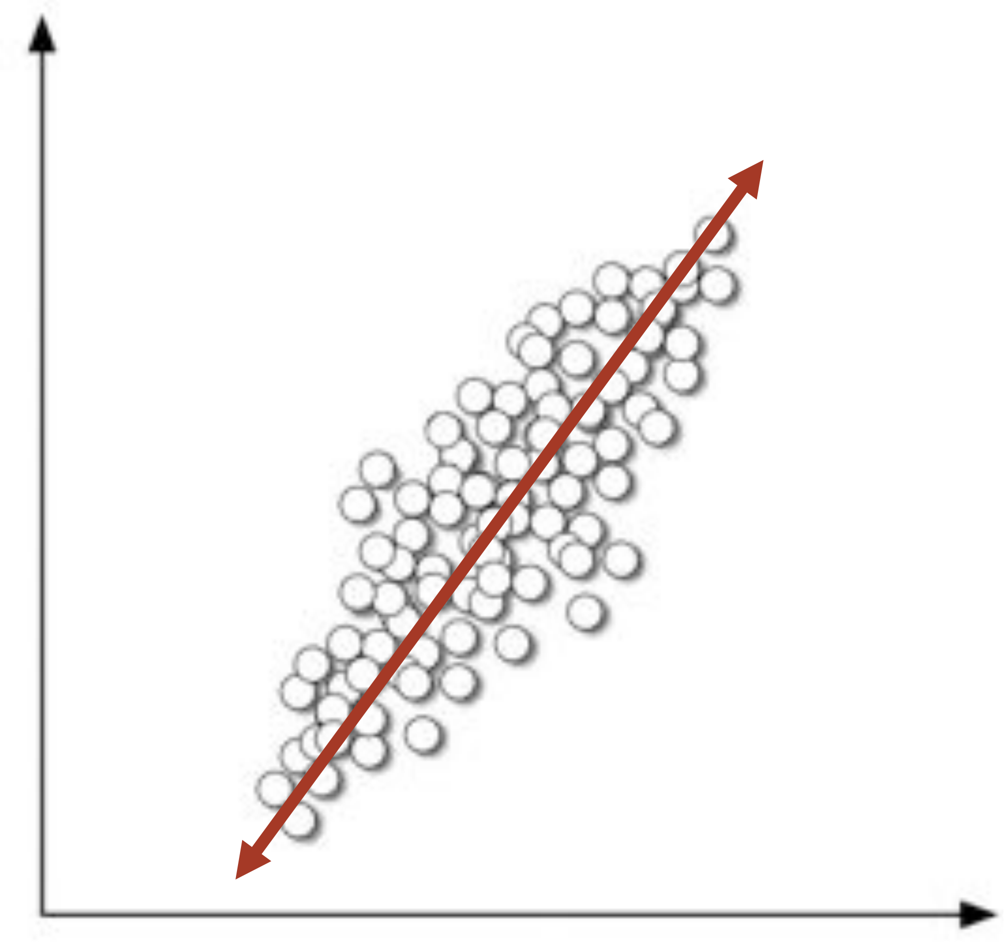
- **MTH211**: multiplying the *data matrix* by a *projection matrix*

$$[Z_1 \quad Z_2] = [X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5] \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} \\ \varphi_{2,1} & \varphi_{2,2} \\ \varphi_{3,1} & \varphi_{3,2} \\ \varphi_{4,1} & \varphi_{4,2} \\ \varphi_{5,1} & \varphi_{5,2} \end{bmatrix}$$

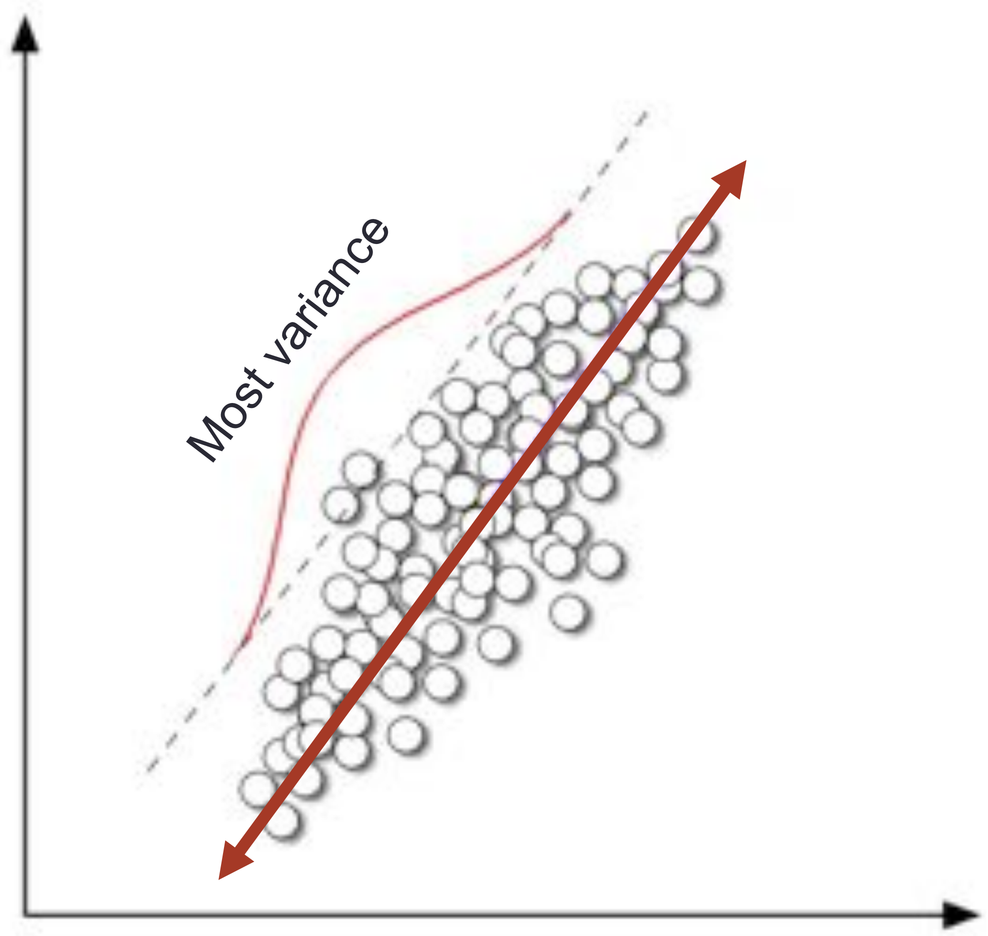
What’s the deal with projection?

- Data can be rotated, scaled, and translated without changing the **underlying relationships**
- This means you’re allowed to look at the data from whatever angle makes your life easier…

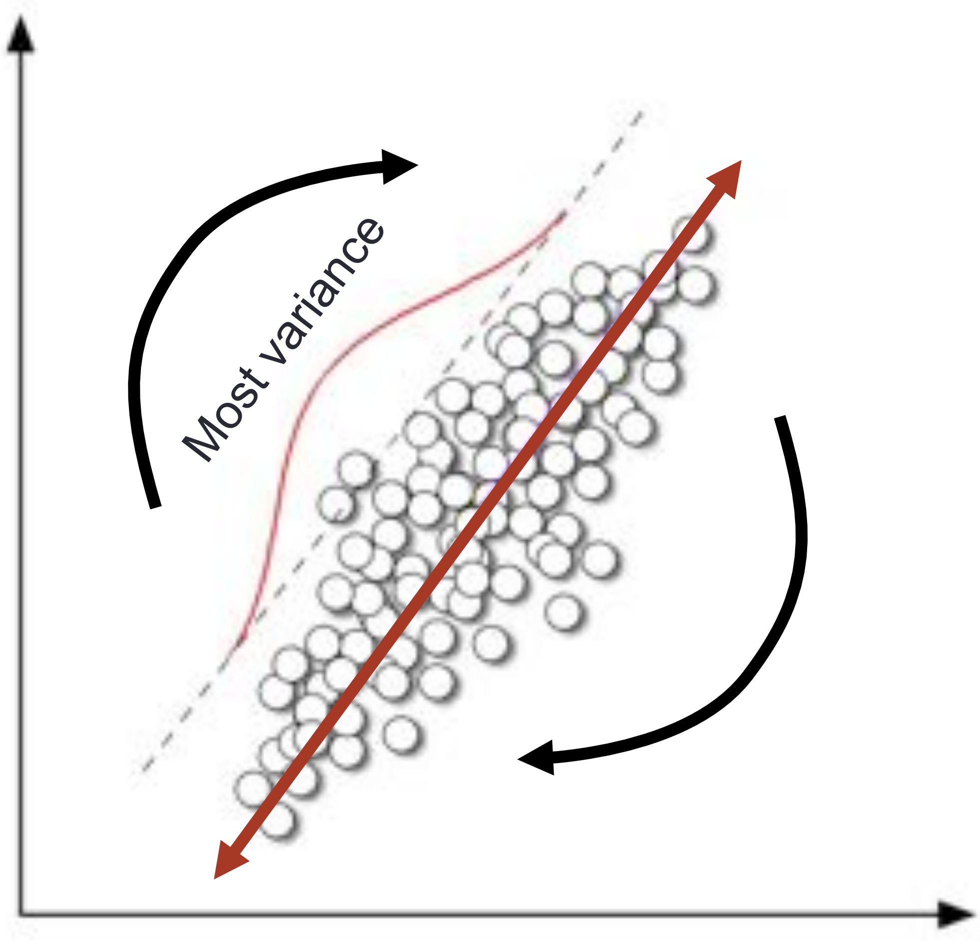
Flashback: why did we pick this line?



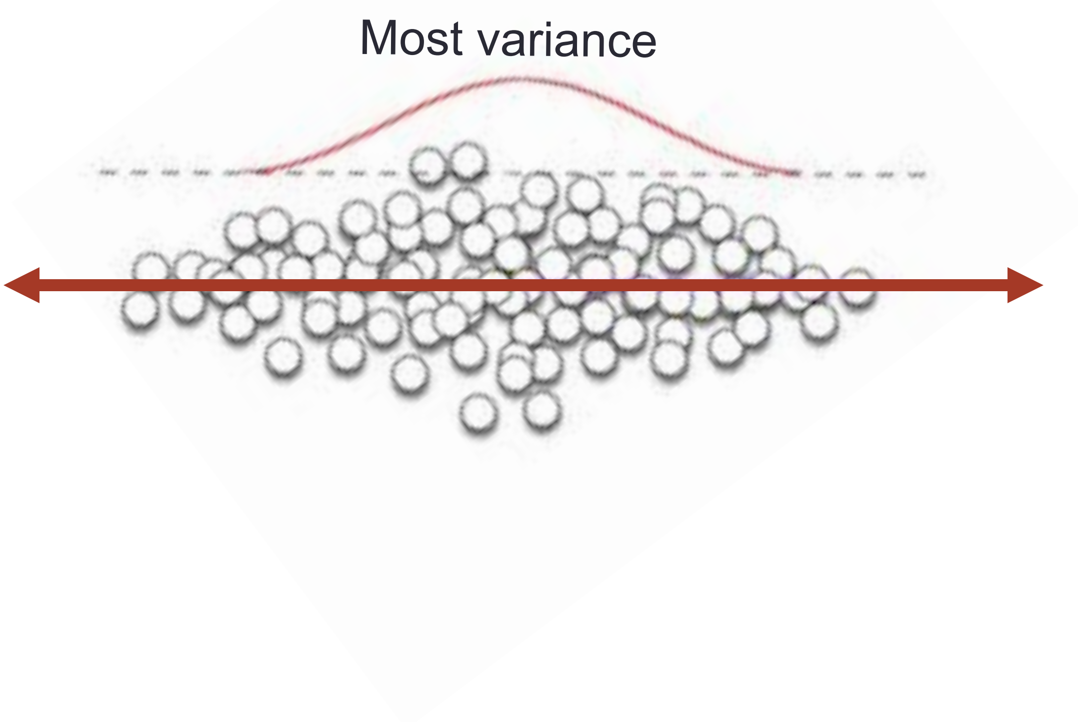
Explains the most **variance** in the data



Imagine this line as a new dimension…



“Principal component”



Mathematically

- The **1st principal component** is the normalized* linear combination of features:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p$$

that has the largest variance

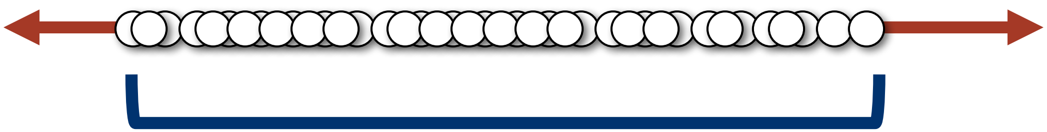
- $\phi_{11}, \ldots, \phi_{p1}$: the **loadings** of the 1st principal component

* By **normalized** we mean: $\sum_{j=1}^{p} \phi_{j1}^2 = 1$

Using loadings to project

Multiply by loading vector to project ("smoosh") each observation onto the line:

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



These values are called the **scores** of the 1st principal component

Additional principal components

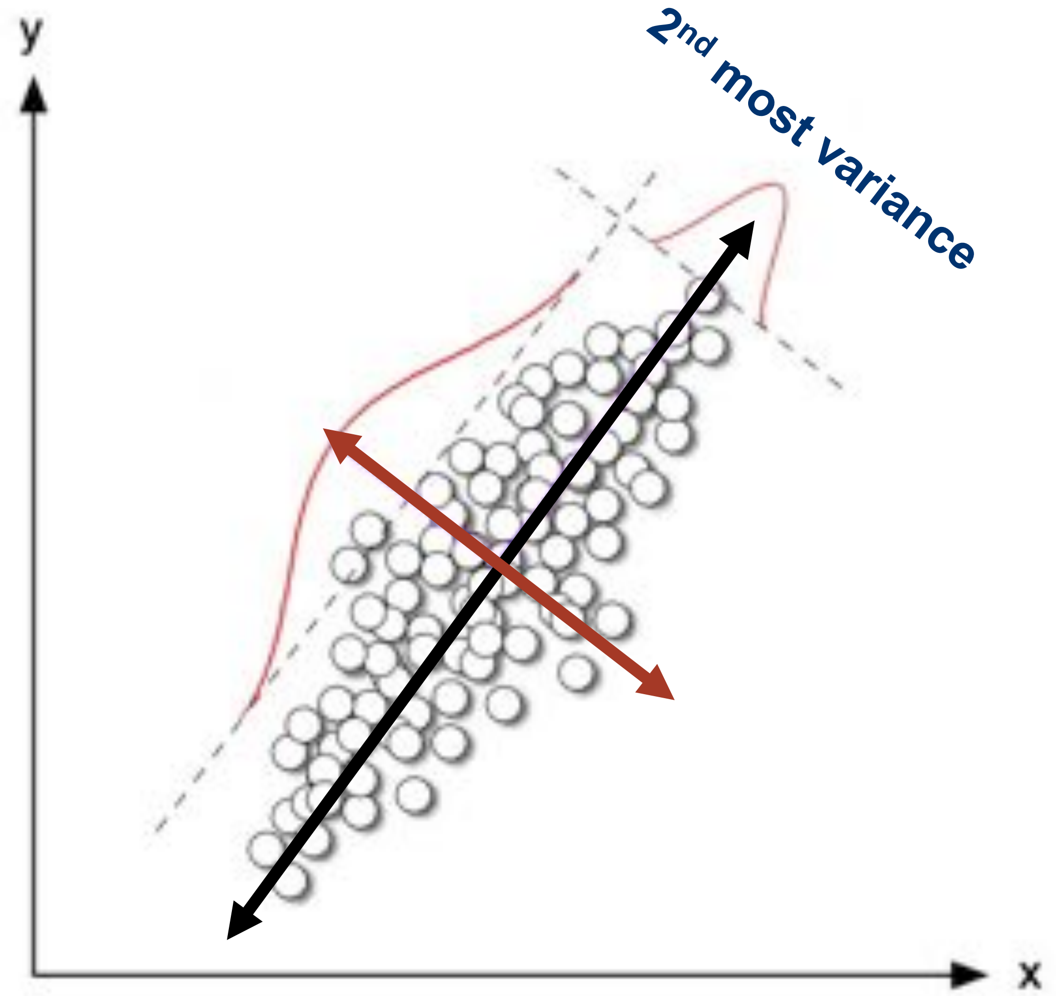
- **2nd principal component** is the normalized linear combination of the features

$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \cdots + \phi_{p2}X_p$$

that has maximal variance out of all linear combinations that are **uncorrelated** with Z_1 (why does that matter?)

- Fun fact:

Principal components are orthogonal



Generating additional principal components

- We can think of this recursively
- To find the M^{th} principal component . . .
  - Find the first $(M - 1)$ principal components
  - Subtract the projection into that space
  - Maximize the variance in the remaining *complementary* space

Regression in the principal components

- **Original objective**: solve for β in

$$Y = \beta_0 + \sum_{i}^{p} \beta_i X_i + \varepsilon$$

(that's still our goal)

- Now we're going to work in the new feature space:

$$Y = \theta_0 + \sum_{i}^{M} \theta_i Z_i + \varepsilon$$

Regression in the principal components

- *Remember*: the new features are **related** to the old ones:

$$Z_j = \sum_{i=1}^{p} \phi_{ij} X_i$$

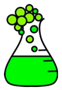
- So we're computing:

$$Y = \theta_0 + \sum_{j=1}^{M} \theta_j Z_j + \varepsilon$$

$$= \theta_0 + \sum_{j=1}^{M} \theta_j \sum_{i=1}^{p} \phi_{ij} X_i + \varepsilon$$

$$\mapsto \beta_i = \sum_{j=1}^{M} \theta_j \phi_{ij}$$

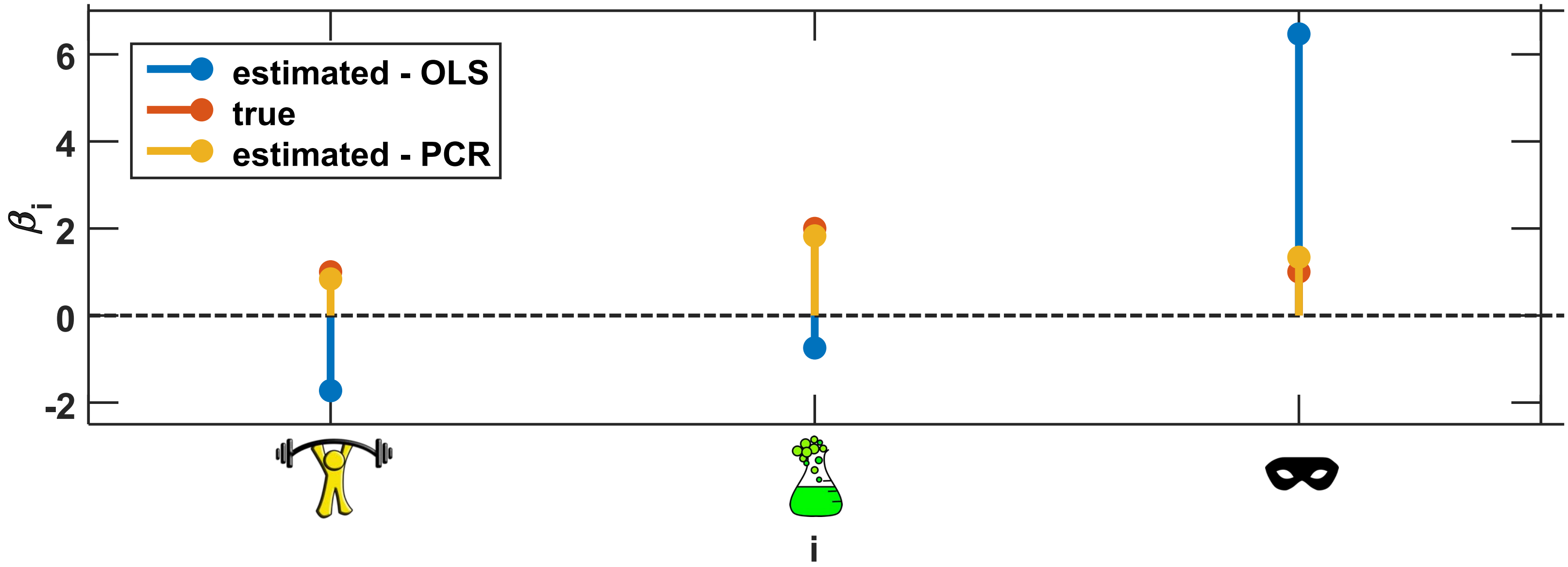
Back to the Guardians

$$\begin{bmatrix} 232.03 \\ 156.29 \\ 113.82 \\ 229.07 \\ 287.72 \end{bmatrix} = 1\begin{bmatrix} 63.9 \\ 28.9 \\ 54.3 \\ 69.8 \\ 50.4 \end{bmatrix} + 2\begin{bmatrix} 54.0 \\ 45.1 \\ 13.3 \\ 49.5 \\ 85.4 \end{bmatrix} + 1\begin{bmatrix} 59.1 \\ 36.9 \\ 33.7 \\ 59.7 \\ 67.9 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

Back to the Guardians

- What happens if we use 2 components instead of 3?



Using only the principal components significantly improves our estimate!

Comparison with ridge regression and the lasso

- What similarities do you see?
  - Reduces dimensionality of the solution space (like Lasso)
  - Finds a solution in the space of all features (like RR)
  - Results can be difficult to interpret (like RR)

Problems with PCR

- We selected principal components based on predictors (not what we're trying to predict!)
- This could be problematic (why?)
  - What if the values you're trying to predict aren't correlated with the first few components?
  - You lose all predictive power!

Partial least squares (PLS)

- A *supervised* form of PCR
- Feature derivation algorithm is similar:
  - Find the (*M*-1) ~~principal~~ **most correlated** components
  - Subtract the projection into that space
  - Maximize the ~~variance~~ **correlation with the response** in the remaining *complementary* space
- As before, we perform least squares on the new features
- We still use the formulation

$$Z_j = \sum_{i=1}^{p} \phi_{ij} X_i$$

- But now ϕ is computed by applying linear regression to *each* predictor

Wrapping up: PCR/PLS comparison

- Both derive a small number of orthogonal predictors for linear regression
- PCR is more biased
  - Emphasizes stability at the expense of versatility
- PLS estimates have higher variance
  - May build new features that aren’t as stable
  - But high variance is better than infinite variance

Lab: PCR and PLS

- To do today’s lab in R: **`pls`**
- To do today’s lab in python: <nothing new>
- Instructions and code:

[course website]/labs/lab11-r.html

[course website]/labs/lab11-py.html

- Full version can be found beginning on p. 256 of ISLR

Flashback: superheroes

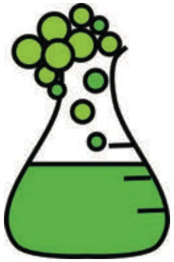
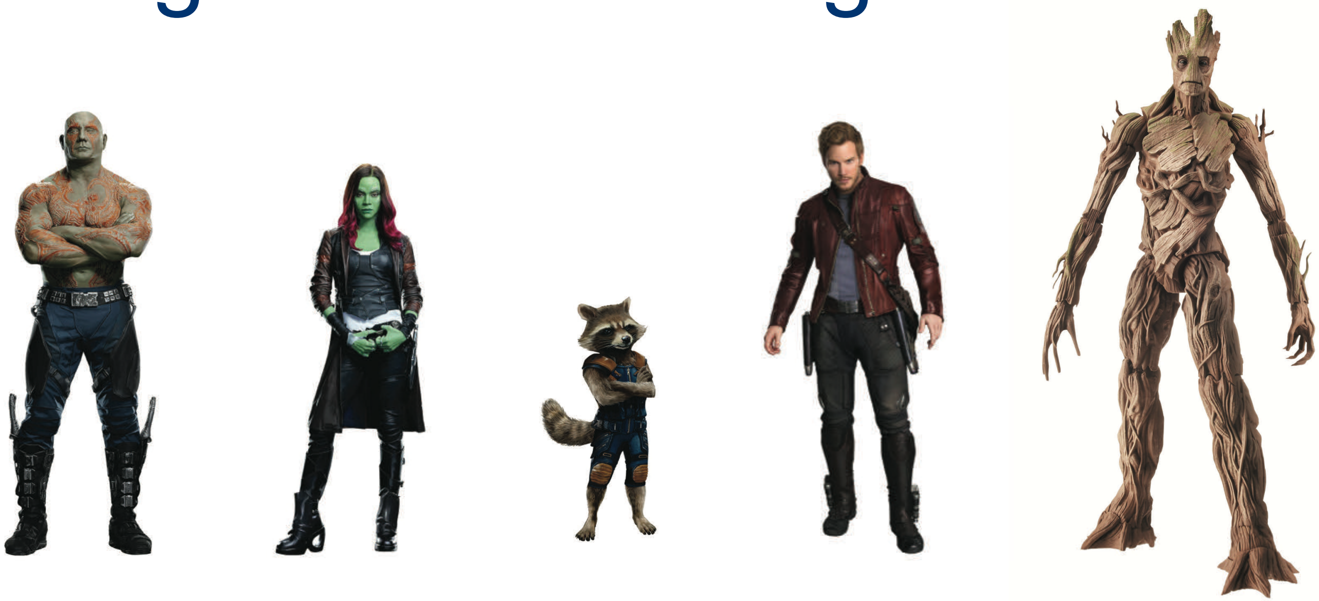
$height = \beta_1$ () $+\beta_2$ () $+\beta_3$ ()

Image credit: Ming Malaykham

Estimating Guardians' Height

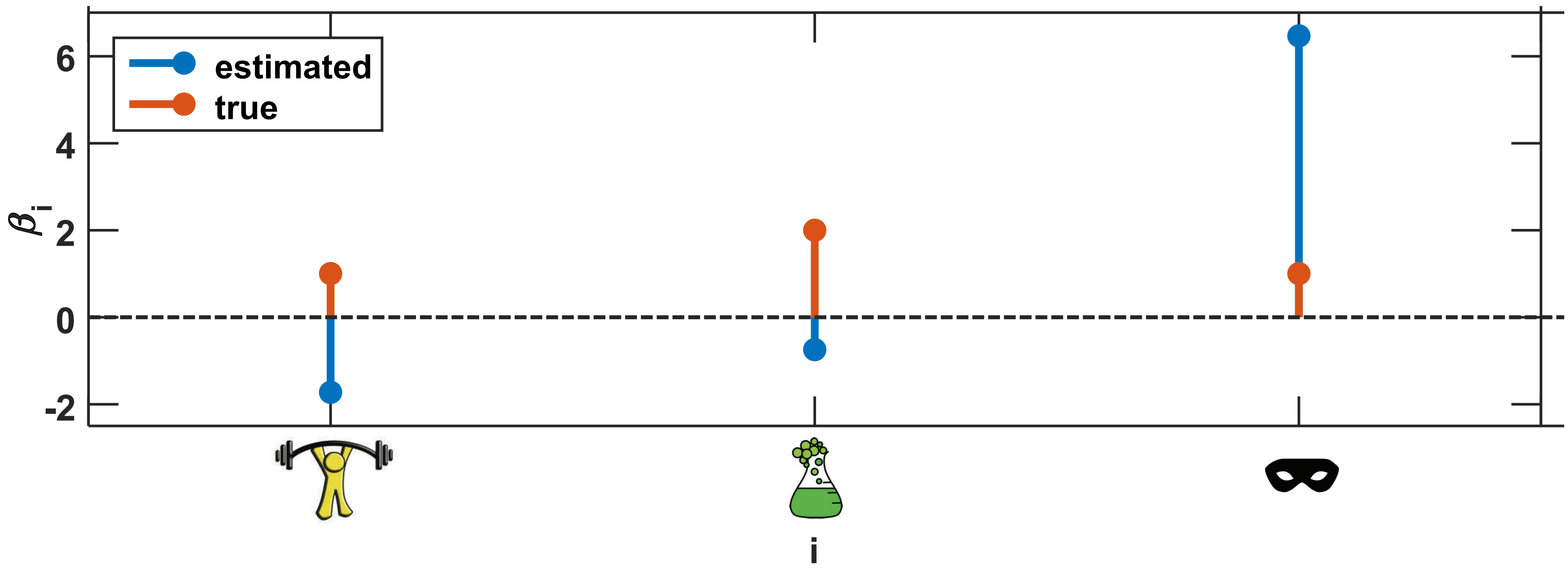


$$\begin{bmatrix} 232.03 \\ 156.29 \\ 113.82 \\ 229.07 \\ 287.72 \end{bmatrix} = 1\begin{bmatrix} 63.9 \\ 28.9 \\ 54.3 \\ 69.8 \\ 50.4 \end{bmatrix} + 2\begin{bmatrix} 54.0 \\ 45.1 \\ 13.3 \\ 49.5 \\ 85.4 \end{bmatrix} + 1\begin{bmatrix} 59.1 \\ 36.9 \\ 33.7 \\ 59.7 \\ 67.9 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

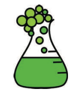
Estimate for β

- When we try to estimate using OLS, we get the following:



(Relatively) huge difference between actual and estimated coefficients

What's going on here?

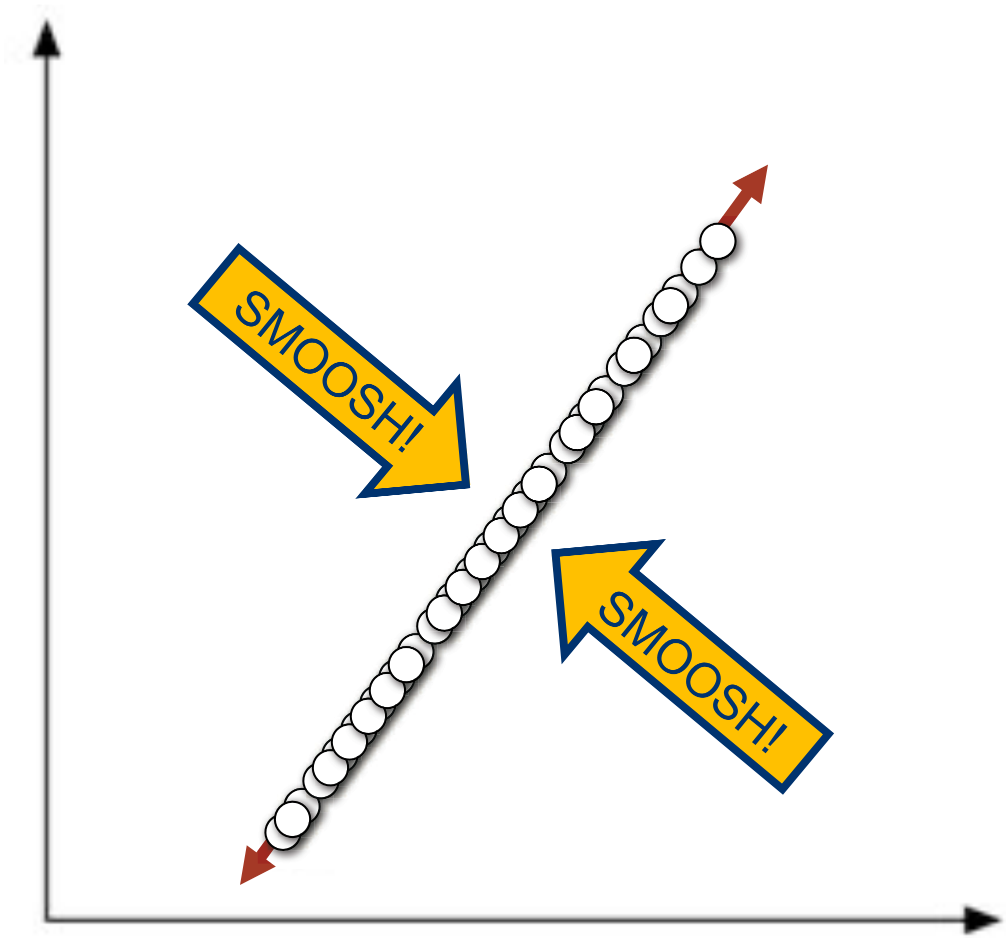
  

$$\begin{bmatrix} 232.03 \\ 156.29 \\ 113.82 \\ 229.07 \\ 287.72 \end{bmatrix} = 1 \begin{bmatrix} 63.9 \\ 28.9 \\ 54.3 \\ 69.8 \\ 50.4 \end{bmatrix} + 2 \begin{bmatrix} 54.0 \\ 45.1 \\ 13.3 \\ 49.5 \\ 85.4 \end{bmatrix} + 1 \begin{bmatrix} 59.1 \\ 36.9 \\ 33.7 \\ 59.7 \\ 67.9 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}$$

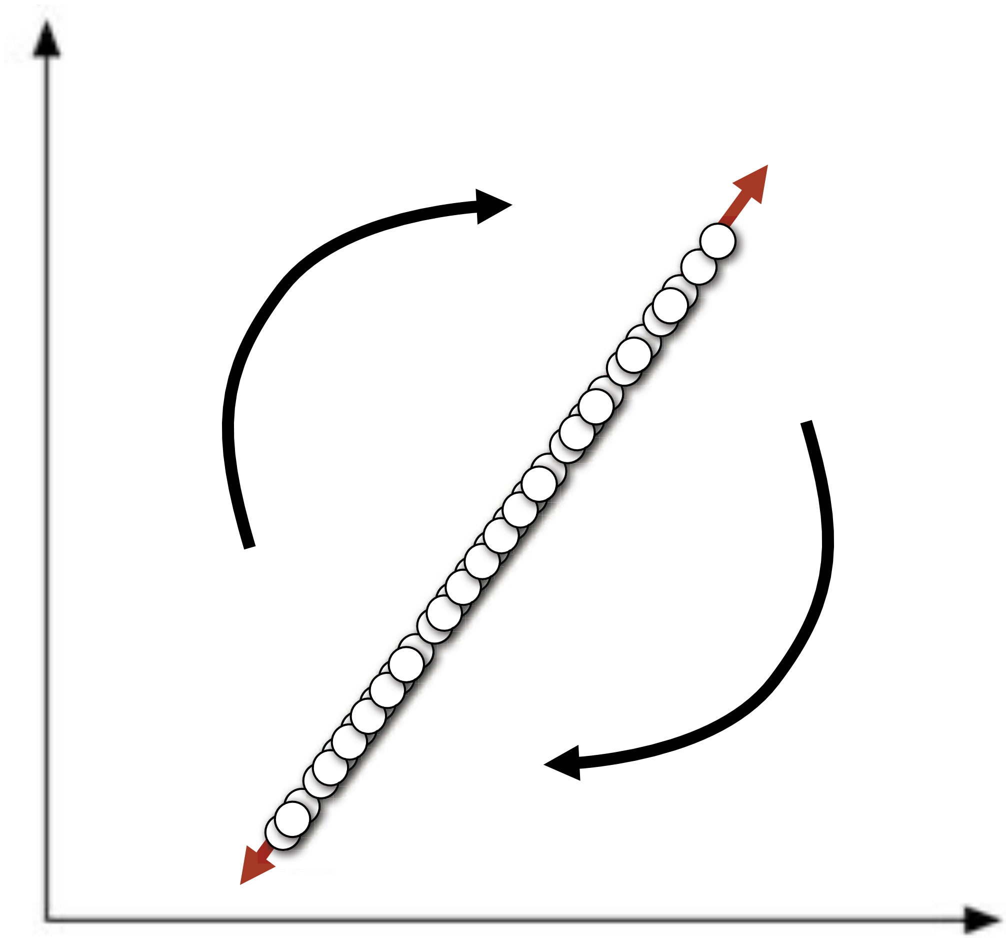
$\approx avg\Big($  ,  $\Big)$

- Some dimensions are redundant
  - Little information in 3rd dimension not captured by the first two
  - In linear regression, redundancy causes noise to be **amplified**

Projection



Projection



Linear projection

- New features are **linear combinations** of original data:

$$Z_j = \sum_{i}^{m} \theta_{ij} X_i$$

- **MTH211**: multiplying the *data matrix* by a *projection matrix*

$$[Z_1 \quad Z_2] = [X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5] \begin{bmatrix} \varphi_{1,1} & \varphi_{1,2} \\ \varphi_{2,1} & \varphi_{2,2} \\ \varphi_{3,1} & \varphi_{3,2} \\ \varphi_{4,1} & \varphi_{4,2} \\ \varphi_{5,1} & \varphi_{5,2} \end{bmatrix}$$

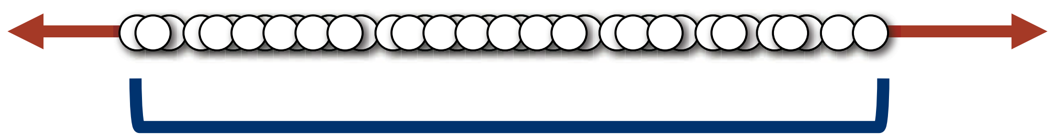
What's the deal with projection?

- Data can be rotated, scaled, and translated without changing the **underlying relationships**
- This means you're allowed to look at the data from whatever angle makes your life easier…

Using loadings to project

Multiply by loading vector to project ("smoosh")
each observation onto the line:

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



These values are called the **scores**
of the 1st principal component

Regression in the principal components

- *Remember*: the new features are **related** to the old ones:

$$Z_j = \sum_{i=1}^{p} \phi_{ij} X_i$$

- So we're computing:

$$Y = \theta_0 + \sum_{j=1}^{M} \theta_j Z_j + \varepsilon$$

$$= \theta_0 + \sum_{j=1}^{M} \theta_j \sum_{i=1}^{p} \phi_{ij} X_i + \varepsilon$$

$$\mapsto \beta_i = \sum_{j=1}^{M} \theta_j \phi_{ij}$$

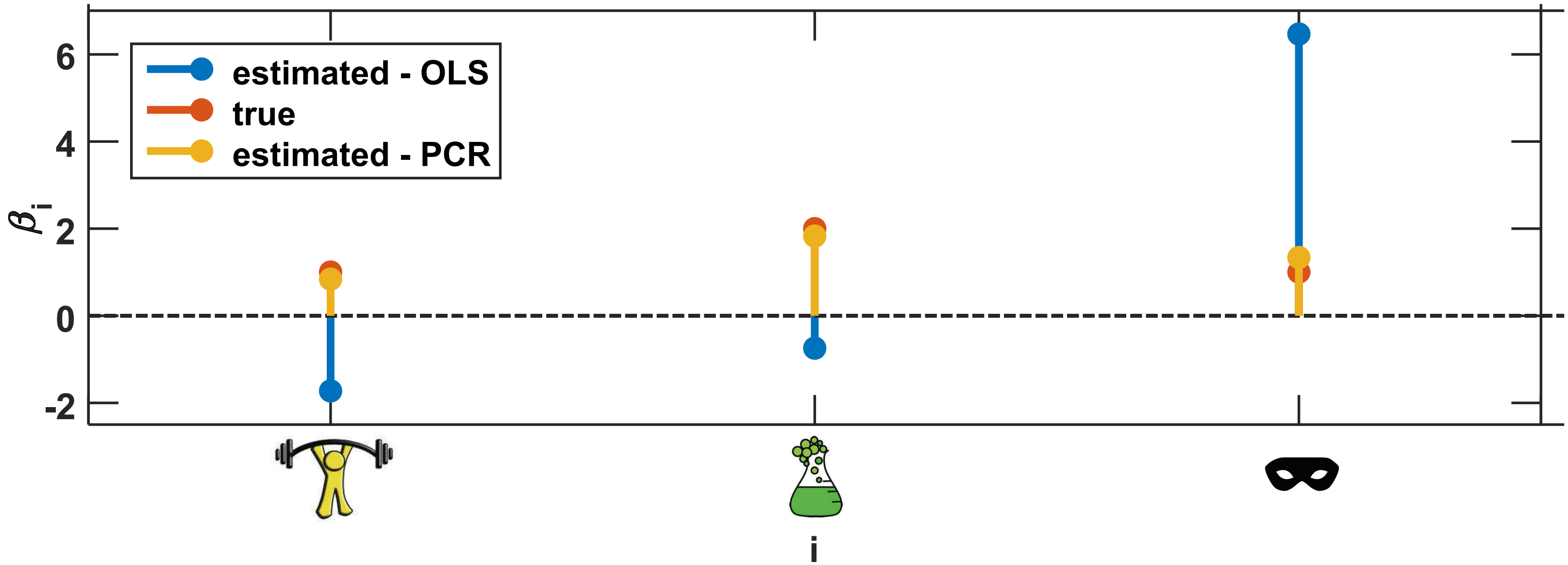
Back to the Guardians

$$
\begin{bmatrix} 232.03 \\ 156.29 \\ 113.82 \\ 229.07 \\ 287.72 \end{bmatrix} = 1 \begin{bmatrix} 63.9 \\ 28.9 \\ 54.3 \\ 69.8 \\ 50.4 \end{bmatrix} + 2 \begin{bmatrix} 54.0 \\ 45.1 \\ 13.3 \\ 49.5 \\ 85.4 \end{bmatrix} + 1 \begin{bmatrix} 59.1 \\ 36.9 \\ 33.7 \\ 59.7 \\ 67.9 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \varepsilon_4 \\ \varepsilon_5 \end{bmatrix}
$$

Back to the Guardians

- What happens if we use 2 components instead of 3?



Using only the principal components significantly improves our estimate!

Comparison with ridge regression and the lasso

- What similarities do you see?
  - Reduces dimensionality of the solution space (like Lasso)
  - Finds a solution in the space of all features (like RR)
  - Results can be difficult to interpret (like RR)

Problems with PCR

- We selected principal components based on predictors (not what we're trying to predict!)
- This could be problematic (why?)
  - What if the values you're trying to predict aren't correlated with the first few components?
  - You lose all predictive power!