Stat 215A - Week 4

Slides on PCA thanks to Rebecca Barter

Lab 1

What was people's experience with the lab?

Any final questions to ask?

Lab 1: peer review

Later today I will push two reports to each of your repositories (plus my report from last year).

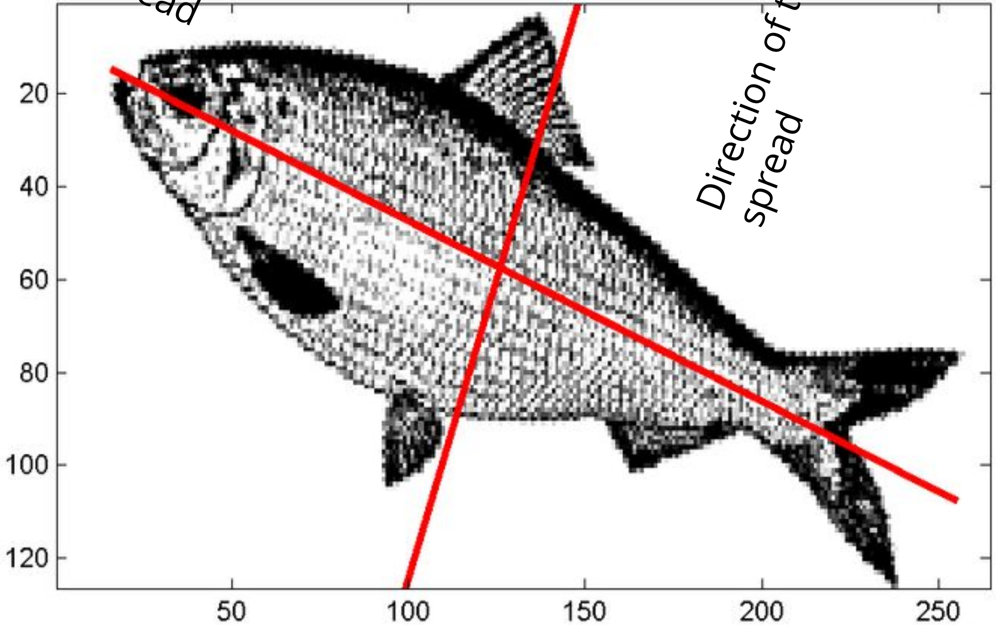
If you do **not have reports to review in your stat-215-a GitHub by the end of the day** that means that I did not get your report and you should **email me ASAP**.

You have **one week to review the report and provide feedback** in the google form that I will distribute.

PCA

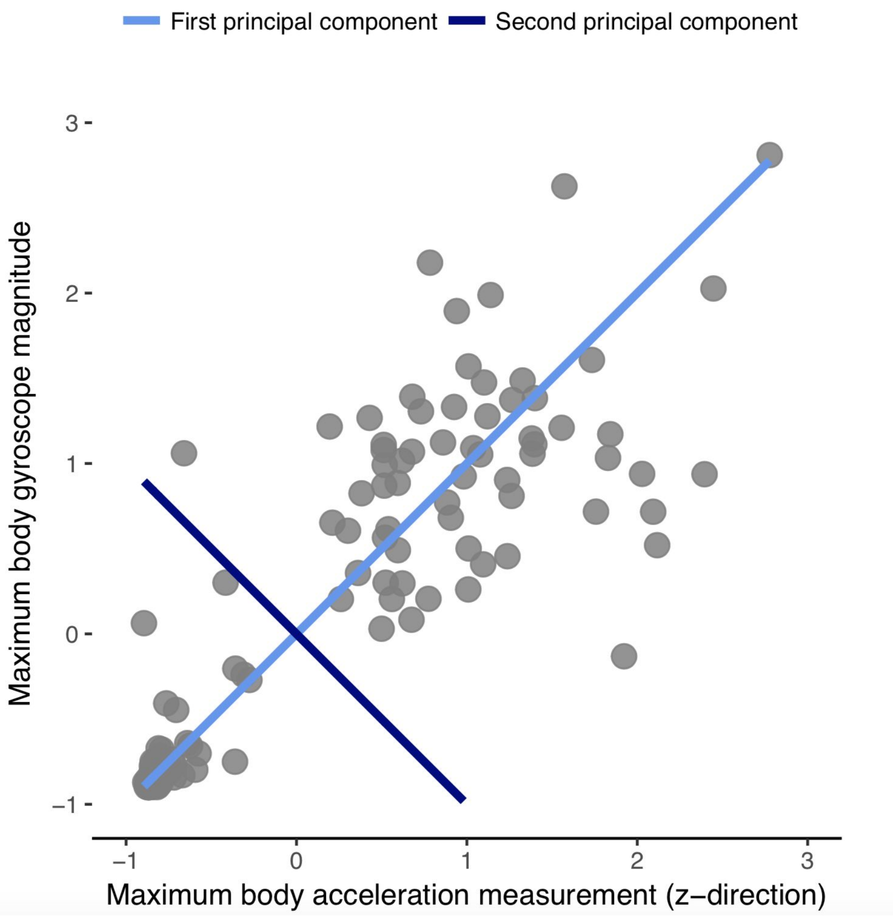
Direction of the most spread

Direction of the second most spread



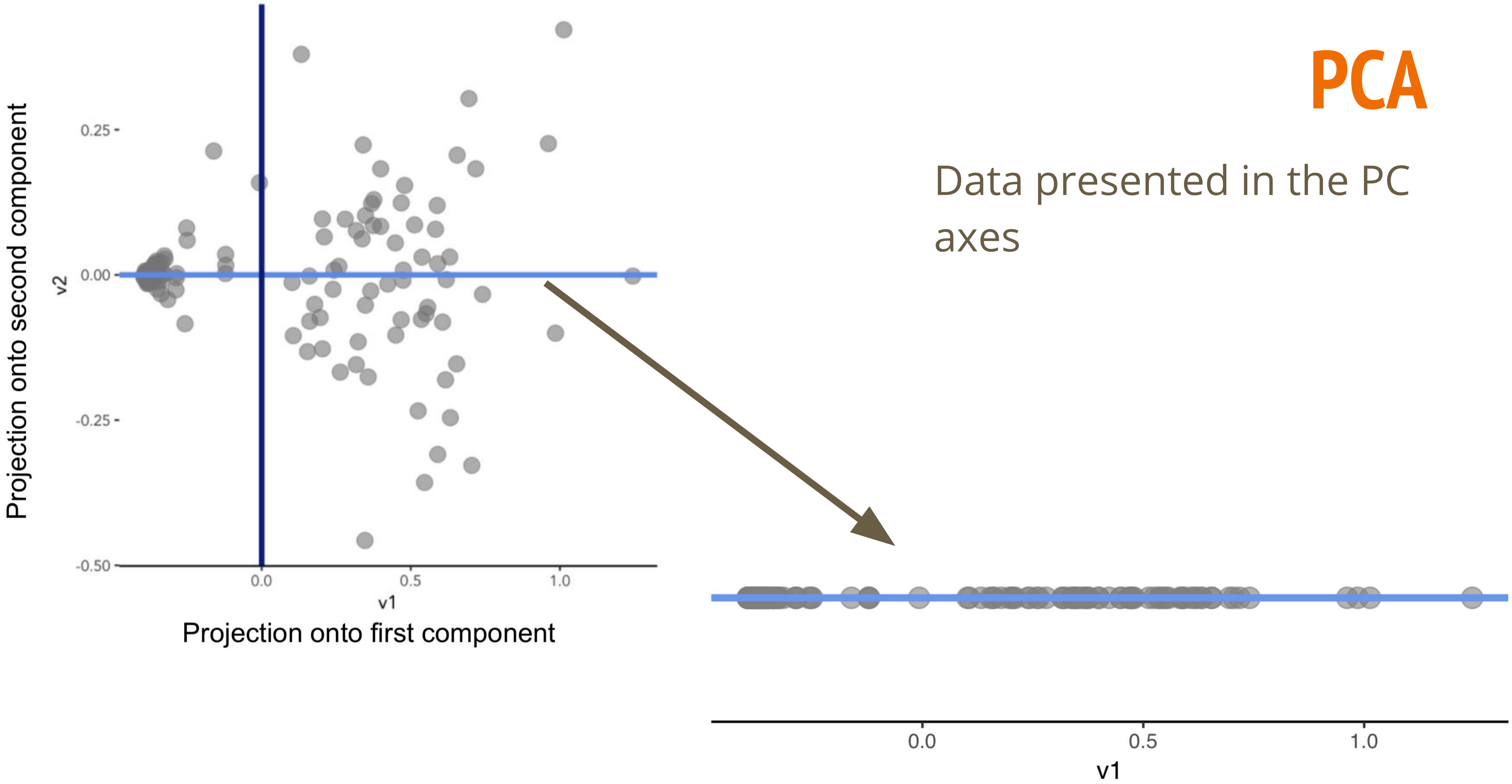
PCA

Data presented with the original axes



PCA

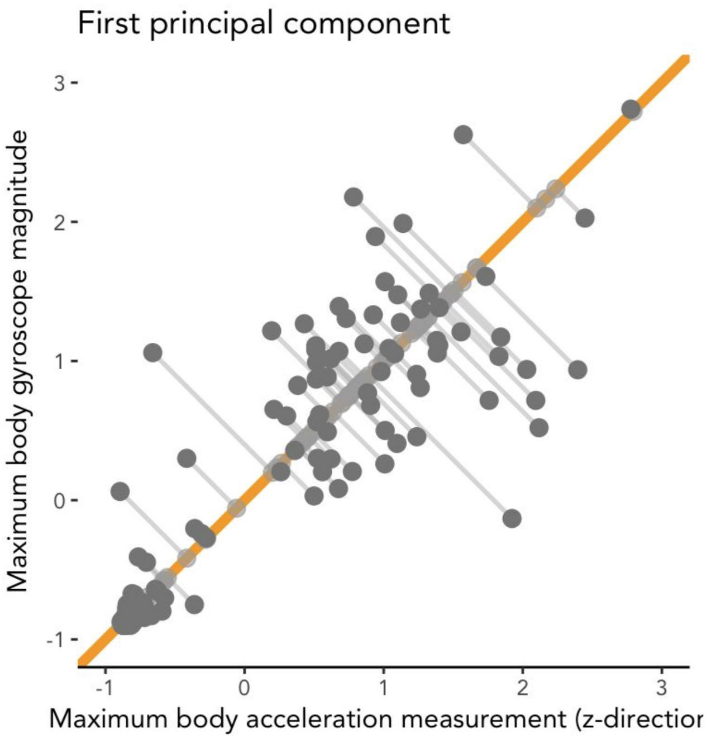
Data presented in the PC axes



Calculating the PCs: first PC

The first PC is the line to which the data have the **smallest average perpendicular distance**

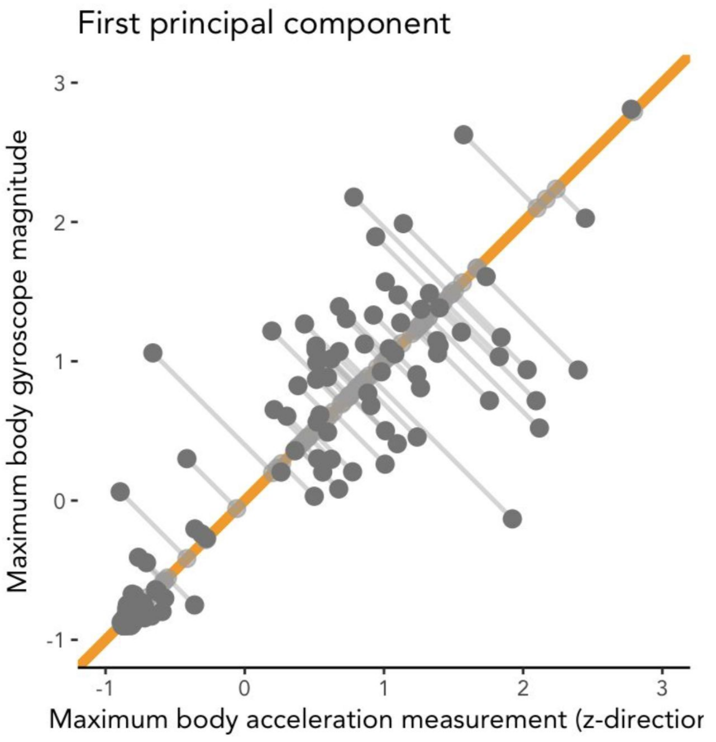
...or, equivalently, the direct along which the data is most spread out



Calculating the PCs

The second PC is the line (perpendicular to the first PC) to which the data have the **next smallest average perpendicular distance.**

In higher dimensions we talk about "**orthogonal**" rather than "perpendicular"



Eigenvectors and Eigenvalues

A vector is a line with a direction and magnitude (length)

Multiplying a vector by a matrix does two things:

1. **Rotate** the vector orientation
2. **Scale** the vector by increasing/decreasing its magnitude

Eigenvectors and Eigenvalues

Eigen**value** of *A*

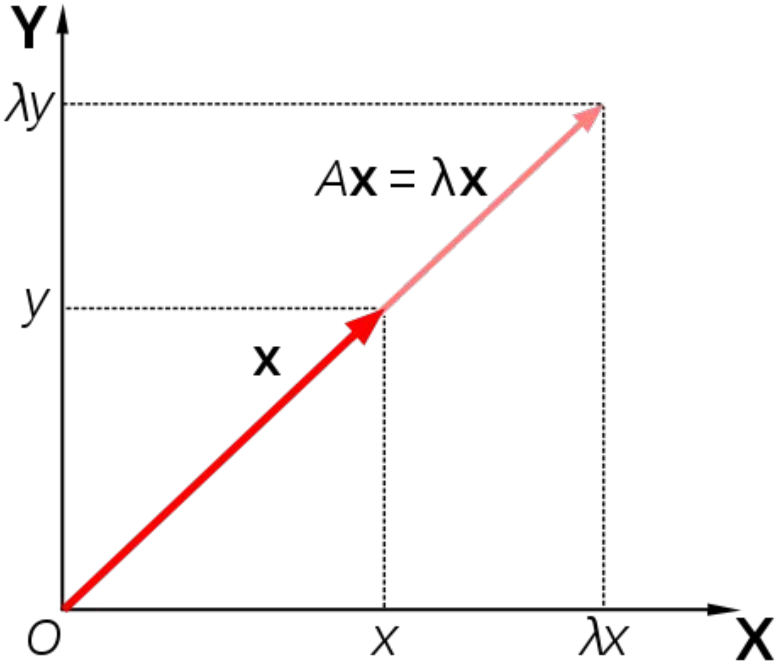
$$A\mathbf{v} = \lambda\mathbf{v}$$

Eigen**vector** of *A*

For an eigenvector ***v*** and an eigenvalue **λ** of ***A***:

Rotating and scaling ***v*** by ***A*** is the same as scaling ***v*** by a scalar **λ**

Eigenvectors and Eigenvalues



I.e. multiplication of an eigenvector of ***A*** by ***A*** does not rotate the vector, it only scales it.

https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors

Eigenvectors and Eigenvalues

Eigenvectors correspond to the “favored” rotation directions of ***A***

https://math.stackexchange.com/questions/243533/how-to-intuitively-understand-eigenvalue-and-eigenvector

Eigenvectors and Eigenvalues

Eigenvectors correspond to the “favored” rotation directions of ***A***.

When ***A*** acts on any vector, it rotates it in the direction of these favored orientations.

https://math.stackexchange.com/questions/243533/how-to-intuitively-understand-eigenvalue-and-eigenvector

Eigenvectors and Eigenvalues

Eigenvectors correspond to the “favored” rotation directions of ***A***.

When ***A*** acts on any vector, it rotates it in the direction of these favored orientations.

When ***A*** is applied to a vector that is already in the favored orientation (an **eigenvector**), it does not rotate, but instead simply scales the vector.

https://math.stackexchange.com/questions/243533/how-to-intuitively-understand-eigenvalue-and-eigenvector

Eigenvectors and Eigenvalues

Eigenvectors correspond to the “favored” rotation directions of ***A***.

When ***A*** acts on any vector, it rotates it in the direction of these favored orientations.

When ***A*** is applied to a vector that is already in the favored orientation (an **eigenvector**), it does not rotate, but instead simply scales the vector.

The amount by which each eigenvector is stretched or compressed is the **eigenvalue**.

https://math.stackexchange.com/questions/243533/how-to-intuitively-understand-eigenvalue-and-eigenvector

Eigendecomposition

It turns out that for any symmetric matrix ***A*** you can factorize it using **eigendecomposition**.

$$A = VDV^T$$

$$D = diag(\lambda_1, \ldots, \lambda_p) = \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ 0 & 0 & \ddots & \lambda_p \end{bmatrix}$$

$$V = \begin{bmatrix} \mathbf{v}_1 & \ldots & \mathbf{v}_p \end{bmatrix} = \begin{bmatrix} v_{11}, & \ldots & v_{p,1} \\ v_{1,2} & \ldots & v_{p,2} \\ \vdots & \vdots & \vdots \\ v_{1,n} & \ldots & v_{p,n} \end{bmatrix}$$

D is a diagonal matrix whose **diagonal entries** are **eigenvalues**

V is a matrix whose **columns** correspond to the **eigenvectors**

Calculating the PCs

The principal components correspond to the **eigenvectors** of the **covariance matrix** of the data

The data can be projected into PC space by multiplying the data by the eigenvector rotation matrix $\boldsymbol{V}$**:**

$$\mathbf{X^*} = \mathbf{XV}$$

Calculating the PCs

The principal components correspond to the **eigenvectors** of the **covariance matrix** of the data

The data can be projected into PC space by multiplying the data by the eigenvector rotation matrix $\boldsymbol{V}$:

$$\mathbf{X}^* = \mathbf{X}\mathbf{V}$$

The eigenvalues correspond to the **proportion of variability explained** by each eigenvector

The PCA algorithm

1. Calculate the covariance matrix of the data

$$G = (X - \overline{X})^T (X - \overline{X}) = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{bmatrix}$$

The PCA algorithm

1. Calculate the covariance matrix of the data

$$G = (X - \overline{X})^T (X - \overline{X}) = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{bmatrix}$$

2. Calculate the eigendecomposition of the covariance matrix

$$G = VDV^T$$

The PCA algorithm

1. Calculate the covariance matrix of the data

$$G = (X - \overline{X})^T(X - \overline{X}) = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{bmatrix}$$

2. Calculate the eigendecomposition of the covariance matrix

$$G = VDV^T$$

3. The first PC is the first column of V and captures $\lambda_1/(\sum_{j=1}^{p} \lambda_j)$ of the total variability in the data

The PCA algorithm

1. Calculate the covariance matrix of the data

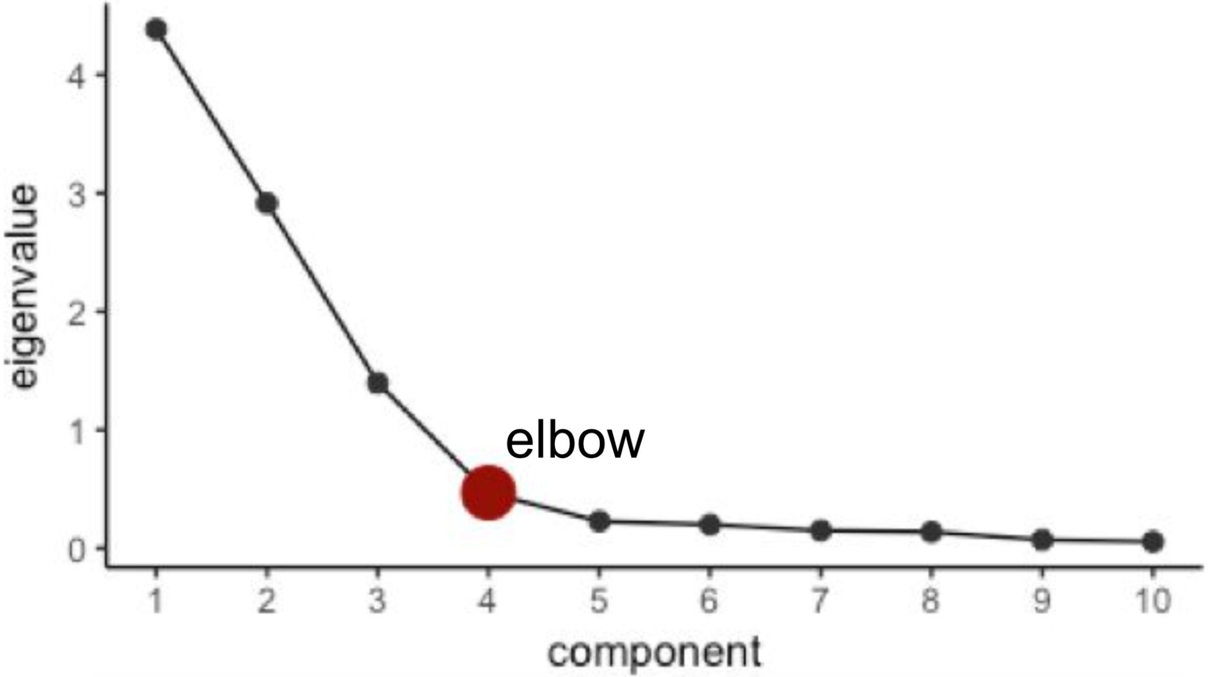
$$G = (X - \overline{X})^T (X - \overline{X}) = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{bmatrix}$$

2. Calculate the eigendecomposition of the covariance matrix

$$G = VDV^T$$

3. The first PC is the first column of V and captures $\lambda_1/(\sum_{j=1}^{p} \lambda_j)$ of the total variability in the data
4. Define a “new” lover-dimensional dataset consisting only of the data projected onto the first few PCs, that account for most of the variation in the data

Scree plot



The PCA algorithm

We can use the output from PCA to answer the question of which of the original variables are most "important"

Do so by calculating the correlation (or "loading") between each variable and the data projected onto the first few PC's

The loading for variable j on principal component i corresponds to the jth entry in the ith eigenvector

Do exercises in pca_exercises.Rmd file