STAT 215A Fall 2017 Week 4

Rebecca Barter
09/15/2017

Lab 1 Conclusion

What did you find difficult about the lab?

What weird things did you find in the data?

Concluding questions about the lab?

Lab 1 Conclusion: peer review

Later today: I will push a single report to review in to each of your repos… remember to pull!

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

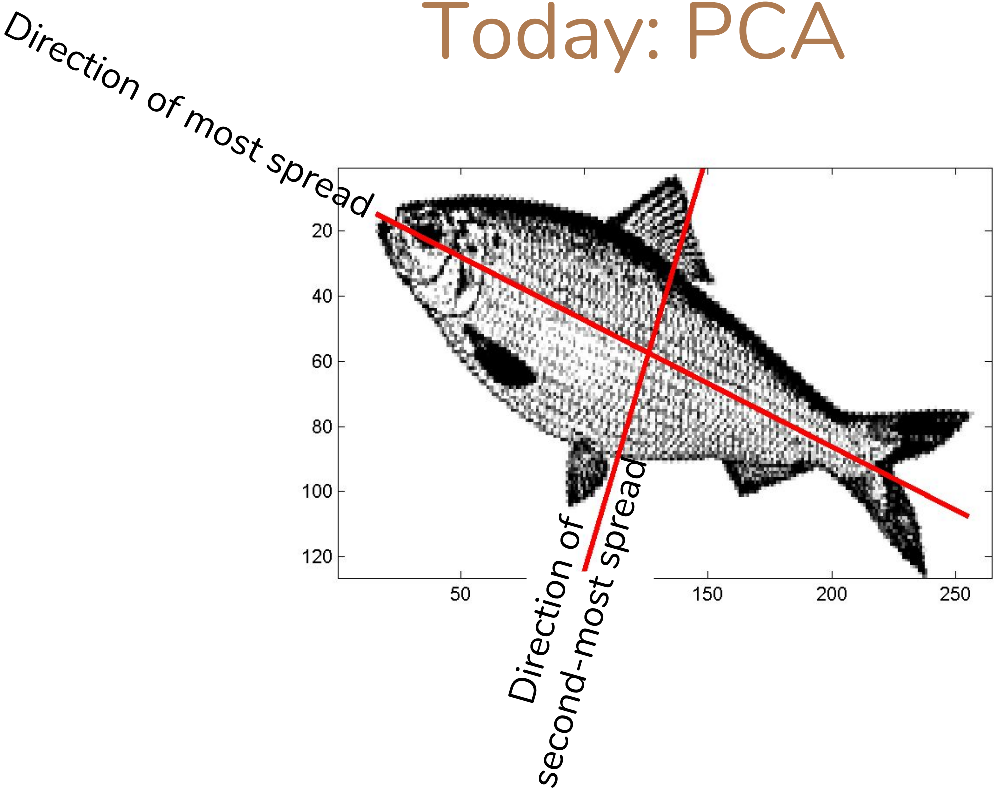
I will grade each of your reports individually (this time!).

Staying on top of recent developments in R

RStudio Blog

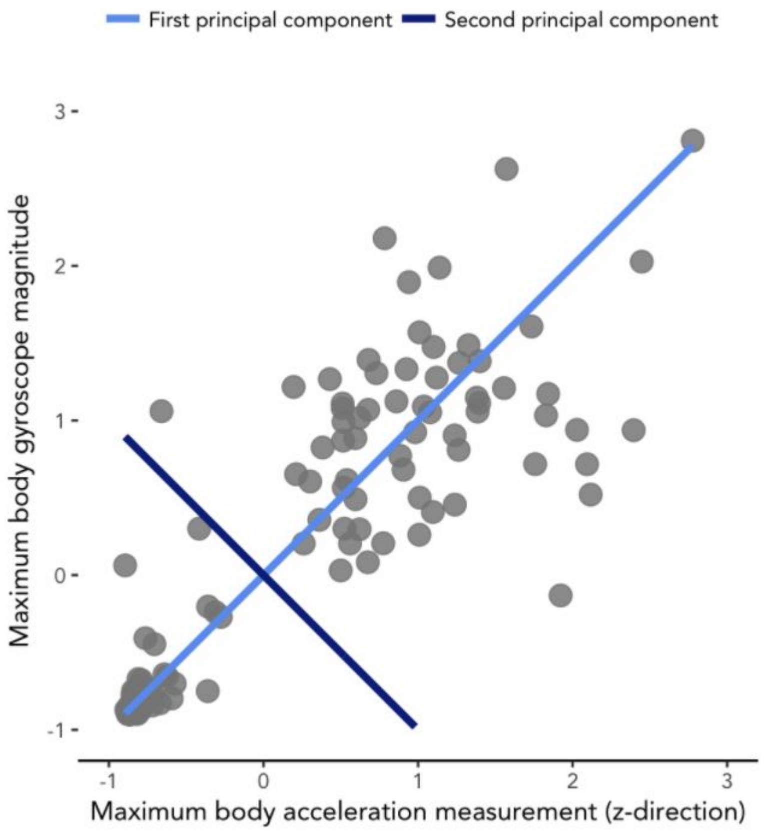
Add to your RSS feed: https://blog.rstudio.com/

Today: PCA


Direction of most spread
Direction of
second-most spread
20
40
60
80
100
120
50
150
200
250

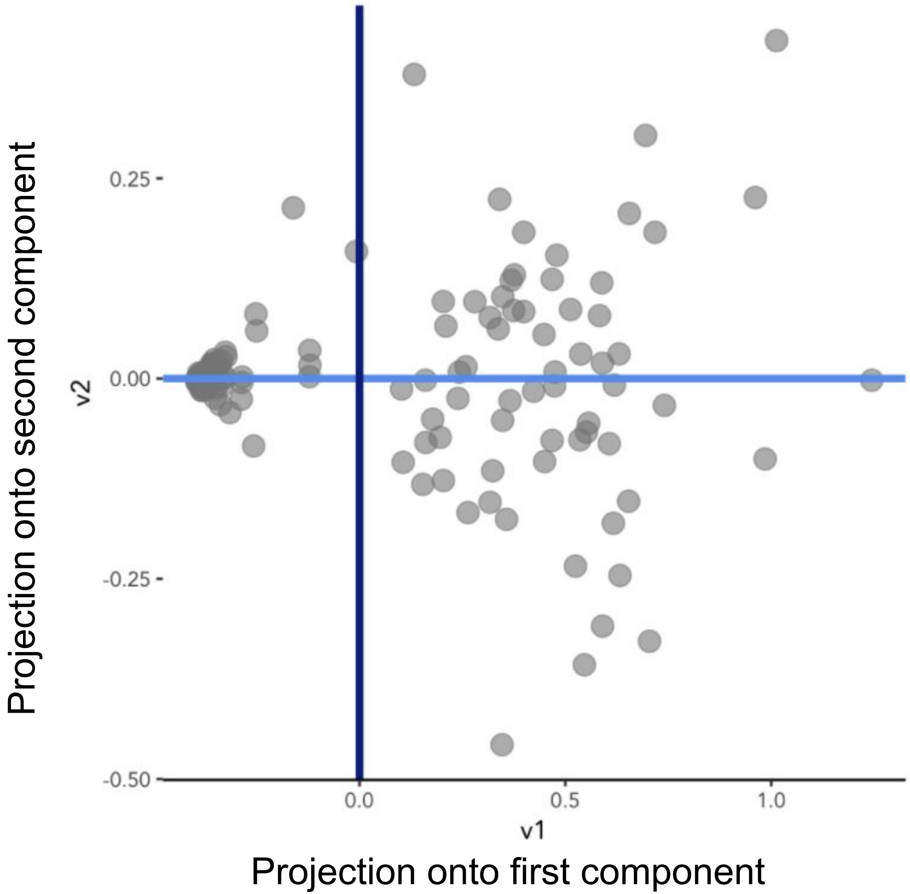
PCA

Data presented with original axes



PCA

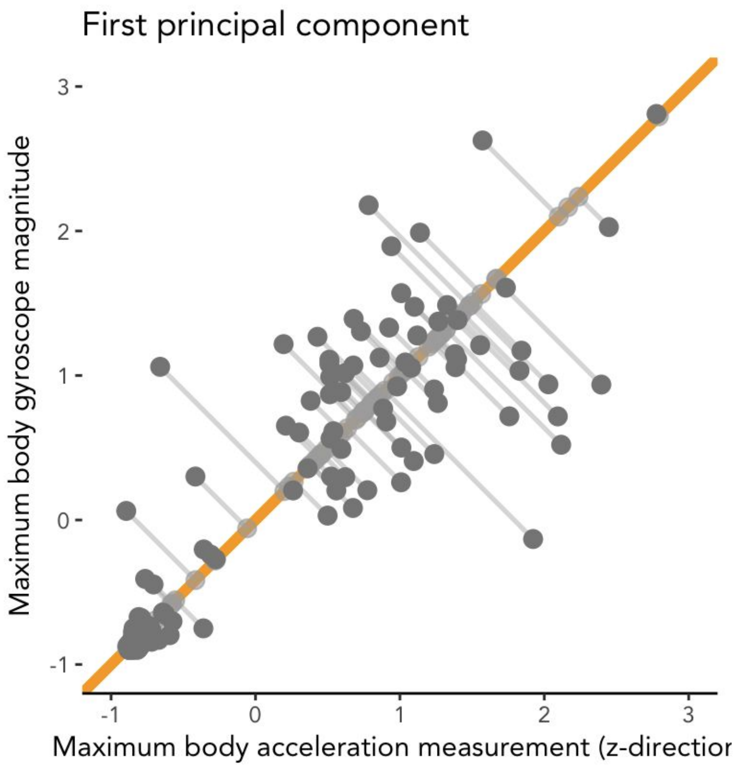
Data presented with PC axes



Calculating the PCs: first PC

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

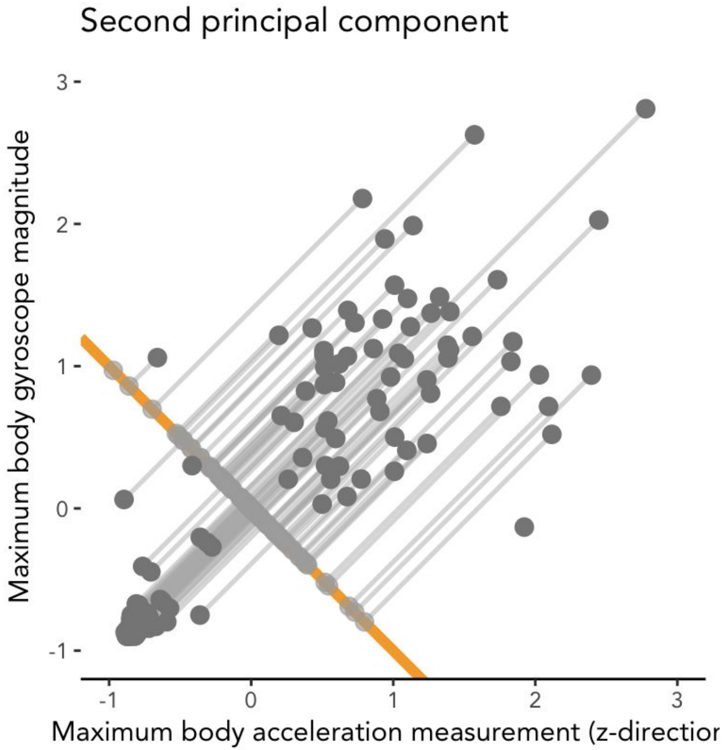
(or equivalently the direction along which the data is most spread out)



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:

- **Rotate** the vector orientation
- **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*

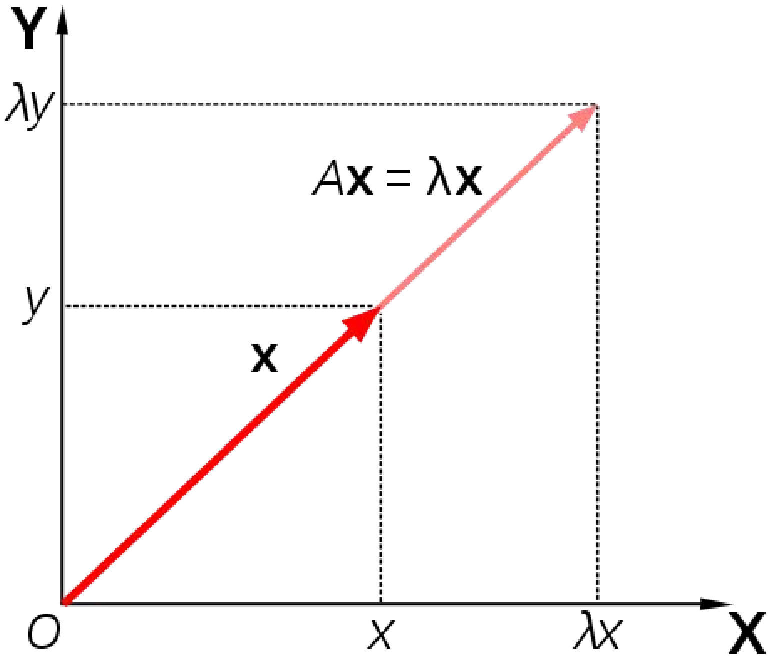
For an eigenvector $\boldsymbol{v}$ and an eigenvalue $\boldsymbol{\lambda}$ of $\boldsymbol{A}$:

Rotating/Scaling v by A

is the same as

scaling v by scalar λ

Eigenvectors and Eigenvalues



i.e. mutiplication of an eigenvector of A by A itself does not rotate the eigenvector; it only scales it!

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

Eigenvectors and Eigenvalues

Eigenvectors correspond to the "*favoured*" rotation directions of A.

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

Eigenvectors and Eigenvalues

Eigenvectors correspond to the "*favoured*" rotation directions of A.

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

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

Eigenvectors and Eigenvalues

Eigenvectors correspond to the "*favoured*" rotation directions of A.

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

When A is applied to a vector that is already in the favoured 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 “*favoured*” rotation directions of A.

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

When A is applied to a vector that is already in the favoured 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**:

$$A = VDV^T$$

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

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

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

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, V:

$$X^* = XV$$

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, V:

$$X^* = XV$$

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

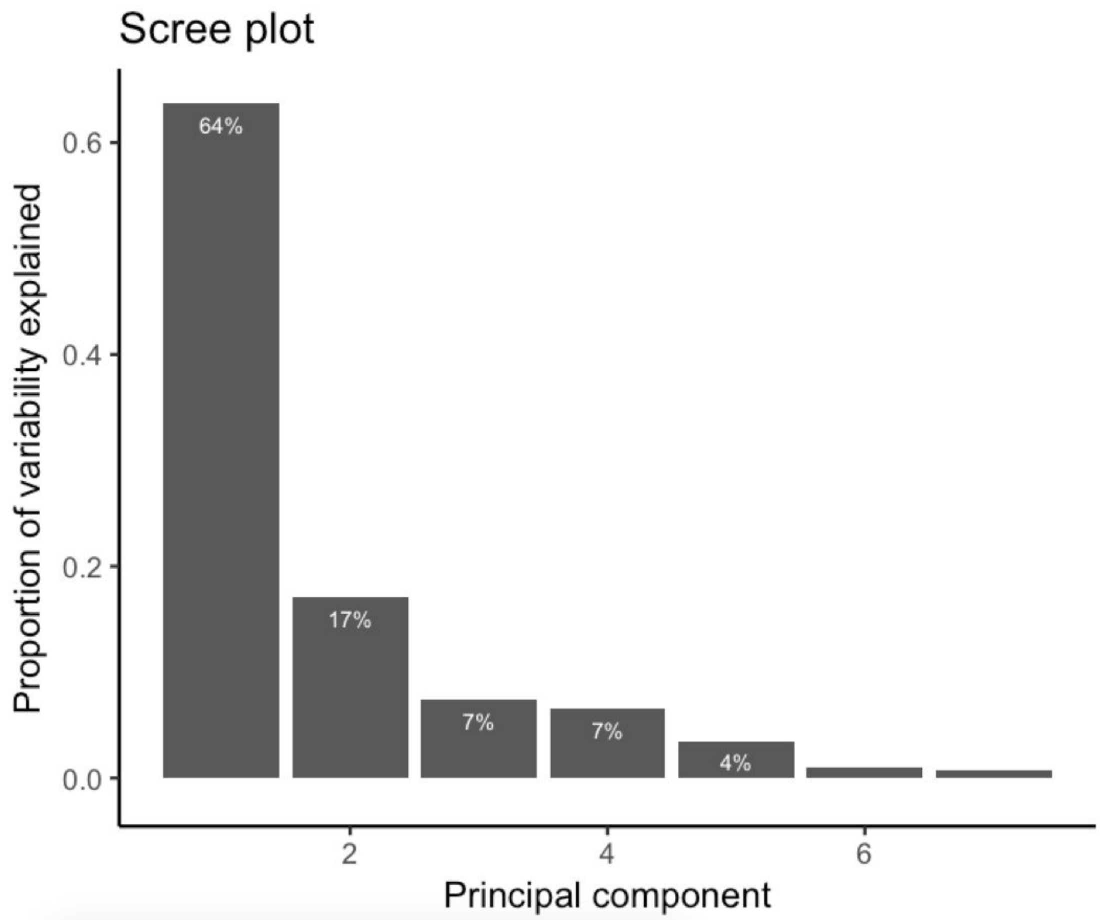
$$G = (X - \overline{X})^T (X - \overline{X}) = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_2, X_1) & Var(X_2) \end{bmatrix}$$

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” lower-dim dataset consisting only of the data projected onto the first few PCs ($XV_{1:3}$) that account for most of the variation in the data (look for the “elbow” in the scree plot)

(Standardized) scree plot



The PCA algorithm

<u>Q: Which of our original variables are the most "important"?</u>

A: Calculate the correlation (the "loading") between each variable and the data projected onto the first few PCs.

Complete the exercises found in
pca_exercises.Rmd