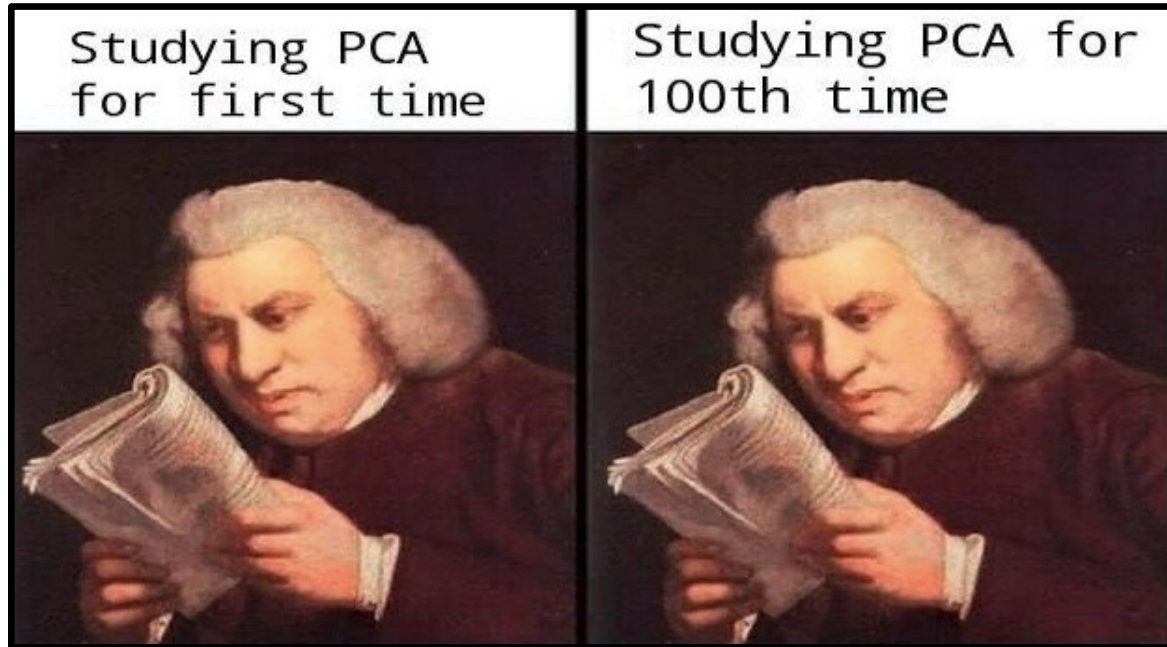


BIOL 501: Multi-variate Analysis (mostly PCA)



Peer-Feedback Survey on [last](#) discussion
and moderation today

Assignment #3: Loops & Publishable Figures

- **Use any dataset**
 - From previous assignments (must improve and reformat high quality Fig)
 - From any BIOL 501 workshop
 - From built in R datasets
 - From online sources (See Assignment #1 handout)
- **Goal 1:** Write a loop or function to execute 2 conditional tasks
 - Needs to do at least 2 calculations or tasks and these tasks need to be performed based on a condition
 - E.g. perform calculation 1 on a subset of data meeting X criteria then perform calculation 2 on a different subset
 - Or Perform 2 calculations on a subset of data meeting X criteria
- **Goal 2:** Create high quality figure on loop/function output meeting journal specs
- **Self-assessment with rubric *prior* to turning it in**

Due Mon. April 17 @9pm

See Canvas for details, R scripts to practice loops, journal specs, and more details

Students Choice Lecture and Workshop April 11 and 13

- No paper to read or discussion
- **Tues April 11th Lecture based on student survey: Loops, Fx, High quality figures, and other tidbits in R**
 - How to export high quality, engaging figures in R (**Assignment #3**)
 - Loops for plotting (both exploratory and final figs), calculating, data manipulation (**Assignment #3**)
 - Writing functions (don't be intimidated, R was designed for functions!)
 - Dealing with time in R (aka Excel hates time)
 - Other useful R Tricks to improve management, efficiency and accuracy
- **Time in Workshop on 13th to do Assignment #3 (come with dataset and loop started)**
- Time in class 11th and 13th to do UBC and BIOL 501 course specific surveys
 - R is a functional programming language: you can wrap up many loops in a **function** and call that function instead of using the loop directly

Tips for Assignment #3: Start now

- Get started on the loop/function part now—find dataset and think what do you want your loop/function to do?
- Write out 1 iteration of the loop step-by-step as an example of what you want it to do perhaps?
- We will go over exporting high resolution figures in Workshop April 13 (**4 days prior to assignment**)
- **Have dataset chosen before April 11th and idea of what tasks you want loop/fx to do**

Outline

- What are multivariate analysis and why do them?
- Ordination, classification, and model fitting
- Principal component analysis (PCA)
- Discriminant analysis (quickly)
- Species presence or absence data
- Distance Data
- Workshop

Why are multivariate stats useful?

- Real-world data are often multivariate
- **In biology:** We typically measure **multiple variables** on populations, species, and ecosystems
- **Challenge:** How to display and analyze measurements of multiple variables?
- **Solution:** need ways to make it easier to find important patterns and relationships among many variables



Multivariate methods are used for

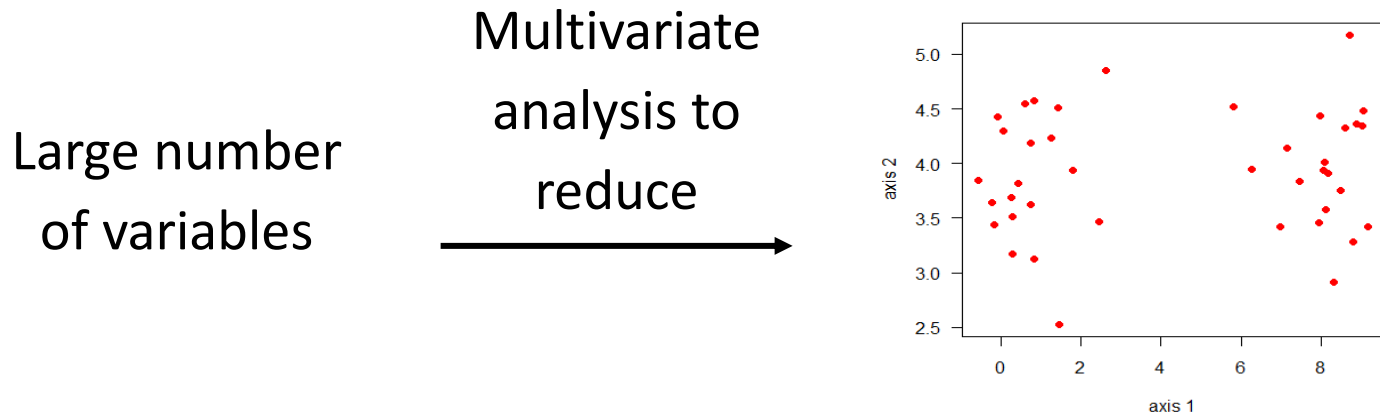
- 1. Ordination:** arrange sampling units along composite variables
 - Principal component analysis
 - Correspondence analysis
- 2. Classification:** place sampling units into groups
 - Discriminant function analysis
- 3. Model fitting:**
 - multivariate analysis of variance
 - multiple regression
 - More than 1 explanatory (X) variable
 - EX with LME
 - Adding another explanatory variable that is continuous as an explanatory variable
 - Different than adding a categorical fixed factor



Focus today

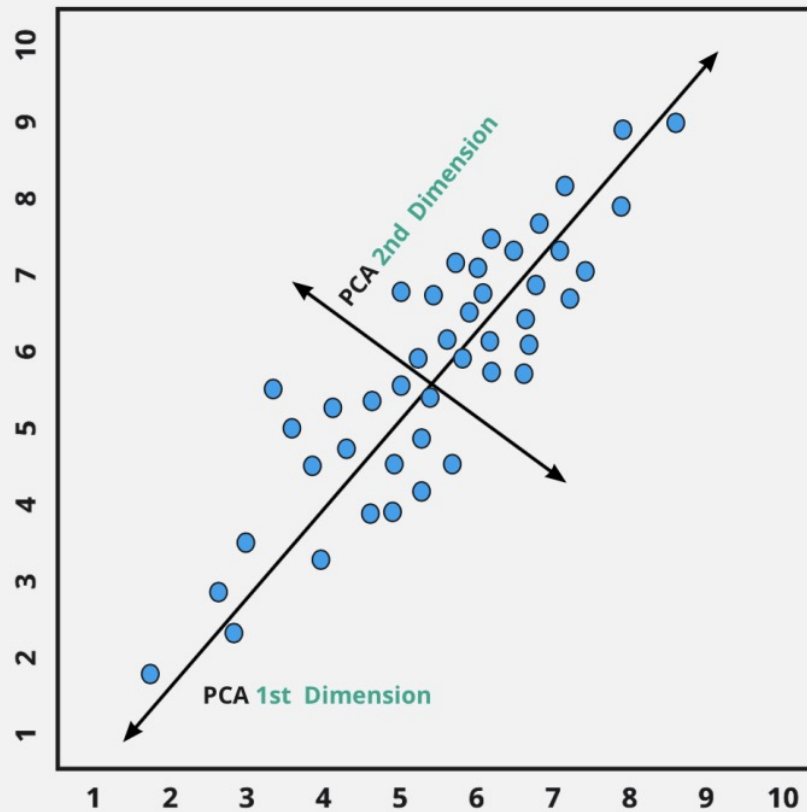
What is ordination

- Arrange sampling units along gradients or according to combinations of variables
- Used to
 - Visualize complex data in a few dimensions
 - Find meaningful combinations of the original variables that can be used in **subsequent** analysis



PCA

PCA: Principal Component Analysis



PCA is a type of ordination to display patterns

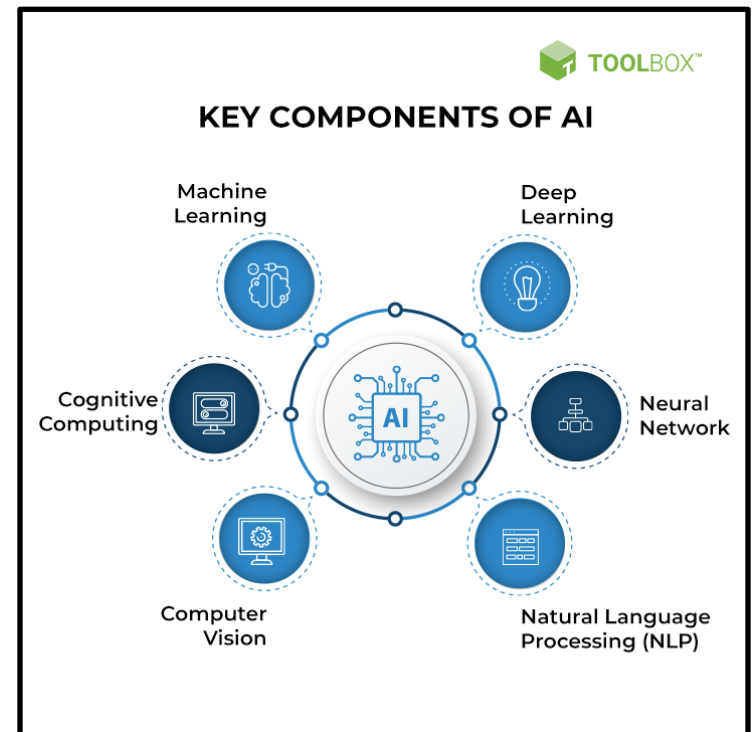
- Collected so much data, that you don't even know where to begin?!
- Use PCA to reduce the data down to usually 2 dimensions (sometimes 3)
- Plot it, and look for structure and patterns



PCA is used in Computer learning and AI

- PCA is also used unsupervised machine learning and various AI applications that require dimensionality reduction such as

- Finding hidden patterns in data with high dimensions
- Computer vision
- Image compression



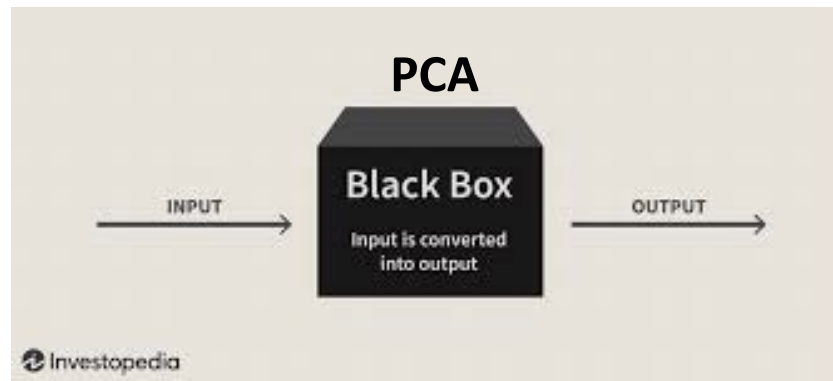
PCA: The bigger picture

- **Simple goal of PCA** is to reduce the number of variables of a dataset while preserving as much of the information as possible
- Specifically, find a small number of **linear combinations** of the variables to capture most of the variation in the dataset as a whole
- The trick in reducing many variables down to a few is to trade a little accuracy for simplicity.

“PCA is a dimensionality reduction technique”

Interpretation of PCA


- “..a black box that is widely used, but poorly understood”¹
- “Calculating PCA is easy. Interpreting what the components mean is hard, and potentially equivocal [*ambiguous*]”



¹Shlens, Jonathon. "A tutorial on principal component analysis." *arXiv preprint arXiv:1404.1100* (2014).<https://www.cs.cmu.edu/~elaw/papers/pca.pdf>

²The R book, Ch 23, pg 732

PCA: How it works...the details



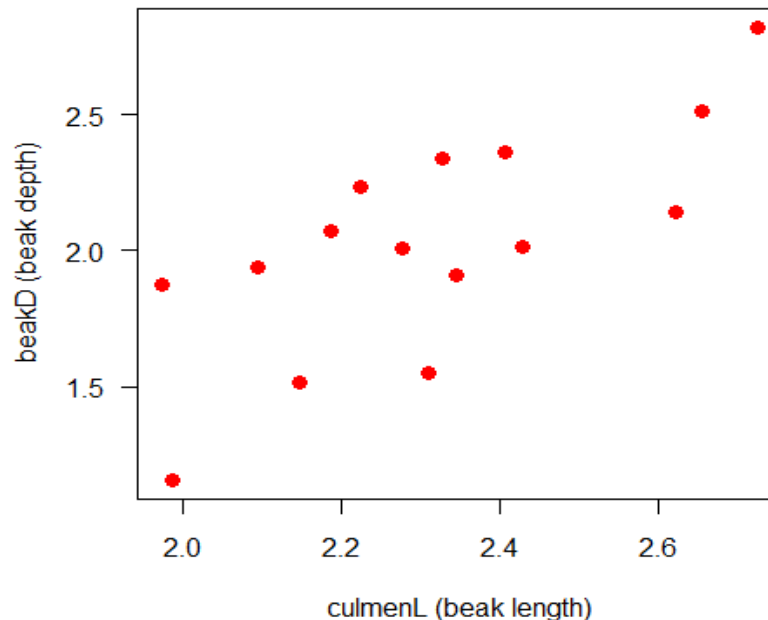
Why care?
Just tell me
the R code?

- **Goal today:** By understanding how PCA works and de-mystifying eigenvectors and eigenvalues, I hope to make interpretation of PCA outputs more meaningful

How does PCA work? Rotation of axes

- Overall amounts to nothing more than a rotation of the axes, allowing you to view much of the data in a smaller number of dimensions

We measured bird beak length and depth and plotted them in a scatterplot



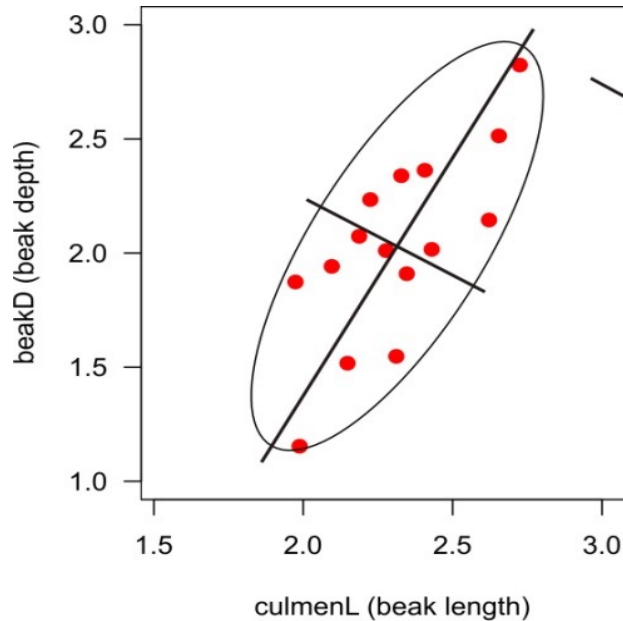
This is our original data units that has been log transformed

Why was it transformed? Next...

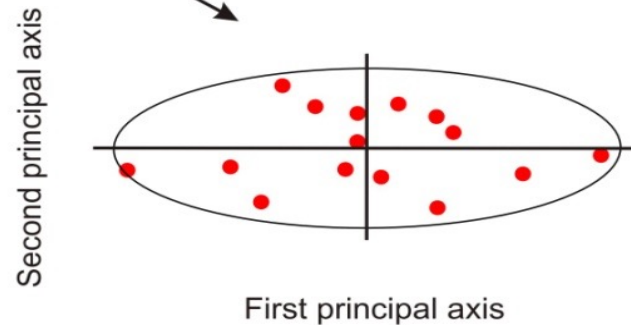
Preparing variables before PCA

- Prepare variables on a common scale (standardization)
- PCA results are more useful if all the variables are on a common scale (e.g. log transform)
- Each variable contributes mostly equally to analysis
- If on a comparable scale, then will have relatively similar variances (within an order of magnitude)

- PCA rotates the original points so that the new axes are uncorrelated



PCs are dimensionless –no units



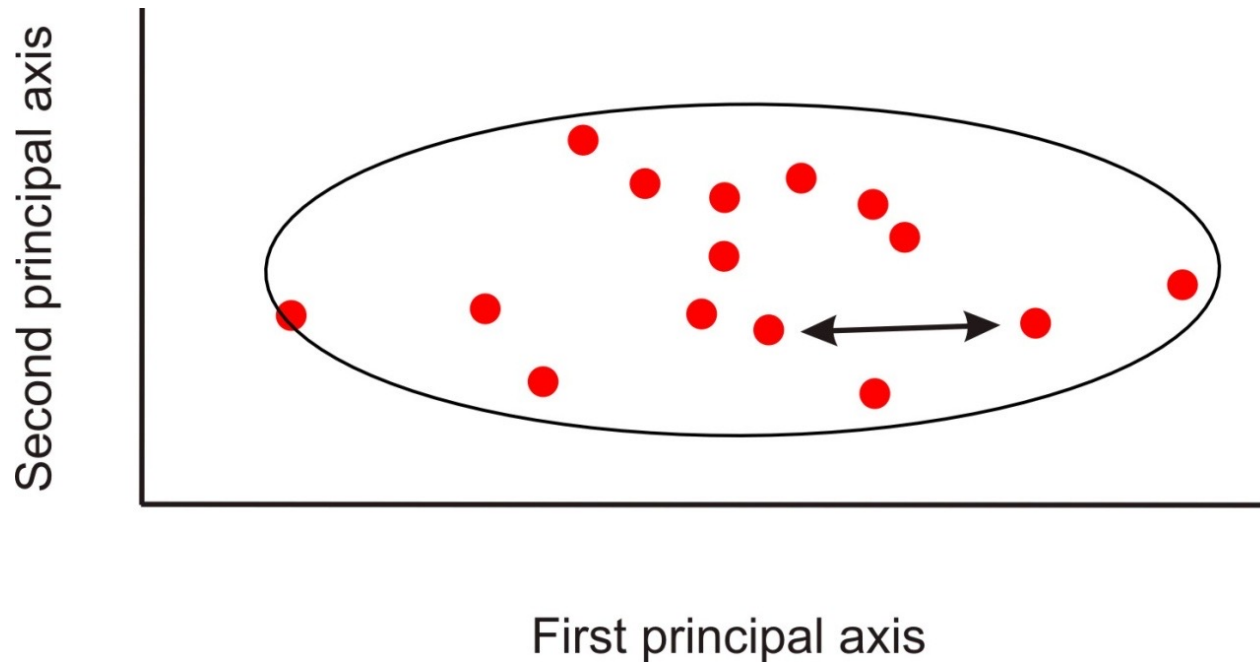
Original data units (cm or mm)

NEW X axis is principal component (PC1)

NEW Y axis is PC 2

PC 2 is always perpendicular to PC1

Because it is nothing more than a rotation of the axes, the distances between pairs of points are unchanged by the transformation (provided that all the PC axes are retained*).



*Warning: in some stats programs the default procedure is to standardize the variables (“correlation matrix”) before carrying out the analysis. Use the correlation matrix only if variables lack a common scale. Euclidean distances will then be based on standardized data, not the original measurements.

What is a principal component (PC)

- Principal components are the underlying structures in the data, where there is the **most variance**
- Principal components are **new variables** that are constructed as **linear combinations** of the original variables
- New variables (the principal components) are uncorrelated and most of the information within the initial variables compressed into the first few components

Goal is to find a linear combination of a set of variables that maximizes variation contained within them.

Video illustrating PCA

- Explains how get PC1 and PC2 from rotating axis
- PC1 spans the direction of the most variation and captures the most variation in the data
- PC2 spans the direction of the second most variation and captures the 2nd most variation in the data



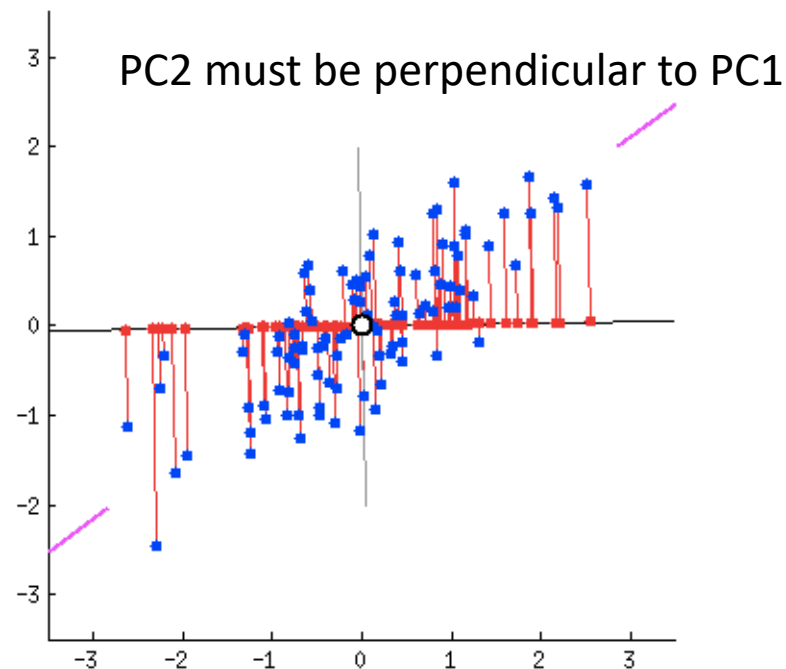
StatQuest 2015 video (elapsed 9:03-11:08)

https://www.youtube.com/watch?v=_UVHneBUBW0

- Another useful animation illustrating how we find PC1 and PC2 and the line that maximizes the variance (maximizing variance is the goal)

Blue dots=our original data

Red dots=projected points onto the first axis (PC1)—want to maximize the red lines



Scroll to mid way “**How PCA Constructs the Principal Components**”

<https://builtin.com/data-science/step-step-explanation-principal-component-analysis>

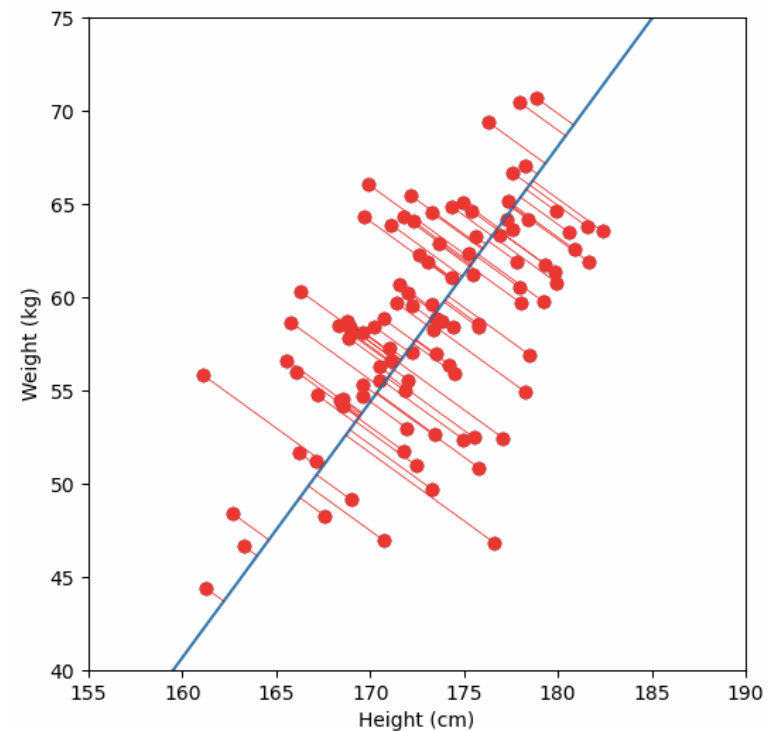
Rotation of axis

- Spatial relationships of the points are unchanged; this process has **merely rotated the data**.
- By performing such a rotation, the new axes might have particular explanations, but interpretation can be tricky
- The **orientations** of these axes relative to the original variables are called the **eigenvectors**, and the **variances** along these axes are called the **eigenvalues** (*More on that later*)

Another article on PCA

- Middle ground between eigenvectors, too much math verses a simple graphical explanation

<http://www.billconnelly.net/?p=697>



More details on the math

- If you want a more detailed mathematical example check out StatQuest 2018 video below
 - <https://www.youtube.com/watch?v=FgakZw6K1QQ>
- Another visual explanation of PCA
 - <https://setosa.io/ev/principal-component-analysis/>

PCA Terminology: Eigenvalues and Eigenvectors

- Underlying linear algebra concepts behind how PCA works
 - Theory of PCA is based on eigenvectors and eigenvalues
 - Used to compute from the covariance matrix to determine the principal components

Focus on how to interpret and use them from PCA output **below** not the math behind how to calculate them (above in grey)

- **They always come in pairs.** Every eigenvector (array of loadings/weights) has an eigenvalue (variances)
- **Why care?** By ranking the eigenvectors in order of their paired eigenvalues from high to low, you get the principal components in order of significance

Eigen-what?

- The main thing to remember is what these values represent when extracting outputs
 - **Eigenvalues=variances**
 - **Eigenvector=array of weights**



Eigen-what?

- German root word “eigen” means “Own, inherent, characteristic”
- Think of values or vectors that are inherent or characteristics in a matrix (*this will make sense later*)
- Eigenvectors and eigenvalues characterize a linear transformation

Eigenvectors and eigenvalues are used to create PCs, which are used to transform the original data into a new set of uncorrelated variables.



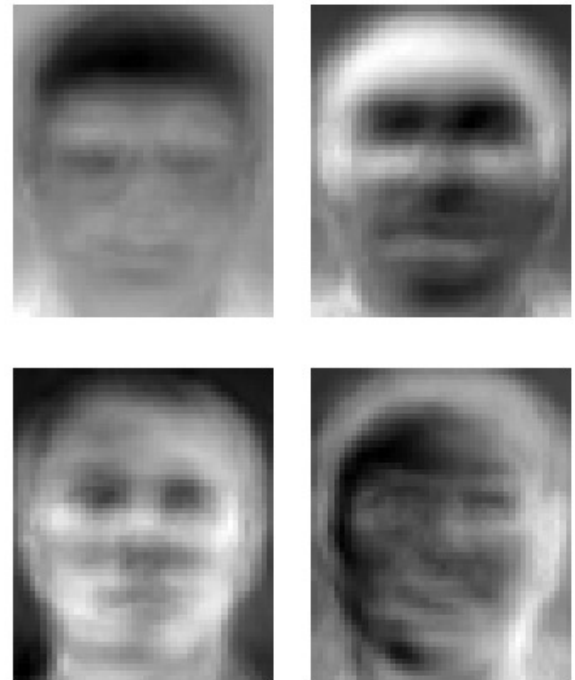
Eigenvalues, Eigenvectors, Loadings

- **Loadings=weights**
 - Indicates the weight (or contribution) that each variable contributes to **the principal component** (this means PC1)
 - Can intuitively think of this as “influence” or how much of the “load” does each variable carry
- **Eigenvector (array of loadings, or array of weights)**
 - **An array of loadings (weights)**
 - Eigenvector value *squared* has the meaning of the contribution of a variable into a pr. Component
 - If it is high (close to 1) the component is well defined by that variable alone.
- **Eigenvalues (variances)**
 - How much of the total variation is explained by the PCs
 - These are plotted in the scree plot (Yaxis)
 - **Common cutoff point for which PCs to retain:** Eigenvalue >1 indicates that PC accounts for $>$ variance than account for by 1 of the original variables. This holds true only when the data are standardized.

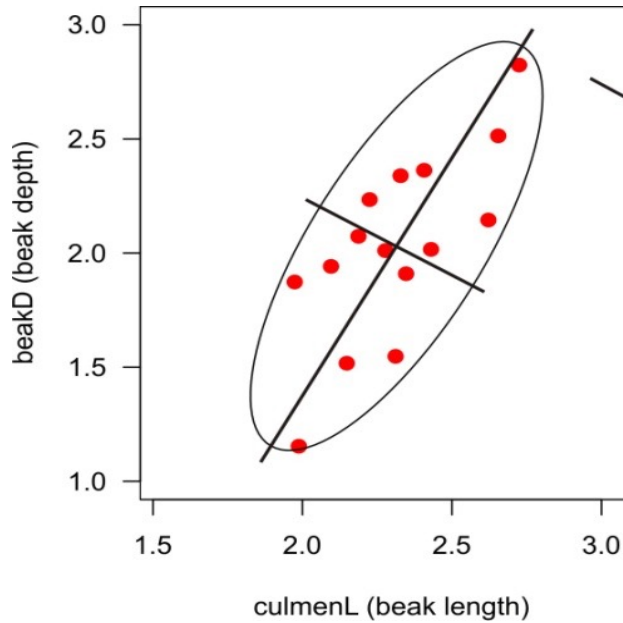
Using PCA in daily life and facial recognition

- Eigenvectors related to **computer vision** and computer facial recognition
- An **eigenface (or eigenimage)** is the name given to a set of **eigenvectors (array of weights)** when used in computer vision problem of facial recognition
- A **set of eigenfaces** can be generated by performing PCA on a large set of images depicting different human faces.

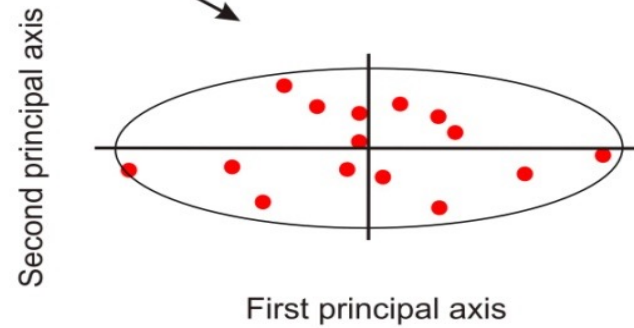
Train computers to capture and interpret information from images and video



Back to the rotation



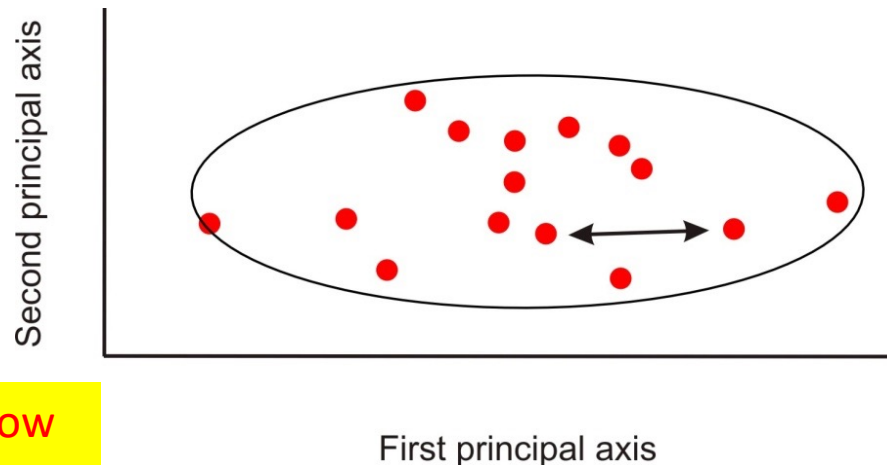
Original data units



**New data with principal components
No units**

Eigenvalues

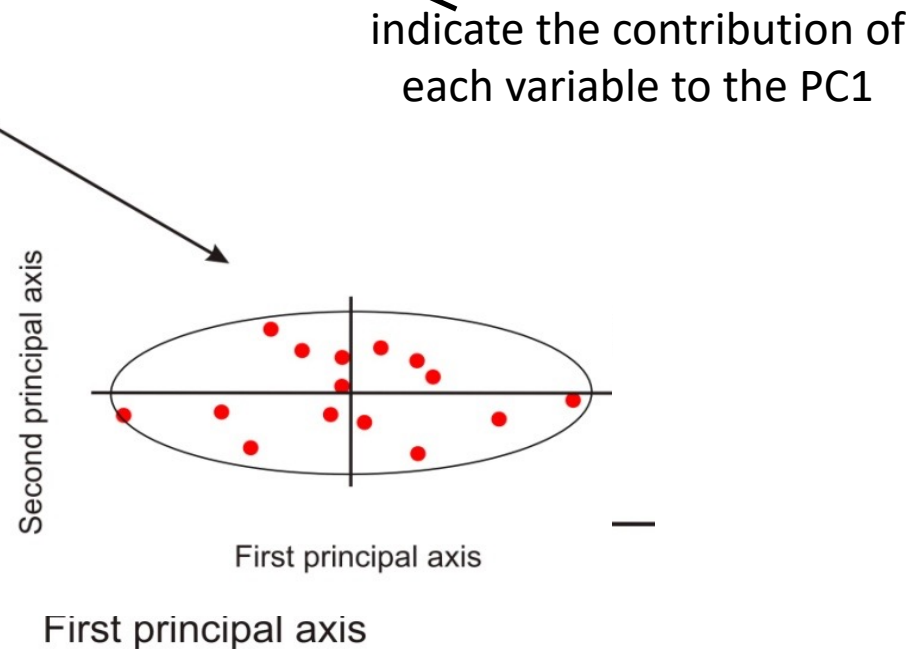
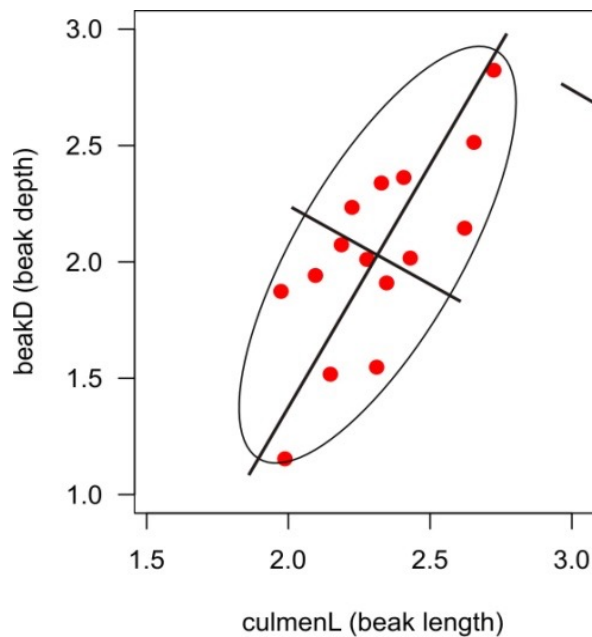
- The covariance matrix of the new, composite variables has variances on the diagonal and zeros off the diagonal.
- These variances are called **eigenvalues**
- Sum to same total as sum of variances of original data points



Eigenvalues (variances): How much of total variation is explained by the PCs

Eigenvector

- **Eigenvectors** contain the constants for transforming the original variables into the PCs
- The constants are called **loadings**

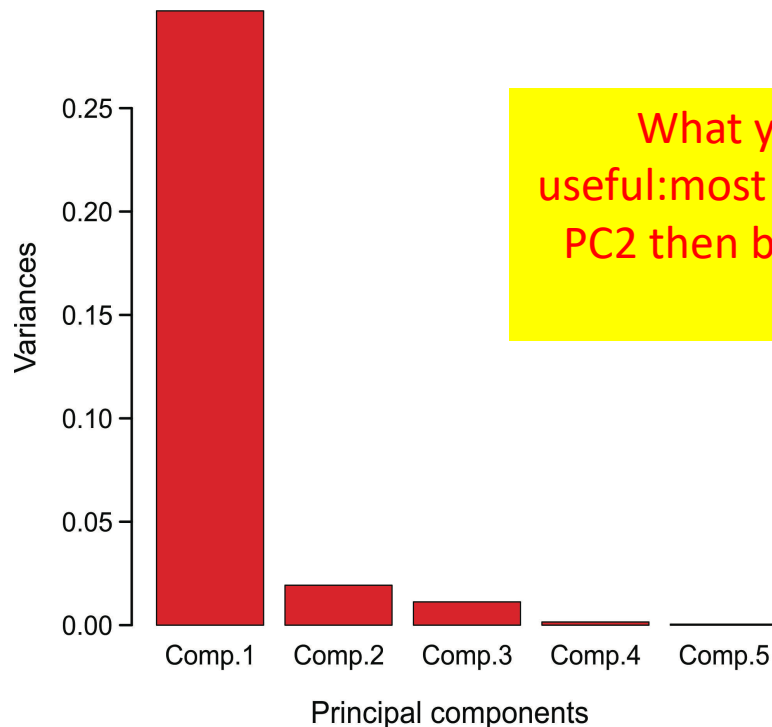


**Eigenvector (array of loadings,
or array of weights)**

Scree plot: How to tell if your PCA is worth anything?

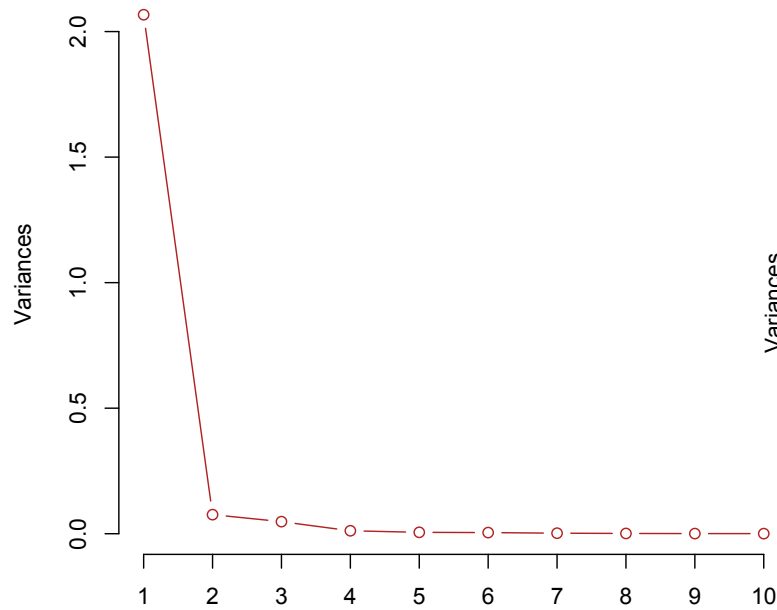
- Diagnostic tool to assess if PCA works well on your data or not
- Plot of the variance vs PC1 and PC2
- Shows the relative importance of each component to the total variance

The variances are also called Eigenvalues

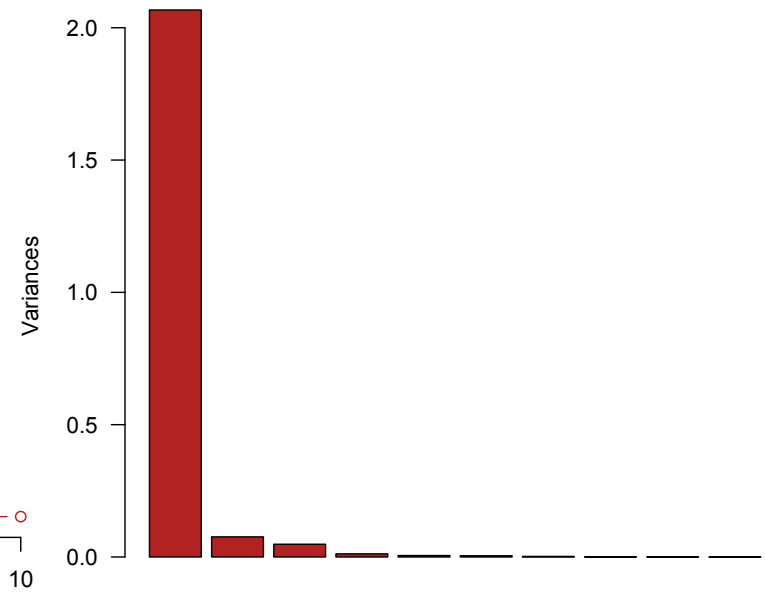


What you want if PCA is useful: most variance is in PC1 and PC2 then bends at "elbow" and levels off

What you want if PCA is useful: most variance is in PC1 and PC2 then bends at “elbow” and levels off



`Screplot(pca.model1, type="line")`

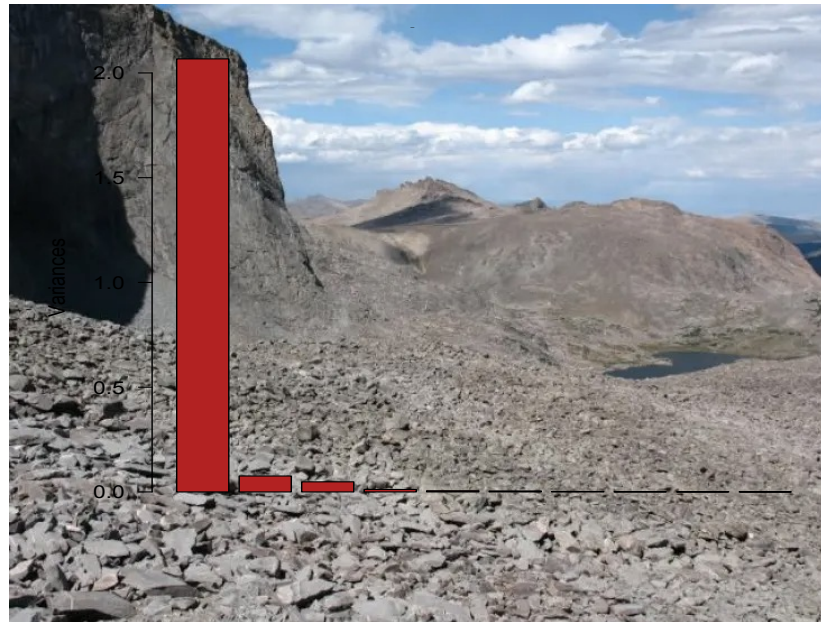


`Screplot(pca.model1, type="barplot")`

These examples have clear levelling off after PC1 and PC2

Scree plot

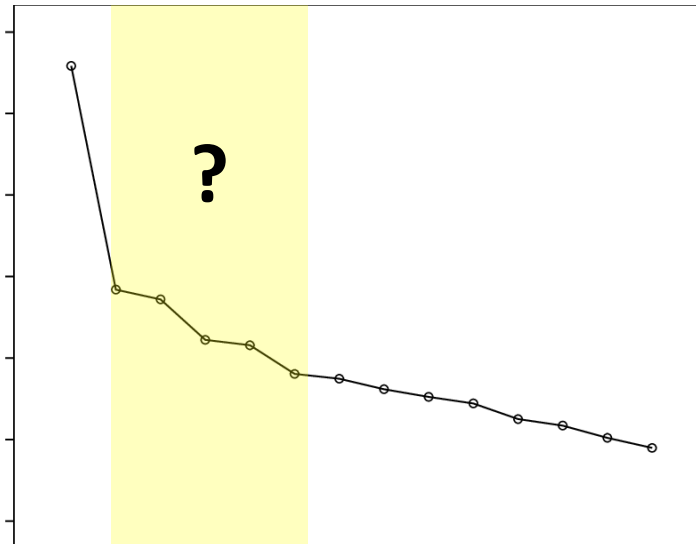
- Looks like a 'scree' slope, where rocks have fallen down and accumulated on the side of a mountain.



Scree is a collection of broken rock fragments at the base of a cliff, broken rock gradually moving downwards

If not-so-ideal or ambiguous PCA Screeplot?

- Scree plots can be subjective to interpret when they exactly level off or not always clearly yield 1-3 PCs
- If too many PCs ($> \sim 3$) that explain variation, then PCA might not be the best type of analysis

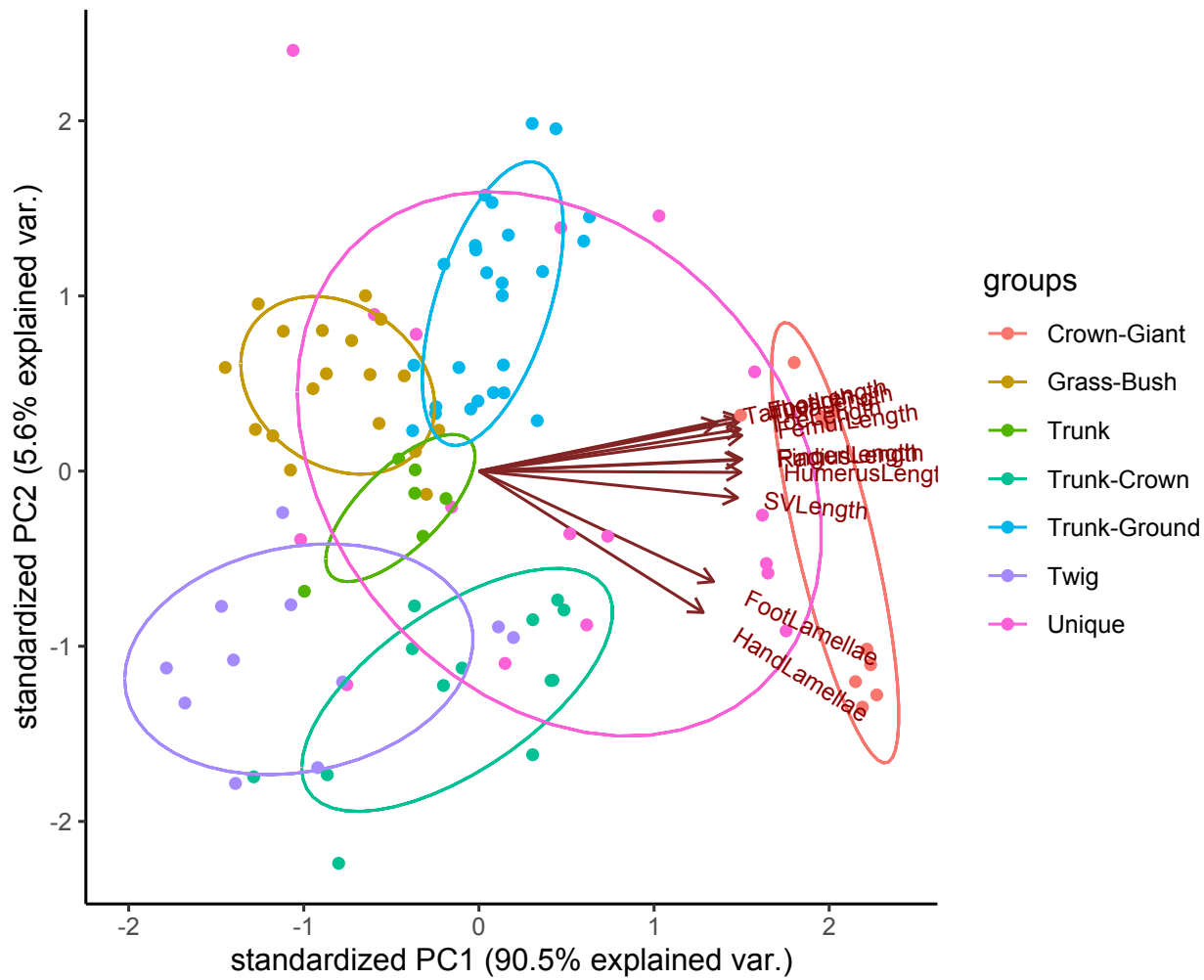


Optional strategies for choosing PCs

- **Retain eigenvalues > 1** (if data are standardized)(Kaiser rule)
- **Threshold:** Set a threshold of explained variance. Which PCs explain at least X% of the variance (80-90% roughly)

Biplot for visualizing PCA Results

- Scatter plot of data points along a pair of principal components (usually PC1 And PC2)
- Overlays arrows to indicate the contributions of each trait to the principal components.
 - Original variables shown by arrows
- Numbers represent the rows of the original dataframe, direction of arrows show relative loadings on PC1 and PC2
- PCA biplot=PCA score plot+loading (weights) plot



Outline

- What are multivariate analysis and why do them?
- Ordination, classification, and model fitting
- Principal component analysis (PCA)
- Discriminant analysis (quickly)
- Species presence or absence data
- Distance Data
- Workshop



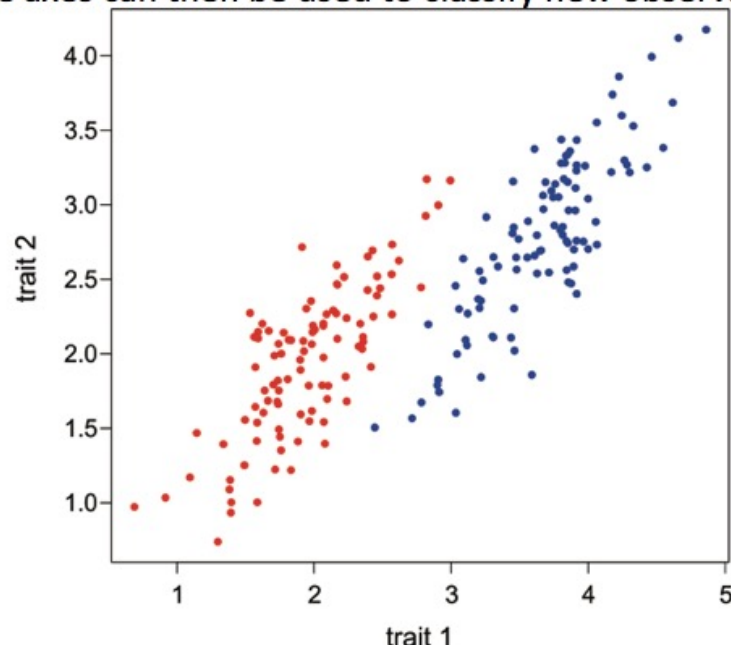
Will upload extra slides on these topics in separate file with Canvas lecture slides

PCA ignores the groups, and only finds the directions of maximum total variance,

Discriminant function analysis, quickly

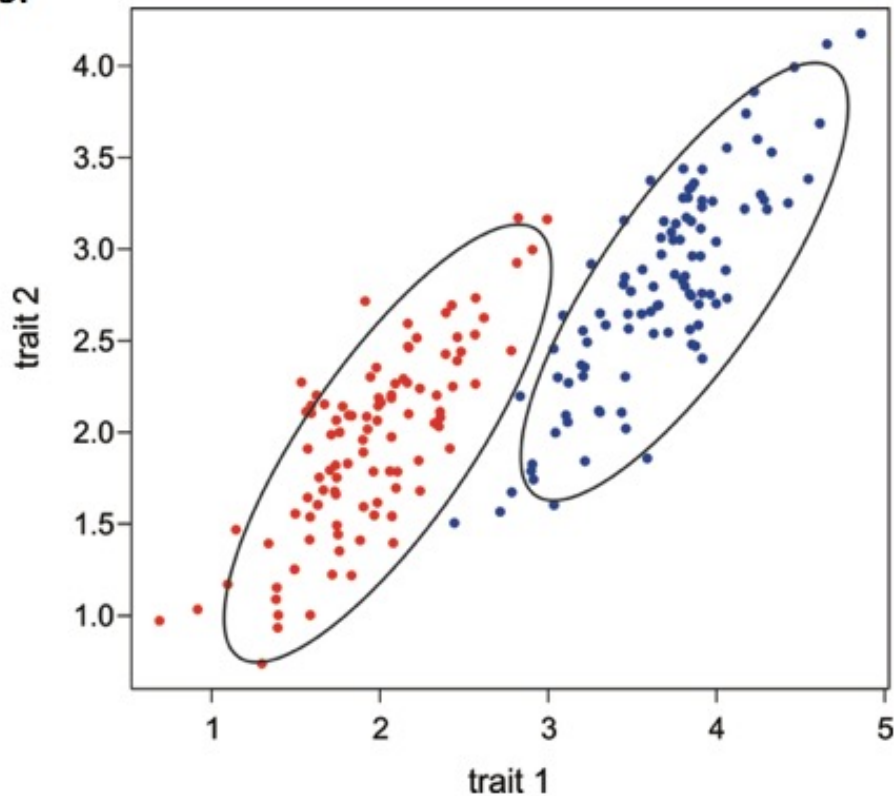
Discriminant function analysis is for classification rather than ordination.

It finds axes that maximally separate two or more previously identified groups. It finds axes that maximize variation among groups relative to variation between groups. These axes can then be used to classify new observations into the same groups.



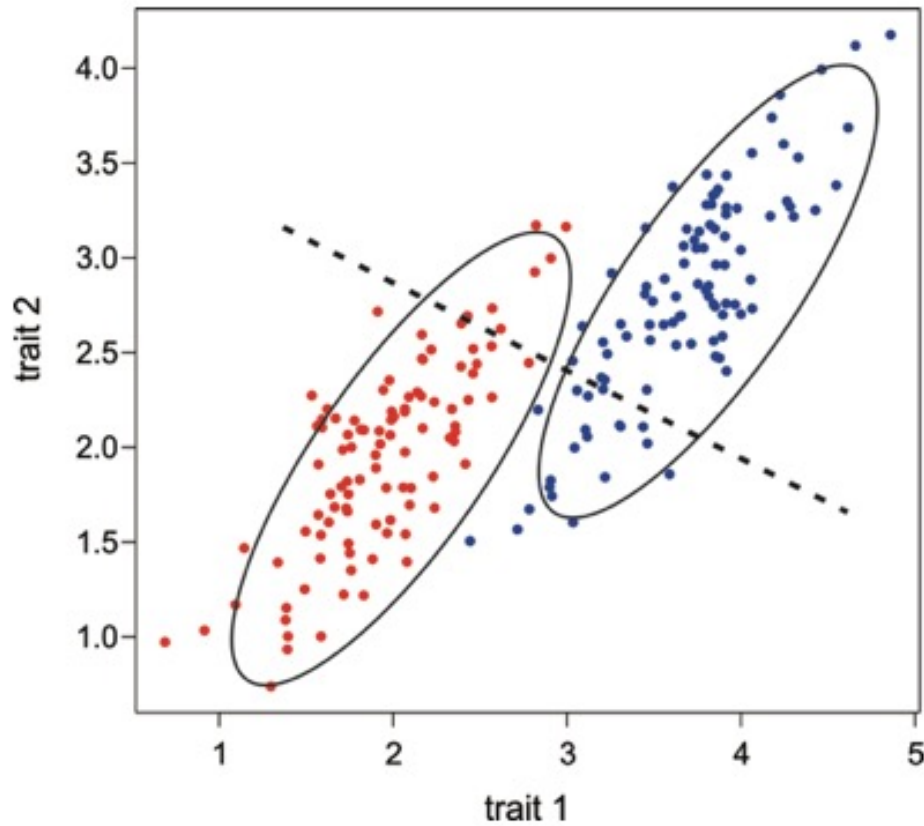
Discriminant function analysis

Discriminant function analysis explicitly finds the axes that best separate the groups. It doesn't do this simply by finding the direction of the biggest difference between means.



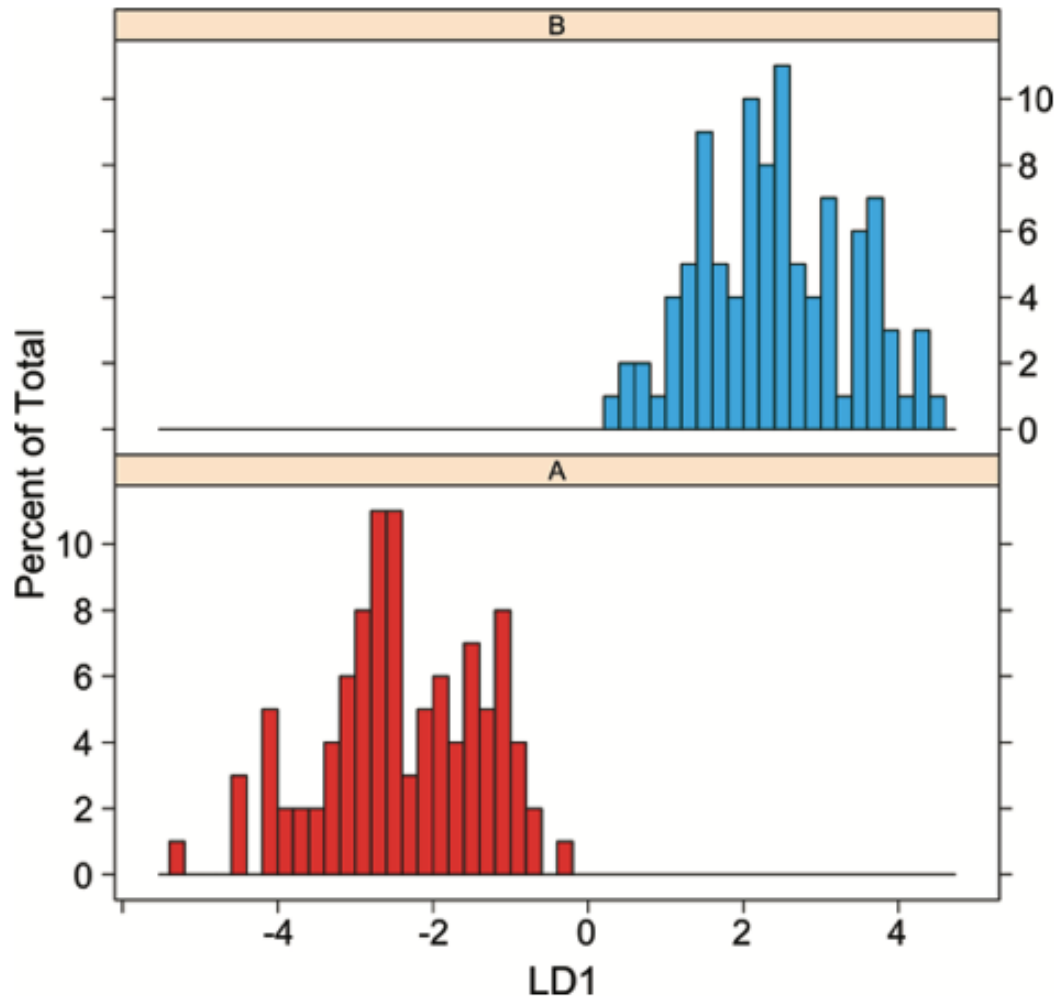
Discriminant function analysis

Instead, it finds the axes that best separate groups relative to within-group variation.



Discriminant function analysis

The resulting axes best separate the groups in the data, and can be used for classification of new observations.



Workshop

Workshop Thurs: Multivariate

- Principal components analysis
 - *Anolis* lizard variation
 - First install **devtools()** [you need devtools to install the next package]
 - **Second, install ggbiplot() but you install it differently**
 - **ggbiplot()** may not be available on CRAN so can get off github and **dplyer()**

- Uses prcomp()
- What is a screeplot?
- What are Eigenvectors

```
#might need this one too
library(ggplot2)

#load devtools first
library(devtools)
#then run this code below to install the package ggbiplot()
install_github("vqv/ggbiplot")

#Next you load ggbiplot()
install_github("vqv/ggbiplot")
library(ggbiplot) # optional
library(dplyr)
```

Workshop Thurs: Multivariate

- Principal components analysis
 - *Anolis* lizard variation
 - First install **devtools()** [you need devtools to install the next package]
 - **Second, install ggbiplot() but you install it differently**
 - **ggbiplot()** may not be available on CRAN so can get off github and **dplyer()**

PCA in R using prcomp()

```
#run PCA model. scale=True because the variances are different
```

```
pca.model1<-prcomp(mydata,scale=TRUE)
```

```
#get the summary of proportions explained
```

```
summary(pca.model1)
```

```
#get all of the output of the PCA model (verbose output contains rotation and standardeviations)
```

```
pca.model1
```

```
#get the Eigenvectors (loadings) from a PCA model for the first 3 PCs."rotation" PCs (the eigenvectors of the covariance matrix), in the original coordinate system
```

```
pca.model1$rotation [,1:3]
```

```
#to get the eigenvalues (variances) square the sdev which are within the pca.model1 output {it is the length of total number of PCs}
```

```
pca.model1$sdev^2
```

```
#Extract first 3 principal component scores which represent measurements of every individual on the principal component axes NOT the original axis
```

```
predict(pca.model1)[, 1:3]
```

Or you can do this step above AND bind it to the original dataframe all at once. Y

```
mydata[, c("pc1","pc2","pc3")] <- predict(pca.model1)[, 1:3]
```

Scree plot R Code

- Remember in R, **plot()** will do the default type of plot for the type of data or object class you are plotting
- The output of `prcomp()` is an object that is classified as “prcomp”
 - Check by doing `class(pca.model1)`
- A better way to do this is with **screepplot()**

```
#Plot a scree plot. The default type of plot for a prcomp() object is a scree plot  
Plot(pca.model1)
```

```
#Scree plot with specialized function
```

```
#Npcs=the # of components to be plotted, otherwise defaults to 10 if not specified  
Screepplot(pca.model1, type="barplot", npcs=X)
```


Workshop Thurs: Model Selection

- Correspondence analysis
 - Rodent ordination
 - Requires library (MASS)

- Discriminant function analysis
 - *Anolis* ecomorph
 - Requires library (MASS)

Workshop Thurs: Model Selection

- Correspondence analysis
 - Rodent ordination
- Discriminant function analysis
 - *Anolis* ectomorph discrimination
- #Anolis Lizard Variation----->principal components analysis