Loops

- **Simple loop**

```r
y <- 5
for (x in 1:10)
  y <- y + x
```

- Loops 10 times, adding x each time: y = 55

- Basis for any kind of permutation test...
- Observed statistic (e.g. R statistic in ANOSIM)
- **Loop:** Permute the dataset, calculate statistic
- Compare observed statistic to distribution of observed values

Loops

- **Two samples, traditional t-test**

```r
tapply(response_var, predictor, mean)
do.call(Summary, list(response_var, predictor))
```

- **Permutation test:**

```r
mean_silhouette <- numeric(15)
for (k in 2:16) {
  cluster_med_variable_k <- pam(morphology, k)$silinfo$avg.width
  mean_silhouette[k] <- cluster_med_variable_k$silinfo$avg.width
}
```

- **cascadeKM**

```r
cascadeKM does this analysis for you with kmeans clustering
```

```r
Note that kmeans does not work with missing data
```

```r
cascadeKM does this analysis for you with kmeans clustering
```

```r
Function returns
- group membership for each group size (similar to cuttree)
- Number of samples in each group
- SSE for each K

```r
```r
cascadeKM does this analysis for you with kmeans clustering
```

```r
Note that kmeans does not work with missing data
```

```r
cascadeKM does this analysis for you with kmeans clustering
```
K-means partitions comparison

Objects

Number of groups in each partition

Values

ssi criterion

Table Function

- The table function builds a contingency table for any number of variables.

```r
species <- factor(c("Bluegill", "Longear"))
distance <- dist(morphology, method = "euclidean")
cut_tree <- cutree(UPGMA, k = 2)
table(species, cut_tree)
```

<table>
<thead>
<tr>
<th>species</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluegill</td>
<td>61</td>
<td>0</td>
</tr>
<tr>
<td>Longear</td>
<td>91</td>
<td>1</td>
</tr>
</tbody>
</table>

```r
cut_tree <- cutree(UPGMA, k = 3)
table(species, cut_tree)
```

<table>
<thead>
<tr>
<th>species</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluegill</td>
<td>58</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Longear</td>
<td>89</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

```r
cut_tree <- cutree(UPGMA, k = 4)
table(species, cut_tree)
```

<table>
<thead>
<tr>
<th>species</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluegill</td>
<td>56</td>
<td>2</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Longear</td>
<td>88</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

```r
cut_tree <- cutree(UPGMA, k = 5)
table(species, cut_tree)
```

<table>
<thead>
<tr>
<th>species</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluegill</td>
<td>56</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Longear</td>
<td>0</td>
<td>88</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Fuzzy Clustering

- Specify k ahead of time, not hierarchical
- Observations may belong to multiple clusters (fuzzy)
- Functions
  - fanny (cluster package)
  - cmeans (e1071 package), fuzzy version of kmeans
- For the morphological dataset we are working with:
  - `fanny(distance, k=0, memb.exp=1.4)`
- How many groups (K)?
  - There are a variety of indices of how many groups a dataset contains. Package clusterCrit provides an easy way to calculate (function intCriteria).
  - Most are based on assessing maximization of among group differences or minimizing within group differences.
Phylogenetic methods in R

- **APE** - Analysis of Phylogenetics and Evolution
  - [http://ape.mpl.ird.fr/](http://ape.mpl.ird.fr/)
  - Read/write trees, DNA sequences
  - Genetic distance
  - Phylogenetic comparative methods (last week of class)
  - Estimate ancestral characters
  - Diversification analysis
  - Neighbor-joining trees
  - Population genetics

- Other packages
  - **Geiger** – analyses of evolutionary diversification
  - **phangorn** – tree estimation (maximum likelihood), also does upgma, neighbor-joining

Simple Phylogenetic Methods

- **First... need data**

```r
centrarchidae <- read.GenBank(refs)
```

- If you already have data: Read.nexus, read.tree, read.caic, read.dna etc.

```r
centrarchidae
```

- 34 DNA sequences in binary format stored in a list.
- All sequences of same length: 1140
- Labels: JF742821.1 EU501116.1 AY115980.1 EU501085.1 EU501064.1 EU501075.1 ... 
- Base composition:
  - a     c     g     t
  - 0.226 0.329 0.160 0.285

```r
centrarchidae$EU501085.1
```

- 1 DNA sequence in binary format stored in a vector.
- Sequence length: 1140
- Base composition:
  - a     c     g     t
  - 0.218 0.332 0.161 0.288

- Function **dist.dna()** will calculate a genetic distance matrix
  - See the help file for various options and metrics
  - The resulting distance matrix can be used in any clustering technique

```r
distance <- dist.dna(centrarchidae, model="F81")
UPGMA <- hclust(distance, method="average")
plot(UPGMA)
```

![Cluster Dendrogram](image)
More likely, you want to construct a tree based on an evolutionary model.

Neighbor-joining approach is hierarchical and agglomerative, where nodes represent common ancestors.

nj(), fastme() and bion() functions produce a phylo object:

Bootstrap analysis
- Function boot.phylo() performs a bootstrap (permutation) analysis of the phylogeny.
- Specify the number of randomizations, how columns are randomized etc.
- Function returns the number of times each node was supported.