# Ordination 2

## Lecture 10.2 PCA

Lauren Sullivan

Module: Multivariate Models

# Readings

**Required for class:**

- NA

**Optional:**

- Hayden, L. (2018) Principal Components Analysis in R.
  - Good reference for customizing ggbiplots.
- Lever, J. and Krzywinski, M. and Altman, N. (2017) Principle component analysis. PCA helps you interpret your data, but will not always find the important patterns. *Nature Methods*.

# Multivariate Analysis

There are several ways to look at multivariate patterns from a matrix of **Y**'s.

1. Linear models: MANOVA/regression to test patterns
2. **Ordination: PCA, nMDS, etc to visualize patterns**
3. Permutation tests: PERMANOVA to test patterns

# Principal Componants Analysis (PCA)

PCA is another form of dimension reduction. You take your original dataset of many Y's and simplify it by turning the original variables into a smaller number of "principal components".

- Principal components are the underlying structure of the data, and are in the directions where most of the variance lies. So in PCA we create new axes that are *linear combinations* of the original data that explain most of the variance.
  - You linearly transform your dataset so the first PCA coordinate or axis explains most of the variance.
  - Each subsequent axis is orthogonal (perpendicular) to the last axis and explains less variance. This makes the axes uncorrelated.
- When a lot of variables are highly correlated, they will all contribute to the same principal component axis.

# Understanding PCA

PCA rotates data so main axis of variation explained (PC1) is horizontal. Then subsequent PC axis is orthogonal, and is ordered to explain less variation.

- You are trying to explain more variation with fewer dimensions.

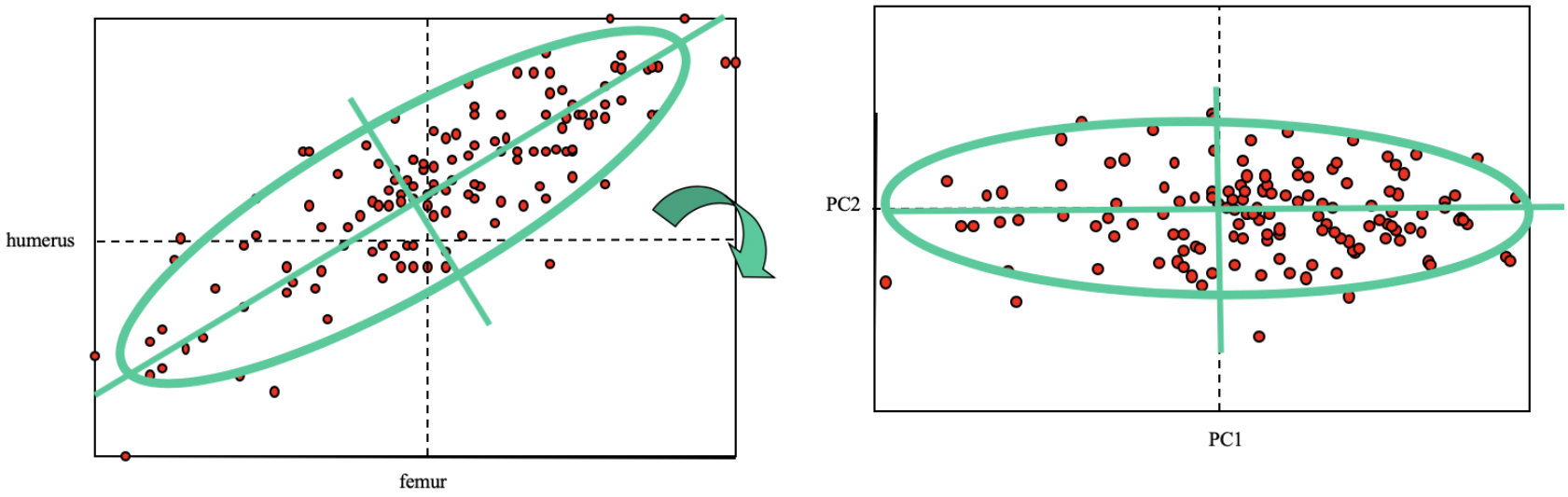



Figure 1: thanks to Dr. Dean Adams for this figure

# Understanding PCA

**Step 1:** Start with data matrix (**Y**).

**Step 2:** Obtain covariance matrix (**S**) or correlation matrix (**R**) from **Y**.

**Step 3:** Run eigenanalysis on **S** to find orthogonal vectors.

- An *eigenvector* is a vector of the transformed matrix that points in the direction of variance explained, while the corresponding *eigenvalue* tells you how much variance there is in the data in that direction.
  - The eigenvector with the largest eigenvalue is PC1.

**Step 4:** Project data onto the vectors to get PC scores for you rows of data (sites, individuals, etc).

# Correlation vs Covariance

Because variables in the $\mathbf{Y}$ matrix with high variation can have a lot of influence on the PCA, it's better to use a correlation matrix ($\mathbf{R}$).

- Correlation values are between 0 and 1.

If you use a covariance matrix ($\mathbf{S}$), then you should scale the variables in $\mathbf{Y}$.

# PCA Interpretation Notes

- PCA preserves Euclidean distances among rows of data.
- If original data variables are uncorrelated, then PCA is not helpful in reducing dimensionality of data.
- PCA does not *find* a particular factor to be significant. It identifies the direction of the most variation, which *may* be interpretable as a factor, but it may not.
- Be careful interpreting all PC axes as biologically meaningful. PC axes are constrained to be orthogonal, biological variability is not.
- *Make sure your data are scaled and/or centered*
  - If you see strange patterns in your PCA (e.g. horseshoe shape, very linear trends), something may be up with scaling.
  - Variables in $\mathbf{Y}$ with high variation unduly influence the PCA.
  - If you don't scale the data, use correlation matrix ($\mathbf{R}$) - between 0 an 1.

# Sparrow Data

Data from Bumpus (1898) - where he measured ~136 sparrows after a bad February storm. Half the birds were dead. Bumpus wanted to investigate natural selection and determine if there was a difference between dead or alive birds.

# Sparrow Data

Data includes: sex, age, survived (TRUE/FALSE), total length (TL), wing extent (AE), mass (WT), beak-head length (BHL), humerus length (HL), femur length (FL), tibiotarsal length (TTL), skull weight (SW), sternum-keel length (SKL).

```
sparrow
```

```
## # A tibble: 136 x 12
##    sex   age   survived    TL    AE    WT   BHL    HL    FL   TTL    SW   SKL
##    <chr> <chr> <lgl>    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
##  1 m     a     TRUE       154   241  24.5  31.2  17.4  17.0  26.0  14.9  21.1
##  2 m     a     FALSE      165   240  26.5  31    18.7  17.9  27.8  15.4  21.5
##  3 m     a     FALSE      160   245  26.1  32    18.7  18.0  28.2  15.5  21.4
##  4 m     a     TRUE       160   252  26.9  30.8  18.7  18.0  30.0  15.3  21.4
##  5 m     a     TRUE       155   243  26.9  30.6  18.6  17.9  29.2  15.3  21.5
##  6 m     a     FALSE      161   249  25.6  32.3  18.9  18.2  28.7  15.3  21.0
##  7 m     a     TRUE       154   245  24.3  31.7  18.8  17.5  29.1  14.8  21.3
##  8 m     a     FALSE      162   246  25.9  32.3  18.7  18.0  28.8  15.4  22.1
##  9 m     a     TRUE       156   247  24.1  31.5  18.2  17.9  28.7  14.6  20.9
## 10 m     a     FALSE      163   250  25.5  32.5  19.1  18.6  30.4  15.8  22.6
## # ... with 126 more rows
```

# PCA

Let's look again at our $\mathbf{Y}$ from the sparrow data to see how variables are correlated and create hypotheses about how sex and survival influence a bird's body size.

- Common functions for running PCA are `prcomp()` - documentation here, and `princomp()` - documentation here.

```
sparrow.pca <- prcomp(sparrow[, -c(1:3)], center = TRUE, scale = TRUE)
summary(sparrow.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.3047 0.9989 0.81280 0.73068 0.67838 0.63625 0.52105
## Proportion of Variance 0.5902 0.1109 0.07341 0.05932 0.05113 0.04498 0.03017
## Cumulative Proportion  0.5902 0.7010 0.77445 0.83377 0.88490 0.92988 0.96005
##                           PC8     PC9
## Standard deviation     0.46169 0.38264
## Proportion of Variance 0.02368 0.01627
## Cumulative Proportion  0.98373 1.00000
```

So PC1 explains 59% of the variation, PC2 explains 11% of the variation, PC3 explains 7% of the variation, and so on. We can explain ~70% of the variaiton from just two axes!

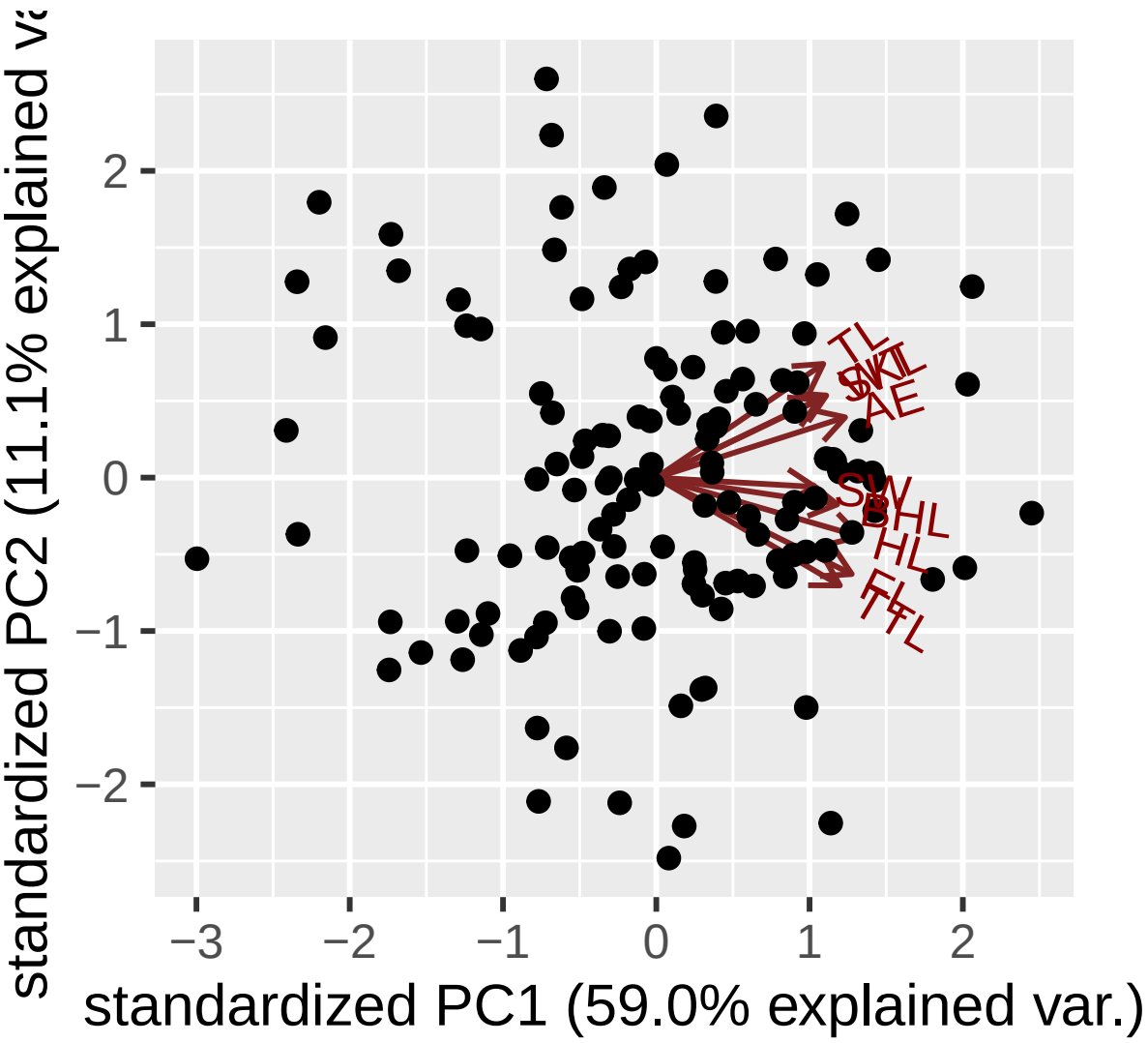
***Note:*** You get as many PC axes as you have columns in your $\mathbf{Y}$, and the proportion of variance explained by all sums to 1.

# Plotting

Plot the PCA using `library(ggbiplot)`, installed from github.

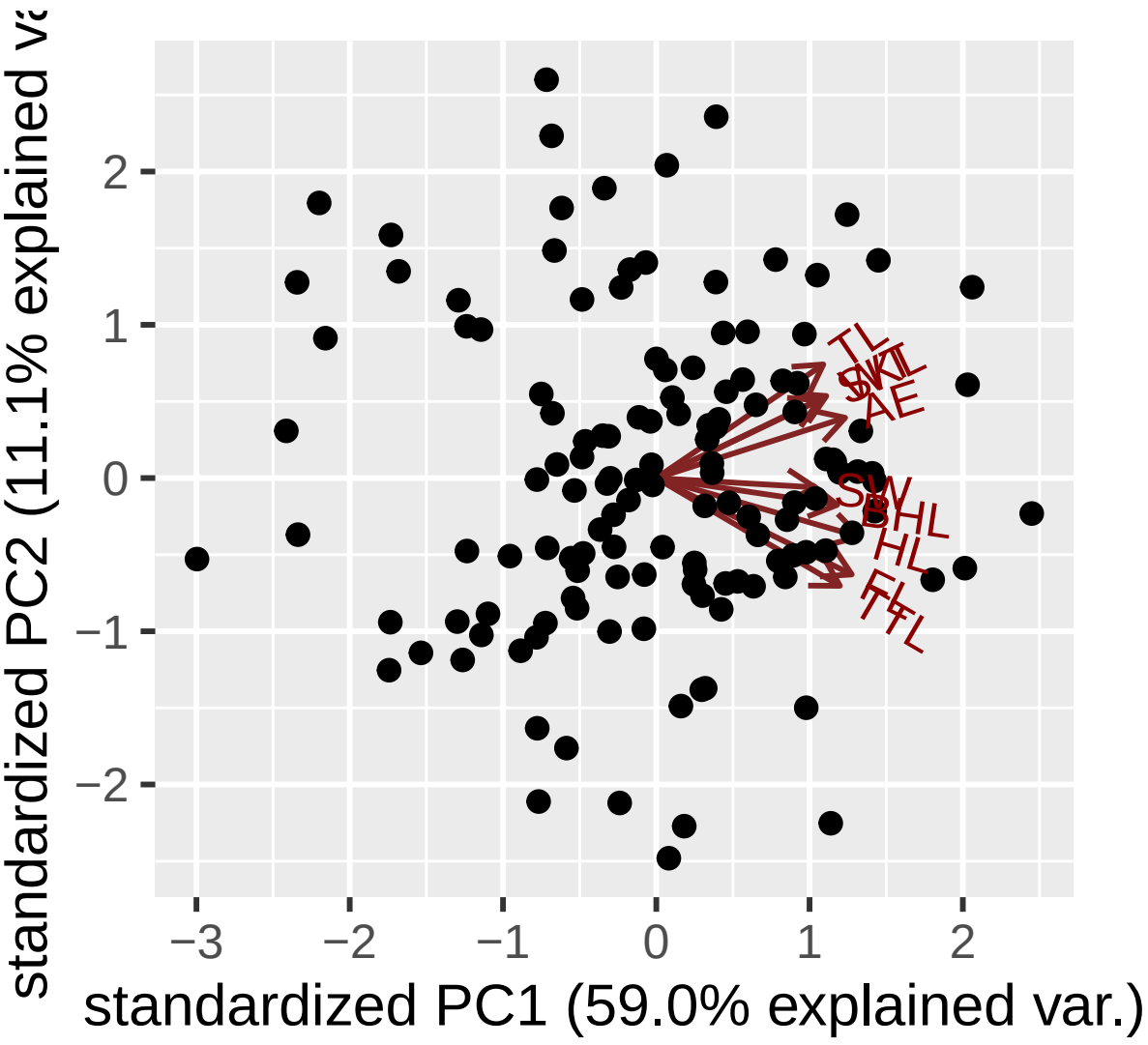
```
library(devtools)
install_github("vqv/ggbiplot")
```

```
ggbiplot(sparrow.pca)
```



# Interpreting - Figures

The data columns are the arrows from the center, and their direction indicates which axes (that are plotted) they relate to.



# Variable Loadings

Loadings for each Y variable show how well each variable corresponds to each PC axis.

- Absolute values gives relationship, +/- gives direction.

```
sparrow.pca$rotation
```

```
##           PC1         PC2         PC3         PC4        PC5         PC6
## TL  0.3100651  0.48676787 -0.05636978  0.45377621 -0.1326996  0.36031440
## AE  0.3505712  0.25833261 -0.33786910  0.25610634 -0.2419548  0.02384397
## WT  0.3155630  0.35142457  0.19791968  0.07935593  0.7038798 -0.46084211
## BHL 0.3358877 -0.11469994  0.29286266 -0.30567863  0.3264084  0.73723935
## HL  0.3779134 -0.25196559 -0.22209147 -0.02197200 -0.0837826 -0.11938625
## FL  0.3628170 -0.41308546 -0.14264156  0.07732550 -0.0761244 -0.04231837
## TTL 0.3408942 -0.46175891 -0.15231419  0.13796632  0.1776658 -0.13117472
## SW  0.2946541 -0.03996533  0.78349062  0.04854279 -0.4795809 -0.23554995
## SKL 0.3017853  0.33274799 -0.22582687 -0.77518179 -0.2179036 -0.15802907
##             PC7         PC8         PC9
## TL  -0.51853028 -0.13136187 -0.15575960
## AE   0.64295510  0.33808212  0.20932445
## WT   0.06240530 -0.10097266  0.09766796
## BHL  0.20681677  0.04051399  0.01834518
## HL   0.25053953 -0.58145771 -0.56723163
## FL  -0.23333115 -0.31239139  0.71538857
## TTL -0.30096834  0.63401809 -0.29496225
## SW   0.04561749  0.08045686 -0.03649455
## SKL -0.24736838  0.11168704  0.01339906
```

# Dimension Reduction with PC Scores

You can extract PC scores that tell you where on each PC axis each row of your data (here, individual bird) fall.

- This is useful for collapsing lots of columns of correlated Y's into just a few transformed Y variables.

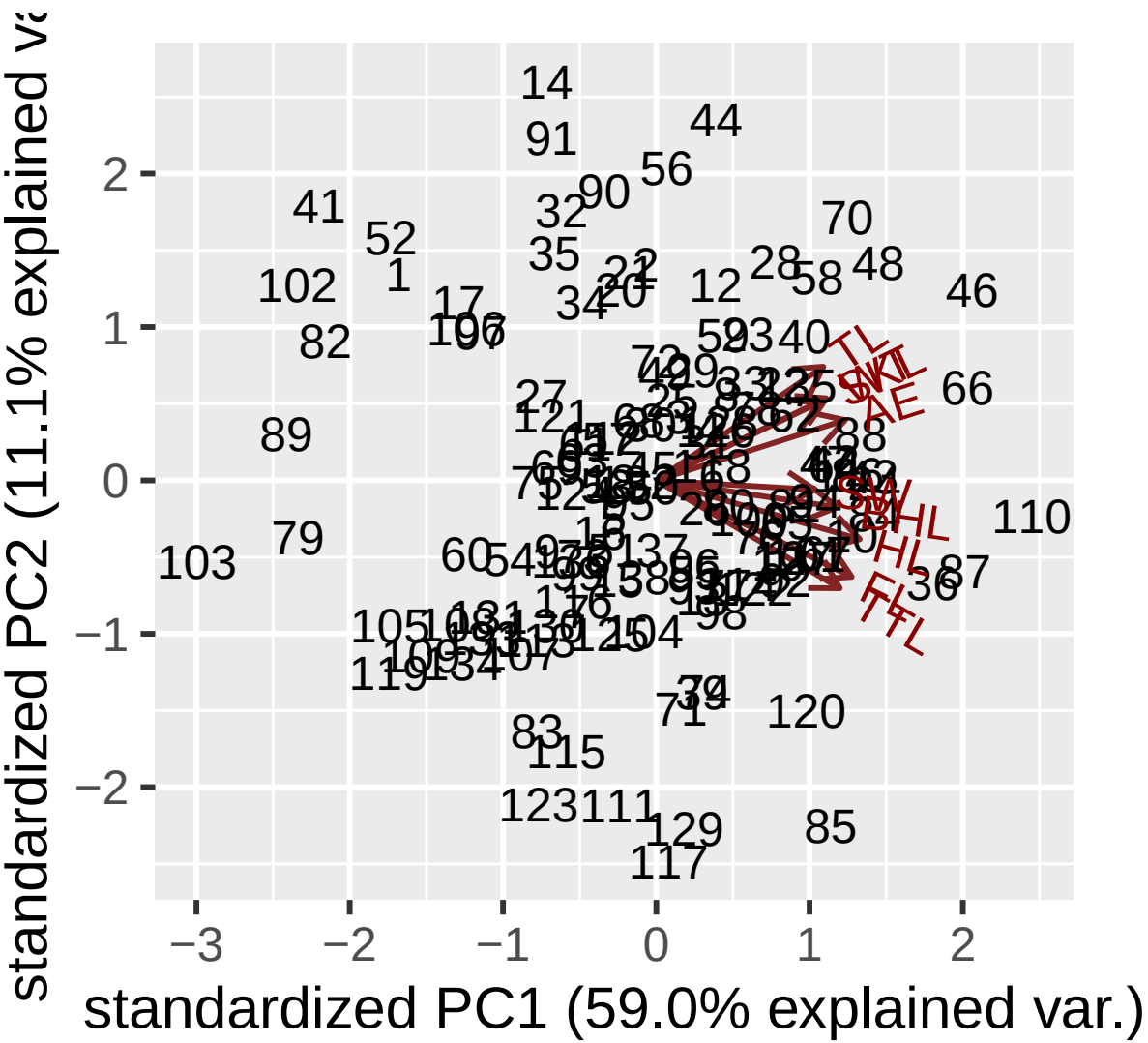
```
sparrow.pca$x[1:12, 1:4]
```

```
##               PC1         PC2         PC3         PC4
##  [1,] -3.8737549  1.34728089  0.41450308 -1.1284675
##  [2,] -0.1569321  1.40448648  0.40572787  0.4647756
##  [3,]  0.3376271  0.41857778  0.76923927 -0.2179035
##  [4,]  0.7889555  0.34359930 -0.78210792  0.9029232
##  [5,] -0.6424813 -0.23933131 -0.09719630 -0.2738306
##  [6,]  0.8408125  0.03490247 -0.01549063  0.2687365
##  [7,] -1.1912270 -0.84880914 -0.95816048 -0.9243453
##  [8,]  1.0531543  0.56289262  0.29542342 -0.5126552
##  [9,] -1.6395080 -0.45571657 -1.38286389 -0.1527217
## [10,]  2.9423077 -0.35676171  0.29563222 -0.3607434
## [11,]  2.7571105  0.03672089 -1.02075495 -0.4352050
## [12,]  0.8964696  1.27740023 -0.17819820 -0.2253507
```

# Fun With Plotting

Add plot names with `labels` (this may be useful sometimes, not in this case).

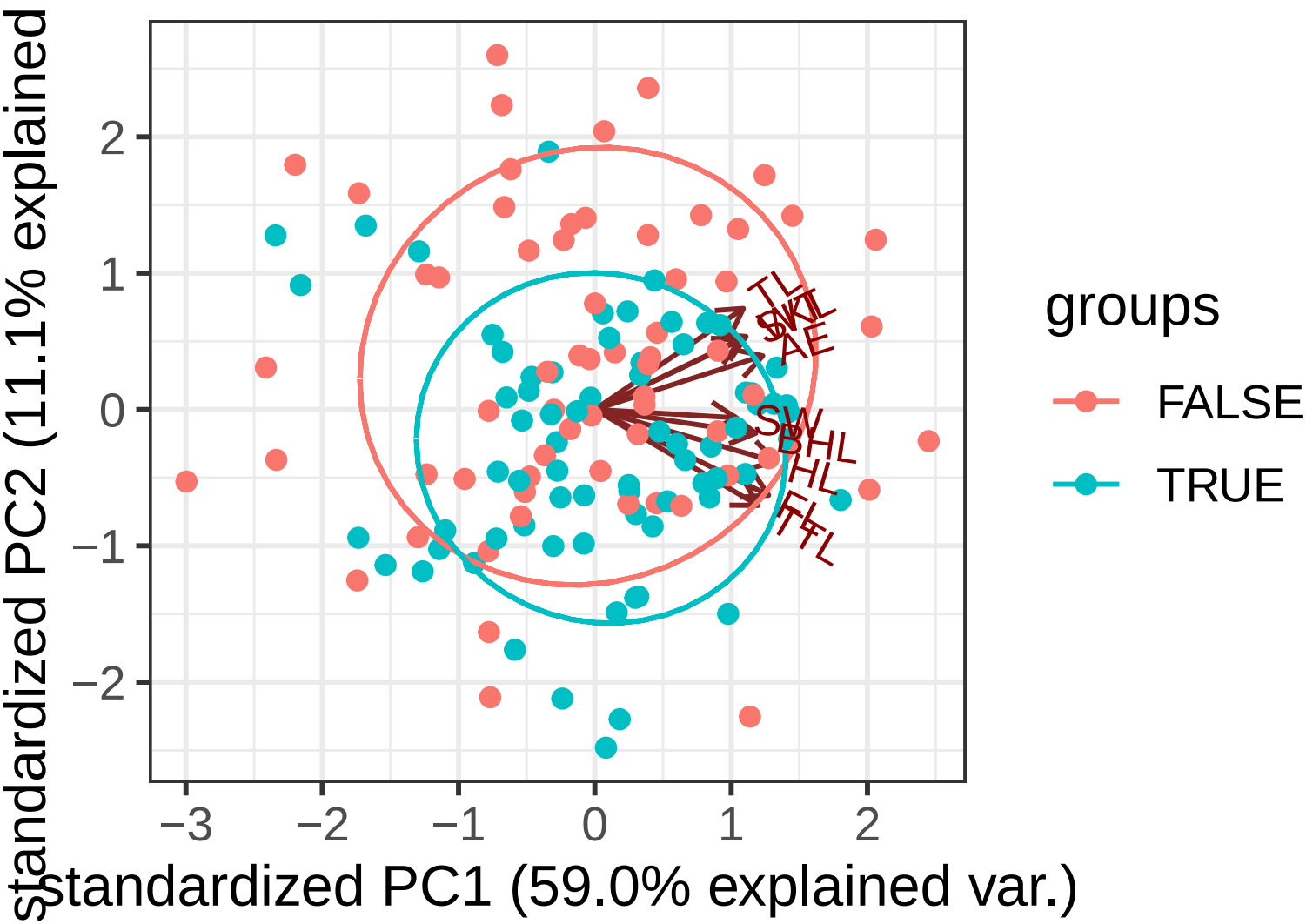
```
ggbiplot(sparrow.pca, labels = rownames(sparrow))
```



# Fun With Plotting

Add groups (like dead or alive birds) with `groups`.

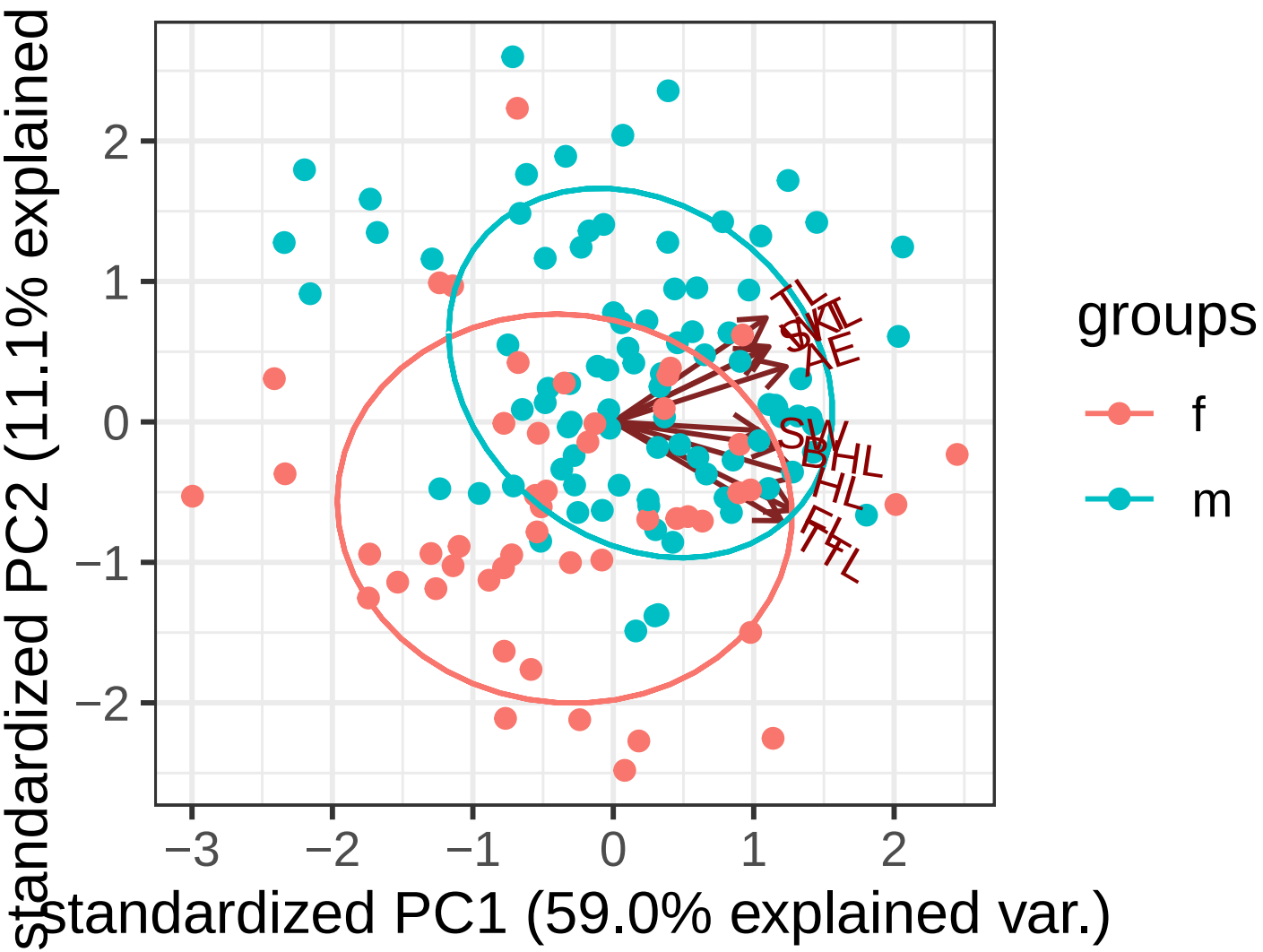
```
ggbiplot(sparrow.pca, ellipse = TRUE, groups = sparrow$survived)+
  theme_bw()+
  theme(text = element_text(size=12))
```



# Fun With Plotting

## Add groups (try sex of bird) with `groups`.

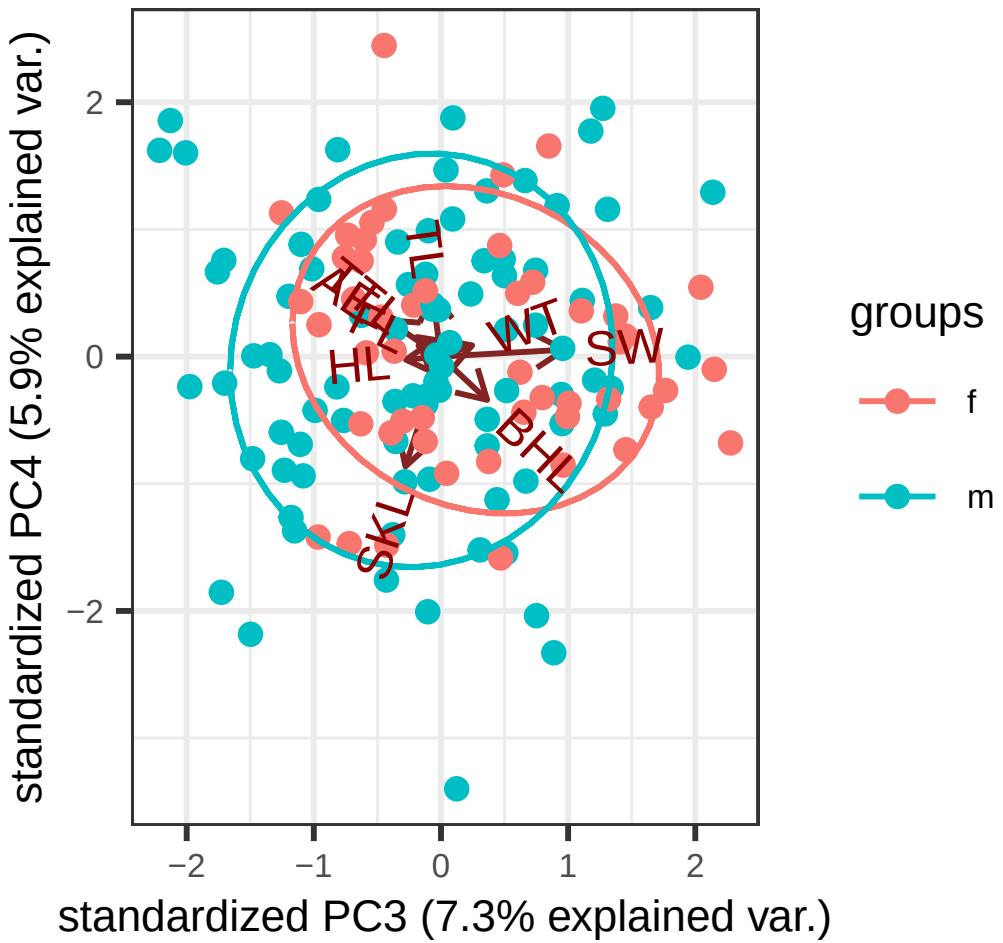
```
ggbiplot(sparrow.pca, ellipse = TRUE, groups = sparrow$sex)+
  theme_bw()+
  theme(text = element_text(size=12))
```



# Fun With Plotting

Try plotting PC3 and PC4 with `choices`.

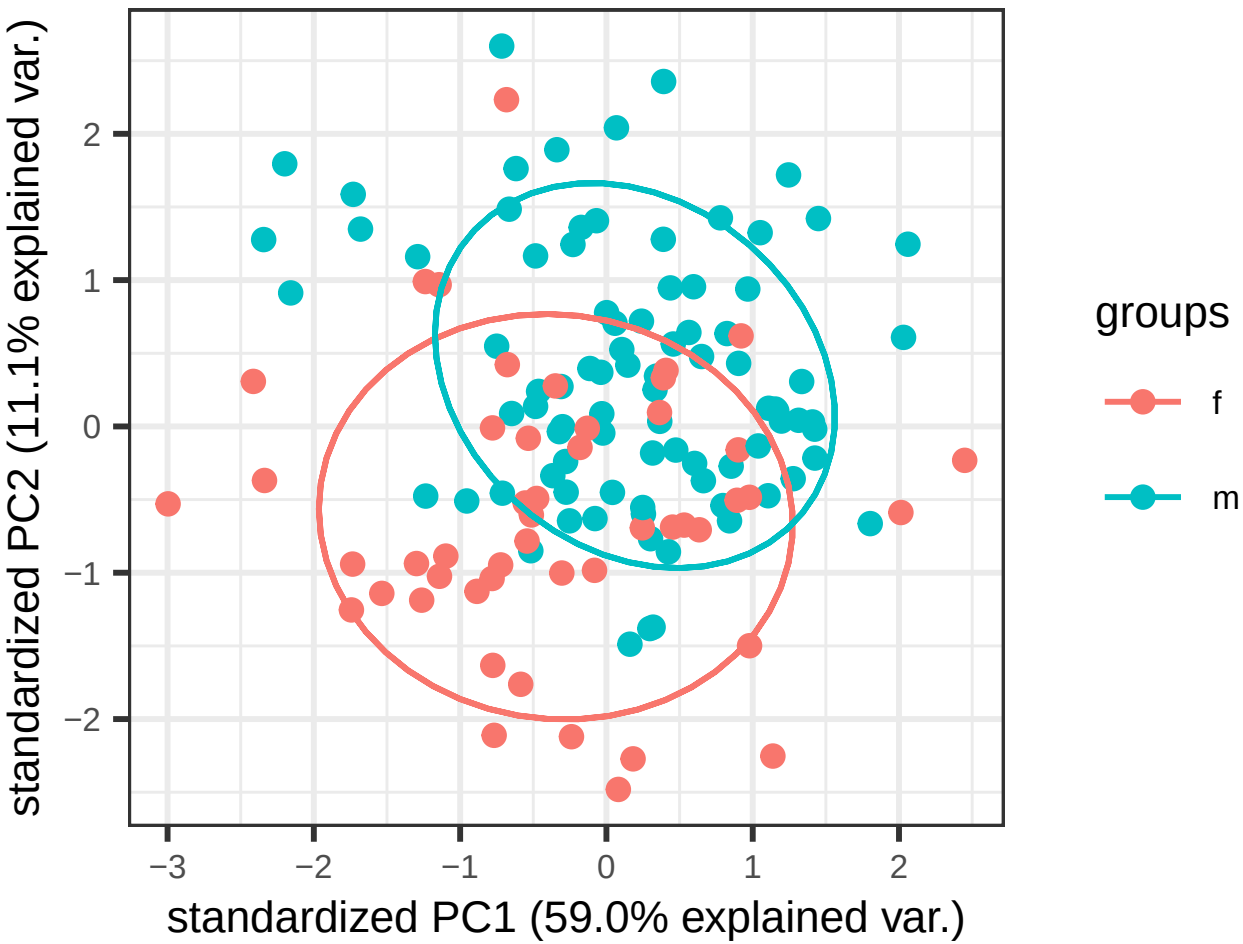
```
ggbiplot(sparrow.pca, ellipse = TRUE, choices = c(3,4),
         groups = sparrow$sex)+
  theme_bw()+
  theme(text = element_text(size=8))
```



# Fun With Plotting

You can remove the arrows with `var.axes`.

```
ggbiplot(sparrow.pca, ellipse = TRUE, var.axes = FALSE,
         groups = sparrow$sex)+
  theme_bw()+
  theme(text = element_text(size=8))
```



# What is PCA (and ordination in general) good for??

1. Visualization
2. Hypothesis creation
3. Dimension reduction

- Use PC scores as variables in other univariate models to account for correlated data