

Multivariate models

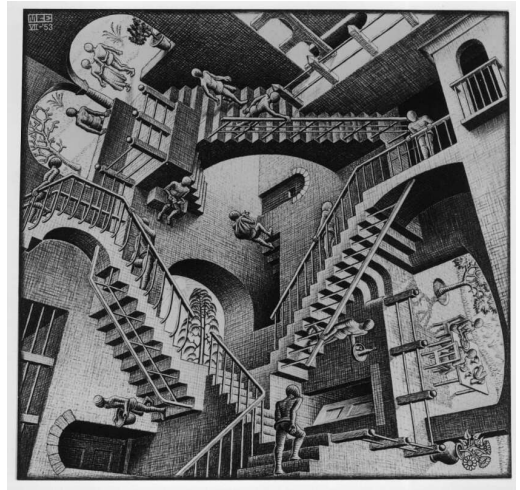
More than one way of seeing things

Samuel Robinson, Ph.D.

Oct 20, 2023

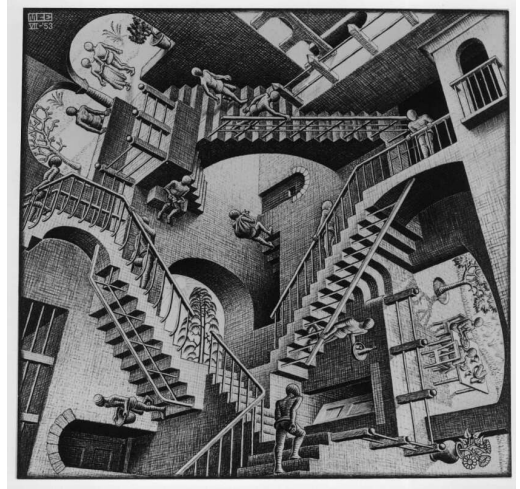
Outline

- What are multivariate data?



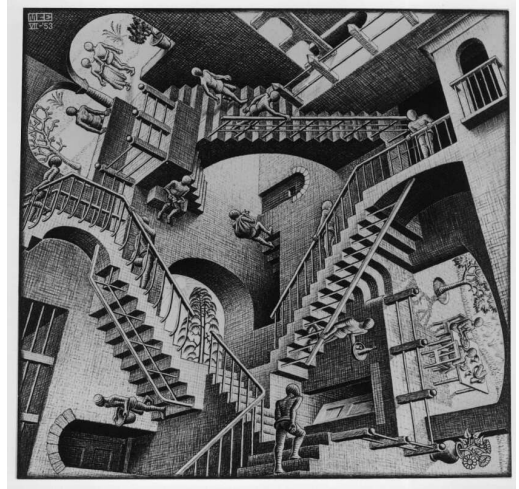
Outline

- What are multivariate data?
- Linear transformations



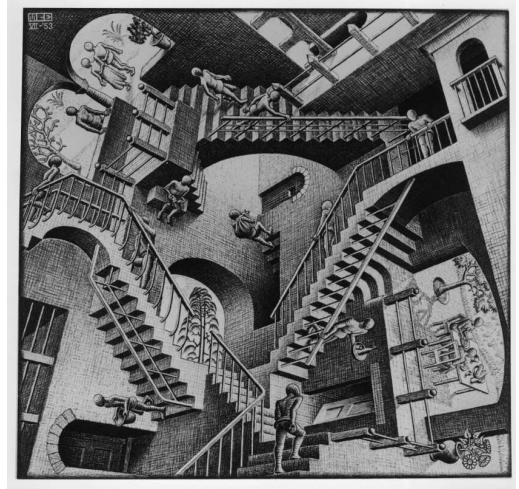
Outline

- What are multivariate data?
- Linear transformations
 - Principle components



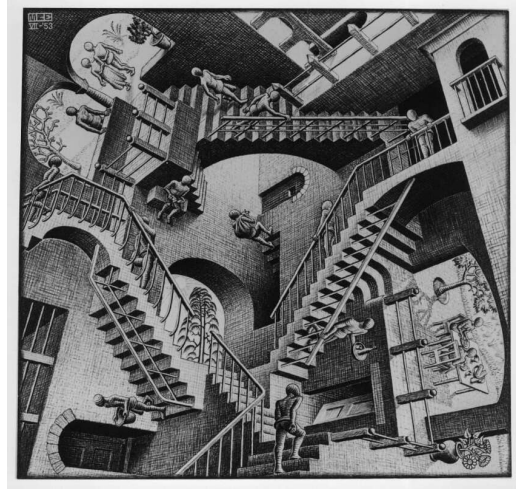
Outline

- What are multivariate data?
- Linear transformations
 - Principle components
 - Some common approaches



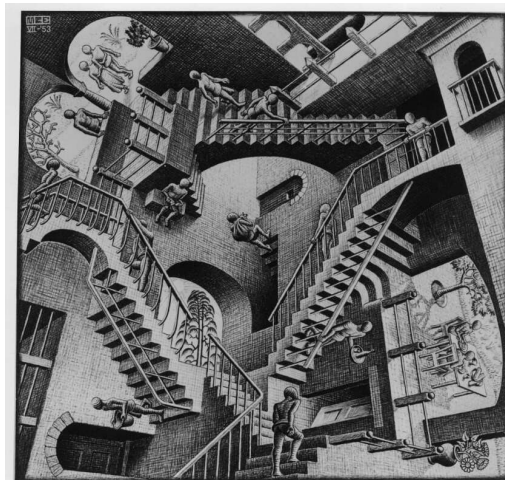
Outline

- What are multivariate data?
- Linear transformations
 - Principle components
 - Some common approaches
- Nonlinear transformations



Outline

- What are multivariate data?
- Linear transformations
 - Principle components
 - Some common approaches
- Nonlinear transformations
 - Non-metric dimensional scaling



Some common problems

- “I’ve got a zillion predictors that could matter in my model, but they’re all collinear”

Some common problems

- “I’ve got a zillion predictors that could matter in my model, but they’re all collinear”
- “I measured a zillion things for each site/critter, but I don’t want to fit a zillion models”

Some common problems

- “I’ve got a zillion predictors that could matter in my model, but they’re all collinear”
- “I measured a zillion things for each site/critter, but I don’t want to fit a zillion models”
- “I measured a zillion things. Do certain things group up into clusters?”

Some common problems

- “I’ve got a zillion predictors that could matter in my model, but they’re all collinear”
- “I measured a zillion things for each site/critter, but I don’t want to fit a zillion models”
- “I measured a zillion things. Do certain things group up into clusters?”
- “My supervisor told me to do a PCA or NMDS for my data, but I have no idea what they’re talking about”

Some common problems

- “I’ve got a zillion predictors that could matter in my model, but they’re all collinear”
- “I measured a zillion things for each site/critter, but I don’t want to fit a zillion models”
- “I measured a zillion things. Do certain things group up into clusters?”
- “My supervisor told me to do a PCA or NMDS for my data, but I have no idea what they’re talking about”

Some common problems

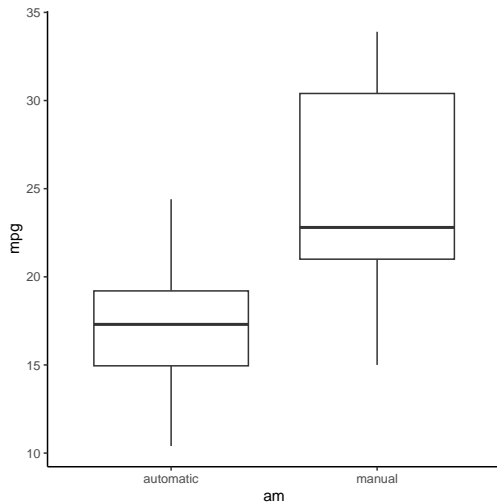
- “I’ve got a zillion predictors that could matter in my model, but they’re all collinear”
- “I measured a zillion things for each site/critter, but I don’t want to fit a zillion models”
- “I measured a zillion things. Do certain things group up into clusters?”
- “My supervisor told me to do a PCA or NMDS for my data, but I have no idea what they’re talking about”

If any of these sound like your situation, then you might need to do **multivariate modeling**!

Part 1: What are multivariate data?

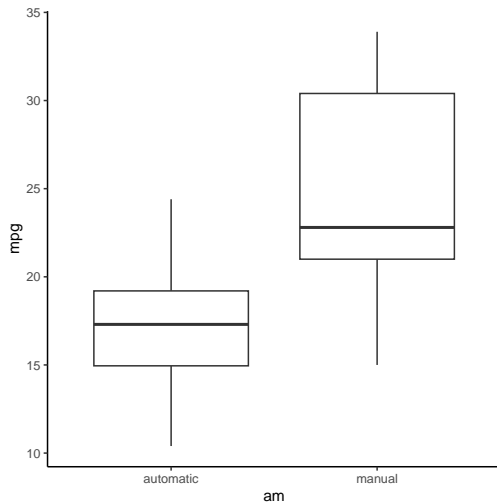
Univariate data

- Up until now, we've dealt mainly with **univariate** data: one thing is changing, and is being affected by other things



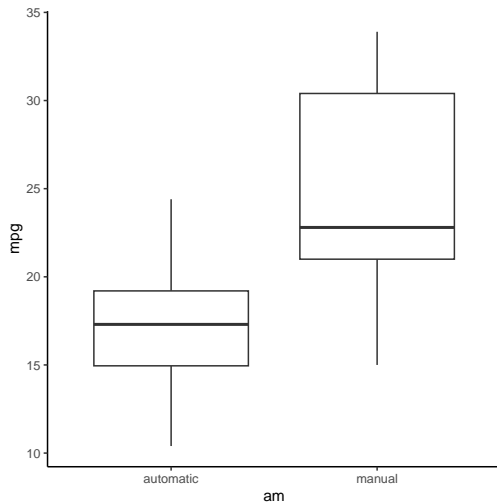
Univariate data

- Up until now, we've dealt mainly with **univariate** data: one thing is changing, and is being affected by other things
- These can be normal, binomial, Poisson, etc. . .



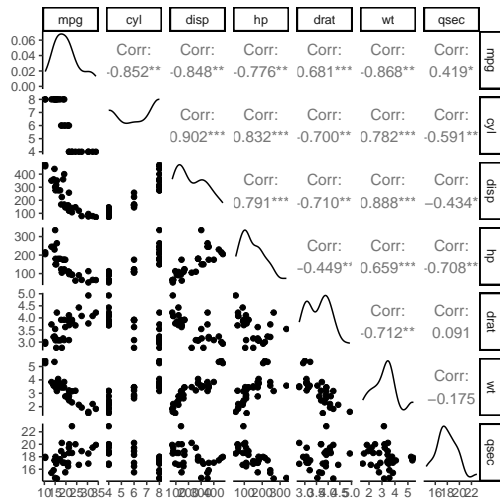
Univariate data

- Up until now, we've dealt mainly with **univariate** data: one thing is changing, and is being affected by other things
- These can be normal, binomial, Poisson, etc. . .
- Single variance term (σ) that controls dispersion



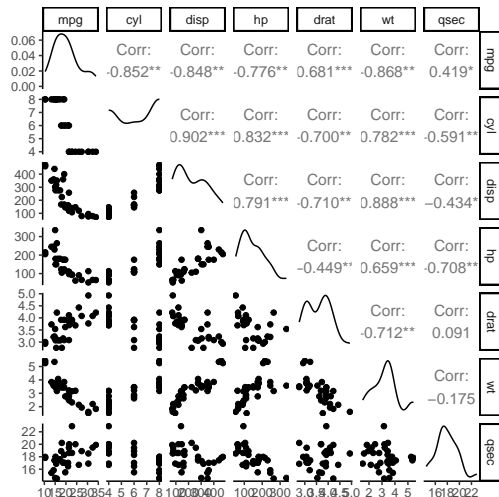
Multivariate data

- With **multivariate** data, we have multiple things changing at once



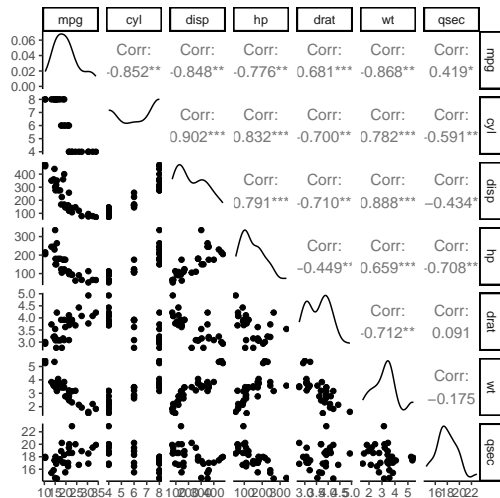
Multivariate data

- With **multivariate** data, we have multiple things changing at once
- *Many things* are changing, with multiple things potentially causing other things



Multivariate data

- With **multivariate** data, we have multiple things changing at once
- *Many things* are changing, with multiple things potentially causing other things
- These are *mostly* normal (non-normal can be tricky)



Multivariate normal

- Normal distributions don't just have a single σ , but actually a *matrix* of values

$$Y \sim \text{Normal}(M, \Sigma)$$

$$Y = \begin{bmatrix} y_{1a} & y_{1b} & y_{1c} \\ y_{2a} & y_{2b} & y_{2c} \\ \vdots & \vdots & \vdots \\ y_{na} & y_{nb} & y_{nc} \end{bmatrix}$$

$$M = [\mu_1, \mu_2, \mu_3]$$

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

Multivariate normal

- Normal distributions don't just have a single σ , but actually a *matrix* of values
- If the columns of our data ($Y = [y_1, y_2, y_3]$) are *independent*, then it looks like this:

$$Y \sim \text{Normal}(M, \Sigma)$$

$$Y = \begin{bmatrix} y_{1a} & y_{1b} & y_{1c} \\ y_{2a} & y_{2b} & y_{2c} \\ \vdots & \vdots & \vdots \\ y_{na} & y_{nb} & y_{nc} \end{bmatrix}$$

$$M = [\mu_1, \mu_2, \mu_3]$$

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

Multivariate normal

- Normal distributions don't just have a single σ , but actually a *matrix* of values
- If the columns of our data ($Y = [y_1, y_2, y_3]$) are *independent*, then it looks like this:
- Zeros mean “ μ_1, μ_2, μ_3 aren't related to each other”

$$Y \sim \text{Normal}(M, \Sigma)$$

$$Y = \begin{bmatrix} y_{1a} & y_{1b} & y_{1c} \\ y_{2a} & y_{2b} & y_{2c} \\ \vdots & \vdots & \vdots \\ y_{na} & y_{nb} & y_{nc} \end{bmatrix}$$

$$M = [\mu_1, \mu_2, \mu_3]$$

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

Multivariate normal

- Normal distributions don't just have a single σ , but actually a *matrix* of values
- If the columns of our data ($Y = [y_1, y_2, y_3]$) are *independent*, then it looks like this:
- Zeros mean “ μ_1, μ_2, μ_3 aren't related to each other”
- Diagonal elements = *variance*,
off-diagonal = *covariance*

$$Y \sim \text{Normal}(M, \Sigma)$$

$$Y = \begin{bmatrix} y_{1a} & y_{1b} & y_{1c} \\ y_{2a} & y_{2b} & y_{2c} \\ \vdots & \vdots & \vdots \\ y_{na} & y_{nb} & y_{nc} \end{bmatrix}$$

$$M = [\mu_1, \mu_2, \mu_3]$$

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

Covariance and Correlation

Things may not be independent from each other. For example:

- $\sigma = 2$ (variance = $\sigma^2 = 4$)

$$R = \begin{bmatrix} 1 & 0.7 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_a & \sigma_{ab} & \sigma_{ac} \\ \sigma_{ab} & \sigma_b & \sigma_{bc} \\ \sigma_{ac} & \sigma_{bc} & \sigma_c \end{bmatrix} = \begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Covariance and Correlation

Things may not be independent from each other. For example:

- $\sigma = 2$ (variance = $\sigma^2 = 4$)
- μ_1 and μ_2 are strongly correlated ($r=0.7$), but μ_3 is not related to anything ($r=0$).
Shown here as a *correlation matrix* (R):

$$R = \begin{bmatrix} 1 & 0.7 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \sigma_a & \sigma_{ab} & \sigma_{ac} \\ \sigma_{ab} & \sigma_b & \sigma_{bc} \\ \sigma_{ac} & \sigma_{bc} & \sigma_c \end{bmatrix} = \begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Covariance and Correlation

Things may not be independent from each other. For example:

- $\sigma = 2$ (variance = $\sigma^2 = 4$)
- μ_1 and μ_2 are strongly correlated ($r=0.7$), but μ_3 is not related to anything ($r=0$).
Shown here as a *correlation matrix* (R):

$$R = \begin{bmatrix} 1 & 0.7 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- When multiplied by the variance, this becomes the *covariance matrix* (Σ)

$$\Sigma = \begin{bmatrix} \sigma_a & \sigma_{ab} & \sigma_{ac} \\ \sigma_{ab} & \sigma_b & \sigma_{bc} \\ \sigma_{ac} & \sigma_{bc} & \sigma_c \end{bmatrix} = \begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Covariance vs Correlation

These are similar concepts, but covariance matrix has *units*, while correlation is *dimensionless*

$$\text{Covariance} = \sum_{i=1}^n \frac{(x-\bar{x})(y-\bar{y})}{(n-1)}$$

$$\text{Correlation} = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$$

$$\text{Covariance matrix} = \begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

$$\text{Correlation matrix} = \begin{bmatrix} 1 & 0.7 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

How does this help with my data?

- Say you've measured a bunch of things, and they're mostly from normal distributions. . .

$$Y \sim \text{Normal}(\textcolor{brown}{M}, \textcolor{red}{\Sigma})$$

$$\textcolor{brown}{M} = [\mu_1, \mu_2, \mu_3]$$

$$\textcolor{brown}{\mu}_1 = b_{01} + b_{11}x_1$$

$$\textcolor{brown}{\mu}_2 = b_{02} + b_{12}x_1$$

$$\textcolor{brown}{\mu}_3 = b_{03} + b_{13}x_1$$

How does this help with my data?

- Say you've measured a bunch of things, and they're mostly from normal distributions. . .
- You've gathered data from a *multivariate normal distribution*!

$$Y \sim \text{Normal}(\textcolor{brown}{M}, \textcolor{red}{\Sigma})$$

$$\textcolor{brown}{M} = [\mu_1, \mu_2, \mu_3]$$

$$\mu_1 = b_{01} + b_{11}x_1$$

$$\mu_2 = b_{02} + b_{12}x_1$$

$$\mu_3 = b_{03} + b_{13}x_1$$

How does this help with my data?

- Say you've measured a bunch of things, and they're mostly from normal distributions. . .
- You've gathered data from a *multivariate normal distribution*!
- Now your task is to model this distribution!

$$Y \sim \text{Normal}(\boldsymbol{M}, \boldsymbol{\Sigma})$$

$$\boldsymbol{M} = [\mu_1, \mu_2, \mu_3]$$

$$\mu_1 = b_{01} + b_{11}x_1$$

$$\mu_2 = b_{02} + b_{12}x_1$$

$$\mu_3 = b_{03} + b_{13}x_1$$

Problem: this doesn't really help

- We're *still* stuck with fitting a zillion linear models!



Problem: this doesn't really help

- We're *still* stuck with fitting a zillion linear models!
- We also have to estimate the covariance as well as the variance. This might be OK for a few columns, but gets much harder when you've got lots of columns



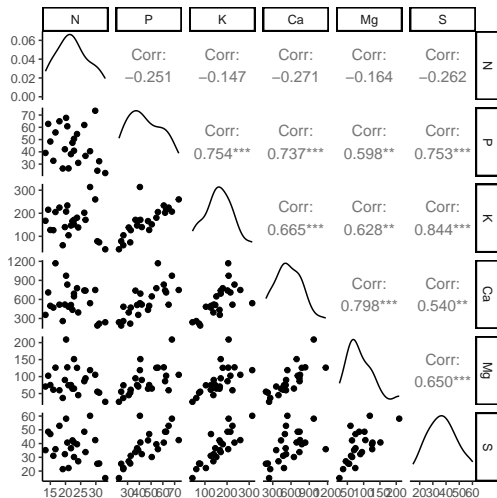
Problem: this doesn't really help

- We're *still* stuck with fitting a zillion linear models!
- We also have to estimate the covariance as well as the variance. This might be OK for a few columns, but gets much harder when you've got lots of columns
- We need a better way for dealing with these multivariate normal data...

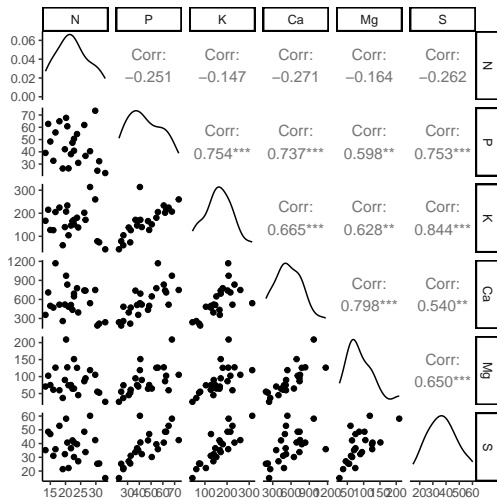


Another approach

- Say we have a multi-column dataset that looks like this:

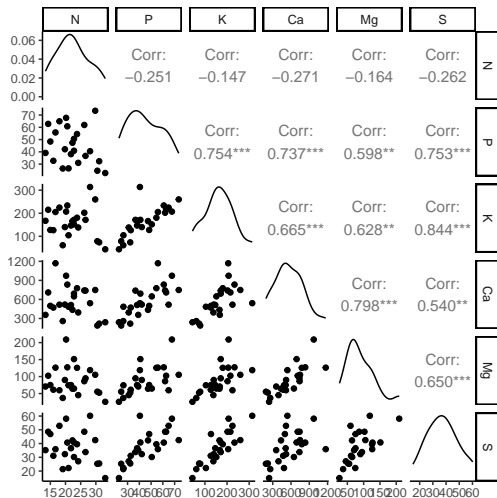


Another approach



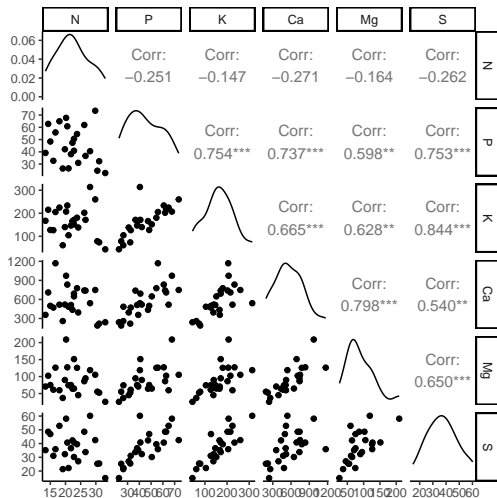
- Say we have a multi-column dataset that looks like this:
- What do you notice about this dataset?

Another approach



- Say we have a multi-column dataset that looks like this:
- What do you notice about this dataset?
- Looks like most of these columns are pretty strongly related. If we're only interested in the total "information" (variation) from this dataset. . .

Another approach



- Say we have a multi-column dataset that looks like this:
- What do you notice about this dataset?
- Looks like most of these columns are pretty strongly related. If we're only interested in the total "information" (variation) from this dataset. . .
- Perhaps we don't need all these columns? Which ones should we throw out or combine?

Part 2: Principal components (linear decompositions)

Matrix Decomposition and Principal Component

- Covariance matrices are a special type of matrix called a *triangular matrix*

Covariance matrix

##	N	P	K	Ca	Mg	S
## N	30.6	-20.8	-52.6	-364.8	-37.1	-16.9
## P	-20.8	223.4	730.8	2683.9	366.5	131.3
## K	-52.6	730.8	4204.5	10500.6	1669.4	638.4
## Ca	-364.8	2683.9	10500.6	59332.2	7974.5	1533.4
## Mg	-37.1	366.5	1669.4	7974.5	1681.9	311.2
## S	-16.9	131.3	638.4	1533.4	311.2	136.1

Decomposition: $X = UDV'$

Eigenvectors (V):

##	PC1	PC2	PC3	PC4	PC5	PC6
## N	0.01	-0.01	0.02	-0.16	0.83	-0.54
## P	-0.04	-0.10	-0.06	0.88	0.35	0.28
## K	-0.18	-0.95	-0.15	-0.16	0.02	0.08
## Ca	-0.97	0.20	-0.11	-0.02	-0.01	-0.03
## Mg	-0.13	-0.11	0.98	-0.01	0.05	0.11
## S	-0.03	-0.16	0.09	0.41	-0.43	-0.78

Eigenvalues (D):

## [1]	62550.01	2371.69	571.21	82.81	31.36	15.21
--------	----------	---------	--------	-------	-------	-------

Matrix Decomposition and Principal Component

- Covariance matrices are a special type of matrix called a *triangular matrix*
- Can be decomposed using a math trick called the *singular value decomposition* that breaks a matrix into its component eigenvectors and eigenvalues

Covariance matrix

##	N	P	K	Ca	Mg	S
## N	30.6	-20.8	-52.6	-364.8	-37.1	-16.9
## P	-20.8	223.4	730.8	2683.9	366.5	131.3
## K	-52.6	730.8	4204.5	10500.6	1669.4	638.4
## Ca	-364.8	2683.9	10500.6	59332.2	7974.5	1533.4
## Mg	-37.1	366.5	1669.4	7974.5	1681.9	311.2
## S	-16.9	131.3	638.4	1533.4	311.2	136.1

Decomposition: $X = UDV'$

Eigenvectors (V):

##	PC1	PC2	PC3	PC4	PC5	PC6
## N	0.01	-0.01	0.02	-0.16	0.83	-0.54
## P	-0.04	-0.10	-0.06	0.88	0.35	0.28
## K	-0.18	-0.95	-0.15	-0.16	0.02	0.08
## Ca	-0.97	0.20	-0.11	-0.02	-0.01	-0.03
## Mg	-0.13	-0.11	0.98	-0.01	0.05	0.11
## S	-0.03	-0.16	0.09	0.41	-0.43	-0.78

Eigenvalues (D):

## [1]	62550.01	2371.69	571.21	82.81	31.36	15.21
--------	----------	---------	--------	-------	-------	-------

Matrix Decomposition and Principal Component

- Covariance matrices are a special type of matrix called a *triangular matrix*
- Can be decomposed using a math trick called the *singular value decomposition* that breaks a matrix into its component eigenvectors and eigenvalues
- Transforms data into new coordinate space, where *most variation falls into a few columns* called **principal components**

Covariance matrix

##	N	P	K	Ca	Mg	S
## N	30.6	-20.8	-52.6	-364.8	-37.1	-16.9
## P	-20.8	223.4	730.8	2683.9	366.5	131.3
## K	-52.6	730.8	4204.5	10500.6	1669.4	638.4
## Ca	-364.8	2683.9	10500.6	59332.2	7974.5	1533.4
## Mg	-37.1	366.5	1669.4	7974.5	1681.9	311.2
## S	-16.9	131.3	638.4	1533.4	311.2	136.1

Decomposition: $X = UDV'$

Eigenvectors (V):

##	PC1	PC2	PC3	PC4	PC5	PC6
## N	0.01	-0.01	0.02	-0.16	0.83	-0.54
## P	-0.04	-0.10	-0.06	0.88	0.35	0.28
## K	-0.18	-0.95	-0.15	-0.16	0.02	0.08
## Ca	-0.97	0.20	-0.11	-0.02	-0.01	-0.03
## Mg	-0.13	-0.11	0.98	-0.01	0.05	0.11
## S	-0.03	-0.16	0.09	0.41	-0.43	-0.78

Eigenvalues (D):

## [1]	62550.01	2371.69	571.21	82.81	31.36	15.21
--------	----------	---------	--------	-------	-------	-------

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:
- Rotation Matrix (V):

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:
- Rotation Matrix (V):

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:
- Rotation Matrix (V):

```
##      PC1    PC2
## P  0.81   0.59
## S  0.59  -0.81
```

- Columns = *Principal components*

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:
- Rotation Matrix (V):

```
##      PC1    PC2
## P  0.81   0.59
## S  0.59  -0.81
```

- Columns = *Principal components*
- Rows = *Factor Loadings*

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:
- Rotation Matrix (V):

```
##      PC1    PC2
## P  0.81   0.59
## S  0.59  -0.81
```

- Columns = *Principal components*
- Rows = *Factor Loadings*
- SD of principal components (\sqrt{D}):

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:
- Rotation Matrix (V):

```
##      PC1    PC2
## P 0.81   0.59
## S 0.59  -0.81
```

- Columns = *Principal components*
- Rows = *Factor Loadings*
- SD of principal components (\sqrt{D}):

Simple example: 2 dimensions

- Principal components are hard to imagine, so let's try it on only 2 dimensions:

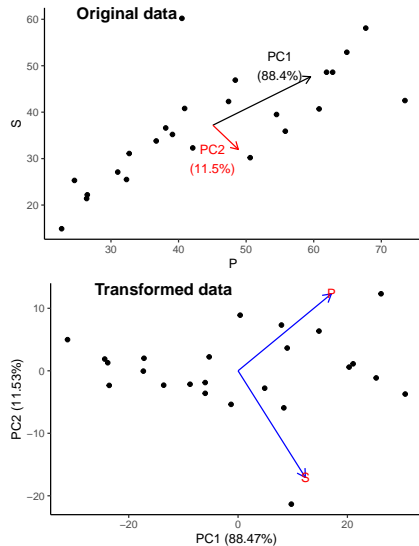
- Rotation Matrix (V):

```
##      PC1    PC2
## P  0.81  0.59
## S  0.59 -0.81
```

- Columns = *Principal components*
- Rows = *Factor Loadings*
- SD of principal components (\sqrt{D}):

```
## [1] 17.84  6.44
```

- Tells you *how strong* the effect of each PC column is



Bigger example: full dataset (14 columns)

Use prcomp to decompose matrix of varechem data:

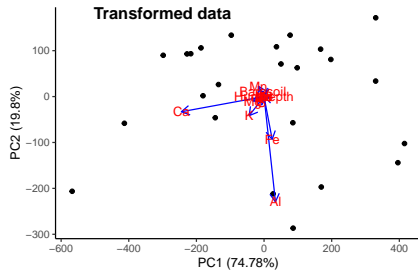
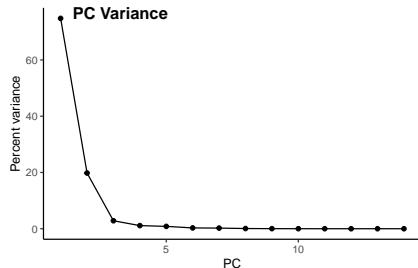
```
pcVare <- prcomp(varechem)
```

Rotation matrix (PCs and factor loadings)

##	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	PC11
## N	0.01	0.00	-0.01	0.06	0.01	0.03	0.09	0.16	-0.95	-0.21	0.15
## P	-0.04	-0.03	-0.09	0.03	-0.07	0.17	-0.16	-0.89	-0.22	0.30	0.01
## K	-0.17	-0.16	-0.86	0.03	0.13	-0.42	-0.05	0.05	-0.02	0.06	-0.02
## Ca	-0.96	-0.13	0.22	0.01	-0.11	-0.04	0.03	0.02	0.00	-0.02	0.00
## Mg	-0.13	-0.04	-0.05	0.02	0.84	0.47	-0.14	0.10	0.00	0.13	0.01
## S	-0.02	-0.05	-0.12	-0.06	0.09	0.11	-0.06	-0.30	0.18	-0.88	0.24
## Al	0.13	-0.90	0.02	-0.32	-0.13	0.20	0.04	0.07	-0.01	0.05	0.00
## Fe	0.09	-0.37	0.19	0.86	0.13	-0.21	0.09	-0.09	0.05	-0.05	-0.02
## Mn	-0.07	0.09	-0.38	0.37	-0.43	0.68	0.07	0.19	0.08	0.02	0.00
## Zn	-0.01	0.00	-0.01	-0.02	0.02	0.06	-0.05	-0.04	-0.10	-0.25	-0.95
## Mo	0.00	0.00	0.00	0.00	0.00	0.01	0.00	-0.01	0.00	-0.01	-0.02
## Baresoil	-0.01	0.05	-0.08	-0.10	0.15	0.04	0.96	-0.18	0.04	0.03	-0.05
## Humdepth	0.00	0.00	0.00	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.14
## pH	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.01	0.01	-0.01

SDs of principal components:

```
## [1] 253.1 130.2 49.2 30.9 26.8 15.8 14.0 7.8 4.8 3.1 1.2 0.4  
## [13] 0.1 0.1
```

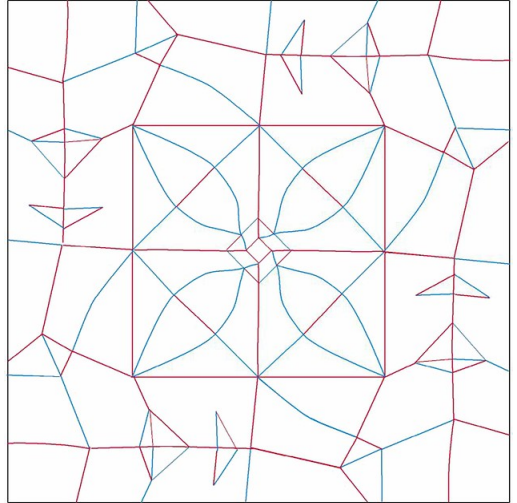


Artistic analogues to this problem

Picasso's *Demoiselle d'Avignon* (1907)



Kawasaki rose crease pattern



First challenge

Let's try this on some biological data

"After a severe storm on February 1, 1898, a number of moribund sparrows were taken to Hermon Bumpus' biological laboratory at Brown University, Rhode Island. Subsequently, about half of the birds died, and Bumpus saw this as an opportunity to see whether he could find any support for Charles Darwin's theory of natural selection. . . "

- Take a look at the bird dataset found [here](#)), and perform a PCA decomposition

##	Survived	Bird	Total_length	Alar_length	BeakHead_Length	Humerus_length
## 1	Yes	1	156	245	31.6	18.5
## 2	Yes	2	154	240	30.4	17.9
## 3	Yes	3	153	240	31.0	18.4
## 4	Yes	4	153	236	30.9	17.7
## 5	Yes	5	155	243	31.5	18.6
## 6	Yes	6	163	247	32.0	19.0

##	Keel_length
## 1	20.5
## 2	19.6
## 3	20.6
## 4	20.2
## 5	20.3
## 6	20.2

First challenge

Let's try this on some biological data

"After a severe storm on February 1, 1898, a number of moribund sparrows were taken to Hermon Bumpus' biological laboratory at Brown University, Rhode Island. Subsequently, about half of the birds died, and Bumpus saw this as an opportunity to see whether he could find any support for Charles Darwin's theory of natural selection. . . "

```
##   Survived Bird Total_length Alar_length BeakHead_Length Humerus_length
## 1      Yes    1          156         245             31.6           18.5
## 2      Yes    2          154         240             30.4           17.9
## 3      Yes    3          153         240             31.0           18.4
## 4      Yes    4          153         236             30.9           17.7
## 5      Yes    5          155         243             31.5           18.6
## 6      Yes    6          163         247             32.0           19.0
##   Keel_length
## 1         20.5
## 2         19.6
## 3         20.6
## 4         20.2
## 5         20.3
## 6         20.2
```

- Take a look at the bird dataset found [here](#)), and perform a PCA decomposition
- Hint: you'll need to transform it into a *matrix* (using `as.matrix` on the relevant columns) before using `prcomp`

First challenge

Let's try this on some biological data

"After a severe storm on February 1, 1898, a number of moribund sparrows were taken to Hermon Bumpus' biological laboratory at Brown University, Rhode Island. Subsequently, about half of the birds died, and Bumpus saw this as an opportunity to see whether he could find any support for Charles Darwin's theory of natural selection. . . "

```
## Survived Bird Total_length Alar_length BeakHead_Length Humerus_length
## 1 Yes 1 156 245 31.6 18.5
## 2 Yes 2 154 240 30.4 17.9
## 3 Yes 3 153 240 31.0 18.4
## 4 Yes 4 153 236 30.9 17.7
## 5 Yes 5 155 243 31.5 18.6
## 6 Yes 6 163 247 32.0 19.0
## Keel_length
## 1 20.5
## 2 19.6
## 3 20.6
## 4 20.2
## 5 20.3
## 6 20.9
```

- Take a look at the bird dataset found [here](#)), and perform a PCA decomposition
- Hint: you'll need to transform it into a *matrix* (using `as.matrix` on the relevant columns) before using `prcomp`
- How many PCs are needed to represent *most* of the variation?

First challenge results

Covariance matrix

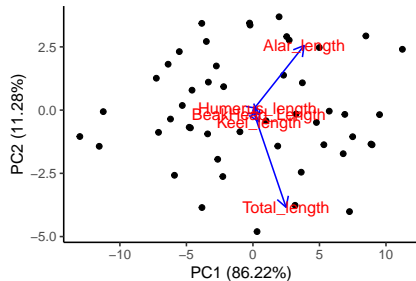
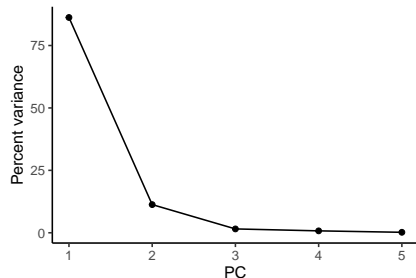
```
##              Total_length Alar_length BeakHead_Length Humerus_length
## Total_length          13.4         13.6           1.9           1.3
## Alar_length           13.6         25.7           2.7           2.2
## BeakHead_Length       1.9          2.7           0.6           0.3
## Humerus_length        1.3          2.2           0.3           0.3
## Keel_length           2.2          2.7           0.4           0.3
##
##              Keel_length
## Total_length          2.2
## Alar_length           2.7
## BeakHead_Length       0.4
## Humerus_length        0.3
## Keel_length           1.0
```

Principal components

```
##              PC1   PC2   PC3   PC4   PC5
## Total_length  0.54 -0.83 -0.16 -0.04 -0.02
## Alar_length   0.83  0.55 -0.06 -0.07  0.04
## BeakHead_Length 0.10 -0.03  0.24  0.90  0.36
## Humerus_length 0.07  0.01  0.20  0.31 -0.93
## Keel_length   0.10 -0.10  0.94 -0.31  0.11
```

Variance per column

```
## [1] 34.81  4.41  0.64  0.36  0.09
```



What's next?

- Now that we've reduced our data to only a few *uncorrelated* columns, we can do a couple things:

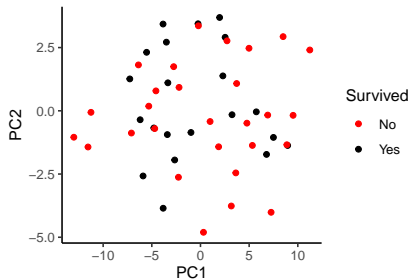
What's next?

- Now that we've reduced our data to only a few *uncorrelated* columns, we can do a couple things:
- Use linear regression (or some other test) on each column, along with some other set of predictor columns

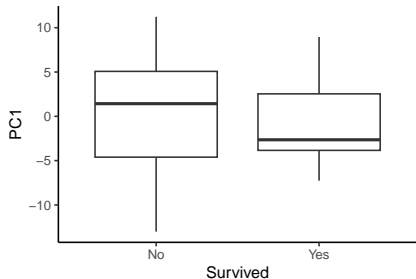
What's next?

- Now that we've reduced our data to only a few *uncorrelated* columns, we can do a couple things:
- Use linear regression (or some other test) on each column, along with some other set of predictor columns
- Use some other test to see if your data are “different” (far away) from each other

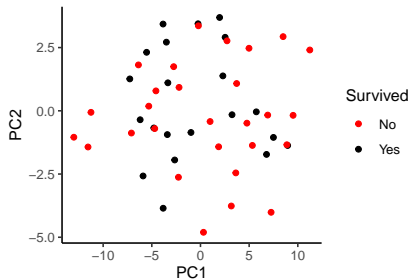
First step: plot your data



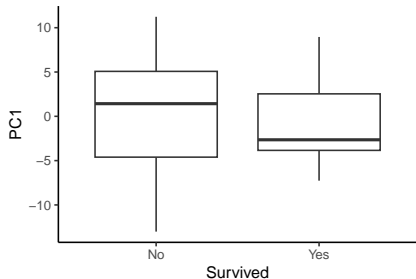
- Usually we want to see if some other thing is changing our data “somehow” (usually using a linear model)



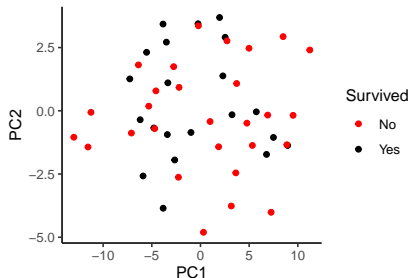
First step: plot your data



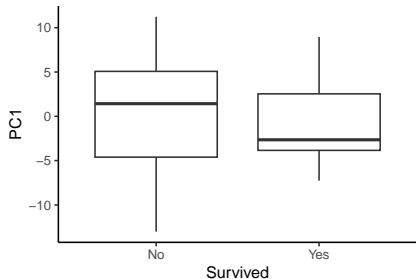
- Usually we want to see if some other thing is changing our data “somehow” (usually using a linear model)
- The first step is to plot your data. I’ve been using autoplot from ggfortify, but we can also plot things manually



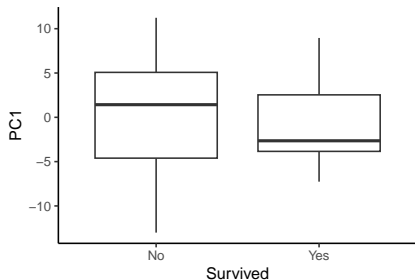
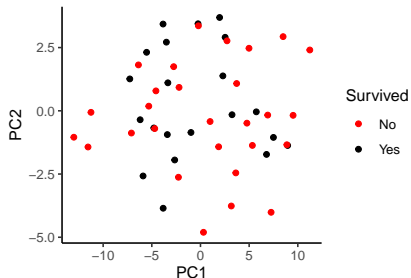
First step: plot your data



- Usually we want to see if some other thing is changing our data “somehow” (usually using a linear model)
- The first step is to plot your data. I’ve been using autoplot from ggfortify, but we can also plot things manually
- These data don’t look particularly different in their centers, but...



First step: plot your data



- Usually we want to see if some other thing is changing our data “somehow” (usually using a linear model)
- The first step is to plot your data. I’ve been using autoplot from ggfortify, but we can also plot things manually
- These data don’t look particularly different in their centers, but...
- They are different in variances! What does this mean biologically?

Formal tests for differences

- `envfit` from the `vegan` package is a common way to test for differences between groups (or continuous variables)

Formal tests for differences

- `envfit` from the `vegan` package is a common way to test for differences between groups (or continuous variables)
- Uses a permutation test to compare “jumbled” data original data. In this case, shows no difference in group averages

Formal tests for differences

- `envfit` from the `vegan` package is a common way to test for differences between groups (or continuous variables)
- Uses a permutation test to compare “jumbled” data original data. In this case, shows no difference in group averages

Formal tests for differences

- `envfit` from the `vegan` package is a common way to test for differences between groups (or continuous variables)
- Uses a permutation test to compare “jumbled” data original data. In this case, shows no difference in group averages
- `rda` and `cca` from `vegan` are also popular (older) methods. They basically fit linear regression to each row in the data matrix, then decompose the fitted values

```
envfit(birdPCA ~ Survived,data=birds)
```

```
##
## ***FACTORS:
##
## Centroids:
##           PC1      PC2
## SurvivedNo  0.4453 -0.2392
## SurvivedYes -0.5938  0.3190
##
## Goodness of fit:
##           r2 Pr(>r)
## Survived 0.0087 0.606
## Permutation: free
## Number of permutations: 999
```

Formal tests for differences

- `envfit` from the `vegan` package is a common way to test for differences between groups (or continuous variables)
- Uses a permutation test to compare “jumbled” data original data. In this case, shows no difference in group averages

```
envfit(birdPCA ~ Survived,data=birds)
```

```
##
## ***FACTORS:
##
## Centroids:
##           PC1      PC2
## SurvivedNo  0.4453 -0.2392
## SurvivedYes -0.5938  0.3190
##
## Goodness of fit:
##           r2 Pr(>r)
## Survived 0.0087  0.606
## Permutation: free
## Number of permutations: 999
```

- `rda` and `cca` from `vegan` are also popular (older) methods. They basically fit linear regression to each row in the data matrix, then decompose the fitted values
- “Constrained” part is related to variables you provided, “Unconstrained” is leftover variance

Formal tests for differences

- `envfit` from the `vegan` package is a common way to test for differences between groups (or continuous variables)
- Uses a permutation test to compare “jumbled” data original data. In this case, shows no difference in group averages

```
envfit(birdPCA ~ Survived,data=birds)
```

```
##
## ***FACTORS:
##
## Centroids:
##           PC1      PC2
## SurvivedNo  0.4453 -0.2392
## SurvivedYes -0.5938  0.3190
##
## Goodness of fit:
##           r2 Pr(>r)
## Survived 0.0087  0.606
## Permutation: free
## Number of permutations: 999
```

- `rda` and `cca` from `vegan` are also popular (older) methods. They basically fit linear regression to each row in the data matrix, then decompose the fitted values
- “Constrained” part is related to variables you provided, “Unconstrained” is leftover variance

Formal tests for differences

- envfit from the vegan package is a common way to test for differences between groups (or continuous variables)
- Uses a permutation test to compare “jumbled” data original data. In this case, shows no difference in group averages

```
envfit(birdPCA ~ Survived,data=birds)
```

```
##
## ***FACTORS:
##
## Centroids:
##           PC1      PC2
## SurvivedNo  0.4453 -0.2392
## SurvivedYes -0.5938  0.3190
##
## Goodness of fit:
##           r2 Pr(>r)
## Survived  0.0087  0.606
## Permutation: free
## Number of permutations: 999
```

- rda and cca from vegan are also popular (older) methods. They basically fit linear regression to each row in the data matrix, then decompose the fitted values
- “Constrained” part is related to variables you provided, “Unconstrained” is leftover variance

```
rda(birdMat ~ Survived,data=birds)
```

```
## Call: rda(formula = birdMat ~ Survived, data = birds)
##
##              Inertia Proportion Rank
## Total              40.969439   1.000000
## Constrained        0.357460   0.008725    1
## Unconstrained    40.611979   0.991275    5
## Inertia is variance
##
## Eigenvalues for constrained axes:
##   RDA1
## 0.3575
##
## Eigenvalues for unconstrained axes:
##   PC1  PC2  PC3  PC4  PC5
## 35.06  4.54  0.62  0.31  0.08
```

Plotting your results

- For plotting your results, you generally want to show the *transformed data* plus the direction of the effects you're interested in

Plotting your results

- For plotting your results, you generally want to show the *transformed data* plus the direction of the effects you're interested in
- ggplot2 has some plotting functionality, but the plots from `vegan` are more purpose-made for these kinds of data

Plotting your results

- For plotting your results, you generally want to show the *transformed data* plus the direction of the effects you're interested in
- ggplot2 has some plotting functionality, but the plots from `vegan` are more purpose-made for these kinds of data
- See the `vegan` vignette (`vignette("intro-vegan")`) for more details

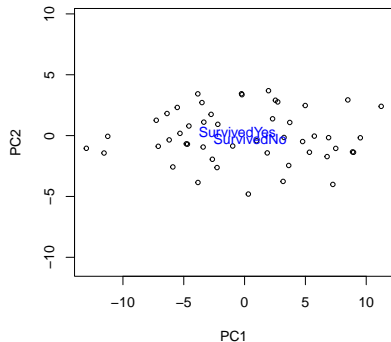
Plotting your results

- For plotting your results, you generally want to show the *transformed data* plus the direction of the effects you're interested in
- ggplot2 has some plotting functionality, but the plots from `vegan` are more purpose-made for these kinds of data
- See the `vegan` vignette (`vignette("intro-vegan")`) for more details

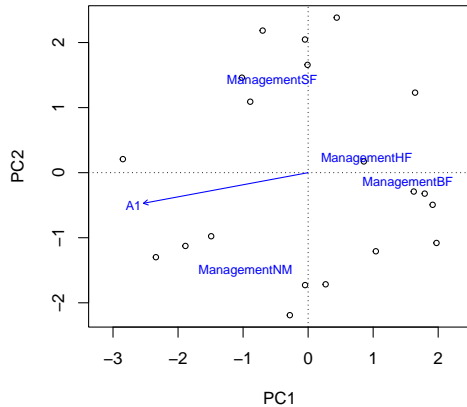
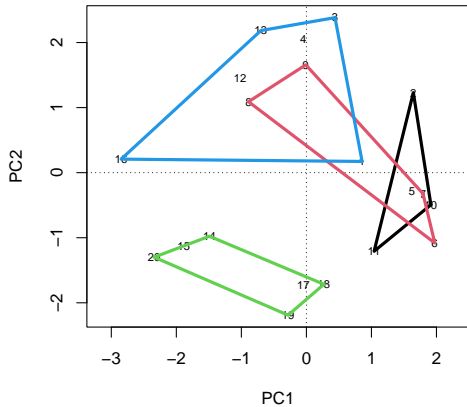
Plotting your results

- For plotting your results, you generally want to show the *transformed data* plus the direction of the effects you're interested in
- ggplot2 has some plotting functionality, but the plots from `vegan` are more purpose-made for these kinds of data
- See the `vegan` vignette (`vignette("intro-vegan")`) for more details

```
ordiplot(birdPCA,type='points',  
         display='sites')  
plot(envfit(birdPCA ~ Survived,data=birds))
```



Other kinds of ordination plots



Second challenge

- Let's try some *community* data (counts of different species)!

##	Date	Method	Aedes.spp	Agriades.glandon	Anthomyiidae
## 1	170	Bowl Trap Control	0	0	0
## 2	170	Bowl Trap OTC	0	0	0
## 3	170	Netting Control	0	0	0
## 4	174	Bowl Trap Control	0	0	0
## 5	174	Bowl Trap OTC	0	0	0
## 6	174	Netting Control	0	0	0

Second challenge

- Let's try some *community* data (counts of different species)!
- [Here](#) (bugDat.csv) is a dataset I collected during my Master's degree, which I spent catching a lot of bugs

##	Date	Method	Aedes.spp	Agriades.glandon	Anthomyiidae
## 1	170	Bowl Trap Control	0	0	0
## 2	170	Bowl Trap OTC	0	0	0
## 3	170	Netting Control	0	0	0
## 4	174	Bowl Trap Control	0	0	0
## 5	174	Bowl Trap OTC	0	0	0
## 6	174	Netting Control	0	0	0

Second challenge

- Let's try some *community* data (counts of different species)!
- [Here](#) (bugDat.csv) is a dataset I collected during my Master's degree, which I spent catching a lot of bugs
- I collected bugs using a couple kinds of collection methods across the season

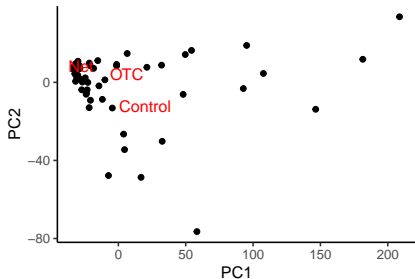
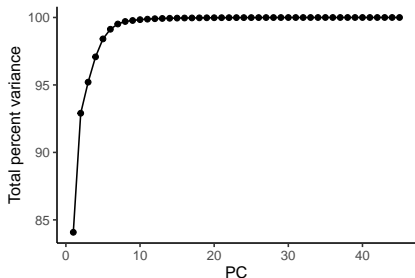
##	Date	Method	Aedes.spp	Agriades.glandon	Anthomyiidae
## 1	170	Bowl Trap Control	0	0	0
## 2	170	Bowl Trap OTC	0	0	0
## 3	170	Netting Control	0	0	0
## 4	174	Bowl Trap Control	0	0	0
## 5	174	Bowl Trap OTC	0	0	0
## 6	174	Netting Control	0	0	0

Second challenge

- Let's try some *community* data (counts of different species)!
- [Here](#) (bugDat.csv) is a dataset I collected during my Master's degree, which I spent catching a lot of bugs
- I collected bugs using a couple kinds of collection methods across the season
- Was there a large difference in collection methods? Maybe date of collection?

##	Date	Method	Aedes.spp	Agriades.glandon	Anthomyiidae
## 1	170	Bowl Trap Control	0	0	0
## 2	170	Bowl Trap OTC	0	0	0
## 3	170	Netting Control	0	0	0
## 4	174	Bowl Trap Control	0	0	0
## 5	174	Bowl Trap OTC	0	0	0
## 6	174	Netting Control	0	0	0

Second challenge results



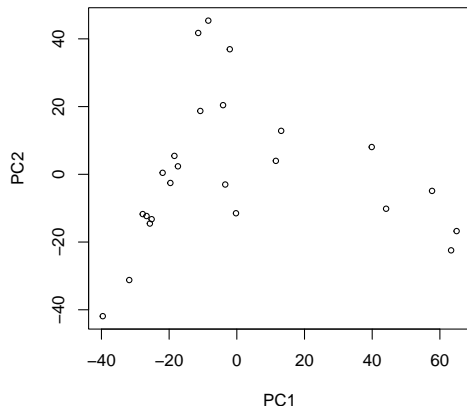
```
(bugFit <- envfit(bugPCA ~ Method, data=bugPreds))
```

```
##
## ***FACTORS:
##
## Centroids:
##              PC1      PC2
## MethodControl  19.8910 -12.1220
## MethodNet      -27.7803  8.3650
## MethodOTC       6.2679  3.9665
##
## Goodness of fit:
##              r2 Pr(>r)
## Method 0.1529  0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 999
```

Part 3: NMDS (nonlinear decompositions)

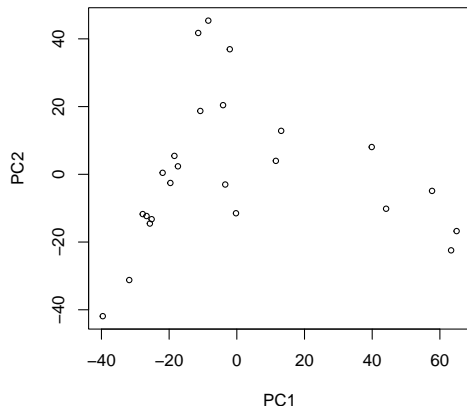
Problems with linear transformations

- Recall: PCA and other decomposition methods use a *linear mapping* onto a new coordinates system



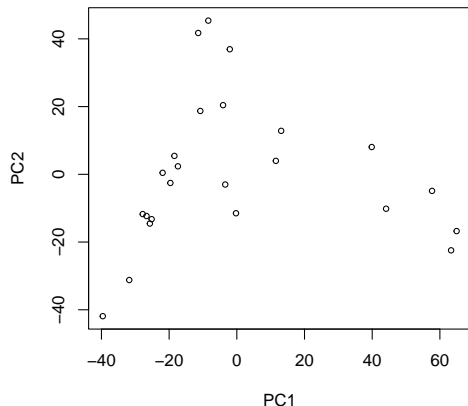
Problems with linear transformations

- Recall: PCA and other decomposition methods use a *linear mapping* onto a new coordinates system
- This doesn't always work well: especially if you have non-normally distributed (e.g. community) data



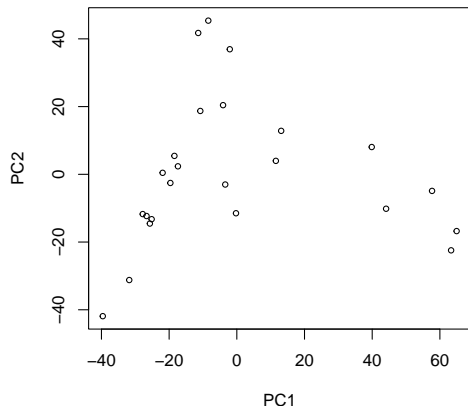
Problems with linear transformations

- Recall: PCA and other decomposition methods use a *linear mapping* onto a new coordinates system
- This doesn't always work well: especially if you have non-normally distributed (e.g. community) data
- Individual species are often normally distributed along a gradient, creating an **arch** in PCA space (see [here](#) for more details)

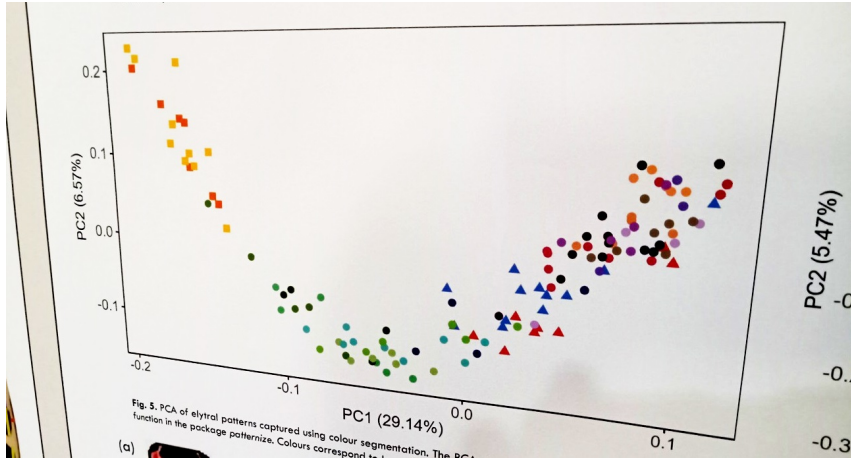


Problems with linear transformations

- Recall: PCA and other decomposition methods use a *linear mapping* onto a new coordinates system
- This doesn't always work well: especially if you have non-normally distributed (e.g. community) data
- Individual species are often normally distributed along a gradient, creating an **arch** in PCA space (see [here](#) for more details)
- Because of this, the y-axis (2nd PCA) isn't really a useful gradient to compare across

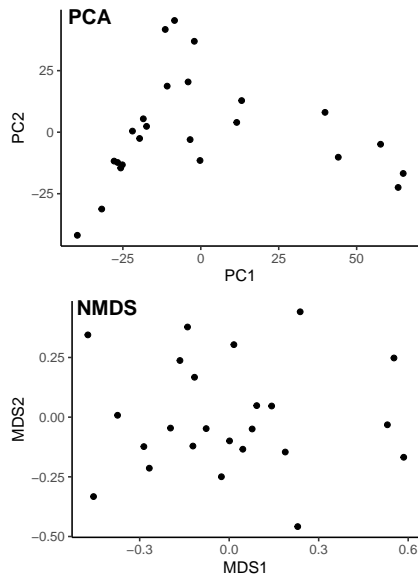


The arch effect (seen at ESC 2023 poster session!)



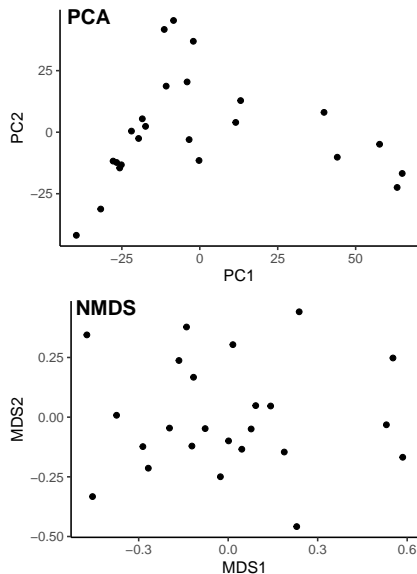
Non-metric dimensional scaling

- **Non-metric multidimensional scaling** (NMDS) is another way of decomposing multidimensional data



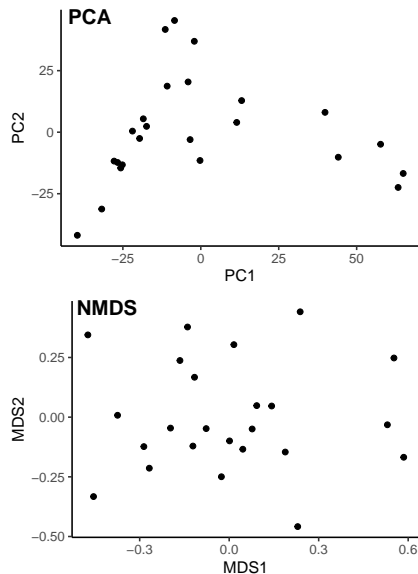
Non-metric dimensional scaling

- **Non-metric multidimensional scaling** (NMDS) is another way of decomposing multidimensional data
- Usually uses *Bray-Curtis* distance (better for community studies): order matters, but not magnitude

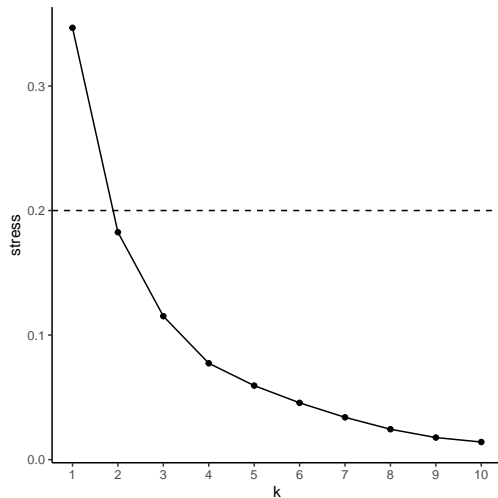
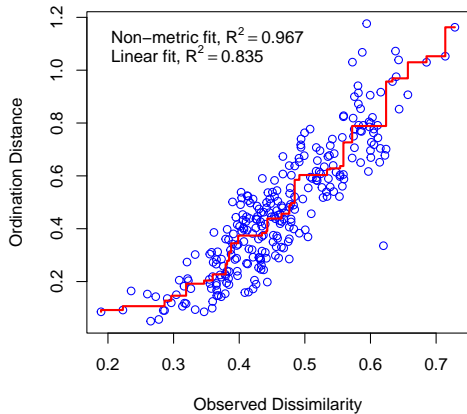


Non-metric dimensional scaling

- **Non-metric multidimensional scaling** (NMDS) is another way of decomposing multidimensional data
- Usually uses *Bray-Curtis* distance (better for community studies): order matters, but not magnitude
- Transforms data into k dimensions (usually 2), compares to original, and “wiggles” the configuration around to minimize stress (usually < 0.2) and retain rank order



Nonlinear “stress” mapping

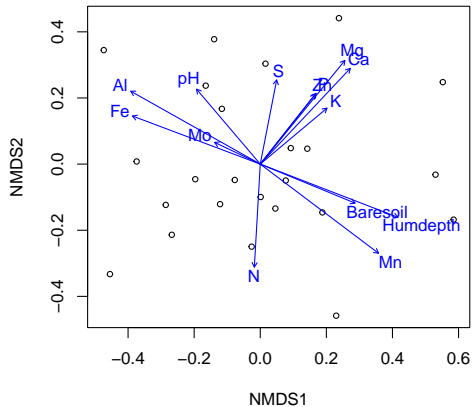


Testing and plotting differences with NMDS

- Testing differences is similar: we use `envfit` from `vegan` to do permutations tests between groups or continuous factors

```
(vareFit <- envfit(vareMDS ~ N + P + K + Ca + Mg,  
  data=varechem))
```

```
##  
## ***VECTORS  
##  
##      NMDS1      NMDS2      r2 Pr(>r)  
## N  -0.05039 -0.99873 0.2081 0.080 .  
## P   0.68703  0.72663 0.1755 0.142  
## K   0.82730  0.56176 0.1657 0.172  
## Ca  0.75015  0.66126 0.2811 0.034 *  
## Mg  0.69676  0.71730 0.3494 0.019 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
## Permutation: free  
## Number of permutations: 999
```

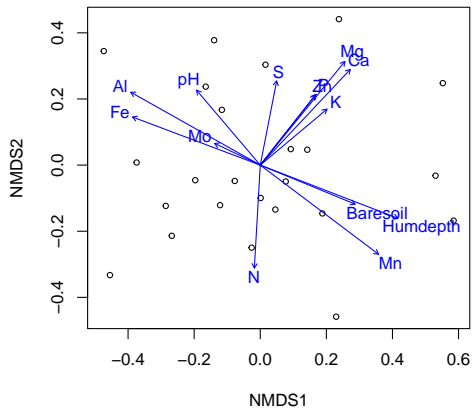


Testing and plotting differences with NMDS

- Testing differences is similar: we use `envfit` from `vegan` to do permutations tests between groups or continuous factors
- Plotting is similar, but we can't assign "% variance" to different NMDS columns

```
(vareFit <- envfit(vareMDS ~ N + P + K + Ca + Mg,  
  data=varechem))
```

```
##  
## ***VECTORS  
##  
##      NMDS1      NMDS2      r2 Pr(>r)  
## N  -0.05039 -0.99873 0.2081 0.080 .  
## P   0.68703  0.72663 0.1755 0.142  
## K   0.82730  0.56176 0.1657 0.172  
## Ca  0.75015  0.66126 0.2811 0.034 *  
## Mg  0.69676  0.71730 0.3494 0.019 *  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
## Permutation: free  
## Number of permutations: 999
```



Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!

Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!
- Half of the room can use *nutrientWater*, and the other half will use *fieldWater*. Both have Treatment and Site columns, and one has a Depth column

Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!
- Half of the room can use *nutrientWater*, and the other half will use *fieldWater*. Both have Treatment and Site columns, and one has a Depth column
- Reduce the dimensionality of these data (either PCA or NMDS), check how many dimensions are needed to represent the data, and make some plots

Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!
- Half of the room can use *nutrientWater*, and the other half will use *fieldWater*. Both have Treatment and Site columns, and one has a Depth column
- Reduce the dimensionality of these data (either PCA or NMDS), check how many dimensions are needed to represent the data, and make some plots
- What causes most of the variation in the data: Treatment or Site (or Depth)?

Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!
- Half of the room can use *nutrientWater*, and the other half will use *fieldWater*. Both have Treatment and Site columns, and one has a Depth column
- Reduce the dimensionality of these data (either PCA or NMDS), check how many dimensions are needed to represent the data, and make some plots
- What causes most of the variation in the data: Treatment or Site (or Depth)?
- Bonus: group the datasets by Site and compare the two using a Mantel test. You will need to use:

Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!
- Half of the room can use *nutrientWater*, and the other half will use *fieldWater*. Both have `Treatment` and `Site` columns, and one has a `Depth` column
- Reduce the dimensionality of these data (either PCA or NMDS), check how many dimensions are needed to represent the data, and make some plots
- What causes most of the variation in the data: `Treatment` or `Site` (or `Depth`)?
- Bonus: group the datasets by `Site` and compare the two using a Mantel test. You will need to use:
 - `dplyr` to group the datasets (more than one way)

Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!
- Half of the room can use *nutrientWater*, and the other half will use *fieldWater*. Both have Treatment and Site columns, and one has a Depth column
- Reduce the dimensionality of these data (either PCA or NMDS), check how many dimensions are needed to represent the data, and make some plots
- What causes most of the variation in the data: Treatment or Site (or Depth)?
- Bonus: group the datasets by Site and compare the two using a Mantel test. You will need to use:
 - dplyr to group the datasets (more than one way)
 - Create dissimilarity matrices (use `vegdist(x,method='euclidean')`)

Third challenge

- One of our comrades here has collected some water quality data (found [here](#)), and we're going to try analyzing it!
- Half of the room can use *nutrientWater*, and the other half will use *fieldWater*. Both have Treatment and Site columns, and one has a Depth column
- Reduce the dimensionality of these data (either PCA or NMDS), check how many dimensions are needed to represent the data, and make some plots
- What causes most of the variation in the data: Treatment or Site (or Depth)?
- Bonus: group the datasets by Site and compare the two using a Mantel test. You will need to use:
 - dplyr to group the datasets (more than one way)
 - Create dissimilarity matrices (use `vegdist(x,method='euclidean')`)
 - Use `mantel` to compare the matrices. What does this tell you about the two sets of measurements?

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**
 - IMRaD structure, *some* Supplementary Materials allowed

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**
 - IMRaD structure, *some* Supplementary Materials allowed
- **Proposal**

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**
 - IMRaD structure, *some* Supplementary Materials allowed
- **Proposal**
 - What data and analysis will you perform? Use data simulation or “example” dataset

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**
 - IMRaD structure, *some* Supplementary Materials allowed
- **Proposal**
 - What data and analysis will you perform? Use data simulation or “example” dataset
- Peer-review among class, following

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**
 - IMRaD structure, *some* Supplementary Materials allowed
- **Proposal**
 - What data and analysis will you perform? Use data simulation or “example” dataset
- Peer-review among class, following
- Conference-style final presentation to class (and committee, if desired)

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**
 - IMRaD structure, *some* Supplementary Materials allowed
- **Proposal**
 - What data and analysis will you perform? Use data simulation or “example” dataset
- Peer-review among class, following
- Conference-style final presentation to class (and committee, if desired)

To do: start planning your paper/proposal

Term project: paper or proposal draft

- **Paper**
 - IMRaD structure, *some* Supplementary Materials allowed
- **Proposal**
 - What data and analysis will you perform? Use data simulation or “example” dataset
- Peer-review among class, following
- Conference-style final presentation to class (and committee, if desired)

Rubric to follow next week