Multivariate models

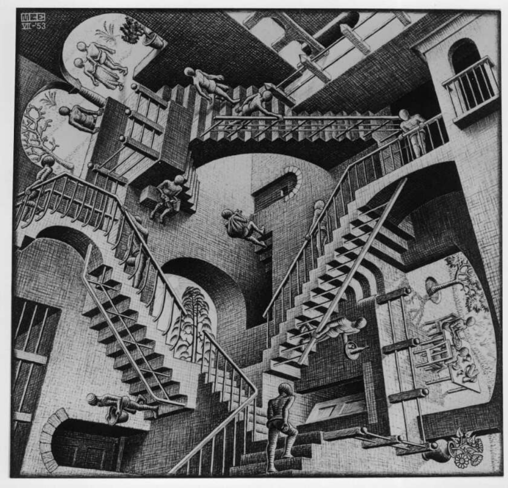
More than one way of seeing things

Samuel Robinson, Ph.D.

Oct 20, 2023

Outline

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



Some common problems

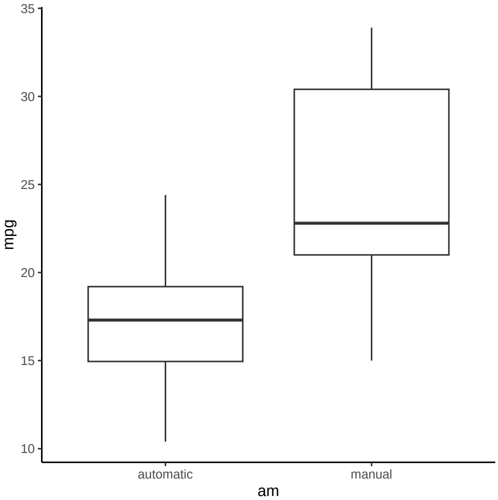
- "I've got a zillion predictors that could matter in my model, but they're all collinear"
- "I measured a zillion things for each site/critter, but I don't want to fit a zillion models"
- "I measured a zillion things. Do certain things group up into clusters?"
- "My supervisor told me to do a PCA or 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?

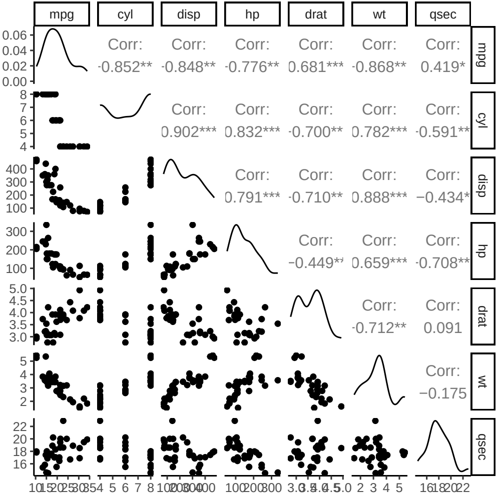
Univariate data

- Up until now, we've dealt mainly with **univariate** data: one thing is changing, and is being affected by other things
- 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
- *Many things* are changing, with multiple things potentially causing other things
- These are *mostly* normal (non-normal can be tricky)



Multivariate normal

- Normal distributions[1] don't just have a single σ, but actually a *matrix* of values
- If the columns of our data are *independent*, then it looks like this:

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

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

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

- Zeros mean "μ_1, μ_2, & μ_3 aren't related to each other"
- Diagonal elements = *variance*, off-diagonal = *covariance*

[1] Multivariate Normal

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)

$$\Sigma = \begin{bmatrix} \sigma_a & \sigma_a b & \sigma_a c \\ \sigma_a b & \sigma_b & \sigma_b c \\ \sigma_a c & \sigma_b c & \sigma_c \end{bmatrix} = \begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Covariance vs Correlation

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

Covariance matrix

$$\begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

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...
- You've gathered data from a *multivariate normal distribution*!
- Now your task is to model this distribution!

$$Y \sim Normal(M, \Sigma)$$

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

$$\Sigma = \begin{bmatrix} \sigma_a & \sigma_a b & \sigma_a c \\ \sigma_a b & \sigma_b & \sigma_b c \\ \sigma_a c & \sigma_b c & \sigma_c \end{bmatrix}$$

Problem: this doesn't really help

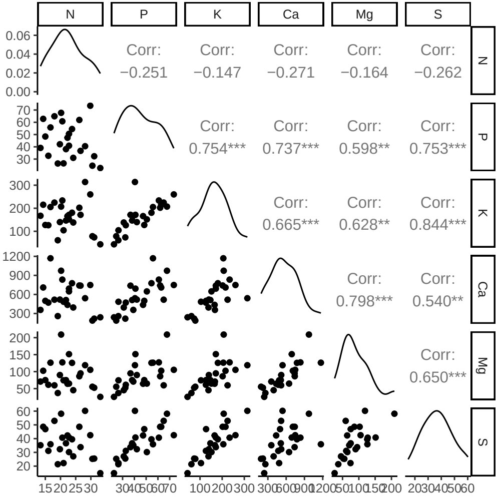
- We're *still* stuck with fitting a zillion models
- Also have estimate covariance - even worse!
- We need a better way for dealing with these data

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

$$\Sigma = \begin{bmatrix} \sigma_a & \sigma_a b & \sigma_a c \\ \sigma_a b & \sigma_b & \sigma_b c \\ \sigma_a c & \sigma_b c & \sigma_c \end{bmatrix}$$

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? Let's look at the data

Back to covariance and correlation

- 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 the matrix into its component eigenvectors and eigenvalues
- Linear transformation of the data into new coordinate space, where *most of the variation falls into a few columns*. These are its **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:

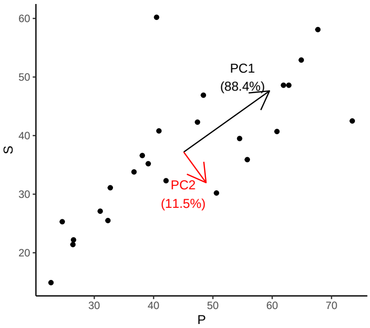
```
## Standard deviations (1, .., p=6):
## [1] 250.059202  48.742293  23.933838   9.146864   5.605743   3.92
##
## Rotation (n x k) = (6 x 6):
##             PC1          PC2         PC3          PC4          PC
## N   0.005932593 -0.006016911  0.01803481 -0.160065974  0.82647967
## P  -0.044874007 -0.102474967 -0.06146157  0.882810055  0.35474786
## K  -0.179882646 -0.954949326 -0.14631853 -0.164441290  0.01978546
## Ca -0.973256734  0.200866159 -0.10593744 -0.021515421 -0.00991773
## Mg -0.132896410 -0.112926024  0.97719626 -0.006750228  0.04951397
## S  -0.026518575 -0.156316865  0.09139386  0.409238162 -0.43375551
```

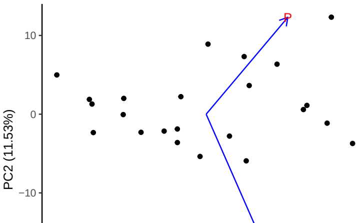
A simpler example: 2 dimensions

Principal components are hard to imagine, so let's break it down into 2 dimensions:

```
prcomp(vc_2)
```

```
## Standard deviations (1, .., p=2):
## [1] 17.835248  6.437148
##
## Rotation (n x k) = (2 x 2):
##          PC1        PC2
## P 0.8109851  0.5850668
## S 0.5850668 -0.8109851
```

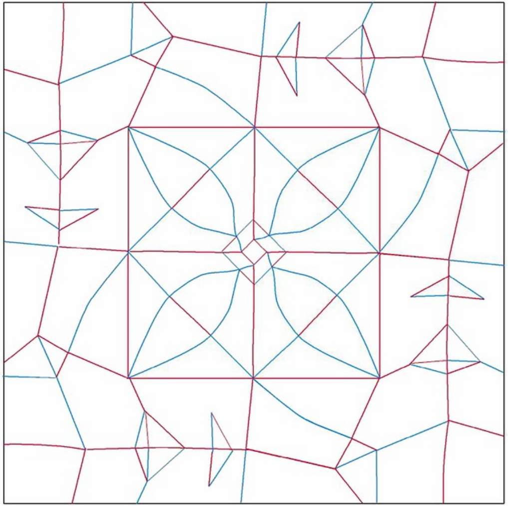




Artistic approaches to this problem

Picasso's *Demoiselle d'Avignon*

Kawasaki rose crease pattern



Example: full dataset

2-column