Statistics for Linguists

Adam J.R. Tallman

2021 06 15 / 2021 06 21

Friedrich Schiller Universität Jena

From last lecture

- Logistic regression
- Multilevel models

This lecture

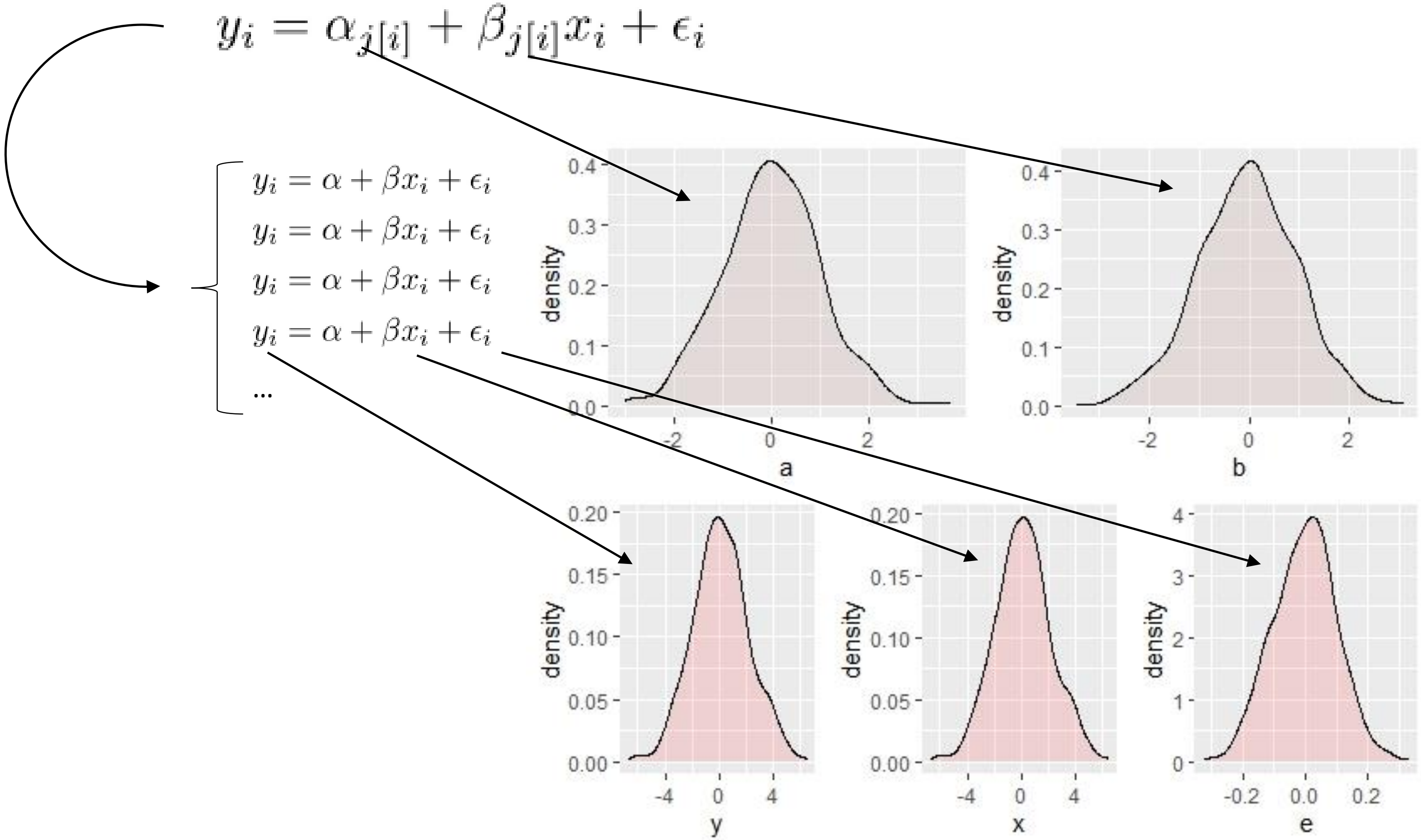
- Multilevel models
- Phoneme inventory vs. Population size
- Exploratory vs. Confirmatory analysis
- Clustering
- Hierarchical clustering
- K-mean clusters

Multilevel models

Multilevel regression and random variables

- In such cases you have a *random variable* for the effects of your model.

- That’s why multilevel models are sometimes called ‘random effects models’



$$y_i = \alpha_{j[i]} + \beta_{j[i]} x_i + \epsilon_i$$

$$y_i = \alpha + \beta x_i + \epsilon_i$$

$$y_i = \alpha + \beta x_i + \epsilon_i$$

$$y_i = \alpha + \beta x_i + \epsilon_i$$

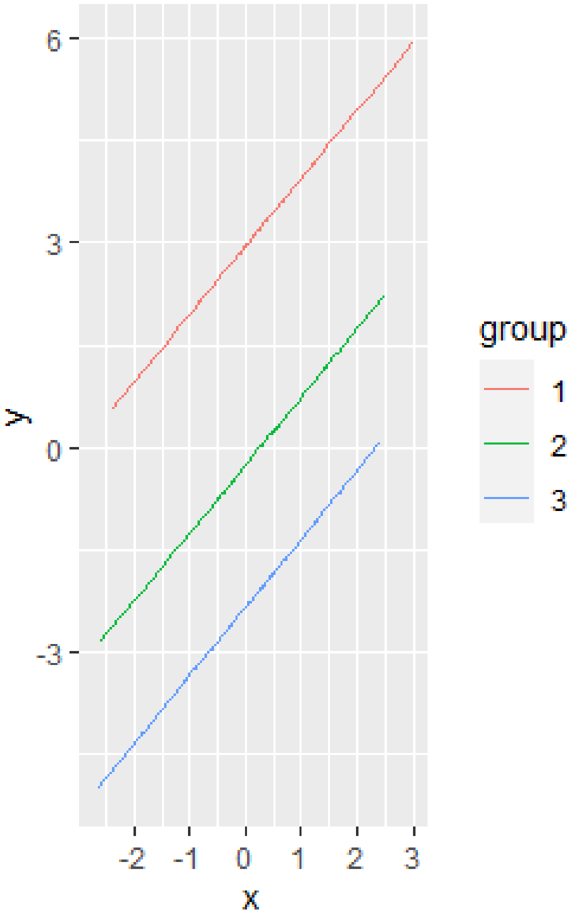
$$y_i = \alpha + \beta x_i + \epsilon_i$$

…

Multilevel regression and random variables

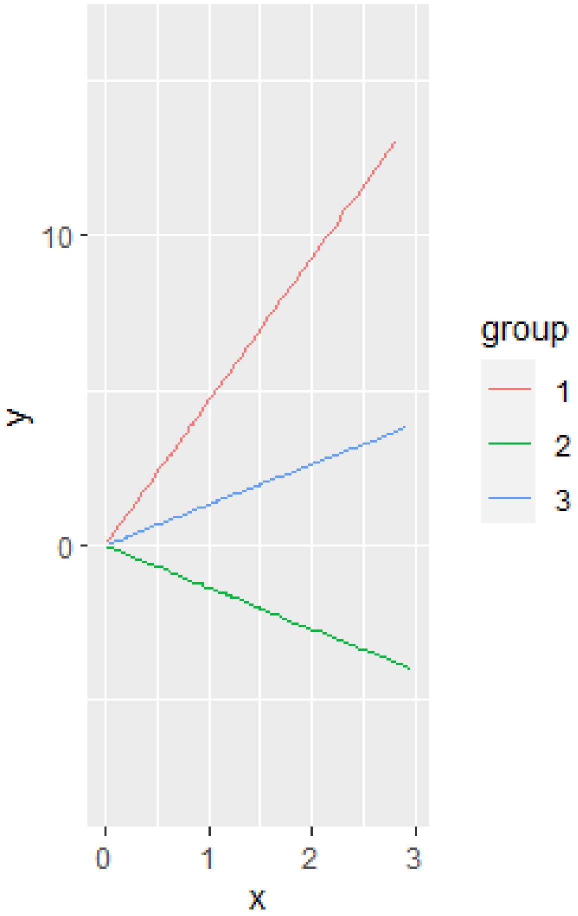
$y_i = \alpha_{j[i]} + \beta x_i + \epsilon_i$

Varying intercept



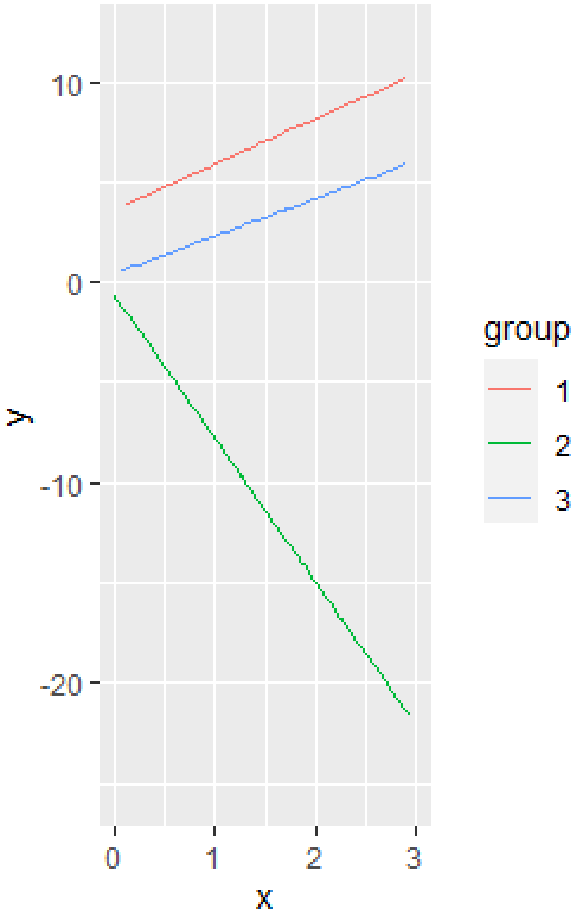
$y_i = \alpha + \beta_{j[i]} x_i + \epsilon_i$

Varying slope



$y_i = \alpha_{j[i]} + \beta_{j[i]} x_i + \epsilon_i$

Varying intercept and slope



Why?

- Why would we ever do this?

- Why no just have lots of separate models for each group?

Schools

- Imagine trying to assess the effectiveness of some new education curriculum of teaching style.

- You have a treatment group and a control group and then you assess the students’ results.

Schools

- But you know that there will be variation between schools.
- Some schools won’t be able to effectively administer the training/treatment because they have less resources.
- Furthermore, you have variation between schools with respect to how many students participated.
- Schools vary in terms of their culture, socioeconomic conditions teachers, size, quality and style of education.
- Yet, the students are all from the same population.

Schools

- You have a measurement for aptitude *y* and you have a treatment variable (trained or not trained) *x*.

- What do you do with the schools variable?

- Complete pool
  - Run a regression ignoring the variation between schools

- No pooling
  - Run a regression for each school
  - Run a regression with school as a factor

Complete pooling

- Complete pooling has the obvious danger of ignoring the variation between schools.

- If one school has more data points it could be an outlier with respect to the effects, but overwhelm the data across cases.

- The results might be biased towards with more data points.

No pooling

- No pooling could tend to exaggerate the variation between schools.

- For schools that do not have very many data points, there is a higher likelihood of variation simply appearing by chance.
  - Think of the law of large numbers

Partial pooling

- Multilevel modelling basically compromises between complete pooling and no pooling.
- ‘Multilevel modeling partially pools the group-level parameters α_j toward their mean μ_α. There is more pooling when the group-level standard deviation σ_α is small, and more smoothing for groups with fewer observations.’
    - Gelman & Hill (2009: 258)

$$\text{estimate of } \alpha_j \approx \frac{\frac{n_j}{\sigma_y^2}}{\frac{n_j}{\sigma_y^2}+\frac{1}{\sigma_\alpha^2}}(\overline{y}-\beta\overline{x}_j)+\frac{\frac{1}{\sigma_\alpha^2}}{\frac{n_j}{\sigma_y^2}+\frac{1}{\sigma_\alpha^2}}\mu_\alpha$$

Partial pooling

- Partial pooling results in *shrinkage* of variance in the coefficients of each group towards the overall mean as a function of their in-group sample size (*n_j*)

$$\text{estimate of } \alpha_j \approx \frac{\frac{\boxed{n_j}}{\sigma_y^2}}{\frac{n_j}{\sigma_y^2}+\frac{1}{\sigma_\alpha^2}}(\overline{y}-\beta\overline{x}_j)+\boxed{\frac{\frac{1}{\sigma_\alpha^2}}{\frac{n_j}{\sigma_y^2}+\frac{1}{\sigma_\alpha^2}}\mu_\alpha}$$

- Shrinkage is less as your in-group sample size is larger.
- Partial pooling towards the mean.

Multilevel models

- Classical regression models can be viewed as a special case of multilevel models

- As $\sigma_\alpha \rightarrow 0$ the model is more like a complete pooling model.

- As $\sigma_\alpha \rightarrow \infty$ the model is more like a no-pooling model.

Fixed or random effects?

- The literature on statistics makes a distinction between fixed and random effects.

- In my statistics training I was taught two conflicting definitions.

  - Fixed -> the coefficient is a single constant; Random -> the coefficient comes from a random variable.
  - Fixed -> Sample exhausts the population of effects; Random -> any other type of group.

Fixed or random effects?

- Gelman (2005) ‘Analysis of Variance – why it is more important than ever’ *The Annals of Statistics* summarizes a number of not necessarily consistent definitions.

Gelman (2005) definitions of fixed and random effects

- 1. Fixed effects are constant across individuals, and random effects vary . For example in a growth study a model with random intercepts α_i and fixed slope β corresponds to parallel lines for different individuals *i*, or the model $y_{it} = \alpha_i + \beta t$.
- 2. Effects are fixed if they are interesting in themselves or random is there is interest in the underlying population.
- 3. “When a sample exhausts the population the corresponding variable is *fixed*; when the sample is a small (i.e. negligible) part of the population the corresponding variable is *random” [Green and Tukey (1960)* quoted in Gelman (2005: 20)]

Gelman (2005) definitions of fixed and random effects

- 4. “If an effect is assumed to be a realized value of a random variable, it is called a random effect [LaMotte (1983) cited in Gelman (2005)]
- 5. Fixed effects are estimated using least squares (or more generally maximum likelihood) and random effects are estimated with shrinkage [“linear unbiased prediction”…]

Constant vs. varying effects

- Gelman (2005) suggests using the term *constant* if the effect is identical for al groups in the population and *varying* if they are allowed to differ from group to group.

Population size vs.phoneme inventory size

Population size vs.phoneme inventory size

- A number of researchers had proposed a relationship between population size and phoneme inventory size.
- Isolation+monolingualism -> small phoneme inventory
    - Haudricort
- A few studies found a correlation between population size and small phoneme inventory size.
- Assuming population is a proxy for isolation and monolingualism [!], perhaps Haudricort's hypothesis has some validity.

Previous studies

- Moran et al. 2012 mention three types of methodologies.
    - 1. Case studies
    - 2. Computer simulations
    - 3. Statistical modelling

Previous studies

- Moran et al. 2012 mention three types of methodologies.
  - 1. Case studies -> Confirmation bias
  - 2. Computer simulations -> Not realistic? No empirical support
  - 3. Statistical modelling -> Different sampling or different measurement technique gets different results

Sampling problem

- How do we choose languages so that the sampling is *not* biased towards particular families?

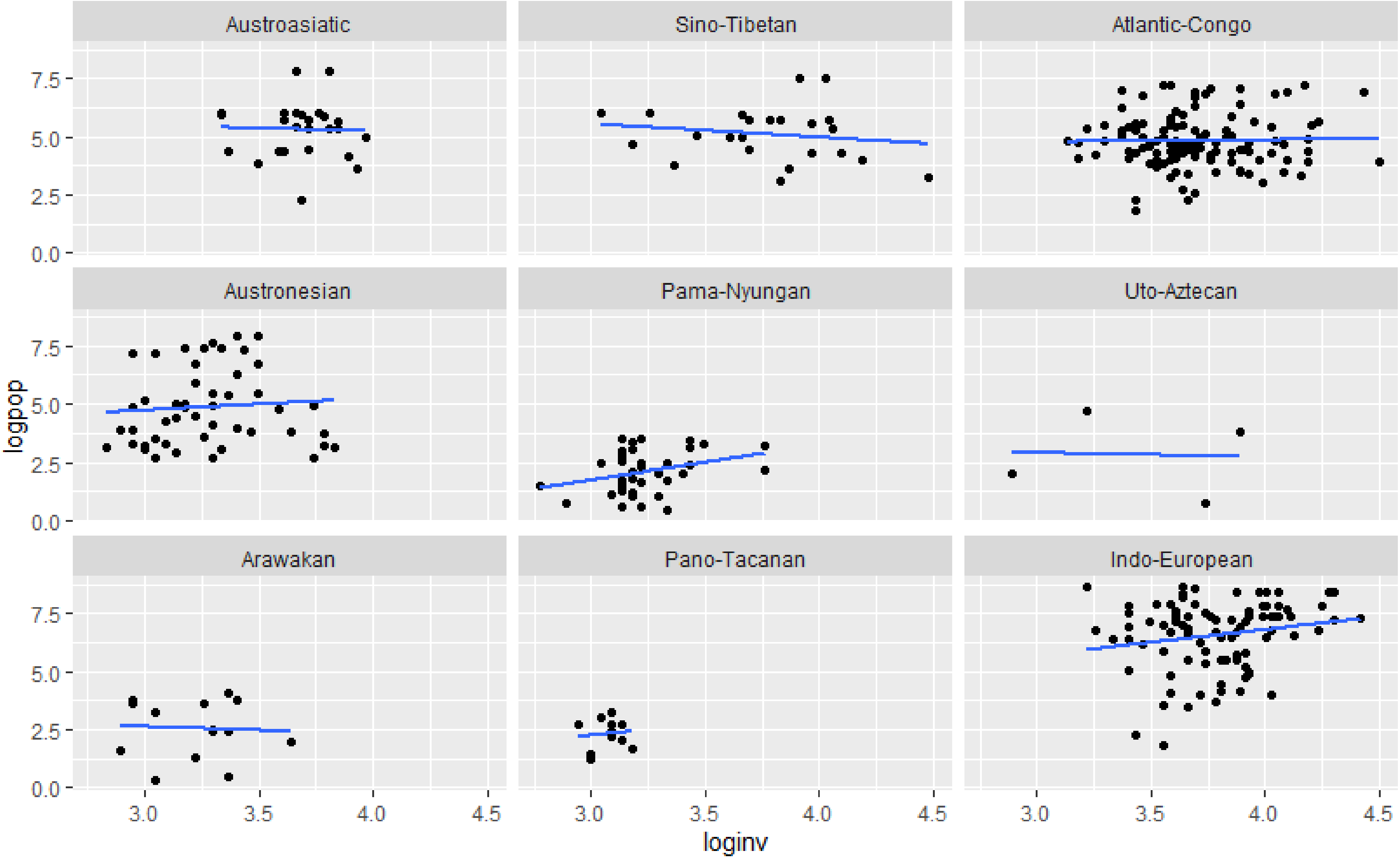
- **Bibliographic bias**: Data tend to include families that are well documented.

Other problems

- Spearman’s rho is overly permissive
  - (since its any monotonic relationship)

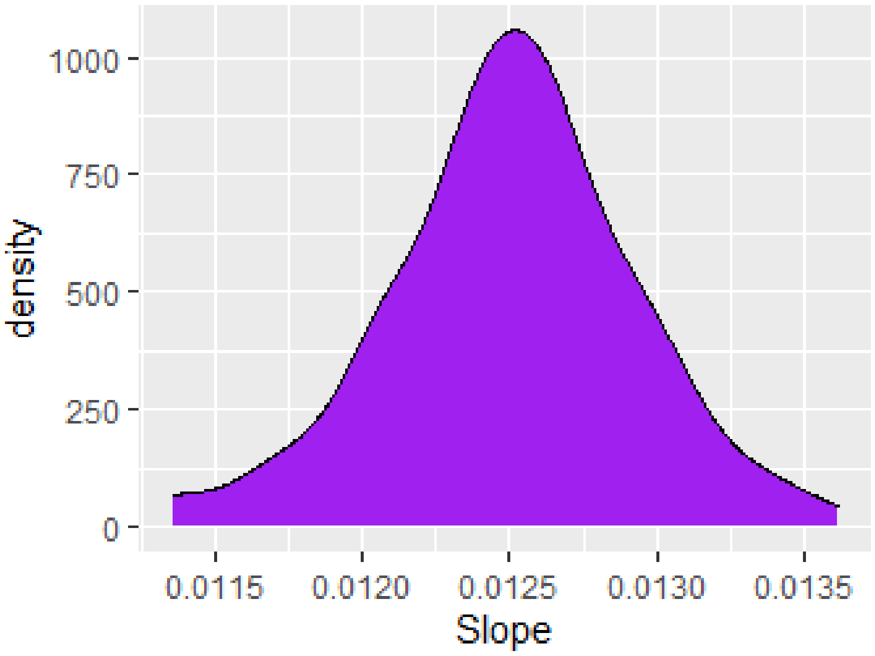
- Statistical significance may not be informative with high sample sizes

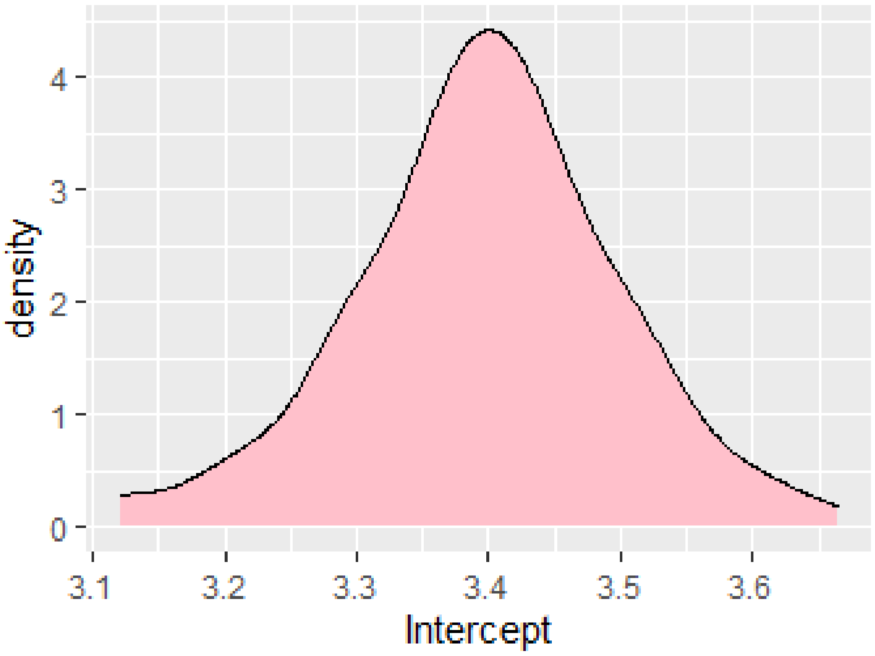
- We should be interested in effect size


Austroasiatic
Sino-Tibetan
Atlantic-Congo
Austronesian
Pama-Nyungan
Uto-Aztecan
Arawakan
Pano-Tacanan
Indo-European
logpop
0.0
2.5
5.0
7.5
loginv
3.0
3.5
4.0
4.5

Practice

- Let’s see if we can reproduce the results…





Exploratory data analysis

Exploratory data analysis

- The basic ideas of exploratory data analysis were pioneered and articulated by John Tukey.
- Developed Fast Fourier transforms, Tukey’s test, Tukey’s lemma etc.
- Developed methods for visualizing data in order to probe assumptions of statistical models or to look for unexpected patterns.
- Invented boxplots and a number of other visual techniques we now use today
- Also invented the term ‘bit’

John Tukey 1915 - 2000

"The most important maxim for data analysts to heed, and one many statisticians seem to have shunned is this: "Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made more precise." Data analysts must progress by approximate answers at best, since its knowledge of what the problem really is will at best be approximate" Tukey (1962:62)

John Tukey 1915 - 2000

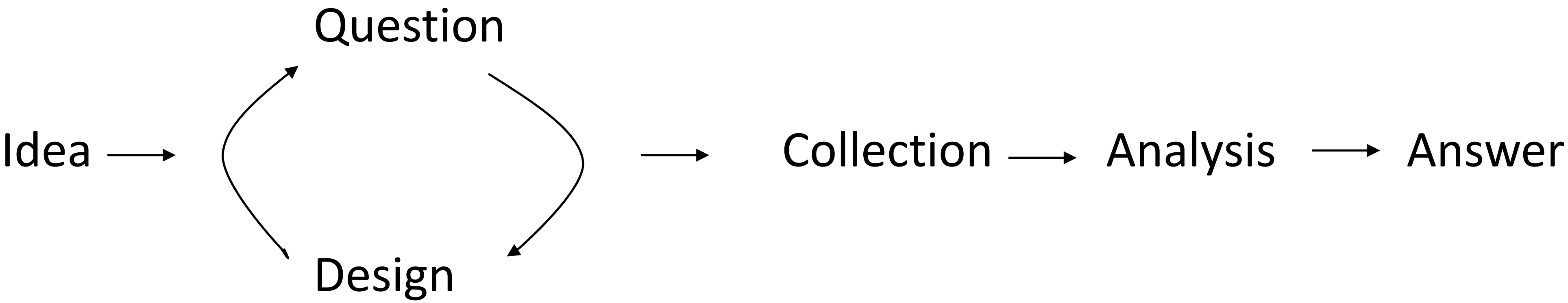
Exploratory data analysis

- Some things that EDA is for ….
  - Guarding against erroneous conclusions from data
  - Looking for nonlinearity in the data (plotting techniques, should the data be logged, etc.?)
  - Advanced graphics (boxplots)
  - Model diagnostics (checking for heteroskedasticity, etc.)
  - Exploring the data to develop new hypotheses
- EDA contrasts with CDA (confirmatory data analysis), or what I referred to as “Inferential statistics” earlier in the course.
- EDA is about iterative hypothesis generation through interaction with the data.

Exploratory data analysis

- Straight line paradigm vs. iterative approach to development of ideas.
  - Tukey (1980)
- Exploratory data analysis is basically the **detective work** that goes on behind the scenes of confirmatory data analysis

Question → Design → Collection → Analysis → Answer



Question

Idea → Collection → Analysis → Answer

Design

Algebra lies

Imagine a model as follows:

$y = a + bx + e$

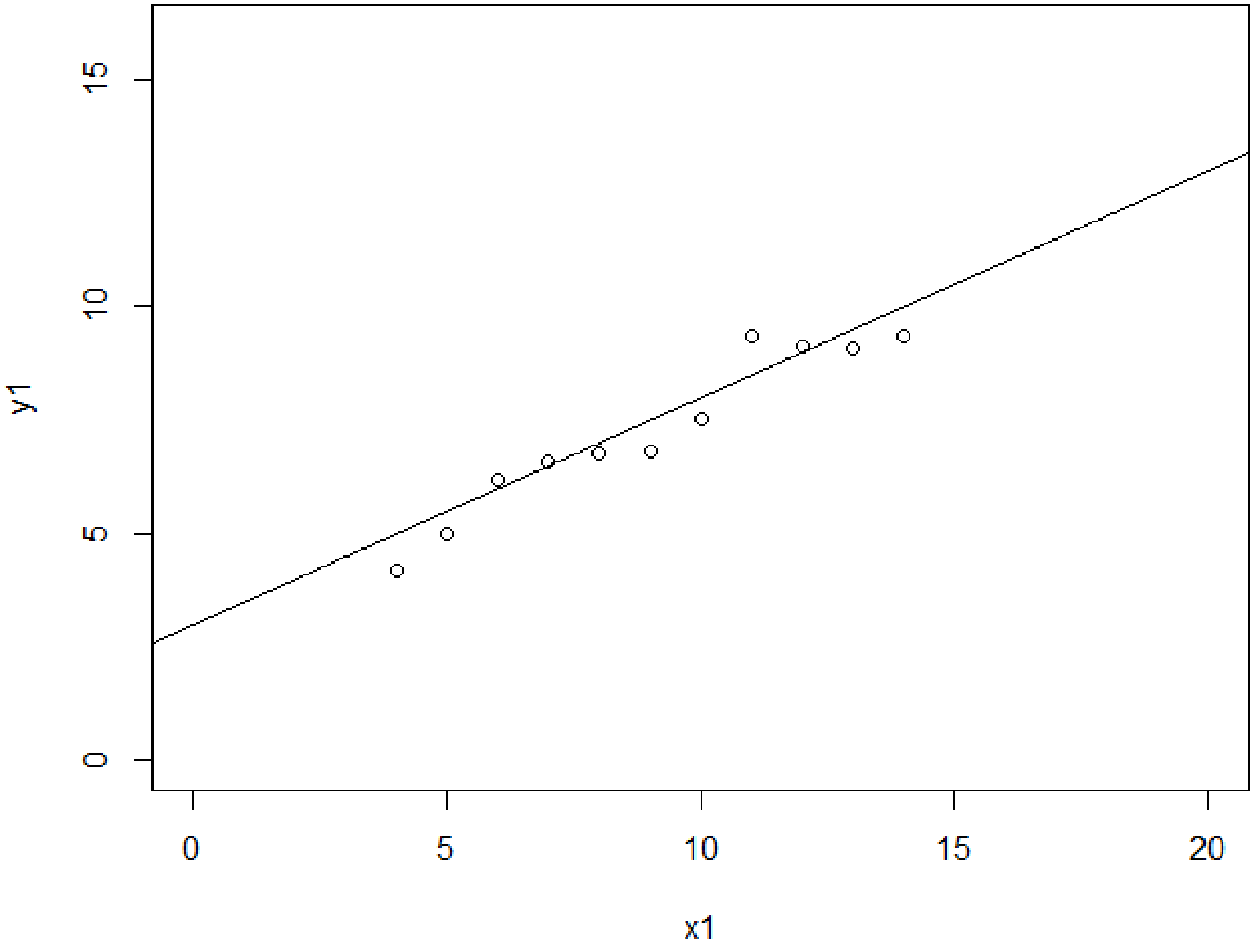
Where a is 3 and b is 0.5.

There are a variety of patterns that this could correspond to – remember you are assuming that

Algebra lies, so you need graphs

- Data analysis is about summation or aggregation over data, but different ways of aggregating the data can provide different results.
- Imagine *x* has a standard deviation of 3.3 and a mean of 9
- Imagine *y* has a standard deviation of 2 and a mean of 7.5.
- For $y = a + bx + e$, $a = 3$ and $b = 0.5$
- There are actually a lot of different patterns that can correspond to this statistical summation.

```
set.seed(3)

x1 <- seq(4,14,by=1)

y1 <- 3 + 0.5*x1+ rnorm(11,0,0.7)

plot(x1,y1, xlim=c(0,20), ylim=c(0,16))

abline(a=3, b = 0.5 )
```



Simulating very different data

```
x2 <- c(4, 5,  6, 7, 8, 9,   10, 11,  12 ,13 ,  14)
y2 <- c(3, 4.5, 6, 7.25, 8, 8.5, 9, 9.25, 9  ,8.5, 8)
lm(y2~x2)
plot(x2,y2, xlim=c(0,20), ylim=c(0,16))
abline(a=3, b = 0.5 )
x3 <- seq(4,14)
y3 <- 4.75 + 0.25*x3
y3[10] = 15
lm(y3~x3)
plot(x3,y3, xlim=c(0,20), ylim=c(0,16))
abline(a=3, b = 0.5 )
```

```
x4 <- rep(8,10)
x4[11] <- 18
y4 <- seq(5,10, by =0.5)
y4[11] <- 12.25
lm(y4~x4)
plot(x4,y4, xlim=c(0,20), ylim=c(0,16))
abline(a=3.25, b = 0.5 )
```

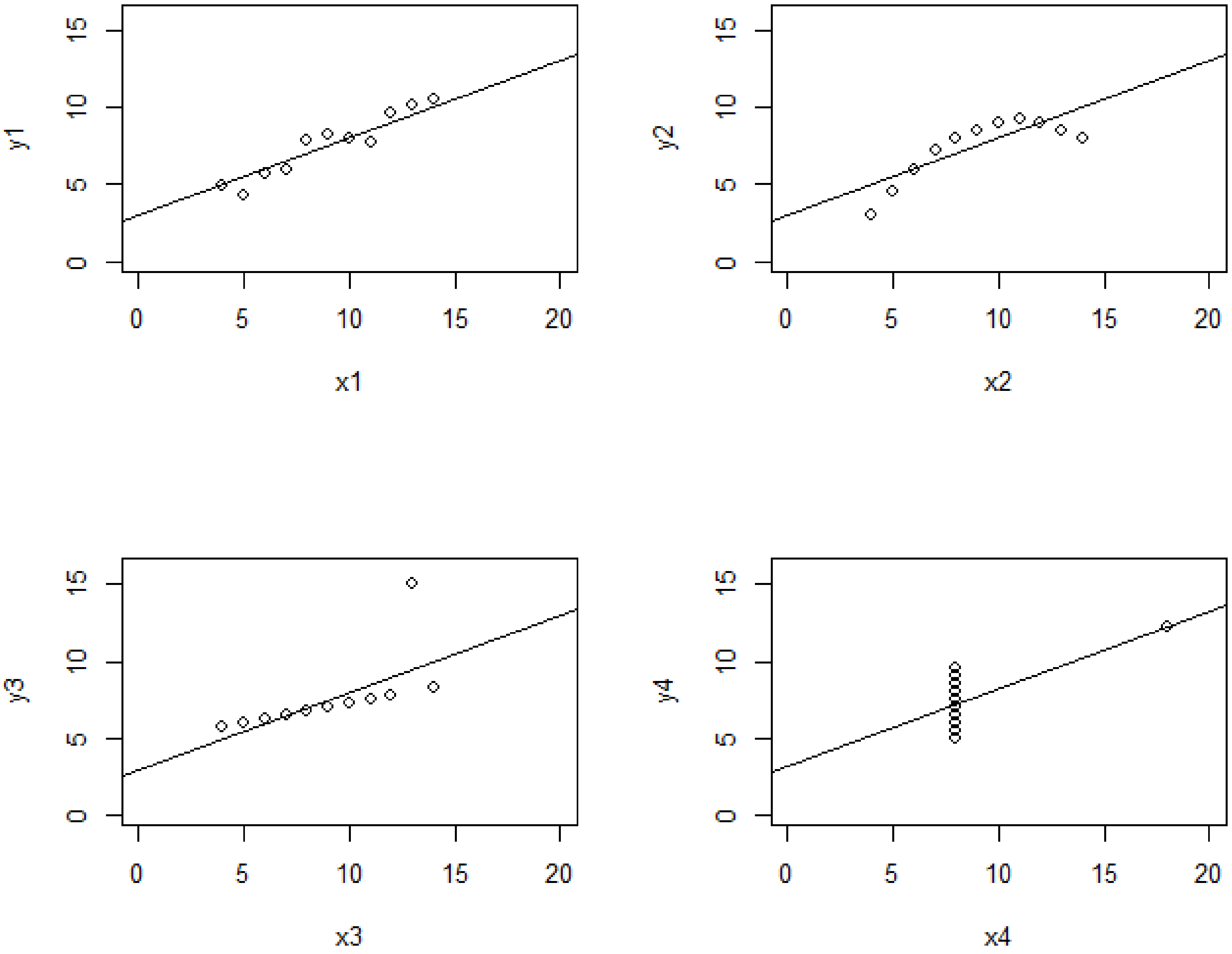
- These will all give you roughly the same result for the slope and intercept.

- But it is obvious that the model is only good for the first graph.

- Slope = 0.5

- Intercept = 3

- The point is that without a visual exploration of the data and/or any attempt to test modelling assumptions, your model will make bad predictions.

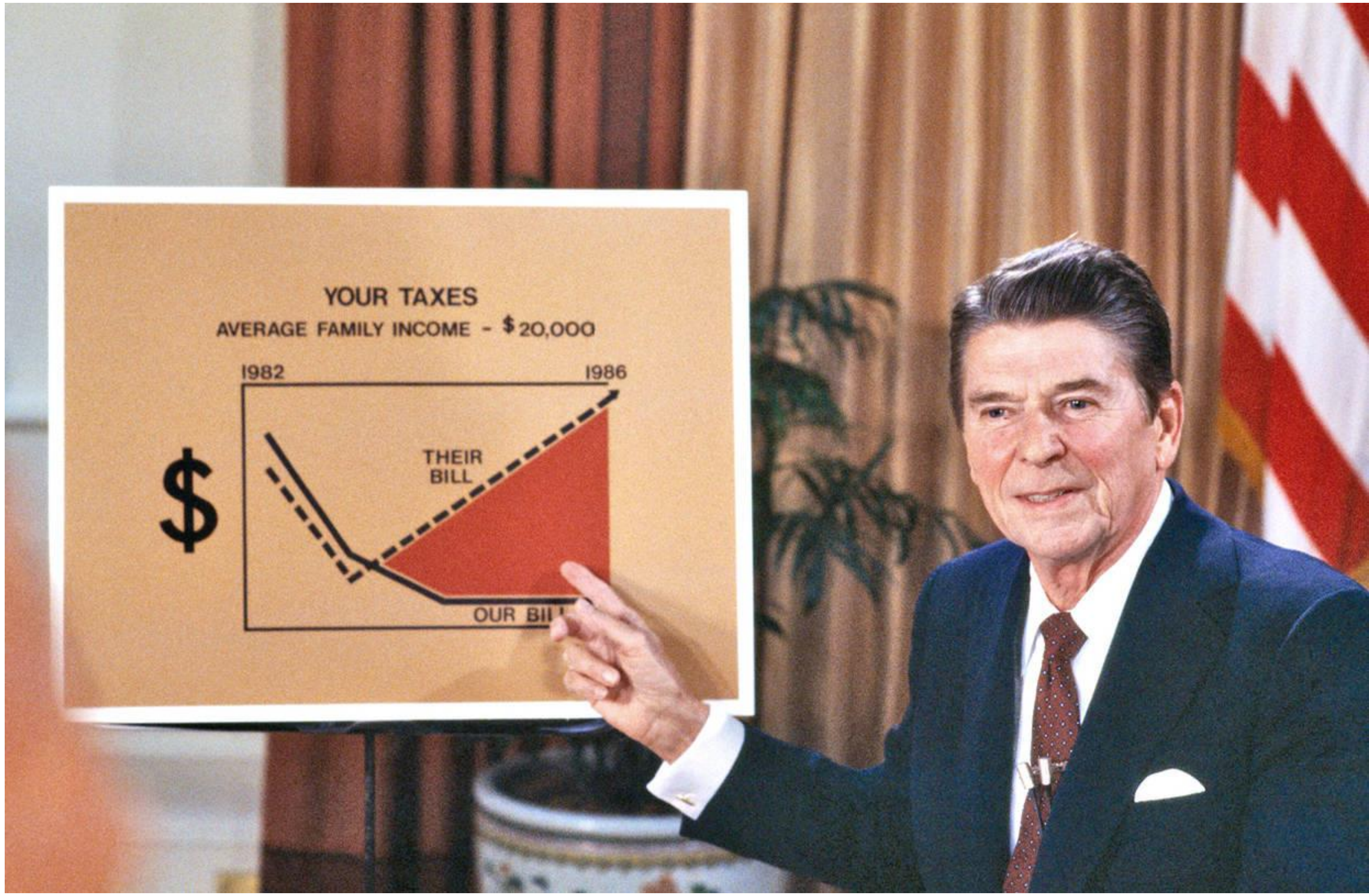


Graphs lie, so you need algebra

- It’s easy to give misleading figures – or its easy to give figures that deceive.

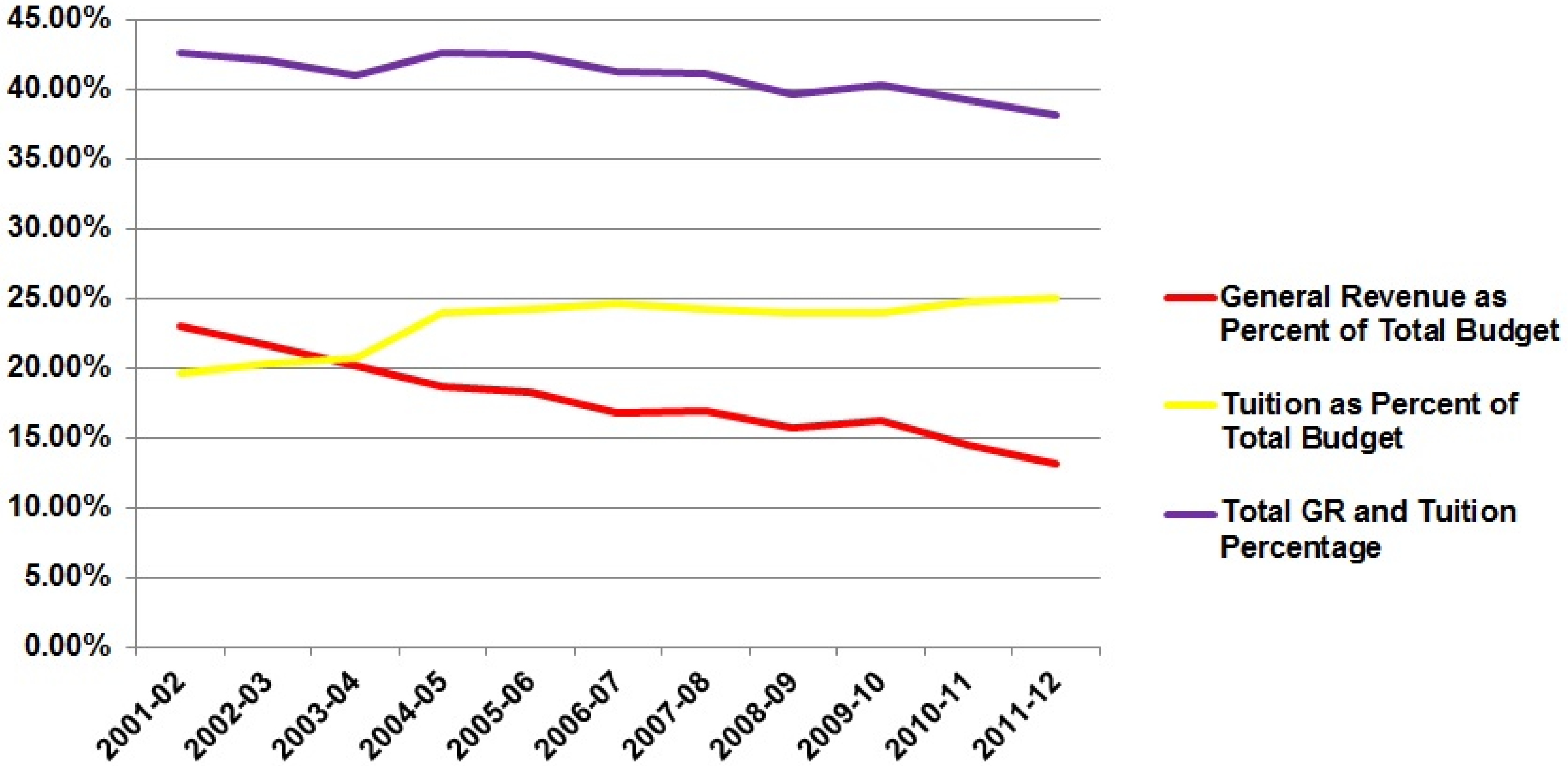
- Governments and other institutional bodies with a vested interest the public or stakeholders not knowing the truth of the situation are experts in this.

- Great example: University administrators in the US!



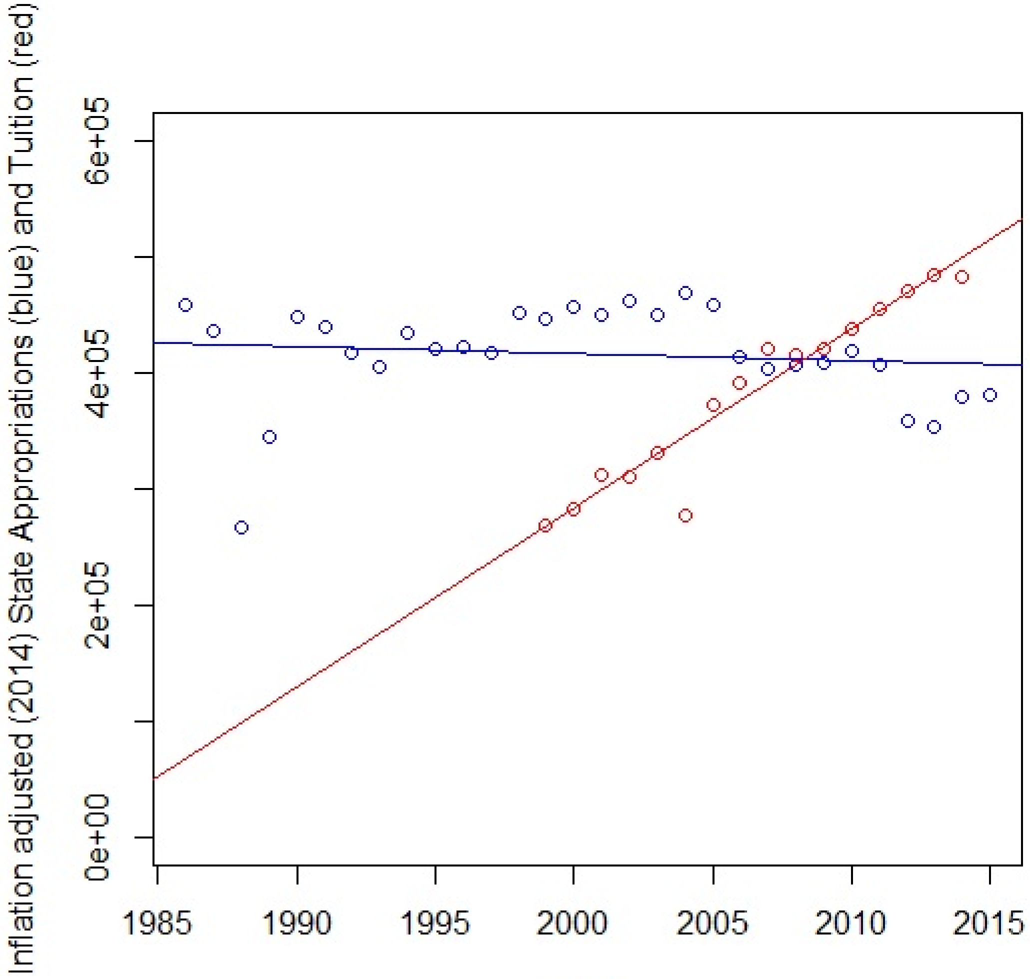
UT Austin, tuition increase arguments

- UT Austin administration uses this graph to argue in favor of tuition increases *to this day*.



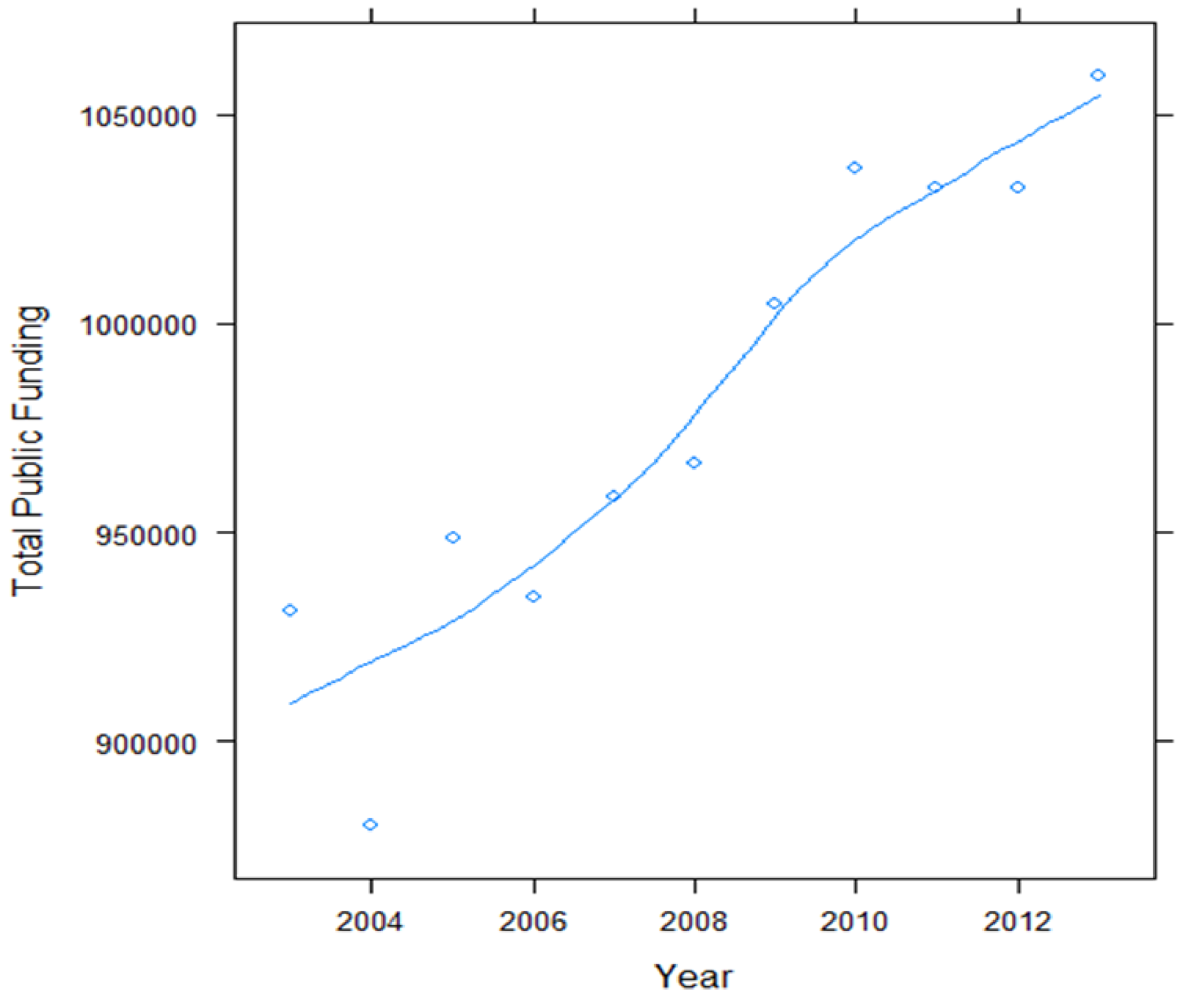
UT Austin, tuition increase arguments

- Here’s another (more honest) graph that we produced looking at the relationship between tuition revenue and state appropriations over time.



UT Austin, tuition increase arguments

- Here’s how much actual public funding the university has from 2004-2013.



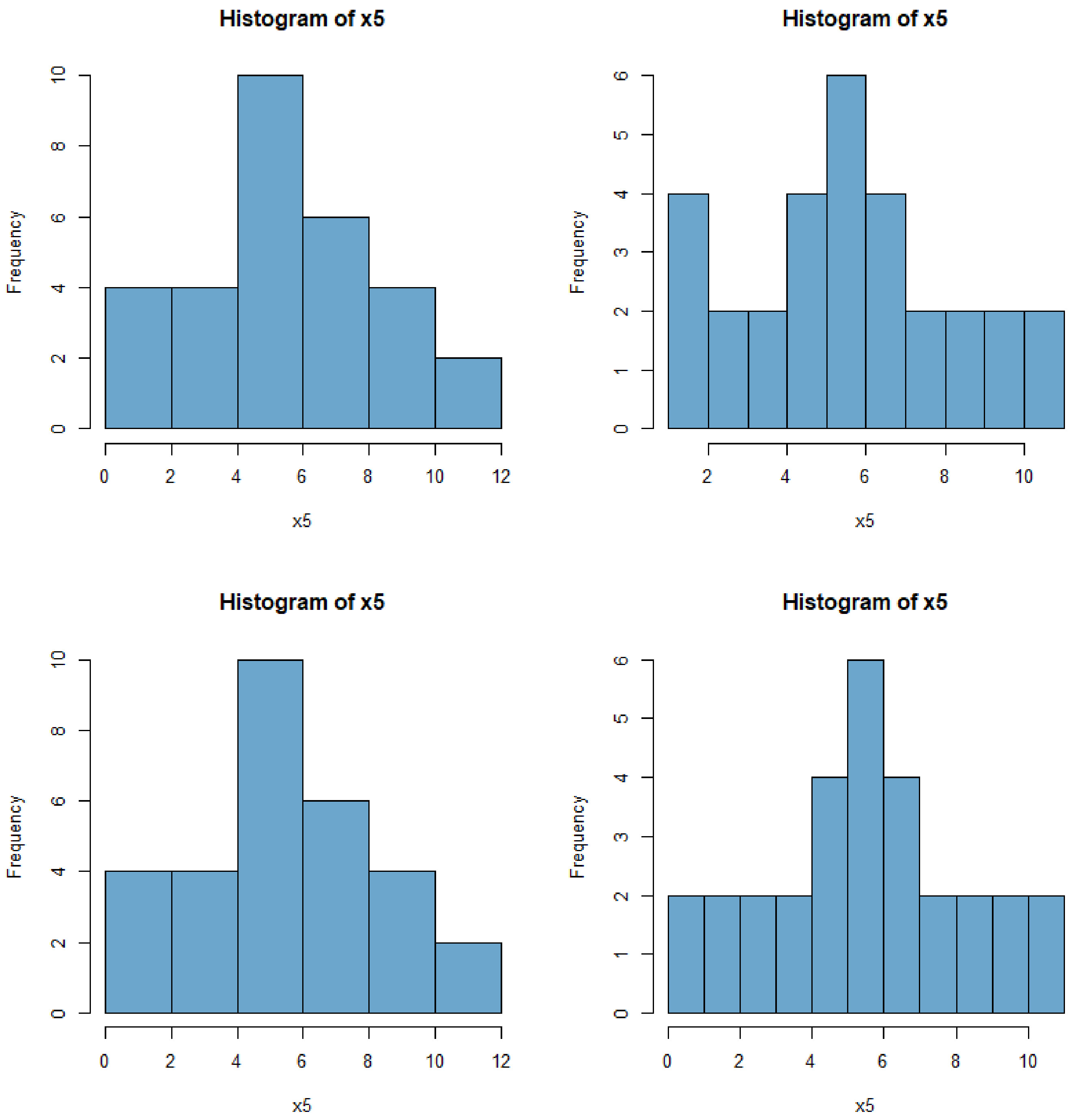
Graphs lie, so you need algebra

- It is also very easy to deceive *yourself* with graphs, because the graphing algorithm can do something to the data that you don’t expect.

```
x5 <- c(1,1,2,2,3,3,4,4,5,5,5,5,6,6,6,6,6,6,7,7,7,7,8,8,9,9,10,10,11,11)
par(mfrow=c(2,2))
hist(x5, col="skyblue3")
hist(x5, col="skyblue3", breaks=10)
hist(x5, col="skyblue3", breaks=5)
hist(x5, col="skyblue3", breaks=c(0,1,2,3,4,5,6,7,8,9,10,11))
```

- These are all from the same data, but setting the bins and bin spacing differently.

- We know that only the last one is an accurate representation of the data.



Hiding structure

- Certain types of graphs can also hide structure.

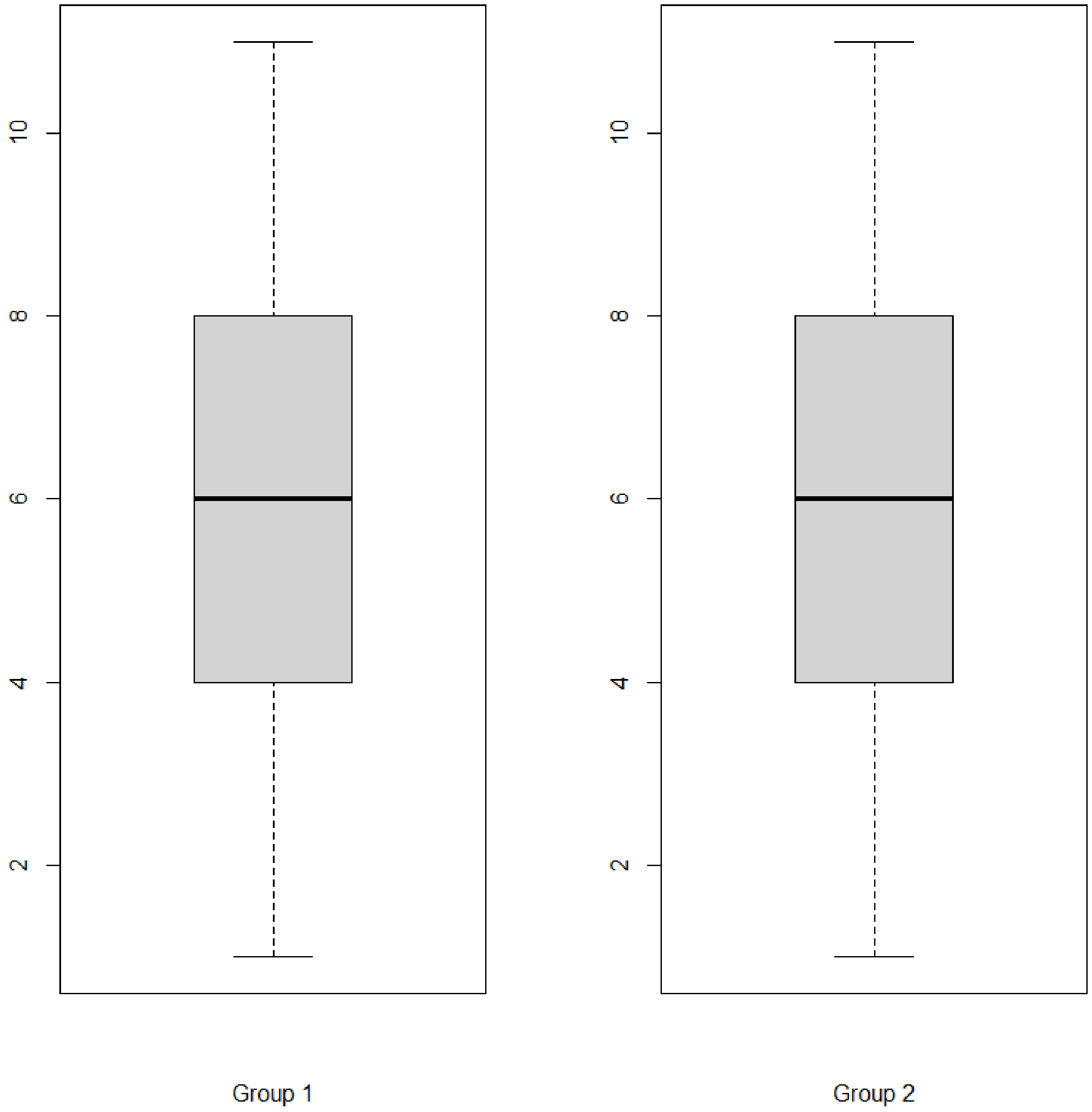
- The boxplot is a graph based on a 5 number summary of the data.
  - The meidan, the first and second quartile and the lowest and highest numbers or something else separating extreme values (e.g. 5% / 95% confidence intervals).

- All graphical techniques hide certain aspects of structure, even boxplots.

Hiding structure

```
group1 <- NULL
group1$y <- c(11,11,10,10,9,9,8,8,7,7,7,7,6,6,6,6,6,6,5,5,5,5,4,4,3,3,2,2,1,1)
group1 <- as.data.frame(group1)
group1$group <- 1
group2 <- NULL
group2$y <- c(11,11,11,11,11,11,11, 8,8,8,8,8,8,8,8,4,4,4,4,4,4,4,4,1,1,1,1,1,1,1)
group2 <- as.data.frame(group2)
group2$group <- 2
groups <- rbind(group1,group2)
```

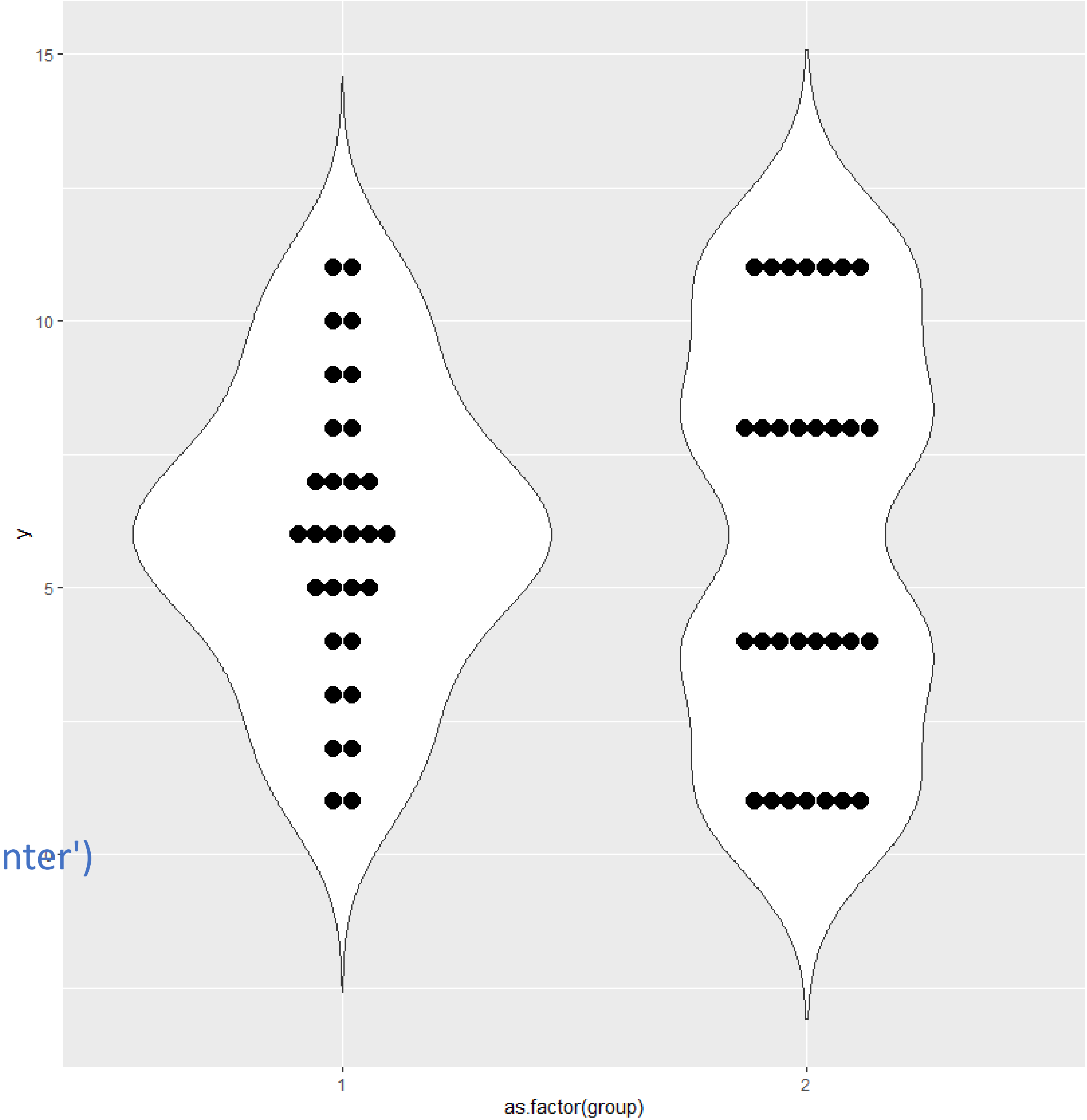
```
par(mfrow=c(1,2))
boxplot(group1$y, xlab="Group 1")
boxplot(group2$y, xlab= "Group 2")
```



Dot plot

- A dot plot in this case might be more revealing.

```
ggplot(groups, aes(x=as.factor(group), y=y))+
        geom_violin(trim=FALSE)+
        geom_dotplot(binaxis='y', stackdir='center')
```



Clustering

Classification vs. clustering

- There are two powerful statistical tools for EDA, beyond graphing.

- Supervised learning or classification is when the categories are labelled in your data.
  - E.g. random forests / decision trees

- Unsupervised learning or clustering is when the categories are unlabeled.

Clustering analysis

- Clustering analysis falls under the umbrella of exploratory data analysis.

- Although there is quiet a bit of literature on using validation techniques that have a confirmatory / inferential flavor – its sort of a madhouse though.

Clustering analysis

- When are you going to use clustering…

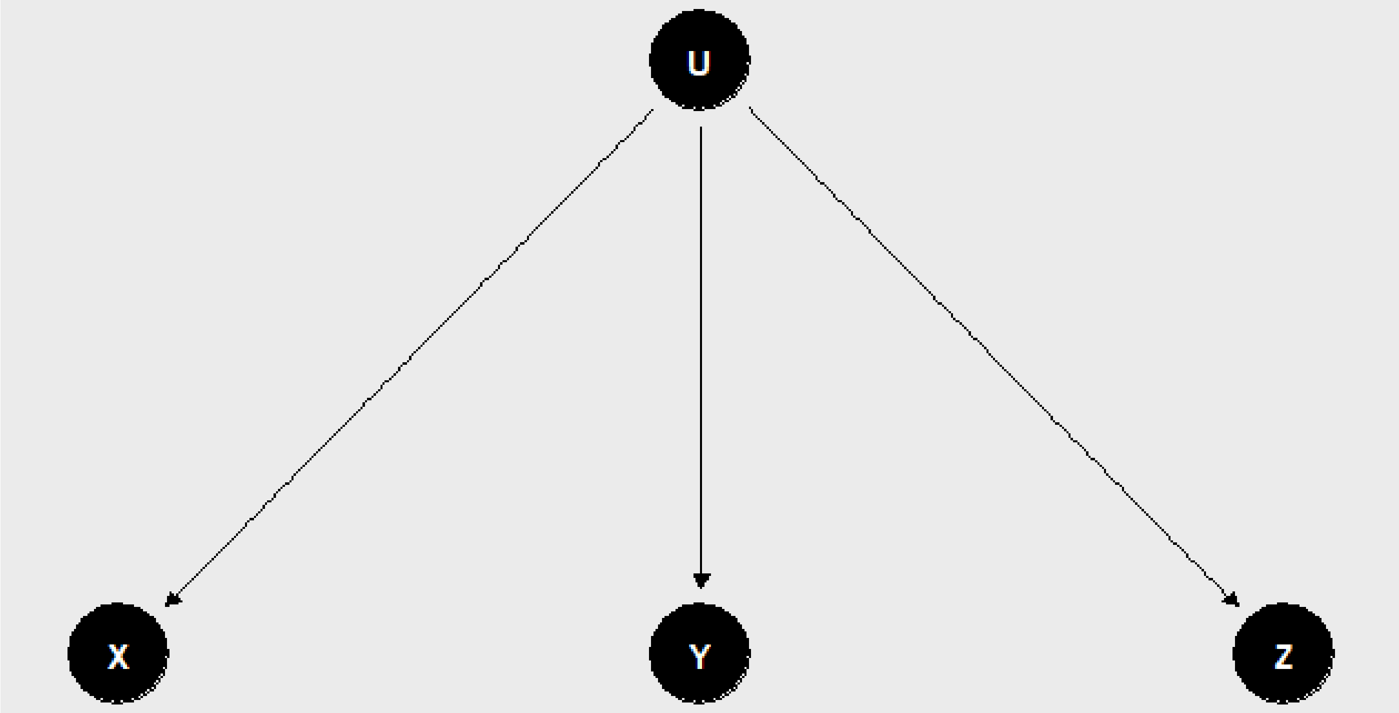
- You don’t have a distinction between dependent and independent variables in your data, just a hypothesis (or set of hypotheses) about the distribution of variables being related by being in groups.

Clustering analysis

- U is an an unmeasured variable (asume its a factor)

- Let us say that U is responsable for the patterns we find in X, Y and Z.

- Can we figure out the value of U for any given datapoint given X, Y and Z?



Clustering analysis

- You can use various algorithms that group data points into groups according to overall closeness in a set of variables considered.

- But there are a lot of different clustering algorithms.

- Why?

Problems with clustering

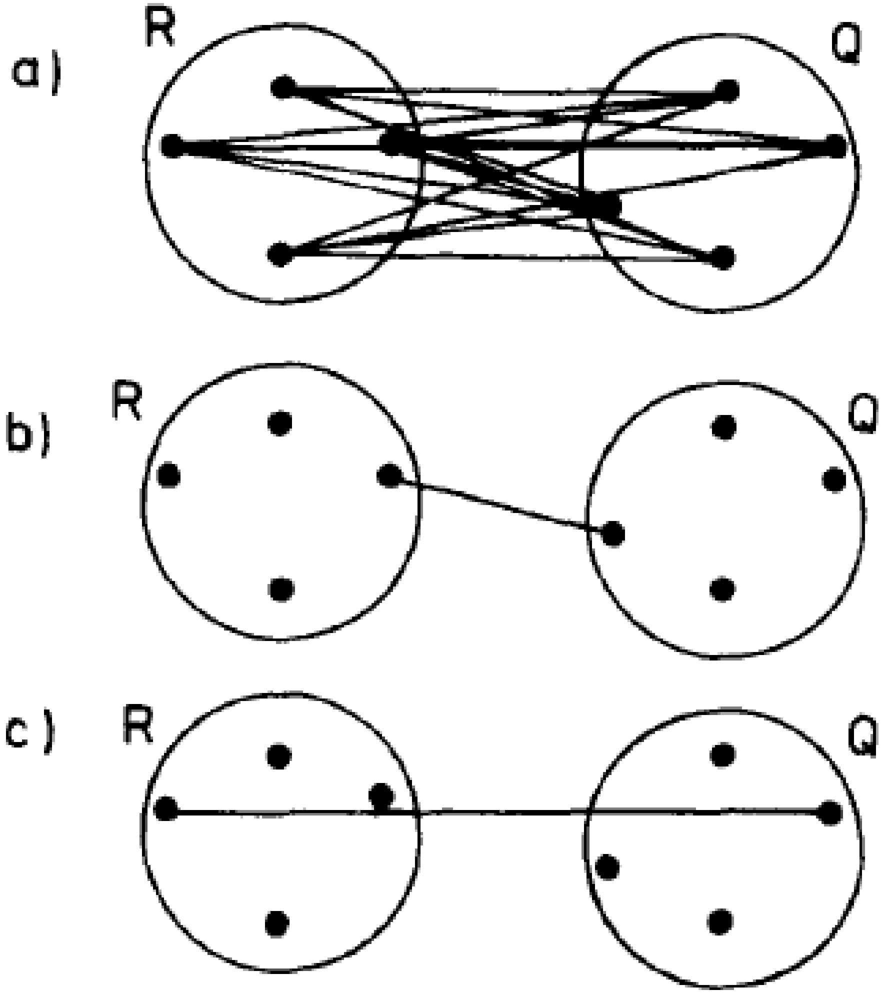
- **Cluster** has no meaning: There “is not standard or even useful definition of the term “cluster”, and many have argued that it is either too late or irrelevant to create one” (Aldenderfer & Blashfield 1984:33).

- Clustering tendency problem: “the problem of deciding whether data exhibit a predisposition to cluster into natural groups without identifying the groups themselves. Clustering algorithms will create clusters whether the data are naturally clustered or purely random” (Jain & Dubes (1988: 201)

Cluster has no definition

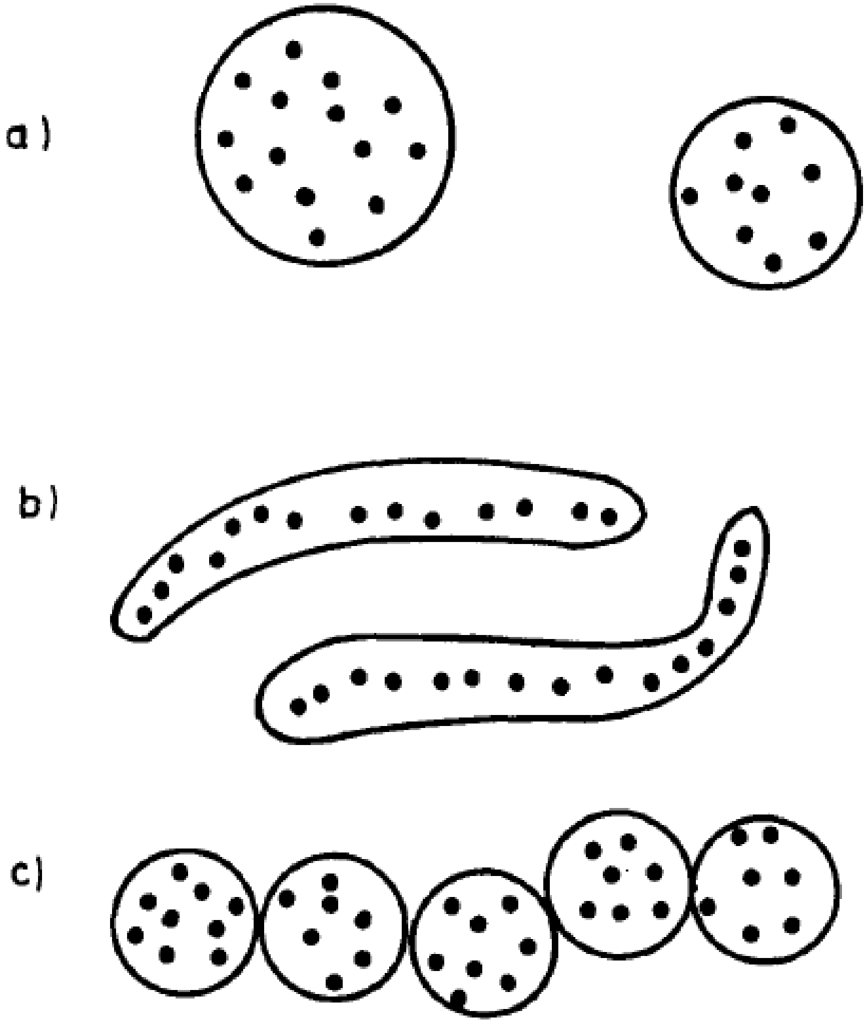
- Clusters are calculated based on some measurement of dissimilarity between datapoints.

- Representation of some definition of intercluster dissimilarity; (a) Group average; (b) Nearest neighbor; (c) Furthest neighbour
- (From Kauffman 1980: 47)



Cluster has no definition

- Types of clusters
- (a) Ball shapes
- (b) elongated
- (c) Compact but not well separated.



Domain dependence

“An ideal cluster can be defined as a set of points that is *compact* and *isolated.* In reality, a cluster is a subjective entity that is in the eye of the beholder and whose significance and interpretation requires domain knowledge. But, while humans are excellent cluster seekers in two and possibly three dimensions, we need automatic algorithms for high-dimensional data.”

(Jain 2010: 652)

Clustering algorithms

- As EDA, clustering is well-known to be a useful tool, but as CDA …

> The clustering tendency problem has not received a great deal of attention but is certainly an important problem. One want to believe that data are clustered and is naturally biased toward believing the results of a cluster analysis….
> The validation of clustering structures is the most difficult and frustrating part of cluster analysis. Without a strong effort in this direction, cluster analysis will remain a black art accessible only to those true believers who have experience and great courage" (Jain & Dubes 1988)

Cluster analysis

- Cluster analysis is used in …
    - Image segmentation
    - Document classification
    - Information retrieval
    - Marketing by classifying
    - Workforce management
    - Genome data

Clustering

- Clustering methods can be divided into *hierarchical* and *partitional*
- **Hierarchical**: Find nested hierarchically organized groups.
- **Partitional**: No imposition of hierarchical structure.
- In both cases, your data has to be organized into a (dis)similarity matrix (a matrix that tells you the distance of each data point vis-à-vis the variables in your dataframe).

Hierarchical clustering

Hierarchical clustering

- There are two main approaches to hierarchical clustering
    - Agglomerative: bottom-up approach – and algorithm assumes that each data point is a group and groups those groups together one by one starting with the most similar ones, until all the data is one large cluster.
    - Divisive: top-down approach: Starts with everything in one large structure and breaks it down into smaller groups…
    - We are going to walk through the agglomerative approach because its more common.

Closeness

- In order to start thinking about clustering we need some notion of "closeness" or "distance" between two data points.

- **Euclidean:** The straight line distance between two points.

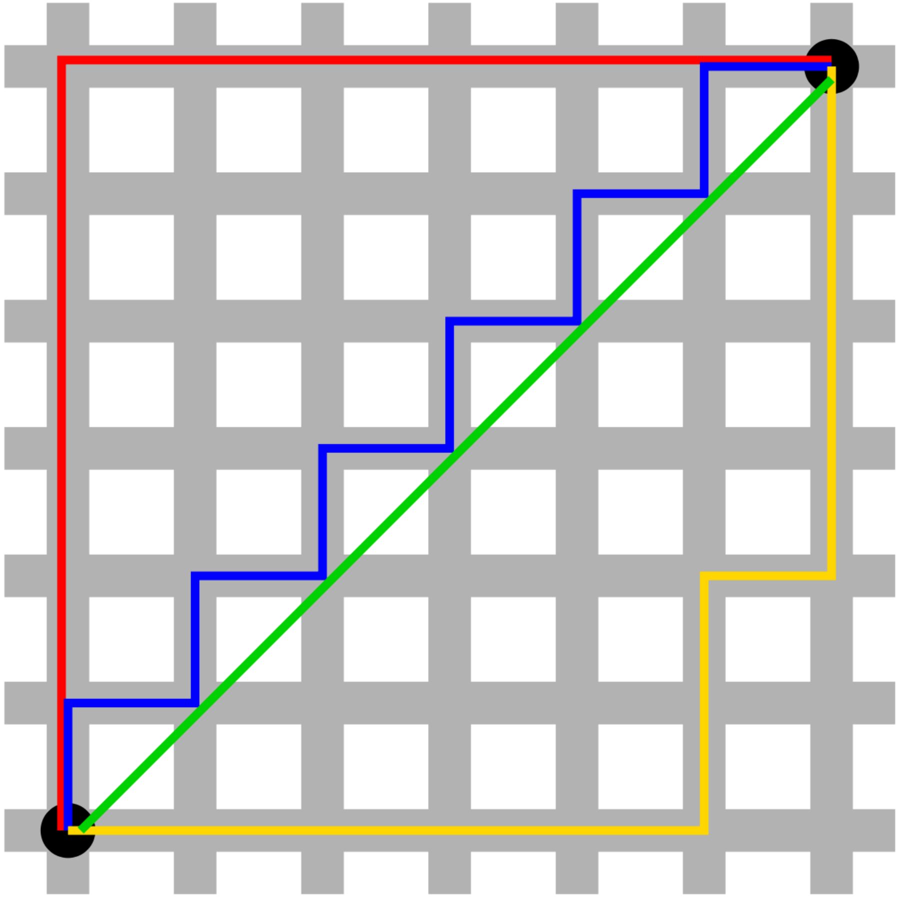
- **Manhattan**: On a grid or lattice, how many units do you have to travel.

Euclidean vs. Manhattan

- Euclidean is green.
- Manhattan is all the other colors

$$Euclidean = [(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + ... + (Y_n - Y_n)^2]^{1/2}$$

$$Manhattan = |A_1 - B_1| + |A_2 - B_2| + ... + |A_n - B_n|$$

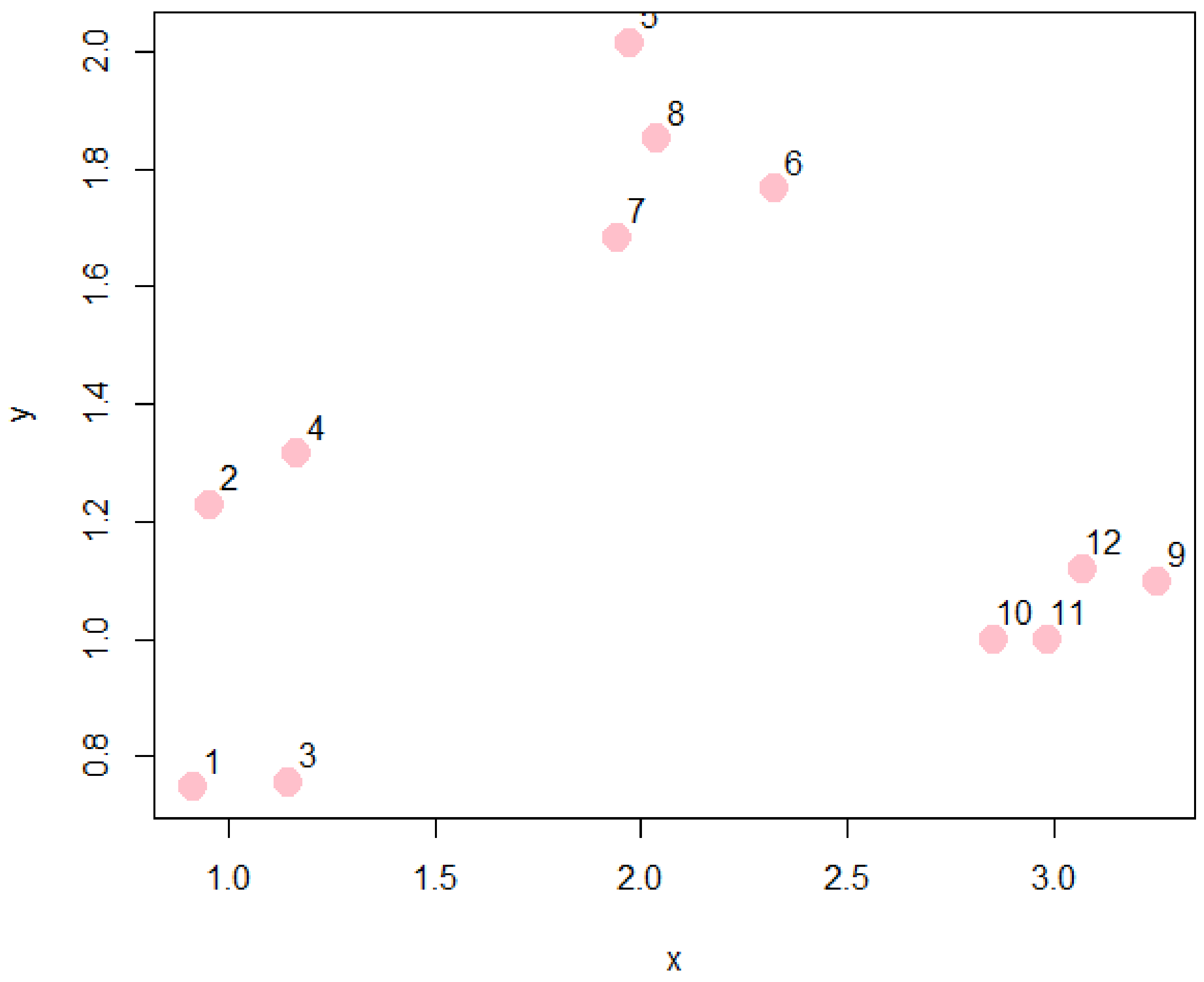


https://commons.wikimedia.org/wiki/File:Manhattan_distance.svg

Agglomerative hierarchical cluster

- Simulating data for clustering

```
set.seed(4321)
x <- rnorm(12, rep(1:3, each = 4), 0.2)
y <- rnorm(12, rep(c(1, 2, 1), each = 4), 0.2)
plot(x, y, col = "pink", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
```



Calculating a distance matrix

- You can calculate a distance matrix with the following code.

```
dataFrame <- data.frame(x=x, y=y)
dist(dataFrame)
```

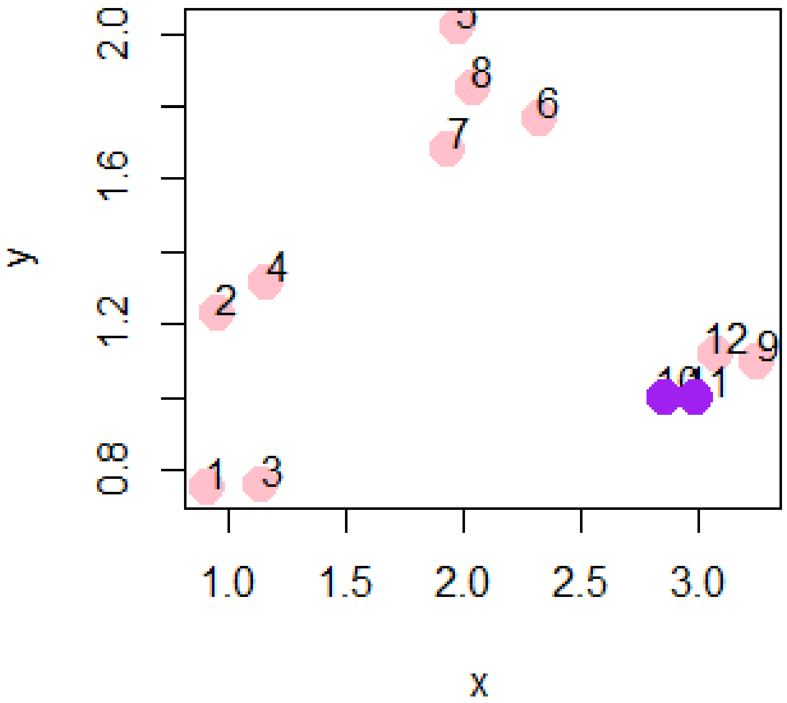
```
 1         2         3         4         5         6         7         8
2  0.4818060
3  0.2290071 0.5083847
4  0.6210186 0.2300065 0.5595680
5  1.6516473 1.2874484 1.5084603 1.0675884
6  1.7363507 1.4683435 1.5515341 1.2383578 0.4279533
7  1.3877613 1.0850453 1.2223180 0.8553386 0.3340584 0.3901532
8  1.5749957 1.2500236 1.4145687 1.0225625 0.1765242 0.2953299 0.1949666
9  2.3594401 2.2966216 2.1320837 2.0912494 1.5701360 1.1421832 1.4327423 1.4247104
10 1.9578445 1.9146662 1.7299970 1.7171607 1.3448566 0.9336432 1.1423308 1.1798333
11 2.0869817 2.0442189 1.8589258 1.8455955 1.4345148 1.0147559 1.2498504 1.2743264
12 2.1859159 2.1164159 1.9592779 1.9106631 1.4145273 0.9877883 1.2612721 1.2632629
```

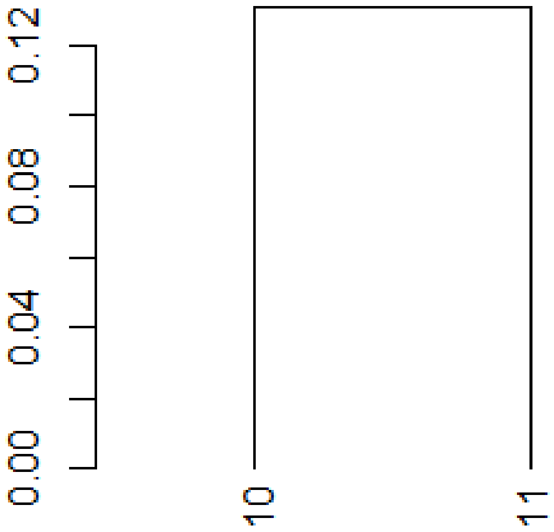
Code for displaying the first merge

```
## Remove diagonal from consideration
diag(rdistxy)
diag(rdistxy) <- diag(rdistxy) + 100000
##Find the index of the points with minimum distance
ind <- which(rdistxy == min(rdistxy), arr.ind = TRUE)
ind
par(mfrow = c(1, 2))

plot(x, y, col = "pink", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
points(x[ind[1, ]], y[ind[1, ]], col = "purple", pch = 19, cex = 2)

hcluster <- dist(dataFrame)
dendro <- as.dendrogram(hclust(hcluster))
plot(cut(dendro, h = 0.5)$lower[[3]])
```

Agglomeration

- Agglomerative method starts with the closest point and turns it into a group.



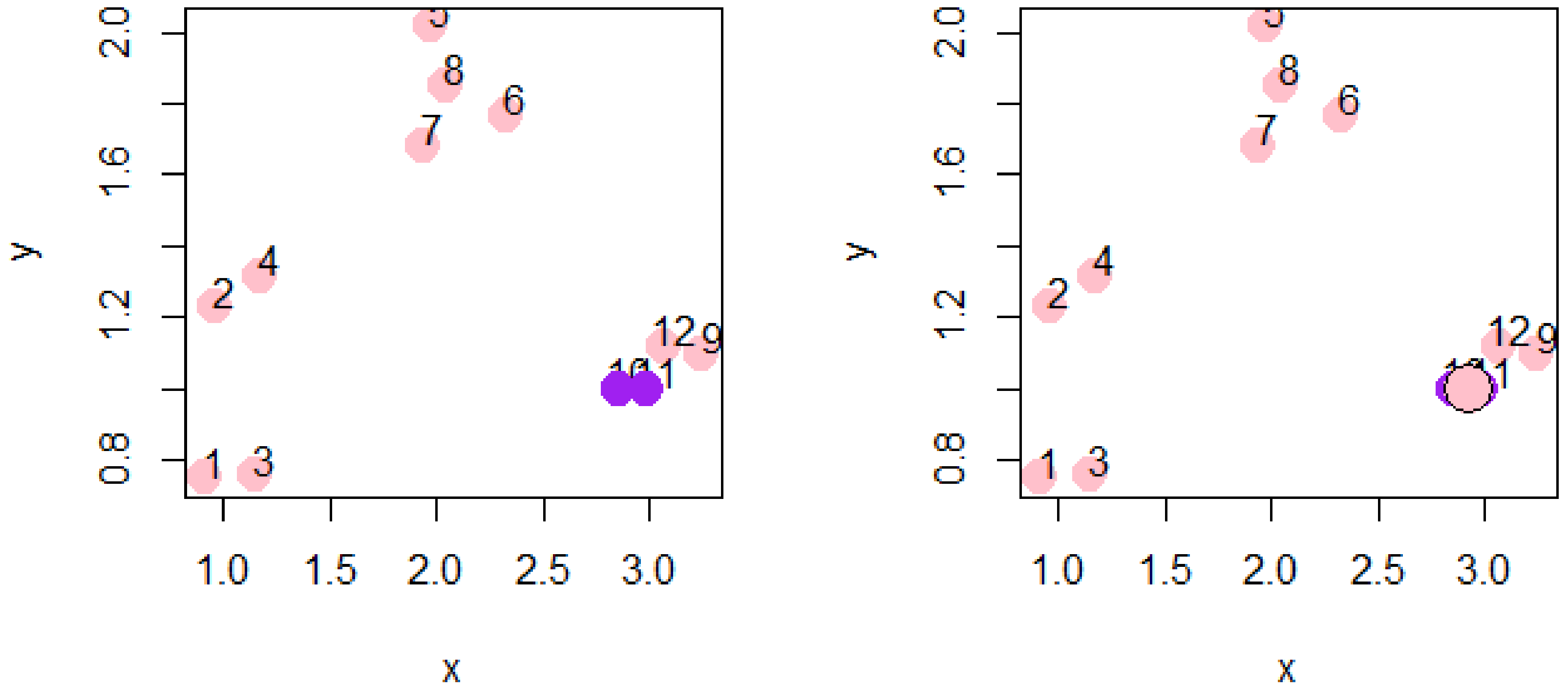


Code for second merge

```
nextmin <- rdistxy[order(rdistxy)][7]
ind2 <- which(rdistxy == nextmin,arr.ind=TRUE)
ind2
##Showing agglomeration
par(mfrow = c(1, 2))
plot(x, y, col = "pink", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
points(x[ind[1, ]], y[ind[1, ]], col = "purple", pch = 19, cex = 2)
plot(x, y, col = "pink", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
points(x[ind[1, ]], y[ind[1, ]], col = "purple", pch = 19, cex = 2)
symbols(x=c(2.93), y=c(1), circles=0.12, add=T, inches=F,
pch=20, bg="pink")
```

Agglomeration

- After a group is formed it is treated as a single point in the next iteration of the algorithm

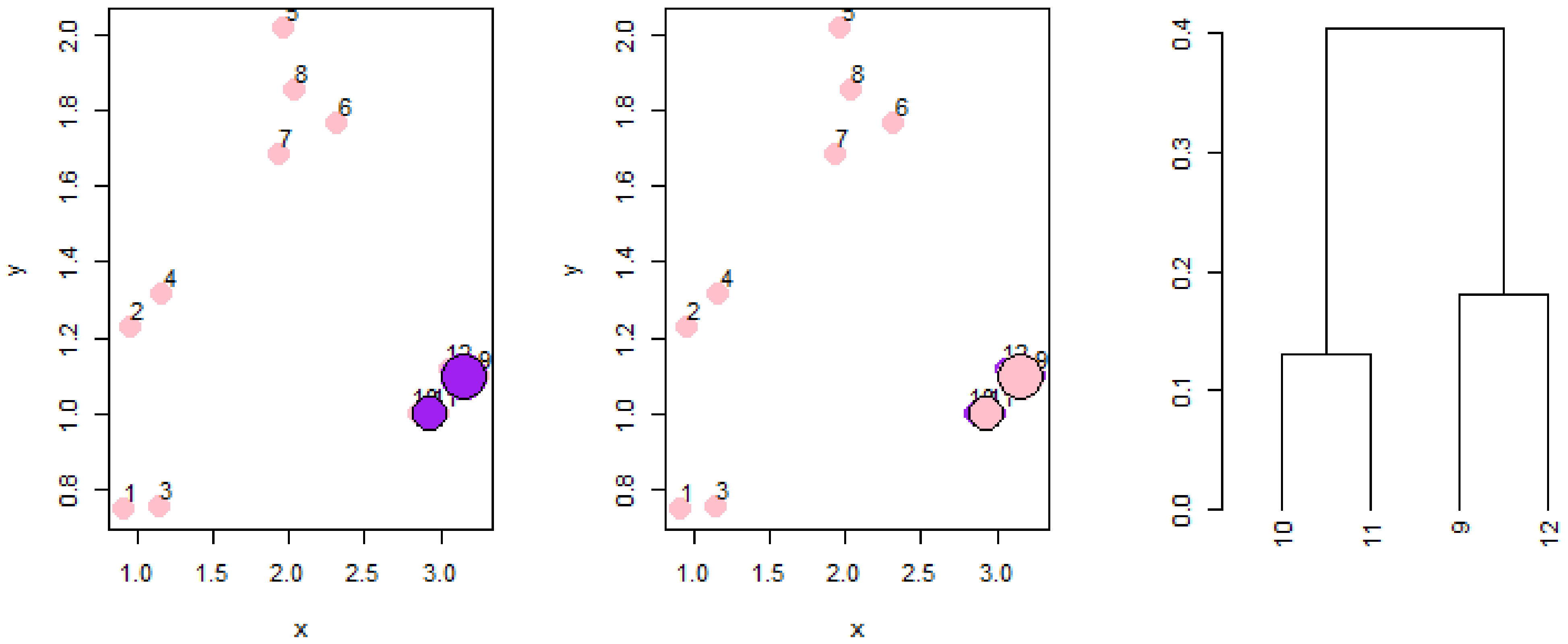


Code for cut dendrogram

```
par(mfrow = c(1, 3))

plot(x, y, col = "pink", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
symbols(x=c(2.93), y=c(1), circles=0.12, add=T, inches=F, pch=20, bg="purple")
symbols(x=c(3.15), y=c(1.1), circles=0.15, add=T, inches=F, pch=20, bg="purple", cex=2)

plot(x, y, col = "pink", pch = 19, cex = 2)
text(x + 0.05, y + 0.05, labels = as.character(1:12))
points(x[ind[1, ]], y[ind[1, ]], col = "purple", pch = 19, cex = 2)
symbols(x=c(2.93), y=c(1), circles=0.12, add=T, inches=F, pch=20, bg="pink")
points(x[ind2[1, ]], y[ind2[1, ]], col = "purple", pch = 19, bg = "purple", cex=2)
symbols(x=c(3.15), y=c(1.1), circles=0.15, add=T, inches=F, pch=20, bg="pink", cex=2)

hcluster <- dist(dataFrame)
dendro <- as.dendrogram(hclust(hcluster))
plot(cut(dendro, h = 0.5)$lower[[3]])
```

Agglomeration

- After a group is formed it is treated as a single point in the next iteration of the algorithm

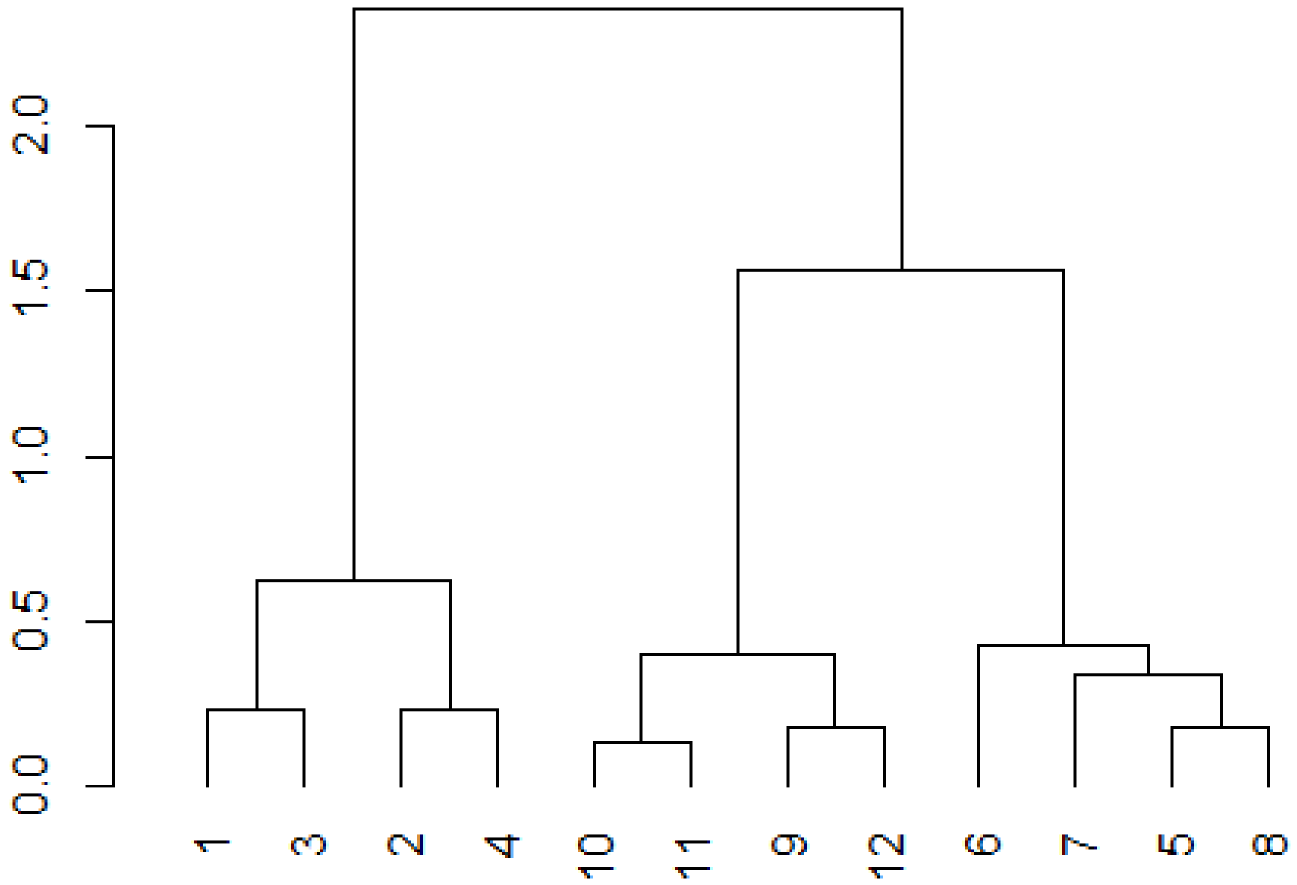


Dendrogram

- The algorithm continues until it builds a tree.

- This is called a dendrogram

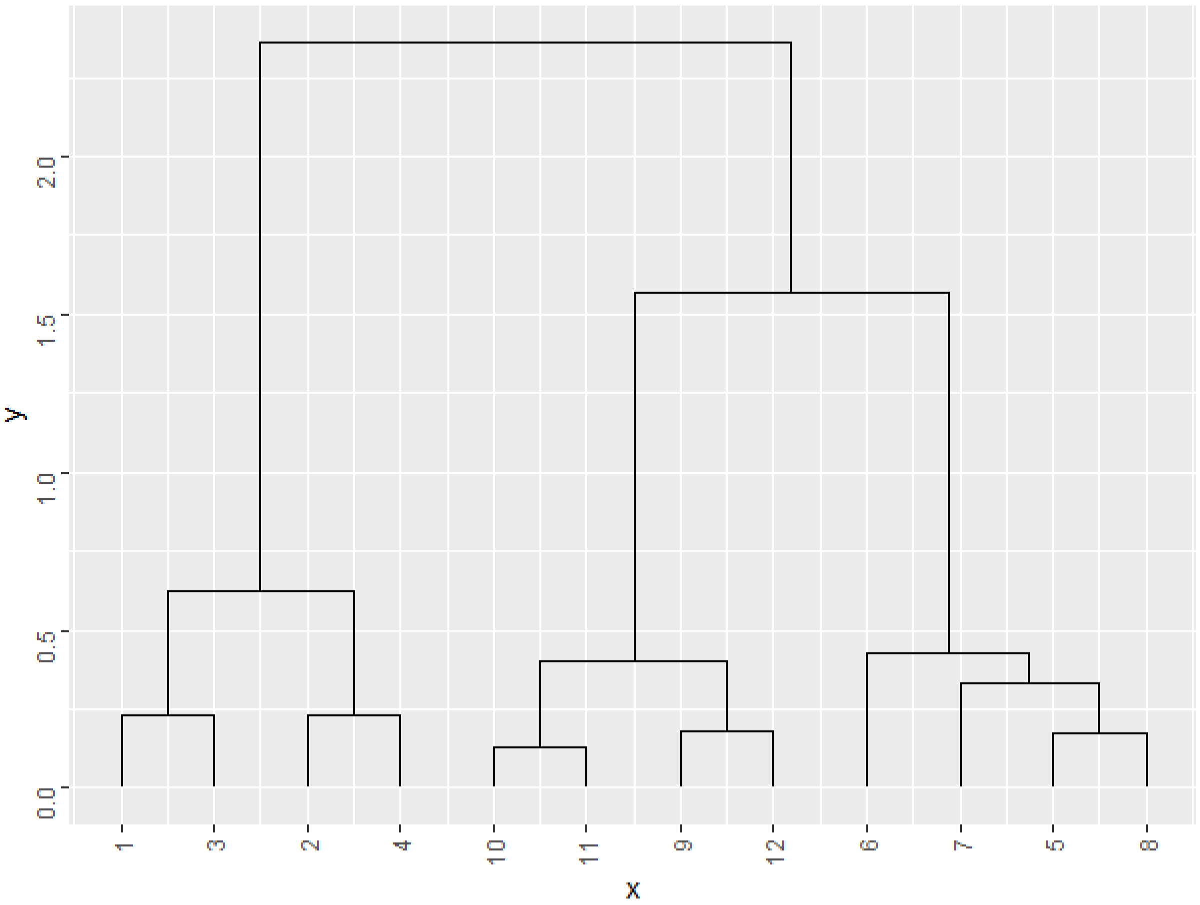
plot(dendro)



ggdendro

- You can get nicer dendrograms with ggdendro()

```
library(ggplot2)
library(ggdendro)

ggdendrogram(dendro,
theme_dendro = FALSE)
```



Assign clusters to vectors

- You can assign the cluster labels back to the data.

```
groups <- cutree(clusters, k=2)
dataFrame$groups3 <- groups
dataFrame
 x        y groups3
1  0.9146485 0.7478030       1
2  0.9552776 1.2278928       1
3  1.1435214 0.7556436       1
4  1.1682891 1.3146632       1
5  1.9743285 2.0146956       2
6  2.3218694 1.7649770       2
7  1.9405665 1.6823476       2
8  2.0392009 1.8505239       2
9  3.2481492 1.0967044       2
10 2.8562604 0.9993949       2
11 2.9865527 0.9982139       2
12 3.0688734 1.1186715       2
```

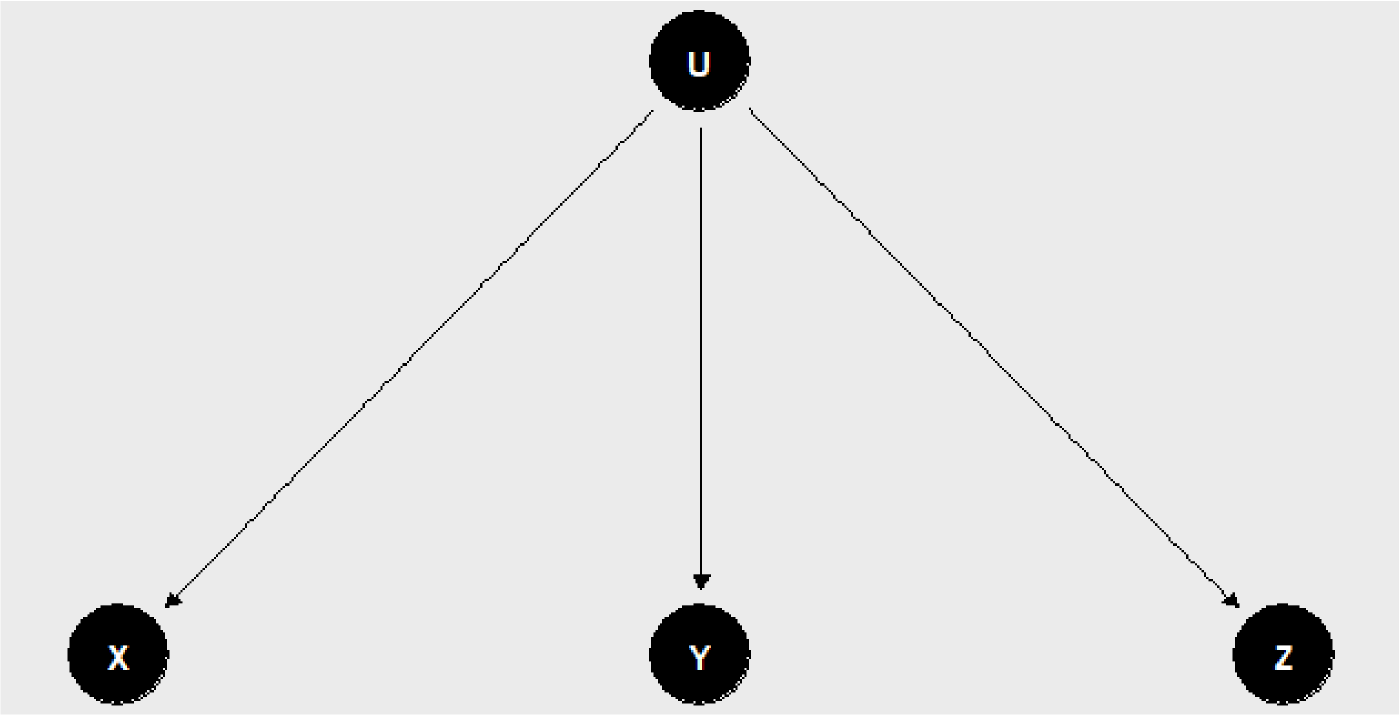
Cluster validation

- There are actually lots of R packages now that do types of cluster validation – providing p-values for cluster labels etc.
- It is important to recognize that cluster validation (along with clustering itself) is very problem specific.
- One common way of validating clusters is to compare the results to some null distribution.

Cluster validation

- Let’s imagine that there are two values for U and they produce differential effects for x, y and z.

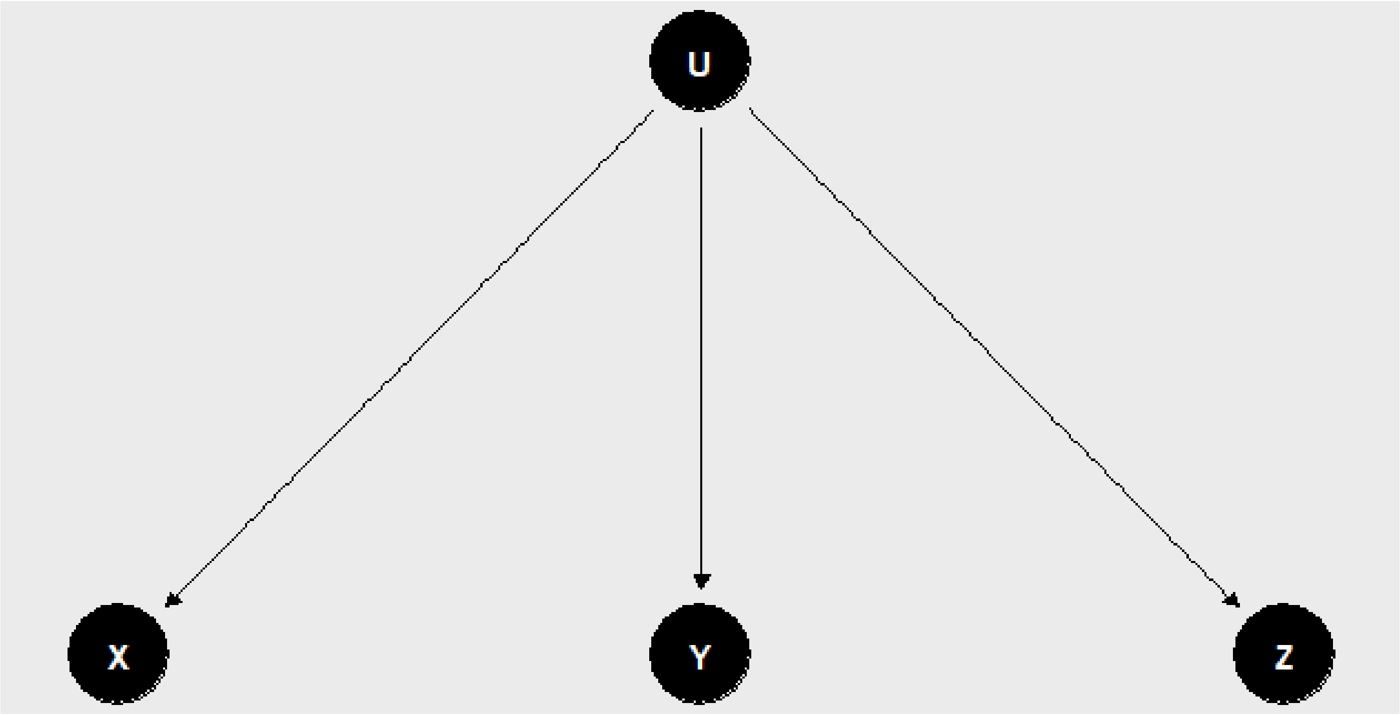
- But we do not actually know what U is.



Cluster validation

- Simulate data based on the causal graph.

```
u <- rbinom(50,1,.4)
bx <- 0.5
x <-  bx*u + rnorm(50, 0, 1)
by <- -3
y <- by*u + rnorm(50, 0, 2)
bz <- 4
z <- bz*u + rnorm(50, 0, 2.5)
d <- data.frame(x,y,z)
glimpse(d)
```



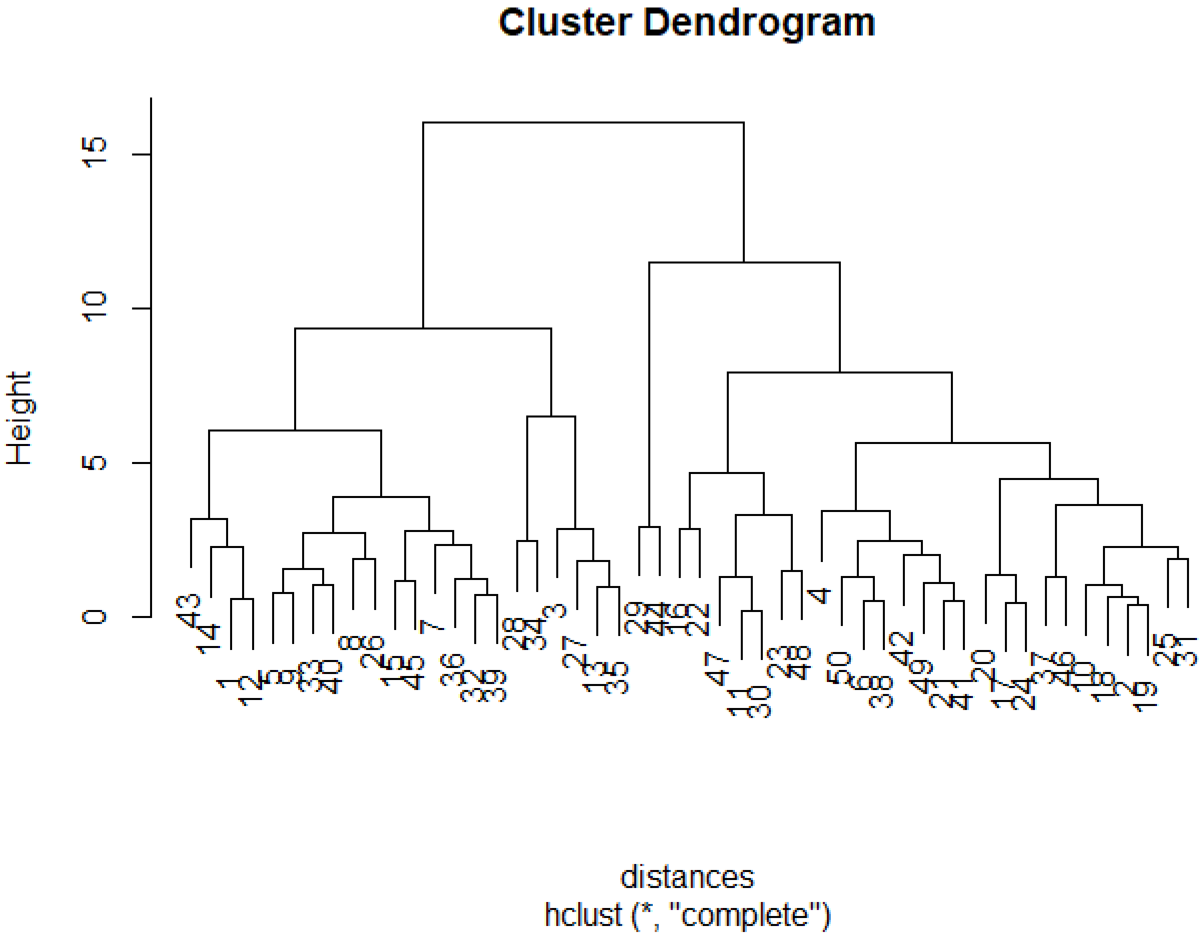
Cluster validation

- Draw a dendrogram

distances <- dist(d) #Make a distance matrix

hc <- hclust(distances) #Make a hierarchical cluster

plot(hc) #plot the hierarchical cluster



Cluster validation

- An **external** criterion would involve just assessing the degree to which the cluster model correctly got U

```
g <- cutree(hc, k=2)
d$g <- g -1
glimpse(d)
d$u <- u
table(d$u, d$g)
chisq.test(d$u, d$g)
```

```
        0       1
0       22      9
1       0       19
```

```
Pearson's Chi-squared test with Yates' continuity correction
data:  d$u and d$g
X-squared = 21.284, df = 1, p-value = 3.96e-06
```

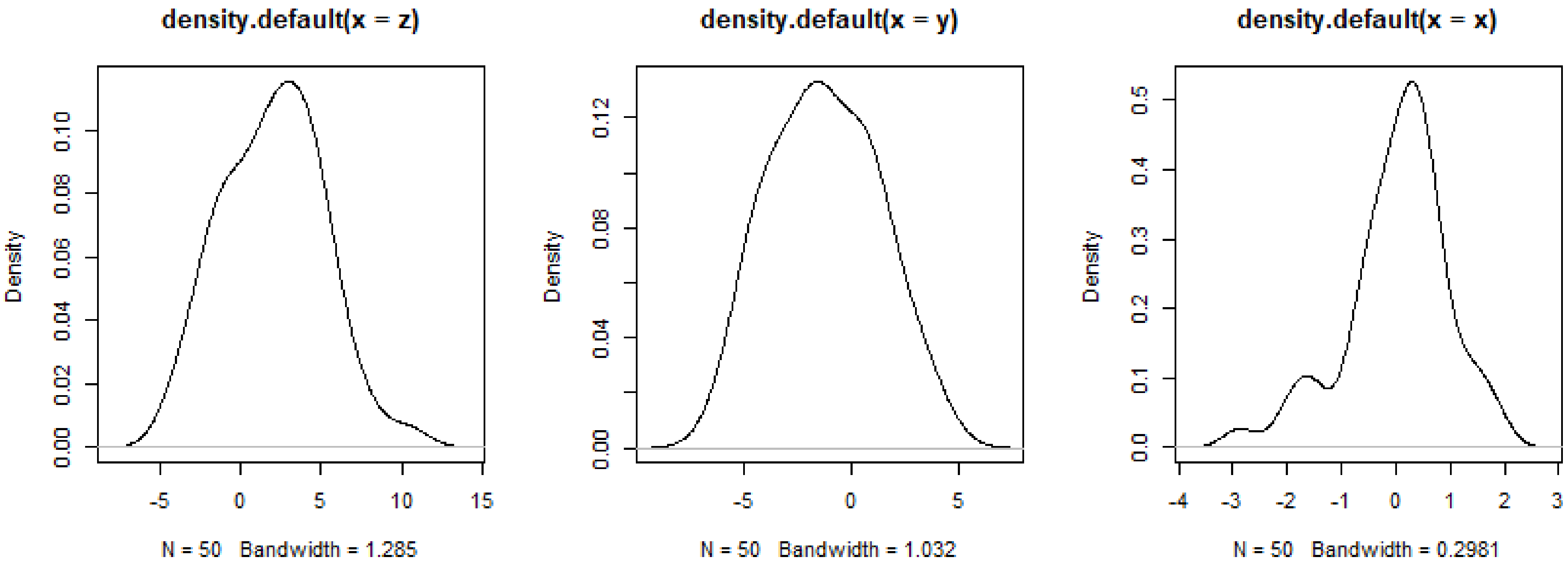
Cluster validation

- But the real world isn’t like this, otherwise we would just do a regression.
- An internal criteria often (but not always) involves seeing if the groups are clustered into their respective groups better than chance.

- So we simulate a null distribution and compare the results

Cluster validation

- We simulate null distributions *as if* U wasn’t a factor.



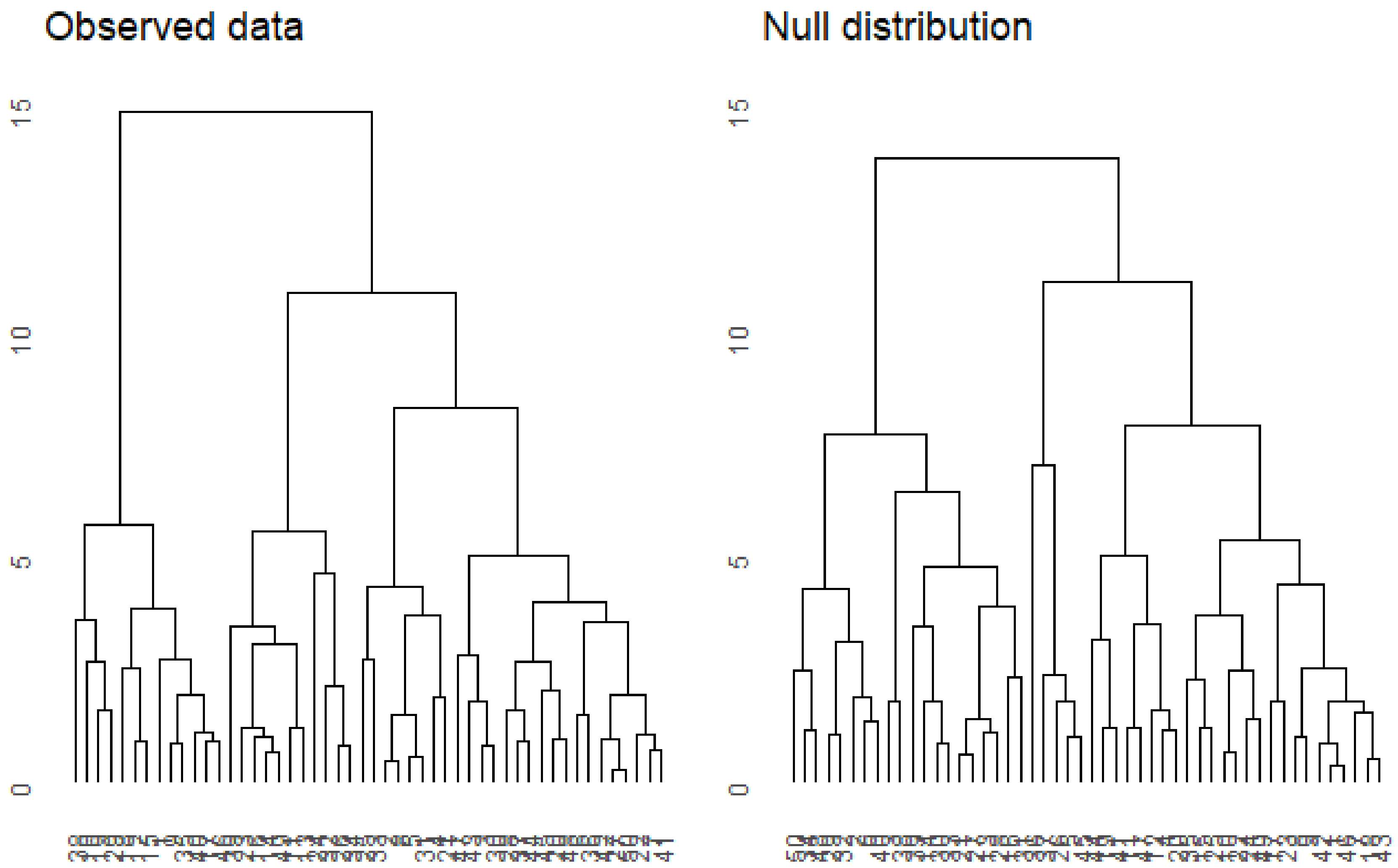
Cluster validation

- Based on what we just saw, we can posit that the null distribution would be all of normally distributed.
- We just construct a situation where they are unrelated to one another.

```
z.null <- rnorm(50, mean(z), sd(z))
y.null <- rnorm(50, mean(y), sd(y))
x.null <- rnorm(50, mean(x), sd(x))
d.null <- data.frame(x.null,y.null,z.null)
d.obs <- data.frame(x,y,z)
distances.null <- dist(d.null) #Make a distance matrix
hc.null <- hclust(distances.null) #Make a hierarchical cluster
distances.obs <- dist(d.obs) #Make a distance matrix
hc.obs <- hclust(distances.obs) #Make a hierarchical cluster
```

Cluster validation

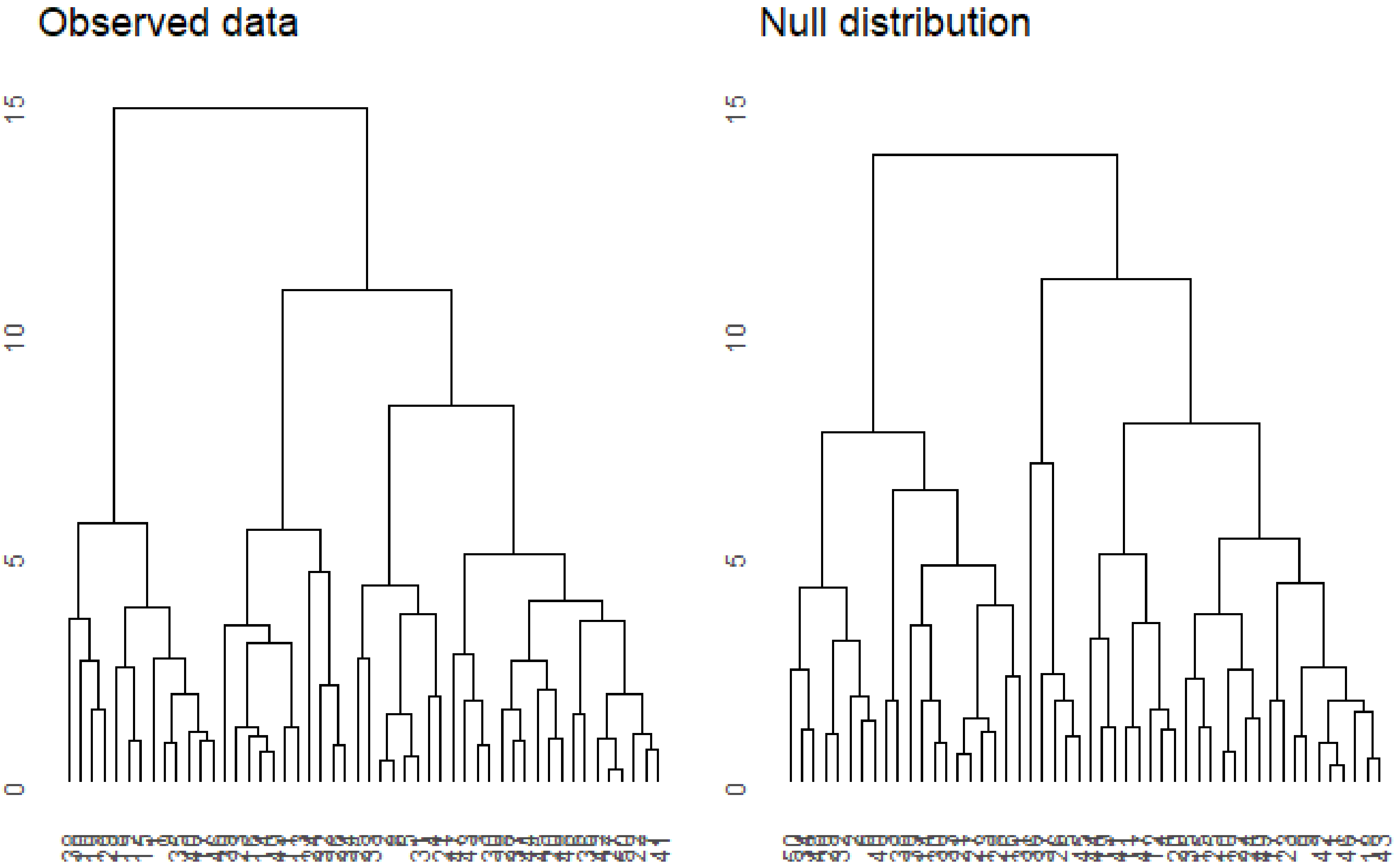
```
p1 <- ggdendrogram(
  as.dendrogram(hc.obs))+
  ylim(0,15)+
  ggtitle("Observed data")
p2 <- ggdendrogram(
  as.dendrogram(hc.null))+
  ylim(0,15)+
  ggtitle("Null distribution")
grid.arrange(p1,p2, ncol=2)
```



Cluster validation

- The height of the first partition is 3.9 for the observed data.
- The height of the first partition for the simulated data is 2.7

```
length(hc.obs$height)
length(hc.null$height)
hc.obs$height[49] - hc.obs$height[48]
hc.null$height[49] - hc.null$height[48]
```



Observed data

Null distribution

Cluster validation

- In such cases, you’ll also want to calculate the cophenetic correlation.
- This is the correlation between the distances based on clusters and the distances based on euclidean distance.

```
coph <- cophenetic(hc.obs)
cor.test(distances.obs, coph)
coph <- cophenetic(hc.null)
cor.test(distances.null, coph)
```

```
Pearson's product-moment correlation

data:  distances.obs and coph
t = 26.913, df = 1223, p-value < 2.2e-16
alternative hypothesis: true correlation is not
equal to 0
95 percent confidence interval:
 0.5734549 0.6438902
sample estimates:
      cor
0.6098755
```

```
Pearson's product-moment correlation

data:  distances.null and coph
t = 24.293, df = 1223, p-value < 2.2e-
16
alternative hypothesis: true
correlation is not equal to 0
95 percent confidence interval:
 0.5314773 0.6071135
sample estimates:
      cor
0.5705038
```

Warning

- There are lots of other validation techniques – and you can find many proposals implemented in R online.

- Using clustering models requires a lot of thinking about the details of your specific problem – beware of the **clustering tendency problem**!

K-means clustering

K-means

- K-means is the most used partitional clustering method.

- You are trying to find center points over a fixed number of cloud like clusters in a hyper-dimensional space.

- You probably don’t need K-means when you just have 2-dimensions…, but for more it becomes a useful tool.

K-means

- Unlike hierarchical clustering you have to prespecify the number of clusters.

- There’s no straightforward formula for getting the results of K-means clustering.

- We use algorithms instead….

K-means

- Rough summary of the algorithm

1. Fix the number of clusters at 2 or more
2. Centroids are placed randomly.
3. Points closest to each centroid are placed within its cluster
4. Reposition centroids and calculate again until distances are minimized.

Hierarchical clustering

Affix-word continuum

- Hierarchical clustering with affix-word database.

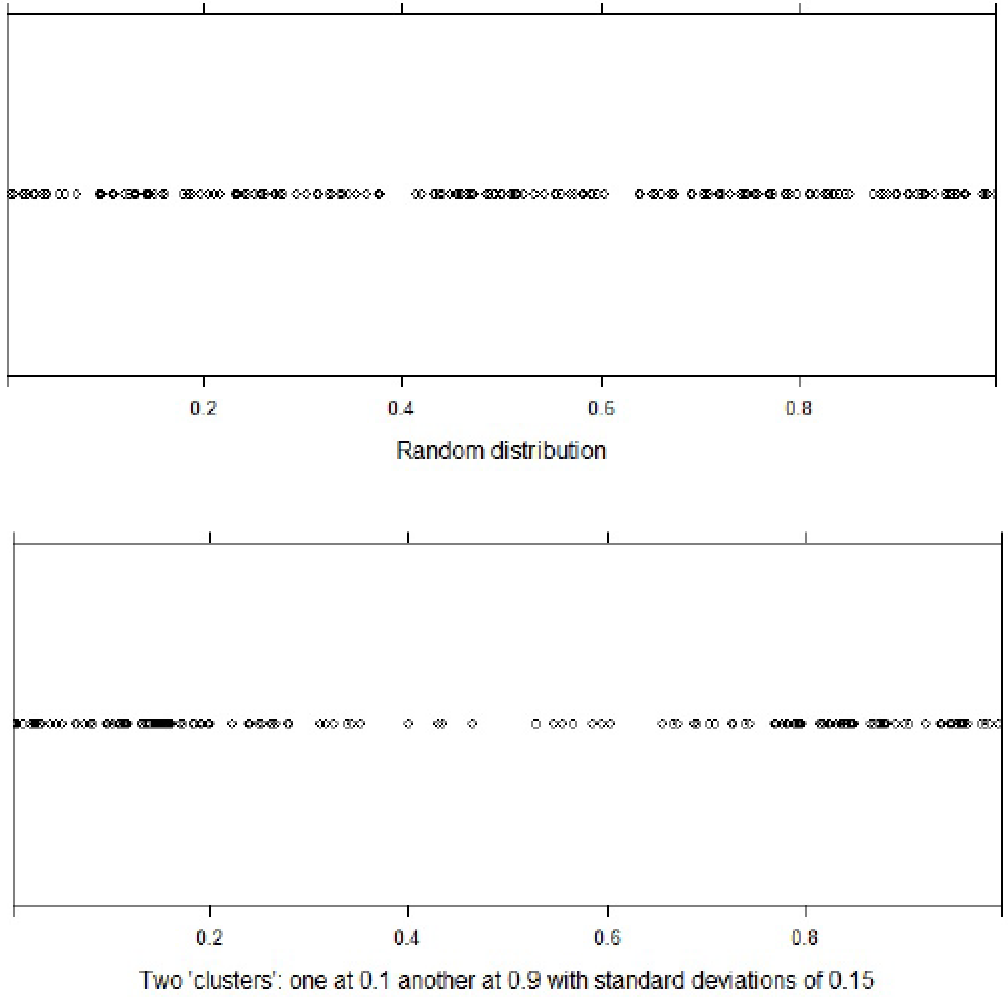


Figure 5: The affix-word continuum: two hypothetical situations

Reading

- Levshina Chapter 15
- Baayen Chapter 5
- Bell et al. 2018