

# PLSC 504

Principal Components Analysis, Factor  
Analysis, and Clustering

October 🍂, 2022

- Often: *data reduction* (many  $\mathbf{X} \rightarrow$  one  $X$ )
- Classification *is* measurement (taxonomy and typology)
- Level and quality of measurement are distinct
- **All measurement implies theory**

- Principal Components Analysis (+ biplots!)
- Factor Analysis
- Cluster Analysis

## Reviewing Some Basics

```
> X <- data.frame(X1=c(0,1,2,3),X2=c(6,5,3,0),  
                  X3=c(7,9,10,13),X4=c(4,1,7,4))
```

```
> X
```

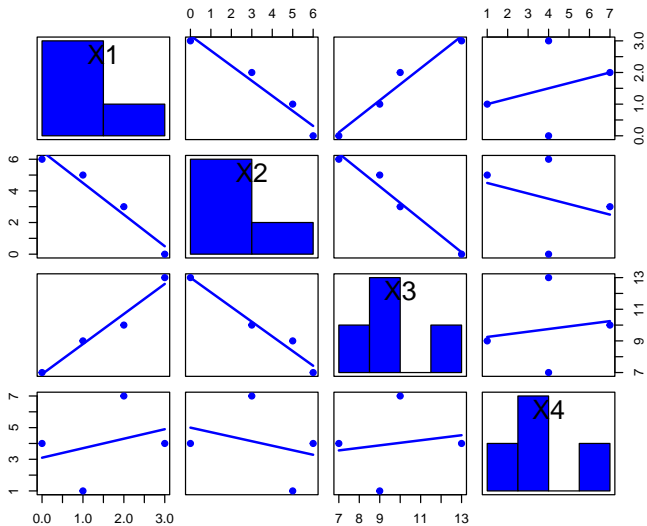
	X1	X2	X3	X4
1	0	6	7	4
2	1	5	9	1
3	2	3	10	7
4	3	0	13	4

```
> CX <- sweep(X,2,colMeans(X),"-") # "centered" X
```

```
> CX
```

	X1	X2	X3	X4
1	-1.5	2.5	-2.75	0
2	-0.5	1.5	-0.75	-3
3	0.5	-0.5	0.25	3
4	1.5	-3.5	3.25	0

# Toy Data



```
> Sigma <- cov(CX) # variance-covariance matrix
```

```
> Sigma
```

	X1	X2	X3	X4
X1	1.667	-3.333	3.167	1
X2	-3.333	7.000	-6.500	-2
X3	3.167	-6.500	6.250	1
X4	1.000	-2.000	1.000	6

```
> R <- cor(CX) # correlation matrix
```

```
> R
```

	X1	X2	X3	X4
X1	1.0000	-0.9759	0.9812	0.3162
X2	-0.9759	1.0000	-0.9827	-0.3086
X3	0.9812	-0.9827	1.0000	0.1633
X4	0.3162	-0.3086	0.1633	1.0000

# Eigenvalues and Eigenvectors

For the variance-covariance matrix  $\Sigma$  of (centered)  $\mathbf{X}$ , we can diagonalize:

$$\Sigma = \mathbf{V}\mathbf{L}\mathbf{V}'$$

where

- $\mathbf{V}$  is the matrix of *eigenvectors* (“principal axes”), and
- $\mathbf{L}$  is the (diagonal) matrix of *eigenvalues*.

Things:

- The sum of the eigenvalues equals the *trace* of  $\Sigma$
- The product of the eigenvalues is  $|\Sigma|$

(Kinda confused? Check out [this](#), or [this](#), or [this](#).)

# Eigenvalues and Eigenvectors

```
> E <- eigen(Sigma)
> E
eigen() decomposition
$values
[1] 1.534e+01 5.491e+00 8.110e-02 9.845e-16

$vectors
      [,1]      [,2]      [,3]      [,4]
[1,]  0.3249 -0.04222  0.67351  0.6626
[2,] -0.6712  0.09804  0.65587 -0.3313
[3,]  0.6197 -0.25156  0.33709 -0.6626
[4,]  0.2447  0.96194  0.05088 -0.1104

> L <- E$values
> V <- E$vectors

> sum(E$values)
[1] 20.92

> tr(Sigma)
[1] 20.92
```



# Singular Value Decomposition

The *singular value decomposition* (SVD) of  $\mathbf{X}$  is:

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}'$$

where  $\mathbf{S}$  is the diagonal matrix of *singular values*,  $\mathbf{U}$  is a unitary (orthogonal) matrix, and  $\mathbf{V}$  is again the matrix of eigenvectors.

Note:

- Elements of  $\mathbf{S}$   $s_i$  are related to the eigenvalues  $v_i$  according to  $v_i = s_i^2 / (N - 1)$ .
- The *principal components* are equal to  $\mathbf{US}$  ( $\equiv \mathbf{XV}$ ).

```

> SVD <- svd(CX)
> SVD
$d
[1] 6.7848667363326811142 4.0586048392632072535 0.4932647655322611735
[4] 0.00000000000000002276

$u
      [,1]      [,2]      [,3] [,4]
[1,] -0.5703  0.2464  0.6033  0.5
[2,] -0.3490 -0.6231 -0.4898  0.5
[3,]  0.2045  0.6783 -0.4982  0.5
[4,]  0.7149 -0.3016  0.3846  0.5

$v
      [,1]      [,2]      [,3]      [,4]
[1,]  0.3249 -0.04222 -0.67351 -0.6626
[2,] -0.6712  0.09804 -0.65587  0.3313
[3,]  0.6197 -0.25156 -0.33709  0.6626
[4,]  0.2447  0.96194 -0.05088  0.1104

> S <- SVD$d
> U <- SVD$u
> otherV <- SVD$v
>
> # Eigenvalues:
>
> (S^2)/(nrow(X)-1)
[1] 1.534e+01 5.491e+00 8.110e-02 1.726e-32

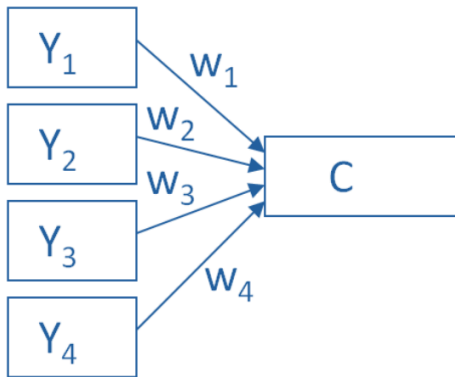
```

# Principal Components (PCA)

PCA is:

- an orthogonal transformation, that
- converts a set of variables  $\mathbf{X}_{N \times K}$  into a set of  $K$  linearly-uncorrelated values, where
- the first principal component has the largest possible variance, and
- the second has the second-highest (subject to orthogonality),
- etc.

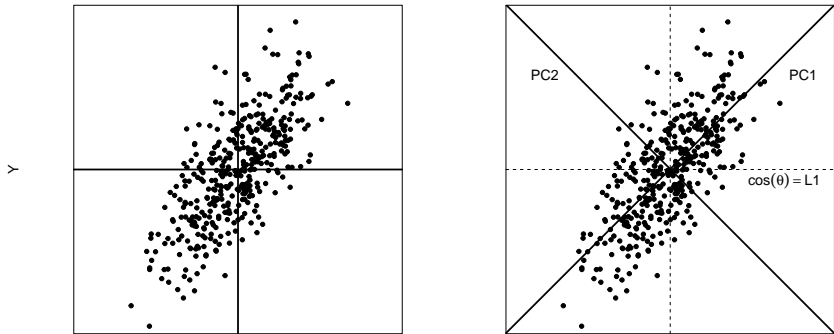
## PCA, Conceptually



$$C = w_1 Y_1 + w_2 Y_2 + w_3 Y_3 + w_4 Y_4$$

(Source)

# PCA Intuition



“(Principal components) can be considered as a rotation of original variable coordinate system to new (orthogonal) axes... such that the new axes coincide with the directions of maximum variation in the original observations.” (Campbell and Atchley 1981)

```
> princomp(CX) # via eigenvalues
```

```
Call:
```

```
princomp(x = CX)
```

```
Standard deviations:
```

```
Comp.1 Comp.2 Comp.3 Comp.4
```

```
3.3924 2.0293 0.2466 0.0000
```

```
4 variables and 4 observations.
```

```
> prcomp(CX) # via SVD
```

```
Standard deviations (1, .., p=4):
```

```
[1] 3.9172446366374114035 2.3432365964829307003 0.2847865451618086241
```

```
[4] 0.0000000000000001314
```

```
Rotation (n x k) = (4 x 4):
```

	PC1	PC2	PC3	PC4
X1	0.3249	-0.04222	-0.67351	-0.6626
X2	-0.6712	0.09804	-0.65587	0.3313
X3	0.6197	-0.25156	-0.33709	0.6626
X4	0.2447	0.96194	-0.05088	0.1104

```
> otherV # from -svd-
```

	[,1]	[,2]	[,3]	[,4]
[1,]	0.3249	-0.04222	-0.67351	-0.6626
[2,]	-0.6712	0.09804	-0.65587	0.3313
[3,]	0.6197	-0.25156	-0.33709	0.6626
[4,]	0.2447	0.96194	-0.05088	0.1104

- *Extract* the principal components
- *Interpret* the components...
- Consider *rotation*
- Choosing the *number of components* (dimensions)
- Generating *scores*

# PCA: A Simulation Example

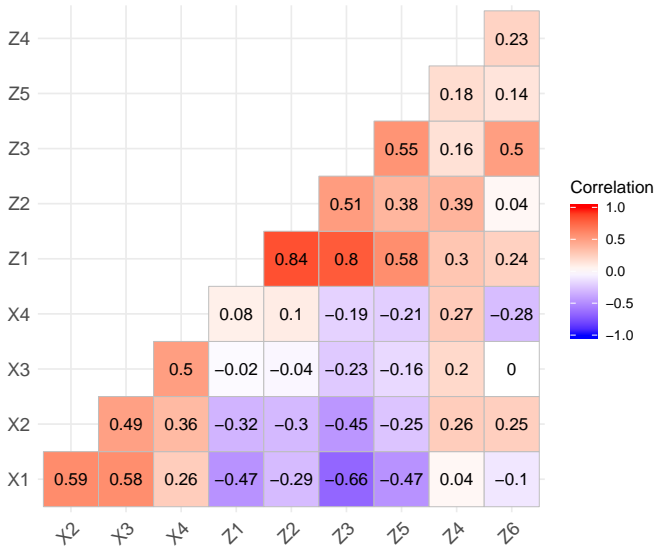
```
> N <- 20
> set.seed(7222009)
> Name <- randomNames(N, which.names="first")
> Z <- rnorm(N)
> Z1 <- Z + 0.2*rnorm(N)
> Z2 <- Z + 0.5*rnorm(N)
> Z3 <- Z + 1*rnorm(N)
> Z4 <- Z + 1.5*rnorm(N)
> Z5 <- Z + 2*rnorm(N)
> Z6 <- Z + 3*rnorm(N)
>
> X <- rnorm(N)
> X1 <- X + rnorm(N)
> X2 <- X + rnorm(N)
> X3 <- X + rt(N,5)
> X4 <- X + rt(N,5)
>
> df <- data.frame(Z1,Z2,Z3,Z4,Z5,Z6,X1,X2,X3,X4)
> rownames(df)<-Name
```



# PCA Simulation: Data

	Z1	Z2	Z3	Z4	Z5	Z6	X1	X2	X3	X4	
Guillermo	-1.2792	-1.19146	-2.647551	-0.4796	-2.5831	-2.4993	1.82780	4.7884	0.00842	2.70394	<--
Rachel	0.6212	0.76287	0.844751	0.4931	0.8397	-4.9998	-2.86988	-2.3021	-1.93999	2.71736	
Deidra	1.0345	0.90827	1.277867	-0.5136	0.1952	-1.0103	-0.58121	-2.1720	0.34358	1.10173	
Quaton	-0.2520	-0.09033	-0.456603	2.4858	-0.1768	-1.1439	2.62444	1.2543	3.40352	0.76009	
Alicia	-0.3546	-0.88944	1.415453	1.7511	-3.3356	4.7557	-0.76564	0.9322	0.25697	1.03088	
Angelique	0.6942	-0.19473	1.610025	-0.8439	2.8762	2.4540	-3.51134	-0.7967	-0.47199	-2.33414	
Johnaton	-0.4012	-0.56944	0.233224	-1.7495	-0.7676	0.3802	1.45076	0.3359	-0.70612	-1.82513	
Javan	0.2157	0.77917	0.006128	-0.3720	1.0505	-1.5419	0.27785	-0.7050	-1.67915	-0.06705	<--
Khulood	-0.1847	0.34502	2.027058	0.6919	-1.1931	3.4407	-3.05080	-1.0084	-4.55345	-2.75049	
Cody	-0.4622	-0.74679	0.739543	-0.7362	-0.1575	-4.0721	-0.66066	-1.6068	-0.88352	-1.57112	
Cameron	0.4913	1.00483	0.991855	0.0758	0.3947	4.7681	1.10206	2.1514	2.49531	0.43303	
Heidi	1.5579	1.27909	3.278130	1.7758	0.3714	4.6774	-0.52198	-1.2479	-0.88286	-0.26236	
Maahir	1.6173	1.87727	1.991062	2.5912	2.7068	-0.6470	-1.58930	-1.0615	-2.00802	0.63861	<--
Rogelio	0.3709	-0.05055	1.662139	-1.1871	-1.5824	-2.1667	-0.14176	-0.8769	1.05351	3.42098	
Erica	-1.1934	-1.20435	-0.027415	0.9325	3.4823	0.9768	-0.03217	-0.6410	-1.81136	-1.15211	
Barren	-1.5323	-0.60161	-2.533198	-1.2494	-6.0360	-2.4041	2.04445	-1.4898	-1.22238	-1.08176	
Kiana	0.4644	0.88736	1.443450	-0.7497	1.1085	-4.3087	-2.56915	-3.4558	-0.98756	-1.31337	
Elyse	-0.8082	0.48497	-1.957171	2.0216	-3.1847	-4.8516	1.13000	0.6208	1.35247	1.53015	
Chadrick	-0.4654	-0.27757	0.285716	1.2810	-0.8640	0.6105	-0.64552	0.3272	1.20596	2.38268	
Tahani	-1.1716	-1.03925	0.044324	-2.8152	-1.8436	-1.3042	-1.01221	-2.3960	-1.50504	-0.73942	

# Correlations, Envisioned



# Friendly PCA using principal

```
> PCSim1 <- principal(df, nfactors=1,rotate="none")
> PCSim1
Principal Components Analysis
Call: principal(r = df, nfactors = 1, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	PC1	h2	u2	com
Z1	0.83	0.694	0.31	1
Z2	0.67	0.454	0.55	1
Z3	0.89	0.798	0.20	1
Z4	0.17	0.030	0.97	1
Z5	0.69	0.482	0.52	1
Z6	0.29	0.083	0.92	1
X1	-0.79	0.618	0.38	1
X2	-0.61	0.372	0.63	1
X3	-0.44	0.194	0.81	1
X4	-0.31	0.093	0.91	1

```

                PC1
SS loadings      3.82
Proportion Var 0.38

Mean item complexity = 1
Test of the hypothesis that 1 component is sufficient.

The root mean square of the residuals (RMSR) is 0.2
with the empirical chi square 72.9 with prob < 0.00018

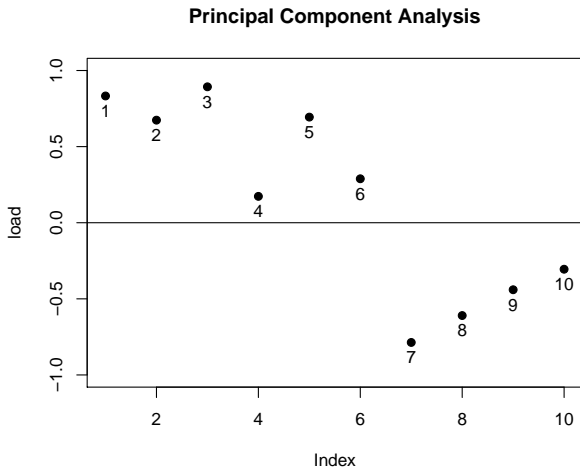
Fit based upon off diagonal values = 0.71
```

## Let's break that down...

- It's a PCA, where we're extracting the first principal component (`nfactors = 1`)
- No rotation (`rotate = "none"`)
- PC1 are the “loadings” of each variable on the first principal component (think of these as the  $w_k$  in the conceptual figure)
- `h2` are *communalities*; the sums of the squared loadings (so, here,  $PC1^2$ )
- `u2` is *uniqueness*; simply  $1 - h2$
- `SS Loadings` is the value(s) of the principal component(s)
- `Proportion Var` is the proportion of the total variance in **X** that that principal component accounts for
- The model fit statistic suggests that the one-component model doesn't fit the data very well (we can reject the hypothesis that one component is sufficient)

# Minimalist PCA Plot

```
> plot(PCSim1,ylim=c(-1,1))
```



# PCA Scores

```
> PCSim1$scores
```

```
          PC1
Guillermo -2.11824 <--
Rachel      0.78544
Deidra      0.63716
Quaton     -0.92891
Alicia     -0.33762
Angelique   1.10640
Johnaton   -0.49378
Javan       0.23606 <--
Khulood     1.06142
Cody        -0.07695
Cameron    -0.02655
Heidi       1.41409
Maahir      1.57304 <--
Rogelio    -0.15324
Erica       -0.11558
Barren     -1.57311
Kiana       1.04688
Elyse      -1.19026
Chadrick   -0.41816
Tahani     -0.42809
```

# PCA with nfactors = 2

```
> PCSim2 <- principal(df, nfactors=2,rotate="none")
> PCSim2
Principal Components Analysis
Call: principal(r = df, nfactors = 2, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	PC1	PC2	h2	u2	com
Z1	0.83	0.44	0.89	0.11	1.5
Z2	0.67	0.45	0.66	0.34	1.8
Z3	0.89	0.13	0.81	0.19	1.0
Z4	0.17	0.71	0.53	0.47	1.1
Z5	0.69	0.11	0.49	0.51	1.1
Z6	0.29	0.24	0.14	0.86	1.9
X1	-0.79	0.32	0.72	0.28	1.3
X2	-0.61	0.52	0.64	0.36	1.9
X3	-0.44	0.68	0.65	0.35	1.7
X4	-0.31	0.63	0.49	0.51	1.4

	PC1	PC2
SS loadings	3.82	2.21
Proportion Var	0.38	0.22
Cumulative Var	0.38	0.60
Proportion Explained	0.63	0.37
Cumulative Proportion	0.63	1.00

```
Mean item complexity = 1.5
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is 0.12
with the empirical chi square 25.8 with prob < 0.47

Fit based upon off diagonal values = 0.9
```

## Let's break that down again...

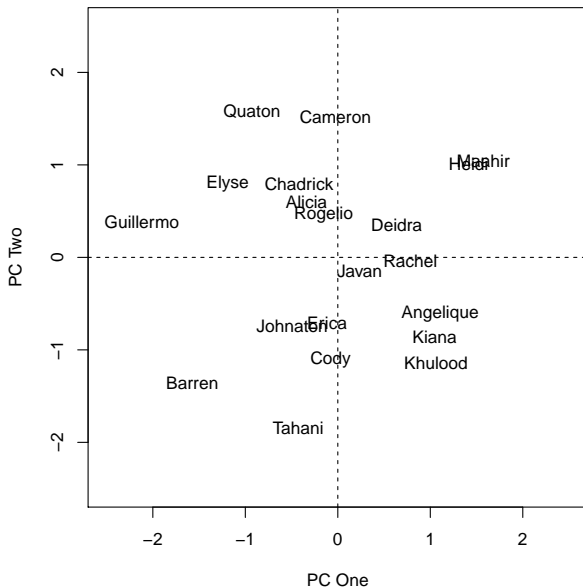
- It's a PCA, where now we're extracting the first two principal components (`nfactors = 2`)
- No rotation (`rotate = "none"`)
- PC1, PC2, h2, and u2 are the same as above
- `com` is the *complexity*  $c_k$  of each measure;  $c_k = \frac{(\sum PC_k^2)^2}{\sum PC_k^4}$
- SS Loadings are again the value(s) of the principal component(s)
- There are now both total and cumulative variance explained statistics
- The model fit statistic now suggests that the model fits well (we cannot reject the hypothesis that two components are sufficient)



# Scores, Redux

```
> PCSim2$scores
      PC1      PC2
Guillermo -2.11824  0.3897 <--
Rachel      0.78544 -0.0387
Deidra      0.63716  0.3501
Quaton     -0.92891  1.5769
Alicia     -0.33762  0.5989
Angelique   1.10640 -0.6102
Johnaton   -0.49378 -0.7382
Javan       0.23606 -0.1504 <--
Khulood     1.06142 -1.1449
Cody        -0.07695 -1.1076
Cameron    -0.02655  1.5239
Heidi       1.41409  1.0112
Maahir      1.57304  1.0410 <--
Rogelio    -0.15324  0.4666
Erica       -0.11558 -0.7050
Barren     -1.57311 -1.3553
Kiana       1.04688 -0.8571
Elyse      -1.19026  0.8022
Chadrick   -0.41816  0.7941
Tahani     -0.42809 -1.8473
```

# PCA Scores



# PCA with nfactors = 3

```
> PCSim3 <- principal(df, nfactors=3,rotate="none")
> PCSim3
Principal Components Analysis
Call: principal(r = df, nfactors = 3, rotate = "none")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	PC1	PC2	PC3	h2	u2	com
Z1	0.83	0.44	-0.15	0.91	0.09	1.6
Z2	0.67	0.45	-0.33	0.77	0.23	2.3
Z3	0.89	0.13	0.19	0.85	0.15	1.1
Z4	0.17	0.71	0.11	0.54	0.46	1.2
Z5	0.69	0.11	0.05	0.50	0.50	1.1
Z6	0.29	0.24	0.88	0.91	0.09	1.4
X1	-0.79	0.32	0.07	0.73	0.27	1.3
X2	-0.61	0.52	0.40	0.80	0.20	2.7
X3	-0.44	0.68	-0.05	0.65	0.35	1.7
X4	-0.31	0.63	-0.48	0.72	0.28	2.4

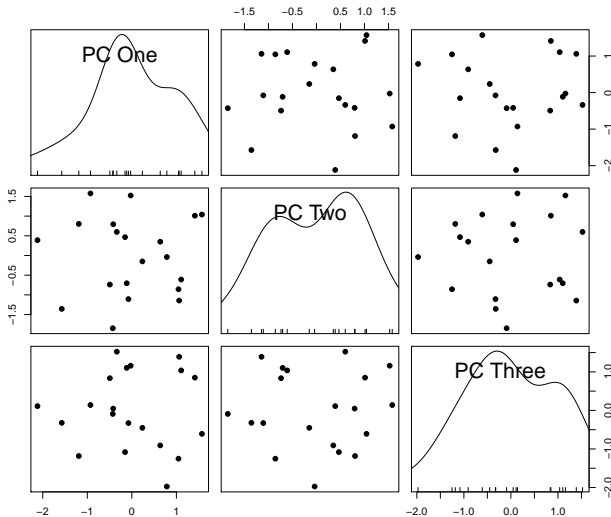
	PC1	PC2	PC3
SS loadings	3.82	2.21	1.35
Proportion Var	0.38	0.22	0.14
Cumulative Var	0.38	0.60	0.74
Proportion Explained	0.52	0.30	0.18
Cumulative Proportion	0.52	0.82	1.00

```
Mean item complexity = 1.7
Test of the hypothesis that 3 components are sufficient.
```

```
The root mean square of the residuals (RMSR) is 0.08
with the empirical chi square 11.12 with prob < 0.89
```

```
Fit based upon off diagonal values = 0.96
```

# PCA Scores, Again



A *biplot* is a graphical representation of a two-axis PCA.

- It plots both loadings (of variables) and scores (of observations)
- It represents the former as vectors from the origin, and the latter as points in the (transformed) space
- Interpretation:
  - Angles between item vectors represent degrees of correlation/covariance
  - Distances between points reflect dissimilarities between those observations
- Details are in Gower and Hand (1996) and Jacoby (1998, Chapter 7)

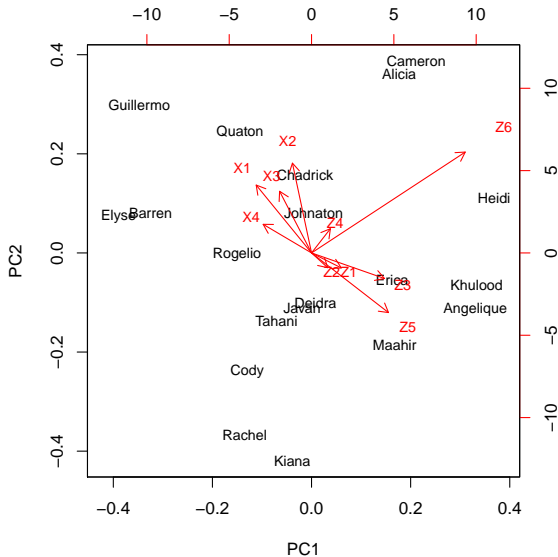
## Biplot Basics (Simulation Data)

```
> foo<-prcomp(df)
```

```
> foo$rotation[,1:2]
```

	PC1	PC2
Z1	0.14329	-0.08291
Z2	0.07769	-0.08154
Z3	0.35128	-0.14139
Z4	0.09075	0.13402
Z5	0.37041	-0.32978
Z6	0.73986	0.55782
X1	-0.26633	0.37528
X2	-0.09196	0.49655
X3	-0.15372	0.34087
X4	-0.23278	0.15817

# A Biplot...



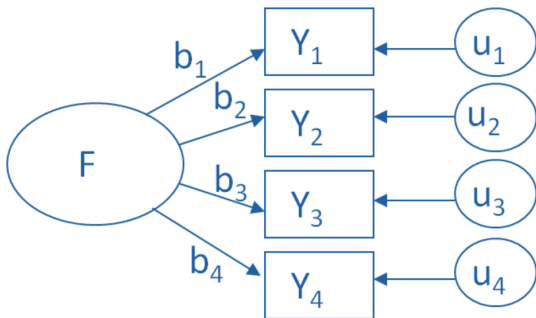
# (Exploratory) Factor Analysis

*Factor analysis* (FA) is a model for the measurement of a latent variable using manifest / observable indicators.

- Observable indicators are manifestations of one or more latent / unobservable *factors*
- Extant indicators are differentially caused by the latent factor(s), and are observed with error
- The goal of FA is to derive measures of the latent factor from the observed data, by estimating factor *loadings* (associations between latent factors and observable variables)



# Factor Analysis, Conceptually



$$Y_1 = b_1 F + u_1$$

$$Y_2 = b_2 F + u_2$$

$$Y_3 = b_3 F + u_3$$

$$Y_4 = b_4 F + u_4$$

(Source)

Formally:

$$\mathbf{Y} = \mathbf{\Lambda F} + \mathbf{U}$$

This implies that the observed covariance matrix  $\Sigma$  can be written:

$$\Sigma = \mathbf{\Lambda \Lambda'} + \Psi$$

where

$$\Psi = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_K^2 \end{bmatrix}$$

- Choose the *number of factors* (dimensions)
- Consider *rotation*
- *Estimate* the factor loadings  $\hat{\Lambda}$
- *Interpret* the factors...
- Generate *factor scores*

# Factor Analysis Simulation

```
> FASim1 <- factanal(df,factors=1,scores="regression",rotation="none")  
> print(FASim1,cutoff=0)
```

Call:

```
factanal(x = df, factors = 1, scores = "regression", rotation = "none")
```

Uniquenesses:

	Z1	Z2	Z3	Z4	Z5	Z6	X1	X2	X3	X4
	0.005	0.290	0.365	0.912	0.667	0.942	0.778	0.896	0.999	0.995

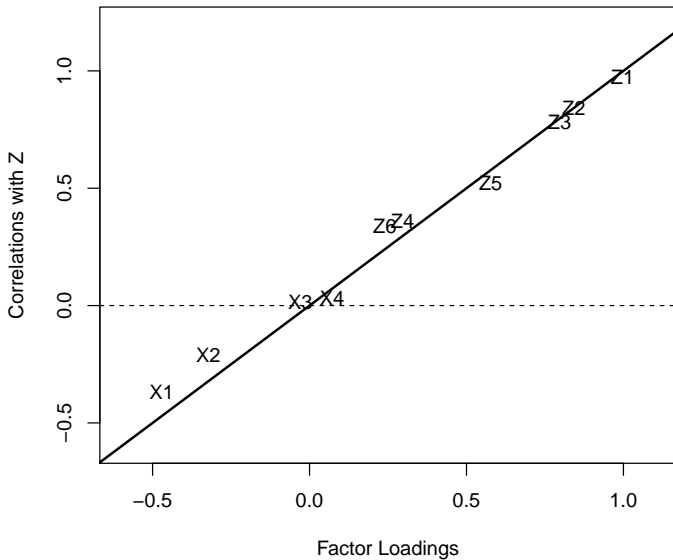
Loadings:

	Factor1
Z1	0.998
Z2	0.843
Z3	0.797
Z4	0.297
Z5	0.577
Z6	0.240
X1	-0.471
X2	-0.322
X3	-0.029
X4	0.072

	Factor1
SS loadings	3.150
Proportion Var	0.315

Test of the hypothesis that 1 factor is sufficient.  
The chi square statistic is 57.3 on 35 degrees of freedom.  
The p-value is 0.0101

# Factor Loadings vs. Correlations with $Z$



# Factor Analysis Simulation: Two Factors

```
> FASim2 <- factanal(df,factors=2,scores="regression",rotation="none")
> print(FASim2,cutoff=0)
```

Call:

```
factanal(x = df, factors = 2, scores = "regression", rotation = "none")
```

Uniquenesses:

	Z1	Z2	Z3	Z4	Z5	Z6	X1	X2	X3	X4
	0.005	0.276	0.226	0.810	0.608	0.938	0.271	0.525	0.444	0.667

Loadings:

	Factor1	Factor2
Z1	0.997	0.013
Z2	0.841	0.129
Z3	0.801	-0.364
Z4	0.295	0.320
Z5	0.579	-0.238
Z6	0.242	-0.060
X1	-0.478	0.707
X2	-0.327	0.607
X3	-0.034	0.745
X4	0.068	0.573

	Factor1	Factor2
SS loadings	3.166	2.064
Proportion Var	0.317	0.206
Cumulative Var	0.317	0.523

Test of the hypothesis that 2 factors are sufficient.  
The chi square statistic is 31.42 on 26 degrees of freedom.  
The p-value is 0.213

# Factor Analysis Simulation: Three Factors

```
> FASim3 <- factanal(df,factors=3,scores="regression",rotation="none")  
> print(FASim3,cutoff=0)
```

Call:

```
factanal(x = df, factors = 3, scores = "regression", rotation = "none")
```

Uniquenesses:

	Z1	Z2	Z3	Z4	Z5	Z6	X1	X2	X3	X4
	0.005	0.246	0.111	0.758	0.621	0.005	0.312	0.276	0.518	0.582

Loadings:

	Factor1	Factor2	Factor3
Z1	0.809	0.021	0.583
Z2	0.582	0.136	0.629
Z3	0.836	-0.379	0.216
Z4	0.334	0.359	0.036
Z5	0.475	-0.207	0.333
Z6	0.761	0.002	-0.645
X1	-0.382	0.674	-0.297
X2	-0.071	0.704	-0.473
X3	-0.024	0.693	-0.032
X4	-0.125	0.567	0.285

	Factor1	Factor2	Factor3
SS loadings	2.775	2.086	1.706
Proportion Var	0.277	0.209	0.171
Cumulative Var	0.277	0.486	0.657

Test of the hypothesis that 3 factors are sufficient.

The chi square statistic is 15.13 on 18 degrees of freedom.

The p-value is 0.653

# Real Data: ANES 2016 Feeling Thermometers

```
> describe(Therms,range=FALSE)
```

	vars	n	mean	sd	skew	kurtosis	se
Asian-Americans	1	2387	70.17	20.20	-0.38	0.02	0.41
Hispanics	2	2387	69.35	20.91	-0.41	0.01	0.43
Blacks	3	2387	69.00	21.19	-0.35	-0.24	0.43
Illegal Immigrants	4	2387	42.54	27.31	0.13	-0.71	0.56
Whites	5	2387	71.63	19.40	-0.46	0.08	0.40
Dem. Pres. Candidate	6	2387	44.12	34.91	0.12	-1.42	0.71
GOP Pres. Candidate	7	2387	40.53	35.65	0.23	-1.43	0.73
Libertarian Pres. Candidate	8	2387	43.61	19.92	-0.58	0.25	0.41
Green Pres. Candidate	9	2387	43.20	20.87	-0.54	0.22	0.43
Dem. VP	10	2387	48.24	25.91	-0.22	-0.44	0.53
GOP VP	11	2387	49.59	33.42	-0.10	-1.21	0.68
John Roberts	12	2387	53.75	18.39	-0.41	1.44	0.38
Pope Francis	13	2387	69.55	25.17	-0.73	0.14	0.52
Christian Fundamentalists	14	2387	48.59	28.48	-0.07	-0.72	0.58
Feminists	15	2387	56.94	26.65	-0.24	-0.47	0.55
Liberals	16	2387	52.27	27.35	-0.24	-0.67	0.56
Labor Unions	17	2387	56.70	24.74	-0.27	-0.29	0.51
Poor People	18	2387	72.20	19.63	-0.36	-0.06	0.40
Big Business	19	2387	49.34	22.52	-0.15	-0.18	0.46
Conservatives	20	2387	55.22	25.91	-0.24	-0.45	0.53
SCOTUS	21	2387	59.34	19.38	-0.32	0.54	0.40
Gays & Lesbians	22	2387	62.83	26.86	-0.46	-0.20	0.55
Congress	23	2387	41.17	22.32	0.02	-0.34	0.46
Rich People	24	2387	53.53	20.69	-0.13	0.52	0.42
Muslims	25	2387	55.80	25.64	-0.29	-0.23	0.52
Christians	26	2387	74.40	23.80	-0.87	0.35	0.49
Jews	27	2387	72.20	21.19	-0.45	-0.14	0.43
Tea Party	28	2387	42.97	27.08	-0.06	-0.70	0.55
Police	29	2387	75.57	22.50	-1.15	1.13	0.46
Transgender People	30	2387	57.29	26.88	-0.28	-0.31	0.55
Scientists	31	2387	77.74	19.23	-0.77	0.39	0.39
BLM	32	2387	48.26	32.66	-0.06	-1.15	0.67



# Factor Analysis: One Factor

```
> FTFA1 <- fa(Therms,nfactors=1,fm="ml",rotate="none")
> print(FTFA1)
Factor Analysis using method = ml
Call: fa(r = Therms, nfactors = 1, rotate = "none", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	ML1	h2	u2	com
Asian-Americans	0.29	0.08306	0.92	1
Hispanics	0.37	0.13456	0.87	1
Blacks	0.39	0.15227	0.85	1
Illegal Immigrants	0.61	0.37552	0.62	1
Whites	-0.03	0.00066	1.00	1
Dem. Pres. Candidate	0.79	0.62770	0.37	1
GOP Pres. Candidate	-0.81	0.65791	0.34	1
Libertarian Pres. Candidate	-0.07	0.00476	1.00	1
Green Pres. Candidate	0.22	0.05026	0.95	1
Dem. VP	0.65	0.42135	0.58	1
GOP VP	-0.80	0.64779	0.35	1
John Roberts	-0.24	0.05942	0.94	1
Pope Francis	0.27	0.07253	0.93	1
Christian Fundamentalists	-0.49	0.23650	0.76	1
Feminists	0.69	0.47926	0.52	1
Liberals	0.80	0.63513	0.36	1
Labor Unions	0.49	0.24414	0.76	1
Poor People	0.25	0.06198	0.94	1
Big Business	-0.31	0.09877	0.90	1
Conservatives	-0.65	0.42099	0.58	1
SCOTUS	0.11	0.01287	0.99	1
Gays & Lesbians	0.62	0.38096	0.62	1
Congress	-0.20	0.04024	0.96	1
Rich People	-0.18	0.03379	0.97	1
Muslims	0.63	0.39894	0.60	1
Christians	-0.32	0.10381	0.90	1
Jews	0.23	0.05481	0.95	1
Tea Party	-0.62	0.38321	0.62	1
Police	-0.31	0.09796	0.90	1
Transgender People	0.65	0.42375	0.58	1
Scientists	0.39	0.15438	0.85	1
BLM	0.73	0.53747	0.46	1
.				
.				
.				

# Factor Analysis: One Factor

(...continued)

ML1  
SS loadings 8.09  
Proportion Var 0.25

Mean item complexity = 1  
Test of the hypothesis that 1 factor is sufficient.

The degrees of freedom for the null model are 496 and the objective function was 16.99 with  
Chi Square of 40352  
The degrees of freedom for the model are 464 and the objective function was 8.87

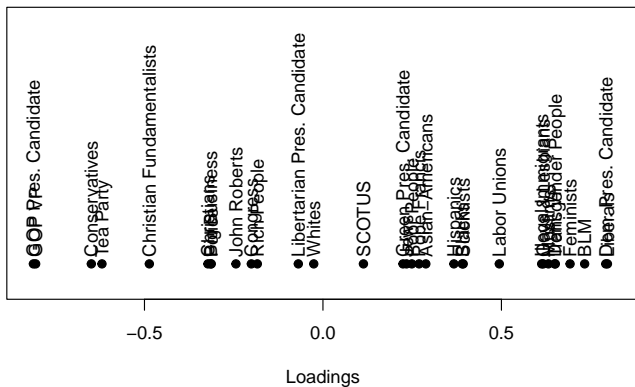
The root mean square of the residuals (RMSR) is 0.15  
The df corrected root mean square of the residuals is 0.16

The harmonic number of observations is 2387 with the empirical chi square 53448 with prob < 0  
The total number of observations was 2387 with MLE Chi Square = 21052 with prob < 0

Tucker Lewis Index of factoring reliability = 0.448  
RMSEA index = 0.137 and the 90 % confidence intervals are 0.135 0.138  
BIC = 17443  
Fit based upon off diagonal values = 0.74  
Measures of factor score adequacy

	ML1
Correlation of scores with factors	0.97
Multiple R square of scores with factors	0.94
Minimum correlation of possible factor scores	0.88

# Factor Loadings



# Factor Analysis: Two Factors

```
> FTFA2 <- fa(Therms,nfactors=2,fm="ml", rotate="none")
> print(FTFA2)
Factor Analysis using method = ml
Call: fa(r = Therms, nfactors = 2, rotate = "none", fm = "ml")
Standardized loadings (pattern matrix) based upon correlation matrix
```

	ML1	ML2	h2	u2	com
Asian-Americans	0.29	0.54	0.375	0.63	1.5
Hispanics	0.37	0.52	0.404	0.60	1.8
Blacks	0.39	0.51	0.406	0.59	1.9
Illegal Immigrants	0.61	0.20	0.408	0.59	1.2
Whites	-0.04	0.52	0.273	0.73	1.0
Dem. Pres. Candidate	0.78	0.01	0.604	0.40	1.0
GOP Pres. Candidate	-0.82	0.17	0.703	0.30	1.1
Libertarian Pres. Candidate	-0.09	0.13	0.024	0.98	1.7
Green Pres. Candidate	0.21	0.11	0.054	0.95	1.5
Dem. VP	0.63	0.06	0.402	0.60	1.0
GOP VP	-0.83	0.26	0.753	0.25	1.2
.					
.					
.					
Police	-0.33	0.43	0.293	0.71	1.9
Transgender People	0.65	0.28	0.500	0.50	1.4
Scientists	0.39	0.32	0.253	0.75	1.9
BLM	0.72	0.15	0.535	0.46	1.1

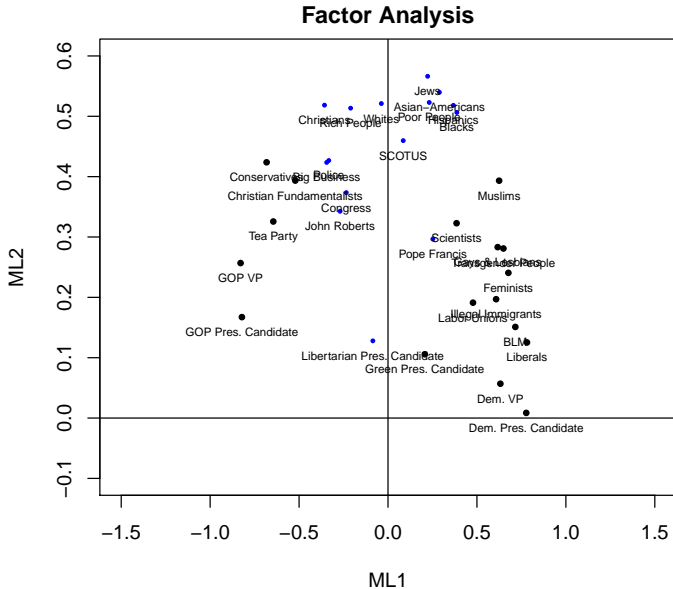
  

	ML1	ML2
SS loadings	8.16	4.29
Proportion Var	0.26	0.13
Cumulative Var	0.26	0.39
Proportion Explained	0.66	0.34
Cumulative Proportion	0.66	1.00

```
Mean item complexity = 1.5
.
.
.
```

# Factor Analysis: Two Factors



PCA / FA are *data reduction* techniques...

- Rotation is exactly that: Rotation of the axes in the transformed space to make the results more interpretable.
- Two broad types:
  - *Orthogonal* rotation (maintains orthogonality of the axes)
  - *Oblique* rotation (allows components / factors to be correlated)
- **The goal of rotation is to improve the interpretability of the PCA/FA results (that is, to reveal “simple structure”)**

## Orthogonal rotations:

- **Varimax** (minimizes the number of variables that have high loadings on each factor.)
- **Quartimax** (minimizes the number of factors needed to explain each variable)
- **Equamax** (a combination of varimax and quartimax)
- Others...

## Oblique rotations (less easily interpretable):

- **Direct Oblimin** (the de facto standard for oblique rotation)
- **Promax** (simpler / faster than oblimin)
- Others...

“Simple structure”: “A condition in which variables load at near 1 (in absolute value) or at near 0 on an eigenvector (factor).

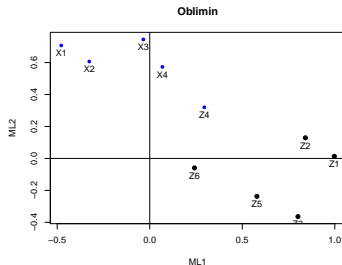
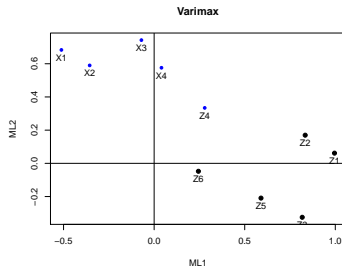
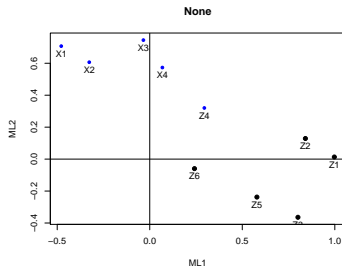
Variables that load near 1 are clearly important in the interpretation of the factor, and variables that load near 0 are clearly unimportant.” (Bryant and Yarnold 1995)

Factor Loading  $\ell$  Guidelines:

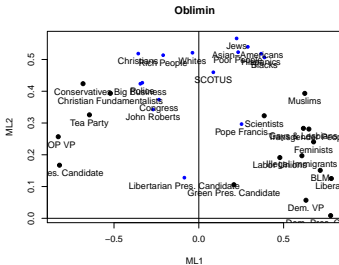
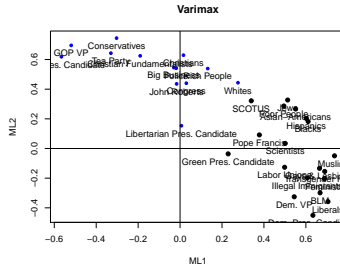
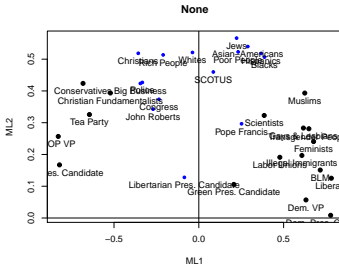
- $0.10 < \ell < -0.10$  are unimportant
- $|\ell| > 0.30$  are important with  $N \geq 100$
- Variables with  $\ell > 0.30$  on more than one factor are *complex*



# Rotation: Simulated Data



## Rotation: Feeling Thermometers



PCA/FA are *data reduction* techniques...

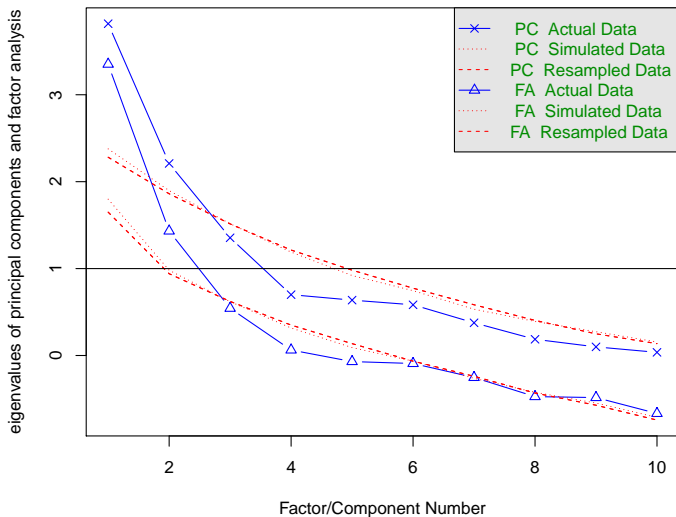
- The sum of the eigenvalues equals  $K$ ; so...
- A factor / component with an eigenvalue less than 1.0 isn't even "explaining itself"
- "Kaiser criterion"

Other approaches:

- *Theory...*
- "Scree plot" (look for the "elbow")
- Target variance explained
- Others...

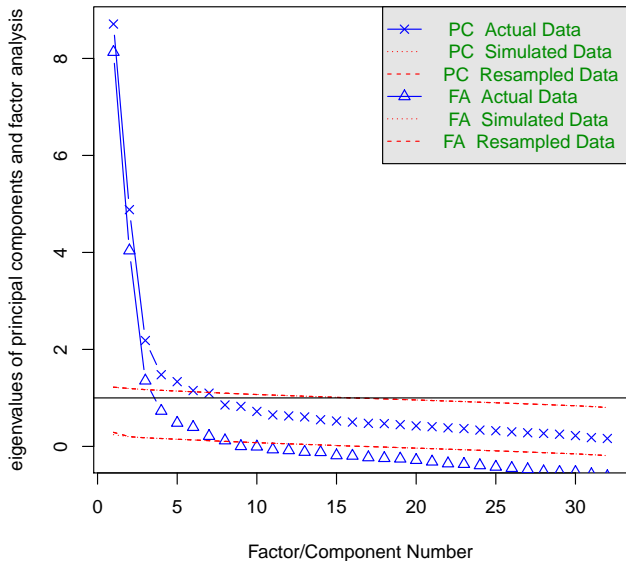
# “Parallel” Scree Plot (Simulated Data)

Parallel Analysis Scree Plots



# “Parallel” Scree Plot (Feeling Thermometer Data)

Parallel Analysis Scree Plots



# Useful References

- Gorsuch, Richard L. 1983. *Factor Analysis*, 2nd Ed. NJ: Lawrence Erlbaum.
- Cudek, Robert and Robert C. MacCallum, Eds. 2007. *Factor Analysis at 100*. NJ: Lawrence Erlbaum.
- Mulaik, Stanley A. 2010. *Foundations of Factor Analysis*, 2nd Ed. Boca Raton, FL: CRC Press.
- Fabrigar, Leandre R., and Duane T. Wegener. 2014. *Exploratory Factor Analysis*. New York: Oxford University Press.

# Useful R Packages and Routines

## PCA and Biplots

- `stats::prcomp` (principal components via SVD)
- `biplot` (biplots)
- `psych::principal` (User-friendly PCA routine)
- Others...

## Factor Analysis

- `nFactors` (Routines for assessing dimensionality / number of factors)
- `FactoMineR` (Hugely expanded FA package...)
- `GPARotation` (Many, many rotation options)

# Cluster Analysis



“...a **statistical operation of grouping objects**. The resulting groups are clusters. Clusters have the following properties:

- We find them during the operation and their number is also not always fixed in advance.
- They are the combination of objects having similar characteristics.”

“...**groups objects (observations, events) based on the information found in the data** describing the objects or their relationships. The goal is that the objects in a group will be similar (or related) to one other and different from (or unrelated to) the objects in other groups. The greater the similarity (or homogeneity) within a group, and the greater the difference between groups, the ‘better’ or more distinct the clustering.”

- Classification / Taxonomy (*description*)
- Data Reduction (*measurement*)
- Identify Relationships (*inductive inference*)
- Prediction (typically out-of-sample)

# Clustering: Intuition



Figure 1a: Initial points.



Figure 1b: Two clusters.

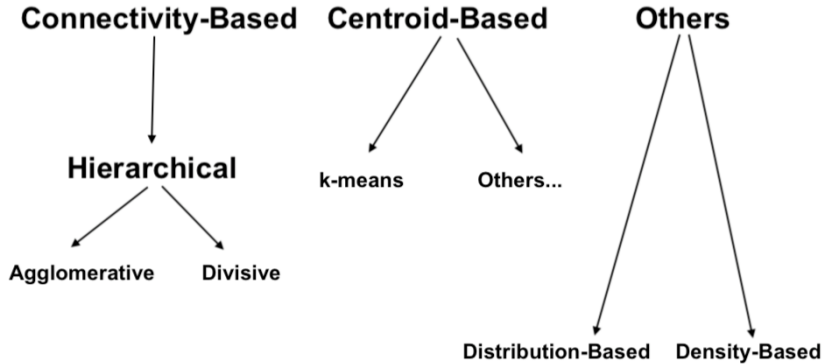


Figure 1c: Six clusters



Figure 1d: Four clusters.

# Cluster Analysis: Typology



Euclidean ( “L2” ) Distance:

$$d_{L2}(\mathbf{X}, \mathbf{Y}) = \sqrt{\sum_{k=1}^K (X_k - Y_k)^2}.$$

“City-Block” / Manhattan ( “L1” ) Distance:

$$d_{L1}(\mathbf{X}, \mathbf{Y}) \equiv \|\mathbf{X} - \mathbf{Y}\|_1 = \sum_{k=1}^K |X_k - Y_k|.$$

Mahalanobis Distance:

$$d_M(\mathbf{X}, \mathbf{Y}) = \sqrt{(\mathbf{X} - \mathbf{Y})' \mathbf{S}^{-1} (\mathbf{X} - \mathbf{Y})}.$$

# Distance Example

Data ( $N = 2$ ):

	X	Y	Z
Tick	1	711	0.08
Arthur	0	588	0.27
Tick - Arthur	1	123	-0.19

Euclidean:

$$\begin{aligned}D_{L2} &= \sqrt{(1 - 0)^2 + (711 - 588)^2 + (0.08 - 0.27)^2} \\&= \sqrt{1 + 15129 + 0.0361} \\&= 123.004\end{aligned}$$

Manhattan:

$$\begin{aligned}D_{L1} &= |1 - 0| + |711 - 588| + |0.08 - 0.27| \\&= 1 + 123 + 0.19 \\&= 124.19\end{aligned}$$

Mahalanobis:

$$\begin{aligned}D_M &= \sqrt{(\text{Tick} - \text{Arthur})' \hat{\mathbf{S}}^{-1} (\text{Tick} - \text{Arthur})} \\&= 1.386\end{aligned}$$

**Lesson: Standardize variables!**

# Defining Intra-Cluster Distances

For two clusters  $C_A$  and  $C_B$ , the distance between can be defined in terms of:

- Single-linkage

$$d_{AB} = \min(d_{a,b})$$

- Complete linkage

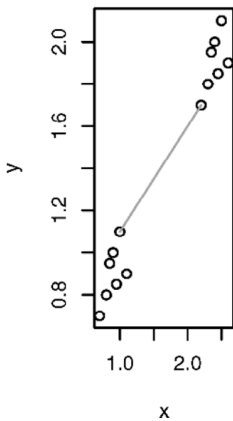
$$d_{AB} = \max(d_{a,b})$$

- Group average

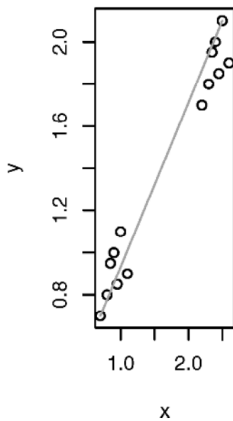
$$d_{AB} = \frac{1}{N_A N_B} \sum_{a=1}^{N_A} \sum_{b=1}^{N_B} (d_{a,b})$$

# Cluster Linkages

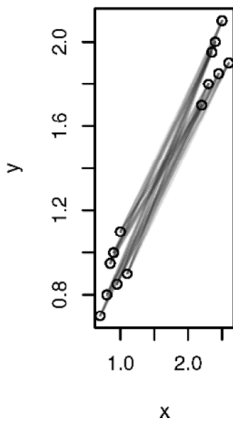
**single**



**complete**



**average**





# Agglomerative Clustering

Basic steps:

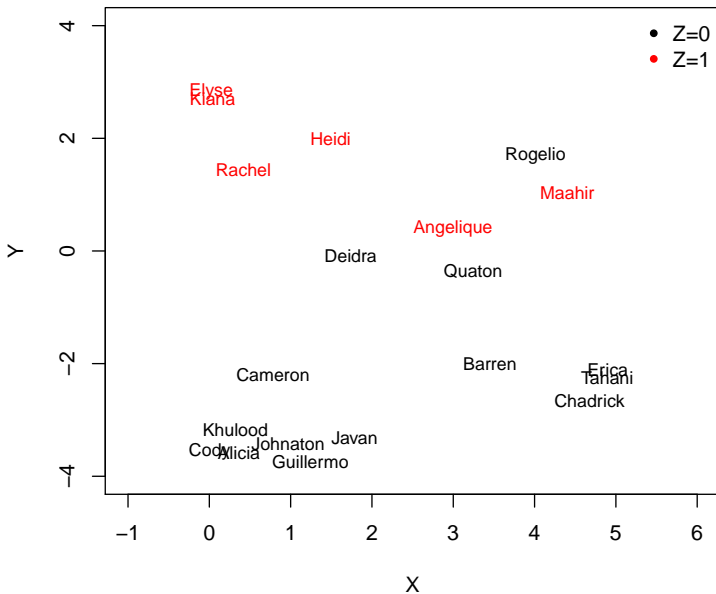
1. Begin with  $N$  observations on  $K$  variables in  $\mathbf{X}$
2. Define each observation as its own “cluster”  $C_i$
3. Find the two clusters  $C_\ell$  and  $C_m$  that are “closest” to each other
4. Merge them into a single cluster, and delete the two component clusters
5. Recalculate the distances between all remaining clusters
6. Repeat steps 3-5 until only one cluster remains

# Simulation Example

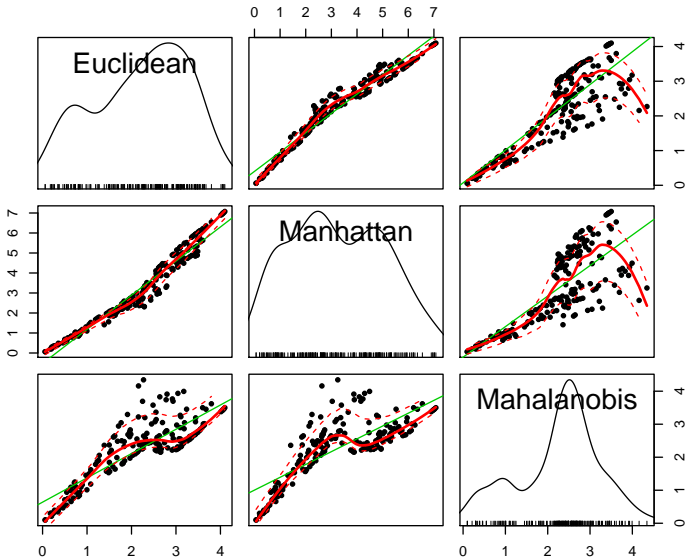
```
> N <- 20
> set.seed(7222009)
> Name <- randomNames(N, which.names="first")
> X <- 5*rbeta(N,0.5,0.5)
> Y <- runif(N,-4,4)
> Z <- rbinom(N,1,pnorm(Y/2))

> df <- data.frame(Name=Name,X=X,Y=Y,Z=Z)
> rownames(df)<-df$Name
>
> # Distances:
> #
> # CENTER AND RESCALE / STANDARDIZE THE DATA:
>
> ds <- scale(df[,2:4])
>
> DL2 <- dist(ds) # L2 / Euclidean distance
> DL1 <- dist(ds,method="manhattan") # L1 / Manhattan distance
> DM <- sqrt(D2.dist(ds,cov(ds))) # Mahalanobis distances
```

# Simulated Data, Plotted



# Distance Comparisons

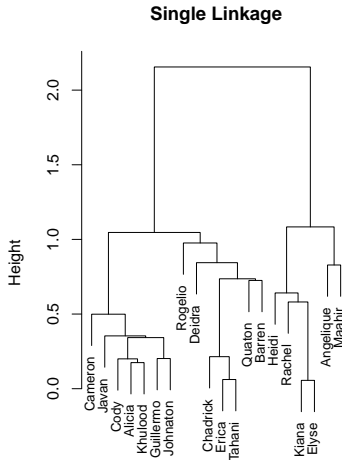
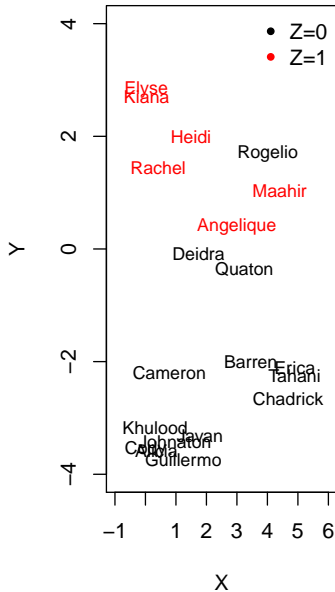


# Using hclust (in cluster)

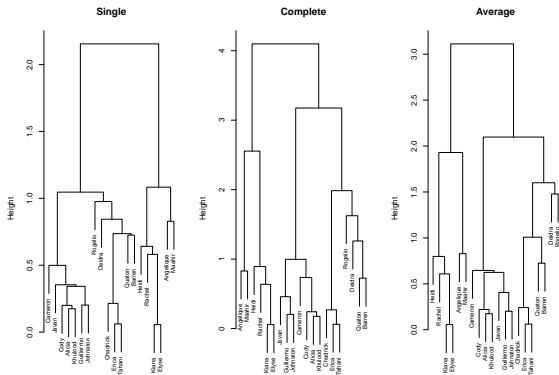
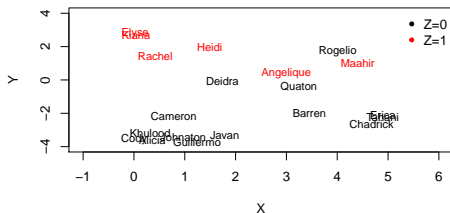
```
> ADL2.s <- hclust(DL2,method="single")
> ADL2.c <- hclust(DL2,method="complete")
> ADL2.a <- hclust(DL2,method="average")

> str(ADL2.s)
List of 7
 $ merge      : int [1:19, 1:2] -17 -15 -5 -9 -1 -19 4 -8 -11 -2 ...
 $ height     : num [1:19] 0.129 0.143 0.36 0.405 0.413 ...
 $ order      : int [1:20] 17 18 2 12 11 8 9 5 10 1 ...
 $ labels     : chr [1:20] "Guillermo" "Rachel" "Deidra" "Quaton" ...
 $ method     : chr "single"
 $ call       : language hclust(d = DL2, method = "single")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

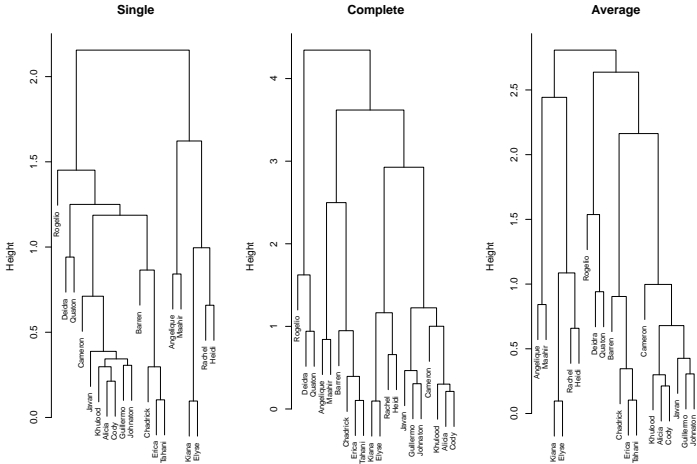
# The Dendrogram



# Comparing Linkages



# Using Mahalanobis Distance





# The Agglomeration Coefficient

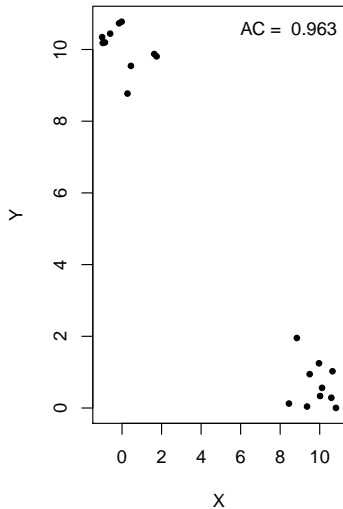
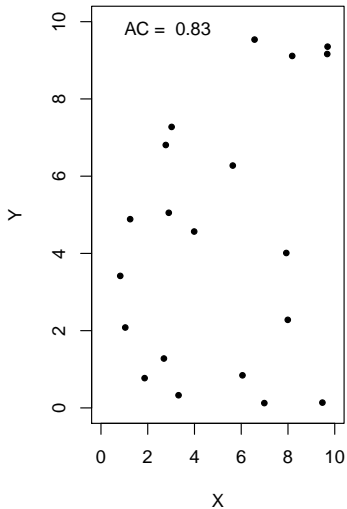
The *agglomeration coefficient*  $AC$  measures the clustering structure of the data. For each observation  $i$ , define  $m_i$  as the dissimilarity of observation  $i$  with the first cluster with which it is merged, divided by the dissimilarity in the final iteration (i.e., the greatest dissimilarity). The coefficient is then:

$$AC = \frac{1}{N-1} \sum_{i=1}^{N-1} 1 - m_i$$

Notes:

- Higher values correspond to greater clustering in the data.
- $AC$  increases with  $N$  so should not be used to compare datasets of very different sizes

# Example AC Values



## Example ACs: Simulated Data

```
> Agnes.s <- agnes(ds, metric="euclidean",method="single")
```

```
> Agnes.s$ac
```

```
[1] 0.805
```

```
> Agnes.c <- agnes(ds, metric="euclidean",method="complete")
```

```
> Agnes.c$ac
```

```
[1] 0.8754
```

```
> Agnes.a <- agnes(ds, metric="euclidean",method="average")
```

```
> Agnes.a$ac
```

```
[1] 0.8398
```

```
> # Using Mahalanobis distance:
```

```
> Agnes.M <- agnes(DM, diss=TRUE, method="average")
```

```
> Agnes.M$ac
```

```
[1] 0.8071
```

- Can calculate  $P$ -values for each cluster (at each agglomeration stage) via multiscale bootstrap resampling
- Reference: Suzuki, R., and H. Shimodaira. 2006. “pvclust: An R package for assessing the uncertainty in hierarchical clustering.” *Bioinformatics* 22:1540-1542.
- The R package is `pvclust`
- Reports “approximately unbiased” and “bootstrap probability”  $P$ -values (use the former)
- “Clusters with high values... are strongly supported by the data.”

```
dst<-data.frame(t(ds))
PVDL2.s <- pvclust(dst,method.hclust="single",
                  method.dist="euclidean",nboot=1001)
> PVDL2.s
```

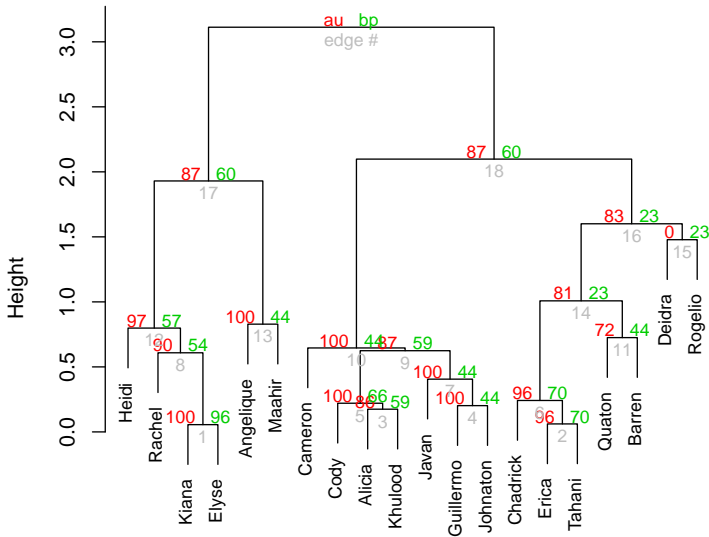
```
Cluster method: average
Distance       : euclidean
```

Estimates on edges:

	au	bp	se.au	se.bp	v	c	pchi
1	0.997	0.957	0.001	0.003	-2.222	0.501	0.607
2	0.963	0.695	0.005	0.006	-1.147	0.636	0.022
3	0.856	0.593	0.013	0.006	-0.648	0.413	0.000
4	0.999	0.445	0.000	0.006	-1.482	1.621	0.105
5	0.997	0.656	0.001	0.006	-1.599	1.198	0.002
6	0.963	0.695	0.005	0.006	-1.147	0.636	0.022
7	0.999	0.445	0.000	0.006	-1.482	1.621	0.105
8	0.902	0.543	0.020	0.006	-0.701	0.592	0.000
9	0.869	0.594	0.012	0.006	-0.681	0.442	0.000
10	0.999	0.445	0.000	0.006	-1.482	1.621	0.105
11	0.721	0.445	0.019	0.006	-0.223	0.362	0.000
12	0.970	0.569	0.008	0.006	-1.028	0.853	0.065
13	0.999	0.439	0.001	0.006	-1.434	1.589	0.095
14	0.807	0.233	0.091	0.007	-0.069	0.797	0.607
15	0.002	0.233	0.002	0.007	1.837	-1.109	0.607
16	0.834	0.233	0.082	0.007	-0.121	0.849	0.607
17	0.866	0.601	0.012	0.006	-0.682	0.427	0.000
18	0.866	0.601	0.012	0.006	-0.682	0.427	0.000
19	1.000	1.000	0.000	0.000	0.000	0.000	0.000

# Dendrogram with P-Values...

## Euclidean/Single Linkage



# Practical Agglomerative Clustering: Linkages

*"The performances of traditional hierarchical clustering methods have been evaluated for a variety of simulated situations. **Single linkage clustering is simple to understand and compute, but has the tendency to build unphysical elongated chains of clusters joined by a single point, especially when unclustered noise is present.** Figure 12.4 of Izenman (2008) illustrates how a single linkage dendrogram can differ considerably from the average linkage, complete linkage and divisive dendrograms, which can be quite similar to each other. Kaufman and Rosseeuw (1990, Section 5.2) report that "Virtually all authors agreed that single linkage was least successful in their [simulation] studies." Everitt et al. (2001, Section 4.2) report that "Single linkage, which has satisfactory mathematical properties and is also easy to program and apply to large data sets, tends to be less satisfactory than other methods because of 'chaining'." Ward's method is successful with clusters of similar populations, but tends to misclassify objects when the clusters are elongated or have very different diameters. **Average linkage is generally found to be an effective technique in simulations, although its results depend on the cluster size.** Average linkage also has better consistency properties than single or complete linkage as the sample size increases towards infinity (Hastie et al. 2009, Section 14.3)."*

– Eric D. Feigelson and G. Jogesh Babu. 2012. *Modern Statistical Methods for Astronomy: With R Applications*. New York: Cambridge University Press, p. 228.

# Divisive Clustering (diana)

Basic steps:

1. Begin with  $N$  observations on  $K$  variables in  $\mathbf{X}$
2. Select the cluster  $C_{maxD}$  with the largest *diameter* (defined as the cluster with the largest dissimilarity between any two of its observations)
3. Select the observation  $j$  in  $C_{maxD}$  that has the highest average dissimilarity to the other observations in the cluster); this is the “seed” of the “splinter group”  $C_{splinter}$
4. Iteratively assign observations to either the splinter group  $C_{splinter}$  or the parent cluster  $C_{parent}$ , based on their dissimilarity to each.
5. Repeat step 4 until each observation in  $C_{maxD}$  is reassigned to either  $C_{parent}$  or  $C_{splinter}$
6. Iterate steps 2-5 until each observation is its own cluster



# Divisive Clustering Example

```
> Diana.L2 <- diana(ds,metric="euclidean")
```

```
> Diana.L2
```

```
Merge:
```

```
      [,1] [,2]
[1,]  -17  -18
[2,]  -15  -20
[3,]   -5   -9
[4,]   -1   -7
[5,]    3  -10
[6,]    2  -19
[7,]    4   -8
[8,]   -2    1
[9,]    5  -11
[10,]   6  -16
[11,]  -6  -13
[12,]  -3   -4
[13,]    8  -12
[14,]    7    9
[15,]   12  -14
[16,]   15   10
[17,]   13   11
[18,]   14   16
[19,]   18   17
```

```
Order of objects:
```

```
 [1] Guillermo Johnaton  Javan    Alicia   Khulood   Cody
 [7] Cameron  Deidra      Quaton   Rogelio   Erica    Tahani
[13] Chadrick Barren     Rachel   Kiana     Elyse    Heidi
[19] Angelique Maahir
```

```
Height:
```

```
 [1] 0.20204 0.45777 0.99653 0.17474 0.24121 0.73438 3.17509 0.84410
 [9] 1.47820 1.98490 0.06146 0.26884 0.80881 4.09856 0.63594 0.05589
[17] 0.89190 2.55486 0.82867
```

```
Divisive coefficient:
```

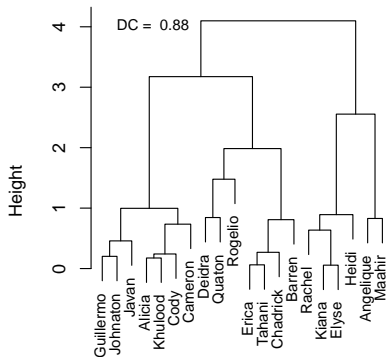
```
[1] 0.8798
```

```
Available components:
```

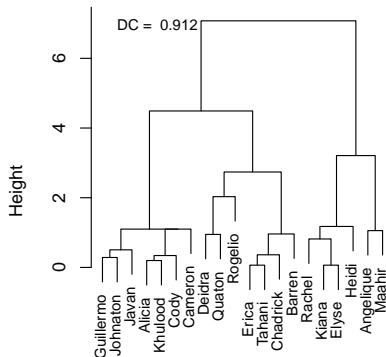
```
[1] "order"      "height"     "dc"         "merge"      "diss"
[6] "call"       "order.lab"  "data"
```

# Divisive Clustering: Dendrograms

**Euclidean Distance**



**Manhattan Distance**



# Non-Hierarchical Clustering: $k$ -Means

$k$ -means clustering “aims to partition the points into  $k$  groups such that the sum of squares from points to the assigned cluster centers is minimized.”

- Formally, find:

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg \min_{\mathbf{S}} \sum_{i=1}^k |S_i| \text{Var } S_i$$

for the set of  $k$  clusters  $S_1 \dots S_k$  in  $\mathbf{S}$ .

- Requires the analyst to designate the number of clusters desired  $k$  *a priori*.
- Standard algorithm:
  0. Initialize a set of  $k$  clusters.
  1. Assign each observation to the cluster whose mean is the least “distant” from it
  2. Calculate the new means as the centroids of the resulting clusters
  3. Repeat steps 1-2 until convergence.

# k-means Clustering: Example ( $k = 2$ )

```
> KM2 <- kmeans(ds,2)
> KM2
K-means clustering with 2 clusters of sizes 7, 13
```

Cluster means:

	X	Y	Z
1	-0.7265	-0.9753	-0.6381
2	0.3912	0.5252	0.3436

Clustering vector:

Guillermo	Rachel	Deidra	Quaton	Alicia	Angelique	Johnaton
1	2	2	2	1	2	1
Javan	Khulood	Cody	Cameron	Heidi	Maahir	Rogelio
1	1	1	1	2	2	2
Erica	Barren	Kiana	Elyse	Chadrick	Tahani	
2	2	2	2	2	2	

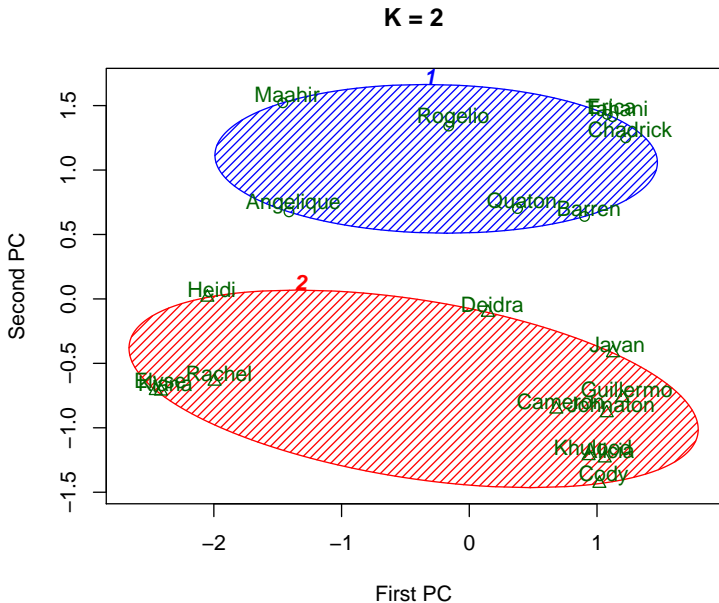
Within cluster sum of squares by cluster:

```
[1] 0.9928 35.6954
(between_SS / total_SS = 35.6 %)
```

Available components:

```
[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

# K-Means Clusters vs. Principal Components ( $k = 2$ )



# k-means Clustering: Example ( $k = 3$ )

```
> KM3 <- kmeans(ds,3)
```

```
> KM3
```

```
K-means clustering with 3 clusters of sizes 7, 7, 6
```

```
Cluster means:
```

	X	Y	Z
1	-0.7265	-0.97528	-0.6381
2	0.9769	-0.03947	-0.6381
3	-0.2921	1.18387	1.4888

```
Clustering vector:
```

Guillermo	Rachel	Deidra	Quaton	Alicia	Angelique	Johnaton
1	3	2	2	1	3	1
Javan	Khulood	Cody	Cameron	Heidi	Maahir	Rogelio
1	1	1	1	3	3	2
Erica	Barren	Kiana	Elyse	Chadrick	Tahani	
2	2	3	3	2	2	

```
Within cluster sum of squares by cluster:
```

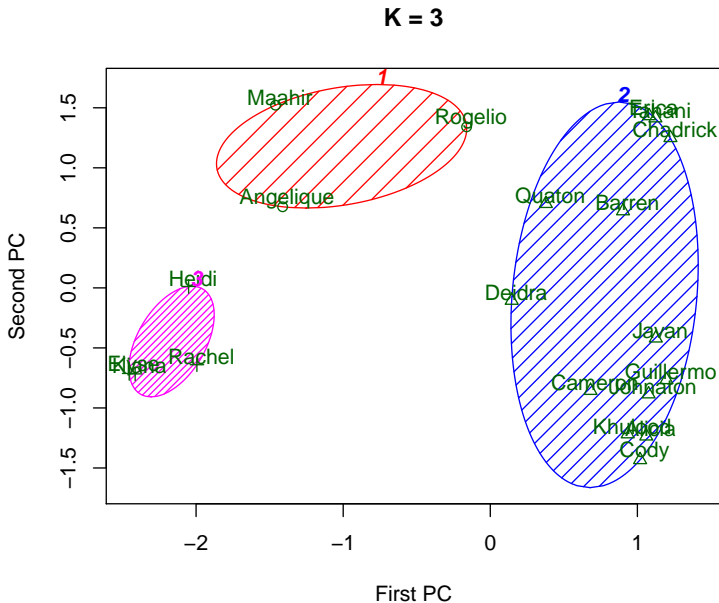
```
[1] 0.9928 5.2115 5.8304
```

```
(between_SS / total_SS = 78.9 %)
```

```
Available components:
```

[1]	"cluster"	"centers"	"totss"	"withinss"
[5]	"tot.withinss"	"betweenss"	"size"	"iter"
[9]	"ifault"			

# K-Means Clusters vs. Principal Components ( $k = 3$ )



## Alternative: "Partitioning Around Medoids" ( $k = 3$ )

```
> PAM3 <- pam(ds,3)
```

```
> PAM3
```

```
Medoids:
```

	ID	X	Y	Z
Johnaton	7	-0.6226	-1.037	-0.6381
Heidi	12	-0.3315	1.297	1.4888
Erica	15	1.5634	-0.468	-0.6381

```
Clustering vector:
```

Guillermo	Rachel	Deidra	Quaton	Alicia	Angelique	Johnaton
1	2	1	3	1	2	1
Javan	Khulood	Cody	Cameron	Heidi	Maahir	Rogelio
1	1	1	1	2	2	3
Erica	Barren	Kiana	Elyse	Chadrick	Tahani	
3	3	2	2	3	3	

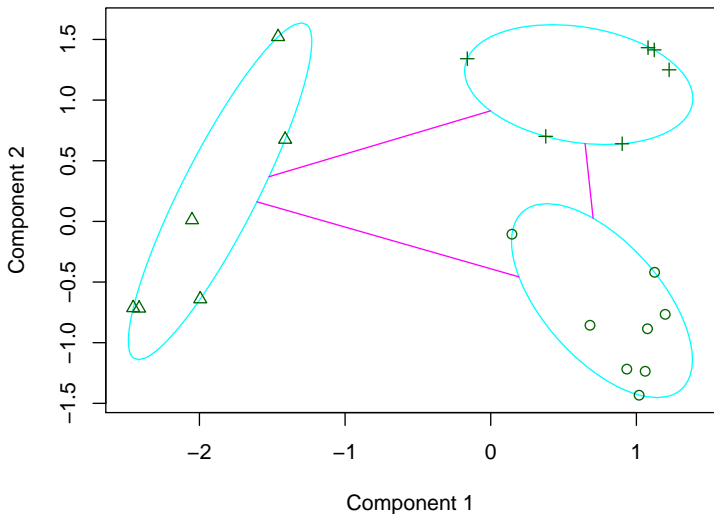
```
Objective function:
```

```
build swap  
0.7054 0.6573
```

```
Available components:
```

[1]	"medoids"	"id.med"	"clustering"	"objective"	"isolation"
[6]	"clusinfo"	"silinfo"	"diss"	"call"	"data"



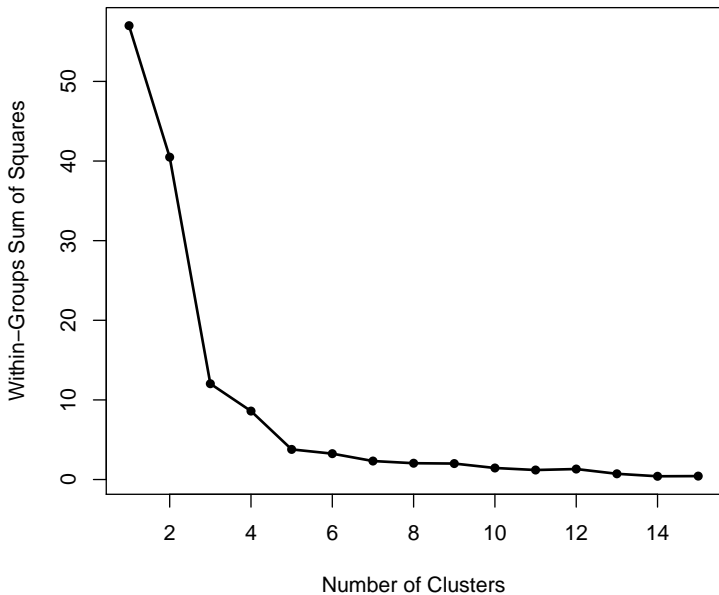
**PAM Cluster Plot (k=3)**

These two components explain 93.86 % of the point variability.

## Practical k-Means: Choosing $k$

- Theory
- Scree plot of WCSS
- “Model-based” approaches

## Choosing $k$ : Scree Plot



# Other Non-Hierarchical Methods

*DBSCAN* (Density-Based Spatial Clustering of Applications with Noise)

- *Density-based* method...
- Does not require prespecification of  $k$
- Also does not (necessarily) assign “outlying” observations to clusters
- R packages: [dbscan](#), others

*Mean-Shift* Clustering

- Operationally similar to DBSCAN
- IME works well with “non-spherical” cluster shapes
- R packages: [meanShiftR](#), [LPCM](#), etc.

# Real-Data Example: U.S. States

```
> url <- getURL("https://raw.githubusercontent.com/PrisonRodeo/
  PLSC504-2020-git/master/Data/States2005.csv")
> States <- read.csv(text = url)
>
> summary(States)
```

statename	Year	CitizenIdeology	GovernmentIdeology	govstaff
Alabama : 1	Min. :2005	Min. :28.2	Min. :10.1	Min. : 8.0
Alaska : 1	1st Qu.:2005	1st Qu.:43.5	1st Qu.:21.9	1st Qu.: 24.0
Arizona : 1	Median :2005	Median :53.1	Median :47.9	Median : 39.0
Arkansas : 1	Mean :2005	Mean :53.2	Mean :49.9	Mean : 59.1
California: 1	3rd Qu.:2005	3rd Qu.:61.3	3rd Qu.:71.8	3rd Qu.: 69.5
Colorado : 1	Max. :2005	Max. :91.2	Max. :92.0	Max. :310.0
(Other) :44				

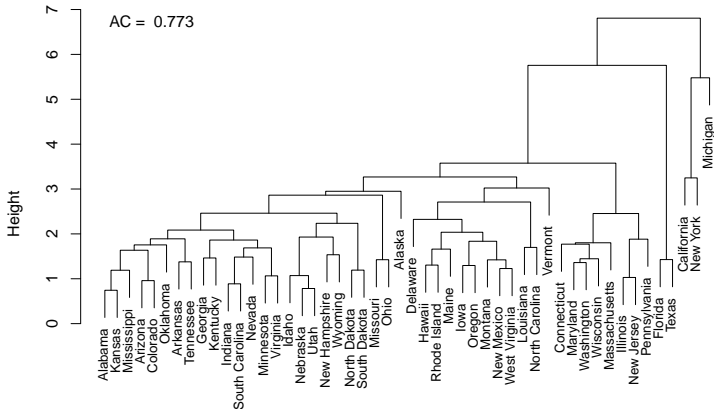
  

govsalary	legcomp	legsession	pop	lnGDP
Min. : 70000	Min. : 200	Min. : 25.0	Min. : 501	Min. :10.0
1st Qu.: 95000	1st Qu.: 15876	1st Qu.: 45.0	1st Qu.: 1772	1st Qu.:11.0
Median :112822	Median : 23696	Median : 67.5	Median : 4210	Median :11.9
Mean :115778	Mean : 31932	Mean : 79.0	Mean : 5918	Mean :11.9
3rd Qu.:131326	3rd Qu.: 41709	3rd Qu.: 99.2	3rd Qu.: 6398	3rd Qu.:12.6
Max. :179000	Max. :118600	Max. :352.0	Max. :36154	Max. :14.3

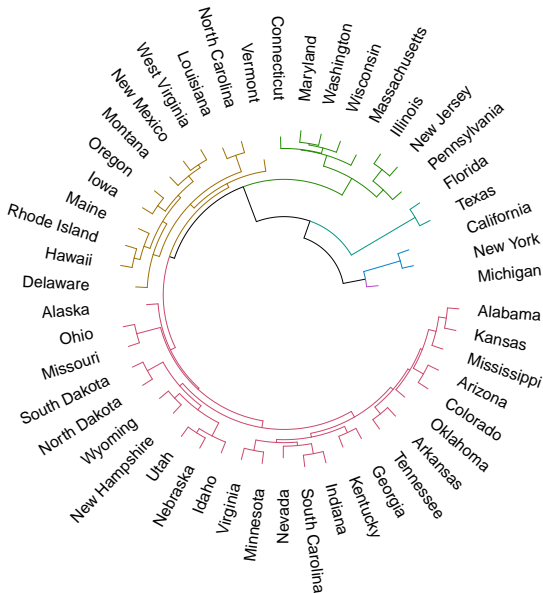
```
> StS <- data.frame(scale(States[,3:10]))
> rownames(StS)<-States$statename
```

# State Data: Agglomerative Dendrogram

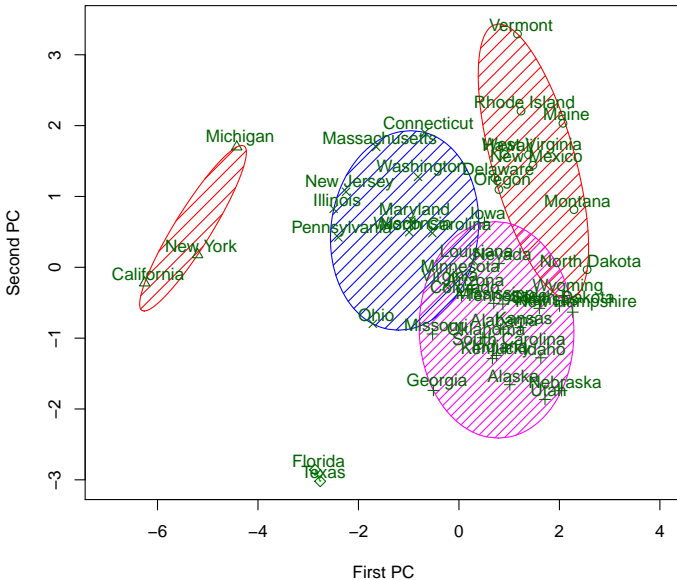
Euclidean Distance / Average Linkage



# State Data: Cooler Agglomerative Dendrogram



## State Data: K-Means Results





# Useful References

- Johnson, S.C. 1967. "Hierarchical Clustering Schemes." *Psychometrika* 32:241-254.
- Reynolds, A., Richards, G., de la Iglesia, B. and Rayward-Smith, V. 1992. "Clustering Rules: A Comparison of Partitioning and Hierarchical Clustering Algorithms." *Journal of Mathematical Modelling and Algorithms* 5:475-504.
- Kaufman, Leonard, and Peter J. Rousseeuw. 2005. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.
- Hennig, Christian, Marina Meila, Fionn Murtagh, and Roberto Rocci, eds. 2015. *Handbook of Cluster Analysis*. New York: Chapman & Hall.
- Everitt, Brian S., Sabine Landau, Morven Leese, and Daniel Stahl. 2011. *Cluster Analysis*, 5th Ed. New York: Wiley.
- Kassambara, Alboukadel. 2017. *Practical Guide to Cluster Analysis in R*. Createspace.

# Useful R Packages and Routines

- `hclust` and `kmeans` (in `stats`)
- `agnes` and `diana` and `pam` (in `cluster`)
- `amap` (alternative agglomerative and  $k$ -means clustering)
- `dendextend` (additional functionality for dendograms; e.g., comparisons)
- `mclust` (model-based clustering via MLE)
- `FactoClass` (combinations of factorial and clustering methods)

... and many more.

- The Cluster Analysis R Task View: <http://cran.cnr.berkeley.edu/web/views/Cluster.html>
- The Data Flair R Clustering tutorial: <https://data-flair.training/blogs/r-clustering-tutorial/>
- The dendextend vignette:  
[https://cran.r-project.org/web/packages/dendextend/vignettes/Cluster\\_Analysis.html](https://cran.r-project.org/web/packages/dendextend/vignettes/Cluster_Analysis.html)