PLSC 504: Fall 2024

Principal Components Analysis, Factor Analysis, and Clustering

October 30, 2024

Measurement: Concepts

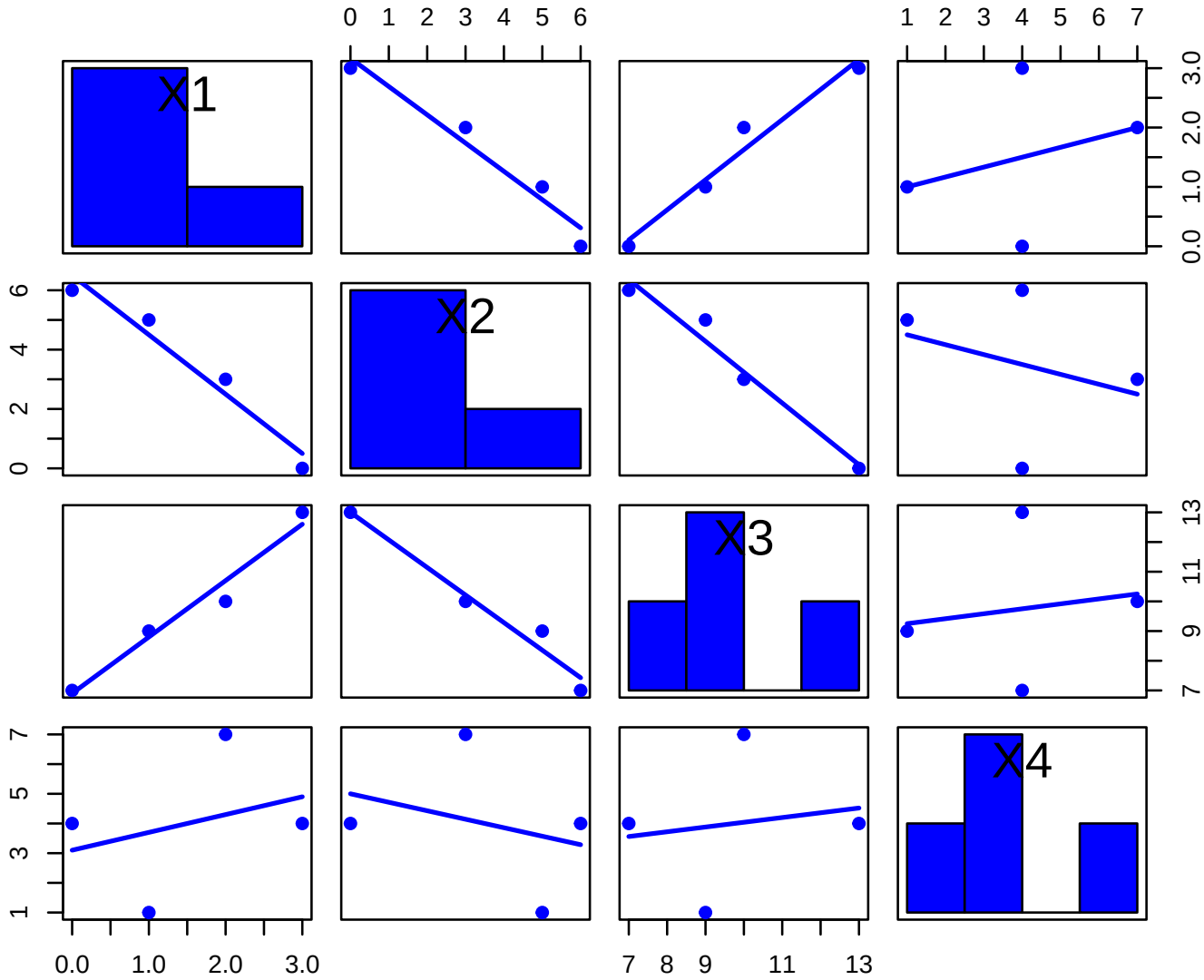
- Often: *data reduction* (many $\mathbf{X} \rightarrow$ one X)
- Classification *is* measurement (taxonomy and typology)
- Level and quality of measurement are distinct
- **All measurement implies theory**

Today's Plan

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

```
> 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


X1
X2
X3
X4
0 1 2 3 4 5 6
1 2 3 4 5 6 7
0.0 1.0 2.0 3.0
0 2 4 6
7 9 11 13
1 3 5 7
0.0 1.0 2.0 3.0
7 8 9 11 13

```
> 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 Σ of (centered) $\mathbf{X}$, we can diagonalize:

$$\Sigma = \mathbf{VLV}'$$

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 Σ
- 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:

$$\mathbf{X} = \mathbf{USV}'$$

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 $\mathbf{US}$ ($\equiv \mathbf{XV}$).

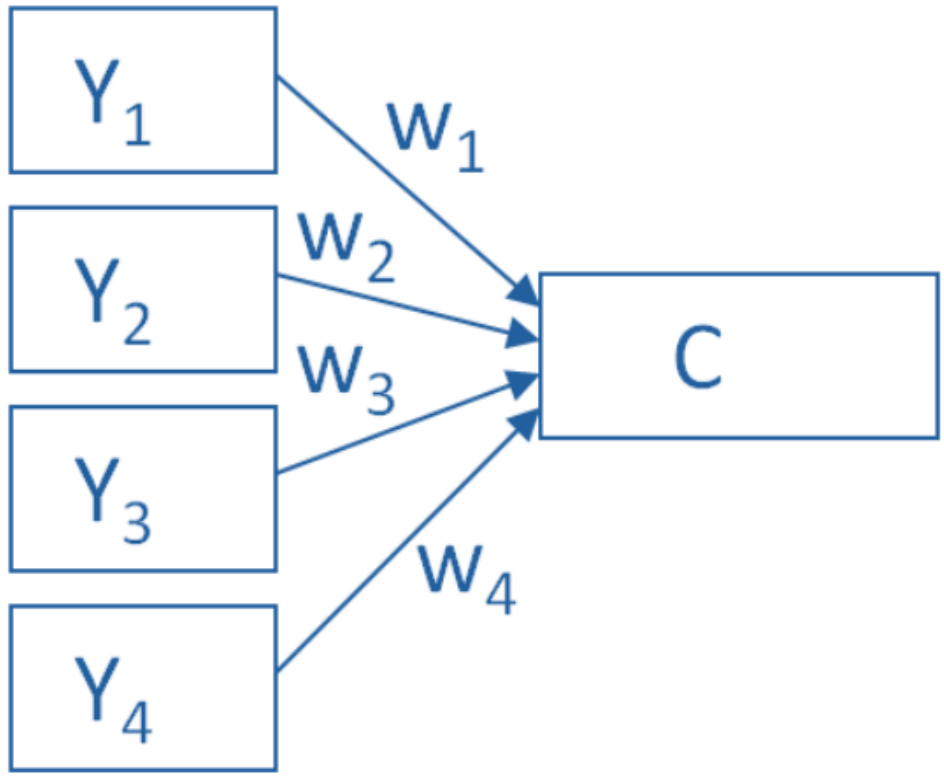
```
> SVD <- svd(CX)
> SVD
$d
[1] 6.7848667363326811142 4.0586048392632072535 0.4932647655322611735
[4] 0.0000000000000002276

$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 $\underset{N \times K}{\mathbf{X}}$ 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.

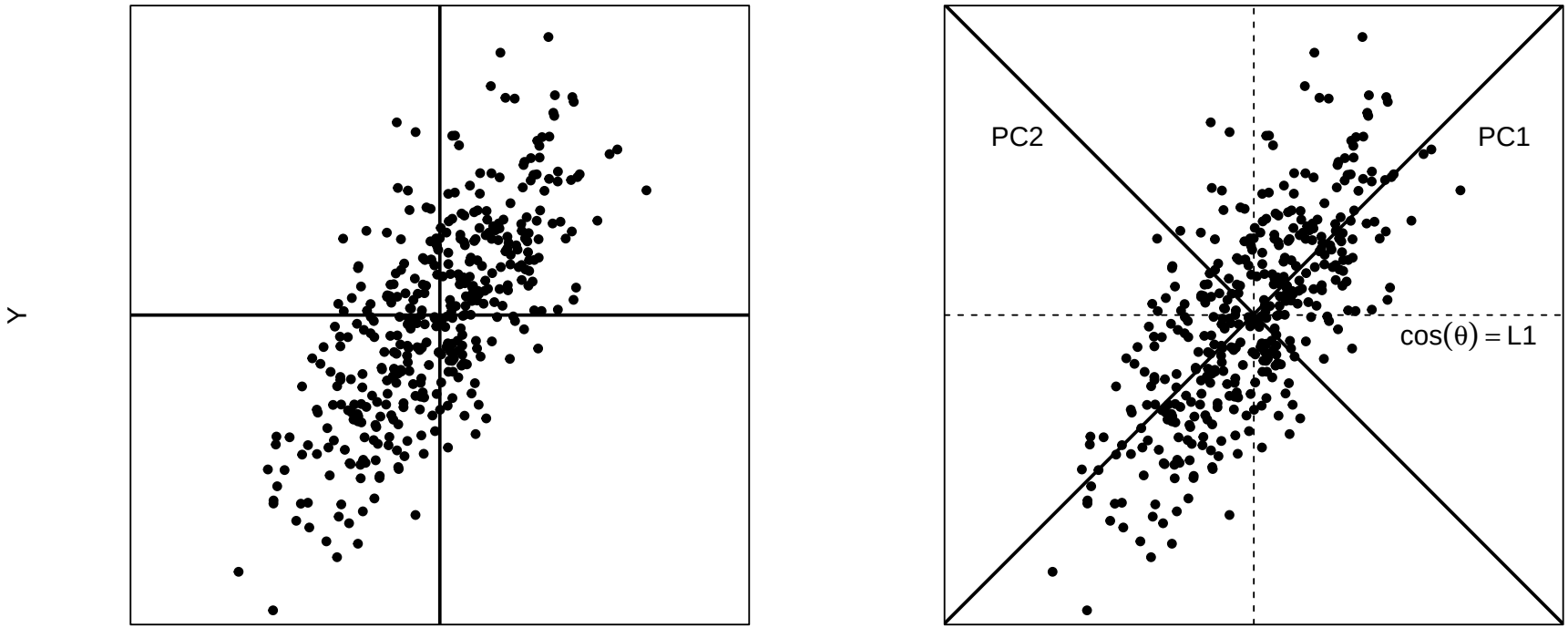
PCA, Conceptually



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

(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
```

PCA Steps

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

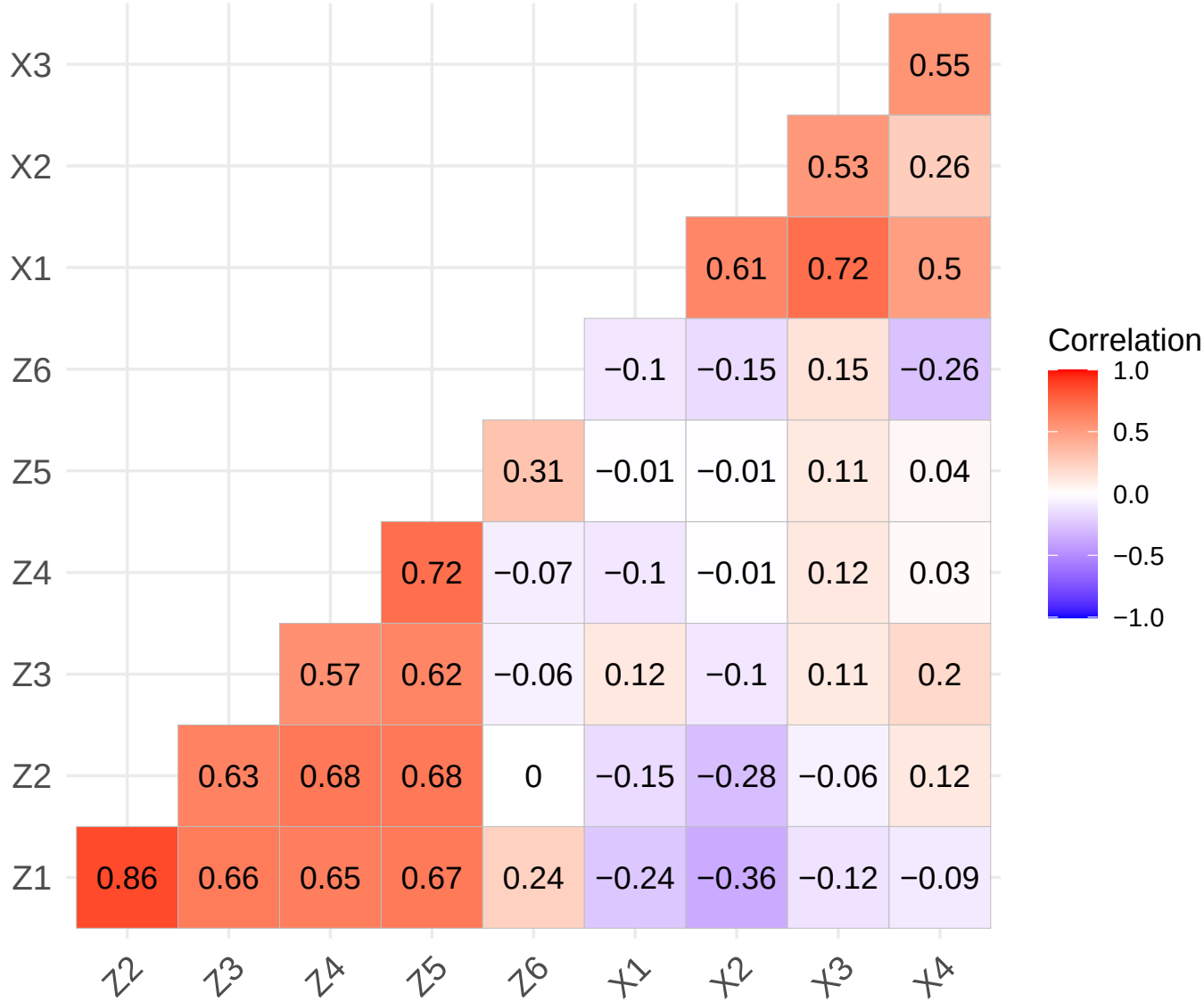
PCA: A Simulation Example

```
> N <- 20
> set.seed(7222009)
> Name <- randomNames(N, which.names="first")
> Z <- rnorm(N)
> Z1 <- Z + 0.2*rnorm(N) # six variables Z1-Z6,
> Z2 <- Z + 0.5*rnorm(N) # correlated with (latent) Z to varying
> Z3 <- Z + 1*rnorm(N)   # degrees
> Z4 <- Z + 1.5*rnorm(N)
> Z5 <- Z + 2*rnorm(N)
> Z6 <- Z + 3*rnorm(N)
>
> X <- rnorm(N)
> X1 <- X + rnorm(N) # four more variables X1-X4,
> X2 <- X + rnorm(N) # correlated with (latent) X,
> X3 <- X + rt(N,5)  # uncorrelated with Z
> 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	
Zane	-0.3835	-0.8551	-1.74863	-2.95738	0.3434	2.3876	1.8148	-0.048860	-0.9089	-2.0279	
Kikiola	0.9500	2.1191	-0.71930	1.29382	1.5645	-0.9844	-0.1672	-2.136591	-1.1117	0.5004	
Kelly	0.7722	0.6555	1.30132	1.31898	1.1723	-2.9590	-2.1983	-1.854578	-2.8231	-1.0201	
Janeth	0.6151	0.9362	0.70202	1.15355	-0.3083	-2.9951	-3.0352	-2.207591	-2.1784	-2.8660	
Jeremy	-0.7086	-0.3088	-0.51730	-0.64314	-1.7469	-0.9893	2.4642	2.755191	2.4353	0.3730	
Kristopher	-0.4598	-1.2545	0.50662	0.25929	-0.4636	-0.7435	1.0900	-0.046470	0.6645	1.2849	
Dakota	-0.6067	0.7625	-0.08669	1.72531	1.2346	-4.5091	-0.8283	-0.686047	-1.6056	1.0410	
Bryanna	0.2787	0.9993	1.13377	-1.86677	-0.9118	-3.5107	0.3449	-2.018978	0.7354	3.1029	
Irma	-2.0618	-2.2968	-1.04520	-4.52706	-3.8013	-0.9035	0.7487	-1.696923	-1.2566	-0.5973	<---
Abdut Tawwab	-1.1526	-2.3654	-3.83437	-1.92830	-2.4938	0.5076	-3.6260	-0.250650	-2.4459	-2.3053	
Ryan	0.5422	0.5225	-2.20508	0.32334	1.4927	4.2313	-1.8403	-0.873431	-1.3248	-1.6506	
Christiana	-0.5757	0.1382	-1.90018	-1.88046	0.1188	1.1511	-1.5067	0.299969	0.8937	1.0714	
Shane	1.4385	1.0143	1.63319	0.41833	0.2214	-2.8103	0.6063	-0.002911	-0.2441	0.9327	
Evelyn	1.3790	1.9599	4.67315	2.01085	6.3573	2.0813	-0.8455	0.870719	-0.9608	-0.1952	<---
Brandon	-0.3556	0.3509	0.55577	1.95601	2.1527	-6.9956	1.0326	0.168683	1.3465	-0.1441	
Chelsey	0.6760	0.3684	0.43527	1.34559	1.7534	6.2837	-0.4531	-2.522037	3.0872	0.1913	
Jonique	-0.1961	0.1497	0.33599	1.27368	2.5173	0.5435	3.6935	3.752891	3.7851	1.0968	<---
Eduardo	0.1541	0.2013	-0.24649	-0.01127	3.6202	2.4549	-0.6014	-1.945691	-0.5654	0.7837	
Naomi	-1.0365	-1.5128	-0.45662	-0.04055	0.6739	-1.1768	1.4737	1.900248	2.2425	1.3754	
Erin	-1.6944	-1.3492	-3.06741	-1.66362	-2.3350	-5.5588	0.6075	2.347850	-0.1236	1.6992	

Correlations, Envisioned



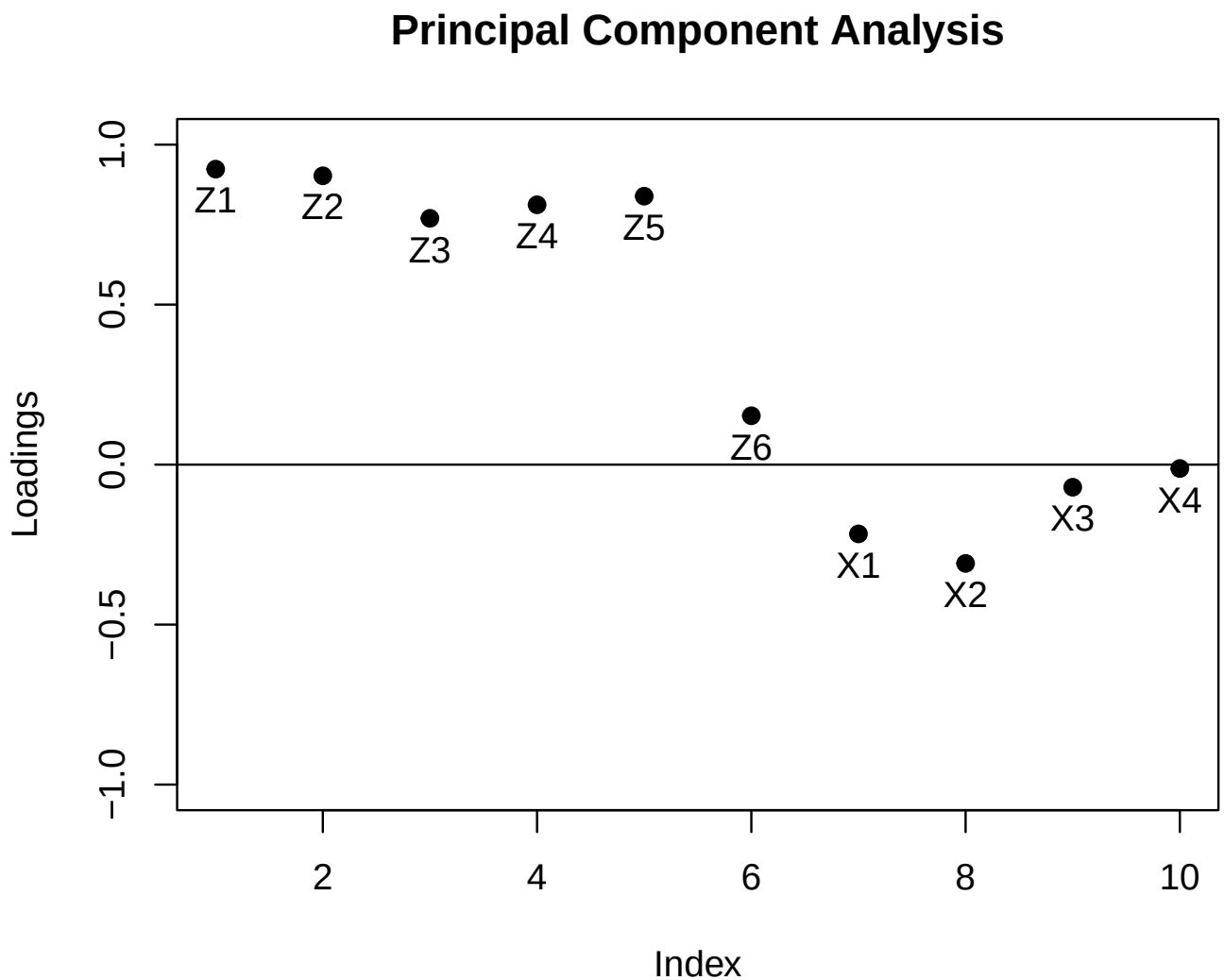
Friendly PCA using principal

```
> 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  0.92 0.85255 0.15   1
Z2  0.90 0.81465 0.19   1
Z3  0.77 0.59243 0.41   1
Z4  0.81 0.65934 0.34   1
Z5  0.84 0.70383 0.30   1
Z6  0.15 0.02331 0.98   1
X1 -0.22 0.04688 0.95   1
X2 -0.31 0.09537 0.90   1
X3 -0.07 0.00501 0.99   1
X4 -0.01 0.00015 1.00   1

                PC1
SS loadings    3.79
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  0.23
 with the empirical chi square  94.71  with prob <  0.00000021

Fit based upon off diagonal values = 0.67
```

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, $\texttt{PC1}^2$)
- `u2` is *uniqueness*; simply $1 - \texttt{h2}$
- `SS Loadings` is the value(s) of the principal component(s)
- `Proportion Var` is the proportion of the total variance in $\mathbf{X}$ 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



PCA Scores

```
> PCSim1$scores

                PC1
Zane         -0.7662
Kikiola       0.9758
Kelly         0.8876
Janeth        0.7121
Jeremy       -0.7649
Kristopher   -0.3504
Dakota        0.3313
Bryanna       0.1087
Irma         -1.9027 <---
Abdut Tawwab -1.4531
Ryan          0.3533
Christiana   -0.4900
Shane         0.7625
Evelyn        2.1131 <---
Brandon       0.3494
Chelsey       0.8004
Jonique       0.1025 <---
Eduardo       0.5495
Naomi        -0.6965
Erin         -1.6226
```

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  0.92 -0.10 0.862 0.14 1.0
Z2  0.90  0.05 0.817 0.18 1.0
Z3  0.77  0.28 0.673 0.33 1.3
Z4  0.81  0.19 0.695 0.30 1.1
Z5  0.84  0.20 0.746 0.25 1.1
Z6  0.15 -0.14 0.043 0.96 2.0
X1 -0.22  0.86 0.792 0.21 1.1
X2 -0.31  0.70 0.585 0.42 1.4
X3 -0.07  0.86 0.744 0.26 1.0
X4 -0.01  0.72 0.519 0.48 1.0

                       PC1  PC2
SS loadings           3.79 2.68
Proportion Var        0.38 0.27
Cumulative Var        0.38 0.65
Proportion Explained  0.59 0.41
Cumulative Proportion 0.59 1.00

Mean item complexity =  1.2
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is  0.1
 with the empirical chi square  17.79  with prob <  0.88

Fit based upon off diagonal values = 0.94
```

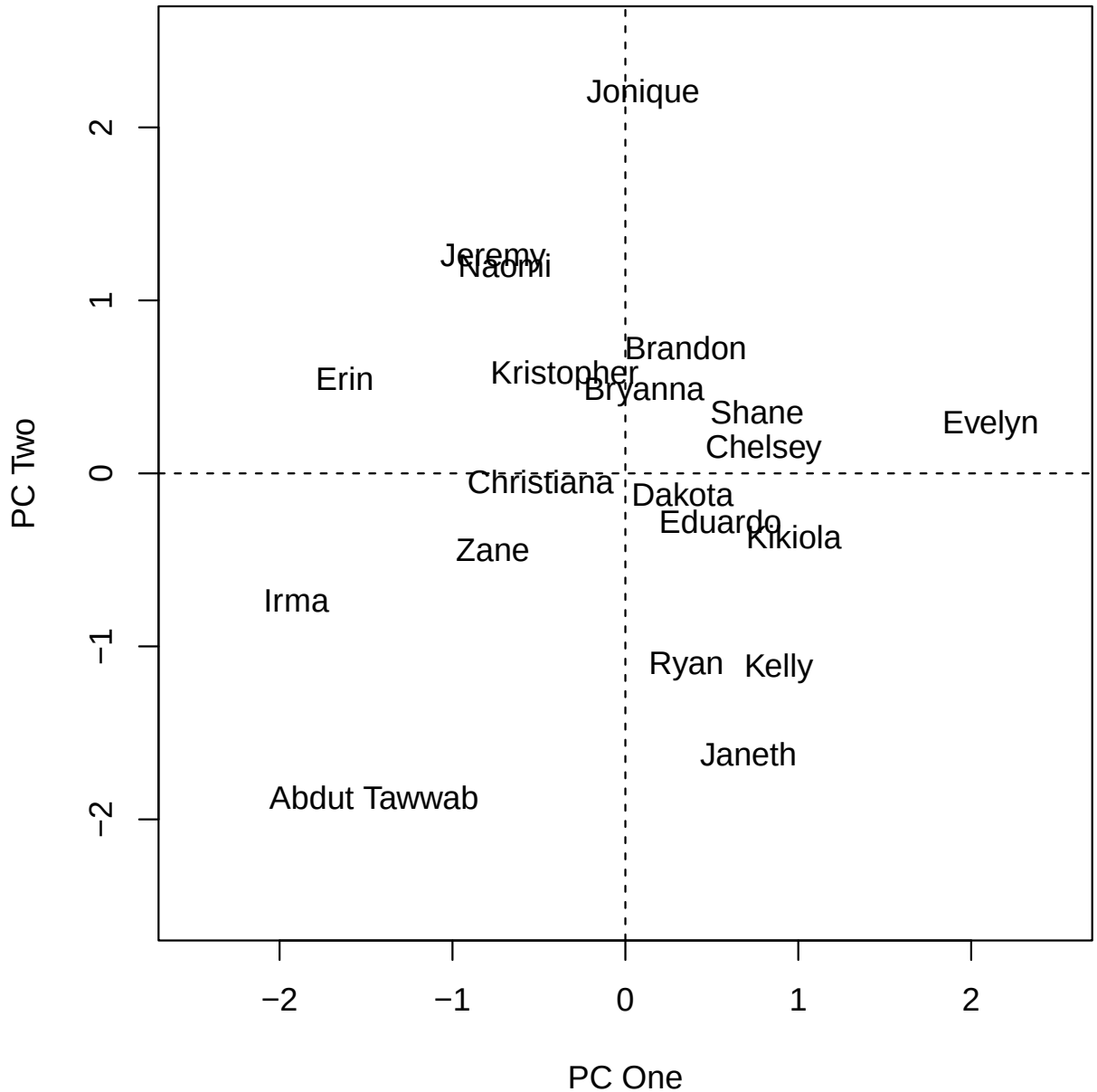
Let's break that down again...

- 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
Zane         -0.7662 -0.44184
Kikiola       0.9758 -0.36603
Kelly         0.8876 -1.12584
Janeth        0.7121 -1.62294
Jeremy       -0.7649  1.24398
Kristopher   -0.3504  0.56805
Dakota        0.3313 -0.12155
Bryanna       0.1087  0.47345
Irma         -1.9027 -0.73273 <---
Abdut Tawwab -1.4531 -1.87429
Ryan          0.3533 -1.11299
Christiana   -0.4900 -0.04596
Shane         0.7625  0.35459
Evelyn        2.1131  0.27739 <---
Brandon       0.3494  0.72381
Chelsey       0.8004  0.13787
Jonique       0.1025  2.19239 <---
Eduardo       0.5495 -0.27627
Naomi        -0.6965  1.19975
Erin         -1.6226  0.54919
```

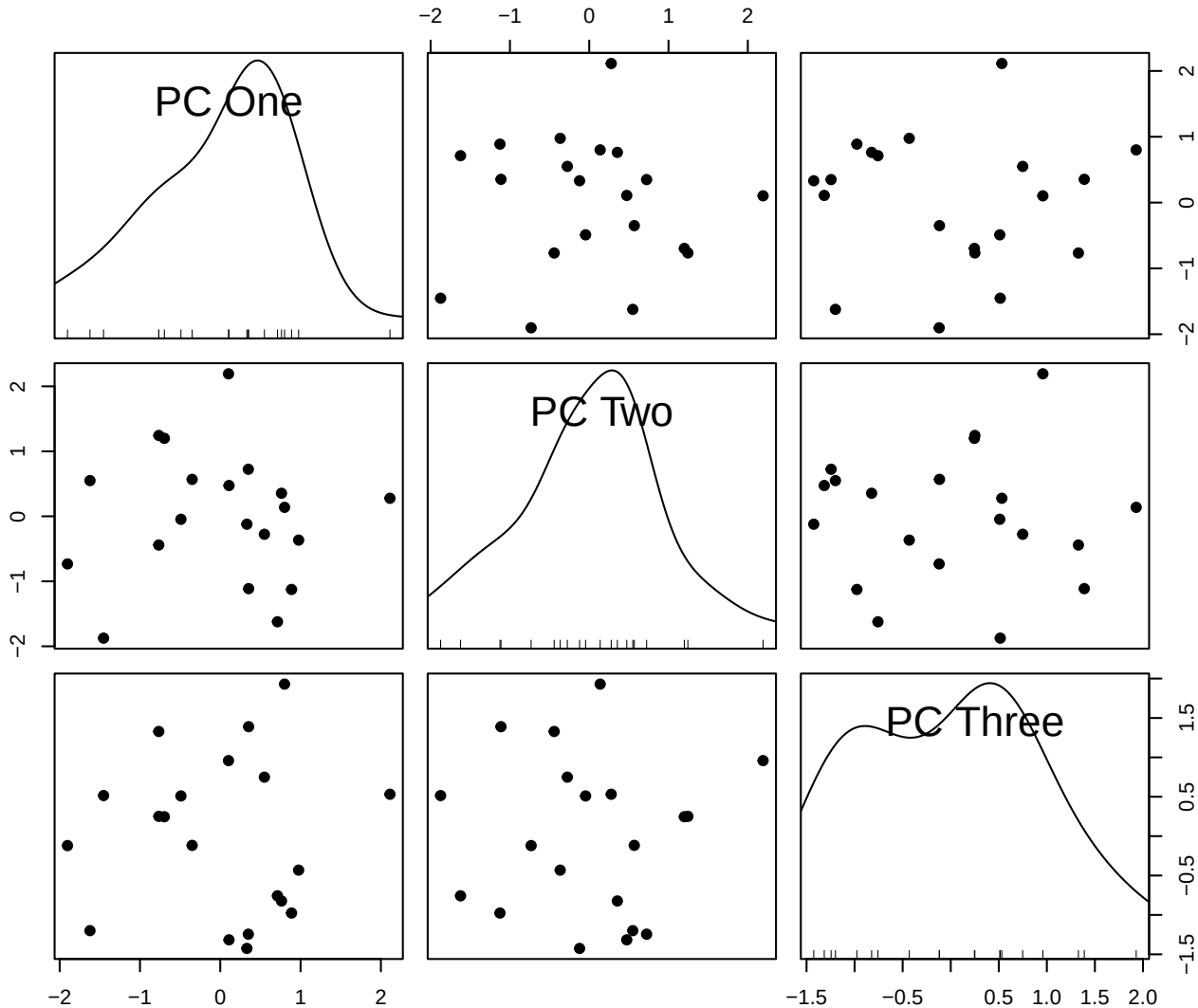
PCA Scores


Jonique
Jeremy
Naomi
Brandon
Erin
Kristopher
Bryanna
Shane
Evelyn
Chelsey
Christiana
Dakota
Eduardo
Kikiola
Zane
Irma
Ryan
Kelly
Janeth
Abdut Tawwab
PC Two
PC One
2
1
0
−1
−2
−2
−1
0
1
2

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  0.92 -0.10  0.07 0.87 0.13 1.0
Z2  0.90  0.05 -0.16 0.84 0.16 1.1
Z3  0.77  0.28 -0.18 0.70 0.30 1.4
Z4  0.81  0.19 -0.11 0.71 0.29 1.1
Z5  0.84  0.20  0.26 0.81 0.19 1.3
Z6  0.15 -0.14  0.95 0.94 0.06 1.1
X1 -0.22  0.86  0.07 0.80 0.20 1.1
X2 -0.31  0.70  0.08 0.59 0.41 1.4
X3 -0.07  0.86  0.31 0.84 0.16 1.3
X4 -0.01  0.72 -0.30 0.61 0.39 1.3

                       PC1  PC2  PC3
SS loadings           3.79 2.68 1.23
Proportion Var        0.38 0.27 0.12
Cumulative Var        0.38 0.65 0.77
Proportion Explained  0.49 0.35 0.16
Cumulative Proportion 0.49 0.84 1.00

Mean item complexity =  1.2
Test of the hypothesis that 3 components are sufficient.

The root mean square of the residuals (RMSR) is  0.07
 with the empirical chi square  9.39  with prob <  0.95

Fit based upon off diagonal values = 0.97
```

PCA Scores, Again



Biplots!

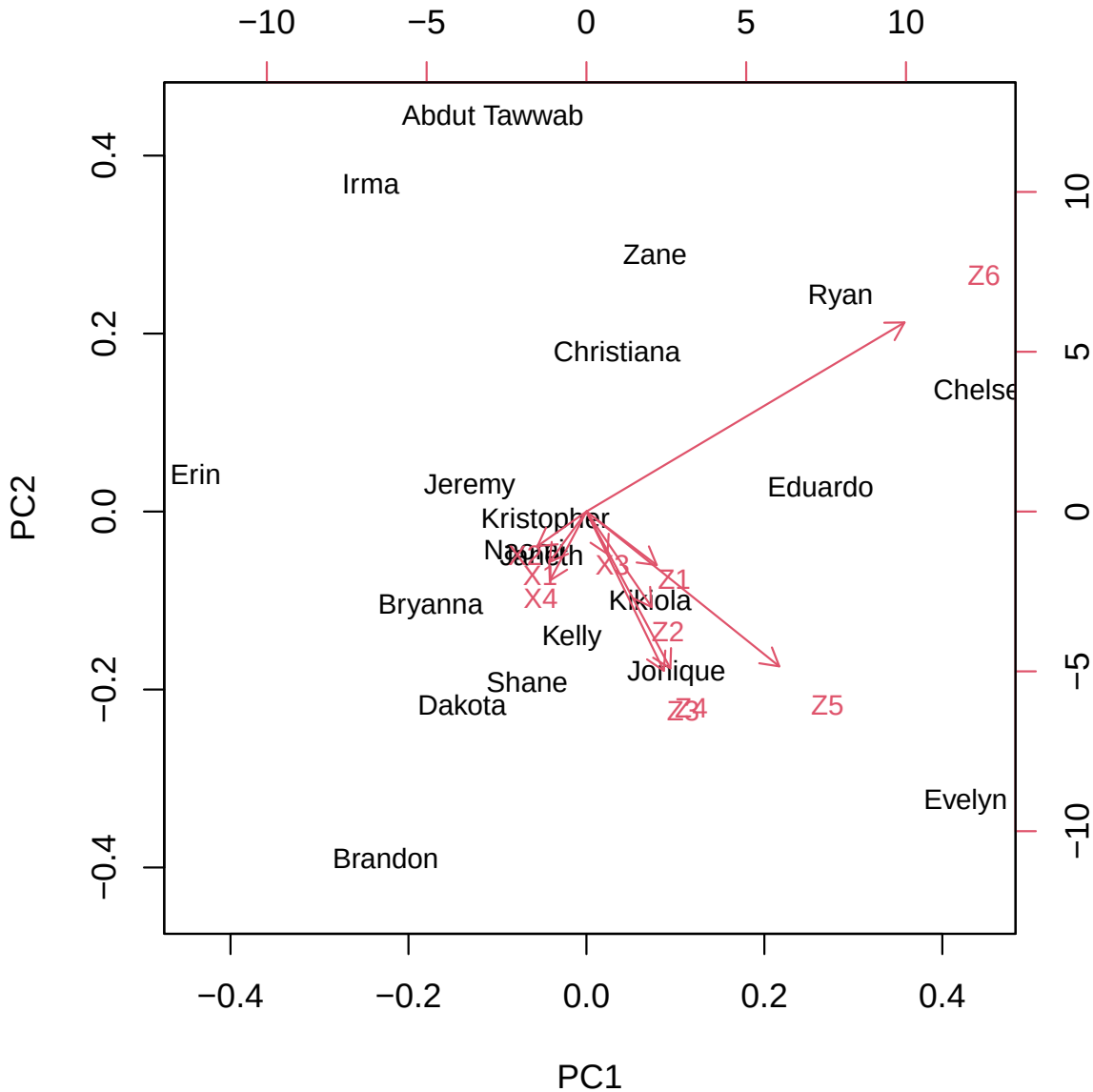
A *biplot* is a graphical representation of a two-axis PCA.

- 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  0.17250 -0.14730
Z2  0.16008 -0.26294
Z3  0.18994 -0.43829
Z4  0.20597 -0.43169
Z5  0.47316 -0.42613
Z6  0.78038  0.52068
X1 -0.09044 -0.13912
X2 -0.12002 -0.09508
X3  0.05148 -0.11673
X4 -0.08948 -0.18971
```

A Biplot...

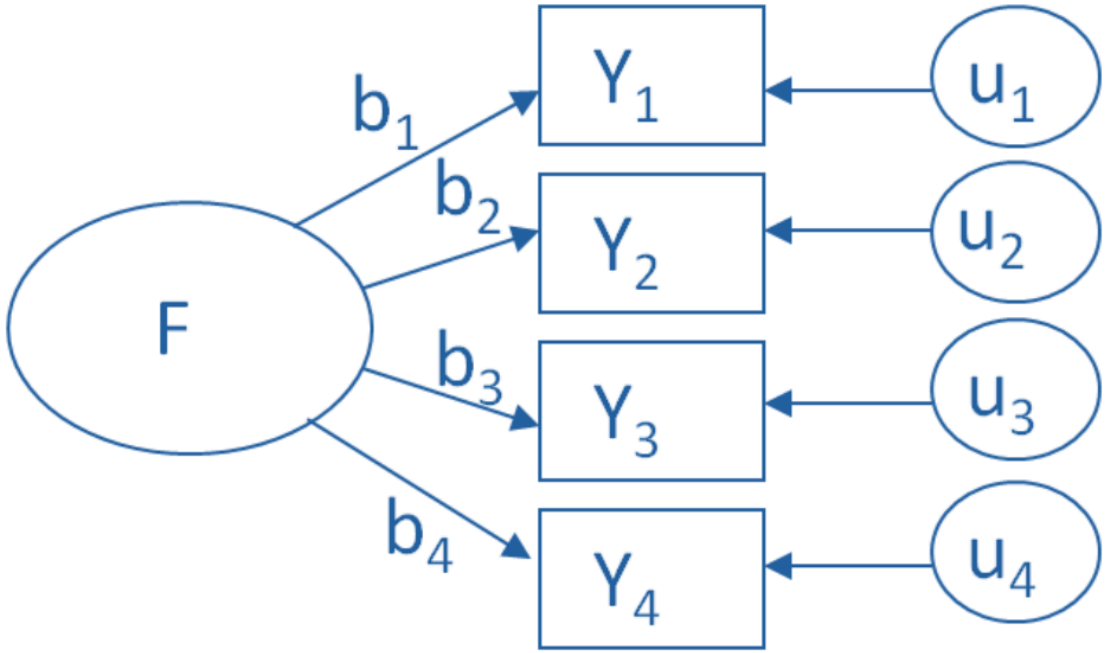

−10
−5
0
5
10
Abdut Tawwab
Irma
Zane
Ryan
Z6
Christiana
Chelse
Erin
Jeremy
Eduardo
Kristopher
Naomi
Janeth
X3
X1
Z1
X4
Bryanna
Kikiola
Kelly
Z2
Jonique
Shane
Dakota
Z3
Z4
Z5
Evelyn
Brandon
0.4
0.2
0.0
−0.2
−0.4
PC2
10
5
0
−5
−10
−0.4
−0.2
0.0
0.2
0.4
PC1

(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



$$
\begin{aligned}
Y_1 &= b_1 F + u_1 \\
Y_2 &= b_2 F + u_2 \\
Y_3 &= b_3 F + u_3 \\
Y_4 &= b_4 F + u_4
\end{aligned}
$$

(Source)

Formally:

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

This implies that the observed covariance matrix Σ can be written:

$$\Sigma = \Lambda\Lambda' + \Psi$$

where

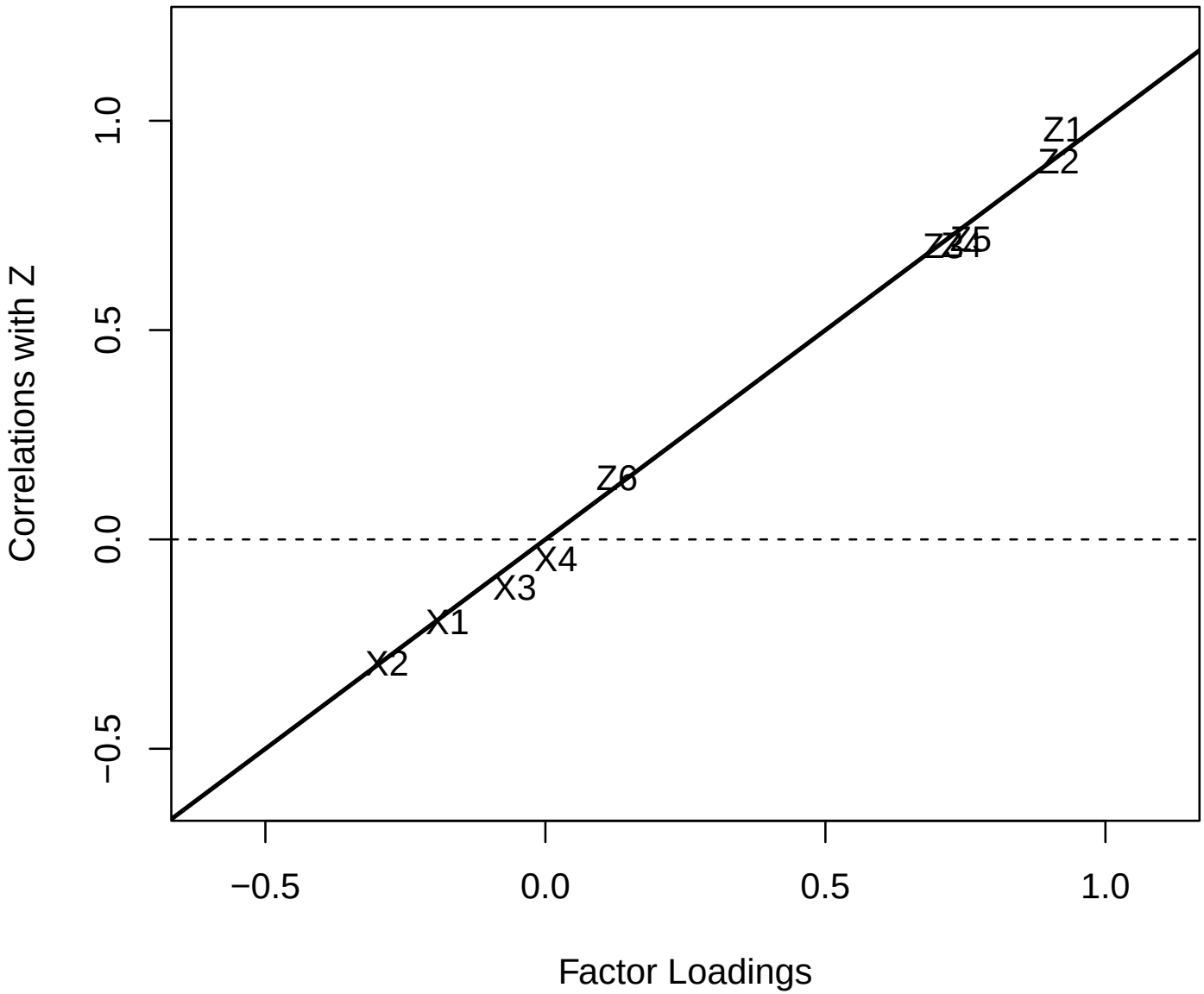
$$\Psi = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_K^2 \end{bmatrix}$$

- 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
0.143 0.160 0.495 0.447 0.422 0.984 0.969 0.920 0.997 1.000

Loadings:
   Factor1
Z1  0.926
Z2  0.917
Z3  0.711
Z4  0.744
Z5  0.760
Z6  0.128
X1 -0.175
X2 -0.283
X3 -0.055
X4  0.019

               Factor1
SS loadings      3.463
Proportion Var   0.346

Test of the hypothesis that 1 factor is sufficient.
The chi square statistic is 53.92 on 35 degrees of freedom.
The p-value is 0.0215
```

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
0.122 0.168 0.426 0.426 0.384 0.978 0.239 0.519 0.310 0.638

Loadings:
   Factor1 Factor2
Z1  0.937  -0.021
Z2  0.907   0.090
Z3  0.695   0.302
Z4  0.731   0.201
Z5  0.745   0.247
Z6  0.142  -0.042
X1 -0.251   0.836
X2 -0.341   0.604
X3 -0.123   0.822
X4 -0.034   0.600

               Factor1 Factor2
SS loadings      3.489   2.301
Proportion Var   0.349   0.230
Cumulative Var   0.349   0.579

Test of the hypothesis that 2 factors are sufficient.
The chi square statistic is 26.5 on 26 degrees of freedom.
The p-value is 0.436
```

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
0.098 0.136 0.414 0.394 0.348 0.005 0.301 0.535 0.171 0.552

Loadings:
   Factor1 Factor2 Factor3
Z1  0.915  -0.023   0.252
Z2  0.926   0.083   0.011
Z3  0.712   0.275  -0.054
Z4  0.745   0.216  -0.060
Z5  0.703   0.238   0.317
Z6 -0.014  -0.001   0.997
X1 -0.228   0.798  -0.104
X2 -0.318   0.584  -0.151
X3 -0.149   0.885   0.152
X4  0.011   0.617  -0.259

               Factor1 Factor2 Factor3
SS loadings      3.428   2.329   1.289
Proportion Var   0.343   0.233   0.129
Cumulative Var   0.343   0.576   0.705

Test of the hypothesis that 3 factors are sufficient.
The chi square statistic is 15.69 on 18 degrees of freedom.
The p-value is 0.614
```

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
```

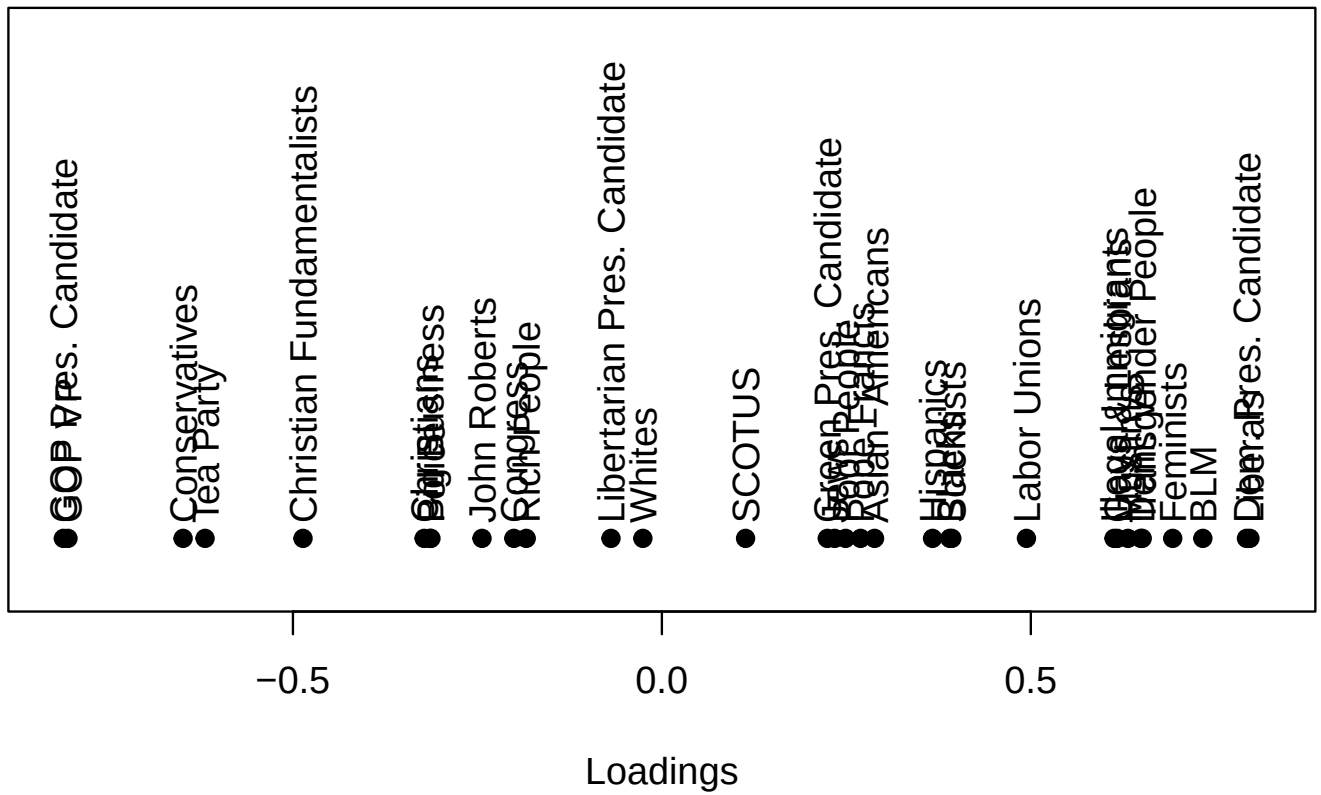
Factor Analysis: One Factor

```
> 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
```

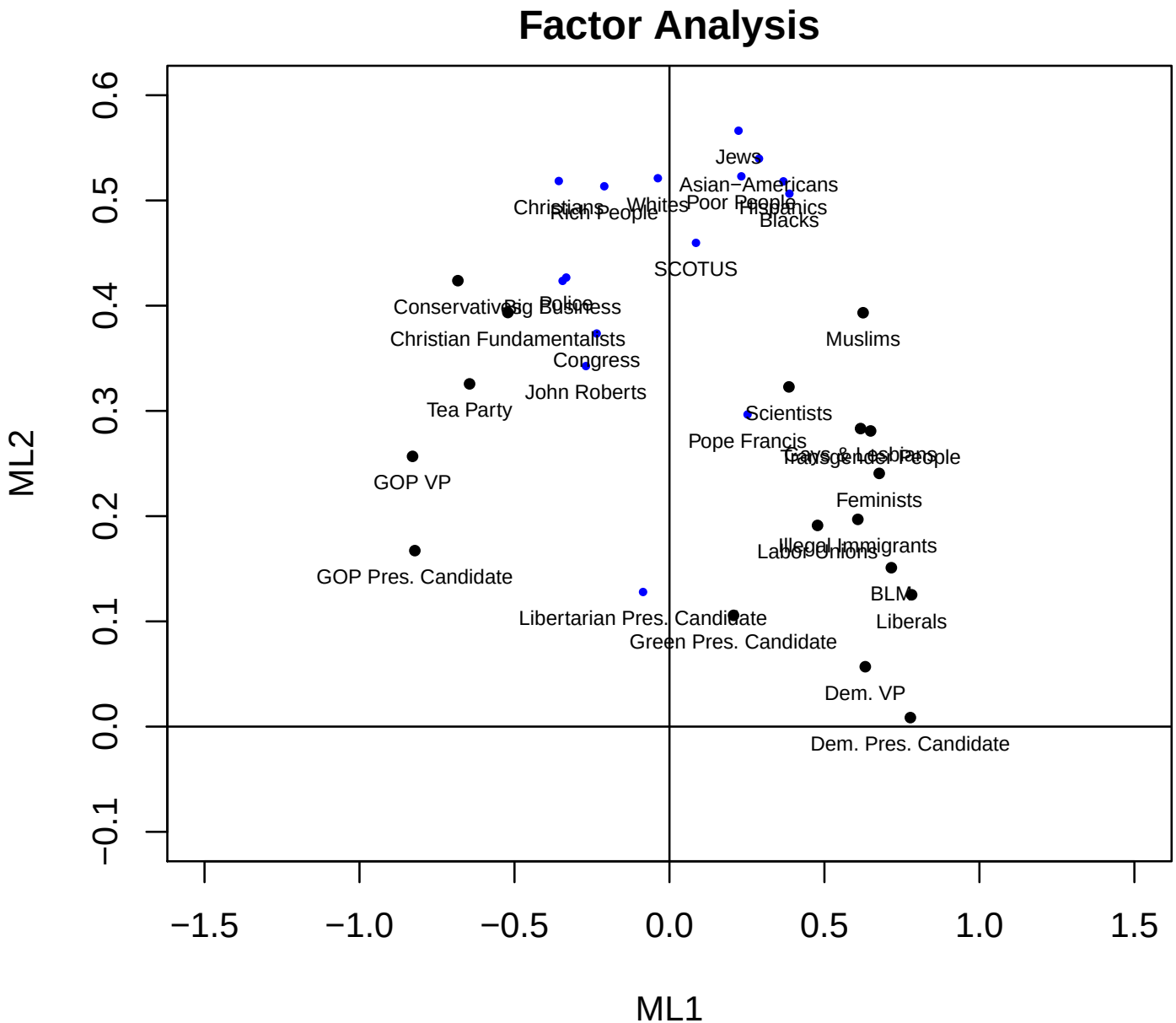
Factor Loadings



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



Topic: Rotation

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")**

Rotation Methods

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...

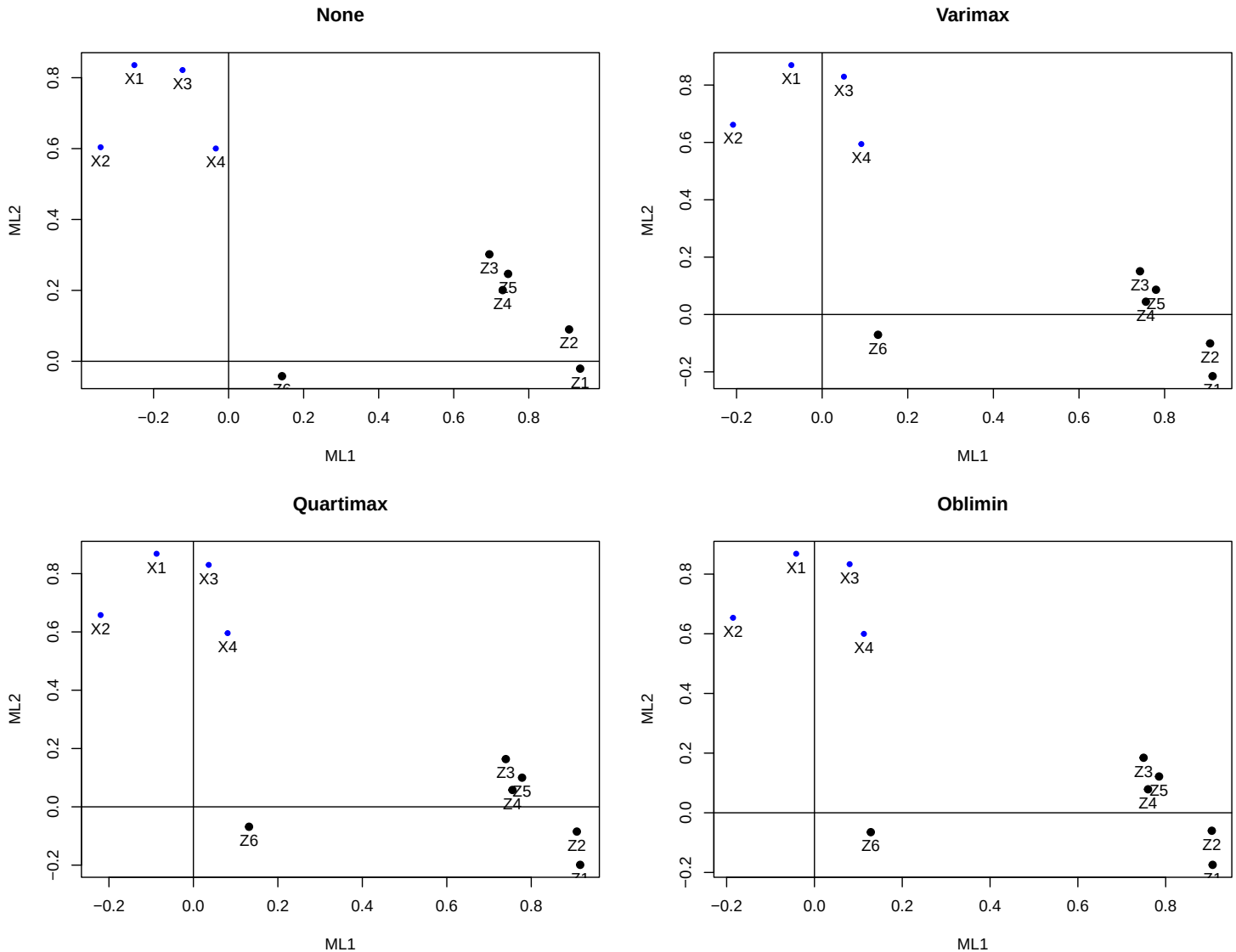
Rotation: Considerations

"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)

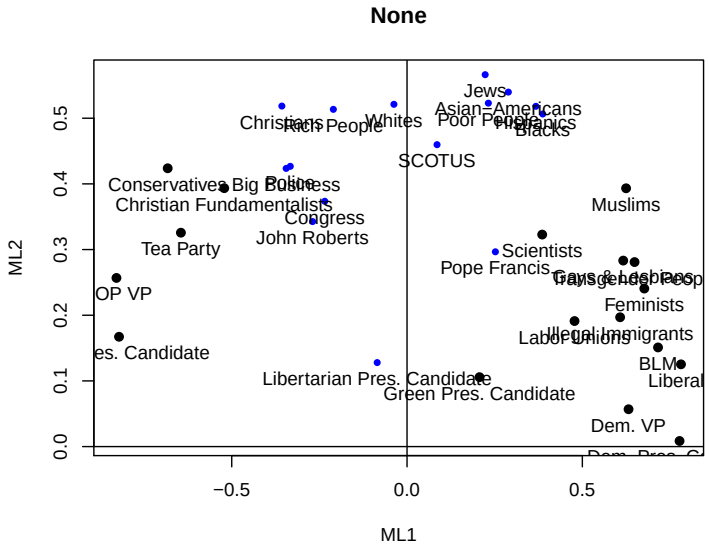
Factor Loading ℓ Guidelines:

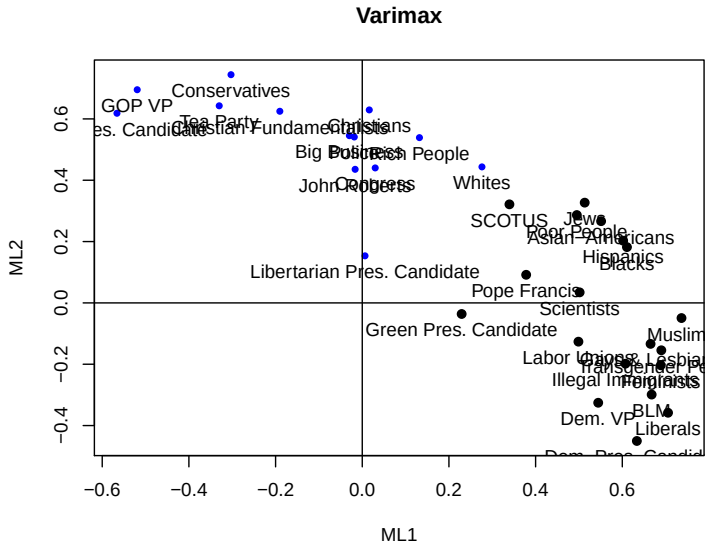
- $0.10 < \ell < -0.10$ are unimportant
- $|\ell| > 0.30$ are important with $N \geq 100$
- Variables with $\ell > 0.30$ on more than one factor are *complex*

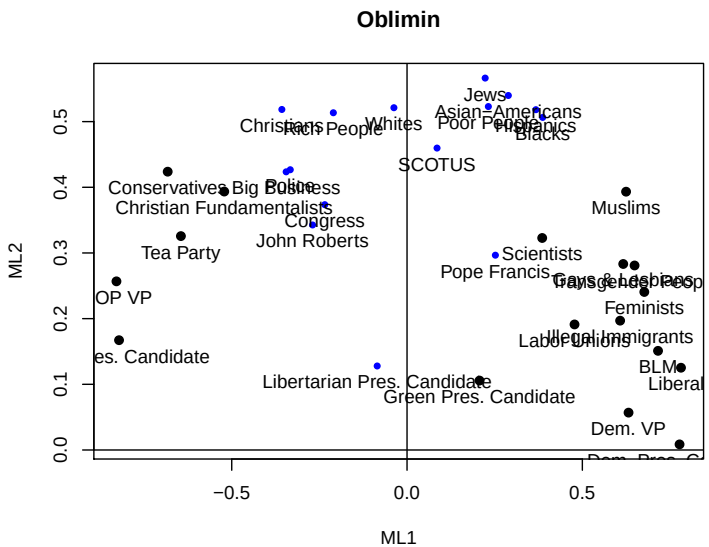
Rotation: Simulated Data



Rotation: Feeling Thermometers







Topic: Dimensionality

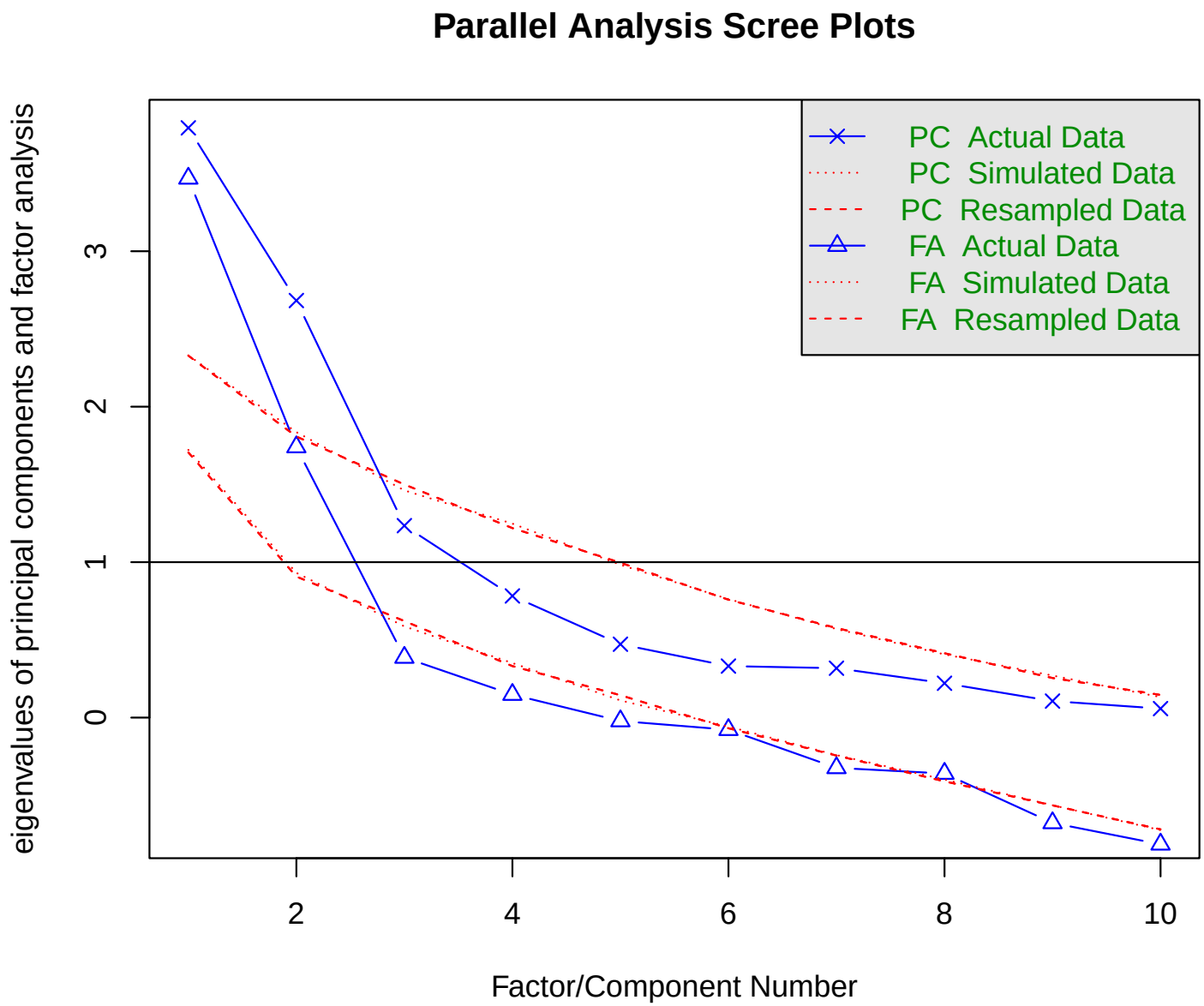
PCA/FA are *data reduction* techniques...

- 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"

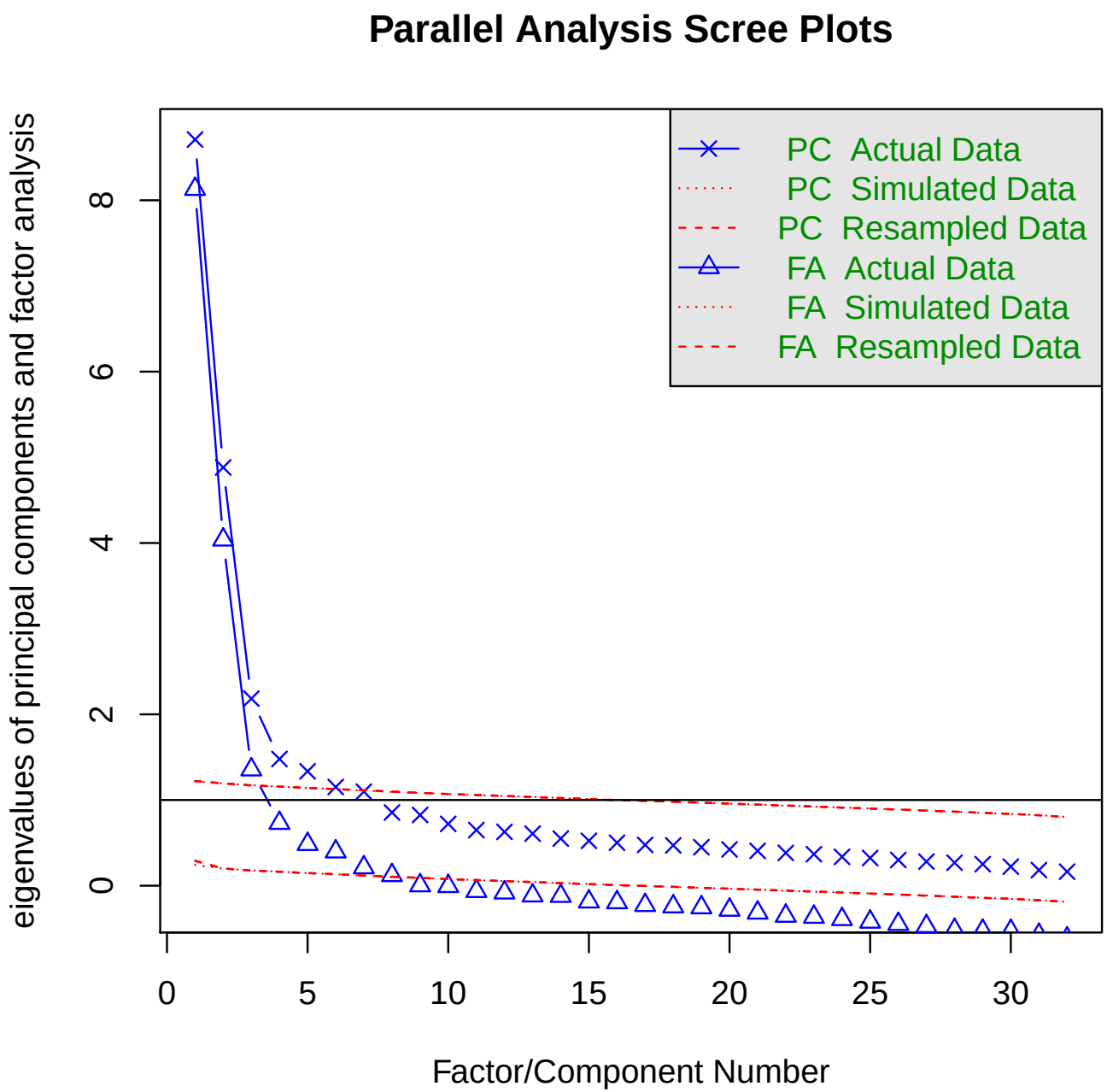
Other approaches:

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

"Parallel" Scree Plot (Simulated Data)



"Parallel" Scree Plot (Feeling Thermometer Data)



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

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."

Why Cluster?

- 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.

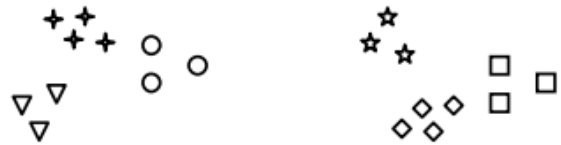


Figure 1c: Six clusters

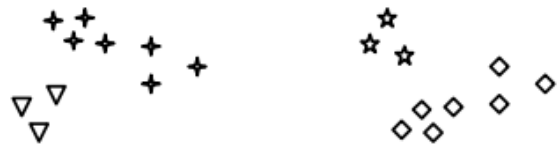
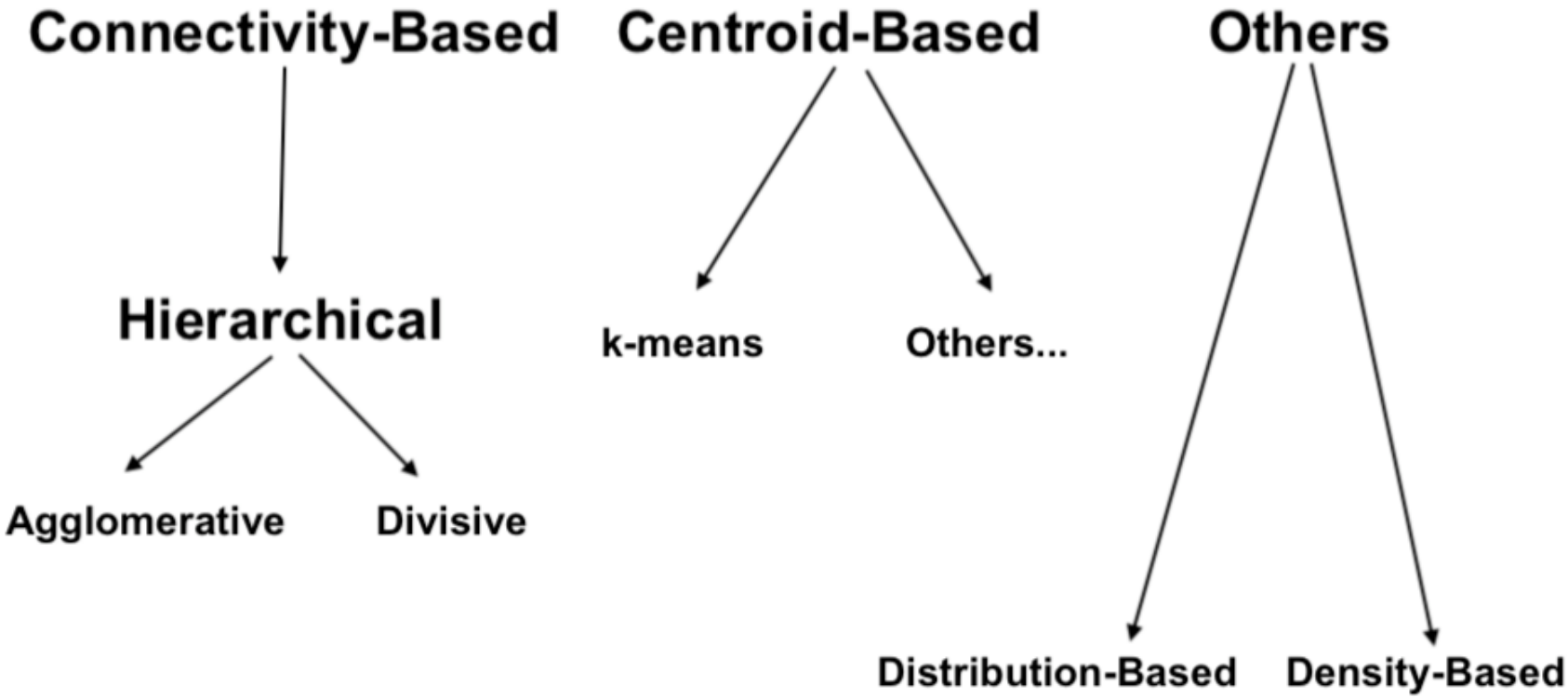


Figure 1d: Four clusters.

(source)

Cluster Analysis: Typology



Euclidean ("L2") Distance:

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

"City-Block" / Manhattan ("L1") Distance:

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

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

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

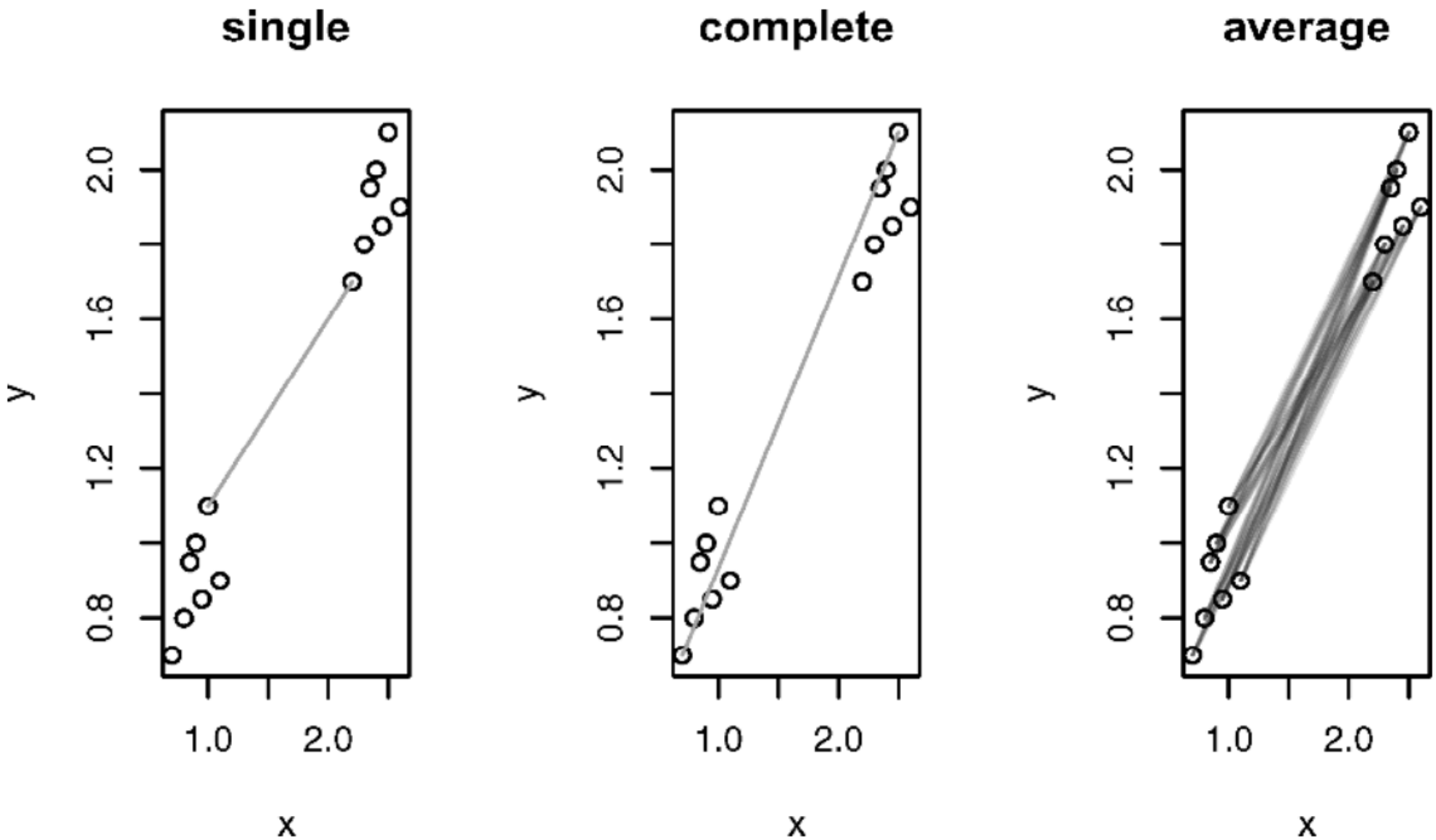
- Complete linkage

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

- 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



Agglomerative Clustering

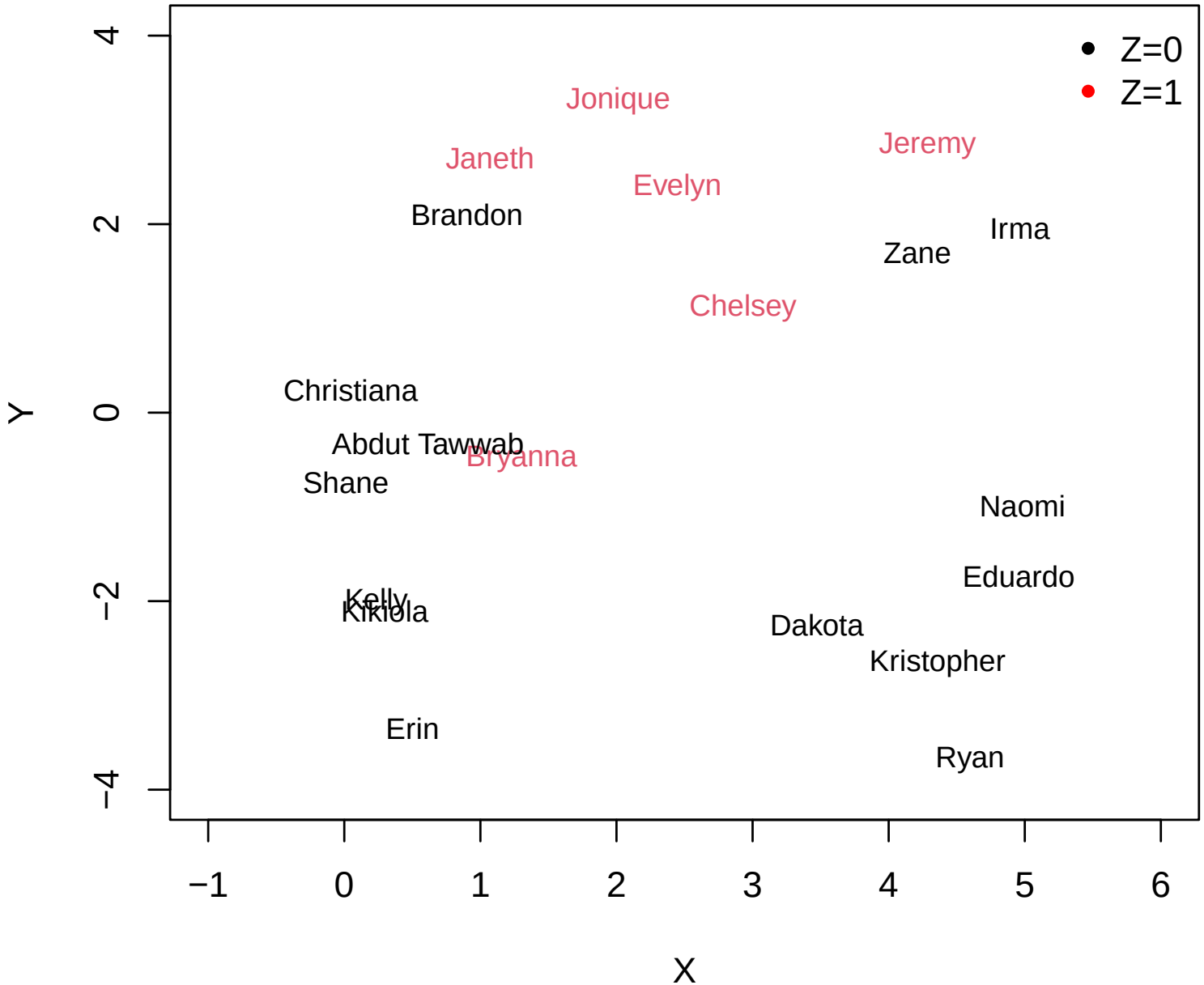
Basic steps:

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_ℓ 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

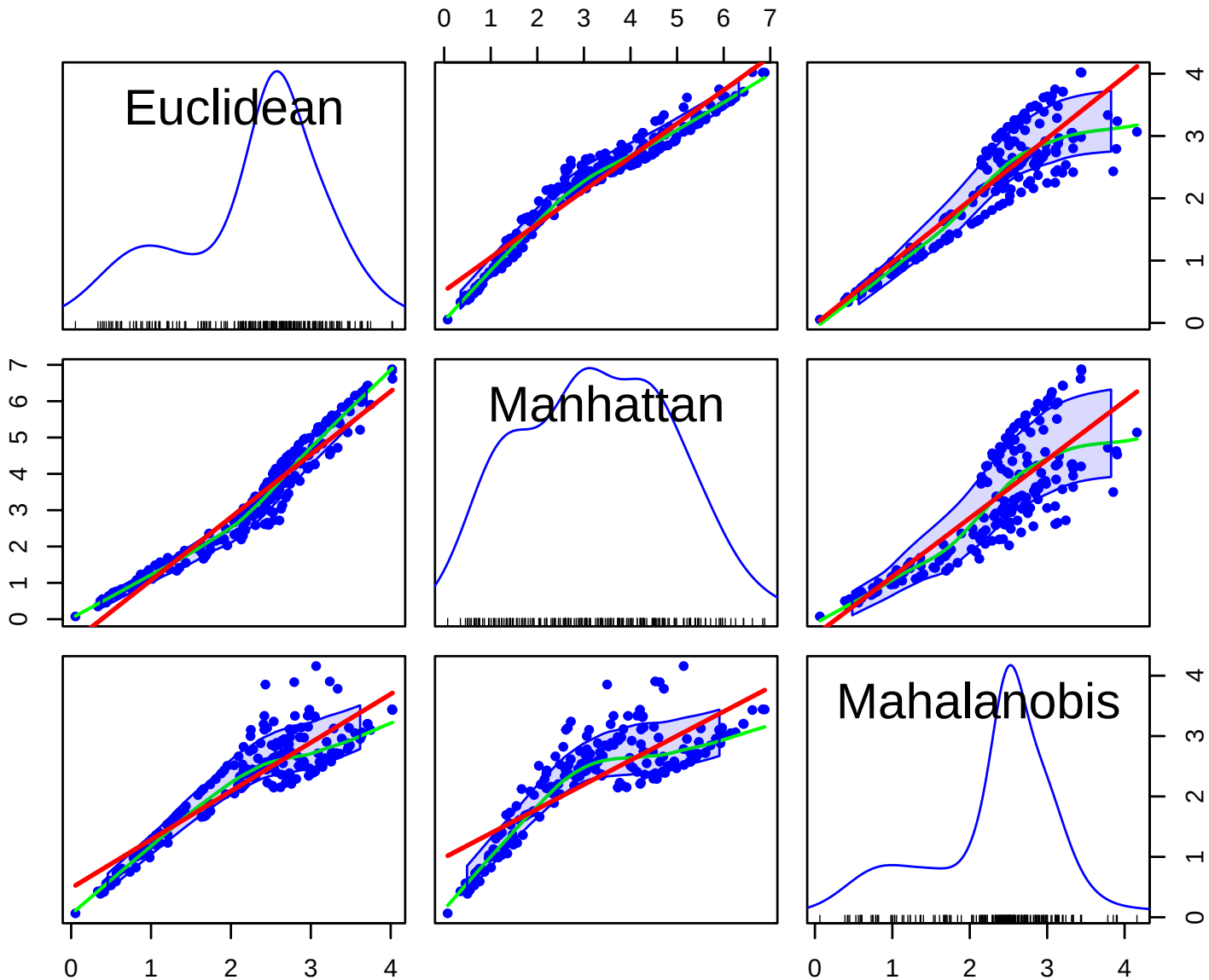
Simulation Example

```
> 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



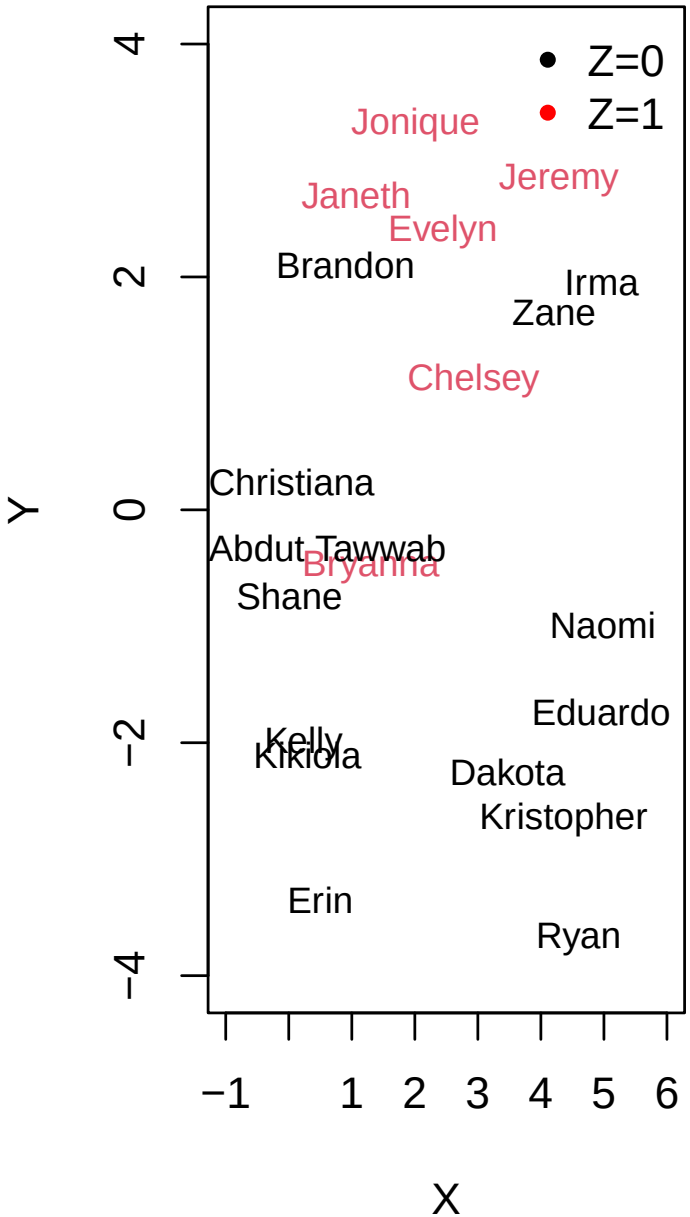
Distance Comparisons

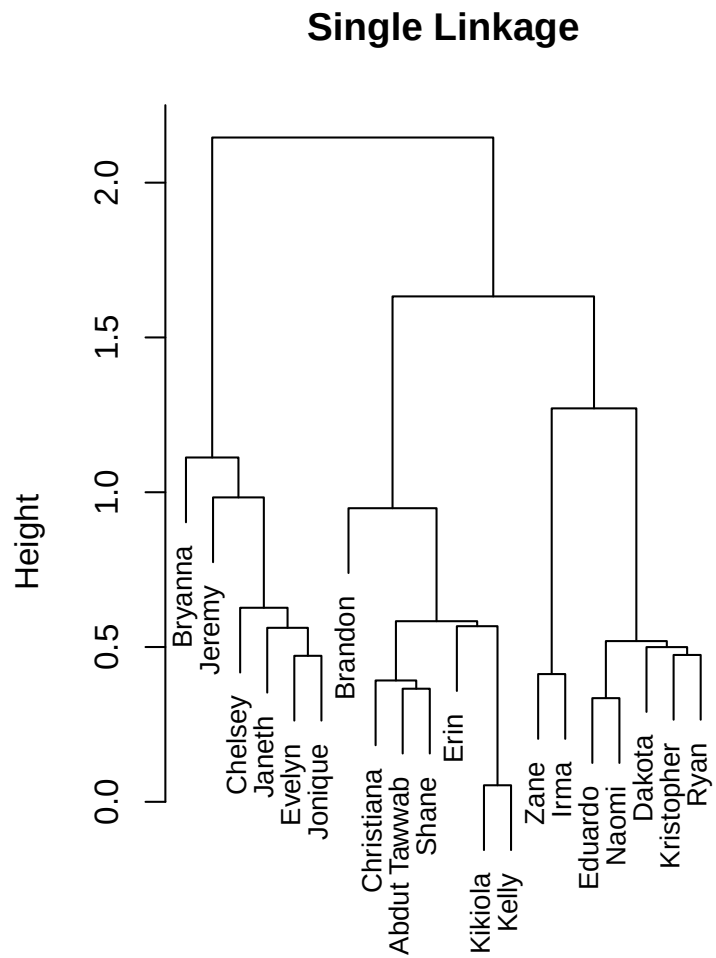


Using hclust (in cluster)

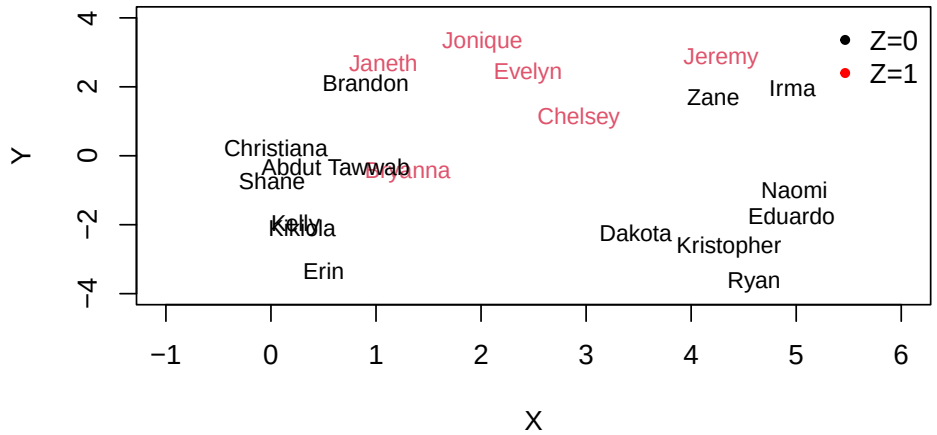
```
> ADL2.s <- hclust(DL2,method="single")
> ADL2.c <- hclust(DL2,method="complete")
> ADL2.a <- hclust(DL2,method="average")

> str(ADL2.s)
List of 7
 $ merge      : int [1:19, 1:2] -2 -18 -10 -12 -1 -14 -6 -7 2 -4 ...
 $ height     : num [1:19] 0.0535 0.3348 0.3653 0.392 0.4126 ...
 $ order      : int [1:20] 8 5 16 4 14 17 15 12 10 13 ...
 $ labels     : chr [1:20] "Zane" "Kikiola" "Kelly" "Janeth" ...
 $ method     : chr "single"
 $ call       : language hclust(d = DL2, method = "single")
 $ dist.method: chr "euclidean"
 - attr(*, "class")= chr "hclust"
```

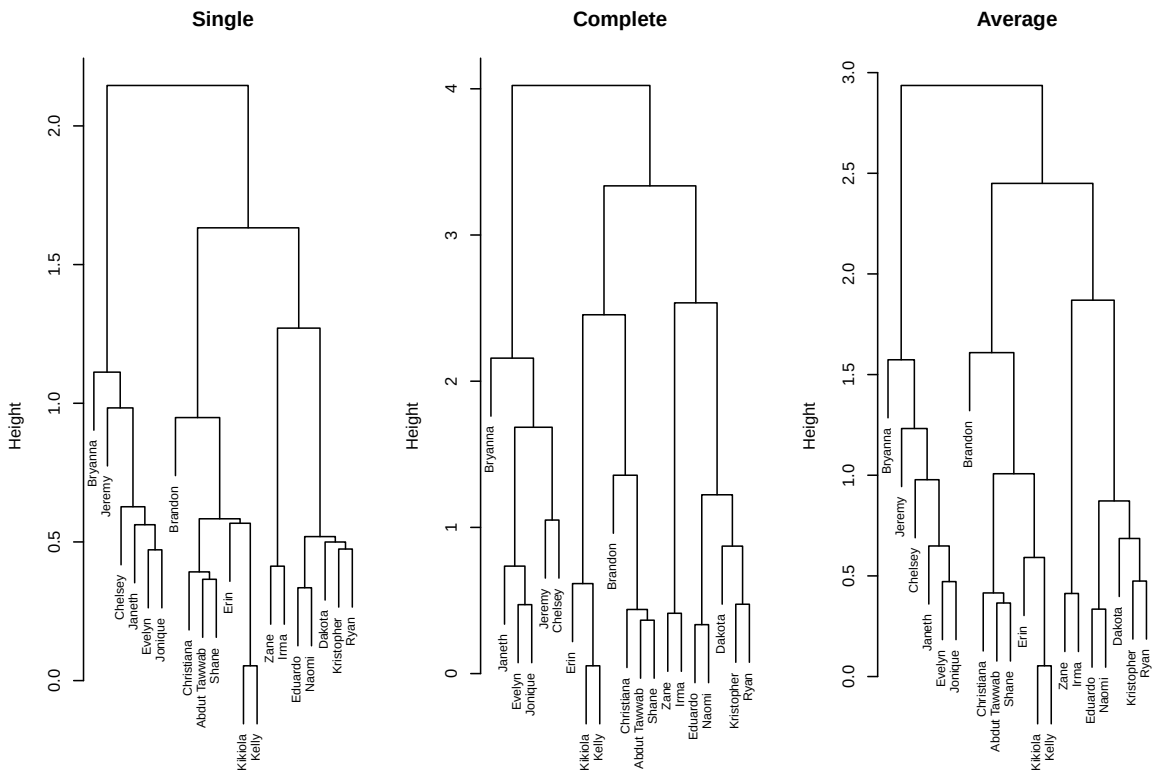
The Dendrogram



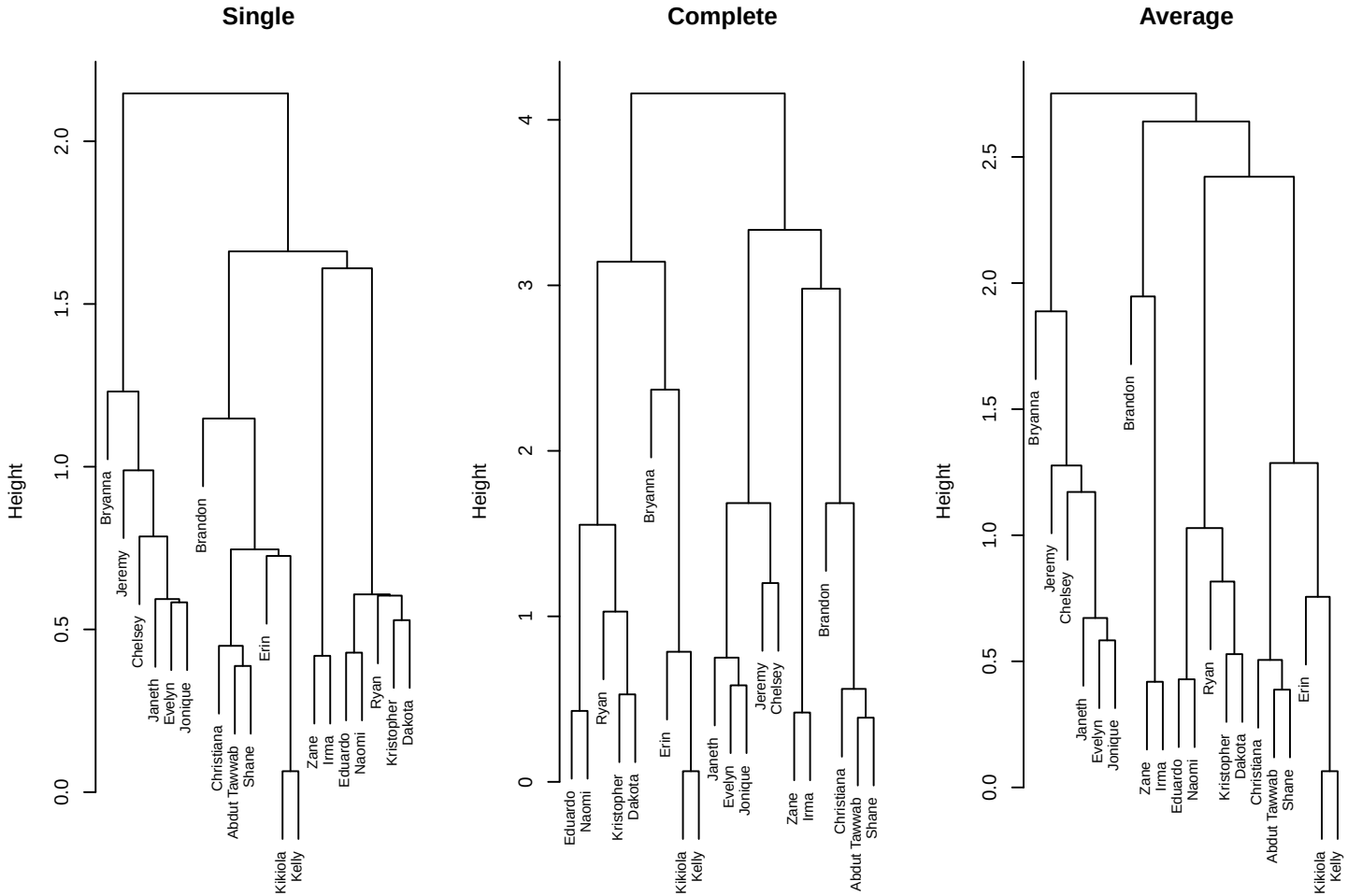


Comparing Linkages





Using Mahalanobis Distance



The Agglomeration Coefficient

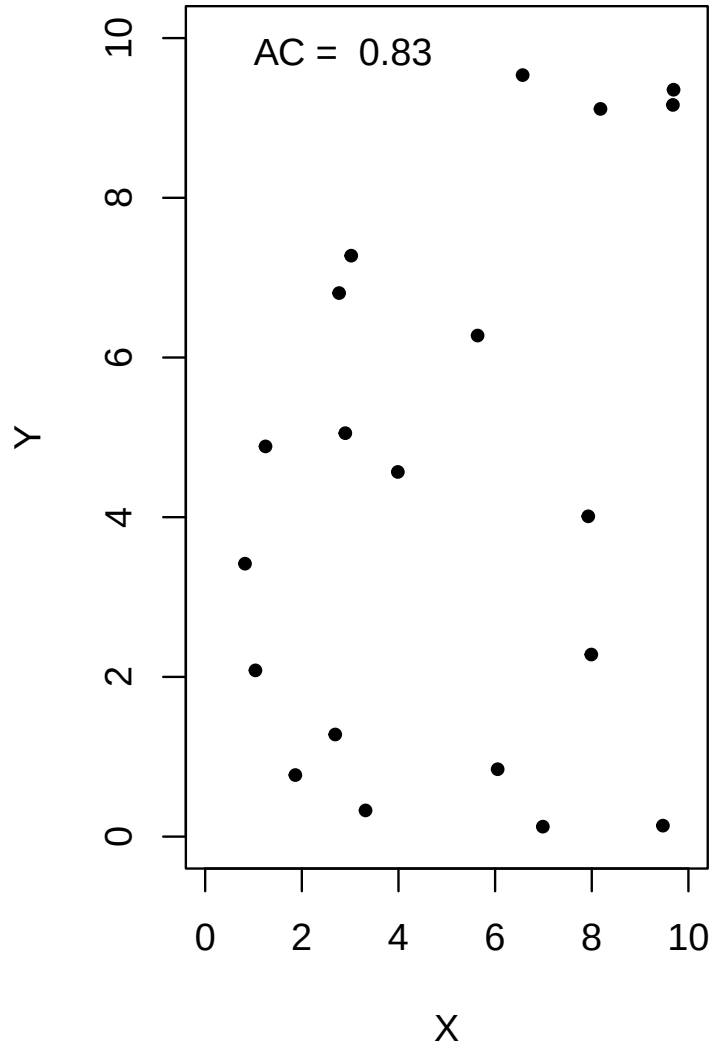
The *agglomeration coefficient* AC measures the clustering structure of the data. For each observation i, define m_i as the dissimilarity of observation i with the first cluster with which it is merged, divided by the dissimilarity in the final iteration (i.e., the greatest dissimilarity). The coefficient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

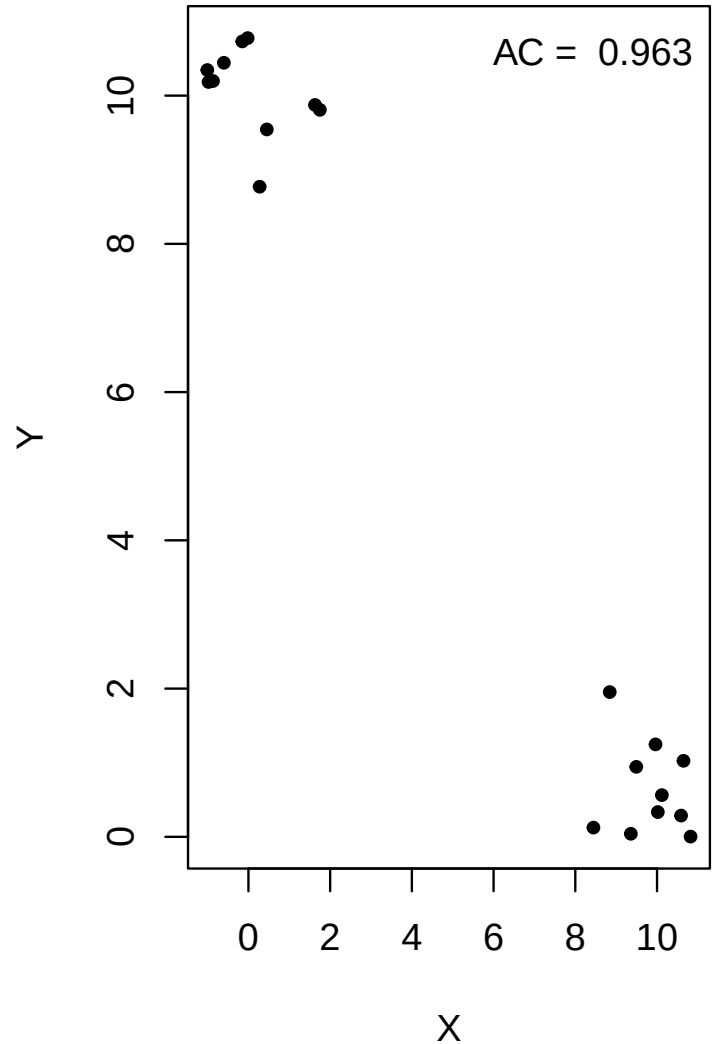
$$AC = \frac{1}{N-1} \sum_{i=1}^{N-1} 1 - m_i$$

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 *AC*s: Simulated Data

```
> Agnes.s <- agnes(ds, metric="euclidean",method="single")
> Agnes.s$ac
[1] 0.7689

> Agnes.c <- agnes(ds, metric="euclidean",method="complete")
> Agnes.c$ac
[1] 0.8446

> Agnes.a <- agnes(ds, metric="euclidean",method="average")
> Agnes.a$ac
[1] 0.7964

> # Using Mahalanobis distance:
> Agnes.M <- agnes(DM, diss=TRUE, method="average")
> Agnes.M$ac
[1] 0.7483
```

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'."...***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`)

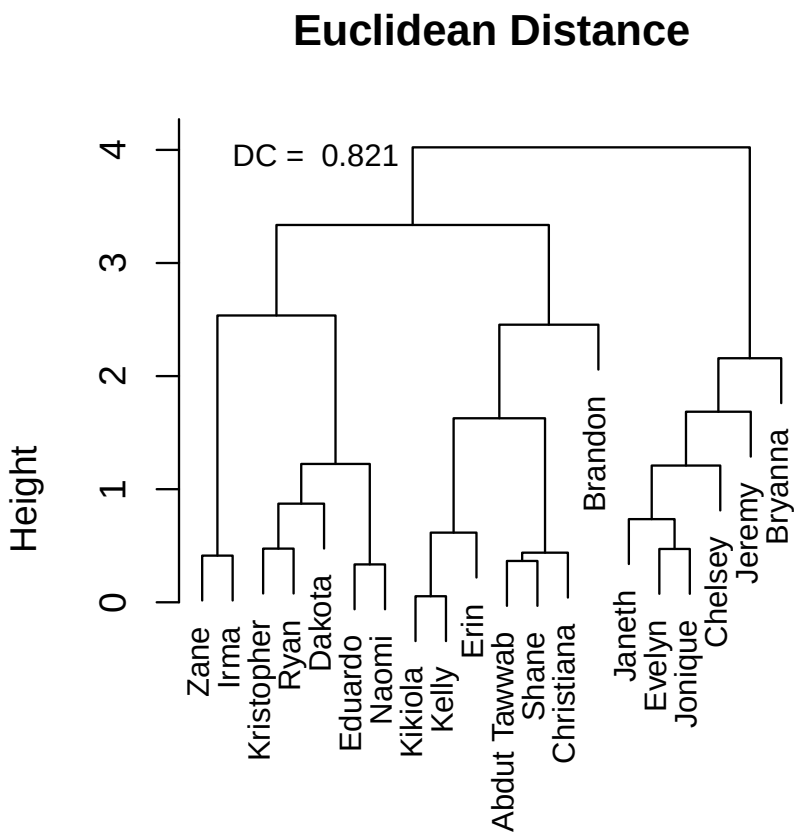
Basic steps:

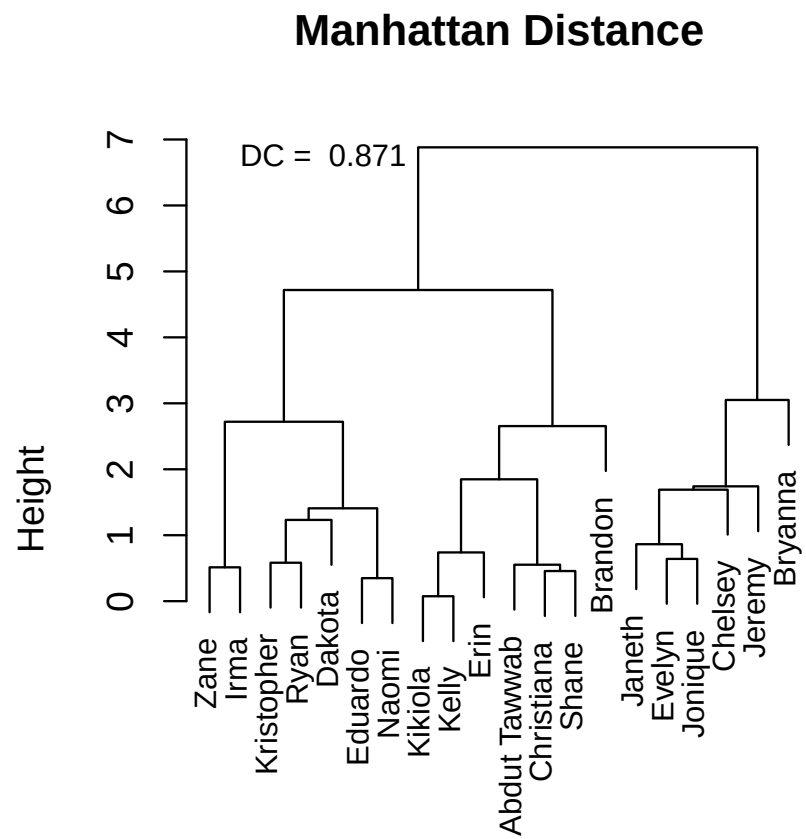
1. Begin with N observations on K variables in $\mathbf{X}$
2. Select the cluster C_{maxD} with the largest *diameter* (defined as the cluster with the largest dissimilarity between any two of its observations)
3. Select the observation j in C_{maxD} that has the highest average dissimilarity to the 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,]   -2   -3
 [2,]  -18  -19
 [3,]  -10  -13
 [4,]   -1   -9
 [5,]    3  -12
 [6,]  -14  -17
 [7,]   -6  -11
 [8,]    1  -20
 [9,]   -4    6
[10,]    7   -7
[11,]    9  -16
[12,]   10    2
[13,]    8    5
[14,]   11   -5
[15,]   14   -8
[16,]   13  -15
[17,]    4   12
[18,]   17   16
[19,]   18   15
Order of objects:
 [1] Zane         Irma         Kristopher   Ryan         Dakota       Eduardo
 [7] Naomi        Kikiola      Kelly        Erin         Abdut Tawwab Shane
[13] Christiana   Brandon      Janeth       Evelyn       Jonique      Chelsey
[19] Jeremy       Bryanna
Height:
 [1] 0.41261 2.53589 0.47441 0.87196 1.22318 0.33484 3.33616 0.05346 0.61569 1.62698
[11] 0.36533 0.43862 2.45437 4.02272 0.73516 0.47160 1.20894 1.68486 2.15781
Divisive coefficient:
[1] 0.8211
Available components:
[1] "order"     "height"    "dc"        "merge"     "diss"      "call"
[7] "order.lab" "data"
```

Divisive Clustering: Dendrograms





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:

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

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

- Requires the analyst to designate the number of clusters desired k *a priori*.

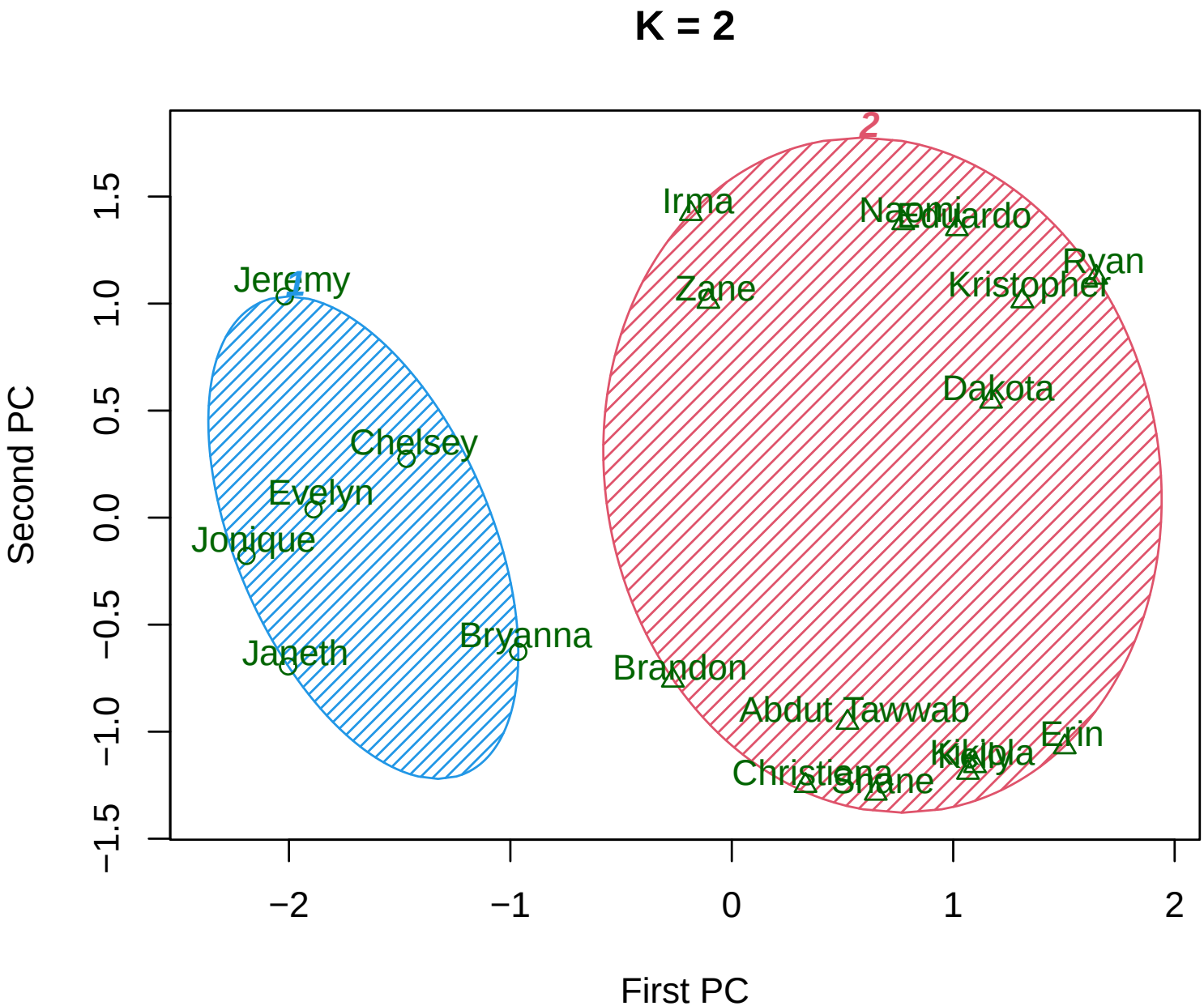
- Standard algorithm:

  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 6, 14

Cluster means:
         X       Y       Z
1 -0.03608  0.9309  1.4888
2  0.01546 -0.3990 -0.6381

Clustering vector:
        Zane      Kikiola        Kelly       Janeth       Jeremy   Kristopher
           2            2            2            1            1            2
      Dakota      Bryanna        Irma Abdut Tawwab         Ryan   Christiana
           2            1            2            2            2            2
       Shane       Evelyn      Brandon      Chelsey      Jonique      Eduardo
           2            1            2            1            1            2
       Naomi         Erin
           2            2

Within cluster sum of squares by cluster:
[1]  3.919 26.641
 (between_SS / total_SS =  46.4 %)

Available components:

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

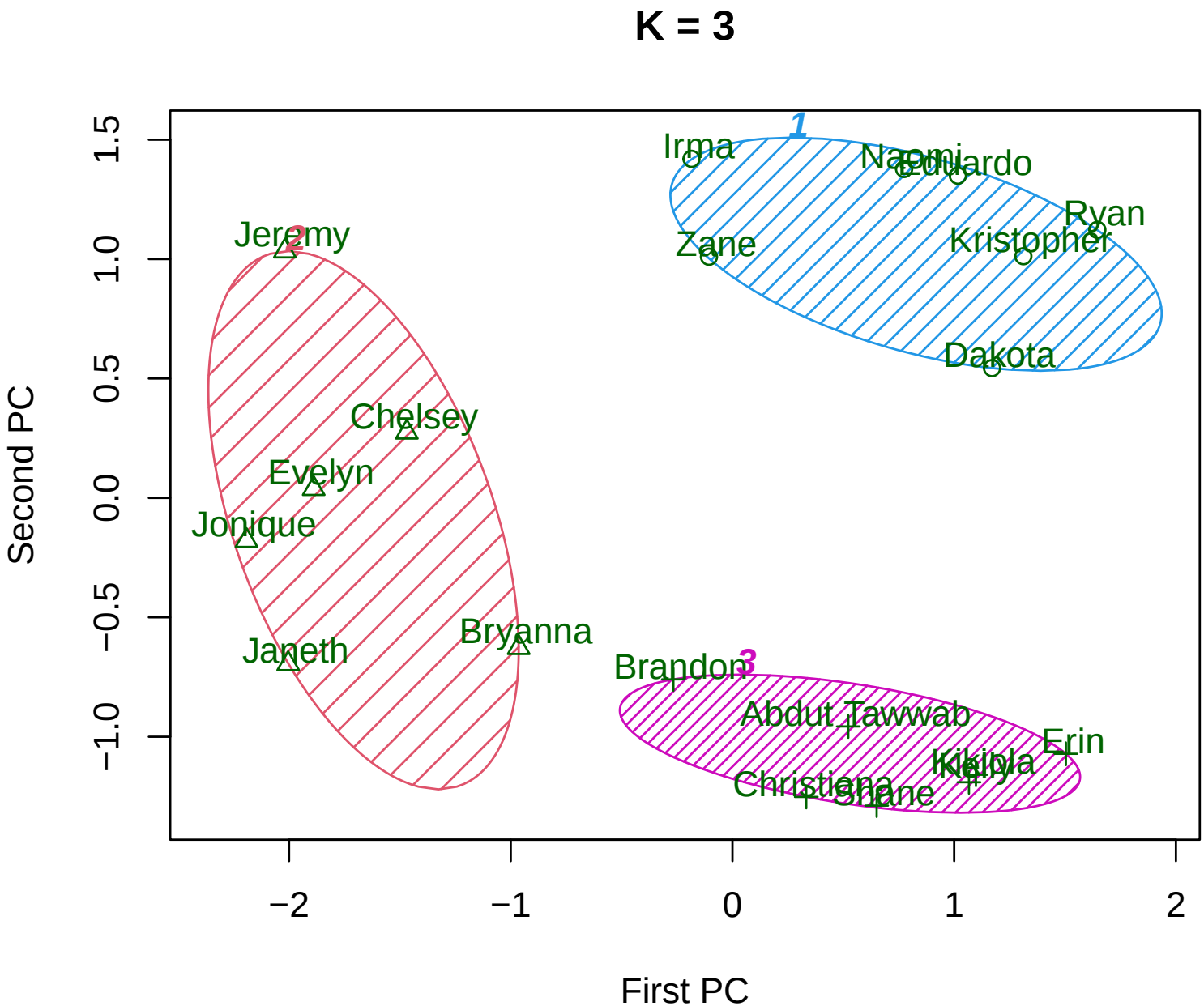
K-Means Clusters vs. Principal Components ($k = 2$)



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

```
> KM3 <- kmeans(ds,3)
> KM3
K-means clustering with 3 clusters of sizes 7, 6, 7

Cluster means:
         X       Y       Z
1  1.09826 -0.4463 -0.6381
2 -0.03608  0.9309  1.4888
3 -1.06734 -0.3516 -0.6381

Clustering vector:
        Zane      Kikiola        Kelly       Janeth       Jeremy   Kristopher
           1            3            3            2            2            1
      Dakota      Bryanna         Irma Abdut Tawwab         Ryan   Christiana
           1            2            1            3            1            3
       Shane       Evelyn      Brandon      Chelsey      Jonique      Eduardo
           3            2            3            2            2            1
       Naomi         Erin
           1            3

Within cluster sum of squares by cluster:
[1] 6.132 3.919 4.063
 (between_SS / total_SS =  75.2 %)

Available components:

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

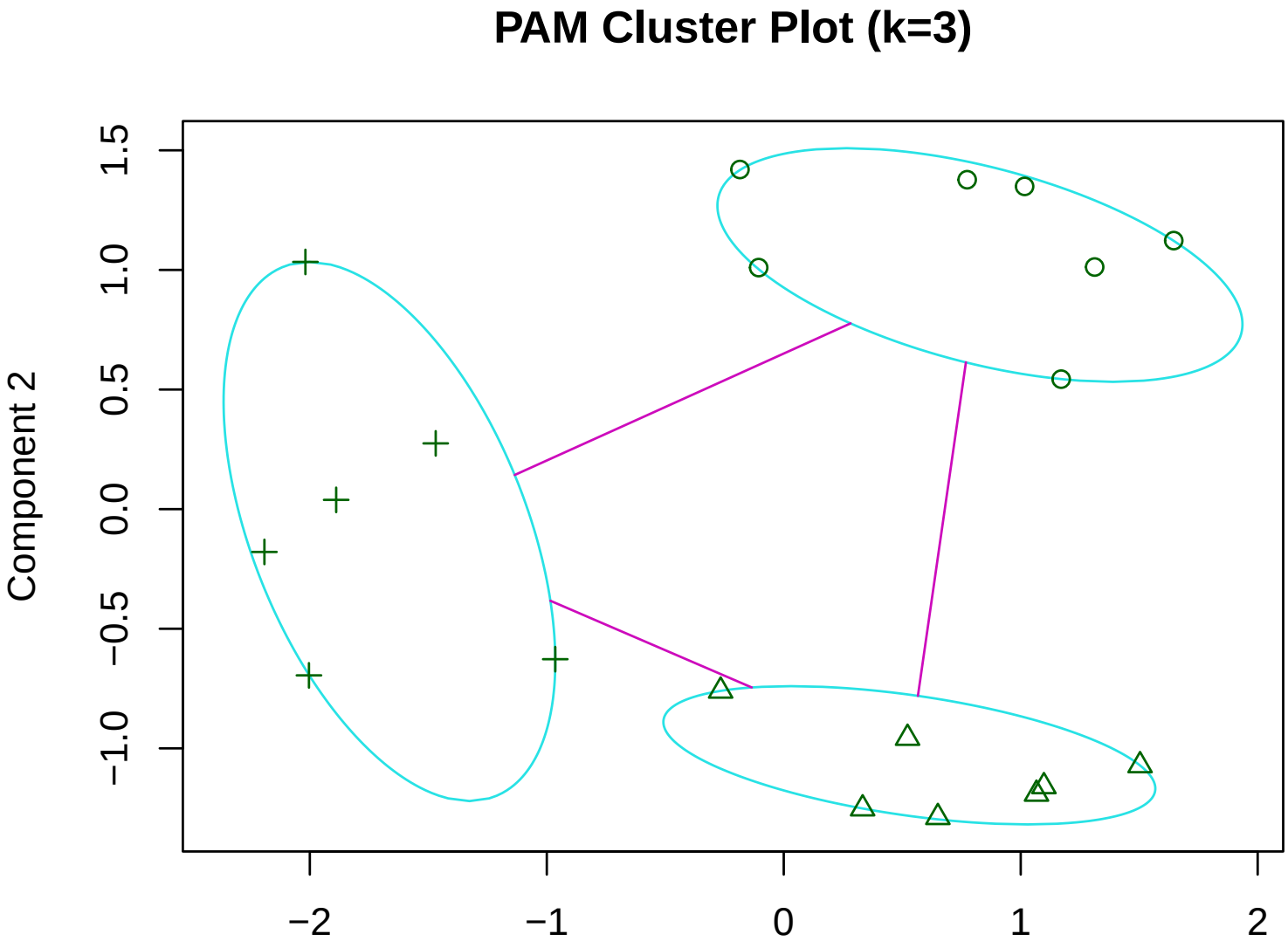
K-Means Clusters vs. Principal Components ($k = 3$)



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

```
> PAM3 <- pam(ds,3)
> PAM3
Medoids:
         ID        X       Y       Z
Eduardo 18  1.33382 -0.7341 -0.6381
Shane   13 -1.25654 -0.2858 -0.6381
Evelyn  14  0.01955  1.1153  1.4888
Clustering vector:
        Zane      Kikiola        Kelly       Janeth       Jeremy   Kristopher
           1            2            2            3            3            1
      Dakota      Bryanna         Irma Abdut Tawwab         Ryan   Christiana
           1            3            1            2            1            2
       Shane       Evelyn      Brandon      Chelsey      Jonique      Eduardo
           2            3            2            3            3            1
       Naomi         Erin
           1            2
Objective function:
 build   swap
0.7424 0.7307

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"  "clusinfo
 [7] "silinfo"    "diss"       "call"       "data"
```

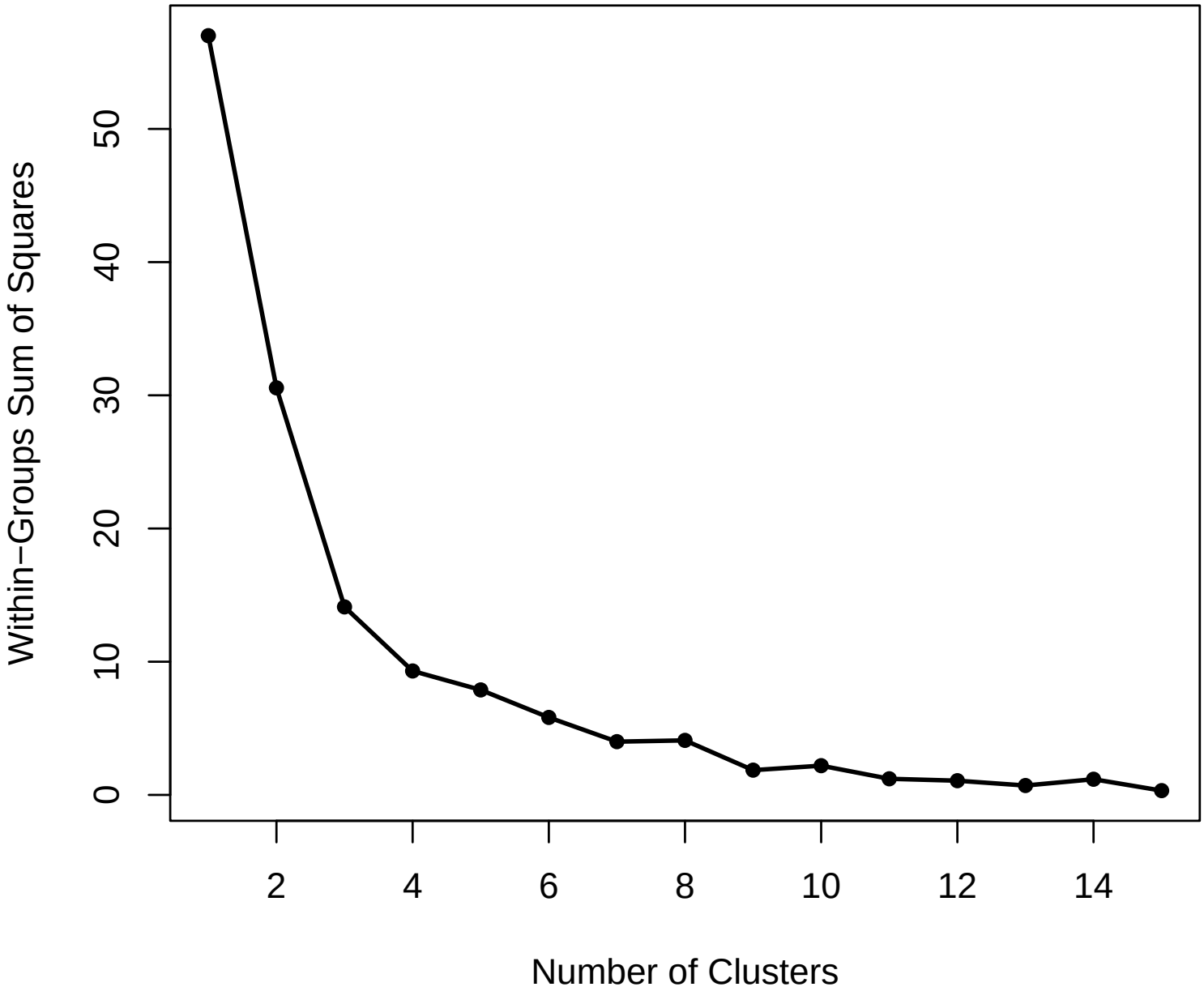
PAM, Illustrated



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

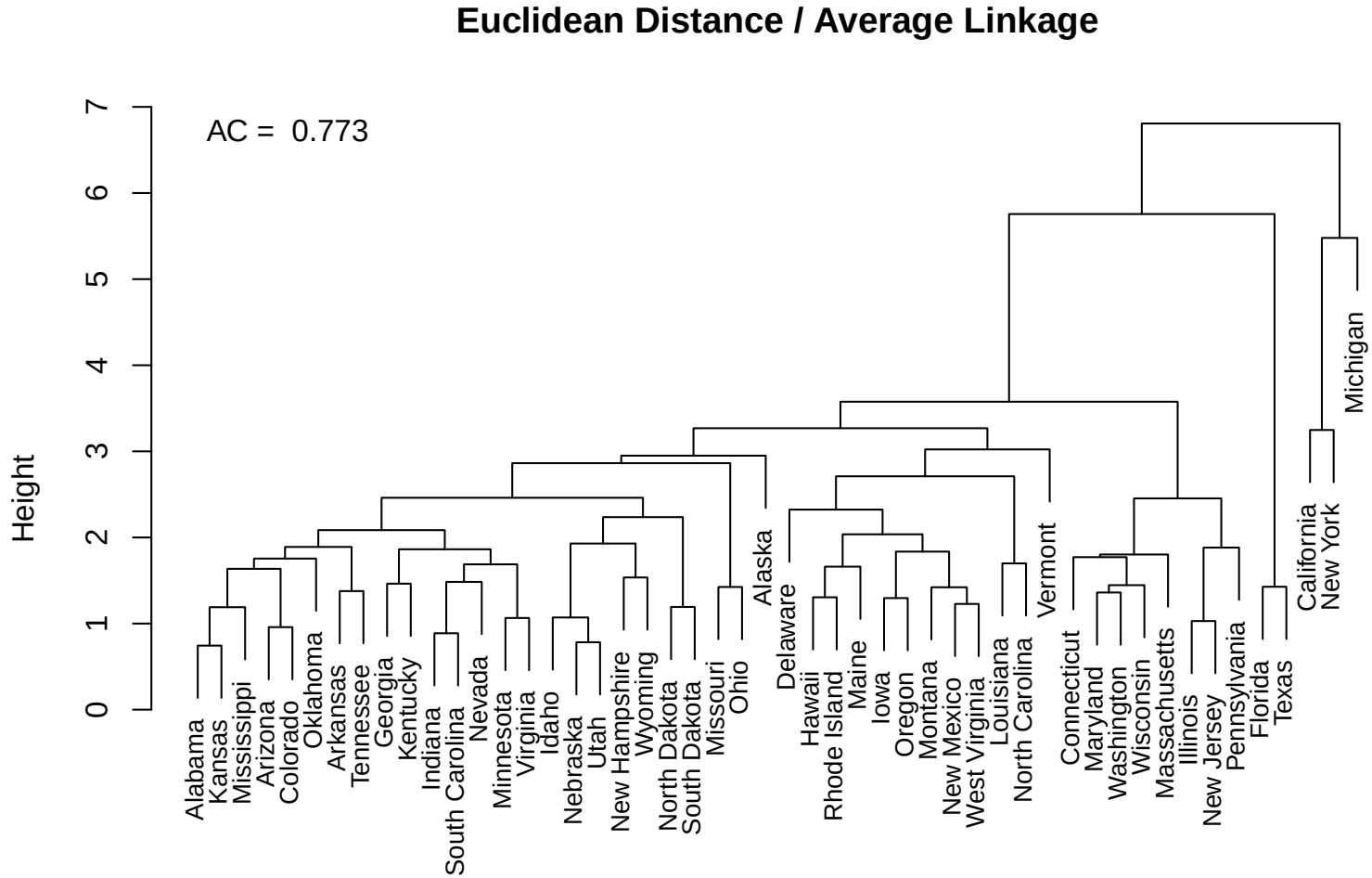
Mean-Shift Clustering

- Operationally similar to DBSCAN
- IME works well with "non-spherical" cluster shapes
- R packages: `meanShiftR`, `LPCM`, etc.

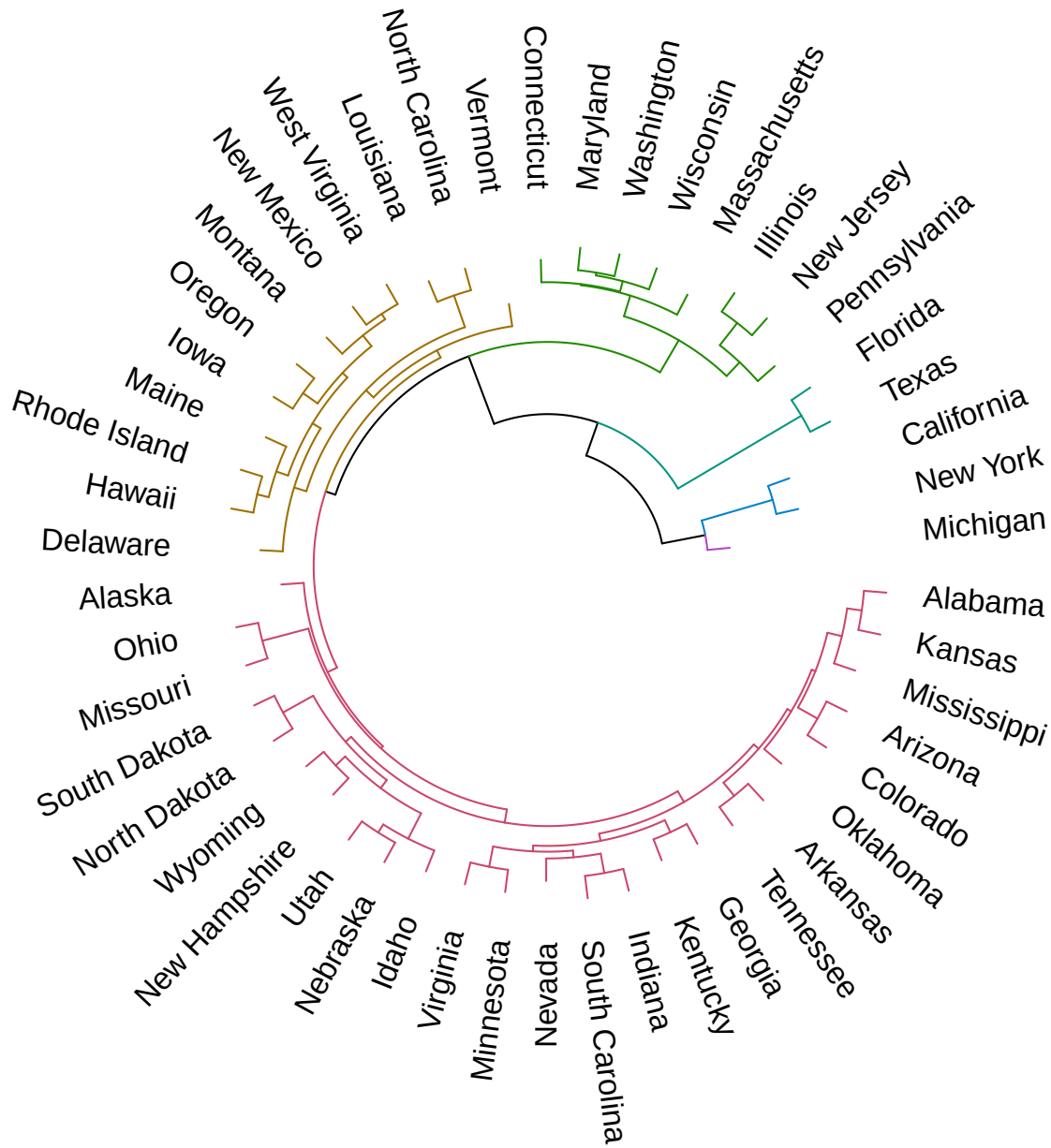
Real-Data Example: U.S. States

```
> url <- getURL("https://raw.githubusercontent.com/PrisonRodeo/
                 PLSC504-2024-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
```

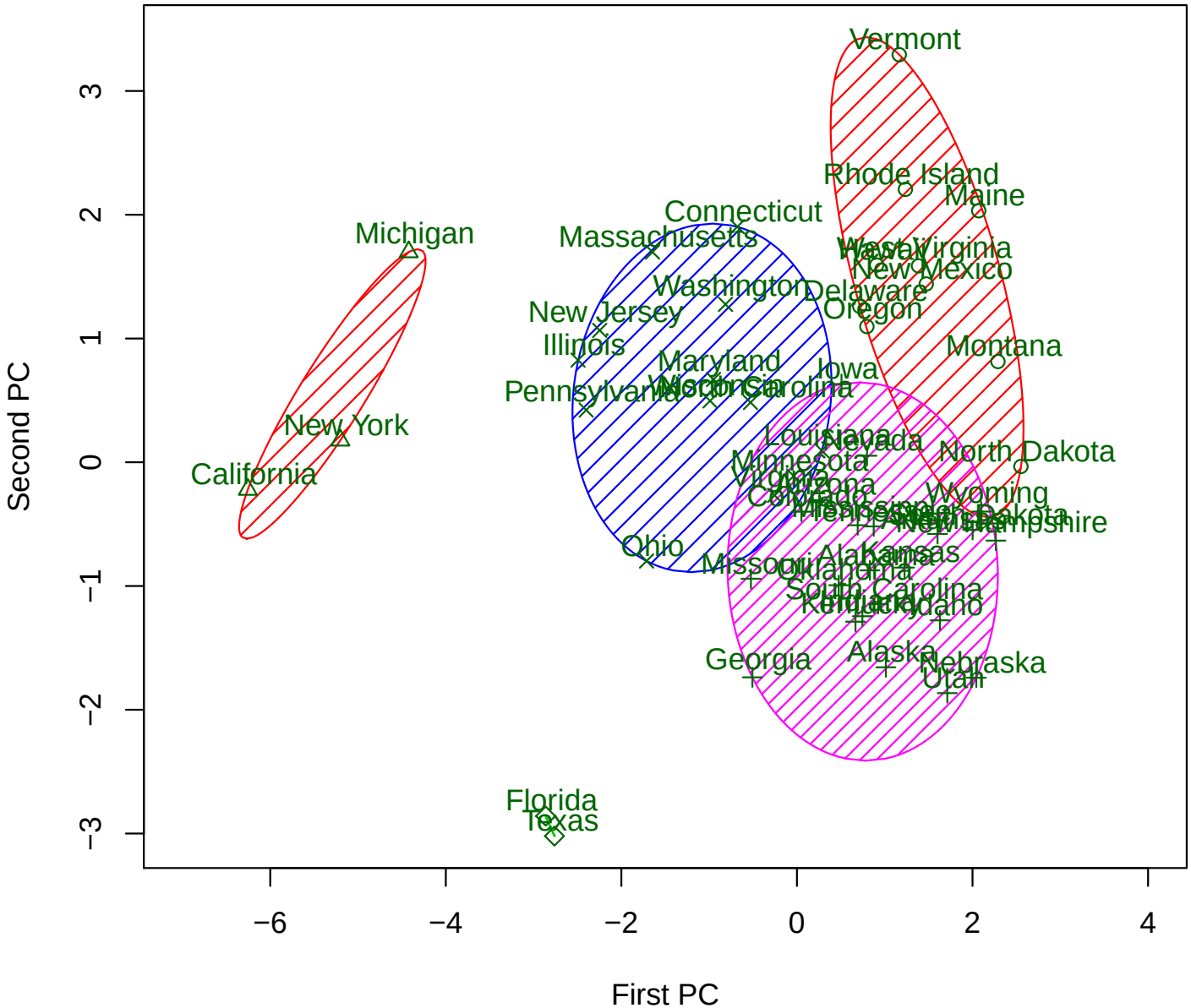
State Data: Agglomerative Dendrogram



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

A partial list:

- `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.

Useful / Interesting Links

- The Cluster Analysis R Task View: https://cran.r-project.org/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
- For you machine learning types: Christopher Molnar, on The Intricate Link Between Compression and Prediction, or, "How GZip + K-Means Outperforms BERT"