Multivariate: Principal components analysis

Table of contents

1 Goals

1.1 Goals

1.1.1 Goals of this lecture

- Principal components analysis (PCA)
  - **Dimension reduction**: reduce number of variables
- A **large set of (potentially correlated) observed variables**
  - Organize the **variance** in those variables to a **smaller set** of orthogonal (uncorrelated) variables

2 Statistical measurement

2.1 Statistical measurement

2.1.1 Measuring things is hard

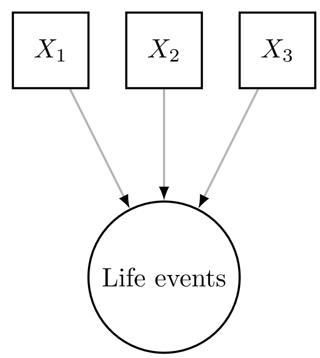
- Psychology: we **cannot directly measure** some constructs
  - No ruler to measure "intelligence" or "introversion"
- We can **indirectly** measure what we really want to measure
  - Want to measure **intelligence**
    - Math ability, verbal ability, spatial ability, reasoning, general knowledge, etc.
  - Intelligence is a **latent variable**
    - Not *directly* observed

2.1.2 Two ways to think about latent variables

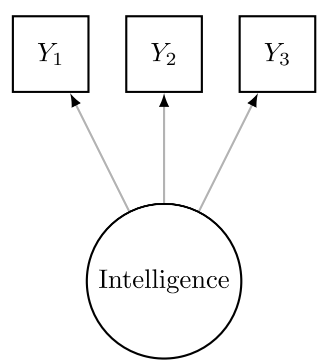
1. Latent variable is a **result** of item responses
   - *Formative* latent variable
   - Principal components analysis (PCA)
   - This week
2. Latent variable **causes** item responses
   - *Reflective* latent variable
   - Factor analysis (FA)
   - Next week (and most of what you'll do)

2.1.3 Formative vs reflective latent variables

- Formative factor



- Reflective factor



2.1.4 Latent variables as dimension reduction

- In each of these examples
  - 3 observed variables and 1 latent variable
- But you can have **many more** observed variables
  - As many measures of the latent variable as you have
- Often more than 1 latent variable
  - Number of latent variables < number of observed variables
    - **Dimension reduction**

3 Super quick review

3.1 Eigenvectors and eigenvalues

3.1.1 Eigenvectors and eigenvalues

- Eigenvectors / values are the solution to **homogenous equations**
  - $[\mathbf{A} - \lambda \mathbf{I}]\nu = 0$
  - λ (lambda) is the eigenvalues, ν (nu) is the eigenvectors
- **Maximize** a function while also imposing some **constraints**
  - In the case of PCA
  - **Maximize** the **variance** (1st eigenvalue is largest)
  - **Constrain** eigenvectors to be **orthogonal**

3.1.2 Eigenvectors

- Eigenvectors are created from a matrix (such as $\mathbf{R}_{XX}$)
  - Form basis or reference axes for that matrix
  - All mutually *orthogonal*
- If matrix is full rank
  - As many eigenvectors as variables (from a corr or cov matrix)
    - p variables means p eigenvalues and eigenvectors
    - 5 variables means 5 eigenvalues and eigenvectors

3.1.3 Eigenvalues

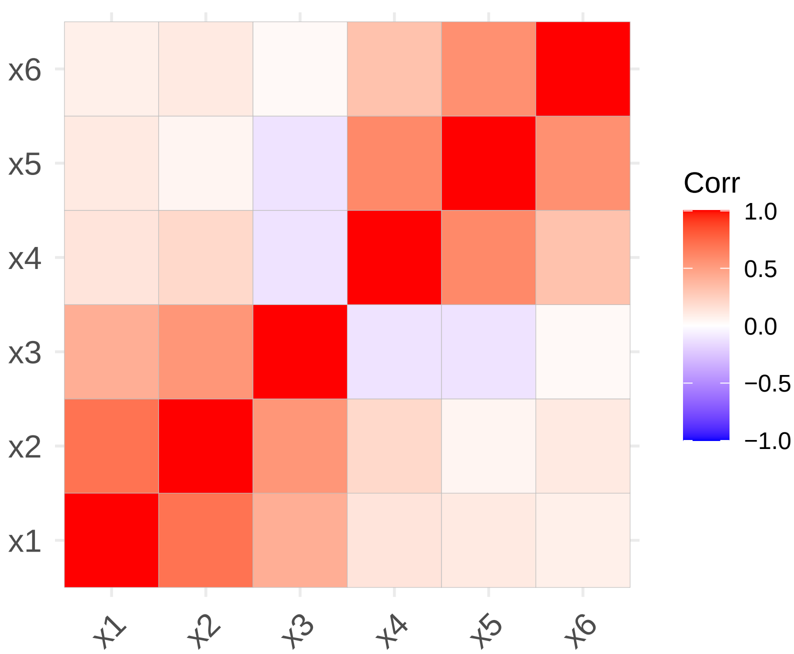
- One eigenvalue for each eigenvector
  - How much **variance** associated with that eigenvector
  - First eigenvector has the highest eigenvalue, then decreases
- Sum of eigenvalues for a matrix = sum of diagonal elements
  - 5×5 correlation matrix $\rightarrow$ eigenvalues add to 5

4 Data Example

4.1 Measure and variables

4.1.1 Simulated data

- Data from last week's class
  - 100 subjects
  - 6 continuous variables
- Color-coded correlation matrix



4.1.2 Observed and latent variables

- Observed variables
  - 6 variables
  - **These are all X variables**: they predict the latent variable
- Latent variables
  - These are the Y variables
  - They are the **components** (P**C**A)
  - We *create* them in the analysis

4.2 Output of the analysis

4.2.1 Data reduction

- The idea behind PCA is to reduce the number of variables
  - Start with **6 items**
    - Want **fewer** than 6 components
    - How many fewer?
- I simulated the data to have 2 "clumps"
  - We talked about this last week
  - So I'll show you a **2 component model** *to start*

4.2.2 PCA results

1. Loadings
   - Relation between observed variable (X) and component (Y)
     - Matrix with rows = # items, columns = # components
     - High loading = that X is highly related to that Y
   - Think: correlation or standardized regression coefficient
     - Range from -1 to 1

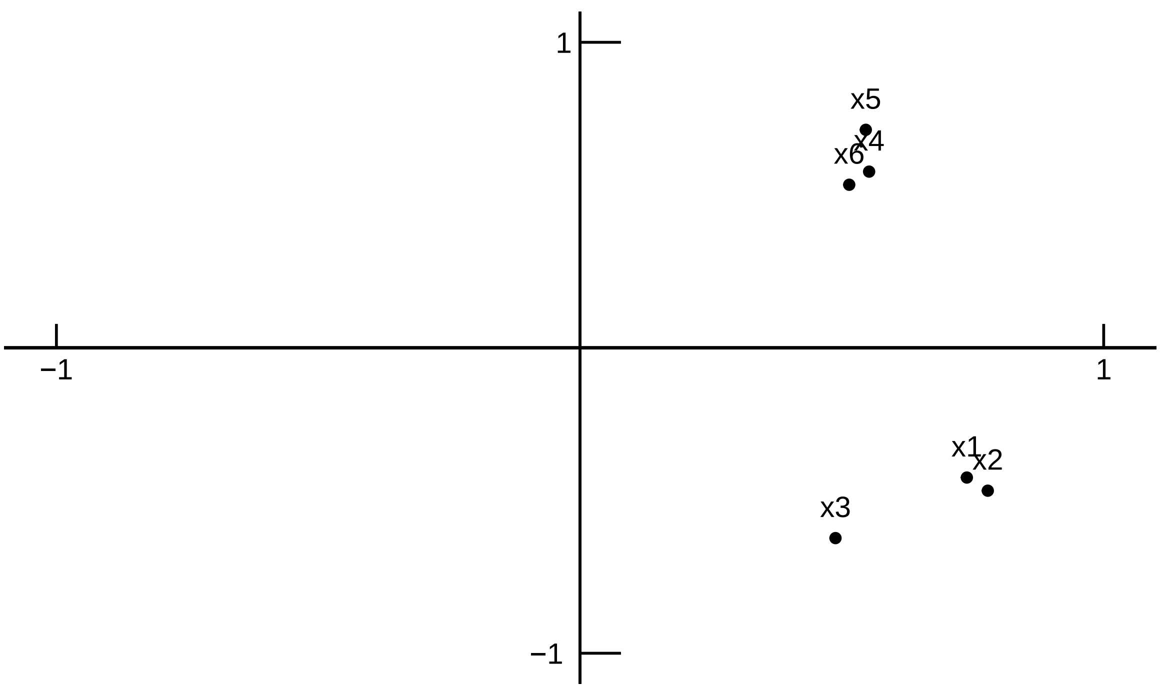
4.2.3 Model results: Loadings in R

```
Loadings:
   PC1    PC2
x1  0.739 -0.425
x2  0.779 -0.468
x3  0.488 -0.623
x4  0.552  0.577
x5  0.546  0.714
x6  0.514  0.534

                 PC1   PC2
SS loadings    2.257 1.914
Proportion Var 0.376 0.319
Cumulative Var 0.376 0.695
```

4.2.4 Model results: Loadings in SPSS

4.2.5 Loadings



Component Matrix[a]

	Component	
	1	2
x1	.739	-.425
x2	.779	-.468
x3	.488	-.623
x4	.552	.577
x5	.546	.714
x6	.514	.534

Extraction Method: Principal Component Analysis.

a. 2 components extracted.

Total Variance Explained

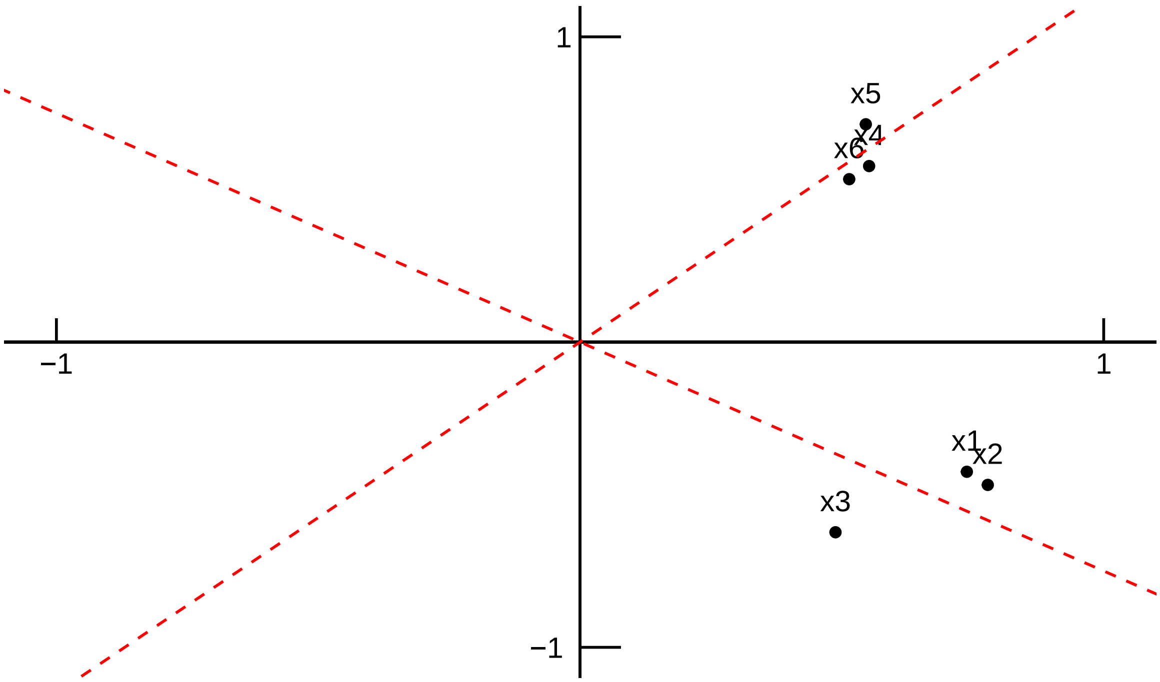
	Initial Eigenvalues			Extraction Sums of Squared Loadings		
Component	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	2.257	37.610	37.610	2.257	37.610	37.610
2	1.914	31.904	69.514	1.914	31.904	69.514
3	.751	12.517	82.031			
4	.496	8.273	90.303			
5	.348	5.804	96.108			
6	.234	3.892	100.000			

Extraction Method: Principal Component Analysis.

4.2.6 Simple structure and rotation

- Solution has **simple structure** if each item has **high loadings** on only one component and **near zero loadings** on all other components
  - i.e., points are near the axes
  - Easier to interpret: items only relate to one axis
- **Rotated solution** rotates the axes to get closer to *simple structure*
  - We'll look at some different ways to rotate the solution
    * I'll show you a conceptual version now
  - Easier to interpret a solution that has simple structure

4.2.7 Loadings on rotated axes



4.2.8 PCA results

2. Communalities
   - Remember that we don't retain all the components
   - Communalities are the proportion of variance in X that's reproduced **by the components (Y) that you do retain**

- Think: $R^2_{multiple}$ for Ys predicting Xs
    - This is weird, right? Yeah, I'll explain more

4.2.9 Model results: Communalities in R

```
       x1        x2        x3        x4        x5        x6
0.7261219 0.8252297 0.6261259 0.6372285 0.8070047 0.5491167
```

4.2.10 Model results: Communalities in SPSS

Communalities

	Initial	Extraction
x1	1.000	.726
x2	1.000	.825
x3	1.000	.626
x4	1.000	.637
x5	1.000	.807
x6	1.000	.549

Extraction Method: Principal Component Analysis.

4.2.11 PCA overview

- **Loadings** tell us *how items are correlated with components*
    - Simple structure makes loadings more interpretable

- **Communalities** tell us how much *variance* in the items is *explained* by *the components we kept*
- But where did the Ys / components even come from?

5 PCA details

5.1 PCA process

5.1.1 Step 1: Correlation matrix

- PCA starts by calculating the correlation matrix

$$\mathbf{R}_{XX} = \begin{bmatrix} 1 & r_{X_1X_2} & r_{X_1X_3} & r_{X_1X_4} & r_{X_1X_5} & r_{X_1X_6} \\ r_{X_2X_1} & 1 & r_{X_2X_3} & r_{X_2X_4} & r_{X_2X_5} & r_{X_2X_6} \\ r_{X_3X_1} & r_{X_3X_2} & 1 & r_{X_3X_4} & r_{X_3X_5} & r_{X_3X_6} \\ r_{X_4X_1} & r_{X_4X_2} & r_{X_4X_3} & 1 & r_{X_4X_5} & r_{X_4X_6} \\ r_{X_5X_1} & r_{X_5X_2} & r_{X_5X_3} & r_{X_5X_4} & 1 & r_{X_5X_6} \\ r_{X_6X_1} & r_{X_6X_2} & r_{X_6X_3} & r_{X_6X_4} & r_{X_6X_5} & 1 \end{bmatrix}$$

5.1.2 Step 1: Correlation matrix

- PCA starts by calculating the correlation matrix

	x1	x2	x3	x4	x5	x6
x1	1.0000	0.7041	0.4157	0.1406	0.1058	0.0814
x2	0.7041	1.0000	0.5428	0.1963	0.0538	0.1087
x3	0.4157	0.5428	1.0000	-0.1208	-0.1177	0.0276
x4	0.1406	0.1963	-0.1208	1.0000	0.6027	0.3249
x5	0.1058	0.0538	-0.1177	0.6027	1.0000	0.5651
x6	0.0814	0.1087	0.0276	0.3249	0.5651	1.0000

5.1.3 Step 2: Eigenvalues and eigenvectors

- **Eigenvalues** of correlation matrix
  - We're not going to do anything with these right now

```
[1] 2.2566146 1.9142128 0.7510163 0.4963613 0.3482518 0.2335431
```

- **Eigenvectors** of correlation matrix: $p \times r$ matrix

- Each **column** is an eigenvector / axis

```
           [,1]       [,2]       [,3]        [,4]       [,5]       [,6]
[1,] -0.4917076  0.3070964  0.2458906  0.54136266 -0.3027305 -0.4676901
[2,] -0.5183409  0.3381863  0.1510558  0.05919316  0.3956537  0.6588544
[3,] -0.3247308  0.4503119 -0.4335595 -0.64643283 -0.2133135 -0.2010403
[4,] -0.3674707 -0.4167786  0.5131075 -0.44899215  0.3249147 -0.3475890
[5,] -0.3632801 -0.5157586 -0.0382021 -0.05343548 -0.6660796  0.3924843
[6,] -0.3421828 -0.3857845 -0.6811809  0.28477700  0.3963310 -0.1785989
```

5.1.4 Step 3: Create latent Y variables

- The matrix of eigenvectors is $\mathbf{A}$
  - If matrix not full rank, fewer columns

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & a_{66} \end{bmatrix}$$

5.1.5 Step 3: Create latent Y variables

$$\underset{(n,r)}{\mathbf{Y}} = \underset{(n,p)}{\mathbf{X}} \underset{(p,r)}{\mathbf{A}}$$

- In this example
  - 100 subjects ($n = 100$)
  - Correlation matrix is full rank so $p = r = 6$
- $\mathbf{Y}$ has 100 rows and 6 columns

5.1.6 Step 3: Create latent Y variables

$$\underset{(n,r)}{\mathbf{Y}} = \underset{(n,p)}{\mathbf{X}} \underset{(p,r)}{\mathbf{A}}$$

- Each person now has
  - 6 X values (specific to each person)

  - 6 Y values (specific to each person)
  - Same values of $\mathbf{A}$: these are **weights** (like in linear regression, same weights for everyone)

5.1.7 Step 3: Create latent Y variables

- Y variables are **linear combinations** of Xs and $\mathbf{A}$
  - Each Y is an $n \times 1$ vector
- First Y variable: $\underline{Y}_1 = a_{11}\underline{X}_1 + a_{21}\underline{X}_2 + a_{31}\underline{X}_3 + a_{41}\underline{X}_4 + a_{51}\underline{X}_5 + a_{61}\underline{X}_6$
- Second Y variable: $\underline{Y}_2 = a_{12}\underline{X}_1 + a_{22}\underline{X}_2 + a_{32}\underline{X}_3 + a_{42}\underline{X}_4 + a_{52}\underline{X}_5 + a_{62}\underline{X}_6$
- Looks like a regression, but note that it's not $\hat{Y}$ and there's no $+e$

5.1.8 Step 4: Use orthogonal Ys to predict original Xs

$$\underset{(n,p)}{\mathbf{X}} = \underset{(n,r)}{\mathbf{Y}}\underset{(r,p)}{\mathbf{B}}$$

- Ys are **orthogonal**
  - Now use them as (uncorrelated) predictors to predict Xs
- $\mathbf{B}$ is the (unrotated) matrix of loadings
  - Rows = components, columns = items

5.1.9 Step 4: Use orthogonal Ys to predict original Xs

$$\underset{(n,p)}{\mathbf{X}} = \underset{(n,r)}{\mathbf{Y}}\underset{(r,p)}{\mathbf{B}}$$

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} \end{bmatrix}$$

5.1.10 Four things about the loadings matrix

- In practice, it will have **fewer rows**
  - We don't retain all the components (e.g., 2 in this example)
- Unlike a lot of matrices we look at
  - **All elements are unique** ($b_{21} \neq b_{12}$)
- In software, the **transpose** of this matrix is given
  - Rows = items, columns = components
- Think of them like **standardized regression coefficients**
  - But since Y are orthogonal, they're **not partial coefficients**

5.1.11 One thing about communalities

- Communalities are the proportion of variance in X that's reproduced **by the components (Y) that you do retain**
  - Think: $R^2_{multiple}$ for Ys predicting Xs
  - But why Y predicting X? That's backward!
- We don't do a perfect job re-creating the information from p variables using fewer than p components
  - How much variance in Xs did we **retain** with the Ys that we **retained**?

6 How many components?

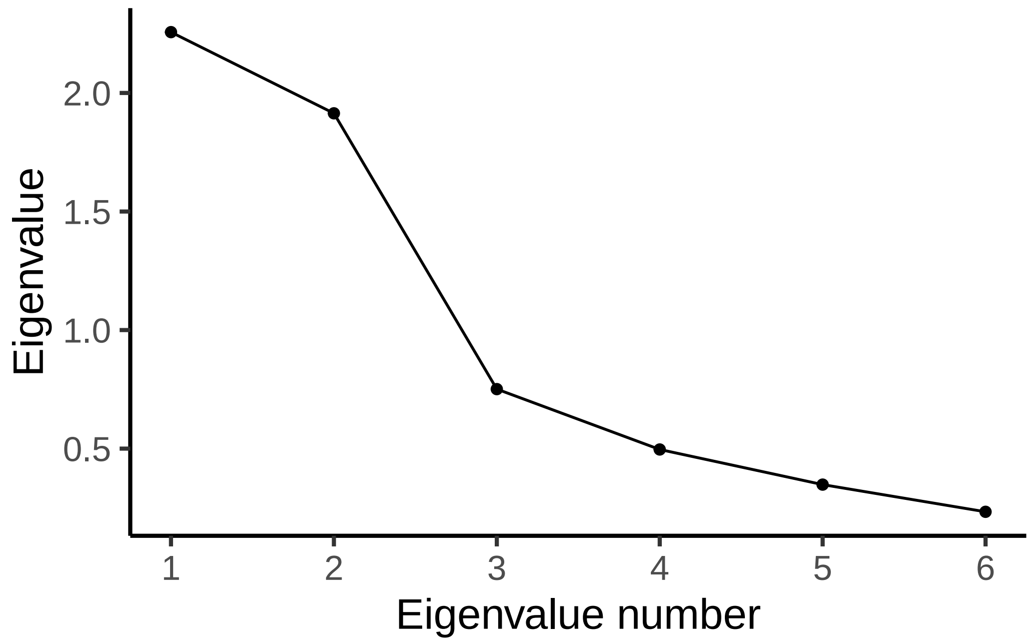
6.1 How many components?

6.1.1 How many components?

- The main objective of PCA is to **reduce the number of variables**
  - Have p X variables
  - Want to be able to describe them with **fewer** than p Y variables
- There are several methods to choose
  - Often give different results

6.2 Scree plot

6.2.1 Scree plot



6.2.2 Scree plot

- First component accounts for the most variance
  - Second component accounts for less, third for even less, etc.
- At what point does adding more components not help account for more variance?
  - Look for "drop" in the scree plot
  - Somewhat arbitrary, can be difficult to determine

6.3 Kaiser criteria

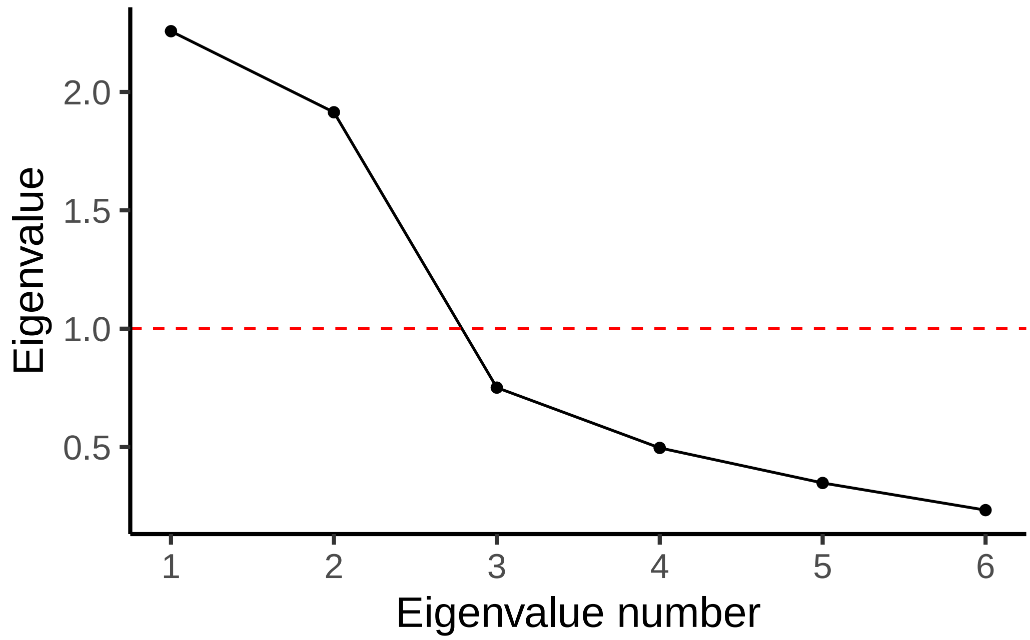
6.3.1 Kaiser criteria: Don't use this

- Also called "eigenvalues greater than 1" criteria
  - With PCA, you're dealing with the **correlation matrix**
  - Diagonals are all 1s
  - If each component accounts for "its share" of the variance

        * Then all eigenvalues are 1
        * Components with eigenvalue > 1 are doing better than that

- Tends to over-extract (too many components)

6.3.2 Kaiser criteria



6.4 Proportion of variance

6.4.1 Proportion of variance accounted for

- Keep any component that accounts for more than a certain percentage of variance
    - Must choose some arbitrary percentage
    - Not commonly used in psychology
        * More commonly used in engineering

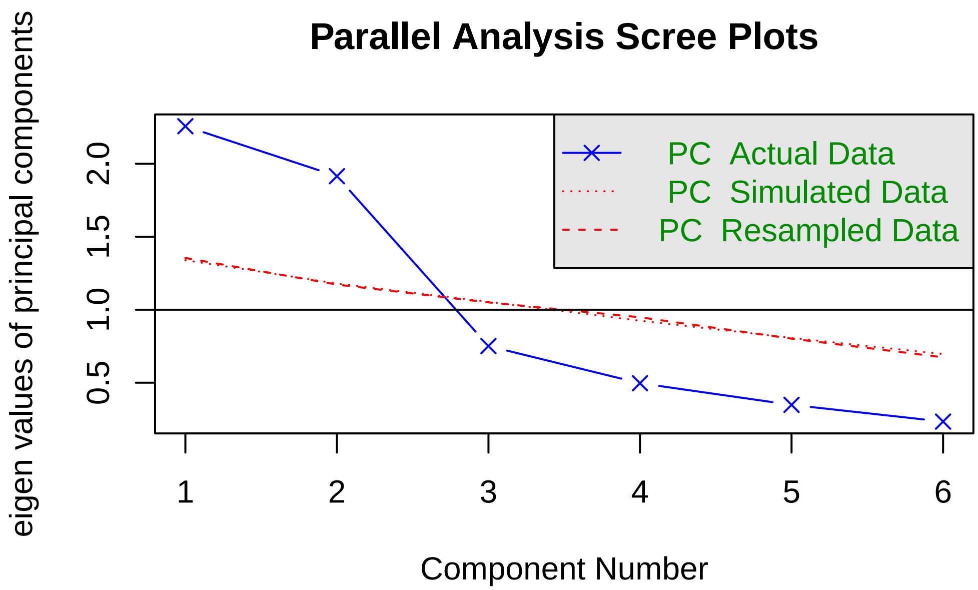
6.5 Parallel analysis

6.5.1 Parallel analysis

- Simulation based method

- Generate **random correlation matrices** with same p and n as data
    - Two ways: new simulated data or re-sample from your data
    - Estimate the eigenvalues from these random correlation matrices
    - Retain components with eigenvalues higher than (default) 95%ile of the random values

6.5.2 Parallel analysis in R




```
Parallel analysis suggests that the number of factors =  NA  and the number of components =
```

6.5.3 Parallel analysis in SPSS

- Requires some external scripts with lots of those `MATRIX` statements
    - Brian O'Connor's website
    - Youtube video explaining
    - Hayton, J. C., Allen, D. G., & Scarpello, V. (2004). Factor retention decisions in exploratory factor analysis: A tutorial on parallel analysis. *Organizational research methods, 7(2)*, 191-205.

```
Run MATRIX procedure:

PARALLEL ANALYSIS:

Principal Components

Specifications for this Run:
Ncases       100
Nvars          6
Ndatsets    1000
Percent       95

Random Data Eigenvalues
         Root         Means      Prcntyle
     1.000000      1.342755      1.494631
     2.000000      1.173098      1.272529
     3.000000      1.043653      1.115745
     4.000000       .932429      1.005241
     5.000000       .819450       .903491
     6.000000       .688615       .788492

------ END MATRIX -----
```

Total Variance Explained						
	Initial Eigenvalues			Extraction Sums of Squared Loadings		
Component	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %
1	2.257	37.610	37.610	2.257	37.610	37.610
2	1.914	31.904	69.514	1.914	31.904	69.514
3	.751	12.517	82.031			
4	.496	8.273	90.303			
5	.348	5.804	96.108			
6	.234	3.892	100.000			

Extraction Method: Principal Component Analysis.

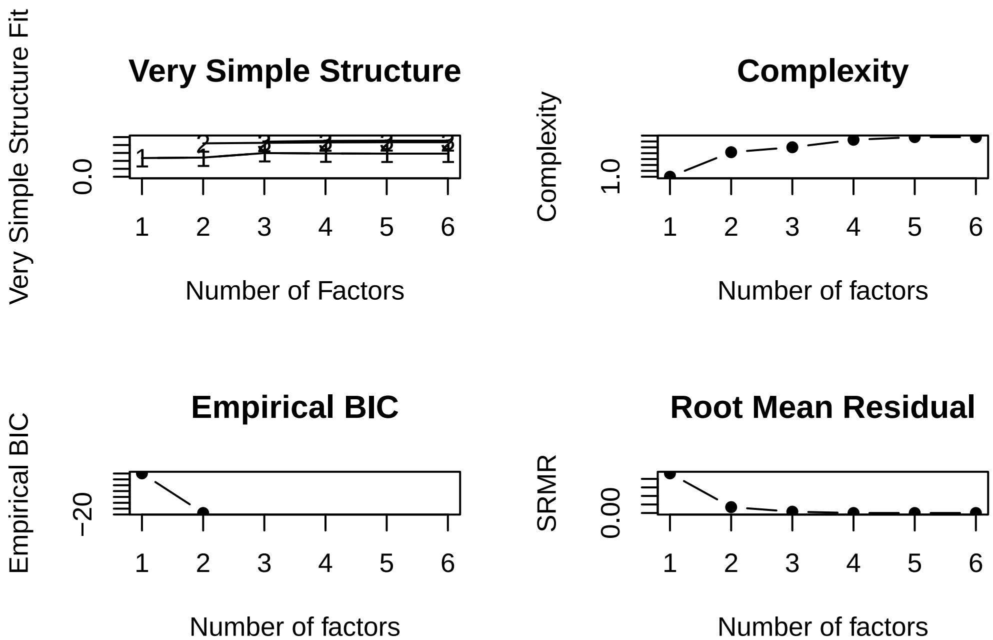
6.5.4 Parallel analysis in SPSS

6.6 MAP

6.6.1 Minimum average partials (MAP)

- Look at "partialed" correlation matrix after each component
  - First component accounts for the most variance
    - After the first component is partialled out, correlations between variables should be smaller
  - Second component account for the next most variance
    - After the second component is partialled out, correlations between variables should be smaller, etc
  - You have **enough components** when average partial correlation is **minimized**

6.6.2 MAP test in R




```
Number of factors
Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
    n.obs = n.obs, plot = FALSE, title = title, use = use, cor = cor)
VSS complexity 1 achieves a maximimum of 0.6  with  3  factors
VSS complexity 2 achieves a maximimum of 0.87  with  5  factors
The Velicer MAP achieves a minimum of 0.12  with  2  factors
Empirical BIC achieves a minimum of  -14.87  with  2  factors
Sample Size adjusted BIC achieves a minimum of  1.77  with  2  factors

Statistics by number of factors
  vss1 vss2  map dof   chisq    prob sqresid  fit RMSEA BIC SABIC complex
1 0.47 0.00 0.20   9 9.3e+01 4.1e-16     5.2 0.47 0.305  52  79.9     1.0
2 0.48 0.84 0.12   4 7.6e+00 1.1e-01     1.5 0.84 0.094 -11   1.8     1.8
3 0.60 0.85 0.23   0 8.3e-01      NA     1.1 0.88    NA  NA    NA     2.0
4 0.59 0.87 0.43  -3 6.2e-09      NA     0.9 0.91    NA  NA    NA     2.3
5 0.58 0.87 1.00  -5 0.0e+00      NA     0.8 0.92    NA  NA    NA     2.3
6 0.58 0.87   NA  -6 0.0e+00      NA     0.8 0.92    NA  NA    NA     2.3
   eChisq    SRMR eCRMS eBIC
1 1.6e+02 2.3e-01 0.300  121
2 3.6e+00 3.4e-02 0.067  -15
3 1.8e-01 7.8e-03    NA   NA
4 1.0e-09 5.8e-07    NA   NA
```

```
5 5.4e-16 4.2e-10    NA   NA
6 5.4e-16 4.2e-10    NA   NA
```

6.6.3 MAP test in SPSS

- See resources for parallel analysis
  - Those include Velicer's MAP test

6.7 Solution makes sense

6.7.1 Solution makes sense (theoretically)

- Do the components make sense?
  - Does it make sense for the items that load highly on each component to belong together?
- Don't use this as your only criterion
  - This is what makes this science
  - Not just a computer spitting out numbers

6.8 Summary of number of components

6.8.1 Summary of choosing number of components

- Several methods available
  - Best case: They'll all agree
  - More likely: They will not
- When in doubt, go with parallel analysis or MAP
  - Scree plot and Kaiser don't work well
- Also consider rotated solutions (next)

7 Rotation

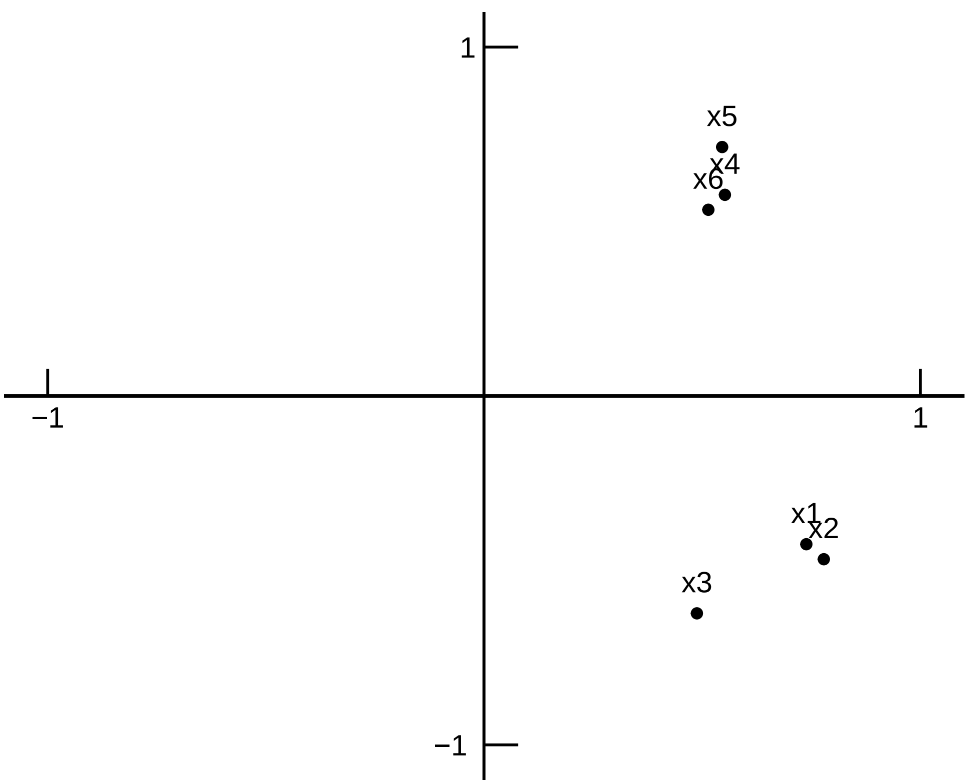
7.1 Simple structure

7.1.1 Simple structure and rotation

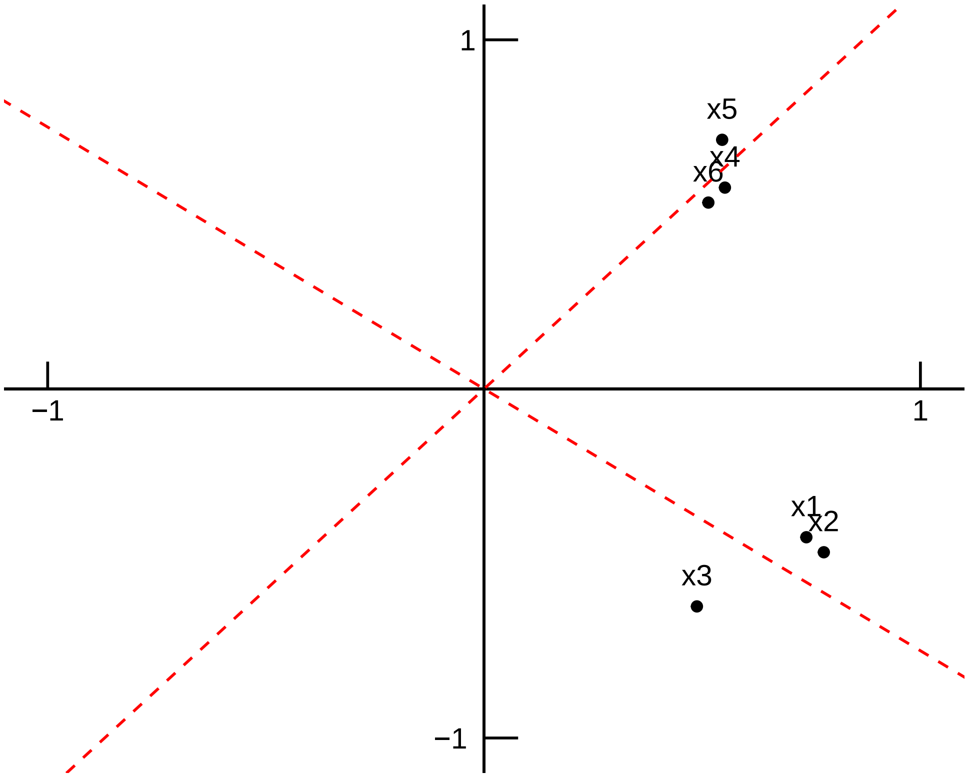
- Solution has **simple structure** if each item has **high loadings** on only one component and **near zero loadings** on all other components
  - i.e., points are near the axes
  - Easier to interpret: items only relate to one axis
- **Rotated solution** rotates the axes to get closer to *simple structure*
  - We'll look at some different ways to rotate the solution
    - I'll show you one way right now
  - Easier to interpret a solution that has simple structure

7.1.2 Loadings on unrotated vs rotated axes

- Loadings on unrotated axes



- Loadings on rotated axes



7.2 Orthogonal and oblique rotation

7.2.1 Orthogonal rotation

- **Orthogonal** means uncorrelated
  - Geometrically, axes are **perpendicular** (right angles)
- Components are all mutually orthogonal to start
  - Because the eigenvectors are mutually orthogonal
- Orthogonal rotation **rotates** the axes but keeps them uncorrelated

7.2.2 Orthogonal rotations

- **Varimax**
  - **Max**imizes the **vari**ance of squared loadings

  - High variance means loadings are bimodal
  - Bimodal: loadings near 0 or 1 (simple structure)

7.2.3 Oblique rotation

- **Oblique** means correlated
  - Geometrically, axes are **NOT perpendicular**
- Oblique rotation **rotates** the axes and **also** changes the angle between them
  - Components are **correlated**
  - Additional output: correlations between components

7.2.4 Oblique rotations

- **Oblimin**
  - Minimize correlation between components while trying to eliminate "in between" loadings (0.1 to 0.3)
- **Promax**
  - Work toward a *target loading matrix*
  - Target matrix is loading matrix raised to a *power*
  - Move axes toward to get closer to target matrix
  - Can be difficult to use well: which power to raise to?

8 Conclusion

8.1 Summary of this week

8.1.1 Summary of this week

- Principal components analysis (PCA)
  - Reduce # of variables (from p variables to $< p$ components)
  - Loadings relate items to components
  - Communalities are how much variance in each item is retained with that number components
  - Rotation to improve interpretability, correlate components

8.2 Next week

8.2.1 Next week

- Factor analysis
  - Related to PCA, but quite different model
  - Different set of assumptions: Aligns with psychology