

APSTA-GE 2094

APSY-GE 2524

Modern Approaches in Measurement: Lecture 2

Klnt Kanopka

New York University

Table of Contents

1. Measurement - Week 2

1. Table of Contents

2. Announcements

2. Validity

3. Classical Test Theory Practice

1. Differential Item Functioning

4. Break

5. Dimensionality Reduction

6. Principal Components Analysis (PCA)

7. Factor Analysis

8. PCA and EFA Practice

9. Wrap Up

Announcements

- PS0 is due tomorrow night @ 11.59p!
 - Submit a rendered .pdf on Gradescope
 - Final late deadline Monday night (2/2 @ 11.59p)
 - 10 point late penalty per day
- PS1 is out!
 - Due in two weeks (2.13 @ 11.59p)
 - Files posted on the Problem Sets tab
 - Again, submit on Gradescope
 - This is longer than PS0, and you can complete a little more than half of it after today
- Had some people pop into office hours yesterday and it was fun! Those are Wednesdays 2-3p in the second floor lobby of Kimball Hall

Check-In

- PollEv.com/klintkanopka

Validity

Validity

- On the most basic level, the idea that we have confidence we are measuring the thing we think we are measuring
- Starts with the definition of a *construct*, or the unobservable thing we care about measuring
- Requires some combination of theory and some amount of empirical study to get at

Validity

1. Take 5-10 minutes and in groups of 3 ± 1 , discuss the differences between Cronbach and Meehl's conception of "construct validity," Kane's idea of validity, and Boorsboom's idea of validity
2. What are the pros/cons of each approach?
3. Where might each approach succeed or fail?
4. What approach do you agree with most?

Cronbach and Meehl

- Construct validity is derived from the nomological network
- Core idea: Since we cannot directly observe the latent construct of interest, we have to collect a dense network of evidence through theoretical arguments and correlations to support our ideas of validity
- The idea of "construct validity" and other "types" of validity is extremely antiquated

Kane

- Validity is associated with the interpretation assigned to test scores rather than with the test scores or the test
- Interpretive arguments lay out how a test score will be interpreted and what evidence needs to be brought to bear to evaluate the argument
- Core idea: Validity is highly contextual and changes to instruments or populations can result in changes to the construct being measured

Boorsboom

- A test is valid for measuring an attribute if (a) the attribute exists and (b) variations in the attribute causally produce variation in the measurement outcomes
- Focuses on causal reasoning
- Core idea: Validity should be a function of an instrument, not a score interpretation

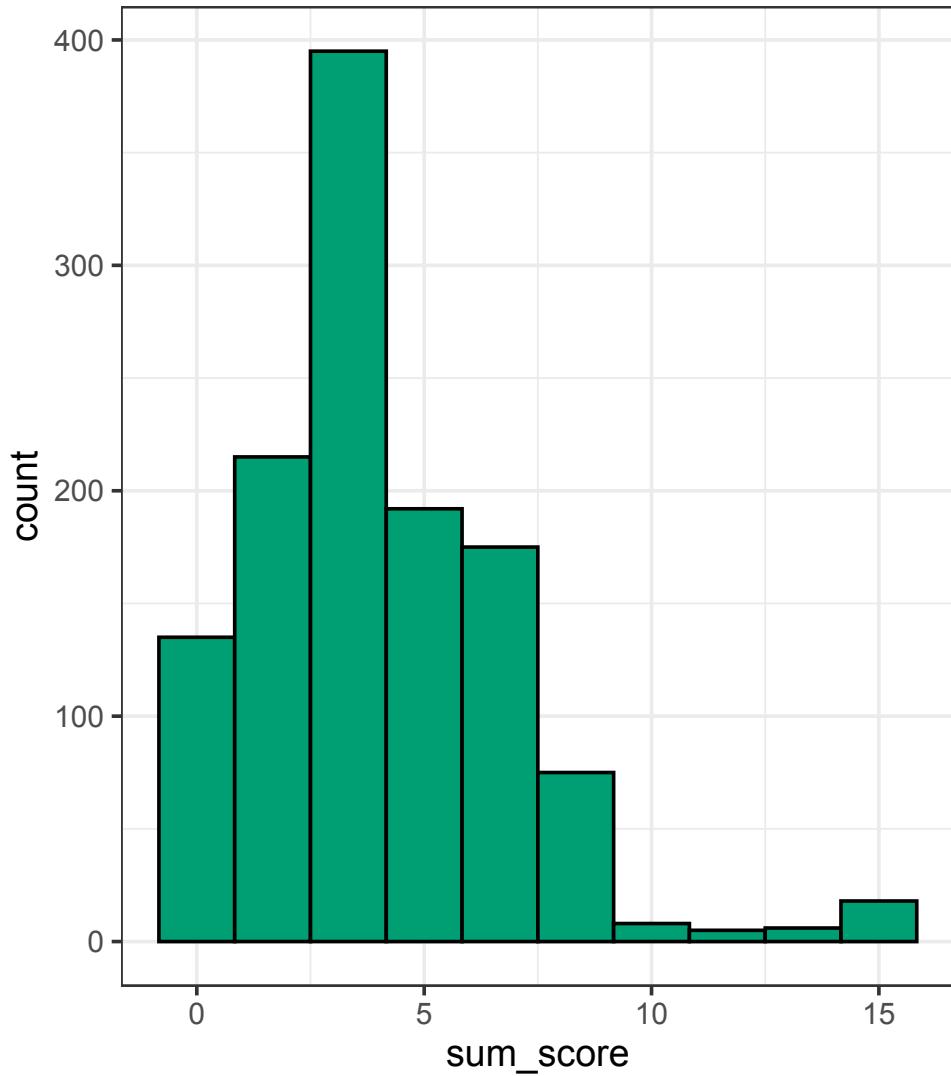
Classical Test Theory Practice

More Item Response Data

- The dataset `animalfights_clean.rds` (downloadable [here](#)) is a pre-cleaned version of the data from PS0.
- Code used in todays lecture is downloadable [here](#)
- In a group of 3 ± 1 take ten minutes to:
 1. Compute the ability (sum score) for each respondent and plot a distribution
 2. Compute the difficulty (p -value) for each item and plot them in order of difficulty
 3. Compute the discrimination (item-total correlation) for each item and plot them in order
 4. Check the item `d_king_cobra` for DIF by gender

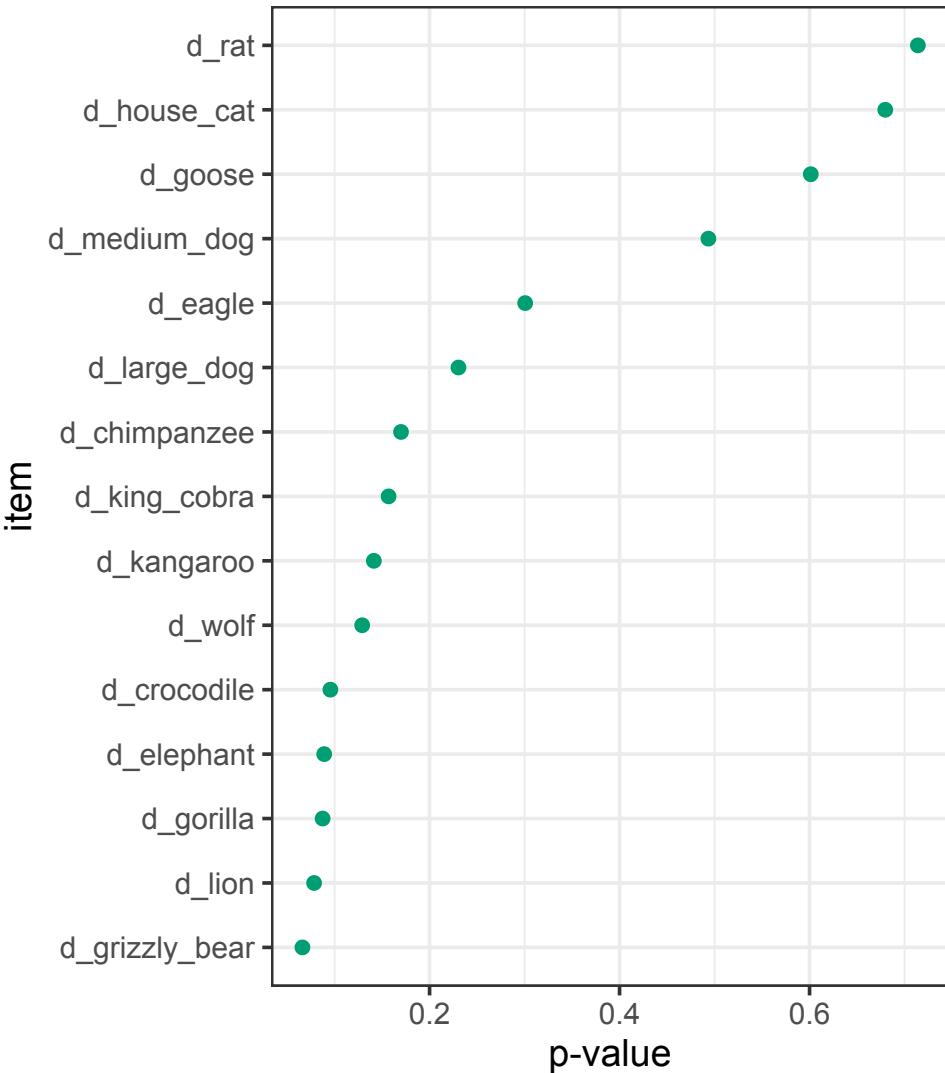
Sum Score

```
1 d <- readRDS('animalfights_clean.rds')
2
3 d_long <- d |>
4   pivot_longer(
5     cols = starts_with('d_'),
6     names_to = 'item',
7     values_to = 'resp'
8   )
9
10 sum_scores <- d_long |>
11   group_by(id) |>
12   summarize(sum_score = sum(resp))
13
14 ggplot(sum_scores, aes(x = sum_score)) +
15   geom_histogram(
16     bins = 10,
17     color = 'black',
18     fill = okabeito_colors(3)
19   ) +
20   theme_bw()
```



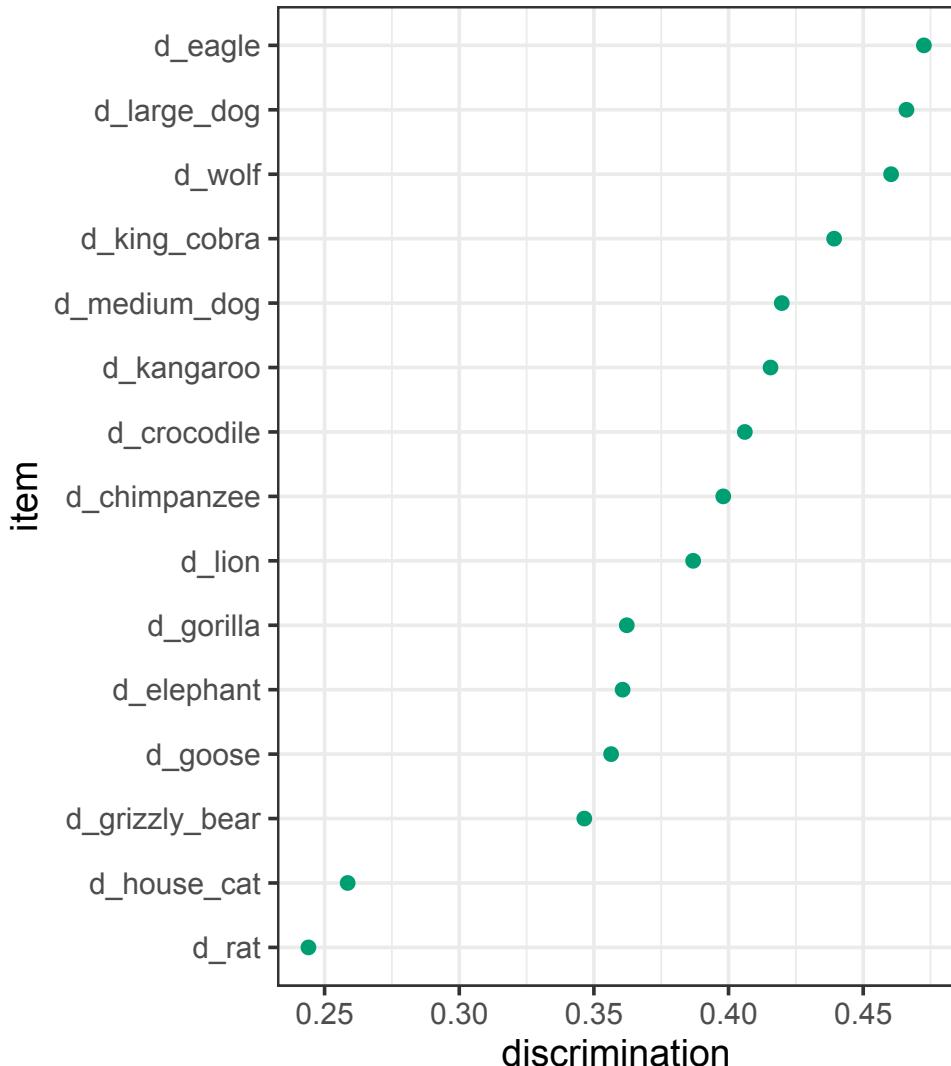
p-value

```
1 diff <- d_long |>
2   group_by(item) |>
3   summarize(p = mean(resp))
4
5 ggplot(diff, aes(x = p, y = reorder(item, p)))
6   geom_point(
7     color = okabeito_colors(3)
8   ) +
9   labs(x = 'p-value', y = 'item') +
10  theme_bw()
```



Item-Total Correlation

```
1 disc <- left_join(d_long, sum_scores, by = 'id'
2   mutate(adjusted_sum_score = sum_score - resp
3   group_by(item) |>
4   summarize(a = cor(resp, adjusted_sum_score))
5
6 ggplot(disc, aes(x = a, y = reorder(item, a)))
7   geom_point(
8     color = okabeito_colors(3)
9   ) +
10  labs(x = 'discrimination', y = 'item') +
11  theme_bw()
```



Differential Item Functioning

- Regression:

```
1 d_long |>
2   left_join(sum_scores, by = 'id') |>
3   mutate(adjusted_sum_score = sum_score - resp) |>
4   filter(item == 'd_king_cobra') |>
5   lm(resp ~ gender + adjusted_sum_score, data= _) |>
6   summary()
```

- Output:

```
1 Residuals:
2      Min       1Q     Median       3Q      Max
3 -0.66917 -0.20893 -0.09386  0.05007  0.99254
4
5 Coefficients:
6                               Estimate Std. Error t value Pr(>|t|)
7 (Intercept)           -0.107599   0.017667  -6.091 1.51e-09 ***
8 genderM              0.086401   0.018991   4.550 5.91e-06 ***
9 adjusted_sum_score   0.057531   0.003625  15.870 < 2e-16 ***
10 ---
11 Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Break

Dimensionality Reduction

Dimensionality Reduction (Conceptually)

- We have data with a bunch of variables (columns)
- We want to approximate the data with fewer variables (columns)
- This is called constructing a *low rank approximation*
 - The *rank* of a matrix is the number of linearly independent columns
 - Cognitively, it's easier to think about and interpret a smaller number of variables
- **Why is this a good idea?**
 - You might have measured a bunch of stuff in order to take a guess at a thing you couldn't really measure, but you don't know how to weight the variables you did measure!
 - Your variables may be redundant measures of some latent construct
 - Maybe you want to look for groups of variables or people



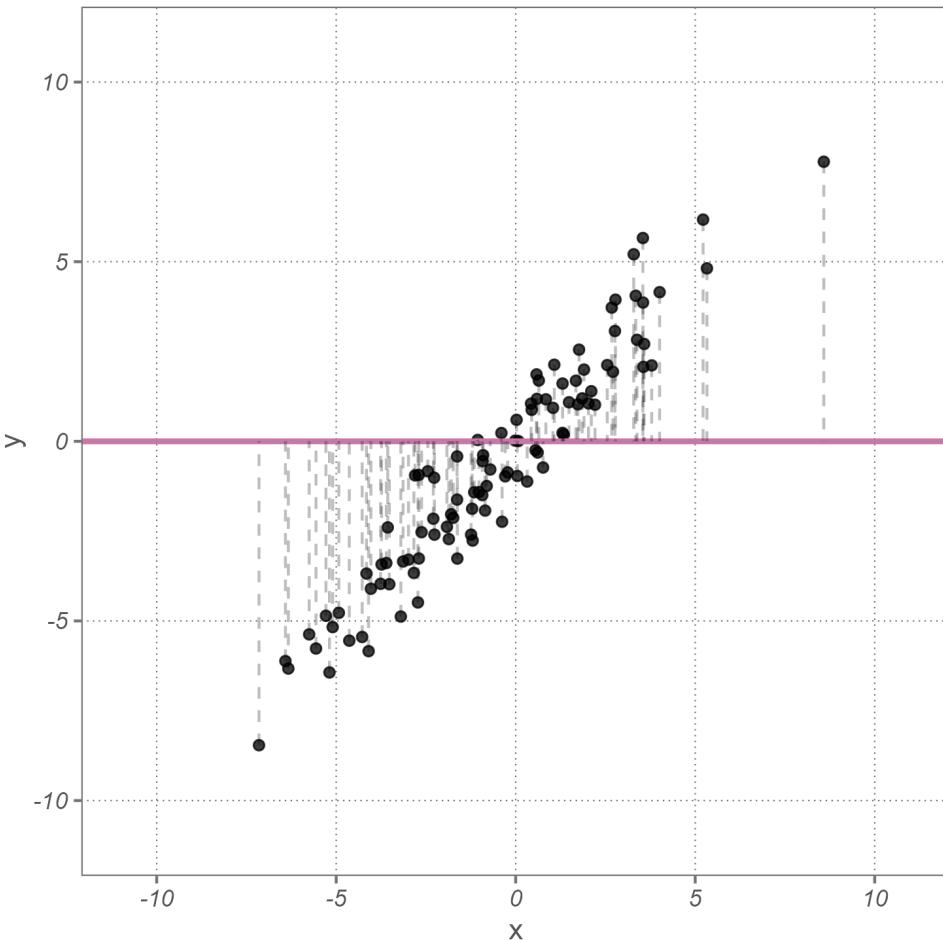
Principal Components Analysis (PCA)

PCA Overview

- Goal: Summarize your data with fewer variables than you have
- Process: Construct a *principal component* (kind of a new variable) as a linear combination of the variables you have that minimizes the *reconstruction error*. Then repeat
 - Reconstruction error is the *orthogonal* distance from each point to the line
- This is usually done with the Singular Value Decomposition
- Note that the task is similar to regression, but you minimize the orthogonal distance from each observation to the line, not the distance from your outcome y to the line!

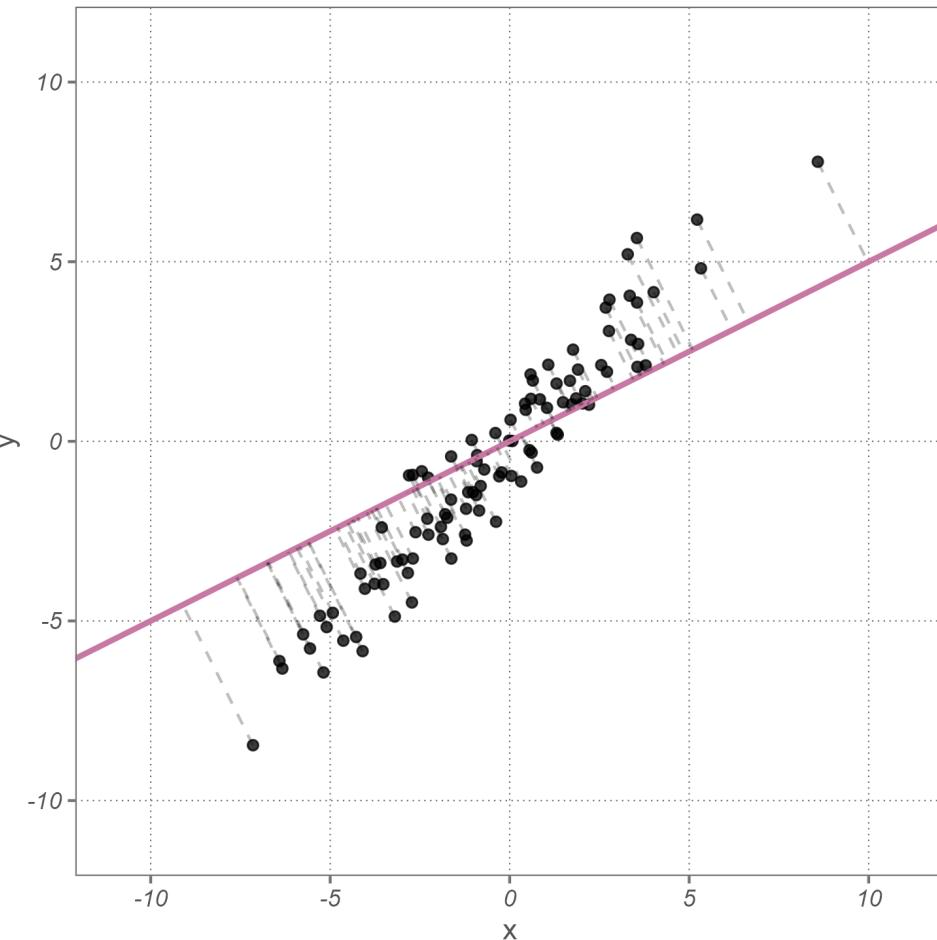
PCA

- The *reconstruction error* is the sum of the squared orthogonal distances from each point to the pink line
- Represented by the dashed lines



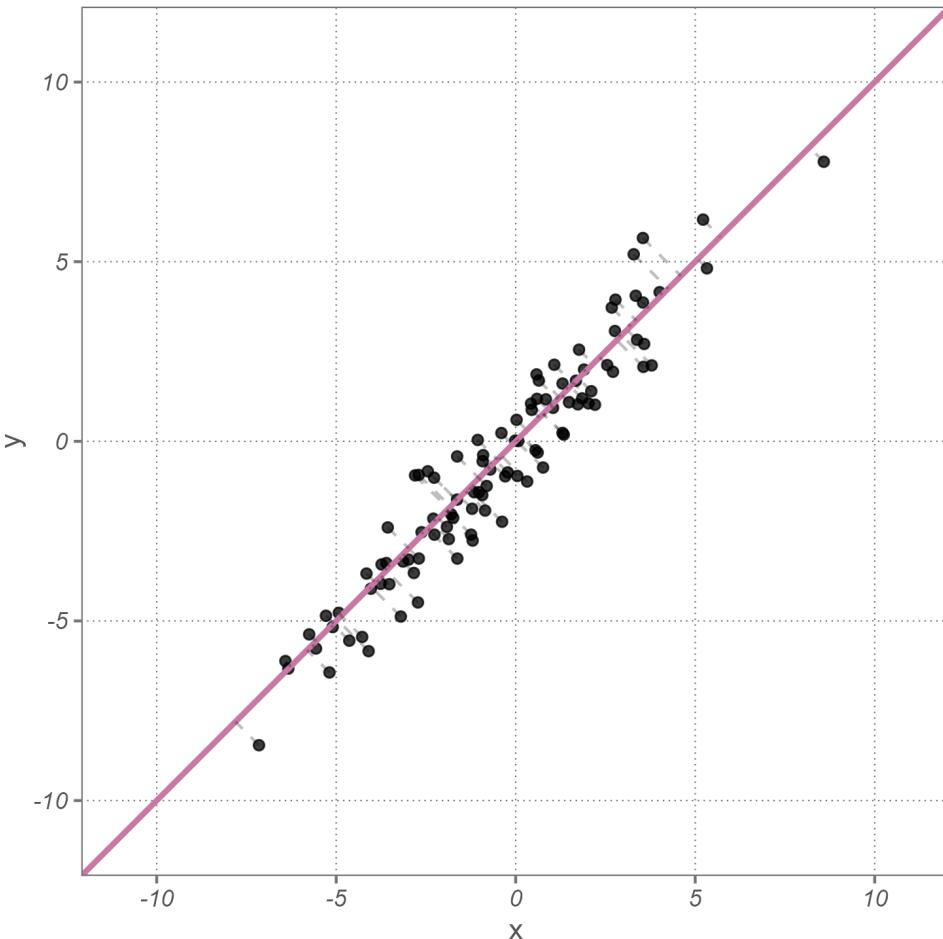
PCA

- The *reconstruction error* is the sum of the squared orthogonal distances from each point to the pink line
- Represented by the dashed lines
- As the line rotates to align with the points, the distances shrink and the projections of each point onto the line spread out



PCA

- The *reconstruction error* is the sum of the squared orthogonal distances from each point to the pink line
- Represented by the dashed lines
- As the line rotates to align with the points, the distances shrink and the projections of each point onto the line spread out
- The first PC does two things:
 1. **Minimizes** the reconstruction error
 2. **Maximizes** the explained variance
- Projections onto the first PC are a single number that retains the most information about the original data



Estimating Principal Components

```
1 library(FactoMineR)
2 library(factoextra)
3
4 # isolate item responses and estimate pca
5 resp <- select(d, starts_with('d_'))
6 pca <- PCA(resp, ncp=10, graph=FALSE)
7
8 # extract eigenvalue information
9 pca_eig <- data.frame(pca$eig) |>
10   rownames_to_column(var='dimension')
11
12 # extract dimensions/loading
13 pca_dim <- data.frame(pca$var$coord) |>
14   rownames_to_column(var='item')
15
16 # extract individual scores/projections
17 pca_resp <- data.frame(pca$ind$coord) |>
18   rownames_to_column(var='person')
```

Eigenvalues: How many PCs are useful?

dimension	eigenvalue	percentage.of.variance	cumulative.percentage.of.variance
comp 1	4.2525175	28.350117	28.35012
comp 2	2.7113610	18.075740	46.42586
comp 3	0.9289215	6.192810	52.61867
comp 4	0.8510230	5.673486	58.29215
comp 5	0.7637384	5.091589	63.38374
comp 6	0.7161497	4.774331	68.15807
comp 7	0.6734326	4.489551	72.64762

Option 1: The Kaiser Criterion

- Select all dimensions with an eigenvalue ≥ 1

dimension	eigenvalue	percentage.of.variance	cumulative.percentage.of.variance
comp 1	4.2525175	28.350117	28.35012
comp 2	2.7113610	18.075740	46.42586
comp 3	0.9289215	6.192810	52.61867
comp 4	0.8510230	5.673486	58.29215

Option 2: Cumulative Variance

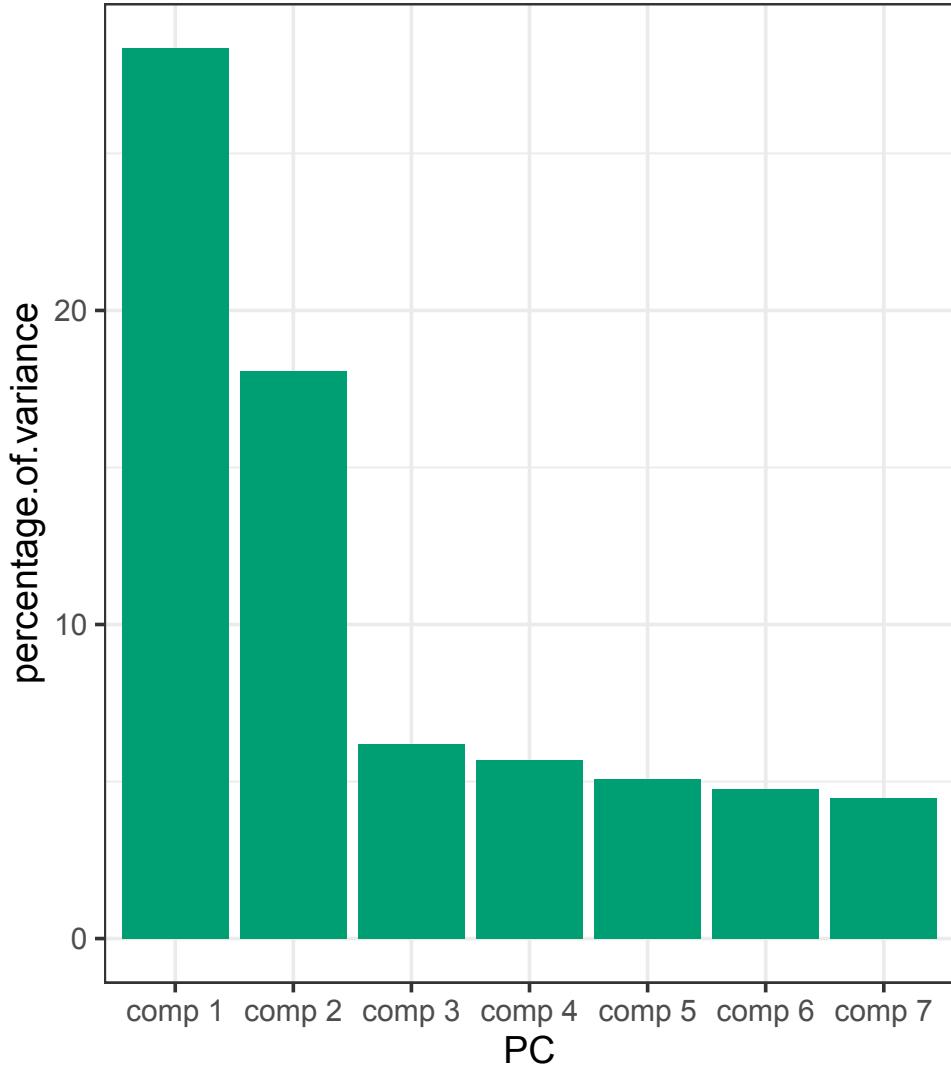
- Pick some cumulative variance that you want (often 50% or 75%)
- Retain dimensions until you exceed that threshold

dimension	eigenvalue	percentage.of.variance	cumulative.percentage.of.variance
comp 1	4.2525175	28.350117	28.35012
comp 2	2.7113610	18.075740	46.42586
comp 3	0.9289215	6.192810	52.61867
comp 4	0.8510230	5.673486	58.29215

Option 3: The Elbow Plot

- Plot the variance explained by each dimension
- Drop dimensions that don't contribute much more over the previous dimension
- This means go up to the "elbow"

```
1 pca_eig |>
2   head(7) |>
3   ggplot(aes(
4     x = dimension,
5     y = percentage.of.variance
6   )) +
7   geom_col(fill = okabeito_colors(3)) +
8   labs(x = 'PC') +
9   theme_bw()
```



Dimensions and Loadings

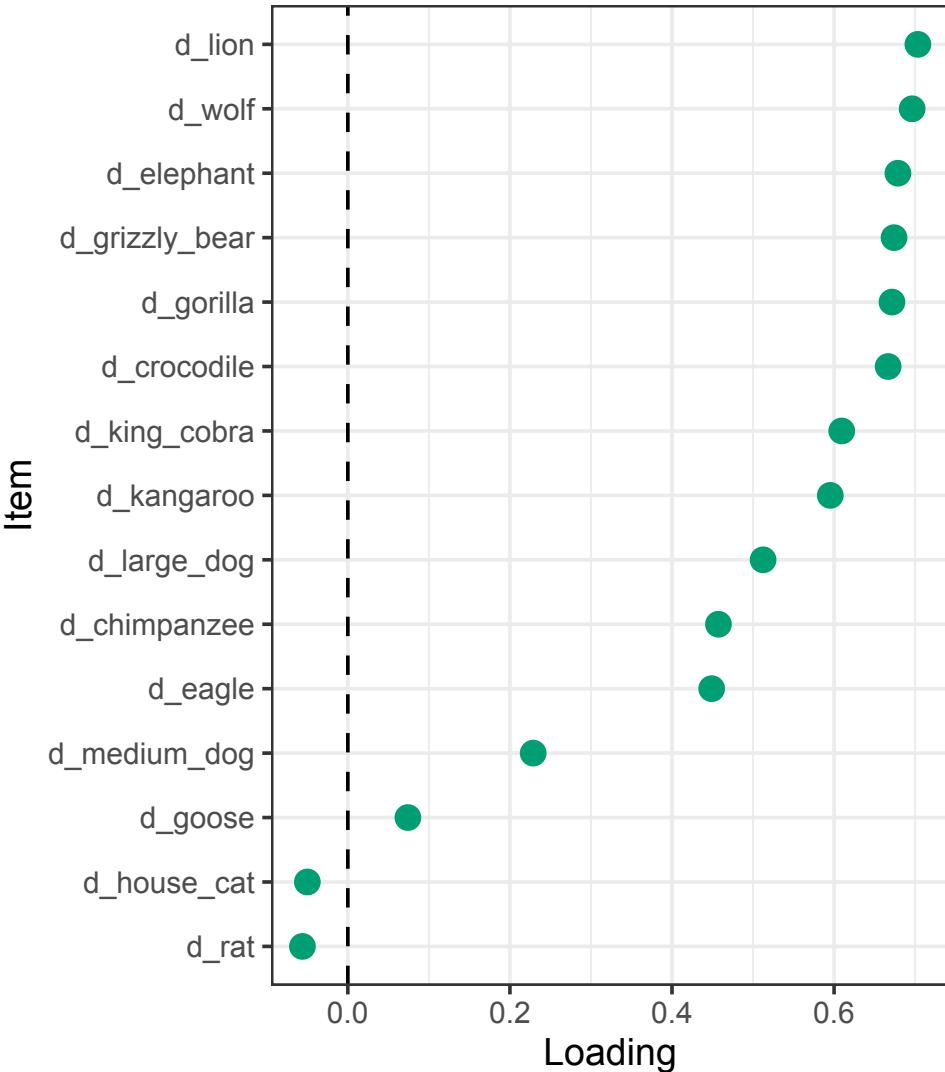
item	Dim.1	Dim.2	Dim.3
d_rat	-0.0560027	0.7478180	0.3220408
d_house_cat	-0.0499744	0.7728305	0.2701365
d_medium_dog	0.2287211	0.6610715	-0.0932625
d_large_dog	0.5124123	0.2998211	-0.3943514
d_kangaroo	0.5953089	0.0527450	-0.1915674
d_eagle	0.4489964	0.4131870	-0.3342900
d_grizzly_bear	0.6740091	-0.2427908	0.3365680

Individuals

person	Dim.1	Dim.2	Dim.3
1	0.277625	1.7352156	-0.7241404
2	-1.362983	0.4078412	1.0116342
3	-1.141138	1.2108529	0.8180881
4	-1.384454	-1.5588112	0.0452277
5	1.669899	2.1264304	-0.3689416

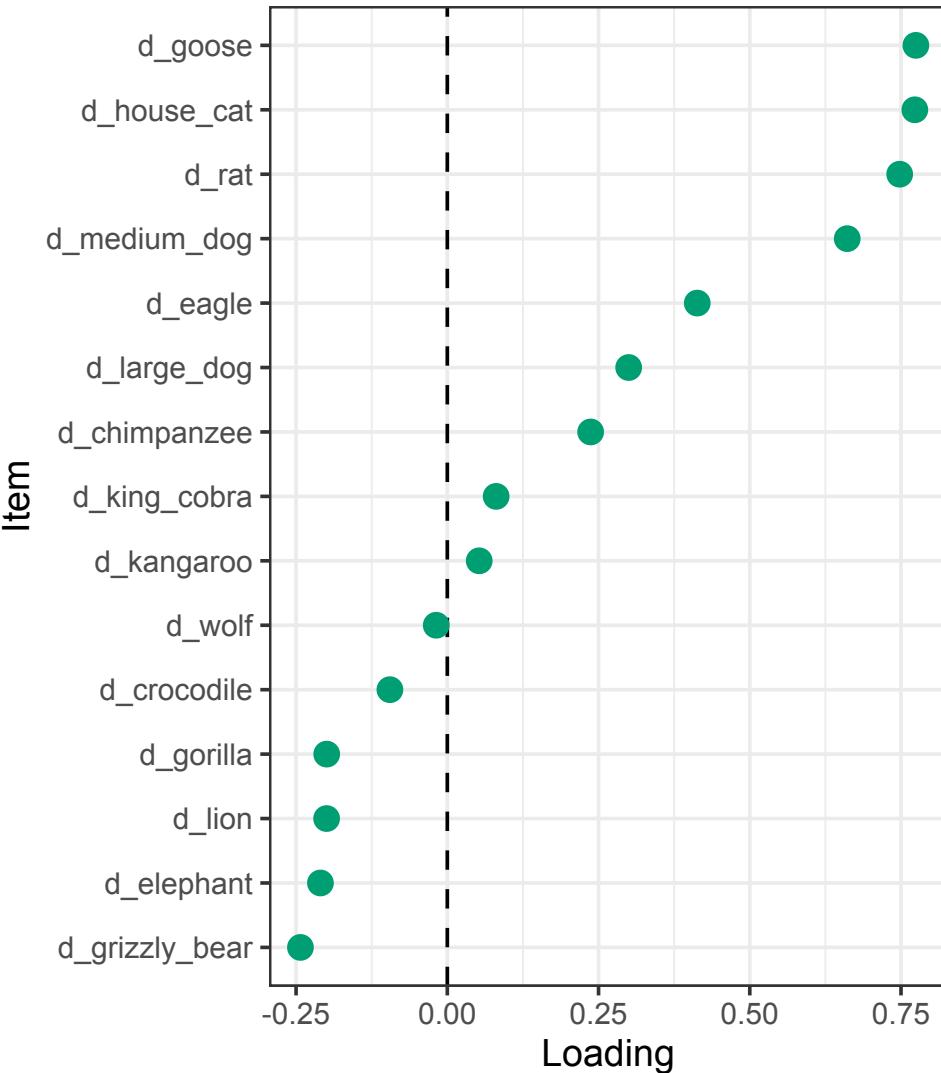
Dimension 1

```
1 ggplot(pca_dim,
2         aes(x = Dim.1,
3               y = reorder(item, Dim.1))) +
4   geom_vline(aes(xintercept = 0), lty = 2) +
5   geom_point(size = 3,
6               color = okabeito_colors(3)) +
7   labs(x = 'Loading', y = 'Item') +
8   theme_bw()
```



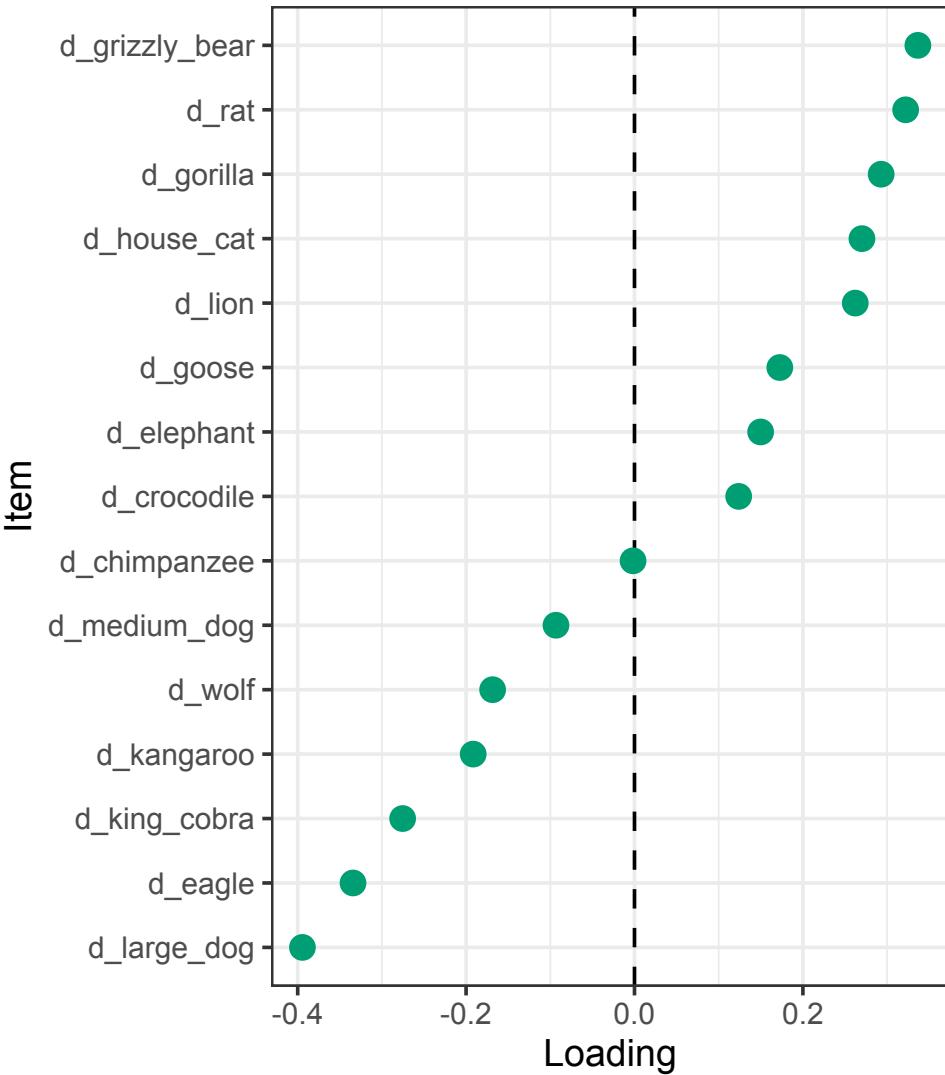
Dimension 2

```
1 ggplot(pca_dim,
2         aes(x = Dim.2,
3               y = reorder(item, Dim.2))) +
4   geom_vline(aes(xintercept = 0), lty = 2) +
5   geom_point(size = 3,
6               color = okabeito_colors(3)) +
7   labs(x = 'Loading', y = 'Item') +
8   theme_bw()
```



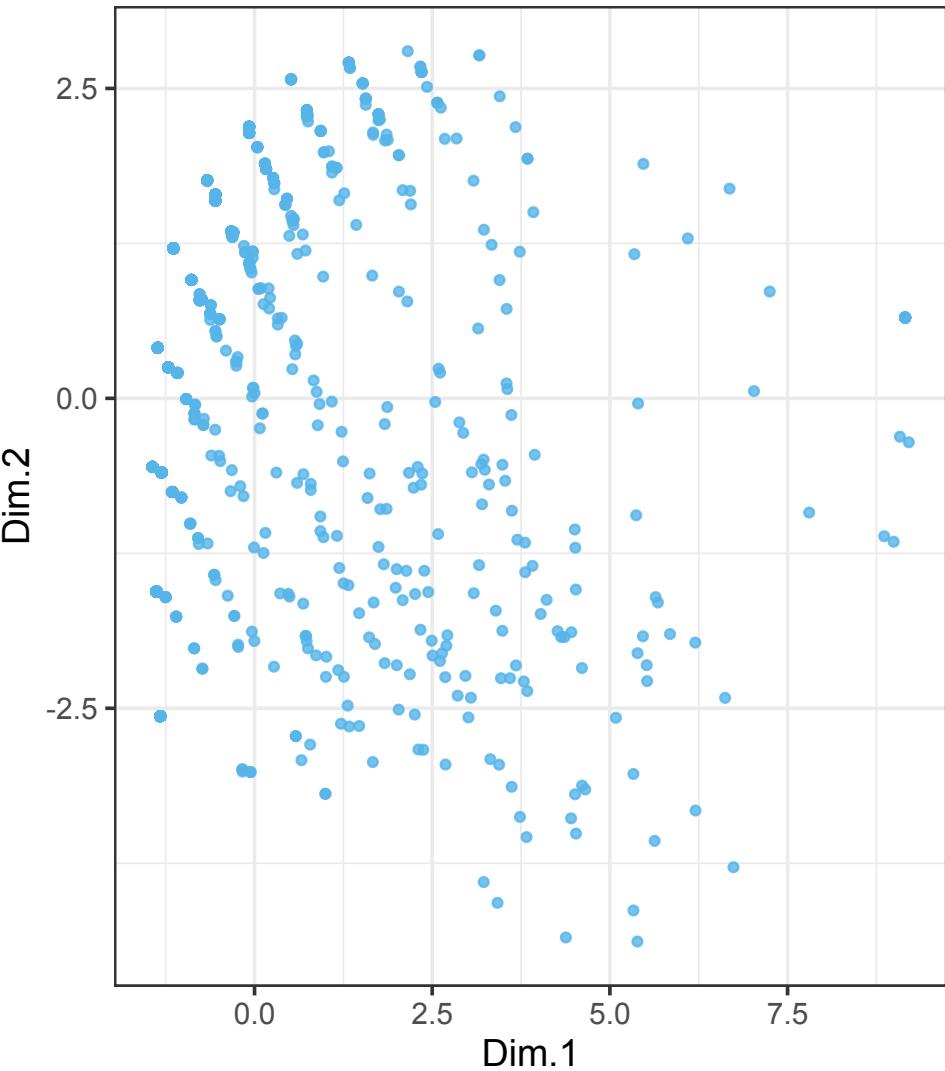
Dimension 3

```
1 ggplot(pca_dim,
2         aes(x = Dim.3,
3               y = reorder(item, Dim.3))) +
4   geom_vline(aes(xintercept = 0), lty = 2) +
5   geom_point(size = 3,
6               color = okabeito_colors(3)) +
7   labs(x = 'Loading', y = 'Item') +
8   theme_bw()
```



Individuals

```
1 ggplot(pca_resp,
2         aes(x = Dim.1, y = Dim.2)) +
3         geom_point(alpha = 0.8, size = 1,
4                     color = okabeito_colors(2)) +
5         theme_bw()
```



PCA Summary

- Constructs successive **orthogonal** dimensions
- Each dimension maximizes the amount of explained residual variance
- Because dimensions are orthogonal, individual locations along these dimensions are uncorrelated **by construction**
- Variable loadings on dimensions can be used to interpret and describe the dimensions
- Individual projections onto these dimensions can be interpreted as scores along those dimensions

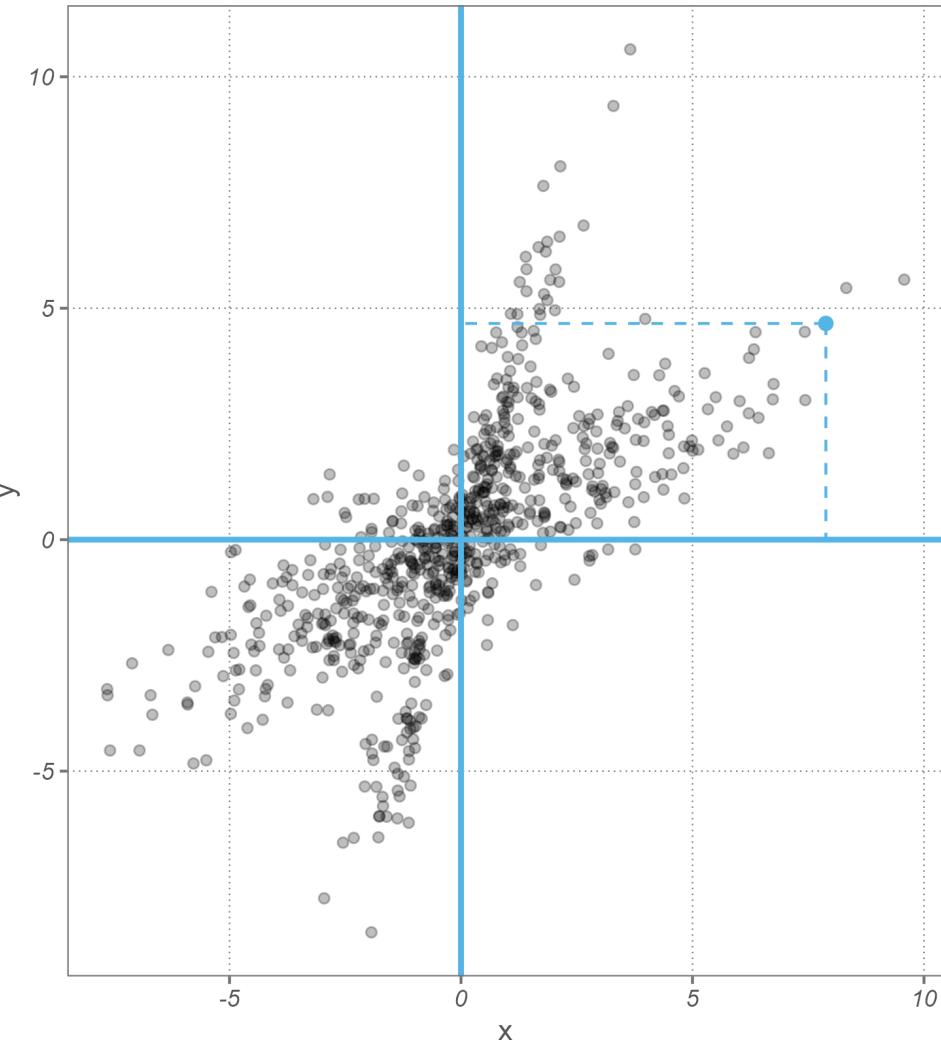
Factor Analysis

(Exploratory) Factor Analysis

- Conceptually different from PCA
- Factor analysis is a latent variable model that seeks to estimate item level associations (loadings) onto latent variables
- The core idea is that levels of the latent variable cause individual item responses, which are observed with error
- "Factor Analysis" is really two things:
 - **Confirmatory Factor Analysis (CFA)** is another name for structural equation modeling (SEM) with item responses
 - **Exploratory Factor Analysis (EFA)** is a dimensionality reduction technique that is conceptually different from PCA but *sometimes* mathematically identical to PCA
- If you allow for correlated factors, EFA is **not** equivalent to PCA

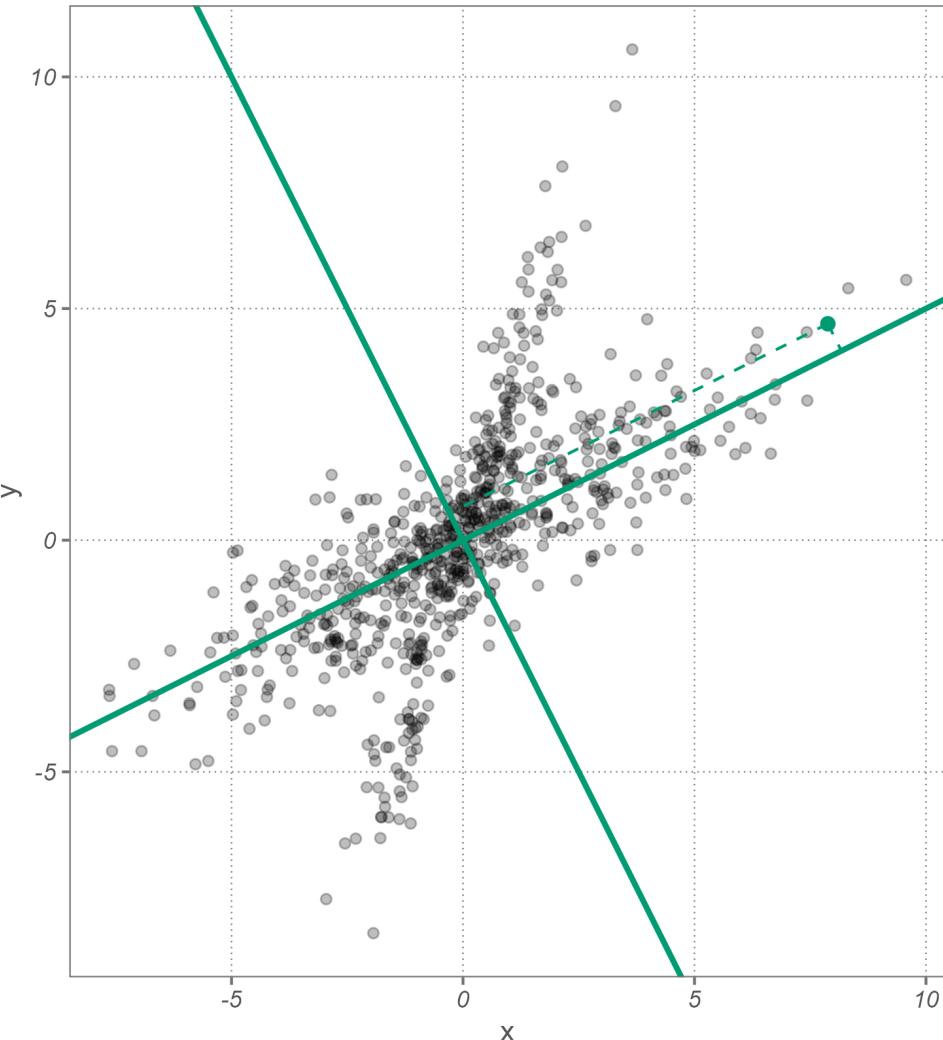
EFA

- Similar to PCA, EFA constructs dimensions that are linear combinations of observed variables and projects points onto them
- EFA models are not identified; there are an infinite number of solutions that describe data equally well (factor indeterminacy problem)



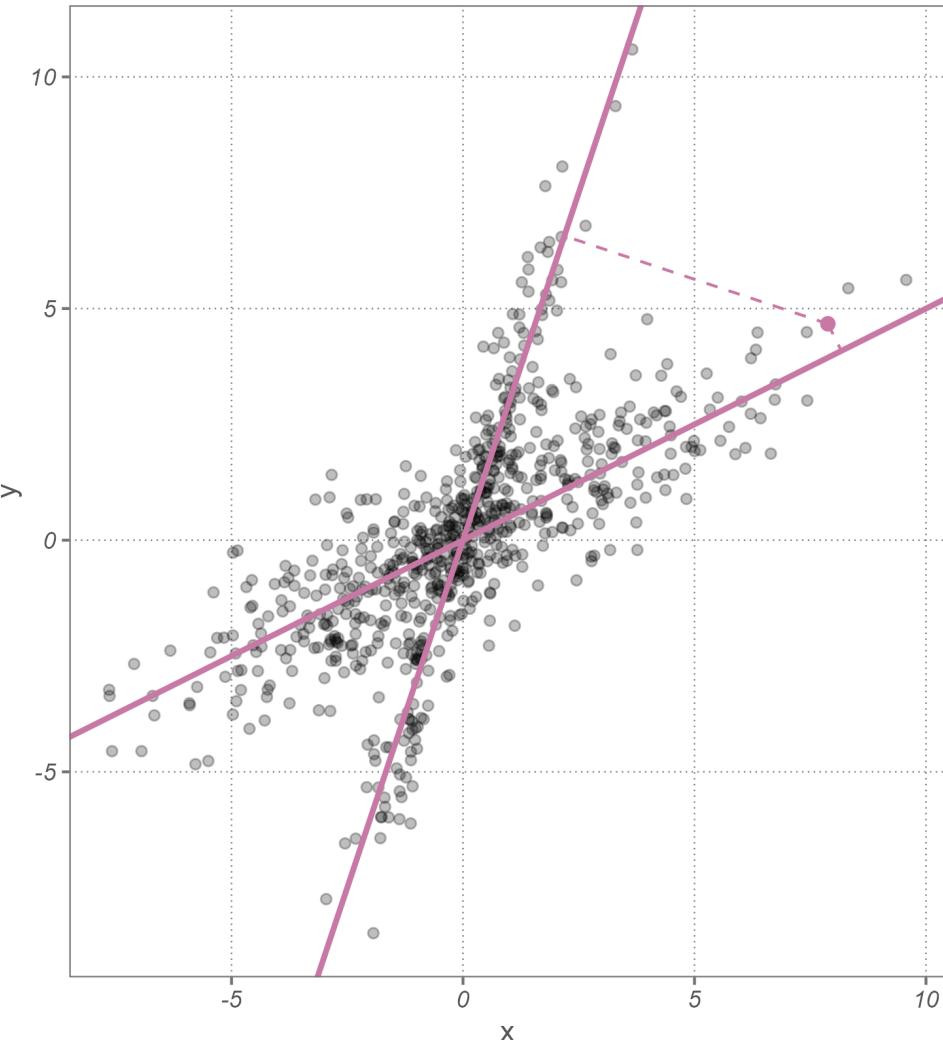
EFA

- Similar to PCA, EFA constructs dimensions that are linear combinations of observed variables and projects points onto them
- EFA models are not identified; there are an infinite number of solutions that describe data equally well (factor indeterminacy problem)
- EFA leverages *rotations* to solve this that try to pick dimensions to satisfy some sort of logical criterion. Varimax rotation is equivalent to PCA



EFA

- Similar to PCA, EFA constructs dimensions that are linear combinations of observed variables and projects points onto them
- EFA models are not identified; there are an infinite number of solutions that describe data equally well (factor indeterminacy problem)
- EFA leverages *rotations* to solve this that try to pick dimensions to satisfy some sort of logical criterion. **Varimax** rotation is equivalent to PCA
- EFA also allows for *oblique* rotations that produce correlated factors. **Promax** is the most commonly used one



Estimating EFA

```
1 library(psych)
2
3 # Check the documentation - it's intense!
4 ?fa
5
6 efa_1 <- fa(resp, nfactors = 1, rotate = 'oblimin')
7 efa_2 <- fa(resp, nfactors = 2, rotate = 'oblimin')
8 efa_3 <- fa(resp, nfactors = 3, rotate = 'oblimin')
```

One Factor Solution

```
1 efa_1$Vaccounted
```

```
1                               MR1
2 SS loadings    3.6045189
3 Proportion Var 0.2403013
```

Two Factor Solution

```
1 efa_2$Vaccounted
```

	MR1	MR2
2 SS loadings	3.6227900	2.1692315
3 Proportion Var	0.2415193	0.1446154
4 Cumulative Var	0.2415193	0.3861348
5 Proportion Explained	0.6254794	0.3745206
6 Cumulative Proportion	0.6254794	1.0000000

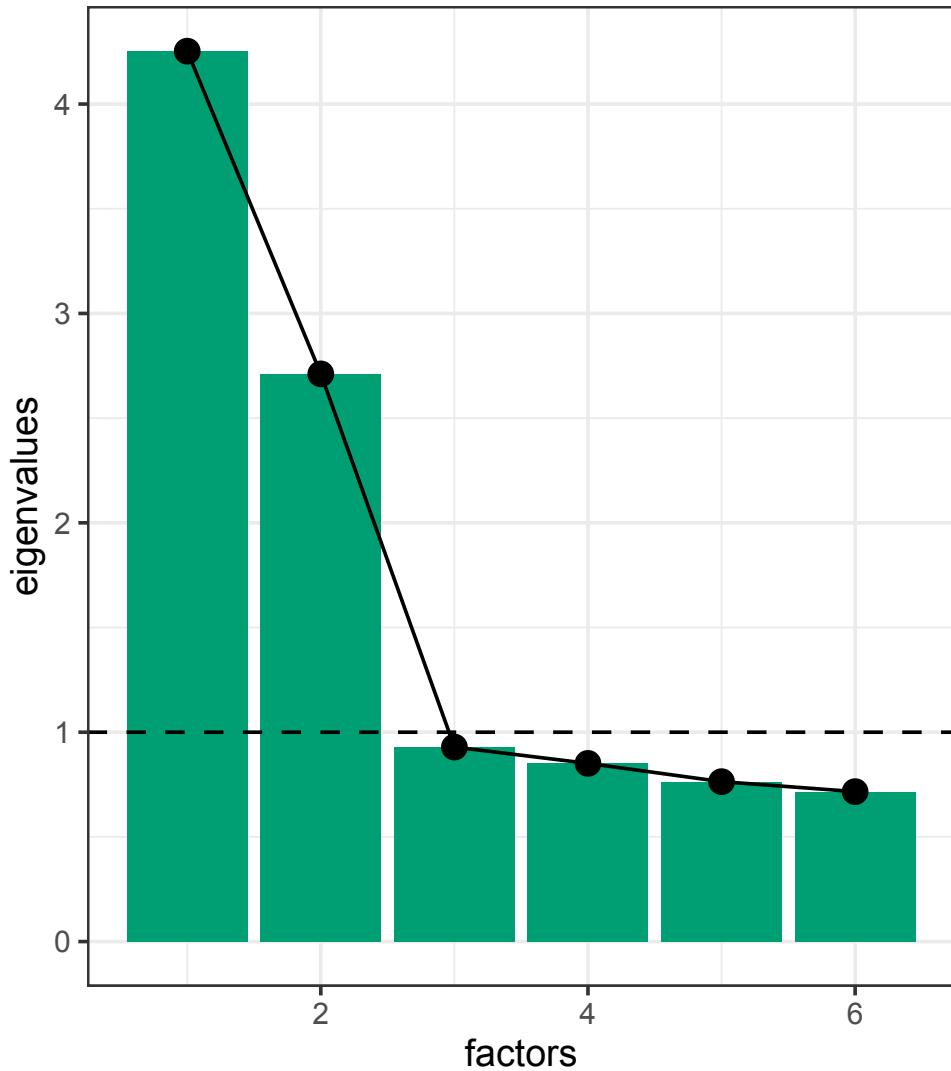
Three Factor Solution

```
1 efa_3$Vaccounted
```

	MR1	MR2	MR3
2 SS loadings	2.5972061	1.8689791	1.7516499
3 Proportion Var	0.1731471	0.1245986	0.1167767
4 Cumulative Var	0.1731471	0.2977457	0.4145223
5 Proportion Explained	0.4177026	0.3005836	0.2817138
6 Cumulative Proportion	0.4177026	0.7182862	1.0000000

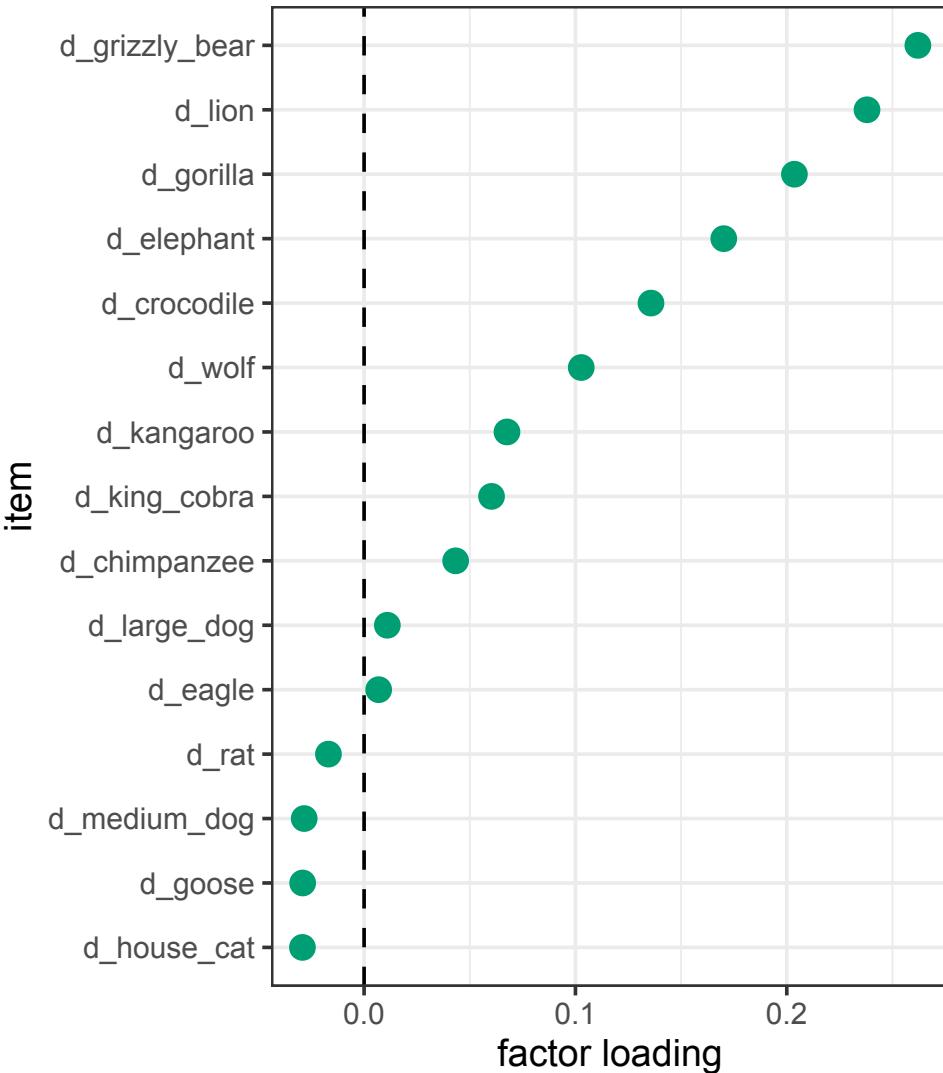
Selecting Number of Factors

```
1 fa_eigenvalues <- data.frame(factors = 1:6,
2   eigenvalues = efa_3$e.values[1:6])
3
4 ## Selecting Number of Factors
5
6 ggplot(fa_eigenvalues,
7   aes(x = factors, y = eigenvalues)) +
8   geom_col(fill = okabeito_colors(3)) +
9   geom_point(size = 3) +
10  geom_line() +
11  geom_hline(aes(yintercept = 1), lty = 2) +
12  theme_bw()
```



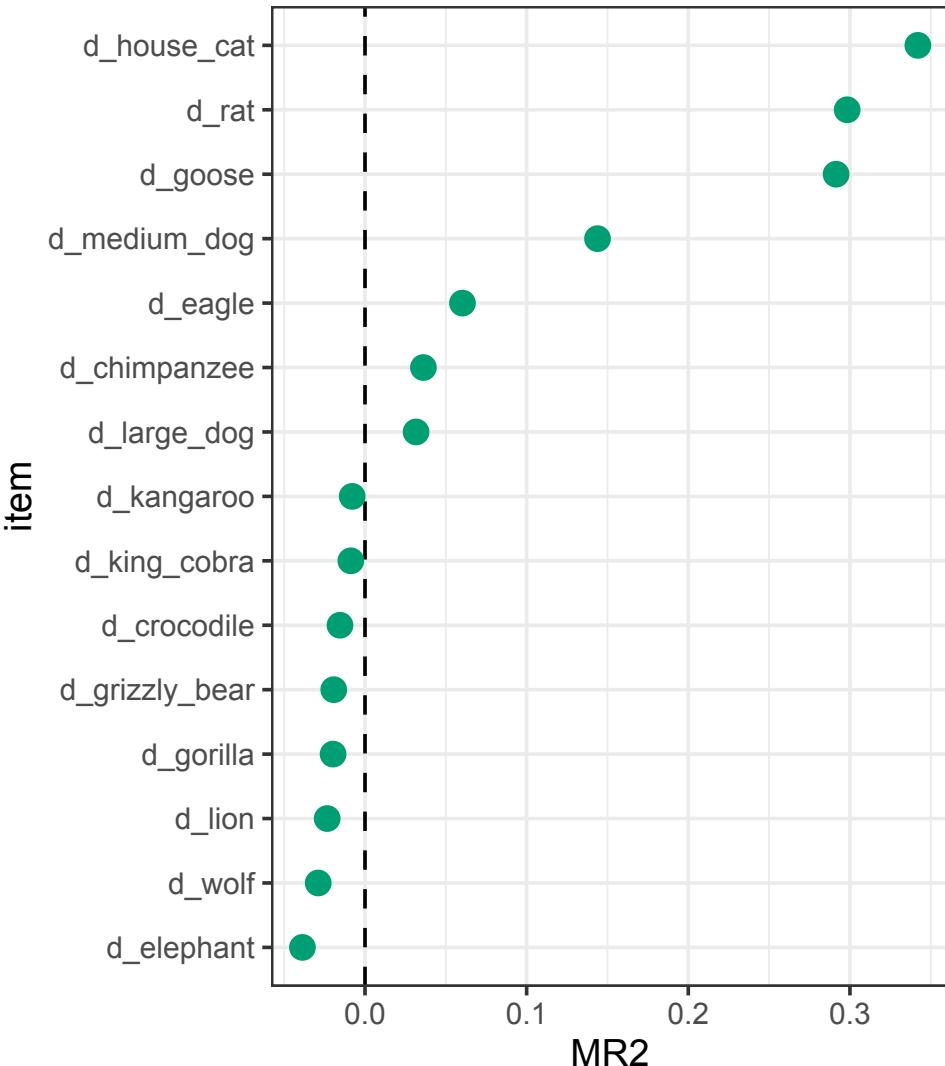
Factor 1 Loadings

```
1 ggplot(efa,
2     aes(x = MR1,
3         y = reorder(item, MR1))) +
4     geom_vline(aes(xintercept = 0), lty = 2) +
5     geom_point(size = 3,
6             color = okabeito_colors(3)) +
7     labs(y = 'item', x = 'factor loading') +
8     theme_bw()
```



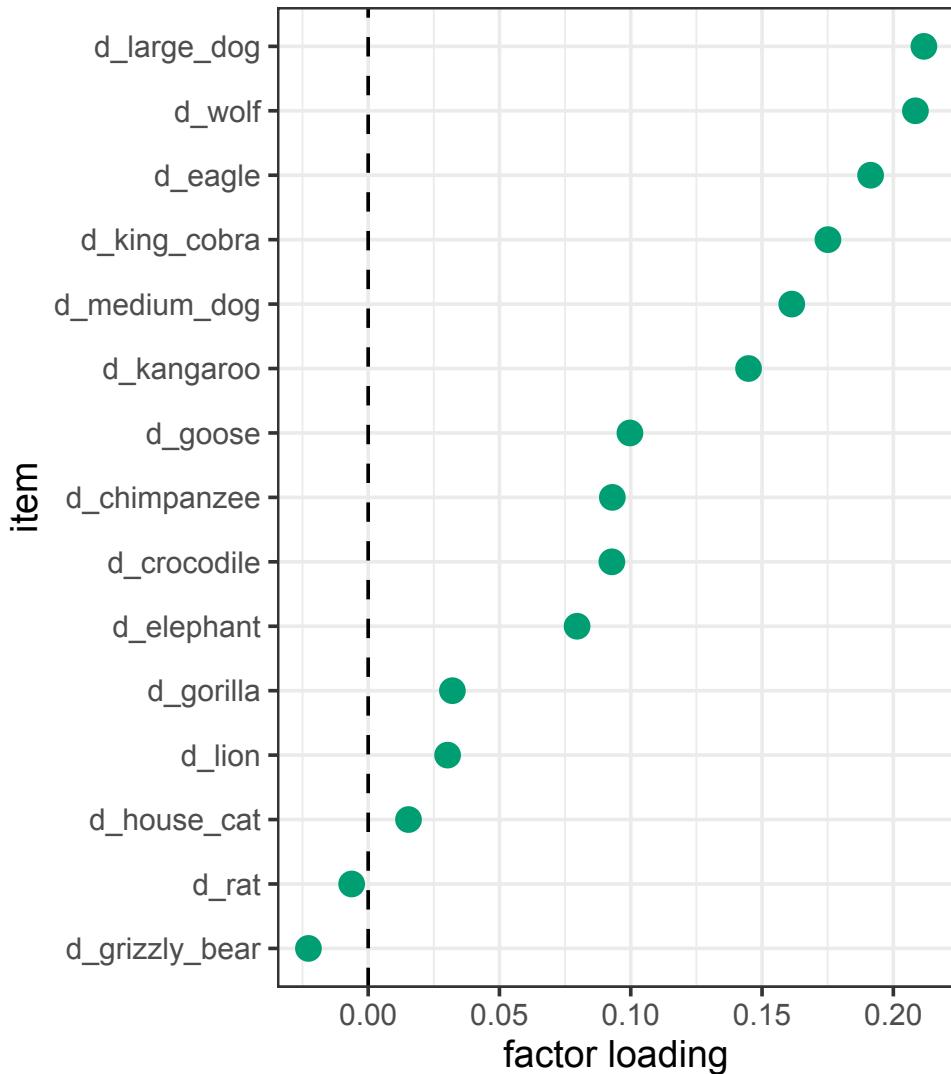
Factor 2 Loadings

```
1 ggplot(efa,
2       aes(x = MR2,
3              y = reorder(item, MR2))) +
4   geom_vline(aes(xintercept = 0), lty = 2) +
5   geom_point(size = 3,
6              color = okabeito_colors(3)) +
7   labs(y = 'item', x = 'factor loading') +
8   theme_bw()
```



Factor 3 Loadings

```
1 ggplot(efa,
2       aes(x = MR3,
3              y = reorder(item, MR3))) +
4   geom_vline(aes(xintercept = 0), lty = 2) +
5   geom_point(size = 3,
6              color = okabeito_colors(3)) +
7   labs(y = 'item', x = 'factor loading') +
8   theme_bw()
```



Individual Scores

```
1 efa_3$scores  
  
1           MR1          MR2          MR3  
2 [1,] -0.285443975 8.387273e-01 0.603740923  
3 [2,] -0.448881256 4.989128e-01 -0.636824429  
4 [3,] -0.505534260 7.864999e-01 -0.314300349  
5 [4,] -0.327039053 -8.288798e-01 -0.873253278  
6 [5,]  0.248132302 9.532238e-01 1.131932390
```

Factor Analysis Summary

- Constructs latent factors from correlation matrix
- Rotates factors to allow for correlated factors (dimensions)
- Factor loadings can be used to interpreted and describe the factors
- Individual scores on these factors can be recovered
- Generally, the use and interpretation are the same as in PCA

PCA and EFA Practice

- In a group of 3 ± 1 :
 - Download [machivallianism_test_main.csv](#)
 - Download [machivallianism_codebook.txt](#)
 - Fit a PCA or EFA
 - Select a number of dimensions or factors
 - Interpret each dimension
 - Make a scatter plot of respondents projected onto the first two factors

Wrap Up

PCA vs. Factor Analysis

- Variables
 - PCA is a model of observed variables
 - Factor analysis is a model of latent variables
- Data
 - PCA focuses on reconstructing the diagonals of the covariance matrix (individual variances)
 - Factor analysis focuses on reconstructing the off-diagonals of the covariance matrix (between-item covariances)
 - PCA produces orthogonal dimensions
 - Factor analysis can produce correlated factors
- Important to note that PCA and factor analysis usually (but not always) produce different results
- Often factor analysis produces more interpretable factors with an oblique rotation that allows for correlated factors

Problem Set 1

- Posted!
- Two weeks to complete
- You won't be able to finish all of it until after next week's lecture!
- Topics covered:
 - Comparing and contrasting PCA and Factor Analysis (with real data)
 - Comparing and contrasting CTT and Item Response Theory (with real data)
- Data will come from the Item Response Warehouse (IRW)
 - Website: <https://itemresponsewarehouse.org/>
 - Paper: <https://link.springer.com/article/10.3758/s13428-025-02796-y>
- Last year, students said this assignment took a long time. I didn't make it shorter, so start soon!

Check-Out

- PollEv.com/klintkanopka