SKIM co-clustering with covariates

Kees van der Wagt

Sawtooth Software

Conference - Orlando

May 2022

Background

- **From Wikipedia:**
  - **Clustering** is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters).
  - **Co-clustering**: a data mining technique which allows simultaneous clustering of the rows and columns of a matrix.
- Clustering of respondents based on their data is a common practice in market research. Jointly or simultaneously clustering respondents and items….not so much and that's a shame
- To make it even better, I wanted to be able to add covariates (respondent and data covariates)
- But, I could not find any existing software or algorithms that could do this, so I came up with an algorithm (And in all honesty, I did not understand the existing algorithms so that I could add in the covariates)

SKIM

Chapters

1 | Clustering

2 | Co-clustering

3 | Why would you?

4 | Covariates

5 | Co-clustering with covariates

6|Why would you?

SKIM

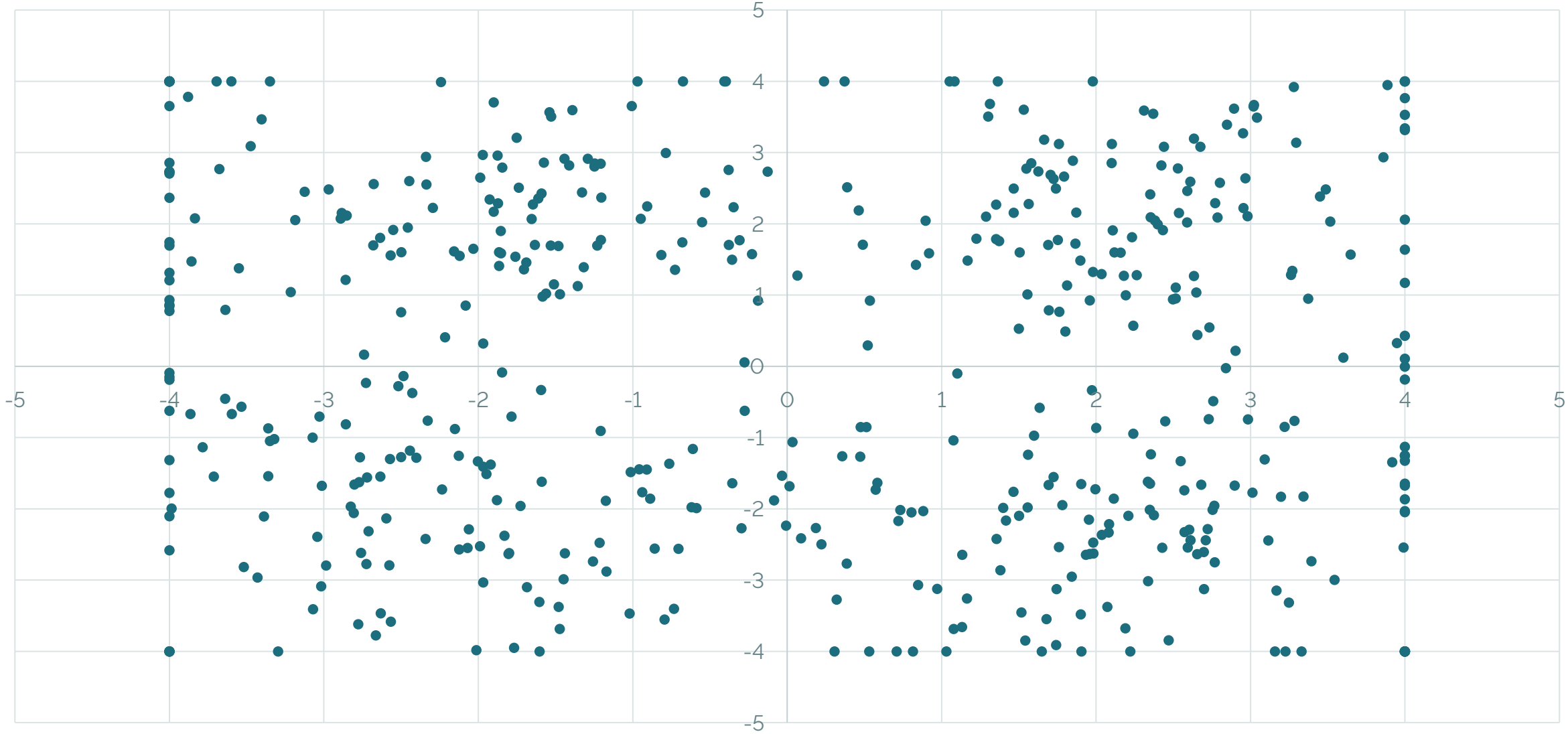
Co-clustering with covariates

SKIM

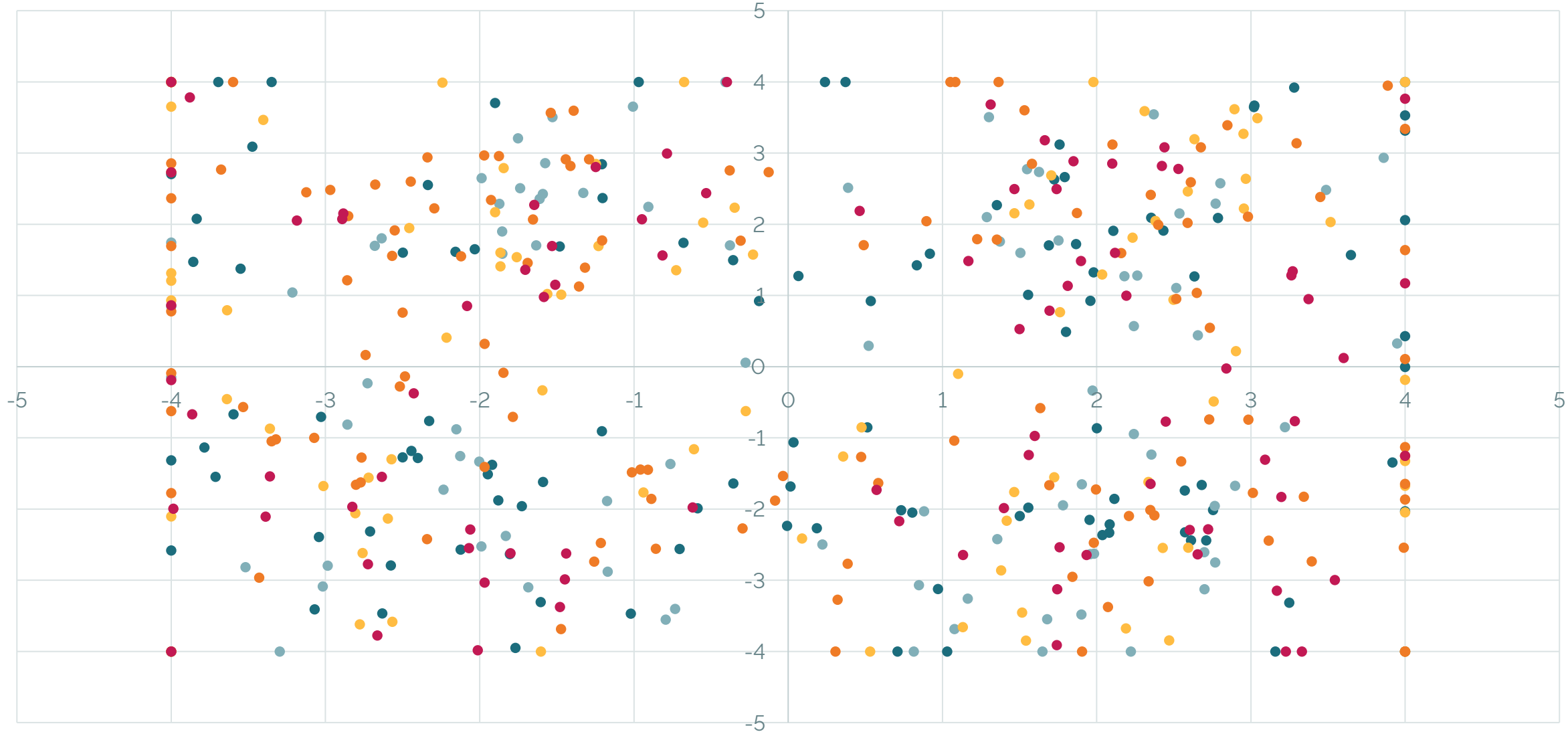
One easy way of clustering that I know

- One of the easiest ways of clustering is K-means

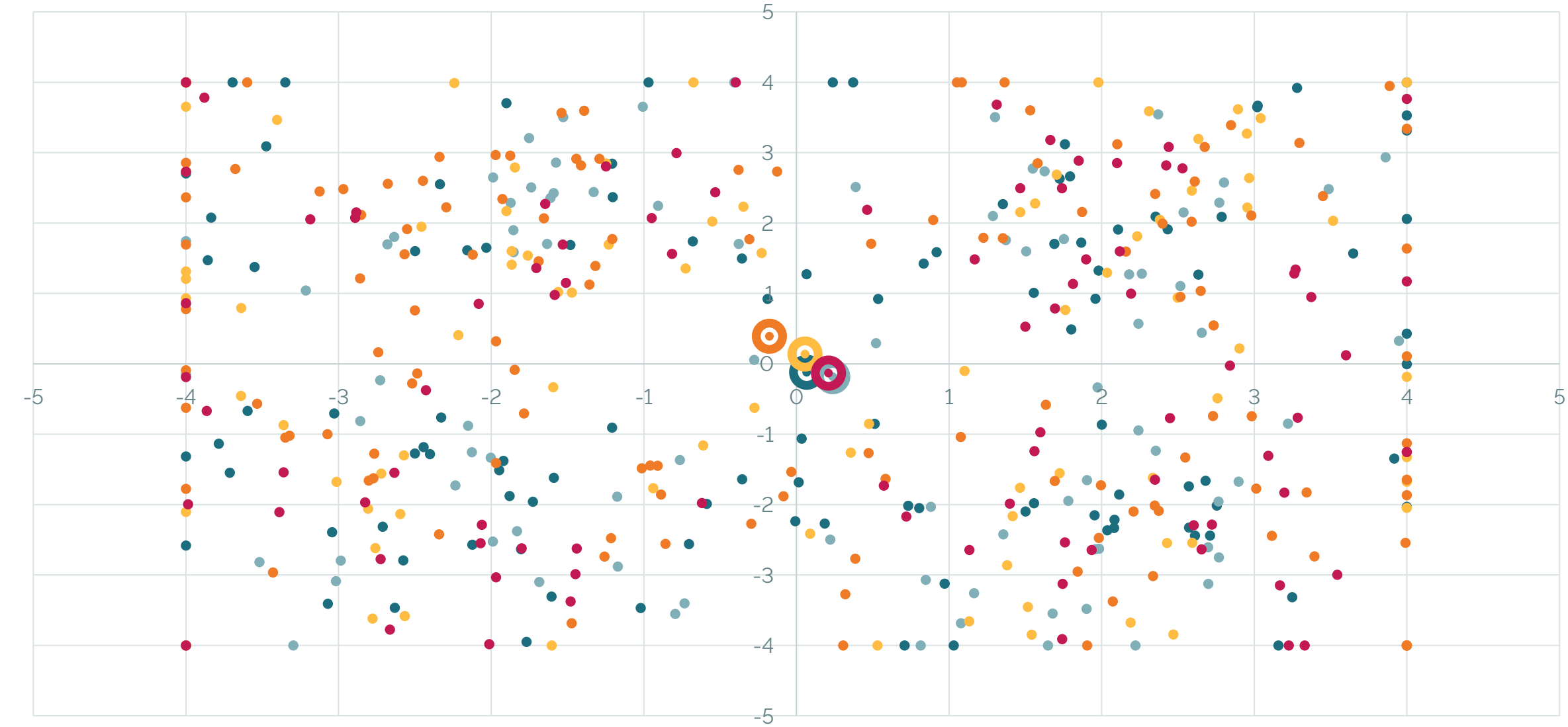
How does kmeans work?



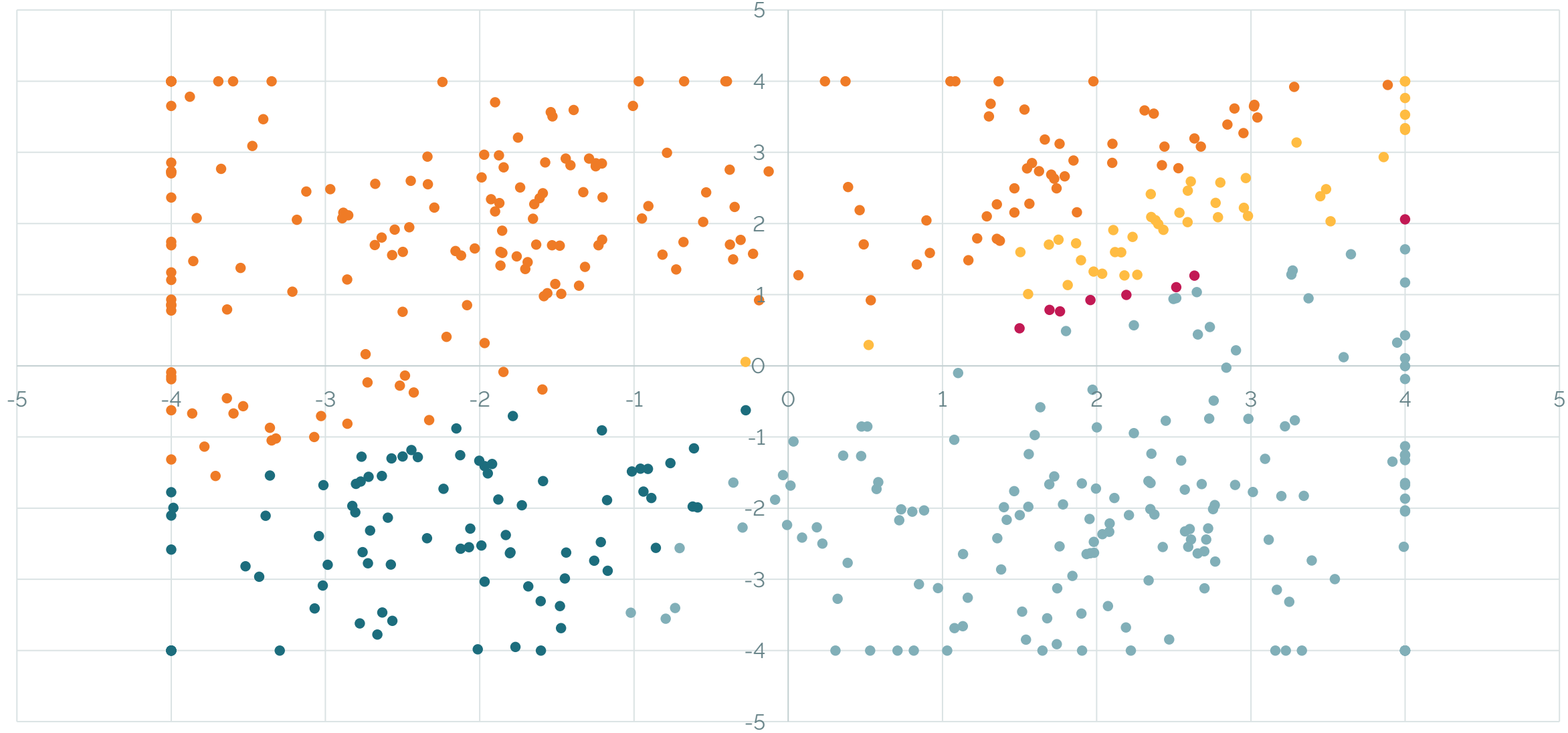
Assign random cluster membership



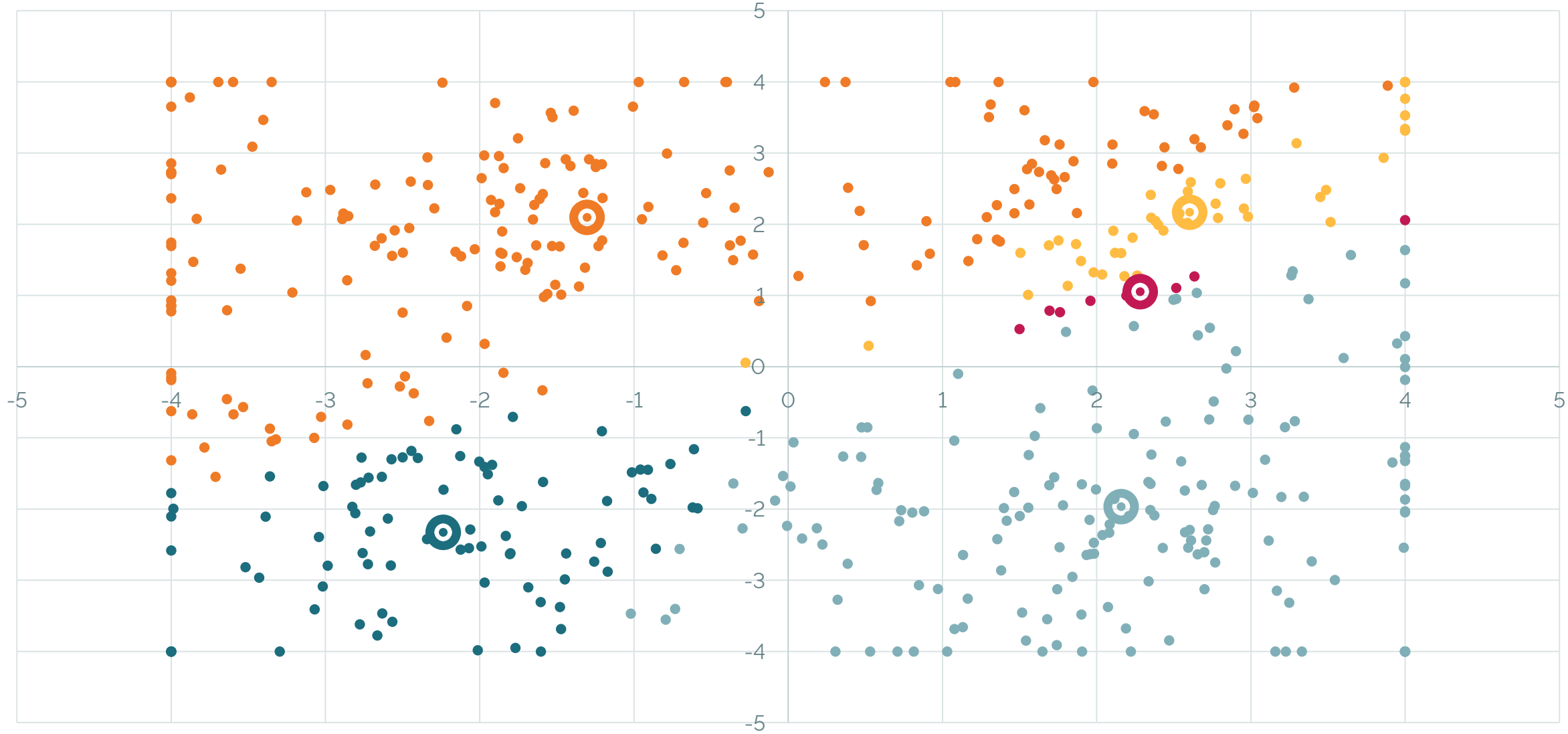
Calculate cluster centers



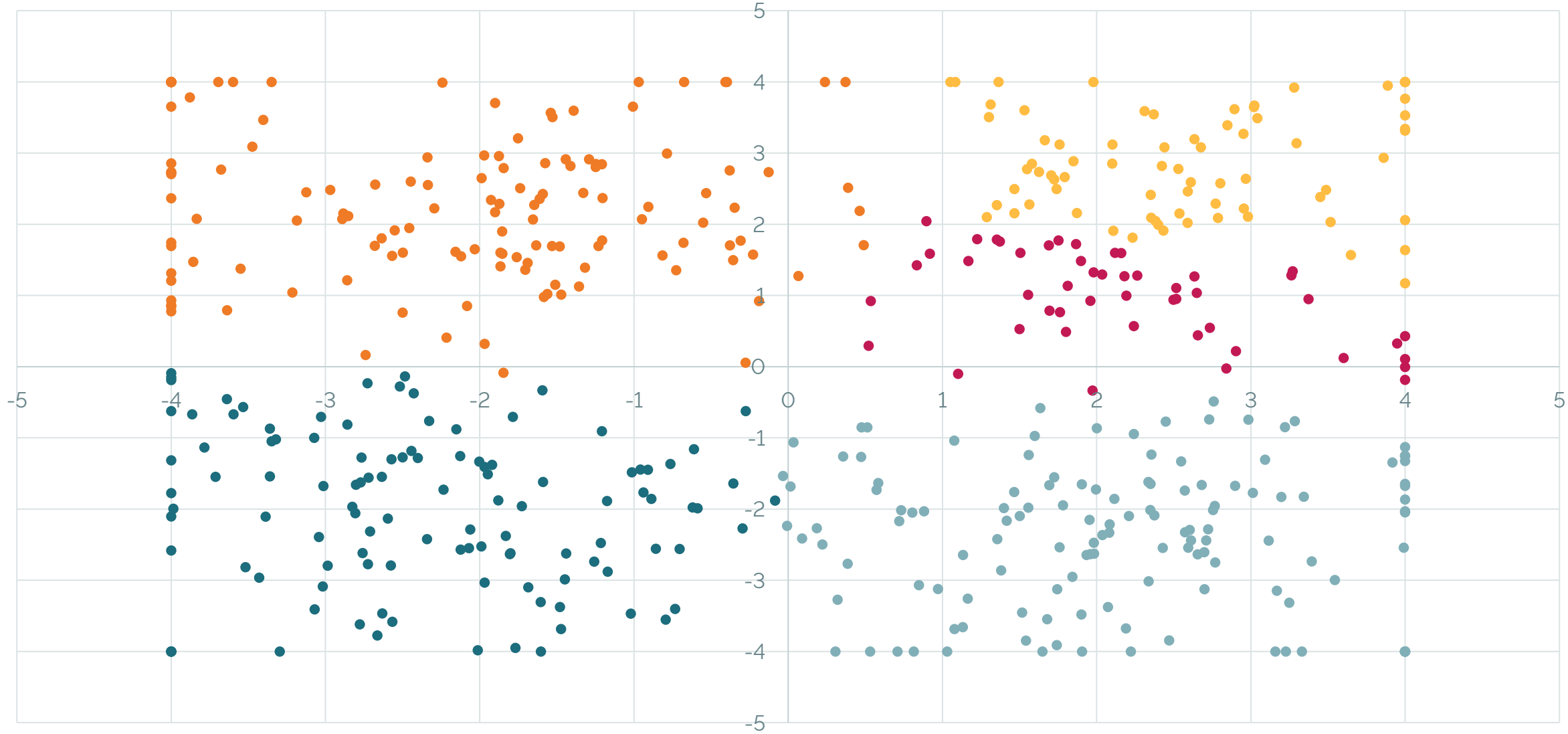
Re-assign cluster membership (assign to the closest center)



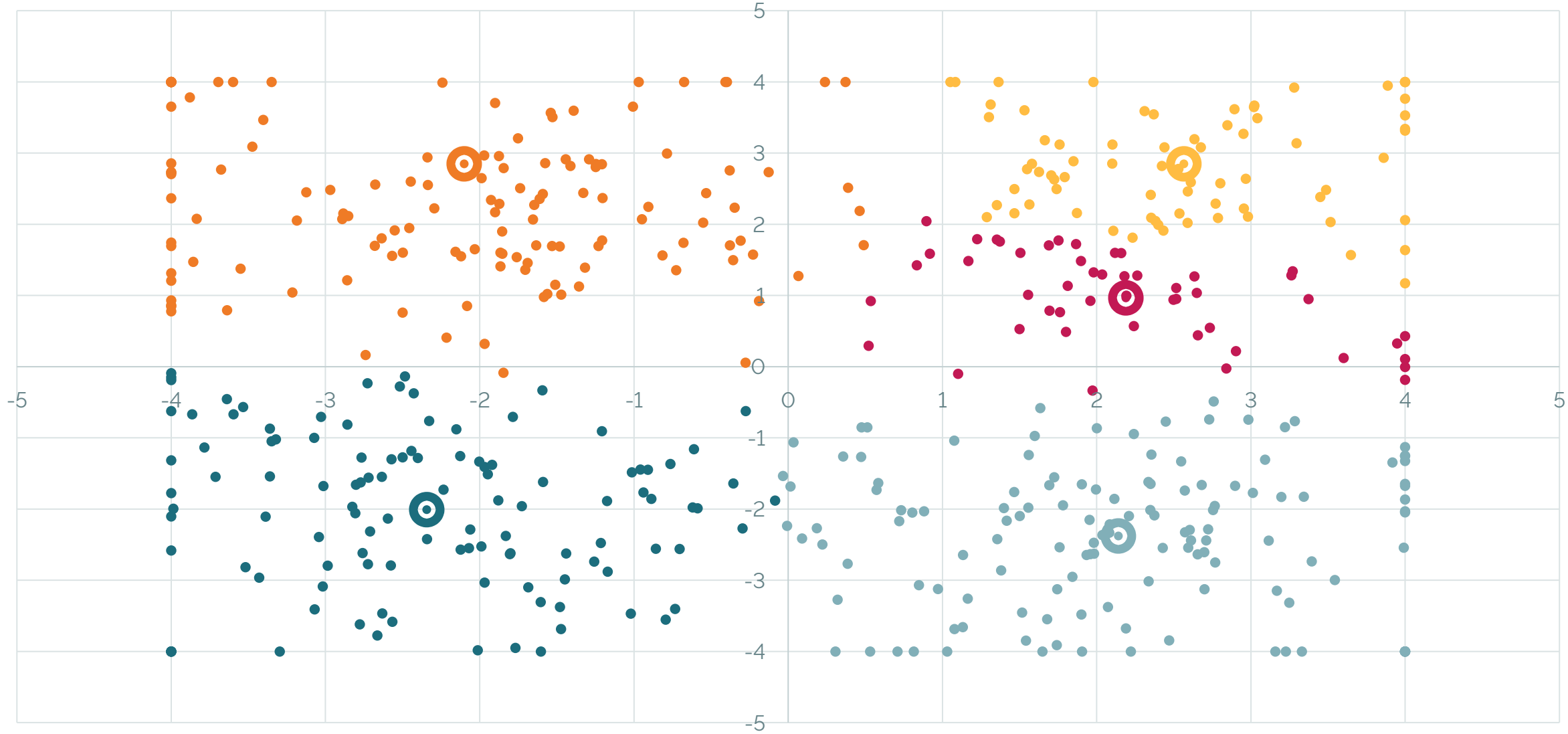
Calculate cluster centers



Re-assign cluster membership

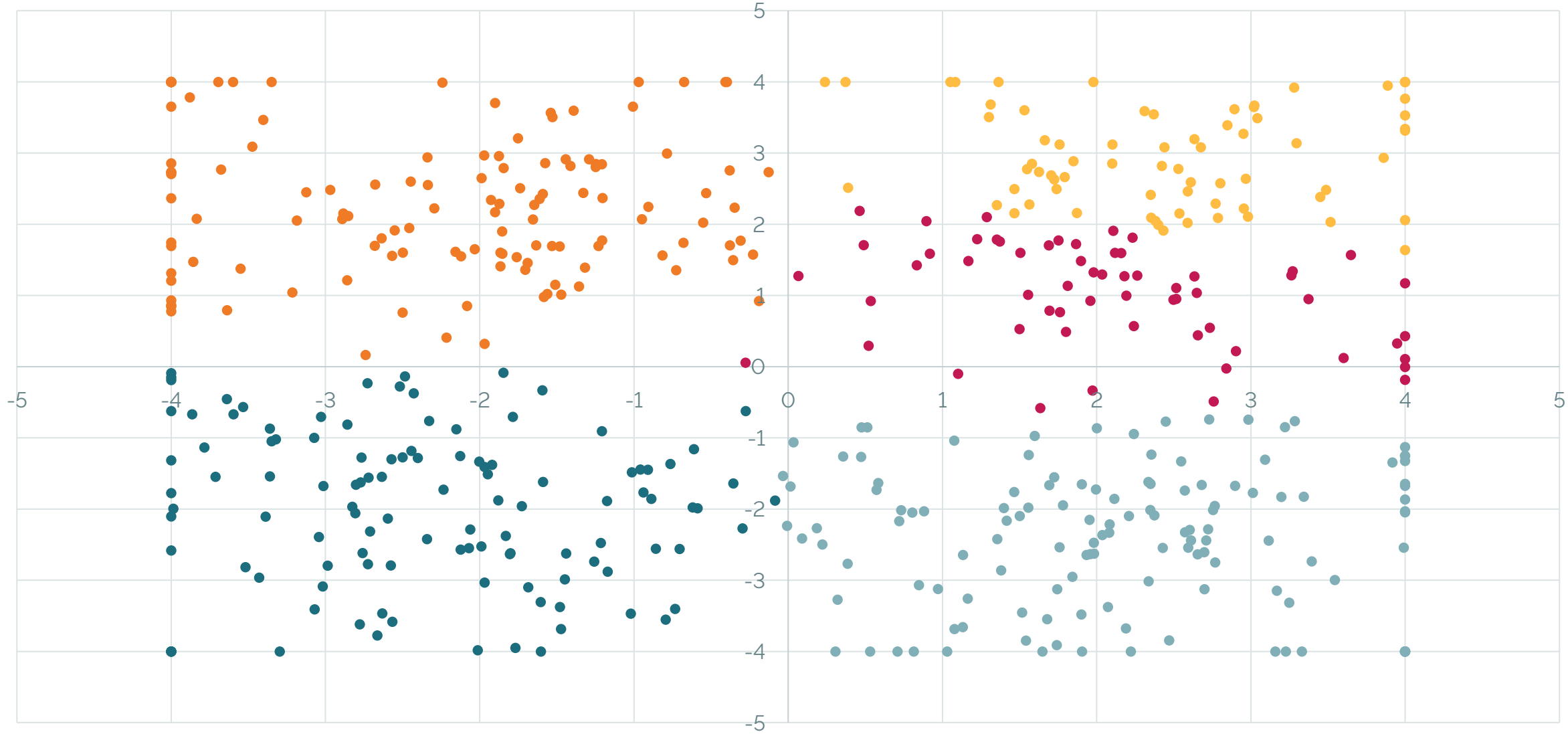


Calculate cluster centers

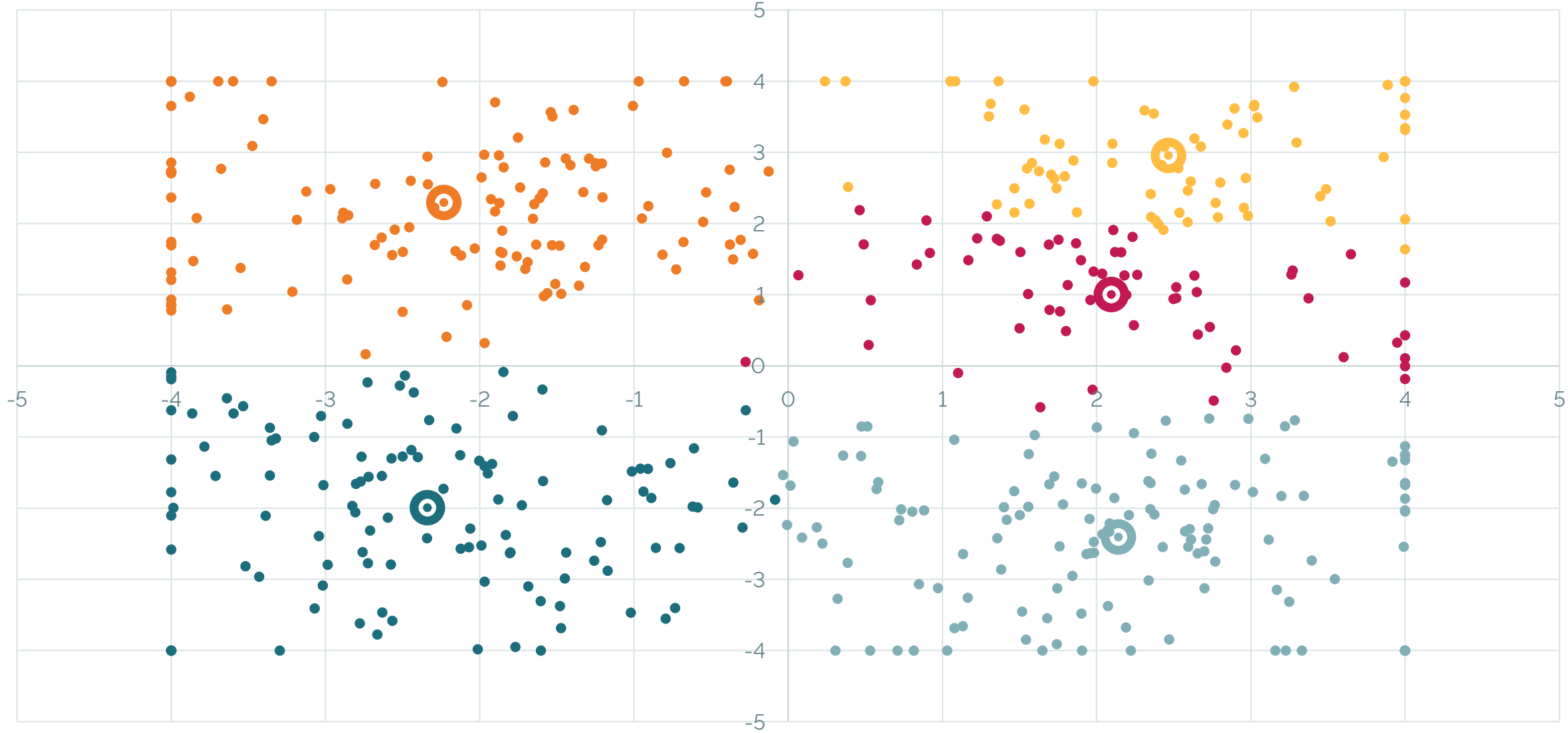


SKIM

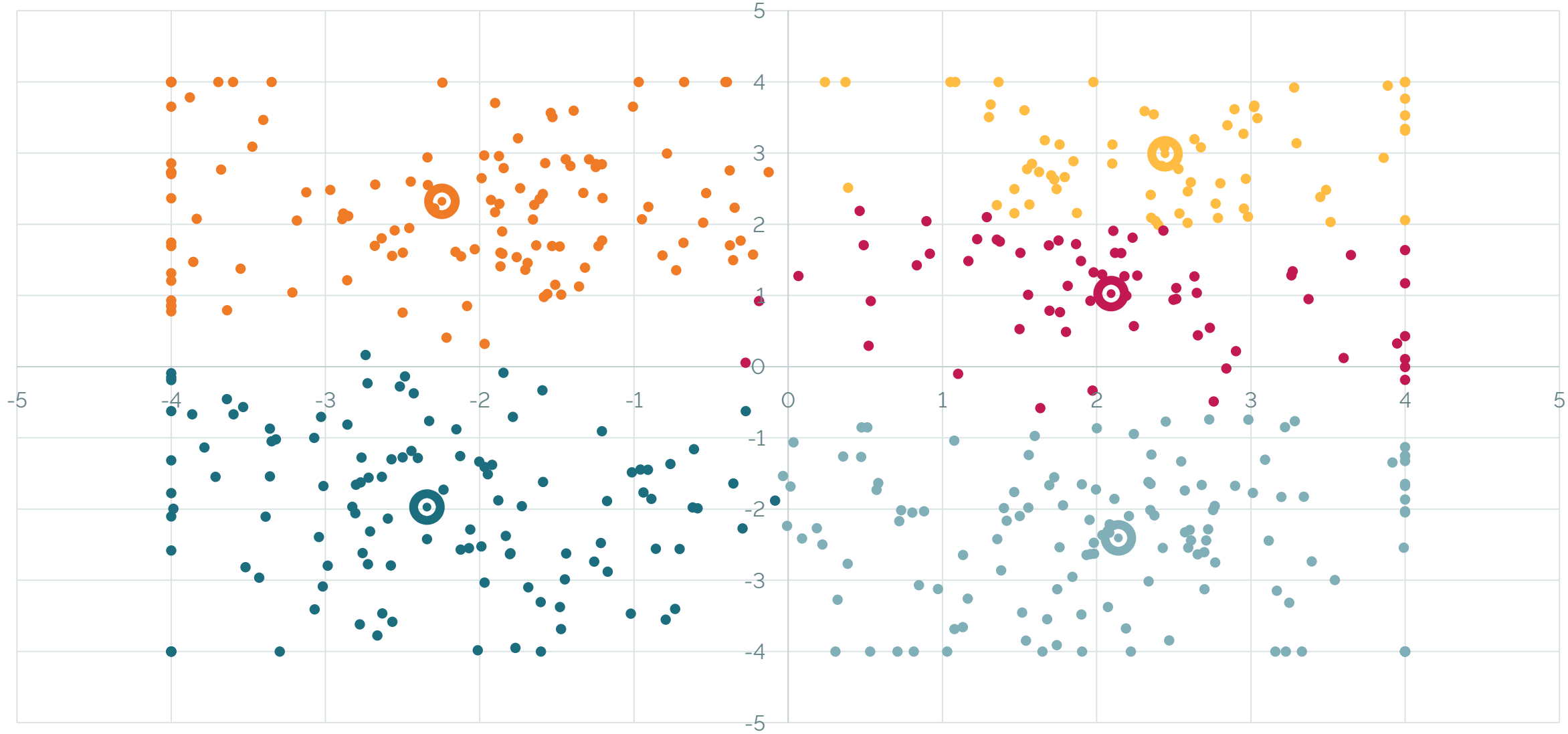
Re-assign cluster membership



Calculate cluster centers…

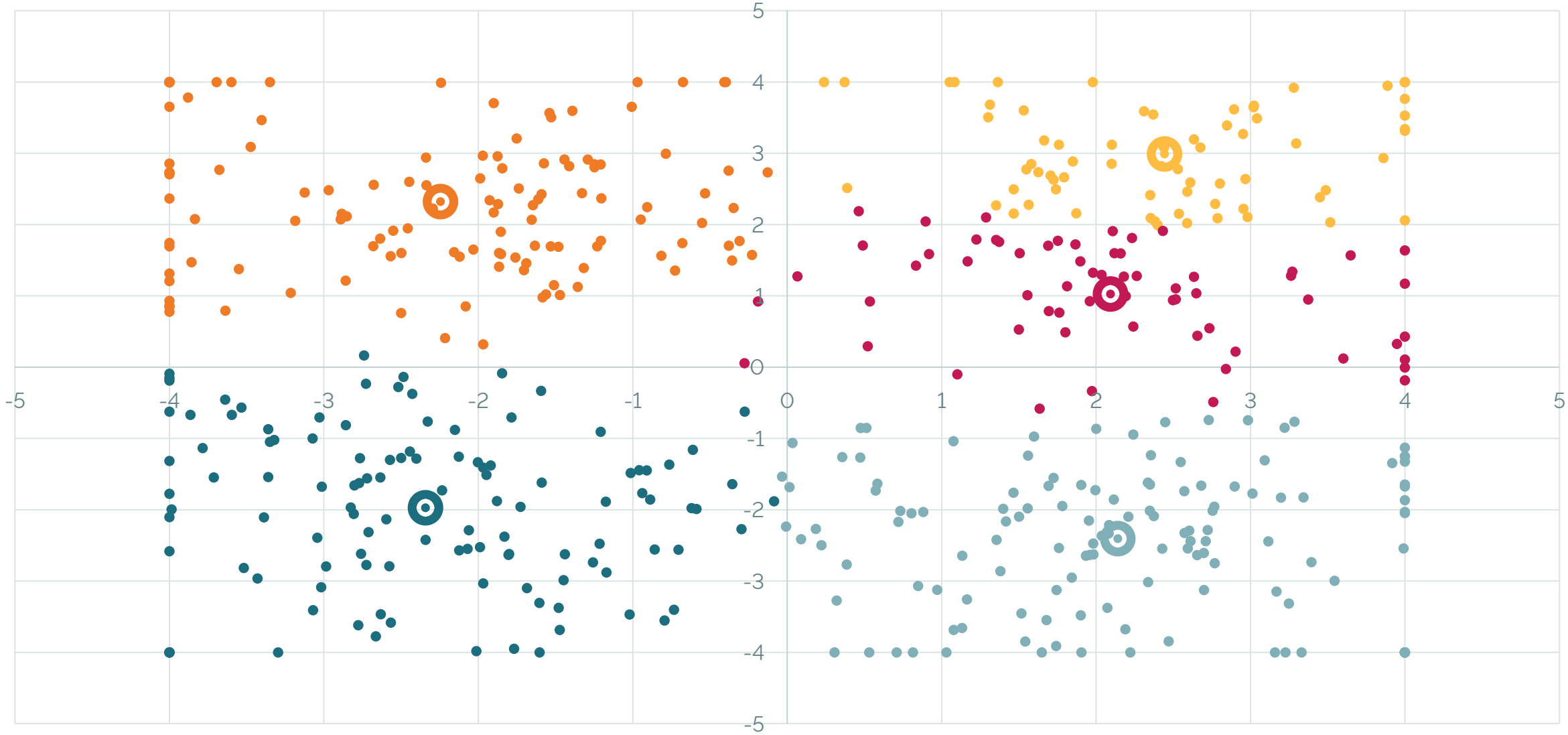


Until convergence

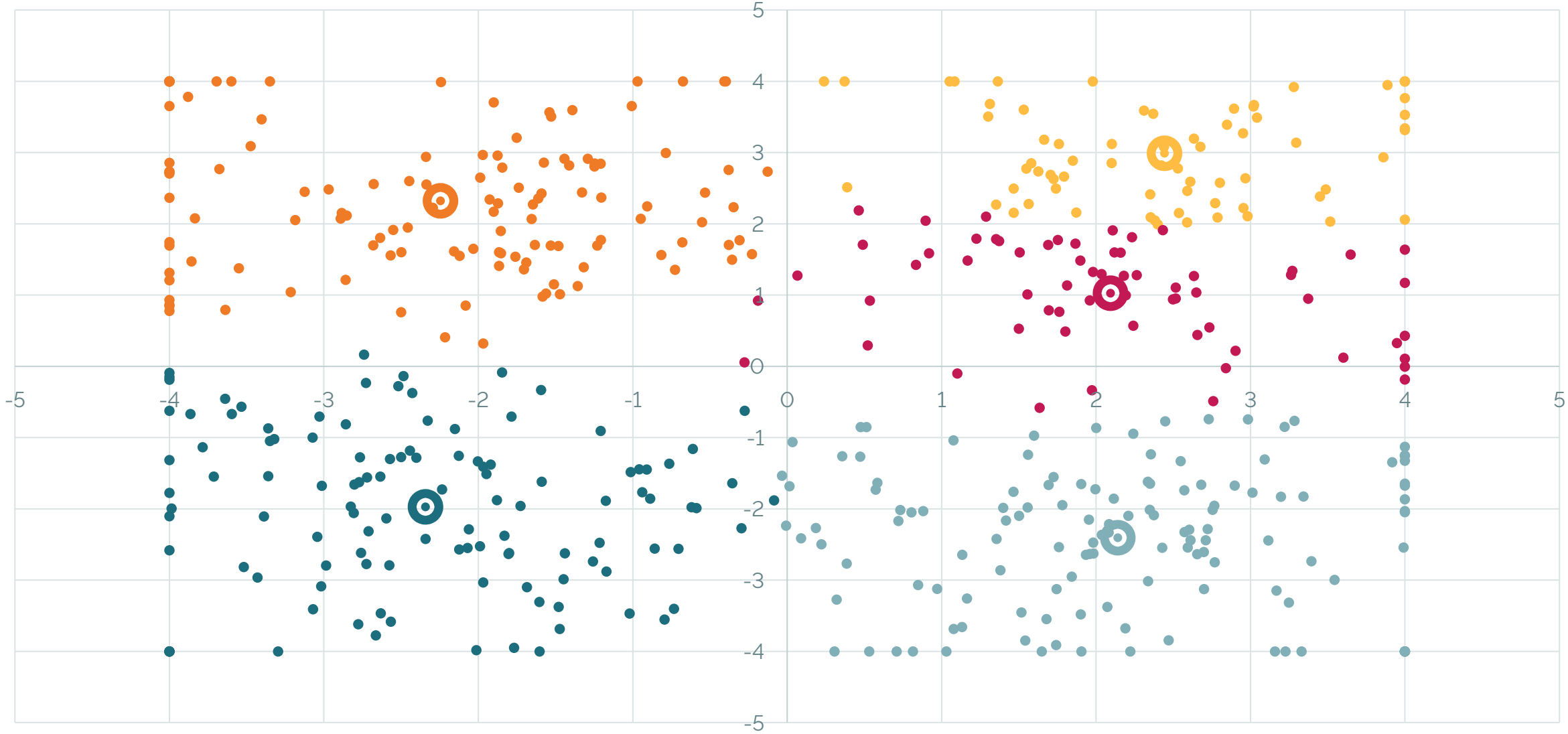


What did I cluster?

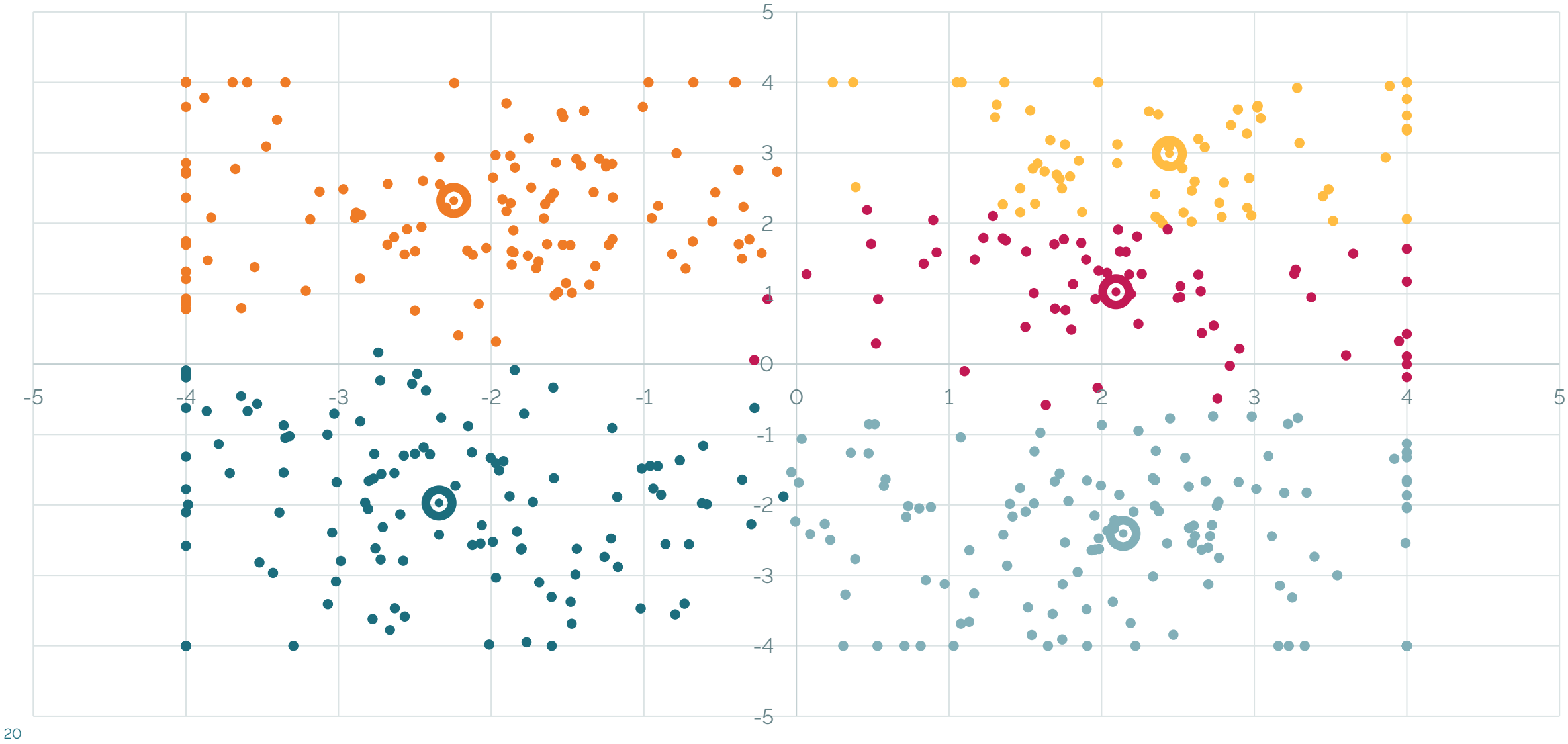
What was this data?



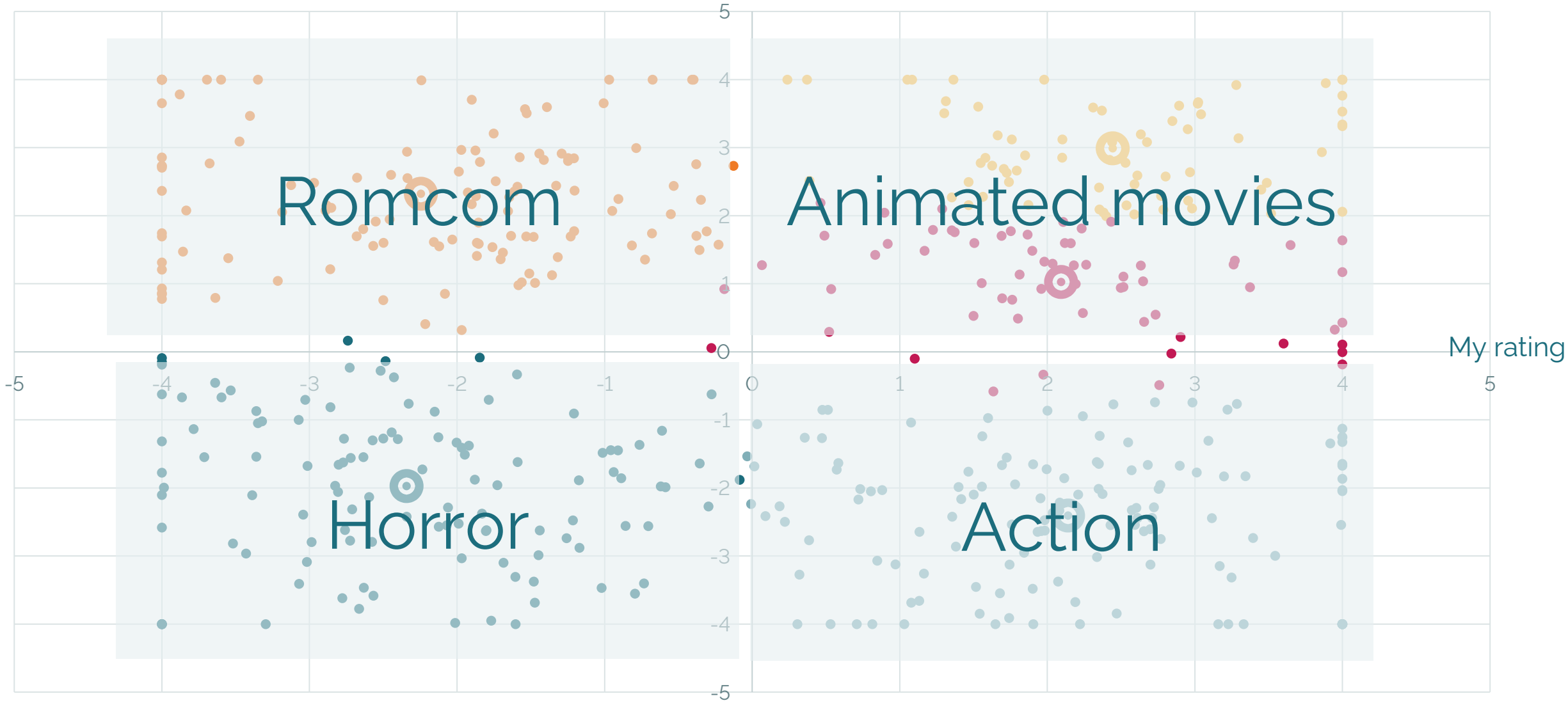
The rating of 2 movies by 500 people?



Nope, it's me and my wife's rating of 500 movies!



You can meaningfully cluster movies based on people's ratings too



Co-clustering with covariates

SKIM

Co-clustering

- Co-clustering is the art of simultaneously clustering respondents and their responses
- Responses could be e.g. maxdiff scores, ratings, rankings, yes/no, etc. whenever it makes sense to (directly) compare values

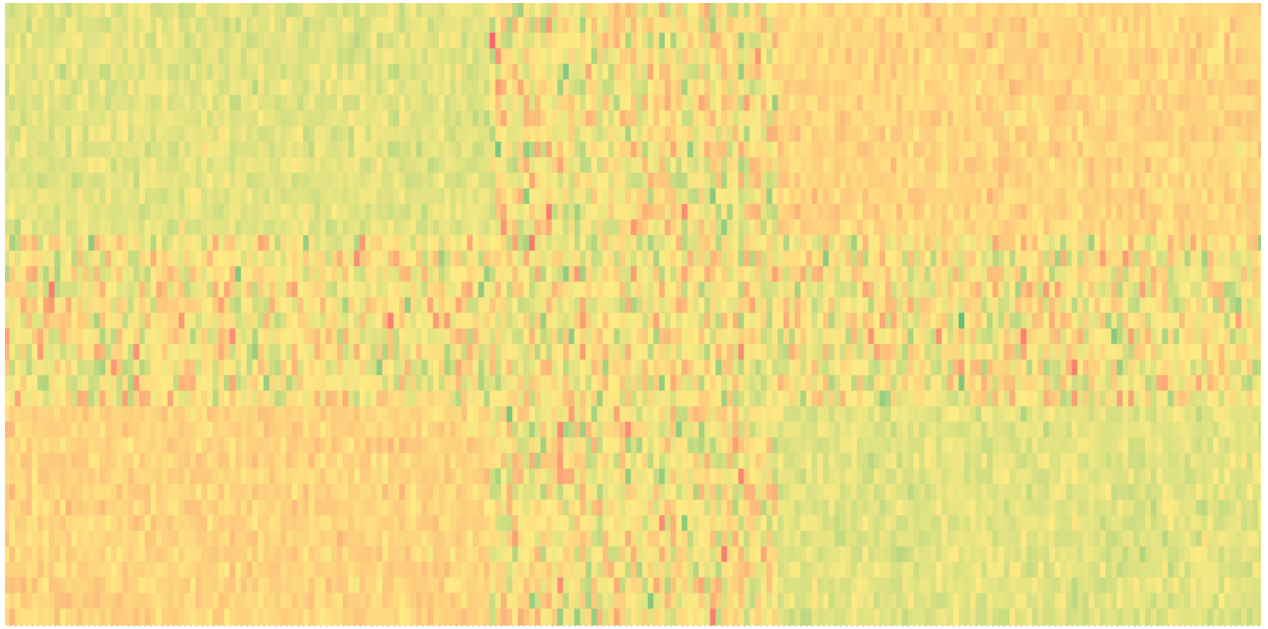
- The way I do it is by sequentially clustering
  - Cluster individual respondents into groups based on how similar their average is for a groups of responses
  - Cluster single responses based on how similar it has been answered by each of the respondent groups

- While typing this, I still confuse my self, so let's show it in a graph

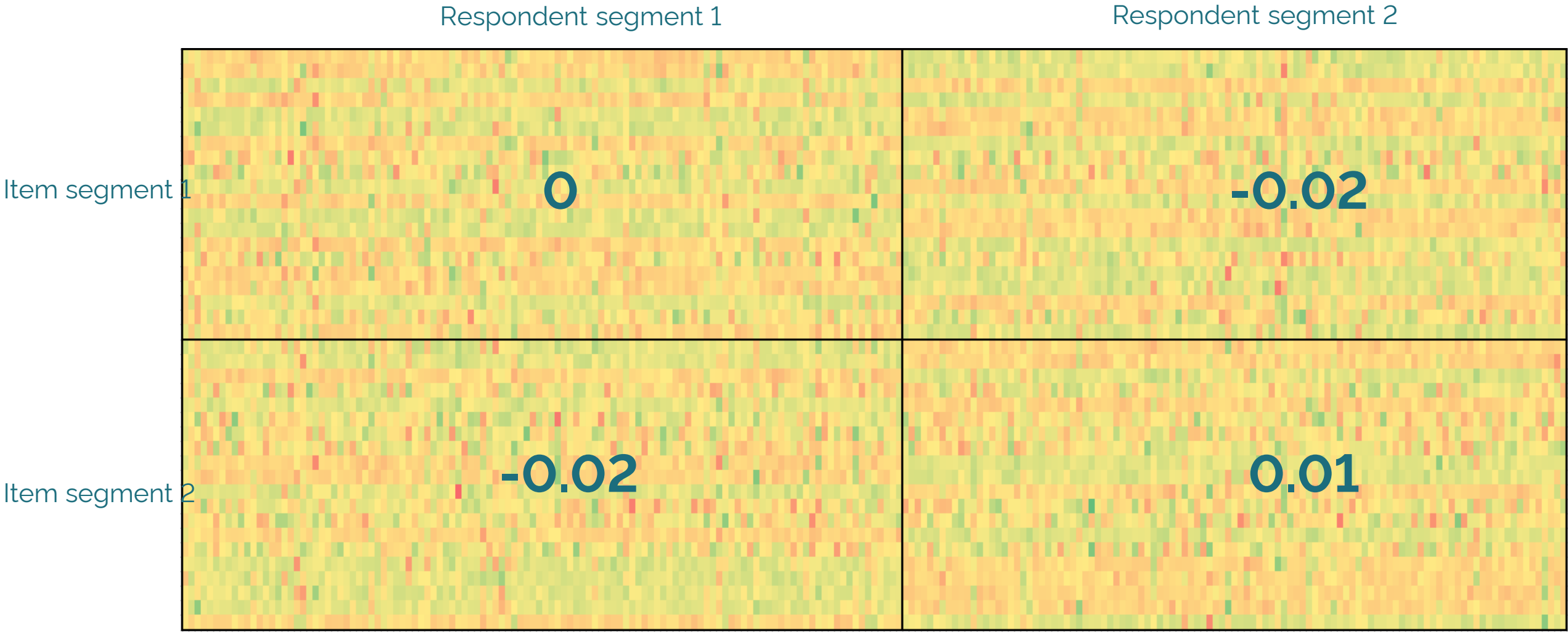
An example

- I created the following data
- 223 respondents, 40 items

	Average (s.d.)		
	40%	20%	40%
Items 1-15	2 (1)	0 (3)	-2 (1)
Items 16-26	0 (3)	0 (3)	0 (3)
Items 27-40	-2 (1)	0 (3)	2(1)

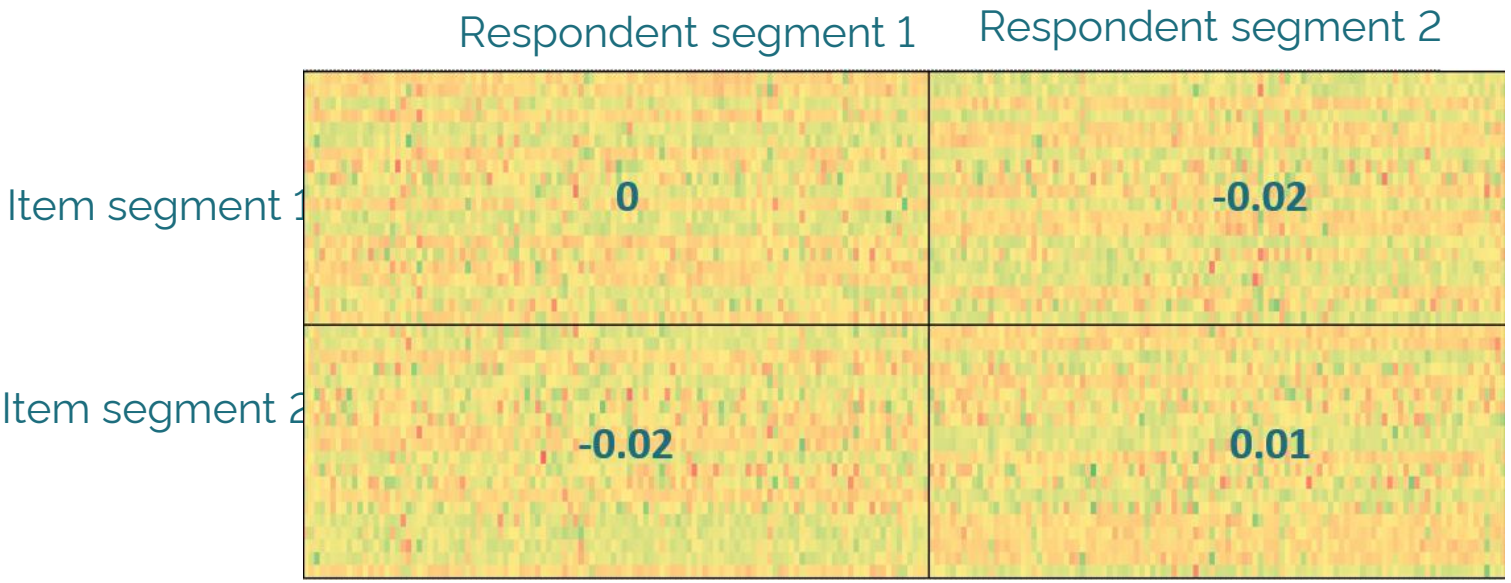


Initial segments. Both respondents and items start at a random segment



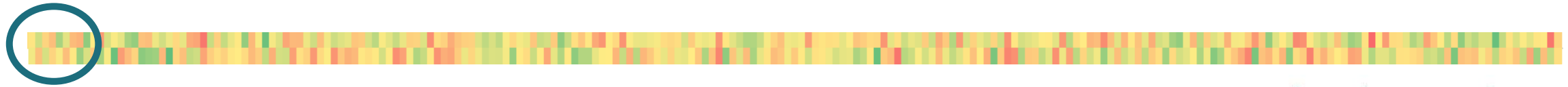
SKIM

Given the current segments, calculate the item segment average for each respondent



SKIM

And check which respondent segment fits better



-0.1	0.4	-0.3	-0.5	0.8
0.1	-0.5	0.4	-0.3	0.1

Update respondent group

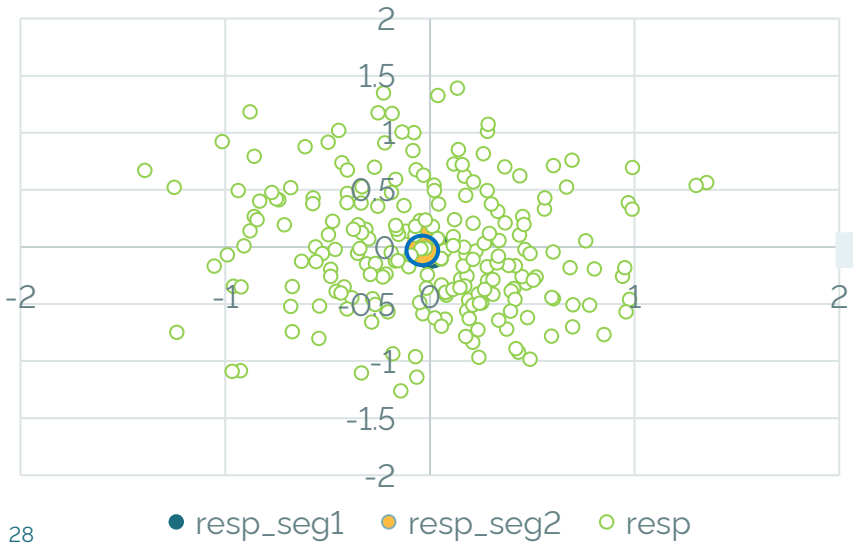
2 1 2 1
1

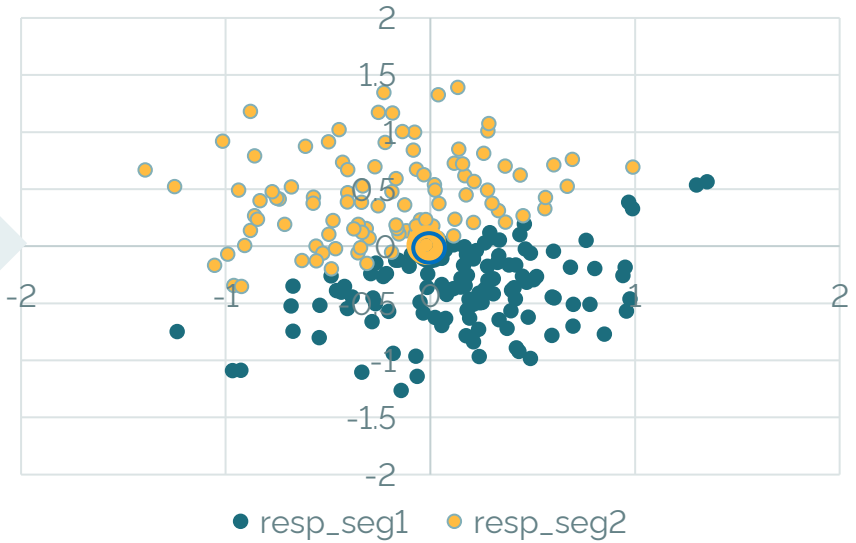
	Respondent segment 1	Respondent segment 2
Item segment 1	0	-0.02
Item segment 2	-0.02	0.01

SKIM

And check which respondent segment fits better

	Respondent segment 1	Respondent segment 2
Item segment 1	0	-0.02
Item segment 2	-0.02	0.01



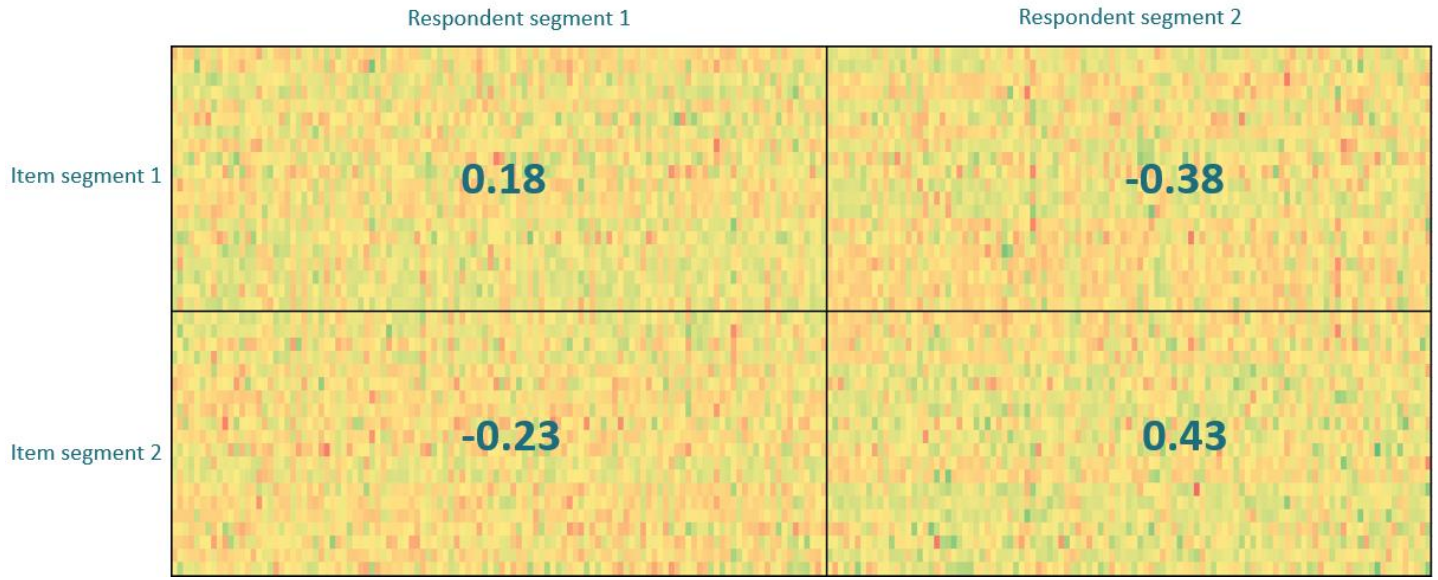


SKIM

Update the numbers, as we now have updated respondent segments

SKIM

Now, we want to update the item segmentation. For each item, calculate the average within each respondent group

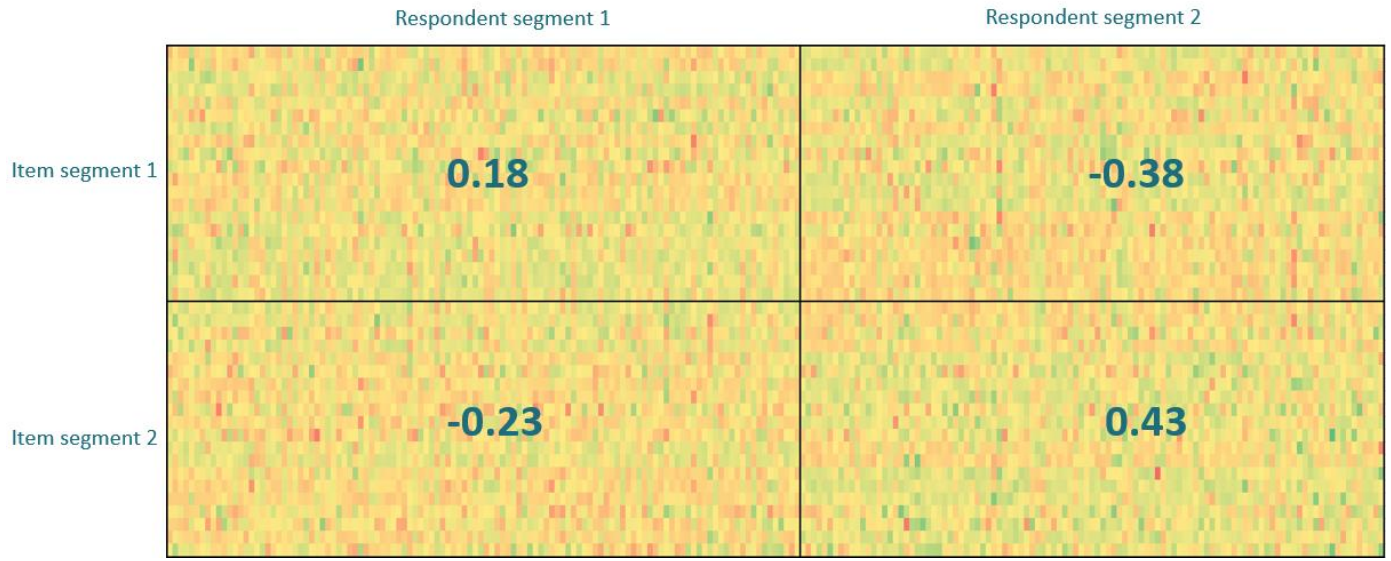


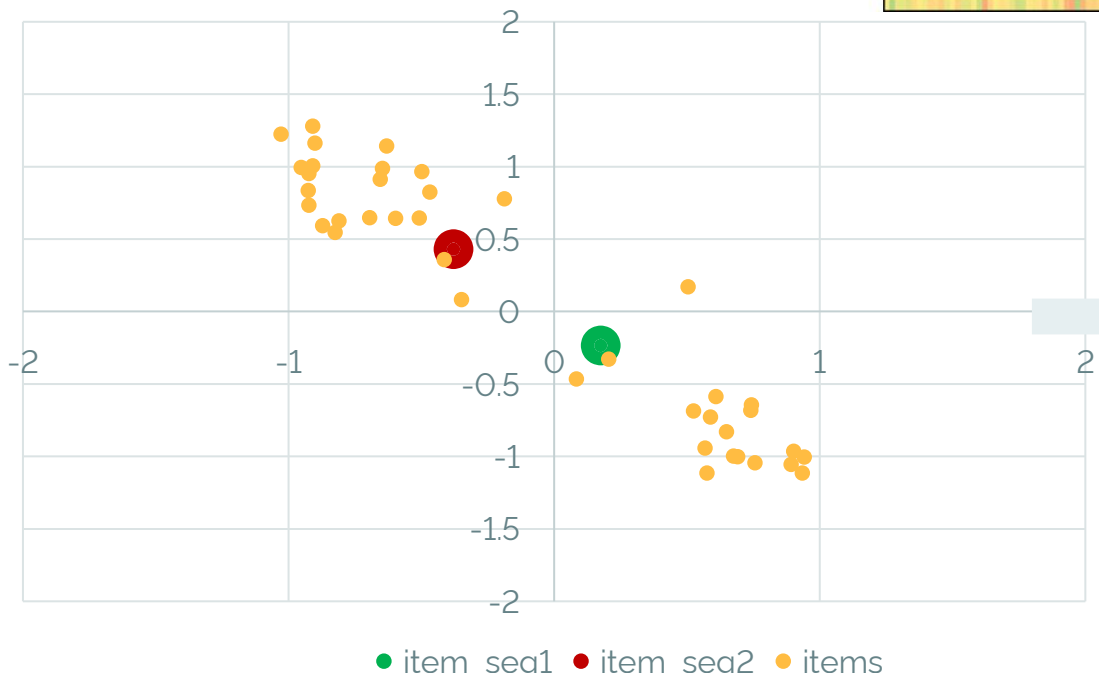
-0.9	0.6
-0.7	1.0
0.9	-1.1
1.0	-1.0
-0.6	0.7
0.2	-0.3
0.6	-0.9
-1.0	0.8
0.5	0.1
0.1	-0.5
-0.7	0.7
-0.5	1.0
-0.7	1.0
1.0	-1.1
0.5	-0.6
0.7	-1.0
0.6	-1.1
0.9	-1.0
0.8	-1.1
0.8	-0.7
0.7	-1.0
0.8	-0.7
-0.3	0.0
0.7	-0.8
-0.7	1.2
-0.4	0.4
-0.9	1.0
-0.9	1.0
-0.6	0.6
-0.9	0.8
-0.8	0.5
0.6	-0.5
0.6	-0.8
-0.9	1.1
-0.9	1.3
-1.1	1.3
-0.7	0.5
-0.4	0.8
-1.0	1.0
-0.2	0.7

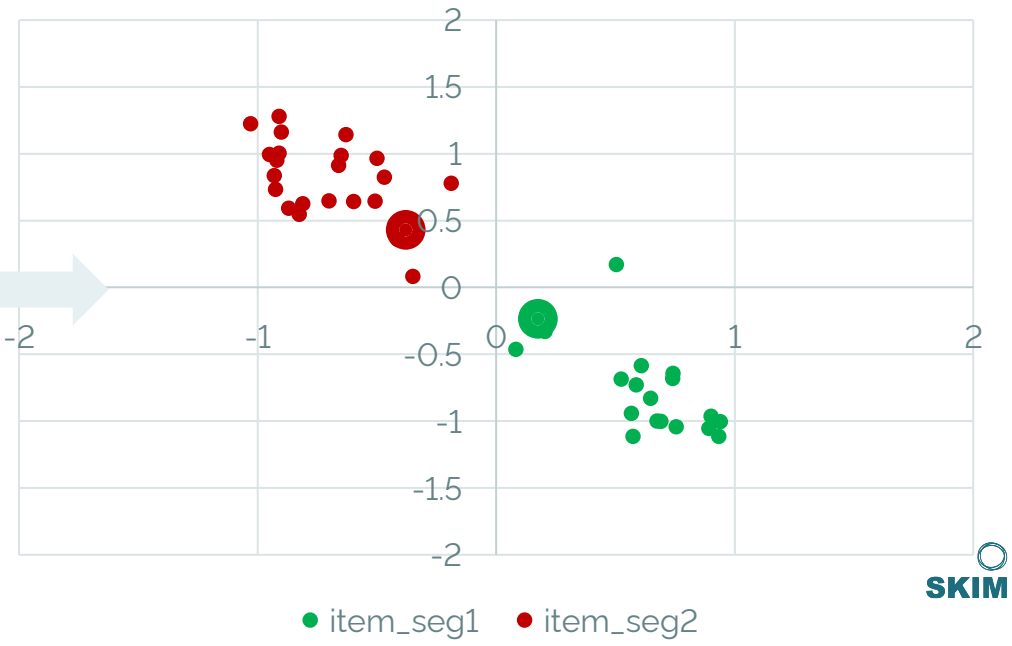
SKIM

And check which item segment fits each of the items better

-0.9	0.6
-0.7	1.0
0.9	-1.1
1.0	-1.0
-0.6	0.7
0.2	-0.3
0.6	-0.9
-1.0	0.8
0.5	0.1
0.1	-0.5
-0.7	0.7
-0.5	1.0
-0.7	1.0
1.0	-1.1
0.5	-0.6
0.7	-1.0
0.6	-1.1
0.9	-1.0
0.8	-1.1
0.8	-0.7
0.7	-1.0
0.8	-0.7
-0.3	0.0
0.7	-0.8
-0.7	1.2
-0.4	0.4
-0.9	1.0
-0.9	1.0
-0.6	0.6
-0.9	0.8
-0.8	0.5
0.6	-0.5
0.6	-0.8
-0.9	1.1
-0.9	1.3
-1.1	1.3
-0.7	0.5
-0.4	0.8
-1.0	1.0
-0.2	0.7





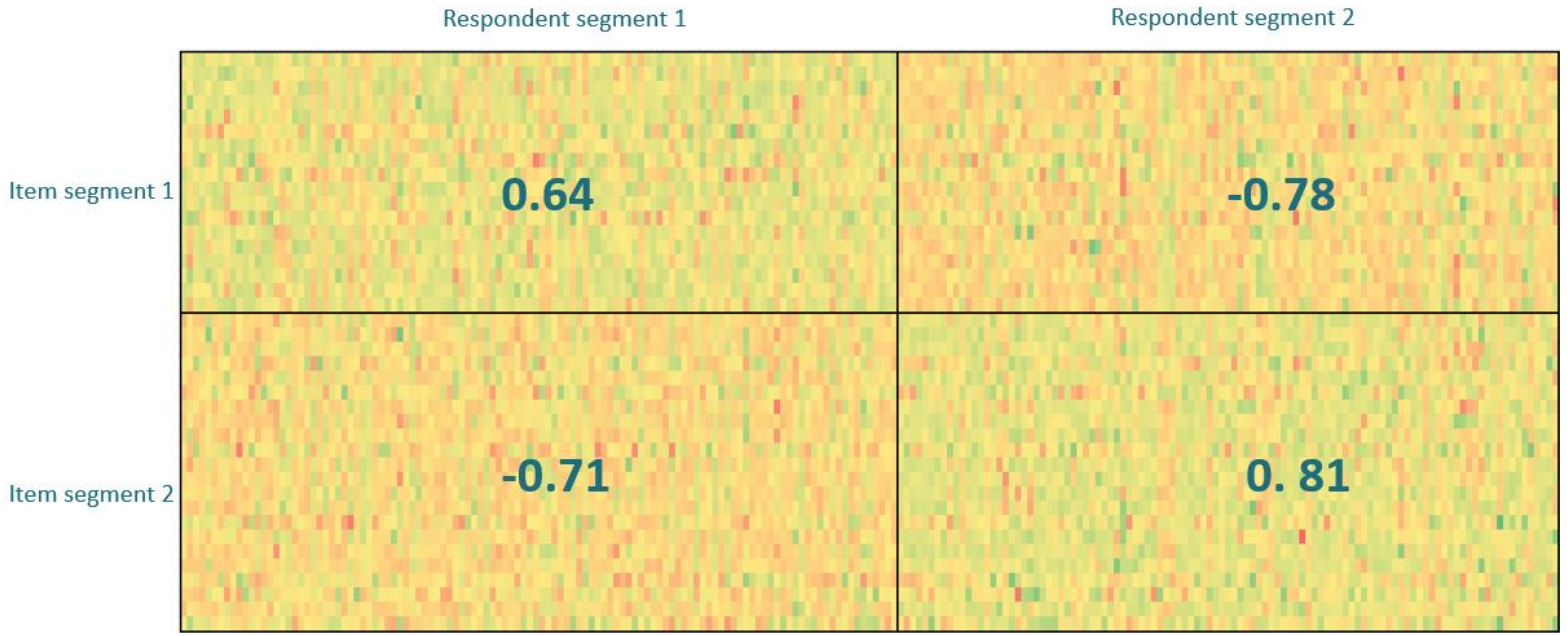


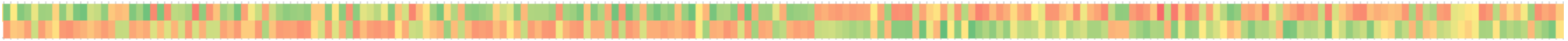
SKIM

Update the numbers, as we now also have new item segments

SKIM

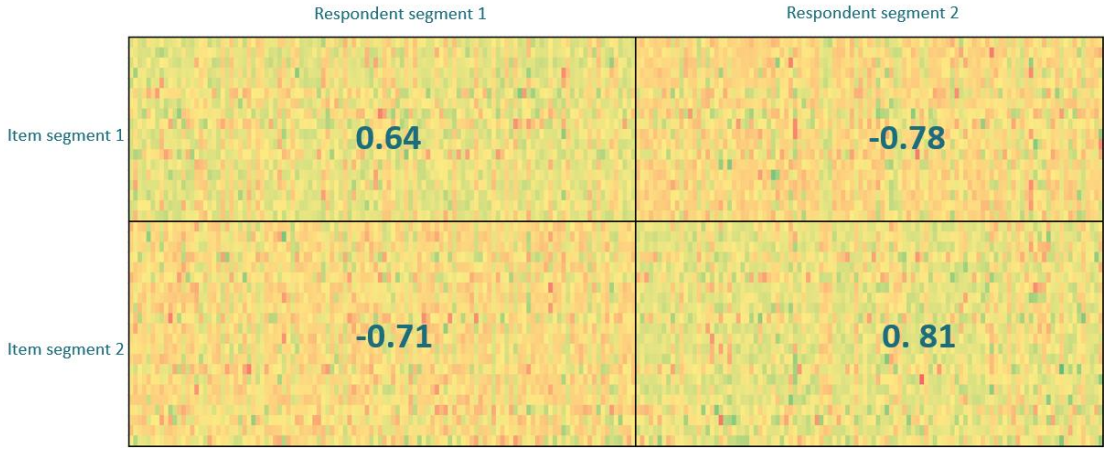
Now, update the respondent segments again. For each respondent, calculate the average score within each item segment

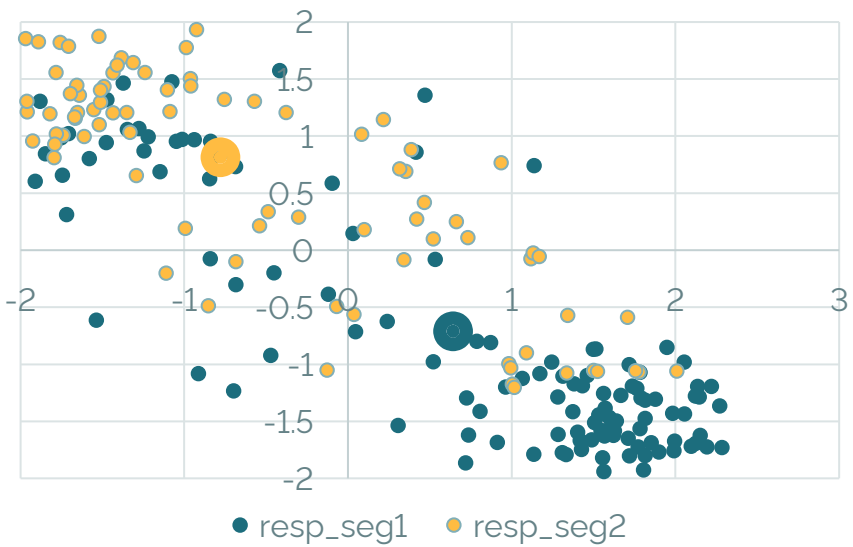


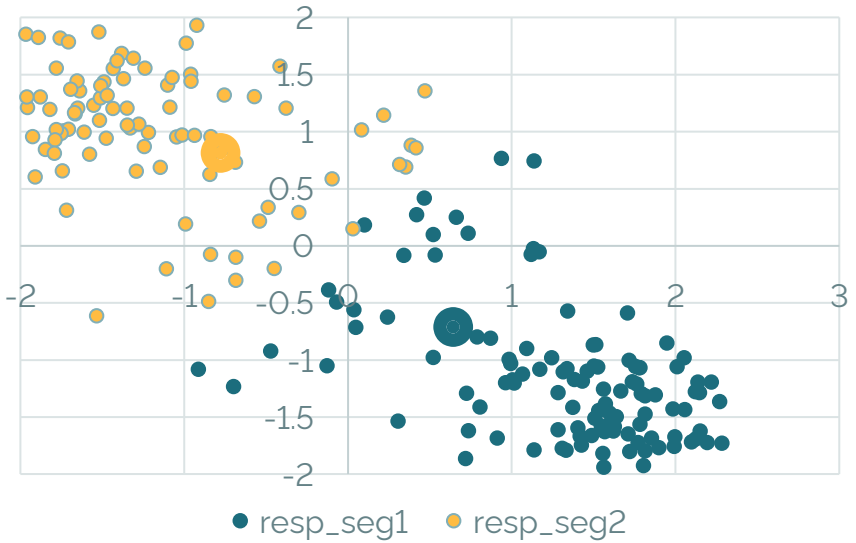


SKIM

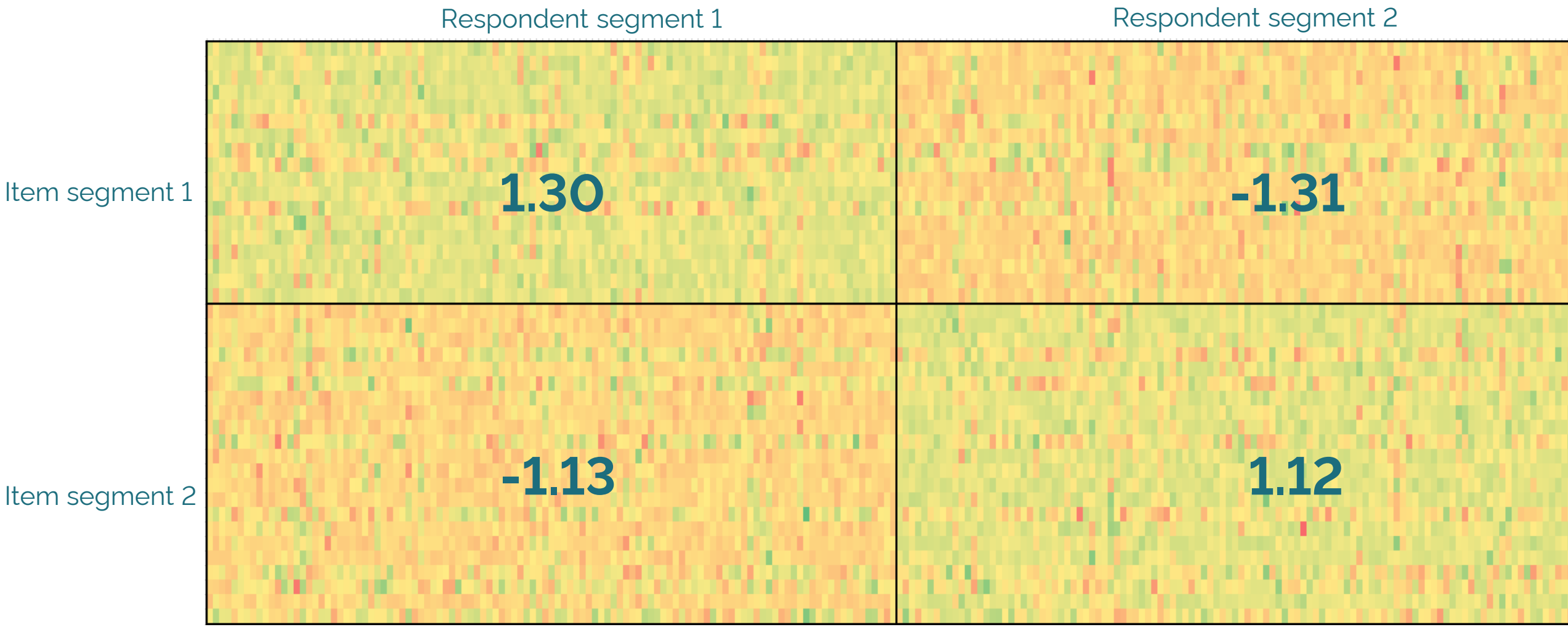
Reassign the respondents based on the updated item segment scores and therefore respondent segment means





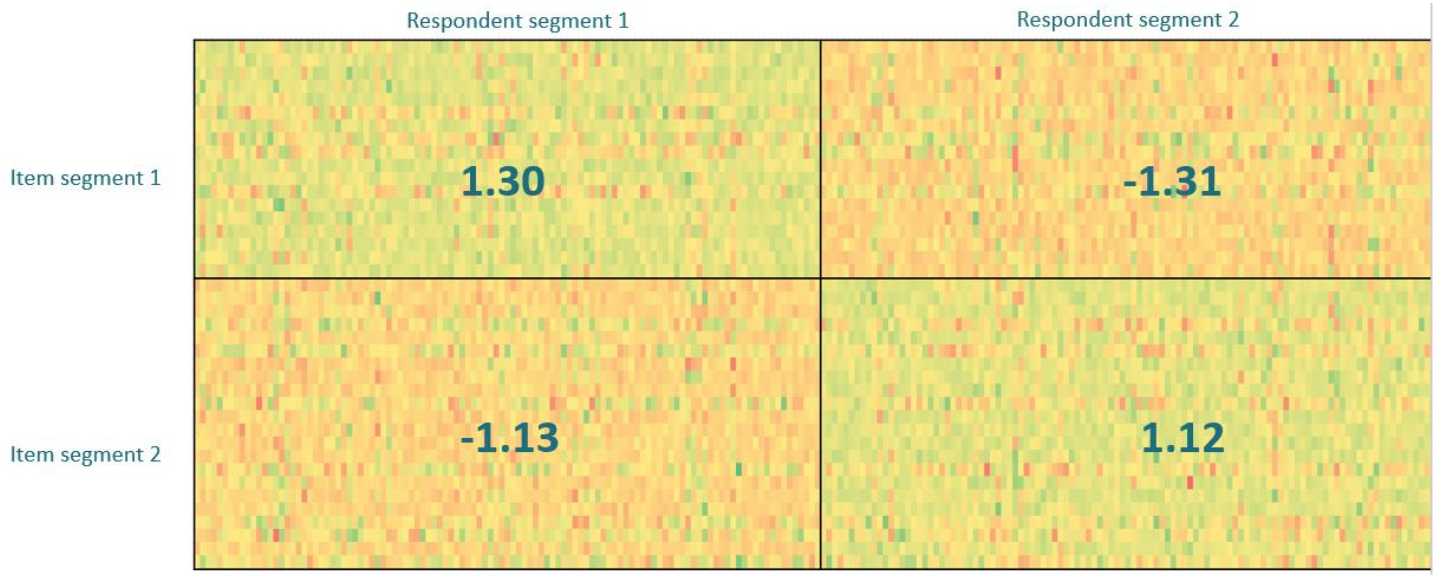


Update the numbers again



SKIM

Update the item segments again.
For each item, calculate the average within each respondent group

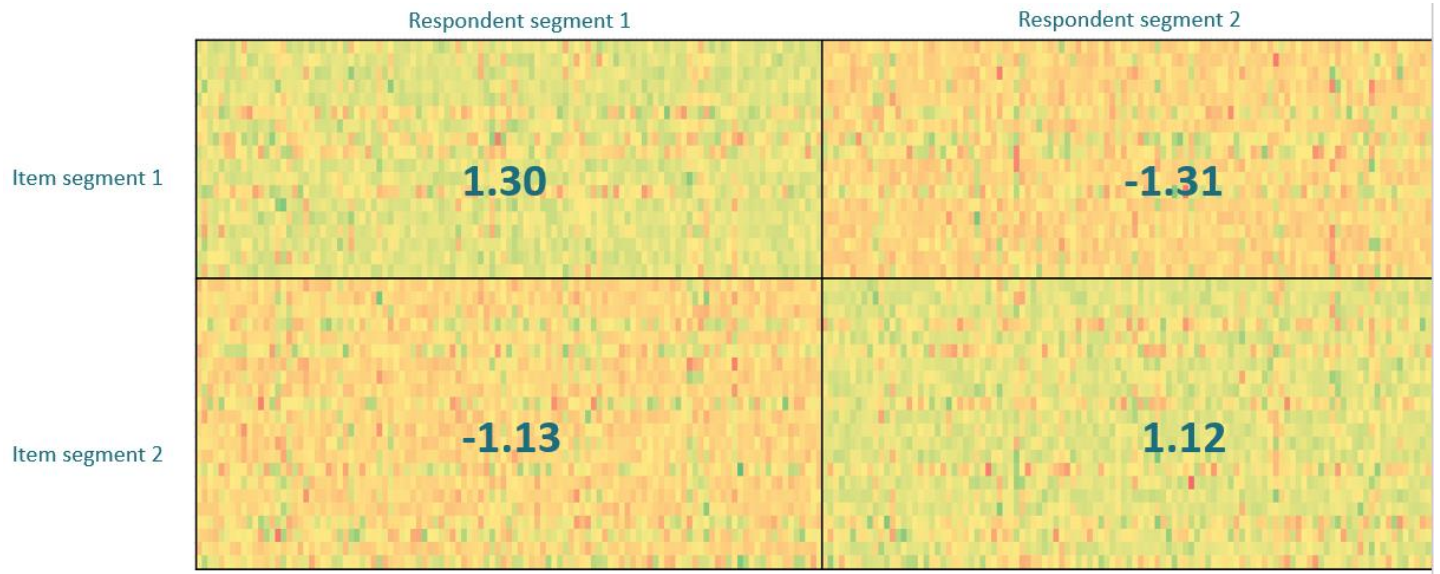


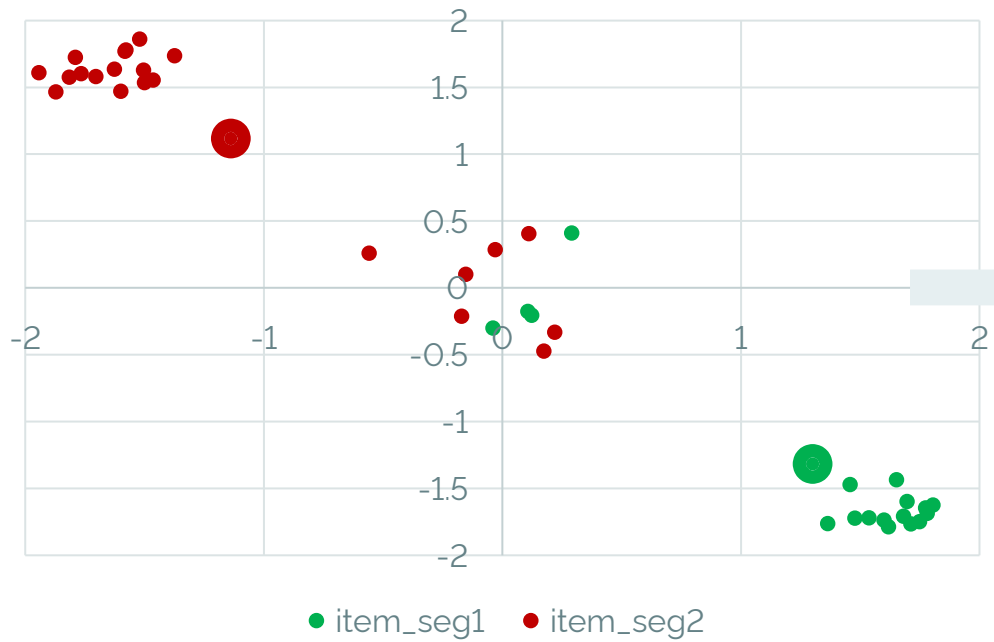
1.5	-1.7
1.5	-1.5
1.7	-1.6
1.8	-1.6
1.7	-1.8
0.1	-0.2
1.5	-1.7
0.3	0.4
0.0	-0.3
1.7	-1.7
1.8	-1.6
0.1	-0.2
1.7	-1.7
1.6	-1.8
1.4	-1.8
1.8	-1.7
1.6	-1.7
1.7	-1.4
-1.9	1.5
-1.5	1.6
-1.5	1.5
0.2	-0.5
-1.4	1.7
0.2	-0.3
-1.9	1.6
-1.6	1.5
-1.8	1.7
-0.2	0.1
-1.8	1.6
-1.8	1.6
-1.5	1.9
-1.6	1.8
-0.2	-0.2
-1.5	1.6
-1.6	1.8
-1.6	1.6
-0.6	0.3
0.0	0.3
-1.7	1.6
0.1	0.4

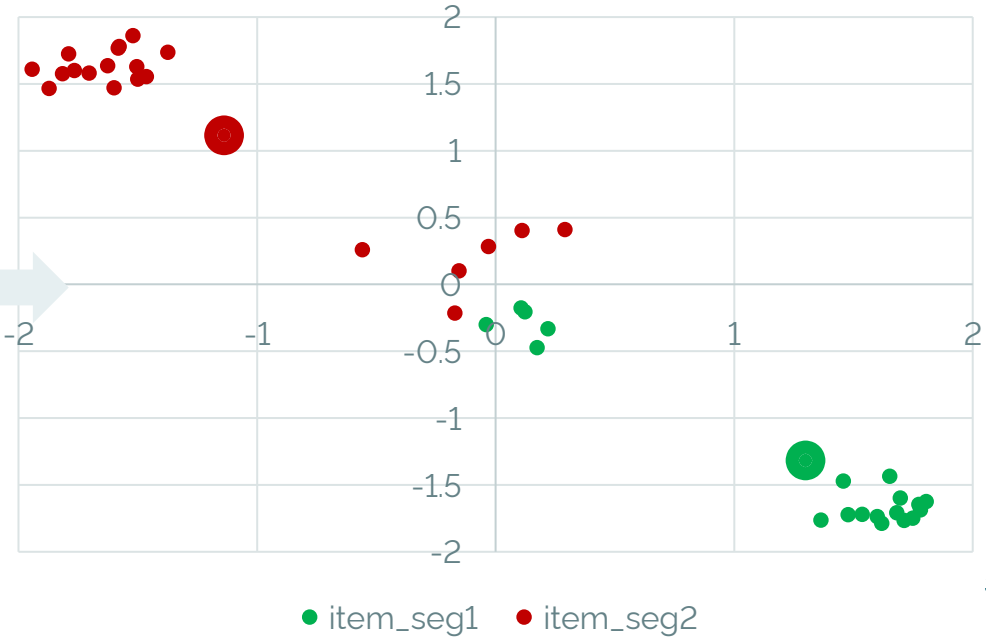
SKIM

And check which item segment fits each of the items better

1.5	-1.7
1.5	-1.5
1.7	-1.6
1.8	-1.6
1.7	-1.8
0.1	-0.2
1.5	-1.7
0.3	0.4
0.0	-0.3
1.7	-1.7
1.8	-1.6
0.1	-0.2
1.7	-1.7
1.6	-1.8
1.4	-1.8
1.8	-1.7
1.6	-1.7
1.7	-1.4
-1.9	1.5
-1.5	1.6
-1.5	1.5
0.2	-0.5
-1.4	1.7
0.2	-0.3
-1.9	1.6
-1.6	1.5
-1.8	1.7
-0.2	0.1
-1.8	1.6
-1.8	1.6
-1.5	1.9
-1.6	1.8
-0.2	-0.2
-1.5	1.6
-1.6	1.8
-1.6	1.6
-0.6	0.3
0.0	0.3
-1.7	1.6
0.1	0.4





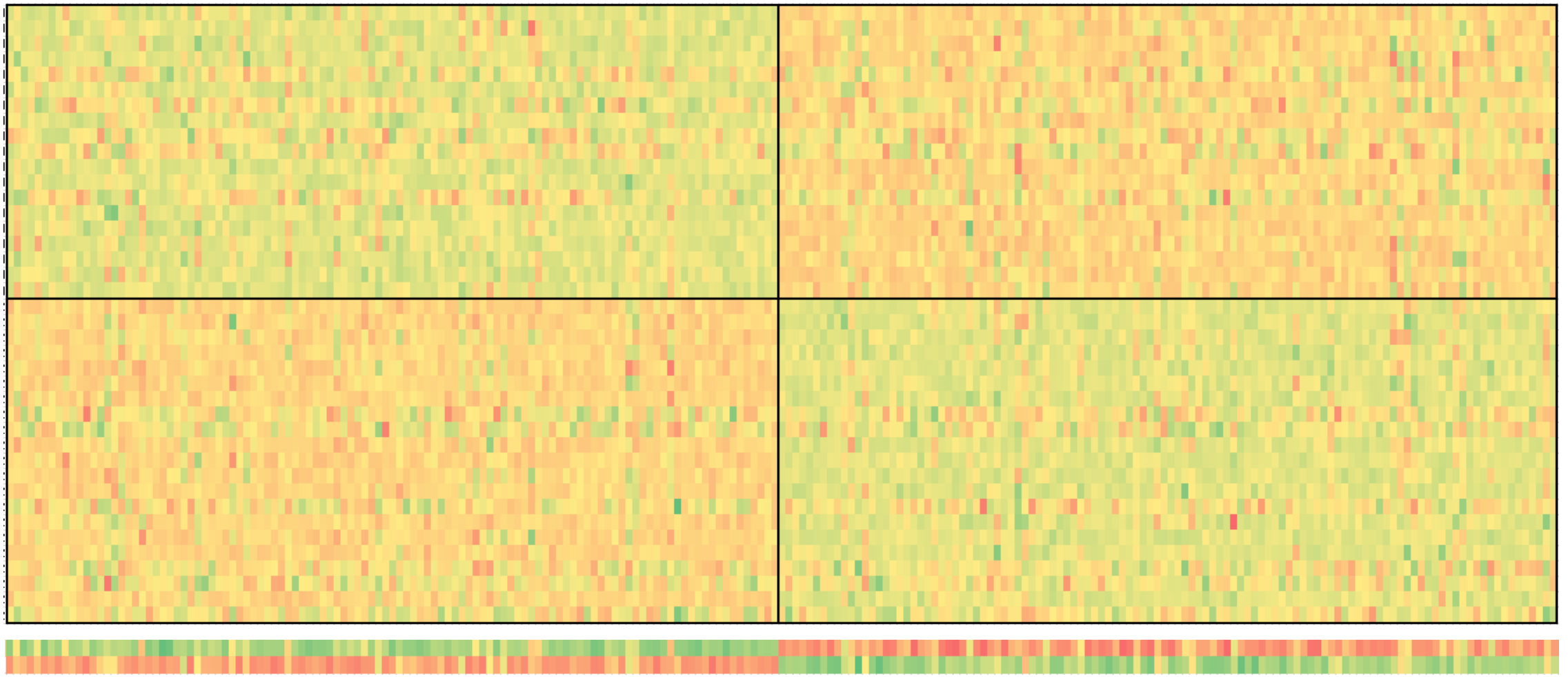


SKIM

Etcetera, etcetera, until convergence

SKIM

Etcetera, etcetera, until convergence



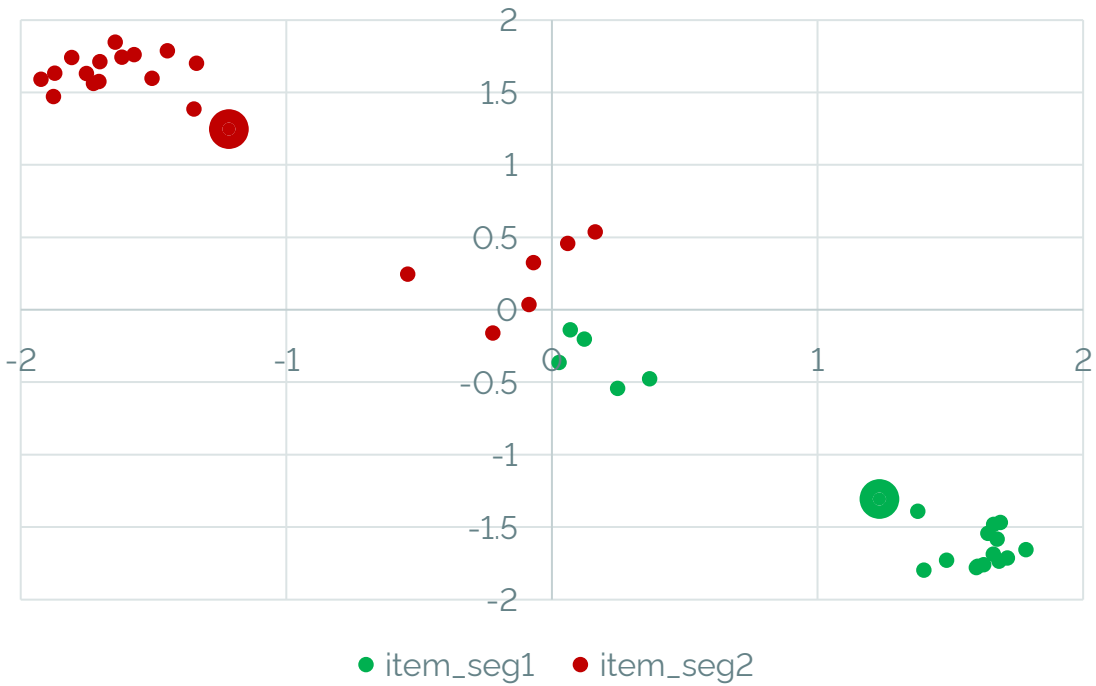
1.5	-1.7
1.5	-1.5
1.7	-1.6
1.8	-1.6
0.2	-0.5
1.7	-1.8
0.1	-0.2
1.5	-1.7
0.2	-0.3
0.0	-0.3
1.7	-1.7
1.8	-1.6
0.1	-0.2
1.7	-1.7
1.6	-1.8
1.4	-1.8
1.8	-1.7
1.6	-1.7
1.7	-1.4
-1.9	1.5
-1.5	1.6
-1.5	1.5
-1.4	1.7
-1.9	1.6
-1.6	1.5
-1.8	1.7
-0.2	0.1
0.3	0.4
-1.8	1.6
-1.8	1.6
-1.5	1.9
-1.6	1.8
-0.2	-0.2
-1.5	1.6
-1.6	1.8
-1.6	1.6
-0.6	0.3
0.0	0.3
-1.7	1.6
0.1	0.4

SKIM

Etcetera, etcetera, until convergence

Each dot is a respondent!

Each dot is an item



Average score within item segment 2

Average respondent score within item segment 1

Average score within respondent segment 2

Average item score within respondent segment

Co-clustering with covariates

Why would you?

SKIM

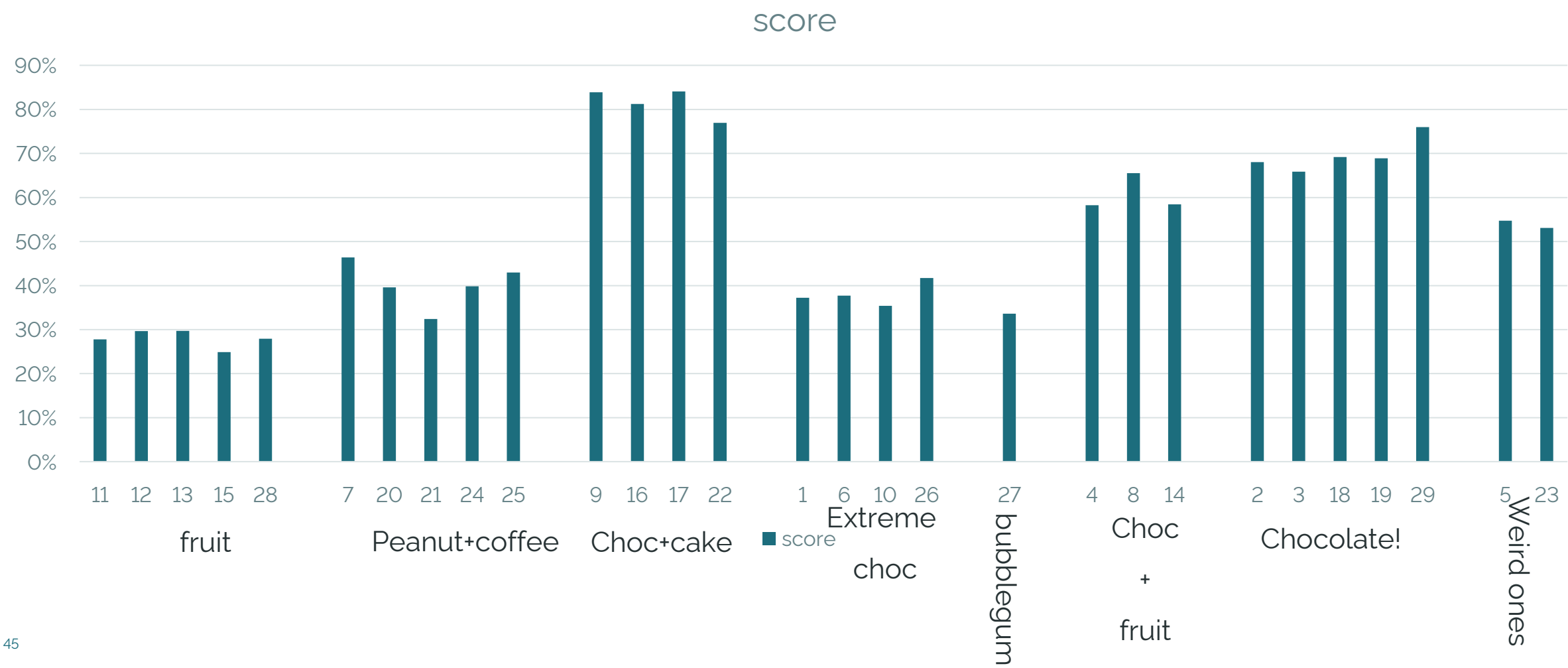
Co-clustering with covariates

Visualisation

SKIM

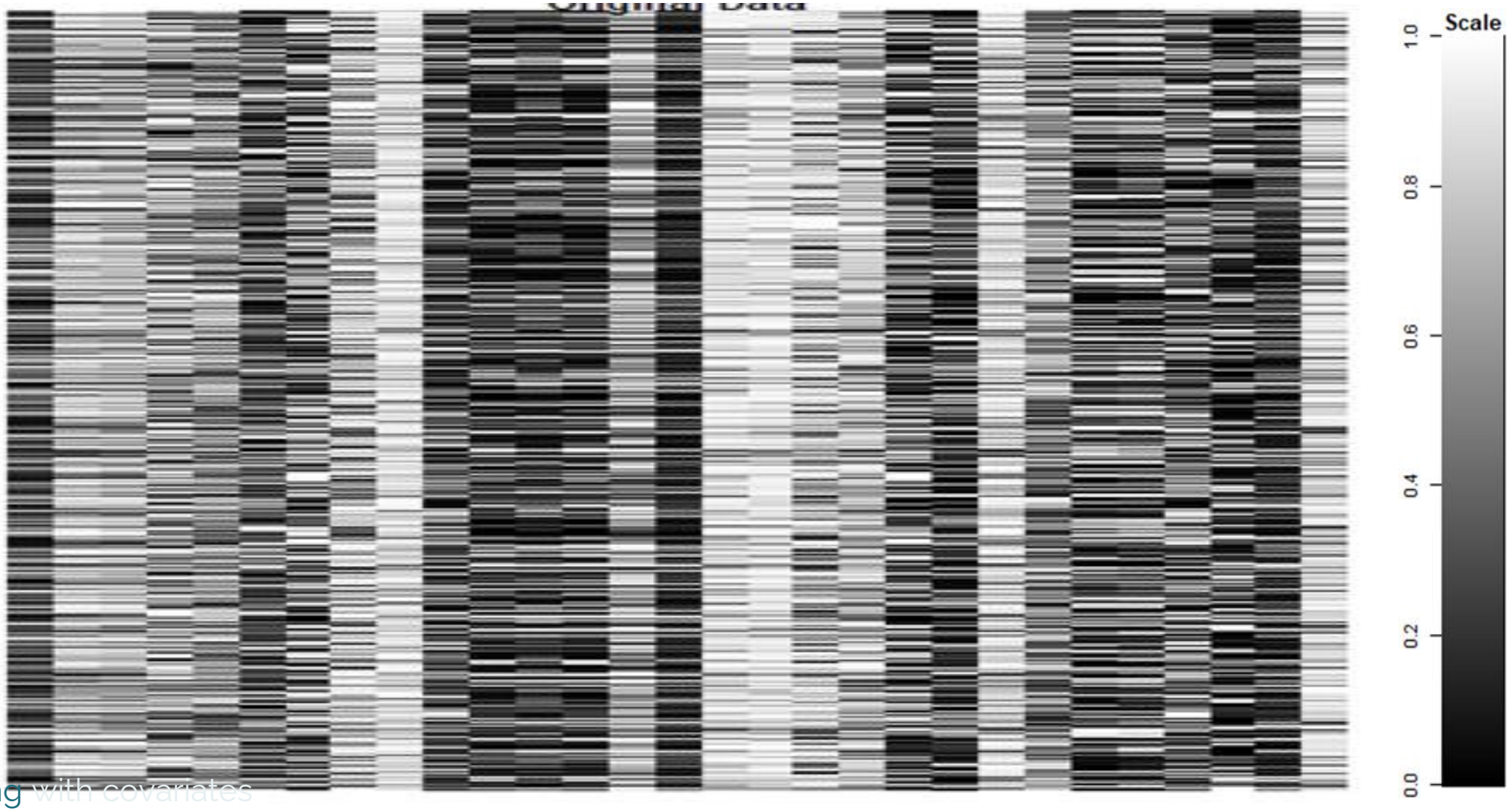
Average scores: Nice to have, but not truly informative

Regrouped scores. Nice, but still not amazingly insightful

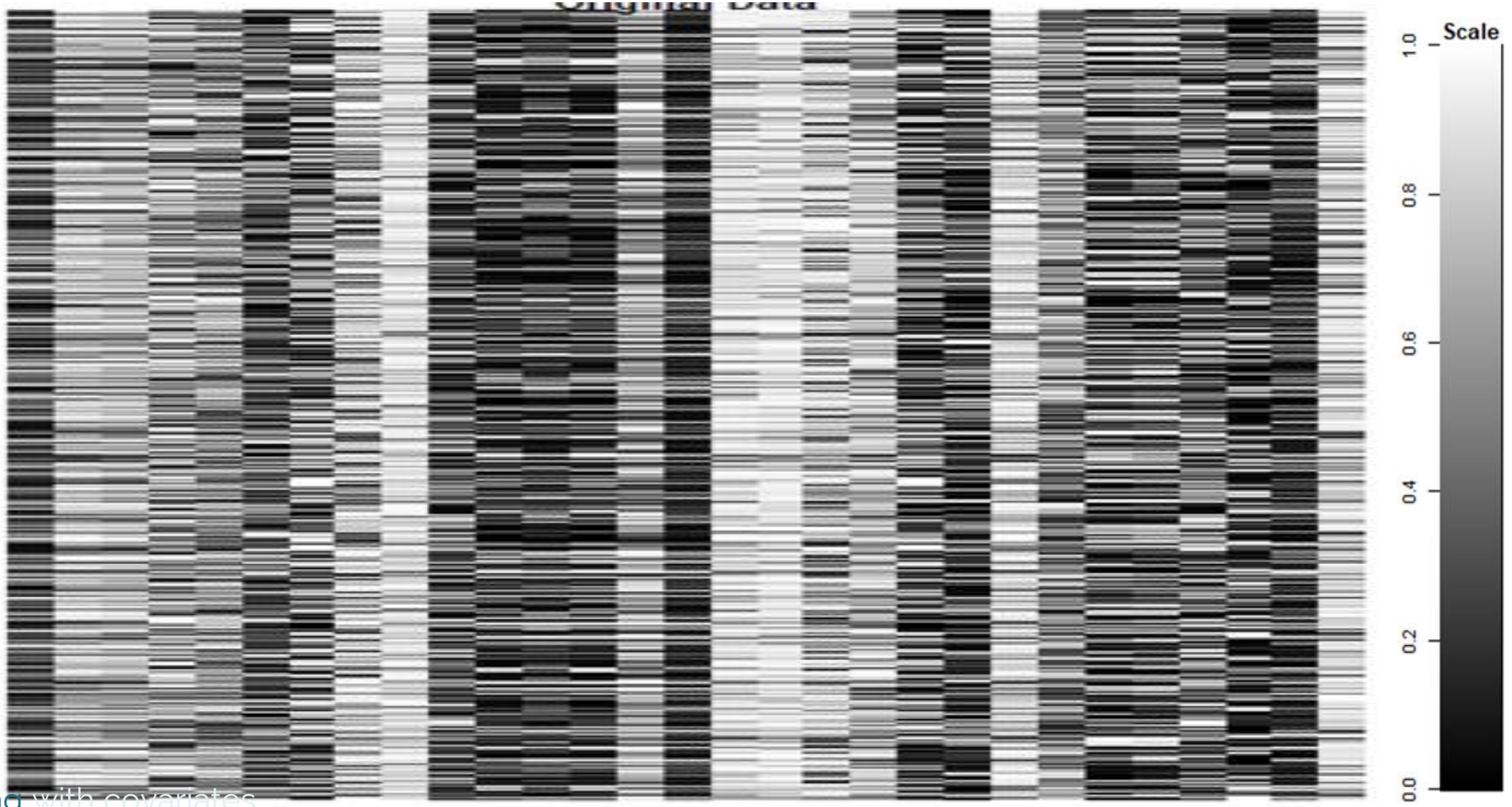


SKIM

Let's take a MaxDiff exercise. Rows are respondents, columns are items. The lighter, the higher the score is



Not really informative this way
At least to me, it is not really clear what's going on with the data



Row height indicates group size
Column width represents average score

Groups of items

Groups of respondents

	fruit	Peanut+coffee	Choc+cake	Extreme choc	Bubblegum	Choc + fruit	Chocolate!	Weird ones
	I1	I2	I3	I4	I5	I6	I7	I8
resp1	8%	45%	93%	42%	20%	54%	81%	47%
resp2	44%	3%	90%	33%	51%	77%	78%	75%
resp3	15%	67%	71%	49%	30%	55%	56%	40%
resp4	83%	10%	62%	26%	61%	79%	53%	73%
resp5	14%	14%	97%	29%	31%	63%	92%	68%
resp6	27%	33%	86%	33%	38%	63%	73%	60%
resp7	1%	64%	95%	44%	21%	41%	85%	44%
resp8	58%	46%	60%	35%	40%	66%	48%	48%
resp9	3%	75%	84%	55%	11%	48%	69%	39%

Or, sorted in order of size (and score), provides excellent understanding for the results of a TURF analysis

Groups of items

Groups of respondents

	Choc+cake	Chocolate!	Choc + fruit	Weird ones	Peanut+coffee	Extreme choc	bubblegum	fruit
	I3	I7	I6	I8	I2	I4	I5	I1
resp6	86%	73%	63%	60%	33%	33%	38%	27%
resp1	93%	81%	54%	47%	45%	42%	20%	8%
resp8	60%	48%	66%	48%	46%	35%	40%	58%
resp3	71%	56%	55%	40%	67%	49%	30%	15%
resp2	90%	78%	77%	75%	3%	33%	51%	44%
resp9	84%	69%	48%	39%	75%	55%	11%	3%
resp5	97%	92%	63%	68%	14%	29%	31%	14%
resp4	62%	53%	79%	73%	10%	26%	61%	83%
resp7	95%	85%	41%	44%	64%	44%	21%	1%

SKIM

Start with choc+cake

	Choc+cake		Choc + fruit	Weird ones	Peanut+coffee	Extreme choc	bubblegum	fruit
	I3	I7	I6	I8	I2	I4	I5	I1
resp6	86%	73%	63%	60%	33%	33%	38%	27%
resp1	93%	81%	54%	47%	45%	42%	20%	8%
resp8	60%	48%	66%	48%	46%	35%	40%	58%
resp3	71%	56%	55%	40%	67%	49%	30%	15%
resp2	90%	78%	77%	75%	3%	33%	51%	44%
resp9	84%	69%	48%	39%	75%	55%	11%	3%
resp5	97%	92%	63%	68%	14%	29%	31%	14%
resp4	62%	53%	79%	73%	10%	26%	61%	83%
resp7	95%	85%	41%	44%	64%	44%	21%	1%

SKIM

This outperforms chocolate!

	Choc+cake	Chocolate!	Choc + fruit	Weird ones	Peanut+coffee	Extreme choc	bubblegum	fruit
	I3	I7	I6	I8	I2	I4	I5	I1
resp6	86%	73%	63%	60%	33%	33%	38%	27%
resp1	93%	81%	54%	47%	45%	42%	20%	8%
resp8	60%	48%	66%	48%	46%	35%	40%	58%
resp3	71%	56%	55%	40%	67%	49%	30%	15%
resp2	90%	78%	77%	75%	3%	33%	51%	44%
resp9	84%	69%	48%	39%	75%	55%	11%	3%
resp5	97%	92%	63%	68%	14%	29%	31%	14%
resp4	62%	53%	79%	73%	10%	26%	61%	83%
resp7	95%	85%	41%	44%	64%	44%	21%	1%

SKIM

And a lot of others, so all that's left to add is choc+fruit and maybe fruit

	Choc+cake	Chocolate!	Choc + fruit	Weird ones	Peanut+coffee	Extreme choc	bubblegum	fruit
	I3	I7	I6	I8	I2	I4	I5	I1
resp6	86%	73%	63%	60%	33%	33%	38%	27%
resp1	93%	81%	54%	47%	45%	42%	20%	8%
resp8	60%	48%	66%	48%	46%	35%	40%	58%
resp3	71%	56%	55%	40%	67%	49%	30%	15%
resp2	90%	78%	77%	75%	3%	33%	51%	44%
resp9	84%	69%	48%	39%	75%	55%	11%	3%
resp5	97%	92%	63%	68%	14%	29%	31%	14%
resp4	62%	53%	79%	73%	10%	26%	61%	83%
resp7	95%	85%	41%	44%	64%	44%	21%	1%

SKIM

Co-clustering with covariates
Data imputation

SKIM

Co-clusters can be used to impute data that was missing at random

	fruit	Peanut+coffee	Choc+cake	Extreme choc	bubblegum	Choc + fruit	Chocolate!	Weird ones
	I1	I2	I3	I4	I5	I6	I7	I8
resp1	8%	45%	93%	42%	20%	54%	81%	47%
resp2	44%	3%	90%	33%	51%	77%	78%	75%
resp3	15%	67%	71%	49%	30%	55%	56%	40%
resp4	83%	10%	62%	26%	61%	79%	53%	73%
resp5	14%	14%	97%	29%	31%	63%	92%	68%
resp6	27%	33%	86%	33%	38%	63%	73%	60%
resp7	1%	64%	95%	44%	21%	41%	85%	44%
resp8	58%	46%	60%	35%	40%	66%	48%	48%
resp9	3%	75%	84%	55%	11%	48%	69%	39%

Let's look at one cell

	I1	I2	I3	I4	I5	I6	I7	I8
	fruit	Peanut+coffee	Choc+cake	Extreme choc	bubblegum	Choc + fruit	Chocolate!	Weird ones
resp1	8%	45%	93%	42%	20%	54%	81%	47%
resp2	44%	3%	90%	33%	51%	77%	78%	75%
resp3	15%	67%	71%	49%	30%	55%	56%	40%
resp4	83%	10%	62%	26%	61%	79%	53%	73%
resp5	14%	14%	97%	29%	31%	63%	92%	68%
resp6	27%	33%	86%	33%	38%	63%	73%	60%
resp7	1%	64%	95%	44%	21%	41%	85%	44%
resp8	58%	46%	60%	35%	40%	66%	48%	48%
resp9	3%	75%	84%	55%	11%	48%	69%	39%

And zoom in (i.e. see what the actual underlying data is)

		item2	item7	item9	item13	item16
resp4	A	3.3%	7.7%	.	13.3%	11.3%
resp6		16.7%	.	0.7%	.	14.3%
resp9		4.0%	8.3%	8.0%	10.7%	1.0%
resp14		13.0%	11.3%	5.3%	2.0%	11.7%
resp18		13.0%	.	10.0%	.	3.0%
resp19		2.0%	0.3%	7.0%	11.3%	16.3%
resp21		7.0%	.	1.3%	4.3%	9.3%
….						
resp 1025		10.0%	0.3%	11.3%	.	.
			B			

You can:

A) Use the entire clustercell average as

the missing data.

Or

B) There is a ton of data used to come up with the number, so use all respondents within the same group

As the imputation is based on data from other people as well, you do not run into collinearity issues when using the data in other analyses

SKIM

Co-clustering with covariates

Why would you want to do co-clustering

- Doing a simultaneous clustering of both respondents and data allows the researcher to summarise the full heterogeneity of the data in one overview
- It provides an excellent understanding of the results of a TURF analysis
- It can be used to impute data
- It runs superfast (For example, a dataset with 914 respondents, 29 items. 9 respondent segments, 8 item segments runs in 0.3 seconds on my old laptop), so why not ☺

Co-clustering with covariates

What is a covariate?

- Covariates are additional explanatory variables, such as usage, behavioral/attitudinal segments, demographics, etc. that can help predict to which segment someone (or something!) belongs

Going back a step. How do you predict the segment someone belongs to?

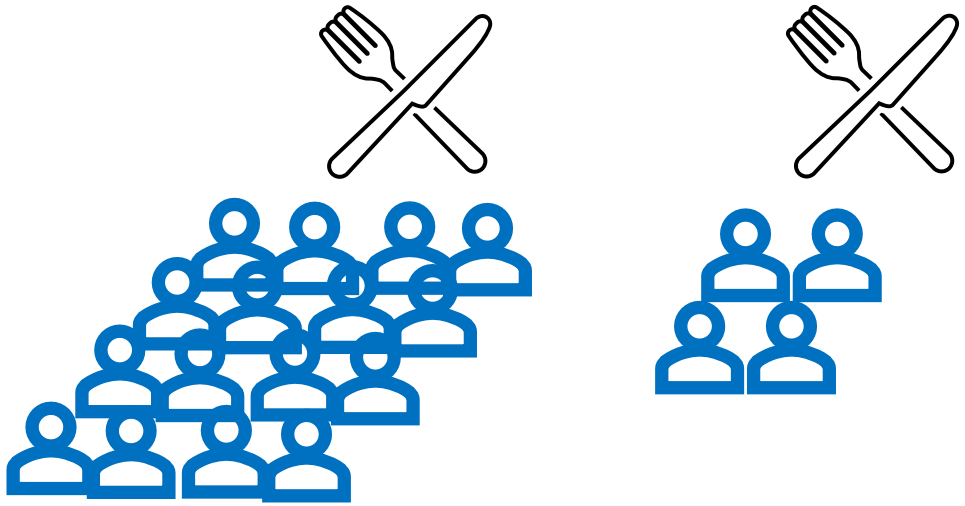
- P (segment | resp_data) ∝ P (resp_data | segment) * P (segment)
- Or in words: the probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the probability of the segment (prior probability of the segment)

Going back a step. How do you predict the segment someone belongs to?

- Or in words: the probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the prior probability of the segment
- I am hungry and I feel like eating Asian food (my data), and I need to select a restaurant (segment)

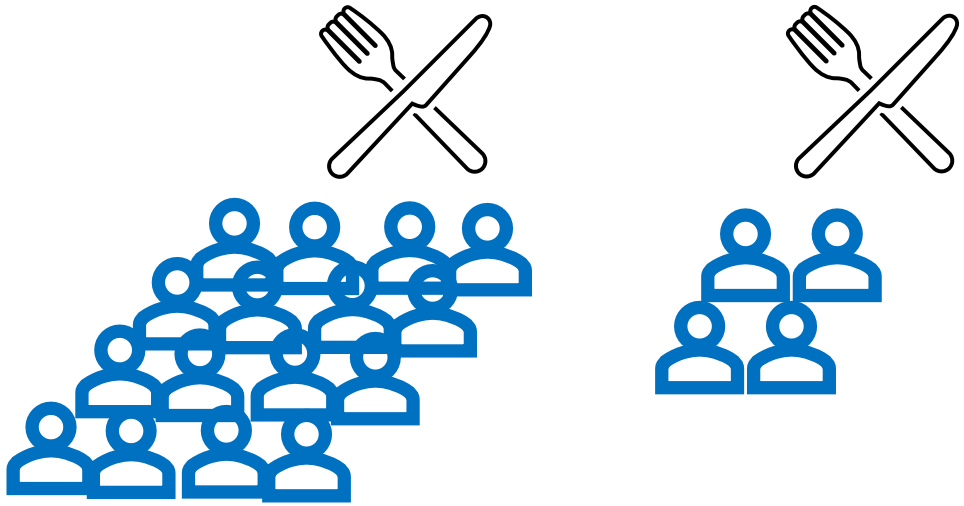
Going back a step. How do you predict the segment someone belongs to?

- Or in words: the probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the prior probability of the segment
- I am hungry and I feel like eating Asian food (my data), and I need to select a restaurant (segment)
- I walk into a street and see two restaurant. One of the restaurants is busy, almost full. The other only has a handful of people



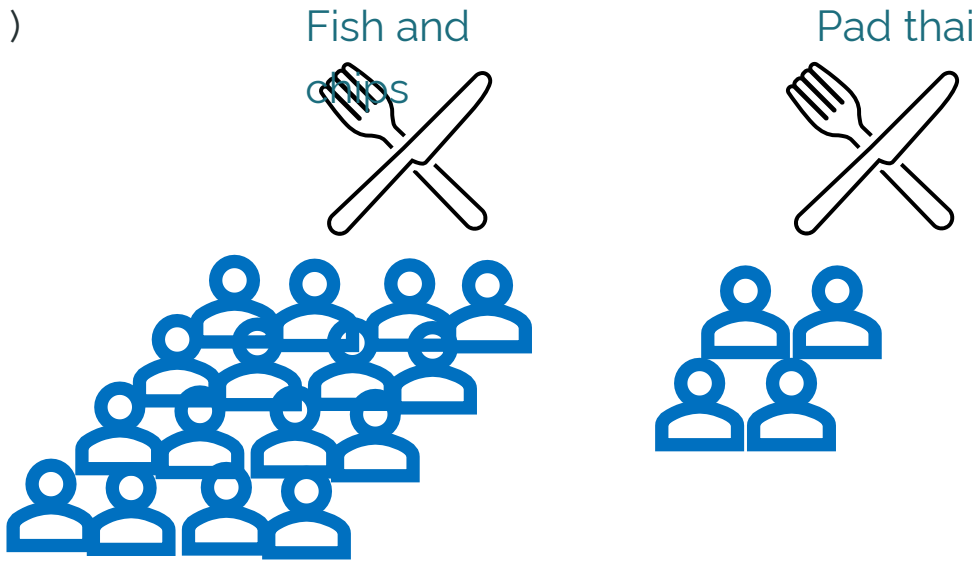
Going back a step. How do you predict the segment someone belongs to?

- The probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the prior probability of the segment
- I am hungry and I feel like eating Asian food (my data), and I need to select a restaurant (segment)
- I walk into a street and see two restaurant. One of the restaurants is busy, almost full. The other only has a handful of people
- If both restaurant would server the same type of food, I would be more inclined to select the fuller restaurant, because that's what most people do



Going back a step. How do you predict the segment someone belongs to?

- The probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the prior probability of the segment
- If both restaurant would server the same type of food, I would be more inclined to select the fuller restaurant.
- But then I look at the "today's specials" and see that the second restaurant serves Pad Thai, which much better aligns with my appetite at that moment. (much better match between my data and the segment)
- Even though the majority goes for the fish and chips restaurant (prior), I still prefer the second (my data fitting the segment)



Match between data and the segment

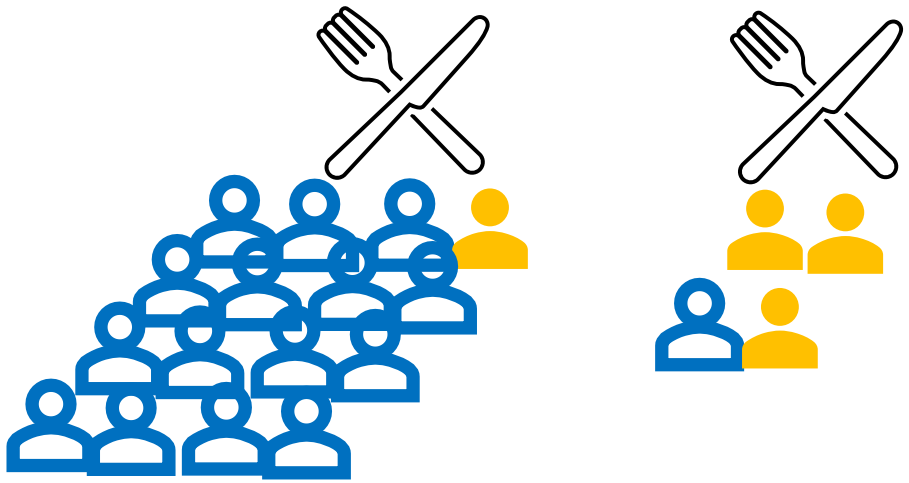
Prior

Going forward a step. How do covariates come into play?

Going forward a step. How do covariates come into play?

- ~~P (segment | resp_data) ∝ P (resp_data | segment) * P (segment)~~
- P (segment | resp_data, covariates) ∝ P (resp_data | segment) * P (segment |covariates)

- Or in words: the probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the probability of the segment, given the covariates (prior probability of the segment)

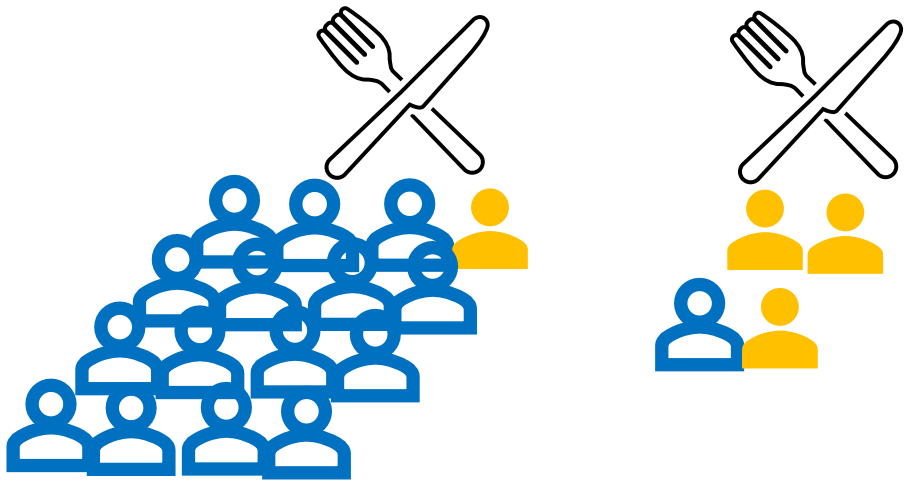


Match between data and the segment

Prior

Going forward a step. How do covariates come into play?

- The probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the probability of the segment, given the covariates (prior probability of the segment)
- I am an orange person . If both restaurant would server the same type of food, I would be more inclined to select, not the fuller restaurant, but the second one, as ¾ of orange people choose it!



Match between data and the segment

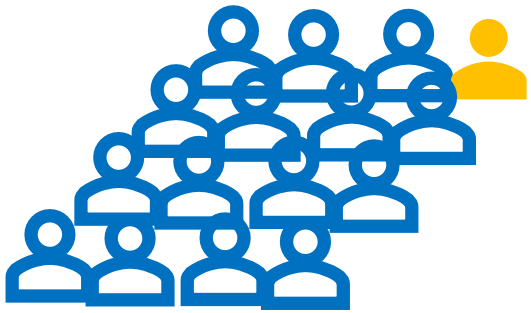
Prior

Going forward a step. How do covariates come into play?

- P (segment | resp_data, covariates) ∝ P (resp_data | segment) * P (segment |covariates)
- The prior segment probability is now a function of the covariates.
- Use good old multinomial logit to predict the segment using covariates as independent variables
- Regress the segment membership from the previous iteration on the covariates)
  - $u=\beta x+\varepsilon$
  - $y=$ exp(u)/sum(exp(u))
  - In this case, x is the covariate data
  - And y is the predicted probability coming from P (segment | resp_data, covariates) in the previous iteration!

Prior

Coding of segment prediction single csv style



15x

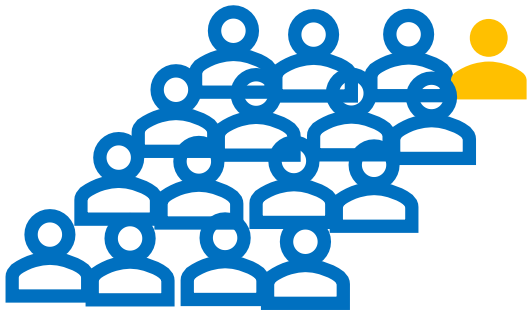
		attributes	
		blue_or_orange_seg	blue_or_orange_se
concept	segment	1	2
1	1	**1**	0
2	2	0	**1**

1x

		blue_or_orange_seg	blue_or_orange_se
concept	segment	1	2
1	1	**2**	0
2	2	0	**2**

SKIM

Coding of segment prediction single csv style



15x

concept	segment	blue_or_orange_seg 1	blue_or_orange_seg 2	"choice"
1	1	**1**	0	100%
2	2	0	**1**	0%

1x

concept	segment	blue_or_orange_seg 1	blue_or_orange_seg 2	"choice"
1	1	**2**	0	100%
2	2	0	**2**	0%

1x

concept	segment	blue_or_orange_seg 1	blue_or_orange_seg 2	"choice"
1	1	**1**	0	0%
2	2	0	**1**	100%

3x

concept	segment	blue_or_orange_seg 1	blue_or_orange_seg 2	"choice"
1	1	**2**	0	0%
2	2	0	**2**	100%

SKIM

Respondents are not the only ones that can have covariates

PG

animation comedy ~~SciFi~~

PG

animation comedy ~~SciFi~~

PG-13

~~animation~~ ~~comedy~~ SciFi

PG-13

~~animation~~ ~~comedy~~ SciFi

SKIM

Now we know how to include covariates. Now we still need to turn distance into a probability

- P (segment | resp_data, covariates) ∝ P (resp_data | segment) * P (segment |covariates)
- The probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the probability of the segment, given the covariates (prior probability of the segment)
- Even for distance, there are many ways to calculate it: Euclidian or Manhattan Block

$$d(\mathbf{p},\mathbf{q}) = \sqrt{(p_1-q_1)^2 + (p_2-q_2)^2 + \cdots + (p_n-q_n)^2}$$

$$d_1(\mathbf{p},\mathbf{q}) = \|\mathbf{p}-\mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

- The easiest way to turn distance into a probability is by first assuming a distribution and then use the probability density function (which is just a fancy word for distance). You can make it as complicated as you want!
- I like

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Now we have it! Co-clustering with covariates

- P (segment | resp_data, covariates) ∝ P (resp_data | segment) * P (segment |covariates)
- The probability someone belonging to a specific segment, given their data, is proportional to how well their data fits the segment multiplied by the probability of the segment, given the covariates (prior probability of the segment)

- The probability someone/something belongs to a specific segment, given their data is a function of:
  - The distance of the data (probability density function(s)) to the segment mean

  and

  - Predicted segment probability based on their covariate(s)

Co-clustering with covariates

Why would you want to have covariates

- Remember that you can also have covariates on the data, not (just) on the respondents. In my opinion this a lot more valuable!
- For example:
  - With ice-cream flavours, you can code the ingredients as covariates (chocolate, fruit, cake, etc.)
  - With claims, you can code the benefit area (e.g. environment, effectiveness, price, etc.)
  - But even better, you can ask respondents what they how items satisfy underlying higher needs. Chocolate may be associated with indulgence, bubble gum with fun, etc. etc.
- This would provide an explanation on why items are scored similarly by respondents and could be used for marketing materials by the client as well!
- When you have information on consumers for whom you only know covariates, you will still be able to predict what their ratings/ scores would be, as you can calculate respondent segment probabilities.
- Similarly, when you have information about items (in terms of covariates), you would be able to predict ratings or scores for untested items!

Imagine you want to know the score of an item you forgot to include
and it's a fruity one

- The probability someone/something belongs to a specific segment, given their data is a function of:
  - The distance of the data (probability density function(s)) to the segment mean ← **We do not know this**

  and
  - Predicted segment probability based on their covariate(s) ← **But we do know this**

SKIM

Imagine you want to know the score of an item you forgot to include
and it's a fruity one

- Using the MNL from with the covariates as input, you can calculate probabilities for each of the item groups
- This is better than "well… I think it should be in I1"

	fruit	Peanut+coffee	Choc+cake	Extreme choc	bubblegum	Choc + fruit	Chocolate!	Weird ones
	I1	I2	I3	I4	I5	I6	I7	I8
resp1	8%	45%	93%	42%	20%	54%	81%	47%
resp2	44%	3%	90%	33%	51%	77%	78%	75%
resp3	15%	67%	71%	49%	30%	55%	56%	40%
resp4	83%	10%	62%	26%	61%	79%	53%	73%
resp5	14%	14%	97%	29%	31%	63%	92%	68%
resp6	27%	33%	86%	33%	38%	63%	73%	60%
resp7	1%	64%	95%	44%	21%	41%	85%	44%
resp8	58%	46%	60%	35%	40%	66%	48%	48%
resp9	3%	75%	84%	55%	11%	48%	69%	39%
Item group probability	80%	0%	0%	0%	5%	10%	0%	5%

Imagine you want to know the score of an item you forgot to include
and it's a fruity one

- And calculate the expected score within each of the respondent groups
- For each of the respondent groups, you can check their underlying demographics to have marketing do it's magic

	fruit	Peanut+coffee	Choc+cake	Extreme choc	bubblegum	Choc + fruit	Chocolate!	Weird ones
	I1	I2	I3	I4	I5	I6	I7	I8
resp1	8%	45%	93%	42%	20%	54%	81%	47%
resp2	44%	3%	90%	33%	51%	77%	78%	75%
resp3	15%	67%	71%	49%	30%	55%	56%	40%
resp4	83%	10%	62%	26%	61%	79%	53%	73%
resp5	14%	14%	97%	29%	31%	63%	92%	68%
resp6	27%	33%	86%	33%	38%	63%	73%	60%
resp7	1%	64%	95%	44%	21%	41%	85%	44%
resp8	58%	46%	60%	35%	40%	66%	48%	48%
resp9	3%	75%	84%	55%	11%	48%	69%	39%
Item group probability	80%	0%	0%	0%	5%	10%	0%	5%

Summary

SKIM

Summary

- Co-clustering
  - summarises the full heterogeneity of the data in one overview
  - provides input for a simple visual overview of the data
  - can be used to impute data

- When you add covariates
  - you add an additional layer of insights
  - you add more data imputation possibilities

Thank you

Kees van der Wagt

Senior Director

k.vanderwagt@skimgroup.com

Appendix

Advanced:
Ensembling item grouping

SKIM

The algorithm is super fast

- Most steps involve matrix multiplication, which runs super fast

- For example, a dataset with 914 respondents, 29 items. 9 respondent segments, 8 item segments runs in 0.3 seconds
- If I go to 90 respondent segments, 2.07seconds

- With k-means, different starting points lead to different results

- Running multiple times is still fast, so why not run it more often and use this information

In pseudo code

- Run the clustering algorithm multiple times
- Save the item segment probabilities
- Use aggregated grouping as starting point for the “final” run

SKIM

Example

- Let's say we still 4 items, with 2 item segments
- In the first run, we have

	item_seg1	item_seg2
item1	100%	0%
item2	100%	0%
item3	0%	100%
item4	0%	100%

- In the second, we have

	item_seg1	item_seg2
item1	0%	100%
item2	0%	100%
item3	100%	0%
item4	100%	0%

- These solutions are identical, it is just a relabelling
- Therefore you must store which items were in the same group!

SKIM

Example - store which items were in the same group!

This is easiest done with (you guessed it!) a matrix multiplication

	item_seg1	item_seg2
item1	0%	100%
item2	0%	100%
item3	100%	0%
item4	100%	0%

*

	item1	item2	item3	item4
item_seg1	0%	0%	100%	100%
item_seg2	100%	100%	0%	0%

=

	item1	item2	item3	item4
item1	1	1	0	0
item2	1	1	0	0
item3	0	0	1	1
item4	0	0	1	1

Identical!

	item_seg1	item_seg2
item1	100%	0%
item2	100%	0%
item3	0%	100%
item4	0%	100%

*

	item1	item2	item3	item4
item_seg1	100%	100%	0%	0%
item_seg2	0%	0%	100%	100%

=

	item1	item2	item3	item4
item1	1	1	0	0
item2	1	1	0	0
item3	0	0	1	1
item4	0	0	1	1

*sidenote: The probabilities are not always 0/1, so I take the square root of the result

SKIM

Update this matrix every time by adding the results from the run

	item1	item2	item3	item4
item1	1	1	0	0
item2	1	1	0	0
item3	0	0	1	1
item4	0	0	1	1

	item1	item2	item3	item4
item1	2	2	0	0
item2	2	2	0	0
item3	0	0	2	2
item4	0	0	2	2

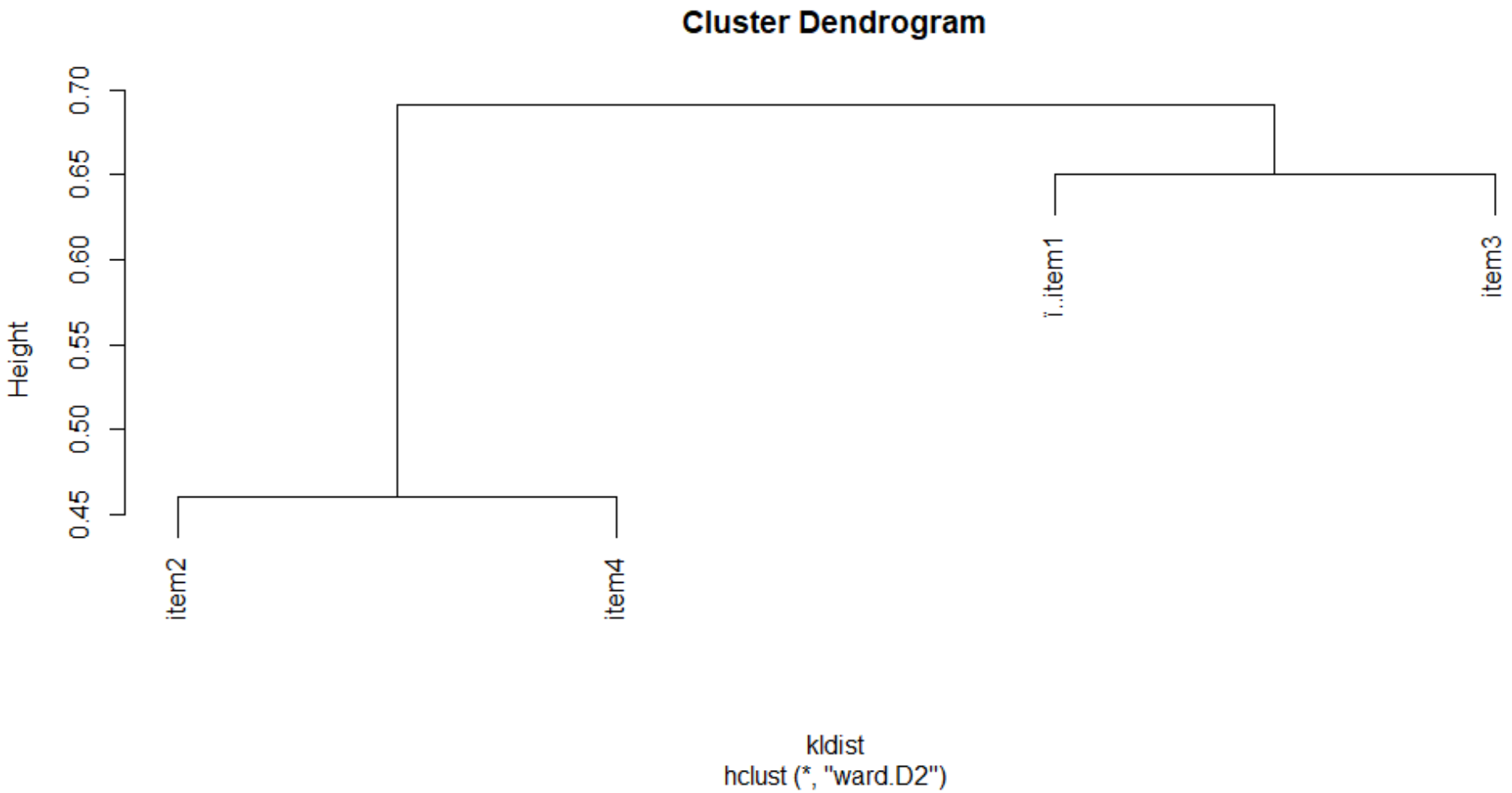
	item1	item2	item3	item4
item1	3	2	0	1
item2	2	3	1	0
item3	0	1	3	2
item4	1	0	2	3

.....

	item1	item2	item3	item4
item1	101	36	36	36
item2	36	101	35	55
item3	36	35	101	41
item4	36	55	41	101

SKIM

Run a hierarchical clustering on the (dis)similarity matrix for the base grouping

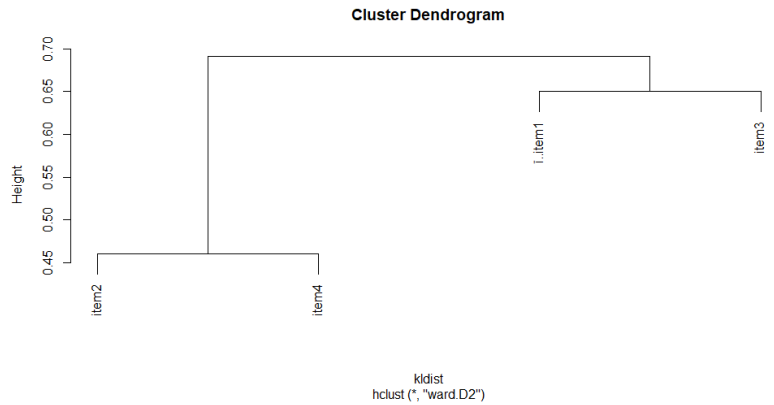


SKIM

Run a hierarchical clustering on the (dis)similarity matrix for the base grouping

- So we have the base structure that has
- Items 2 and 4 together in group 1
- Items 1 and 3 together in group 2

- Or [1 2 1 2]



SKIM

Run a hierarchical clustering on the (dis)similarity matrix for the base grouping

- So we have the base structure that has
- Items 2 and 4 together in group 1
- Items 1 and 3 together in group 2

- Or [1 2 1 2]

	item1	item2	item3	item4
item1	101	36	36	36
item2	36	101	35	55
item3	36	35	101	41
item4	36	55	41	101

	seg2	seg1	seg2	seg1
item1	101	36	36	36
item2	36	101	35	55
item3	36	35	101	41
item4	36	55	41	101

SKIM

And convert to probabilities

	seg2	seg1	seg2	seg1
item1	101	36	36	36
item2	36	101	35	55
item3	36	35	101	41
item4	36	55	41	101

	seg1	seg2
item1	72	137
item2	156	71
item3	76	137
item4	156	77

	seg1	seg2
item1	34%	66%
item2	69%	31%
item3	36%	64%
item4	67%	33%

SKIM

And use these probabilities as starting seed for the final run

	seg1	seg2
item1	34%	66%
item2	69%	31%
item3	36%	64%
item4	67%	33%

SKIM

You could also do this for respondent grouping

- But this will be a lot slower

- For me, the main focus is on the item grouping

SKIM

R-code

SKIM

Ugly uncommented R-code

```
itemseg<-matrix(runif(ncol(my_data)*N_item_Seg),ncol=N_item_Seg)
itemseg<-itemseg/rowSums(itemseg)
for (iter in 1:N_iterations){
  itemseg<-t(itemseg)
  itemseg<-t(itemseg/rowSums(itemseg))
  resp_itemsegscore<-(my_data %*% itemseg)
  respseg<-t(respseg)
  respseg<-respseg/rowSums(respseg)
  respseg_itemseg_score<-respseg%*%resp_itemsegscore
  dist<-c()
  for (i in 1:N_resp_Seg){
    afstand<-resp_itemsegscore-
matrix(respseg_itemseg_score[i,],nrow=nrow(resp_itemsegscore),ncol=N_item_Seg,byrow=
```