RESEARCHSOFTWARE
A DIVISION OF DISPLAYR

TIM BOCK PRESENTS

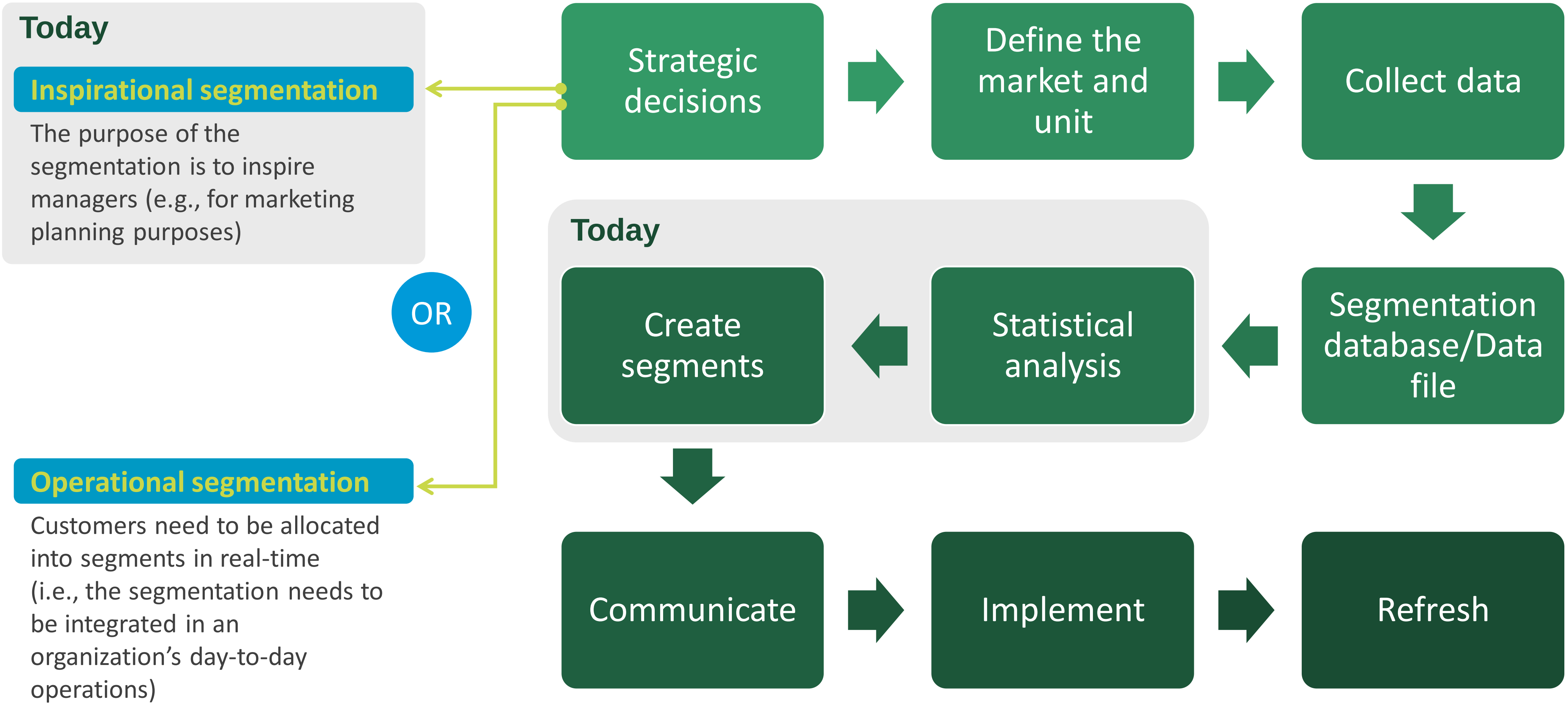
DIY Advanced Analysis

Session 4: Segmentation

Software comparison

- Q was designed for DIY segmentation
- Displayr has some of Q's tools
- R is poor for segmentation. To the best of my knowledge, there is no R package that can deal with all of the following:
  - Missing values
  - Weights
  - Ranking data
  - Conjoint/Choice modeling data
  - Max-diff
  - Multiple separate data types
- Some useful R packages
  - Displayr/flipCluster::Kmeans – kMeans with weights and missing data
  - poLCA – Latent class of categorical variables
  - mclust – Latent class of numeric data
  - flexmix – General-purpose latent class tool, requires programming and statistical skills to use

Phases in the segmentation process



Today

Inspirational segmentation

The purpose of the segmentation is to inspire managers (e.g., for marketing planning purposes)

OR

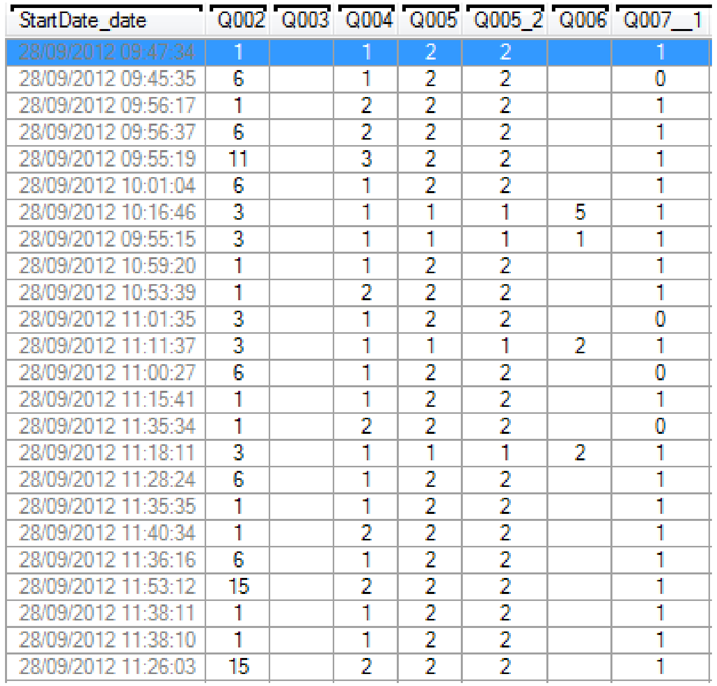
Operational segmentation

Customers need to be allocated into segments in real-time (i.e., the segmentation needs to be integrated in an organization’s day-to-day operations)

Strategic decisions → Define the market and unit → Collect data → Segmentation database/Data file → Statistical analysis → Create segments → Communicate → Implement → Refresh

Today: Statistical analysis, Create segments

The goal: turn raw data into segments



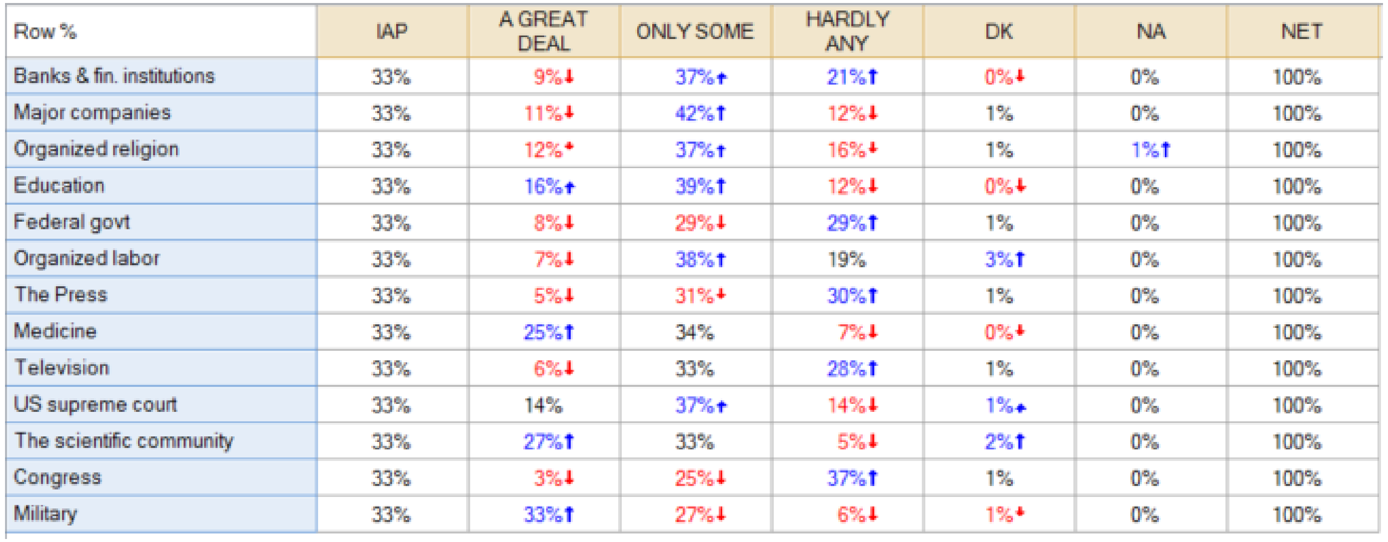
StartDate_date	Q002	Q003	Q004	Q005	Q005_2	Q006	Q007__1
28/09/2012 09:47:34	1		1	2	2		1
28/09/2012 09:45:35	6		1	2	2		0
28/09/2012 09:56:17	1		2	2	2		1
28/09/2012 09:56:37	6		2	2	2		1
28/09/2012 09:55:19	11		3	2	2		1
28/09/2012 10:01:04	6		1	2	2		1
28/09/2012 10:16:46	3		1	1	1	5	1
28/09/2012 09:55:15	3		1	1	1	1	1
28/09/2012 10:59:20	1		1	2	2		1
28/09/2012 10:53:39	1		2	2	2		1
28/09/2012 11:01:35	3		1	2	2		0
28/09/2012 11:11:37	3		1	1	1	2	1
28/09/2012 11:00:27	6		1	2	2		0
28/09/2012 11:15:41	1		1	2	2		1
28/09/2012 11:35:34	1		2	2	2		0
28/09/2012 11:18:11	3		1	1	1	2	1
28/09/2012 11:28:24	6		1	2	2		1
28/09/2012 11:35:35	1		1	2	2		1
28/09/2012 11:40:34	1		2	2	2		1
28/09/2012 11:36:16	6		1	2	2		1
28/09/2012 11:53:12	15		2	2	2		1
28/09/2012 11:38:11	1		1	2	2		1
28/09/2012 11:38:10	1		1	2	2		1
28/09/2012 11:26:03	15		2	2	2		1

		Segment Names (Toothpaste market)			
		Worrier	**Sociable**	**Sensory**	**Independent**
Who	*Demographics*	Large families 25-40	Teens Young smokers	Children	Males 35-50
	Psychographics	Conservative Hypochondriacs	Highly social Active	Self-involved Pleasure-seeking	Self-sufficient
What	*Main brand*	Crest	MacLeans Ultrabrite	Colgate	Supermarket brand
	Pack size preference	Large dispensers	Large	Medium	Small
	Price paid	Low	High	Medium	Low
Where	*Channel*	Supermarket	Supermarket	Supermarket	Neighborhood shops
Why	*Benefits sought*	Stop decay	Attract attention	Flavour	Functionality
Segment size		50%	30%	15%	5%
Potential for growth		Low	High	Medium	Low

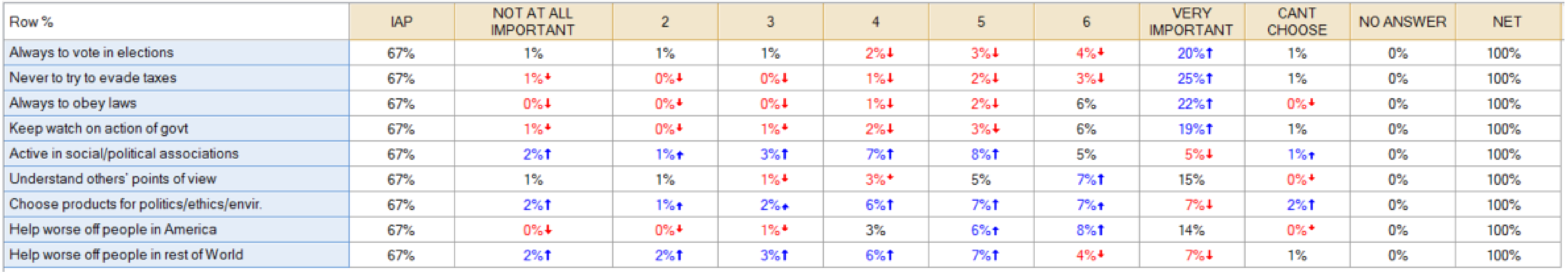
Adapted from: Haley, R. I. (1968). "Benefit Segmentation: A Decision Oriented Research Tool." Journal of Marketing 30(July): 30-35.

Case Study 1: General Social Survey

- US data
- 2014
- 3842 cases and 380 variables
- NORC at the University of Chicago
- Download the data set used here from http://wiki.q-researchsoftware.com/wiki/DIY_Advanced_Analysis
- Get the original data set (messy) from http://gss.norc.org/Get-The-Data



Row %	IAP	A GREAT DEAL	ONLY SOME	HARDLY ANY	DK	NA	NET
Banks & fin. institutions	33%	9%↓	37%↑	21%↑	0%↓	0%	100%
Major companies	33%	11%↓	42%↑	12%↓	1%	0%	100%
Organized religion	33%	12%↓	37%↑	16%↓	1%	1%↑	100%
Education	33%	16%↑	39%↑	12%↓	0%↓	0%	100%
Federal govt	33%	8%↓	29%↓	29%↑	1%	0%	100%
Organized labor	33%	7%↓	38%↑	19%	3%↑	0%	100%
The Press	33%	5%↓	31%↓	30%↑	1%	0%	100%
Medicine	33%	25%↑	34%	7%↓	0%↓	0%	100%
Television	33%	6%↓	33%	28%↑	1%	0%	100%
US supreme court	33%	14%	37%↑	14%↓	1%↑	0%	100%
The scientific community	33%	27%↑	33%	5%↓	2%↑	0%	100%
Congress	33%	3%↓	25%↓	37%↑	1%	0%	100%
Military	33%	33%↑	27%↓	6%↓	1%↓	0%	100%



Row %	IAP	NOT AT ALL IMPORTANT	2	3	4	5	6	VERY IMPORTANT	CANT CHOOSE	NO ANSWER	NET
Always to vote in elections	67%	1%	1%	1%	2%↓	3%↓	4%↓	20%↑	1%	0%	100%
Never to try to evade taxes	67%	1%↓	0%↓	0%↓	1%↓	2%↓	3%↓	25%↑	1%	0%	100%
Always to obey laws	67%	0%↓	0%↓	0%↓	1%↓	2%↓	6%	22%↑	0%↓	0%	100%
Keep watch on action of govt	67%	1%↓	0%↓	1%↓	2%↓	3%↓	6%	19%↑	1%	0%	100%
Active in social/political associations	67%	2%↑	1%↑	3%↑	7%↑	8%↑	5%	5%↓	1%↑	0%	100%
Understand others' points of view	67%	1%	1%	1%↓	3%↓	5%	7%↑	15%	0%↓	0%	100%
Choose products for politics/ethics/envir.	67%	2%↑	1%↑	2%↑	6%↑	7%↑	7%↑	7%↓	2%↑	0%	100%
Help worse off people in America	67%	0%↓	0%↓	1%↓	3%	6%↑	8%↑	14%	0%↓	0%	100%
Help worse off people in rest of World	67%	2%↑	2%↑	3%↑	6%↑	7%↑	4%↓	7%↓	1%	0%	100%

Overview of issues

Data preparation issues

1. Don’t knows and non-responses
2. Ordinal variables
3. Nominal variables
4. Nominal variables with >3 categories
5. Using max-diff/conjoint/choice data
6. Questions with different ranges
7. Weights
8. No observations have complete data
9. Missing data is not MCAR
10. Missing data is non-ignorable

Issues addressed while forming segments

11. Yeah-saying bias in ratings
12. Scale effect in max-diff/conjoint/choice data
13. Working out the best number of segments
14. Algorithm has not converged
15. Local optima
16. The segmentation is dominated by a small number of variables
17. Measuring replicability
18. The segments are not replicable
19. Finding the most managerially-useful segments
20. Increasing the predictability of the segments

Issue 1: Don’t knows and non-responses

Issue

The basic logic of cluster analysis and latent class analysis is inconsistent with the whole concept of a “don’t know” and non-response.

Options (ranked from best to worst)	**Comments**
Set them as missing values	Only a good idea if the data is MAR or MCAR (discussed later)
Merge all small categories (less than c25%) and use latent class analysis with *multinomial distribution* (**Pick One** or **Pick One - Multi** in Q)	Only available in latent class analysis
Merge all small categories (less than about 25%) and convert to binary variables (aka indicator variables aka one hot encoding)	Do this if using cluster analysis & there are two or three other categories (this can be achieved by merging categories)

Issue 2: Ordinal Variables

Issue

Most cluster analysis and latent class analysis algorithms are not designed to deal with ordinal variables (i.e., variables with ordered categories, such as Unimportant, Somewhat Important, Important)

Options (ranked from best to worst)	**Comments**
Make the data *numeric*, making sure the values are appropriate	Change the data to numeric, making sure the values are appropriate. In Q: Change the **Question Type** to **Number** or **Number- Multi** and check the *value attributes*
Use algorithms specifically designed for ordinal variables	Modelling them as ordinal rarely improves the quality of the analysis, and often leads to a worse outcome as the algorithms are slower so less validation gets performed.
Merge into two categories, and then treat as numeric or multiple response	In Q: Change the **Question Type** to **Pick Any**
Use HOMALs or Multiple Correspondence Analysis to convert them to create new numeric components	This can be dangerous, as these techniques throw out a lot of data and reweight the data, so can get vastly inferior results

Issue 3: Nominal Variables

Issue

Most cluster analysis and latent class analysis algorithms are not designed to deal with *nominal* variables (i.e., variables with unordered categories, such as brand preference; in Q jargon: **Pick One** and **Pick One - Multi** questions).

Options (ranked from best to worst)	**Comments**
Use algorithms specifically designed for nominal variables	In Q & Displayr: **Latent Class Analysis** and **Mixed-Mode Cluster Analysis** In R**: poLCA**
Merge into two categories, and then treat as numeric or multiple response	In Q: Change the **Question Type** to **Pick Any**
Convert to *binary variables* (aka *indicator variables* aka *one hot encoding*)	This is the standard solution when using cluster analysis
Use HOMALs or Multiple Correspondence Analysis to convert them into numeric variables	This can be dangerous, as these techniques throw out a lot of data *and* reweight the variables, so can get vastly inferior results

Issue 4: Nominal variables have more than 3 categories

Issue

The fewer people that select category, the less influential it is in the segmentation. The consequence of this is that where there are lots of small categories in nominal variables, the resulting segments can often be counter-intuitive (e.g., segments containing people that gave ratings of Not Important and Very Important). A second problem is that it can be painful to interpret the segmentations, as there is too much data to look at.

Options (ranked from best to worst)	**Comments**
Merge similar categories	In Q: If you have a **Pick One - Multi** question, it is a good idea to first duplicate it and split it into multiple **Pick One** question (In the **Variables and Questions tab**, right-click and select: **Split Variables from Question**)
Ignore the problem	If the problem exists, it will become clear when you try and interpret the data.

Issue 5: Using max-diff/conjoint/choice data

	Options (ranked from best to worst)	**Comments**
Issue Most cluster analysis and latent class algorithms are not designed to deal with max-diff, choice, and conjoint data.	Use an algorithm specifically designed for this type of data	In Q and Displayr: **Latent Class Analysis** and **Mixed- Mode Cluster Analysis**
	Compute individual-level parameters/coefficients/scores and use them in cluster analysis or latent class analysis	The methods for producing the parameters/coefficients/scores produce what can be characterised as rough guesses, so using this data in segmentation means that your segments may be driven by these rough guesses

Issue 6: Questions with different ranges

Issue

Cluster analysis and most latent class analysis methods take differences in scale into account. E.g., if you have a 10-point scale and a 3-point scale, the likelihood is that the segments will differ primarily in terms of the data with the 10-point scale.

Options (ranked from best to worst)	**Comments**
Use algorithms that automatically allow for different questions having different ranges.	Q's **Latent Class Analysis** and **Mixed-Mode Cluster Analysis** automatically correct for differences between questions (but not within a question)
Scale the variables to have a constant range (e.g., of 1)	
Scale the variables to have a constant standard deviation (e.g., of 1)	
Use PCA, factor analysis, HOMALs or Multiple Correspondence Analysis to convert them to create new numeric components	This can be dangerous, as these techniques throw out a lot of data and implicitly focus the analysis on variables that are moderately correlated with other variables (highly correlated variables are greatly reduced in importance, and uncorrelated variables end up being excluded entirely)

Issue 7: Weights

Issue	**Options** (ranked from best to worst)	**Comments**
Issue Many cluster analysis and latent class analysis algorithms ignore weights.	Use algorithms specifically designed for weights	In Q: **Latent Class Analysis, Mixed Mode Cluster Analysis,** In Q and R and DIsplayr: **K-Means (batch)**
	Bootstrap: create a new sample by randomly sampling with replacement in proportion to the weights	Difficult to explain to clients, who struggle with the whole “a random sample of a random sample is a random sample” concept + adds “noise” to the data

Issue 8: No or few observations have complete data

Issue

Most cluster analysis methods only form segments using observations with no missing data (some then allocate observations with partial data to the segments)

Options (ranked from best to worst)	**Comments**
Use cluster analysis methods or latent class methods that address missing values	In Q and Displayr: **Latent Class Analysis, Mixed Mode Cluster Analysis** In Q and R and Displayr: **K-Means (batch)**
Impute missing values	In Q: **Automate > Browse Online Library > Missing Data > Impute** This is dangerous, as the imputed values are guesses, and the segmentation can be driven by these guesses.
Perform the analysis based only on complete observations	This is often dangerous (see the next slide). Where no observations have complete data, most cluster analysis algorithms will return an error.

Issue 9: Missing data is not MCAR

Issue

Most cluster analysis algorithms assume that the data is ***Missing Completely At Random*** (MCAR; i.e., other than that some variables have more missing values than others, there is no pattern of any kind in the missing data). This can be tested using *Little's MCAR test.*

Options (ranked from best to worst)	**Comments**
Use cluster analysis and latent class analysis methods that make the *missing at random (MAR)* assumption, rather than the *MCAR* assumption.	In Q and Displayr: **Latent Class Analysis, Mixed Mode Cluster Analysis** In Q, R and Displayr: **K-Means (batch)**
Impute missing values	In Q: **Automate > Browse Online Library > Missing Data > Impute.** This method is inferior because, when done properly, imputation adds some random noise data, and this will add random data to your results

Issue 10: Missing data is non-ignorable

Issue

Missing data is *non-ignorable* when people with missing data are fundamentally different to those without missing data for one or more variables. Example 1: we only asked a question to men. Example 2: people have not provided ratings on a product because they are not familiar with the product.

A deep understanding of how the data was collected is central to working out if there is an issue. Plots of missing values can be informative.

Options (ranked from best to worst)	**Comments**
Remove the variables with this problem from the analysis	The variables are usually in the analysis because they are relevant, so this is not ideal
Hire an expert	There really are very few genuine experts and there is little chance they will be interested in your problem
Use MAR cluster analysis and MAR latent class analysis methods and cross your finger	In Q and Displayr: **Latent Class Analysis**, **Mixed Mode Cluster Analysis** In Q, R, and Displayr**: K-Means (batch)**
Impute missing values and cross your fingers	In Q: **Automate > Browse Online Library > Missing Data > Impute**

Overview of issues

Data preparation issues

1. Don’t knows and non-responses
2. Ordinal variables
3. Nominal variables
4. Nominal variables with >3 categories
5. Using max-diff/conjoint/choice data
6. Questions with different ranges
7. Weights
8. No observations have complete data
9. Missing data is not MCAR
10. Missing data is non-ignorable

Issues addressed while forming segments

11. Yeah-saying bias in ratings
12. Scale effect in max-diff/conjoint/choice data
13. Working out the best number of segments
14. Algorithm has not converged
15. Local optima
16. The segmentation is dominated by a small number of variables
17. Measuring replicability
18. The segments are not replicable
19. Finding the most managerially-useful segments
20. Increasing the predictability of the segments

Issue 11: Yeah-saying bias in ratings

	Options (ranked from best to worst)	**Comments**
Issue When we run the analysis, we find that the key difference between segments is the average rating (e.g., a segment that says everything is important, and another that says nothing is important). The easiest way to check for this is to create a two segment solution.	Modify each person's data to have a mean of 0 and standard deviation of 1	In Q: **Standardize Data by Case.** This can be dangerous if there are missing data (as standardization implicitly assumes that each person has seen the same options).
	Change the distributional assumptions to focus on relativities	In Q: Duplicate the question, change the **Question Type** to **Ranking**, re-run the segmentation. This will only work with **Latent Class Analysis** and **Mixed Mode Cluster Analysis**

Issue 12: Scale effect in max-diff/conjoint/choice data

	Options (ranked from best to worst)	**Comments**
Issue When people answer max-diff and choice modelling/conjoint questions, they differ in how noisy they are. Some people give consistent responses. Others are a lot less consistent. This manifests itself by some segments seeming to regard everything as being relatively unimportant, while other segments have much stronger preferences.	Scale-adjusted latent class analysis	This can’t be done in Q. Try Latent Gold
	Estimate individual-level utilities/parameters, and then adjust them so that each respondent has a common standard deviation (or common maximum, or common minimum)	In Q: Create new R variables and write some code

Issue 13: Working out the best number of segments

Issue

How do you decide on the best number of segments?

Use your judgment and trade-off	Comments
Compare using a metric designed for automatically working out the number of segments	All the metrics are pretty dodgy. In Q, if using latent class analysis, the BIC is the default. Note that it automatically stops at a maximum of 10 (you can change this)
How strongly do the segments relate to other data?	In Q: **Smart Tables** and **Random Forest** are good ways of doing this
The fewer the segments the better	4 is often the "magic number". More than 8 is usually unwieldy. This is discussed in more detail on the next slide
Are the segments easy to name?	If you can't name them, they are hard to use
Are the segments inspirational?	Usually it is a good idea to engage the end-user in this stage
Perhaps: How replicable are the segments?	While this often appears in academic research, its practical relevance is doubtful.

Issue 14: Algorithm has not converged

	Solution	Comments
Issue Most cluster analysis algorithms were written a long time ago when computers were slow. Many have a “hack” built in whereby they stop their computation after an arbitrary amount of time. For example, the default k-means algorithm in R and SPSS run for 10 iterations.	Change the number of iterations to 1,000.	Most software will warn you if this problem occurs, but you will only see the warning if you read the technical outputs. Q defaults to 1,000 for the latent class and mixed-mode cluster analysis (with large number of segments this can be too small).

Issue 15: Local optima

	Solution	Comments
Issue *All* cluster analysis and latent class algorithms have guesses built into their algorithms (e.g., random start points, the order of the data file), which they seek to improve upon. These guesses can be poor, and the algorithm may get stuck in a poor solution (a *local optima*).	Re-run the algorithm multiple times, with different “guesses” (e.g., random start points)	In Q: All the algorithms have options for doing this. In the case of **Latent class analysis** and **Mixed-mode cluster analysis** it is an option in **Advanced: Number of starts** With cluster analysis algorithms good practice is to re-run them at least 100 times, and 1,000 times if you have the time. It is rarely practical to do this with max-diff, conjoint, choice, and ranking data, as they take too long to compute.

Issue 16:

The segmentation is dominated by a small number of variables

	Options (ranked from best to worst)	**Comments**
Issue Examples: 1. You have included data from two questions in your analysis, but the segmentation only reflects the data from one 2. There are 20 variables in the analysis, but you are only finding differences on 3	Reweight the variables/questions or remove variables or add variables in multiple times	In Q: Change the weight of questions and their distributional assumptions **(Advanced > Question Specific assumptions > Weight, Distribution),** and, change the range of variables within a question. In other programs, just change the range
	Use HOMALs or Multiple Correspondence Analysis to convert them to create new numeric components	This can be dangerous, as these techniques throw out a lot of data and reweight the data, so can get vastly inferior results

Issue 17: Measuring replicability

Issue

Ideally, the segmentation that you create could be reproduced by another person with a similar data set. This is relevant in academic environments where it is evidence that you have discovered something of scientific interest. But, in commercial segmentations this is more of a nice to have, as in reality you can have multiple good segmentations of most data sets (e.g., gender, age, lifestage). Furthermore, the most replicable segmentations tend to be uninteresting (e.g., yeah-saying biases replicate well) and can be local optima.

Options (ranked from best to worst)	**Comments**
Compute bootstrap replication many times	This is done in Q using **Create > Segments > Legacy Cluster Analysis.** However, this algorithm assumes MCAR and numeric data, so use with care!
Compute the bootstrap replication a small number of times (e.g., once)	See the next page.
Split-sample replication: split the sample into, say, halves, and compare the results you get when performing the segmentation in each half	• This tests that a similar segmentation can be replicated, but does not actually test your segmentation, as your segmentation will be based on the entire sample • Reweight the variables • In Q: Change the weight of questions and their distributional assumptions **(Advanced > Question Specific assumptions > Weight, Distribution),** and, change the range of variables within a question. In other programs, just change the range.

Computing bootstrap replication

- Basic algorithm
  - Create a new sample, the same size as the existing sample, by randomly sampling with replacement from the original
  - Compute your segments in the new *bootstrap sample*
  - Compare the allocation of respondents to segments (e.g., what proportion of people are in the same segment, remembering that the segment numbers are arbitrary). The standard way of doing this is to use the *adjusted rand statistic.*
    - In Q: Insert an **R Output** with code: `flipCluster::AdjustedRand(variable1, variable2)`
- A computational trick for doing this is to create a new variable which, for each case in the data file, shows the number of times it was randomly selected (i.e., 0, 1, 2, etc.). This trick both makes everything quite simple to do, and, allows incorporation of sampling weights.
  - In Q:
    - Insert a JavaScript variable
    - Set the **Access all data rows (advanced)**
    - Paste in the code to the right as the **Expression**
    - Tag the variable as a weight
    - Re-run the clustering or latent class analysis with this weight on

```
var wgt = new Array(N); // Creating an array to
                        // store the variable.
for (var i = 0; i < N; i++) // Setting the
                            //initial values to 0.
    wgt[i] = 0;
// Using random sampling with replacement to
// count up the number of times to replicate
// the data.
var seed = 1;
function random() { // Slightly dodgy
// see http://stackoverflow.com/a/19303725/1547926
    var x = Math.sin(seed++) * 10000;
    return Math.floor((x - Math.floor(x)) * (N - 1));
}
var counter = 0;
while (counter++ < N)
    wgt[random()]++; // Incrementing the number
              //of times the value has been selected

// Replace 'WEIGHT' with the name of your weight
// variable. If there is none, delete the next 2 lines
for (var i = 0; i < N; i++) {
    wgt[i] *= WEIGHT[i];
}
wgt
```

Issue 18: Increasing replicability

Issue

If the replicability of the segments is low (e.g., less than 80%), and this is considered to be a problem (as discussed, while replicability is nice, it is not really a must have.)

Options (ranked from best to worst)	**Comments**
Use a method specifically designed to maximize replicability, such as bagging	In Q: Insert an R Output and use the bagged function in the e1071 package (by writing code). However, this algorithm assumes MCAR and numeric data, so use with care!
Try different combinations of variables.	

Issue 19: Finding the most managerially useful segments

Issue

If you follow all of the steps above, you can still end up with a segmentation that is entirely uninteresting. This is because the steps above are all designed to address statistical issues, but segmentation is ultimately about management.

Solution

The solution is to use lots of different methods and compare them (using the same techniques that are used to select the number of segments). Things that can work are:

- Investigate different number of segments
- Change the data used
- Increase the number of starts
- Use PCA, factor analysis, HOMALS, or multiple correspondence analysis on the input variables
- Standardizing the data in a different way
- If using latent class analysis, change the various distributional assumptions for latent class analysis (but not for Ranking and Experiment questions)
- Using a different algorithm (e.g., in Q, K-Means with Bagging)

Comments

As discussed in Issue 13, we can compare based on:

- How strongly do the segments relate to other data?
- The fewer segments the better
- Are the segments easy to name?
- Are the segments inspirational?
- Perhaps: replicability

This process should take days, and a systematic process should be used

Evaluating the segments

	Number 0.001 tables of other data	Easy to name	How inspirational
2 Segments unscaled	4	Moderate	
4 Segments Unscaled	10		
6 Segments unscaled	13		
8 Segments unscaled	13		
2 Segments standardized	11	Yes	
4 Segments Standardized	16	No	
Etc.			

Issue 20: Increasing the predictability of the segments

Issue

The segments do not correlate with any other data. Or, the correlations are weak.

Options (ranked from best to worst)	**Comments**
For each of the segmentation variables (i.e., the variables that we are using in the cluster or latent class analysis), build a predictive model where they are the *outcome* and the *profiling variables* (i.e., the variables that we want to be correlated with the segments) are the *predictors*. Weight the variables in the segmentation according to the accuracy of the predictive model (see Issue 16).	In Q: **Classifier > Random Forest** is probably the most appropriate predictive model. If you get weird technical errors, it probably means you have categories with really small sample sizes, that need to be merged, or variables with too little data that need to be excluded.
Same as the previous option, except that the *predicted values* of the predictor models are used as the segmentation variables.	This guarantees a high level of correlation, but it will, in part, be spurious, due to the inevitable over-fitting of the predictive models.
Use concomitant/covariate variable latent class algorithms.	Q does not offer such models. This is a theoretically elegant solution, but I have never seen it actually work to solve this issue.
Add demographics or other predictor variables to the segmentation	While this can work, it is a high-risk approach. It can also lead to a lot of implementation problems, whereby the segments that are described in the research end up being very different to those that are experienced in the implementation. When doing this approach, it can be useful to also use PCA, multiple correspondence analysis, or HOMALs and use the resulting components in the segmentation.

Overview of issues

Data preparation issues

1. Don’t knows and non-responses
2. Ordinal variables
3. Nominal variables
4. Nominal variables with >3 categories
5. Using max-diff/conjoint/choice data
6. Questions with different ranges
7. Weights
8. No observations have complete data
9. Missing data is not MCAR
10. Missing data is non-ignorable

Issues addressed while forming segments

11. Yeah-saying bias in ratings
12. Scale effect in max-diff/conjoint/choice data
13. Working out the best number of segments
14. Algorithm has not converged
15. Local optima
16. The segmentation is dominated by a small number of variables
17. Measuring replicability
18. The segments are not replicable
19. Finding the most managerially-useful segments
20. Increasing the predictability of the segments

Summary: Algorithm choice

Hierarchical cluster analysis

A nice idea for its time. Its time was 70 years ago…

Neural networks
(e.g. SOM, auto-encoding)

The best solution for a very small number of exotic problems (e.g., online learning), which rarely occur in survey research.

Cluster analysis
(e.g. *k*-means, *k*-medoids, bagged *k*-means)

Great for some problems: big, numeric data. OK for most problems.

Latent Class Analysis

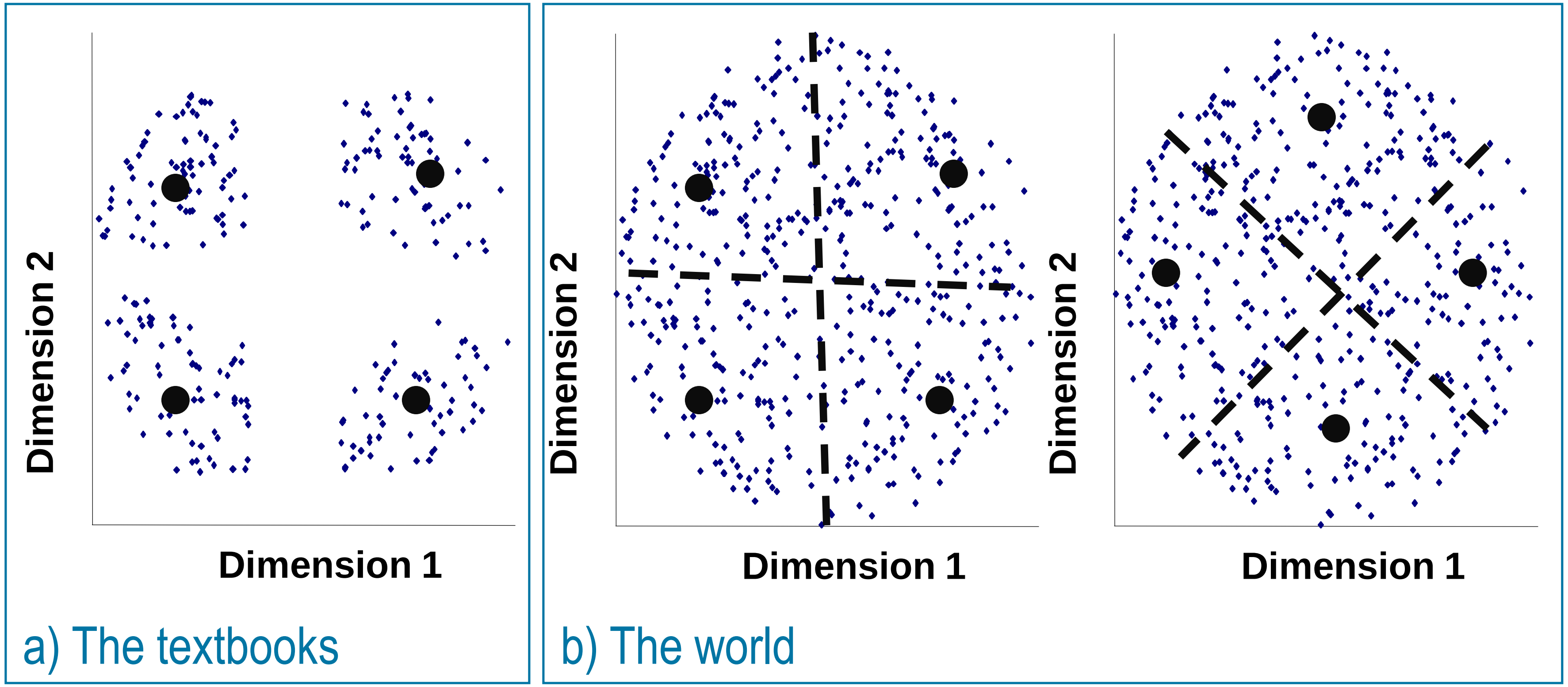
The best or equal-best solution for the vast majority of problems (the previous slides explained why)

RESEARCHSOFTWARE

A DIVISION OF DISPLAYR

TIM BOCK PRESENTS

Appendices



a) The textbooks

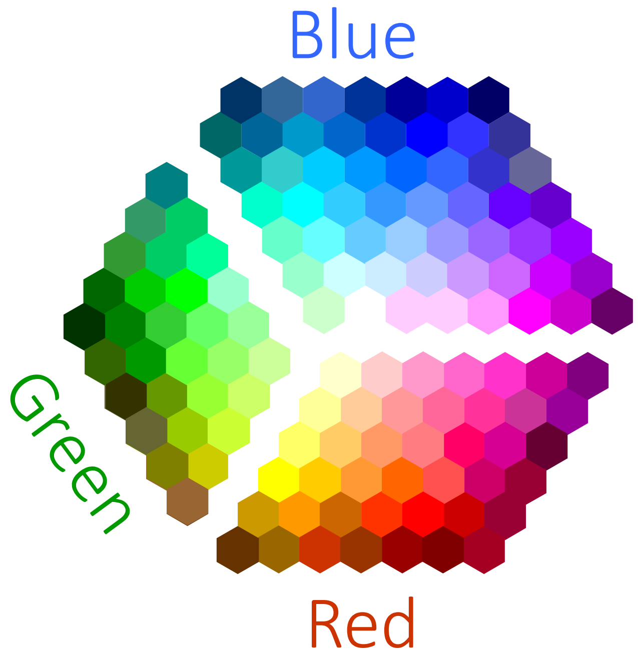
b) The world

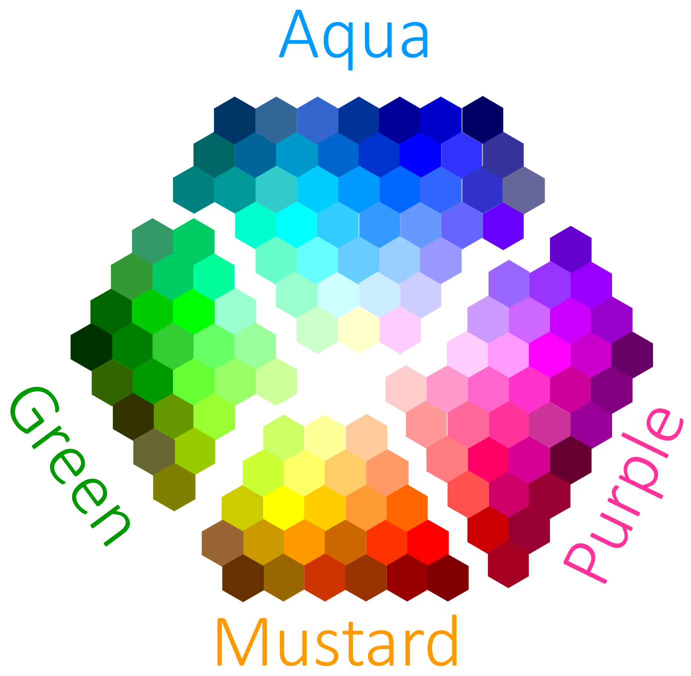
STUDIES HAVE FOUND THAT MARKETS DO NOT CONTAIN A SMALL NUMBER
OF DIFFERENT TYPES OF CONSUMERS

There are many different types of consumers

IN ANY MARKET, THERE ARE AS MANY "UNIQUE" CONSUMERS AS THERE ARE COLOURS

There are thus many possible segmentations





MYTH: SEGMENTS ARE “IDENTIFIED” BY RESEARCH
REALITY: THE “DATA” CANNOT TELL US HOW MANY SEGMENTS WE NEED, OR, WHERE THE BOUNDARIES BETWEEN THE SEGMENTS SHOULD BE

RESEARCHSOFTWARE
A DIVISION OF DISPLAYR

TIM BOCK PRESENTS

Q&A Session