Week 6: *Conjoint Questions*

🏛 EMSE 6035: Marketing Analytics for Design Decisions

👤 John Paul Helveston

📅 October 02, 2024

Some Quarto tips

Convert a data frame to a markdown table with `kable()`

```
library(tidyverse)

mtcars %>%
   kable()
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4

Example from last year

```
library(tidyverse)
library(here)

df <- read_csv(here("data", "competitors.csv"))
df %>%
  kable()
```

brand	volumeInOz	quantity	price	pricePerCup	biodegrability	opacity	logo
greatValue	9	50	$2.98	$0.06	FALSE	TRUE	FALSE
decorRack	9	50	$5.99	$0.12	FALSE	TRUE	FALSE
tigerChef	9	100	$7.99	$0.08	FALSE	TRUE	FALSE
smartly	9	80	$2.79	$0.03	FALSE	FALSE	FALSE
solo	9	50	$4.04	$0.08	FALSE	FALSE	FALSE
greatValue	9	100	$3.76	$0.04	FALSE	FALSE	FALSE
ecoProducts	9	1000	$187.69	$0.19	TRUE	FALSE	TRUE
worldCentric	9	2000	$220.00	$0.11	TRUE	FALSE	TRUE
hefty	18	50	$3.98	$0.08	FALSE	TRUE	FALSE

More `kable()` formatting options:
{kableExtra} package

References

Simple approach: Insert a footnote with `^[]`

This markdown…

```
The Eiffel Tower is 324 meters
tall^[From the [Eiffel Tower
wikipedia page]
(https://en.wikipedia.org/wiki
/Eiffel_Tower)]
```

…renders as this

The Eiffel Tower is 324 meters tall[1]

[1]From the Eiffel Tower wikipedia page

References

Complex (but more complete) approach: Use bibtex

https://quarto.org/docs/authoring/footnotes-and-citations.html

You can insert citations with `[@citekey]`, and a "References" table will be automatically created.

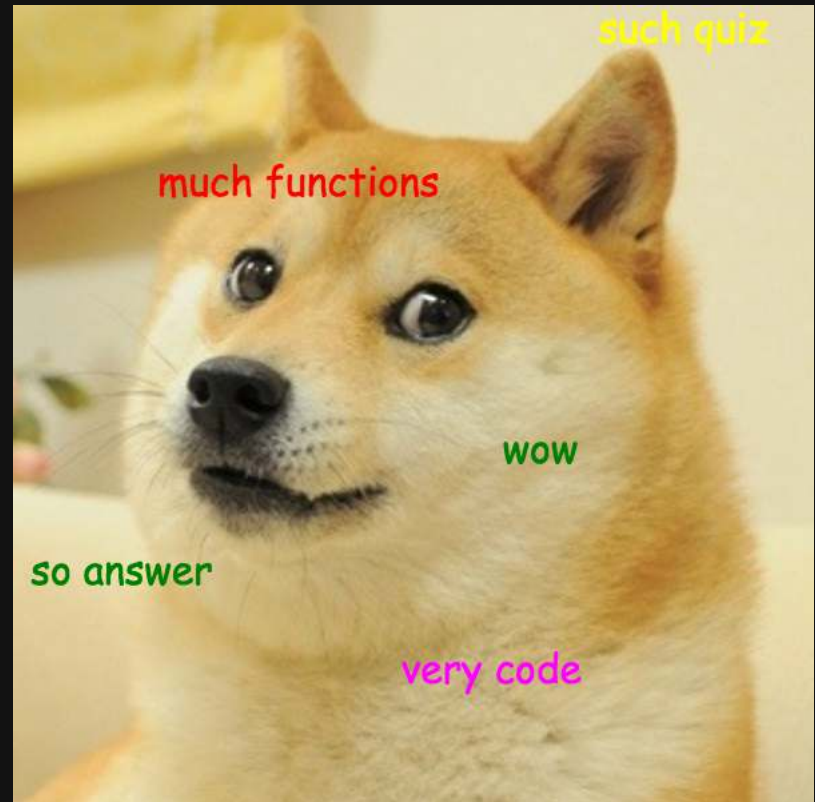**Footnotes are perfectly fine for this class**

Quiz 2

Download the template from the #class channel

Make sure you unzip it!

When done, submit your `quiz2.qmd` on Blackboard

10:00

Week 6: *Conjoint Questions*

1. Defining choice questions in R

2. Displaying choice questions in surveydown

Week 6: *Conjoint Questions*

1. Defining choice questions in R

2. Displaying choice questions in surveydown

Download & unzip two repos:

- logitr-cars

- demo-choice-based-conjoint

(code used in the blog post on conjoint in surveydown)

Open `logitr-cars.Rproj`

We'll be using the {cbcTools} package today

```r
install.packages("cbcTools")
```

Choice question components

1. Generate `profiles` for each attribute and level

2. Create a survey `design` data frame from `profiles`

Basic Design

Any combination of attributes can be shown in each choice question

Question 1

Option:	1	2	3
Price:	$15	$25	$25
Fuel Economy:	25 (mpg)	25 (mpg)	25 (mpg)
Accel. Time:	6 (s)	7 (s)	8 (s)
Powertrain:	Electric	Gasoline	Electric

Question 2

Option:	1	2	3
Price:	$20	$25	$25
Fuel Economy:	25 (mpg)	20 (mpg)	25 (mpg)
Accel. Time:	7 (s)	6 (s)	7 (s)
Powertrain:	Gasoline	Gasoline	Electric

Labeled Design

One attribute is used as the "label" - choice options are fixed according to the label

Question 1

Option:	Gasoline	Electric
Price:	$15	$20
Fuel Economy:	20 (mpg)	30 (mpg)
Accel. Time:	8 (s)	7 (s)

Question 2

Option:	Gasoline	Electric
Price:	$20	$20
Fuel Economy:	20 (mpg)	20 (mpg)
Accel. Time:	7 (s)	7 (s)

Design with a "None" option

A "none" option means they can choose an "other" option

Question 1

Option:	1	2	3	None
Price:	$15	$20	$20	
Fuel Economy:	20 (mpg)	25 (mpg)	25 (mpg)	
Accel. Time:	7 (s)	6 (s)	7 (s)	
Powertrain:	Gasoline	Electric	Gasoline	

Question 2

Option:	1	2	3	None
Price:	$15	$25	$20	
Fuel Economy:	25 (mpg)	30 (mpg)	25 (mpg)	
Accel. Time:	7 (s)	8 (s)	8 (s)	
Powertrain:	Electric	Gasoline	Gasoline	

Attribute-specific features

Some attributes may only be valid for certain levels of other attributes

> **Example**: The driving range of an electric vehicle (EV) only applies to EVs and not gasoline-powered vehicles.

To implement this, edit `profiles` prior to using `cbc_design()`

(see `logitr-cars` code 1.2)

Restricted profiles

Sometimes you may want to not allow a specific combination of features - use `cbc_restrict()` to implement this

(see `logitr-cars` code 1.3)

Warning: Avoid restrictions if possible!

Your Turn

Open `survey.Rproj` in "demo-choice-based-conjoint"

With your team, discuss the specific choice question design for your project

- Regular or labeled?
- Include a "none" option (outside good) or not?
- Include restrictions?

Edit the `make-choice-questions.R` file to design your choice questions.

Week 6: *Conjoint Questions*

1. Defining choice questions in R

2. Displaying choice questions in surveydown

Displaying choice questions in your survey

1. Read in the design file (`choice_questions.csv`)

2. Sample and store a random respondent ID

3. Filter the design for the respondentID

4. Define a function to create question options

5. Create the options for each choice question

6. Create each choice question (6 in total)

1. Read in the design file (choice_questions.csv)

```r
design <- readr::read_csv("choice_questions.csv")
```

2. Sample and store a random respondent ID

Sample a random respondentID

```
respondentID <- sample(design$respID, 1)
```

Store the respondentID

```
sd_store_value(respondentID, "respID")
```

In your survey data, you will have a `respID` column

3. Filter the design for the respondentID

```r
df <- design %>%
  filter(respID == respondentID) %>%
  mutate(image = paste0("images/", image))
```

Images are stored in `"images"` folder, so we paste on that path

Result looks like this:

```r
head(df)
```

```
#> # A tibble: 6 × 9
#>    respID   qID altID obsID profileID type        price freshness image
#>     <dbl> <dbl> <dbl> <dbl>     <dbl> <chr>       <dbl> <chr>     <chr>
#> 1   1646     1     1 13161        37 Gala            1 Average   images/gala.jpg
#> 2   1646     1     2 13161         3 Honeycrisp      1 Excellent
#> images/honeycrisp.jpg
#> 3   1646     1     3 13161        47 Gala            2 Average   images/gala.jpg
#> 4   1646     2     1 13162        51 Fuji          2.5 Average   images/fuji.jpg
#> 5   1646     2     2 13162        31 Fuji            4 Excellent images/fuji.jpg
#> 6   1646     2     3 13162        75 Red Delicious   1 Poor
```

4. Define a function to create question options

```r
make_cbc_options <- function(df) {
  alt1 <- df |> filter(altID == 1)
  alt2 <- df |> filter(altID == 2)
  alt3 <- df |> filter(altID == 3)

  options <- c("option_1", "option_2",
"option_3")

  names(options) <- c(
    glue("
      **Option 1**<br>
      <img src='{alt1$image}' width=100><br>
      **Type**: {alt1$type}<br>
      **Price**: $ {alt1$price} / lb
    "),
    glue("
      **Option 2**<br>
      <img src='{alt2$image}' width=100><br>
      **Type**: {alt2$type}<br>
      **Price**: $ {alt2$price} / lb
    "),
    glue("
      **Option 3**<br>
      <img src='{alt3$image}' width=100><br>
      **Type**: {alt3$type}<br>
      **Price**: $ {alt3$price} / lb
    ")
  )
  return(options)
}
```

4. Define a function to create question options

Make one-row data frames for each alternative.

For example, `alt1` looks like this:

```
#> Rows: 1
#> Columns: 9
#> $ respID    <dbl> 1646
#> $ qID       <dbl> 1
#> $ altID     <dbl> 1
#> $ obsID     <dbl> 13161
#> $ profileID <dbl> 37
#> $ type      <chr> "Gala"
#> $ price     <dbl> 1
#> $ freshness <chr> "Average"
#> $ image     <chr> "images/gala.jpg"
```

```r
make_cbc_options <- function(df) {
  alt1 <- df |> filter(altID == 1)
  alt2 <- df |> filter(altID == 2)
  alt3 <- df |> filter(altID == 3)

  options <- c("option_1", "option_2",
"option_3")

  names(options) <- c(
    glue("
      **Option 1**<br>
      <img src='{alt1$image}' width=100><br>
      **Type**: {alt1$type}<br>
      **Price**: $ {alt1$price} / lb
    "),
    glue("
      **Option 2**<br>
      <img src='{alt2$image}' width=100><br>
      **Type**: {alt2$type}<br>
      **Price**: $ {alt2$price} / lb
    "),
    glue("
      **Option 3**<br>
      <img src='{alt3$image}' width=100><br>
      **Type**: {alt3$type}<br>
      **Price**: $ {alt3$price} / lb
    ")
  )

  return(options)
}
```

4. Define a function to create question options

```
alt1[c('image', 'type', 'price')]
```

```
#> # A tibble: 1 × 3
#>   image           type  price
#>   <chr>           <chr> <dbl>
#> 1 images/gala.jpg Gala      1
```

Highlighted section renders as:

```
#> **Option 1**<br>
#> <img src='images/gala.jpg' width=100>
<br>
#> **Type**: Gala<br>
#> **Price**: $ 1 / lb
```

```r
make_cbc_options <- function(df) {
  alt1 <- df |> filter(altID == 1)
  alt2 <- df |> filter(altID == 2)
  alt3 <- df |> filter(altID == 3)

  options <- c("option_1", "option_2",
"option_3")

  names(options) <- c(
    glue("
    **Option 1**<br>
    <img src='{alt1$image}' width=100><br>
    **Type**: {alt1$type}<br>
    **Price**: $ {alt1$price} / lb
    "),
    glue("
    **Option 2**<br>
    <img src='{alt2$image}' width=100><br>
    **Type**: {alt2$type}<br>
    **Price**: $ {alt2$price} / lb
    "),
    glue("
    **Option 3**<br>
    <img src='{alt3$image}' width=100><br>
    **Type**: {alt3$type}<br>
    **Price**: $ {alt3$price} / lb
    ")
  )

  return(options)
}
```

5. Create the options for each choice question

```
cbc1_options <- make_cbc_options(df |> filter(qID == 1))
```

Value stored in your data:

```
#> option_1
```

Label:

```
#> [1] "  **Option 1**<br>\n  <img
src='images/gala.jpg' width=100><br>\n
**Type**: Gala<br>\n  **Price**: $ 1 / lb"
```

Renders as:

Option 1

Type: Gala
Price: $ 1 / lb

5. Create the options for each choice question

Now make the options for each question

```r
cbc1_options <- make_cbc_options(df |> filter(qID == 1))
cbc2_options <- make_cbc_options(df |> filter(qID == 2))
cbc3_options <- make_cbc_options(df |> filter(qID == 3))
cbc4_options <- make_cbc_options(df |> filter(qID == 4))
cbc5_options <- make_cbc_options(df |> filter(qID == 5))
cbc6_options <- make_cbc_options(df |> filter(qID == 6))
```

6. Create each choice question (6 in total)

Code:

```
sd_question(
  type   = 'mc_buttons',
  id     = 'cbc_q1',
  label  = "(1 of 6) If these were your
only options, which would you choose?",
  option = cbc1_options
)
```

Renders as:

TABLE layout: Show options in a table with `kable()`

```r
library(dplyr)

alts <- df %>%
  filter(qID == 1) %>%
  mutate(
    # Add $ sign to price and html code for image path
    price = paste(scales::dollar(price), "/ lb"),
    image = paste0('<img src="', image, '" width=100>')) %>%
  # Make nicer attribute labels
  select(
    `Option:`    = altID,
    ` `          = image,
    `Price:`     = price,
    `Type:`      = type,
    `Freshness:` = freshness)

# Drop row names
row.names(alts) <- NULL
```

Option: Show options in a table with `kable()`

Display the *transpose*, `t(alts)`

```
kbl(t(alts), escape = FALSE) %>%
  kable_styling(
    bootstrap_options = c("striped", "hover", "condensed"),
    full_width = FALSE,
    position = "center"
  )
```

Option:	1	2	3
Price:	$1 / lb	$1 / lb	$2 / lb
Type:	Gala	Honeycrisp	Gala
Freshness:	Average	Excellent	Average

TABLE layout: Show options in a table with `kable()`

Your choice question can be defined in the `survey.qmd` file as it is just choosing one of three options:

Your Turn

With your team, choose a format to work with (buttons or table) for your choice question, then modify the example survey to match your project.

Versions:

Buttons layout: demo-choice-based-conjoint

Table layout: demo-choice-based-conjoint-table