Fielding Surveys with formr

John Paul Helveston, Ph.D.

The George Washington University | Dept. of Engineering Management and Systems Engineering

June 15, 2023

Making a survey in formr

Building a survey in formr

- Use *RMarkdown / html* to create survey elements

- Copy elements to a *Google Sheet*

- Import Google Sheets into formr *surveys*

- Link surveys together in formr *runs*

My Recommentation:
Draft your survey in RMarkdown

Survey content in `demoSurvey.Rmd`

Google sheet

Live survey

formr row types (more here)

Type	Description
`note`	Display content in `label` column
`submit`	Next page button
`mc`	Multiple choice question (single choice)
`mc_multiple`	Multiple choice question (multiple choices)
`mc_button`	Multiple choice question (large buttons)
`select_one`	Drop down menu (choose one)
`text`	Open text, single row
`textarea`	Open text, block

Some Guidelines

- Be sure that any data / images are hosted somewhere on the web
- Consider each new page a **New R Session** (reload libraries, etc.)

Embedding images

I recommend just writing html code, like this

```
<img src="https://github.com/jhelvy/2023-qux-conf-conjoint/blob/main/images/logo.png?raw=t
```

Centered image

I recommend just writing html code, like this

```
<center>
<img src="https://github.com/jhelvy/2023-qux-conf-conjoint/blob/main/images/logo.png?raw=t
</center>
```

Check your urls carefully!

This is the link to the **Github page** with the image:

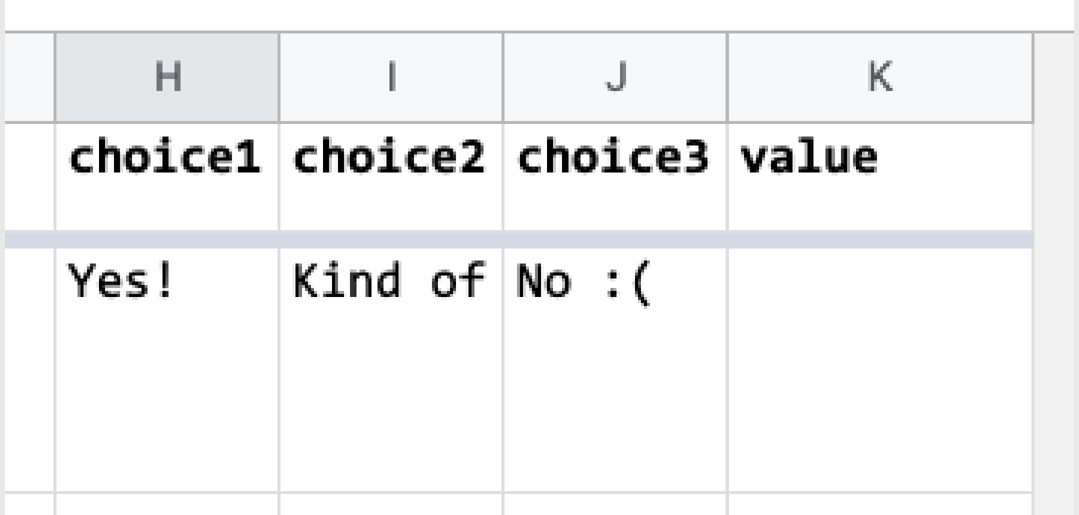
https://github.com/jhelvy/2023-qux-conf-conjoint/blob/main/images/logo.png

This is the link to the **actual image**:

https://github.com/jhelvy/2023-qux-conf-conjoint/blob/main/images/logo.png?raw=true

Two ways to define choice options

Add "choice" columns



H	I	J	K
choice1	choice2	choice3	value
Yes!	Kind of	No :(	

]

Use `choices` tab
(when you have a lot of choices)

Example: "Year of birth" in this demo

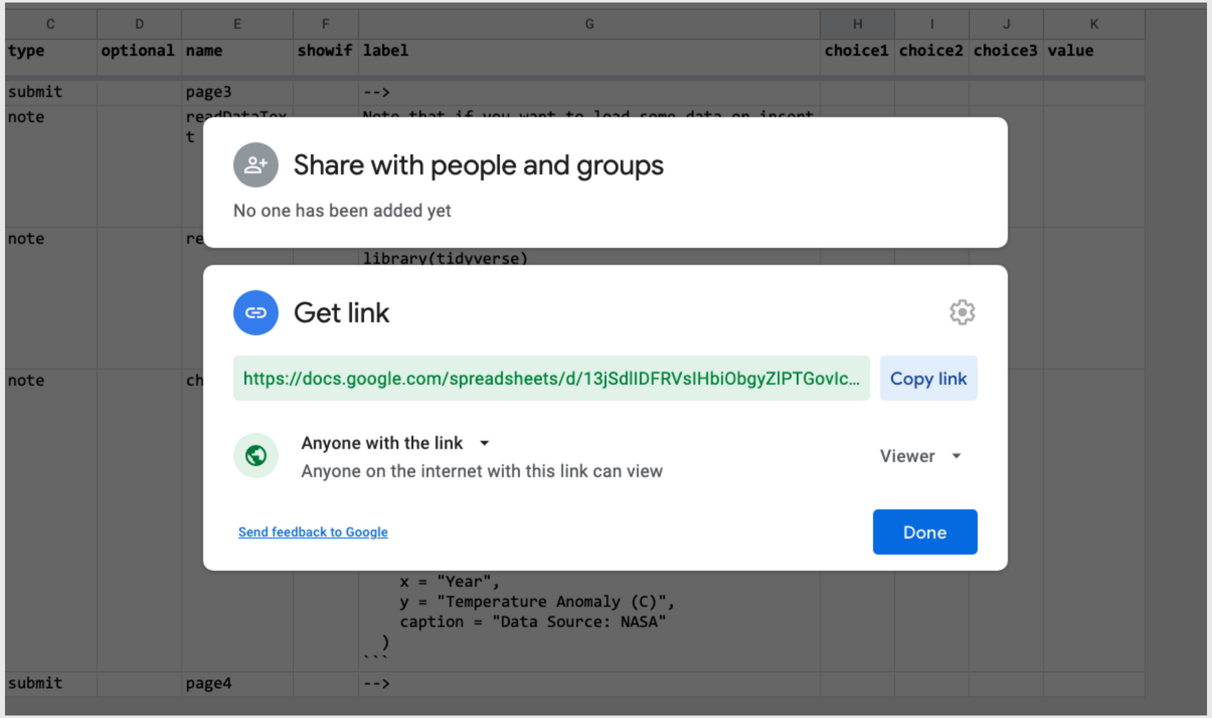
Control the way things look in `class` column

(options here)

Importing survey into formr

formr.org --> Admin --> Surveys --> Create new survey

(Make sure your Google Sheet is visible!)



C	D	E	F	G	H	I	J	K
type	optional	name	showif	label	choice1	choice2	choice3	value
submit		page3		-->				
note		readDataText		Note that if you want to load some data on insert				
note		re		library(tidyverse)				
note		ch		x = "Year", y = "Temperature Anomaly (C)", caption = "Data Source: NASA") ```				
submit		page4		-->				

Share with people and groups

No one has been added yet

Get link

https://docs.google.com/spreadsheets/d/13jSdlIDFRVsIHbiObgyZlPTGovlc...

Copy link

Anyone with the link

Anyone on the internet with this link can view

Viewer

Send feedback to Google

Done

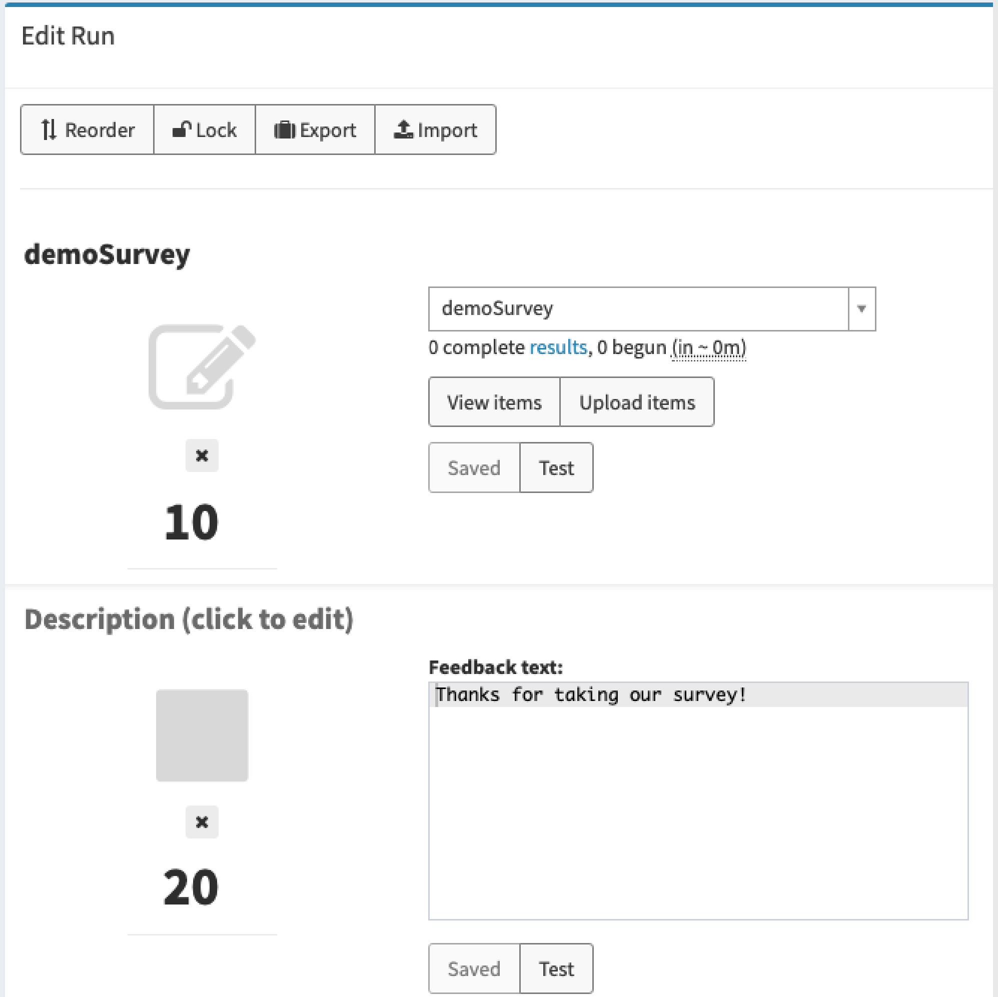
Make a run

formr.org --> Admin --> Runs --> Create new run

Insert survey with ✎

Insert stop with ■

Change order by adjusting numbers & clicking "Reorder"



Edit Run

demoSurvey

demoSurvey

0 complete results, 0 begun (in ~ 0m)

10

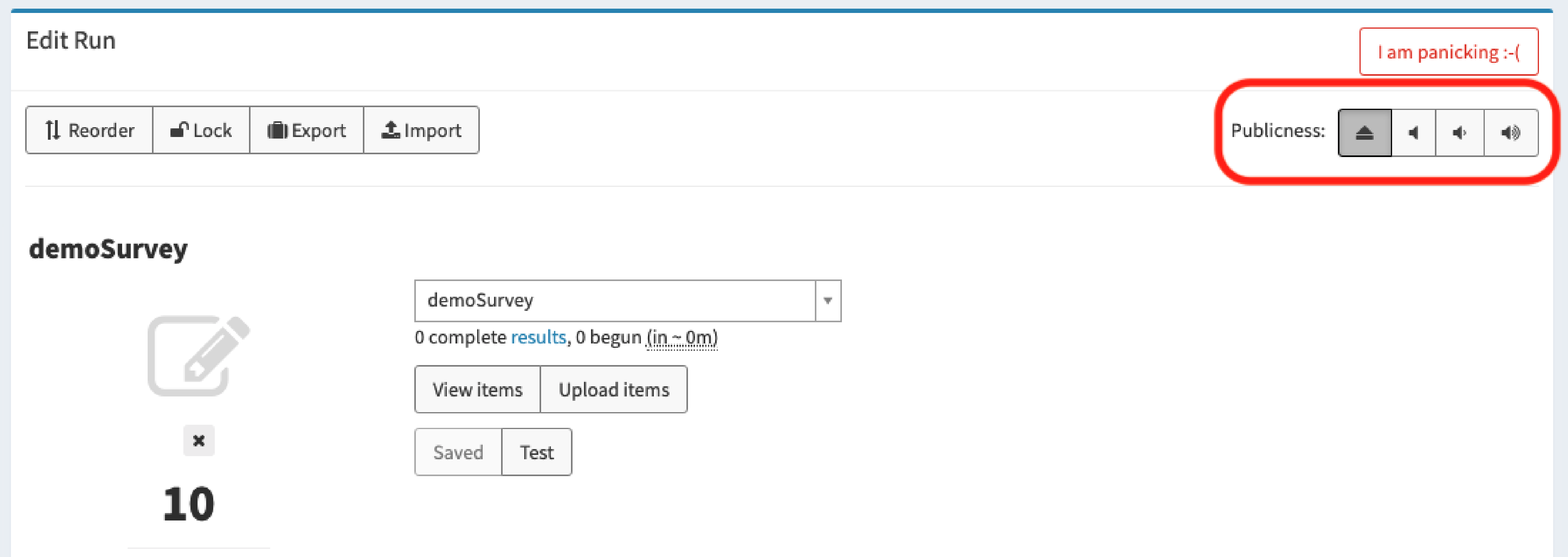
Description (click to edit)

Feedback text:

Thanks for taking our survey!

20

Make it "live" with the volume buttons



Edit Run

demoSurvey

demoSurvey

0 complete results, 0 begun (in ~ 0m)

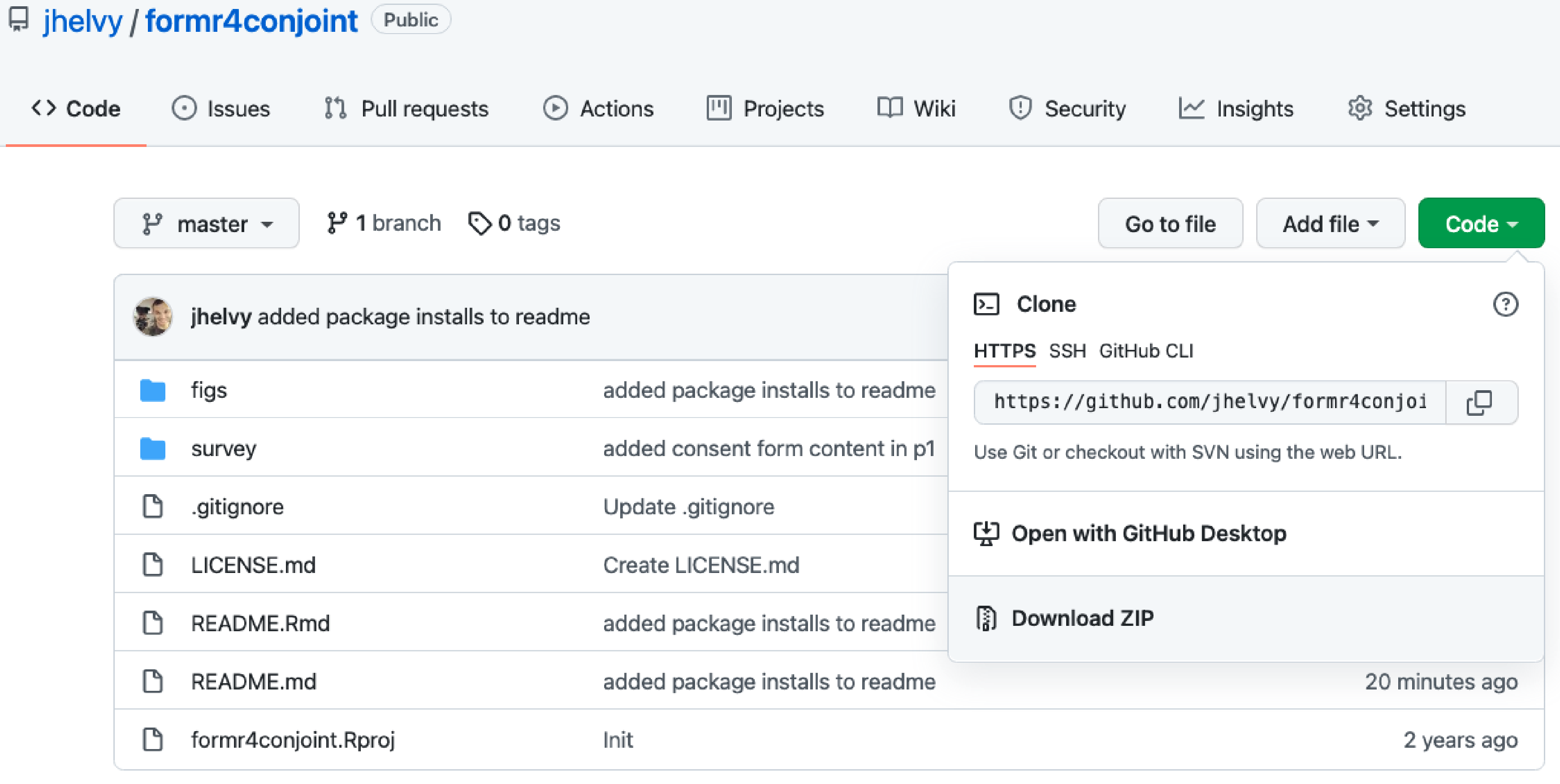
Fine tune look & feel in "Settings"

Making a *conjoint* survey in formr

(Detailed demo in this blog post)

Full demo in the formr4conjoint repo from GitHub

(code used in the related blog post)



jhelvy / formr4conjoint Public

jhelvy added package installs to readme

figs	added package installs to readme	
survey	added consent form content in p1	
.gitignore	Update .gitignore	
LICENSE.md	Create LICENSE.md	
README.Rmd	added package installs to readme	
README.md	added package installs to readme	20 minutes ago
formr4conjoint.Rproj	Init	2 years ago

Clone

HTTPS SSH GitHub CLI

```
https://github.com/jhelvy/formr4conjoi
```

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

3 Parts

- **Part 1**: Intro
- **Part 2**: Conjoint questions
- **Part 3**: Other / demographic questions

3 Parts

- **Part 1**: Intro --> screen for target population
- **Part 2**: Conjoint questions --> screen for random answers
- **Part 3**: Other / demographic questions

Displaying your choice questions online

(See example in part two demo google sheet)

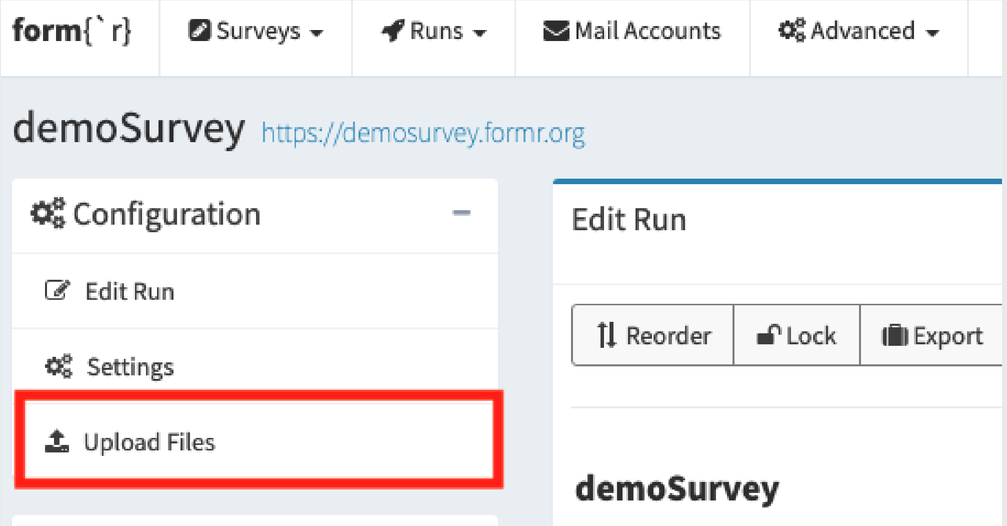
1. Export your choice questions as a .csv file
2. Upload your .csv file somewhere (e.g. GitHub)
3. Use R code to extract the values to display
4. Use RMarkdown to display the values

1. Export your experiment design (from {cbcTools}) as a .csv file

```
write_csv(design, here('choice_questions.csv'))
```

2. Upload your .csv file somewhere

Inside a formr run (private)



github.com (public)

apples example

Serialize the experiment design

Converts a data frame to one long string

```
df
```

```
#>   profileID altID price fuelEconomy accelTime powertrain
#> 1         9     1    25          30         6   Gasoline
#> 2        11     2    20          20         7   Gasoline
#> 3        23     3    20          25         8   Gasoline
```

```
df_json <- jsonlite::serializeJSON(df)
df_json
```

```
#> {"type":"list","attributes":{"names":{"type":"character","attributes":{},"value":["prof
```

Using the calculate type (example sheet)

RMarkdown

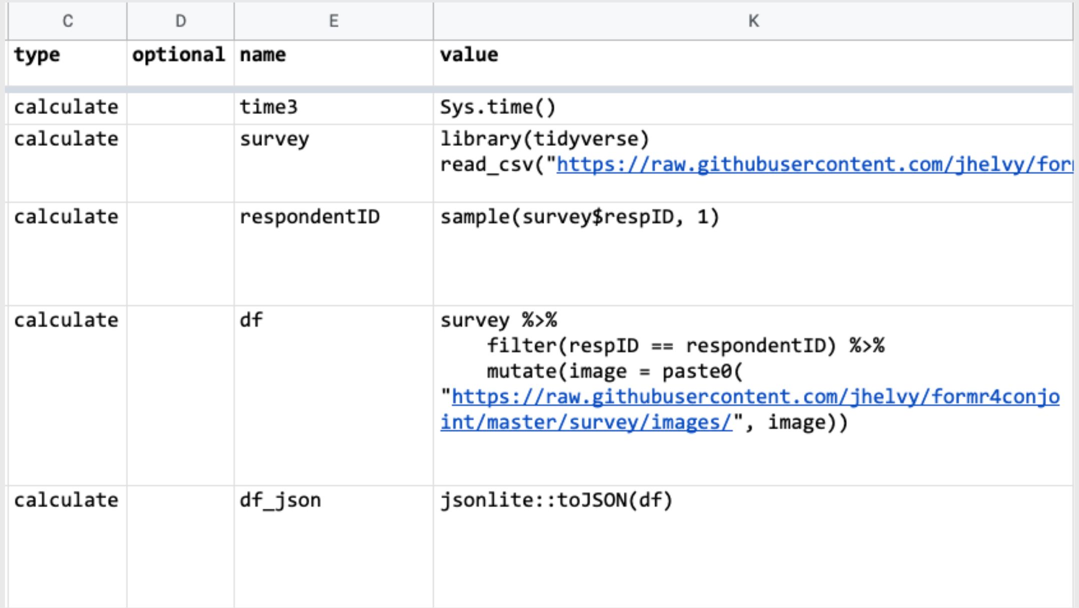
```r
# Read in the choice questions
library(tidyverse)
design <- read_csv("https://raw.githubuserco

# Define the respondent ID
respondentID <- sample(design$respID, 1)

# Create the subset of rows for that respond
df <- design %>%
    filter(respID == respondentID) %>%
    mutate(image = paste0("https://raw.githu

# Convert df to json
df_json <- jsonlite::serializeJSON(df)
```

Google sheet



C	D	E	K
type	**optional**	**name**	**value**
calculate		time3	Sys.time()
calculate		survey	library(tidyverse) read_csv("https://raw.githubusercontent.com/jhelvy/for
calculate		respondentID	sample(survey$respID, 1)
calculate		df	survey %>% filter(respID == respondentID) %>% mutate(image = paste0("https://raw.githubusercontent.com/jhelvy/formr4conjoint/master/survey/images/", image))
calculate		df_json	jsonlite::toJSON(df)

Random choice questions as **buttons**

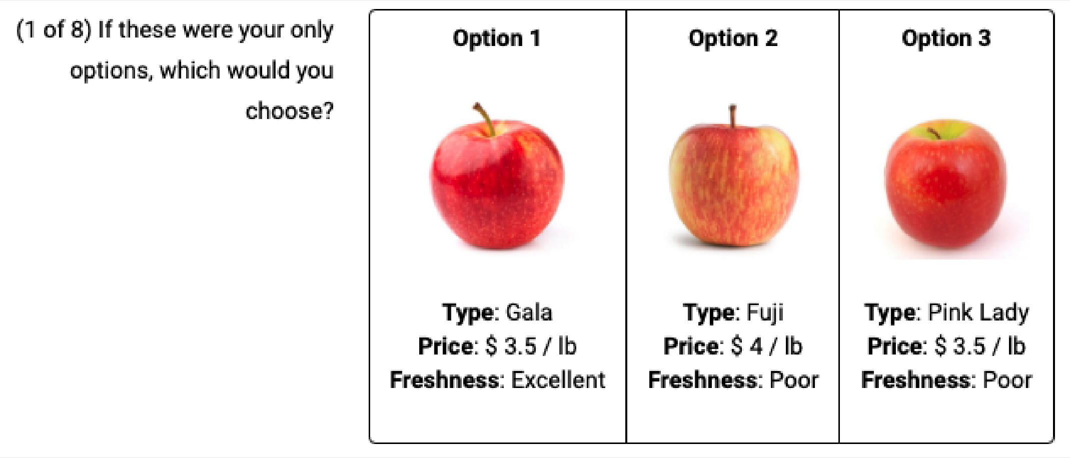
Use the `mc_button` question type

`label`

- Show your question text
- Insert a code chunk to create one-row data frame for each alternative

`choice` columns

- Insert RMarkdown code to display each alternative



(1 of 8) If these were your only options, which would you choose?

Option 1	Option 2	Option 3
Type: Gala	**Type**: Fuji	**Type**: Pink Lady
Price: $ 3.5 / lb	**Price**: $ 4 / lb	**Price**: $ 3.5 / lb
Freshness: Excellent	**Freshness**: Poor	**Freshness**: Poor

Random choice questions as **buttons**

Create separate data frames for each alternative

```r
library(dplyr)

alts <- jsonlite::unserializeJSON(df_json)
alt1 <- alts %>% filter(altID == 1)
alt2 <- alts %>% filter(altID == 2)
alt3 <- alts %>% filter(altID == 3)
```

Use RMarkdown formatting to display content in each alternative

```
**Option 1**

**Price**: $ `r alt1$price`
**Powertrain**: $ `r alt1$powertrain`
**Fuel Economy**: `r alt1$fuelEconomy` mpg
**0-60 Accel. Time**: `r alt1$accelTime` s
```

Option 1

Price: $ 25
Powertrain: $ Gasoline
Fuel Economy: 30 mpg
0-60 Accel. Time: 6 s

Random choice questions as **table**

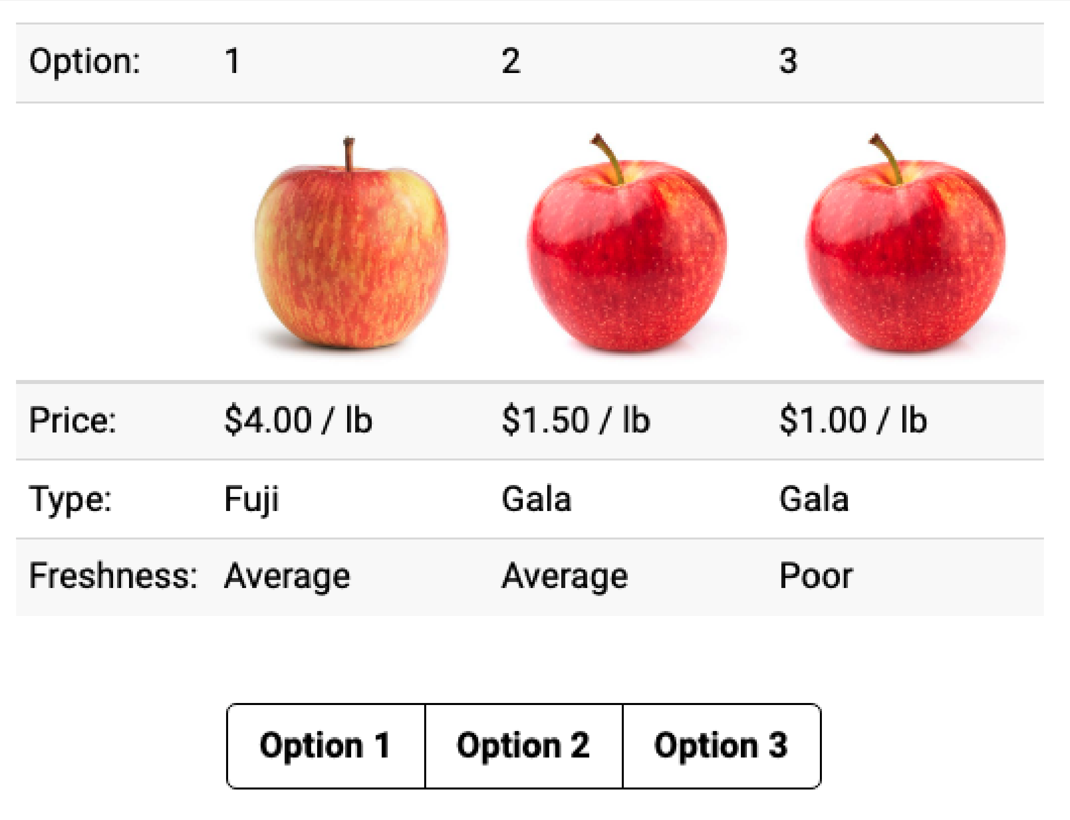
- Use the `mc_button` question type

`label`

- Show your question text
- Insert a code chunk to modify `alts` data frame & display it using `kable()`
- Use kableExtra to control table styling

`choice` columns

- Simple text / number for each option



Option:	1	2	3
Price:	$4.00 / lb	$1.50 / lb	$1.00 / lb
Type:	Fuji	Gala	Gala
Freshness:	Average	Average	Poor

Random choice questions as **table**

```r
library(dplyr)

alts <- jsonlite::unserializeJSON(df_json) %
  # Add $ sign to price
  mutate(price = scales::dollar(price)) %>%
  # Make nicer attribute labels
  select(
    `Option:`              = altID,
    `Powertrain:`          = powertrain,
    `Price:`               = price,
    `Fuel Economy (mpg):` = fuelEconomy,
    `Accel. Time (s):`     = accelTime)

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

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

```r
kable(t(alts))
```

Option:	1	2	3
Powertrain:	Gasoline	Gasoline	Gasoline
Price:	$25	$20	$20
Fuel Economy (mpg):	30	20	25
Accel. Time (s):	6	7	8

Back to workshop website:

https://jhelvy.github.io/2023-qux-conf-conjoint/

@JohnHelveston
@jhelvy
@jhelvy
jhelvy.com
jph@gwu.edu