Statistical Perceptions Serverless Survey

Contents

These documents outline how to use for researchers and for potential contributors.

Details that are for contributors denoted with the </> icon, when mixed into a page with other information it will be in a dropdown.

Working with Configuration Files

A configuration file is yaml

How YAML works

No shared parameters

This file is setup like:

Skip to main content

```
  name_of_var1_for_q1: value_for_var1_q1
  name_of_var2_for_q1: value_for_var2_q1
  figure_values:
    name_of_fig_var1_for_q1: value_for_fig_var1_for_q1
- question_id: unique_id_for_q2
  name_of_var1_for_q2: value_for_var1_q2
  name_of_var3_for_q2: value_for_var3_q2
  name_of_var3_for_q2: |
        value_for_var3_q2_line_1
        value_for_var3_q2_line_3
        value_for_var3_q2_line_3
  figure_values:
    name_of_fig_var1_for_q2: value_for_fig_var1_for_q2
```

Notes:

- Each `name_of_varX_for_qY` has to be a variable that the `make_question_page` function accepts
- reference for the functions is at the top of the notebook
- any variables not specified will get the default value as stated inthe documentation
- `figure_values` is a special variable that takes more variables. the nams of the fig variables are the ones for the `normal_curve_slider` function
- the variables can be in any order
- `question_id` must be stated, there is no default value for it
- only the first variable for each question gets a `-`
- `name_of_var3_for_q2` is an example of how to format a long value if you do not leave it on a single line.

Some come from the question and others are for the pate

With shared parameters

To share values across question it can be set up like

```
shared:
  name_of_var1_shared: value_for_var1_shared
  name_of_var2_shared: value_for_var2_shared
  figure_values:
    name_of_fig_var2_shared: value_for_fig_var2_shared
unique:
- question_id: unique_id_for_q1
  name_of_var4_for_q1: value_for_var4_q1
  figure_values:
```

Skip to main content

```
  name_of_var4_for_q2: value_for_var4_q2
  figure_values:
    name_of_fig_var1_for_q2: value_for_fig_var1_for_q2
```

This is equivalent to (but, for many questions, more compact than)

```
- question_id: unique_id_for_q1
  name_of_var4_for_q1: value_for_var4_q1
  name_of_var1_shared: value_for_var1_shared
  name_of_var2_shared: value_for_var2_shared
  figure_values:
    name_of_fig_var2_shared: value_for_fig_var2_shared
    name_of_fig_var1_for_q1: value_for_fig_var1_for_q1
- question_id: unique_id_for_q2
  name_of_var4_for_q2: value_for_var4_q2
  name_of_var1_shared: value_for_var1_shared
  name_of_var2_shared: value_for_var2_shared
  figure_values:
    name_of_fig_var2_shared: value_for_fig_var2_shared
    name_of_fig_var1_for_q2: value_for_fig_var1_for_q2
```

Configuring your study

To configure the study you will need the urls to each follow-up survey. They do not have to be fully configured first though.

Page level Settings

Skip to main content

- `question_id` : string {required} name for the question internally
- `figure_type` : string string name of valid plot type in ssbuilder
- `figure_values` : dictionary parameters to pass to plotting function
- `page_title` : string what to show in the tab title default = ‘Normal Curve Question’,
- `question_text` : string the text of the questions
- `confirm_message` : text prompt for confirmation
- `skip_message` : text prompt for skipping
- `button_text` : string text on button
- `out_html_file` : string name fo the html file, that will be in the url for the participant if not passed will add “.html” to the questionid
- `out_rel_path` : string or file buffer where to write the files.
- `logging_vars` : dictionary dictionary of names for the variable types the specific question requires
- `confirm_var_name` : string {‘confirm’} name for the variable, if not passed will be question_id + ‘confirm’ +question_id
- `var_name_suffix` : boolean {True} if true, add question_id to the passed values for all _var_names. Default is True, can be changed to False if you specify the variable names directly
- `pass_through_vars` : list of strings [‘id’] list of variables to pass through from previous to next
- `next_question` : string question id or url for the qualtrics question
- `pretty_url` : boolean {False} if True make pages like `/IndentiCurve/name/` instead of `/IdentiCurve/name.html`
- `full_html` : boolean {True} generate a full html page or if False, generate only a segment of the page (eg for combining or embedding)

Figure specific Settings

These settings vary by question type and the options are detailed on Question Types

Building Level Options

```
%%bash
ssgeneratehtml --help
```

Skip to main content

```
Usage: ssgeneratehtml [OPTIONS]

  Generate html files from a configuration file

  Parameters ---------- config_file : string or None     file name, if none,
  configureation.yml assumed repo_name : string {None}     name of the repo
  gh_org : string {None}     name of the gh org or user that owns the repo to
  build the URL debug : bool     print debuggin information or not fragment :
  bool     generate a fragment or not all_in_one : bool     merge files to a
  single htmlfile

Options:
  -f, --config-file TEXT
  -p, --out_rel_path TEXT
```

Skip to main content

```
  -r, --repo_name TEXT
```

```
  -o, --gh_org TEXT
```

```
  -d, --debug
```

```
  --fragment
```

```
  -a, --all_in_one
```

```
  --help                  Show this message and exit.
```

Question Types

Question Implementation

Each question type is implemented as a class the class also specifies the HTML/js templates to use for that question type. The constructor documents the logging variables that can be passed. See more on

All questions must have certain parameters:

- `question_text`
- `question_id`

the `question_text` can include markdown formatting which will be rendered with the markdown package

Normal Curve Questions

This question has two normal curves, one moves and one does not.

Skip to main content

- `static_name` : string legend text for static curve
- `static_color` : hex including # hex code for the color to use for static curve, including a # sign as the first character
- `static_mean` : number location of the static curve
- `static_curve_width` : number width of curve, as the scipy.norm scale
- `dynamic_name` : string legend text for dynamic curve
- `dynamic_color` : hex including # hex code for the color to use for dynamic curve, including a # sign as the first character
- `dynamic_starting_mean` : number the location where the slider starts
- `curve_width` : number width of curves
- `num_slider_locs` : integer number of slilder locations
- `min_slider_value` : number the minimum value for the slider
- `max_slider_value` : number the maximum value for the slider
- `overlap_decimals` : integer number of place values to round the % overlap value to for both display and reporting, positive to the right of the decimal, negative for left of decimal (eg -2 rounds to nearest 100)
- `mean_decimals` : integer number of place values to round the mean (position) value to for both display and reporting positive to the right of the decimal, negative for left of decimal (eg -2 rounds to nearest 100)
- `xaxis_title` : string text label for the x axis

Trade Off Questions

this question type trades off between two two extremes over a number of models in the middle

Line Graph

Use `figure_type: TradeoffLine` with the following parameters for use in `figure_values`:

Skip to main content

- `pretty_data_file` : string file name of a tidy (tall) dataset with pretty content. that is any data transformations should occur on the data (eg scaling .7523943 to 75.23943 and expanding column names) column names can still rely on python conventions, before display the `_` will be converted to space
- `slider_column` : string name of column to use for the slider
- `slider_label` : string name to display when labeling the slider postion values (and in hovertext)
- `x_col,y_col` : string name of column to use for the xor y axis
- `trace_value1,trace_value2` : same as the values of `x_col` in the data file first,second value to filter (left, right metric)
- `trace1_hover,` trace2_hover : string noun versions to use in the hovertext
- `y_min,y_max` : numerical minimum and maximum values to fix the plot axies, if none, allow plotly to decide
- `num_digits` : num digits to display color_col: string name of colum to use for the colring of the lines
- `color_hover` : string noun to use for groups
- `disable_zoom` : bool disable the zoom on the generated plot
- `anchor_name` : string name for vertical bar default_selection :int model that is selected when laoding

Bar Graph

Use `figure_type: TradeoffBar` with the following parameters for use in `figure_values`

Skip to main content

- `pretty_data_file` : string file name of a tidy (tall) dataset with pretty content. that is any data transformations should occur on the data (eg scaling .7523943 to 75.23943 and expanding column names) column names can still rely on python conventions, before display the `_` will be converted to space
- `slider_column` : string name of column to use for the slider
- `slider_label` : string name to display when labeling the slider postion values (and in hovertext)
- `x_col,y_col` : string name of column to use for the xor y axis
- `x_value1,x_value2` : same as the values of `x_col` in the data file first,second value to filter (left, right metric)
- `x_value1_hover,` x_value2_hover : string noun versions to use in the hovertext
- `y_min,y_max` : numerical minimum and maximum values to fix the plot axies, if none, allow plotly to decide
- `num_digits` : num digits to display color_col: string name of colum to use for the colring of the bars color_hover: hover text to use for groups created by color
- `disable_zoom` : bool disable the zoom on the generated plot

Coordinating with Qualtrics

> **Important**
>
> This is the base documentation that may need updates

Overview

1. the first qualtrics survey sends data to normal curve only
2. middle ones receive data from and send data to the normal curve
3. the last one receives data only

Qualtrics -> SS

1. Add an embedded data block with the identifier to forward (eg panel ID or Response ID)

Skip to main content

3. embed the response id in the forwarding url:

Template

```
https://statistical-perceptions.github.io/IdentiCurve/<question_out_html_file>.html?id=<
```

Example

```
https://statistical-perceptions.github.io/sample-nobackend/?id=${e://Field/ResponseID}
```

in this case, I only had one question page so there is no question_id set and I used the ResponseID feild.

note:

- if needed, we can pass more than a single unique identifier on, but that requires code changes

SS -> qualtrics

1. set up embedded data as the first block on the workflows tab
2. (if applicable) use piped text to refer to those values in the question text
3. (optional) add a branch after the embedded data to have people skip the survey “if id is Equal to demo”

- getting data from url
-

Qualtrics Help

- Piped text
- sum question
- embedded data

Configuration Details

Skip to main content

Page level

class **ssbuilder.builder.**make_question_page(***question_id, figure_type='NormalCurveSlider', figure_values=None, page_title='Normal Curve Question', question_text='Move the slider', confirm_message='Confirm my answer', skip_message='Prefer not to answer', button_text='Submit', out_html_file=None, out_rel_path=None, logging_vars=None, confirm_var_name=None, var_name_suffix=True, pretty_url=False, pass_through_vars=['id'], out_url=None, next_question_url='https://uri.co1.qualtrics.com/jfe/form/SV_3rU4XfDtiVN8HMW', debug=False, full_html=True***)

generate html file

Parameters:

- **question_id** (*string {required}*) – name for the question internally
- **figure_type** (*string*) – string name of valid plot type in ssbuilder
- **figure_values** (*dictionary*) – parameters to pass to plotting function
- **page_title** (*string*) – what to show in the tab title default = ‘Normal Curve Question’,
- **question_text** (*string*) – the text of the questions
- **confirm_message** (*text*) – prompt for confirmation
- **skip_message** (*text*) – prompt for skipping
- **button_text** (*string*) – text on button
- **out_html_file** (*string*) – name fo the html file, that will be in the url for the participant if not passed will add “.html” to the questionid
- **out_rel_path** (*string or file buffer*) – where to write the files.
- **logging_vars** (*dictionary*) – dictionary of names for the variable types the specific question requires
- **confirm_var_name** (*string {'confirm'}*) – name for the variable, if not passed will be question_id + ‘confirm’ +question_id
- **var_name_suffix** (*boolean {True}*) – if true, add question_id to the passed values for all _var_names. Default is True, can be changed to False if you specify the variable names directly
- **pass_through_vars** (*list of strings ['id']*) – list of variables to pass through from previous to

Skip to main content

- **next_question** (*string*) – question id or url for the qualtrics question
- **pretty_url** (*boolean {False}*) – if True make pages like */IndentiCurve/name/* instead of */IdentiCurve/name.html*
- **full_html** (*boolean {True}*) – generate a full html page or if False, generate only a segment of the page (eg for combining or embedding)

Notes

variables with _var_name + “id” will be passed to qualtrics

Building from config

`ssbuilder.generate_from_configuration`

alias of <Command generate-from-configuration>