



Getting started with surveydown

Meet your instructor!



John Helveston, Ph.D.

Associate Professor, Engineering Management & Systems Engineering

- 2016-2018 Postdoc at [Institute for Sustainable Energy](#), Boston University
- 2016 PhD in Engineering & Public Policy at Carnegie Mellon University
- 2015 MS in Engineering & Public Policy at Carnegie Mellon University
- 2010 BS in Engineering Science & Mechanics at Virginia Tech
- Website: www.jhelvy.com

Meet the dev team!



John Paul Helveston



Pingfan Hu





Bogdan Bunea

Before we start...

1. Be sure to [install everything](#)
2. Download the [examples](#)

Getting started with surveydown

jhelvy.shinyapps.io

 **Welcome to the Ultimate Superhero Survey!** 

Thank you for participating in our research on superhero preferences and powers! This survey will help us understand what makes superheroes appealing to different people.

This survey should take about 3-5 minutes to complete.

Do you have a favorite super hero?*

☐ Yes

☒ No

Next

WYSIWYG Interface

The image shows a web browser window displaying the Google Forms editor. The browser's address bar shows 'docs.google.com'. The form is titled 'Super Hero Survey!' and has a subtitle 'Form description'. The 'Questions' tab is selected in the top navigation bar. A question is being edited: 'Do you have a favorite super hero?'. Below the question, there are three radio button options: 'Yes', 'No', and 'Add option or add "Other"'. A context menu is open over the question, showing various question types: 'Short answer', 'Paragraph', 'Multiple choice' (selected), 'Checkboxes', 'Dropdown', 'File upload', 'Linear scale', 'Rating' (marked 'New'), 'Multiple choice grid', 'Checkbox grid', 'Date', and 'Time'. On the right side of the editor, there is a vertical toolbar with icons for adding new questions, inserting images, text, video, and tables. The top right of the interface includes a 'Publish' button and a user profile icon.

Super Hero Survey!

Form description

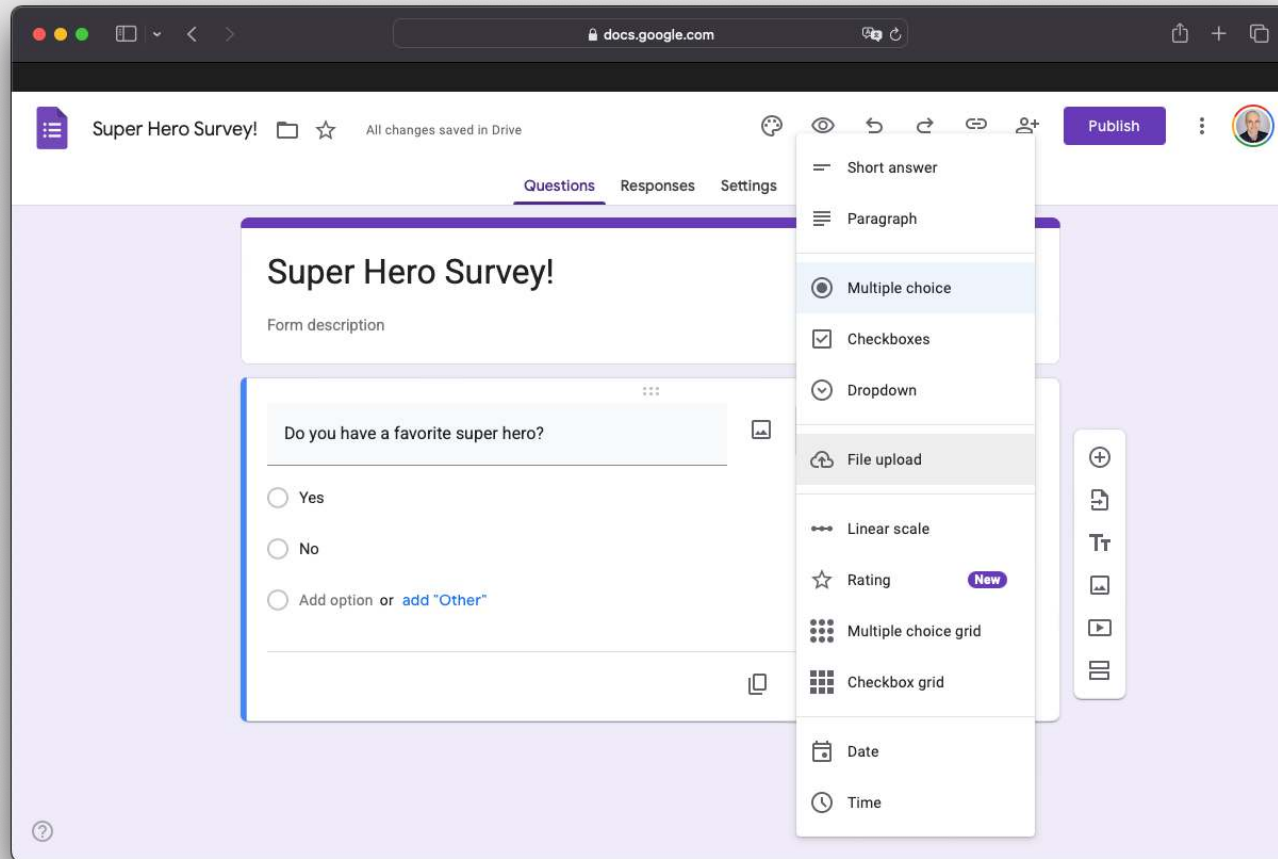
Do you have a favorite super hero?

☐ Yes

☐ No

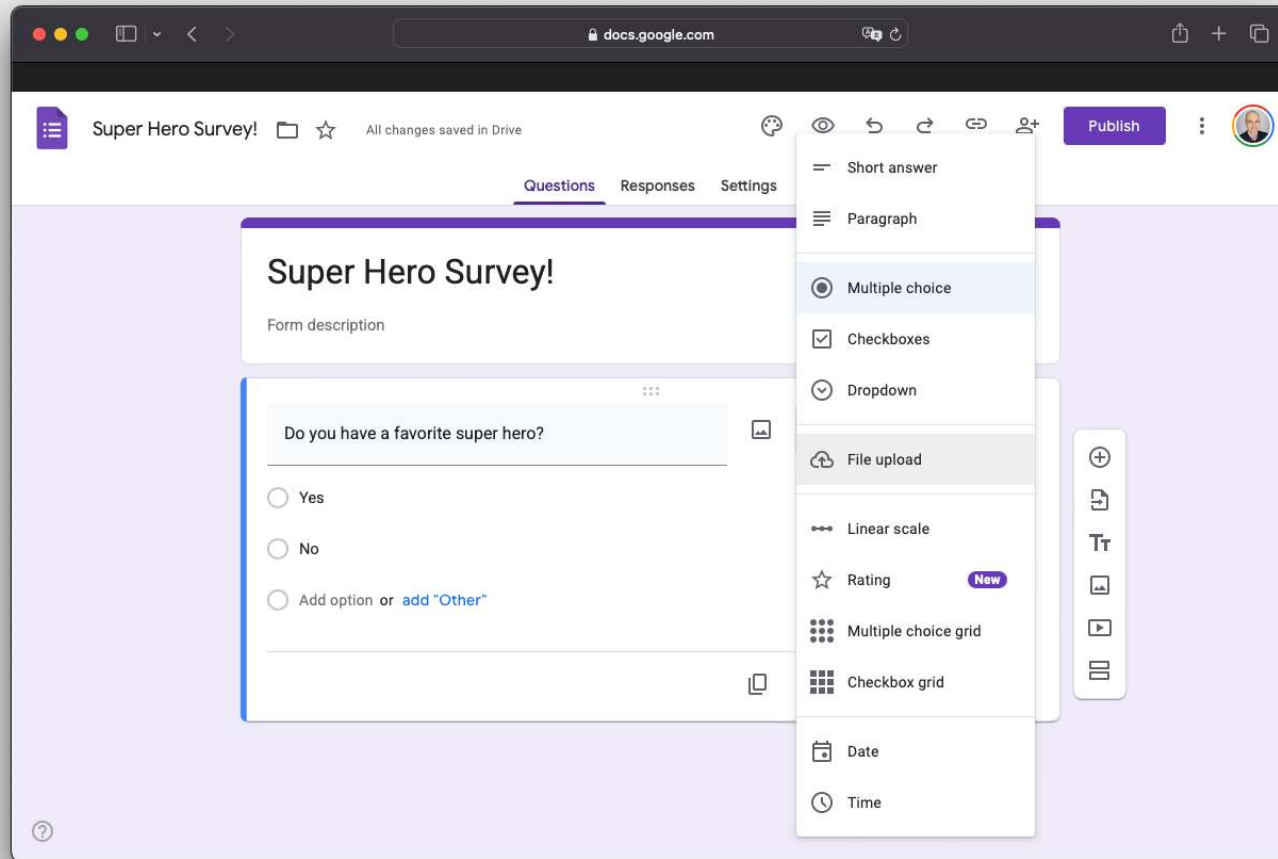
☐ Add option or [add "Other"](#)

- Short answer
- Paragraph
- Multiple choice**
- Checkboxes
- Dropdown
- File upload
- Linear scale
- Rating New
- Multiple choice grid
- Checkbox grid
- Date
- Time



Limitations

✗ Reproducibility



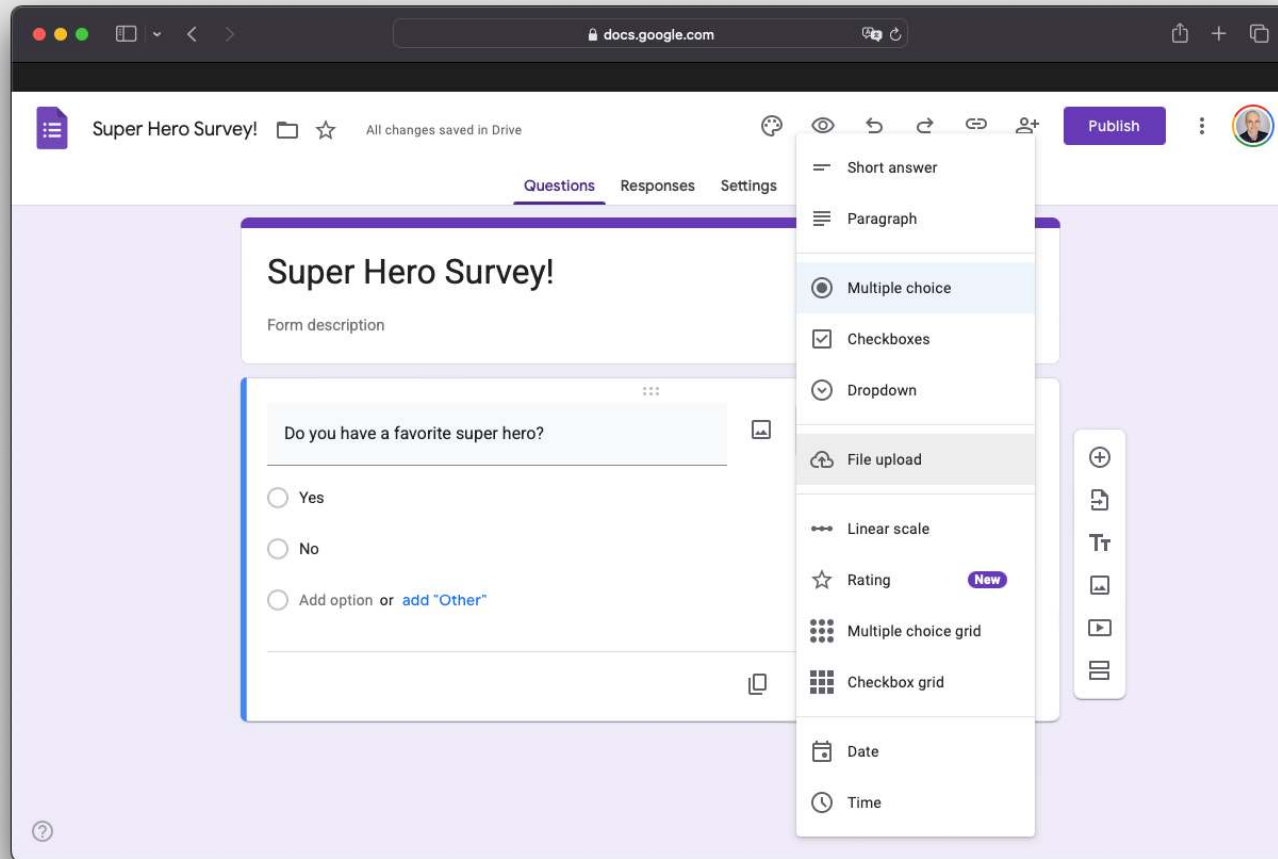
Limitations

✗ Reproducibility

✗ Version control

Untitled 138.docx
Untitled 241.doc
Untitled 138 copy.docx
Untitled 138 copy 2.docx
Untitled 139.docx
Untitled 40 MOM ADDRESS.jpg
Untitled 242.doc
Untitled 243.doc
Untitled 243 IMPORTANT.doc
Untitled 41.jpg



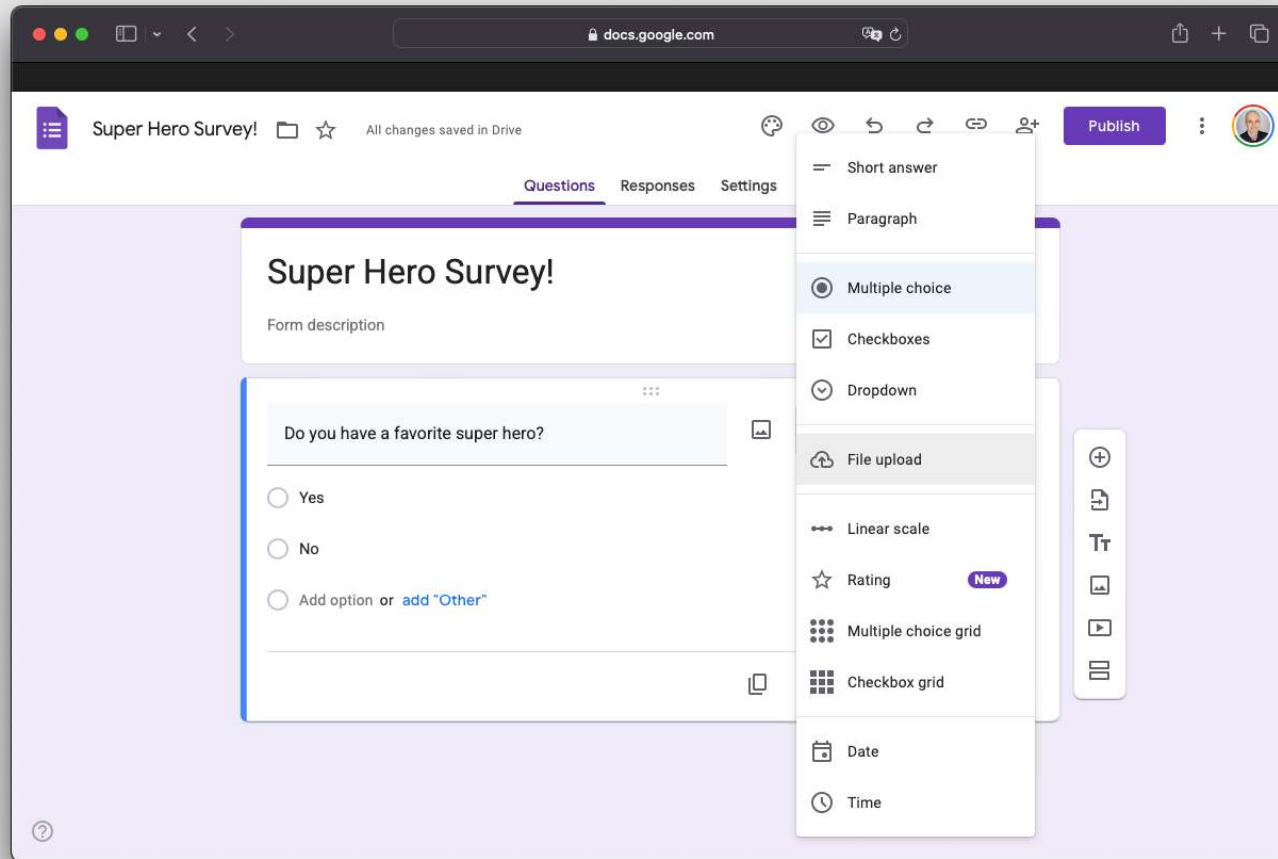


Limitations

✗ Reproducibility

Version control

✗ Limited features



Limitations

✗ Reproducibility

Version control

Limited features

✗ Open source

Why not make surveys from code?

Why not make
surveys from code?

~~Limitations~~
Features

- ✓ Reproducibility
 - Version control
 - Lots of features
- ✓ Open source

Introducing surveydown!



Getting started with surveydown

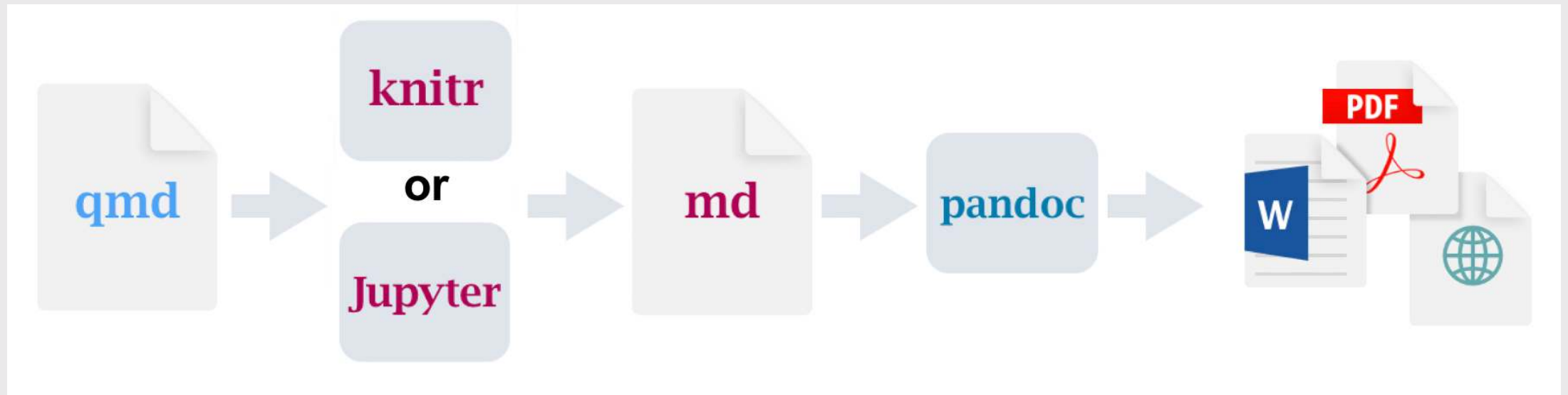
1. What is surveydown?
2. Adding survey content

R package that renders Quarto files into surveys



What is Quarto?

Quarto is a publishing system that blends
markdown-formatted text & code



example.qmd

```
---
format: html
title: "HTML Page with R Code"
---

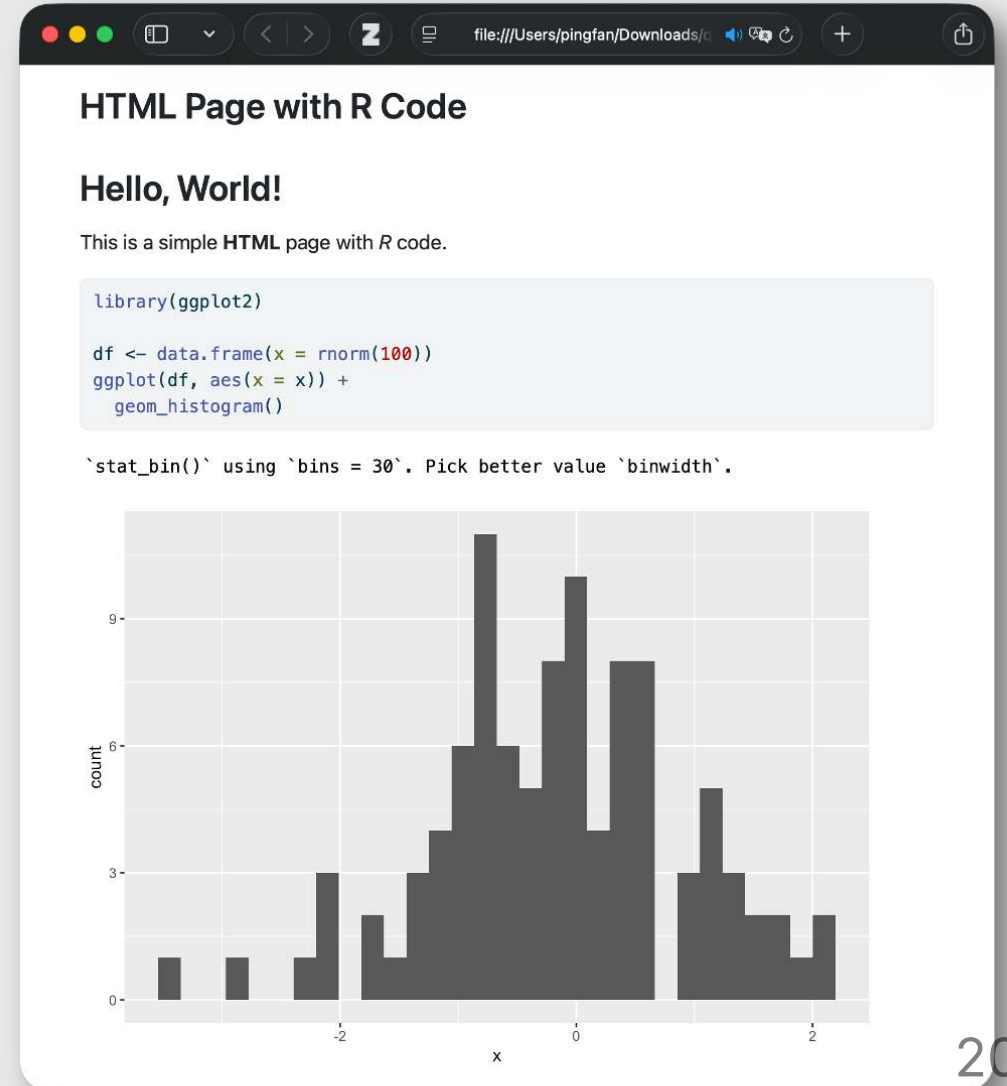
# Hello, World!

This is a simple HTML page with R code.

```{r}
library(ggplot2)

df <- data.frame(x = rnorm(100))
ggplot(df, aes(x = x)) +
 geom_histogram()
```
```

Rendered output



survey.qmd

```
---
format: html
echo: false
warning: false
---

```{r}
library(surveydown)
```

--- welcome

# Welcome to our survey!

```{r}
sd_question(
 type = "mc",
 id = "penguins",
 label = "What's your favorite penguin?",
 option = c(
 "Adélie" = "adelie",
 "Chinstrap" = "chinstrap",
 "Gentoo" = "gentoo"
)
)

sd_next()
```

--- end

This is the last page of the survey.

```{r}
sd_close()
```
```

YAML Header

Load `surveydown` package

```
---  
format: html  
echo: false  
warning: false  
---
```

```
```${r}  
library(surveydown)
```
```

--- welcome

Welcome to our survey!

```
```${r}  
sd_question(
 type = "mc",
 id = "penguins",
 label = "What's your favorite penguin?",
 option = c(
 "Adélie" = "adelie",
 "Chinstrap" = "chinstrap",
 "Gentoo" = "gentoo"
)
)
```

```
sd_next()
```
```

--- end

This is the last page of the survey.

```
```${r}  
sd_close()
```

```

format: html
echo: false
warning: false

```${r}  
library(surveydown)  
```
```

```
--- welcome
```

```
Welcome to our survey!
```

```
```${r}  
sd_question(  
  type = "mc",  
  id = "penguins",  
  label = "What's your favorite penguin?",  
  option = c(  
    "Adélie" = "adelie",  
    "Chinstrap" = "chinstrap",  
    "Gentoo" = "gentoo"  
  )  
)  
  
sd_next()  
```
```

```
--- end
```

```
This is the last page of the survey.
```

```
```${r}  
sd_close()
```

Define page breaks like this:

```
--- page_id
```

```
Page content
```

```

---
format: html
echo: false
warning: false
---

```{r}
library(surveydown)
```

--- welcome

# Welcome to our survey!

```{r}
sd_question(
 type = "mc",
 id = "penguins",
 label = "What's your favorite penguin?",
 option = c(
 "Adélie" = "adelie",
 "Chinstrap" = "chinstrap",
 "Gentoo" = "gentoo"
)
)

sd_next()
```

--- end

This is the last page of the survey.

```{r}
sd_close()

```

Use markdown for page content

Use `sd_question()` in code chunk for survey questions

Use `sd_next()` for next button



```

format: html
echo: false
warning: false

```{r}
library(surveydown)
```

--- welcome

Welcome to our survey!

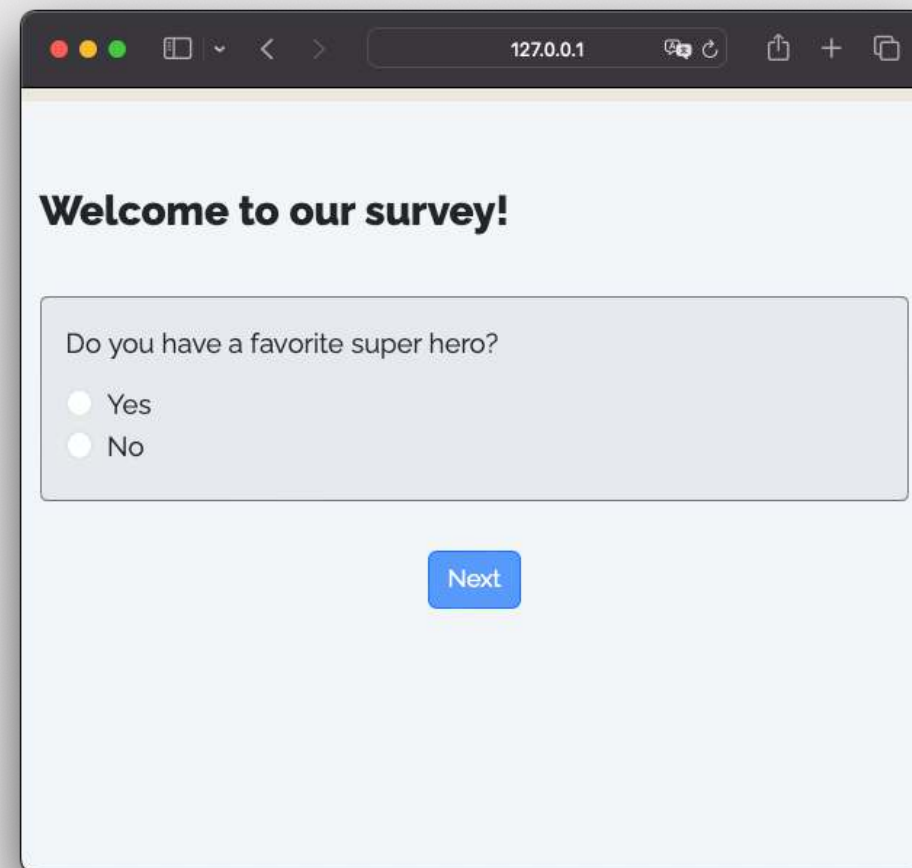
```{r}
sd_question(
  type = "mc",
  id = "penguins",
  label = "What's your favorite penguin?",
  option = c(
    "Adélie" = "adelie",
    "Chinstrap" = "chinstrap",
    "Gentoo" = "gentoo"
  )
)

sd_next()
```

--- end

This is the last page of the survey.

```{r}
sd_close()
```
```



127.0.0.1

## Welcome to our survey!

Do you have a favorite super hero?

☐ Yes

☐ No

Next

# Use Shiny to drive the web app



# Survey Components

`survey.qmd`

A **Quarto doc** defining the main survey content (pages, text, images, questions, etc).

`app.R`

An **R script** defining the survey Shiny app. It sets up the database and server, and launches the survey.

# app.R

```
library(surveydown)

Connects to database
db <- sd_db_connect()

Main UI
ui <- sd_ui()

server <- function(input, output, session) {

 # Main server
 sd_server(db)

}

shiny::shinyApp(
 ui = ui,
 server = server
)
```

```
library(surveydown)
```

```
Connects to database
```

```
db <- sd_db_connect()
```

```
Main UI
```

```
ui <- sd_ui()
```

```
server <- function(input, output, session) {
```

```
 # Main server
```

```
 sd_server(db)
```

```
}
```

```
shiny::shinyApp(
 ui = ui,
```

```
 server = server
```

```
)
```

Load package

Connect to database

Make standard UI

Run surveydown server

05:00

# Your turn - start with a template

**Step 1:** In R console, run:

```
surveydown::sd_create_survey()
```

*Optional:* Specify a **path** and / or a **template**:

```
surveydown::sd_create_survey(
 template = "question_types",
 path = "path/to/folder"
)
```

**Step 2:** In your IDE, open the **app.R** file, click "Run App", or in R console run:

```
shiny::runApp('app.R')
```

# *Getting started with surveydown*

1. What is surveydown?
2. Adding survey content

# YAML Header

The YAML header is at the top of the `.qmd` file:

```

format: html
echo: false
warning: false

```



# Change the survey theme

Pick a different [bootswatch theme](#) with the [theme](#) key:

```

format: html
echo: false
warning: false
theme: united

```

Make a custom theme with a [custom.scss](#) file

```

format: html
echo: false
warning: false
theme: [united, custom.scss]

```

# Progress bar

You can modify the survey progress bar with the `barcolor` and `barposition` keys:

Change to any color with `barcolor`:

```

format: html
echo: false
warning: false
barcolor: #768692

```

Change position: `top`, `bottom`, or `none`

```

format: html
echo: false
warning: false
barposition: bottom

```

# Inserting pages

Insert new pages with the shorthand syntax using three dashes `---` followed by the page ID:

```
--- page1
```

```
Page 1 content here
```

```
--- page2
```

```
Page 2 content here
```

# Inserting next buttons

Insert buttons with the `sd_next()` function to go to the next page:

```
```{r}  
sd_next()  
```
```

Specify a specific page to go to:

```
```{r}  
sd_next('next_page_id')  
```
```

# Inserting questions

Insert questions using the `sd_question()` function, like this:

## Code

```
sd_question(
 type = 'mc',
 id = 'fruit',
 label = "1. Do you like fruit?",
 option = c(
 'Yes!' = 'yes',
 'Kind of' = 'kind_of',
 'No :(' = 'no'
)
)
```

Format is "Label = value"

## Output

1. Do you like fruit?

- ☐ Yes!
  - ☐ Kind of
  - ☐ No :(
- \*

## Data

yes, kind\_of, or no

# surveydown supports lots of question types

Some common types you may want to use:

| Type                     | Description                                 |
|--------------------------|---------------------------------------------|
| <code>mc</code>          | Multiple choice question (single choice)    |
| <code>mc_multiple</code> | Multiple choice question (multiple choices) |
| <code>mc_buttons</code>  | Multiple choice question (large buttons)    |
| <code>select</code>      | Drop down menu (choose one)                 |
| <code>text</code>        | Open text, single row                       |
| <code>textarea</code>    | Open text, block                            |

# Embed images with html

I recommend just writing html code, like this

```
<center>

</center>
```



# Embed images with markdown

```
{fig-align="center" width="250"}
```





Try out the surveydown **studio app**

```
sdstudio::launch()
```

# Your turn

- Open and edit the `survey.qmd` file that you previously made.
- Pick a topic for your survey (a food, an animal, a sports team...whatever).
- Draft a survey about that topic. Include the following:
  - Page 1: A welcome message in large font ("Welcome to a survey about [topic]") and an image about the topic (find an image somewhere).
  - Pages 2 to N: Add different questions about the topic using several different question types.
- Try out different themes to change the look and feel.

OR:

Open and play with the examples:

- `part-1-question-types`
- `part-1-question-yml`