



(scan me for the slides)

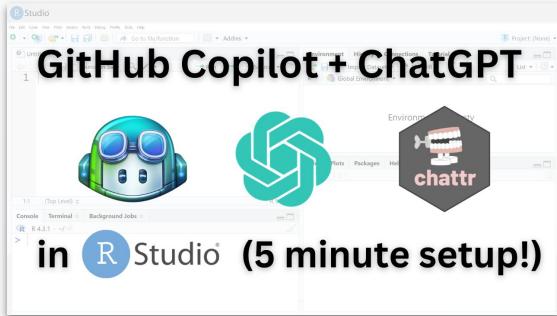
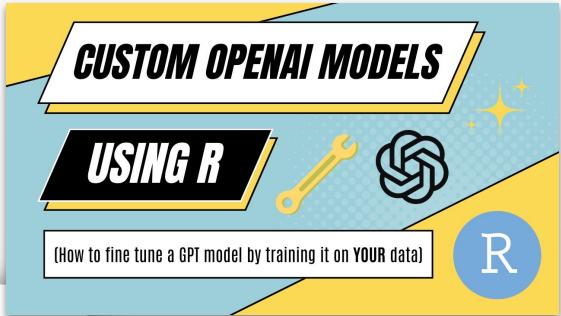
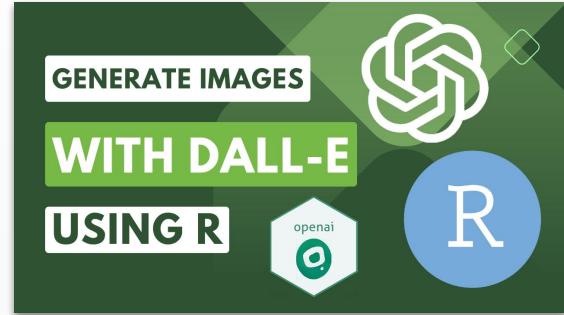
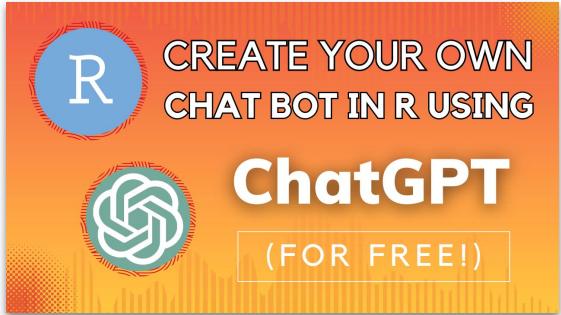
# Practical Tips for Using Generative AI In Data Science Workflows

---

posit::conf(2024)  
Melissa Van Bussel

*Senior Data Analyst, Statistics Canada*

Have any questions? Go to sli.do and use #posit-keynote





I was GenAI-Fearful.

I became a GenAI-Fanatic.

I ended up becoming  
**GenAI-Fatigued.**

**“Upskill now or be left behind!”**

“The **only** course you’ll need  
to become an **AI Engineer!**”

**“Double** your salary **overnight**  
with this one simple trick!”

“The **Age** of **Generative AI...**”

I'm still (mostly) a  
GenAI-Fanatic.

**1. GenAI-Fanatic**

**2. GenAI-Fatigued**

**3. GenAI-Fearful**

**1. GenAI-Fanatic**

**2. GenAI-Fatigued**

**3. GenAI-Fearful (or GenAI-Hesitant)**

**1. GenAI-Fanatic**

**2. GenAI-Fatigued**

**3. GenAI-Fearful**

# **1. GenAI-Fanatic**

# **2. GenAI-Fatigued**

# **3. GenAI-Fearful**

The latest and greatest, and why you might care as someone who works with data

1. GenAI-Fanatic
2. GenAI-Fatigued
3. GenAI-Fearful

Creative use cases  
to hopefully (?) get  
you excited about  
Gen-AI again

# **1. GenAI-Fanatic**

# **2. GenAI-Fatigued**

# **3. GenAI-Fearful**

Best practices for  
working with  
Gen-AI responsibly

# For the GenAI-Fanatics





GO SENS GO

Bell

GO SENS GO

GO SENS GO

GO SENS

FACES do Mitel SPARTACUS CHRISTIE TIRE Mobil

pizza pizza

do Mitel

Ottawa

canadian

airline

ottawa

canadian

airline

ottawa

canadian

airline

ottawa

canadian

airline

ESTON

HARDWARE

LIVERNATION.com

Shuttle

Shuttle

YOKO



loveisblindnetflix • [Follow](#)

loveisblindnetflix New season, new cast, new connections! Tag who you're watching the premiere with on February 14th!

29w



January 17



Add a comment...

[Post](#)



Reels ▾



LOVE IS  
❤️ BLIND



Likes



615

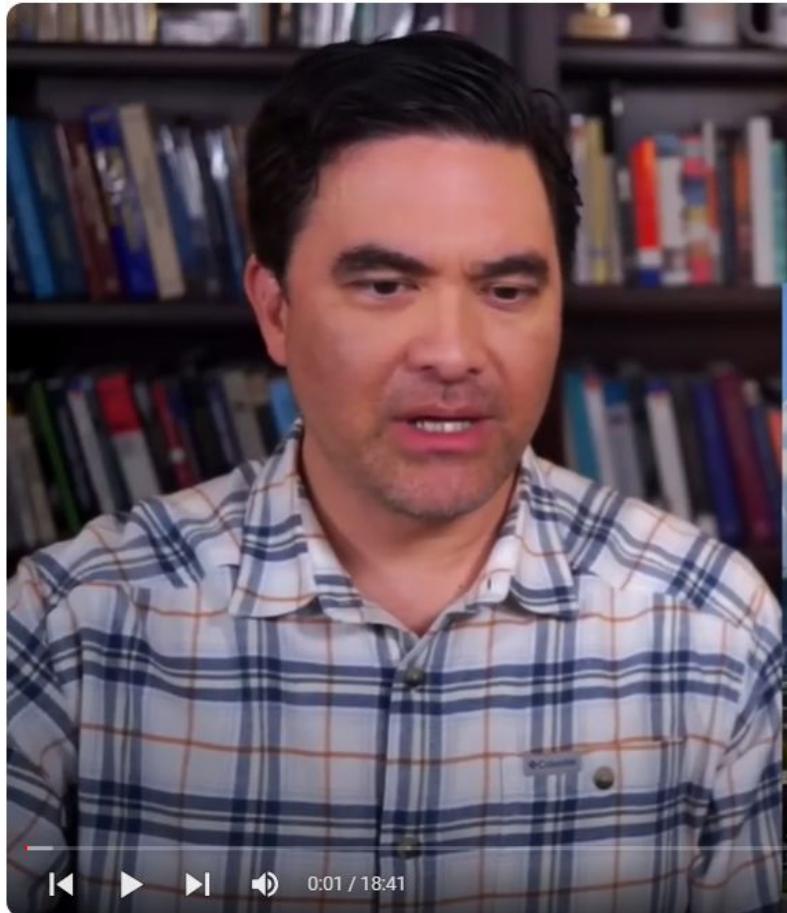


64.1K

CUTEST  
COUPLES

# Psychology in Seattle

LOVE  
IS  
BLIND



0:01 / 18:41



Love Is Blind - Season 4 - #49 - (Codependent) - Therapist Reacts



Psychology In Seattle

366K subscribers

Subscribe



1K



Share

Download

Thanks

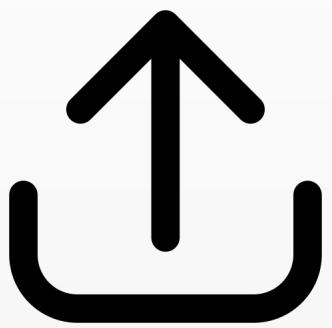


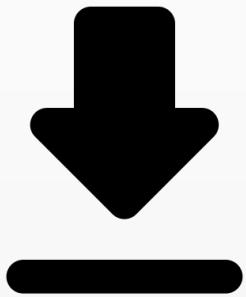
**Being a fan == always wanting more**



OpenAI

# **Text, vision, and audio**

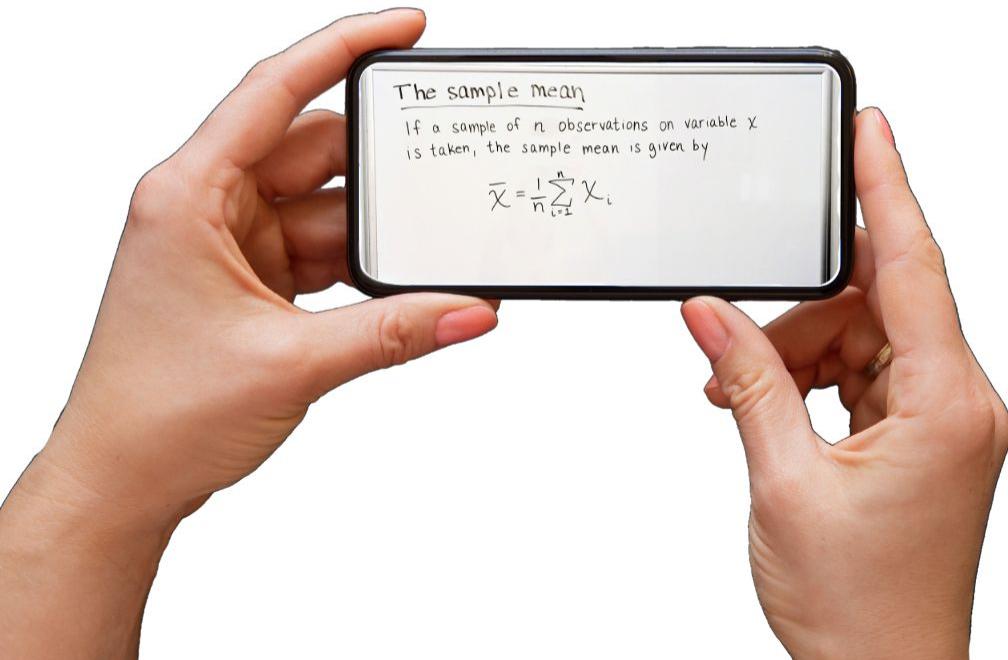






## Step 2: Upload in chat

### Step 1: Take a photo



## The sample mean

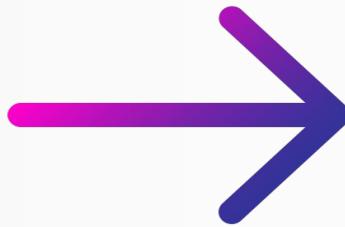
If a sample of  $n$  observations on variable  $X$  is taken, the sample mean is given by

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

### The sample mean

If a sample of  $n$  observations on variable  $X$  is taken, the sample mean is given by

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$



The sample mean

If  
is

Convert the text in the image into a Quarto document, using `format: html`, and keep the same text formatting (bold, italic, underline, etc.)

---

format: html

---

# The sample mean

If a sample of  $n$  observations on variable  $X$  is taken, the sample mean is given by

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

# The sample mean

If a sample of  $n$  observations on variable  $X$  is taken, the sample mean is given by

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

### Problem 1

The number of chocolate bars sold by a store was recorded each week for 10 weeks:

47, 48, 41, 47, 29, 40, 42, 41, 40, 46

Find the sample mean.

### Solution 1

The sample mean is given by:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{x} = \frac{1}{10} (47 + 48 + 41 + 47 + 29 + 40 + 42 + 41 + 40 + 46)$$

$$\bar{x} = \frac{1}{10} (421)$$

$$\bar{x} = \frac{421}{10}$$

$$\bar{x} = 42.1$$

∴ The sample mean is 42.1

## Problem 1

Convert the text in the image into a Quarto document, using format: `html`, and use styling to make sure that the font styles (e.g., bold, italic) and colours match the image, where possible. Align the math around the equals signs.

$\therefore$  The sample mean is 42.1.

## Problem 1

The number of chocolate bars sold by a store was recorded each week for 10 weeks:

47, 48, 41, 47, 29,  
40, 42, 41, 40, 46

Find the sample mean.

## Solution 1

The sample mean is given by:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{x} = \frac{1}{10} (47 + 48 + 41 + 47 + 29 + 40 + 42 + 41 + 40 + 46)$$

$$\bar{x} = \frac{1}{10} (421)$$

$$\bar{x} = \frac{421}{10}$$

$$\bar{x} = 42.1$$

Therefore, the sample mean is **42.1**.

Job Role	Industry	Annual Salary
Consultant	Healthcare	73900
Engineer	Technology	85800
Designer	Manufacturing	94400
Consultant	Finance	73500
Engineer	Retail	60100
Manager	Finance	88700
Analyst	Finance	74100
Analyst	Retail	104200
Analyst	Healthcare	28200
Consultant	Manufacturing	46500
Manager	Technology	88400
Consultant	Education	74200
Designer	Technology	93900
Consultant	Education	79500
Manager	Technology	92700
Manager	Retail	79700
Designer	Manufacturing	55300
Manager	Education	76000
Manager	Healthcare	54500
Analyst	Technology	83000
Designer	Technology	115900
Salesperson	Education	83100
Salesperson	Healthcare	77600
Manager	Healthcare	70400
Salesperson	Retail	103600

Job Role	Industry	Annual Salary
Consultant	Healthcare	73900
Engineer	Technology	85800
Designer	Manufacturing	68500
Salesperson	Retail	103600

Convert the table in  
the image into a  
downloadable CSV.

Job Role	Industry	Annual Salary
Salesperson	Healthcare	77600
Manager	Healthcare	70400
Salesperson	Retail	103600



I have converted the table into a CSV file. You can download it using the link below:

Download [job\\_salaries.csv](#)



	Job Role	Industry	Annual Salary
1	Consultant	Healthcare	73900
2	Engineer	Technology	85800
3	Designer	Manufacturing	94400
4	Consultant	Finance	73500
5	Engineer	Retail	60100
6	Manager	Finance	88700

Here is a preview of the job salaries dataset. Let me know if you need further assistance.



“Annual Salaries” column

X



Please format this column as a dollar value.

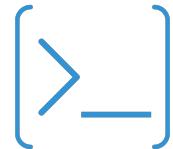


Simulate additional observations so that there are 1000 rows. Create two additional variables, "Years of Experience" and "Education level". Ensure that the generated information and salaries are realistic.

Here is a preview of the job salaries dataset.

View analysis

Let me know if you need further assistance.



python



Copy code

```
# Define ranges for years of experience and their corresponding salary ranges
experience_salary_ranges = {
    (1, 5): (30000, 50000),
    (6, 10): (50001, 70000),
    (11, 15): (70001, 90000),
    (16, 20): (90001, 110000),
    (21, 25): (110001, 130000),
    (26, 30): (130001, 150000),
    (31, 35): (150001, 170000),
    (36, 40): (170001, 190000)
}

# Define education level salary adjustments
education_salary_adjustments = {
    'High School': (0.9, 1.0),
    'Bachelor': (1.0, 1.1),
    'Master': (1.1, 1.2),
    'PhD': (1.2, 1.3)
}
...
...
```





Connect to Google Drive



Connect to Microsoft OneDrive



Upload from computer

Provide a short overview of the dataset, including the data types for each variable.



**Total Rows:** 1000

#### **Columns and Data Types:**

1. **Job Role** (*string*): The role of the employee
2. **Industry** (*string*): The industry in which the employee works
3. **Annual Salary** (*float*): The annual salary of the employee
4. **Years of Experience** (*int*): The number of years the employee has been working, ranging from 1 to 40 years
5. **Education level** (*string*): The highest level of education attained by the employee

Provide summary statistics.



Here are the summary statistics for each column in the dataset:

### **Summary Statistics:**

#### **Annual Salary:**

- **Count:** 1000
- **Mean:** ~\$100,000
- **Standard Deviation:** ~\$34,000
- **Min:** \$30,000
- **25th Percentile:** \$67,500
- **Median (50th Percentile):** \$100,000



Analyze the dataset and extract interesting facts. Use data storytelling techniques to weave together a short (75 word) paragraph that pulls together key insights.



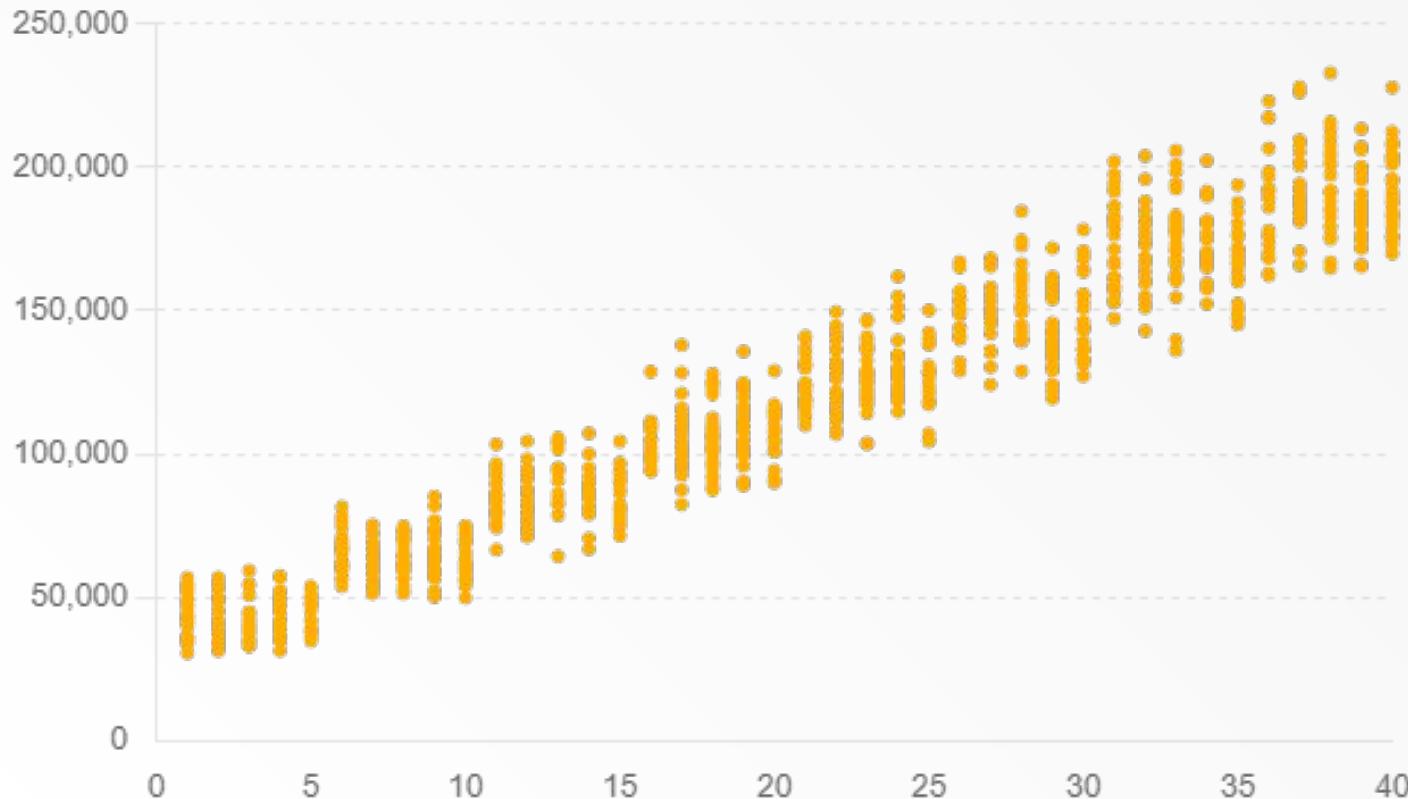


The dataset showcases a skilled and seasoned workforce with an **average of 20.5 years of experience**, primarily in technology and consultancy roles. The **technology sector leads in compensation**, reflecting its high demand, with an average salary of \$100,000. **Most employees hold Bachelor's degrees**, underscoring the importance of higher education. Notably, **"Consultant" is the most prevalent job role**, highlighting the industry's reliance on specialized expertise. This diverse workforce exemplifies the intersection of education, experience, and high-paying tech jobs.

# Scatterplot of Years of Experience Vs Annual Salary



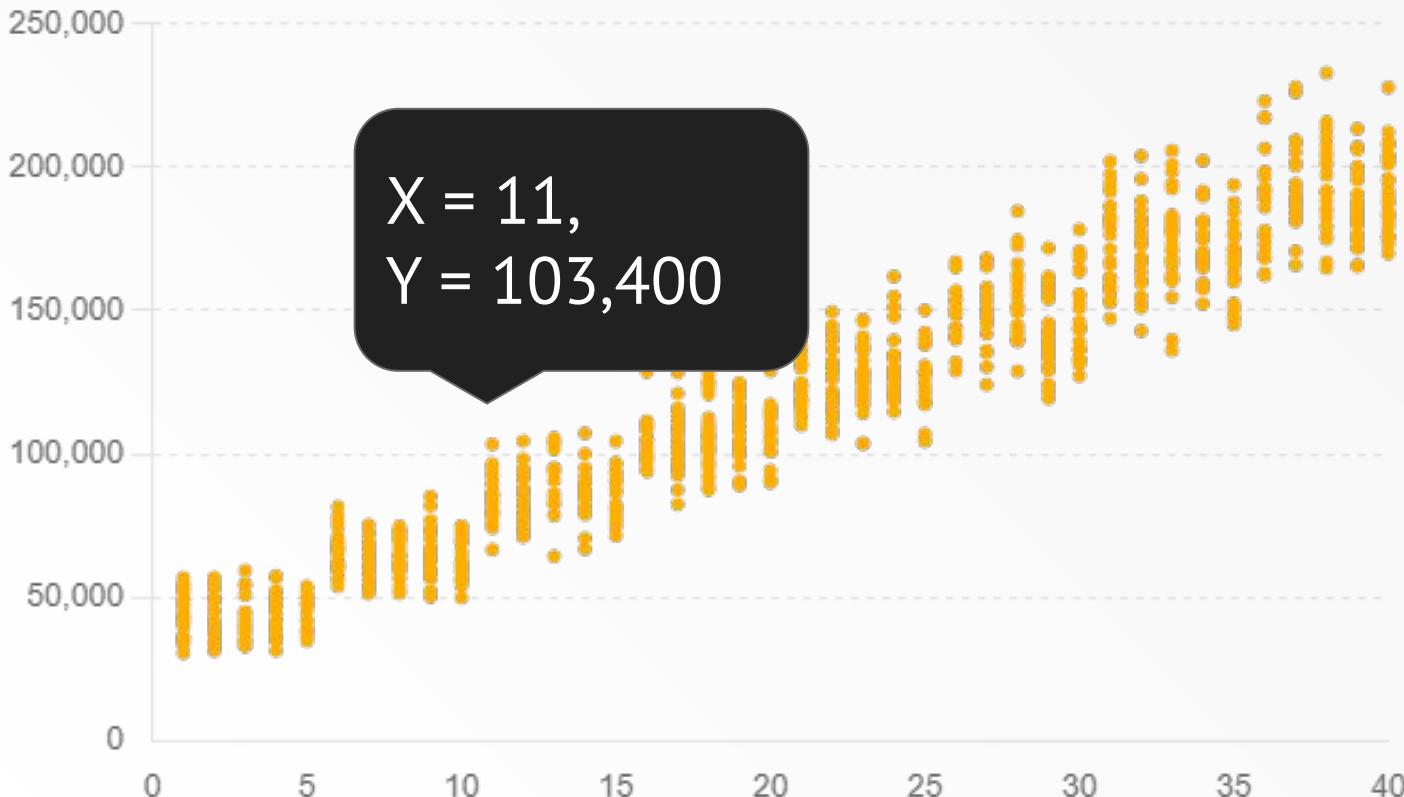
Y Annual Salary by X Years of Experience



# Scatterplot of Years of Experience Vs Annual Salary



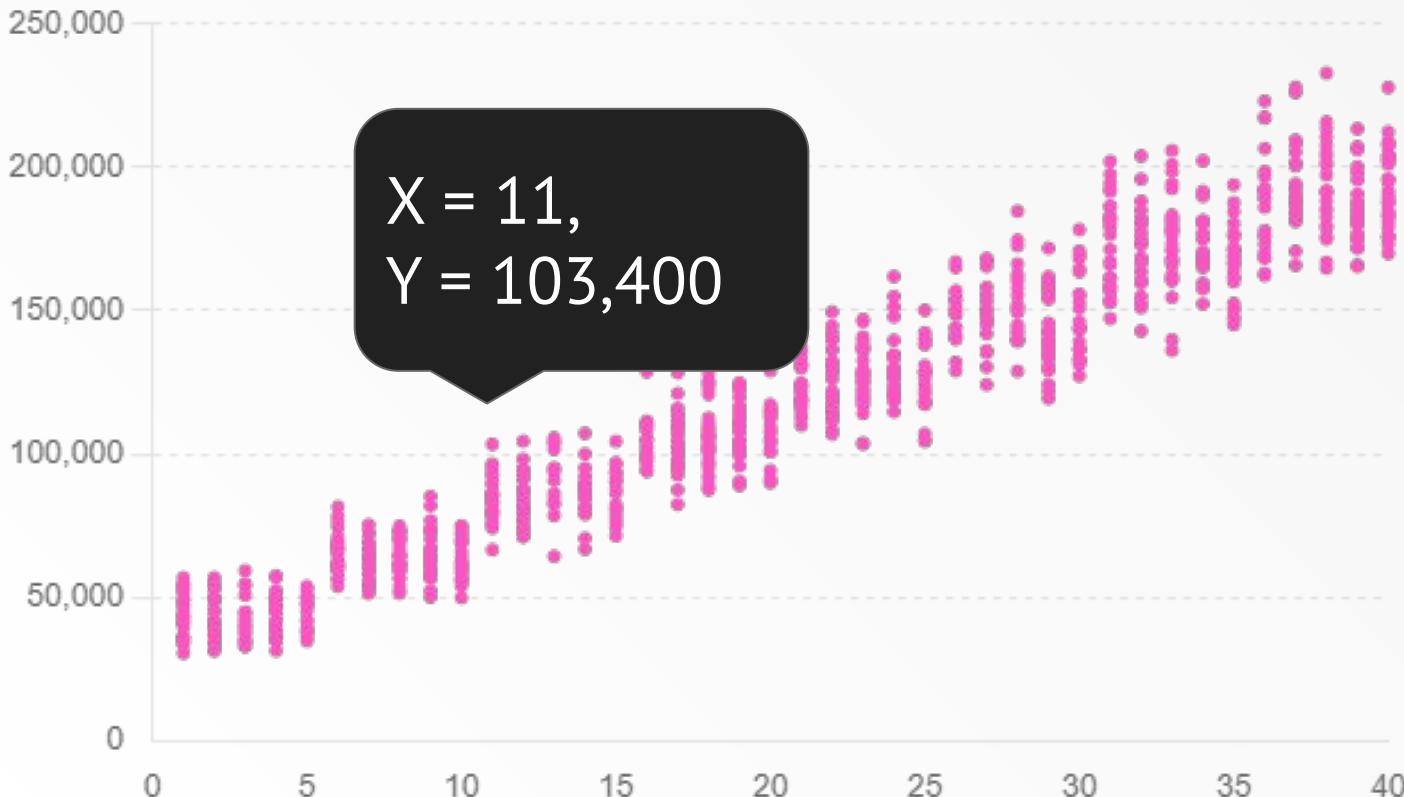
Y Annual Salary by X Years of Experience



# Scatterplot of Years of Experience Vs Annual Salary



Y Annual Salary by X Years of Experience

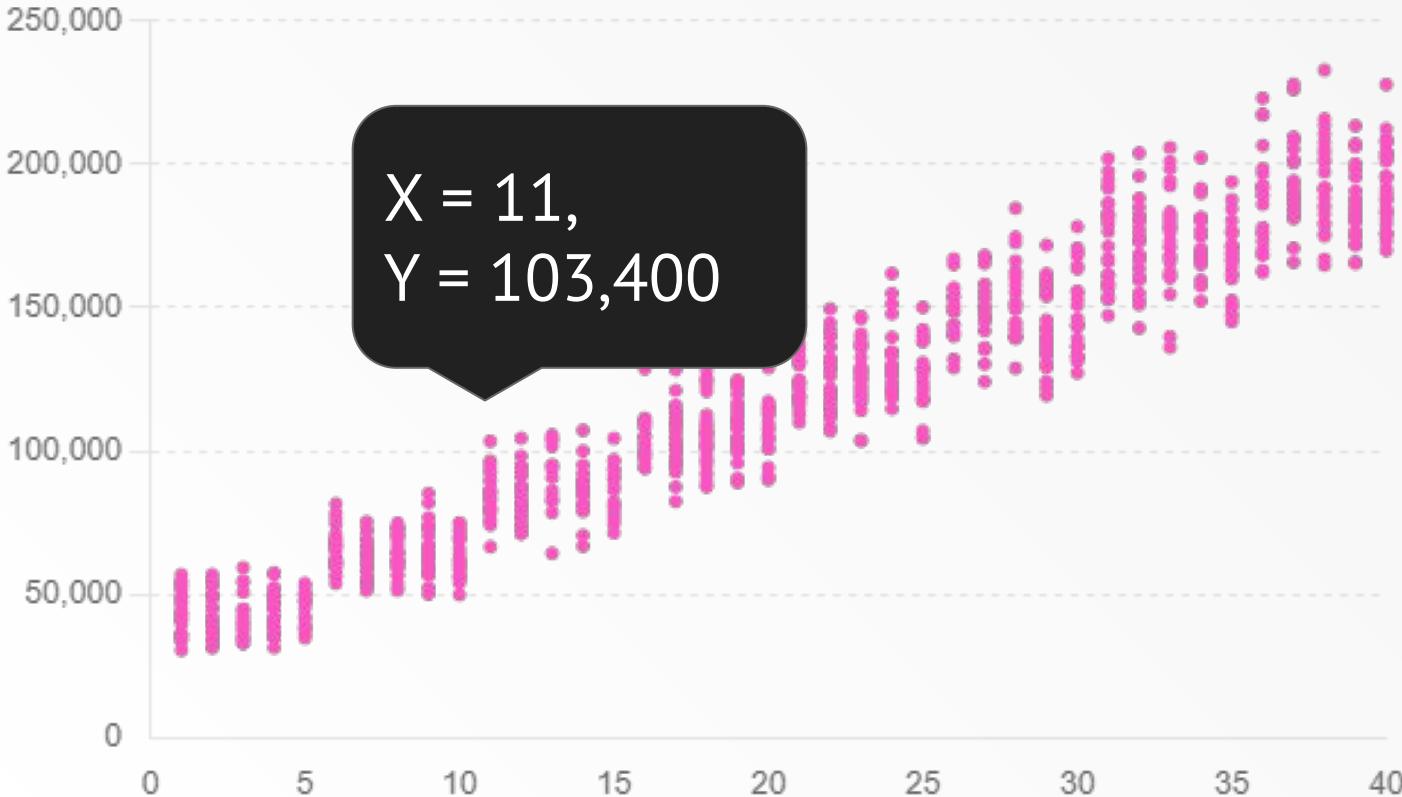


# Scatterplot of Years of Experience Vs Annual Salary

Y Annual Salary by X Years of Experience

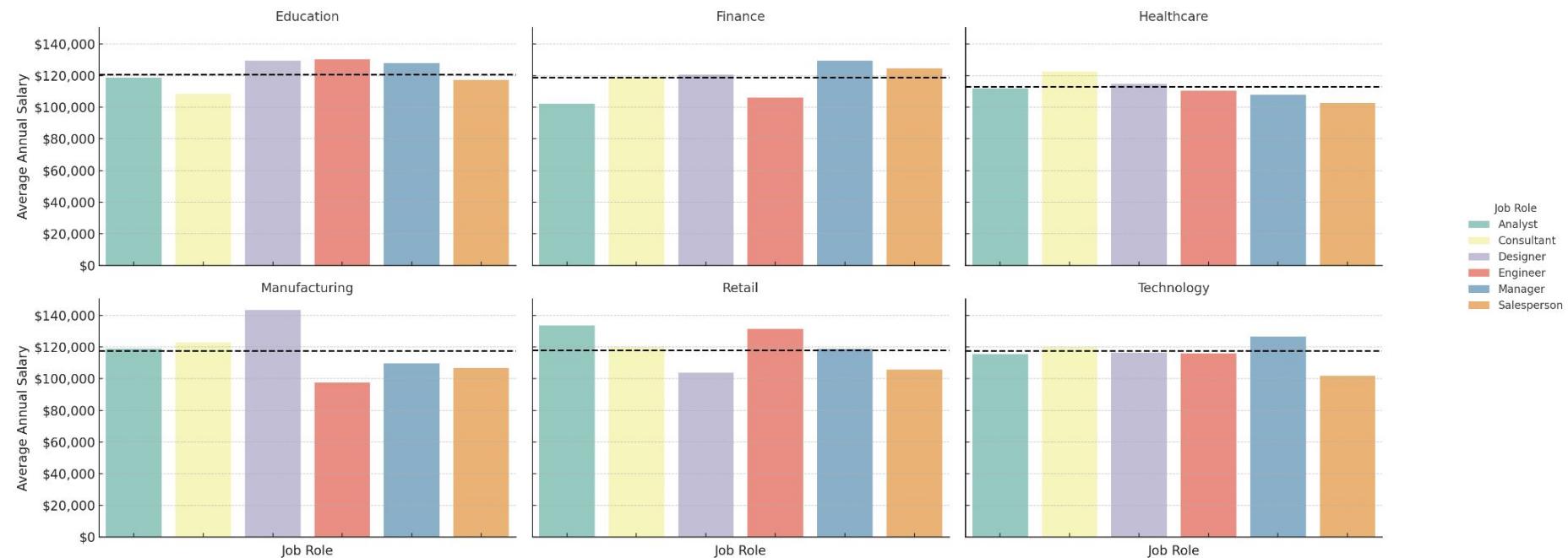


X = 11,  
Y = 103,400

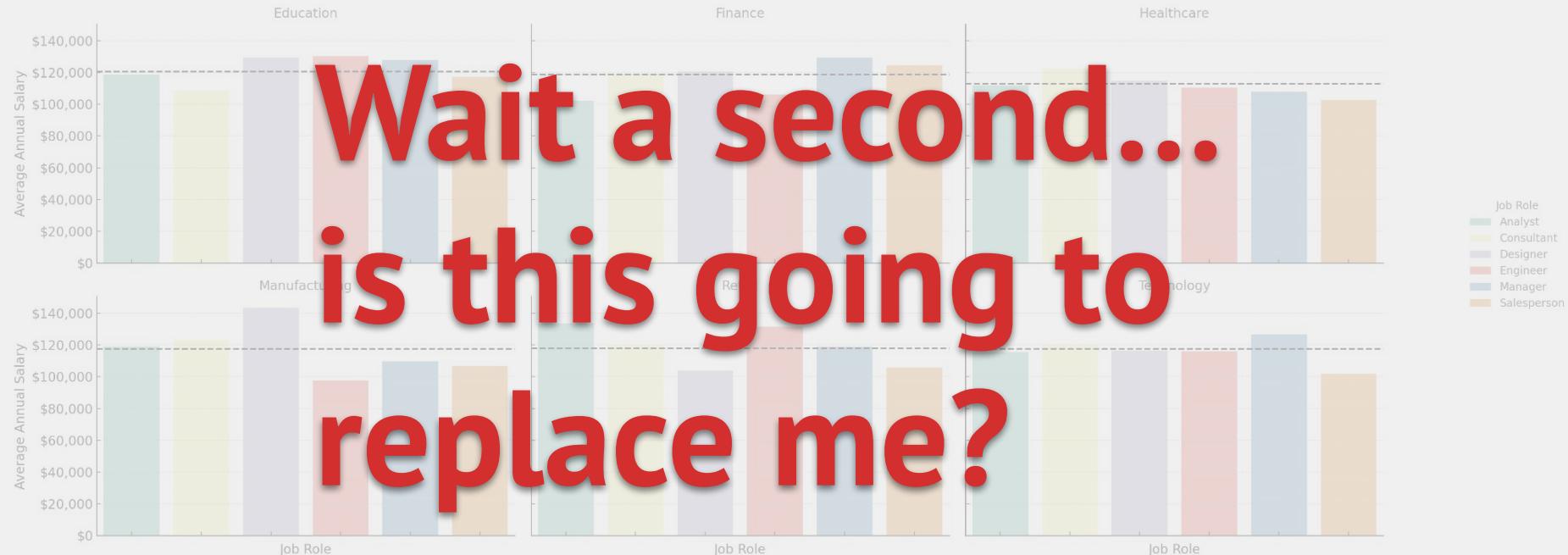


Download  
as image

Average Annual Salary by Job Role and Industry



Average Annual Salary by Job Role and Industry

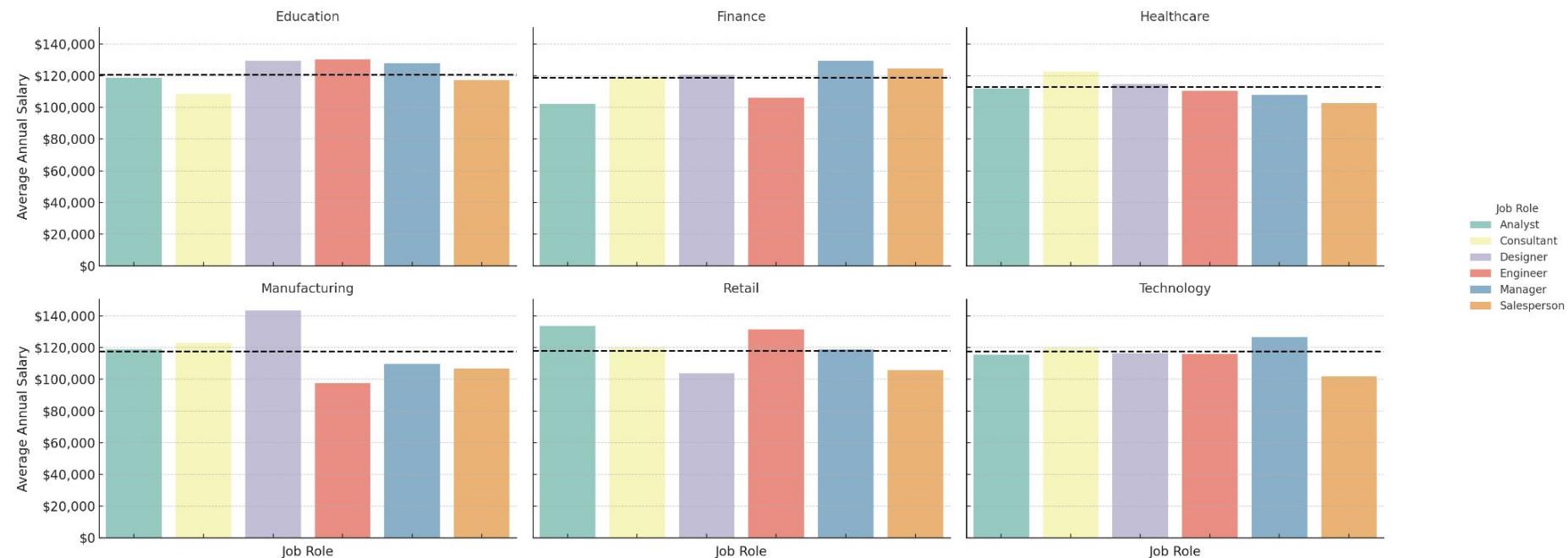


**Writing effective prompts**

**Your existing expertise**



Average Annual Salary by Job Role and Industry



**“Upskill now or be left behind!”**

“The **only** course you’ll need  
to become an **AI Engineer!**”

**“Double** your salary **overnight**  
with this one simple trick!”

“The **Age** of **Generative AI...**”

## Natural language description:

Create a data visualization using the salaries dataset.

## Code equivalent\*:

```
salaries |>  
  ggplot()
```

## Grammar of graphics:

Data

\*=assuming some extra summary variables were added to salaries

## Natural language description:

Put **average salary** on the **y-axis**,  
and **job role** on the **x-axis**. Use a  
different **colour** for each **job role**.

## Code equivalent:

```
salaries |>  
  ggplot(aes(x = job,  
             y = avg_salary,  
             fill = job))
```

## Grammar of graphics:



Aesthetics

Natural language description:

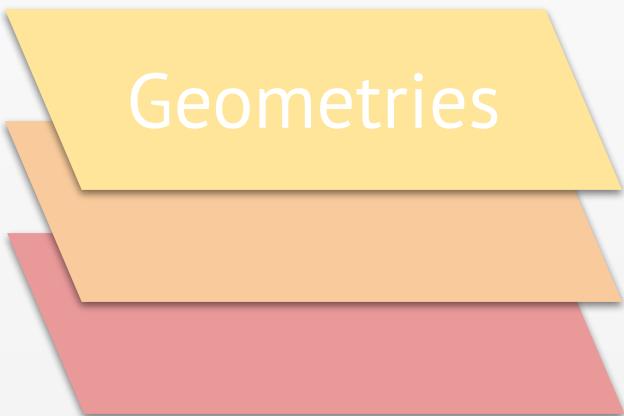
Create a **bar chart**...

---

Code equivalent:

```
geom_bar(  
  stat = "identity",  
  position = "dodge"  
)
```

Grammar of graphics:



Geometries

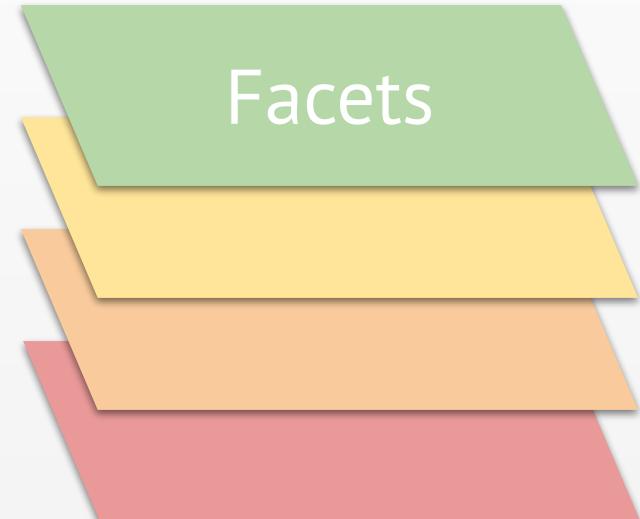
## Natural language description:

...that's **faceted** by the  
**industry** variable

## Code equivalent:

```
facet_wrap(~industry)
```

## Grammar of graphics:



Facets

## Natural language description:

Add a **horizontal black line** showing the **average salary** for each industry.

## Code equivalent:

```
geom_hline(  
  industry_avg_salary,  
  color = "black",  
  linetype = "dashed"  
)
```

## Grammar of graphics:

Statistics

## Natural language description:

Format the **y-axis** as **dollar amounts**. Use a **modern colour palette**...

## Code equivalent:

```
scale_y_continuous(  
    labels = scales::dollar_format()  
) +  
scale_fill_brewer(palette = "Set3")
```

## Grammar of graphics:

Coordinates



## Natural language description:

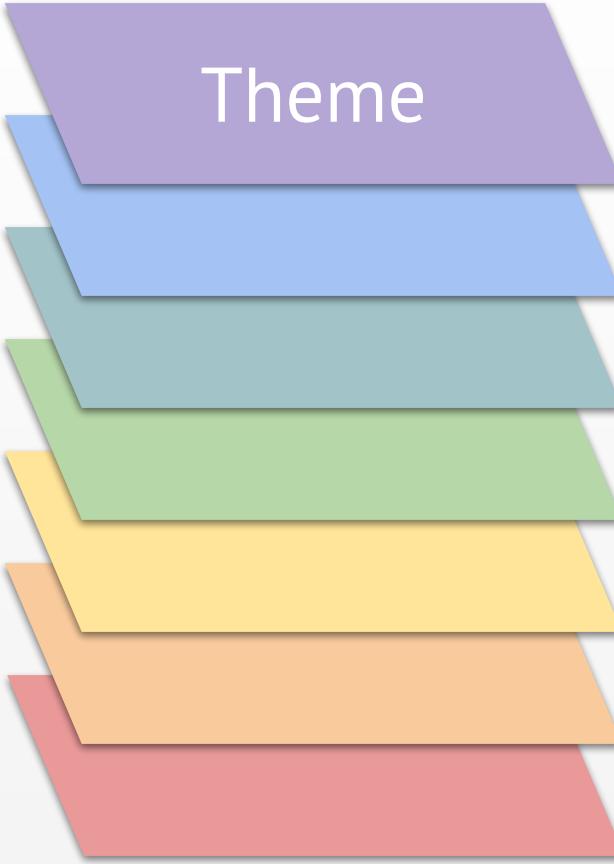
...and a **minimal theme**. Display a **legend** on the **right**.

## Code equivalent:

```
theme_minimal() +  
theme(legend.position = "right")
```

## Grammar of graphics:

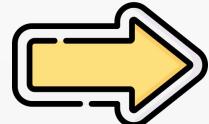
Theme



Create a data visualization using the salaries dataset. Put average salary on the y-axis, and job role on the x-axis. Use a different colour for each job role. Create a bar chart that's faceted by the industry variable. Add a horizontal black line showing the average salary for each industry. Format the y-axis as dollar amounts. Use a modern colour palette and a minimal theme. Display a legend on the right.

# Input: Prompt

Create a data visualization using the salaries dataset. Put average salary on the y-axis, and job role on the x-axis. Use a different colour for each job role. Create a bar chart that's faceted by the industry variable. Add a horizontal black line showing the average salary for each industry. Format the y-axis as dollar amounts. Use a modern colour palette and a minimal theme. Display a legend on the right.

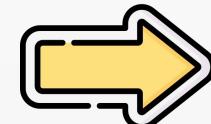
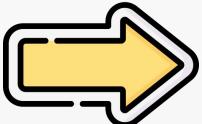


# Output: Python code

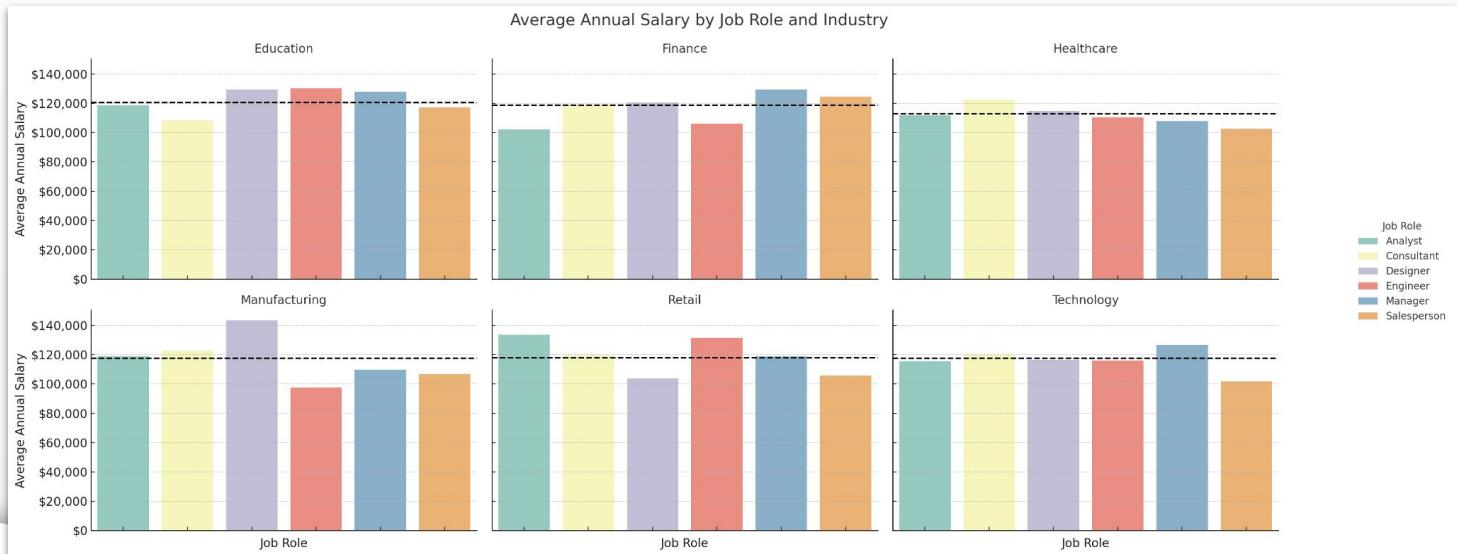
```
python
```

 Copy code

```
# Create the plot using seaborn
g = sns.catplot(
    data=average_salary_by_role,
    x='Job Role',
    y='Annual Salary',
    hue='Job Role',
    kind='bar',
    col='Industry',
    col_wrap=3,
    height=4,
    aspect=1.5,
    palette='Set3',
    dodge=False,
    ci=None
)
# Add horizontal lines for average salary in each industry
...
```



# Python code is executed to produce: Image



# You can now run Python code directly within the chat...

- See if there's errors right away
- No longer need to copy-paste into IDE
- Less switching back and forth between windows

<https://openai.com/index/hello-gpt-4o/>

# For the GenAI-Fatigued

# Ice Cream Maker



Ready in  
20 minutes!

After putting the  
“freezer bowl” in the  
freezer for 24 hours



# Amount of time required:

**Using the Ice Cream Maker:**

Total time required: **24.33 hours**

- Pre-freezing the bowl: 24 hours
- Making the ice cream: 20 minutes

**Buying store-bought ice cream:**

Total time required: **24.33 hours**

- Working up the energy to do groceries: 24 hours
- Buying the ice cream: 20 minutes

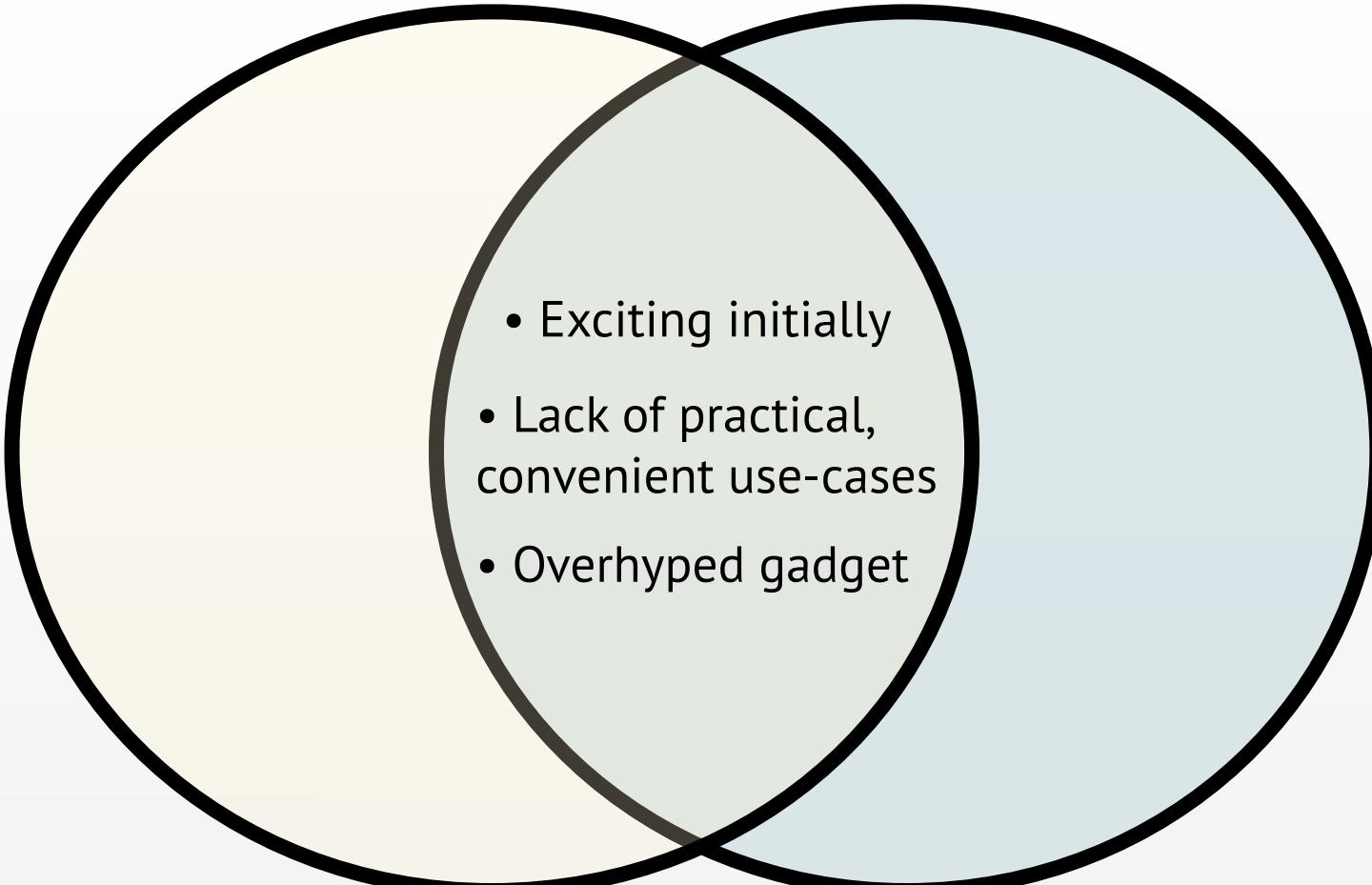
# Ice Cream Maker



Ready in  
20 minutes!

## Ice Cream Maker

## Generative AI

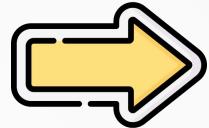
- 
- A Venn diagram consisting of two overlapping circles. The left circle is light beige and labeled 'Ice Cream Maker'. The right circle is light blue and labeled 'Generative AI'. The intersection of the two circles contains a list of three bullet points.
- Exciting initially
  - Lack of practical, convenient use-cases
  - Overhyped gadget

**A quick preview  
before we get into  
technical stuff...**

# A Quarto revealjs theme generator...

Enter a phrase  
for the theme:

abstract  
purple minimal



Presentation Title

Your Name Goes Here

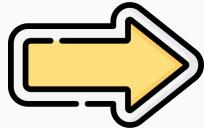
Generate A white mouse cursor icon pointing at the "Generate" button.

# A way to generate hex sticker designs...

Enter a phrase  
for the sticker:

cute baby  
python

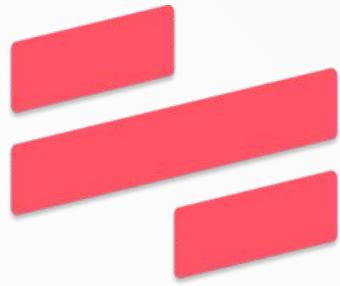
Generate 



pkgname



Some tools that make teaching/communicating about data a bit easier...

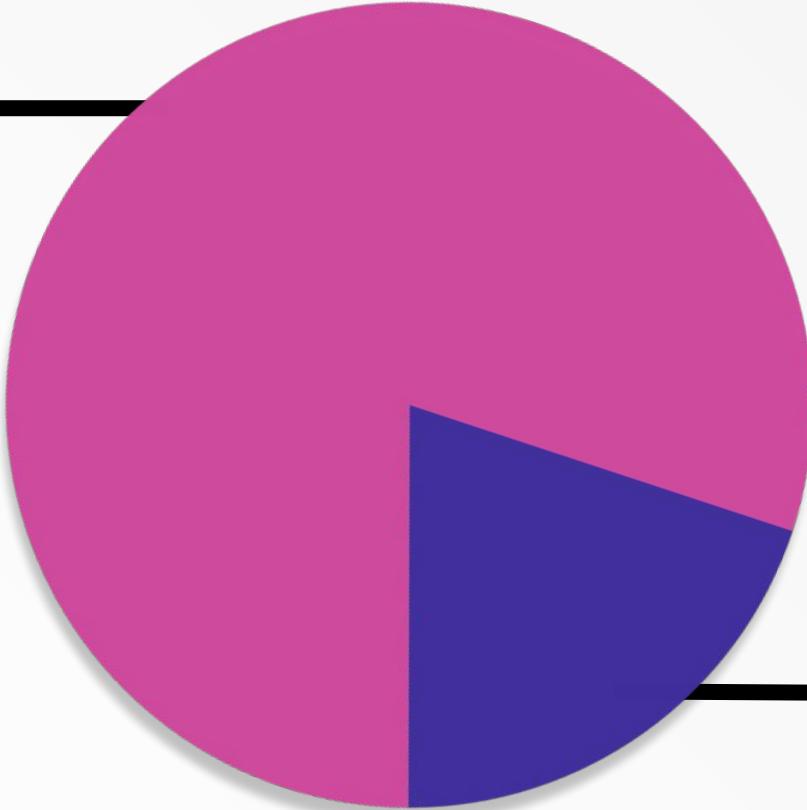


Scribe



descript

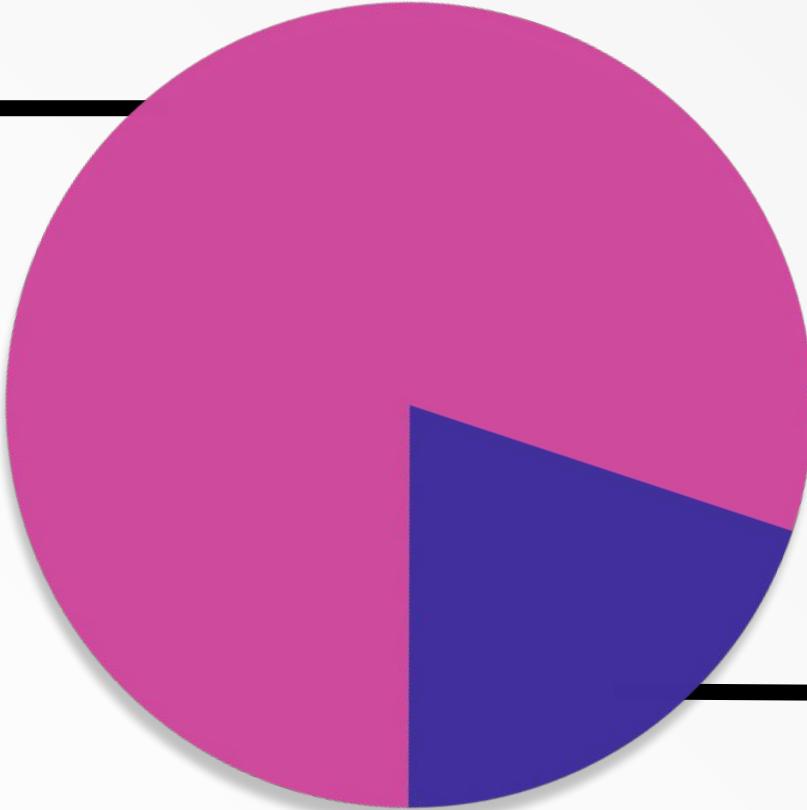
**Data** —————  
**Cleaning**



**Everything  
Else**

~~Data  
Cleaning~~

**Making  
Things  
Look  
Pretty**

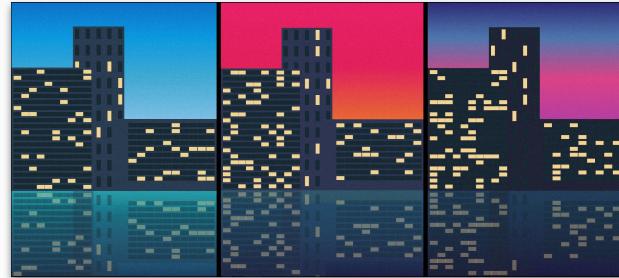


**Everything  
Else**

# Making things look pretty is something that this particular community is really good at, e.g.,



**Allison Horst's**  
Data Science  
Illustrations



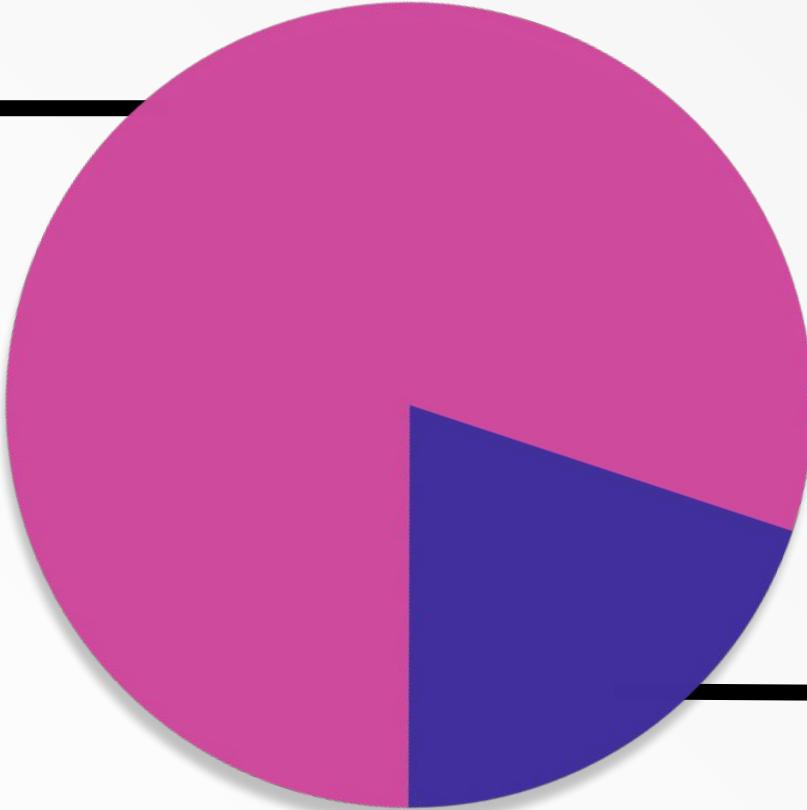
**Meghan Harris'**  
Generative Art  
using R

A screenshot of a presentation slide titled "Slidecraft 101: plot backgrounds". It shows a scatter plot with a background gradient from blue to orange. Below the slide, there are three smaller examples: "hi-orange-blue" showing a similar plot with a different color scheme, "prismatic with asciicast" showing a grid of colored squares, and "Base R - after" showing a standard R scatter plot.

**Emil Hvitfeldt's**  
Slidecraft 101  
series

~~Data  
Cleaning~~

**Making  
Things  
Look  
Pretty**



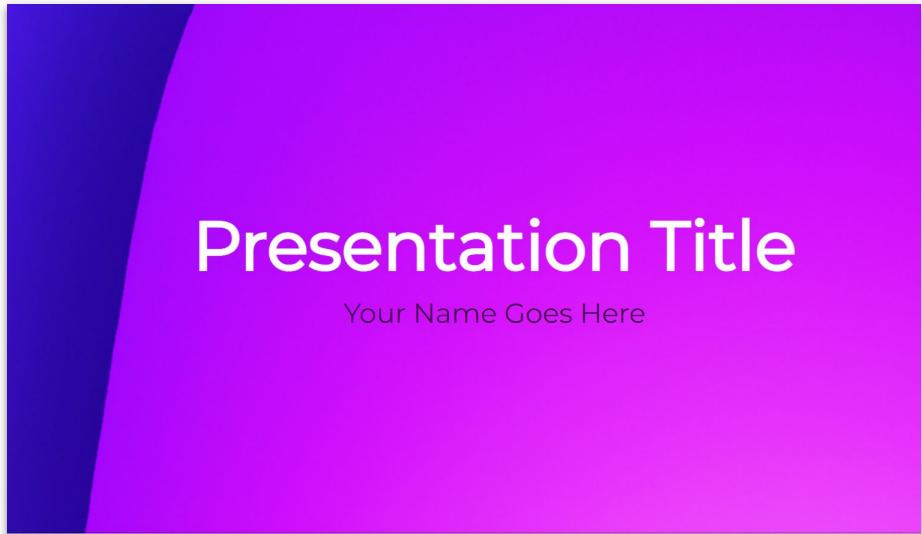
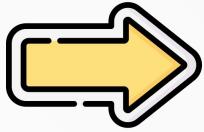
**Everything  
Else**

# A Quarto revealjs theme generator...

Enter a phrase  
for the theme:

abstract  
purple minimal

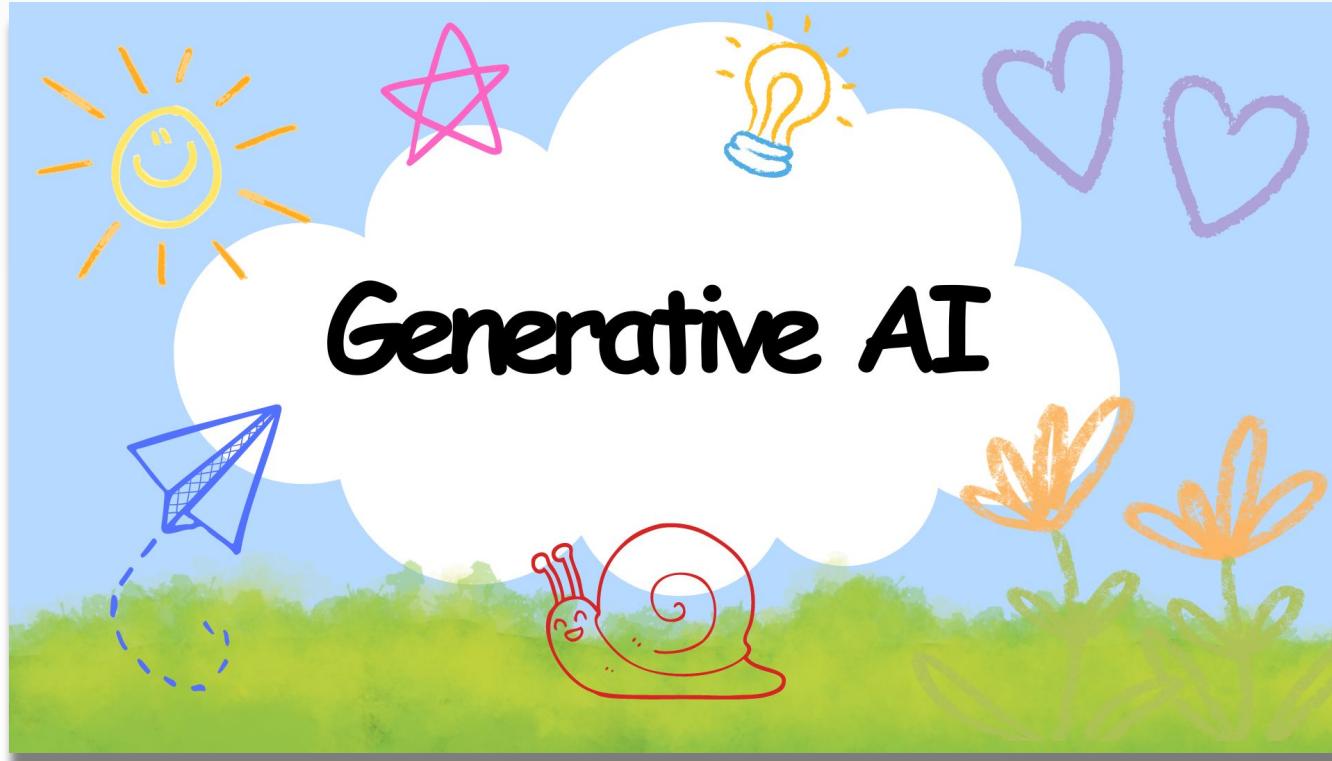
Generate 

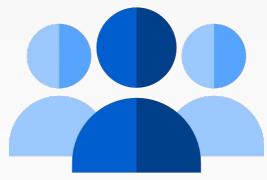


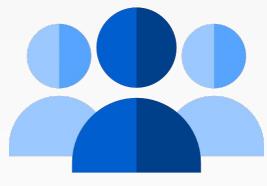
I want a new slide  
theme for every talk I  
give

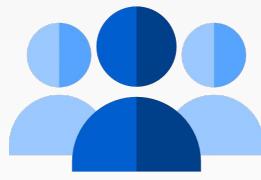


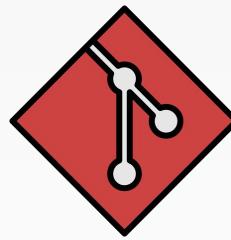
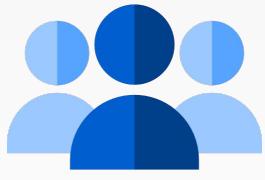
# Not OK for a talk about AI:

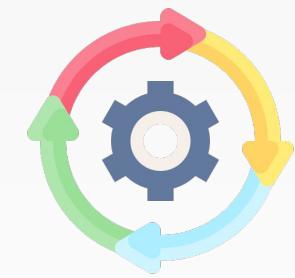
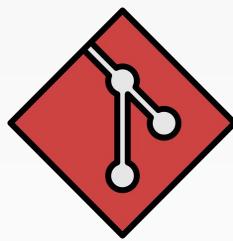
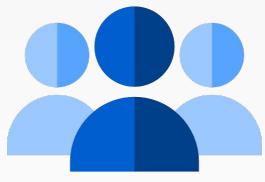






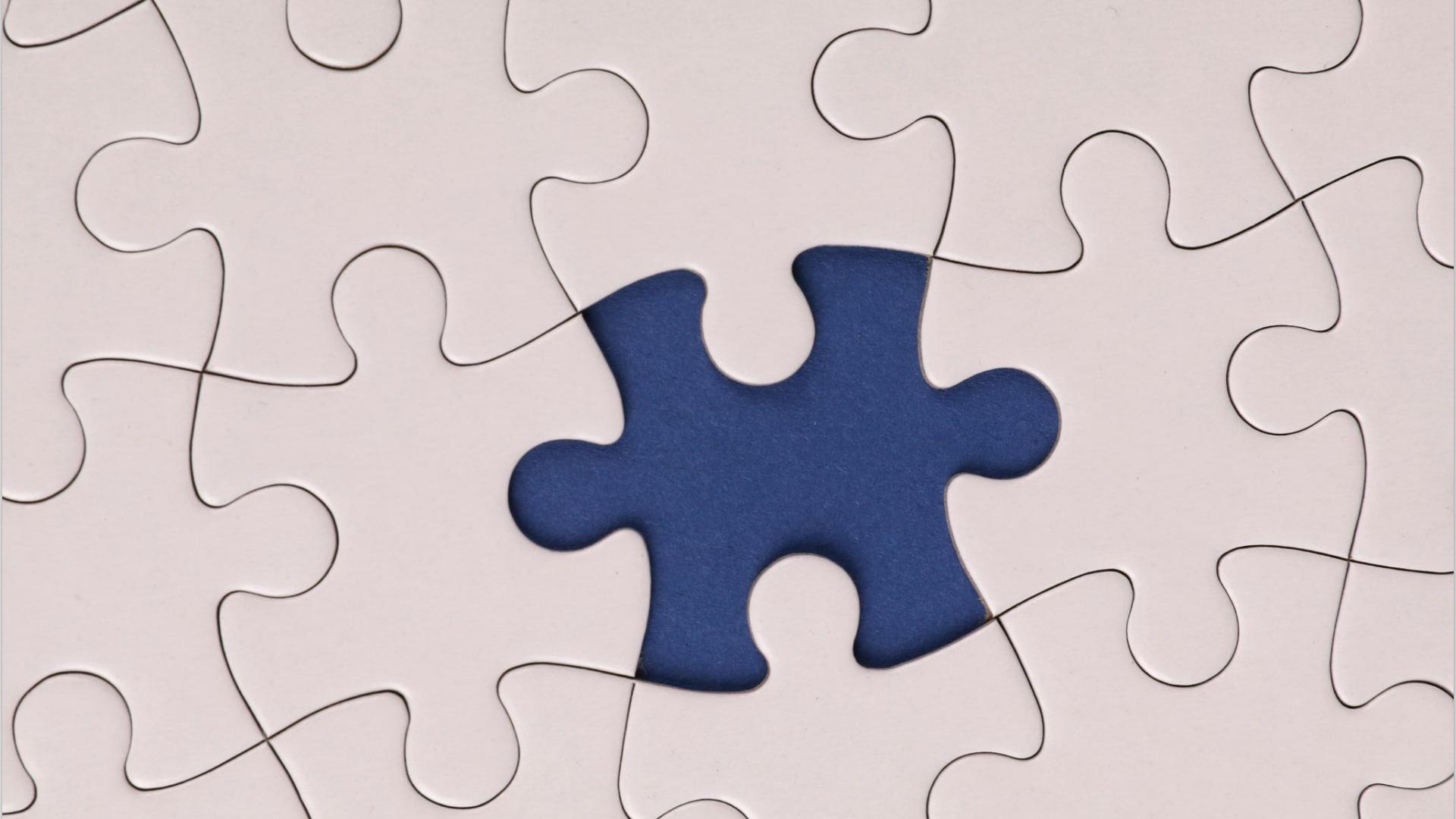


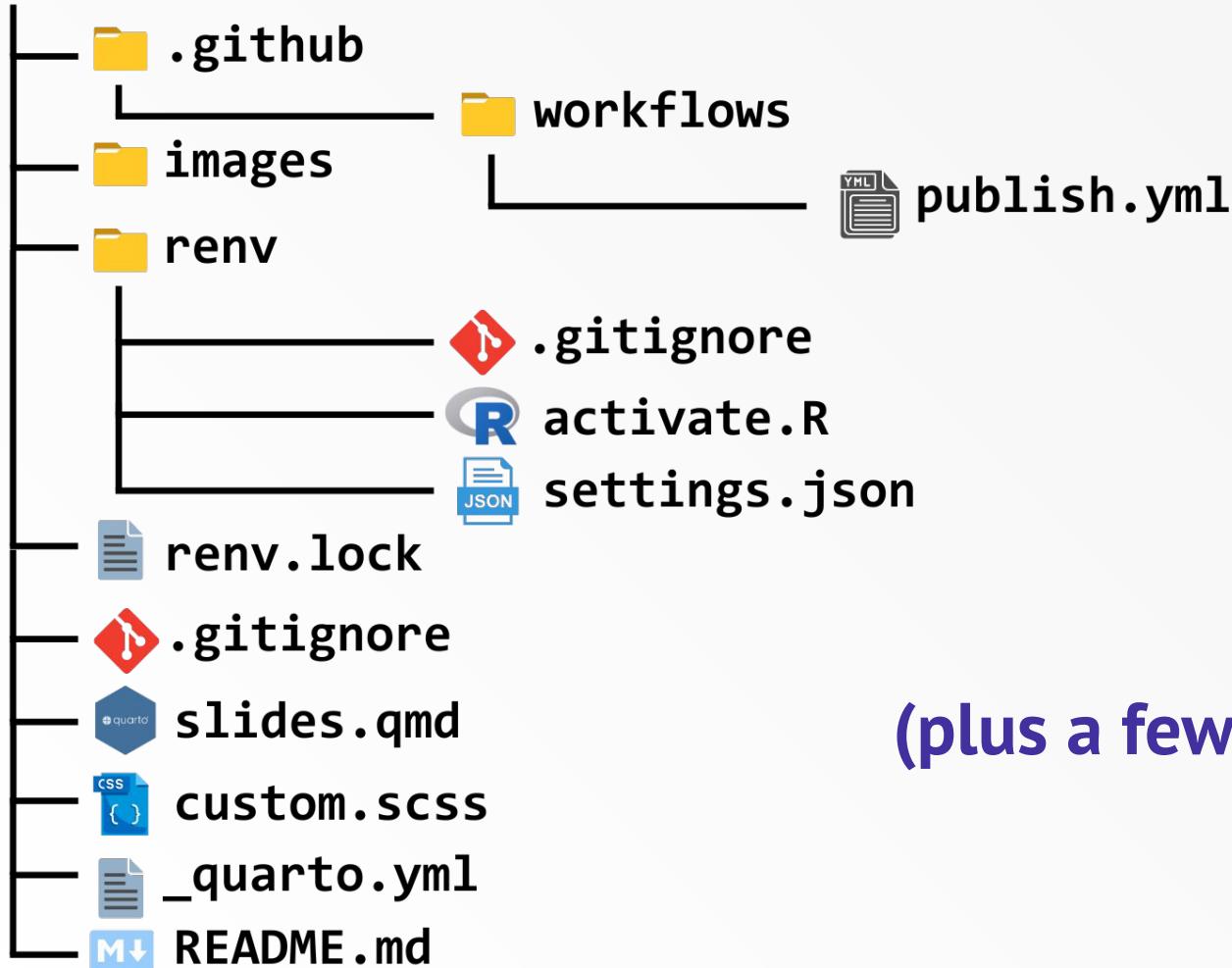




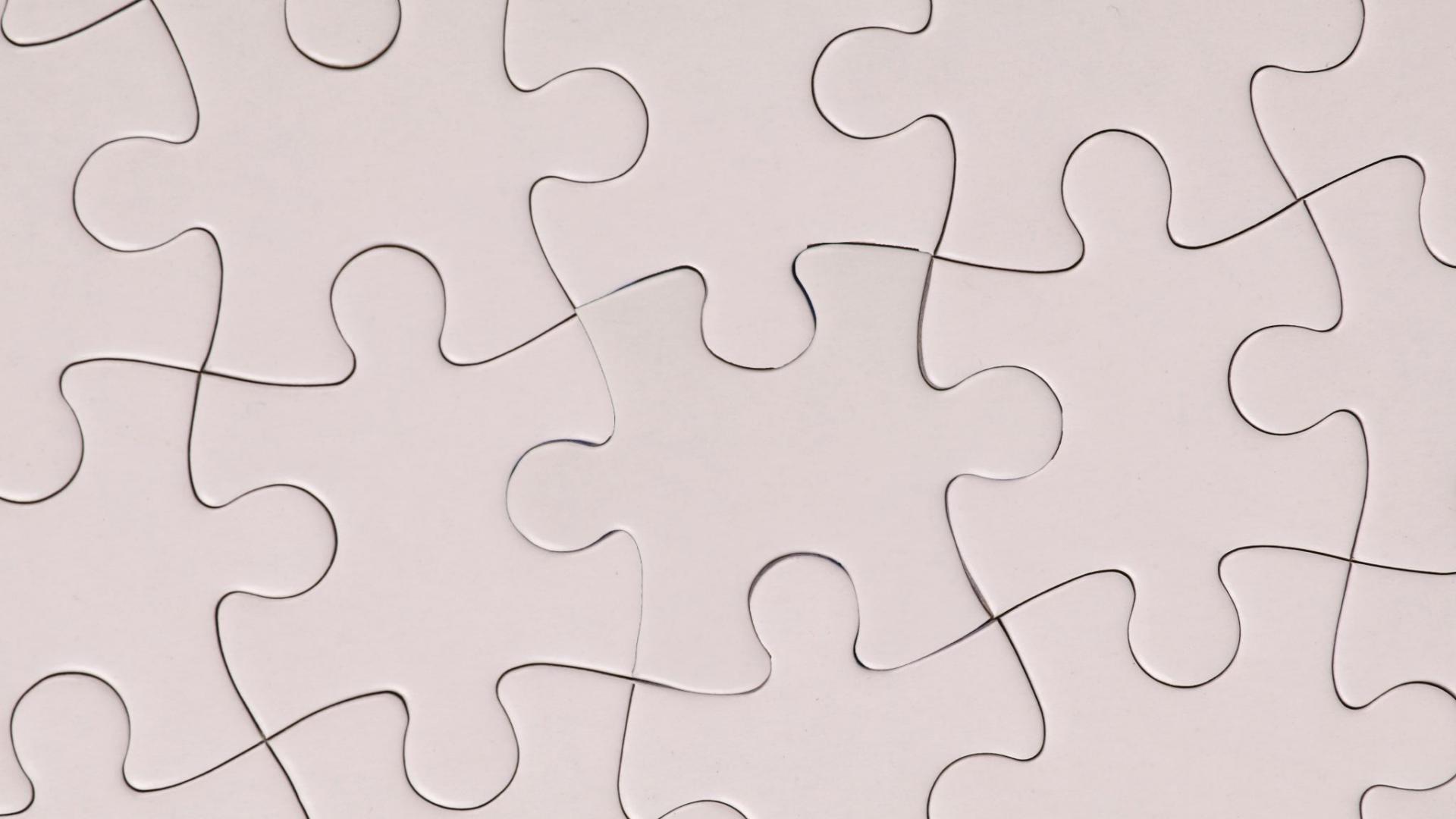


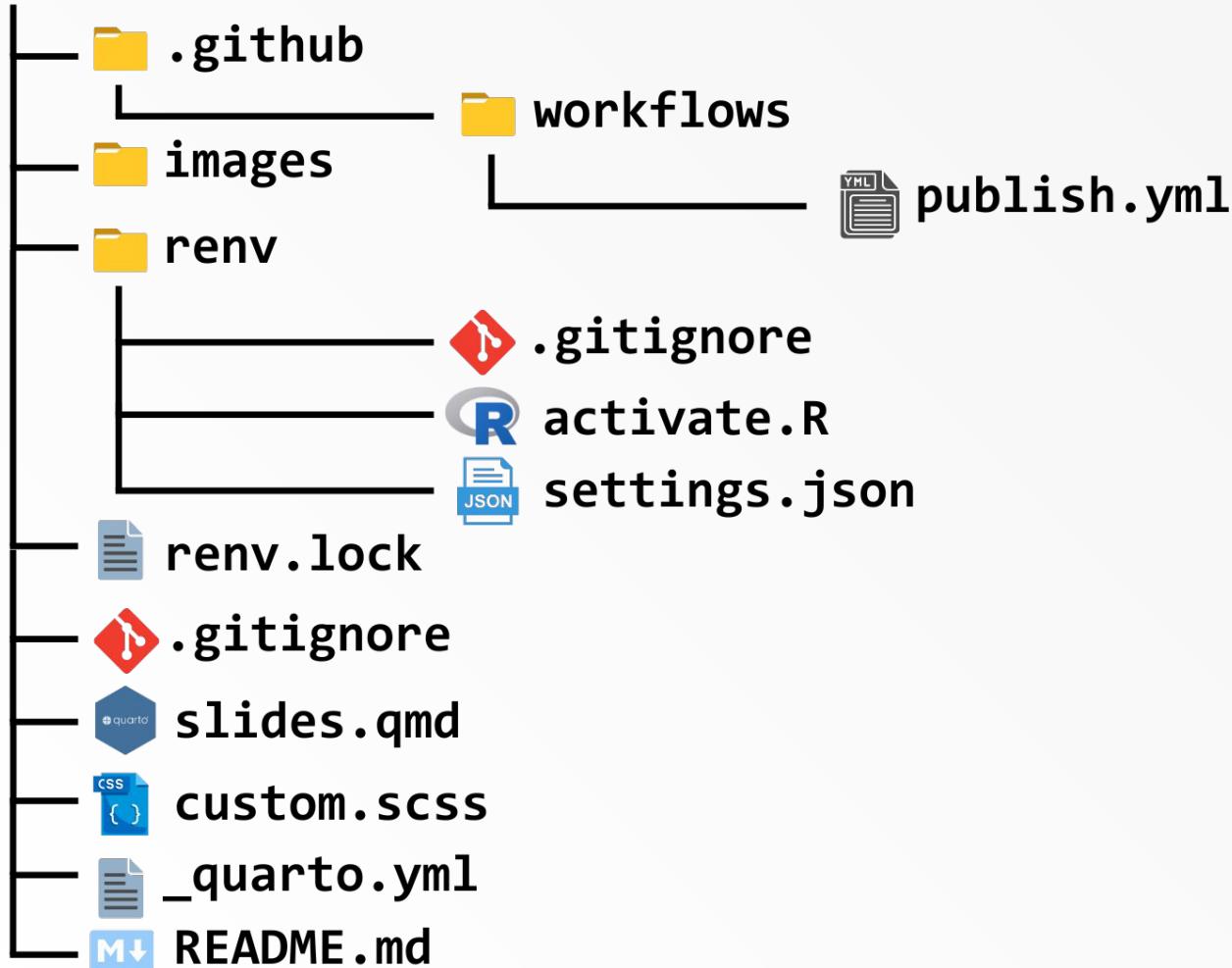
(or GitLab Pages...  
same idea!)

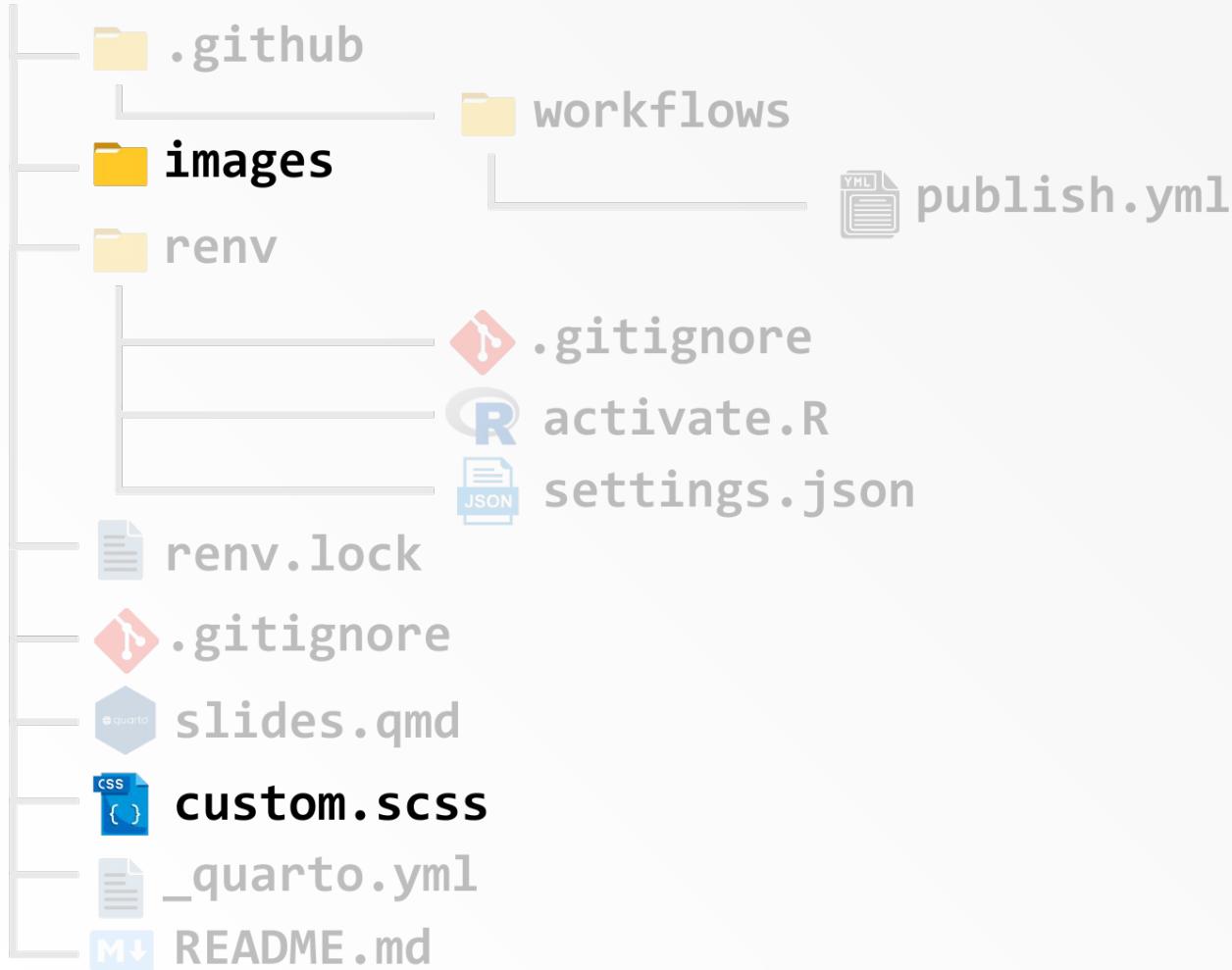




(plus a few more files...)







# custom.scss



```
/*-- scss:defaults --*/
$primary-color: #800080;
$secondary-color: #BF00BF;
$tertiary-color: lighten($secondary-color, 15%);
$accent-color: #F6851B;
$theme-white: #fff;
$theme-black: #000;

@import url('https://fonts.googleapis.com
            /css2?family=Poppins&display=swap');

$font-family-sans-serif: 'Poppins', sans-serif;
```

# custom.scss



```
/*-- scss:defaults --*/
$primary-color: #800080;
$secondary-color: #BF00BF;
$tertiary-color: lighten($secondary-color, 15%);
$accent-color: #F6851B;
$theme-white: #fff;
$theme-black: #000;

@import url('https://fonts.googleapis.com
            /css2?family=Poppins&display=swap');

$font-family-sans-serif: 'Poppins', sans-serif;
```

# Folder structure stays the same

**Folder structure stays the same**

+

**Making a new Quarto theme means getting new background images and making minimal styling changes**

**Folder structure stays the same**

+

**Making a new Quarto theme means getting new background images and making minimal styling changes**

+

**Gen AI is good\* at generating images and writing code**

\* = good enough

**Folder structure stays the same**

+

**Making a new Quarto theme means getting new background images and making minimal styling changes**

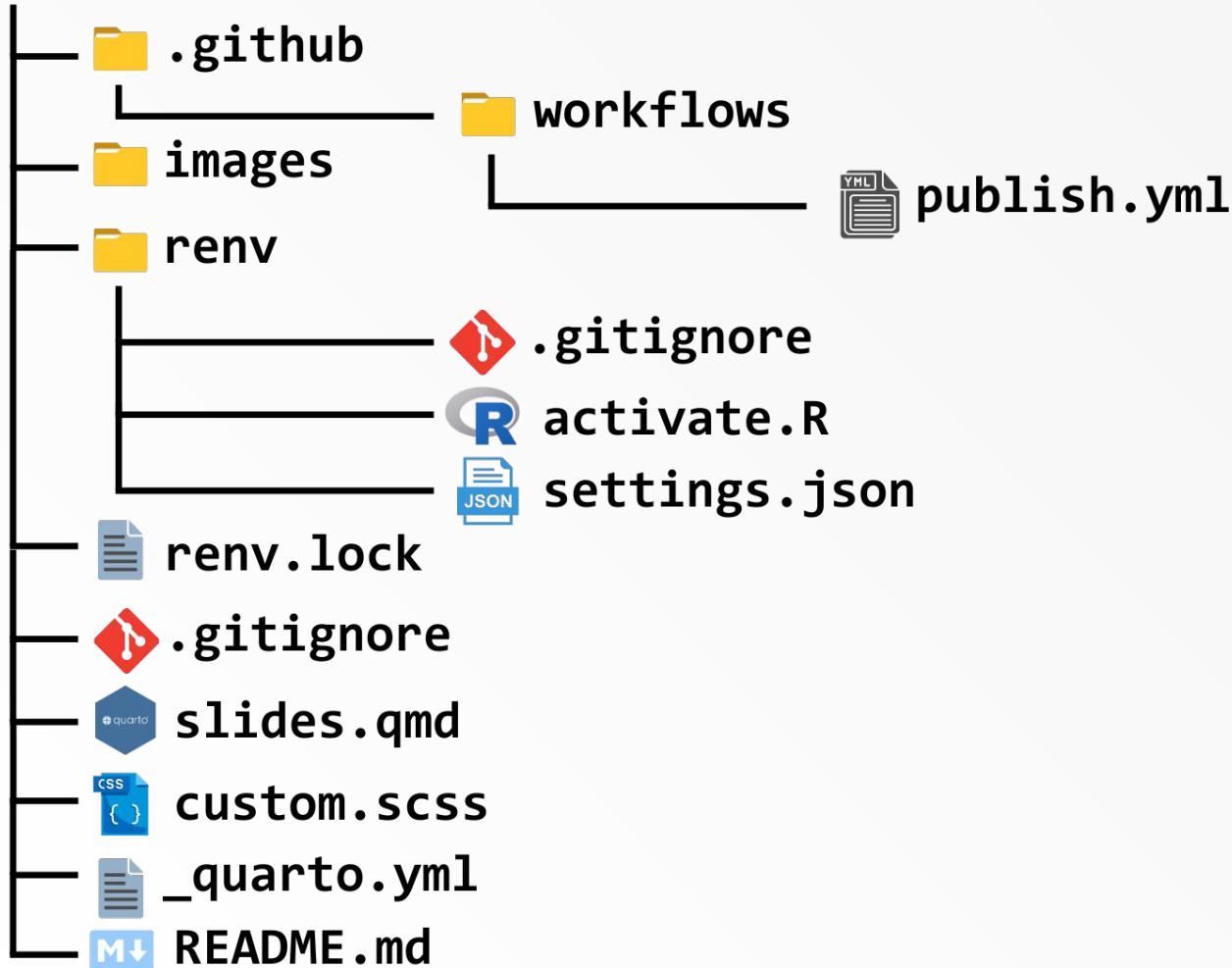
+

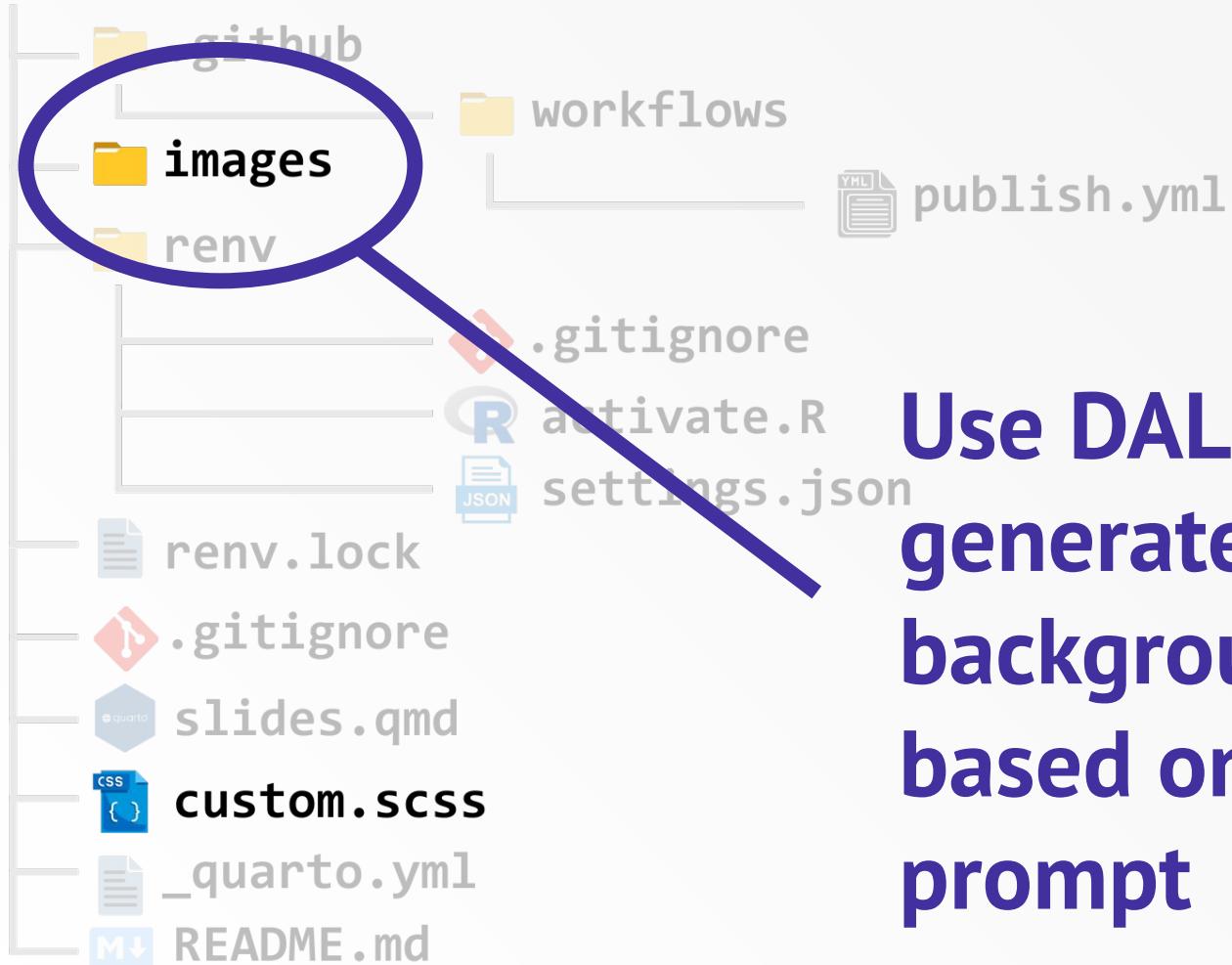
**Gen AI is good\* at generating images and writing code**

=

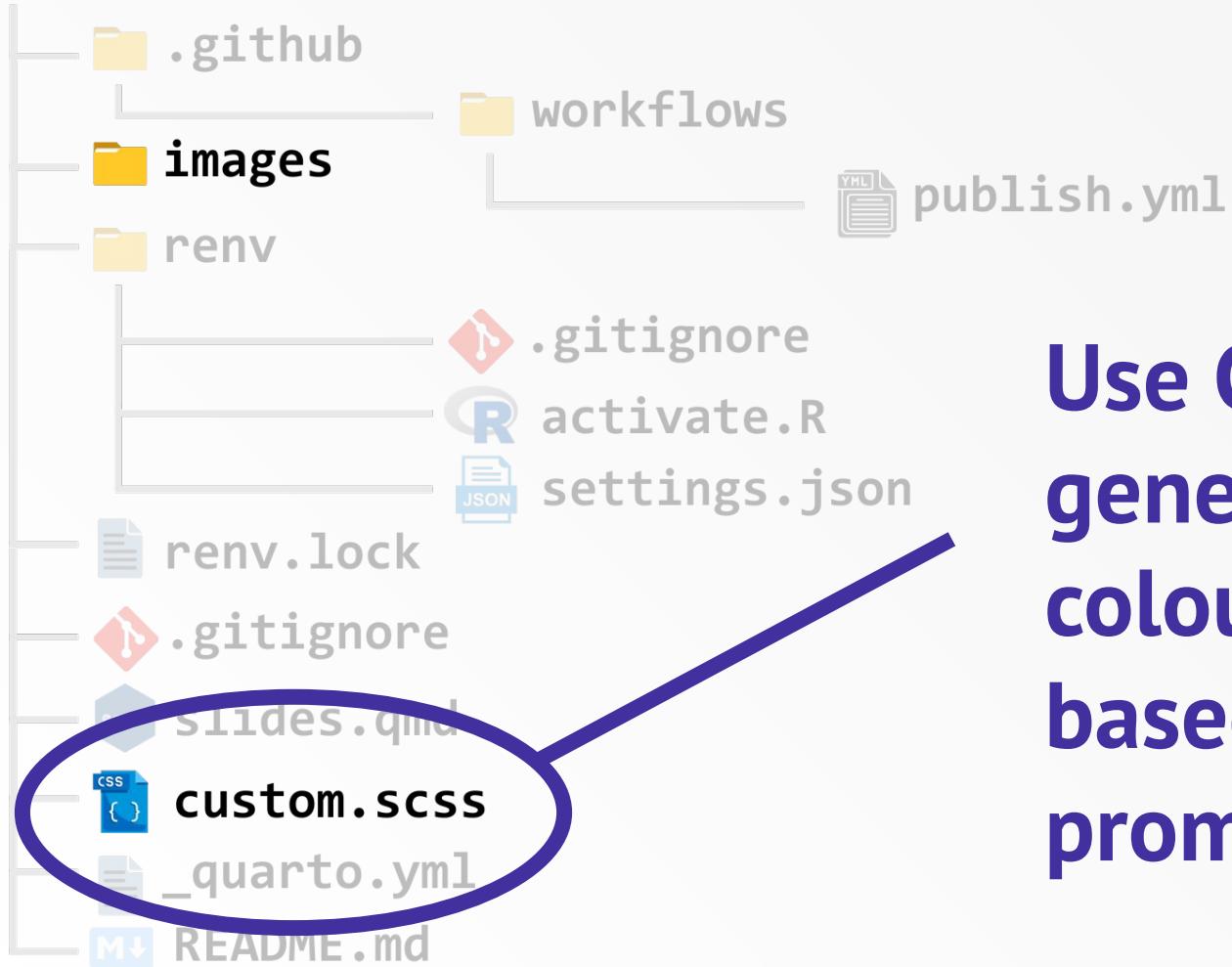
**Perfect use case for Generative AI**

\* = good enough





**Use DALL-E 3 to  
generate slide  
background images  
based on user  
prompt**



Use ChatGPT to  
generate  
colours/fonts  
based on user  
prompt

# GENERATIVE AI QUARTO THEME GENERATOR

Enter a phrase for the theme:

Abstract purple minimal

Enter your OpenAI API key

To generate images for slide backgrounds, use the checkbox below. Note that you must run it locally and not from shinyapps.io to generate images, as the app will time out in the browser. Download the app from [my GitHub](#) to run it locally.

Generate images in addition to CSS?

**GENERATE MY THEME**

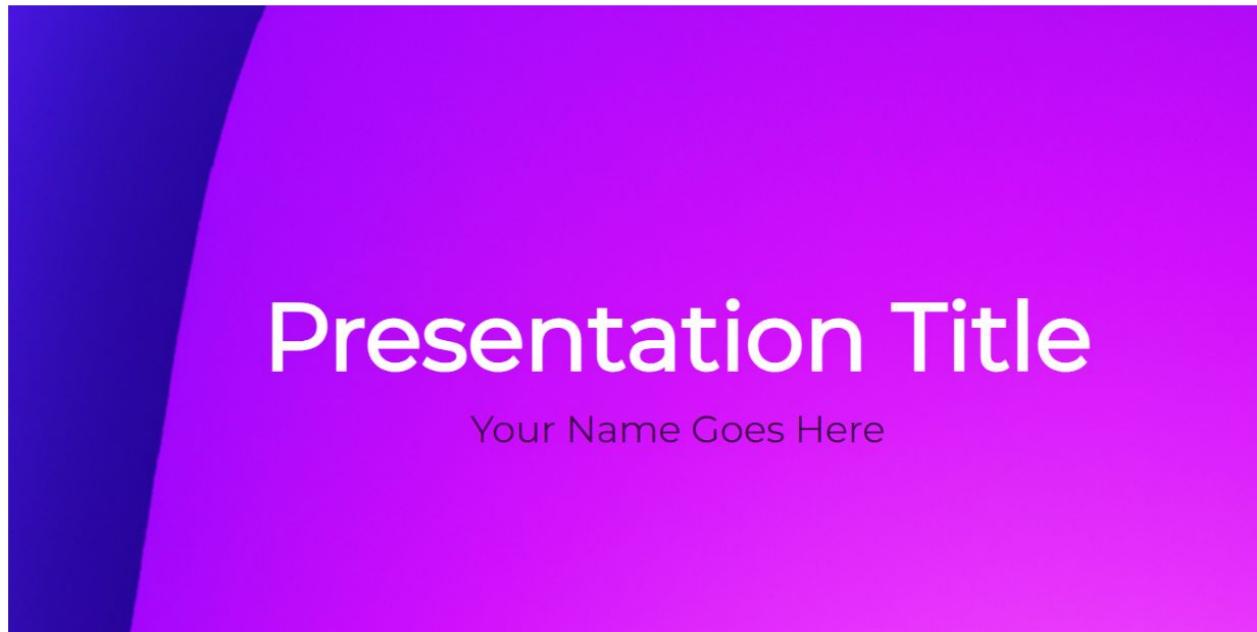
Once you see a message highlighted in purple saying "Theme generated successfully!", you may download all the generated files.

 **DOWNLOAD ALL FILES**

© 2024 [Melissa Van Bussel](#)

Theme has not yet been generated, so there are no files to download.

This app will take your phrase, and pass it to ChatGPT to choose fonts and colours that the app will then use to generate a Quarto revealjs theme. If the "generate images" checkbox is checked off, 10 images will be generated by the image generation endpoint as part of this process. Please see the [OpenAI API pricing page](#) to determine the cost of using this feature (approximately \$0.40 as of June 2024). If the "generate images" checkbox is not checked, the cost of running the app will be less than a penny based on June 2024 API pricing.



# Presentation Title

Your Name Goes Here

# Section 1

# Section 2

# Section 3

# Slide 1

This is some regular text, and this is some **bold text**, and this is a hyperlink. This is some `inline_code`.

- This is a list item
- So is this
- And this

```
1 print('This is some code')
```



# Slide 2

This is some regular text, and this is some **bold text**, and this is a hyperlink. This is some `inline_code`.

- This is a list item
- So is this
- And this

```
1 print('This is some code')
```



# Slide 3

This is some regular text, and this is some **bold text**, and this is a hyperlink. This is some `inline_code`.

- This is a list item
- So is this
- And this

```
1 print('This is some code')
```



# Slide 4

This is some regular text, and this is some **bold text**, and this is a hyperlink. This is some `inline_code`.

- This is a list item
- So is this
- And this

```
1 print('This is some code')
```



# Slide 5

This is some regular text, and this is some **bold text**, and this is a hyperlink. This is some `inline_code`.

- This is a list item
- So is this
- And this

```
1 print('This is some code')
```



# Slide 6

This is some regular text, and this is some **bold text**, and this is a hyperlink. This is some `inline_code`.

- This is a list item
- So is this
- And this

```
1 print('This is some code')
```



...

```
request("https://api.openai.com/v1") |>  
  req_url_path_append("/images/generations") |>  
  req_auth_bearer_token(input$api_key) |>  
  req_body_json(list(  
    prompt = paste(user_input, "desktop background")))) |>  
  req_perform() |>  
  resp_body_json()
```

...

...

```
request("https://api.openai.com/v1") |>  
req_url_path_append("/images/generations") |>  
req_auth_bearer_token(input$api_key) |>  
req_body_json(list(  
  prompt = paste(user_input, "desktop background")) |>  
req_perform() |>  
resp_body_json()
```

...

...

```
request("https://api.openai.com/v1") |>  
req_url_path_append("/images/generations") |>  
req_auth_bearer_token(input$api_key) |>  
req_body_json(list(  
  prompt = paste(user_input, "desktop background")))) |>  
req_perform() |>  
resp_body_json()
```

...

**User input:**  
“Pastel chalk”

**Prompt passed to DALL-E 3:**  
“Pastel chalk desktop background”

# “Pastel chalk”:



# “Pastel chalk desktop background”:



```
background-image: url('slide1.png');  
@include background-full;  
opacity: 0.2;
```

...

```
request("https://api.openai.com/v1") |>  
req_url_path_append("/chat/completions") |>
```

...



...

```
messages = list(  
  list(  
    role = "system",  
    content = "You are a design assistant."  
)  
,  
  list(role = "user", content = prompt_to_send)  
)
```

...

ser

paste("Based on the theme", user\_input, ", recommend a Google font family and three colors for primary, secondary, and accent. The accent color should contrast the other colors. The font family should be the full name used by Google Fonts. Also recommend a pandoc highlight-style name that matches. Provide the recommendations in JSON format with keys: 'font\_family', 'primary\_color', 'secondary\_color', 'accent\_color', and 'highlight\_style'."

ser

paste("Based on the theme", user\_input, " recommend  
a Google font family and three colors for primary,  
secondary, and accent. The accent color should  
contrast the other colors. The font family should be  
the full name used by Google Fonts. Also recommend  
a pandoc highlight-style name that matches. Provide  
the recommendations in JSON format with keys:  
'font\_family', 'primary\_color', 'secondary\_color',  
'accent\_color', and 'highlight\_style'."

paste("Based on the theme", user\_input, " recommend  
a Google font family and three colors for primary,  
secondary, and accent. The accent color should  
contrast the other colors. The font family should be  
the full name used by Google Fonts. Also recommend  
a pandoc highlight-style name that matches. Provide  
the recommendations in JSON format with keys:

'font\_family', 'primary\_color', 'secondary\_color',  
'accent\_color', and 'highlight\_style'."



```
...
scss_content <- paste0(
  "/**-- scss:defaults --*/\n",
  "$primary-color: ", recommendations$primary_color, ";\n",
  "$secondary-color: ", recommendations$secondary_color, ";\n",
  "$tertiary-color: lighten($secondary-color, 15%);\n",
  "$accent-color: ", recommendations$accent_color, ";\n",
  "$theme-white: #fff;\n",
  "$theme-black: #000;\n",
  \n
  font_url,
  \n
  "$font-family-sans-serif: '", recommendations$font_family,
  "'", sans-serif;\n"
  \n
```

Try the app on shinyapps.io:



[melissavanbussel.shinyapps.io/quarto\\_theme\\_generator](https://melissavanbussel.shinyapps.io/quarto_theme_generator)

My YouTube channel:



[youtube.com/c/ggnot2](https://youtube.com/c/ggnot2)



Ha I think the crowd would really love some applications of genai in data science - so far it's not entirely clear what a cat picture generator can do there (at least to me).



# R/Python User Group (RPUG)





# R/Python User Group (RPUG)



JB

Jude Bashto, 9:02 AM

**Happy new year, everyone!**



17



12





# R/Python User Group (RPUG)



JB

Jude Bashto, 9:36 AM

Welcome back, everyone!



Why does this pug  
have two ties?!

9



11





Ha I think the crowd would really love some applications of genai in data science - so far it's not entirely clear what a cat picture generator can do there (at least to me).

pkgname



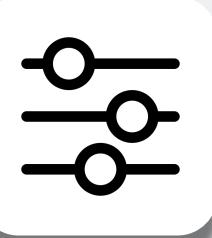
# laserpointer





<https://www.recraft.ai>

Art  
style

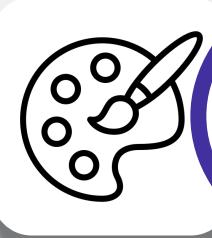
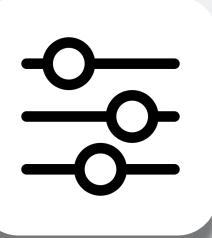


W: 1024  
H: 1024

cute kitten

Generate

Art  
style

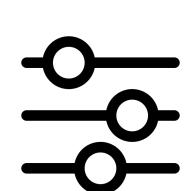


W: 1024  
H: 1024

cute kitten

Generate

Art  
style

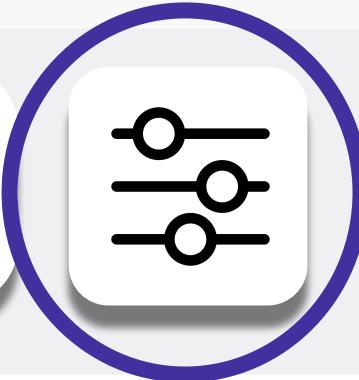


W: 1024  
H: 1024

cute kitten

Generate

Art  
style

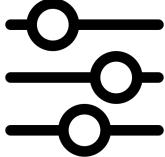


W: 1024  
H: 1024

cute kitten

Generate

Art  
style



W: 1024  
H: 1024

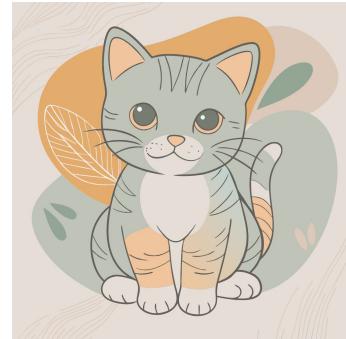
cute kitten

Generate

# Seamless



# Colour blobs



# Kawaii



# Doodle line



# Cartoon



# Offset doodle



# Seamless



# Colour blobs



# Kawaii



# Doodle line



# Cartoon

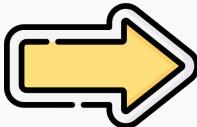


# Offset doodle





recolour



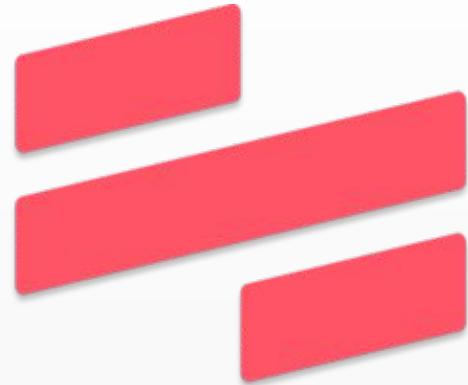
# laserpointer



# Recraft AI: Mock-ups



**Two tools that use  
Generative AI to make  
teaching and communicating  
about data easier**



# Scribe

**Automatically create  
step-by-step  
tutorials without  
copy-pasting  
screenshots or  
recording videos**



<https://posit.cloud>



Start capture

1

Navigate to <https://posit.cloud>

2

Click "New Project"

The screenshot shows the Posit Cloud workspace interface. At the top, there is a navigation bar with a menu icon, the text "Your Workspace", a dropdown for "Menu", and a user name "Melissa Van Bussel". On the far right of the header are three circular icons: a pencil, a plus sign, and the text "ALT". Below the header, the main area is titled "Your Content" with a count of "(2)". A search bar at the top of this section includes filters for "TYPE", "ACCESS", "SORT", and a search input field. A prominent blue button labeled "New Project" with a dropdown arrow is located in the upper right of the content area. This button is circled in orange. Below it, two project entries are listed: "Example Project" and "My Example Project", each with a preview, creation date, and a set of action icons (copy, delete, download, more). The bottom of the screen features a dark footer bar with the "posit Cloud" logo, social media links for Facebook, LinkedIn, Instagram, and GitHub, and the copyright notice "© 2022 Posit Software, PBC". There are also links for "Terms" and "Status".

3

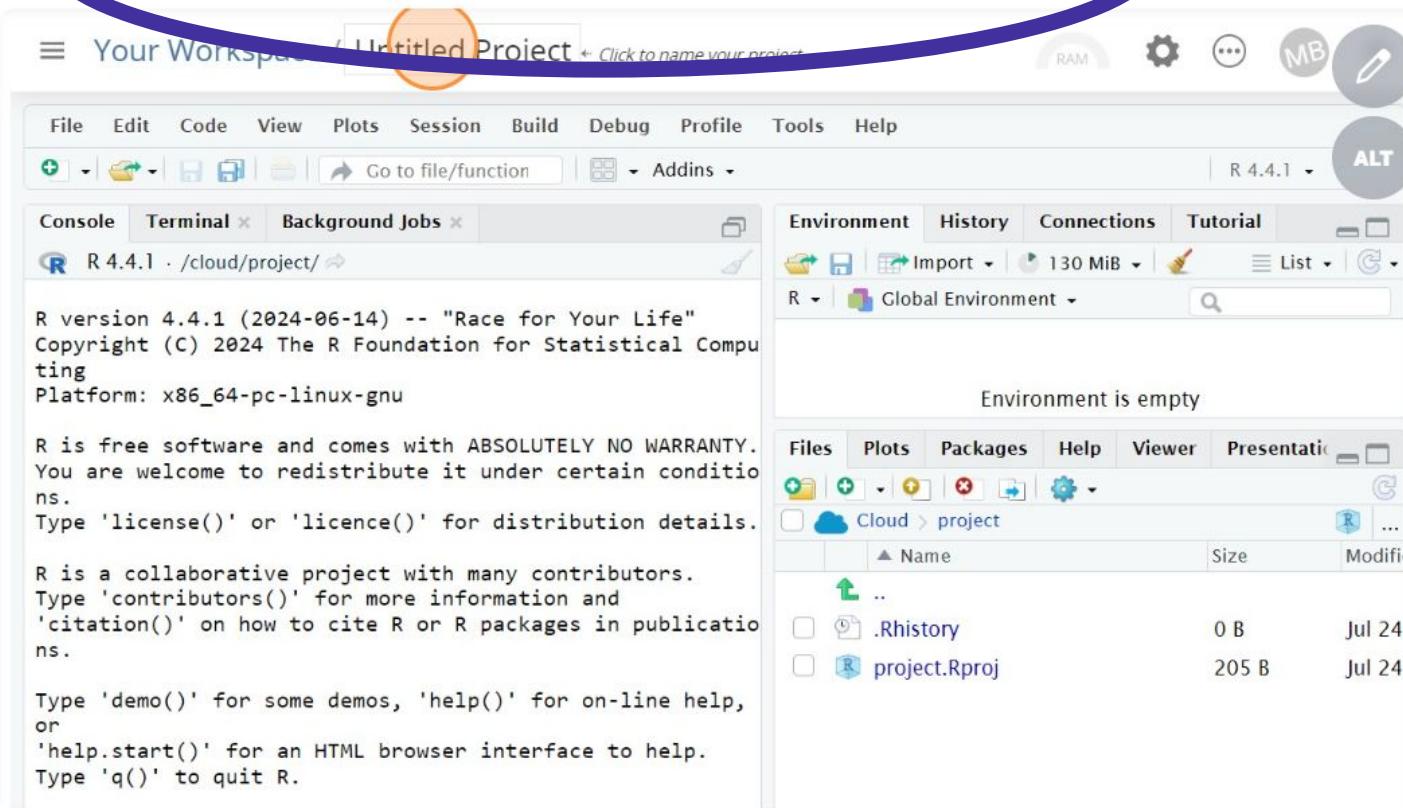
Click "New RStudio Project"

The screenshot shows the RStudio interface with the following elements:

- Header:** Your Workspace, Menu, Melissa Van Bussel.
- Content Area:** Your Content (2). It lists two projects:
  - Example Project:** RStudio Project, Private, Created May 30, 2024 5:53 PM.
  - My Example Project:** RStudio Project, Private, Created Jul 24, 2024 10:31 AM.
- Toolbar:** TYPE, ACCESS, SORT.
- Context Menu (opened over 'New RStudio Project'):** New Project dropdown menu with options:
  - New Project from Template
  - New RStudio Project** (highlighted with an orange circle)
  - New Jupyter Project
  - New Project from Git Repository
- Keyboard Shortcuts:** MB, ALT.

4

Click "Untitled Project" and rename the project.





## Create a New RStudio Project and Name it

This guide provides step-by-step instructions on how to create a new RStudio project and give it a custom name. It is helpful for individuals who want to organize their R code and data into separate projects and need guidance on setting up a new project in RStudio.

Melissa | 3 steps | 29 seconds



Posit

[Share](#)[Embed](#)[Export](#)

Share link over email/chat or send directly to specific people.

[Add emails to share](#) M

Melissa

Owner  Not shared ▾ Copy Link

# Embed as an iframe, for example on a Quarto website:

The screenshot shows a user interface for creating a new RStudio project. At the top is a circular icon with a blue 'X' symbol. Below it is the title 'Create a New RStudio Project and Name it'. A descriptive text block follows, stating: 'This guide provides step-by-step instructions on how to create a new RStudio project and give it a custom name. It is helpful for individuals who want to organize their R code and data into separate projects and need guidance on setting up a new project in RStudio.' Below this is author information: 'Melissa | 4 Steps | 29 seconds'. A 'Posit' logo is present. At the bottom, there's a navigation bar with icons for home, back, forward, and search, along with a 'Get Started' button. The footer indicates the guide was 'Made with Scribe'.

Create a New RStudio Project and Name it

This guide provides step-by-step instructions on how to create a new RStudio project and give it a custom name. It is helpful for individuals who want to organize their R code and data into separate projects and need guidance on setting up a new project in RStudio.

Melissa | 4 Steps | 29 seconds

Posit

Get Started

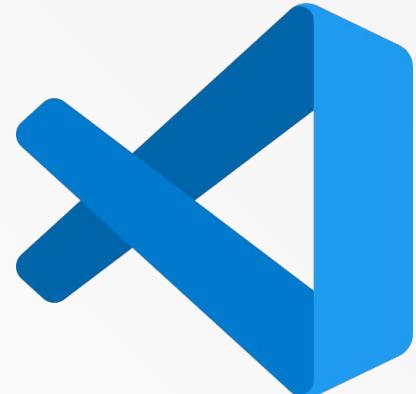
Made with Scribe



<https://posit.cloud>



Start capture

The R Studio logo icon is a blue circle containing a white letter 'R'. To the right of the icon, the word "Studio" is written in a large, dark gray sans-serif font, with a registered trademark symbol (®) at the top right of the 'o'.

[Share](#)[Embed](#)[Export](#)

View in other formats. These options do not automatically update.



Export as PDF PRO ⚡

[Preview](#)

Export as HTML PRO ⚡

Works well with Microsoft Word, Google Docs and other apps.

[Export](#)

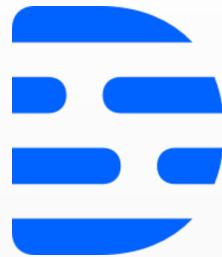
Export as Markdown PRO ⚡

Works well with Notion, GitHub and other apps.

[Export](#)

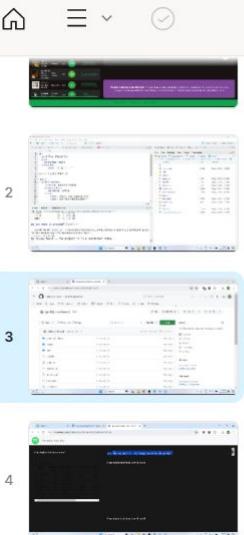
Export to Confluence PRO ⚡

[Export](#)



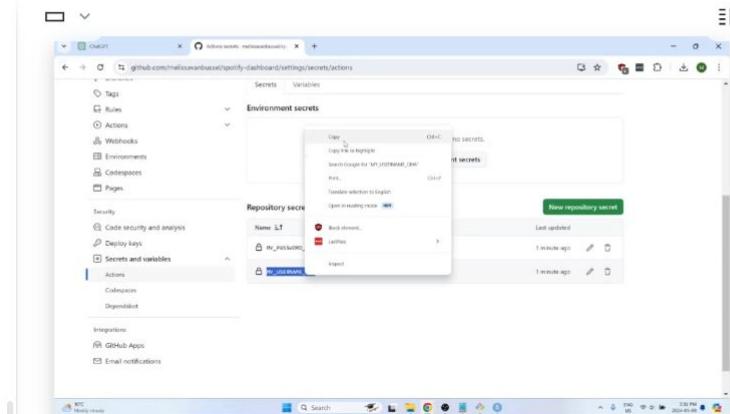
# descript

**Edit videos and audio/podcasts as if  
you're editing a Word document**



Write

scripts. On the left we're going to say the name of the environment variable and this is going to be the name that we use when we run the sys.getenv function. So I'm going to call this one my username and then just like the syntax above, I'm going to copy and paste that and then instead of it saying GitHub token, this is where you're going to put that name that you had defined here.



Hide timeline

07:04.5 / 07:37.4



5

7:03.0

7:03.5

7:04.0

7:04.5

7:05.0

7:05.5

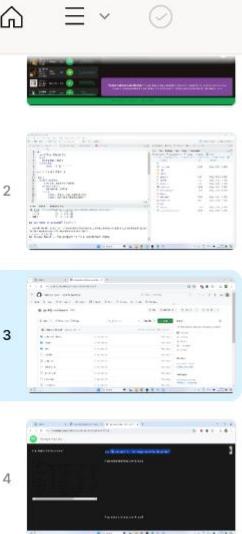
7:06.0

that you had defined here.

So . the first one is ... my



More



29w | 8.25s

scripts. On the left we're going to say the name of the environment variable and this is going to be the name that we use when we run the sys.getenv function. So I'm going to call this one my username and then just like the syntax above, I'm going to copy and paste that and then instead of it saying GitHub token, this is where you're going to put that name that you had defined here.

```
name: specify-database -- specify-database

on: [push]

jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Install R
        uses: r-lib/actions/setup-rbv2@v2
        with:
          r-version: '4.2.0'
      - name: Install R dependencies
        uses: r-lib/actions/setup-remdV2@v2
        with:
          cache-version: 1
      - name: Render and publish
        uses: quarto-dev/quarto-actions/publishDv2@v2
        with:
          target: gh-pages
          env:
            GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
            MY_USERNAME: ${{ secrets.MY_USERNAME }}
```

Hide timeline

06:56.2 / 07:37.4



6:56

6:57

6:58

6:59

7:00

7:01

7:02

7:03

7:04

I'm copy and pas that and then .. instead of it saying ... GitHub token, ... this is where yo put that name ... that y ha defined here. ... So

8.250s

Underlord

Project

Scene

Layer

Record

Elements

More



What would you like to do?



Remove filler words...



| A | Shorten word gaps...



Jump to...



Search in script...

Remove  
awkward  
silences



What would you like to do?



Remove filler words...



Shorten word gaps...



Jump to...



Search in script...

**Get rid of  
stuff like  
“um”, “uh”,  
“you know”**

**Clone your own voice in  
as little as 60 seconds**



Regenerate



# Overdub

## AI Features

Market / Promote

Look Good

Sound Good

Create clips

Eye Contact

Edit for clarity

YouTube  
descriptions

Green Screen

Studio Sound

Show notes

Automatic multicam

Remove filler  
words

Translation

Video Regenerate SOON

Remove retakes

Regenerate  
speech

## AI Features

Market / Promote

Create clips

YouTube  
descriptions

Show notes

Translation

Look Good

**Eye Contact**

Green Screen

Automatic multicam

Video Regenerate SOON

Sound Good

Edit for clarity

Studio Sound

Remove filler  
words

Remove retakes

Regenerate  
speech

## AI Features

Market / Promote

Look Good

Sound Good

Create clips

Eye Contact

Edit for clarity

YouTube  
descriptions

Green Screen

Studio Sound

Show notes

Automatic multicam

Remove filler  
words

Translation

Video Regenerate SOON

Remove retakes

Regenerate  
speech

## AI Features

Market / Promote

Look Good

Sound Good

Create clips

Eye Contact

Edit for clarity

YouTube  
descriptions

Green Screen

Studio Sound

Show notes

Automatic multicam

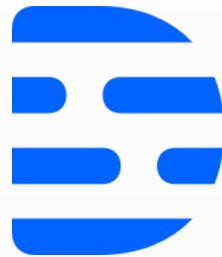
Remove filler  
words

Translation

Video Regenerate SOON

Remove retakes

Regenerate  
speech



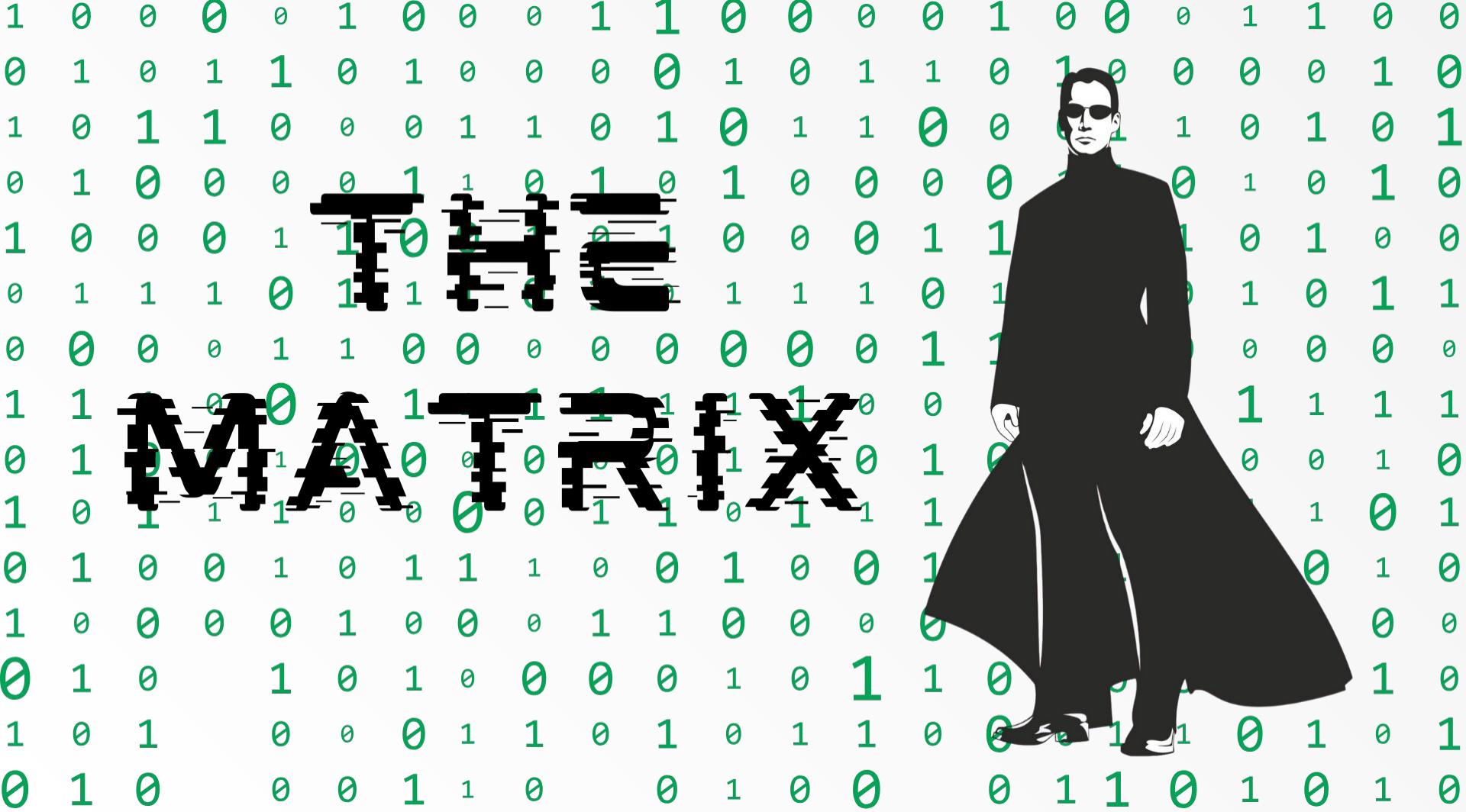
# descript

**Edit videos and audio/podcasts as if  
you're editing a Word document**

# For the GenAI-Fearful

- 1. Fanatic**
- 2. Fatigued**

- 1. Fanatic**
- 2. Fatigued**
- 3. Fearful**

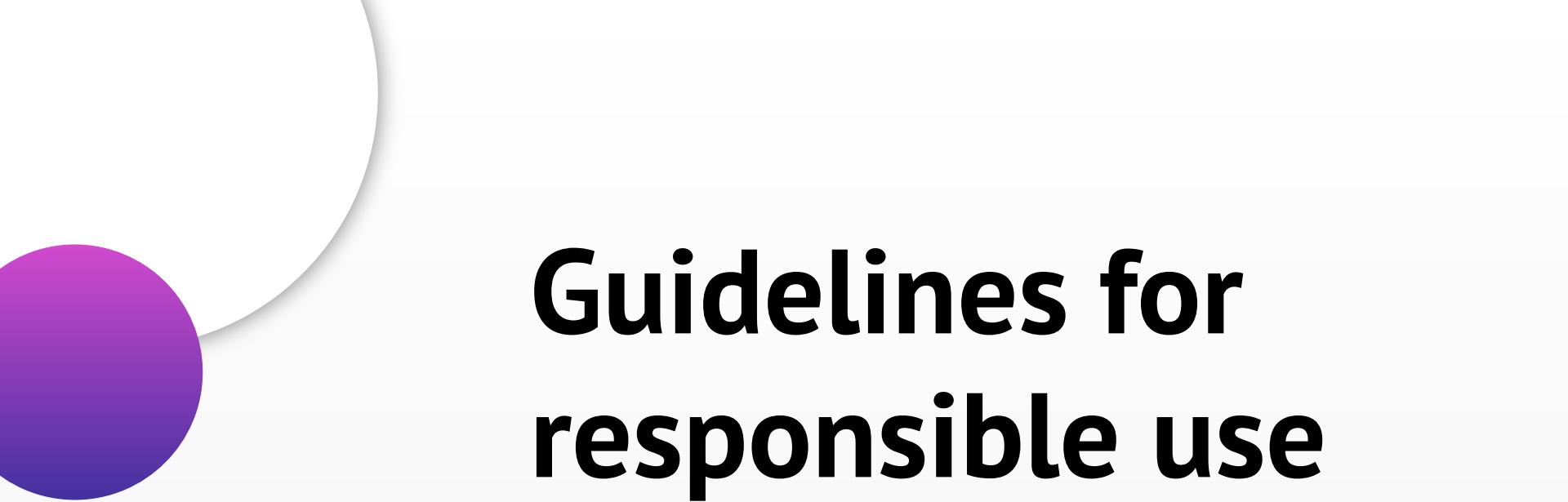


- 1. Fanatic**
- 2. Fatigued**
- 3. Fearful**

1. Fanatic
2. Fatigued

~~3. Fearful~~

3. Hesitant



# **Guidelines for responsible use**



# Search on Google: “Guide on the use of Generative AI Government of Canada”

<https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/responsible-use-ai/guide-use-generative-ai.html>

**FASTER**  
stands for...

**Fair**  
**Accountable**  
**Secure**  
**Transparent**  
**Educated**  
**Relevant**

# Fair

# Fair

- Determine what was used as training data

# Fair

- Determine what was used as training data
- Always review outputs; remove biased, non-inclusive or discriminatory content

# Fair

- Determine what was used as training data
- Always review outputs; remove biased, non-inclusive or discriminatory content
- Don't use Generative AI to make decisions that could legally or materially impact an individual

# Fair

## OpenAI Terms of Use:

“You **must not** use any Output relating to a person for any purpose that could have a legal or material impact on that person, such as making **credit, educational, employment, housing, insurance, legal, medical,** or other important decisions about them.”

# Fair

## Google's Generative AI Prohibited Use Policy:

“You **must not** use the Google services that reference this policy to...[Make] automated decisions in domains that affect material or individual rights or well-being (**e.g., finance, legal, employment, healthcare, housing, insurance, and social welfare**)”

# Accountable

- Accurate
- Legal
- Ethical

# Accountable

- Accurate
- Legal
- Ethical

# Accountable

- Accurate
- Legal
- Ethical
- Outputs should always be verified by a human

# Accountable

- Accurate
- Legal
- Ethical
- Outputs should always be verified by a human
- Don't use a LLM as a search engine

# Accountable

- Accurate
- Legal
- Ethical
- Outputs should always be verified by a human
- Don't use a LLM as a search engine
- Never use LLMs for a skill you don't already possess

# Accountable

- Accurate
- Legal
- Ethical
- Consult your legal department

# Accountable

- Accurate
- Legal
- Ethical
- Consult your legal department
- Determine if the training data were obtained legally+ethically

# Accountable

- Accurate
- Legal
- Ethical
- Consult your legal department
- Determine if the training data were obtained legally+ethically
- Edit/improve content you wrote, rather than generating from scratch

# Secure

Question I *want* to ask:

Use the [attached CSV](#) to create a scatterplot, with `year` along the x-axis and `company\_revenue` along the y-axis.

Prompt I actually use:

Write code to read in a file called “[my\\_data.csv](#)”. Create a scatterplot, with the `x\_var` variable along the x-axis and the `y\_var` variable along the y-axis.

# Secure

- (Actually) read the terms of service
- Opt-out of using your conversations to train future models, where possible

# Transparent

# Transparent

- Keep a detailed record about usage

# Transparent

- Keep a detailed record about usage
- Inform users when Gen-AI is used

# Transparent

- Keep a detailed record about usage
- Inform users when Gen-AI is used
  - Exact tool used and version
  - How it works
  - Why it's being used
  - To what extent content was reviewed or modified

# Educated

- Understanding strengths, limitations, and responsible use

# Educated

- Understanding strengths, limitations, and responsible use
- Writing effective prompts

# Educated

- Understanding strengths, limitations, and responsible use
- Writing effective prompts
- Keeping up-to-date

# This Session Was Not Generated by AI

Wednesday, August 14th | 2:40 PM - 4:00 PM PDT

- **Rafi Kurlansik:** Elevating enterprise data through open source LLMs
- **Mark Wang:** Using GitHub Copilot in R Shiny Development
- **Laura Gast:** Uniquely Human: Data Storytelling in the Age of AI
- **Alok Pattani:** Using Generative AI to Increase the Impact of Your Data Science Work

# Relevant

Always ask yourself:

Is Generative AI the  
**best** tool for the  
task?

# Relevant

Or is it just...

- Overkill
- Time-consuming
- Messy



**FASTER**  
stands for...

**Fair**  
**Accountable**  
**Secure**  
**Transparent**  
**Educated**  
**Relevant**

**1. GenAI-Fanatic**

**2. GenAI-Fatigued**

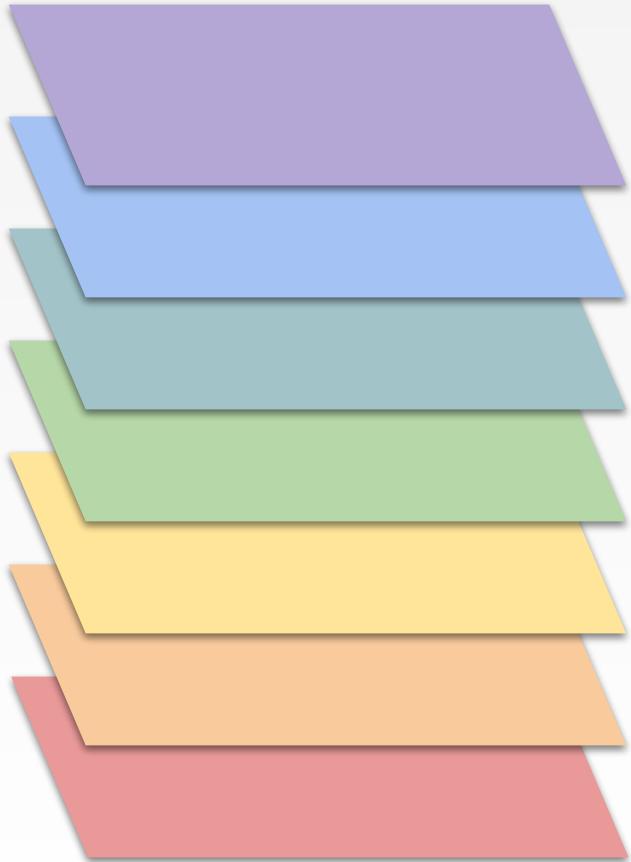
**3. GenAI-Fearful**

**1. GenAI-Fanatic**

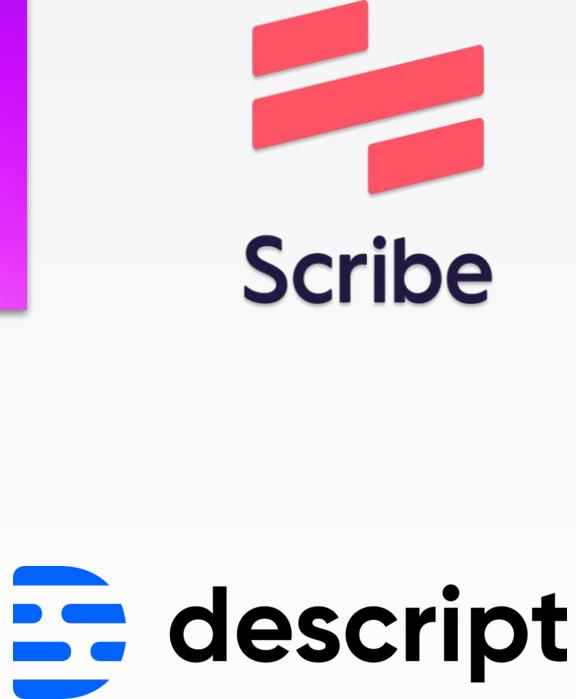
**2. GenAI-Fatigued**

**3. GenAI-Fearful**

- Be specific and detailed with prompts
- Build things up layer by layer
- Use existing mental models and your own expertise



Find  
practical,  
creative  
use cases



**“Upskill now or be left behind!”**

“The **only** course you’ll need  
to become an **AI Engineer!**”

**“Double** your salary **overnight**  
with this one simple trick!”

“The **Age** of **Generative AI...**”

**Find a  
framework for  
using  
Gen AI more  
responsibly**

**Fair  
Accountable  
Secure  
Transparent  
Educated  
Relevant**

# Thanks!

---

Have any questions?

Go to [sli.do](#) and use #posit-keynote 😊

Slides available on my GitHub



[youtube.com/c/ggnot2](https://youtube.com/c/ggnot2)



[linkedin.com/in/melissavanbussel](https://linkedin.com/in/melissavanbussel)



[github.com/melissavanbussel](https://github.com/melissavanbussel)



[melissavanbussel.com](http://melissavanbussel.com)

CREDITS: Please see next slide for full details. Slide template by **Slidesgo**, icons from **Flaticon** and images from various sources including **Pexels**, **Recraft.ai** and **Canva**.

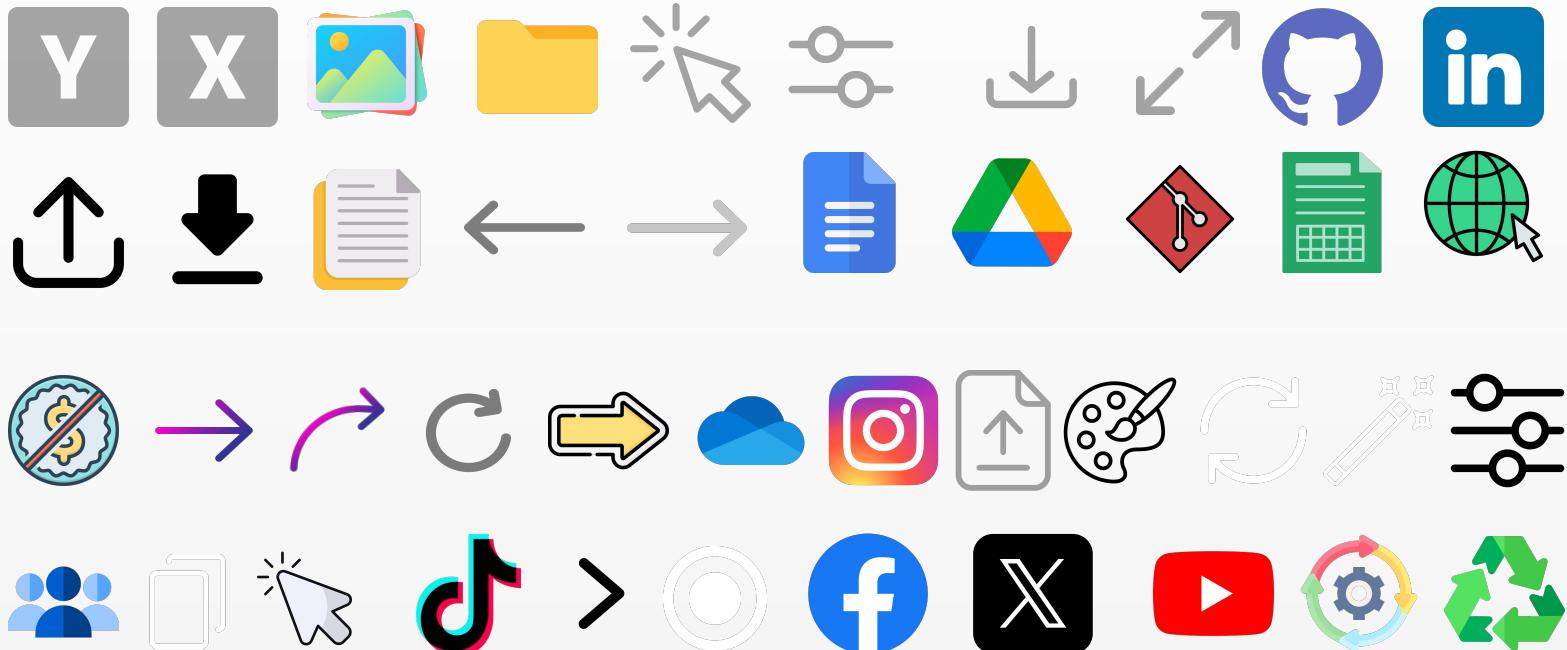


# Credits

- Google Slides Presentation theme by [Slidesgo](#)
- [Image of Rideau Canal \(license\)](#) by “John B. Codrington”; unmodified
- [Image of Ottawa Senators hockey game \(license\)](#) by “Jfvoll”; unmodified
- GPT-4o was used for brainstorming ideas for this talk. :)

Credits continue on the remaining slides.

# Icons from Flaticon

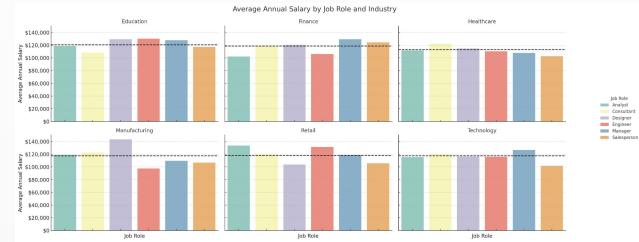
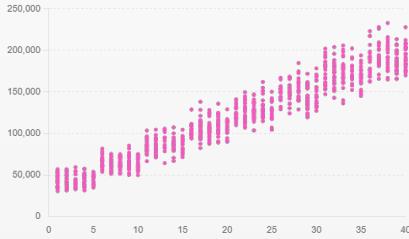
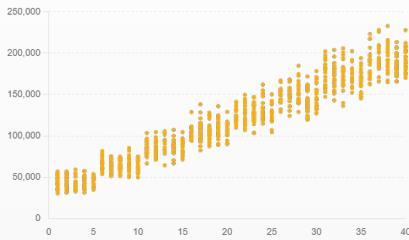


# Images from Pexels

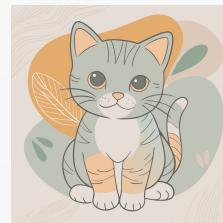


(Used the original image, as well as a modified version of the image that was modified using Photoshop)

# Images created using GPT-4o



# Images created using Recraft.ai

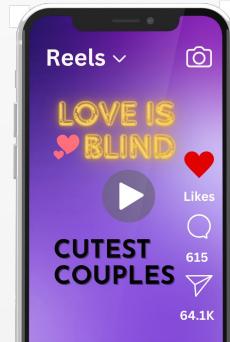
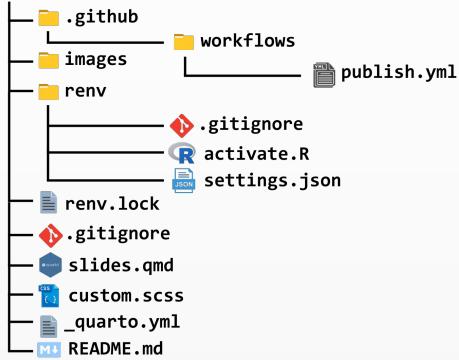


# Mockups created using Recraft.ai



(Mockups created using Recraft.ai, text/images on top of mockups created using Canva)

# Images created using Canva



# Images created using Canva

The image features a black and white silhouette of Neo from The Matrix standing on the right side. He is wearing his signature black trench coat and has his signature red and blue wireframe glasses. To his left, the words "THE MATRIX" are written in a large, blocky, and slightly distorted font. The background consists of a grid of binary digits (0s and 1s) in green, which serves as a visual representation of the digital world from the movie.



(Silhouette image obtained from [Schmector](#), licensed under [CC0 1.0](#), and modified using Canva)

A screenshot of a social media post from the R/Python User Group (RPUG). The post features a pug dog wearing a tan suit jacket and a white shirt, standing in an office cubicle. Behind the pug are large, colorful balloons spelling out "2024". The post includes the text "Jude Bashto, 9:02 AM" and "Happy new year, everyone!". Below the image are two interactive buttons: one for 17 reactions and one for 12 hearts.

Jude Bashto, 9:36 AM  
Welcome back, everyone!

Hey, Jim! It's been a while. I haven't seen you since last year!

Why does this pug have two ties?!

## Why does this pug have two ties?!