Using Quarto for Parameterized Reports

Sam Van Horne

Parameterized Reports with Quarto

- Quarto is an open-source publishing platform for producing websites, PDFs, Word documents, slide presentations (and more!)
- It was developed by Posit (formerly RStudio) to bring benefits of Rmarkdown to more users
- It enables researchers to use different programming languages (currently R, Python, Javascript, and Julia) in the same project
- It is included in the free desktop version of RStudio

More about Quarto

- Quarto uses markdown language, so it will be familiar to R markdown users (.rmd files can be converted to .qmd)
- Templates for journal articles and websites
- Supports parameterized reporting

How did I use Quarto to complete a project?

- Collaborated with PPE and DDOE on project to create district and school reports in response to SB4 and HB304
- Processed responses from completed Qualtrics surveys that included:
  - Assessment screeners for diagnosing difficulties with reading
  - Reading curriculums at each grade level
  - Percent of students at different benchmarks
- Needed one report per district (and charter) & one per elementary school (ended up being ~160 reports)

How do Word documents look?

- Headers in R become headers in Word
- When making tables with the {gt} package, table captions become table numbers
- Margins are pretty standard (but spacing between tables can be an issue)

Steps for making automating reports with Quarto

- Write your code in a Quarto (.qmd) document
- Create a data frame with two columns
  - One for your output file
  - One with the values of your parameters
- Use code chunks for your programs

Structure of a Quarto document

- YAML header
- Body of report
- Code chunks
- Text in markdown language

YAML Header

- YAML Header includes title, output format, and parameters

```yaml
---
title: "Example of Quarto Report"          # Set the title and author
author: "SVH"
format: juniper-html                       # Select the output format
code-fold: true
code-summary: "Show the code"
params:                                    # Choose default parameters in key-value pairs
  year: 2023
  district: Cape Henlopen School District
---
```

When you click the **Render** button a document will be generated that includes:

- Content authored with markdown
- Output from executable code

Parameters are referenced in code

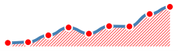
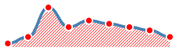
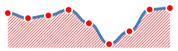
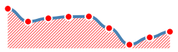
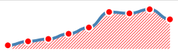
```
1 str(params)
```

```
List of 2
 $ year    : int 2023
 $ district: chr "Smyrna School District"
```

```
1 district_enrollment <- enrollment |>
2   filter(schoolcode == 0 & district == params$district & grade=="All Students" &
3            geography=="All Students" & specialdemo!="All Students"  &
4            gender=="All Students" & race =="All Students")
```

Example Output from Parameters

- At rendering, the value is what is set in header

Trend in Enrollment for Special Demographic Groups in Smyrna School District	
Demographic Group	Trend for 2015-2023
Active EL Students	
Foster Care	
Homeless	
Low-Income	
Students with Disabilities	

Percent change in EoY Enrollment from 2015 until 2023			
School District	Demographic Group	% Change in **District**	% Change in State
Smyrna School District	Active EL Students	105.77%	51.23%
Smyrna School District	Foster Care	8.33%	-9.98%
Smyrna School District	Homeless	7.23%	11.06%
Smyrna School District	Low-Income	-13.89%	-12.73%
Smyrna School District	Students with Disabilities	30.81%	45.46%

How can we automate this?

Create dataframe with parameters

First obtain the combinations of your parameters

```
district <- c("Cape Henlopen School District", "Brandywine School District", "Appoquinimink School District")
year <- 2023
districts <- data.frame(district, year)
head(districts)
```

```
                    district year
1 Cape Henlopen School District 2023
2    Brandywine School District 2023
3 Appoquinimink School District 2023
```

Make your output and list columns

```
reports <- districts |>
  dplyr::mutate(
    output_file = paste0(
      `district`, "_", gsub(" ", "", year), "_Report.html"
    ),
    execute_params = purrr::map2(
      district, year,
      \(x, y) list(district = x, year = y)
    )
  ) |>
  dplyr::select(output_file, execute_params)
head(reports)
```

```
                                output_file
1 Cape Henlopen School District_2023_Report.html
2    Brandywine School District_2023_Report.html
3 Appoquinimink School District_2023_Report.html
                      execute_params
1 Cape Henlopen School District, 2023
2    Brandywine School District, 2023
3 Appoquinimink School District, 2023
```

Publish the documents

```
reports|>
  purrr::pwalk(
    jph::quarto_render_move,
    input = here("Code/CRESP_quarto_example.qmd"),
    output_format = "html",
    output_dir = here("Code")
  )
```

Resources

- Tutorial for making parameterized reports in Quarto
- Another tutorial for different aspects of using Quarto
- Guide to using Quarto at Posit