

# LAB: Reproducible Reports with Quarto

Rian DeFaccio

## Table of contents

<b>Introduction</b>	<b>1</b>
<b>STEP 1: Build Report Template</b>	<b>2</b>
Code to Build Workload Plot . . . . .	4
Code to Build Diagnoses Plot . . . . .	5
<b>STEP 2: Build Report from R Script</b>	<b>6</b>
<b>STEP 3: Add a Parameter</b>	<b>7</b>
<b>STEP 4: Use Parameter in R Script</b>	<b>8</b>
<b>STEP 5: Render Multiple Reports Using Template</b>	<b>9</b>
<b>STEP 6: Include Conditional Content</b>	<b>10</b>
<b>Conclusion</b>	<b>11</b>

## Introduction

In this exercise you will practice building some parameterized reports using Quarto documents.

## STEP 1: Build Report Template

Begin by creating a new Quarto document. This report will summarise some information about the physician Christopher Canada.

The title of this report should be “**Physician Report**”.

Adjust any other settings you would like to for this report. The output format should be HTML.

Then, you can add the following R code to a code cell to import some data.

```
# import data and some data cleaning and formatting
clinics <- read.csv('../data/healthcare2/clinic.csv')
staff <- read.csv('../data/healthcare2/Staff.csv') %>%
  rename(StaffFirstName = FirstName, StaffLastName = LastName)
visits <- read.csv('../data/healthcare2/OutpatientVisit.csv') %>%
  left_join(staff, by = 'StaffID') %>%
  left_join(clinics, by = 'ClinicCode')
disease_map <- read.csv('../data/healthcare2/DiseaseMap.csv') %>%
  mutate(Condition = str_replace_all(Condition, "_", " ") %>%
    str_replace('Liver', 'Liver ') %>%
    str_to_title() %>%
    str_replace('With', 'with') %>%
    str_replace('Hiv', 'HIV'))

visits$VisitDateDT <- ymd(visits$VisitDate)
visits$VisitMonth <- strftime(visits$VisitDateDT, '%Y-%m')
```

Next, add the following markdown to your document:

This report is for Christopher Canada, who works at Healing Waters Hospital.

## Workload Over Time

How has Christopher's workload changed over time?

How does this compare to other physicians at his clinic?

## Common Diagnoses

What conditions does Christopher encounter most often?

Finally, we are going to build a visualization under each header. We have supplied some R code to get this done. Add code cells to your Quarto document and copy the following code under the appropriate headings.

## Code to Build Workload Plot

```
# build plot of workload over time
current_staff <- visits %>% filter(StaffID == 45)

staff_workload <- current_staff %>%
  group_by(VisitMonth) %>%
  count() %>%
  rename(Month = VisitMonth, `Number of Visits` = n)

staff_hosp <- current_staff %>% slice(1) %>% pull(ClinicCode)
hosp_workload <- visits %>%
  filter(ClinicCode == staff_hosp) %>%
  group_by(StaffID, VisitMonth) %>%
  count() %>%
  ungroup() %>%
  group_by(VisitMonth) %>%
  summarise(mean_visits = mean(n))

ggplot(staff_workload,
       aes(x = ym(Month), y = `Number of Visits`)) +
  geom_col() +
  stat_smooth(data = hosp_workload,
             aes(x = ym(VisitMonth), y = mean_visits),
             color = 'red', se = FALSE) +
  theme_minimal() +
  theme(panel.grid.minor = element_blank(),
        panel.grid.major.x = element_blank()) +
  xlab('Month') +
  ggtitle(
    'Workload over Time for Christopher Canada',
    subtitle = 'Compared to Average Physician at Healing Waters Hospital'
  )
```

## Code to Build Diagnoses Plot

```
# build plot of common diagnoses for this staff
current_staff %>%
  select(VisitID, starts_with('ICD')) %>%
  pivot_longer(
    cols = starts_with('ICD'),
    names_to = 'ICDPosition',
    values_to = 'ICD10'
  ) %>%
  left_join(disease_map, by = 'ICD10') %>%
  group_by(Condition) %>%
  count() %>%
  filter(!is.na(Condition)) %>%
  ggplot(aes(x = reorder(Condition, n), y = n)) +
  geom_col() +
  coord_flip() +
  xlab('Disease Condition') + ylab('Number of Diagnoses') +
  theme_minimal() +
  ggtitle('All Diagnoses for Christopher Canada') +
  scale_y_continuous(label = scales::comma)
```

Finally, render your report and see how it looks! Make any adjustments based on how the rendered report looks.

- Do you want the code to be shown in your rendered report?
- Do you want to adjust any theme elements in the plots?

## STEP 2: Build Report from R Script

Now take the report from step 1 and render it using an R script. The name of the HTML file should be “Christopher Canada.html”.

You will need to create a new R script in this step, and render your report by running the code in the R script.

one of the first steps in your R script should be to change the working directory to be the *labs/* folder.

With the **quarto** package we are currently limited to rendering the HTML document to the same directory as the source *.qmd* file. However, we can use R to move the rendered HTML file to another location afterwards. Add and adjust the following code to your R script after your rendering code:

```
file.rename('this_file.html', 'my_stuff/this_file.html')
```

## STEP 3: Add a Parameter

Now let's use a Quarto parameter inside of our *.qmd* document.

Add a parameter to the YAML section called **StaffID**. Set the default value of this parameter to be the number 45.

Add the following R code in an early code cell. This will generate variables that you will use further down in the report.

```
staff_first_name <- staff %>%  
  filter(StaffID == params$StaffID) %>%  
  pull(StaffFirstName)  
staff_name <- staff %>%  
  filter(StaffID == params$StaffID) %>%  
  mutate(FullName = paste(StaffFirstName, StaffLastName)) %>%  
  pull(FullName)  
staff_hosp <- visits %>%  
  filter(StaffID == params$StaffID) %>%  
  slice(1) %>%  
  pull(ClinicName)
```

Adjust your report to utilize the parameter **StaffID** so that you never explicitly mention the number 45, the name “Christopher Canada”, or the hospital name in the document. We want to report to generate dynamically based entirely on the value of this parameter.

Finally, render your report to make sure everything still looks as expected.

## STEP 4: Use Parameter in R Script

This time use your parameter to render a version of your report from your R script. See if you can generate a version of your report for Jordan Smith.

Copy the following R code into your R script.

```
staff <- read.csv('../data/healthcare2/Staff.csv')
first_name <- 'Jordan'
last_name <- 'Smith'
full_name <- paste(first_name, last_name)
file_name <- paste0(full_name, '.html')
jordan_id <- staff %>%
  filter(FirstName == first_name & LastName == last_name) %>%
  pull(StaffID)
```

Finally, make sure you achieve the two following points:

- All markdown and data visualizations in the report should relate to Jordan Smith.
- The name of the file should be “*Jordan Smith.html*”.

As before, move this report into the *labs/my\_stuff/* folder after you render it.



## STEP 5: Render Multiple Reports Using Template

Your next task is to programmatically render reports for the first 10 physicians in the staff file. Use the following R code in your R script to get yourself started with this section:

```
# first, we have to find the first 10 physicians!
physicians <- staff %>%
  filter(StaffType == 'Physician') %>%
  slice(1:10)

# now go through most the same steps as in STEP 4
first_names <- physicians$FirstName
last_names  <- physicians$LastName
full_names  <- paste(first_names, last_names)
file_names  <- paste0(full_names, '.html')
IDs         <- physicians %>% pull(StaffID)

# now generate a report for each physician
for (i in 1:length(IDs)) {
  quarto::quarto_render(
    input          = _____,
    output_format  = 'html',
    output_file    = _____,
    execute_params = list(StaffID = _____)
  )
  # move the HTML file to the my_stuff folder
  file.rename(_____)
}
```

## STEP 6: Include Conditional Content

Finally, your last task is to include some conditional content. Add a new section in your R script for this step.

Let's look only at nurses this time. Your first step will be to alter your R script to get the first 10 nurses instead of the first ten physicians.

Then, add some conditional content. Nurses can receive either hourly pay or a salary. Create two sections at the bottom of your report that tell how much the nurse earns. Only one of the sections should show, depending on which type of pay the nurse receives.

Use the following code to determine the pay type for each nurse. This will go into the *.qmd* file before your conditional content.

```
# get the pay type for the staff
pay_type <- staff %>%
  filter(StaffID == params$StaffID) %>%
  pull(PayType)
```

You can use the following code in the *.qmd* file to generate conditional content for nurses paid hourly. Repeat and alter this code for nurses that are paid by salary.

```
`r if (pay_type == "Salary") " ::: {.content-hidden}"`

## Hourly Pay Details

-----
calculate hourly rate here in an R code cell
-----

`r staff_name` is an hourly employee. Their hourly rate is $`r hourly_rate`.

`r if (pay_type == "Salary") " ::: "`
```

And here is the R code to calculate the rate.

```
hourly_rate <- staff %>%
  filter(StaffID == params$StaffID) %>%
  pull(HourlyRate) %>%
  round(2) %>%
  format(nsmall=2)
```

Also don't forget to change the title of your report to something like "Nurse Report".

## Conclusion

That's it! By this point you are able to do the following:

- Build a template Quarto document.
- Use parameters and an R control script to re-use your template over and over again.
- Use conditional content to generate different versions of your report for different types of audiences.