In this post we generate a PowerPoint report from a Shiny app. This is Part 2 from the 3-post series about generating reports with Shiny. Part 1 covered [Excel Report Generation with Shiny](https://www.tychobra.com/posts/2019_07_22_excel_report_generation/). See part 1 for a quick intro and description of [the Shiny app](https://tychobra.shinyapps.io/claims-dashboard/) that can generate Excel, PowerPoint, and PDF reports.

**PowerPoint Report**

These are the slides from the PowerPoint report in our Shiny app:

[Click here](https://github.com/Tychobra/shiny-insurance-examples/blob/master/basic-insurer-dashboard/example%20reports/claims-report-as-of-2019-06-24.pptx) to download the PowerPoint file from GitHub.

For code guidance , please follow: <https://github.com/Tychobra/shiny-insurance-examples/find/master>

**Creating a Basic PowerPoint with R**

We use the [officer](https://github.com/davidgohel/officer) R package to create the PowerPoint. officer connects R to Microsoft Word and PowerPoint (similar to how [openxlsx](https://github.com/awalker89/openxlsx) connects R to Excel). We use the [flextable](https://github.com/davidgohel/flextable) package to format our PowerPoint tables.

To begin creating the PowerPoint, we assign a blank PowerPoint to the ppt\_report variable:

library(officer)

ppt\_report <- read\_pptx()

We can now create a PowerPoint file from our ppt\_report object by using the generic print() function and setting target = "" as below:

print(ppt\_report, target = "my\_ppt\_file.pptx")

# run `?print.rpptx` for more info on the above function

The PowerPoint we created above is completely empty, so it’s not terribly interesting. Let’s add a basic title page:

library(dplyr)

ppt\_report <- ppt\_report %>%

add\_slide(

layout = "Title Slide",

master = "Office Theme"

) %>%

ph\_with\_text(

type = "ctrTitle",

str = c(

"Example Client Name",

"Workers' Compensation Claims Report"

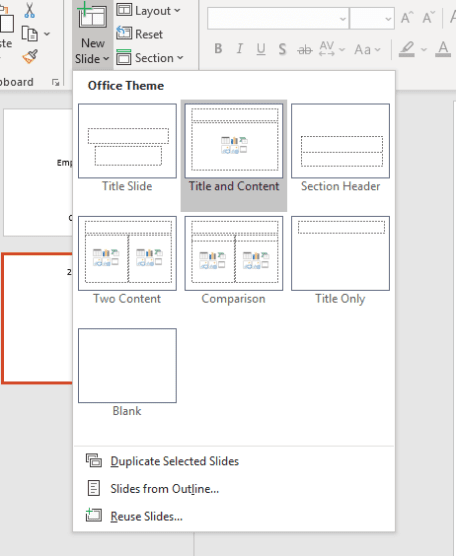
)

)

print(ppt\_report, target = "my\_ppt\_file.pptx")

The new version of our PowerPoint now has a title page.

You can specify the slide theme with the master argument to add\_slide() (“Office Theme” in our above example). All the layouts available with PowerPoint can be set with the layout argument. e.g. you can make slides with all the following layouts:



**Using Flextables**

officer can add tables, but it does not have many table formatting options. For more customized tables we use the flextable package.

Check out the [code here](https://github.com/Tychobra/shiny-insurance-examples/blob/abc7811dea9857c16b610da9f21b834888372351/basic-insurer-dashboard/server/04-report-srv/04-powerpoint-srv.R#L15) to see the full formatting of the PowerPoint in our Shiny app.

**Demo App**

Below is a complete but very simple Shiny app that can generate a PowerPoint report. Simply run the following directly in your R console to run the app yourself!

library(shiny)

library(officer)

library(flextable)

library(dplyr)

my\_table <- data.frame(

Name = letters[1:4],

Age = seq(20, 26, 2),

Occupation = LETTERS[15:18],

Income = c(50000, 20000, 30000, 45000)

)

ui <- fluidRow(

column(

width = 12,

align = "center",

tableOutput("data"),

br(),

downloadButton("download\_powerpoint", "Download Data to PowerPoint")

)

)

server <- function(input, output) {

output$data <- renderTable({

my\_table

})

output$download\_powerpoint <- downloadHandler(

filename = function() {

"employee\_data.pptx"

},

content = function(file) {

flextable\_prep <- flextable(my\_table) %>%

colformat\_num(col\_keys = c("Age", "Income"), digits = 0) %>%

width(width = 1.25) %>%

height\_all(height = 0.35) %>%

theme\_zebra() %>%

align(align = "center", part = "all")

example\_pp <- read\_pptx() %>%

add\_slide(layout = "Title Slide", master = "Office Theme") %>%

ph\_with\_text(

type = "ctrTitle",

str = "Employee Data"

) %>%

ph\_with(

location = ph\_location\_type(type = "subTitle"),

value = "Company 2019 Report"

) %>%

add\_slide(layout = "Title and Content", master = "Office Theme") %>%

ph\_with\_text(

type = "title",

str = "2019 Data"

) %>%

ph\_with\_flextable\_at(

value = flextable\_prep,

left = 2.5,

top = 2

)

print(example\_pp, target = file)

}

)

}

shinyApp(ui, server)

Which will download this basic PowerPoint:

