

M55-Query & Report

learningSpoonsR@gmail.com



yaml Header

```
---  
title: "M55 - Query & Report"  
output:  
  flexdashboard::flex_dashboard:  
    source_code: embed  
    theme: journal  
runtime: shiny  
---
```

Setup chunk

```
```{r global, include=FALSE}
library(readxl)
library(dplyr)
library(shiny)
library(ggplot2)
library(DT)
library(tidyr)
library(data.table)
library(rmarkdown)
library(stringr)
dataset <- read_excel("input_data/retail.xlsx")
dataset <- dataset %>%
 select(`Order ID`, `Order Date`, Category, `Sub-Category`,
 `Product Name`, Quantity, Profit)
dataset$`Order Date` <- as.Date(dataset$`Order Date`)
```
```

Sidebar (1)

```

Sidebar {.sidebar}
=====

```{r}
dateRangeInput(inputId = "dt_rng", label = "Date Range",
 min = min(dataset$`Order Date`), start = max(dataset$`Order Date`)-14,
 max = max(dataset$`Order Date`), end = max(dataset$`Order Date`))
radioButtons(inputId = "class", label = "Category",
 choices = c("Furniture", "Office Supplies", "Technology"))
downloadButton("downloadData", label = "Save to CSV")
downloadHandler(
 filename = function() {
 "dataset_filtered.csv"
 },
 content = function(file) {
 write.csv(datasetFiltered(), file)
 }
)

```

## Sidebar (2)

```
actionButton("render_docx", "Report using csv")
observeEvent(input$render_docx, {
 rmarkdown::render(
 input = "M55-summary_docx.Rmd",
 output_file =
 paste0("summary_report_",
 Sys.time() %>%
 strftime("%Y-%m-%d %H:%M:%OS") %>%
 str_replace_all(":", "-") %>%
 str_replace_all(" ", "-"),
 ".docx"),
 encoding = "UTF-8")
})

```

## Main page

Main

=====

```
```{r}
datasetFiltered <- reactive({
  dataset %>%
    filter(`Order Date` >= input$dt_rng[1] & `Order Date` <= input$dt_rng[2]) %>%
    filter(Category == input$class)
})
# Display the filtered table
DT::renderDataTable({ datasetFiltered() })
```
```