

# ~1 Pager: Dynamic Tables with Reactable

## Why?

Tables may not get as much attention as other data visualizations but two things: readability and precision, make them incredibly powerful. Thanks to packages like reactable, we can make tables interactive as well, which makes them more compelling to users and invites data exploration. Let's work through how you can use R and the reactable package to do things that in the past you might have put in a spreadsheet then emailed around.

## How? With Reactable (and Penguins!)

- In the RStudio IDE, click on File > New File > R Script
- Paste the text below inside that script to build a table using some data on penguins; we'll call that data "tux" and look at it in traditional way as a table in the R console.

```
library(reactable)
library(palmerpenguins)
library(dplyr)
library(sparklines)

tux <- penguins %>%
  select(species, sex, body_mass_g, flipper_length_mm) %>%
  rename(Species = species, Sex = sex, Mass = body_mass_g, Flippers = flipper_length_mm) %>%
  filter(complete.cases(.))
tux
```

Now let's apply the reactable function to tux and give it superpowers.

```
reactable(tux)
```

First, let's use our new powers to control the aesthetics of how the table *looks*. The options we've selected here such as `borderless = TRUE` are fairly self-explanatory, and there is much more to explore by using `?reactable`.

```
reactable(tux,
  borderless = TRUE,
  highlight = TRUE,
  fullWidth = FALSE,
  resizable = TRUE,
  defaultColDef = colDef(
    align = "center",
    headerStyle = list(background = "lightgray")),
  columns = list(
    Species = colDef(minWidth = 180)
  ),
)
```

Second, let's use reactable to give us tools to explore the data: sortable, searchable, filterable, conditional formatting, summary statistics, and so forth.

```
reactable(tux,
  defaultSorted = c("Mass"),
  filterable = TRUE,
  searchable = TRUE,
  borderless = TRUE,
  highlight = TRUE,
  resizable = TRUE,
  defaultColDef = colDef(
    align = "center",
    headerStyle = list(background = "lightgray")),
  columns = list(
    Species = colDef(minWidth = 140, footer="Mean"),
    Flippers = colDef(style=function(value) {
      if (value > 190) clr <- "red" else clr <- "black"
      list(color=clr, fontWeight="bold" )},
    footer = function(value) sprintf("%.0f", mean(value, na.rm=TRUE)))
  )
)
```

Third, let's group rows together and make them expandable. Note we can even include plots!

```
reactable(tux, groupBy = "Species",
  columns = list(
    Flippers = colDef(aggregate = "mean", format = colFormat(suffix = " mm", digits = 0)),
    Mass = colDef(aggregate = "mean", format = colFormat(suffix = " g", digits =0)),
    Sex = colDef(aggregate = "frequency")),
  borderless = TRUE,
  highlight = TRUE,
  defaultColDef = colDef(
    align = "center",
    headerStyle = list(background = "lightgray"),
    footer = function(values) {
      if (!is.numeric(values)) return()
      sparkline(values, chart_type = "box") }
  )
)
```

Lastly, how can we scale this? As with ggplot, we can create a **theme** one time and then apply that theme again and again to all our reactables.

```
options(reactable.theme = reactableTheme( # here is a dark theme
  color = "hsl(233, 9%, 87%)",
  backgroundColor = "hsl(233, 9%, 19%)",
  borderColor = "hsl(233, 9%, 22%)",
  stripedColor = "hsl(233, 12%, 22%)",
  highlightColor = "hsl(233, 12%, 24%)",
  inputStyle = list(backgroundColor = "hsl(233, 9%, 25%)"),
  selectStyle = list(backgroundColor = "hsl(233, 9%, 25%)"),
  pageButtonHoverStyle = list(backgroundColor = "hsl(233, 9%, 25%)"),
  pageButtonActiveStyle = list(backgroundColor = "hsl(233, 9%, 28%)")
)
```

```

))

reactable(tux,
  borderless = TRUE,
  highlight = TRUE,
  defaultColDef = colDef(
    align = "center"),
  columns = list(
    Species = colDef(minWidth = 180)
  ),
)

options(reactable.theme = NULL) # But now let's go back to the default theme

reactable(tux,
  borderless = TRUE,
  highlight = TRUE,
  fullWidth = FALSE,
  defaultColDef = colDef(
    align = "center"),
  columns = list(
    Species = colDef(minWidth = 180)
  ),
)

```

## FAQ's

- Where can I learn more? [Here](#)
- Are there some fun examples to check out? Yes, this [one](#) remakes a published table from 538.com about the Women's World Cup.
- Who can I talk to about this? [brian.law@rstudio.com](mailto:brian.law@rstudio.com)