



Advanced configuration examples

Source: [vignettes/extra/advanced-configuration.Rmd](#)

```
library(apexcharter)
library(dplyr)
```

Here some advanced configuration examples to use the full potential of [ApexCharts](#).

Bar chart

This example is taken from [{hrbrthemes}](#) readme, it use `mpg` dataset from [{ggplot2}](#).

Transform data and initialize the chart :

```
data("mpg", package = "ggplot2")

bars <- count(mpg, class) %>%
  mutate(pct = n/sum(n)) %>%
  apex(mapping = aes(class, pct), type = "column")
```

Change color used :

```
bars <- bars %>%
  ax_colors("#617a89")
```

Modify the y-axis and add a percent symbol after the labels :

```
bars <- bars %>%
  ax_yaxis(
    title = list(text = "Weight (tons)"),
    labels = list(
      formatter = format_num("%.0%")
    ),
    tickAmount = 6,
    max = 0.3
  )
```

Add a title to the x-axis :

On this page

[Bar chart](#)

[Lines](#)

[Scatter plot](#)

[Heatmap](#)

apexcharter 0.4.4.9800 [Get started](#) [Reference](#) [Changelog](#)

```
title = list(text = fuel_efficiency (mpg) )  
)
```

Modify the tooltip to display “Percentage” instead of the variable name “pct” :

```
bars <- bars %>%  
  ax_tooltip(  
    y = list(  
      title = list(  
        formatter = JS("function() {return 'Percentage';}")  
      )  
    )  
  )  
)
```

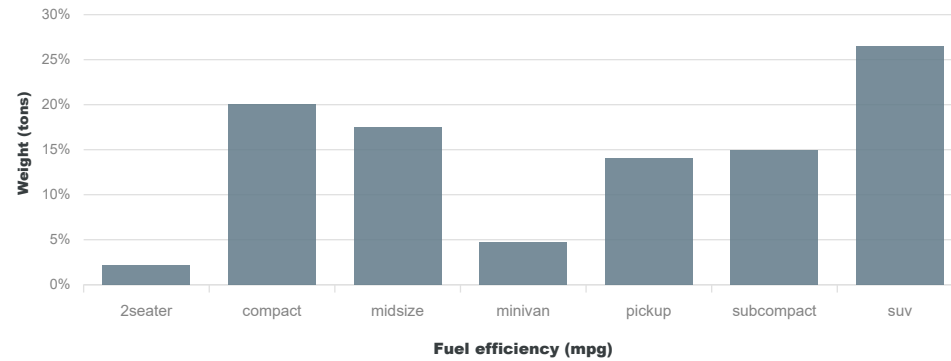
Add title and subtitle and format them :

```
bars <- bars %>%  
  ax_labs(  
    title = "Seminal ggplot2 column chart example with percents",  
    subtitle = "Example taken from {hrbrthemes} readme"  
  ) %>%  
  ax_title(  
    style = list(fontSize = "22px")  
  ) %>%  
  ax_subtitle(  
    style = list(fontSize = "16px", color = "#BDBDBD")  
  )
```

Final result looks like :

```
bars
```

apexcharter 0.4.4.9800 [Get started](#) [Reference](#) [Changelog](#)



► [View full code](#)

Lines

Dataset used is from the UNHCR (The UN Refugee Agency) and contains data about UNHCR's populations of concern summarised by continent of origin.

Transform data and initialize the chart :

```
data("unhcr_ts")

lines <- unhcr_ts %>%
  filter(population_type == "Refugees (incl. refugee-like situations)") %>%
  mutate(date = as.Date(paste0(year, "-01-01"))) %>%
  apex(aes(date, n, group = continent_origin), type = "line")
#> Warning in make_series(mapdata, mapping, type, serie_name): apex: all groups
#> must have same length! You can use `tidyr::complete` for this.
```

Put the legend at the bottom :

```
lines <- lines %>%
  ax_legend(position = "bottom")
```

Change the width of the lines :



Change the colors (Viridis palette) :

```
lines <- lines %>%
  ax_colors("#440154", "#414487", "#2A788E",
            "#22A884", "#7AD151", "#FDE725")
```

Data are in million, in the y-axis we divide by 1e6 to limit the number of digits :

```
lines <- lines %>%
  ax_yaxis(
    labels = list(
      formatter = JS("function(val) {return (val/1e6).toFixed(0);}")
    ),
    title = list(text = "Number of refugees (in million)")
  )
```

Only display the years in the x-axis labels :

```
lines <- lines %>%
  ax_xaxis(labels = list(format = "yyyy"))
```

Same in tooltip, and a thousand separator in the value displayed :

```
lines <- lines %>%
  ax_tooltip(
    x = list(format = "yyyy"),
    y = list(
      formatter = JS(
        # thousand separator in javascript
        "function(value) {return value.toString().replace(/\\B(?=(\\d{3})+(?!\\d))/g, \"
      )
    )
  )
```

Add an annotation to the chart to identify the [Great Lakes refugee crisis](#) in 1994 :

```
lines <- lines %>%
  ax_annotations(
    points = list(
```

apexcharter 0.4.4.9800 [Get started](#) [Reference](#) [Changelog](#)



```
label = list(text = "Great Lakes refugee crisis", offsetY = 0),
marker = list(size = 6)
)
)
)
```

Add title and subtitle and format them :

```
lines <- lines %>%
  ax_labs(
    title = "Continent of origin for refugees population",
    subtitle = "Data from the UN Refugee Agency"
  ) %>%
  ax_title(
    style = list(fontSize = "22px")
  ) %>%
  ax_subtitle(
    style = list(fontSize = "16px", color = "#BDBDBD")
  )
)
```

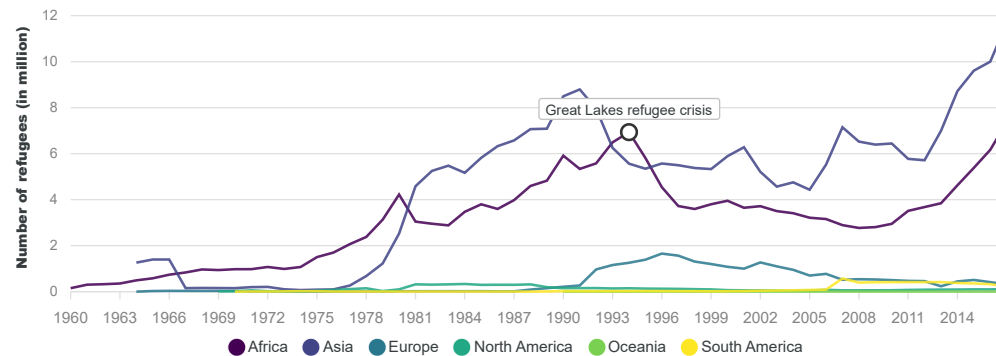
Final result looks like :

lines

Continent of origin for refugees population



Data from the UN Refugee Agency



► View full code



Scatter plot

Dataset used is from `{gapminder}`.

Transform data and initialize the chart :

```
data("gapminder", package = "gapminder")

scatter <- gapminder %>%
  filter(year == 2007) %>%
  mutate(
    gdpPercap = log(gdpPercap),
    pop = sqrt(pop / pi) / 1500
  ) %>%
  apex(
    mapping = aes(gdpPercap,
                  lifeExp,
                  z = pop,
                  group = continent,
                  label = country),
    type = "scatter",
    height = "500px"
  )
```

Enable zoom on both axis :

```
scatter <- scatter %>%
  ax_chart(zoom = list(
    enabled = TRUE, type = "xy"
  ))
```

Show y-axis border and ticks, no decimals in labels :

```
scatter <- scatter %>%
  ax_yaxis(
    decimalsInFloat = 0,
    axisBorder = list(show = TRUE),
    axisTicks = list(show = TRUE),
    title = list(text = "life expectancy at birth (in years)")
  )
```

apexcharter 0.4.4.9800 Get started Reference

Changelog



regardless there's a lot of decimals).

```
scatter <- scatter %>%
  ax_xaxis(
    tickAmount = 8,
    labels = list(
      formatter = JS("function(val) {return val.toFixed(2);}"),
    ),
    tooltip = list(enabled = FALSE),
    title = list(text = "GDP per capita (log-scale)")
  )
```

Display vertical grid lines (on the x-axis, those on the y-axis are enabled by default) :

```
scatter <- scatter %>%
  ax_grid(xaxis = list(lines = list(show = TRUE)))
```

Legend on the right and slightly offset downwards :

```
scatter <- scatter %>%
  ax_legend(position = "right", offsetY = 70)
```

Add a custom tooltip with an HTML string, data used in aesthetic can be accessed in JavaScript with `w.config.series[seriesIndex].data[dataPointIndex].x` (x is the variable `gdpPercap`), Or `w.config.series[seriesIndex].data[dataPointIndex].label` (the country), it's possible to use custom aesthetics to include more data in the chart configuration script.

```
scatter <- scatter %>%
  ax_tooltip(custom = JS(paste(
    "function({ series, seriesIndex, dataPointIndex, w }) {",
    "console.log(w); return (",
    "'<div>' +",
    "",
    "'<div class = \"apexcharts-tooltip-title\">' +",
    "w.config.series[seriesIndex].data[dataPointIndex].label",
    "+ '</div>' +",
    "'<div style = \"padding: 5px;\">' +",
    "'<div class = \"apexcharts-tooltip-y-group\">' +",
    "",
    "'<span class = \"apexcharts-tooltip-text-label\">' +",
```



```

    "Math.round(Math.pow(w.config.series[seriesIndex].data[dataPointIndex].z * 1500, 2)
    toString().replace(/\\B(?=(\\d{3})+(?!\\d))/g, "\\,\\") +",
    "'</span>' +",

    "'</br>' +",

    "'<span class = \\\"apexcharts-tooltip-text-label\\\">' +",
    "'GDP per capita: ' +",
    "'</span>' +",
    "'<span class = \\\"apexcharts-tooltip-text-value\\\">' +",
    "Math.round(Math.exp(w.config.series[seriesIndex].data[dataPointIndex].x)) +",
    "'</span>' +",

    "'</br>' +",

    "'<span class = \\\"apexcharts-tooltip-text-label\\\">' +",
    "'Life expectancy: ' +",
    "'</span>' +",
    "'<span class = \\\"apexcharts-tooltip-text-value\\\">' +",
    "w.config.series[seriesIndex].data[dataPointIndex].y +",
    "'</span>' +",

    "'</div>' +",
    "'</div>' +",

    "'</div>'",
    ");",
    "}", sep = "\\n"
  )))

```

Add title and subtitle and format them :

```

scatter <- scatter %>%
  ax_labs(
    title = "Life expectancy, GDP and population",
    subtitle = "gapminder dataset from {gapminder}"
  ) %>%
  ax_title(
    style = list(fontSize = "22px")
  ) %>%
  ax_subtitle(

```

apexcharter 0.4.4.9800 Get started Reference Changelog

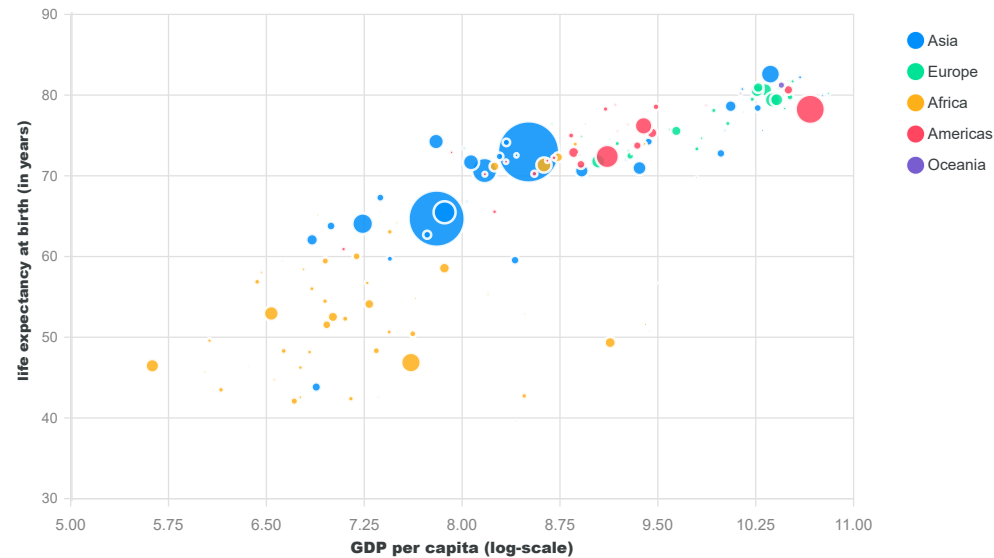


Final result looks like :

scatter

Life expectancy, GDP and population

gapminder dataset from {gapminder}



► View full code

Heatmap

This is an adaption of this fancy [{highcharter} example](#), based from this [WSJ visualization](#).

```
data("vaccines", package = "highcharter")

heatmap <- apex(
  vaccines,
  aes(year, state, fill = count),
  type = "heatmap",
```

apexcharter 0.4.4.9800 [Get started](#) [Reference](#)[Changelog](#)

Remove the animations (little slow otherwise) :

```
heatmap <- heatmap %>%  
  ax_chart(animations = list(enabled = FALSE))
```

Remove values displayed in the heatmap :

```
heatmap <- heatmap %>%  
  ax_dataLabels(enabled = FALSE)
```

Remove space between squared of the heatmap :

```
heatmap <- heatmap %>%  
  ax_stroke(width = 0)
```

That's not possible to make a continuous scale in the legend (like with highcharter), so we use breakpoints :

```
heatmap <- heatmap %>%  
  ax_plotOptions(  
    heatmap = heatmap_opts(  
      radius = 0,  
      enableShades = FALSE,  
      colorScale = list(  
        ranges = list(  
          list(  
            from = 0,  
            to = 0.001,  
            name = "Missing",  
            color = "#FFF"  
          ),  
          list(  
            from = 0.001,  
            to = 4,  
            name = "low",  
            # color = "#000004"  
            color = "#FDE725"  
          ),  
          list(  
            from = 4,  
            to = 70,
```



```

    ),
    list(
      from = 70,
      to = 290,
      name = "mid-high",
      # color = "#ED6925",
      color = "#31688E"
    ),
    list(
      from = 290,
      to = 3000,
      name = "high",
      # color = "#FCFFA4",
      color = "#440154"
    )
  )
)
)
)
)

```

Missing values are colored by default, above we set them to be displayed in white, and now we hide the corresponding legend :

```

heatmap <- heatmap %>%
  ax_legend(
    formatter = JS(
      "function(seriesName, opts) {
        if (seriesName == 'Missing') return null; else return seriesName;
      }"
    ),
    offsetY = -15
  )

```

Set size of the y-axis labels :

```

heatmap <- heatmap %>%
  ax_yaxis(
    labels = list(
      style = list(fontSize = "8px")
    )
  )
)

```

apexcharter 0.4.4.9800 [Get started](#) [Reference](#) [Changelog](#)



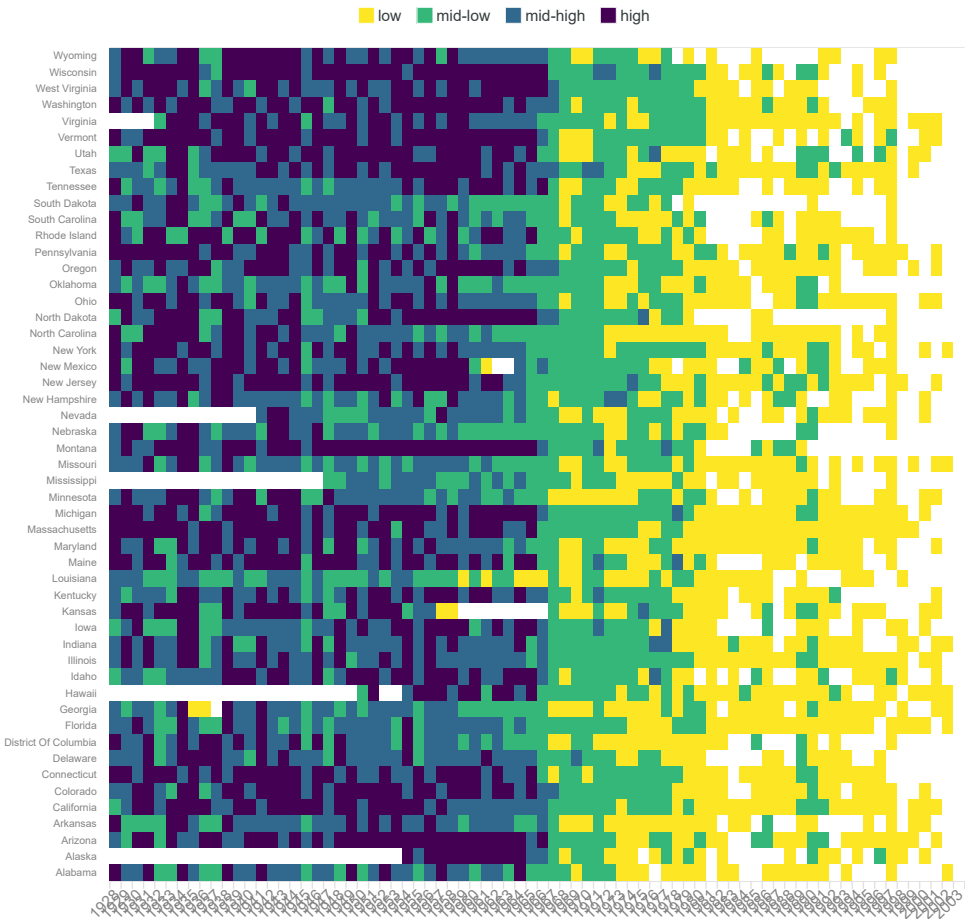
```
heatmap <- heatmap %>%
  ax_annotations(
    xaxis = list(
      list(
        x = 1963, x2 = 1963.1,
        strokeDashArray = 0,
        opacity = 1,
        borderColor = "firebrick",
        fillColor = "firebrick",
        label = list(
          borderColor = "firebrick",
          style = list(color = "#FFF", background = "firebrick"),
          text = "Vaccine Introduced",
          orientation = "horizontal",
          position = "bottom",
          offsetY = 0
        )
      )
    )
  )
)
```

As usual, add title and subtitle and format them :

```
heatmap <- heatmap %>%
  ax_labs(
    title = "Infectious Diseases and Vaccines",
    subtitle = "vaccines dataset from {highcharter}"
  ) %>%
  ax_title(
    style = list(fontSize = "22px")
  ) %>%
  ax_subtitle(
    style = list(fontSize = "16px", color = "#BDBDBD")
  )
```

Final result looks like :

heatmap



► View full code

Developed by  [Victor Perrier](#)  [Fanny Meyer](#)

Site built with [pkgdown](#) 2.2.0.