Data visualization with highcharter

R-Ladies Paris Meetup

María Paula Caldas

2017-11-09

Intro

Hello, I am María Paula Caldas 👋

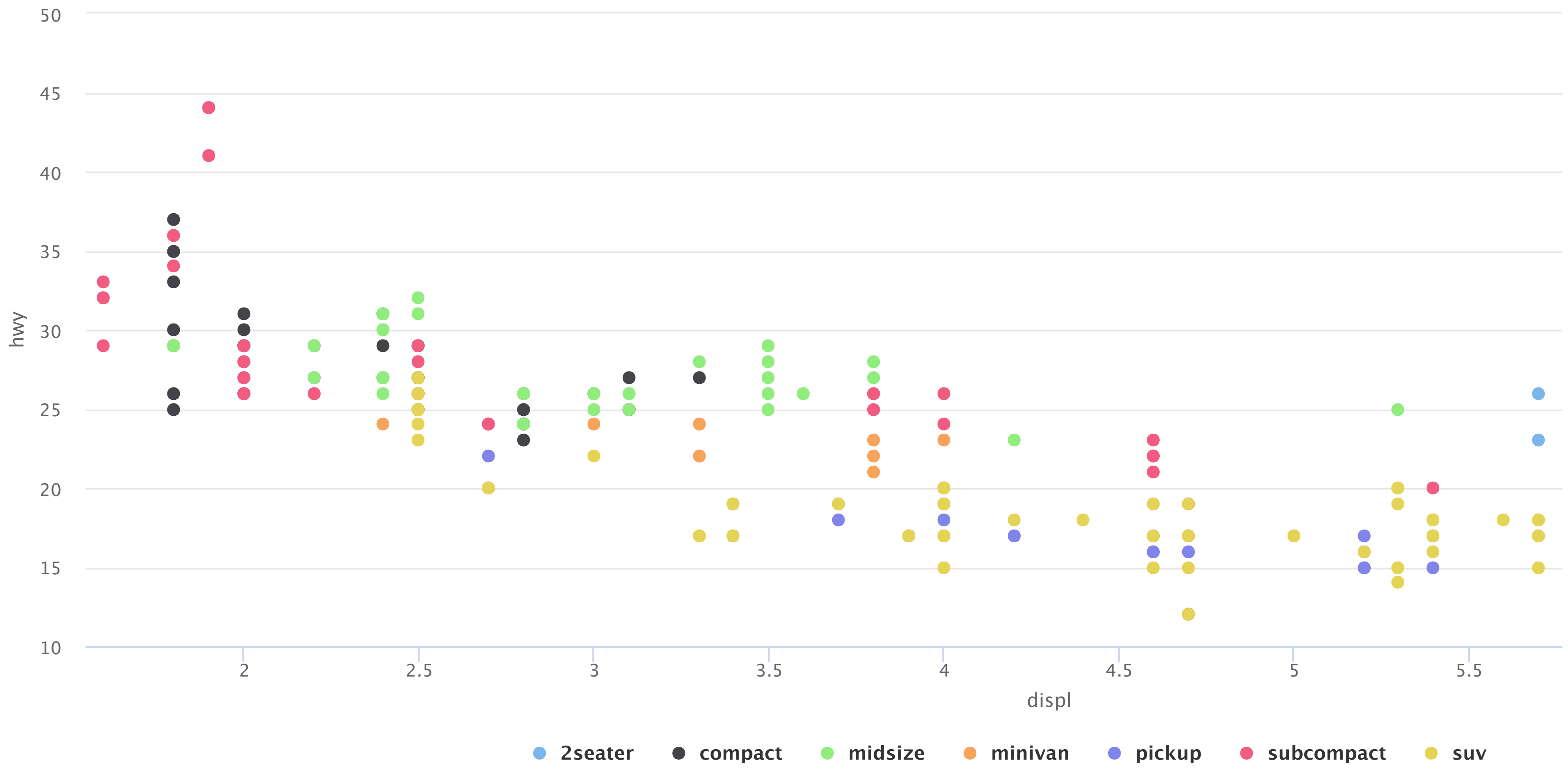
I am an economist, currently working as a consultant at Deloitte. My work involves a healthy mix of **economics**, **data analysis** and **data visualization**.

For one of my cooler projects, I got to explore the `highcharter` package. My goal today is to introduce you to the package and to share with you some of the lessons I learned.

Please don't hesitate to ask questions 🙋

Interactive visualizations

Interactive vs static visualisations



Packages for interactive visualizations

Most R packages for interactive data visualizations are powered by `htmlwidgets`: a R package that provides a framework for creating R bindings to JavaScript libraries.

Here are some of the packages you may have heard about:

- `plotly`
- `DiagrammeR`
- `leaflet`
- `networkD3`
- `highcharter`

Highcharts & `highcharter`

`highcharter` is a R wrapper for the **Highcharts** JavaScript charting library and its modules.

Some of the nice features of the package include:

- `hchart()` function
- Layering syntax + use of `magrittr` pipes (`%>%`)
- 10+ built-in themes
- Support for **Highstock** and **Highmaps** charts
- Implementation of other Highcharts plug-ins

Please keep in mind that **Highcharts** is a software product that is *not* free for commercial and Governmental use.

Building charts with `highcharter`

`highcharter` VS `ggplot2`

- Both allow you to create highly customizable visualizations
- Both build plots by layers
  - `ggplot2` with `+`
  - `highcharter` with `%>%`
- `ggplot2` is useful both for **exploration** and **presentation**
- `highcharter` is better suited for dynamic visualizations in **Shiny** or **RMarkdown** documents
- You can access the majority of the documentation for `ggplot2` directly via R/R Studio, you will have to look through the Highcharts API for details on function parameters
  - Joshua Kunst (package maintainer) more than makes up for this with his detailed package website, vignettes and blog posts

How do you start a plot?

There are two ways to start a plot:

- `hchart()`, similarly to ggplot2's `qplot()`, is meant for making quick charts with minimal information from the user
- `highcharter()` followed by either one of two functions:
    - `hc_add_series()`
    - `hc_add_series_list()`
- Both `hchart()` and `hc_add_series()` accept an aesthetic mapping function *à la ggplot2*: `hcaes()`

An example

I am going to recreate a plot that appeared in one of FiveThirtyEight's articles: Comic Books Are Still Made By Men, For Men And About Men. The data are available at Github.

Here is a glimpse of the data that we will use in the next charts:

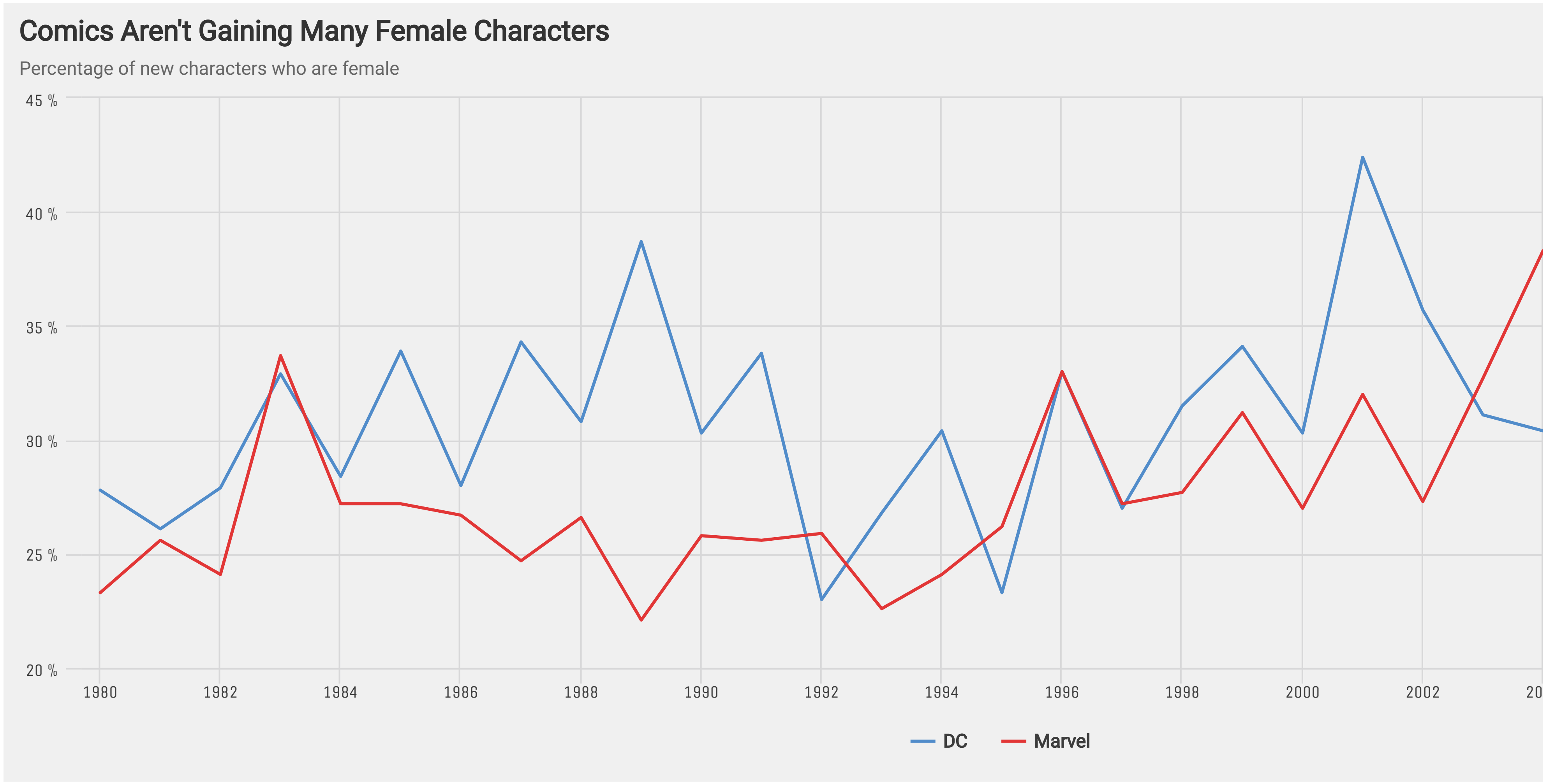
```
new_fem_per_year
```

```
## # A tibble: 64 x 5
##    comic  year               sex     n share_gender
##    <chr> <int>             <chr> <int>        <dbl>
##  1    DC  1980 Female Characters    10         27.8
##  2    DC  1981 Female Characters    31         26.1
##  3    DC  1982 Female Characters    31         27.9
##  4    DC  1983 Female Characters    53         32.9
##  5    DC  1984 Female Characters    40         28.4
##  6    DC  1985 Female Characters    39         33.9
##  7    DC  1986 Female Characters    37         28.0
##  8    DC  1987 Female Characters    87         34.3
##  9    DC  1988 Female Characters    88         30.8
## 10    DC  1989 Female Characters   103         38.7
## # ... with 54 more rows
```

hchart()

```r
hc1 <- hchart(
  new_fem_per_year,
  "line",
  hcaes(x = year, y = share_gender, group = comic),
  color = c("#518cca", "#e23636")
  ) %>%
  hc_title(text = "Comics Aren't Gaining Many Female Characters") %>%
  hc_subtitle(text = "Percentage of new characters who are female") %>%
  hc_xAxis(title = list(text = "")) %>%
  hc_yAxis(
    title  = list(text = ""),
    labels = list(format = "{value} %")
  ) %>%
  hc_tooltip(
    pointFormat = "{series.name}: <b>{point.y}</b><br/>",
    shared = TRUE,
    valueSuffix = " %",
    crosshairs = TRUE
  ) %>%
  hc_add_theme(hc_theme_538())
```

hc1


Comics Aren't Gaining Many Female Characters
Percentage of new characters who are female
45 %
40 %
35 %
30 %
25 %
20 %
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
20
DC
Marvel

hc_add_series()

```r
hc2 <- highchart() %>%
  hc_add_series(
    data = new_fem_per_year,
    type = "line",
    hcaes(x = year, y = share_gender, group = comic),
    color = c("#518cca", "#e23636"),
    marker = list(enabled = FALSE) # not needed in hchart()
  ) %>%
  hc_title(text = "Comics Aren't Gaining Many Female Characters") %>%
  hc_subtitle(text = "Percentage of new characters who are female") %>%
  hc_xAxis(title = list(text = "")) %>%
  hc_yAxis(
    title  = list(text = ""),
    labels = list(format = "{value} %")
    ) %>%
  hc_tooltip(
    pointFormat = "{series.name}: <b>{point.y}</b><br/>",
    shared = TRUE,
    valueSuffix = " %",
    crosshairs = TRUE
  ) %>%
  hc_add_theme(hc_theme_538())
```

hc2

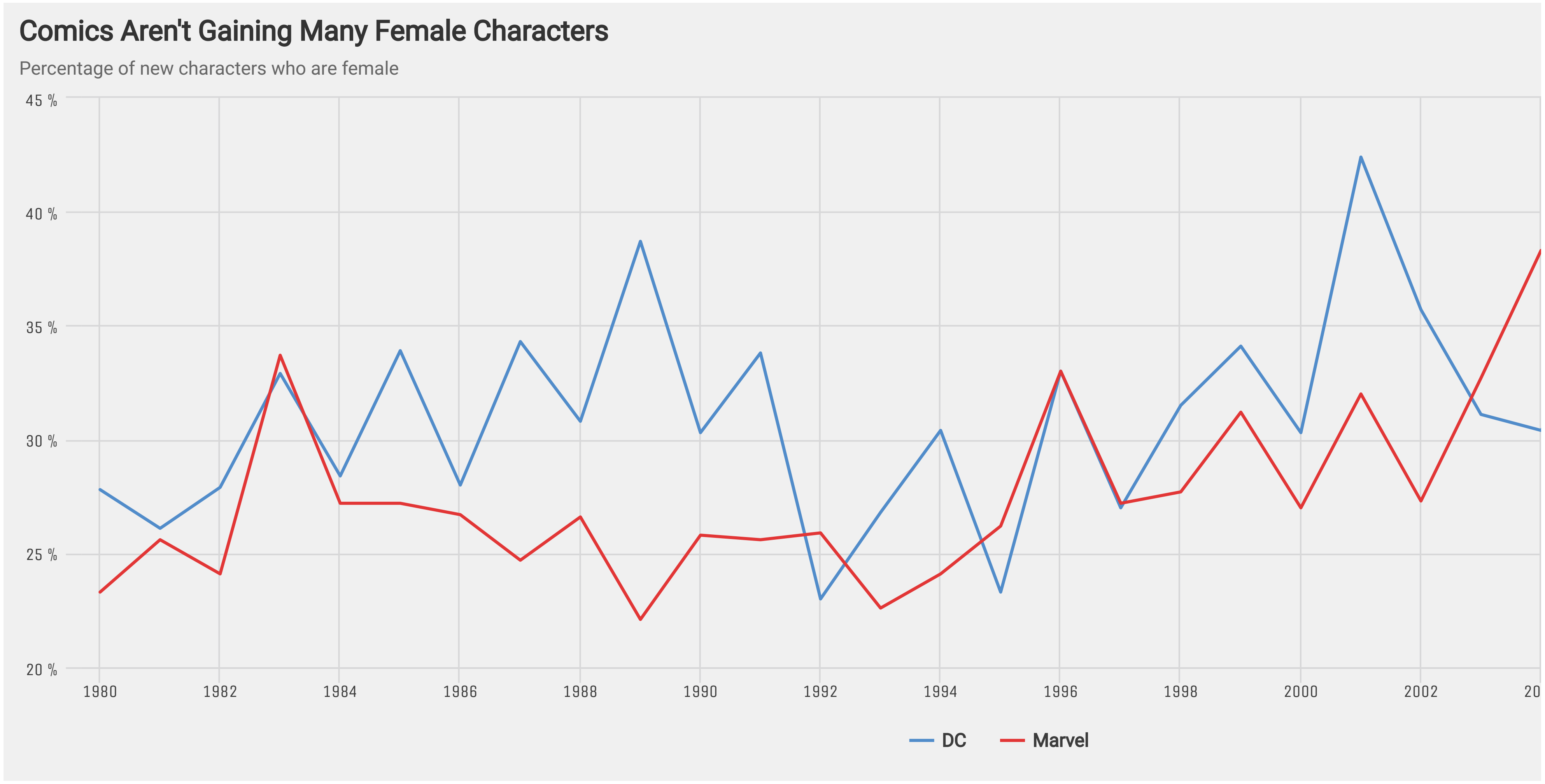
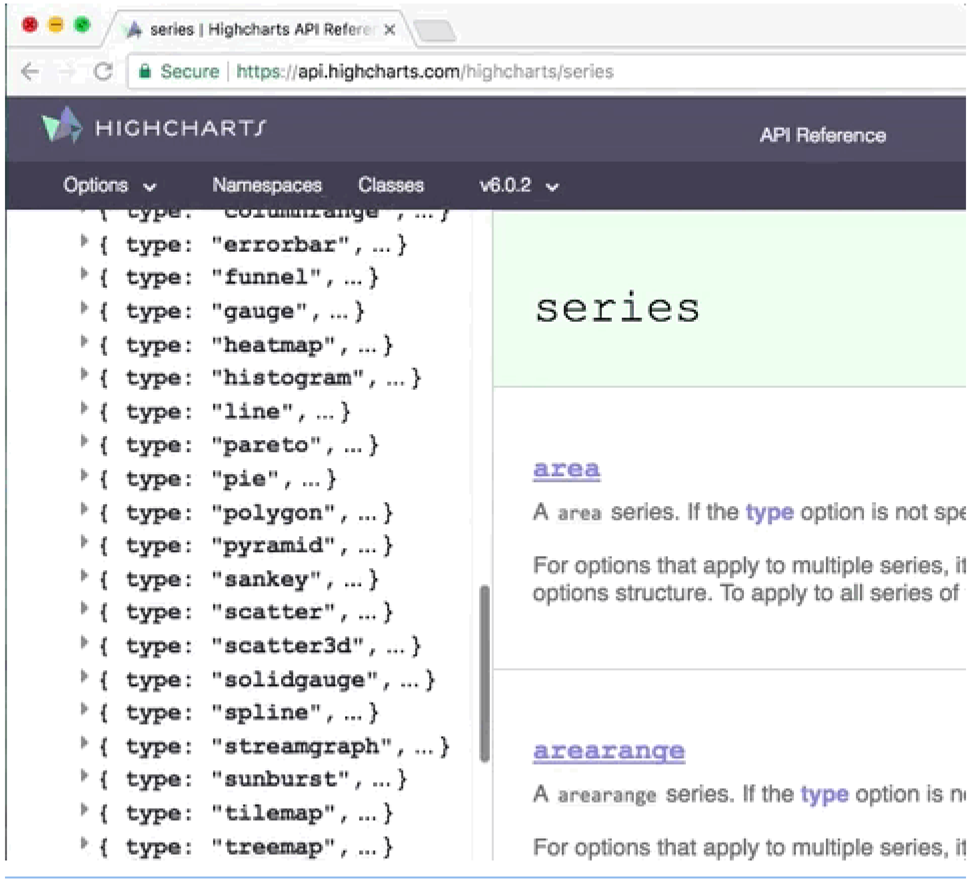

Comics Aren't Gaining Many Female Characters
Percentage of new characters who are female
45 %
40 %
35 %
30 %
25 %
20 %
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
20
DC
Marvel

Chart types


series | Highcharts API Refere
Secure | https://api.highcharts.com/highcharts/series
HIGHCHARTS
API Reference
Options
Namespaces
Classes
v6.0.2

```
{ type: "columnrange", ... }
{ type: "errorbar", ... }
{ type: "funnel", ... }
{ type: "gauge", ... }
{ type: "heatmap", ... }
{ type: "histogram", ... }
{ type: "line", ... }
{ type: "pareto", ... }
{ type: "pie", ... }
{ type: "polygon", ... }
{ type: "pyramid", ... }
{ type: "sankey", ... }
{ type: "scatter", ... }
{ type: "scatter3d", ... }
{ type: "solidgauge", ... }
{ type: "spline", ... }
{ type: "streamgraph", ... }
{ type: "sunburst", ... }
{ type: "tilemap", ... }
{ type: "treemap", ... }
```

series

area

A area series. If the type option is not spe

For options that apply to multiple series, it
options structure. To apply to all series of

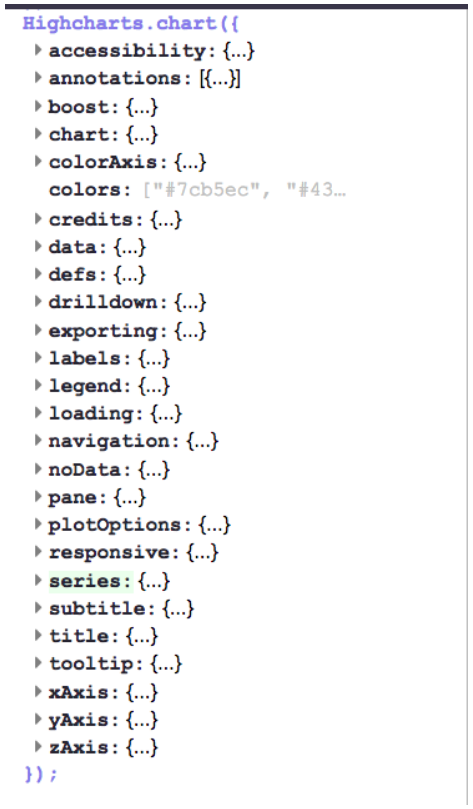
arearange

A arearange series. If the type option is n

For options that apply to multiple series, it

Style options

Functions



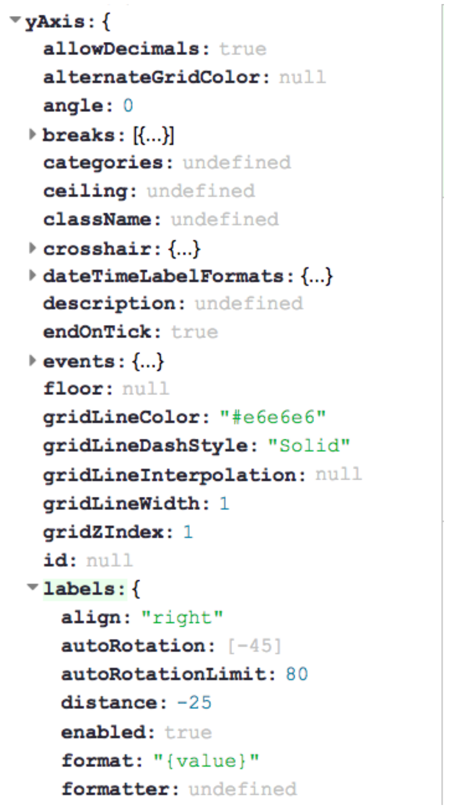
```
Highcharts.chart({
 ▸ accessibility: {…}
 ▸ annotations: [{…}]
 ▸ boost: {…}
 ▸ chart: {…}
 ▸ colorAxis: {…}
   colors: ["#7cb5ec", "#43…
 ▸ credits: {…}
 ▸ data: {…}
 ▸ defs: {…}
 ▸ drilldown: {…}
 ▸ exporting: {…}
 ▸ labels: {…}
 ▸ legend: {…}
 ▸ loading: {…}
 ▸ navigation: {…}
 ▸ noData: {…}
 ▸ pane: {…}
 ▸ plotOptions: {…}
 ▸ responsive: {…}
 ▸ series: {…}
 ▸ subtitle: {…}
 ▸ title: {…}
 ▸ tooltip: {…}
 ▸ xAxis: {…}
 ▸ yAxis: {…}
 ▸ zAxis: {…}
});
```

Options that modify the graphical parameters of the plot start with `hc_`. For example:

- `hc_title()`

Style options

Parameters



```
▾ yAxis: {
    allowDecimals: true
    alternateGridColor: null
    angle: 0
  ▸ breaks: [{…}]
    categories: undefined
    ceiling: undefined
    className: undefined
  ▸ crosshair: {…}
  ▸ dateTimeLabelFormats: {…}
    description: undefined
    endOnTick: true
  ▸ events: {…}
    floor: null
    gridLineColor: "#e6e6e6"
    gridLineDashStyle: "Solid"
    gridLineInterpolation: null
    gridLineWidth: 1
    gridZIndex: 1
    id: null
  ▾ labels: {
      align: "right"
      autoRotation: [-45]
      autoRotationLimit: 80
      distance: -25
      enabled: true
      format: "{value}"
      formatter: undefined
```

The parameters inside the `hc_` function family are also defined in the API.

Note that some options may have additional sub-options. To set these, you will have to used **named lists**

```
hc_yAxis(
  title  = list(text = ""),
  labels = list(format = "{value} %")
  )
```

Good-to-knows

The tooltip

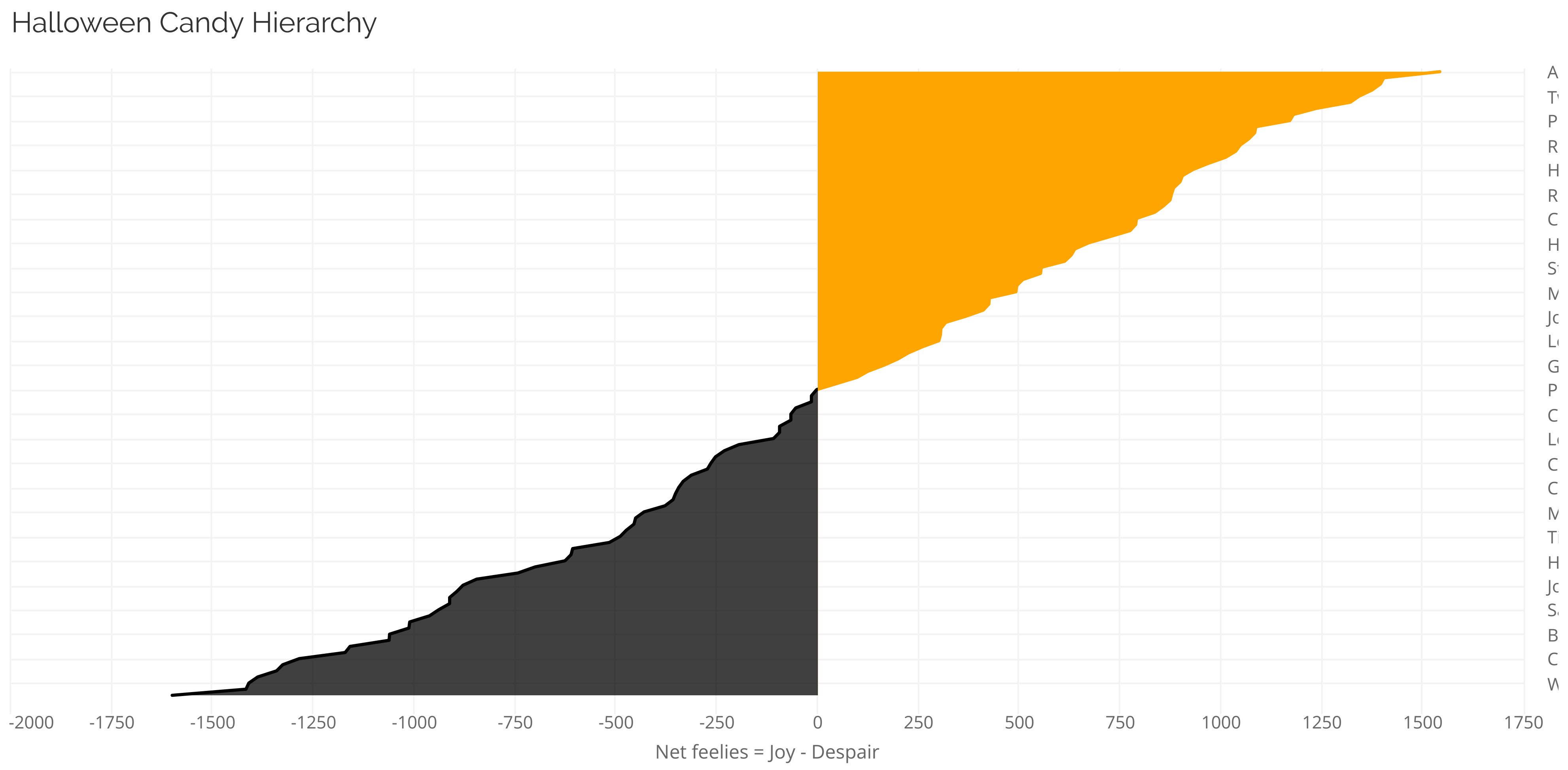
- The tooltip is a great way of conveying secondary information in your graphs.
- The function `tooltip_table()` helps you make quick HTML tables that can be passed to the `pointFormat` argument of `hc_tooltip()`

Here is an example of the tooltip I created for the following chart. I named this object `tltip`

```
left <- c("heart_eyes", "neutral_face", "sob") %>% map_chr(emo::ji)
right <- sprintf("{point.%s}", c("JOY", "MEH", "DESPAIR"))

tltip <- tooltip_table(left, right)
```

```r
hc3 <- hchart(
  candy_feelies,
  "area",
  hcaes(x = candy, y = net_feelies),
  threshold = 0,
  color = "orange",
  negativeColor = "black",
  marker = list(enabled = FALSE)
  ) %>%
  hc_chart(inverted = TRUE) %>%
  hc_xAxis(
    title = list(text = ""),
    opposite = TRUE,
    tickLength = 0
    ) %>%
  hc_yAxis(
    title = list(text = "Net feelies = Joy - Despair")
    ) %>%
  hc_tooltip(useHTML = TRUE, pointFormat = tltip) %>%
  hc_title(text = "Halloween Candy Hierarchy", align = "left") %>%
  hc_credits(
    enabled = TRUE,
    text = "Source: UBC - THE SCIENCE CREATIVE QUARTERLY",
    href = "https://www.scq.ubc.ca/so-much-candy-data-seriously/"
  ) %>%
  hc_add_theme(hc_theme_elementary())
```

hc3


Halloween Candy Hierarchy
-2000
-1750
-1500
-1250
-1000
-750
-500
-250
0
250
500
750
1000
1250
1500
1750
Net feelies = Joy - Despair
A
T
P
R
H
R
C
H
S
M
Jo
L
G
P
C
L
C
C
M
T
H
Jo
S
B
C
W

Good-to-knows

Facetting

Not automatically possible like in `ggplot2`, but you can get around it using `purrr::map()` and `hw_grid()`.

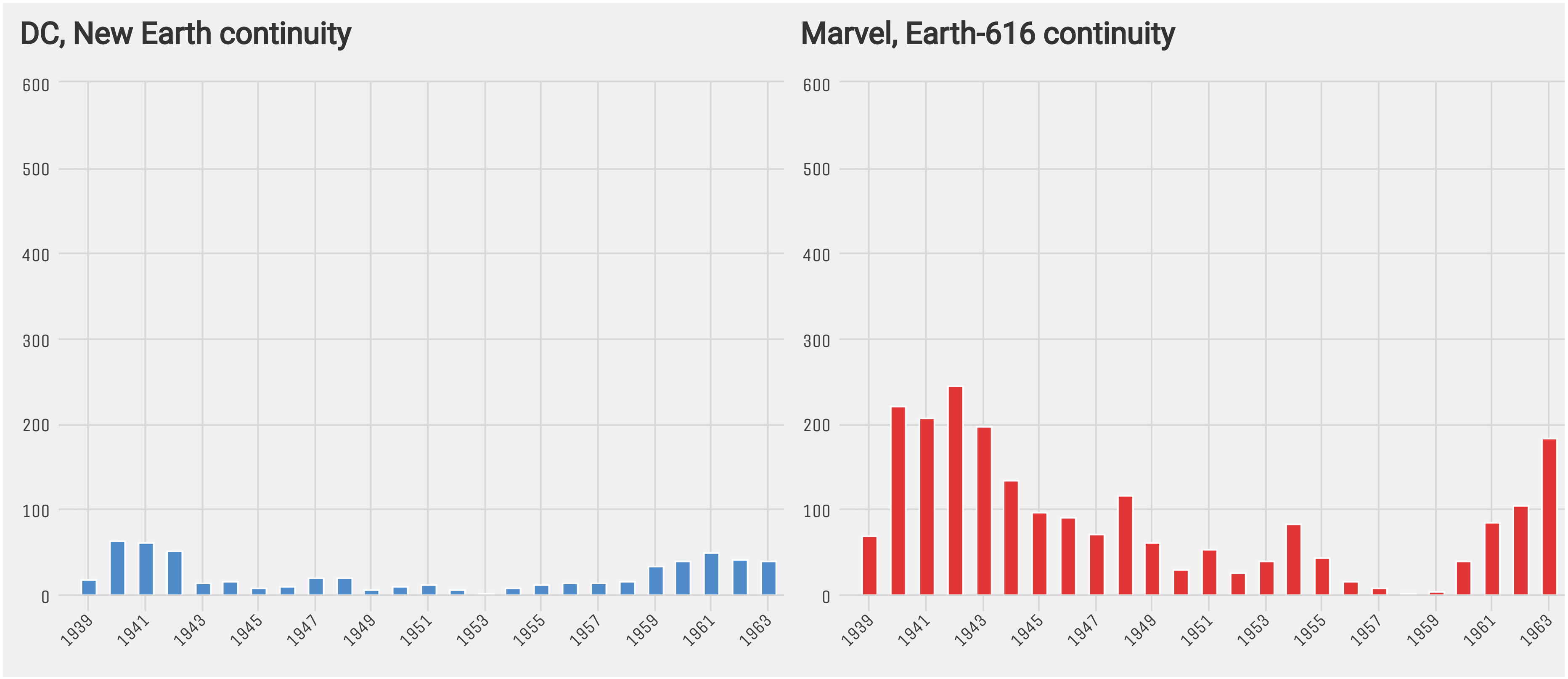
Here's another example using FiveThirtyEight's data.

```
str(new_char_per_year)
```

```
## List of 2
##  $ DC    :Classes 'tbl_df', 'tbl' and 'data.frame':  73 obs. of  5 variables:
##   ..$ comic  : chr [1:73] "DC" "DC" "DC" "DC" ...
##   ..$ year   : Factor w/ 73 levels "1939","1940",..: 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ n      : int [1:73] 18 64 61 52 14 15 7 9 20 20 ...
##   ..$ couleur: chr [1:73] "#518cca" "#518cca" "#518cca" "#518cca" ...
##   ..$ titre  : chr [1:73] "DC, New Earth continuity" "DC, New Earth continuity" "DC, New Earth continuity"
##  $ Marvel:Classes 'tbl_df', 'tbl' and 'data.frame':  73 obs. of  5 variables:
##   ..$ comic  : chr [1:73] "Marvel" "Marvel" "Marvel" "Marvel" ...
##   ..$ year   : Factor w/ 73 levels "1939","1940",..: 1 2 3 4 5 6 7 8 9 10 ...
##   ..$ n      : int [1:73] 69 221 207 244 198 134 97 90 72 116 ...
##   ..$ couleur: chr [1:73] "#e23636" "#e23636" "#e23636" "#e23636" ...
##   ..$ titre  : chr [1:73] "Marvel, Earth-616 continuity" "Marvel, Earth-616 continuity" "Marvel, Earth-616
```

```r
make_bars <- function(p){

  tltip <- tooltip_table("Characters", "{point.n}")

  couleur <- unique(p$couleur)
  titre   <- unique(p$titre)

  hchart(
    p,
    "column",
    hcaes(x = year, y = n),
    showInLegend = FALSE,
    color = couleur
    ) %>%
    hc_add_theme(hc_theme_538()) %>%
    hc_yAxis(title = list(text = ""), max = 560) %>%
    hc_xAxis(title = list(text = "")) %>%
    hc_title(text = titre) %>%
    hc_tooltip(useHTML = TRUE, pointFormat = tltip)
}

hc4 <- new_char_per_year %>%
  map(make_bars) %>%
  hw_grid(ncol = 2, rowheight = 400) %>%
  htmltools::browsable()
```

hc4



DC, New Earth continuity
Marvel, Earth-616 continuity
600
500
400
300
200
100
0
1939
1941
1943
1945
1947
1949
1951
1953
1955
1957
1959
1961
1963

Other resources

Useful links

Official websites:

- Highcharter package
- Highcharts API
  - API Options Reference

Blog posts:

- Joshua Kunst's blog (maintainer of the package)
- Thinking in highcharter: How to build any Highcharts plot in R
- Creating interactive plots with R and Highcharts
- Making a Shiny dashboard with Highcharter

Misc:

- For more on `hchart()` vs `hc_add_series()` check out the discussion on issue #302 at Github on Github.
- And on faceting, see discussion here and example here

Thank you!

Any questions?