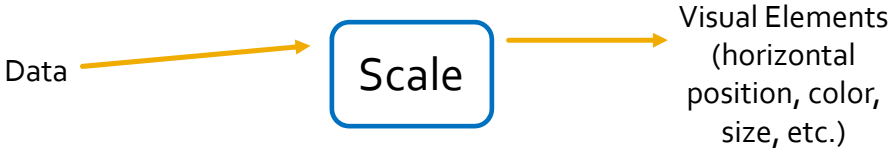
Introduction to ggplot2 Scale Customization (Part 1 of 2)

2

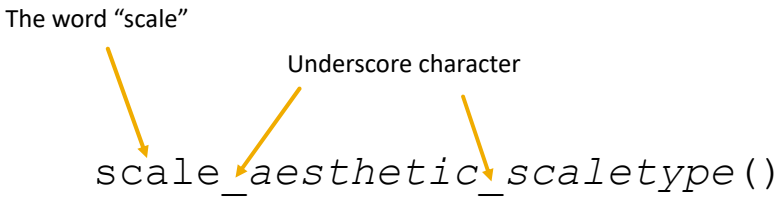
I. The Role of Scales



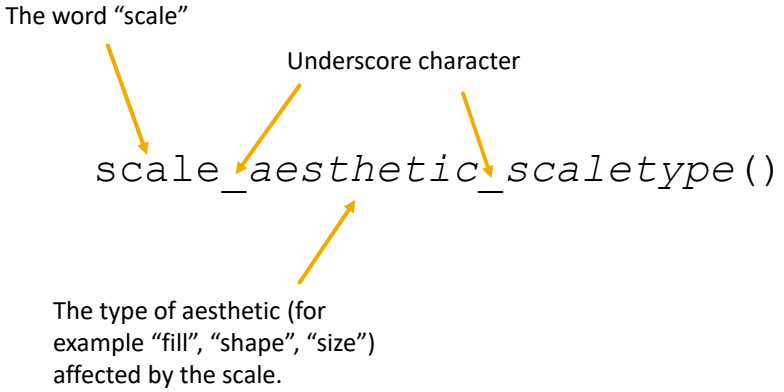
3

II. General Syntax for Scales

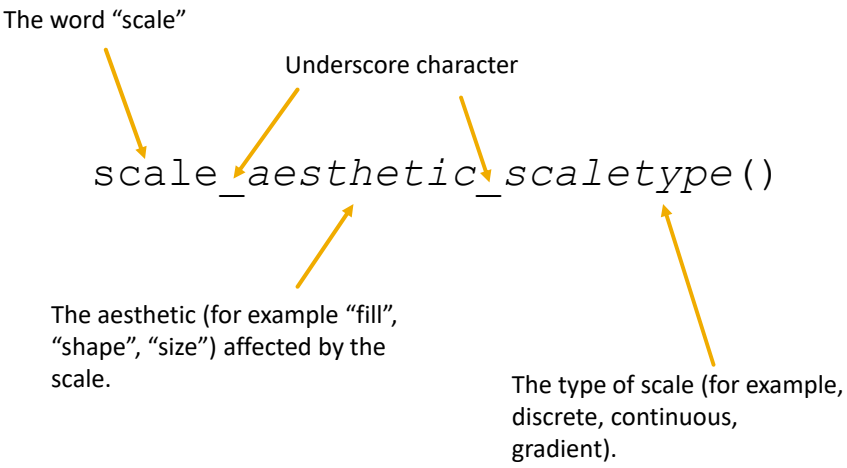
General format of scale function syntax:



4



5



6

Example: Aesthetics for geom_point()

Aesthetics

`geom_point` understands the following aesthetics (required aesthetics are in bold):

- `x`
- `y`
- `alpha`
- `colour`
- `fill`
- `group`
- `shape`
- `size`
- `stroke`

Learn more about setting these aesthetics in `vignette("ggplot2-specs")`

Source: www.rdocumentation.org/packages/ggplot2/versions/3.0.0/topics/geom_point

7

Frequently used types of scales:

- Manual scales: converts discrete data to a set of manually specified colors, shapes, sizes, or line types.
- Color scales: converts continuous and discrete data to colors
- Continuous position scales: converts numeric and date values to X and Y locations in a visualization space (e.g. a Cartesian coordinate plane) (next week)

8

III. Some Color Scales

Default color scales for continuous data (normally color and fill)

`scale_aes_gradient`	Creates a two-color gradient (low to high).
`scale_aes_gradient2`	Creates a diverging color gradient (low-mid-high).
`scale_aes_gradientn()`	Creates an n-color gradient

Adapted from: ggplot2.tidyverse.org/reference/scale_gradient.html

9

Color scales for continuous data. Use with aesthetics color and fill. Examples below use the "fill" aesthetic.

`scale_fill_gradient()`	two-color gradient from low to high
`scale_fill_gradient2()`	three-color gradient from low to mid to high
`scale_fill_gradientn()`	n-color gradient

Some arguments:

`low`	Color for the low end of the gradient (color name or hex code).
`mid`	Used with `gradient2`; color for the middle of the gradient (color name or hex code).
`high`	Color for the high end of the gradient (color name or hex code).
`limits`	Vector that specifies the minimum and maximum values of the scale.
`name`	A custom name for the scale. Appears in the title or axis. If `NULL`, the label is omitted.

10

Some (more) arguments:

`breaks` Specifies where the axis or legend guide will display break points on a continuous scale. One of:

- `NULL` for no breaks.
- A numeric vector of positions on the scale.
- A function that takes the limits as input and returns breaks as output.
- `waiver()` for the default breaks computed by the transformation object.

`labels` Specifies labels of break points on the scale. Some options:

- `NULL` for no labels.
- A character vector giving labels (must be same length as breaks).
- A function that takes the breaks as input and returns breaks as output.
- `waiver()` for the default labels.

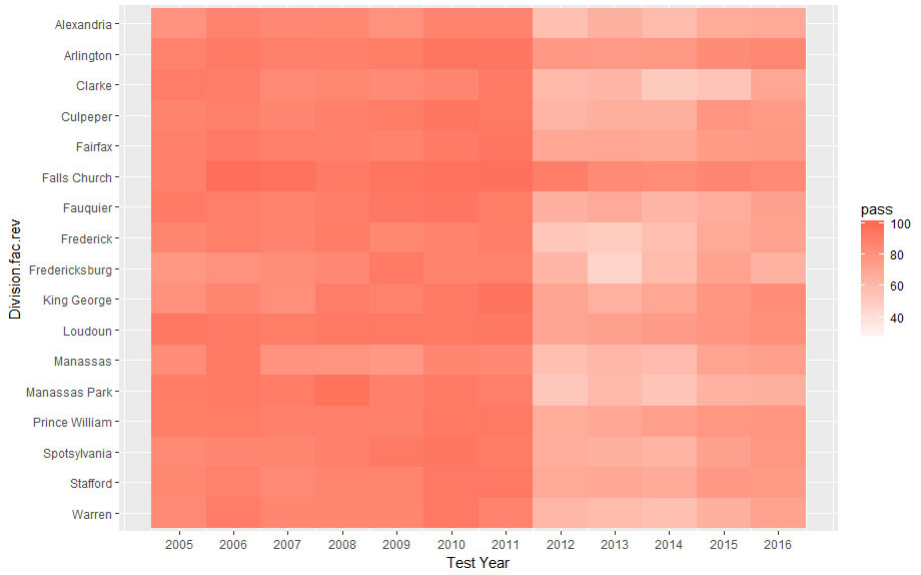
Source: ggplot2.tidyverse.org/reference/scale_gradient.html

11

```
# heat map with new color map
# add the custom x-axis scale
ggplot(data = nva.math) +
  geom_tile(mapping = aes(x = `Test Year`,
            y = Division.fac.rev,
                          fill = pass)) +
  scale_x_continuous(breaks = seq(2005, 2016)) +
  scale_fill_gradient(low = "#fff1ef",
                      high = "#ff6b56",
                      limits = c(30, 100))
```

12

Customize this legend?



13

```
ggplot(data = nva.math) +
  geom_tile(mapping = aes(x = `Test Year`,
                          y = Division.fac.rev,
                          fill = pass)) +
  scale_x_continuous(breaks = seq(2005, 2016)) +
  scale_fill_gradient(low = "#fff1ef",
                      high = "#ff6b56",
                      limits = c(30, 100),
                      breaks = seq(30, 100, by = 10))
```

14

15

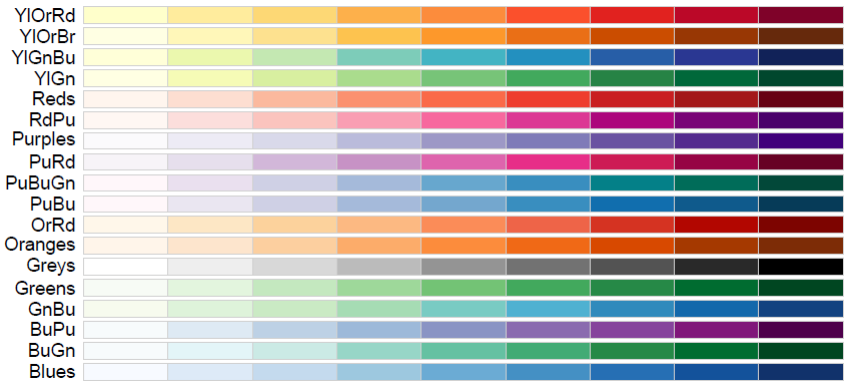
Use with aesthetics color and fill. Examples below use "fill".

`scale_fill_manual()`	Converts data values to colors in a custom color palette that you create. (We saw this earlier.)
`scale_fill_brewer()`	Converts data values to colors in a prepackaged "ColorBrewer" palette.
`scale_fill_grey()`	Converts data values to levels of gray.

16

Palettes from the RColorBrewer package*

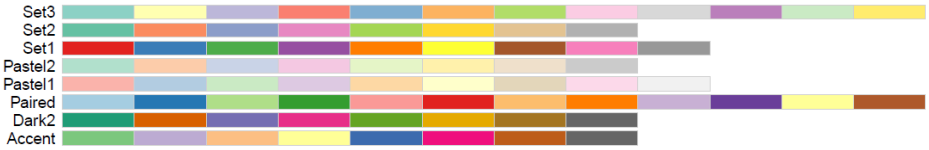
Sequential color palettes `(type = seq)`



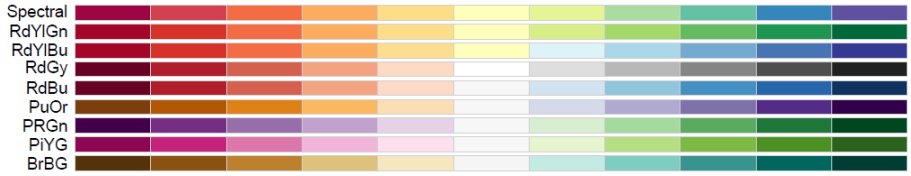
*The function `display.brewer.all()` will display the palettes. See cran.r-project.org/web/packages/RColorBrewer/RColorBrewer.pdf for more information.

17

Qualitative (categorical) color palettes `(type = qual)`



Diverging color palettes `(type = div)`



18

`scale_fill_brewer()`	Converts data values mapped to the fill aesthetic to colors in a "ColorBrewer" palette.
`scale_color_brewer()`	Converts data values mapped to the color aesthetic to colors in a "ColorBrewer" palette.

Some arguments:

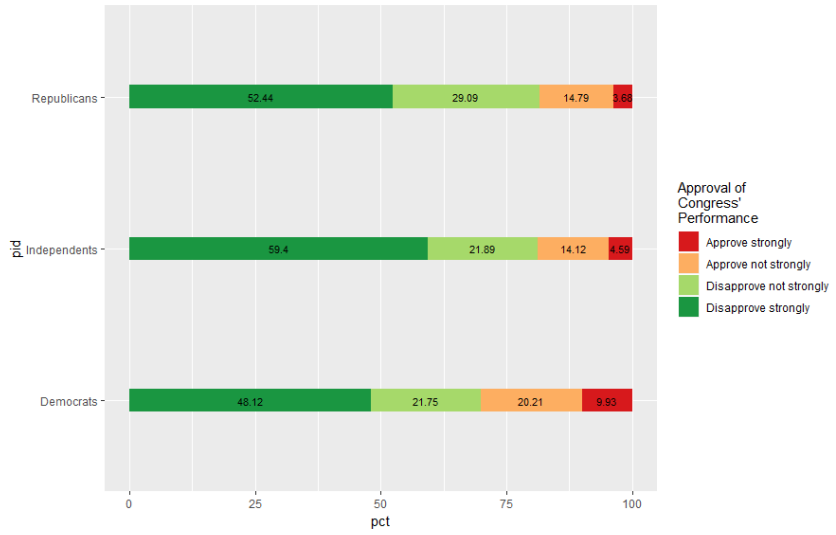
`type`	One of `seq` (sequential), `div` (diverging) or `qual` (qualitative).
`palette`	If a string, will use that named ColorBrewer palette.
`guide`	Type of legend. Use `colorbar` for continuous color bar, or `legend` for discrete color legend.
`na.value`	Color to use for missing values.
`direction`	Sets the order of colors in the scale. If `1`, the default, colors are as output in the original order. If `-1`, the order of colors is reversed.

Adapted from: ggplot2.tidyverse.org/reference/scale_brewer.html

19

```
#flip the coordinate space for a horizontal bar chart
ggplot(data = congapp,
       aes(x = pid, y = pct, fill = approval)) +
  geom_bar(stat = "identity",
           width = 0.15) +
  theme(plot.margin = unit(c(1, 1, 1, 1), "cm")) +
  geom_text(aes(label = pct),
            position = position_stack(vjust = 0.5),
            size = 3) +
  coord_flip() +
scale_fill_brewer(name = "Approval of\nCongress'\nPerformance",
                  type = div,
                  palette = "RdYlGn")
```

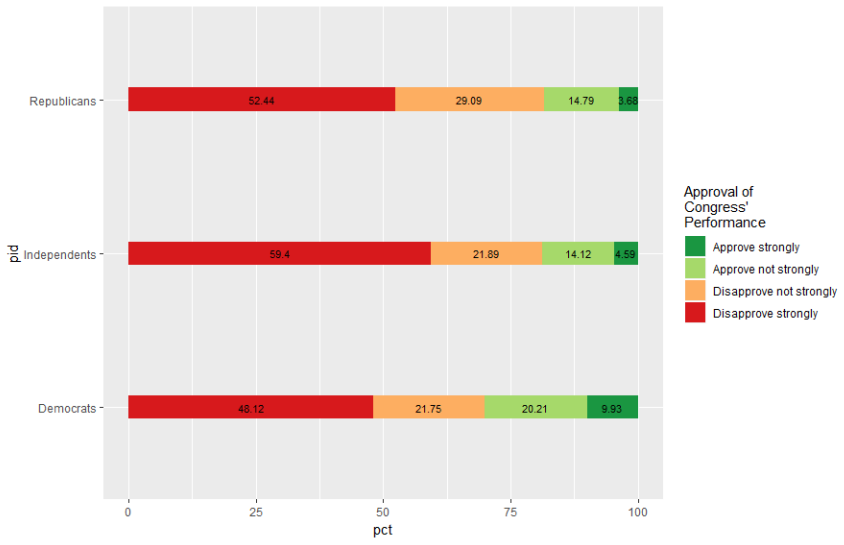
20



21

```
#flip the coordinate space for a horizontal bar chart
ggplot(data = congapp,
       aes(x = pid, y = pct, fill = approval)) +
  geom_bar(stat = "identity",
           width = 0.15) +
  theme(plot.margin = unit(c(1, 1, 1, 1), "cm")) +
  geom_text(aes(label = pct),
            position = position_stack(vjust = 0.5),
            size = 3) +
  coord_flip() +
scale_fill_brewer(name = "Approval of\nCongress'\nPerformance",
                  type = div,
                  palette = "RdYlGn",
                  direction = -1)
```

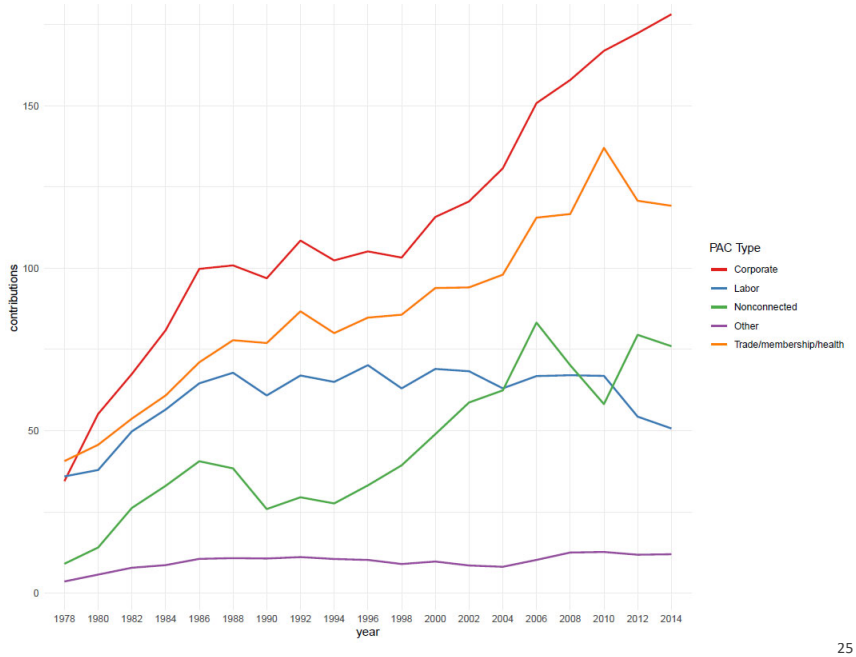
22



23

```
library(RColorBrewer)
pac_plot +
  scale_color_brewer(name = "PAC Type",
                     type = qual,
                     palette = "Set1")
```

24



25