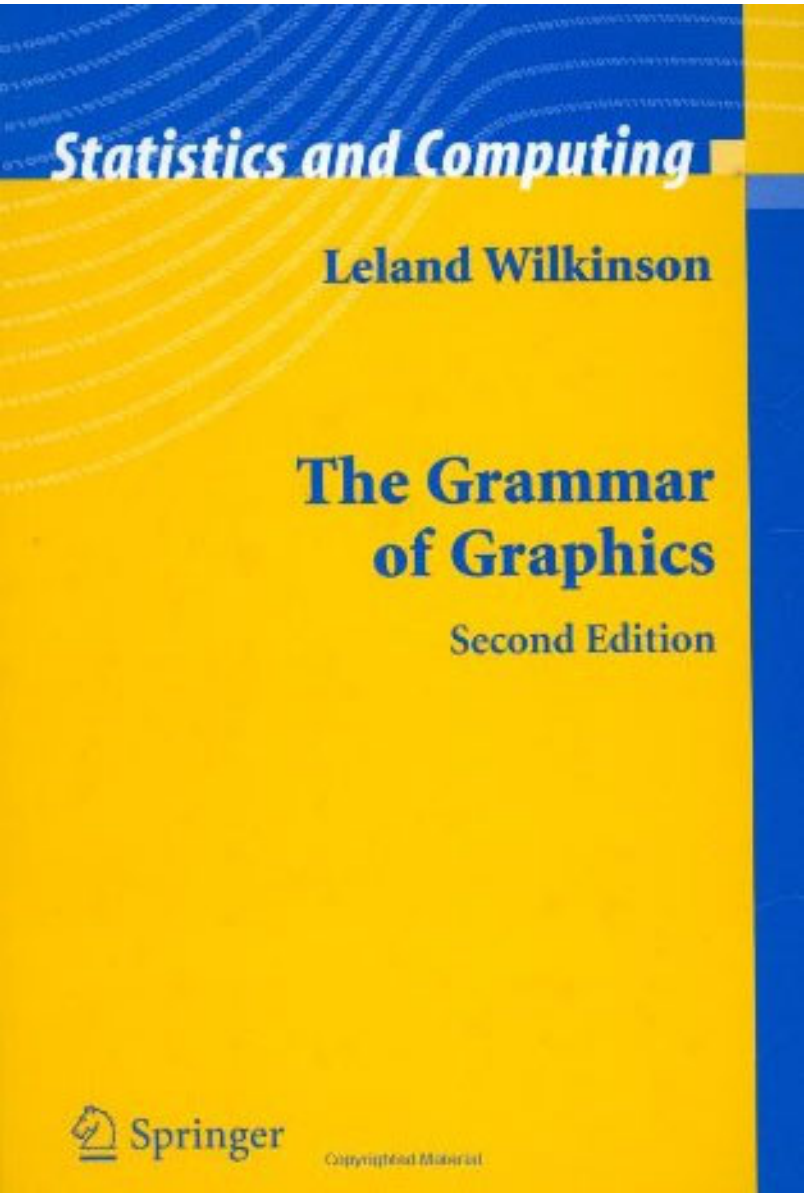
Endless Forms Most Beautiful: Creating Customized Data Visualizations with ggplot2

Lisa Federer, PhD, MLIS
Data Science and Open Science Librarian
Office of Strategic Initiatives
National Library of Medicine
National Institutes of Health

Workshop overview

- The Grammar of Graphics: components of visualizations
- Practical considerations and design choices
- Creating plots in RStudio with ggplot2
- Your questions

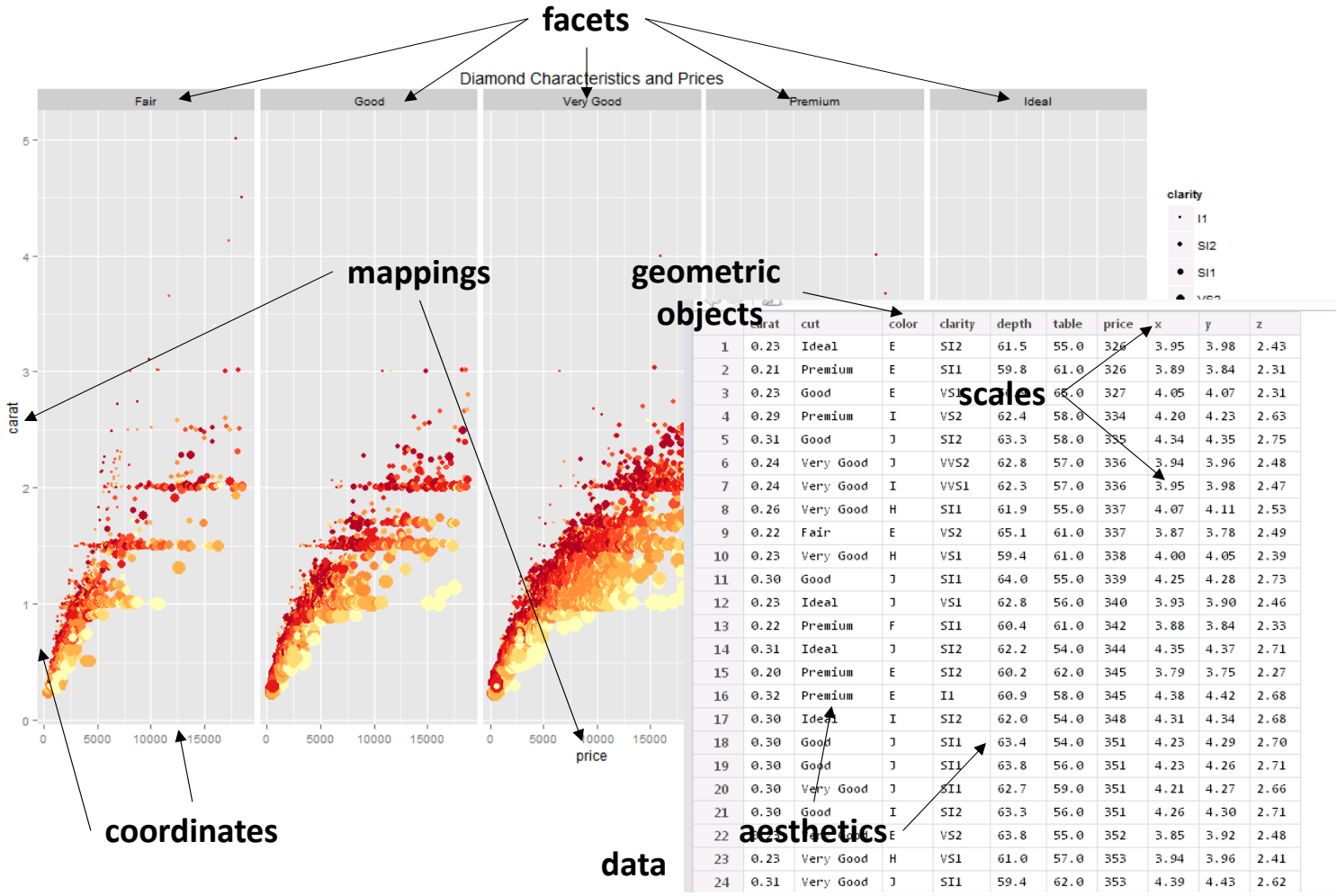


The Grammar of Graphics

"A language consisting of words and no grammar expresses only as many ideas as there are words. By specifying how words are combined in statements, a grammar expands a language's scope…The grammar of graphics takes us beyond a limited set of charts (words) to an almost unlimited world of graphical forms (statements)."

Grammar of Graphics "parts of speech"

- **Data**: what is being visualized.
- **Mappings**: mappings between variables in the data and components of the chart.
- **Geometric Objects**: geometric objects that are used to display the data, such as points, lines, or shapes.
- **Aesthetic Properties**: qualities about geometric objects that convey details about the data
- **Scales:** control how variables are mapped to aesthetics.
- **Coordinates:** describe how data is mapped to the plot
- **Statistical Transformations**: applied to the data to summarize it.
- **Facets:** describe how the data is partitioned into subsets and how these different subsets are plotted.



	carat	cut	color	clarity	depth	table	price	x	y	z
1	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
2	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
3	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
4	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
5	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
6	0.24	Very Good	J	VVS2	62.8	57.0	336	3.94	3.96	2.48
7	0.24	Very Good	I	VVS1	62.3	57.0	336	3.95	3.98	2.47
8	0.26	Very Good	H	SI1	61.9	55.0	337	4.07	4.11	2.53
9	0.22	Fair	E	VS2	65.1	61.0	337	3.87	3.78	2.49
10	0.23	Very Good	H	VS1	59.4	61.0	338	4.00	4.05	2.39
11	0.30	Good	J	SI1	64.0	55.0	339	4.25	4.28	2.73
12	0.23	Ideal	J	VS1	62.8	56.0	340	3.93	3.90	2.46
13	0.22	Premium	F	SI1	60.4	61.0	342	3.88	3.84	2.33
14	0.31	Ideal	J	SI2	62.2	54.0	344	4.35	4.37	2.71
15	0.20	Premium	E	SI2	60.2	62.0	345	3.79	3.75	2.27
16	0.32	Premium	E	I1	60.9	58.0	345	4.38	4.42	2.68
17	0.30	Ideal	I	SI2	62.0	54.0	348	4.31	4.34	2.68
18	0.30	Good	J	SI1	63.4	54.0	351	4.23	4.29	2.70
19	0.30	Good	J	SI1	63.8	56.0	351	4.23	4.26	2.71
20	0.30	Very Good	J	SI1	62.7	59.0	351	4.21	4.27	2.66
21	0.30	Good	I	SI2	63.3	56.0	351	4.26	4.30	2.71
22	0.23	Ideal	E	VS2	63.8	55.0	352	3.85	3.92	2.48
23	0.23	Very Good	H	VS1	61.0	57.0	353	3.94	3.96	2.41
24	0.31	Very Good	J	SI1	59.4	62.0	353	4.39	4.43	2.62

From code to chart

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color, size = clarity)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) +
scale_colour_brewer(palette = "YlOrRd")
```

From code to chart: *data*

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color, size
= clarity)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) +
scale_colour_brewer(palette = "YlOrRd")
```

From code to chart: *mappings*

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) +
scale_colour_brewer(palette = "YlOrRd")
```

From code to chart: *geometric objects*

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color, size = clarity)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) + scale_colour_brewer(palette = "YlOrRd")
```

From code to chart: *aesthetic properties*

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color, size = clarity)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) + scale_colour_brewer(palette = "YlOrRd")
```

From code to chart: *scales*

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color, size = clarity)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) +
scale_colour_brewer(palette = "YlOrRd")
```

From code to chart: *coordinates*

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color, size = clarity)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) +
scale_colour_brewer(palette = "YlOrRd")
```

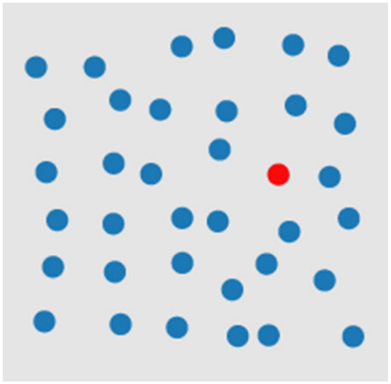
From code to chart: *facets*

```
diamonds %>% ggplot(aes(x = price, y = carat, col = color, size
= clarity)) +
geom_point(stat = "unique") +
coord_cartesian(xlim = c(0,20000)) +
xlab("Price, US $") + ylab("Carat") +
ggtitle("Prices and Characteristics of Round Cut Diamonds") +
facet_wrap(~cut, nrow=1) +
scale_colour_brewer(palette = "YlOrRd")
```

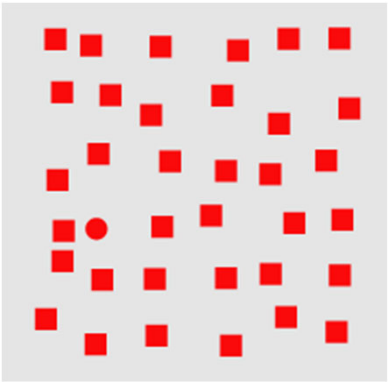

Practical considerations and design choices
Working effectively with color and chart choices

Pre-attentive processing

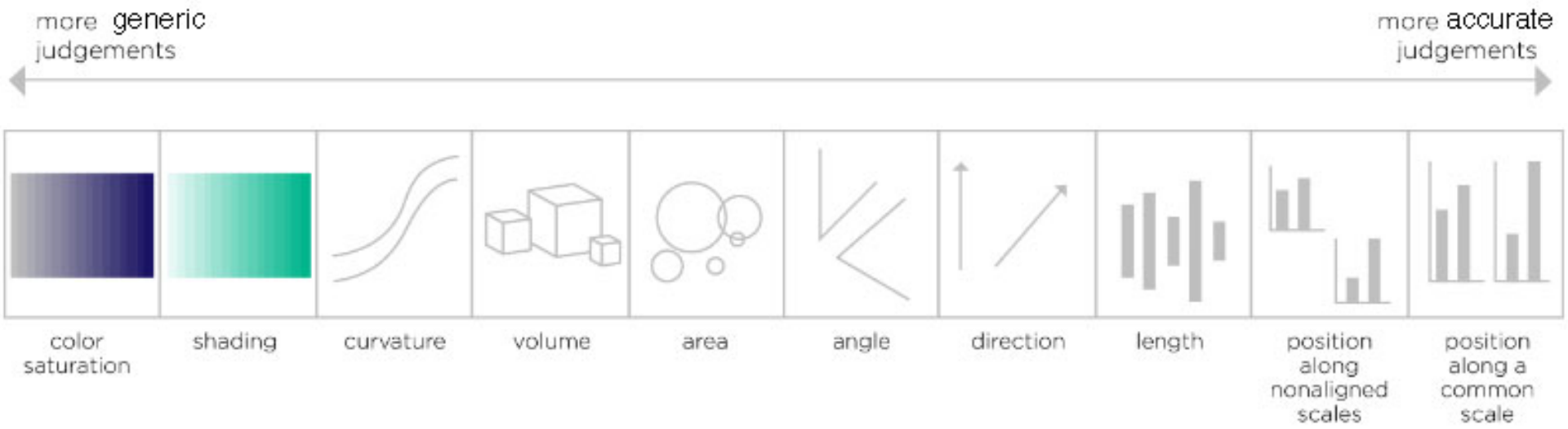
Differences in hue



Differences in shape

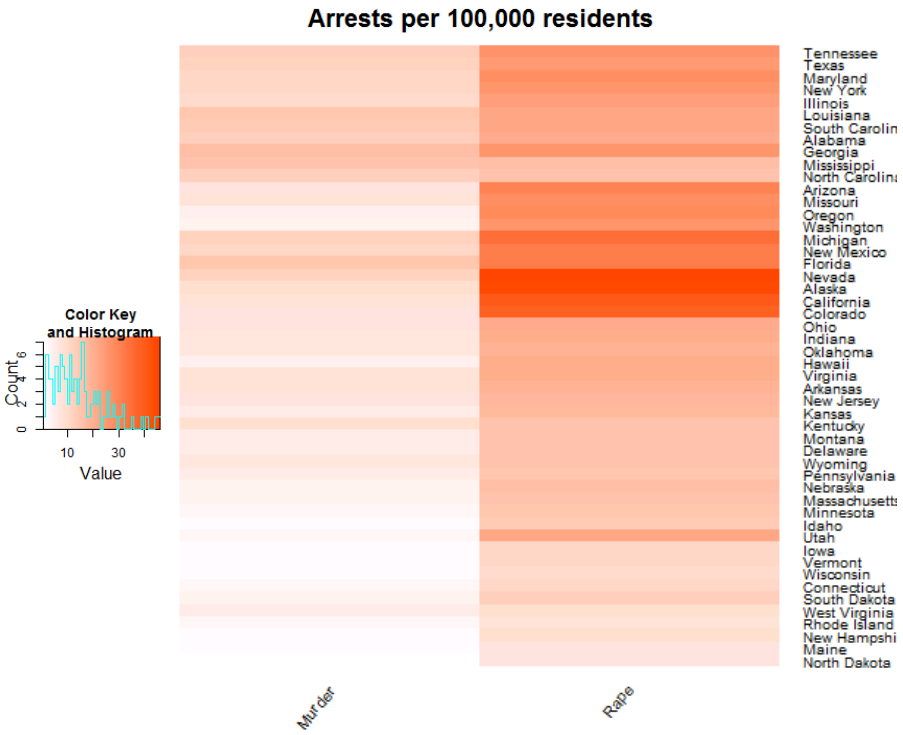


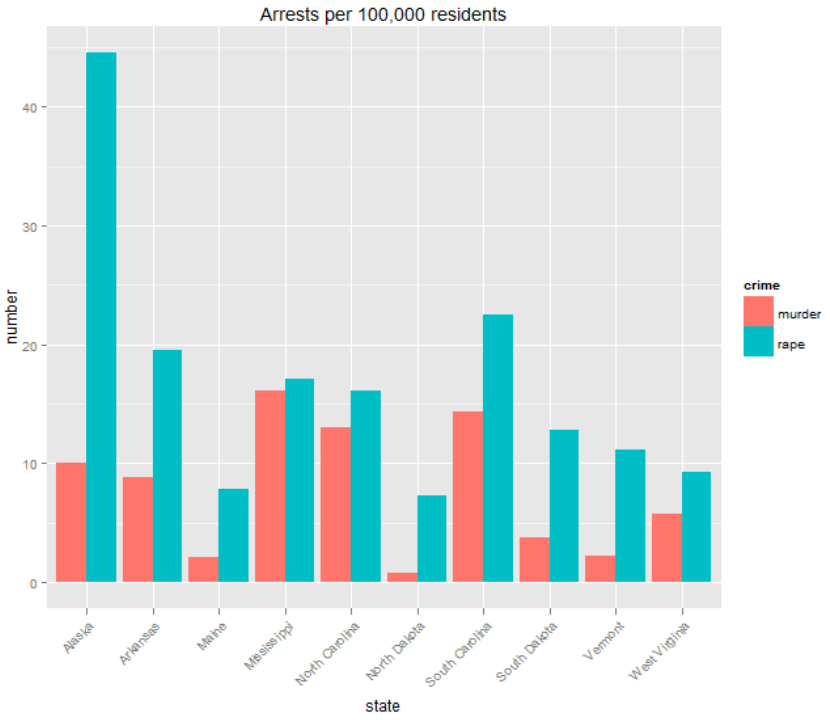
Perceptual tasks



From Alberto Cairo, *The Functional Art*
Adaptation of Cleveland and McGill's scale from "Graphical Perception: Theory, Experimentation and Application to the Development of Graphical Methods," available at https://web.cs.dal.ca/~sbrooks/csci4166-6406/seminars/readings/Cleveland_GraphicalPerception_Science85.pdf

Design for ease of perceptual processing



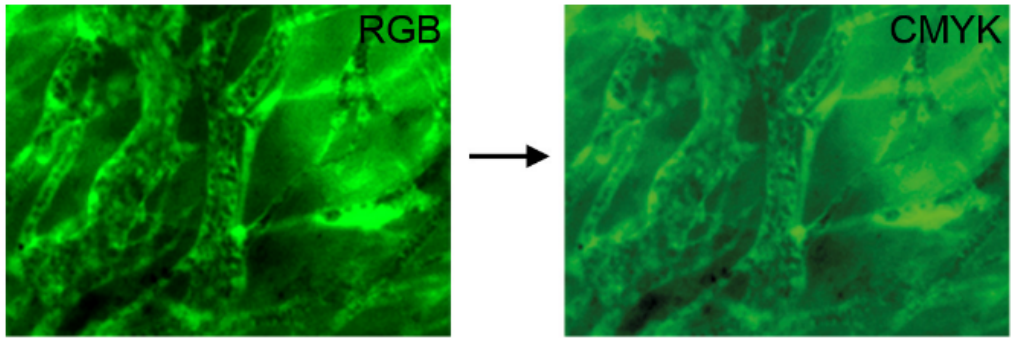


Colorspaces (ggplot default = RGB)

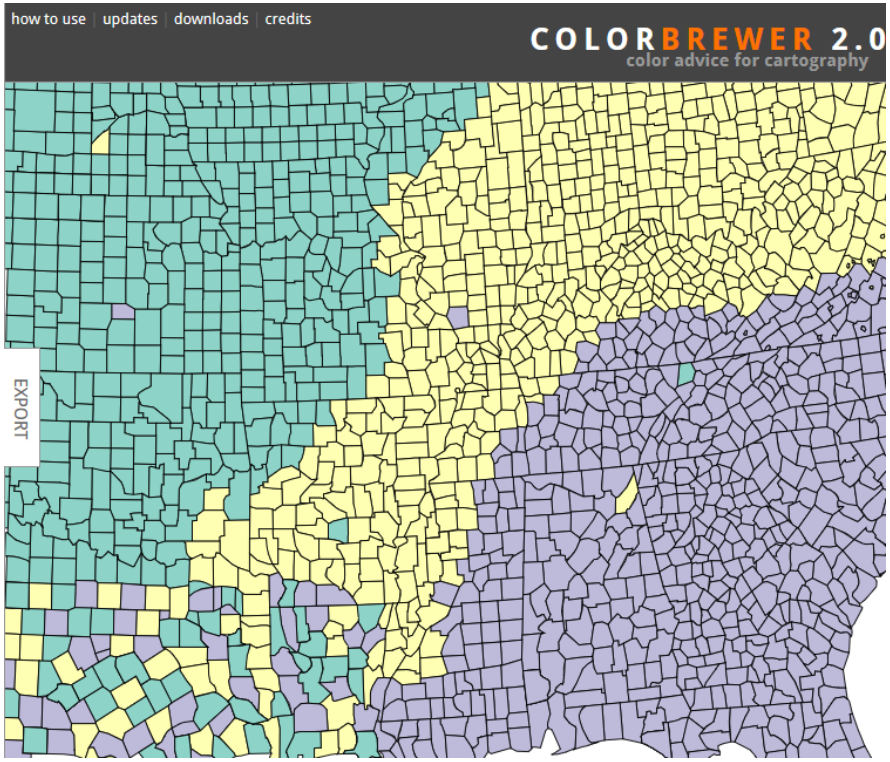
4. COLOURS

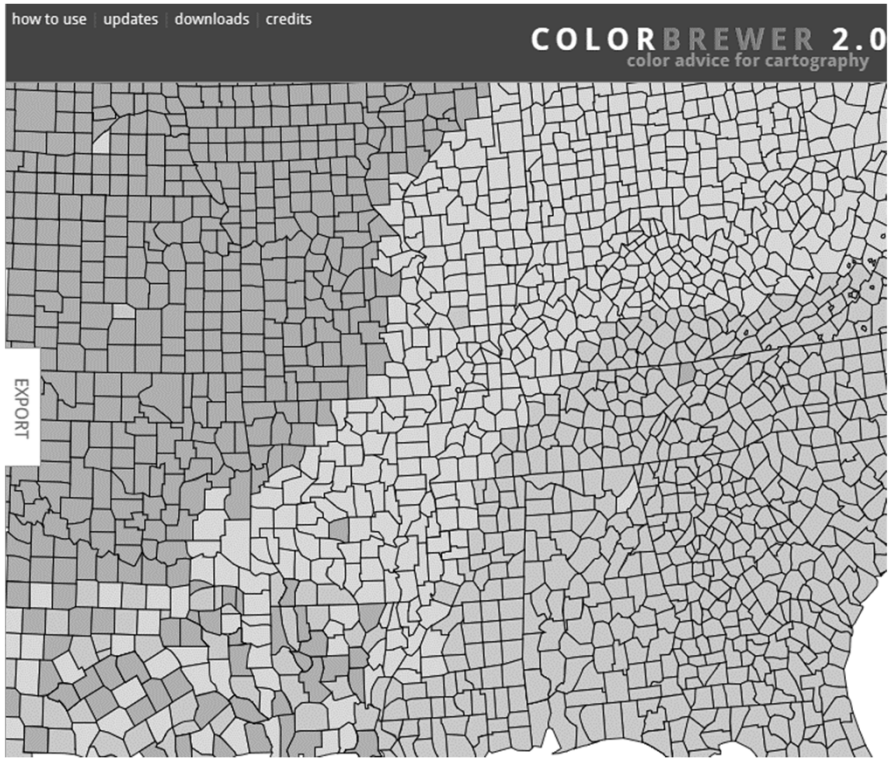
Editors may request for colour figures but reserve the right to convert these to black and white at their discretion.

Full colour artwork should be provided in CMYK format; ensure that you are happy with the conversion before submission of final artwork. The example below shows the shift in colour between RGB and the equivalent colour shown in CMYK - subtle details are often lost during the conversion.

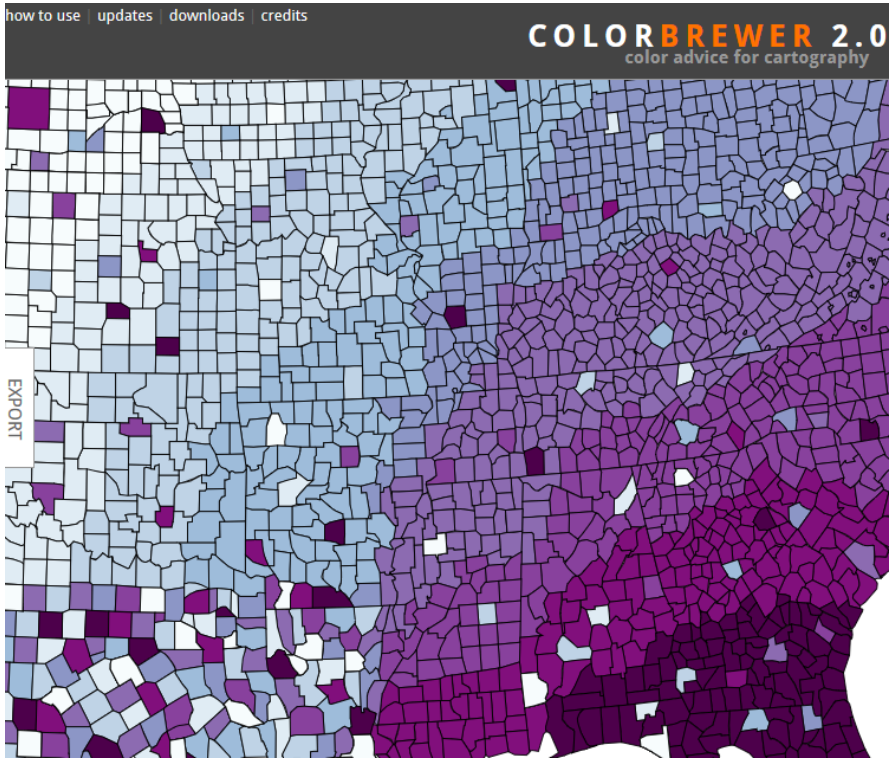


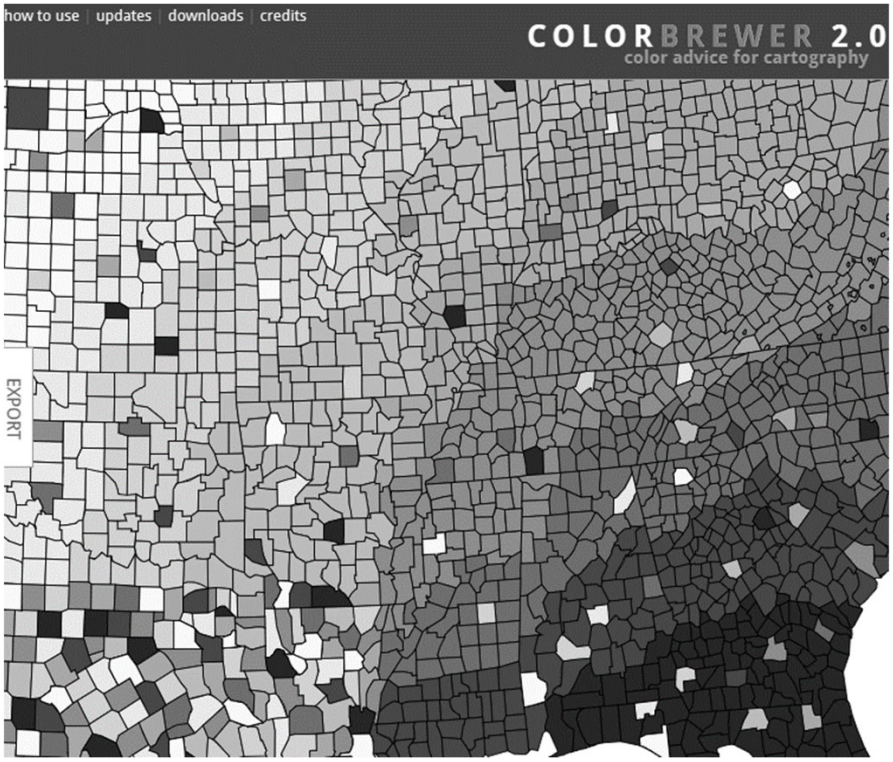
Greyscale (“photocopy safe”)



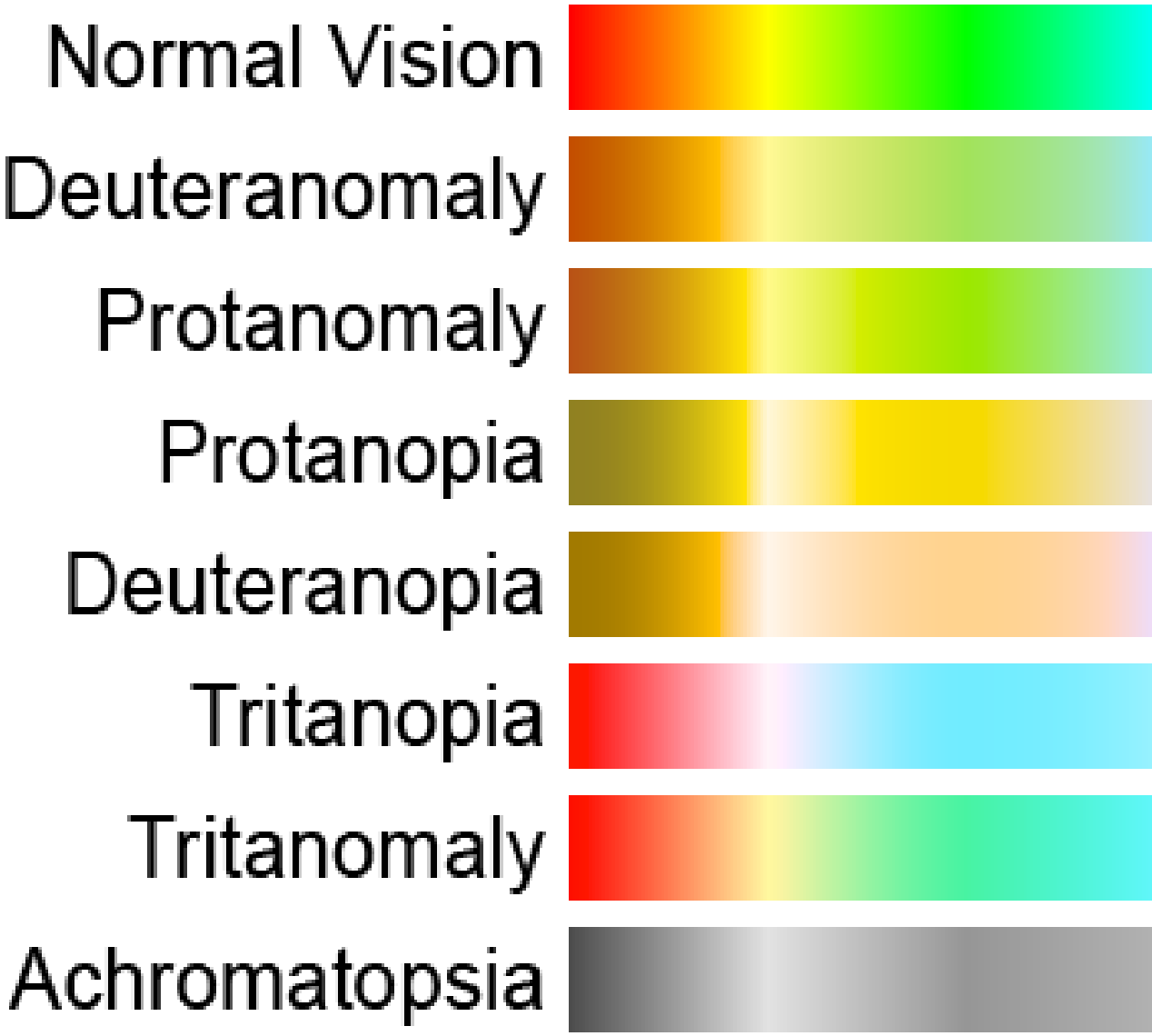


Greyscale – nope!





Color blindness

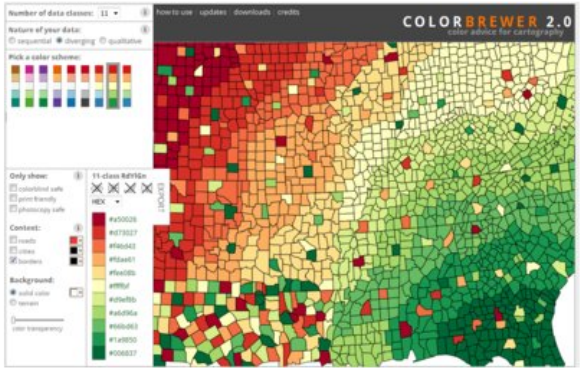


http://www.vischeck.com/vischeck

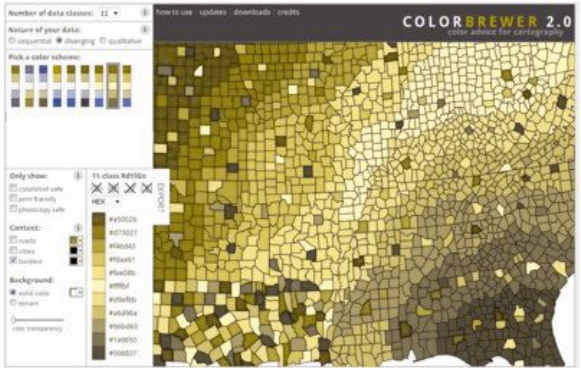
Try Vischeck on Your Image Files

Your Results:

Original Image



Deuteranope Simulation



Select the type of color vision to simulate:

(•) Deuteranope (a form of red/green color deficit)

( ) Protanope (another form of red/green color deficit)

( ) Tritanope (a blue/yellow deficit- very rare)

Image file: Browse... No file selected. Run Vischeck!

Named colors in R

color	name	color	name
	darkgreen		deepskyblue
	darkgrey		deepskyblue1
	darkkhaki		deepskyblue2
	darkmagenta		deepskyblue3
	darkolivegreen		deepskyblue4
	darkolivegreen1		dimgray
	darkolivegreen2		dimgrey
	darkolivegreen3		dodgerblue
	darkolivegreen4		dodgerblue1
	darkorange		dodgerblue2
	darkorange1		dodgerblue3
	darkorange2		dodgerblue4
	darkorange3		firebrick
	darkorange4		firebrick1
	darkorchid		firebrick2
	darkorchid1		firebrick3

http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

Color Brewer palettes