

Color choices

Oliver W. Layton

CS251/2: Data Analysis and Visualization

Lecture 9

Spring 2021

Perceptual uniformity

- Whether your color scale data is continuous or discrete, you want equidistant data points to appear equally different in color (**perceptual uniformity**).
- 0.0 - 0.1 should look like 0.9 - 1.0!
- Popular rainbow scale is NOT perceptually uniform!
- ColorBrewer scales tend to do a good job.

rainbow scale

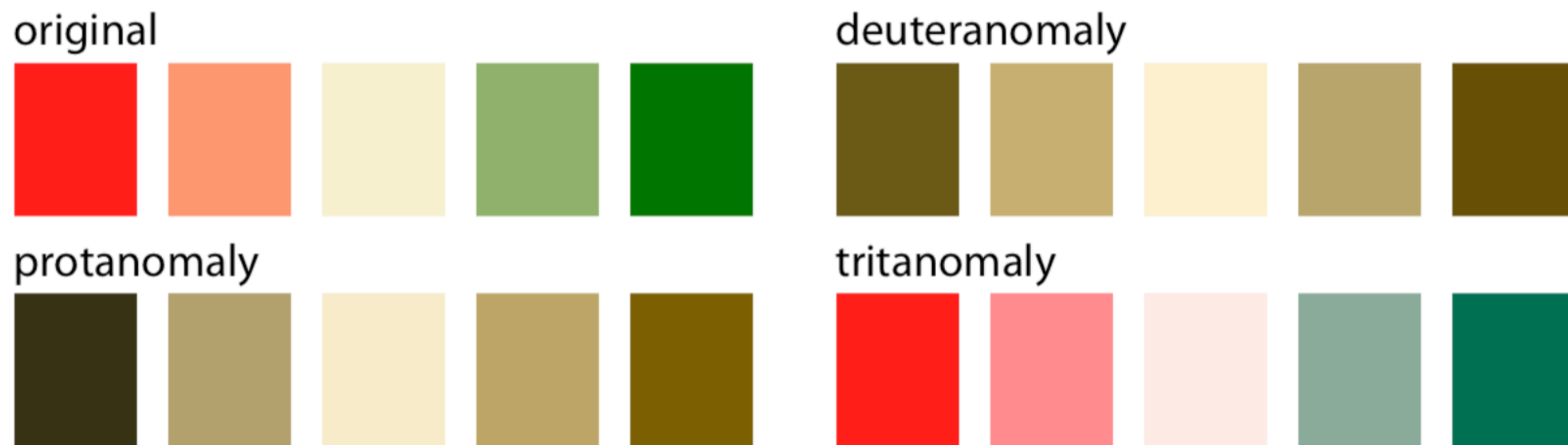


rainbow converted to grayscale



Colorblindness

- 8% of men (including myself!) and 0.5% of women are colorblind. Problematic when designing color scales.
- Most common: red-green (deuteranopia and protanopia). Also blue-yellow (tritanopia).
- Most people find that colors between green/red exhibit high contrast... I do not!
- Example: Qualitative categorical color scales:



Use colorblind friendly color palettes! It is easy!

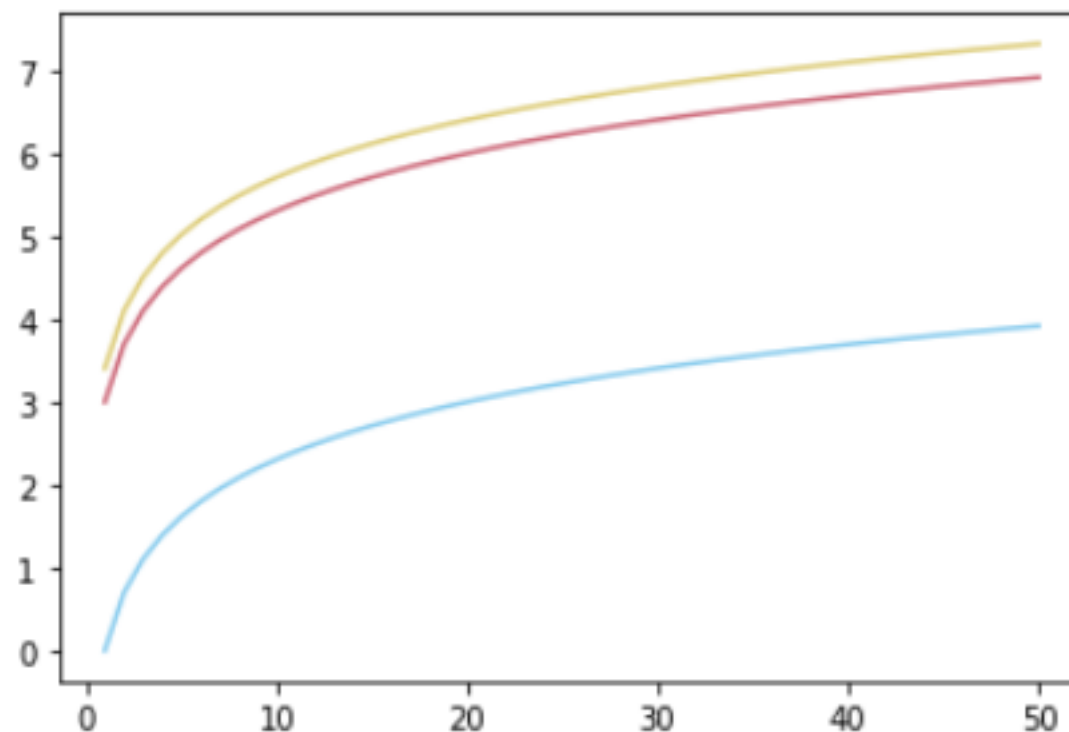
- **Use colorblind friendly color palettes! Colorbrewer2 palettes usually do a good job.**
- Install:
pip3 install palettable
- Then in Python:
`from palettable import cartocolors`
- Pre-defined, vetted color palettes for different types of data available here.

Matplotlib Example: Plot

```
# Define ColorBrewer color map palette with 3 colors  
# as a list of RGB tuples  
brewer_colors = cartocolors.qualitative.Safe_3.mpl_colors
```

```
x = np.linspace(1, 50)  
y = np.log(x)  
y_2 = np.log(20*x)  
y_3 = np.log(30*x)
```

```
plt.plot(x, y, color=brewer_colors[0])  
plt.plot(x, y_2, color=brewer_colors[1])  
plt.plot(x, y_3, color=brewer_colors[2])  
plt.show()
```



Matplotlib Example:

2D Scatterplot with color scale (plot 3D data in 2D)

```
# Define ColorBrewer color map palette
brewer_colors = cartocolors.qualitative.Safe_4.mpl_colormap

# Generate example data (N=100): (x, y) points
X = np.random.randn(100,)
Y = np.random.randn(100,)
# Each (x, y) value has an associated z int value (0, 1, 2, or 3)
Z = np.random.randint(low=0, high=4, size=(100,))

# Set the color map (cmap) to the colorbrewer one
scat = plt.scatter(X, Y, c=Z, cmap=brewer_colors)
# Show the colorbar
plt.colorbar(scat)
plt.show()
```

