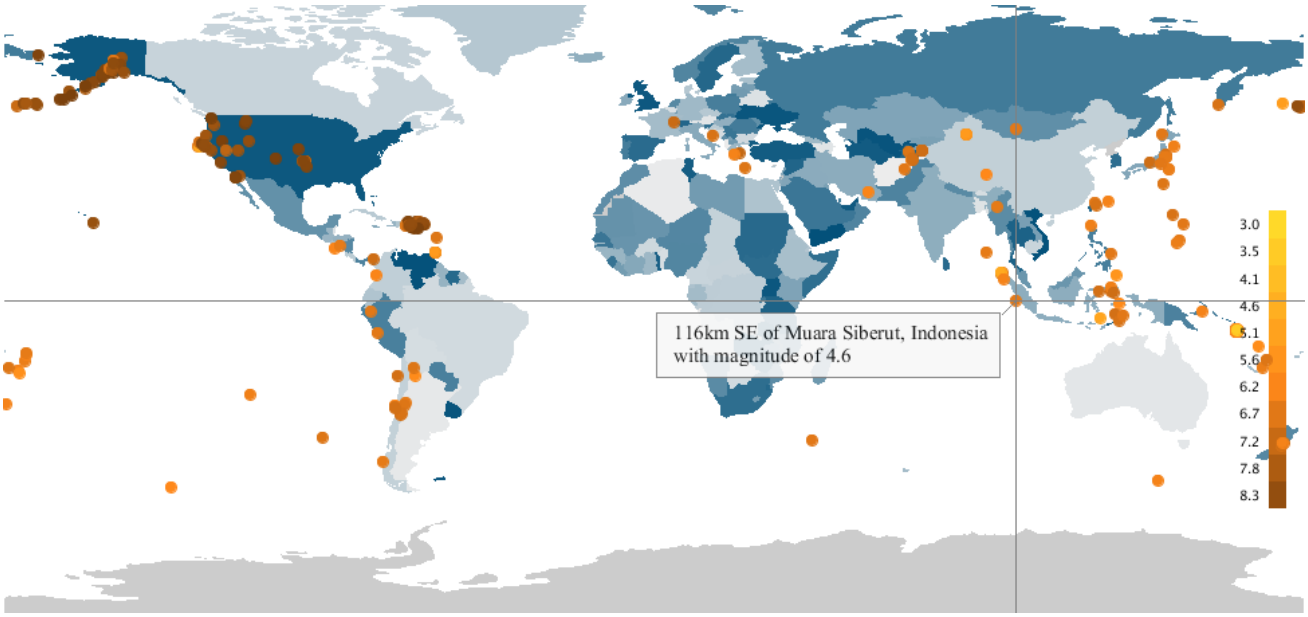
Visualization

Visualization improvements include plot annotations; new plots for thermophysical data, signal processing, and statistics; and more tools for working with colors and palettes in the ColorTools package.

Plot Annotations

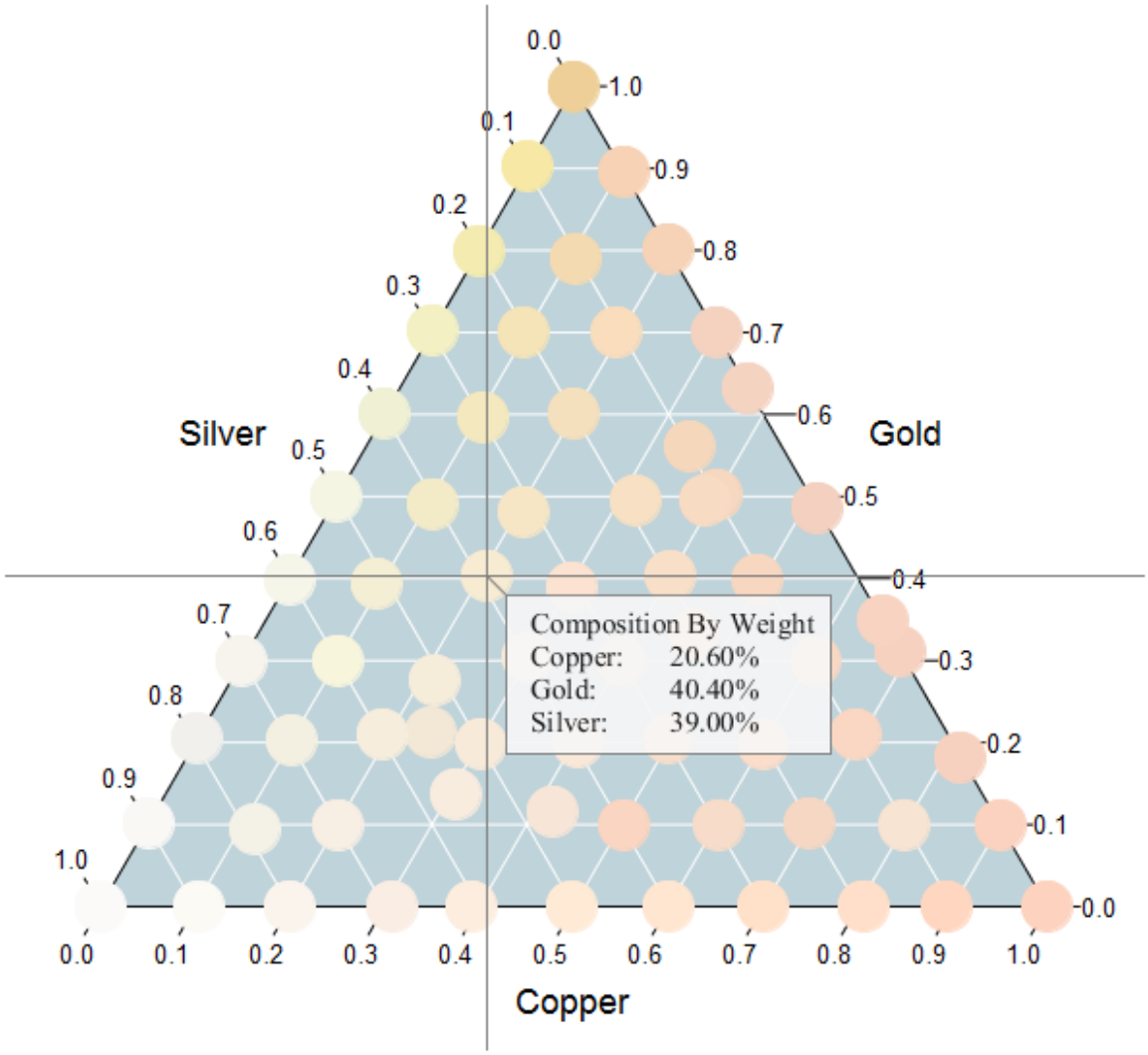
Plots can now include annotations on points and lines using the annotation plot option. Annotations appear when you mouse over a point or a line with an annotation.

In this example, the annotation gives the location and the magnitude of an earthquake:



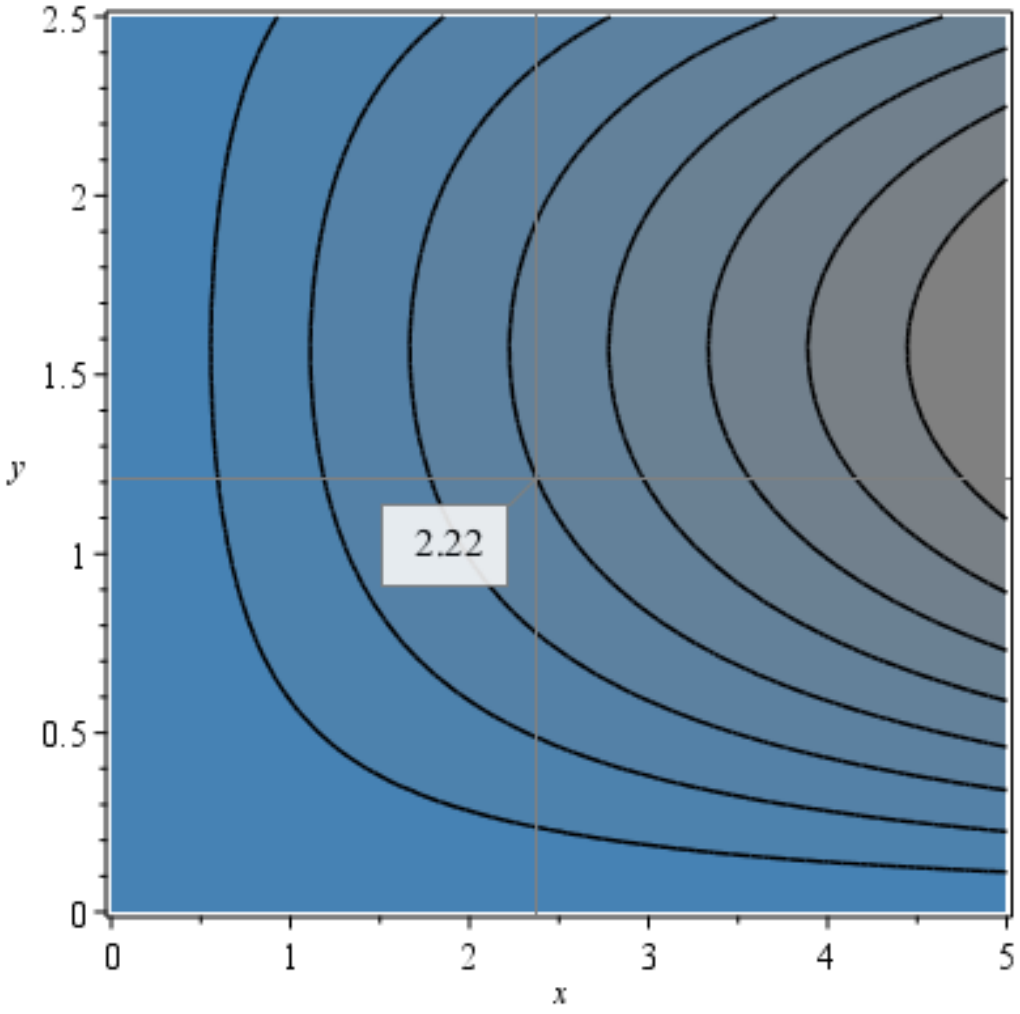
The world map for this visualization was generated by Maple 2017's new map tools.

This is a ternary plot of the color of gold-silver-copper alloys under specific lighting conditions.



Moving the mouse over the points reveals the composition of the alloy.

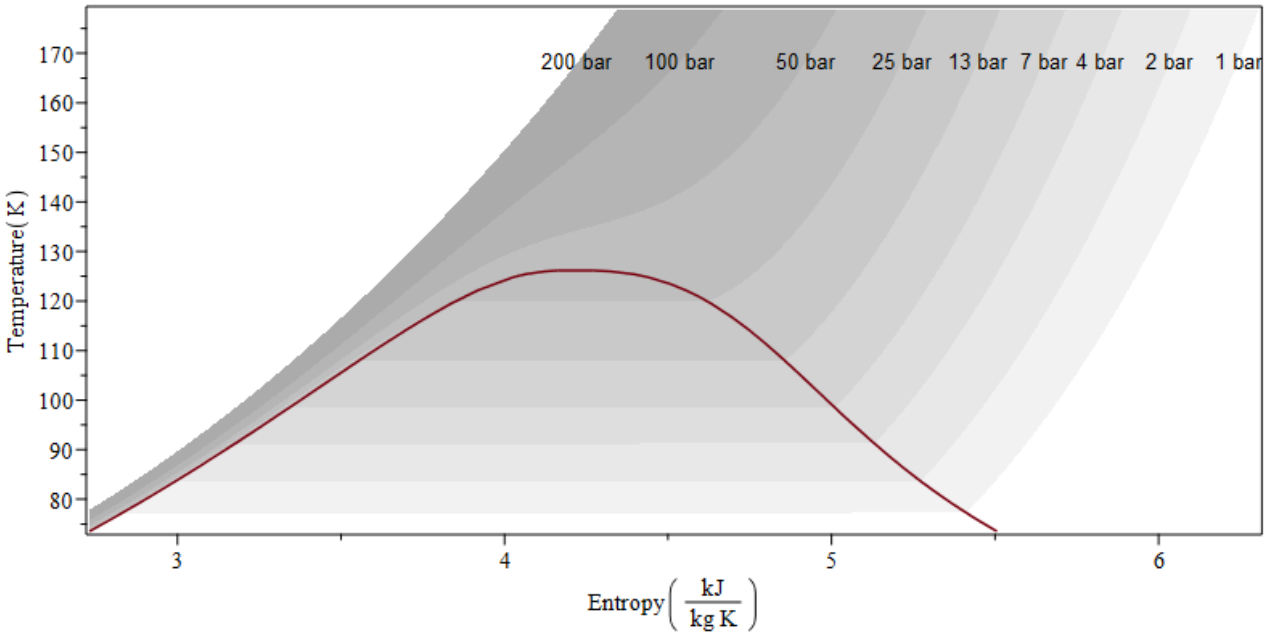
Contour plots now have labels on the contours that appear when the pointer is placed over a contour line.



You can control and customize this feature using the **contourlabels** option as described on the contour plots help page. These labels are available for 2-D contour plots.

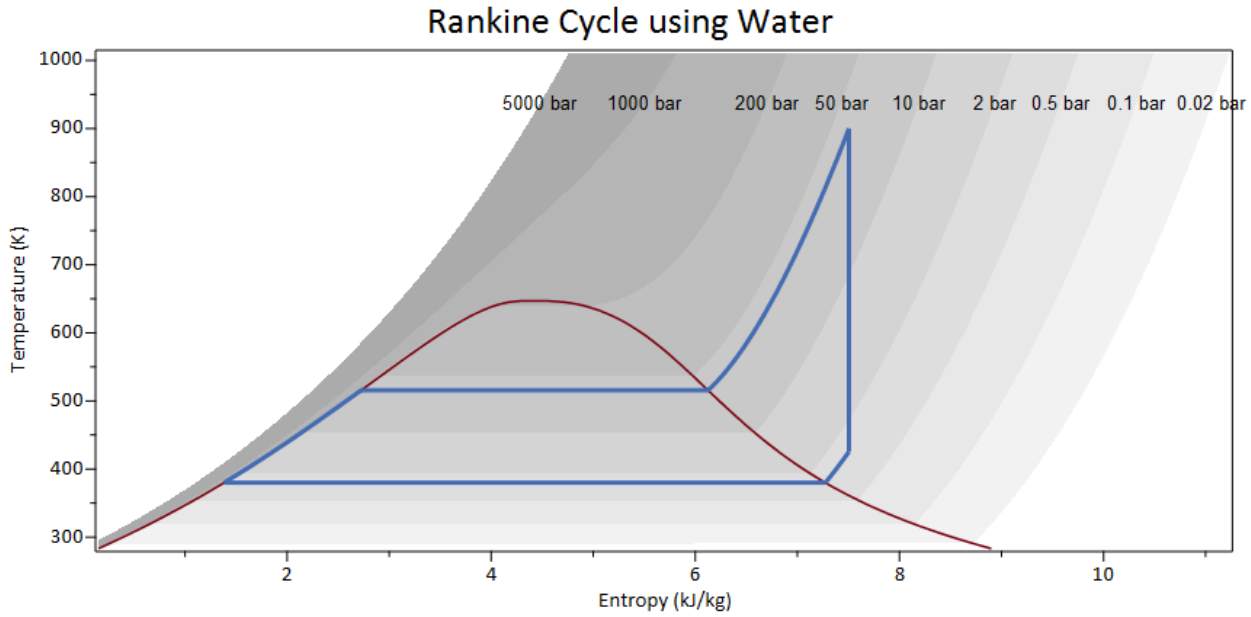
ThermophysicalData

The ThermophysicalData package now generates temperature-entropy charts for any pure fluid.



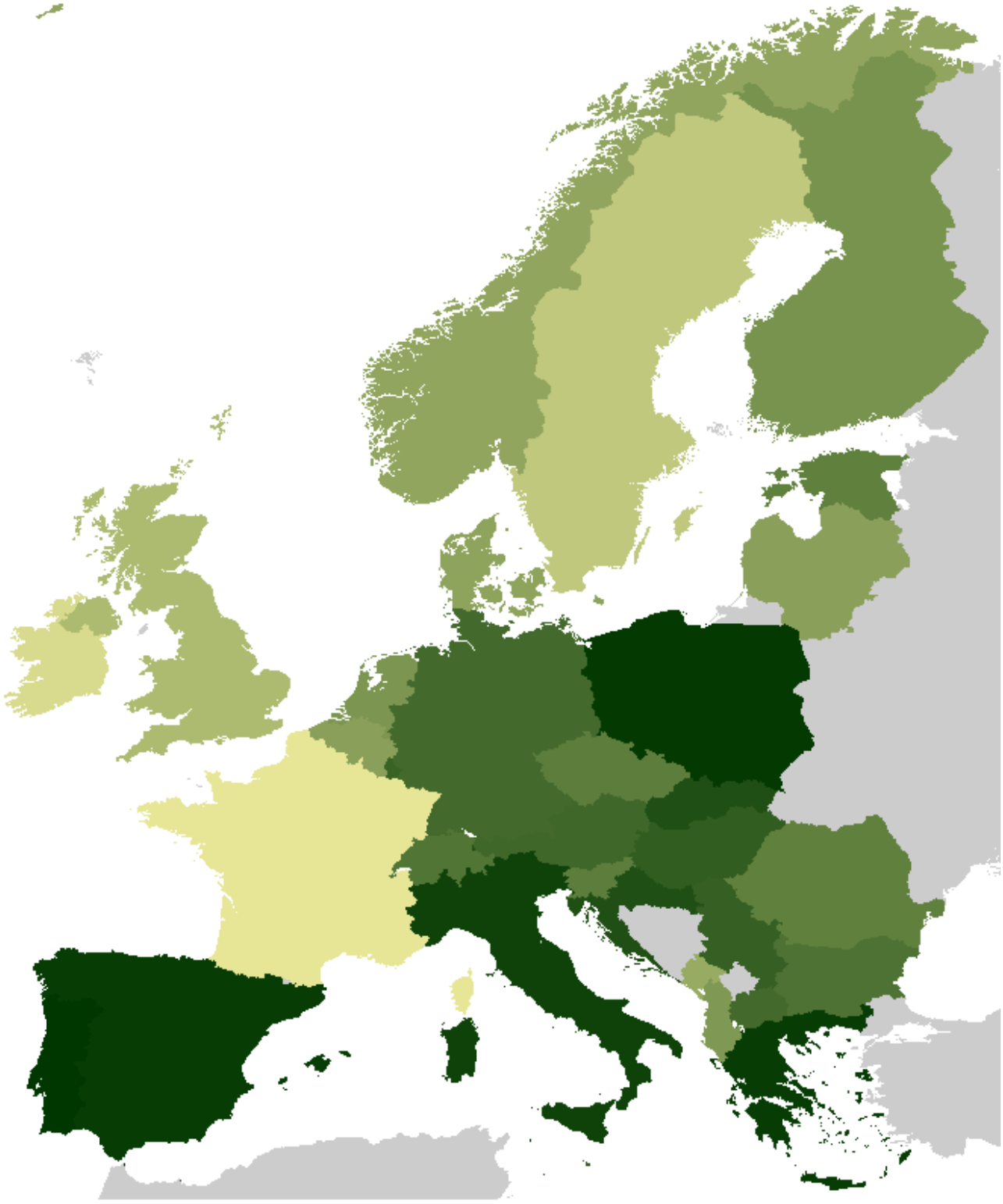
Note that the plot includes the two-phase dome. You can customize the plot: the temperature and entropy range can be changed and isobars can be modified.

Additionally, thermodynamic cycles can be visualized by placing lines on top of the chart. The following, for example, is a visualization of a Rankine cycle.



Maps

Maple 2017 features new tools for generating and customizing world maps. This map, for example, is a choropleth that visualizes European fertility rates. Lighter shades of green indicates higher levels of fertility.

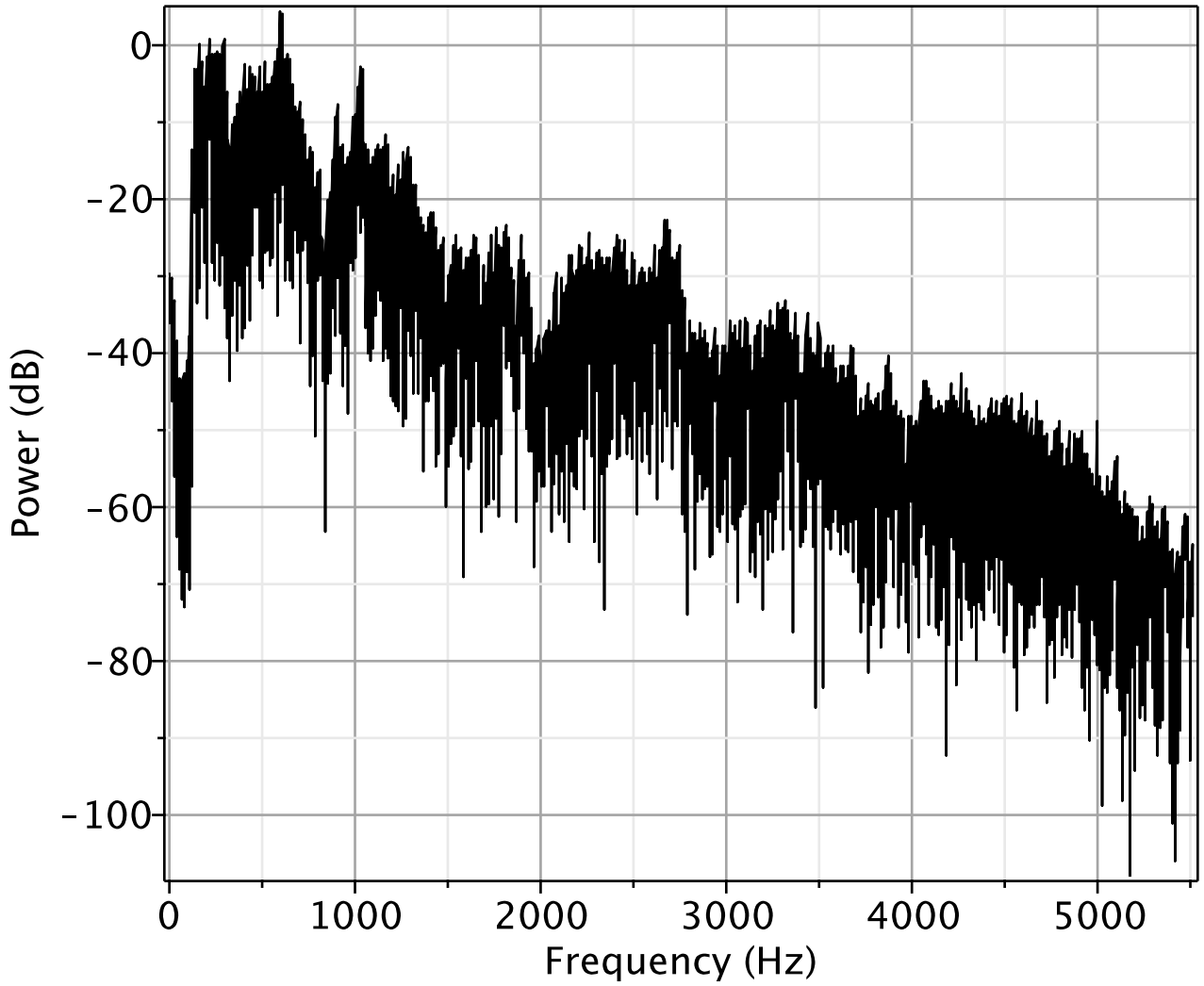


SignalProcessing

The new Periodogram function from the SignalProcessing package generates a power spectrum plot.

This, for example, is a periodogram of a speech sample.

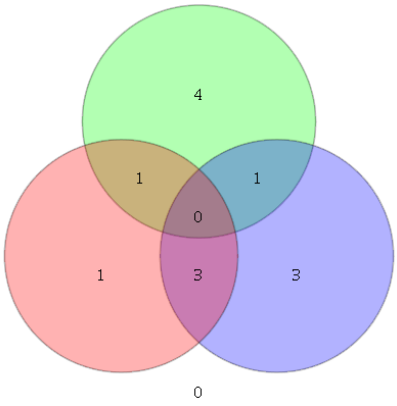
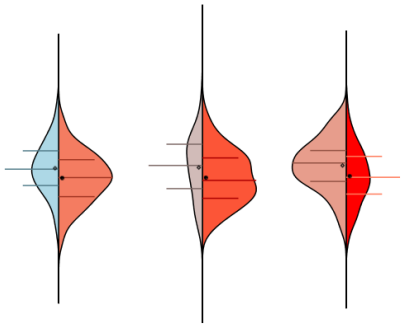
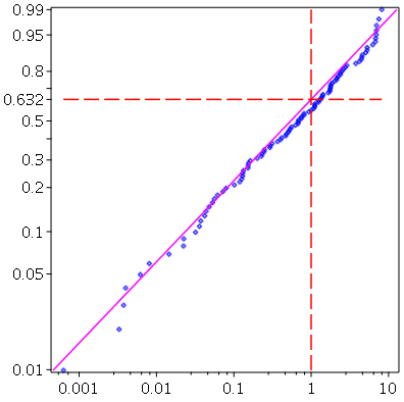
```
signal := AudioTools:-Read(cat(kernelopts(datadir), "/audio/MapleSimMono11025.wav")) :
SignalProcessing:-Periodogram(signal, size = [800, 400])
```



This new tool, when allied with the Spectrogram function, gives greater insight into the frequency contents of a signal.

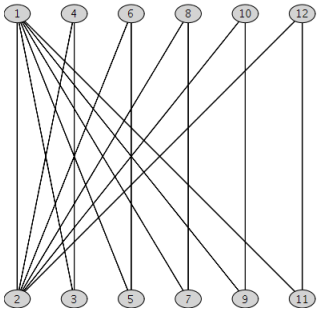
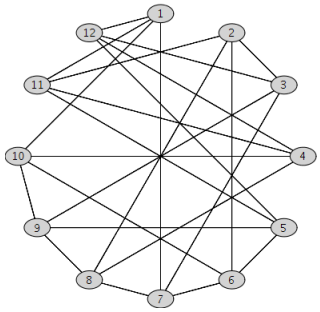
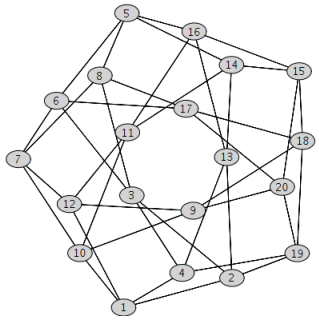
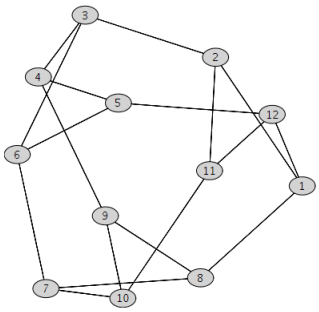
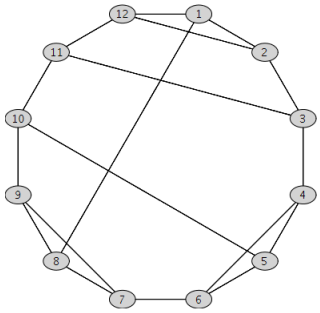
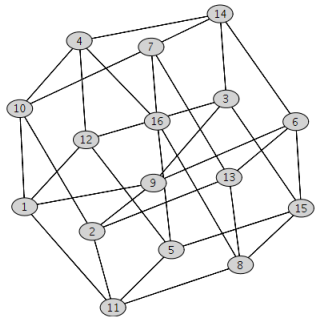
Statistics

There are many updates to existing Statistics visualizations as well as three new plots:

		
Venn Diagram	**ViolinPlot**	**WeibullPlot**

Graph Theory

The updated GraphTheory package offers several new special graphs, and features a new greyscale color scheme for the visualizations.

Book Graph	Chvatal Graph	Folkman Graph
		
Franklin Graph	**Frucht Graph**	**Hoffman Graph**
		

ColorTools

The ColorTools package has seen a few changes in this release.

$with(ColorTools)$:

There are five new commands relating to ColorTools[Color] objects: Chroma, Hue, Luma, ToPlotColor, and ToRGB. In addition, the Darken and Lighten commands both accept a new option called **best**. These commands and options can be used as follows.

$greenish := Color([0.380, 0.400, 0.027])$;

$$greenish := \langle \text{RGB : 0.38 0.4 0.027} \rangle$$

$darker := [seq(Darken(greenish, q, best), q = 0.9 .. 0.6, -0.1)]$;

$darker := [\langle \text{RGB : 0.34 0.36 0.000351} \rangle, \langle \text{RGB : 0.302 0.32 0} \rangle, \langle \text{RGB : 0.265 0.281 0.000191} \rangle, \langle \text{RGB : 0.229 0.243 0} \rangle]$

$lighter := [seq(Lighten(greenish, q, best), q = 1.2 .. 1.6, 0.2)]$;

$lighter := [\langle \text{RGB : 0.481 0.496 0.144} \rangle, \langle \text{RGB : 0.555 0.566 0.214} \rangle, \langle \text{RGB : 0.611 0.62 0.266} \rangle]$

$shades := [op(darker), greenish, op(lighter)]$;

$shades := [\langle \text{RGB : 0.34 0.36 0.000351} \rangle, \langle \text{RGB : 0.302 0.32 0} \rangle, \langle \text{RGB : 0.265 0.281 0.000191} \rangle, \langle \text{RGB : 0.229 0.243 0} \rangle, \langle \text{RGB : 0.38 0.4 0.027} \rangle, \langle \text{RGB : 0.481 0.496 0.144} \rangle, \langle \text{RGB : 0.555 0.566 0.214} \rangle, \langle \text{RGB : 0.611 0.62 0.266} \rangle]$

$lumas := map(Luma, shades)$;

$lumas := [0.3143831233, 0.27941840, 0.2452529810, 0.21193181, 0.35297, 0.45262667, 0.52411167, 0.57852909]$

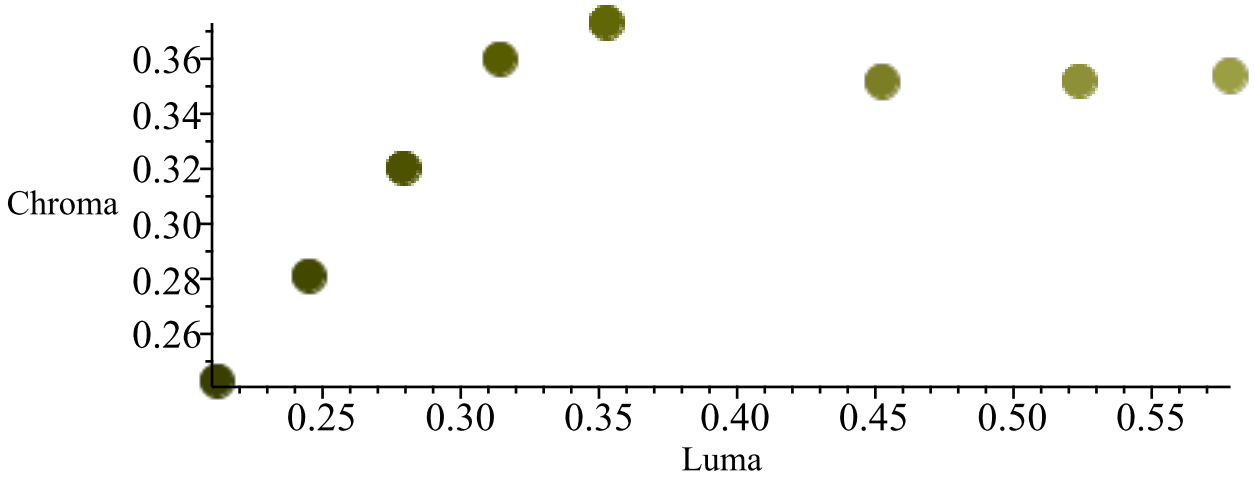
$chromas := map(Chroma, shades)$;

$chromas := [0.359709788, 0.320170, 0.280881991, 0.242659, 0.373, 0.351633, 0.351703, 0.353711]$

$coordinates := zip((x, y) \to [x, y], lumas, chromas)$;

$coordinates := [[0.3143831233, 0.359709788], [0.27941840, 0.320170], [0.2452529810, 0.280881991], [0.21193181, 0.242659], [0.35297, 0.373], [0.45262667, 0.351633], [0.52411167, 0.351703], [0.57852909, 0.353711]]$

```
plots:-pointplot(coordinates, color = shades, scaling = constrained, symbolsize = 25, symbol
    = solidcircle, size = [1000, 300], labels = ["Luma", "Chroma"]);
```

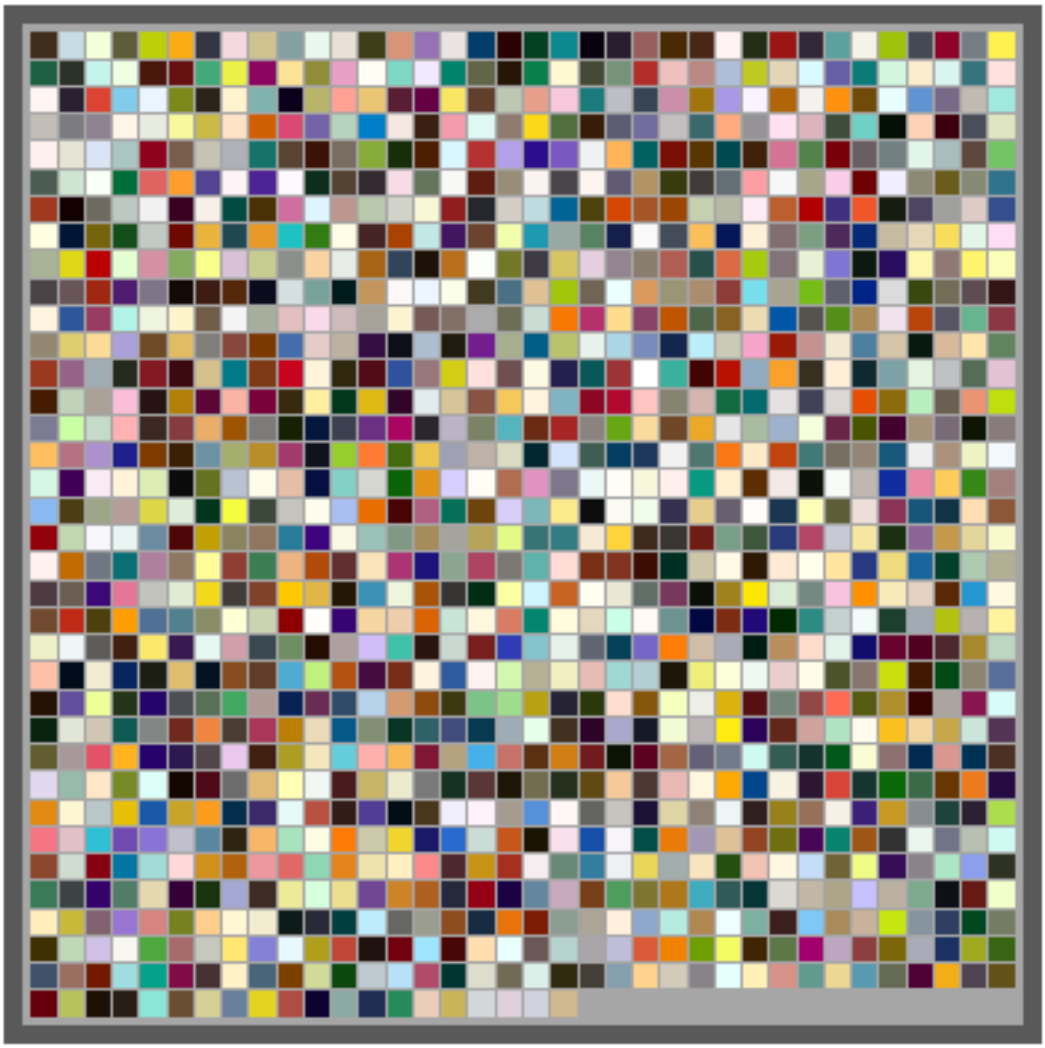


It is clear that the Darken and Lighten commands change the colors nonlinearly in this color space.

The new DisplayPalette command provides a helpful way to view the colors in a palette by displaying the names and associated colors in a table. This command is used on the Plot Color Names help page as well as on the help pages for the predefined color palettes, which are listed on the ColorTools page in the **List of Color Collections** section.

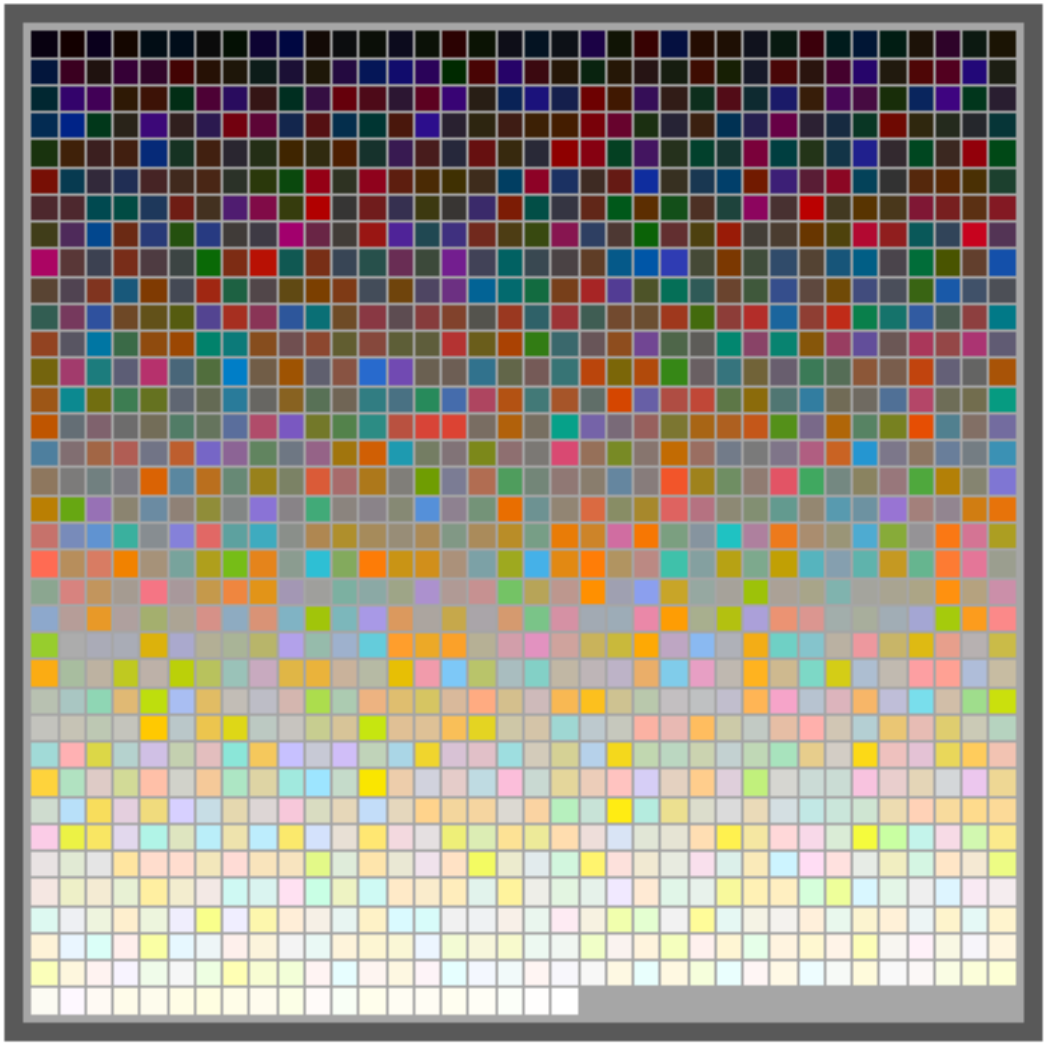
There are six other new commands relating to ColorTools[Palette] objects: GetPaletteType, GetColors, KnownColor, Lookup, numcolors, and numelems. They can be used as follows.

$RP := GetPalette(\text{"Resene"})$:



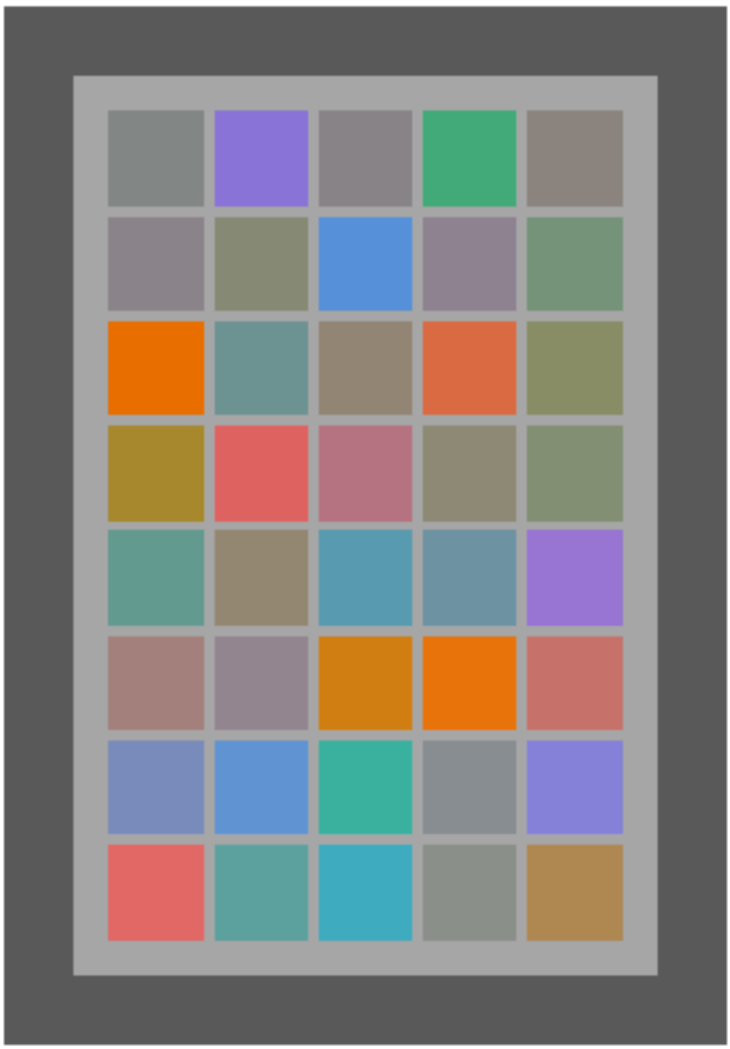
$colors := GetColors(RP)$:

$sorted := sort(colors, key = Luma)$:



Suppose we find a color we particularly like in the sorted list of colors, somewhere in the middle of the palette. We can look it up in the original palette.

$$Swatches(sorted[620..659], rows = 8);$$



clementine := *sorted*[630];

clementine := ⟨RGB : Clementine⟩

colorname := *ColorDescription*(*clementine*);

colorname := "Clementine"

Lookup(*RP*, *colorname*);

⟨RGB : Clementine⟩

PlotBuilder

The updated PlotBuilder helps you interactively build 2-D and 3-D plots without the use of commands.