Comparison of Graphics Capabilities between Mathematica 12 and Maple 2019

Summary

As in many areas of functionality, apparent similarities in visualization capabilities between Maple and Mathematica are only skin deep.

- Mathematica automates more of the graphic creation to give more accurate or more understandable results in more cases.
- Mathematica automates more sensible aesthetic choices for professional-looking results.
- Mathematica supports a wider range of visualization routines.
- Mathematica visualizations make use of dynamic elements for richer electronic presentations.
- Mathematica visualizations support a greater range of inputs.

If you want to produce professional, publication-quality graphics that are clear and accurate with the minimum amount of effort, then Mathematica is the obvious choice.

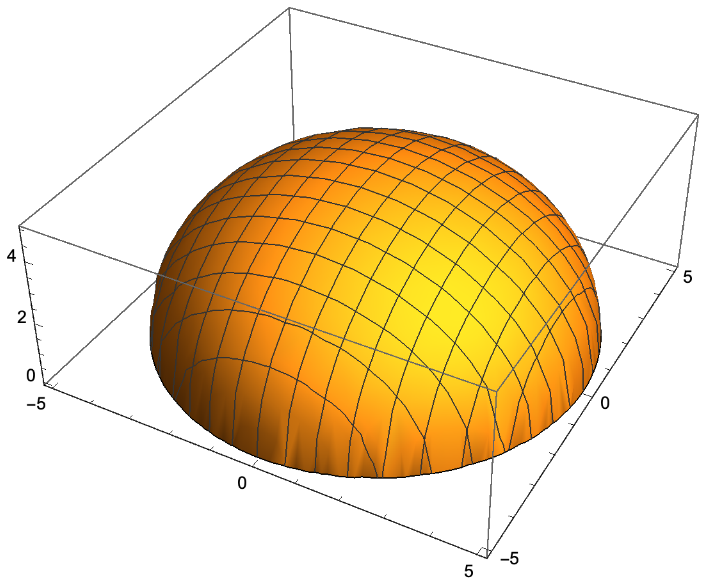
Accuracy of representation

Most Mathematica routines use adaptive resampling to achieve smooth results even where functions are rapidly changing, without the computational overhead of extra sampling where the function is not changing rapidly.

Most Maple visualizations do not. Without this capability, Maple is unable to trace the smooth circular perimeter of this plot.

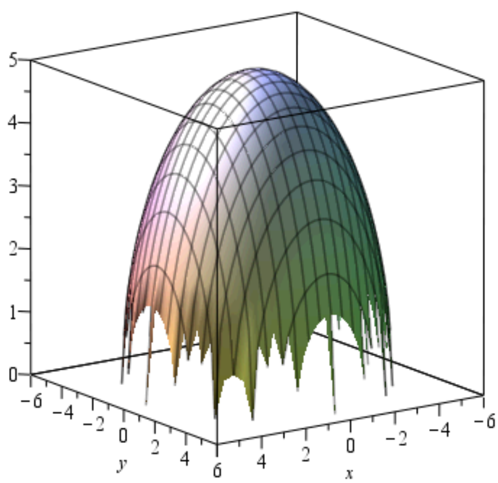
Mathematica

$\texttt{Plot3D}\left[\sqrt{25 - x^2 - y^2}, \{x, -6, 6\}, \{y, -6, 6\}\right]$



Maple

$plot3d\left(\sqrt{25 - x^2 - y^2}, x = -6..6, y = -6..6\right)$

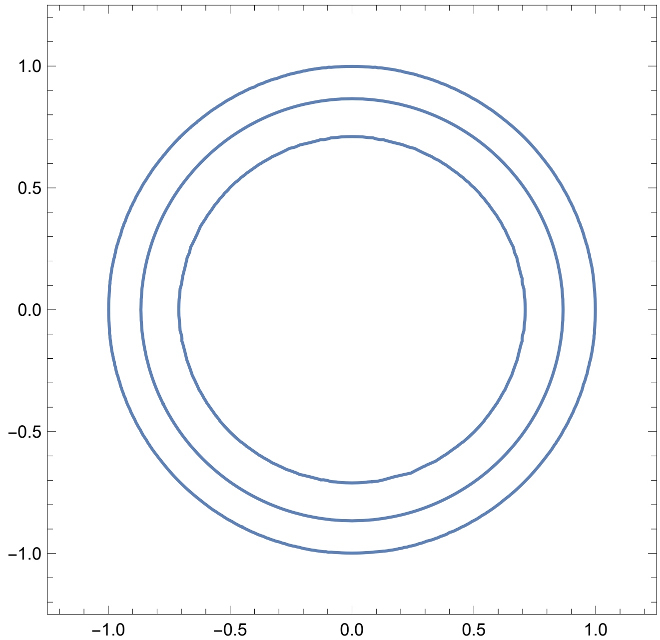


(Notice also that the Mathematica 3D images are rendered using perspective. Maple's use no perspective and appear as if viewed from an infinite distance, giving a slightly unnatural look.)

A similar accuracy problem presents itself in this simple plot of implicit functions.

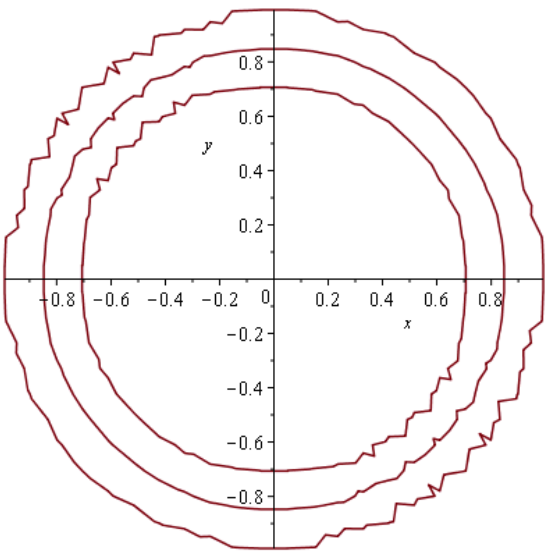
Mathematica

```
ContourPlot[(x^2 + y^2 - 1) (x^2 + y^2 - 3/4) (x^2 + y^2 - 1/2) == 0, {x, -1.2, 1.2}, {y, -1.2, 1.2}]
```



Maple

$with(plots, implicitplot):$
$fn := (x^2 + y^2 - 1) \cdot (x^2 + y^2 - 0.73) \cdot (x^2 + y^2 - 0.5):$
$implicitplot(fn, x = -1.2..1.2, y = -1.2..1.2)$

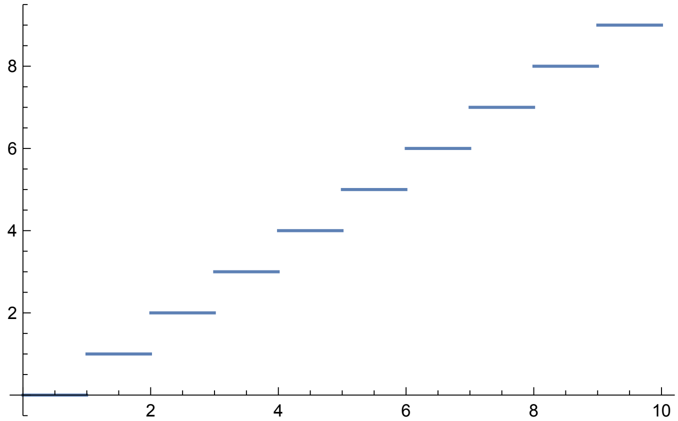


As well as failing to accurately represent smoothness in graphics, Maple does not attempt to convey important information such as discontinuities. In this simple plot of the floor function, the Maple plot seems to imply that all output values can be achieved.

Mathematica

Mathematica automatically detects many kinds of discontinuity.

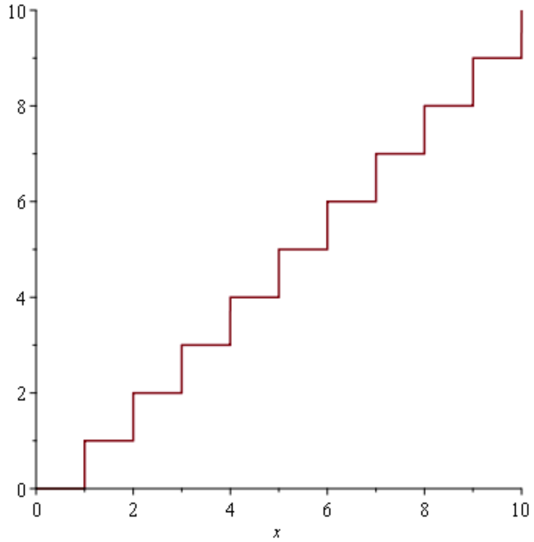
```
Plot[Floor[x], {x, 0, 10}]
```



Maple

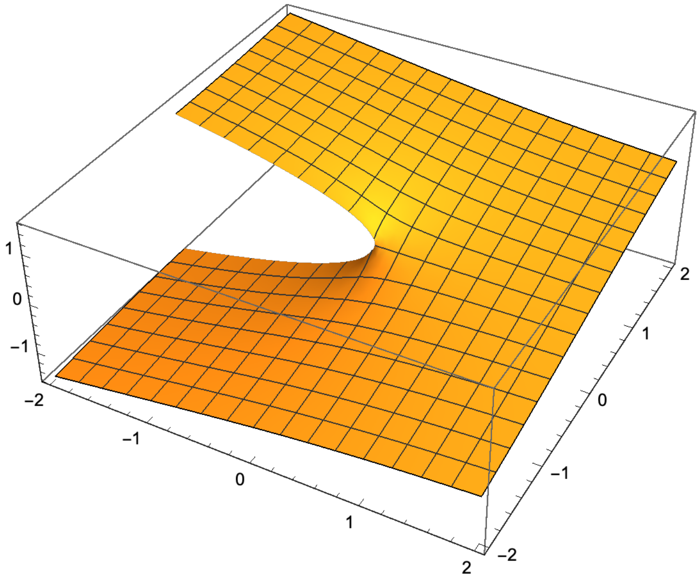
Maple incorrectly implies that there is a value of floor(x) = $\frac{1}{2}$.

$plot(\text{floor}(x), x = 0..10)$



Mathematica automatically detects many other kinds of branch cuts and discontinuities.

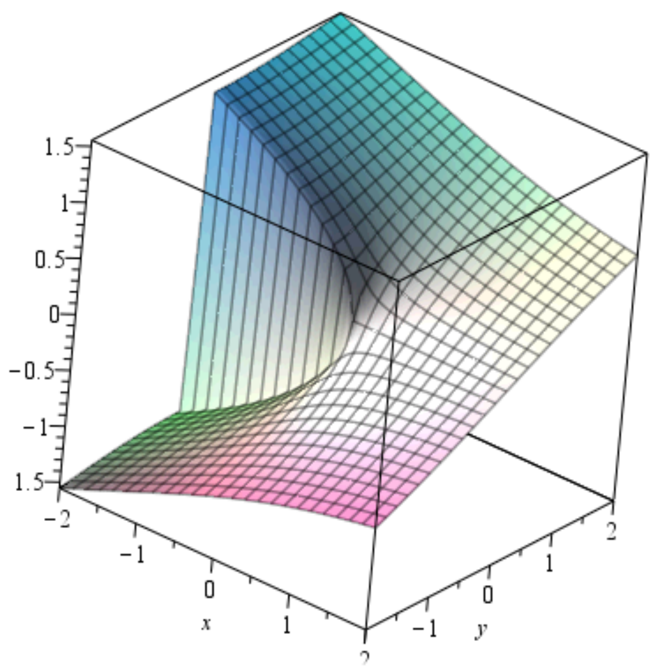
Plot3D[Im[$\sqrt{\mathtt{x + I\ y}}$], {x, -2, 2}, {y, -2, 2}]



Maple

Maple incorrectly joins up the branch cut to make the function appear continuous.

plot3d(Im(sqrt($x + I \cdot y$)), $x = -2..2$, $y = -2..2$)



Note: The output of the Maple plot has been manually rotated to match the Mathematica viewpoint.

Clarity of presentation

Through careful choices of defaults and the application of automatic optimizations, Mathematica graphics are designed to be easy to interpret.

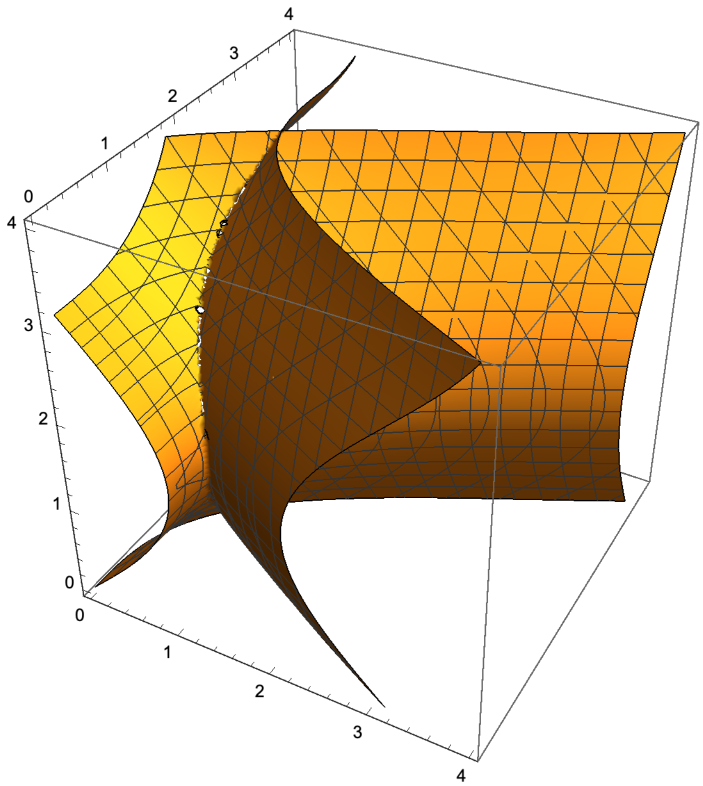
Using mesh lines to enhance interpretation

Mesh lines are a very important part of conveying meaning in 3D graphics, but in Maple they are simply a side effect of sampling.

In this simple implicit equation plot, Mathematica's choice of mesh lines enhances our understanding of the surface curvature.

Mathematica

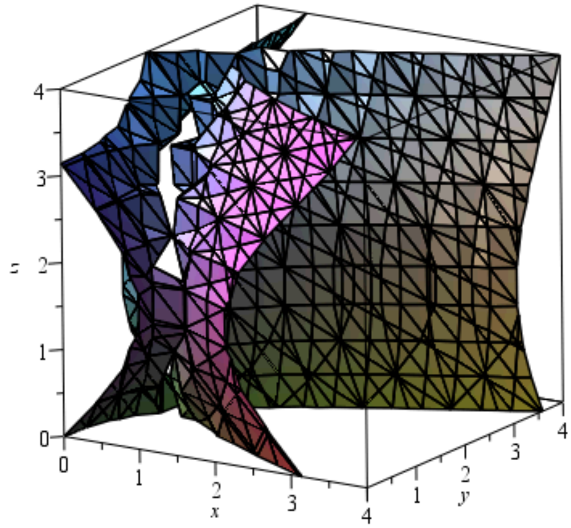
```
ContourPlot3D[Sin[x + Sin[y]] == Sin[y + Sin[z]], {x, 0, 4}, {y, 0, 4}, {z, 0, 4}]
```



Maple

However, in Maple, many unnecessary mesh lines obscure that information. Combined with the lack of adaptive sampling, especially around the intersection of the two surfaces, this makes the plot difficult to interpret.

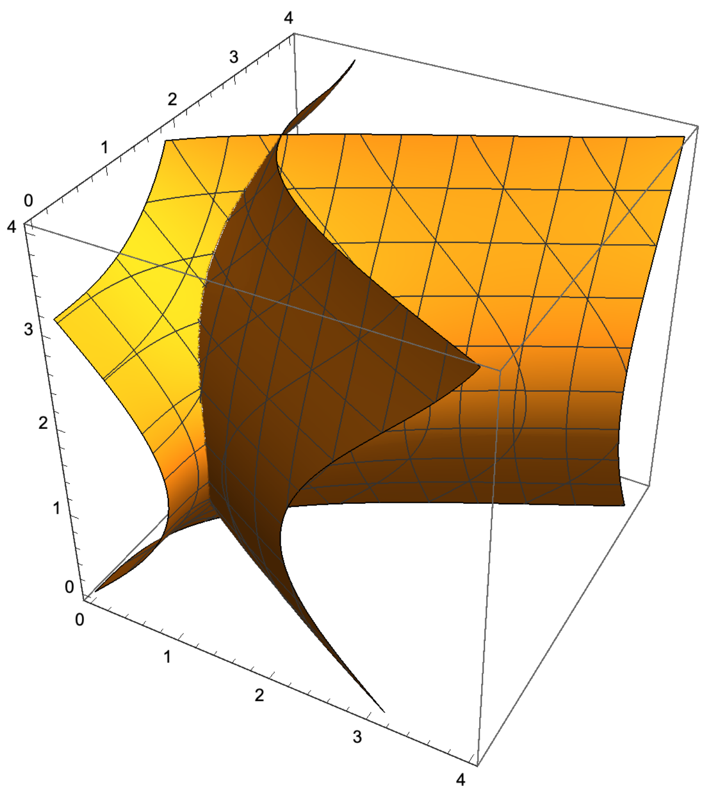
```
implicitplot3d(sin(x + sin(y)) = sin(y + sin(z)), x = 0 ..4, y = 0 ..4 , z =-0 ..4)
```



Mathematica

In Mathematica, mesh lines are independent of sampling, so we can simultaneously increase the quality of the sampling and choose a sparser set of mesh lines.

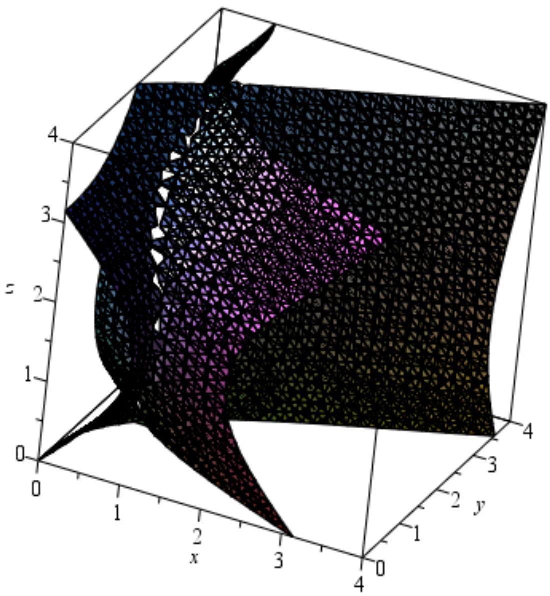
```
ContourPlot3D[Sin[x + Sin[y]] == Sin[y + Sin[z]],
 {x, 0, 4}, {y, 0, 4}, {z, 0, 4}, PlotPoints → 25, Mesh → 8]
```



Maple

In Maple, increasing the sampling introduces even more mesh lines until they obscure the plot itself. The only recourse is to turn mesh lines off completely.

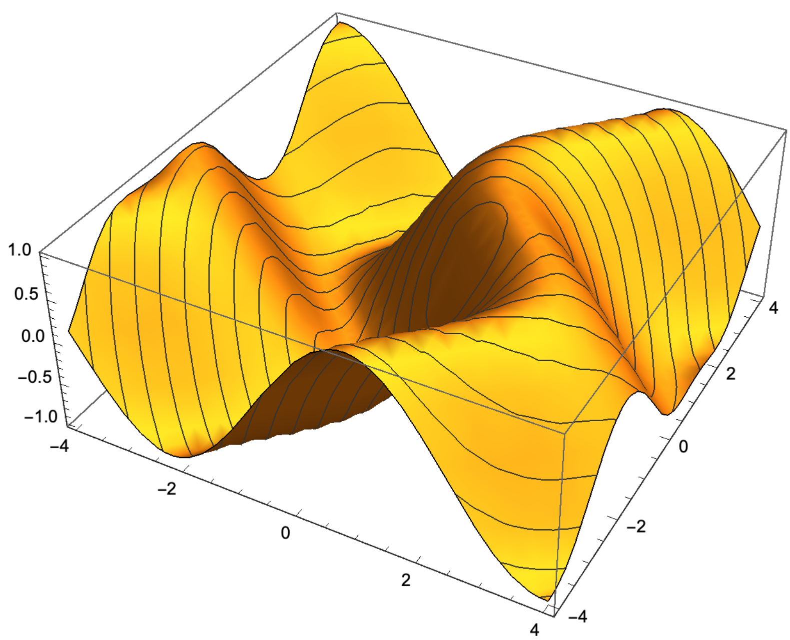
implicitplot3d(sin(x + sin(y)) = sin(y + sin(z)), x = 0 ..4, y = 0 ..4 , z =−0 ..4, *grid* = [25, 25, 25])



Most Mathematica 3D visualization routines give optional arbitrary control over mesh lines. For example, here Mathematica is instructed to place meshes at isolines in distance from {1,0,0}. In Maple, only sample point meshes and *z*-contour lines are available.

Mathematica

```
Plot3D[Sin[x + Sin[y]], {x, -4, 4}, {y, -4, 4},
 MeshFunctions → (EuclideanDistance[{#1 - 1, #2, #3}, {0, 0, 0}] &)]
```

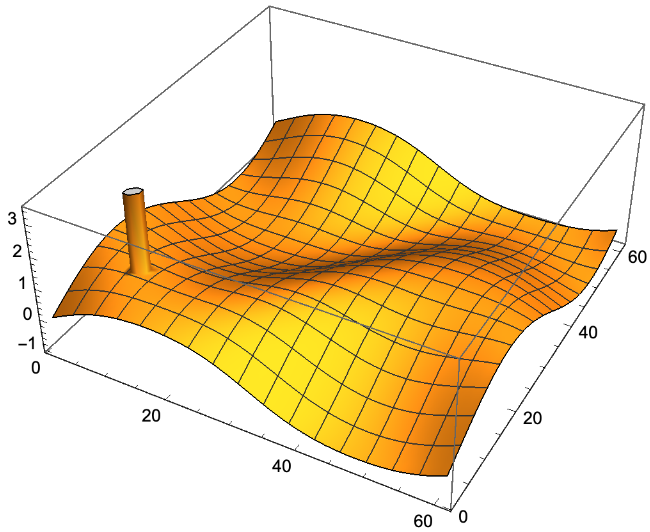


Automatic plot ranges

Automatic plot range choices ensure that the maximum amount of useful information is included in the plot. In this example, we generate a table of values for a curved surface with a single, strong outlier. Mathematica's automatic plot range preserves most of the interesting detail in the image at the expense of the outlier.

Mathematica

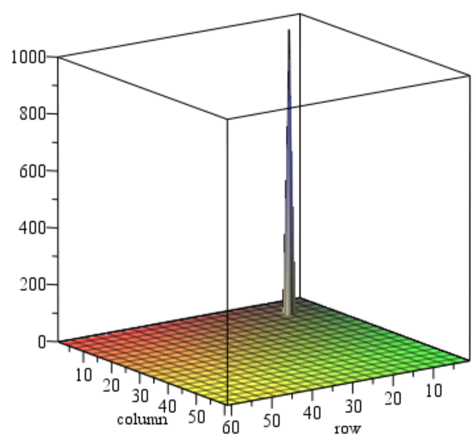
```
data = Table[Sin[x + Sin[y]], {x, 0, 6, 0.1}, {y, 0, 6, 0.1}];
data[[10, 10]] = 1000;
ListPlot3D[data]
```



Maple

However, all useful detail in the Maple plot is lost in order to include the single outlier.

```
data := [seq([seq(sin(x + sin(y)), x = 0 ..6, 0.1)], y = 0 ..6, 0.1)] :
data[10, 10] := 1000 :
with(plots) :
matrixplot(data)
```

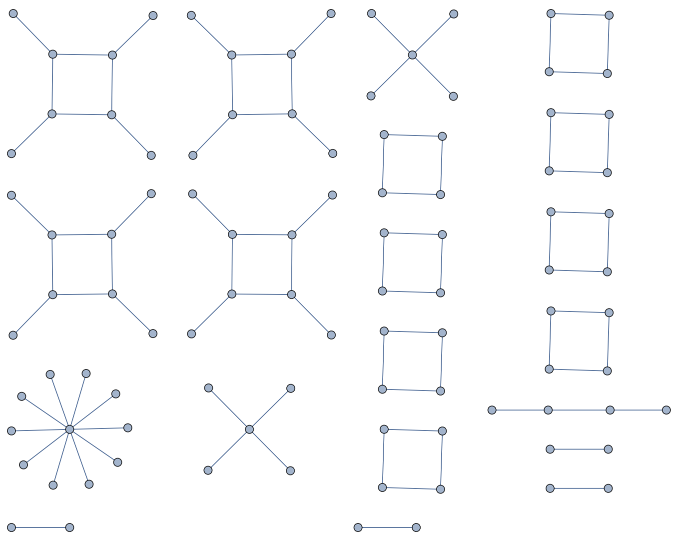


Automated layout

By using more sophisticated default algorithms, Mathematica is able to produce a much clearer visualization of this simple network graph.

Mathematica

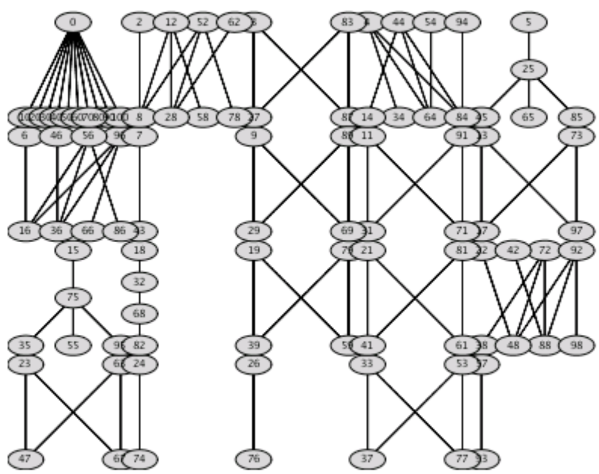
```
GraphPlot[ network data ]
```



Maple

However, when presented with identical data, Maple does not optimize the layout to prevent edges crossing or vertex labels overlapping.

with(*GraphTheory*) : *DrawGraph*(*network*)



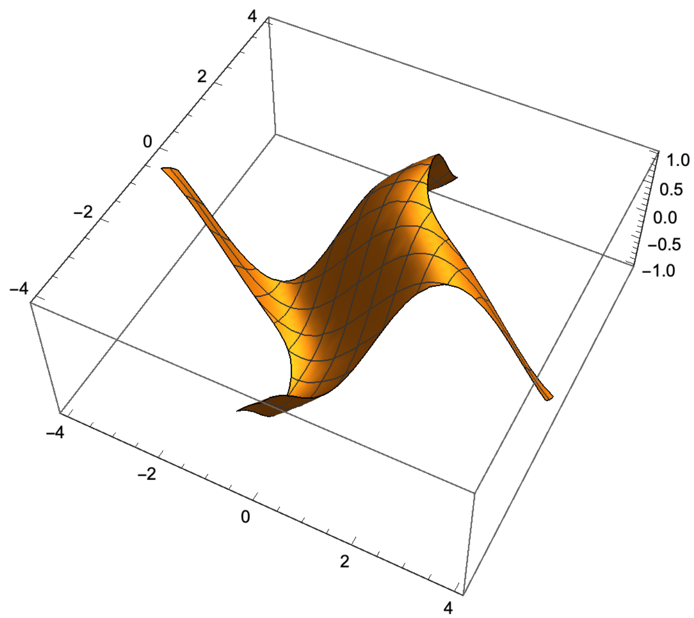
Plot region support

Often the region in which data or functions are valid is an important piece of context, and visualizations should be able to present that information.

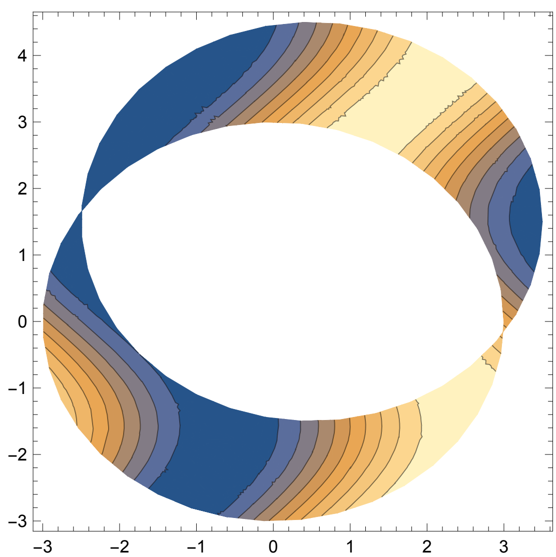
In Maple, the region over which visualizations can be created is always either rectangular or (for function plots) one in which the second independent variable is an interval that is a function of the first variable. Mathematica visualizations can be created over any region, specified implicitly or explicitly, or by using geometric constructs or arbitrary meshes. The following visualizations would be very hard to create in Maple.

Mathematica

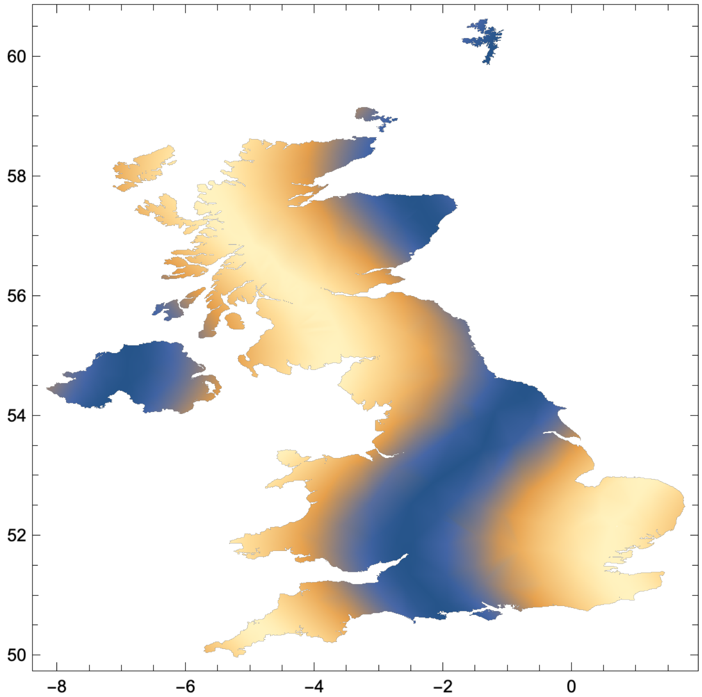
```
Plot3D[Sin[x + Sin[y]], {x, -4, 4}, {y, -4, 4}, RegionFunction → (-1 < #1 #2 < 1 &)]
```




```
ContourPlot[Sin[x + Sin[y]],
 {x, y} ∈ RegionSymmetricDifference[Disk[{0, 0}, 3], Disk[{0.5, 1.5}, 3]]]
```



```
DensityPlot[Sin[x + Sin[y]],
 {x, y} ∈ DiscretizeGraphics[CountryData["UnitedKingdom", "Polygon"]]]
```

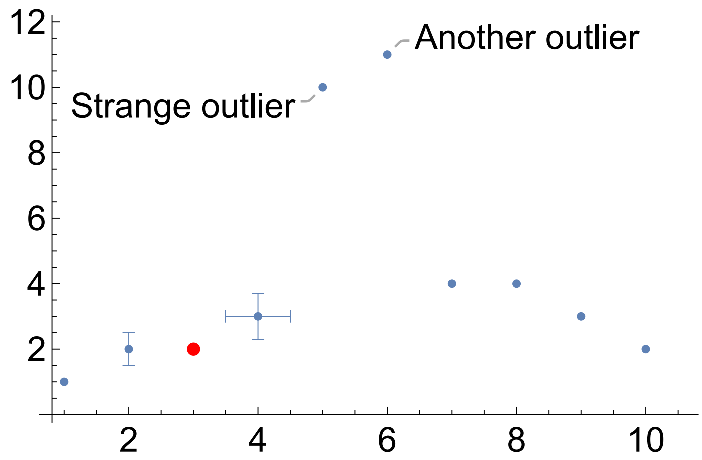


Labeling

While both Maple and Mathematica can be programmed to place text on a graphic, only Mathematica automates this process with symbolic wrappers for data points. Labels, callouts, tooltips, status area updates, error bars and mouseover effects are supported. Callout placement is automated to avoid overlapping text. Maple supports only tooltips.

```
data = {1, Around[2, 0.5],
   Style[2, Red, PointSize[0.02]],
   Around[3, {{0.5, 0.5}, {0.7, 0.7}}],
   Callout[10, "Strange outlier"],
   Callout[11, "Another outlier"],
   4, 4, 3, 2};
```

```
ListPlot[data, PlotRange → All, BaseStyle → 18]
```

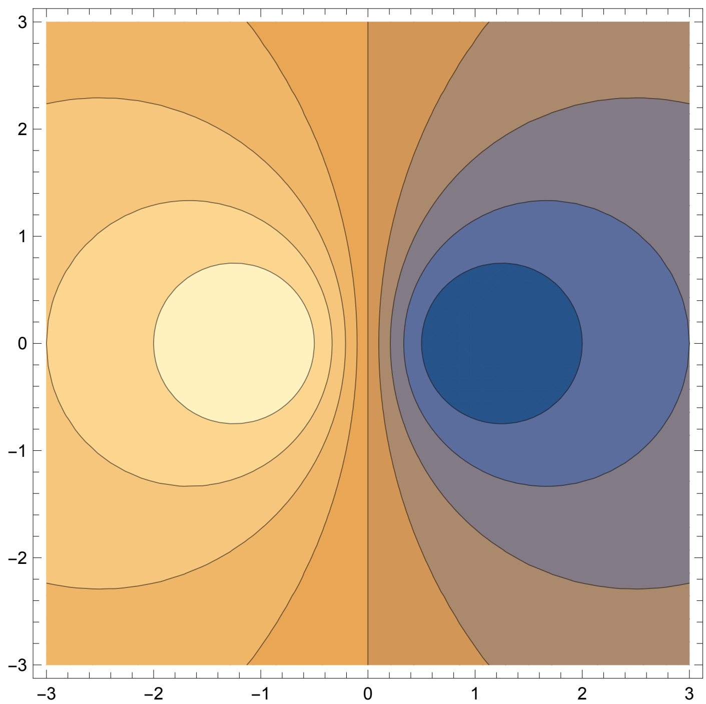


Enhancing interpretation

Simple use of shading makes it easier to understand that this contour plot represents a peak on the left and a valley on the right.

Mathematica

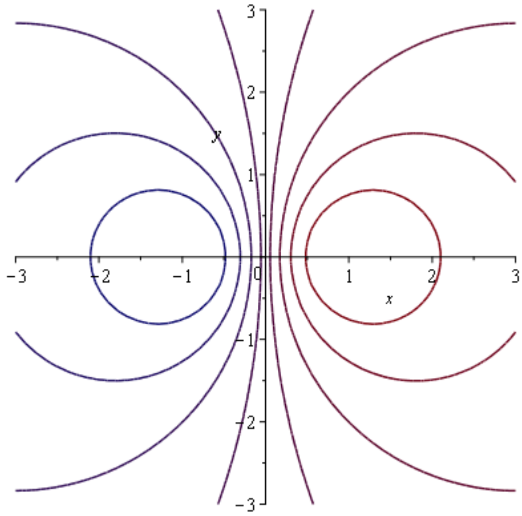
$$\texttt{ContourPlot}\left[-\frac{5\,x}{x^2+y^2+1},\ \{x,-3,3\},\ \{y,-3,3\}\right]$$



Maple

In Maple, default coloring is more subtle and uses lighter shades to represent lower values.

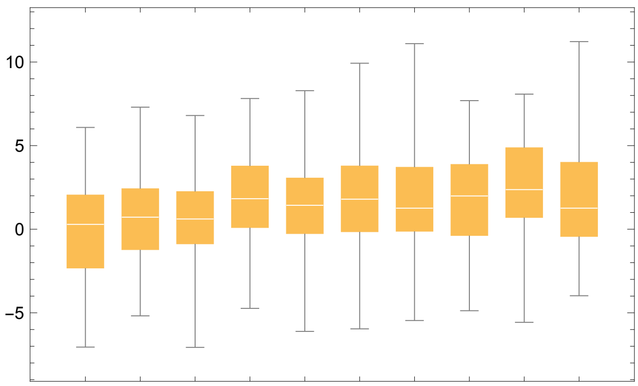
$with(plots) : contourplot\left(-\frac{5\,x}{x^2+y^2+1}, x=-3..3, y=-3..3\right)$



Equally, the unnecessary use of color can have a negative effect. In these box-and-whisker charts comparing 10 datasets, Mathematica treats each of the datasets with a uniform style. Maple chooses to color each box differently. This coloring does not represent any useful information and raises the risk of erroneous interpretation from the user (e.g. that it represents means or ranges).

Mathematica

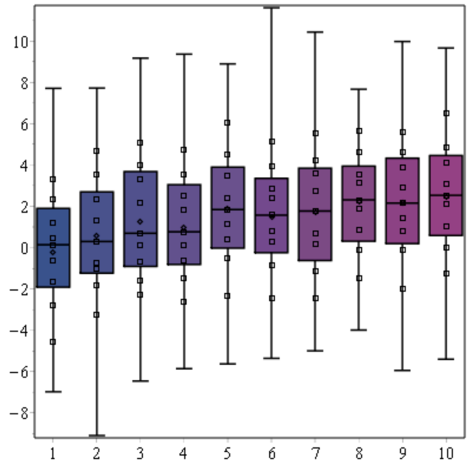
```
BoxWhiskerChart[
 data = Table[RandomVariate[NormalDistribution[Log[i], 3], 100], {i, 10}]]
```



Maple

Maple does not use tooltips in any graphics except **`contourplot`**. The graduated color scheme used by Maple conveys no additional meaning and is there only for questionable aesthetic appeal.

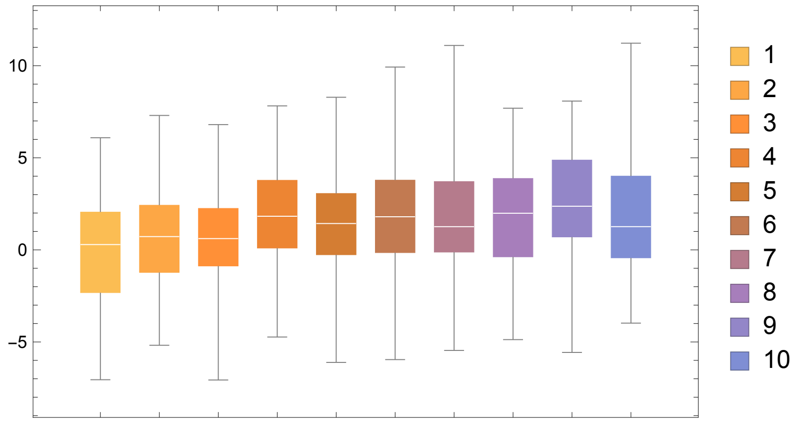
```
with(Statistics) :
A := [seq(Sample(Normal(ln(i), 3), 100), i = 1 ..10)] :
BoxPlot(A)
```



Mathematica

A more useful reason to color the datasets would be for identification, but the Maple colors are too subtle for this use (and Maple's `BoxPlot` function does not support the legend option). When Mathematica chooses colors for identifcation, they are clearly distinguishable.

```
BoxWhiskerChart[{data}, ChartLegends → Range[10]]
```



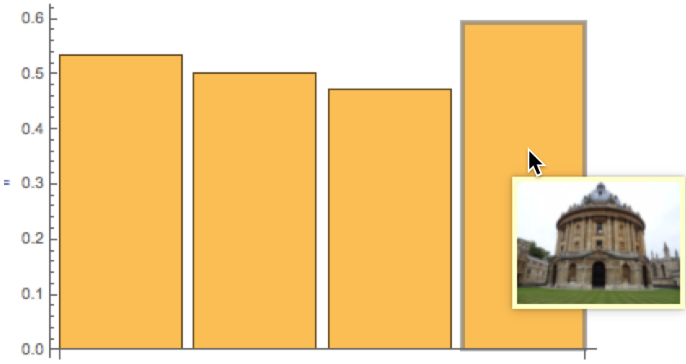
In addition to the clearer static presentation, if you move your mouse over the individual box-whisker elements, tooltips appear automatically.

max	13.4124
75%	4.99292
median	3.22043
25%	0.39816
min	−4.65948

Mathematica does this automatically for `DistributionChart` and `ContourPlot`. Maple provides this automatically only for its `contourplot`.

Both systems provide the ability to add user-specified text tooltips, but Mathematica also allows aribitrary content in tooltips such as images, typeset math or other graphics. Mathematica goes further, allowing arbitrary programmatic actions on mouse clicks, double-click, entry, exit or drag.

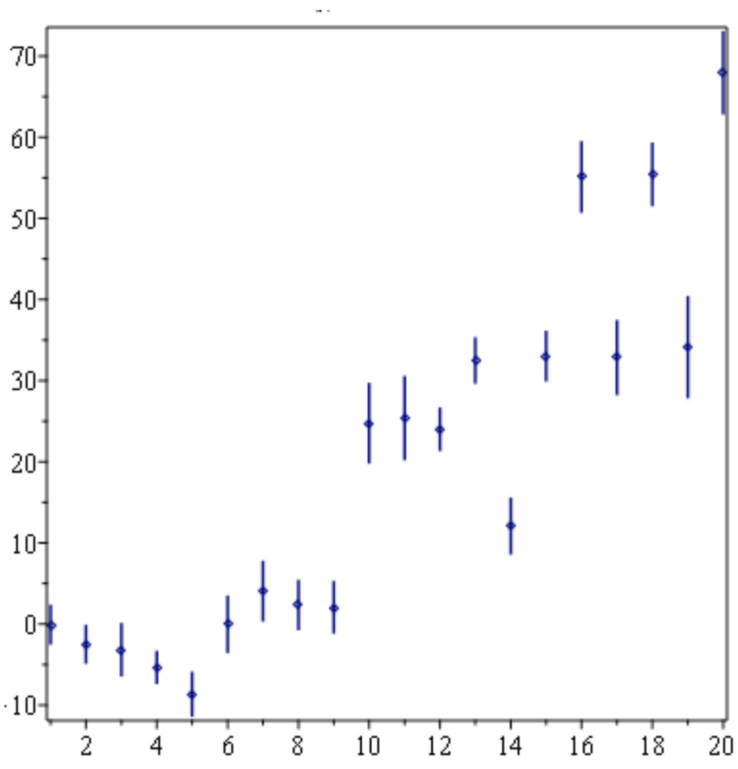
```
BarChart[Tooltip[ImageMeasurements[#, "MeanIntensity"], ImageResize[#, 100]] & /@
  ExampleData[{"TestImage", #}] /@ {"Marruecos", "Moon", "Peppers", "RadcliffeCamera"}]
```



Error bars

Maple provides error bars only for its basic point-plotting function and only for uncorrelated errors.

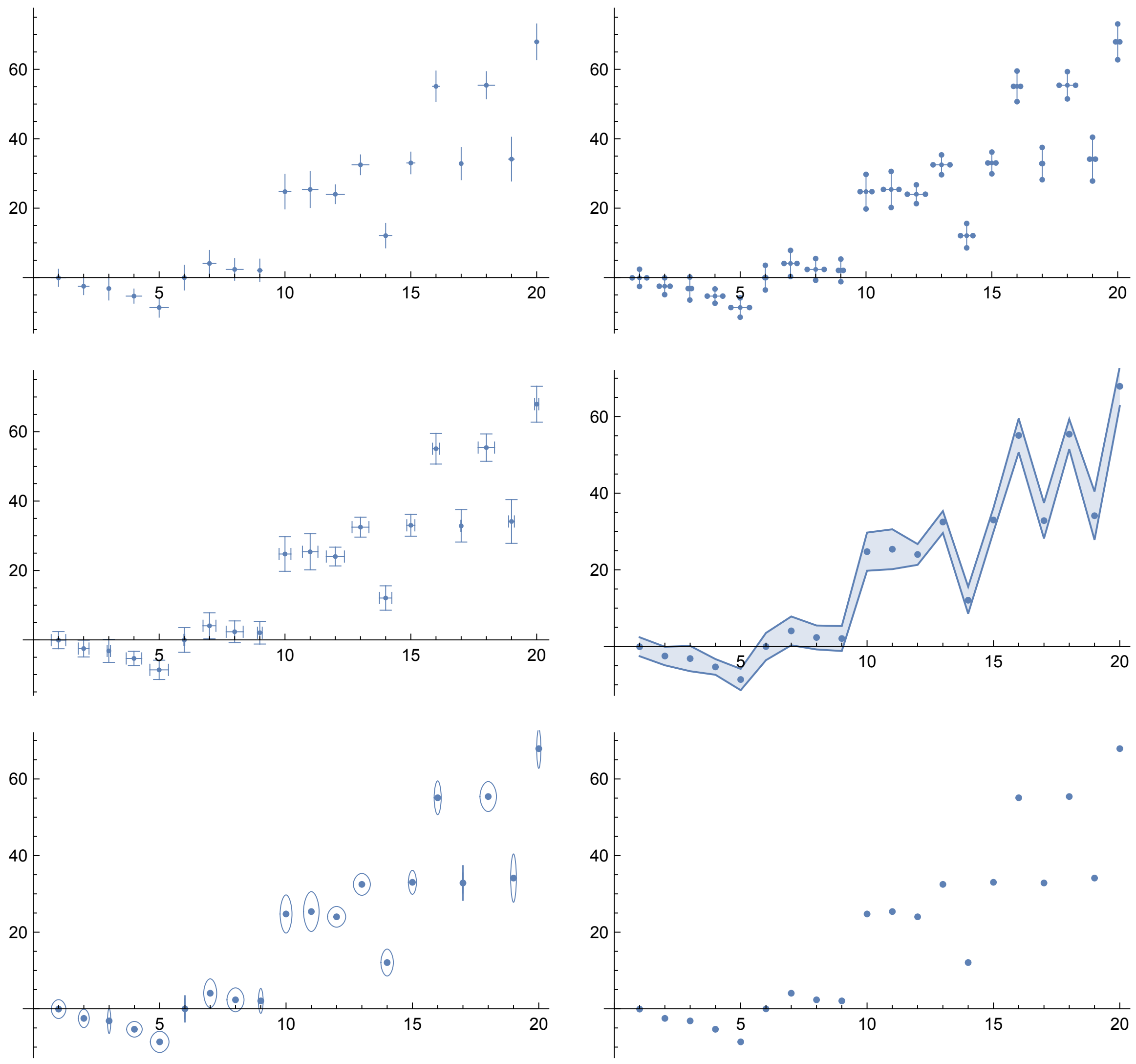
Maple



Mathematica

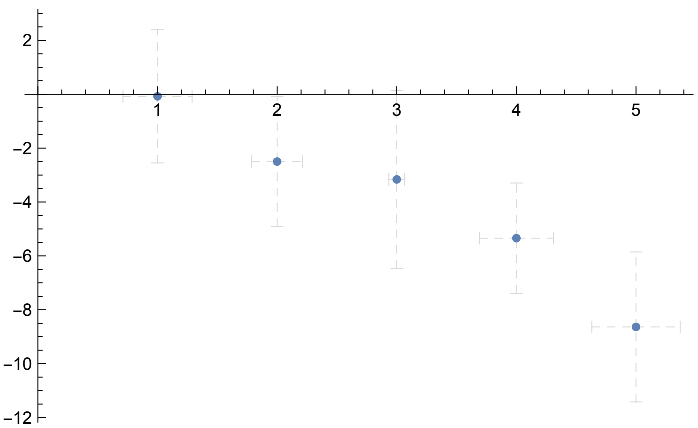
Like many of Mathematica's visualization capabilites, interval markers are provided in a range of built-in styles.

```
GraphicsGrid[Partition[Table[ListPlot[ Data , IntervalMarkers → i, ImageSize → 300],
  {i, {"Bars", "Points", "Fences", "Bands", "Ellipses", None}}], 2]]
```



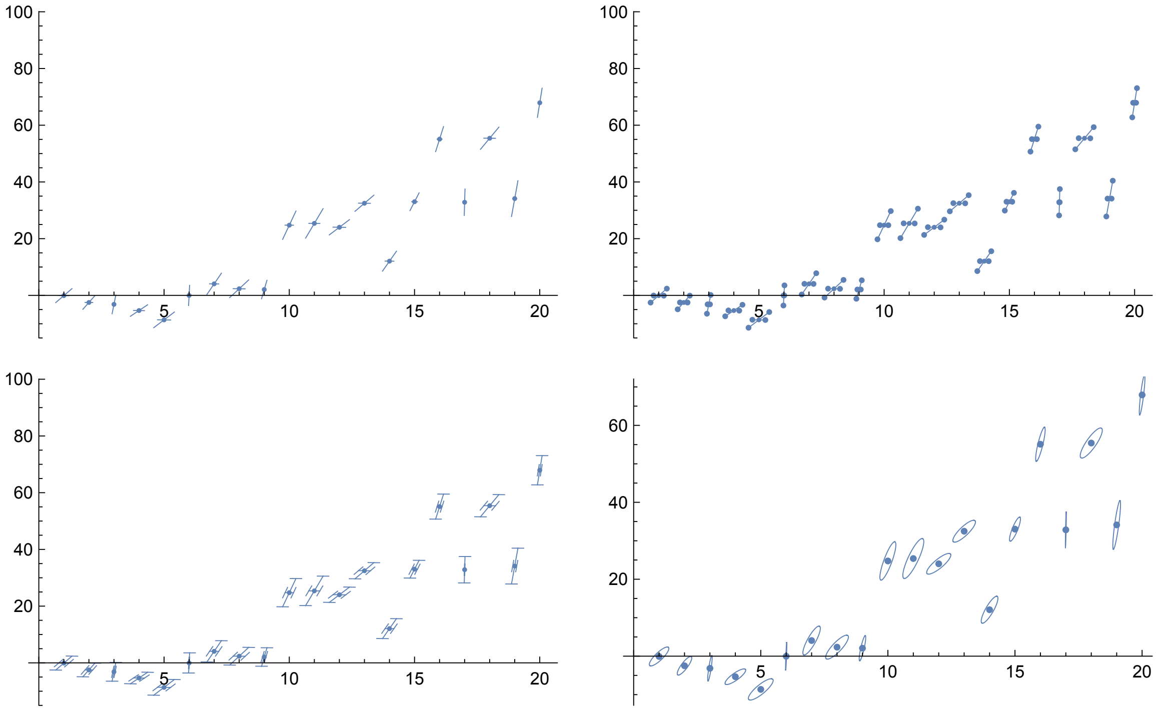
They are fully customizable, while Maple only lets you choose their color.

```
ListPlot[Take[data, 5], IntervalMarkersStyle → Directive[ LightGray, Dashed]]
```



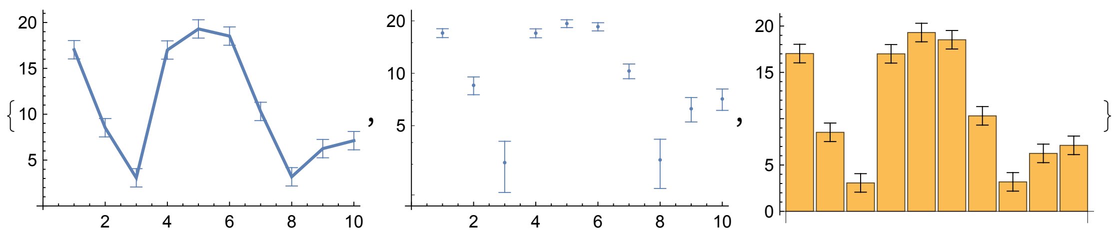
Mathematica supports correlated errors.

```
GraphicsGrid[Partition[Table[ListPlot[ Data , IntervalMarkers → i],
    {i, {"Bars", "Points", "Fences", "Ellipses"}}], 2]]
```



As well as **ListPlot**, uncertain data is supported in other Mathematica visualizations.

```
Through[{ListLinePlot, ListLogPlot, BarChart}[ Data ]]
```



Robustness of application

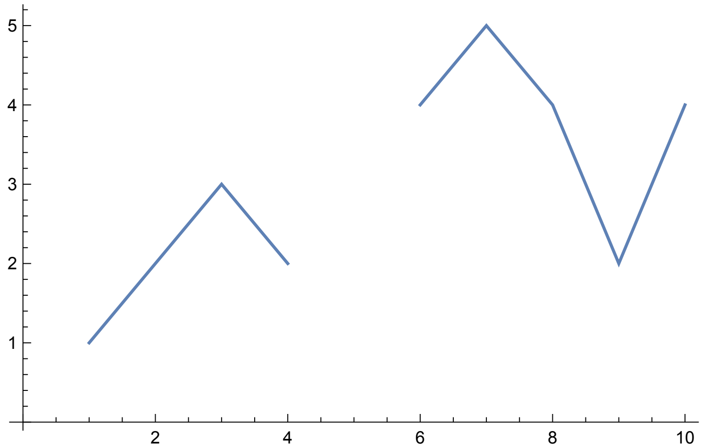
Mathematica visualizations are designed to handle real-world problems that are not always ideally posed. They robustly handle all kinds of potential problems in their application.

Missing data

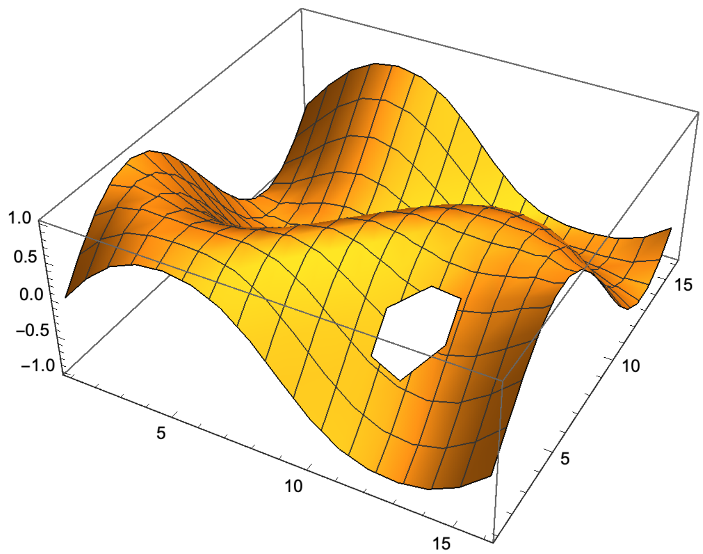
Mathematica

Data plots in Mathematica automatically skip over unplottable points such as symbols or NaN values in data plots.

```
ListLinePlot[{1, 2, 3, 2, Missing[], 4, 5, 4, 2, 4}]
```




```
data = Table[Sin[x + Sin[y]], {x, 0, 6, 0.4}, {y, 0, 6, 0.4}];
data[[4, 12]] = x;
ListPlot3D[data]
```



Maple

In each case, a single bad value in a dataset causes Maple to abandon the entire visualization. It is your responsibility to check and clean data before attempting to visualize it in Maple.

> $listplot([1, 2, 3, 2, missing(\,), 4, 5, 4, 2, 4])$

```
Error, (in plots:-pointplot) points cannot be converted to floating-point values
```

$data := [seq([seq(\sin(x + \sin(y)), x = 0 ..6, 0.4)\ ], y = 0 ..6, 0.4)] : data_{4,\,12} := \mathrm{x} :$

$with(plots) : matrixplot(data)$

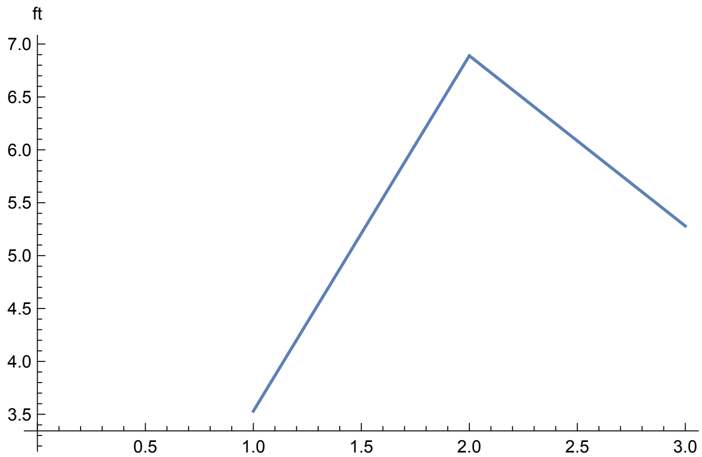
```
Error, (in plots/matrixplot) cannot convert first argument to a floating-point matrix
```

Units

Mathematica graphics can accept data with associated units, automatically converting to a common unit system.

Mathematica

```
ListLinePlot[{
  Quantity[3.53, "Feet"],
  Quantity[2.1, "Meters"],
  Quantity[0.001, "Miles"]}, AxesLabel → Automatic]
```



Maple

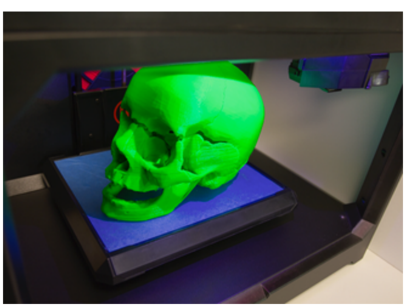
While 2D function plots in Maple can use units to label axes, data plots cannot handle data with associated units. Maple requires that you specify the units you wish to convert to, then strip out the units before plotting and then specify the units back into the axes labels.

```
> listplot([3.53 ⟦'ft'⟧, 4.1 ⟦'m'⟧])
Error, (in plots:-pointplot) points cannot be converted to floating-point values
```

3D printing support

Mathematica provides fully integrated capabilities to directly 3D print geometric models, using either an online printing service or your own printer. You can algorithmically generate geometric models or import and transform 3D models from files and immediately output physical 3D objects.

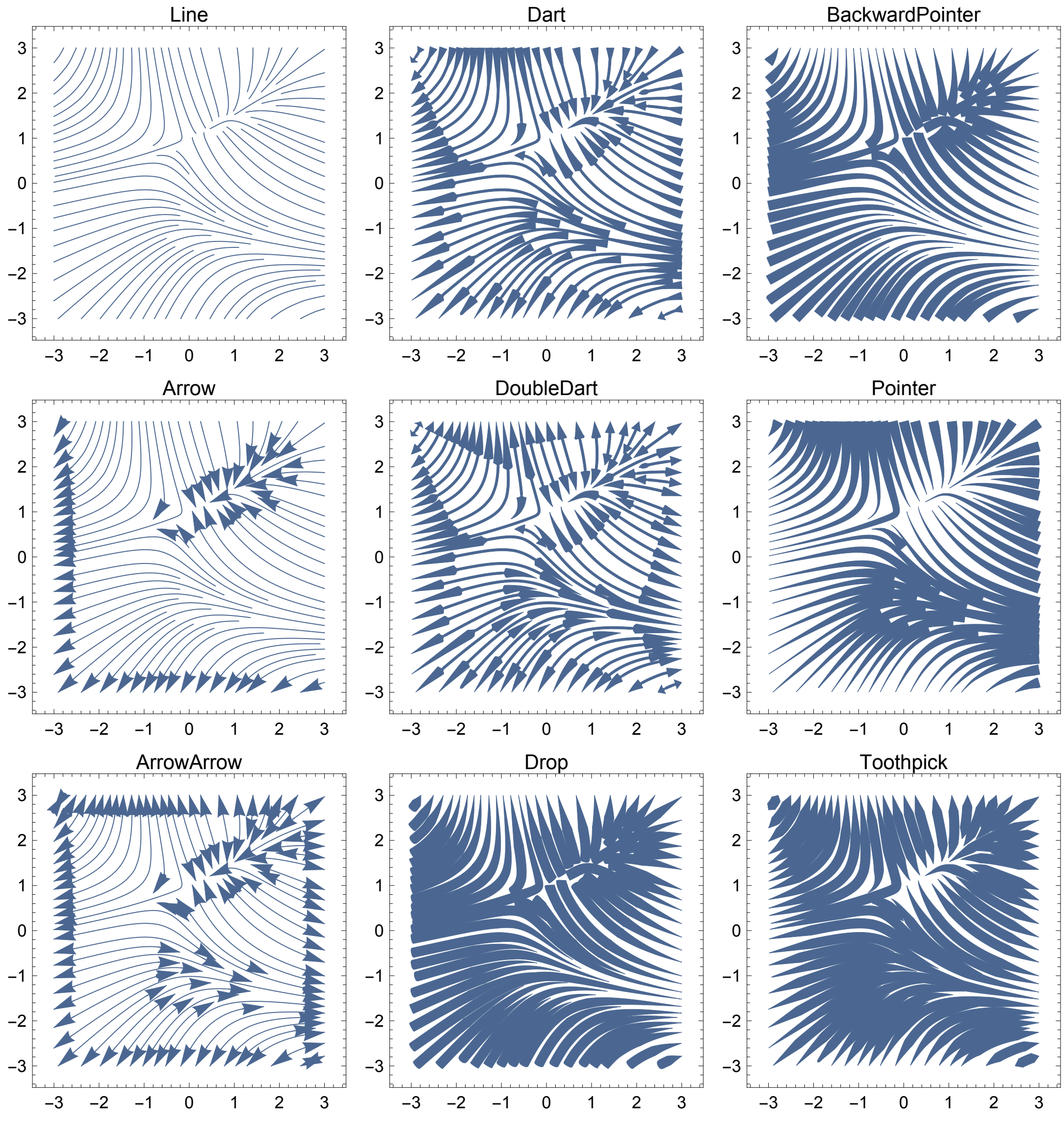
Printout3D[]



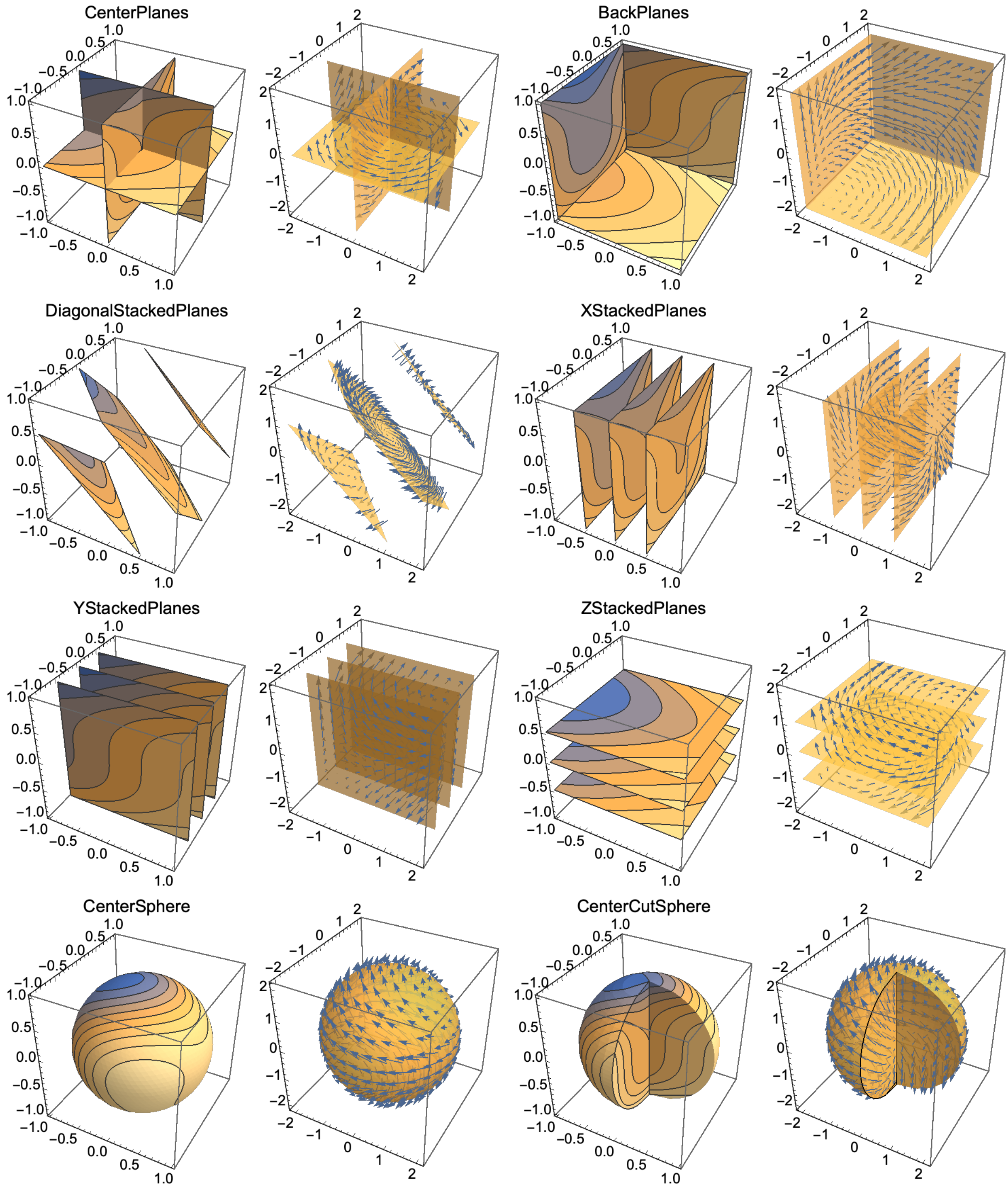
Breadth of capability

The range of built-in visualization types is much larger in Mathematica than Maple.
Maple has no direct way to produce ***any*** *of the following visualizations.*

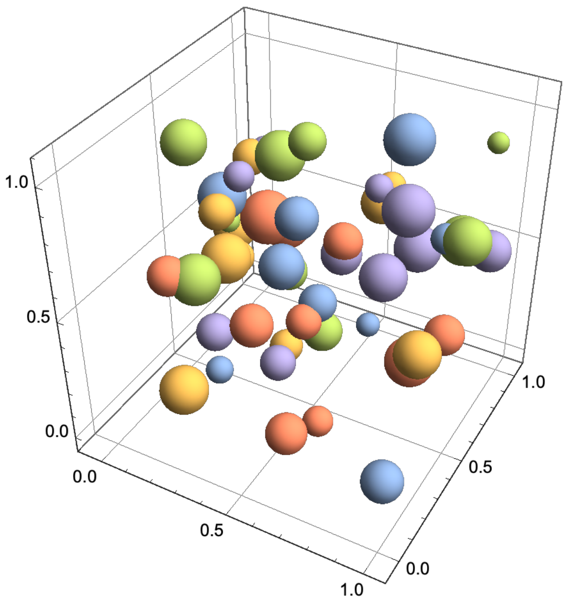
```
Multicolumn[
 Table[StreamPlot[{-1 - x^2 + y, 1 + x - y^2}, {x, -3, 3}, {y, -3, 3}, PlotLabel → s,
   StreamScale → {Full, All, 0.05}, StreamStyle → s], {s, {"Line", "Arrow", "ArrowArrow",
    "Dart", "DoubleDart", "Drop", "BackwardPointer", "Pointer", "Toothpick"}}], 3]
```



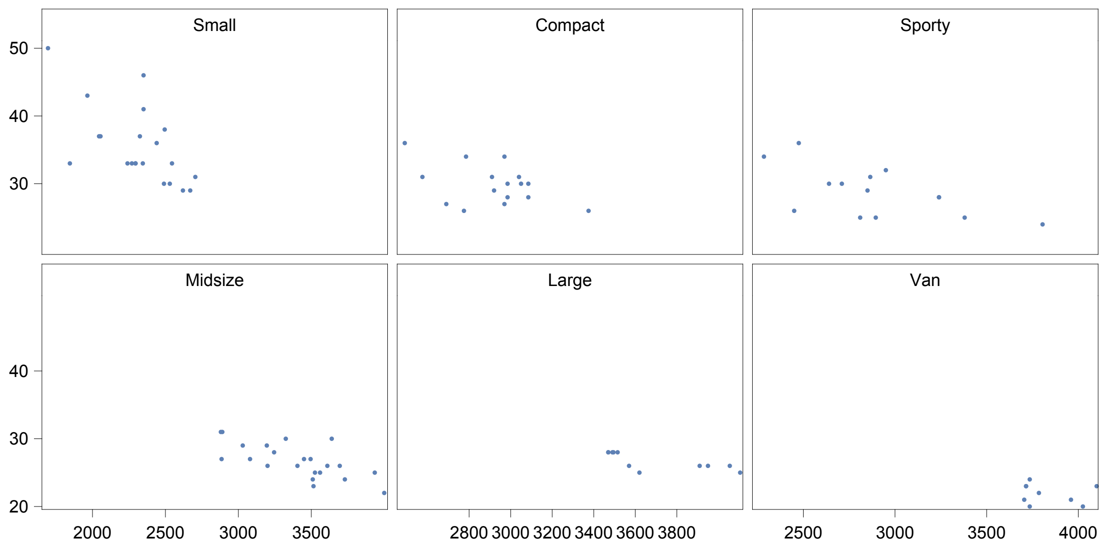
```
slices = {"CenterPlanes", "BackPlanes", "DiagonalStackedPlanes", "XStackedPlanes",
  "YStackedPlanes", "ZStackedPlanes", "CenterSphere", "CenterCutSphere"};
Multicolumn[Flatten[
  Table[{SliceContourPlot3D[Sin[x] + y^2 - z^3,
     sl, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}, PlotLabel → sl],
    SliceVectorPlot3D[{y, -x, z}, sl, {x, -2, 2}, {y, -2, 2}, {z, -2, 2}]},
   {sl, slices}]], 4, Appearance → "Horizontal"]
```



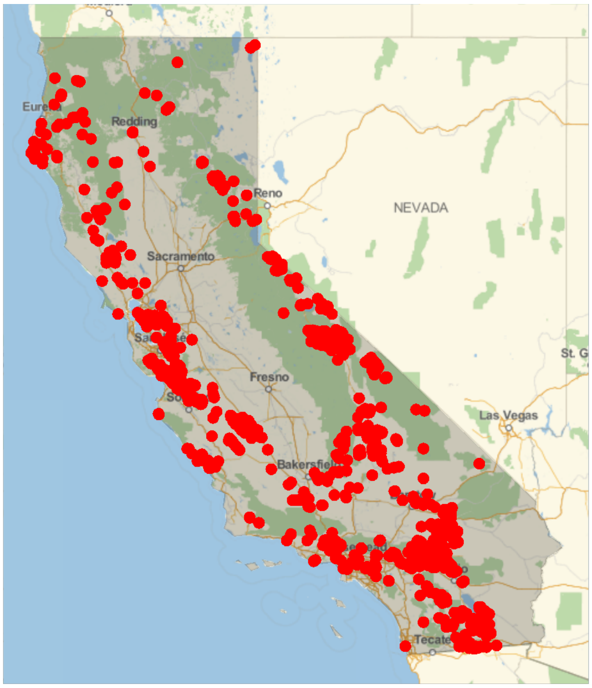
```
BubbleChart3D[RandomReal[1, {5, 10, 4}]]
```



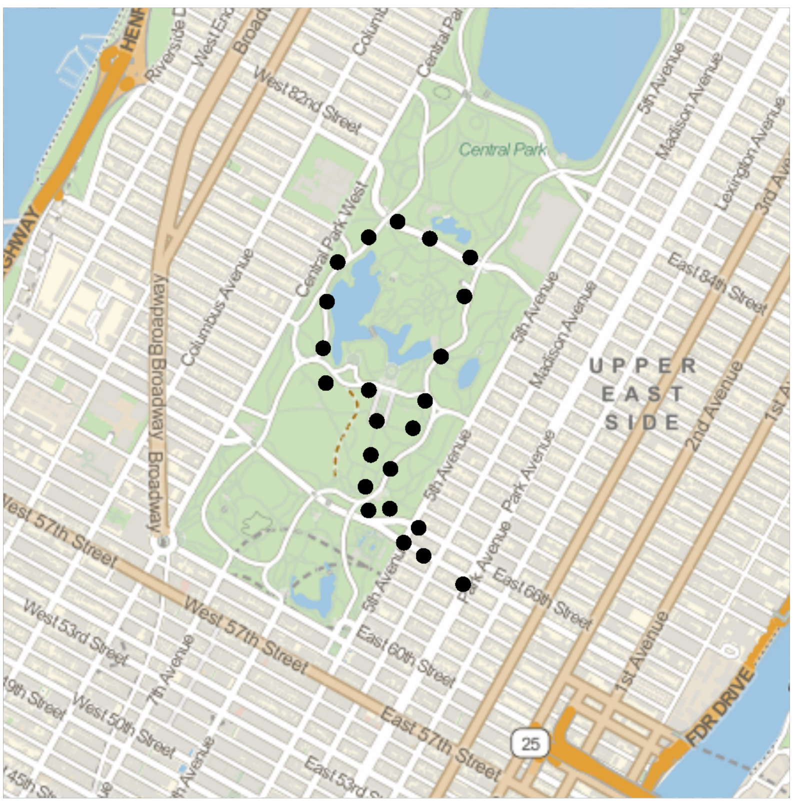

```
data = ⟨|"Small" → {{2705, 31}, {2270, 33}, {2670, 29}, {2295, 33}, {1845, 33},
    {2530, 30}, {1695, 50}, {2350, 46}, {2345, 33}, {2620, 29}, {2325, 37},
    {2440, 36}, {2295, 33}, {2545, 33}, {2350, 41}, {2495, 38}, {2045, 37},
    {2490, 30}, {1965, 43}, {2055, 37}, {2240, 33}}, "Midsize" → {⋯} +,
  "Compact" → {⋯} +, "Large" → {⋯} +, "Sporty" → {⋯} +, "Van" → {⋯} + |⟩;
ListPlot[data, PlotLayout → {"Column", 3}, PlotLabels → Keys[data]]
```



```
dat = EarthquakeData[ California, United States  ADMINISTRATIVE DIVISION ,
    4, {{1980, 1, 1}, {2014, 12, 31}}, "Position"]["Values"];
GeoGraphics[{Polygon[ California, United States  ADMINISTRATIVE DIVISION ],
  Red, PointSize[.02], Point[dat]}, ImageSize → 350]
```



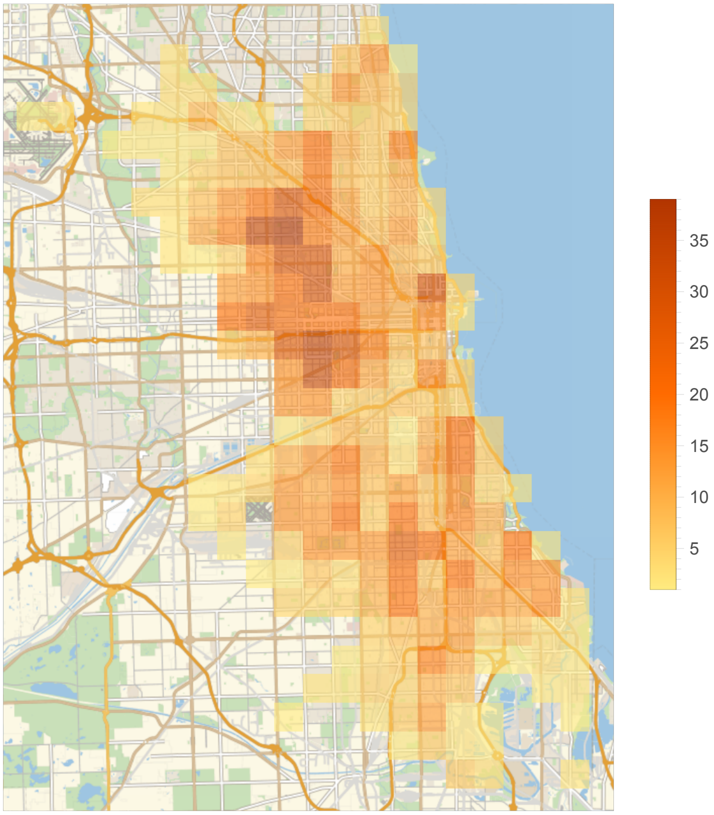

```
GeoGraphics[{PointSize[0.02], Point[ GPS Data ]}, GeoRange → 1 mi ]
```



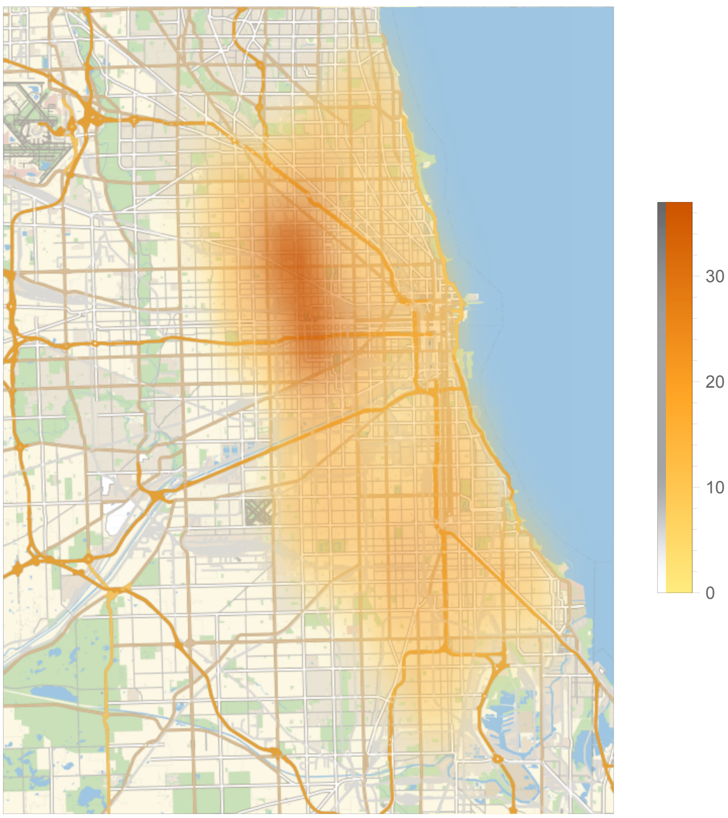
```
GeoBubbleChart[Data, GeoRange → South America COUNTRIES]
```

```
Multicolumn[GeoBubbleChart[Data, GeoBackground → #] & /@
  {"StreetMap", "Satellite", "ReliefMap"}, 3]
```

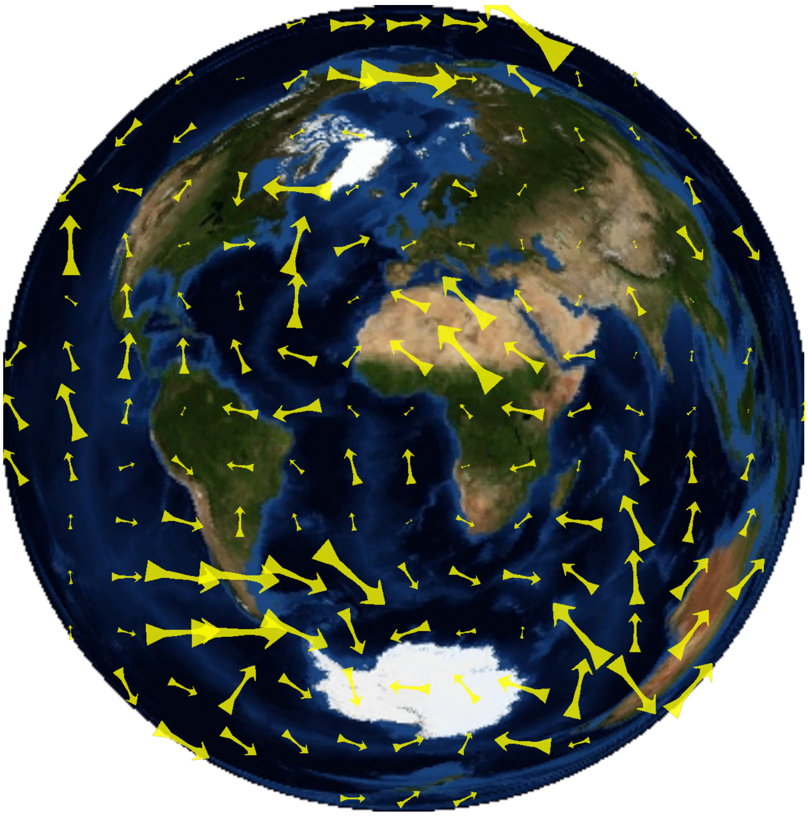
```
GeoHistogram[ Auto Theft Locations ,
 {"Rectangle", Quantity[1, "Miles"]}, PlotLegends → Automatic]
```



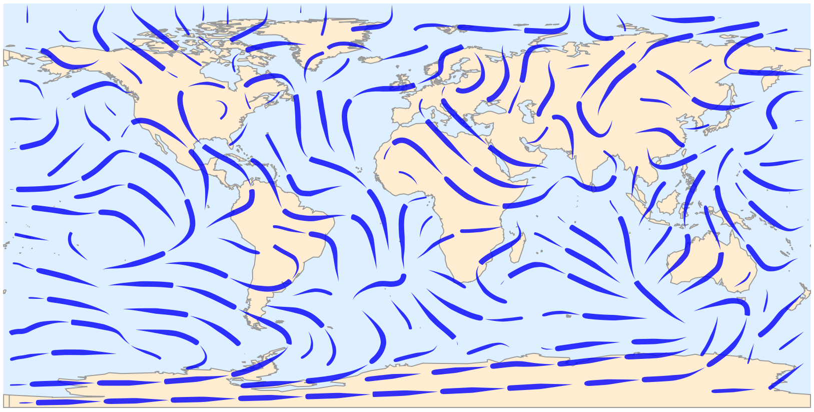

```
GeoSmoothHistogram[ Auto Theft Locations , PlotLegends → Automatic]
```



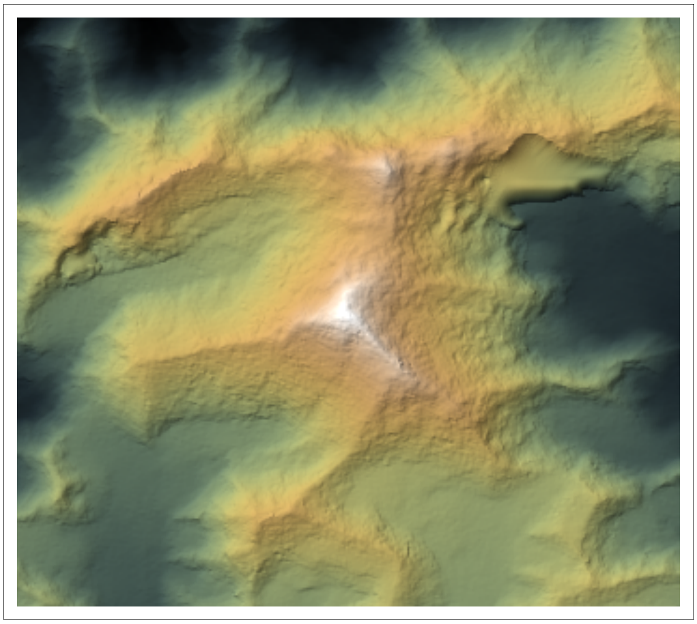
```
GeoVectorPlot[ Data , VectorMarkers → "Dart", VectorScale → Large, VectorStyle → Yellow,
 GeoProjection → "LambertAzimuthal", GeoBackground → "Satellite"]
```



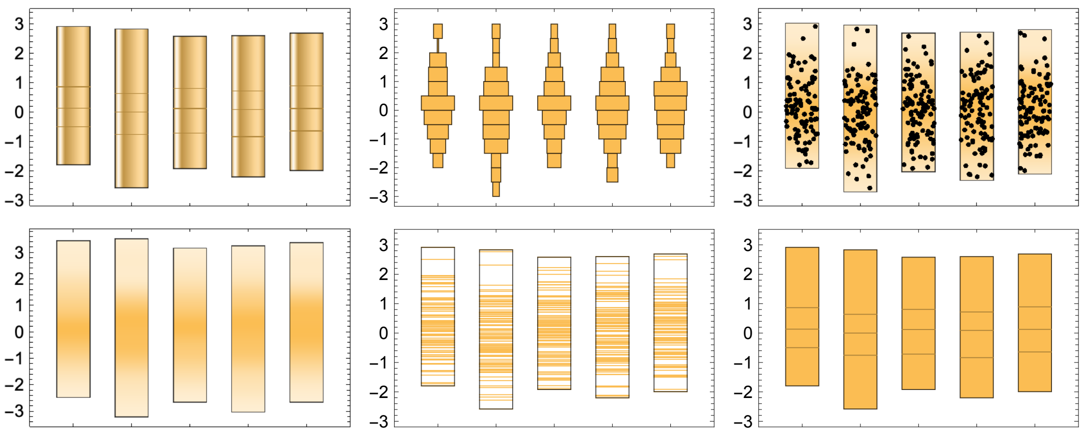
```
GeoStreamPlot[ Data , StreamMarkers → "Drop",
 StreamStyle → Blue, GeoBackground → "Coastlines"]
```



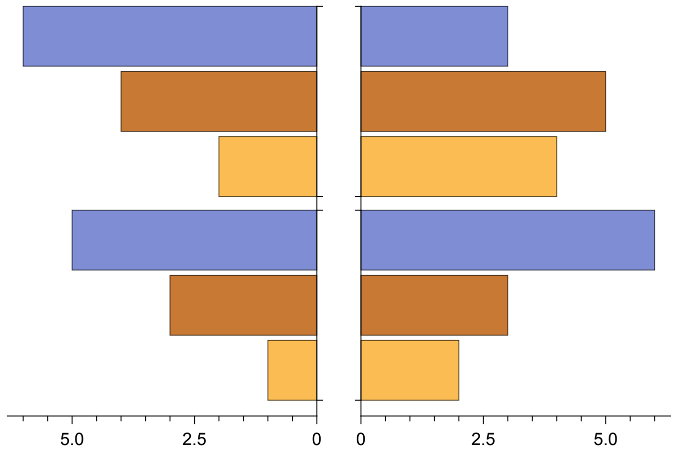
```
ReliefPlot[GeoElevationData[ Mount Everest MOUNTAIN , GeoRange → 6 km ],
 ColorFunction → "GreenBrownTerrain"]
```



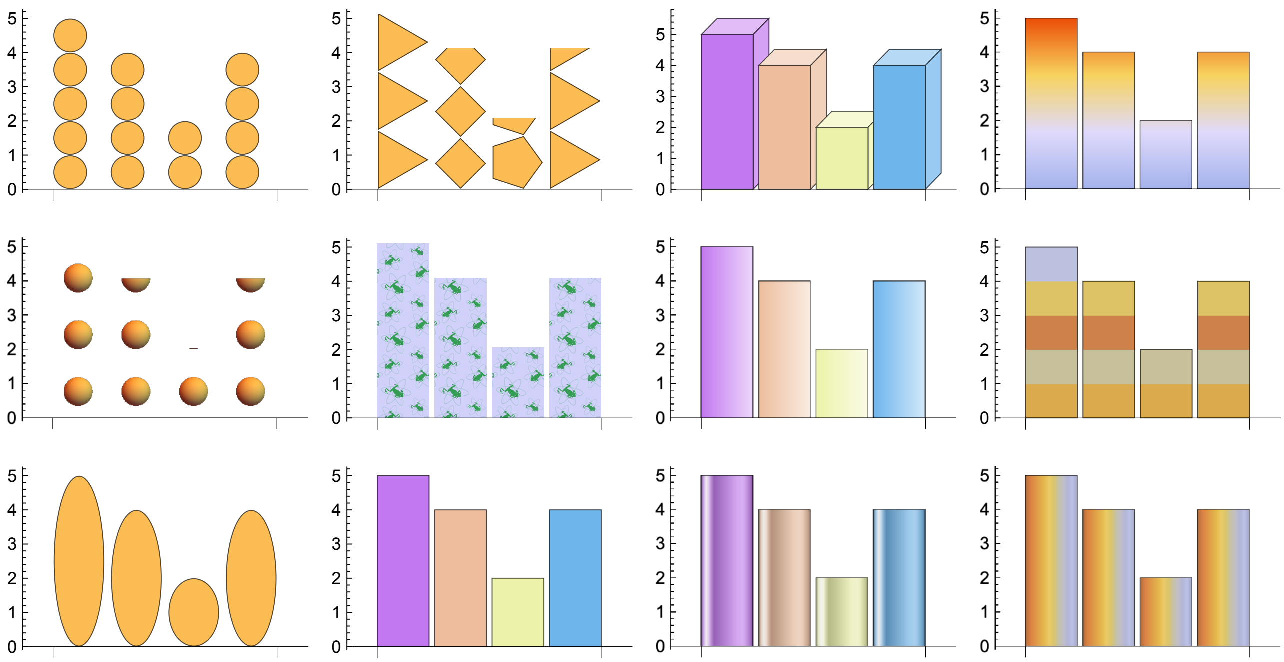
```
Multicolumn[
 Table[DistributionChart[ Data , ChartElementFunction → f], {f, {"GlassQuantile",
    "Density", "HistogramDensity", "LineDensity", "PointDensity", "Quantile"}}], 3]
```

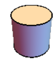


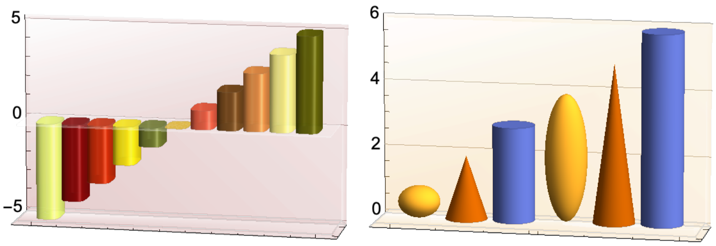
```
PairedBarChart[{{1, 3, 5}, {2, 4, 6}}, {{2, 3, 6}, {4, 5, 3}}]
```

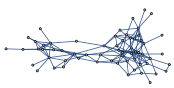


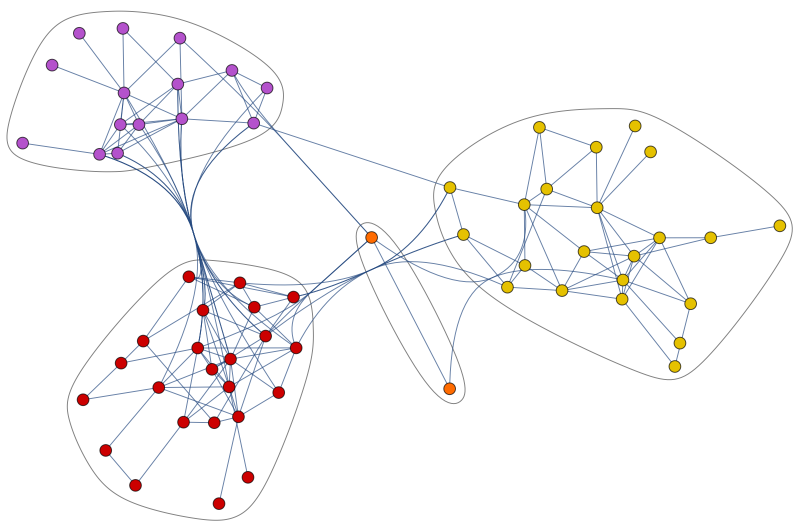

```
Multicolumn[Join[Table[BarChart[{5, 4, 2, 4}, ChartElements → i],
  {i, {{ , {1, 1}}, {Graphics3D[Sphere[], Boxed → False]}, { , All},
    { , , }, ExampleData[{"ColorTexture", "FrogsPattern"}]}}],
  Table[BarChart[{5, 4, 2, 4}, ChartElementFunction → f, ChartStyle → "Pastel"],
  {f, {"Rectangle", "ObliqueRectangle", "FadingRectangle", "GlassRectangle",
    "GradientScaleRectangle", "SegmentScaleRectangle", "GradientRectangle"}}]], 4]
```



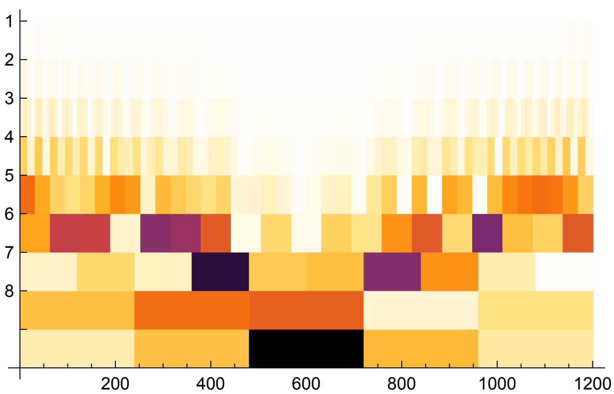
Multicolumn[
 {BarChart3D[Range[-5, 5], ChartStyle → 53, ChartElementFunction → "ProfileCube",
   ChartBaseStyle → Directive[EdgeForm[Gray], Opacity[0.8], Specularity[White, 30]]],
  BarChart3D[{{1, 2, 3}, {4, 5, 6}}, ChartElements → {  ,  ,  }]}, 2]



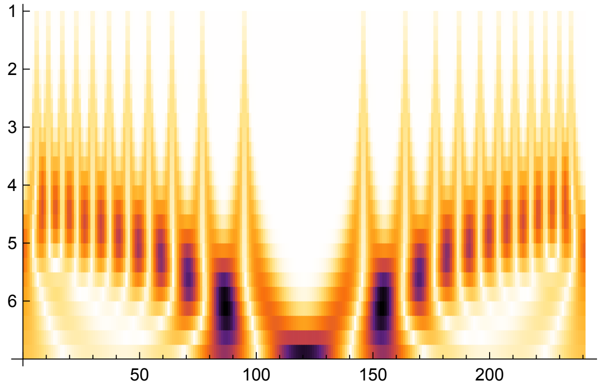
CommunityGraphPlot[]



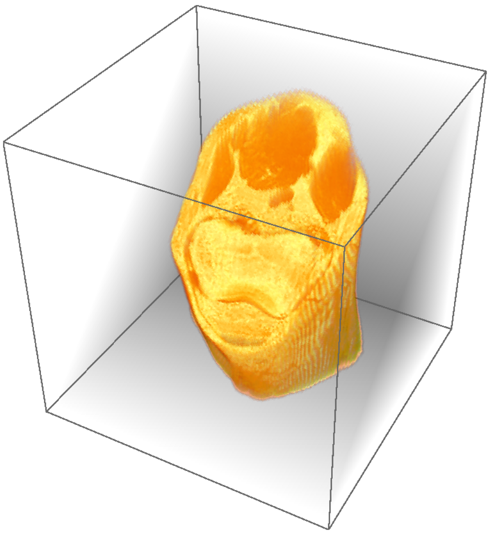
WaveletScalogram[
 DiscreteWaveletTransform[Table[Sin[x^2], {x, -6, 6, 0.01}], Automatic, 8]]



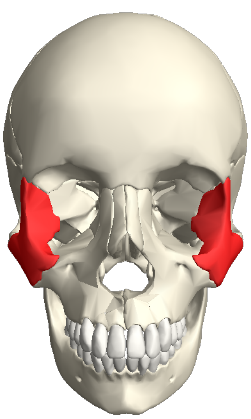
```
WaveletScalogram[ContinuousWaveletTransform[Table[Sign[Cos[x^2]], {x, -6, 6, 0.05}]]]
```



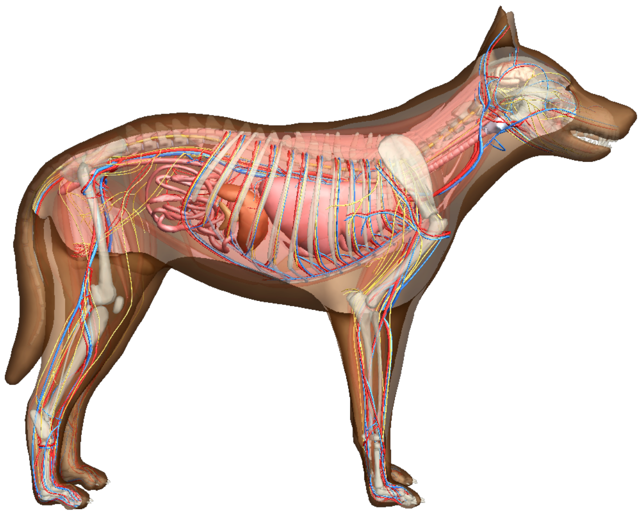

```
Show[ExampleData[{"TestImage3D", "MRknee"}], ClipPlanes → {{0, 1, -1, 0}}]
```



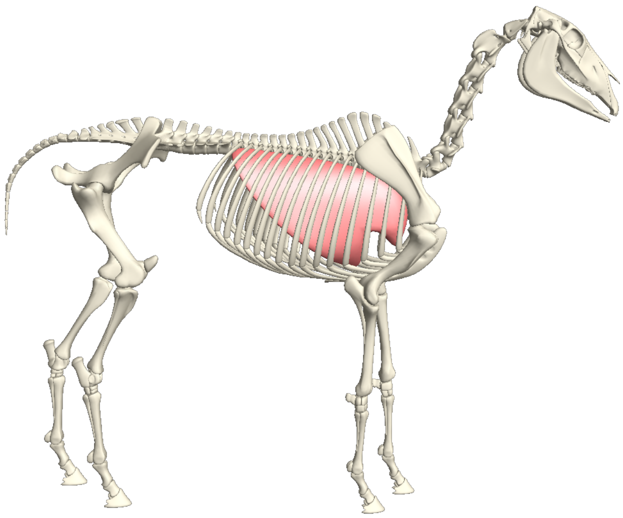
```
AnatomyPlot3D[{neurocranium ANATOMICAL STRUCTURE, sphenoid bone ANATOMICAL STRUCTURE,
  nasal bone ANATOMICAL STRUCTURE, maxilla ANATOMICAL STRUCTURE, maxillary dentition ANATOMICAL STRUCTURE,
  mandible ANATOMICAL STRUCTURE, mandibular dentition ANATOMICAL STRUCTURE, Red,
  zygomatic bone ANATOMICAL STRUCTURE}, PlotRange → skull ANATOMICAL STRUCTURE]
```



```
AnatomyPlot3D[{ alimentary system (dog) ANIMAL ANATOMICAL STRUCTURE ,
  respiratory system (dog) ANIMAL ANATOMICAL STRUCTURE ,
  cardiovascular system (dog) ANIMAL ANATOMICAL STRUCTURE ,
  nervous system (dog) ANIMAL ANATOMICAL STRUCTURE , Opacity[.5],
  skeleton (dog) ANIMAL ANATOMICAL STRUCTURE , ClipPlanes →
   Dynamic[{InfinitePlane[{{-78, -200, 0}, {-78, 300, 0}, {-78 + 50, 0, 500}}]}],
  Opacity[.1], muscular system (dog) ANIMAL ANATOMICAL STRUCTURE , Opacity[.7],
  skin (dog) ANIMAL ANATOMICAL STRUCTURE }, ViewPoint → Left]
```

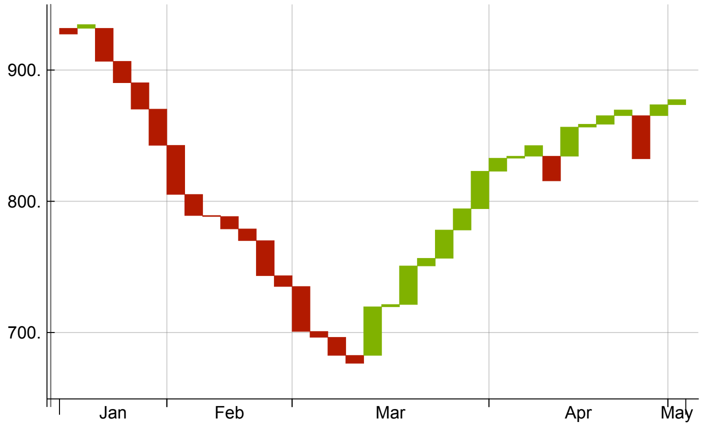


```
AnatomyPlot3D[{ set of bones (horse) ANIMAL ANATOMICAL STRUCTURE ,
  lung (horse) ANIMAL ANATOMICAL STRUCTURE }, ViewPoint → Left]
```

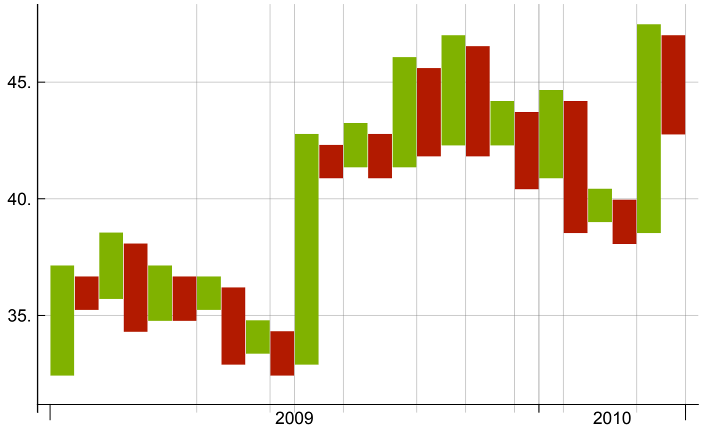


```
TradingChart["SP500", {"Volume", "SimpleMovingAverage", "BollingerBands"}]
```

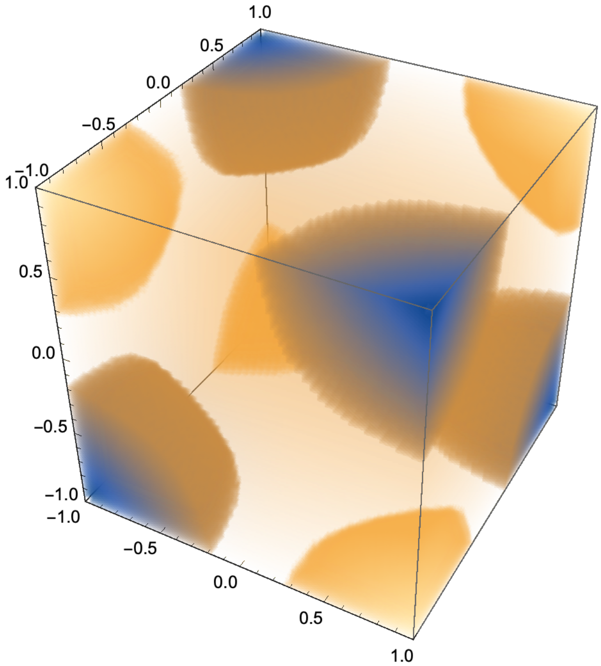

```
LineBreakChart[{"^GSPC", {{2009, 1, 1}, {2009, 4, 31}}}]
```



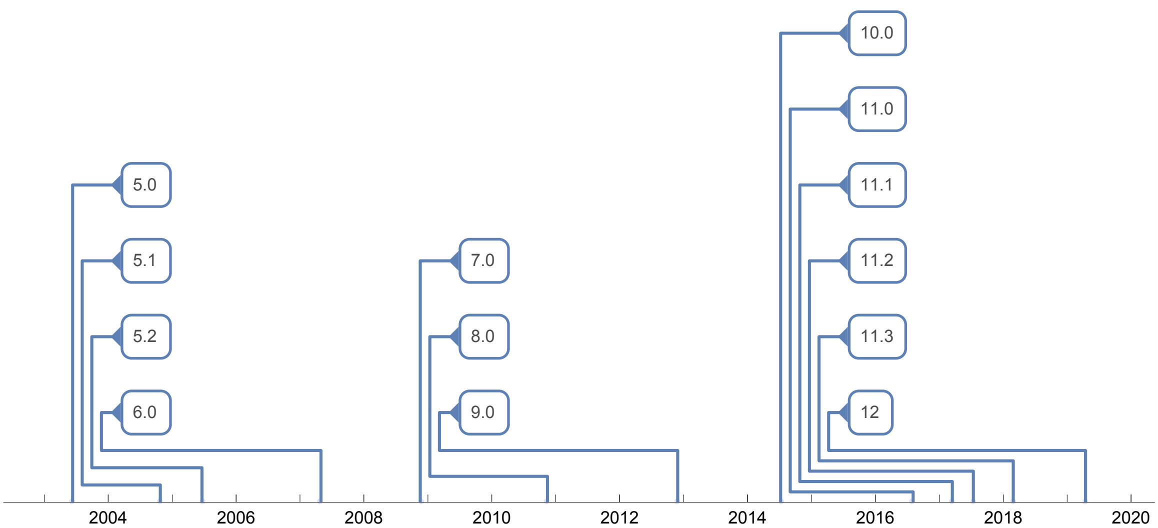

```
PointFigureChart[FinancialData["JPM", "Close", {{2009, 5, 1}, {2010, 4, 30}}]]
```

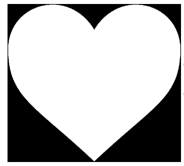


```
DensityPlot3D[x y z, {x, -1, 1}, {y, -1, 1}, {z, -1, 1}]
```

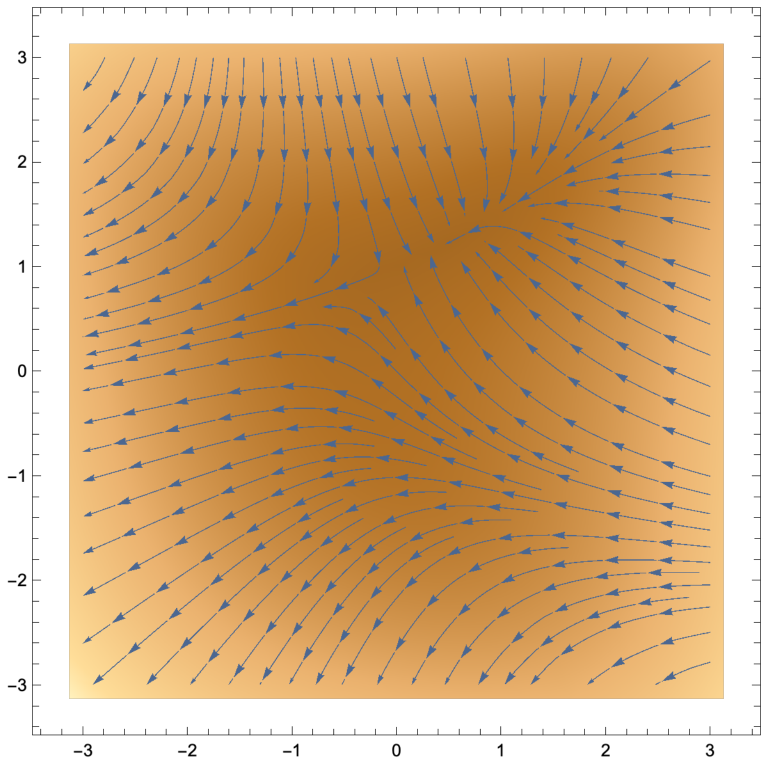



```
TimelinePlot[⟨|"5.0" → Thu 12 Jun 2003, "5.1" → Mon 25 Oct 2004,
  "5.2" → Mon 20 Jun 2005, "6.0" → Tue 1 May 2007, "7.0" → Tue 18 Nov 2008,
  "8.0" → Mon 15 Nov 2010, "9.0" → Wed 28 Nov 2012, "10.0" → Wed 9 Jul 2014,
  "11.0" → Thu 4 Aug 2016, "11.1" → Day: Thu 16 Mar 2017, "11.2" → Day: Fri 14 Jul 2017,
  "11.3" → Day: Tue 27 Feb 2018, "12" → Day: Tue 16 Apr 2019|⟩]
```

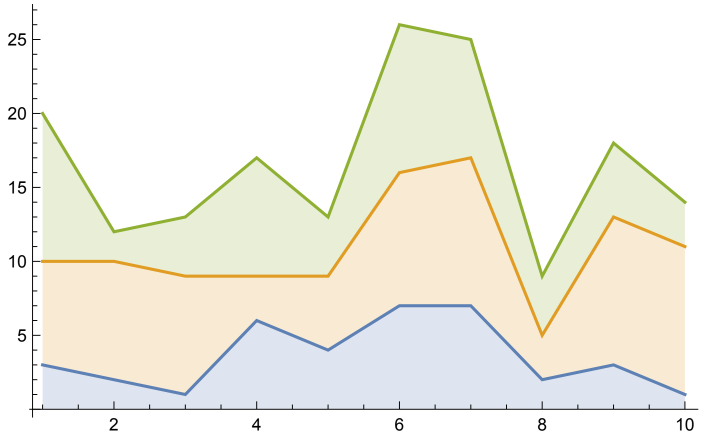


`WordCloud[EntityValue[CountryData[], {"Name", "Population"}],`  `]`

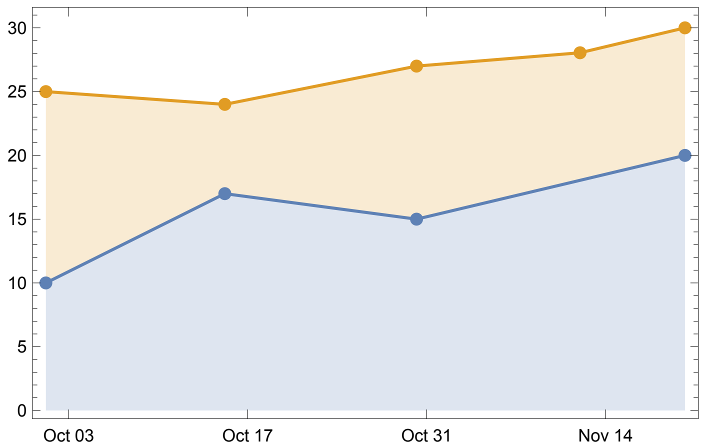
`StreamDensityPlot[{-1 - x^2 + y, 1 + x - y^2}, {x, -3, 3}, {y, -3, 3}]`



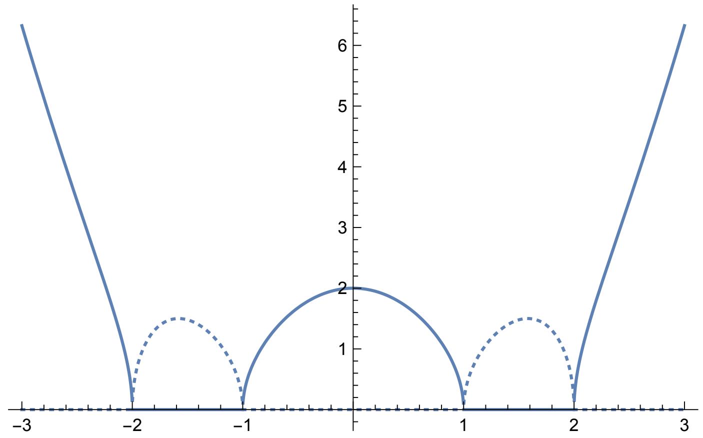
```
StackedListPlot[{{3, 2, 1, 6, 4, 7, 7, 2, 3, 1},
  {7, 8, 8, 3, 5, 9, 10, 3, 10, 10}, {10, 2, 4, 8, 4, 10, 8, 4, 5, 3}}]
```



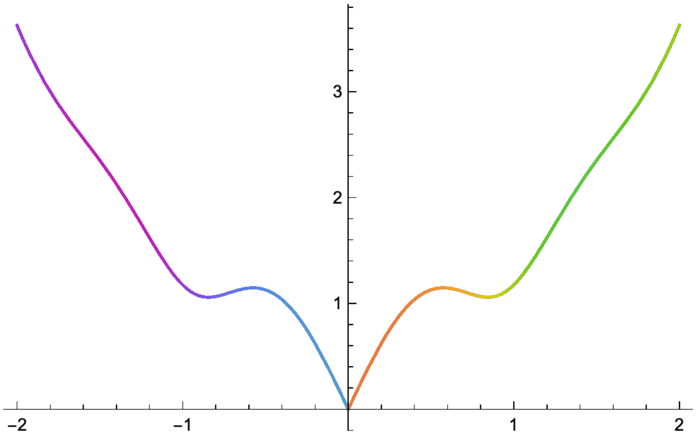

```
data1 = {{ Day: Sat 1 Oct 2016 , 10}, { Day: Sat 15 Oct 2016 , 17},
   { Day: Sun 30 Oct 2016 , 15}, { Day: Sun 20 Nov 2016 , 20}};
data2 = {{ Day: Sat 1 Oct 2016 , 15}, { Day: Sat 15 Oct 2016 , 7}, { Day: Sun 30 Oct 2016 , 12},
   { Day: Sat 12 Nov 2016 , 10}, { Day: Sun 20 Nov 2016 , 10}};
StackedDateListPlot[{data1, data2}]
```



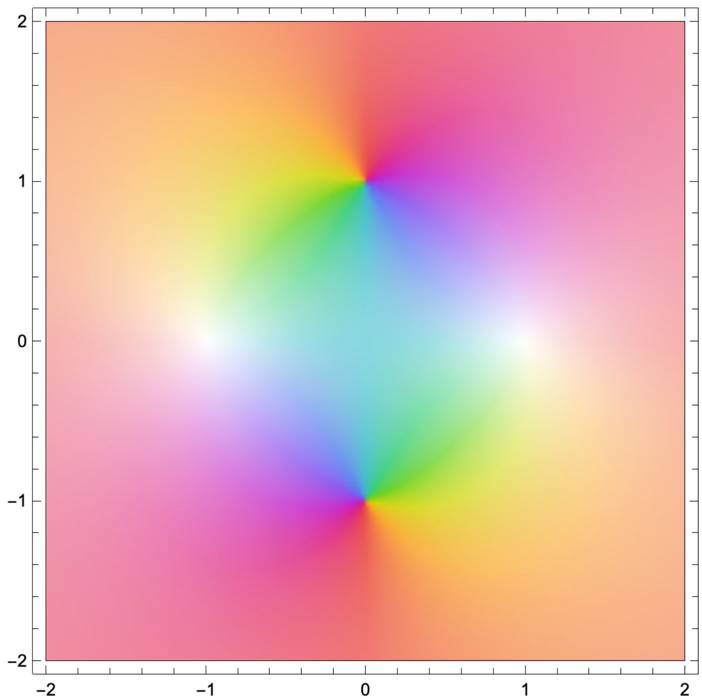
ReImPlot$\left[\sqrt{(x^2 - 1)\ (x^2 - 4)}\right.$, {x, -3, 3}]



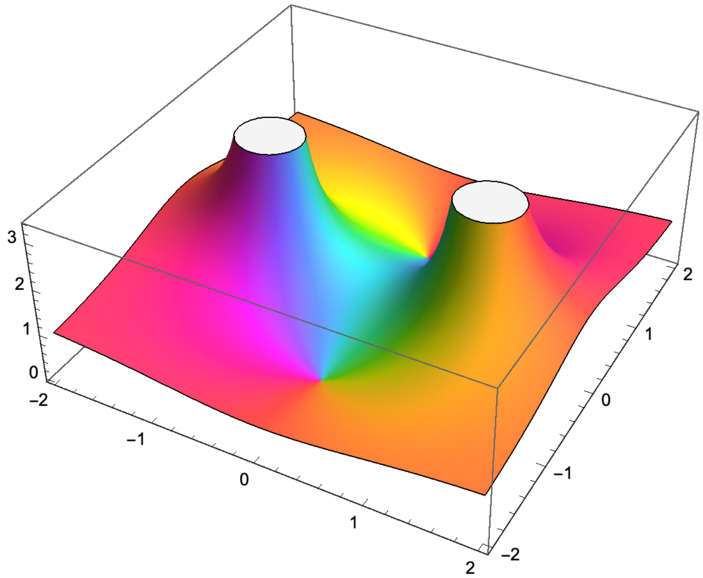
```
AbsArgPlot[Sin[I x] + Sin[Pi x], {x, -2, 2}]
```



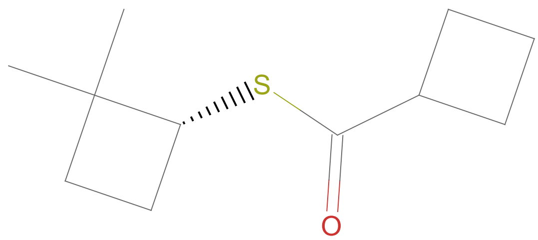
$$\texttt{ComplexPlot}\left[\frac{z^2+1}{z^2-1}, \{z, -2-2\,I, 2+2\,I\}\right]$$



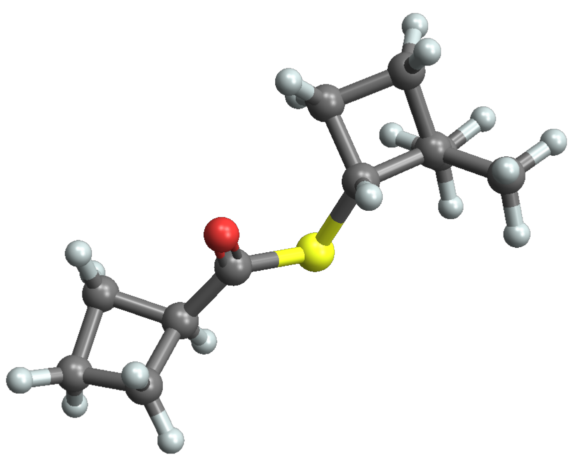
```
ComplexPlot3D[(z^2 + 1)/(z^2 - 1), {z, -2 - 2 I, 2 + 2 I}]
```



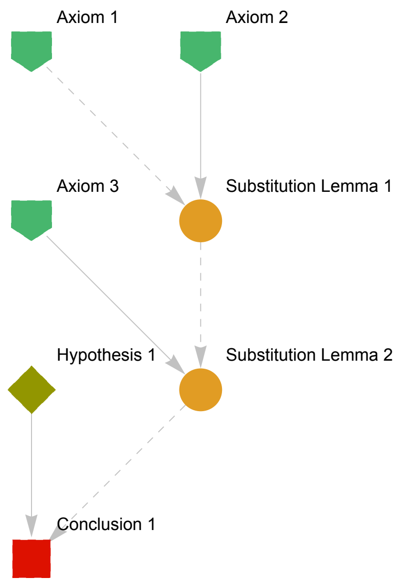

```
MoleculePlot[Molecule["O=C(C1CCC1)S[C@@H]1CCC1(C)C"]]
```



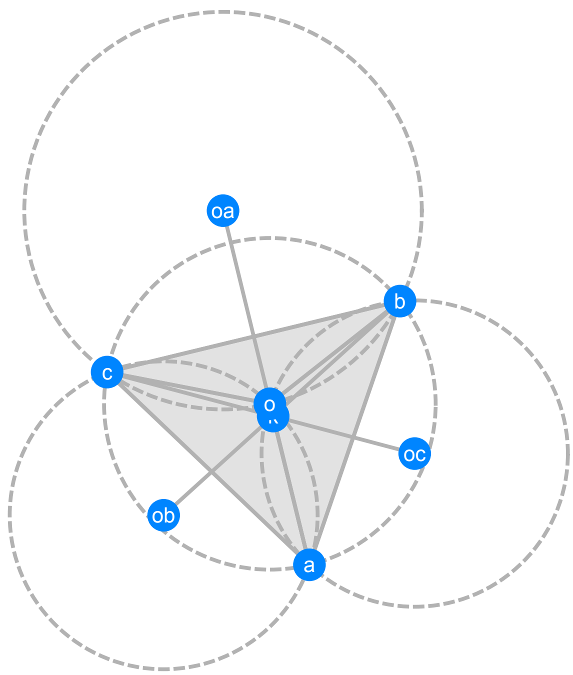

```
MoleculePlot3D[Molecule["O=C(C1CCC1)S[C@@H]1CCC1(C)C"]]
```



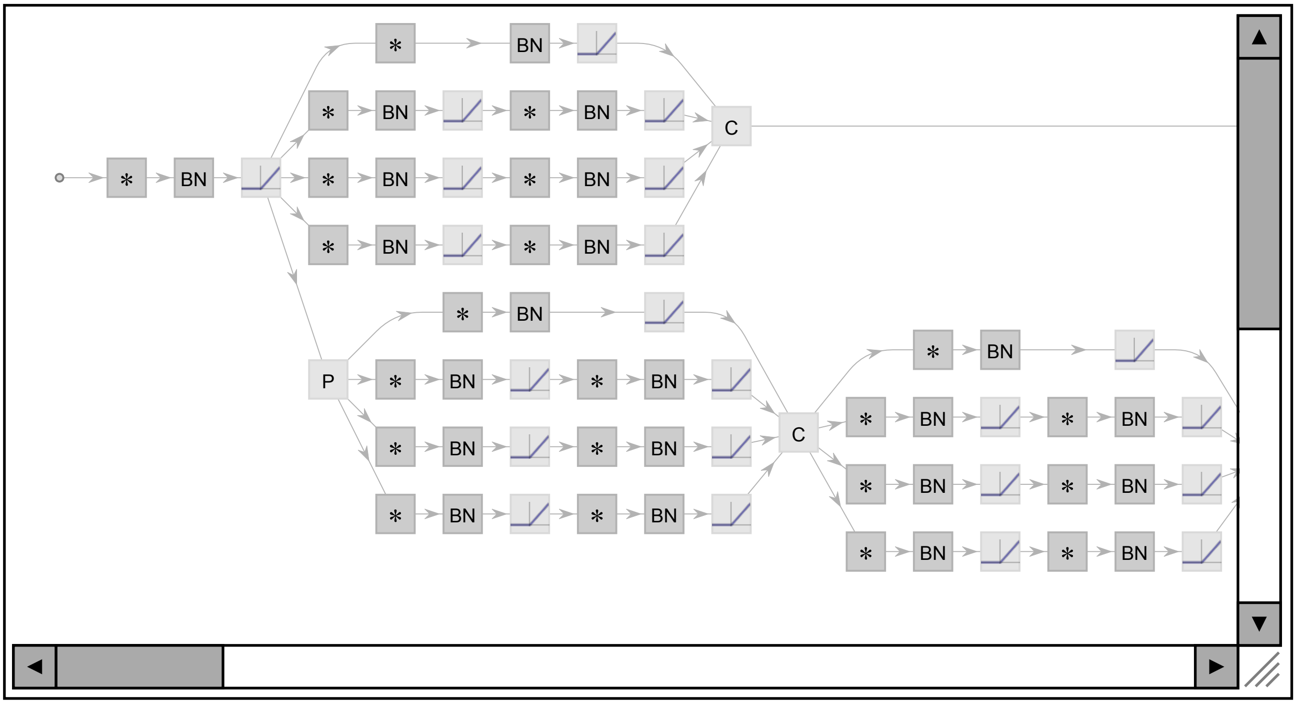
```
FindEquationalProof[a == d, {a == b, b == c, c == d}]["ProofGraph"]
```



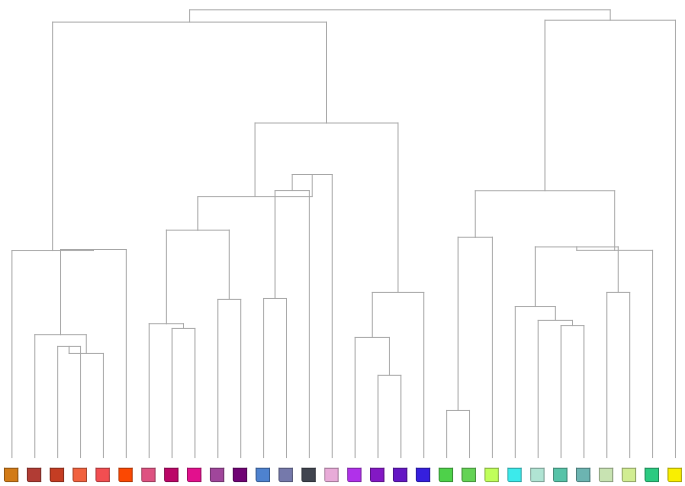

```
RandomInstance[GeometricScene[{a, b, c, o, oa, ob, oc, k},
  {o == TriangleCenter[{a, b, c}, "Circumcenter"],
   oa == TriangleCenter[{o, b, c}, "Circumcenter"],
   ob == TriangleCenter[{a, o, c}, "Circumcenter"],
   oc == TriangleCenter[{a, b, o}, "Circumcenter"], Line[{a, k, oa}],
   Line[{b, k, ob}], Line[{c, oc}]}], RandomSeeding → 5]
```

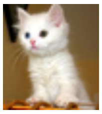


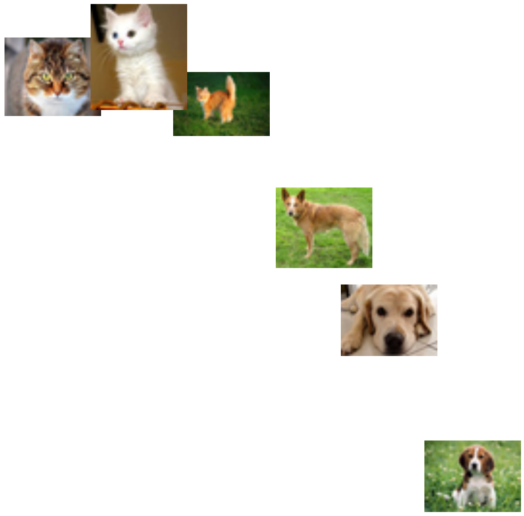
```
Framed@Pane[NetInformation[
   NetModel["Single-Image Depth Perception Net Trained on Depth in the Wild Data"],
   "FullSummaryGraphic"], {660, 350}, Scrollbars → True]
```

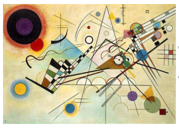


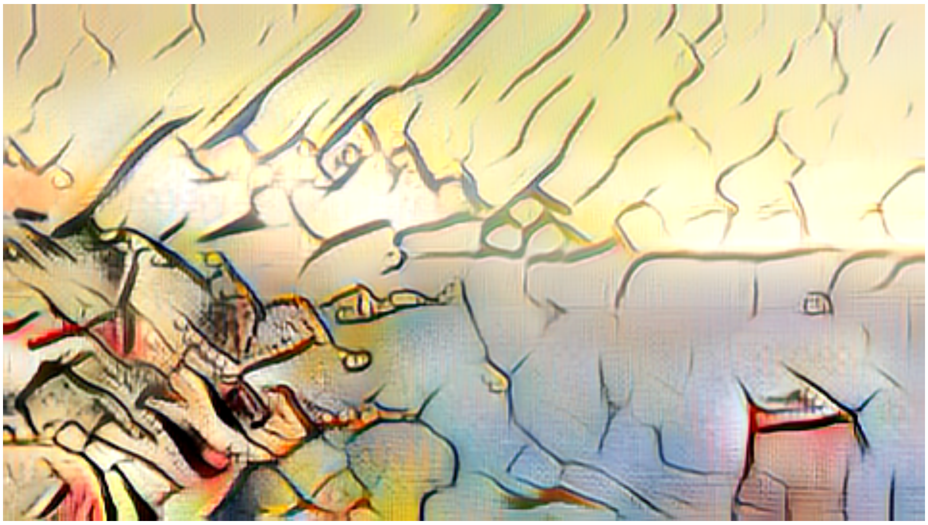

```
Dendrogram[RandomColor[30], ClusterDissimilarityFunction → "Centroid"]
```



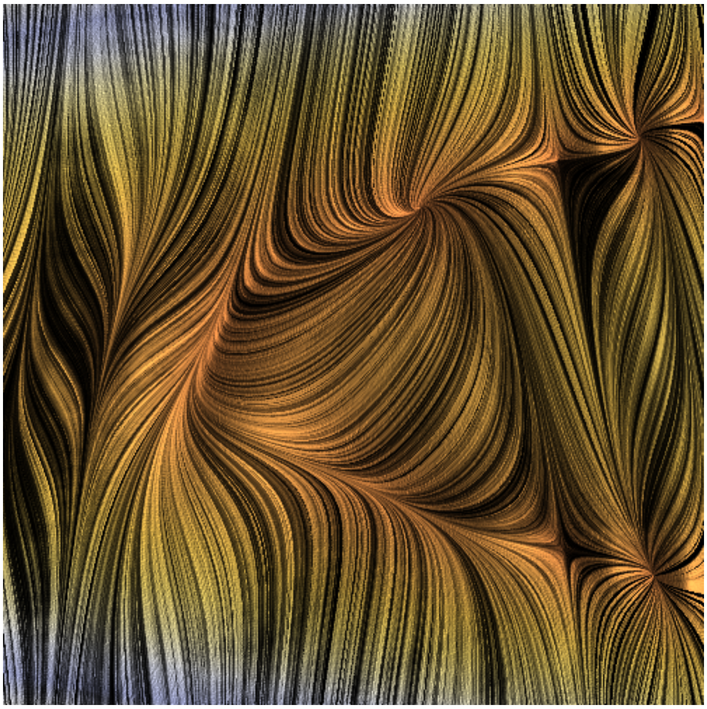
FeatureSpacePlot[{, , , , , **}, LabelingSize → 60]**



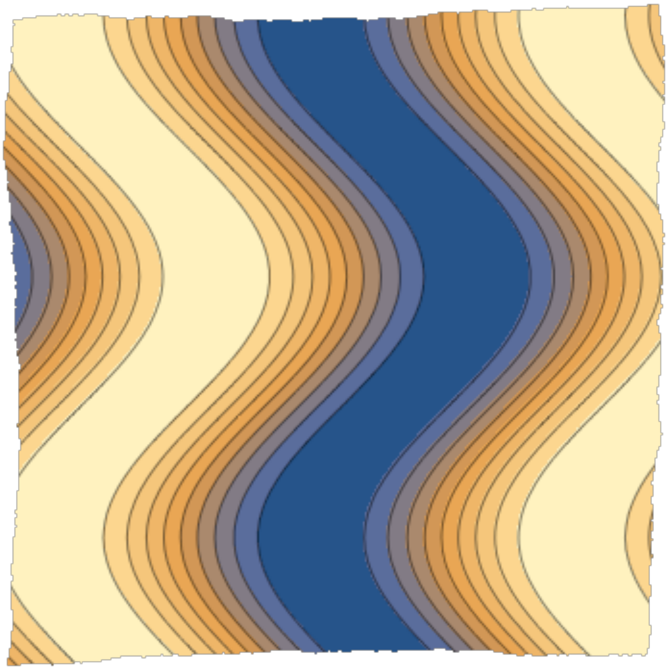
ImageRestyle[, **]**



```
LineIntegralConvolutionPlot[{{Cos[x² + y], 1 + x - y²}, {"noise", 500, 500}},
 {x, -3, 3}, {y, -3, 3}, ColorFunction → "BeachColors",
 LightingAngle → 0, LineIntegralConvolutionScale → 3, Frame → False]
```



```
ImageEffect[ContourPlot[Sin[x + Sin[y]], {x, 0, 8}, {y, 0, 8}, Frame → None], "TornFrame"]
```



Notes

- Images were generated using Mathematica 12 and Maple 2019
- Images have been copied using a screen capture tool to preserve pixel-level screen rendering. Printing this document will not represent the resolution that printing from the original application would achieve.
- Except where stated, all comparisons use default options. Both systems allow manual control over plot details, and in some cases, with sufficient work, a user may overcome some of the Maple deficiencies described in this comparison.
- Some plots have been manually rotated so they can be compared from similar viewpoints.