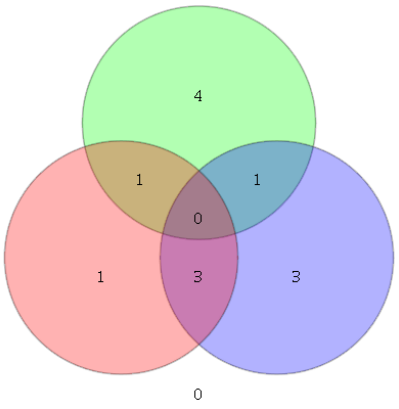
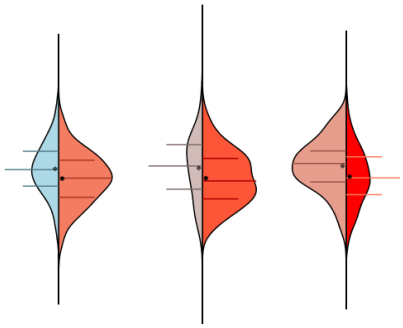
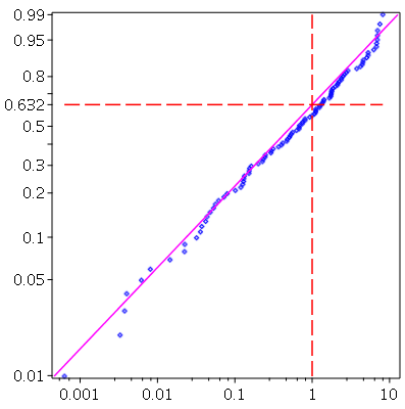
Statistics and Data Analysis

Visualizations

There are three new visualizations in Statistics:

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

Venn Diagrams

VennDiagrams are a method of data visualization showing the relationships between multiple sets of data by depicting these sets as regions inside closed curves.

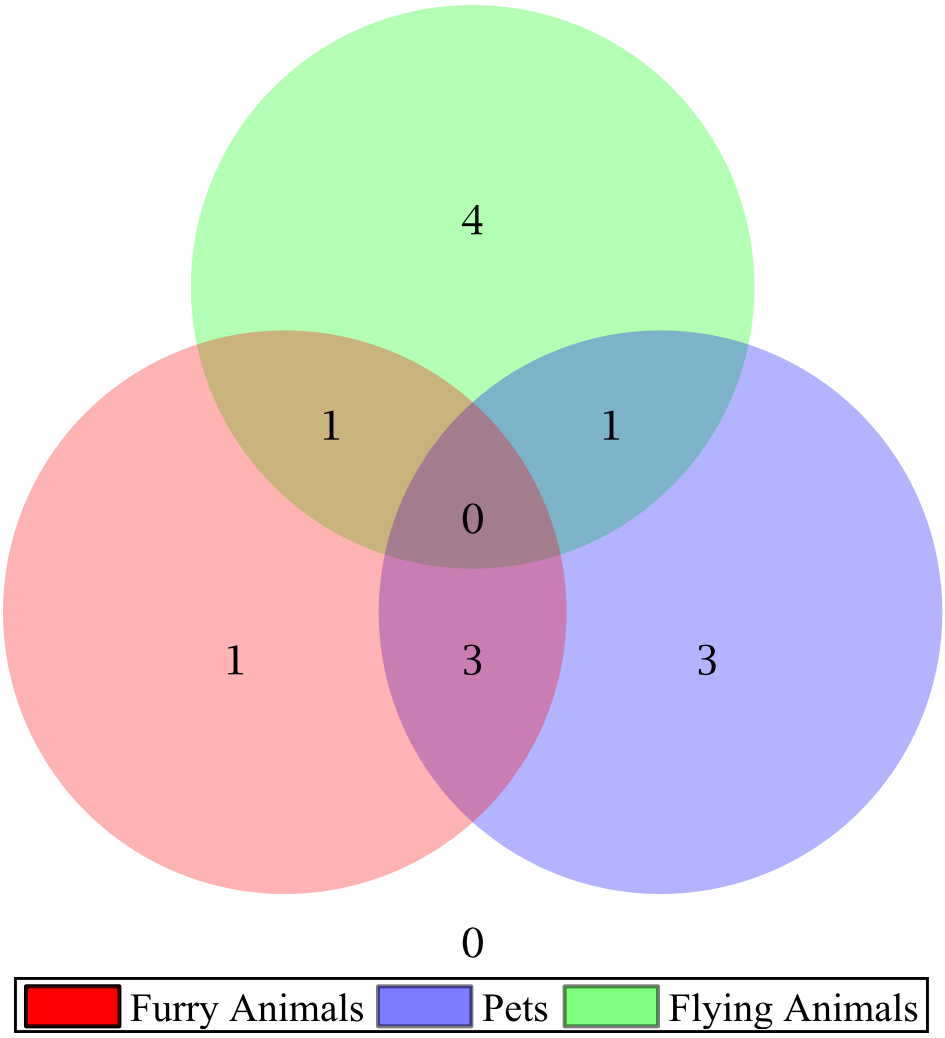
```
> with(Statistics):
```

```
> FurryAnimals := {"Bat", "Cat", "Caterpillar", "Dog",
  "Gerbil"}:
```

```
> Pets := {"Cat", "Dog", "Gerbil", "Goldfish", "Lizard",
  "Parrot", "Snake"}:
```

```
> FlyingAnimals := {"Bat", "Butterfly", "Eagle", "Parrot",
  "Vulture", "Wasp"}:
```

```
> VennDiagram(FurryAnimals, Pets, FlyingAnimals, legend=["Furry
  Animals", "Pets", "Flying Animals"]);
```



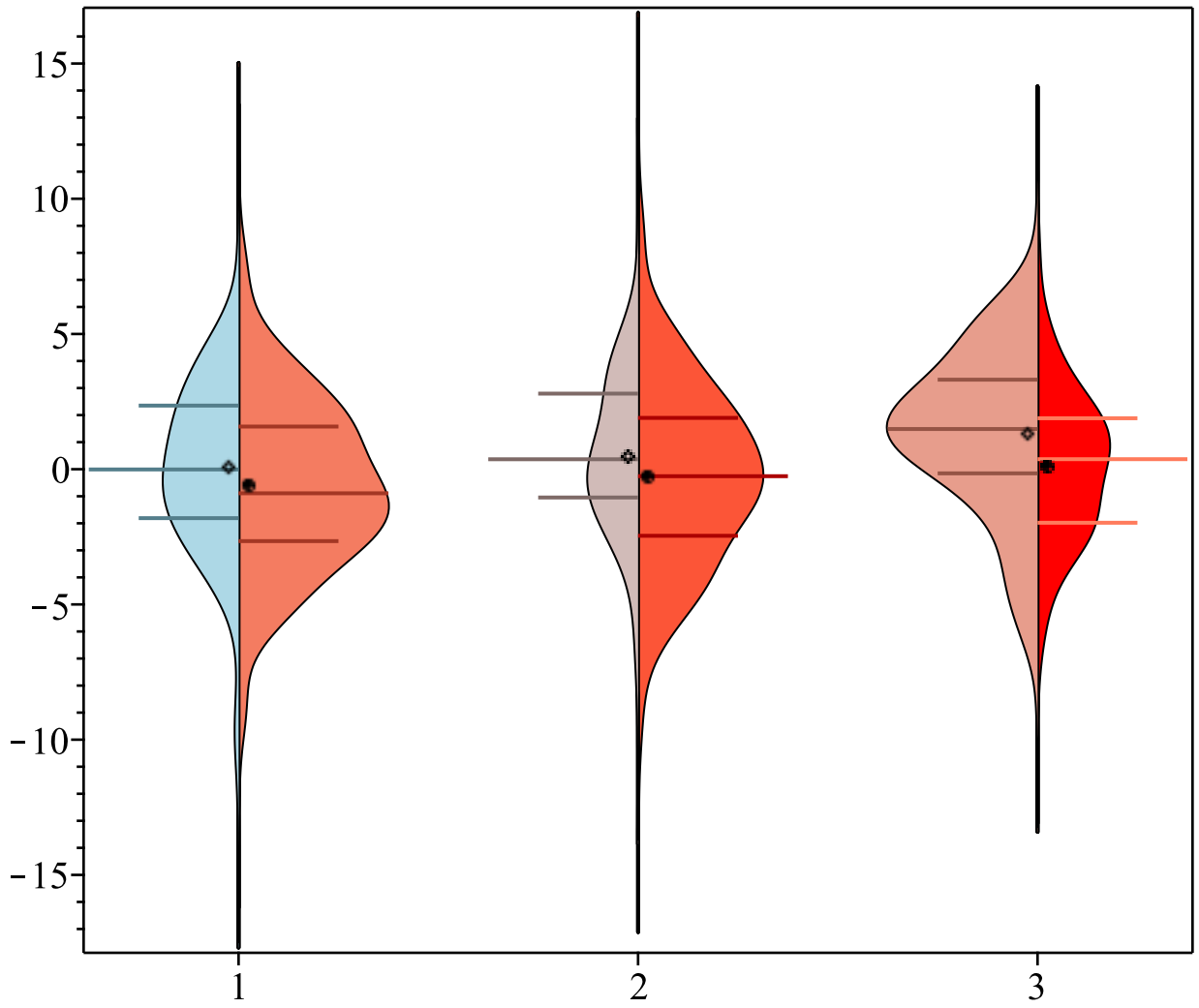
ViolinPlots

ViolinPlots are a visualization of the distribution of data consisting of a rotated kernel density plot and markers for the quartiles and the mean.

```
> C := [seq(Sample(Normal(ln(i), 3), 60), i = 1 .. 20)]:
```

```
> F := [seq(Sample(Normal(sin(i*Pi), 3), 120), i = 1 .. 20)]:
```

```
> ViolinPlot( C[1..3], F[1..3], size = [800,400], color =
  "LightBlue" .. "red", scale = area('pairwise', [1,1,2], [2,3,
  1]));
```



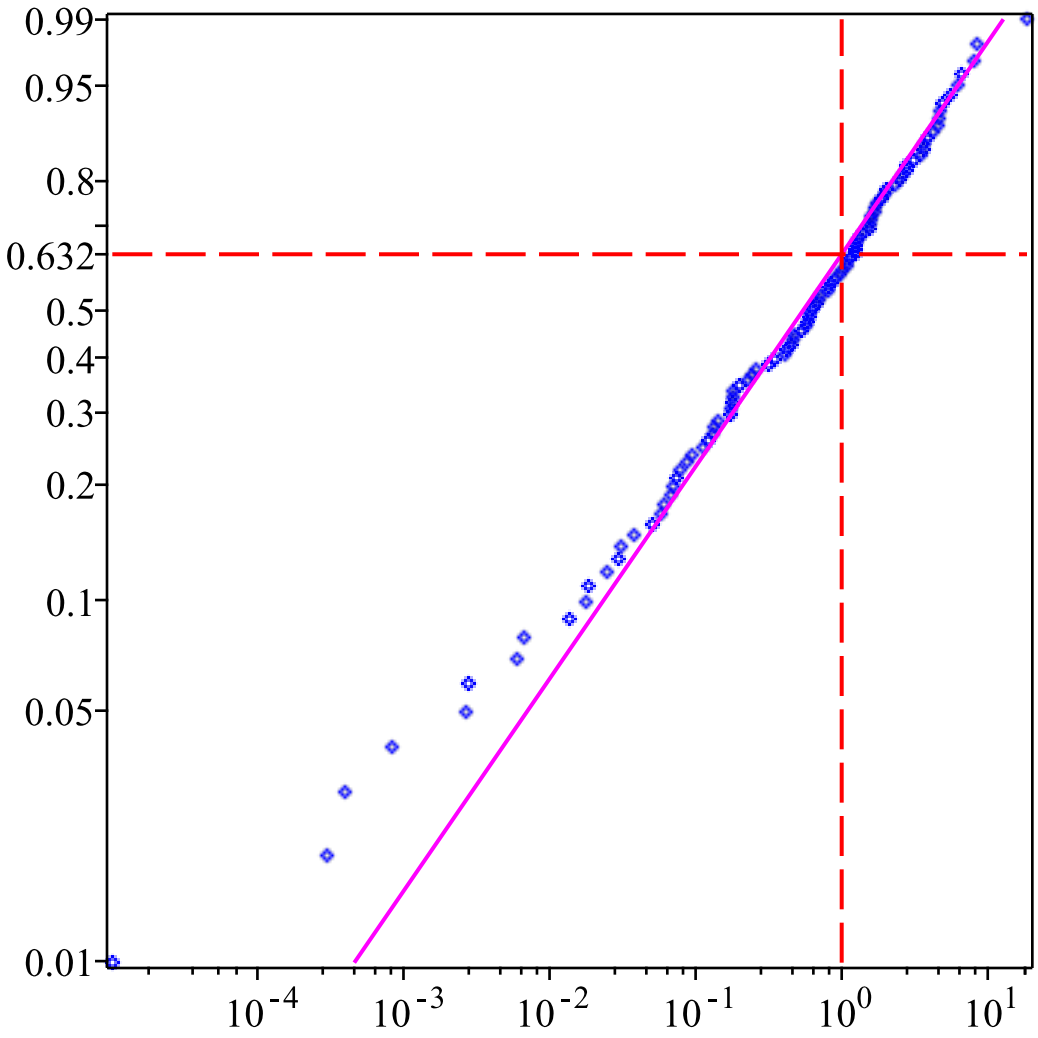
Weibull Plots

Weibull plots are used to verify whether a particular data set follows the Weibull distribution and provides additional information about the estimated shape and scale parameters.

```
> X := RandomVariable(Weibull(1, 0.6)):

> A := Sample(X, 100):

> WeibullPlot(A, scale = 1, shape = 0.6, color = [blue,magenta]
  );
```



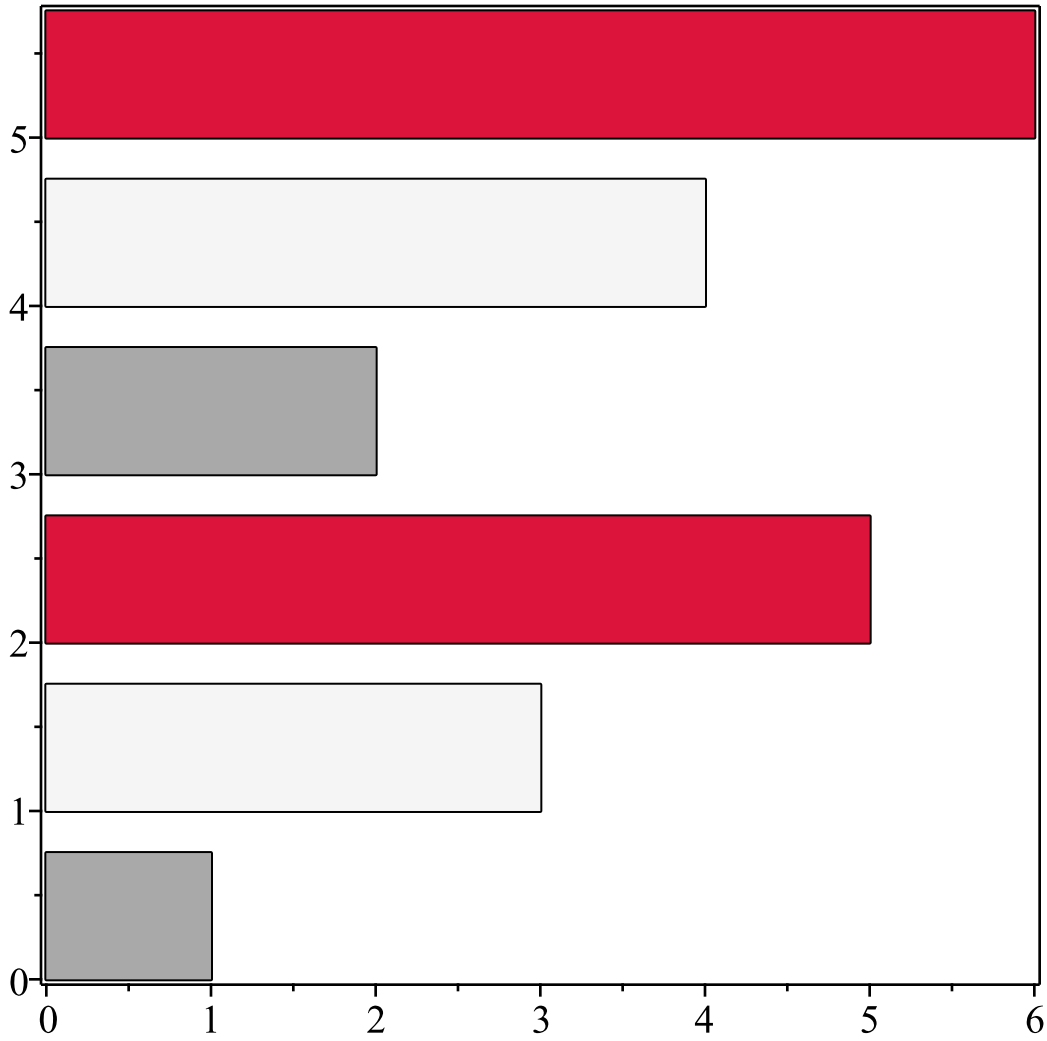
More updates to Statistics Visualizations

Several existing visualizations have also been updated or have new optional arguments.

BarCharts and ColumnGraphs

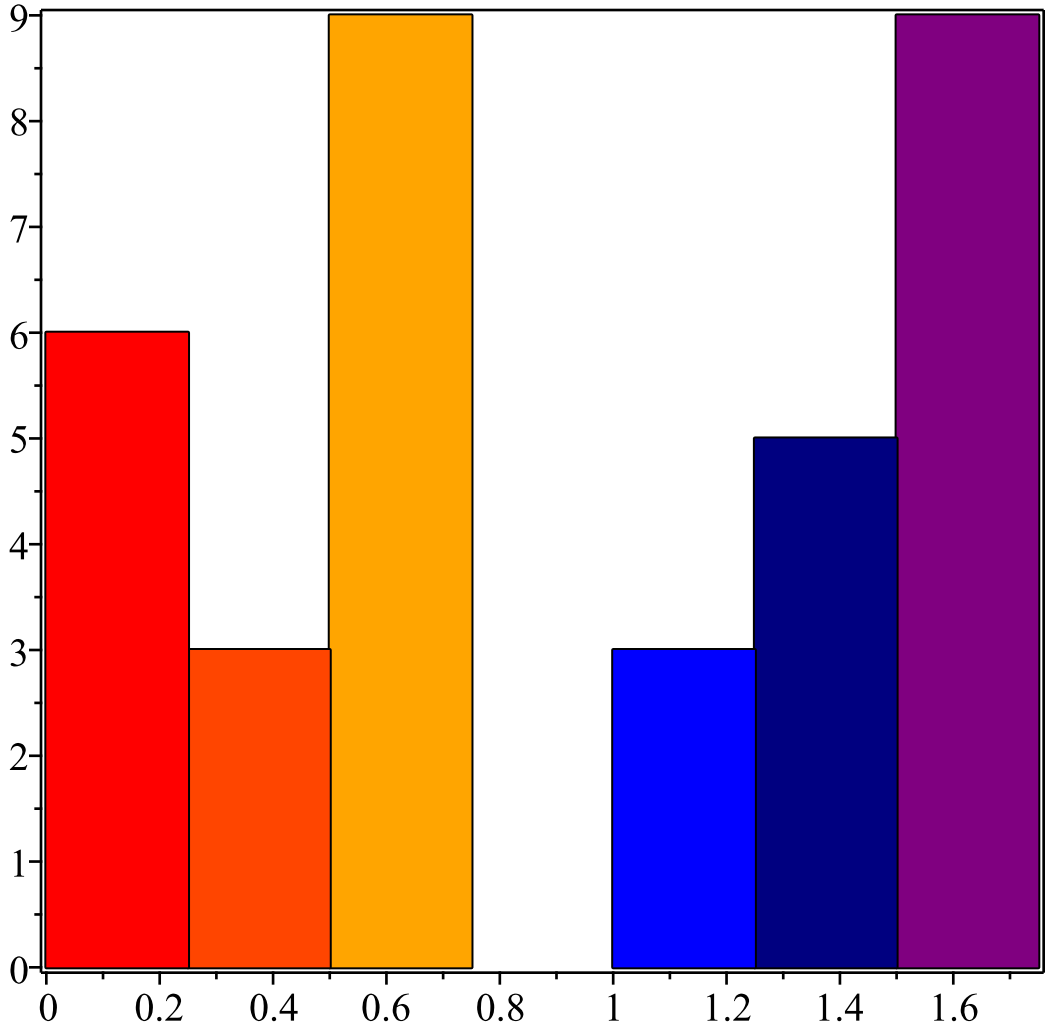
The BarChart / ColumnGraph routines have been updated to support the colorscheme option:

```
> BarChart( < 1, 3, 5, 2, 4, 6 >, colorscheme = [
  "valuesplit", [ 1..2 = "DarkGrey", 3..4 = "WhiteSmoke", 5.
  .6 = "Crimson" ] ] );
```



BarCharts and ColumnGraphs also now support individual color specification for each bar or column.

```
> ColumnGraph(< 6,3,9; 3,5,9>, color = Matrix( [ [ "Red",
  "OrangeRed", "Orange"], ["Blue", "Navy", "Purple" ] ] ) );
```



DensityPlots

The DensityPlot command has a new discont option for the detection of discontinuities.

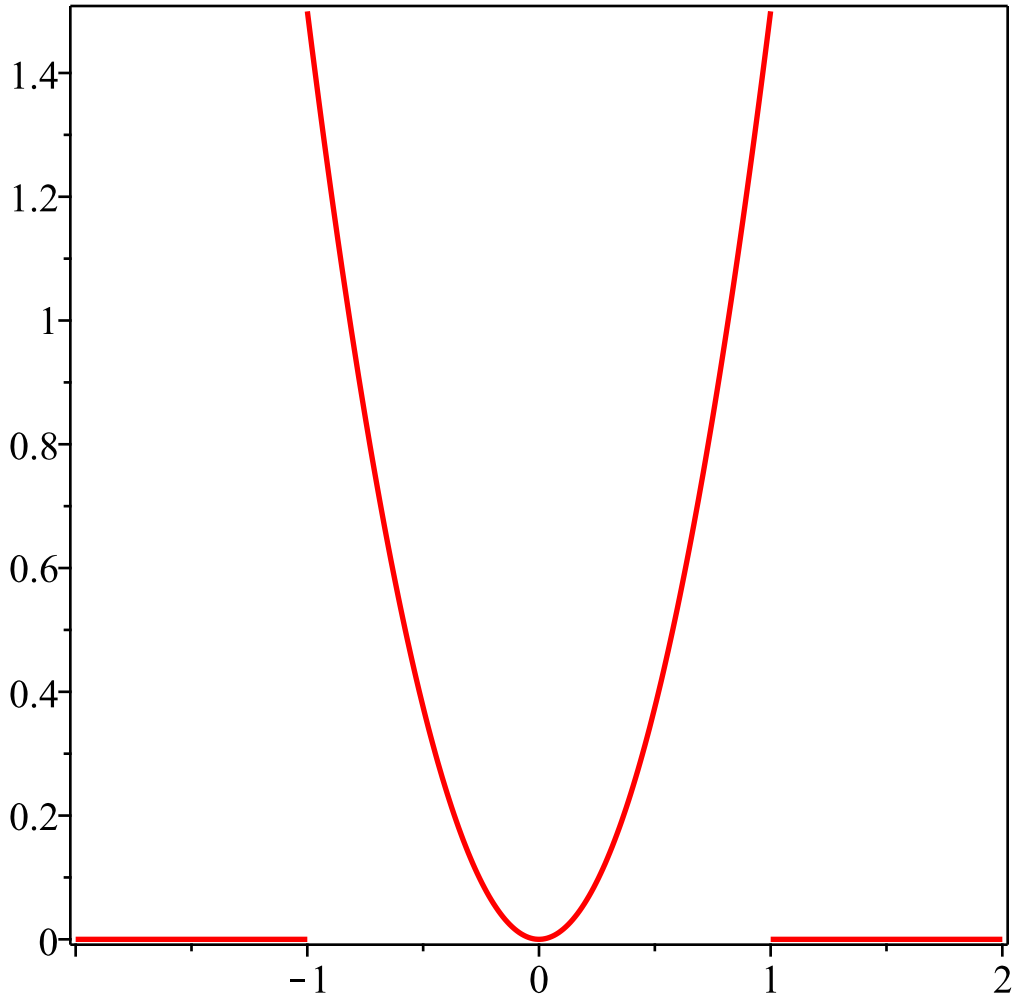
```
> f:=x -> piecewise(-1 < x and x < 1,3/2 *x^2,0);
```

$$f := x \mapsto \begin{cases} \frac{3\,x^2}{2} & -1 < x < 1 \\ 0 & otherwise \end{cases}$$

```
> F:=Distribution(PDF=f):

> Z:=RandomVariable(F):

> DensityPlot(Z,range=-2..2,thickness=2,color=red,discont=
  true);
```



DataFrames and DataSeries

There are several new commands for working with DataFrames and DataSeries: Remove and SubsDatatype. Several existing commands and packages have also been updated to support DataFrames and DataSeries, including sort, Describe, and the CurveFitting package.

Remove

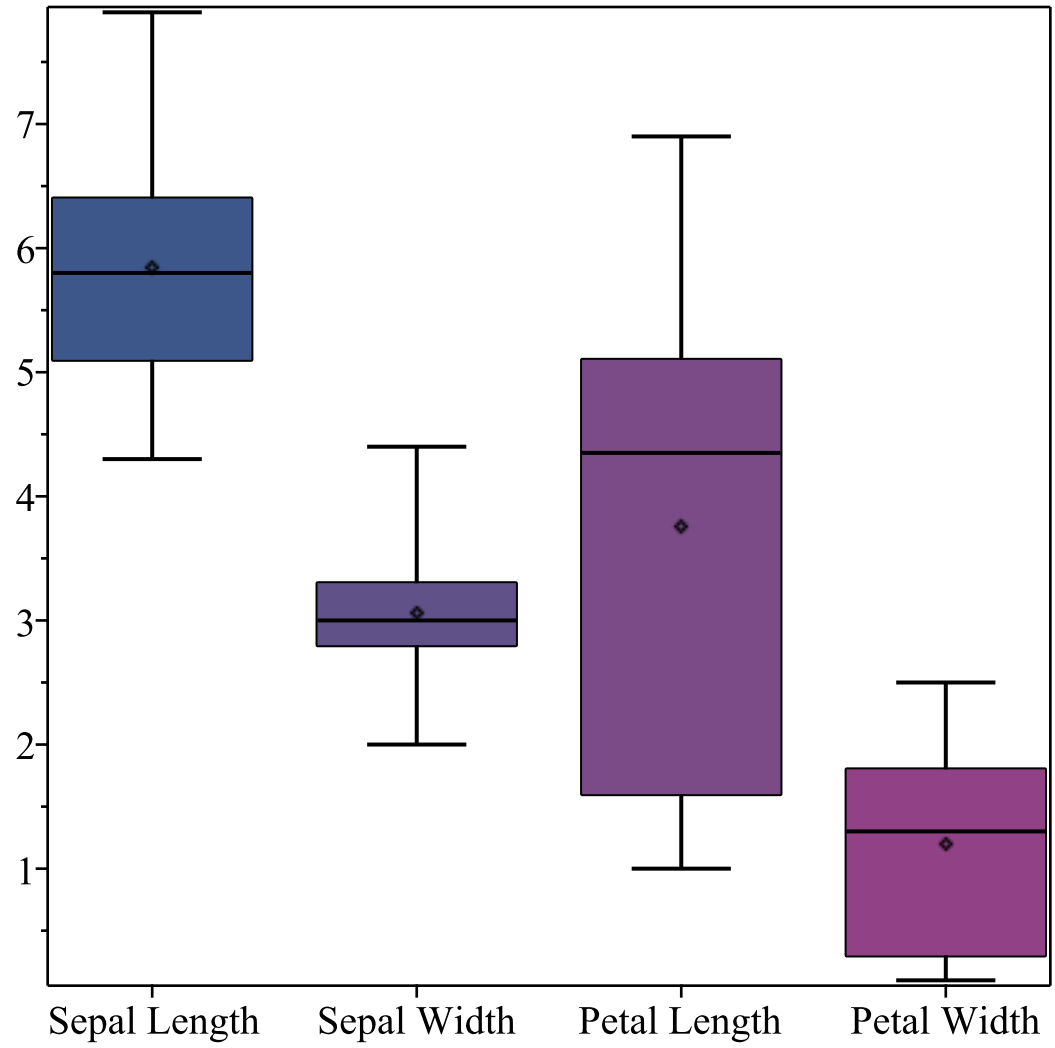
The Remove command makes it easier to remove one or more columns from a DataFrame. For example, many commands in the Statistics package assume that the supplied data is strictly numeric. In the case that there are one or more columns that contain non-numeric data, this command makes it possible to "remove" those columns and produce a result.

```
> with(Statistics):
> IrisData := Import( "datasets/iris.csv", base=datadir );
```

$$IrisData := \begin{bmatrix} & Sepal\ Length & Sepal\ Width & Petal\ Length & Petal\ Width & Species \\ 1 & 5.1 & 3.5 & 1.4 & 0.2 & \text{"setosa"} \\ 2 & 4.9 & 3 & 1.4 & 0.2 & \text{"setosa"} \\ 3 & 4.7 & 3.2 & 1.3 & 0.2 & \text{"setosa"} \\ 4 & 4.6 & 3.1 & 1.5 & 0.2 & \text{"setosa"} \\ 5 & 5 & 3.6 & 1.4 & 0.2 & \text{"setosa"} \\ 6 & 5.4 & 3.9 & 1.7 & 0.4 & \text{"setosa"} \\ 7 & 4.6 & 3.4 & 1.4 & 0.3 & \text{"setosa"} \\ 8 & 5 & 3.4 & 1.5 & 0.2 & \text{"setosa"} \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

Note that the fifth column, "Species", contains strings. Attempting to plot this DataFrame as is would result in an error. However, if the "Species" column is removed, a result is returned.

```
> BoxPlot( :-Remove( IrisData, Species ) );
```



Note that the Remove command does not act in-place. In order to permanently remove a column, re-assignment is necessary.

SubsDataType

The SubsDatatype command changes the specified datatype of a DataSeries. It will also attempt to coerce any data in the DataSeries into the given datatype.

```
> ds := DataSeries( < 1, 2, 3 >, datatype = integer );
```

$$ds := \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix}$$

```
> Datatype( ds );
```

$$integer$$

```
> ds := SubsDatatype( ds, float );
```

$$ds := \begin{bmatrix} 1 & 1. \\ 2 & 2. \\ 3 & 3. \end{bmatrix}$$

```
> Datatype( ds );
```

$$float_8$$

```
> SubsDatatype( IrisData, Species, name, conversion = ( (x) ->
  convert(x,name) ) );
```

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5	3.4	1.5	0.2	setosa
...	...	...	...	...	...

More Updates for DataFrames and DataSeries

There have also been more updates to expand the number of commands that support DataFrame and DataSeries objects. The CurveFitting package now supports each and is also available from the right-click context menu.

sort

The sort command can also be used with DataFrames.

```
> berries := DataFrame( < < 220, 288, 136 > | < 11.94, 18.1,
  7.68 > |
                          < Russia, China, USA > | < "Rubus",
  "Vitis", "Fragaria" > >,
            columns = [ Energy, Carbohydrates, `Top
  Producer`, Genus ],
            rows = [ Raspberry, Grape, Strawberry ] );
```

$$berries := \begin{bmatrix} & Energy & Carbohydrates & Top\ Producer & Genus \\ Raspberry & 220 & 11.94 & Russia & "Rubus" \\ Grape & 288 & 18.1 & China & "Vitis" \\ Strawberry & 136 & 7.68 & USA & "Fragaria" \end{bmatrix}$$

Here the DataFrame is sorted in order of ascending energy level:

```
> sort( berries, Energy );
```

$$\begin{bmatrix} & Energy & Carbohydrates & Top\ Producer & Genus \\ Strawberry & 136 & 7.68 & USA & "Fragaria" \\ Raspberry & 220 & 11.94 & Russia & "Rubus" \\ Grape & 288 & 18.1 & China & "Vitis" \end{bmatrix}$$

It is also possible to sort columns with string and named values:

```
> sort( berries, Genus );
```

$$\begin{bmatrix} & Energy & Carbohydrates & Top\ Producer & Genus \\ Strawberry & 136 & 7.68 & USA & "Fragaria" \\ Raspberry & 220 & 11.94 & Russia & "Rubus" \\ Grape & 288 & 18.1 & China & "Vitis" \end{bmatrix}$$

Describe

The Describe command returns a printed description for procedures, modules and objects. In Maple 2017, the Describe command has been extended to provide a description for DataFrames and DataSeries that includes the number of observations (rows), the number of variables (columns), as well as the type of each column (if specified). In addition, for numeric columns, the minimum and maximum values are displayed, and for truefalse, string or name columns, the distinct levels are given.

```
> Describe( berries );
```

```
berries :: DataFrame: 3 observations for 4 variables
Energy:         Type: anything  Min: 136.00  Max: 288.00
Carbohydrates:  Type: anything  Min: 7.68  Max: 18.10
Top Producer:   Type: anything  Tally: [Russia = 1, China = 1,
USA = 1]
Genus:          Type: anything  Tally: ["Vitis" = 1, "Fragaria"
= 1, "Rubus" = 1]
```