AMS-512 Capital Markets and Portfolio Theory

Univariate and Multivariate Distributions

Robert J. Frey, Research Professor
Stony Brook University, Applied Mathematics and Statistics

Robert.Frey@StonyBrook.edu
http://www.ams.sunysb.edu/~frey

Univariate Normal

PDF

$$f_X(x) \geq 0$$

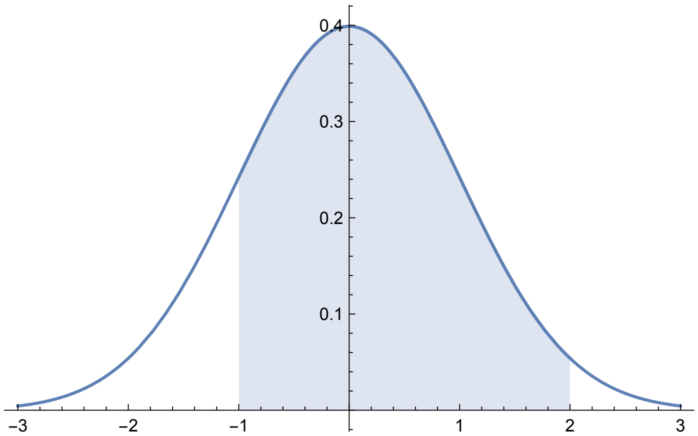
$$\int_{-\infty}^{\infty} f_X(x)\, dx = 1$$

$$P\{X \in [a,\ b]\} \equiv \int_a^b f_X(x)\, dx$$

Example - Standard Normal Distribution - Probability an outcome is ϵ [-1, 2] equals the area under the curve.

In[225]:=
```
Show[
 Plot[PDF[NormalDistribution[0, 1], x], {x, -3, 3}],
 Plot[PDF[NormalDistribution[0, 1], x], {x, -1, 2}, Filling → 0]
]
```

Out[225]=



In[226]:=
```
∫_{-1}^{2} PDF[NormalDistribution[0., 1.], x] ⅆx
```

Out[226]= 0.818595

CDF

$$F_X(x) \equiv P\{X \le x\} = \int_{-\infty}^{x} f_X(u)\, du$$

$$F_X(-\infty) = 0 \quad \wedge \quad F_X(\infty) = 1$$

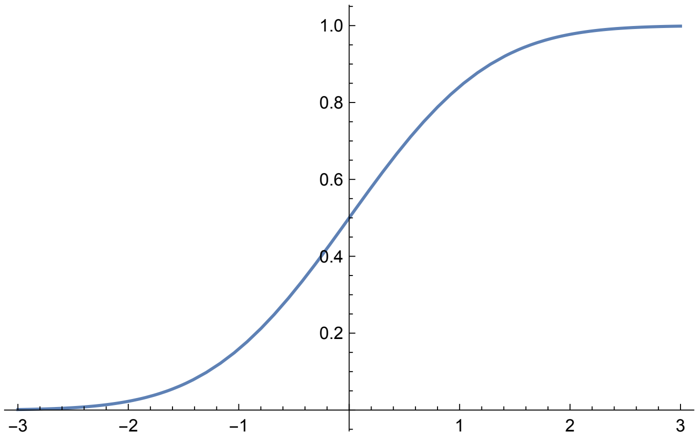
Also, note, therefore, that

$$P\{X \in [a,\ b]\} \equiv \int_a^b f_X(x)\, dx = F_X(b) - F_X(a)$$

Example - Standard Normal Distribution - Probability an outcome is $\in$ [-1, 2] equals the difference in cumulative probabilities

In[227]:=
```
Plot[CDF[NormalDistribution[0, 1], x], {x, -3, 3}]
```

Out[227]=



In[228]:=
```
CDF[NormalDistribution[0., 1.], 2] - CDF[NormalDistribution[0., 1.], -1]
```

Out[228]= 0.818595

In[229]:=
```
CDF[NormalDistribution[0, 1], x]
```

Out[229]= $\frac{1}{2}\,\mathrm{Erfc}\left[-\frac{x}{\sqrt{2}}\right]$

In[230]:=
```
PDF[NormalDistribution[0, 1], x]
```

Out[230]= $\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$

In[231]:=
```
PDF[NormalDistribution[μ, σ], x]
```

Out[231]= $\frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$

Characteristic Function

$$\phi_X(\omega) \equiv \mathrm{E}\left\{e^{i\,\omega\,X}\right\} = \int_{-\infty}^{\infty} f_X(x)\, e^{i\,\omega\,x}\, dx$$

Example - Standard Normal Distribution

The characteristic function of the Standard Normal Distribution has the following simple form.

In[232]:=
```
CharacteristicFunction[NormalDistribution[0, 1], ω]
```

Out[232]= $e^{-\frac{\omega^2}{2}}$

The characteristic function is a suitably parameterized Fourier Transform of the PDF. Read the *Mathematica* documentation for FourierTransform[] for justification for the setting of the FourierParamaters option.

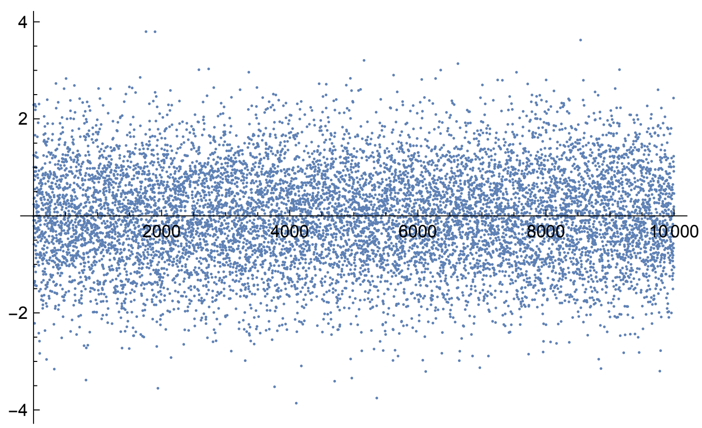
In[233]:=
```
FourierTransform[PDF[NormalDistribution[0, 1], x],
 x, ω, FourierParameters -> {1, -1}]
```

Out[233]= $e^{-\frac{\omega^2}{2}}$

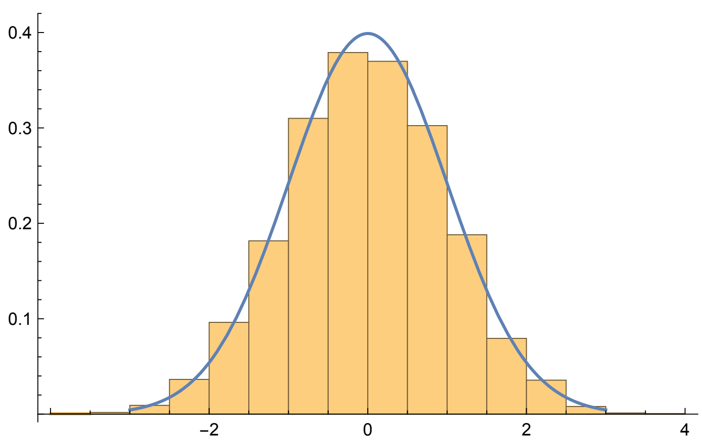
Summary Statistics

In[234]:=
```
sim = RandomVariate[NormalDistribution[], 10 000];
```

In[235]:=
```
ListPlot[sim, PlotRange → All]
```

Out[235]=


In[236]:=
```
Show[
 Histogram[sim, Automatic, "PDF"],
 Plot[PDF[NormalDistribution[], x], {x, -3, 3}]
]
```

Out[236]=


In[237]:=
```
PDF[NormalDistribution[μ, σ], x]
```

Out[237]= $$\frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi}\,\sigma}$$

Mean

$$\mu = E[X] = \int_{-\infty}^{\infty} x\, f_X(x)\, dx$$

In[238]:=
```
Mean[NormalDistribution[μ, σ]]
```

Out[238]= μ

In[239]:=
```
Mean[sim]
```

Out[239]= -0.0178969

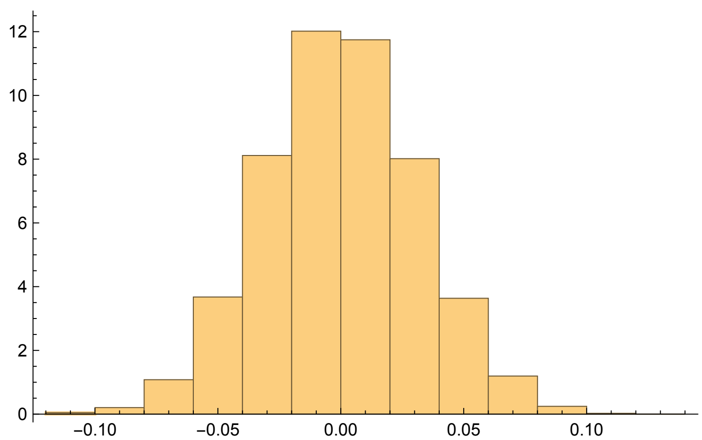
In[240]:=
```
sim10K = RandomVariate[NormalDistribution[], {10 000}];
```

In[241]:=
```
Mean[sim10K]
```

Out[241]= -0.00242826

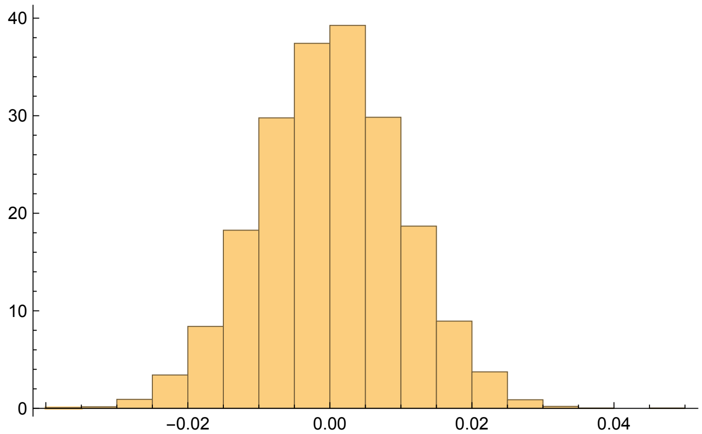
In[242]:=
```
Histogram[
 Table[
  Mean[RandomVariate[NormalDistribution[], {1000}]],
  {10 000}
 ],
 Automatic,
 "PDF"
]
```

Out[242]=



In[243]:=
```
Histogram[
 Table[
  Mean@RandomVariate[NormalDistribution[], {10 000}],
  {10 000}
 ],
 20,
 "PDF"
]
```

Out[243]=



In[244]:=
```
? Histogram
```

Out[244]=

Symbol

Histogram[{x_1, x_2, …}] plots a histogram of the values x_i.

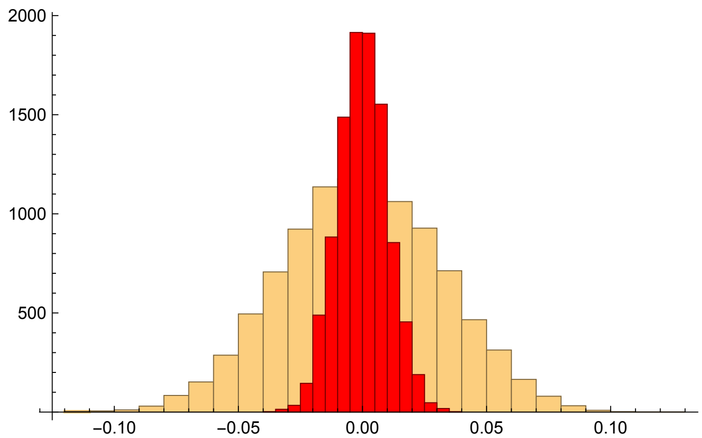
Histogram[{x_1, x_2, …}, *bspec*] plots a histogram with bin width specification *bspec*.

Histogram[{x_1, x_2, …}, *bspec*, *hspec*] plots a histogram with bin heights computed according to the specification *hspec*.

Histogram[{$data_1$, $data_2$, …}, …] plots histograms for multiple datasets $data_i$.

In[245]:=
```
Show[
 Histogram[
  Table[
   Mean@RandomVariate[NormalDistribution[], {1000}],
   {10 000}
  ],
  20
 ],
 Histogram[
  Table[
   Mean@RandomVariate[NormalDistribution[], {10 000}],
   {10 000}
  ],
  20,
  ChartStyle → Red
 ],
 PlotRange → All
]
```

Out[245]=



Variance

$$\sigma^2 = E\left[(X-\mu)^2\right] = \int_{-\infty}^{\infty} (x-\mu)^2 f_X(x)\, dx$$

In[246]:=
```
Variance[NormalDistribution[μ, σ]]
```

Out[246]= σ^2

In[247]:=
```
Variance[sim]
```

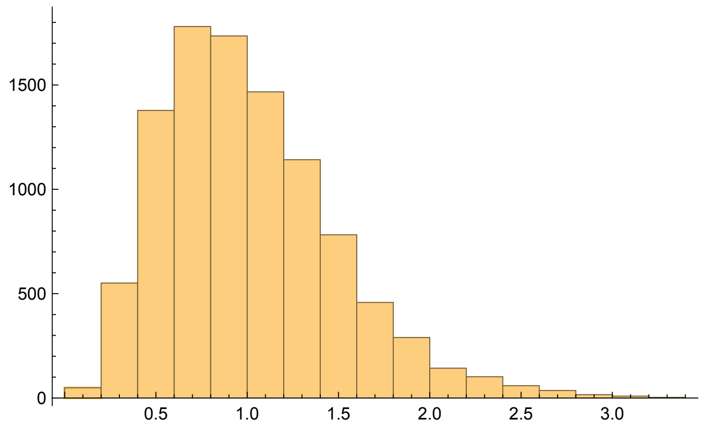
Out[247]= 1.00831

In[248]:=
```
StandardDeviation[sim]
```

Out[248]= 1.00415

In[249]:=
```
Histogram[
 Table[
  Variance@RandomVariate[NormalDistribution[], {10}],
  {10 000}
 ],
 20
]
```

Out[249]=



Skewness

$$\gamma_3 = E\left[\left(\frac{X-\mu}{\sigma}\right)^3\right] = \frac{1}{\sigma^3}\int_{-\infty}^{\infty}(x-\mu)^3 f_X(x)\,dx$$

In[250]:=
```
Skewness[NormalDistribution[μ, σ]]
```

Out[250]= 0

In[251]:=
```
Skewness[ChiSquareDistribution[10.]]
```

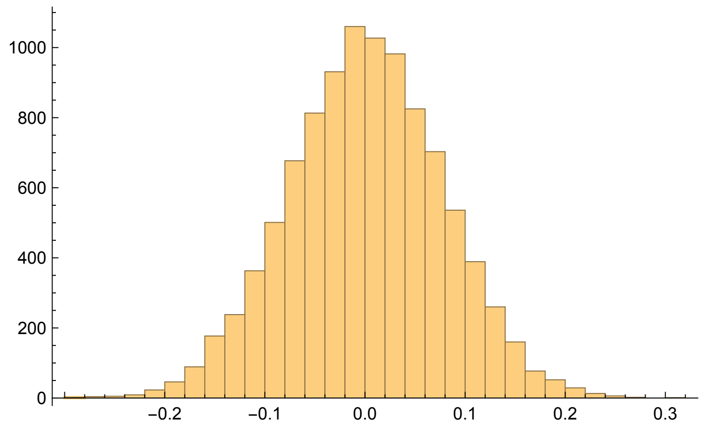
Out[251]= 0.894427

In[252]:=
```
Skewness[sim]
```

Out[252]= -0.0181231

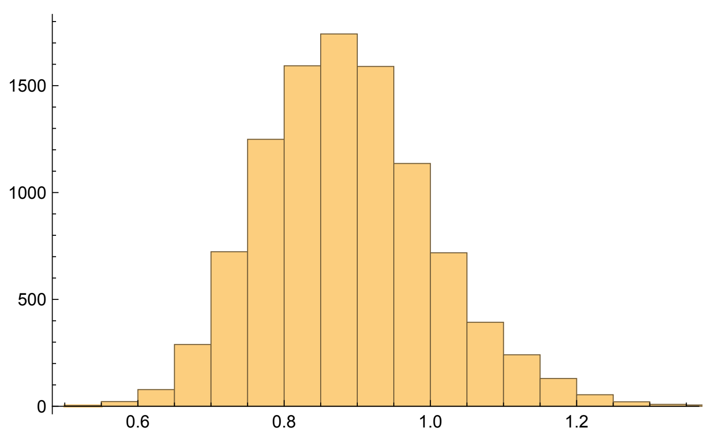
In[253]:=
```
Histogram[
 Table[
  Skewness@RandomVariate[NormalDistribution[], {1000}],
  {10 000}
 ],
 20
]
```

Out[253]=



In[254]:=
```
Histogram[
 Table[
  Skewness@RandomVariate[ChiSquareDistribution[10.], {1000}],
  {10 000}
 ],
 20
]
```

Out[254]=



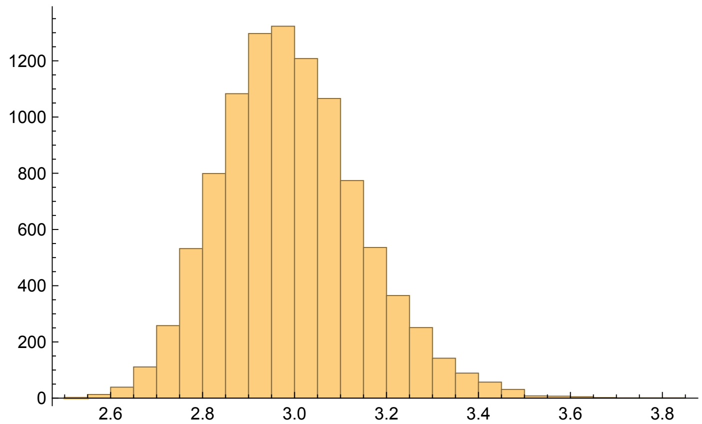
Kurtosis

$$\gamma_4 = E\left[\frac{(X-\mu)^4}{\sigma^4}\right] = \frac{1}{\sigma^4}\int_{-\infty}^{\infty}(x-\mu)^4 f_X(x)\,dx$$

In[255]:=
```
Kurtosis[NormalDistribution[μ, σ]]
```
Out[255]= 3

In[256]:=
```
Kurtosis[sim]
```
Out[256]= 2.95497

In[257]:=
```
Histogram[
 Table[
  Kurtosis@RandomVariate[NormalDistribution[], {1000}],
  {10 000}
 ],
 20
]
```
Out[257]=



Central Limit Theorem

Sums of i.i.d. random variables convert to a class of *stable distributions*. (See https://en.wikipedia.org/wiki/Central_limit_theorem.) In the case of continuous distributions with finite variance the stable distribution is the Gaussian or Normal distribution. In contrast, the sum of a number of i.i.d. random variables with power law tail distributions decreasing as $|x|^{-(1+\alpha)}$, where $0 < \alpha < 2$ (and therefore having infinite variance) will tend to an alpha-stable distribution with stability parameter (or index of stability) of α as the number of variables grows. (See https://en.wikipedia.org/wiki/Stable_distribution.)

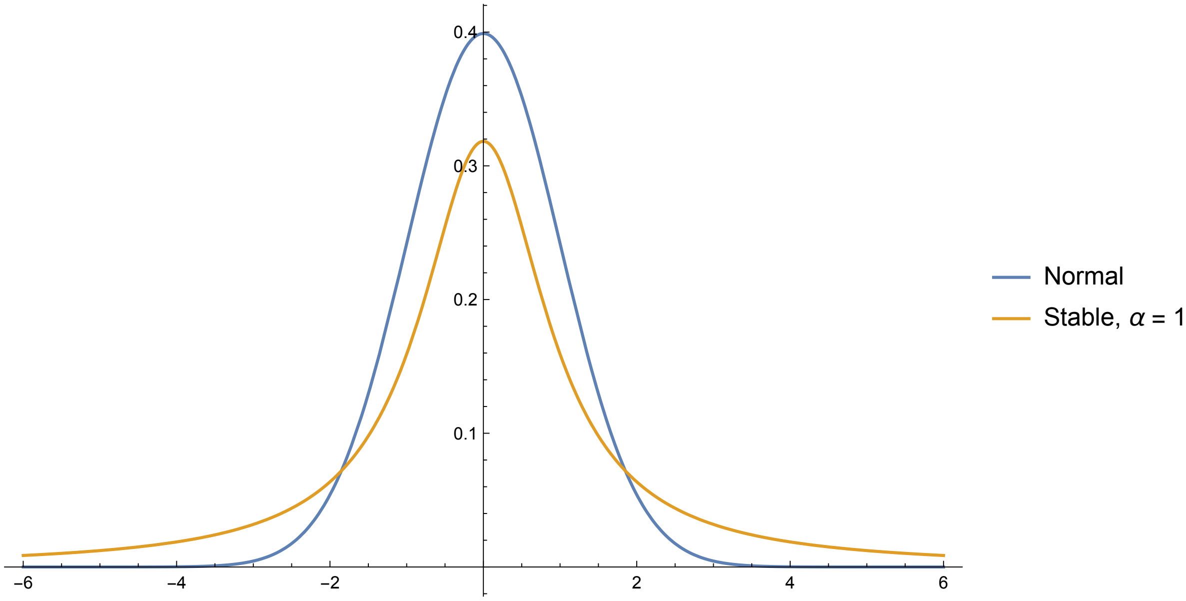
In[258]:=
```
? StableDistribution
```
Out[258]=

Symbol

StableDistribution[*type*, α, β, μ, σ] represents the stable distribution S_{type} with index of stability α, skewness parameter β, location parameter μ, and scale parameter σ.

In[259]:=
```
Plot[
 Evaluate[{PDF[NormalDistribution[], x], PDF[StableDistribution[1, 0, 0, 1], x]}],
 {x, -6, 6}, PlotLegends → {"Normal", "Stable, α = 1"}, ImageSize → 500]
```

Out[259]=



Distributions

Normal Distribution

In[260]:=
```
? NormalDistribution
```

Out[260]=

Symbol

NormalDistribution[μ, σ] represents a normal (Gaussian) distribution with mean μ and standard deviation σ.
NormalDistribution[] represents a normal distribution with zero mean and unit standard deviation.

Cauchy Distribution

In[261]:=
```
? CauchyDistribution
```

Out[261]=

Symbol

CauchyDistribution[a, b] represents a Cauchy distribution with location parameter a and scale parameter b.
CauchyDistribution[] represents a Cauchy distribution with location parameter 0 and scale parameter 1.

In[262]:=
```
PDF[CauchyDistribution[a, b], x]
```

Out[262]=

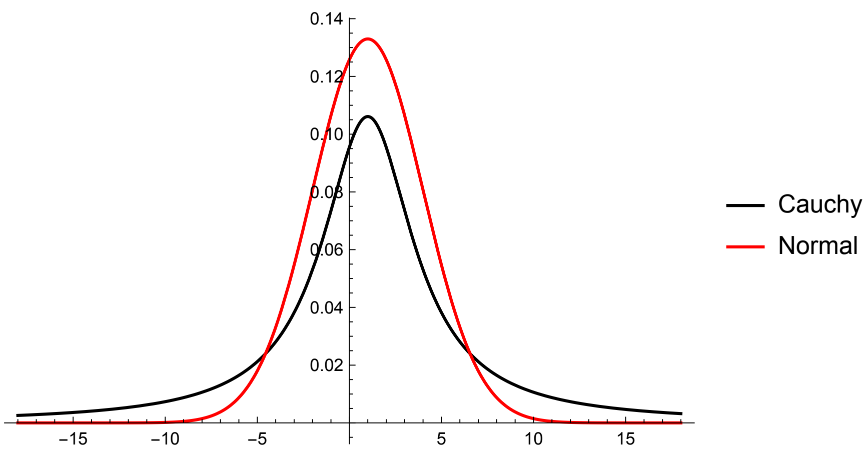
$$\frac{1}{b\,\pi\left(1+\frac{(-a+x)^2}{b^2}\right)}$$

In[263]:=
```
Mean[CauchyDistribution[a, b]]
```

Out[263]=
```
Indeterminate
```

In[264]:=
```
Plot[
 Evaluate@{
   PDF[CauchyDistribution[1, 3], x],
   PDF[NormalDistribution[1, 3], x]
  },
 {x, -18, 18},
 PlotStyle → {Black, Red},
 PlotLegends → {"Cauchy", "Normal"}
]
```

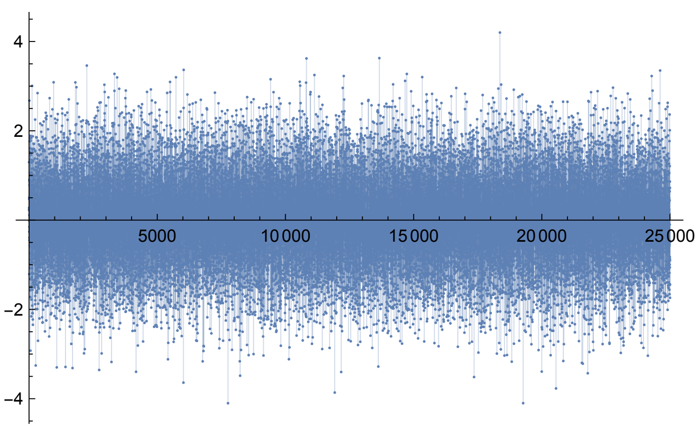
Out[264]=



In[265]:=
```
simN = RandomVariate[NormalDistribution[], 25 000];
simC = RandomVariate[CauchyDistribution[], 25 000];
```

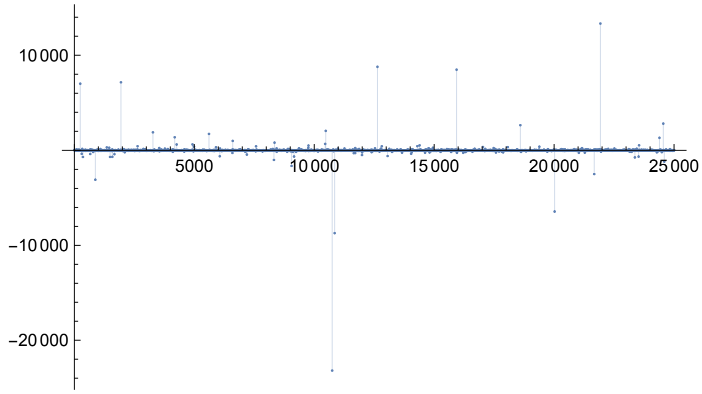
In[267]:=
```
ListPlot[simN, Filling → 0, PlotRange → All]
```

Out[267]=



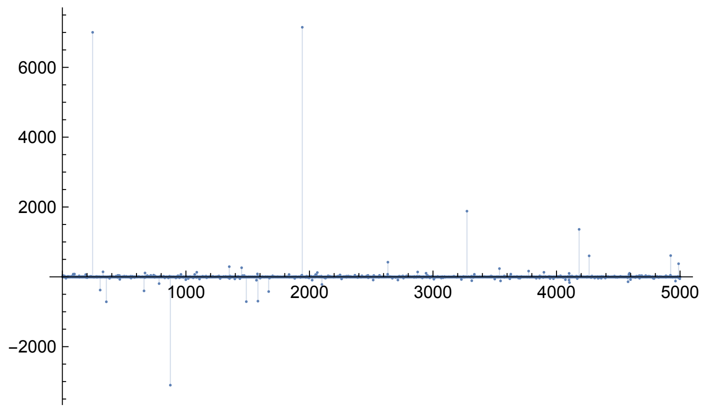
In[268]:=
```
ListPlot[simC, Filling → 0, PlotRange → All]
```

Out[268]=



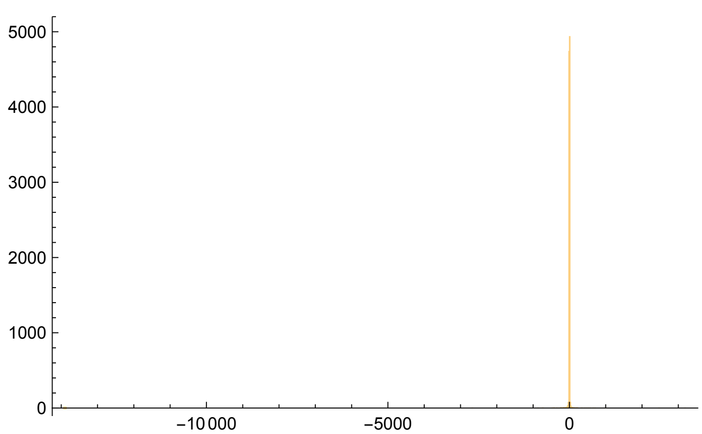
In[269]:=
```
ListPlot[simC⟦ ;; 5000⟧, Filling → 0, PlotRange → All]
```

Out[269]=



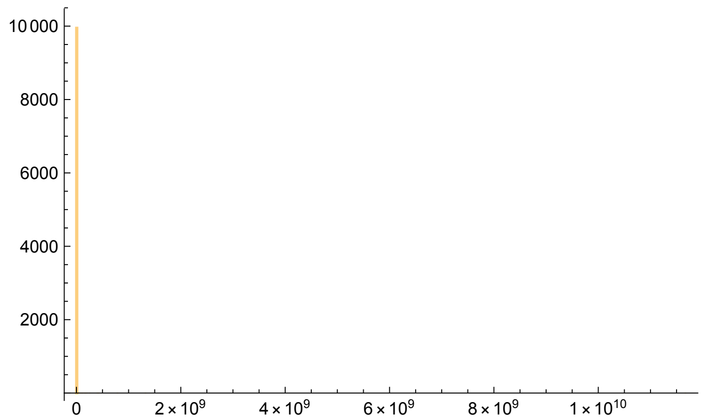
In[270]:=
```
Histogram[
 Table[
  Mean@RandomVariate[CauchyDistribution[], {1000}],
  {10 000}
 ],
 Automatic,
 PlotRange → All
]
```

Out[270]=



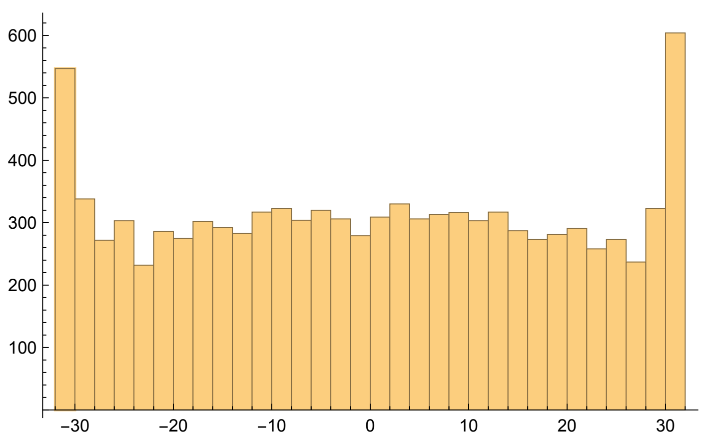
In[271]:=
```
Histogram[
 Table[
  Variance@RandomVariate[CauchyDistribution[], {1000}],
  {10 000}
 ],
 Automatic,
 PlotRange → All
]
```

Out[271]=



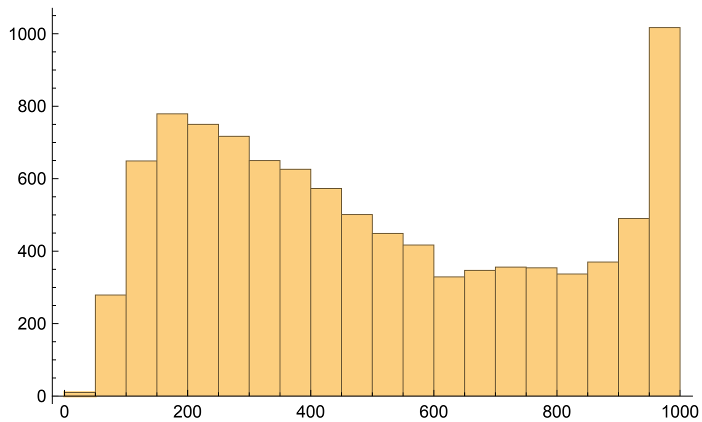
In[272]:=
```
Histogram[
 Table[
  Skewness@RandomVariate[CauchyDistribution[], {1000}],
  {10 000}
 ],
 Automatic,
 PlotRange → All
]
```

Out[272]=



In[273]:=
```
Histogram[
 Table[
  Kurtosis@RandomVariate[CauchyDistribution[], {1000}],
  {10 000}
 ],
 Automatic,
 PlotRange → All
]
```

Out[273]=



Student t Distribution

In[274]:=
```
? StudentTDistribution
```

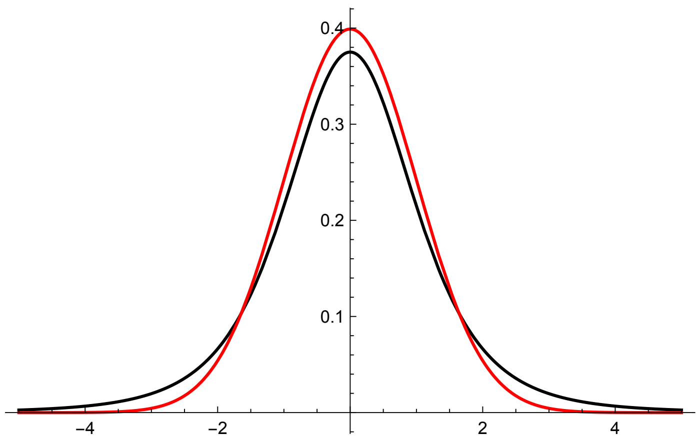
Out[274]=

Symbol

StudentTDistribution[ν] represents a Student t distribution with ν degrees of freedom.
StudentTDistribution[μ, σ, ν] represents a Student t distribution
with location parameter μ, scale parameter σ, and ν degrees of freedom.

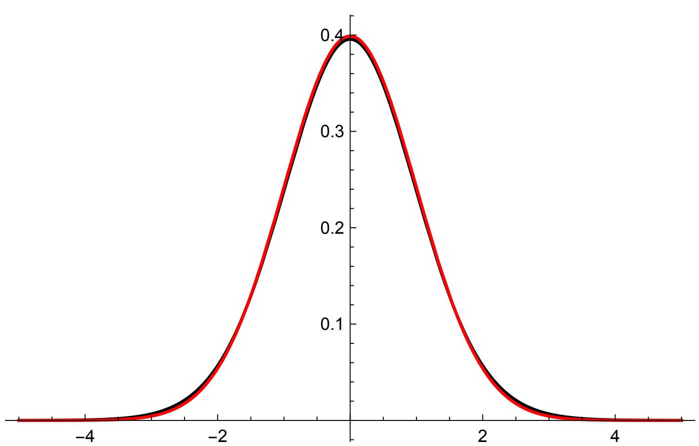
In[275]:=
```
Plot[
 Evaluate@{
   PDF[StudentTDistribution[4], x],
   PDF[NormalDistribution[], x]
  },
 {x, -5, 5},
 PlotStyle → {Black, Red}
]
```

Out[275]=



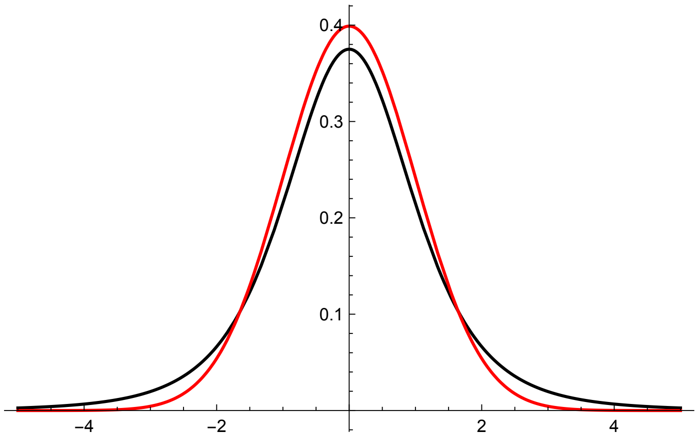
In[276]:=
```
Plot[
 Evaluate@{
   PDF[StudentTDistribution[30], x],
   PDF[NormalDistribution[], x]
  },
 {x, -5, 5},
 PlotStyle → {Black, Red}
]
```

Out[276]=



In[277]:=
```
Plot[
 Evaluate@{
   PDF[StudentTDistribution[4], x],
   PDF[NormalDistribution[], x]
  },
 {x, -5, 5},
 PlotStyle → {Black, Red}
]
```

Out[277]=



Log Normal Distribution

In[278]:=
```
? LogNormalDistribution
```

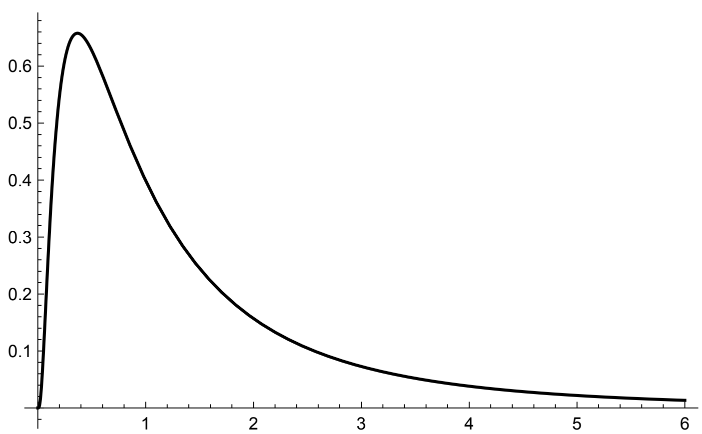
Out[278]=

Symbol

LogNormalDistribution[μ, σ] represents a lognormal distribution derived from a normal distribution with mean μ and standard deviation σ.

In[279]:=
```
Plot[Evaluate[PDF[LogNormalDistribution[0, 1], x]], {x, 0, 6}, PlotStyle → {Black}]
```

Out[279]=



In[280]:= `Mean[LogNormalDistribution[μ, σ]]`

Out[280]= $e^{\mu+\frac{\sigma^2}{2}}$

In[281]:= `StandardDeviation[LogNormalDistribution[μ, σ]]`

Out[281]= $\sqrt{e^{2\mu+\sigma^2}\left(-1+e^{\sigma^2}\right)}$

In[282]:= `Kurtosis[LogNormalDistribution[μ, σ]]`

Out[282]= $-3+3\,e^{2\sigma^2}+2\,e^{3\sigma^2}+e^{4\sigma^2}$

Gamma Distribution

In[283]:= `? GammaDistribution`

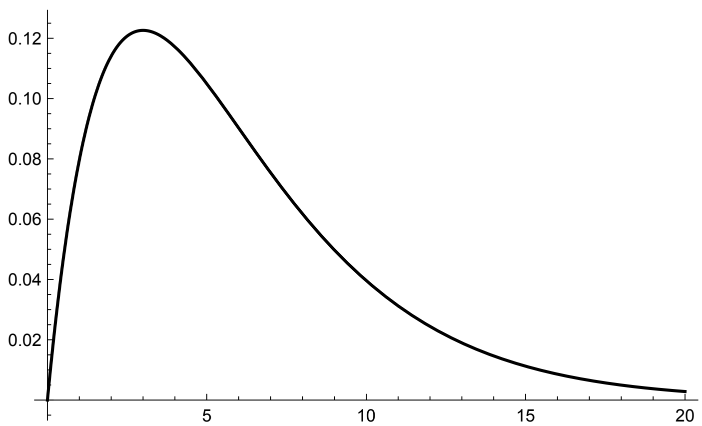
Out[283]=

Symbol

GammaDistribution[α, β] represents a gamma distribution with shape parameter α and scale parameter β.
GammaDistribution[$\alpha, \beta, \gamma, \mu$] represents a generalized gamma distribution with shape parameters α and γ, scale parameter β, and location parameter μ.

In[284]:= `Plot[Evaluate[PDF[GammaDistribution[2, 3], x]], {x, 0, 20}, PlotStyle → {Black}]`

Out[284]=



Multivariate

PDF

$$f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n) \geq 0 \quad \wedge \quad \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n)\,dx_1\ldots dx_n = 1$$

$$P[\{X_1,\ \ldots,\ X_n\} \in \mathcal{R}] \equiv \int\cdots\int_{\mathcal{R}} f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n)\,dx_1\ldots dx_n$$

$$P\{X_1 \in [a_1,\ b_1] \wedge \ldots \wedge X_n \in [a_n,\ b_n]\} = \int_{a_1}^{b_1} \ldots \int_{a_n}^{b_n} f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n)\, dx_1 \ldots dx_n$$

Example - Multivariate Normal Distribution

$$\mathbf{x} = (x_1,\ \ldots,\ x_n)$$

$$f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n \mid \boldsymbol{\mu},\ \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(\boldsymbol{\Sigma})}}\, e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

In[285]:= `PDF[MultinormalDistribution[{μ1, μ2}, {{σ1^2, σ1,2}, {σ1,2, σ2^2}}], {x1, x2}]`

$$\text{PDF}\left[\text{MultinormalDistribution}\left[\{\mu_1, \mu_2\}, \{\{\sigma_1{}^2, \sigma_{1,2}\}, \{\sigma_{1,2}, \sigma_2{}^2\}\}\right], \{x1, x2\}\right]$$

Out[285]=

$$\frac{e^{\frac{1}{2}\left(-\frac{(x2-\mu_2)\,(x2\,\sigma_1^2-\mu_2\,\sigma_1^2-x1\,\sigma_{1,2}+\mu_1\,\sigma_{1,2})}{\sigma_1^2\,\sigma_2^2-\sigma_{1,2}^2}-\frac{(x1-\mu_1)\,(-x1\,\sigma_2^2+\mu_1\,\sigma_2^2+x2\,\sigma_{1,2}-\mu_2\,\sigma_{1,2})}{-\sigma_1^2\,\sigma_2^2+\sigma_{1,2}^2}\right)}}{2\,\pi\,\sqrt{\sigma_1^2\,\sigma_2^2-\sigma_{1,2}^2}}$$

Example - Bivariate Standard Multivariate Normal Distribution

The standard multivariate Normal distribution has $\boldsymbol{\mu} = \{0, 0\}$ and $\boldsymbol{\Sigma} = \{\{1, 0\}, \{0, 1\}\}$.

$$\phi_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n) = \frac{1}{(2\pi)^{n/2}}\, e^{-\frac{1}{2}\mathbf{x}^T\mathbf{x}}$$

In[286]:= `PDF[MultinormalDistribution[{0, 0}, {{1, 0}, {0, 1}}], {x1, x2}]`

Out[286]=

$$\frac{e^{\frac{1}{2}\left(-x1^2-x2^2\right)}}{2\,\pi}$$

In[287]:= `PDF[MultinormalDistribution[{0, 0, 0}, IdentityMatrix[3]], {x1, x2, x3}]`

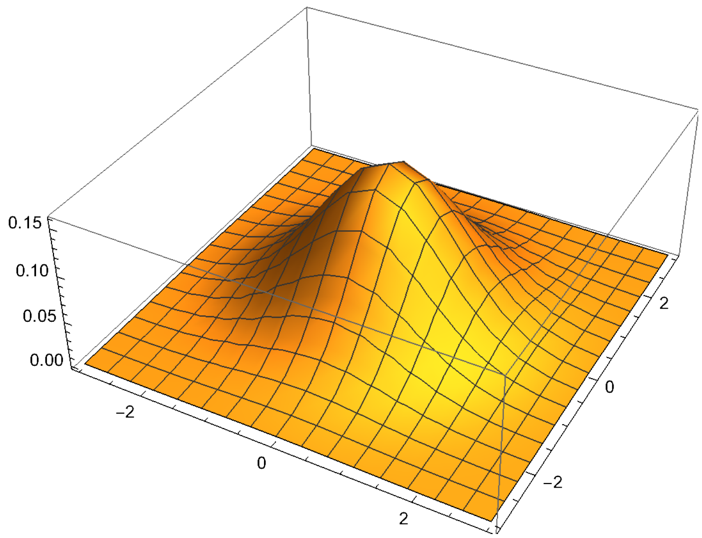
Out[287]=

$$\frac{e^{\frac{1}{2}\left(-x1^2-x2^2-x3^2\right)}}{2\,\sqrt{2}\,\pi^{3/2}}$$

Example - Bivariate Normal

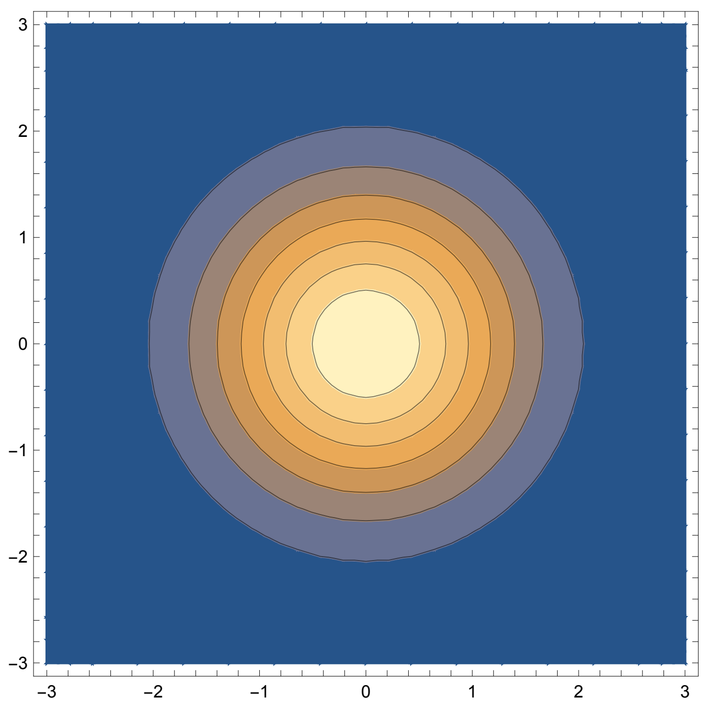
In[288]:=
```
Plot3D[Evaluate@PDF[MultinormalDistribution[{0, 0}, IdentityMatrix[2]], {x1, x2}],
 {x1, -3, 3}, {x2, -3, 3}]
```

Out[288]=



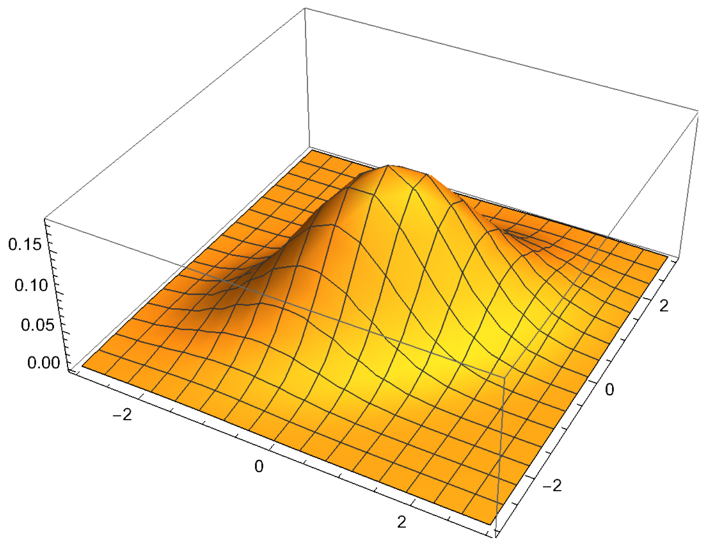
In[289]:=
```
ContourPlot[PDF[MultinormalDistribution[{0, 0}, IdentityMatrix[2]], {x1, x2}],
 {x1, -3, 3}, {x2, -3, 3}]
```

Out[289]=



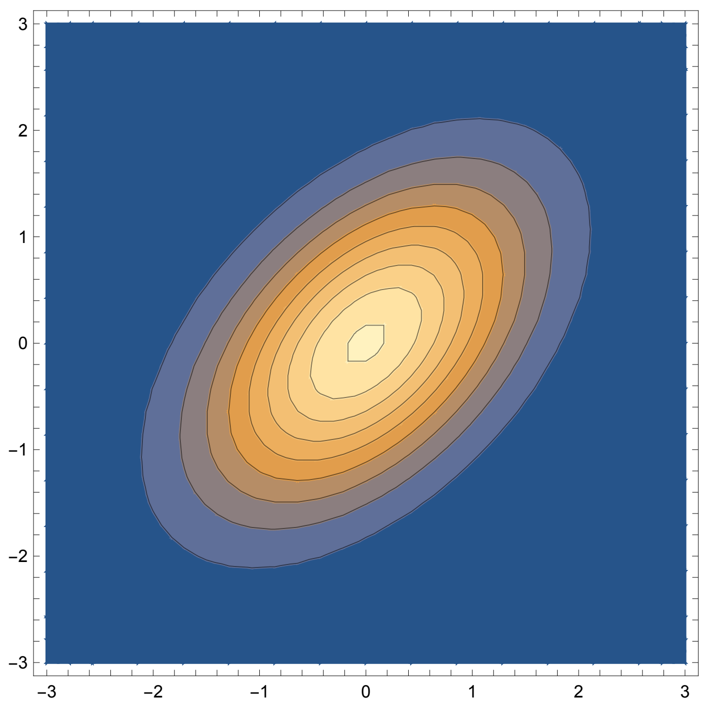
In[290]:=
```
Plot3D[PDF[MultinormalDistribution[{0, 0}, {{1, 0.5}, {0.5, 1}}], {x1, x2}],
 {x1, -3, 3}, {x2, -3, 3}]
```

Out[290]=



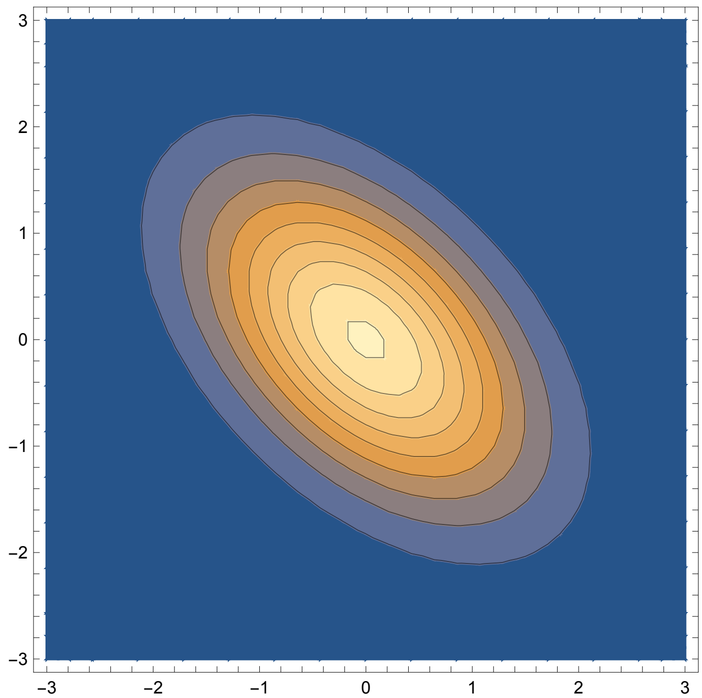
In[291]:=
```
ContourPlot[PDF[MultinormalDistribution[{0, 0}, {{1, 0.5}, {0.5, 1}}], {x1, x2}],
 {x1, -3, 3}, {x2, -3, 3}]
```

Out[291]=



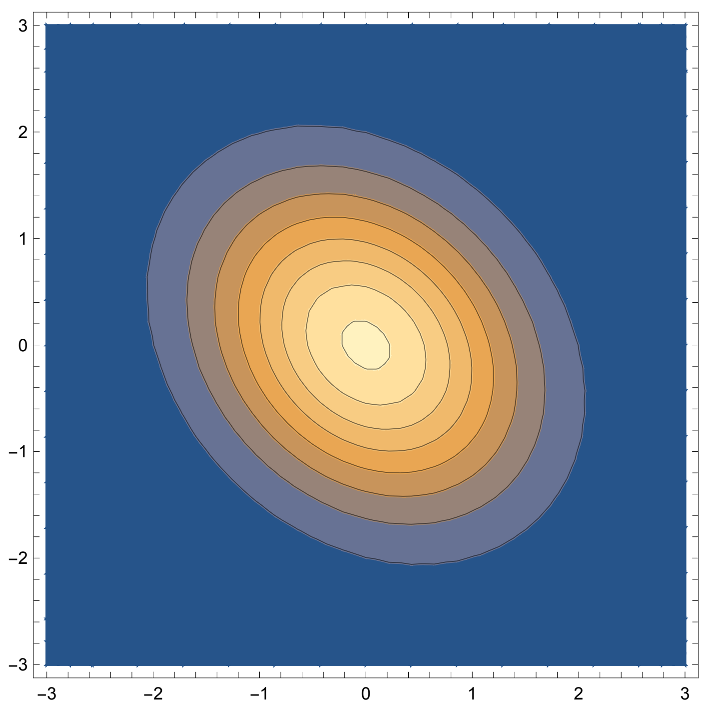
In[292]:=
```
ContourPlot[PDF[MultinormalDistribution[{0, 0}, {{1, -0.5}, {-0.5, 1}}], {x1, x2}],
 {x1, -3, 3}, {x2, -3, 3}]
```

Out[292]=



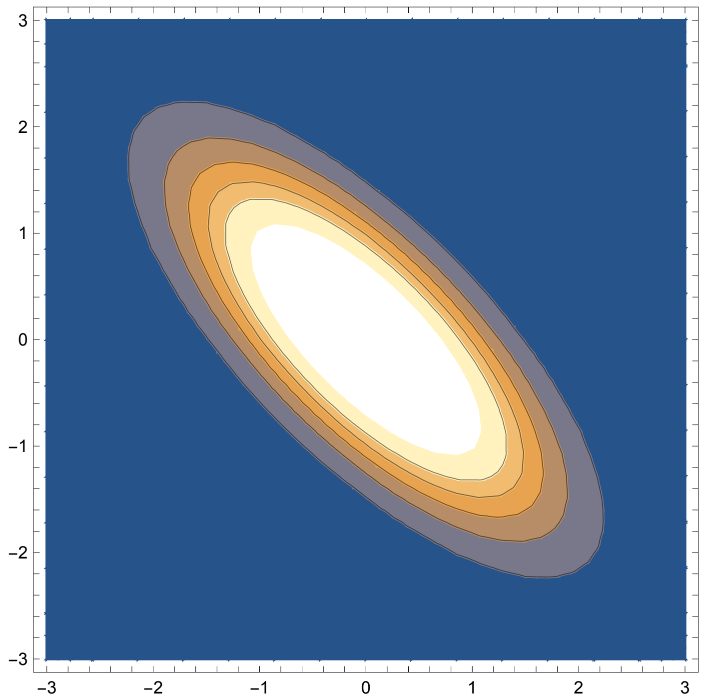
In[293]:=
```
ContourPlot[PDF[MultinormalDistribution[{0, 0}, {{1, -0.25}, {-0.25, 1}}], {x1, x2}],
 {x1, -3, 3}, {x2, -3, 3}]
```

Out[293]=



In[294]:=
```
ContourPlot[PDF[MultinormalDistribution[{0, 0}, {{1, -0.75}, {-0.75, 1}}], {x1, x2}],
 {x1, -3, 3}, {x2, -3, 3}]
```

Out[294]=



Elliptical Contours

In[295]:=
```
vnM = {0, 0};
MatrixForm[vnM]
```

Out[296]//MatrixForm=

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

In[297]:=
```
mnΣ = {{1, 0.5}, {0.5, 1}};
MatrixForm[mnΣ]
```

Out[298]//MatrixForm=

$$\begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

In[299]:=
```
Det[mnΣ]
```

Out[299]= 0.75

What we need, then, for the transformation matrix is the inverse of a "square root" of the inverse covariance matrix.

In[300]:=
```
? CholeskyDecomposition
```

Out[300]=

Symbol

CholeskyDecomposition[m] gives the Cholesky decomposition of a matrix m.

In[301]:=
```
mnP = Inverse[CholeskyDecomposition[Inverse[mnΣ]]];
MatrixForm[mnP]
```

Out[302]//MatrixForm=

$$\begin{pmatrix} 0.866025 & 0.5 \\ 0. & 1. \end{pmatrix}$$

In[303]:=
```
SingularValueDecomposition[mnΣ]
```

Out[303]=
```
{{{-0.707107, -0.707107}, {-0.707107, 0.707107}},
 {{1.5, 0.}, {0., 0.5}}, {{-0.707107, -0.707107}, {-0.707107, 0.707107}}}
```

In[304]:=
```
svd = SingularValueDecomposition[Inverse[mnΣ]]
```

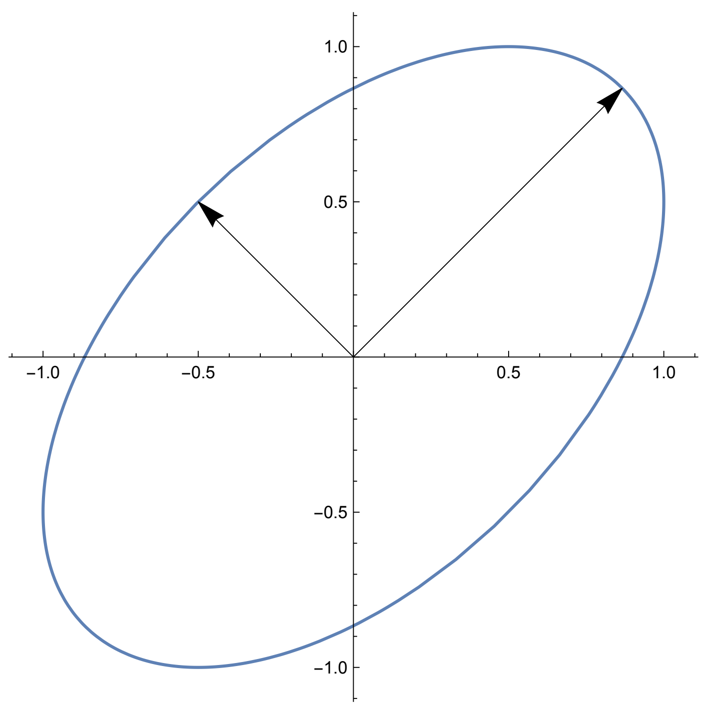
Out[304]=
```
{{{-0.707107, 0.707107}, {0.707107, 0.707107}},
 {{2., 0.}, {0., 0.666667}}, {{-0.707107, 0.707107}, {0.707107, 0.707107}}}
```

In[305]:=
```
√svd[[2, 2, 2]] svd[[1, 2]]
```

Out[305]=
```
{0.57735, 0.57735}
```

In[306]:=
```
Show[
 ParametricPlot[mnP.{Cos[θ], Sin[θ]}, {θ, -π, π}],
 Graphics[Arrow[{{0, 0}, svd[[1, All, 1]] / √svd[[2, 1, 1]] }]],
 Graphics[Arrow[{{0, 0}, svd[[1, All, 2]] / √svd[[2, 2, 2]] }]]
]
```

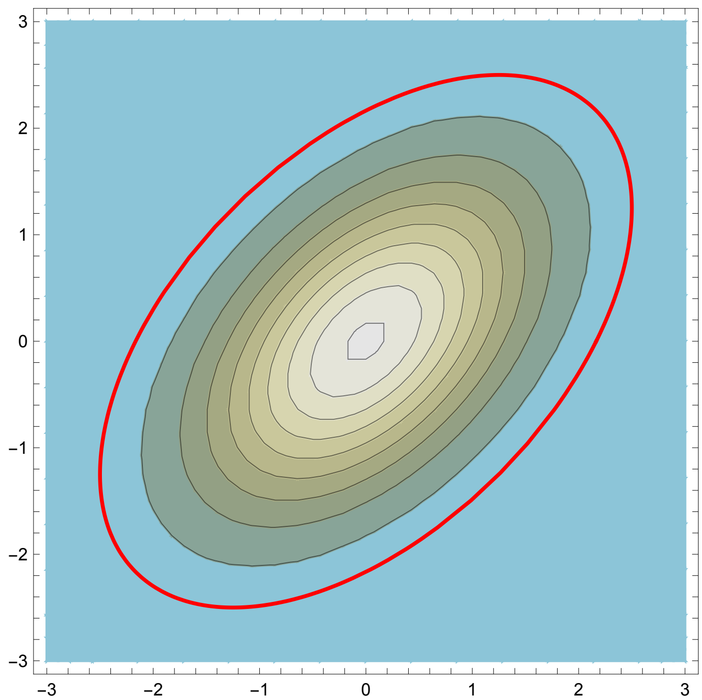
Out[306]=



In[307]:=
```
√Det[Inverse[mnΣ]]
1 / Det[mnP]
```
Out[307]= 1.1547

Out[308]= 1.1547

In[309]:=
```
Show[
 ContourPlot[PDF[MultinormalDistribution[vnM, mnΣ], {x1, x2}],
  {x1, -3, 3}, {x2, -3, 3}, ColorFunction → ColorData["LightTerrain"]],
 ParametricPlot[2.5 mnP.{Cos[θ], Sin[θ]}, {θ, 0, 2 π}, PlotStyle → {Thick, Red}]
]
```
Out[309]=



CDF

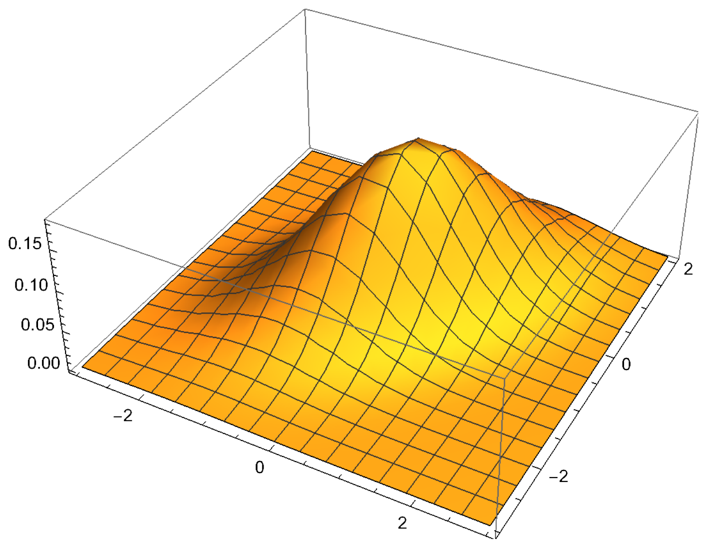
$$F_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n) \equiv P\{X_1 \le x_1 \wedge \ldots \wedge X_n \le x_n\} = \int_{-\infty}^{x_1} \ldots \int_{-\infty}^{x_n} f_{X_1,\ldots,X_n}(u_1,\ \ldots,\ u_n)\, du_1 \ldots du_n$$

$$F_{X_1,\ldots,X_n}(-\infty,\ \ldots,\ -\infty) = 0 \quad \wedge \quad F_{X_1,\ldots,X_n}(\infty,\ \ldots,\ \infty) = 1$$

Example - Bivariate Multivariate Normal Distribution

In[310]:=
```
Plot3D[PDF[MultinormalDistribution[vnM, mnΣ], {x1, x2}], {x1, -3, 3}, {x2, -3, 2}]
```

Out[310]=



In[311]:=
```
CDF[MultinormalDistribution[vnM, mnΣ], {1, 2}]
```

Out[311]= 0.831861

Characteristic Function

$$f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n) \geq 0 \quad \wedge \quad \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n)\, dx_1 \ldots dx_n = 1$$

$$\phi_{X_1,\ldots,X_n}(\omega_1,\ \ldots,\ \omega_n) \equiv \mathrm{E}\left\{e^{i\,\omega\,X}\right\}$$

Example - Standard Bivariate Normal Distribution

In[312]:=
```
CharacteristicFunction[MultinormalDistribution[{0, 0}, IdentityMatrix[2]], {ω1, ω2}]
```

Out[312]= $e^{\frac{1}{2}\left(-\omega 1^2-\omega 2^2\right)}$

Marginal Distributions

$$f_{X_1}(x_1) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} f_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n)\, dx_2 \ldots dx_n$$

Example - Standard Bivariate Normal

In[313]:=
```
PDF[MultinormalDistribution[{0, 0}, IdentityMatrix[2]], {x1, x2}]
```

Out[313]= $\frac{e^{\frac{1}{2}\left(-x1^2-x2^2\right)}}{2\pi}$

Note that the margin distributions are standard Normal distributions.

In[314]:= $\int_{-\infty}^{\infty}$ `PDF[MultinormalDistribution[{0, 0}, IdentityMatrix[2]], {x1, x2}]` $dx2$

Out[314]= $\frac{e^{-\frac{x1^2}{2}}}{\sqrt{2\pi}}$

In[315]:=
```
PDF[NormalDistribution[0, 1], x1]
```

Out[315]= $\frac{e^{-\frac{x1^2}{2}}}{\sqrt{2\pi}}$

Copula

Consider a multivariate distribution $F_{X_1,\ldots,X_n}(x_1, \ldots, x_n)$ with marginal distributions $F_{X_1}(x_1), \ldots, F_{X_n}(x_n)$. We wish to define a function C, called its *copula* that maps the marginals into the multivariate as follows

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = C\left(F_{X_1}(x_1), \ldots, F_{X_n}(x_n)\right)$$

Assuming that the marginals satisfy certain simple conditions, the natural way to construct C is

$$C(u_1, \ldots, u_n) \equiv F_{X_1,\ldots,X_n}\left(F_{X_1}{}^{-1}(u_1), \ldots, F_{X_n}{}^{-1}(u_n)\right) = F_{X_1,\ldots,X_n}(x_1, \ldots, x_n)$$

Note that the domain of C is the unit hypercube of dimension n.

$$u_i = F_{X_i}(x_i) \Rightarrow U_i \sim U[0, 1] \wedge x_i = F_{X_i}{}^{-1}(u_i) \Rightarrow X_i \sim F_{X_i}$$

Using the copula C we can form a new distribution $G_{Y_1,\ldots,Y_n}$ using marginals $G_{Y_1}, \ldots, G_{Y_n}$ as follows

$$G_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n) = C\left(G_{Y_1}(y_1), \ldots, G_{Y_n}(y_n)\right)$$

Thus, the multivariate distribution $G_{Y_1,\ldots,Y_n}(y_1, \ldots, y_n)$ will have the margin distributions $G_{Y_1}(y_1), \ldots, G_{Y_n}(y_n)$ and the same dependency structure as $F_{X_1,\ldots,X_n}(x_1, \ldots, x_n)$. Note that the marginals $G_{Y_i}(y_i)$ do not all have to be the same type of distribution nor do they need to be related to the $F_{X_i}(x_i)$.

Example - Archimedean Copula

$$C(u_1, u_2) = \psi^{-1}(\psi(u_1) + \psi(u_2))$$

$$F_{X_1,X_2}(x_1, x_2) = C\left(F_{X_1}(x_1), F_{X_n}(x_2)\right) = \psi^{-1}\left(\psi(F_{X_1}(x_1)) + \psi(F_{X_2}(x_2))\right)$$

The generator ψ must be continuous strictly decreasing convex funcion $[0, 1] \to [0, \infty]$ with $\psi(1) = 0$ and $\lim_{x\to 0}\psi(x) = \infty$

Consider the generator $\phi(u) = \frac{1}{x^2} - 1$.

In[316]:=
```
ψ[x_] := 1/x^2 - 1;
```

The inverse is

In[317]:=
```
invψ[x_] := (1 + x)^(-1/2);
```

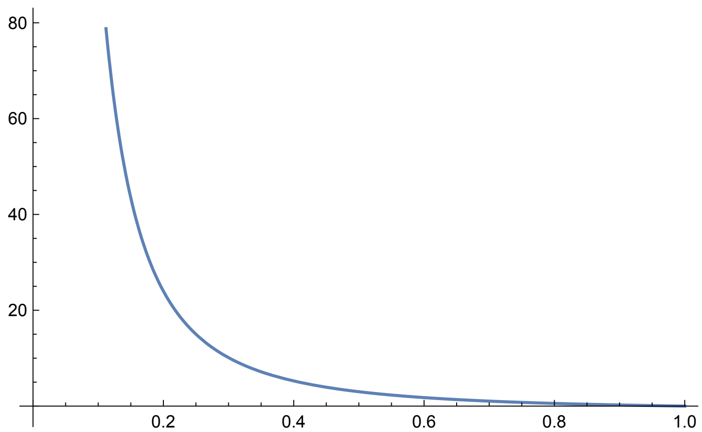
which checks out using the appropriate assumptions on the domain.

In[318]:=
```
FullSimplify[invψ[ψ[x]], Assumptions → x ∈ Reals && 0 ≤ x ≤ 1]
```

Out[318]= x

The function ψ is strictly decreasing and convex.

In[319]:=
```
Plot[ψ[x], {x, 0, 1}]
```

Out[319]=


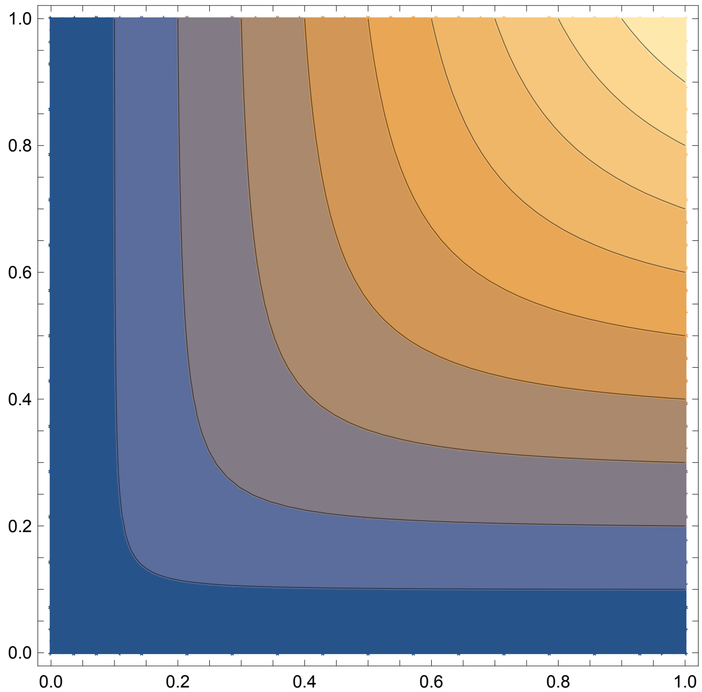
Consider the uniform distribution on $[0, 1]$: $F_X(x) = x$ for $0 \le x \le 1$. We can apply the copula with uniform marginals by

In[320]:=
```
invψ[ψ[x1] + ψ[x2]]
```

Out[320]=
$$\frac{1}{\sqrt{-1 + \frac{1}{x1^2} + \frac{1}{x2^2}}}$$

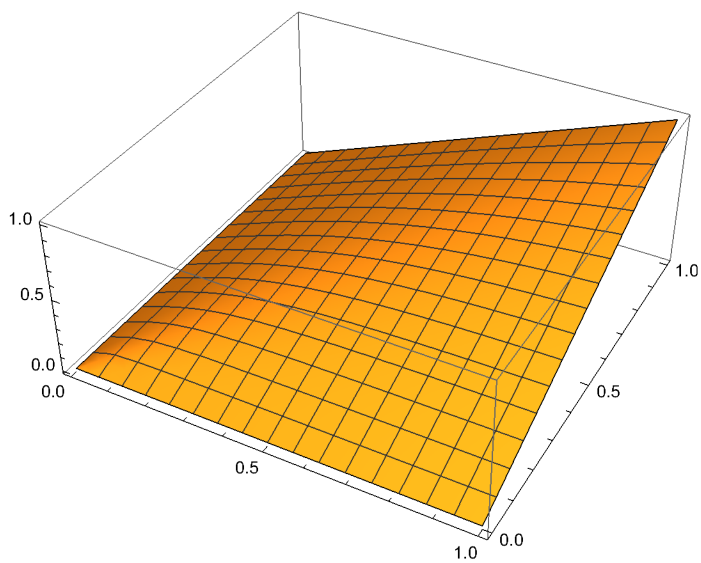
In[321]:= `ContourPlot[invψ[ψ[x1] + ψ[x2]], {x1, 10^-9, 1}, {x2, 10^-9, 1}]`

Out[321]=



In[322]:= `Plot3D[invψ[ψ[x1] + ψ[x2]], {x1, 10^-9, 1}, {x2, 10^-9, 1}]`

Out[322]=



We can recover the PDF by differentiating.

In[323]:= `p = D[invψ[ψ[x1] + ψ[x2]], x1, x2]`

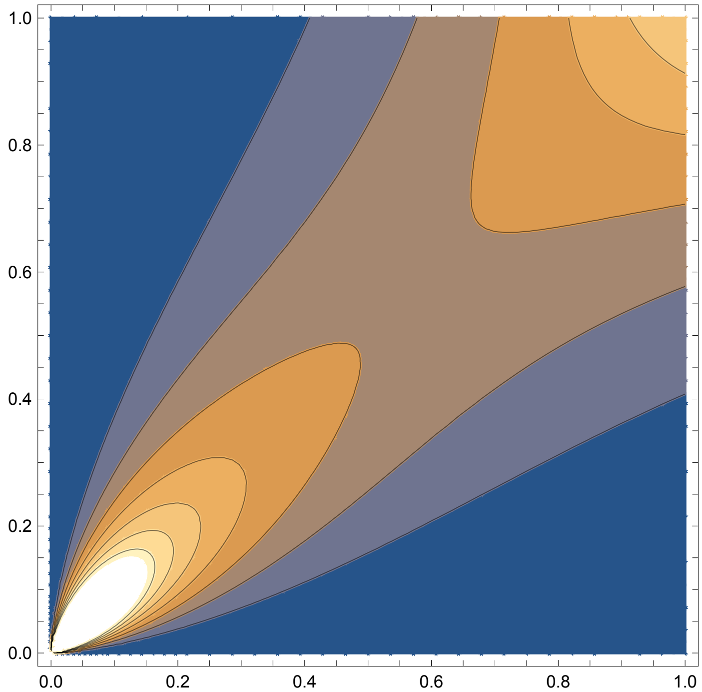
Out[323]=

$$\frac{3}{x1^3 \left(-1 + \frac{1}{x1^2} + \frac{1}{x2^2}\right)^{5/2} x2^3}$$

If we plot the pdf we can see that x_1 and x_2 are related but in a non-linear way. In particular, note the behavior near zero. This copula, one of the general class of Archimedean copulas called Clayton copulas, is used when we need a general "background" correlation that increases dramatically for "lower tail" values. Thus, it is useful in financial markets where the “conventional wisdom” is that "all correlations tend to one in a crash".

In[324]:=
```
ContourPlot[p, {x1, 10^-6, 1}, {x2, 10^-6, 1}]
```

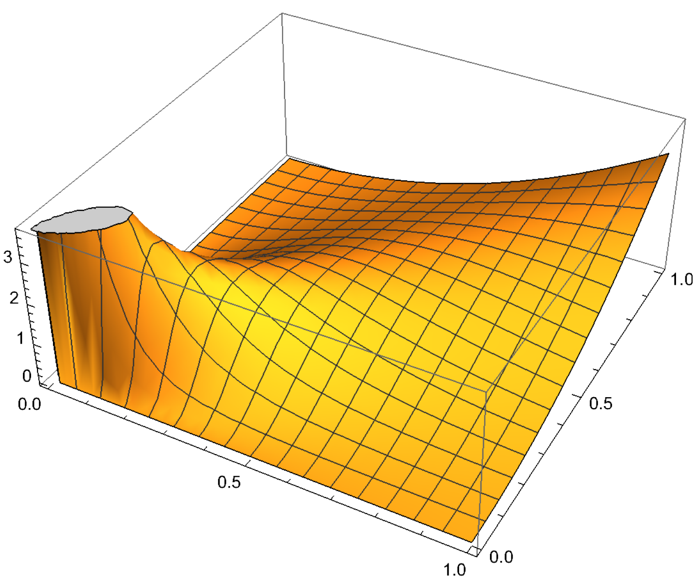
Out[324]=



Note that as $\{x_1, x_2\} \to \{0, 0\}$ the PDF $\to \infty$, so *Mathematica* attempts to plot the "interesting" parts of the function.

In[325]:=
```
Plot3D[p, {x1, 10^-6, 1}, {x2, 10^-6, 1}]
```

Out[325]=



Mixture Distributions

General Form for Finite Mixtures

Consider the mixture distribution

$$g_{X_1,\ldots,X_n}(x_1,\ \ldots,\ x_n) = \sum_{i=1}^{k} w_i\, f_i(x_1,\ \ldots,\ x_n)$$

where $0 \le w_i \le 1,\ \sum_{i=1}^{k} w_i = 1,$ and $f_1(x_1,\ \ldots,\ x_n),\ \ldots,\ f_k(x_1,\ \ldots,\ x_n)$ are valid PDFs.

Portfolio Distributions

Given a multivariate mixture distribution representing the distribution of returns, then the distribution of a portfolio's returns is a univariate mixture distribution with the same weighting of the univariate portfolio distributions associated with each of the multivariate component distributions. We will have more to say about this later in the course.

Example - Negatively skewed with increasing correlation in the lower tail.

```
In[326]:= MatrixForm@Array[If[#1 == #2, 1, 0] &, {2, 2}]
Out[326]//MatrixForm=
```

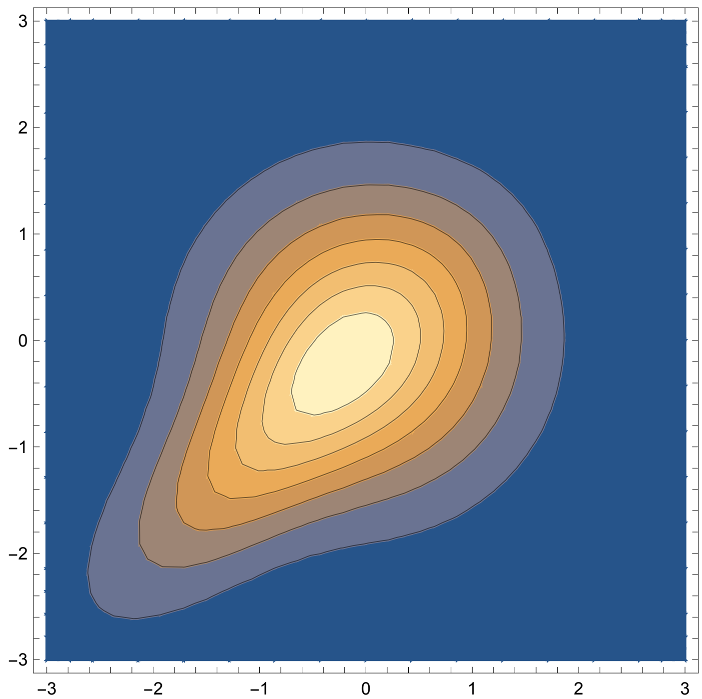
$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

```
In[327]:= MatrixForm@Array[If[#1 == #2, 1, 0.75] &, {2, 2}]
Out[327]//MatrixForm=
```

$$\begin{pmatrix} 1 & 0.75 \\ 0.75 & 1 \end{pmatrix}$$

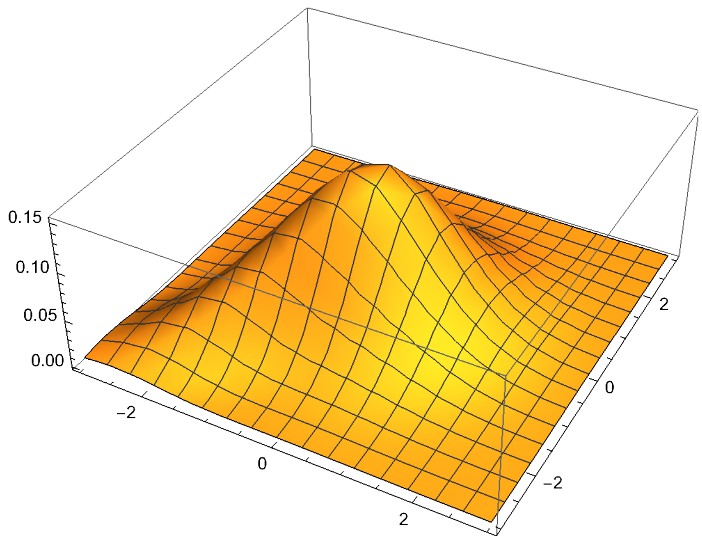
In[328]:=
```
ContourPlot[
 Evaluate[
  0.7 PDF[MultinormalDistribution[{0, 0}, Array[If[#1 == #2, 1, 0] &, {2, 2}]],
     {x1, x2}] + 0.3 PDF[MultinormalDistribution[{-1, -1},
      Array[If[#1 == #2, 1, 0.75] &, {2, 2}]], {x1, x2}]
 ],
 {x1, -3, 3}, {x2, -3, 3}
]
```

Out[328]=



In[329]:=
```
Plot3D[
 Evaluate[
  0.7 PDF[MultinormalDistribution[{0, 0}, Array[If[#1 == #2, 1, 0] &, {2, 2}]],
     {x1, x2}] + 0.3 PDF[MultinormalDistribution[{-1, -1},
      Array[If[#1 == #2, 1, 0.75] &, {2, 2}]], {x1, x2}]
 ],
 {x1, -3, 3}, {x2, -3, 3}
]
```

Out[329]=



In[330]:=

Mathematica provides special functions for dealing with mixture distributions, *e.g.*,

In[331]:=
```
? MixtureDistribution
```

Out[331]=

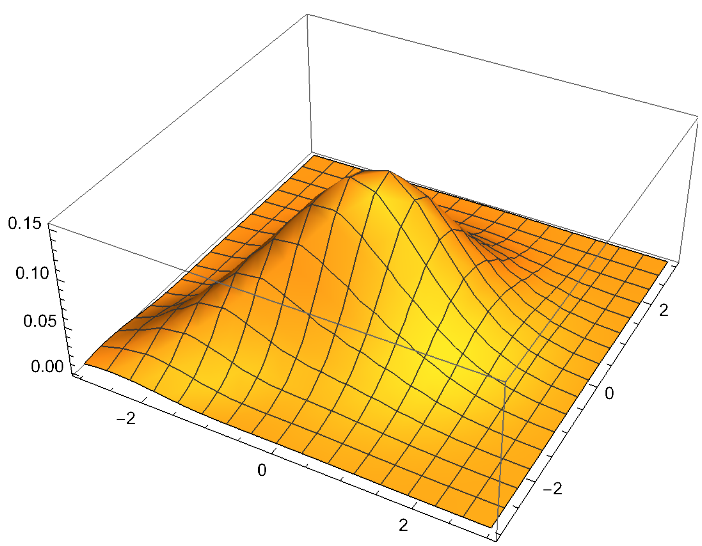
Symbol

MixtureDistribution[$\{w_1, \ldots, w_n\}$, $\{dist_1, \ldots, dist_n\}$] represents a mixture distribution whose CDF is given as a sum of the CDFs of the component distributions $dist_i$, each with weight w_i.

In[332]:=
```
Evaluate[
 PDF[
  MixtureDistribution[
   {0.7, 0.3},
   {MultinormalDistribution[{0, 0}, Array[If[#1 == #2, 1, 0] &, {2, 2}]],
    MultinormalDistribution[{-1, -1}, Array[If[#1 == #2, 1, 0.75] &, {2, 2}]]}
  ],
  {x1, x2}
 ]
]
```

Out[332]= $0.111408\, e^{\frac{1}{2}\left(-x1^2-x2^2\right)} +$
$0.0721859\, e^{\frac{1}{2}\left(-(1+x1)\,(2.28571\,(1+x1)-1.71429\,(1+x2))-(1+x2)\,(-1.71429\,(1+x1)+2.28571\,(1+x2))\right)}$

In[333]:=
```
Plot3D[
 Evaluate[
  PDF[
   MixtureDistribution[
    {0.7, 0.3},
    {MultinormalDistribution[{0, 0}, Array[If[#1 == #2, 1, 0] &, {2, 2}]],
     MultinormalDistribution[{-1, -1}, Array[If[#1 == #2, 1, 0.75] &, {2, 2}]]}
   ],
   {x1, x2}
  ]
 ],
 {x1, -3, 3}, {x2, -3, 3}
]
```

Out[333]=


In[334]:=
```
RandomVariate[
  MixtureDistribution[
   {0.7, 0.3},
   {MultinormalDistribution[{0, 0}, Array[If[#1 == #2, 1, 0] &, {2, 2}]],
    MultinormalDistribution[{-1, -1}, Array[If[#1 == #2, 1, 0.75] &, {2, 2}]]}
  ],
  {1000}
 ];
```

Elliptical Distributions

PDF

An elliptical distribution can be defined by its PDF:

$$f_{\mathbf{X}}(\mathbf{x}) = \mathcal{E}\ell[\boldsymbol{\mu}, \boldsymbol{\Sigma}, g_n] = \frac{1}{\sqrt{|\boldsymbol{\Sigma}|}}\, g_n\left[(\mathbf{x}-\boldsymbol{\mu})^T\, \boldsymbol{\Sigma}^{-1}\, (\mathbf{x}-\boldsymbol{\mu})\right]$$

Construction of g_n

Let $h : [0, \infty) \longrightarrow [0, \infty)$ be a PDF satisfying

$$\int_0^\infty x^{n/2-1}\, h[x]\, dx < \infty$$

Define a normalizing constant c_n computed from h by

$$c_n = \frac{\Gamma[n/2]}{\pi^{n/2}} \bigg/ \int_0^\infty x^{n/2-1}\, h[x]\, dx$$

Note that this is simple to derive from the fact that the volume of an n-dimensional hypersphere is

$$\frac{\pi^{n/2}}{\Gamma(n/2+1)}$$

Then $g_n[x] = c_n\, h[x]$ is a valid generator function for an elliptical distribution of dimensionality n.

Characteristic Function

An elliptical distribution can also be defined in terms of its characteristic function, where ψ is the characteristic generator of the distribution:

$$\mathbb{E}\left[e^{i\,\mathbf{t}^T\,\mathbf{X}}\right] = e^{i\,\mathbf{t}^T\,\mu}\, \psi\left(\mathbf{t}^T\, \boldsymbol{\Sigma}\, \mathbf{t}\right)$$

Mean and Covariance

If the mean of the distribution exists, it is the location parameter:

$$\mathbb{E}[\mathbf{X}] = \boldsymbol{\mu}$$

If the covariance of the distribution exists, it is *proportional* to the dispersal parameter:

$$\mathrm{Cov}[\mathbf{X}] = -\frac{\partial \psi(0)}{\partial t}\,\mathbf{\Sigma}$$

It is important to note that, unlike the Normal distribution where $\sigma_{i,j} = 0$ also implies independence, this is not true for elliptical distributions in general.

Portfolio Distributions

The utility of elliptical distributions in portfolio optimization comes from the fact that given an allocation vector $\mathbf{x} \in \mathbb{R}^n$ and returns distributed according to the elliptical distribution $f_{\mathbf{R}}(\mathbf{r}) = \mathcal{E}\ell[\boldsymbol{\mu}, \mathbf{\Sigma}, g_n]$, then the mean and variance of the univariate portfolio return distribution $\mathcal{P}_{\mathbf{x}}$ associated with $\mathbf{x}$ is also elliptically distributed:

$$\mathcal{P}_{\mathbf{x}} \sim \mathcal{E}\ell\left[\boldsymbol{\mu}^T \mathbf{x}, \mathbf{x}^T \mathbf{\Sigma}\,\mathbf{x}, g_1\right]$$

where if $g_n[x] = c_n\, h[x]$, then $g_1[x] = c_1\, h[x]$. Note that $c_1 = 1 \Big/ \left(\int_0^\infty \frac{1}{\sqrt{x}}\, h[x]\, dx\right)$.

Example - Uniform Hyperspherical Distribution

Consider the following generator with normalization constant k.

$$g(u) \;=\; k\,\mathbb{I}[u \le 1]$$

Setting $u = \mathbf{x}^T \mathbf{x}$ (which corresponds to the case $\mu = \mathbf{0}$ and $\mathbf{\Sigma} = \mathbf{I}$)

$$g\left(\mathbf{x}^T \mathbf{x}\right) \;=\; k\,\mathbb{I}[\mathbf{x}^{\mathrm{T}}\,\mathbf{x} \le 1]$$

Where $\mathbb{I}[b]$ is an indicator function which returns 1 when b is true and 0 otherwise. Thus, $\mathbf{x}$ is a unit hypersphere of dimension N. To represent a valid PDF, the volume must be normalized to unity. The volume of an N-dimensional unit hypersphere is

$$\frac{\pi^{\frac{N}{2}}}{\Gamma\left(\frac{N}{2}+1\right)}$$

hence, the appropriate normalization constant is the reciprocal.

$$g\left(\mathbf{x}^T \mathbf{x}\right) \;=\; \frac{\Gamma\left(\frac{N}{2}+1\right)}{\pi^{\frac{N}{2}}}\,\mathbb{I}[\mathbf{x}^{\mathrm{T}}\,\mathbf{x} \le 1]$$

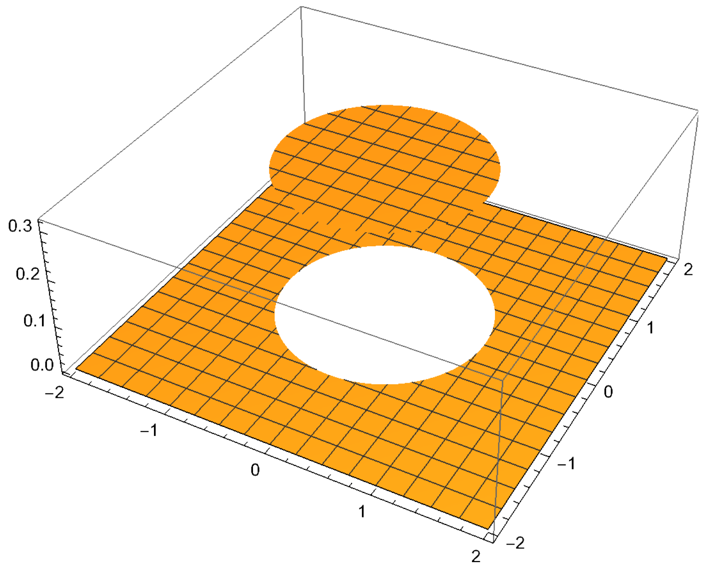
$$f_{g,\mu,\mathbf{Q}}(\mathbf{x}) \;=\; \frac{\Gamma\left(\frac{N}{2}+1\right)}{\pi^{\frac{N}{2}}\,\sqrt{\det(Q)}}\,\mathbb{I}\left[(\mathbf{x}-\boldsymbol{\mu})^T\,\mathbf{Q}^{-1}\,(\mathbf{x}-\boldsymbol{\mu}) \le 1\right]$$

Here is a plot of the 2-dimensional case.

In[335]:=

$$\texttt{Plot3D}\left[\texttt{If}\left[\# \leq 1, \frac{1}{\pi}, 0\right] \&\left[x^2 + y^2\right], \{x, -2, 2\}, \{y, -2, 2\}, \texttt{PlotPoints} \to 200\right]$$

Out[335]=



Example - μ and Q

In[336]:=

$$\texttt{vnM} = \{1, -1\};$$

$$\texttt{mnQ} = \left\{\left\{1, \frac{1}{2}\right\}, \left\{\frac{1}{2}, 1\right\}\right\};$$

```
mnIQ = Inverse[mnQ];
```

In[339]:=

$$\texttt{euhPDF}[x\_, y\_] := \texttt{Evaluate}\left[\frac{1}{\pi\sqrt{\texttt{Det}[\texttt{mnQ}]}}\texttt{If}\left[\left(\{x, y\} - \texttt{vnM}\right).\texttt{mnIQ}.\left(\{x, y\} - \texttt{vnM}\right) \leq 1, 1, 0\right]\right]$$

In[340]:=

```
? euhPDF
```

Out[340]=

Symbol

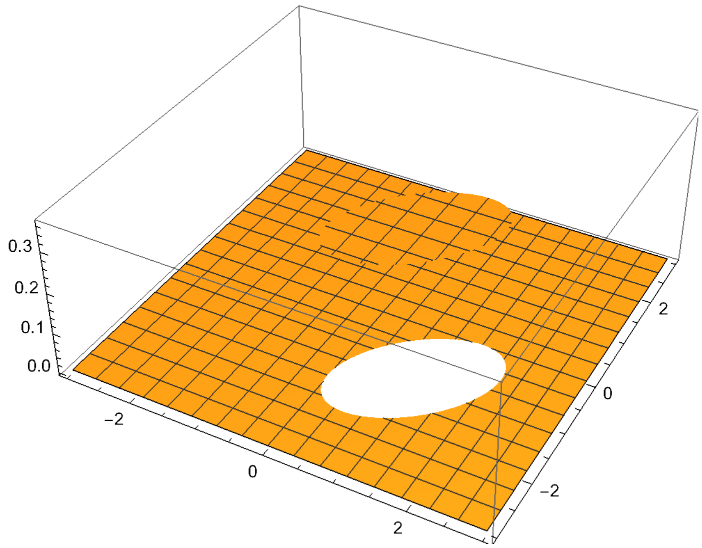
Global`euhPDF

Definitions euhPDF[x_, y_] := $\frac{2\,\mathrm{If}\left[(-1+x)\left(\frac{4}{3}(-1+x)-\frac{2(1+y)}{3}\right)+(1+y)\left(-\frac{2}{3}(-1+x)+\frac{4(1+y)}{3}\right)\leq 1,1,0\right]}{\sqrt{3}\,\pi}$

Full Name Global`euhPDF

In[341]:= `Plot3D[euhPDF[x, y], {x, -3, 3}, {y, -3, 3}, PlotPoints → 200]`

Out[341]=



Example - Marginals and Copulas

We will compare Normal and Student *t* marginals with dependency being modeled by the Gaussian and Clayton copulas.

S&P 500 Data

First, however, we want to develop some ideas about the distribution of monthly returns. To do this we will use monthly data for the S&P 500 from 1900 to 2015.

Download and Process Data

The Import[ ] assumes that the file is in the same directory as this notebook. It contains monthly index data for the S & P 500 Total Return Index from {1899, 12} to {2015, 12}. The term "total return" in this instance includes the reinvestment of dividends back into the index so that it properly represents the total wealth from investing in the index. Although this index did not exist over this period, it is a reconstruction of it by Global Financial Data.

In[342]:=
```
mxSP500Index = Import[FileNameJoin[{NotebookDirectory[], "mxSP500Index.m"}]];
Short[mxSP500Index, 7]
```

Out[343]//Short= `Null`

In[344]:= `DateListLogPlot[mxSP500Index]`

··· DateListLogPlot: Null is not a valid dataset or list of datasets.

Out[344]= `DateListLogPlot[Null]`

We can compute the log returns of the index by taking the first order differences of the log of the index. Note that we also have to drop the first date.

In[345]:=
```
mxSP500LogR =
  Transpose[{Rest[mxSP500Index⟦All, 1⟧], Differences@Log[mxSP500Index⟦All, 2⟧]}];
```

Symbol: Symbol called with 0 arguments; 1 argument is expected.

Rest: Cannot take Rest of expression Symbol[] with length zero.

Symbol: Symbol called with 0 arguments; 1 argument is expected.

Differences: List or SparseArray or structured array expected at position 1 in Differences[Log[Symbol[]]].

Transpose: The first two levels of {Rest[Symbol[]], Differences[Log[Symbol[]]]} cannot be transposed.

In[346]:=
```
DateListPlot[mxSP500LogR, Joined → True, PlotRange → All]
```

DateListPlot: Transpose[{Rest[Symbol[]], Differences[Log[Symbol[]]]}] is not a valid dataset or list of datasets.

Out[346]=
```
DateListPlot[Transpose[{Rest[Symbol[]], Differences[Log[Symbol[]]]}],
 Joined → True, PlotRange → All]
```

In[347]:=
```
Dimensions[mxSP500LogR]
```

Out[347]=
```
{1}
```

In[348]:=
```
vnSP500Moments =
 Through[{Mean, StandardDeviation, Skewness, Kurtosis}[mxSP500LogR⟦All, 2⟧]]
```

Out[348]=
```
{Mean[Transpose[Differences[Log[Symbol[]]]]],
 StandardDeviation[Transpose[Differences[Log[Symbol[]]]]],
 Skewness[Transpose[Differences[Log[Symbol[]]]]],
 Kurtosis[Transpose[Differences[Log[Symbol[]]]]]}
```

In[349]:=
```
{m, s} = vnSP500Moments⟦{1, 2}⟧
```

Out[349]=
```
{Mean[Transpose[Differences[Log[Symbol[]]]]],
 StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]}
```

Fitting a Normal Distribution

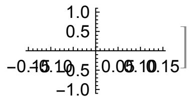
In[350]:=
```
distSP500N = EstimatedDistribution[mxSP500LogR⟦All, 2⟧, NormalDistribution[ṃ, ṣ]]
```

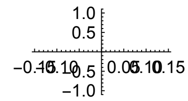
EstimatedDistribution: One or more data points are not in support of the process or distribution NormalDistribution[m, s].

Out[350]=
```
EstimatedDistribution[
 Transpose[Differences[Log[Symbol[]]]], NormalDistribution[ṃ, ṣ]]
```

In[351]:=
```
Show[
 Histogram[mxSP500LogR〚All, 2〛, Automatic, "PDF"],
 Plot[Evaluate[PDF[distSP500N, x]], {x, -0.15, 0.15}]
]
```

••• Histogram: Transpose[Differences[Log[Symbol[]]]] is not a valid dataset or list of datasets.

••• Show: Could not combine the graphics objects in Show[Histogram[Transpose[Differences[Log[Symbol[]]]], Automatic,
PDF], ].

Out[351]=
```
Show[Histogram[Transpose[Differences[Log[Symbol[]]]], Automatic, PDF], 
```
]

Fitting a Student t Distribution

In[352]:=
```
distSP500T =
 EstimatedDistribution[mxSP500LogR〚All, 2〛, StudentTDistribution[m, s, d]]
```

••• EstimatedDistribution: One or more data points are not in support of the process or distribution StudentTDistribution[m, s, d].

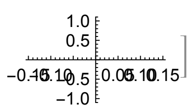
Out[352]=
```
EstimatedDistribution[
 Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]
```

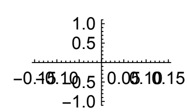
In[353]:=
```
Through[{Mean, StandardDeviation, Skewness, Kurtosis}[distSP500T]]
```

Out[353]=
```
{Mean[EstimatedDistribution[
   Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]],
 StandardDeviation[EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
   StudentTDistribution[m, s, d]]], Skewness[EstimatedDistribution[
   Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]],
 Kurtosis[EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
   StudentTDistribution[m, s, d]]]}
```

In[354]:=
```
Show[
 Histogram[mxSP500LogR〚All, 2〛, Automatic, "PDF"],
 Plot[Evaluate[PDF[distSP500T, x]], {x, -0.15, 0.15}]
]
```

••• Histogram: Transpose[Differences[Log[Symbol[]]]] is not a valid dataset or list of datasets.

••• Show: Could not combine the graphics objects in Show[Histogram[Transpose[Differences[Log[Symbol[]]]], Automatic,
PDF], ].

Out[354]=
```
Show[Histogram[Transpose[Differences[Log[Symbol[]]]], Automatic, PDF], 
```
]

Normal Marginals-Gaussian Copula

The typical “default” assumption people use is to assume that returns are multivariate Normal. For the example here assume two instruments with identical annual means and standard deviations of 10% and 15%, respectively. A multivariate Normal can, in terms of its copula, be expressed as univariate Normal marginals with a Gaussian copula.

We simulate 10,000 months of returns assume a multivariate Normal with correlation $\rho = 0.5$.

In[355]:=
```
simNG = RandomVariate[CopulaDistribution[{"Binormal", 0.5},
    {NormalDistribution[m, s], NormalDistribution[m, s]}], 10 000];
```

```
NormalDistribution: Parameter Mean[Transpose[Differences[Log[Symbol[]]]]] at position 1 in
   NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]], StandardDeviation[Transpose[Differences[Log[
         Symbol[]]]]]] is expected to be real.
```

In[356]:=
```
ListPlot[simNG, PlotStyle → PointSize[Medium], PlotLabel → "Normal-Gaussian",
 Mesh → None, ColorFunction → "Pastel", ImageSize → 300, PlotRange → All]
```

```
RandomVariate: The array dimensions 10000.`16. given in position 2 of
   RandomVariate[CopulaDistribution[{Binormal, 0.5}, {NormalDistribution[Mean[Transpose[Differences[Log[≪1≫]]]],
         StandardDeviation[Transpose[Differences[Log[≪1≫]]]]], NormalDistribution[Mean[Transpose[Differences[Log[
               ≪1≫]]]], ≪17≫[≪1≫]]}], 10000.00000000000] should be a list of
   non−negative machine−sized integers giving the dimensions for the result.

NormalDistribution: Parameter Mean[Transpose[Differences[Log[Symbol[]]]]] at position 1 in
   NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]], StandardDeviation[Transpose[Differences[Log[
         Symbol[]]]]]] is expected to be real.

RandomVariate: The array dimensions 10000.`16. given in position 2 of
   RandomVariate[CopulaDistribution[{Binormal, 0.5}, {NormalDistribution[Mean[Transpose[Differences[Log[≪1≫]]]],
         StandardDeviation[Transpose[Differences[Log[≪1≫]]]]], NormalDistribution[Mean[Transpose[Differences[Log[
               ≪1≫]]]], ≪17≫[≪1≫]]}], 10000.00000000000] should be a list of
   non−negative machine−sized integers giving the dimensions for the result.

NormalDistribution: Parameter Mean[Transpose[Differences[Log[Symbol[]]]]] at position 1 in
   NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]], StandardDeviation[Transpose[Differences[Log[
         Symbol[]]]]]] is expected to be real.

ListPlot:
 RandomVariate[CopulaDistribution[{Binormal, 0.5}, {NormalDistribution[Mean[Transpose[Differences[Log[≪1≫]]]],
         StandardDeviation[Transpose[Differences[Log[≪1≫]]]]], NormalDistribution[Mean[Transpose[Differences[Log[
               ≪1≫]]]], ≪17≫[≪1≫]]}], 10000.] is not a list of numbers or pairs of
   numbers.
```

Out[356]=
```
ListPlot[RandomVariate[CopulaDistribution[{Binormal, 0.5},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000],
 PlotStyle → PointSize[Medium], PlotLabel → Normal-Gaussian,
 Mesh → None,
 ColorFunction → Pastel,
 ImageSize → 300,
 PlotRange → All]
```

```
In[357]:= vnMeanNG = Mean@simNG;
MatrixForm@vnMeanNG
vnSdevNG = StandardDeviation@simNG;
MatrixForm@vnSdevNG
mnCorNG = Correlation[simNG];
MatrixForm@mnCorNG
```

```
Out[358]//MatrixForm=
Mean[RandomVariate[CopulaDistribution[{Binormal, 0.5},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000]]
```

```
Out[360]//MatrixForm=
StandardDeviation[RandomVariate[CopulaDistribution[{Binormal, 0.5},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000]]
```

••• Correlation: The first argument must be either a vector, a matrix, or a multivariate distribution.

```
Out[362]//MatrixForm=
Correlation[RandomVariate[CopulaDistribution[{Binormal, 0.5},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000]]
```

```
In[363]:= simPortNG = {0.5, 0.5}.# & /@ simNG;
```

••• RandomVariate: The array dimensions {0.5, 0.5}.10000 given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

In[364]:=
```
simPortNGMoments = Through[{Mean, StandardDeviation, Skewness, Kurtosis}[simPortNG]]
```

Out[364]=
```
{Mean[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
   {0.5, 0.5}.10 000]], StandardDeviation[RandomVariate[
   {0.5, 0.5}.CopulaDistribution[{Binormal, 0.5},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
   {0.5, 0.5}.10 000]], Skewness[RandomVariate[
   {0.5, 0.5}.CopulaDistribution[{Binormal, 0.5},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
   {0.5, 0.5}.10 000]], Kurtosis[RandomVariate[
   {0.5, 0.5}.CopulaDistribution[{Binormal, 0.5},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[
        Transpose[Differences[Log[Symbol[]]]]]]}], {0.5, 0.5}.10 000]]}
```

In[365]:=
```
Histogram[simPortNG, Automatic, "PDF",
 PlotLabel → "Normal-Gaussian", PlotRange → All]
```

••• Histogram: ≪1≫ is not a valid dataset or list of datasets.

••• Histogram: ≪1≫ is not a valid dataset or list of datasets.

••• Histogram: ≪1≫ is not a valid dataset or list of datasets.

••• General: Further output of Histogram::ldata will be suppressed during this calculation.

Out[365]=
```
Histogram[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5},
    {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
      StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
     NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
      StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
  {0.5, 0.5}.10 000], Automatic, PDF, PlotLabel → Normal-Gaussian,
 PlotRange → All]
```

Normal Marginals-Clayton Copula

In[366]:=
```
simNC = RandomVariate[CopulaDistribution[{"Clayton", 1},
   {NormalDistribution[m, s], NormalDistribution[m, s]}], 10 000];
```

... NormalDistribution: Parameter Mean[Transpose[Differences[Log[Symbol[]]]]] at position 1 in NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]], StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]] is expected to be real.

In[367]:=
```
ListPlot[simNC, PlotStyle → PointSize[Medium], PlotLabel → "Normal-Clayton",
 Mesh → None, ColorFunction → "Pastel", ImageSize → 300, PlotRange → All]
```

... RandomVariate: The array dimensions 10000.`16. given in position 2 of RandomVariate[CopulaDistribution[{Clayton, 1.000000000000000}, {NormalDistribution[Mean[Transpose[Differences[Log[≪1≫]]]], StandardDeviation[Transpose[Differences[Log[≪1≫]]]]], NormalDistribution[Mean[Transpose[Differences[Log[≪1≫]]]], ≪17≫[≪1≫]]}], ≪10≫] should be a list of non−negative machine−sized integers giving the dimensions for the result.

... NormalDistribution: Parameter Mean[Transpose[Differences[Log[Symbol[]]]]] at position 1 in NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]], StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]] is expected to be real.

... NormalDistribution: Parameter Mean[Transpose[Differences[Log[Symbol[]]]]] at position 1 in NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]], StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]] is expected to be real.

... ListPlot: RandomVariate[CopulaDistribution[{Clayton, 1.}, {NormalDistribution[Mean[Transpose[Differences[Log[≪1≫]]]], StandardDeviation[Transpose[Differences[Log[≪1≫]]]]], NormalDistribution[Mean[Transpose[Differences[Log[≪1≫]]]], ≪17≫[≪9≫[≪1≫]]]}], 10000.] is not a list of numbers or pairs of numbers.

Out[367]=
```
ListPlot[RandomVariate[CopulaDistribution[{Clayton, 1},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000],
 PlotStyle → PointSize[Medium], PlotLabel → Normal-Clayton,
 Mesh → None,
 ColorFunction → Pastel,
 ImageSize → 300,
 PlotRange → All]
```

In[368]:=
```
vnMeanNC = Mean@simNC;
MatrixForm@vnMeanNC
vnSdevNC = StandardDeviation@simNC;
MatrixForm@vnSdevNC
mnCorNC = Correlation[simNC];
MatrixForm@mnCorNC
```

Out[369]//MatrixForm=
```
Mean[RandomVariate[CopulaDistribution[{Clayton, 1},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000]]
```

Out[371]//MatrixForm=
```
StandardDeviation[RandomVariate[CopulaDistribution[{Clayton, 1},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000]]
```

Correlation: The first argument must be either a vector, a matrix, or a multivariate distribution.

Out[373]//MatrixForm=
```
Correlation[RandomVariate[CopulaDistribution[{Clayton, 1},
   {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
    NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
     StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}], 10 000]]
```

In[374]:=
```
simPortNC = {0.5, 0.5}.# & /@ simNC;
```

RandomVariate: The array dimensions {0.5, 0.5}.10000 given in position 2 of RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[Transpose[Differences[≪1≫]]], StandardDeviation[Transpose[Differences[≪1≫]]]], NormalDistribution[Mean[Transpose[Differences[≪1≫]]], ≪17≫[≪1≫]]}], {0.5, 0.5}. ≪5≫] should be a list of non−negative machine−sized integers giving the dimensions for the result.

In[375]:=
```
simPortNCMoments = Through[{Mean, StandardDeviation, Skewness, Kurtosis}[simPortNC]]
```

Out[375]=
```
{Mean[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
   {0.5, 0.5}.10000]], StandardDeviation[RandomVariate[
   {0.5, 0.5}.CopulaDistribution[{Clayton, 1},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
   {0.5, 0.5}.10000]], Skewness[RandomVariate[
   {0.5, 0.5}.CopulaDistribution[{Clayton, 1},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
   {0.5, 0.5}.10000]], Kurtosis[RandomVariate[
   {0.5, 0.5}.CopulaDistribution[{Clayton, 1},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[
        Transpose[Differences[Log[Symbol[]]]]]]}], {0.5, 0.5}.10000]]}
```

In[376]:=
```
Histogram[simPortNC, Automatic, "PDF", PlotLabel → "Normal-Clayton", PlotRange → All]
```

··· Histogram:
RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[Transpose[Differences[≪1≫]]], StandardDeviation[Transpose[Differences[≪1≫]]]], NormalDistribution[Mean[Transpose[Differences[≪1≫]]], ≪17≫[≪1≫]]}], {0.5, 0.5}. ≪5≫] is not a valid dataset or list of datasets.

··· Histogram:
RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[Transpose[Differences[≪1≫]]], StandardDeviation[Transpose[Differences[≪1≫]]]], NormalDistribution[Mean[Transpose[Differences[≪1≫]]], ≪17≫[≪1≫]]}], {0.5, 0.5}. ≪5≫] is not a valid dataset or list of datasets.

··· Histogram:
RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[Transpose[Differences[≪1≫]]], StandardDeviation[Transpose[Differences[≪1≫]]]], NormalDistribution[Mean[Transpose[Differences[≪1≫]]], ≪17≫[≪1≫]]}], {0.5, 0.5}. ≪5≫] is not a valid dataset or list of datasets.

··· General: Further output of Histogram::ldata will be suppressed during this calculation.

Out[376]=
```
Histogram[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1},
    {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
      StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
     NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
      StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
   {0.5, 0.5}.10 000], Automatic, PDF, PlotLabel → Normal-Clayton,
  PlotRange → All]
```

Student *t* Marginals-Clayton Copula

In[377]:=
```
simTC = RandomVariate[
   CopulaDistribution[{"Clayton", 1}, {distSP500T, distSP500T}], 10 000];
```

In[378]:=
```
ListPlot[simTC, PlotStyle → PointSize[Medium], PlotLabel → "t-Clayton",
  Mesh → None, ColorFunction → "Pastel", ImageSize → 300, PlotRange → All]
```

··· RandomVariate: The array dimensions 10000.`16. given in position 2 of
RandomVariate[CopulaDistribution[{Clayton, 1.000000000000000}, {EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]}], 10000.00000000000] should be a list of non−negative machine−sized integers giving the dimensions for the result.

··· ListPlot: RandomVariate[CopulaDistribution[{Clayton, 1.}, {EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]], StudentTDistribution [m, s, d]]}], 10000.] is not a list of numbers or pairs of numbers.

Out[378]=
```
ListPlot[RandomVariate[CopulaDistribution[{Clayton, 1},
    {EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
      StudentTDistribution[m, s, d]], EstimatedDistribution[
      Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]}],
   10 000], PlotStyle → PointSize[Medium], PlotLabel → t-Clayton,
  Mesh → None, ColorFunction → Pastel,
  ImageSize → 300,
  PlotRange → All]
```

```
In[379]:= vnMeanTC = Mean@simTC;
          MatrixForm@vnMeanTC
          vnSdevTC = StandardDeviation@simTC;
          MatrixForm@vnSdevTC
          mnCorTC = Correlation[simTC];
          MatrixForm@mnCorTC
Out[380]//MatrixForm=
          Mean[RandomVariate[CopulaDistribution[{Clayton, 1},
             {EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
               StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[
                Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]}], 10 000]]
Out[382]//MatrixForm=
          StandardDeviation[
           RandomVariate[CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[
               Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]],
              EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
               StudentTDistribution[m, s, d]]}], 10 000]]

          ••• Correlation: The first argument must be either a vector, a matrix, or a multivariate distribution.
Out[384]//MatrixForm=
          Correlation[RandomVariate[CopulaDistribution[{Clayton, 1},
             {EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
               StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[
                Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]}], 10 000]]

In[385]:= simPortTC = {0.5, 0.5}.# & /@ simTC;

          ••• RandomVariate: The array dimensions {0.5, 0.5}.10000 given in position 2 of
              RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[Transpose[Differences[Log[≪1≫]]],
                    StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[
                      m, s, d]]}], {0.5, ≪4≫}.≪5≫] should be a list of non−negative machine−sized
              integers giving the dimensions for the result.
```

In[386]:=
```
simPortTCMoments = Through[{Mean, StandardDeviation, Skewness, Kurtosis}[simPortTC]]
```

Out[386]=
```
{Mean[
  RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[
       Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]],
      EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
       StudentTDistribution[m, s, d]]}], {0.5, 0.5}.10 000]],
 StandardDeviation[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1},
     {EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
       StudentTDistribution[m, s, d]], EstimatedDistribution[
       Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]}],
   {0.5, 0.5}.10 000]], Skewness[RandomVariate[
   {0.5, 0.5}.CopulaDistribution[{Clayton, 1},
     {EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
       StudentTDistribution[m, s, d]], EstimatedDistribution[
       Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]]}],
   {0.5, 0.5}.10 000]], Kurtosis[RandomVariate[{0.5, 0.5}.
    CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[
       Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]],
      EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
       StudentTDistribution[m, s, d]]}], {0.5, 0.5}.10 000]]}
```

In[387]:=
```
Histogram[simPortTC, Automatic, "PDF", PlotLabel → "t-Clayton", PlotRange → All]
```

Histogram: RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[m, s, d]]}], {0.5, ≪4≫}.≪5≫] is not a valid dataset or list of datasets.

Histogram: RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[m, s, d]]}], {0.5, ≪4≫}.≪5≫] is not a valid dataset or list of datasets.

Histogram: RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[m, s, d]]}], {0.5, ≪4≫}.≪5≫] is not a valid dataset or list of datasets.

General: Further output of Histogram::ldata will be suppressed during this calculation.

Out[387]=
```
Histogram[
 RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[
      Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]],
     EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
      StudentTDistribution[m, s, d]]}], {0.5, 0.5}.10 000],
 Automatic, PDF, PlotLabel → t-Clayton, PlotRange → All]
```

Comparisons

In[388]:=
```
TableForm[{{vnMeanNG, vnSdevNG, mnCorNG, simPortNGMoments},
   {vnMeanNC, vnSdevNC, mnCorNC, simPortNCMoments},
   {vnMeanTC, vnSdevTC, mnCorTC, simPortTCMoments}},
  TableHeadings → {{"NG", "NC", "TC"}, {"Means", "Sdevs", "Cov", "Portfolio Mmts"}}]
```

Out[388]//TableForm=

```
NG    Mean[RandomVariate[CopulaDistribution[{Binormal, 0.5}, {NormalDistribution[Mean

NC      Mean[RandomVariate[CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[

TC                                                     Mean[RandomVariate[CopulaDistribution[{Clayto
```

In[389]:=
```
distNG = SmoothKernelDistribution[simPortNG];
distNC = SmoothKernelDistribution[simPortNC];
distTC = SmoothKernelDistribution[simPortTC];
```

••• SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

••• SmoothKernelDistribution: The input data SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[Transpose[≪1≫]]], NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[≪1≫]]}], {0.5, 0.5}. ≪5≫]] should be a vector or a matrix of real numbers or a valid TemporalData object.

••• SmoothKernelDistribution: The input data SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[Transpose[Differences[≪1≫]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[≪1≫]], ≪20≫[m, s, d]]}], {0.5, ≪4≫}.≪5≫]] should be a vector or a matrix of real numbers or a valid TemporalData object.

In[392]:=
```
Plot[{CDF[distNG, x], CDF[distNC, x], CDF[distTC, x]}, {x, -.2, -0.1},
 PlotLabel → "Lower Tail Comparison", PlotLegends → {"NG", "NC", "TC"}]
```

RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

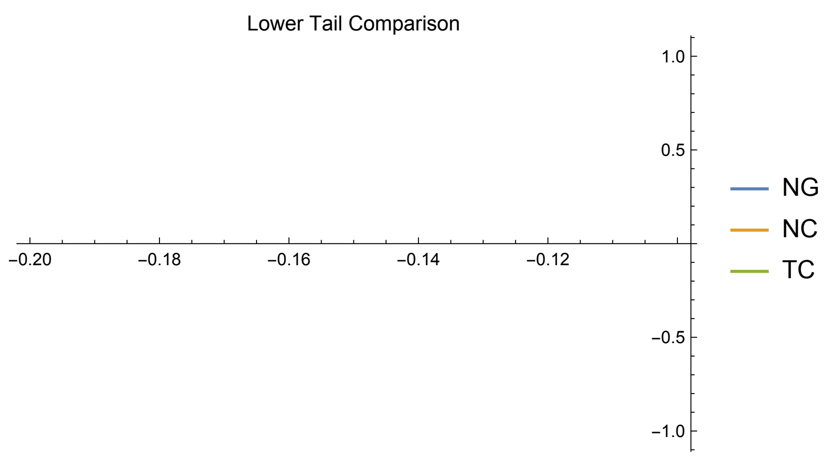
RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

General: Further output of RandomVariate::array will be suppressed during this calculation.

SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

General: Further output of SmoothKernelDistribution::invldd will be suppressed during this calculation.

Out[392]=



In[393]:=
```
nVaRNG = InverseCDF[distNG, 0.01]
nVaRNC = InverseCDF[distNC, 0.01]
nVaRTC = InverseCDF[distTC, 0.01]
```

Out[393]=
```
InverseCDF[SmoothKernelDistribution[
  RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
    {0.5, 0.5}.10000]], 0.01]
```

Out[394]=
```
InverseCDF[
 SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1},
     {NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]],
      NormalDistribution[Mean[Transpose[Differences[Log[Symbol[]]]]],
       StandardDeviation[Transpose[Differences[Log[Symbol[]]]]]]}],
    {0.5, 0.5}.10000]], 0.01]
```

Out[395]=
```
InverseCDF[SmoothKernelDistribution[
  RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[
       Transpose[Differences[Log[Symbol[]]]], StudentTDistribution[m, s, d]],
      EstimatedDistribution[Transpose[Differences[Log[Symbol[]]]],
       StudentTDistribution[m, s, d]]}], {0.5, 0.5}.10000]], 0.01]
```

In[396]:=

$$nCVaRNG = \frac{1}{0.01} \text{NIntegrate}[x\ \text{PDF}[distNG, x], \{x, -\infty, nVaRNG\}]$$

$$nCVaRNC = \frac{1}{0.01} \text{NIntegrate}[x\ \text{PDF}[distNC, x], \{x, -\infty, nVaRNC\}]$$

$$nCVaRTC = \frac{1}{0.01} \text{NIntegrate}[x\ \text{PDF}[distTC, x], \{x, -\infty, nVaRTC\}]$$

••• RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

••• SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

••• RandomVariate: The array dimensions {0.5, 0.5}.10000 given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

••• SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

••• RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

••• General: Further output of RandomVariate::array will be suppressed during this calculation.

••• SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

••• General: Further output of SmoothKernelDistribution::invldd will be suppressed during this calculation.

... NIntegrate: The integrand
x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5}, {NormalDistribution[≪2≫], NormalDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to non−numerical values for all sampling points in the region with boundaries {{−∞, 0. + InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]], 0.01]}}.

... NIntegrate: The integrand
x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5}, {NormalDistribution[≪2≫], NormalDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to non−numerical values for all sampling points in the region with boundaries {{−∞, 0. + InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]], 0.01]}}.

Out[396]= 100. NIntegrate[x PDF[distNG, x], {x, -∞, nVaRNG}]

... RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {NormalDistribution[Mean[Transpose[Differences[≪1≫]]], StandardDeviation[Transpose[Differences[≪1≫]]]], NormalDistribution[Mean[Transpose[Differences[≪1≫]]], ≪17≫[≪9≫[≪1≫]]]}], {≪1≫}.≪7≫] should be a list of non−negative machine−sized integers giving the dimensions for the result.

... SmoothKernelDistribution: The input data SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[Transpose[≪1≫]]], NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[≪9≫[≪1≫]]]}], {≪1≫}.≪7≫]] should be a vector or a matrix of real numbers or a valid TemporalData object.

... RandomVariate: The array dimensions {0.5, 0.5}.10000 given in position 2 of RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[Transpose[Differences[≪1≫]]], StandardDeviation[Transpose[Differences[≪1≫]]]], NormalDistribution[Mean[Transpose[Differences[≪1≫]]], ≪17≫[≪1≫]]}], {0.5, 0.5}. ≪5≫] should be a list of non−negative machine−sized integers giving the dimensions for the result.

... SmoothKernelDistribution: The input data SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[Transpose[≪1≫]]], NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[≪1≫]]}], {0.5, 0.5}. ≪5≫]] should be a vector or a matrix of real numbers or a valid TemporalData object.

... RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {NormalDistribution[Mean[Transpose[Differences[≪1≫]]], StandardDeviation[Transpose[Differences[≪1≫]]]], NormalDistribution[Mean[Transpose[Differences[≪1≫]]], ≪17≫[≪9≫[≪1≫]]]}], {≪1≫}.≪7≫] should be a list of non−negative machine−sized integers giving the dimensions for the result.

... General: Further output of RandomVariate::array will be suppressed during this calculation.

... SmoothKernelDistribution: The input data SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[Transpose[≪1≫]]], NormalDistribution[Mean[Transpose[≪1≫]], StandardDeviation[≪9≫[≪1≫]]]}], {≪1≫}.≪7≫]] should be a vector or a matrix of real numbers or a valid TemporalData object.

... General: Further output of SmoothKernelDistribution::invldd will be suppressed during this calculation.

... NIntegrate: The integrand
x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[≪2≫], NormalDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to non−numerical values for all sampling points in the region with boundaries {{−∞, 0. + InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]], 0.01]}}.

NIntegrate: The integrand
x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[≪2≫],
NormalDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to non−numerical
values for all sampling points in the region with boundaries {{−∞, 0.
+InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]],
0.01]}}.

Out[397]= 100. NIntegrate[x PDF[distNC, x], {x, -∞, nVaRNC}]

RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of
RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {EstimatedDistribution[Transpose[Differences[Log[≪1≫]]],
StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[
m, s, d]]}], ≪1≫.≪7≫] should be a list of non−negative machine−sized integers
giving the dimensions for the result.

SmoothKernelDistribution: The input data
SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {EstimatedDistribution[Transpose[
Differences[≪1≫]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[≪1≫]],
≪20≫[m, s, d]]}], ≪1≫.≪7≫]] should be a vector or a matrix of real numbers or a
valid TemporalData object.

RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of
RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {EstimatedDistribution[Transpose[Differences[Log[≪1≫]]],
StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[
m, s, d]]}], ≪1≫.≪7≫] should be a list of non−negative machine−sized integers
giving the dimensions for the result.

SmoothKernelDistribution: The input data
SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {EstimatedDistribution[Transpose[
Differences[≪1≫]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[≪1≫]],
≪20≫[m, s, d]]}], ≪1≫.≪7≫]] should be a vector or a matrix of real numbers or a
valid TemporalData object.

NIntegrate: The integrand
x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[≪2≫],
EstimatedDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to
non−numerical values for all sampling points in the region with boundaries {{−∞, 0.
+InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]],
0.01]}}.

RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of
RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {EstimatedDistribution[Transpose[Differences[Log[≪1≫]]],
StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[Log[≪1≫]]], StudentTDistribution[
m, s, d]]}], ≪1≫.≪7≫] should be a list of non−negative machine−sized integers
giving the dimensions for the result.

General: Further output of RandomVariate::array will be suppressed during this calculation.

SmoothKernelDistribution: The input data
SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1.}, {EstimatedDistribution[Transpose[
Differences[≪1≫]], StudentTDistribution[m, s, d]], EstimatedDistribution[Transpose[Differences[≪1≫]],
≪20≫[m, s, d]]}], ≪1≫.≪7≫]] should be a vector or a matrix of real numbers or a
valid TemporalData object.

General: Further output of SmoothKernelDistribution::invldd will be suppressed during this calculation.

NIntegrate: The integrand
x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[≪2≫],
EstimatedDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to
non−numerical values for all sampling points in the region with boundaries {{−∞, 0.
+InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]],
0.01]}}.

Out[398]= 100. NIntegrate[x PDF[distTC, x], {x, -∞, nVaRTC}]

In[399]:=
```
TableForm[{{simPortNGMoments, nVaRNG, nCVaRNG},
  {simPortNCMoments, nVaRNC, nCVaRNC}, {simPortTCMoments, nVaRTC, nCVaRTC}},
 TableHeadings → {{"NG", "NC", "TC"}, {"Portfolio Mmts", "VaR", "CVaR"}}]
```

RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

RandomVariate: The array dimensions {0.5, 0.5}.10000 given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

RandomVariate: The array dimensions {0.5, 0.5}.10000. given in position 2 of ≪1≫ should be a list of non−negative machine−sized integers giving the dimensions for the result.

General: Further output of RandomVariate::array will be suppressed during this calculation.

SmoothKernelDistribution: The input data SmoothKernelDistribution[≪1≫] should be a vector or a matrix of real numbers or a valid TemporalData object.

General: Further output of SmoothKernelDistribution::invldd will be suppressed during this calculation.

NIntegrate: The integrand x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5}, {NormalDistribution[≪2≫], NormalDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to non−numerical values for all sampling points in the region with boundaries {{−∞, 0. + InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]], 0.01]}}.

NIntegrate: The integrand x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribution[≪2≫], NormalDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to non−numerical values for all sampling points in the region with boundaries {{−∞, 0. + InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]], 0.01]}}.

NIntegrate: The integrand x PDF[SmoothKernelDistribution[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {EstimatedDistribution[≪2≫], EstimatedDistribution[≪2≫]}], {0.5, 0.5}.10000]], x] has evaluated to non−numerical values for all sampling points in the region with boundaries {{−∞, 0. + InverseCDF[SmoothKernelDistribution[RandomVariate[{≪2≫}.CopulaDistribution[≪2≫], {≪2≫}.10000.]], 0.01]}}.

General: Further output of NIntegrate::inumr will be suppressed during this calculation.

Out[399]//TableForm=

```
NG   Mean[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5}, {NormalDistri
     StandardDeviation[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5},
     Skewness[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5}, {NormalDi
     Kurtosis[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Binormal, 0.5}, {NormalDi
NC     Mean[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDistribu
       StandardDeviation[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {N
       Skewness[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDis
       Kurtosis[RandomVariate[{0.5, 0.5}.CopulaDistribution[{Clayton, 1}, {NormalDis
TC                                       Mean[RandomVariate[{0.5, 0.5}.CopulaDistributi
                                         StandardDeviation[RandomVariate[{0.5, 0.5}.Cop
                                         Skewness[RandomVariate[{0.5, 0.5}.CopulaDistri
                                         Kurtosis[RandomVariate[{0.5, 0.5}.CopulaDistri
```