

```

Manipulate[
  bivariateNormal[u1, u2, sigma1, sigma2, N[p], xyRange, contours, colorScheme,
    nRandomPoints, keepCentered, densityPlotType, condProb, givenValueForCondProb],
  Style["Parameter Values", Bold],
  {{u1, 0, "Mean of X,  $u_x$ "}, -100, 100, .1, ImageSize → Tiny, Appearance → "Labeled"},
  {{u2, 0, "Mean of Y,  $u_y$ "}, -100, 100, .1, ImageSize → Tiny, Appearance → "Labeled"},
  {{sigma1, 1, " $\sigma_x$ "}, 0.0001, 10, .1, ImageSize → Tiny, Appearance → "Labeled"},
  {{sigma2, 1, " $\sigma_y$ "}, 0.0001, 10, .1, ImageSize → Tiny, Appearance → "Labeled"},
  {{p, .9, Style["correlation factor p=", 9]},
   -.9999, .9999, .01, ImageSize → Tiny, Appearance → "Labeled"}},
  {{keepCentered, True, Style["Center plots\nnon joint mean?", 9]}, {True, False}},
  Delimiter,
  Style["Contour options", Bold],
  {{contours, 5, Style["contour levels?", 9]}, {
    1 → "1",
    2 → "2",
    3 → "3",
    4 → "4",
    5 → "5",
    6 → "6",
    7 → "7",
    8 → "8",
    9 → "9",
    10 → "10",
    15 → "15",
    20 → "20",
    30 → "30",
    40 → "40"
  }, ControlType → PopupMenu, ImageSize → Tiny},
  {{colorScheme, 0, Style["Color scheme?", 9]}, {
    0 → "None",
    1 → "Blue",
    2 → "Gray",
    3 → "Green",
    4 → "Black",
    5 → "TemperatureMap",
    6 → "BlueGreenYellow",
    7 → "RustTones"
  }, ControlType → PopupMenu, ImageSize → Tiny},
  {{densityPlotType, 1, "Select extra plot type"}, {1 → contour, 2 → density, 3 → stairs}},
  Delimiter,
  Style["Other Misc. options", Bold],
  {{xyRange, 3, Style["How many std\nfrom mean to\ndisplay for plots?", 9]},
   1, 4, .1, Appearance → "Labeled", ImageSize → Tiny},
  {{nRandomPoints, 100, Style["How many points\nFor random plot?", 9]},
   1, 1000, 10, Appearance → "Labeled", ImageSize → Tiny},
  Delimiter,
  Style["Conditional probability options", Bold],
  {condProb, 0, "Show Conditional Probability?"},
  {0 → "No", 1 → "P(Y|X)", 2 → "P(X|Y)", ControlType → RadioButtonBar, ImageSize → Tiny},
  {{givenValueForCondProb, 0, "Enter the given R.V. value="},
   ControlType → InputField[0, Number], FieldSize → 5,
   Enabled → If[condProb == 0, False, True], ImageSize → Tiny},

```

```

Delimiter,
{{runIt, True, ""},
 Button[Style["Click to process", Bold, 14], runIt = True] &, ControlPlacement -> Left},
ControlPlacement -> Left, Alignment -> Left, ContinuousAction -> False, Initialization :>
{
Get["MultivariateStatistics`"];

(*****)
(*      getXYRange          *)
(*****)

getXYrange[var_, μ_, buffer_, isCentered_] := Module[{},
  If[isCentered, Return[{var, μ - buffer, μ + buffer}]];
  If[μ ≤ 0, Return[{var, μ - buffer, buffer}]];
  Return[{var, -buffer, μ + buffer}];
];

getXYrange[var_, μ_, buffer_, isCentered_, increment_] := Module[{},
  If[isCentered, Return[{var, μ - buffer, μ + buffer, increment}]];
  If[μ ≤ 0, Return[{var, μ - buffer, buffer, increment}]];
  Return[{var, -buffer, μ + buffer, increment}];
];

getDataRange[μx_, xBuffer_, μy_, yBuffer_, isCentered_] := Module[{rx, ry},
  If[isCentered,
    Return[ { {μx - xBuffer, μx + xBuffer}, {μy - yBuffer, μy + yBuffer} } ] ];
  If[μx ≤ 0, rx = {μx - xBuffer, μx + xBuffer}, rx = {-xBuffer, μx + xBuffer}];
  If[μy ≤ 0, ry = {μy - yBuffer, μy + yBuffer}, ry = {-yBuffer, μy + yBuffer}];
  Return[{rx, ry}];
];

conditionalBivariateNormalDensity[conditionalOfVariable_, givenVariable_, σOfVariable_,
σGivenVariable_, μOfVariable_, μGivenVariable_, correlationFactor_] :=
Module[{y = conditionalOfVariable, x = givenVariable, oy = σOfVariable,
ox = σGivenVariable, μx = μOfVariable, μy = μGivenVariable, ρ = correlationFactor},

$$N\left[\frac{1}{\sigma_y \sqrt{2 \pi (1 - \rho^2)}} \text{Exp}\left[-\frac{1}{2} \frac{(y - \mu_y - \rho \frac{\sigma_y}{\sigma_x} (x - \mu_x))^2}{\sigma_y^2 (1 - \rho^2)}\right], 16\right]$$

];
];

(*****)
(*      processBivariateNormal      *)
(*****)

bivariateNormal[μx_, μy_, ox_, σy_, p_, xyRange_, contours_, colorName_, nRandomPoints_,
keepCentered_, densityPlotType_, condProb_, givenValueForCondProb_] :=
DynamicModule[{oxx, oyy, oxy, correlationMatrix, p1, p2, detCorrelationMatrix,
optContour, optPlot3D, px, py, xrange, yrange, plotXrange, plotYrange, tbl, pRandom,
pCDF, sizeRandomPlot, sizeCDFplot, sizePDFplot, sizeMarginalplot, sizeContourPlot,
nMeshLines, m, detCorrelationMatrixSymbolic, viewPoint, pdfDataPoints, cdfDataPoints,
xrangeForPDFTable, yrangeForPDFTable, xrangeForCDFTable, yrangeForCDFTable,
skewness, kurtosis, charFunction, var, label, pdfDataIncrement, cdfDataIncrement,
pdfDataPointsForStairs, xrangeForPDFTableStairs, yrangeForPDFTableStairs,

```

```

dataRange, conditionalProbabilitySlice, pCondProb, pMarginalPlot},

pdfDataIncrement = .1;
cdfDataIncrement = .5;

xRange = getXYrange[x,  $\mu_x$ , xyRange *  $\sigma_x$ , keepCentered];
yRange = getXYrange[y,  $\mu_y$ , xyRange *  $\sigma_y$ , keepCentered];
xRangeForPDFTable = getXYrange[x,  $\mu_x$ , xyRange *  $\sigma_x$ , keepCentered, pdfDataIncrement];
xRangeForCDFTable = getXYrange[x,  $\mu_x$ , xyRange *  $\sigma_x$ , keepCentered, cdfDataIncrement];
yRangeForPDFTable = getXYrange[y,  $\mu_y$ , xyRange *  $\sigma_y$ , keepCentered, pdfDataIncrement];
yRangeForCDFTable = getXYrange[y,  $\mu_y$ , xyRange *  $\sigma_y$ , keepCentered, cdfDataIncrement];

dataRange = getDataRange[ $\mu_x$ , xyRange *  $\sigma_x$ ,  $\mu_y$ , xyRange *  $\sigma_y$ , keepCentered];

nMeshLines = 20;

 $\sigma_{xx}$  =  $\sigma_x^2$ ;
 $\sigma_{yy}$  =  $\sigma_y^2$ ;
 $\sigma_{xy}$  = p *  $\sigma_{xx}$  *  $\sigma_{yy}$ ;
correlationMatrix = N[{{ $\sigma_{xx}$ ,  $\sigma_{xy}$ }, { $\sigma_{xy}$ ,  $\sigma_{yy}$ }]];
detCorrelationMatrixSymbolic =
  MatrixForm[{{{" $\sigma_x^2$ ", " ", "p  $\sigma_x^2 \sigma_y^2$ "}, {"p  $\sigma_x^2 \sigma_y^2$ ", " ", " $\sigma_y^2$ "}}];
detCorrelationMatrix = Det[correlationMatrix];
m = MatrixForm[correlationMatrix];
If[detCorrelationMatrix <= 0,
  If[detCorrelationMatrix == 0,
    Return[
      Text[Row[{"Opps, your input generated a covariance Matrix with ZERO determinant!
        try again ", detCorrelationMatrixSymbolic, " = ", m}]]],
    Return[Text[Row[{"Opps, your input generated a covariance Matrix
      with NEGATIVE determinant! try again
      ", detCorrelationMatrixSymbolic, " = ", m}]]]
  ];
];

If[condProb > 0,
  Which[
    condProb = 1, (*given X*)
    conditionalProbabilitySlice = Table[If[x != givenValueForCondProb, -1,
      conditionalBivariateNormalDensity[y, givenValueForCondProb,  $\sigma_y$ ,  $\sigma_x$ ,  $\mu_y$ ,  $\mu_x$ , p]],
      Evaluate[yRangeForPDFTable], Evaluate[xRangeForPDFTable]],
    condProb = 2,
    conditionalProbabilitySlice = Table[If[y != givenValueForCondProb, -1,
      conditionalBivariateNormalDensity[x, givenValueForCondProb,  $\sigma_y$ ,  $\sigma_x$ ,  $\mu_y$ ,  $\mu_x$ , p]],
      Evaluate[yRangeForPDFTable], Evaluate[xRangeForPDFTable]]
  ]
];

sizeRandomPlot = {200};
sizeCDFplot = {300};
sizePDFplot = {300};
sizeMarginalplot = {200};
sizeContourPlot = {300};

```

```

Which[
  colorName == 0, {optPlot3D = ColorFunction -> None, optContour = ContourShading -> None},
  colorName == 1, {optPlot3D = PlotStyle -> Blue, optContour = optPlot3D},
  colorName == 2, {optPlot3D = PlotStyle -> Gray, optContour = optPlot3D},
  colorName == 3, {optPlot3D = PlotStyle -> Green, optContour = optPlot3D},
  colorName == 4, {optPlot3D = PlotStyle -> Black, optContour = optPlot3D},
  colorName == 5, {optPlot3D = ColorFunction -> "TemperatureMap"; optContour = optPlot3D},
  colorName == 6, {optPlot3D = ColorFunction -> "BlueGreenYellow"; optContour = optPlot3D},
  colorName == 7, {optPlot3D = ColorFunction -> "RustTones"; optContour = optPlot3D}
];

viewPoint = {.5, -2, 1};

skewness = Skewness[MultinormalDistribution[{μx, μy}, correlationMatrix]];
kurtosis = Kurtosis[MultinormalDistribution[{μx, μy}, correlationMatrix]];
charFunction =
  CharacteristicFunction[MultinormalDistribution[{μx, μy}, correlationMatrix], {x, y}];
var = Variance[MultinormalDistribution[{μx, μy}, correlationMatrix]];

pdfDataPoints = N[Table[ PDF[MultinormalDistribution[{μx, μy}, correlationMatrix],
  {x, y}], Evaluate[xRangeForPDFTable], Evaluate[yRangeForPDFTable]]];

label = Row[{ Style["Joint PDF fx,y(x,y).    skewness=", 10], skewness,
  Style["\nkurtosis=", 9], kurtosis, Style["  Variance=", 9], var, "\n"}];

conditionalProbabilitySlice = Chop[conditionalProbabilitySlice];

If[condProb > 0,
  pCondProb = ListPlot3D[conditionalProbabilitySlice,
    PlotLabel -> label,
    AxesLabel -> {x, y, fxy},
    PlotRange -> {All, All, {0,
      Evaluate[Max[ Max[conditionalProbabilitySlice], Max[pdfDataPoints] ]]}},
    ImageSize -> sizePDFplot,
    PlotStyle -> Red,
    ViewPoint -> viewPoint,
    AspectRatio -> Automatic,
    DataRange -> dataRange
  ],
  ];
]

p1 = ListPlot3D[pdfDataPoints,
  PlotLabel -> label,
  AxesLabel -> {x, y, fxy},
  PlotRange -> {All, All, {0, Evaluate[Max[
    If[condProb > 0, Max[conditionalProbabilitySlice], 0], Max[pdfDataPoints]]]}},
  ImageSize -> sizePDFplot,
  Evaluate[optPlot3D],
  ViewPoint -> viewPoint,
  AspectRatio -> Automatic,
  DataRange -> dataRange
];

If[condProb > 0, p1 = Show[pCondProb, p1]];

```

```

Which[
  densityPlotType = 1,
  p2 = ListContourPlot[pdfDataPoints,
    ContourLabels → (Text[Framed[#3], {#1, #2}, Background → White] &),
    ImageSize → sizeContourPlot,
    Evaluate[optContour],
    Contours → contours,
    AxesLabel → {"X", "Y"},
    PlotLabel → "Contour Plot",
    AspectRatio → Automatic,
    DataRange → dataRange
  ],
  densityPlotType = 2,
  p2 = ListDensityPlot[pdfDataPoints,
    Mesh → None,
    InterpolationOrder → 0,
    ColorFunction → "SouthwestColors",
    PlotLabel → "Density Plot",
    DataRange → dataRange,
    AspectRatio → Automatic,
    ImageSize → sizeContourPlot],
  densityPlotType = 3,
  {
    pdfDataIncrement = .2;
    xRangeForPDFTableStairs =
      getXYrange[x, μx, xyRange * σx, keepCentered, pdfDataIncrement];
    yRangeForPDFTableStairs = getXYrange[y, μy, xyRange * σy,
      keepCentered, pdfDataIncrement];

    pdfDataPointsForStairs =
      N[Table[ PDF[MultinormalDistribution[{μx, μy}, correlationMatrix], {x, y}],
        Evaluate[xRangeForPDFTableStairs], Evaluate[yRangeForPDFTableStairs]]];
    p2 = ListPlot3D[pdfDataPointsForStairs,
      PlotLabel → "Stairs view of joint PDF",
      AxesLabel → {X, Y, FXY},
      ImageSize → sizeCDFplot,
      PlotRange → All,
      Mesh → None,
      InterpolationOrder → 0,
      ColorFunction → "SouthwestColors",
      Filling → Axis,
      DataRange → dataRange,
      AspectRatio → Automatic,
      ViewPoint → viewPoint]
  }
];
px = Plot[PDF[NormalDistribution[μx, σx], x], Evaluate[xRange],
  PlotRange → All, ImageSize → sizeMarginalplot, PlotStyle → Red];
py = Plot[PDF[NormalDistribution[μy, σy], y], Evaluate[yRange],

```

```

    PlotRange -> All, ImageSize -> sizeMarginalplot, PlotStyle -> Black];
pMarginalPlot = Show[px, py, PlotLabel -> {Style["Marginal fx(x)", Red],
                                         Style["Marginal fy(y)", Black]}];

tbl = Table[RandomReal[
    MultinormalDistribution[{μx, μy}, correlationMatrix]], {i, nRandomPoints}];

pRandom = ListPlot[tbl, PlotLabel -> Style[
    "Scatter Plot independent pairs of \nbivariate normal random variables", Smaller],
    AxesLabel -> {"x", "y"}, ImageSize -> sizeRandomPlot, PlotRange -> All];

cdfDataPoints = N[Table[ CDF[MultinormalDistribution[{μx, μy}, correlationMatrix],
    {x, y}], Evaluate[xRangeForCDFTable], Evaluate[yRangeForCDFTable]]];

label = Row[{Style["CDF Fx,y(x,y)", 10],
             Style["\nCharacteristicFunction ", 10], Text[charFunction]}];
pCDF = ListPlot3D[cdfDataPoints,
    PlotLabel -> label,
    AxesLabel -> {X, Y, Fxy},
    ImageSize -> sizeCDFplot,
    PlotRange -> All,
    AspectRatio -> Automatic,
    DataRange -> dataRange
];
Grid[{
    { Grid[{ {p1, pCDF} }, Frame -> None, Spacings -> 1, ItemSize -> {Full, Full}] }
    , { Grid[{{p2, Grid[{ {pRandom}, { Grid[{{pMarginalPlot}}, Frame ->
        None, Spacings -> 1, ItemSize -> {Full, Full}}] } ] } },
        Frame -> None, Spacings -> 1, ItemSize -> {Full, Full}] }
    } ,
    Frame -> All, Spacings -> 1, ItemSize -> {Full, Full}]
];
}
]
] (*end Manipulate*)

```

**Parameter Values**

Mean of X,  $u_X =$

Mean of Y,  $u_Y =$

$\sigma_X =$

$\sigma_Y =$

correlation factor  $p =$

Center plots  on joint mean?

---

**Contour options**

contour levels?

Color scheme?

Select extra plot type

---

**Other Misc. options**

How many std from mean to display for plots?

How many points for random plot?

---

**Conditional probability options**

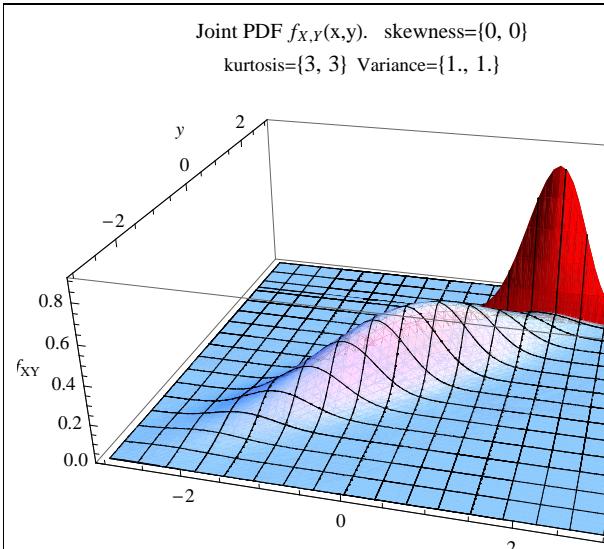
Show Conditional Probability?

Enter the given R.V. value=

---

**Click to process**

Joint PDF  $f_{X,Y}(x,y)$ . skewness={0, 0}  
kurtosis={3, 3} Variance={1., 1.}

**Contour Plot**

