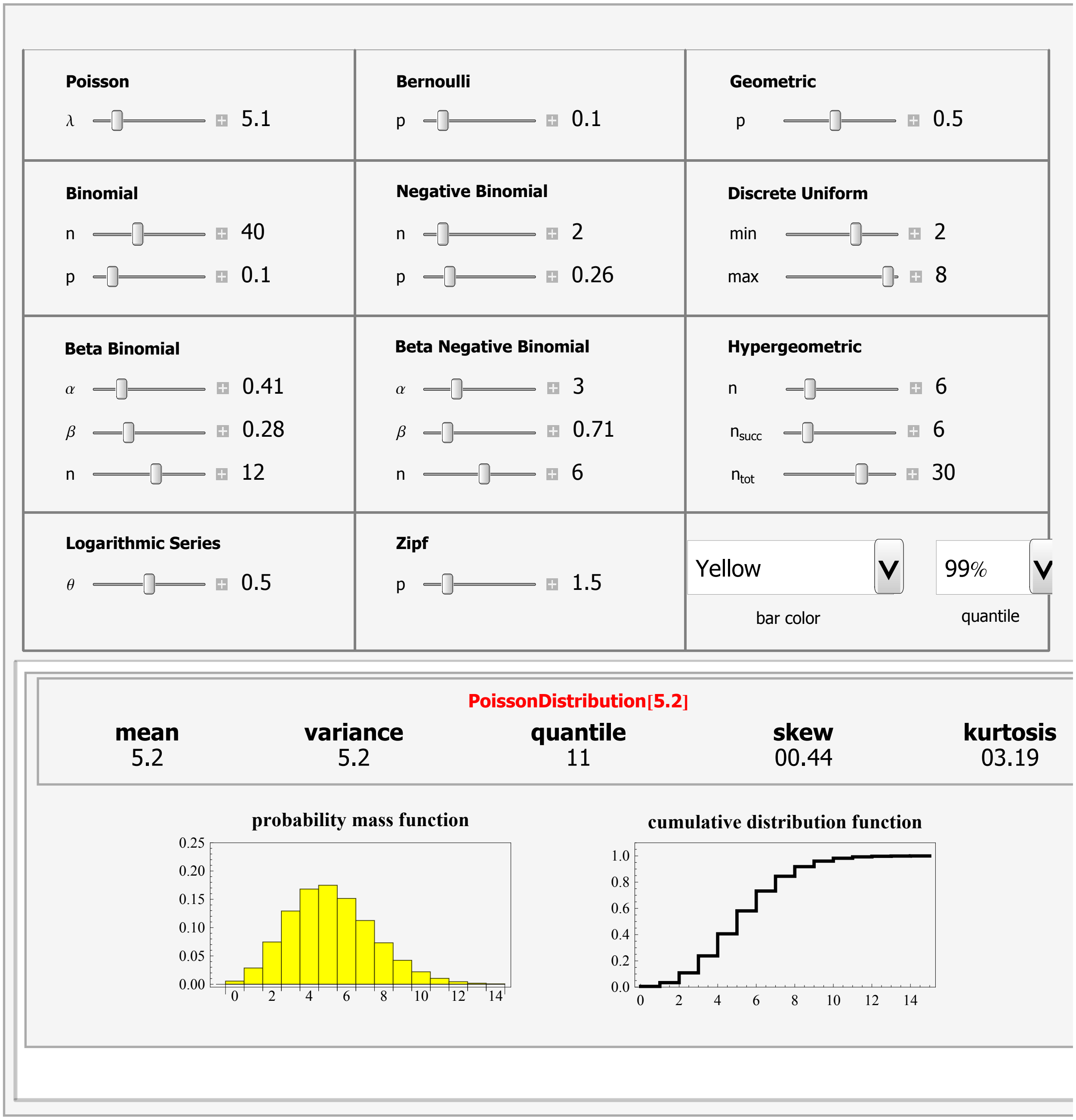
```
Manipulate[
 Row[{
   Dynamic[Refresh[
     Switch[
      dynamicsInitialized,
      {True, True, True, True, True, True, True, True, True, True, True, False},
      dynamicsInitialized[[12]] = True; gdist = gdistInitial;
      gfrom = gfromInitial; gcdf = gcdfInitial,

      {True, True, True, True, True, True, True, True, True, True, True, True},
      gdistInitial = gdist; gfromInitial = gfrom; gcdfInitial = gcdf
     ];
     process[gdist, gfrom, gcdf, barChartColor, quantile],
     TrackedSymbols :→ {quantile, barChartColor, θ, pZipf, nHyperGeometric
        nBlackHyperGeometric, nTotalHyperGeometric, αBetaNegativeBinomial,
       βBetaNegativeBinomial, nBetaNegativeBinomial, nBetaBinomial, αBetaBinomial,
       βBetaBinomial, min, discreteUniformMax, nNegativeBinomial, pNegativeBinomial,
       nBinomial, pBinomial, pGeometric, pBernoulli, λ, dynamicsInitialized}]
   ],
   Dynamic[Refresh[dynamicsInitialized[[1]] = True;
     gdist = PoissonDistribution[λ]; gfrom = 0; gcdf = .999; "", TrackedSymbols → {λ}]],
   Dynamic[Refresh[dynamicsInitialized[[2]] = True; gdist = BernoulliDistribution[
       pBernoulli]; gfrom = 0; gcdf = 1; "", TrackedSymbols → {pBernoulli}]],
   Dynamic[Refresh[dynamicsInitialized[[3]] = True; gdist = GeometricDistribution[
       pGeometric]; gfrom = 0; gcdf = .999; "", TrackedSymbols → {pGeometric}]],
   Dynamic[Refresh[dynamicsInitialized[[4]] = True;
     gdist = BinomialDistribution[Round[nBinomial], pBinomial]; gfrom = 0;
     gcdf = .999; "", TrackedSymbols → {nBinomial, pBinomial}]],
   Dynamic[Refresh[dynamicsInitialized[[5]] = True; gdist = NegativeBinomialDistribution[
       Round[nNegativeBinomial], If[pNegativeBinomial == 0, 0.01, pNegativeBinomial]];
     gfrom = 0; gcdf = .999; "", TrackedSymbols → {nNegativeBinomial, pNegativeBinomial}]],
   Dynamic[Refresh[dynamicsInitialized[[6]] = True; gdist = DiscreteUniformDistribution[
       {min, If[discreteUniformMax < min, discreteUniformMax = min, discreteUniformMax]}];
     gfrom = min - 1; gcdf = 1; "", TrackedSymbols → {discreteUniformMax, min}]],
   Dynamic[Refresh[dynamicsInitialized[[7]] = True; gdist = BetaBinomialDistribution[
       αBetaBinomial, βBetaBinomial, Round[nBetaBinomial]]; gfrom = 0; gcdf = 1;
     "", TrackedSymbols → {αBetaBinomial, βBetaBinomial, nBetaBinomial}]],
   Dynamic[Refresh[dynamicsInitialized[[8]] = True; gdist =
      BetaNegativeBinomialDistribution[αBetaNegativeBinomial, βBetaNegativeBinomial,
       Round[nBetaNegativeBinomial]]; gfrom = 0; gcdf = .85; "", TrackedSymbols →
      {αBetaNegativeBinomial, βBetaNegativeBinomial, nBetaNegativeBinomial}]],
   Dynamic[Refresh[dynamicsInitialized[[9]] = True;
     gdist = HypergeometricDistribution[Round[If[nHyperGeometric > nTotalHyperGeometric,
         nHyperGeometric = nTotalHyperGeometric, nHyperGeometric]],
       Round[If[nBlackHyperGeometric > nTotalHyperGeometric,
         nBlackHyperGeometric = nTotalHyperGeometric, nBlackHyperGeometric]],
       Round[nTotalHyperGeometric]]; gfrom = 0; gcdf = 1; "",
     TrackedSymbols → {nHyperGeometric, nTotalHyperGeometric, nBlackHyperGeometric}]],
   Dynamic[Refresh[dynamicsInitialized[[10]] = True; gdist = LogSeriesDistribution[θ];
     gfrom = 0; gcdf = .99; "", TrackedSymbols → {θ}]],
   Dynamic[Refresh[dynamicsInitialized[[11]] = True; gdist = ZipfDistribution[pZipf];
     gfrom = 0; gcdf = .99; "", TrackedSymbols → {pZipf}]]
 }],
Grid[{ {
   Labeled[Control[{{λ, 5.1, Style["λ", 10, AutoSpacing → False]}, 0.1, 30, .1, ImageSize →
       Tiny, Appearance → "Labeled"}] , Style["Poisson", 10, Bold], {{Top, Left}}],

   Labeled[Control[{{pBernoulli, 0.1, Style["p", 10, AutoSpacing → False]},
      0, 1, .025, ImageSize → Tiny, Appearance → "Labeled"}],
    Style["Bernoulli", 10, Bold], {{Top, Left}}],

   Labeled[Control[{{pGeometric, 0.5, Style[" p     ", 10, AutoSpacing → False]},
      0.1, 1, .01, ImageSize → Tiny, Appearance → "Labeled"}],
    Style["Geometric", 10, Bold], {{Top, Left}}]
  }, (*second row *){
   Labeled[Grid[{{
       Control[{{nBinomial, 40, Style["n", 10, AutoSpacing → False]},
         1, 100, 1, ImageSize → Tiny, Appearance → "Labeled"}]} ,
      {Control[{{pBinomial, 0.1, Style["p", 10, AutoSpacing → False]},
         0, 1, 0.05, ImageSize → Tiny, Appearance → "Labeled" }]
      }}], Style["Binomial", 10, Bold], {{Top, Left}}
   ],
   Labeled[
    Grid[ {{
       Control[{{nNegativeBinomial, 2, Style["n", 10, AutoSpacing → False]},
         1, 10, 1, ImageSize → Tiny, Appearance → "Labeled" }]} ,
      {Control[{{pNegativeBinomial, 0.26, Style["p", 10, AutoSpacing → False]},
         .1, 1, 0.1, ImageSize → Tiny, Appearance → "Labeled" }]
      }}], Style["Negative Binomial", 10, Bold], {{Top, Left}}
   ],
   Labeled[
    Grid[ {{
       Control[{{min, 2, Style["min  ", 10, AutoSpacing → False]},
         -8, 8 - 1, 1, ImageSize → Tiny, Appearance → "Labeled" }]} ,
      {Control[{{discreteUniformMax, 8, Style["max  ", 10, AutoSpacing → False]},
         -8 + 1, 8, 1, ImageSize → Tiny, Appearance → "Labeled" }]
      }}], Style["Discrete Uniform", 10, Bold], {{Top, Left}}]},
  { (*third row *)
   Labeled[
    Grid[ {{
       Control[{{αBetaBinomial, .41, Style["α", 10, AutoSpacing → False]},
         0.01, 2, 0.01, ImageSize → Tiny, Appearance → "Labeled" }]} ,
      {Control[{{βBetaBinomial, 0.28, Style["β", 10, AutoSpacing → False]},
         0.01, 1, 0.01, ImageSize → Tiny, Appearance → "Labeled"}]},
      {Control[{{nBetaBinomial, 12, Style["n", 10, AutoSpacing → False]},
         1, 20, 1, ImageSize → Tiny, Appearance → "Labeled"}]
      }}], Style["Beta Binomial", 10, Bold], {{Top, Left}}
   ],
   Labeled[
    Grid[ {{
       Control[{{αBetaNegativeBinomial, 3, Style["α", 10, AutoSpacing → False]},
         2.01, 6, 0.1, ImageSize → Tiny, Appearance → "Labeled"}]} ,
      {Control[{{βBetaNegativeBinomial, .71, Style["β", 10, AutoSpacing → False]},
         0.01, 5, 0.1, ImageSize → Tiny, Appearance → "Labeled"}]},
      {Control[{{nBetaNegativeBinomial, 6, Style["n", 10, AutoSpacing → False]},
         1, 10, 1, ImageSize → Tiny, Appearance → "Labeled" }]
      }}], Style["Beta Negative Binomial", 10, Bold], {{Top, Left}}
   ],
   Labeled[
    Grid[ {{
       Control[{{nHyperGeometric, 6, Style["n       ", 10, AutoSpacing → False]},
         0, 40, 1, ImageSize → Tiny, Appearance → "Labeled"}]} ,
      {Control[{{nBlackHyperGeometric, 6, Style["n_succ ", 10, AutoSpacing → False]},
         0, 40, 1, ImageSize → Tiny, Appearance → "Labeled"}]},
      {Control[{{nTotalHyperGeometric, 30, Style["n_tot  ", 10, AutoSpacing → False]},
         1, 40, 1, ImageSize → Tiny, Appearance → "Labeled" }]
      }}], Style["Hypergeometric", 10, Bold], {{Top, Left}}
   ]}, { (*4th row *)

   Labeled[Control[{{θ, 0.5, Style["θ", 10, AutoSpacing → False]},
      0.01, 1 - .02, 0.01, ImageSize → Tiny, Appearance → "Labeled"}] ,
    Style["Logarithmic Series", 10, Bold], {{Top, Left}}],

   Labeled[Control[
     {{pZipf, 1.5, Style["p", 10, AutoSpacing → False]}, 1, 4, 0.01, ImageSize → Tiny,
      Appearance → "Labeled" }] , Style["Zipf", 10, Bold], {{Top, Left}}],

   Grid[{{Labeled[
       Control[{{barChartColor, Yellow, ""},
         {Red → Style["Red", sz], LightRed → Style["Light Red", sz],
          Green → Style["Green", sz], LightGreen → Style["Light Green", sz],
          Yellow → Style["Yellow", sz], Blue → Style["Blue", sz],
          LightBlue → Style["Light Blue", sz], Black → Style["Black", sz],
          Gray → Style["Gray", sz], LightGray → Style["Light Gray", sz],
          Cyan → Style["Cyan", sz], LightCyan → Style["Light Cyan", sz],
          Magenta → Style["Magenta", sz], LightMagenta → Style["Light Magenta", sz],
          Brown → Style["Brown", sz], LightBrown → Style["Light Brown", sz],
          Orange → Style["Orange", sz], LightOrange → Style["Light Orange", sz],
          Pink → Style["Pink", sz], LightPink → Style["Light Pink", sz]},
         ControlType → PopupMenu}], Style["bar color", 10], Bottom],
      Labeled[
       Control[{{quantile, .99, ""}, {.99 → Style["99%", sz],
          .98 → Style["98%", sz], .97 → Style["97%", sz], .96 → Style["96%", sz],
          .95 → Style["95%", sz], .90 → Style["90%", sz], .75 → Style["75%", sz],
          .50 → Style["50%", sz], .25 → Style["25%", sz], .15 → Style["15%", sz],
          .05 → Style["5%", sz], .025 → Style["2.5%", sz], .01 → Style["1%", sz]},
         ControlType → PopupMenu}], Style["quantile", 10], Bottom]}}]}}
 , Frame → All, FrameStyle → Directive[AbsoluteThickness[.1], Gray], Spacings → {0, 2},
 ItemSize → {{20, 20, 22}}],

{{gdist, PoissonDistribution[5.2]}, ControlType → None},
{{gfrom, 0}, ControlType → None},
{{gcdf, .999}, ControlType → None},
{{gdistInitial, PoissonDistribution[5.2]}, ControlType → None},
{{gfromInitial, 0}, ControlType → None},
{{gcdfInitial, .999}, ControlType → None},
{{sz, 10}, ControlType → None}, (*size of labels for controls, see above *)
{{plotWidth, 246}, ControlType → None},
{{plotHeight, 153}, ControlType → None},
{{plotImagePadding, 32}, ControlType → None},
{{cdft, Style["cumulative distribution function", Medium, Bold]}, ControlType → None},
{{pmft, Style["probability mass function", Medium, Bold]}, ControlType → None},
{{dynamicsInitialized, {False, False, False, False, False,
   False, False, False, False, False, False, False}}, ControlType → None},
FrameMargins → 0,
ImageMargins → 0,
ContinuousAction → False,
SynchronousUpdating → True,
AutorunSequencing → Range[3],

Initialization :→ (
  dynamicsInitialized =
   {False, False, False, False, False, False, False, False, False, False, False, False};
  process[dist_, from_, cdfUpper_, barChartColor_, quantile_] := Module[
    {label, cdfLabel, mean, var, vquantile, skew, kurtosis, k, pdf, cdf, max, tbl, to},

    {mean, var, vquantile, skew, kurtosis} = getStats[dist, quantile];
    to = InverseCDF[dist, cdfUpper];
    tbl = Table[PDF[dist, k], {k, from, to}];
    max = Max[tbl];

    pdf = BarChart[tbl,
      BarSpacing → None,
      ChartStyle → barChartColor,
      ImageSize → {plotWidth, plotHeight},
      AspectRatio → 0.48,
      Frame → True,
      AxesOrigin → {0, 0},
      FrameTicks → {{Automatic, None}, {None, None}},
      TicksStyle → Small,
      PlotLabel → None,
      FrameLabel → {{None, None}, {None, pmft}},
      PlotRange → {Automatic, {0, 1.4 max}},
      ImagePadding → plotImagePadding,
      ChartLabels → Placed[chartLabels[Length[tbl], {from, to}], Axis],
      LabelingFunction → (Placed[Style[#, Red, Bold], Tooltip] &)
     ];

    tbl = makeCDFdata[Table[CDF[dist, k], {k, from, to}] , from, to];

    cdf = ListPlot[tbl,
      AspectRatio → 0.48,
      Joined → True,
      Frame → True,
      ImagePadding → plotImagePadding,
      TicksStyle → Small,
      ImageSize → {plotWidth, plotHeight},
      PlotRange → {Automatic, {0, 1.1}},
    AxesOrigin → {0, 0},
    FrameTicks → {{Automatic, None}, {Automatic, None}},
    FrameLabel → {{None, None}, {None, cdft}},
    PlotLabel → None,
    PlotStyle → {Black, Thick},
    Axes → {True, False}
   ];

  Panel[
   Grid[{
     {
      Panel[
       Grid[{
          {Style[ToString[dist], 11, Red, Bold]},
          {Grid[
            {{Style["mean", Bold], Style["variance", Bold], Style["quantile", Bold],
              Style["skew", Bold], Style["kurtosis", Bold]}, {mean, var,
              vquantile, AccountingForm[skew, {4, 2}, NumberPadding → {"0", "0"},
               NumberSigns → {"", ""}], AccountingForm[kurtosis, {4, 2},
               NumberPadding → {"0", "0"}, NumberSigns → {"", ""}]}
            }, ItemSize → {{11, 12, 13, 12, 11}}, Alignment → Center, Spacings → {0, 0}
           ]
          }
         }
        ], FrameMargins → Medium
      ], SpanFromLeft
     },
     {Grid[{{pdf, cdf}}]}
    }, Frame → None, Alignment → Center, Spacings → {0, 0}, ItemSize → {{61}}
   ], FrameMargins → 0, Alignment → Center
  ]
 ];

chartLabels[len_, limits_] :=
 Module[{y = 10, r, incr, from = limits[[1]], to = limits[[2]]},
  If[len ≤ y, r = Range[from, to],
   {incr = Round[len / y];
    r = Table["", {i, 0, len - 1}];
    For[i = 1, i ≤  len, i = i + 1, If[Mod[i - 1, incr] == 0, r[[i]] = from + (i - 1)]]
   }]; r
 ];

makeCDFdata[data_, from_, to_] := Module[{i, x, d},
  x = Table[i, {i, from, to, 1}];
  d =
   Flatten[Table[ {{ x[[i]], data[[i]]}}, {x[[i]] + 1, data[[i]] }}, {i, 1, Length[x]}], 1
   ];
  Flatten[Table[ {Tooltip[ d[[i]], N[d[[i, 2]]]]}, {i, 1, Length[d]}], 1]
 ];

getStats[dist_, quantile_] := Module[{mean, var, vquantile, skew, kurtosis},
  mean = N[Quiet[Mean[dist], {Power::infy, Infinity::indet}]];
```

```
      If[mean === ComplexInfinity, mean = Infinity];

      var = N[Quiet[Variance[dist], {Power::infy, Infinity::indet}]];
      If[var === ComplexInfinity, var = Infinity];

      vquantile = Quiet[Quantile[dist, quantile], {Power::infy, Infinity::indet}];
      If[vquantile === ComplexInfinity, vquantile = Infinity];

      skew = N[Quiet[Skewness[dist], {Power::infy, Infinity::indet}]];
      If[skew === ComplexInfinity, skew = Infinity];

      kurtosis = N[Quiet[Kurtosis[dist], {Power::infy, Infinity::indet}]];
      If[kurtosis === ComplexInfinity, kurtosis = Infinity];

      {mean, var, vquantile, skew, kurtosis}
     ];
   ), SynchronousInitialization → True
 ]
```



Poisson

λ 5.1

Bernoulli

p 0.1

Geometric

p 0.5

Binomial

n 40

p 0.1

Negative Binomial

n 2

p 0.26

Discrete Uniform

min 2

max 8

Beta Binomial

α 0.41

β 0.28

n 12

Beta Negative Binomial

α 3

β 0.71

n 6

Hypergeometric

n 6

n_{succ} 6

n_{tot} 30

Logarithmic Series

θ 0.5

Zipf

p 1.5

bar color: Yellow

quantile: 99%

PoissonDistribution[5.2]

mean	variance	quantile	skew	kurtosis
5.2	5.2	11	00.44	03.19

probability mass function

cumulative distribution function