Probability Distributions

ref. https://www.youtube.com/watch?v=ixHKOz2vnoA

Table of Contents


```
begin
  using PlutoUI
  PlutoUI.TableOfContents(indent=true, depth=4, aside=true)
end
```

```
using Distributions  #, StatsPlots
```

```
using CairoMakie
```

Normal Distribution

Mean (μ): 1.5

```
md"Mean (μ): $(@bind μ PlutoUI.Slider(-3.0:0.1:3.0, 0.0, true))"
```

Standard Deviation (σ): 0.87

```
md"Standard Deviation (σ): $(@bind σ PlutoUI.Slider(0.5:0.01:3.0, 1.0, true))"
```

▸ [-1.16371, 1.55837, 0.499365, 2.10549, 1.94221, 2.90714, 1.18755, 1.31161, 0.303046, -0.46241, 1.34516, 0.923364, 2.82132, 1

```
begin
    const N₁ = 1_000_000
    nd = Distributions.Normal(μ, σ)
    v₁ = rand(nd, N₁);
end
```

```
draw_density! (generic function with 1 method)
```

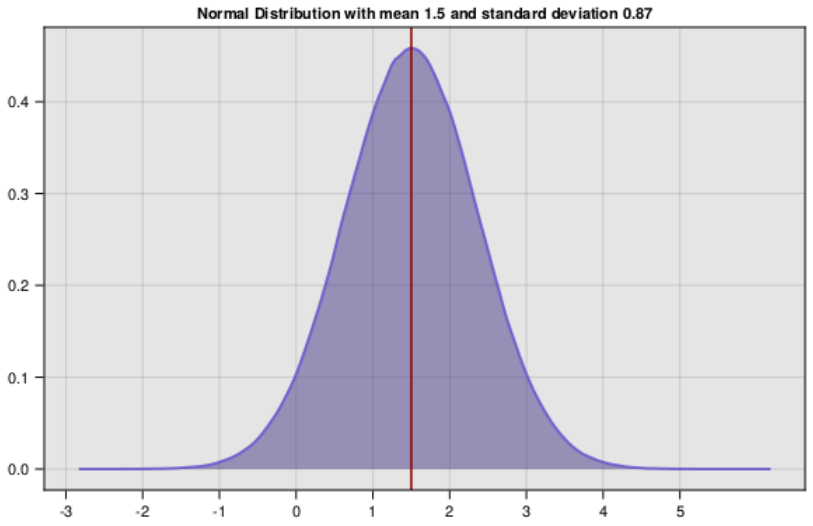
```
begin
    const Resolution = (600, 400)
    const Xlims  = 20:5:100
    const Ylims = 0:5:100
    const FontSize = 11

    def_fig() = CairoMakie.Figure(resolution = Resolution, fontsize=FontSize)

    def_ax(fig, title::String; xlims=Xlims, ylims=Ylims) = CairoMakie.Axis(
        fig[1, 1],
        xticks = xlims,
        yticks = ylims,
        title = title,
        backgroundcolor = :gray90,
    )

    draw_histo!(ax, v; norm=:pdf, bins=100) = CairoMakie.hist!(
        ax,
        v,
        normalization = norm,
        strokewidth = 0.2,
        strokecolor = (:black, 0.2),
        bins = bins
    )

    draw_density!(ax, v; norm=:pdf) = CairoMakie.density!(
        ax,
        v,
        normalization = norm,
        color=(:darkslateblue, 0.5),
        strokecolor = (:slateblue, 0.9),
        strokewidth = 2,
        strokearound = false
    )
end
```




```
begin
    f₁ = def_fig()
    ax₁ = def_ax(f₁, "Normal Distribution with mean $(μ) and standard deviation $(σ)";
        xlims=-5.:5., ylims=0.:0.1:0.9
    )
    draw_density!(ax₁, v₁;
        norm=:pdf
    )
    CairoMakie.vlines!(ax₁, μ, color = :darkred)
    f₁
end
```

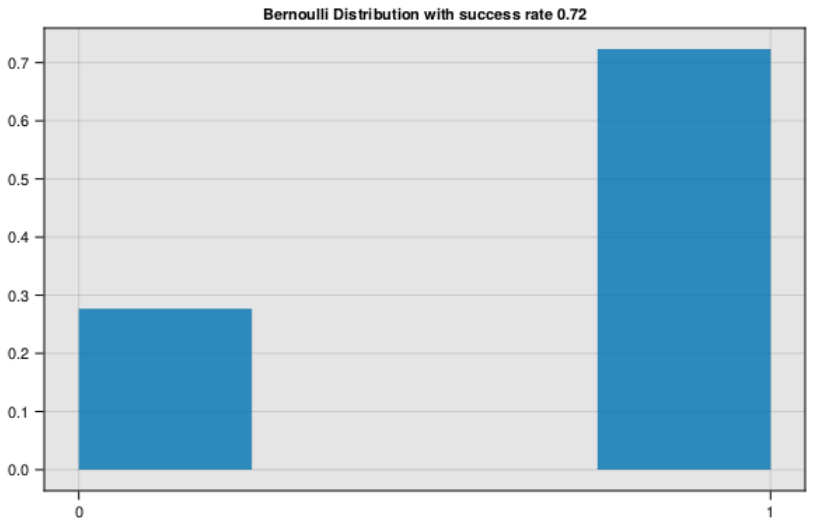
Bernoulli Distribution

Success rate (p): 0.72

```
md"Success rate (p): $(@bind p PlutoUI.Slider(0.0:0.01:1.0, 0.4, true))"
```

▸ [true, true, true, true, false, true, true, true, true, true, false, true, false, true, true, true, false, true, true, true,

```
begin
    const N₂ = 10_000
    b = Distributions.Bernoulli(p)
    v₂ = rand(b, N₂);
end
```




```
begin
    f₂ = def_fig()
    ax₂ = def_ax(f₂, "Bernoulli Distribution with success rate $(p)";
        xlims=[0, 1], ylims=0.:.1:1.,
    )
    draw_histo!(ax₂, v₂; norm=:probability, bins=4)
    f₂
end
```

Beta Distribution

Alpha (α): 1.4

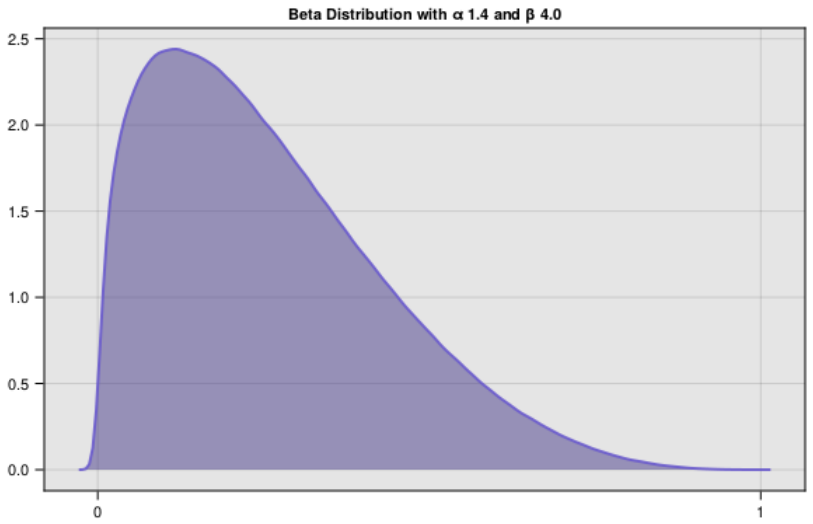
```
md"Alpha (α): $(@bind α PlutoUI.Slider(0.1:0.1:5., 1.0, true))"
```

Beta (β): 4.0

```
md"Beta (β): $(@bind β PlutoUI.Slider(0.1:0.1:5., 1.0, true))"
```

▸ [0.0652084, 0.388949, 0.364192, 0.335202, 0.248893, 0.338968, 0.177291, 0.146464, 0.0927353, 0.479976, 0.660362, 0.106643, (

```
begin
    const N₃ = 5_000_000
    bd = Distributions.Beta(α, β)
    v₃ = rand(bd, N₃);
end
```




```
begin
    f₃ = def_fig()
    ax₃ = def_ax(f₃, "Beta Distribution with α $(α) and β $(β)";
        xlims=[0., 1.], ylims=0.:0.5:5,
    )
    draw_density!(ax₃, v₃;
        norm=:pdf
    )
    f₃
end
```