# A Month of Playing with Probability using Mathematica

This notebook summarises all the copying I did from Nassim Taleb from his Tweets, his book, as well as his online MOOCs playlist. This summary was my attempt to get started in playing with probability using Mathematica/Wolfram Language.

## Learning probability through playing

This Mathematica notebook is an organised distillation of how I played with Mathematica to learn probability. This started with a fascination with the way Nassim Taleb did probability starting with Monte Carlo simulations. Unfortunately I do not have the luxury to learn it because of greed and fear of risks in the financial markets. Nor do I have the brains to think independently and come up with my own ideas. I only have the drive to get started with the remaining time I had with a Mathematica license from school. I did eventually end up purchasing Mathematica however.

This is in line with his recommendation to start with doing Monte Carlo simulations first, before trying to learn probability from a book.

## Connecting concepts with Mathematica

I wanted to start with some useful tips to use Mathematica as a computational explorer. For example, we can check the properties of any function in Mathematica with the following code.

```
In[40]:= ? ReliabilityDistribution
```

Out[40]=

Symbol

ReliabilityDistribution[*bexpr*, {{$x_1$, $dist_1$}, {$x_2$, $dist_2$}, …}] represents the reliability distribution for a system with components $x_i$ having reliability distribution $dist_i$, where the whole system is working when the Boolean expression *bexpr* is True, and component $x_i$ is working when $x_i$ is True.

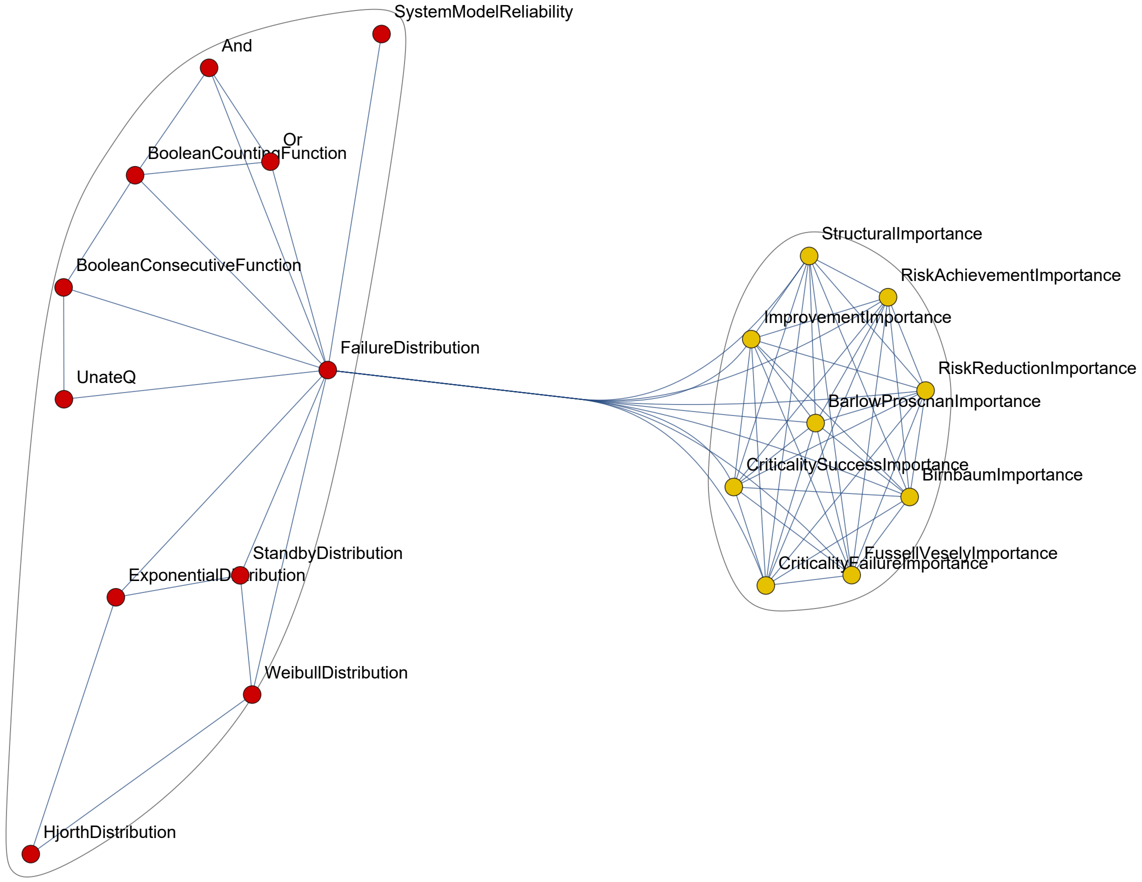
This brings up the help for the particular function. The help function actually does a good job of explaining the function being used, with numerous examples within it.

To really explore the various concepts, we can actually find a relationship community graph of any

Wolfram language function. This allows one to simply freely explore and take a look around of similar concepts.

In[41]:=
```
Show[WolframLanguageData["ReliabilityDistribution",
  "RelationshipCommunityGraph"], ImageSize → 600]
```

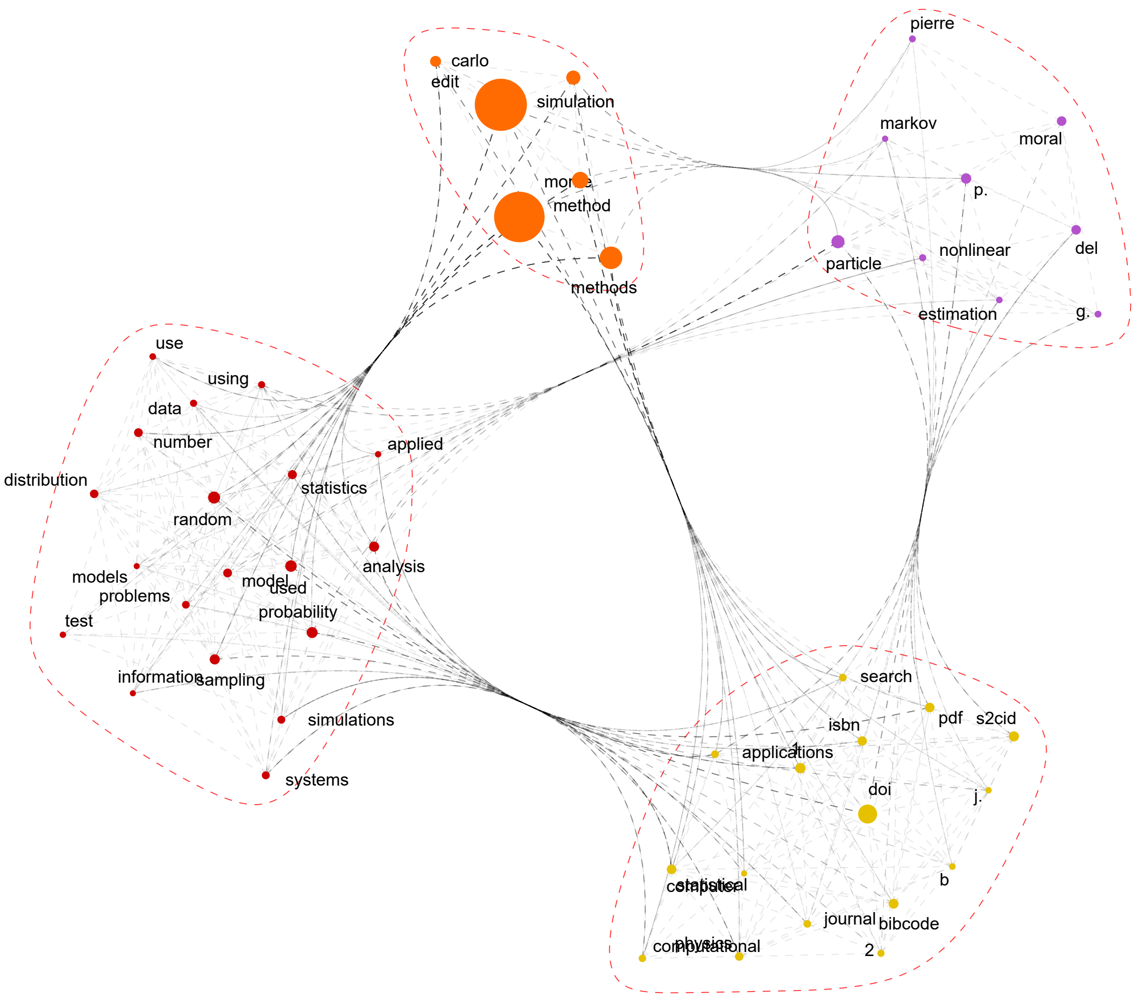
Out[41]=




Another technique I found useful in summarising an article is to import it using Mathematica and use one of the custom resource functions to delete the stopwords and plot a community graph plot of the relevant concepts.

In[59]:=
```
CommunityGraphPlot[ResourceFunction["KeywordsGraph"][
  DeleteStopwords[Import["https://en.wikipedia.org/wiki/Monte_Carlo_method"]], 50,
  VertexSize → "VertexWeight", EdgeStyle → Directive[Black, Dashed, Opacity[0.10]]],
 CommunityBoundaryStyle → Directive[Red, Dashed, Opacity[0.8]],
 ImageSize → 600, GraphLayout → "RadialEmbedding"]
```

Out[59]=

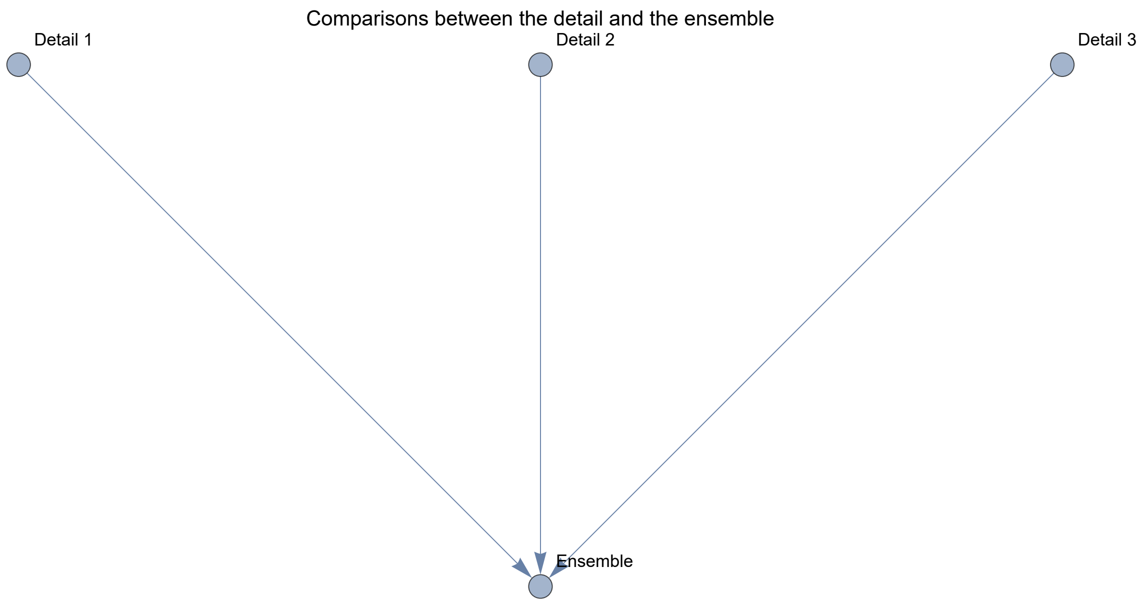



## A comment on disinformation, ensembles versus anecdotes

To try and extrapolate the detail from the ensemble is to commit an error in the reverse of the law of large numbers. Therefore, it is important to be rigid and discuss the particular scale of the problem [3].

In[97]:=
```
DirectedGraph[{"Detail 1" → "Ensemble", "Detail 2" → "Ensemble",
  "Detail 3" → "Ensemble"}, VertexLabels → "Name", ImageSize → 600,
 PlotLabel → "Comparisons between the detail and the ensemble"]
```

Out[97]=




# A simple demo with the Central Limit Theorem

The Monte Carlo class of methods rely on repeatedly sampling from a selected distributions so that one can get numerical results. It is important to get an intuition for the problem using a Monte Carlo simulation to understand the truth. This avoids the green lumber problem.

One can consult the “Statistical Consequences of Fat Tails” [1] for a complete definition of the Central Limit Theorem, with the full textbook form. The main demonstration of how the central limit theorem fails for fat tailed distribution. This code takes the sum of 100 Pareto random variables, and runs this summation $10^5$ times to form a histogram [2].

Before we begin, here is a reminder of the PostFix form.

In[60]:=
```
? //
```

Out[60]=

Symbol

Postfix[*f*[*expr*]] prints with *f*[*expr*] given in default postfix form: *expr* // *f*.

Postfix[*f*[*expr*], *h*] prints as *exprh*.

This will be used to create a table that sums a random variable drawn from a probability distribution.

In[61]:=
```
? ParetoDistribution
```
Out[61]=

Symbol

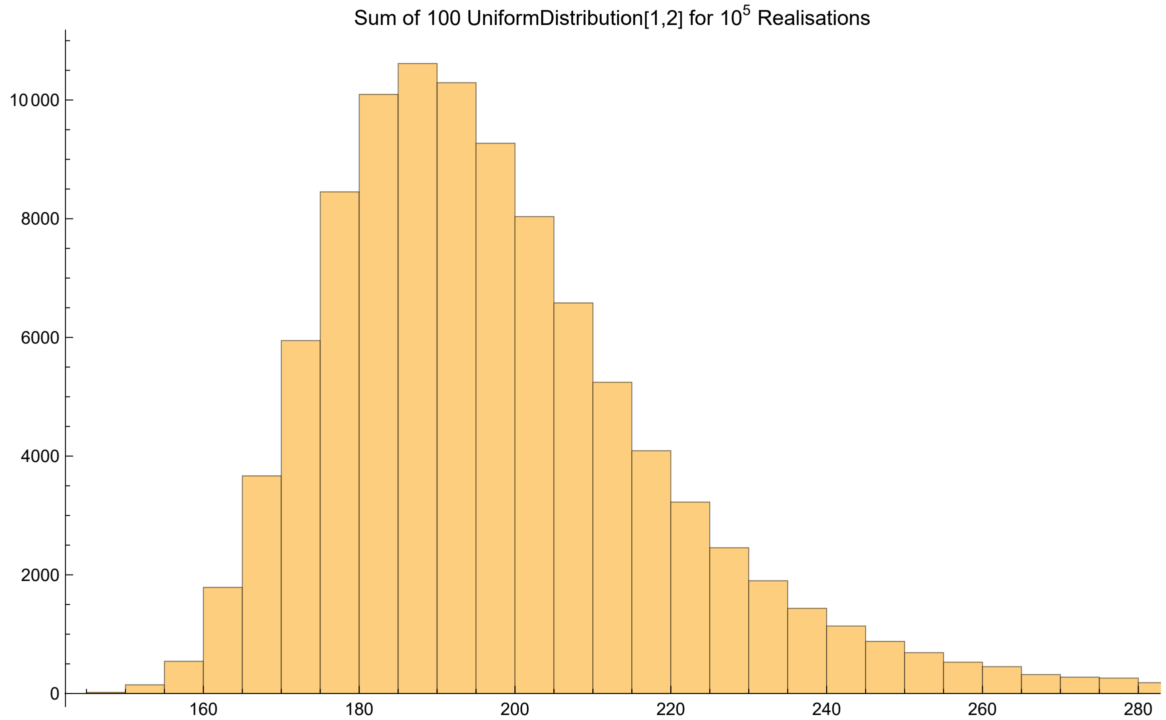
ParetoDistribution[$k$, $\alpha$] represents a Pareto

distribution with minimum value parameter $k$ and shape parameter $\alpha$.

ParetoDistribution[$k$, $\alpha$, $\mu$] represents a Pareto type II distribution with location parameter $\mu$.

ParetoDistribution[$k$, $\alpha$, $\gamma$, $\mu$] represents a Pareto type IV distribution with shape parameter $\gamma$.

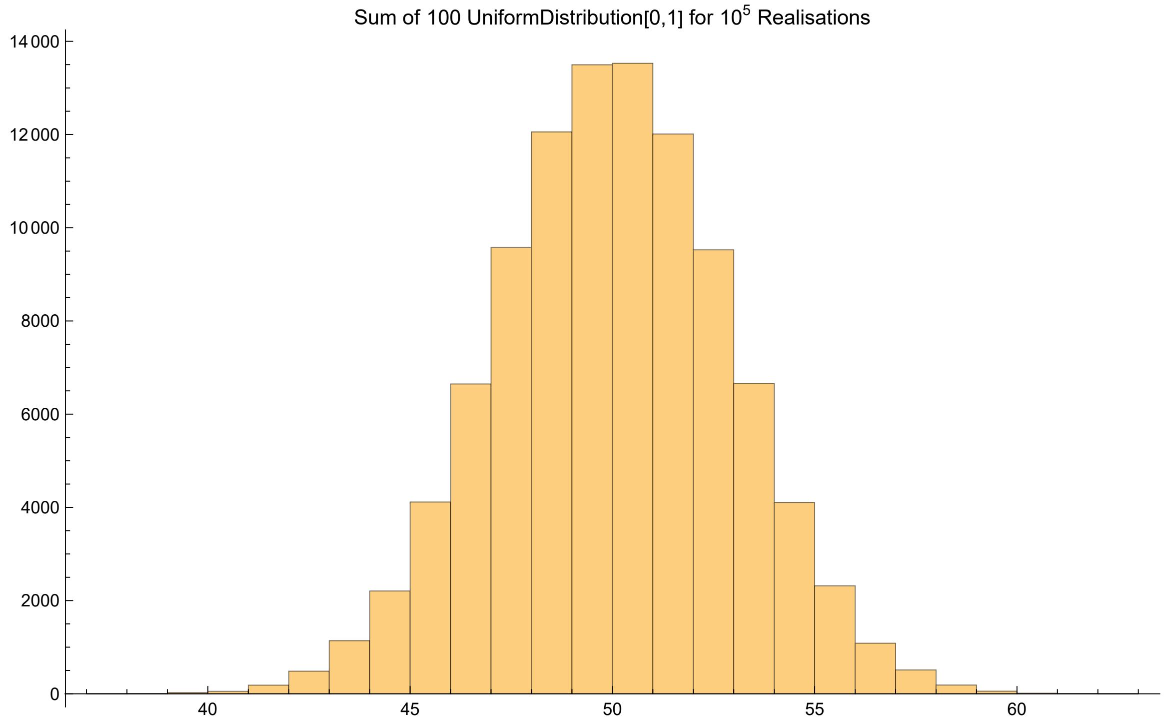
In[88]:=
```
distpareto := RandomVariate[ParetoDistribution[1, 2]];
Histogram[Table[Table[distpareto, {100}] // Total, {10^5}], PlotLabel →
  "Sum of 100 UniformDistribution[1,2] for 10^5 Realisations", ImageSize → 600]
```
Out[89]=




The central limit theorem fail here simply because the number of realisations for the variable is not infinity. This is pre-asymptotics. Note that the sum of the uniform distribution quickly looks like the Gaussian in contrast. This motivates the study of fat tailed distributions like the Pareto distribution. Repeat the same logic for the uniform distribution.

In[86]:=
```
distuniform := RandomVariate[UniformDistribution[]];
Histogram[Table[Table[distuniform, {100}] // Total, {10^5}], PlotLabel →
  "Sum of 100 UniformDistribution[0,1] for 10^5 Realisations", ImageSize → 600]
```

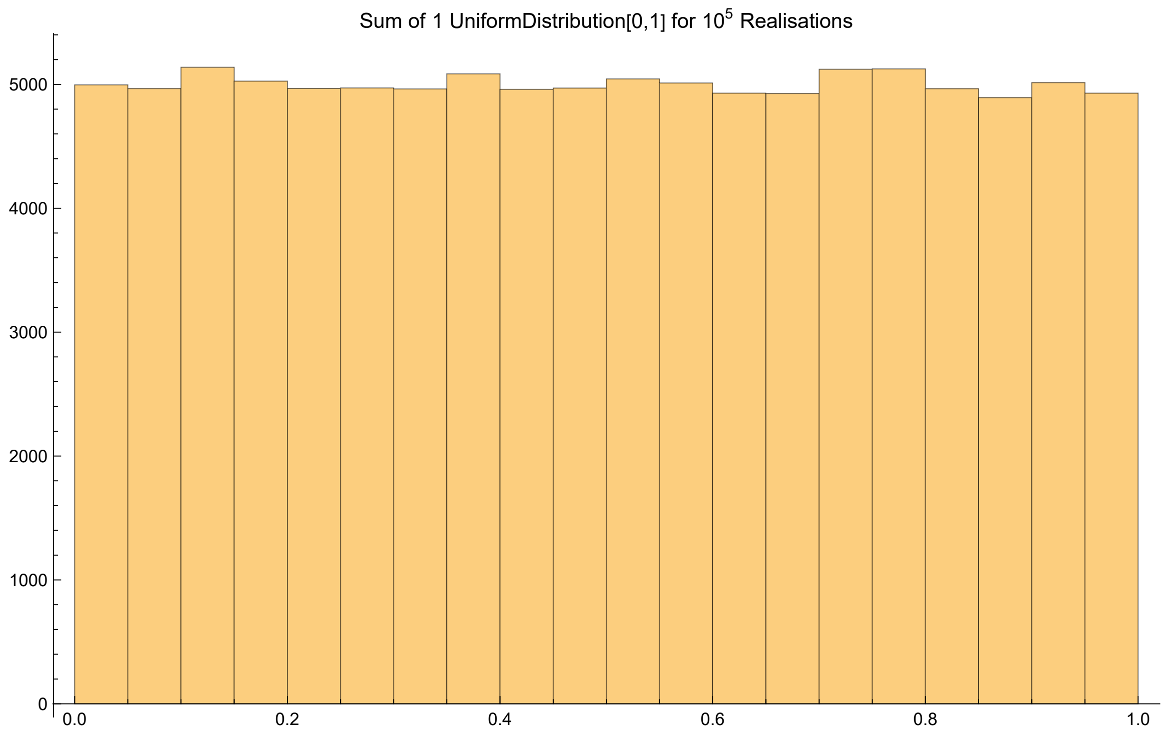
Out[87]=




Here is the how the uniform distributions looks closer and closer to the normal distribution.

In[85]:=
```
Histogram[Table[distuniform, {10^5}],
 PlotLabel → "Sum of 1 UniformDistribution[0,1] for 10^5 Realisations", ImageSize → 600]
```

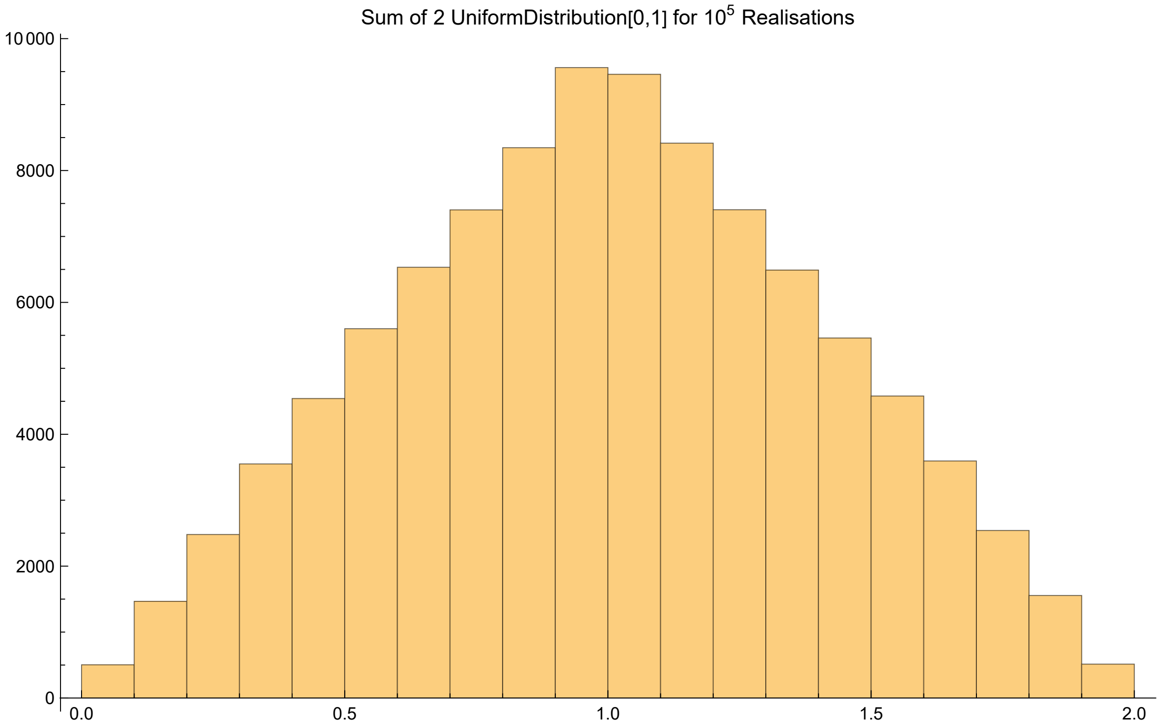
Out[85]=



It looks like a triangle when we take the sum twice.

In[84]:=
```
Histogram[Table[Sum[distuniform, {i, 1, 2}], {10^5}],
 PlotLabel → "Sum of 2 UniformDistribution[0,1] for 10^5 Realisations", ImageSize → 600]
```

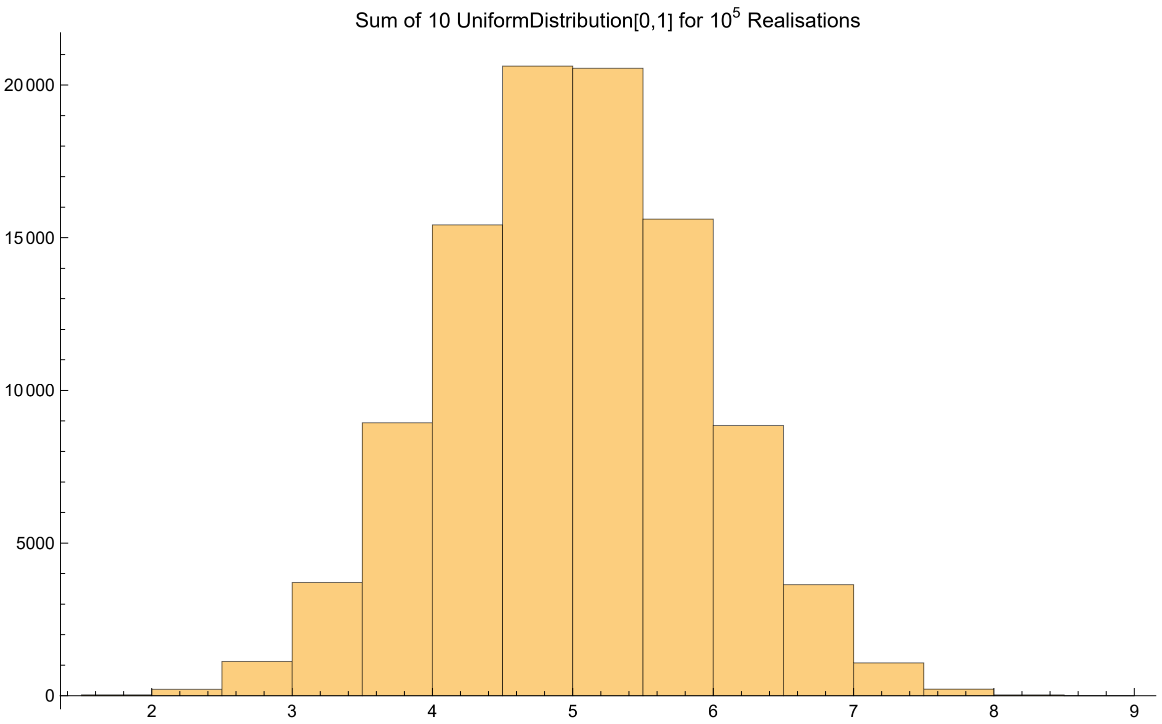
Out[84]=




Now we take this sum 10 times, and it already looks fairly close to a normal distribution.

In[83]:=
```
Histogram[Table[Sum[distuniform, {i, 1, 10}], {10^5}], PlotLabel →
  "Sum of 10 UniformDistribution[0,1] for 10^5 Realisations", ImageSize → 600]
```

Out[83]=



## Application of the Uniform Distribution: Miles Per Annum cannot be uniform

Dan Ariely conducted a study on the distribution of miles driven. The standard deviation was unusually high, with it being half the mean. This must be (roughly) a uniform distribution, however miles per annum cannot be uniform since they are cumulative.

One can reverse engineer to get the left and right bounds of the uniform distribution obtained using Mathematica. Before this, here are some quick reminders on the properties of the uniform distribution.

In[107]:=

```
? UniformDistribution
```

Out[107]=

Symbol

UniformDistribution[{*min*, *max*}] represents a continuous uniform statistical distribution giving values between *min* and *max*.

UniformDistribution[] represents a uniform distribution giving values between 0 and 1.

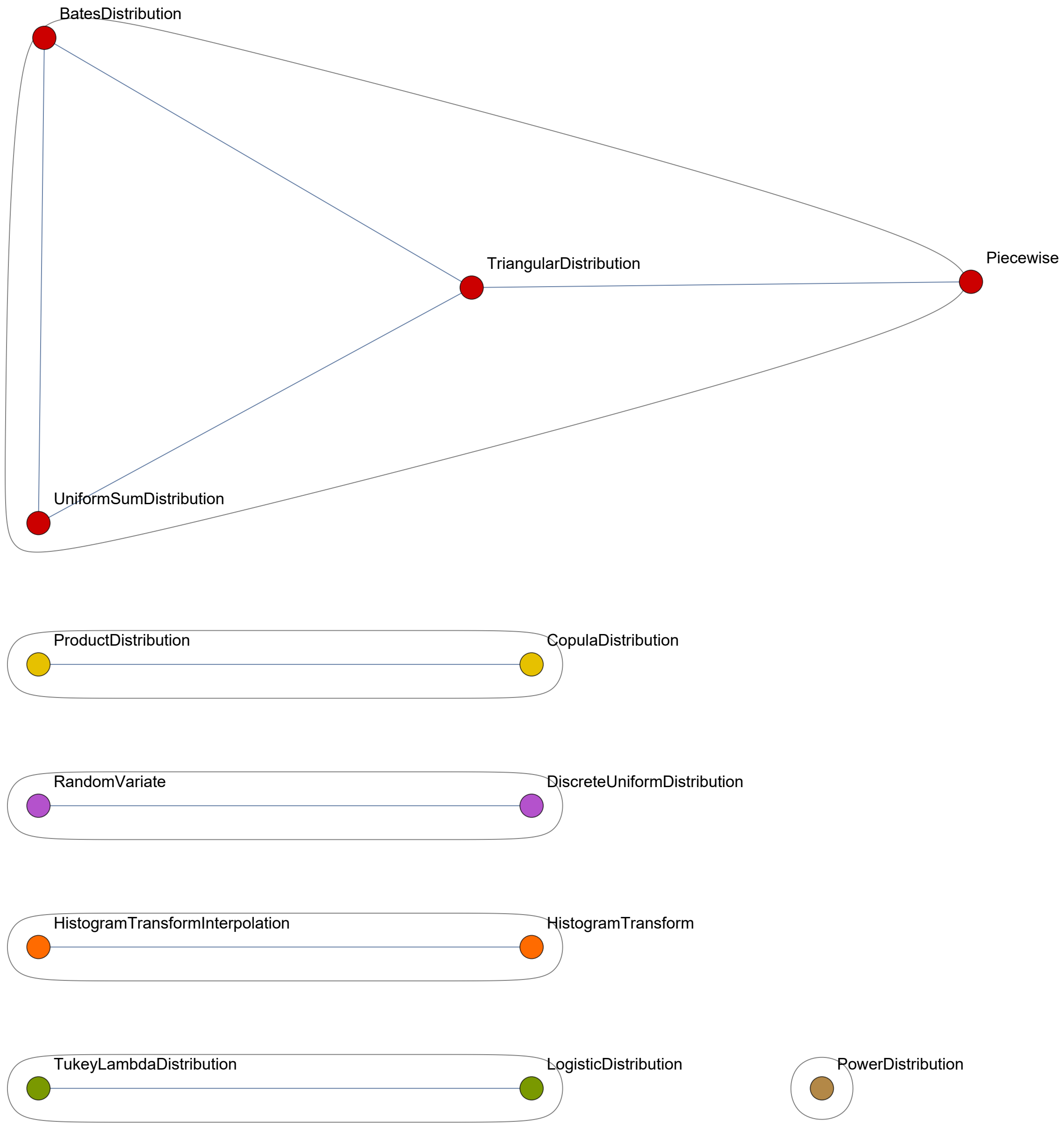
UniformDistribution[{{$x_{min}$, $x_{max}$}, {$y_{min}$, $y_{max}$}, …}] represents a multivariate uniform distribution over the region {{$x_{min}$, $x_{max}$}, {$y_{min}$, $y_{max}$}, …}.

UniformDistribution[$n$] represents a multivariate uniform distribution over the standard $n$ dimensional unit hypercube.

In[108]:=

```
Show[WolframLanguageData["UniformDistribution",
  "RelationshipCommunityGraph"], ImageSize → 600]
```

Out[108]=




In[109]:=

```
Mean[UniformDistribution[{a, b}]]
```

Out[109]=

$$\frac{a + b}{2}$$

In[110]:=

```
StandardDeviation[UniformDistribution[{a, b}]]
```

Out[110]=

$$\frac{-a + b}{2\sqrt{3}}$$

Now we can attempt to reverse engineer the fabricated (approximately) uniform distribution given the mean and the standard deviation.

In[111]:=

```
uniformsolve1 = Solve[{(a + b)/2 == 23670, (b - a)/(2 Sqrt[3]) == 12621}, {a, b}] // Flatten // N
```

$$\text{uniformsolve1} = \text{Solve}\left[\left\{\frac{a+b}{2} == 23\,670,\ \frac{b-a}{2\sqrt{3}} == 12\,621\right\},\ \{a,\ b\}\right] \mathbin{//} \text{Flatten} \mathbin{//} \text{N}$$

Out[111]=

$\{a \to 1809.79,\ b \to 45\,530.2\}$

Of course, you can do this with other distributions.

In[112]:=

$$\text{uniformsolve2} = \text{Solve}\left[\left\{\frac{a+b}{2} == 26\,094,\ \frac{b-a}{2\sqrt{3}} == 12\,253\right\},\ \{a,\ b\}\right] \mathbin{//} \text{Flatten} \mathbin{//} \text{N}$$

Out[112]=

$\{a \to 4871.18,\ b \to 47\,316.8\}$

In[123]:=

```
? UniformDistribution
```

Out[123]=

Symbol

UniformDistribution[{*min*, *max*}] represents a continuous uniform statistical distribution giving values between *min* and *max*.

UniformDistribution[] represents a uniform distribution giving values between 0 and 1.

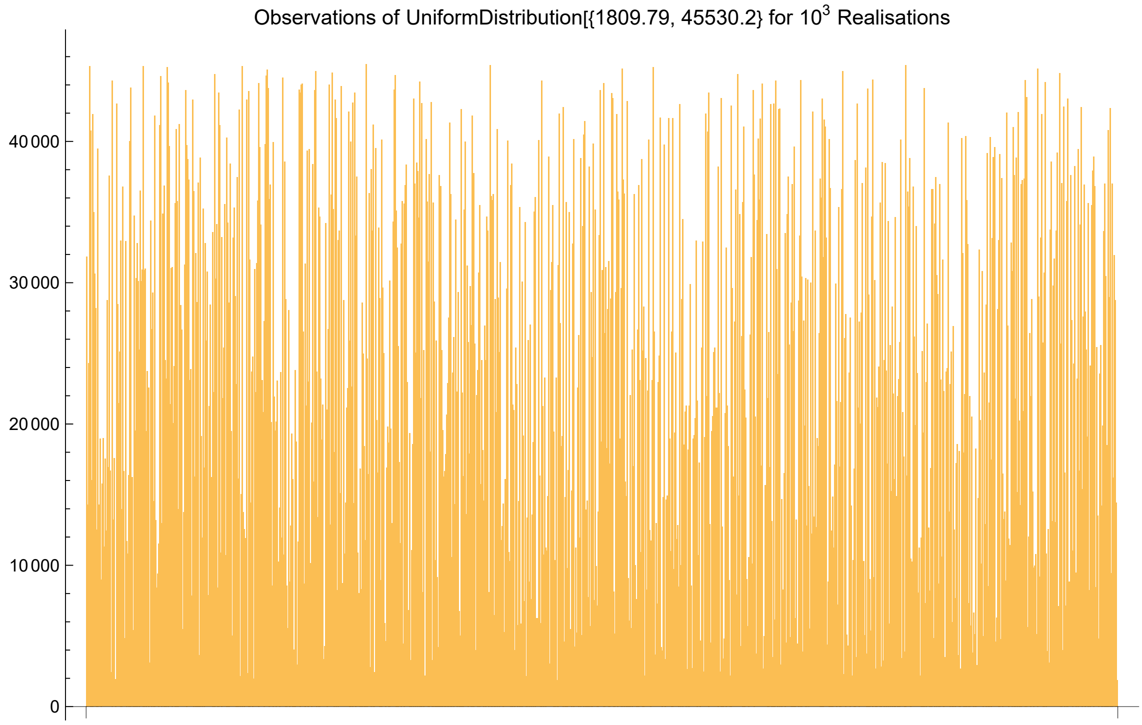
UniformDistribution[{{$x_{min}$, $x_{max}$}, {$y_{min}$, $y_{max}$}, …}] represents a multivariate uniform distribution over the region {{$x_{min}$, $x_{max}$}, {$y_{min}$, $y_{max}$}, …}.

UniformDistribution[*n*] represents a multivariate uniform distribution over the standard *n* dimensional unit hypercube.

In[128]:=

```
BarChart[
 Table[RandomVariate[UniformDistribution[{1809.79, 45530.2}]], {1000}], PlotLabel →
  "Observations of UniformDistribution[{1809.79, 45530.2} for 10^3 Realisations",
 ImageSize → 600]
```

Out[128]=




# The Pareto distribution, and the law of large numbers

The power law class is very interesting, somehow it does not get to a steady value with increasing number of realisations.

In[71]:=

```
?ParetoDistribution
```

Out[71]=

Symbol

ParetoDistribution[$k$, $\alpha$] represents a Pareto distribution with minimum value parameter $k$ and shape parameter $\alpha$.

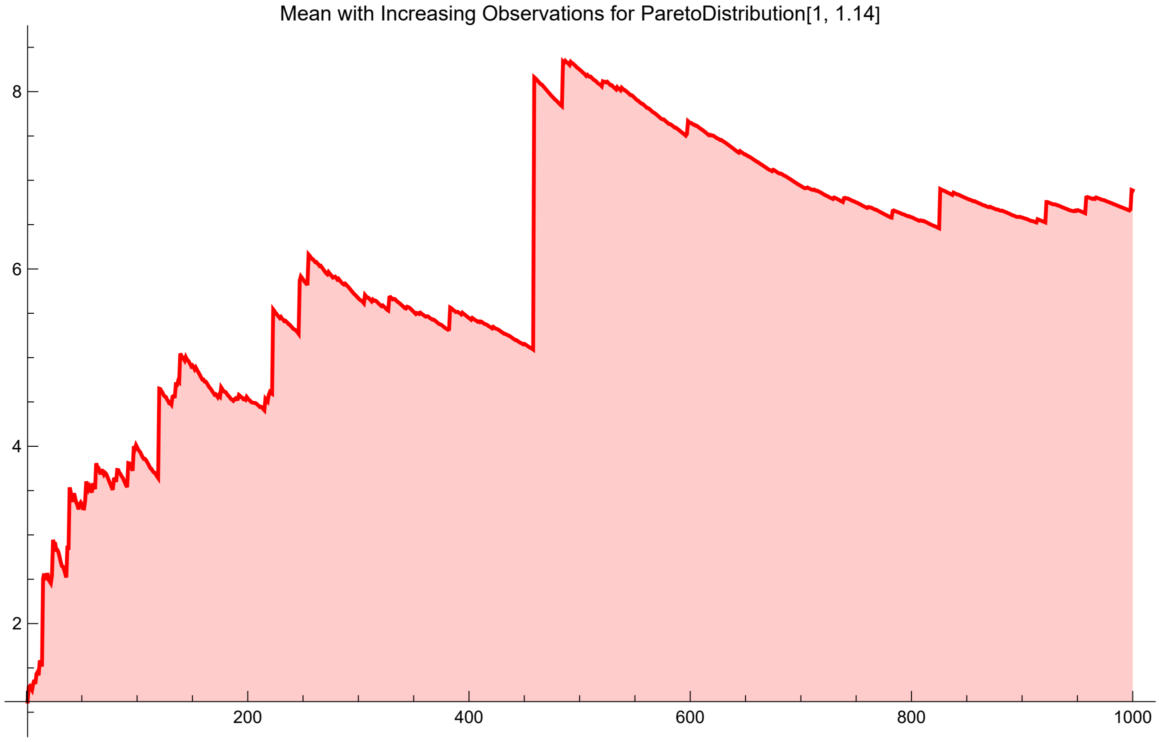
ParetoDistribution[$k$, $\alpha$, $\mu$] represents a Pareto type II distribution with location parameter $\mu$.

ParetoDistribution[$k$, $\alpha$, $\gamma$, $\mu$] represents a Pareto type IV distribution with shape parameter $\gamma$.

I found it easy to just use this to copy paste the definitions here so that it is easy to compare and refer to.

In[77]:=
```
distpareto2 := RandomVariate[ParetoDistribution[1, 1.14]];
tapareto2 = Table[distpareto2, {1000}];
DiscretePlot[Mean[tapareto2〚1 ;; i〛], {i, 1, Length[tapareto2]},
 PlotStyle → Red, PlotRange → All, ImageSize → 600,
 PlotLabel → "Mean with Increasing Observations for ParetoDistribution[1, 1.14]"]
```

Out[79]=




Compare this to a normal distribution. This one settles at the mean very quickly.

In[72]:=
```
? NormalDistribution
```

Out[72]=

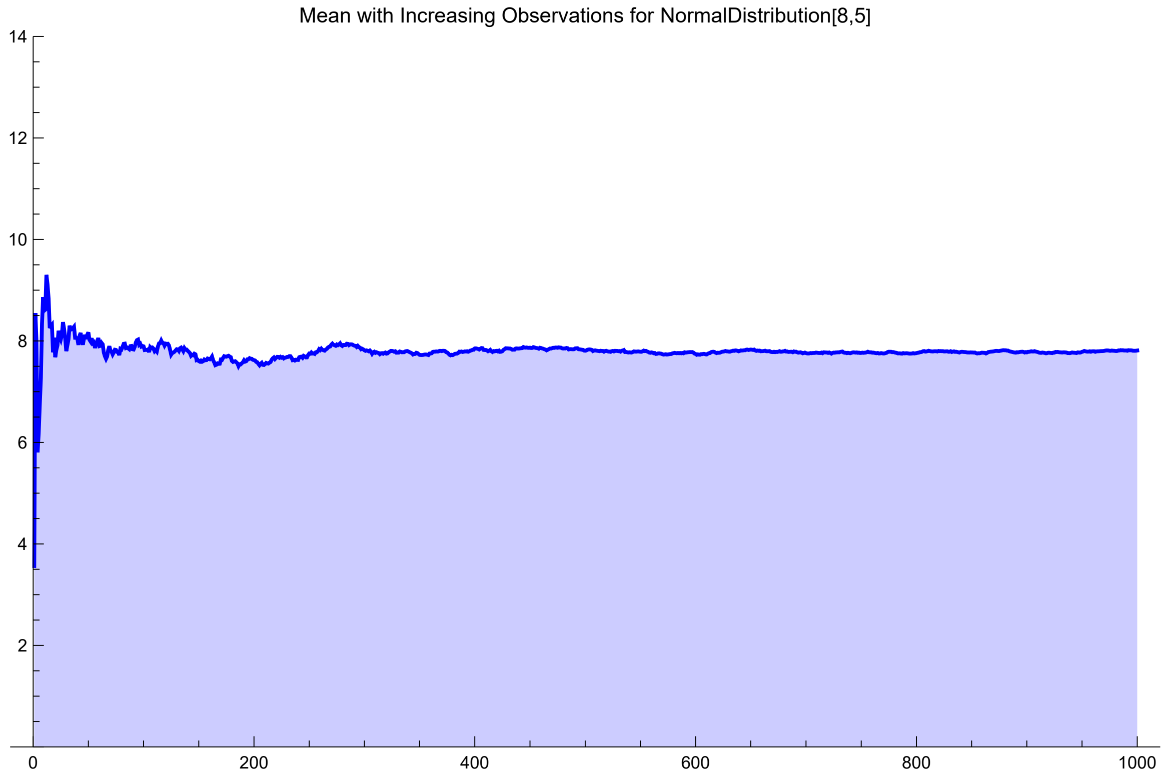
Symbol

NormalDistribution[$\mu$, $\sigma$] represents a normal (Gaussian) distribution with mean $\mu$ and standard deviation $\sigma$.

NormalDistribution[] represents a normal distribution with zero mean and unit standard deviation.

In[80]:=
```
distnorm2 := RandomVariate[NormalDistribution[8, 5]];
tanorm2 = Table[distnorm2, {1000}];
DiscretePlot[Mean[tanorm2[[1 ;; i]]], {i, 1, Length[tanorm2]},
 PlotStyle → Blue, PlotRange → {0, 14}, ImageSize → 600,
 PlotLabel → "Mean with Increasing Observations for NormalDistribution[8,5]"]
```

Out[82]=

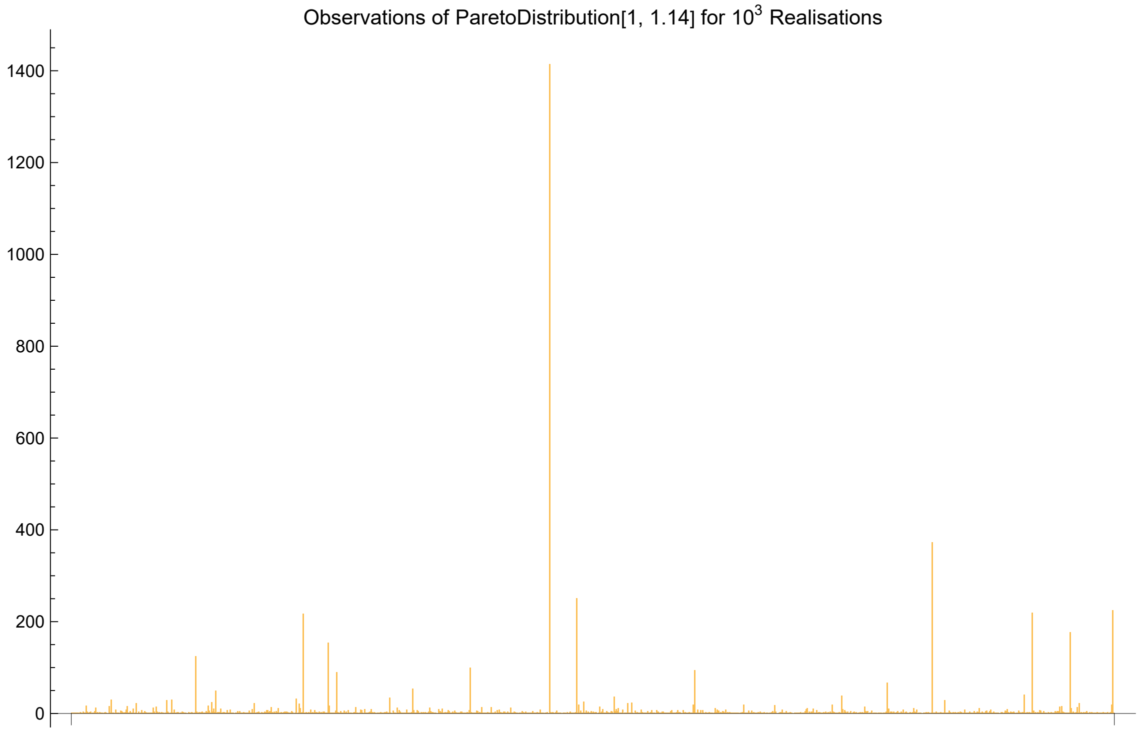



We can compare the bar charts of these two tables.

In[129]:=

```
BarChart[tapareto2, PlotLabel →
  "Observations of ParetoDistribution[1, 1.14] for 10^3 Realisations", ImageSize → 600]
```

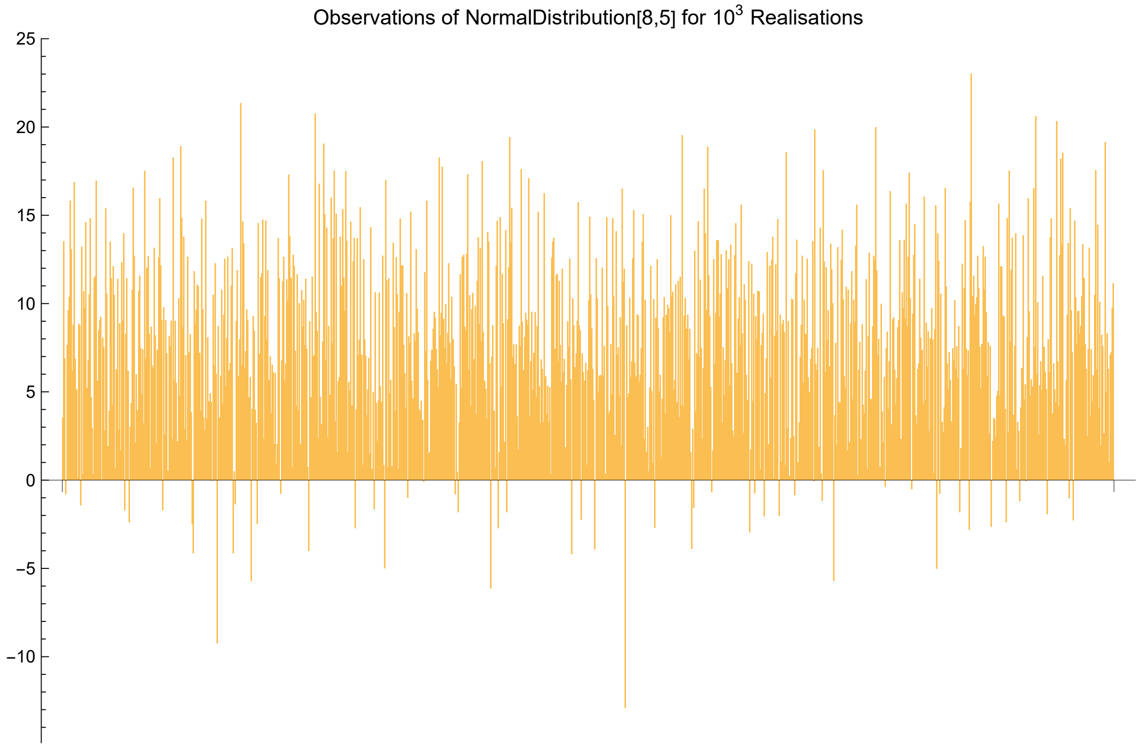
Out[129]=




In[130]:=

```
BarChart[tanorm2, PlotLabel →
  "Observations of NormalDistribution[8,5] for 10^3 Realisations", ImageSize → 600]
```

Out[130]=



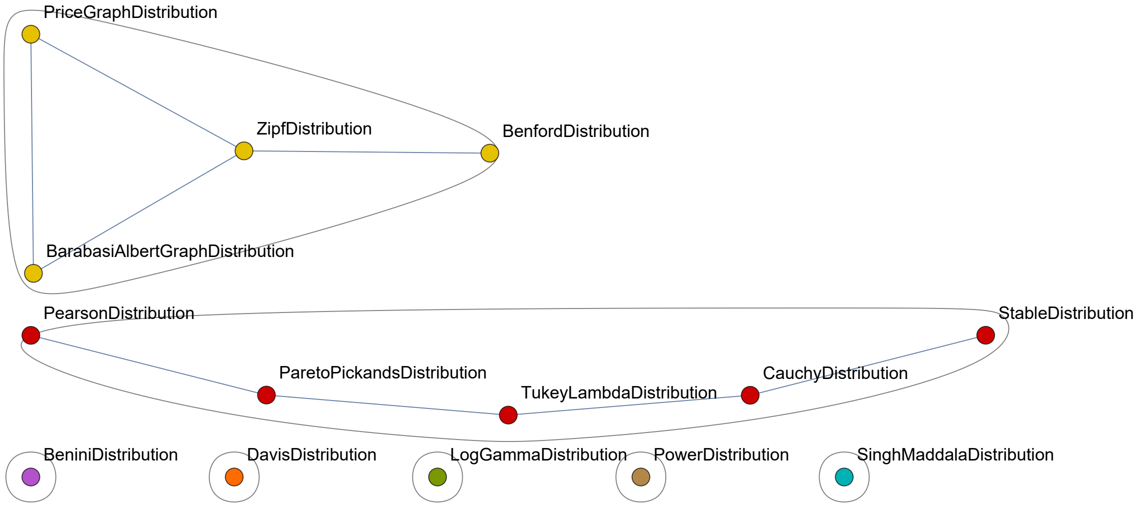

We can see how the various distributions are related to the Pareto distribution based on this. In fact,t he next of our interest is the stable distribution which we will discuss later.

# Looking at the Pareto distribution a bit more carefully

With Mathematica, one can see how the distributions are related. You can consult Wikipedia for more information here: https://en.wikipedia.org/wiki/Pareto_distribution.

In[93]:=
```
Show[WolframLanguageData["ParetoDistribution",
  "RelationshipCommunityGraph"], ImageSize → 600]
```

Out[93]=




In[98]:=
```
? ParetoDistribution
```

Out[98]=

Symbol

ParetoDistribution[$k$, $\alpha$] represents a Pareto

distribution with minimum value parameter $k$ and shape parameter $\alpha$.

ParetoDistribution[$k$, $\alpha$, $\mu$] represents a Pareto type II distribution with location parameter $\mu$.

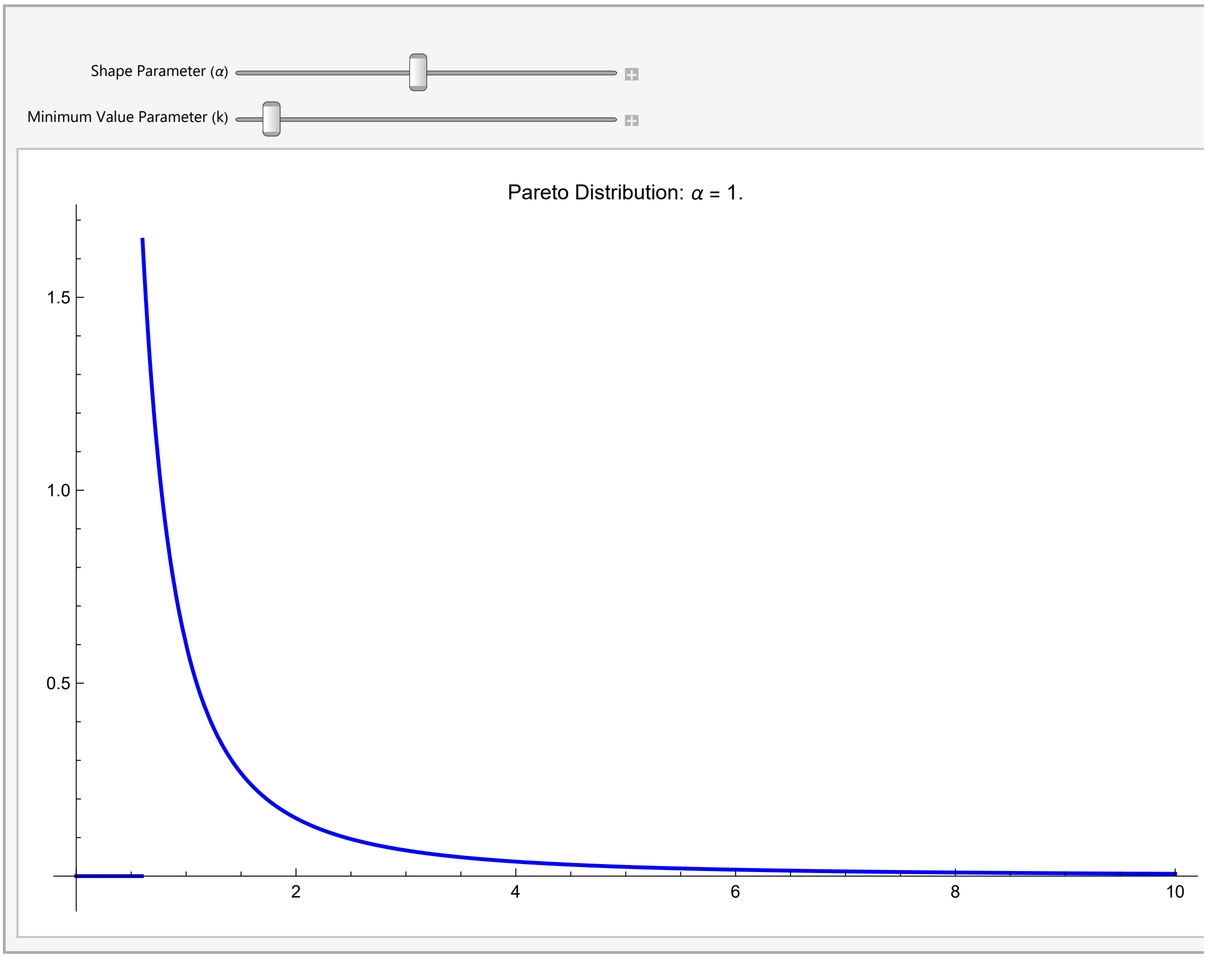
ParetoDistribution[$k$, $\alpha$, $\gamma$, $\mu$] represents a Pareto type IV distribution with shape parameter $\gamma$.

One can see how the parameters change with these parameters. We can set the sliders to change.

In[106]:=

```
Manipulate[Module[{dist, plot}, dist = ParetoDistribution[k, alpha];
  plot = Plot[PDF[dist, x], {x, 0, 10}, PlotRange → All,
    PlotStyle → Blue,
    ImageSize → 600,
    PlotLabel → Row[{"Pareto Distribution: α = ", alpha}]]],
 {{alpha, 1, "Shape Parameter (α)"}, 0.1, 2, 0.1},
 {{k, 0.1, "Minimum Value Parameter (k)"}, 0.1, 10, 0.1}]
```

Out[106]=




## Visualising the effect of power laws

In the power law class, the rare event contributes to most of the properties. One can study this by looking at the inverse survival function of the Pareto distribution. Recall that the survival function is the probability of a variable exceeding a given parameter. There is actually a closed form one can pull out with Mathematica [5].

We solve for the survival function such that K exceeds L where L is the minimum value parameter of the Pareto distribution for a particular shape parameter $\alpha$.

In[135]:=

```
? ParetoDistribution
```

Out[135]=

Symbol

ParetoDistribution[$k$, $\alpha$] represents a Pareto

distribution with minimum value parameter $k$ and shape parameter $\alpha$.

ParetoDistribution[$k$, $\alpha$, $\mu$] represents a Pareto type II distribution with location parameter $\mu$.

ParetoDistribution[$k$, $\alpha$, $\gamma$, $\mu$] represents a Pareto type IV distribution with shape parameter $\gamma$.

In[156]:=

```
solK = Solve[
   FullSimplify[SurvivalFunction[ParetoDistribution[L, α], K], K > L] == p, K] // Quiet
```

Out[156]=

$\{\{K \to L\, p^{-1/\alpha}\}\}$

We can use Mathematica to get a share of a particular quantile given the associated tail exponent.

In[144]:=

```
? PowerExpand
```

Out[144]=

Symbol

PowerExpand[$expr$] expands all powers of products and powers.

PowerExpand[$expr$, {$x_1$, $x_2$, …}] expands only with respect to the variables $x_i$.

We can find the probability of exceeding of a certain threshold to get this expression. We do this by finding the normalised expression using the integral of the value multiplied by its associated PDF for all cases past K towards infinity divided by the mean of the Pareto distribution. This allows us to get the proportion of the threshold of the body required.

In[141]:=

```
threshold =
 Integrate[x PDF [ParetoDistribution[L, α], x], {x, K, ∞}, Assumptions → K > L > α > 1] /
     Simplify[Mean[ParetoDistribution[L, α]], Assumptions → K > L > α > 1] /.
    solK[[1]] // PowerExpand // FullSimplify
```

Out[141]=

$p^{\frac{-1+\alpha}{\alpha}}$

We can for example solve for the tail exponent of the 80/20 rule, where 20% of the people own 80% of the land, as classically defined by Pareto. This gives the tail exponent to be $\alpha = 1.16096$.

In[153]:=

```
Solve[(threshold /. p → 0.2) == 0.8, α] // Quiet
```

Out[153]=

$\{\{\alpha \to 1.16096\}\}$

We can check how much the top 1% own (1/100), we find that the top 1% must own about 52.8% of the land under the Pareto distribution.

In[143]:=

```
threshold /. α → 1.16096 /. p → (1 / 100) // N
```

Out[143]=

```
0.528095
```

In[154]:=

```
TableForm[Table[{a, 1 - a, Solve[(threshold /. p → a) == 1 - a, α]}, {a, 0, 0.4, 0.01}],
  TableHeadings → {None,
    {"Minority Ratio (p)", "Majority ratio (1-p)" , "Tail exponent (α)"}}] // Quiet
```

Out[154]//TableForm=

| Minority Ratio (p) | Majority ratio (1–p) | Tail exponent (α) |
|---|---|---|
| 0. | 1. | |
| 0.01 | 0.99 | $\alpha \to$ 1.00219 |
| 0.02 | 0.98 | $\alpha \to$ 1.00519 |
| 0.03 | 0.97 | $\alpha \to$ 1.00876 |
| 0.04 | 0.96 | $\alpha \to$ 1.01284 |
| 0.05 | 0.95 | $\alpha \to$ 1.01742 |
| 0.06 | 0.94 | $\alpha \to$ 1.02249 |
| 0.07 | 0.93 | $\alpha \to$ 1.02806 |
| 0.08 | 0.92 | $\alpha \to$ 1.03414 |
| 0.09 | 0.91 | $\alpha \to$ 1.04076 |
| 0.1 | 0.9 | $\alpha \to$ 1.04795 |
| 0.11 | 0.89 | $\alpha \to$ 1.05574 |
| 0.12 | 0.88 | $\alpha \to$ 1.06416 |
| 0.13 | 0.87 | $\alpha \to$ 1.07326 |
| 0.14 | 0.86 | $\alpha \to$ 1.08308 |
| 0.15 | 0.85 | $\alpha \to$ 1.09369 |
| 0.16 | 0.84 | $\alpha \to$ 1.10514 |
| 0.17 | 0.83 | $\alpha \to$ 1.11751 |
| 0.18 | 0.82 | $\alpha \to$ 1.13087 |
| 0.19 | 0.81 | $\alpha \to$ 1.14532 |
| 0.2 | 0.8 | $\alpha \to$ 1.16096 |
| 0.21 | 0.79 | $\alpha \to$ 1.17791 |
| 0.22 | 0.78 | $\alpha \to$ 1.19631 |
| 0.23 | 0.77 | $\alpha \to$ 1.21631 |
| 0.24 | 0.76 | $\alpha \to$ 1.23809 |
| 0.25 | 0.75 | $\alpha \to$ 1.26186 |
| 0.26 | 0.74 | $\alpha \to$ 1.28787 |
| 0.27 | 0.73 | $\alpha \to$ 1.31641 |
| 0.28 | 0.72 | $\alpha \to$ 1.34782 |
| 0.29 | 0.71 | $\alpha \to$ 1.38251 |
| 0.3 | 0.7 | $\alpha \to$ 1.42096 |
| 0.31 | 0.69 | $\alpha \to$ 1.46376 |
| 0.32 | 0.68 | $\alpha \to$ 1.51164 |
| 0.33 | 0.67 | $\alpha \to$ 1.5655 |
| 0.34 | 0.66 | $\alpha \to$ 1.62644 |
| 0.35 | 0.65 | $\alpha \to$ 1.69589 |
| 0.36 | 0.64 | $\alpha \to$ 1.77566 |
| 0.37 | 0.63 | $\alpha \to$ 1.86813 |
| 0.38 | 0.62 | $\alpha \to$ 1.97648 |
| 0.39 | 0.61 | $\alpha \to$ 2.10504 |
| 0.4 | 0.6 | $\alpha \to$ 2.25985 |

Of course we could be a lot more strict. However with this logic of minority ratio and majority ratio being such that the sum is 1, the tail exponent does not seem to go below 1. This shows that errors in the tail exponent can lead to very serious mistakes.

In[155]:=

```
TableForm[Table[{a, 1 - a, Solve[(threshold /. p → a) == 1 - a, α]}, {a, 0, 0.01, 0.001}],
  TableHeadings → {None,
    {"Minority Ratio (p)", "Majority ratio (1-p)" , "Tail exponent (α)"}}] // Quiet
```

Out[155]//TableForm=

| Minority Ratio (p) | Majority ratio (1-p) | Tail exponent (α) |
|---|---|---|
| 0. | 1. | |
| 0.001 | 0.999 | α → 1.00014 |
| 0.002 | 0.998 | α → 1.00032 |
| 0.003 | 0.997 | α → 1.00052 |
| 0.004 | 0.996 | α → 1.00073 |
| 0.005 | 0.995 | α → 1.00095 |
| 0.006 | 0.994 | α → 1.00118 |
| 0.007 | 0.993 | α → 1.00142 |
| 0.008 | 0.992 | α → 1.00167 |
| 0.009 | 0.991 | α → 1.00192 |
| 0.01 | 0.99 | α → 1.00219 |

# References

Please forgive me if I did not reference a particular Tweet from Nassim Taleb.

[1] Nassim Nicholas Taleb. *Statistical Consequences of Fat Tails: Revised Edition* (2022). Accessed: Jan. 2, 2024. Available: https://arxiv.org/pdf/2001.10488.pdf

[2] Nassim Nicholas Taleb. *Doing Probability with Mathematica* (2019). Accessed: Jan. 2, 2024. Available: https://www.youtube.com/watch?v=JTz4wkD-mxU

[3] Nassim Nicholas Taleb. *Disinformation and Fooled By Randomness (2023)*. Accessed. Jan. 2. 2024. Available: https://www.youtube.com/watch?app=desktop&v=T3ScFshhhnU

[4] Nassim Nicholas Taleb. *Comments on Ariely Honesty Paper (2021)*. Accessed. Jan. 3. 2024. Available: https://twitter.com/nntaleb/status/1429295773938814978?lang=en

[5] Nassim Nicholas Taleb. *MINI LECTURE 8c: A simple tricks to see the effect of power laws (2021)*. Accessed. Jan. 3. 2024. Available: https://www.youtube.com/watch?v=XhTHG3QmVwM