DYALOG
2021

QA for Statistical Distributions in Dyalog V18.2

Ron Murray

First, Some Context...

- The code I'll be running is in https://github.com/RonM-Dyalog/DistributionsDemo.git

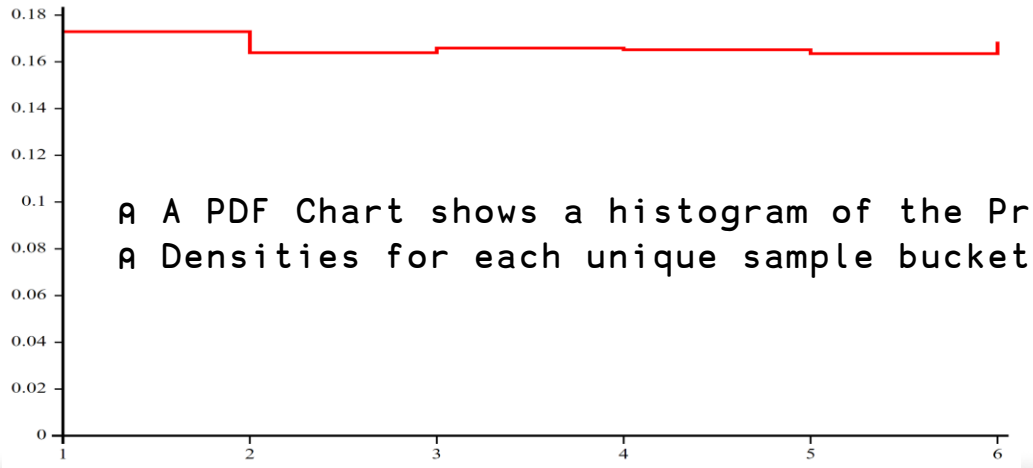
- The Samples from R are provided by Kimmo Linna's GitHub Repository at https://github.com/kimmolinna/rsconnect.git

Dyalog is considering retiring the distributed "rconnect" workspace in favor of Kimmo's open-source community project for accessing R.

In the beginning we had ?

Monadic ? gave us uniform sampling with replacement:

```
      ?10⍴6
5 3 5 2 1 2 3 2 6 5

      PDFStepChart ↓⍉⍪⍳6,[1.5]+/(÷1E4)×(⍳6)∘.=?1E4⍴6
1 0.1729
2 0.1639
3 0.1659
4 0.1652
5 0.1635
6 0.1686
```




```
      ⍝ A PDF Chart shows a histogram of the Probability
      ⍝ Densities for each unique sample bucket.
```

Version 14 Added ?0

- For floating point random numbers uniformly distributed between 0 and 1:

```
      3 3⍴?9⍴0
0.7362027295 0.4042132435 0.08190525355
0.8256649279 0.3752891013 0.3504641678
0.5020267581 0.939049347  0.1225829509
```

Version 18 Added More Distributions

- (16808I) Provides 16 additional distributions:

Beta	Binomial	Cauchy	Chi Squared
Exponential	F	Gamma	Inverse Gamma
Laplace	Log Normal	Logistic	Normal
Poisson	Student T	Uniform	Weibull

Sampling from the new distributions follows a common pattern

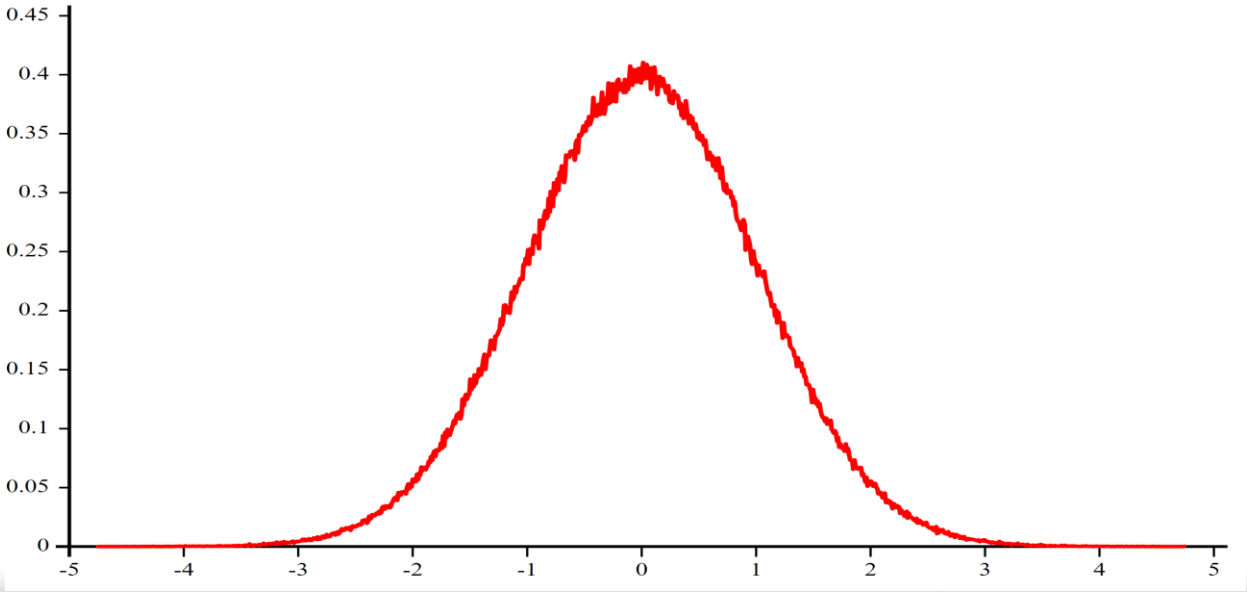
```
samples←ControlParams (16808⌶) DistributionName ResultShape
```

To get 1000000 random numbers from the Normal distribution with a mean of 0 and standard deviation of 1, you'd execute:

```
      rv← 0 1 (16808⌶) 'Normal' 1E6
      ⍴rv
1000000
      {((+/÷≢)⍵),(⌊/,⌈/)⍵} rv
0.0003315298649 ¯4.503307821 5.407314722
```

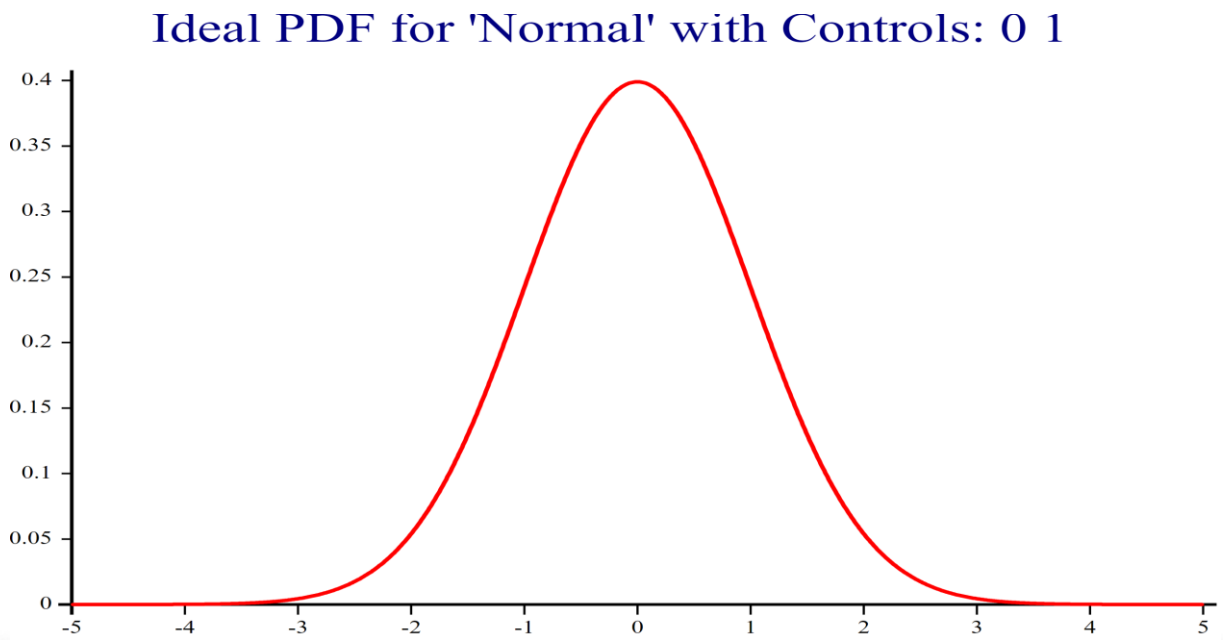
Here's the PDF for those Normal Samples

```
0 1 ShowActualPDF 'Normal'
```



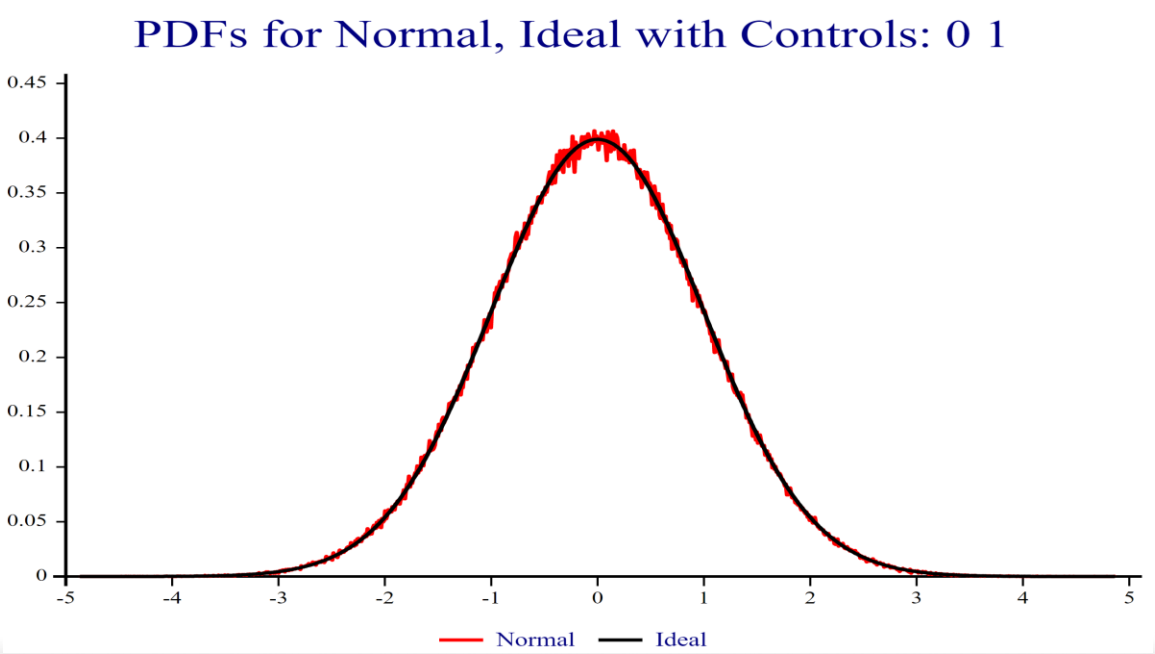
With infinite samples we should see…

```
0 1 ShowIdealPDF 'Normal'
```



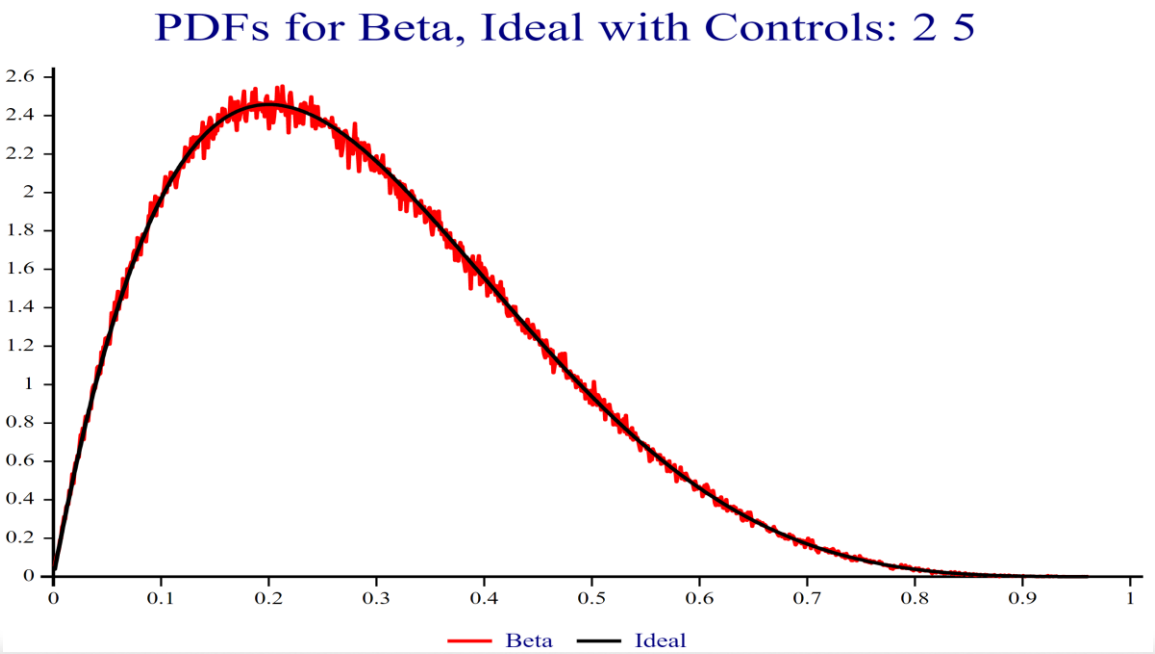
Are the two graphs “similar” enough?

```
0 1 ShowPDFs 'Normal'
```



How about Beta Distribution Samples?

```
2 5 ShowPDFs 'Beta'
```



By Eyeball they look close…

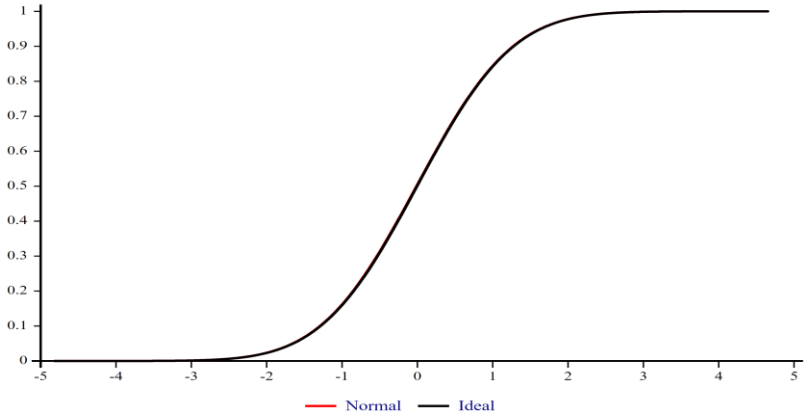
- But we need a more exact numerical approach for algorithm quality control.
- Two Russian mathematicians found a good way to measure the difference between distributions.
  - Andrey Kolmogorov
  - Nikolai Smirnov.

The Kolmogorov-Smirnov Statistic

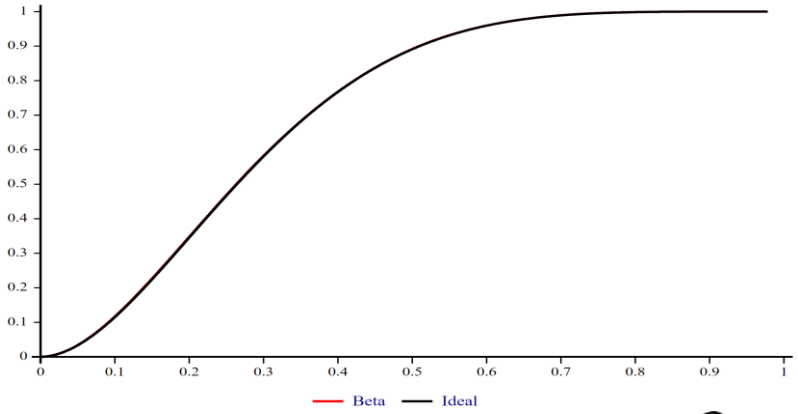
Instead of looking at PDF graphs, they examine normalized cumulative distributions.

```
NCD←{cumm←+\⍵ ◇ cumm÷¯1↑cumm} ⍝ CDF from PDF!
```

```
0 1 ShowCDFs 'Normal'
```




```
2 5 ShowCDFs 'Beta'
```



We can calculate a KS statistic by

```
NCD←{cumm←+\⍵ ◇ cumm÷¯1↑cumm}

IdealCDF←NCD IdealPDF

ActualCDF←NCD ActualPDF

KS_Statistic←⌈/|ActualCDF - IdealCDF
```

For the two examples we saw

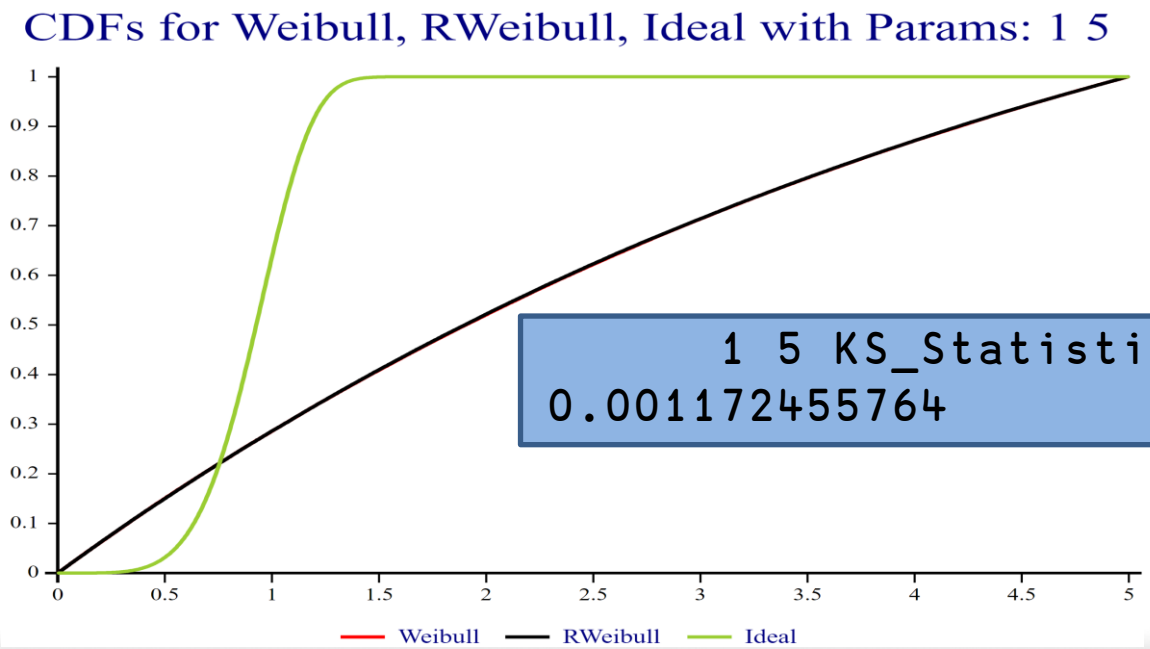
```
      0 1 KS_Statistic 'Normal'
0.002237780068

      2 5 KS_Statistic 'Beta'
0.001561492869
```

The KS Statistic is our QA Test!

- For all but one of our new distributions we have good KS agreement between actual and ideal results.
- The outlier is the Weibull distribution.
  - Our results are a close match for the Weibull results from R!
  - Perhaps our Ideal assumptions aren't correct.

Let's look at CDFs for Weibull, RWeibull, Ideal

```
1 5 ShowCDFs 'Weibull' 'RWeibull'
```

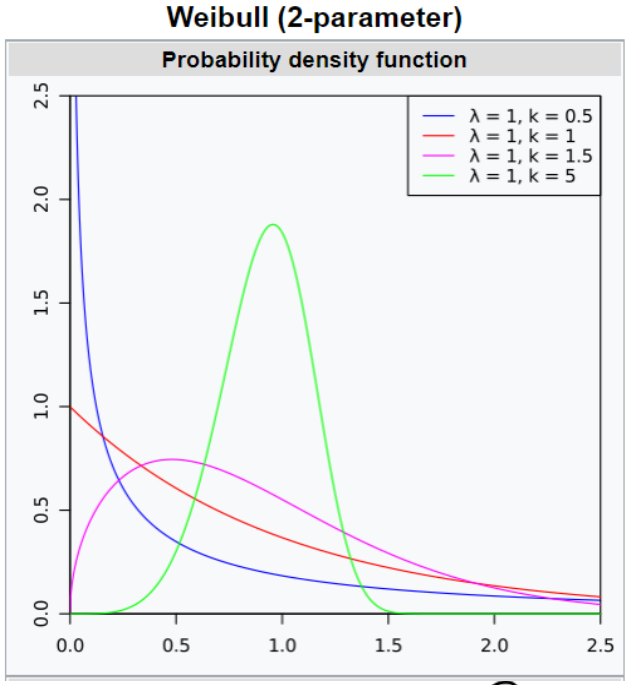



```
      1 5 KS_Statistic 'Weibull' 'RWeibull'
0.001172455764
```

Clearly our code for the PDF of the ideal Weibull distribution is not correct…

```
      IdealPDF←{
(a b)←VerifyControlParameters ⍺
(⍵≥0)×(b÷a)×((b-1)*¨⍵÷a)××-b*¨⍵÷a
}
⍝ Constructed from the Wikipedia Specification:
```

Parameters	$\lambda \in (0, +\infty)$ scale $k \in (0, +\infty)$ shape
Support	$x \in [0, +\infty)$
PDF	$f(x) = \begin{cases} \frac{k}{\lambda}\left(\frac{x}{\lambda}\right)^{k-1} e^{-(x/\lambda)^k} & x \geq 0 \\ 0 & x < 0 \end{cases}$



What is the correct specification?

One experiment looked promising

```
      IdealPDF←{
(a b)←VerifyControlParameters ⍺
(⍵≥0)×(b÷a)×((b-1)*⍨⍵÷a)××-b*⍨⍵÷a
}

      1 .5  KS_Statistic 'Weibull'
0.0007189751945
      1 1   KS_Statistic 'Weibull'
0.006767136088
      1 1.5 KS_Statistic 'Weibull'
0.03569322736
      1 5   KS_Statistic 'Weibull'
0.3679324294

⍝ But the last two results were very problematic.
```

Our Weibull seems to match RWeibull!

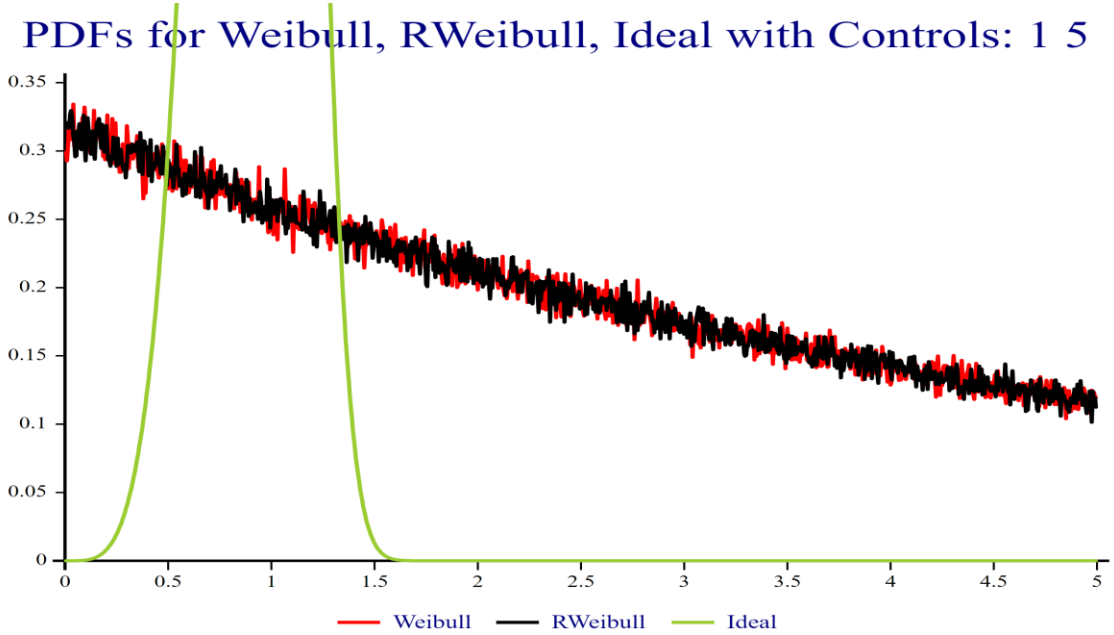
```
      1 .5  KS_Statistic 'Weibull' 'RWeibull'
0.0006962655484
      1 1   KS_Statistic 'Weibull' 'RWeibull'
0.001506052
      1 1.5 KS_Statistic 'Weibull' 'RWeibull'
0.001436363446
      1 5   KS_Statistic 'Weibull' 'RWeibull'
0.001172455764

⍝ So if our Weibull distribution has a bug,
⍝ Then so does R's Weibull distribution!
```

Start of unused slides

- The remaining slide(s) were not used for the presentation.

Let's look at the PDFs for Weibull, Rweibull, and Ideal

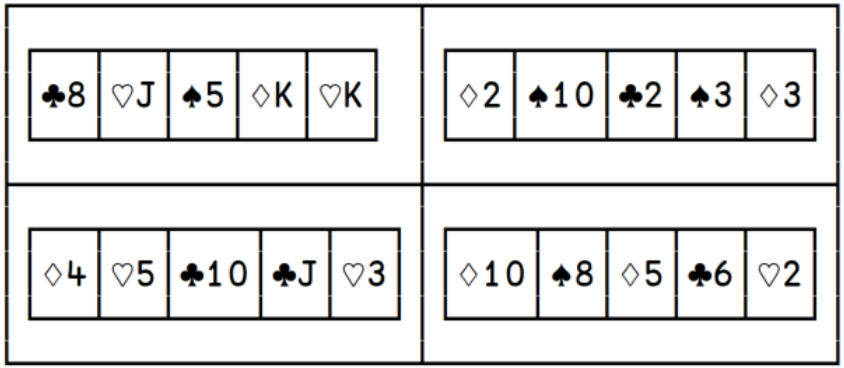
```
1 5 ShowPDFs 'Weibull' 'RWeibull'
```



In the beginning we had ?

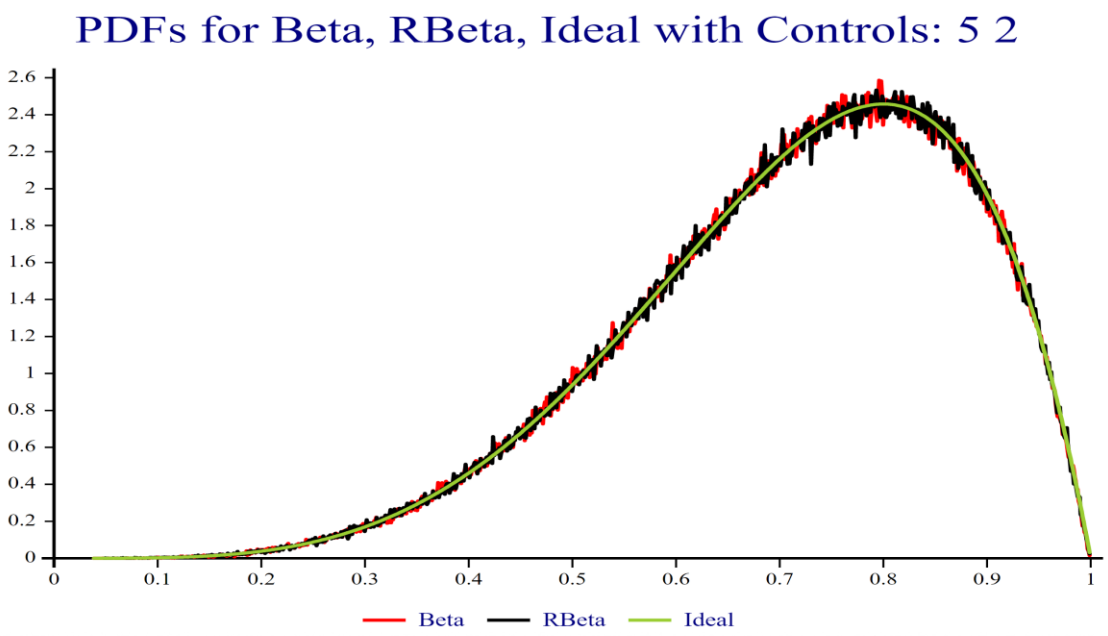
Dyadic ? which gave Sampling us without Replacement:

```
Suite←⎕UCS (16⊥2 6 5 15)+⍳4
Rank←(⊆⍨'A23456789JQK'),⊂'10'
2 2⍴⊂[2]4 5⍴(,Suites∘.,Rank)[52?52]
```



Here are PDFs for Beta, RBeta, and Ideal

```
5 2 ShowPDFs 'Beta' 'RBeta'
```



Here are CDFs for Beta, RBeta, and Ideal

```
5 2 ShowCDFs 'Beta' 'RBeta'
```