---

author: "Sanjana Suresh" output: pdf_document: default html_document: df_print: paged —

```
library(fitdistrplus)
```

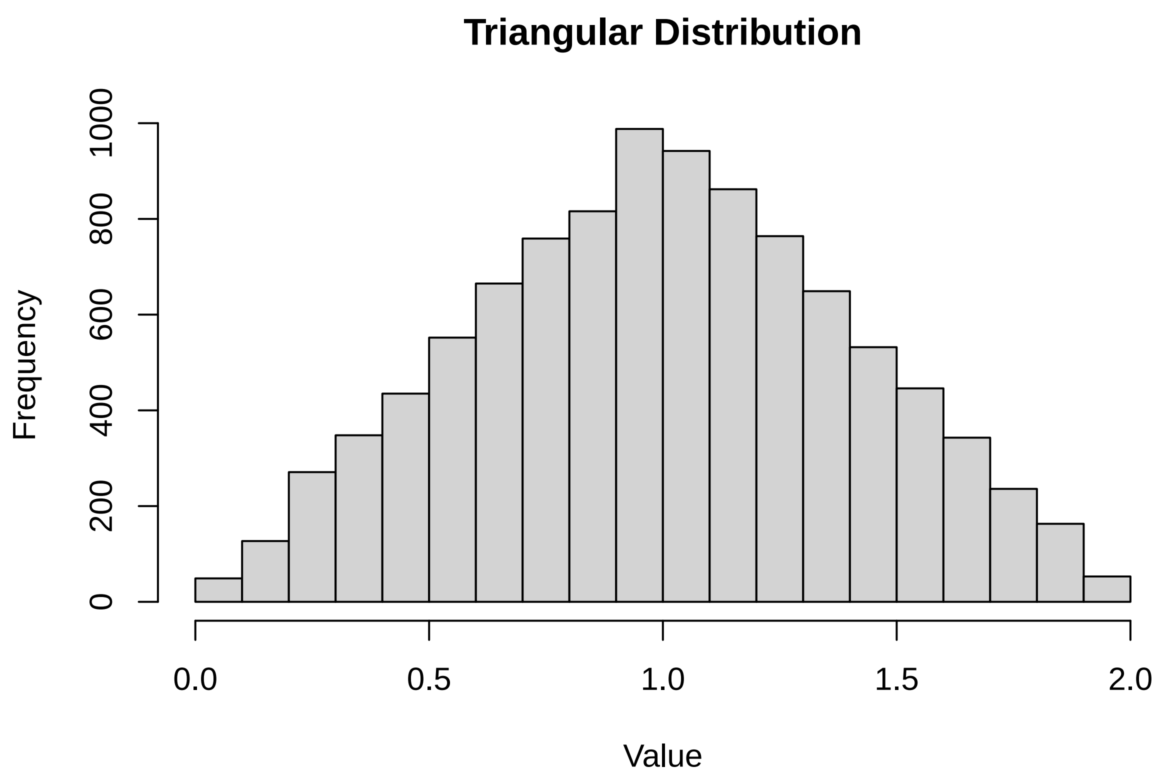
```
## Loading required package: MASS
```

```
## Loading required package: survival
```

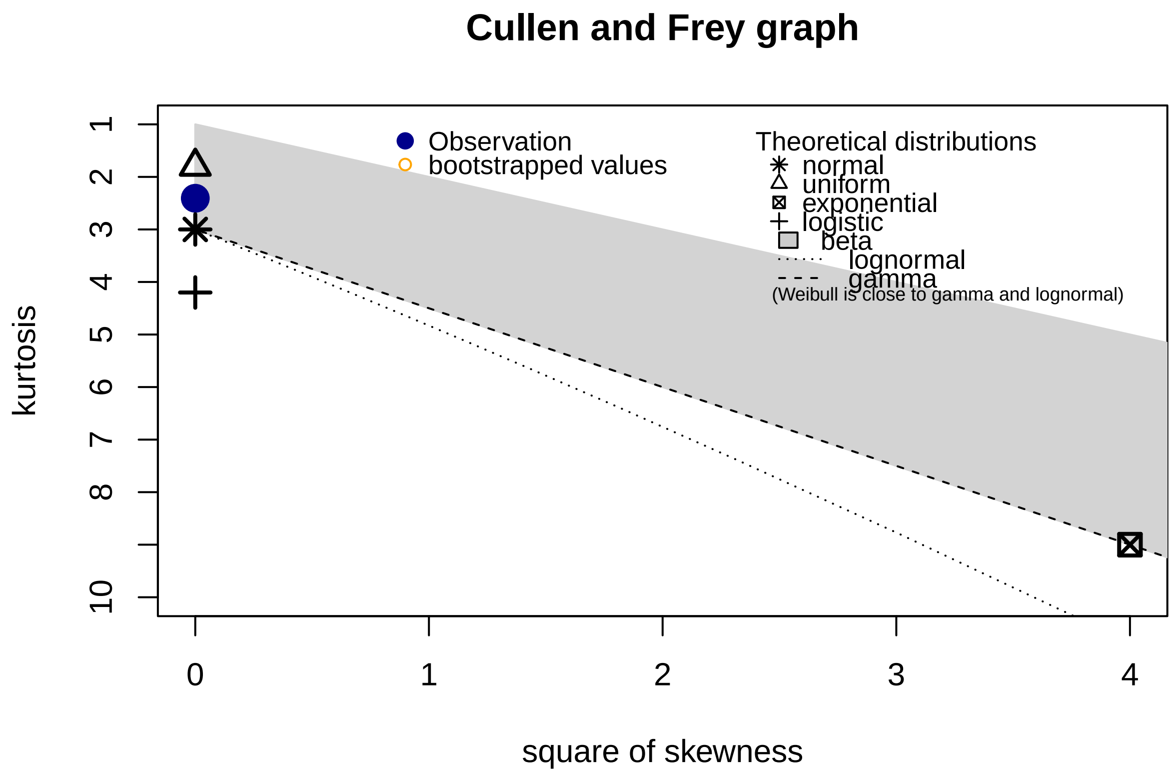
1. TRIANGULAR

FORMULA : u1+u2

```
set.seed(475) #last 3 digits of register number set as seed.
u1<-runif(n=10000, min=0, max=1)
u2<-runif(n=10000,min=0,max=1)
hist(u1+u2, main = "Triangular Distribution", xlab = "Value", ylab = "Frequency")
```




```
descdist(u1+u2,discrete=FALSE,boot=500)
```




```
## summary statistics
## ------
## min:  0.01513938   max:  1.985916
## median:  0.9994113
## mean:  1.000771
## estimated sd:  0.4063818
## estimated skewness:  0.01301422
## estimated kurtosis:  2.406842
```

CROSS-VALIDATION:

- For a triangular distributin, the kurtosis value is lower than that of normal distribution which is 3
- Here, since the observation point is such that the kurtosis=2.5(less than 3), it verifies that the distribution is triangular

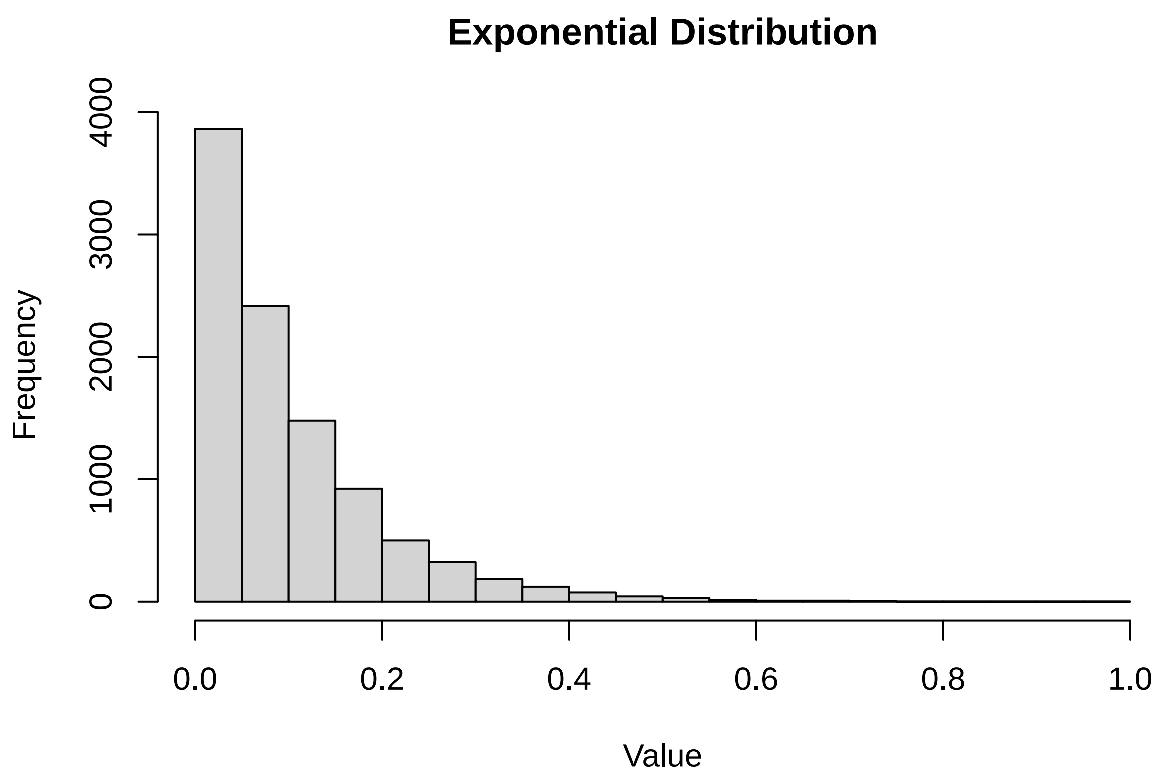
Real world example :

- **Monte Carlo simulation** is a triangular distribution that uses multiple values and averages the results
- Used to estimate the probability of cost overruns in large projects in business and investing
- Triangular distribution is used in Project analysis related to cash flow and value of the project.

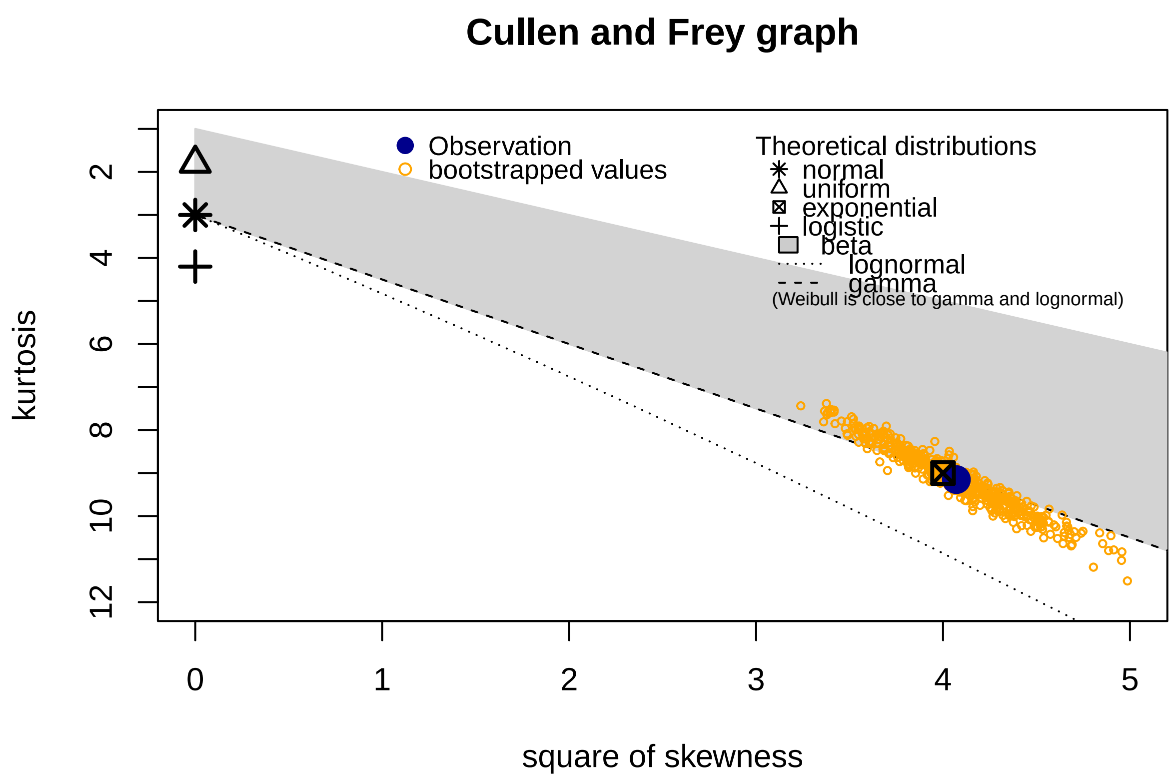
2. EXPONENTIAL

FORMULA: (-1/lambda)*log(1-u)

```
exponential_dist <- function(dist, lambda=10){
  exp_dist <- function(value){
    return ((-1/lambda)*log(1-value))
  }
  return (sapply(dist, exp_dist))
}
final_exp <- exponential_dist(u1)
hist(final_exp, main = "Exponential Distribution", xlab = "Value", ylab = "Frequency")
```




```
descdist(final_exp,discrete=FALSE,boot=500)
```




```
## summary statistics
## ------
## min:  6.066191e-06   max:  0.9741999
## median:  0.06967558
## mean:  0.1002682
## estimated sd:  0.1000278
## estimated skewness:  2.017764
## estimated kurtosis:  9.151716
```

CROSS-VALIDATION:

- Since the observation point overlaps with the exponential point of the Cullen and Frey graph, it verifies that the distribution is exponential

Real world example :

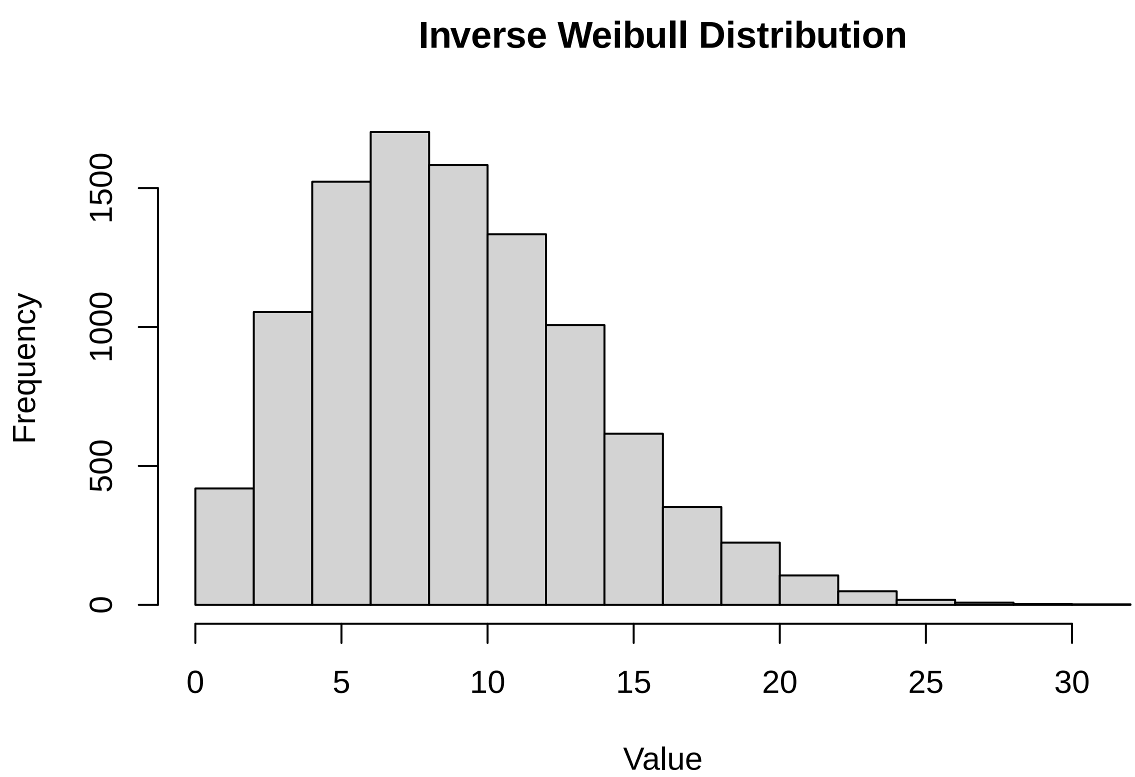
- The distribution of individual income is given by an exponential function for the U.S. Census data 1996.
- If individual income increases linearly with the hierarchical level, then the income distribution is exponential.

3. INVERSE WEIBULL

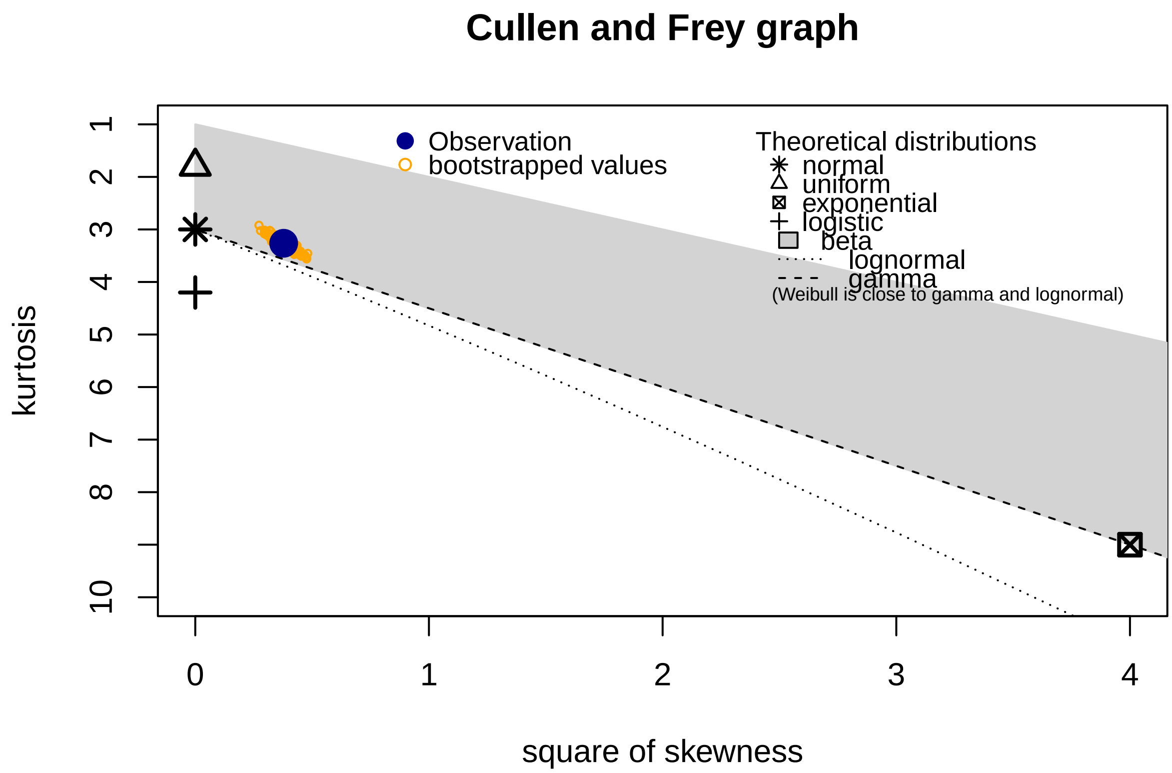
Assuming k=2 and lambda 10

FORMULA : (lambda* -ln(1-u))^(1/k)

```
inverseWB_dist <- function(dist, lambda=10, k=2){
  IWB_dist <- function(value){
    return (lambda * (-1* log(1-value))^(1/k))
  }
  return (sapply(dist, IWB_dist))
}
final_IWB <- inverseWB_dist(u1)
hist(final_IWB,main = "Inverse Weibull Distribution", xlab = "Value", ylab = "Frequency")
```




```
descdist(final_IWB,discrete=FALSE,boot=500)
```




```
## summary statistics
## ------
## min:  0.07788575   max:  31.21218
## median:  8.34719
## mean:  8.87418
## estimated sd:  4.63889
## estimated skewness:  0.615118
## estimated kurtosis:  3.263346
```

CROSS-VALIDATION

- Observation point near gamma and lognormal indicates inverse weibull distribution

Real world example :

- The inverse Weibull distribution has the ability to model failure rates which are quite common in reliability and biological studies
- This is critical for understanding and predicting the failure rates of systems or the lifetimes of living organism, the decay of biological processes, or the time until certain events (e.g., cell division, death) occur. This can help biologists understand the aging and survival patterns of organisms.

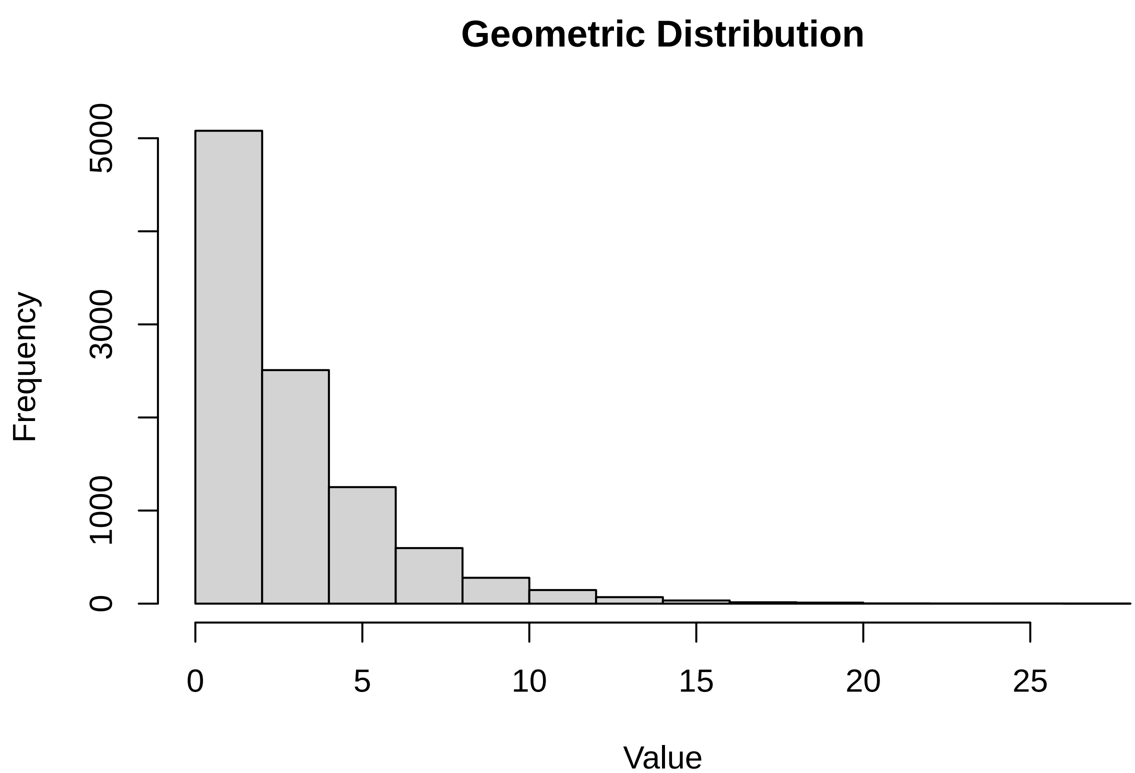
4. GEOMETRIC

FORMULA: log (1-u1)/log(1-p)

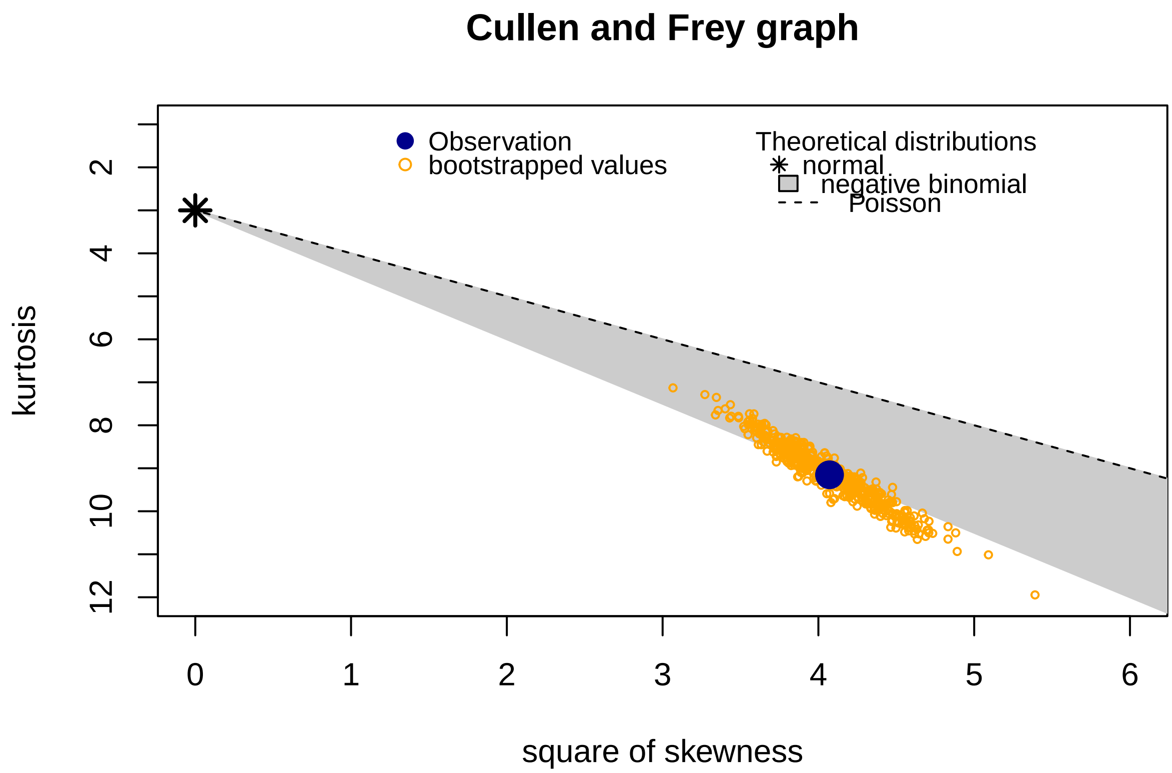
```
geometric_dist <- function(dist, p=0.3){
  geo_dist <- function(value){

      return (log(1-value)/log(1-p))

  }
  return (sapply(dist, geo_dist))
}
final_geo <- geometric_dist(u1)
hist(final_geo, main = "Geometric Distribution", xlab = "Value", ylab = "Frequency")
```




```
descdist(final_geo,discrete=TRUE,boot=500)
```




```
## summary statistics
## ------
## min:  0.0001700762   max:  27.31338
## median:  1.953475
## mean:  2.811193
## estimated sd:  2.804453
## estimated skewness:  2.017764
## estimated kurtosis:  9.151716
```

CROSS-VALIDATION

- Kurtosis of a geometric distribution is calculated as 6 / (1 - p). -Here, since p=0.3, Kurtosis should be ~= 8.5. Here since it is around 9, it verifies that the distribution is geometric in nature.

Real world example :

- In production processes and administrative processes, the occurrence of certain events is best described by a geometric distribution
- Production: For example, the number of attempts it takes for a machine to successfully produce a defect-free item
- Administration - number of attempts it takes to complete a specific administrative task, such as receiving payment from a client, or answering a customer's inquiry.

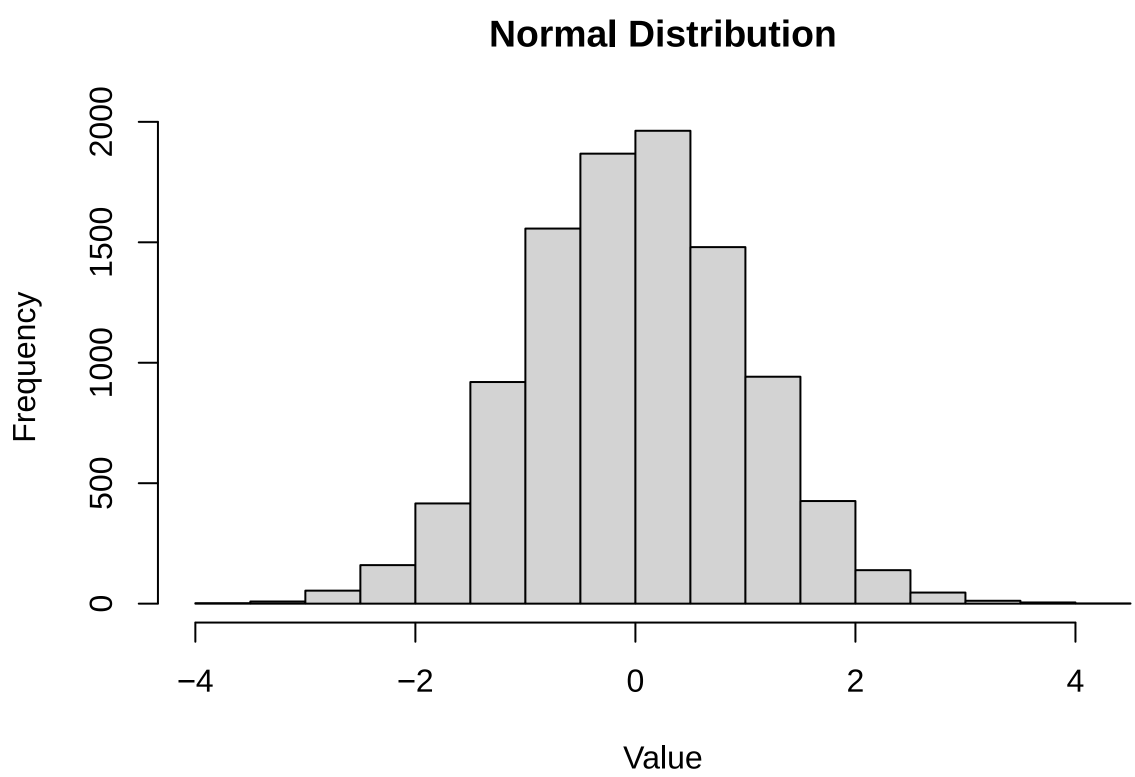
5. NORMAL DISTRIBUTION

FORMULA: sqrt(-2ln(u1)) cos(2*pi*u2)

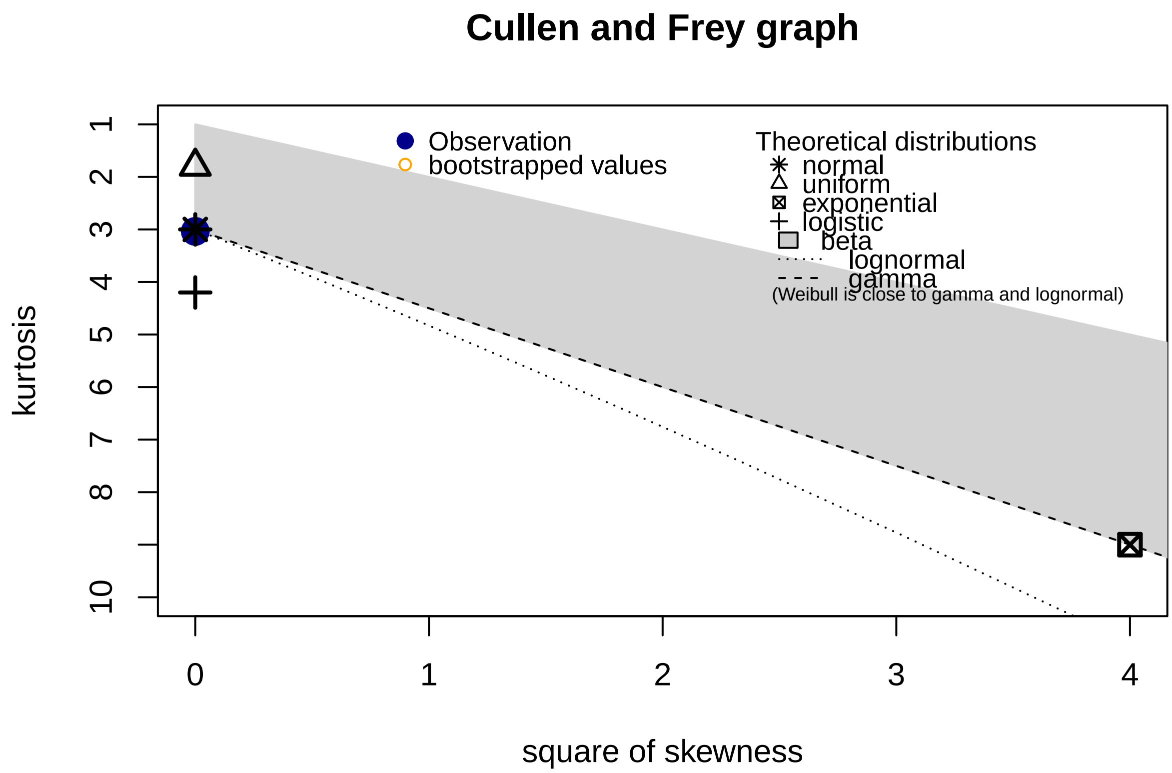
```
normal_dist <- function(dist1, dist2) {
  norm_dist <- function(value1, value2) {
    sqrt(-2 * log(value1)) * cos(2 * pi * value2)
  }

  return(mapply(norm_dist, dist1, dist2))

}
final_norm <- normal_dist(u1,u2)
hist(final_norm, main = "Normal Distribution", xlab = "Value", ylab = "Frequency")
```




```
descdist(final_norm,discrete=FALSE,boot=500)
```




```
## summary statistics
## ------
## min:  -3.727219   max:  4.007647
## median:  0.003368021
## mean:  -0.004942528
## estimated sd:  0.9922265
## estimated skewness:  0.002098099
## estimated kurtosis:  3.037697
```

CROSS-VALIDATION

- Observation point overlaps with the star point indicating normal distribution

Real world example :

- The variability inherent in the diagnosis of dyslexia can be both quantified and predicted with use of the normal distribution model.
- Dyslexia was defined in terms of a discrepancy score: actual reading achievement - achievement predicted for intelligence.
- It was observed that each of the discrepency scores followed a univariate normal distribution

REFERENCES:

1. **Triangular:**

- https://rpubs.com/pac25/la1#:~:text=Uses,use%20in%20simulatin%20business%20decisions
- https://www.jstor.org/stable/90001156

2. **Exponential:**

- https://arxiv.org/pdf/cond-mat/0008305.pdf

3. **Inverse Weibull Distribution:**

- https://link.springer.com/article/10.1007/s00362-009-0271-3

4. **Geometric Distribution:**

- https://www.tandfonline.com/doi/abs/10.1080/00224065.1992.12015229

5. **Normal Distribution:**

- https://www.nejm.org/doi/full/10.1056/nejm199201163260301

1. Exponential Distribution

$$F_x(x) = 1 - e^{-\lambda x} = u$$

$$\Rightarrow e^{-\lambda x} = 1 - u$$

$$\Rightarrow -\lambda x = \ln(1-u)$$

$$\Rightarrow \boxed{X = \frac{-1}{\lambda} \ln(1-u)}$$

2. Inverse Weibull Distribution

$$F_x(x) = 1 - e^{(x/\lambda)^k} = u$$

$$\Rightarrow e^{(x/\lambda)^k} = 1 - u$$

$$\Rightarrow \left(\frac{x}{\lambda}\right)^k = \ln(1-u)$$

$$\Rightarrow x^k = \lambda^k \ln(1-u)$$

$$\Rightarrow x = \lambda (\ln(1-u))^{1/k}$$

3. Geometric Distribution

$$X = \frac{\log u_1}{\log(1-p)}$$

Figure 1: Distbns

3. Box-Muller Transform

3.1. The Definition of the Algorithm

We can therefore identify an algorithm that maps the values drawn from a uniform distribution into those of a normal distribution. The algorithm that we describe here is the Box-Muller transform. **This algorithm is the simplest one to implement in practice**, and it performs well for the pseudorandom generation of normally-distributed numbers.

The algorithm is very simple. We first start with two random samples of equal length, u_1 and u_2, drawn from the uniform distribution $U(0, 1)$. Then, we generate from them two normally-distributed random variables z_1 and z_2. Their values are:

- $z_1 = \sqrt{-2\ln(u_1)}\cos(2\pi u_2)$
- $z_2 = \sqrt{-2\ln(u_1)}\sin(2\pi u_2)$

Figure 2: normalDistbn