Assignment 2

Jonathan De Los Santos

4/14/2021

Contents

1 Problem 1 **1**

1.1 a) Develop a Monte Carlo simulation model 2

1.2 b) Create a histogram 2

1.3 c) Estimate the probability 3

2 Problem 2 **4**

2.1 a) Estimate the expected (mean) net present value (NPV) over 5 years 4

2.2 b) What is the probability of negative NPV? 6

3 Problem 3 **7**

3.1 a) Suggest the optimal purchase quantity 7

3.2 b) Report a 95% confidence interval for the optimal profit 8

4 Problem 4 **8**

4.1 Confirm IID 11

4.2 Failure Count 17

4.3 Repair Time 20

4.4 Drive Time 26

5 Problem 5 **29**

1 Problem 1

The time for an automated storage and retrieval system in a warehouse to locate a part consists of three movements. Let X be the time to travel to the correct aisle. Let Y be the time to travel to the correct location along the aisle. And let Z be the time to travel up to the correct location on the shelves. Assume that the distributions of X, Y, and Z are as follows:

- X normal with mean 25 and standard deviation 4 seconds
- Y triangular with minimum 10, maximum 22 seconds, and most likely travel time of 15 seconds.
- Z truncated-normal with minimum 4, mean 6, and standard deviation 1 seconds

1.1 a) Develop a Monte Carlo simulation model

> Based on 100,000 random observations that estimate the mean and standard deviation of the total time it takes to locate a part.

Developing this model requires the use of three separate functions for each variable's specific continuous distribution: normal, triangular, and truncated-normal.

Using these respective functions, we can add together the result and calculate an estimated mean of 46.74 and and estimated standard deviation of 4.8.

Install triangle and truncnorm packages if necessary:

```
#install.packages('triangle')
#install.packages('truncnorm')
```

```
library(triangle)
library(truncnorm)

set.seed(123)
sim = 100000

x <- rnorm(sim, mean = 25, sd = 4)

y <- rtriangle(sim, a=10, b=22, c=15)

z <- rtruncnorm(sim, a = 4, mean = 6, sd = 1)

t_locate <- x + y + z

cat("Estimated mean: ", mean(t_locate), "\n")
```

```
## Estimated mean:  46.73611
```

```
cat("Estimated standard deviation: ", sd(t_locate))
```

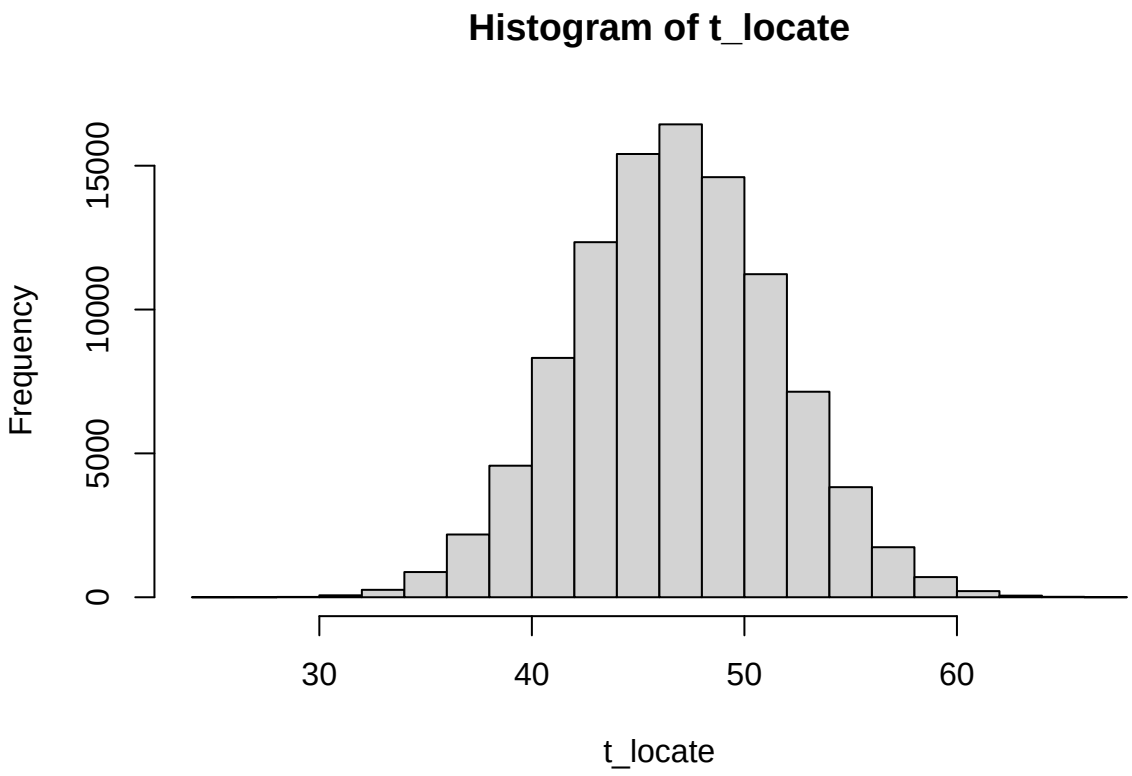
```
## Estimated standard deviation:  4.80026
```

1.2 b) Create a histogram

> For the total time that it takes to locate a part.

The histogram of our final model `t_locate` shows a normal distribution around the mean of 46.74.

```
hist(t_locate)
```



1.3 c) Estimate the probability

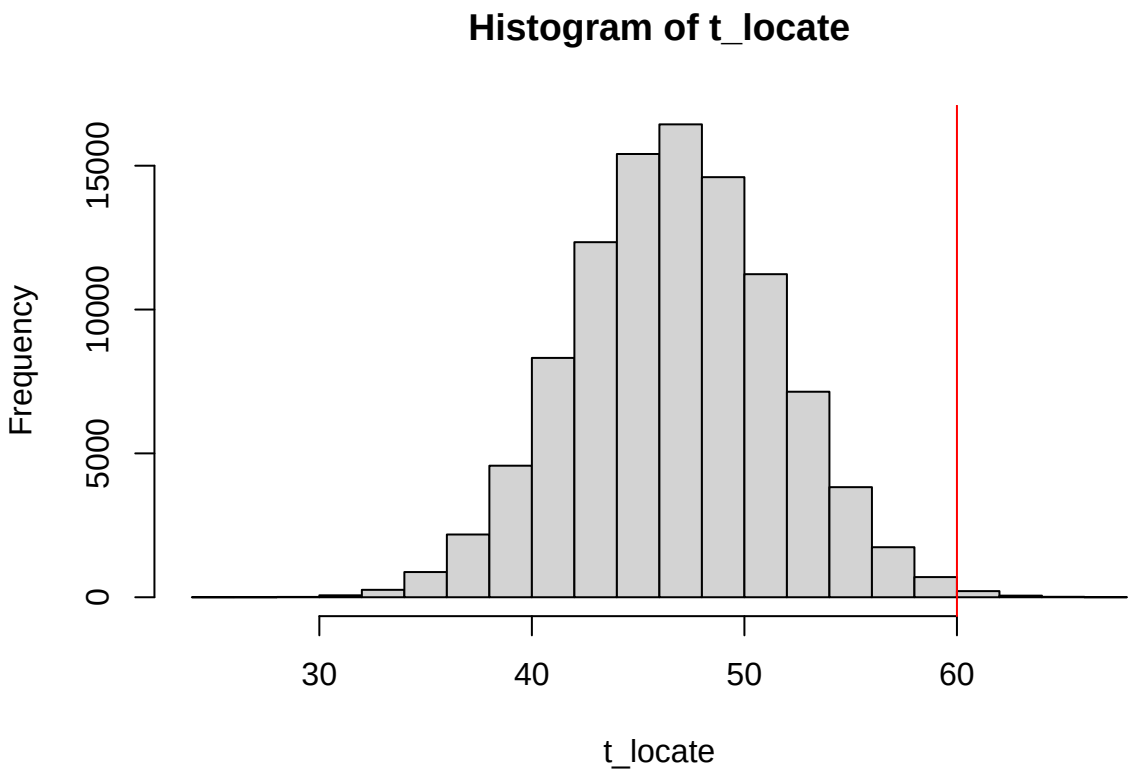
> ...that the time to locate a part exceeds 60 seconds. Graphically explain the probability by adding a vertical line at 60 seconds to the histogram of part b.

The probability that the time to locate is greater than 60 seconds is the mean of our simulation over 60. The low result of 0.29% can be visualized by the distance of this value plotted on the histogram.

```
cat("Likelihood of time to locate > 60:", mean(t_locate > 60))
```

```
## Likelihood of time to locate > 60: 0.00287
```

```
hist(t_locate)
abline(v = 60, col = "red")
```



2 Problem 2

Miller Pharmaceuticals needs to decide whether to conduct clinical trials and seek FDA approval for a newly developed drug. Suppose that analysts have made the following assumptions:

- R&D costs: Triangular(min= $450, max = $800, most likely = $700) in millions of dollars
- Clinical trials costs: Uniform(min= $135, max = $170) in millions of dollars
- Market size: Normal(mean = 2,200,000, sd = 250,000)
- Market share in year 1: Uniform(min = 5%, max = 10%)
- Discount Rate = 0.10

All other data are considered constant:

- Market size growth = 3% per year
- Market share growth = 15% per year
- Monthly revenue/prescription = $130
- Monthly variable cost/prescription = $40

2.1 a) Estimate the expected (mean) net present value (NPV) over 5 years

To find net present value, we have to simulate our stochastic variables and loop through the models with the simulated values. This allows us to determine estimated annual profit to which we can apply the discount rate and arrive at the net present profit and subsequently, the net present value.

```r
library(triangle)

set.seed(123)
sim = 1000

# Financial Data
year = 5
discount = 0.10
mSizeGrowth = 0.03
mShareGrowth = 0.15

# Unit Profit Variables
unitRevenue = 130
unitCost = 40

# Annual Profit Model
AnnualProfit = function(year, mSize, mSizeGrowth, mShare, mShareGrowth, unitRevenue, unitCost){
  mSize = mSize*(1+mSizeGrowth)^(year-1)
  mShare = mShare*(1+mShareGrowth)^(year-1)
  sale = mSize*mShare
  annualRevenue = 12*sale*unitRevenue
  annualCost = 12*sale*unitCost
  return(annualRevenue-annualCost)
}

# Net Present Profit
nPP <- function(discount, profit, year, projectCost){
  netProfit <- 0
  for(t in 1:year) {
    netProfit <- netProfit + profit[t]/(1+discount)^t
  }
  return(netProfit)
}

simulated_npv <- c()
for(i in 1:sim){
  # Market Estimates
  mSize =  rnorm(sim, mean = 2.2, sd = 0.25)
  mShareY1Est <- runif(sim, min = 0.05, max = 0.1)

  # Cost Estimates
  rdCost = rtriangle(sim, a=450, b=800, c=700)

  trialCost = runif(sim, min=135, max=170)

  projectCost = rdCost + trialCost

  # Profit Model for X Years
  profit <- AnnualProfit(1:year, mSize, mSizeGrowth, mShareY1Est, mShareGrowth, unitRevenue, unitCost)

  # Net present profit
  netPP <- nPP(discount, profit, year)
```

```r
  # Net present value
  npv = netPP-projectCost

  # Simulated NPV for risk calculation
  simulated_npv <- c(simulated_npv, npv)
}

cat("Net Present Value:", mean(npv))
```

```
## Net Present Value: 158.3745
```

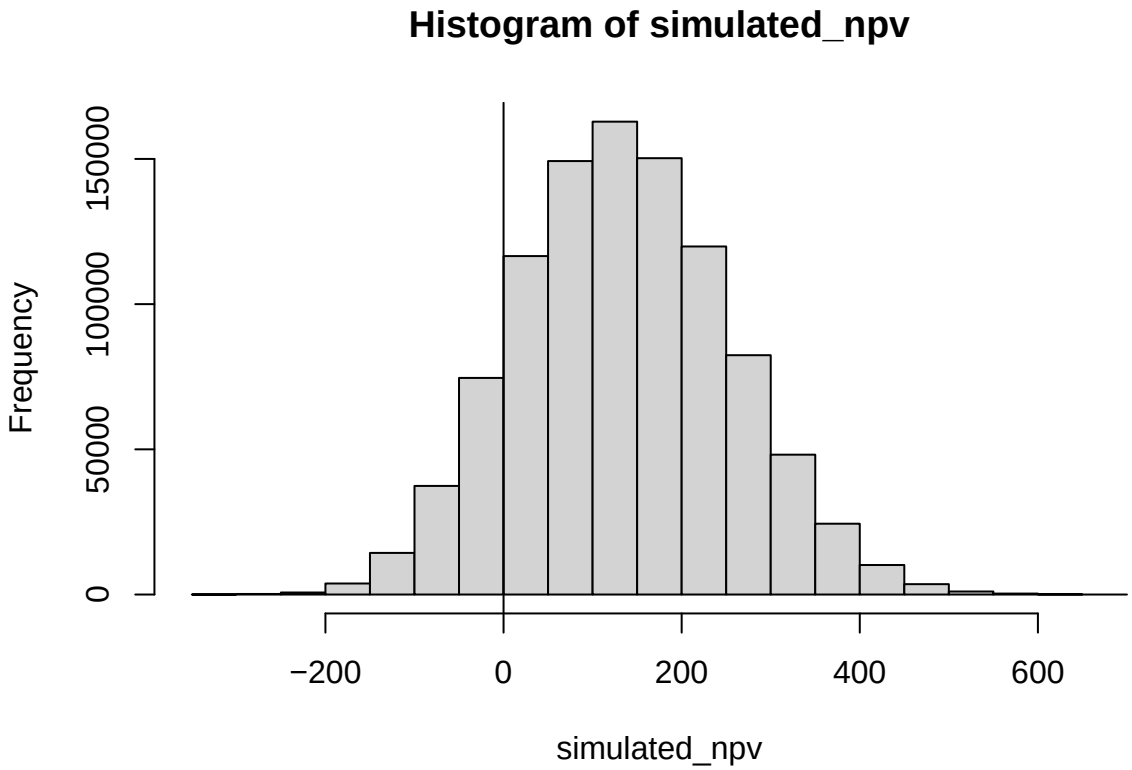
2.2 b) What is the probability of negative NPV?

There is a 0.52% chance the NPV is negative or zero. The histogram reflects this visually by demonstrating that zero lies in the left tail of the distribution.

```r
mean(simulated_npv <= 0)
```

```
## [1] 0.131047
```

```r
hist(simulated_npv)
abline(v = 0)
```



3 Problem 3

> Use the Newsvendor Model to set up and run a Monte Carlo simulation assuming that demand is Truncated-Normal with a mean of 45, the standard deviation of 3, the minimum value of 40, and the maximum value of 50.

3.1 a) Suggest the optimal purchase quantity

> ...when the cost per unit (C) is $12, selling price (R) is $18, and the salvage value (S) is $9.

This solution revolves around the purchase decision profit model:

$netprofit = R \times min(Q, D) + S \times max(0, Q - D) - C \times Q$

By creating this into the function `netProfit`, we can iterate over the model with simulated values to find expected profit for several values. Based on this formulation, the optimal purchase quantity is 50.

```
c = 12
r = 18
s = 9
d = 45
q = 46
range = 40:50
j = 0 # increment counter for matrix loop

set.seed(123)
sim = 1000

# Profit function
netProfit = function(d, q, r, s, c){
  r*min(d,q) + s * max(0, q-d) - c*q
}


# Resample
sim_demand <- sample(d, sim, replace = TRUE)

# Profit matrix
expectedProfit <- matrix(nrow = sim, ncol = length(range))

for (r in range) {
  j = j+1
  for (i in 1:sim) {
    expectedProfit[i,j] = netProfit(sim_demand[i], q, r, s, c)
  }
}

# Column means represent expected profit for entire range
colMeans(expectedProfit)
```

```
##  [1] 574.225 597.200 620.175 643.150 666.125 689.100 712.075 735.050 758.025
## [10] 781.000 803.975
```

3.2 b) Report a 95% confidence interval for the optimal profit

```
mu = 45
sd = 3
error <- qnorm(0.95)*sd/sqrt(sim)
left <- mu-error
right <- mu+error

cat("The 95% CI for the optimal profit is between", left, "and", right)
```

```
## The 95% CI for the optimal profit is between 44.84396 and 45.15604
```

4 Problem 4

The observations available in the wind turbine file represent

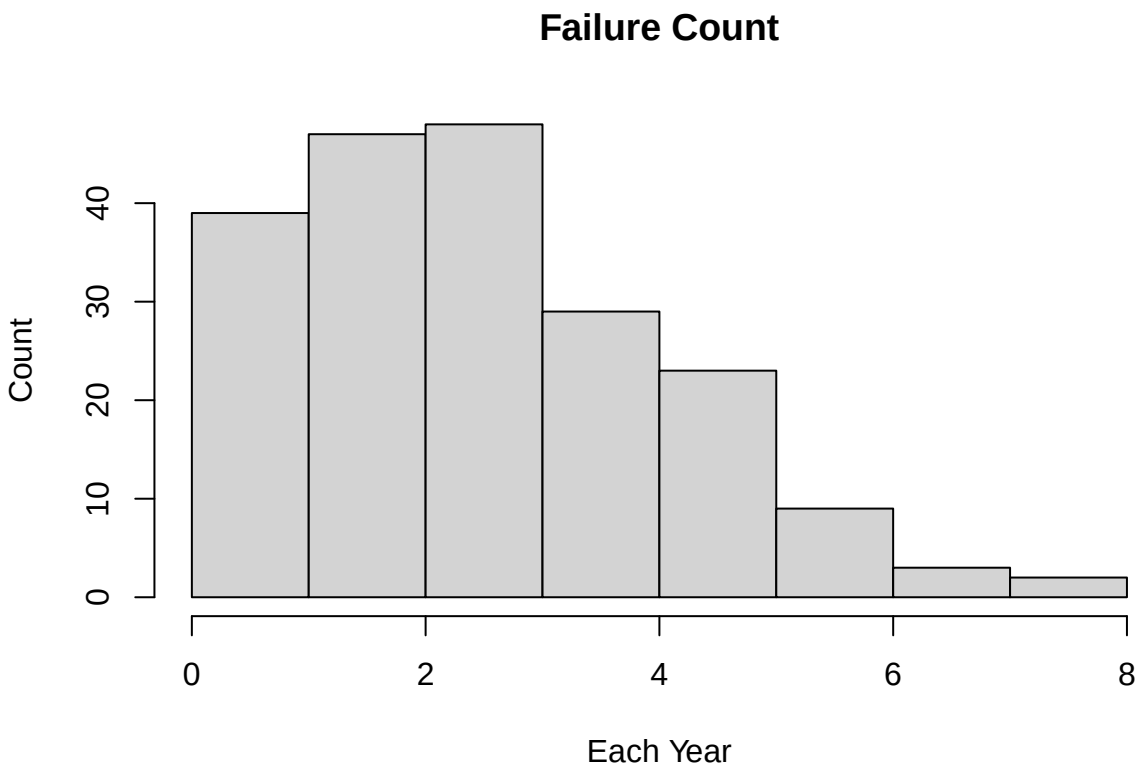
- "FailCount": the count of the number of failures on a windmill turbine farm per year
- "RepairTime": the time that it takes to repair a windmill turbine on each occurrence in minutes.
- "DriveTime": the time that it takes to notice the failure and drive from the control station to the windmill turbine in minutes

Using the techniques discussed in the input-analysis lecture, recommend an input distribution model for the "FailCount," "RepairTime," and "DriveTime" variables. Please pay attention to the fact that which variable is naturally discrete or continuous.

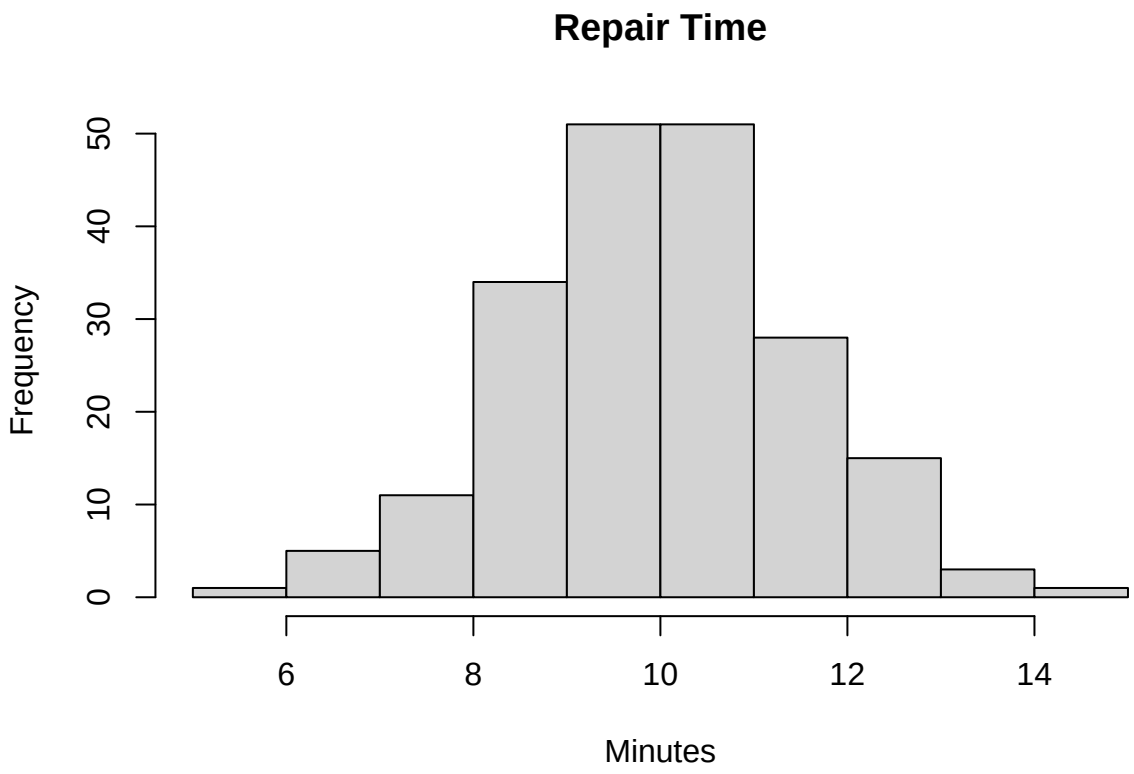
Begin by importing and analyzing the data. In the wind turbine dataset, we can visually estimate the following about each variable from the histogram:

- Failure Count appears discrete, right skewed and unimodal
- Repair Time appears continuous and normally distributed
- Drive Time appears continuous and bimodal

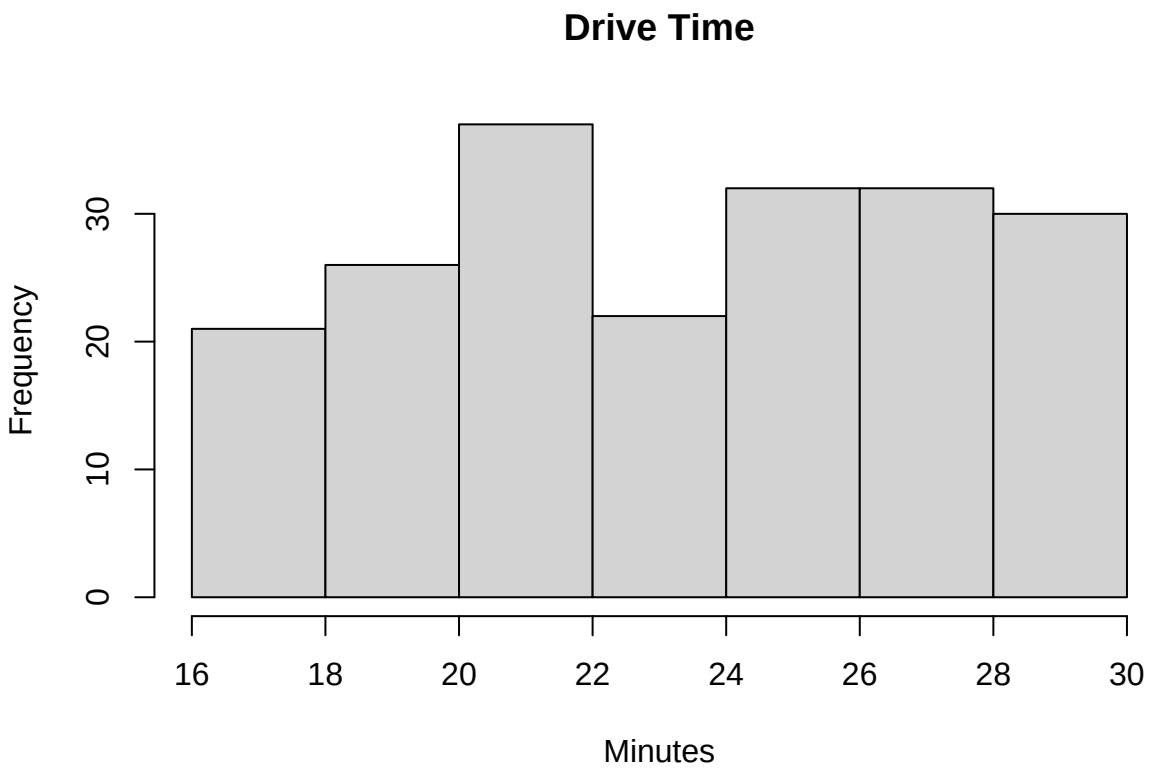
```
windata <- read.csv("Data Sets/windTurbineData.csv")
# Fix the data typo
colnames(windata)[2] <- "repair.time"

hist(windata$failure.count, main="Failure Count", ylab = "Count", xlab = "Each Year")
```




```
hist(windata$repair.time, main="Repair Time", ylab = "Frequency", xlab = "Minutes")
```



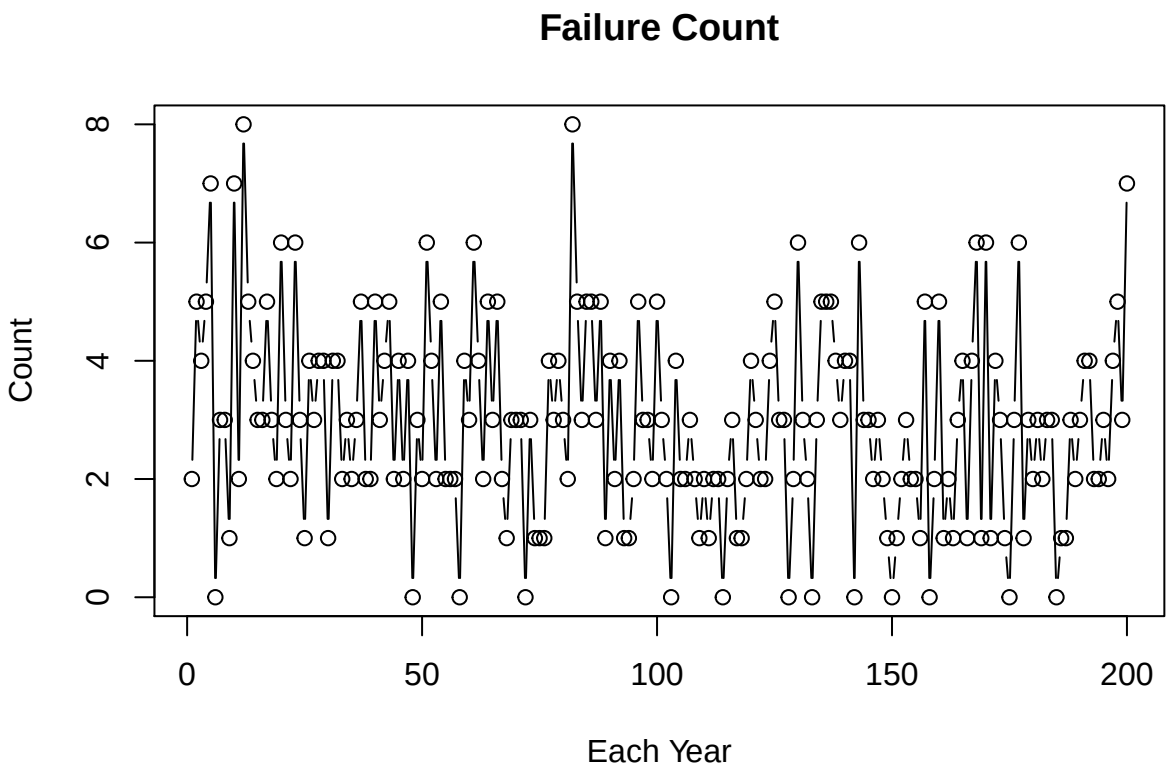

```
hist(windata$drive.time, main="Drive Time", ylab = "Frequency", xlab = "Minutes")
```



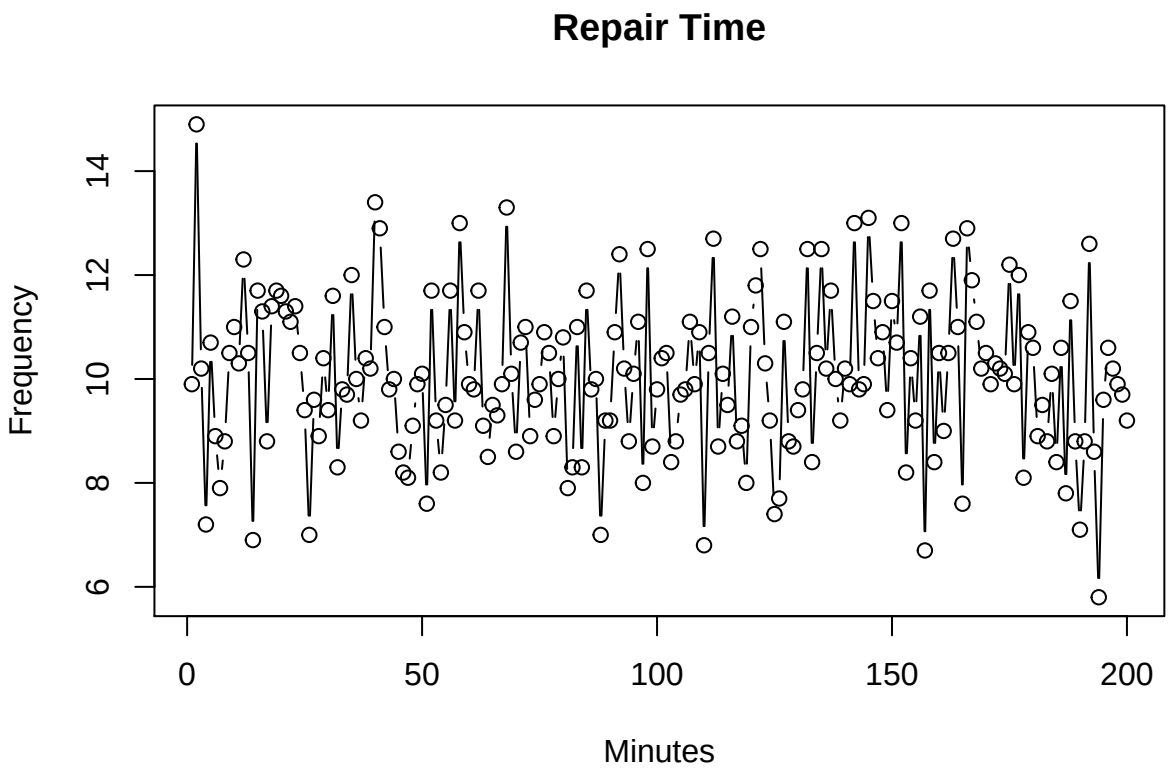
4.1 Confirm IID

Next we examine the time-series plots to check for trends. No noticeable patterns stick out in any of the variables.

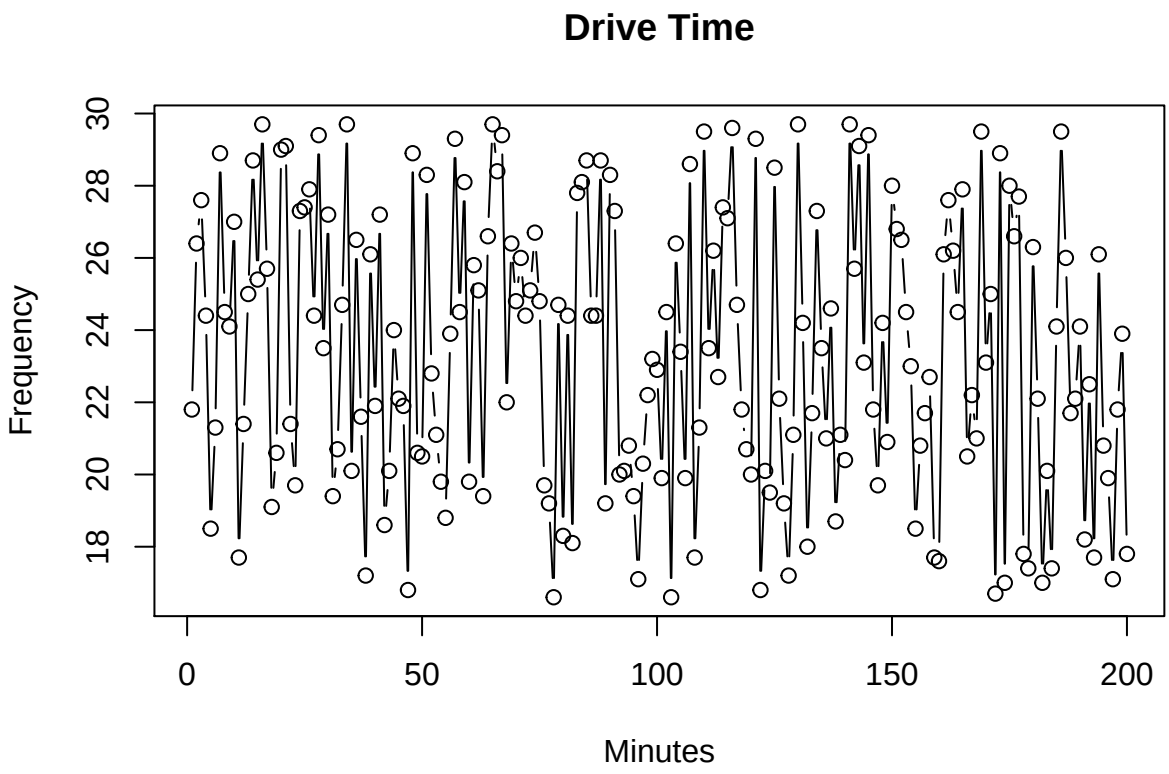
```
plot(windata$failure.count, type="b", main="Failure Count", ylab = "Count", xlab = "Each Year")
```




```
plot(windata$repair.time, type="b", main="Repair Time", ylab = "Frequency", xlab = "Minutes")
```

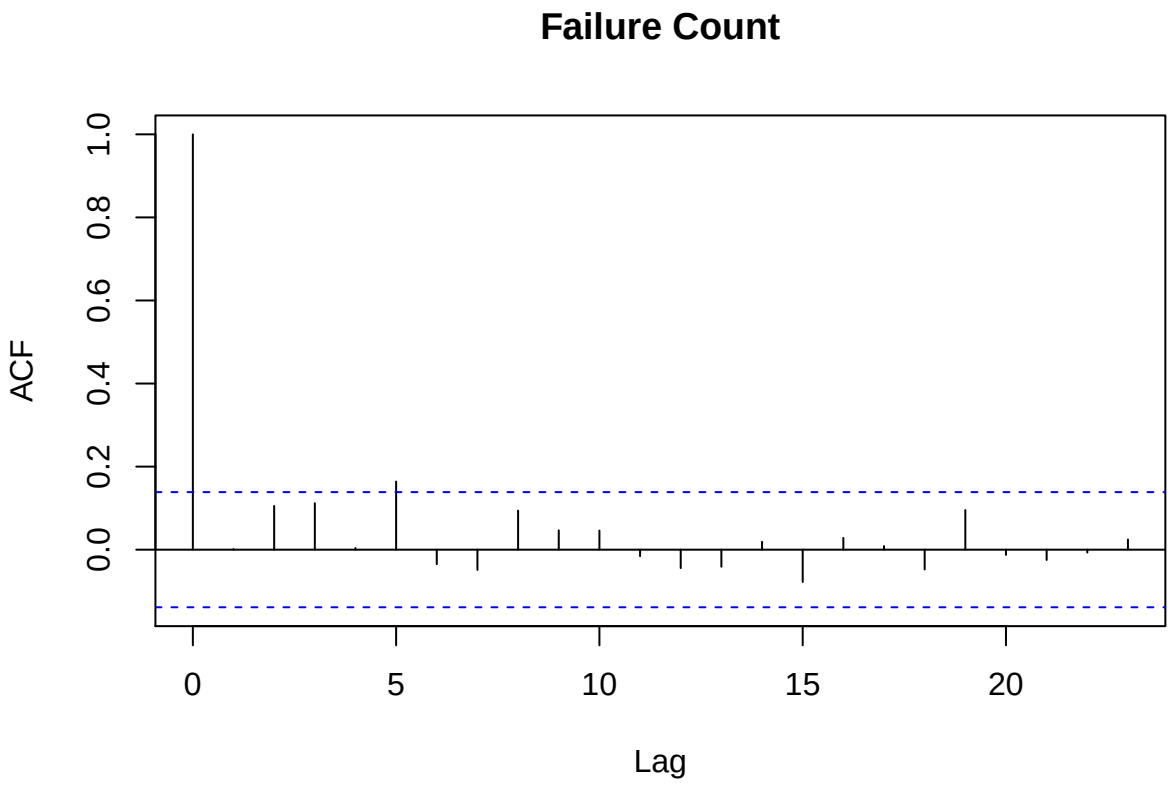



```
plot(windata$drive.time, type="b", main="Drive Time", ylab = "Frequency", xlab = "Minutes")
```

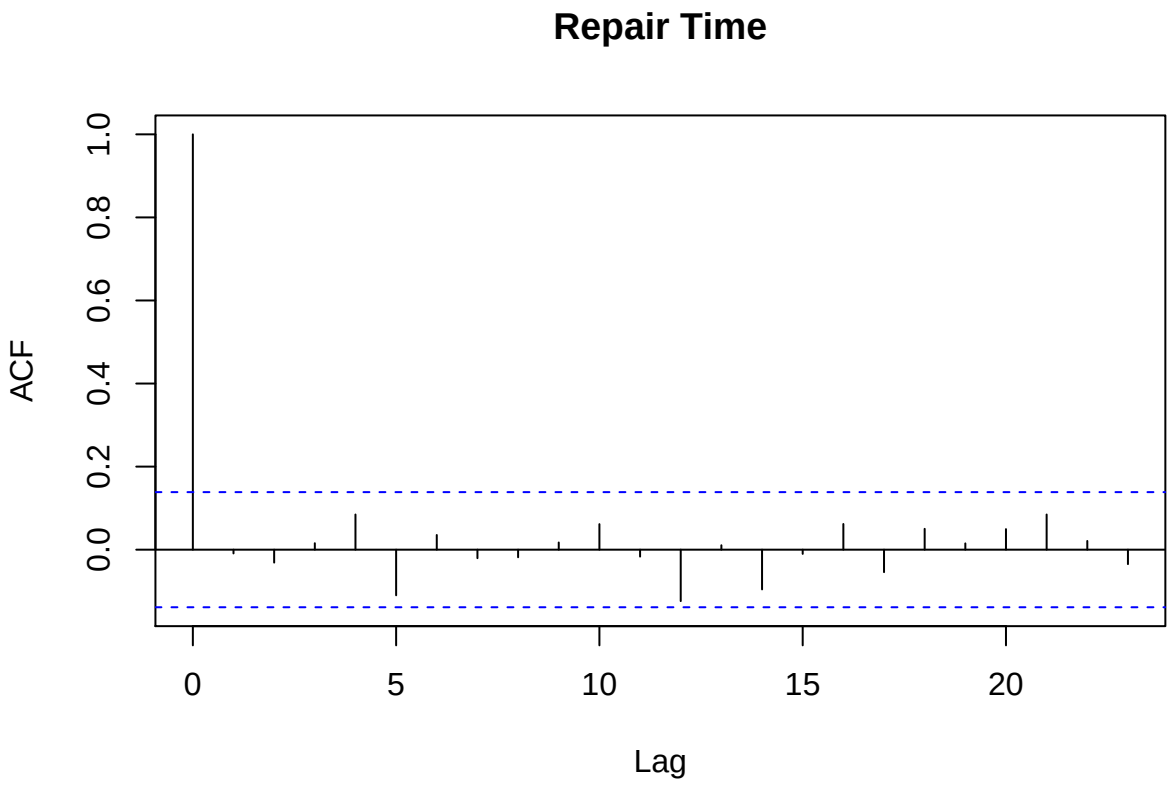


Finally, we confirm that all three variables are IID by checking the auto-correlation plot. There are a few lags that leave the confidence interval, but overall the three variables appear stationary.

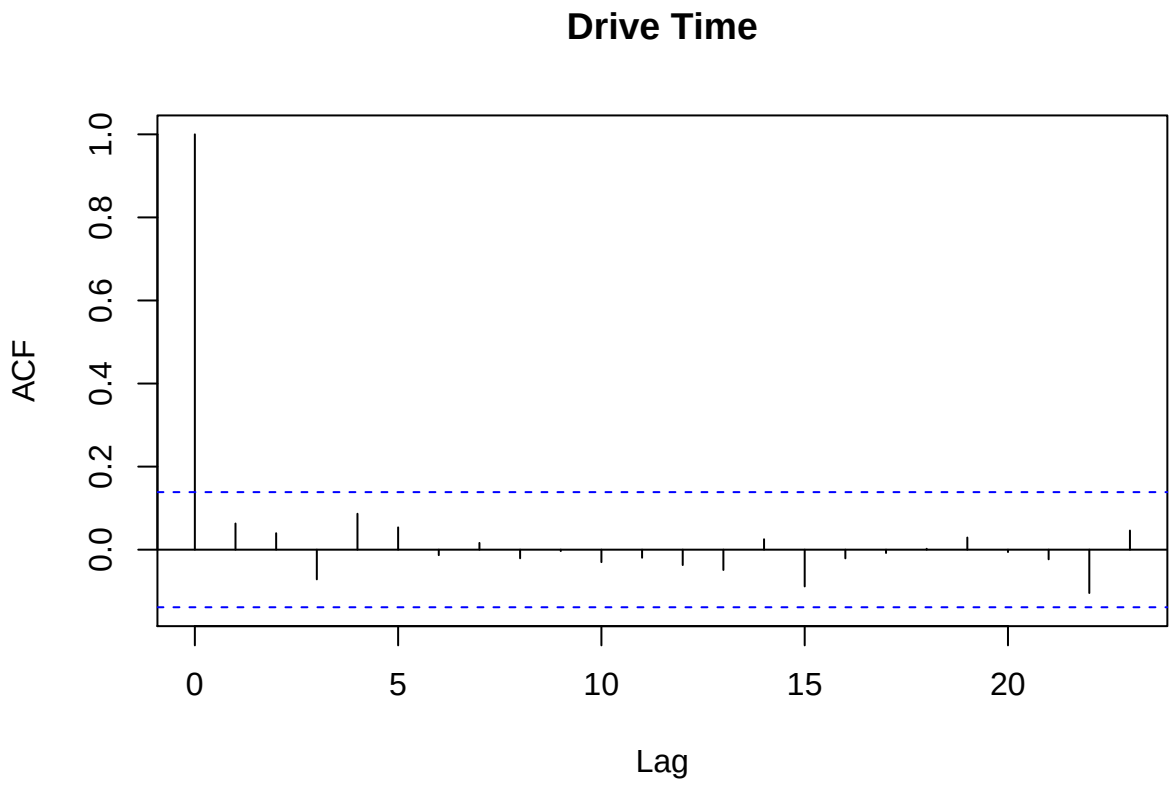
```
acf(windata$failure.count, main="Failure Count")
```




```
acf(windata$repair.time, main="Repair Time")
```




```
acf(windata$drive.time, main="Drive Time")
```



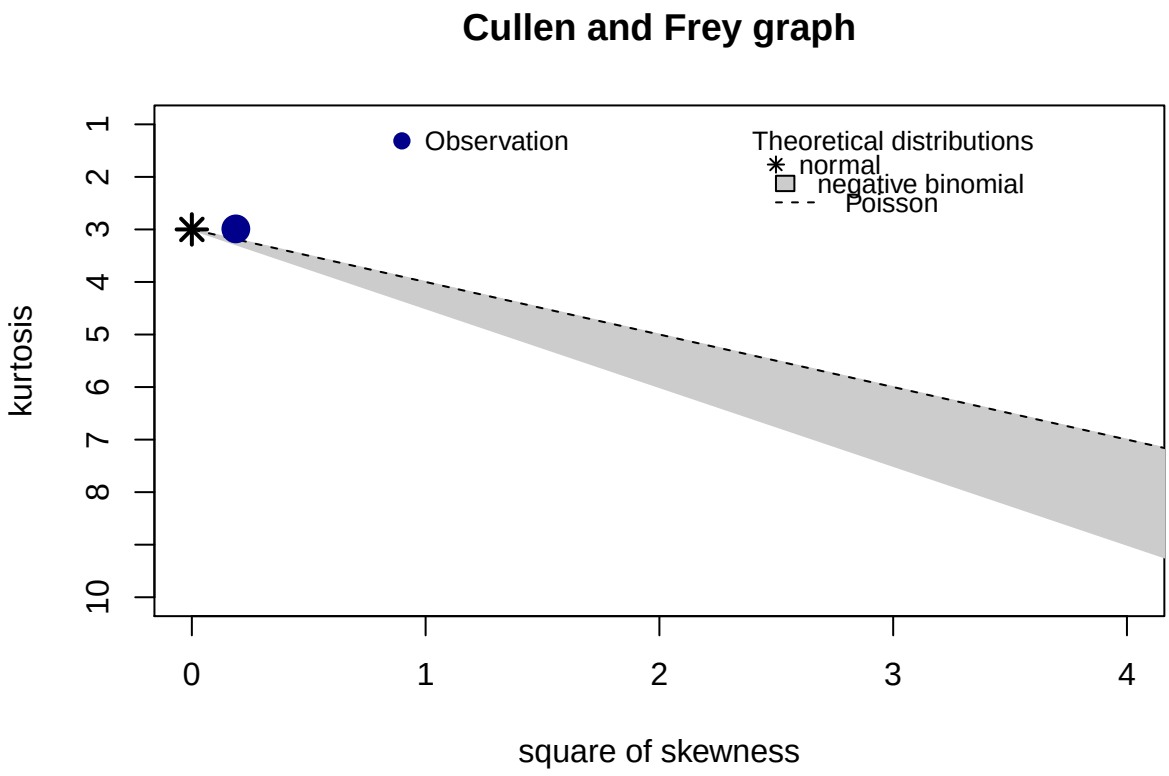
To easily estimate which distributions may be the best fit for each variable, we can use the `descdist()` function in the `fitdistrplus` package as a starting point to test distributions.

```
#install.packages('fitdistrplus')
library(fitdistrplus)
```

4.2 Failure Count

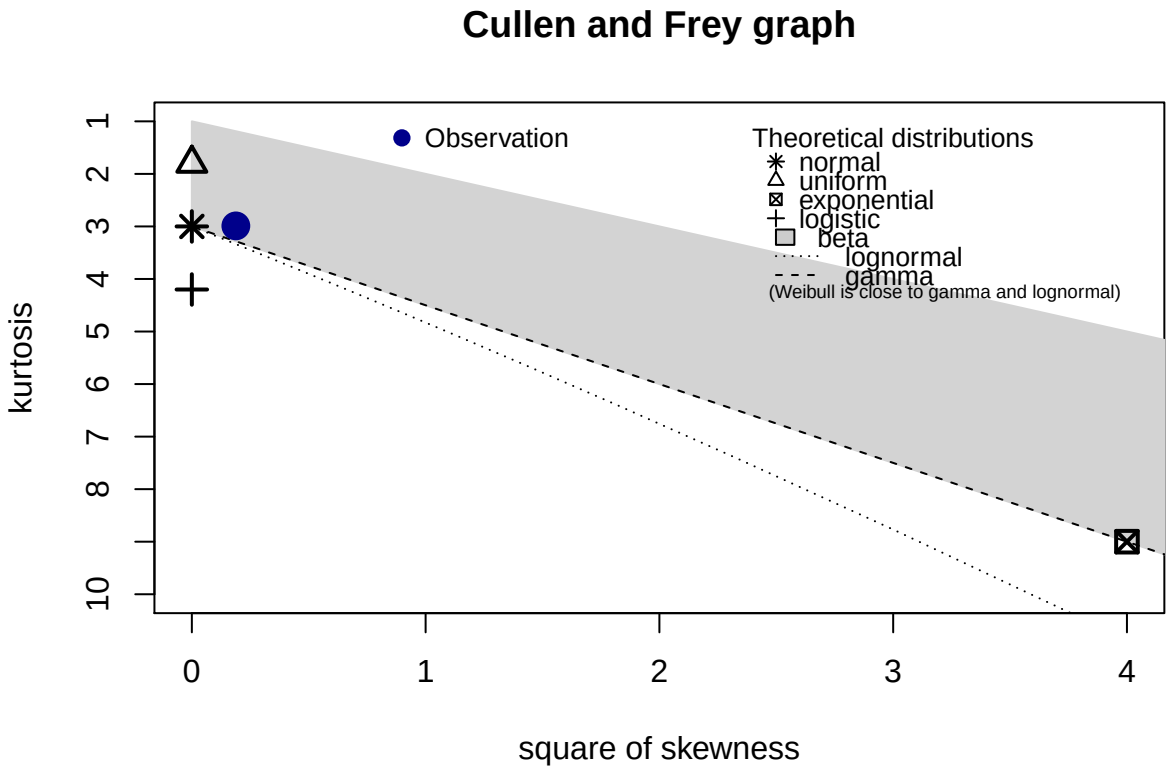
Based on the Cullen and Frey graph, the most likely distributions for our variable are Poisson and negative binomial We will test both to make a determination. The variable appears discrete, and the graph fit does not change significantly if attempted as continuous so we will test discrete distributions.

```
descdist(windata$failure.count, discrete = TRUE)
```




```
## summary statistics
## ------
## min:  0   max:  8
## median:  3
## mean:  2.93
## estimated sd:  1.688001
## estimated skewness:  0.4338968
## estimated kurtosis:  2.989638
```

```
descdist(windata$failure.count, discrete = FALSE)
```




```
## summary statistics
## ------
## min:  0   max:  8
## median:  3
## mean:  2.93
## estimated sd:  1.688001
## estimated skewness:  0.4338968
## estimated kurtosis:  2.989638
```

Both negative binomial and Poisson distributions return a $p > 0.05$ and are valid fits for the data. The Poisson fit is slightly higher and is the recommended distribution for simulating this data.

```
negBinom = fitdist(windata$failure.count, "nbinom")
summary(negBinom)
```

```
## Fitting of the distribution ' nbinom ' by maximum likelihood
## Parameters :
##          estimate Std. Error
## size 1.222852e+06 61.1606955
## mu   2.929835e+00  0.1210305
## Loglikelihood:  -382.9048   AIC:  769.8096   BIC:  776.4062
## Correlation matrix:
##              size           mu
## size 1.000000e+00 1.051932e-07
## mu   1.051932e-07 1.000000e+00
```

```r
poisBinom = fitdist(windata$failure.count, "pois")
summary(poisBinom)
```

```
## Fitting of the distribution ' pois ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## lambda     2.93  0.1210372
## Loglikelihood:  -382.9048   AIC:  767.8095   BIC:  771.1079
```

```r
gofstat(poisBinom)
```

```
## Chi-squared statistic:  1.78401
## Degree of freedom of the Chi-squared distribution:  4
## Chi-squared p-value:  0.7754066
## Chi-squared table:
##      obscounts theocounts
## <= 1  39.00000   41.97007
## <= 2  47.00000   45.84082
## <= 3  48.00000   44.77120
## <= 4  29.00000   32.79491
## <= 5  23.00000   19.21782
## > 5   14.00000   15.40518
##
## Goodness-of-fit criteria
##                                1-mle-pois
## Akaike's Information Criterion   767.8095
## Bayesian Information Criterion   771.1079
```

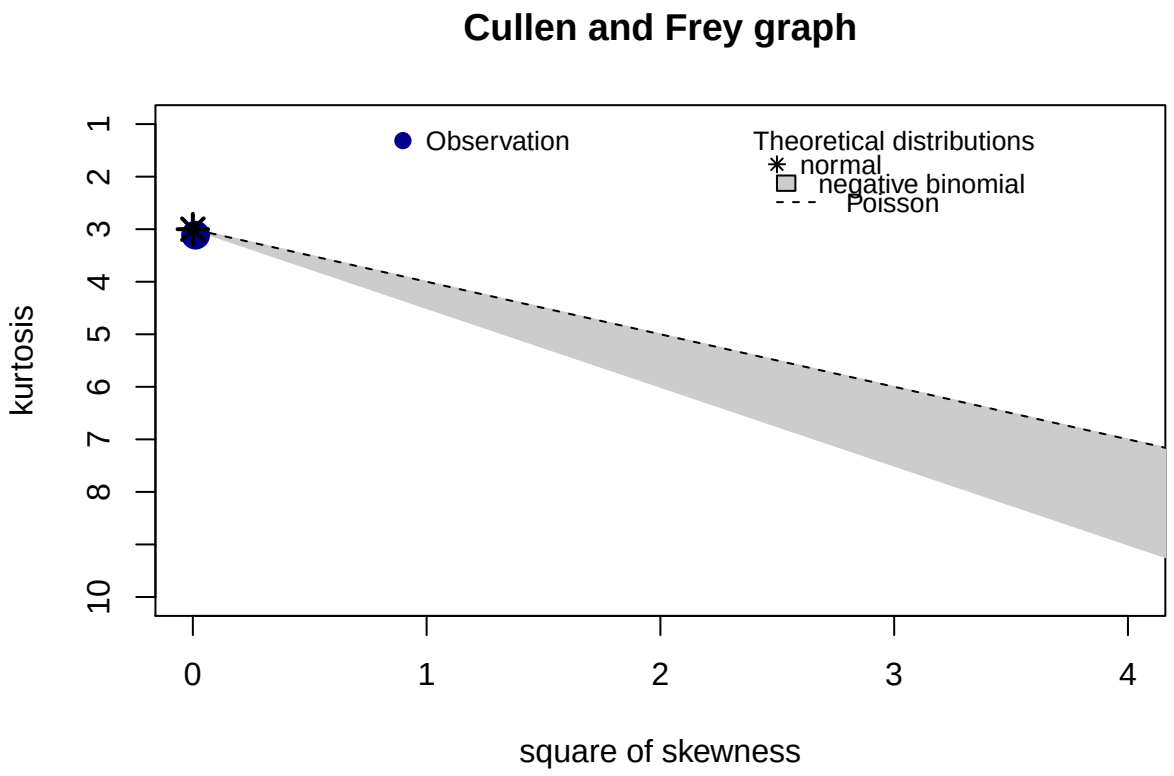
```r
gofstat(negBinom)
```

```
## Chi-squared statistic:  1.784626
## Degree of freedom of the Chi-squared distribution:  3
## Chi-squared p-value:  0.6182862
## Chi-squared table:
##      obscounts theocounts
## <= 1  39.00000   41.97530
## <= 2  47.00000   45.84320
## <= 3  48.00000   44.77097
## <= 4  29.00000   32.79290
## <= 5  23.00000   19.21557
## > 5   14.00000   15.40206
##
## Goodness-of-fit criteria
##                                1-mle-nbinom
## Akaike's Information Criterion     769.8096
## Bayesian Information Criterion     776.4062
```

4.3 Repair Time

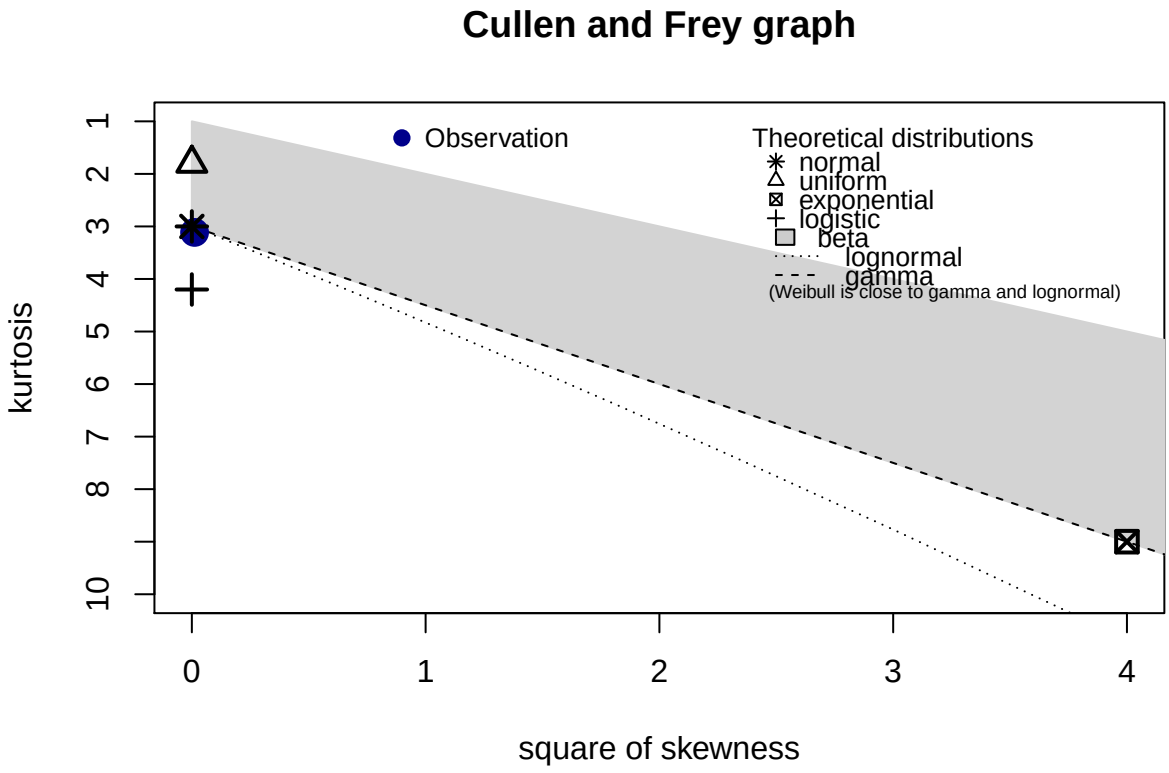
The `descdist()` plots are difficult to read for this data, but the oberservation appears to be slightly outside of the discrete range and the the data itself appear continuous. We will examine normal, lognormal, gamma, and Weibull distributions since the observation is close to all of these.

```r
descdist(windata$repair.time, discrete = TRUE)
```




```
## summary statistics
## ------
## min:  5.8   max:  14.9
## median:  10
## mean:  10.0495
## estimated sd:  1.511962
## estimated skewness:  0.1078961
## estimated kurtosis:  3.113694
```

```r
descdist(windata$repair.time, discrete = FALSE)
```




```
## summary statistics
## ------
## min:  5.8   max:  14.9
## median:  10
## mean:  10.0495
## estimated sd:  1.511962
## estimated skewness:  0.1078961
## estimated kurtosis:  3.113694
```

```
norm = fitdist(windata$repair.time, "norm")
summary(norm)
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## mean 10.049500 0.10664426
## sd    1.508178 0.07540873
## Loglikelihood:  -365.9681   AIC:  735.9362   BIC:  742.5329
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

```
lnorm = fitdist(windata$repair.time, "lnorm")
summary(lnorm)
```

```
## Fitting of the distribution ' lnorm ' by maximum likelihood
## Parameters :
##          estimate Std. Error
## meanlog 2.2959775 0.01084708
## sdlog   0.1534009 0.00766858
## Loglikelihood:  -368.0431   AIC:  740.0863   BIC:  746.6829
## Correlation matrix:
##         meanlog sdlog
## meanlog       1     0
## sdlog         0     1
```

```
gamma = fitdist(windata$repair.time, "gamma")
summary(gamma)
```

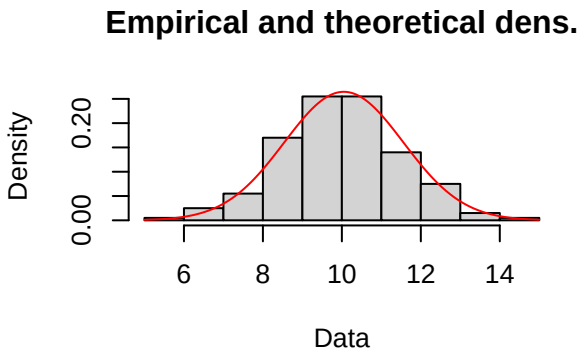
```
## Fitting of the distribution ' gamma ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## shape 43.454570   4.328873
## rate   4.324012   0.433241
## Loglikelihood:  -366.5354   AIC:  737.0708   BIC:  743.6675
## Correlation matrix:
##           shape      rate
## shape 1.0000000 0.9942526
## rate  0.9942526 1.0000000
```

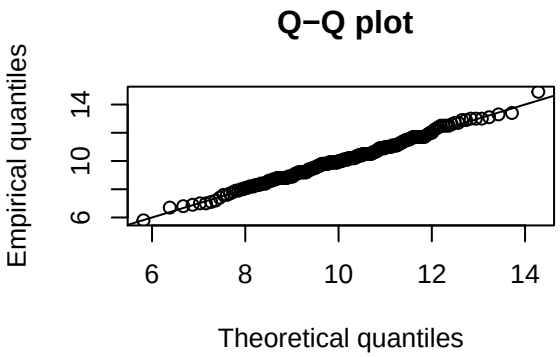
```
weibull = fitdist(windata$repair.time, "weibull")
summary(weibull)
```

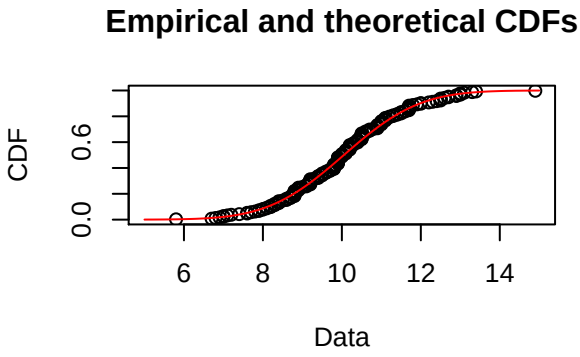
```
## Fitting of the distribution ' weibull ' by maximum likelihood
## Parameters :
##        estimate Std. Error
## shape  7.077176  0.3665029
## scale 10.699577  0.1131761
## Loglikelihood:  -372.9337   AIC:  749.8674   BIC:  756.4641
## Correlation matrix:
##           shape     scale
## shape 1.0000000 0.3286808
## scale 0.3286808 1.0000000
```

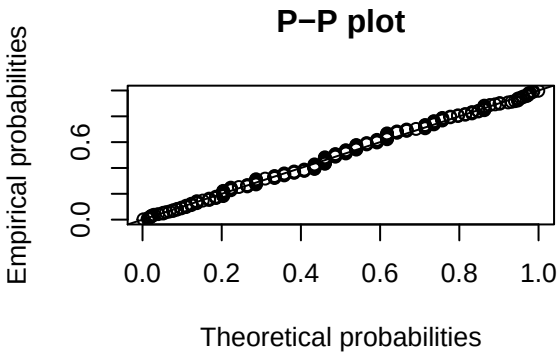
Unsurprisingly from the graph, the Log-likelihoods of all of these distributions is close. Let's plot the three most likely: normal, lognormal, and gamma to make our determination.

```
plot(norm)
```

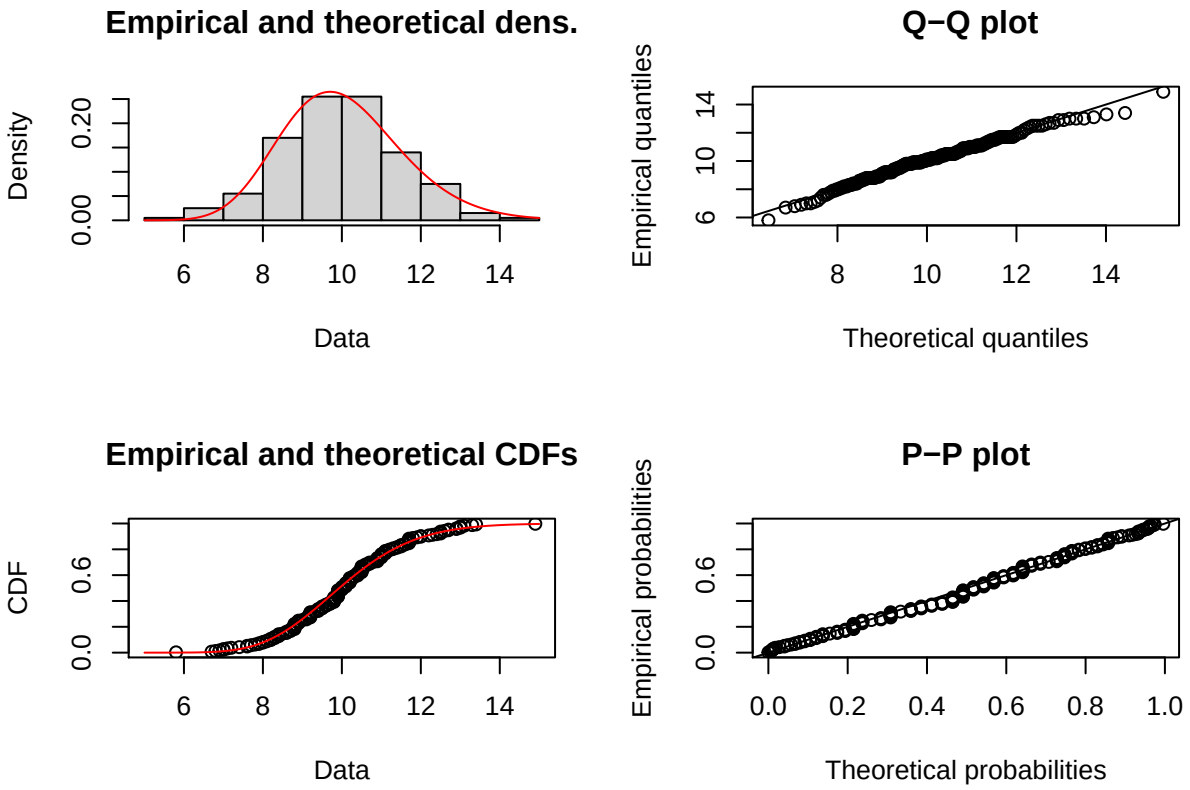

Empirical and theoretical dens.
Density
0.00
0.20
6
8
10
12
14
Data


Q−Q plot
Empirical quantiles
6
10
14
6
8
10
12
14
Theoretical quantiles

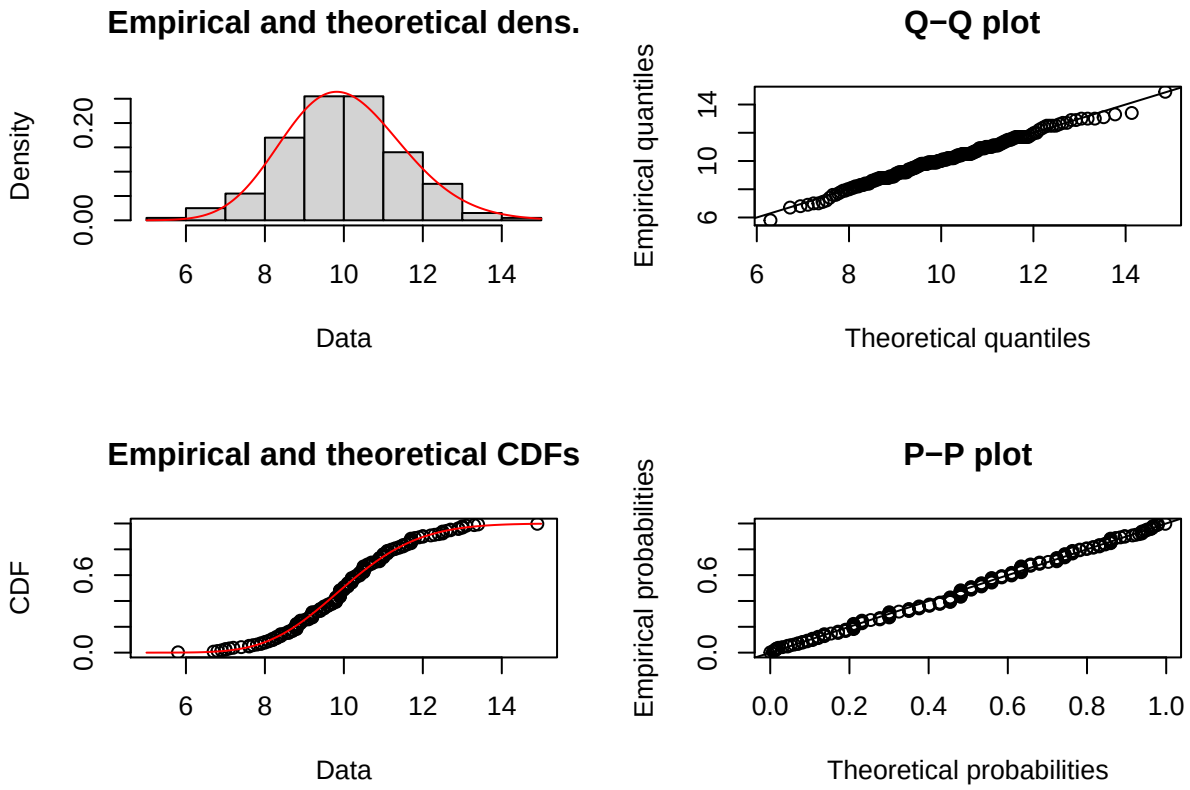

Empirical and theoretical CDFs
CDF
0.0
0.6
6
8
10
12
14
Data


P−P plot
Empirical probabilities
0.0
0.6
0.0
0.2
0.4
0.6
0.8
1.0
Theoretical probabilities

```
plot(lnorm)
```


Empirical and theoretical dens.
Density
0.00
0.20
6
8
10
12
14
Data
Q−Q plot
Empirical quantiles
6
10
14
8
10
12
14
Theoretical quantiles
Empirical and theoretical CDFs
CDF
0.0
0.6
6
8
10
12
14
Data
P−P plot
Empirical probabilities
0.0
0.6
0.0
0.2
0.4
0.6
0.8
1.0
Theoretical probabilities

```
plot(gamma)
```

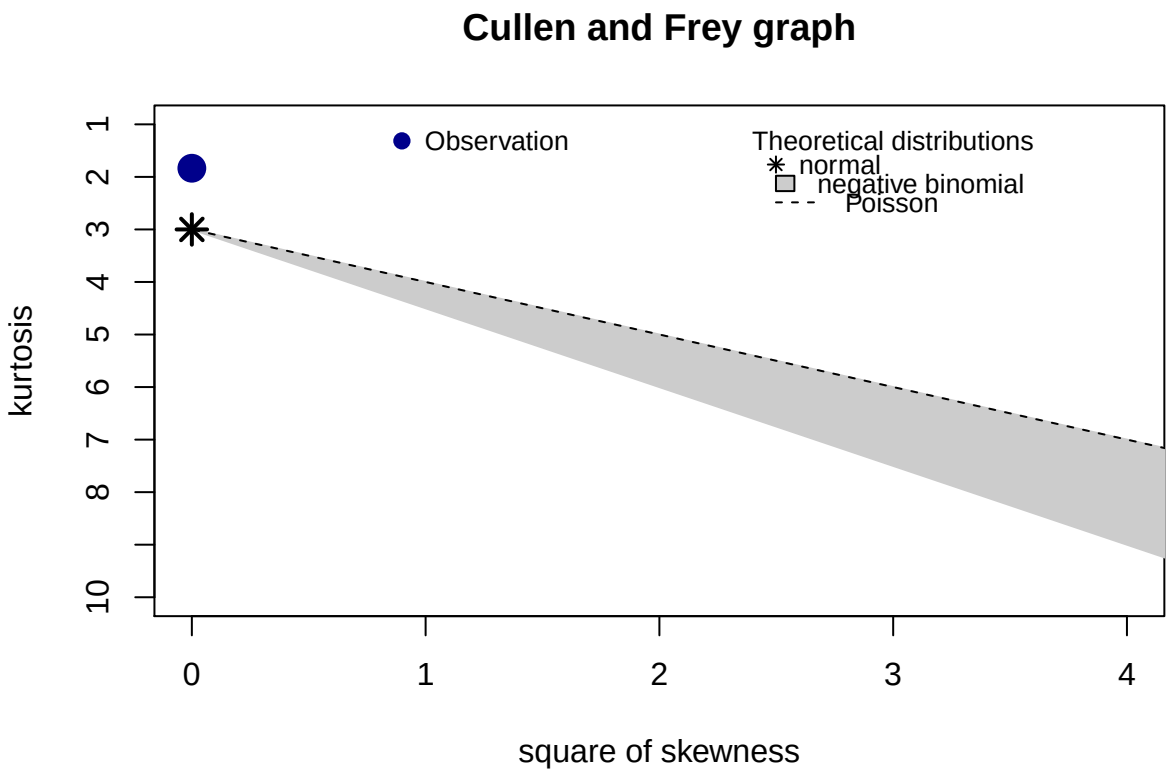


The normal distribution fits the closest and should be selected for the simulation.

4.4 Drive Time

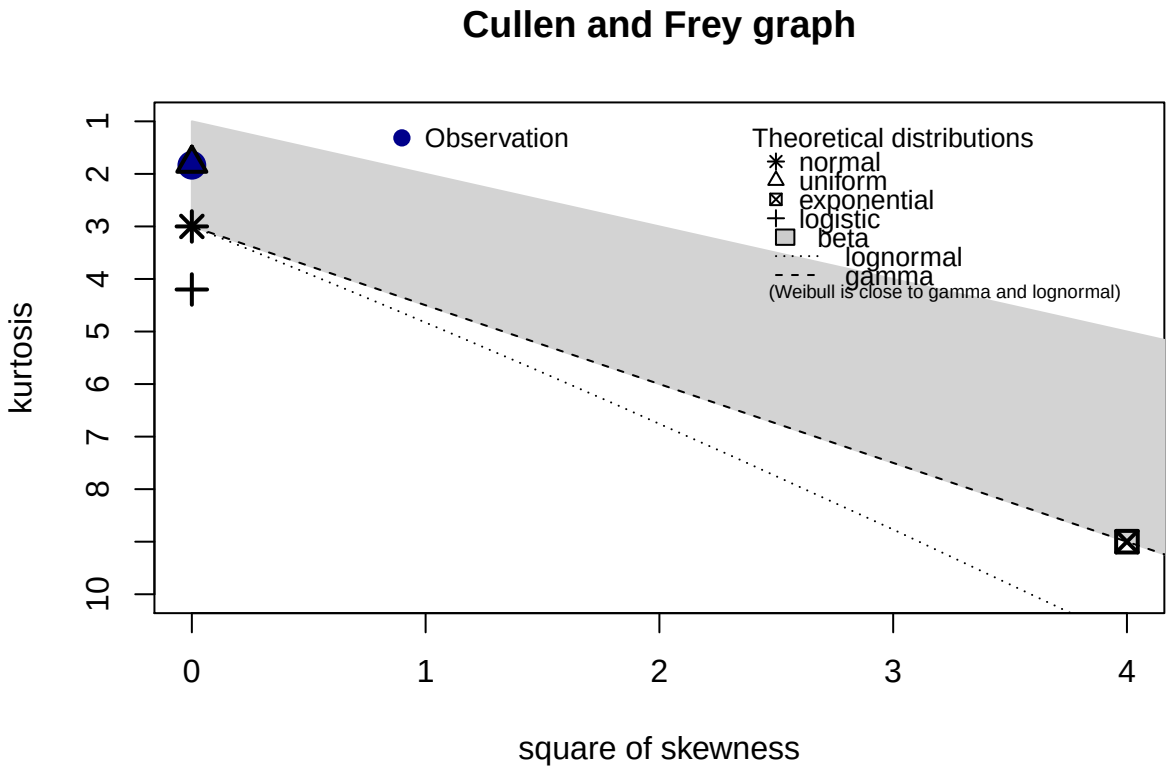
Drive time appears continuous in the data and does not fall within any theoretical discrete distribution. If we examine the graph, the observation lands directly on uniform distribution but we will also check the normal distribution for comparison.

```
descdist(windata$drive.time, discrete = TRUE)
```




```
## summary statistics
## ------
## min:  16.6   max:  29.7
## median:  23.45
## mean:  23.372
## estimated sd:  3.844779
## estimated skewness:  0.008484737
## estimated kurtosis:  1.835217
```

```
descdist(windata$drive.time, discrete = FALSE)
```




```
## summary statistics
## ------
## min:  16.6   max:  29.7
## median:  23.45
## mean:  23.372
## estimated sd:  3.844779
## estimated skewness:  0.008484737
## estimated kurtosis:  1.835217
```

As expected, the uniform distribution has the highest log-likelihood and I would feel comfortable recommending it at this point.

```
normal = fitdist(windata$drive.time, "norm")
summary(normal)
```

```
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##       estimate Std. Error
## mean 23.372000  0.2711864
## sd    3.835155  0.1917577
## Loglikelihood:  -552.6297   AIC:  1109.259   BIC:  1115.856
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1
```

```r
unif = fitdist(windata$drive.time, "unif")
summary(unif)
```

```
## Fitting of the distribution ' unif ' by maximum likelihood
## Parameters :
##     estimate Std. Error
## min     16.6         NA
## max     29.7         NA
## Loglikelihood:  -514.5224   AIC:  1033.045   BIC:  1039.642
## Correlation matrix:
## [1] NA
```

5 Problem 5

Using the "Prestige" dataset, estimate the following for the "income" variable using bootstrapping:

5.0.1 a) True median and 95% CI for the median

To see how effective bootstrapping can be, we will first examine the true median of the income variable in the "Prestige" dataset:

```r
library("car")
data("Prestige")
income <- Prestige$income

median(income)
```

```
## [1] 5930.5
```

By sampling our data with replacement, we can derive a useful estimate of the confidence interval for median. The actual median of 5930.5 sits nicely between our median CI of 5239.5 - 6573.0.

```r
boot_med <- function(x){
  median(sample(x, replace = TRUE))
}

sim = 2000
income_replication <- replicate(sim, boot_med(income))
quantile(income_replication, c(0.05, 0.95))
```

```
##     5%    95%
## 5239.5 6573.0
```

5.0.2 b) True standard deviation and 95% CI for the standard deviation

Again, the true standard deviation gives us a reference point to compare the bootstrapped solution:

```r
sd(income)
```

```
## [1] 4245.922
```

This median is almost in the exact center of our confidence interval for standard deviation:

```r
boot_sd <- function(x){
  sd(sample(x, replace = TRUE))
}

sd_replication <- replicate(sim, boot_sd(income))

quantile(sd_replication, c(0.05, 0.95))
```

```
##       5%      95%
## 3191.897 5147.548
```