Package: ExpRep (via r-universe)

September 10, 2024

Type Package

Title Experiment Repetitions

Version 1.0

Date 2017-06-22

Author Larisa Zamora-Matamoros and Jorge Diaz-Silvera

Maintainer Larisa Zamora-Matamoros <larisa@uo.edu.cu>

Description Allows to calculate the probabilities of occurrences of an event in a great number of repetitions of Bernoulli experiment, through the application of the local and the integral theorem of De Moivre Laplace, and the theorem of Poisson. Gives the possibility to show the results graphically and analytically, and to compare the results obtained by the application of the above theorems with those calculated by the direct application of the Binomial formula. Is basically useful for educational purposes.

License Unlimited

NeedsCompilation no

Repository CRAN

Date/Publication 2017-06-28 15:55:16 UTC

Contents

`ExpRep-package` *Experiment Repetitions*

Description

The package ExpRep, which basically responds to educational purposes, allows to calculate the probabilities of occurrences of an event in a great number of repetitions of Bernoulli experiment, through the application of the local and the integral theorem of De Moivre Laplace, and the theorem of Poisson. It gives the possibility to show the results graphically and analytically, and to compare the results obtained by the application of the above theorems with those calculated by the direct application of the Binomial formula.

Details

The DESCRIPTION file:

Package:	ExpRep
Type:	Package
Title:	Experiment Repetitions
Version:	1.0
Date:	2017-06-22
Author:	Larisa Zamora-Matamoros and Jorge Diaz-Silvera
Maintainer:	Larisa Zamora-Matamoros <larisa@uo.edu.cu>
Description:	Allows to calculate the probabilities of occurrences of an event in a great number of repetitions of Bernoulli ex
License:	Unlimited

Index of help topics:

```
ApplicIntegralTheo      Applications of the Integral Theorem of
                        DeMoivre-Laplace.
Buffon                  Buffon
ExpRep-package          Experiment Repetitions
Integral_Theorem        Integral Theorem of DeMoivre-Laplace
Local_Theorem           Local Theorem of DeMoivre-Laplace
Poisson_Theorem         Poisson Theorem.
S_Integral_Theorem      Simulations of the Integral Theorem of
                        DeMoivre-Laplace.
S_Local_Limit_Theorem   Simulations of Local Theorem of
                        DeMoivre-Laplace
S_Poisson_Theorem       Simulations of Poisson Theorem
```

Author(s)

Larisa Zamora-Matamoros and Jorge Diaz-Silvera

Maintainer: Larisa Zamora-Matamoros <larisa@uo.edu.cu>

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

Examples

```
ProbL<-Local_Theorem(n=100,m=50,p=0.02)
ProbL

ProbI<-Integral_Theorem(n=100,p=0.5,linf=0,lsup=50)
ProbI

ProbP<-Poisson_Theorem(n=100,m=50,p=0.002)
ProbP

beta<-ApplicIntegralTheo(Applic="beta",n=369,p=0.4,alpha=0.05)
beta

alpha<-ApplicIntegralTheo(Applic="alpha",n=369,p=0.4,beta=0.95)
alpha

n<-ApplicIntegralTheo(Applic="n",p=0.4,alpha=0.05,beta=0.95)
n

S_Local_Limit_Theorem(n = 170, p = 0.5, Compare = TRUE, Table = TRUE, Graph = TRUE,
      GraphE = TRUE)

S_Poisson_Theorem(n = 169, p = 0.002, Compare = TRUE, Table = TRUE, Graph = TRUE,
      GraphE = TRUE)

S_Integral_Theorem(n=100, p=0.5, linf = 0, lsup = 50, Compare = TRUE, Table = TRUE,
      Graph = TRUE, GraphE = TRUE)

Buffon(p = 0.5, width = 0.2, r = c(100, 500, 1000, 1500))
```

`ApplicIntegralTheo` *Applications of the Integral Theorem of DeMoivre-Laplace.*

Description

This function shows three applications of the integral theorem of DeMoivre-Laplace: 1. To estimate the probability (beta) that the frequency of occurrence of the successful event will deviate from the probability that this event will happen in any single Bernoulli experiment (p) in a quantity not bigger than alpha. 2. To calculate the least number of experiments that must be carried out (n). 3. To determine the boundary of possible variations between the frequency of occurrence of the successful event and the probability p (alpha).

Usage

```
ApplicIntegralTheo(Applic = "alpha", n = 10000, p = 0.5, alpha = 0.01, beta = 0.9)
```

Arguments

`Applic`	It indicates the calculation to be carried out: if "n", the function calculates the number of repetitions, if "alpha", the function calculates the boundary of possible variations of abs(frequency-p) if "beta", the function calculates the probability that the frequency of occurrence of the successful event will deviate from the probability p by no more than alpha.
`n`	Number of repetitions of the Bernoulli trial.
`p`	Probability that a successful event happens in any single Bernoulli trial (called the probability of success).
`alpha`	The boundary of possible variations of abs(frequency-p).
`beta`	Probability that the frequency of occurrence of the successful event will deviate from the probability p by no more than alpha.

Value

Numeric value representing the values of n, alpha or beta according to the value that the parameter "Applic" takes ("n", "alpha" or "beta").

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

See Also

Integral_Theorem.

Examples

```
beta<-ApplicIntegralTheo(Applic="beta",n=369,p=0.4,alpha=0.05)
beta

alpha<-ApplicIntegralTheo(Applic="alpha",n=369,p=0.4,beta=0.95)
alpha

n<-ApplicIntegralTheo(Applic="n",p=0.4,alpha=0.05,beta=0.95)
n

## The function is currently defined as
function (Applic = "alpha", n = 10000, p = 0.5, alpha = 0.01,
    beta = 0.9)
{
```

```
  Alpha <- function(n, p, beta) {
      a <- (beta + 1)/2
      alpha <- ((p * (1 - p)/n)^0.5) * qnorm(a)
      return(alpha)
  }
  Beta <- function(n, p, alpha) {
      b <- alpha * (n/(p * (1 - p)))^0.5
      beta <- 2 * pnorm(b) - 1
      return(beta)
  }
  Repetitions <- function(p, alpha, beta) {
      a <- (beta + 1)/2
      n <- (p * (1 - p) * ((qnorm(a)/alpha)^2))%/%1 + 1
      return(n)
  }
  options(digits = 17)
  value <- switch(Applic, alpha = Alpha(n, p, beta), beta = Beta(n,
      p, alpha), n = Repetitions(p, alpha, beta))
  return(value)
}
```

Buffon *Buffon*

Description

Simulations of the experiment of Buffon.

Usage

```
Buffon(p = 0.5, width = 0.2, r = c(100, 500, 1000, 1500))
```

Arguments

`p`	Probability of occurrence of some event.
`width`	Width of the band where the probabilities are represented.
`r`	Array of four values, representing the numbers of repetitions of the experiment that will be carried out.

Value

Four graphics, each one is the simulation of the experiment of Buffon for the number of repetitions contained in the array r.

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

Examples

```
Buffon(p = 0.5, width = 0.2, r = c(100, 500, 1000, 1500))

## The function is currently defined as
function (p = 0.5, width = 0.2, r = c(100, 500, 1000, 1500))
{
    Position <- function(k, colum) {
        PE <- k%/%colum
        Resto <- k%%colum
        if (Resto == 0) {
            fila <- PE
            columna <- colum
        }
        else {
            fila <- PE + 1
            columna <- Resto
        }
        Position <- list(fila, columna)
        return(Position)
    }
    nf <- layout(matrix(c(1, 2, 3, 4), 2, 2, byrow = TRUE), TRUE)
    k <- 0
    la <- p - width
    lb <- p + width
    if (la < 0)
        la <- 0
    if (lb > 1)
        lb <- 1
    for (j in 1:4) {
        k <- k + 1
        Probcara <- array(0, dim = r[j])
        for (i in 1:r[j]) {
            binomial <- rbinom(i, 1, p)
            cara <- length(binomial[binomial == 1])
            Probcara[i] <- cara/i
        }
        P <- Position(k, 2)
        fila <- P[[1]]
        colum <- P[[2]]
        mfg <- c(fila, colum, 2, 2)
        a <- as.character(r[j])
        plot(Probcara, type = "p", main = paste0("n=", a), xlab = "Repetitions",
            ylab = "Probability", font.main = 3, col = "blue",
            ylim = c(la, lb))
```

```
        abline(h = p, col = "red", lty = 1, lwd = 2)
      }
    }
```

Integral_Theorem *Integral Theorem of DeMoivre-Laplace*

Description

Given n Bernoulli experiments, with success probability p, this function calculates the probability that a successful event occurs between linf and lsup times.

Usage

```
Integral_Theorem(n = 100, p = 0.5, linf = 0, lsup = 50)
```

Arguments

n	An integer value representing the number of repetitions of the Bernoulli experiment.
p	A real value with the probability that a successful event will happen in any single Bernoulli experiment (called the probability of success).
linf	An integer value representing the minimum number of times that the successful event should happen.
lsup	An integer value representing the maximum number of times that the successful event should happen.

Details

Bernoulli experiments are sequences of events, in which successive experiments are independent and at each experiment the probability of appearance of a "successful" event (p) remains constant. The value of n must be high and the value of p must be small. It is necessary that linf < lsup.

Value

A real value representing the approximate probability that a successful event occurs between linf and lsup times, in n repetitions of a Bernoulli experiment.

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

See Also

Poisson_Theorem, Local_Theorem.

Examples

```
Prob<-Integral_Theorem(n=100,p=0.5,linf=0,lsup=50)
Prob

## The function is currently defined as
function (n = 100, p = 0.5, linf = 0, lsup = 50)
{
    A <- (linf - n * p)/sqrt(n * p * (1 - p))
    B <- (lsup - n * p)/sqrt(n * p * (1 - p))
    P <- pnorm(B) - pnorm(A)
    return(P)
  }
```

`Local_Theorem` *Local Theorem of DeMoivre-Laplace*

Description

Given n Bernoulli experiments, with success probability p, this function calculates the approximate probability that a successful event occurs exactly m times.

Usage

```
Local_Theorem(n, m, p)
```

Arguments

n	An integer value representing the number of repetitions of the Bernoulli experiment.
m	An integer value representing the number of times that a successful event occurs in the n repetitions of the Bernoulli experiment.
p	A real value with the probability that a successful event will happen in any single Bernoulli experiment (called the probability of success).

Details

Bernoulli experiments are sequences of events, in which successive experiments are independent and at each experiment the probability of appearance of a "successful" event (p) remains constant. The value of n must be high and the value of p must be small.

Value

A real value representing the approximate probability that a successful event occurs exactly m times in n repetitions of a Bernoulli experiment.

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

See Also

Integral_Theorem, Poisson_Theorem.

Examples

```
Prob<-Local_Theorem(n=100,m=50,p=0.02)
Prob

## The function is currently defined as
function (n, m, p)
{
    a <- n * p
    b <- sqrt(a * (1 - p))
    x <- (m - a)/b
    P <- dnorm(x, 0, 1)/b
    return(P)
  }
```

Poisson_Theorem *Poisson Theorem.*

Description

Given n Bernoulli experiments, with success probability p (p small), this function calculates the approximate probability that a successful event occurs exactly m times.

Usage

```
Poisson_Theorem(n, m, p)
```

Arguments

- `n` — An integer value representing the number of repetitions of the Bernoulli experiment.
- `m` — An integer value representing the number of times that a successful event occurs in the n repetitions of the Bernoulli experiment.
- `p` — A real value with the probability that a successful event will happen in any single Bernoulli experiment (called the probability of success).

Details

Bernoulli experiments are sequences of events, in which successive experiments are independent and at each experiment the probability of appearance of a "successful" event (p) remains constant. The value of n must be high and the value of p must be very small.

Value

A numerical value representing the approximate probability that a successful event occurs exactly m times.

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

See Also

Integral_Theorem, Local_Theorem.

Examples

```
Prob<-Poisson_Theorem(n=100,m=50,p=0.002)
Prob

## The function is currently defined as
function (n, m, p)
{
    landa <- n * p
    P <- dpois(m, landa)
    return(P)
  }
```

`S_Integral_Theorem` *Simulations of the Integral Theorem of DeMoivre-Laplace.*

Description

Given n Bernoulli experiments with success probability p, this function calculates and plots the exact probability and the approximate probability that a successful event occurs between linf+i (0<=i<=lsup-linf-1) and lsup times. It also calculates the difference between these probabilities and shows all the computations in a table.

Usage

```
S_Integral_Theorem(n = 200, p = 0.5, linf = 0, lsup = 100, Compare = TRUE,
     Table = TRUE, Graph = TRUE, GraphE = TRUE)
```

Arguments

`n`	An integer vector with the numbers of repetitions of the Bernoulli experiment.
`p`	A real value with the probability that a successful event will happen in any single Bernoulli experiment (called the probability of success).
`linf`	An integer vector, of the same length than n, with the minimum numbers of times that the successful event should happen.
`lsup`	An integer vector, of the same length than n, with the maximum numbers of times that the successful event should happen.
`Compare`	A logical value, if TRUE the function calculates both the exact probability and the approximate probability that a successful event occurs and compares these probabilities.
`Table`	A logical value, if TRUE the function shows a table with the carried out computations.
`Graph`	A logical value, if TRUE the function plots both the exact probability and the approximate probability that a successful event occurs.
`GraphE`	A logical value, if TRUE the function shows the graphic of the errors in the approximation.

Details

Bernoulli experiments are sequences of events, in which successive experiments are independent and at each experiment the probability of appearance of a "successful" event (p) remains constant. It is necessary that linf < lsup.

Value

A graph and/or a table.

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

See Also

Integral_Theorem, Local_Theorem.

Examples

```
S_Integral_Theorem (n = 200, p = 0.5, linf = 0, lsup = 100, Compare = TRUE, Table = TRUE,
    Graph = TRUE, GraphE = TRUE)

## The function is currently defined as
function (n = 200, p = 0.5, linf = 0, lsup = 100, Compare = TRUE,
    Table = TRUE, Graph = TRUE, GraphE = TRUE)
  { Integral_Theorem <- function(n = 100, p = 0.5, linf = 0,
        lsup = 50) {
        A <- (linf - n * p)/sqrt(n * p * (1 - p))
        B <- (lsup - n * p)/sqrt(n * p * (1 - p))
        P <- pnorm(B) - pnorm(A)
        return(P)
    }
    layout(matrix(1))
    PNormal <- numeric()
    Dif <- numeric()
    PBin <- numeric()
    k <- lsup - linf - 1
    PNormal[1] <- Integral_Theorem(n, p, linf, lsup)
    PBin[1] <- 0
    for (j in linf:lsup) PBin[1] <- PBin[1] + dbinom(j, n, p)
    Dif[1] <- abs(PBin[1] - PNormal[1])
    for (i in 1:k) {
        linf_i <- linf + i
        PNormal[i + 1] <- Integral_Theorem(n, p, linf_i, lsup)
        if (Compare == TRUE) {
            PBin[i + 1] <- 0
            for (j in linf_i:lsup) PBin[i + 1] <- PBin[i + 1] +
                dbinom(j, n, p)
            Dif[i + 1] <- abs(PBin[i + 1] - PNormal[i + 1])
        }
    }
    if (Graph == TRUE & GraphE == TRUE) {
        layout(matrix(c(1, 1, 2, 2), 2, 2, byrow = TRUE))
    }
```

```
    if (Graph == TRUE) {
        ymini <- min(PNormal[k + 1], PBin[k + 1]) - 0.05
        ymaxi <- max(PNormal[1], PBin[1]) + 0.05
        mfg <- c(1, 1, 2, 2)
      plot(PNormal, ylim = c(ymini, ymaxi), type = "l", main = "The Integral Limit Theorem",
            xlab = "k (linf<=k<=lsup)", ylab = "Probability",
            col = "red")
        mtext("Integral Theorem", line = -1, side = 1, adj = 1,
            col = "red")
        if (Compare == TRUE) {
            points(PBin, type = "p", col = "blue")
            mtext("Binomial Probability", line = -2, side = 1,
                adj = 1, col = "blue")
        }
    }
    if (GraphE == TRUE) {
        mfg <- c(2, 1, 2, 2)
        dmini <- min(Dif) - 0.01
        dmaxi <- max(Dif) + 0.01
        plot(Dif, ylim = c(dmini, dmaxi), type = "b", main = "Errors",
            xlab = "m", ylab = "Errors", col = "green")
        abline(a = 0, b = 0, col = "red")
    }
    if (Table == TRUE) {
        Ak <- array(1:(k + 1))
        if (Compare == TRUE)
            TablaR <- data.frame(k = Ak, PBinomial = PBin, T_Integral = PNormal,
                Difference = Dif)
        else TablaR <- data.frame(K = Ak, T_Integral = PNormal)
        TablaR
    }
  }
```

S_Local_Limit_Theorem *Simulations of Local Theorem of DeMoivre-Laplace*

Description

Given n Bernoulli experiments, with success probability p, this function calculates and plots the exact probability and the approximate probability that a successful event occurs exactly m times (0<=m<=n). It also calculates the difference between these probabilities and shows all the computations in a table.

Usage

```
S_Local_Limit_Theorem(n = 170, p = 0.5, Compare = TRUE, Table = TRUE,
     Graph = TRUE, GraphE = TRUE)
```

Arguments

`n`	An integer value representing the number of repetitions of the experiment.
`p`	A real value with the probability that a successful event will happen in any single Bernoulli experiment (called the probability of success).
`Compare`	A logical value, if TRUE the function calculates both the exact probability and the approximate probability that a successful event occurs exactly m times and compares these probabilities.
`Table`	A logical value, if TRUE the function shows a table with the carried out computations.
`Graph`	A logical value, if TRUE the function plots both the exact probability and the approximate probability that a successful event occurs exactly m times.
`GraphE`	A logical value, if TRUE the function shows the graphic of the errors in the approximation.

Details

Bernoulli experiments are sequences of events, in which successive experiments are independent and at each experiment the probability of appearance of a "successful" event (p) remains constant. The value of n must be high and the value of p must be small.

Value

A graph and/or a table.

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

See Also

Integral_Theorem, Local_Theorem.

Examples

```
S_Local_Limit_Theorem(n = 170, p = 0.5, Compare = TRUE, Table = TRUE, Graph = TRUE,
     GraphE = TRUE)

## The function is currently defined as
function (n = 170, p = 0.5, Compare = TRUE, Table = TRUE, Graph = TRUE, GraphE = TRUE)
  { layout(matrix(1))
```

```
    m <- array(0:n)
    x <- numeric()
    PNormal <- numeric()
    a <- n * p
    b <- sqrt(a * (1 - p))
    for (mi in 1:(n + 1)) {
        x[mi] <- (mi - 1 - a)/b
        PNormal[mi] <- dnorm(x[mi], 0, 1)/b
    }
    if (Compare == TRUE) {
        PBin <- numeric()
        for (mi in 1:(n + 1)) PBin[mi] <- dbinom(mi - 1, n, p)
        Dif <- abs(PBin - PNormal)
    }
    if (Graph == TRUE & GraphE == TRUE) {
        layout(matrix(c(1, 1, 2, 2), 2, 2, byrow = TRUE))
    }
    if (Graph == TRUE) {
        mfg <- c(1, 1, 2, 2)
        plot(PNormal, type = "p", main = "The Local Limit Theorem",
            xlab = "m", ylab = "Probability", col = "red")
        mtext("Local Theorem", line = -1, side = 3, adj = 1,
            col = "red")
        if (Compare == TRUE) {
            points(PBin, type = "p", col = "blue")
            mtext("Binomial Probability", line = -2, side = 3,
                adj = 1, col = "blue")
        }
    }
    if (GraphE == TRUE) {
        mfg <- c(2, 1, 2, 2)
        dmini <- min(Dif) - 0.01
        dmaxi <- max(Dif) + 0.01
        plot(Dif, ylim = c(dmini, dmaxi), type = "b", main = "Errors",
            xlab = "m", ylab = "Errors", col = "green")
        abline(a = 0, b = 0, col = "red")
    }
    if (Table == TRUE) {
        if (Compare == TRUE)
            TablaR <- data.frame(m = m, x = x, PBinomial = PBin,
                TLocal = PNormal, Difference = Dif)
        else TablaR <- data.frame(m = m, x = x, TLocal = PNormal)
        TablaR
    }
  }
```

S_Poisson_Theorem *Simulations of Poisson Theorem*

Description

Given n Bernoulli experiments, with success probability p, this function calculates and plots the exact probability and the approximate probability that a successful event occurs exactly m times (0<=m<=n). It also calculates the difference between theses probabilities and shows all the computations in a table.

Usage

```
S_Poisson_Theorem(n = 2000, p = 0.002, Compare = TRUE, Table = TRUE,
     Graph = TRUE, GraphE = FALSE)
```

Arguments

`n`	An integer value representing the number of repetitions of the Bernoulli experiment.
`p`	A real value with the probability that a successful event will happen in any single Bernoulli experiment (called the probability of success)
`Compare`	A logical value, if TRUE the function calculates both the exact probability and the approximate probability that a successful event occurs exactly m times and compares these probabilities.
`Table`	A logical value, if TRUE the function shows a table with the carried out computations.
`Graph`	A logical value, if TRUE the function plots both the exact probability and the approximate probability that a successful event occurs exactly m times.
`GraphE`	A logical value, if TRUE, the function shows the graphics corresponding to the differences between the probabilities using the Binomial approach and Poisson theorem, and the probabilities using the Binomial approach and Local theorem.

Details

Bernoulli experiments are sequences of events, in which successive experiments are independent and at each experiment the probability of appearance of a "successful" event (p) remains constant. The value of n must be high and the value of p must be very small.

Value

A graph and/or a table.

Note

Department of Mathematics. University of Oriente. Cuba.

Author(s)

Larisa Zamora and Jorge Diaz

References

Gnedenko, B. V. (1978). The Theory of Probability. Mir Publishers. Moscow.

See Also

Integral_Theorem, Local_Theorem.

Examples

```
S_Poisson_Theorem(n = 169, p = 0.002, Compare = TRUE, Table = TRUE, Graph = TRUE,
    GraphE = TRUE)
## The function is currently defined as
function (n = 2000, p = 0.002, Compare = TRUE, Table = TRUE,
    Graph = TRUE, GraphE = FALSE)
  { layout(matrix(1))
    m <- array(0:n)
    PPoisson <- numeric()
    a <- n * p
    for (mi in 1:(n + 1)) PPoisson[mi] <- dpois(mi - 1, a)
    if (Compare == TRUE) {
        PBin <- numeric()
        x <- numeric()
        PNormal <- numeric()
        Dif1 <- numeric()
        Dif2 <- numeric()
        b <- sqrt(a * (1 - p))
        for (mi in 1:(n + 1)) {
            x[mi] <- (mi - 1 - a)/b
            PBin[mi] <- dbinom(mi - 1, n, p)
            PNormal[mi] <- dnorm(x[mi], 0, 1)/b
        }
        Dif1 <- abs(PBin - PPoisson)
        Dif2 <- abs(PBin - PNormal)
    }
    if (Graph == TRUE & GraphE == TRUE) {
        layout(matrix(c(1, 1, 2, 2), 2, 2, byrow = TRUE))
    }
    if (Graph == TRUE) {
        mfg <- c(1, 1, 2, 2)
        ll <- length(which(Dif1 > 5e-07))
        plot(PPoisson[1:ll], type = "b", main = "The Poisson Theorem",
            xlab = "m", ylab = "Probability", col = "red")
        mtext("Poisson Theorem", line = -1, side = 3, adj = 1,
            col = "red")
        if (Compare == TRUE) {
            points(PBin[1:ll], type = "b", col = "green")
            points(PNormal[1:ll], type = "b", col = "blue")
            mtext("Local Theorem", line = -2, side = 3, adj = 1,
                col = "blue")
            mtext("Binomial Probability", line = -3, side = 3,
                adj = 1, col = "green")
        }
```

```
  }
  if (GraphE == TRUE) {
      mfg <- c(2, 1, 2, 2)
      ll <- length(which(Dif1 > 5e-07))
      plot(Dif2[1:ll], type = "b", main = "Errors", xlab = "m",
          ylab = "Differences", col = "red")
      mtext("Binomial-Poisson", line = -1, side = 3, adj = 1,
          col = "red")
      points(Dif1[1:ll], type = "b", col = "green")
      mtext("Binomial-Local Theorem", line = -2, side = 3,
          adj = 1, col = "green")
  }
  if (Table == TRUE) {
      if (Compare == TRUE)
          TablaR <- data.frame(m = m, x = x, PBinomial = PBin,
              TPoisson = PPoisson, Difference1 = Dif1, TLocal = PNormal,
              Difference2 = Dif2)
      else TablaR <- data.frame(m = m, TPoisson = PPoisson)
      TablaR
  }
}
```

Index