Assignment 2

Method of Moment Estimator for probability distributions

In statistics, the method of moments is a method of estimation of population parameters.

In [2]:
```
#import libraries
library(Rlab)
library(MASS)

options(warn=-1)
```

Method of Moment Estimator Function

In [3]:

```
Method_of_Moment_Generator <- function(dist, X)
{
    #Child Function for Bernoilli Method Of Moment estimation
    Bernoulli <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        p <- Xbar
        sprintf("p = %s", p)
    }


    #Child Function for Binomial Method Of Moment estimation
    Binomial <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        SD2 <- sum((X - Xbar)^2)/n
        P <- 1 - (SD2/Xbar)
        N <- Xbar/P
        sprintf("N = %s & P = %s", N, P)
    }


    #Child Function for Geometric Method Of Moment estimation
    Geometric <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        P <- 1/Xbar
        sprintf("P = %s", P)
    }


    #Child Function for Poisson Method Of Moment estimation
    Poisson <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        lambda <- Xbar
        sprintf("lambda = %s", lambda)
    }


    #Child Function for Uniform Method Of Moment estimation
    Uniform <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        SD2 <- sum((X - Xbar)^2)/n
        A <- ((2*Xbar) - sqrt(12*SD2))/2
        B <- (sqrt(12*SD2) + (2*Xbar))/2
        sprintf("A = %s & B = %s", A, B)
    }


```

```
    #Child Function for Normal Method Of Moment estimation
    Normal <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        mu <- Xbar
        SD2 <- sum((X - mu)^2)/n
        sigma_sq <- SD2
        sprintf("mu = %s & sigma sq = %s", mu, sigma_sq)
    }


    #Child Function for Exponential Method Of Moment estimation
    Exponential <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        beta <- Xbar
        sprintf("beta = %s", beta)
    }


    #Child Function for Gamma Method Of Moment estimation
    Gamma <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        SD2 <- sum((X - Xbar)^2)/n
        alpha <- (Xbar^2)/SD2
        beta <- SD2/Xbar
        sprintf("alpha = %s & beta = %s", alpha, 1/beta)
    }


    #Child Function for Beta Method Of Moment estimation
    Beta <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        Sd2 <- sum((X - Xbar)^2)/n
        alpha <- Xbar*(((Xbar*(1-Xbar))/Sd2) - 1)
        beta <- (1 - Xbar) * (((Xbar*(1-Xbar))/Sd2) - 1)
        sprintf("alpha = %s & beta sq = %s", alpha, beta)
    }


    #Child Function for T Method Of Moment estimation
    T <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        delta <- Xbar
        sprintf("delta = %s", delta)
    }


    #Child Function for chi2 Method Of Moment estimation
```

```
    Chi2 <- function(X)
    {
        n <- length(X)
        Xbar <- sum(X)/n
        p <- Xbar
        sprintf("p = %s", p)
    }


    multinomialnormal <- function(X)
    {

      a=nrow(X)
      p<-c(0,0,0,0,0)
      for(i in 1:a)
        p[i]<-1-((var(X[i,]))/mean(X[i,]))
      n=sum(rowMeans(X))/sum(p[1:a])
      #n=mean(c(rowMeans(x)/p[1:a]))
      sprintf("For Multinomial Distribution the parameter 'n' is %s and the p

    }

    multivariatenormal <- function(X)
    {

      mu<- colMeans(X)
      summ<- var(X)
      sprintf("For Multivariate Normal Distribution the parameters 'mu' is %s
    }

    switch(dist,'Bernoulli' = Bernoulli(X), 'Binomial' = Binomial(X), 'Geomet
}
```

Calling Method of Moment Estimator for probability distributions

In [5]:

```
#Bernoulli
n <- 1000
prob <- 0.3
bernlist <- rbern(n, prob)
Method_of_Moment_Generator('Bernoulli',bernlist)
```

'p = 0.293'

In [6]:

```
#Binomial
n <- 1000
size <- 10
prob <- 0.3
binomlist <- rbinom(n, size, prob)
Method_of_Moment_Generator('Binomial',binomlist)
```

'N = 9.8343299702327 & P = 0.310646481178393'

In [7]:

```
#Geometric
n <- 1000
size <- 10
prob <- 0.3
geomlist <- rgeom(n, prob)
Method_of_Moment_Generator('Geometric',geomlist)
```

'P = 0.435540069686411'

In [8]:

```
#poisson
n <- 1000
size <- 10
prob <- 0.3
poilist <- rpois(n, prob)
Method_of_Moment_Generator('Poisson',poilist)
```

'lambda = 0.305'

In [9]:

```
#uniform
n <- 1000
min <- 1
max <- 5
unilist <- runif(n, min = min, max = max)
Method_of_Moment_Generator('Uniform',unilist)
```

'A = 0.977548375020186 & B = 4.96344273037239'

In [10]:

```
#normal
n <- 1000
mean <- 0
sd <- 3
normlist <- rnorm(n, mean = mean, sd = sd)
Method_of_Moment_Generator('Normal',normlist)
```

'mu = -0.0349601434557909 & sigma sq = 8.6768679098428'

In [11]:

```
#exponential
n <- 1000
rate <- 5
explist <- rexp(n, rate = rate)
Method_of_Moment_Generator('Exponential',explist)
```

'beta = 0.197318175094277'

In [12]:

```
#beta
n <- 1000
shape1 <- 1
shape2 <- 2
betalist <- rbeta(n,shape1, shape2)
Method_of_Moment_Generator('Beta',betalist)
```

'alpha = 0.977004556324639 & beta sq = 2.07107880170482'

In [13]:
```
#gamma
n <- 1000
alpha <- 1
beta <- 5
gammalist <- rgamma(n,alpha, beta)
Method_of_Moment_Generator('Gamma', gammalist)
```

'alpha = 0.931780236934039 & beta = 4.48202989647715'

In [14]:
```
#T Distribution
n <- 1000
df <- 9
tlist <- rt(n, df)
Method_of_Moment_Generator('T', tlist)
```

'delta = 0.0337822505642689'

In [4]:
```
#ChiSquare
n <- 1000
df <- 9
chilist <- rchisq(n, df, ncp = 0)
Method_of_Moment_Generator('Chi2',chilist)
```

'p = 8.93094923179235'

In [15]:
```
#multivariatenormal
n <- 1000
Sigma <- matrix(c(10,3,3,2),2,2)
multivariatelist <- mvrnorm(n, rep(0, 2), Sigma)
Method_of_Moment_Generator('multivariateNormal', multivariatelist)
```

'For Multivariate Normal Distribution the parameters \'mu\' is 0.058261958734577 and Summation is 9.58039073208182'
'For Multivariate Normal Distribution the parameters \'mu\' is 0.0537489267292221 and Summation is 2.85490523179302'
'For Multivariate Normal Distribution the parameters \'mu\' is 0.058261958734577 and Summation is 2.85490523179302'
'For Multivariate Normal Distribution the parameters \'mu\' is 0.0537489267292221 and Summation is 1.88464504369496'

In [16]:
```
#multinomimalnormal
n <- 1000
size <- 5
multinominallist <- rmultinom(n,size,c(0.15,0.05,0.4,0.1,0.3))
Method_of_Moment_Generator('multinominalNormal', multinominallist)
```

'For Multinomial Distribution the parameter \'n\' is 5.36811977146384 and the parameter \'p\' is 0.167968238143679'
'For Multinomial Distribution the parameter \'n\' is 5.36811977146384 and the parameter \'p\' is -0.0213958814781661'
'For Multinomial Distribution the parameter \'n\' is 5.36811977146384 and the parameter \'p\' is 0.446825466242943'
'For Multinomial Distribution the parameter \'n\' is 5.36811977146384 and the parameter \'p\' is 0.0540227727727896'
'For Multinomial Distribution the parameter \'n\' is 5.36811977146384 and the parameter \'p\' is 0.284004225832479'