# Assignment_5

GROUP PROJECT: SriKaavya Toodi(st938), Rithvik (ra830),Aravind reddy(ad1452),Sowmya(sa1607)

10/13/2020

### Function to estimate

Method to estimate parameters for different distributions is below. Input parameters are "distribution_type" : Type of distribution. To be given as a string input "data_input" : Data generated using the specific distribution to be given as input

Mean and variance and given data is calculated and using Method of moments, calculations will be done to estimate specific parameter values. These will be printed after calculations.

We've to note that calculating variance for the data set implicitly has second moment in it. Using variance for estimating a parameter is almost similar to using second moment of the given data.

```
library(purrr)
library(MASS)
 Parameter_finder <- function( distribution_type , data_input , k) {
  # This function is used to calculate parameters for
  #different distributions of data using mean and variance
  if(!distribution_type=="multinomial"){
   print(head(data_input))
  }
  mean_data = mean(data_input)
  var_data = var(data_input)

  if ( distribution_type == "Poisson") {
    print(paste0("Poisson distribution parameters - Lambda value is ", mean_data))
  }
  else if (  distribution_type == "Point mass at a") {
    print(paste0("Point mass at a distribution parameters - a value is", mean_data))
  }
  else if (  distribution_type == "Bernoulli") {
    print(paste0("Bernoulli distribution parameters - p value is", mean_data))
  }
  else if (  distribution_type == "Binomial") {
    p = var_data/mean_data
    p = 1-p
    n = mean_data/p
    print(paste0("Binomial distribution parameters - p value is ",p))
    print(paste0("N value is ", n))
  }
  else if (  distribution_type == "Geometric") {
    p = 1/mean_data
    print(paste0("Geometric distribution parameters - p value is ", p))
```

```r
}
else if (  distribution_type == "Uniform") {
  #sub_ba = (var_data/mean_data) * 6
  #sum_ab = mean_data * 2
  b = mean_data + (sqrt(3*var_data))
  a = mean_data - (sqrt(3*var_data))
  print(paste0("Uniform distribution parameters - a value is ", a,", b value is ", b))
}
else if (  distribution_type == "Normal") {
  print(paste0("Normal distribution parameters - Mean value is ", mean_data))
  print(paste0("Standard deviation is ", sqrt(var_data)))
}
else if (  distribution_type == "Exponential") {
  print(paste0("Exponential distribution parameters - rate value is ", 1/mean_data))
}
else if (  distribution_type == "Gamma") {
  beta = var_data/mean_data
  alpha = mean_data/beta
  print(paste0("Gamma distribution parameters - alpha value is", alpha))
  print(paste0("Beta value is ",beta, " and rate is ", 1/beta))
}
else if (  distribution_type == "Beta") {
  m <- mean_data
  v <- var_data
  alpha <- m*((m*(1-m)/v)-1)
  beta <- alpha*(1-m)/m
  print(paste0("Beta distribution parameters - alpha value is", alpha))
  print(paste0("Beta value is ",beta))
}
else if (  distribution_type == "tdist") {
  v = 2*var_data/(var_data-1)
  print(paste0("t distribution distribution parameters -v is ", v))
}
else if (  distribution_type == "Chi Square") {
  print(paste0("Chi Square distribution parameters - p value is ", mean_data))
}

else if (distribution_type=="multinomial")
{
  n_row = nrow(data_input)
  prob = c(0,0,0,0)
  for(i in 1:n_row)
    p[i]<-1-((var(data_input[i,]))/mean(data_input[i,]))
  n = sum(rowMeans(data_input))/sum(p[1:n_row])
  print(list(n=n, p=p))
}
else if (distribution_type=="multivariatenormal")
{
  mean = colMeans(data_input)
  Sigma = var(data_input)
  print(list(mu_val=mean, summation=Sigma))
}else if (distribution_type=="hypergeometric") {
  N <- length(data_input)
```

```
    p <- mean_data/k
    product <- var_data/(k * p * (1-p))
    m_plus_n <- (k-product)/(1-product)
    m <- p * m_plus_n
    print(paste0("N value is ", N, " Subclass 1(m) value is ", m, " Subclass 2(n) value is ", m_plus_n-
  }
  #print(paste0('Mean of data is ', mean_data))
  #print(paste0('Variation of data is ', var_data))
}
```

### Testing distributions

Testing different distributions below

```
# Bernoulli is Binomial with size as 1
bernoulli_data <- rbinom(1000, 1, 0.75)
distribution_type = 'Bernoulli'
Parameter_finder(distribution_type,bernoulli_data)
```

```
## [1] 1 1 1 1 0 0
## [1] "Bernoulli distribution parameters - p value is0.768"
```

```
#Binomial distribution
binom_data <- rbinom(100, 1000, 0.75)
distribution_type = 'Binomial'
Parameter_finder(distribution_type,binom_data)
```

```
## [1] 751 760 756 771 743 737
## [1] "Binomial distribution parameters - p value is 0.733397046486386"
## [1] "N value is 1023.78377932824"
```

```
# Geometric distribution. Here parameter couln't be estimated even with really high value of N
geom_data <- rgeom(100000, 0.25)
distribution_type = 'Geometric'
Parameter_finder(distribution_type,geom_data)
```

```
## [1]  2 13  5  8  4  1
## [1] "Geometric distribution parameters - p value is 0.333044694598015"
```

```
# Poisson distribution. Lamdba is accurately estimated
poisson_data <- rpois(40000, lambda = 3)
distribution_type = 'Poisson'
Parameter_finder(distribution_type,poisson_data)
```

```
## [1] 7 3 1 3 1 1
## [1] "Poisson distribution parameters - Lambda value is 3.0032"
```

```
#Uniform distribution data. Here a and B are accurately estimated if gap between them is large enough.
#If not,it seems inaccurate
 uniform_data <- runif(1000000, 1, 100)
distribution_type = 'Uniform'
Parameter_finder(distribution_type,uniform_data)
```

```
## [1] 50.228441 43.208789  5.444854 77.886212 98.108988  3.855769
## [1] "Uniform distribution parameters - a value is 1.06722685249467, b value is 100.024897200006"
```

```r
# Normal distribution
norm_data <- rnorm(100000, 20, 2)
distribution_type = 'Normal'
Parameter_finder(distribution_type,norm_data)
```

```
## [1] 21.87800 22.59300 24.46863 17.50718 18.10413 20.81067
## [1] "Normal distribution parameters - Mean value is 19.9910723775174"
## [1] "Standard deviation is 2.00178609383717"
```

```r
# Exponential distribution
exp_data <- rexp(100000, 5)
distribution_type = 'Exponential'
Parameter_finder(distribution_type,exp_data)
```

```
## [1] 0.14212729 0.10511083 0.01109111 0.10576077 0.33417595 0.43495792
## [1] "Exponential distribution parameters - rate value is 5.02496026245548"
```

```r
# Gamma distribution
gamma_data <- rgamma(10000, 2, 3)
distribution_type = 'Gamma'
Parameter_finder(distribution_type,gamma_data)
```

```
## [1] 0.06970137 0.24377793 0.11733750 0.26149109 0.76327512 0.96985317
## [1] "Gamma distribution parameters - alpha value is1.98672506742025"
## [1] "Beta value is 0.335259177933098 and rate is 2.98276696305553"
```

```r
# Beta distribution
beta_data <- rbeta(10000, 2, 8)
distribution_type = 'Beta'
Parameter_finder(distribution_type,beta_data)
```

```
## [1] 0.24574431 0.09636654 0.12864885 0.05188635 0.07096758 0.12087265
## [1] "Beta distribution parameters - alpha value is2.02712084406855"
## [1] "Beta value is 8.15829719857456"
```

```r
# Student T distribution
tdist_data = rt(100000, 6)
distribution_type = 'tdist'
Parameter_finder(distribution_type,tdist_data)
```

```
## [1]  1.6150679  0.4516914 -0.1660867 -1.8020881 -0.9687962  0.8854000
## [1] "t distribution distribution parameters -v is 5.89084014842776"
```

```r
# CHI-SQUARE dustribution
chi2_data <- rchisq(10000, 5)
distribution_type = 'Chi Square'
Parameter_finder(distribution_type,chi2_data)
```

```
## [1] 2.051394 1.101921 6.353598 3.864569 3.322056 2.678814
## [1] "Chi Square distribution parameters - p value is 4.98725135472518"
```

```r
#Multinomial Distribution
 p = c(0.20,0.40,0.05,0.30)
 data = rmultinom(1000,size=4,p)
 Parameter_finder("multinomial", data)
```

```
## $n
## [1] 4.200003
##
## $p
## [1] 0.15873016 0.42595736 0.04167325 0.32601953
```

```r
# Multi variate normal distribution
Sum = matrix(c(7,3,3,7),2,2)
data = mvrnorm(n = 100000, rep(0, 2), Sum)
distribution_type = 'multivariatenormal'
Parameter_finder(distribution_type, data)
```

```
##            [,1]       [,2]
## [1,] -0.5195364  0.5640343
## [2,]  1.7338070 -0.3226992
## [3,]  2.2374753  5.9403655
## [4,] -1.8997285 -0.9350794
## [5,]  1.9784065  1.3551868
## [6,]  3.5042940  2.0507666
## $mu_val
## [1] -9.143227e-05 -2.677354e-03
##
## $summation
##          [,1]     [,2]
## [1,] 6.984144 2.961936
## [2,] 2.961936 6.957802
```

```r
# Hyper Geometric Distribution
k <- 3
data <- rhyper(40000, 10, 15, 3)
distribution_type = 'hypergeometric'
Parameter_finder(distribution_type, data, k)
```

```
## [1] 0 1 1 2 0 2
## [1] "N value is 40000 Subclass 1(m) value is 9.54858929125295 Subclass 2(n) value is 14.165285477478
```