Proiect final la Probabilitati si Statistica

Leopea Catalina si Misaila Alina  
Grupa: 243

|  |
| --- |
| 1. 2. 3.  N = 1000;  culori = c("red","blue","orange","green","purple");  Bin = function() {  #pt 5 valori diferite ale probabilitatii  probabilitati = c(0.2, 0.5, 0.6, 0.7, 1);  valori = c(100, 200, 300, 400, 500);    x=seq(1, N, 1);  for(i in 1:5) {  #pentru fiecare probabilitate  p = probabilitati[i];  #calculam functia de masa  densitate = dbinom(x, N, p);  if(i == 1){  #la prima iteratie cream graficul  plot(x, densitate, type= "p", col = culori[i]);  obs = rbinom(N, 1, p);  #si calculam media si variatia (vezi simulare 1)  media = mean(obs);  variatia = var(obs);  } else {  #apoi doar adaugam linii la el  lines(x, densitate, type = "p", col = culori[i]);  }  }    legend("bottomright", legend = probabilitati, col = culori, pch = 16, cex = 1);    for(i in 1:5) {  #si facem la fel pentru functia de repartitie  probabilitate = pbinom(x, valori[i], probabilitati[i]);  if(i == 1){  plot(x, probabilitate, type="p", col = culori[i]);  } else {  lines(x, probabilitate, type="p", col = culori[i]);  }  }  legend("bottomright", legend = c("p = 0.2 , n = 100", "p = 0.5 , n = 200", "p = 0.6 , n = 300", "p = 0.7 , n = 400", "p = 1 , n = 500"), col = culori,  pch = 16, cex = 1);  }  Pois = function() {  #pt 5 valori diferite ale lui lambda  lambda = c(5, 10, 15, 20, 25); |

***Problema 1***

|  |  |
| --- | --- |
| for(i in 1:5) {  x = seq(1, 2 \* lambda[i], 1);  densitate = dpois(x, lambda[i]);  if(i == 1){  plot(x, densitate, type= "p", col = culori[i], xlim = c(lambda[1], lambda[5]));  obs = rpois(N, lambda[i]);  #calculam media si variatia (vezi simulare 2)  media = mean(obs);  variatia = var(obs);  } else {  lines(x, densitate, type = "p", col = culori[i]);  }  }    legend("topleft", legend = lambda, col = culori, pch = 16, cex = 1);      for(i in 1:5) {  x = seq(1, 2 \* lambda[i], 1);  probabilitate = ppois(x, lambda[i]);  if(i == 1){  plot(x, probabilitate, type="p", col = culori[i], xlim = c(lambda[1], lambda[5]));  } else {  lines(x, probabilitate, type="p", col = culori[i]);  }  }  legend("topleft", legend = lambda, col = culori, pch = 16, cex = 1);  }  Exp = function() {  #pt 5 valori diferite ale lui lambda  lambda = c(0.1, 0.5, 1, 1.5, 2);    for(i in 1:5) {  obs = rexp(N, lambda[i]);  densitate = dexp(obs, lambda[i]);  if(i == 1){  plot(obs, densitate, type= "p", col = culori[i]);  #calculam media si variatia (vezi simulare 3)  media = mean(obs);  variatia = var(obs);  } else {  lines(obs, densitate, type = "p", col = culori[i]);  }  }  legend("bottomright", legend = lambda, col = culori, pch = 16, cex = 1);      for(i in 1:5) {  x = seq(1, N, lambda[i]);  probabilitate = pexp(x, lambda[i]); | |
| if(i == 1){  plot(x, probabilitate, type="p", col = culori[i]);  } else {  lines(x, probabilitate, type="p", col = culori[i]);  }  }  legend("bottomright", legend = lambda, col = culori, pch = 16, cex = 1);  }  Norm = function() {  #pt 5 valori diferite ale mean si sd  mean = c(0.1, 0.5, 1, 1.5, 2);  sd = c(1, 1.5, 2, 3, 2.5);    for(i in 1:5) {  obs = rnorm(N, mean[i], sd[i]);  densitate = dnorm(obs, mean[i], sd[i]);  if(i == 1){  plot(obs, densitate, type= "p", col = culori[i]);  #calculam media si variatia (vezi simulare 4)  media = mean(obs);  variatia = var(obs);  } else {  lines(obs, densitate, type = "p", col = culori[i]);  }  }  legend("topleft",  legend = c("mean = 0.1, sd = 1", "mean = 0.5, sd = 1.5", "mean = 1, sd = 2", "mean = 1.5, sd = 3", "mean = 2, sd = 2.5"),  col = culori, pch = 16, cex = 1);      for(i in 1:5) {  x = rnorm(N, mean[i], sd[i]);  probabilitate = pnorm(x, mean[i], sd[i]);  if(i == 1){  plot(x, probabilitate, type="p", col = culori[i]);  } else {  lines(x, probabilitate, type="p", col = culori[i]);  }  }  legend("topleft",  legend = c("mean = 0.1, sd = 1", "mean = 0.5, sd = 1.5", "mean = 1, sd = 2", "mean = 1.5, sd = 3", "mean = 2, sd = 2.5"),  col = culori, pch = 16, cex = 1);  }  Bin() #(vezi Figura 1.1 si 1.2)  Pois() #(vezi Figura 2.1 si 2.2)  Exp() #(vezi Figura 3.1 si 3.2)  Norm() #(vezi Figura 4.1 si 4.2) | |

**Binomiala**

|  |
| --- |
| C:\Users\Alina\Desktop\fig1-1.PNG  Figura 1.1 |

|  |
| --- |
| C:\Users\Alina\Desktop\fig1-2.PNG  Figura 1.2 |

|  |
| --- |
| Simulare 1 |
| C:\Users\Alina\Desktop\fig2-1.PNG  Figura 2.1 | |

**Poisson**

|  |
| --- |
| C:\Users\Alina\Desktop\fig2-2.PNG  Figura 2.2 |

|  |
| --- |
| Simulare 2 |

**Exponentiala**

|  |
| --- |
| C:\Users\Alina\Desktop\fig3-1.PNG  Figura 3.1 |

|  |
| --- |
| C:\Users\Alina\Desktop\fig3-2.PNG  Figura 3.2 |

|  |
| --- |
| Simulare 3 |

**Normala**

|  |
| --- |
| C:\Users\Alina\Desktop\fig4-1.PNG  Figura 4.1 |

|  |
| --- |
| C:\Users\Alina\Desktop\fig4-2.PNG  Figura 4.2 |

|  |
| --- |
| Simulare 4 |
| **4.**  n = c(25, 50, 100) p = c(0.05, 0.1) k = 1:10  lambda = 1 pois\_int = function(x) {  suma = (exp(-lambda)) \*((lambda ^x) /factorial(x))  return(suma) } pois = function(k) {  return(sum(pois\_int(0:k))) }   norm\_int = function(k, n, p) {  return((k -n \*p) / sqrt(n \*p \*(1-p))) } norm = pbinom(q = norm\_int(k, n, p), n, p)   norm\_c\_int = function(k, n, p) {  return((k + 0.5 - n \* p) / sqrt((n \* p) \* (1 - p))) } norm\_c = pbinom(q = norm\_c\_int(k, n, p), n, p)   camp\_int = function(k, n, p) {   a = 1 / (9 \* (n - k))   b = 1 / (9 \* (k + 1))   r = abs((k + 1) \* (1 - p)) / abs(p \* (n - k))   c = ((1 - b) \* r ^ (1 / 3))  niu = 1 - a  delt = sqrt(a + b \* r ^ (2 / 3))  return(((c - niu) / delt)) } camp = pbinom(q = camp\_int(k, n, p), n, p)  tabel\_p =function(k, n, p) {  norm = pbinom(q = norm\_int(k, n, p), n, p)  norm\_c = pbinom(q =norm\_c\_int(k, n, p), n, p)  camp =pbinom(q =camp\_int(k, n, p), n, p)  bin = c(pbinom(q =k, size =n, prob =p))  pois=k;  for(i in k){  pois[i]=pois(i)  }  tabel = data.frame(k, bin, pois, norm, norm\_c, camp)  colnames(tabel)=c("k", "Binomiala", "Poisson", "Normala", "Normala Corectie", "Camp-Paulson")  rm(k, bin, pois, norm, norm\_c, camp)  return(tabel)  } tabel\_d = function(k, n, p) {  norm = pbinom(q = norm\_int(k, n, p), n, p)  norm\_c = pbinom(q =norm\_c\_int(k, n, p), n, p)  camp =pbinom(q =camp\_int(k, n, p), n, p)  bin = c(dbinom(x =k, size =n, prob =p))  pois=k;  for(i in k){  pois[i]=pois(i)  }  tabel = data.frame(k, bin, pois, norm, norm\_c, camp)  colnames(tabel)=c("k", "Binomiala", "Poisson", "Normala", "Normala Corectie", "Camp-Paulson")  rm(k, bin, pois, norm, norm\_c, camp)  return(tabel) } tabele = vector(mode = "list", length = 12) contor=1; for(i in 1:3) {  for(j in 1:2) {  lambda = n[i] \* p[j];  tabele[[contor]]=tabel\_p(k,n[i],p[j])  tabele[[contor+1]]=tabel\_d(k,n[i],p[j])  contor=contor+2  } } | |

**Tabelele pentru functia de repartitie**

**n =25 , p = 0.05**

|  |
| --- |
|  |

**n =25 , p = 0.1**

|  |
| --- |
|  |

**n =50 , p = 0.05**

|  |
| --- |
|  |

**n =50 , p=0.1**

|  |
| --- |
|  |

**n =100 , p = 0.05**

|  |
| --- |
|  |

**n =100 , p = 0.1**

|  |
| --- |
|  |

|  |
| --- |
| **5.**  n = c(25, 50, 100) p = 0.1 k = 1:10 lambda = n[1] \* p  #Aproximarea Poisson pois\_int = function(x) {  suma = (exp(-lambda)) \*((lambda ^x) /factorial(x))  return(suma) } pois = function(k) {  return(sum(pois\_int(0:k))) }  #Aproximarea Normală (rezultată din Teorema Limită Centrală norm\_int = function(k, n, p) {  return((k -n \*p) / sqrt(n \*p \*(1-p))) } norm = pbinom(q = norm\_int(k, n, p), n, p)  #Aproximarea Normală cu factor de corecție norm\_c\_int = function(k, n, p) {  return((k + 0.5 - n \* p) / sqrt((n \* p) \* (1 - p))) } norm\_c = pbinom(q = norm\_c\_int(k, n, p), n, p)  #Aproximarea Camp-Paulson camp\_int = function(k, n, p) {   a = 1 / (9 \* (n - k))   b = 1 / (9 \* (k + 1))   r = abs((k + 1) \* (1 - p)) / abs(p \* (n - k))   c = ((1 - b) \* r ^ (1 / 3))  niu = 1 - a  delt = sqrt(a + b \* r ^ (2 / 3))  return(((c - niu) / delt)) } camp = pbinom(q = camp\_int(k, n, p), n, p)  eroarea = function(k, n, p) {  rez\_binom = pbinom(q = k, size = n, prob = p)    rez\_pois = k  for(i in k)   rez\_pois[i] = pois(i)    rez\_max = k  for(i in k) {   rez\_max[i] = max(rez\_binom[i] - rez\_pois[i],  rez\_binom[i] - norm[i],  rez\_binom[i] - norm\_c[i],  rez\_binom[i] - camp[i])  }  return(rez\_max) }  max1 = eroarea(k, n[1], p) max2 = eroarea(k, n[2], p) max3 = eroarea(k, n[3], p) plot(x = k, y = max1, type = "p", xlim = c(0, 10), ylim =c(0.01, 0.5), col ="red", pch =0, cex =1) points(x = k, y = max2, col = "brown", pch = 1) points(x = k, y = max3, col = "blue", pch = 2) |

|  |
| --- |
|  |

***Problema 2***

|  |
| --- |
| N = 1000 culori = c("blue","red") plot(1,1, xlim = c(-1,1), ylim = c(-1,1)); X = 1:1000 Y = 1:1000  for(i in 1:N){  X[i] = runif(n = 1, -1, 1);  Y[i] = runif(n = 1, -1, 1);    if((X[i] \* X[i] + Y[i] \* Y[i]) <= 1){  points(X[i], Y[i], type = "p", col = culori[1]);  } else{  points(X[i], Y[i], type = "p", col = culori[2]);  } }  2. |

|  |
| --- |
|  |