CA1: Advanced analytics

Anastasia Kasara

15 March 2018

Question 1

(1a)

The probability model corresponding for each agent is shown in table

- Agent 1: Poisson, defined with one parameter (average number of customers λ):

$$P(\lambda, t) = \frac{\lambda t}{k!} e^{-\lambda t}$$

- Agent 2: Exponential, defined with one parameter (average service rate r)

$$E(\lambda, t) = \lambda e^{-\lambda t}$$

- Agent 3: Binomial, defined with 3 parametes: n the sample size, p the probability of succes and k the number of outcomes.

$$B(j, n, p) = \frac{n!}{j!(n-j)!} p^j (1-p)^{n-j}$$

(1b)

The probability, that the third agent will report more than 2 customers purchased the product, is calculated as below. Since in the present case the probability model is binomial, with $p = 0.4$ to buy the product and n the sample size (number of customers in the sell zone), then to have 3 or more successes we need to sum up all the probabilities for $j = 3, 4, .., 20$:

$$P(j > 2, n, p) = \sum_{j=3}^{n} p_b(j, n, p) = 1 - \sum_{j=0}^{2} p_b(j, n, p)$$

where $p_b(j, n, q)$ represents the binomial probability. For the summations the 'dbinom' function is utilized:

```
# (question 1b)
p=.4          # probability for product purchase
n=20          # number of customers
# probability for more than 2 reports
p2 = 1-pbinom(2,n,p)    #
```

(1c)

In the below 100 samples are generated from each of the above models (Poisson, Exponential, Binomial). We need to specify the corresponding parameters each time. So we set,

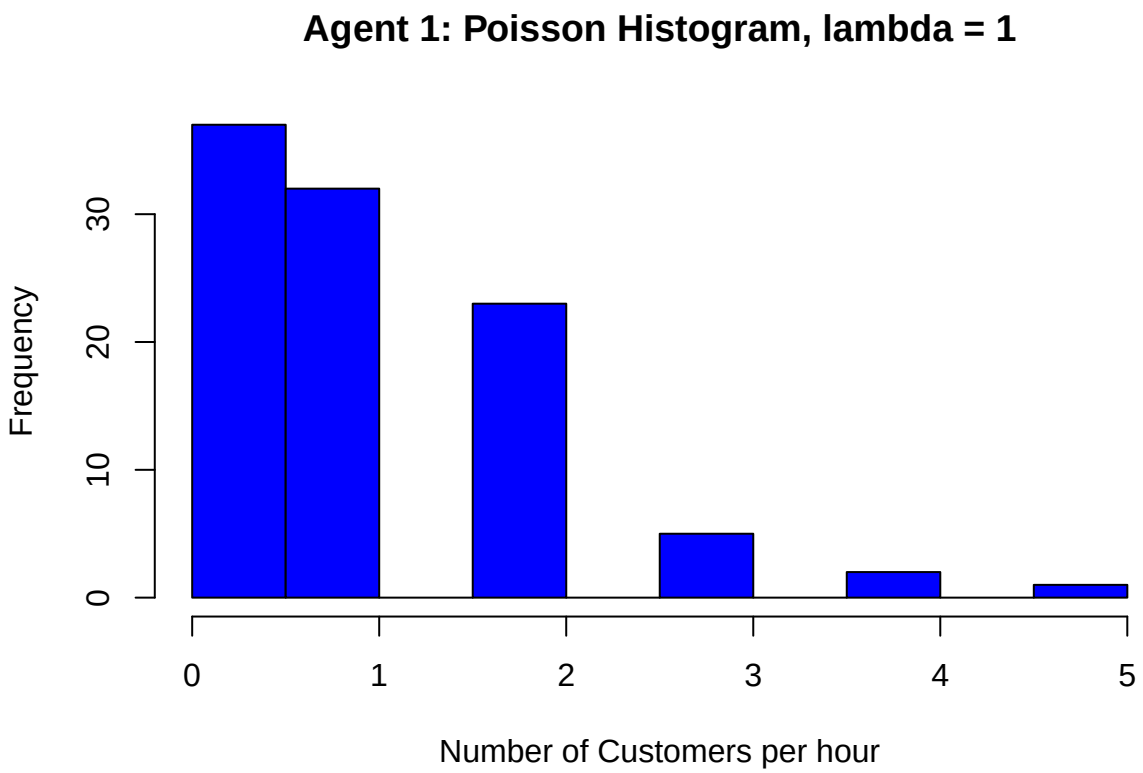
$$\lambda = 1\ (Poisson), \qquad r = 1\ (Exponential), \qquad n = 30,\ p = 0.4\ (Binomial)$$

```
s.poisson <- rpois(n=100, lambda = 1)  # 100 samples of a1 (Poisson)
s.exp     <-  rexp(n=100, rate = 1)  # 100 samples of a2 (exponential)
# 100 samples of a3 (binom), number of customer = 30
s.binom   <-rbinom(n=100, size=30, p = 0.4)
```

(1d)

– agent 1: Poisson, histogram plot and summary

```
hist(s.poisson, xlab="Number of Customers per hour", col = "blue", main="Agent 1: Poisson Histogram, la
```




```
summ.poisson <- summary(s.poisson)  # summary
var.poisson  <- var(s.poisson)      # variance
print(summ.poisson)
```

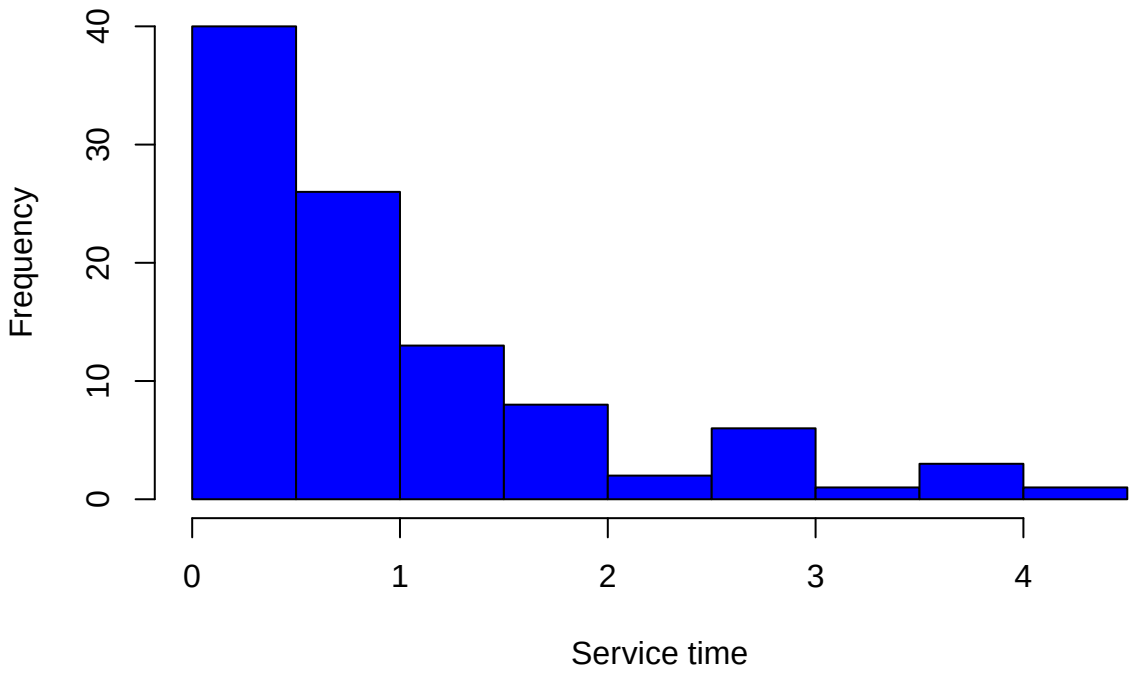
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    1.00    1.06    2.00    5.00
```

```
print(var.poisson)
```

```
## [1] 1.147879
```

– Agent 2: Exponential, histogram plot and summary

```
# agent 2: exponential, histogram, summary and var
hist(s.exp,   xlab="Service time", col = "blue",  main="Agent 2: Exponential Histogram, lambda = 1")
```




```
summ.exp <- summary(s.exp)          # summary
var.exp  <- var(s.exp)              # variance
print(summ.exp)
```

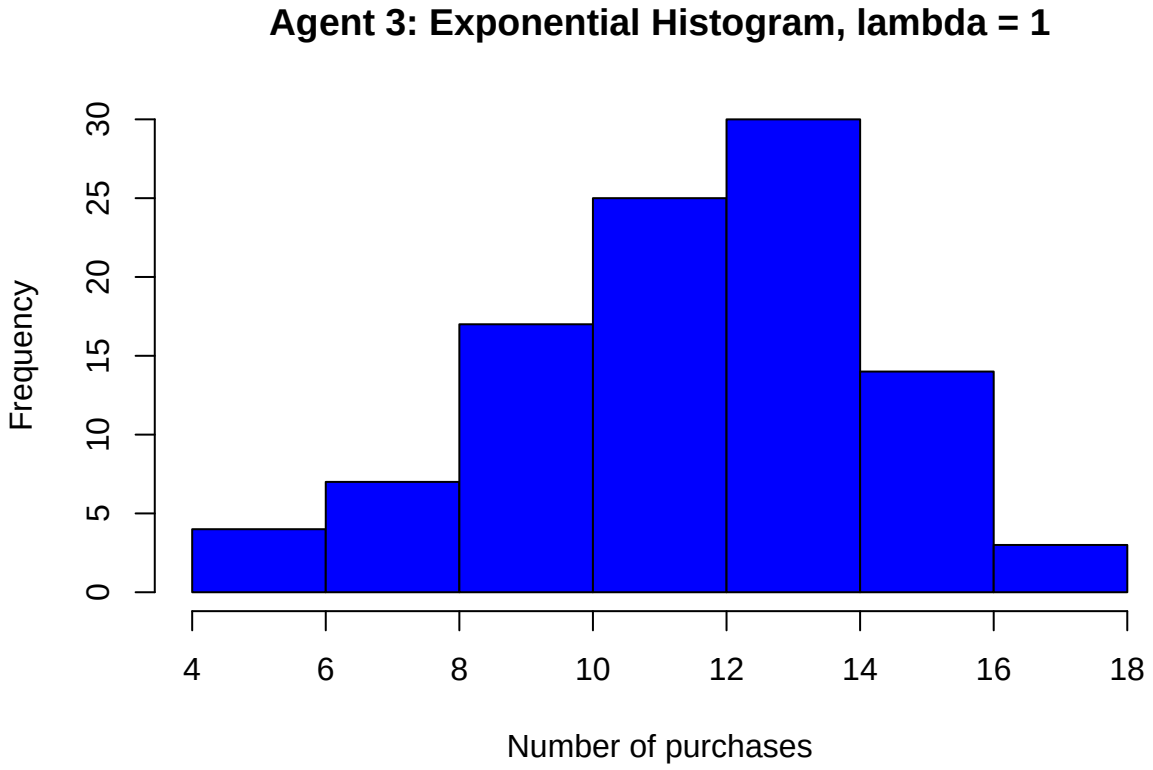
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01517 0.32613 0.64041 0.97543 1.25622 4.32425
```

```
print(var.exp)
```

```
## [1] 0.9337985
```

– Agent 3: Binomial, histogram plot and summary

```
# agent 3: binom, histogram and summary
hist(s.binom,  xlab="Number of purchases", col = "blue",  main="Agent 3: Exponential Histogram, lambda =
```




```
summ.binom <- summary(s.binom)      # summary
print(summ.binom)                   # variance
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       4      10      12      12      14      18
```

Question 2

(2a)

Assuming that the sensor's security probability model is the Beta function, $B(a,b) = B(4,8)$ the the expectation value, (E) and the variance, (V) are given by,

$$E = \frac{a}{a+b} = \frac{4}{8+4} = 1/3, \qquad V = \frac{ab}{(a+b)^2(a+b+1)} = ... = \frac{2}{117}$$

Next I take as probability failure for each individual sensor the expectation value, $p = 1/3$. For the given 4 sensors, complete failure exists only when the 3 or 4 sensors fail. So the probability for $j > 2$ failures should be calculated. The probability for each sensor is binomial, therefore,

$$p(j > 2) = \sum_{j=3,4} p_b(j, 4, p) = 1 - \sum_{j=0,1,2} p_b(j, 4, p), \qquad p = 1/3.$$

The last summation is performed by using 'pbinom' from the R. Below is the corresponding code.

```
##############################
#
# Beta distribution with a = 4, b=8, Beta(4,8)
```

```r
# question (2a)
a<-4
b<-8
E<-a/(a+b)                          #  E = 1/3
varhat<-a*b/((a+b)^2*(a+b+1))       #  V = 2/117
print(E)
```

```
## [1] 0.3333333
```

```r
print(varhat)
```

```
## [1] 0.01709402
```

```r
#####
# from the n = 4 sensors at least k = 2 are secured
# probability for each individual sensor p = E
n<-4                    # number of sensors
p<-1/3                  # expectation of beta(4,8)
y<-1-pbinom(2,n,p)      # probability that at ;east 2 are secured
print(y)
```

```
## [1] 0.1111111
```

(2(b-c))

In the below 25 samples for each sensor are generated using the 'rbinom' function. After calculating the mean value of each of the random samples the preferred sensor in terms of security is the one with smaller variance. The behaviour of this sensor is more 'predictable' than the others as it will vary less around it's mean value probability (being chosen to be $p = 1/3$). The R code where the random samples are generated are shown below.

```r
# question (2b)
s <- 1
n <- 25
p<- 1/3


b1 <-rbinom(n,s,p)      # sensor 1
b2 <-rbinom(n,s,p)      # sensor 2
b3 <-rbinom(n,s,p)      # sensor 3
b4 <-rbinom(n,s,p)      # sensor 4

mn1 <- mean(b1)
vr1 <-  var(b1)
print(mn1)
```

```
## [1] 0.16
```

```r
print(vr1)
```

```
## [1] 0.14
```

```r
#
mn2 <- mean(b2)
vr2 <-  var(b2)
print(mn1)
```

```
## [1] 0.16
```

```
print(vr1)
```

```
## [1] 0.14
```

```
#
mn3 <- mean(b3)
vr3 <-  var(b3)
print(mn3)
```

```
## [1] 0.28
```

```
print(vr3)
```

```
## [1] 0.21
```

```
#
mn4 <- mean(b4)
vr4 <-  var(b4)
print(mn4)
```

```
## [1] 0.32
```

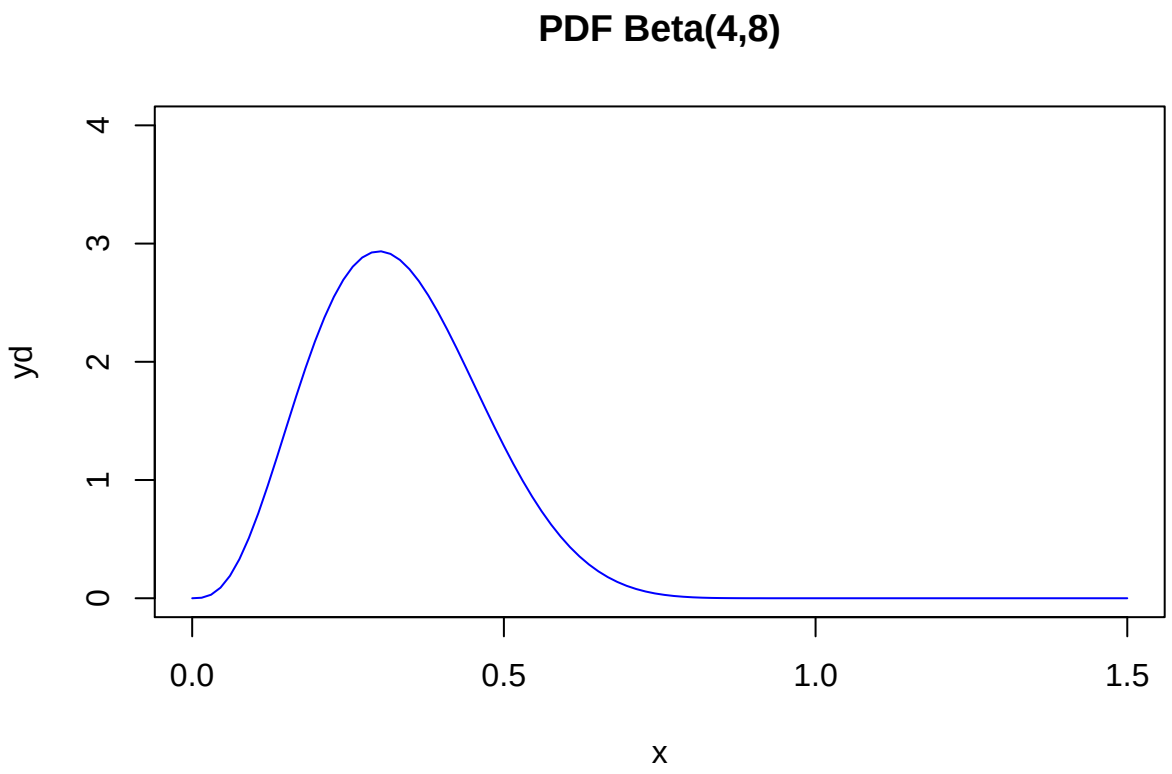
```
print(vr4)
```

```
## [1] 0.2266667
```

```
#



# (d)
par(mfrow=c(1,2))


plot
```

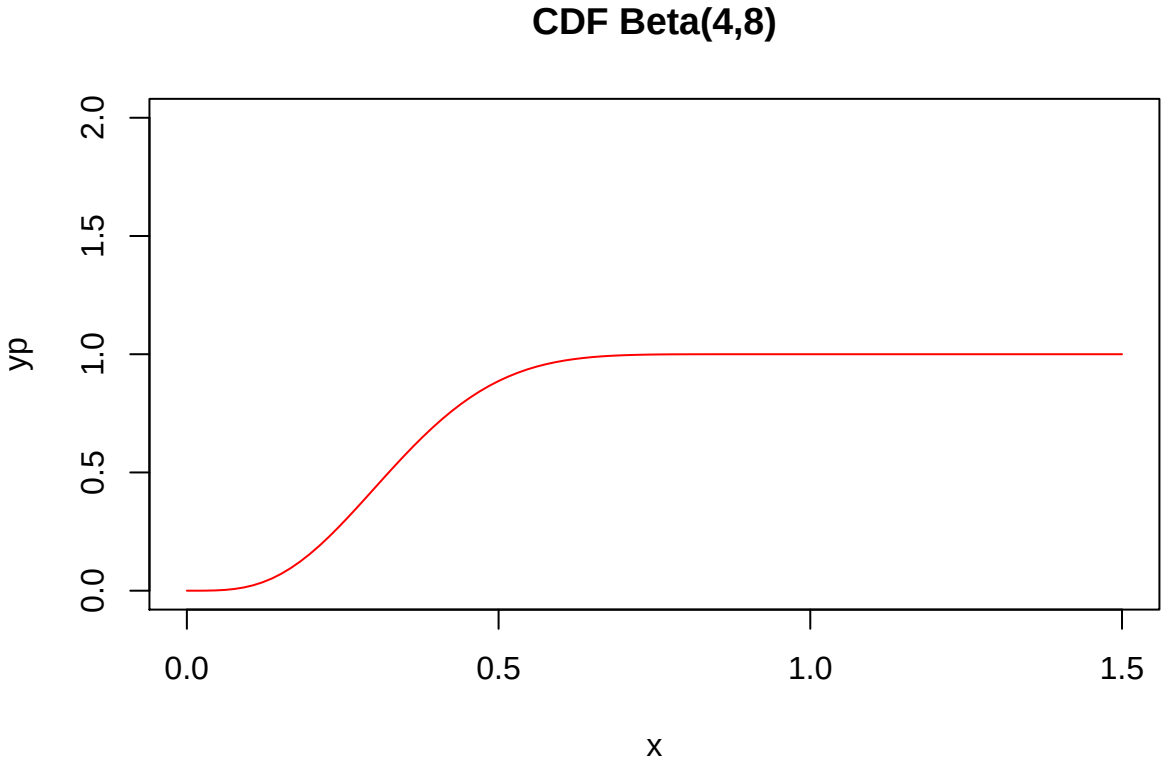
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x3bc1888>
## <environment: namespace:graphics>
```

```
x=seq(0,1.5,length=100)
yd=dbeta(x,4,8)
plot(x,yd, type="l", col="blue",xlab="x", ylim=c(0,4), main="PDF Beta(4,8)" )
```



Below is the plot for the CDF of the beta distribution:

```
x=seq(0,1.5,length=100)
yp=pbeta(x,4,8)
plot(x,yp, type="l", col="red",xlab="x", ylim=c(0,2), main="CDF Beta(4,8)" )
```



Question 3

Here we need to use the Normal distribution The below is known as the density of the normal distribution, $N(\mu, \sigma)$. :

$$f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

The probability is given by integrating the above probability density from $x = 500$ to infinity:

$$p(x > 500) = \int_{500}^{\infty} dx f(x)$$

This can be achieved by recalling the 'pnorm' R function as below:

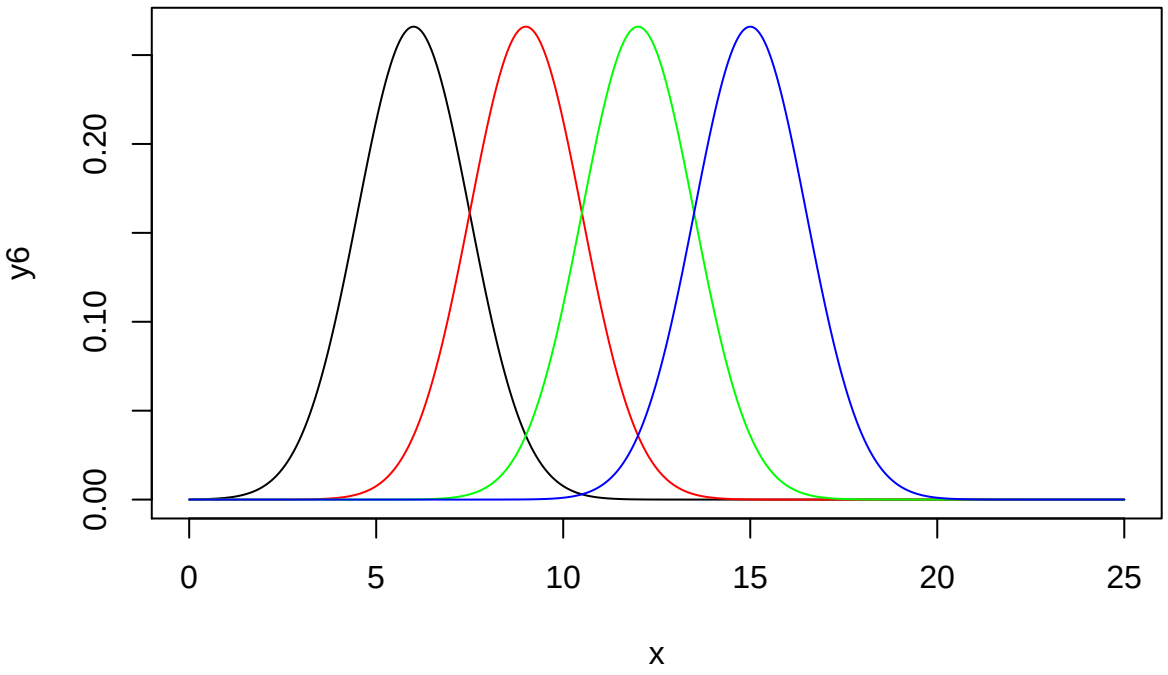
```
# Normal distribution N(527,112)  mean = 527, std = 112
mu<-527
sigma<-112
p1<-1-pnorm(500,mu,sigma)    # probability to score above 500
print(p1)
```

```
## [1] 0.5952501
```

(3a)

Next we generate 4 plots with different means but same standard variance. The chosen means are $\mu = 6, 9, 12$ and 15. As standard deviation the value $\sigma = 1.5$ was chosen.

```
# question (3a) plot

x   <- seq(0,25,length=2000)
y6  <- dnorm(x,mean=6,  sd=1.5)
y9  <- dnorm(x,mean=9,  sd=1.5)
y12 <- dnorm(x,mean=12, sd=1.5)
y15 <- dnorm(x,mean=15, sd=1.5)

plot( x,y6,type="l", xlab="x", col="black")
lines(x,y9,type="l", xlab="x",col="red"  )
lines(x,y12,type="l", xlab="x",col="green")
lines(x,y15,type="l", xlab="x",col="blue" )
```



(3b)

The probability for an individual scoring between $x_1 = 400$ and $x_2 = 500$ is given by,

$$p(400 < x < 500) = \int_{400}^{500} dx f(x) = \int_{0}^{500} dx f(x) - \int_{0}^{400} dx f(x)$$

Similarly, for the above the 'pnorm' function can be used as below.

```
# question (3b)
# probability for GMAT score between 400 and 500
p2 = pnorm(500,mu,sigma) - pnorm(400,mu,sigma)
p2
```

```
## [1] 0.2763376
```

Question 4

The proper probability model for the present case is the multinomial probability for 4 different outcomes of equal probability, so that $p_1 = p_2 = p_3 = p_4 = 1/4$. The expression for the multinomial probability is.

$$p_m(x_1, x_2, x_3, x_4, p_1, p_2, p_3, p_4) = \frac{n!}{x_1! x_2! x_3! x_4!} p_1^{x_1} p_2^{x_2} p_3^{x_3} p_4^{x_4}, \qquad \sum_{i=1,4} x_i = n, \qquad \sum_{i=1,4} p_i = 1$$

(4a)

From a sample of size $n = 10$ we require $x_1 = 9$ successes for the first feature (with probability $p_1 = 1/4$) and anything else. So, the other 3 features have probability $1 - p_1 = 3/4$. We can then use the binomial probability with $n = 10$, $k = 9$ and $p_1 = 1/4$, to obtain:

$$p(x_1 = 9, 10, 1/4) = \frac{10!}{9!1!}(\frac{1}{4})^9(\frac{3}{4})^1 = 3 \times 10(\frac{1}{4})^{10} = 2.861023 \times 10^{-5} << 1$$

This is a very small probability. The relevant code is below:

```
p1<- 1/4
n <- 10
xi<- 9
p1<-dbinom(xi,n,p1)
print(p1)
```

```
## [1] 2.861023e-05
```

(4b)

The correlation betweem the features '1' and '2' may be found from the following relation: The correlation between two random variables x_1, x_2 is defined as,

$$\rho_{12} = \frac{C_{12}}{\sqrt{V_1}\sqrt{V_2}}$$

where V_1, V_1 are the corresponding *variances* and C_{12} is the *covariance*. But for the multinomial distribution:

$$C_{12} = -np_1p_2 = n, \qquad V_i = np_i(1-p_i), \qquad i = 1, 2$$

The above formulas are calculated as below ($p_1 = p_2 = p = 1/4$) and $n = 4$:

```
n<-4
p<-1/4
c12 <- -n*p*p
v1 <- n*p*(1-p)
v2 <- n*p*(1-p)
r12 <-c12/(sqrt(v1*v2))
print(r12)
```

```
## [1] -0.3333333
```

Simulation

Simulating 100 samples to find numerically the expected correlation. For this the use of 'rmultinom' provides the sample random vectors with probabilities distributed according the $p_1 = p_2 = p_3 = p_4 = 1/4$. The size of the sample is $s = 10$ and $n = 100$. After we have generated the samples a matrix 4×100 is created. The rows of length (4) contains the distributions over the 4 features while the columns of length (n) are the different samples. As it is more natural to have the samples as rows and the features as columns, I take the transpose of the generated matrix. To calculate the correlation between the features $'1'$ and $'2'$ the corresponding probabilities are needed 'p1' and 'p2'. For this I first calculate the frequencies for each row as:

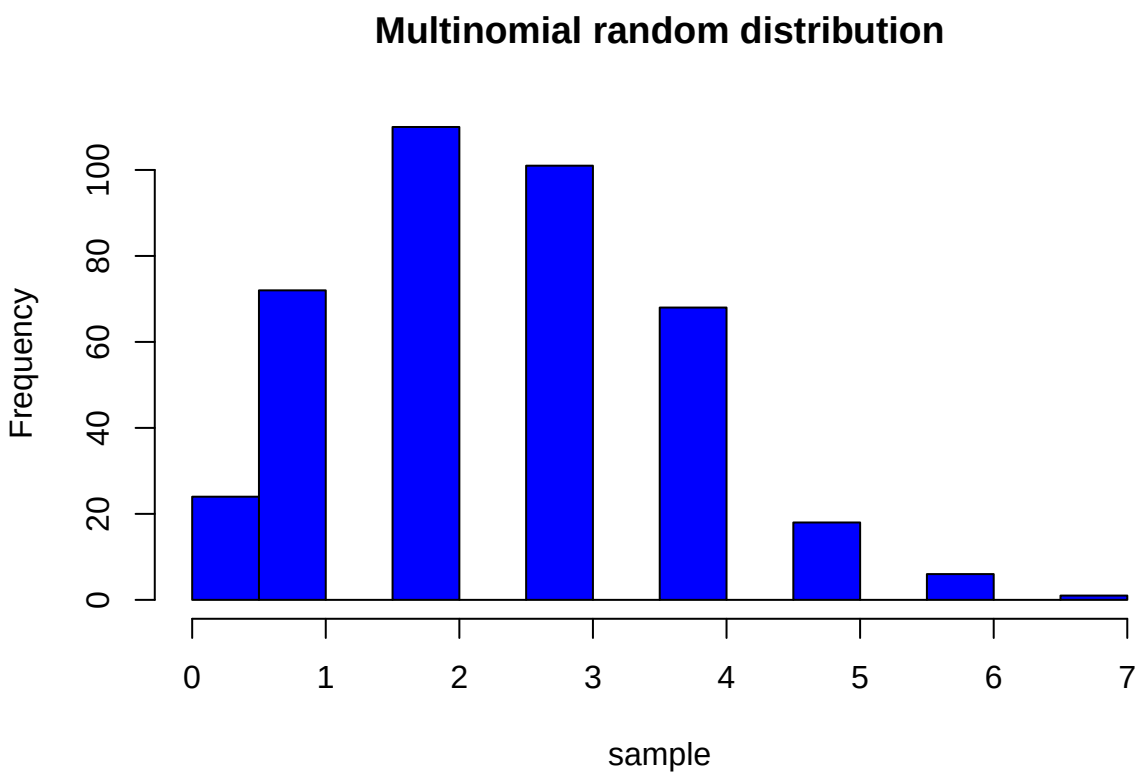
$$f_{1j} = \frac{r_{1j}}{s}, \quad f_{2j} = \frac{r_{2j}}{s}, \qquad j = 1, 2, .., n = 100, \qquad s = 10$$

and then average over the samples to find the mean probabilities:

$$p_1 = \frac{\sum_{j=1}^{n} f_{1j}}{n}, \qquad p_2 = \frac{\sum_{j=1}^{n} f_{2j}}{n}, \qquad n = 100$$

```
# simulation for 100 samples.

pm = c(1/4,1/4,1/4,1/4)
size = 10
```

```r
samples =100
# assume the number of rows in each feature to be n=20
s.multi = rmultinom(samples,size,pm)
hist(s.multi, xlab="sample", col = "blue", main="Multinomial random distribution"    ) # histogram
```




```r
# set as rows the samples (100) and as column the features (4)
ts.multi<-t(s.multi) # transpose
#
table(ts.multi)
```

```
## ts.multi
##   0   1   2   3   4   5   6   7
##  24  72 110 101  68  18   6   1
```

```r
#View(ts.multi)
#print(t)

# First we find the frequencies for each sample (sample =100):
f1 <- ts.multi[,1]/size   # size of f1 should be 100
f2 <- ts.multi[,2]/size   # size of f2 should be 100
# then we take the average over the samples (100)
w1 <- mean(f1)
w2 <- mean(f2)


#pp<-c(0.1,0.6,0.3)
#n<-10
#k<-200
#print(rmultinom(10,k,pp))
```

Having calculated the probabilities the calculation of the correlation of ρ_{12} follows, according the expressions above in the text.

```
c12 <- -n*w1*w2           #  1-2

v11  <- n*w1*(1-w1)       # variance of 1
v22  <- n*w2*(1-w2)       # variance of 2
r12 <- c12/sqrt(v11*v22)  # correlation of 1-2
print(r12)
```

```
## [1] -0.3771686
```

Note that the model calculated value for the correlation coefficient was $\rho_{12} = -0.333$ to be compared with the numerical above.