Math 3070 § 1. **Joint Probability Mass Function** Name: Example
Treibergs May 28, 2011

R Session:

```
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[R.app GUI 1.31 (5538) powerpc-apple-darwin8.11.1]

[Workspace restored from /Users/andrejstreibergs/.RData]

> #################### ENTER JOINT PMF #################################
> # This is the joint p.m.f. from Devore, problem  5.22
>
> p <- matrix(c(.02,.04,.01,.06,.15,.15,.02,.20,.14,.10,.10,.01),ncol=4)
> p
     [,1] [,2] [,3] [,4]
[1,] 0.02 0.06 0.02 0.10
[2,] 0.04 0.15 0.20 0.10
[3,] 0.01 0.15 0.14 0.01
> # is it pmf?
> sum(p)
[1] 1
> # Yes. All nonnegative entries and sum is 1.
> # Enter corresponding  X  and  Y  values.
> x<- c(0,5,10); y<- c(0,5,10,15)
```

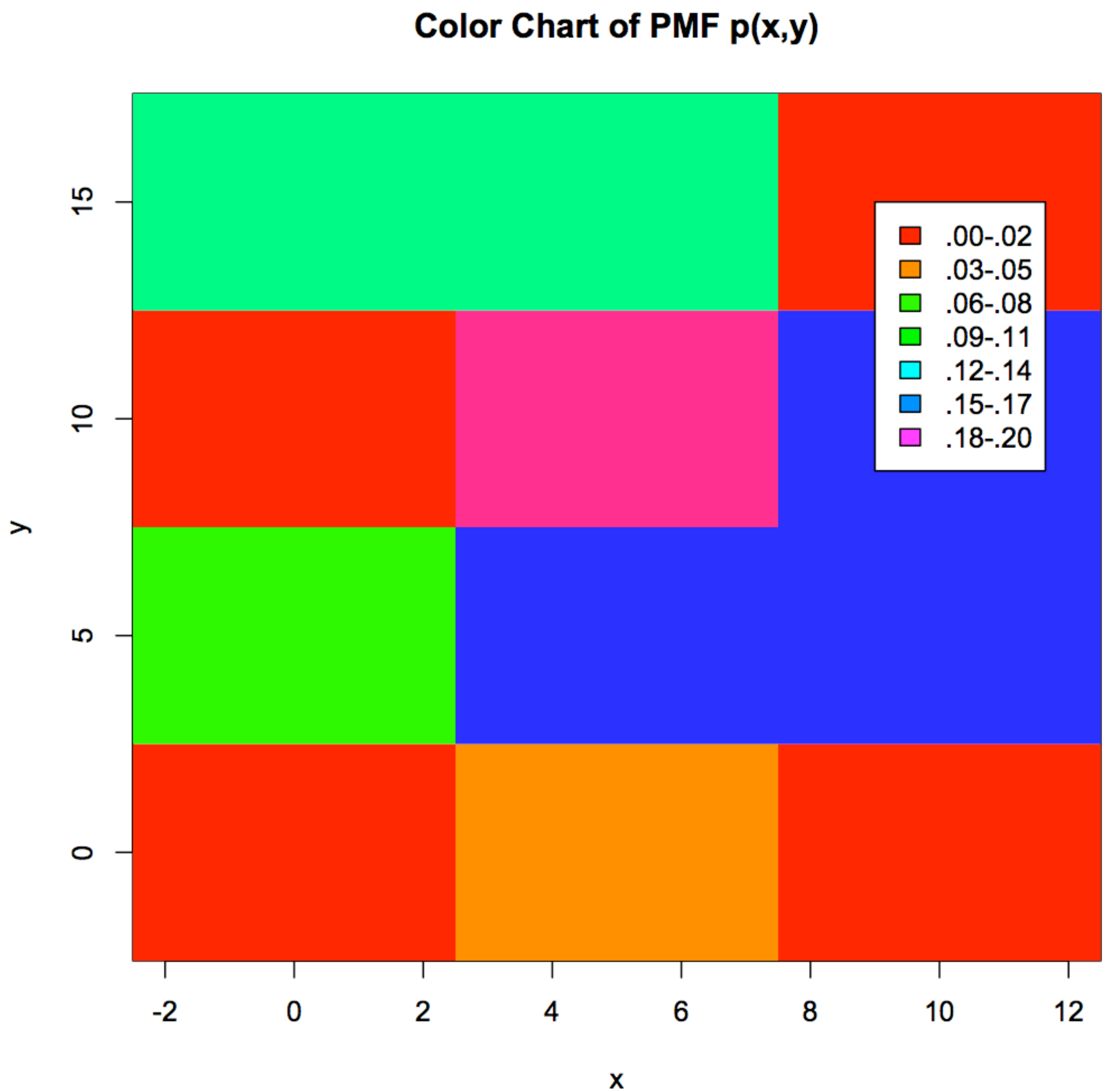
```
> ###################### MARGINAL PROBABILITIES #######################
>
> # Automatically adds rows and columns and augments the matrix with sums.
>
> addmargins(p)
     [,1] [,2] [,3] [,4] [,5]
[1,] 0.02 0.06 0.02 0.10 0.20
[2,] 0.04 0.15 0.20 0.10 0.49
[3,] 0.01 0.15 0.14 0.01 0.31
[4,] 0.07 0.36 0.36 0.21 1.00

> # The marginal probabilities can be picked off as row and column sums.
> px <- margin.table(p,1);px
[1] 0.20 0.49 0.31
> py <- margin.table(p,2);py
[1] 0.07 0.36 0.36 0.21
>
> ################ EXPECTED VALUES, VARIANCE OF X, Y ################
>
> # Remember, R does vector arithmetic COMPONENTWISE.
> # The entries of px*x are px[i]*x[i]. Expectation is sum of these
> EX <- sum(px*x); EX
[1] 5.55
> EY <- sum(py*y); EY
[1] 8.55
> VX <- sum(px*x^2) - EX^2; VX
[1] 12.4475
> VY <- sum(py*y^2)-EY^2;VY
[1] 19.1475
>
> # pick off the dimensions
> dim(p)
[1] 3 4
> nx <- dim(p)[1]; ny <- dim(p)[2]; c(nx,ny)
[1] 3 4
>
> # E(XY) is sum over all (i,j) of  p[i,j]*x[i]*y[j]
> # Run a double loop. Accumulate the sum as you go.
> A <-0
> for(i in 1:nx)
+               {
+                for(j in 1:ny)
+                              {
+                               A <- A+p[i,j]*x[i]*y[j]
+                              }
+               }
> EXY <- A; EXY
[1] 44.25
```

```
> # Qestion 5.30 wants rho and cov of 5.22.
> CovXY <- EXY - EX*EY; CovXY
        [,1]
[1,] -3.2025
>
> ###################### CORRELATION ######################################
> rhoXY <- CovXY/(sqrt(VX)*sqrt(VY)); rhoXY
           [,1]
[1,] -0.2074398

> # Alternately, we may use matrix multiplication for EXY.  t() is transpose
>
> t(x)
     [,1] [,2] [,3]
[1,]    0    5   10
>
>  t(x) %*% p %*% y
      [,1]
[1,] 44.25


> ######################## PLOT COLOR IMAGE OF PMF #######################
> r12 <- rainbow(12)
> image(x,y,p,col=r12)
> legend(9, 15, legend=
+ c(".00-.02",".03-.05",".06-.08",".09-.11",".12-.14",".15-.17",".18-.20"),
+ fill=c(r12[1:2],r12[4:5],r12[7:8],r12[11]),bg="white")
> title("Color Chart of PMF p(x,y)")
> # M3074JointPMF1.pdf
```



Color Chart of PMF p(x,y)
.00-.02
.03-.05
.06-.08
.09-.11
.12-.14
.15-.17
.18-.20
y
15
10
5
0
-2
0
2
4
6
8
10
12
x

```
> ###################### EXPECTATION OF FUNCTION f(x,y) ##################
> f <- function(x,y){x+y}
> A <-0
> for(i in 1:nx)
+               {
+                for(j in 1:ny)
+                              {
+                               A <- A+p[i,j]*f(x[i],y[j])
+                              }
+                }
> Ef <- A; Ef
[1] 14.1
> # Of course, the sum function is linear so E(X+Y) equals
> EX+EY
[1] 14.1
>
> # Here is a vector way to do the same. outer(x,y,"f")
> # gives the matrix whose [i,j]  component is f(x[i],y[j]).
> outer(x,y,"f")
     [,1] [,2] [,3] [,4]
[1,]    0    5   10   15
[2,]    5   10   15   20
[3,]   10   15   20   25
>
> # Same as
> outer(x,y,"+")
     [,1] [,2] [,3] [,4]
[1,]    0    5   10   15
[2,]    5   10   15   20
[3,]   10   15   20   25
>
> outer(x,y,"f")*p
     [,1] [,2] [,3] [,4]
[1,]  0.0 0.30  0.2 1.50
[2,]  0.2 1.50  3.0 2.00
[3,]  0.1 2.25  2.8 0.25
>
> sum(outer(x,y,"f")*p)
[1] 14.1
>
> # Part b. asks to find the expectation of max(X,Y)
> A <-0
> for(i in 1:nx)
+               {
+                for(j in 1:ny)
+                              {
+                               A <- A+p[i,j]*max(x[i],y[j])
+                              }
+               }
> EmaxXY <- A; EmaxXY
[1] 9.6
```

```
> # Alternatively, matrix whose entries are x[i]*y[j] is outer product.
> x %o% y
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0   25   50   75
[3,]    0   50  100  150
> # Another way to get E(XY)
> sum( (x %o% y) * p)
[1] 44.25
>
> ##################### INDEPENDENCE OF  X  AND  Y ###########################
> # Are the variables X and Y independent? Check if all  p[i,j]=px[i]*py[j]
> px %o% py
       [,1]   [,2]   [,3]   [,4]
[1,] 0.0140 0.0720 0.0720 0.0420
[2,] 0.0343 0.1764 0.1764 0.1029
[3,] 0.0217 0.1116 0.1116 0.0651
>
> p == px %o% py
      [,1]  [,2]  [,3]  [,4]
[1,] FALSE FALSE FALSE FALSE
[2,] FALSE FALSE FALSE FALSE
[3,] FALSE FALSE FALSE FALSE
> # None of the entries are equal.
> # (This is BAD PROGRAMMING!, roundoff may cause inequlity!)
>
> ############### CONDITIONAL PROBABILITY, CONDITIONAL EXPECTATION ###########
> # cond prob p_Y|X(y|x) = p(x,y)/px(x)
> px
[1] 0.20 0.49 0.31
> p
     [,1] [,2] [,3] [,4]
[1,] 0.02 0.06 0.02 0.10
[2,] 0.04 0.15 0.20 0.10
[3,] 0.01 0.15 0.14 0.01

> # We write three vectors cpgxi[j] = p_Y|X(y[j]|x[i]). Check sum is one.
>  cpgx1<-p[1,1:4]/px[1]; cpgx1; sum(cpgx1)
[1] 0.1 0.3 0.1 0.5
[1] 1
>  cpgx2<-p[2,1:4]/px[2]; cpgx2; sum(cpgx2)
[1] 0.08163265 0.30612245 0.40816327 0.20408163
[1] 1
>  cpgx3<-p[3,1:4]/px[3]; cpgx3; sum(cpgx3)
[1] 0.03225806 0.48387097 0.45161290 0.03225806
[1] 1
> # find E(Y^3 | X=x[2])
> sum(cpgx2*y^3)
[1] 1135.204
```

```
> ######################## THE INDUCED pmf FOR A FUNCTION ####################
>
> # find the pmf for the function XY. Find the values xy takes.
> txy <- outer(x,y,"*"); txy
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0   25   50   75
[3,]    0   50  100  150
> sort(c(txy))
 [1]   0   0   0   0   0   0  25  50  50  75 100 150
> vxy <- c(0,25,50,75,100,150)
> m <- length(vxy); m
[1] 6
>
> pxy <- p
> for(i in 1:6 )pxy[i]<- sum(p[txy==v[i]])
Error: object 'v' not found
> for(i in 1:6 )pxy[i]<- sum(p[txy==vxy[i]])
>
> # We use conditional indices. If txy[i,j]=vxy[k], include p[i,j] in sum.
> pxy <- 1:6
> for(k in 1:6 )pxy[k]<- sum(p[txy==vxy[k]])
> pxy
[1] 0.25 0.15 0.35 0.10 0.14 0.01
> sum(pxy)
[1] 1
> # example pxy[1]=sum of p[i,j] where x[i]*y[j]=vxy[1]=0
> pxy[1]
[1] 0.25
> sum(c(p[1,1:4],p[2:3,1]))
[1] 0.25
>
> Tn the old variables,
> EXY
      [,1]
[1,] 44.25
> In the new variable, Z=XY, E(Z)= sum pxy[k]*vxy[k]
> sum(pxy*vxy)
[1] 44.25
>
```