Advanced Mathematical Methods

Probability and Statistics - Exercises 2

1. Consider the following continuous pdf:

$f(x) = 3x^2, 0 \le x \le 1$

(a) Verify that this is a pdf.

(b) Find E(X) and Var(X).

(c) Find the probability P(X >1).

(d) Find E(1/X).

(e) Give an interpretation to the meaning of E(1/X).

The following questions require you to use MATLAB

2. Start up MATLAB and select the Statistics Toolbox and 'Help'. To see the Statistics Toolbox you may need to select the tab 'Launch Pad'. Under 'Help' you should see two categories of functions:

Probability density functions (pdf).

Cumulative Distribution functions (cdf).

You can use MATLAB to sketch the shape of many pdfs. Here are some examples:-

Y = normpdf(X,MU,SIGMA) Returns the normal pdf with mean, MU, and standard deviation, SIGMA, at the values in X.

The following MATLAB code will plot the N(0,1) distribution between -3 and +3 (for example):

```
>> x = -3:0.1:+3; %makes an array x from -3 to +3 in increments of 0.1
>> y = normpdf(x,0,1);
>> plot(x,y);
```

Plot each of the following pdfs:

(a) N(5,20)

(b) N(5,5)

(c) t-distribution (tpdf) with degrees of freedom 1,10,20,30,40.

(d) Chi-squared distribution (chi2pdf) with degrees of freedom 1,20,30,30,40

(e) F-distribution (fpdf) with degrees of freedom (1,1), (1,20), (20,1).

Note – you can also use the interactive demo called 'disttool' which shows you a number of distributions, and allows you interactively change the parameters.

Now use the cdfs to find the following probabilities:-

(a) for the normal distribution Z~N(0,1) show that

$P(-1.68 < Z < 1.68) = 0.90$
$P(-1.96 < Z < 1.96) = 0.95$
$P(-2.58 < Z < 2.58) = 0.99$

and hence show that for the general normal distribution $X \sim N(\mu,\sigma^2)$ that:

90% of the area is between $\mu \pm 1.68\sigma$
95% of the area is between $\mu \pm 1.96\sigma$
99% of the area is between $\mu \pm 2.58\sigma$

(b) For $X \sim \chi^2(10)$ (chi-squared with 10 d.f.) find $P(X > 20)$.

(c) For $F \sim F(5,10)$ find $P(F > 4)$.

3. A further category of MATLAB functions is called 'Critical Values of Distribution functions'. In general suppose that F(x) is a distribution function (i.e. $F(x) = P(X<x)$) and we require the solution of the equation $F(x) = \alpha$ where α is a given probability. For example, in the case of the normal distribution we want to know the value of x such that $F(x) = 0.025$. Then we can use the 'inverse cumulative distribution function' – for example:

```
>>norminv(0.025,0,1)
```

gives the answer -1.96.

Use these 'inverse distributions' to find each of the following:

(a) For $Z \sim N(0,1)$ find w such that $P(-w < Z < w) = 0.80$

(b) For $Z \sim N(0,1)$ find w such that $P(-w < Z < w) = 0.99$

(c) For $X \sim \chi^2(10)$ find w such that $P(X > w) = 0.95$

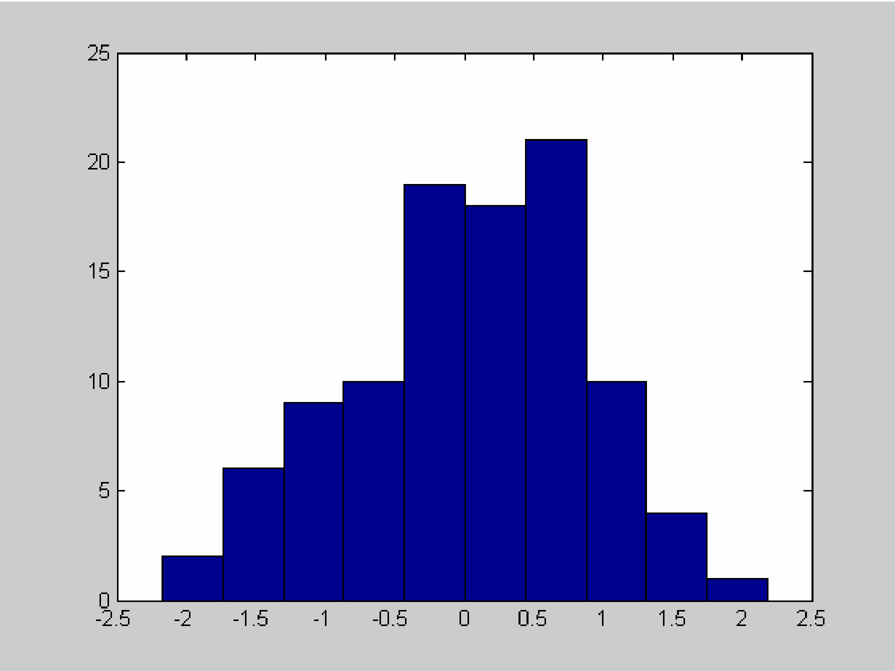
(d) For $F \sim F(5,10)$ find w such that $P(F > w) = 0.95$

4. This example shows how you can use MATLAB for simulations. There is another MATLAB set of functions called 'Random Number Generators'. These functions generate pseudo random observations on a variety of distributions.

Look at the following MATLAB code:-

```
>> m = zeros(1,100); %make an array of size 1*100 zeros to represent the means
>> sd = ones(1,100);   %make an array of size 1*100 zeros to represent the standard deviations

>> x = normrnd(m,sd); %generate 100 observations on N(0,1)
>> size(x)                              %check that we really did get 100
ans =
         1   100
>> hist(x)                              %plot the histogram – shown below
```



Now repeat the above for 500, 1000, … as high as you like, and check that the shape of the distribution does approximate closer and closer to the true N(0,1) as the sample size increases.

In help under 'Descriptive Statistics' you will find functions mean and std which calculate the mean and standard deviations of arrays. Find the mean and standard deviation of each of the simulated sets of observations you get.

5. We have seen that the Chi-squared distribution may be thought of as arising as the sum of squares of n N(0,1) random variables. We can see this at work through simulation.

```
>> x2all = []; %make an empty array
>> for i = 1:1000        %start of a for loop 1000 times
z = normrnd(zeros(1,20),ones(1,20));            %generate 20 obs on N(0,1)
x2 = sum(z.*z); %take the sum of squares
x2all = [x2all,x2];       %append to the end of the array
end; %end of for loop

hist(x2all) %plots the histogram of x2
```

This should look like the Chi-Squared distribution. Experiment with this for different sizes of the number of squared normal variables to be added, so that you are simulating Chi-squared with different degrees of freedom.

Simulate and plot the t-distribution. Recall that

$t = \frac{N(0,1)}{\sqrt{\frac{X^2}{n}}}$ has a t(n) distribution where X^2 is a Chi-Squared with n degrees of freedom.

Note that you can use 'chi2rnd' to generate observations on Chi-squared, the square root function is 'sqrt', and that element by element division is x./y.

(To understand the '.' do the following

x = 1:10; %x = [1,2,3,…,10];
y = 1:10; %y = [1,2,3,…,10];

Compare x./y with x/y).

Now do the same for the F-distribution, which is a ratio of two independent Chi-squares each divided by their degrees of freedom.

6. The Central Limit Theorem: For any distribution at all with mean μ and variance σ^2 we take n independent observations on this distribution and find the mean. The mean is of course itself a random variable. Then the Central Limit Theorem states that for large n (usually interpreted as n>30) the distribution of the mean will be $N(\mu,\sigma^2/n)$.

We can verify this through simulation. Let's work with the 'uniform distribution' on [0,1] as the base distribution (this itself is far from being like the Normal distribution).

unifrnd – can be used to generate random observations on the uniform distribution.

Consider the following MATLAB code:

```
>> allmeans = [];        %this makes an empty array
>> for i=1:100           %start of a for loop that will run 100 times
    x = unifrnd(zeros(1,30),ones(1,30));%generates 30 observations on uniform(0,1)
    allmeans = [allmeans,mean(x)];    %concatenates the mean of these 30 to the array

end;    %ends the for loop

>> hist(allmeans)        %plots the histogram
```

You will find that the histogram approximates the shape of the normal and if you find the mean and variance of 'allmeans' these should approximate what is predicted by the CLT.

You should repeat this for different sample sizes from the uniform(0,1) (i.e., try less than 30 and also much higher than 30).

You should also repeat this for different distributions – there are many to choose from in the MATLAB toolbox.