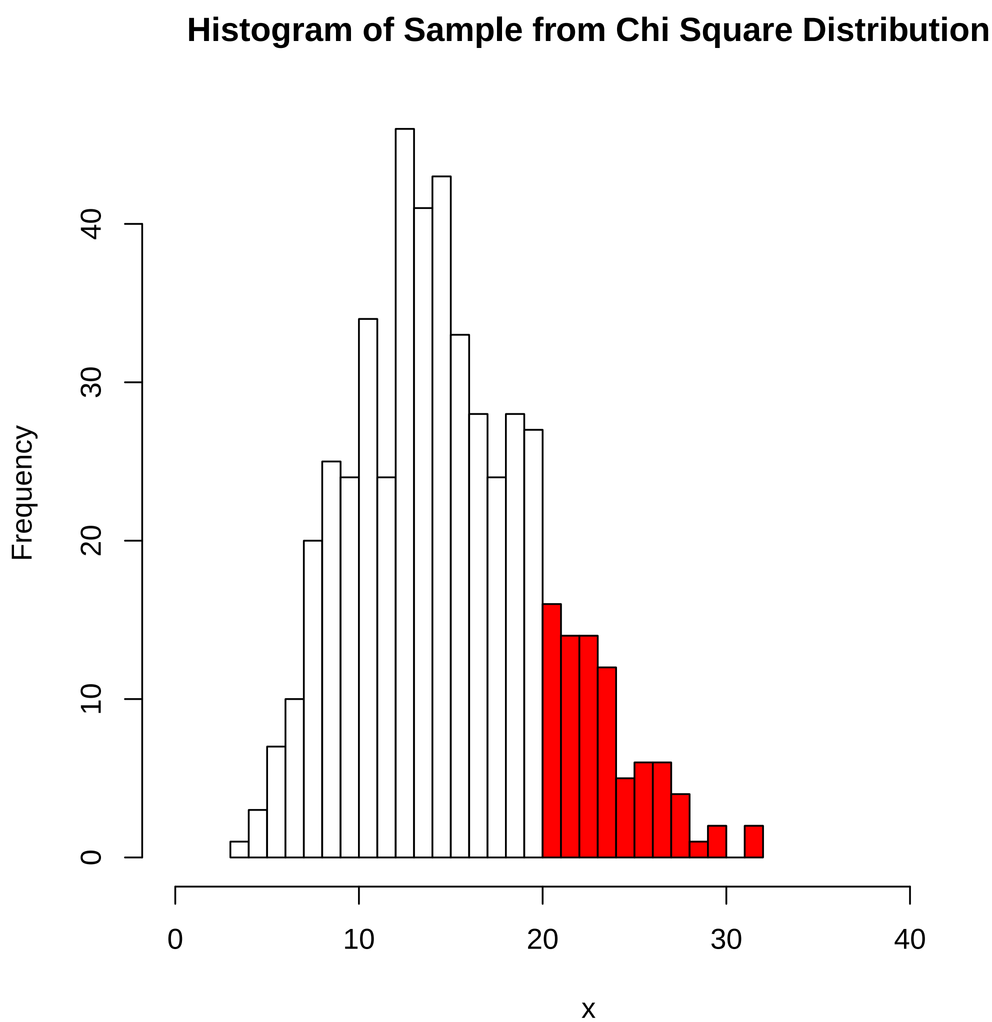
```
Problem 1
```

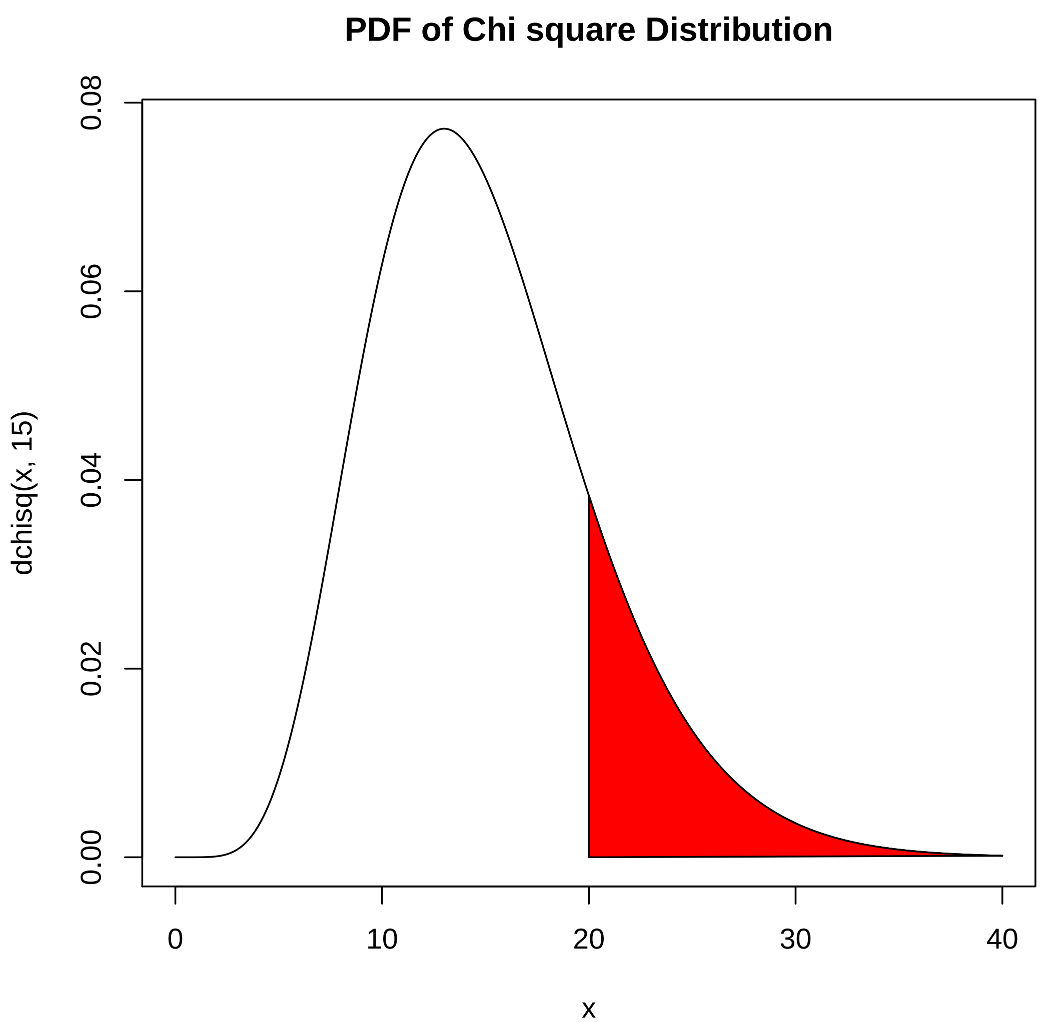
First we show the probability in histogram. We generate a sample of size=500 from Chi square distribution:

```
> x=rchisq(500,15)
> h=hist(sample,breaks=25,plot=FALSE)
> bin=cut(seq(21,37,by=1),h$breaks)
> clr=rep("white",length(h$counts))
> clr[bin]="red"
> plot(h,col=clr,xlim=c(0,45),
+ main="Histogram of Sample from Chi Square Distribution")
```



Then we show the same probability in probability density function graphic:

```
> x=seq(0,40,by=0.01)
> plot(x,dchisq(x,15),main="PDF of Chi square Distribution",type="l")
> y=dchisq(x,15)
> xx=seq(20,40,by=0.01)
> yy=dchisq(xx,15)
> yyy=seq(0,yy[1],by=0.1)
> xxx=rep(20,length(yyy))
> yy=c(yyy,yy)
> xx=c(xxx,xx)
> polygon(xx,yy,col="red")
```



Problem 2

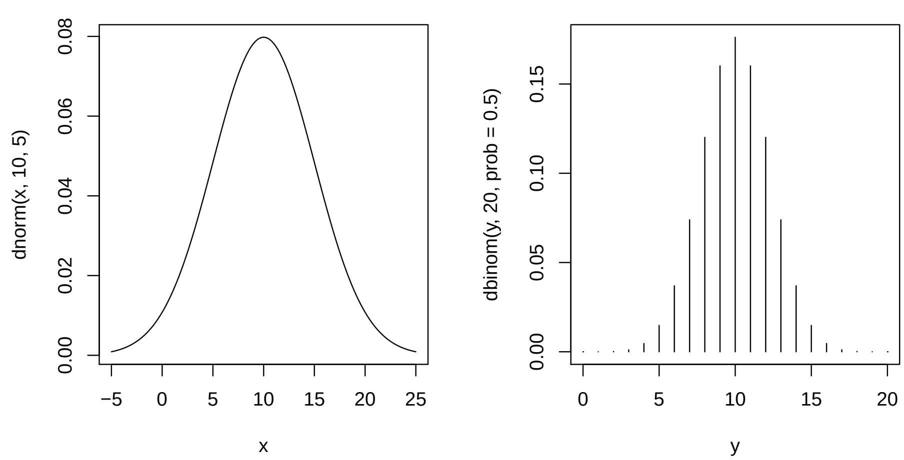
There are 2 differences:

- First, when the pdf of normal distribution is going to infinity on both sides of x-axis, the pdf of binomial distribution is finite on x-axis.
- Second, random variable of normal distribution is continuous, which makes the pdf very smooth. And random variable of binomial distribution is discrete, thus the pdf is discontinuous in lots of points.

The code is following:

```
> par(mfrow=c(1,2))
> x=seq(-4,4,by=0.01)
> plot(x,dnorm(x,0,1),type="l")
> y=0:20
> plot(y,dbinom(y,20,prob=0.2),type="h")
```

Normal vs Binomial Probability Density Function



Problem 3

(a) In this problem, it's obviously the number of successful operation follows binomial distribution, with size=20 and probability=0.8. Thus,

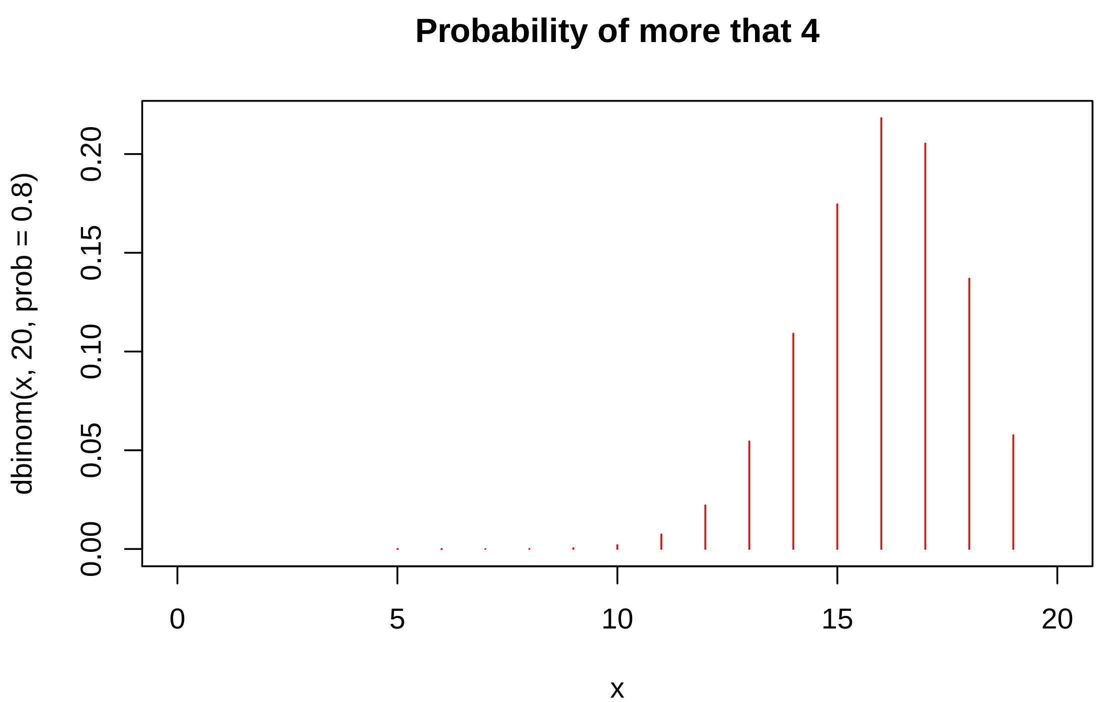
$$P(X = 4) = \binom{20}{4}(0.8)^4(0.2)^{16}$$

(b) We could use the rule of complement:

$$\begin{aligned}
P(X>4) =& 1-P(X\le 4)\\
=& 1-P(X=0)-P(X=1)-P(X=2)-P(X=3)-P(X=4)\\
=& 1-\binom{20}{0}(0.8)^0(0.2)^{20}-\binom{20}{1}(0.8)^1(0.2)^{19}-\binom{20}{2}(0.8)^2(0.2)^{18}\\
&-\binom{20}{3}(0.8)^3(0.2)^{17}-\binom{20}{4}(0.8)^4(0.2)^{16}
\end{aligned}$$

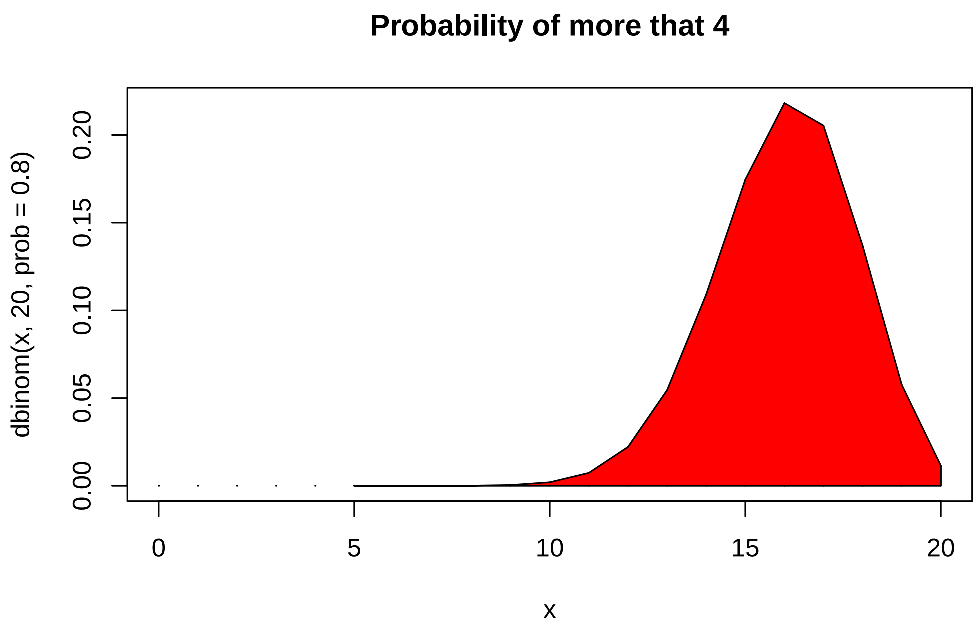
(c) There are 2 methods to show the graphic, just notice in the first one, the first 5 points are not red which are almost invisible. The first one is:

```
> x=0:20
> y=seq(5,20,by=1)
> bin=cut(y,x)
> clr=rep("white",length(x))
> clr[bin]="red"
> plot(x,dbinom(x,20,prob=0.8),col=clr,type="h",
+ main="Probability of more that 4")
```



The second one is:

```
> x=0:20
> plot(x,dbinom(x,20,prob=0.8),type="h", main="Probability of more that 4")
> xx=5:20
> yy=dbinom(xx,20,prob=0.9)
> yyy=seq(0,yy[1],by=0.0001)
> xxx=rep(20, length(yyy))
> xx=c(xxx,xx)
> yy=c(yyy,yy)
> polygon(xx,yy,col="red")
```



(d) Actually none of below is right. Because in this case, $p = 0.8$ but not 0.2. If we all change them to 0.8, then the following are correct for part (a):

```
> pbinom(4,size=20,prob=0.80)-pbinom(3,size=20,prob=0.80)
> dbinom(4,size=20,prob=0.80)
```

And the one below is correct for part (b)

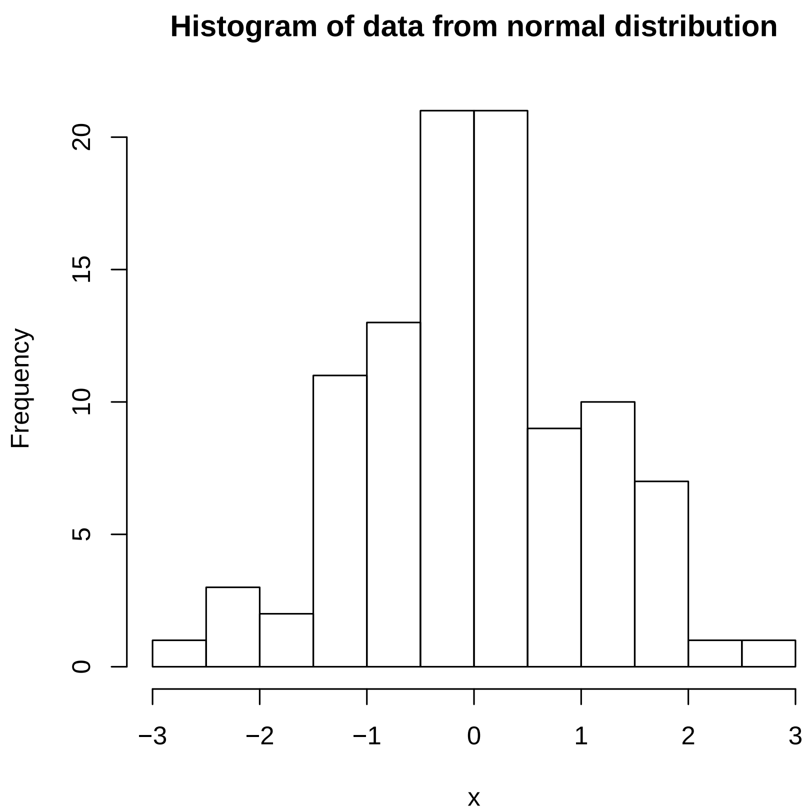
```
> 1-pbinom(4,size=20,prob=0.80)
```

Problem 4

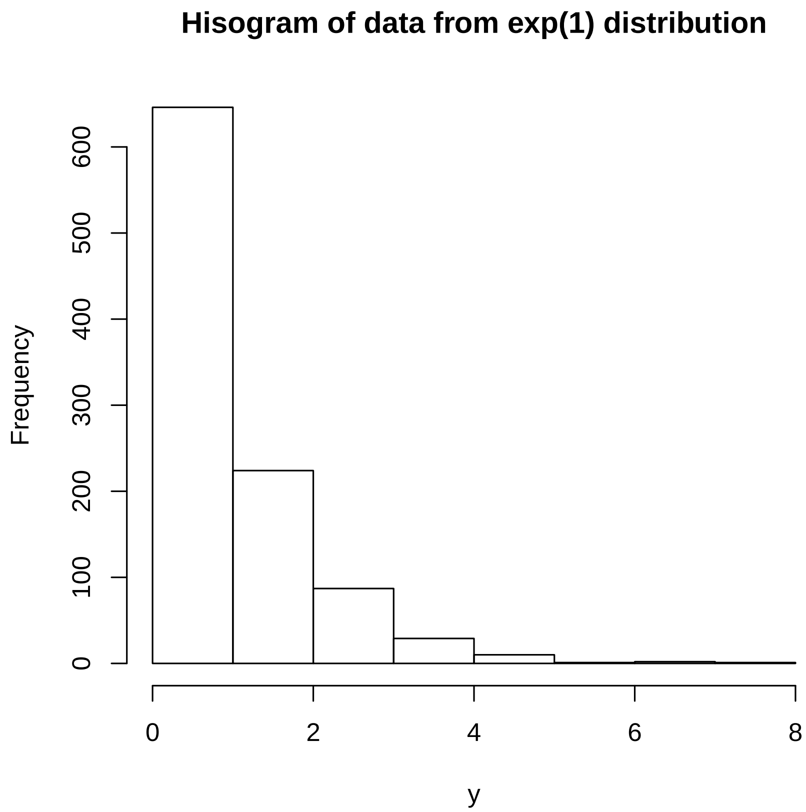
(a) We use the following R commands:

```
> x = rnorm(100)
> y = rexp(1000, 1)
> hist(x, main = "Histogram of data from normal distribution")
> hist(y, main = "Histogram of data from exp(1) distribution")
```

Which produces the following two plots:



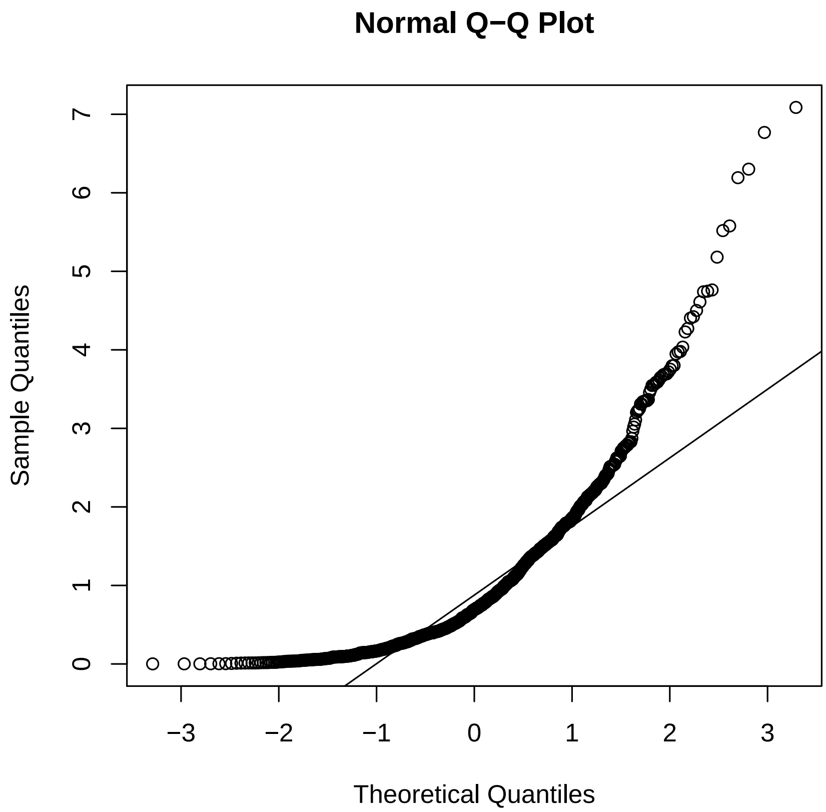
and



(b) We use R command:

```
> qqnorm(y)
> qqline(y)
```

and see plot

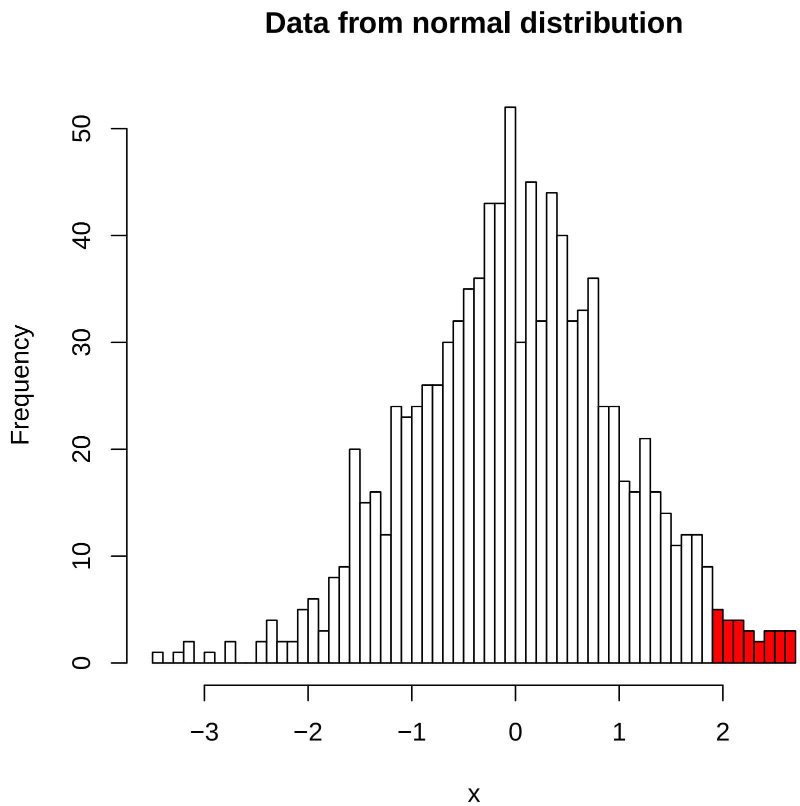


which shows that the exponential(1) distribution has a heavier right tail and a lighter left tail than the normal distribution.

(c) Since $P(X \geq 5)$ is less than .01%, we have that $P(2 < X < 5) \sim P(X > 2) \approx .9772$. To see this graphically, we use commands

```
> x = rnorm(1000)
> h<- hist(x, plot=FALSE, breaks = 50)
> h$breaks
>bin<-cut(c(2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8), h$breaks)
clr<-rep("white", length(h$counts))
clr[bin]<-"red"
plot(h, col = clr, main = "Data from normal distribution")
```

to get the following figure:



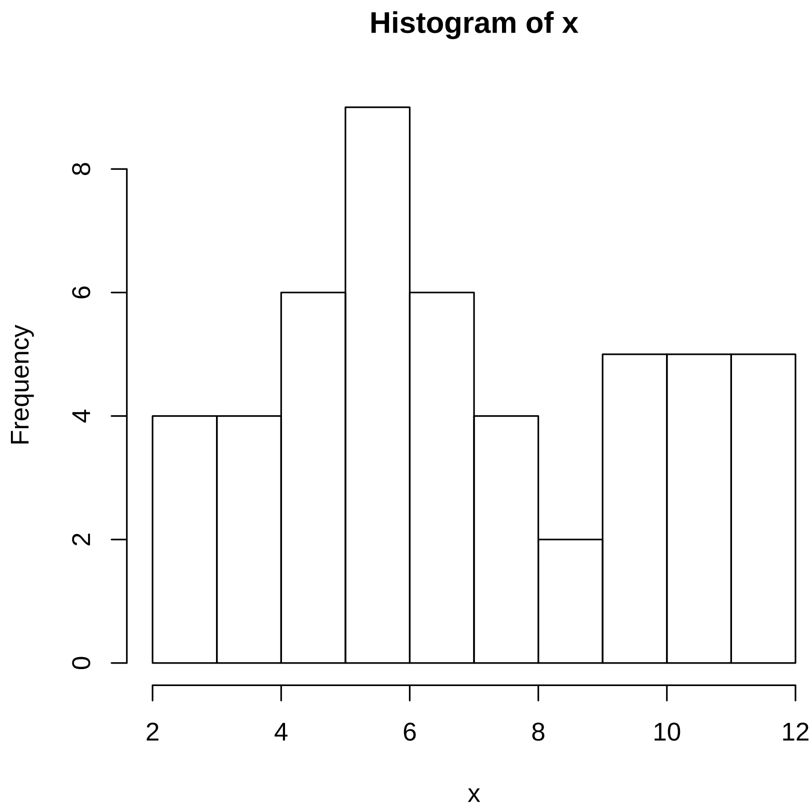
Problem 5

(a) The proportion of memory sticks that will be scrapped is the proportion of times that the uniform distribution between 2 and 12mm will lie between 10 and 12. This proporition is $\frac{12-10}{12-2} = .2$

(b) We use commands:

```
> x=runif(50, 2, 12)
> mean(x)
> var(x)
> hist(x)
```

and obtain plot



My sample mean was 6.5 and my sample variance was 7.65. My data deviates from a uniform distribution a bit, but within expectation based on only 50 trials.

`Problem 6`

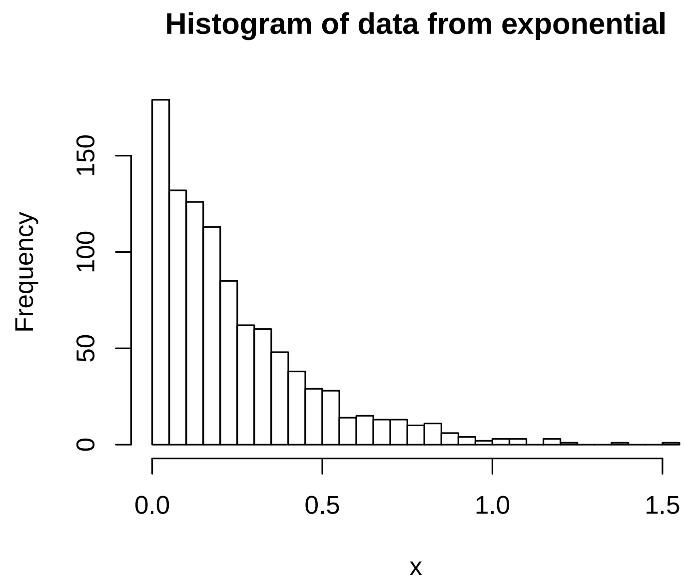
(a) If we want interarrival times of a Poisson random variable with $\lambda = 4$, then we just use an exponential random variable with the same λ. To generate 1000 samples, we just use:

```
> x = rexp(1000,4)
```

Plotting the histogram just involves the command:

```
> hist(x,breaks=30)
```

giving us something that looks like this:



(b) To find the mean of our data, we use the mean() command:

```
> mean(x)
```

Since the expected value of our exponential random variable is $\frac{1}{\lambda} = 0.25$, we should expect something close to that. (When I ran this, I got 0.2433735, which is indeed close to 0.25.)

(c) $P(T \leq 0.25)$ can be computed with the following command:

```
> pexp(0.25,4)
```

We could also estimate it from our data using the ecdf() function:

```
> F = ecdf(x)
> F(0.25)
```

I get 0.6321206 from the first method, and 0.634 from the second (keep in mind that the latter will vary slightly, since we all get a different collection of samples when we use rexp).

`Problem 7`

(a) We have $\mu = 100, \sigma = \sqrt{256} = 16$. So $P(X \leq 120) = P(Z \leq \frac{120-100}{16}) = P(Z \leq \frac{20}{16})$, where Z is a standard normal. We either look that up in a table, or type into R:

```
> pnorm(1.25,0,1)
```

The result is approximately 0.8944.

(b) $P(X \leq 120 | X \geq 110) = P(110 \leq X \leq 120)/P(X \geq 110)$ by the conditional probability formula. If F(x) denotes the cdf of a standard normal, our answer is $\frac{F(20/16)-F(10/16)}{1-F(10/16)}$. I get 0.6027988 when I use R.

If we want to find the answer via simulation, we could do the following:

```
> x = rnorm(10000,100,16)
> y = x[x>110]
> z = y[y<120]
> length(z)/length(y)
```

I took 10000 random samples from our distribution and stored them in x. y is the array of entries of x that are larger than 110. z is the array of entries of x between 110 and 120. The proportion of entries of y that are in z gives us the probability of seeing a number less than 120, given the number is larger than 110.

I got 0.6075528 when I ran those commands.

(c) To estimate the IQR using R, just enter:

```
> quantile(x,0.75) - quantile(x,0.25)
```

where x is the array of samples we generated earlier.

`Problem 8`

(a) Since normals are symmetric about their mean/median, for a standard normal we have $P(X \leq -a) = P(X \geq a)$. So $P(|X| \geq 2\sigma) = P(|X| \geq 2) = 2P(X \leq -2) = 2F_X(-2)$. In R, typing:

```
> 2*pnorm(-2,0,1)
```

results in 0.04550026, which is nearly 0.05. So, which interval centered around the mean contains exactly 95

```
> qnorm(.025,0,1)
```

I get -1.959964. That's very nearly 2! So the interval [-1.96,1.96] contains 95

If you do the same computation for a nonstandard normal, upon transforming to a standard normal, you will just end up with the previous calculation: $P(|X-\mu| \geq 2\sigma) = P(|\frac{X-\mu}{\sigma}| \geq 2)$

(b) To find the positions of the quartiles, we can just find the 3rd quartile, since the 1st quartile will be equally distant from the mean in the opposite direction. The 3rd quartile of a standard normal can be approximately found in a table. Looking in a table, we see that it's between 0.67 and 0.68. If you type into R:

```
> qnorm(0.75,0,1)
```

you can find the 3rd quantile of a standard normal. I get 0.6744898. Since the standard deviation of a standard normal is 1, we get the 1st and 3rd quantiles are approximately 0.68 standard deviations away from the mean.

As in (a), if you try this for a nonstandard normal, it'll just turn into the computation we just did. In other words, the 3rd quartile of a nonstandard normal will be the point a such that $P(\frac{X-\mu}{\sigma} \leq \frac{a-\mu}{\sigma}) = 0.75$, so $\frac{a-\mu}{\sigma} = 0.68$, which means precisely that a is 0.68 standard deviations away from the mean.