R Program Notes
Biostatistics: A Guide to Design, Analysis, and Discovery Second Edition
by Ronald N. Forthofer, Eun Sul Lee, Mike Hernandez
Chapter 5: Probability Distributions

Program Notes Outline

Chapter 5 Formulas

Distribution		Formula	Characteristics
Binomial	Probability Mass Function	$\Pr(X = x) = \binom{n}{x}\pi^x(1-\pi)^{1-x}$ where $\mathbf{x} = 0, 1, 2, \ldots, n$	$\boldsymbol{\pi}$ is the probability of success for any trial and **n** is the number of trials
Poisson	Probability Mass Function	$\Pr(X = x) = \frac{e^{-\mu}\mu^x}{x!}$ where $\mathbf{x} = 0, 1, 2, \ldots$	$\boldsymbol{\mu}$ is the parameter of the Poisson distribution that refers to both the mean and variance
Normal	Probability Density Function	$f(x) = \frac{1}{\sqrt{2\pi \cdot \sigma^2}}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$ where $-\infty < \mathbf{x} < \infty$	Where $\boldsymbol{\mu}$ is the mean and $\boldsymbol{\sigma^2}$ is the variance

Program Note 5.1 – Finding binomial and Poisson probabilities

1. Finding binomial probabilities:

Consider the mathematical notation for the binomial distribution as shown below:

$$Pr(X = x) = \binom{n}{x}\pi^x(1-\pi)^{n-x},\ where\ x = 0, 1, 2, \cdots, n.$$

In the textbook, we use the notation ***X ~ Binom(n,p)*** to indicate that ***X*** is a random variable that follows a binomial distribution with ***n*** trials with a probability of success on any given trial equal to ***p***. In **R**, the command **dbinom(x, size, prob, log = FALSE)** is used to compute probabilities from the binomial distribution where **x** specifies the number of successes, **size** refers to the number of trials, and **prob** is the probability of success on any given trial. For example, if we have 4 trials and the probability of success on any given trial is 0.25 (i.e. ***X~Binom(4,0.25)***), then probability of observing 0 successes is expressed in mathematical notation as:

$$Pr(X = 0) = \binom{4}{0}(0.25)^0(1-0.25)^{4-0} = 0.3164$$

The *Prob(X = 0)* can be computed using **R** as shown below.

```
R commands:

dbinom(x=0, size=4, prob=0.25)


R output:

0.3164063
```

To calculate the *Prob(X < 2)*, given that there are 4 trials and the probability of success on any given trial is 0.25, expressed in mathematical notation as:

$$Pr(X < 2) = \binom{4}{0}(0.25)^0(1-0.25)^{4-0} + \binom{4}{1}(0.25)^1(1-0.25)^{4-1},$$

we could use the **R** commands below.

R commands:

```
cumprob<- dbinom(x=0, size=4, prob=0.25) + dbinom(x=1, size=4, prob=0.25)
cumprob
```

R output:

```
0.7382812
```

An easier approach for calculating cumulative probabilities is to use **R**'s **pbinom** function. Therefore, to calculate the *Prob(X < 2)* given that there are 4 trials and the probability of success on any given trial is 0.25, we simply use the **R** command: **pbinom(q=1, size = 4, prob= 0.25)** and obtain the same result as shown below.

R commands:

```
pbinom(q=1, size = 4, prob= 0.25)
```

R output:

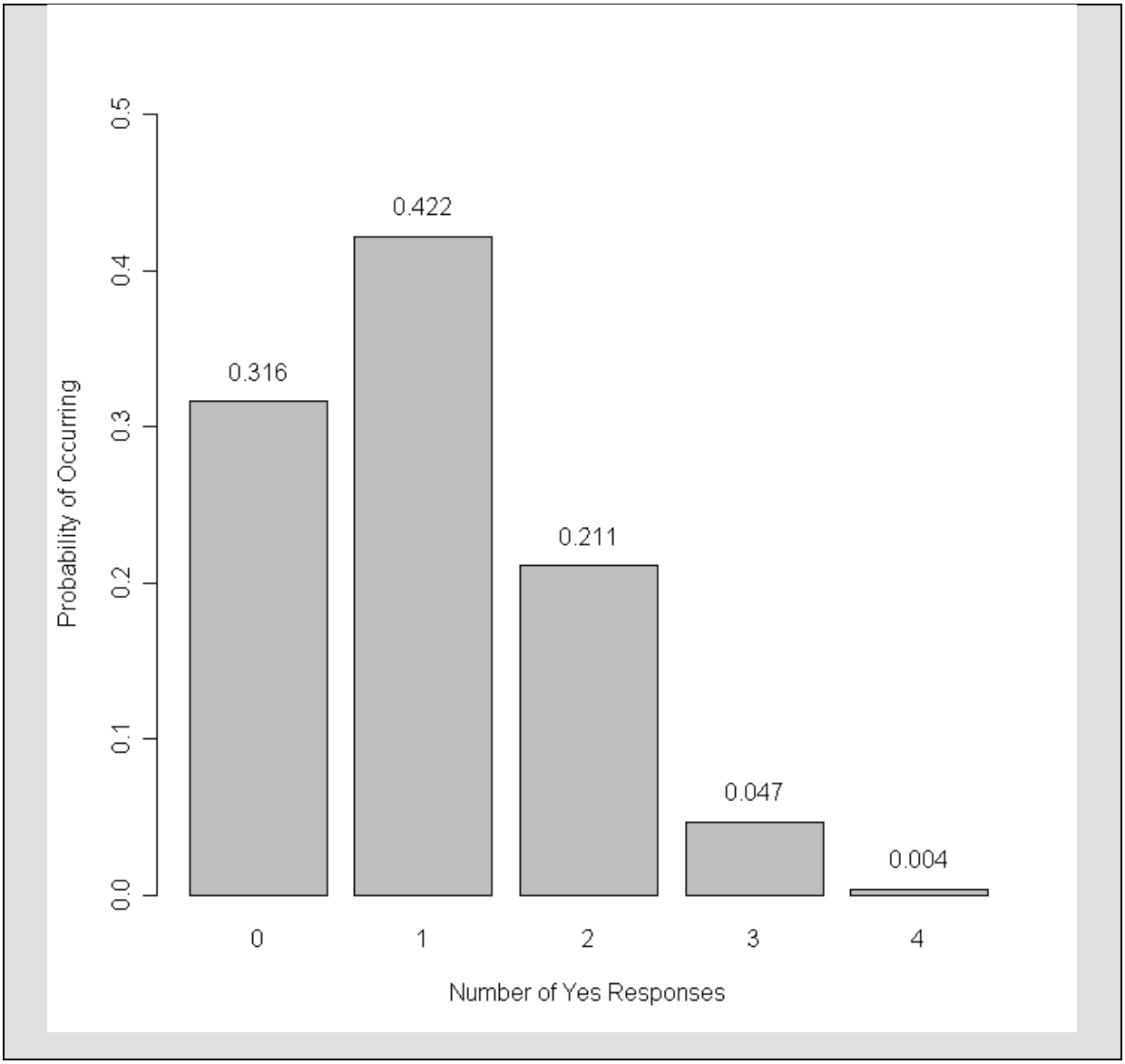
```
0.7382812
```

Below we show the R commands used to create the **Figure 5.1 Bar chart showing the binomial distribution for n=4 and π=0.25** in the textbook.

R commands:

```
barplot(dbinom(x=0:4, size=4, prob=0.25),names.arg=0:4,
ylim=c(0,0.5),  ylab="Probability  of  Occurring",  xlab="Number  of  Yes
Responses")
x<-c(0.7, 1.9, 3.1, 4.3, 5.5)
y<-round(dbinom(x=0:4, size=4, prob=0.25),digits=3)
text(x,y+0.02,label=c(y))
```

R output:



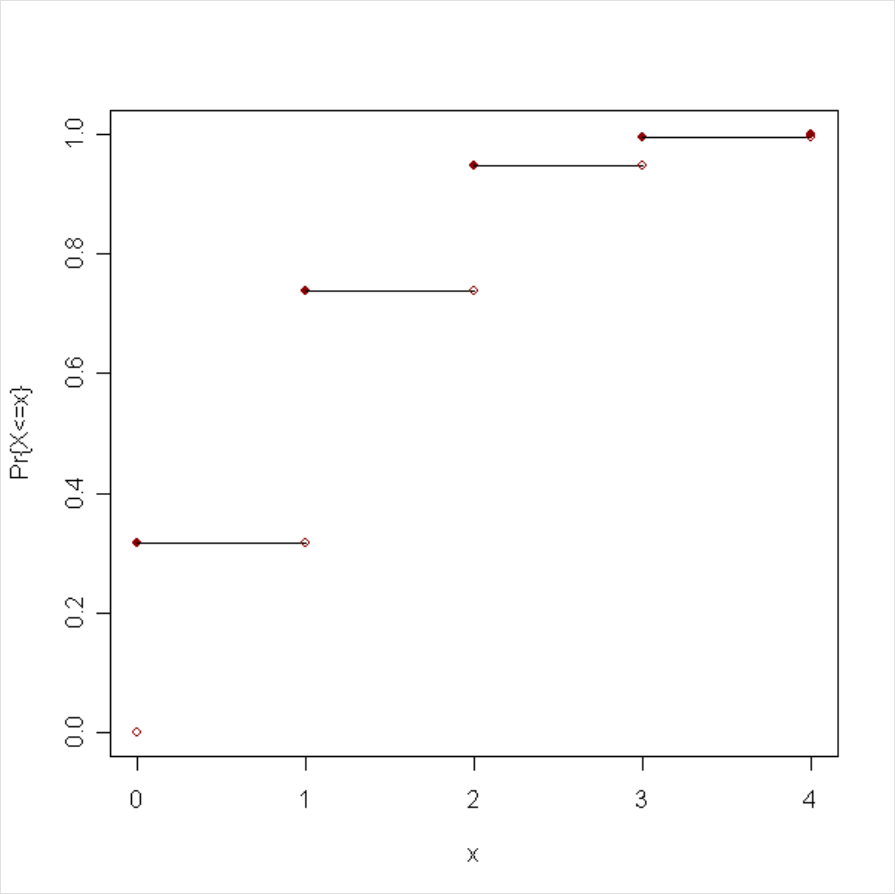
The **R** code below produces a plot similar to **Figure 5.2 Cumulative binomial distribution for n = 4 and π = 0.25** in the textbook.

R commands:

```
x<- c(0,4)
y<- c(0,1)
plot(x,y,type="n",xlab="x",ylab="Pr{X<=x}")

points(0,0,type="p",col="dark red")

points(0,pbinom(q=0, size=4, prob=0.25),type="p", pch=19,col = "dark red")
segments(0,pbinom(q=0, size=4, prob=0.25),1,pbinom(q=0, size=4, prob=0.25))
points(1,pbinom(q=0, size=4, prob=0.25),type="p", col = "dark red")
```

```
points(1,pbinom(q=1, size=4, prob=0.25),type="p", pch=19,col = "dark red")
segments(1,pbinom(q=1, size=4, prob=0.25),2,pbinom(q=1, size=4, prob=0.25))
points(2,pbinom(q=1, size=4, prob=0.25),type="p", col = "dark red")

points(2,pbinom(q=2, size=4, prob=0.25),type="p", pch=19,col = "dark red")
segments(2,pbinom(q=2, size=4, prob=0.25),3,pbinom(q=2, size=4, prob=0.25))
points(3,pbinom(q=2, size=4, prob=0.25),type="p", col = "dark red")

points(3,pbinom(q=3, size=4, prob=0.25),type="p", pch=19, col = "dark red")
segments(3,pbinom(q=3, size=4, prob=0.25),4,pbinom(q=3, size=4, prob=0.25))
points(4,pbinom(q=3, size=4, prob=0.25),type="p", col="dark red")

points(4,1,type="p",pch=19, col="dark red")
```

R output:

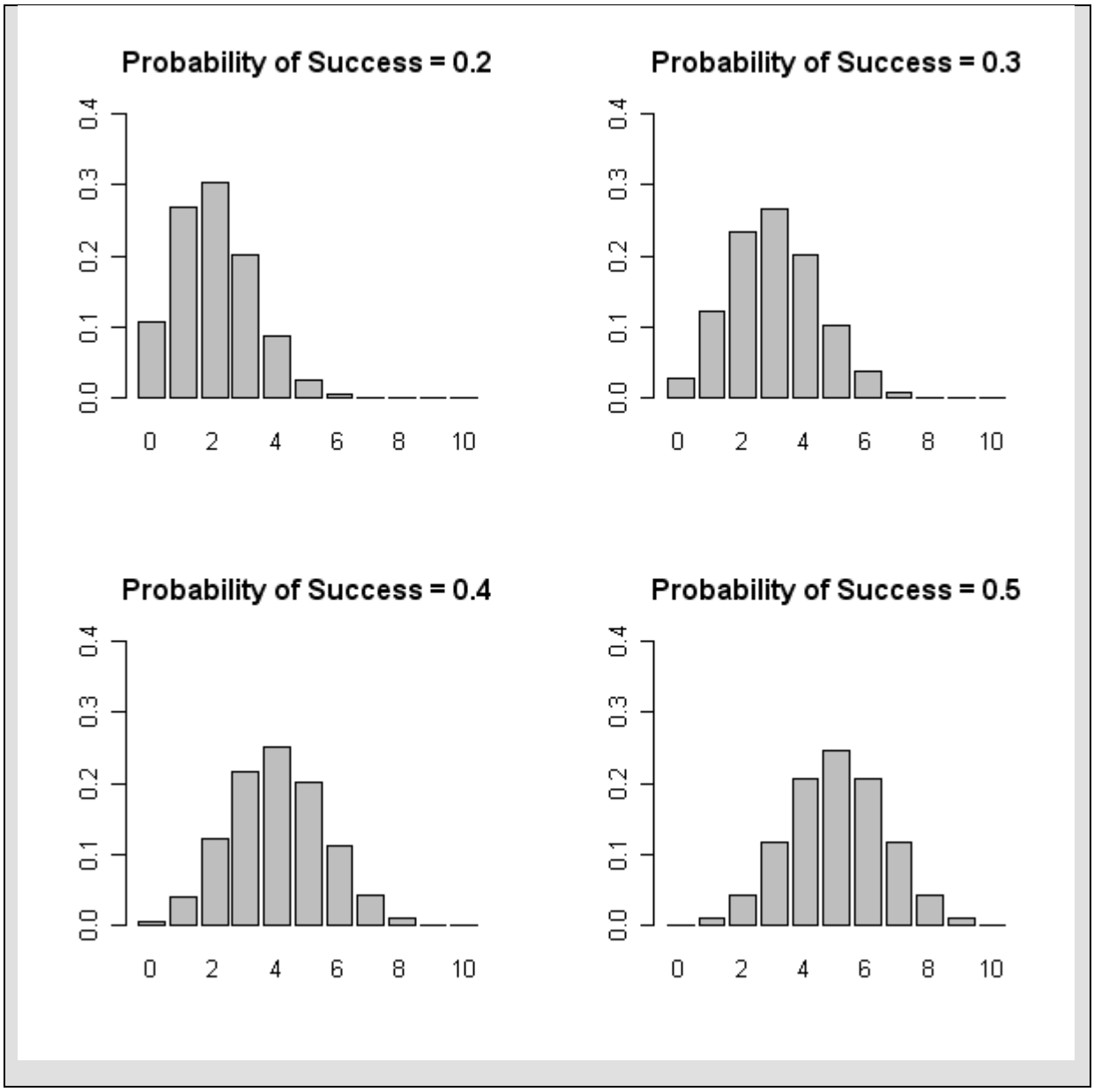


To show how the appearance of the binomial distribution changes as the probability of success changes, we plot the binomial distribution with 10 trials and change the probability of success on any given trial from 0.2 to 0.5 using increments of 0.1. Notice how the distribution becomes more symmetric as the probability of success gets closer to 0.5. See **Figure 5.3 Binomial probabilities for n = 10 and π = 0.1, 0.2, and 0.5** in the textbook.

R commands:

```
par(mfrow=c(2,2))

barplot(dbinom(x=0:10, size=10, prob=0.2),names.arg=0:10,
ylim=c(0,0.4), main="Probability of Success = 0.2")

barplot(dbinom(x=0:10, size=10, prob=0.3),names.arg=0:10,
ylim=c(0,0.4), main="Probability of Success = 0.3")

barplot(dbinom(x=0:10, size=10, prob=0.4),names.arg=0:10,
ylim=c(0,0.4), main="Probability of Success = 0.4")

barplot(dbinom(x=0:10, size=10, prob=0.5),names.arg=0:10,
ylim=c(0,0.4), main="Probability of Success = 0.5")
```

R output:



2. Finding Poisson probabilities:

Consider the mathematical notation for the Poisson distribution as shown below:

$$Pr(X = x) = \frac{e^{-\mu}\mu^{x}}{x!}, \quad where\ x = 0, 1, 2, \cdots$$

In the textbook, we use the notation ***X ~ Poiss(μ)*** to indicate that ***X*** is a random variable that follows a Poisson distribution with Poisson parameter **μ**. In **R**, the command **dpois(x, lambda,**

log = FALSE) is used to compute probabilities from a Poisson distribution. For example, for μ = 2,

$$Pr(X = 3) = \frac{e^{-2}2^3}{3!} = 0.1804$$

as shown in the textbook. The **R** commands to do this are shown below.

R commands:

```
dpois(x=3,lambda=2)
```

R output:

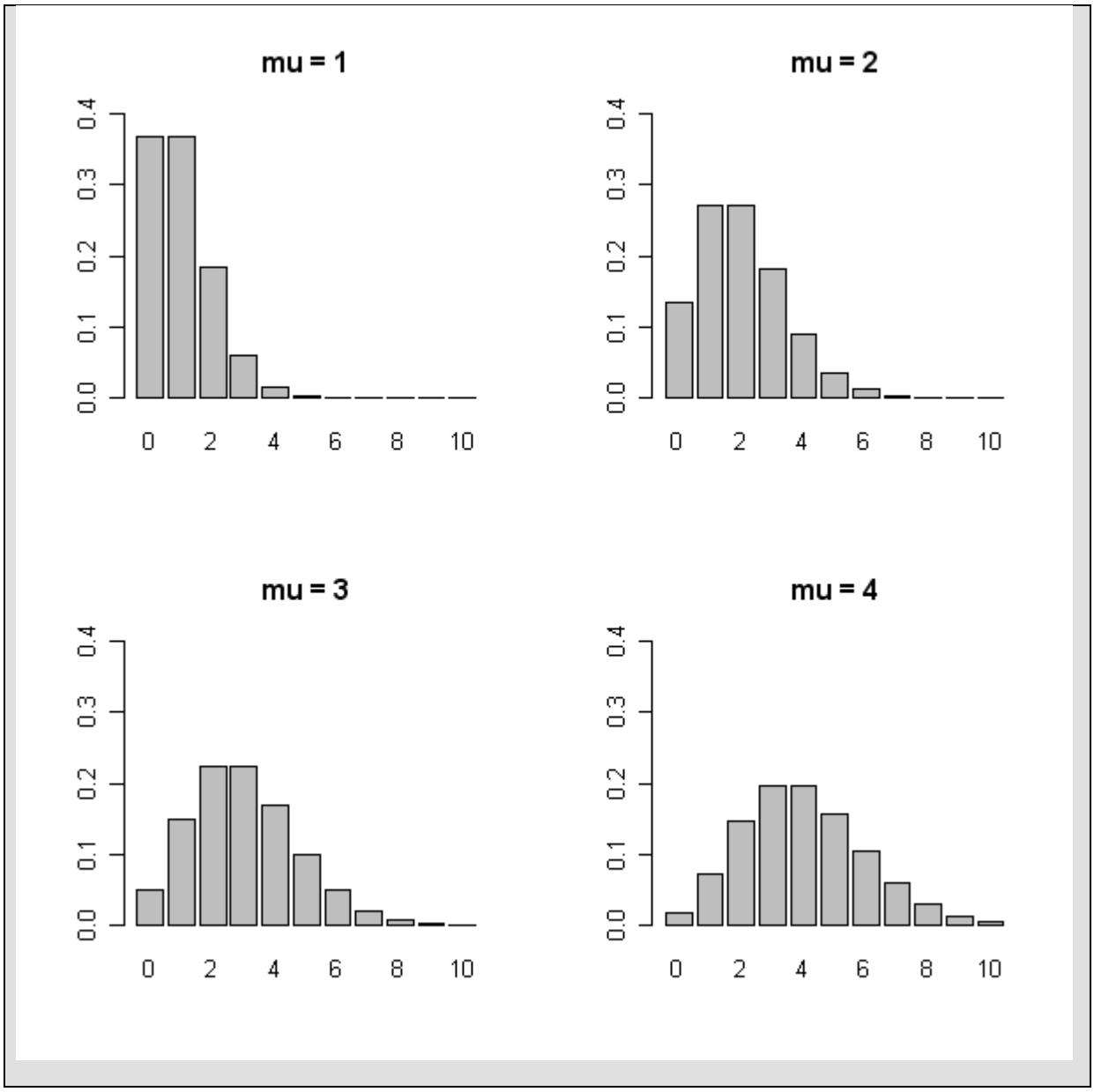
```
0.1804470
```

To show how the appearance of the Poisson distribution changes as μ changes, we plot the probability mass function for the Poisson distribution with values of μ increasing from 1 to 4 as shown below.

R commands:

```
par(mfrow=c(2,2))

barplot(dpois(x=0:10, lambda=1),names.arg=0:10,
ylim=c(0,0.4), main="mu = 1")

barplot(dpois(x=0:10, lambda=2),names.arg=0:10,
ylim=c(0,0.4), main="mu = 2")

barplot(dpois(x=0:10, lambda=3),names.arg=0:10,
ylim=c(0,0.4), main="mu = 3")

barplot(dpois(x=0:10, lambda=4),names.arg=0:10,
ylim=c(0,0.4), main="mu = 4")
```

R output:


mu = 1
mu = 2
mu = 3
mu = 4
0.0
0.1
0.2
0.3
0.4
0
2
4
6
8
10

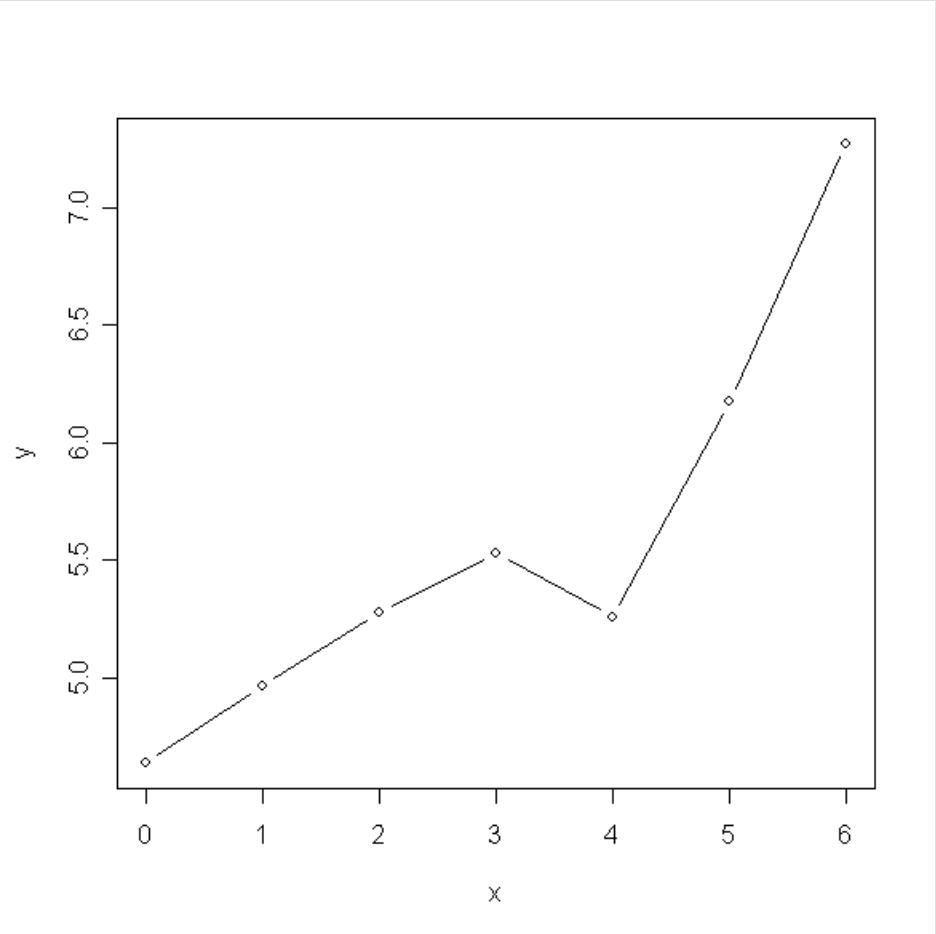
Program Note 5.2 – Creating a Poissonness plot

Using the frequency values from **Table 5.4**, we can create a *Poissonness plot* like the one shown in **Figure 5.5** in the textbook. Basically we are plotting { *ln(freq(x)) + ln(x!) versus x* }. This is a simple scatter plot that can be created using the **R** commands below. In **R**, the function **log( )** refers to the natural logarithm. Use **help(log)** in **R** for more information on changing the base of a logarithm.

R commands:

```
x<- c(0,1,2,3,4,5,6)
freq<- c(103,143,98,42,8,4,2)
y<- log(freq)+log(factorial(x))
plot(y~x,type="b")
```

R output:



Program Note 5.3 – **Finding normal probabilities**

Consider the mathematical notation for the probability density function (pdf) for the normal distribution as shown below:

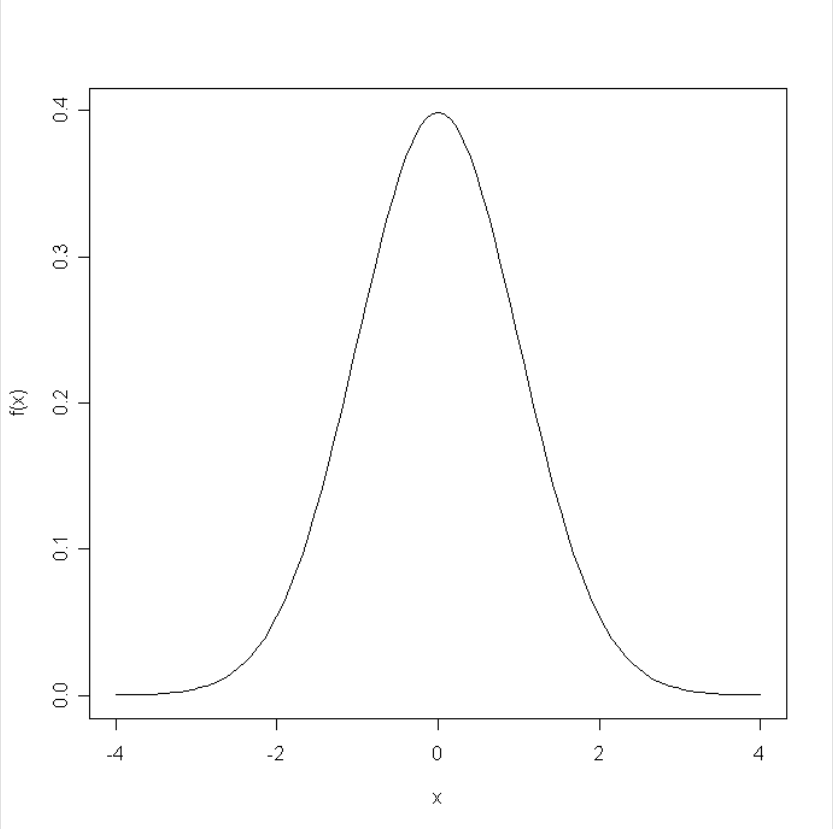
$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad -\infty < x < \infty$$

Allowing μ = 0 and σ =1, we get the probability density function for the standard normal distribution. The **curve** function in **R** is used to draw functions over a specified interval. Below, we use the expression for the standard normal distribution and draw its shape using the **curve** function on the interval from -4 to 4 as shown in **Figure 5.6 pdf of standard normal distribution**.

R commands:

```
curve(1/sqrt(2*pi)*exp(-(x^2)/2), ylab="f(x)", -4, 4)
```

R output:



To calculate the Pr{X ≤ x} where X is a random variable that follows a normal distribution with mean = μ and variance = σ^2, we can use the **R** function **pnorm(q, mean=, sd=, lower.tail=TRUE, log.p=FALSE)** where the user must specify the mean and standard deviation of the normal distribution being used. For example to calculate Pr{X ≤ 95} given that X is a random variable that follows a normal distribution with mean = 80 and standard deviation = 10, we get 0.9331928 using **pnorm(q=95, mean=80, sd=10, lower.tail=TRUE, log.p=FALSE)**.

```
R commands:

pnorm(q=95, mean=80, sd=10, lower.tail=TRUE, log.p=FALSE)


R Output:

0.9331928
```

In **Example 5.3** in the textbook, we describe how to calculate Pr{X > 95} given that X is a random variable that follows a normal distribution with mean = 80 and standard deviation = 10. Since Pr{X > 95} = 1 - Pr{X ≤ 95}, we can use the following **R** commands below.

```
R commands:

1 - pnorm(q=95, mean=80, sd=10, lower.tail=TRUE, log.p=FALSE)

R Output:

0.0668072
```

The **R** commands below provides the same result.

```
R commands:

pnorm(q=95, mean=80, sd=10, lower.tail=FALSE, log.p=FALSE)

R Output:

0.0668072
```

In **Example 5.4** in the textbook, we describe how to find the 95th percentile of a normal distribution with mean = 80 and standard deviation = 10. To do this in **R**, we use the function **qnorm(p=0.95, mean=80, sd=10, lower.tail=TRUE, log.p=FALSE)** as shown below.

R commands:

```
qnorm(p=0.95, mean=80, sd=10, lower.tail=TRUE, log.p=FALSE)
```

R Output:

```
96.44854
```

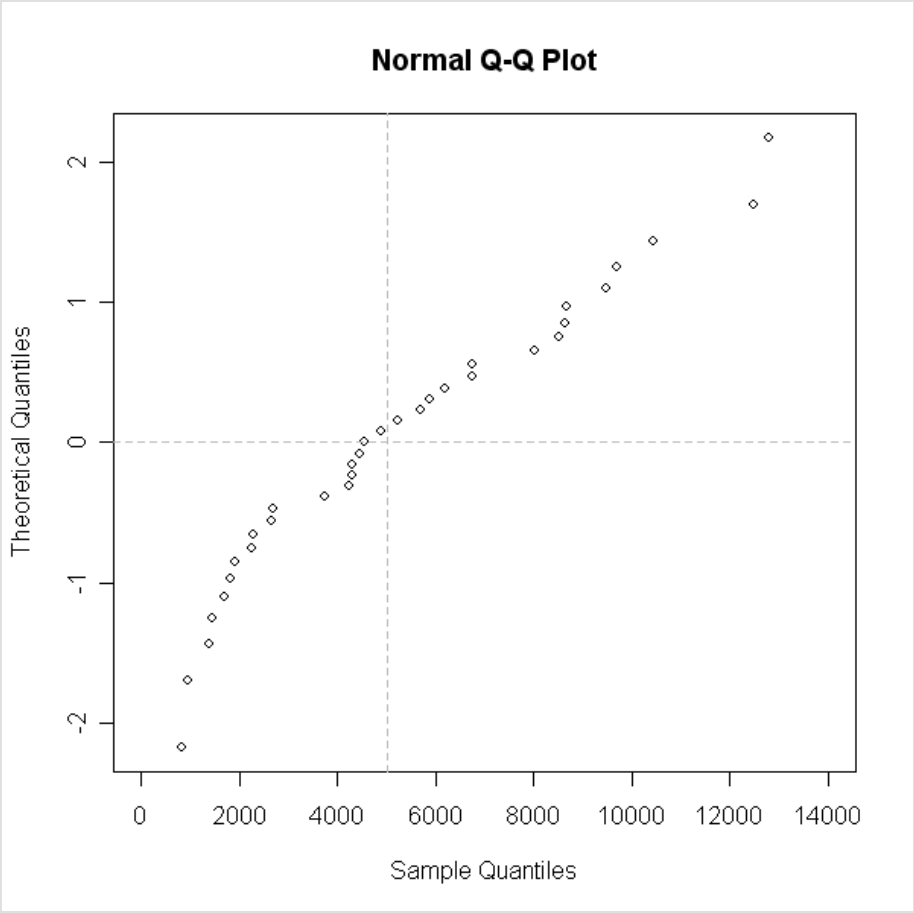
Program Note 5.4 – Creating a normal probability plot

We can create a normal probability plot like the one in **Example 5.6** using the **R** commands below. Note that the example in the textbook uses percentiles along the y-axis while the plot **R** produces using the **qqnorm( )** function uses standard normal deviates. Therefore the 50^{th} percentile of the standard normal distribution occurs where the standard normal deviate is 0.

R commands:

```
vit.a<-c(820, 964, 1379, 1459, 1704, 1826, 1921, 2246, 2284, 2671, 2687,
3747, 4248, 4288, 4315, 4450, 4535, 4876, 5242, 5703, 5874, 6202,
6754, 6761, 8034, 8516, 8631, 8675, 9490, 9710, 10451, 12493, 12812)

qqnorm(y=vit.a,datax=TRUE,ylim=c(0,14000))
abline(v=5000,lty=2,col="gray")
abline(h=0,lty=2,col="gray")
```

R Output:



To create **Figure 5.12** in the textbook, we generate data from a normal distribution with mean = 80 and standard deviation = 10 and then use the data to create a Normal Q-Q plot. In **R**, we can use the **rnorm(n= , mean= , sd=)** function to generate random samples of size **n** from a normal distribution with a specific mean and standard deviation. Notice in the **R** commands below that we first generate a sample of 200 observations from a normal distribution with mean = 80 and standard deviation = 10 and then we use the **qqnorm( )** and **qqline( )** functions to create the Normal Q-Q plot.

R commands:

```
# Generate a random sample of 200 observations
set.seed(123)
obs<-rnorm(n=200, mean=80, sd=10)

# Create a Normal Q-Q Plot
qqnorm(y=obs,datax=TRUE)
qqline(y=obs,datax=TRUE)
```

R Output: