The University of Sydney
School of Mathematics and Statistics

# Solutions to Tutorial Weeks 7 and 8

MATH1905: Statistics (Advanced) Semester 2, 2017

Web Page: `http://sydney.edu.au/science/maths/MATH1905`
Lecturer: Michael Stewart

*There is a quiz in week 7 but these exercises for weeks 7 and 8 are provided ahead of time. Also please complete any unfinished exercises from week 6 and discuss any difficulties with your tutor, or attend a consultation session.*

**1.** (**Multiple Choice**) The expected value, $E(X)$ of the random variable $X$ having probability distribution

| $x$ | 2 | 4 | 6 |
|---|---|---|---|
| $P(X = x)$ | 0.1 | 0.3 | 0.6 |

is

(a) 4
(b) 0.3
(c) 0.5
(d) 5

***Solution:***

$$E(X) = \sum_x xP(X = x) = (2 \times 0.1) + (4 \times 0.3) + (6 \times 0.6) = 0.2 + 1.2 + 3.6 = 5\,.$$

So the correct answer is (d).

**2.** Use R to simulate a set of $n = 25$ observations from the distribution given in the previous question using the `sample()` function, then find the sample mean:

```
x = c(2,4,6)                    # the possible outcomes

p = c(0.1,0.3,0.6)              # the probabilities associated
                                # with each outcome

samp = sample(x,size=25,replace=TRUE,prob=p)
mean(samp)
```

The mean you obtained is probably pretty close to the true value. Let's try running this experiment 10,000 times, each time generating a sample of size $n = 25$ and capturing the sample mean:

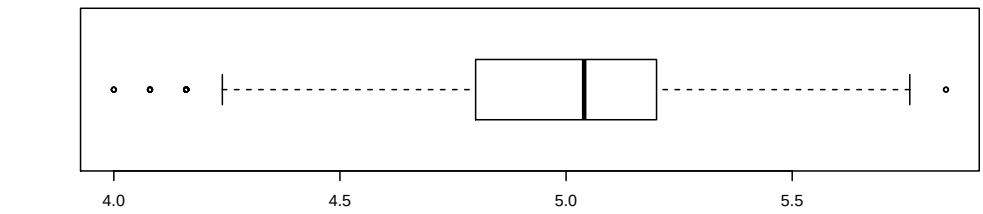
```
mx = 0                          # initialise an object that we
                                # will use to store the means

for(i in 1:10000){              # start the loop

                                # at each iteration generate
                                # a sample of size 25:
        samp = sample(x,size=25,replace=TRUE,prob=p)

                                # at each iteration calculate
```

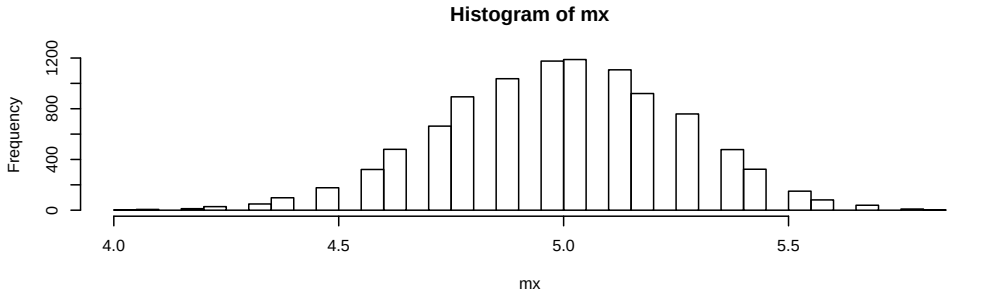
```
                                # the sample mean and store it:
        mx[i] = mean(samp)
}
```

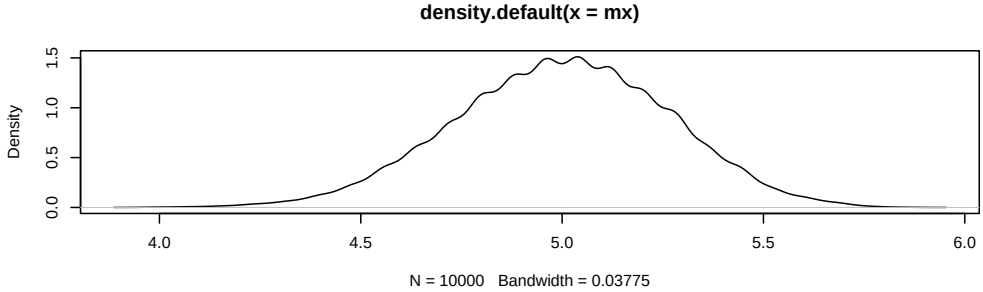
Now calculate the mean of the vector of means, `mx`. Is it closer to the true value? Also plot a boxplot, histogram and also an estimated density (like a smoothed histogram) using `plot(density(mx))`. What do you notice? Try increasing the sample size to $n = 500$. Does anything change? What phenomenon (or theorem) are we observing here?

***Solution:*** We are observing the central limit theorem in action. The data are discrete (far from a normal distribution) but the distribution of the sample mean is pretty close to normal - and gets closer to a normal as the sample size grows.

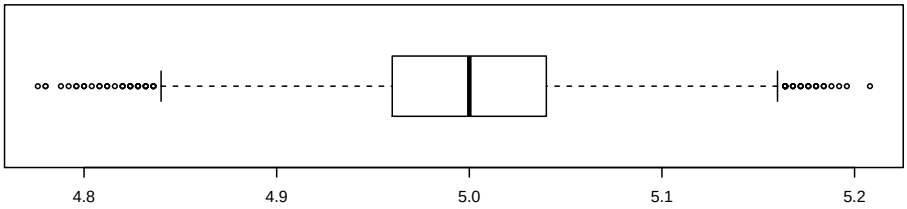
```
par(mfrow=c(3,1))
x = c(2,4,6)
p = c(0.1,0.3,0.6)
mx = 0
for(i in 1:10000){
        samp = sample(x,size=25,replace=TRUE,prob=p)
        mx[i] = mean(samp)
}
boxplot(mx,horizontal=T)
hist(mx,n=50)
plot(density(mx))
```

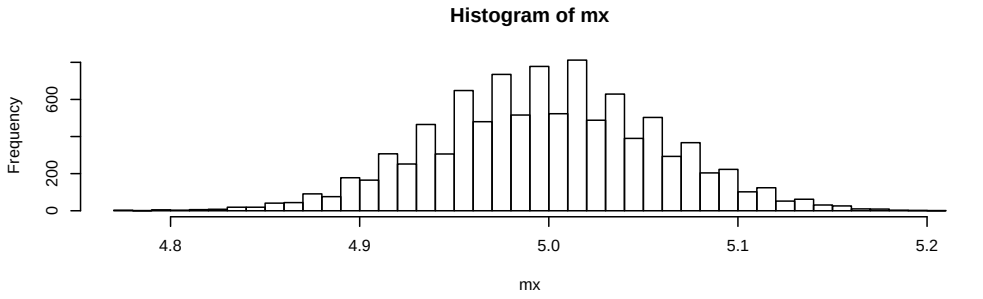


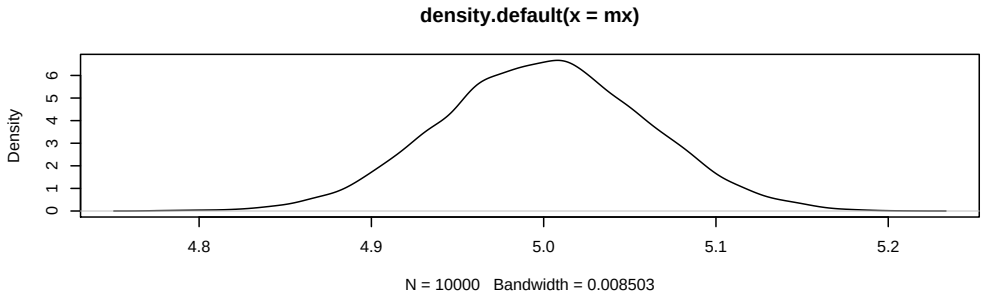



```
par(mfrow=c(3,1))
x = c(2,4,6)
p = c(0.1,0.3,0.6)
mx = 0
for(i in 1:10000){
        samp = sample(x,size=500,replace=TRUE,prob=p)
        mx[i] = mean(samp)
}
boxplot(mx,horizontal=T)
hist(mx,n=50)
plot(density(mx))
```




**3.** (**Multiple Choice**) $X_1, X_2, \ldots, X_{25}$ represents a random sample from a distribution with mean $\mu = 10$ and standard deviation $\sigma = 20$. Indicate which of the following distributions is a good approximation to the distribution of $\bar{X} = \frac{1}{25}\sum_{i=1}^{25} X_i$.

(a) $N(10, 20^2)$

(b) $N(0, 1)$

(c) $N(10, 0.8)$

(d) $N(10, 4)$

(e) $N\left(10, \frac{20^2}{25}\right)$

***Solution:*** The sample sum and also the sample mean are approximately normal here, simply because the sample size is "large enough" (although this is slighly rough: it really depends on how

"non-normal" the true distribution of the $X_i$'s is, but for the purposes of this question we assume 25 is "large enough").

The appropriate normal distribution is simply that with the same expectation and variance as $\bar{X}$. In this case

$$E(\bar{X}) = \mu = 10\,;$$
$$Var(\bar{X}) = \frac{\sigma^2}{n} = \frac{20^2}{25}\,.$$

So the correct answer is (e).

**4.** $X$ is binomial with $n = 100$ and $p = 0.4$ Which of the following normal distributions is a good approximation to the distribution of $X$?

(a) $N(100, 0.4)$

(b) $N(0, 1)$

(c) $N(40, 0.16)$

(d) $N(40, 24)$

(e) $N(40, 0.24)$

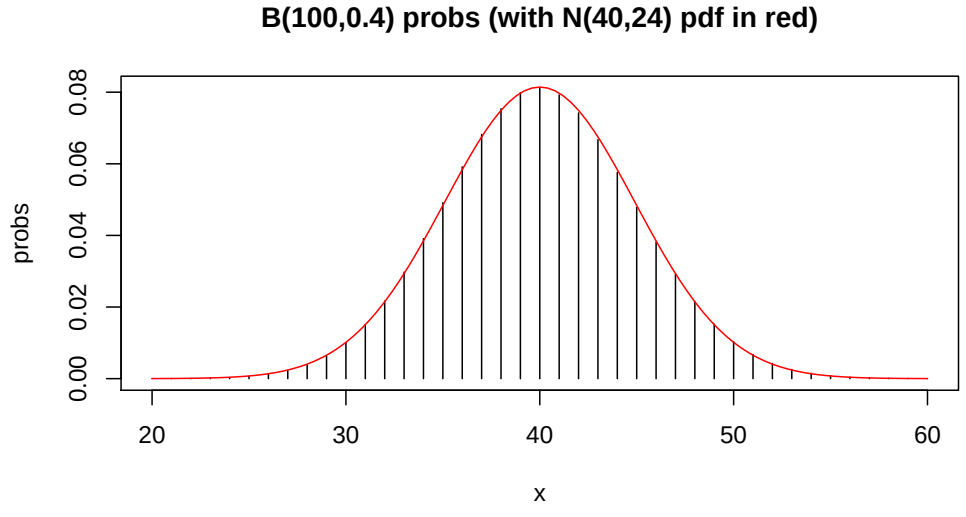
***Solution:*** Since a binomial random variable can be represented as a sum of independent (and identically distributed) random variables, the same principle used in the previous question applies here. We only need to determine the expectation and variance.

As $X \sim B(100, 0.4)$, $E(X) = np = 40$ and $Var(X) = np(1-p) = 24$ the correct answer is (d) *so long as $n$ is "large enough".*

In the binomial setting, deciding when $n$ is "large enough" can be reduced to asking the question: is the binomial distribution being considered here (reasonably) symmetric? Binomial distributions are reasonably symmetric so long as not too much of the "mass" is near the endpoints 0 and $n$. This translates into the "rule of thumb" that both $np$ and $n(1-p)$ are at least 5.

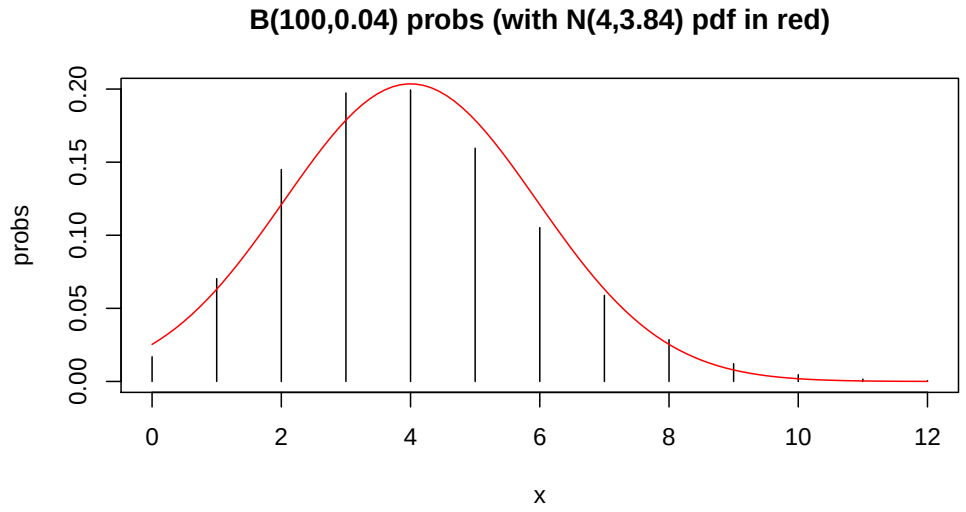
In this case, this is easily satisfied since $5 < 40 = np < n(1-p)$; the corresponding binomial distribution is indeed symmetric, and the approximating normal PDF follows the "heights" very closely:

```
x=20:60
probs=dbinom(x,100,0.4)
plot(x,probs,type="h",
     main="B(100,0.4) probs (with N(40,24) pdf in red)")
curve(dnorm(x,m=40,sd=sqrt(24)),col="red",add=T)
```



However, if we make $p$ very close to 0 or 1 so that the "rule of thumb" is violated, the binomial distribution becomes asymmetric and the normal approximation is not so good:

```
x=0:12
probs=dbinom(x,100,0.04)
plot(x,probs,type="h",
     main="B(100,0.04) probs (with N(4,3.84) pdf in red)")
curve(dnorm(x,m=4,sd=sqrt(4*0.96)),col="red",add=T)
```




**5.** If $X \sim B(64, 0.5)$, the approximating normal variable $Y$ is $N(32, 4^2)$. Using the correction for continuity, $P(32 < X < 36)$ is approximated by

(a) $P(32 < Y < 36)$

(b) $P(32.5 < Y < 35.5)$

(c) $P(32.5 < Y < 36.5)$

(d) $P(31.5 < Y < 36.5)$

(e) $P(31.5 < Y < 35.5)$

***Solution:*** The correct answer is (b). The chief difficulty is working out whether to "add $\frac{1}{2}$" or "subtract $\frac{1}{2}$" from each endpoint.

A good strategy is to try adding or subtracting $\frac{1}{2}$ **before replacing $X$ with $Y$** and see if the binomial probability is unchanged (the proper continuity correction has this property).

Note that **since $X$ only takes integer values**, we can rewrite the binomial probability as

$$P(32 < X < 36) = P(X = 33) + P(X = 34) + P(X = 35)\,;$$

it includes 33 (but not 32), it includes 35 (but not 36). We may thus express it in various other ways:

$$P(32 < X < 36) = P(32.5 < X < 35.5) = P(32.5 \le X \le 35.5) = P(33 \le X \le 35) \qquad (1)$$

since the values 32.5 and 35.5 have probability zero under the (discrete) binomial distribution of $X$.

However, if we replace $X$ in each of these expressions with the (continuous, normal) random variable $Y$ we have

$$P(32 < Y < 36) \neq P(32.5 < Y < 35.5) = P(32.5 \le Y \le 35.5) \neq P(33 \le Y \le 35)\,.$$

So the four binomial probabilities in (1) *which are all equal*, give 3 different normal probabilities. when $X$ is replaced with $Y$. The two versions in the middle are of course correct, both corresponding to (b) (remember, the two values 32.5 and 35.5 also have probability zero under the continuous normal distribution of $Y$).

**6.** (a) Use R to find $c$ if

(i) $P(t_{12} > c) = 0.01$

(ii) $P(t_5 \leq c) = 0.95$

(iii) $P(|t_{25}| > c) = 0.05$.

***Solution:***

```
qt(0.99,12)
```

```
[1] 2.680998
```

```
qt(0.95,5)
```

```
[1] 2.015048
```

```
qt(0.975,25)
```

```
[1] 2.059539
```

(b) Use $R$ to find

(i) $P(t_{11} > 2.5)$

(ii) $P(|t_{15}| > 2.2)$.

***Solution:***

```
1-pt(2.5,11)
```

```
[1] 0.01475319
```

```
2*(1-pt(2.2,15))
```

```
[1] 0.04389558
```

**7.** (Illustration of the Central Limit Theorem using R)

(a) Use a loop (as below) to generate 1000 samples of size 25 from an exponential distribution with parameter $\lambda = 1$. For each sample compute the observed sample mean and store it in a vector called `sample.mean.obs` (as below).

```
sample.mean.obs=0                    # a place to store the result

for (i in 1:1000){                   # begin for-loop

  x.obs=rexp(25,1)                   # generate 25 "random"
                                     # exp(1) values

  sample.mean.obs[i]=mean(x.obs)     # compute the sample mean
                                     # for the ith sample

}                                    # end for-loop
```

Type `par(mfrow=c(2,2))` to prepare the graphics window for 4 plots. Note that at the end of the loop the object `x.obs` contains the 1000th (simulated) exponential sample.

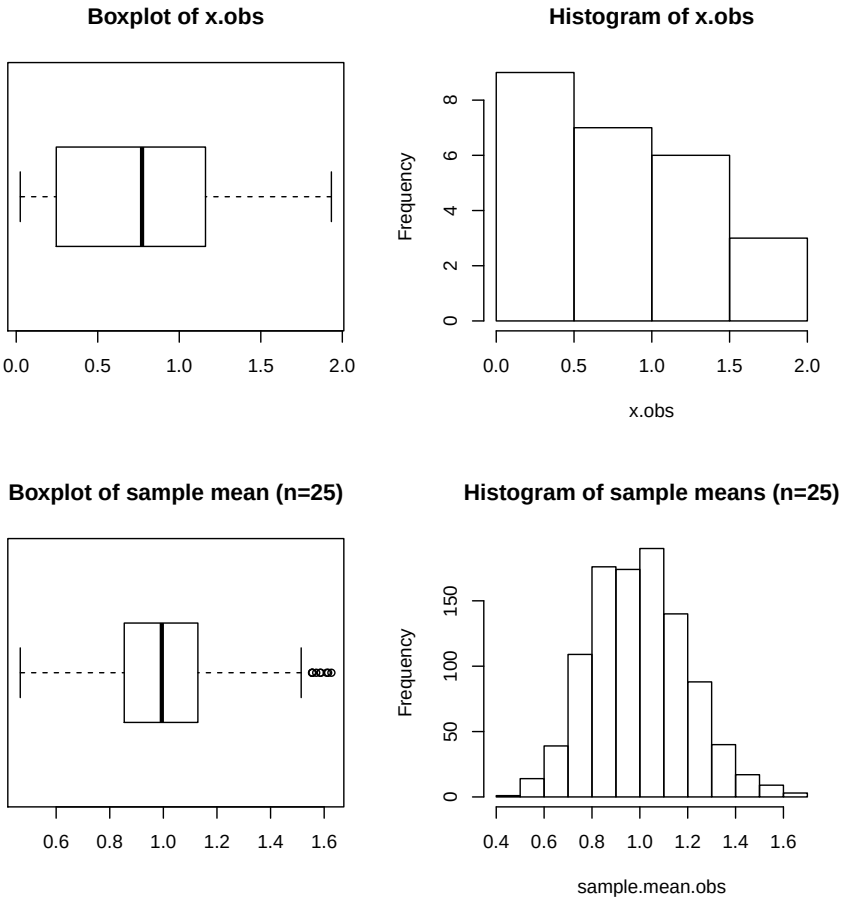
Add the appropriate code to obtain a boxplot and histogram of the final (simulated) sample `x.obs` and then also the vector of sample means `sample.mean.obs` (you can add a useful heading by using a command like

```
boxplot(sample.mean.obs,horizontal=T,
        main="Boxplot of sample means (n=25)")
```

Comment on the shapes of both distributions. Repeat this question with 1000 samples of size 250.

***Solution:***

```
par(mfrow=c(2,2))
sample.mean.obs=0                    # a place to store the result

for (i in 1:1000){                   # begin for-loop

  x.obs=rexp(25,1)                   # generate 25 "random"
                                     # exp(1) values

  sample.mean.obs[i]=mean(x.obs)     # compute the sample mean
                                     # for the ith sample

}                                    # end for-loop

boxplot(x.obs,horizontal=T,main="Boxplot of x.obs")
hist(x.obs)
boxplot(sample.mean.obs,horizontal=T,
        main="Boxplot of sample mean (n=25)")
hist(sample.mean.obs,main="Histogram of sample means (n=25)")
```

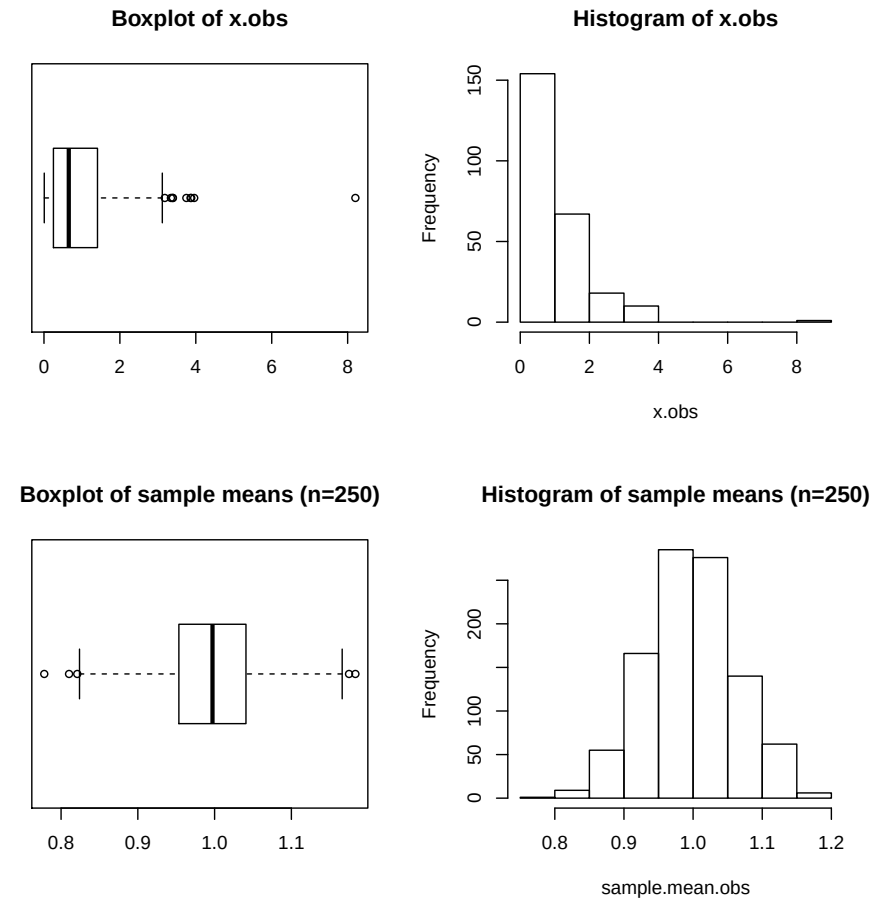



Clearly, the sample means are more "normal-like" than the exponential observations. The single sample `x.obs` is highly right-skewed, while the sample means are *more symmetric and bell-shaped*; however they are not perfectly so: there is still visible skewness in boxplot and histogram of the sample means.

This is also a good example to highlight the limitations of the Central Limit Theorem rule of thumb that says "sample size $n = 25$ is large enough"; clearly here it is not, mainly because the distribution of the underlying observations being added is very non-normal.

Now, the sample size is increased from 25 to 250...

```
par(mfrow=c(2,2))
for (i in 1:1000){                  # begin for-loop

  x.obs=rexp(250,1)                 # generate 1000 "random"
                                    # exp(1) values

  sample.mean.obs[i]=mean(x.obs)    # compute the sample mean
                                    # for the ith sample

}                                   # end for-loop

boxplot(x.obs,horizontal=T,main="Boxplot of x.obs")
hist(x.obs)
boxplot(sample.mean.obs,horizontal=T,
        main="Boxplot of sample means (n=250)")
hist(sample.mean.obs,main="Histogram of sample means (n=250)")
```




...and we see that the boxplot and histogram of the sample means is much more normal-like (mainly, more symmetric) in this case compared with the $n = 25$ case above.

Note also that (of course) they are less spread-out (since the variance/standard deviation of the sample mean is less if the sample size is larger).