The Central Limit Theorem, Sampling Error, and Confidence Intervals

Kris Gunsalus

Fall 2019

Review: Summary statistics

Populations

Location / central tendency: mean, average

$$E(X) = \mu = \frac{1}{N}\sum x_i$$

Scale: variance (standard deviation)

$$Var(X) = \sigma^2 = \frac{1}{N}\sum (x_i - \mu)^2$$

The standard deviation is a measure of how far away from the average most of the individuals in the population are.

Sample estimators: mean, variance (standard deviation)

$$\bar{X} = \frac{\sum(x)}{n}, \quad s^2 = \frac{\sum(x_i - \bar{X})^2}{n-1}$$

Standard normal: a normal distribution with mean 0 and sd = 1

$$Z = \frac{X - \mu}{\sigma}$$

Any outcome from a normal distribution can be standardized as a z-score:

$$z_i = \frac{x_i - \mu}{\sigma}$$

Estimates of scale: IQR, SD

Rule of thumb:

- The central 50% of the data are within the IQR.
- Around 99% of the data are within ~2.5 sd (IQR +/- 1.5*IQR)
- Around 2/3 of the data are within 1 sd of the mean.
- Around 95% of the data are within 2 sd
- Around 99.7% of the data are within 3 sd

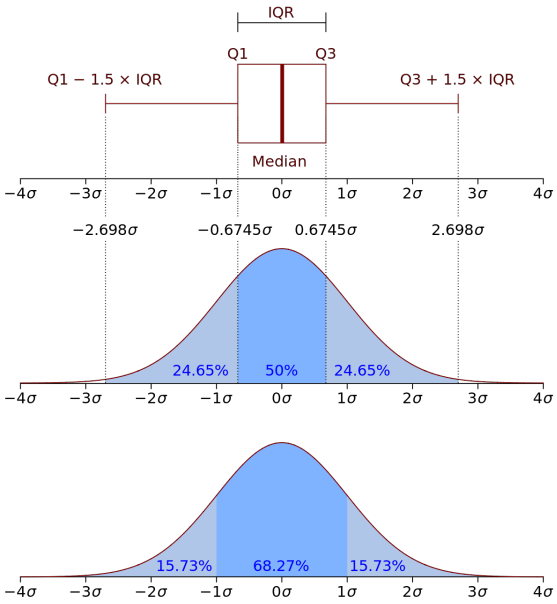


Figure 1: IQR for a normal distribution (*from Wikipedia*)

What is the Central Limit Theorem?

The CLT is one of the most fundamental concepts in statistics. It says that if you take a bunch of random samples from a large population, then the more observations you make per sample (the greater the sample size), the more likely you are to get an estimate of the mean that accurately represents the population.

More specifically, the CLT says that:

1. The **variation** in the distribution of sample means will be inversely proportional to the sample size N.
2. Moreover – and this is key – the sample means will follow a **normal distribution** centered around the population mean.

⇒ As it turns out, this distribution has *maximum entropy*. This means that any individual sample gives no additional information about any of the other samples, i.e. they are uncorrelated. Anytime the measurements you use to estimate a random variable are *independent and identically distributed (iid)*, repeated estimates of your sample statistic will show a normal distribution.

⇒ More generally, any sample statisic will follow the same general pattern: as the sample size increases, your estimator will converge on the true population parameter. Your *confidence* in that value will increase because the variation among the sample estimates will get smaller.

Let's see if we can show this empirically.

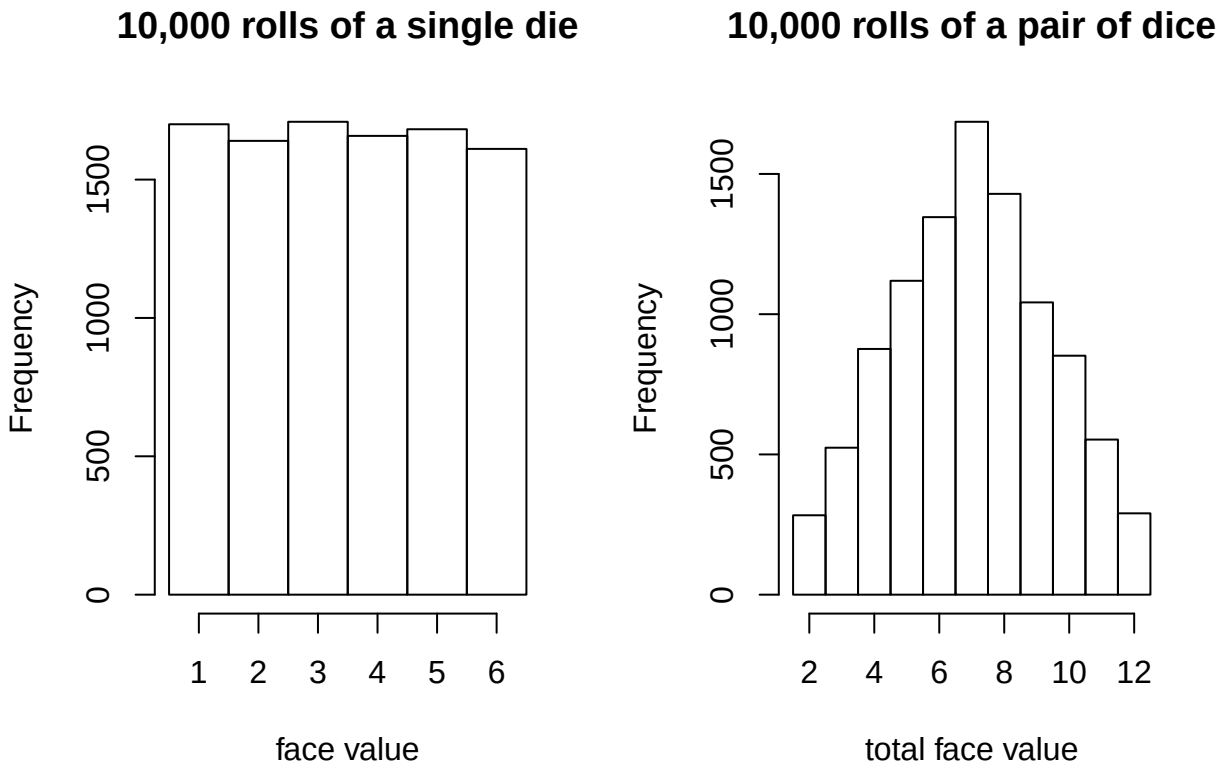
I. Sampling from a discrete uniform distribution

Here, we sample from a flat distribution where the probability of any outcome is the same. This could be the chance of winning a lottery ticket, the month of the year in which anyone in the Biology Department was born, or the amount of time you will wait for a subway train that arrives (punctually!) every 15 minutes, if you enter the subway at any given moment. The lesson is the same regardless of the specific uniform distribution we sample from.

Let's start by rolling a single six-sided die. Assuming the die is fair, then the chance of landing on any side is the same. This is an example of a *discrete* uniform distribution.

Instead, if we roll 2 dice 10,000 times, we see a different pattern. Let's plot these two scenarios together:

```
# roll a single die 10,000 times and record the face value
one_die <- sample(1:6,10000,replace=TRUE)

# roll two dice 10,000 times and record the sum of the face values
pair_of_dice <- sample(1:6,10000,replace=TRUE) + sample(1:6,10000,replace=TRUE)

# plot the distributions using histograms
par(mfrow=c(1,2))  # side by side plots

hist(one_die,
     breaks=seq(.5,6.5,1),
     xlab="face value",
     main="10,000 rolls of a single die"
)

hist(pair_of_dice,
     breaks = seq(1.5,12.5),
     xlab="total face value",
     main = "10,000 rolls of a pair of dice")
```



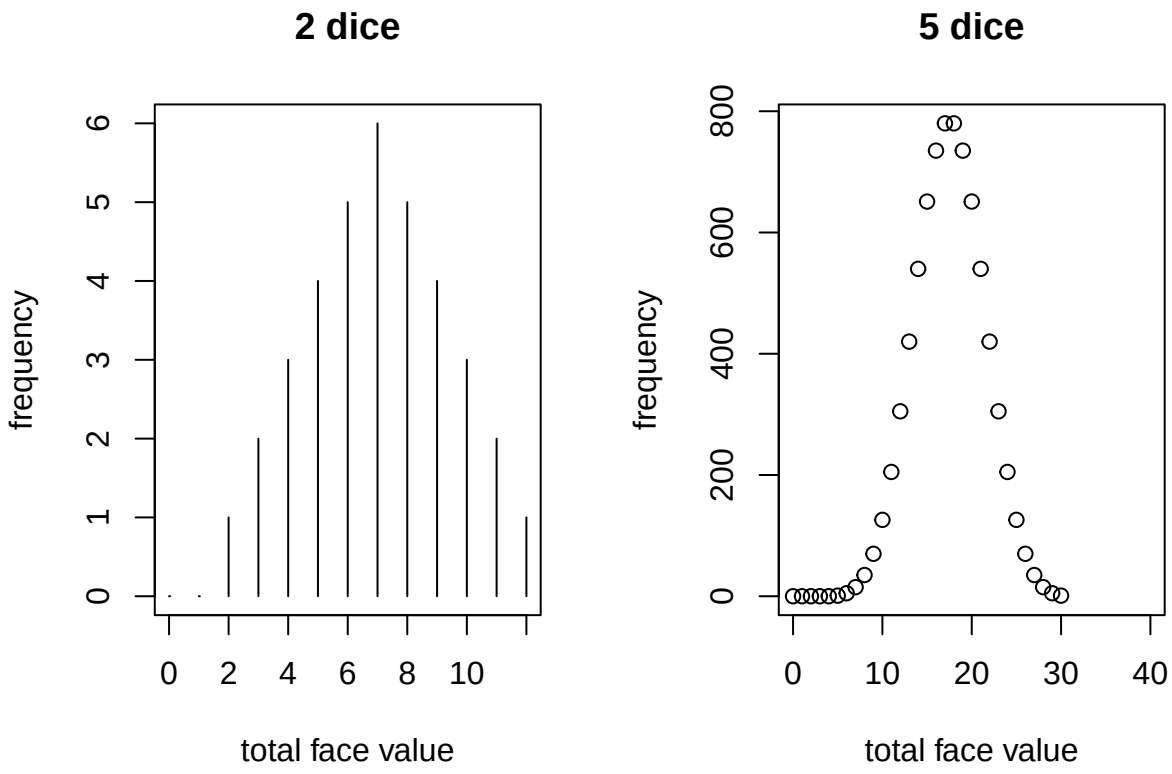
Exercise

⇒ ***Why does this happen?***

First, let's compute how many ways are there to roll different total face values when throwing a pair of dice. Below are two different solutions that give the same output for two dice. One is more general.

```
## A short brute force function to compute number of ways to get
## a total face value for 2 fair dice, ranging from 0 to 12
ncomb.2dice = function (value) {
  # initialize an empty vector (i1 => 0, i2 => 1, ... i13 => 12)
  combos_2dice = rep(0,13)
  for (i in 1:6) {  # compute all the combinations
    for (j in 1:6) {
      index = i+j+1
      combos_2dice[index] = combos_2dice[index] + 1
    }
  }
  return(combos_2dice[value+1]) # return the value(s) of interest
}
```

```
## A more elegant general solution by Chris for any number of dice
ncomb.ndice <- function (value, n_dice=2) {
  outcomes <- table(rowSums(do.call(expand.grid,
                                    rep(list(1:6), n_dice))))
  all.outcomes <- rep(0, (n_dice*6) + 1)
  all.outcomes[as.numeric(names(outcomes)) + 1] <- as.numeric(outcomes)
  return(all.outcomes[value + 1])
}

## test the result for a bunch of different outcomes
ncomb.ndice(0)
## [1] 0
ncomb.ndice(3)
## [1] 2
ncomb.ndice(7)
## [1] 6
ncomb.ndice(12)
## [1] 1
ncomb.ndice(0:12) # surprise! this works for a vector of all values!
##  [1] 0 0 1 2 3 4 5 6 5 4 3 2 1

# and we can plot these
par(mfrow=c(1,2))
plot(0:12,ncomb.ndice(0:12),type="h",main = "2 dice",
     xlab="total face value", ylab="frequency")
plot(0:40, ncomb.ndice(0:40,n_dice=5), main = "5 dice",
     xlab="total face value", ylab="frequency")
```



What kind of sampling distribution is this? Explain your answer.

This is actually an example of a binomial distribution.

What other distributions can approximate this distribution, and under what conditions? Explain your answer.

As you saw in the marbles example, if the number of samples is large, then a (continuous) normal distribution is often a good approximation for the (discrete) binomial distribution.

This does not hold when p is low (because sometimes the normal approximation will give negative probabilities, which is not possible – as we saw with the marbles example). In that case, which distribution is a better approximation of the binomial?

II. Sampling from a continuous uniform distribution

The problem can be generalized to a **continuous** uniform distribution, in which all possible values in an interval are represented with equal probability.

To get a feel for this, let's pick a bunch of numbers at random between zero and one. (This seems like a pretty boring thing to do, but we will see later why it's relevant to a lot of estimation problems.)

Let's sample randomly from a uniform distribution:

```
size <- 100  # the sample size

# draw a 'sample' of 100 observations from the
# the uniform distribution (range = 0,1)
sample_unif <- runif(size, min=0, max=1)
```

Let's look at our results:

```r
head(sample_unif) # just look at the first few values
```

```
## [1] 0.7578026 0.1317306 0.0992167 0.5602503 0.6424118 0.2779443
```

We can plot the distribution of our samples with a histogram. Let's write a function to do all this for us in one command:

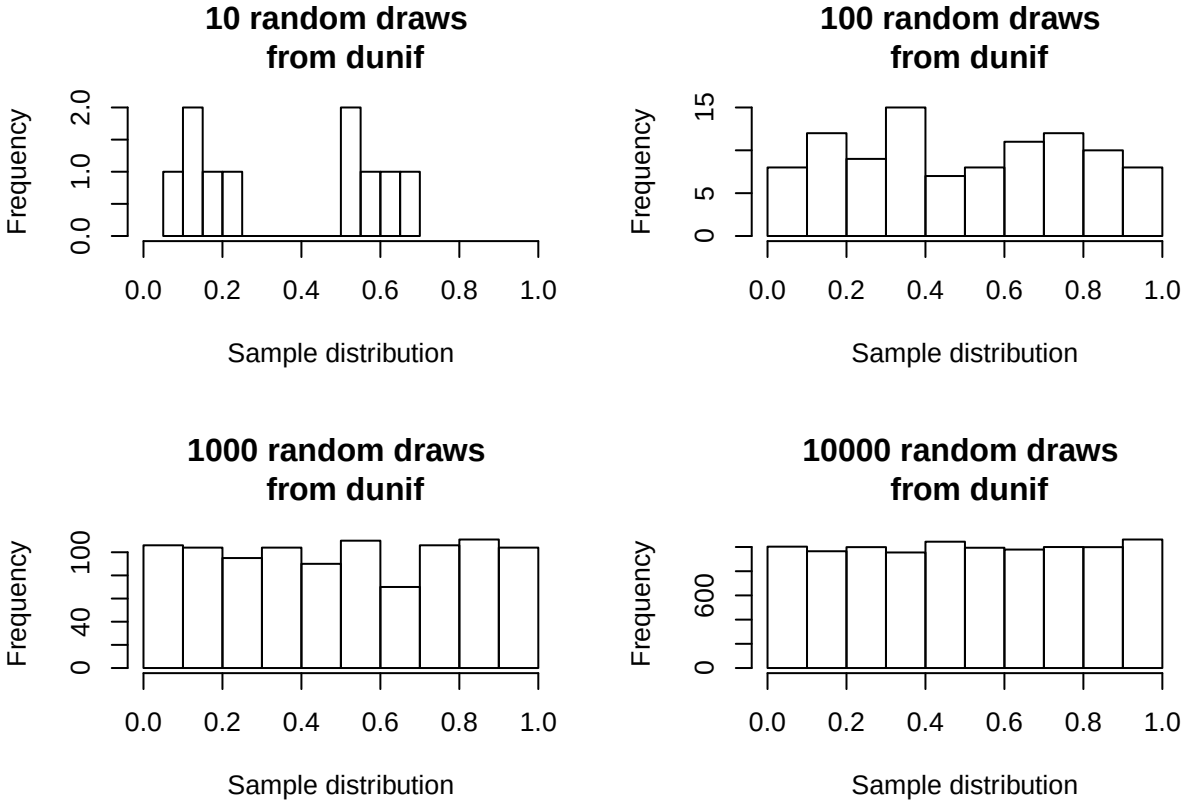
```r
# draw a histogram for a sample of size 'size' between zero and one
hist_unif <- function(size) {
  hist(runif(size,0,1), # an anonymous (unnamed) sampling function
     xlim = c(0,1), # the limits of the range of values to plot
     xlab = "Sample distribution",
     breaks = 10,
     main = paste(size,"random draws \n from dunif",sep=" "))
  }
```

Exercise

Now we can use our function to plot random samples of any size.

⇒ ***What happens if we do this 10, 100, 1000, or 10,000 times? Try it out yourself.***

```r
# draw a histogram of random samples from the uniform distribution
# across different sample sizes
par(mfrow=c(2,2))
hist_unif(size=10)
hist_unif(size=100)
hist_unif(size=1000)
hist_unif(size=10000)
```



⇒ ***How does the sample distribution change with sample size?***

The sample distribution starts to converge toward the theoretical distribution as the sample size increases.

III. Sample estimate of the population mean

So far, we've drawn *individual samples* from a uniform continuous distribution (akin to rolling a single die).

Now let's look at the **mean of multiple samples**. This will tell us, for example, how long on average we expect to wait for our subway train on the way to work (assuming there are no unexpected delays).

Here, we will look at the same uniform distribution as above. Keep in mind that the true mean of a continuous uniform distribution is the range divided by two (here, that would be 0.5).

Let's calculate the mean of 10 observations:

```
# mean of 10 draws from a continuous uniform distribution
# ranging from 0 to 1
print("Sample mean for one sample of 10 observations:\n")
```

```
## [1] "Sample mean for one sample of 10 observations:\n"
```

```
mean(runif(10, min=0, max=1))
```

```
## [1] 0.3446865
```

⇒ ***How close is that to the true population average? What happens if we repeat this a whole bunch of times?***

Try this out yourself:

```
# find the mean of 10 draws from the population,
# and repeat this 10 times
print("Sample means for each of 10 samples of size 10:\n")
## [1] "Sample means for each of 10 samples of size 10:\n"
replicate( 10, mean( runif(10, min=0, max=1) ) )
##  [1] 0.3863881 0.6473126 0.5868513 0.4612966 0.4637817 0.4946255 0.3196612
##  [8] 0.5641761 0.5131749 0.4194958
```

We can run this code a bunch of times and see what happens:

```
# calculate the mean and sd of the mean estimate from drawing 10 items on 10 different occasions
print("Mean and SD of 10 sample means:\n")
```

```
## [1] "Mean and SD of 10 sample means:\n"
```

```
for (i in 1:5) {
  a = replicate( 10, mean( runif(10, min=0, max=1) ) )
  print(paste("Mean:",round(mean(a),3),"SD:",round(sd(a),3)))
}
```

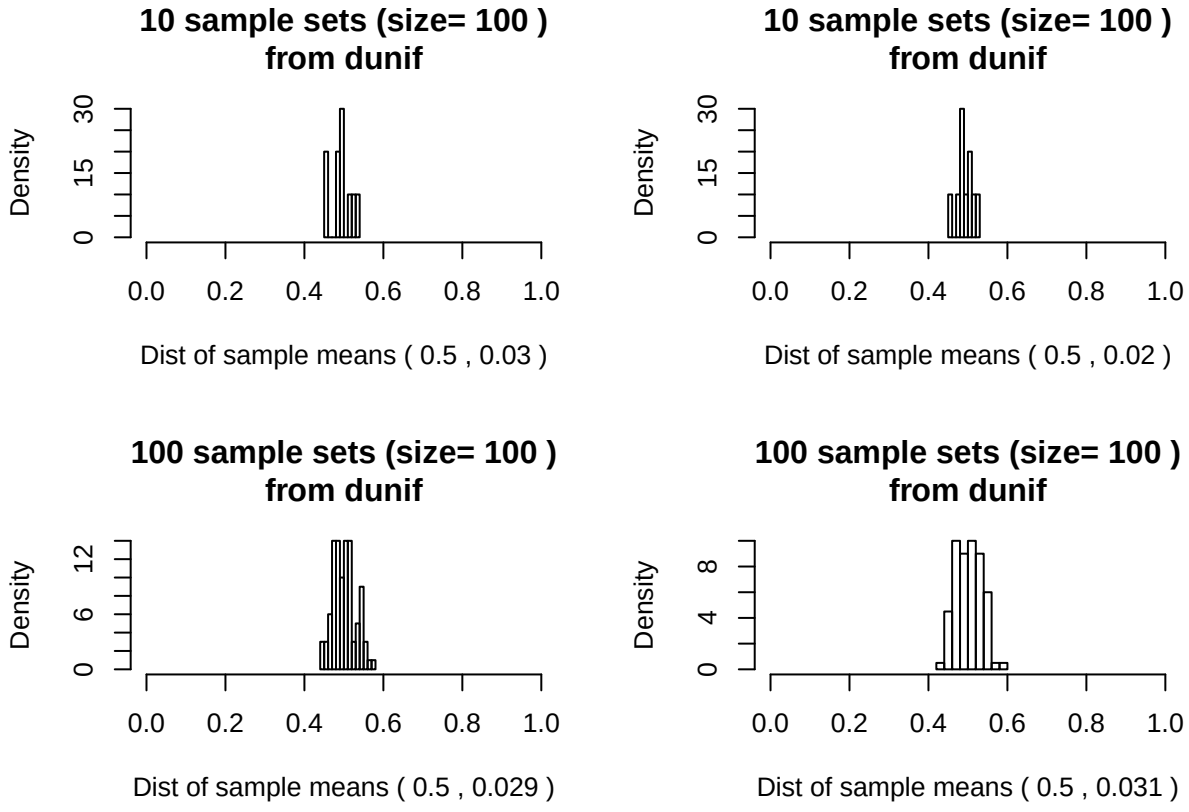
```
## [1] "Mean: 0.498 SD: 0.05"
## [1] "Mean: 0.505 SD: 0.083"
## [1] "Mean: 0.521 SD: 0.071"
## [1] "Mean: 0.478 SD: 0.06"
## [1] "Mean: 0.531 SD: 0.113"
```

Let's create a function to draw a histogram of our samples, which will make it a lot easier to visualize our results.

```
# create a function that takes three parameters:
#   n = number of times to replicate the sampling
#   size = sample size (number of draws) for each replicate
ud.mean.hist <- function(n, size, capture=FALSE) {
  hist( x <- replicate(n, mean(runif(size, min=0, max=1)) ),
        xlim=c(0,1), prob=T, breaks=10,
        # below is a trick to limit the number of significant digits
        # displayed for the mean and SD (2 for 100, 3 for 1000, etc.)
        xlab=paste("Dist of sample means (",
                   signif(mean(x),log10(n)), ",",
                                  signif(sd(x),log10(n)),")"),
        main=paste(n, "sample sets (size=", size,
                   ")\nfrom dunif", sep=" ") )
  # so we can access the individual samples later if we want to
  if (capture) { return(x) }
}
```

Let's see how the results vary if we repeat this exercise a bunch of times:

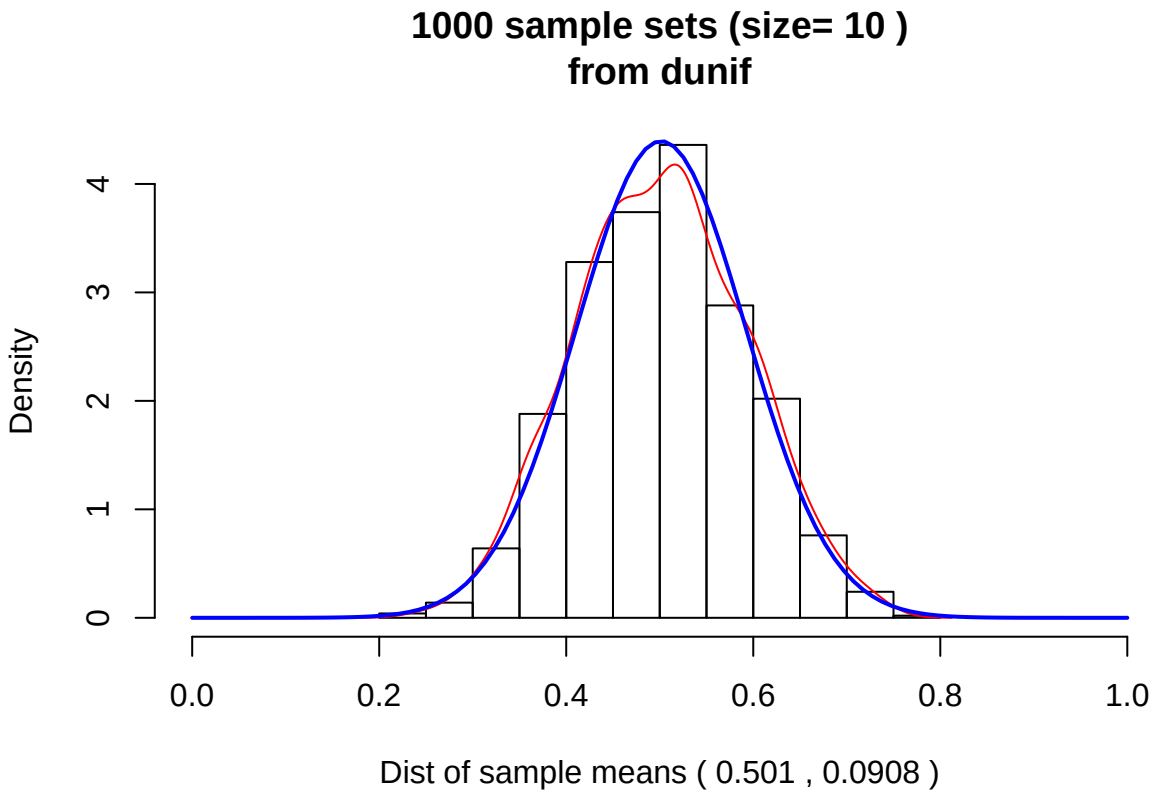
```
# plot the mean of 10 or 100 sets of 10 random samples
# do this independently 4 times and look at the results

par(mfrow=c(2,2))
ud.mean.hist(n=10,size)
ud.mean.hist(n=10,size)
ud.mean.hist(n=100,size)
ud.mean.hist(n=100,size)
```



We can plot the density of our sample along with a normal curve with the same mean and SD on top of the

histogram. Let's try this for n = 1000, size = 10.

```
# make a histogram and capture the original distribution
# 3rd arg can be anything that doesn't evaluate to FALSE
x <- ud.mean.hist(n=1000,size=10,capture=TRUE)

# plot density of the samples
x.density <- density(x)
lines(x.density, col="red")

# compare to normal curve with same mean and s.d.
xfit<-seq(0,1,length=100)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
lines(xfit, yfit, col="blue", lwd=2)
```



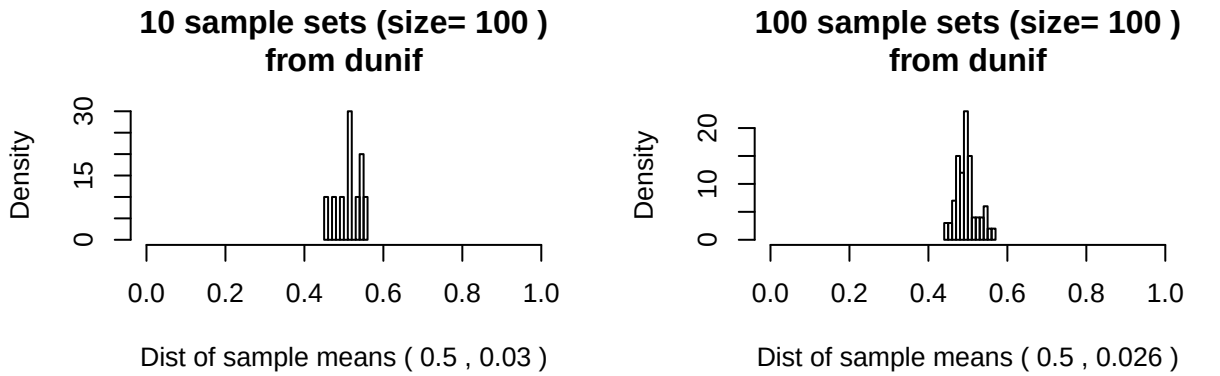
Exercise

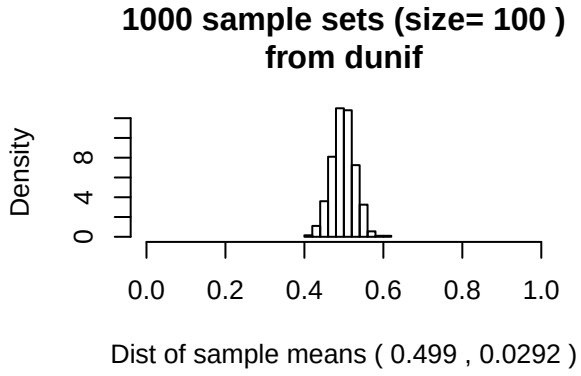
Experiment with how the plots change the n and *size* parameters vary. Try different combinations of n and size across several orders of magnitude (e.g. 10, 100, 1000, and 10000). First, hold the sample size constant and vary n. Then, holding n constant, repeat the exercise for different sample sizes. (Note: 10k x 10k will take a lot longer to compute.)

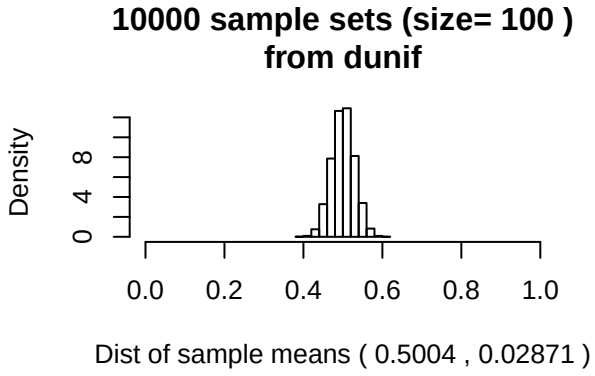
You may do this by brute force or using a loop . . .

```
## try four different options for one variable, holding the other constant
## Option 2: using a loop

c = 100  # hold one variable constant
v = c(10,100,1000,10000)  # vary the other one
```

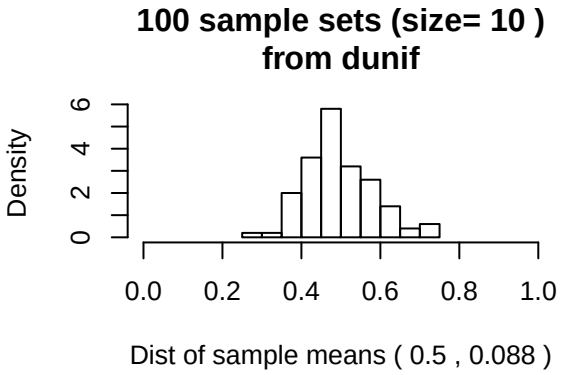
```
par(mfrow=c(2,2))
for (i in v) {
  ud.mean.hist(i,c)
}
```

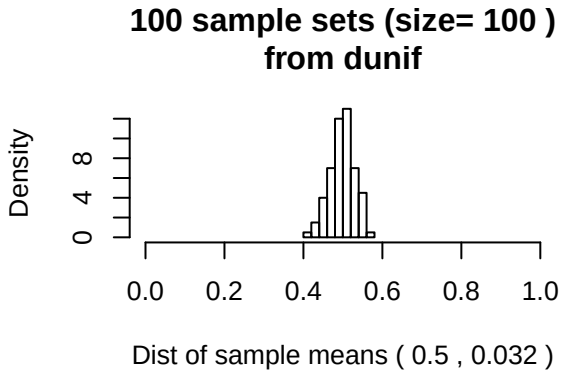


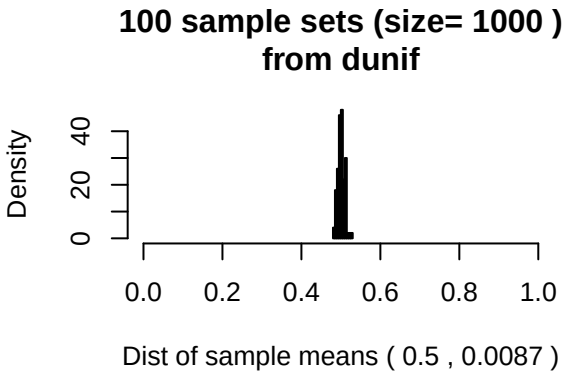


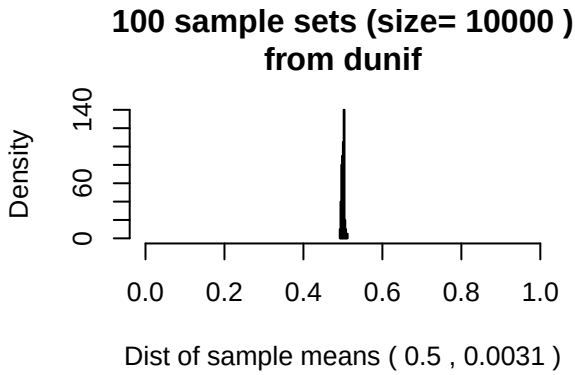



```
par(mfrow=c(2,2))
for (i in v) {
  ud.mean.hist(c,i)
}
```









⇒*How did the distributions change when you increased the number of sample sets, but kept the sample size the same?*

Increasing the number of samples sets doesn't make that much difference.

⇒*How did the distributions change when you increased the sample size?*

Increasing the sample size narrows the distributions considerably.

⇒*What does this tell you about the importance of* ***sample size*** *vs. the* ***number of samples*** *to your estimate of the population mean?*

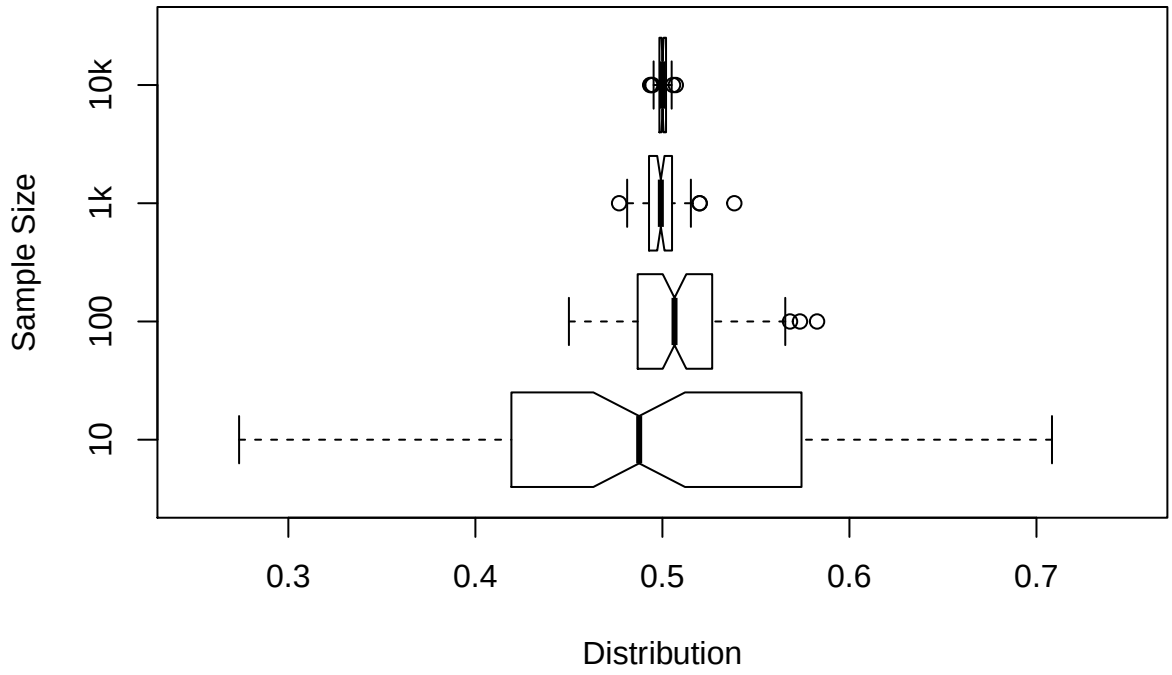
Sample size has much more of an effect on the estimate of the sample mean.

Box plots

We can also use **box plots** to summarize the distributions we generated above, which make it a little easier to compare them visually. Below we create a boxplot showing four distributions of the sample means for 100 samples, with varying sample sizes of 10, 100, 1000, and 10000.

```
# a convenient function that returns a list of means for "n" samples of size "size"
resample_mean_unif <- function (n, size) {
  replicate( n, mean( runif(size, min=0, max=1) ) )
}

# resample a constant number of times for different sample sizes
a <- resample_mean_unif(100,10)
b <- resample_mean_unif(100,100)
c <- resample_mean_unif(100,1000)
d <- resample_mean_unif(100,10000)
```

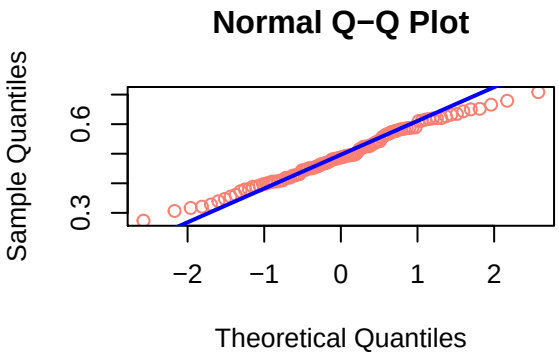
```r
boxplot(a,b,c,d,
        horizontal=TRUE, ylim=c(0.25,0.75), range=1, notch=T,
        names=c("10","100","1k","10k"),
        xlab = "Distribution",
        ylab = "Sample Size"
        )
```

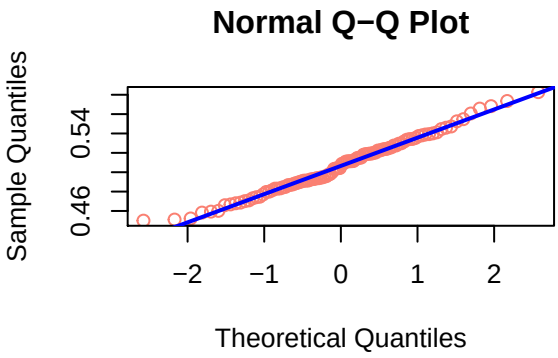


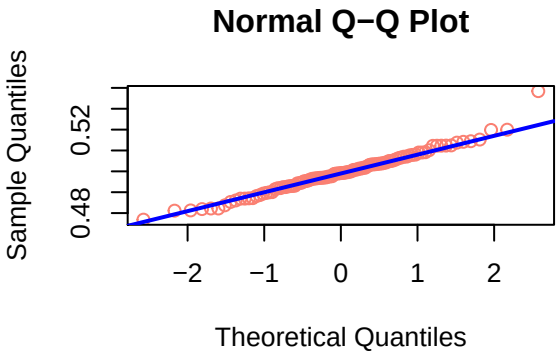
QQ Plots

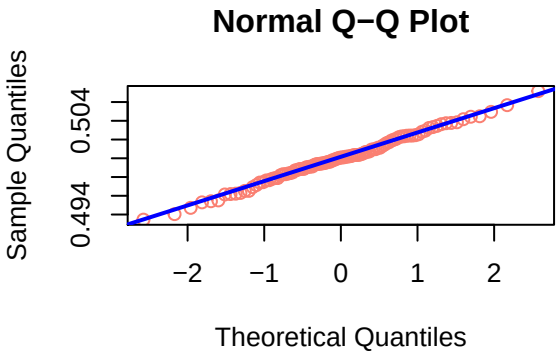
We can also use **quantile-quantile (Q-Q) plots** to visualize the similarity between the actual distributons and the theoretical (normal) distributions of the sample means. Here we will create QQ plots for the same sample sets shown in the boxplot above.

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

qqnorm(a, col = "salmon", main = "Normal Q-Q Plot")
qqline(a, col = "blue", lwd = 2)

qqnorm(b, col = "salmon", main = "Normal Q-Q Plot")
qqline(b, col = "blue", lwd = 2)

qqnorm(c, col = "salmon", main = "Normal Q-Q Plot")
qqline(c, col = "blue", lwd = 2)

qqnorm(d, col = "salmon", main = "Normal Q-Q Plot")
qqline(d, col = "blue", lwd = 2)
```







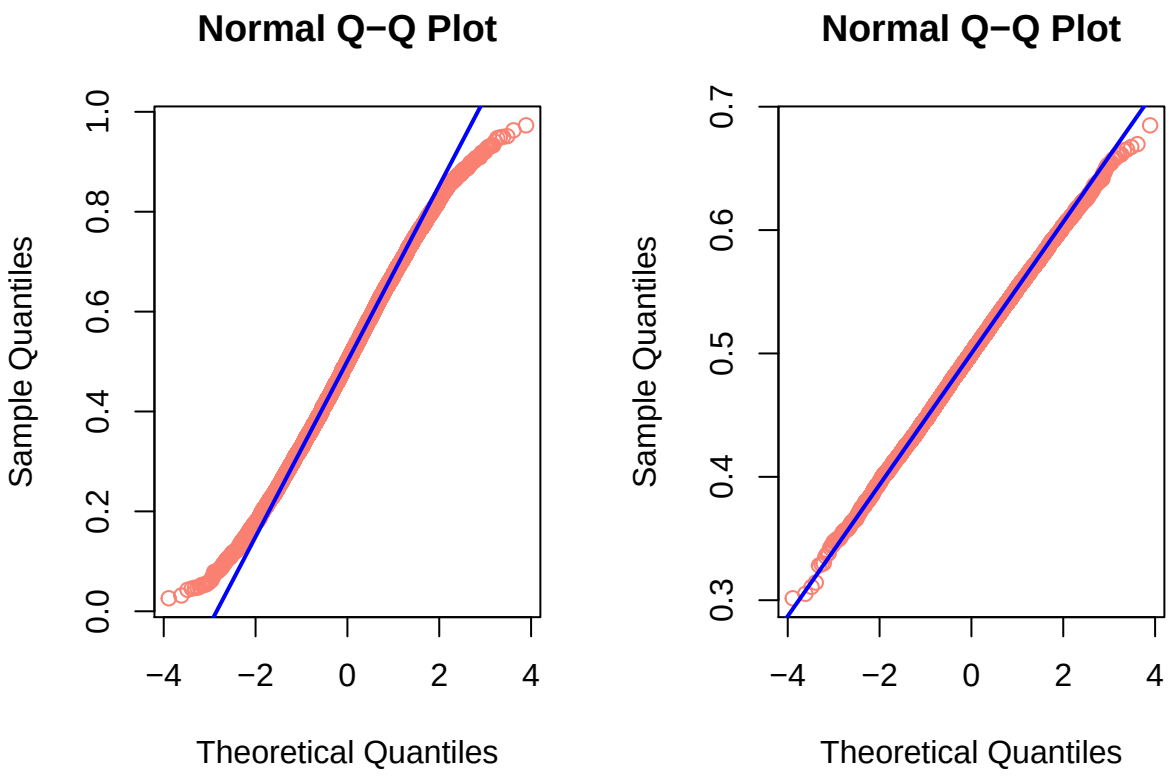


⇒ ***How do the distributions of the sample means look? Are they relatively normal? Where do you see the greatest variation from normality, and why?***

- The observed sample means look to be pretty normally distributed.
- For all of them, there is *greater variation at the tails, which contain less density.*
- However, *note the difference in scales on the Y axis.*

When the number of samples is too low, the normal approximation does not hold anymore:

```
a = resample_mean_unif(10000,3)  # really small
b = resample_mean_unif(10000,30) # bigger

par(mfrow=c(1,2))
qqnorm(a, col = "salmon", main = "Normal Q-Q Plot")
qqline(a, col = "blue", lwd = 2)

qqnorm(b, col = "salmon", main = "Normal Q-Q Plot")
qqline(b, col = "blue", lwd = 2)
```



IV. Variation of the Sample Mean

In Part III, we saw that the **mean of the sample distributions** tends toward (converges on) the **population mean** as the **sample size increases**. We also saw that **the variation** in the mean is **inversely proportional to the sample size**. In other words:

- It doesn't really matter how many times you perform the resampling – for the same sample size, the shape of the histogram stays about the same no matter how many times you run the experiment. *[Except when the number of replicates is really small, since we have so few measurements; we will return to this issue later.]*
- In contrast, increasing the sample *size* dramatically changes the width of the histogram. If we have few samples per replicate, the histogram is wide, and with many samples per replicate, it is very narrow.

Sampling distribution of the sample mean, $\bar{X}$

We can actually quantify the statements above. With sufficiently large N, the *sampling distribution of the sample mean* is **approximately normal**, with mean μ and variance σ^2/N.

The sample mean is now our random variable! It is denoted $\bar{X}$, so we can write:

$$\bar{X} \sim \mathcal{N}(\mu, \frac{\sigma^2}{N})$$

In other words: if you sample repeatedly from a population, then the distribution of the sample means will look approximately normal, and the variation of your estimates will shrink as the sample size gets bigger.

Standard Error of the Mean (SEM)

As we saw above, for a sample of size N, the **variance** of the expected value of the sample mean, $\bar{X}$, is the population variance σ^2 divided by the number of samples N:

$$Var(\bar{X}) = \frac{\sigma^2}{N}$$

The standard deviation of a sample statistic is called its **standard error**. For the sample mean, the standard error is called the **standard error of the mean (SEM)**.

The SEM gives us an idea of how far away from the true mean our sample mean is likely to be, and it is defined as the square root of $Var(\bar{X})$:

$$SEM = \sqrt{\frac{\sigma^2}{N}} = \frac{\sigma}{\sqrt{N}}$$

$\Rightarrow$ From the above equation, we can see that **the SEM of the sampling distribution decreases as the square root of the sample size.** In the limit, the SEM goes to zero as N goes to infinity, i.e. when the sample includes the entire population: $\lim_{N\to\infty} \frac{\sigma}{\sqrt{N}} = 0$. This is entirely consistent with the sampling results we obtained above.

In practice, we usually do not know the true amount of variation in the population. When the sample size is large (typically $>$ ~30), we can use the SD of our samples, s, as an **approximation** for the population parameter σ. This also allows us to approximate the SEM using the sample standard deviation:

$$SEM = \frac{\sigma}{\sqrt{N}} \approx \frac{s}{\sqrt{N}}$$

Note: Keep in mind that s is a **sample estimate of the variation in the parent distribution** *from which our samples are drawn (here, the uniform distribution), and is not to be confused with the SEM, a measure of the* **variation of the sampling distribution of the sample means**.

Exercise

$\Rightarrow$ *Verify that these statements are true when we take samples of different sizes from the uniform distribution.* The variance of a uniform distribution is 1/12 (so the SD is $\sqrt{1/12} \approx 0.289$).*

```
# mean and SEM of different size samples
for ( i in c(10,100,1000,10000) ) {
  sample <- (runif(i, min=0, max=1))
  sigma <- sqrt(1/12)               # population variation sigma
  mean_sample <- mean(sample)       # sample estimate of
                                    # variation in the population
  sem_sigma <- sqrt(1/12)/sqrt(i)   # population parameter
  sem_sd   <- sd(sample)/sqrt(i)    # sample estimate
  cat("Mean for sample size N =",i,":",mean_sample,
      "\n---",
      "\nPopulation standard deviation:",sigma,
      "\nSample standard deviation:",sd(sample),
      "\n---",
      "\nSEM (using population SD, sigma):",sem_sigma,
      "\nSEM (using sample SD, s):",sem_sd,
      "\n\n\n",fill=FALSE)
}
```

```
## Mean for sample size N = 10 : 0.4353939
## ---
## Population standard deviation: 0.2886751
## Sample standard deviation: 0.3209873
## ---
## SEM (using population SD, sigma): 0.09128709
## SEM (using sample SD, s): 0.1015051
##
##
## Mean for sample size N = 100 : 0.5021873
## ---
## Population standard deviation: 0.2886751
## Sample standard deviation: 0.2741087
## ---
## SEM (using population SD, sigma): 0.02886751
## SEM (using sample SD, s): 0.02741087
##
##
## Mean for sample size N = 1000 : 0.5055628
## ---
## Population standard deviation: 0.2886751
## Sample standard deviation: 0.2925596
## ---
## SEM (using population SD, sigma): 0.009128709
## SEM (using sample SD, s): 0.009251547
##
##
## Mean for sample size N = 10000 : 0.4975144
## ---
## Population standard deviation: 0.2886751
## Sample standard deviation: 0.2875568
## ---
## SEM (using population SD, sigma): 0.002886751
## SEM (using sample SD, s): 0.002875568
```

The following table summarizes these results.

Sample Size	Typical Spread	Mean	SEM
1	0-1	NA	~0.3
10	0.2-0.8	~0.5	~0.1
100	0.4-0.6	~0.5	~0.03
1000	0.47-0.53	0.50	~0.01
10000	0.49-0.51	0.50	~0.003

These tests show empirically that we need a **100-fold increase in the sample size** in order to get a **10-fold decrease in the SEM**. So, the SEM indeed decreases as the square root of the sample size.

The SEM computed with the sample SD approximates the SEM computed using the true population SD for the larger sample sizes (n = 100, 1000, 10000).

Standard normal distribution for $\bar{X}$

Now we can describe the sampling distribution of the population mean as a standard normal, or ***Z*-distribution**, with mean = 0 and SD = 1:

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{N}} = \sqrt{N}\frac{\overline{X} - \mu}{\sigma} \approx \sqrt{N}\frac{\overline{X} - \mu}{s}$$

This allows us to express individual outcomes of $\bar{X}$ in terms of a ***z*-score**, i.e. the number of standard deviations they are away from the true population mean by rearranging the above equation:

$$\bar{X} = \mu + z_i\frac{s}{\sqrt{N}}$$

It is important to remember that the ***CLT*** applies to the ***limiting*** situation in which the number of samples is large, so that s approximates σ. In practice, the CLT works well for sample sizes of $N \approx 30$.

As we also saw above, the normal distribution does not hold for smaller samples. For small samples, we must instead use the ***t-distribution***.

V. Confidence Intervals

It is very rare that we know the true population parameters. We can report our uncertainty about how well a random variable estimates a population parameter using a **confidence interval (CI)**. The CI of the mean gives us an idea of how likely it is that the true mean falls within a specified range.

For example, the **95% CI** gives us an interval that we expect will contain the true population mean 95% of the time, given a sample of size N. It is typical to see 90%, 95%, and 99% confidence intervals.

How do we calculate the CI? Since we know our **sample estimate of the population mean** is **normally distributed**, we know that around 95% of the time our sample estimate of the population mean, $\bar{X}$, will be within two standard deviations of the true mean (even though every once in a while it will be rather far off because we are taking random samples). This is because around 95% of any normal distribution falls within 2 SD of the mean:

```
pnorm(2) - pnorm(-2)        # z-score=2 is approximately 95%
```

```
## [1] 0.9544997
```

```
pnorm(1.96) - pnorm(-1.96) # this is closer to 95%
```

```
## [1] 0.9500042
```

We can now find the range of the 95% CI for the population mean, which is approximately ±2 SD from the sample mean, $\bar{X}$:

$$\overline{X} - 2s_x/\sqrt{N} \leq \mu_X \leq \overline{X} + 2s_x/\sqrt{N}$$

Technically, the 95% CI specifies that **95% of random intervals $\overline{X} \pm 1.96s_x/\sqrt{N}$ will contain the true population mean**.

Note that since our sample estimate is a random variable, the edges of the interval are also random. Any particular CI either does or does not contain the true population mean, and around 5% of randomly sampled intervals will not contain the mean.

We can express any $100(1-\alpha)\%$ CI as a function of the central $1-\alpha$ proportion of the standardized normal distribution of $\bar{X}$:

$$(1-\alpha/2)\%\ CI = \bar{X} \pm z_{1-(\alpha/2)} * \frac{\sigma}{\sqrt{N}}$$

where $z_{1-(\alpha/2)}$ is the z-quantile function at probability $1-(\alpha/2)$.

For a 95% CI, $\alpha = 0.05$ and we can use `qnorm(0.975)` to get the limits of the interval, which corresponds to a z-score of 1.96.

$$95\%\ CI \approx \bar{X} \pm 1.96\frac{\sigma}{\sqrt{N}}$$

Exercise

⇒ ***Calculate the 95% CI for 4 samples ranging in size from 100-10,000. What can you learn from these comparisons?***

```
# We will use z=1.96, which is technically more correct than using z=2 for 95% CI.
# Since normal is symmetric, we can add and subtract this to get the CI.
Q <- qnorm(0.975)

# mean, SEM, and CI of our samples
for ( i in c(100,1000,10000) ) {
  sample <- (runif(i, min=0, max=1))
  mean_sample <- mean(sample)
  sem <- sd(sample)/sqrt(i)
  interval <- c(mean_sample - Q*sem, mean_sample + Q*sem)

  cat("Sample size:",i,"\nMean:",mean_sample,
      "\nSEM:",sem,"\nCI:",interval,"\n\n",fill=FALSE)
}
```

```
## Sample size: 100
## Mean: 0.5432015
## SEM: 0.02885196
## CI: 0.4866526 0.5997503
##
## Sample size: 1000
## Mean: 0.4863006
## SEM: 0.009095293
## CI: 0.4684741 0.504127
##
## Sample size: 10000
## Mean: 0.4982337
## SEM: 0.002897095
## CI: 0.4925555 0.5039119
```

Notice how our 95% CI decreases as the sample size increases. If we repeat each of these 100 times, then 95 out of the 100 intervals will contain the true population mean.

Connection between the CI and the p-value

In ***hypothesis testing***, which we will discuss next week, we choose a significance threshold like $p = 0.05$ to reject the null hypothesis that our sample comes from the null distribution. Correspondingly, if the 95% CI does not contain the mean for the null hypothesis, then the p-value for our sample statistic is less than 0.05. We will discuss this in a lot more detail next week.

Reference material

Irizarry: Inference chapter

Aho:

- Section 3.2.2.2 (Normal distribution)
- Section 5.2 (Sampling Distributions)
- 5.2.2 (Sampling Distribution of $\bar{X}$)
- 5.2.2.1 (Central Limit Theorem)
- Section 5.3 (Confidence Intervals)