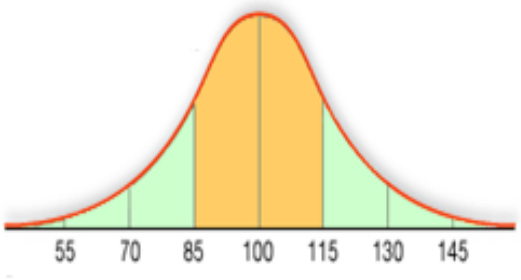
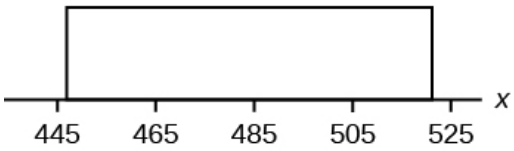
Statistics & Bootstrapping – Part 1

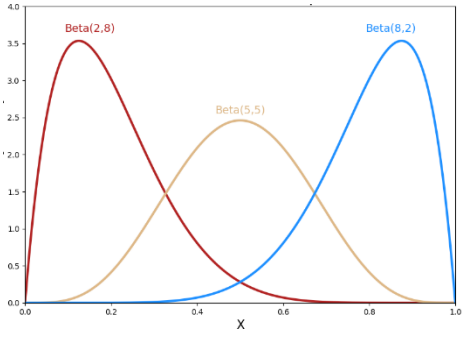
Population: is everyone/everything in a group of interest.

Sample: subset (smaller group) of the population.

Probability Distribution: it's a function that describes the possible values and likelihoods that a specific random variable can take.



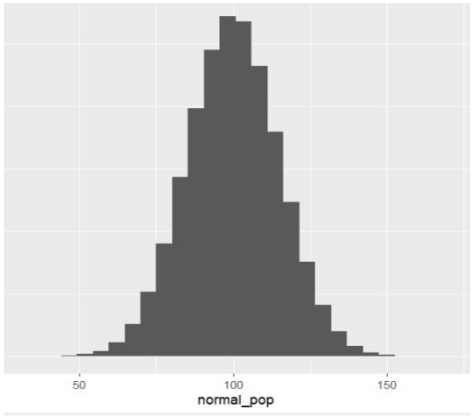




Creating a Population

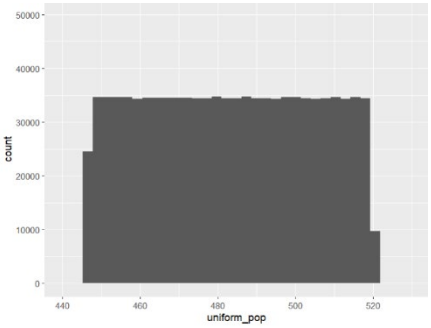
Normal Distribution: Mean 100, Standard deviation 15.

```
N <- 1000000
normal_pop <- rnorm(N, mean = 100, sd = 15)

plot_1 <- data.frame(normal_pop) %>%
  ggplot(aes(normal_pop))+
  geom_histogram()
plot_1
```



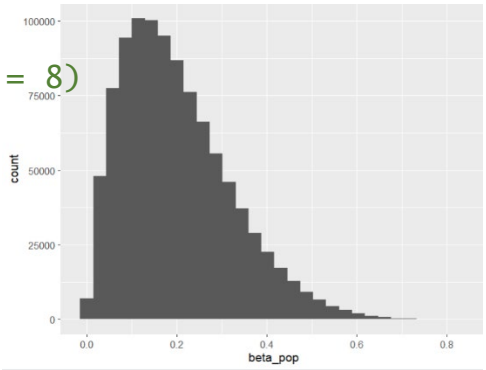
Uniform: min = 446, max = 520

```
N <- 1000000
uniform_pop <- runif(N, min = 446, max = 520)

plot_2 <- data.frame(uniform_pop) %>%
  ggplot(aes(uniform_pop))+
  geom_histogram()
plot_2
```



Beta: shape1 = 2, shape2 = 8.

```
N <- 1000000
beta_pop <- rbeta(N, shape1 = 2, shape2 = 8)

plot_3 <- data.frame(beta_pop) %>%
  ggplot(aes(beta_pop))+
  geom_histogram()
plot_3
```



Creating One Sample

We randomly sample 100 data points from the population of the normal distribution to create our sample then compute the mean.

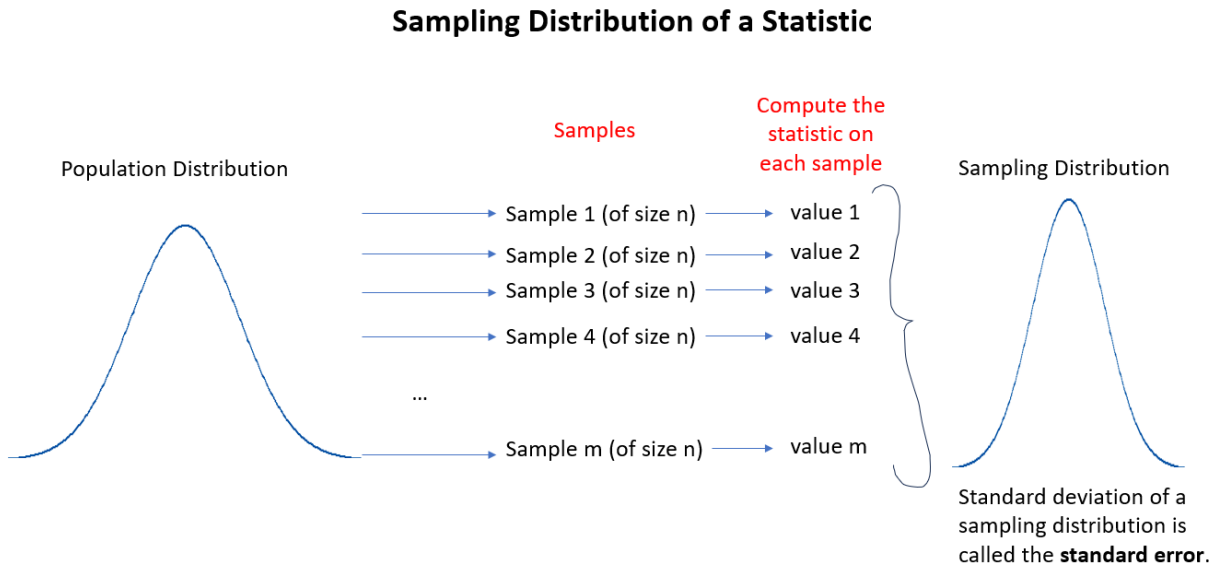
```
n <- 100
one_sample <- sample(normal_pop, n)
one_sample_mean <- mean(one_sample)
one_sample_mean
```

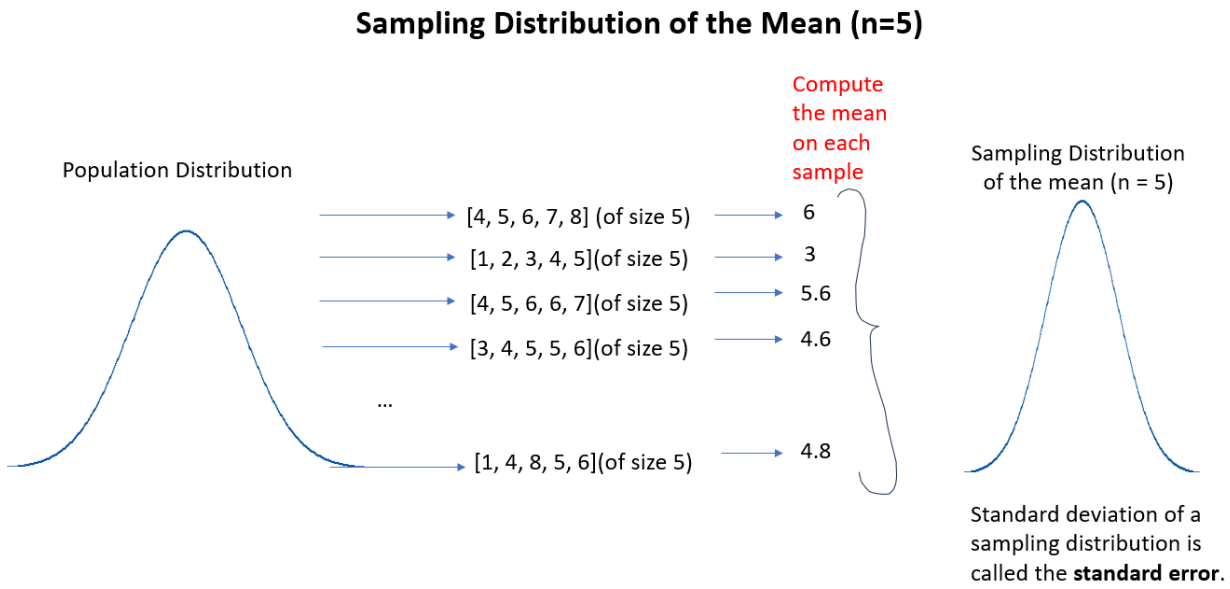
Output:

Creating a Sampling Distribution of the Mean

What is a "statistic"?: A statistic is a numerical value or measure that summarizes some aspect of a sample. (i.e., mean, median, sample standard deviation... etc.)

Sampling Distribution: it's a distribution of a sample statistic based on all possible simple random samples of the same size from the same population.





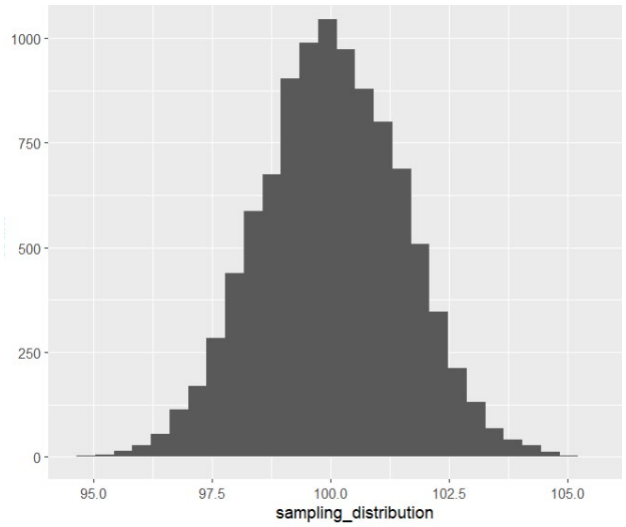
Creating multiple samples and computing the mean of each sample.

```
get_one_sample_mean  <- function(i, population_vector, n) {
  one_sample <- sample(population_vector, n)
  one_sample_mean <- mean(one_sample)
  return(one_sample_mean)
}
```

Sampling Distribution of the mean with n = 100.

```
sampling_distribution  <-  map_dbl(1:10000,  get_one_sample_mean,
population_vector= normal_pop, n = 100)
```

```
plot_4 <- data.frame(sampling_distribution) %>%
  ggplot(aes(sampling_distribution))+
  geom_histogram()
plot_4
```



The standard error is the standard deviation of the sampling distribution.

```
st_error <- sd(sampling_distribution)
st_error
```

Output:

Statistics & Bootstrapping – Part 2

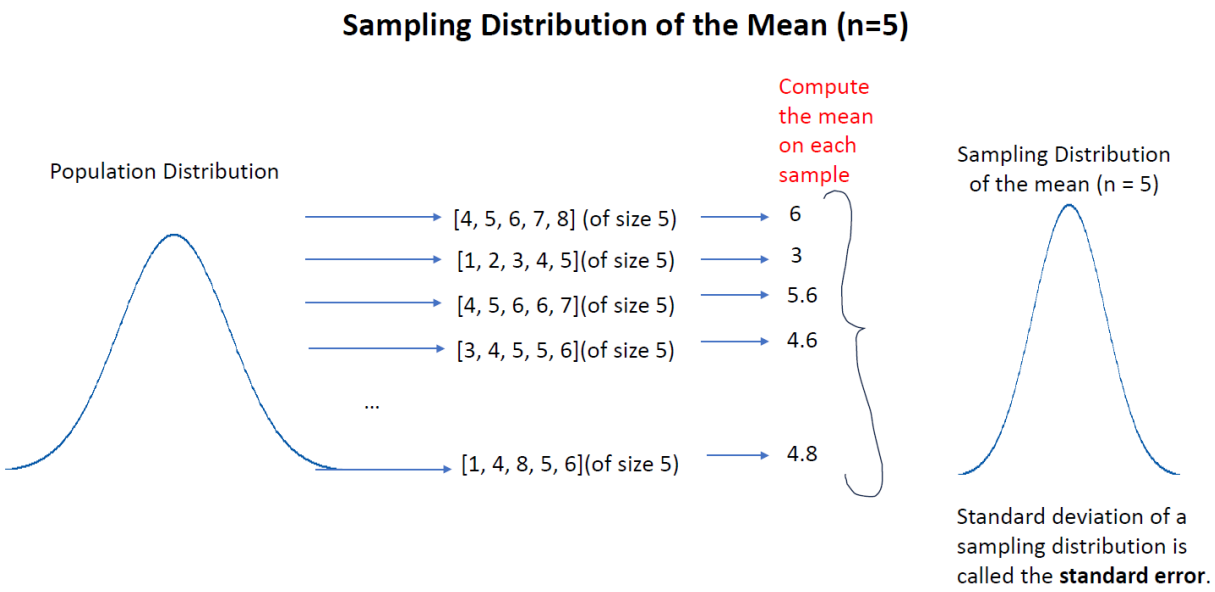
Sampling With Replacement: When you sample with replacement, the object you selected is put back into the pool before another object is sampled.

Example:

```
vector <- 1:6
sample(vector,  3,  replace = TRUE)
```

Output:

Statistical Foundations



Standard Error: it's the standard deviation of a sampling distribution.

Review: Computing the Standard Error of a Sampling Distribution of the Mean with n = 100

Population

```
normal_pop <- rnorm(1000000, mean = 100, sd = 15)
```

Function

```
get_one_sample_mean  <- function(i, population_vector, n) {
  one_sample <- sample(population_vector, n)
  one_sample_mean <- mean(one_sample)
  return(one_sample_mean)
}
```

Sampling Distribution of the Mean with n = 100

```
sampling_distribution <- map_dbl(1:10000, get_one_sample_mean,
population_vector= normal_pop, n = 100)
```

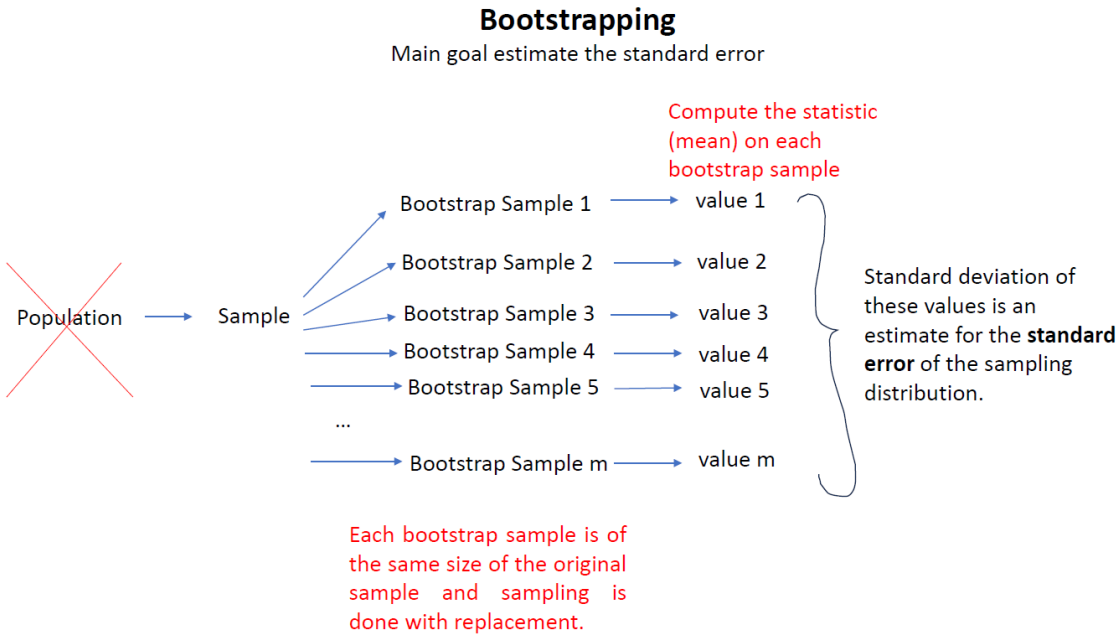
Standard Error

```
st_error <- sd(sampling_distribution)
st_error
```

Bootstrapping

The bootstrap is a resampling technique used to estimate standard errors and confidence intervals for sample statistics. It's particularly useful when you don't know the underlying data distribution.

A bootstrap sample is a sample of the same size of the original sample and sampling is done with replacement.

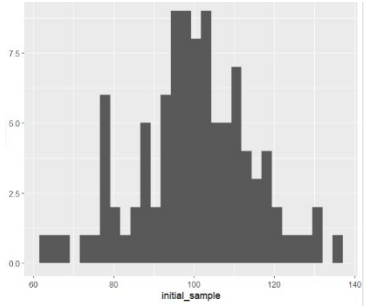


Step 1: Start with a sample.

```
initial_sample <- sample(normal_pop, 100)
```

Plot of the Initial Sample

```
sample_plot <- data.frame(initial_sample) %>%
  ggplot(aes(x=initial_sample)) +
  geom_histogram()
sample_plot
```



Step 2: Create a function that draws one bootstrap sample from a given sample and computes a statistic on the bootstrap sample. In this example the statistic is the mean.

```
boot_mean <- function(i, y){
  boot_sample <- sample(y, length(y), replace = TRUE)
  value <- mean(boot_sample)
  return(value)
}
```

Step 3: Using the function from Step 2, draw 10000 bootstrap samples and compute the mean on each.

```
all_values <- map_dbl(1:10000, boot_mean, y = initial_sample)
```

Step 4: Compute the standard deviation of the 10000 bootstrap sample means from Step 3. That is the estimated standard error.

```
sd(all_values)
```

Output: 1.435214