Chapter 14. Generating more samples; resampling methods

Steve Elston

2/28/2021

Introduction to Resampling

"There were others who had forced their way to the top from the lowest rung by the aid of their bootstraps." James Joyce, 'Ulysses' 1922

Resampling methods are powerful and widely used in computational statistics. By repeatedly re-sampling data some of the assumptions of classical statistical methods can be relaxed. These computationally intensive methods are largely products of the computer age. Resampling methods provide a natural way to find uncertainty when performing statistical inferences.

Resampling methods draw heavily on the central limit theorem (CLT) (Chapter XX) and the weak law of large numbers (Chapter XX). The weak law of large numbers tells us that a resampled estimate of a static converges to the correct value, when certain conditions are met. The CLT tells us that the sampling distribution of mean estimates are converge to a Normal, as the number of resamples increases.

There are a great many use cases for resampling methods. Specifically re-sampling methods:

- Estimate a probability distribution of a statistic.
- Make minimal distributional assumptions, when compared to classical frequentist statistics.
- Are computationally intensive, but often highly parallelizable.

Commonly used re-sampling methods include:

- **Randomization or Permutation methods:** for hypothesis tests.
- **Non-parametric bootstrap resampling:** to compute statistics.
- **Jackknife:** or leave one out re-sampling to compute statistics.
- **Cross validation:** resample into multiple folds without replacement to assess performance of statistical and machine learning models.

Randomization and permutation methods

Randomization and permutation methods were pioneered by Fisher as early as 1911. Fisher fully developed the theory in his 1935 book. Scalability of fully rank permutation methods remain limited, even with modern computers. But, modern methods using limited numbers of resamples have proved robust and scalable.

A null distribution is estimated by randomly permuting the response variable. The statistic is computed many times using a different permutation each time. This result represents a sampling distribution of a null model. A null distribution holds if the statistic is not statistically significant and the model is not predictive. A test statistic is then compared to the quantile of the null distribution.

Jack knife methods

Jack knife methods are often effective when there are only limited data samples. Maurice Quenouille originally suggested this method in 1949. The jack knife was fully developed by John W. Tukey, who gave the method its name, in 1958. Tukey saw that method as a simple tool useful for many purposes like a pocket knife.

The jack knife computes multiple values of a statistic by **leaving out one observation** each time. Therefore, for n observations there are n estimates of the statistic. The expected value of the statistic is then the mean of the resampled estimates.

Jack knife estimates often work surprisingly well for small samples. As a result of this and some other useful properties, jack knife methods are still in use today.

Cross-validation

Today, cross-validation is widely used in the testing of machine learning models. Cross-validation was originally proposed by Kurtz in 1948. Mosier extended the method to double cross validation in 1951. The modern method of nested or multicross-validation were introduced by Krus and Fuller in 1982.

At each resample, the cross validation algorithm evaluates a model by dividing the cases into **k folds**. For number of observations n, each fold contains n/k samples. The model parameters are estimated (model trained in machine learning terminology) using k-1 folds and then evaluated using the kth fold. This process is repeated k times. The average model performance and the variance of the performance metrics is then computed from the k cross validation estimates.

When $k = n$, cross validation is a leave one out algorithm. In this case, cross validation is similar to the jack knife algorithm.

Bootstrap

The bootstrap is an extremely general and powerful re-sampling method. In principle, the bootstrap algorithm and can provide estimates of the distributions of most any statistic. The bootstrap method was first suggested by Efron and Hinkley in 1978 and further developed by Efron in 1979. A full treatment was provided in Efron's 1980 book.

By repeatedly re-sampling the data, bootstrap methods relax some of the assumptions of classical statistical methods. For example, nonparametric bootstrap methods do not require any assumptions about a sampling distribution. In effect, bootstrap methods trade intensive computations for the mind power of the statistician.

As with other re-sampling methods, the bootstrap algorithm is computationally intensive. However, with increased computing power, use of bootstrap methods continues to expand. Further, the algorithm can be readily parallelized.

The bootstrap algorithm is the focus of this chapter.

Pitfalls of Resampling Methods

Re-sampling methods are general and powerful but, there is no magic involved! There are pitfalls one should be aware of!

Resampled estimate of a statistic can be no better that the original sample of observations allows. If a sample is biased, the re-sampled statistic estimate based on that sample will be biased. As an example consider that the bootstrap estimate of mean is an **unbiased sample estimate**, $\bar{x}$. But, there is no guarantee this estimate is unbiased with respect to the population parameter, μ.

The resampled variance and confidence intervals can be no better than the sample distribution allows. In fact, bootstrap CIs are known to be optimistically biased. Be suspicious if the confidence intervals you compute seem too good to be true!

All resampling methods are computationally intensive. However, all of the commonly used methods are highly parallelizable. Thus, in 21st Century computing environments, resampling methods are quite scalable. But there are limits. Computing resamples statistics from very large data sets directly can be prohibitive.

Point Estimates vs. Distribution Estimates

The goal of **frequentist statistics** is to compute a **point estimate** of a statistic or parameter and **confidence interval** for the point estimate. By a point estimate, we mean a single most likely value. The confidence interval is based on the properties of some assumed sampling distribution.For example, for the difference in means, we estimate the confidence intervals by assuming a t-distribution for the sampling distribution.

Bootstrap methods are firmly in this frequentist camp. The goal is to estimate the sampling distribution using bootstrap resamples, $\hat{\mathcal{F}}^*$, of the original sample, $\hat{\mathcal{F}}$. The statistic computed from each resample, $s(\hat{\mathcal{F}}^*)$. is assumed to arise from the sampling distribution.

Rather than computing a point estimate directly, bootstrap methods compute a **bootstrap distribution** of the statistic. The bootstrap distribution is an approximation of the sampling distribution of the statistic. The concept of sampling the bootstrap distribution is shown in the figure.

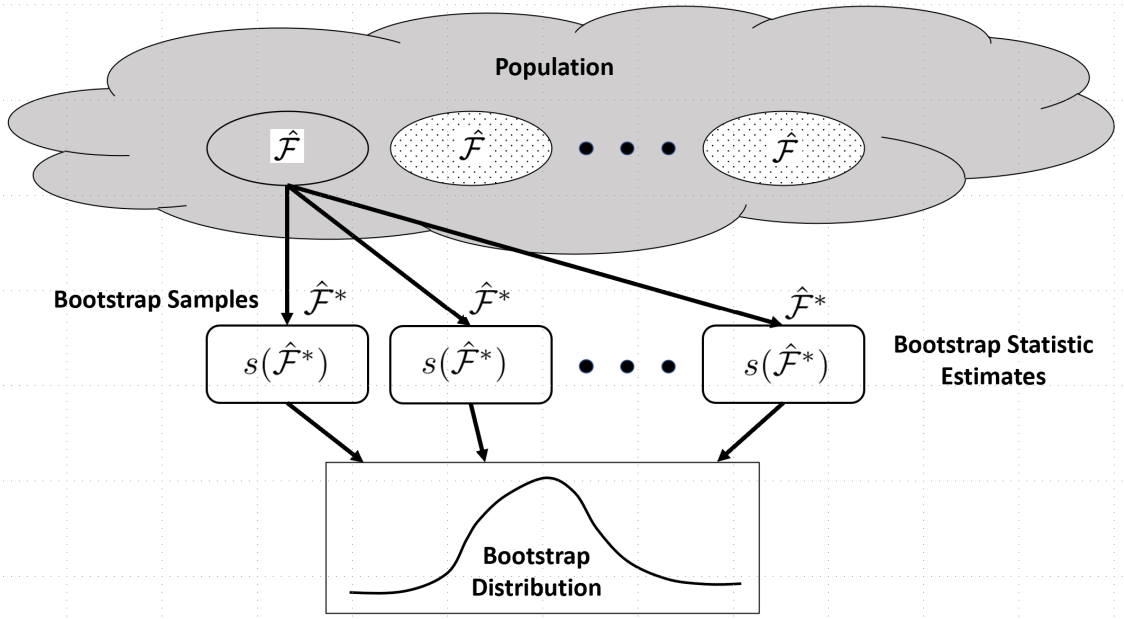


Figure 1: Resampling to estiamte the bootstrap distribution of a statistic

The bootstrap distribution is comprised of values of the statistic computed from **bootstrap samples** of the original data sample. Based on this distribution a mostly likely point estimate of the statistic, or **bootstrap estimate**, is computed as the mean of the bootstrap distribution. The **bootstrap confidence interval** is also computed from the bootstrap distribution. This approach is in contrast to the purely frequentist approach of computing point estimates and confidence intervals using an assumed sampling distribution.

Overview of the Nonparametric Bootstrap Algorithm

The nonparametric bootstrap algorithm is used to compute an estimate of the sampling distribution of most any statistic. The term **nonparametric** is applied in this case, since no assumptions about parametric sampling distributions are required. Instead, the resample estimates of the statistic are an approximation of the sampling distribution.

Bootstrap samples and the bootstrap distributon

Rather than computing a point estimate directly, bootstrap methods compute a **bootstrap distribution** of a statistic. The bootstrap distribution is comprised of values of the statistic computed from bootstrap resamples of the original observations (data sample). Computing the nonparametric bootstrap distribution requires **no assumptions about population distribution!**.

The nonparametric **bootstrap estimate** of a statistic is mostly likely point estimate of the statistic given the bootstrap distribution. As a consequence of the central limit theorem, the bootstrap estimate is the mean of the bootstrap distribution. We will address computing bootstrap confidence intervals shortly.

The nonparametric bootstrap algorithm

The nonparametric bootstrap method follows a simple algorithm. Estimates of the statistic are accumulated by these steps:
1. **Randomly sample with replacement** (e.g. Bernoulli sample) N values from an original data sample of N values. That is, the re-sample is the same size as the original data sample. This resample is called a **bootstrap sample**.
2. Re-compute the statistic with the current bootstrap sample. This is a **bootstrap estimate** of the statistic.
3. Repeat steps 1 and 2 to accumulate the required number of bootstrap estiamtes of the statistic. 4. The accumulated statistic values form the **bootstrap distribution**.
5. The mean of the computed statistic values is the **bootstrap point estimate** of the statistic.
For example, you can compute the bootstrap mean as:

$$Meanboot = \frac{\Sigma_i mean(sample_i)}{nsample}$$

where, for example, given 10 data values, the ith bootstrap sample might be:

$$sample_i = X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_1 + X_5$$

Notice two key points about the bootstrap sample which are results of randomly sampling with replacement:
1. Some values from the orriginal sample will not be included in a bootstrap sample.
2. Some values from the orriginal sample will occur multiple times in the bootstap sample.

This nonparametric bootstrap algorithm is illustrated in the figure.

Example; one-sample bootstrap

Computing a bootstrap distribution of a mean estimate is one of the simplest examples of applying the nonparametric bootstrap algorithm. A bootstrap estimates of the mean are computed a number of times from a single original sample. The result is a bootstrap distribution of the mean. Since a single original sample is resampled, this methods is know as the **one-sample bootstrap algorithm**. While we focus on the mean statistic in this example, it is important to realize that this algorithm is applicable to **most any one-sample statistic**.

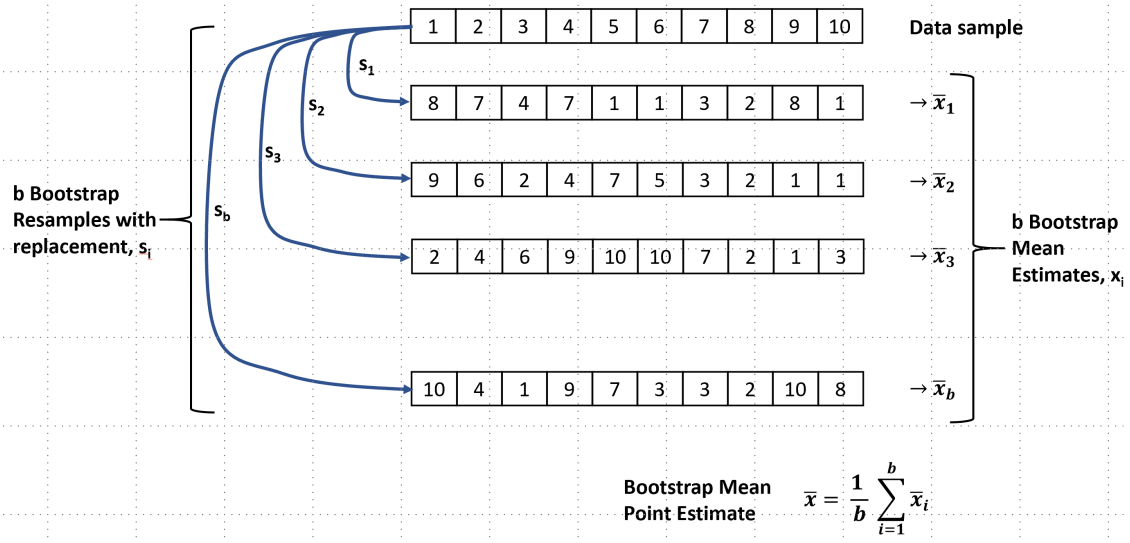


Figure 2: Outline of bootstrap resampling algorithm to compute mean

In this section we will compute the bootstrap distribution and bootstrap mean estimates from student standardized math test scores. This data set is known as HSB2 and is a subset of a larger sample. The data were collected by the National Center for Education Statistics.

We start by examining the head of a data frame containing these data.

```
import pandas as pd
import numpy as np
import numpy.random as nr
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm

test_scores = pd.read_csv('../data/hsb2.csv', index_col=0)
test_scores.head()
```

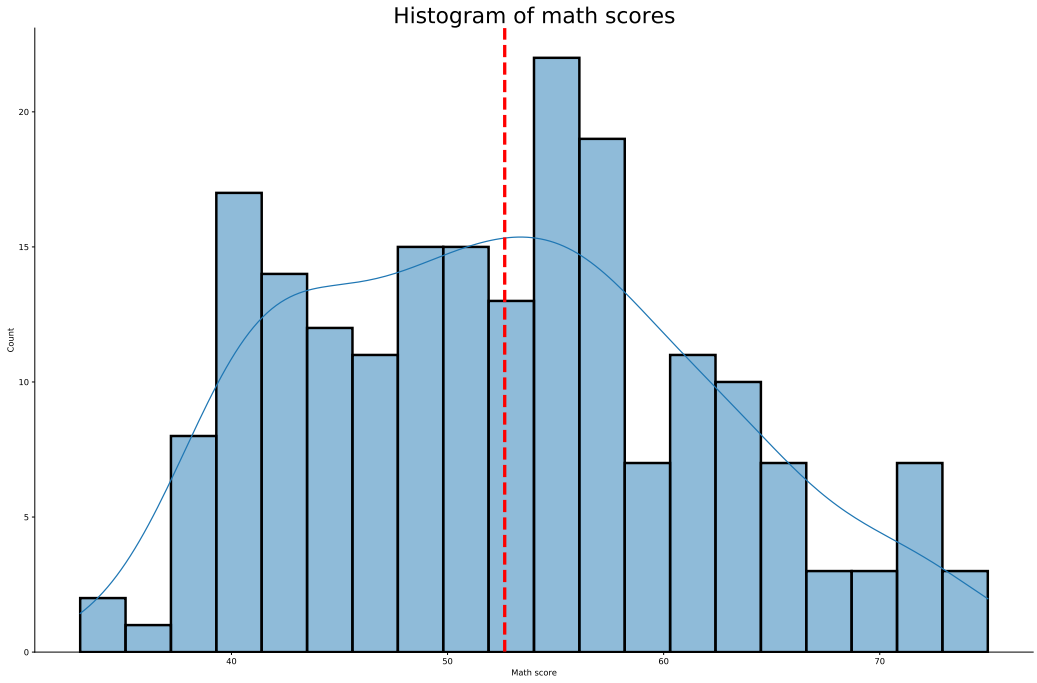
```
##      female  race  ses  schtyp  prog  read  write  math  science  socst
## id
## 70        0     4    1       1     1    57     52    41       47     57
## 121       1     4    2       1     3    68     59    53       63     61
## 86        0     4    3       1     1    44     33    54       58     31
## 141       0     4    3       1     3    63     44    47       53     56
## 172       0     4    2       1     2    47     52    57       53     61
```

The first three columns show the student's sex, race and socioeconomic status (SES). The next columns indicate the type of school (public or private) and the type of program the student is in (general, academic, vocational). The final columns contain the students' scores on standardized tests for five subjects.

The code in the cell below displays a histogram with kernel density estimate of the math scores of all students in the sample.

```
## Plot the histogram of the math scores
def plot_hist(x, xlab, title, bins=20, height=15):
       sns.displot(x, bins=bins, kde=True, height=height, aspect=1.4, linewidth=3)
```

```
        plt.rcParams.update({'font.size': 22})
        plt.axvline(x=np.mean(x), color='red', linestyle='dashed', linewidth=4)
        plt.subplots_adjust(left=0.1, bottom=0.1, right=0.9, top=0.8)
        plt.xlabel(xlab)
        plt.title(title)

math = test_scores.loc[:,'math']
plot_hist(math, 'Math score', 'Histogram of math scores')
plt.show()
```



It is questionable if the distribution of math scores is Normally distributed. Fortunately, using bootstrap methods we do not need to concern ourselves with either the population or sampling distribution assumptions.

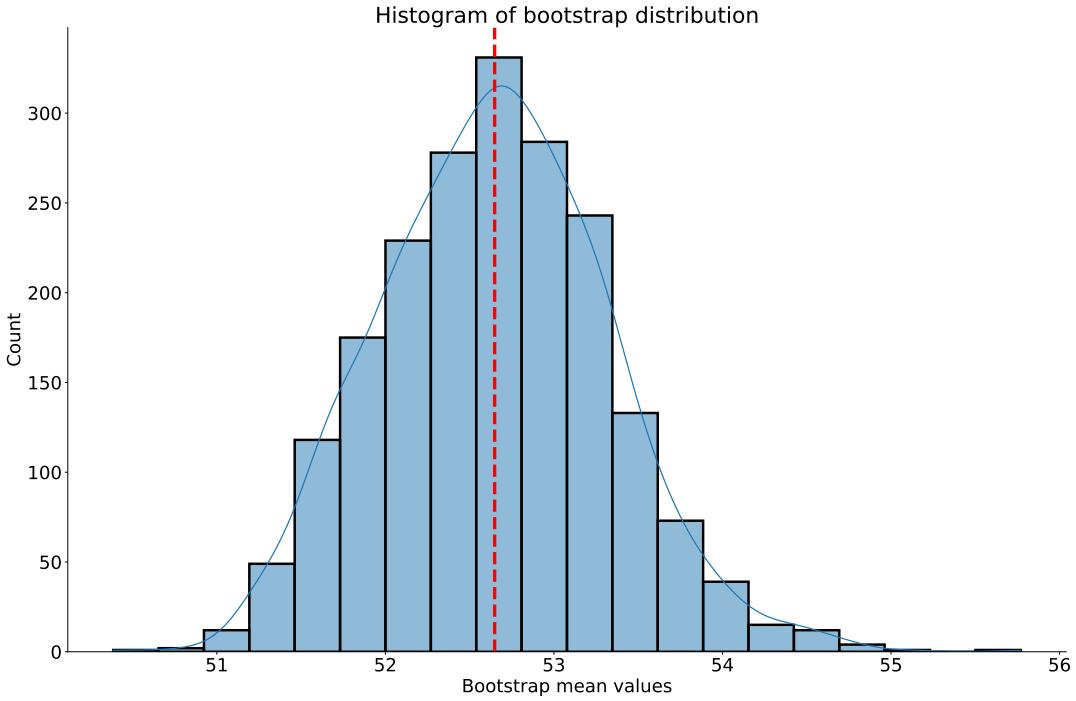
The code below generates bootstrap and bootstrap mean estimates from those samples samples. These resamples are drawn with replacement from the math scores of all students using the numpy.random.choice function. The point estimate of the mean is printed and a histogram of the bootstrap distribution is then displayed.

```
## Compute and plot the one-sample bootstrap distribution of the mean
def bootstrap_statistic(x, b, statistic):
    n_samps = len(x)
    boot_vals = []
    for _ in range(b):
        boot_vals.append(statistic(nr.choice(x, size=n_samps, replace=True)))
    boot_estimate = np.mean(boot_vals)
```

```
    print('Bootstap point estimate = {:6.2f}'.format(boot_estimate))
    return(boot_estimate, boot_vals)

bootstrap_mean_estimate, boot_means = bootstrap_statistic(math, 2000, np.mean)
```

```
## Bootstap point estimate =  52.65
```

```
plot_hist(boot_means, 'Bootstrap mean values', 'Histogram of bootstrap distribution')
plt.show()
```



You can see that the bootstrap distribution of the mean estimate is close to Normally distributed. This is a result of the CLT.

You may well wonder how many bootstrap samples should you use to estimate the bootstrap distribution? Efrom and Tibshirani (1993) and Efron and Hasti (2016) recommend using at least 200 bootstrap samples for point estimates. More recent work by several authors, including Chihara and Hesterberg (2018) indicate that more samples are desirable. With modern computers using 2,000 or more samples is considered good practice.

> **Exercise 14-1:** In order to verify that the bootstrap distribution of the mean estimate for the math scores is nearly Normally distributed compute and display the quantile-quantial (Q-Q) plot of the bootstrap mean estimates. You can use the statsmodels.graphics.gofplots.qqplot function, with the `line=45` argument. With the excepition of a few outliers, does the bootstrap distribution appear Normally distributed?

Exercise 14-2: As has been discussed, the one-sample bootstrap algorithm can be applied to most any suitable statistic. Using 200 resamples, compute the bootstrap point estiamte of the meadian of the agregate math scores from the SBS2 dataset. Display the histogram and Q-Q plots of the bootstrap distribution. Notice the deviations from the Normal distribution of this statistic. Why do you think this should be the case?

correlation coefficient as a two-sample estiamte

Do one-sample bootstrap methods only apply to statistics using a single variable. Not at all! An example is the correlation coefficient.

Correlation coefficients measure the dependency of one variable on another. But, correlation coefficients are computed from a one-sample, in the form of pairs of values of random variables. Consider several of the many possible formulations of the Pearson's correlation coefficient:

$$\rho_{\mathbf{X},\mathbf{Y}} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 (y_i - \bar{y})^2}} \tag{1}$$

$$= \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y} \tag{2}$$

$$= \frac{\mathbf{E}[(x_i - \bar{x})(y_i - \bar{y})]}{\sigma_x \sigma_y} = \frac{cov}{\sigma_x \sigma_y} \tag{3}$$

Computation of of the correlation coefficient between two random variables, $\mathbf{X}$ and $\mathbf{Y}$, is performed using matched pairs of samples, x_i and y_i. These are not random samples from each variable. They are sampled as a single related pair; that is, a one-sample.

As a result of the foregoing, the nonparametric bootstrap distribution of correlations coefficients is computed by bootstrap resampling the pair of variables, $\mathbf{X}$ and $\mathbf{Y}$. In other words, a single bootstrap resample drawing related values. Notice that the sample nonparametric bootstrap can be applied to any of the commonly used correlation coefficients, not just Pearson's.

Bootstrap Confidence Intervals

Now that we can compute a bootstrap distribution the next step is to find the confidence intervals so that we can perform some statistical inference. Classical methods of computing the bounds of confidence intervals rely on assumptions of the sampling distribution. Whereas, nonparametric confidence intervals are free of these assumptions.

The direct approach to computing nonparametric confidence intervals is know as the **percentile method**. In simple terms, percentile method finds the bootstrap sample values at the $\alpha/2$ and $1 - \alpha/2$ points of the bootstrap distribution. The percentile confidence interval algorithm is nicely simple and has few steps:
1. Define confidence level; 95% or $\alpha = 0.05$
2. Order b bootstrap samples, $s_i\$$, by value
3. Lower CI index; $i = b * \alpha/2$
4. Upper CI index; $i = b * (1 - \alpha/2)$

As has been noted already, the percentile algorithm is quite simple. However, the resulting confidence intervals are known to be biased! The bias is generally worse for statistics with asymmetric sampling distributions. Often, this bias results in overly optimistic confidence intervals. There are several well-known bias corrections which are typically applied to percentile method confidence intervals. For now, we will just say that one should be highly suspicious of confidence intervals that seem too good to be true!

In the preceding section we computed and displayed the nonparametric bootstrap estimates of the mean of the students' math scores. Now, we will extend this analysis to include estimates of the confidence intervals. The code shown using the basic percentile method to do just this.

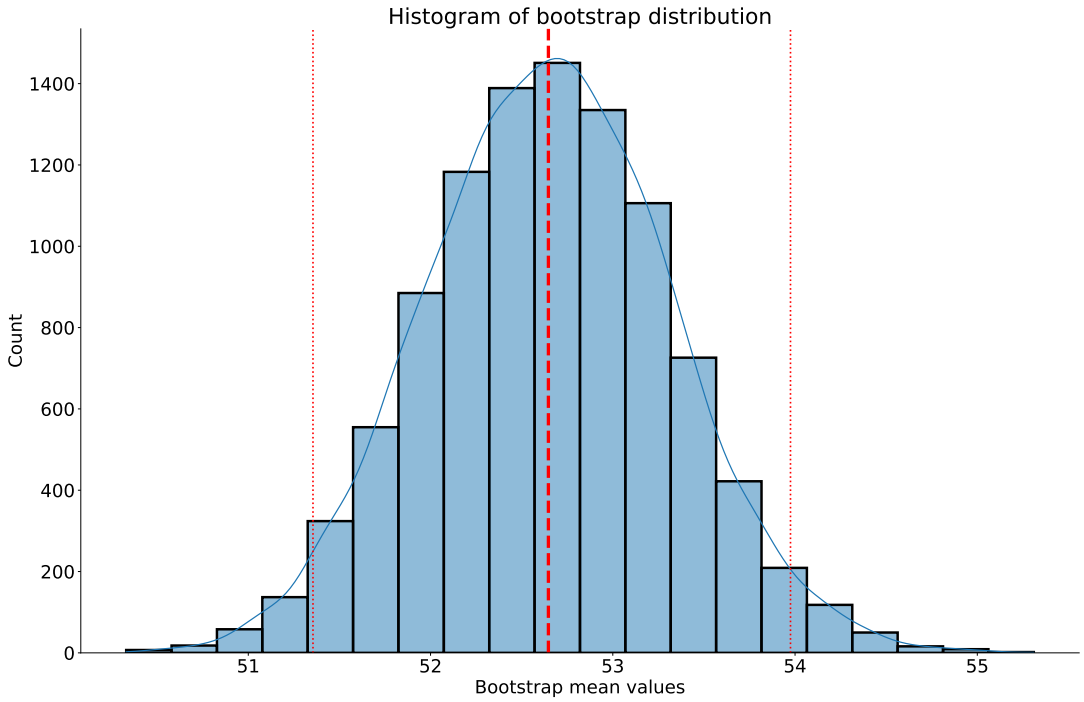
```
## Compute and plot the one-sample bootstrap of means with confidence intervals
def bootstrap_cis(boot_samples, alpha=0.05):
      n = len(boot_samples)
      sorted = np.sort(boot_samples)
      index_lci = int(n * alpha / 2)
      index_uci = int(n * (1 - alpha / 2))
      print('At alpha = {0:3.2f}, lower and upper bootstap confidence intervals = {1:6.2f}   {2:6.2f}'.
      return(sorted[index_lci], sorted[index_uci])

bootstrap_mean_estimate, boot_means = bootstrap_statistic(math, 10000, np.mean)
```

```
## Bootstap point estimate =  52.65
```

```
LCI, UCI = bootstrap_cis(boot_means)
```

```
## At alpha = 0.05, lower and upper bootstap confidence intervals =  51.35    53.98
```

```
plot_hist(boot_means, 'Bootstrap mean values', 'Histogram of bootstrap distribution')
plt.axvline(x=LCI, color='red', linestyle='dotted', linewidth=2)
plt.axvline(x=UCI, color='red', linestyle='dotted', linewidth=2)
plt.show()
```



Now we have an idea of how much variation or uncertainty we can expect for the mean estimate of the students' math scores.

As with the bootstrap point estimates, there is a question of how many bootstrap samples should you use to estimate the bootstrap confidence intervals? Since confidence intervals are estimated in the tails of the bootstrap distribution, more samples than the case of point estimates should be used. Efrom and Tibshirani (1993) and Efron and Hasti (2016) recommend using at least 2,000 bootstrap samples for confidence interval estimates. More recent work by several authors, including Chihara and Hesterberg (2018) indicate that more samples are desirable. With modern computers using 10,000 or more samples is considered good practice.

Exercise 14-3: In exercise 14-2, you computed the bootstrap distibution and point estiamte of the median of the student math scores. Now you will compute and print the 95% confidence intervals of this bootstrap distribution. Plot the historgram of the bootstrap distribution with vertical lines showing the point estiamte and the confidence interval. What does do these statistics tell you about how confident you can be about the estimate of the median?

Exercise 14-4: As discussed, the bootstrap distribution is an estiamte of the sampling distribution of a statistic. The bootstrap confidence intervals should not change with the number of resamples, except for a small variation in error. To demonstrate this idea compute bootstrap distributions of the mean estiamte of the student math scores using 500, 2000, 5000, and 10000 resamples. Compute and print the 95% confidence intervals for each of these bootstrap distributions. Is there any substanitial or systematic change in the confidence intervals?

Exercise 14-5: The sampling distribution of the variance for Normally distributed random variables is well known to be χ^2. This result is still a reasonably good approximation for cases where a random variable is not Normally distributed, as is the case with the math scores. Using 1,000 bootstap resamples you will compute the bootstrap distribution, the point estiamte and the confidence intervals can all be computed. You will now use the one-sample bootstrap algorithm to compute and print the variance and confidence interval for the students' math scores. Plot the historgram of the bootstrap distribution with vertical lines showing the point estiamte and the bounds of the confidence interval. What does do these statistics tell you about how confident you can be about the estimate of the variance?

Exercise 14-6: In a pervious section, we have discussed how the one-sample nonparametric bootstrap algorithm can be applied to Pearson's correlation coefficients. Now you will compute the bootstrap distributuion of the correlation between the students' math and science scores using 2,000 resamples.

The sampling distribution of the correlation coefficient is limited to the range $-1 \leq \rho_{\mathbf{X},\mathbf{Y}} \leq 1$. These bounds can lead to a biased sampling distribution. Fisher proposed a simple transformation to avoid this problem when computing confidence intervals of Pearson's correlation coefficients:

$$F(\rho_{\mathbf{X},\mathbf{Y}}) = arctan(\rho_{\mathbf{X},\mathbf{Y}})$$

You should apply the FIsher's transformation to your bootrarap estiamtes of the correlation coefficient.

Once you have computed the bootstrap distribution, compute and print the mean and 95% confidence intervals of the correlation between the math and scince score varaibles. Next, plot the histogram including vertical lines for the mean and confidence interval bounds. Does it appear that there is consistently postive correlation between these two variables? Given the confidence intervals, would you say that this correlation is realatively weak of strong?

Two-sample Bootstrap

In the preceding we have worked only with one-sample statistics. How can we apply the bootstrap algorithm for two-sample statistics? two-sample statistics are used extensively for statistical inference. For example, we might want to perform inference on the difference of means of two independently sampled populations. Our null hypothesis is that there is no significant difference.

In classical statistical theory the difference of means of approximately Normally distributed random variables is assumed to be t-distributed. This assumption leads to the well-know t family of hypothesis tests. Using nonparametic bootstrap resampling frees us of these assumptions. We can make inferences of difference of means for non-Normally distributed samples. Further, we can easily perform inferences on the differences of most any statistic.

The basic idea of computing nonparametric bootstrap distributions can be applied to two-sample statistics. In fact, it can be applied to multiple sample statistics. But, to do so one must carefully consider how the bootstrap samples are generated.

Can we just sample the concatenation of the two-samples? **No!** The problem is that we what the resample of each of the random variables to have the size of the respective original samples. If we resample in a naive way, there is no guarantee of having correct number of resamples for each random variable. This imbalance in the sizes of the resamples leads to significant biases. The solution is to **independently resample** the two random variables in order to correctly sample each of the populations.

Algorithm for the two-sample bootstrap

The two-sample nonparametric bootstrap algorithm ensures that the two random variables are independently resampled. This algorithm is requires some extra steps. For example, the two-sample statistic may itself be computed using statistics of the two random variables. For example, to compute the difference of means, independent bootstrap mean estimates of each of the random variables is required.

The basic two-sample nonparametric bootstrap algorithm can be described by these steps:
1. Independently randomly sample (e.g. Bernoulli sample) n data with replacement from each original data sample ensuring the number of resamples for each random variable is the number of original samples for each population.
2. Compute the statistic (e.g. the mean) for the two resamples.
3. Compute the two-sample statistic; e.g. difference of means.
4. Repeat steps 1, 2, and 3 to accumulate the required number of bootstrap samples of the bootstrap distribution.
5. The mean of the bootstrap distribution values is the bootstrap point estimate of the statistic.
6. Compute CIs from bootstrap distribution.

Once the two-sample bootstrap distribution and confidence intervals has been computed we can perform statistical inference. Typically, the inference involves determining if one can reject a null hypothesis given the bounds on the confidence interval. For example, if we are interested in the difference of means, the null hypothesis is that the difference is not significant with the confidence specified. However, if the confidence bounds of the statistic do not include zero we can say the difference is significant and reject this null hypothesis. In other words, we can reject the null hypothesis if the sign of the confidence bounds are the same; indicating the confidence interval is above or below zero. z

Example of the two-sample bootstrap

Let's consider a simple example of two-sample nonparametric bootstrap inference. In this case, we will determine if there is a significant difference in the mean math scores of the students drawn from the populations with low and mid socioeconomic status. Our null hypothesis is that there is no significant difference. We

can perform this inference confidently dispite the fact that the students' math scores do not appear to be Normally distributed.

The code example applies the two-sample nonparametric algorithm to the difference in means of the math scores of the low and mid socioeconomic status students. There are a few important points to notice about this code:
- The number of resamples is equal to the original sample size for each of the random variables.
- The mean statistic of each of the resampled random variables is computed independently and the difference in mean statistic is computed from these results.

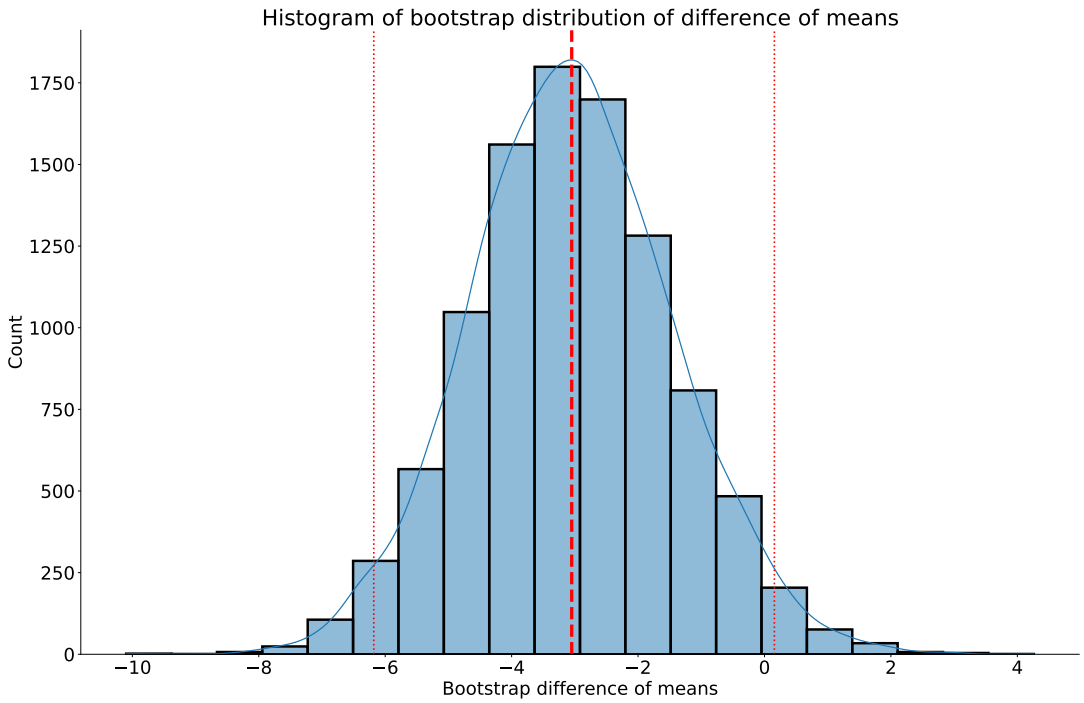
```
# Bootstrap the difference of means of low and mid SES students
def two_boot_two_stat(sample_1, sample_2, b, statistic_1, two_samp_statistic):
    two_boot_values = []
    n_samps_1 = len(sample_1)
    n_samps_2 = len(sample_2)
    for _ in range(b):
      boot_estimate_1 = statistic_1(nr.choice(sample_1, size=n_samps_1, replace=True))
      boot_estimate_2 = statistic_1(nr.choice(sample_2, size=n_samps_2, replace=True))
      two_boot_values.append(two_samp_statistic(boot_estimate_1, boot_estimate_2))
    boot_estimate = np.mean(two_boot_values)
    print('Bootstap point estimate = {:6.2f}'.format(boot_estimate))
    return(boot_estimate, two_boot_values)

math_low_ses = test_scores.loc[test_scores.loc[:,'ses']==1,'math']
math_mid_ses = test_scores.loc[test_scores.loc[:,'ses']==2,'math']
bootstrap_diff_of_mean, boot_diffs = two_boot_two_stat(math_low_ses, math_mid_ses, 10000, np.mean, lambd
```

```
## Bootstap point estimate =  -3.05
```

```
LCI, UCI = bootstrap_cis(boot_diffs)
```

```
## At alpha = 0.05, lower and upper bootstap confidence intervals =  -6.18     0.16
```

```
plot_hist(boot_diffs, 'Bootstrap difference of means', 'Histogram of bootstrap distribution of differen
plt.axvline(x=LCI, color='red', linestyle='dotted', linewidth=2)
plt.axvline(x=UCI, color='red', linestyle='dotted', linewidth=2)
plt.show()
```



The bounds of the confidence interval have different signs, but only barely. In other words, we cannot quite have 95% confidence in rejecting the null hypothesis that this difference in treatments (different socioeconomic status) does not affect the student's test scores. Still, we do see that there does seem to be at least some effect. A larger sample size might well change our inference on this problem.

Exercise 14-7: There are significant mathematical challenges with performing classical inference on the difference of medians between two random variables. But fortunately, we can use nonparametric two-sample bootstrap methods to perform such inference. You will now compute the nonparametric bootstrap distribution of the difference of medians between low and mid socioeconomic status students. Using this distrubiton compute and print the mean of the difference of medians along with the bounds of the confidence interval. Then, plot a histogram of the bootstrap distribution showing the mean of the difference of medians and the bounds of the confidence intervals as vertical lines. Can you reject the null hyporthesis that there is no median difference in math scores between these two socioeconomic groups and why?

Exercise 14-8: The small sample size makes inference between differences the low and mid socioeconomic status students difficlut. But what inferences can you make between the low and high socioeconomic status students? Repeat the process of computing the nonparametric bootstrap distribution of the difference of medians of math scores of low and high socioeconomic status students. Using this distrubiton compute and print the mean of the difference of medians along with the bounds of the confidence interval. Then, plot a histogram of the bootstrap distribution showing the mean of the difference of medians and the bounds of the confidence intervals as vertical lines. Can you reject the null hyporthesis that there is no median difference in math scores between these two socioeconomic groups and why?