EMAT30007 Applied Statistics

Lab 3: Confidence Intervals

This lab focusses on the material covered in Lecture 3: quantifying the uncertainty of an estimated quantity.

1. z-CI and t-CI for the mean of a Normal RV

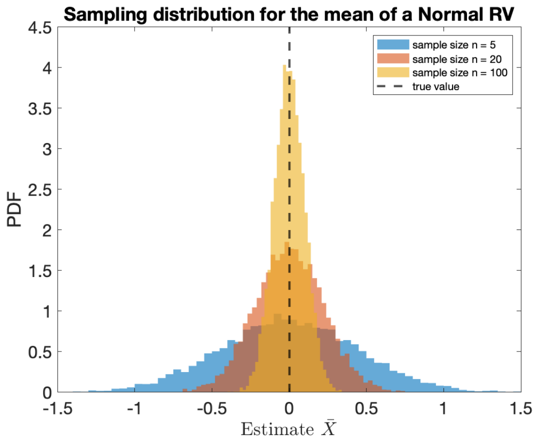
(CI = Confidence Interval)

1. Generate 10000 samples and plot the sampling distribution of the mean for different values of the sample size n.
2. Use the empirical quantiles to compute the empirical CIs for the various n for a 95% confidence level $\alpha = 0.05$.
3. Build the z-CI (Lecture 3 Eq.(12)) and t-CI (Lecture 3 Eq.(13)) for each of the samples, using the formulae. Is the length of the empirical CIs close to that of the z-CIs? And of the t-CIs?
4. How many estimated z-CIs contain the true value? (For how many estimates the true value fall inside the z-CI ?) and the t-CIs?

Solution

```
% Normal RV
true_mu = 0;
true_sigma = 1;
pd = makedist('Normal', 'mu', true_mu, 'sigma', true_sigma);

% sampling distribution
% generate samples and estimates
ns = [5 20 100];  % size of samples
S = 10000;  % number of samples

estimates = zeros(length(ns), S);
sample_stdevs = zeros(length(ns), S);
for i = 1:length(ns)
    n = ns(i);
    for j = 1:S
        % sample from the normal RV
        x = random(pd, 1, n);
        % compute the estimate for the sample
        estimates(i, j) = mean(x);
        sample_stdevs(i, j) = std(x);
    end
end
```

```
% plot the sampling distributions

clf;
for i = 1:length(ns)
    x = estimates(i, :);
    histogram(x, 'Normalization', "pdf", 'EdgeColor','none');
    hold on
end
xline(true_mu, '--k', 'Linewidth', 2);
hold on
xlim([-1.5 1.5]);
title('Sampling distribution for the mean of a Normal RV') %, 'Interpreter', 'latex')
legend('sample size n = 5', 'sample size n = 20', 'sample size n = 100', 'true value', 'FontSize', 10.0);
xlabel('Estimate $\bar{X}$', 'Interpreter', 'latex');
ylabel('PDF');
set(gca, 'FontSize', 16.0);
```

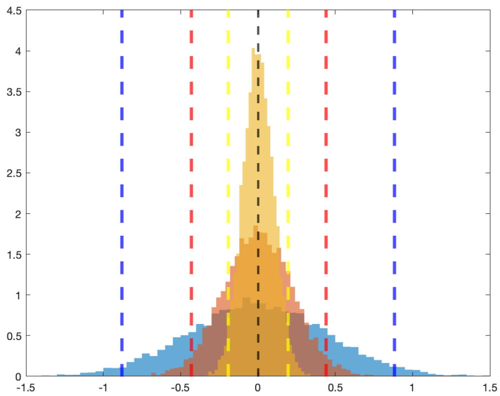



```
% compute empirical Confidence Intervals
clf

alpha = 0.05;

eCIs = zeros(length(ns), 2);
for i = 1:length(ns)
    eCIs(i, 1) = quantile(estimates(i, :), alpha / 2);
    eCIs(i, 2) = quantile(estimates(i, :), 1 - alpha / 2);
    % lengths
    eCIs(i, 2) - eCIs(i, 1)

    % plot pdf
    x = estimates(i, :);
    histogram(x, 'Normalization', "pdf", 'EdgeColor','none');
    hold on
end
```

```
ans = 1.7623
ans = 0.8700
ans = 0.3863
```

```
xline(true_mu, '--k', 'Linewidth', 2);
hold on
```

```
xline(eCIs(1, 1), '--b', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(1, 2), '--b', 'Linewidth', 3,'HandleVisibility','off');
hold on
xline(eCIs(2, 1), '--r', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(2, 2), '--r', 'Linewidth', 3,'HandleVisibility','off');
hold on
xline(eCIs(3, 1), '--y', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(3, 2), '--y', 'Linewidth', 3,'HandleVisibility','off');
xlim([-1.5 1.5]);
```



```
% compute the z-CI (Lecture 3, Eq. 12) for each sample

zCIs1  = zeros(length(ns), S);  % lower edges
zCIs2  = zeros(length(ns), S);  % upper edges
for i = 1:length(ns)
    zCIs1(i, :) = estimates(i, :) - norminv(1 - alpha / 2) * 1 / (ns(i))^0.5;
    zCIs2(i, :) = estimates(i, :) - norminv(alpha / 2) * 1 / (ns(i))^0.5;
    % average lengths
    mean(zCIs2(i, :) - zCIs1(i, :), 2)
end
```

```
ans = 1.7530
ans = 0.8765
ans = 0.3920
```

```
% fraction of intervals that contain the true mean, for the various sizes n
mean(zCIs1 < true_mu & true_mu < zCIs2, 2)  % mean over the second dimension (axis)
```

```
ans = 3×1
    0.9490
    0.9512
    0.9532
```

The size of the zCIs are very similar to those of the empirical CIs.

The fractions of intervals containing the true value are all close to 0.95, the expected number for a 95% confidence level.

```
% compute the t-CI (Lecture 3, Eq. 13) for each sample

tCIs1  = zeros(length(ns), S);  % lower edges
tCIs2  = zeros(length(ns), S);  % upper edges
for i = 1:length(ns)
    tCIs1(i, :) = estimates(i, :) - tinv(1 - alpha / 2, ns(i) - 1) * sample_stdevs(i, :) / (ns(i))^0.5;
    tCIs2(i, :) = estimates(i, :) - tinv(alpha / 2, ns(i) - 1) * sample_stdevs(i, :) / (ns(i))^0.5;
    % average sizes
    mean(tCIs2(i, :) - tCIs1(i, :), 2)
end
```

```
ans = 2.3245
ans = 0.9250
ans = 0.3960
```

```
% fraction of intervals that contain the true mean, for the various sizes n
mean(tCIs1 < true_mu & true_mu < tCIs2, 2)  % mean over the second dimension (axis)
```

```
ans = 3×1
    0.9495
    0.9519
    0.9515
```

The size of the tCIs are bigger than the empirical CIs, especially for small sample sizes n. This means that, while the tCIs do contain the true value 95% of the times, they are larger than the zCIs, for the same n. This originates from the fact that the t-distribution has a larger variance than the standard Normal distribution, especially for small n (see last plot at https://uk.mathworks.com/help/stats/students-t-distribution.html). When computing the tCI we had to estimate the standard deviation σ using the data, which introduces more uncertainty in the estimate, and hence larger tCIs are necessary for the same confidence level.

The fractions of intervals containing the true value are all close to 0.95, the expected number for a 95% confidence level.

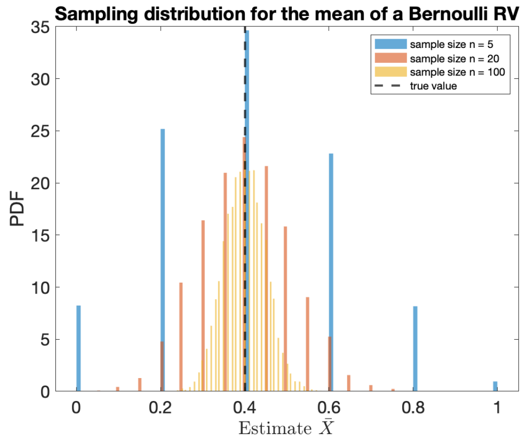
2. CI for the mean of a Bernoulli RV & the Central Limit Theorem

1. Central limit theorem (CLT): generate samples from a Bernoulli RV with $p = 0.4$ and plot the sampling distribution of the mean for different values of the sample size n . Show that the sampling distribution approaches a Normal distribution for large n.
2. Use the empirical quantiles to compute the empirical CIs for the various n for a confidence level $\alpha = 0.05$.
3. Build the z-CI for each sample using the formula Eq.(14) of Lecture 3. Is the mean length of the z-CIs close to that of the the empirical CIs ?
4. How many estimated z-CI contain the true value? (For how many estimates the true value fall inside the z-CI ?)
5. Build the t-CI for each sample using the formula Eq.(13) of Lecture 3 and compute the mean lengths and the fraction that contain the true value. What do you observe?

Solution

```
% Bernoulli RV
true_p = 0.4;
pd = makedist('Binomial', 'p', true_p);

% % The CLT is true for any distribution (with finite variance). Try other ones!
```

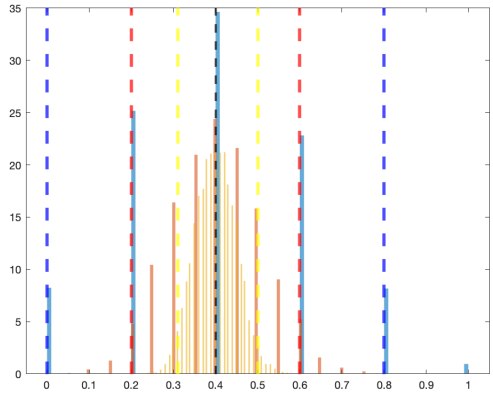
```
% pd = makedist('Exponential', 'mu', 10);
% pd = makedist('Uniform');

% sampling distribution
% generate samples and estimates
ns = [5 20 100];  % size of samples
S = 10000;  % number of samples

estimates = zeros(length(ns), S);
sample_stdevs = zeros(length(ns), S);
for i = 1:length(ns)
    n = ns(i);
    for j = 1:S
        % sample from the normal RV
        x = random(pd, 1, n);
        % compute the estimate for the sample

        estimates(i, j) = mean(x);
        sample_stdevs(i, j) = std(x);
    end
end
% plot the sampling distribution

clf;
for i = 1:length(ns)
    x = estimates(i, :);
    histogram(x, 100, 'Normalization', "pdf", 'EdgeColor','none');  % if the PDF doesn't look good, increase the number of bins
    hold on
end
xline(true_p, '--k', 'Linewidth', 2);
hold on
title('Sampling distribution for the mean of a Bernoulli RV') %, 'Interpreter', 'latex')
legend('sample size n = 5', 'sample size n = 20', 'sample size n = 100', 'true value', 'FontSize', 10.0);
xlabel('Estimate $\bar{X}$', 'Interpreter', 'latex');
ylabel('PDF');
set(gca, 'FontSize', 16.0);
```




```
% compute empirical Confidence Intervals
clf

alpha = 0.05;

eCIs = zeros(length(ns), 2);
for i = 1:length(ns)
    eCIs(i, 1) = quantile(estimates(i, :), alpha / 2);
    eCIs(i, 2) = quantile(estimates(i, :), 1 - alpha / 2);
    % sizes
    eCIs(i, 2) - eCIs(i, 1)

    x = estimates(i, :);
    histogram(x, 100, 'Normalization', "pdf", 'EdgeColor','none');
    hold on
end
```

```
ans = 0.8000
ans = 0.4000
ans = 0.1900
```

```
xline(true_p, '--k', 'Linewidth', 2);
hold on

xline(eCIs(1, 1), '--b', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(1, 2), '--b', 'Linewidth', 3,'HandleVisibility','off');
hold on
xline(eCIs(2, 1), '--r', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(2, 2), '--r', 'Linewidth', 3,'HandleVisibility','off');
hold on
xline(eCIs(3, 1), '--y', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(3, 2), '--y', 'Linewidth', 3,'HandleVisibility','off');
```




```
% compute the z-CI (Lecture 3, Eq. 12) for each sample

zCIs1  = zeros(length(ns), S);  % lower edges
zCIs2  = zeros(length(ns), S);  % upper edges
for i = 1:length(ns)
    zCIs1(i, :) = estimates(i, :) - norminv(1 - alpha / 2) * (estimates(i, :) .* (1 - estimates(i, :)) / ns(i)) .^ 0.5;
    zCIs2(i, :) = estimates(i, :) - norminv(alpha / 2) * (estimates(i, :) .* (1 - estimates(i, :)) / ns(i)) .^ 0.5;
    % average sizes
    mean(zCIs2(i, :) - zCIs1(i, :), 2)
end
```

```
ans = 0.7273
ans = 0.4179
ans = 0.1910
```

```
% fraction of intervals that contain the true mean, for the various sizes n
mean(zCIs1 < true_p & true_p < zCIs2, 2)  % mean over the second dimension (axis)
```

```
ans = 3×1
    0.8264
    0.9322
    0.9492
```

The size of the z-CIs are similar to those of the empirical CIs only for large n.

The fractions of intervals containing the true value are close to 0.95 (the expected number for a 95% confidence level) only for large n.

When a RV's distribution is not Normal, the z-CIs for the mean are correct only for large n, when the sampling distribution becomes similar to a Normal PDF because of the CLT.

```
% compute the t-CI (Lecture 3, Eq. 13) for each sample

tCIs1  = zeros(length(ns), S);  % lower edges
tCIs2  = zeros(length(ns), S);  % upper edges
for i = 1:length(ns)
    tCIs1(i, :) = estimates(i, :) - tinv(1 - alpha / 2, ns(i) - 1) * sample_stdevs(i, :) / (ns(i))^0.5;
    tCIs2(i, :) = estimates(i, :) - tinv(alpha / 2, ns(i) - 1) * sample_stdevs(i, :) / (ns(i))^0.5;
    % average sizes
    mean(tCIs2(i, :) - tCIs1(i, :), 2)
end
```

```
ans = 1.1519
ans = 0.4578
ans = 0.1944
```

```
% fraction of intervals that contain the true mean, for the various sizes n
mean(tCIs1 < true_p & true_p < tCIs2, 2)  % mean over the second dimension (axis)
```

```
ans = 3×1
    0.9081
    0.9322
    0.9492
```

t-CIs are usually larger for small n, but for large n they are comparable with the z-CIs.

3. Exercises

Fruit flies

An experiment was conducted to determine the effectiveness of heat treatment to kill fruit fly eggs in mangoes. From 5903 eggs in treated mangoes, 637 adults hatched. What is the probability p that an egg will survive the heat treatment?

1. Work out an estimate for p.
2. Work out a standard deviation for the estimate p.
3. Work out a 99% confidence interval for p.

Solution

```
n = 5903;
alpha = 0.01;

% point estimate
p = 637 / n
```

```
p = 0.1079
```

```
% standard error
se = (p * (1 - p) / n)^0.5
```

```
se = 0.0040
```

```
% z-CI
zCI = [p - norminv(1 - alpha / 2) * se p - norminv(alpha / 2) * se]
```

```
zCI = 1×2
    0.0975    0.1183
```

Call centre

The call centre for a bank samples $n = 58$ incoming phone calls and records the time taken to answer each. It is found that the average call time is 99 seconds and the variance is estimated to be 5762 sec. Find a 90% confidence interval for the mean call time. (List the assumptions you make.)

Solution

```
% using a t-CI, assuming the call time distribution is normal or not too skewed:

n = 58;
alpha = 0.1;
est = 99;
std = 2576^0.5;
tCIlower = est - tinv(1 - alpha / 2, ns(i) - 1) * std / n^0.5
```

```
tCIlower = 87.9345
```

```
tCIupper = est - tinv(alpha / 2, ns(i) - 1) * std / n^0.5
```

```
tCIupper = 110.0655
```

Exit polls

You are conducting an exit poll for a referendum. You ask $n = 100$ voters at random how they voted. You have 45% of yes in your sample.

1. Find a 99% confidence interval for the overall proportion of yes in the population.
2. Based on your sample, how confident can you be that the yes will not win the referendum?

Solution

```
n = 100;
alpha = 0.01;

% point estimate
p = 0.45
```

```
p = 0.4500
```

```
% standard error
se = (p * (1 - p) / n)^0.5
```

```
se = 0.0497
```

```
% z-CI
zCI = [p - norminv(1 - alpha / 2) * se p - norminv(alpha / 2) * se]
```

```
zCI = 1×2
    0.3219    0.5781
```

To answer #2, we have to find the value of α for which the upper edge of the CI is 0.5, given our n and p. Because of symmetry, the lower edge of the CI will then be at 0.4. For this CI,

- with probability $1 - \alpha$ the true proportion of yes will be between 0.4 and 0.5,
- with probability $\alpha/2$ the true proportion of yes will be below 0.4 and
- with probability $\alpha/2$ the true proportion of yes will be above 0.5.

Hence, we are $(1 - \alpha + \alpha/2)$% confident that the yes will not win the referendum.

We have to find α that solves this equation: $p - z_{\alpha/2}\sqrt{\frac{p(1-p)}{n}} = 0.5$.

We get $z_{\alpha/2} = F_Z^{-1}(\alpha/2) = (p - 0.5)\sqrt{\frac{n}{p(1-p)}}$ which gives

$\alpha = 2F_Z\left((p - 0.5)\sqrt{\frac{n}{p(1-p)}}\right)$

```
alpha = 2 * normcdf((p - 0.5) / se)
```

```
alpha = 0.3149
```

```
% z-CI
zCI = [p - norminv(1 - alpha / 2) * se p - norminv(alpha / 2) * se]
```

```
zCI = 1×2
    0.4000    0.5000
```

```
% confidence that yes will not win
1 - alpha + alpha/2
```

```
ans = 0.8426
```

4. When the t-CI fails: the mean of a Pareto distribution

1. Generate samples from a Pareto distribution $P_X(x) = \frac{\theta}{x^{\theta+1}}$ for $x > 1$ with $\theta = 1.5$ (use the code `x = rand(1, n) .^ (-1/theta);` from Lab 2) and plot the sampling distribution of the mean for different values of the sample size n . Is the sampling distribution of $\bar{X}$ symmetric for small n ? Does it approach a Normal distribution for large n?
2. Use the empirical quantiles of the sampling distribution to compute the empirical CIs of the mean for the various n for a confidence level $\alpha = 0.05$. Are the CIs symmetric with respect to the true value? (why?)
3. Build the t-CI for each sample. Is the mean length of the t-CIs close to that of the the empirical CIs ?
4. How many estimated t-CI contain the true value? (For how many estimates the true value fall inside the t-CI ?)

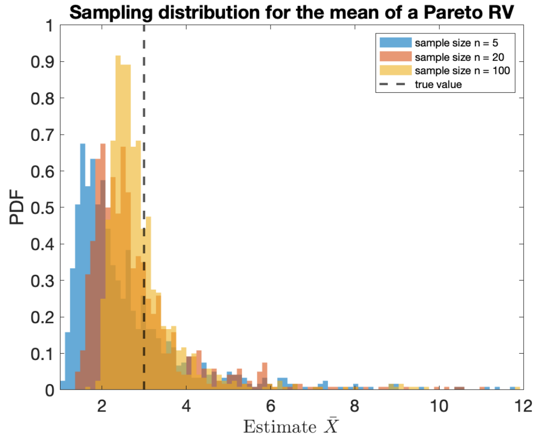
Solution

```
% Pareto RV
theta = 1.5;
% x = rand(1, n) .^ (-1 / theta);
true_m = theta / (theta - 1)
```

```
true_m = 3
```

```
% sampling distribution
% generate samples and estimates
ns = [5 20 100];  % size of samples
S = 1000;  % number of samples

estimates = zeros(length(ns), S);
sample_stdevs = zeros(length(ns), S);
for i = 1:length(ns)
    n = ns(i);
    for j = 1:S
        % sample from the normal RV
        x = rand(1, n) .^ (-1 / theta);
        % compute the estimate for the sample
```

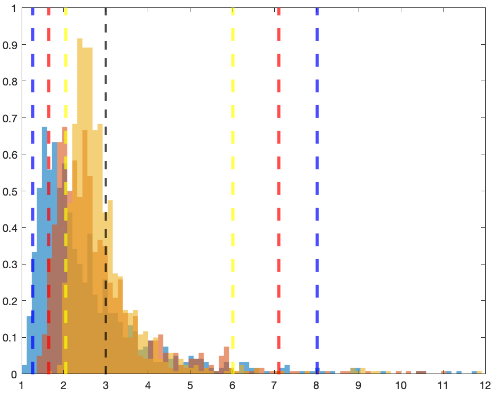
```
        estimates(i, j) = mean(x);
        sample_stdevs(i, j) = std(x);
    end
end

% plot the sampling distribution

clf;
for i = 1:length(ns)
    x = estimates(i, :);
    histogram(x, 1:true_m*4/100:true_m*4, 'Normalization', "pdf", 'EdgeColor','none');  % if the PDF doesn't look good, increase the number of b
    hold on
end
xline(true_m, '--k', 'Linewidth', 2);
hold on

title('Sampling distribution for the mean of a Pareto RV') %, 'Interpreter', 'latex')
legend('sample size n = 5', 'sample size n = 20', 'sample size n = 100', 'true value', 'FontSize', 10.0);
xlabel('Estimate $\bar{X}$', 'Interpreter', 'latex');
ylabel('PDF');
set(gca, 'FontSize', 16.0);
% set(gca, 'XScale', 'log');
xlim([1 true_m*4]);
```




```
% compute empirical Confidence Intervals
clf

alpha = 0.05;

eCIs = zeros(length(ns), 2);
for i = 1:length(ns)
    eCIs(i, 1) = quantile(estimates(i, :), alpha / 2);
    eCIs(i, 2) = quantile(estimates(i, :), 1 - alpha / 2);
    % sizes
    eCIs(i, 2) - eCIs(i, 1)

    x = estimates(i, :);
    histogram(x, 1:true_m*4/100:true_m*4, 'Normalization', "pdf", 'EdgeColor','none');
    hold on
end
```

```
ans = 6.7529
ans = 5.4716
ans = 3.9691
```

```
xline(true_m, '--k', 'Linewidth', 2);
hold on

xline(eCIs(1, 1), '--b', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(1, 2), '--b', 'Linewidth', 3,'HandleVisibility','off');
hold on
xline(eCIs(2, 1), '--r', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(2, 2), '--r', 'Linewidth', 3,'HandleVisibility','off');
hold on
xline(eCIs(3, 1), '--y', 'Linewidth', 3,'HandleVisibility','off');
xline(eCIs(3, 2), '--y', 'Linewidth', 3,'HandleVisibility','off');
xlim([1 true_m*4]);
```



```
% compute the t-CI (Lecture 3, Eq. 13) for each sample

tCIs1  = zeros(length(ns), S);  % lower edges
```

```
tCIs2  = zeros(length(ns), S);  % upper edges
for i = 1:length(ns)
    tCIs1(i, :) = estimates(i, :) - tinv(1 - alpha / 2, ns(i) - 1) * sample_stdevs(i, :) / (ns(i))^0.5;
    tCIs2(i, :) = estimates(i, :) - tinv(alpha / 2, ns(i) - 1) * sample_stdevs(i, :) / (ns(i))^0.5;
    % average sizes
    mean(tCIs2(i, :) - tCIs1(i, :), 2)
end
```

```
ans = 7.1445
ans = 4.1795
ans = 2.5025
```

```
% fraction of intervals that contain the true value, for the various sizes n
mean(tCIs1 < true_m & true_m < tCIs2, 2)  % mean over the second dimension (axis)
```

```
ans = 3×1
    0.6290
    0.6690
    0.6990
```

The sampling distributions of the mean are skewed to the left for all n and the empirical CIs are strongly asymmetric.

The size of the t-CIs are different from that of the empirical CIs for all n.

The fractions of intervals containing the true value are smaller that 0.95 (the expected number for a 95% confidence level) even for very large n (try $n = 100000$).

The t-CI is not appropriate because the Pareto distribution has an infinite variance when $\theta \leq 2$, hence one of the assumptions of the Central Limit Theorem does not hold and the sampling distribution of the mean does not converge to a normal distribution for large n. (Choose a $\theta > 2$ and check that for very large n the sampling distribution of the mean is approximately Normal).

5. Bootstrapping

bootstrap-CI for the mean of a Normal RV

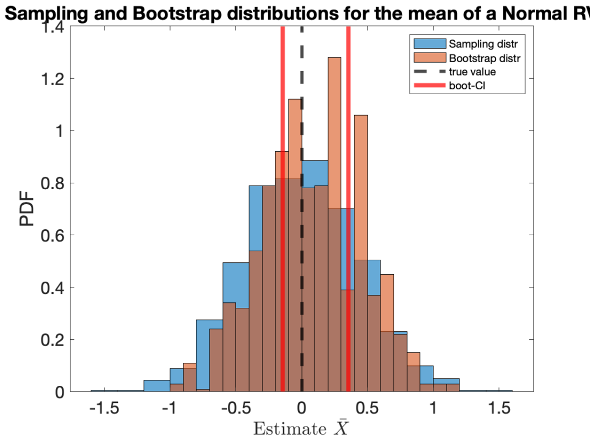
1. Plot the empirical sampling distribution for the mean of a Normal RV with $\mu = 0$ and $\sigma = 1$ (see Exercise 1) generating $S = 1000$ samples of size $n = 5$.
2. Plot the bootstrap distribution obtained generating $S = 1000$ bootstrap samples, where each sample is generated by drawing n elements with replacement from an original sample of size n
3. Compare the sampling and bootstrap distributions: do they have the same mean? do they have the same spread? Do your answers change if you vary n?
4. Compute the bootstrap-CI for a 50% confidence level and add 3 vertical lines (one for the true value and two for the boot-CI) to the plot with the sampling and bootstrap distribution.
5. Repeat steps 1-4 a few times and count the fraction of boot-CI that don't contain the true value: are they rhoughly 50%?
6. Write a for loop that repeats steps 2-4 1000 times for a confidence level of 95%: (a) do boot-CIs and empirical CIs have similar length? (b) How many boot-CIs contain the true value? What happens when n increases?

Solution

```
% Normal RV
mu = 0;
sigma = 1;
true_m = mu;

% observed sample
n = 5;
d = normrnd(mu, sigma, 1, n);

S = 1000;
estimates = zeros(1, S);
bootstrap_ests = zeros(1, S);
for i = 1 : S
    % sampling distribution
    x = normrnd(mu, sigma, 1, n);  % this is a new sample from the Normal distrib
    estimates(i) = mean(x);

    % bootstrap distribution
    x = datasample(d, n);  % this is sampled with replacement from the original data
    bootstrap_ests(i) = mean(x);
end

% compute the boot-CI
alpha = 0.5;
bCI = quantile(bootstrap_ests, [alpha/2 1-alpha/2])
```

```
bCI = 1×2
   -0.1459    0.3554
```

```
% plot
clf
histogram(estimates, 'normalization', 'pdf')
hold on
histogram(bootstrap_ests, 'normalization', 'pdf')
hold on

xline(true_m, '--k', 'Linewidth', 3);
hold on
xline(bCI(1), '-r', 'Linewidth', 4);
hold on
xline(bCI(2), '-r', 'Linewidth', 4);
title('Sampling and Bootstrap distributions for the mean of a Normal RV') %, 'Interpreter', 'latex')
legend('Sampling distr', 'Bootstrap distr', 'true value', 'boot-CI', 'FontSize', 10.0);
xlabel('Estimate $\bar{X}$', 'Interpreter', 'latex');
ylabel('PDF');
set(gca, 'FontSize', 16.0);
```



The shape of the bootstrap distribution is similar to the shape of the sampling distribution, especially for large n.

The "spread" (standard deviations) of the two distributions are similar, while the means can be different.

```
alpha = 0.05;

R = 1000;
bCIs = zeros(R, 2);
for j = 1:R
    % observed sample
    n = 5;
    d = normrnd(mu, sigma, 1, n);

    S = 1000;
    bootstrap_ests = zeros(1, S);
    for i = 1 : S
        % bootstrap distribution
        x = datasample(d, n);  % this is sampled with replacement from the original data
        bootstrap_ests(i) = mean(x);
    end
    % compute the boot-CI
    bCIs(j, :) = quantile(bootstrap_ests, [alpha/2 1-alpha/2]);
end

% average size of b-CIs
mean(bCIs(:, 2) - bCIs(:, 1))
```

```
ans = 1.4460
```

```
% fraction of intervals that contain the true mean, for the various sizes n
mean(bCIs(:, 1) < true_m & true_m < bCIs(:, 2), 1)  % mean over the second dimension (axis)
```

```
ans = 0.8460
```

Average length of b-CIs gets closer to that of empirical CI (from Exercise 1) as n increases.

The fraction of b-CIs containing the true value gets closer to the desired confidence level as n increases.

bootstrap-CI for the mean of a Pareto RV

1. Plot the empirical sampling distribution for the mean of a Pareto RV with $\theta = 1.5$ (see Exercise 4) generating $S = 1000$ samples of size $n = 5$.
2. Plot the bootstrap distribution obtained generating $S = 1000$ bootstrap samples, where each sample is generated by drawing n elements with replacement from an original sample of size n (use Matlab's `datasample`).
3. Compare the sampling and bootstrap distributions: do they have the same mean? do they have the same spread? Do your answers change if you vary n?
4. Compute the bootstrap-CI for a 50% confidence level and add 3 vertical lines (one for the true value and two for the boot-CI) to the plot with the sampling and bootstrap distribution.
5. Repeat steps 1-4 a few times and count the fraction of boot-CI that don't contain the true value: are they rhoughly 50%?
6. Write a for loop that repeats steps 2-4 1000 times for a confidence level of 95%: How many boot-CIs contain the true value? What happens when n increases? How is the performance of b-CIs compared to that of t-CIs (see Exercise 4)?

Matlab's function for bootstrapping is `bootci` : the code below computes the b-CIs using the percentiles of the bootstrap distribution (`d` is the original sample):

`parameter = @(y) mean(y);`

`[ci, bootstrap_ests] = bootci(S, {parameter, d}, 'alpha', alpha, 'type', 'percentile');`

(check that `ci` is equal to `quantile(bootstrap_ests, [alpha/2 1-alpha/2])`)

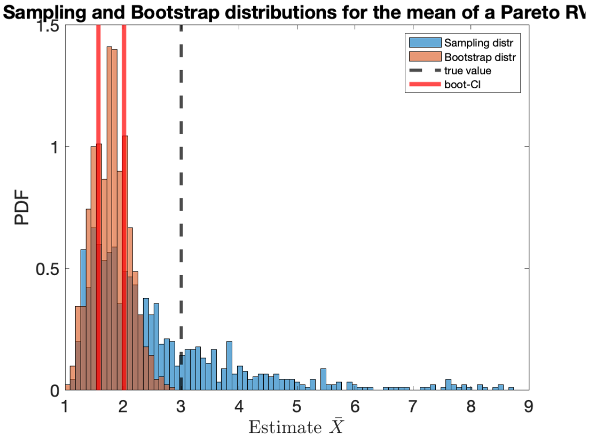
Solution

```
% Pareto RV
theta = 1.5;
true_m = theta / (theta - 1);

% observed sample
n = 5;
d = rand(1, n) .^ (-1 / theta);

S = 1000;
estimates = zeros(1, S);
bootstrap_ests = zeros(1, S);
for i = 1 : S
    % sampling distribution
    x = rand(1, n) .^ (-1 / theta);  % this is a new sample from the Pareto distrib
    estimates(i) = mean(x);

    % bootstrap distribution
    x = datasample(d, n);  % this is sampled with replacement from the original data
    bootstrap_ests(i) = mean(x);
end

% compute the boot-CI
alpha = 0.5;
bCI = quantile(bootstrap_ests, [alpha/2 1-alpha/2])
```

```
bCI = 1×2
```

```
    1.5803    2.0196
```

```
% plot
clf
histogram(estimates, 1:true_m*3/100:true_m*3, 'normalization', 'pdf')
hold on
histogram(bootstrap_ests, 1:true_m*3/100:true_m*3, 'normalization', 'pdf')
hold on

xline(true_m, '--k', 'Linewidth', 3);
hold on
xline(bCI(1), '-r', 'Linewidth', 4);
hold on
xline(bCI(2), '-r', 'Linewidth', 4);
title('Sampling and Bootstrap distributions for the mean of a Pareto RV') %, 'Interpreter', 'latex')
legend('Sampling distr', 'Bootstrap distr', 'true value', 'boot-CI', 'FontSize', 10.0);
xlabel('Estimate $\bar{X}$', 'Interpreter', 'latex');
ylabel('PDF');
set(gca, 'FontSize', 16.0);
xlim([1 true_m*3])
```



The shape of the bootstrap distribution is not similar to the shape of the sampling distribution, and even for large n they can be different.

```
alpha = 0.05;
parameter = @(y) mean(y);

R = 1000;
bCIs = zeros(R, 2);
for j = 1:R
    % observed sample
    n = 5;
    d = rand(1, n) .^ (-1 / theta);

    S = 1000;
    ci = bootci(S, {parameter, d}, 'alpha', alpha, 'type', 'percentile');
    bCIs(j, :) = ci;
end

% average size of b-CIs
mean(bCIs(:, 2) - bCIs(:, 1))
```

```
ans = 4.9954
```

```
% fraction of intervals that contain the true mean, for the various sizes n
mean(bCIs(:, 1) < true_m & true_m < bCIs(:, 2), 1)  % mean over the second dimension (axis)
```

```
ans = 0.5230
```

The fraction of b-CIs containing the true value is much smaller than 0.95 and it is simlar to the fraction of t-CIs (see Exercise 4).

(Other bootstrap-based methods compute CIs that are singnificantly more accurate than t-CIs, especially for skewed distributions: try `bootci` with option `'type', 'bca' or 'student'`.)

bootstrap-CI for the standard deviation of a population

Use the bootstrap percentile CI to estimate the standard deviation of the student heights at 95% confidence level.

The measured heights of random students are:

```
d = [176, 165, 189, 180, 172, 169, 162, 161, 183, 170];
```

Solution

```
% observed sample
d = [176, 165, 189, 180, 172, 169, 162, 161, 183, 170];
n = length(d);

s = std(d)
```

```
s = 9.2382
```

```
S = 10000;
estimates = zeros(1, S);
bootstrap_ests = zeros(1, S);
for i = 1 : S
%     % sampling distribution
%     x = normrnd(mu, sigma, 1, n);  % this is a new sample from the Normal distrib
%     estimates(i) = mean(x);

    % bootstrap distribution
    x = datasample(d, n);  % this is sampled with replacement from the original data
    bootstrap_ests(i) = std(x);
end

% compute the boot-CI
```

```
alpha = 0.05;
bCI = quantile(bootstrap_ests, [alpha/2 1-alpha/2])
```

```
bCI = 1×2
    5.2988   11.5031
```

```
% plot
clf
% histogram(estimates, 'normalization', 'pdf')
% hold on
histogram(bootstrap_ests, 'normalization', 'pdf')
hold on

xline(s, '--k', 'Linewidth', 3);
hold on
xline(bCI(1), '-r', 'Linewidth', 4);
hold on
xline(bCI(2), '-r', 'Linewidth', 4);
title('Bootstrap distributions for the mean of a Normal RV') %, 'Interpreter', 'latex')
legend('Bootstrap distr', 'true value', 'boot-CI', 'FontSize', 16.0);
xlabel('Estimate $\bar{X} = 9.24$', 'Interpreter', 'latex');
ylabel('PDF');
set(gca, 'FontSize', 16.0);

title('Bootstrap distribution for the stdev of a Normal RV') %, 'Interpreter', 'latex')
legend('Bootstrap distr.', '$\hat{s}^* = 9.24$', '95\% boot-CIs', 'FontSize', 16.0, 'Interpreter', 'latex');
xlabel('Estimate $s$', 'Interpreter', 'latex');
ylabel('PDF');
set(gca, 'FontSize', 16.0);
```