Investigate the distributions of the mean and variance of a normal variable.

For a random normal variable, X, with 0 mean and unit variance, investigate:

- the distribution of the mean: find the Monte Carlo average and confidence intervals,
- the distribution of the variance: find the Monte Carlo average and confidence intervals,
- consider various sample sizes, N = 10,50,100,500 ,
- consider various replications, R = 100,500,1000 .

Simulation settings

```
N    = 50;     % sample size
xmu  = 0;      % mean  (0 basic)
xvar = 1;      % variance (note: std = sqrt(xvar))  (1 basic)
R    = 100;   % number of Monte Carlo replications
```

Preallocate storage for estimates (Mean, Variance)

```
MCest = nan(R, 2);
```

(Optional) If you want named columns, use a table instead:

MCestTbl = table(nan(R,1), nan(R,1), 'VariableNames', {'Mean','Variance'});

Monte Carlo simulation

```
% Simulate all data at once: each column = one replication
X = normrnd(xmu, sqrt(xvar), [N, R])
```

```

% Compute the statistics of interest across rows (dimension 1)
MCmean = mean(X, 1);          % mean of each column
MCvar  = var(X, 1, 1);        % population variance (set 0 for sample variance)

% Combine results
MCest = [MCmean.' MCvar.'];

% (Optional) Display a preview
disp(array2table(MCest(1:5,:), 'VariableNames', {'Mean','Variance'}))
```

```matlab
%% Compare the histograms
figure;

subplot(1,2,1)
histogram(MCest(:,1))
title('Mean')

subplot(1,2,2)
histogram(MCest(:,2))
title('Variance')

% Restore default layout (optional)
set(gcf, 'Position', [100 100 800 400]) % nicer side-by-side view
```

```matlab
%% Compare the densities
figure;

% --- Density for the Mean ---
subplot(1,2,1)
[f, xi] = ksdensity(MCest(:,1));              % kernel density estimate
plot(xi, f, 'LineWidth', 2)
hold on
% Normal curve with same mean & sd
mu1 = mean(MCest(:,1));
sd1 = std(MCest(:,1));
y_norm = normpdf(xi, mu1, sd1);
plot(xi, y_norm, '--b', 'LineWidth', 2)
hold off
title('MC Density Estimate (Mean)')
legend('Kernel Density', 'Normal Fit')



% --- Density for the Variance ---
subplot(1,2,2)
[f, xi] = ksdensity(MCest(:,2));
plot(xi, f, 'LineWidth', 2)
hold on
mu2 = mean(MCest(:,2));
sd2 = std(MCest(:,2));
y_norm = normpdf(xi, mu2, sd2);
plot(xi, y_norm, '--b', 'LineWidth', 2)
hold off
```

```
title('MC Density Estimate (Variance)')
legend('Kernel Density', 'Normal Fit')

% Optional: adjust layout
set(gcf, 'Position', [100 100 1000 400])
```

```
%% Compare the empirical CDFs
figure;

% --- ECDF for the Mean ---
subplot(1,2,1)
[f, x] = ecdf(MCest(:,1));                     % empirical CDF
plot(x, f, 'r', 'LineWidth', 2)
hold on
% Theoretical normal CDF
x_theor = linspace(min(x), max(x), 200);
F_theor = normcdf(x_theor, mean(MCest(:,1)), std(MCest(:,1)));
plot(x_theor, F_theor, '--b', 'LineWidth', 2)
hold off
title('Empirical CDF - Mean')
legend('Empirical CDF', 'Normal CDF', 'Location', 'best')

% --- ECDF for the Variance ---
subplot(1,2,2)
[f, x] = ecdf(MCest(:,2));
plot(x, f, 'r', 'LineWidth', 2)
hold on
x_theor = linspace(min(x), max(x), 200);
F_theor = normcdf(x_theor, mean(MCest(:,2)), std(MCest(:,2)));
plot(x_theor, F_theor, '--b', 'LineWidth', 2)
hold off
title('Empirical CDF - Variance')
legend('Empirical CDF', 'Normal CDF', 'Location', 'best')

% Optional: adjust figure size
set(gcf, 'Position', [100 100 1000 400])
```

Compute the **95% confidence intervals** (based on quantiles of the Monte Carlo estimates)

```
%% 95% Monte Carlo confidence intervals
alpha = 0.05;
```

```
% For the mean estimates
CI_mean = quantile(MCest(:,1), [alpha/2, 1 - alpha/2]);

% For the variance estimates
CI_var  = quantile(MCest(:,2), [alpha/2, 1 - alpha/2]);

% Display results
fprintf('95%% CI for Mean: [%.6f, %.6f]\n', CI_mean(1), CI_mean(2));
fprintf('95%% CI for Variance: [%.6f, %.6f]\n', CI_var(1), CI_var(2));
```

- **OLS**: purely geometric — fits the best line by minimizing vertical distances (squared).
- **MLE**: probabilistic — finds parameters that make the data most likely under a statistical model.

Consider a random normal variable $X \sim \mathcal{N}(0, 1)$.

We investigate the distributions of the sample mean and sample variance under different sample sizes and numbers of Monte Carlo replications.

Effect of Sample Size on the mean and varian

- The sample mean $\bar{X}$ is unbiased: $\mathbb{E}[\bar{X}] = 0$.
- Its standard deviation is $\mathrm{SD}(\bar{X}) = \dfrac{1}{\sqrt{N}}$.
- Therefore, the 95% confidence interval for the mean shrinks as N increases:

$$\text{half-width} \approx \frac{1.96}{\sqrt{N}}.$$

N	10	50	100	500
Half-width	0.62	0.28	0.20	0.09

For the unbiased estimator

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2,$$

we have $\mathbb{E}[s^2] = 1$.

For the **MLE estimator**

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(X_i - \bar{X})^2,$$

we have $\mathbb{E}[\hat{\sigma}^2] = \frac{N-1}{N}$, which is slightly biased downward.

The approximate standard error of s^2 is

$$\mathrm{SE}(s^2) \approx \sqrt{\frac{2}{N-1}}.$$

Example values:

N	10	50	100	500
$\mathrm{SE}(s^2)$	0.47	0.20	0.14	0.06

Confidence intervals for a variance are asymmetric (based on the chi-square distribution), but they become clearly tighter as N increases.

Larger N makes both the sample mean and variance estimates more stable, and their confidence intervals narrower.

For small N, the estimates fluctuate more, and the MLE variance is slightly biased low.

Effect of Number of Replications

- Changing R does not alter the statistical properties of the estimators.
- It only affects how precisely the Monte Carlo average or confidence interval is estimated.
- The Monte Carlo standard error of an estimated quantity decreases as $1/\sqrt{R}$.
- For example, the standard deviation of the Monte Carlo average of the sample mean is

$$\mathrm{SD}\left(\bar{\bar{X}}\right) = \frac{1}{\sqrt{NR}}.$$

Increasing R reduces simulation noise in the Monte Carlo estimates (making them smoother and closer to theoretical values), but it does not change the underlying variability determined by N.

Parameter	Effect
$N \uparrow$	Smaller sampling variability; narrower CIs; less bias
$R \uparrow$	More stable Monte Carlo summaries; smoother results

Do as above for a random normal variable, X, with mean equal to 5 and variance equal to 20. Has anything changed?

Simulation settings

```
N    = 50;     % sample size
xmu  = 5;      % mean  (0 basic)
xvar = 20;      % variance (note: std = sqrt(xvar))  (1 basic)
R    = 2000;   % number of Monte Carlo replications
```

Preallocate storage for estimates (Mean, Variance)

```
MCest = nan(R, 2);
```

(Optional) If you want named columns, use a table instead:

MCestTbl = table(nan(R,1), nan(R,1), 'VariableNames', {'Mean','Variance'});

Monte Carlo simulation

```
% Simulate all data at once: each column = one replication
X = normrnd(xmu, sqrt(xvar), [N, R]);

% Compute the statistics of interest across rows (dimension 1)
MCmean = mean(X, 1);          % mean of each column
MCvar  = var(X, 1, 1);        % population variance (set 0 for sample variance)
```

```
% Combine results
MCest = [MCmean.' MCvar.'];

% (Optional) Display a preview
disp(array2table(MCest(1:5,:), 'VariableNames', {'Mean','Variance'}))



%% Compare the histograms
figure;

subplot(1,2,1)
histogram(MCest(:,1))
title('Mean')

subplot(1,2,2)
histogram(MCest(:,2))
title('Variance')

% Restore default layout (optional)
set(gcf, 'Position', [100 100 800 400]) % nicer side-by-side view



%% Compare the densities
figure;

% --- Density for the Mean ---
subplot(1,2,1)
[f, xi] = ksdensity(MCest(:,1));              % kernel density estimate
plot(xi, f, 'LineWidth', 2)
hold on
% Normal curve with same mean & sd
mu1 = mean(MCest(:,1));
sd1 = std(MCest(:,1));
y_norm = normpdf(xi, mu1, sd1);
plot(xi, y_norm, '--b', 'LineWidth', 2)
hold off
title('MC Density Estimate for the Mean')
legend('Kernel Density', 'Normal Fit')





% --- Density for the Variance ---
subplot(1,2,2)
[f, xi] = ksdensity(MCest(:,2));
plot(xi, f, 'LineWidth', 2)
```

```
hold on
mu2 = mean(MCest(:,2));
sd2 = std(MCest(:,2));
y_norm = normpdf(xi, mu2, sd2);
plot(xi, y_norm, '--b', 'LineWidth', 2)
hold off
title('MC Density Estimate for the Variance')
legend('Kernel Density', 'Normal Fit')

% Optional: adjust layout
set(gcf, 'Position', [100 100 1000 400])



%% Compare the empirical CDFs
figure;

% --- ECDF for the Mean ---
subplot(1,2,1)
[f, x] = ecdf(MCest(:,1));                     % empirical CDF
plot(x, f, 'r', 'LineWidth', 2)
hold on
% Theoretical normal CDF
x_theor = linspace(min(x), max(x), 200);
F_theor = normcdf(x_theor, mean(MCest(:,1)), std(MCest(:,1)));
plot(x_theor, F_theor, '--b', 'LineWidth', 2)
hold off
title('Empirical CDF - Mean')
legend('Empirical CDF', 'Normal CDF', 'Location', 'best')

% --- ECDF for the Variance ---
subplot(1,2,2)
[f, x] = ecdf(MCest(:,2));
plot(x, f, 'r', 'LineWidth', 2)
hold on
x_theor = linspace(min(x), max(x), 200);
F_theor = normcdf(x_theor, mean(MCest(:,2)), std(MCest(:,2)));
plot(x_theor, F_theor, '--b', 'LineWidth', 2)
hold off
title('Empirical CDF - Variance')
legend('Empirical CDF', 'Normal CDF', 'Location', 'best')

% Optional: adjust figure size
set(gcf, 'Position', [100 100 1000 400])
```

Compute the **95% confidence intervals** (based on quantiles of the Monte Carlo estimates)

```
%% 95% Monte Carlo confidence intervals
alpha = 0.05;

% For the mean estimates
CI_mean = quantile(MCest(:,1), [alpha/2, 1 - alpha/2]);

% For the variance estimates
CI_var  = quantile(MCest(:,2), [alpha/2, 1 - alpha/2]);

% Display results
fprintf('95%% CI for Mean: [%.6f, %.6f]\n', CI_mean(1), CI_mean(2));
fprintf('95%% CI for Variance: [%.6f, %.6f]\n', CI_var(1), CI_var(2));
```

When repeating the Monte Carlo simulation for a normal variable with mean 5 and variance 20, the **shape of the sampling distributions** for the sample mean and variance remain the same — they are still approximately normal and positively skewed, respectively.

However, their **locations and scales** change:

- The sampling distribution of the mean is now centered around 5 (instead of 0).
- The sampling distribution of the variance is centered around 20 (instead of 1).
- The **spread** of the sample mean estimates increases, because the variance of the mean estimator depends on the underlying population variance:

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

Therefore, nothing changes in principle — the logic and shape of the distributions are the same — but the numerical **scale** and **location** of the results reflect the new parameters.