Large Sample Approximations

Sujit Sandipan Chaugule[1*], Dr. Amiya Ranjan Bhowmick[2]

[1*]Department of Pharmaceutical Sciences and Technology, Institute of Chemical Technology, Mumbai

[2]Department of Mathematics, Institute of Chemical Technology, Mumbai

Consistent estimator

We say that a sequence of estimators W_n is a consistent estimator for $\theta \in \Theta$ if for every $\epsilon > 0$, and for every $\theta \in \Theta$, the following holds:

$$\lim_{n\to\infty} P_\theta(|W_n - \theta| < \epsilon) = 1.$$

The statement can also be equivalently written as:

$$\lim_{n\to\infty} P_\theta(|W_n - \theta| \geq \epsilon) = 0$$

and by the Chebychev's inequality, we have:

$$P_\theta(|W_n - \theta| < \epsilon) \leq \frac{\mathbb{E}_\theta(W_n - \theta)^2}{\epsilon^2}.$$

Therefore, a sufficient condition for an estimator W_n to be a consistent estimator for θ is to test whether:

$$\mathbb{E}_\theta(W_n - \theta)^2 \to 0 \quad \text{as } n \to \infty \quad \text{for every } \theta \in \Theta.$$

In addition, we have the following well-known decomposition, given by:

$$\mathbb{E}_\theta(W_n - \theta)^2 = \mathrm{Var}_\theta(W_n) + (\mathrm{Bias}_\theta W_n)^2.$$

Therefore, we have the following theorem:

Characterization of consistent estimator (Casella and Berger 2002)

If W_n is a sequence of estimators of a parameter θ satisfying the following conditions:

- $\lim_{n\to\infty} \mathrm{Var}_\theta(W_n) = 0$
- $\lim_{n\to\infty} \mathrm{Bias}_\theta(W_n) = 0$

for every $\theta \in \Theta$, then W_n is a sequence of consistent estimators of θ.

It is important to note that the sequence of estimators must have finite variance, which is not a necessary condition to be consistent. One can construct a different sequence of consistent estimators as well by virtue of the following theorem:

Many consistent estimators

If $W_n!$ is a consistent sequence of estimators of a parameter θ, let (a_n) and (b_n) be sequences of real numbers satisfying:

- $\lim_{n\to\infty} a_n = 1$
- $\lim_{n\to\infty} b_n = 0.$

Then the sequence

$$U_n = a_n W_n + b_n$$

is a consistent sequence of estimators of θ.

Using the software, we can demonstrate that:

$$P\left(\left|\overline{X}_n - \beta\right| < \epsilon\right) \to 0$$

as $n \to \infty$ for every choice of $\epsilon > 0$ and for all $\beta \in (0, \infty)$.

In the following code, we compute the above probability, which is basically the integration:

$$\int_{n(\beta-\epsilon)}^{n(\beta+\epsilon)} \frac{e^{-\frac{y}{\beta}} y^{n-1}}{\Gamma(n)\beta^n} I_{(0,\infty)}(y)\, dy.$$

In [1]:
```
using Plots, Distributions, LaTeXStrings # Load the package
```

In [2]:
```
n = 10
beta = 3.0
n_vals = 1:5000
eps = 0.1
```

Out[2]:
```
0.1
```

In [3]:
```
prob_vals= zeros(length(n_vals)) # store the values
for n in n_vals
    prob_vals[n] = cdf(Gamma(n, beta), n * (beta + eps)) -
    cdf(Gamma(n, beta), n * (beta - eps))
end
```

In [4]:
```
plot(n_vals, prob_vals, lw = 2, title = "P(|x̄ₙ - β| < ε)", label = "",
    xlabel = "n_vals",
    ylabel = "prob_vals")
annotate!(4500, 0.95, text(L"\epsilon = 0.1", 12))
```

Out[4]:

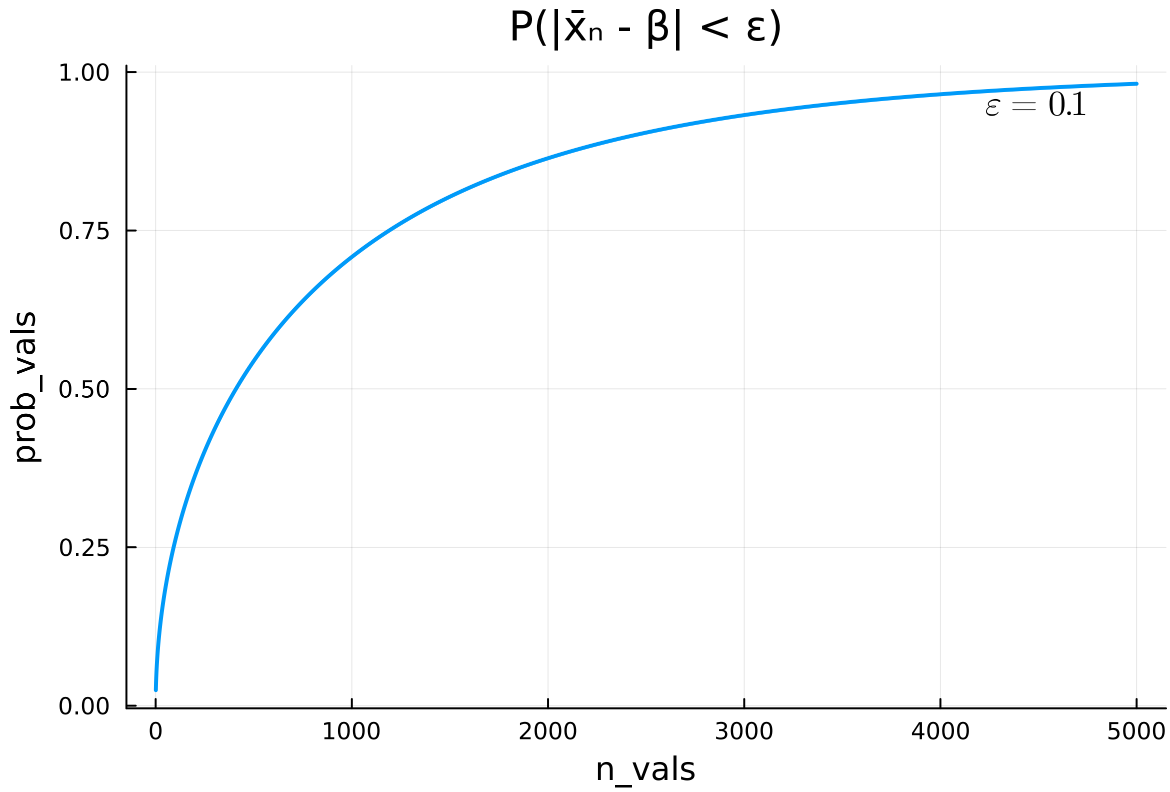

Figure 1: As the sample size increases, the probability converges to 1. Students are encouraged to experiment with different choices of β and ϵ

In [5]:
```
using SpecialFunctions, QuadGK
```

```
WARNING: using SpecialFunctions.beta in module Main conflicts with an existing identif
ier.
```

In [6]:
```
# set the parameters
n = 100
beta = 3.0    # beta
eps = 0.1
```

Out[6]: 0.1

In [7]:
```
# Define the function
f(x) = ((exp(-x/beta)*x^(n-1))/(gamma(n)*beta^n))*(x>0)
```

Out[7]: f (generic function with 1 method)

In [8]:
```
quadgk(f,n*(beta-eps), n*(beta+eps))  # numerical integration
```

Out[8]: (0.26099142618227217, 5.551115123125783e-17)

2. Large Sample Approximation of Variance of Estimators

Limiting Variance

For an estimator T_n, if:

$$\lim_{n\to\infty} k_n \mathrm{Var}(T_n) = \tau^2 < \infty,$$

where $\{k_n\}$ is a sequence of constants, then τ^2 is called the **limiting variance** or **limit of variances**.

In [9]:

```
using Plots, Distributions, Statistics, StatsPlots, LaTeXStrings # required packages
```

In [10]:

```
n_vals = [3, 5, 10, 25, 50, 100]
mu = 2 # true values
rep = 1000 # no of replications
sigma = 0.5 # population sd
```

Out[10]:

```
0.5
```

In [11]:

```
gr()  # Set the plotting backend to GR

# Create subplots
fig = plot(layout = (2, 3), size = (900, 600))


for (idx, n) in enumerate(n_vals)
    sample_means = zeros(rep)
    t_n = zeros(rep)
    for i in 1:rep
        x = rand(Normal(mu, sigma), n)
        sample_means[i] = mean(x)
        t_n[i] = 1 / mean(x)
    end
    hist = histogram!(t_n, bins = 30, normalize = true, color = :lightblue,
                      linecolor = :black, label = "", subplot = idx)
    scatter!([1 / mu], [0], color = :red, markersize = 8, label = "",
        subplot = idx)
    xlabel!(L"t_n", subplot = idx)
    ylabel!("Density", subplot = idx)
    title!(fig[idx], "n = $n")
end

display(fig) # display the plot
```

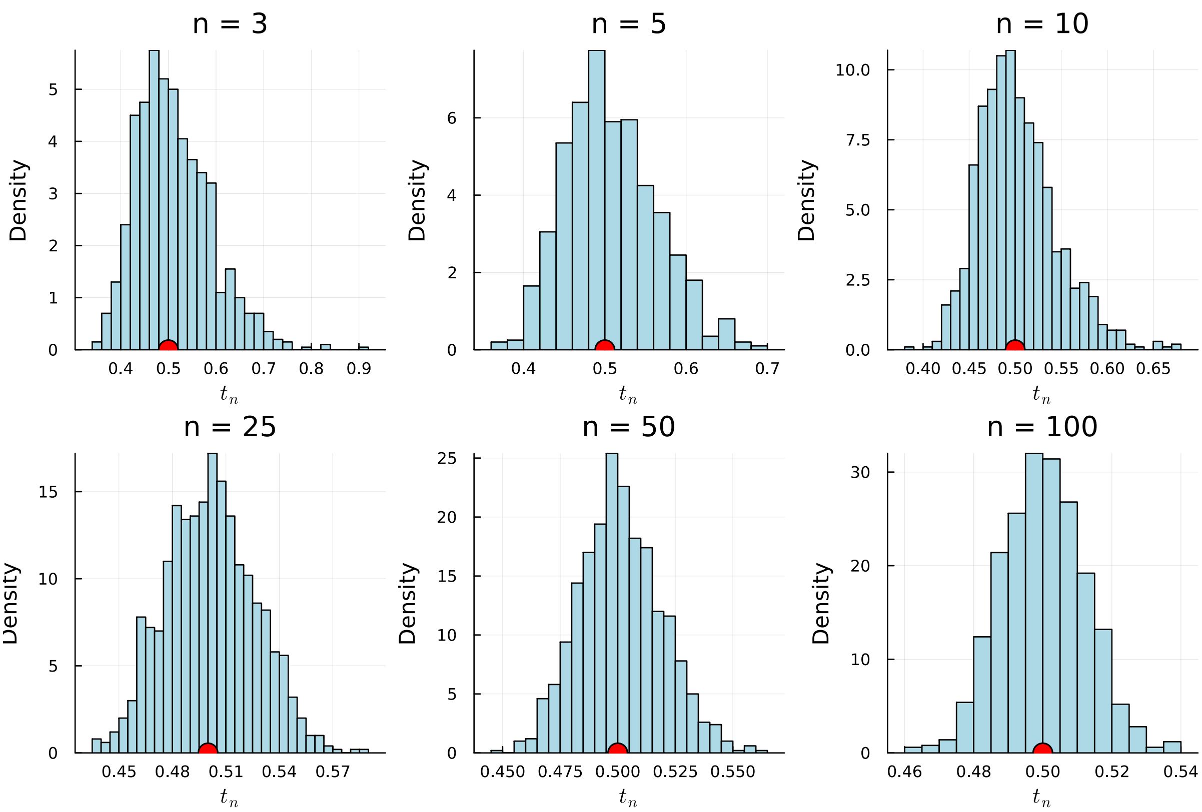

Figure 2: The sampling distribution of $1/\bar{X}_n$ is visualized using the histograms based on 1000 simulations for different sample size As the sample size increases, the sampling distribution gets highly concentrated about the value $1/\mu$

In [12]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings
```

In [13]:
```
mu = 2
sigma = 0.5
n_vals = 1:1000
t_n =  zeros(length(n_vals))
sample_mean = zeros(length(n_vals))

for n in n_vals
    x = rand(Normal(mu, sigma), n)
    sample_mean[n] = mean(x)
    t_n[n] =  1/mean(x)
end
```

In [14]:
```
p1 = scatter(n_vals, sample_mean, color = "grey", label = "",
     xlabel = "sample size(n)", title = L"\bar{X_n}" )
hline!([mu], color = "red", linestyle = :dash, lw = 3, label = "" )

p2 = scatter(n_vals, t_n, color = "grey", label = "",
     xlabel = "sample size(n)", title = L"T_n = 1/\bar{X_n}")
hline!([1/mu], color = "red", linestyle = :dash, lw = 3, label = "" )

plot(p1, p2 , layout= (1,2))
```

Out[14]:

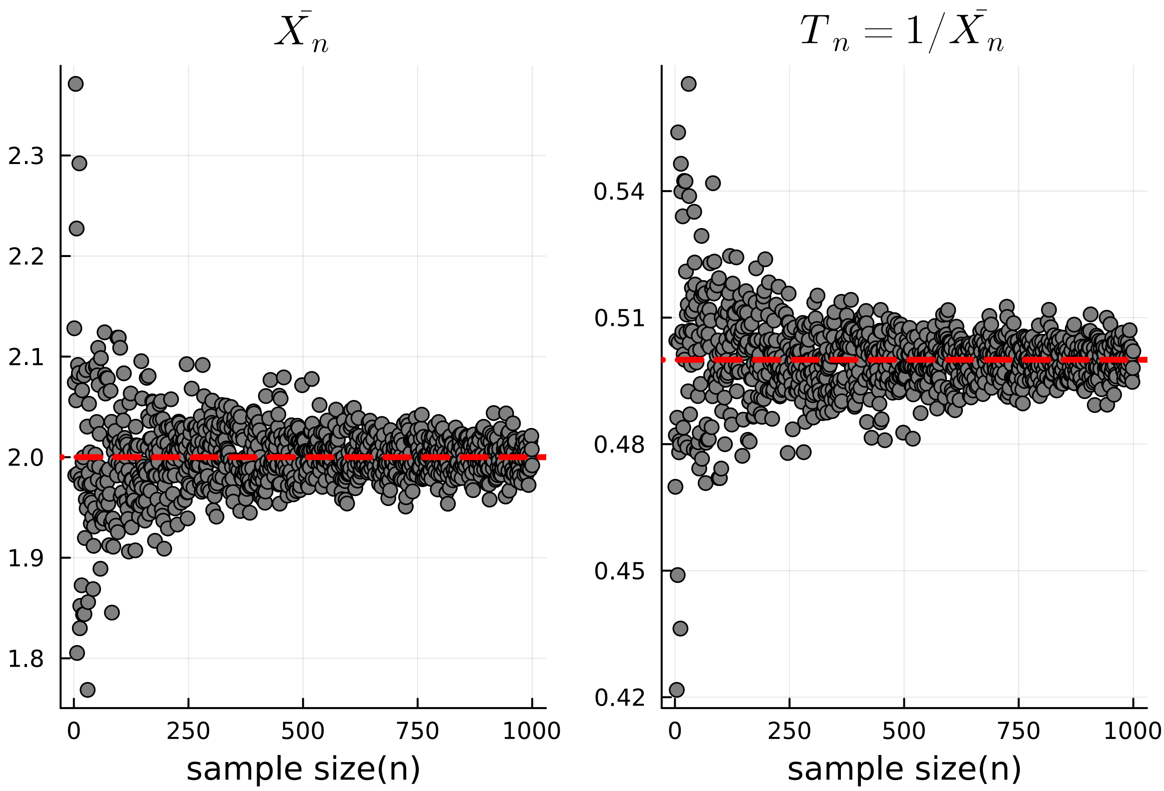

Figure 3: For large n, as $\bar{X}_n$ values become close to μ, then $1/\bar{X}_n$ values get closer to $1/\mu$ for $\mu \neq 0$. In fact, it shows that as $n \to \infty$, $1/\bar{X}_n \to 1/\mu$.

Instead of the sample mean, one may also aim to estimate $1/\mu$ using the inverse of the sample median. For large n, the approximations may be compared if we can compute the limiting variances of the inverse of the sample median. Before going into any mathematical computations, first let us check how the estimator based on the sample median behaves for large n values.

In [15]:
```
using Plots, Statistics, Random, Distributions, LaTeXStrings
```

In [16]:
```
mu = 2
sigma = 0.5
n_vals = 1:1000
t_n =  zeros(length(n_vals))
sample_median = zeros(length(n_vals))

for n in n_vals
    x = rand(Normal(mu, sigma), n)
    sample_median[n] = median(x)
    t_n[n] =  1/median(x)
end
```

In [17]:
```
p1 = scatter(n_vals, sample_median, color = "grey", label = "",
     xlabel = "sample size(n)", title = L"Med({X_n})" )
hline!([mu], color = "red", linestyle = :dash, lw = 3, label = "" )

p2 = scatter(n_vals, t_n, color = "grey", label = "",
    xlabel = "sample size(n)", title = L"T_n = 1/Med({X_n})")
hline!([1/mu], color = "red", linestyle = :dash, lw = 3, label = "" )

plot(p1, p2 , layout= (1,2))
```

Out[17]:

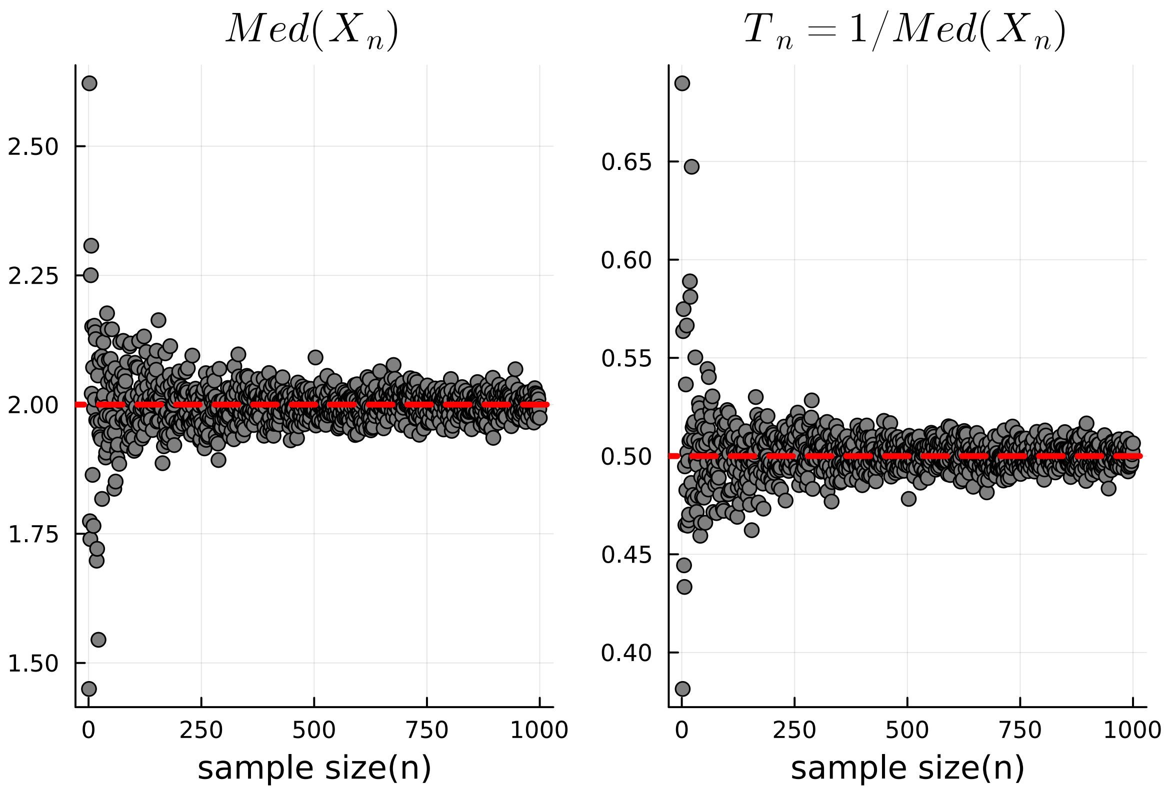

Figure 4: The inverse of the sample median also appears to be a consistent estimator for $1/\mu$

Now let us check, how the sampling distribution of the estimator $1/\mu$ of behaves for large values $1/Med(X_n)$ behaves for large n values

In [18]:
```
using Plots, Distributions, Statistics, StatsPlots, LaTeXStrings # Load the required
```

In [19]:

```
n_vals = [3, 5, 10, 25, 50, 100]
mu = 2 # true values
rep = 1000 # no of replications
sigma = 0.5 # population sd
```

Out[19]:

```
0.5
```

In [20]:

```
gr()  # Set the plotting backend to GR

# Create subplots
fig = plot(layout = (2, 3), size = (900, 600))


for (idx, n) in enumerate(n_vals)
    sample_median = zeros(rep)
    t_n = zeros(rep)
    for i in 1:rep
        x = rand(Normal(mu, sigma), n)
        sample_median[i] = median(x)
        t_n[i] = 1 / median(x)
    end
    hist = histogram!(t_n, bins = 30, normalize = true, color = :lightblue,
                      linecolor = :black, label = "", subplot = idx)
    scatter!([1 / mu], [0], color = :red, markersize = 8, label = "",
        subplot = idx)
    xlabel!(L"t_n", subplot = idx)
    ylabel!("Density", subplot = idx)
    title!(fig[idx], "n = $n")
end

display(fig) # display the plot
```

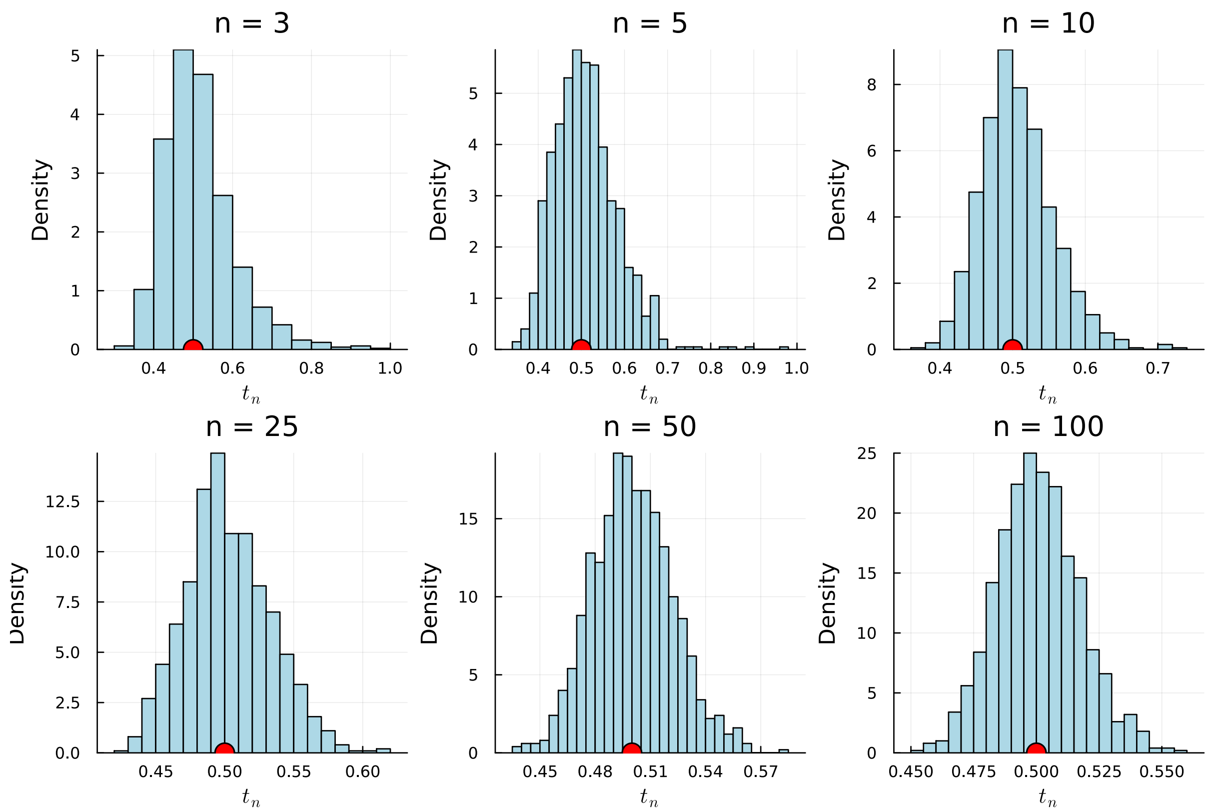

Figure 5: The histograms represents the approximate sampling distribution of the sample median, where the population distribution is normal with mean $\mu = 2$ and variance $\sigma^2 = 0.5$

For the above simulation experiments, it appears that both inverse of the sample mean and the sample median appears to be a nice choice and both are approximately normally distribution for large .Let us now compare the inverse of the sample mean and sample median with respect to their asymptotic variances. We basically obtain the sampling distribution of the following two random variables for large value

$$\sqrt{n}\left(\frac{1}{\bar{X}_n} - \frac{1}{\mu}\right)$$

and

$$\sqrt{n}\left(\frac{1}{\text{Med}(X_n)} - \frac{1}{\mu}\right)$$

In [21]:

```
using Plots, Statistics, Distributions, LaTeXStrings, StatsBase, KernelDensity
```

In [22]:

```
# Parameters
n_vals = [3, 5, 10, 25, 50, 100]
mu = 2
rep = 1000
sigma = 0.5

fig = plot(layout=(2, 3), size=(900, 600)) # set the figure layout

for (idx, n) in enumerate(n_vals)
    t_n = zeros(rep) # store the values
    w_n = zeros(rep) # store the values
    for i in 1:rep
        x = rand(Normal(mu, sigma), n)
        t_n[i] = sqrt(n) * (1 / mean(x) - 1 / mu)
        w_n[i] = sqrt(n) * (1 / median(x) - 1 / mu)
    end

    # Kernel density estimation
    density_t_n = kde(t_n)
    density_w_n = kde(w_n)

    # Define x-range for plotting
    x_range = range(minimum([minimum(t_n), minimum(w_n)]),
                    maximum([maximum(t_n), maximum(w_n)]),
                    length=500)
    # Compute density values over x_range
    density_t_values = pdf(density_t_n, x_range)
    density_w_values = pdf(density_w_n, x_range)

    # Plot densities in the current subplot
    plot!(fig, x_range, density_t_values, color=:red, linewidth=2,
          label=L"\bar{X}_n", layout=(2, 3), subplot=idx)
    plot!(fig, x_range, density_w_values, color=:blue, linewidth=2,
          label=L"1/Med({X_n})", subplot=idx)
    hline!(fig, [0], color=:black, linestyle=:dash, linewidth=0.5,
           label="", subplot=idx)
    title!(fig[idx], "n = $n")
end

display(fig) # display the plot
```

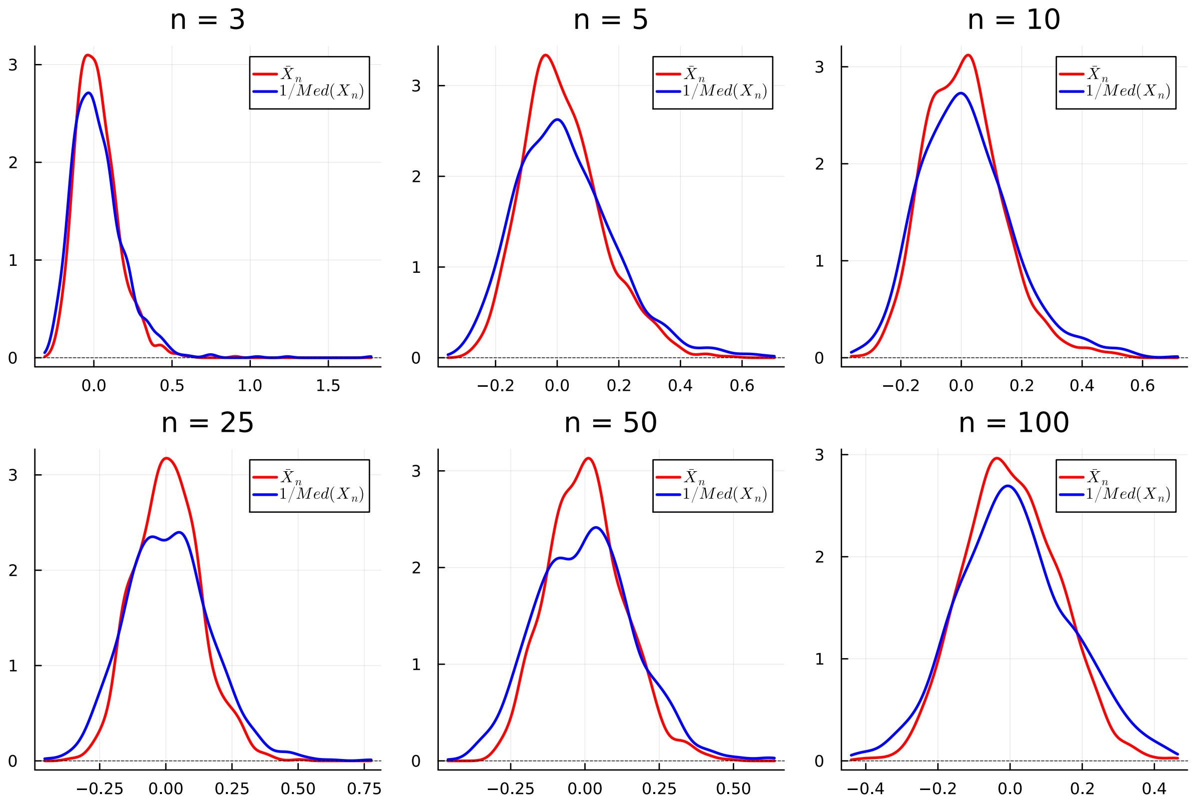

Figure 6: The simulation clearly demonstrates the comparison of the limiting variances of two estimators of $1/\mu$

Variance of the limit distribution of T_n

For an estimator T_n, suppose that

$$k_n\left(T_n-\tau(\theta)\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$$

in distribution. The parameter σ^2 is called the asymptotic variance or variance of the limiting distribution of T_n.

In the above problem $T_n = \overline{X}_n^{-1}$ and

$$\sqrt{n}\left(T_n-\frac{1}{\mu}\right) \xrightarrow{d} \mathcal{N}\left(0, \frac{\sigma^2}{\mu^4}\right).$$

It is interesting to note that although the theoretical variance of $(T_n = \overline{X}_n^{-1})$, $(\mathrm{Var}(T_n) = \infty)$, it has finite asymptotic variance $(\frac{\sigma^2}{\mu^4})$ for $(\mu \neq 0)$, which is in fact more useful. The computation follows by a simple application of the Delta method, which gives $\mathrm{Var}(T_n) \approx \frac{\sigma^2}{n\mu^4} < \infty$.

Asymptotically Efficient

A sequence of estimators of W_n is asymptotically efficient for a parameter $\tau(\theta)$ if

$$\sqrt{n}\left(W_n-\tau(\theta)\right) \xrightarrow{d} \mathcal{N}(0, v(\theta)),$$

where

$$v(\theta) = \frac{\left(\tau'(\theta)\right)^2}{\mathbb{E}_\theta\left(\left(\frac{\partial}{\partial\theta}\log f(X|\theta)\right)^2\right)},$$

that is, the asymptotic variance of W_n) achieves the Cramer-Rao lower bound.

A natural question arises how to obtain an asymptotically efficient estimator, and we are lucky that the MLE is itself an algorithmic way of obtaining asymptotically efficient estimators. In the following section, we discuss this in the light of an example.

3. MLE is Asymptotic Efficient

Suppose that $X_1, \cdots, X_n$ be a random sample of size n from the Poisson distribution with parameter λ. The Fisher Information $I(\lambda) = \lambda^{-1}$. The MLE of the parameter λ is given by $\hat{\lambda} = \overline{X}_n$. It can be easily shown by the CLT that

$$\sqrt{n}\left(\overline{X}_n - \lambda\right) \xrightarrow{d} \mathcal{N}(0, \lambda),$$

in distribution. Therefore, the asymptotic variance of $\overline{X}_n$ is λ. In fact, it is the exact variance as well (why?). Let us perform some simulation experiments to see whether the claim is indeed true or not.

In [23]:
```
using Plots,Distributions,Statistics,StatsBase, LaTeXStrings,StatsPlots
```

In [24]:
```
lambda = 3
rep = 1000
n = 10
```

Out[24]: 10

In [25]:
```
w_n = zeros(rep)

for i in 1:rep
    x = rand(Poisson(lambda), n)
    w_n[i] = sqrt(n)*(mean(x)-lambda)
end
```

In [26]:
```
histogram(w_n, normalize = true, label = "", xlabel = L"w_n",
          title = "n = $n" )
x = range(minimum(w_n), stop=maximum(w_n), length=500) # range
normal_curve = pdf.(Normal(0, sqrt(lambda)), x)
plot!(x, normal_curve, color="red", lw=2, label="")
```

Out[26]:

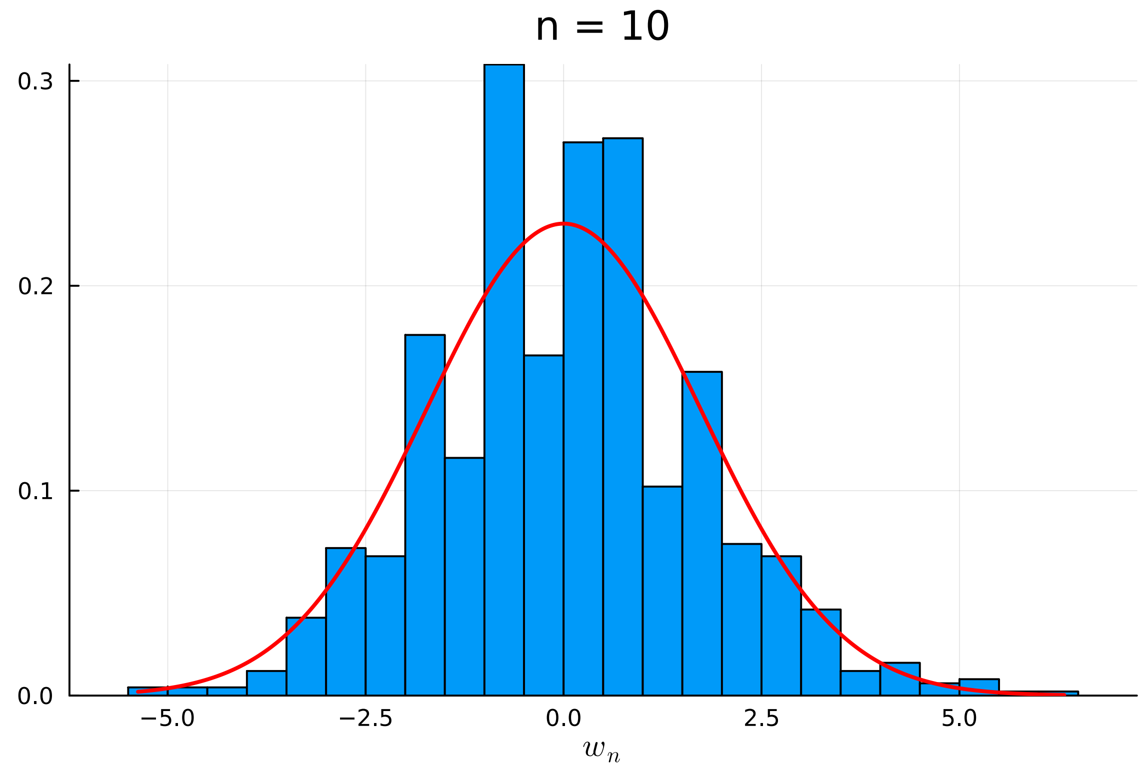

Figure 7: The experiment can be carried out for different choices of n. The overlaying of the normal distribution with the asymptotic variance agrees with the theoretical claim.

The above idea can be extended for estimating any continuous function of λ as well, say $h(\lambda)$. We start with a concrete example. Suppose, we are interested in estimating

$$h(\lambda) = P(X = 2) = \frac{e^{-\lambda}\lambda^2}{2}.$$

Therefore, the estimator is given by

$$h(\hat{\lambda}) = e^{-\bar{X}_n}\frac{\bar{X}_n^2}{2},$$

which is a highly nonlinear function of $\bar{X}_n$. The theory suggests that

$$\sqrt{n}\left(h(\hat{\lambda}) - h(\lambda)\right) \to \mathcal{N}(0, v(\lambda)),$$

in distribution where

$$v(\lambda) = \frac{(v'(\lambda))^2}{I(\lambda)} = \frac{\lambda^3 e^{-2\lambda}(2-\lambda)^2}{4}.$$

In [27]:
```
using Plots,Distributions,Statistics,StatsBase, LaTeXStrings,StatsPlots
```

In [28]:
```
h(lambda) = lambda^2*exp(-lambda)/2 # define the function
```

Out[28]: h (generic function with 1 method)

In [29]:
```
lambda = 3
n = 3
rep = 1000
```

Out[29]: 1000

In [30]:
```julia
v_n = zeros(rep)
for i in 1:rep
    x = rand(Poisson(lambda), n)
    v_n[i] = sqrt(n) * (h(mean(x)) - h(lambda))
end
```

In [31]:
```julia
histogram(v_n, normalize=true, label="", title="n = $n", xlabel=L"v_n")
x = range(minimum(v_n), stop=maximum(v_n), length=500)
normal_curve = pdf.(Normal(0,
            sqrt(lambda^3 * exp(-2 * lambda) * (2 - lambda)^2 / 4)), x)
plot!(x, normal_curve, color="red", lw=2, label="")
```

Out[31]:

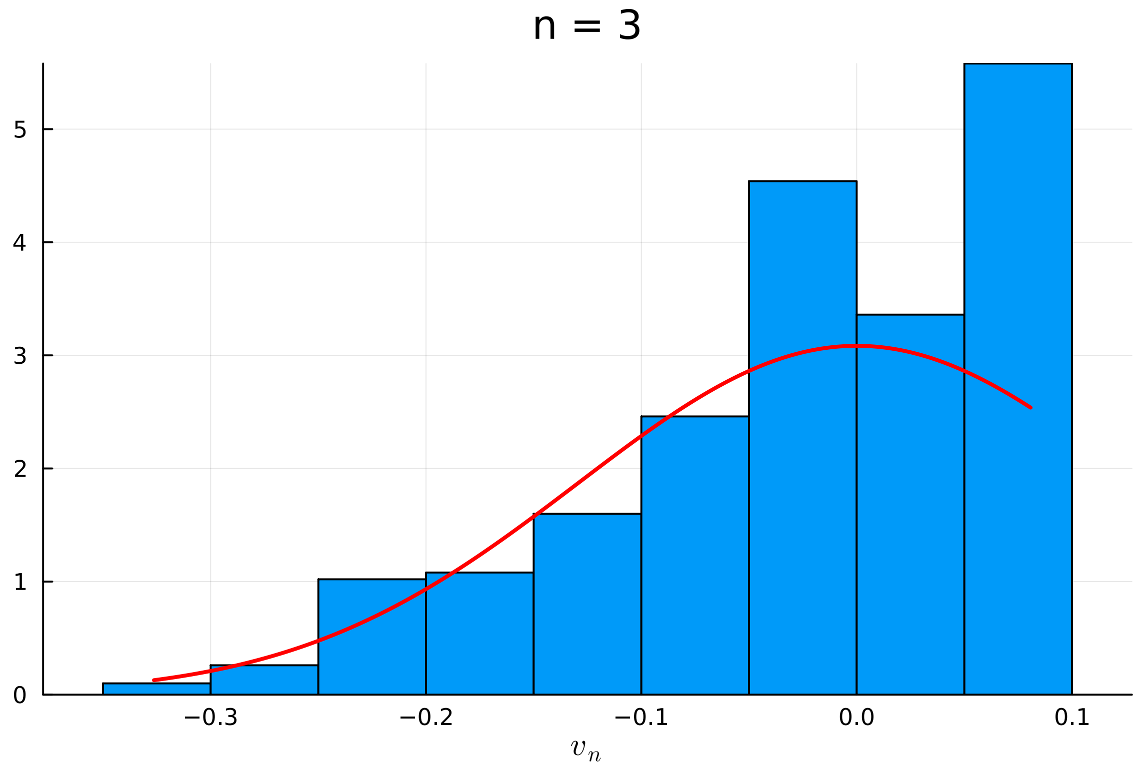

Figure 8: The sample size is small, therefore, the histogram is not a good approximation of the normal distribution. The reader is encouraged to do the simulation with different values of n

In [32]:
```julia
using Plots,Distributions,Statistics,StatsBase, LaTeXStrings,StatsPlots
```

In [33]:
```julia
n_vals = [5,10,25,50, 100, 500]
lambda = 3
rep = 1000
```

Out[33]: 1000

In [34]:
```julia
fig = plot(layout=(2, 3), size=(900, 600)) # set the figure layout

for (idx, n) in enumerate(n_vals)
    v_n = zeros(rep)
    for i in 1:rep
        x = rand(Poisson(lambda),n)
        v_n[i] = sqrt(n)*(h(mean(x)) - h(lambda))
    end
    hist = histogram!(v_n,normalize = true, color = :lightblue,
           linecolor = :black,  label = "", subplot = idx, xlabel = L"v_n",
           title = "n = $n")
```

```
    x = range(minimum(v_n), stop=maximum(v_n), length=500)
    normal_curve = pdf.(Normal(0,
            sqrt(lambda^3 * exp(-2 * lambda) * (2 - lambda)^2 / 4)), x)
    plot!(x, normal_curve, color="red", lw=2, label="", subplot = idx)
    println(var(v_n)) # print the variance
end

display(fig) # display the plot
```

```
0.011390068299257192
0.013091954883046663
0.014658568454995184
0.015903899558623105
0.015449857962328253
0.01590558801305986
```

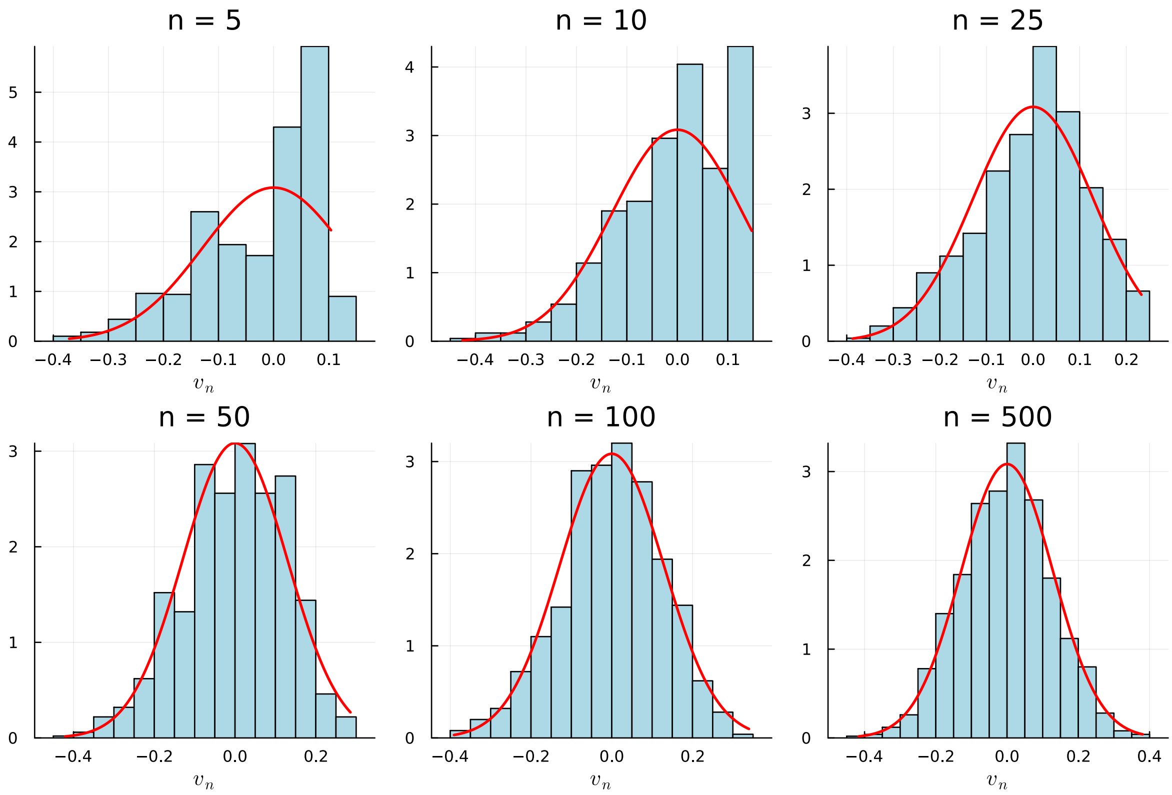

Figure 9: As the sample size increases, the approximation to the normal distribution is cleraly visible with the variance equal to the asymptotic variance.

In the following, we numerically (through simulation) verify how accurate the approximation of the variance by plugging in the $\hat{\lambda}$ in place of λ.

$$\mathrm{Var}\left(h\left(\hat{\lambda}\right)|\lambda\right) \approx \frac{(h'(\lambda))^2}{I_n(\lambda)}$$

$$= \frac{(h'(\lambda))^2}{\mathbb{E}_\lambda\left(-\frac{\partial^2}{\partial\lambda^2}\log\mathcal{L}(\theta|\mathbf{X})\right)}$$

$$\approx \frac{\left[h'(\hat{\lambda})\right]^2}{-\frac{\partial^2}{\partial\lambda^2}\log\mathcal{L}(\theta|\mathbf{X})|_{\lambda=\hat{\lambda}}}.$$

In the above computation, two approximations have been carried out. In the first approximation, the computation of the asymptotic variance has been carried out by the first order Taylor's approximation, whereas in the second approximation, the expectation has been approximated by plugging in the MLE at the Fisher Information.

In [35]:
```
using Plots,Distributions,Statistics,StatsBase, LaTeXStrings,StatsPlots
```

In [36]:
```
h(lambda) = lambda^2*exp(-lambda)/2 # define the function
lambda = 3
n_vals = 1:1000
asym_var = lambda^3*exp(-2*lambda)*(2-lambda)^2/4
rep = 1000
```

Out[36]: 1000

In [37]:
```
var_v_n = zeros(length(n_vals))

for n in n_vals
    v_n = zeros(rep)
    for i in 1:rep
        x = rand(Poisson(lambda), n)
        v_n[i] = sqrt(n)*(h(mean(x)) - h(lambda))
    end
    var_v_n[n] = var(v_n)
end
```

In [38]:
```
scatter(n_vals,var_v_n, color = "grey", xlabel = "sample size (n)",
       label = "", title = L"Var(h(\hat{\lambda_n}))")
hline!([asym_var], color = "red", linestyle = :dash, lw = 3,
    label = L"\frac{\lambda^3 e^{-2\lambda} (2 - \lambda)^2}{4}")
```

Out[38]:

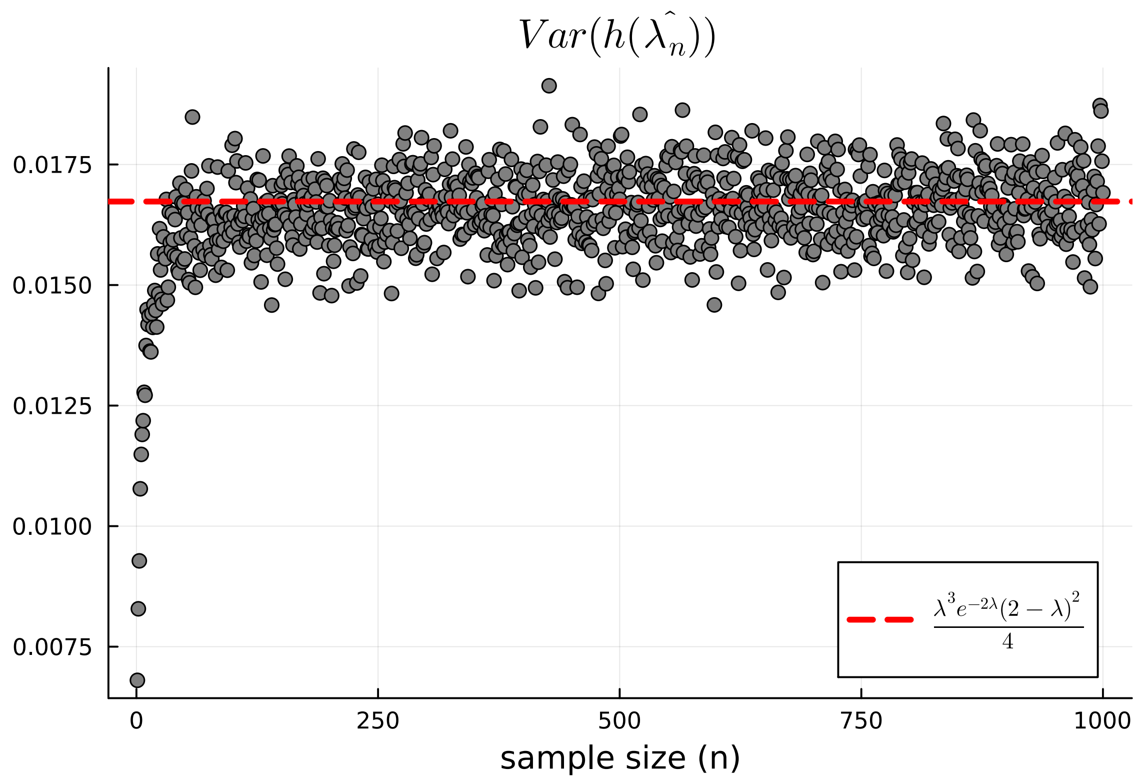

Figure 10: As the sample size increases, the approximated variance is close to the asymptotic variance. The asymptotic variance is shown using the dotted blue color line.

Asymptotic efficiency of MLE

Let $X_1, \ldots, X_n, \ldots$ be i.i.d. $f(x|\theta)$, let $\hat{\theta}$ denote the MLE of θ, and let $\tau(\theta)$ be a continuous function of θ. Under the regularity conditions on $f(x|\theta)$, and, hence on $\mathcal{L}(\theta|x)$, the likelihood function,

$$\sqrt{n}\left(\tau(\hat{\theta}) - \tau(\theta)\right) \to \mathcal{N}(0, v(\theta)),$$

where $v(\theta)$ is the Cramer-Rao Lower Bound. That is, $\tau(\hat{\theta})$ is a consistent and asymptotically efficient estimator of $\tau(\theta)$.

Statistical Model for Contaminated data

Suppose that we have a random sample of size n from the normal distribution with mean μ and variance σ^2. However, there is a contamination with some values from another distribution as well.

Consider the statistical model for the data with contamination as

$$X \sim \begin{cases} \mathcal{N}(\mu, \sigma^2), & \text{with probability } 1-\delta, \\ f(x), & \text{with probability } \delta. \end{cases}$$

In the following, we simulate a random sample of size n from the distribution with 100% contamination.

In [39]:
```
using Plots,Distributions,Statistics,StatsBase, LaTeXStrings,StatsPlots
```

In [40]:
```
mu = 2
sigma2 = 0.5

theta = 5
tau2 = 0.5

delta = 0.1
n = 1000
```

Out[40]:
```
1000
```

In [41]:
```
x = zeros(n)
for i in 1:n
    if rand(Binomial(1, 1 - delta)) == 1
        x[i] = rand(Normal(mu, sqrt(sigma2)))
    else
        x[i] = rand(Normal(theta, sqrt(tau2)))
    end
end
```

In [42]:
```
histogram(x, normalize = true, label = "", title = "histogram of x",
          xlabel = "x")
```

Out[42]:

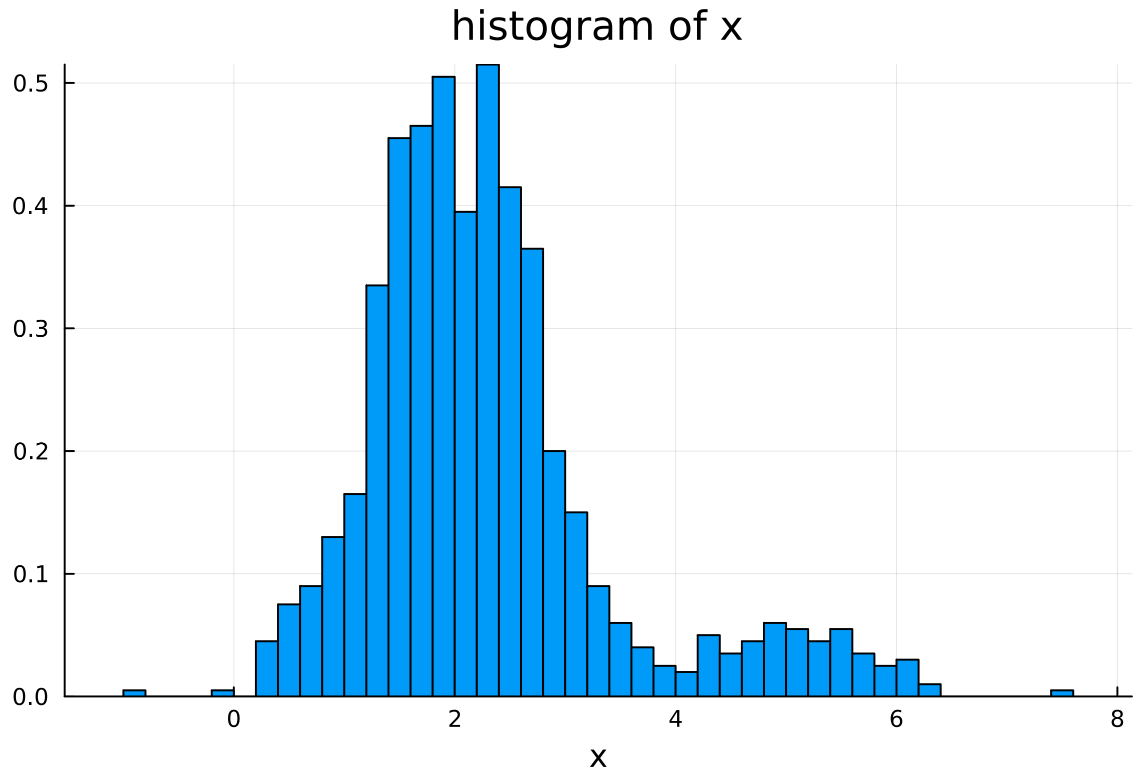

Figure 11: Histogram illustrating the density distribution of a randomly generated sample from a contaminated normal distribution. The distribution consists of two components: a primary normal distribution with mean $\mu = 2$ and variance $\sigma^2 = 0.5$, and a contamination component with mean $\theta = 5$ and variance $\tau^2 = 0.5$. The contamination proportion is set at $\delta = 0.1$, meaning 10% of the sample is drawn from the contamination distribution. The histogram is shaded in grey, representing the probability density of the generated sample.

the mean and variance of $\bar{X}_n$ given by

$$\text{Var}\left(\overline{X}_n\right) = (1-\delta)\frac{\sigma^2}{n} + \delta\frac{\tau^2}{n} + \frac{\delta(1-\delta)(\theta-\mu)^2}{n}.$$

If $\theta \approx \mu$ and $\sigma \approx \tau$, then $\text{Var}\left(\overline{X}_n\right) \approx \frac{\sigma^2}{n}$, which means it achieves nearly optimal efficiency. However, the choice of $f(x)$ plays a critical role. For example, if $f(x)$ is the Cauchy distribution, then the variance becomes infinite.

You are encouraged to do some simulation considering the Cauchy distribution and plot the sampling distribution of $\overline{X}_n$ for different choices of δ.

Breakdown value

Let $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$ be an ordered sample of size n, and let T_n be a statistic based on this sample. T_n has a breakdown value b, $0 \leq b \leq 1$, if for every $\varepsilon > 0$,

$$\lim_{X_{((1-\delta)n)} \to \infty} T_n < \infty \quad \text{and} \quad \lim_{X_{((1-\delta)n)} \to \infty} T_n = \infty.$$

- The sample mean $\overline{X}_n$ has a breakdown value of 0.
- The sample median M_n has a breakdown value of $\frac{1}{2}$.

Asymptotic normality of the M_n

Suppose that $X_1, \ldots, X_n$ be a random sample of size n from the population density function $f(x)$ with CDF $F(x)$. Assume that the CDF is differentiable and the median is μ, that is $F(\mu) = \frac{1}{2}$.

- **Verify that, if n is odd, then**

$$P\left(\sqrt{n}(M_n - \mu) \leq a\right) = P\left(\frac{\sum Y_i - np_n}{\sqrt{np_n(1-p_n)}} \geq \frac{(n+1)/2 - np_n}{\sqrt{np_n(1-p_n)}}\right)$$

- **Show that as $n \to \infty$, $p_n \to p = F(\mu) = \frac{1}{2}$ and**

$$\frac{(n+1)/2 - np_n}{\sqrt{np_n(1-p_n)}} \to -2aF'(\mu) = -2af(\mu).$$

- **It is clear from the statement**

$$P\left(\sqrt{n}(M_n - \mu) \leq a\right) \to P\left(Z \geq -2af(\mu)\right)$$

that $\sqrt{n}(M_n - \mu)$ is asymptotically normal with mean 0 and variance $\frac{1}{(2f(\mu))^2}$

First let us understand the above result in terms of computer simulation and visualization. In the following we first perform the experiment with the sampling from the normally distributed population.

In [43]:
```
using Plots,Distributions,Statistics,StatsBase, LaTeXStrings,StatsPlots
```

In [44]:
```
mu = 2
sigma2 = 1
f(x) = pdf(Normal(mu, sqrt(sigma2)), x)
n_vals = [3, 5, 10, 25, 50, 100]
rep = 1000
```

Out[44]: 1000

In [45]:
```
# Set the figure layout
fig = plot(layout = (2, 3), size = (900, 600))

for (idx, n) in enumerate(n_vals)
    M_n = zeros(rep)
    W_n = zeros(rep)
    for i in 1:rep
        x = rand(Normal(mu, sqrt(sigma2)), n)
        M_n[i] = median(x)
        W_n[i] = sqrt(n) * (M_n[i] - mu)
    end
    histogram!(W_n, normalize = true, color = :lightblue, linecolor = :black,
               label = "", subplot = idx, xlabel = L"W_n", title = "n = $n")
    x = range(minimum(W_n), stop = maximum(W_n), length = 500)
    normal_curve = pdf.(Normal(0, sqrt(1 / (4 * f(mu)^2))), x)
    plot!(x, normal_curve, color = "red", lw = 2, label = "",
        subplot = idx)
end
display(fig)
```

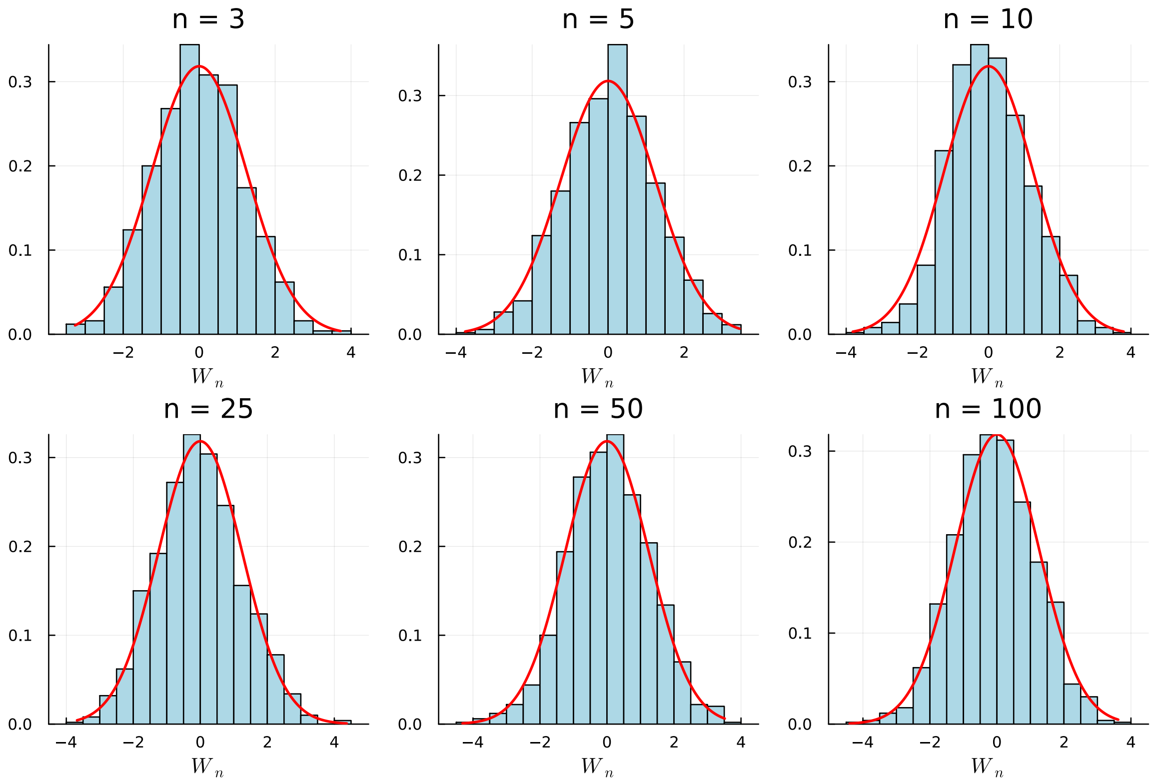

Figure 12: The sampling distribution of M_n is approximately normally distributed with asymptotic variance $\frac{1}{(2f(\mu))^2}$. For simulation, $\mu = 2$ and $\sigma^2 = 1$ have been considered.

In the following, we perform the experiment with the exponential distribution with rate parameter λ. The median of the exponential distribution is given by $\mu = \frac{\ln 2}{\lambda}$. We simulate the distribution of $\sqrt{n}\left(M_n - \frac{\ln 2}{\lambda}\right)$ for different values of n, and as $n \to \infty$, the normal approximation with the desired asymptotic variance is evident from the figures.

In [46]:
```
using Plots,Distributions,Statistics,StatsBase, LaTeXStrings,StatsPlots
```

In [47]:
```
lambda_rate = 2
mu_value = log(2) / lambda_rate
f(x) = pdf(Exponential(1/lambda_rate), x)
n_vals = [3, 5, 10, 25, 50, 100]
rep = 1000
```

Out[47]: 1000

In [48]:
```
fig = plot(layout = (2, 3), size = (900, 600))

for (idx, n) in enumerate(n_vals)
    M_n = zeros(rep)
    W_n = zeros(rep)
    for i in 1:rep
        x = rand(Exponential(1/lambda_rate), n)
        M_n[i] = median(x)
        W_n[i] = sqrt(n) * (M_n[i] - mu_value)
    end

    histogram!(W_n, normalize = true, color = :lightblue, linecolor = :black,
               label = "", subplot = idx, xlabel = L"W_n", title = "n = $n")
    x_range = range(minimum(W_n), stop = maximum(W_n), length = 500)
    normal_curve = pdf.(Normal(0, sqrt(1 / (4 * f(mu_value)^2))), x_range)
    plot!(x_range, normal_curve, color = "red", lw = 2, label = "",
```

```
        subplot = idx)
end

display(fig)
```

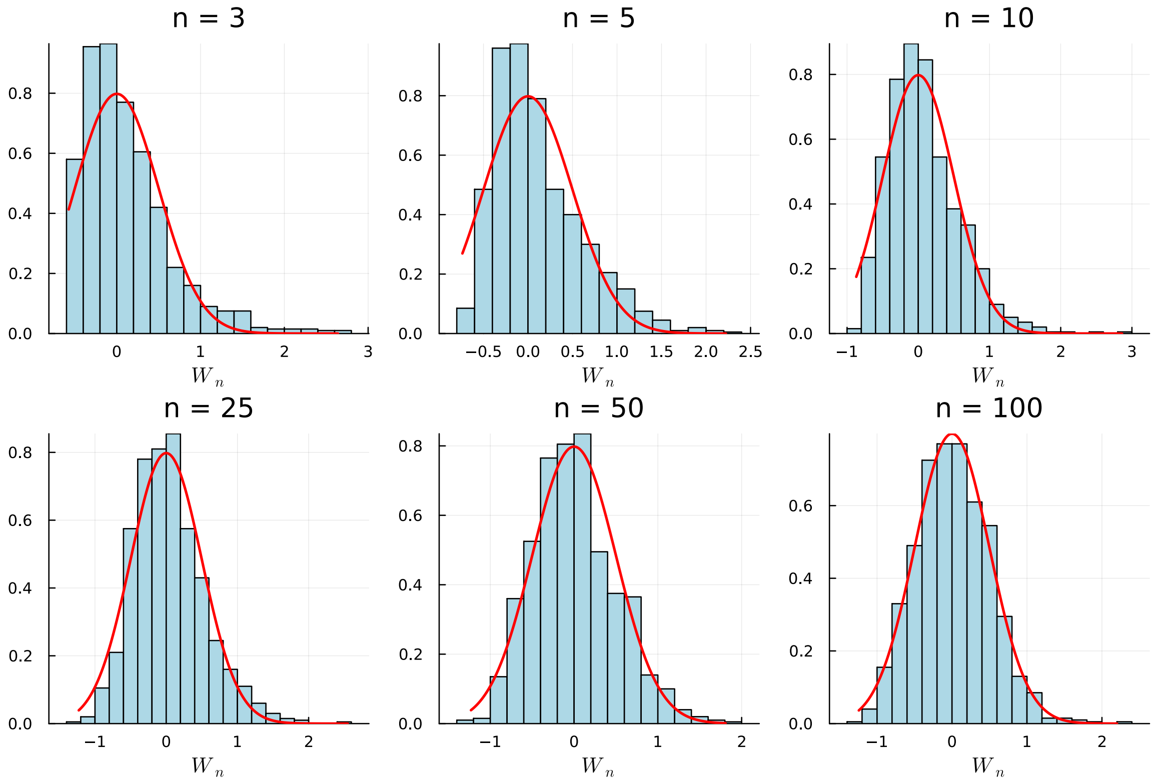

Figure 13:The simulation has been carried from the exponential distribution with parameter $\lambda = 2$, therefor the true median is $\mu = 0.3465736$

References

- Casella, G., & Berger, R. L. (2002). Statistical inference. 2nd ed. Australia ; Pacific Grove, CA, Thomson Learning.