Approximation of Functions of Random Variables: Delta Method

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

1. Introduction

We start our discussion with the celebrated Central Limit Theorem, called as CLT. The idea is that if we draw random sample of size n from a population density with finite variance, then the sampling distribution of the sample mean approximately behaves like a normal distribution for large n. Technically,

$$\frac{\sqrt{n}\left(\overline{X}_n - \mu\right)}{\sigma} \xrightarrow{d} \mathcal{N}(0,1), \quad \text{in distribution.}$$

In other words, $\overline{X}_n \approx \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right)$, for large n values. We have understood this idea using simulation in the Module `Convergence Concepts`.

2. Approximate Sampling Distribution of Sample Mean

We start our discussion with a coin tossing experiment. Suppose that we have iid observations $X_1, X_2, \ldots, X_n$ from $\text{Bernoulli}(p)$ distribution with $0 < p < 1$. Call the sample mean $\hat{p}_n = \overline{X}_n$, and we know that $\hat{p}_n \to p$ in probability (WLLN) and

$$\frac{\sqrt{n}(\hat{p}_n - p)}{\sqrt{p(1-p)}} \xrightarrow{d} \mathcal{N}(0,1) \quad \text{in distribution (CLT).}$$

The estimated standard error of $\hat{p}_n$ is given by

$$\widehat{\text{SE}}(\hat{p}_n) = \sqrt{\frac{\hat{p}_n(1-\hat{p}_n)}{n}}.$$

In fact, it is possible to show that

$$\frac{\hat{p}_n - p}{\text{SE}(\hat{p}_n)} \xrightarrow{d} \mathcal{N}(0,1) \quad \text{in distribution.}$$

Therefore, for large n, an approximate $(1-\alpha)\%$ CI can be constructed as

$$\left(\hat{p}_n - z_{\alpha/2}\widehat{\text{SE}}(\hat{p}_n),\ \hat{p}_n + z_{\alpha/2}\widehat{\text{SE}}(\hat{p}_n)\right)$$

where $Z \sim \mathcal{N}(0,1)$ and $\mathbb{P}(Z > z_{\alpha/2}) = \frac{\alpha}{2}$.

In [1]:
```julia
using Plots, Statistics, Distributions
using StatsBase, LaTeXStrings
```

In [2]:
```julia
plt = plot(layout = (2, 2), size = (800, 500))
n_vals = [5, 10, 30, 50]
p = 0.4
rep = 10000

for (idx, n) in enumerate(n_vals)
    mean_vals = zeros(rep)
    for j in 1:rep
        mean_vals[j] = mean(rand(Binomial(1, p), n))
    end
    histogram!(mean_vals, normalize = true, bins = :auto, xlabel = L"\overline{X
        ylabel = "density",label = "", title = "n = $n", xlims=extrema(mean_vals
        , subplot = idx)
    plot!(x -> pdf.(Normal(p, sqrt(p * (1 - p) / n)), x),
        color = "red",label = "" ,subplot = idx)
end

display(plt)
```

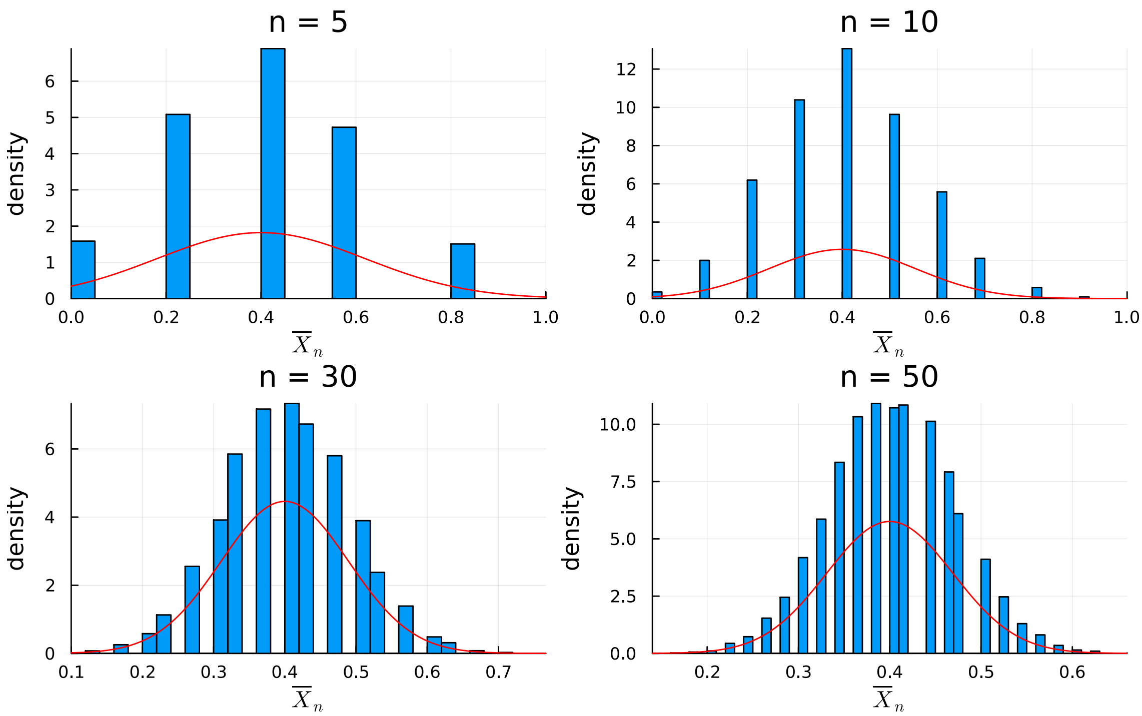

Figure 1: Experiment with binomial(p) distribution. As $n \to \infty$, sampling distribution of $\hat{p}_n$ is approximately normal.

In the following, we also check whether the asymptotic confidence coefficient of the interval is $(1-\alpha)$. It is important to realize that the interval CI_n is a random interval. If we repeatedly draw a sample of size n from the population, we shall get different different intervals. We try to estimate approximately how many of these intervals contain the true value of p. The simulation scheme for doing this analysis is given below:

- Fix $p \in (0,1)$
- Fix n, simulate $X_1, X_2, \ldots, X_n \sim \text{binomial}(1, p)$
- Construct $CI_n = \left(\hat{p}_n - z_{\alpha/2}\widehat{\text{SE}}(\hat{p}_n),\ \hat{p}_n + z_{\alpha/2}\widehat{\text{SE}}(\hat{p}_n)\right)$

- If CI_n contains p, set the indicator as 1, otherwise 0.
- Repeat the previous three steps, `rep = 100000` times and compute the proportion of 1's to obtain an approximate coverage probability of CI_n.
- Repeat the process for different choices of n and plot the approximate coverage probabilities as a function of n.

In [3]:

```
p = 0.4
α = 0.05  # alpha
n_vals = collect(5:2:500)  # varied samples
emp_prob = zeros(length(n_vals))
rep = 1000   # no. of replications

for i in 1:length(n_vals)
    n = n_vals[i]
    counter = ones(Int, rep)
    for j in 1:rep
        p_n = mean(rand(Binomial(1, p), n))
        p_n_se = sqrt(p_n * (1 - p_n) / n)
        if (abs(p_n - p) / p_n_se) > quantile(Normal(0, 1), 1 - α/2)
            counter[j] = 0
        end
    end
    emp_prob[i] = sum(counter) ./ rep
end
```

In [4]:

```
plot(n_vals, emp_prob, color = "red", lw = 2, xlabel = "sample size(n)",
ylabel = "Coverage Probability", label = "")
hline!([1-α], color = "blue", linestyle = :dash, lw = 2, label = "")
```

Out[4]:

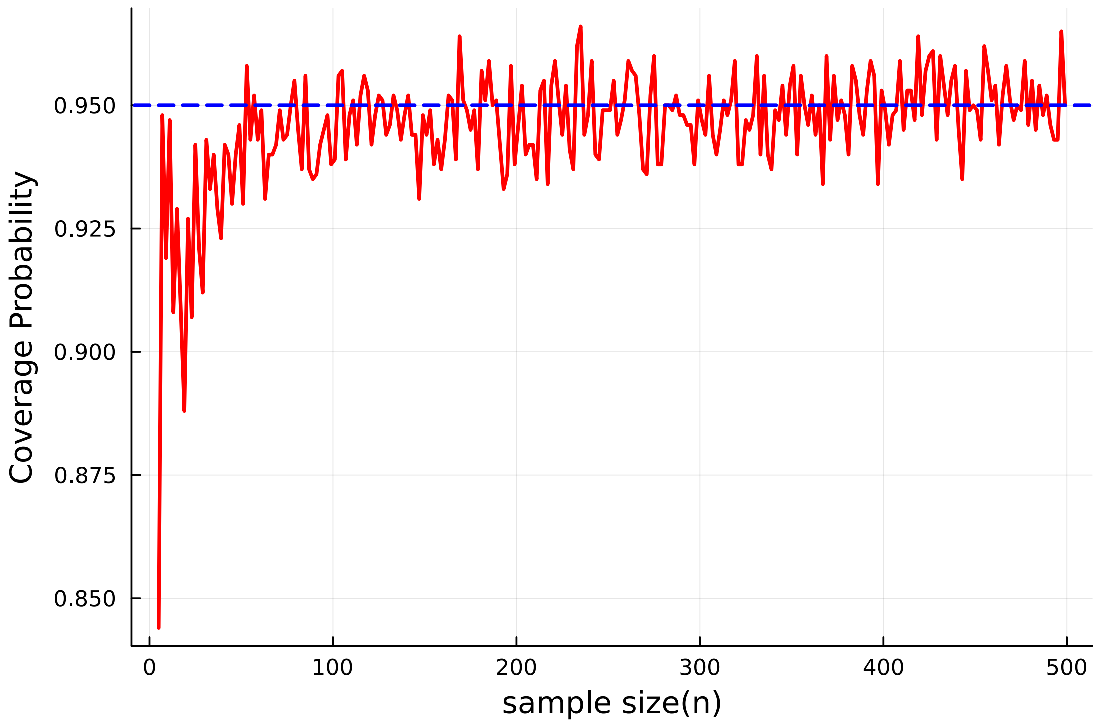

Figure 2: As $n \to \infty$, $\mathbb{P}(\mathrm{CI}_n \ni p) \to 1 - \alpha$.

It might happen that for the $\mathrm{bernoulli}(p)$ distribution we are lucky to have aapproximate normal distribution of the sample mean for large n values.Let us try with some alternative distribution,say exponential density with parameter 1.

In [5]:
```
plt = plot(layout = (2, 2), size = (800, 500))
n_vals = [5, 10, 30, 50]
λ = 1
rep = 10000

for (idx, n) in enumerate(n_vals)
    mean_vals = zeros(rep)
    for j in 1:rep
        mean_vals[j] = mean(rand(Exponential(1/λ), n))
    end
    histogram!(mean_vals, normalize = true, color = "lightgrey",
        xlabel = L"\overline{X}_n", ylabel = "density", xlims = extrema(mean_val
        label = "",title = "n = $n",subplot = idx)
    plot!(x->pdf.(Normal(λ, λ/sqrt(n)), x), color = "red", lw = 2,
        label = "", subplot = idx)
end
display(plt)
```

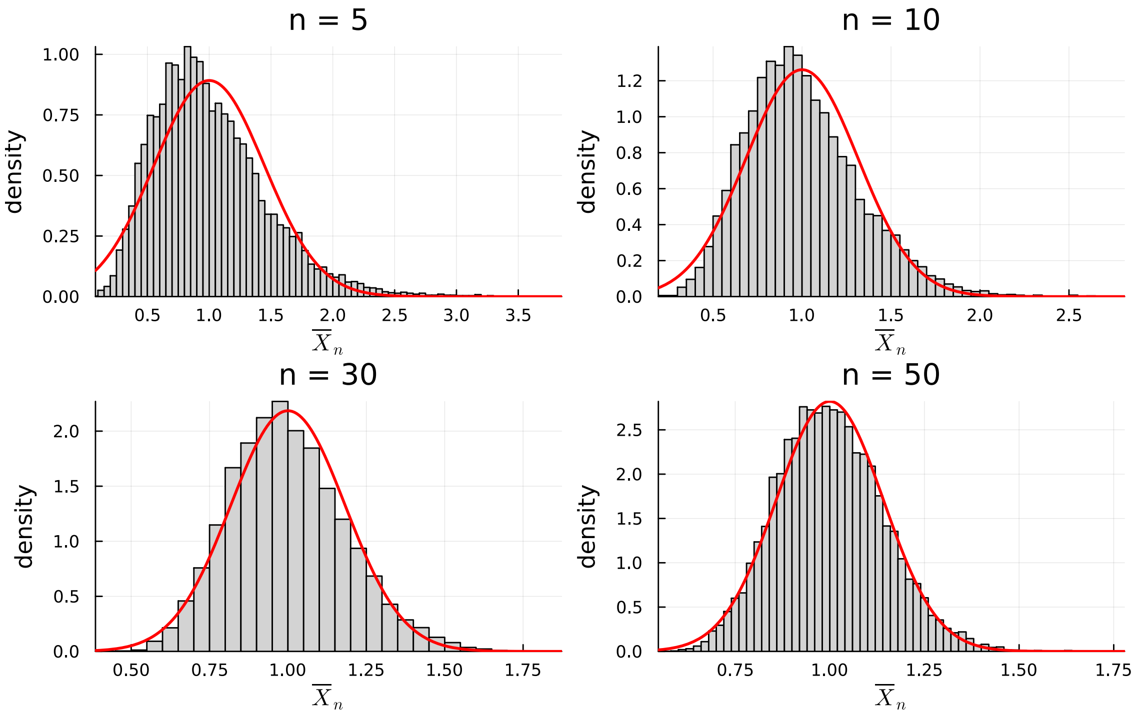

Figure 3: Experiment with $\text{exponential}(1)$ distribution. As $n \to \infty$, the sampling distribution of $\overline{X}_n$ is approximately normal.

We are tempted to make a conclusion that in all circumstances, the approximate sampling distribution of the sample mean seems to be the normal distribution for large n. However, for simulation using the Cauchy distribution, we observe that the distribution of $\overline{X}_n$ does not converge to the normal distribution for large n values.

In [6]:
```
using StatsPlots, Distributions, Random

plt = plot(layout = (2, 2), size = (700, 500))
n_vals = [4, 10, 100, 150]
rep = 10000

for (idx, n) in enumerate(n_vals)
    mean_vals = zeros(rep)
```

```
    for j in 1:rep
        mean_vals[j] = mean(rand(Cauchy(0,1), n))
    end
    clipped_means = mean_vals[abs.(mean_vals) .< 50]
    histogram!(clipped_means, normalize = true, xlabel = L"\overline{X}_n",
        ylabel = "density", label = "", subplot = idx, bins = 100)
end

display(plt)
```

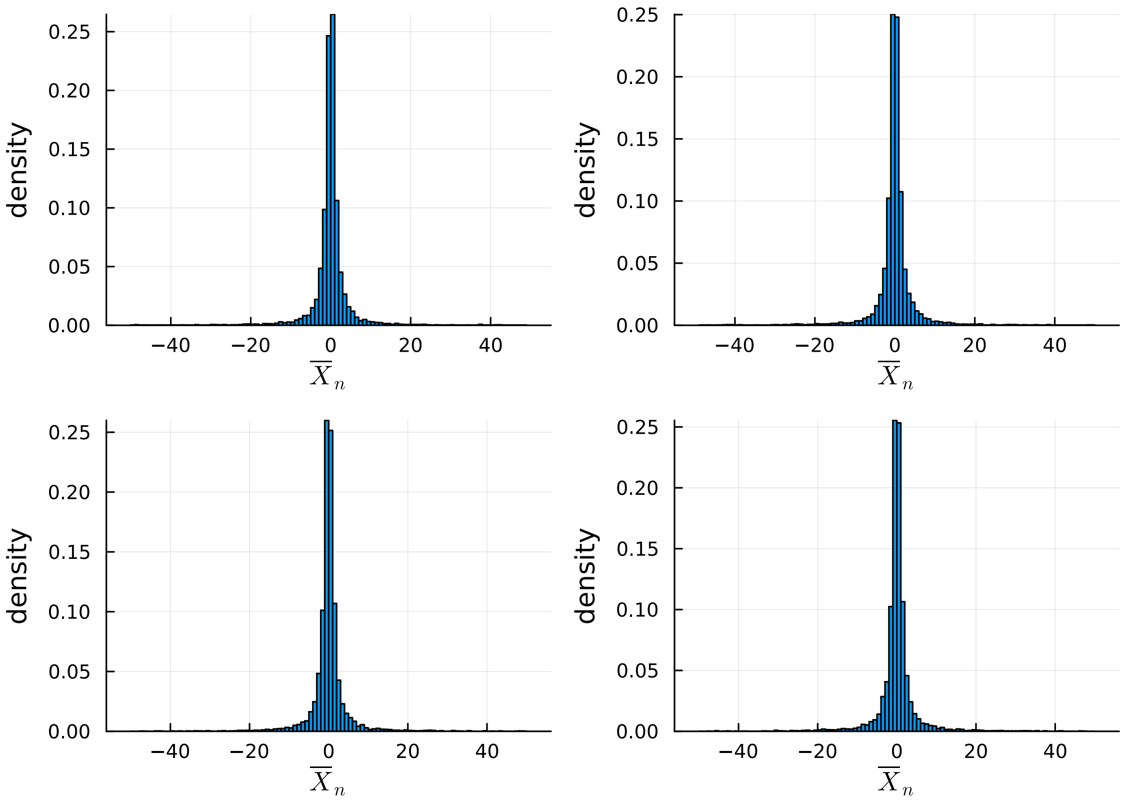

Figure 4: Experiment with the Cauchy distribution. Here, as n increases, the convergence to normality does not occur

3. Generalization

If we have a random sample of size n from a population with finite mean μ and finite variance σ^2, then for large n

$$\overline{X}_n \approx \mathcal{N}\left(\mu, \frac{\sigma^2}{n}\right).$$

This is convergence in distribution. In the following, we formally state the Central Limit Theorem. The Berry-Esseen inequality also provides a bound on the accuracy of the approximation.

Central Limit Theorem

Let $X_1, X_2, \ldots, X_n$ be iid with mean μ and variance σ^2. Let $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Then

$$\lim_{n\to\infty} \mathbb{P}\left(\frac{\sqrt{n}(\overline{X}_n - \mu)}{\sigma} \le z\right) = \lim_{n\to\infty} \mathbb{P}(Z_n \le z) = \Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx.$$

Berry-Esseen Inequality
Suppose that $\mathbb{E}|X_1|^3 < \infty$. Then

$$\sup_z |\mathbb{P}(Z_n \le z) - \Phi(z)| \le \frac{33}{4} \cdot \frac{\mathbb{E}|X_1 - \mu|^3}{\sqrt{n}\,\sigma^3}.$$

Remark

For large sample size n, we can construct an approximate $(1-\alpha)\%$ CI for the population mean based on the standard normal cut-off.

3.1. Approximate Sampling distribution of functions of Sample mean

Suppose, we are interested to approximate $g(p) = p(1-p)$, which is the population variance. A natural choice of the estimator of $g(p)$ is $g(\hat{p}_n) = \hat{p}_n(1-\hat{p}_n)$. We aim to approximate the sampling distribution of $g(\hat{p}_n) = g(\overline{X}_n)$ for large n. The approximation will be carried out using the Taylor's polynomial approximation theorem which is stated below.

Taylor's theorem

If a function $g(x)$ has a derivative of order r, that is, $g^{(r)}(x)$ exists, then for any constant a, the **Taylor polynomial of order** r about a is given by

$$T_r(x) = \sum_{i=0}^{r} \frac{g^{(i)}(a)}{i!}(x-a)^i.$$

The remainder from this approximation, $g(x) - T_r(x)$, always tends to 0 faster than the higher-order explicit term.
Consider a random variable X with $\mathbb{E}_\mu(X) = \mu$. For some function $g(\cdot)$, the approximate mean and variance of $g(X)$ can be evaluated using the first order Taylor's series expansion. Let g be differentiable at μ and $g'(\mu) \ne 0$. Then

$$g(X) \approx g(\mu) + (X-\mu)g'(\mu) \tag{1}$$
$$\mathbb{E}(g(X)) \approx g(\mu) + \mathbb{E}(X-\mu)g'(\mu) = g(\mu) \tag{2}$$

and for approximating the variance, we consider

$$g(X) - g(\mu) \approx (X-\mu)g'(\mu)$$
$$\mathbb{E}[(g(X)-g(\mu))^2] \approx \mathbb{E}[(X-\mu)g'(\mu)]^2$$
$$\mathrm{Var}(g(X)) \approx (g'(\mu))^2\mathrm{Var}(X).$$

Remark: It might happen that the exact mean and variance of $g(X)$ does not exist.

3.2. Worked Example I

$X \sim \exp(3)$, and $g(X) = \sqrt{X}$.
$\mathbb{E}(X) = 3$ and $\mathrm{Var}(X) = 3^2 = 9$.
$g'(x) = \frac{1}{2\sqrt{x}}$ and $g'(3) = \frac{1}{2\sqrt{3}} \neq 0$.

Therefore,
$\mathbb{E}(\sqrt{X}) \approx \sqrt{3}$ and
$\mathrm{Var}(\sqrt{X}) \approx (g'(3))^2 \times 9 = \frac{3}{4}$.

You are encouraged to find the exact mean and variance of $\sqrt{X}$.

Let
$g(p) = p(1-p), \quad 0 < p < 1$ and $g'(p) = 1 - 2p \neq 0$ for $p \neq \frac{1}{2}$.

Therefore,
$\mathbb{E}[g(\hat{p}_n)] \approx g(p)$
and

$$\mathrm{Var}(g(\hat{p}_n)) \approx (1-2p)^2\,\mathrm{Var}(\hat{p}_n) = (1-2p)^2 \times \frac{p(1-p)}{n}.$$

Before drawing any conclusion regarding the sampling distribution for large n, let us visualize the distribution by simulation. The following chunk of R code will do the task.

In [7]:
```
using Statistics, Plots, Distributions
using StatsBase, LaTeXStrings
```

In [8]:
```
plt = plot(layout = (2, 3), size = (800, 500))
n_vals = [5, 10, 30, 50, 100, 500] # different sample size
p = 0.4 # true value of p

# function of g(p) = p*(1-p)
g(x) = x .*(1 .- x)
rep = 1000

for (idx, n) in enumerate(n_vals)
    g_vals =  zeros(rep)
    for j in 1:rep
        g_vals[j] = g.(mean(rand(Binomial(1, p), n)))
    end
    histogram!(g_vals, normalize = true, color = "lightgrey",
    xlabel = L"\widehat{p}_n", ylabel = "density", label = "",
    title = "n = $n", subplot = idx)
    scatter!([g(p)], [0], color = "red", markersize = 6, label = "",
    subplot = idx)
    x_range = range(minimum(g_vals), stop = maximum(g_vals), length = 500)
    normal_curve = pdf.(Normal(g(p), sqrt(((1-2*p)^2)*p*(1-p)/n) ), x_range)
    plot!(x_range, normal_curve, color = "red", lw = 2, label = "",
    subplot = idx)
end

display(plt)
```

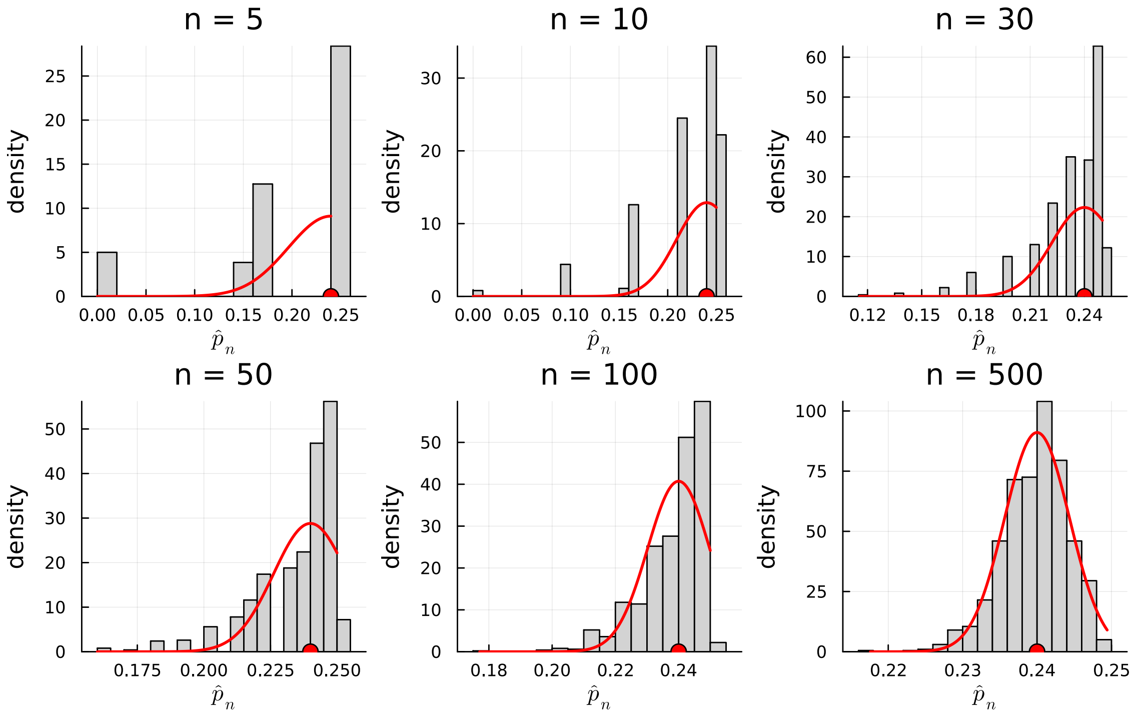

Figure 5: The sampling distribution of $g(\hat{p}_n)$ for different values of n visualized based on 10000 replication. A normal density curve is overlaid with mean $g(p) = g(0.4) = 0.4(1-0.4) = 0.24$ and variance which is equal to $(1-2\times 0.4)^2 \times \frac{0.4(1-0.4)}{n} = \frac{0.0096}{n}$. It is observed that as $n \to \infty$, the sampling distribution is approximately normal.

3.4. Generalization for any function of sample mean

The above observation can be generalized for any functions of sample mean as follows:

Due to CLT

$$\overline{X}_n \sim \mathcal{N}\left(p, \frac{p(1-p)}{n}\right) \quad \text{in distribution.}$$

Due to Delta method

$$g(\overline{X}_n) \sim \mathcal{N}\left(p(1-p), \frac{(1-2p)^2 p(1-p)}{n}\right)$$

Both these results are valid for large n values. When we have an asymptotic normality, we can construct a normality-based $(1-\alpha)$ confidence interval for $g(p) = p(1-p)$, which will be given as

$$\left(\hat{p}_n(1-\hat{p}_n) \ \pm \ z_{\alpha/2}\sqrt{\frac{(1-2\hat{p}_n)^2\hat{p}_n(1-\hat{p}_n)}{n}}\right)$$

3.5. Worked Example II

Let $X_1, X_2, \ldots, X_n$ be i.i.d. with mean μ and finite variance σ^2. By CLT we have $\sqrt{n}(\overline{X}_n - \mu)/\sigma \to \mathcal{N}(0,1)$. Let $W_n = e^{\overline{X}_n}$. So, $W_n = g(\overline{X}_n)$ where $g(t) = e^t$. Since

$g'(t) = e^t$, the Delta method implies

$$W_n \approx \mathcal{N}\left(e^{\mu}, \frac{e^{2\mu}\sigma^2}{n}\right).$$

Students are encouraged to perform the simulation study to ensure that the convergence in distribution to the normality for large n values.

3.6. Second Order Delta Method $(p = \frac{1}{2})$

When $p = \frac{1}{2}$, then $g'(1/2) = 0$, therefore, variance becomes zero. For those points where the first order derivatives vanish, we use higher order approximation from the Taylor series as follows:

$$g(\hat{p}_n) = g(1/2) + \left(\hat{p}_n - \frac{1}{2}\right) g'\left(\frac{1}{2}\right) + \left(\hat{p}_n - \frac{1}{2}\right)^2 \frac{g''\left(\frac{1}{2}\right)}{2!} + \text{Reminder}$$

Therefore,

$$\hat{p}_n(1 - \hat{p}_n) - \frac{1}{4} \approx \left(\hat{p}_n - \frac{1}{2}\right)^2 \frac{(-2)}{2} = -\left(\hat{p}_n - \frac{1}{2}\right)^2$$

Let us find out the approximate sampling distribution for large n. Due to CLT, for $p = \frac{1}{2}$, $\sqrt{n} \cdot \frac{\hat{p}_n - \frac{1}{2}}{\sqrt{\frac{1}{4}(1-\frac{1}{2})}} = \sqrt{n} \cdot \frac{\hat{p}_n - \frac{1}{2}}{\sqrt{\frac{1}{4} \cdot \frac{1}{2}}} \sim \mathcal{N}(0, 1)$ for large n.

Therefore,

$$n\left(\hat{p}_n - \frac{1}{2}\right)^2 \sim \chi_1^2 \quad \text{for large } n.$$

So,

$$n\left[\hat{p}_n(1 - \hat{p}_n) - \frac{1}{4}\right] \sim -\frac{1}{4}\chi_1^2 \quad \text{for large } n.$$

In [9]:

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

In [10]:

```
plt = plot(layout = (2, 3), size = (800, 500))
n_vals = [10, 20, 50, 100, 250, 500] # different sample size
p = 0.5    # true probability
rep = 1000 # number of replications

for (idx, n) in enumerate(n_vals)
    vals = zeros(rep)
    for j in 1:rep
        p_n = mean(rand(Binomial(1,p), n))
        vals[j] = 4*n*(1/4 .- p_n .*(1 .- p_n))
    end
    histogram!(vals, normalize = true, color = "lightgrey", xlabel = "vals",
    ylabel = "density", xlims = extrema(vals),ylims = (0, 0.8),
        label = "",title = "n = $n",subplot = idx)
```

```
    plot!(x->pdf.(Chisq(1), x), color = "red", lw = 2,
    label = L"\chi^2_1",subplot = idx)
end

display(plt)
```

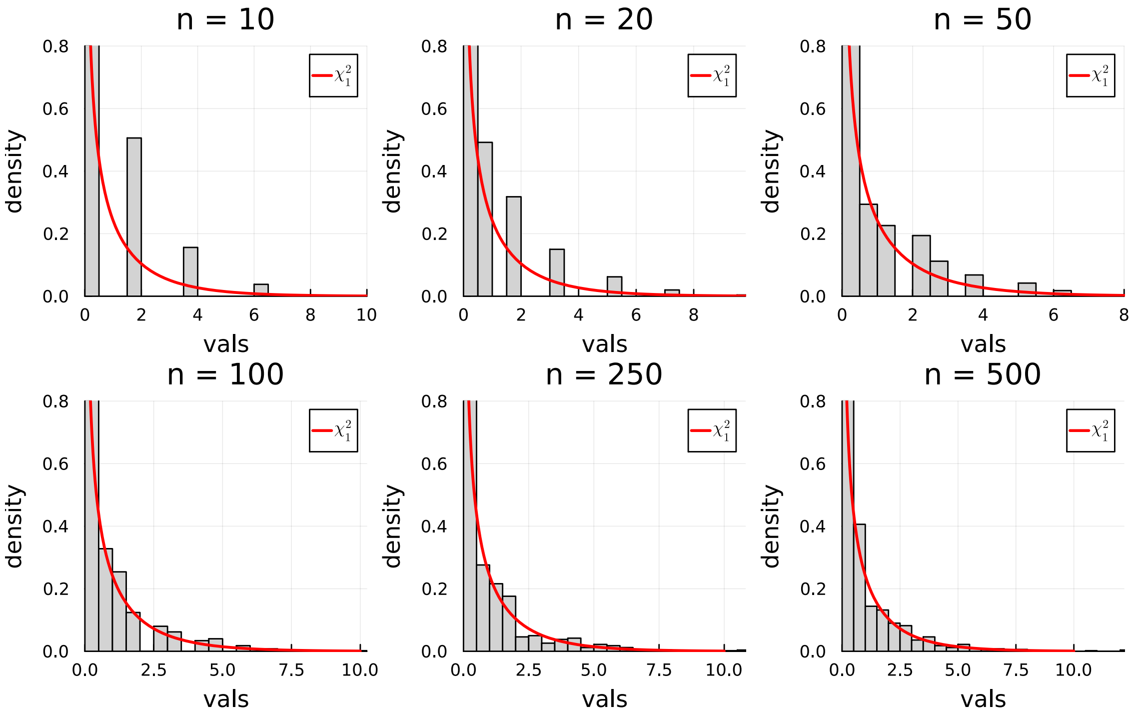

Figure 6: The sampling distribution of $-4n\left[\hat{p}_n(1-\hat{p}_n)-\frac{1}{4}\right]$ is well approximated by χ^2_1 distribution for large n. A χ^2_1 density is overlaid for visualization.

Remark: In general, if $g^{(k)}(\mu) \neq 0$ and $g^{(i)}(\mu) = 0$ for $1 \leq i \leq k-1$, then

$$g(Y_n) - g(\mu) \sim \frac{1}{k!} g^{(k)}(\mu) \left(\frac{\sigma}{\sqrt{n}}\right)^k Z^k \quad \text{in distribution}$$

for large n, where $Z \sim \mathcal{N}(0,1)$.

3.7. Second Order Delta Method

Let Y_n be a sequence of random variables with $\sqrt{n}(Y_n - \mu) \to \mathcal{N}(0, \sigma^2)$ in distribution. For a given function $g(\cdot)$ and a specific value of μ, suppose that $g'(\mu) = 0$ and $g''(\mu) \neq 0$, then

$$n\left[g(Y_n) - g(\mu)\right] \to \sigma^2 \frac{g''(\mu)}{2!} \chi^2_1 \quad \text{in distribution.}$$

4. Multivariate Delta Method

In many instances, we are not only interested to approximate the function of a single parameter, it may be a real-valued function of two or more parameters as well. Suppose X and Y are random variables with nonzero means μ_X and μ_Y, respectively. The parametric function to be estimated is $g(\mu_X, \mu_Y) = \frac{\mu_X}{\mu_Y}$. For any function $g(\cdot, \cdot)$, we can write

$$g(X,Y) = g(\mu_X, \mu_Y) + (X - \mu_X)\frac{\partial g}{\partial X}\bigg|_{(\mu_X, \mu_Y)} + (Y - \mu_Y)\frac{\partial g}{\partial Y}\bigg|_{(\mu_X, \mu_Y)} + R \quad (3)$$

where $\frac{\partial g}{\partial \mu_X} = \frac{1}{\mu_Y}$ and $\frac{\partial g}{\partial \mu_Y} = -\frac{\mu_X}{\mu_Y^2}$. The first order approximation gives

$$\mathbb{E}\left(\frac{X}{Y}\right) \approx \frac{\mu_X}{\mu_Y} \quad (4)$$

and

$$\begin{aligned} \mathrm{Var}\left(\frac{X}{Y}\right) &\approx \frac{1}{\mu_Y^2}\mathrm{Var}(X) + \frac{\mu_X^2}{\mu_Y^4}\mathrm{Var}(Y) - 2\frac{\mu_X}{\mu_Y^3}\mathrm{Cov}(X,Y) \quad (5) \\ &= \left(\frac{\mu_X}{\mu_Y}\right)^2 \left[\frac{\mathrm{Var}(X)}{\mu_X^2} + \frac{\mathrm{Var}(Y)}{\mu_Y^2} - 2\frac{\mathrm{Cov}(X,Y)}{\mu_X \mu_Y}\right] \end{aligned}$$

At this point, it is better to recall the bivariate normal distribution and the problem of multivariate delta method more contextually.

Bivariate normal distribution

Let $-\infty < \mu_X < \infty$, $-\infty < \mu_Y < \infty$, $0 < \sigma_X < \infty$, $0 < \sigma_Y < \infty$, and $-1 < \rho < 1$ be five real numbers. The bivariate normal probability density function with means μ_X and μ_Y, variances σ_X^2 and σ_Y^2, and correlation ρ is given by

$$f(x,y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}}\exp\left[-\frac{1}{2(1-\rho^2)}\left(\left(\frac{x-\mu_X}{\sigma_X}\right)^2 - 2\rho\left(\frac{x-\mu_X}{\sigma_X}\right)\left(\frac{y-\mu_Y}{\sigma_Y}\right.\right.\right.$$

for $-\infty < x < \infty$ and $-\infty < y < \infty$. The following may be considered as an exercise or you can refer to any standard books.

- The marginal distribution of X is $\mathcal{N}(\mu_X, \sigma_X^2)$.
- The marginal distribution of Y is $\mathcal{N}(\mu_Y, \sigma_Y^2)$.
- The correlation between X and Y is $\rho_{XY} = \rho$.
- For any constants a and b, the distribution of $aX + bY$ is $\mathcal{N}(a\mu_X + b\mu_Y, a^2\sigma_X^2 + b^2\sigma_Y^2 + 2ab\rho\sigma_X\sigma_Y)$.
- The conditional distribution of Y given $X = x$ is

$$\mathcal{N}\left(\mu_Y + \rho\left(\frac{\sigma_Y}{\sigma_X}\right)(x - \mu_X),\, \sigma_Y^2(1-\rho^2)\right). \quad (6)$$

Beyond the formula or technical result, the following graphical demonstration will help to understand the distribution better. We shall utilize the `MASS` package to simulate observations from the bivariate normal distribution.

In [11]:
```
using Plots, Distributions, Random
using Statistics, StatsBase
```

```
using StatsPlots, LaTeXStrings
using DataFrames
```

In [12]:
```
plt = plot(layout = (2, 3), size = (800, 500))
Random.seed!(123)

n = 100  # sample size
μ_x = 1  # mean of x
μ_y = 1  # mean of y
σ_x = 1  # sd of x
σ_y = 1   # sd of y
ρ_vals = [-0.9,-0.5, 0, 0.5, 0.7, 0.95]  # different cov(x,y) values

for (idx, ρ) in enumerate(ρ_vals)
    μ = [μ_x, μ_y]
    Σ = [σ_x^2  ρ*σ_x*σ_y; ρ*σ_x*σ_y  σ_y^2]   # matrix form
    data = rand(MvNormal(μ, Σ),n)
    bivarate_data = DataFrame(x = data[1, :], y = data[2, :])
    scatter!(bivarate_data.x, bivarate_data.y, color = "red", xlabel = "x",
        ylabel = "y",title = "ρ = $ρ",label = "" ,subplot = idx)
    scatter!(μ_x, μ_y, color = "red", label = "", subplot = idx )
end
display(plt)
```

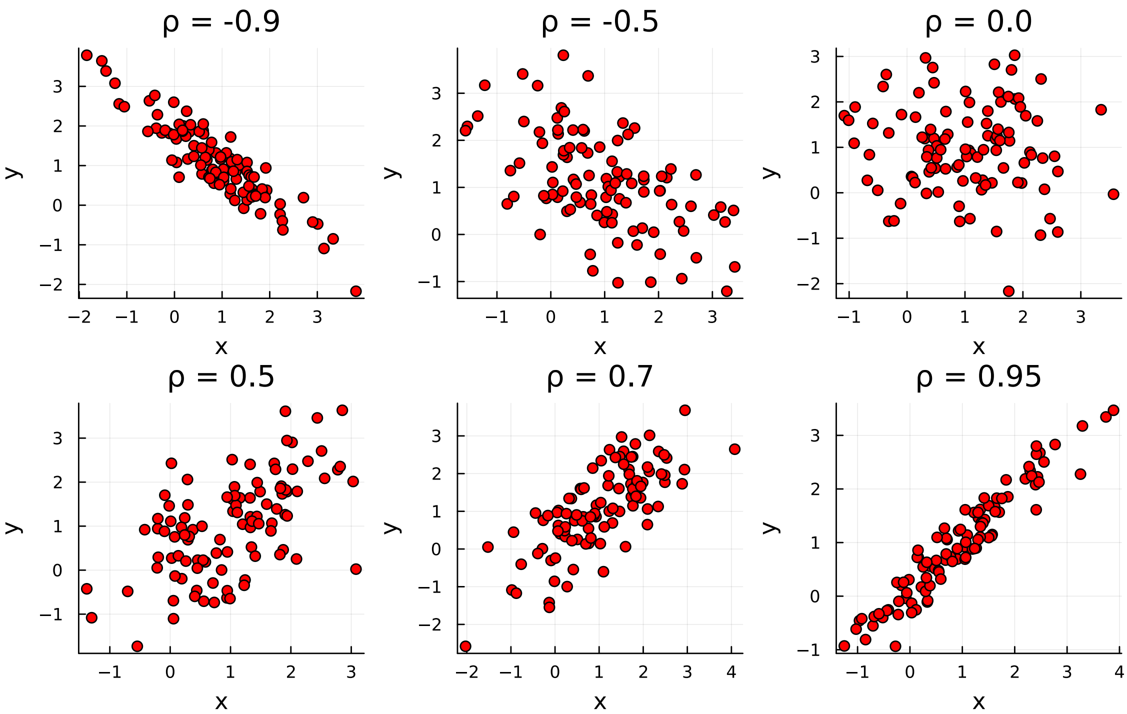

Figure 7: Simulated scatter plot of the realizations from th ebivariat enormal distribution with varying correlation coefficient.

As ρ converges to -1 or $+1$ the conditional variance $\sigma_Y^2(1-\rho^2)$ converges to 0. Therefore, the conditional distribution of Y given $X = x$ becomes more concentrated about the point $\mu_Y + \rho\left(\frac{\sigma_Y}{\sigma_X}\right)(x-\mu_X)$ and the joint probability distribution of (X, Y) becomes more concentrated about the line $y = \mu_Y + \rho\left(\frac{\sigma_Y}{\sigma_X}\right)(x-\mu_X)$.

This illustrates the fact that as correlation is close to 1 or -1, it means that there is a line $y = a + bx$ about which the values (X, Y) cluster with high probability.

4.1. Delta Method and Approximation of Ratio of Means

Suppose that we have paired observations of size n, $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ from a bivariate normal distribution. We are interested to approximate the ratio of means $\frac{\mu_X}{\mu_Y}$ and constitute a CI for the same. Assume that $\mu_Y \neq 0$. A natural approximator would be $\frac{\overline{X}_n}{\overline{Y}_n}$. We know that $\overline{X}_n \sim \mathcal{N}\left(\mu_X, \frac{\sigma_X^2}{n}\right)$ and $\overline{Y}_n \sim \mathcal{N}\left(\mu_Y, \frac{\sigma_Y^2}{n}\right)$. By Delta method

$$\mathbb{E}\left(\frac{\overline{X}_n}{\overline{Y}_n}\right) \approx \frac{\mu_X}{\mu_Y} \tag{7}$$

and

$$\mathrm{Var}\left(\frac{\overline{X}_n}{\overline{Y}_n}\right) \approx \frac{1}{n}\left(\frac{\mu_X}{\mu_Y}\right)^2 \left[\frac{\sigma_X^2}{\mu_X^2} + \frac{\sigma_Y^2}{\mu_Y^2} - 2\rho\frac{\sigma_X \sigma_Y}{\mu_X \mu_Y}\right] \tag{8}$$

and for large n

$$\frac{\frac{\overline{X}_n}{\overline{Y}_n} - \frac{\mu_X}{\mu_Y}}{\sqrt{\mathrm{Var}\left(\frac{\overline{X}_n}{\overline{Y}_n}\right)}} \sim \mathcal{N}(0,1), \quad \text{asymptotically.} \tag{9}$$

In [13]:
```
using Plots, Distributions, Random
using Statistics, StatsBase
using StatsPlots, LaTeXStrings
using DataFrames
```

In [14]:
```
plt = plot(layout = (2, 2), size = (800, 600))
n_vals = [10, 25, 50, 100]
rep = 1000
μ_x = 1  # mean of x
μ_y = 1  # mean of y
σ_x = 1  # sd of x
σ_y = 1   # sd of y
ρ = 0.8
μ = [μ_x, μ_y]
δ_mean = μ_x / μ_y
δ_var = (μ_x / μ_y)^2 * (σ_x^2 / μ_x^2 + σ_y^2 /
    μ_y^2 - 2 * ρ * σ_x * σ_y / (μ_x * μ_y))
Σ = [σ_x^2  ρ * σ_x * σ_y; ρ * σ_x * σ_y  σ_y^2]

for (idx, n) in enumerate(n_vals)
    ratio_vals = zeros(rep)
    for j in 1:rep
        data = rand(MvNormal(μ, Σ), n)
        bivariate_data = DataFrame(x = data[1, :], y = data[2, :])
        ratio_vals[j] = mean(bivariate_data.x) / mean(bivariate_data.y)
    end
    histogram!(ratio_vals, normalize = true, color = "lightgrey",
        xlabel = L"\frac{\overline{X}_n}{\overline{Y}_n}", ylabel = "density",
        xlims = extrema(ratio_vals), label = "", title = "n = $n",subplot = idx)
    plot!(x -> pdf(Normal(δ_mean, sqrt(δ_var / n)), x), color = "red", lw = 2,
```

```
        label = "", subplot = idx)
    scatter!([μ_x / μ_y], [0], color = "red", markersize = 6,
        label = "",subplot = idx)
end

display(plt)
```

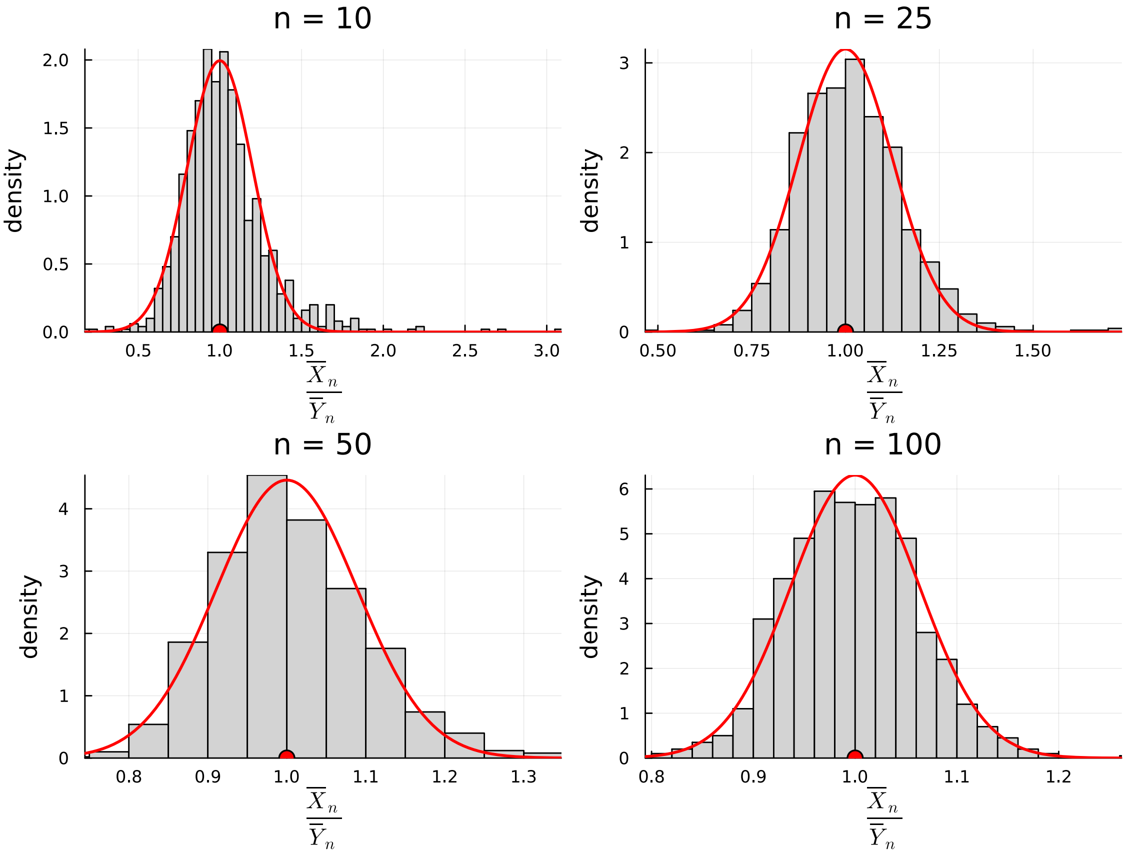

Figure 8: As $n \to \infty$, the sampling distribution of the ratio estimator tends to the normal distribution

Due to the approximation to the normality, an approximate $(1-\alpha)\%$ CI for $\frac{\mu_X}{\mu_Y}$ can be obtained as

$$\left(\frac{\overline{X}_n}{\overline{Y}_n} \pm z_{\alpha/2}\sqrt{\widehat{\operatorname{Var}}\left(\frac{\overline{X}_n}{\overline{Y}_n}\right)}\right)$$

5. Real Life Applications

5.1. The Logistic Growth Equation

Consider a natural population whose growth dynamics can be modeled using the logistic growth equation which is given below:

$$\frac{dx}{dt} = rx\left(1-\frac{x}{K}\right), \quad x(0) = x_0 \tag{10}$$

where $r > 0$ is the population intrinsic growth rate and K is the carrying capacity of the environment and x_0 is the initial population size. The solution of the differential equation is given by

$$x(t) = \frac{K}{1 + \left(\frac{K}{x_0} - 1\right) e^{-rt}}. \tag{11}$$

The point of inflection t_{inf} can be obtained by equating $x''(t) = 0$ and is given by

$$t_{\text{inf}} = \frac{1}{r} \ln\left(\frac{K}{x_0} - 1\right). \tag{12}$$

For inferring about the growth of natural populations, the identification of the point of inflection, that is, the time when the population has the maximum growth potential, is an important concept for the practitioners and conservation managers.
From the data of the population time series data, we need to estimate the parameters r, K and x_0 and then compute the t_{inf}. Suppose, we perform nonlinear least squares to obtain the parameter estimates $\hat{r}$, $\hat{K}$ and $\hat{x}_0$. Then

$$\hat{t}_{\text{inf}} = \frac{1}{\hat{r}} \ln\left(\frac{\hat{K}}{\hat{x}_0} - 1\right)$$

which is a nonlinear function of nls estimators. Therefore, this is a scenario where the Delta method can potentially be applied to approximate the sampling distribution of $\hat{t}_{\text{inf}}$. We shall do this by using simulation below.

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

In [16]:
```
t = range(0, 15, length = 40)  # time points
r = 0.5  # true value of r
K = 100
x0 = 30
r_vals = [0.3, 0.5, 1.0, 1.5]

for i in 1:length(r_vals)
    r = r_vals[i]
    x = zeros(length(t))
    x = K ./(1 .+ ((K ./x0) .-1) .*exp.(-r .*t))

    if i == 1
        p1 = plot(t, x, color = i+1, label = "r = $r", ylims = (20, 120),
            xlabel = "t", ylabel = "x", lw = 3, linestyle = :dash)
    else
        plot!(t, x, color = i+1, lw = 3, linestyle = :dash, label = "r = $r")
    end
end

scatter!([0], [x0], color = "red", markersize = 10, label = "")
hline!([K], color = "black", lw = 2, linestyle = :dash, label ="")
annotate!([2],[105], "K = 100")
```

Out[16]:

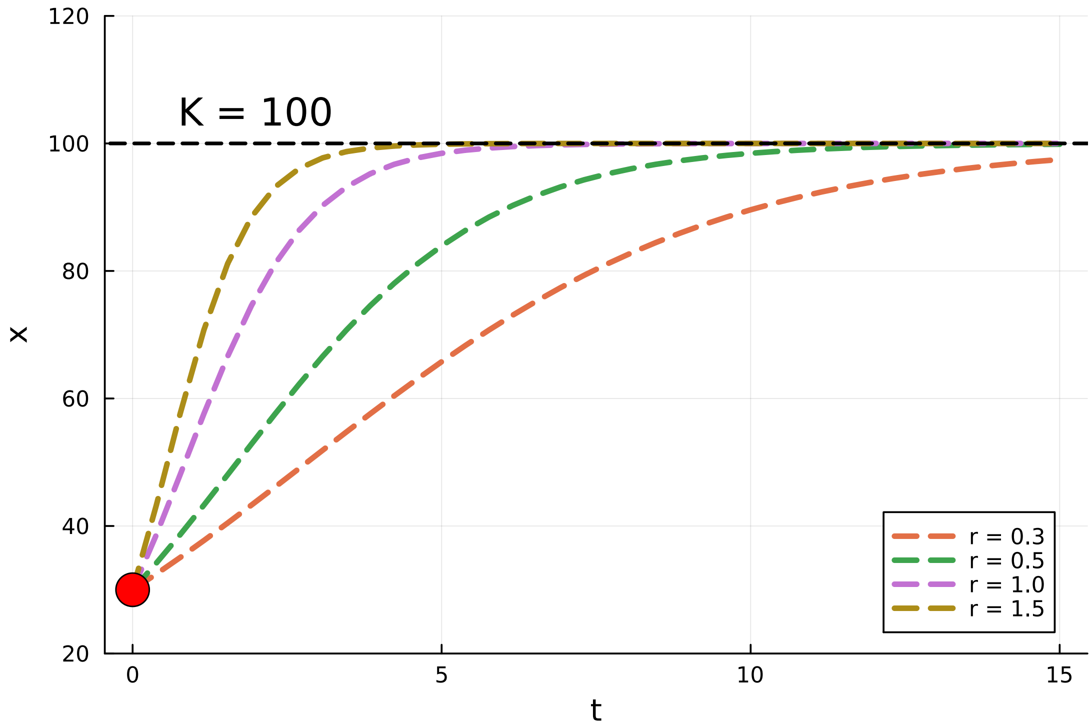

Figure 9: Growth profile of the logistically growing population with differernt values of r. Represents the mean size profile

In [17]:

```
#  Simulation under randomness is carried out belo
μ = 0
σ = 2

for i in 1:length(r_vals)
    r = r_vals[i]
    x = K ./ (1 .+ ((K / x0) .- 1) .* exp.(-r .* t)) .+
          rand(Normal(μ, σ), length(t))

    if i == 1
        p2 = plot(t, x, color = i + 1, lw = 2, linestyle = :dash, title = "",
                  ylims = (20, 110), marker = :circle,
            xlabel = "t", ylabel = "x",label = "r = $r")
    else
        plot!(t, x, color = i + 1, lw = 2, linestyle = :dash,
              marker = :circle, label = "r = $r")
    end
end

scatter!([0], [x0], color = "red", markersize = 10, label = "")
hline!([K], color = "black", lw = 2, linestyle = :dash, label = "")
```

Out[17]:

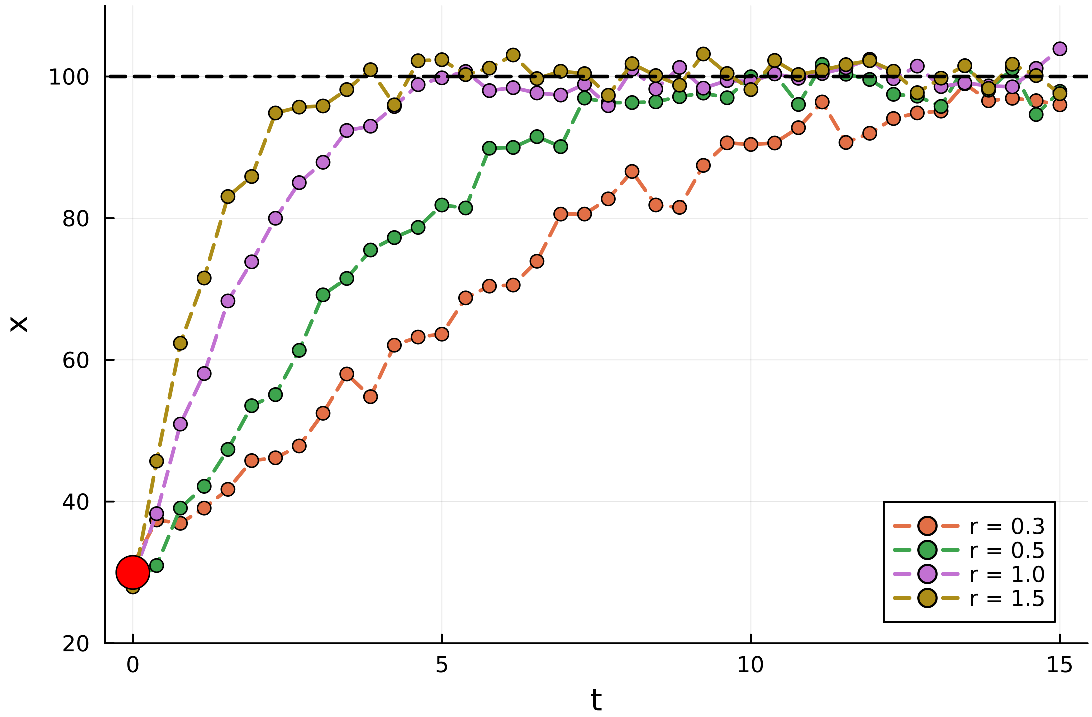

Figure 10: Growth profile of the logistically growing population with different values of r. Indicates simulated population sizes where the population size X_t is simulated using $X_t = \frac{K}{1+\left[\left(\frac{K}{x_0}-1\right)e^{-rt}\right]} + \mathcal{N}(0, \sigma^2)$. The parameter values are fixed as $K = 100$, $x_0 = 30$, $\sigma = 2$.

5.2. Model Estimation: Minimizing the Error Sum of Squares

In the previous subsection, we have discussed the simulation scheme of the population size over time following a logistic growth profile. The above simulation scheme may not be realistic as the error component is not auto correlated. Therefore, population size X_{t+1} is independent of X_t or previous population sizes. However, in real population dynamics, population size at $t+1$ depends on the previous population sizes. One can also carry out the simulation of $(X_0, X_1, \ldots, X_n)'$ by assuming by mean as the logistic growth equation and covariance structure is of $Cov(X_i, X_j) = \sigma^2\rho^{|i-j|}$, $0 \leq i, j \leq n$, where $Var(X_t) = \sigma^2$ for all t and $\rho \in (-1, 1)$ is the correlation between X_t and X_{t+1} for all t. This is known as Koopman covariance structure. Several of its applications can be found in (`Bhowmick, Chattopadhyay, and Bhattacharya 2014; Karim, Bhagat, and Bhowmick 2022`). The simulation code in the above section can be modified to include the Koopman structure by passing the modified matrix in the `MvNormal function (from Distributions.jl in Julia).`
In this section, we simulate the population size following the same scheme executed in previous section. We perform the minimization of the sum of squared errors to obtain the least squares estimates of the parameters. We use the `optimize` function, available in Julia (from the `Optim.jl` package), to perform the minimization task. We explicitly show how the estimates of the variance-covariance matrix of $\hat{x}_0$, $\hat{r}$, and $\hat{K}$ can be extracted from the output of the `optimize` function. The same can be done for any other functions as well. In addition, the following procedure is equivalent to computing the parameter estimates using the nonlinear least squares method (`curve_fit` function from `LsqFit.jl` in Julia).

In [18]:
```
using Optim, ForwardDiff, LinearAlgebra
using DataFrames, LaTeXStrings
```

In [19]:
```
using Distributions, Optim, Plots

# True parameters and time points
t = range(0, 15, length = 40)
r = 0.5
K = 100
x0 = 30
μ = 0
σ = 3

# Simulated noisy logistic growth data
x = K ./ (1 .+ ((K / x0) .- 1) .* exp.(-r .* t)) .+ rand(Normal(μ, σ), length(t)

function fun_ess(β)
    r = β[1]
    K = β[2]
    x0 = β[3]
    x_pred = K ./ (1 .+ ((K / x0) .- 1) .* exp.(-r .* t))
    return sum((x .- x_pred).^2)
end

# Optimization using Nelder-Mead
fit_optim = optimize(fun_ess, [0.6, 90, 20], NelderMead(), autodiff = :forward)

# Extract parameter estimates
par = Optim.minimizer(fit_optim)
println("Estimated parameters: r̂ = $(round(par[1], digits=3)),
    K̂ = $(round(par[2], digits=3)), x̂₀ = $(round(par[3], digits=3))")
```

```
Estimated parameters: r̂ = 0.469,
    K̂ = 99.412, x̂₀ = 32.851
```

In [20]:
```
p1 = scatter(t, x, color = "red", xlabel = "t", ylabel = "x", label = "")
x_hat =  par[2] ./ (1 .+ ((par[2] / par[3]) .- 1) .* exp.(-par[1] .* t))
plot!(t, x_hat, color = "blue", lw = 2, label = "")

er = x .- x_hat # error
p2 = histogram(er, normalize = true, color = "lightgrey", xlabel = "residuals",
ylabel = "density", label = "")
sigma2_hat = sum((er) .^2) ./(length(x) .- 3)
println(sigma2_hat)
plot!(x->pdf.(Normal(0, sqrt(sigma2_hat)),x), color = "red", lw = 2, label = "")

plot(p1, p2, layout = (1,2), size = (700, 400))
```

```
8.041840739217765
```

Out[20]:

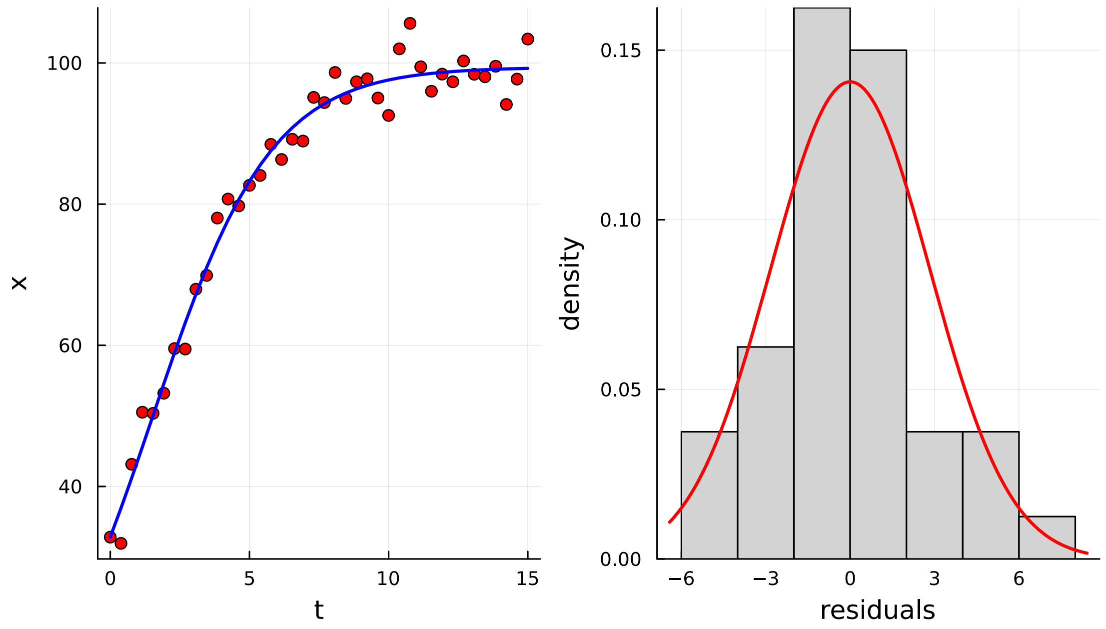

Figure 11: Fitting of the logistic growth equation to data set by direct minimization of the sum of squares. The distribution of the residuals. This should be approximately normally distributed. Parameter values are fixed as $r = 0.5$, $K = 100$, $x_0 = 30$ and $\sigma = 3$ for simulation.

In [21]:
```
b_hat = fit_optim.minimizer
H =  ForwardDiff.hessian(fun_ess, b_hat)  # hessian Matrix
est_cov = 2*sigma2_hat*inv(H)
labels = [L"r", L"K", L"x_0"]
est_cov_df = DataFrame(est_cov, Symbol.(labels))
est_cov_df = hcat(DataFrame(Row = labels), est_cov_df)
println(est_cov_df)
```

```
3×4 DataFrame
 Row │ Row          $r$           $K$         $x_0$
     │ LaTeXStr…   Float64       Float64     Float64
─────┼──────────────────────────────────────────────────
   1 │ $r$          0.000564745  -0.011386   -0.0252659
   2 │ $K$         -0.011386      0.61308     0.338708
   3 │ $x_0$       -0.0252659     0.338708    1.75908
```

5.3. The Sampling Distribution of the Estimators

We can visualize the sampling distribution of the nonlinear least squares estimators of the parameters by simulating the population size a large number of times by keeping parameters at some fixed values. For each simulation, we will get an estimate by minimizing the sum of squares. This estimate is a realization from the sampling distribution of the estimators. If the process is replicated 1000 times, then histogram of 1000 values of the estimators will give a good approximation of the sampling distribution.

In [22]:
```
using Optim, ForwardDiff, LinearAlgebra
using DataFrames, LaTeXStrings
```

In [23]:
```
rep = 1000  # number of replications
est_r = zeros(rep)
est_K = zeros(rep)
```

```
est_x0 = zeros(rep)

for i in 1:rep
    x = K ./ (1 .+ ((K / x0) .- 1) .* exp.(-r .* t)) .+
    rand(Normal(μ, σ), length(t))
    fit_optim = optimize(fun_ess, [0.6, 90, 20], NelderMead(),
        autodiff = :forward)
    par = Optim.minimizer(fit_optim)
    est_r[i] = par[1]
    est_K[i] = par[2]
    est_x0[i] = par[3]
end
```

In [24]:

```
p1 = histogram(est_r, normalize = true, xlabel = L"\widehat{r}",
ylabel = "density", ylims = (0, 17),label = "")
scatter!([r],[0], color = "red", markersize = 10, label = "")
plot!(x->pdf.(Normal(r, std(est_r)), x), color = "red", lw = 2, label = "")

p2 = histogram(est_K, normalize = true, xlabel = L"\widehat{K}",
ylabel = "density", ylims = (0, 0.5),label = "")
scatter!([K],[0], color = "red", markersize = 10, label = "")
plot!(x->pdf.(Normal(K, std(est_K)), x), color = "red", lw = 2, label = "")

p3 = histogram(est_x0, normalize = true, xlabel = L"\widehat{x}_0",
ylabel = "density", ylims = (0, 0.3),label = "")
scatter!([x0],[0], color = "red", markersize = 10, label = "")
plot!(x->pdf.(Normal(x0, std(est_x0)), x), color = "red", lw = 2, label = "")

plot(p1, p2,p3, layout = (2,2), size = (800, 500))
```

Out[24]:

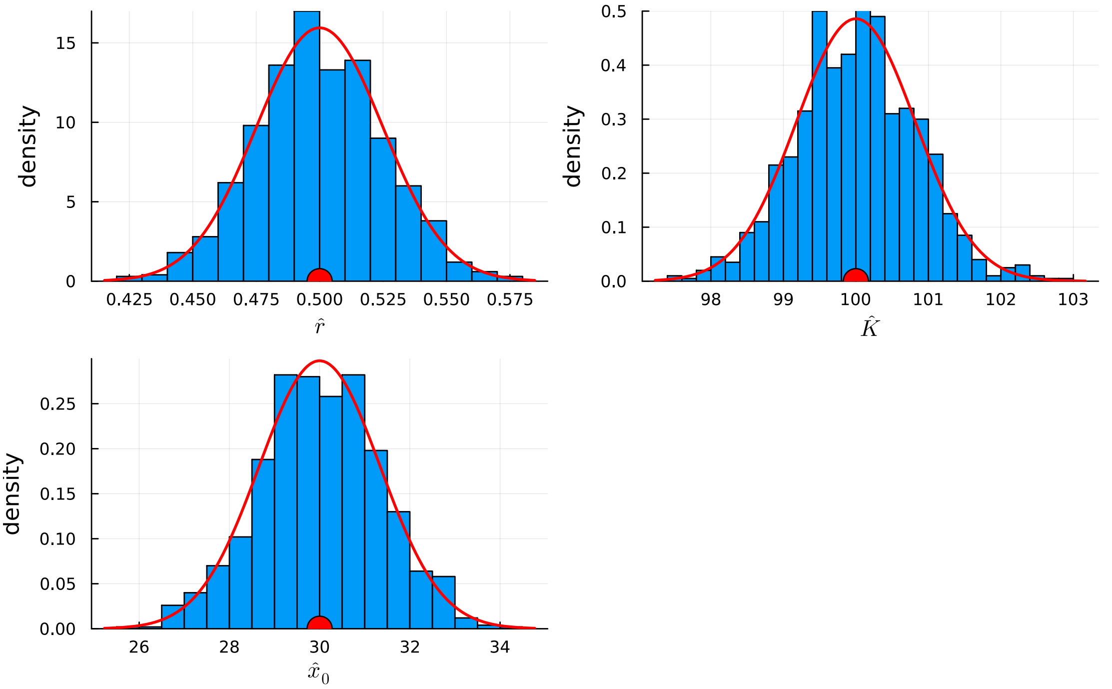

Figure 12: The sampling distributions of the nonlinear least squares estimators are well approximated by the normal distribution. In addition, the histograms are centered about the true parameters values (represented using red dots). True parameter values are fixed as $r = 0.5$, $K = 100$, $x_0 = 30$, and $\sigma = 3$ for simulation.

We can apply the **Multivariate Delta method** to obtain an approximate sampling distribution of $\hat{t}_{\text{inf}}$. The partial derivatives of $t_{\text{inf}} = g(r, K, x_0)$ are given as

$$g_r = -\frac{1}{r^2}\ln\left(\frac{K}{x_0} - 1\right), \quad g_K = \frac{1}{r}\cdot\frac{1}{x_0}, \quad g_{x_0} = \frac{1}{r}\cdot\frac{1}{\left(\frac{K}{x_0} - 1\right)}\left(-\frac{K}{x_0^2}\right)$$

By Delta method:

$$\mathbb{E}\left(\hat{t}_{\text{inf}}\right) \approx t_{\text{inf}}$$

and

$$\text{Var}\left(\hat{t}_{\text{inf}}\right) \approx \sigma_{\hat{r}}^2 (g_r)^2 + \sigma_{\hat{K}}^2 (g_K)^2 + \sigma_{\hat{x}_0}^2 (g_{x_0})^2 + 2\left[\sigma_{\hat{r}\hat{K}} g_r g_K + \sigma_{\hat{K}\hat{x}_0} g_K g_{x_0} + \sigma_{\hat{r}\hat{x}_0} g_r g_{x_0}\right]$$

where σ_{XY} denotes the covariance between X and Y.

Remark:

In the previous example, we dealt with the approximation of the ratio of means for bivariate normal distribution; we knew about the sampling distribution of the $\bar{X}_n$ and $\bar{Y}_n$ and their covariances as well. Therefore, we could verify the Delta method using simulation studies. In the approximation of the point of inflection problem, we do not know the exact variance of $\hat{r}$, $\hat{K}$ and $\hat{x}_0$, we have only the estimates based on a single simulation.
Therefore, very close approximation of the true values of $\text{Var}(\hat{r})$, $\text{Var}(\hat{K})$, $\text{Var}(\hat{x}_0)$ have been computed by performing the optimization 1000 times and computing the variance and covariance of $\hat{r}$, $\hat{K}$ and $\hat{x}_0$ based on these 1000 estimates.

.

In [37]:

```
using Distributions, Statistics, Optim, Plots, LaTeXStrings

rep = 1000  # number of replications
r = 0.5
K = 100.0
x0 = 30.0
μ = 0.0
σ = 3.0
t = range(0, stop=10, length=50)

tinf = (1/r) * log(K / x0 - 1)
est_r = zeros(rep)
est_K = zeros(rep)
est_x0 = zeros(rep)
est_tinf = zeros(rep)

for i in 1:rep
    x = K ./ (1 .+ ((K / x0) .- 1) .* exp.(-r .* t)) .+
        rand(Normal(μ, σ), length(t))
    fit_optim = optimize(fun_ess, [0.6, 90, 20],
        NelderMead(); autodiff = :forward)
    par = Optim.minimizer(fit_optim)
```

```
    est_r[i] = par[1]
    est_K[i] = par[2]
    est_x0[i] = par[3]
end

est_tinf .= (1 ./ est_r) .* log.(est_K ./ est_x0 .- 1)

histogram(est_tinf, normalize = true,xlabel = L"\widehat{t}_{\text{inf}}",
    ylabel = "density", ylims = (0, 6),  bins = 30, label = "")

# partial derivative with respect to r
g_r(r, K, x0) = (-1 / r^2) * log(K / x0 - 1)

# partial derivative with respect to K
g_K(r, K, x0) = (1 / r) * (1 / (K / x0 - 1)) * (1 / x0)

# partial derivative with respect to x0
g_x0(r, K, x0) = (1 / r) * (1 / (K / x0 - 1)) * (-K / x0^2)

var_tinf = var(est_r) * g_r(r, K, x0)^2 +
           var(est_K) * g_K(r, K, x0)^2 +
           var(est_x0) * g_x0(r, K, x0)^2 +
           2 * (cov(est_r, est_K) * g_r(r, K, x0) * g_K(r, K, x0) +
                cov(est_K, est_x0) * g_K(r, K, x0) * g_x0(r, K, x0) +
                cov(est_r, est_x0) * g_r(r, K, x0) * g_x0(r, K, x0))

plot!(x -> pdf.(Normal(tinf, sqrt(var_tinf)), x), color = "red", lw = 2, label =
```

Out[37]:

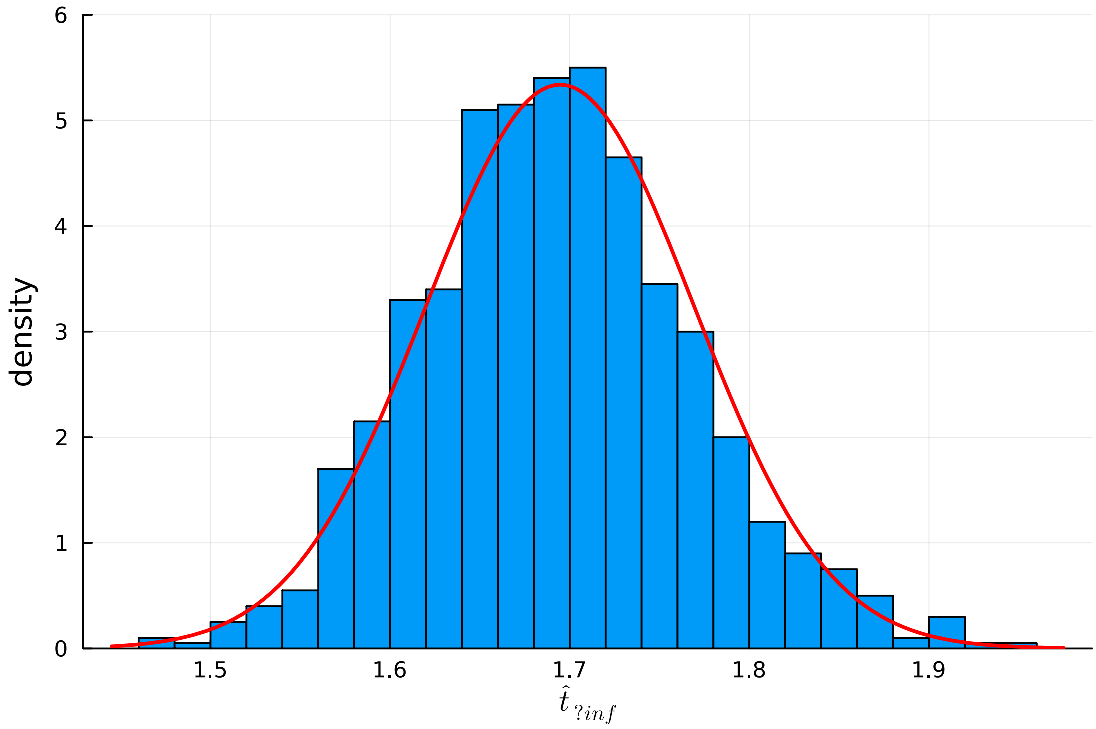

Figure 13 :The approximation of the sampling distribution by the Delta method for the point of inflection. We observe that the approximation by the normal distribution is very good.

Remark

For real data applications, an approximate confidence interval with confidence coefficient $(1-\alpha)$ for the point of inflection can be given as:

$$\left(\hat{t}_{\text{inf}} - z_{\alpha/2} \cdot \mathrm{SE}\left(\hat{t}_{\text{inf}}\right), \quad \hat{t}_{\text{inf}} + z_{\alpha/2} \cdot \mathrm{SE}\left(\hat{t}_{\text{inf}}\right)\right)$$

which will give an idea of the time interval when the population had grown at a maximum rate. The reader is encouraged to perform a simulation exercise in which she will be testing whether the approximate confidence coefficient of the above normality-based interval is $1 - \alpha$. The algorithm can be implemented as follows:

1. Fix the parameters x_0, r, K and σ.
2. Fix the number of replications, M (say), and length of the time series $n + 1$.
3. Simulate M time series of length $n + 1$ using the method as described in [Real life applications].
4. For each of the M series, obtain the estimates $\widehat{x}_0$, $\hat{r}$, $\widehat{K}$ and the point of inflection $\hat{t}_{\text{inf}}$ and construct the confidence interval for t_{inf} using the above formula.
5. Compute the proportion of CIs obtained in Step 4 that contain the true value t_{inf}. This estimated proportion should be approximately $(1 - \alpha)$ for large M.

6. Conclusion

In summary, the Delta method is quite a powerful method. It demonstrates extraordinary utility of the Taylor’s polynomial in statistical applications. In real data applications, the use of nonlinear functions are widespread and the Delta method is a useful tool to approximate the uncertainty associated with the applications of nonlinear functions. Another interesting phenomena is the widespread application of the Gaussian distribution. It is a wonders of nature only that by God’s grace, almost in every scenario we are ending with the normal distribution.