Loss Function and the Risk Function

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

Introduction

In the study of statistical computing, a fundamental principle is that the ultimate objective of statistics is to uncover the truth. However, this truth remains beyond complete human reach—not only in the present but also in the foreseeable future. When data is collected from natural processes, the aim is to understand the underlying mechanisms of nature through mathematical models. Yet, all efforts based on observed data serve only as approximations of nature's true functioning, as achieving absolute accuracy is inherently impossible.
This naturally leads to an important question: If the absolute truth is unattainable, how can we validate statistical approximations and assess their accuracy? This question lies at the very core of understanding the principles and philosophy of statistical science.
However, we need to devise strategies to evaluate how accurate statistical approximations are in estimating the unknown parameters. Suppose that we have a random sample of size n from the population density function $f(x;\theta)$ for $\theta \in \Theta$. We can approximate θ based on some function $\psi(X_1,\dots,X_n)$, based on a sample of size n. We need to measure the closeness of the estimator ψ_n to θ.

Simulation experiment with Bernoulli distribution

In this section, we approximate the risk function of the sample proportion as an estimator of the true proportion when sampling from the Bernoulli(p) populations. We perform this task in three steps.

- **Step - I**
  - Fix the parameter p
  - Fix the sample size n
  - Simulation $X_1, X_2, \dots, X_n \sim \mathrm{binomial}(1,p)$.
  - Compute the loss $(\hat{p}_n - p)^2$

Repeat the following codes multiple times to realize that the loss function $l(\hat{p}_n, p)$ is a random variable.

In [1]:

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

In [2]:

```julia
p = 0.1 # true probability
n = 10 # sample size
x = rand(Binomial(1,p),n)  # simulation
println(first(x, 6))
```

```
[0, 0, 0, 0, 0, 0]
```

In [3]:

```
pn_hat = mean(x)
```

Out[3]:

```
0.0
```

In [4]:

```
(pn_hat-p)^2  # computing squared loss
```

Out[4]:

```
0.010000000000000002
```

- **Step II**

To compute the average loss, we need to repeat the above process M times (say) to get an estimate of the risk at a specific value of p.

In [5]:

```
M = 1000   # number of replications
pn_loss = zeros(M)

for m in 1:M
    x = rand(Binomial(1,p),n)  # simulation
    pn_hat = mean(x)
    pn_loss[m] = (pn_hat-p)^2
end

println(first(pn_loss, 6))
```

```
[0.010000000000000002, 0.010000000000000002, 0.010000000000000002, 0.01000000000000000
2, 0.010000000000000002, 0.010000000000000002]
```

In [6]:

```
pn_risk = mean(pn_loss)  # compute the risk
```

Out[6]:

```
0.009430000000000003
```

Step - I*and Step - II have been carried out for a fixed value of p. If we want to obtain the performance of the estimator irrespective of the true value, we must evaluate its performance for each $p \in (0,1)$, which will give us the risk profile of the estimator. This is also called a risk function, which is a function of the parameter p and contains no randomness.

$$R(\hat{p}_n, p) = \mathbb{E}\left(l(\hat{p}_n, p)\right).$$

We can plot this function as a function of p.

- **Step - III**
  - Discretize the parameter space $(0,1)$ into distinct points $p_1 < p_2 < \ldots < p_L$.
  - For each p_j, $1 \leq j \leq L$, perform **Step - I** and **Step - II**.

In [7]:

```
prop_vals = 0.01:0.01:0.99
pn_risk = zeros(length(prop_vals))
M = 1000

for i in 1:length(prop_vals)
    p = prop_vals[i]
    pn_loss = zeros(M)
    for m in 1:M
        x = rand(Binomial(1,p),n)  # simulation
        pn_hat = mean(x)
        pn_loss[m] = (pn_hat-p)^2
    end
    pn_risk[i] = mean(pn_loss)
end

println(first(pn_risk, 6))
```

```
[0.0009720000000000009, 0.0019059999999999997, 0.0028200000000000026, 0.00377199999999
99963, 0.004840000000000005, 0.005627999999999995]
```

In [8]:

```
scatter(prop_vals, pn_risk, xlabel = "p", markersize = 5, color = "red",
    ylabel = L"R{(\hat{P_n},P)}", label = "", title = "n = $n" )
plot!(x -> (x * (1 - x) / n), color = "blue", lw = 2, label = "")
```

Out[8]:

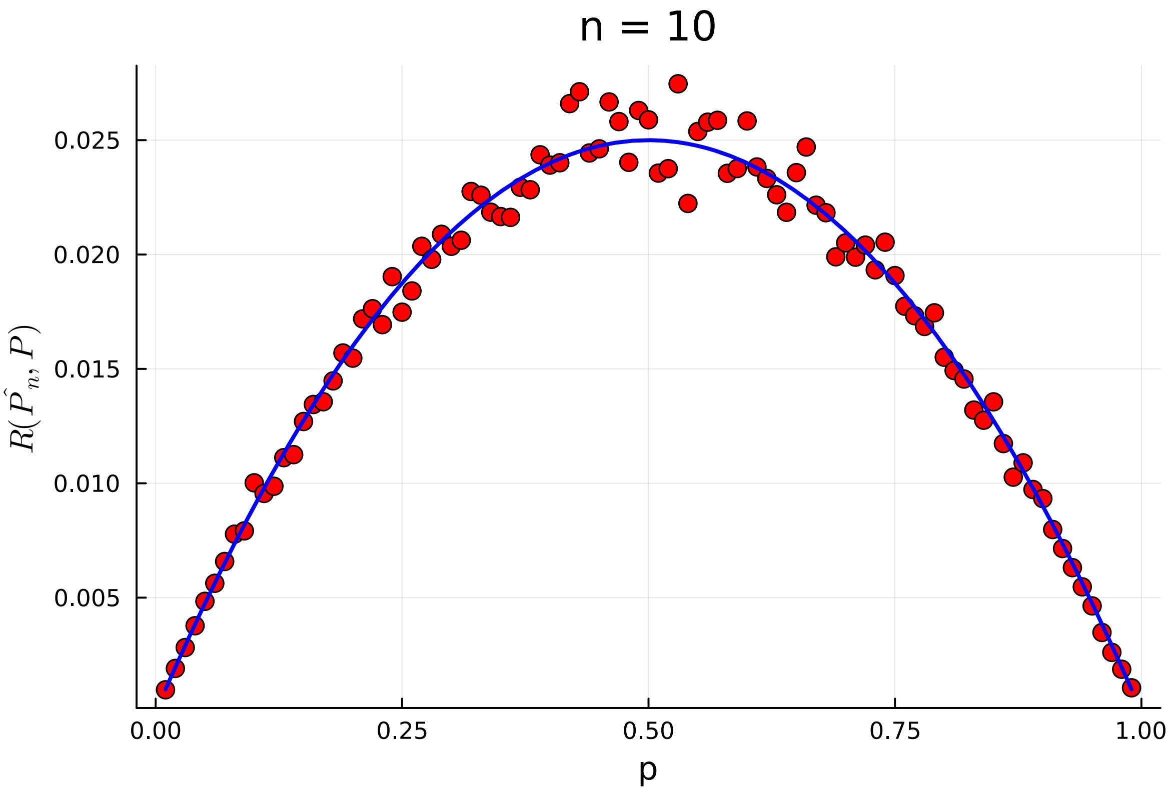

Figure 1: The simulated risk function of the MLE of the probability of head *p* under the squared error loss. The risk, that is, expected loss has been approximated based on the 1000 replications.

Basically, the risk function says that on average, what will be the mistake in squared error scale, if we replace the true value of p with the sample proportion.

- **Step - IV**

A natural question arises: what will happen if we change the sample size n? Intuition suggests that as n increases, the distance between the truth and estimate would be small, that $\mathcal{R}(\hat{p}_n, p)$ would be a decreasing function of n at each $p \in (0, 1)$. In the following code, we see the behavior of the risk function for different choices of p.

In [9]:

```
n_vals = [5, 10, 20, 30, 50, 100]
M = 1000
plt = plot(layout=(2, 3), size=(800, 600))

for (idx, n) in enumerate(n_vals)
    prop_vals = 0.01:0.01:0.99
    pn_risk = zeros(length(prop_vals))
    for i in 1:length(prop_vals)
        p = prop_vals[i]
        pn_loss = zeros(M)
        for m in 1:M
            x = rand(Binomial(1, p), n)
            pn_hat = mean(x)
            pn_loss[m] = (pn_hat - p)^2
        end
        pn_risk[i] = mean(pn_loss)
    end
    scatter!(prop_vals, pn_risk, color="red", markersize=3,
        xlabel="p", ylabel=L"R(\hat{p_n}, p)", label = "",
        title="n = $n", ylim=(0, 0.055), subplot=idx)
end

display(plt)
```

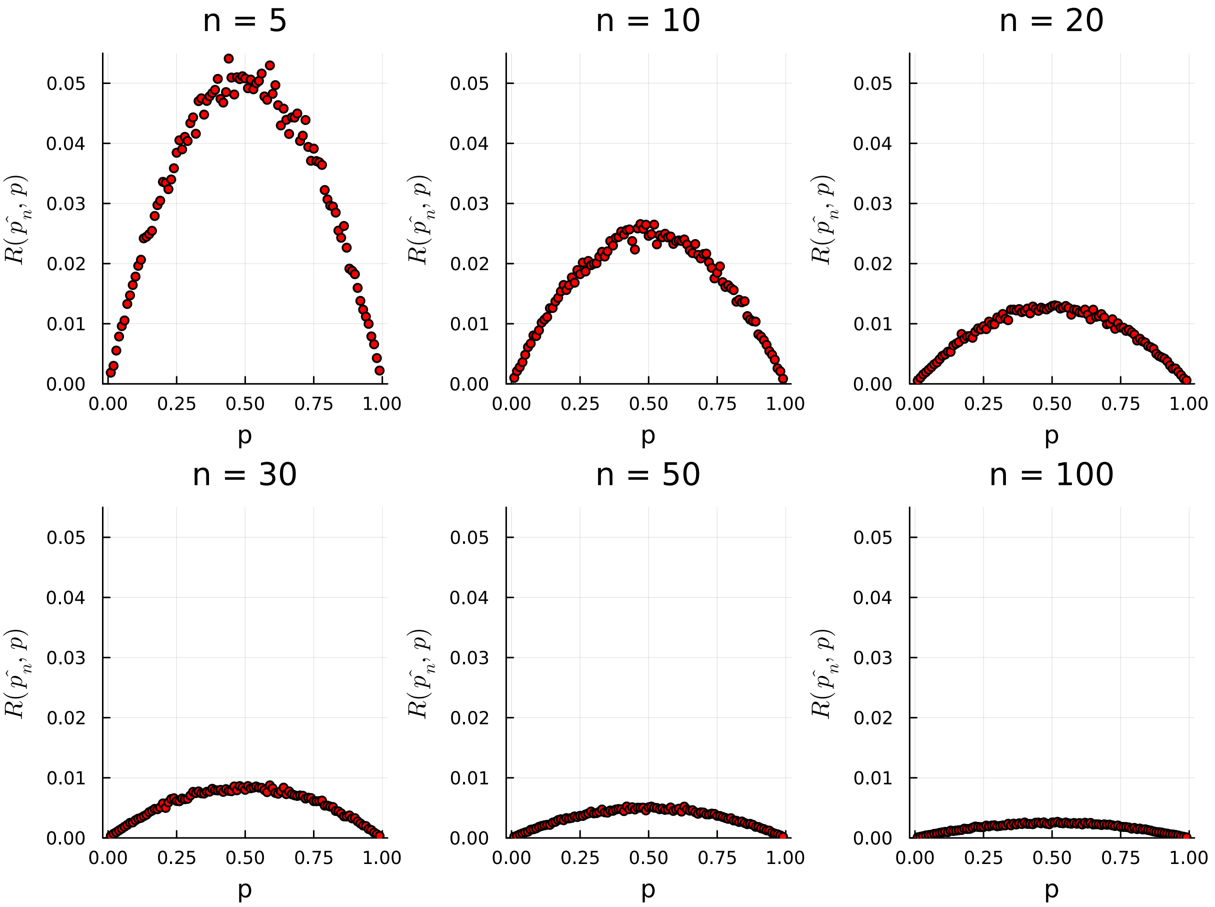

Figure 2: The risk function is obtained by using computer simulation based on 1000 replications and for different sample sizes n. It is clear that as $n \to \infty$, the risk goes to zero.

It would be interesting to see whether $\mathcal{R}(\hat{p}_n, p) \to 0$ as $n \to \infty$ for all $p \in (0, 1)$. Theoretical computation can help us in establishing/disproving this claim. The simulated shapes actually support the theoretical computations as well. The theoretical computation

$$\mathbb{E}(\hat{p}_n) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i) = \frac{1}{n}np = p.$$

Therefore,

$$\mathbb{E}\left(\overline{X_n} - p\right)^2 = \mathrm{Var}_p\left(\overline{X_n}\right) = \frac{1}{n^2}n \times \mathrm{Var}(X_1) = \frac{p(1-p)}{n}.$$

In the above simulated plots of the risk profiles for different choices of n, you can add the curve $p(1-p)/n$ and see that the curve is in well agreement with the simulation.

Simulation experiment with normal distribution

Suppose that we have a random sample of size n from the $\mathcal{N}(\mu, 1)$ population distribution. We aim to estimate the parameter μ using two different estimators $T_1 = \overline{X_n}$ and $T_2 = \text{Median}(X_1, \ldots, X_n)$.
We are aiming to compare the performance of these two estimators at different values of the unknown parameter μ. Therefore, we compute

$$\mathbb{E}(T_1 - \mu)^2 = R(T_1, \mu) \quad \text{and} \quad \mathbb{E}(T_2 - \mu)^2 = R(T_2, \mu).$$

Consider an interval of μ values of your own choice and compute the risk functions using computer simulations and plot them in a single plot window. What is the conclusion about the choice of the estimator?

Sample Mean or Sample Median

In [10]:

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

In [11]:

```julia
n = 10
sigma = 1  # fixed
a = -1
b = 1
M = 1000 # no. of replicates
mu = a

x = rand(Normal(mu,sigma), n)
mean(x)
```

Out[11]:

```
-1.228009985455773
```

In [12]:

```julia
median(x)
```

Out[12]:

```
-1.453245440907367
```

In [13]:

```julia
sample_mean = zeros(M)
sample_median = zeros(M)

for m in 1:M
    x = rand(Normal(mu,sigma), n)
    sample_mean[m] = mean(x)
    sample_median[m] = median(x)
end
```

In [14]:

```julia
mean((sample_mean .- mu) .^ 2)
```

Out[14]:

```
0.09673149365969101
```

In [15]:

```julia
mean((sample_median .- mu) .^ 2)
```

Out[15]:

```
0.13831057055599366
```

In [16]:

```
p1 = histogram(sample_mean, normalize = true, color = "lightgrey",
bins = 30, xlabel = L"\bar{X_n}",title = "n = $n",label = "")
p2 = histogram(sample_median, normalize = true, color = "lightgrey",
bins = 30, xlabel = L"Med({X_n})",title = "n = $n",label = "")
plot(p1, p2, layout = (1, 2), size = (800, 400))
```

Out[16]:

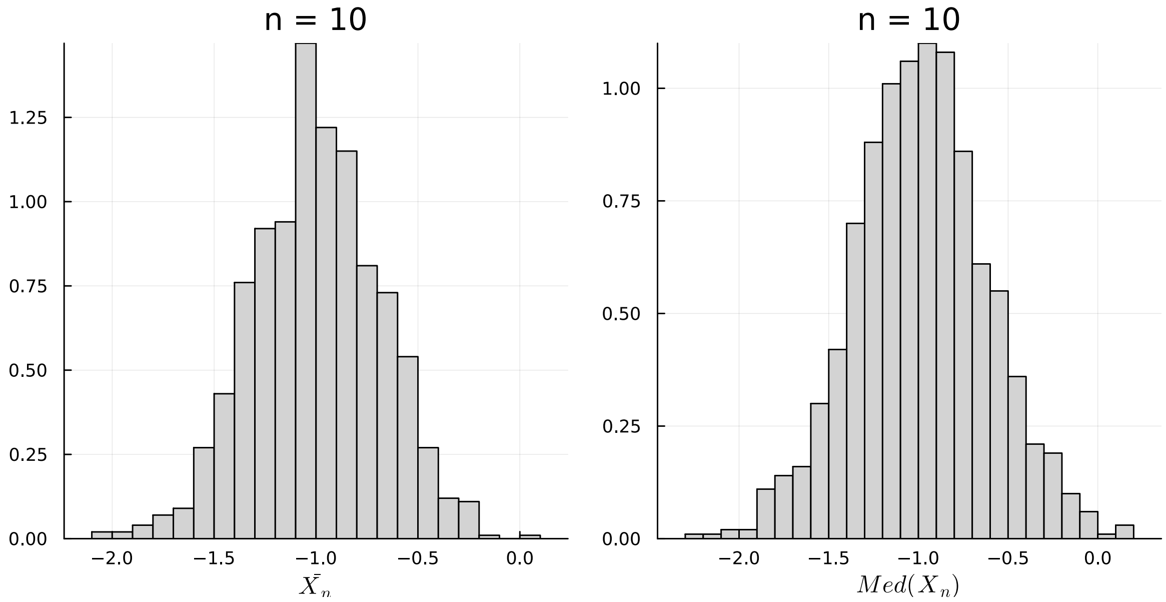

Figure 3: The sampling distribution of the sample mean and the sample median have been simulated for sample size $n = 10$ based on 1000 replications. We consider the normal distribution with mean -1 and variance 1 for demonstration.

As the value of μ changes, we need to check the performance of the sample median and the sample mean

In [17]:

```
mu_vals = a:0.1:b
risk_mean = zeros(length(mu_vals))
risk_median = zeros(length(mu_vals))
M = 1000
for i in 1:length(mu_vals)
    sample_mean = zeros(M)
    sample_median = zeros(M)

    for m in 1:M
        x = rand(Normal(mu,sigma), n)
        sample_mean[m] = mean(x)
        sample_median[m] = median(x)
    end
    risk_mean[i] = mean((sample_mean .- mu) .^ 2)
    risk_median[i] = mean((sample_median .- mu) .^ 2)
end
```

In [18]:

```
scatter(mu_vals, risk_median, color = "red", xlabel = L"\mu",
    markersize = 6,ylabel = L"R_T(\mu)", label = L"T_1",
    ylim = (0.00, 0.16))
scatter!(mu_vals, risk_mean, color = "blue", markersize = 6,
    label = L"T_2")
```

Out[18]:

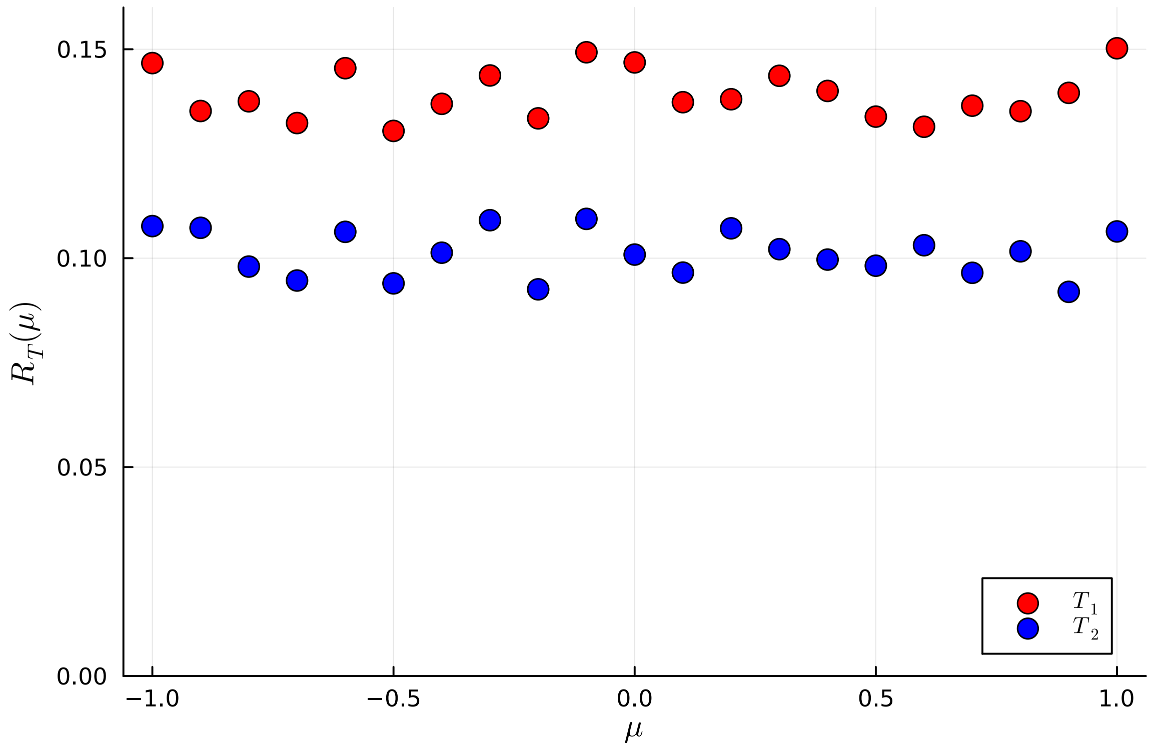

Figure 4: The risk values will be constant when the number of replications is large, that is, $M \to \infty$. From the simulation, it is clear that the sample mean has uniformly lesser risk than the sample median when used as an estimator for estimating the mean of a normally distributed population. Students are encouraged to experiment with this simulation for different values of μ and sample size n.

A desirable property of the sample mean and sample median would be to become close to the true value of μ as $n \to \infty$. Using the following visualization, we can check whether these estimators are consistent estimators of the population mean. The following simulation suggests that indeed both of them are consistent, however, fluctuations about the true value are more for the sample median as compared to the sample mean. The following algorithm is implemented to check the consistency.

- Fix $\mu = \mu_0$
- Fix σ^2
- Consider $n \in \{1, 2, \ldots, n_{\max}\}$ (sample size)
- For each n,
  - Simulate $X_1, X_2, \ldots, X_n \sim \mathcal{N}(\mu_0, \sigma^2)$
  - Compute Sample Mean $\overline{X}_n$
  - Compute Sample Median $\operatorname{Med}(X_n)$
- Plot $(n, \overline{X}_n)$ and $(n, \operatorname{Med}(X_n))$ for $n \in \{1, 2, \ldots, n_{\max}\}$.

In [19]:

```julia
n_vals = 1:1000
mu = 0
sigma = 1
sample_mean = zeros(length(n_vals))
sample_median = zeros(length(n_vals))

for n in n_vals
    x = rand(Normal(mu, sigma), n)
    sample_mean[n] = mean(x)
    sample_median[n] = median(x)
```

```
end

p1 = plot(n_vals, sample_mean, color = "lightgrey",
    xlabel ="sample size" ,title = L"\bar{X_n}", label = "" )
hline!([mu], color = "blue", lw = 2, linestyle = :dash, label = "")

p2 =  plot(n_vals, sample_median, color = "lightgrey",
    xlabel ="sample size" ,title = L"Med(X_n)", label = "" )
hline!([mu], color = "blue", lw = 2, linestyle = :dash, label = "")

plot(p1, p2, layout = (1, 2), size = (800, 400))
```

Out[19]:

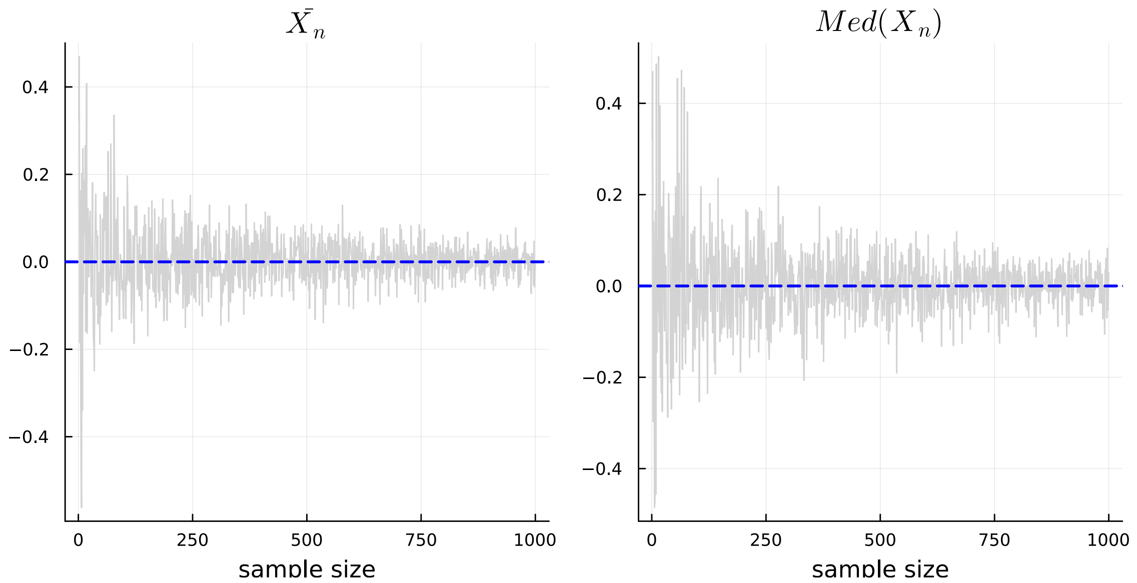

Figure 5: As the sample size increases, the sample mean and sample median converge to the true value μ as $n \to \infty$

> **Asymptotic Comparison of Sample Mean and Sample Median**

If $(X_1, X_2, ..., X_n)$ be a random sample of size n from the $\mathcal{N}(\mu, \sigma^2)$. Then it can be shown that

$$\sqrt{n}\left(\overline{X_n} - \mu\right) \xrightarrow{D} \mathcal{N}\left(0, \sigma^2\right)$$

and

$$\sqrt{n}\left(\mathrm{Med}(X_n) - \mu\right) \xrightarrow{D} \mathcal{N}\left(0, \sigma^2 \frac{\pi}{2}\right).$$

This result also supports the simulation experiment which showed that both sample mean and sample median converged to the true value of μ, however, sample median has a larger variance as compared to the sample mean.

Simulation experiment with poisson distribution

Sample Mean or Sample Median

To estimate the parameters of the population, the method of moments is a way to obtain estimator(s). In this method, population moments are made equal to the sample moments

and the population parameters are expressed as the function of samples.
If the population distribution follows the Poisson distribution with mean λ, then to obtain the MoM estimator of λ, we equate the population mean and the sample mean $\overline{X_n}$, which gives the following equation:

$$\overline{X_n} = \mathbb{E}(X) = \lambda.$$

Therefore, the MoM moment estimator of λ is $\overline{X_n}$. However, one can also observe that the population variance is also λ, therefore, equating the sample variance S_n^2 with the population variance (λ), we find that the sample variance is also an MoM estimator of λ. Therefore, the first conclusion is that the Method of Moment estimator is not unique.
A natural question arises: which estimator to prefer in estimating the unknown parameter λ ? We have the following observations:

$$\mathbb{E}\left(\overline{X_n}\right) = \lambda = \mathbb{E}\left(S_n^2\right).$$

Therefore, both are unbiased estimators of λ. We now approximate the risk functions corresponding to $\overline{X_n}$ and S_n^2, which are denoted by $\mathcal{R}\left(\overline{X_n}, \lambda\right)$ and $\mathcal{R}\left(S_n^2, \lambda\right)$, respectively, for $\lambda \in (0, \infty)$.
In the following

In [20]:

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

In [21]:

```
n = 5
M = 1000
lambda = 1
sample_mean = zeros(M)
sample_var = zeros(M)

for i in 1:M
    x = rand(Poisson(lambda), n)
    sample_mean[i] = mean(x)
    sample_var[i]  = var(x)
end

p1 = histogram(sample_mean, normalize = :pdf ,xlabel = L"\overline{X_n}",
    ylabel = "density", title = "n = $n",label = "")
scatter!([lambda], [0], color = "red", markersize = 6, label = "")

p2 = histogram(sample_var, normalize = :pdf ,xlabel = L"S_n^2",
    ylabel = "density", title = "n = $n",label = "")
scatter!([lambda], [0], color = "red", markersize = 6, label = "")

plot(p1, p2, layout = (1,2), size = (800, 400))
```

Out[21]:

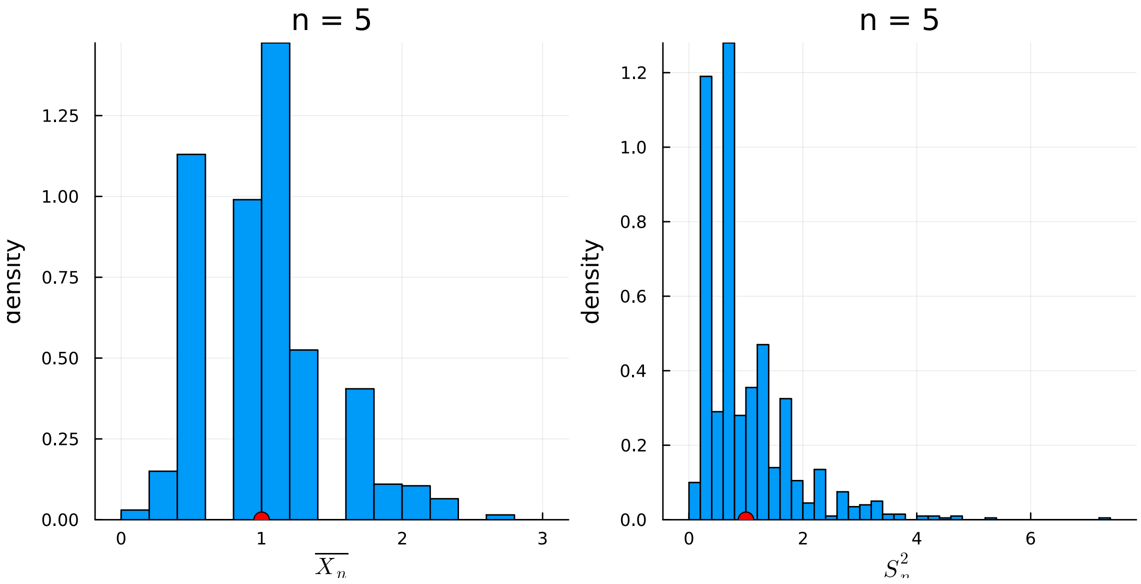

Figure 6: The simulated histograms are obtained based on 1000 replications. The population parameter λ is fixed at 1 and the sample size $n = 5$ is fixed for simulation. The histograms clearly suggest that the spread is more for the sample variance as compared to the sample mean about the true value of λ, which is denoted by the red dot.

In [22]:

```
## computing the risk values
risk_sample_mean = mean((sample_mean .- lambda).^2)

println("The estimated risk the sample mean under squared error
    loss function is : ",
    risk_sample_mean)
```

```
The estimated risk the sample mean under squared error
    loss function is : 0.20051999999999999
```

In [23]:

```
risk_sample_var = mean((sample_var .- lambda).^2)
println("The estimated risk of the sample variance under squared error loss
    function is :", risk_sample_var)
```

```
The estimated risk of the sample variance under squared error loss
    function is :0.6324800000000003
```

Since, we do not know what is the true value of λ, in the following code, we performed the same task at different choices of λ. For simulation, purpose, we discretize the (0.1,3) interval for possible choices of λ

In [24]:

```
n = 5
M = 1000
lambda_vals = 0.1:0.03:2
risk_sample_mean = zeros(length(lambda_vals))
risk_sample_var = zeros(length(lambda_vals))

for j in 1:length(lambda_vals)
    lambda = lambda_vals[j]
    sample_mean = zeros(M)
    sample_var = zeros(M)
    for i in 1:M
        x = rand(Poisson(lambda), n)
        sample_mean[i] = mean(x)
        sample_var[i] = var(x)
    end
    risk_sample_mean[j] = mean((sample_mean .- lambda).^2)
    risk_sample_var[j] = mean((sample_var .- lambda).^2)
end

scatter(lambda_vals, risk_sample_var, color = "red", markersize = 8,
    xlabel = L"\lambda", ylabel = L"R_T(\lambda)", label = L"S_n^2")
plot!(lambda_vals -> lambda_vals/n+ 2*lambda_vals^2/(n-1), color = "black",
lw = 3, linestyle = :dash, label = "")
scatter!(lambda_vals, risk_sample_mean, color = "magenta", markersize = 8,
    label = L"\overline{X_n}")
plot!(lambda_vals -> lambda_vals/n, color = "black",
lw = 3, linestyle = :dash, label = "")
```

Out[24]:

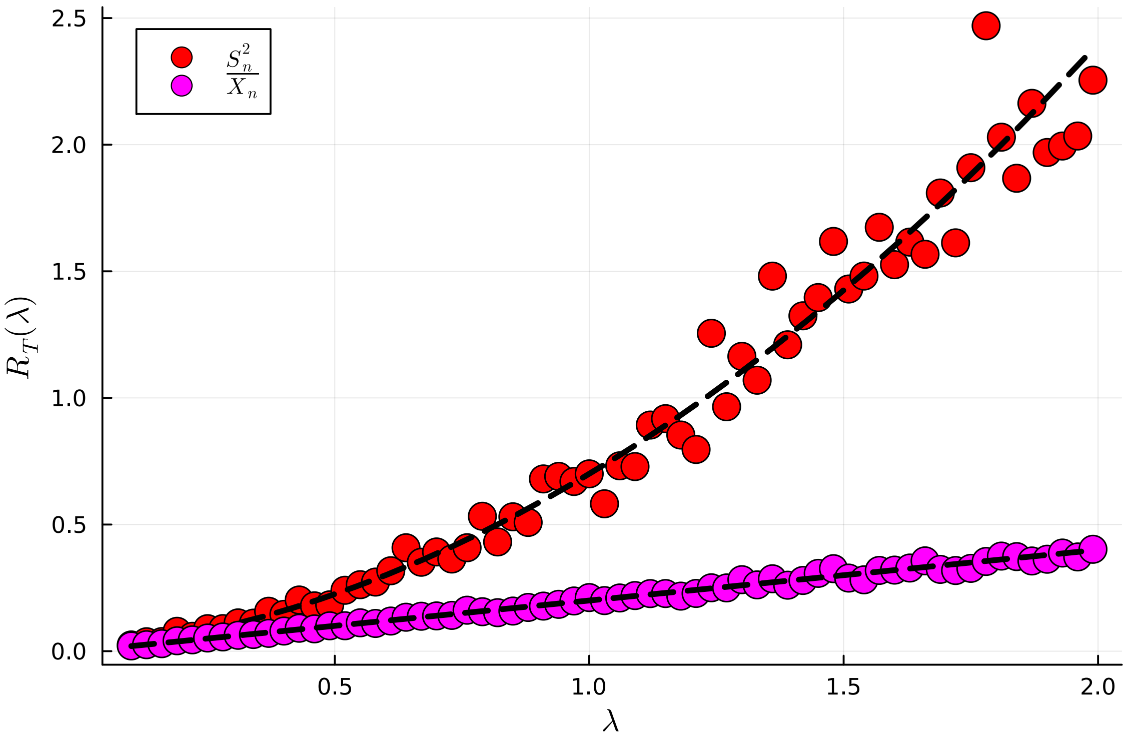

Figure 7: Approximation of the risk function for the sample mean and sample variance in estimating the parameter λ for the Poisson distribution. It is obvious from the plot that the sample mean has a lower risk as compared to the sample variance. Therefore, $\overline{X_n}$ is a better estimator as compared to S_n^2, although both are unbiased estimators of λ. The magenta lines indicate the exact risk function obtained by theoretical computation.

Which method would likely converge faster to the true value. The idea is related to consistency, which means that as $n \to \infty$, the following claims are true or not.

$$\overline{X_n} \xrightarrow{P} \lambda, \quad S_n^2 \xrightarrow{P} \lambda$$

To answer this question, let us assume that the true value is $\lambda = \lambda_0 = 2$. We simulate a sample of size n from the Poisson(λ_0) and compute $\overline{X_n}$ and S_n^2 and check how these random quantities behave as $n \to \infty$. The following code will do this task.

In [25]:

```
lambda_0 = 2
n_vals = 1:1000
sample_mean = zeros(length(n_vals))
sample_var = zeros(length(n_vals))

for n in n_vals
    x = rand(Poisson(lambda_0), n)
    sample_mean[n] = mean(x)
    sample_var[n] = var(x)
end

plot(n_vals, sample_var, color = "red", xlabel = "sample size (n)", lw = 2,
ylab = "estimator", label = L"S_n^2")
plot!(n_vals, sample_mean, color = "grey", lw = 2,label = L"\overline{X_n}")
hline!([lambda_0], color = "blue", lw = 2, linestyle = :dash, label = "")
```

Out[25]:

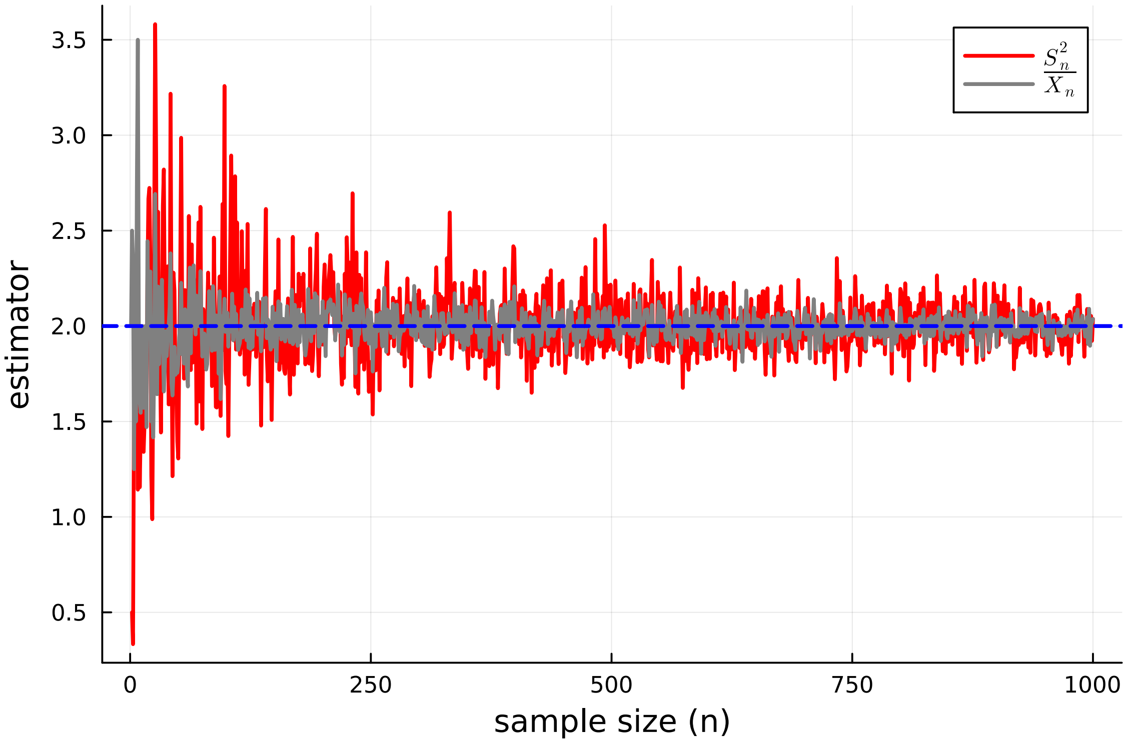

Figure 8: It can be observed that as the sample size increases, the convergence of the sample variance is much faster as compared to the sample mean to the true value of $\lambda = 2$.

Exact computing of the variance S_n^2

We recall that if we have a random sample of size n from a population with mean μ and variance σ^2. Then

$$\operatorname{Var}\left(\overline{X_n}\right) = \frac{\sigma^2}{n}.$$

In addition, $S_n^2 = \frac{1}{n-1}\sum_{i=1}^{n}(X_i - \overline{X_n})^2$ is an unbiased estimator of σ^2. To compute the variance $\operatorname{Var}(S_n^2)$, we observe that the sample variance can also be expressed as

$$S_n^2 = \frac{1}{2n(n-1)}\sum_{i=1}^{n}\sum_{j=1}^{n}(X_i - X_j)^2.$$

The following general results hold for all population distributions with finite fourth-order moment.

- $\mathbb{E}(S_n^2) = \sigma^2$.
- $\operatorname{Var}(S_n^2) = \frac{1}{n}(\mu_4 - \frac{n-3}{n-1}\mu_2^2)$, where $\mu_1 = \mathbb{E}(X_i)$ and$\mu_j = \mathbb{E}(X_i - \mu_1)^j, (j = 2, 3, 4)$.

- In addition, the covariance between $\overline{X_n}$ and S_n^2, $\text{Cov}\left(\overline{X_n}, S_n^2\right)$ can also be expressed in terms of $\mu_1, \mu_2, \mu_3, \mu_4$.

For our problem, the population follows the Poisson distribution with parameter λ. Therefore, the expectation of the sample variance is:

$$\mathbb{E}(S_n^2) = \lambda.$$

The fourth-order central moment for the Poisson distribution:

$$\mu_4 = \mathbb{E}[(X - \lambda)^4] = \sum_{i=0}^{4} \binom{4}{i} \mathbb{E}[X^{4-i}](-\lambda)^i$$

$$\mu_4 = 3\lambda^2 + \lambda.$$

The raw moments $\mathbb{E}(X^i)$, $i = 1, 2, 3, 4$, can be computed using the Moment Generating Function (MGF) by taking derivatives and evaluating at $t = 0$. The MGF of a Poisson distribution is given by:

$$M_X(t) = e^{\lambda(e^t - 1)}, \quad -\infty < t < \infty.$$

The first four derivatives evaluated at $t = 0$ are:

$$\begin{aligned} M'_X(0) &= \lambda, \\ M''_X(0) &= \lambda^2 + \lambda, \\ M'''_X(0) &= \lambda^3 + 3\lambda^2 + \lambda, \\ M''''_X(0) &= \lambda^4 + 6\lambda^3 + 7\lambda^2 + \lambda. \end{aligned}$$

This gives:

$$\mu_4 = 3\lambda^2 + \lambda.$$

Therefore, the variance of S_n^2 is given by:

$$\begin{aligned} \text{Var}(S_n^2) &= \frac{1}{n}\left(3\lambda^2 + \lambda - \frac{n-3}{n-1}\lambda^2\right) \\ &= \frac{\lambda}{n} + \frac{2\lambda^2}{n-1}. \end{aligned}$$

In the following Julia code, we add the analytically computed risk function with the risk function estimated by using simulation. The exact risk function is given by

$$\mathcal{R}(S_n^2, \lambda) = \frac{\lambda}{n} + \frac{2\lambda^2}{n-1}, \quad 0 < \lambda < \infty.$$

Making the problem more interesting

We found that both $\overline{X}_n$ and S_n^2 are unbiased estimators of λ. However, after comparing the risk functions, we found that

$$\mathcal{R}\left(\overline{X}_n, \lambda\right) \leq \mathcal{R}\left(S_n^2, \lambda\right)$$

for all $\lambda \in (0, \infty)$. We can extend this to a class of estimators defined for every constant a as follows:

$$W_a \left(\overline{X}_n, S_n^2\right) = a\overline{X}_n + (1-a)S_n^2.$$

For every a, the estimator W_a is an unbiased estimator of λ. Although $\overline{X}_n$ is better than S_n^2, is it better than W_a for all constants a? In other words, is the following statement true?

$$\mathcal{R}\left(\overline{X}_n, \lambda\right) = \mathcal{R}\left(W_1, \lambda\right) \leq \mathcal{R}\left(W_a, \lambda\right), \quad \text{for all } 0 < \lambda < \infty.$$

Before getting into some statistical theories that may guide us to determine whether $\overline{X}_n$ is indeed the best choice, you can plot the risk function of $\mathcal{R}(W_a, \lambda), 0 < \lambda < \infty$ for different choices of a using computer simulation. Note that that for $a = 1$ (corresponds to $\overline{X}_n$) and $a = 0$ (corresponds to S_n^2), risk functions are already obtained in the previous section. In addition, check whether for certain values of a, the risk function falls below $\mathcal{R}(\overline{X}_n, \lambda)$ for some λ values. In the following, let us try to compute the risk of W_a.

$$\begin{aligned}
\mathcal{R}(W_a, \lambda) &= E_\lambda (W_a - \lambda)^2 \\
&= E\Big[a\left(\overline{X}_n - \lambda\right) + (1-a)\left(S_n^2 - \lambda\right)\Big]^2 \\
&= a^2 \mathcal{R}(\overline{X}_n, \lambda) + (1-a)^2 \mathcal{R}(S_n^2, \lambda) + 2a(1-a) \times \mathrm{Cov}\left(\overline{X}_n, S_n^2\right) \\
&= a^2 \frac{\lambda^2}{n} + (1-a)^2 \left(\frac{\lambda}{n} + \frac{2\lambda^2}{n-1}\right) + 2a(1-a)\frac{\lambda}{n}.
\end{aligned}$$

To obtain the best choice of a, we need to minimize the risk function as a function of a. The equation

$$\frac{d}{da}\mathcal{R}(W_a, \lambda) = 0,$$

gives $a^* = 1$ and

$$\frac{d^2}{da^2}\mathcal{R}(W_a, \lambda)\Big|_{a=1} = \frac{2\lambda^2}{n-1} > 0.$$

Therefore, the minimum risk is obtained at $a = 1$, which corresponds to the sample mean $\overline{X}_n$ for all $\lambda \in (0, \infty)$.

Therefore, we are able to establish that if we consider the class of estimators W_a (unbiased) indexed by constant $a \in \mathbb{R}$, then $\overline{X}_n = W_1$ is the best estimator in this class when compared with respect to the risk function under the squared error loss.

However, it does not guarantee that the sample mean is the best estimator amongst all estimators of λ existing in this globe. Therefore, we need to develop theories that will be helpful to determine whether the sample mean is indeed THE BEST CHOICE to estimate the unknown parameter λ.

A natural extension is to extend the class to any unbiased estimator of λ. That is, can we claim that for any unbiased estimator T_n of λ,

$$\mathcal{R}\left(\overline{X}_n, \lambda\right) \leq \mathcal{R}(T_n, \lambda)$$

for all $\lambda \in (0, \infty)$?

Search for the holy grail (best estimator)

see next chapter (Unbaised estimation)

Conceptual Exercises

Suppose that a random sample $(X_1, X_2, \ldots, X_n)$ of size n is drawn from a population characterized by the following probability density function:

$$f(x) = \begin{cases} \frac{2x}{\theta^2}, & 0 < x < \theta \\ 0, & \text{otherwise} \end{cases}$$

where $\theta \in \Theta = (0, \infty)$ is the parameter space. We are interested in estimating the parameter θ based on the given random sample. Let $Y_n = \max(X_1, X_2, \ldots, X_n)$ be the maximum order statistic, and you decided to estimate θ using Y_n. Answer the following:
1. Plot the above PDF for different choices of θ in a single plot window. Obtain the CDF of the given PDF and plot the CDF for different choices of θ in a single plot window. Make a side-by-side plot using plot(lyoout(1,2)).
2. Obtain the exact sampling distribution of Y_n and write down both the probability density function and the cumulative distribution function of Y_n.
3. Under the squared error loss function, compute the risk function $\mathcal{R}(Y_n, \theta)$ for $0 < \theta < \infty$. Also, plot the risk function.
4. Show that as $n \to \infty$, $Y_n \to \theta$ in probability. For a fixed $\epsilon > 0$, compute the limit

$$\lim_{n\to\infty} P(|Y_n - \theta| > \epsilon)$$

and show that the limit is equal to zero as $n \to \infty$. Another possibility is to apply some inequality to bound the probability $P(|Y_n - \theta| > \epsilon)$ with a constant a_n that converges to 0 as $n \to \infty$ (Hint: Markov Inequality). Write a computer simulation to visualize that the largest order statistics Y_n indeed converges to θ as $n \to \infty$.
5. Compute the expected value of Y_n and compute the bias of the estimator Y_n, which is given by

$$\text{Bias}_\theta(Y_n) = \mathbb{E}(Y_n) - \theta, \quad \theta \in (0, \infty).$$

Also, check whether the bias $\text{Bias}_\theta(Y_n)$ tends to zero as $n \to \infty$. That means, is the estimator asymptotically unbiased? Compute the value of the constant c_n so that

$$\mathbb{E}_\theta(c_n Y_n) = \theta \quad \text{for all } \theta \in (0, \infty).$$

6. Compute the variance of Y_n, which can be computed as

$$\text{Var}_\theta(Y_n) = \mathbb{E}_\theta(Y_n^2) - (\mathbb{E}_\theta(Y_n))^2.$$

7. Show the following identity holds for all $\theta \in (0, \infty)$:

$$\mathcal{R}(Y_n, \theta) = \mathbb{E}_\theta(Y_n - \theta)^2 = \text{Var}_\theta(Y_n) + (\text{Bias}_\theta(Y_n))^2.$$

8. Using computer simulation, approximate the risk function Y_n, $\mathcal{R}(Y_n, \theta), \theta \in (0, \infty)$. The challenge, the reader may find is to simulate from the population distribution as some inbuilt function in Julia may not be available. To simulate a random number from the given

distribution (a) Compute the CDF $F_X(x)$, (b) Simulate $U \sim \mathrm{Uniform}(0,1)$. (c) Compute $X = F_X^{-1}(U)$, which is the inverse image of U under F_X. Then $X \sim f(x)$. Overlay the analytically computed risk function on the plot of the risk function obtained by computer simulation.

Another Illustrative Example

Suppose that we have a sample of size n $(X_1, X_2, \ldots, X_n)$ from a population which is characterized by the following probability density function and our goal is to estimate the parameter $\theta \in (0, \infty)$.

$$f(x) = \begin{cases} \frac{\theta}{(1+x)^{1+\theta}}, & 0 < x < \infty \\ 0, & \text{otherwise} \end{cases}$$

At the first step, we compute the cumulative distribution function and plot both PDF and CDF for different choices of θ values.

$$F_X(x) = \begin{cases} 1-(1+x)^{-\theta}, & 0 \leq x < \infty \\ 0, & \text{otherwise} \end{cases}$$

In [26]:

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

In [27]:

```julia
theta = 3

f(x) = (theta/(1+x)^(1+theta))*(x>0)

F(x) = (1- (1+x)^(-theta))*(x>0)

p1 = plot(f, -1, 4, color = "red", lw = 2, xlabel = L"x",
ylabel = L"f(x)", label = "" )

p2 = plot(F, -1, 4, color = "red", lw = 2, xlabel = L"x",
ylabel = L"F(x)", label = "" )

plot(p1, p2, layout = (1,2), size = (900, 500))
```

Out[27]:

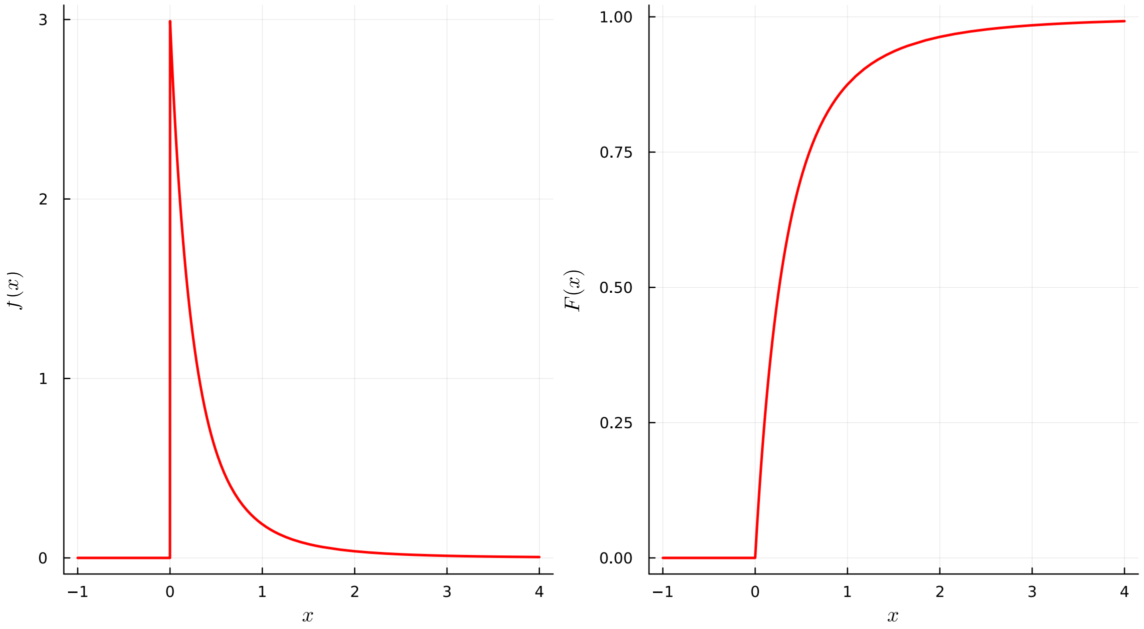

Figure 9: The PDF and CDF of the population random variable X is shown. The value of $\theta = 3$ is considered. The reader is encouraged to modify the Julia codes to draw different shapes for different choices of θ values.

.

Suppose one starts with the maximum and minimum order statistics to estimate the parameter θ, which are defined as $Y_n = \max(X_1, \ldots, X_n)$ and $Y_1 = \min(X_1, X_2, \ldots, X_n)$. First of all, we compute the sampling distribution of Y_1 and Y_n and plot the PDFs.

$$f_{Y_1}(y) = \begin{cases} n\theta(1+y)^{-(n\theta+1)}, & 0 < y < \infty, \\ 0, & \text{otherwise.} \end{cases}$$

$$f_{Y_n}(y) = \begin{cases} n\left(1-(1+y)^{-\theta}\right)^{n-1}\frac{\theta}{(1+y)^{1+\theta}}, & 0 < y < \infty, \\ 0, & \text{otherwise.} \end{cases}$$

In [28]:

```
n = 30
function f_Y1(x)
     (n*theta*(1+x)^(-n*theta-1))*(x>0)
end

p1 = plot(f_Y1, -1, 4, color = "red", xlabel = L"y", lw = 2,
ylabel = L"f_{Y1}(y)", title = "n = $n", label = "")

function f_Yn(x)
     n*((1-(1+x)^(-theta))^(n-1))*theta/(1+x)^(1+theta)*(x>0)
end

p2 = plot(f_Yn, -1, 10, color = "red", xlabel = L"y", lw = 2,
ylabel = L"f_{Yn}(y)", title = "n = $n", label = "")

plot(p1, p2, layout = (1,2), size = (900, 500))
```

Out[28]:

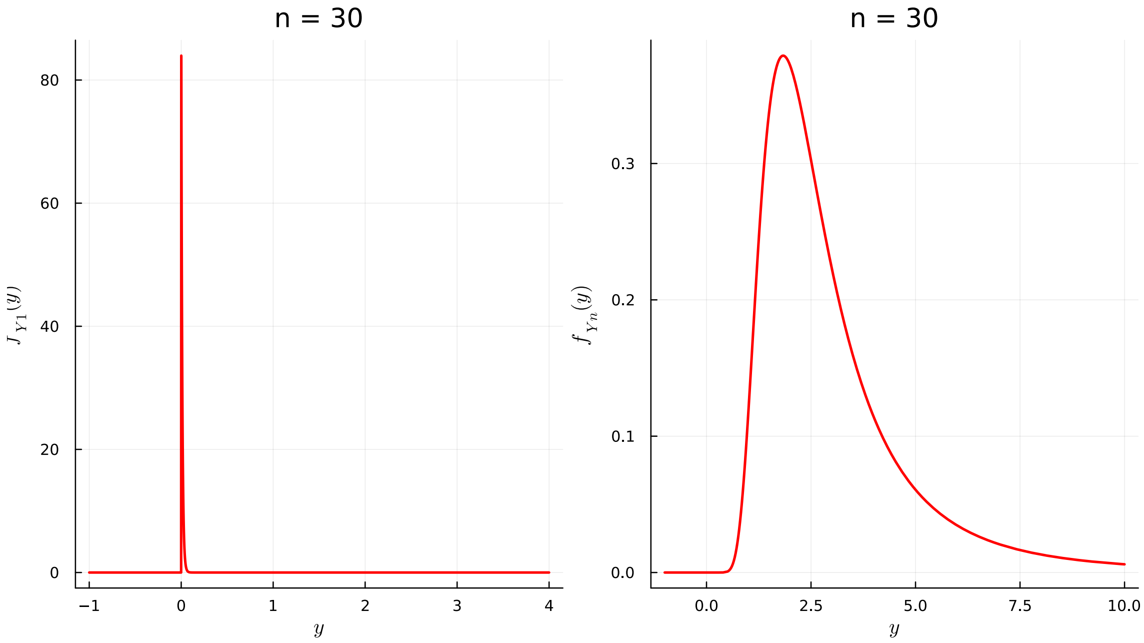

Figure 10: The sampling distribution of the Y_1 and Y_n. The value of θ is set to 3. It is clear that Y_1 is highly concentrated at 0 and the concentration increases as n becomes large. Therefore, Y_1 does not appear to be a good choice for estimating θ. The sample size is fixed at $n = 30$.

One may probably think of using the maximum order statistics $Y_n = \max(X_1, X_2, \ldots, X_n)$ to estimate the parameter θ. Let us investigate the properties of the estimator Y_n. At the first step, we can compute the mean and higher-order moments of Y_n. The mean of Y_n can be computed by computing the expectation of $Y_n + 1$ as follows:

$$\begin{aligned} E_\theta(Y_n+1) &= \int_0^\infty (1+y)n\left[1-(1+y)^{-\theta}\right]^{n-1}\cdot\frac{\theta}{(1+y)^{1+\theta}}dy \\ &= nB\left(1-\frac{1}{\theta}, n\right), \quad \theta\in(1,\infty). \end{aligned}$$

The expectation exists only when $\theta \in (1, \infty)$. A similar computation will give us:

$$E_\theta(Y_n+1)^2 = nB\left(1-\frac{2}{\theta}, n\right), \quad \theta\in(2,\infty).$$

The above result can be generalized as...

$$\begin{aligned} \mathbb{E}_\theta(Y_n+1)^m &= \int_0^\infty (1+y)^m\cdot n\left[1-(1+y)^{-\theta}\right]^{n-1}\cdot\frac{\theta}{(1+y)^{1+\theta}}dy \\ &= n\mathcal{B}\left(1-\frac{m}{\theta}, n\right), \quad \theta\in(m,\infty). \end{aligned}$$

Then mth order raw moments exist when $\theta \in (m, \infty)$. Use the idea of change of variable as $z = (1+y)^{-\theta}$ to compute the above integrals. Therefore, we can evaluate the risk of Y_n under the squared error loss as

$$\begin{aligned} \mathcal{R}(Y_n,\theta) &= \mathbb{E}_\theta(Y_n-\theta)^2 = \mathbb{E}_\theta[(Y_n+1)-(\theta+1)]^2 \\ &= n\mathcal{B}\left(1-\frac{2}{\theta}, n\right) - 2n(\theta+1)\mathcal{B}\left(1-\frac{1}{\theta}, n\right) + (1+\theta)^2. \end{aligned}$$

We need to keep in mind that the risk function exists only when $\theta \in (2, \infty)$. Let us plot the risk function for different choices of θ. The following figure suggests that the performance of this estimator is reasonable at a small subset of the parameter space.

In [29]:

```julia
using SpecialFunctions
```

In [30]:

```julia
function risk_fun_Yn(theta)
     n*beta(1-2/theta,n)-2*n*(theta+1)*beta(1-1/theta,n) +(1+theta)^2
end

plot(risk_fun_Yn, 2.3, 10, color = "red", lw = 2, xlabel = L"\theta",
ylabel = L"R(Y_n, \theta)", label = "")
```

Out[30]:

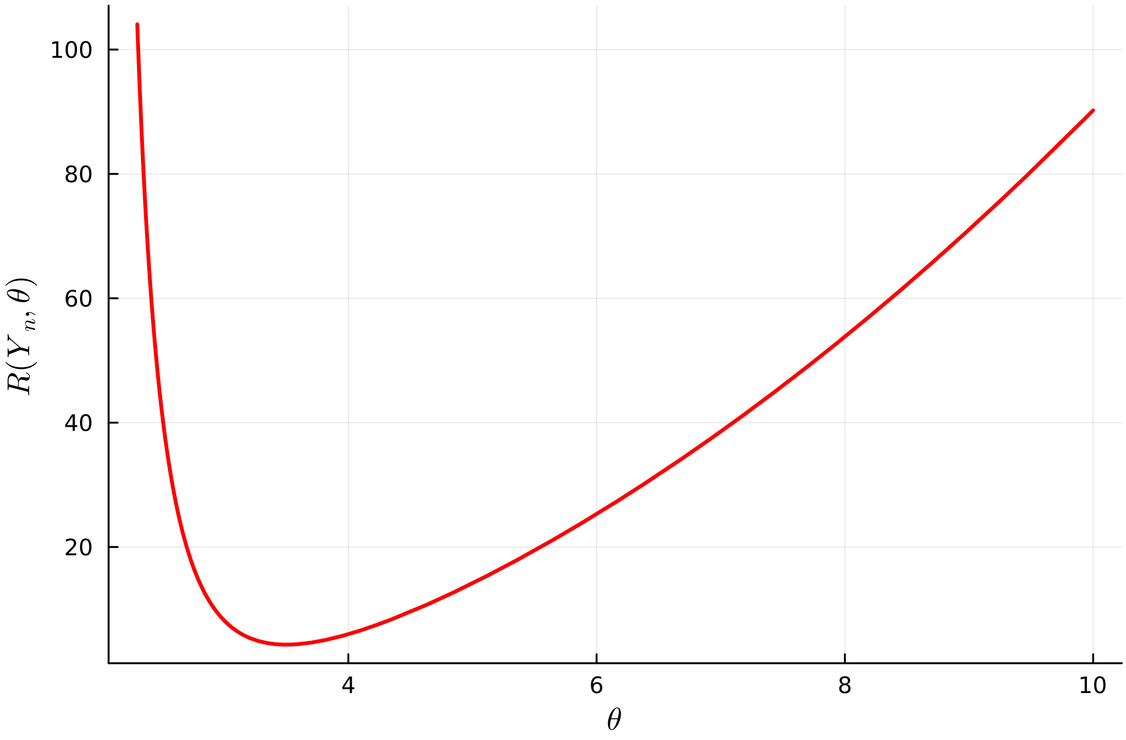

Figure 12: The risk function of Y_n as a function of θ. The sample size is fixed at $n = 10$

Let us check how the shapes changes with respect to sample size n

In [31]:

```julia
n_vals = [10,20,50,100, 500]

n = 5
plot(risk_fun_Yn, 2.3, 10, color = "red", lw = 2, xlabel = L"\theta",
ylabel = L"R(Y_n, \theta)", ylims = (0, 80),label = "")

for i in 1:length(n_vals)
    n = n_vals[i]
    plot!(risk_fun_Yn, 2.3, 10, color = i, linestyle = :dash,
    label = "n = $n")
end

display(plot!())
```

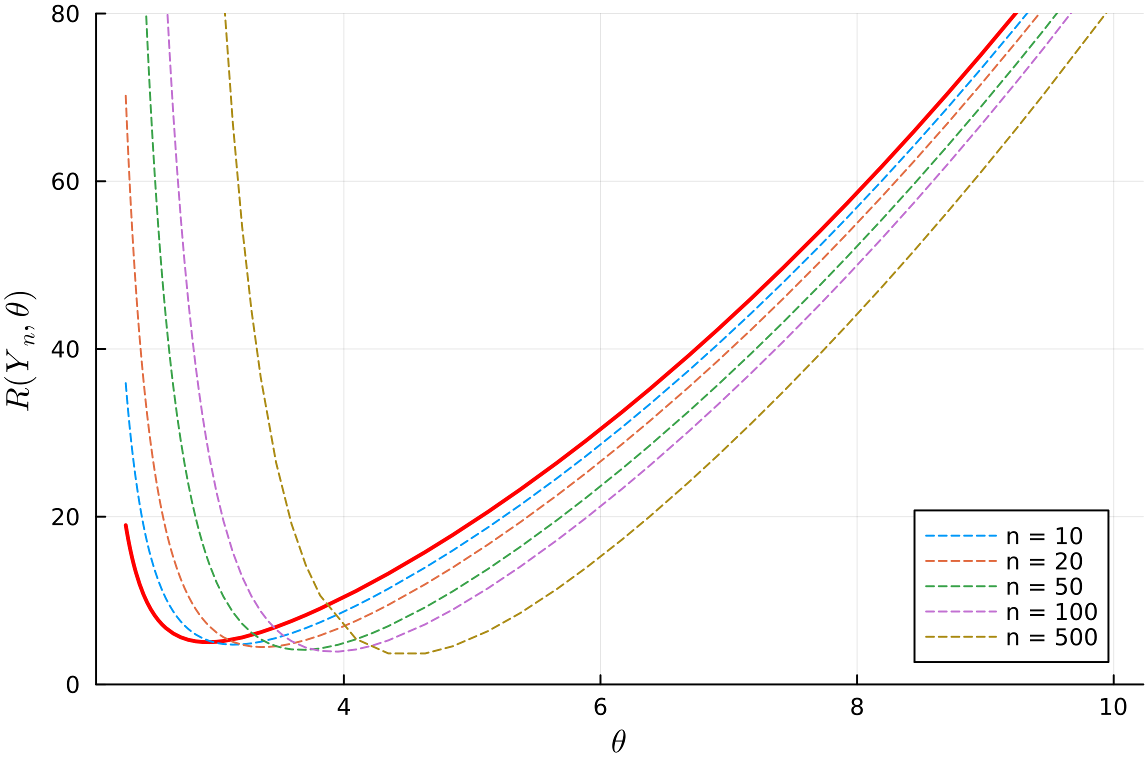

Figure 13: The risk function of the maximum order statistics for different choices of the n values

Method of Moment Estimator

Using the method of moments, we can obtain an estimate of θ. The population mean $\mathbb{E}(X) = \frac{1}{\theta - 1}$, which exists only when $\theta \in (1, \infty)$. Therefore, equating the sample mean $\overline{X}_n$ with the population mean, we obtain the method of moment estimator of θ as

$$\hat{\theta}_{\text{MOM}} = \frac{1}{\overline{X}_n + 1} = V_n \text{ (say)}$$

Method of Maximum Likelihood

Using the method of maximum likelihood, we can also obtain the estimator of θ as follows: The likelihood function is given by

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \frac{\theta}{(1 + x_i)^{1+\theta}}, \quad 0 < \theta < \infty$$

and the log-likelihood function is given by

$$l(\theta) = n \log \theta - n \sum_{i=1}^{n} \log(1 + x_i).$$

Equating $l'(\theta) = 0$ implies,

$$\theta^* = \frac{n}{\sum_{i=1}^{n} \log(1 + x_i)}$$

and

$$l''(\theta) = -\frac{n}{\theta^2} < 0, \quad \text{for all } \theta \in (0, \infty).$$

Therefore, $l(\theta)$ attains its maximum at θ^*. Thus, the Maximum Likelihood Estimator of θ is given by

$$\hat{\theta}_{\text{MLE}} = \frac{n}{\sum_{i=1}^{n} \log(1 + X_i)} = W_n \text{ (say)}.$$

As per our notation, V_n and W_n are the method of moments and maximum likelihood estimators of θ, respectively. Now we have three estimators of θ and we are interested in comparing their risk functions which are listed as $\mathcal{R}(Y_n, \theta)$, $\mathcal{R}(V_n, \theta)$, and $\mathcal{R}(W_n, \theta)$. It is important to note that the risk functions may exist only on a subset of the parameter space $\Theta = (0, \infty)$.

We first try to compute the sampling distribution of W_n. We compute it step by step.

- Step – I

Consider $Y = \log(1 + X)$, then the CDF of Y is given by

$$\begin{aligned} F_Y(y) &= P(Y \leq y) = P(\log(1 + X) \leq y) \\ &= P(X \leq e^y - 1) \\ &= 1 - e^{-\theta y}, \quad 0 < y < \infty. \end{aligned}$$

Therefore, $Y = \log(1 + X) \sim$ Exponential(rate = θ), which is $\mathcal{G}(\alpha = 1, \beta = \frac{1}{\theta})$ distribution.

- Step – II

Let $Y_i = \log(1 + X_i)$, $1 \leq i \leq n$, then by using the Moment Generating Function technique, we can easily show that

$$\begin{aligned} Z_n = \sum_{i=1}^{n} Y_i &= \sum_{i=1}^{n} \log(1 + X_i) \\ &\sim \mathcal{G}(\alpha = n, \beta = \frac{1}{\theta}), \end{aligned}$$

whose PDF is given by

$$f_{Z_n}(z) = \frac{\theta^n z^{n-1} e^{-\theta z}}{\Gamma(n)}, \quad 0 < z < \infty.$$

and zero otherwise.

- Step – III

The MLE $W_n = \frac{n}{Z_n}$. Using the transformation formula, we obtain $f_{W_n}(w)$ as

$$f_{W_n}(w) = f_{Z_n}\left(\frac{n}{w}\right) \left| \frac{d}{dw}\left(\frac{n}{w}\right) \right|, \quad 0 < w < \infty,$$

which gives the PDF of W_n as

$$f_{W_n}(w) = \frac{n(n\theta)^n e^{-n\theta/w}}{w^{n+2}\Gamma(n)}, \quad 0 < w < \infty,$$

and zero otherwise.

In the following Julia code, we visualize the sampling distribution of the MLE of θ (PDF) for different choices of θ and sample size n.

In [32]:

```julia
# here we take the log of the function to define the function
function f_Wn(w, n, theta)
    if w > 0
        log_f = log(n) + n*log(n*theta) - n*theta/w -
        (n+2)*log(w) - loggamma(n)
        return exp(log_f)
    else
        return 0
    end
end

theta = 3
n = 20
p1 = plot(w -> f_Wn(w, n, theta), -0.5, 7, color="red", lw=2, xlabel=L"w",
     ylabel=L"F_{W_n}(w)", title=L"\theta = %$theta", label="n = $n")
n = 10
plot!(w -> f_Wn(w,n, theta), -0.5, 7, color="blue", lw=2,
    label = "n = $n")
n = 5
plot!(w -> f_Wn(w,n, theta), -0.5, 7, color="magenta", lw=2,
    label = "n = $n")
scatter!([theta],[0], color = "green", markersize = 7 ,label = "")

n = 10
theta = 3
p2 = plot(w -> f_Wn(w, n, theta), -0.5, 7, color="red", lw=2, xlabel=L"w",
     ylabel=L"F_{W_n}(w)", title="n=$n", label=L"\theta = %$theta")
theta = 5
plot!(w -> f_Wn(w,n, theta), -0.5, 7, color="magenta", lw=2,
    label = L"\theta = %$theta")

plot(p1, p2, layout = (1,2), size = (900, 400))
```

Out[32]:

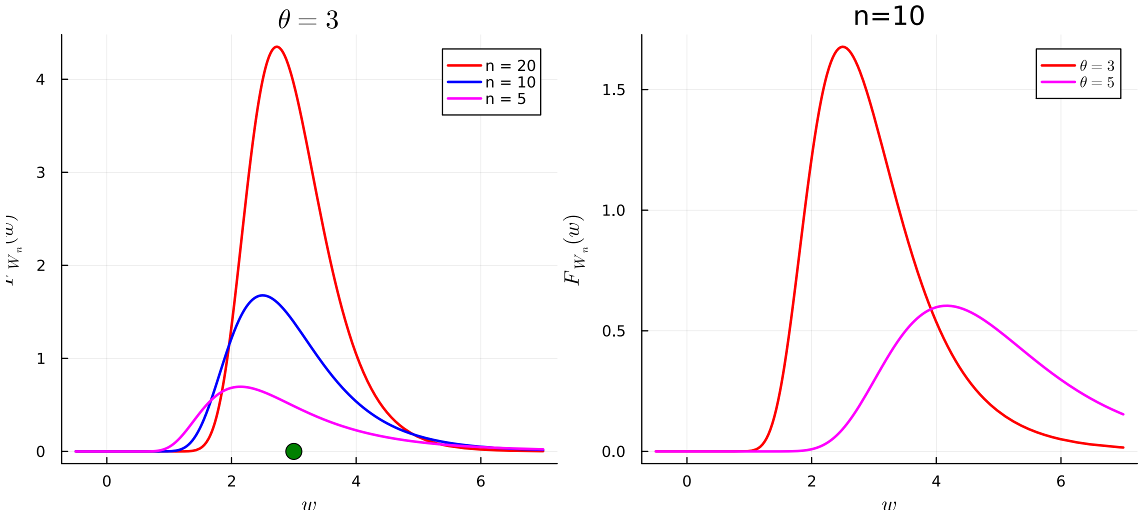

Figure 14:The sampling distribution of the MLE W_n of θ for different choices of θ and different sample size. Left panel: As the sample size increases, the sampling distribution of W_n is highly concentrated about the true

value of $\theta = 3$, which is a desirable property of the MLE (consistency).

- Step - IV

To compute the risk $\mathcal{R}(W_n, \theta)$, we compute

$$\mathbb{E}(W_n) = \frac{n\theta}{n-1}, \quad n > 1,$$

and

$$\mathbb{E}(W_n^2) = \frac{n^2\theta^2}{(n-1)(n-2)}, \quad n > 2.$$

Therefore, the risk under the squared error loss is given by

$$\begin{aligned}\mathbb{E}_\theta(W_n - \theta)^2 &= \mathbb{E}_\theta(W_n^2) - 2\theta\mathbb{E}_\theta(W_n) + \theta^2 \\ &= \frac{\theta^2(n+2)}{(n-1)(n-2)}, \quad n > 2.\end{aligned}$$

The following observation is important.

- $Bias_\theta(W_n)$ is given by

$$\begin{aligned}\text{Bias}_\theta(W_n) &= \mathbb{E}_\theta(W_n) - \theta \\ &= \frac{n\theta}{n-1} - \theta \\ &= \frac{\theta}{n-1} \to 0 \quad \text{as} \quad n \to \infty.\end{aligned}$$

Therefore, W_n is an asymptotically unbiased estimator.

$\mathcal{R}(W_n, \theta) = \mathbb{E}_\theta(W_n - \theta)^2 \to 0$ as $n \to \infty$. Therefore, $W_n \to \theta$ in probability as $n \to \infty$ (a simple application of Markov Inequality).
Let us now verify whether the theoretical computation of the risk function for the W_n is supported by the simulation study as well. We follow the same scheme as previously.

- Fix θ and fix n, sample size.
- Simulate $X_1, X_2, \ldots, X_n \sim f(x|\theta)$.
- Compute W_n.
- Compute the loss $(W_n - \theta)^2$.
- Repeat the above three steps M times to compute the average loss.
- Repeat the above five steps for different choices of θ from the parameter space.
- Plot the average loss values (approximate risk) at each value of θ.

In the above algorithm, the primary challenge is to simulate from the given PDF as some inbuilt function may not be available to directly simulate using R or Python. Let us use the probability integral transform to simulate $X_1, \ldots, X_n \sim f(x|\theta)$. Simulate $U \sim \text{Uniform}(0,1)$ and compute $X = F_X^{-1}(U)$, where F_X^{-1} is the inverse of the CDF. In this case, the equation to generate $X \sim f(x|\theta)$ becomes

$$X = (1-U)^{-\frac{1}{\theta}} - 1, \quad 0 < U < 1.$$

In the following, we verify this via a simulation study.

In [33]:

```
# define the pdf of function
f(x, theta) = (theta > 0 && x ≥ 0) * theta * (1 + x)^(-theta - 1)
n = 1000
theta = 3
u = rand(Uniform(0,1), n)
x = (1 .- u) .^ (-1 / theta) .- 1
histogram(x, normalize = true, xlabel = "x", ylabel = "density",
    label = "")
plot!(x->f(x,theta), color = "red", label = "")
```

Out[33]:

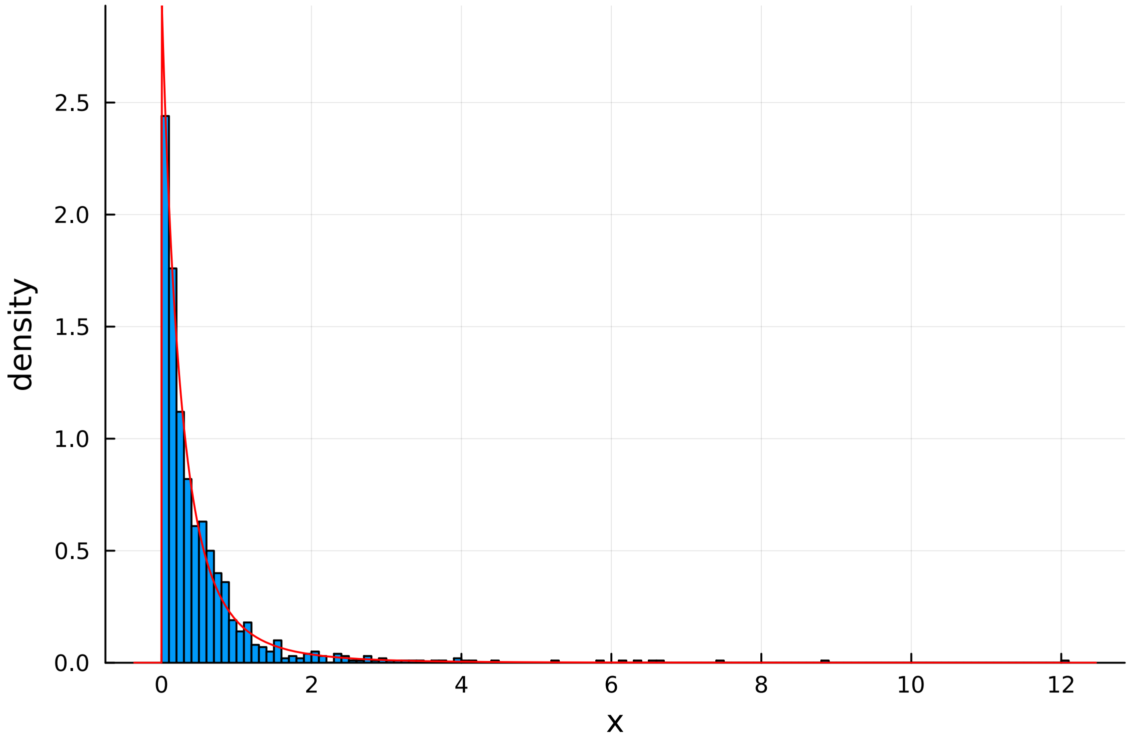

Figure 15: Simulated realization from the given PDF $f(x|\theta = 3)$ using probability integral transform.

Let us now execute simulation of the risk function using computer simulation and verify with the theoretical result.

In [34]:

```
M = 1000 # number of replications
n = 10
theta_vals = 0.1:0.1:4
risk_vals = zeros(length(theta_vals))

for i in 1:length(risk_vals)
    theta = theta_vals[i]
    loss_vals = zeros(M)
    for j in 1:M
        u = rand(Uniform(0,1), n)
        x = (1 .- u) .^ (-1 / theta) .- 1
        W_n = n/sum(log.(1 .+ x))
        loss_vals[j] = (W_n .- theta).^2
    end
    risk_vals[i] = mean(loss_vals)
end

scatter(theta_vals, risk_vals, color = "red", markersize = 6,
xlabel = L"\theta", ylabel = L"R(W_n, \theta)", label = "" )
```

```
plot!(x->x^2*(n+2)/((n-1)*(n-2)), lw = 2, color = "blue", linestyle = :dash,
label = "")
```

Out[34]:

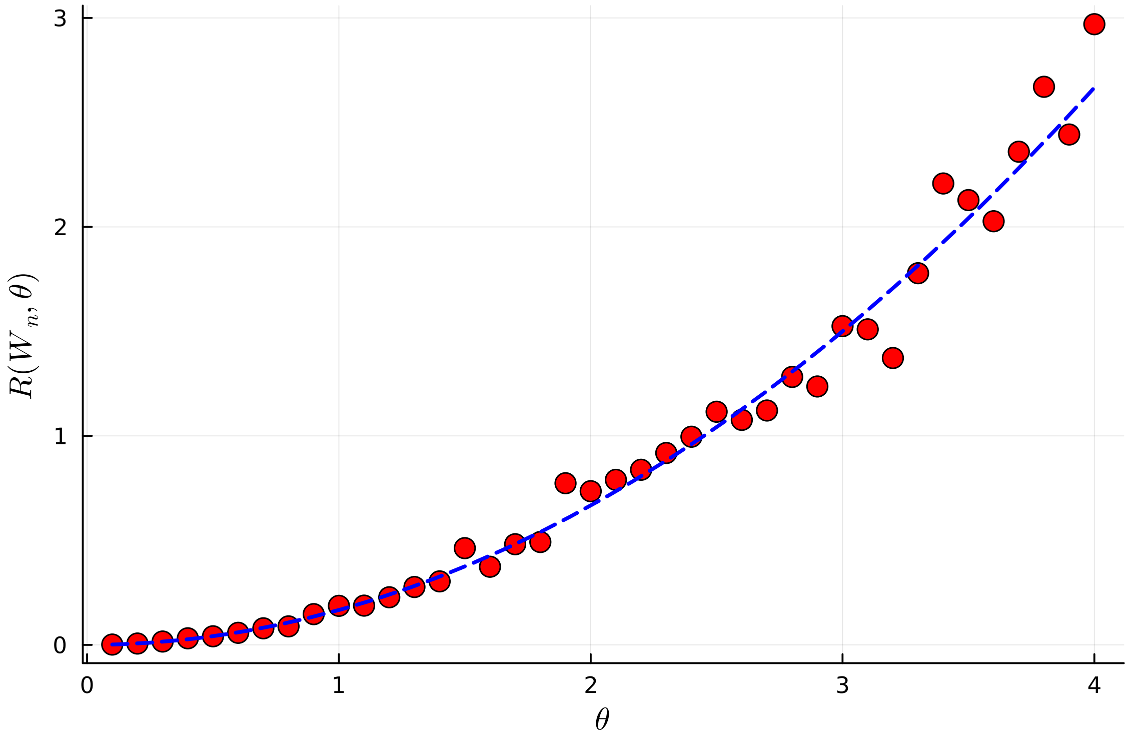

Figure 16: The risk function is obtained by computer simulation for the MLE W_n of θ. True risk function is added for reference (blue dotted line).

The MLE W_n has desirable properties. We are yet to check it for the Method of Moment estimator V_n.

Approximation of $\mathcal{R}(V_n, \theta)$

The exact sampling distribution of the method of moment estimator $V_n = \overline{X_n}^{-1} + 1$ is difficult to obtain. Therefore, an analytically tractable expression of the risk function is almost impossible to obtain. However, we can possibly obtain an approximation of the risk function for large sample size n for a subset of the parameter space $\Theta = (0, \infty)$. In the following, we first obtain the risk function by computer simulation.

In [35]:

```
M = 1000 # no of replications
n = 10
theta_vals = 0.1:0.1:10
risk_vals = zeros(length(theta_vals))

for i in 1:length(theta_vals)
    theta = theta_vals[i]
    loss_vals = zeros(M)
    for j in 1:M
        u = rand(Uniform(0,1), n)
        x = (1 .- u) .^ (-1 / theta) .- 1
        V_n = 1/mean(x) + 1
        loss_vals[j] = (V_n .- theta).^2
    end
    risk_vals[i] = mean(loss_vals)
end
```

```
scatter(theta_vals, risk_vals, color = "red", markersize = 6,
xlabel = L"\theta", ylabel = L"R(V_n, \theta)", label = "" )
```

Out[35]:

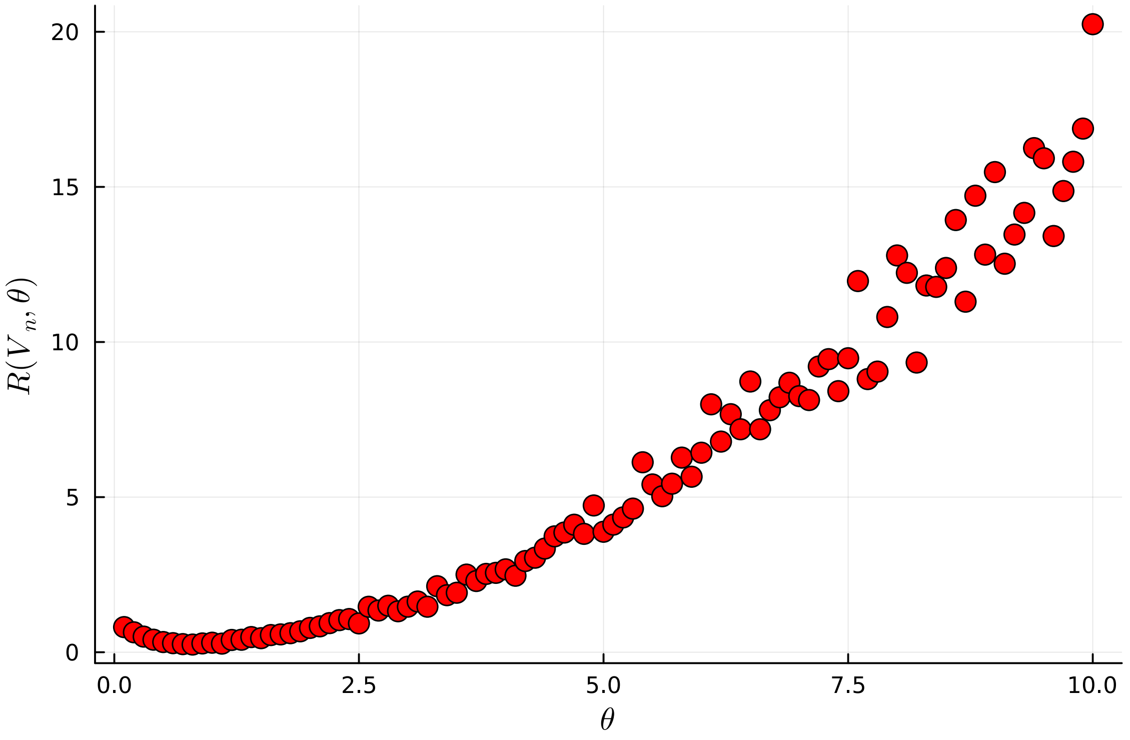

Figure 17: The risk function is obtained by computer simulation for the MLE W_n of θ.

$$E(X) = \mu = \frac{1}{\theta - 1} \text{ exists for } \theta > 1 \text{ and } Var(X) = \sigma^2 = \frac{2}{(\theta-1)(\theta-2)} \text{ which exists fo}$$

Therefore, when $\theta \in (2, \infty)$, we can apply the CLT, which ensures the large sample approximation of the sampling distribution of $\overline{X}_n$ as

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

Now we can approximate the sampling distribution of $g(\overline{X}_n) = \overline{X}_n^{-1} + 1$ by using first order Taylor's approximation and $g(\mu) = \frac{1}{\mu} + 1 = \theta$.

$$g\left(\overline{X}_n\right) \approx g(\mu) + (\overline{X}_n - \mu)g'(\mu)$$

which implies

$$V_n = \frac{1}{\overline{X}_n} + 1 \approx \frac{1}{\mu} + 1 + (\overline{X}_n - \mu)\left(-\frac{1}{\mu^2}\right)(-(\theta-1)^2)$$

Therefore, by the first-order Taylor Approximation,

$$E_\theta(V_n) \approx \theta$$

and the variance is obtained as

$$\begin{aligned} Var_\theta(V_n) &\approx E_\theta(V_n-\theta)^2 \\ &= E_\theta\left(\overline{X}_n - \frac{1}{\theta-1}\right)^2(\theta-1)^4 \\ &= Var_\theta(\overline{X}_n)(\theta-1)^2 \\ &\approx \frac{1}{n(\theta-1)(\theta-2)}(\theta-1)^4 \\ &= \frac{(\theta-1)^3}{n(\theta-2)}. \end{aligned}$$

Therefore, by the application of the Delta method,

$$V_n = g\left(\overline{X}_n\right) \sim \mathcal{N}\left(\theta, \frac{(\theta-1)^3}{n(\theta-2)}\right), \quad \text{for large } n, \theta \in (2, \infty)$$

In the following code, the above approximation is verified by computer simulation.

In [36]:

```
M = 1000
theta = 4  # true value of theta
n_vals = [5, 10, 30, 50, 100, 500]
plt = plot(layout=(2, 3), size=(800, 600))

for (idx, n) in enumerate(n_vals)
    xbar = zeros(M)
    for i in 1:M
        u = rand(Uniform(0, 1), n)
        x = (1 .- u).^(-1/theta) .- 1
        xbar[i] = mean(x)
    end
    histogram!(xbar, normalize=true, bins=:auto, label="",
        xlims = extrema(xbar),title="n = $n",
        xlabel=L"\bar{X_n}", subplot = idx)
    plot!(x -> pdf(Normal(1/(theta-1),
                sqrt(1/(n*(theta-1)*(theta-2)))), x),
        lw=2, color="red", label = "", subplot = idx)
end

display(plt)
```

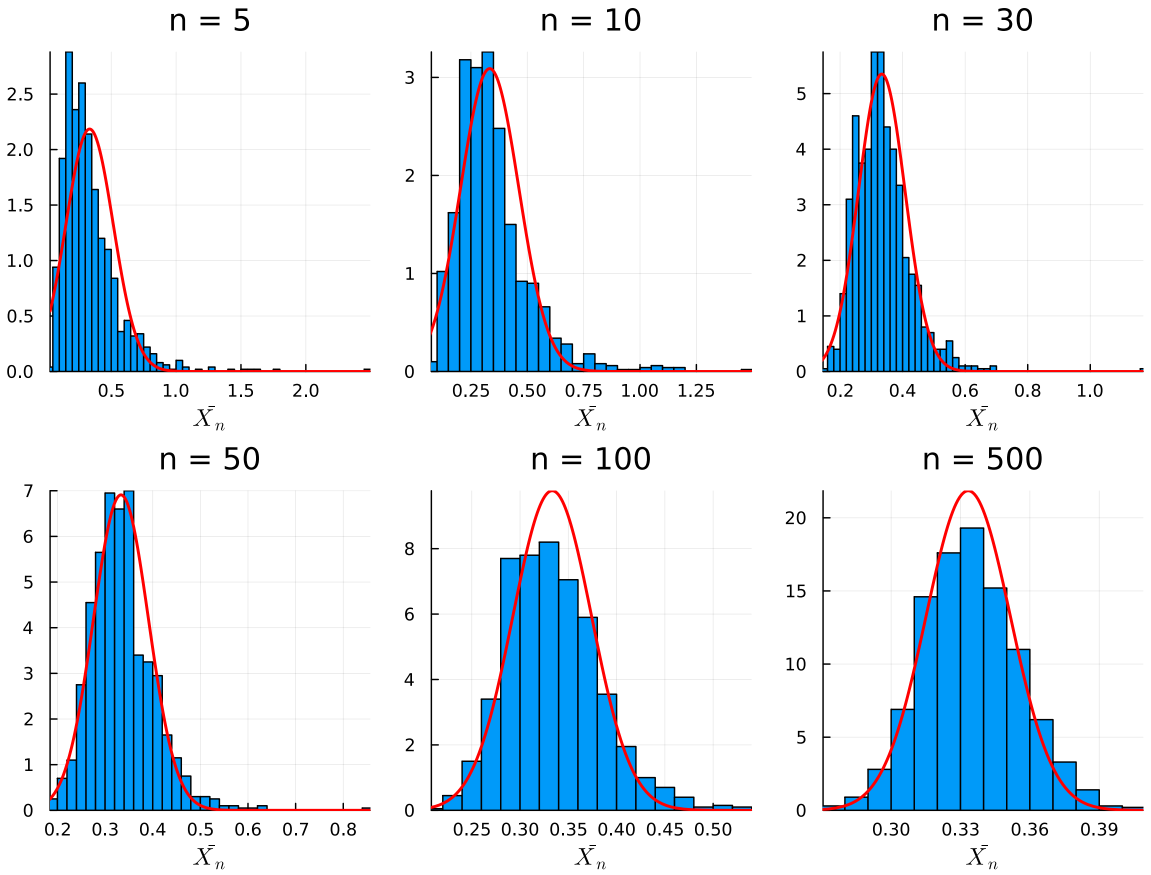

Figure 18: As the sample size increases, the sample mean $\overline{X}_n$ is approximately normally distributed with mean $\mu = \frac{1}{\theta-1}$ and variance $\frac{\sigma^2}{n} = \frac{1}{n(\theta-1)(\theta-2)}$. Using the first-order Taylor's approximation, the sampling distribution of V_n is obtained by computer simulation.

In [37]:

```
M = 1000
theta = 4  # true value of theta
n_vals = [5, 10, 30, 50, 100, 500]
plt = plot(layout=(2, 3), size=(800, 600))

for (idx, n) in enumerate(n_vals)
    g_xbar = zeros(M)
    for i in 1:M
        u = rand(Uniform(0, 1), n)
        x = (1 .- u).^(-1/theta) .- 1
        g_xbar[i] = 1 / mean(x) + 1
    end
    histogram!(g_xbar, normalize=true, bins=:auto, label="",
        title="n = $n", xlims = extrema(g_xbar),
        xlabel=L"V_n", subplot = idx )
    plot!(x -> pdf(Normal(theta, sqrt((theta-1)^3 / (n*(theta-2)))), x),
        lw=2, color="red", label = "", subplot = idx)
end

display(plt)
```

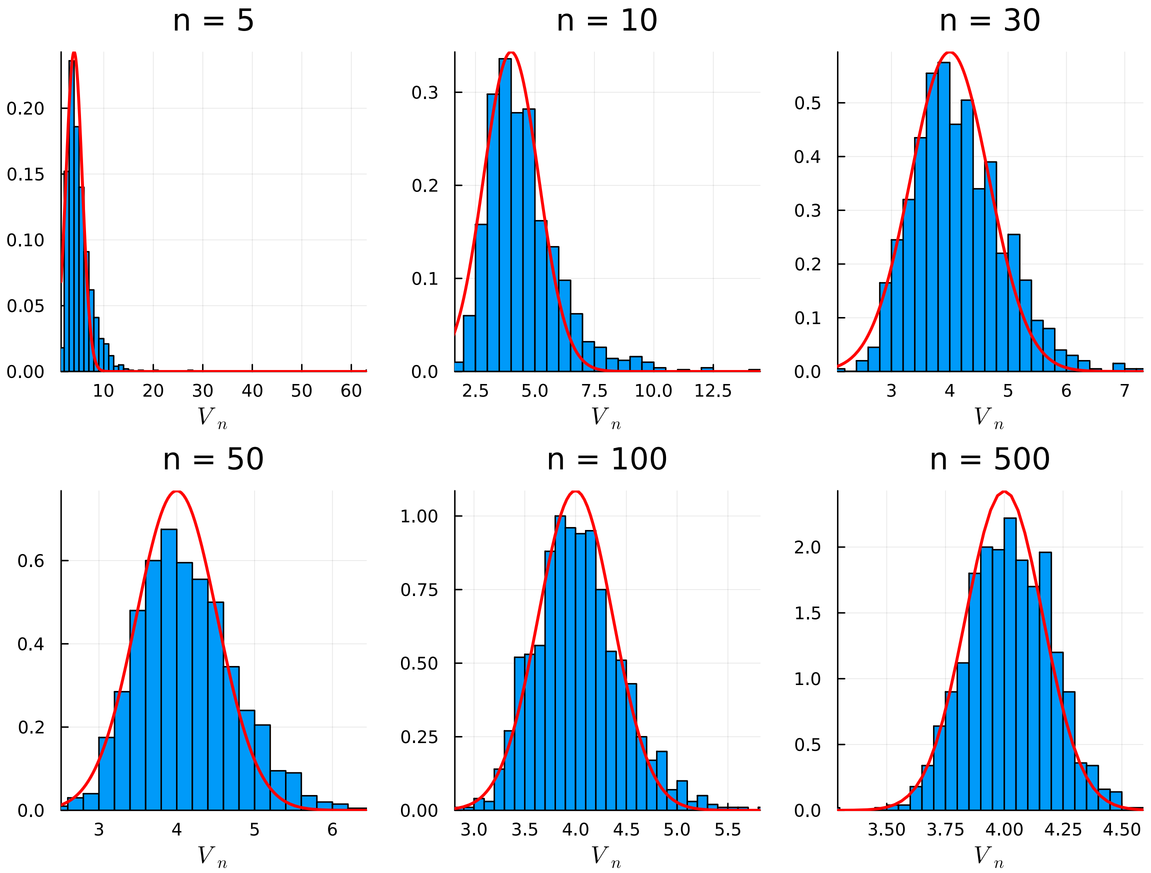

Figure 19: As the sample size increases, the sample mean $\overline{X}_n$ is approximately normally distributed with mean $\mu = \frac{1}{\theta-1}$ and variance $\frac{\sigma^2}{n} = \frac{1}{n(\theta-1)(\theta-2)}$. Using the first-order Taylor's approximation, the sampling distribution of V_n is obtained by computer simulation.

In [38]:

```julia
using Distributions, Plots

n_vals = [5, 10, 30, 50, 100, 500]
theta_vals = 3:0.5:20
plt = plot(layout=(2, 3), size=(800, 600))

M = 500 # number of replications

for (idx, n) in enumerate(n_vals)
    risk_vals = zeros(length(theta_vals))

    for i in eachindex(theta_vals)
        theta = theta_vals[i]
        loss_vals = zeros(M)

        for j in 1:M
            u = rand(Uniform(0,1), n)
            x = (1 .- u) .^ (-1 / theta) .- 1
            V_n = 1 / mean(x) + 1
            loss_vals[j] = (V_n - theta)^2
        end

        risk_vals[i] = mean(loss_vals)
    end

    scatter!(theta_vals, risk_vals, color = "red", markersize = 6,
        xlabel = L"\theta", ylabel = L"R(V_n, \theta)", label = "",
        title = "n = $n", subplot = idx)
    plot!(x -> (x - 1)^3 / (n * (x - 2)), lw = 2, color = "blue", linestyle = :das
        label = "", subplot = idx)
end

display(plt)
```

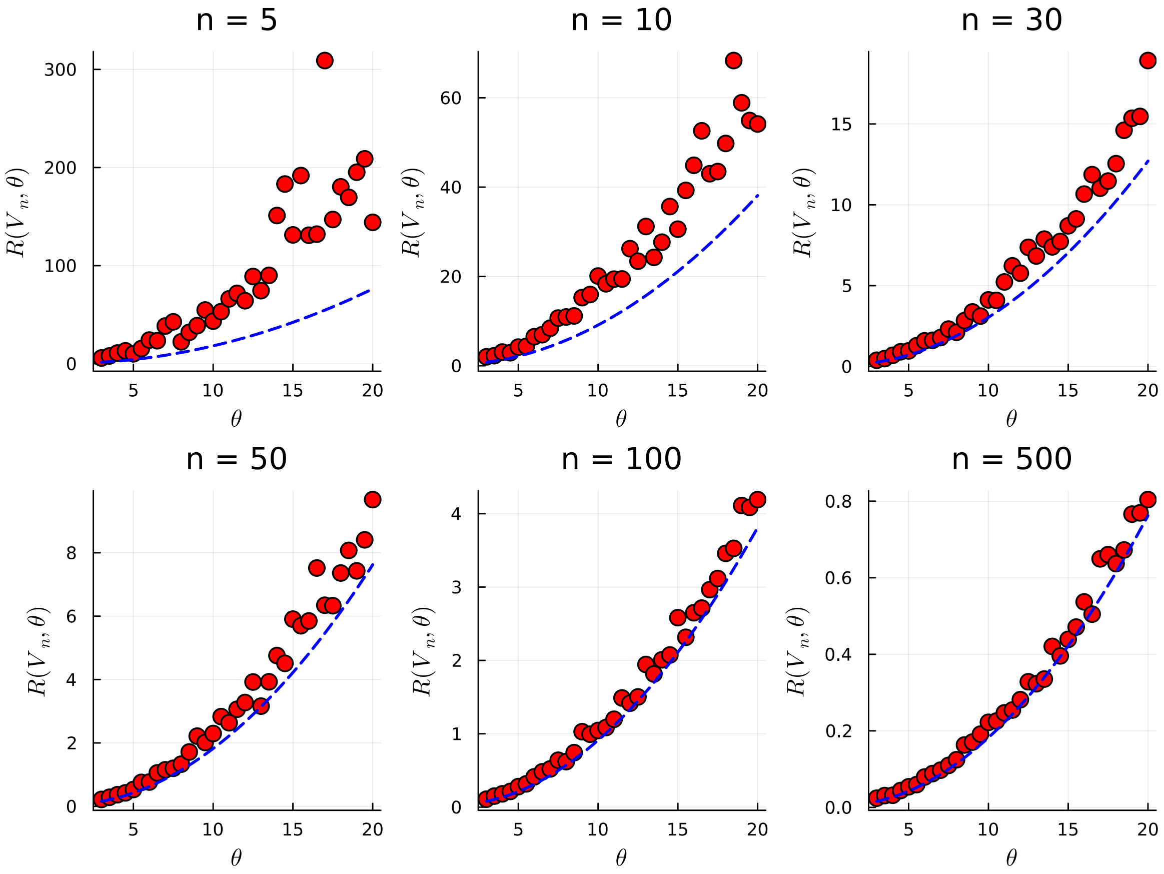

Figure 20: The approximation of the risk function of V_n by the first order Taylor's approximation. As $n \to \infty$, the approximations are accurate.

Comparing all estimators Y_n, V_n, and W_n

In [39]:

```
n = 50
theta_vals = 0.1:0.05:5
M = 1000
risk_Vn = zeros(length(theta_vals)) # method of moments
risk_Wn = zeros(length(theta_vals)) # method of MLE

for i in 1:length(theta_vals)
    theta = theta_vals[i]
    loss_Vn = zeros(M)
    loss_Wn = zeros(M)
    for j in 1:M
        u = rand(Uniform(0,1), n)
        x = (1 .- u) .^ (-1 / theta) .- 1
        V_n = 1 / mean(x) + 1
        W_n =  n/sum(log.(1 .+ x))
        loss_Vn[j] = (V_n - theta)^2
        loss_Wn[j] = (W_n - theta)^2
    end
    risk_Vn[i] = mean(loss_Vn)
    risk_Wn[i] = mean(loss_Wn)
end
```

```
plot(theta_vals, risk_Wn, color = "red", lw = 2, xlabel = L"\theta",
ylabel = L"R(. ,\theta)", title = "n = $n", label = L"W_n")
plot!(theta_vals, risk_Vn, color = "blue", lw = 2, label = L"V_n")
```

Out[39]:

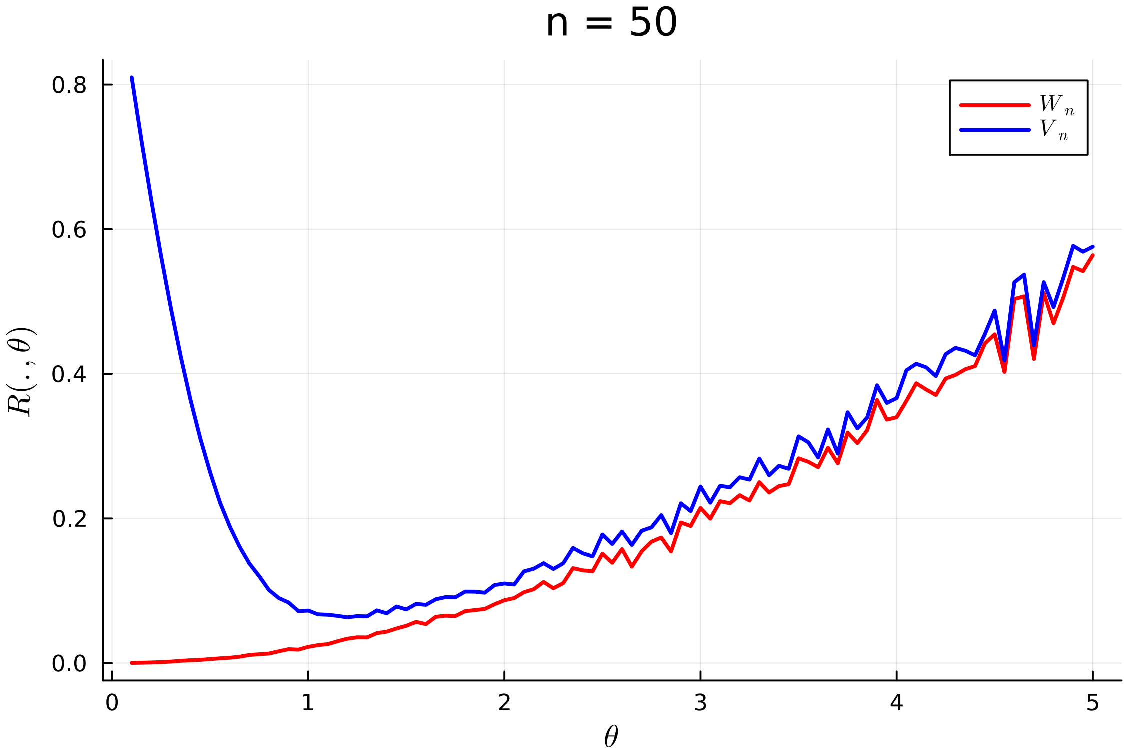

Figure 21: Comparison of the risk function of the method of moments and method of maximum likelihood. It can be observed that the MLE has uniformly smaller risk than the MoM estimator. Therefore, W_n is a preferred estimator than V_n. Similar exercise can be carried out for Y_n as well, and we can find that MLE has outperformed both the estimator. A natural question arises, is MLE the best among all estimators of θ? This will be answered in the next chapter.