正規分布モデルの共役事前分布によるベイズ統計

- 黒木玄
- 2022-09-03～2022-09-04

目次

In [1]:

```
using Distributions
using LinearAlgebra
using StatsPlots
default(fmt=:png, size=(500, 350),
    titlefontsize=10, tickfontsize=6, guidefontsize=9,
    plot_titlefontsize=10)
using SymPy
using Turing
```

In [2]:

```
# Override the Base.show definition of SymPy.jl:
# https://github.com/JuliaPy/SymPy.jl/blob/29c5bfd1d10ac53014fa7fef468bc8deccadc2fc/src/types.

@eval SymPy function Base.show(io::IO, ::MIME"text/latex", x::SymbolicObject)
    print(io, as_markdown("\\displaystyle " *
            sympy.latex(x, mode="plain", fold_short_frac=false)))
end
@eval SymPy function Base.show(io::IO, ::MIME"text/latex", x::AbstractArray{Sym})
    function toeqnarray(x::Vector{Sym})
        a = join(["\\displaystyle " *
                sympy.latex(x[i]) for i in 1:length(x)], "\\\\")
        """\\left[ \\begin{array}{r}$a\\end{array} \\right]"""
    end
    function toeqnarray(x::AbstractArray{Sym,2})
        sz = size(x)
        a = join([join("\\displaystyle " .* map(sympy.latex, x[i,:]), "&")
                for i in 1:sz[1]], "\\\\")
        "\\left[ \\begin{array}{" * repeat("r",sz[2]) * "}" * a * "\\end{array}\\right]"
    end
    print(io, as_markdown(toeqnarray(x)))
end
```

In [3]:
```julia
function pvalue_tdist(x̄, s², n, μ)
    t = (x̄ - μ)/√(s²/n)
    2ccdf(TDist(n-1), abs(t))
end

function pvalue_tdist(x, μ)
    x̄, s², n = mean(x), var(x), length(x)
    pvalue_tdist(x̄, s², n, μ)
end

function confint_tdist(x̄, s², n; α = 0.05)
    c = quantile(TDist(n-1), 1-α/2)
    [x̄ - c*√(s²/n), x̄ + c*√(s²/n)]
end

function confint_tdist(x; α = 0.05)
    x̄, s², n = mean(x), var(x), length(x)
    confint_tdist(x̄, s², n; α)
end

confdist_tdist(x̄, s², n) = x̄ + √(s²/n)*TDist(n-1)
confdist_tdist(x) = confdist_tdist(mean(x), var(x), length(x))

preddist_tdist(x̄, s², n) = x̄ + √(s²*(1 + 1/n))*TDist(n-1)
preddist_tdist(x) = preddist_tdist(mean(x), var(x), length(x))
```

Out[3]:
```
preddist_tdist (generic function with 2 methods)
```

1 正規分布モデルの共役事前分布とその応用

1.1 逆ガンマ正規分布

平均 $\mu \in \mathbb{R}$, 分散 $v = \sigma^2 \in \mathbb{R}_{>0}$ の正規分布の確率密度函数を次のように表す:

$$p_{\mathrm{Normal}}(y|\mu, v) = \frac{1}{\sqrt{2\pi v}} \exp\left(-\frac{1}{2v}(y-\mu)^2\right) \quad (y \in \mathbb{R}).$$

分散パラメータ σ^2 を v に書き直している理由は, σ^2 を1つの変数として扱いたいからである.

パラメータ $\kappa, \theta > 0$ の逆ガンマ分布の確率密度函数を次のように書くことにする:

$$p_{\mathrm{InverseGamma}}(v|\kappa, \theta) = \frac{\theta^\kappa}{\Gamma(\kappa)} v^{-\kappa-1} \exp\left(-\frac{\theta}{v}\right) \quad (v > 0).$$

v がこの逆ガンマ分布に従う確率変数だとすると,

$$\frac{1}{v} \sim \mathrm{Gamma}\left(\kappa, \frac{1}{\theta}\right) = \frac{1}{2\theta}\mathrm{Gamma}\left(\frac{2\kappa}{2}, 2\right) = \frac{1}{2\theta}\mathrm{Chisq}(2\kappa),$$
$$E[v] = \frac{\theta}{\kappa-1}, \qquad \mathrm{var}(v) = \frac{E[v]^2}{\kappa-2}.$$

A と B が μ, v に関する定数因子の違いを除いて等しいことを $A \propto B$ と書くことにする.

逆ガンマ正規分布の密度函数を次のように定義する:

$$\begin{aligned}p_{\mathrm{InverseGammaNormal}}(\mu, v|\mu_*, v_*, \kappa, \theta) &= p_{\mathrm{Normal}}(\mu|\mu_*, v_* v) p_{\mathrm{InverseGamma}}(v|\kappa, \theta) \\ &\propto v^{-(\kappa+1/2)-1} \exp\left(-\frac{1}{v}\left(\theta + \frac{1}{2v_*}(\mu-\mu_*)^2\right)\right).\end{aligned}$$

この逆ガンマ正規分布の密度函数に従う確率変数を μ, v と書くと,

$$E[v] = \frac{\theta}{\kappa-1}, \quad \mathrm{var}(v) = \frac{E[v]^2}{\kappa-2}, \quad \mathrm{cov}(\mu, v) = 0, \quad E[\mu] = \mu_*, \quad \mathrm{var}(\mu) = v_* E[v].$$

この逆ガンマ正規分布が正規分布の共役事前分布になっていることを次の節で確認する.

1.2 共役事前分布のBayes更新

データの数値 $y_1, \ldots, y_n$ が与えられたとき, 正規分布モデルの尤度函数は

$$\prod_{i=1}^n p_{\mathrm{Normal}}(y_i|\mu, v) \propto v^{-n/2} \exp\left(-\frac{1}{2v}\sum_{i=1}^n (y_i - \mu)^2\right)$$

の形になる. このとき,

$$\bar{y} = \frac{1}{n}\sum_{i=1}^n y_i, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n (y_i - \bar{y})^2.$$

とおくと,

$$\sum_{i=1}^n (y_i - \mu)^2 = n(\mu - \bar{y})^2 + n\hat{\sigma}^2$$

なので, 尤度を最大化する μ, v は $\mu = \bar{y}, v = \hat{\sigma}^2$ になることがわかる.

さらに, 次が成立することもわかる:

$$\begin{aligned}
&\prod_{i=1}^n p_{\mathrm{Normal}}(y_i|\mu, v) \times p_{\mathrm{InverseGammaNormal}}(\mu, v|\mu_*, v_*, \kappa, \theta) \\
&\propto v^{-n/2} \exp\left(-\frac{n}{2v}\left((\mu - \bar{y})^2 + \hat{\sigma}^2\right)\right) \times v^{-(\kappa+1/2)-1} \exp\left(-\frac{1}{v}\left(\theta + \frac{1}{2v_*}(\mu - \mu_*)^2\right)\right) \\
&= v^{-(\kappa+n/2+1/2)-1} \exp\left(-\frac{1}{v}\left(\theta + \frac{n}{2}\left(\hat{\sigma}^2 + \frac{(\bar{y} - \mu_*)^2}{1 + nv_*}\right) + \frac{1 + nv_*}{2v_*}\left(\mu - \frac{\mu_* + nv_*\bar{y}}{1 + nv_*}\right)^2\right)\right).
\end{aligned}$$

ゆえに共役事前分布から得られる事後分布のパラメータは次のようになる:

$$\begin{aligned}
\tilde{\kappa} &= \kappa + \frac{n}{2} = \frac{n}{2}\left(1 + \frac{2\kappa}{n}\right), \\
\tilde{\theta} &= \theta + \frac{n}{2}\left(\hat{\sigma}^2 + \frac{(\bar{y} - \mu_*)^2}{1 + nv_*}\right) = \frac{n\hat{\sigma}^2}{2}\left(1 + \frac{2\theta}{n\hat{\sigma}^2} + \frac{(\bar{y} - \mu_*)^2}{(1 + nv_*)\hat{\sigma}^2}\right), \\
\tilde{\mu}_* &= \frac{\mu_* + nv_*\bar{y}}{1 + nv_*} = \bar{y}\frac{1 + \mu_*/(nv_*\bar{y})}{1 + 1/(nv_*)}, \\
\tilde{v}_* &= \frac{v_*}{1 + nv_*} = \frac{1}{n}\frac{1}{1 + 1/(nv_*)}.
\end{aligned}$$

In [4]:
```
@vars n ȳ v̂ μ v μ0 v0 κ θ
```

Out[4]: (n, ȳ, v̂, μ, v, μ0, v0, κ, θ)

In [5]:
```
negloglik = n/2*log(v) + n/(2v)*((μ - ȳ)^2 + v̂)
```

Out[5]:

$$\frac{n\log(v)}{2} + \frac{n\left(\hat{v} + (-\bar{y} + \mu)^2\right)}{2v}$$

In [6]:
```
neglogpri = (κ + 1//2 + 1)*log(v) + 1/v*(θ + 1/(2v0)*(μ-μ0)^2)
```

Out[6]:

$$\left(\kappa + \frac{3}{2}\right)\log(v) + \frac{\theta + \frac{(\mu-\mu_0)^2}{2v_0}}{v}$$

In [7]:
```
neglogpost = (κ + n/2 + 1//2 + 1)*log(v) +  1/v*(
    θ + n/2*(v̂ + 1/(1+n*v0)*(ȳ - μ0)^2) +
    (1 + n*v0)/(2v0)*(μ - (μ0 + n*v0*ȳ)/(1 + n*v0))^2)
```

Out[7]:

$$\left(\frac{n}{2} + \kappa + \frac{3}{2}\right)\log(v) + \frac{\frac{n\left(\hat{v} + \frac{(\bar{y}-\mu_0)^2}{nv_0+1}\right)}{2} + \theta + \frac{\left(\mu - \frac{nv_0\bar{y}+\mu_0}{nv_0+1}\right)^2(nv_0+1)}{2v_0}}{v}$$

In [8]:
```
simplify(negloglik + neglogpri - neglogpost)
```

Out[8]: 0

1.3 μの周辺事前・事後分布および事前・事後予測分布

確率密度函数

$$p(\mu|\mu_*, v_*, \kappa, \theta) = \int_{\mathbb{R}_{>0}} p_{\mathrm{InverseGammaNormal}}(\mu, v|\mu_*, v_*, \kappa, \theta)\,dv$$

で定義されるμ の周辺事前分布は次になる:

$$\mu \sim \mu_* + \sqrt{\frac{\theta}{\kappa}v_*}\ \mathrm{TDist}(2\kappa).$$

確率密度函数

$$p_*(y_{\mathrm{new}}|\mu_*, v_*, \kappa, \theta) = \iint_{\mathbb{R}\times\mathbb{R}_{>0}} p_{\mathrm{Normal}}(y_{\mathrm{new}}|\mu, v) p_{\mathrm{InverseGammaNormal}}(\mu, v|\mu_*, v_*, \kappa, \theta)\,d\mu\,dv$$

で定義される y_{new} の事前予測分布は次になる:

$$y_{\mathrm{new}} \sim \mu_* + \sqrt{\frac{\theta}{\kappa}(1+v_*)}\ \mathrm{TDist}(2\kappa).$$

パラメータをBayes更新後のパラメータ

$$\begin{aligned}
\tilde{\kappa} &= \kappa + \frac{n}{2} = \frac{n}{2}\left(1 + \frac{2\kappa}{n}\right),\\
\tilde{\theta} &= \theta + \frac{n}{2}\left(\hat{\sigma}^2 + \frac{(\bar{y}-\mu_*)^2}{1+nv_*}\right) = \frac{n\hat{\sigma}^2}{2}\left(1 + \frac{2\theta}{n\hat{\sigma}^2} + \frac{(\bar{y}-\mu_*)^2}{(1+nv_*)\hat{\sigma}^2}\right),\\
\tilde{\mu}_* &= \frac{\mu_* + nv_*\bar{y}}{1+nv_*} = \bar{y}\frac{1+\mu_*/(nv_*\bar{y})}{1+1/(nv_*)},\\
\tilde{v}_* &= \frac{v_*}{1+nv_*} = \frac{1}{n}\frac{1}{1+1/(nv_*)}.
\end{aligned}$$

に置き換えればこれは μ の周辺事後分布および事後予測分布になる.

その事後分布を使った区間推定の幅は

- n が大きいほど狭くなる.
- κ が大きいほど狭くなる.
- θ が大きいほど広くなる.
- $|\bar{y}-\mu_*|/\hat{\sigma}$ が大きいほど広くなる.
- $|\bar{y}-\mu_*|/\hat{\sigma}$ が大きくても, v_* がさらに大きければ狭くなる.

1.4 Jeffreys事前分布の場合

パラメータ空間が $\{(\mu, v) = (\mu, \sigma^2) \in \mathbb{R}\times\mathbb{R}_{>0}\}$ の 2 次元の正規分布モデルのJeffreys事前分布 $p_{\mathrm{Jeffreys}}(\mu, v)$ は

$$p_{\mathrm{Jeffreys}}(\mu, v) \propto v^{-3/2}$$

になることが知られている. ただし, 右辺の $(\mu, v) \in \mathbb{R}\times\mathbb{R}_{>0}$ に関する積分は ∞ になるので, この場合のJeffreys事前分布はimproperである.

逆ガンマ正規分布の密度函数

$$p_{\mathrm{InverseGammaNormal}}(\mu, v|\mu_*, v_*, \kappa, \theta) \propto v^{-(\kappa+1/2)-1}\exp\left(-\frac{1}{v}\left(\theta + \frac{1}{2v_*}(\mu-\mu_*)^2\right)\right).$$

と比較すると, Jeffreys事前分布に対応する共役事前分布のパラメータ値は形式的に次になることがわかる:

$$\kappa \to 0, \quad \theta \to 0, \quad v_* \to \infty.$$

そのとき, Bayes更新後のパラメータの公式は次のようにシンプルになる:

$$\tilde{\kappa} = \frac{n}{2}, \quad \tilde{\theta} = \frac{n\hat{\sigma}^2}{2}, \quad \tilde{\mu}_* = \bar{y}, \quad \tilde{v}_* = \frac{1}{n}.$$

さらに, 前節の公式から, $n\to\infty$ のとき, 一般のパラメータ値に関するBayes更新の結果は, $n\to\infty$ のとき漸近的にこのJeffreys事前分布の場合に一致する.

さらに, Jeffreys事前分布の場合には

$$\frac{\tilde{\theta}}{\tilde{\kappa}} = \hat{\sigma}^2, \quad \tilde{v}_* = \frac{1}{n}, \quad 2\tilde{\kappa} = n.$$

ゆえに, μ に関する周辺事後分布は

$$\mu \sim \bar{y} + \frac{\hat{\sigma}}{\sqrt{n}}\ \mathrm{TDist}(n)$$

になり, 事後予測分布は次になる:

$$y_{\mathrm{new}} \sim \bar{y} + \hat{\sigma}\sqrt{1 + \frac{1}{n}}\ \mathrm{TDist}(n).$$

1.5 平均と対数分散について一様な事前分布の場合

平均 μ と分数の対数 $\log v = \log \sigma^2$ に関する一様な事前分布は

$$p_{\mathrm{flat}}(\mu, v) \propto v^{-1}$$

になる. ただし, 右辺の $(\mu, v) \in \mathbb{R} \times \mathbb{R}_{>0}$ に関する積分は ∞ になるので, この事前分布はimproperである.

逆ガンマ正規分布の密度函数

$$p_{\mathrm{InverseGammaNormal}}(\mu, v | \mu_*, v_*, \kappa, \theta) \propto v^{-(\kappa+1/2)-1} \exp\left(-\frac{1}{v}\left(\theta + \frac{1}{2v_*}(\mu - \mu_*)^2\right)\right).$$

と比較すると, 平均と対数分散について一様な事前分布に対応する共役事前分布のパラメータ値は形式的に次になることがわかる:

$$\kappa \to -\frac{1}{2}, \quad \theta \to 0, \quad v_* \to \infty.$$

このとき, Bayes更新後のパラメータの公式は次のようになる:

$$\tilde{\kappa} = \frac{n-1}{2}, \quad \tilde{\theta} = \frac{n\hat{\sigma}^2}{2}, \quad \tilde{\mu}_* = \bar{y}, \quad \tilde{v}_* = \frac{1}{n}.$$

この場合には

$$\frac{\tilde{\theta}}{\tilde{\kappa}} = \frac{n\hat{\sigma}^2}{n-1} = s^2, \quad \tilde{v}_* = \frac{1}{n}, \quad 2\tilde{\kappa} = n - 1.$$

ここで, s^2 はデータの数値 $y_1, \dots, y_n$ の不偏分散

$$s^2 = \frac{1}{n-1}\sum_{i=1}^n (y_i - \bar{y})^2 = \frac{n\hat{\sigma}^2}{n-1} > \hat{\sigma}^2$$

であり, s はその平方根である.

ゆえに, μ に関する周辺事後分布は

$$\mu \sim \bar{y} + \frac{s}{\sqrt{n}}\ \mathrm{TDist}(n-1)$$

になり, y_{new} に関する事後予測分布は次になる:

$$y_{\mathrm{new}} \sim \bar{y} + s\sqrt{1 + \frac{1}{n}}\ \mathrm{TDist}(n-1).$$

したがって, 前節の結果と比較すると, Jeffreys事前分布の事後分布と予測分布による区間推定よりもこの場合の区間推定は少し広くなる.

1.6 平均と対数分散について一様な事前分布の場合の結果の数値的確認

In [9]:
```
@model function normaldistmodel_flat(y)
    log_v ~ Flat()
    σ² = exp(log_v)
    μ ~ Flat()
    y ~ MvNormal(fill(μ, length(y)), σ²*I)
end
```

Out[9]:
```
normaldistmodel_flat (generic function with 2 methods)
```

In [10]:
```
μ_true, σ_true, n = 10.0, 3.0, 5
y = rand(Normal(μ_true, σ_true), n)
```

Out[10]:
```
5-element Vector{Float64}:
 11.842482465802226
  6.259923440993461
  9.034841005068056
 10.488612932484408
  8.350869289000904
```

In [11]:

```julia
L = 10^5
n_threads = min(Threads.nthreads(), 10)
chn = sample(normaldistmodel_flat(y), NUTS(), MCMCThreads(), L, n_threads)
```

```
┌ Info: Found initial step size
│   ϵ = 0.8
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.8
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.2
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.05
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.4
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.4
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.05
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.4
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.4
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.05
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
```

Out[11]:

```
Chains MCMC chain (100000×14×10 Array{Float64, 3}):

Iterations        = 1001:1:101000
Number of chains  = 10
Samples per chain = 100000
Wall duration     = 23.73 seconds
Compute duration  = 225.8 seconds
parameters        = log_v, μ
internals         = lp, n_steps, is_accept, acceptance_rate, log_density, hamiltonian_energy, ha
miltonian_energy_error, max_hamiltonian_energy_error, tree_depth, numerical_error, step_size, no
m_step_size

Summary Statistics
  parameters      mean       std   naive_se      mcse          ess      rhat  ⋯
      Symbol   Float64   Float64    Float64   Float64      Float64   Float64  ⋯

       log_v    1.7758    0.8035     0.0008    0.0014  306549.2305    1.0000  ⋯
           μ    9.1948    1.3417     0.0013    0.0026  273041.8922    1.0000  ⋯
                                                              1 column omitted

Quantiles
  parameters      2.5%     25.0%     50.0%     75.0%     97.5%
      Symbol   Float64   Float64   Float64   Float64   Float64

       log_v    0.4804    1.2071    1.6811    2.2386    3.6177
           μ    6.5507    8.4928    9.1944    9.8996   11.8411
```

In [12]:

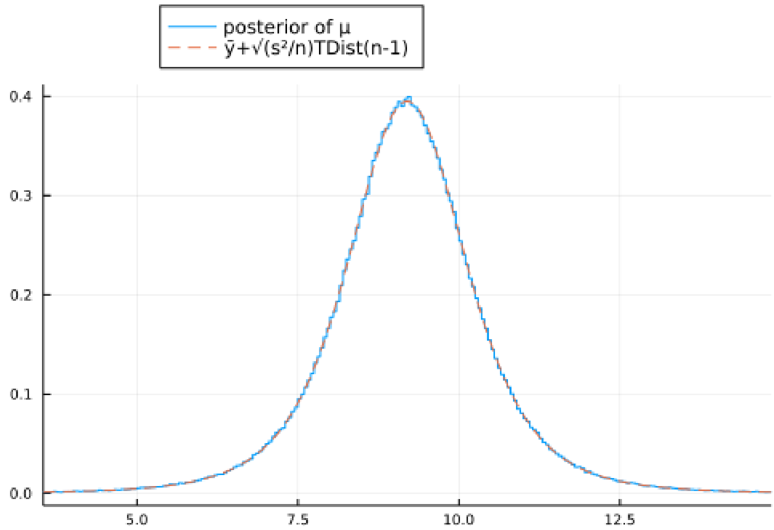
```julia
@show confint_tdist(y);
```

```
confint_tdist(y) = [6.558784143228545, 11.831907510111078]
```

In [13]:

```julia
dist_conf = confdist_tdist(y)
plot(legend=:outertop)
stephist!(vec(chn[:μ]); norm=true, label="posterior of μ")
plot!(dist_conf; label="ȳ+√(s²/n)TDist(n-1)", ls=:dash)
plot!(xlim=quantile.(dist_conf, (0.002, 0.998)))
```

Out[13]:

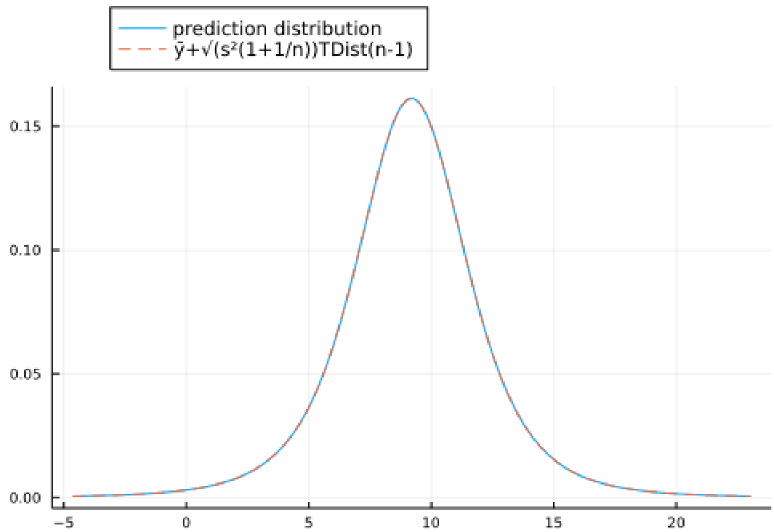


In [14]:

```julia
pdf_pred(y_new) = mean(pdf(Normal(μ, exp(0.5log_v)), y_new)
    for (μ, log_v) in zip(vec(chn[:μ]), vec(chn[:log_v])))
dist_pred = preddist_tdist(y)

plot(legend=:outertop)
plot!(pdf_pred, quantile.(dist_pred, (0.002, 0.998))...;
    label="prediction distribution")
plot!(y_new -> pdf(dist_pred, y_new);
    label="ȳ+√(s²(1+1/n))TDist(n-1)", ls=:dash)
```

Out[14]:



1.7 通常の信頼区間と予測区間との比較

通常の t 分布を使う平均の信頼区間と次の値の予測区間の構成では以下を使う:

$$\frac{\bar{y}-\mu}{s/\sqrt{n}} \sim \mathrm{TDist}(n-1), \qquad \frac{y_{\mathrm{new}}-\bar{y}}{s\sqrt{1+1/n}} \sim \mathrm{TDist}(n-1).$$

ここで, s^2 はデータの数値の不偏分散であり, s はその平方根である.

したがって, 前節の結果と比較すると, 通常の信頼区間と予測区間は, 平均と対数分散に関する一様事前分布に関する事後分布と予測分布を用いた区間推定に一致する.

1.8 データの数値から事前分布を決めた場合

$a, b > 0$ であると仮定する.

データの数値から共役事前分布のパラメータを次の条件によって決めたと仮定する:

$$E[\mu] = \mu_* = \bar{y}, \quad E[v] = \frac{\theta}{\kappa - 1} = \hat{\sigma}^2, \quad \mathrm{var}(\mu) = v_* E[v] = a\hat{\sigma}^2, \quad \mathrm{var}(v) = \frac{E[v]^2}{\kappa - 2} = b\hat{\sigma}^4.$$

これは次と同値である:

$$\mu_* = \bar{y}, \quad v_* = a, \quad \kappa = 2 + \frac{1}{b}, \quad \theta = \hat{\sigma}^2\left(1 + \frac{1}{b}\right).$$

これのBayes更新の結果は以下のようになる:

$$\begin{aligned}
\tilde{\kappa} &= 2 + \frac{1}{b} + \frac{n}{2} = \frac{n}{2}\left(1 + \frac{2(2+1/b)}{n}\right) && \to 2 + \frac{n}{2}, \\
\tilde{\theta} &= \hat{\sigma}^2\left(1 + \frac{1}{b} + \frac{n}{2}\right) + \frac{n}{2}\frac{(\bar{y}-\bar{y})^2}{1+na} = \frac{n\hat{\sigma}^2}{2}\left(1 + \frac{2(1+1/b))}{n}\right) && \to \hat{\sigma}^2\left(1 + \frac{n}{2}\right), \\
\tilde{\mu}_* &= \frac{\bar{y} + nv_*\bar{y}}{1 + nv_*} = \bar{y} && \to \bar{y}, \\
\tilde{v}_* &= \frac{a}{1+na} = \frac{1}{n}\frac{1}{1+1/(na)} && \to \frac{1}{n}.
\end{aligned}$$

以上における $\to$ は $a \to \infty$, $b \to \infty$ での極限を意味する.

以上の構成のポイントは, $\mu_* = \bar{y}$ となっているおかげで, $\tilde{\mu}_*$ も $\tilde{\mu}_* = \bar{y}$ となってバイアスが消え, さらに, $\tilde{\theta}$ の中の $\dfrac{n}{2}\dfrac{(\bar{y}-\mu_*)^2}{1+na}$ の項が消えて, 区間推定の幅が無用に広くならずに済むことである.

ただし, この場合には

$$\frac{\tilde{\theta}}{\tilde{\kappa}} = \hat{\sigma}^2\frac{1+2(1+1/b)/n}{1+2(2+1/b)/n} < \hat{\sigma}^2, \qquad v_* = \frac{1}{n}\frac{1}{1+1/(na)} < \frac{1}{n}$$

なので, 区間推定の幅はJeffreys事前分布の場合よりも少し狭くなる.

しかし, n が大きければそれらの違いは小さくなる.

In [15]:

```
function posterior(y;
        a = 2.5, b = 2.5, n = length(y), ȳ = mean(y), σ̂² = var(y; corrected=false),
    )
    μ_star = ȳ
    v_star = a/(1 + n*a)
    κ = 2 + 1/b + n/2
    θ = σ̂²*(1 + 1/b + n/2)
    (μ_star, v_star, κ, θ)
end

postdist_μ(μ_star, v_star, κ, θ) = μ_star + √(θ/κ * v_star) * TDist(2κ)
preddist(μ_star, v_star, κ, θ) = μ_star + √(θ/κ * (1 + v_star)) * TDist(2κ)
```

Out[15]:

```
preddist (generic function with 1 method)
```

1.9 n = 5 ではデフォルト事前分布の場合と無情報事前分布の場合の結果が結構違う.

In [16]:
```julia
@model function normaldistmodel(y;
        a = 2.5, b = 2.5, ȳ = mean(y), σ̂² = var(y; corrected=false),
        μ_star = ȳ, v_star = a, κ = 2 + 1/b, θ = σ̂²*(1 + 1/b)
    )
    σ² ~ InverseGamma(κ, θ)
    μ ~ Normal(μ_star, √(v_star * σ²))
    y ~ MvNormal(fill(μ, length(y)), σ²*I)
end
```

Out[16]:
```
normaldistmodel (generic function with 2 methods)
```

In [17]:
```julia
y
```

Out[17]:
```
5-element Vector{Float64}:
 11.842482465802226
  6.259923440993461
  9.034841005068056
 10.488612932484408
  8.350869289000904
```

In [18]:
```
chn = sample(normaldistmodel(y), NUTS(), MCMCThreads(), L, n_threads)
```

```
┌ Info: Found initial step size
│   ϵ = 0.4
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.2
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.8
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.05
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.4
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.2
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.05
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.05
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.8
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 0.05
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
```

Out[18]:
```
Chains MCMC chain (100000×14×10 Array{Float64, 3}):

Iterations        = 1001:1:101000
Number of chains  = 10
Samples per chain = 100000
Wall duration     = 21.81 seconds
Compute duration  = 175.57 seconds
parameters        = σ², μ
internals         = lp, n_steps, is_accept, acceptance_rate, log_density, hamiltonian_energy, ha
miltonian_energy_error, max_hamiltonian_energy_error, tree_depth, numerical_error, step_size, no
m_step_size

Summary Statistics
  parameters      mean       std   naive_se      mcse          ess      rhat  ⋯
      Symbol   Float64   Float64    Float64   Float64      Float64   Float64  ⋯

          σ²    3.6076    2.1076     0.0021    0.0030  513714.3064    1.0000  ⋯
           μ    9.1964    0.8180     0.0008    0.0011  598792.3096    1.0000  ⋯
                                                            1 column omitted

Quantiles
  parameters      2.5%     25.0%     50.0%     75.0%     97.5%
      Symbol   Float64   Float64   Float64   Float64   Float64

          σ²    1.3934    2.2838    3.0791    4.2831    8.9695
           μ    7.5687    8.6849    9.1958    9.7077   10.8263
```

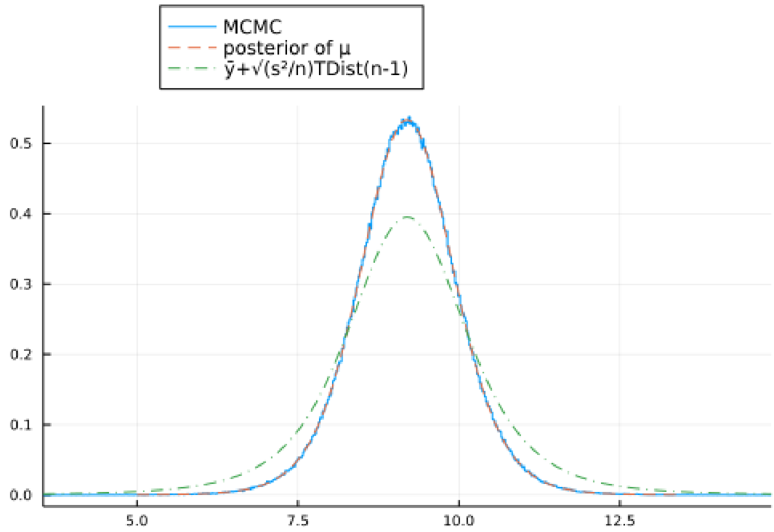
In [19]:
```
@show confint_tdist(y);
```

```
confint_tdist(y) = [6.558784143228545, 11.831907510111078]
```

In [20]:

```
dist_conf = confdist_tdist(y)
dist_post = postdist_μ(posterior(y)...)
plot(legend=:outertop)
stephist!(vec(chn[:μ]); norm=true, label="MCMC")
plot!(dist_post; label="posterior of μ", ls=:dash)
plot!(dist_conf; label="ȳ+√(s²/n)TDist(n-1)", ls=:dashdot)
plot!(xlim=quantile.(dist_conf, (0.002, 0.998)))
```

Out[20]:

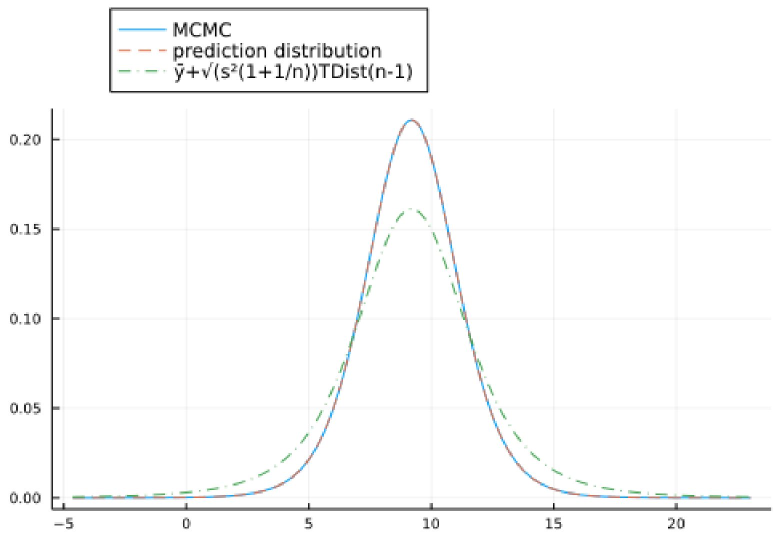


In [21]:

```
pdf_pred(y_new) = mean(pdf(Normal(μ, √σ²), y_new)
    for (μ, σ²) in zip(vec(chn[:μ]), vec(chn[:σ²])))
dist_pred_bayes = preddist(posterior(y)...)
dist_pred_tdist = preddist_tdist(y)
#xlim = quantile.(dist_pred_bayes, (0.002, 0.998))
xlim = quantile.(dist_pred_tdist, (0.002, 0.998))

plot(legend=:outertop)
plot!(pdf_pred, xlim...;
    label="MCMC")
plot!(dist_pred_bayes, xlim...;
    label="prediction distribution", ls=:dash)
plot!(dist_pred_tdist, xlim...;
    label="ȳ+√(s²(1+1/n))TDist(n-1)", ls=:dashdot)
```

Out[21]:



1.10 n = 20 ではデフォルト事前分布の場合と無情報事前分布の場合の結果が近付く.

In [22]:
```
# データの数値をかなり大きくする.
μ_true, σ_true, n = 1e4, 1e2, 20
@show dist_true = Normal(μ_true, σ_true) n
y = rand(dist_true, n);
```

```
dist_true = Normal(μ_true, σ_true) = Normal{Float64}(μ=10000.0, σ=100.0)
n = 20
```

In [23]:
```julia
chn = sample(normaldistmodel(y), NUTS(), MCMCThreads(), L, n_threads)
```

```
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
```

```
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
```

```
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
```

```
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 9.765625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
│
```

```
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 0.0001953125
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 2.44140625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 9.765625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191

┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
Sampling (10 threads): 100%|████████████████████████████| Time: 0:00:00
```

Out[23]:

```
Chains MCMC chain (100000×14×10 Array{Float64, 3}):

Iterations        = 1001:1:101000
Number of chains  = 10
Samples per chain = 100000
Wall duration     = 17.66 seconds
Compute duration  = 151.51 seconds
parameters        = σ², μ
internals         = lp, n_steps, is_accept, acceptance_rate, log_density, hamiltonian_energy, ha
miltonian_energy_error, max_hamiltonian_energy_error, tree_depth, numerical_error, step_size, no
m_step_size

Summary Statistics
  parameters         mean         std    naive_se      mcse           ess     ⋯
      Symbol      Float64     Float64     Float64   Float64       Float64   Flo ⋯

          σ²   11761.4226   3643.0566      3.6431    4.2257   759629.9461    1. ⋯
           μ   10002.1003     24.0140      0.0240    0.0249   873401.2031    1. ⋯
                                                             2 columns omitted

Quantiles
  parameters         2.5%       25.0%       50.0%       75.0%       97.5%
      Symbol      Float64     Float64     Float64     Float64     Float64

          σ²    6646.1856   9214.8721  11110.2744  13569.6704  20658.5740
           μ    9954.6223   9986.3725  10002.1061  10017.8546  10049.5262
```

In [24]:

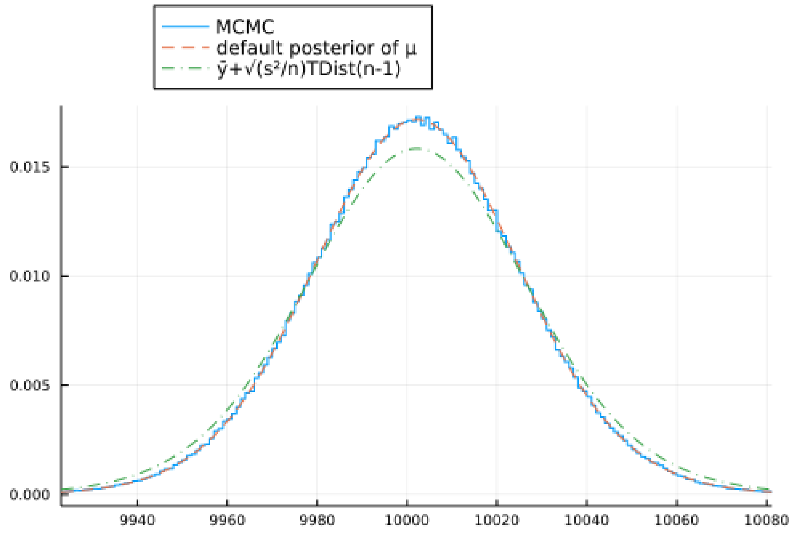
```julia
@show confint_tdist(y);
```

```
confint_tdist(y) = [9950.075345254349, 10054.199644479408]
```

In [25]:

```
dist_conf = confdist_tdist(y)
dist_post = postdist_μ(posterior(y)...)
plot(legend=:outertop)
stephist!(vec(chn[:μ]); norm=true, label="MCMC")
plot!(dist_post; label="default posterior of μ", ls=:dash)
plot!(dist_conf; label="ȳ+√(s²/n)TDist(n-1)", ls=:dashdot)
plot!(xlim=quantile.(Ref(vec(chn[:μ])), (0.001, 0.999)))
```

Out[25]:

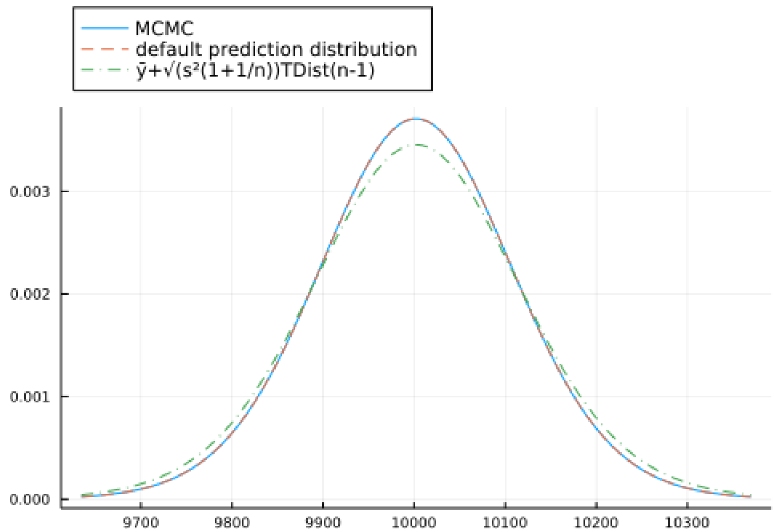


In [26]:

```
pdf_pred(y_new) = mean(pdf(Normal(μ, √σ²), y_new)
    for (μ, σ²) in zip(vec(chn[:μ]), vec(chn[:σ²])))
dist_pred_bayes = preddist(posterior(y)...)
dist_pred_tdist = preddist_tdist(y)
xlim = quantile.(dist_pred_bayes, (0.001, 0.999))
#xlim = quantile.(dist_pred_tdist, (0.001, 0.999))

plot(legend=:outertop)
plot!(pdf_pred, xlim...;
    label="MCMC")
plot!(dist_pred_bayes, xlim...;
    label="default prediction distribution", ls=:dash)
plot!(dist_pred_tdist, xlim...;
    label="ȳ+√(s²(1+1/n))TDist(n-1)", ls=:dashdot)
```

Out[26]:



1.11 n = 20 で事前分布とデータの数値の相性が悪い場合

In [27]:

```
a, b = 10.0, 10.0
μ_star, v_star, κ, θ = 0.0, a, 2 + 1/b, 1 + 1/b
@show μ_star v_star κ θ
Eμ, Ev = μ_star, θ/(κ - 1)
var_μ, var_v = v_star*Ev, Ev^2/(κ - 2)
@show Eμ Ev var_μ var_v
model = normaldistmodel(y; μ_star, v_star, κ, θ)
```

```
μ_star = 0.0
v_star = 10.0
κ = 2.1
θ = 1.1
Eμ = 0.0
Ev = 1.0
var_μ = 10.0
var_v = 9.999999999999991
```

Out[27]:

```
DynamicPPL.Model{typeof(normaldistmodel), (:y, :a, :b, :ȳ, :σ̂², :μ_star, :v_star, :κ, :θ), (:a,
:b, :ȳ, :σ̂², :μ_star, :v_star, :κ, :θ), (), Tuple{Vector{Float64}, Float64, Float64, Float64, Fl
oat64, Float64, Float64, Float64, Float64}, NTuple{8, Float64}, DynamicPPL.DefaultContext}(norma
ldistmodel, (y = [10166.062790496948, 10008.826245713864, 10099.687439746831, 10052.31698949658
7, 9906.384613622007, 9971.965667388273, 10213.945709762043, 9812.776061487099, 10006.8365844506
82, 9966.833374320784, 10006.56265386828, 10199.270140637116, 9920.849533678655, 9904.1120416149
72, 9982.079189890888, 10006.20676985534, 9832.637200506008, 10070.920004305975, 9882.7849352951
43, 10031.691951200066], a = 2.5, b = 2.5, ȳ = 10002.137494866878, σ̂² = 11755.723252897847, μ_st
ar = 0.0, v_star = 10.0, κ = 2.1, θ = 1.1), (a = 2.5, b = 2.5, ȳ = 10002.137494866878, σ̂² = 1175
5.723252897847, μ_star = 10002.137494866878, v_star = 2.5, κ = 2.4, θ = 16458.012554056986), Dyn
amicPPL.DefaultContext())
```

In [28]:

```
chn = sample(model, NUTS(), MCMCThreads(), L, n_threads)
```

```
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
```

```
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
```

```
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 9.765625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 9.765625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
```

```
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 9.765625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 2.44140625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 4.8828125e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191

┌ Info: Found initial step size
│   ϵ = 2.44140625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Info: Found initial step size
│   ϵ = 2.44140625e-5
└ @ Turing.Inference D:\.julia\packages\Turing\szPqN\src\inference\hmc.jl:191
┌ Warning: The current proposal will be rejected due to numerical error(s).
│   isfinite.((θ, r, ℓπ, ℓκ)) = (true, false, false, false)
└ @ AdvancedHMC D:\.julia\packages\AdvancedHMC\51xgc\src\hamiltonian.jl:47
Sampling (10 threads): 100%|████████████████████████████| Time: 0:00:02
```

Out[28]:

```
Chains MCMC chain (100000×14×10 Array{Float64, 3}):

Iterations        = 1001:1:101000
Number of chains  = 10
Samples per chain = 100000
Wall duration     = 16.24 seconds
Compute duration  = 156.81 seconds
parameters        = σ², μ
internals         = lp, n_steps, is_accept, acceptance_rate, log_density, hamiltonian_energy, ha
miltonian_energy_error, max_hamiltonian_energy_error, tree_depth, numerical_error, step_size, no
m_step_size

Summary Statistics
  parameters          mean          std   naive_se       mcse           ess   ⋯
      Symbol       Float64      Float64    Float64    Float64       Float64   ⋯

          σ²   459019.7855  144892.4124   144.8924   166.8436   727782.4295   ⋯
           μ     9952.7599     151.3976     0.1514     0.1668   837843.1094   ⋯
                                                               2 columns omitted

Quantiles
  parameters          2.5%        25.0%        50.0%        75.0%         9 ⋯
      Symbol       Float64      Float64      Float64      Float64       Flo ⋯

          σ²   257081.1217  357850.1936  432828.7365  530120.7663   813043. ⋯
           μ     9654.0826    9853.5857    9952.5870   10052.0031    10251. ⋯
                                                                1 column omitted
```

In [29]:

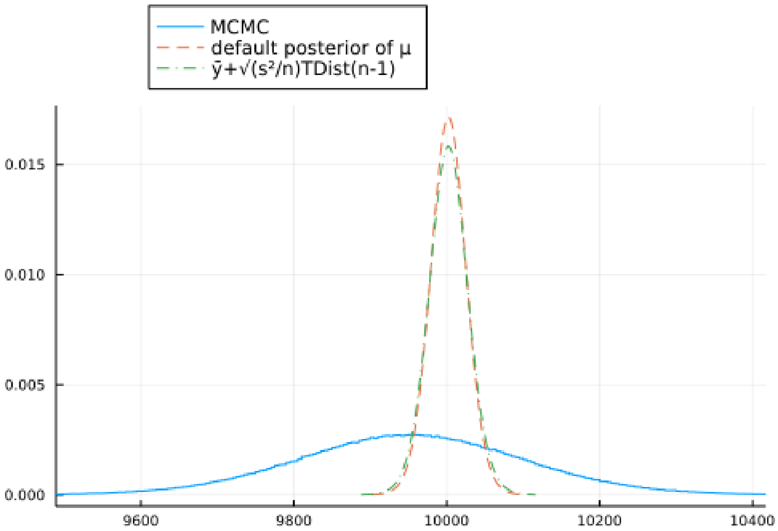
```
1 @show confint_tdist(y);
```

```
confint_tdist(y) = [9950.075345254349, 10054.199644479408]
```

In [30]:

```julia
dist_conf = confdist_tdist(y)
dist_post = postdist_μ(posterior(y)...)
plot(legend=:outertop)
stephist!(vec(chn[:μ]); norm=true, label="MCMC")
plot!(dist_post; label="default posterior of μ", ls=:dash)
plot!(dist_conf; label="ȳ+√(s²/n)TDist(n-1)", ls=:dashdot)
plot!(xlim=quantile.(Ref(vec(chn[:μ])), (0.002, 0.998)))
```

Out[30]:

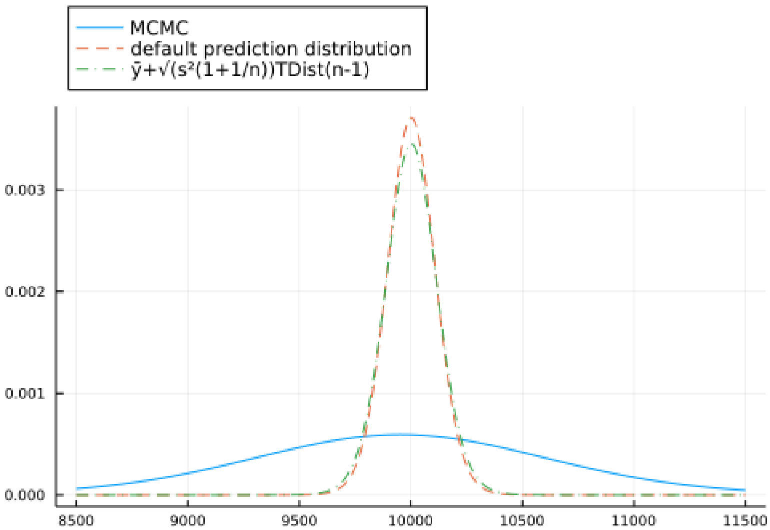


In [31]:

```julia
pdf_pred(y_new) = mean(pdf(Normal(μ, √σ²), y_new)
    for (μ, σ²) in zip(vec(chn[:μ]), vec(chn[:σ²])))
dist_pred_bayes = preddist(posterior(y)...)
dist_pred_tdist = preddist_tdist(y)
#xlim = quantile.(dist_pred_bayes, (0.002, 0.998))
#xlim = quantile.(dist_pred_tdist, (0.002, 0.998))
xlim = (8500, 11500)

plot(legend=:outertop)
plot!(pdf_pred, xlim...;
    label="MCMC")
plot!(dist_pred_bayes, xlim...;
    label="default prediction distribution", ls=:dash)
plot!(dist_pred_tdist, xlim...;
    label="ȳ+√(s²(1+1/n))TDist(n-1)", ls=:dashdot)
```

Out[31]:



In [ ]:

```julia

```