Kullback-Leibler情報量と記述統計

黒木玄

2019-09-13〜2019-09-23, 2019-10-15, 2020-01-03, 2020-11-02

このファイルのJupyter notebook版は

- Kullback-Leibler情報量と記述統計 Jupyter notebook版 (https://nbviewer.jupyter.org/github/genkuroki/Statistics/blob/master/KL%20information%20and%20descriptive%20statistics.ipyn

で閲覧でき, PDF版は

- Kullback-Leibler情報量と記述統計 PDF版 (https://genkuroki.github.io/documents/Statistics/KL%20information%20and%20descriptive%20statistics.pdf)

で閲覧できる.

目次

In [1]:

```
using Base64
displayfile(mime, file; tag="img") = open(file) do f
    display("text/html", """<$tag src="data:$mime;base64,$(base64encode(f))">""")
end

using Distributions
using Statistics
using StatsBase
using LaTeXStrings

using StatsPlots
using Plots
default(:bglegend, plot_color(default(:bg), 0.5))
default(:fglegend, plot_color(ifelse(isdark(plot_color(default(:bg))), :white, :black),
0.6));
pal = palette(:default)

pyplotclf() = if backend() == Plots.PyPlotBackend(); PyPlot.clf(); end

function retina(P::Plots.Plot; scale=3)
    Q = deepcopy(P)
    Q[:dpi] = round(Int, scale*P[:dpi])
    base64 = base64encode(show, MIME("image/png"), Q)
    mime = "image/png"
    width = P[:size][1]
    display("text/html", """<img width=$(width) src="data:$(mime);base64,$(base64)"/>""")
    pyplotclf()
end

retina(; scale=3) = (P -> retina(P; scale=scale))

function retinasavefig(P::Plots.Plot, fn::AbstractString; scale=3)
    Q = deepcopy(P)
    Q[:dpi] = round(Int, scale*P[:dpi])
    savefig(Q, fn)
    pyplotclf()
end

function retinasavedisp(P::Plots.Plot, fn::AbstractString; scale=3)
    retinasavefig(P, fn; scale=scale)
    displayfile("image/png", fn; tag="img width=$(P[:size][1])")
    pyplotclf()
end

retinasavedisp(fn::AbstractString; scale=3) = (P -> retinasavedisp(P, fn; scale=scale))

pyplot(fmt=:auto, size=(400, 250), titlefontsize=10)

using Random
using QuadGK

using SpecialFunctions
SpecialFunctions.lgamma(x::Real) = logabsgamma(x)[1]
```

1 Kullback-Leibler情報量とSanovの定理

1.1 Stirlingの公式

正の整数 n の階乗 $n!$ は $n \to \infty$ で

$$n! \sim n^n e^{-n} \sqrt{2\pi n} \quad \left(\frac{n!}{n^n e^{-n} \sqrt{2\pi n}} \to 1 \right) \tag{$*$}$$

と近似される. これを**Stirlingの公式**と呼ぶ. 確率関係の計算では階乗が頻出なのでこの公式は近似公式として非常に便利であり, よく使われる. 以下でこれを証明しよう.

証明の出発点として, $n!$ がガンマ函数によって次のように表わされることを採用しよう:

$$n! = \Gamma(n+1) = \int_0^\infty e^{-x} x^n \, dx.$$

右辺の積分は被積分函数が大きな部分の x に関する局所的な積分で近似されるはずである. そこで被積分函数の対数

$$f(x) = \log(e^{-x}x^n) = -x + n\log x$$

の最大点の近くでの概形を確認しよう. $f'(x) = -1 + n/x$ より, $f'(n) = 0$ となり, $f(x)$ $(x > 0)$ は $x = n$ で最大になることがわかる. さらに, $f''(x) = -n/x^2, f'''(x) = 2n/x^3, f^{(4)}(x) = -3!/x^4$ なので $f(n) = n\log n - n, f'(n) = 0, f''(n) = -1/n$, $f'''(n) = 2/n^2, f^{(4)}(n) = -3!/n^4$ なので, $f(x)$ を $x = n$ においてTaylor展開すると,

$$f(n+t) = n\log n - n - \frac{t^2}{2n} + \frac{t^3}{3n^2} - \frac{t^4}{4n^3} + \cdots.$$

ゆえに, $t = \sqrt{n}\,y$ とおくと,

$$f(n+\sqrt{y}) = n\log n - n - \frac{y^2}{2} + O(1/\sqrt{n}).$$

ゆえに, $(*)$で $x = n + \sqrt{n}\,y$ とおくと,

$$n! = \int_{-\sqrt{n}}^{\infty} \exp\left(n\log n - n - \frac{y^2}{2} + O(1/\sqrt{n})\right)\sqrt{n}\,dy = n^n e^{-n}\sqrt{n}\int_{-\sqrt{n}}^{\infty}\exp\left(-\frac{y^2}{2} + O(1/\sqrt{n})\right)dy.$$

そして, Gauss積分の公式 $\int_{-\infty}\infty e^{-y^2/a}\,dy = \sqrt{a\pi}\ (a > 0)$より,

$$\int_{-\sqrt{n}}^{\infty}\exp\left(-\sqrt{n}\,y + n\log\left(1 + \frac{y}{\sqrt{n}}\right)\right)dy \to \int_{-\infty}^{\infty}\exp\left(-\frac{y^2}{2}\right)dy = \sqrt{2\pi}.$$

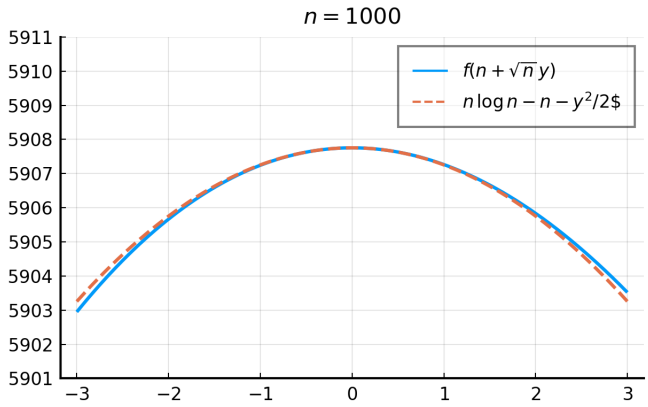
以上をまとめるとStirlingの公式が得られる.

証明のポイントは $n!$ を積分表示しておくことと, 積分変数の変換 $x = n + \sqrt{n}$ によって, Stirlingの公式の $n^n e^{-n}\sqrt{n}$ の部分が即得られることである. $\sqrt{2\pi}$ の因子は極限で出て来るGauss積分から得られる.

補足: 以上において, ガンマ函数 $\Gamma(s) = \int_0^{\infty} e^{-x}x^{s-1}\,dx\ (s > 0)$ やGauss積分 $\int_{-\infty}^{\infty} e^{-x^2/a}\,dx = \sqrt{a\pi}\ (a > 0)$ やTaylor展開 $\log(1+z) = z - z^2/2 + z^3/3 - z^4/4 + \cdots\ (|z| < 1)$ を用いた. これらはちょうど大学1年の微積分の講義で習う事柄である. 高校数学を習得後に新たに習う積分や函数としてGauss積分やガンマ函数は優先度が最上位に属するものだと思われる. そして, 微分学について学ぶことはTaylorの定理(Taylor展開)について学ぶことだと言ってもよいくらい, Taylor展開は多用される基本的な道具になっている. □

グラフ: 被積分函数 $f(x) = n\log x - x$ と $x = n + \sqrt{n}\,y$ で得られるその近似 $n\log n - n - y^2/2$ を $n = 1000$ で比較してみよう.

In [2]:
```
n = 1000
f(x) = n*log(x) - x
g(y) = n*log(n) - n - y^2/2
y = range(-3, 3, length=400)
plot(title="\$n = $n\$", ylim=(5901, 5911), ytick=5901:5911)
plot!(y, @.(f(n + √n*y)), label=L"f(n+\sqrt{n}\,y)", lw=1.5)
plot!(y, g.(y), label=L"n\,\log\,n - n - y^2/2\$", lw=1.5, ls=:dash) ▷ retina
```



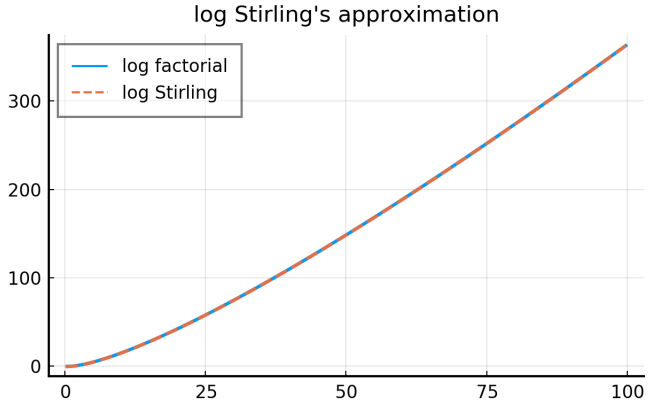
確かによく一致している. $y < 0$ で f より $n\log n - n - y^2/2$ が大きく, $y > 0$ ではその逆になる. 実際に積分すると, その違いが互いに打ち消し合うことによって精度が上がる仕組みになっている.

グラフ: 対数版のStirlingの公式

$$\log n! = n\log n - n + \frac{1}{2}\log n + \log\sqrt{2\pi} + o(1)$$

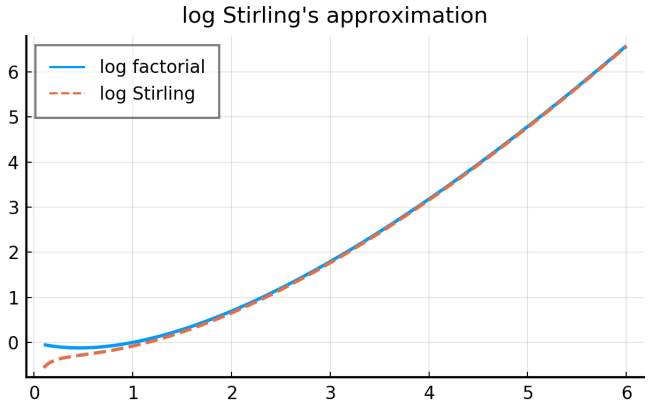
を数値的に確認してみよう. 以下のプロットを見よ.

In [3]:
```
logfact(n) = lgamma(n+1)
logstirling(n) = n*log(n) - n + 1/2*log(n) + log(√(2π))
x = range(0.1, 100, length=400)
plot(title="log Stirling's approximation")
plot!(x, logfact.(x), label="log factorial", lw=1.5)
plot!(x, logstirling.(x), label="log Stirling", lw=1.5, ls=:dash) ▷ retina
```



このようにStirlingの公式による近似の精度は非常に高く, ほとんどぴったり一致しているように見える.

In [4]:
```
x = range(0.1, 6, length=400)
plot(title="log Stirling's approximation")
plot!(x, logfact.(x), label="log factorial", lw=1.5)
plot!(x, logstirling.(x), label="log Stirling", lw=1.5, ls=:dash) ▷ retina
```



階乗とStirlingの公式による近似があまり一致しないのは n が小さな場合だけである.

1.2 KL情報量

$i = 1, 2, \ldots, r$ の目がそれぞれ p_i の確率で出るサイコロを n 回ふったときに, i の目が k_i 回出る確率は次のように表わされる:

$$\frac{n!}{k_1!k_2!\cdots k_r!}p_1^{k_1}p_2^{k_2}\cdots p_r^{k_r} = n!\prod_{i=1}^{r}\frac{p_i^{k_i}}{k_i!}.$$

$\sum_{i=1}^{r} k_i = n$ などが成立していることに注意せよ. $q_i \geqq 0$, $\sum_{i=1}^{r} q_i = 1$ を満たす固定された q_i 達を与えて,

$$k_i \approx nq_i$$

が成立していると仮定し, さらに, Stirlingの公式 $k! \approx k^k e^{-k}\sqrt{2\pi k}$ を上の確率の公式の中の階乗のそれぞれに適用すると,

$$n!\prod_{i=1}^{r}\frac{p_i^{k_i}}{k_i!} \approx n^n e^{-n}\sqrt{2\pi n}\prod_{i=1}^{r}\frac{p_i^{nq_i}}{(nq_i)^{nq_i}e^{nq_i}\sqrt{2\pi nq_i}} = \frac{1}{\sqrt{(2\pi n)^{r-1}q_1\cdots q_r}}\left(\prod_{i=1}^{r}\frac{p_i}{q_i}\right)^{nq_i}.$$

ゆえに

$$\log\left(n!\prod_{i=1}^{r}\frac{p_i^{k_i}}{k_i!}\right) = -n\sum_{i=1}^{r} q_i \log\frac{q_i}{p_i} + o(n).$$

すなわち,

$$D(q||p) = \sum_{i=1}^{r} q_i \log\frac{q_i}{p_i}$$

とおくと,

$$n!\prod_{i=1}^{r}\frac{p_i^{k_i}}{k_i!} = \exp(-nD(q||p) + o(n)) \quad \text{if } k_i \approx nq_i \qquad \text{(Sanovの定理)}.$$

これは, i の目が p_i の確率で出るサイコロを n 回ふった結果がまるで i の目が出る確率が q_i であるかのように見える確率が, n についてほぼ指数函数的に減少し, その減少の指数函数的な速さが $D(q||p)$ であることを意味している. その減少の速さは確率分布 p_i が確率分布 q_i をどれだけよく近似しているかの指標の1つとみなされる.

以下では $D(q||p)$ を確率分布 p_i による確率分布 q_i のシミュレーションの**予測誤差**と呼んだり, **Kullback-Leibler情報量**と呼んだりすることにする.

以上と同様の結果は連続的な確率分布 $p(x), q(x)$ についても, 予測誤差＝KL情報量を

$$D(q||p) = \int q(x)\log\frac{q(x)}{p(x)}\,dx \quad \text{(右辺はある定積分)}$$

と定めれば成立している(以下では主にこちら側を使うことになる). すなわち次が成立している.

Sanovの定理: 確率分布 $p(x)$ の乱数列 $X_1, \ldots, X_n$ がまるで確率分布 $q(x)$ の乱数列であるかのように見える確率は, $n \to \infty$ において

$$\exp(-nD(q||p) + o(n))$$

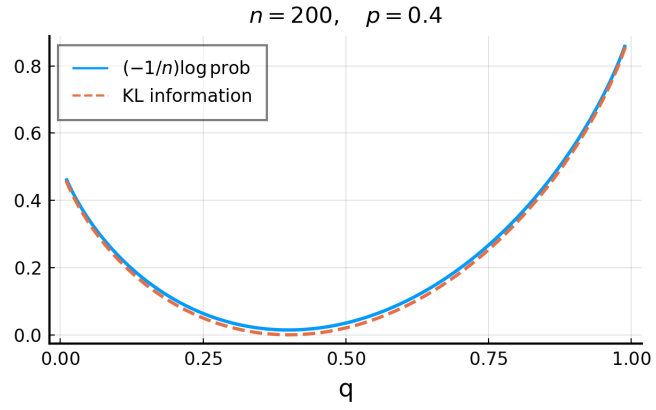
のように振る舞う. (「〜のように見える」の定義の曖昧さは $o(n)$ の項に吸収される.) □

グラフ: $r = 2$ の二項分布の確率の場合のSanovの定理を数値的に確認してみよう. そのためには $k = nq$ の場合に

$$-\frac{1}{n}\log\frac{n!p^k(1-p)^{n-k}}{k!(n-k)!}, \quad q\log\frac{q}{p} + (1-q)\log\frac{1-q}{1-p}$$

を比較すればよい. $n = 200, p = 0.4$ でこれら2つを比較してみよう.

In [5]:
```
logprob_bin(n, k, p) = -1/n * (logfact(n) + k*log(p) + (n-k)*log(1-p) - logfact(k) -
logfact(n-k))
kl_bin(q, p) = q*log(q/p) + (1-q)*log((1-q)/(1-p))
n = 200
p = 0.4
q = range(0.01, 0.99, step=0.005)
plot(xlabel="q", title="\$n = $n, \\quad p = $p\$")
plot!(q, logprob_bin.(n, n*q, p), label=L"(-1/n)\log\,\mathrm{prob}", lw=1.5)
plot!(q, kl_bin.(q, p), label="KL information", lw=1.5, ls=:dash) ▷ retina
```



確かに2つはよく一致している.

1.3 汎化誤差

ある特定の確率分布の集合 $\mathcal{P}$ に含まれる確率分布 p で固定された確率分布 q を最もよく予測するものを見つけたければ, $\mathcal{P}$ に含まれる分布 p で予測誤差 $D(q||p)$ を最小になるものを見付ければよい.

$$D(q||p) = G(q||p) - S(q), \quad G(q||p) = -\int q(x)\log p(x)\,dx, \quad S(q) = G(q||q)$$

なので, 実際には　$\mathcal{P}$ に含まれる分布 p で $G(q||p)$ を最小にするものを求めればよい. このとき, $G(q||p)$ を分布 p による分布 q の予測に関する**汎化誤差**と呼ぶ.

一般に $G(q||p)$ は分布 p の分布 q への**不適合度**を表す. 以下ではKL情報量 $D(q||p)$ そのものよりも, p の q への不適合度 $G(q||p)$ の方を主に利用することになる.

1.4 尤度函数と汎化誤差の関係と最尤法の定義

未知の分布 $q(x)$ に従う乱数発生の独立試行によって乱数列 $X_1, \dots, X_n$ が得られたとする. このとき, $X_1, \dots, X_n$ は未知の母集団分布 $q(x)$ のサイズ n の**サンプル** (**標本**)であるという.

このとき, 分布 $p(x)$ で未知の母集団分布 $q(x)$ を近似していると期待されるものが与えられているとき, $p(x)$ を**予測分布**と呼ぶ.

母集団分布 $q(x)$ のサンプル $X_1, \dots, X_n$ と予測分布 $p(x)$ に対して,

$$p(X_1)\dots p(X_n)$$

を予測分布のサンプルに関する尤度と呼ぶ. 尤度の対数の $-1/n$ 倍

$$-\frac{1}{n}\sum_{k=1}^{n}\log p(X_k)$$

は $n \to \infty$ のとき, 大数の法則によって, 分布 $q(x)$ に関する $-\log p(x)$ の平均値

$$\int q(x)(-\log p(x))\,dx = G(q||p)$$

に収束する. すなわち, 予測分布 $p(x)$ の対数尤度の $-1/n$ 倍は汎化誤差に収束する.

これは, 対数尤度の $-1/n$ 倍が汎化誤差の推定量として使用可能であることを意味している.

分布 p の分布 q の予測に関する汎化誤差は分布 p の分布 q への不適合度という意味を持っているのであった. 汎化誤差は対数尤度の $-1/n$ 倍で推定可能なので, 尤度が高い予測分布 p をうまく見付けることができれば予測誤差が小さな予測分布が得られると考えられる(警告: これは常に正しいとは限らない. 後の方の注意を参照せよ).

この考え方に基く推定法が最尤法である. すなわち, パラメーター $w = (w_1, \dots, w_d)$ を持つ確率分布 $p(x|w)$ の尤度函数

$$w \mapsto p(X_1|w)\cdots p(X_n|w)$$

を最大化する $w = \hat{w}$ を $p(x|w)$ に代入してできる $\hat{p}(x) = p(x|\hat{w})$ を予測分布とみなす推定法を**最尤法**と呼ぶ.

注意: サンプル $X_1, \dots, X_n$ の分布を

$$q_{\text{sample}}(x) = \frac{1}{n}\sum_{k=1}^{n}\delta(x - X_k)$$

と書くと, 分布 $p(x|w)$ の対数尤度の $-1/n$ 倍は

$$G(w) = G(q_{\text{sample}}||p(\cdot|w)) = \int q_{\text{sample}}(x)(-\log p(x|w))\,dx = -\frac{1}{n}\sum_{k=1}^{n}\log p(X_k|w)$$

と表わされる. すなわち, 対数尤度の $-1/n$ 倍 $G(w)$ は分布 $p(x|w)$ のサンプルの分布への不適合度にちょうどなっている. だから, 尤度を最大にすること(=対数尤度の $-1/n$ 倍を最小化すること)は分布 $p(x|w)$ のサンプルの分布に最も適合するようにパラメーター w を調節することに等しい.

しかし, 我々が欲しいのはサンプルを生成した未知の分布 $q(x)$ に適合するような予測分布の方である.

もしも, 大量のパラメーターを含む確率分布のパラメーターを調節して, サンプルにぴったりフィットする予測分布を作れば, その予測分布の尤度は当然大きくなる. しかし, 尤度＝予測分布のサンプルへの適合度を予測分布の性能を測る指標として使うことは, 予測分布を作るために使ったサンプルと同じサンプルで予測分布の性能を測っていることになり、無意味なことをしている可能

性が出て来てしまう. だから, 尤度そのものを予測分布の性能評価に使うべきではない. このような点に配慮した最尤法の予測分布の汎化誤差の実用的な推定量は赤池弘次氏によって作られ, AICと呼ばれている:

$$\mathrm{AIC} = -2\sum_{k=1}^{n} \log p(X_k|\hat{w}) + d.$$

対数尤度の -2 倍にモデルのパラメーターの個数 d を足して, AICは定義される. モデルのパラメーターの個数を増やすとAICは上昇する. AICが小さい方の予測分布の方が予測精度が高いと推測するので(注意!この推測は確率的に間違うことがある!), AICを使った予測分布の選択ではパラメーターの個数が多いモデルは不利になる. □

補足: パラメーターを増やしたモデルの最尤法で予測精度が下がる現象については以下のリンク先のノートブックの例を参照せよ:

- 過学習の過程の動画 LASSO版 2018-03-25, 2018-04-04, 2018-09-19 (https://nbviewer.jupyter.org/gist/genkuroki/c08b416648d4d7db4948ffac6abeadfd)

特に $n = 16$, degree $= 5, 7, 13$, LASSO $\lambda = 0.0$ の動画を参照せよ. 推定の過程でも大きく汎化誤差(generalization error)が上昇している. □

注意: 有限サイズ n のサンプルを使った最尤推定の結果は, 確率モデルが十分に単純なものであれば, $n \to \infty$ で収束する. 問題は有限の n でどれだけ収束しているかである. 尤度函数は推定の収束に関する豊富な情報を持っている. しかし, 最尤法では尤度函数を最大化するパラメーターの値 $\hat{w}$ から予測分布 $\hat{p}(x) = p(x|\hat{w})$ を作っているので, 尤度函数が持っていた推定の収束具合に関する情報を捨ててしまっている. この点は尤度函数全体の情報をシンプルな方法で利用するベイズ統計の方法を使えば改善される. □

2 平均と分散

2.1 平均と分散の特徴付け

確率変数 X について, $X - a$ の二乗平均 $E[(X-a)^2]$ を最小にする a は X の平均 $E[X]$ に等しいことを示そう.

$E[(X-a)^2]$ を a で微分して 1/2 をかけると,

$$\frac{1}{2}\frac{\partial}{\partial a}E[(X-a)^2] = -E[X-a] = a - E[X].$$

これより $E[(X-a)^2]$ は a の函数として $a > E[X]$ で単調増加し, $a < E[X]$ で単調減少する. ゆえに $E[(X-a)^2]$ は $a = E[X]$ で最小になる. そのときの $E[(X-a)^2]$ は X の分散と呼ばれる.

このように, 確率変数の平均と分散はそれぞれ $E[(X-a)^2]$ を最小にする a とその最小値として特徴付けられる.

2.2 平均と分散の推定は正規分布モデルによる推定に等しい

$\mathbb{R}$ 上の分布 $q(x)$ を最もよく予測する正規分布を見付けるには,

$$p(x) = p(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-(x-\mu)^2/(2\sigma^2)}$$

とおくとき, 汎化誤差

$$G(q||p) = -\int q(x)\log p(x|\mu,\sigma^2)\,dx$$

を最小化する μ, σ^2 を求めればよい. 実際に求めてみよう. まず, $q(x)$ の平均と分散をそれぞれ μ_0, σ_0^2 と書く:

$$\mu_0 = \int q(x)x\,dx, \quad \sigma_0^2 = \int q(x)(x-\mu_0)^2\,dx.$$

一方,

$$\begin{aligned}-2\log p(x||\mu,\sigma^2) &= \log(2\pi) + \log\sigma^2 + \frac{(x-\mu^2)}{\sigma^2} \\ &= \log(2\pi) + \log\sigma^2 + \frac{(x-\mu_0)^2 + (\mu-\mu_0)(x-\mu_0) + (\mu-\mu_0)^2}{\sigma^2}\end{aligned}$$

なので,

$$
\begin{aligned}
2G(q||p) &= \log(2\pi) + \log\sigma^2 + \frac{\sigma_0^2 + (\mu-\mu_0)(\mu_0-\mu_0) + (\mu-\mu_0)^2}{\sigma^2} \\
&= \log(2\pi) + \log\sigma^2 + \frac{\sigma_0^2 + (\mu-\mu_0)^2}{\sigma^2}.
\end{aligned}
$$

σ^2 を固定するとき, これを最小化する μ は $\mu = \mu_0$ となる. そして, $\log\sigma^2 + \frac{\sigma_0^2}{\sigma^2}$ を最小化する σ^2 が σ_0^2 となることも容易に確かめられる. (例えば両辺を σ^2 を1つの変数だと思って微分すると $(\sigma^2 - \sigma_0^2)/\sigma^4$ となることから, そのことが確かめられる.)

これで汎化誤差 $G(q||p)$ を最小化する μ, σ^2 はそれぞれ μ_0, σ_0^2 になることがわかった.

これは, 分布 $q(x)$ を最もよく予測する正規分布が, $q(x)$ と同じ平均と分散を正規分布であることを意味している.

分布 $q(x)$ の平均と分散を推定することは, 分布 $q(x)$ を最もよく予測する正規分布を推定することに等しい.

このように, 平均と分散の推定は, 正規分布モデルによる推定に等しい.

2.3 平均と分散の正規分布モデルによる最尤推定

サンプル $X_1, \ldots, X_n$ に関する正規分布モデルの対数尤度の $-2/n$ 倍は

$$
\hat{\mu} = \frac{1}{n}\sum_{k=1}^{n} X_k, \quad \hat{\sigma}^2 = \frac{1}{n}\sum_{k=1}^{n}(X_k - \hat{\mu})^2
$$

とおくと, 前節とまったく同様にして,

$$
\begin{aligned}
-\frac{2}{n}\sum_{k=1}^{n}\log p(X_k|\mu,\sigma^2) &= \frac{1}{n}\sum_{k=1}^{n}\left(\log(2\pi) + \log\sigma^2 + \frac{(X_k-\hat{\mu})^2 + (\mu-\hat{\mu})(X_k-\hat{\mu}) + (\mu-\hat{\mu})^2}{\sigma^2}\right) \\
&= \log(2\pi) + \log\sigma^2 + \frac{\hat{\sigma}^2 + (\mu-\hat{\mu})^2}{\sigma^2}
\end{aligned}
$$

と書ける. 前節と同様にして, これを最小化する μ, σ^2 はそれぞれ $\hat{\mu}, \hat{\sigma}^2$ になることがわかる.

すなわち, 正規分布モデルの最尤法での解はサンプルの平均と分散に等しい.

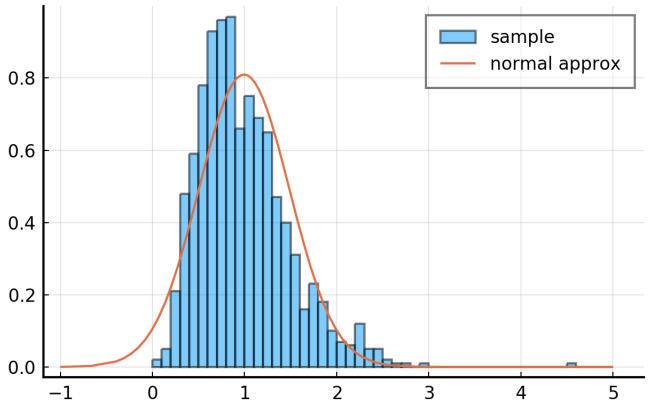
逆に言えば, サンプルの平均と分散を求めることは, 正規分布モデルの最尤法を実行することに等しい.

2.4 平均と分散の推定の例

In [6]:

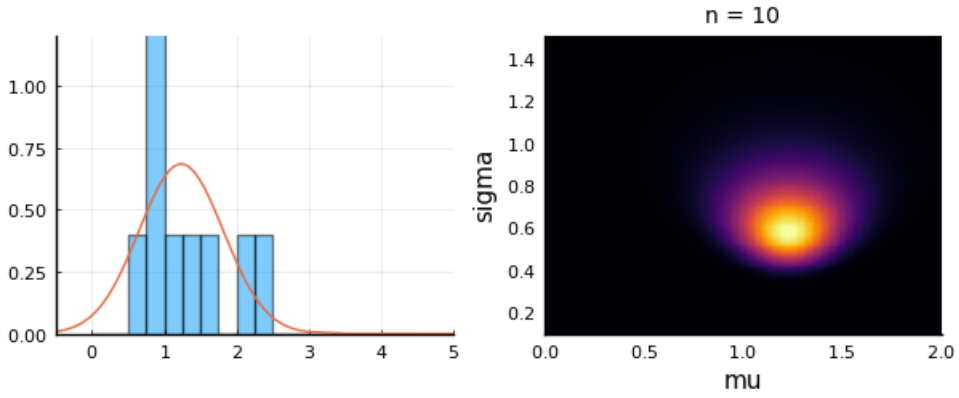
```
Random.seed!(37346491)

gdist = Gamma(4.0, 0.25)
@show mean(gdist)
@show std(gdist)

X = rand(gdist, 1000)
@show mean(X)
@show std(X; corrected=false)

@show normal_gdist = fit_mle(Normal, X)

sleep(0.1)
histogram(X; normed=true, bins=range(0, 5, step=0.1), alpha=0.5, label="sample")
plot!(x -> pdf(normal_gdist, x), -1, 5; label="normal approx") ▷ retina
```

```
mean(gdist) = 1.0
std(gdist) = 0.5
mean(X) = 0.9955804447660762
std(X; corrected = false) = 0.49273267483475164
normal_gdist = fit_mle(Normal, X) = Normal{Float64}(μ=0.9955804447660762, σ=0.4927326748347515)
```



In [7]:
```
# 尤度函数のプロット

Random.seed!(37346491)

gdist = Gamma(4.0, 0.25)
N = 1000
XX = rand(gdist, N)

ns = [10;10;10;10;10:3:300;300;300;300;300]
@time anim = @animate for n in ns
    X = @view XX[1:n]

    normal_gdist = fit_mle(Normal, X)
    P1 = histogram(X; normed=true, bins=range(0, 5, step=0.25), alpha=0.5, label="sample")
    plot!(x -> pdf(normal_gdist, x); label="normal approx")
    plot!(legend=false, xlim=(-0.5, 5), ylim=(0, 1.2))

    loglik(μ, σ) = sum(logpdf(Normal(μ, σ), x) for x in X)
    μ = range(0, 2, length=200)
    σ = range(0.1, 1.5, length=200)
    P2 = heatmap(μ, σ, exp.(loglik.(μ', σ)); xlabel="mu", ylabel="sigma",
        size=(300, 320), colorbar=false, title="n = $n")

    plot(P1, P2, size=(600, 250))
end
pyplotclf()
gif(anim, "images/lik_normal.gif", fps=10)
displayfile("image/gif", "images/lik_normal.gif")
```

```
49.264732 seconds (33.65 M allocations: 1.412 GiB, 0.76% gc time, 9.21% compilation time)
```




```
┌ Info: Saved animation to
│   fn = C:\Users\genkuroki\OneDrive\work\Statistics\images\lik_normal.gif
└ @ Plots C:\Users\genkuroki\.julia\packages\Plots\5ItHH\src\animation.jl:104
```

3 中央値

3.1 中央値の定義

確率変数 X の中央値 a_0 は $P(X \leqq a_0) \leqq 1/2$ かつ $P(X \geqq a_0) \geqq 1/2$ という条件で定義される.

例えば, $P(X < a_0) = 0.3$, $P(X = a_0) = 0.3$, $P(X > a_0) = 0.4$ のとき, a_0 は確率変数 X の中央値になる.

中央値は一意的に決まるとは限らない. 例えば $1, 2, 3, 4, 5, 6$ のどれかの値に等確率でなる確率変数 X の中央値に a_0 がなるための必要十分条件は $3 \leqq a_0 \leqq 4$ である. $a_0 = 3$ のとき $P(X \leqq a_0) = 3/6 = 1/2$ かつ $P(X \geqq a_0) = 4/6 > 1/2$ となる.

確率変数 X の中央値 a_0 について $P(X = a_0) = 0$ ならば, $P(X < a_0) = P(X > a_0) = 1/2$ となる. 逆にこの条件が満たされるとき, a_0 は $P(X = a_0) = 0$ となるような X の中央値になる.

3.2 中央値の性質

定理: 確率変数 X の中央値 a_0 は a の函数 $E[|X - a|]$ を $a = a_0$ で最小化する.

証明: 実数 a_0 は確率変数 X の中央値であると仮定する. すなわち, $P(X \leqq a_0) \geqq 1/2$ かつ $P(X \geqq a_0) \geqq 1/2$ であると仮定する.

$P(X=a_0)=0$ ならば $r=1/2$ とおき, $P(x=a_0)>0$ ならば $P(X<a_0)+rP(X=a_0)=1/2$ となるように実数 r を定める. このとき, $P(X>a_0)+(1-r)P(X=a_0)=1/2$ となる. さらに, $P(X<a_0)+P(X=a_0)\geqq 1/2$ と $P(X<a_0)+rP(X=a_0)=1/2$ の差を考えることによって, $(1-r)P(X=a_0)\geqq 0$ を得るので, $r\leqq 1$ となることがわかる. 同様に, $P(X>a_0)+P(X=a_0)\geqq 1/2$ と $P(X>a_0)+(1-r)P(X=a_0)=1/2$ の差を考えることによって, $rP(X=a_0)\geqq 0$ を得るので, $r\geqq 0$ となることもわかる. これで,

$$0\leqq r\leqq 1,\quad P(X<a_0)+rP(X=a_0)=1/2,\quad P(X>a_0)+(1-r)P(X=a_0)=1/2$$

が成立するようにできた. $r=1$ ならば $P(X\geqq a_0)=P(X<a_0)=1/2$ となり, $r=0$ ならば $P(X\leqq a_0)=P(X>a_0)=1/2$ となることにも注意せよ.

記号 1_A で X に関する条件 A を満たすとき 1 になり, そうでないとき 0 になる X の函数を表す. このとき, 上の段落の結果より,

$$0=E[1_{X<a_0}+r1_{X=a_0}]-E[1_{X>a_0}+(1-r)1_{X=a_0}].$$

これの両辺に $a-a_0$ をかけたものを

$$E[|X-a_0|]=E[(X-a_0)(1_{X>a_0}+r1_{X=a_0})]-E[(X-a_0)(1_{X<a_0}+(1-r)1_{X=a_0})]$$

の両辺から引くと次が得られる:

$$E[|X-a_0|]=E[(X-a)(1_{X>a_0}+r1_{X=a_0})]-E[(X-a)(1_{X<a_0}+(1-r)1_{X=a_0})].$$

これを

$$E[|X-a|]=E[(X-a)1_{X>a}]-E[(X-a)1_{X<a}]$$

の両辺から引くと,

$$\begin{aligned}E[|X-a|]-E[|X-a_0|]=&E[(X-a)(1_{X>a}-1_{X>a_0}-r1_{X=a_0})]\\&-E[(X-a)(1_{X<a}-1_{X<a_0}-(1-r)1_{X=a_0})].\end{aligned}$$

(i) $a<a_0$ と仮定する. このとき

$$1_{X>a}-1_{X>a_0}=1_{a<X<a_0}+1_{X=a_0},\quad 1_{X<a}-1_{X<a_0}=-1_{a<X<a_0}-1_{X=a}$$

なので

$$\begin{aligned}(X-a)(1_{X>0}-1_{X>a_0}-r1_{X=a_0})&=(X-a)(1_{a<X<a_0}+(1-r)1_{X=a_0}),\\-(X-a)(1_{X<0}-1_{X<a_0}-(1-r)1_{X=a_0})&=(X-a)(1_{a<X<a_0}+(1-r)1_{X=a_0}).\end{aligned}$$

ゆえに

$$E[|X-a|]-E[|X-a_0|]=2E[(X-a)(1_{a<X<a_0}+(1-r)1_{X=a_0})]\geqq 0.$$

(ii) $a>a_0$ と仮定する. このとき

$$1_{X>a}-1_{X>a_0}=-1_{a_0<X<a}-1_{X=a},\quad 1_{X<a}-1_{X<a_0}=1_{a_0<X<a}+1_{X=a_0}$$

なので

$$\begin{aligned}(X-a)(1_{X>0}-1_{X>a_0}-r1_{X=a_0})&=-(X-a)(1_{a_0<X<a}+r1_{X=a_0}),\\-(X-a)(1_{X<0}-1_{X<a_0}-(1-r)1_{X=a_0})&=-(X-a)(1_{a_0<X<a}+r1_{X=a_0}).\end{aligned}$$

ゆえに

$$E[|X-a|]-E[|X-a_0|]=2E[-(X-a)(1_{a_0<X<a}+r1_{X=a_0})]\geqq 0.$$

以上によって, 実数 a_0 が確率変数 X の中央値ならば a の函数 $E[|X-a|]$ が $a=a_0$ で最小になることがわかった. □

3.3 中央値の特徴付け

上の議論は初等的だが煩雑である. この節では前節の結果の逆も成立することを示す. そのために以下の補題を用いる.

補題: 可積分な確率変数 X について, 任意の実数 a,b について次の公式が成立する:

$$E[|X-b|]-E[|X-a|]=\int_a^b (P(X\leqq t)-P(X\geqq t))\,dt.$$

証明: t,X に関する条件 A が成立するときに 1, そうでないとき 0 になる t,X の函数を 1_A と書くことにする.

$$P(X\leqq t)-P(X\geqq t)=E[1_{X\leqq t}-1_{X\geqq t}]$$

であることと, x が正, 負のときそれぞれ $1, -1$ になる $\mathrm{sign}(x)$ について,

$$|X-a| = \int_X^a \mathrm{sign}(t-X)\,dt$$

が成立していることより,

$$|X-b| - |X-a| = \int_a^b \mathrm{sign}(t-X)\,dt = \int_a^b (1_{X\leqq t} - 1_{X\geqq t})\,dt$$

より, X について期待値を取る操作と t に関する積分を交換すれば欲しい公式が得られる. □

定理: 可積分な確率変数 X について, X の中央値であることと, a の函数 $E[|X-a|]$ を最小化することは同値である.

証明: 補題より,

$$E[|X-b|] - E[|X-a|] = \int_a^b (P(X\leqq t) - P(X\geqq t))\,dt = \int_b^a (P(X\geqq t) - P(X\leqq t))\,dt.$$

t の函数として $f(t) = P(X\leqq t) - P(X\geqq t)$ は広義単調増加函数であり, $-f(t) = P(X\geqq t) - P(X\leqq t)$ は広義単調減少函数であることに注意せよ.

a は X の中央値だと仮定する: $P(X\leqq a)\geqq 1/2$, $P(X\geqq a)\geqq 1/2$.

(i) $a<b$ のとき, $a<t<b$ ならば

$$P(X\leqq t)\geqq P(X\leqq a)\geqq 1/2,\quad P(X\geqq t)\leqq P(X>a)\leqq 1/2$$

なので

$$f(t) = P(X\leqq t) - P(X\geqq t)\geqq 1/2 - 1/2 = 0$$

となるから, $E[|X-b|] - E[|X-a|] = \int_a^b (P(X\leqq t) - P(X\geqq t))\,dt \geqq 0$ となる.

(ii) $b<a$ のとき, $b<t<a$ ならば

$$P(X\geqq t)\geqq P(X\geqq a)\geqq 1/2,\quad P(X\leqq t)\leqq P(X<a)\leqq 1/2$$

なので

$$-f(t) = P(X\geqq t) - P(X\leqq t)\geqq 1/2 - 1/2 = 0$$

となるから, $E[|X-b|] - E[|X-a|] = \int_b^a (P(X\geqq t) - P(X\leqq t))\,dt \geqq 0$ となる.

これで, 中央値 a に関する $E[|X-a|]$ が $E[|X-b|]$ の中で最小になることがわかった.

逆の対偶を証明するために, a は X の中央値ではないと仮定する.

t の函数として $P(X\leqq t)$ は右連続であり(右から t に近付くときに連続), $P(X\geqq t)$ は左連続であることに注意せよ.

(i) $P(X\leqq a)<1/2$ と仮定する. このとき,

$$f(t) = P(X\leqq t) - P(X\geqq t)\leqq P(X\leqq t) - P(X>t) = 2P(X\leqq t) - 1$$

より, $f(a)<0$ となる. さらに $P(X\leqq t)$ の t に関する右連続性より, ある実数 $b>a$ で $f(b)<0$ となるものが存在することがわかる. $f(t)$ は広義単調増加函数なので, そのとき,

$$E[|X-b|] - E[|X-a|] = \int_a^b f(t)\,dt \leqq (b-a)f(b) < 0$$

となり, $E[|X-a|]$ は $E[|X-b|]$ 達の中で最小にならない.

(ii) $P(X\geqq a)<1/2$ と仮定する. このとき,

$$-f(t) = P(X\geqq t) - P(X\leqq t)\leqq P(X\geqq t) - P(X<t) = 2P(X\geqq t) - 1$$

より, $-f(a)<0$ となる. さらに $P(X\geqq t)$ の t に関する左連続性より, ある実数 $b<a$ で $f(b)<0$ となるものが存在することがわかる. $-f(t)$ は広義単調減少函数なので, そのとき,

$$E[|X-b|] - E[|X-a|] = \int_b^a (-f(t))\,dt \leqq (a-b)f(b) < 0$$

となり, $E[|X-a|]$ は $E[|X-b|]$ 達の中で最小にならない.

これで示すべきことがすべて示された. □

3.4 中央値と中央値との差の絶対値の平均とLaplace分布モデルによる推定

$\mathbb{R}$ 上の分布 $q(x)$ を最もよく予測するLaplace分布

$$p(x) = p(x|a,b) = \frac{1}{2b}e^{-|x-a|/b}$$

を見付けるためには, 汎化誤差

$$G(q||p) = -\int q(x)\log p(x|a,b)\,dx = \log 2 + \log b + \frac{1}{b}\int q(x)|x-a|\,dx$$

を最小化する a, b を求めればよい. 分布 $q(x)$ に従う確率変数 X について

$$E[|X-a|] = \int q(x)|x-a|\,dx$$

なので, b を固定したときに汎化誤差を最小化する a は分布 $q(x)$ の中央値になる. 以下, a は $q(x)$ の中央値 a_0 に等しいと仮定し, $b_0 = E[|X-a_0|]$ とおく. このとき, $\log b + b_0/b$ を最小化する b は b_0 に等しい. これで, 汎化誤差を最小化する a, b はそれぞれ分布 $q(x)$ の中央値と中央値との差の絶対値の平均になることがわかった.

これは, 未知の分布 $q(x)$ の中央値と中央値との差の絶対値の平均を推定することはLaplace分布でも分布 $q(x)$ を最もよく予測するものを推定することに等しいことがわかる.

3.5 Laplace分布モデルによる最尤推定

サンプル $X_1, \ldots, X_n$ に関する正規分布モデルの対数尤度の $-1/n$ 倍は, 前節と同様に,

$$-\frac{1}{n}\sum_{k=1}^{n}\log p(X_k|a,b) = \log 2 + \log b + \frac{1}{b}\sum_{k=1}^{n}|X_k - a|$$

と書ける. やはり, 前節と同様にして, これを最小化する a, b がそれぞれ

$$\hat{a} = (X_1, \ldots, X_n \text{の中央値}), \quad \hat{b} = \frac{1}{n}\sum_{k=1}^{n}|X_k - \hat{a}|$$

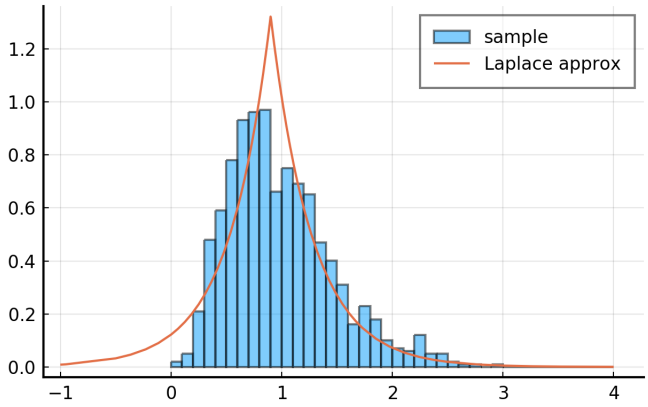
に等しいこともわかる. これは, サンプルの中央値と中央値との差の絶対値の平均を求めることは, Laplace分布モデルによる最尤法に等しいことを意味している.

3.6 中央値の推定の例

In [8]:

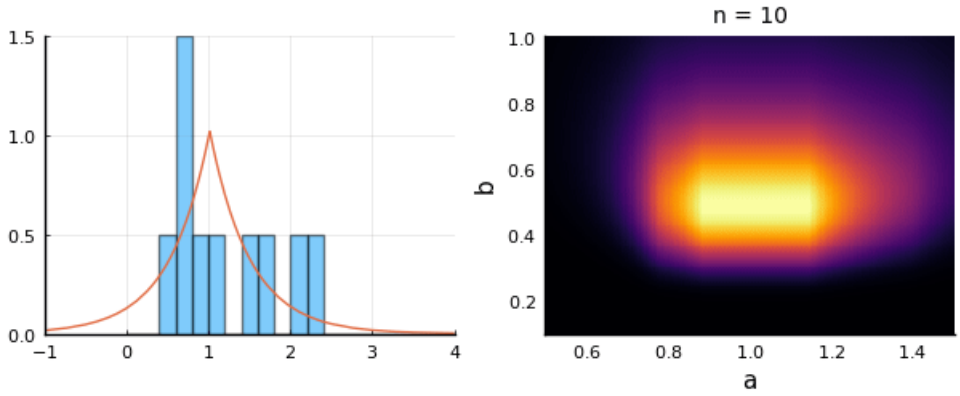
```
Random.seed!(37346491)

gdist = Gamma(4.0, 0.25)
@show a₀ = median(gdist)
b₀ = quadgk(x->pdf(gdist, x)*abs(x-a₀), 0, Inf)[1]
@show b₀

X = rand(gdist, 1000)
@show a_hat = median(X)
@show b_hat = mean(abs(x - a_hat) for x in X)

@show laplace_gdist = Laplace(a_hat, b_hat)

sleep(0.1)
histogram(X; bins=range(0, 4, step=0.1), normed=true, alpha=0.5, label="sample")
plot!(x -> pdf(laplace_gdist, x), -1, 4; label="Laplace approx") ▷ retina
```

```
a₀ = median(gdist) = 0.9180151872127242
b₀ = 0.3852143558605696
a_hat = median(X) = 0.901053827946683
b_hat = mean((abs(x - a_hat) for x = X)) = 0.3773964067342682
laplace_gdist = Laplace(a_hat, b_hat) = Laplace{Float64}(μ=0.901053827946683, θ=0.3773964067342
682)
```



In [9]:
```
# 尤度函数のプロット

Random.seed!(37346491)

gdist = Gamma(4.0, 0.25)
N = 1000
XX = rand(gdist, N)

ns = [10;10;10;10;10;10:10:1000;1000;1000;1000;1000]
@time anim = @animate for n in ns
    X = @view XX[1:n]

    a_hat = median(X)
    b_hat = mean(abs(x - a_hat) for x in X)
    laplace_gdist = Laplace(a_hat, b_hat)
    P1 = histogram(X; bins=range(0, 4, step=0.2), normed=true, alpha=0.5, label="sample")
    plot!(x -> pdf(laplace_gdist, x), -1, 4; label="Laplace approx")
    plot!(legend=false, xlim=(-1, 4), ylim=(0, 1.5))

    loglik(a, b) = sum(logpdf(Laplace(a, b), x) for x in X)
    a = range(0.5, 1.5, length=200)
    b = range(0.1, 1.0, length=200)
    P2 = heatmap(a, b, exp.(loglik.(a', b)); xlabel="a", ylabel="b",
        size=(300, 320), colorbar=false, title="n = $n")

    plot(P1, P2, size=(600, 250))
end
pyplotclf()
gif(anim, "images/lik_laplace.gif", fps=10)
displayfile("image/gif", "images/lik_laplace.gif")
```

```
55.465508 seconds (28.05 M allocations: 1.052 GiB, 0.45% gc time, 2.75% compilation time)
```




```
┌ Info: Saved animation to
│   fn = C:\Users\genkuroki\OneDrive\work\Statistics\images\lik_laplace.gif
└ @ Plots C:\Users\genkuroki\.julia\packages\Plots\5ItHH\src\animation.jl:104
```

4 最頻値

4.1 最頻値の特徴付け

確率変数 X の確率分布が確率密度函数 $q(x)$ を持つとき,

$$E[\delta(X-a)] = \int q(x)\delta(x-a)\,dx = q(a)$$

を最大化する $a = a_0$ を確率変数 X の**最頻値**と呼ぶ. 最頻値は一意に決まるとは限らない. 最頻値 a_0 における $q(x)$ の値 $q(a_0)$ の大きさは最頻値への分布の集中の大きさを表しているとみなされる.

最頻値についても, ある確率モデルによる推定と関係付けることができる. しかし, デルタ函数 $\delta(x-a)$ を用いているので, 平均値や中央値の場合と違ってすっきりとは行かず, 近似的な関係に留まる.

4.2 最頻値の近似的な推定

母集団分布 $q(x)$ のサンプル $X_1, \dots, X_n$ を考える. 確率変数 X は分布 $q(x)$ に従うと仮定する.

このとき, 函数 $f(x)$ は $x=0$ の周囲のみで大きな値を取り, $x=0$ から十分離れるとほぼ定数であるような函数であるとする. このような函数の極端な場合がデルタ函数 $\delta(x)$ であるとみなされる. そして, 経験損失函数 $L_n(a)$ を

$$L_n(a) = -\frac{1}{n}\sum_{k=1}^{n} f(X_k - a)$$

と定める. 大数の法則より, これは $n \to \infty$ で

$$L(a) = -\int q(x)\log f(x-a)\,dx == -E[f(X-a)].$$

に収束する. $f(x)$ がデルタ函数に近ければ, $E[f(X-a)]$ は $E[\delta(X-a)] = q(a)$ に近くなるので, $L(a)$ を最小化する a を求めることは近似的に $q(a)$ を最大化する a すなわち分布 $q(x)$ の最頻値を求めることになると考えられる. さらに, 経験損失函数 $L_n(a)$ を $L(a)$ の近似とみなせば, $L_n(a)$ を最小化する a を求めることは, $q(x)$ の最頻値の近似値を推定することになっていると考えられる.

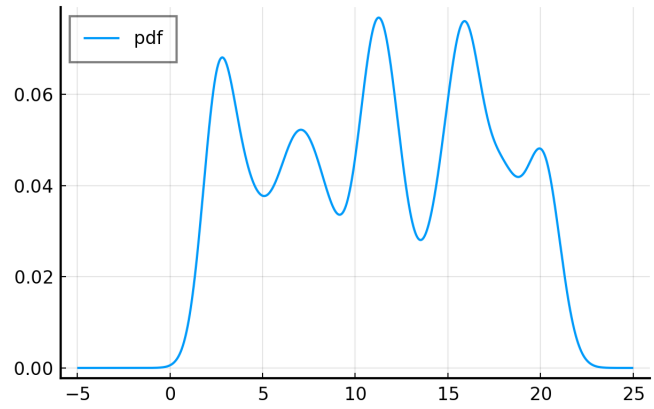
もしも, x にも a にもよらない定数 c が存在して,

$$p(x|a) = \exp(f(x-a) - c)$$

の形の確率密度函数が存在すれば, 以上の手続きは確率モデル $p(x|a)$ に関する最尤法に一致する. しかし, $f(x)$ は $x=0$ から離れるとほぼ定数函数になると仮定していたので, このような函数を $x=-\infty$ から ∞ まで積分すると値は必ず ∞ になる. だから, 積分範囲を十分大きな有限区間に制限してやらなければいけなくなる. この点に関してはすっきり行かない.

4.3 最頻値の近似的な推定の例

In [10]:
```
Random.seed!(37346491)

r = 10
noise1 = 0.4randn(r)
noise2 = 0.2randn(r)
dist = MixtureModel([Normal(2k+noise1[k], 1.0+noise2[k]) for k in 1:r], [1/10 for k in 1:r])
x = range(-5, 2r+5, length=2000)
plot(x, pdf.(dist, x), legend=:topleft, label="pdf") ▷ retina
```

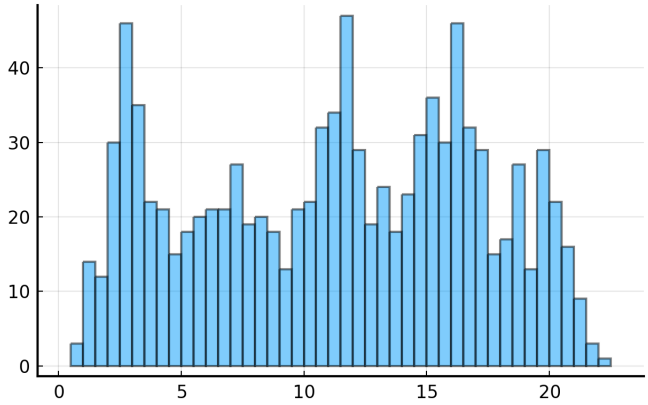


In [11]:
```
# mode of dist
a₀ = x[findmax(pdf.(dist, x))[2]]
```

Out[11]: 11.268134067033516

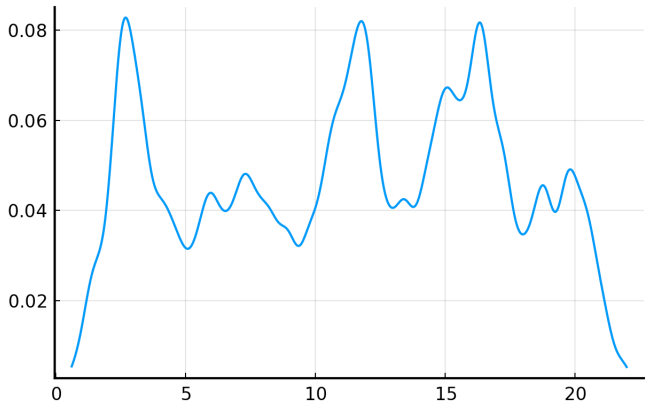
In [12]:
```
# generate sample X

X = rand(dist, 1000)
@show mean(X), std(X)
@show median(X), StatsBase.mad(X; normalize=false)
@show mode(X)
histogram(X; bin=40, legend=false, alpha=0.5) ▷ retina
```

```
(mean(X), std(X)) = (11.282125384708527, 5.71533297077284)
(median(X), StatsBase.mad(X; normalize = false)) = (11.657135380798387, 4.719954415479611)
```



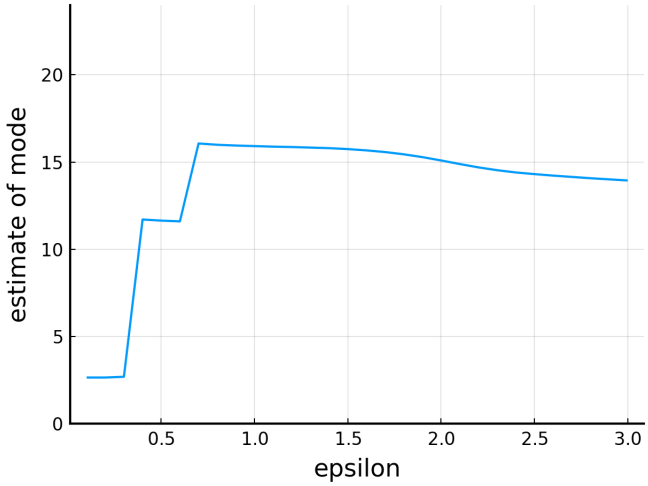

```
mode(X) = 2.692692489221092
```

In [13]:
```
f(x, ε) = pdf(Normal(0, ε), x)
L(a; ε=0.3) = mean(f(x-a, ε) for x in X)

a = range(extrema(X)..., length=2000)
plot(a, L.(a), legend=false) ▷ retina
a_hat = a[findmax(L.(a))[2]]
```



Out[13]: 2.6806144476313984

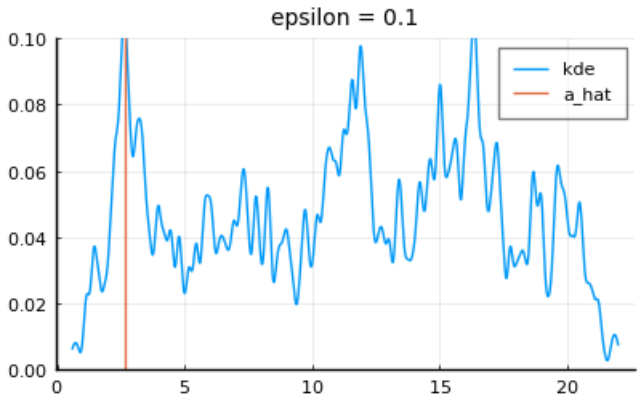
In [14]:
```
ε = range(0.1, 3.0, step=0.1)
y = [a[findmax(L.(a; ε=ε))[2]] for ε in ε]
plot(ε, y; size=(400, 300), legend=false, ylim=(0, 24), xlabel="epsilon", ylabel="estimate
of mode") ▷ retina
```



サンプルの分布を丸めるために使った正規分布の標準偏差を 0.1 から 2.0 まで動かして, 最頻値に推定値がどのように変化するかをプロットすると, 上のサンプルの例では, 0.1 から 1.0 までは最頻値の推定値が 11 程度になっているが、それより大きくなると突然 16 程度に最頻値の推定値がジャンプする.

In [15]:
```
a = range(extrema(X)..., length=2000)
@time anim = @animate for ε in range(0.1, 3.0, step=0.1)
    a_hat = a[findmax(L.(a; ε=ε))[2]]
    plot(a, L.(a; ε=ε), legend=:topright, label="kde", ylim=(0.0, 0.1))
    vline!([a_hat], label="a_hat")
    title!("epsilon = $ε")
end
pyplotclf()
gif(anim, "images/mode_estimation.gif", fps=5)
displayfile("image/gif", "images/mode_estimation.gif")
```

```
9.437591 seconds (4.84 M allocations: 257.688 MiB, 0.98% gc time, 13.46% compilation time)
```




```
┌ Info: Saved animation to
│   fn = C:\Users\genkuroki\OneDrive\work\Statistics\images\mode_estimation.gif
└ @ Plots C:\Users\genkuroki\.julia\packages\Plots\5ItHH\src\animation.jl:104
```

5 加法平均と乗法平均

正の実数上の確率分布 $q(x)$ を考える. $x \leqq 0$ のとき $q(x) = 0$ と仮定しておく.

分布 $q(x)$ の平均と対数平均をそれぞれ $\mu_0, \log\lambda_0$ と書いておく:

$$\mu_0 = \int q(x)x\,dx, \quad \log\lambda_0 = \int q(x)\log x\,dx.$$

Jensenの不等式より, $\log\mu_0 \geqq \log\lambda_0$. λ_0 は分布 $q(x)$ の「相乗平均」とでも呼べそうな量になっている.

5.1 ガンマ分布モデルの汎化誤差

パラメーター $\alpha, \theta > 0$ のガンマ分布の確率密度函数は

$$p(x) = p(x|\alpha,\theta) = \frac{1}{\Gamma(\alpha)\theta^\alpha}e^{-x/\theta}x^{\alpha-1}$$

と書ける. $x \leqq 0$ のとき $p(x|\alpha,\theta) = 0$ と仮定しておく. このとき,

$$-\log p(x|\alpha,\theta) = \log\Gamma(\alpha) + \alpha\log\theta + \frac{x}{\theta} - (\alpha-1)\log x$$

なので, 汎化誤差 $G(q|p)$ は次のように書ける:

$$G(q||p) = \log\Gamma(\alpha) + \alpha\log\theta + \frac{\mu_0}{\theta} - (\alpha-1)\log\lambda_0.$$

これを最小にする α, θ がどうなるかについて考えよう. そのために $\mu = \alpha\theta$ とおく. (μ はガンマ分布の平均に等しい.) このとき,

$$\begin{aligned} G(q||p) &= \log\Gamma(\alpha) + \alpha\log\frac{\mu}{\alpha} + \alpha\frac{\mu_0}{\mu} - (\alpha-1)\log\lambda_0 \\ &= \log\Gamma(\alpha) - \alpha\log\alpha - (\alpha-1)\log\lambda_0 + \alpha\left(\log\mu + \frac{\mu_0}{\mu}\right). \end{aligned}$$

ゆえに α を固定するとき, これを最小化する μ は $\mu = \mu_0$ となることがわかる. $\mu = \mu_0$ のとき, 汎化誤差の α による偏導函数は次のようになる:

$$\left.\frac{\partial}{\partial\alpha}\right|_{\mu=\mu_0} G(q||p) = \psi(\alpha) - \log\alpha - \log\lambda_0 + \log\mu_0.$$

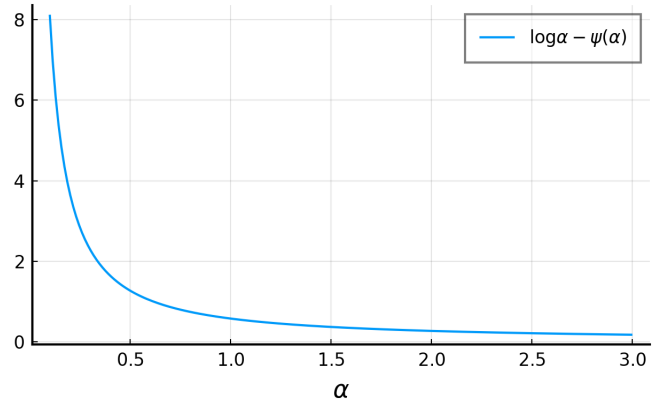
ここで, $\psi(\alpha) = \Gamma'(\alpha)/\Gamma(\alpha)$ はdigamma函数と呼ばれている.

函数 $\log\alpha - \psi(\alpha)$ は $\alpha > 0$ の狭義単調減少函数で像が正の実数全体になるので(下の方のプロットを参照せよ), 上の偏導函数が 0 になるという条件

$$\log\alpha - \psi(\alpha) = \log\mu_0 - \log\lambda_0$$

を満たす α が一意に存在して(それを α_0 と書く), $\mu = \mu_0$ に制限された汎化誤差は $\alpha = \alpha_0$ で最小になることがわかる.

In [16]:
```
# digamma函数に関するプロット

f(α) = log(α) - digamma(α)
α = range(0.1, 3, length=200)
plot(α, f.(α), xlabel="\$\\alpha\$", label="\$\\log\\alpha - \\psi(\\alpha)\$") ▷ retina
```



5.2 ガンマ分布モデルの最尤推定

$X_1, X_2, \ldots, X_n > 0$ は分布 $q(x)$ のサンプルであると仮定し,

$$\overline{X} = \frac{1}{n}\sum_{k=1}^{n} X_k, \quad \Lambda(X) = (X_1 X_2 \cdots X_n)^{1/n}$$

とおく. $\Lambda(X)$ の定義より,

$$\log \Lambda(X) = \frac{1}{n}\sum_{k=1}^{n} \log X_k.$$

$\mu = \alpha\theta$ とおくと, 上のサンプルに関するガンマ分布モデルの大数尤度函数の $-1/n$ 倍は前節と同様の計算によって次のように書かれる:

$$\frac{1}{n}\sum_{k=1}^{n} \log p(X_k|\alpha,\mu) = \log\Gamma(\alpha) - \alpha\log\alpha - (\alpha-1)\log\Lambda(X) + \alpha\left(\log\mu + \frac{\overline{X}}{\mu}\right).$$

ゆえに前節と同様にして, これを最小にする α, θ は $\mu = \alpha\theta = \overline{X}$ と条件

$$\log\alpha - \psi(\alpha) = \log\overline{X} - \log\Lambda(X)$$

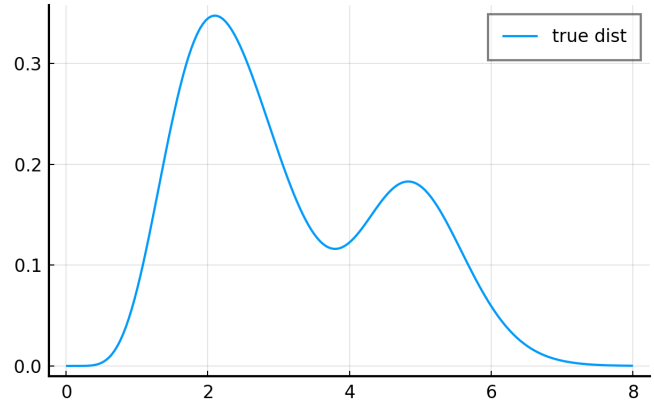
という条件で一意に決まる.

以上の事実は, ガンマ分布による最尤推定は本質的にサンプルの相加平均 $\overline{X}$ と相乗平均 $\Lambda(X)$ を求めることに等しいことを意味している.

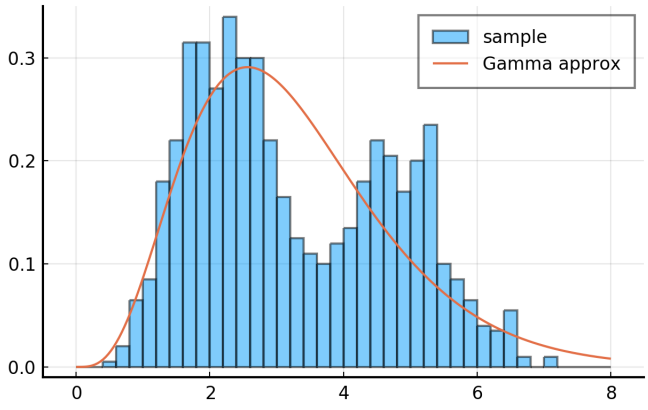
このように高校数学の花形の一つである相加・相乗平均の話題はガンマ分布の最尤法の形で復活する!

5.3 ガンマ分布モデルによる推定の例

In [17]:
```
dist_mg = MixtureModel([Gamma(8,0.3), Gamma(50, 0.1)], [0.7, 0.3])
x = range(0, 8, length=200)
plot(x, pdf.(dist_mg, x), label="true dist") ▷ retina
```



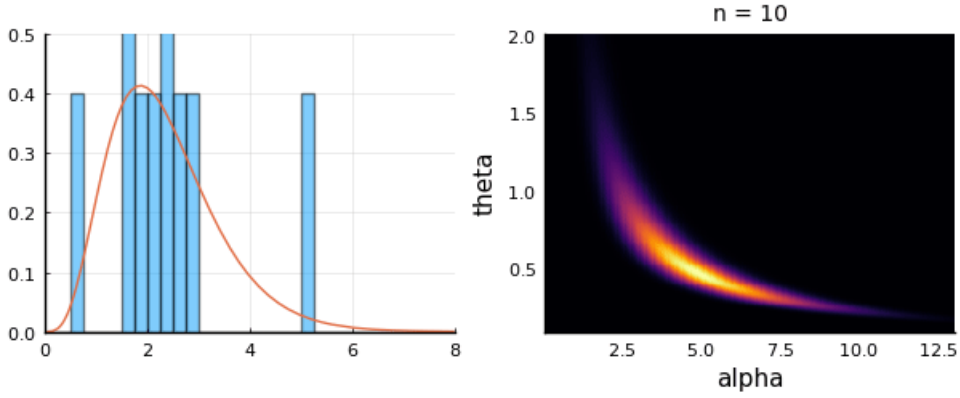
In [18]:
```
Random.seed!(37346491)

X = rand(dist_mg, 1000)
@show dist_g = fit_mle(Gamma, X)
histogram(X, norm=true, bins=range(0, 8, step=0.2), label="sample", alpha=0.5)
plot!(x, pdf.(dist_g, x), label="Gamma approx") ▷ retina
```




```
dist_g = fit_mle(Gamma, X) = Gamma{Float64}(α=4.65981716100742, θ=0.7006271805694427)
```

In [19]:
```
# 尤度函数のプロット

Random.seed!(37346491)

N = 300
XX = rand(dist_mg, N)

ns = [fill(10, 10); 10:3:300; 300; fill(300, 10)]
@time anim = @animate for n in ns
    X = @view XX[1:n]

    dist_g = fit_mle(Gamma, X)
    P1 = histogram(X; normed=true, bins=range(0, 8, step=0.25), alpha=0.5, label="sample")
    plot!(x -> pdf(dist_g, x), 0, 8; label="Gamma approx")
    plot!(legend=false, xlim=(0, 8), ylim=(0, 0.5))

    loglik(α, θ) = sum(logpdf(Gamma(α, θ), x) for x in X)
    α = range(0.1, 13, length=200)
    θ = range(0.1, 2.0, length=200)
    P2 = heatmap(α, θ, exp.(loglik.(α', θ)); xlabel="alpha", ylabel="theta",
        size=(300, 320), colorbar=false, title="n = $n")

    plot(P1, P2, size=(600, 250))
end
pyplotclf()
gif(anim, "images/lik_gamma.gif", fps=10)
displayfile("image/gif", "images/lik_gamma.gif")
```

```
240.745882 seconds (26.78 M allocations: 986.861 MiB, 0.09% gc time, 0.28% compilation time)
```

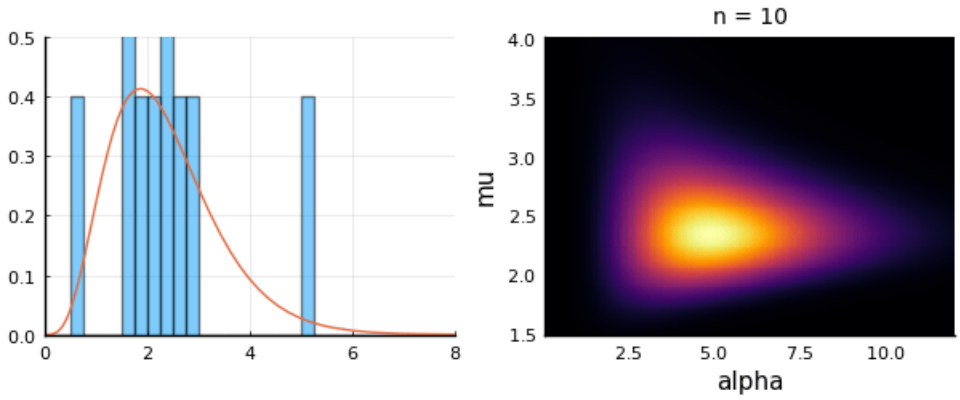



```
┌ Info: Saved animation to
│   fn = C:\Users\genkuroki\OneDrive\work\Statistics\images\lik_gamma.gif
└ @ Plots C:\Users\genkuroki\.julia\packages\Plots\5ItHH\src\animation.jl:104
```

In [20]:

```
# 尤度函数のプロット

Random.seed!(37346491)

N = 300
XX = rand(dist_mg, N)

ns = [fill(10, 10); 10:3:300; 300; fill(300, 10)]
@time anim = @animate for n in ns
    X = @view XX[1:n]

    dist_g = fit_mle(Gamma, X)
    P1 = histogram(X; normed=true, bins=range(0, 8, step=0.25), alpha=0.5, label="sample")
    plot!(x -> pdf(dist_g, x), 0, 8; label="Gamma approx")
    plot!(legend=false, xlim=(0, 8), ylim=(0, 0.5))

    loglik(α, μ) = sum(logpdf(Gamma(α, μ/α), x) for x in X)
    α = range(0.1, 12, length=200)
    μ = range(1.5, 4, length=200)
    P2 = heatmap(α, μ, exp.(loglik.(α', μ)); xlabel="alpha", ylabel="mu",
        size=(300, 320), colorbar=false, title="n = $n")

    plot(P1, P2, size=(600, 250))
end
pyplotclf()
gif(anim, "images/lik_gamma2.gif", fps=10)
displayfile("image/gif", "images/lik_gamma2.gif")
```

```
245.309444 seconds (26.77 M allocations: 986.960 MiB, 0.08% gc time, 0.27% compilation time)
```

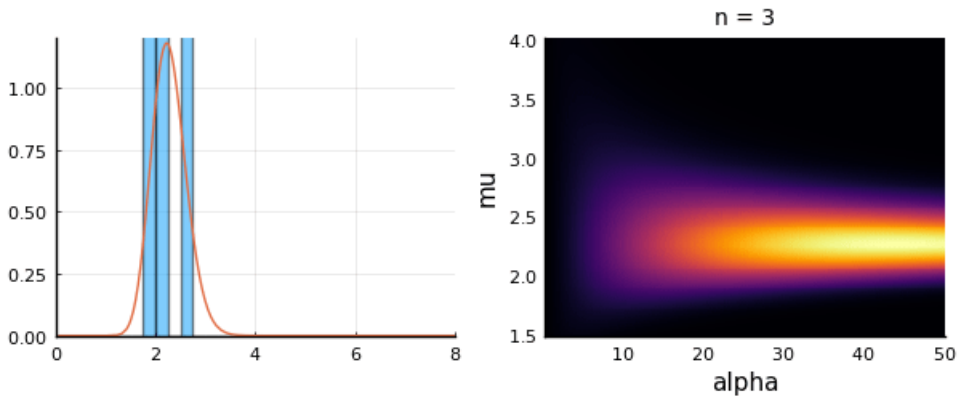



```
┌ Info: Saved animation to
│   fn = C:\Users\genkuroki\OneDrive\work\Statistics\images\lik_gamma2.gif
└ @ Plots C:\Users\genkuroki\.julia\packages\Plots\5ItHH\src\animation.jl:104
```

In [21]:

```
# 尤度函数のプロット

Random.seed!(37346491)

N = 300
XX = rand(dist_mg, N)

ns = [fill(3, 6); 4;4;4;4;5;5;5;6;6;6;7;7;8;8;9;9; 10:20; 23:3:100; 110:10:300 ; fill(300, 5)]
@time anim = @animate for n in ns
    X = @view XX[1:n]

    dist_g = fit_mle(Gamma, X)
    P1 = histogram(X; normed=true, bins=range(0, 8, step=0.25), alpha=0.5, label="sample")
    plot!(x -> pdf(dist_g, x), 0, 8; label="Gamma approx")
    plot!(legend=false, xlim=(0, 8), ylim=(0, 1.2))

    loglik(α, μ) = sum(logpdf(Gamma(α, μ/α), x) for x in X)
    α = range(0.5, 50, length=200)
    μ = range(1.5, 4, length=200)
    P2 = heatmap(α, μ, exp.(loglik.(α', μ)); xlabel="alpha", ylabel="mu",
        size=(300, 320), colorbar=false, title="n = $n")

    plot(P1, P2, size=(600, 250))
end
pyplotclf()
gif(anim, "images/lik_gamma3.gif", fps=3)
displayfile("image/gif", "images/lik_gamma3.gif")
```

```
83.969869 seconds (19.35 M allocations: 717.699 MiB, 0.18% gc time, 0.74% compilation time)
```




```
┌ Info: Saved animation to
│   fn = C:\Users\genkuroki\OneDrive\work\Statistics\images\lik_gamma3.gif
└ @ Plots C:\Users\genkuroki\.julia\packages\Plots\5ItHH\src\animation.jl:104
```

In [ ]:

```
1
```