Identifying genes associated with prostate cancer

Load packages and dataset

```
begin
    using DataFrames
    using Empirikos
    using Plots
    using PGFPlotsX
    using LaTeXStrings
    using MosekTools
    using JuMP
    using Setfield
    using Random
end
```

```
begin
    pgfplotsx()
    empty!(PGFPlotsX.CUSTOM_PREAMBLE)
    push!(PGFPlotsX.CUSTOM_PREAMBLE, raw"\usepackage{amssymb}")
    push!(PGFPlotsX.CUSTOM_PREAMBLE, raw"\newcommand{\PP}[2][]
{\mathbb{P}_{#1}\left[#2\right]}")
    push!(PGFPlotsX.CUSTOM_PREAMBLE, raw"\newcommand{\EE}[2][]
{\mathbb{E}_{#1}\left[#2\right]}")
end;
```

```
theme(:default;
      background_color_legend = :transparent,
      foreground_color_legend = :transparent,
      grid=nothing,
      frame=:box,
      legendfonthalign = :left,
      thickness_scaling=1.3)
```

```
Zs = Prostate.ebayes_samples();
```

Marginal distribution of z-scores

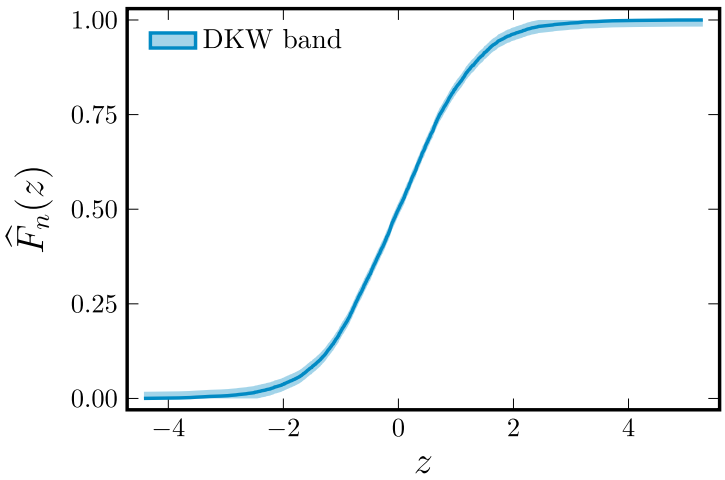
DKW-F-Localization

```
dkw_floc =   DvoretzkyKieferWolfowitz(0.05, 1000)
```

```
dkw_floc = DvoretzkyKieferWolfowitz(0.05)
```

```
fitted_dkw =
  FittedDvoretzkyKieferWolfowitz(SortedDict(Z= -4.4239 | σ=1.0   ⇒ 0.000165755, Z= -4.3343
```

```
fitted_dkw = fit(dkw_floc, Zs)
```




```
dkw_plot =
```

```
dkw_plot = plot(fitted_dkw, label="DKW band",
     xlab=L"z", ylab=L"\widehat{F}_n(z)",  size=(380,280))
```

```
# savefig(dkw_plot, "prostate_dkw_band.tikz")
```

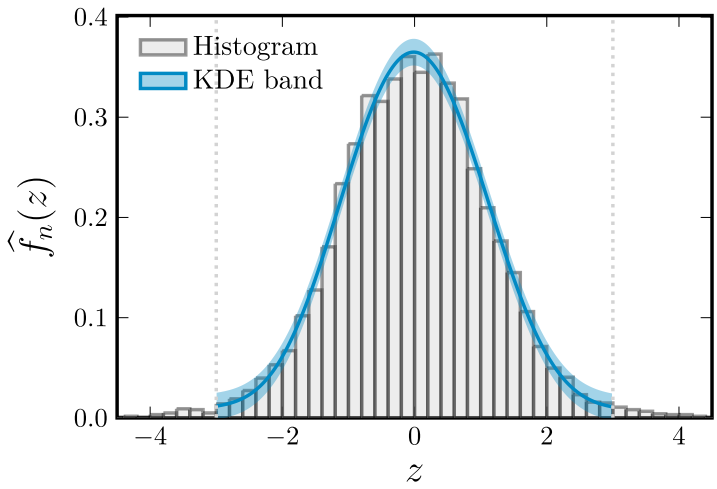
KDE-F-Localization

```
infty_floc =
  InfinityNormDensityBand(-3.0, 3.0, 1024, FlatTopKernel | bandwidth = DataBasedDefault(),
```

```
infty_floc = Empirikos.InfinityNormDensityBand(;a_min=-3.0,a_max=3.0,α=0.05)
```

```
fitted_infty_floc =
  FittedInfinityNormDensityBand(0.013304119753475209, -3.0, 3.0, UnivariateKDE(-6.457524041
```

```
fitted_infty_floc = fit(infty_floc, Zs)
```




```
prostate_kde_plot =
```

```
prostate_kde_plot = begin
    prostate_marginal_plot = histogram([response(Z) for Z in Zs],
          bins=50, normalize=true,
          label="Histogram", fillalpha=0.4, linealpha=0.4, fillcolor=:lightgray,
          size=(380,280), xlims=(-4.5,4.5))
    plot!(prostate_marginal_plot, fitted_infty_floc,
          label="KDE band", xlims=(-4.5,4.5),
          yguide=L"\widehat{f}_n(z)", xguide=L"z")
    plot!(prostate_marginal_plot, [-3.0;3.0], seriestype=:vline,
          linestyle=:dot, label=nothing, color=:lightgrey)
end
```

```
# savefig("prostate_kde_band.tikz")
```

Confidence intervals

```
quiet_mosek =
  OptimizerWithAttributes(Optimizer (generic function with 2 methods), [RawParameter("QUIET
```

```
quiet_mosek = optimizer_with_attributes(Mosek.Optimizer, "QUIET" => true)
```

```
gcal_locmix = MixturePriorClass (K = 121)
              Distributions.Normal{Float64}(μ=-3.0, σ=0.25)
              Distributions.Normal{Float64}(μ=-2.95, σ=0.25)
              Distributions.Normal{Float64}(μ=-2.9, σ=0.25)
              Distributions.Normal{Float64}(μ=-2.85, σ=0.25)
              Distributions.Normal{Float64}(μ=-2.8, σ=0.25)
              Distributions.Normal{Float64}(μ=-2.75, σ=0.25)
              Distributions.Normal{Float64}(μ=-2.7, σ=0.25)
              Distributions.Normal{Float64}(μ=-2.65, σ=0.25)
              The rest are omitted ...
```

```
gcal_locmix = MixturePriorClass(Normal.(-3:0.05:3, 0.25))
```

```
gcal_scalemix =
GaussianScaleMixtureClass | σs = [0.1, 0.11, 0.121, 0.1331, 0.14641, 0.161051, 0.177156, 0
```

```
gcal_scalemix = Empirikos.set_defaults(GaussianScaleMixtureClass(), Zs; hints =
Dict(:grid_scaling => 1.1))

```

```
discr =
 Discretizer([( .. -3.0], (-3.0 .. -2.995], (-2.995 .. -2.99], (-2.99 .. -2.985], (-2.985
```

```
discr = interval_discretizer(-3.0:0.005:3.0)
```

```
Intervals.Interval{Float64, Intervals.Unbounded, Intervals.Closed}(nothing, -3.0)
```

```
discr.sorted_intervals[1]
```

```
floc_method_dkw_locmix =
EB intervals with F-Localization: DvoretzkyKieferWolfowitz{Float64, Int64}(0.05, 1000)
                 𝒢: MixturePriorClass (K = 121)
Distributions.Normal{Float64}(μ=-3.0, σ=0.25)
Distributions.Normal{Float64}(μ=-2.95, σ=0.25)
Distributions.Normal{Float64}(μ=-2.9, σ=0.25)
Distributions.Normal{Float64}(μ=-2.85, σ=0.25)
Distributions.Normal{Float64}(μ=-2.8, σ=0.25)
Distributions.Normal{Float64}(μ=-2.75, σ=0.25)
Distributions.Normal{Float64}(μ=-2.7, σ=0.25)
Distributions.Normal{Float64}(μ=-2.65, σ=0.25)
The rest are omitted ...
```

```
floc_method_dkw_locmix = FLocalizationInterval(flocalization = dkw_floc,
                                convexclass = gcal_locmix, solver = quiet_mosek)

```

```
floc_method_kde_locmix =
EB intervals with F-Localization: InfinityNormDensityBand(-3.0, 3.0, 1024, FlatTopKernel |
                 𝒢: MixturePriorClass (K = 121)
Distributions.Normal{Float64}(μ=-3.0, σ=0.25)
Distributions.Normal{Float64}(μ=-2.95, σ=0.25)
Distributions.Normal{Float64}(μ=-2.9, σ=0.25)
Distributions.Normal{Float64}(μ=-2.85, σ=0.25)
Distributions.Normal{Float64}(μ=-2.8, σ=0.25)
Distributions.Normal{Float64}(μ=-2.75, σ=0.25)
Distributions.Normal{Float64}(μ=-2.7, σ=0.25)
Distributions.Normal{Float64}(μ=-2.65, σ=0.25)
The rest are omitted ...
```

```
floc_method_kde_locmix = FLocalizationInterval(flocalization = infty_floc,
                              convexclass = gcal_locmix, solver = quiet_mosek)

```

```
amari_kde_locmix =
AMARI with  F-Localization: InfinityNormDensityBand(-3.0, 3.0, 1024, FlatTopKernel | bandw
           𝒢: MixturePriorClass (K = 121)
Distributions.Normal{Float64}(μ=-3.0, σ=0.25)
Distributions.Normal{Float64}(μ=-2.95, σ=0.25)
Distributions.Normal{Float64}(μ=-2.9, σ=0.25)
Distributions.Normal{Float64}(μ=-2.85, σ=0.25)
Distributions.Normal{Float64}(μ=-2.8, σ=0.25)
Distributions.Normal{Float64}(μ=-2.75, σ=0.25)
Distributions.Normal{Float64}(μ=-2.7, σ=0.25)
Distributions.Normal{Float64}(μ=-2.65, σ=0.25)
The rest are omitted ...
```

```

amari_kde_locmix = Empirikos.AMARI(
                        convexclass = gcal_locmix,
                        flocalization = (@set infty_floc.α=0.01),
                        discretizer=discr,
                        solver=quiet_mosek, )


```

```
floc_method_dkw_scalemix =
EB intervals with F-Localization: DvoretzkyKieferWolfowitz{Float64, Int64}(0.05, 1000)
                 𝒢: GaussianScaleMixtureClass | σs = [0.1, 0.11, 0.121, 0.1331, 0.14641, (
```

```
floc_method_dkw_scalemix = FLocalizationInterval(flocalization = dkw_floc,
                              convexclass = gcal_scalemix, solver = quiet_mosek)

```

```
floc_method_kde_scalemix =
EB intervals with F-Localization: InfinityNormDensityBand(-3.0, 3.0, 1024, FlatTopKernel |
                 𝒢: GaussianScaleMixtureClass | σs = [0.1, 0.11, 0.121, 0.1331, 0.14641, (
```

```
floc_method_kde_scalemix = FLocalizationInterval(flocalization = infty_floc,
                              convexclass = gcal_scalemix, solver = quiet_mosek)

```

```
amari_kde_scalemix =
AMARI with  F-Localization: InfinityNormDensityBand(-3.0, 3.0, 1024, FlatTopKernel | bandw
            𝒢: GaussianScaleMixtureClass | σs = [0.1, 0.11, 0.121, 0.1331, 0.14641, 0.1610
```

```
amari_kde_scalemix = Empirikos.AMARI(
                        convexclass = gcal_scalemix,
                        flocalization = (@set infty_floc.α=0.01),
                        discretizer=discr,
                        solver=quiet_mosek, )
```

```
ts = -3.0:0.2:3.0
```

```
ts= -3:0.2:3
```

```
postmean_targets =
  [PosteriorMean(Z=    -3.0 | σ=1.0  ), PosteriorMean(Z=    -2.8 | σ=1.0  ), PosteriorMean(
```

```
postmean_targets = Empirikos.PosteriorMean.(StandardNormalSample.(ts))
```

```
lfsrs =
  [PosteriorProbability(Z=    -3.0 | σ=1.0  , [0 .. )), PosteriorProbability(Z=    -2.8 | σ
```

```
lfsrs = Empirikos.PosteriorProbability.(StandardNormalSample.(ts),
                                        Interval(0,nothing))

```

```
postmean_ci_dkw_locmix =
  [lower = -2.378, upper = -0.1786, α = 0.05  (PosteriorMean{StandardNormalSample{Float64}}
```

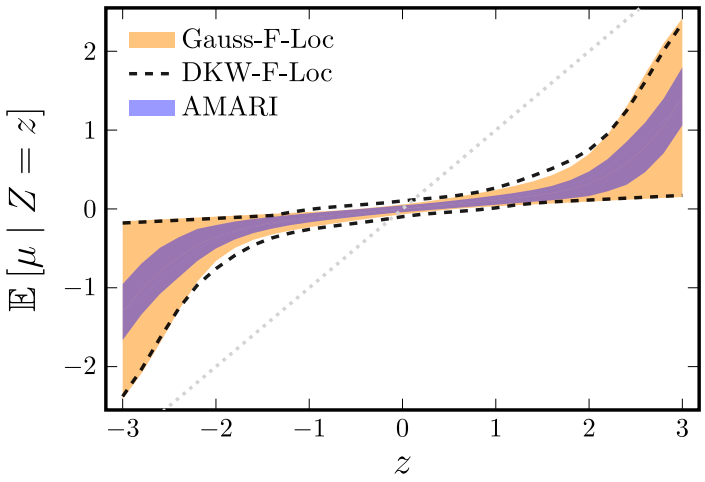
```
postmean_ci_dkw_locmix = confint.(floc_method_dkw_locmix, postmean_targets, Zs)
```

```
postmean_ci_kde_locmix =
  [lower = -2.411, upper = -0.1569, α = 0.05  (PosteriorMean{StandardNormalSample{Float64}}
```

```
postmean_ci_kde_locmix = confint.(floc_method_kde_locmix, postmean_targets, Zs)
```

```
postmean_ci_amari_locmix =
  [lower = -1.663, upper = -0.9528, α = 0.05  (PosteriorMean{StandardNormalSample{Float64}}
```

```
postmean_ci_amari_locmix = confint.(amari_kde_locmix, postmean_targets, Zs)
```



postmean_locmix_plot =

```
postmean_locmix_plot = begin
    postmean_locmix_plot = plot(ts, postmean_ci_kde_locmix, label="Gauss-F-Loc",
        fillcolor=:darkorange, fillalpha=0.5, ylim=(-2.55,2.55),
        xguide = L"z", yguide=L"\EE{\mu \mid Z=z}",
        size=(380,280))
    plot!(postmean_locmix_plot, ts, postmean_ci_dkw_locmix, label="DKW-F-Loc",
        show_ribbon=false, alpha=0.9, color=:black)
    plot!(postmean_locmix_plot, ts, postmean_ci_amari_locmix, label="AMARI",
        show_ribbon=true, fillcolor=:blue, fillalpha=0.4)
    plot!(postmean_locmix_plot, [-3.0;3.0], [-3.0; 3.0], seriestype=:line,
        linestyle=:dot, label=nothing, color=:lightgrey)
end

```

```
# savefig(postmean_locmix_plot, "prostate_locmix_postmean.tikz")
```

postmean_ci_dkw_scalemix =

```
[lower = -1.937, upper = -0.1928, α = 0.05  (PosteriorMean{StandardNormalSample{Float64}}
```

```
postmean_ci_dkw_scalemix = confint.(floc_method_dkw_scalemix, postmean_targets, Zs)
```

postmean_ci_kde_scalemix =

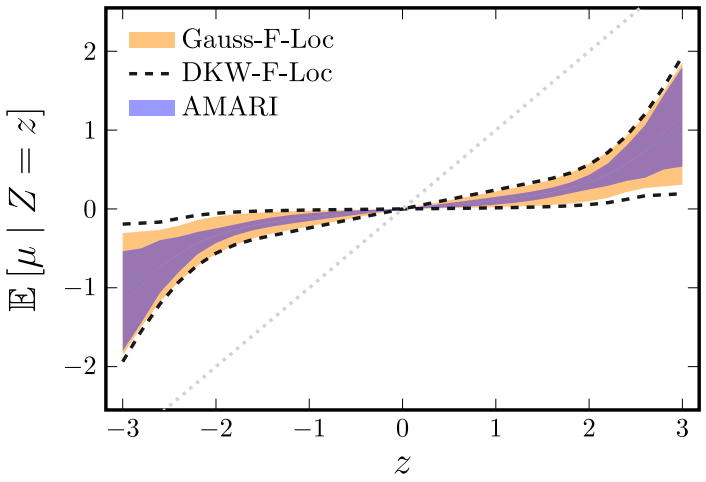
```
[lower = -1.857, upper = -0.3067, α = 0.05  (PosteriorMean{StandardNormalSample{Float64}}
```

```
postmean_ci_kde_scalemix = confint.(floc_method_kde_scalemix, postmean_targets, Zs)
```

postmean_ci_amari_scalemix =

```
[lower = -1.801, upper = -0.5366, α = 0.05  (PosteriorMean{StandardNormalSample{Float64}}
```

```
postmean_ci_amari_scalemix = confint.(amari_kde_scalemix, postmean_targets, Zs)
```



`postmean_scalemix_plot =`

```
postmean_scalemix_plot = begin
    postmean_scalemix_plot = plot(ts, postmean_ci_kde_scalemix,
        label="Gauss-F-Loc", fillcolor=:darkorange, fillalpha=0.5, ylim=
(-2.55,2.55),
        xguide = L"z",
        yguide=L"\EE{\mu \mid Z=z}",
        size=(380,280))
    plot!(postmean_scalemix_plot, ts, postmean_ci_dkw_scalemix,
        label="DKW-F-Loc",show_ribbon=false, alpha=0.9, color=:black)
    plot!(postmean_scalemix_plot, ts, postmean_ci_amari_scalemix,
        label="AMARI",show_ribbon=true, fillcolor=:blue, fillalpha=0.4)
    plot!(postmean_scalemix_plot, [-3.0;3.0], [-3.0; 3.0], seriestype=:line,
        linestyle=:dot, label=nothing, color=:lightgrey)
end
```

```
# savefig(postmean_scalemix_plot, "prostate_scalemix_postmean.tikz")
```

`lfsr_ci_dkw_locmix =`

```
[lower = 0.03489, upper = 0.231, α = 0.05  (PosteriorProbability{StandardNormalSample{Flo
```

```
lfsr_ci_dkw_locmix = confint.(floc_method_dkw_locmix, lfsrs, Zs)
```

`lfsr_ci_kde_locmix =`

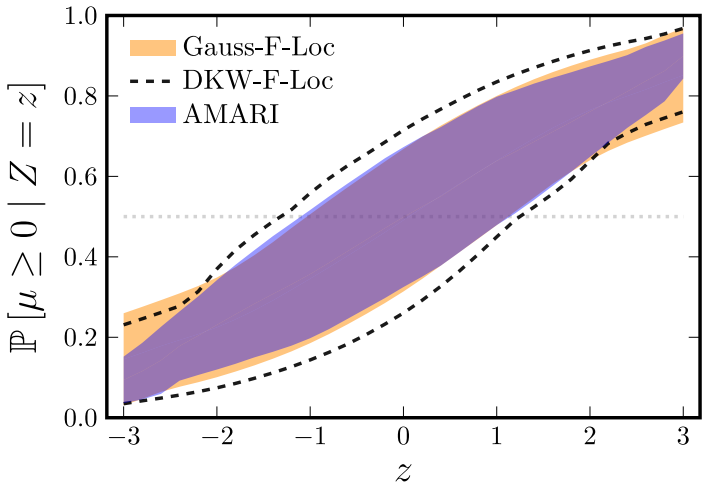
```
[lower = 0.03077, upper = 0.2597, α = 0.05  (PosteriorProbability{StandardNormalSample{Fl
```

```
lfsr_ci_kde_locmix = confint.(floc_method_kde_locmix, lfsrs, Zs)
```

`lfsr_ci_amari_locmix =`

```
[lower = 0.03588, upper = 0.1519, α = 0.05  (PosteriorProbability{StandardNormalSample{Fl
```

```
lfsr_ci_amari_locmix = confint.(amari_kde_locmix, lfsrs, Zs)
```



lfsr_locmix_plot =

```
lfsr_locmix_plot = begin
    lfsr_locmix_plot = plot([-3;3], [0.5; 0.5], seriestype=:line,
        linestyle=:dot, label=nothing, color=:lightgrey,
        xguide = L"z",
        yguide=L"\PP{\mu \geq 0 \mid Z=z}", size=(380,280))
    plot!(lfsr_locmix_plot, ts, lfsr_ci_kde_locmix,
        label="Gauss-F-Loc", fillcolor=:darkorange, fillalpha=0.5, ylim=(0,1))
    plot!(lfsr_locmix_plot, ts, lfsr_ci_dkw_locmix,
        label="DKW-F-Loc",show_ribbon=false, alpha=0.9, color=:black)
    plot!(lfsr_locmix_plot, ts, lfsr_ci_amari_locmix,
        label="AMARI",show_ribbon=true, fillcolor=:blue, fillalpha=0.4)
end
```

```
# savefig(lfsr_locmix_plot, "prostate_locmix_lfsr.tikz")
```

lfsr_ci_dkw_scalemix =

```
[lower = 0.0678, upper = 0.3162, α = 0.05  (PosteriorProbability{StandardNormalSample{Flo
```

```
lfsr_ci_dkw_scalemix = confint.(floc_method_dkw_scalemix, lfsrs, Zs)
```

lfsr_ci_kde_scalemix =

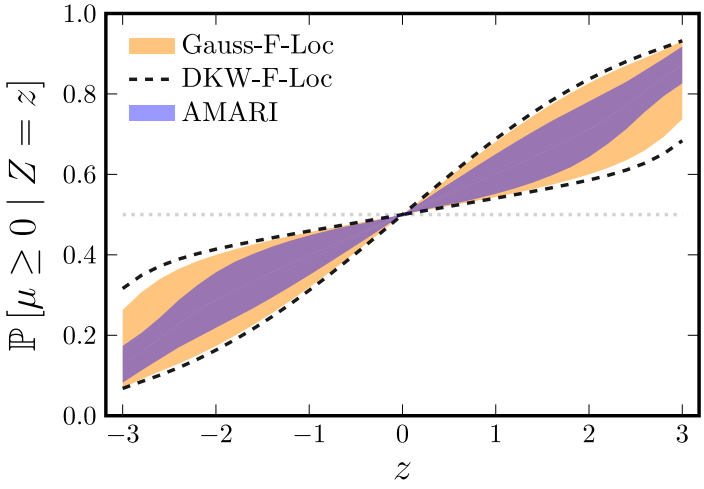
```
[lower = 0.06824, upper = 0.2629, α = 0.05  (PosteriorProbability{StandardNormalSample{Fl
```

```
lfsr_ci_kde_scalemix = confint.(floc_method_kde_scalemix, lfsrs, Zs)
```

lfsr_ci_amari_scalemix =

```
[lower = 0.08205, upper = 0.1736, α = 0.05  (PosteriorProbability{StandardNormalSample{Fl
```

```
lfsr_ci_amari_scalemix = confint.(amari_kde_scalemix, lfsrs, Zs)
```



lfsr_scalemix_plot =

```
lfsr_scalemix_plot = begin
    lfsr_scalemix_plot = plot([-3;3], [0.5; 0.5], seriestype=:line,
        linestyle=:dot, label=nothing, color=:lightgrey,
        xguide = L"z",
        yguide=L"\PP{\mu \geq 0 \mid Z=z}", size=(380,280))
    plot!(lfsr_scalemix_plot, ts, lfsr_ci_kde_scalemix,
        label="Gauss-F-Loc", fillcolor=:darkorange, fillalpha=0.5, ylim=(0,1))
    plot!(lfsr_scalemix_plot, ts, lfsr_ci_dkw_scalemix,
        label="DKW-F-Loc",show_ribbon=false, alpha=0.9, color=:black)
    plot!(lfsr_scalemix_plot, ts, lfsr_ci_amari_scalemix,
        label="AMARI",show_ribbon=true, fillcolor=:blue, fillalpha=0.4)
end
```

```
# savefig(lfsr_scalemix_plot, "prostate_scalemix_lfsr.tikz")
```

Sensitivity Analysis and goodness-of-fit

MersenneTwister(1)

```
Random.seed!(1)
```

n_half = 3017

```
n_half = ceil(Int64, nobs(Zs)/2)
```

```
train_idx = sample(1:nobs(Zs), n_half, replace=false);
```

```
test_idx = setdiff(1:nobs(Zs), train_idx);
```

```
Zs_train =
 [Z=   0.128 | σ=1.0  , Z= -1.8266 | σ=1.0  , Z= -0.5089 | σ=1.0  , Z= -1.7712 | σ=1.0  , Z
```

```
Zs_test =
 [Z=  1.4724 | σ=1.0  , Z=  3.5729 | σ=1.0  , Z=  0.9588 | σ=1.0  , Z=  1.0681 | σ=1.0  , Z
```

```
τs =  [0.02, 0.1, 0.25, 0.5, 0.55]
```

```
gcal_np = DiscretePriorClass | support = -4.0:0.005:4.0
```

```
npmle_fit = Fitted NPMLE with Mosek.Optimizer and 𝒢:
            DiscretePriorClass | support = -4.0:0.005:4.0
```

```
postmean_target =  PosteriorMean(Z=     2.0 | σ=1.0  )
```

```
lfsr_target =
 PosteriorProbability(Z=     2.0 | σ=1.0  , Intervals.Interval{Int64, Intervals.Closed, In
```

```
for (i, τ) in enumerate(τs)

    gcal_loc = MixturePriorClass(Normal.(-3:0.002:3, τ))

    npmle_loc_fit = fit(NPMLE(convexclass=gcal_loc, solver=Mosek.Optimizer),
                        Zs_test)

    e_value = exp(loglikelihood(Zs_test, npmle_fit.prior) -
                  loglikelihood(Zs_test, npmle_loc_fit.prior))

    floc_tmp = FLocalizationInterval(flocalization = fitted_infty_floc,
                convexclass = gcal_loc;
                solver = quiet_mosek)

    local floc_fit_lfsr
    local ci_lfsr
    local floc_fit_postmean
    local ci_postmean
    try
        floc_fit_lfsr = fit(floc_tmp, lfsr_target, Zs)
        ci_lfsr = confint(floc_fit_lfsr)

        floc_fit_postmean = fit(floc_tmp, postmean_target, Zs)
        ci_postmean = confint(floc_fit_postmean)
    catch
        floc_fit_lfsr = "error"
        ci_lfsr = "error"
        floc_fit_postmean = "error"
        ci_postmean = "error"
    end
    res_list[i] = (floc_fit_lfsr = floc_fit_lfsr, ci_lfsr = ci_lfsr,
                   floc_fit_postmean = floc_fit_postmean, ci_postmean = ci_postmean,
                   e_value=e_value)
end
```

	τ	SLR	Lfsr_CI	Postmean_CI
1	0.02	0.00302162	lower = 0.1859, upper = 0.9996, α = 0.(	lower = 0.01957, upper =
2	0.1	0.00315256	lower = 0.4491, upper = 0.9746, α = 0.(	lower = 0.03016, upper =
3	0.25	0.00415248	lower = 0.6396, upper = 0.8905, α = 0.(	lower = 0.09218, upper =
4	0.5	1.88465	lower = 0.8043, upper = 0.8325, α = 0.(	lower = 0.3834, upper = 0
5	0.55	74.1061	"error"	"error"

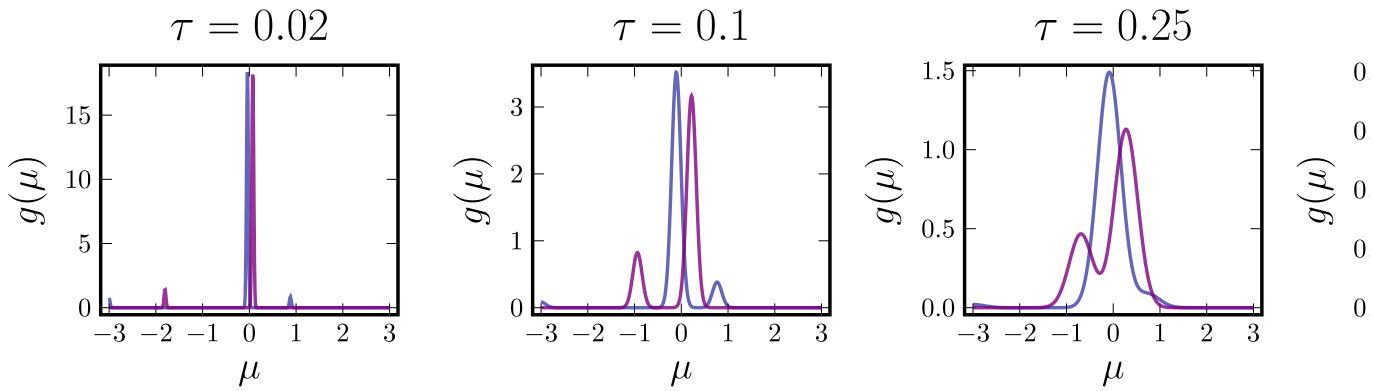
```
DataFrame(τ = τs,
          SLR = getproperty.(res_list, :e_value),
          Lfsr_CI = getproperty.(res_list, :ci_lfsr),
          Postmean_CI = getproperty.(res_list, :ci_postmean))
```

```
us = -3.0:0.01:3.0
```

```
us = -3:0.01:3
```

```
_plots = [plot(us,
    [pdf.(res_list[j].floc_fit_lfsr.g1, us) pdf.(res_list[j].floc_fit_lfsr.g2, us)],
    title=L"\tau = %$(τs[j])", color = [:darkblue :purple],
    linealpha = [0.6 0.8], xguide=L"\mu", yguide=L"g(\mu)")
    for j in 1:4];
```

```
worst_case_plots =
```




```
worst_case_plots = plot(_plots..., layout=(1,4),size=(900,200), legend=false)
```

```
# savefig(worst_case_plots, "worst_case_priors.tikz")
```

```
lower = 0.09292, upper = 0.6959, α = 0.05  (PosteriorMean{StandardNormalSample{Float64}}(Z=
```

```
postmean_ci_kde_locmix[findfirst(ts.==2.0)] #sanity check 1
```

```
lower = 0.6402, upper = 0.89, α = 0.05  (Empirikos.PosteriorProbability{StandardNormalSamp
```

```
lfsr_ci_kde_locmix[findfirst(ts.==2.0)] #sanity check 2
```