MONTE CARLO IN JULIA

James Doss-Gollin

CEVE 543

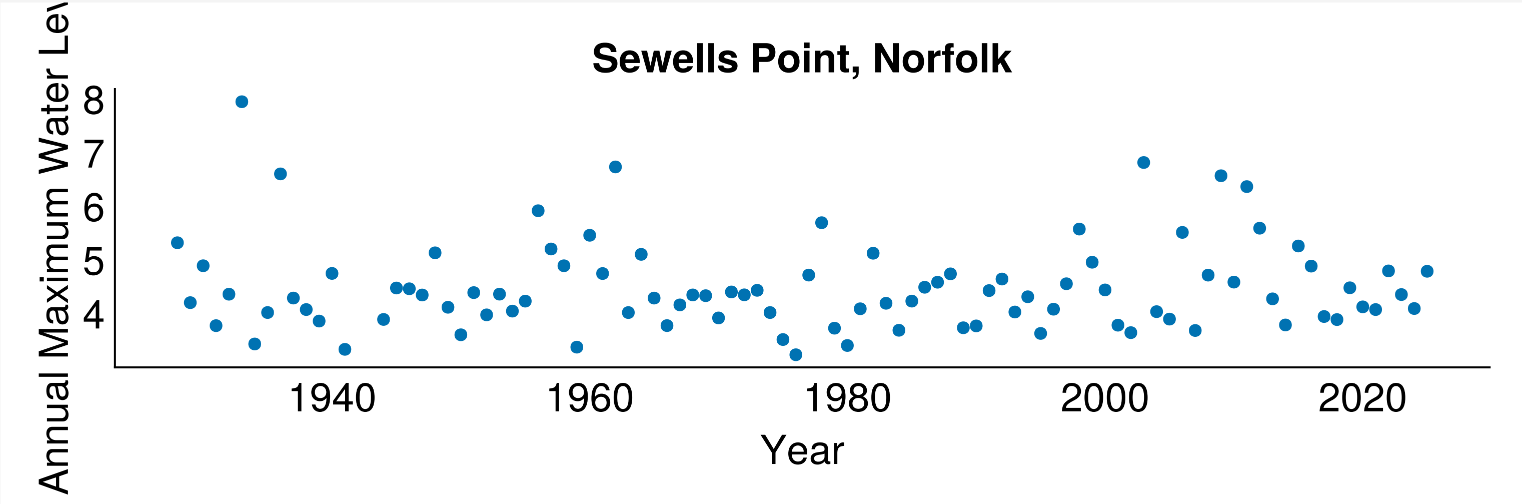
2026-08-28

Hazard

Risk

REALISTIC DATA

```julia
rng = MersenneTwister(824)
annmax = CSV.read("8638610-annmax.csv", DataFrame)
```

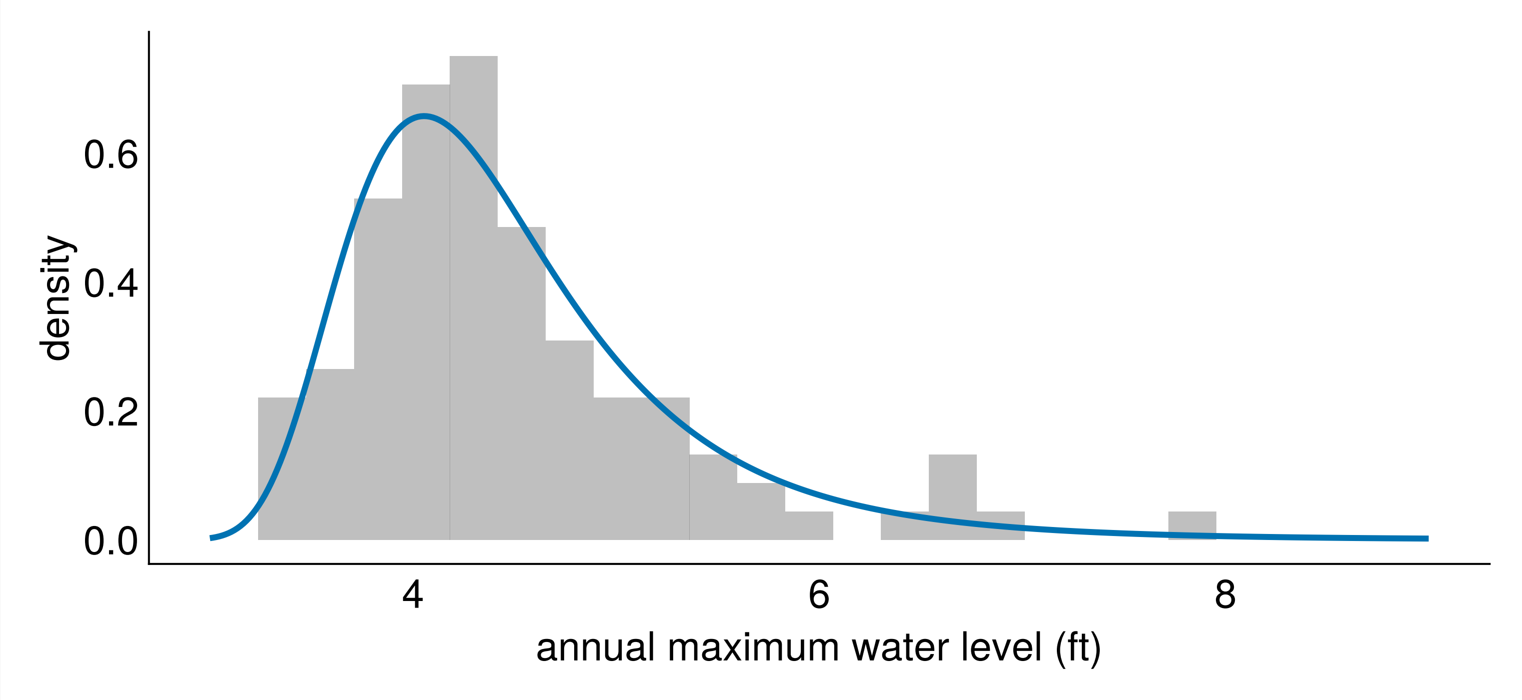


FITTING

We fit a Generalized Extreme Value distribution (more on that soon!)

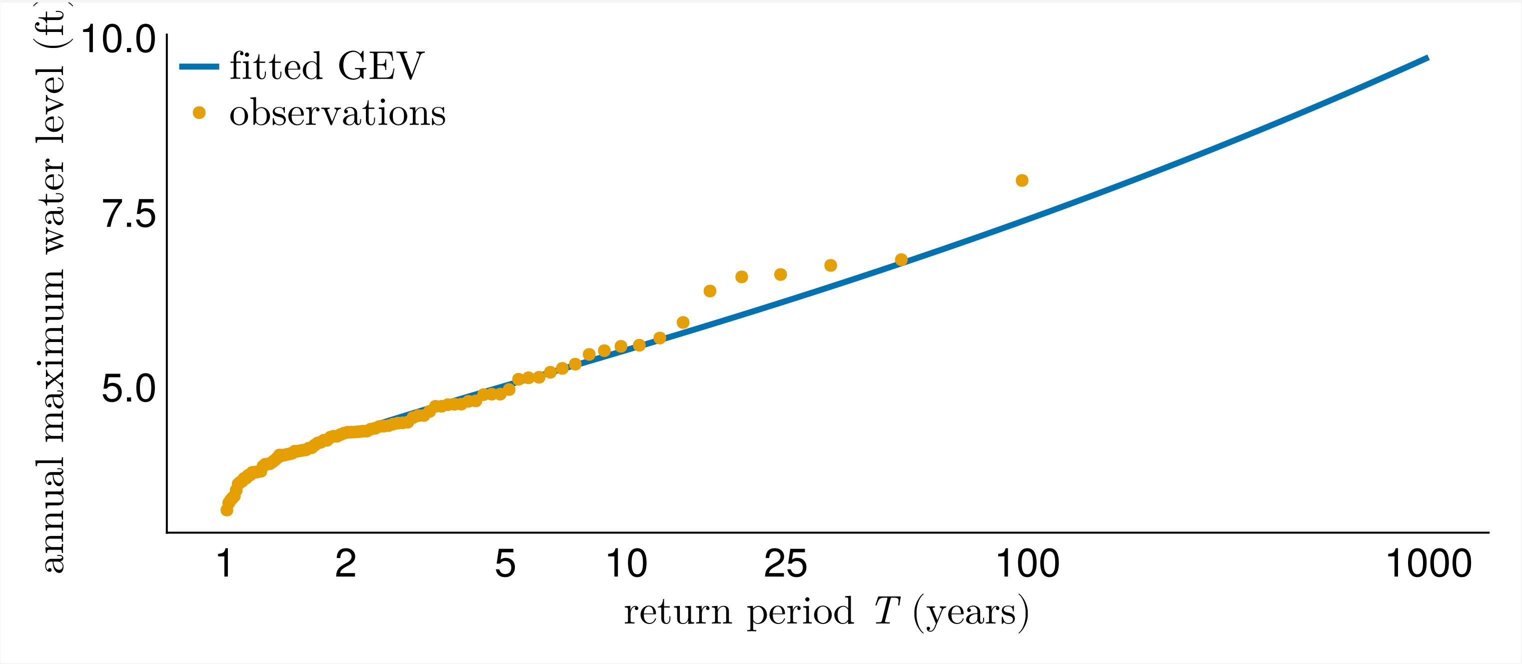
```julia
gev = only(getdistribution(gevfit(annmax.lsl)))
#| echo: false
fig = Figure(; size=(760, 350))
ax = Axis(fig[1, 1]; xlabel="annual maximum water level (ft)", ylabel="density")
hist!(ax, annmax.lsl; bins=20, normalization=:pdf, color=(:gray, 0.5))
lines!(ax, range(3, 9; length=400), pdf.(gev, range(3, 9; length=400)); linewidth=3)
fig
```

FITTING



RETURN LEVELS

```
n_obs = nrow(annmax)
lsl = sort(annmax.lsl; rev=true)
p_obs = (1:n_obs) ./ (n_obs + 1)
```



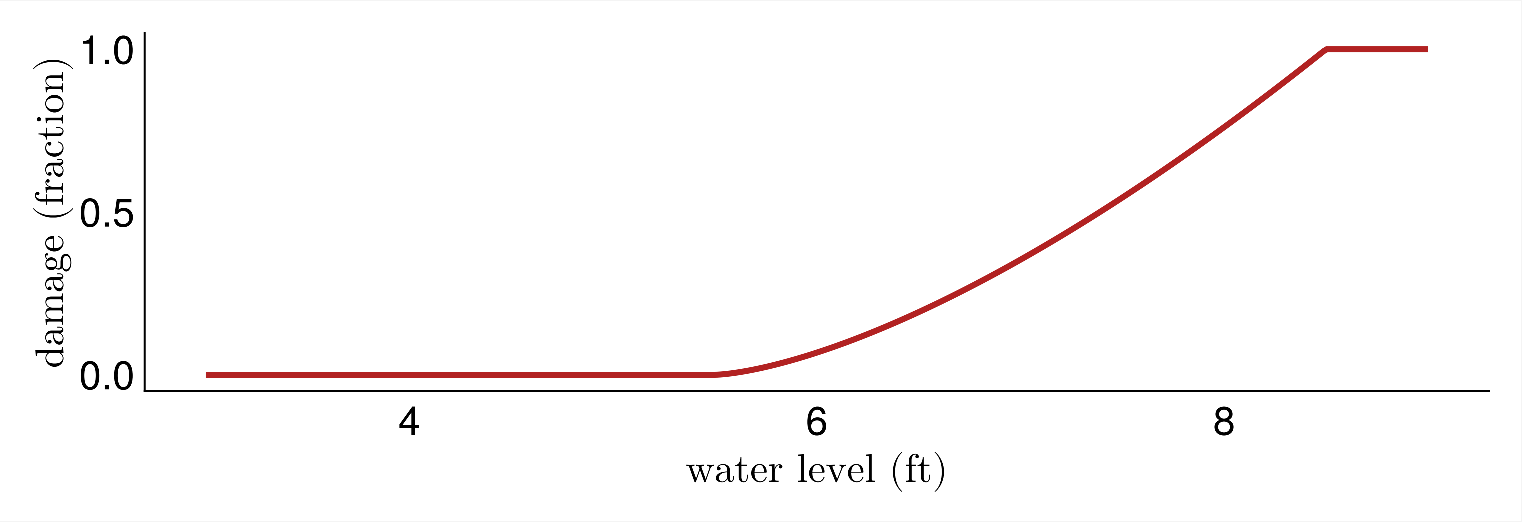
Hazard

Risk

ILLUSTRATIVE DAMAGE FUNCTION

Let h be the water level at the gauge, in feet. A building sits 5.5 ft above the gauge. Assume total loss at 8.5 ft; between those two the curve is convex.

```
damage(h) = clamp((h - 5.5) / 3, 0, 1)^1.5
```



EXPECTED DAMAGE WITH $n = 10$

```julia
levels = rand(rng, gev, 10)

println("level    ", round.(levels; digits=2))
println("damage   ", round.(damage.(levels); digits=2))
@printf("average %.3g", mean(damage.(levels)))
```

```
level   [4.98, 6.16, 5.69, 3.39, 5.39, 3.94, 4.56, 3.99, 5.17, 4.19]
damage  [0.0, 0.1, 0.02, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
average 0.0119
```

THE ESTIMATOR

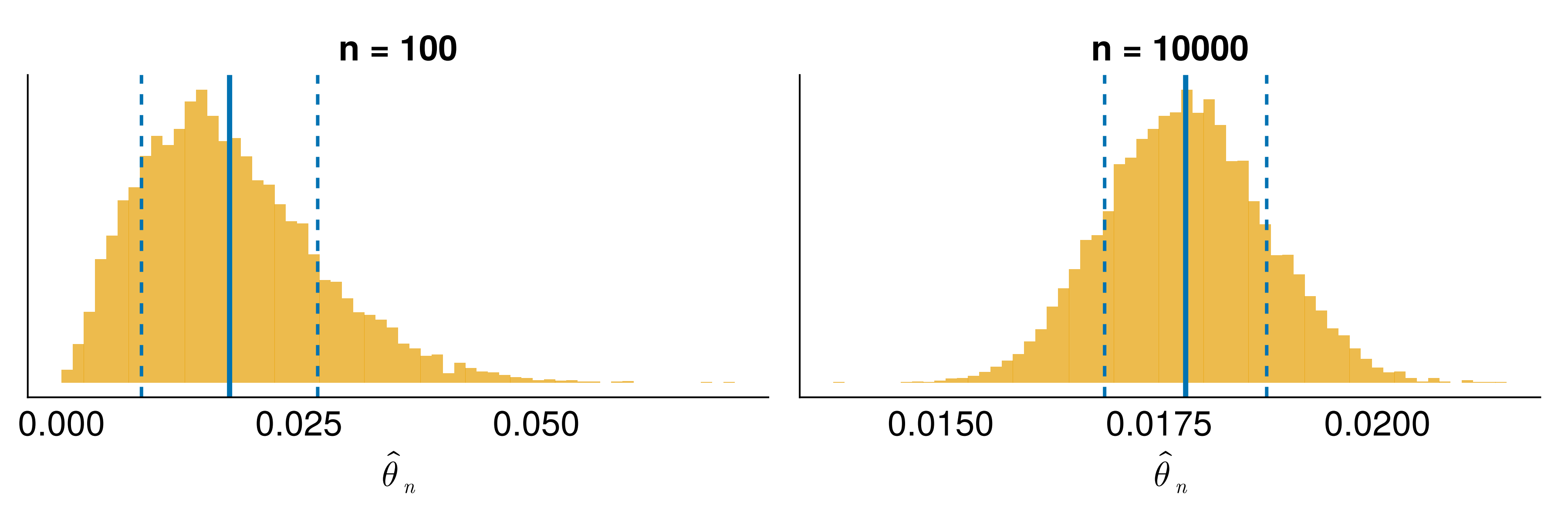
$\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} g(X_i)$, with X the water level and g the damage function, estimates $\theta = E[g(X)]$:

1. Draw n years.
2. Apply g to each.
3. Average.

```julia
function expected_damage(dist, n)
    levels = rand(rng, dist, n)
    return mean(damage.(levels))
end
```

$\hat{\theta}_n$: A RANDOM VARIABLE WITH $\mathrm{SE} = \mathrm{sd}(\mathrm{g}(X))/\sqrt{n}$

```
sd_g = std(damage.(draws))
sample_sizes = (100, 10_000)
estimates = [[expected_damage(gev, n) for _ in 1:10_000] for n in sample_sizes]
```



HOW MANY YEARS DO WE NEED TO SIMULATE?

From the whiteboard, we know we need $n = (\mathrm{CV}/\varepsilon)^2$ simulations, where ε is one standard error expressed as a fraction of the answer.

```
cv(v) = std(v) / mean(v)
n_years(c, ε) = ceil(Int, (c / ε)^2)
```

COMPARING SIMULATIONS

How many draws do we need to estimate damage versus hazard with $\varepsilon = 0.01$?

```julia
ε = 0.01
cv_damage = cv(damage.(draws))
cv_hazard = cv(draws)

@printf("damage  CV %.3g, n %d\n", cv_damage, n_years(cv_damage, ε))
@printf("hazard  CV %.3g, n %d\n", cv_hazard, n_years(cv_hazard, ε))
```

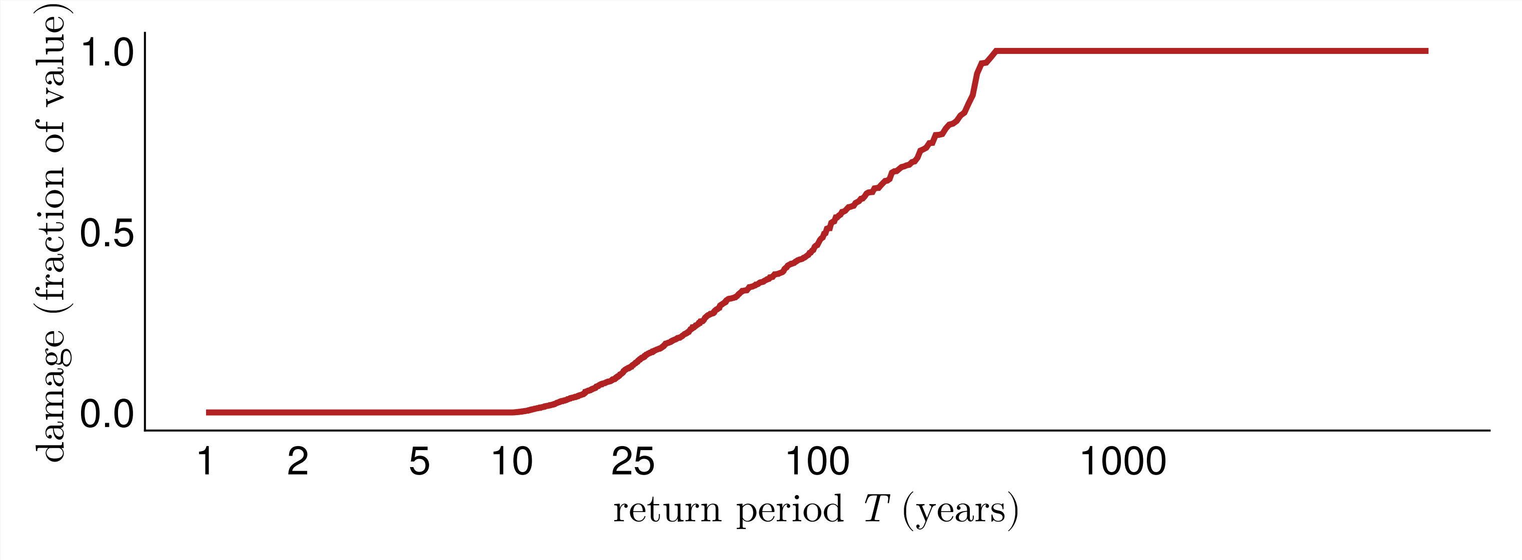
```
damage  CV 5.2, n 270669
hazard  CV 0.187, n 350
```

THE EXCEEDANCE PROBABILITY CURVE

Sort the simulated damages from largest to smallest. Rank i gets $\hat{p}_i = i/(n+1)$ and $\hat{T}_i = 1/\hat{p}_i$ per Weibull plotting positions.

```
n_sim = 10_000
losses = sort(damage.(rand(rng, gev, n_sim)); rev=true)
p_sim = (1:n_sim) ./ (n_sim + 1)
```

THE EXCEEDANCE PROBABILITY CURVE



If we know F, we can use smarter sampling

Write F for the cumulative distribution function of the water level and f for its density. Substituting $u = F(x)$ turns the integral into an average over quantiles, which we evaluate on a uniform grid of u.

$$\theta = \int d(x) f(x)\,\mathrm{d}x = \int_0^1 d(F^{-1}(u))\,\mathrm{d}u$$

THE TWO COMPARED AT TWO SAMPLE SIZES

```julia
function run_mc_comparison(n)
    u = (0.5:1.0:n) ./ n
    theta_grid = mean(damage.(quantile.(gev, u)))

    @printf(
        "n = %6d  Monte Carlo %.4g  quantiles %.4g\n",
        n,
        expected_damage(gev, n),
        theta_grid
    )
end
run_mc_comparison(1_000)
run_mc_comparison(10_000)
```

THE TWO COMPARED AT TWO SAMPLE SIZES

```
n =   1000  Monte Carlo 0.02049  quantiles 0.0178
n =  10000  Monte Carlo 0.01732  quantiles 0.01781
```

WHEN MIGHT WE PREFER BRUTE FORCE MONTE CARLO?

Set your Poll Everywhere screen name to your Rice NetID so your answers are recorded.

PollEV.com/jdossgollin