Lab 3: Depth-Damage Models

DataFrames and Distributions

Jonah Schaechter js336

Fri., Jan. 26

Table of contents

using CSV  
using DataFrames  
using DataFramesMeta  
using Distributions  
using Interpolations  
using Plots  
using StatsPlots  
using Unitful  
  
Plots.default(; margin=6Plots.mm)

# 1. Site information

Choosing Galveston Pier 21, Texas The guage is at 29° 18.6 N, 94° 47.6 W https://maps.app.goo.gl/GyanSMA2fp9rkVrT9

Our building is 302 17th St, Galveston, TX 77550, A multi-family residential unit 0.5 miles from the pier.

It is 4.41 feet or 1.34 meters above sea level in elevation. Looking at it on street view, the house appears to be on concrete blocks about 6 inches tall, giving it an effective height of 4.91 feet.

Row 98 from the data is two-story, no basement in Galveston, so we’ll be using that. The home is on concrete blocks, so we can be confident that it doesn’t have a basement.

# 2. Depth-Damage

scatter(  
 dd.depths,  
 dd.damages;  
 xlabel="Flood Depth at House",  
 ylabel="Damage (%)",  
 label="$(dd.description) ($(dd.source))",  
 legend=:bottomright,  
 size=(700, 500),  
)

function get\_depth\_damage\_function(  
 depth\_train::Vector{<:T}, dmg\_train::Vector{<:AbstractFloat}  
) where {T<:Unitful.Length}  
  
 # interpolate  
 depth\_ft = ustrip.(u"ft", depth\_train)  
 interp\_fn = Interpolations.LinearInterpolation(  
 depth\_ft,  
 dmg\_train;  
 extrapolation\_bc=Interpolations.Flat(),  
 )  
  
 damage\_fn = function (depth::T2) where {T2<:Unitful.Length}  
 return interp\_fn(ustrip.(u"ft", depth))  
 end  
 return damage\_fn  
end

get\_depth\_damage\_function (generic function with 1 method)

damage\_fn = get\_depth\_damage\_function(dd.depths, dd.damages)

#79 (generic function with 1 method)

p = let  
 depths = uconvert.(u"ft", (-10.0u"ft"):(1.0u"inch"):(30.0u"ft"))  
 damages = damage\_fn.(depths)  
 scatter(  
 depths,  
 damages;  
 xlabel="Flood Depth",  
 ylabel="Damage (%)",  
 label="$(dd.description) ($(dd.source))",  
 legend=:bottomright,  
 size=(800, 400),  
 linewidth=2,  
 )  
end  
p

The depth damage curve here shows how damage increases rapidly with flood depth at first, but then begins to slow down. This makes sense, as the bottom of the curve essentially represents the difference between not flooding,and flooding. Once a high enough depth is reached, damage begins to increase linearly with depth.

# 3. Expected annual damages

gauge\_dist = GeneralizedExtremeValue(5, 1.5, 0.1)

GeneralizedExtremeValue{Float64}(μ=5.0, σ=1.5, ξ=0.1)

I know plotting this isn’t necessary, but I like to plot things.

p1 = plot(  
 gauge\_dist;  
 label="Gauge Distribution",  
 xlabel="Water Level (ft)",  
 ylabel="Probability Density",  
 legend=:topright,  
 linewidth=2,  
)

offset = 4.91 #The house is 4.91 feet above the guage (Land is 4.41 ft + 0.5 ft concrete blocks the house is built on.)  
house\_dist = GeneralizedExtremeValue(gauge\_dist.μ - offset, gauge\_dist.σ, gauge\_dist.ξ)

GeneralizedExtremeValue{Float64}(μ=0.08999999999999986, σ=1.5, ξ=0.1)

plot!(p1, house\_dist; label="House Distribution", linewidth=2)

Monte Carlo Algorithm: 1. Sample values from the distribution of hazard 2. For each value, estimate the damage using the depth-damage function 3. Average the damages

1. Sample values from the distribution of hazard

N = 1000000 #N = one million  
Samples = rand(house\_dist, N) #Get one million random samples from our distribution

1000000-element Vector{Float64}:  
 1.8015169417537238  
 -1.0011994152648265  
 -0.19035603096870624  
 -0.9291523844118692  
 -0.40471965599593085  
 1.2344808339151354  
 4.073783218450265  
 -1.8918552443381214  
 3.2464556608422552  
 1.0051870117485464  
 0.4792959598435221  
 0.022200403910814737  
 1.3766998981754683  
 ⋮  
 0.5069116732837212  
 1.7694930170190712  
 -1.2073775534163693  
 -1.2812615571968604  
 -0.37656292497252114  
 0.1812154523405771  
 0.25625100018069913  
 0.4319179311099318  
 -0.15569685935312455  
 2.2510112147002  
 -0.6046619147320399  
 0.00612117870849288

1. For each value, estimate the damage using the depth-damage function

FtSamples = eval(Samples)u"ft"  
house\_damage = damage\_fn.(FtSamples)

1000000-element Vector{Float64}:  
 25.809101650522344  
 0.0  
 4.048219845156469  
 0.354238077940654  
 2.9764017200203456  
 22.40688500349081  
 34.22134965535079  
 0.0  
 31.739366982526764  
 21.03112207049128  
 12.668735357496352  
 5.355206462573036  
 23.26019938905281  
 ⋮  
 13.11058677253954  
 25.616958102114424  
 0.0  
 0.0  
 3.1171853751373946  
 7.899447237449233  
 9.100016002891186  
 11.910686897758909  
 4.2215157032343775  
 28.0040448588008  
 1.9766904263398004  
 5.097938859335886

1. Average the damages

#print(length(house\_damage), "\n")  
AvgDamage = sum(house\_damage)/length(house\_damage)  
print("According to our Monte Carlo, the expected annual damages for this propery are ", round(AvgDamage; digits=2), " dollars.")

According to our Monte Carlo, the expected annual damages for this propery are 16.06 dollars.

# 4. Discussion

Let’s take a look at 100 of our sampled points and plot them on the depth-damage curve to see how they’re distributed.

We can see that most of the points are along the lower ends of the curve. There are, however, still a few points high on the curve, reflecting the inevitability of large floods once enough time passes and enough samples are taken.

HundSamples = rand(Samples, 100) #one hundred of our samples (Julia doesn't like to plot one million)  
HundFtSamples = eval(HundSamples)u"ft"  
Hundhouse\_damage = damage\_fn.(HundFtSamples)  
  
  
scatter(  
 HundFtSamples,  
 Hundhouse\_damage;  
 xlabel="Flood Depth at House",  
 ylabel="Damage (%)",  
 label="100 Points sampled for our MC algorithm",  
 legend=:bottomright,  
 size=(700, 500),  
)

Sea levels are expected to rise by 4.4 ft in this location by 2080 according to NOAA according to intermediate-high projections: https://coast.noaa.gov/slr/#/layer/sce/4/-10554960.5624167/3413062.4218919543/12/satellite/85/0.8/2080/interHigh/midAccretion

Let’s re-do our Monte Carlo approximation for an extra 4.4ft of sea level rise by lowering the height of our home above sea level by 4.4ft.

offset2080 = 0.51 #The house is 0.51 feet above the guage (Land is 4.41 ft + 0.5 ft concrete blocks - 4.4 ft sea level rise.)  
house\_dist2080 = GeneralizedExtremeValue(gauge\_dist.μ - offset2080, gauge\_dist.σ, gauge\_dist.ξ)

GeneralizedExtremeValue{Float64}(μ=4.49, σ=1.5, ξ=0.1)

N = 1000000 #N = one million  
Samples2080 = rand(house\_dist2080, N) #Get one million random samples from our distribution

1000000-element Vector{Float64}:  
 2.897377485550785  
 6.977095537706781  
 6.067656344179918  
 5.566998856222131  
 3.3817579572353216  
 3.790709359806942  
 7.362419910608396  
 16.440298894076008  
 4.18670851593461  
 2.0478466694341204  
 3.333304864103834  
 3.387963502634421  
 8.557128081134273  
 ⋮  
 4.635309680039584  
 4.468607171157648  
 5.511551283948492  
 3.452922683120928  
 4.224959617915194  
 5.678779460783211  
 4.365125313525235  
 4.315776607698157  
 7.314971266843784  
 3.74888557233367  
 9.361811483551993  
 4.349561902431231

FtSamples2080 = eval(Samples2080)u"ft"  
house\_damage2080 = damage\_fn.(FtSamples2080)

1000000-element Vector{Float64}:  
 30.58950994220314  
 39.97709553770678  
 39.067656344179916  
 38.13399771244426  
 32.14527387170597  
 33.372128079420825  
 40.0  
 60.880597788152016  
 34.56012554780383  
 27.19138667773648  
 31.9999145923115  
 32.16389050790326  
 41.114256162268546  
 ⋮  
 35.90592904011875  
 35.405821513472944  
 38.023102567896984  
 32.35876804936278  
 34.67487885374558  
 38.35755892156642  
 35.09537594057571  
 34.94732982309447  
 40.0  
 33.24665671700101  
 42.723622967103985  
 35.04868570729369

AvgDamage = sum(house\_damage2080)/length(house\_damage2080)  
print("According to our Monte Carlo, the expected annual damages for this property are ", round(AvgDamage; digits=2), " dollars in the year 2080.")

According to our Monte Carlo, the expected annual damages for this property are 36.99 dollars in the year 2080.

plot!(p1, house\_dist2080; label="2080 House Distribution", linewidth=2)