Lab 5: Sea-Level Rise

Jonathan Gan wg18

Fri., Feb. 16

1 Setup

1.1 The usual

As always:

1. Clone the lab repository to your computer
2. Open the lab repository in VS Code
3. Open the Julia REPL and activate, then instantiate, the lab environment
4. Make sure you can render: `quarto render template.qmd` in the terminal.
    - If you run into issues, try running `] build IJulia` in the Julia REPL (`]` enters the package manager).
    - If you still have issues, try opening up `blankfile.py`. That should trigger VS Code to give you the option to install the Python extension, which you should do. Then you should be able to open a menu in the bottom right of your screen to select which Python installation you want VS Code to use.

1.2 Load packages

```julia
using CSV
using DataFrames
using DataFramesMeta
using Distributions
using Plots
using StatsPlots
using Unitful

Plots.default(; margin=5Plots.mm)
```

1.3 Local package

```julia
using Revise
using HouseElevation
```

1.4 House

```
house = let
    haz_fl_dept = CSV.read("data/haz_fl_dept.csv", DataFrame) # read in the file
    id = 140
    row = @rsubset(haz_fl_dept, :DmgFnId == id)[1, :] # select the row I want
    area = 1200u"ft^2"
    height_above_gauge = 2u"ft"
    House(
        row;
        area=area,
        height_above_gauge=height_above_gauge,
        value_usd=400_000,
    );
end
```

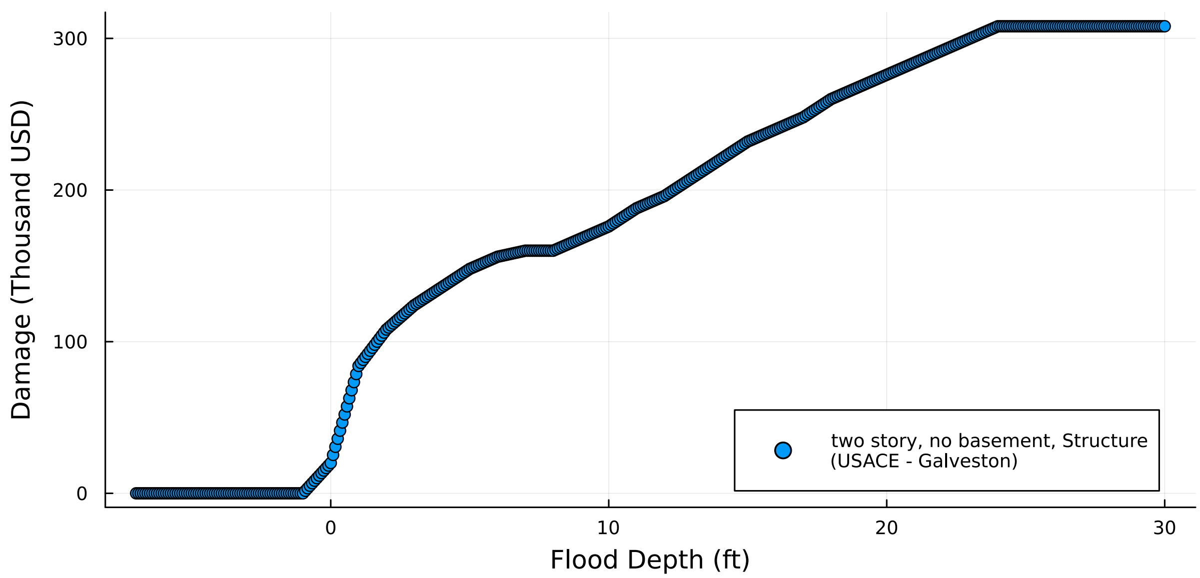
```
House{Int64}(1200, 400000, 2, DepthDamageFunction{Interpolations.Extrapolation{Float64, 1, Inte
  0.0
  0.0
  0.0
  0.0
  5.0
 21.0
 27.0
 31.0
 34.0
 37.0
 39.0
 40.0
 40.0

 52.0
 55.0
 58.0
 60.0
 62.0
 65.0
 67.0
 69.0
 71.0
 73.0
 75.0
 77.0), String7("RES1"), "140", String31("USACE - Galveston"), "two story, no basement, Structu
```

I obtained the house value and area from Zillow, where a 1500sqft house in the vicinity of the gauge had a value of approximately $400k. I used a depth damage function from the USACE Galveston data set to ensure that the function would be most appropriate for the locality that my chosen house is located in, and the structure of the house also matches the description of the depth damage function (two stories, no basement).

1.5 Depth Damage Curve

```
let
    depths = uconvert.(u"ft", (-7.0u"ft"):(1.0u"inch"):(30.0u"ft"))
    damages = house.ddf.(depths) ./ 100
    damages_1000_usd = damages .* house.value_usd ./ 1000
    scatter(
        depths,
        damages_1000_usd;
        xlabel="Flood Depth",
        ylabel="Damage (Thousand USD)",
        label="$(house.description)\n($(house.source))",
        legend=:bottomright,
        size=(800, 400),
        yformatter=:plain, # prevents scientific notation
    )
end
```

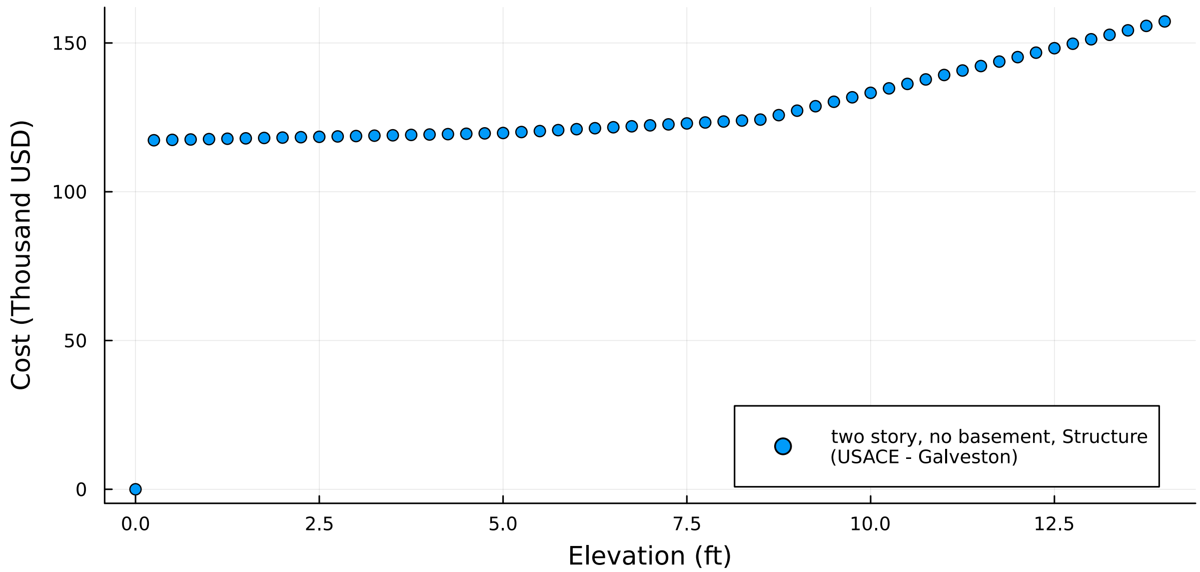


1.6 Elevation Cost

```
let
    elevations = 0u"ft":0.25u"ft":14u"ft"
    costs = [elevation_cost(house, e ) for e  in elevations]
    scatter(
        elevations,
        costs ./ 1_000;
        xlabel="Elevation",
```

```
        ylabel="Cost (Thousand USD)",
        label="$(house.description)\n($(house.source))",
        legend=:bottomright,
        size=(800, 400),
        yformatter=:plain, # prevents scientific notation
    )

end
```



1.7 Sea Level Data

```
slr_scenarios = let
    df = CSV.read("data/slr_oddo.csv", DataFrame)
    [Oddo17SLR(a, b, c, tstar, cstar) for (a, b, c, tstar, cstar) in eachrow(df)]
end
println("There are $(length(slr_scenarios)) parameter sets")
```

```
There are 34895 parameter sets
```

1.8 Storm Surge and Discount Rate

```
function draw_surge_distribution()
     = rand(Normal(5, 1))
     = rand(Exponential(1.5))
     = rand(Normal(0.1, 0.05))
    GeneralizedExtremeValue( ,  ,  )
end

[draw_surge_distribution() for _ in 1:1000]
```

```

function draw_discount_rate()
    return rand(Normal(0.04, 0.02))
end
```

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

1.9 Single simulation

```
p = ModelParams(
    house=house,
    years=2024:2100
)

sow = SOW(
    rand(slr_scenarios),
    draw_surge_distribution(),
    draw_discount_rate()
)

a = Action(5.0u"ft")

res = run_sim(a, sow, p)
```

```
-343331.7162850235
```

1.10 Large simulations

```
sows = [SOW(rand(slr_scenarios), draw_surge_distribution(), draw_discount_rate()) for _ in 1:1
range = 0u"ft":1u"ft":10u"ft"
actions = [Action(height) for height in range]
results = [run_sim(a, s, p) for (a, s) in zip(actions, sows)]

df = DataFrame(
    npv=results,
    Δh_ft=[a.Δh_ft for a in actions],
    slr_a=[s.slr.a for s in sows],
    slr_b=[s.slr.b for s in sows],
    slr_c=[s.slr.c for s in sows],
    slr_tstar=[s.slr.tstar for s in sows],
    slr_cstar=[s.slr.cstar for s in sows],
    surge_ =[s.surge_dist.  for s in sows],
    surge_ =[s.surge_dist.  for s in sows],
    surge_ =[s.surge_dist.  for s in sows],
    discount_rate=[s.discount_rate for s in sows],
)
```

	npv	Δh_ft	slr_a	slr_b	slr_c	slr_tstar	slr_cstar	surge_	
	Float64	Int64	Float64	Float64	Float64	Float64	Float64	Float64	
1	-6.02133e6	0	35.2794	2.12838	0.0015293	2037.33	14.9556	4.08362	...
2	-5.04679e6	1	48.6024	2.53905	0.00492611	2069.3	24.6171	6.14631	...
3	-2.06565e6	2	13.3172	1.1593	-0.00534686	2016.71	14.7752	5.71988	...
4	-1.90478e6	3	19.5371	1.73964	-0.00070208	2024.83	18.3066	4.12548	...
5	-3.44375e6	4	51.7384	2.71214	0.00516874	2080.18	28.3083	6.50794	...
6	-6.08948e5	5	42.4716	2.32743	0.00250945	2033.42	11.996	3.80531	...
7	-3.62146e5	6	35.6391	2.14407	0.000929007	2064.23	33.2604	5.28525	...
8	-7.07332e5	7	30.4574	2.08525	0.00204452	2054.25	26.2393	4.8882	...
9	-1.23602e5	8	22.1212	2.26554	0.00395675	2051.96	25.0203	2.79568	...
10	-127245.0	9	56.7672	2.63149	0.00453798	2066.07	4.14236	4.80007	...
11	-1.58234e5	10	41.2219	2.20374	0.00121126	2033.17	22.728	3.72102	...

I chose to sample a range of actions from 0ft to 10ft to determine the elevation at which would bring the greatest benefit to the house.

1.11 Analysis

From my analysis, it appears that there is no noticable correlation between the height of elevating my house to gaining benefit. In fact, all the NPVs that I've calculated are negative. However, through my iterations it appears that elevating the house higher would result in a less negative NPV.

The most important parameters other than the height of elevating our houses is the distribution of storm surges. With this model, it is not very sophisticated and we're just using standard distributions to attempt to model a typical storm surge without accounting for the actual weather and storm conditions at our house location.

If I had unlimited computing power, I would try to run more simulations, but not before refining the model to include better distributions of storm surges and discount rates to ensure that the distributions that are used can accurately depict conditions at my chosen house.

These results, particularly in my case of having negative NPVs, suggest that sometimes inaction might make the mose sense on the individual level, since it would not make financial sense to spend money on elevating houses that are already at high risk of being damaged by future storm surges due to their location and geography. Rather, we should explore developing zoning policies and making choices that would have us build in locations that do not have these weather risks, and thus we are able to build cities that would become more resillient to climate change.