Lab 4: House Elevation NPV Analysis

Catherine Jackson

Thu., Feb. 8

1 Lab 3: Previous Work

1.1 Packages

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

Plots.default(; margin=6Plots.mm)
```

1.2 Site information

1.2.1 Site of Interest

I have chosen the Galveston Pier 21 site as my water gage and a home at 304 18th St, Galveston, TX 77550 to be my site of interest. I chose this site as it is a residential building within half a mile of the water gage itself. Furthermore, it is still very close to the coast of Galveston channel. For this reason, it is a good candidate for exploring and understanding the impacts of coastal flooding and the complexities of making decisions on flood mitigation strategies.

1.3 Depth-Damage

```
include("depthdamage.jl")
haz_fl_dept = CSV.read("data/haz_fl_dept.csv", DataFrame);
demo_row = @rsubset(
    haz_fl_dept, :Description == "two story, no basement, Structure", :Occupancy == "RES1",
)[
    1, :,
]
dd = DepthDamageData(demo_row);
fieldnames(typeof(dd));
```

The function below is taken from index.qmd and is used to interpolate the depth-damage curve.

```
global damage_fn = get_depth_damage_function(dd.depths, dd.damages);
```

```
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,                                                    ①
        dmg_train;
        extrapolation_bc=Interpolations.Flat(),                      ②
    )

    damage_fn = function (depth::T2) where {T2<:Unitful.Length}
        return interp_fn(ustrip.(u"ft", depth))                      ③
    end
    return damage_fn                                                 ④
end
```

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

1.4 Expected Annual Damages

1.4.1 Flood distribution

Below, we have the flood distribution of our gage adjusted based on the elevation of our site.

```
gauge_dist = GeneralizedExtremeValue(5, 1.5, 0.1)
p1 = plot(
    gauge_dist;
    label="Gauge Distribution",
    xlabel="Water Level (ft)",
    ylabel="Probability Density",
    legend=:topright,
    linewidth=2,
)
offset = 4.3;
house_dist = GeneralizedExtremeValue(gauge_dist.  - offset, gauge_dist. , gauge_dist. );
```

1.4.2 Monte Carlo Sampling

```
n_samples = 1_000_000;
vecsamples = rand(house_dist, n_samples);
vecsamples = vecsamples .* u"ft"
damages = damage_fn.(vecsamples);
expecteddamages = mean(damages)
```

```
20.231598315049364
```

2 Lab 4: House Elevation NPV Analysis

2.1 Packages

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

Plots.default(; margin=6Plots.mm)

include("depthdamage.jl")
```

2.2 Site information

As with Lab 03, I chose to study a property near Pier 21 in Galveston TX, 304 18th St, Galveston, TX 77550. Using information from Zillow and HAR, I was able to find information about the value of the home itself and the square footage of the property. The home is valued at $264,456 and has a square footage of 881 square feet. However, this is made complicated by the fact that floods only damage **structures** and not the land itself. For this reason, I found nearby lots, without homes or structures, and analyzed their values. Based on this analysis, I found that a similar location, close to the water and heart of Galveston, could go for anywhere around $70,000. This leaves a structural value of $194,456.

```
house_area = 881*u"ft^2" ;
house_value = 194456 ;
Δh = 1.0*u"ft" ;
```

2.3 Expected Damages

As a quick reminder, the expected annual damages are calculated as follows:

```
n_samples = 1_000_000;
vecsamples = rand(house_dist, n_samples);
vecsamples = vecsamples .* u"ft"
damages = damage_fn.(vecsamples);
expecteddamages = mean(damages)
```

```
20.264530977084476
```

This expecteddamages value presents the expected **percentage** of the structural value lost during a flood event. Running from the baseline code we have above, we find that the expected annual damage falls around 20%, a shockingly high number. This is consistent with the clear vulnerability of the Galveston community and increasing exposure to flooding.

However, in this lab, non-stationarity in the parameters which define this flood distribution will not be considered. However, this lab will use the methodology of expected damages to better understand the implications of raising this home to mitigate flood damages.

Here, before discussing the functions, I redefine the variables from Lab 03.

```
flood_dist = house_dist; #already includes the offset
damage_fn = damage_fn;
house_area = house_area;
house_value = house_value;
elevation_cost = get_elevation_cost_function() ;
```

Next, we can take a quick preemptive look at what it might cost to elevate the house:

```
Δh2 = 1.0*u"ft";
c_constr = elevation_cost(Δh2, house_area)
```

```
91919.228
```

This line tells us that, to elevate the house one foot it could take around $90,000 dollars. Admittedly, I did some research of my own and this value does seem a little high. However, the point of this lab is not to get the exact value of the cost of elevation, but to understand the implications of elevation on the NPV of the home. However, it could be an interesting exercise to go back and quantify this cost of elevation more precisely.

2.4 Single Year Cost-Benefit

Below, we define the cost benefit for a single year. This function takes in the flood distribution, the damage function, the cost of elevation, the area of the house, the value of the house, and the elevation change. It then calculates the expected damages (using the flood distribution, consequent damage function, and a Monte Carlo Sampling to find the percent damages to multiply against the house value) and the cost of elevation (calculated with the house elevation height, square footage, and elevation_cost function) and returns the total cost and benefit.

```
function single_year_cost_benefit(flood_dist, damage_fn, elevation_cost, house_area, house_v

```

```
    # calculate the expected damages
    n_samples = 1_000_000;
    new_flood_dist = GeneralizedExtremeValue(flood_dist.  - ustrip(Δh), flood_dist. , flood_d
    vecsamples = rand(new_flood_dist, n_samples);
    vecsamples = vecsamples .* u"ft"
    damages = damage_fn.(vecsamples);
    expecteddamages = mean(damages)
    c_dmg = house_value * expecteddamages / 100;

    # calculate the cost of elevating
    c_constr = elevation_cost(Δh, house_area)

    # return the total cost and benefit
    return -c_constr - c_dmg
end
```

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

2.5 Net Present Value

However, this just gives us the expected damages or cost for a singular year. We need to be able to calculate the total damages we might experience over a period of time. For this reason, we define the function `npv_cost_benefit` which takes in the flood distribution, the damage function, the cost of elevation, the area of the house, the value of the house, the elevation change, the time period, and the discount rate. It then calculates the expected damages for each year (applies a discount to account for the fact that, for many people, a dollar today is "worth" more than a dollar in the future), and returns the expected damages over the entire period.

It is important to note that **elevation is only considered in the first year**; the house is elevated as a one time cost. Consequently, as seen in the "if" condition below, Δh is only input into the function when i=1 (e.g. the first year). However, we also need to remember that, once we elevate the house, the flood_distribution changes. Consequently, a new variable `post_elev_flood_dist` is defined to represent the flood distribution after the elevation. This is then used to calculate the expected damages for the subsequent years.

```
function npv_cost_benefit(flood_dist, damage_fn, elevation_cost, house_area, house_value, Δh
    npv=0;
    years = 1:T
    for i in years
        if i == 1
            snpv = single_year_cost_benefit(flood_dist, damage_fn, elevation_cost, house_are
            npv = npv+snpv
        else
            post_elev_flood_dist = GeneralizedExtremeValue(flood_dist.  - ustrip(Δh), flood_
            snpv = single_year_cost_benefit(post_elev_flood_dist, damage_fn, elevation_cost,
            npv = npv+snpv
        end
    end
```

```
    return npv
end
```

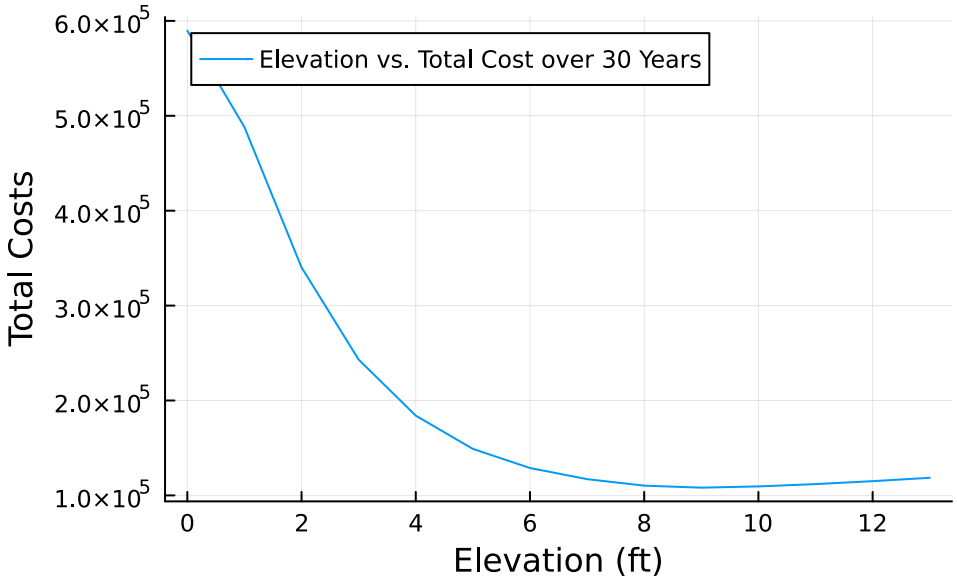
```
npv_cost_benefit (generic function with 1 method)
```

2.6 Discussion

To put this into context, when, this equation tells us how much we might pay over our time period of interest. This includes the initial elevation cost, which we only incur once, and the expected damages over the time period (with a discount rate). In this way, with our npv representing the amount of money we will spend given a different action a (e.g. the elevation), we can look for the choice a which minimizes the npv, or the total amount we have to spend over time T.

Consider looking at different elevation heights over a time period T = 30 years!

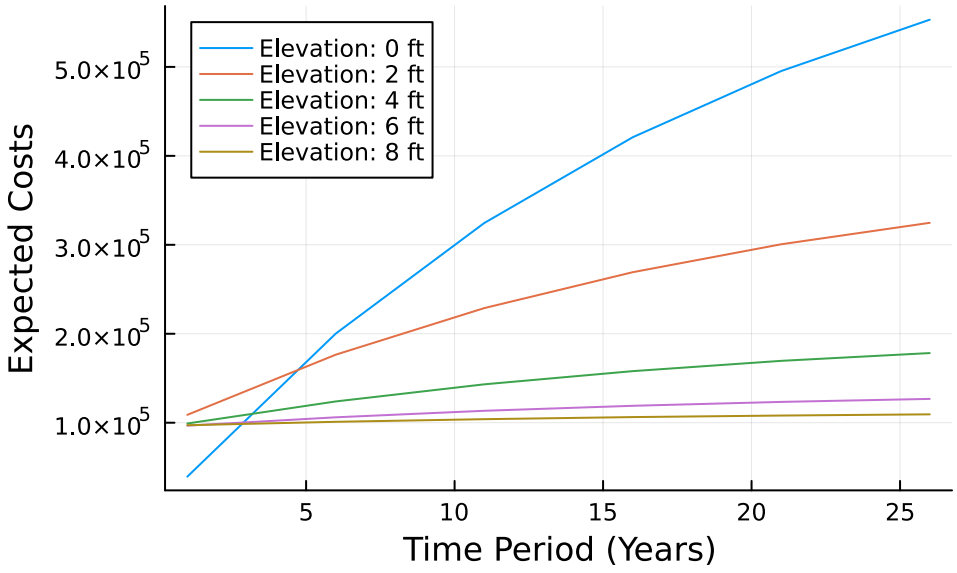
```
elevations = [];
totalcosts =[];
discount_rate = 0.05;
T = 30;

for i = 0:13
    push!(elevations, i)
    npv = npv_cost_benefit(flood_dist, damage_fn, elevation_cost, house_area, house_value, i
    push!(totalcosts, npv)
end

positive_totalcosts = abs.(totalcosts)

plot()
plot(elevations, positive_totalcosts, xlabel="Elevation (ft)", ylabel="Total Costs", label="
```



This graph, though it is limited by the bounds of the elevation cost function, shows that there is a critical point at which the benefits of avoiding flood damages no longer outweighs the cost of elevating the house. In other words, elevating the house has diminishing returns. This is a good point to find on the graph because it can inform the optimal elevation height. Based on the graph produced by the block above, it seems that elevating to around 9 feet produces the smallest expected cost over these 30 years.

We can also visualize this another way. Consider plotting the expected damages as the time period grows, with each line representing a different elevation height.

```
p = plot()
for i = 0:2:9
    npvs = [];
    years = [];
    for j = 1:5:30
        npv = npv_cost_benefit(flood_dist, damage_fn, elevation_cost, house_area, house_valu
        push!(npvs, npv)
        push!(years, j)
    end
    p = plot!(years, abs.(npvs), label="Elevation: $i ft")
end
p = plot!(xlabel="Time Period (Years)", ylabel="Expected Costs", legend=:topleft)
p
```

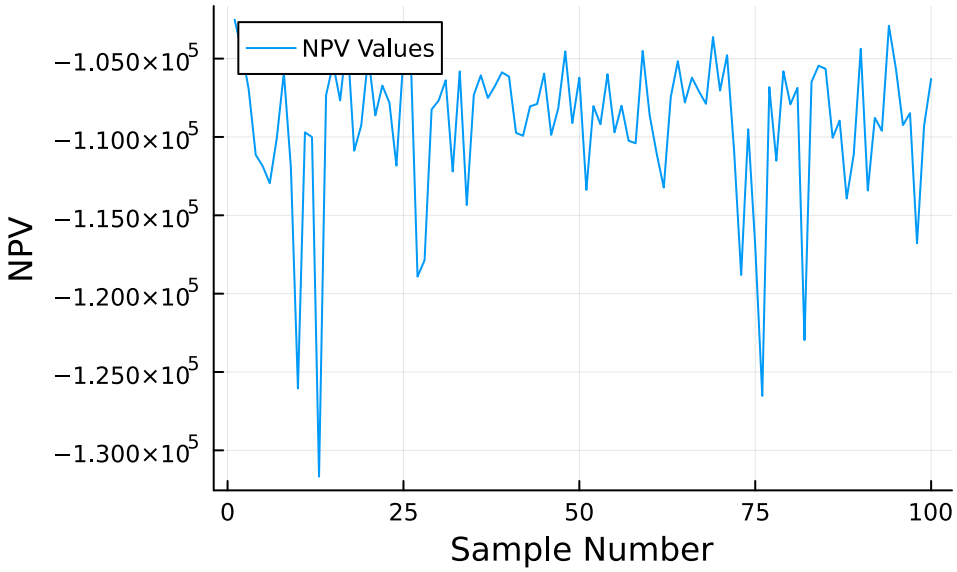


This graph plots, with each elevation as a line, the expected costs as T goes from 0 to 30. For the first line, the expected damages start very small, as there is no elevation cost the first year, but they grow quickly as the home is vulnerable to flooding every single year. As the house is elevated, the slope of this line decreases as the vulnerability each year declines. Eventually, the line becomes almost flat as the intercept is determined by the elevation cost and there is relatively no year to year vulnerability.

Building on the discussion above, this helps us understand the NPV for different actions. Elevating the house helps mitigate flooding, reducing the expected damages during our time period. However, there is a point at which the threat of flooding at that new elevation becomes so small that further elevation is not cost effective.

This type of analysis is clearly very beneficial when it comes to decision comparisons.

```
discount_rate_dist = Normal(0.05, 0.03)
n_samples = 100
vecsamples = rand(discount_rate_dist, n_samples)
npv_values = [npv_cost_benefit(flood_dist, damage_fn,
    elevation_cost, house_area, house_value, 9*u"ft", 30, discount_rate) for discount_rate i
expected_npv = mean(npv_values)
plot(1:n_samples, npv_values, label="NPV Values", xlabel="Sample Number", ylabel="NPV", lege
```



The example above shows a potential scenario in which we include uncertainty in the discount value. This is a wise parameter in which we can consider uncertainty as it is a very hypothetical representation of the fact that people tend to value money in the present more than in the future. It is a very subjective estimation, and considering this uncertainty is important.

Note: I would have liked to include more samples, but due to the limitations of my own personal laptop it was not possible.

Furthermore, we can discuss the limitations of this model. Firstly, there are many assumptions made to simplify this analysis.

1. First, the flood distribution is assumed to be constant over time. This is not realistic, as flood distributions are known to change over time. This is especially true in the context of climate change, where the frequency and intensity of floods are expected to increase.
2. Second, the damage function is assumed to be constant over time. This is also not realistic, as the value of the home and the cost of construction are expected to change over time.
3. Third, this model assumes that, if damage to the home is incurred in one year, it is instantaneously repaired.
4. Fourth, the elevation costs produced by the function from depthdamage.jl seemed to be very high. This is a very important parameter to consider, and it is important to have a more accurate representation of this cost.
5. Etc.

These are just a few examples of the limitations; when it comes to modeling, understanding that there are many complex and interlacing sources of uncertainty is key. However, this model is a good starting point for understanding the implications of elevation on the NPV of a home.