Lab 4: House Elevation NPV Analysis

Andres Calvo - ac228

Thu., Feb. 15

1 Data preparation and functions

As in the previous lab, the functions for reading the csv file, interpolation, and damage management are used.

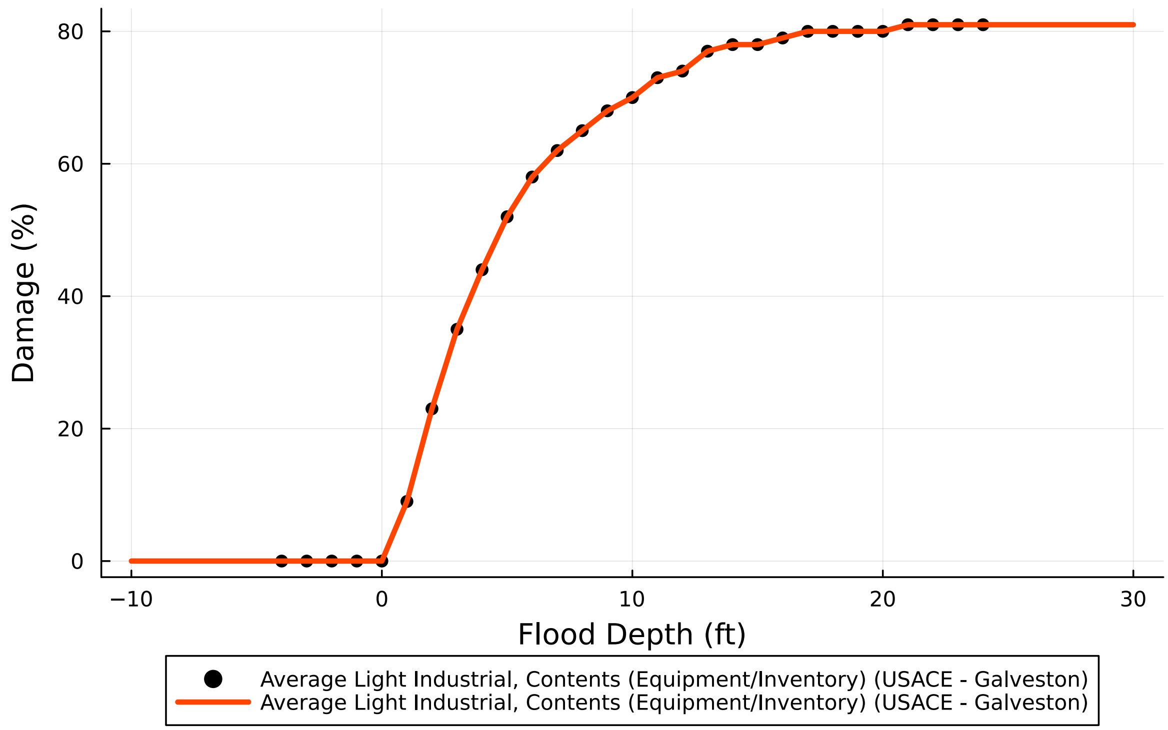
```
using CSV
using DataFrames
using DataFramesMeta
using Distributions
using Interpolations
using Plots
using StatsPlots
using Unitful

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

# Depth-damage dataset
haz_fl_dept = CSV.read("data/haz_fl_dept.csv", DataFrame)
# Data management function
include("depthdamage.jl")

# Interpolation function generator
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
```

1.1 DD - Curve

The final DD-curve to use is the following one that corresponds to the "Average Ligth Industrial, Contents (Equipment/Inventory)" from USACE in Galveston.

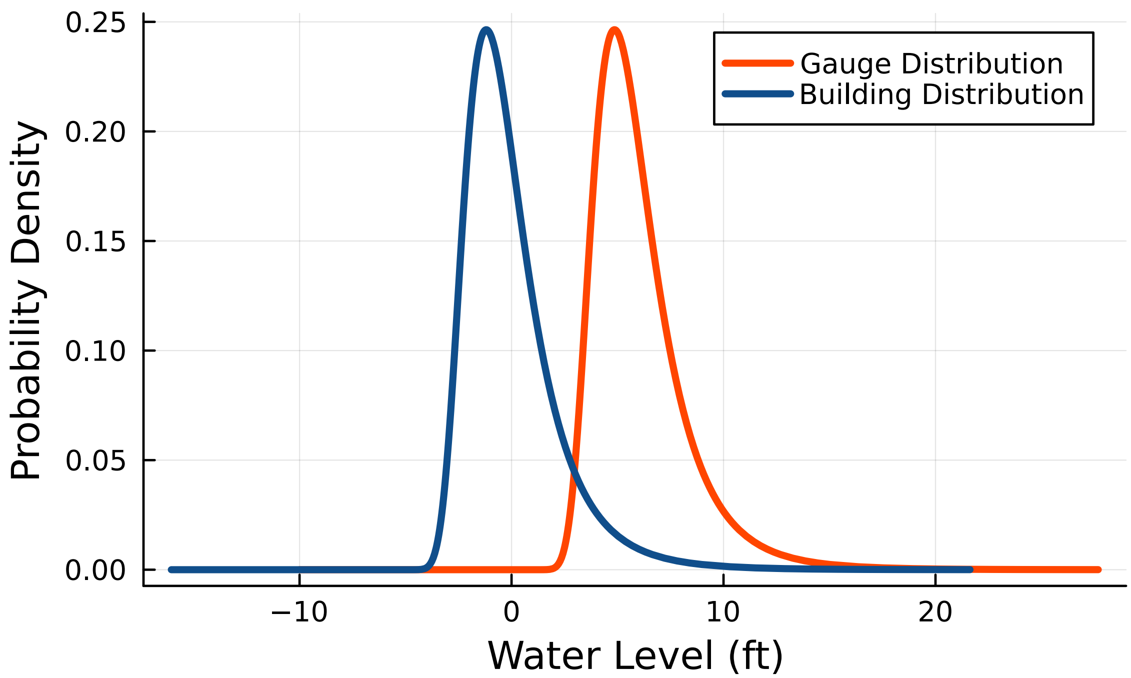
```
dd_industrial = @rsubset(
    haz_fl_dept, :Description == "Average Light Industrial, Contents (Equipment/Inventory)"
)[1, :,]

dd_ind = DepthDamageData(dd_industrial)

scatter(
    dd_ind.depths,
    dd_ind.damages;
    xlabel = "Flood deph",
    ylabel = "Structural Damage (%)",
    label = "$(dd_ind.description) ($(dd_ind.source))",
    legend = :outertop,
    color = "black",
    size = (700,500),
)
dmg_fn_ind = get_depth_damage_function(dd_ind.depths, dd_ind.damages)

p = let
    depths = uconvert.(u"ft", (-10.0u"ft"):(1.0u"inch"):(30.0u"ft"))
    dmg_ind = dmg_fn_ind.(depths)
    plot!(
        depths,
        dmg_ind;
        xlabel = "Flood Depth",
        ylabel = "Damage (%)",
        label = "$(dd_ind.description) ($(dd_ind.source))",
        legend = :outerbottom,
        size = (700, 500),
        color = "orangered",
        linewidth = 3,
    )
end
p
```



The flood depth distribution is the following one for the gauge and the building with an offset of 6.05 ft that uses a Generalized Extreme Value distribution with parameters that are yet to be calibrated for the analized gauge data.

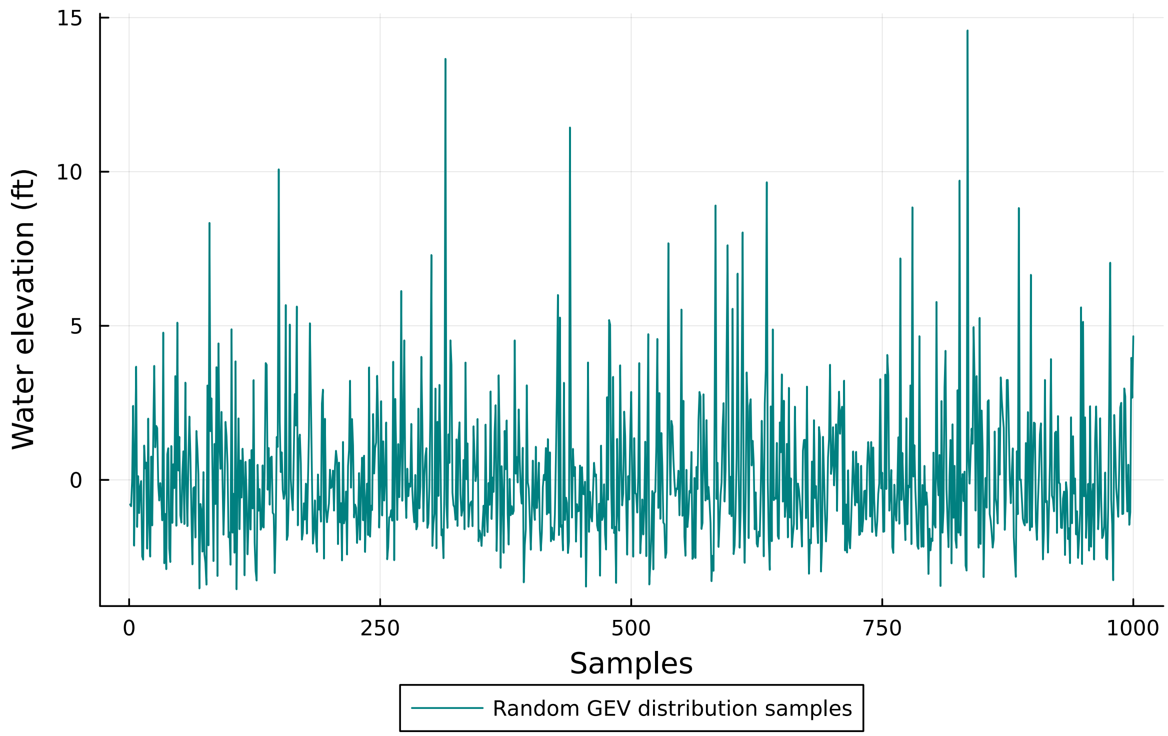
```
gauge_dist = GeneralizedExtremeValue(5, 1.5, 0.1)
p1 = plot(
    gauge_dist;
    label = "Gauge Distribution",
    xlabel = "Water Level (ft)",
    ylabel = "Probability Density",
    legend = :topright,
    color = "orangered",
    linewidth = 3,
)

offset = 12.29 - 6.24 # Industrial buiding is 6.05 feet above gauge
building_dist = GeneralizedExtremeValue(gauge_dist.  - offset, gauge_dist. , gauge_dist. )

plot!(
    p1,
    building_dist;
    label = "Building Distribution",
    color = "dodgerblue4",
```

```
    linewidth = 3)
```

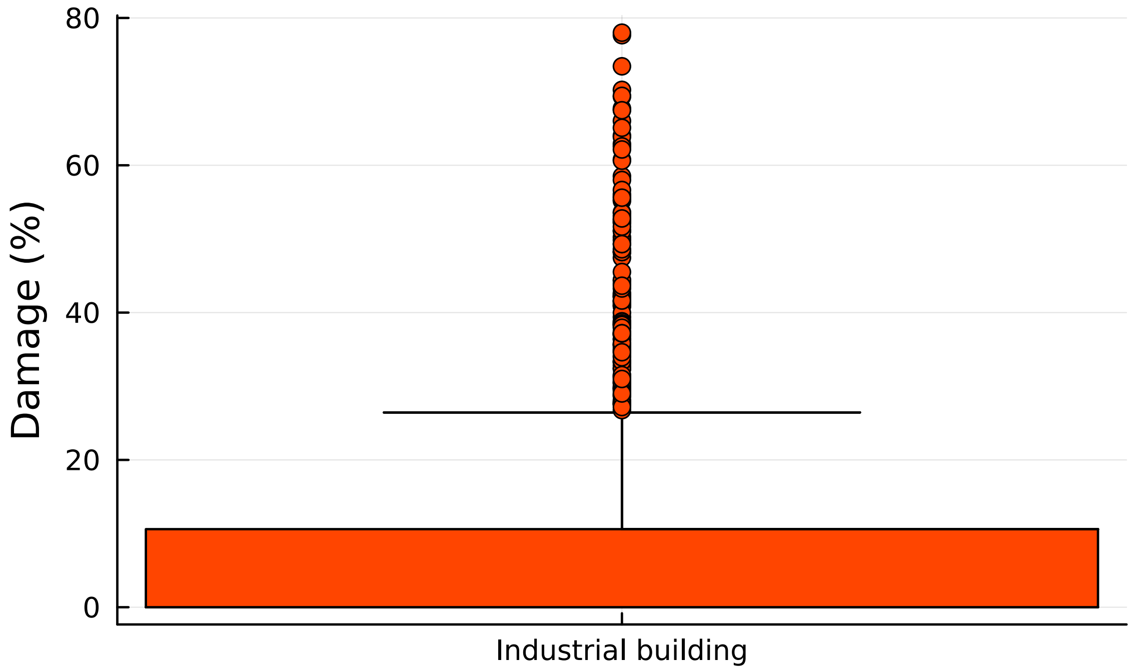


The damage distribution and average for 1000 samples from the flood depth hazard distribution are the following:

```
N_samples = rand(building_dist,1000)
N_depths = uconvert.(u"ft", (N_samples)u"ft")
plot(
    N_samples;
    xlabel = "Samples",
    ylabel = "Water elevation (ft)",
    label = "Random GEV distribution samples",
    legend = :outerbottom,
    size = (700, 500),
    color = "teal",
)
```




```
dmg_ind = dmg_fn_ind.(N_depths)

boxplot(
    ["Industrial building"],
    dmg_ind,
    ylabel = "Damage (%)",
    legend = false,
    color = "orangered",
)
```




```
expected_damage = mean(dmg_ind)
print("Industrial building mean damage: $(round(expected_damage,digits = 2)) (%)\n")
```

```
Industrial building mean damage: 8.71 (%)
```

2 Building value evaluation

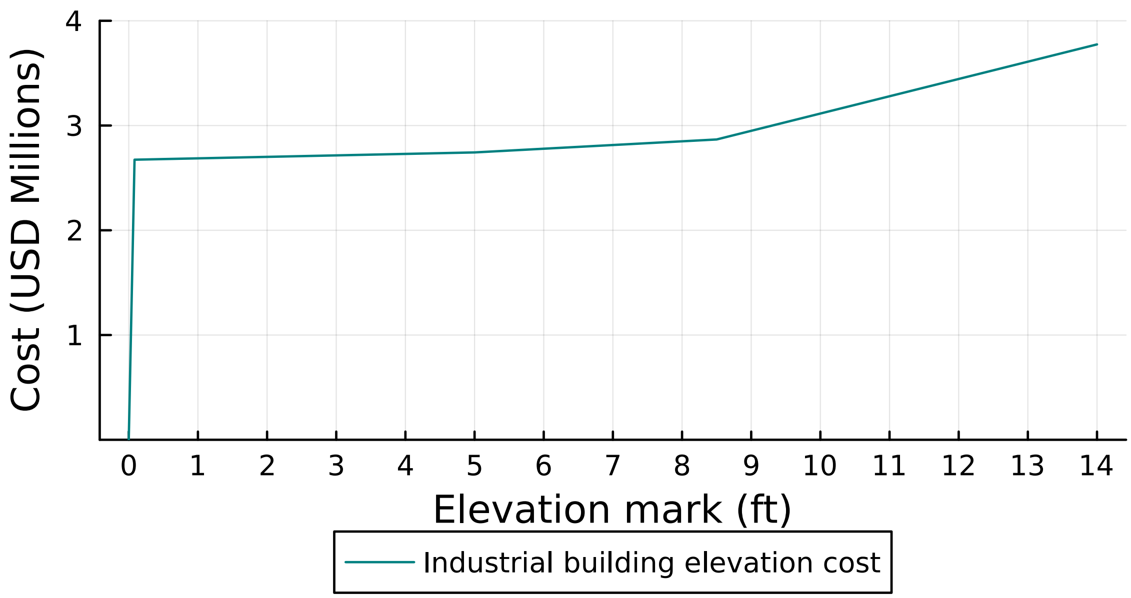
The value of the industrial building was calculated using a statistical average value per sqft available at APX construction group was $15 /sqft on average. To value the contents, a ratio of 10 is used (this value can be calibrated using data in the future). The expected damage valuation is calculated as follows:

```
unit_value = 15
bldng_L = 275
bldng_W = 120
cnt_str_ratio = 10
building_area = bldng_L * bldng_W
structurevalue = building_area  * unit_value
building_value = structurevalue * (1 + cnt_str_ratio)
building_area = (building_area)u"ft^2"
expec_dmg_usd = building_value * expected_damage / 100
```

```
474384.11699049675
```

The following correspond to the elevation cost valuation. This value can be also revise in the future for industrial buildings. For elevetions higher than 8 ft, the cost increments with a higher slope than for smaller elevations:

```
elevation_cost = get_elevation_cost_function()
heights = uconvert.(u"ft", (0u"ft"):(1u"inch"):(14u"ft"))
plot(
    heights,
    elevation_cost.(heights, building_area)/ 1e6;
    xlabel = "Elevation mark",
    xticks = (0u"ft"):(1u"ft"):(14u"ft"),
    yticks = 1:1:4,
    ylims = (0,4),
    ylabel = "Cost (USD Millions)",
    label = "Industrial building elevation cost",
    color = "teal",
    legend = :outerbottom,
)
```



3 Single year cost function

The following is the function for estimating the total cost in a given year that also includes the construction cost of elevating the building a given quantity. The flood cost corresponds to the expected damage to the building considering 1000 depth samples.

```
function single_year_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, building_area, bui
    N_samples = rand(building_dist,1000)
```

```
    N_depths = uconvert.(u"ft", (N_samples)u"ft")
    dmg_ind = dmg_fn_ind.(N_depths)
    expected_damage = mean(dmg_ind)
    c_dmg = building_value * expected_damage / 100

    c_constr = elevation_cost(Δh, building_area)

    return  - c_constr - c_dmg
end

single_year_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, building_area, building_val
```

```
-3.125440857585036e6
```

The following is the function for estimating the Net Present Value of different cost considering the flood depth cost and an initial construction cost of elevating the building. The discount ration is fixed for any year in the time frame.

```
function npv_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, building_area, building_va

    cost = ones(1,T)

    for t in (1:T)
        if t == 1
            cost[1,t] = single_year_cost_benefit(building_dist,dmg_fn_ind, elevation_cost, buil
        else
            cost[1,t] = single_year_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, bui
        end
    end

    NPV = sum([cost[1,t] * (1 - dscnt_rate)^(t - 1) for t in 1:T])

    return NPV
end
npv_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, building_area, building_value, 1u"
```

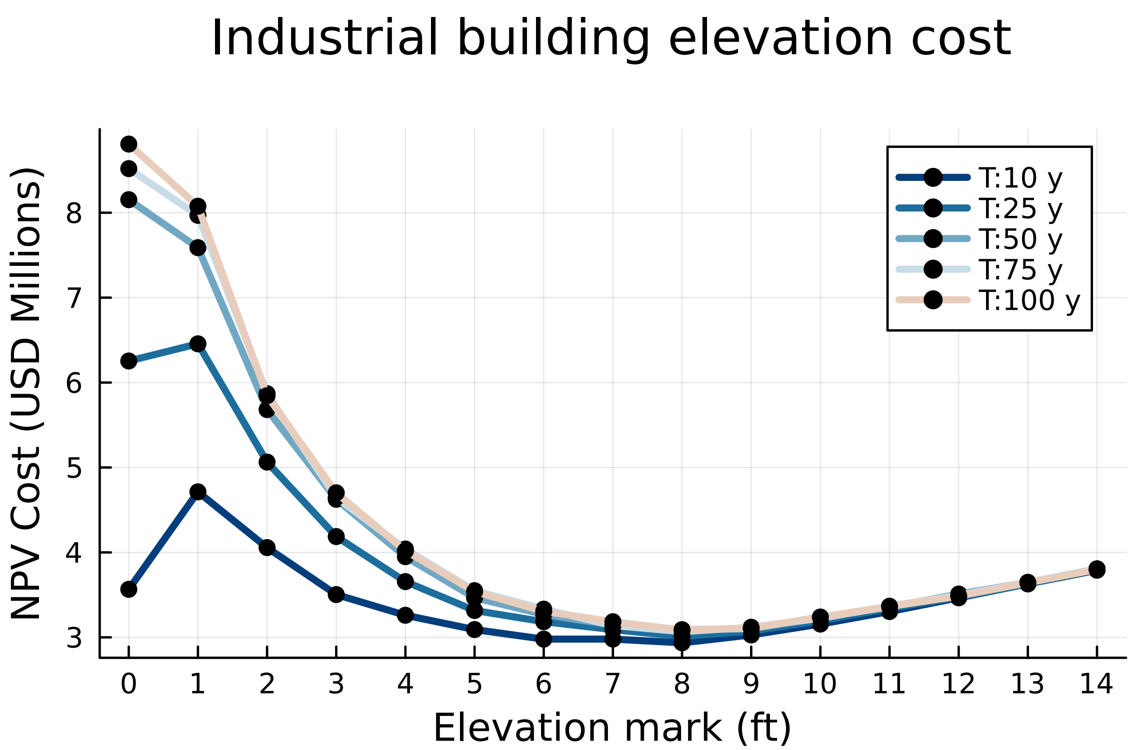
```
-3.564946263691749e6
```

4 NPV analysis

The following analysis considers the different actions (elevation height) and the NPV considering different time horizons (10 to 100 years). The figure shows 15 actions (0 - 14 ft elevations) and a discount rate of 5%.

```
Δh = (0:1:14)u"ft"
dscnt_rate = 0.05
T = [10,25,50,75,100]
NPV_SOW = ones(size(T)[1],size(Δh)[1])
p = plot()
```

```
for timeT in 1:size(T)[1]
    for height in 1:size(Δh)[1]
        offset = 12.29 - 6.24 + Δh[height] / 1u"ft"
        building_dist = GeneralizedExtremeValue(gauge_dist.  - offset, gauge_dist. , gauge_dist
        NPV_SOW[timeT,height] = npv_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, bu
    end
    p = plot!(
        Δh,
        -NPV_SOW[timeT,:]./1e6;
        xlabel = "Elevation mark",
        xticks = (0u"ft"):(1u"ft"):(14u"ft"),
        ylabel = "NPV Cost (USD Millions)",
        label = "T:$(T[timeT]) y",
        title = "Industrial building elevation cost",
        palette = :vik10,
        markershape = :circle,
        markercolor = :"black",
        linewidth = 3,
        legend = :topright,
    )
end
p
```

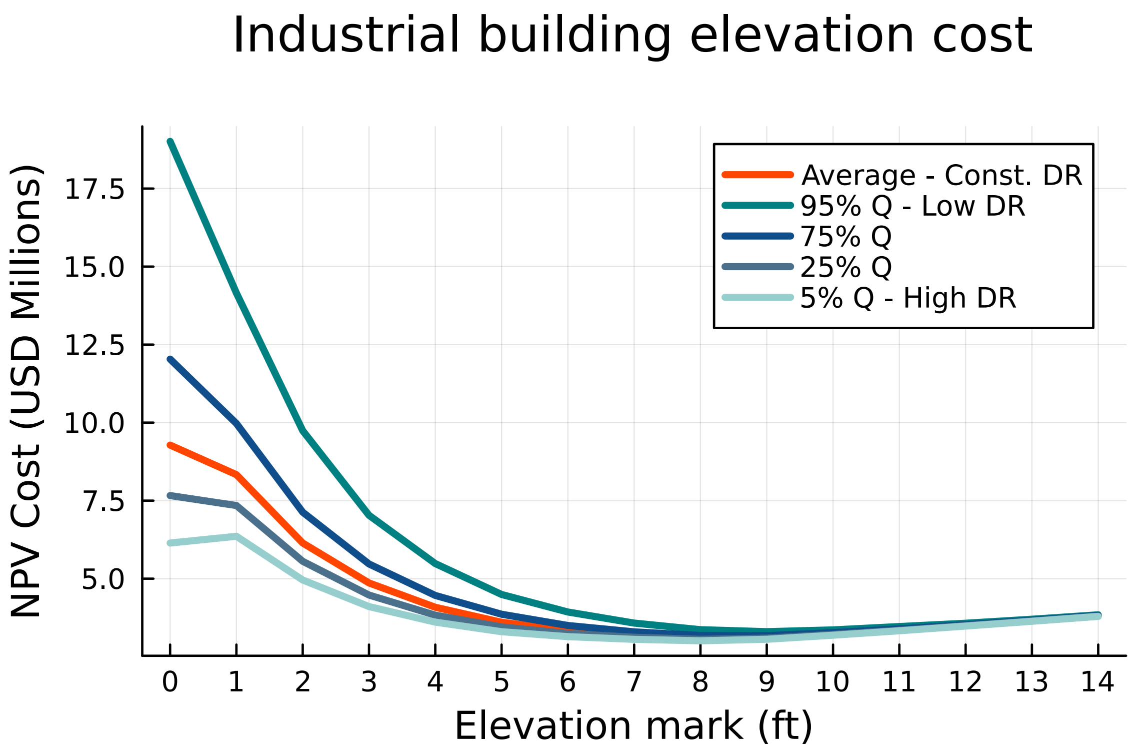


5 Discount rate

1. The first approach is to model a constant yet uncertain discount rate. The 50 year time-window is used for this analysis. 1000 thousand simulatios were used.

```
dscnt_dist = Normal(0.04,0.02)
N_dscnt_rate = rand(dscnt_dist,1000)
NPV_SOW = ones(size(N_dscnt_rate)[1],size(Δh)[1])
p = plot()
for dscntN in 1:size(N_dscnt_rate)[1]
    for height in 1:size(Δh)[1]
        offset = 12.29 - 6.24 + Δh[height] / 1u"ft"
        building_dist = GeneralizedExtremeValue(gauge_dist. - offset, gauge_dist. , gauge_dist
        NPV_SOW[dscntN,height] = npv_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, bu
    end
end
NPV_stats = ones(size(Δh)[1],5)
for height in 1:size(Δh)[1]
NPV_stats[height,1] = quantile(NPV_SOW[:,height],0.05)
NPV_stats[height,2] = quantile(NPV_SOW[:,height],0.25)
NPV_stats[height,3] = quantile(NPV_SOW[:,height],0.50)
NPV_stats[height,4] = quantile(NPV_SOW[:,height],0.75)
NPV_stats[height,5] = quantile(NPV_SOW[:,height],0.95)
end

p = plot!(
        Δh,
        -NPV_stats[:,3]./1e6;
        xlabel = "Elevation mark",
        xticks = (0u"ft"):(1u"ft"):(14u"ft"),
        ylabel = "NPV Cost (USD Millions)",
        title = "Industrial building elevation cost",
        color = "orangered",
        linewidth = 3,
        legend = :topright,
        label = "Average - Const. DR",
    )

p = plot!(
        Δh,
        -NPV_stats[:,1]./1e6;
        label = "95% Q - Low DR",
        color = "teal",
        linewidth = 3,
    )
p = plot!(
        Δh,
        -NPV_stats[:,2]./1e6;
```

```
        label = "75% Q",
        color = "dodgerblue4",
        linewidth = 3,
    )
p = plot!(
        Δh,
        -NPV_stats[:,4]./1e6;
        label = "25% Q",
        color = "skyblue4",
        linewidth = 3,
    )
p = plot!(
        Δh,
        -NPV_stats[:,5]./1e6;
        label = "5% Q - High DR",
        color = "paleturquoise3",
        linewidth = 3,
    )
p
```



2. Alternatively, the functions can be modified to generate a random discount rate for every year in the time horizon.

```
function npv_cost_benefit_VDR(building_dist, dmg_fn_ind, elevation_cost, building_area, buildi

```

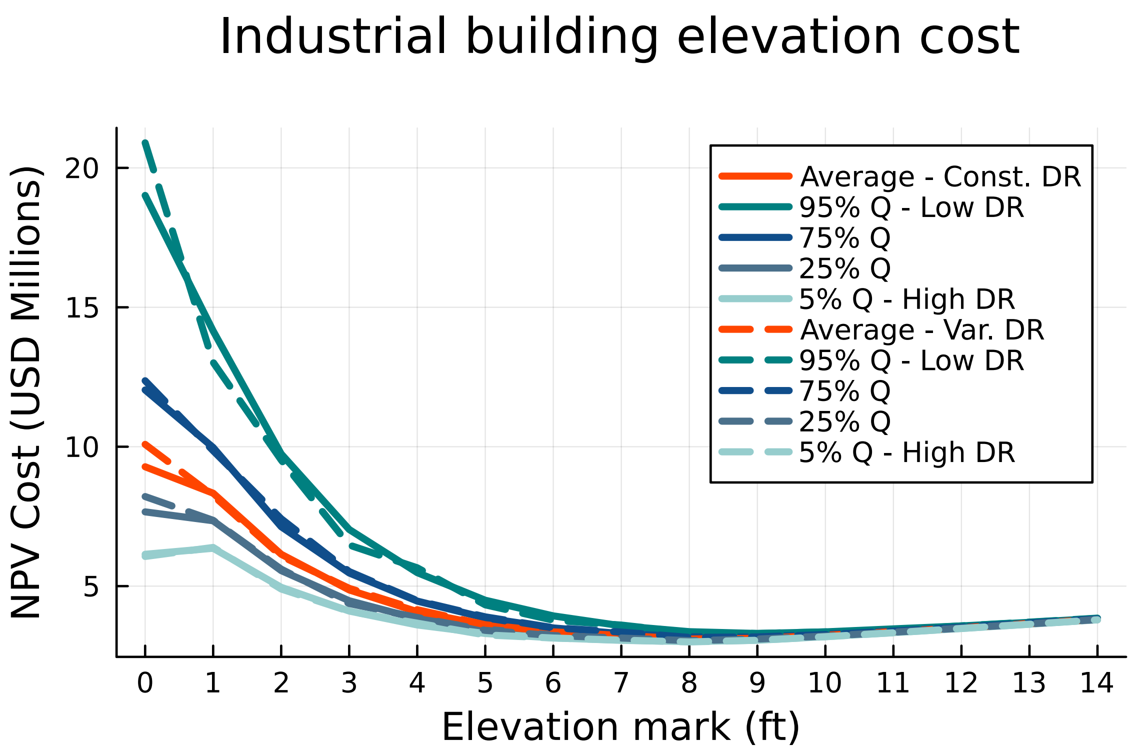
```
    cost = ones(1,T)
    dscnt_dist = Normal(0.04,0.02)

    for t in (1:T)
        if t == 1
            cost[1,t] = single_year_cost_benefit(building_dist,dmg_fn_ind, elevation_cost, bui
        else
            cost[1,t] = single_year_cost_benefit(building_dist, dmg_fn_ind, elevation_cost, bu
        end
    end
    N_dscnt_rate = rand(dscnt_dist,1)
    NPV = sum([cost[1,t] * (1 - N_dscnt_rate[1])^(t - 1) for t in 1:T])

    return NPV
end
```

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

In this case, 1000 realizations are performed by calling the new function.

```
NPV_SOW = ones(100,size(Δh)[1])
for realization in 1:100
    for height in 1:size(Δh)[1]
        offset = 12.29 - 6.24 + Δh[height] / 1u"ft"
        building_dist = GeneralizedExtremeValue(gauge_dist. - offset, gauge_dist. , gauge_dist
        NPV_SOW[realization,height] = npv_cost_benefit_VDR(building_dist, dmg_fn_ind, elevatio
    end
end
NPV_stats = ones(size(Δh)[1],5)
for height in 1:size(Δh)[1]
NPV_stats[height,1] = quantile(NPV_SOW[:,height],0.05)
NPV_stats[height,2] = quantile(NPV_SOW[:,height],0.25)
NPV_stats[height,3] = quantile(NPV_SOW[:,height],0.50)
NPV_stats[height,4] = quantile(NPV_SOW[:,height],0.75)
NPV_stats[height,5] = quantile(NPV_SOW[:,height],0.95)
end

p = plot!(
        Δh,
        -NPV_stats[:,3]./1e6;
        xlabel = "Elevation mark",
        xticks = (0u"ft"):(1u"ft"):(14u"ft"),
        ylabel = "NPV Cost (USD Millions)",
        title = "Industrial building elevation cost",
        color = "orangered",
        linewidth = 3,
        legend = :topright,
        label = "Average - Var. DR",
```

```
        linestyle = :dash,
    )

p = plot!(
        Δh,
        -NPV_stats[:,1]./1e6;
        label = "95% Q - Low DR",
        color = "teal",
        linewidth = 3,
        linestyle = :dash,
    )
p = plot!(
        Δh,
        -NPV_stats[:,2]./1e6;
        label = "75% Q",
        color = "dodgerblue4",
        linewidth = 3,
        linestyle = :dash,
    )
p = plot!(
        Δh,
        -NPV_stats[:,4]./1e6;
        label = "25% Q",
        color = "skyblue4",
        linewidth = 3,
        linestyle = :dash,
    )
p = plot!(
        Δh,
        -NPV_stats[:,5]./1e6;
        label = "5% Q - High DR",
        color = "paleturquoise3",
        linewidth = 3,
        linestyle = :dash,
    )
p
```



6 Discussion

6.1 What do you notice about the NPV for different actions?

The expected losses depend on the considered time window. For short time frames, not elevating the building can be the better choice and financially smart. In that case, the cost of elevation is too high in comparison to expected losses to be avoided. This is true when compared to elevating the building 1 to 3 ft. However, when the building is rise up to 4 ft, the NPV value is the same as the do-nothing option. For higher elevations (>5 ft) the NPV gets lower and for every ft in elevation the benefits marginally increase. Nevertheless, elevating the building beyond 8 ft becomes unpractical (Economically and technically) leading to a higher NPV.

In contrast, for larger time windows (>25 years), the benefit of elevating the building are more important and clearly outperform the do-nothing action. The expected losses due to flood events in a >25-year window is important (relative to the building value). The amount of elevation is also very important from one level to another. This is true up to a global NPV minimum at ~8 ft elevation. After this mark, the NPV starts rising again (although they still are better than the do-nothing action)

6.2 What do you notice about the sensitivity test?

The discount rate (DR) has a very impactful effect on the analysis. For a given time window, in this case, a 50-year time frame, the expected losses of not elevating the building are characterized with the largest variability. The NPV for do-nothing action can be 5 to 20 M$ for high to low discount rates, respectively. Nevertheless, the impact of the discount rate gets lower as the elevation actions increases in height. In those cases the NPV has smaller variations. This behavior response to the depth distribution where probabilities for depth larger than the elevation mark are lower.

Additionally, there is not much "analysis" difference when using multiple samples of *constant* DR values for every year of the analysis in comparison to sample a different DR for every year. Both models show very similar behavior. Nevertheless, this could be different if a DR function in time is given.

6.3 What are some limitations of this analysis?

1. Flood characteristics and time-evolving considerations can be incorporated.
2. The hazard modeling can be improved by using the available data for the gauge. This also can be addressed by perhaps changing or calibrating the probabilistic distribution or also using some synthetic/event scenarios with a given return period.
3. There is the lack of hydraulic model or flooding evolution considering the particularities of a given site which can potentially improve the real depth elevation, nevertheless, for near buildings with "natural" or unimproved terrain, the "offset" approach might be enough.
4. The economic valuation of the building and the elevation cost can be improved to better represent the particular building being analyzed. Probabilistic models can be used, or some level of uncertainty can be also assigned.