BEE 4750 Homework 3: Uncertain Sea-Level Rise and Levee Reliability

Name: Anthony Nicolaides

ID: ajn68

> **Due Date**
>
> Friday, 10/06/23, 9:00pm

Overview

Instructions

This assignment asks you to conduct a Monte Carlo analysis of levee reliability in the face of uncertain changes to local sea levels. You will propagate uncertainty in equilibrium climate sensitivity through the energy balance model to obtain a distribution of temperatures, which will then drive a model of sea-level rise. You will finally use this distribution to assess the probability that a planned levee will achieve its desired reliability standard.

Load Environment

The following code loads the environment and makes sure all needed packages are installed. This should be at the start of most Julia scripts.

```
In [ ]: import Pkg
        Pkg.activate(@__DIR__)
        Pkg.instantiate()
```

```
  Activating project at `~/Documents/BEE4750/hw/hw3-anthonynic28`
```

```
In [ ]: using Random
        using Plots
        using LaTeXStrings
        using Distributions
        using CSV
        using DataFrames
```

Problems (Total: 40 Points)

Problem 1 (12 points)

Recall from class that the simple energy balance model (EBM) of planetary energy balance links changes in radiative forcing (F) to global mean temperature (T) changes through the discretized equation

$$T_{i+1} = T_i + \frac{F_i - \lambda T_i}{cd} \times \Delta t,$$

where i is the current time step, $c = 4.184 \times 10^6$ J/K/m^2 is the heat capacity of water per unit area, d is the (uncertain) depth of the mixing layer, Δt is the annual time step in seconds and $\lambda = F_{\text{2xCO}_2}/S$ is the climate feedback parameter in W/m^2/$^\circ$ C, where S is the equilibrium climate sensitivity (the uncertain equilibrium temperature change resulting from a doubling of atmospheric CO$_2$). Finally, while total radiative forcing can be the result of non-aerosol and aerosol effects, we do not know the relative intensity of aerosol forcing, so we represent this with an uncertain aerosol scaling factor α.

We can implement this model with the following Julia function. We will assume an ocean mixing depth $d = 100$ m and an aerosol scaling factor $\alpha = 1.3$ so we can focus on the uncertainty in S.

The last technical concern is that "global mean temperature" does not make sense in absolute terms as a marker of climate change. Instead, we typically refer to temperature changes relative to some historical pre-industrial baseline. In this case, we will use the period from 1880-1900, though this choice can vary.

In [ ]:
```julia
# we need to split up the aerosol and non-aerosol forcings when we call the
function energy_balance_model(S, forcing_aerosol, forcing_non_aerosol)

    d = 100 # ocean mixing depth [m]
    α = 1.3 # aerosol scaling factor
    F2xCO₂ = 4.0 # radiative forcing [W/m²] for a doubling of CO₂
    λ = F2xCO₂ / S

    c = 4.184e6 # heat capacity/area [J/K/m²]
    C = c * d # heat capacity of mixed layer (per area)

    F = forcing_non_aerosol + α * forcing_aerosol # radiative forcing

    Δt = 31558152.0 # annual timestep [s]

    T = zero(F)
    for i in 1:length(F)-1
        T[i+1] = T[i] + (F[i] - λ * T[i]) / C * Δt
    end
    # return temperature anomaly relative to 1880-1900 baseline
    return T .- mean(T[1:21])
end
```
```
energy_balance_model (generic function with 1 method)
```

Finally, we need to load some radiative forcing data. We will use the radiative forcing scenario RCP 8.5. We can load this data, which is in a `.csv` (comma-delimited) file, into a `DataFrame`, which is a tabular data structure. Rows and columns in a `DataFrame` can be accessed using their numerical index (like a matrix), but columns also have names; you can access a particular column in a dataframe `df` by name using `df.colname` or `df[:, "colname"]`.

Of note: this data set goes from 1750–2500, so you will need to take care to make sure you are using the right years at each step. For example, here we will constrain the data to 1880–2100, which is the period we are interested in.

```
In [ ]: # The CSV is read into a DataFrame object, and we specify that it is comma d
        forcings_all_85 = CSV.read("data/ERF_ssp585_1750-2500.csv", DataFrame, delim

        # get the years corresponding to the forcings
        t = Int64.(forcings_all_85[!, "year"]) # Ensure that years are interpreted a
        # find the indices of the years 1880 and 2100
        # we can do this with the indexin function
        time_bounds = indexin([1880, 2100], t)
        years = time_bounds[1]:time_bounds[2] # create range of years

        # Separate out the individual components
        forcing_co2_85 = forcings_all_85[years, "co2"]
        # Get total aerosol and non-aerosol forcings
        forcing_aerosol_rad_85 = forcings_all_85[years, "aerosol-radiation_interacti
        forcing_aerosol_cloud_85 = forcings_all_85[years, "aerosol-cloud_interaction
        forcing_aerosol_85 = forcing_aerosol_rad_85 + forcing_aerosol_cloud_85 # aer
        forcing_total_85 = forcings_all_85[years, "total"]
        forcing_non_aerosol_85 = forcing_total_85 - forcing_aerosol_85 # non-aerosol
```

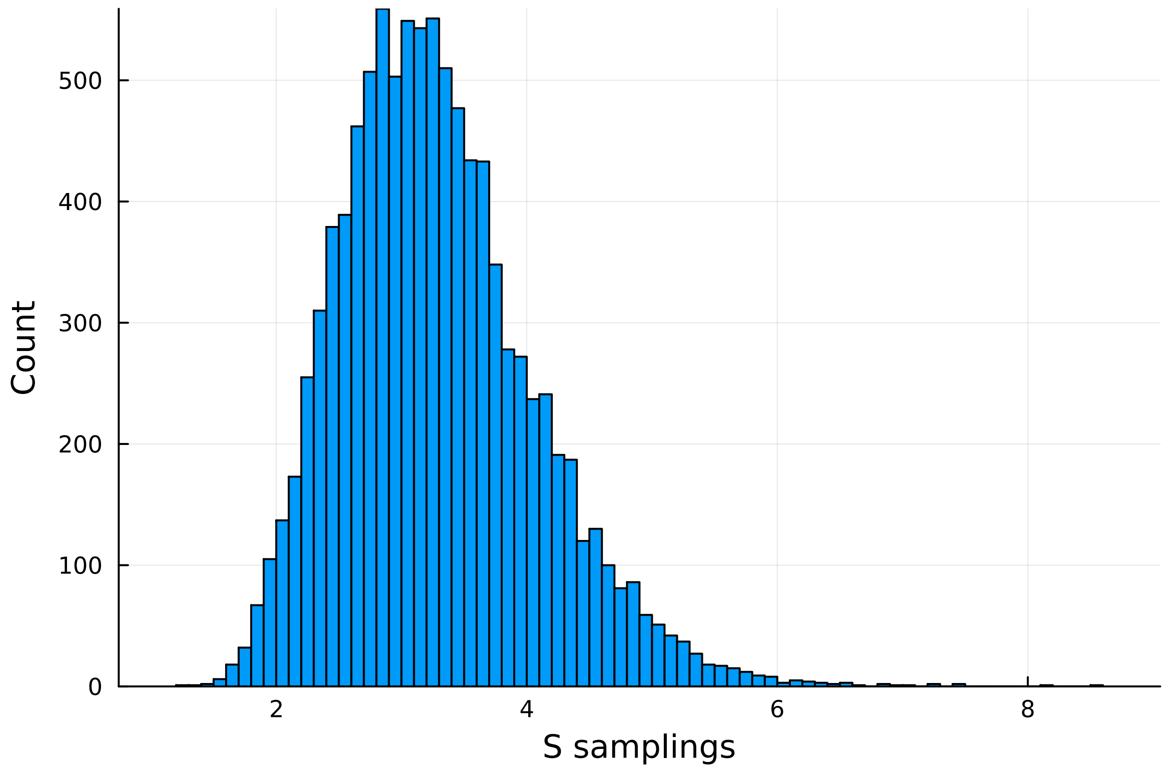
```
221-element Vector{Float64}:
  0.42741479112315905
  0.4487940147601447
  0.4900144276528058
 -0.019811270078689047
 -1.480725700367619
 -0.3154905163510021
  0.233186559774844
  0.2923308952663089
  0.45833872365810924
  0.5168189581089915
  ⋮
  9.879065698371564
  9.946850640754889
 10.002169546141578
 10.061009359425011
 10.115684195707905
 10.182946497594184
 10.249699157847772
 10.331407709334023
 10.424544495740134
```

For this assignment, you can use the `forcing_aerosol_85` and `forcing_non_aerosol_85` vectors as is to correspond to the relevant forcings. You will need to use the vector `t` to find the appropriate years for analysis.

Problem 1.1 (3 points)

Assume that S is distributed according to $\mathrm{LogNormal}(\log(3.2), \log 2/3)$ (as in class). Draw 10,000 samples from this distribution and make a histogram.

In [ ]:
```
# create distribution for ECS
S_dist = LogNormal(log(3.2), log(2) / 3)
Random.seed!(1) # get consistent output from random sampling
samples = 10000
S = rand(S_dist, samples) # array of 10000 samples
histogram(S,
    legend=:false,
    label="Sampling from S Distribution",
    xlabel="S samplings",
    ylabel="Count")
```

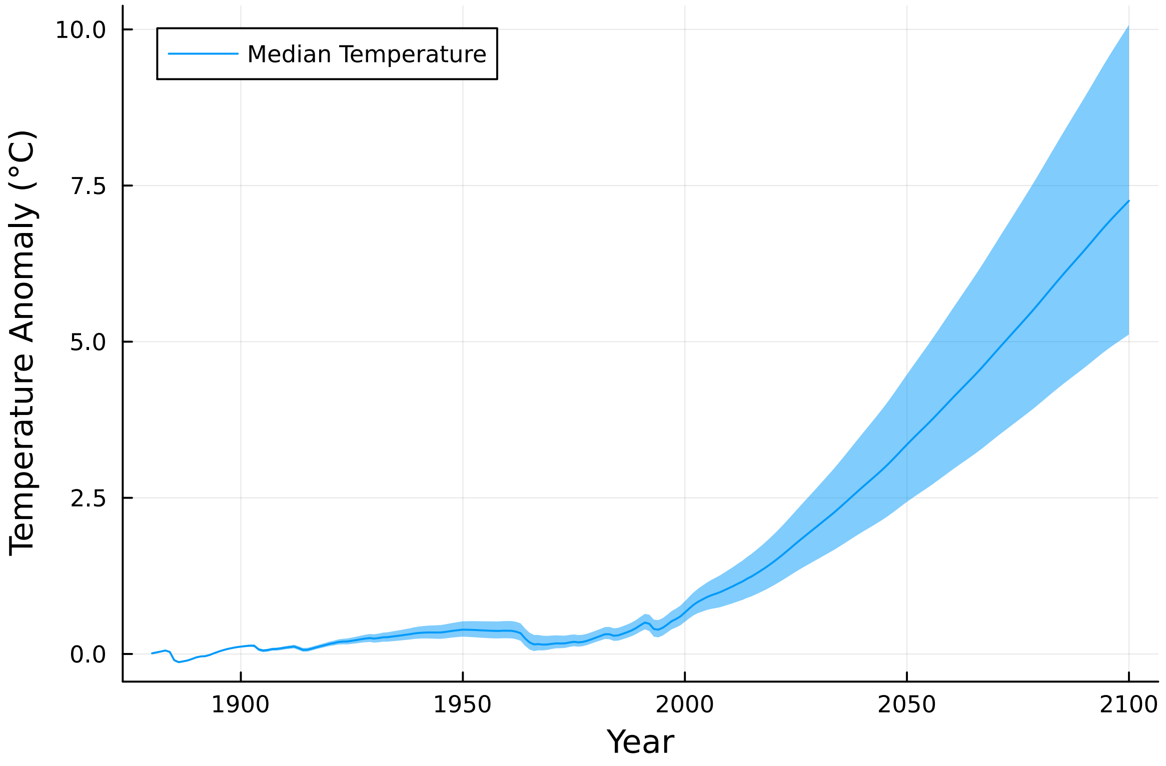


Problem 1.2 (5 points)

Use the EBM to propagate your samples of S to a distribution of global mean temperature. Plot the median and 90% predictive distribution (between the .05 and .95 quantiles) from 1880-2100.

In [ ]:
```
yearRange = 1880:2100
ebm_output = zeros(samples, length(yearRange))
```

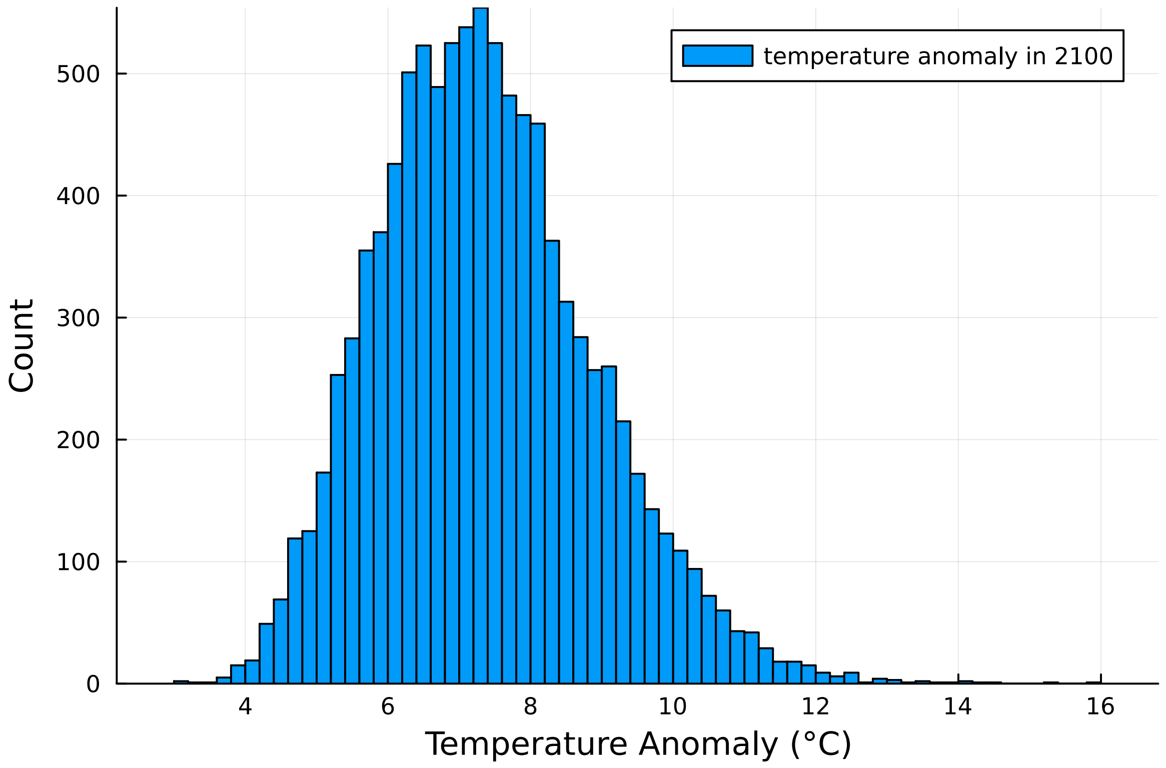
```julia
for n = 1:samples
    # rows = temp anomaly, columns = year
    ebm_output[n, :] = energy_balance_model(S[n],
        forcing_aerosol_85,
        forcing_non_aerosol_85)
end

# find median
temp_median = quantile.(eachcol(ebm_output), 0.5)

# find 5% interval
temp_quantile_5 = quantile.(eachcol(ebm_output), 0.05)

# find 95% interval
temp_quantile_95 = quantile.(eachcol(ebm_output), 0.95)

# plot with 90% confidence projection
plot(yearRange, temp_median,
    ribbon=(temp_median - temp_quantile_5,
        temp_quantile_95 - temp_median),
    label="Median Temperature",
    legend=:topleft,
    xlabel="Year",
    ylabel="Temperature Anomaly (°C)")
```



Problem 1.3 (4 points)

Make a histogram of global mean temperature projections in 2100. If you compare this distribution to the distribution of S from Problem 1.1, what do you observe?

```
In [ ]: # index 221 is the last index of the columns/years so it is the year 2100
        ebm_output_2100 = ebm_output[:, 221] # temp anomalies in 2100
        histogram(ebm_output_2100,
            label="temperature anomaly in 2100",
            xlabel="Temperature Anomaly (°C)",
            ylabel="Count")
```



Comparing the graph of the global mean tmperature projections in 2100 to the S distribution from Problem 1.1, you can see that the shape of the distributions are pretty similar. This suggests that the smaller the ECS, the less of a temperature anomaly and the higher the ECS, the greater the temperature anomaly. Futhermore, the mean value of the S distribution (~3) is less than this distribution (~7), so using the mean S value will output a temperature anomaly of about 7 C.

Problem 2 (15 points)

Changes to global temperatures cause changes in global sea levels through several mechanisms, including thermal expansion (the change in ocean volume due to increased heat content) and melting land-based ice. One simple way to represent this link is through the following model, proposed by Rahmstorf (2007).

$$\frac{dH}{dt} = a(T - T_0),$$

where H is the global mean sea level in mm, T is global mean temperature, T_0 is an equilibrium temperature (where there is no change in sea levels), and a is a proportionality constant. This model can be discretized to give

$$H_{i+1} - H_i = a(T_i - T_0).$$

Note that, like with global mean temperature, the notion of "global mean sea level" does not make sense in absolute terms (were sea levels ever at "zero"?). Instead, we want to normalize this relative to some historical baseline. In this case (with a view towards Problem 3), we will compute our sea levels relative to the 2010 sea level. Note that in addition to the model parameters, we also need an initial sea-level parameter H_0 which will give us the right anomaly level.

The best estimates for these parameters are:

- $a = 1.86$;
- $H_0 = -223$;
- $T_0 = -0.62$

Problem 2.1 (5 points)

Write a function `sea_level_model()` to implement the mathematical sea-level rise model described above. It should take in needed parameters and a vector of temperatures and return a vector of sea levels. To test your function, use the provided temperature series `historical_temps` (read in below) to compute the global mean sea level anomaly in 2022 (the last year of the dataset) with the parameter values above; you should get a value of approximately 40mm.

In [ ]:
```
historical_temp_data = CSV.read("data/HadCRUT.5.0.1.0.analysis.summary_serie
# column 2 is the temperature anomaly, column 1 is the year
temp_bds = indexin([1880, 1900], historical_temp_data[!, 1]) # find the inde
historical_temp_data[:, 2] .-= mean(historical_temp_data[temp_bds[1]:temp_bc
historical_temps = historical_temp_data[temp_bds[1]:end, 2]
```

```
143-element Vector{Float64}:
  0.07297116761904765
  0.15655772761904765
  0.09327316761904769
  0.04232884761904765
 -0.10351681238095234
 -0.08232033238095232
 -0.03210037238095231
 -0.10998251238095236
  0.00942435761904764
  0.13890768761904768
  ⋮
  1.0616748976190478
  1.2139176176190476
  1.3217303776190477
  1.2339774976190476
  1.1514572476190477
  1.2798758476190477
  1.3115970476190477
  1.1506591476190478
  1.1896871476190476
```

In [ ]:
```julia
function sea_level_model(a, H0, T0, T)
    # H(i+1) = a(T(i) - T0) + H(i)
    # H(i) = a(T(i-1) - T0) + H(i-1)
    H = zeros(length(T) + 1)

    # add inital condition to array to make for loop simple
    H[1] = a * (T0 - T0) + H0 # i = 1
    for i = 2:length(H)
        H[i] = (a * (T[i-1] - T0)) + H[i-1]
    end
    H = H[2:end] # remove initial condition from array
    # only return sea levels associated with the given T array
    return H # vector of sea levels (mm)
end
```

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

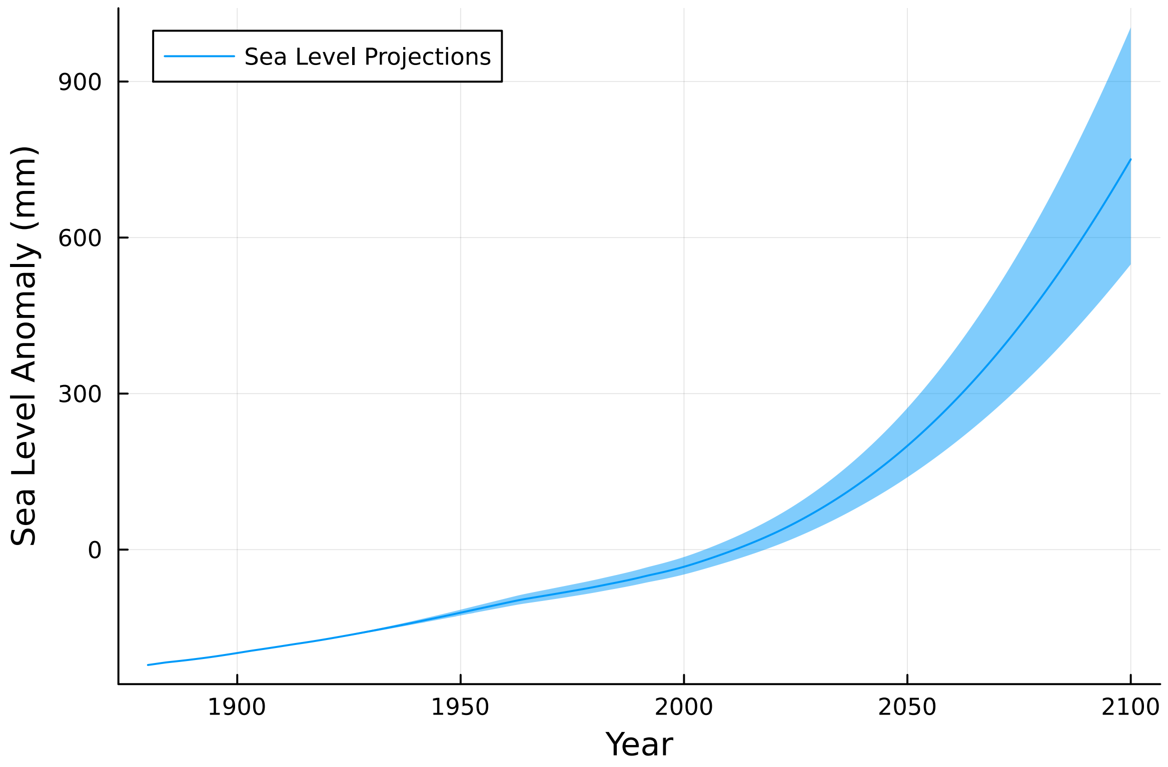
In [ ]:
```julia
# test function
sea_levels = sea_level_model(1.86, -223, -0.62, historical_temps)
println("Global mean sea level anomaly in 2022 is ",
    round(sea_levels[end], digits=2), "mm")
```

```
Global mean sea level anomaly in 2022 is 38.54mm
```

Problem 2.2 (5 points)

Evaluate `sea_level_model()` using the projected temperature ensemble from Problem 1. Plot the 90% projection interval of the sea levels.

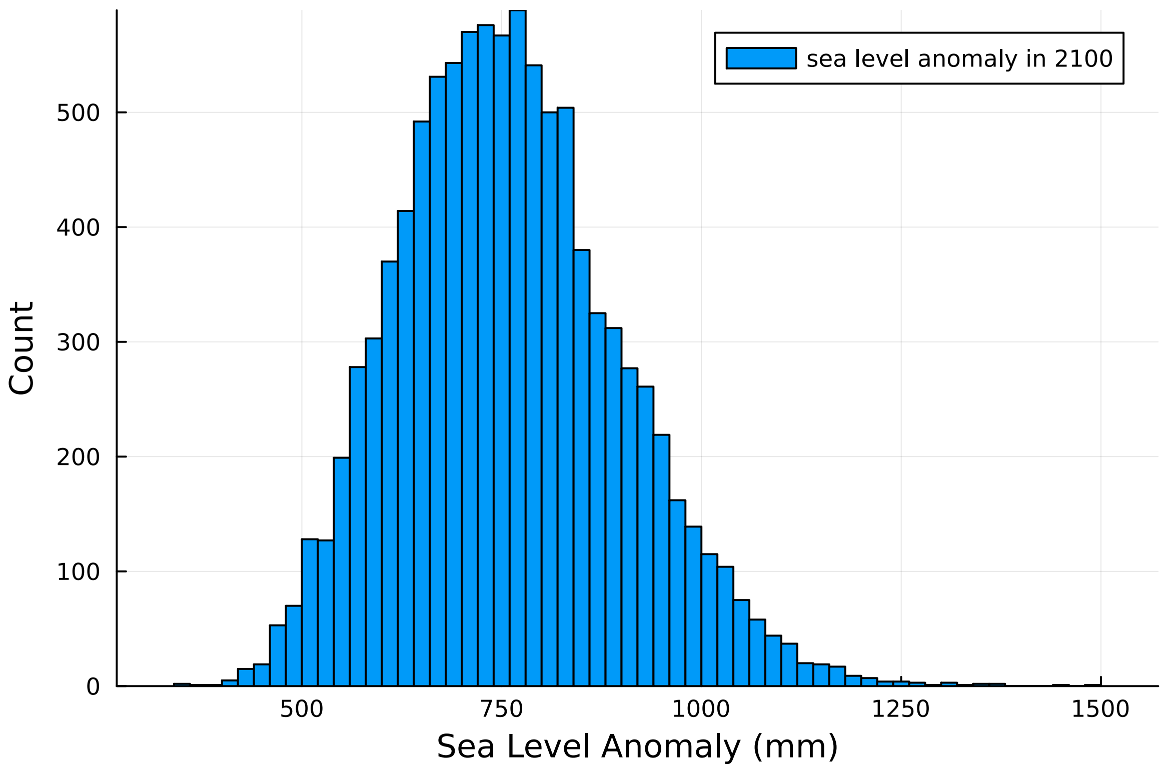
In [ ]:
```julia
sea_levels_ebm = zeros(samples, 221)
for n = 1:samples
    # rows = sea level anomaly, columns = year
    sea_levels_ebm[n, :] = sea_level_model(1.86,
        -223,
        -0.62,
        ebm_output[n, :])
end

# find median
sea_levels_median = quantile.(eachcol(sea_levels_ebm), 0.5)

# find 5% interval
sea_levels_quantile_5 = quantile.(eachcol(sea_levels_ebm), 0.05)

# find 95% interval
sea_levels_quantile_95 = quantile.(eachcol(sea_levels_ebm), 0.95)

# plot with 90% confidence projection
plot(yearRange, sea_levels_median,
    ribbon=(sea_levels_median - sea_levels_quantile_5,
        sea_levels_quantile_95 - sea_levels_median),
    label="Sea Level Projections",
    legend=:topleft,
    xlabel="Year",
    ylabel="Sea Level Anomaly (mm)")
```



Problem 2.3 (5 points)

Make a histogram of the sea-level anomaly in 2100. What can you observe about how the ECS uncertainty has impacted sea-level uncertainty under this radiative forcing scenario? What might the implications be of only using the best-estimate ECS value?

In [ ]:
```
# last year index (221) is for 2100
histogram(sea_levels_ebm[:, 221],
    label="sea level anomaly in 2100",
    xlabel="Sea Level Anomaly (mm)",
    ylabel="Count")
```



The ECS and sea level histograms are very similar in shape. This suggests they are directly proportional to one another, in other words the smaller the ECS, the less of a sea level anomaly and the higher the ECS, the greater the sea level anomaly. Using the mean ECS value will suggect a sea level anomaly of about 750 mm.

Problem 3 (13 points)

You've been asked to consult on a levee reliability analysis. For context, levees in the United States are supposed to only fail once in 100 years, or, in other words, to have at most a 1% chance of failure in a given year. We will assume for this problem that the only way in which a levee fails is by being overtopped (note: this is unrealistic).

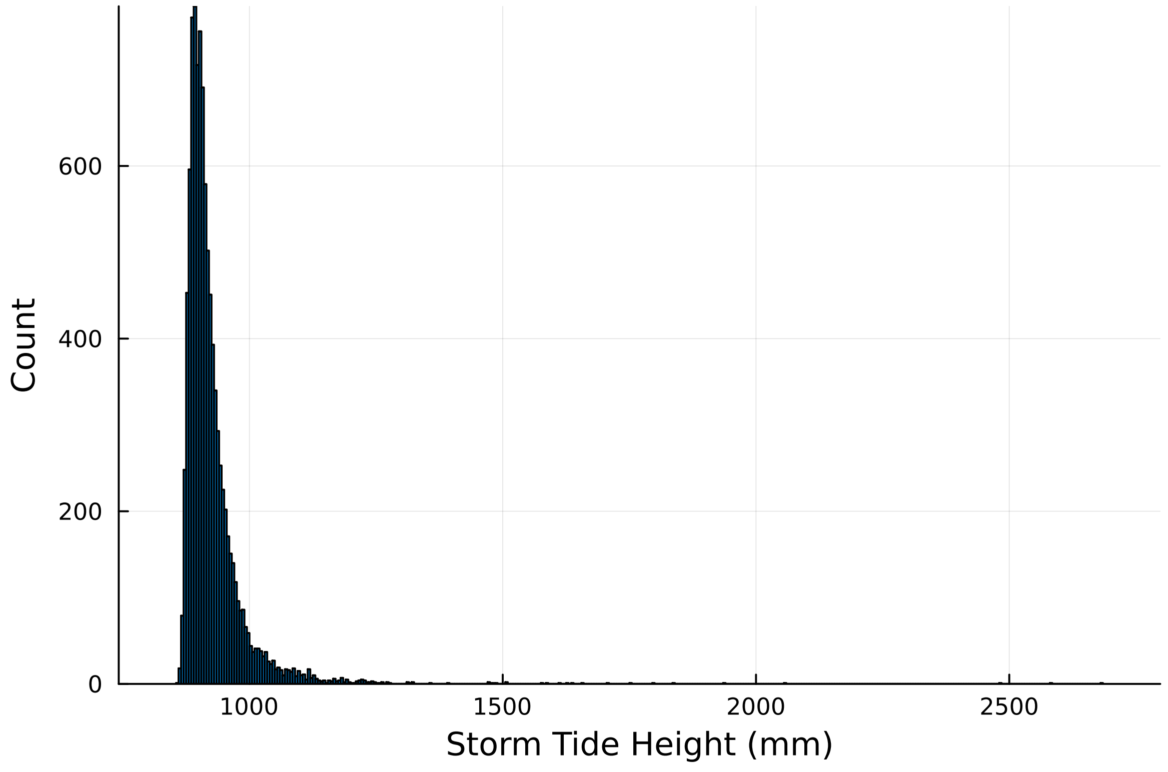
We can assess the probability of levee overtopping by comparing its height to a distribution of extreme sea levels. A common approach is to look at the distribution of the highest sea level each year. These extreme sea levels can be obtained by combining the absolute sea level (we will use our distribution of global sea levels for this), the rate of subsidence (how much the ground sinks), and the distribution of storm tides (the highest tide level, which is often the result of storm surges combining with high tide).

Assume for this problem that:

1. the annual rate of subsidence ν is 1.2mm/yr;

2. the distribution of annual storm tide maxima, above the mean sea level, is (and is expected to continue to be) given by a $\text{GeneralizedExtremeValue}(900, 25, 0.3)$ distribution, which looks like this:

In [ ]:
```
tide_distribution = GeneralizedExtremeValue(900, 25, 0.3)
histogram(rand(tide_distribution, 10000), xlabel="Storm Tide Height (mm)", y
```



Feel free to just sample from `tide_distribution` in your solution below.

Problem 3.1 (2 points)

How would you use your sea-level simulations and the above information to compute a distribution of extreme sea levels in 2100 relative to 2010 mean sea level? Write down the approach in clear steps, with equations as needed.

1. To create a distribution of extreme sea levels in 2100, use the sea level model to get the sea level anomalies in 2100 (see Problem 2.3).

2. Subtract the annual rate of subsidence, 1.2mm/yr, from the sea level anomalies.

3. Then add the storm tide height to the sea level anomalies.

4. After naming the varibale, this gives the equation:

   seaLevel = 2100 sea levels distribution

   ν = annual rate of subsidence, 1.2mm/yr tide = Storm tide height (sampled from distribution)

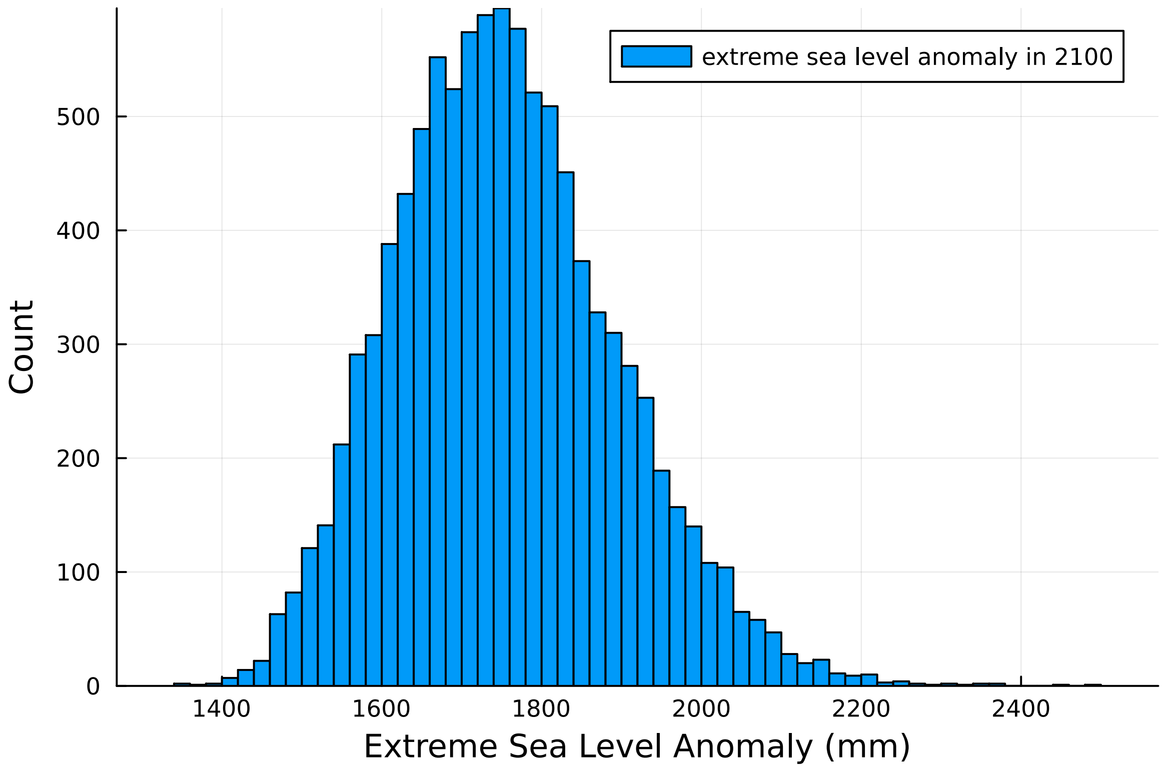
$$ExtremeSeaLevels = seaLevel - \nu + tide$$

5. This will result in a distribution of extreme sea levels in 2100.

Problem 3.2 (3 points)

Follow the steps above and produce a histogram of the extreme sea levels in 2100 relative to 2010.

In [ ]:
```
# step 1
sea_levels_2100 = sea_levels_ebm[:, 221] # last year index (221) is for 2100

# get variable for step 2
v = 1.2 # mm/yr (is converted to mm by multipling it by 1 year)

# get variable for step 3
Random.seed!(1) # get consistent output from random sampling
tide_sample = rand(tide_distribution) # storm tide height sample

# step 4
extreme_sea_levels_2100 = sea_levels_2100 .- v .+ tide_sample

# step 5
histogram(extreme_sea_levels_2100,
    label="extreme sea level anomaly in 2100",
    xlabel="Extreme Sea Level Anomaly (mm)",
    ylabel="Count")
```



Problem 3.3 (5 points)

The current levee was heightened in 2010 to 2m above the 2010 mean sea level. Based on your analysis above, what is the probability that the levee will be overtopped in 2100 (remember that the reliability standard is 1%)?

```
In [ ]: # levee will only be overtopped if extreme sea level is over 2000mm
        levee_2m = [x for x in extreme_sea_levels_2100 if x > 2000]

        # probability is the area of the distribution that is greater than 2000, but
        #   this needs to be converted to a probability, so divide it by total area
        prob_of_overtopping = length(levee_2m) / length(extreme_sea_levels_2100)
        println("The probability of the 2m levee being overtopped in 2100 is ",
            prob_of_overtopping)
```

```
The probability of the 2m levee being overtopped in 2100 is 0.0502
```

Problem 3.4 (3 points)

Based on your analysis, would you recommend that the levee be heightened again in the future, and if so, how high? What other information might you need, if any, to make your recommendation?

Based on the information currently known, the levee does not need to be heightened, as a height of 2m gives a 0.5% chance of failure, which is lower that the reliabliity standard of 1% chance of failure. Other information needed to give more confidence in my recommendation would be erosion rates of the levee (e.g. if the levee's erosion rate is fast, then by 2100 it can potentially not be up the standard of 1% chance of failure). Furthermore, more information about the specific structure of the levee would also be helpful (i.e. is the structural integrity up to standard, is there regular maintance, any signs of deterioration/wear-and-tear, etc.), these details could increase/decrease the risk of failure, when taken into account.

References

List any external references consulted, including classmates.

BEE 4750 9/22 Lecture "Simple Climate Models and Uncertainty Propagation" Slides & provided code (for graphing 90% confidence)