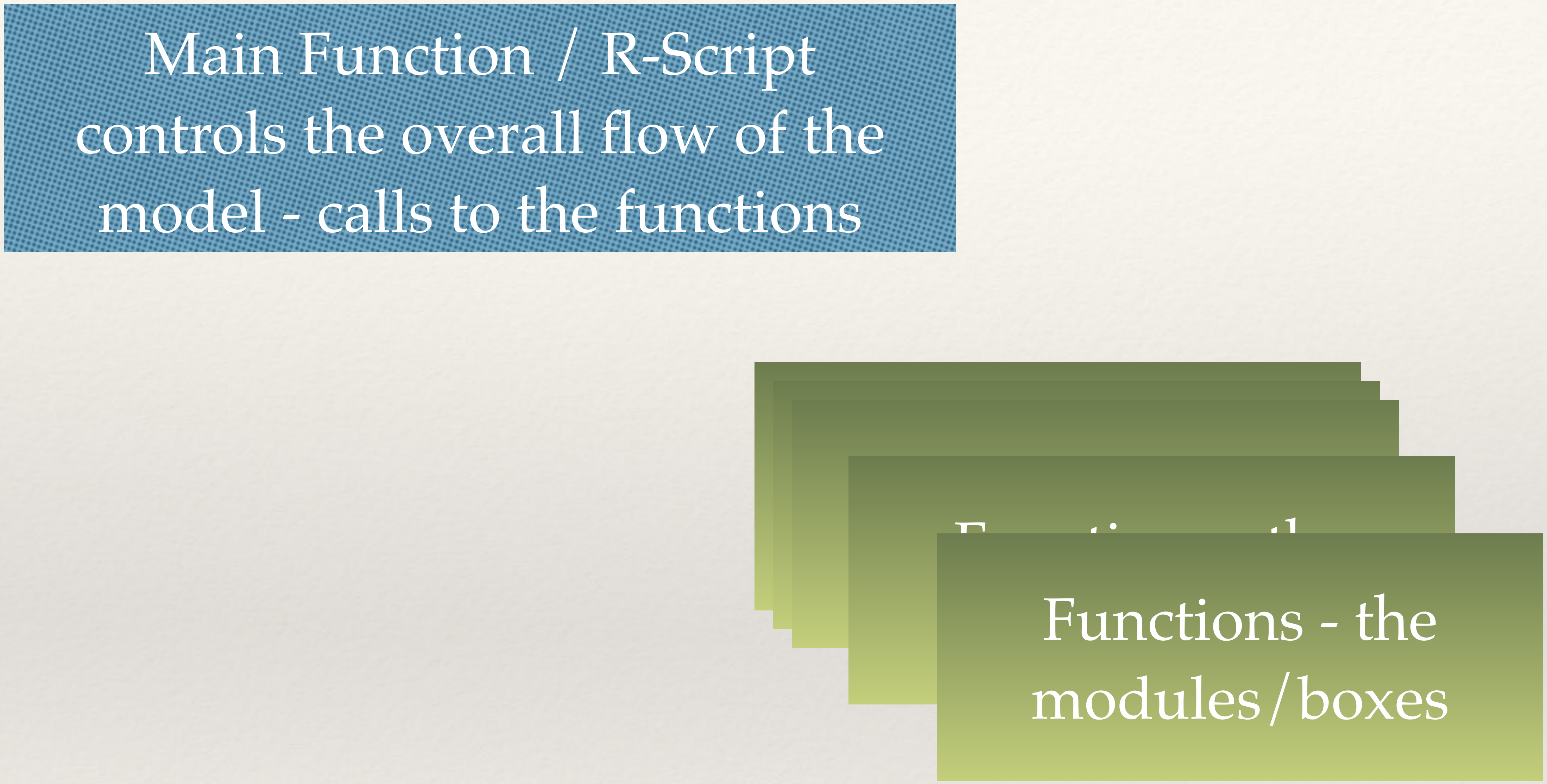
Building Models



- A model is often multiple pieces put together

A Cost-Benefit Analysis of Adaptations to Climate Change

Comparing ecosystem-based adaptation versus engineered approaches to tropical storm variability

A Project at the Bren School of Environmental Science & Management, UCSB April 2012

Sarah Clark, Teo Grossman, Nick Przyuski, Cassidee Shinn, Danielle Storz Advised by: Naomi Tague Client: Conservation International

Example: Illustrate Modularity and the Use of Data Structures

- Group Project
- *Problem*: TNC needs to be able to argue that ecosystem based adaptation can be cost effective - don't have cost estimates
- *Model Goal* -
  - Compare the benefits of ecosystem based adaptation (specifically mangroves) with the benefits of engineering based climate adaptation (sea walls) for coastal flood protection in Fiji
  - Time Frame (next 50 years)
- *Output* - Costs and Benefits of 3 approaches Mangroves, Sea Walls and Do Nothing (no adaptation) - in 2010 dollars

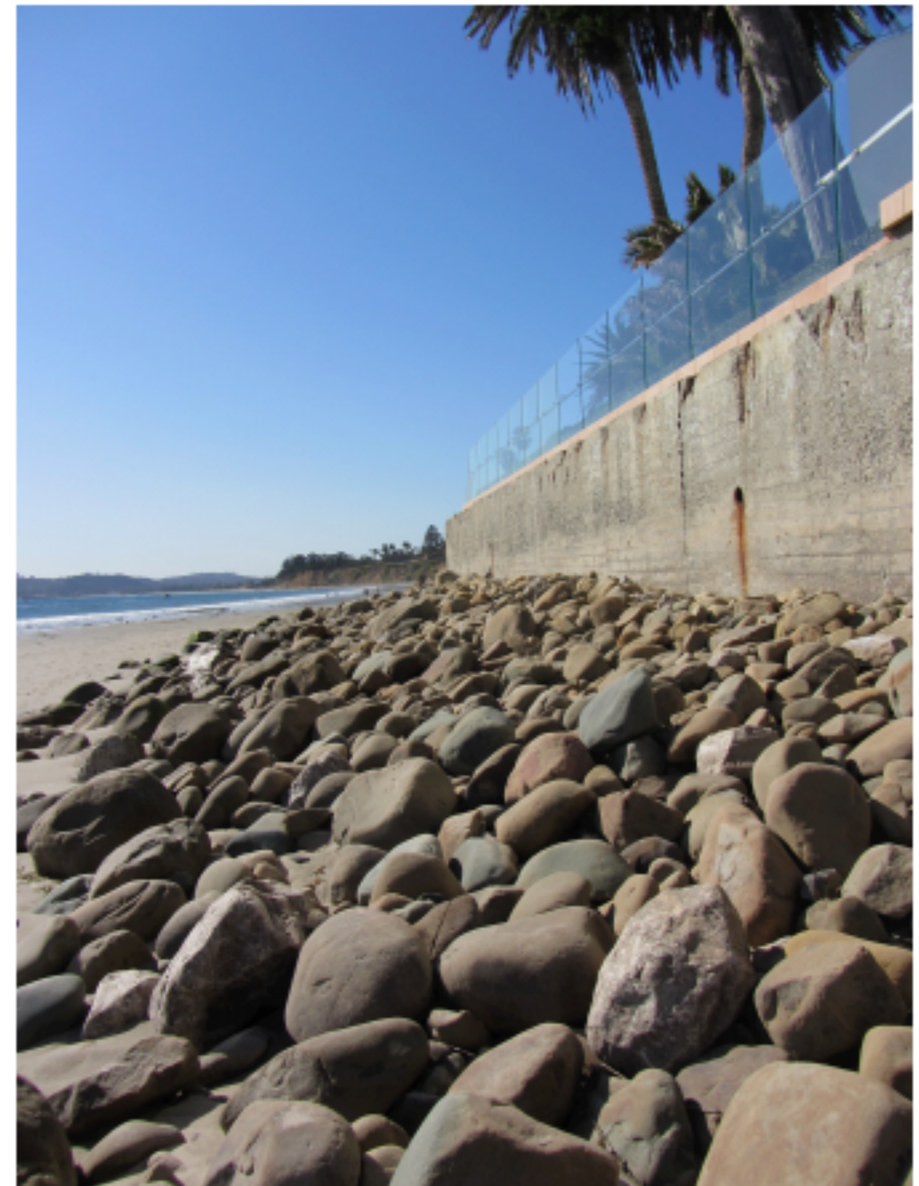
We are just going to look at relative costs….(to simplify for the class)

Mangroves vs Seawalls

The nation of Fiji has the third most extensive mangrove forest cover of all the South Pacific islands. One of the largest mangrove areas is on the southeast coast of Viti Levu and serves as our model of an EbA option.

Mangrove and Seawall both effectively reduce surge height

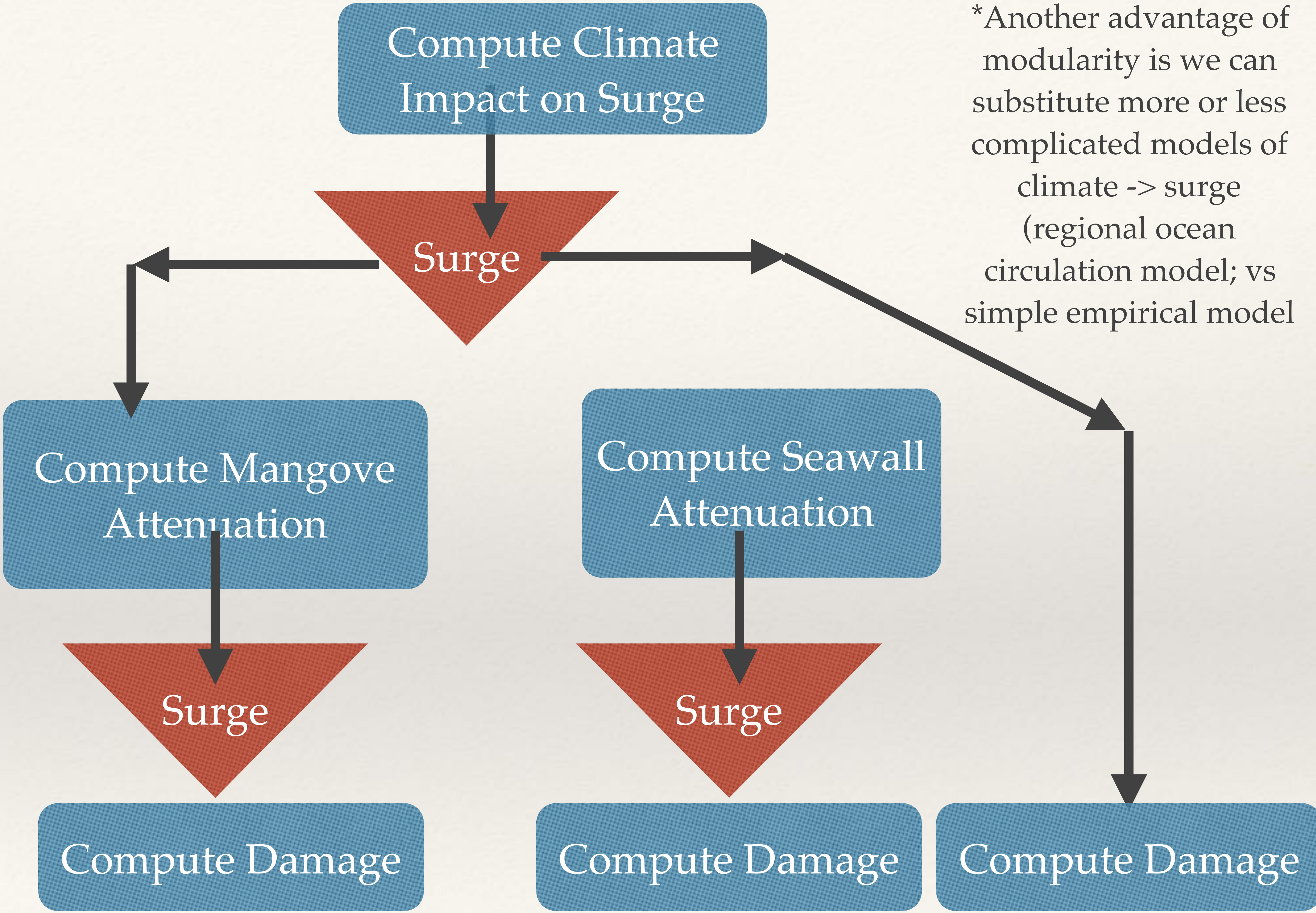
Assume that surge height can be related to damage from storms (makes intuitive sense; and local data support this assumption)



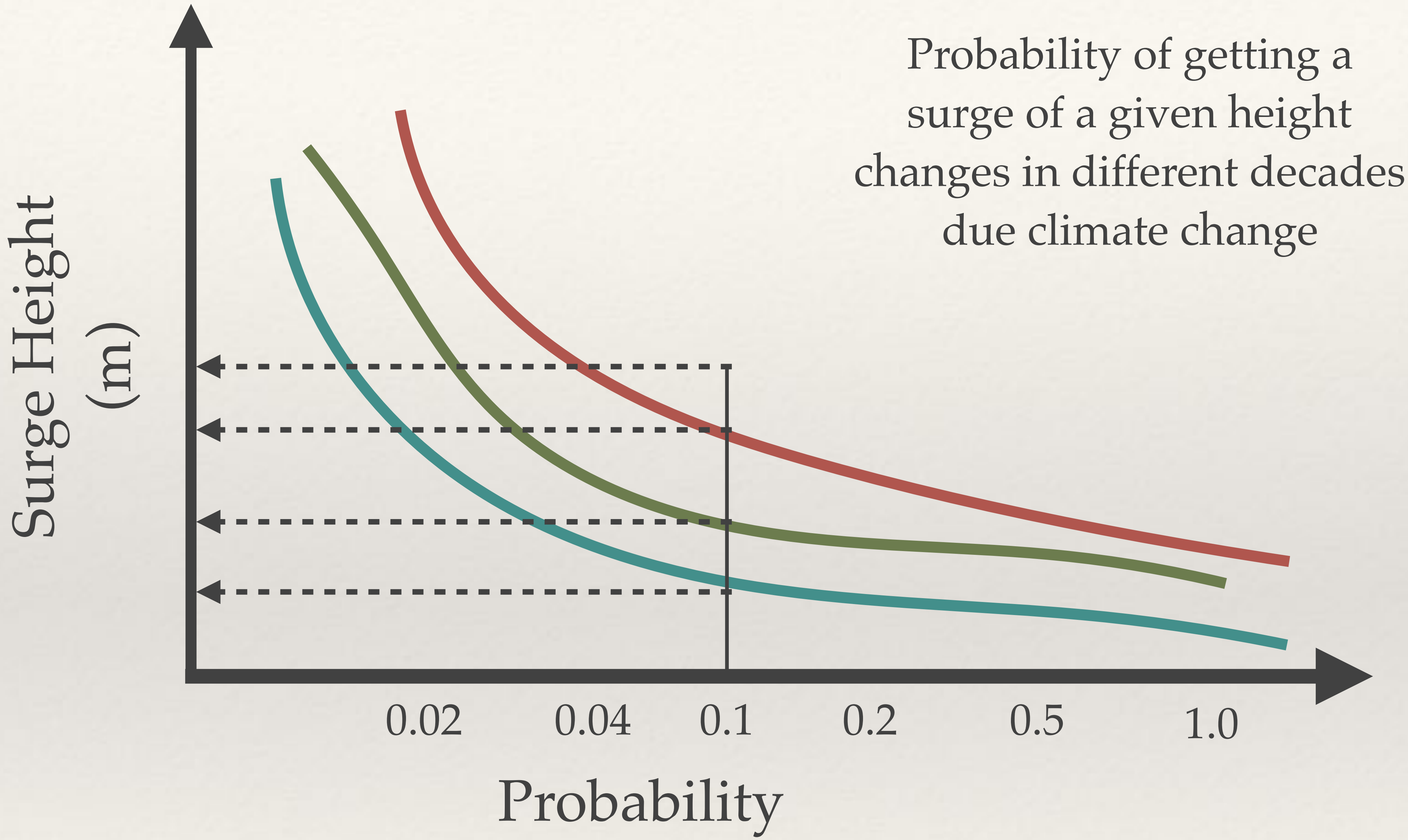
We used the construction of a seawall as our engineered-based adaptation approach to storm surge inundation. Seawalls protected 51% of the 11.8 miles of coastline surveyed on the Suva Peninsula in 1996.

HTTP://FIESTA.BREN.UCSB.EDU/~ADAPTATION
ADAPTATION@LISTS.BREN.UCSB.EDU

*Another advantage of modularity is we can substitute more or less complicated models of climate -> surge (regional ocean circulation model; vs simple empirical model



Storm Surge Model: Stochastic (Return Probability)

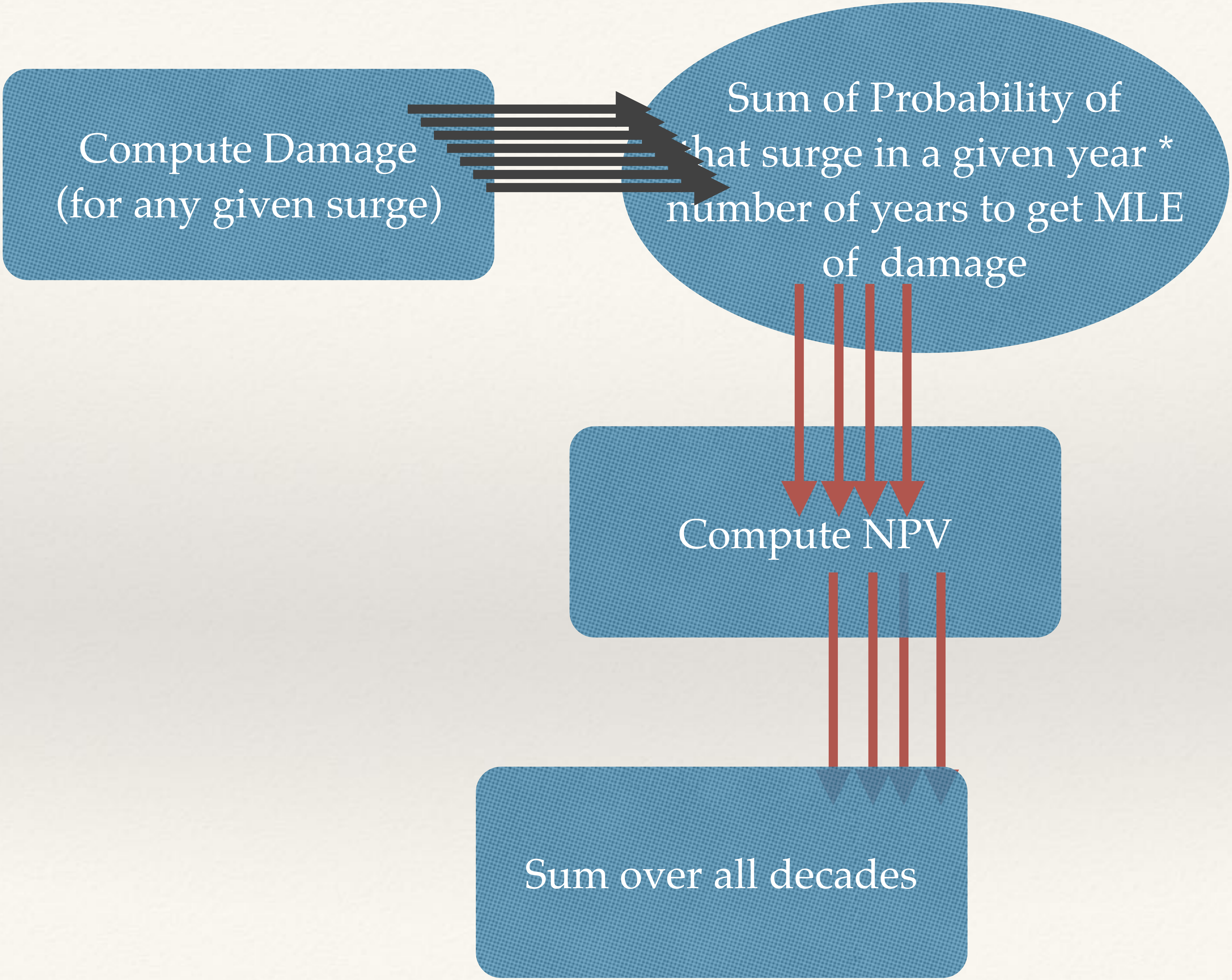


We only do the calculations for discrete probabilities - WHY?

Note that we don't look at storms with a return interval of > 50 years

(because we are looking at a 50 year time horizon

ISSUES with that? MODEL ASSUMPTIONS/LIMITATIONS



Compute Damage
(for any given surge)
Sum of Probability of
that surge in a given year *
number of years to get MLE
of damage
Compute NPV
Sum over all decades

R-components

Main Program	mangrove.modular.R mangrove.modular.asfunction.R
Compute Mangove Attenuation	mangrove_adjustmento_surge.R
Compute Climate Impact on Surge	compute_climatebased_surge.R
Compute Damage	surge_to_damage.R
Compute NPV	compute_NPV.R

R-components

mangrove.modular.R

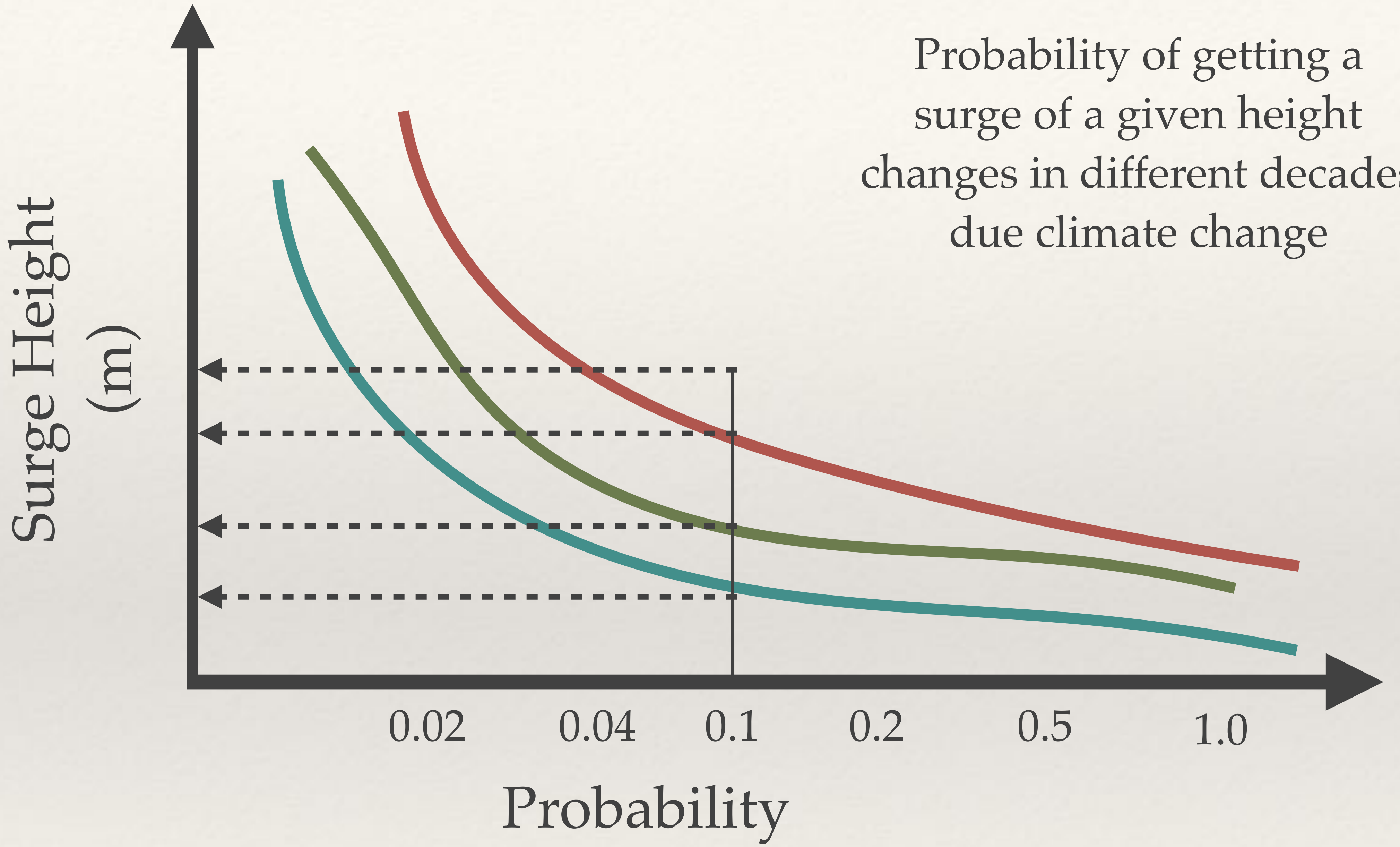
compute_climatebased_surge.R

mangrove_adjustmento_surge.R

surge_to_damage.R

compute_NPV.R

Storm Surge Model: Stochastic (Return Probability)



We only do the calculations for discrete probabilities - WHY?

Note that we don't look at storms with a return interval of > 50 years

(because we are looking at a 50 year time horizon

ISSUES with that? MODEL ASSUMPTIONS/LIMITATIONS

Data structure

- So we are going to have surges, with different probabilities - and this is going to change every decade

Return	Decades		
0.02			
0.04			
0.1			
0.2			
0.5			
0.1			

X

Climate Change Scenarios

Climate Scenario	
base	
A1B1	
A2	
B1	
B2	

Initialization data

- Data on baseline surge heights and their return intervals

RETURN INTERVALS			
SOPAC, 1996		Carter, 1990	
RI	Surge (m)	RI	Wind (knots)
1	0.13	1	22
2	0.28	2	30
5	0.48	10	50
10	0.63	50	85
25	0.83		
50	0.98		

Table 4. Historic return intervals for surge heights and wind speeds on the Suva Peninsula (Solomon & Kruger 1996, Carter 1990)

compute_climatebased_surge

```
compute_climatebased_surge = function(baseline.surge, climate.scenario, ndecades) {

nscenarios = nrow(climate.scenario)

surge = array(dim=c(length(baseline.surge), ndecades, nscenarios) )

for (decadei in 1:ndecades) {

for (scenarioi in 1:nscenarios) {

   for (return_intervali in 1:length(baseline.surge.by.return)) {

         # for a given return interval storm, add intensity increase, and sea level rise

         surge[return_intervali, decadei,  scenarioi] =
           (1 + climate.scenario$ssf[scenarioi]*decadei)*
           baseline.surge.by.return[return_intervali] +
            climate.scenario$slr[scenarioi]*decadei }
   }
 }

return(surge)
}
```

Data structure of Inputs

climate.scenarios	slr	ssf
base		
A1B1		
A2		
B1		
B2		

baseline.surge	
0.02	
0.04	
0.1	
0.2	
0.5	
0.1	

Note that the number of rows for both of these can change
the function would still work with 20 climate scenarios and 25 baseline surges (for different probabilities)

ndecades: a single number

RETURN INTERVALS			
SOPAC, 1996		Carter, 1990	
RI	Surge (m)	RI	Wind (knots)
1	0.13	1	22
2	0.28	2	30
5	0.48	10	50
10	0.63	50	85
25	0.83		
50	0.98		

Table 4. Historic return intervals for surge heights and wind speeds on the Suva Peninsula (Solomon & Kruger 1996, Carter 1990)

Output data structure

- surges, for different return probabilities in each decade for each climate scenario

Return	Decades		
0.02			
0.04			
0.1			
0.2			
0.5			
0.1			

X

Climate Change Scenarios

Climate Scenario	
base	
A1B1	
A2	
B1	
B2	

3-dimensions: return interval, decade, climate scenario

compute_climatebased_surge

```
compute_climatebased_surge = function(baseline.surge, climate.scenario, ndecades) {

nscenarios = nrow(climate.scenario)

surge = array(dim=c(length(baseline.surge), ndecades, nscenarios) )

for (decadei in 1:ndecades) {

for (scenarioi in 1:nscenarios) {

   for (return_intervali in 1:length(baseline.surge.by.return)) {

        # for a given return interval storm, add intensity increase, and sea level rise

        surge[return_intervali, decadei,  scenarioi] =
          (1 + climate.scenario$ssf[scenarioi]*decadei)*
          baseline.surge.by.return[return_intervali] +
           climate.scenario$slr[scenarioi]*decadei }
   }
 }

return(surge)
}
```

mangrove_adjustment_to_surge

```r
#' mangrove_adjustment_to_surge
#'
#' compute the new surge after mangroves
#' @param surge (m) original surge depth
#' @param attenK (percent) attenuation coefficient
#' @return new surge depth (m)




mangrove_adjustment_to_surge = function(surge, attenK) {
 new.surge = (1-attenK)*surge
 new.surge = pmax(new.surge,0)
 return(new.surge)
}
```

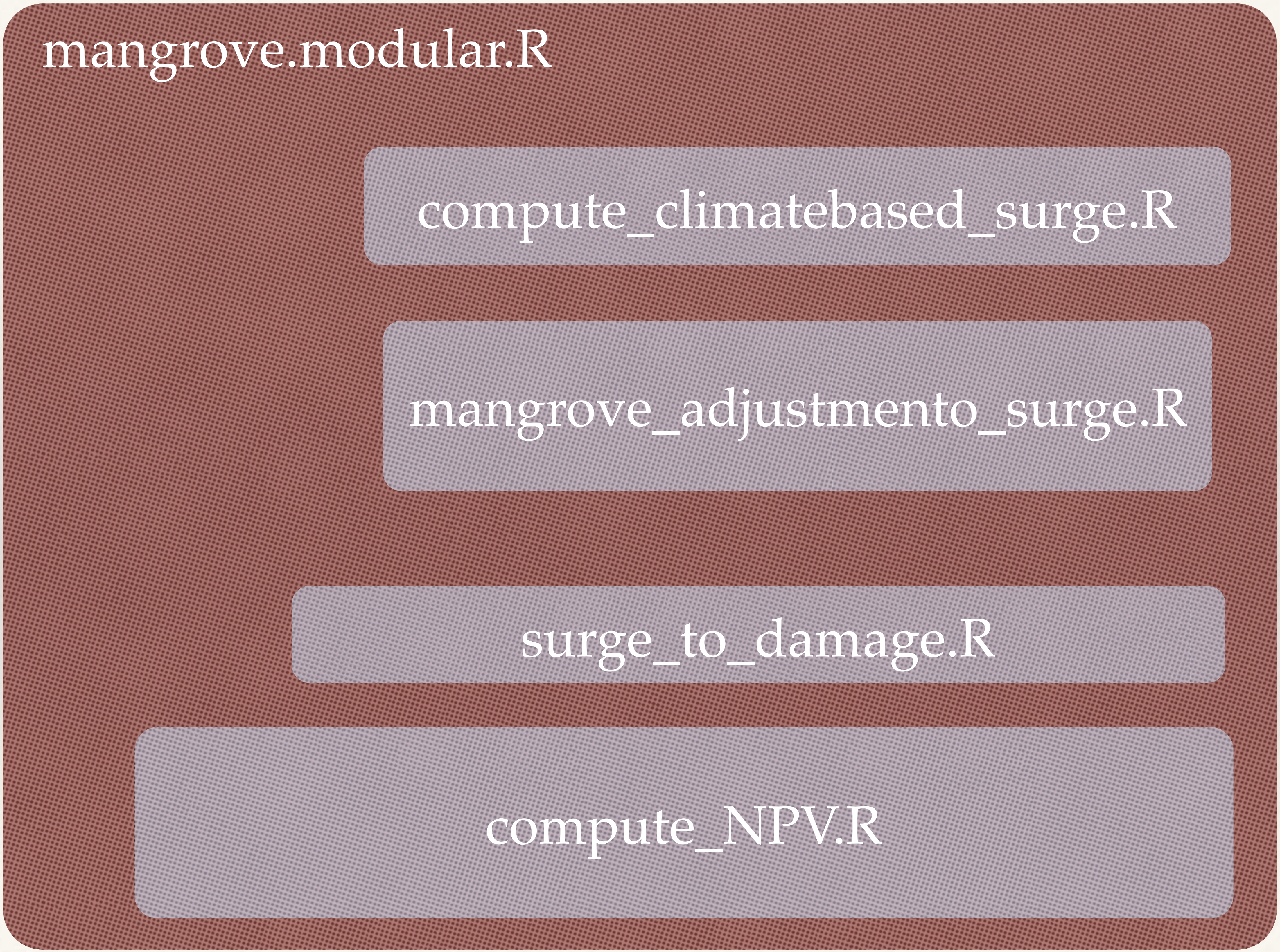
surge_to_damage

```r
#' surge_to_damage
#'
#' Function to compute the costs of damages as a function of surge
#' height
#' @param surge (m)
#' @param surge.min (m) minimum surge that causes damage
#' @param base ($) costs associated with any surge above the minimum
#' @param K ($/m) slope of the surge/damage linear relationship
#'
#' @return damage in $


surge_to_damage = function(surge, surge.min, base, K) {

 flood = ifelse(surge > surge.min, surge-surge.min, 0)
 damage = K*flood+base
 return(damage)
 }
```

surge_to_damage

```
#' compute_NPV
#'
#' compute net present value
#' @param value/cost ($)
#' @param time in the future that cost/value occurs (years)
#' @param discount rate
#' @return value in $


compute_NPV = function(value, time, discount) {
 result = value / (1 + discount)**time
 result
}
```

R-components (submodels and flow-control module)



Option 1: function that runs entire model

- Output is simply damage comparison for adaptation (across a range of mangrove K's and no-adaptation)
- Plots and values
- two options - just a single average mangrove K or uncertainty across multiple K
  - option 1, averages results across uncertainty in K
  - option 2, shows uncertainty due to K

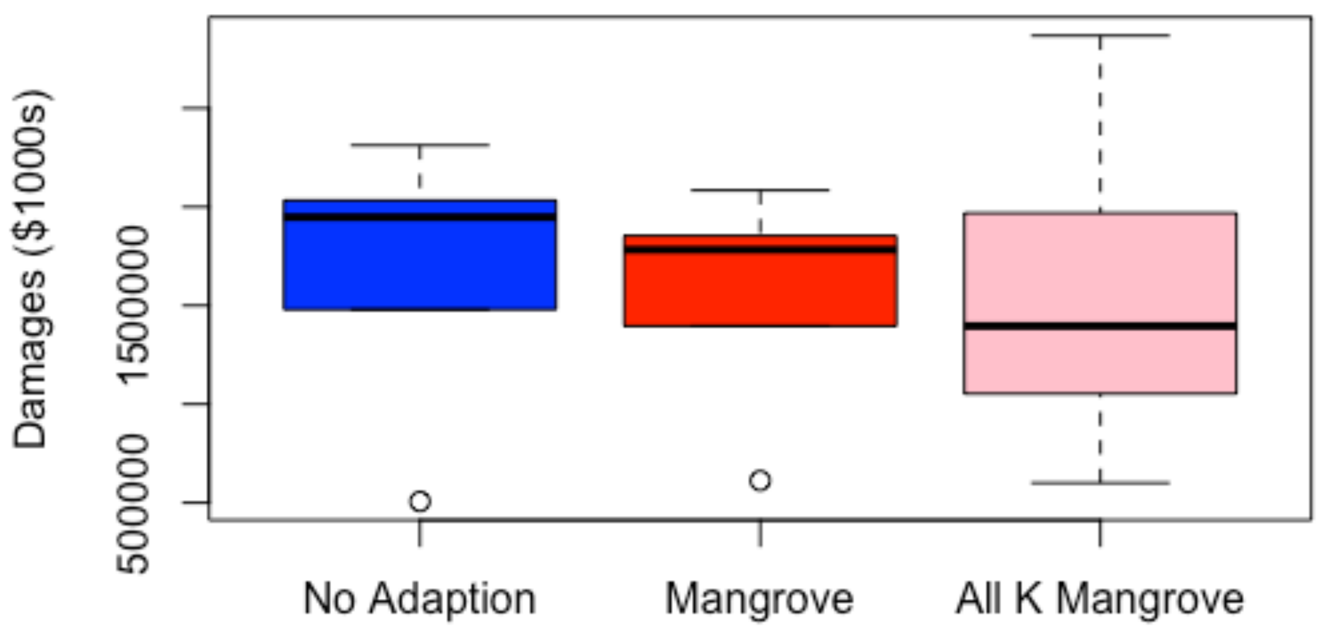
Example model implementation - R script

- use “parameter.R” to see an example implementation of the model
- first download mangrove project, open project and use build:load_all
- now open parameter.R and run

```
# baseline surge heights and associated return intervals
baseline.surge.by.return = c(0.13, 0.28, 0.48, 0.63, 0.83, 0.98)
return_intervals = c(1,2,5,10,25,50)
# turn into probability of surge
return.prob =  1/return_intervals

# build climate change scenarios

# names of scenarios - including baseline
scen.names = c("base","A1B1","A2","B1","B2")

# storm intensity increase per decade for each scenarios
ssf  = c(0,4.30, 3.60, 2.10, 3.60)
ssf.perdecade = ssf/10

# sea level rise per decade for each scenario
slr = c(0,0.59,0.51,0.38,0.43)
slr.perdecade = slr/10
```

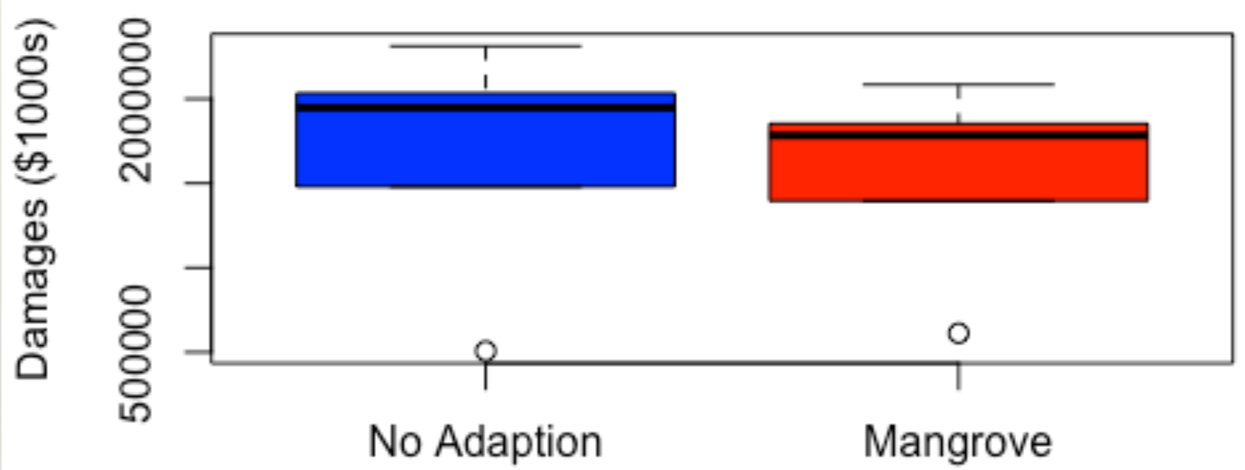
```
# put together for climate scenario
climate.scenario = data.frame(slr=slr, ssf=ssf)
rownames(climate.scenario)=scen.names

# list of possible mangrove attenuation; assume all equally likely
mangrove.atten = c(0.1,0.4,0.7)

# decide on number (and names) of decades to look at
decades = c(90,00,10,20,30,50)
ndecades =  length(decades)
discount=0.01
```

```
adaptation_comparison (baseline.surge.by.return, return.prob, climate.scenario,
                    mangrove.atten, decades, discount, plot.allK=T)
```



```
result = adaptation_comparison (baseline.surge.by.return, return.prob, climate.scenario,
                    mangrove.atten, decades, discount, plot.allK=F)
result
```

Try this variation and see what you get



In class exercise

- play with parameter.R to see how damages vary with assumptions (or could be a different place)
  - add in another climate scenario with a greater sea level rise (slr) per decade
  - try a restored mangrove with greater or less attenuation capability
  - change the discount rate

Improving (or comparing) submodels

- that current damage function is pretty simple - it model damage as a linear function of surge height (above a threshold)
- look at the code
- save is as surge_to_damage_old.R (so you have a copy; we could use github)
- edit surge_to_damage.R to try an alternative model (imagine you had data to derive this) - save, load_all and re -run to see how it changed

A simple example (see surge_to_damage2 but change name why?)

```
#' surge_to_damage
#'
#' Function to compute the costs of damages as a function of surge
#' height
#' @param surge (m)
#' @param surge.min (m) minimum surge that causes damage
#' @param base ($) costs associated with any surge above the minimum
#' @param K ($/m) slope of the surge/damage linear relationship
#'
#' @return damage in $



surge_to_damage = function(surge, surge.min, base, K) {

    flood = ifelse(surge > surge.min, surge-surge.min, 0)
    damage = ifelse(surge > surge.min, K*flood+base, 0)
    return(damage)
    }
```

```r
#' adaptation_comparison
#'
#' compare and plot NPV of no adaptation and mangroves
#' @param baseline.surge baseline surges (before climate change) (m)
#' @param return.prob return probability for each surge in baseline.surge
#' @param climate.scenario an array that gives sea level rise (m) and
#' change in storm intensity (percent) per decade, multiple scenarios can be input
#' @param mangrove.atten vector of mangrove attenuation coeffcients
#' @param decades vectors of decades
#' @param discount discount rate
#' @return list with means for noadapt, mangrove, mangrove.allK



adaptation_comparison = function(baseline.surge.by.return, return.prob,
 climate.scenario, mangrove.atten, decades, discount, plot.allK=T) {

….

}
```

Only creates the final box plot; and saves the mean total NPV of damages for each mangrove and noadaption;
Includes an option for including mangrove attenuation uncertainty

```r
adaptation_comparison = function(baseline.surge.by.return, return.prob, climate.scenario,
 mangrove.atten, decades, discount, plot.allK=T) {


….
# or include climate scenario in a box plot
if (plot.allK==T)
boxplot(total.damage.noadapt.NPV, total.damage.mangrove.meanK.NPV, total.damage.mangrove.K.NPV,
names=c("No Adaption", "Mangrove","All K Mangrove"), ylab="Damages ($1000s)",
 col=c("blue","red","pink"))
else
boxplot(total.damage.noadapt.NPV, total.damage.mangrove.meanK.NPV,
names=c("No Adaption", "Mangrove"), ylab="Damages ($1000s)", col=c("blue","red"))

mean.noadapt=mean(total.damage.noadapt.NPV)
mean.mangrove=mean(total.damage.mangrove.meanK.NPV)
mean.mangrove.allK =mean(total.damage.mangrove.K.NPV)

if(plot.allK==T)
 return(list(noadapt=mean.noadapt, mangrove=mean.mangrove, mangrove.allK=mean.mangrove.allK))
else
return(list(noadapt=mean.noadapt, mangrove=mean.mangrove))


}
```

Note the use of a list to return multiple values

```
# parameter definitions
baseline.surge.by.return = c(0.13, 0.28, 0.48, 0.63, 0.83, 0.98)
return_intervals = c(1,2,5,10,25,50)
return.prob =  1/return_intervals


ssf  = c(0,4.30, 3.60, 2.10, 3.60)
ssf.perdecade = ssf/10
slr = c(0,0.59,0.51,0.38,0.43)
slr.perdecade = slr/10
climate.scenario = data.frame(slr=slr, ssf=ssf)
scen.names = c("base","A1B1","A2","B1","B2")
rownames(climate.scenario)=scen.names
mangrove.atten = c(0.1,0.4,0.7)

decades = c(90,00,10,20,30,50)
ndecades =  length(decades)
discount=0.01

adaptation_comparison(baseline.surge.by.return, return.prob, climate.scenario, mangrove.atten, decades,
 discount=0.2, plot.allK=T)
adaptation_comparison(baseline.surge.by.return, return.prob, climate.scenario, mangrove.atten, decades,
 discount=0.02, plot.allK=T)
```

We still need a script or (input file) to set all our parameters to run our function - but we can change things very easily

```
# parameter definitions


ssf  = c(0,4.30, 3.60, 2.10, 3.60,3.6)
ssf.perdecade = ssf/10
slr = c(0,0.59,0.51,0.38,0.43,0.9)
slr.perdecade = slr/10
climate.scenario = data.frame(slr=slr, ssf=ssf)
scen.names = c(“base”,”A1B1","A2","B1","B2","extreme")
rownames(climate.scenario)=scen.names


adaptation_comparison(baseline.surge.by.return, return.prob,
 climate.scenario, mangrove.atten, decades, discount, plot.allK=T)
```

Now we can add a more extreme climate scenario (big sea-level rise per decade)

```
a=adaptation_comparison(baseline.surge.by.return, return.prob, climate.scenario,
 mangrove.atten, decades, discount, plot.allK=T)
b= adaptation_comparison(baseline.surge.by.return, return.prob,
 climate.scenario[1:5,], mangrove.atten, decades, discount, plot.allK=T)

> a
$noadapt
[1] 1788410

$mangrove
[1] 1654477

$mangrove.allK
[1] 1654957

> b
$noadapt
[1] 1656168

$mangrove
[1] 1545861

$mangrove.allK
[1] 1546438
```

```
> climate.scenario
   slr ssf
1 0.00 0.0
2 0.59 4.3
3 0.51 3.6
4 0.38 2.1
5 0.43 3.6
6 0.90 3.6
```