

# Mathematical representation of the drought decision model - Shiny Version

Trisha Shrum

August 6, 2017

## 1 Model

The goal of the player is to maximize the following objective function:

$$\sum_{t=1}^T \pi_t (1+r)^{-t} \tag{1}$$

## 2 Scripts

### 2.1 global.R

1. Sources other scripts
2. Javascript coding
3. Populate a new environment with rainfall gauge info: `getStationGauge()`
4. Populate a new environment with constant (user) variables: `getConstantVars()`
5. Setting additional variables: acres, start years, simulation lengths
6. Create state variables for practice and full runs: `getSimVars()`
7. Create lists of variables for practice and full runs: `practiceRuns`, `simRuns`
8. Establish additional settings

### 2.2 load.R

Loads necessary packages

## 2.3 shinySupportFunctions.R

1. `getJulyInfo` function: Calculates available and predicted forage in July, creates a UI to display info and allows user to select adaptation level.
  - Called in `simUI.R`
2. `getCowSell` function: Creates a UI for the user to select how many cows and calves to sell. Called in `simUI.R`.
3. `shinyInsurance` function: Calculates premium and indemnification for a specific year and grid cell. Currently, returns are summed but this could be done on a index interval basis instead.

## 2.4 forageFunctions.R

- `getForagePotential` function: Returns an index representing annual forage production for a given gridcell or station gauge's annual precipitation record. Called in `calfCowFunctions.R`.
- `whatIfForage` function: calculates expected forage for a given scenario. Called in `shinySupportFunctions.R` and `simUI.R`.
- `getMLRAWeights` function: Computes forage potential weights using the mean of plant growth curves by MLRA for a specified state. Called in `initialFunctions.R`.
- `COOP_in_MLRA` function: Returns the MLRA in which a specified coop site is located. Called in `initialFunctions.R`.

## 2.5 adaptationFunctions.R

- `calculateAdaptationIntensity` function: Takes forage potential and an adaptation intensity factor to provide a scalar of drought action. If forage potential is above 1 (no drought), then this variable goes to 0 (no adaptation). Called in `shinySupportFunctions.R` and `simUI.R`.

## 2.6 costRevenueFunctions.R

- `calculateExpSales` function: Calculates expected calf revenues for non-drought year.
- `calculateFeedCost` function: Calculates the costs of purchasing additional feed. Called in `getAdaptCost` in `costRevenueFunctions.R`.

- `CalculateRentPastCost` function: Calculates the costs of renting pasture and trucking pairs. Called in `getAdaptCost` in `costRevenueFunctions.R`.
- `getAdaptCost` function: Calculates the cost of adaptation based on strategy, intensity needed, days, and herd size. Called in `shinySupportFunctions.R` and `simUI.R`.

## 2.7 `initialFunctions.R`

- `getConstantVars` function: Reads in constant variables into a `constvars` environment using the file `data/constant_vars.csv`. Called in `global.R`.
- `getSimVars` function: Creates list of simulation variables. Called in `global.R`.
- `getStationGauge` function: Returns precipitation record and locational attributes for the target location. Default is Central Plains Experimental Range (CPER) but alternative locations at COOP sites across Colorado may be specified. Called in `global.R`.
- `createResultsFrame` function: This function creates a theoretical previous result from the year before the simulation begins right now this assumes that there was no drought the year before the simulation and revenues were 0. These assumptions are likely unrealistic and can be adjusted to accomodate different scenarios. Called in `shinySupportFunctions.R` and `server.R`.

## 2.8 `calfCowFunctions.R`

- `AdjWeanSuccess` function: Adjusts weaning success downward for the year of the drought and the following year. Called in `simUI.R`.
- `calfDroughtWeight` function: If forage potential is less than 1, then the calf weight is less than the optimal weight. Called in `shinySupportFunctions.R` and `simUI.R`.
- `calfWeanWeight` function: Computes calf weights based on station/grid cell forage potential for a n-year period. Called in `initialFunctions.R`.
- `shinyHerd` function: calculates the size of herd for the shiny app. Called in `simUI.R`.

## 2.9 `assetFunctions.R`

- `CalcCowAssets` function: Calculates the cow assets for each year. Called in `initialFunctions.R`.

# 3 Function Details

Key functions in the model are listed and described in alphabetical order.

### 3.1 AdjWeanSuccess

- Function: `AdjWeanSuccess`
  - Description: Adjusts weaning success downward for the year of the drought and the following year. If forage production falls below 1 in year  $t = 1$ , then weaning percentage falls slightly in year  $t = 1$  and more drastically in year  $t = 2$ . If forage production falls below 1 in a year  $t = 1$  where weaning percentage was already decremented because of previous forage production deficits or insufficient culling, then weaning percentage falls further in years  $t = 1, 2$  than it would have if the starting point was at the maximum weaning percentage.
  - Inputs: `totalForage`, `totalForage1ya`, `normal.wn.succ`
  - Output: `wn.succ`
  - Assumptions: This equation is based on what I consider to be “reasonable” estimates of weaning success based on forage potential. These fall roughly in line with body condition scores from the *Nutrient Requirements of Beef Cattle*, but are only ballpark estimates.

The weaning percentage maximum is 88%, represented by  $\tilde{\omega}$ . Weaning percentage in year  $t$ ,  $\omega_t$ , is given by:

$$\omega_t = \begin{cases} \tilde{\omega} & \text{if } F_t, F_{t-1} \geq 1 \\ \tilde{\omega} * F_t^{(1/4)} & \text{if } F_t < 1 \wedge F_{t-1} \geq 1 \\ \tilde{\omega} * F_{t-1} & \text{if } F_t \geq 1 \wedge F_{t-1} < 1 \\ \tilde{\omega} * F_t^{(1/4)} * F_{t-1} & \text{if } F_t, F_{t-1} < 1 \end{cases} \quad (2)$$

**Code:**

```
wn.succ <- NULL
if(totalForage < 1 & totalForage1ya >= 1){
  wn.succ <- normal.wn.succ * totalForage^.25
}else if(totalForage >= 1 & totalForage1ya < 1){
  wn.succ <- normal.wn.succ * totalForage1ya
}else if(totalForage < 1 & totalForage1ya < 1){
  wn.succ <- normal.wn.succ * totalForage1ya * totalForage^.25
}else if(totalForage >= 1 & totalForage1ya >= 1){
  wn.succ <- normal.wn.succ
}
```

### 3.2 getStationGauge

The model is currently set for a default to the Central Plains Experimental Range (CPER), but alternative locations for COOP sites in Colorado may also be used.

- Function: `getStationGauge`
  - Description: Returns precipitation record and locational attributes for the target location. Default is Central Plains Experimental Range (CPER).
  - Inputs: `target.loc` (target location. default set to CPER)
  - Outputs: a list called `station.gauge` that contains `monthlyPrecipWeights` (the weight given to precipitation in each month in terms of how much it contributes to annual forage), `precip` (station or gauge precipitation data from 1948 to 2015), `gridCell` (Target Grid Cell for reading in PRF index values at a given point in time).

#### If CPER (default):

1. Monthly precipitation weights are based on the plant growth curves for rangeland with loamy soils in major land resource area 67B (see Figure 1, an area that encompasses a large portion of Eastern Colorado including the CPER site (Site ID: R067BY002CO). We use the plant growth curve for Western Wheatgrass, Blue Grama, Green Needlegrass, Fourwing Saltbush Plant Community (Growth curve number CO6701), which is well suited for grazing and is maintained with properly stocked prescribed grazing.

These monthly weights are the percentage of plant growth that is expected in each month during a normal year. Under normal conditions, total annual production averages 1300 pounds per acre. These weights add up to 1.

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0	0	0.02	0.08	0.2	0.28	0.15	0.12	0.1	0.05	0	0

2. Station Gauge, historical precipitation totals dating back to 1948, which are also read in from the Excel model. Precip totals are collected at CPER itself and do not rely on precip data from COOP sites.
3. The target grid cell 25002 is assigned to the PRF grid cell (assuming this is the correct grid cell)

## 4 Default Model Values

- Starting herd size: 600 cows

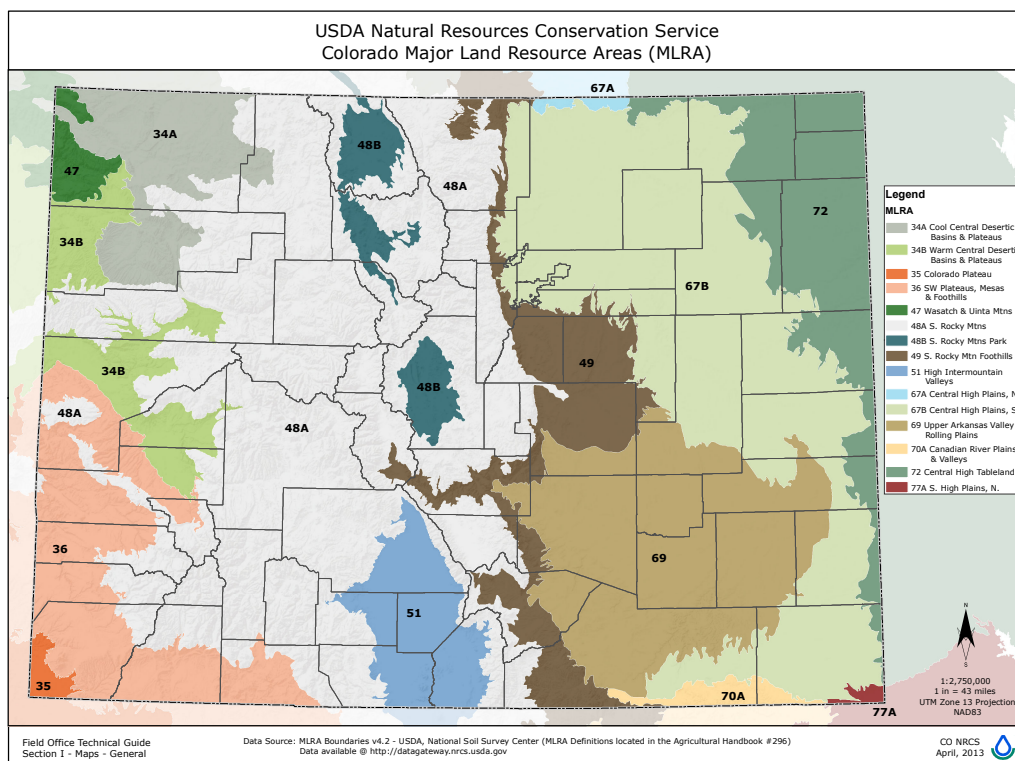


Figure 1: Major Land Resource Areas for Colorado

- Ranch size: 3000 acres
- Carrying capacity: 5 acres/cow
- Maximum calf weaning weight: 600 lbs
- Default calf sale rate: 75%
- Maximum weaning percentage: 88%
- Cow price: \$850/cow
- Calf price at weaning: \$1.40/pound
- Cow mortality rate: 4%
- Practice round years: 1951-1955
- Game round years: 1999-2008
- Household expenses: \$60,000
- Starting cash assets: \$90,000
- Operating costs: \$500/cow
- Insurance Coverage Level: 90%
- Interest rate: 5%