**Overview, Design Concepts, and Details for the FFAR-ABModel**

**Version addressed: July 13, 2022**

A method of providing an overview, design concepts and details of agent-based models is now commonly used and known as the ODD (Overview, Design concepts, Detail) protocol. The protocol (Grimm et al. 2006) has been through two revisions (Grimm et al. 2010; Grimm et al. 2020). We describe FFAR-ABModel below using the ODD protocol, with elements in that protocol removed if not relevant, as encouraged by the authors (Grimm et al. 2006).

**1. Purpose**

The FFAR-ABModel was created to allow us to address scenarios relevant to Colorado and the Denver Public Schools under the Good Food Purchasing Program. Questions under that program include, for example, changing the origin of commodities purchased to favor Colorado products or shifts in differentiated commodities to promote the purchase of presumably healthier options or those that may contribute less to carbon emissions or use less water. Four commodities were selected for inclusion in the model, potatoes, wheat, beef, and peaches.

In this ODD, detail in some places mirrors the sequencing of development of the model for the four commodities included. Potatoes were the first commodity to be developed, followed by wheat, beef, and then peaches. There are analogous components used in representing the different commodities, and repeated descriptions of those components would make this ODD lengthier and more cumbersome. Instead, we focus on potatoes in some detail, review wheat in detail as well, and describe novel components of the beef and peach food systems.

**2. Entities, state variables, and scales**

Entities in the model include patches of landscape that comprise Colorado and agent-types involved in the provisioning of the four commodities to Denver public schools (DPS). Agent-types early in the food supply chain are commodity-specific. Later in the chain, agents manage any commodity. The agent-types include:

* Producers (i.e., potato farmers, wheat farmers, peach farmers, and beef ranchers)
* Shippers and distributors (i.e., potato shippers, wheat elevators, distributors)
* Processors (i.e., potato repackers, potato processors, beef packing-houses, peach processors, wheat mills, wheat bakers)
* Purchasers and those offering products to consumers (i.e., DPS buyer, peach alternative markets such as farmers markets)

Schools themselves are not represented, in-so-far as products purchased by the DPS buyer have only one outlet (overstocking and other waste aside), to be prepared for students. The inclusion of schools and students would therefore be somewhat redundant. Landscape patches are 1 km x 1 km in resolution, such that Colorado is represented in a spatial arena that is 625 km by 453 km. State variables for each patch are cited in Table 1.

**Table 1.** Summary of state variables for the 1 km x 1 km landscape patches representing Colorado in the FFAR-ABModel. Commodity-specific yield variables have the commodity as a prefix that is not shown here.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Common** | |  |  |
|  | county-id; county-denver | Number | County identifiers |
|  | occupied | Logical | An indicator as to whether a patch is occupied by a farmer or rancher |
|  | farmer | Number | An identifier for the farmer owning the patch |
|  | my-farmer | Agentset | The set of the farmer agent occupying the patch |
|  | certified-organic | Logical | Whether a patch has organic status or not |
|  | rotation | Text | The current rotation for the patch, if growing potatoes (e.g., “potato”, “barley”) and wheat (e.g., “wheat”, “corn”, “fallow”) |
|  | remainder-patch | Hectares | If positive, the area of potato production that exceeds the area of full 100 ha patches cultivated |
|  | organic-conversion-tracker | Year | Tracking years until conversion to organic status |
|  | in-organic-transition | Logical | Whether or not a patch is in transition to organic status |
|  |  |  |  |
| **Potatoes** | |  |  |
|  | county-potato; potato-yes-no | Logical | Whether or not a patch occurs in a county producing potatoes, and whether it is producing potatoes or not |
|  | county-potato-name | Text | The county name |
|  | canela-mean-yield; canela-org-mean-yield; purple-mean-yield; purple-org-mean-yield; russet-canela-current-yield; russet-canela-org-current-yield; purple-current-yield; purple-org-current-yield | kg ha-1 | Annual mean yields estimated for conventional and differentiated products, including conventional and organic types. Commodities include russet canela potatoes and purple potatoes. These include yields calculated from an ecosystem model (e.g., mean-yield) and values that include annual variation (e.g., current-yield) |
|  |  |  |  |
| **Wheat** | |  |  |
|  | county-wheat; wheat-yes-no | Logical | Whether or not a patch occurs in a county producing wheat, and whether it is producing wheat or not |
|  | county-wheat-name | Text | The county name |
|  | hrw-conv-mean-yield; hrw-org-mean-yield; snowmass-conv-mean-yield; hrw-conv-current-yield; hrw-org-current-yield; snowmass-conv-current-yield | bushel ac-1 | Annual mean yields estimated for conventional and differentiated products, including conventional and organic types. Commodities include hard red winter wheat and snowmass wheat. These include yields calculated from an ecosystem model (e.g., mean-yield) and values that include annual variation (e.g., current-yield) |
|  | patch-wheat-fallow-rotate; patch-corn-wheat-fallow-rotate | Logical | Whether or not wheat is in rotation with a fallow state, and whether or not wheat is in rotation with corn and a fallow state |
|  |  |  |  |
| **Peaches** | |  |  |
|  | county-peach; peach-yes-no | Logical | Whether or not a patch occurs in a county producing peaches, and whether it is producing peaches or not |
|  | county-peach-name | Text | The county name |
|  | conventional-mean-yield; organic-mean-yield; conventional-current-yield; organic-current-yield | lbs ha-1 | Annual mean yield estimated for conventional and organic peach production. These include yields calculated from an ecosystem model (i.e., mean-yield) and values that include annual variation (i.e., current-yield) |
|  |  |  |  |
| **Beef** | |  |  |
|  | county-beef | Logical | Whether or not a patch occurs in a county producing beef |
|  | public-private | Number | A value that indicates whether the patch is public land or private land |
|  | mean-forage-production; annual-forage production | kg ha-1 | Annual mean forage production from rangelands within the landscape patch. This includes forage production estimated for the area (mean-forage-production) and a value modified to capture annual variability |
|  | year-precip | mm | Mean annual precipitation for the landscape patch |
|  | precip-pct | Percent | The percent of precipitation in the year compared to the mean annual precipitation |
|  | average-precip | mm | Average annual precipitation read in from a surface |
|  | irrigated | Logical | Whether or not the landscape patch is irrigated |
|  | grazable | Logical | Whether or not the landscape patch is grazable |
|  |  |  |  |

**Table 2.** Summary of state variables for potato farmers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id; farm-id | Number | A numeric identifier of the farmer, and a secondary identifier |
|  | agent-name | Text | A string identifying the agent group |
|  | home-county | Text | The identifier of the county of the farmer |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the farmer shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final farm-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | my-russet-conventional-hectares-for-costs; my-russet-organic-hectares-for-costs; my-purple-conventional-hectares-for-costs; my-purple-organic-hectares-for-costs | ha | Hectares in production of commodities, used in cost calculations |
|  | potatoes-russet-conventional-fresh\_price-to-sell; potatoes-russet-organic-fresh\_price-to-sell; potatoes-purple-conventional-fresh\_price-to-sell; potatoes-purple-organic-fresh\_price-to-sell; potatoes-russet-conventional-fresh-small\_price-to-sell | $ | The price to sell the indicated commodity |
|  |  |  |  |
| **Production** | |  |  |
|  | hectares-farmed-potatoes | ha | Area of potatoes farmed |
|  | hectares-farmed-total | ha | The total area farmed, which may include rotation crops |
|  | initialized | Logical | A logical value used in initialization |
|  | patches-occupied | Number | The number of patches occupied by the farm |
|  | my-patches | Patches | A NetLogo list of patches owned by the farmer |
|  | grows-organic | Logical | Whether or not the farmer has organic patches |
|  | fully-organic | Logical | Whether or not the farmer has all organic patches |
|  | russet-canela-allocation; purple-allocation | Percent | The percentage of potato production for the farmer in the cited commodities |
|  | russet-canela-current-production; russet-canela-org-current-production; purple-current-production; purple-org-current-production | lbs | Annual total production of the commodities |
|  | patches-in-potatoes | Number | The number of patches required to grow the amount of potatoes associated with each commodity |
|  | patches-in-rotation | Number | The number of patches (if any) that are in the rotation crop barley |
|  | potato-farmers\_yield-estimated\_russet-conventional; potato-farmers\_yield-estimated\_russet-organic; potato-farmers\_yield-estimated\_purple-conventional; potato-farmers\_yield-estimated\_purple-organic; potato-farmers\_yield-estimated\_russet-conventional-small | lbs | Estimated annual production of the indicated commodities |
|  |  |  |  |
| **Inventory** | |  |  |
|  | potatoes-russet-conventional-fresh\_inventory-current; potatoes-russet-conventional-fresh\_inventory-maximum; potatoes-russet-organic-fresh\_inventory-current; potatoes-russet-organic-fresh\_inventory-maximum; potatoes-purple-conventional-fresh\_inventory-current; potatoes-purple-conventional-fresh\_inventory-maximum; potatoes-purple-organic-fresh\_inventory-current; potatoes-purple-organic-fresh\_inventory-maximum; potatoes-russet-conventional-fresh-small\_inventory-current; potatoes-russet-conventional-fresh-small\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | potatoes-russet-conventional-fresh-promised; potatoes-russet-organic-fresh-promised; potatoes-purple-conventional-fresh-promised; potatoes-purple-organic-fresh-promised; potatoes-russet-conventional-fresh-small-promised | lbs | The amount of potatoes of the commodity indicated that have been contracted for provision to purchasers |
|  | potatoes-russet-conventional\_passthrough-annual; potatoes-russet-organic\_passthrough-annual; potatoes-purple-conventional\_passthrough-annual; potatoes-purple-organic\_passthrough-annual  potatoes-russet-conventional-small\_passthrough-annual | lbs | Tracking inventory managed by the farmer for different commodities, accumulating any product handled by the farmer. This is helpful in plotting |
|  |  |  |  |
| **Decision-making** | |  |  |
|  | transition-likelihood-score; tls-initial; tls-final; tls-change | Index | A transition likelihood score indicating the likelihood that a farmer will transition some parcels from conventional to organic methods. Initial values, final values, and the change in values are tracked as well |
|  | male-gender | Logical | Whether or not the farmer is a male |
|  | first-time-farmer | Logical | Whether or not the person is a first-time farmer |
|  | young-farmer | Logical | Whether or not the farmer is young |
|  | not-white | Logical | Whether or not the farmer is white |
|  | primary-income | Logical | Whether or not the farm is the primary source of income for the farmer |
|  | alternative-farmer | Logical | Whether or not the farmer is apt to adopt experimental approaches or otherwise at greater likelihood to take risks in management |
|  | farm-size | Rank | The relative size of the farm |
|  | farm-transitioning | Logical | Whether or not the farm is transitioning to organic production methods |
|  | this-year-mediocre-production-year; last-year-mediocre-production-year; this-year-bad-production-year; last-year-bad-production-year | Logical | Used to affect transition likelihood scores, based on previous production amounts |
|  |  |  |  |

**Table 3**. Summary of state variables for potato repackers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the repacker |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the repacker shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | potatoes-russet-conventional-fresh\_price-to-sell; potatoes-russet-organic-fresh\_price-to-sell; potatoes-purple-conventional-fresh\_price-to-sell; potatoes-purple-organic-fresh\_price-to-sell; potatoes-russet-conventional-fresh-small\_price-to-sell | $ | The price to sell the indicated commodity |
|  | potatoes-russet-conventional-fresh\_price-to-buy; potatoes-russet-organic-fresh\_price-to-buy; potatoes-purple-conventional-fresh\_price-to-buy; potatoes-purple-organic-fresh\_price-to-buy; potatoes-russet-conventional-fresh-small\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | potatoes-russet-conventional-fresh\_inventory-current; potatoes-russet-conventional-fresh\_inventory-maximum; potatoes-russet-organic-fresh\_inventory-current; potatoes-russet-organic-fresh\_inventory-maximum; potatoes-purple-conventional-fresh\_inventory-current; potatoes-purple-conventional-fresh\_inventory-maximum; potatoes-purple-organic-fresh\_inventory-current; potatoes-purple-organic-fresh\_inventory-maximum; potatoes-russet-conventional-fresh-small\_inventory-current; potatoes-russet-conventional-fresh-small\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | potatoes-russet-conventional-fresh-promised; potatoes-russet-organic-fresh-promised; potatoes-purple-conventional-fresh-promised; potatoes-purple-organic-fresh-promised; potatoes-russet-conventional-fresh-small-promised | lbs | The amount of potatoes of the commodity indicated that have been contracted for provision to purchasers |
|  | potatoes-russet-conventional-fresh\_contract-space-available; potatoes-russet-conventional-fresh\_space-available; potatoes-russet-organic-fresh\_contract-space-available; potatoes-purple-conventional-fresh\_contract-space-available; potatoes-purple-organic-fresh\_contract-space-available; potatoes-russet-conventional-fresh-small\_contract-space-available; potatoes-russet-conventional-fresh-small\_space-available | lbs | Inventory and contract space available, used in tracking sales |
|  | potatoes-russet-conventional-fresh\_my-incoming-total; potatoes-russet-organic-fresh\_my-incoming-total  potatoes-purple-conventional-fresh\_my-incoming-total; potatoes-purple-organic-fresh\_my-incoming-total  potatoes-russet-conventional-fresh-small\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 4.** Summary of state variables for potato shippers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the repacker |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the repacker shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | potatoes-russet-conventional-fresh\_price-to-sell; potatoes-russet-organic-fresh\_price-to-sell; potatoes-purple-conventional-fresh\_price-to-sell; potatoes-purple-organic-fresh\_price-to-sell; potatoes-russet-conventional-fresh-small\_price-to-sell | $ | The price to sell the indicated commodity |
|  | potatoes-russet-conventional-fresh\_price-to-buy; potatoes-russet-organic-fresh\_price-to-buy; potatoes-purple-conventional-fresh\_price-to-buy; potatoes-purple-organic-fresh\_price-to-buy; potatoes-russet-conventional-fresh-small\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | potatoes-russet-conventional-fresh\_inventory-current; potatoes-russet-conventional-fresh\_inventory-maximum; potatoes-russet-organic-fresh\_inventory-current; potatoes-russet-organic-fresh\_inventory-maximum; potatoes-purple-conventional-fresh\_inventory-current; potatoes-purple-conventional-fresh\_inventory-maximum; potatoes-purple-organic-fresh\_inventory-current; potatoes-purple-organic-fresh\_inventory-maximum; potatoes-russet-conventional-fresh-small\_inventory-current; potatoes-russet-conventional-fresh-small\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | potatoes-russet-conventional-fresh-promised; potatoes-russet-organic-fresh-promised; potatoes-purple-conventional-fresh-promised; potatoes-purple-organic-fresh-promised; potatoes-russet-conventional-fresh-small-promised | lbs | The amount of potatoes of the commodity indicated that have been contracted for provision to purchasers |
|  | potatoes-russet-conventional-fresh\_contract-space-available; potatoes-russet-conventional-fresh\_space-available; potatoes-russet-organic-fresh\_contract-space-available; potatoes-purple-conventional-fresh\_contract-space-available; potatoes-purple-organic-fresh\_contract-space-available; potatoes-russet-conventional-fresh-small\_contract-space-available; potatoes-russet-conventional-fresh-small\_space-available | lbs | Inventory and contract space available, used in tracking sales |
|  | potatoes-russet-conventional-fresh\_my-incoming-total; potatoes-russet-organic-fresh\_my-incoming-total  potatoes-purple-conventional-fresh\_my-incoming-total; potatoes-purple-organic-fresh\_my-incoming-total  potatoes-russet-conventional-fresh-small\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 5**. Summary of state variables of potato processors.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the repacker |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the repacker shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | potatoes-russet-conventional-fresh\_price-to-sell; potatoes-russet-organic-fresh\_price-to-sell; potatoes-purple-conventional-fresh\_price-to-sell; potatoes-purple-organic-fresh\_price-to-sell; potatoes-russet-conventional-fresh-small\_price-to-sell; potatoes-russet-conventional-processed\_price-to-sell; potatoes-purple-conventional-processed\_price-to-buy;  potatoes-purple-conventional-processed\_price-to-sell; potatoes-purple-organic-processed\_price-to-buy;  potatoes-purple-organic-processed\_price-to-sell | $ | The price to sell the indicated commodity, including those that have been processed |
|  | potatoes-russet-conventional-fresh\_price-to-buy; potatoes-russet-organic-fresh\_price-to-buy; potatoes-purple-conventional-fresh\_price-to-buy; potatoes-purple-organic-fresh\_price-to-buy; potatoes-russet-conventional-fresh-small\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | potatoes-russet-conventional-fresh\_inventory-current; potatoes-russet-conventional-fresh\_inventory-maximum; potatoes-russet-organic-fresh\_inventory-current; potatoes-russet-organic-fresh\_inventory-maximum; potatoes-purple-conventional-fresh\_inventory-current; potatoes-purple-conventional-fresh\_inventory-maximum; potatoes-purple-organic-fresh\_inventory-current; potatoes-purple-organic-fresh\_inventory-maximum; potatoes-russet-conventional-fresh-small\_inventory-current; potatoes-russet-conventional-fresh-small\_inventory-maximum; potatoes-russet-conventional-processed\_inventory-current; potatoes-russet-conventional-processed\_inventory-maximum; potatoes-purple-conventional-processed\_inventory-current;  potatoes-purple-conventional-processed\_inventory-maximum; potatoes-purple-organic-processed\_inventory-current; potatoes-purple-organic-processed\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity, including processed commodities |
|  | potatoes-russet-conventional-fresh-promised; potatoes-russet-organic-fresh-promised; potatoes-purple-conventional-fresh-promised; potatoes-purple-organic-fresh-promised; potatoes-russet-conventional-fresh-small-promised; potatoes-russet-conventional-processed-promised; potatoes-purple-conventional-processed-promised; potatoes-purple-organic-processed-promised | lbs | The amount of potatoes of the commodity indicated that have been contracted for provision to purchasers |
|  | potatoes-russet-conventional-fresh\_contract-space-available; potatoes-russet-conventional-fresh\_space-available; potatoes-russet-organic-fresh\_contract-space-available; potatoes-purple-conventional-fresh\_contract-space-available; potatoes-purple-organic-fresh\_contract-space-available; potatoes-russet-conventional-fresh-small\_contract-space-available; potatoes-russet-conventional-fresh-small\_space-available; potatoes-russet-conventional-processed\_contract-space-available; potatoes-purple-conventional-processed\_contract-space-available; potatoes-purple-organic-processed\_contract-space-available | lbs | Inventory and contract space available, used in tracking sales |
|  | potatoes-russet-conventional-fresh\_my-incoming-total; potatoes-russet-organic-fresh\_my-incoming-total  potatoes-purple-conventional-fresh\_my-incoming-total; potatoes-purple-organic-fresh\_my-incoming-total  potatoes-russet-conventional-fresh-small\_my-incoming-total; potatoes-russet-conventional-processed\_my-incoming-total; potatoes-purple-conventional-processed\_my-incoming-total; potatoes-purple-organic-processed\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 6.** Summary of state variables for wheat farmers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id; farm-id | Number | A numeric identifier of the farmer, and a secondary identifier |
|  | agent-name | Text | A string identifying the agent group |
|  | home-county | Text | The identifier of the county of the farmer |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the farmer shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final farm-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | my-hrw-conventional-hectares-for-costs; my-hrw-organic-hectares-for-costs; my-snowmass-conventional-hectares-for-costs | ha | Hectares in production of commodities, used in cost calculations |
|  | wheat-hrw-conventional-fresh\_price-to-sell; wheat-hrw-organic-fresh\_price-to-sell; wheat-snowmass-conventional-fresh\_price-to-sell | $ | The price to sell the indicated commodity |
|  |  |  |  |
| **Production** | |  |  |
|  | hectares-farmed-wheat | ha | Area of wheat farmed |
|  | hectares-farmed-total | ha | The total area farmed, which may include rotation crops |
|  | initialized | Logical | A logical value used in initialization |
|  | patches-occupied | Number | The number of patches occupied by the farm |
|  | my-patches | Patches | A NetLogo list of patches owned by the farmer |
|  | grows-organic | Logical | Whether or not the farmer has organic patches |
|  | fully-organic | Logical | Whether or not the farmer has all organic patches |
|  | hrw-allocation; snowmass-allocation | Percent | The percentage of wheat production for the farmer in the cited commodities |
|  | wheat-hrw-conventional-current-production; wheat-hrw-organic-current-production; wheat-snowmass-conventional-current-production; wheat-snowmass-organic-current-production | bushel | Annual total production of the commodities |
|  | wheat-fallow-rotate; corn-wheat-fallow-rotate | Number | An indicator of the type of rotation used in the landscape patch |
|  | patches-in-wheat | Number | The number of patches required to grow the amount of wheat associated with each commodity |
|  | patches-in-rotation | Number | The number of patches (if any) that are in the rotation crop |
|  | wheat-farmers\_yield-estimated\_hrw-conventional; wheat-farmers\_yield-estimated\_hrw-organic; wheat-farmers\_yield-estimated\_snowmass-conventional | bushel | Estimated annual production of the indicated commodities |
|  |  |  |  |
| **Inventory** | |  |  |
|  | wheat-hrw-conventional-fresh\_inventory-current; wheat-hrw-conventional-fresh\_inventory-maximum; wheat-hrw-organic-fresh\_inventory-current; wheat-hrw-organic-fresh\_inventory-maximum; wheat-snowmass-conventional-fresh\_inventory-current; wheat-snowmass-conventional-fresh\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | wheat-hrw-conventional-fresh-promised; wheat-hrw-organic-fresh-promised; wheat-snowmass-conventional-fresh-promised | lbs | The amount of wheat of the commodity indicated that have been contracted for provision to purchasers |
|  | wheat-hrw-conventional\_passthrough-annual; wheat-hrw-organic\_passthrough-annual; wheat-snowmass-conventional\_passthrough-annual | lbs | Tracking inventory managed by the farmer for different commodities, accumulating any product handled by the farmer. This is helpful in plotting |
|  |  |  |  |
| **Decision-making** | |  |  |
|  | transition-likelihood-score; tls-initial; tls-final; tls-change | Index | A transition likelihood score indicating the likelihood that a farmer will transition some parcels from conventional to organic methods. Initial values, final values, and the change in values are tracked as well |
|  | male-gender | Logical | Whether or not the farmer is a male |
|  | first-time-farmer | Logical | Whether or not the person is a first-time farmer |
|  | young-farmer | Logical | Whether or not the farmer is young |
|  | not-white | Logical | Whether or not the farmer is white |
|  | primary-income | Logical | Whether or not the farm is the primary source of income for the farmer |
|  | alternative-farmer | Logical | Whether or not the farmer is apt to adopt experimental approaches or otherwise at greater likelihood to take risks in management |
|  | farm-size | Rank | The relative size of the farm |
|  | farm-transitioning | Logical | Whether or not the farm is transitioning to organic production methods |
|  | this-year-mediocre-production-year; last-year-mediocre-production-year; this-year-bad-production-year; last-year-bad-production-year | Logical | Used to affect transition likelihood scores, based on previous production amounts |
|  |  |  |  |

**Table 7**. Summary of state variables of wheat elevators.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the elevator |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the elevator shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | wheat-hrw-conventional\_price-to-sell; wheat-hrw-organic\_price-to-sell; wheat-snowmass-conventional\_price-to-sell | $ | The price to sell the indicated commodity |
|  | wheat-hrw-conventional\_price-to-buy; wheat-hrw-organic\_price-to-buy; wheat-snowmass-conventional \_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | wheat-hrw-conventional \_inventory-current; wheat-hrw-conventional \_inventory-maximum; wheat-hrw-organic \_inventory-current; wheat-hrw-organic\_inventory-maximum; wheat-snowmass-conventional \_inventory-current; wheat-snowmass-conventional \_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | wheat-hrw-conventional-promised; wheat-hrw-organic- promised; wheat-snowmass-conventional- promised | lbs | The amount of wheat of the commodity indicated that have been contracted for provision to purchasers |
|  | wheat-hrw-conventional\_contract-space-available; wheat-hrw-organic\_contract-space-available; wheat-snowmass-conventional\_contract-space-available | lbs | Inventory and contract space available, used in tracking sales |
|  | wheat-hrw-conventional- my-incoming-total; wheat-hrw-organic\_my-incoming-total; wheat-snowmass-conventional\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 8.** Summary of state variables of wheat mills.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the elevator |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the elevator shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | wheat-hrw-conventional\_price-to-sell; wheat-hrw-organic\_price-to-sell; wheat-snowmass-conventional\_price-to-sell;  wheat-hrw-conventional-flour\_price-to-sell; wheat-hrw-organic-flour\_price-to-sell; wheat-snowmass-conventional-flour\_price-to-sell | $ | The price to sell the indicated commodity |
|  | wheat-hrw-conventional\_price-to-buy; wheat-hrw-organic\_price-to-buy; wheat-snowmass-conventional \_price-to-buy;  wheat-hrw-conventional-flour\_price-to-buy; wheat-hrw-organic-flour\_price-to-buy; wheat-snowmass-conventional-flour\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | wheat-hrw-conventional \_inventory-current; wheat-hrw-conventional \_inventory-maximum; wheat-hrw-organic \_inventory-current; wheat-hrw-organic\_inventory-maximum; wheat-snowmass-conventional \_inventory-current; wheat-snowmass-conventional \_inventory-maximum;  wheat-hrw-conventional-flour\_inventory-current; wheat-hrw-conventional- flour\_inventory-maximum; wheat-hrw-organic-flour \_inventory-current; wheat-hrw-organic-flour\_inventory-maximum; wheat-snowmass-conventional-flour \_inventory-current; wheat-snowmass-conventional-flour\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | wheat-hrw-conventional-promised; wheat-hrw-organic- promised; wheat-snowmass-conventional- promised; wheat-hrw-conventional-flour-promised; wheat-hrw-organic-flour- promised; wheat-snowmass-conventional-flour-promised | lbs | The amount of wheat of the commodity indicated that have been contracted for provision to purchasers |
|  | wheat-hrw-conventional-space-available; wheat-hrw-organic\_contract-space-available; wheat-snowmass-conventional\_contract-space-available; wheat-hrw-conventional-flour\_contract-space-available; wheat-hrw-organic-flour\_contract-space-available; wheat-snowmass-conventional-flour\_contract-space-available | lbs | Inventory and contract space available, used in tracking sales |
|  | wheat-hrw-conventional- my-incoming-total; wheat-hrw-organic\_my-incoming-total; wheat-snowmass-conventional\_my-incoming-total; wheat-hrw-conventional-flour\_ my-incoming-total; wheat-hrw-organic-flour\_my-incoming-total; wheat-snowmass-conventional-flour\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 9.** Summary of state variables for wheat bakers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the baker |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the baker shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | wheat-hrw-conventional-bread\_price-to-sell; wheat-hrw-organic-bread\_price-to-sell; wheat-snowmass-conventional-bread\_price-to-sell;  wheat-hrw-conventional-flour\_price-to-sell; wheat-hrw-organic-flour\_price-to-sell; wheat-snowmass-conventional-flour\_price-to-sell | $ | The price to sell the indicated commodity |
|  | wheat-hrw-conventional-bread\_price-to-buy; wheat-hrw-organic-bread\_price-to-buy; wheat-snowmass-conventional-bread\_price-to-buy; wheat-hrw-conventional-flour\_price-to-buy; wheat-hrw-organic-flour\_price-to-buy; wheat-snowmass-conventional-flour\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | wheat-hrw-conventional-bread\_inventory-current; wheat-hrw-conventional-bread \_inventory-maximum; wheat-hrw-organic-bread\_inventory-current; wheat-hrw-organic\_inventory-maximum; wheat-snowmass-conventional \_inventory-current; wheat-snowmass-conventional \_inventory-maximum;  wheat-hrw-conventional-flour\_inventory-current; wheat-hrw-conventional- flour\_inventory-maximum; wheat-hrw-organic-flour \_inventory-current; wheat-hrw-organic-flour\_inventory-maximum; wheat-snowmass-conventional-flour \_inventory-current; wheat-snowmass-conventional-flour\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | wheat-hrw-conventional-bread-promised; wheat-hrw-organic-bread-promised; wheat-snowmass-conventional-bread-promised; wheat-hrw-conventional-flour-promised; wheat-hrw-organic-flour- promised; wheat-snowmass-conventional-flour-promised | lbs | The amount of wheat of the commodity indicated that have been contracted for provision to purchasers |
|  | wheat-hrw-conventional—bread-space-available; wheat-hrw-organic-bread-\_contract-space-available; wheat-snowmass-conventional-bread\_contract-space-available; wheat-hrw-conventional-flour\_contract-space-available; wheat-hrw-organic-flour\_contract-space-available; wheat-snowmass-conventional-flour\_contract-space-available | lbs | Inventory and contract space available, used in tracking sales |
|  | wheat-hrw-conventional-bread-my-incoming-total; wheat-hrw-organic-bread\_my-incoming-total; wheat-snowmass-conventional-bread\_my-incoming-total; wheat-hrw-conventional-flour\_ my-incoming-total; wheat-hrw-organic-flour\_my-incoming-total; wheat-snowmass-conventional-flour\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 10.** Summary of state variables of peach farmers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the farmer |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the farmer shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final farm-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | peaches-seconds\_price-to-sell-wholesale; peaches-iqf\_price-to-sell | $ | The price to sell the indicated commodity |
|  | peaches-seconds\_price-to-buy; peaches-iqf\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
|  |  |  |  |
| **Production** | |  |  |
|  | hectares-farmed-peaches | ha | Area of peaches farmed |
|  | hectares-farmed-total | ha | The total area farmed |
|  | initialized | Logical | A logical value used in initialization |
|  | patches-in-peaches | Number | The number of patches growing peaches |
|  | my-patches | Patches | A NetLogo list of patches owned by the farmer |
|  | grows-organic | Logical | Whether or not the farmer has organic patches |
|  | fully-organic | Logical | Whether or not the farmer has all organic patches |
|  | peach-farmers\_yield-estimates\_conventional; peach-farmers\_yield-estimated\_organic; peach-farmers\_yield-estimated\_seconds | lbs | The estimated yield of peaches for the commodities indicated |
|  | patches-occupied | Number | The number of patches required to grow the amount of peaches required |
|  |  |  |  |
| **Inventory** | |  |  |
|  | peaches-seconds\_inventory-current; peaches-seconds\_inventory-maximum; peaches-iqf\_inventory-current; peaches-iqf\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | peaches-seconds-promised; peaches-iqf-promised | lbs | The amount of wheat of the commodity indicated that have been contracted for provision to purchasers |
|  | peaches-conventional\_passthrough-annual; peaches-organic\_passthrough-annual; peaches-seconds\_passthrough-annual | lbs | Tracking inventory managed by the farmer for different commodities, accumulating any product handled by the farmer. This is helpful in plotting |
|  | peaches-seconds\_contract-space-available; peaches-iqf\_contract-space-available | lbs | The space available to contract for peach sales |
|  |  |  |  |
| **Decision-making** | |  |  |
|  | peaches-seconds\_my-incoming-total; peaches-iqf\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 11.** Summary of state variables for peach processors.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id; farm-id | Number | A numeric identifier of the farmer, and a secondary identifier |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the farmer shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final farm-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | peaches-seconds\_price-to-sell-wholesale; peaches-iqf\_price-to-sell | $ | The price to sell the indicated commodity |
|  | peaches-seconds\_price-to-buy; peaches-iqf-price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory** | |  |  |
|  | peaches-seconds\_inventory-current; peaches-seconds\_inventory-maximum; peaches-iqf \_inventory-current; peaches-iqf\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | peaches-seconds- promised; peaches-iqf-promised | lbs | The amount of wheat of the commodity indicated that have been contracted for provision to purchasers |
|  | peaches-seconds\_contract-space-available; peaches-iqf\_contract-space-available | lbs | The space available to contract for peach sales |
|  |  |  |  |
| **Decision-making** | |  |  |
|  | peaches-seconds\_my-incoming-total; peaches-iqf\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 12.** Summary of state variables for peach alternative markets.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id; farm-id | Number | A numeric identifier of the farmer, and a secondary identifier |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the farmer shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final farm-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | peaches-conventional\_price-to-sell-wholesale; peaches-organic\_price-to-sell | $ | The price to sell the indicated commodity |
|  | peaches-conventional\_price-to-buy; peaches-organic-price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory** | |  |  |
|  | peaches-conventional\_inventory-current; peaches-conventional\_inventory-maximum; peaches-organic \_inventory-current; peaches-organic\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | peaches-conventional-promised; peaches-organic-promised | lbs | The amount of wheat of the commodity indicated that have been contracted for provision to purchasers |
|  | peaches-conventional\_contract-space-available; peaches-organic\_contract-space-available | lbs | The space available to contract for peach sales |
|  |  |  |  |
| **Decision-making** | |  |  |
|  | peaches-conventional\_my-incoming-total; peaches-organic\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 13.** Summary of state variables for beef producers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id; farm-id | Number | A numeric identifier of the rancher, and a secondary identifier |
|  | agent-name | Text | A string identifying the agent group |
|  | home-county | Text | The identifier of the county of the rancher |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the rancher shows selling loyalty |
|  | category; specialty-product | Number | Identifiers that indicate categories of rancher or whether a product produced is differentiated |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final ranch-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | beef-conventional-current-value; beef-grassfed-current-value; beef-colorado-source-id-current-value; beef-animal-welfare-current-value; total-beef-current-value | $ | The value of the commodities indicated |
|  | beef-conventional\_price-to-sell; beef-grassfed\_price-to-sell; beef-colorado-source-id\_price-to-sell; beef-animal-welfare\_price-to-sell; cull-cows\_price-to-sell | $ | The price to sell the indicated commodity |
|  | year-calf-revenue; year-cull-revenue | $ | Annual revenue from the animal groups indicated |
|  | year-calf-sale | Number | The number of calves sold in the year |
|  |  |  |  |
| **Production** | |  |  |
|  | beef-conventional-current-production; beef-grassfed-current-production; beef-colorado-source-id-current-production; beef-animal-welfare-current-production | lbs | Production of the commodities indicated |
|  | cows-that-calved; total-cows; total-cows-baseline; replacement-cows; |  |  |
|  | forage-need | lbs | Forage needed to support the cattle owned |
|  | stocking-rate | Number | The number of animals per unit area |
|  | year-calves-weaned | Number | The number of calves weaned in the year |
|  | owns-irrigated; leases-grazing; irrig-owned; non-irrig-owned; total-ha-grazed | ha | Areas in irrigated forage, leased lands for grazing, other types of lands used, and total hectares grazed |
|  | input-grazing-ratio; input-grazing-ratio-private; input-grazing-ratio-public; input-hay-ratio | Number | Ratios used to reflect sources of food for cattle |
|  | ranch-average-precip | mm | Average annual precipitation for the ranch |
|  | year-rainfall-ratio | Number | The ratio of rainfall of the year to the average, indicating drought stress |
|  | beef-ranchers\_yield-estimated\_conventional; beef-ranchers\_yield-estimated\_grassfed; beef-ranchers\_yield-estimated\_colorado-source-id; beef-ranchers\_yield-estimated\_animal-welfare | lbs | Estimated yield for the commodities indicated |
|  | patches-in-beef | Number | The number of patches used by the rancher for beef production |
|  | year-cull-cows-conventional; year-cull-cows-grassfed; year-cull-cows-colorado-source-id; year-cull-cows-animal-welfare | Number | The number of animals culled for the different beef commodities |
|  |  |  |  |
| **Inventory** | |  |  |
|  | beef-conventional\_inventory-current; beef-conventional\_inventory-maximum; beef-grassfed\_inventory-current; beef-grassfed\_inventory-maximum; beef-colorado-source-id\_inventory-current; beef-colorado-source-id\_inventory-maximum; beef-animal-welfare\_inventory-current; beef-animal-welfare\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | cull-cows\_available; year-cull-cows | Number | The number of cattle to be culled that are available or accumulated in the year |
|  | beef-conventional\_passthrough-annual; beef-grassfed\_passthrough-annual; beef-colorado-source-id\_passthrough-annual; beef-animal-welfare\_passthrough-annual | lbs | Tracking inventory managed by the rancher for different commodities, accumulating any product handled by the rancher. This is helpful in plotting |
|  | beef-animal-welfare\_promised; beef-conventional\_promised; beef-grassfed\_promised; beef-colorado-source-id\_promised; cull-cows\_promised | lbs | Quantities of beef commodities promised through contracts |
|  |  |  |  |
| **Decision-making** | |  |  |
|  | transition-likelihood-score; tls-initial; tls-final; tls-change | Index | A transition likelihood score indicating the likelihood that a rancher will transition some parcels from conventional to organic methods. Initial values, final values, and the change in values are tracked as well |
|  | male-gender | Logical | Whether or not the farmer is a male |
|  | first-time-farmer | Logical | Whether or not the person is a first-time farmer |
|  | young-farmer | Logical | Whether or not the farmer is young |
|  | not-white | Logical | Whether or not the farmer is white |
|  | distance-to-craig | Number | The linear distance to the city of Craig, Coloado |
|  | farm-size | Rank | The relative size of the ranch |
|  | farm-transitioning | Logical | Whether or not the ranch is transitioning to organic production methods |
|  | this-year-mediocre-production-year; last-year-mediocre-production-year; this-year-bad-production-year; last-year-bad-production-year | Logical | Used to affect transition likelihood scores, based on previous production amounts |
|  |  |  |  |

**Table 14.** Summary of state variables for beef packing houses.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id; farm-id | Number | A numeric identifier of the rancher, and a secondary identifier |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the rancher shows selling loyalty |
|  | category; specialty-product | Number | Identifiers that indicate categories of rancher or whether a product produced is differentiated |
|  | small-size | Number | A flag to indicate whether a packing house is small in scope. |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final ranch-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | beef-conventional-current-value; beef-grassfed-current-value; beef-colorado-source-id-current-value; beef-animal-welfare-current-value; total-beef-current-value | $ | The value of the commodities indicated |
|  | beef-conventional\_price-to-sell; beef-grassfed\_price-to-sell; beef-colorado-source-id\_price-to-sell; beef-animal-welfare\_price-to-sell; cull-cows\_price-to-sell | $ | The price to sell the indicated commodity |
|  | beef-conventional\_price-to-buy; beef-grassfed\_price-to-buy; beef-colorado-source-id\_price-to-buy; beef-animal-welfare\_price-to-buy; cull-cows\_price-to-buy | $ | The price to buy the indicated commodity |
|  | year-calf-revenue; year-cull-revenue | $ | Annual revenue from the animal groups indicated |
|  | year-calf-sale | Number | The number of calves sold in the year |
|  |  |  |  |
| **Production** | |  |  |
|  | beef-conventional-current-production; beef-grassfed-current-production; beef-colorado-source-id-current-production; beef-animal-welfare-current-production | lbs | Production of the commodities indicated |
|  | cows-that-calved; total-cows; total-cows-baseline; replacement-cows; |  |  |
|  | forage-need | lbs | Forage needed to support the cattle owned |
|  | stocking-rate | Number | The number of animals per unit area |
|  | year-calves-weaned | Number | The number of calves weaned in the year |
|  | owns-irrigated; leases-grazing; irrig-owned; non-irrig-owned; total-ha-grazed | ha | Areas in irrigated forage, leased lands for grazing, other types of lands used, and total hectares grazed |
|  | input-grazing-ratio; input-grazing-ratio-private; input-grazing-ratio-public; input-hay-ratio | Number | Ratios used to reflect sources of food for cattle |
|  | ranch-average-precip | mm | Average annual precipitation for the ranch |
|  | year-rainfall-ratio | Number | The ratio of rainfall of the year to the average, indicating drought stress |
|  | beef-ranchers\_yield-estimated\_conventional; beef-ranchers\_yield-estimated\_grassfed; beef-ranchers\_yield-estimated\_colorado-source-id; beef-ranchers\_yield-estimated\_animal-welfare | lbs | Estimated yield for the commodities indicated |
|  | patches-in-beef | Number | The number of patches used by the rancher for beef production |
|  | year-cull-cows-conventional; year-cull-cows-grassfed; year-cull-cows-colorado-source-id; year-cull-cows-animal-welfare | Number | The number of animals culled for the different beef commodities |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | beef-conventional\_inventory-current; beef-conventional\_inventory-maximum; beef-grassfed\_inventory-current; beef-grassfed\_inventory-maximum; beef-colorado-source-id\_inventory-current; beef-colorado-source-id\_inventory-maximum; beef-animal-welfare\_inventory-current; beef-animal-welfare\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | beef-animal-welfare\_promised; beef-conventional\_promised; beef-grassfed\_promised; beef-colorado-source-id\_promised; cull-cows\_promised | lbs | Quantities of beef commodities promised through contracts |
|  | beef-conventional\_contract-space-available; beef-grassfed\_contract-space-available; beef-colorado-source-id\_contract-space-available; beef-animal-welfare\_contract-space-available | lbs | Contracting space available, used in fulfilling contract sales |
|  | beef-conventional\_my-incoming-total; beef-grassfed\_my-incoming-total; beef-colorado-source-id\_my-incoming-total; beef-animal-welfare\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 15.** Summary of state variables for distributors

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the distributor |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the rancher shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final ranch-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | out-of-state-french-fry\_price-to-sell; potatoes-russet-conventional-fresh\_price-to-sell; potatoes-russet-organic-fresh\_price-to-sell; potatoes-purple-conventional-fresh\_price-to-sell; potatoes-purple-organic-fresh\_price-to-sell; potatoes-russet-conventional-fresh-small\_price-to-sell; potatoes-russet-conventional-processed\_price-to-sell; potatoes-russet-organic-processed\_price-to-sell;  potatoes-purple-conventional-processed\_price-to-sell; potatoes-purple-organic-processed\_price-to-sell; wheat-hrw-conventional-flour\_price-to-sell; wheat-hrw-organic-flour\_price-to-sell; wheat-snowmass-conventional-flour\_price-to-sell; wheat-hrw-conventional-bread\_price-to-sell; wheat-hrw-organic-bread\_price-to-sell; wheat-snowmass-conventional-bread\_price-to-sell;  peaches-out-of-state\_price-to-sell; peaches-conventional\_price-to-sell; peaches-organic\_price-to-sell; peaches-seconds\_price-to-sell; peaches-iqf\_price-to-sell; beef-conventional\_price-to-sell; beef-grassfed\_price-to-sell; beef-colorado-source-id\_price-to-sell;  beef-animal-welfare\_price-to-sell | $ | The price to sell the indicated commodity |
|  | out-of-state-french-fry\_price-to-buy; potatoes-russet-conventional-fresh\_price-to-buy; potatoes-russet-organic-fresh\_price-to-buy; potatoes-purple-conventional-fresh\_price-to-buy; potatoes-purple-organic-fresh\_price-to-buy;  potatoes-russet-conventional-fresh-small\_price-to-buy; potatoes-russet-conventional-processed\_price-to-buy; potatoes-russet-organic-processed\_price-to-buy; potatoes-purple-conventional-processed\_price-to-buy;  potatoes-purple-organic-processed\_price-to-buy; wheat-hrw-conventional-flour\_price-to-buy; wheat-hrw-organic-flour\_price-to-buy; wheat-snowmass-conventional-flour\_price-to-buy; wheat-hrw-conventional-bread\_price-to-buy;  wheat-hrw-organic-bread\_price-to-buy; wheat-snowmass-conventional-bread\_price-to-buy; peaches-out-of-state\_price-to-buy; peaches-conventional\_price-to-buy; peaches-organic\_price-to-buy;  peaches-seconds\_price-to-buy; peaches-iqf\_price-to-buy; beef-conventional\_price-to-buy; beef-grassfed\_price-to-buy; beef-colorado-source-id\_price-to-buy; beef-animal-welfare\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | out-of-state-french-fry\_inventory-current; out-of-state-french-fry\_inventory-maximum; potatoes-russet-conventional-fresh\_inventory-current; potatoes-russet-conventional-fresh\_inventory-maximum; potatoes-russet-organic-fresh\_inventory-current; potatoes-russet-organic-fresh\_inventory-maximum; potatoes-purple-conventional-fresh\_inventory-current; potatoes-purple-conventional-fresh\_inventory-maximum; potatoes-purple-organic-fresh\_inventory-current; potatoes-purple-organic-fresh\_inventory-maximum; potatoes-russet-conventional-fresh-small\_inventory-current; potatoes-russet-conventional-fresh-small\_inventory-maximum; potatoes-russet-conventional-processed\_inventory-current; potatoes-russet-conventional-processed\_inventory-maximum;  potatoes-russet-organic-processed\_inventory-current; potatoes-russet-organic-processed\_inventory-maximum; potatoes-purple-conventional-processed\_inventory-current; potatoes-purple-conventional-processed\_inventory-maximum; potatoes-purple-organic-processed\_inventory-current; potatoes-purple-organic-processed\_inventory-maximum; wheat-hrw-conventional-flour\_inventory-current; wheat-hrw-conventional-flour\_inventory-maximum; wheat-hrw-organic-flour\_inventory-current; wheat-hrw-organic-flour\_inventory-maximum; wheat-snowmass-conventional-flour\_inventory-current; wheat-snowmass-conventional-flour\_inventory-maximum; wheat-hrw-conventional-bread\_inventory-current; wheat-hrw-conventional-bread\_inventory-maximum; wheat-hrw-organic-bread\_inventory-current; wheat-hrw-organic-bread\_inventory-maximum; wheat-snowmass-conventional-bread\_inventory-current; wheat-snowmass-conventional-bread\_inventory-maximum; peaches-out-of-state\_inventory-current; peaches-out-of-state\_inventory-maximum; peaches-conventional\_inventory-current; peaches-conventional\_inventory-maximum; peaches-organic\_inventory-current; peaches-organic\_inventory-maximum; peaches-seconds\_inventory-current; peaches-seconds\_inventory-maximum; peaches-iqf\_inventory-current; peaches-iqf\_inventory-maximum; beef-conventional\_inventory-current; beef-conventional\_inventory-maximum; beef-grassfed\_inventory-current; beef-grassfed\_inventory-maximum; beef-colorado-source-id\_inventory-current; beef-colorado-source-id\_inventory-maximum; beef-animal-welfare\_inventory-current; beef-animal-welfare\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | potatoes-russet-conventional-fresh-promised; potatoes-russet-organic-fresh-promised; potatoes-purple-conventional-fresh-promised; potatoes-purple-organic-fresh-promised;  potatoes-russet-conventional-fresh-small-promised; potatoes-russet-conventional-processed-promised; potatoes-russet-organic-processed-promised; potatoes-purple-conventional-processed-promised;  potatoes-purple-organic-processed-promised; wheat-hrw-conventional-flour-promised; wheat-hrw-organic-flour-promised; wheat-snowmass-conventional-flour-promised;  wheat-hrw-conventional-bread-promised; wheat-hrw-organic-bread-promised; wheat-snowmass-conventional-bread-promised; peaches-out-of-state\_promised;  peaches-conventional\_promised; peaches-organic\_promised; peaches-seconds\_promised; peaches-iqf\_promised; beef-conventional\_promised; beef-grassfed\_promised; beef-colorado-source-id\_promised; beef-animal-welfare\_promised; | lbs | Quantities of commodities promised through contracts |
|  | potatoes-russet-conventional-fresh\_contract-space-available; potatoes-russet-organic-fresh\_contract-space-available; potatoes-purple-conventional-fresh\_contract-space-available;  potatoes-purple-organic-fresh\_contract-space-available; potatoes-russet-conventional-fresh-small\_contract-space-available; potatoes-russet-conventional-processed\_contract-space-available;  potatoes-russet-organic-processed\_contract-space-available; potatoes-purple-conventional-processed\_contract-space-available; potatoes-purple-organic-processed\_contract-space-available;  wheat-hrw-conventional-flour\_contract-space-available; wheat-hrw-organic-flour\_contract-space-available; wheat-snowmass-conventional-flour\_contract-space-available;  wheat-hrw-conventional-bread\_contract-space-available; wheat-hrw-organic-bread\_contract-space-available; wheat-snowmass-conventional-bread\_contract-space-available;  peaches-out-of-state\_contract-space-available; peaches-conventional\_contract-space-available; peaches-organic\_contract-space-available;  peaches-seconds\_contract-space-available; peaches-iqf\_contract-space-available; beef-conventional\_contract-space-available;  beef-grassfed\_contract-space-available; beef-colorado-source-id\_contract-space-available; beef-animal-welfare\_contract-space-available; | lbs | Contracting space available, used in fulfilling contract sales |
|  | potatoes-russet-conventional-fresh\_my-incoming-total; potatoes-russet-organic-fresh\_my-incoming-total; potatoes-purple-conventional-fresh\_my-incoming-total;  potatoes-purple-organic-fresh\_my-incoming-total; potatoes-russet-conventional-fresh-small\_my-incoming-total; potatoes-russet-conventional-processed\_my-incoming-total;  potatoes-russet-organic-processed\_my-incoming-total; potatoes-purple-conventional-processed\_my-incoming-total; potatoes-purple-organic-processed\_my-incoming-total;  wheat-hrw-conventional-flour\_my-incoming-total; wheat-hrw-organic-flour\_my-incoming-total; wheat-snowmass-conventional-flour\_my-incoming-total;  wheat-hrw-conventional-bread\_my-incoming-total; wheat-hrw-organic-bread\_my-incoming-total; wheat-snowmass-conventional-bread\_my-incoming-total;  peaches-out-of-state\_my-incoming-total; peaches-conventional\_my-incoming-total; peaches-organic\_my-incoming-total;  peaches-seconds\_my-incoming-total; peaches-iqf\_my-incoming-total; beef-conventional\_my-incoming-total;  beef-grassfed\_my-incoming-total; beef-colorado-source-id\_my-incoming-total; beef-animal-welfare\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**Table 16.** Summary of state variables for Denver Public Schools buyers.

|  |  |  |  |
| --- | --- | --- | --- |
| **Category** | |  |  |
|  | **Variable(s)** | **Units** | **Description** |
|  |  |  |  |
| **Identifiers** | |  |  |
|  | id | Number | A numeric identifier of the distributor |
|  | agent-name | Text | A string identifying the agent group |
|  | loyalty-to-buyer-id | Number | The identifier of a buyer to which the rancher shows selling loyalty |
|  |  |  |  |
| **Assets, Incomes, and Costs** | |  |  |
|  | assets; assets-initial; assets-final; assets-change | $ | Current, initial, and final ranch-based assets, as well as changes in assets |
|  | profitability-average | $ | Annual average profitability |
|  | out-of-state-french-fry\_price-to-buy; potatoes-russet-conventional-fresh\_price-to-buy; potatoes-russet-organic-fresh\_price-to-buy; potatoes-purple-conventional-fresh\_price-to-buy; potatoes-purple-organic-fresh\_price-to-buy;  potatoes-russet-conventional-fresh-small\_price-to-buy; potatoes-russet-conventional-processed\_price-to-buy; potatoes-russet-organic-processed\_price-to-buy; potatoes-purple-conventional-processed\_price-to-buy;  potatoes-purple-organic-processed\_price-to-buy; wheat-hrw-conventional-flour\_price-to-buy; wheat-hrw-organic-flour\_price-to-buy; wheat-snowmass-conventional-flour\_price-to-buy; wheat-hrw-conventional-bread\_price-to-buy;  wheat-hrw-organic-bread\_price-to-buy; wheat-snowmass-conventional-bread\_price-to-buy; peaches-out-of-state\_price-to-buy; peaches-conventional\_price-to-buy; peaches-organic\_price-to-buy;  peaches-seconds\_price-to-buy; peaches-iqf\_price-to-buy; beef-conventional\_price-to-buy; beef-grassfed\_price-to-buy; beef-colorado-source-id\_price-to-buy; beef-animal-welfare\_price-to-buy | $ | The price to buy the indicated commodity |
|  |  |  |  |
| **Inventory and Contracting** | |  |  |
|  | out-of-state-french-fry\_inventory-current; out-of-state-french-fry\_inventory-maximum; potatoes-russet-conventional-fresh\_inventory-current; potatoes-russet-conventional-fresh\_inventory-maximum; potatoes-russet-organic-fresh\_inventory-current; potatoes-russet-organic-fresh\_inventory-maximum; potatoes-purple-conventional-fresh\_inventory-current; potatoes-purple-conventional-fresh\_inventory-maximum; potatoes-purple-organic-fresh\_inventory-current; potatoes-purple-organic-fresh\_inventory-maximum; potatoes-russet-conventional-fresh-small\_inventory-current; potatoes-russet-conventional-fresh-small\_inventory-maximum; potatoes-russet-conventional-processed\_inventory-current; potatoes-russet-conventional-processed\_inventory-maximum;  potatoes-russet-organic-processed\_inventory-current; potatoes-russet-organic-processed\_inventory-maximum; potatoes-purple-conventional-processed\_inventory-current; potatoes-purple-conventional-processed\_inventory-maximum; potatoes-purple-organic-processed\_inventory-current; potatoes-purple-organic-processed\_inventory-maximum; wheat-hrw-conventional-flour\_inventory-current; wheat-hrw-conventional-flour\_inventory-maximum; wheat-hrw-organic-flour\_inventory-current; wheat-hrw-organic-flour\_inventory-maximum; wheat-snowmass-conventional-flour\_inventory-current; wheat-snowmass-conventional-flour\_inventory-maximum; wheat-hrw-conventional-bread\_inventory-current; wheat-hrw-conventional-bread\_inventory-maximum; wheat-hrw-organic-bread\_inventory-current; wheat-hrw-organic-bread\_inventory-maximum; wheat-snowmass-conventional-bread\_inventory-current; wheat-snowmass-conventional-bread\_inventory-maximum; peaches-out-of-state\_inventory-current; peaches-out-of-state\_inventory-maximum; peaches-conventional\_inventory-current; peaches-conventional\_inventory-maximum; peaches-organic\_inventory-current; peaches-organic\_inventory-maximum; peaches-seconds\_inventory-current; peaches-seconds\_inventory-maximum; peaches-iqf\_inventory-current; peaches-iqf\_inventory-maximum; beef-conventional\_inventory-current; beef-conventional\_inventory-maximum; beef-grassfed\_inventory-current; beef-grassfed\_inventory-maximum; beef-colorado-source-id\_inventory-current; beef-colorado-source-id\_inventory-maximum; beef-animal-welfare\_inventory-current; beef-animal-welfare\_inventory-maximum | lbs | The current or maximum inventory of the indicated commodity |
|  | potatoes-russet-conventional-fresh-promised; potatoes-russet-organic-fresh-promised; potatoes-purple-conventional-fresh-promised; potatoes-purple-organic-fresh-promised;  potatoes-russet-conventional-fresh-small-promised; potatoes-russet-conventional-processed-promised; potatoes-russet-organic-processed-promised; potatoes-purple-conventional-processed-promised;  potatoes-purple-organic-processed-promised; wheat-hrw-conventional-flour-promised; wheat-hrw-organic-flour-promised; wheat-snowmass-conventional-flour-promised;  wheat-hrw-conventional-bread-promised; wheat-hrw-organic-bread-promised; wheat-snowmass-conventional-bread-promised; peaches-out-of-state\_promised;  peaches-conventional\_promised; peaches-organic\_promised; peaches-seconds\_promised; peaches-iqf\_promised; beef-conventional\_promised; beef-grassfed\_promised; beef-colorado-source-id\_promised; beef-animal-welfare\_promised; | lbs | Quantities of commodities promised through contracts |
|  | potatoes-russet-conventional-fresh\_contract-space-available; potatoes-russet-organic-fresh\_contract-space-available; potatoes-purple-conventional-fresh\_contract-space-available;  potatoes-purple-organic-fresh\_contract-space-available; potatoes-russet-conventional-fresh-small\_contract-space-available; potatoes-russet-conventional-processed\_contract-space-available;  potatoes-russet-organic-processed\_contract-space-available; potatoes-purple-conventional-processed\_contract-space-available; potatoes-purple-organic-processed\_contract-space-available;  wheat-hrw-conventional-flour\_contract-space-available; wheat-hrw-organic-flour\_contract-space-available; wheat-snowmass-conventional-flour\_contract-space-available;  wheat-hrw-conventional-bread\_contract-space-available; wheat-hrw-organic-bread\_contract-space-available; wheat-snowmass-conventional-bread\_contract-space-available;  peaches-out-of-state\_contract-space-available; peaches-conventional\_contract-space-available; peaches-organic\_contract-space-available;  peaches-seconds\_contract-space-available; peaches-iqf\_contract-space-available; beef-conventional\_contract-space-available;  beef-grassfed\_contract-space-available; beef-colorado-source-id\_contract-space-available; beef-animal-welfare\_contract-space-available; | lbs | Contracting space available, used in fulfilling contract sales |
|  | potatoes-russet-conventional-fresh\_my-incoming-total; potatoes-russet-organic-fresh\_my-incoming-total; potatoes-purple-conventional-fresh\_my-incoming-total;  potatoes-purple-organic-fresh\_my-incoming-total; potatoes-russet-conventional-fresh-small\_my-incoming-total; potatoes-russet-conventional-processed\_my-incoming-total;  potatoes-russet-organic-processed\_my-incoming-total; potatoes-purple-conventional-processed\_my-incoming-total; potatoes-purple-organic-processed\_my-incoming-total;  wheat-hrw-conventional-flour\_my-incoming-total; wheat-hrw-organic-flour\_my-incoming-total; wheat-snowmass-conventional-flour\_my-incoming-total;  wheat-hrw-conventional-bread\_my-incoming-total; wheat-hrw-organic-bread\_my-incoming-total; wheat-snowmass-conventional-bread\_my-incoming-total;  peaches-out-of-state\_my-incoming-total; peaches-conventional\_my-incoming-total; peaches-organic\_my-incoming-total;  peaches-seconds\_my-incoming-total; peaches-iqf\_my-incoming-total; beef-conventional\_my-incoming-total;  beef-grassfed\_my-incoming-total; beef-colorado-source-id\_my-incoming-total; beef-animal-welfare\_my-incoming-total | lbs | Storing inventory coming into contracting and exchange, and used in mathematics to account for changes in totals stored |
|  |  |  |  |

**3. Process overview and scheduling**

The FFAR-ABModel uses a weekly time-step, and those steps control scheduling of commodity production, harvest, sale, processing, and consumption. Individual commodity models (i.e., potato, wheat, beef, and peaches) are joined by an umbrella module. In this version of the application, the umbrella module can call procedures that simulate any of the commodities, but multiple commodities cannot be simulated at the same time. Following initialization (see Section 5), a loop is executed that runs until a maximum number of years is completed, set at 10 years now. Scheduling for commodities proceeds in that loop, beginning in the first year in late summer:

***Potatoes***

In week 31 (early August), products that are in storage beyond their period of expiration are cleared (*reset-storage-potatoes*). That same week (except in the first year), a check is performed to see if potato production using conventional and organic means are apt to be profitable (*profitability-check-potatoes*). The outcome of those calculations may affect a number of patches depending upon producer decisions, converting them from conventional production to transition toward organic production. Patches producing potatoes conventionally may be transitioned to organic production. If it is not economically viable for medium and large farms producing potatoes organically, the farms transition back to conventional production (*transition-patches-potatoes*). The status of potatoes shifting to organic production is updated (*update-organic-potatoes* calling *into-organic-potatoes*). The display of status for potato patches are then updated based on the setting of a chooser on the program interface (*update-display-potatoes*).

Contracts for potato sales are cleared and then reestablished in weeks 5 (late January), week 18 (early May), and week 31 (early August). This includes clearing the amount of potatoes promised to those with whom contracts have been made (*reset-promised-amounts-potatoes*), resetting the table structures used to store contract information (*reset-tables-potatoes*), and reestablishing contracts (*set-contracts-potatoes*)

In week 36 (early September), potatoes are harvested (*produce-potatoes*), which includes a suite of procedures called. Potato yields for patches for the commodities in question are set based on spatial surfaces showing annual yield estimates from the ecosystem model (*set-potato-yield*) (see Section 6). Based on the yields read in, the simulation assesses whether it is a normal production year or a bad production year (*update-bad-production-year-potatoes*), which is used in assessing the likelihood a farmer will transition to organic production. The allocation of potato types is set, currently 80% russet-canela and 20% purple potatoes for all farmers (*set-potato-farm-crop-allocation*). Rotation of potato production with barley is set, with croplands moving through potato production, a transitional year, and a year of barley production (*set-potato-barley-rotation*). The annual production for each farmer is then calculated based upon the specific patches the farmer owns or leases (*calculate-annual-production-potatoes*). Production is then summed for all farmers (*update-potato-production*).

In week 7 (mid-February), week 19 (mid-May), and week 37 (mid-September), potato contracts are filled (*fulfill-contracts-potatoes*). In this procedure, contracts specific to scenarios are included in optional clauses processed, such that only specific contract processes are processed if the given scenario is enabled. Potato processors manage their inventory (*process-potatoes*), then contracts between processors and distributors are fulfilled, and between the distributor and Denver Public Schools buyer (*fulfill-processor-contracts-potatoes*).

A final clause concludes scheduling for potatoes. In the baseline scenario for that commodity, during school periods (fall term with a break: weeks 34 to 46, weeks 48 to 52, and spring term with a break: weeks 3 to 12, and week 14 to 21), the Denver Public School System buyer purchases French fries from out-of-state suppliers.

***Wheat***

In week 31 (early August), depending upon if the transition likelihood scores for wheat farms are above 0.5, they may transition a number of patches to organic production (*transition-patches-wheat*). Patches that are selected for that change are indicated as being in transition, with patches moving through transition before being certified organic. For medium or large wheat farms, or for small farms or medium farms with alternative management, if their transition likelihood scores are below 0.5, patches that are in organic production will revert to conventional production. The status of patches producing wheat organically are updated (*update-organic-wheat*), with those in transition converted to organic after three years. The display showing the status of wheat patches is then updated based on the setting of a chooser on the program interface (*update-display-wheat*).

Contracts for wheat sales are cleared and then reestablished in weeks 5 (late January), week 18 (early May), and week 31 (early August). This includes clearing the amount of wheat promised to those with whom contracts have been made (*reset-promised-amounts-wheat*), resetting the table structures used to store contract information (*reset-tables-wheat*), and reestablishing contracts (*set-contracts-wheat*)

In week 36 (early September), wheat is produced (*produce-wheat*) through a series of procedures being called. Yields are set for the different wheat commodities, hard red winter conventional, hard red winter organic, and snowmass (*set-wheat-yield*) based on spatially-explicit estimates of wheat yields derived in the ecosystem production models used. The allocation of wheat production for each farmer is then set (*set-wheat-farm-crop-allocation*), with 5% of production in snowmass wheat. The rotation used by each farmer is set (*set-wheat-rotation*) based on their rotation schedule assigned at initialization (see Section 5), either a wheat-fallow rotation or a corn-wheat-fallow rotation. Wheat farmers then have their annual production calculated (*calculate-annual-production-wheat*). Patches in each of the commodity types are counted, costs of production for those patches are calculated, and then production and inventories are updated. *Produce-wheat* ends with calculations to determine if production was a mediocre or bad year (*update-bad-production-year-wheat*). Whether or not years are mediocre or bad in wheat production influence the transition likelihood scores used in deciding production methods. Summaries of wheat production across all farmers are made (*update-wheat-production*).

In week 6 (early February), week 19 (mid-May) or week 37 (mid-September), wheat contracts are filled (*fulfill-contracts-wheat*) through calls to procedures for each of the commodities, with farmers selling to elevators, elevators selling to mills, mills to the distributor or bakers, and bakers to the distributor. Then, based on scenarios selected, different commodities are sold by distributors to the Denver Public School buyer. Mills convert the wheat they have purchased to flour (*make-flour*) and sell flour to bakers (*fulfill-mill-contracts-wheat*). Similarly, bread producers make bread from the wheat they receive (*make-bread*) and sell that to the distributor (*fulfill-baker-contracts-wheat*) which in turn sells that bread to the DPS buyer.

***Beef***

As one may anticipate, beef production is unique compared to the growth of crops like potatoes and wheat, or even the perennial crop peaches. Activities all happen in week 31, early August. First, beef storage is reset to clear product past its expiration (*reset-storage-beef*). That procedure in turn calls

*reset-storage\_beef*; note the underscore in the second term). This is done for beef ranchers, beef packing houses, the distributor, and the Denver School System buyer, with *reset-storage\_beef* resetting the appropriate variables based on the type of agent passed to it. Items set to zeros or to appropriate values include assets, current inventories, and a tracker of beef from outside the system.

The precipitation for the year is set (*set-precip*) based on spatial surfaces that report annual precipitation from 2001 to 2010, with the year used moving sequentially through the list. The ratio of precipitation to average precipitation is calculated and the number of cows that calve is set, with 90% of cows giving birth.

Forage production is simulated for each non-irrigated patch based on the percent of average annual precipitation for the patch multiplied by the mean forage production (*set-forage*). The display is then updated (*update-display-beef*), with the thematic layer shown on the display determined by what a chooser on the interface is set. In *ranch-stats*, calculations proceed based on the yearly rainfall compared to the long-term average. Proportions and production estimates vary in modest ways in wet, normal, and dry years. The outcome of this procedure includes calculation of numbers of cows culled for different production pathways (i.e., conventional, grass-fed, Colorado-sourced, animal welfare), calves weaned, cows replaced, yearlings for sale, and the total number of cows, plus others. Following calculation of production of beef, the summed production across all ranches is calculated for the given year and in total, for the different commodities (*update-beef-production*).

Beef packing houses conduct sales in a single procedure, rather than using the contracting methods outlined for other commodities (*sell-beef-spot-market*). Packing houses vary by size, and expected beef needed is set separately for small or larger packing houses for the different commodities (i.e., baseline, grass-fed, Colorado sourced, or animal welfare). Then the beef needed by each packer is purchased from ranchers. In this version a bidding process is conducted with the winning bidder selected based on the highest transition likelihood score. That rancher sells the cattle that are available, with the price set based on a random normal draw from a defined distribution. Inventories and other summation variables are updated.

Still within the *sell-beef-spot-market* procedure, distributor processes are tracked if the scenario is the baseline. Distributors purchase conventionally produced beef from packing houses. Assets and inventories are then adjusted to track the exchange.

Lastly in the *sell-beef-spot-market* procedure, attributes for the Denver Public School buyer are updated. Pathways through this section are dependent upon the scenario set (i.e., baseline, grass-fed, Colorado sourced, or animal welfare). Appropriate sellers are selected based on the scenario, then the amount of beef needed and the amount available from the seller are set. Prices are set based on random normal draws from set distributions, then money is exchanged, assets are tracked, and inventories adjusted.

***Peaches***

In week 26 (early July), peach farmers calculate their likely profitability to compare conventional versus organic pathways of production (*profitability-check-peaches*). Estimated yields for the different products and their costs of production and likely revenues are calculated and compared. Small- and medium-sized orchards with predicted organic profits higher than conventional production update their transition likelihood scores to be higher, and the opposite is true if lower. Patches are then transitioned (*transition-patches-peaches*) if the transition livelihood score is > 0.5. If more than 25% of patches for a given farmer are in organic production, the rest of their ownership is put into transition. If a transition score is < 0.5 and the farm is large, or if medium-sized and not in alternative management, or small and not in alternative management and not full organic, patches in organic production are converted back to conventional methods. If a large farm, a patch is put toward conversion to organic without adjustments to the transition likelihood score. Patches growing peaches are then checked to see if they should be moved from a transitional status to organic (*update-organic-peaches*). The display is then updated (*update-display-peaches*) based on settings indicated in the simulation interface.

Peaches that are promised to contract holders are cleared in the same week (week 26) (*reset-promised-amounts-peaches*) and tables storing contracts are reset to clear them (*reset-tables-peaches*). Contracts are then re-established (*set-contracts-peaches*). Contracts are established for conventional, organic, and seconds peaches, linking peach farmers and packers, packers selling to distributors, and farmers selling to alternative markets, processors, or the Denver School System buyer.

In the following week (week 27), peaches are produced (*produce-peaches*). Yields are set (*set-peach-yield*), which in this version is done through random normal selection from designated distributions. Then, as for other commodities, a procedure is used to determine if the period is a mediocre or bad production year (*update-bad-production-year-peach*). Annual production for individual farmers is calculated (*calculate-annual-production-peaches*), where patches in the different commodities are counted and costs of production calculated. Assets are updated and production is summed across the patches, and then inventories are updated. Peach farmers update their inventories for each of the commodities (*update-inventories-peaches*). Lastly, production is updated across all peach farmers for each commodity annually and across the entire simulation (*update-peach-production*).

In week 29 (mid- to late-July), peach farmers update their inventories for each of the commodities (*update-inventories-peaches*). Peach contracts are then fulfilled (*fulfill-contracts-peaches*). If the scenario is the baseline, the Denver School System buyer purchases peaches from out-of-state (*dps-purchases-out-of-state-peaches*). Distributors sell these peaches but are a separate variable that tracks out-of-state purchase.

Peach processing ends with resetting of storage (*reset-storage-peaches*, which calls *reset-storage\_peaches*; note the underscore in the second term). This is done for peach farmers, alternative markets, peach processors, the distributor, and the DPS buyer. Items set to zeros or to appropriate values include assets, current inventories, and a tracker of peaches from outside the system.

***End of Simulation***

When the timing controlling the simulation reaches the end of the allotted time (i.e., *ticks* equals *max-ticks*), several concluding procedures are called. *Calculate-lca* is called, which contains a lengthy series of simple calculations that multiply the quantity of each of the commodities produced for both the annual sum (for the final year) and the entire model run times the lifecycle analysis coefficients calculated for the various responses (i.e., CO2 emissions and water use per unit of commodity). The lifecycle measures are therefore calculated by multiplying yields of products by their unit-of-production lifecycle coefficients.

The profitability of agents of various types are then calculated (*calculate-profitability*), based upon the type of model being simulated, plus the distributor and Denver Public School buyer. A procedure with an agent-set passed to it (*profitabilityCalculation*) calculates for each agent the final assets, subtracts the final assets from initial assets, and uses that value to calculate a profitability per year.

The final step is to calculate transition likelihood scores for the model that is enabled (*calculate-tls*). Like the profitability procedure, this procedure simply stores the final transition likelihood score for the farmers or ranchers and subtracts the initial value from the final transition likelihood score to identify the change.

Note: Output procedures to report the outcomes of a simulation would typically be in this procedure. They have been removed, however, to enable the use of NetLogo’s BehaviorSpace to produce the output needed.

***End of Year***

The procedure controlling execution of processes (i.e., the *Go* procedure) ends with a tick command, which advances the time of the simulation by one week.

**4. Design concepts**

**4.1. Basic principles**

Our goal was to simulate the full Colorado food system for potatoes, wheat, beef, and peaches to the degree possible given the nature of the effort. We did not wish to recreate complex and well-established crop production models or life-cycle assessments. Instead, we linked output from established models (e.g., DSSAT) tuned to simulate Colorado crop production to the FFAR-ABModel, and included coefficients determined by experts and standardized to unit amounts of commodities to calculate life-cycle results, here CO2 emissions and water use. We sought a spatially explicit simulation, linking real-world or randomly placed farms with production that is positionally accurate. Commodities were then produced by farmers or ranchers and flow from them to intermediate entities in the supply chain (e.g., farmer to shipper, to distributor, to mill, to bakery, and to the Denver Public School System). At each stage contracts are established and fulfilled, inventories are tracked, and assets and commodities are transferred. Commodities bought by the Denver Public School buyer are provided to the schools without those schools being represented directly.

The tool was programmed in NetLogo (Wilensky 1999). Programmatically, name conventions were established and followed to make the NetLogo code more easily interpreted. These include the use of dashes to separate words describing the general category of a variable followed by an underscore, and then the specific variable again using dashes as needed. For example, *wheat-hrw-organic-flour\_price-to-sell* is readily interpreted as the price asked to sell hard red winter wheat flour. Another unique aspect of the program was the extensive use of “Includes” in the code. This option is formally experimental in NetLogo but have been codified in the tool for many years. We therefore used this approach and created 19 NLS code sets that are included in the main NetLogo code or secondary modules.

**4.2. Emergence**

We have observed a stabilizing tendency in prices for commodities as a result of adjustments buyers make on the prices they offer. Producers convert their lands selectively to organic production based on their transition likelihood scores. The transition likelihood scores are responsive to the nature of the farmer or rancher and the farms that are geographically nearby.

**4.3. Adaptation**

Adaptations for producers are primarily through transition likelihood scores and the parameters that influence those (e.g., farm size, producer ethnicity, age) to shift their production to organic if it is likely to be more profitable than traditional production. That conversion may follow either pathway, with landscape cells converting (over three years) to organic production, or organic cells being converted back to traditional production. Farmers, and all supply chain participants with product to sell, will select the highest price from bids by buyers made to them. Buyers participate in a bidding contest, with the number of bidders variable and reflecting the seller’s market knowledge. Buyers are aware of other offers and will incrementally narrow their offered price.

Spatial relationships between agents generally play no role in this version of FFAR-ABModel. One case where it plays a role is in altering transition likelihood scores. Small farms or those that are medium and managed in an alternative way calculate their nearest three neighbors that share their attributes. The sum of the area in transition to organic for those neighboring farms can then influence the farm’s transition-likelihood-score. A similar procedure is done for large farms or medium farms traditionally managed, although any three nearest neighboring farms is used.

**4.4. Objectives**

The objective of all entities in the food supply chains (except the final Denver Public School buyer) is to maximize profit. The Denver Public School buyer seeks to minimize expenses while meeting the needs of school children. Profit motives notwithstanding, some producers are more likely to adopt organic production because of their personal traits (e.g., termed alternative farm managers in the model), perhaps at the expense of short-term profits.

**4.5. Prediction**

Producers predict the utility of converting to organic production through potential profitability calculations, which influence a transition likelihood score. The producers also alter those scores as a mean of prediction in response to mediocre or bad years. Contracting includes a form of prediction, where anticipated yields are allocated to contracted buyers before production has occurred. Contracts may not be completely filled if yields are lower than anticipated, or surplus crops may be sold out-of-state or as seconds if production exceeds the quantity needed by the Denver Public School System buyer.

**4.6. Sensing**

Farmers do more typical sensing when considering whether or not to transition to organic production, in that they are influenced by the transition scores of their nearest neighbors. Other sensing is as may be expected. Sellers of products are unaware of others selling the same product, but they have awareness of potential buyers participating in a bidding contest, given them the ability to select the highest offer and sell to that buyer. After sales are complete, buyers essentially compare prices and those with extreme prices adjust their prices toward the central value.

**4.7. Stochasticity**

A suite of years of crop production, or rainfall in beef modeling associated with forage production, are used in the model. Early in our work, the model selected a given year from the set available randomly to represent production in each year. That made comparisons between years and between scenarios difficult, because differences of interest in scenarios (e.g., buying potatoes as French fries from a national supplier versus buying Colorado-produced potatoes) were swamped by differences in yields across randomly-selected years. Instead, years used to estimate production are set as constant in the model (e.g., year 1 using 2000 yield estimates, year 2 using 2001 estimates, and year 3 using 2002 estimates).

Potatoes are positioned correctly in the model based on spatial mapping of center-pivot circles in San Louis Valley from CropScape (nassgeodata.gmu.edu/CropScape/). Farmers, however, are randomly assigned to the correctly placed patches based on numbers and farm areas reported in agricultural statistics and project-specific surveys. Wheat is represented similarly, with mapped wheat from CropScape used to place wheat-producing patches and farmers are defined and assigned randomly based on USDA Census of Agriculture statistics. Beef ranches are drawn randomly from grazable lands within Routt and Moffatt Counties, again with ranches themselves defined using random draws from counts or distributions defined based on agricultural statistics. Peaches are modeled similarly, with orchards drawn randomly from areas in Mesa and Delta Counties and the nature and number of orchards and their owners established using agricultural statistics.

Most other aspects of the model follow typical stochastic methods in NetLogo models. Agents are selected randomly from sets to create subsets, conduct bidding contests, and conduct analyses.

**4.8. Observation**

Plots and monitors on the FFAR-ABModel report amounts purchased and changes in assets and inventories, but these outputs are secondary. A procedure exists too that produces model output in files, but that is disabled and likely incomplete. The main output is specific to commodities and scenarios and incorporated in BehaviorSpace. For example, for the baseline potato model, BehaviorSpace output includes 300+ values produced, including the minimum, maximum, mean, and standard deviation for many responses. The tool seems to perform best when BehaviorSpace is used to run a single instance of the simulation for any number of repetitions needed, and using Table output. Results may then be summarized in Excel, R, or by other means.

**5. Initialization**

The Setup procedure initializes this and most models in NetLogo. The interface is cleared (*clear-all*) and timing of the model reset (*reset-ticks*). Then the procedure *setup-world* is executed. That includes calls to *create-map*, *setup-start-dates*, and *setup-gis-landscape*. *Create-map* sets the world size (i.e., 0 to 624 east-west, and 0 to 452 north-south, the dimensions of Colorado in km) and the patch size, which only designates how many pixels are used to display each patch. *Setup-start-dates* initializes the year to 1 and the week to 30, such that the model starts in late July.

A procedure, *setup-gis-landscape*, is then run to initialize the landscape. Spatial surfaces in ASCII format (see Section 6) are loaded by the program into the appropriate global variables (e.g., a county map, forage production in beef counties, patches where potatoes and wheat are grown, mean yields for types of potatoes). The ca. dozen surfaces are then loaded into patch variables, with these steps done using the GIS extension of NetLogo. The procedure then does initial painting of the spatial arena, and then assigns counties as appropriate to their various commodities (e.g., setting *county-wheat* or *county-peach* to 1, indicating those patches might produce the given commodity). The procedure ends with patches that are used in beef production having their mean forage production calculated and the annual forage production initialized to that value.

The *setup-static-names* procedure establishes some global variables (outside the typical place for globals to be defined) and sets their values to strings that may then be used in processing. The procedure *setup-parameters* has a lengthy list of *set* commands for global variables that control model execution. Note: Parameters that may be changed in scenarios. The procedure ends with either peach patches or wheat patches having their yields set for the appropriate commodities, and other global parameters set. A central procedure in initialization is *setup-dashboard*, which then calls the appropriate procedures depending upon the commodity group enabled. Farmers or ranchers are created (e.g., *setup-potato\_farmers*) and if ranchers are enabled, they are located on the landscape (*locate-beef\_ranchers*). In *setup-potato\_farmers*, for example, a number of potato farmers are created in Alamosa County, then another group in Rio Grande County, and the final set in Saguache County. In each case, identifiers are set for the farmer and for subsets of farmer groups, farm sizes are assigned. These steps allow for creation of potato farmers with appropriate numbers and farm areas to match observed agricultural statistics. For all potato farmers, their assets, maximum inventories of various commodities, and patches occupied by each farmer are set. Personal attributes for farmers are assigned (gender, first time farmer or not, young farmer or not, white or not, and whether the farm is for primary income), and these are used to assign initial transition likelihood scores. Lastly, the rotation used by each of the farmers is set. Wheat (*setup-wheat-farmers*) and peaches (*setup-peach-farmers*) have steps analogous to potatoes, although rotations do not apply to peaches. For beef ranchers (*setup-beef-ranchers*), there too groups of ranchers are created for different counties and at different scales (e.g., diversified or small scale) and their identifiers and cattle herd sizes set. Then for ranchers in each of the two categories, the ratio of public to private grazing is set and the total number of cows set. The attributes of individual ranchers are set for use in calculating transition likelihood indices. The procedure ends with a calculation of forage needed by the cows in the herd, providing each with 12.5 kg of feed daily, and setting initial assets.

Following the creation of producers, for potato farmers and wheat farmers, their crop allocations are set (*setup-potato-farm-crop-allocation*, *setup-wheat-farm-crop-allocation*). In this version, these routines simply assign each potato farmer to producing 80% russet canela potatoes and 20% purple potatoes, and wheat farmers producing 95% hard red winter wheat and 5% Snowmass wheat.

Agents are then created for the model that is active (i.e., *setup-potato-agents*, *setup-wheat-agents*, *setup-peach-agents*, *setup-beef-agents*). Others involved in the supply chain are then created. For example, for potatoes, repackers (*setup-potato\_repackers*), shippers (*setup-potato\_shippers*), and processors (*setup-potato\_processors*) are created. In each case, the appropriate number of agents are created and then their identifiers are set. Maximum inventories for the commodities they manage are set, and purchase prices are set based upon random normal draws from indicated distributions. Analogous settings are made for wheat elevators (*setup-wheat\_elevators*), mills (*setup-wheat\_mills*), and bakers (*setup-wheat\_bakers*). Peach agents (s*etup-peach-agents*) establish similar values for processors (*setup-peach\_processors*) and agents representing alternative markets (*setup-peach\_altmarks*).

Estimated yields (*setup-estimated-yields*) are calculated for the different commodities produced by potato, wheat, and peach farmers. These estimates are used in establishing contracts based on expected production; actual production affecting inventories and assets are modeled during simulation rather than initialization. The numbers of patches producing the given commodity is tallied, then calculations based on yields per unit area are made to calculate total production for the given commodity and farmer. Lastly, *setup-tables* is called to establish tables that are used in contracting.

The setup procedure ends by calculating max-ticks, which are weekly in the model, such that the *years-in-model-run* is multiplied by 52.

**6. Input data**

Data informing the FFAR-ABModel fall into three general categories: 1) spatial surfaces that are used to initialize the Colorado food system for the focal commodities, 2) spatial surfaces providing yield estimates generated from ecosystem models for a range of years, and 3) spatial surfaces providing annual precipitation values from 2000 to 2010. The surfaces are stored as GRIDASCII files, which include a header of six lines and then a rectangular matrix of values equal in size to the dimensions of Colorado.

Spatial surfaces include those that describe the 1) counties of Colorado, patches relevant to potato production including 2) patches that may produce potatoes, 3) mean yields for russet canela potatoes, 4) russet canela potatoes grown using organic methods, 5) purple potatoes, and 6) purple potatoes grown organically. Three spatial surfaces each year describe production of wheat, including 7) conventionally produced hard red winter wheat, 8) organic production of hard red winter wheat, and 9) conventional production of white Snowmass wheat. Five surfaces are used to aid in modeling beef production: 10) public versus private grazing lands, 11) forage production estimates for those lands, 12) the long-term average of precipitation, 13) areas that are irrigated, and 14) areas legally grazable in the areas where beef is produced.

Surfaces for potato yields include yield estimates from DSSAT (Jones et al. 2003; Hoogenboom 2019, 2021) for russet canela, russet canela organic, purple, and purple organic from 1980 to 2018. Analogous surfaces from DSSAT are used for red wheat-fallow rotation conventional production, red wheat-fallow rotation organic, and white wheat fallow rotation conventional production from 1981 to 2018. Precipitation surfaces are used in beef forage production estimates. These surfaces form the foundation for yield estimates in FFAR-ABModel. Yields within the ranges of data available are generated randomly, and then the surface corresponding to that year read in and used to determine yields for the year.

**7. Submodels**

* **FFAR\_ABMmodel\_umbrella.nlogo**
  + *setup*

Setup is typical of many NetLogo models, being comprised of a series of calls to other procedures for the most part. The model is reset (*clear-all*) and ticks reset (*reset-ticks*). The model then calls *setup-world*, *setup-static-names*, *setup-parameters*, and *setup-dashboard*. Lastly, the total number of ticks (i.e., time-steps) in the model is calculated. In general, this procedure orchestrates initialization of the model, addressed in Section 5.

* + *setup-parameters*

This procedure is lengthy but simple, in so far as it sets a long series of parameters to appropriate values. These values may be adjusted to update inputs or in scenarios. The procedure ends with calls to patches used in peach or wheat production and setting initial yields.

* + *go*

As in most NetLogo models, the *go* procedure in FFAR-ABModel controls the passage of time, being run repeatedly until the model completes. As such, it is the controller of the processes and scheduling reviewed in Section 3.

* + *manage-dates*

This procedure is brief and increments weeks in the model. If the week exceeds 52, the year is incremented, and weeks reset to 1.

* + *updateTls*

The *updateTLs* procedure is brief and calls three routines, *first-time-farmer-update*, *neighborUpdate*, and *bad-harvest-update*. The procedure then trims the transition-likelihood-score to ensure it is between 0 and 1.

* + *first-time-farmer-update*

The *first-time-farmer-update* procedure alters the transition score for first time farmers after year 5 in the model. That is, all farmers loose that designation after the fifth year. The index used to designate an alternative farmer is updated at the end of the procedure.

* + *neighborUpdate*

The *neighborUpdate* procedure includes a rare use of spatial relationships in FFAR-ABModel. Small farms or those that are medium and managed in an alternative way calculate their nearest three neighbors that share their attributes. The sum of the area in transition to organic for those neighboring farms can then influence the farm’s transition-likelihood-score. A similar procedure is done for large farms or medium farms traditionally managed, although any three nearest neighboring farms is used.

* + *bad-harvest-update*

This brief procedure adjusts transition likelihood scores if the year has been a mediocre or bad year, setting the indices higher.

* **setupWorld.nls**
  + *setup-world*

This is the entry into the umbrella program, setting up the world in which the simulation occurs. Three procedures are called: *create-map*, *setup-start-dates*, and *setup-gis-landscape*. No other statements are included.

* + *create-map*

The spatial arena used in the model is defined here. The size of the spatial arena is defined and the dimension of an individual cell (i.e., patch) in the interface is defined. The dimensions of the spatial arena are relevant to the simulation, but the size of individual patches affects the display of the arena only.

* + *setup-start-dates*

The *setup-start-dates* routine sets the year variable to 1 and the week variable to 30. This the model is initialized to start at year 1 and a week in late July (i.e., starting about July 20th).

* + *setup-gis-landscape*

This procedure sets up the landscape in the spatial arena. A series of spatial surfaces in GRIDASCII format are brought into NetLogo using the two-step process adopted in the GIS extension of that software package. Surfaces are brought into temporary variables, and those are then applied to patch-level variables. Surfaces brought into the simulation include: counties, public versus private lands, forage production estimates, precipitation, areas that are irrigated, areas that are grazable, and mean yields for a variety of potato and wheat types.

After surfaces are brought into patch variables, counties are painted and assigned to different commodity types. For counties that produce beef, mean forage production is converted from pounds per acre to kilograms per hectare.

* **setupStaticNames.nls**
  + *setup-static-names*

In this procedure, sets of variables for the different main commodities (i.e., \**\_price, \*\_quantity, \*\_sent, and \*\_outstanding\_quantity*) are assigned static names. Prior to that, the variables are defined in a GLOBALS statement.

* **setupDashboard.nls**
  + *setup-dashboard*

The *setup-dashboard* procedure is a calling routine that runs a series of setup procedures to define agents and place them on the landscape. The display is setup, estimated yields for commodities are also set, and tables used in contracting are defined.

* + *setup-potato\_farmers*

This procedure creates 26 potato farms in Alamosa County, 41 farmers in Rio Grande County, and 29 in Saguache County. A given county has a set of statements, which are repeated for each county.

After a farmer is created, its county and a unique identifier are assigned. Following the creation of the set of farmers, based on governmental statistics (e.g., US Agricultural Census), a subset of agents with similar-sized farms are randomly selected and assigned a farm size drawn from a random normal distribution. A category of farm size is assigned as well (i.e., “small”, “medium”, “large”). This process is repeated until all potato farmers in the given county have been assigned areas for potato farming, in hectares.

Following assignment of farm areas, other attributes of potato farmers are assigned. The assets are initialized to $10,000 US, maximum inventories of different potato types are assigned, and initial sale prices are set based on a draw from a random normal distribution set by average prices and their standard deviation. Total farm area is assigned based on the area in potatoes farmed times a multiplier.

Farmers are moved to an available patch within their county. For each farmer, available neighboring patches are selected and assigned to that farmer until the area assigned to the farmer equals that of the farm (i.e., patches-occupied).

Farmers in each county are assigned socio-demographic attributes (i.e., gender, first-time farmer, age and race categories, and whether or not farming is the primary income). Following that, those attributes are used with multipliers to assign a transition likelihood score that helps determine whether farmers transition from conventional to organic methods.

The procedure ends by assigning a portion of a farmer’s lands to barley, initially. This allows the most common rotational patterned adopted in the San Luis Valley to be represented.

* + *setup-wheat\_farmers*

The procedure creates 234 wheat farmers in Washington County. The farmers are initialized and identifiers assigned, then the sizes of farms are assigned based on random normal values drawn for three groups, smaller farms, mid-sized farms, and large farms.

Attributes of all farmers are then set, such as initial assets and maximum inventory and prices to sell conventional and differentiated products. Total acres farmed are then adjusted to account for the typical potato-barley rotation adopted in the region. Farmers are moved to a randomly selected suitable patch, and then a routine is employed where neighboring patches that are unassigned are assigned to the farmer in question, until the size of the farm is as initialized.

Subsets of farmers are assigned personal attributes (e.g., race, age class, first-time farmer, whether or not the farm is for primary income). Those attributes are used to assign levels of risk aversion, in an accumulative way, with attributes adjusting the aversion in sequence.

The procedure ends by assigning rotations to farmers. A portion of the farmers are assigned to be growing wheat when the model is initialized, and the remaining farmers are growing barley.

* + *setup-peach\_farmers*

This procedure initializes peach farmers in Delta and Mesa Counties. In Delta County, 26 farmers are set and initialized with identifiers. Those 26 farmers are then assigned orchard acreages based on five categories (small, three mid-range levels, and large). Between 3 and 11 peach farmers are in each category. The acreages of orchards are assigned based on random normal values drawn based on values unique to each category. Analogous processes are carried out for 41 peach farmers in Mesa County.

The peach farmer with the largest orchard is assigned as an outsource packer. Other farmers may be assigned as medium farmers or half mediums, or split packers, who divide their production, selling product to those that outsource and sell peaches from the farm as well.

Peach farmers are assigned attributes, such as initial assets and their initial loyalty to a given buyer. Attributes for the outsource packer are assigned uniquely given that farmgate prices are not needed, but in general, maximum inventories and prices to sell conventional and differentiated products are assigned. Like for other commodities, a routine is then implemented that moves a farmer to an unassigned and otherwise appropriate location, then grows the orchard based on the assignment of neighboring cells to the farmer in question, until the designated orchard size is reached.

Demographics of peach farmers are assigned based on observed values for the two counties. Those demographics are then used to assign likelihood of transition scores, reflecting how likely people in different classes (e.g., age, ethnicity, first-time farmer or well established) will transition from conventional to organic production.

* + *setup-beef\_ranchers*

The procedure initializes beef ranchers raising cow-calf pairs in the northwestern part of the state. Large-scale (i.e., “diversified”) ranchers are assigned to those in Moffat and Routt Counties, followed by small-scale ranchers for both counties. Identifiers and the number of cows that calve are assigned to the ranches.

For large and small ranches in turn, the amount of irrigated lands is assigned, then the amount of forage produced from grazing is assigned versus food required to be purchased from leased land or hay.

The routine ends by moving ranchers to available and randomly selected patches in their appropriate counties. An identifier is assigned and the forage needs by cattle is assigned.

* + *locate-beef\_ranchers*

The procedure locates beef ranchers and assigns patches to owners. Ranchers are moved to an unoccupied patch with forage production and without irrigation. The same is done for non-irrigated land, with the patch selected shown as occupied and the farmer identifier assigned. The same logic is then used to position small ranches, with those having irrigated lands and those without irrigated lands handled separately. The routine ends by assigning to all ranchers the area grazed, the stocking rate, and average precipitation.

* + *setup-certified-organic*

This brief procedure establishes some proportion of patches for each crop as organic. The patches themselves are assigned as certified organic, and then the owners that have certified organic patches have an indicator set as such.

* + *setup-potato-farm-crop-allocation*

This brief procedure sets the initial allocation for each of the potato farmers as 80% canela potatoes and 20% purple potatoes.

* + *setup-wheat-farm-crop-allocation*

This brief procedure sets the initial allocation for each wheat farmer as 95% hard red winter wheat and 5% Snowmass wheat.

* + *setup-display*

The *setup-display* procedure paints patches in the spatial arena different colors based upon their locations and attributes. A “chooser” widget allows the user to select from different display options. If the option “potato vs barley rotation” is selected, those attributes are painted different colors for patches with the variable rotation set to one of those options. The same is true for “wheat vs barley.” The same types of information are painted for organic versus conventional croplands, and farm size and ideology for the different crops.

* + *setup-potato-agents*

This is a simple calling procedure, calling in sequence *setup-potato\_repackers*, *setup-potato\_shippers*, and *setup-potato\_processors*.

* + *setup-potato\_repackers*

The procedure creates the number of potato repackers as set by the user. Their identifiers, positions, and other attributes are set. Their initial assets are set, and then the maximum inventory they can store for the various commodities is set, currently to a very large value. Prices to buy conventional russet potatoes, organic potatoes, purple potatoes, purple organic potatoes, and small russets are set based upon random draws from a normal distribution with commodity-specific parameters.

* + *setup-potato\_shippers*

The procedure creates the number of potato shippers as set by the user. Their identifiers, positions, and other attributes are set. Their initial assets are set, and then the maximum inventory they can manage for the various commodities are assigned. Prices to buy conventional russet potatoes, organic potatoes, purple potatoes, purple organic potatoes, and small russets are set based upon random draws from a normal distribution with commodity-specific parameters.

* + *setup-potato\_processors*

The procedure creates the number of potato processors as set by the user. Their identifier, position, and other attributes are set. Their initial assets are set, and then the maximum inventory they can store, currently set to a very large value. Prices to buy conventional russet potatoes, organic potatoes, purple potatoes, purple organic potatoes, and small russets are set based upon random draws from a normal distribution with commodity-specific parameters.

* + *setup-wheat-agents*

This brief procedure consists of three calls to other procedures, *setup-wheat\_elevators*, *setup-wheat\_mills*, and *setup-wheat\_bakers*.

* + *setup-wheat\_elevators*

The procedure creates the number of wheat elevators requested by the user. Their attributes are set, such as a random location, an identifier, and initial assets. The maximum inventory for hard red winter wheat, Snowmass wheat, and organic hard red winter wheat are set. The purchase prices for each of the commodities is set based on a random draw from a normal distribution. If these values are too low, they are reset to the value that allows the farmer to break even.

* + *setup-wheat\_mills*

The procedure creates the number of wheat mills requested by the user. Their attributes are set, such as a random location, an identifier, and initial assets. The maximum inventory for hard red winter wheat, Snowmass wheat, and organic hard red winter wheat are set to very large values. The purchase prices for each of the commodities is set based on a random draw from a normal distribution. If these values are too low, they are reset to the value of the farmgate average.

* + *setup-wheat\_bakers*

The procedure creates the number of wheat bakers requested by the user. Their attributes are set, such as a random location, an identifier, and initial assets. The maximum inventory for hard red winter wheat, Snowmass wheat, and organic hard red winter wheat are set to very large values. The purchase prices for each of the commodities is set based on a random draw from a normal distribution, with parameters unique to each commodity. If these values are too low, they are reset to the average price an elevator receives.

* + *setup-peach-agents*

This procedure is a single line, calling *setup-peach\_processors*.

* + *setup-peach\_processors*

The procedure creates the number of peach processors requested by the user. Their attributes are set, such as a random location, an identifier, and initial assets. The maximum inventory for conventional and organic peaches are set. The purchase prices for those commodities is set based upon a random draw from a normal distribution using parameters unique to the commodities.

* + *setup-beef-agents*

This procedure contains a single line, calling *setup-beef\_packing-houses*.

* *setup-beef\_packing-houses*

The procedure creates the number of beef packing houses requested by the user. Their attributes are set, including their random location, assets, which are initialized to zero, and inventories. Beef commodities (conventional, grass-fed, Colorado-sourced, and produced with animal welfare requirements) have their prices to buy set. Lastly, one of the beef packing houses is assigned a small size.

* + *setup-distributors*

The procedure creates the number of distributors requested by the user. The distributors are given a random location and their attributes are set, including their initial assets. Distributor agents manage all the commodities represented in the merged model. With that, maximum inventories are set for each of the eight potato conventional and differentiated products. Maximum inventories for the six wheat commodities are then set. The three peach commodities then have their maximum inventories set to very large values. The inventories for the four beef commodities are set as well.

Conventional and organic potato products are assigned prices to buy based on random normal draws based on averages received from shippers. Random draws that are below average farmgate prices are reset to those values. An option then follows that is specific to specialty products, potentially resetting prices to custom values if the scenario is engaged.

Prices are then set for wheat, beef, and peach commodities akin to the potato commodities, using random normal draws based on commodity-specific parameters. Those prices are then compared to corresponding average prices and set to those prices if too low.

* + *setup-dps\_buyers*

The *setup-dps\_buyers* procedure creates and initializes the Denver Public School buyers (although a single agent is created). Attributes of the buyer is set, and then maximum inventories for different commodities is set, depending upon the scenario being simulated. Prices to buy commodities are then set, often using random normal draws for set distributions. The procedure ends with the buyer being placed spatially.

* + *setup-estimated-yields*

This lengthy procedure sets the estimated yields for each of the crop commodities. For russet conventional potatoes, the set of farmers is divided in a series of statements until the group of interest is obtained. The estimated yield for russet conventional potatoes is then assigned based on the acreage in potatoes for each farm. Two-thirds of the potatoes produced are assigned as typical in size, and one-third are assigned as small russet conventional potatoes. Yields for purple potatoes are set following this. For organic production, a new set of farmers is created, restricted to those doing organic production. Then the estimated yield for russet organic and purple organic potatoes is set.

Wheat farmers are subsetted into three groups, hard red winter wheat, Snowmass, and hard red winter wheat organic. Formula are used to assign yield estimates to those entries.

Peaches are the final commodity with estimated yields assigned. Sets are created based on the attributes of peach farmers, then conventional estimated yield is assigned. A new set is created, and then organic estimated yield is assigned.

* + *setup-tables*

Tables are used to manage contracting for commodities. Those tables are initialized for the three crop commodities, for conventional and differentiated products.

* **modelOutputs.nls**
  + *calculate-lca*

This procedure is very lengthy but simple, in that it sets a long series of parameters associated with life cycle analyses. The amount of commodity produced, in the appropriate units, was multiplied by the appropriate unit-normalized life cycle analysis variable to yield a total for the given emission or water use.

* + *calculate-profitability*

The *calculate-profitability* procedure is logically straightforward, calling the appropriate agent-types to calculate profitability based upon the commodity being simulated. Regardless of commodity, distributors and Denver Public School buyers are called, in that they are in every simulation. This procedure uses a variable agentset passed to the *profitabilityCalculation* procedure, allowing that procedure to run commands for the given agentset.

* + *profitabilityCalculation*

This brief procedure accepts whatever agentset is passed to it, and then stores the final assets and calculates a change in assets from the start to the end of the simulation and average yearly change in assets.

* **potatoDashboard.nls**
  + *reset-storage-potatoes*

The *reset-storage-potatoes* procedure is comprised of calls to *reset-storage\_potatoes* (note the underscore, distinct from this procedure) for each of the six agent types in the potato model.

* + *reset-promised-amounts-potatoes*

For each agent type in the potato simulation, commodity amounts promised and incoming quantities are reset to 0.

* + *reset-tables-potatoes*

In *reset-tables-potatoes*, calls are made to an intrinsic routine *table:clear* for each of the nine commodities (both raw and processed). This resets contractual relationships.

* + *profitability-check-potatoes*

This procedure is used to forecast profitability for the potato farmers. If a farm includes patches that are in transition to organic production, that transition and the transition likelihood scores are updated. If a farm is not in transition, the potential profit is calculated based on an estimated yield and typical price at the farmgate. The outcomes of these calculations (*predicted-org-profit* and *predicted-conv-profit*) are compared, and if organic production is thought to likely yield higher profits, transitional likelihood scores are increased (or decreased if the reverse is true) and some patches begin transition to organic production (or revert). Large farms do not revert; if organic production is likely to be more profitable some land is transitioned, and the procedure stops.

* + *transition-patches-potatoes*

This procedure transitions patches used in potato production based on the transition likelihood score of a farmer. If that score is > 0.5, the percentage of patches that are in organic production is calculated. If that value is > 25%, all remaining patches transition to organic production. Otherwise, a randomly selected patch is converted to be in transition. The procedure concludes by allowing farmers with transition likelihood scores < 0.5 and with various relative sizes and management combinations to transition their organic patches to conventional production. Here all patches in organic production convert in a single step.

* + *update-organic-potatoes*

This simple procedure sets patches that are growing potatoes and in organic transition to indicate that status. Note that this is patch-based, rather than farmer-based, helping patches to track their organic production status.

* + *into-organic-potatoes*

The *into-organic-potatoes* procedure transitions patches that are in transition to organic production. It increments patches marked as in transition, and if in the fourth year, they convert to organic production.

* + *update-display-potatoe*s

The *update-display-potatoes* procedure updates the spatial arena display of Colorado and the status of potatoes in the counties that produce them. A chooser named *display-style* controls the nature of what is drawn to the screen. That chooser may be set to entries such as “farm ownership” or “organic vs conventional”. Commands under different if statements then pain the appropriate elements.

* + *set-contracts-potatoes*

This procedure has a lengthy set of calls that set contracts for the different potato commodities and the respective sellers and buyers (e.g., farmers selling to shippers, farmers to repackers), including some specific to scenarios. The calls are to individual combination-specific procedures to manage contracts. A single example among dozens is: *setupContracts\_russetConvensionalFresh potato\_shippers distributors number-distributors TRUE.* That includes passing the seller, the buyer, the number of buyers involved in contract negotiations, and whether or not loyalty plays a role in contract formation.

* + *produce-potatoes*

This procedure is a series of five calls to procedures that produce potatoes. The five procedures called are the next five in this list.

* + *set-potato-yield*

The *set-potato-yield* connects FFAR-ABModel to the ecosystem model that predicted potato yields for a series of years. The procedure uses an assigned year within the range of the spatial surfaces available, and then reads in those data and fills variables for patches with the provided yield estimates.

* + *update-bad-production-year-potatoes*

The *update-bad-production-year-potatoes* procedure determines if the year’s potato production is below average. First, the procedure saves the current values for mediocre production year and bad production year into variables labeled as the previous year. Then the year’s yield is compared to long-term yields. If production is less than 70% of long-term yield the current year is labeled a bad yield year. If production is between 70% and 100% it is labeled a mediocre year.

* + *set-potato-farm-crop-allocation*

This procedure is a simple routine that sets initial production for each farmer to 80% russet canela and 20% purple potatoes.

* + *set-potato-barley-rotation*

The *set-potato-barley-rotation* procedure manages advancement of rotational schedules for potato production. Note: In this version, all producers follow the same schedule, with synchronized rotations; this may be changed if helpful. *Potato-rotation-year* controls the year of rotation, and patches are advanced by assigning the variable *rotation* to a string, either “barley”, “transition”, or “potato”.

* + *calculate-annual-production-potatoes*

Whereas *set-potato-yield* sets yields for patches based on spatial surfaces, *calculate-annual-production-potatoes* calculates yields for each specific farmer based on their individual ownership. Areas in each of four commodities are calculated for use in input cost calculations. Assets are decremented to account for the costs of cultivating the areas in question. Then production is calculated for the four commodities (i.e., russet canela and purple potatoes in conventional and organic modes of production). Lastly, inventories are updated and passthrough variables set.

* + *fulfill-contracts-potatoes*

The *fulfill-contracts-potatoes* procedure is a series of calls to procedures that are unique to commodities, but have similar names (e.g., *fulfillContracts\_russetConventionalFresh, fulfillContracts\_purpleOrganicFresh, fulfillContracts\_smallFresh*). Different procedures of a similar nature are called depending upon the scenario being simulated.

* + *process-potatoes*

In this procedure, potato processors work with their inventory. First, the current inventory is set. If there is inventory, space for processed materials is assessed, and if greater than the current inventory held, the potatoes are processed and inventories updated. If less space is available, a portion of the potatoes are processed.

* + *fulfill-processor-contracts-potatoes*

In *fulfill-processor-contracts-potatoes*, only the “scenario specialty” scenario causes procedures to be called. Those procedures are for fulfilling contracts, much like in *fulfill-contracts-potatoes*.

* + *dps-purchases-baseline-french-fries*

In *dps-purchases-baseline-french-fries*, the Denver Public School buyer purchases French fries. Distributors set a price to sell out-of-state French fries based on a random normal draw and an assigned distribution. The Denver Public School buyer selects the distributor with the lowest price and then purchases sufficient fries to fill the inventory. Assets and inventories are then updated.

* **potatoContractNegotation.nls**
  + *setContracts\_russetConventionalFresh*

This lengthy procedure sets contracts for conventional fresh potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_russetConventionalFresh*

This procedure adjusts the prices buyers offer for conventional fresh potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_russetConventionalFresh*

This function simply calculates available space for conventional fresh potatoes based on the current inventory and maximum inventory for buyers.

* + *setContracts\_russetConventionalProcessed*

This lengthy procedure sets contracts for conventional processed potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_russetConventionalProcessed*

This procedure adjusts the prices buyers offer for conventional processed potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_russetConventionalProcessed*

This function simply calculates available space for conventional processed potatoes based on the current inventory and maximum inventory for buyers.

* + *setContracts\_russetOrganicFresh*

This lengthy procedure sets contracts for organic fresh potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_russetOrganicFresh*

This procedure adjusts the prices buyers offer for organic fresh potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_russetOrganicFresh*

This function simply calculates available space for organic fresh potatoes based on the current inventory and maximum inventory for buyers.

* + *setContracts\_purpleConventionalFresh*

This lengthy procedure sets contracts for purple conventional fresh potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_purpleConventionalFresh*

This procedure adjusts the prices buyers offer for purple fresh potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_purpleConventionalFresh*

This function simply calculates available space for purple fresh potatoes based on the current inventory and maximum inventory for buyers.

* + *setContracts\_purpleConventionalProcessed*

This lengthy procedure sets contracts for purple conventional processed potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_purpleConventionalProcessed*

This procedure adjusts the prices buyers offer for purple processed potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_purpleConventionalProcessed*

This function simply calculates available space for purple processed potatoes based on the current inventory and maximum inventory for buyers.

* + *setContracts\_purpleOrganicFresh*

This lengthy procedure sets contracts for purple organic fresh potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_purpleOrganicFresh*

This procedure adjusts the prices buyers offer for purple organic potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_purpleOrganicFresh*

This function simply calculates available space for purple organic fresh potatoes based on the current inventory and maximum inventory for buyers.

* + *setContracts\_purpleOrganicProcessed*

This lengthy procedure sets contracts for purple organic processed potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_purpleOrganicProcessed*

This procedure adjusts the prices buyers offer for purple organic potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_purpleOrganicProcessed*

This function simply calculates available space for purple organic processed potatoes based on the current inventory and maximum inventory for buyers.

* + *setContracts\_smallFresh*

This lengthy procedure sets contracts for conventional fresh small potatoes for seller and buyer agentsets passed to it. It begins with calls to *priceAdjustments* and *contractSpaceAvailable* for the commodity. Then negotiations begin for potato farmers as sellers. Potatoes available for sale are calculated based on the nature of buyers; 35% of stock for shippers and 100% of stock for repackers. If instead sellers are potato shippers, analogous calculations with the same percentages are made for them.

Prices are then negotiated. Potatoes that are promised by the seller are compared to the quantities available and whether or not any buyers have that quantity of potatoes available. Buyers with the capacity are identified. From those and the market knowledge set, buyers bidding on potatoes are identified. The buyer offering the highest price to purchase is then identified, and that price is determined. (If the buyer is a Denver Public School purchaser and a scenario is being represented (i.e., “GFPP” or “small potatoes”), the price is drawn from a known random normal distribution.) Prices are adjusted 10% higher in the off season. That price is then stored in the appropriate table using a call to *table\_set*.

Quantities to be sold are then negotiated, selling to various buyers and incrementing a promised tally until the inventory to be sold is allocated to buyers. The space available for the buyer is determined and the quantities of potatoes available set, and the minimum of those determines the quantity to be purchased by that buyer. The quantity to be sold this time step and to this buyer is then stored in a table using a call to *table\_create\_and\_sum*, and the amount that remains outstanding and not yet sold are stored in another table with a call to the same procedure. The quantity of potatoes promised is then incremented by the contracted amount, increasing the promised amount, and the space available for buyers is updated.

This procedure uses frequent print statements to track elements. Those are not described above and do not affect program logic, but are helpful in program validation.

* + *priceAdjustments\_smallFresh*

This procedure adjusts the prices buyers offer for small fresh potatoes. A number of buyers are selected randomly, with the number used reflecting a degree of market knowledge. Then the buyer with the lowest price offered increases that price by 10%, and the buyer with the highest price decreases their price by 10%. This serves to provide some tendency toward equilibrium in pricing.

* + *contractSpaceAvailable\_smallFresh*

This function simply calculates available space for small fresh potatoes based on the current inventory and maximum inventory for buyers.

* **potatoContractFulfillment.nls**
  + *fulfillContracts\_russetConventionalFresh*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, russet conventional fresh potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* + *fulfillContracts\_russetConventionalProcessed*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, russet conventional processed potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* + *fulfillContracts\_russetOrganicFresh*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, russet organic fresh potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* + *fulfillContracts\_purpleConventionalFresh*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, purple conventional fresh potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* + *fulfillContracts\_purpleConventionalProcessed*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, russet purple conventional processed potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* + *fulfillContracts\_purpleOrganicFresh*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, purple organic fresh potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* + *fulfillContracts\_purpleOrganicProcessed*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, purple organic processed potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* + *fulfillContracts\_smallFresh*

The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. Here, conventional small potatoes are exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of potato exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

The procedure ends with a clause that applies only to farmers and with the GFPP scenario active. If that is the case, a premium is given to one of the farmers reflecting price gains associated with the Good Food Purchasing Program.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* **potatoResetStorage.nls**
  + *reset-storage\_potatoes*

As expected, the *reset-storage\_potatoes* procedure resets the storage for different types of potatoes. That said, values aren’t always reset to zero or some other predetermined value, but rather sums are tracked. This procedure may be called for different agentsets, depending upon the agentset passed to the procedure. The items reset are controlled by if statements that identify the given agentset type. Assets are summed based upon the current holdings plus the value of the recent sales of the different commodities. Potatoes that remain in inventory are sold to interests outside the Denver Public School system (i.e., *potatoes-outside*). The current inventory for the given commodity is then reset to zero. These processes are carried out for every commodity, although many agents will have no data for some commodities. If the agentset is farmers, then the passthrough values that track annual or sub-annual inventories are reset to zero, and if the agentset is Denver Public School System buyers, inventories are simply set to zero after selling remaining inventory to outside interests. Final assets are then set to assets.

* **wheatDashboard.nls**

**Note**: For brevity, components of the wheat, beef, and peach commodity supply chains that are structured like those in the potato supply chain will note that similarity rather than repeating the comments given earlier in the ODD. Also, procedures that apply the same steps to different differentiated commodities are grouped.

* + *reset-storage-wheat*

Analogous to *reset-storage-potatoes*, but for wheat\_farmers, wheat\_elevators, wheat\_mills, wheat\_bakers, and wheat commodities held by distributors and the dps\_buyer. This procedure is a series of calls to *reset-storage\_wheat* cited below (note underscore).

* + *reset-promised-amounts-wheat*

For each agent type in the wheat simulation, commodities promised and incoming commodities are reset to 0.

* + *reset-tables-wheat*

Analogous to *reset-tables-potatoes*, the *reset-tables-wheat* procedure clears tables used in contracting for the three wheat commodities (conventional hard red winter wheat, organic hard red winter wheat, and conventional Snowmass wheat). Three more calls are made to *table:clear* for flours, and three more for breads.

* + *profitability-check-wheat*

This procedure is analogous to that described for potatoes. Profitability of conventional and organic production pathways is compared.

* + *transition-patches-wheat*

The *transition-patches-wheat* procedure is structured differently than from the same procedure for potatoes. Four procedures are called in transition-patches-wheat, including *updateTls* with wheat\_farmers, *profitability-check-wheat*, *transition-check-wheat*, and *update-organic-wheat*. Proportions of patches in organic production and in transition are calculated and those are used in plotting results. Proportions are again calculated, but for individual wheat farmers. These commands repeat what is near the bottom of the analogous potato procedure.

* + *transition-check-wheat*

This procedure transitions patches to onto an organic transitioning pathway depending upon the farmer’s transition likelihood score. Wheat farmers with given attributes (e.g., small or medium, or large actively in transition are assessed as to their organic and transitional status, and patches are moved to transitional status if appropriate. This code is akin to code within *transition-patches-potatoes*, if the description there is helpful.

* + *update-organic-wheat*

This procedure is analogous to the code from the *update-organic-potatoes* procedure.

* + *update-display-wheat*

This procedure draws content relevant to wheat to the spatial arena in NetLogo, analogous to that done for the matching potato procedure.

* + *set-contracts-wheat*

The *set-contracts-wheat* procedure is analogous to the matching potato procedure. Commodity-specific calls are made to procedures to set contracts between wheat product buyers and sellers. For example, farmers sell to elevators the different commodities, elevators sell to mills, and mills sell to bakers. Scenarios alter the contracts put in place, specifically the products sold by distributors to the DPS buyer.

* + *produce-wheat*

The *produce-wheat* procedure is brief, calling three other procedures, *set-wheat-yield*, *calculate-annual-production-wheat*, and *update-bad-production-year-wheat*.

* + *set-wheat-yield*

Like for potatoes, the *set-wheat-yield* procedure connects the FFAR-ABModel to the products from the ecosystem modeling. Years are connected to annual estimates for wheat production, and the corresponding yield estimate surfaces are read in. Yields are set and conversions occur to move from bushels per acre to pounds per hectare. Means are calculated for all patches, for use elsewhere in the code.

* + *update-bad-production-year-wheat*

This procedure is analogous to the corresponding procedure for potatoes. The coefficient used to determine if a year is bad or mediocre is 60% of average production rather than 70%, as for potatoes.

* + *update-wheat-rotation*

This procedure manages wheat rotation, assigning appropriate cropping to farmers as they move through either wheat and fallow rotation, or wheat, corn, and fallow rotation.

* + *update-wheat-production*

The *update-wheat-production* procedure updates plots that summarize wheat production. This includes summing annual production and production across the entire model run.

* + *calculate-annual-production-wheat*

This procedure is analogous to the corresponding potato procedure, calculating yields for individual farms and updating inventories.

* + *fulfill-stage1-contracts-wheat*

This procedure is a series of calls to other procedures with similar names and specific to commodities (e.g., *fulfillContracts\_hrwConventionalWheat*, *fulfillContracts\_hrwOrganicWheat*). Calls pass corresponding sellers and buyers to fulfill contracts, including farmers, elevators, and mills.

* + *fulfill-stage2-contracts-wheat*

The *fulfill-stage2-contract-wheat* procedure is structurally the same as the stage 1 procedure, but different buyers and sellers are passed to procedures that sell flour. The sellers are mills and buyers are bakers and distributors.

* + *fulfill-stage3-contracts-wheat*

Bread and flour are sold in the stage 3 of contract fulfillment for wheat. Bakers sell bread to distributors, and then depending upon the scenario being simulated, flour and bread of a given commodity are sold by distributors to the Denver Public School System buyer.

* + *make-flour*

In this procedure, for each of the commodities simulated (conventional hard red winter wheat, organic hard red winter wheat, and conventional Snowmass), wheat is converted to flour and changes in inventories are tracked.

* + *make-bread*

Similar to the *make-flour* procedure, this procedure steps through each of the bakers and if they have flour in their inventory for a given commodity and sufficient space in their inventory for bread, the flour is converted to bread and inventories are updated.

* **wheatContractNegotation.nls**
  + *setContracts\_hrwConventionalWheat*
  + *setContracts\_hrwConventionalFlour*
  + *setContracts\_hrwConventionalBread*
  + *setContracts\_hrwOrganicWheat*
  + *setContracts\_hrwOrganicFlour*
  + *setContracts\_hrwOrganicBread*
  + *setContracts\_snowmassConventionalWheat*
  + *setContracts\_snowmassConventionalFlour*
  + *setContracts\_snowmassConventionalBread*

These lengthy procedures are analogous to the *setContract\_\** procedures used for potatoes. Space available for whatever buyers are passed to the procedure are calculated. Estimates of quantities available for sale are updated for select sellers. Buyers within the market knowledge of the seller are queried as to their capacity to purchase product. Then bids are gathered by the seller from the buyers, and product sold to the bidder with the highest offer. Anticipated inventories are updated and contracts are created until the seller has allocated their anticipated production to buyers.

* + *priceAdjustments\_hrwConventionalWheat*
  + *priceAdjustments\_hrwConventionalFlour*
  + *priceAdjustments\_hrwConventionalBread*
  + *priceAdjustments\_hrwOrganicWheat*
  + *priceAdjustments\_hrwOrganicFlour*
  + *priceAdjustments\_hrwOrganicBread*
  + *priceAdjustments\_snowmassConventionalWheat*
  + *priceAdjustments\_snowmassConventionalFlour*
  + *priceAdjustments\_snowmassConventionalBread*

These brief procedures are analogous to the *priceAdjustments\_*\* procedures used for potatoes. Buyers of a given commodity are passed to the appropriate procedure. Buyers then examine prices of other buyers within their market knowledge and the buyer with the lowest purchase price increases their price offering by 10%, and buyers with the highest offering decrease their price by 10%.

* **wheatContractFulfillment.nls**
  + *fulfillContracts\_hrwConventionalWheat*
  + *fulfillContracts\_hrwConventionalFlour*
  + *fulfillContracts\_hrwConventionalBread*
  + *fulfillContracts\_hrwOrganicWheat*
  + *fulfillContracts\_hrwOrganicFlour*
  + *fulfillContracts\_hrwOrganicBread*
  + *fulfillContracts\_snowmassConventionalWheat*
  + *fulfillContracts\_snowmassConventionalFlour*
  + *fulfillContracts\_snowmassConventionalBread*

These procedures are analogous to *fulfillContracts\_*\* procedures used for potatoes. The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. A given wheat commodity is exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of wheat exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* **wheatResetStorage.nls**
  + *reset-storage\_wheat*

The *reset-storage\_wheat* procedure is analogous to the *reset-storage\_potatoes* procedure. This procedure receives an agent set that may be any of the wheat food system participants. Then depending upon the agent set, the procedure essentially sells any remaining inventory of the appropriate commodity to an outside buyer and then resets the current inventory to zero. DPS buyers do not sell excess inventory to outside buyers.

* **peachDashboard.nls**
  + *reset-promised-amounts-peaches*

Analogous to *reset-storage-potatoes*, but for peach\_farmers, peach\_processors, and peach commodities held by distributors and the dps\_buyer. Promised peach quantities and incoming totals are set to zero.

* + *reset-tables-peaches*

Analogous to *reset-tables-potatoes*, the *reset-tables-peaches* procedure clears tables used in contracting conventional and organic peaches.

* + *profitability-check-peaches*

This procedure is analogous to the *profitability-check-potatoes* procedure. Profitability of conventional and organic production pathways for peaches are compared and TLS indices may be updated.

* + *transition-patches-peaches*

The *transition-patches-peaches* procedure is structured differently than from the same procedure for potatoes. The procedure assigns patches to begin their transition to organic production based on transition likelihood scores. If a transition to organic is to occur and more than one-quarter of the farm is organic, the remaining farmland begins to transition. Otherwise, a patch begins transitioning. If a farmer’s TLS is low and based on farm size and farmer attributes, the farm may transition out of organic production and back to conventional.

* + *update-organic-peaches*

The *update-organic-peaches* is analogous to the procedure used for potatoes. If a patch is in peach production and in transition, it calls *into-organic-peaches*.

* + *into-organic-peaches*

The *into-organic-peaches* procedure increments the time in transition for peach orchards each year, and if three years have passed, assigns them as certified organic. A farmer’s ownership is then queried, and if all lands are organic the farm is labeled as fully organic.

* + *update-display-peaches*

This procedure draws content relevant to peach production to the spatial arena in NetLogo, analogous to that done for the corresponding potato procedure.

* + *set-contracts-peaches*

The *set-contracts-peaches* procedure is analogous to the matching potato procedure. Commodity-specific calls are made to procedures to set contracts between peach product buyers and sellers. Commodities of peaches represented include conventionally produced, organic, and peach seconds. Scenarios alter the contracts put in place, specifically the products sold by distributors to the DPS buyer and seconds sold to processors.

* + *produce-peaches*

This procedure makes calls to *set-peach-yield*, *update-pad-production-year-peach*, *calculate-annual-production-peaches*, and *update-inventories-peaches*.

* + *set-peach-yield*

The *set-peach-yield* procedure assigns yields for the year to all peach-producing patches based on random draws from a normal distribution for conventionally and organically produced peaches.

* + *calculate-annual-production-peaches*

This procedure is analogous to the corresponding potato procedure. Based on yields assigned to patches producing peaches, the procedure calculates yields for individual farms and updates inventories.

* + *update-inventories-peaches*

The procedure updates inventories, assigning a percentage of fruit to be seconds and updating inventories for farmers. Annual production is tracked as well.

* + *update-peach-production*

Annual and total model run peach production are accumulated in this procedure, summing inventories for each commodity.

* + *fulfill-contracts-peaches*

The *fulfill-contracts-peaches* procedure is a series of calls to the *fulfillContracts*\_\* procedures for the appropriate commodities (i.e., conventional, organic, or seconds). Farmers sell to packers, packers to distributors, and farmers to distributors and alternative markets. Depending upon the scenario being represented, farmers may sell to processors and/or Denver Public School System buyers.

* + *dps-purchases-out-of-state-peaches*

As the name suggests, this procedure captures Denver Public School System buyers purchasing out-of-state peaches from one of the distributors. The price is dependent upon a random normal draw from parameters provided. Assets and inventories are then updated.

* + *update-bad-production-year-peach*

This procedure determines if the production year is a mediocre or bad one based upon a comparison of current yield to 70% of conventional or organic yield, yielding a bad year if below that value, and yielding a mediocre year if above that value but below 100% of typical yield.

* + *reset-storage-peaches*

This procedure calls *reset-storage\_peaches* (note underscore) for the agent types involved in the peach commodity chain. These include farmers, alternative markets, processors, distributors, and the Denver Public School System buyer.

* **peachContractNegotation.nls**
  + *setContracts\_peachesConventional*
  + *setContracts\_iqfConventional*
  + *setContracts\_peachesOrganic*

These lengthy procedures are analogous to the *setContract\_\** procedures used for potatoes. Prices are adjusted for the commodity, and then space available for whatever buyers are passed to the procedure are calculated. An assigned quantity of peaches are sold to buyers based on the nature of the farm. If a farm is large and distributors are purchasing product, 89% of the product is sold. If the farm is small or medium, 70% is sold to the distributor. If the farm is “split-pack”, 66% is sold to a packer if that is the buyer, and 23.8% sold to a distributor if that is the case. Estimates of quantities available for sale are updated for select sellers. Buyers within the market knowledge of the seller are queried as to their capacity to purchase product. Then bids are gathered by the seller from the buyers, and product sold to the bidder with the highest offer. Anticipated inventories are updated and contracts are created until the seller has allocated their anticipated production to buyers.

* + *priceAdjustments\_peachesConventional*
  + *priceAdjustments\_iqfConventional*
  + *priceAdjustments\_peachesOrganic*

These brief procedures are analogous to the *priceAdjustments\_*\* procedures used for potatoes. Buyers of a given commodity are passed to the appropriate procedure. Buyers then examine prices of other buyers within their market knowledge and the buyer with the lowest purchase price increases their price offering by 10%, and buyers with the highest offering decrease their price by 10%.

* + *contractSpaceAvailable\_peachesConventional*
  + *contractSpaceAvailable\_iqfConventional*
  + *contractSpaceAvailable\_peachesOrganic*

These brief procedures are analogous to the *contractSpaceAvailable\_*\* procedures for potatoes. If the buyer is a peach packer, one of the peach farmers with a duel role as packer is assigned that role. Buyers then have their inventory space available updated based on their maximum inventory minus the inventory they currently hold.

* **peachContractFulfillment.nls**
  + *fulfillContracts\_peachesConventional*
  + *fulfillContracts\_iqfConventional*
  + *fulfillContracts\_peachesOrganic*

These procedures are analogous to *fulfillContracts\_*\* procedures used for potatoes. The *fulfillContract* procedure type carries out the fulfillment of contracts among agents of different types passed to it for sellers and buyers. A given peach commodity is exchanged and money changes hand. The procedure begins by looping over sellers, which in turn loop over buyers. First, quantities available to sell and inventory of the buyer are double-checked. From these, the amount of peach exchanged is calculated and price set for the transaction, drawing these values from their locations in tables. Inventories are then updated, assets are changed to reflect the exchange of funds, and the transaction is updated in the tables to reflect the sale.

Echo statements in this procedure are not included in the above logic. They do not affect program logic, but rather provide information useful in validation.

* **peachResetStorage.nls**
  + *reset-storage\_peaches*

The *reset-storage\_peaches* procedure is analogous to the *reset-storage\_potatoes* procedure. This procedure receives an agent set that may be any of the peach food system participants. Then depending upon the agent set, the procedure sells any remaining inventory of the appropriate commodity to an outside buyer and then resets the current inventory to zero. DPS buyers do not sell excess inventory to outside interests.

* **beefDashboard.nls**
  + *set-precip*

The procedure *set-precip* assigns a yearly precipitation surface to use to estimate forage production, from 2001 to 2010, and then loads that image. The procedure calculates a percentage of precipitation relative to a long-term average, and then for each beef rancher, assigns a yearly rainfall ratio and sets the number of cows that calve.

* + *set-forage*

For patches that are involved in beef production and are not irrigated, annual forage production is estimated based on a mean forage production value multiplied by the percent of normal precipitation result.

* + *update-display-beef*

The *update-display-beef* is analogous to the *update-display-potatoes* procedure. A widget on the interface allows for a given type of information to be displayed on the spatial arena in the NetLogo interface, and this procedure colors the beef ranches patches according to that choice (e.g., beef: public-private; beef: grazable; farm ownership).

* + *reset-storage-beef*

This procedure is simply a series of calls to *reset-storage\_beef* (note underscore), one for each of the agent sets involved in the beef supply chain (i.e., ranchers, packing houses, distributors, and the Denver Public School System buyer).

* + *produce-beef*

The *produce-beef* procedure calls in turn *set-precip*, *set-forage*, and *ranch-stats*.

* + *ranch-stats*

The lengthy *ranch-stats* procedure calculates ranch characteristics, with different values potentially used based on the yearly rainfall ratio. If yearly rainfall is 80% or more below normal, a given set of calculations update cows culled in the year, calves weaned, replacement cows purchased, calf sales, and cows that calve. If precipitation is between 80% and 110% of normal, the same calculations occur with potentially different coefficients, plus cows in commodity groups (i.e., conventional, grassfed, Colorado sourced, and animal welfare) are culled. If rainfall is above 110% of normal, similar calculations are made, but with additional cattle purchased to return the herd to a baseline number. The procedure ends with variable passthrough calculations that are scenario specific.

* + *update-beef-production*

This procedure updates the annual and total model run production accumulators for each of the beef commodities, estimating 1200 pounds produced from each culled animal.

* + *sell-beef-spot-market*

The *sell-beef-spot-market* is the location where beef are sold in FFAR-ABModel, counter to other commodities that use contracting. First, beef packing houses calculate the numbers of animals needed. If the packing house is large, only conventional animals are needed. If the packing house is small, all commodity types are produced. In either case, the numbers of animals needed by the packing house is calculated based on the maximum inventory space of the Denver Public School System buyer and the average meat production from each animal of 1200 pounds.

Packing houses then buy beef from ranchers until the packing houses have met their need. Bids are gathered from ranchers with animals to sell, and then the rancher with the highest transition likelihood score is selected for a purchase. Animals are purchased from that rancher with the price determined by the scenario set by the user. Cows are then exchanged, assets are updated, and inventories are adjusted. An analogous process happens for distributors purchasing product from packers.

The procedure ends with the Denver Public School System buyer purchasing from either the distributors in the baseline scenario, or from the packing houses in scenarios with an abbreviated supply chain. Inventories are adjusted for the seller and the school system buyer, and prices are determined from random draws from normal curves defined based on inputs to the FFAR-ABModel. Assets are tracked as meat is provided, decreasing for the Denver Public School System buyer and increasing for the meat seller.

* **beefResetStorage.nls**
  + *reset-storage\_beef*

This procedure is analogous to the *reset-storage\_potatoes* procedure. For a given agent set passed to it (e.g., beef rancher, packing houses, distributors, Denver Public School System buyer), the set of commodities appropriate to the agent are sold to a buyer outside the Colorado system and assets and inventories are adjusted. The Denver Public School System buyer does not sell surplus product to an outside interest.

* **matrixToTableAPI.nls**
  + *createContractFromScratch*

This reporter creates an element in a table and fills that element with a contract key and contract value. The element is returned.

* + *debugMatrix*

The *debugMatrix* procedure formats a table passed to it for printing in a legible way.

* + *putContractValue*

For a given table passed to the procedure, a contract is put in place for parties that are passed, with the value of the contract filling an element of the linked players involved.

* + *sumContractValue*

The procedure sums the value of a contract and updates that value in the passed table.

* + *table\_set*

The *table\_set* procedure is passed a contract table, identifiers for those contracting, and a key and value. If the table exists, the value in the table is updated with the value passed. If not, the table is created and then updated.

* + *table\_create\_and\_sum*

The *table\_create\_and\_sum* procedure is similar to *table\_set*, but does a summation of the values in the passed table if it exists. Otherwise, it creates the table as in *table\_set*.

* + *table\_sum*

Through a call to *SumContractValue*, *table\_sum* provides a sum of the contract between the parties involved.

* + *check-entry*

In the reporter *check-entry*, a check is made to see if parties passed to the procedure have a key in the table passed. If it exists and the value is greater than zero, TRUE is returned, otherwise FALSE is returned. If no key exists, *update-outstanding-quantity* is called and the reporter continues as before.

* + *update-outstanding-quantity*

This procedure subtracts a contract value from an entry for corresponding participants and table passed to the procedure. It does so through a call to *subtractContractValue*.

* + *subtractContractValue*

The *subtractContractValue* procedure subtracts an entry from a contract entry for which there is a valid key for the participants and table passed to it.

* + *check-bigger-than-entry*

This reporter looks-up an entry in a table passed to it, and if larger than a comparison value passed as well, reports TRUE, otherwise the reporter provides FALSE.

* + *report-value*

The report-value reporter provides the value of a contract table entry for a row and column pass to it.

**Literature Cited**

Grimm, V, Berger U, Bastiansen F, Eliassen S, Ginot V, et al. 2006. A standard protocol for describing individual-based and agent-based models. Ecological Modelling 198:115-126.

Grimm V, Berger U, DeAngelis DL, Polhill JG, Giske J, Railsback SF. 2010. The ODD protocol: a review and first update. Ecological Modelling 221:2760-2768.

Grimm V, Railsback SF, Vincenot CE, Berger U, Gallagher C, et al. 2020. The ODD protocol for describing agent-based and other simulation models: a second update to improve clarity, replication and structural realism. Journal of Artificial Societies and Social Simulation 23(2):7. Http://jasss.soc.surrey.ac.uk/23/2/7.html. DOI: 10.18564/jasss.4259.

Hoogenboom, G., C.H. Porter, K.J. Boote, V. Shelia, P.W. Wilkens, U. Singh, J.W. White, S. Asseng, J.I. Lizaso, L.P. Moreno, W. Pavan, R. Ogoshi, L.A. Hunt, G.Y. Tsuji, and J.W. Jones. 2019. The DSSAT crop modeling ecosystem. In: p.173-216 [K.J. Boote, editor] Advances in Crop Modeling for a Sustainable Agriculture. Burleigh Dodds Science Publishing, Cambridge, United Kingdom

Hoogenboom, G., C.H. Porter, V. Shelia, K.J. Boote, U. Singh, J.W. White, W. Pavan, F.A.A. Oliveira, L.P. Moreno-Cadena, J.I. Lizaso, S. Asseng, D.N.L. Pequeno, B.A. Kimball, P.D. Alderman, K.R. Thorp, M.R. Jones, S.V. Cuadra, M.S. Vianna, F.J. Villalobos, T.B. Ferreira, W.D. Batchelor, J. Koo, L.A. Hunt, and J.W. Jones. 2021. Decision Support System for Agrotechnology Transfer (DSSAT) Version 4.8 (DSSAT.net). DSSAT Foundation, Gainesville, Florida, USA.

Jones, J.W., G. Hoogenboom, C.H. Porter, K.J. Boote, W.D. Batchelor, L.A. Hunt, P.W. Wilkens, U. Singh, A.J. Gijsman, and J.T. Ritchie. 2003. DSSAT Cropping System Model. European Journal of Agronomy 18:235-265.

Repasy K, Llieva RT, Willingham C. 2019. Bringing the Good Food Purchasing Program to New York City: barriers and facilitators for select institutions. CUNY Urban Food Policy Institute, New York.

Wilensky, U. 1999. NetLogo. Center for Connected Learning and Computer-based Modeling, Northwestern University, Evanston, IL.