



## Chapter 3 – Intermediate Operations

### Lesson A| Aggregate | 5 Minutes

#### Lesson Goals

In this Lesson, you will:


- Use the Aggregate Transform to create data summarization
- Create a Flat Aggregate


#### Lesson Instructions

**YOUR GOAL:** Compute the average number of market days for each state.

- 1. Add an additional Recipe to the US Farmers Market Dataset**

You now want to build an aggregation on top of the current US Farmers Market dataset. However, an aggregation will change the granularity of the dataset, and you do not want to lose any of the individual market-level data. Consequently, you need to add a recipe specifically for this aggregation step, while retaining all of the details of the dataset in a separate **Wrangled Dataset**.

Return to the **Training** Flow details page. Click on the icon for the Wrangled US Farmers Markets Dataset, and click the  icon.

Select **Add new Recipe**. Trifacta will create a new recipe. Select the new Wrangled Dataset off of this new recipe and again click the  icon. **Edit name and description** and change the name to US Farmers Markets Aggregate. You will build the aggregation inside this new recipe. You can now see that you have two different recipes, and two different Wrangled datasets branching off from the source dataset for **US Farmers Markets**

Click on the **US Farmers Markets Aggregate** wrangled dataset and **Edit Recipe**.

  - Notice that the new dataset begins where the last Wrangled US Farmers Markets dataset left off.



2.	<p><b>Aggregate data by state</b></p> <p>Click on the Column Menu for the Spring_Occur_Total column, find <b>Aggregate</b>, select <b>As a new table</b>, and finally choose <b>Average</b>.</p> <p>This will generate a template in builder to find the average of the Spring_Occur_Total column.</p> <p>The <b>Functions</b> section is where you can aggregate your data with mathematical functions such as sum, average, max, etc.</p> <p>In the <b>Group by</b> box choose the <b>Address_State</b> column.</p> <div data-bbox="315 703 889 884"><p>The <b>Group By</b> parameter specifies the column or columns unique values on which to apply the calculations in the <b>Function</b> parameter.</p></div> <p>Notice the preview in yellow above the Builder. You can scroll through to preview the results.</p> <p><b>Add</b> the Aggregation to your recipe.</p>	<pre>aggregate value: average(Spring_Occur_Total) group: Address_State</pre>
3.	<p><b>Flat Aggregate</b></p> <p>If you want to maintain the details of your dataset, you can aggregate using the <b>Derive</b> transform. Insert a step before the Aggregate step we just created by hovering over the aggregate step, choosing the three dot icon, and select <b>insert step before</b>.</p> <p>Again, click on the Column menu of the Spring_Occur_Total column and find <b>Aggregate</b>. This time, we will select <b>As a new column</b> and again will chose <b>Average</b>. Group by Address_State, and you'll see in the preview we get the same values as the Aggregate, but as a new column while maintaining the details of the dataset. Click Cancel in the Transform Builder.</p>	<pre>Derive value: average(Spring_Occur_Total) group: Address_State</pre>