The Great Dynamo Dig
Mining Your Revit Model with Computation

Autodesk University, 2014

Nathan Miller, n.miller@case-inc.com

Mining HDR Architecture's Central Washington Hospital with Dynamo

1 - Class Summary

This class will introduce concepts that will help you creatively capitalize on your Revit software data with the Dynamo visual programming language extension. The Dynamo extension enables users with powerful data-mining capabilities through a graphical user interface. These capabilities, once only available to Revit software API experts, have made it easier to get to your Revit software data, to manipulate it, and to stream it to many external sources. The class will introduce key concepts for accessing, formatting, and sharing Revit software model data using the Dynamo extension and packages such as LunchBox and Slingshot.

Learning Objectives

- Learn how to use Dynamo for Revit data mining.
- Learn how to extract element parameters and geometry from Revit
- Learn how to format Revit software data for use by other applications
- Learn how to connect the Dynamo extension to relational databases.

Class Itinerary

- The Basics: From Lists to Tables
- Reading and Writing Data with Excel Nodes
- Get Data From the Revit Model
- Design a Dashboard with Revit Data
- Databases and Beyond!

2 - The Basics: From Lists to Tables

Before we attempt to start navigating the data sets that Revit and Dynamo can provide us with, we should begin with a fundamental understanding of how we can think of lists and nested lists as a way to represent tables in Excel or a Database.

This node shows two sets of lists as they would appear within a table structure. Note how each list appears as a row within the table. (2 rows, 5 columns)
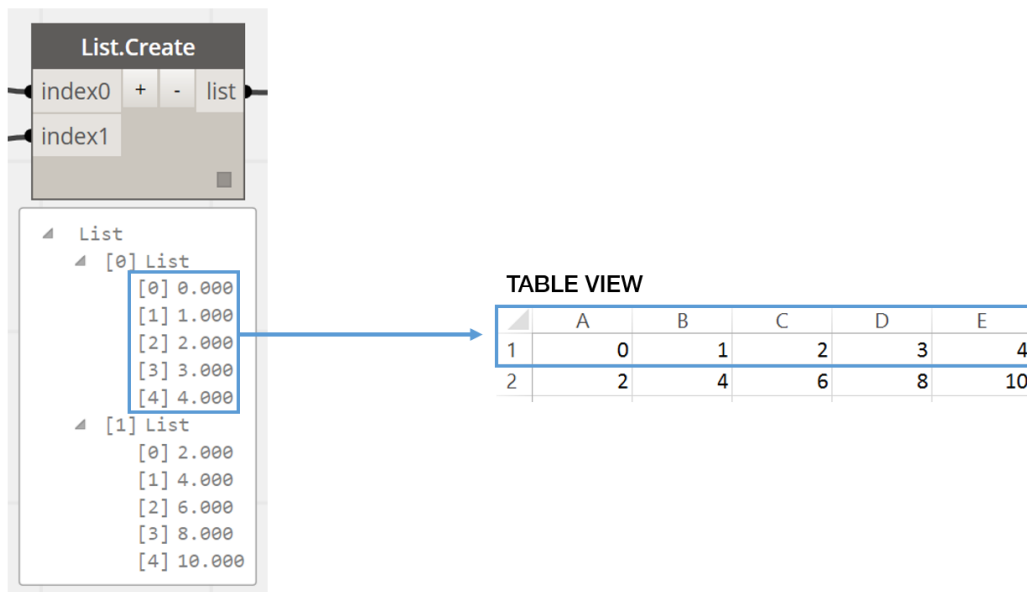

By "Transposing" the Dynamo lists, we can re-organize our table to be based on 5 rows and 2 columns
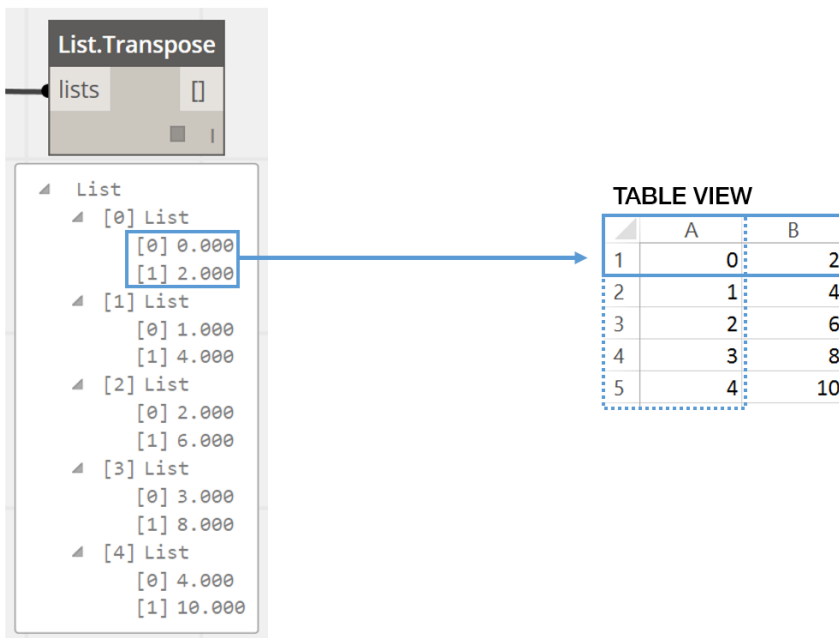
3 - Reading and Writing Data with Excel Nodes

Dynamo comes with some great out-of-the-box tools for putting these basic list-table concepts to use. By searching for "Excel" within Dynamo, you can see that two nodes for reading and writing Excel files are available to use. It should be noted that the Excel nodes are designed to work with 'normal' tables of data on worksheets with a simple column / row organization.
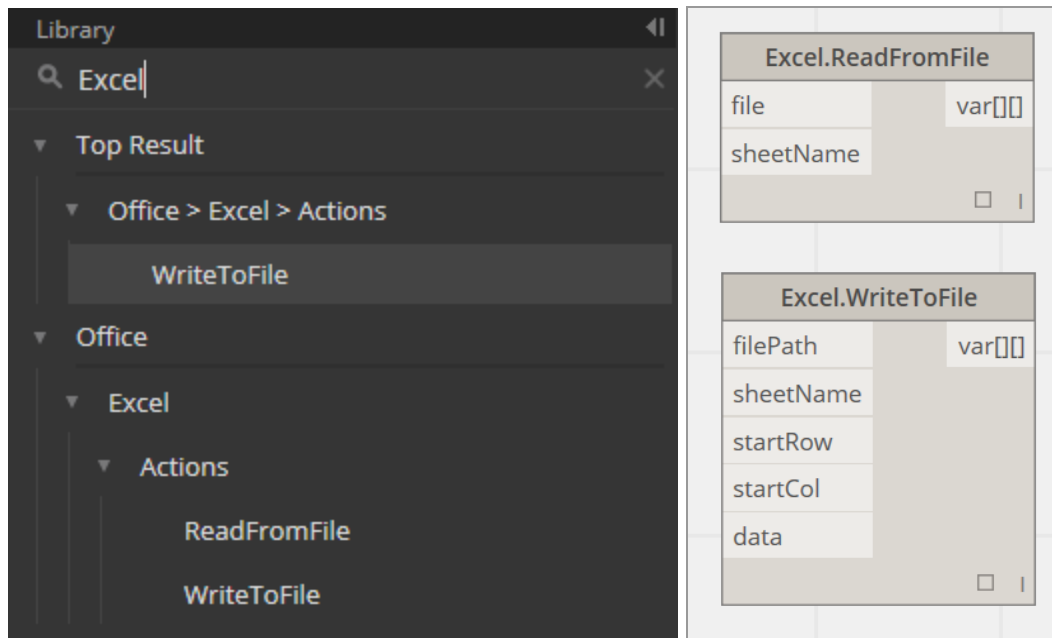
Reading Excel data requires that the user specify a path to an existing Excel file and determine the name of the sheet to pull data from. The Excel node will return a nested list structure ***organized by row.***
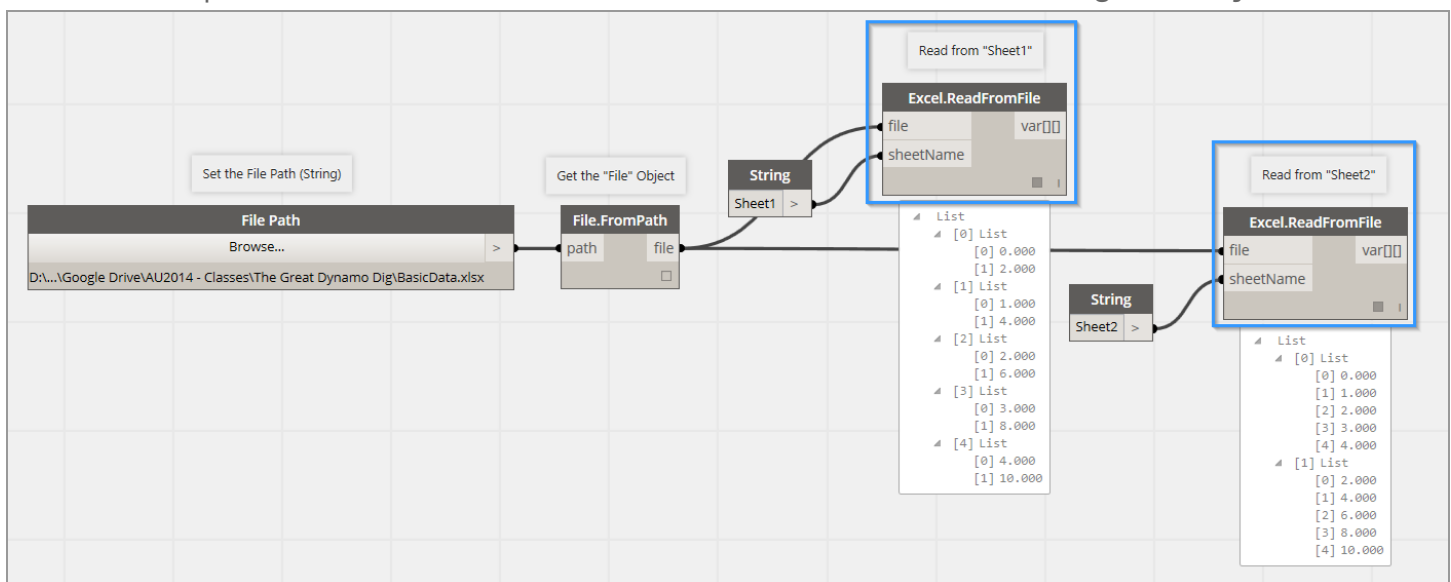
Writing Excel data is just as straight forward although you can also specify the starting position of the table within the worksheet. 0,0 will align with Excel field A:1
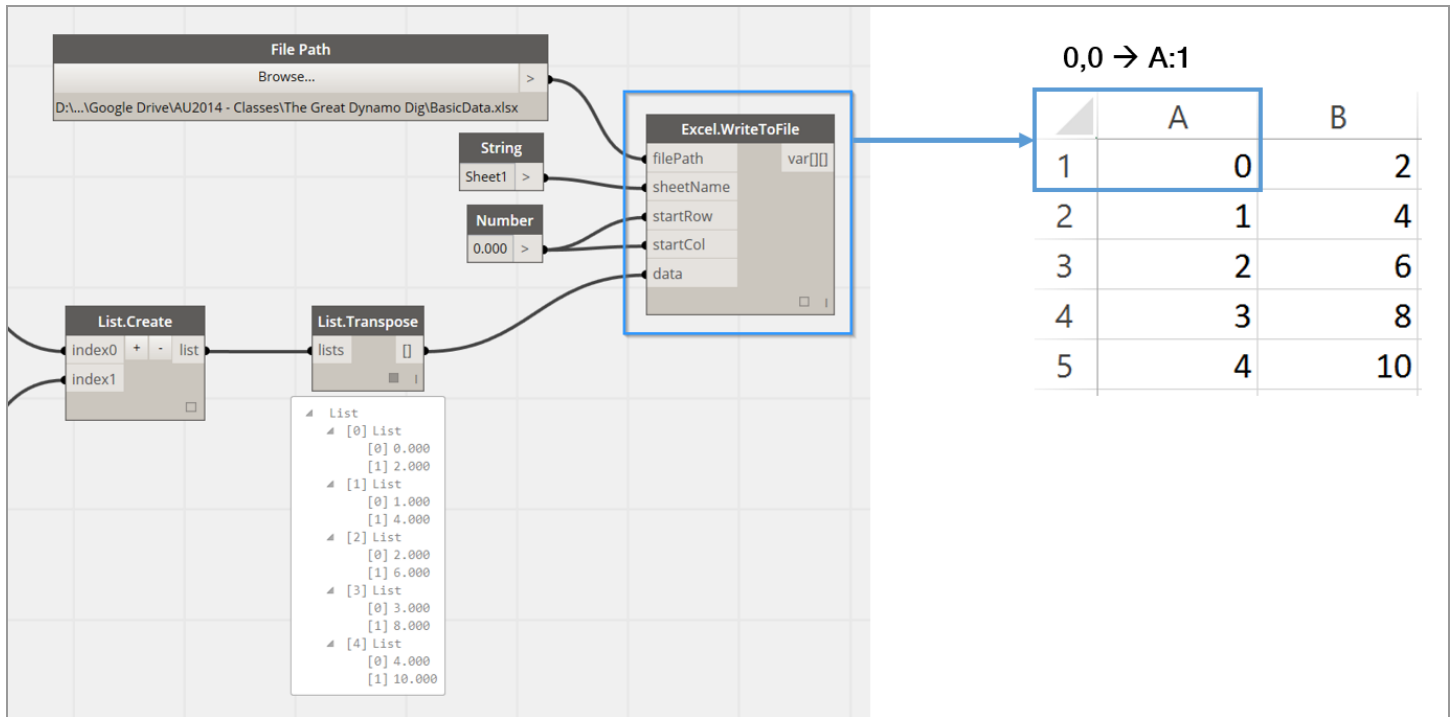
4 - Getting Data from the Model

When launched inside of Revit, Dynamo is granted access to just about every element from the model along with data about that element. There are basic tools that come out-of-the-box with Dynamo, to more advanced toolkits freely available on the package manager, such as LunchBox. Once you have access to this information within Dynamo, you can use the Excel workflows to read and explore the data.

Example: Revit Rooms to Dynamo and Excel with LunchBox Nodes
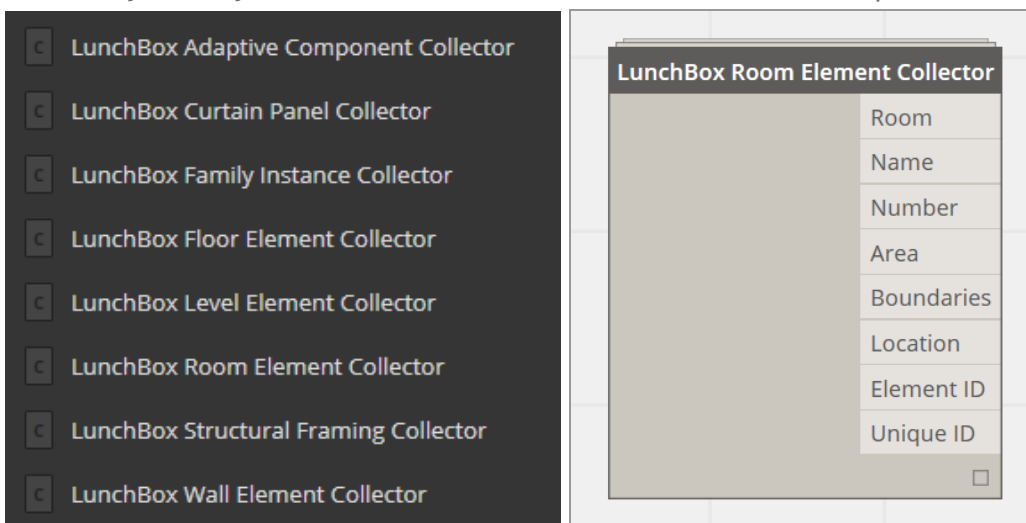
LunchBox is a free Python-based package available for Dynamo. LunchBox is authored and maintained by yours truly :)   After you have LunchBox installed, you will have access to several "Collector" nodes that are designed to collect elements from the model and provide instant access to key element data. The "Room Element Collector" is one of the more powerful nodes as it allows the user to collect all placed rooms in the model including properties such as Name, Number and Area. The LunchBox nodes also expose values such as the Element ID and Unique ID which are not exposed by Revit through the UI.

These are rooms in the 'advanced' sample project that ships with Revit.

Using Lunchbox, we can access each of these elements using the **Room Element Collector Node**

We can then format the Room data for use with the Excel node. The definition below outlines a common approach to this. One key idea here is adding a "Header" value to the front end of each data list.
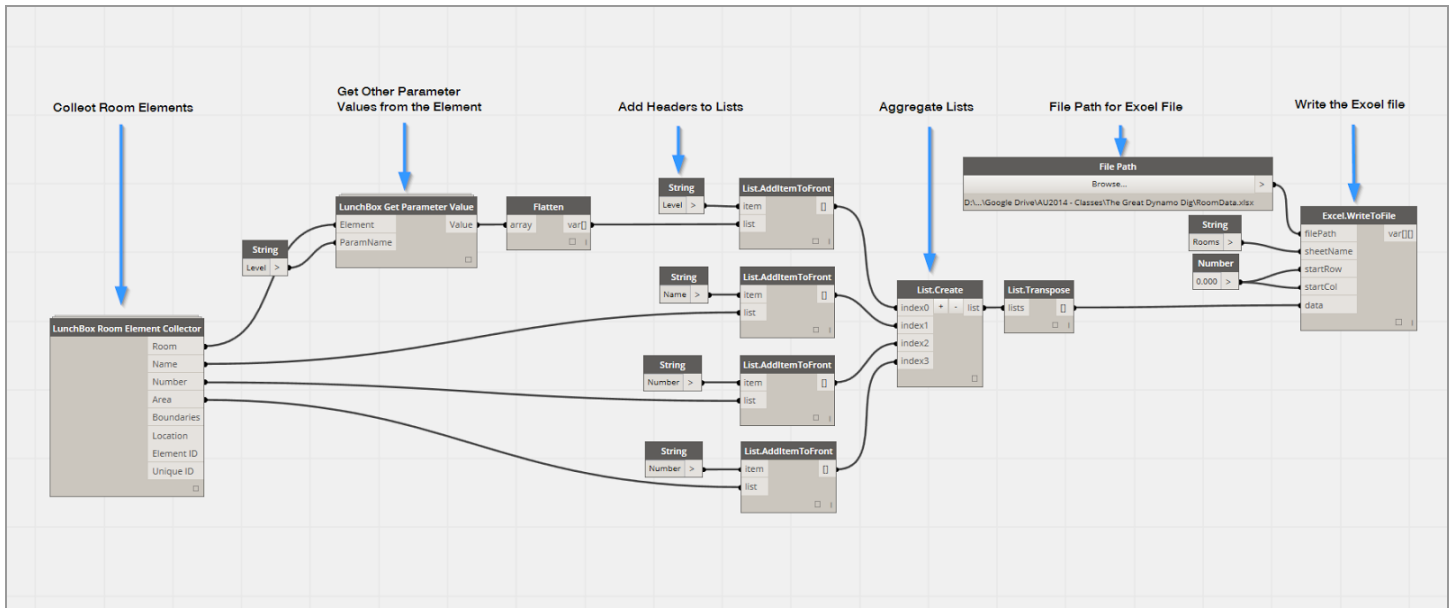
When run, the definition will produce the following Excel table.

|	A	B	C	D
1	Level	Name	Number	Number
2	01 - Entry Level	Vest.	101	439.1314479
3	01 - Entry Level	Lobby	102	3521.836303
4	01 - Entry Level	Cafeteria	121	1581.870504
5	01 - Entry Level	Prep/Dish	122	239.0518895
6	01 - Entry Level	Dry Storage	124	91.09187116
7	01 - Entry Level	Electrical	125	61.25026603
8	01 - Entry Level	Conference	123	446.7195144
9	01 - Entry Level	Office	127	165.3274404
10	01 - Entry Level	Admin	126	169.0894565
11	01 - Entry Level	Storage	128	104.4129538
12	01 - Entry Level	Toilet	129	60.01597648
13	01 - Entry Level	Stair	130	203.6271364
14	01 - Entry Level	Corridor	131	587.9719503
15	01 - Entry Level	Sprinkler	119	95.80441299
16	01 - Entry Level	Electrical	118	187.0550029
17	01 - Entry Level	Instruction	117	522.6845317
18	01 - Entry Level	Lounge	120	439.942834
19	01 - Entry Level	Conference	116	339.7879723

5 - Design a Dashboard with Revit Data

Now that we have a table of Revit data within Excel, we can start to be creative with how we start to work with the data and gain insights into the building. Even with the room data that we have, we can produce a rather rich dashboard and understand the information in a way that a Revit schedule could not communicate.

Within Excel, let's start by creating a **pivot table**. A pivot table allows us to create summaries of our raw data in a way that makes sense. We can then use these tables to create pivot charts and graphs.
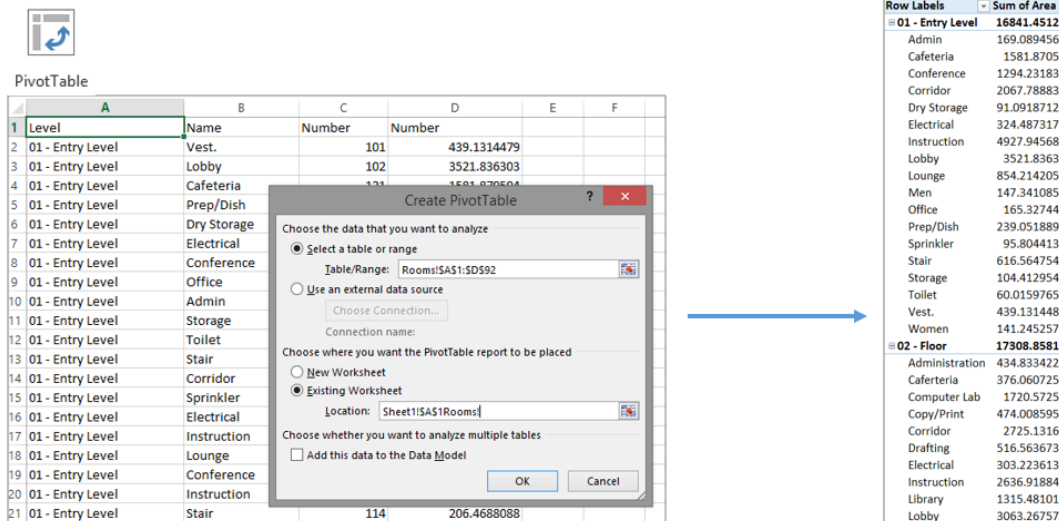
With a **pivot chart** we can easily begin to slice and dice the data and begin to construct different ways to interact with the data such as creating charts and graphs.
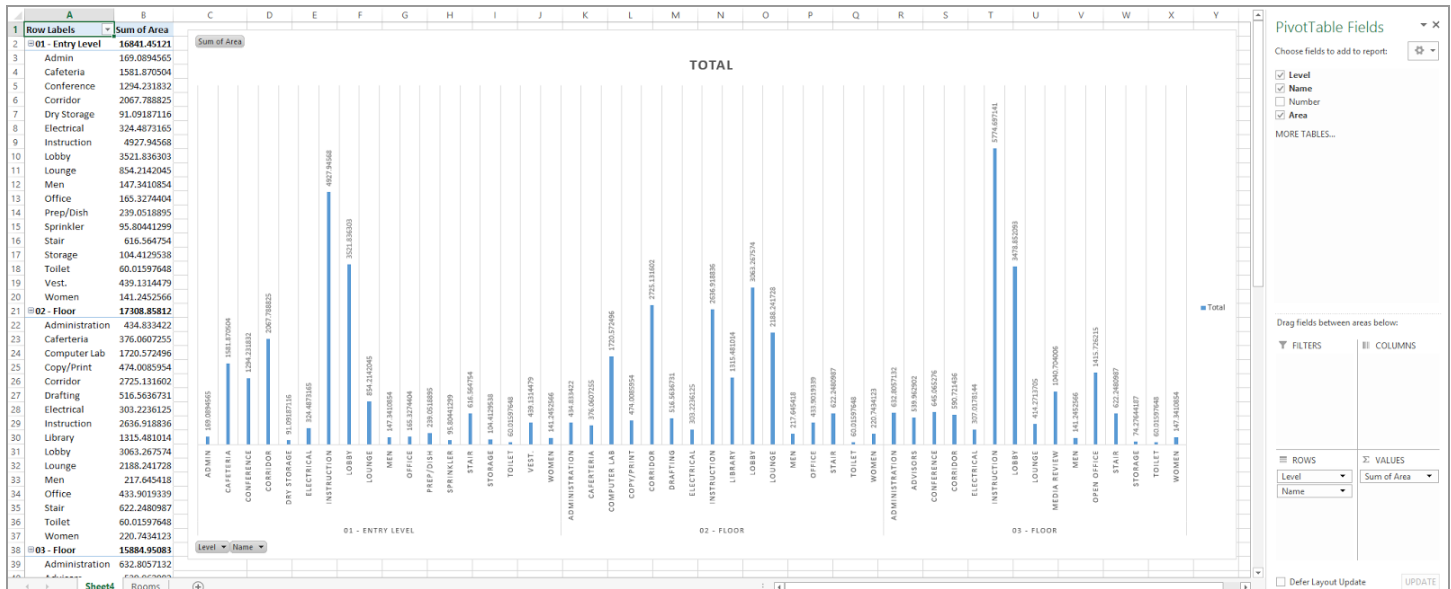
A slicer provides an interactive way to manage data in pivot tables and charts. For example, I below is a slicer to for isolating program by level in the building. The slicer controls the information being displayed in the pivot table and the pivot chart.
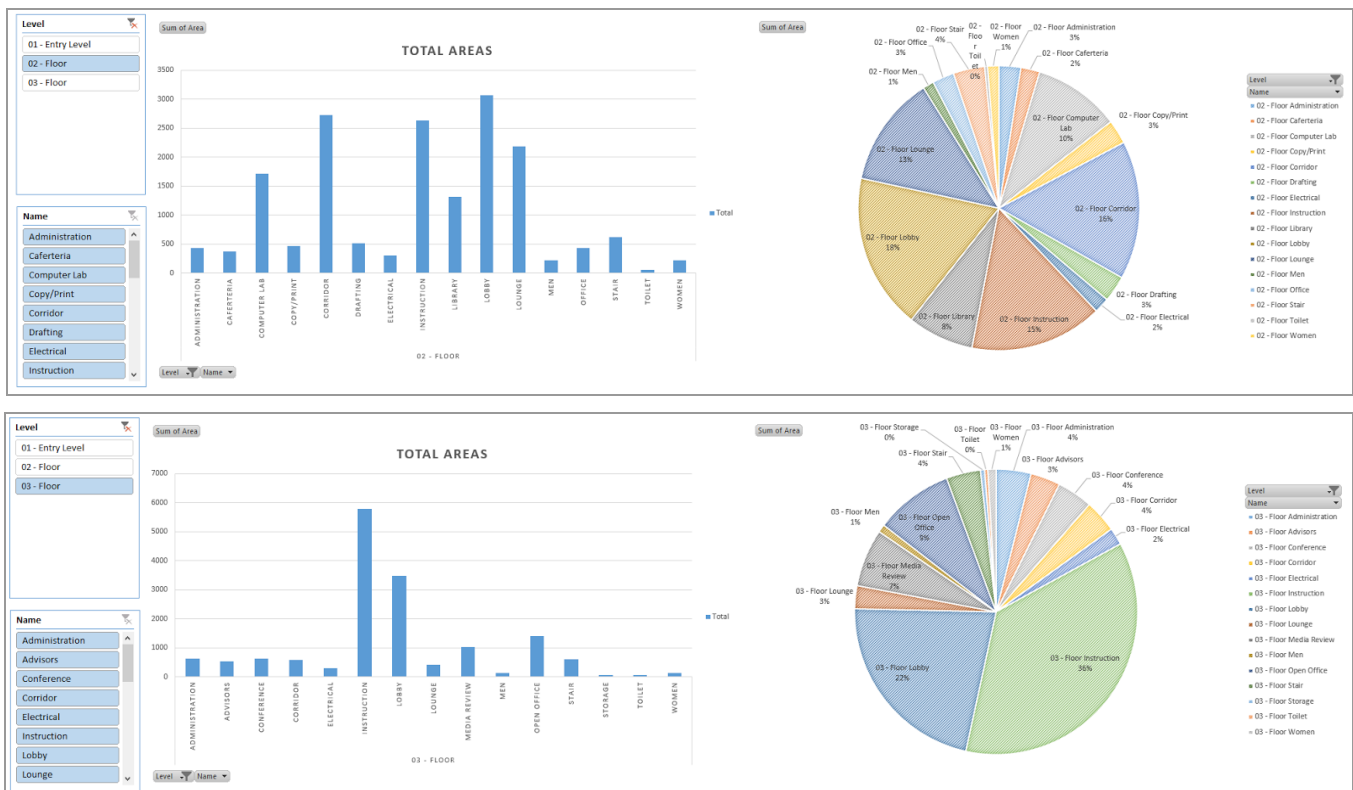
6 - Databases and Beyond!

As you begin to develop your own personal approach to using Dynamo for mining Revit information, you might find that using Excel worksheets and pivot charts does not scale well for large datasets and information aggregated from multiple models. You might consider implementing similar approaches by using a relational database with a tool, like Tableau, which is designed for visualizing large data sets and more complex queries. To help facilitate this, the open source **Slingshot package** for Dynamo provides data connections to databases such as MySQL, SQLite, and others supporting ODBC or OLE DB connections.