

AS320955-L

# Revit Data Visualized with Dynamo

**Carl Storms**  
IMAGINiT Technologies

## Learning Objectives

- Explore ways to manipulate Revit data using Dynamo
- Use Dynamo to add outside data to your Revit model
- Explore Dynamo packages that improve Revit data visualization
- Learn to create a basic dashboard with Dynamo

## Description

This Lab is **NOT** for Dynamo pro's, but for the Dynamo novice looking to expand their skills and do more with their data.

In this session, we will learn how to manage and manipulate data from a Revit file with Dynamo. We will also learn how to use Dynamo to help us add data to a Revit model from another source, like excel. These days when working with project data, visualization of that data is a must, we will explore this using the Archi-lab Mandrill Dynamo package from Konrad Sobon.

We will also explore some of the new features and updates of the latest release of Dynamo.



QR Code to  
access the  
latest version  
of this handout  
and  
presentation

## Speaker



**Carl Storms** – Solutions Consultant

**IMAGINiT Technologies** – Building Solutions Division

Drawing on his 20+ years of experience in architecture, engineering, and construction, **Carl Storms** provides a practical and well-rounded understanding of building information modeling (BIM) to clients. He's worked in residential and commercial architecture, as well as construction and has nearly 10 years of teaching experience at the collegiate and industry level. This experience aids him in providing the business case for BIM and how it makes the most of collaboration, coordination and design tools and processes.

Through implementations, instruction, mentoring, webinars and global speaking engagements, Carl assists the AECO community with the adoption of design technology and BIM processes. As someone who truly enjoys the process of building information modeling, Carl spreads his love of all things BIM via Twitter [@theBIMsider](https://twitter.com/theBIMsider), on his Blog [theBIMsider.com](http://theBIMsider.com), or as a co-host on two AECO Industry Podcasts: [The Simply Complex Podcast - Conference Edition](#), as well as [BluePrints](#) a podcast sharing ideas and knowledge about the AECO industry.

## Session Handout and Presentation

Get the most current PDF version of this handout and presentation using the Dropbox link below:

[https://www.dropbox.com/sh/snfhd8mw7qqdmbd/AAC\\_RjrnPJOz7cPxavRmdlla?dl=0](https://www.dropbox.com/sh/snfhd8mw7qqdmbd/AAC_RjrnPJOz7cPxavRmdlla?dl=0)

# Table of Contents

<b>Exercises .....</b>	<b>5</b>
<b>Packages Required .....</b>	<b>5</b>
Dynamo & Revit Version .....	5
<b>Ex 0) Data .....</b>	<b>5</b>
Task 1 .....	5
Task 2 .....	10
Task 3 .....	13
<b>Ex 1) Warnings .....</b>	<b>16</b>
Task 1 .....	16
Task 2 .....	18
<b>Ex 2) Count and Colour .....</b>	<b>21</b>
Task 1 .....	21
Task 2 .....	23
<b>Ex 3) Doors .....</b>	<b>29</b>
Task 1 .....	29
Task 2 .....	30
<b>Ex 4a) Pie - NodeModelCharts .....</b>	<b>38</b>
Task 1 .....	38
Bonus Task .....	44
<b>Ex 4b) Pie – Data-Shapes .....</b>	<b>48</b>
Task 1 .....	48
<b>Ex 4c) Donut - Mandrill .....</b>	<b>58</b>
Task 1 .....	58
Task 2 .....	62
Bonus Task .....	67
<b>Bonus) Dashboard .....</b>	<b>70</b>
Task 1 .....	70

<b>Resources.....</b>	<b>79</b>
List@Level .....	79
Past Labs and Presentations .....	80
Blogs & Websites.....	81
Courses & Training.....	83
Free Graphs .....	85
Podcasts & Videos.....	86
23 Dynamo Packages You Should Check Out .....	87



## Exercises

### Packages Required

- Archi-lab\_Mandril
- archilab
- Bang
- Clockwork for Dynamo 2.x
- Data-Shapes
- NodeModelCharts
- Rhythm

### Dynamo & Revit Version

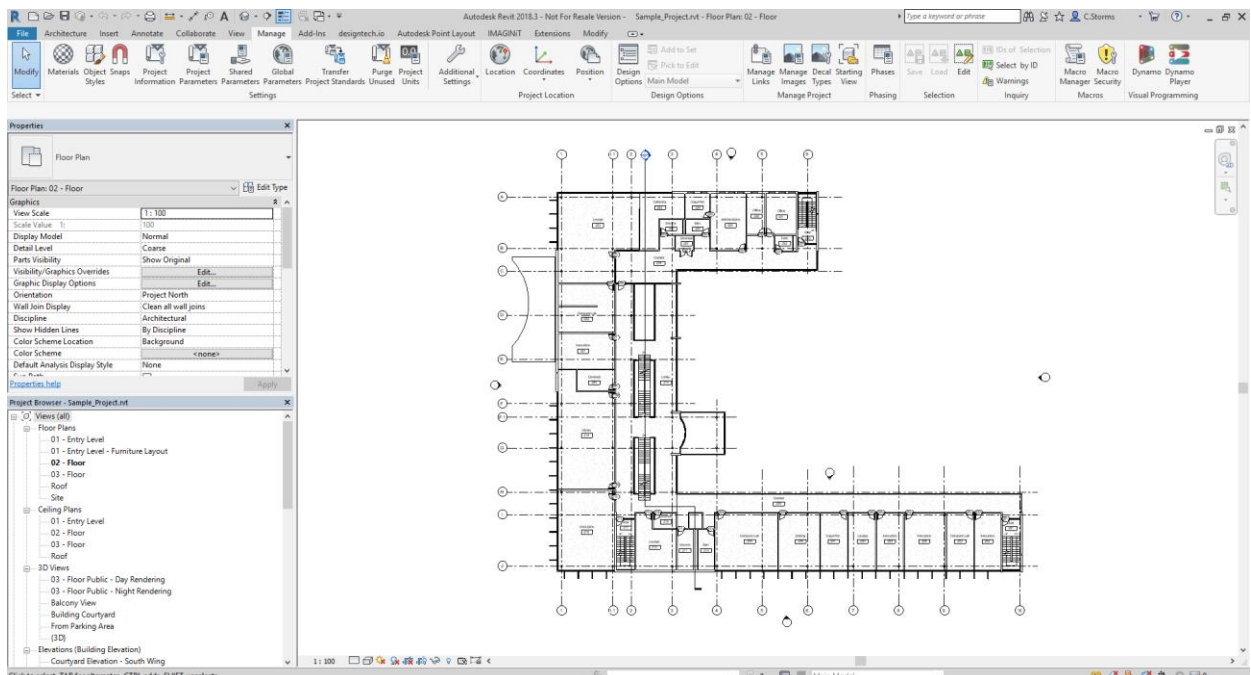
All exercises were created using Revit 2020.1 and Dynamo Revit 2.1.0 (Core 2.2.1)

### Ex 0) Data

In this exercise we will look at some useful nodes to help quickly review data from within your Revit project.

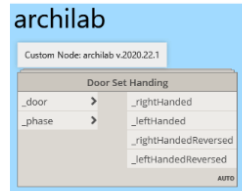
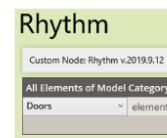
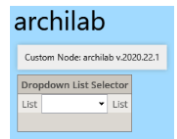
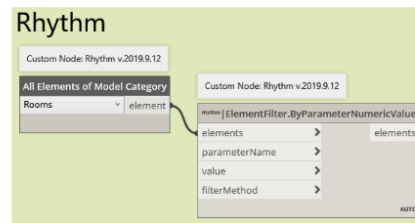
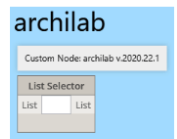
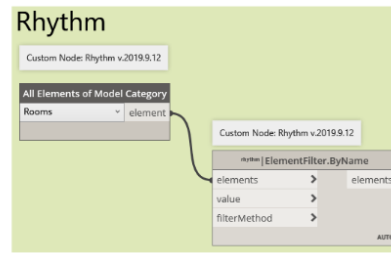
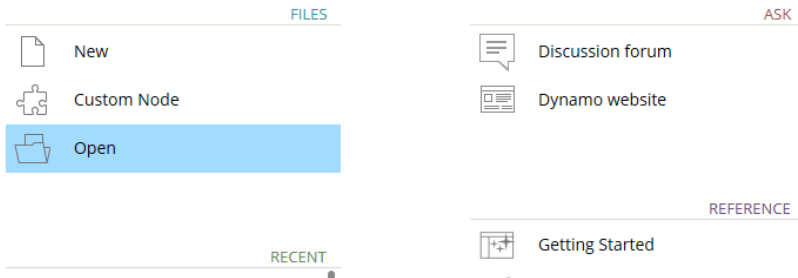
#### Task 1

1. From the dataset folder **Ex\_0** open the following file: **Sample\_Project.rvt** in Revit.



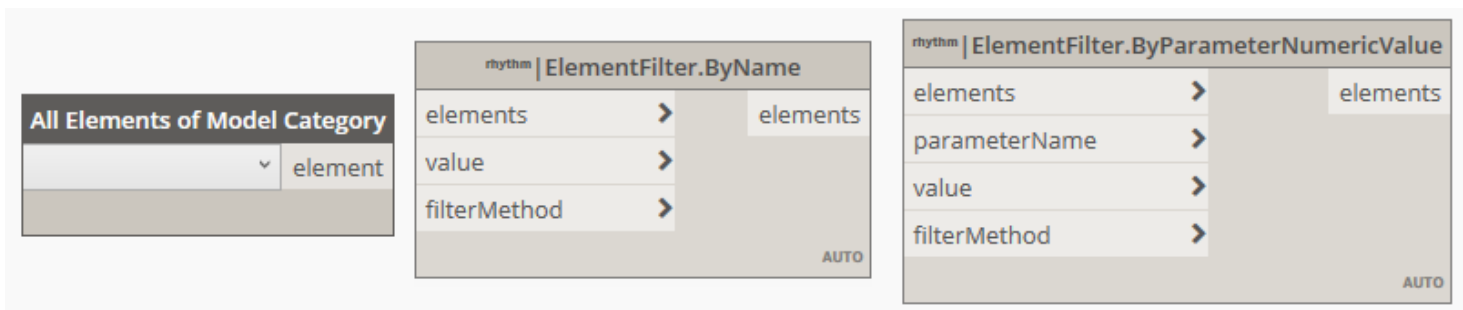
2. Start Dynamo from the Manage tab.
3. From the Start Page, under the FILES section click Open. Then navigate to the class dataset folder **Ex\_0** open the following file: **Ex0\_Data\_T1\_Begin.dyn**.

# Dynamo



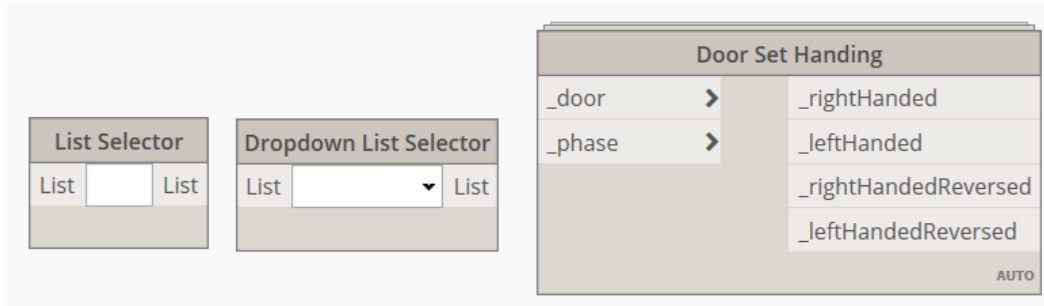
4. With the graph we just opened, we are now going to see what some custom nodes can do with our Revit project data. We will be looking at three nodes from the **Rhythm** package from [John Pierson](#).

- ***All Elements of Model Category***
- ***ElementFilter.ByName***
- ***ElementFilter.ByParameterNumericValue***

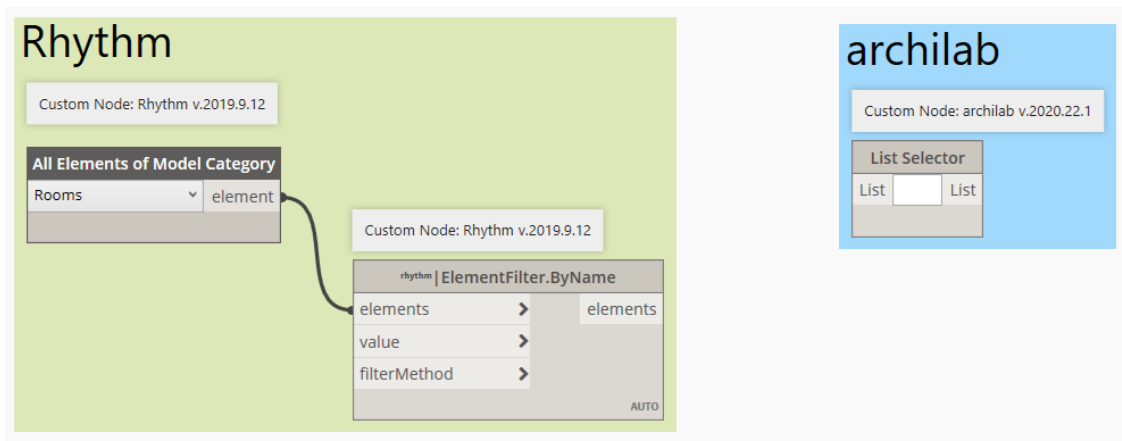


As well as three nodes from the **archilab** package from [Konrad K Sobon](#).

- **List Selector**
- **Dropdown List Selector**
- **Door Set Handing**

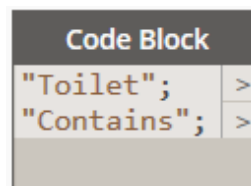


5. We will begin with the top example in the graph and add a few nodes to be able to filter rooms in the project by a key word, in this case: *Toilet*. We will be adding six out of the box nodes: **Code Block**, **Room.Number**, **Room.Name** & **Watch** (x3).

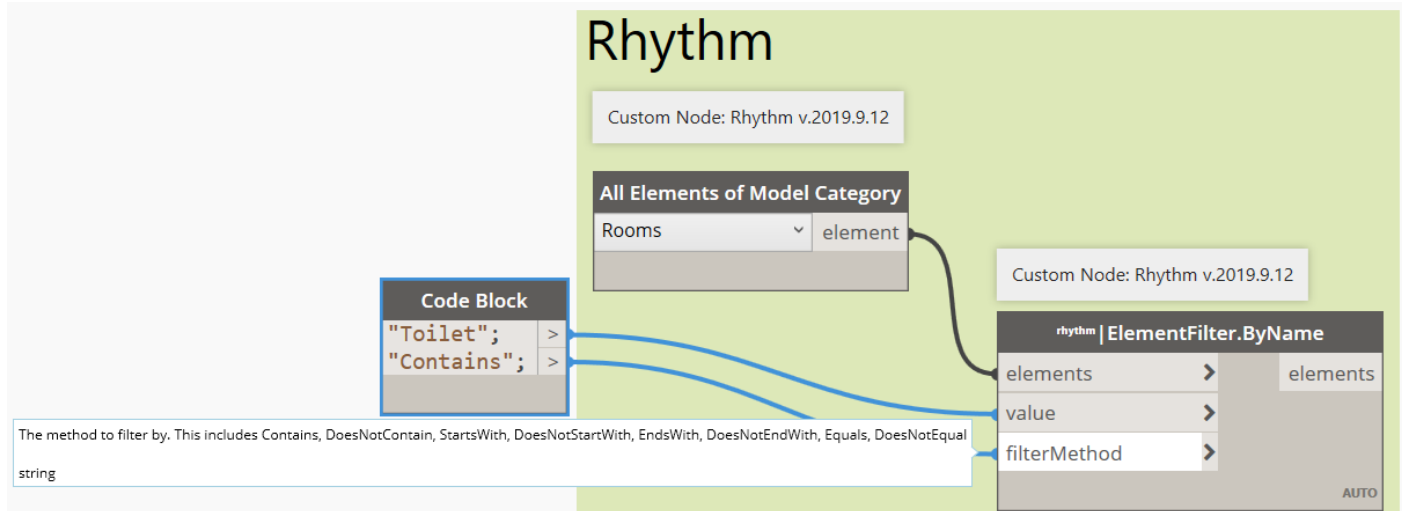


6. Confirm that *Rooms* is selected in the **All Elements of Model Category** node.
7. Double click in the graph to add the **Code Block** to the screen. Click inside the **Code Block** to add the following as shown.

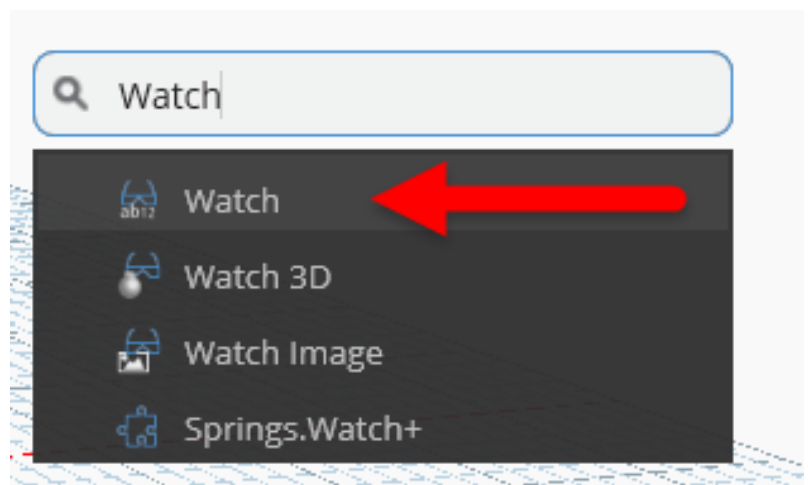
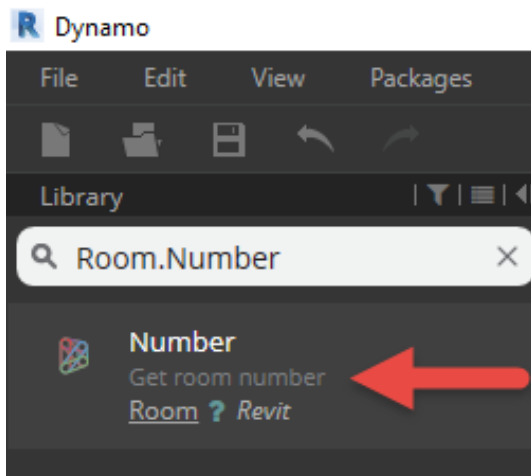
- "Toilet";
- "Contains";



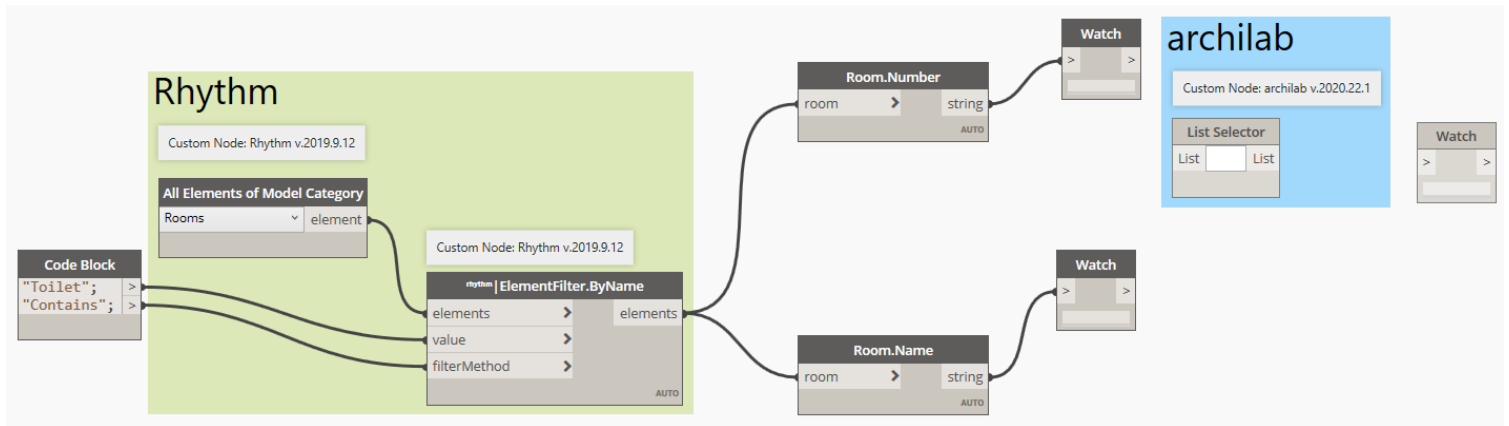
8. Next you can connect the wires from the **Code Block** to the **ElementFilter.ByName** node. "Toilet" goes into the *value* input and "Contains" goes into the *filterMethod* input. **Note:** if you hover over the *filterMethod* input you will get a pop-up that tells you all the different filter methods this input will work with: Contains, Does Not Contain, Starts With, Does Not Start With, Ends With, Does Not End With, Equals, Does Not Equal.



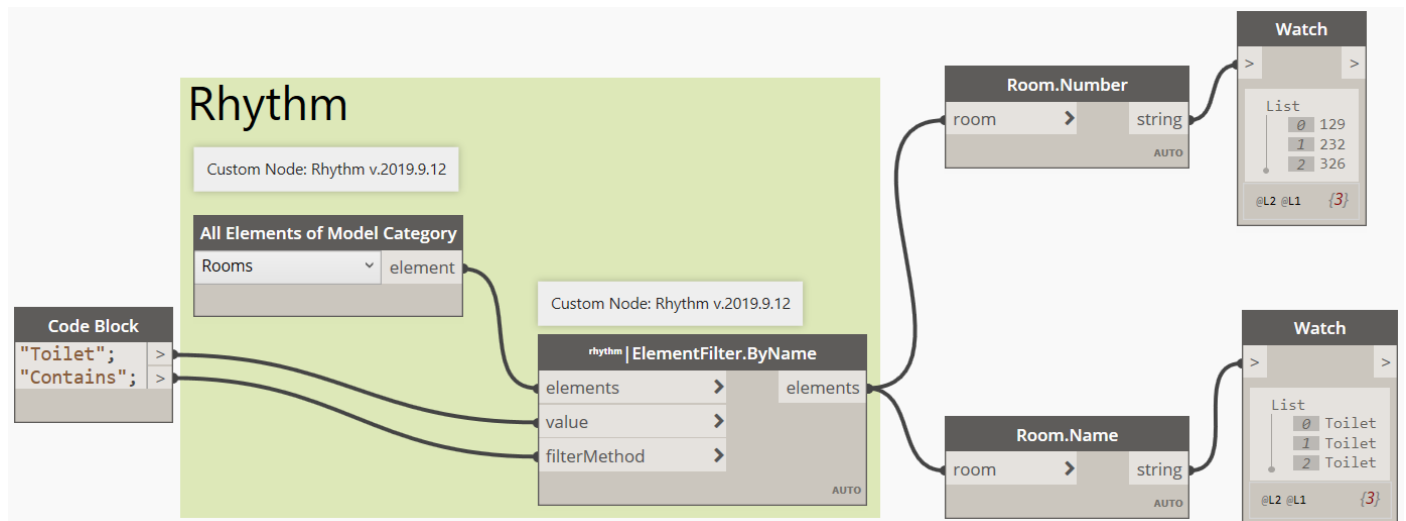
9. Next we will add the **Room.Number** and the **Room.Name** nodes which allow us to see the data we are extracting from the Revit file in a way that is useful. We will also add the three **Watch** nodes at this stage. You can find these nodes by searching for them in the Library, or right clicking on the canvas and search for them in the pop-up search line that appears. You can use CTRL + C and CTRL + V to copy and paste the Watch node once it's on the canvas.



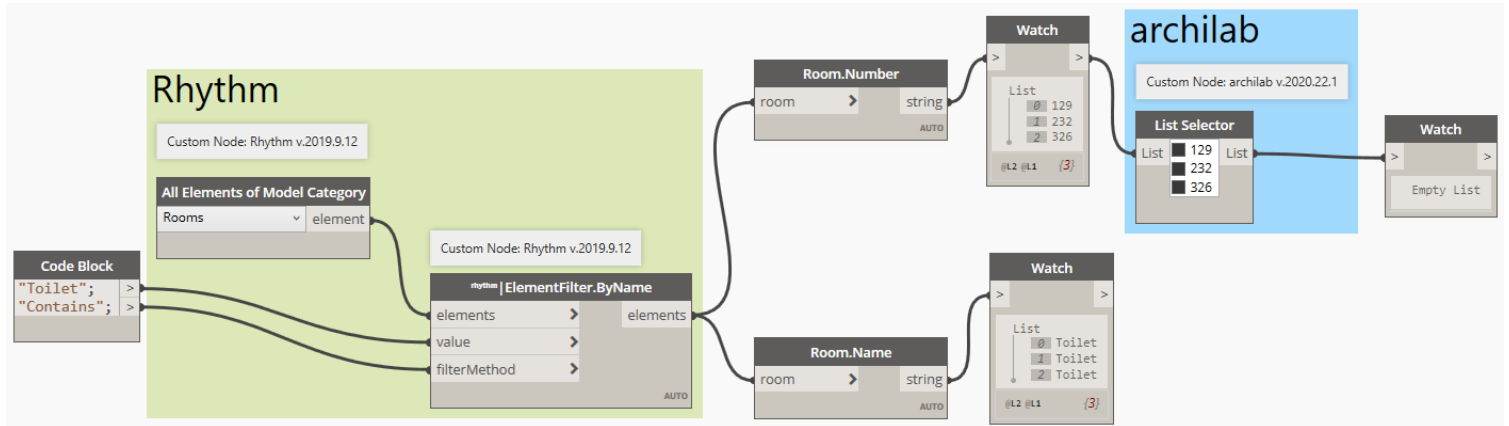
10. Connect the elements output into the input of the **Room.Number** and **Room.Name** nodes. Then connect their outputs into two of the **Watch** nodes.



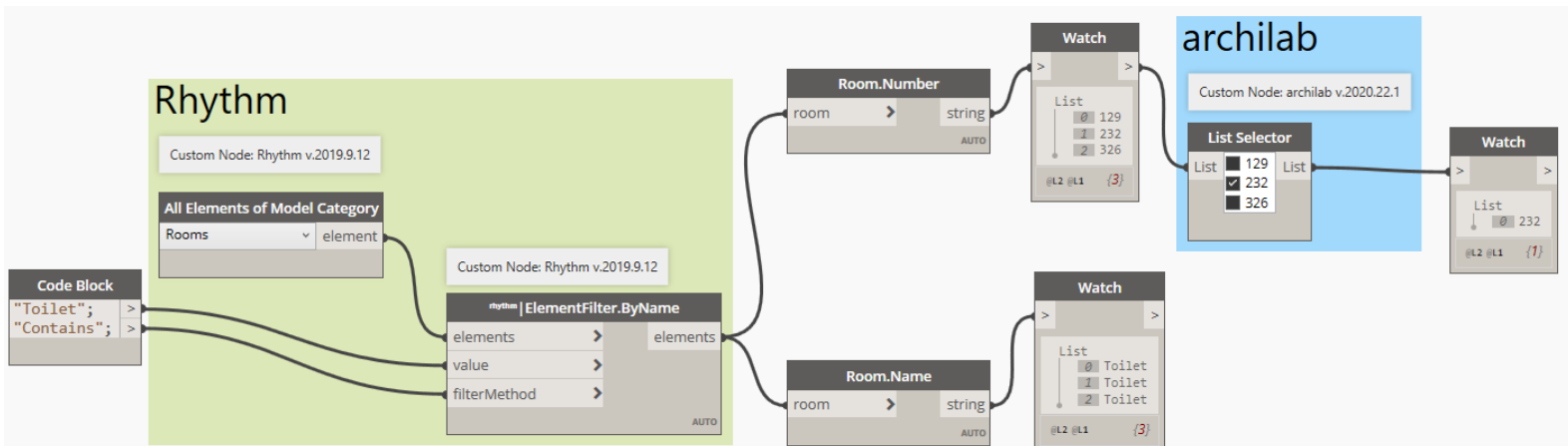
11. Now we can hit the Run button and see which rooms contain the keyword *Toilet* in their name, and what the room name and number are.



12. Now, we will connect the output of the top **Watch** node (one connected to **Room.Number**) to the List input of the **List Selector** node. Then connect the List output of the **List Selector** node to the final **Watch** node's input. Then hit Run, or F5.



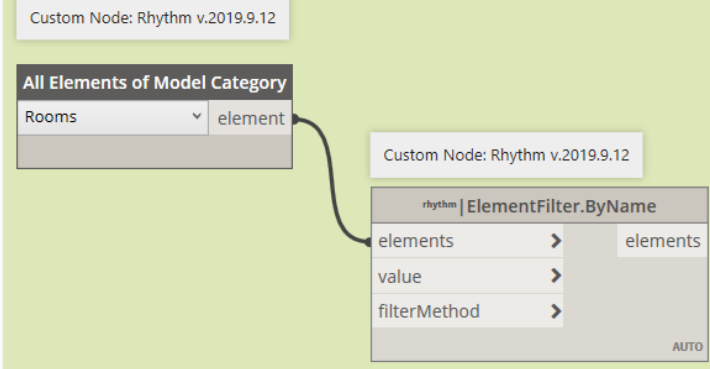
13. You will see that the final **Watch** node displays *Empty List*. This is because we didn't select any of the list items available in the **List Selector** node. If you check the box beside number 232 and hit F5 (or the Run button) you will see the new list inside the final **Watch** node will display 232 now. The **List Selector** node gives you an easy way to choose the data you want out of a given list.



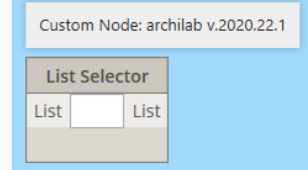
## Task 2

- For this task, you can continue to work with the Revit file and Dynamo graph from task 1, or you can open the **Sample\_Project.rvt** Revit file, and then open the Dynamo file **Ex0\_Data\_T2\_Begin.dyn** file from the **Ex\_0** folder in the lab dataset.
- We will work with the second example in the graph we have open and add a few nodes to be able to filter rooms in the project by a key value (number) based on a parameter, in this case Area. We will be adding eight out of the box nodes: **Code Block**, **Room.Area**, **Room.Name**, **Room.Number** & **Watch** (x4).

## Rhythm

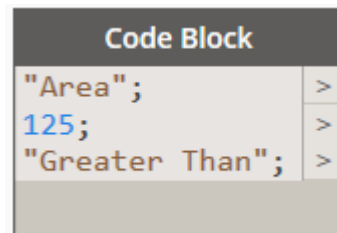


## archilab

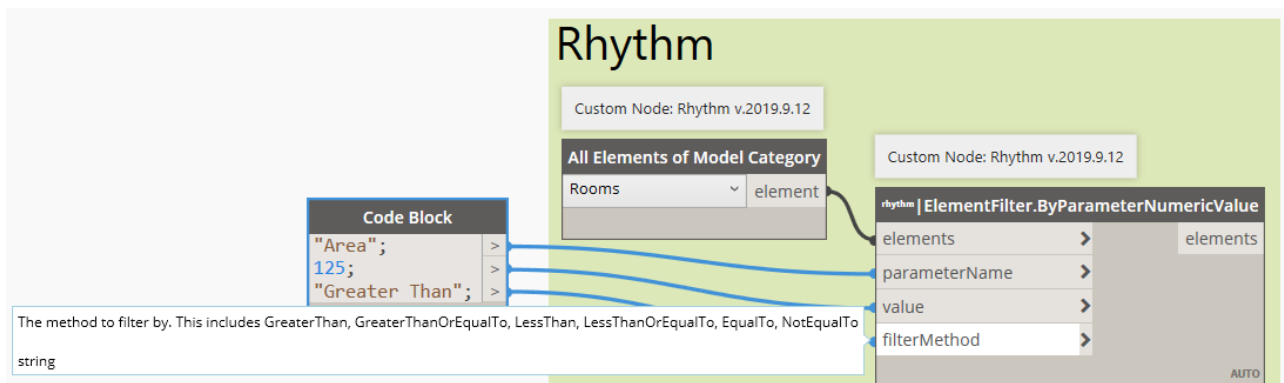


- Confirm that *Rooms* is selected in the **All Elements of Model Category** node.
- Double click to add the **Code Block** to the screen. Click inside the **Code Block** to add the following as shown below.

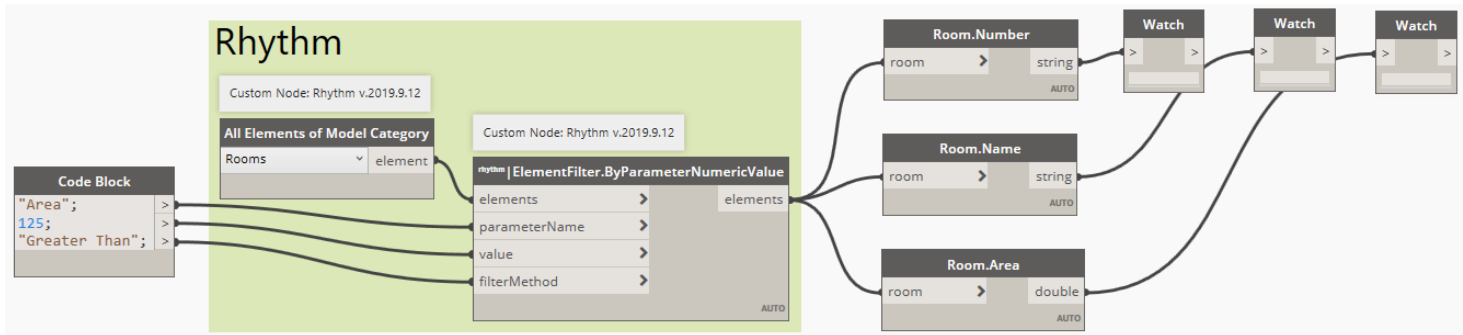
- "Area";
- 125;
- "Greater Than";



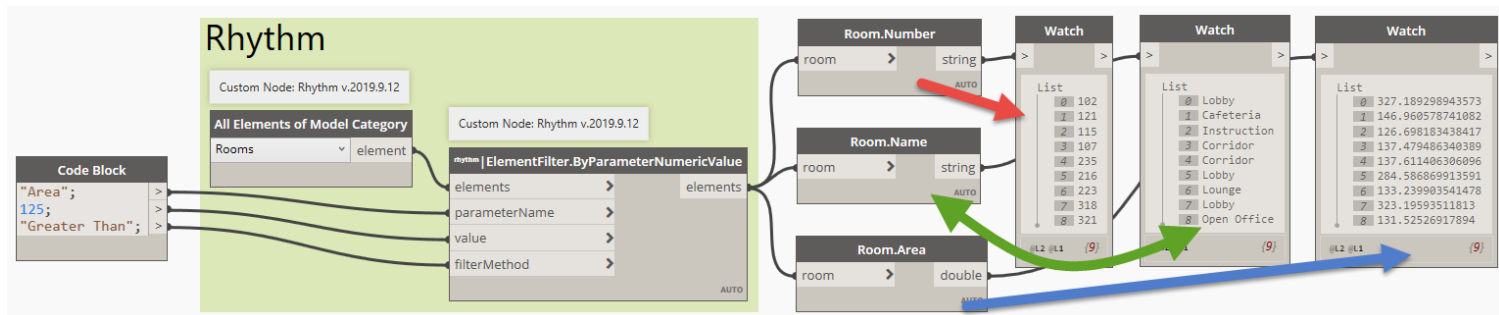
- Next you can connect the wires from the **Code Block** to the **ElementFilter.ByParameterNumericValue** node. "Area" goes into *parameterName* then "125" goes into *value* and "Greater Than" goes into *filterMethod*. **Note:** if you hover over the *filterMethod* input you will get a pop-up that tells you all the different filter methods this input will work with: Greater Than, Greater Than or Equal To, Less Than, Less Than or Equal To, Equal To, Not Equal To.



- Next we add the **Room.Number**, **Room.Name** & **Room.Area** nodes. These nodes allow us to see the data we are extracting from the Revit file in a way that is useful. We also need three **Watch** nodes. Search for these nodes and add them to the canvas.
- Connect the **elements** output into the input of the **Room.Number**, **Room.name** and **Room.Area** nodes. Then connect their outputs to three **Watch** node inputs.

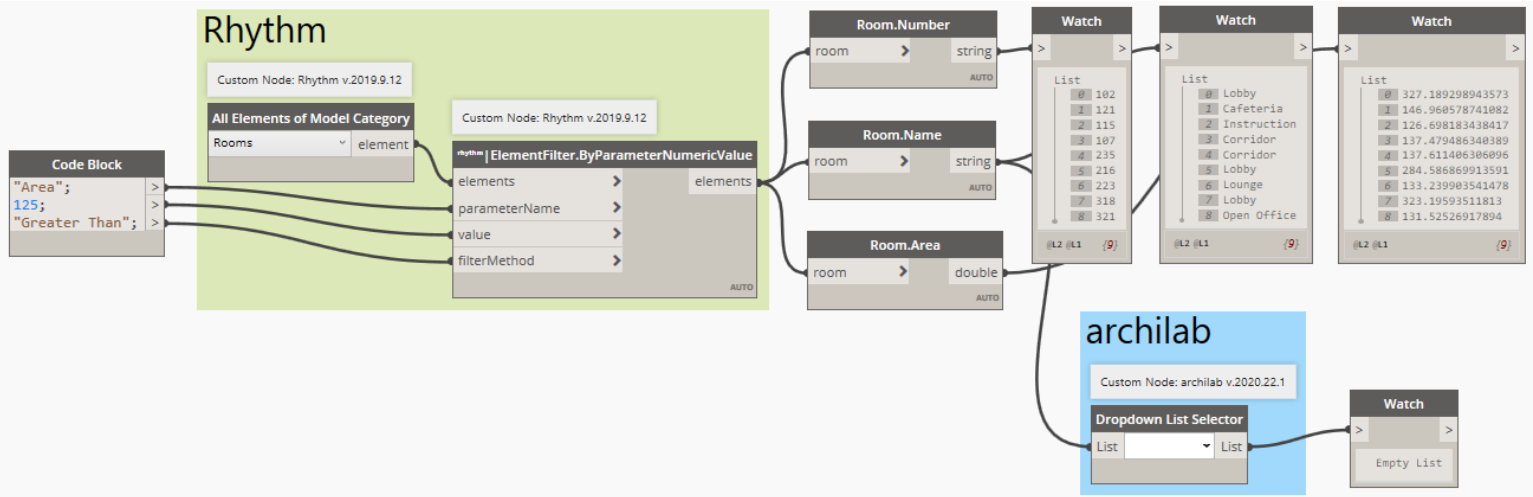


- Now we can hit the Run button and see which rooms are greater than 125 m<sup>2</sup> and what their name, number and area are.

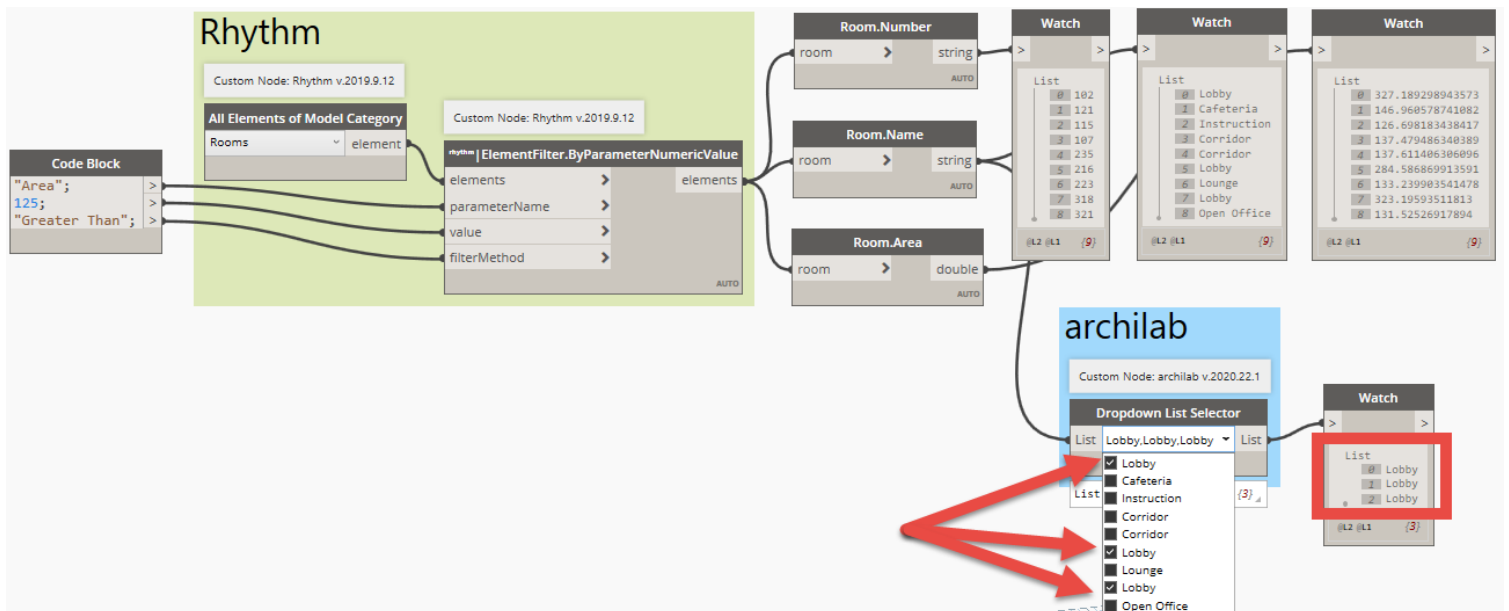


- To finish this task, we will connect the output of the **Room.Name** node to the input of the **Dropdown List Selector** node. Then run a wire from the **Dropdown List Selector** node output to the input of the final **Watch** node, then hit F5 to run the graph.





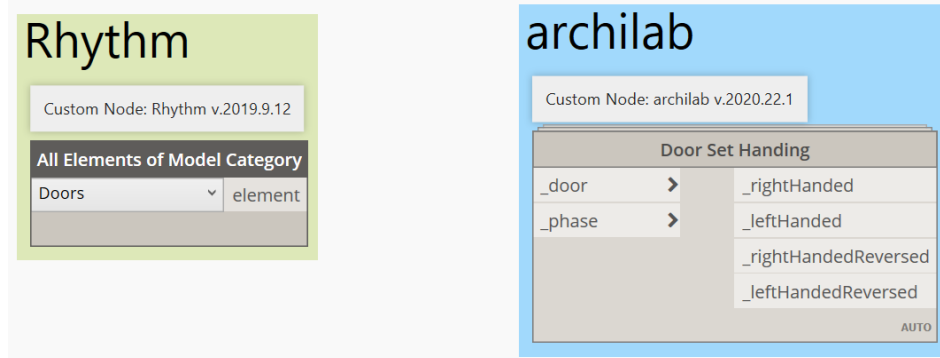
10. After running the graph, it looks like nothing happened. However, if you select the dropdown within the **Dropdown List Selector** node you will see the list of the rooms along with check boxes beside them. Select the three lobby rooms from the dropdown list, these will be included in the new list (which will display in the **Watch** node after you run the graph again).



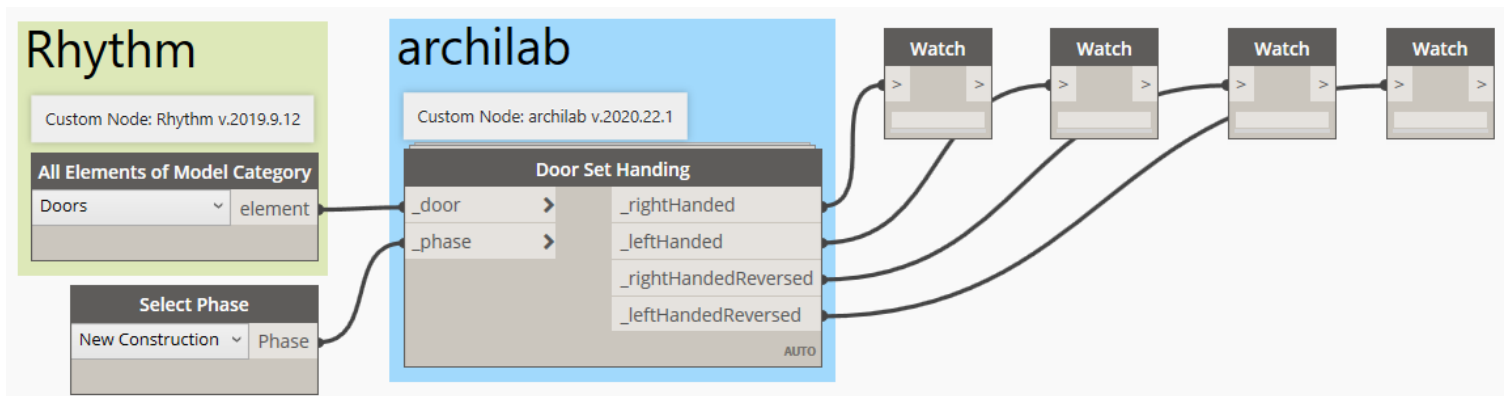
### Task 3

- For this task, you can continue to work with the Revit file and Dynamo graph from task 2, or you can open the **Sample\_Project.rvt** Revit file, and then open the Dynamo file **Ex0\_Data\_T3\_Begin.dyn** file from the **Ex\_0** folder in the lab dataset.

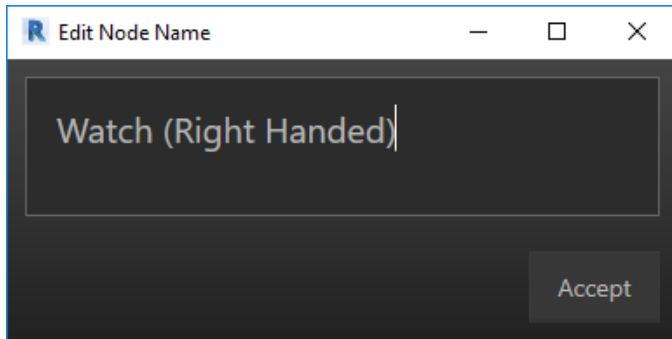
- We will work with the third example in the graph we have open and add a few nodes to be able to filter doors in the project by which handing they are. We will be adding five out of the box nodes: **Select Phase & Watch** (x4).



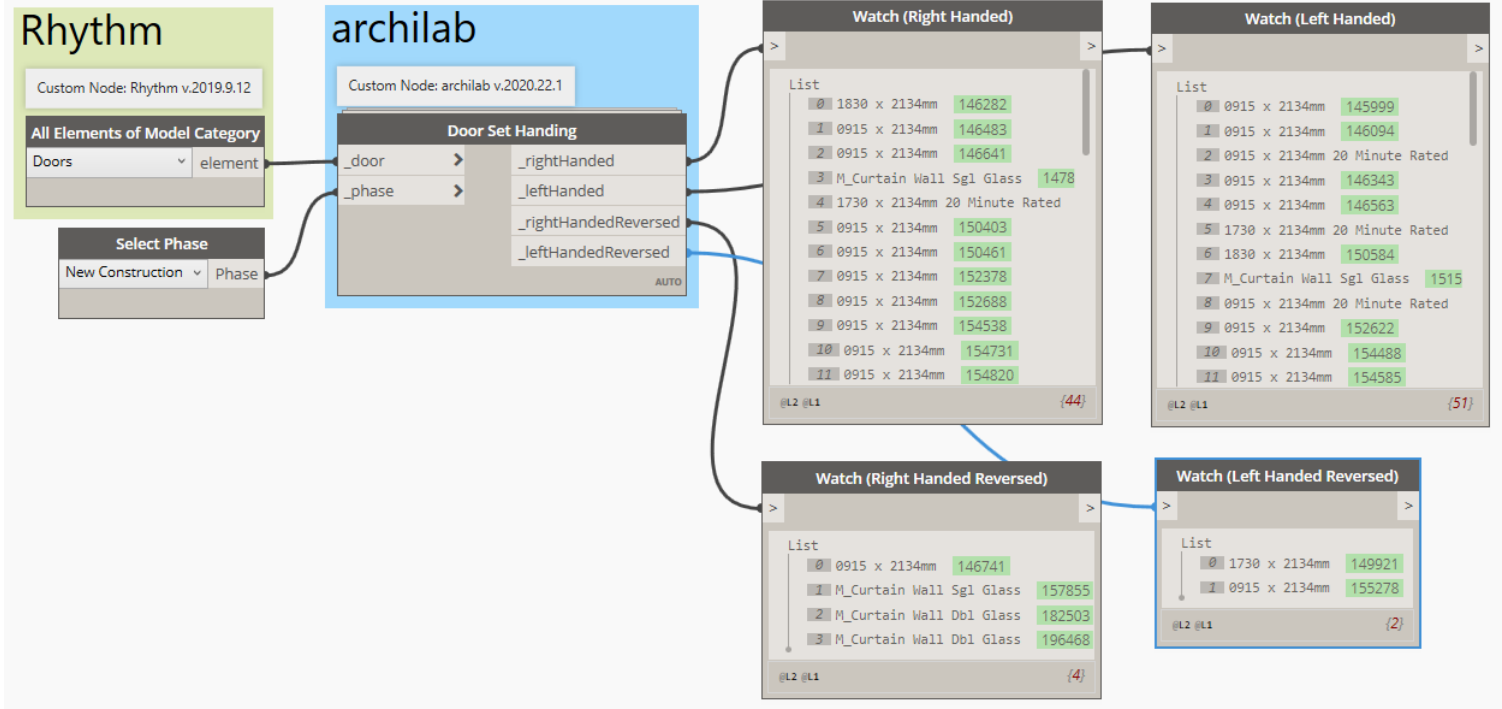
- Confirm that *Doors* is selected in the **All Elements of Model Category** node.
- Now we can add the needed nodes before we hit Run which are the **Select Phase** node, and four **Watch** nodes. Search for these nodes and add them to the canvas.
- Confirm that *New Construction* is selected in the **Select Phase** node.
- Connect the *Phase* output into the *Phase* input of the **Door Set Handling** node. Then connect the four outputs of the **Door Set Handling** node to the four **Watch** nodes.



- Quick tip, you can double click on any node and change or update its name. This can be helpful to let those using the graph that didn't create it understand the outcomes better. Let's add the name of the output which is being connected to each **Watch** node.



8. Now we can hit the Run button and see the handing of all the doors in our Revit project.

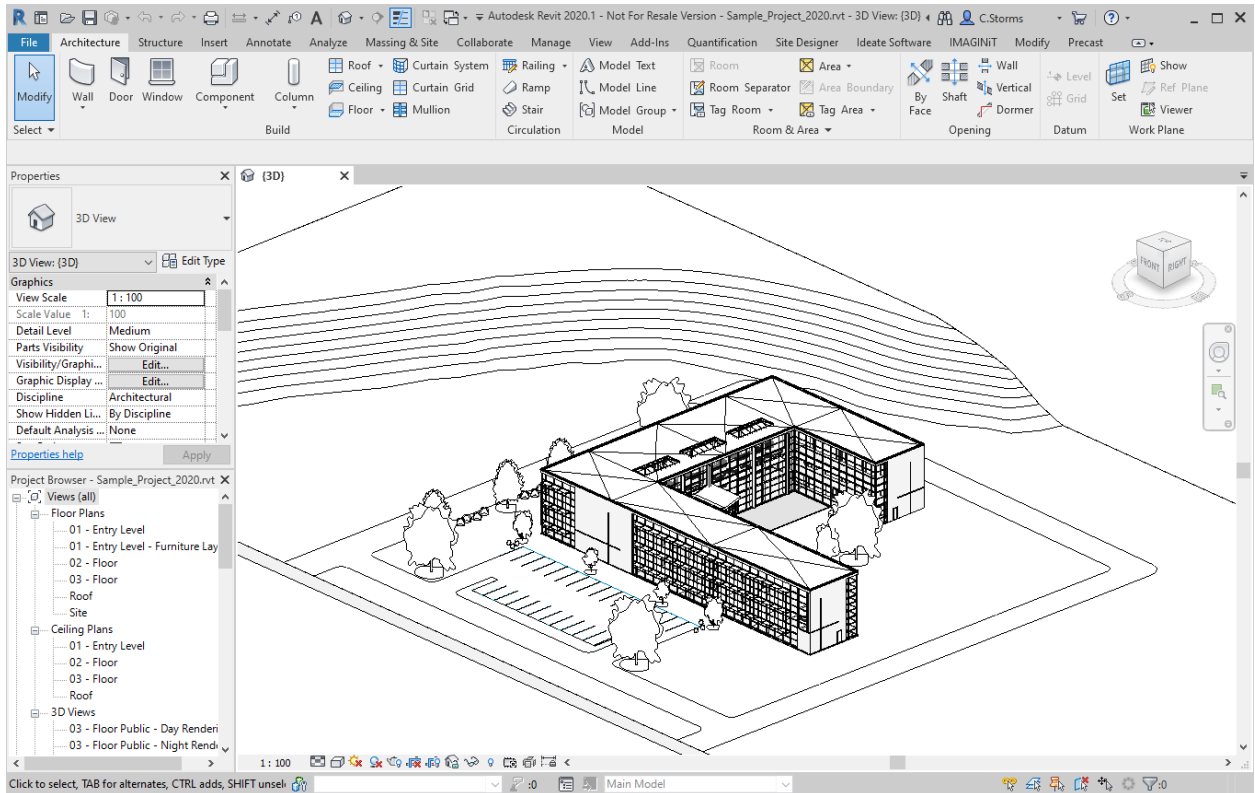


## Ex 1) Warnings

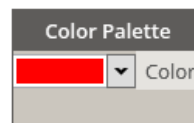
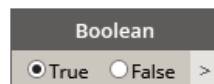
In this exercise we will use the **Bang** package from [John Pierson](#) to visualize warnings inside Revit using colours.

### Task 1

1. From the dataset folder **Ex\_1** open the following file: **Sample\_Project.rvt** in Revit.



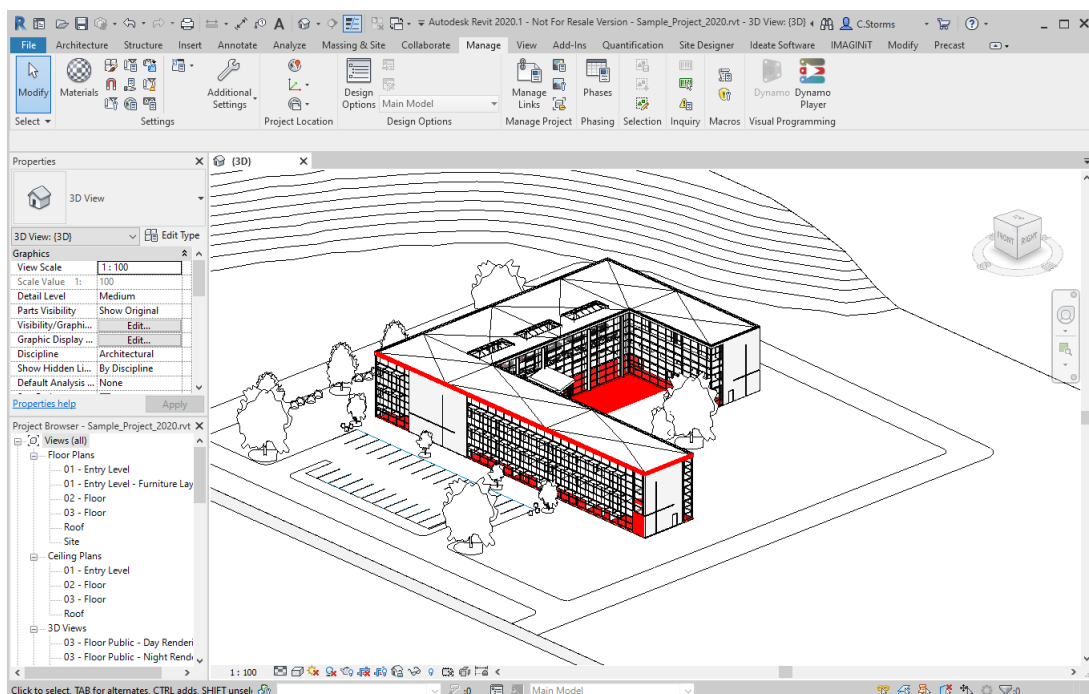
2. Start Dynamo from the Manage tab.
3. From the Start Page, under the FILES section click Open. Then navigate to the class dataset folder **Ex\_1** open the following file: **Ex1\_Warnings\_T1\_Begin.dyn**.



- We are starting with two nodes: **Boolean** and **Color Palette** (set to red). We only need to add two more nodes to make the graph complete. Search for and add the following nodes to the canvas: **WarningTools.GetWarnings** from the **Bang** package and OOTB node **Element.OverrideColorInView**. Once all the nodes are on the canvas connect them as shown below.

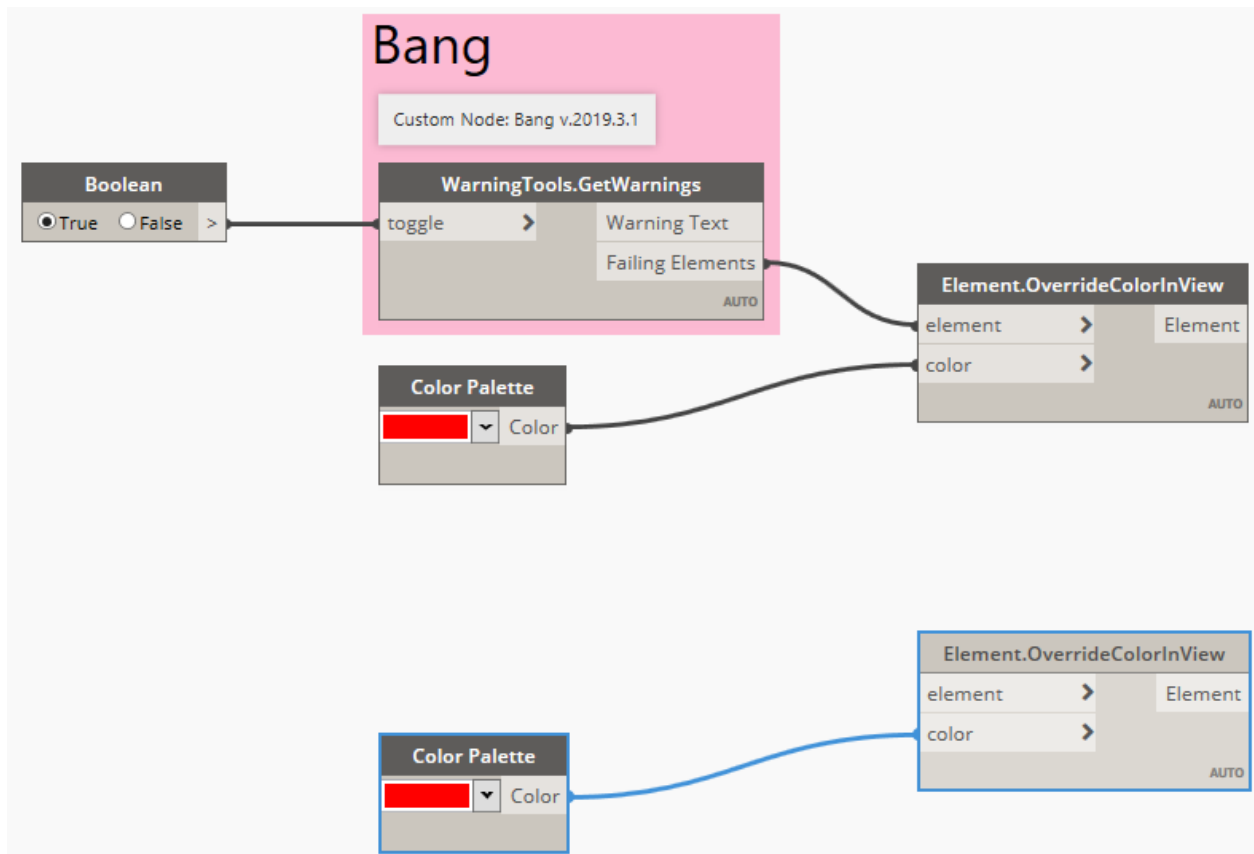


- Now we can run the graph by hitting F5 and see what warnings are now showing in red in the current Revit view?



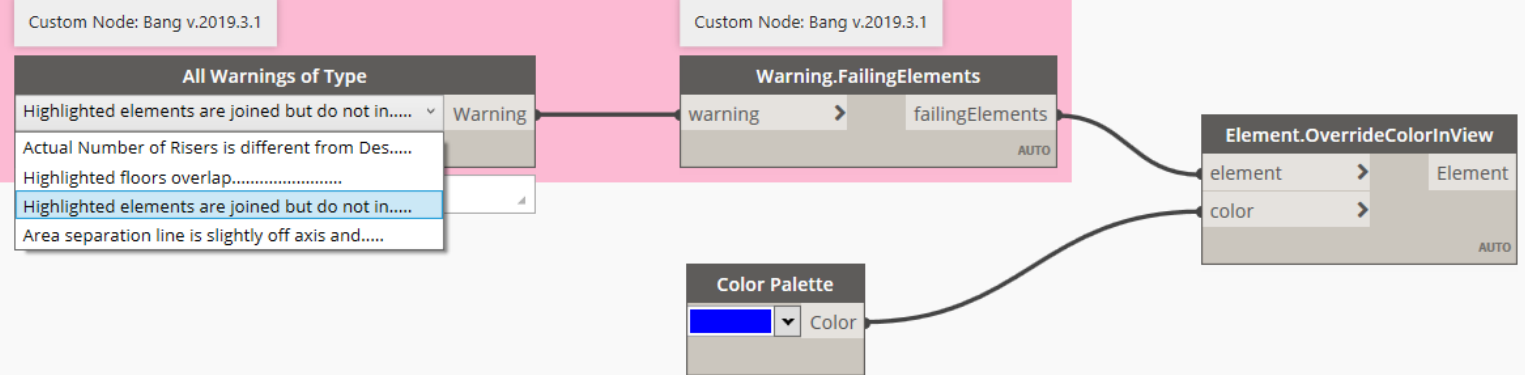
## Task 2

1. For this task, you can continue to work with the Revit file and Dynamo graph from task 1, or you can open the **Sample\_Project.rvt** Revit file, and then open the Dynamo file **Ex1\_Warnings\_T2\_Begin.dyn** file from the **Ex\_1** folder in the lab dataset.
2. We will start this task by selecting and copying the **Color Palette** and **Element.OverrideColorInView** nodes to a location lower on the canvas.

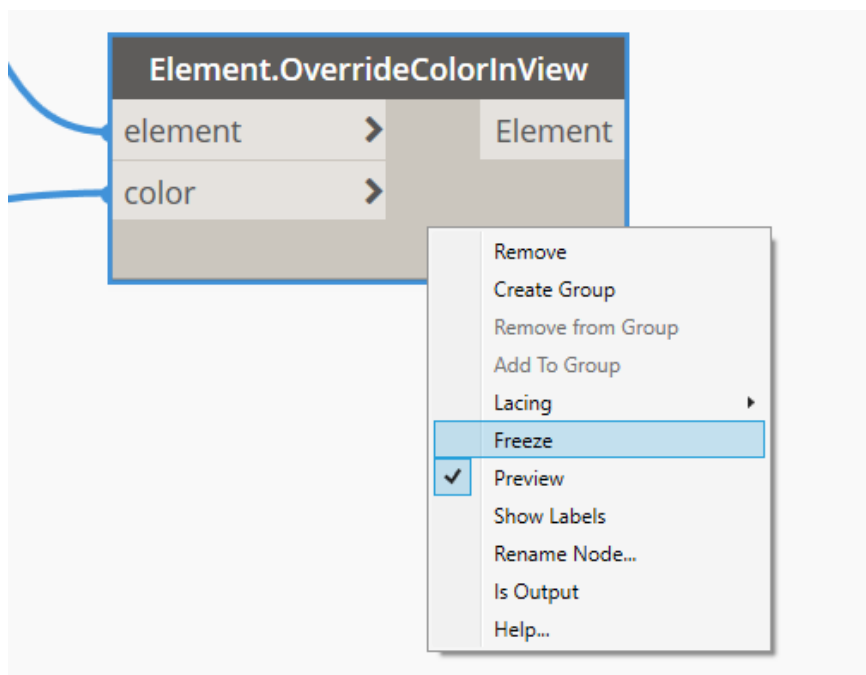


3. Before we add more nodes let's change the colour in the newly copied **Color Palette** node to **blue**.
4. Now we can add two new nodes from the **Bang** package: **All Warnings of Type** and **Warning.FailingElements**. The **All Warnings of Type** node allows us to filter the warnings we want to display in **blue** by the warning type available in the node dropdown.

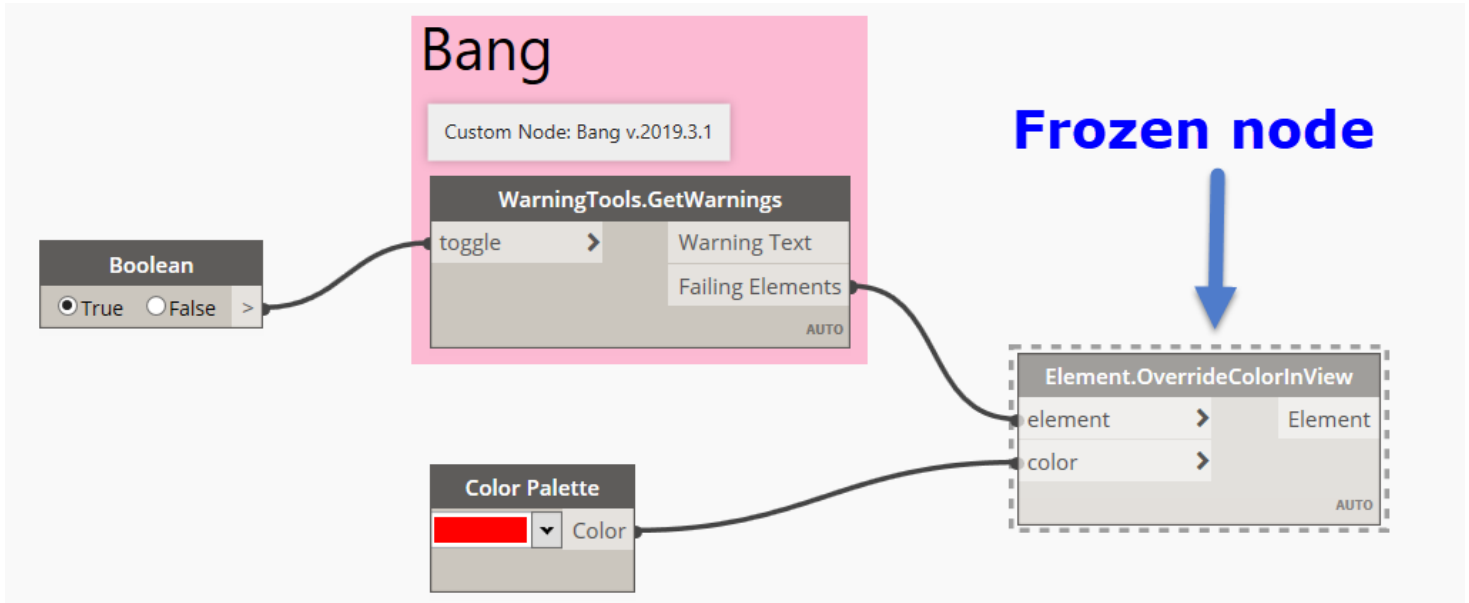
## Bang



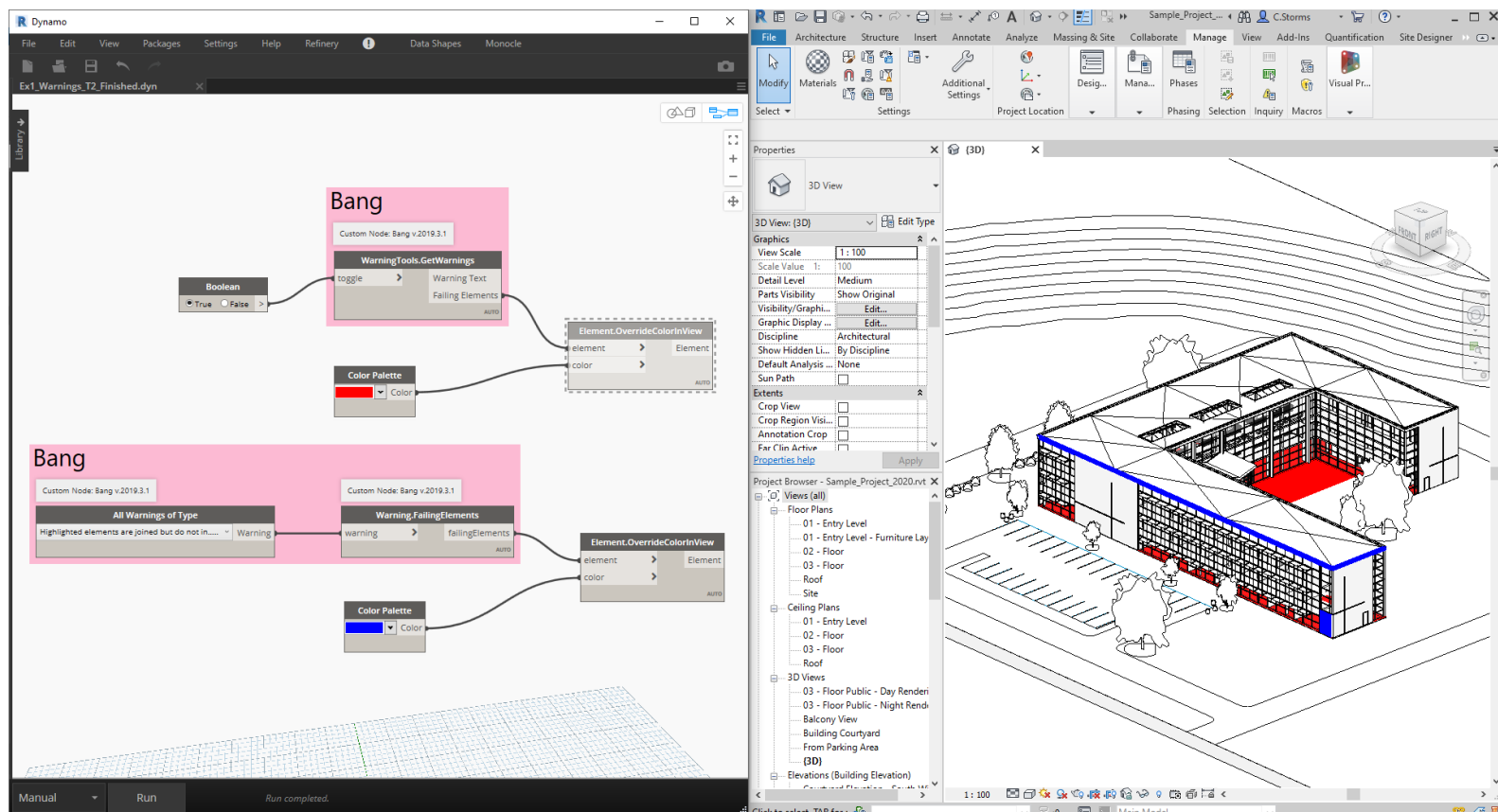
5. Make sure that *Highlighted elements are joined but do not in .....* is selected in the **All Warnings of Type** dropdown.
6. Before we run this graph, we want to first freeze the **Element.OverrideColorInView** node from the task 1 portion of the graph. You can do this by right clicking on the node and selecting *Freeze* from the menu.



7. After the node has been frozen it will look like the following image.



8. Now when we hit run, we will only see the effects of the new portion of the graph, which will show only the filtered warning we selected in **blue** in the Revit model.



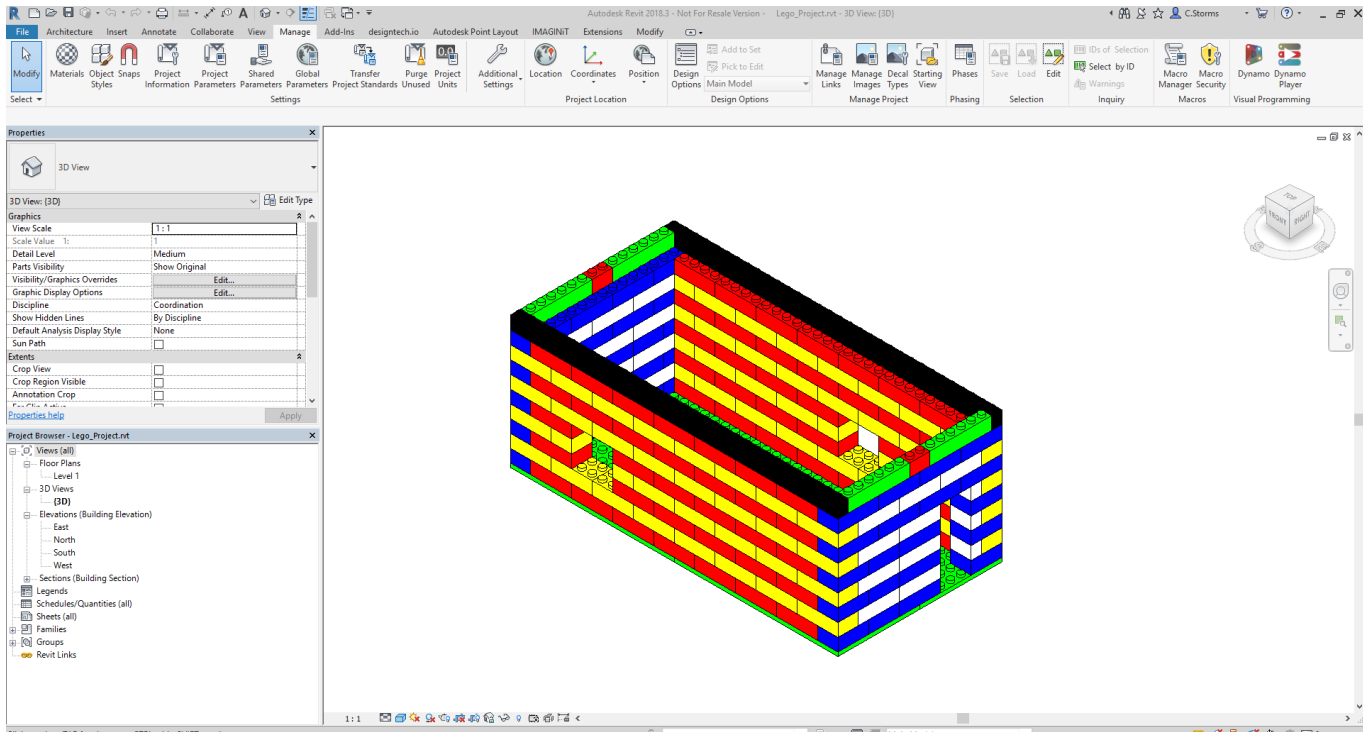


## Ex 2) Count and Colour

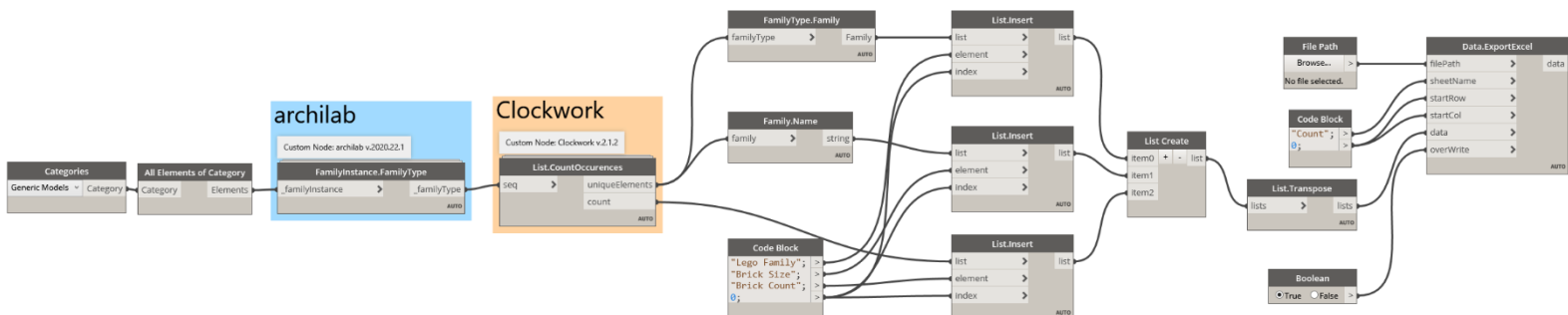
In this exercise we will extract some data from our Revit project using List@Level and group by key sorting so we can export it to Excel in the format we want.

### Task 1

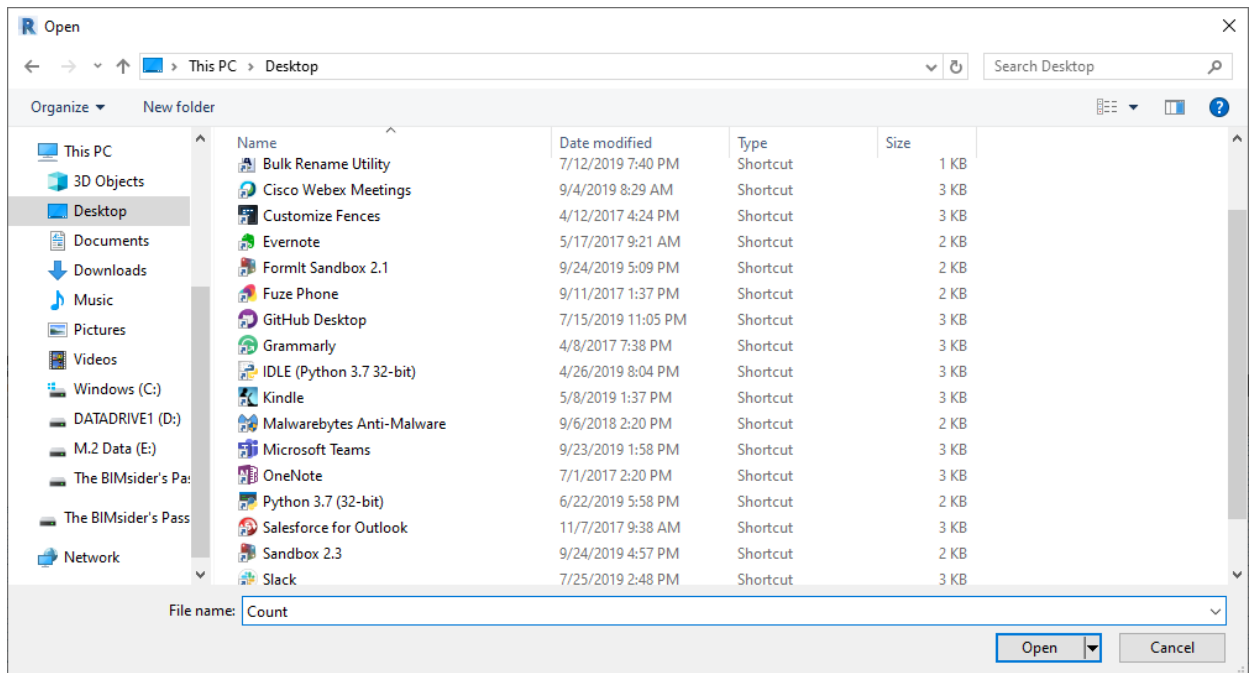
1. From the dataset folder **Ex\_2** open the following file: **Lego\_Project.rvt** in Revit.



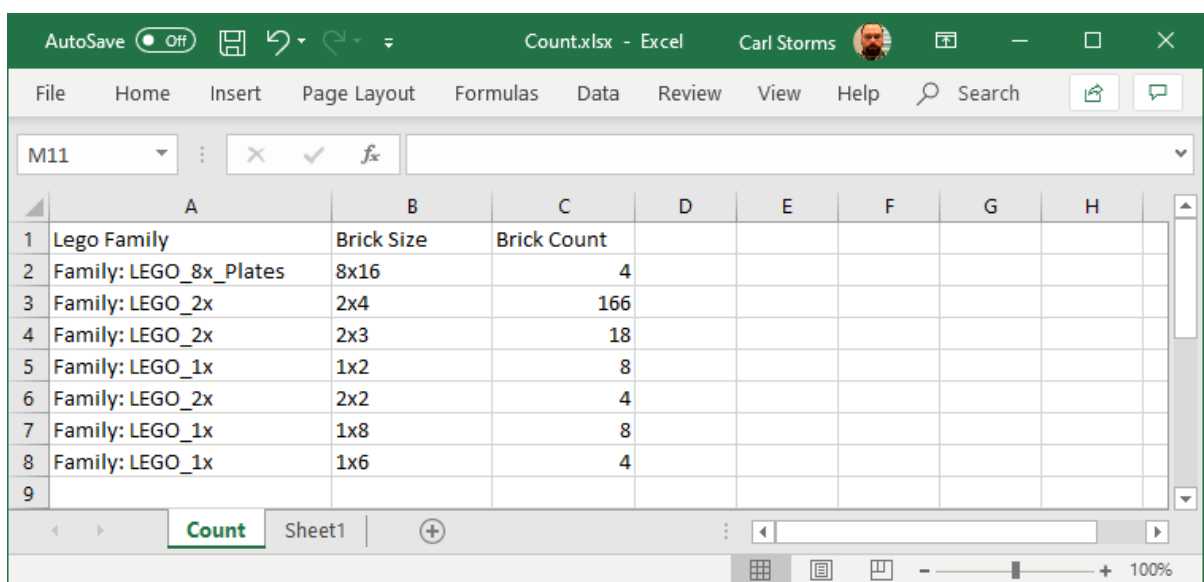
2. Start Dynamo from the Manage tab.
3. Now Open the Dynamo graph called **Ex2\_Colour\_and\_Count\_T1\_Begin.dyn** from the dataset folder **Ex\_2**.



- This graph allows for the collection of three pieces of information from the Revit project and export it to Excel. Before trying the graph make sure the **File Path** node is pointing to the correct location on the computer by selecting **Browse...** For this exercise we want to save the file to the computer desktop. You can also name the Excel file once you select the save location, we will call this file **Count**.



- Hit the Run button to see how it works, and how the information looks in in Excel.



The screenshot shows an Excel spreadsheet titled 'Count.xlsx'. The spreadsheet has a table with 8 rows and 4 columns. The columns are labeled 'Lego Family', 'Brick Size', and 'Brick Count'. The data is as follows:

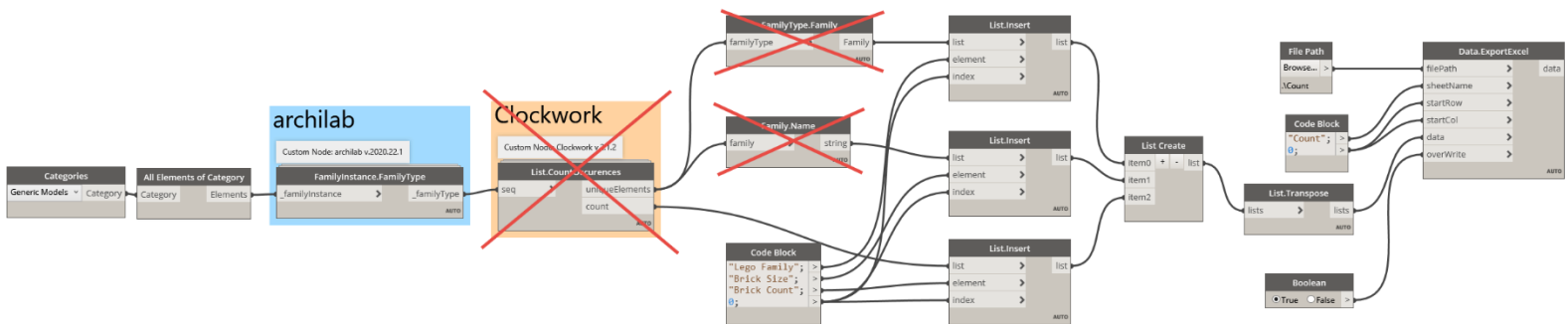
	A	B	C	D	E	F	G	H
1	Lego Family	Brick Size	Brick Count					
2	Family: LEGO_8x_Plates	8x16	4					
3	Family: LEGO_2x	2x4	166					
4	Family: LEGO_2x	2x3	18					
5	Family: LEGO_1x	1x2	8					
6	Family: LEGO_2x	2x2	4					
7	Family: LEGO_1x	1x8	8					
8	Family: LEGO_1x	1x6	4					
9								

The spreadsheet is titled 'Count' and is on 'Sheet1'. The status bar at the bottom shows '100%' zoom.

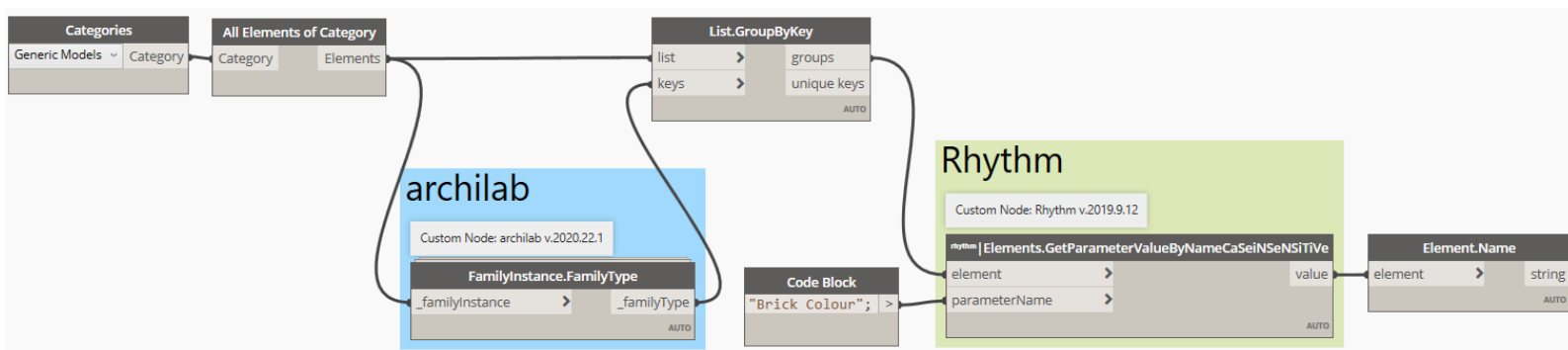
## Task 2

In this task we will update the graph from task 1 to get the information we want.

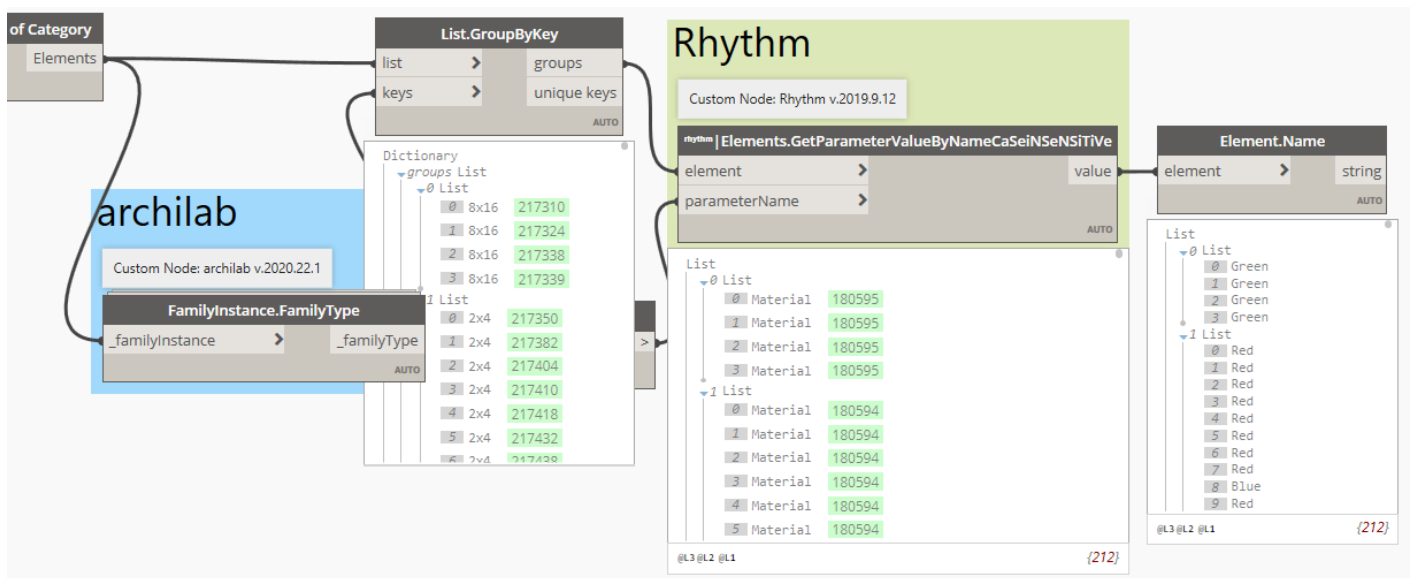
1. For this task, you can continue to work with the Revit file and Dynamo graph from task 1, or you can open the **Lego\_Project.rvt** Revit file, and then open the Dynamo file **Ex2\_Colour\_and\_Count\_T2\_Begin.dyn** file from the **Ex\_2** folder in the lab dataset.
2. The info from the task 1 graph was helpful information, but what we really need to know is how many bricks there are of each size, and what colour those bricks are. To get this information we can start with what we had in task 1 but will need to make some changes. Let's start by deleting three nodes we won't need.



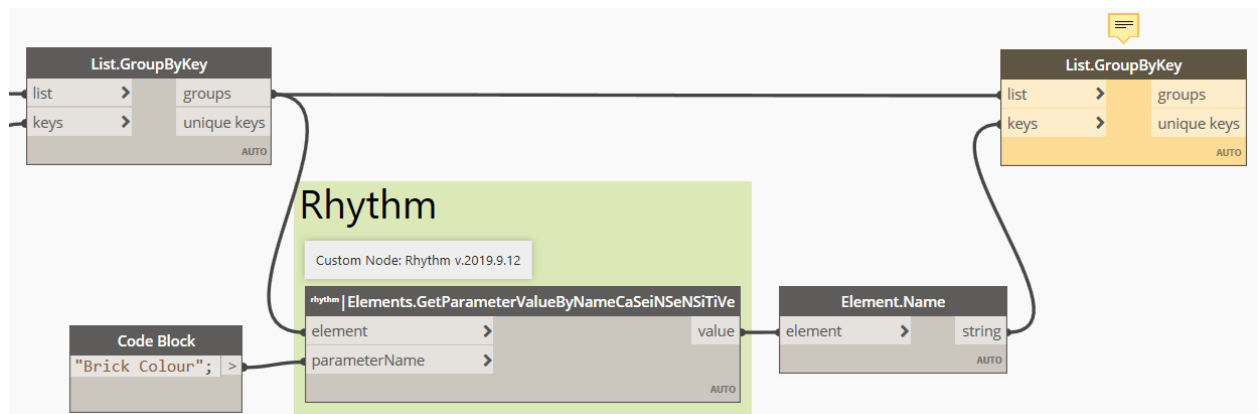
3. To get the information we want, brick colour and quantity per brick size, we will now have to add some new nodes. Search for the **Element.Name**, **List.GroupByKey** and **Element.GetParameterValueByNameCaSeiNsEnSiTiVe** nodes, the latter is a custom node from the **Rhythm** package. We will also need another **Code Block** added to the canvas with "Brick Colour" add to it. Once all the new nodes are on the canvas connect them like shown in the following image.



4. The **List.GroupByKey** node allows us to sort the 212 bricks by a key, which in this case is the size (which is also the family type in this example). We then get the *Brick Colour* parameter information of each brick, and its name (the material name, which in this example is also the colour).

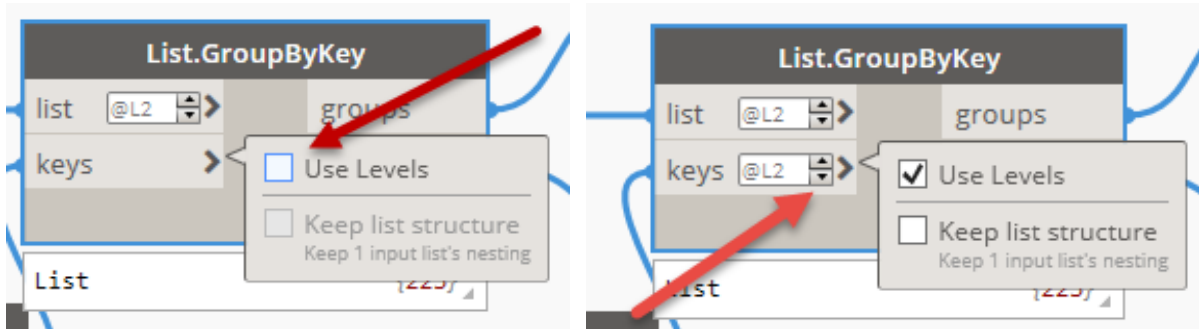


5. Now we will use another **List.GroupByKey** node to group this information together in a single list. However, when we add the node and connect it, we get a yellow error/warning?

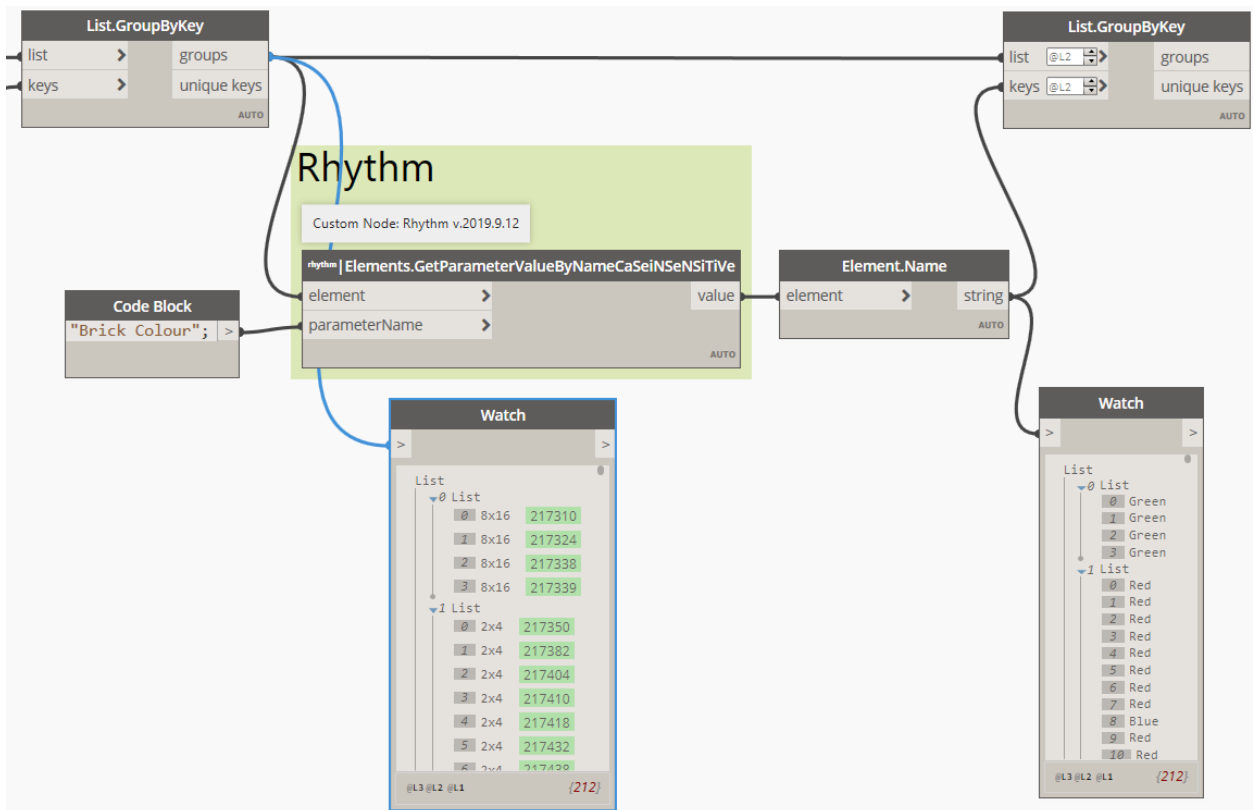


6. This is because we are not asking the **List.GroupByKey** node to look at the right level of the lists. This is where the **List@Level** function comes into play. We can fix this by setting the **List@Level** to **@L2** for both the *list* and *keys* input. To do this click on the **>** icon by the inputs and check the **Use Levels** box. Then using the up and down arrows adjust the level until it says **@L2** (which should be the default).

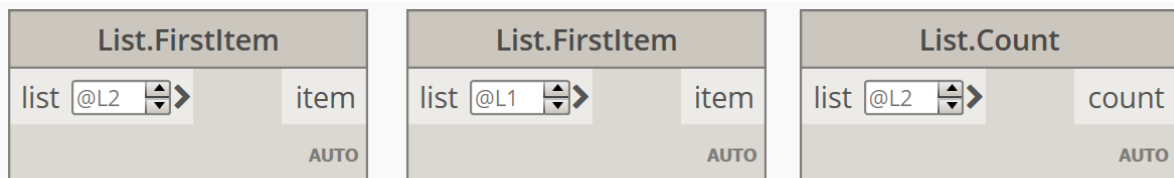
**Note:** To learn more about List@Level review the information and links in the resource section of this handout.



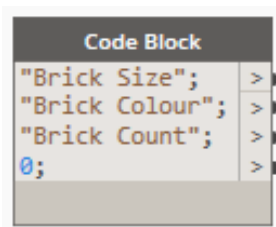
- Now we can see the error goes away, and we can see that the level 2 information with the brick size, and the brick colour.



- To complete this graph we need three more nodes, **List.FirstItem(x2)** and **List.Count**. We need to set the Levels of these nodes as shown in the following image.



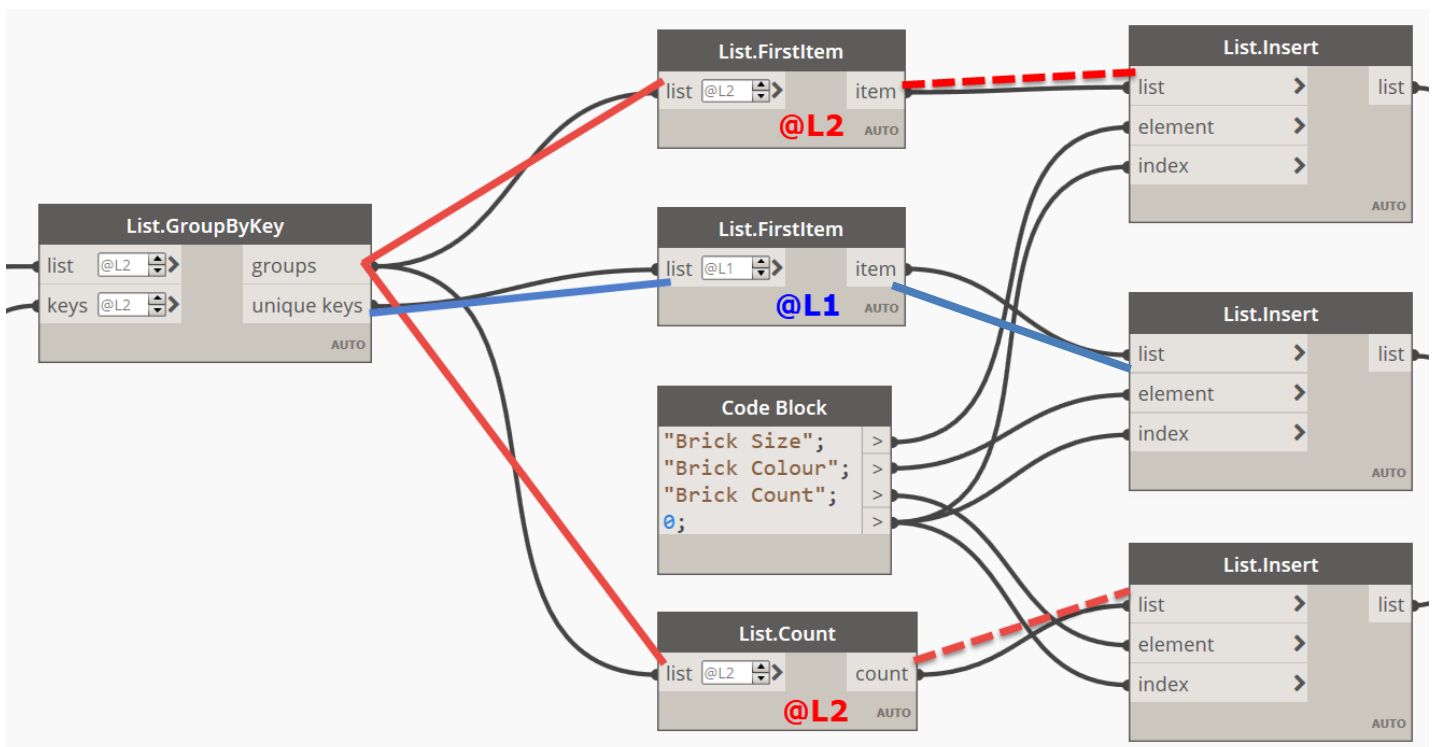
9. These nodes will help us connect this new part of the graph to the old Excel part. The existing Excel part of the graph will almost work as is. You will need to change the **Code Block** a little and confirm that the **File Path** node is pointing to your desktop. The updated **Code Block** should look like the one following, you can also double click and create a new **Code Block** if you prefer. The **element** inputs for all 3 **Insert** nodes are in order from top to bottom in the **Code Block**, and the 0 is the **index** input for all 3 **Insert** nodes.



```

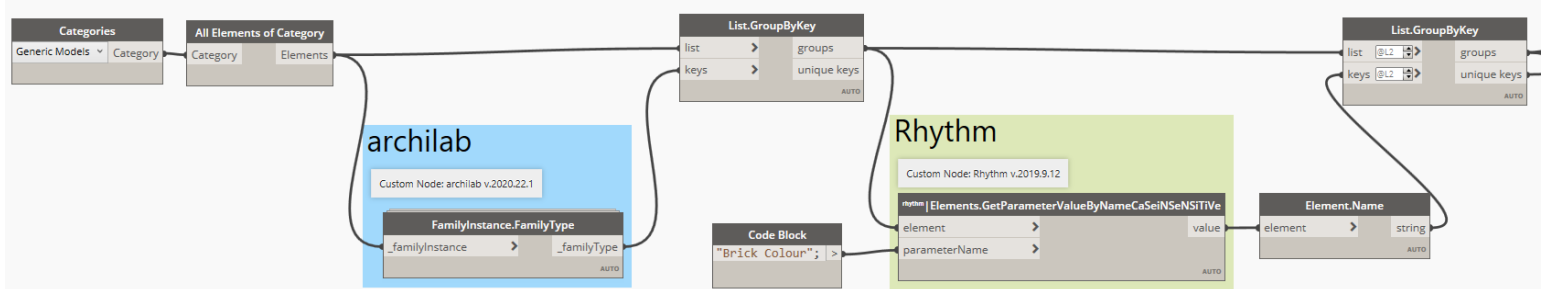
Code Block
"Brick Size"; >
"Brick Colour"; >
"Brick Count"; >
0; >
  
```

10. When connected, the three nodes and updated **Code Block** should look like this:

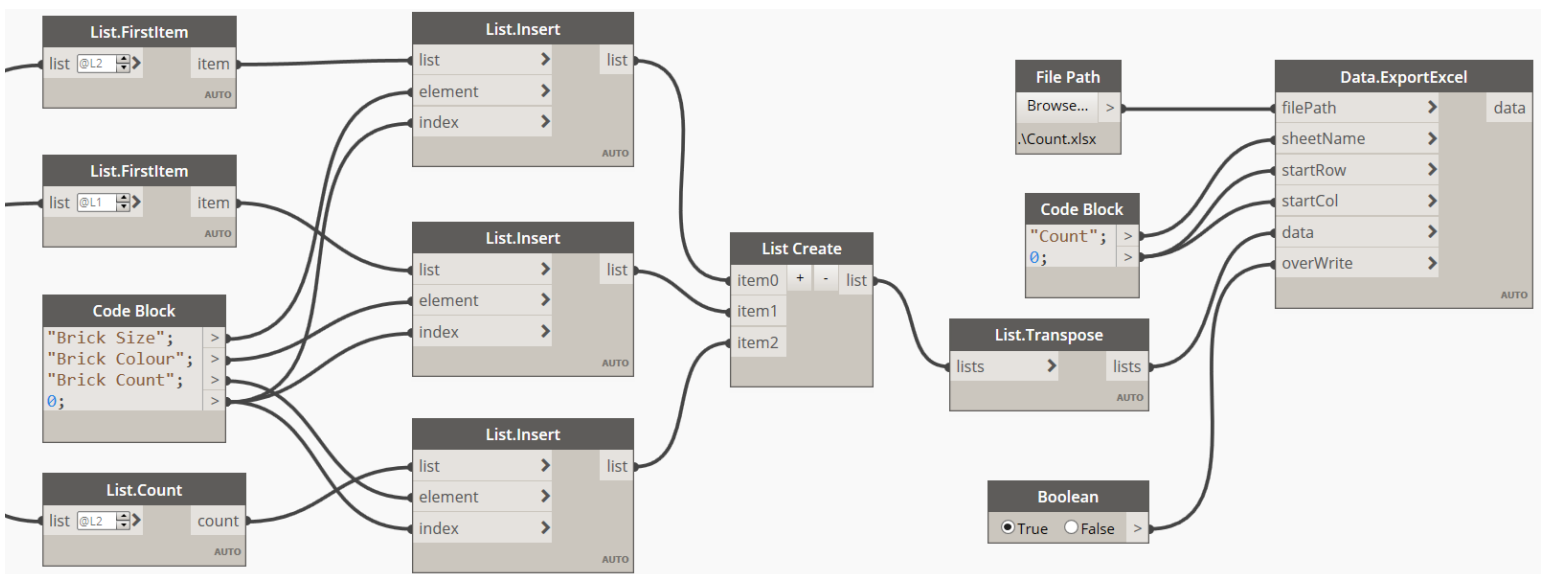


11. The complete graph should now look something like the images that follow.







### Part 1 of the graph









### Part 2 of the graph




12. Hit the Run button and see what information we now get inside of Excel.




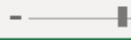
AutoSave ☐ Off    Count.xlsx -... Carl Storms   

File Home Insert Page Layout Formulas Data Review View Help  Search  

H17   

	A	B	C	D	E	F	G	H
1	Brick Size	Brick Colour	Brick Count					
2	8x16	Green	4					
3	2x4	Red	66					
4	2x4	Blue	22					
5	2x4	Yellow	60					
6	2x4	White	18					
7	2x3	Blue	18					
8	1x2	White	6					
9	1x2	Red	2					
10	2x2	Yellow	4					
11	1x8	Black	8					
12	1x6	Green	4					
13								

Count Sheet1 

   -  + 100%

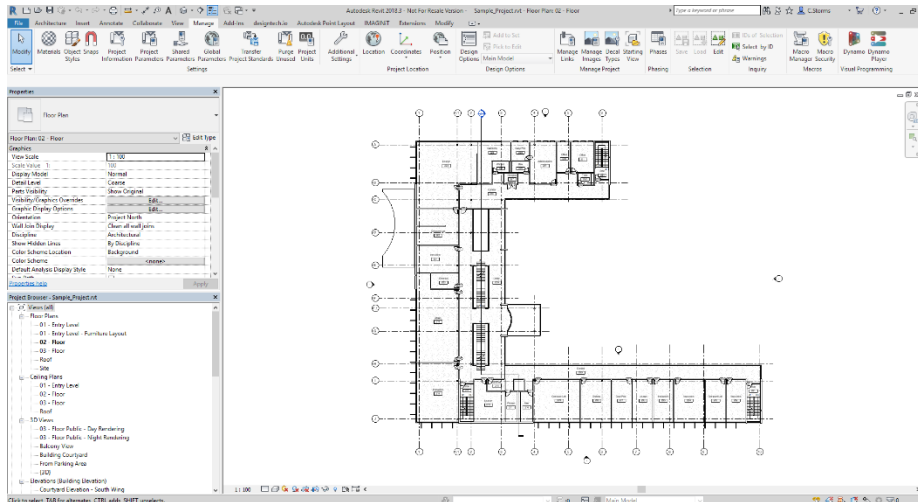


## Ex 3) Doors

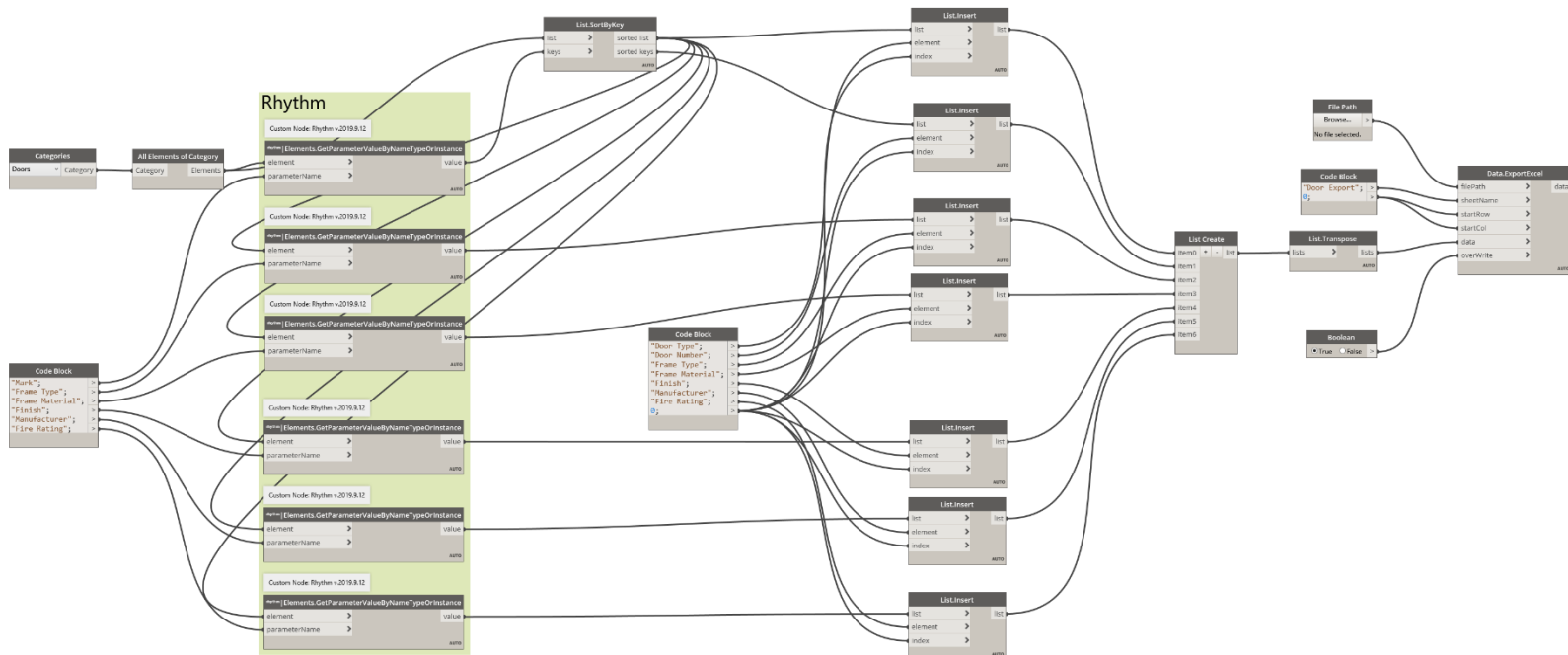
In this exercise we will bring data into our Revit project from an Excel file.

### Task 1

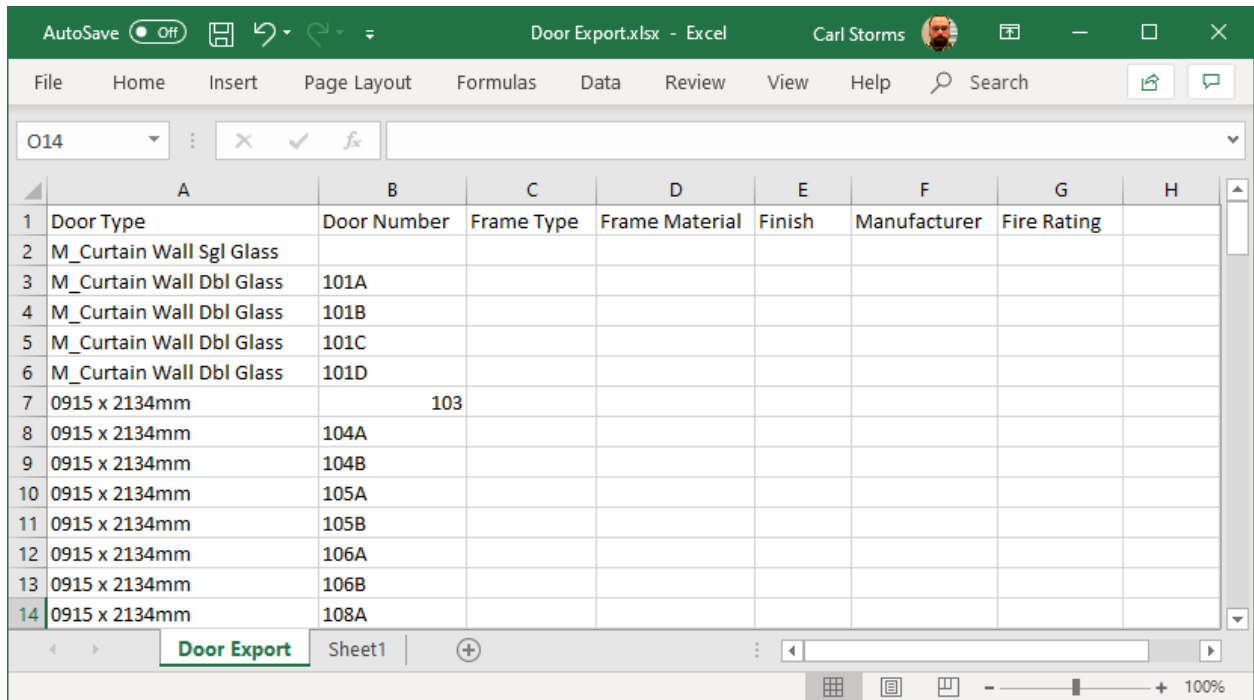
1. From the dataset folder **Ex\_3** open the following file: **Sample\_Project.rvt** in Revit.



2. Start Dynamo from the Manage tab.
3. Now Open the Dynamo graph called **Ex3\_Doors\_T1\_Begin.dyn** from the dataset folder **Ex\_3**.



4. As we have already learned how to export information from Revit to Excel, the purpose of this graph is to allow us to extract door Information that we can edit in Excel then import back into our Revit project. Before trying the graph make sure the **File Path** node is pointing to the correct location on the computer by selecting **Browse...** We want it pointing to the computer desktop.
5. Now hit the Run button and review the resulting Excel data.



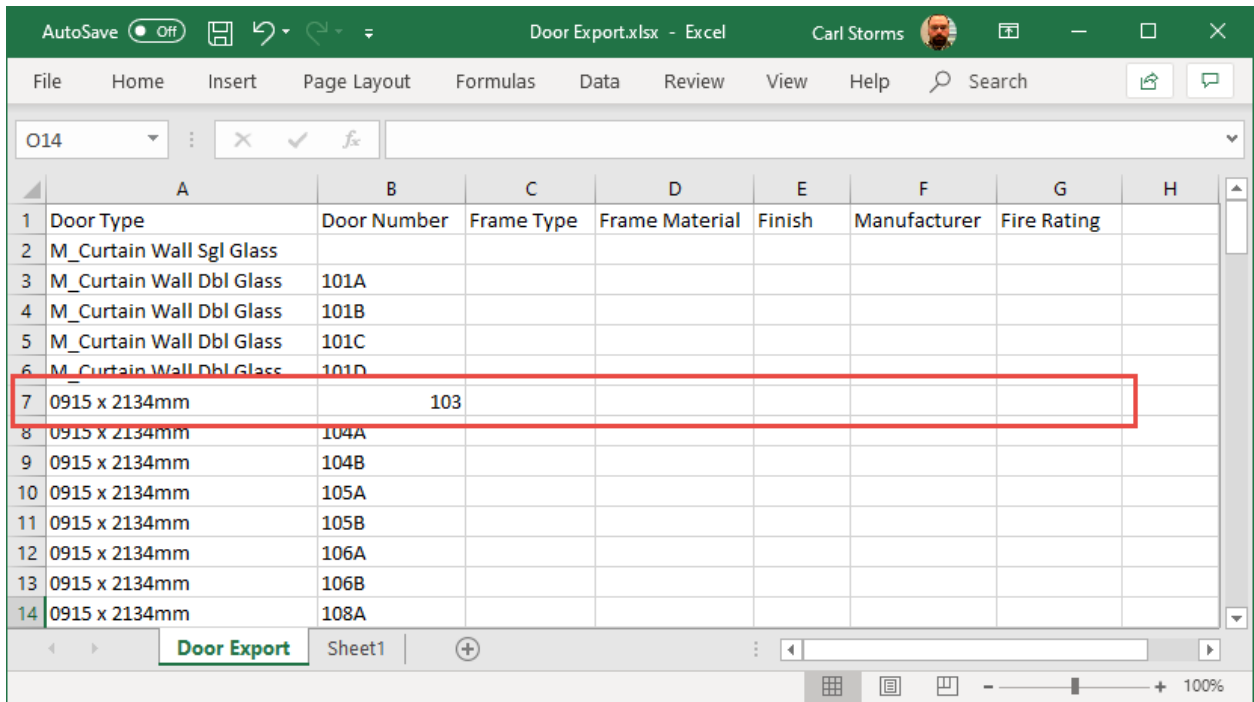
	A	B	C	D	E	F	G	H
1	Door Type	Door Number	Frame Type	Frame Material	Finish	Manufacturer	Fire Rating	
2	M_Curtain Wall Sgl Glass							
3	M_Curtain Wall Dbl Glass	101A						
4	M_Curtain Wall Dbl Glass	101B						
5	M_Curtain Wall Dbl Glass	101C						
6	M_Curtain Wall Dbl Glass	101D						
7	0915 x 2134mm	103						
8	0915 x 2134mm	104A						
9	0915 x 2134mm	104B						
10	0915 x 2134mm	105A						
11	0915 x 2134mm	105B						
12	0915 x 2134mm	106A						
13	0915 x 2134mm	106B						
14	0915 x 2134mm	108A						

6. You will notice that there are empty cells in the Excel file, this is the information we will add in Excel, then import back into our Revit Project in the next Task.

## Task 2

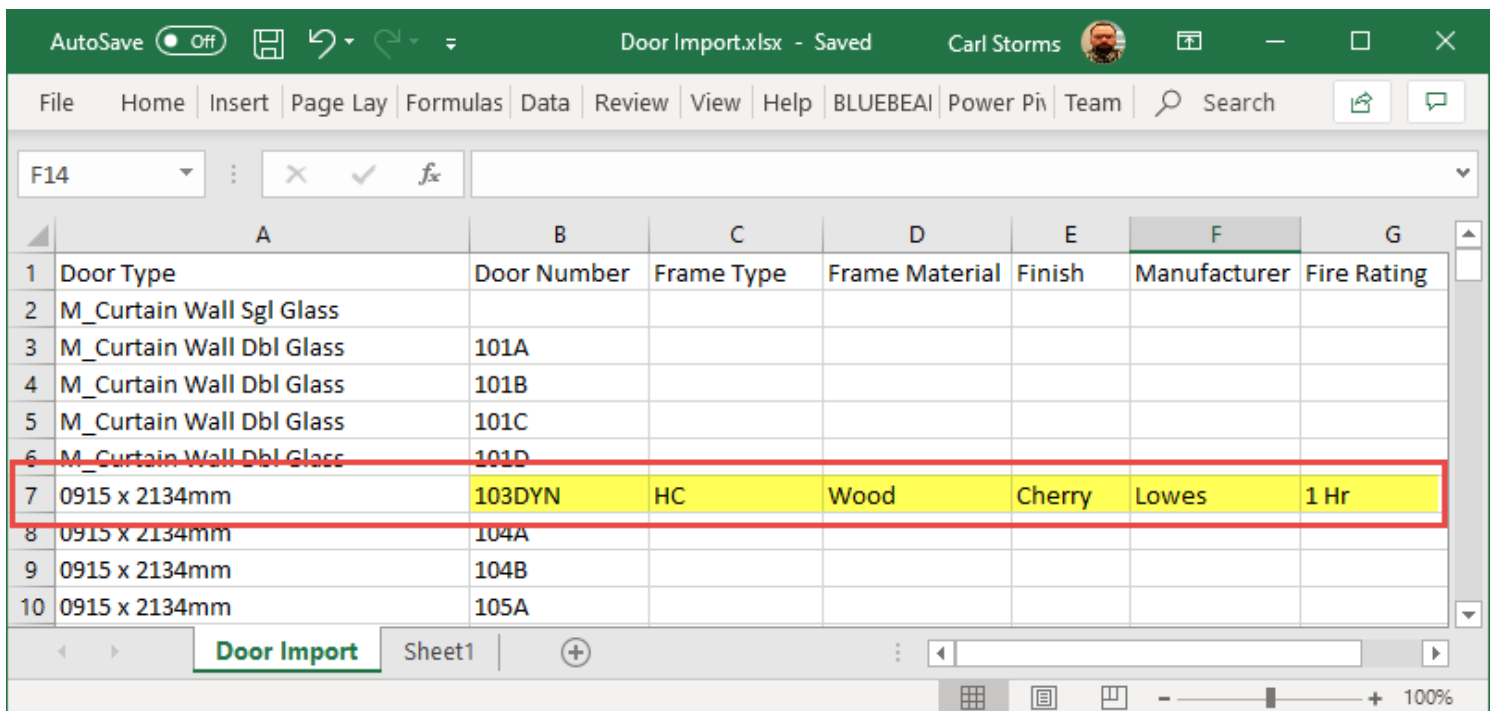
In this task we will use the Excel file created in the last task and add more data to it. Then we will import that data from Excel into our Revit project using Dynamo.

1. For this task, you can continue to work with the Revit file and Dynamo graph from task 1, or you can open the **Sample\_Project.rvt** Revit file, and then open the Dynamo file **Ex3\_Doors\_T2\_Begin.dyn** file from the **Ex\_3** folder in the lab dataset.
2. Now we can open the Excel file we created in the last task or open the one called **Door Export.xlsx** from the **Ex\_3** folder in the lab dataset. Inside the Excel file we are going to add information to this one row of cells.



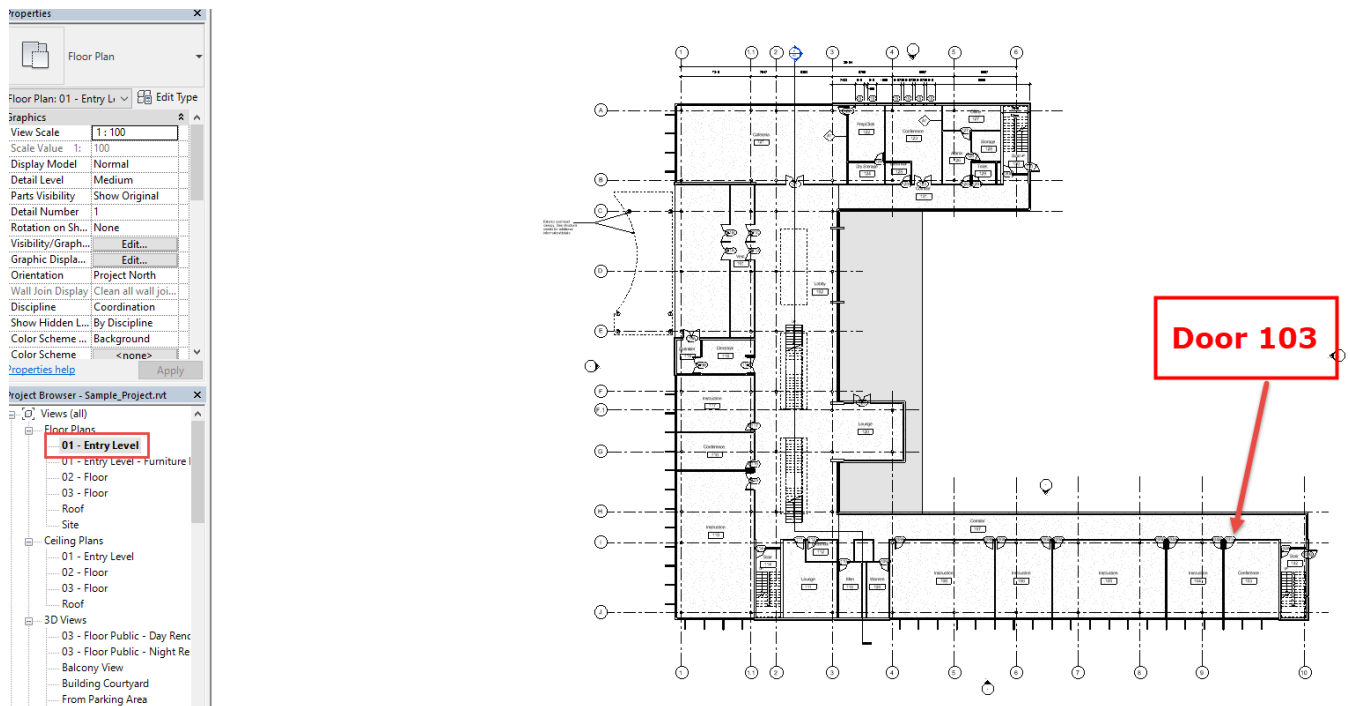
	A	B	C	D	E	F	G	H
1	Door Type	Door Number	Frame Type	Frame Material	Finish	Manufacturer	Fire Rating	
2	M_Curtain Wall Sgl Glass							
3	M_Curtain Wall Dbl Glass	101A						
4	M_Curtain Wall Dbl Glass	101B						
5	M_Curtain Wall Dbl Glass	101C						
6	M_Curtain Wall Dbl Glass	101D						
7	0915 x 2134mm	103						
8	0915 x 2134mm	104A						
9	0915 x 2134mm	104B						
10	0915 x 2134mm	105A						
11	0915 x 2134mm	105B						
12	0915 x 2134mm	106A						
13	0915 x 2134mm	106B						
14	0915 x 2134mm	108A						

3. In the real world we would be adding information to all or most of the cells, but for this exercise we will focus on just the one row (door) to make it easy to review inside Revit. Please add the information as shown below. Then save the Excel file with the name: **Door Import.**

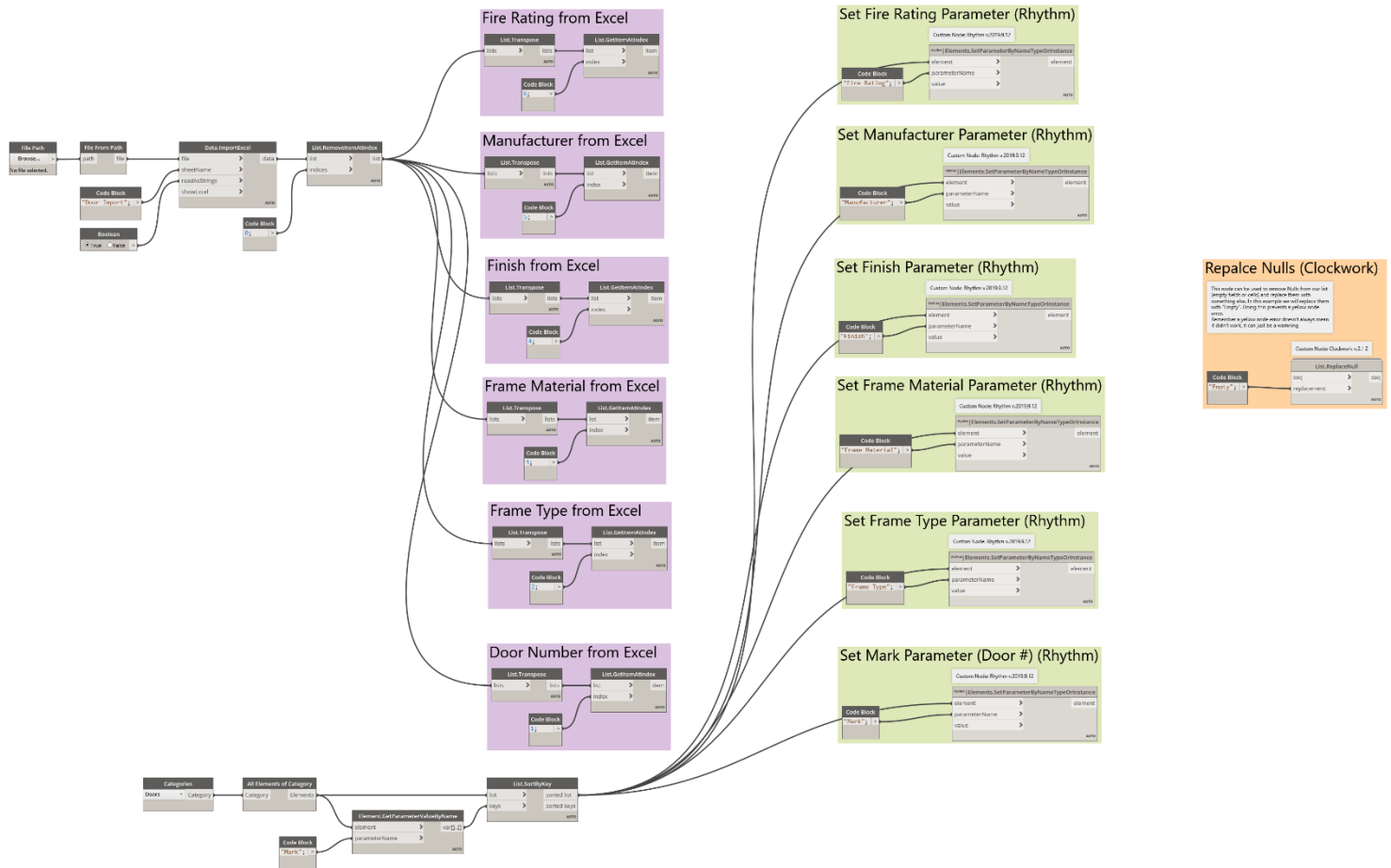


	A	B	C	D	E	F	G
1	Door Type	Door Number	Frame Type	Frame Material	Finish	Manufacturer	Fire Rating
2	M_Curtain Wall Sgl Glass						
3	M_Curtain Wall Dbl Glass	101A					
4	M_Curtain Wall Dbl Glass	101B					
5	M_Curtain Wall Dbl Glass	101C					
6	M_Curtain Wall Dbl Glass	101D					
7	0915 x 2134mm	103DYN	HC	Wood	Cherry	Lowes	1 Hr
8	0915 x 2134mm	104A					
9	0915 x 2134mm	104B					
10	0915 x 2134mm	105A					

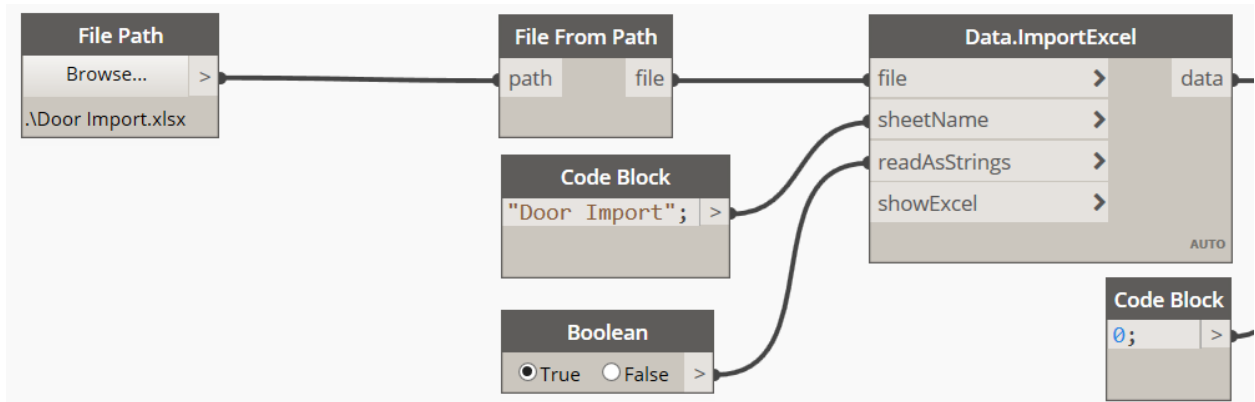
- Return to the Revit file and make sure you are on the **01 – Entry Level** Floor Plan then zoom into the bottom right corner, so you have an unobstructed view of Door 103.



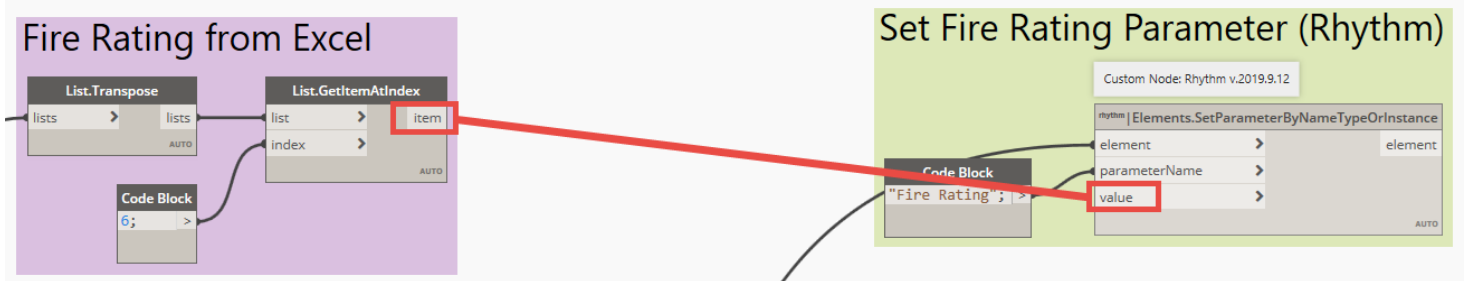
- We will now prep the Dynamo graph (**Ex3\_Doors\_T2\_Begin.dyn**) to import our new Door data.



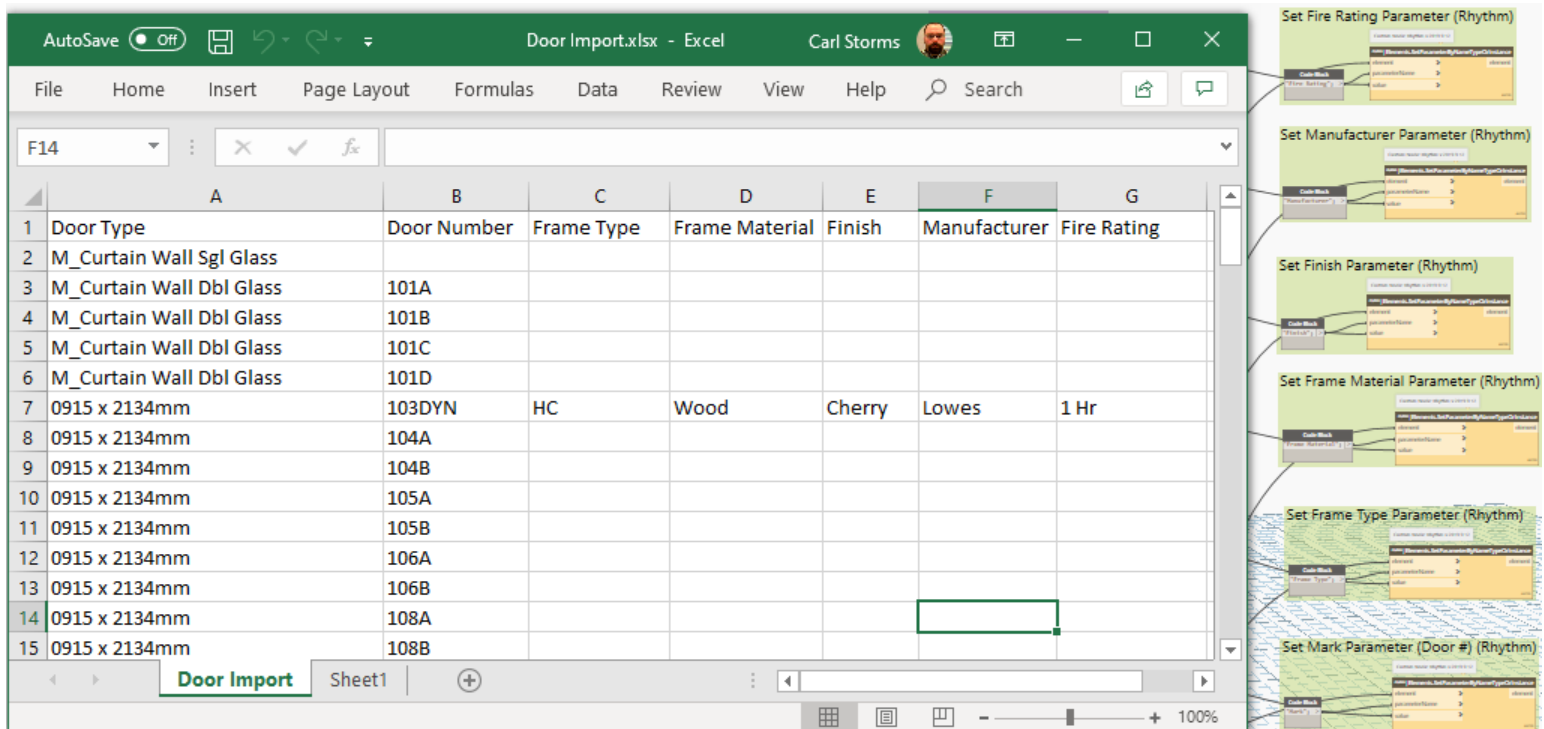
- We need to make sure that the graph is pointing to the correct location for the Excel file we just saved. You can do this by hitting the **Browse...** button in the **File Path** node and navigate to where you saved the file. You can also use the **Door Import.xlsx** file from the **Ex\_3** folder in the lab dataset.



- Now we need to connect the *item* output from the **List.GetItemAtIndex** nodes in the purple groups to the corresponding *value* input of the **Element.SetParameterByNameTypeOrInstance** nodes (**Rhythm** custom node) in the green groups.

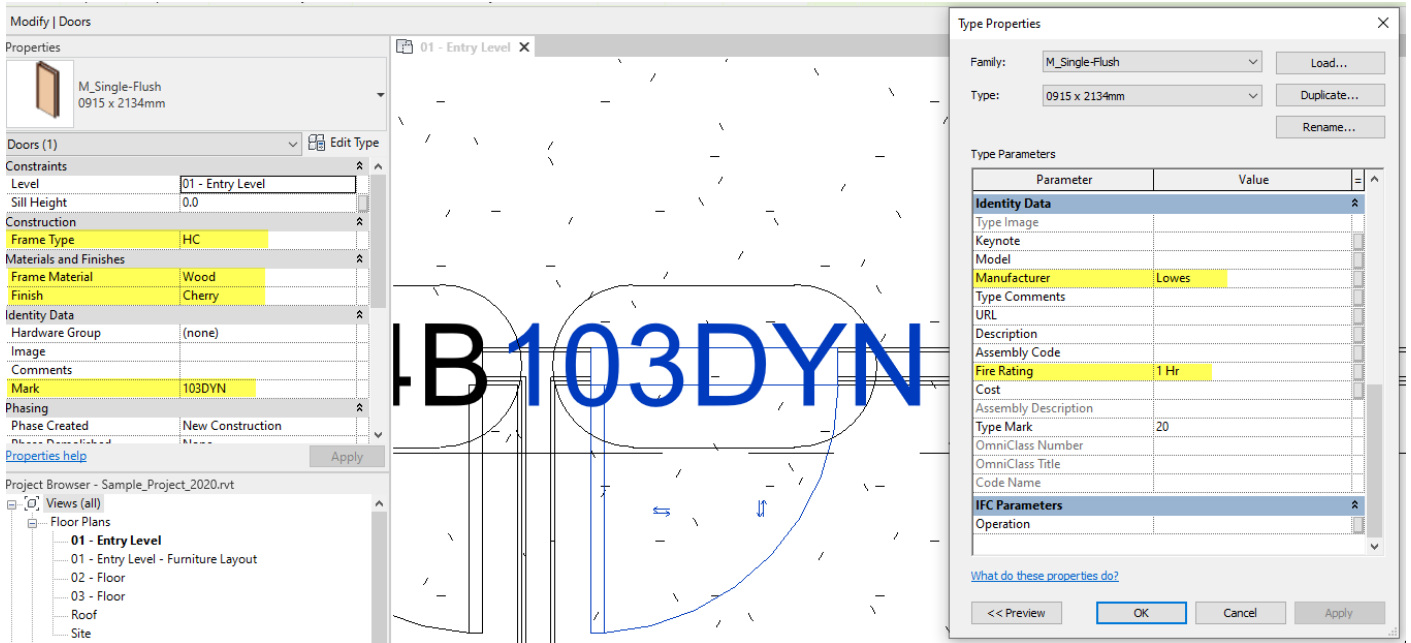


- For this example, we will connect all six data options, and then hit the F5 to see what happens.
- You will see that the that the Excel file opens, and there are six yellow error/warnings.

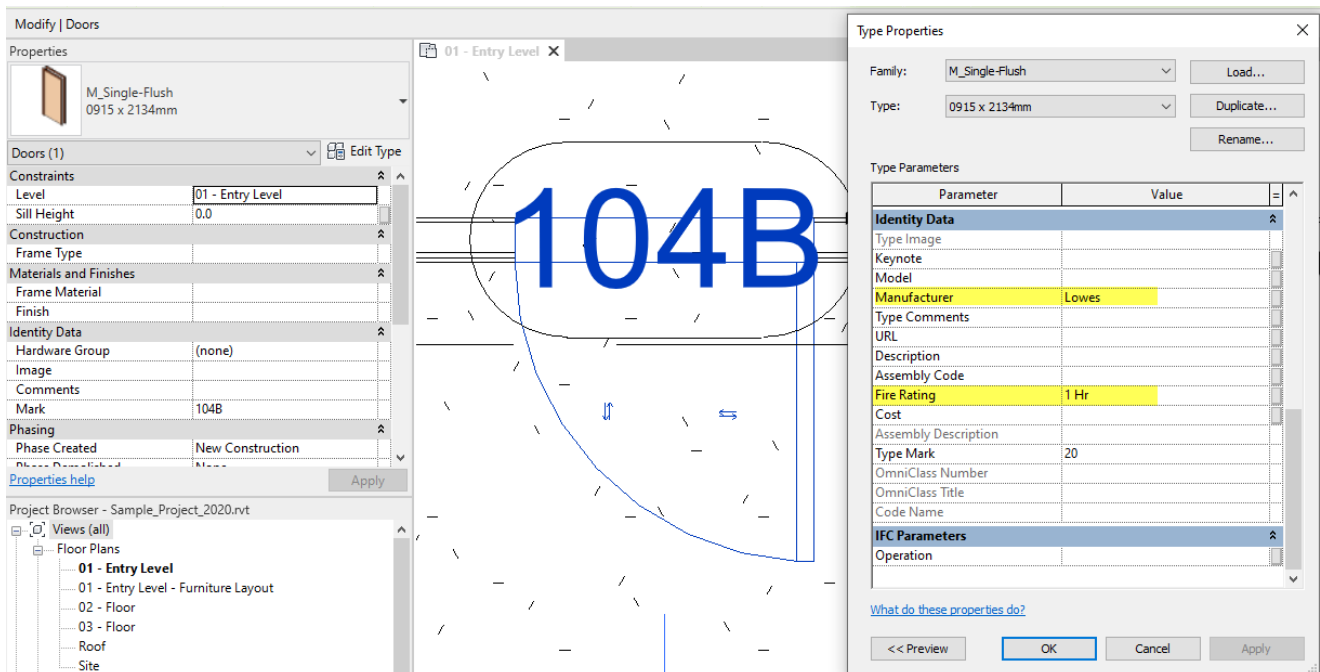


Door Type	Door Number	Frame Type	Frame Material	Finish	Manufacturer	Fire Rating
M_Curtain Wall Sgl Glass	101A					
M_Curtain Wall Dbl Glass	101B					
M_Curtain Wall Dbl Glass	101C					
M_Curtain Wall Dbl Glass	101D					
0915 x 2134mm	103DYN	HC	Wood	Cherry	Lowes	1 Hr
0915 x 2134mm	104A					
0915 x 2134mm	104B					
0915 x 2134mm	105A					
0915 x 2134mm	105B					
0915 x 2134mm	106A					
0915 x 2134mm	106B					
0915 x 2134mm	108A					
0915 x 2134mm	108B					

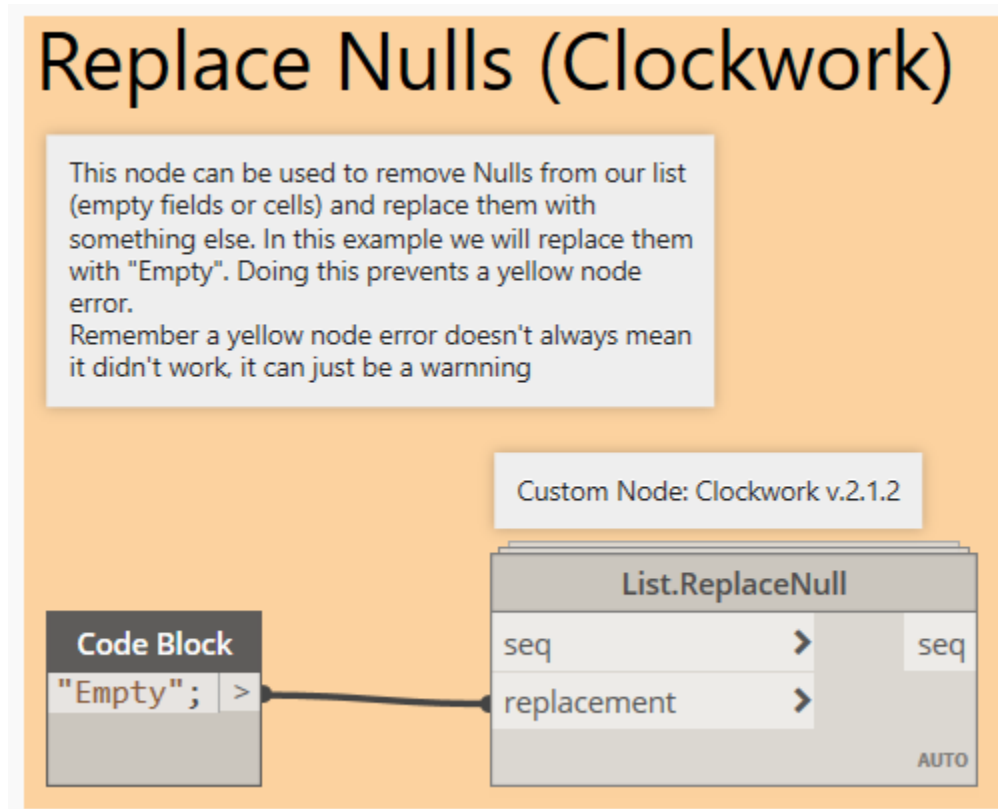
10. This is one of the cases when yellow means warning and not error. If we go to the Revit view and click on Door 103 we see that all the new information has been added.



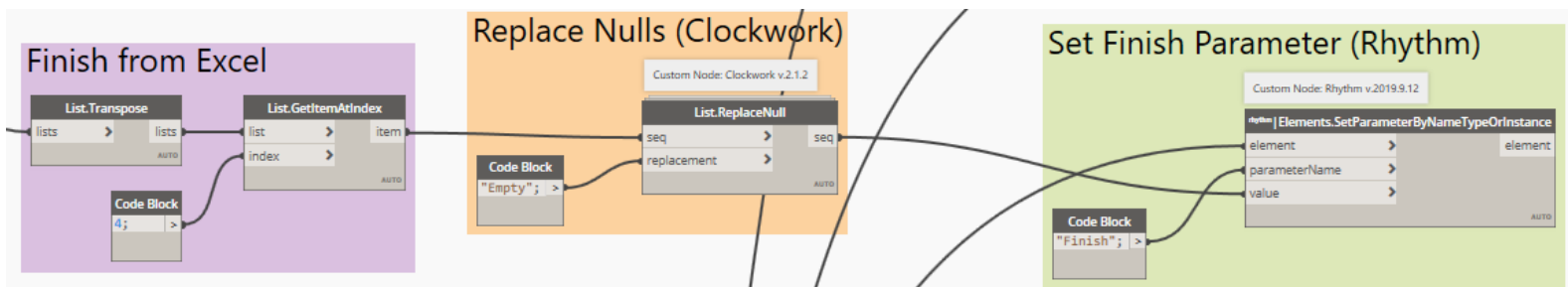
11. You can also see that the Type Parameters have also been added to all the other doors in the project of the same family type (0915 x 2134mm). You will also see that none of the instance parameters changed (as expected) for the other doors of the same Type Parameters.



12. The reason we got the yellow warnings is because there was some Null (empty) values that we tried to import back into Revit, and Dynamo doesn't like that. We can use a custom node from the **Clockwork** package to replace the Null value with a string (text value of your choice). In our example we are replacing the nulls with the text *Empty*.

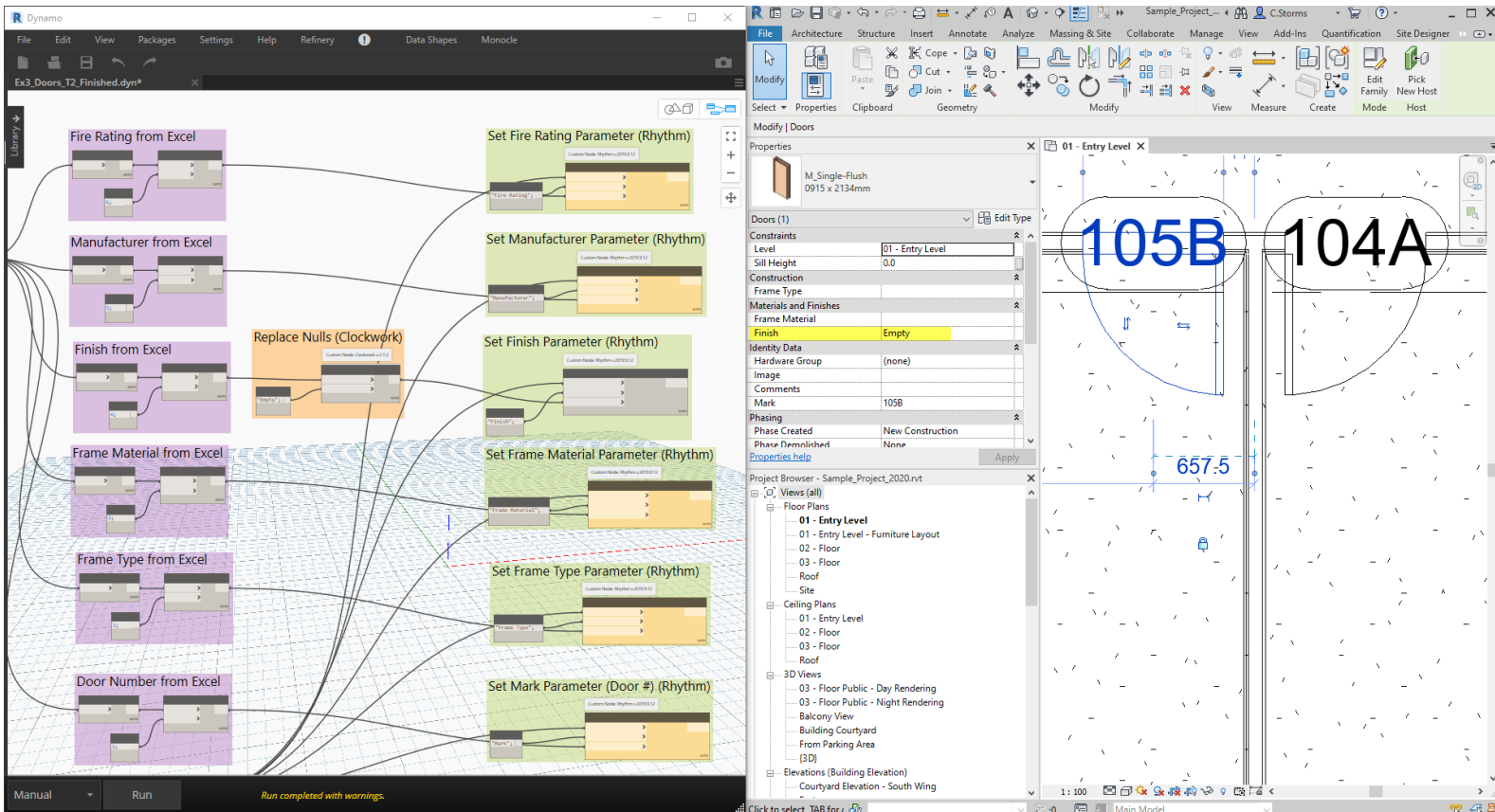


13. To use this node, we run our information from the purple group through the **List.ReplaceNull** node, then into the green group.





14. If we try this with the *Finish* parameter as shown above and run the graph again, we will see that we now only have five yellow warnings, and all of the Finish parameters now have *Empty* in them if we didn't assign them a finish in our Excel file.

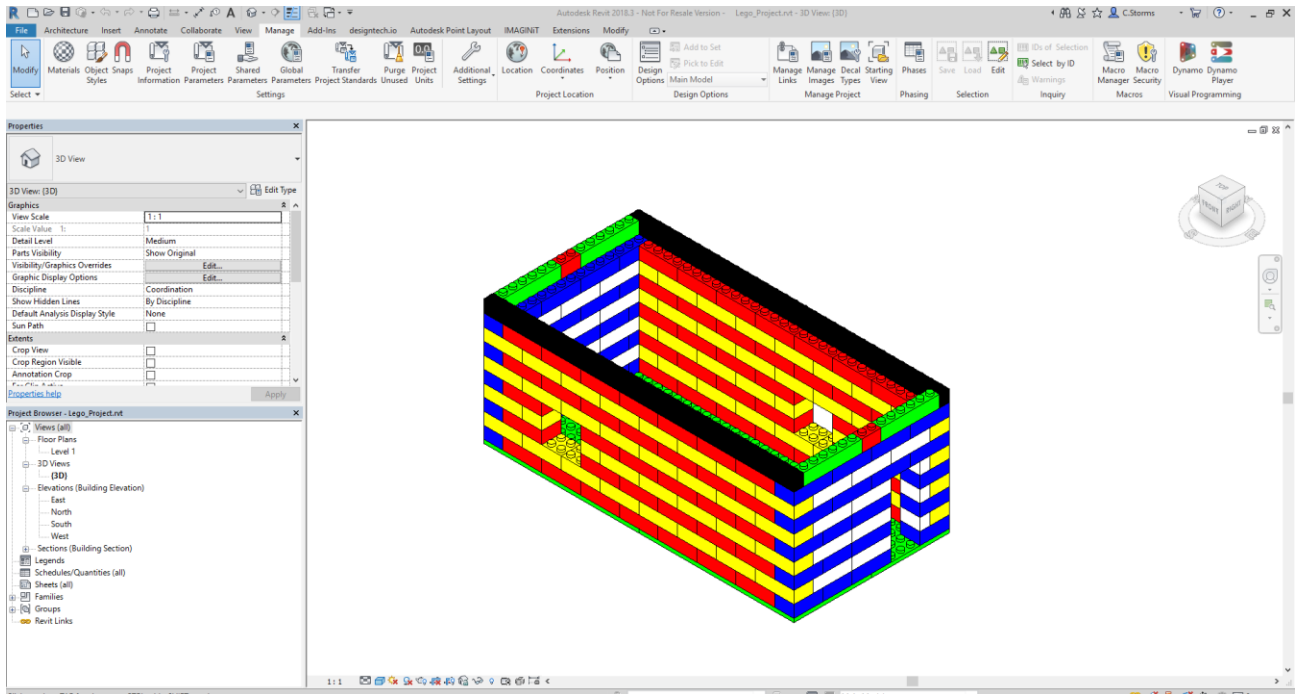


## Ex 4a) Pie - NodeModelCharts

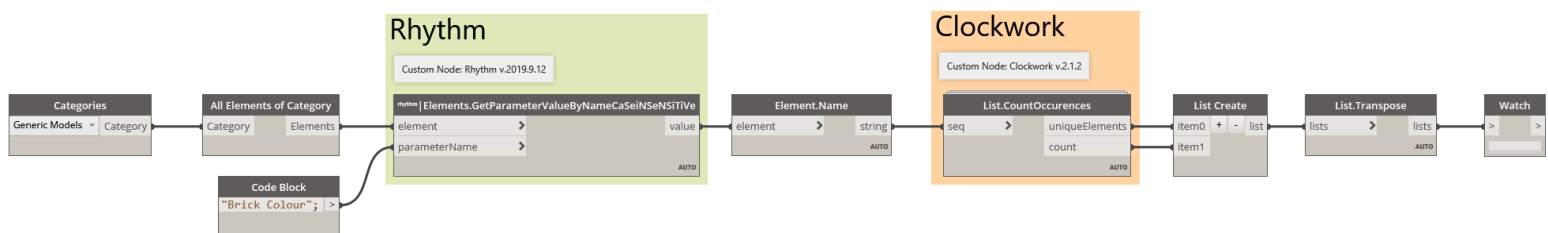
In this exercise we will extract some data from our Revit project and display it visually using the custom **NodeModelCharts** package.

### Task 1

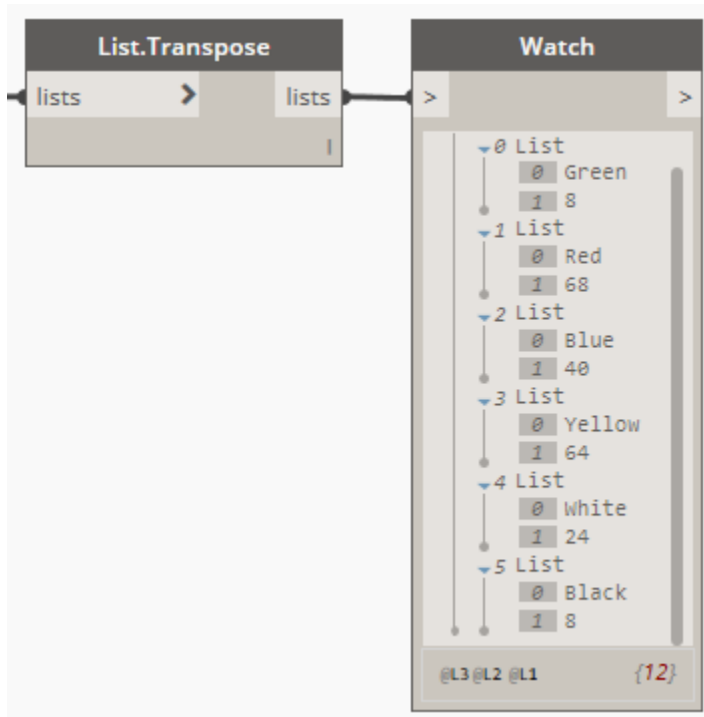
1. From the dataset folder **Ex\_4a** open the following file: **Lego\_Project.rvt** in Revit.



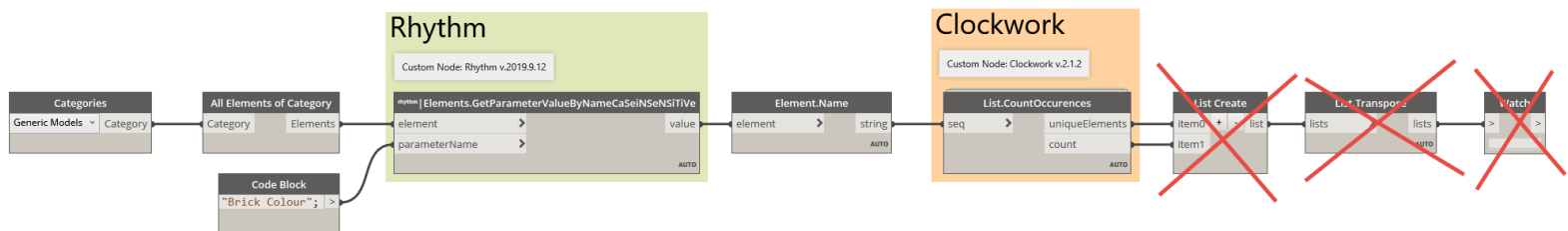
2. Start Dynamo from the Manage tab.
3. Now Open the Dynamo graph called **Ex4a\_Pie\_T1\_Begin.dyn** from the dataset folder **Ex\_4a**.



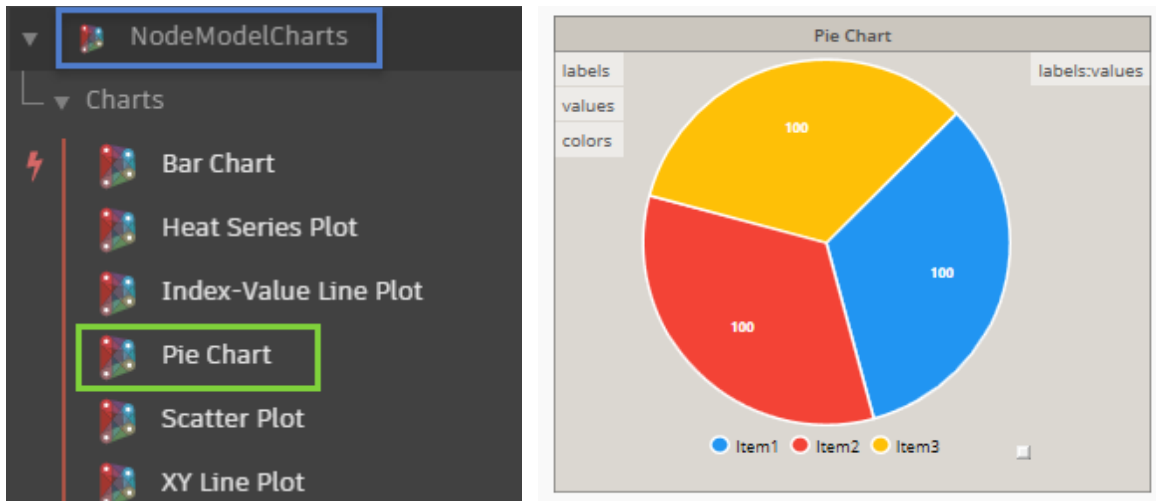
4. Hit the Run button to see the information we get, with this graph as is.
5. The final output is a list, of six lists, each with two items in it. We can now see each colour, and the number of Lego bricks that are in that colour.



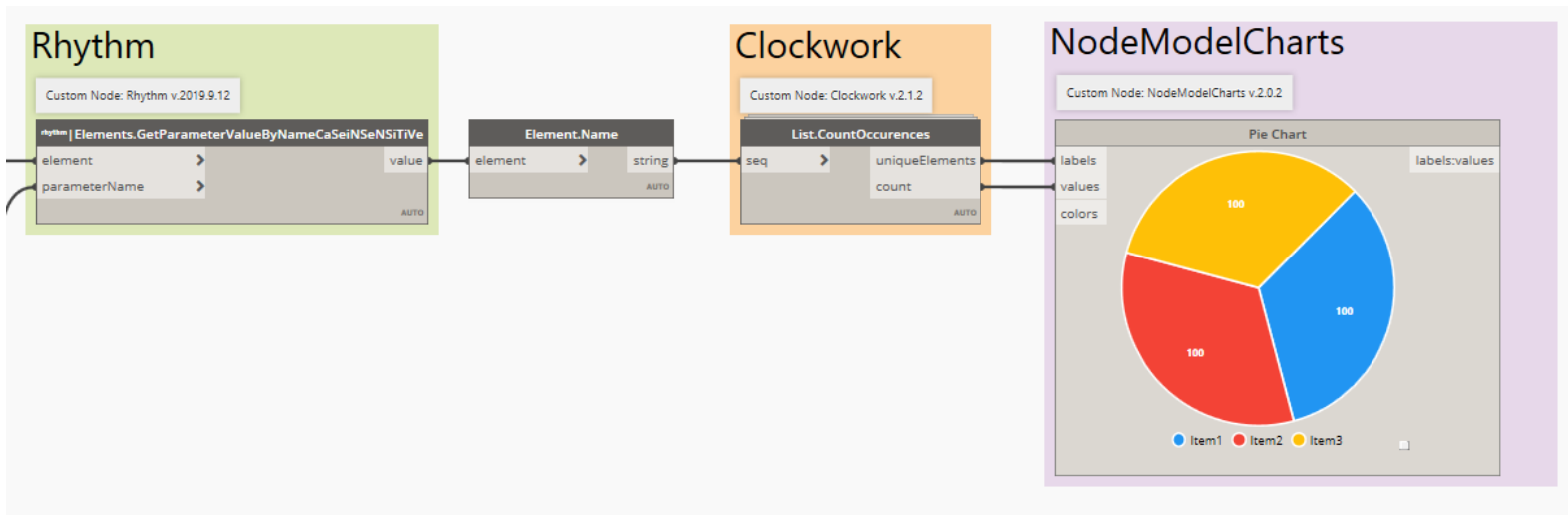
6. This gives us the information we wanted, how many bricks there are of each colour, however the **Watch** node is not the best way to visualize this information. By adding just one node from the **NodeModelCharts** custom package we can make this data POP!
7. Before we add more nodes to our graph, we need to first delete three of the existing nodes from the end of the graph: **List Create**, **List Transpose** and **Watch**.



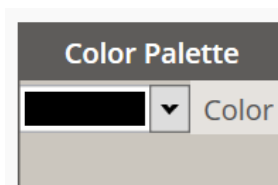
8. We are going to make a Pie Chart using a **NodeModelCharts** package node. The **Pie Chart** node can be found under NodeModelCharts > Charts > Pie Chart.



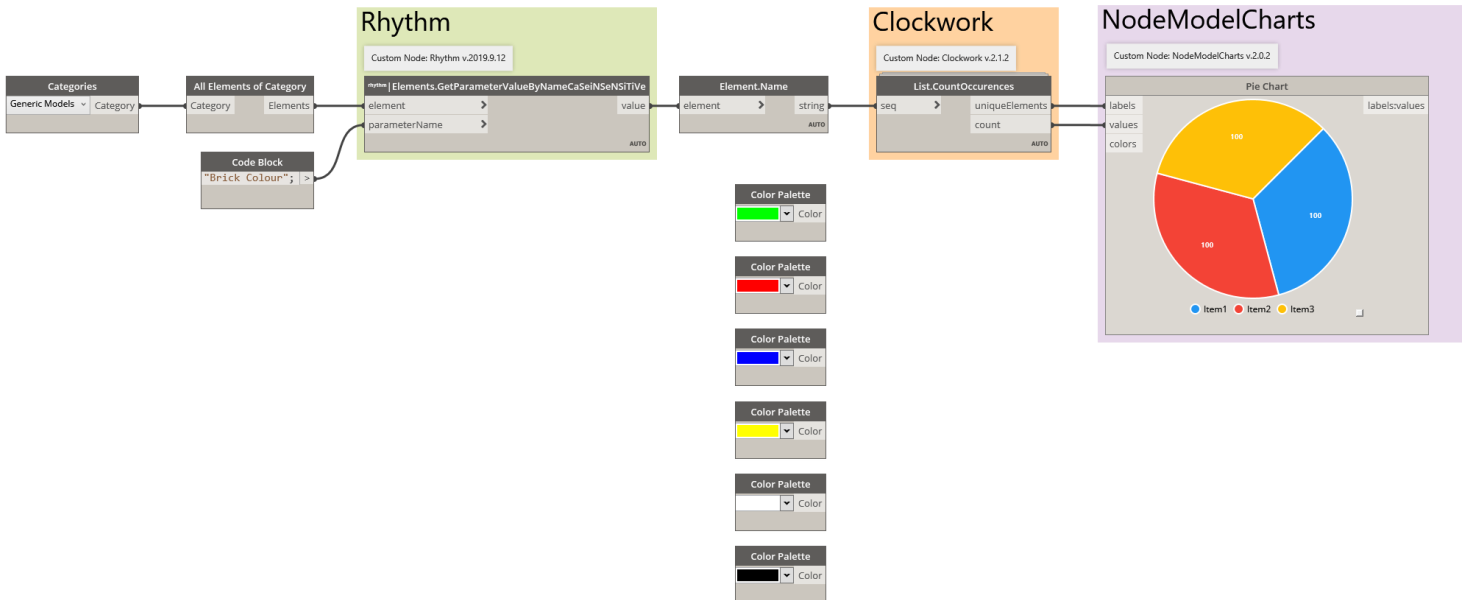
9. We can now connect both outputs from the **List.CountOccurrences** node to the top two inputs of the **Pie Chart** node as shown below.



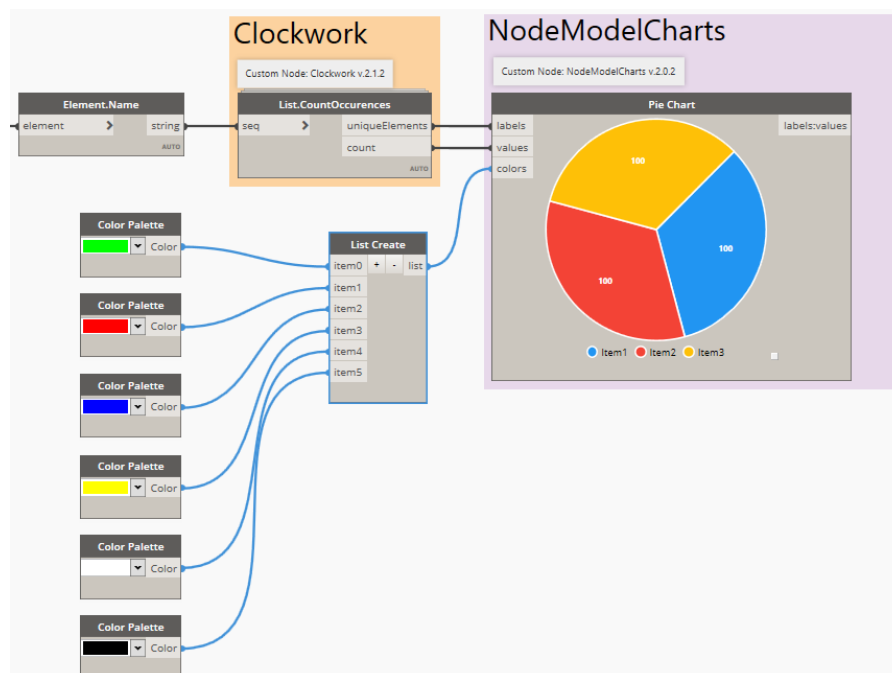
10. Now to get the data to POP we also need to add six **Color Palette** nodes. These will give us a way to pick the colours we want to be displayed for each one of our data categories (brick colours).



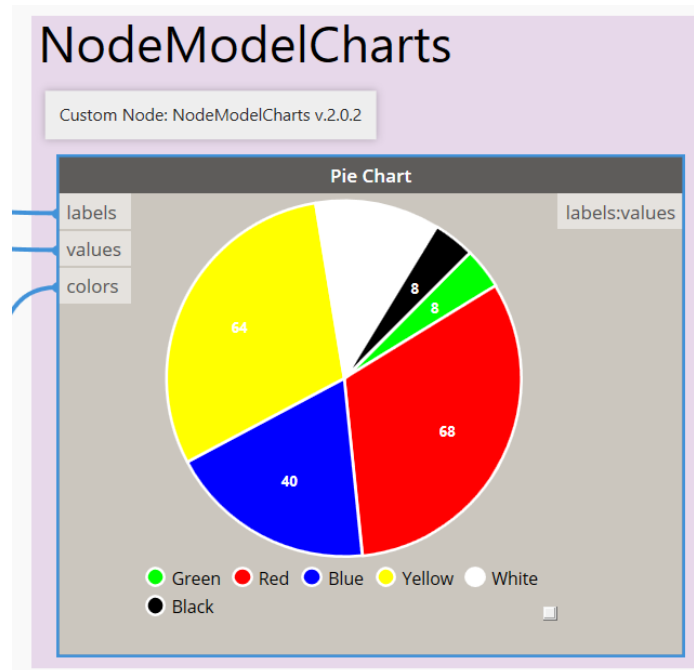
11. Once you have those six nodes on the canvas stack them in a vertical column. Then starting from the top working your way down pick the following colours: Green, Red, Blue, Yellow, White, Black.



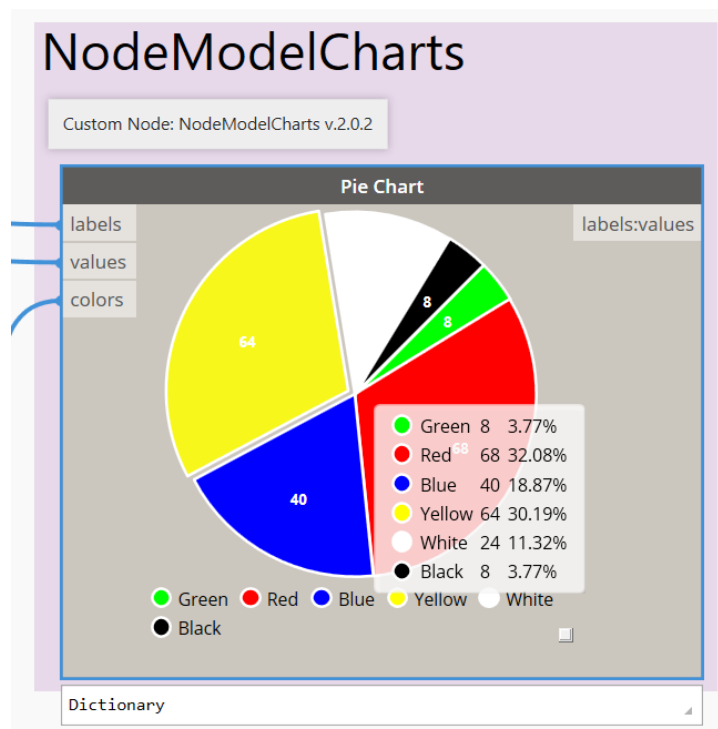
12. For the six colours we need to create a list, we can do this by adding a **List Create** node on the canvas and hitting the “+” sign five times to make six inputs. Then we connect the **List Create** output to the **colors** input in the **Pie Chart** node.



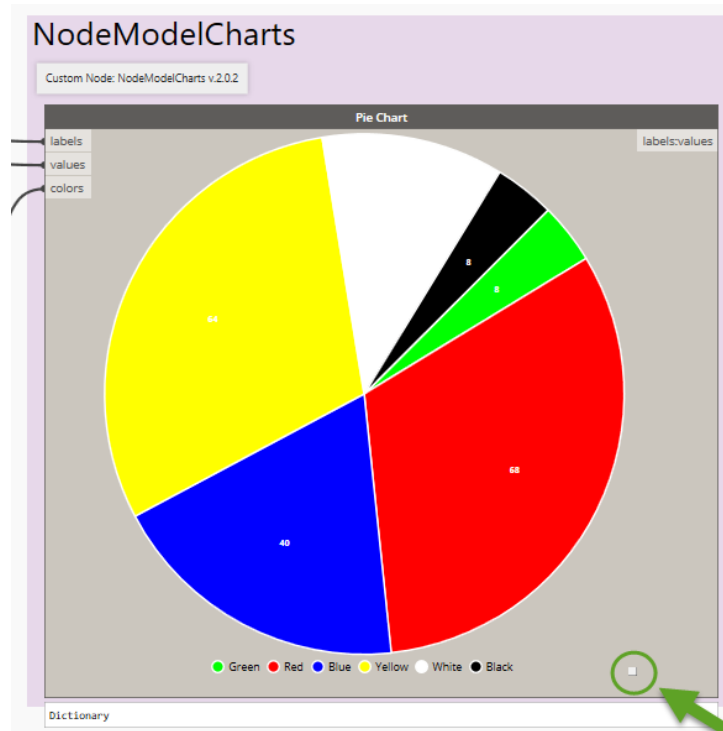
13. Now we can hit F5 to run the graph and see what the pie chart looks like.



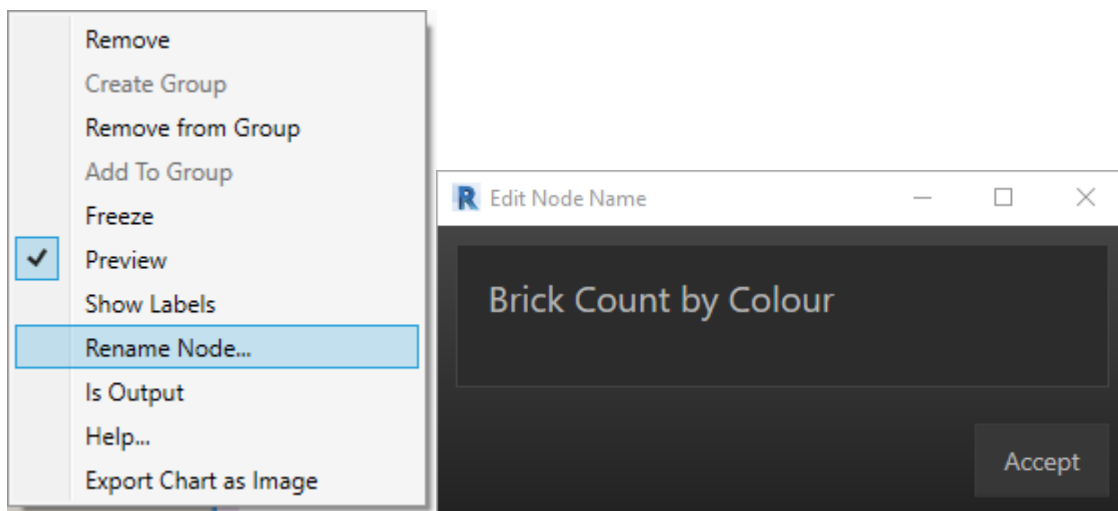
14. If you hover over one of the pie segments you get a pop-up that shows the results for all the segments.



**Note:** You may need to adjust the size of the **Pie Chart** node, you can do so by dragging outwards the small white square grip in the bottom right corner.



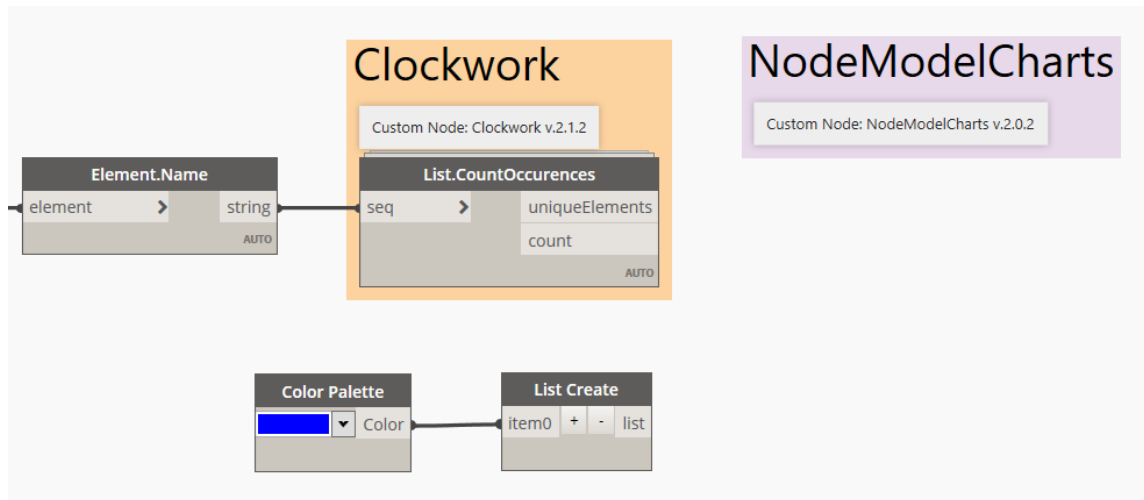
15. If you right click on the **Pie Chart** node you can change the name of the node using the context menu. In this case let's call the node: Brick Count by Colour.



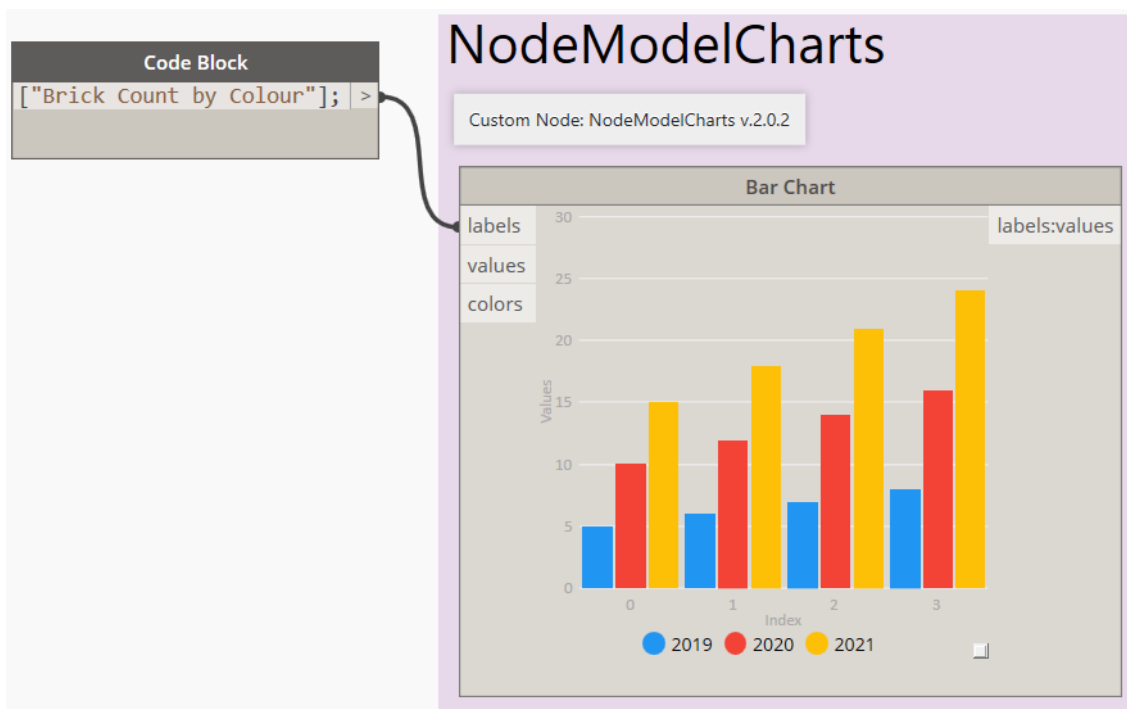




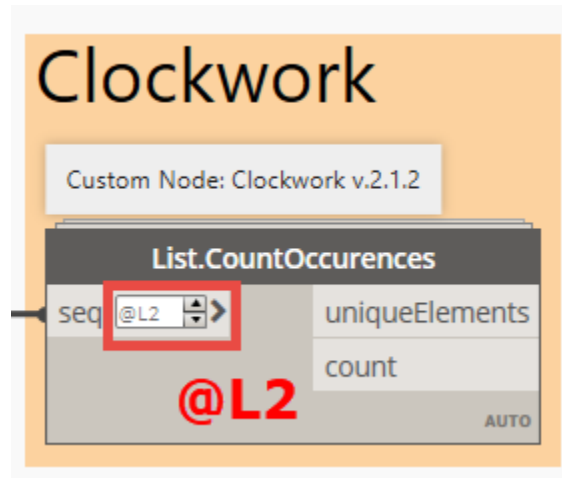
- First, we will delete the **Pie Chart** node, and all but the blue **Color Palette** nodes. We will also hit the “-” button on the **List Create** node until it’s back to just one input (item 0). You will have to reconnect the **Color Palette** node output to the *item 0* input of the **List Create** node.



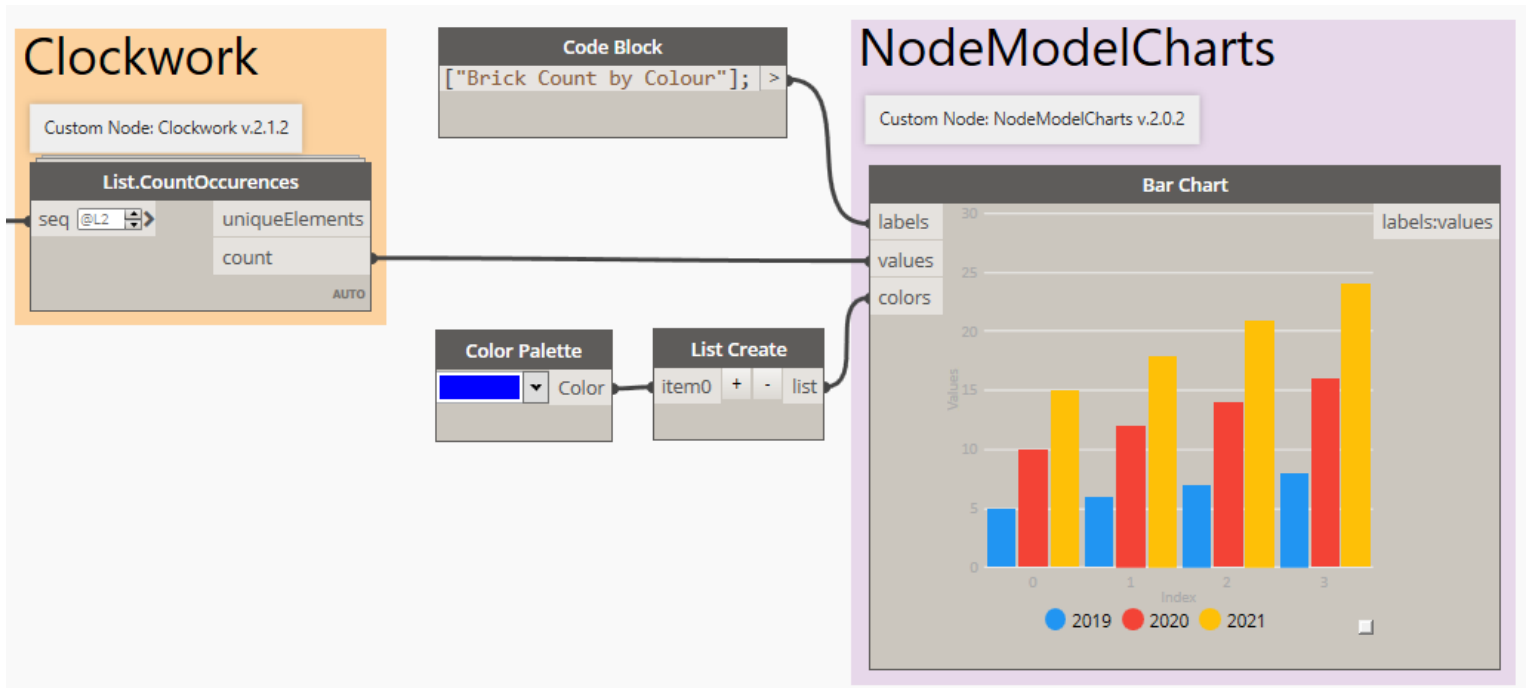
- Next we can add the **Bar Chart** node from the **NodeModelCharts** package. We will also need to add a **Code Block** with the following added to it [“Brick Count by Colour”]; which will connect to the *labels* input for the **Bar Chart** node. **Note:** the [ ] are required in the **Code Block**, the output must be in a list format for the **Bar Chart** node to work correctly.



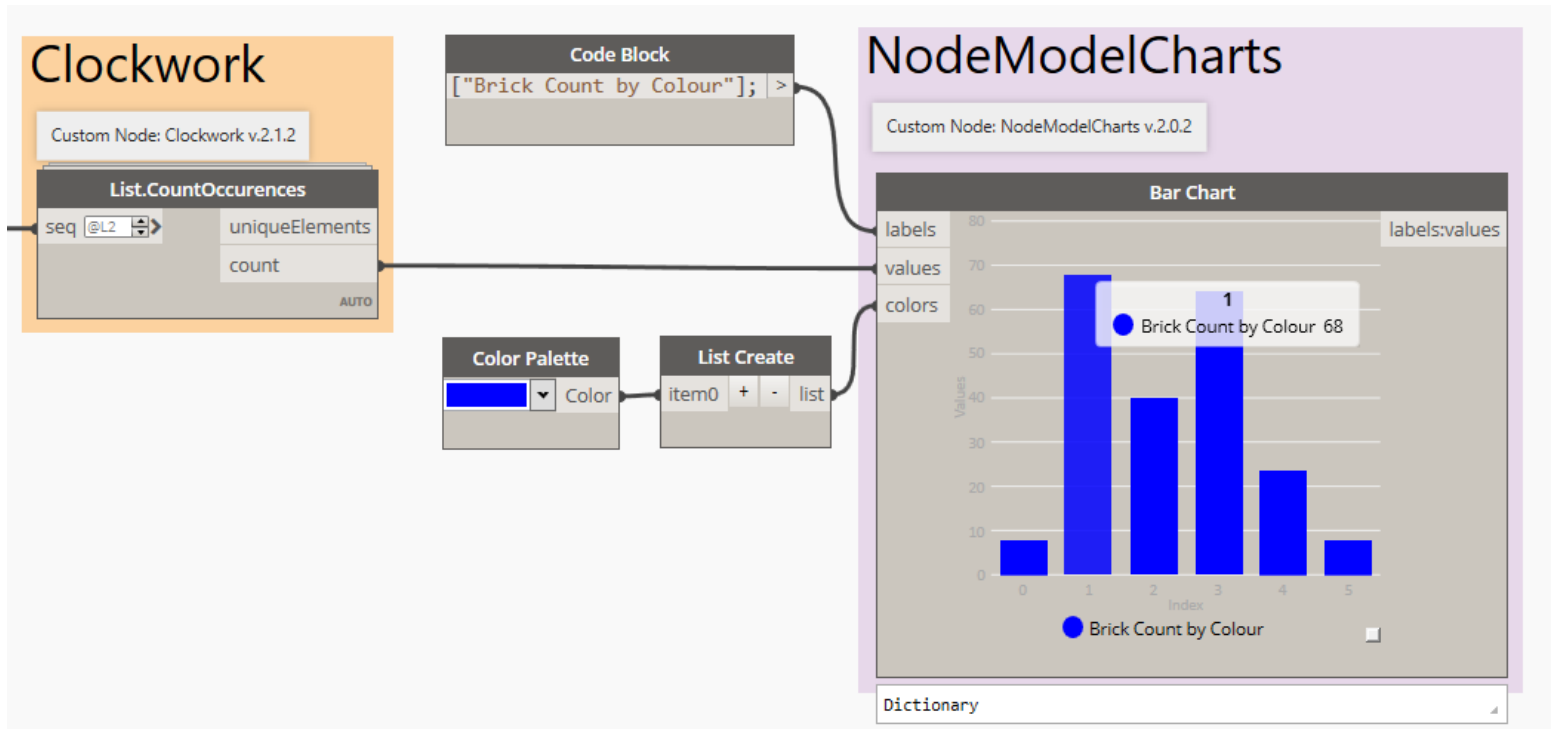
- Before we connect the *count* output from the **List.CountOccurrences** node into the **Bar Chart** node *values* input, we need to change the List@Level of the **List.CountOccurrences** node to @L2.



- Now we can connect the **List.CountOccurrences** node and the **List Create** node to the **Bar Chart** node as shown in the following image (*count* output to *values* input).



- The last step is to hit the Run button and see the outcome after the graph has ran.



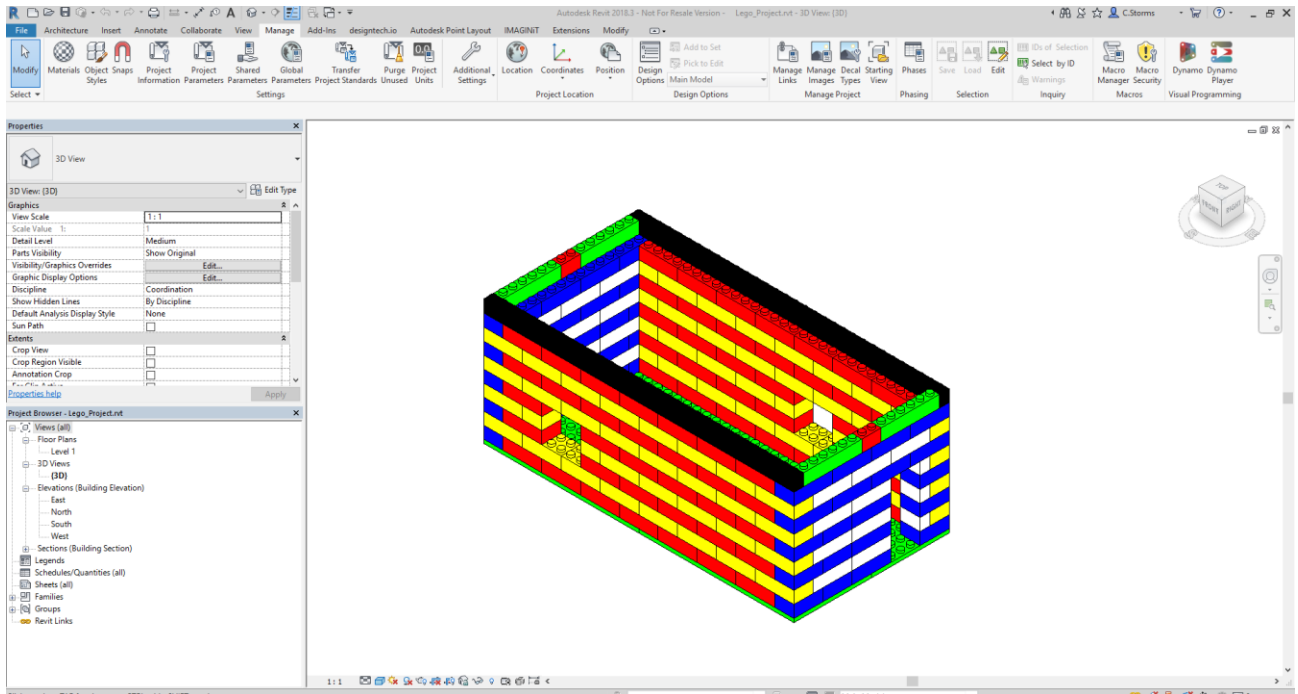
**Note:** That if you hover over a bar you will get a pop-up that displays the information about that specific bar.

## Ex 4b) Pie – Data-Shapes

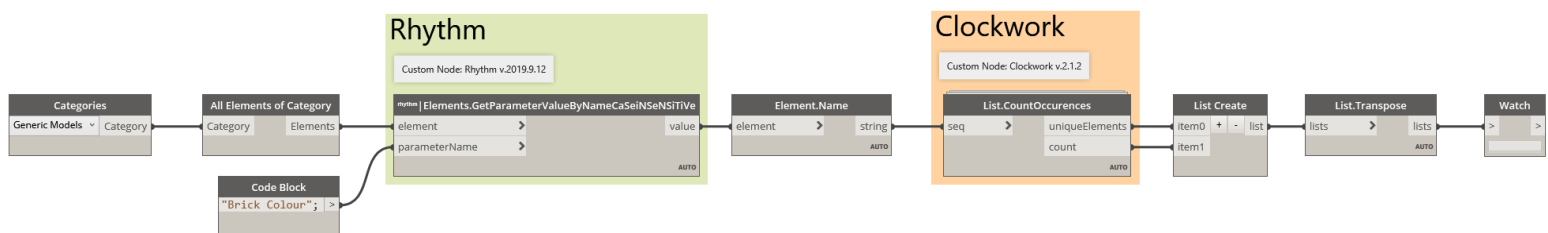
In this exercise we will extract some data from our Revit project and display it visually using the custom **Data-Shapes** package.

### Task 1

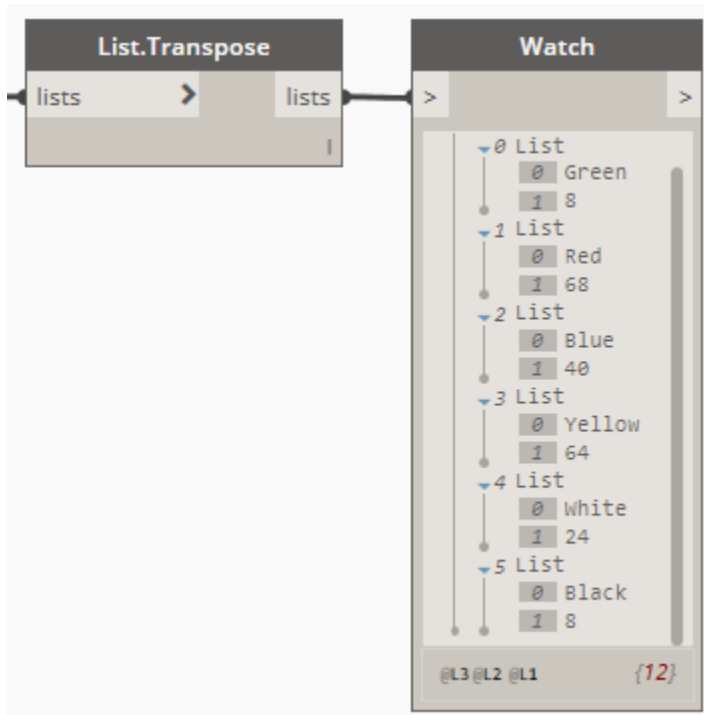
1. From the dataset folder **Ex\_4b** open the following file: **Lego\_Project.rvt** in Revit.



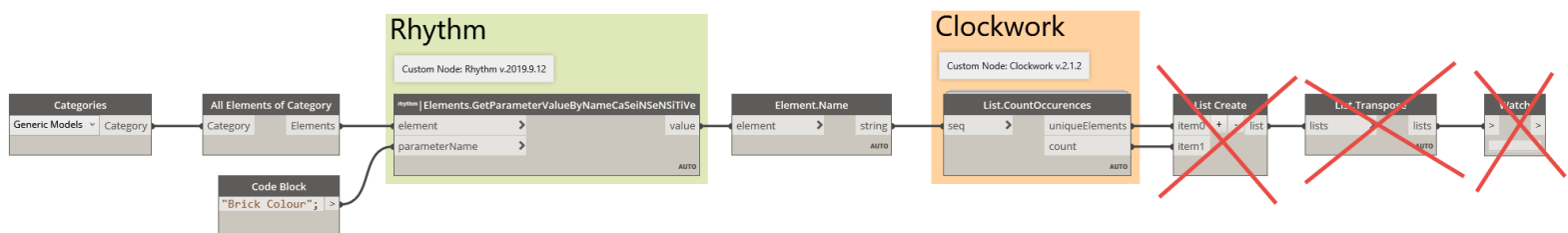
2. Start Dynamo from the Manage tab.
3. Now Open the Dynamo graph called **Ex4b\_Pie\_T1\_Begin.dyn** from the dataset folder **Ex\_4b**.



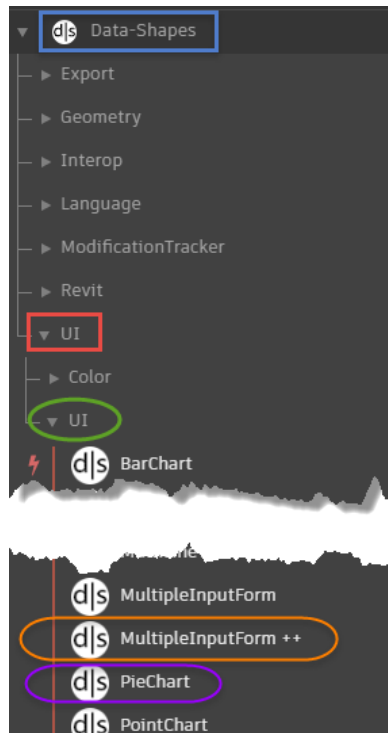
4. Hit the Run button to see the information we get, with this graph as is.
5. The final output is a list, of six lists, each with two items in it. We can now see each colour, and the number of Lego bricks that are in that colour.



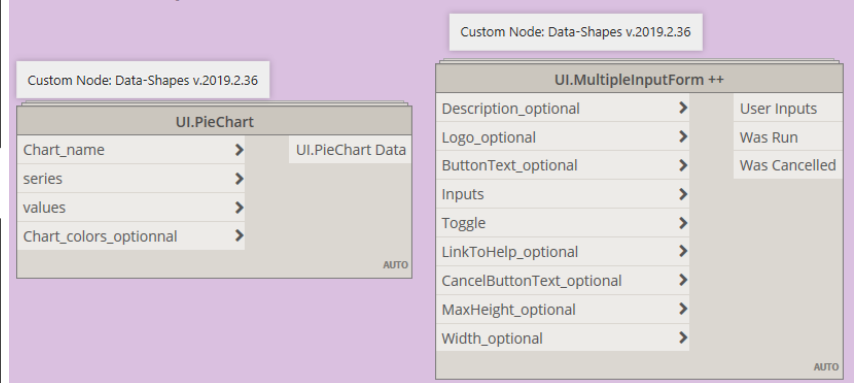
6. This gives us the information we wanted, how many bricks there are of each colour, however the **Watch** node is not the best way to visualize this information. By adding just two nodes from the **Data-Shapes** custom package we can make this data POP!
7. Before we add more nodes to our graph, we need to first delete three of the existing nodes from the end of the graph: **List Create**, **List Transpose** and **Watch**.



8. We are going to make a Pie Chart using the following **Data-Shapes** package nodes. **UI.PieChart** and **UI.MultipleInputForm ++** which can be found under Data-Shapes > UI > PieChart and Data-Shapes > UI > UI > MultipleInputForm ++.

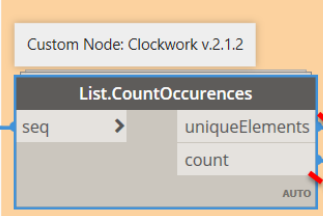


## Data-Shapes

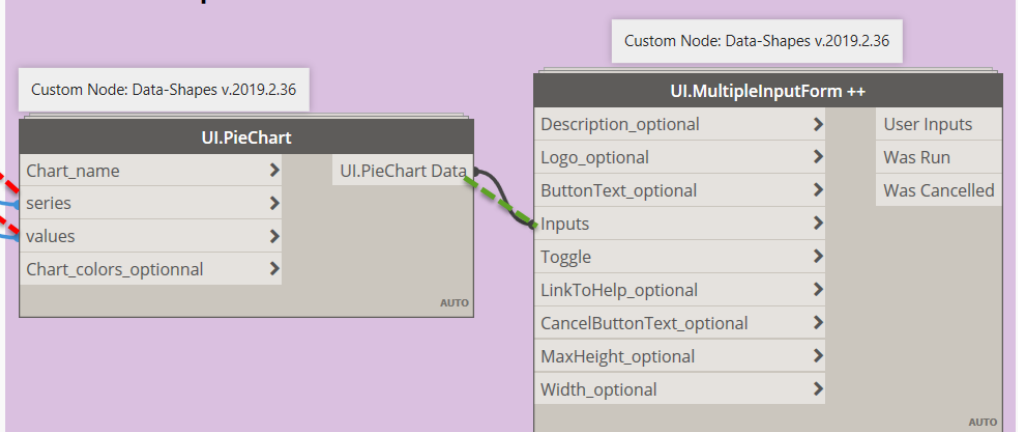


- We can now connect both outputs from the **List.CountOccurences** node to the second and third two inputs of the **UI.PieChart**. Then we can also connect the ouput from the **UI.PieChart** node to the *inputs* port of the **UI.MultipleInputForm ++** node as shown in the following image.

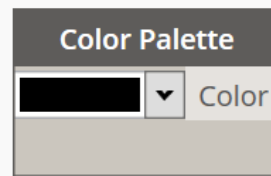
## Clockwork



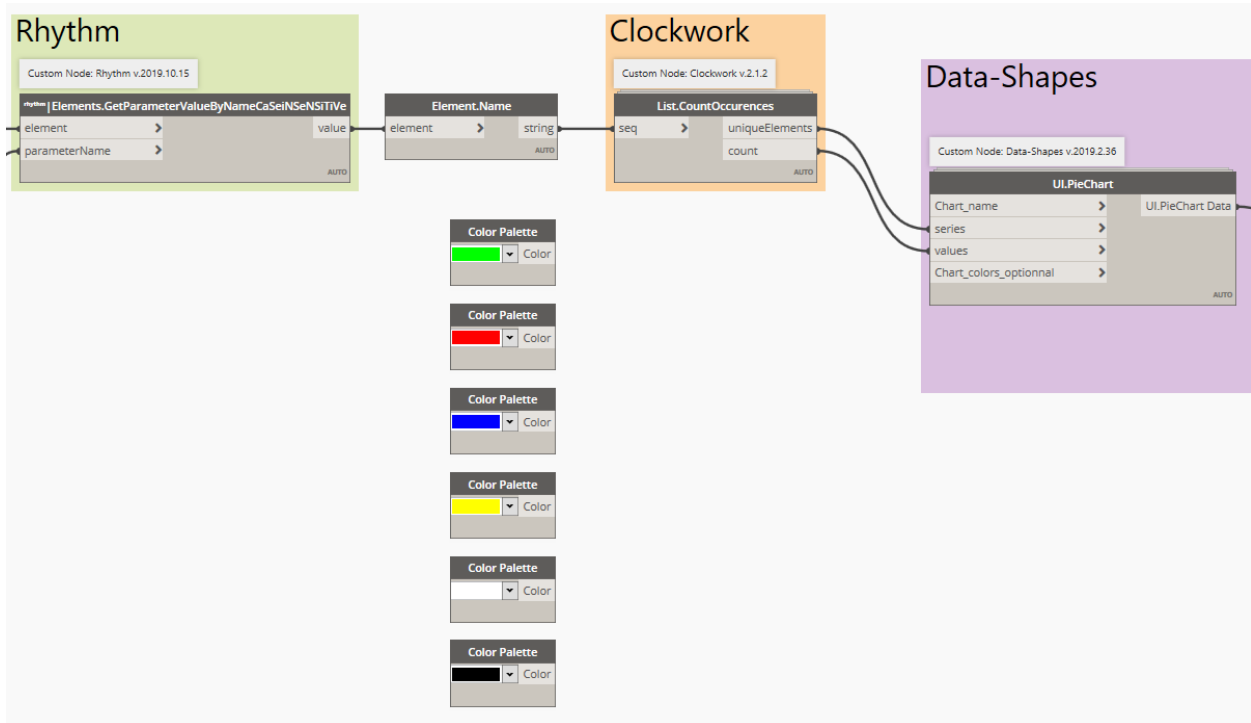
## Data-Shapes



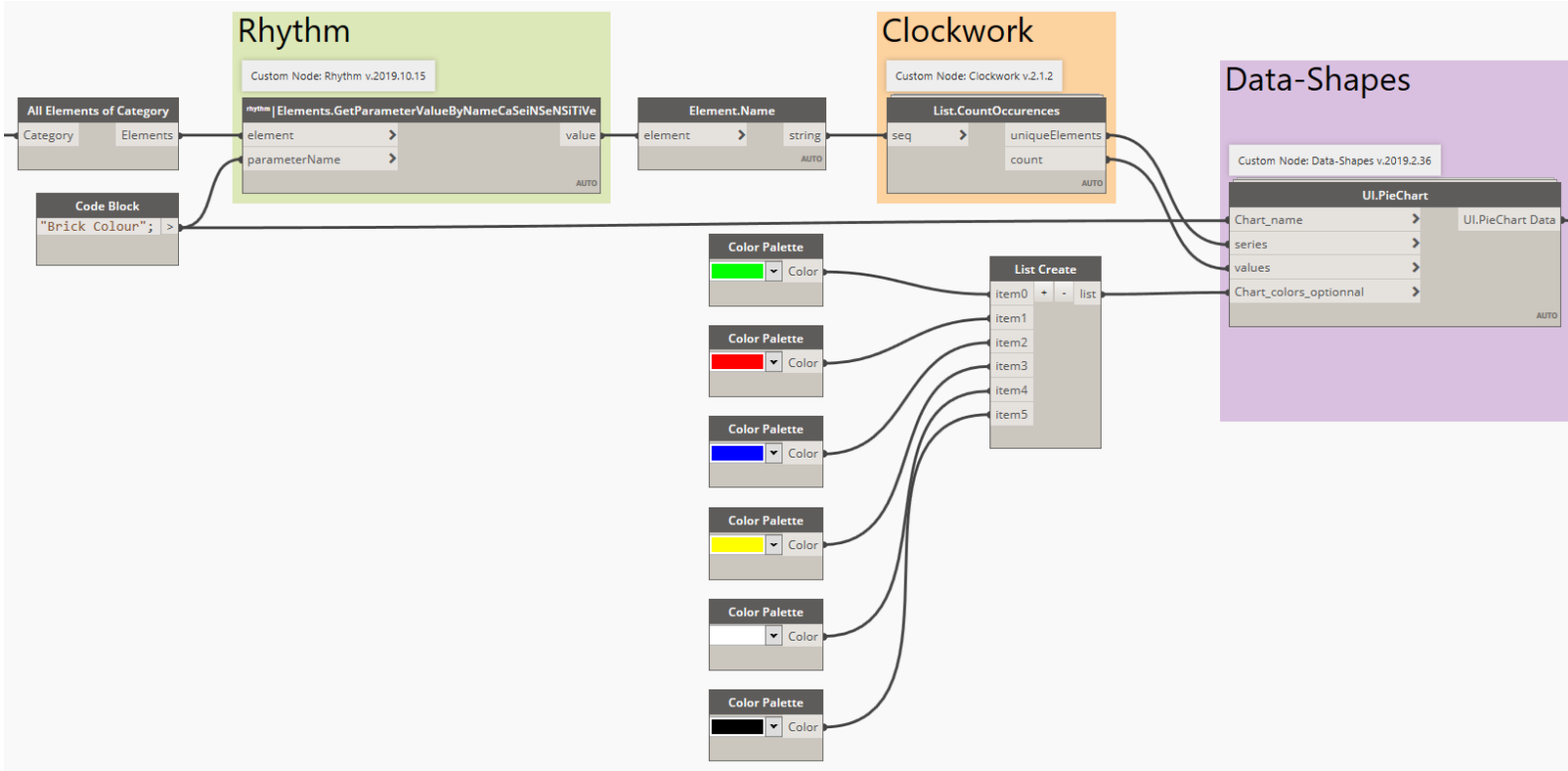
10. Now to get the data to POP we also need to add six **Color Palette** nodes. These will give us a way to pick the colours we want to be displayed for each one of our data categories (brick colours).



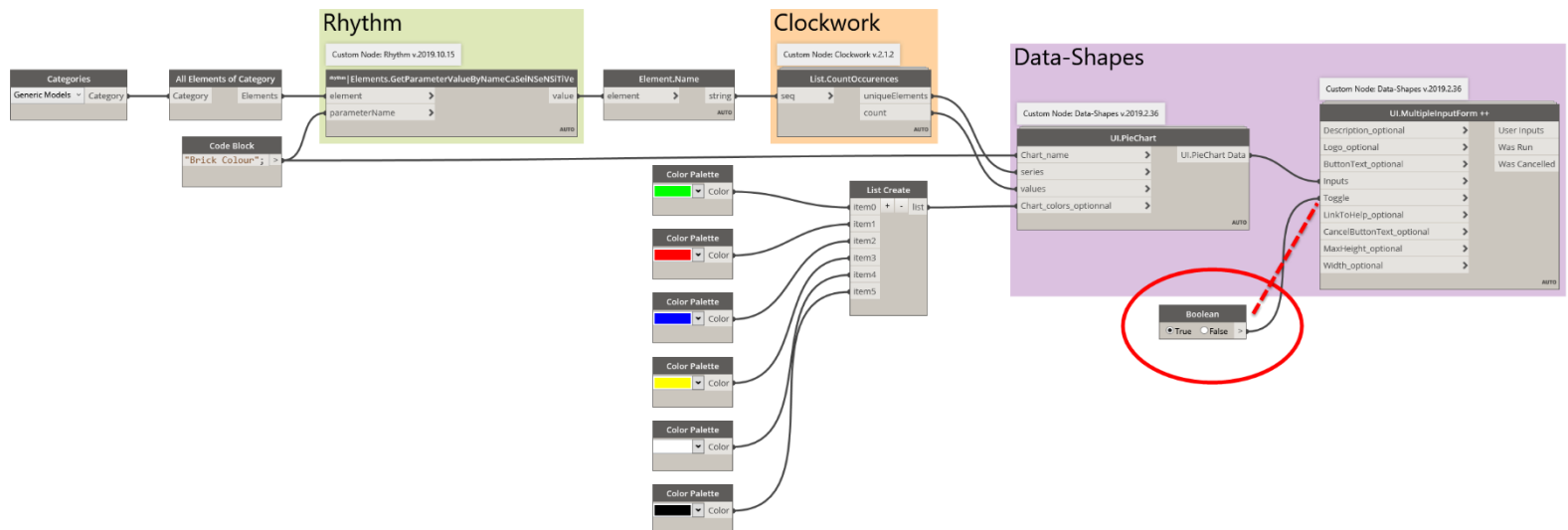
11. Once you have those six nodes on the canvas stack them in a vertical column. Then starting from the top working your way down pick the following colours: Green, Red, Blue, Yellow, White, Black.



12. For the six colours we need to create a list, we can do this by adding a **List Create** node on the canvas and hitting the “+” sign five times to make six inputs. Then we connect the **List Create** output to the *Chart\_colors\_optional* input in the **UI.PieChart** node. We can also connect the output from the **Code Block** to the *Chart\_name* input in the **UI.PieChart** node.

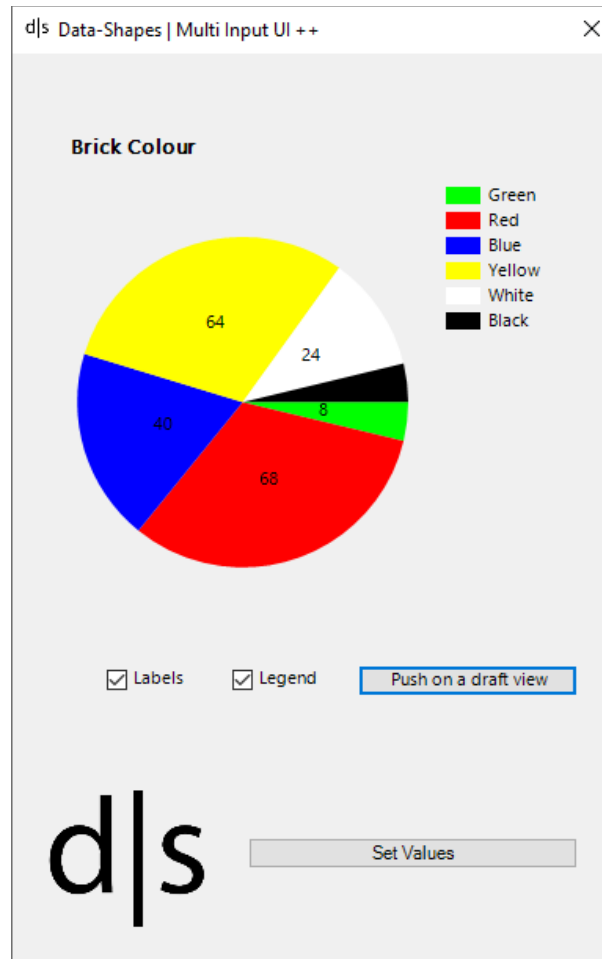


13. The last node we need to add is a **Boolean** node to connect to the *Toggle* input the **UI.MultipleInputForm ++** node. Ensure that the **Boolean** node is set to: **True**.

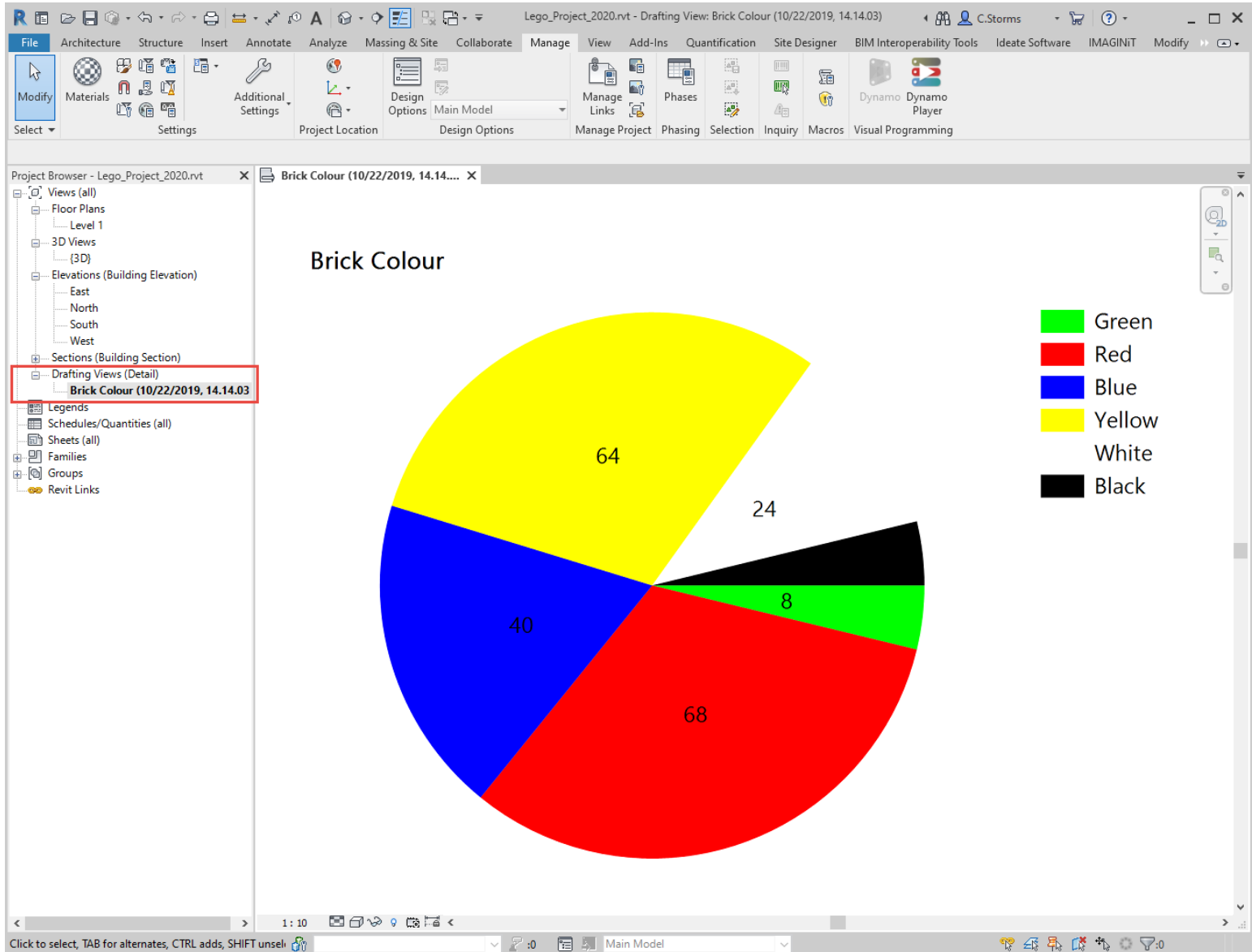




14. Next we with F5 to run the graph and see what the outcome will be.



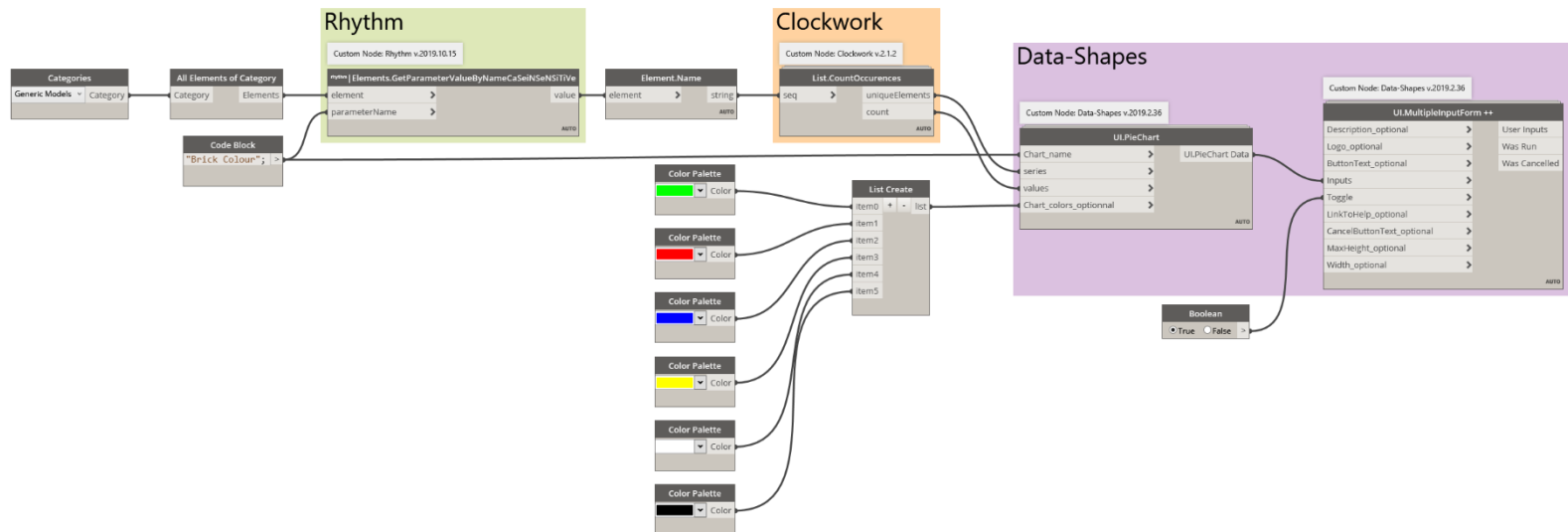
15. Notice that you can turn on or off the **Labels** and **Legend** options with the check boxes. If you hover over a section of the pie you will get a tooltip that tells you that sections percentage of the complete pie. You also have the option to hit the ***Push on a draft view*** button to save an image of the pie chart onto a newly created Drafting View within your Revit model. If you hit the **Set Values** button the ***Data-Shapes*** Pie Chart pop-up will close.



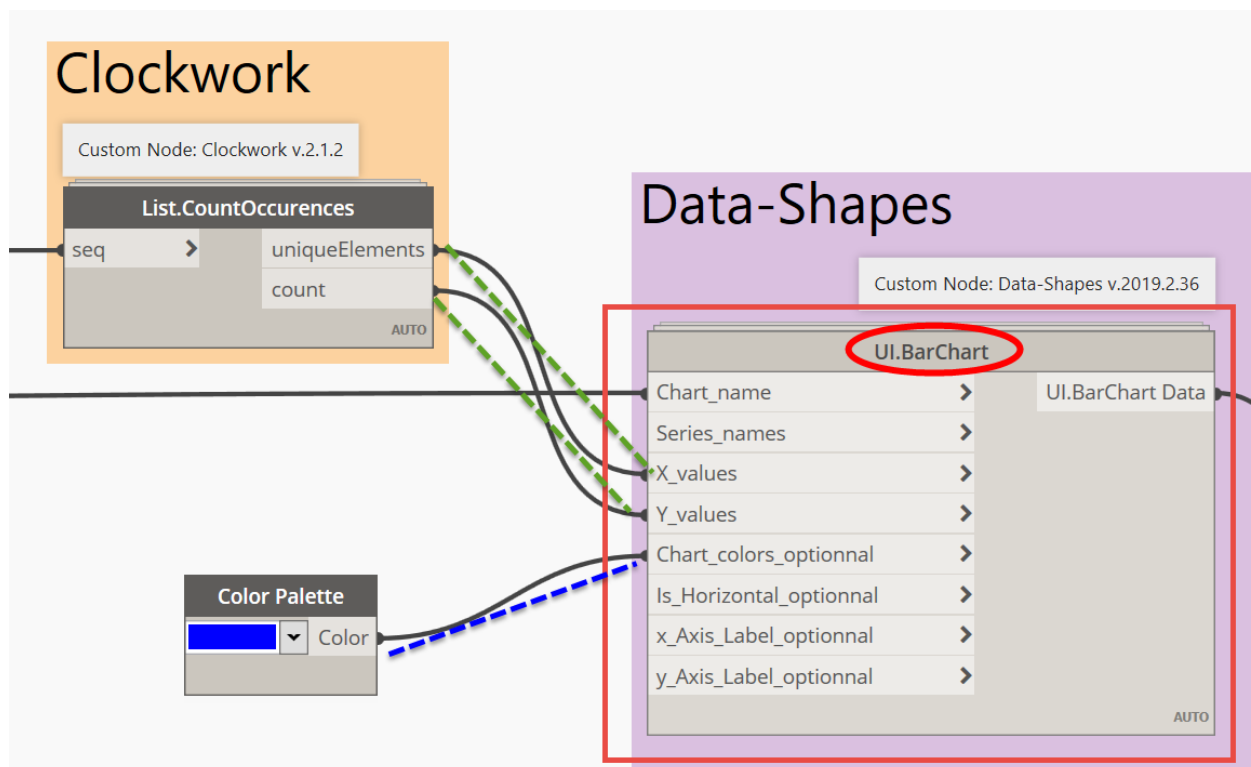
## Bonus Task

For this bonus task we will create a quick Bar Chart to replace our Pie Chart from the previous tasks.

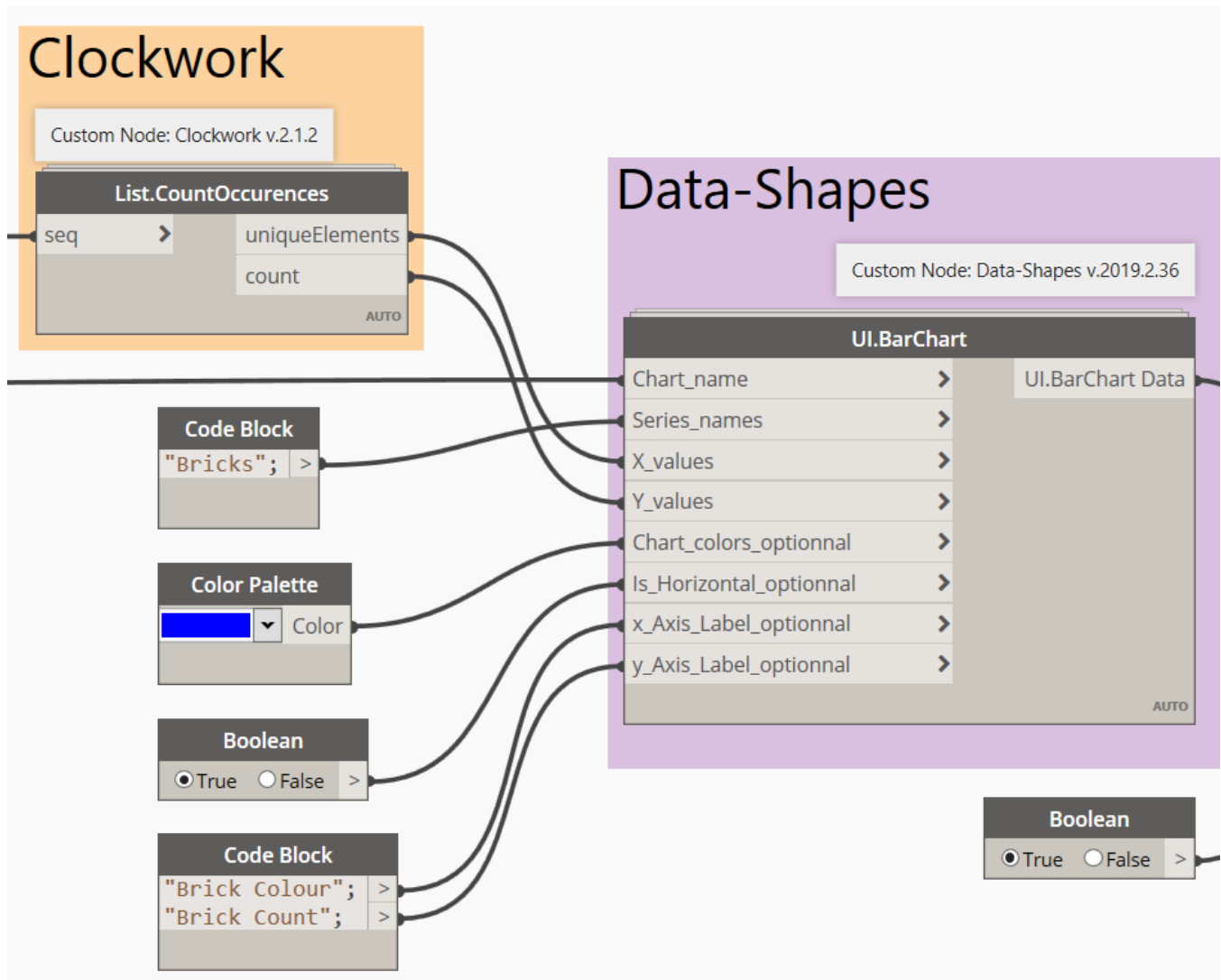
1. For this task, you can continue to work with the Revit file and Dynamo graph from task 1, or you can open the **Lego\_Project.rvt** Revit file, and then open the Dynamo file **Ex4b\_Pie\_TB1\_Begin.dyn** file from the **Ex\_4b** folder in the lab dataset.



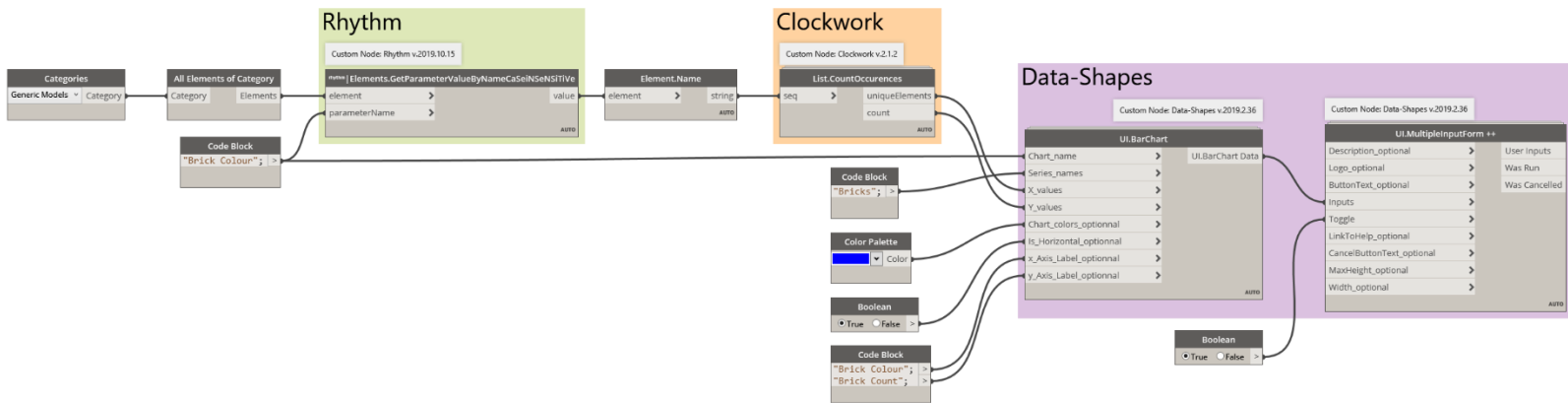
2. We can start by deleting the **UI.PieChart** node, and replacing it with the **UI.BarChart** node. We can also delete the **List.Create** node, as well as all the **Color Palette** nodes, just leaving the blue one, which will plug into the **Chart\_colors\_optionnal** input of the **UI.BarChart** node. The outputs from the **List.CountOccurrences** node can be plugged into the **X\_values** and **Y\_values** of the **UI.BarChart** node as shown in the following image.



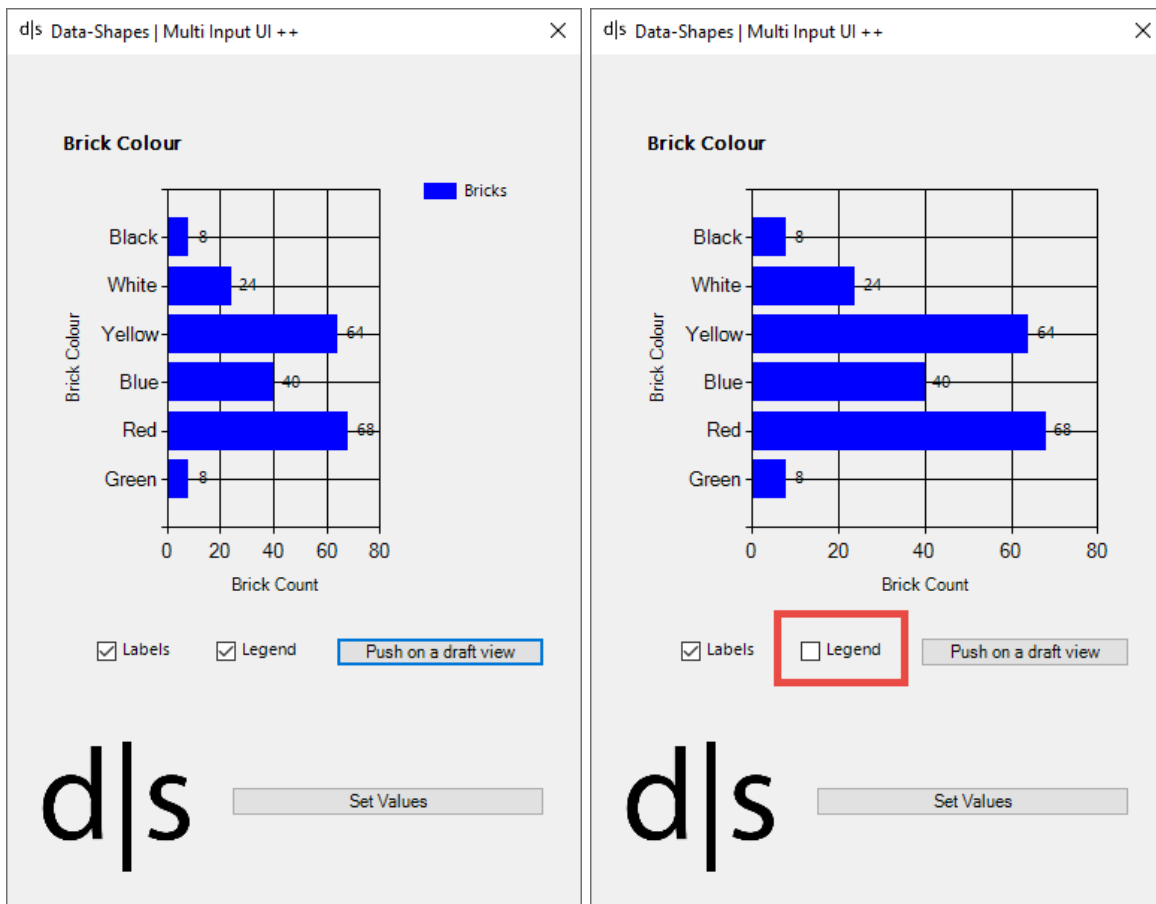
- To finish off this graph we need to add two **Code Blocks**, and a **Boolean** node (set to **True**) One **Code Block** will contain “Bricks”; the other **Code Block** will contain “Brick Colour”; “Brick Count”; These nodes should be connected to the **UI.BarChart** node as shown in the following image.



- Last we need to connect the existing **Code Block** that contains “Brick Colour”; to the **Chart\_name** input of the **UI.BarChart** node.



5. Now we are ready to run the graph and see what the Bar Chart looks like.



Legend on

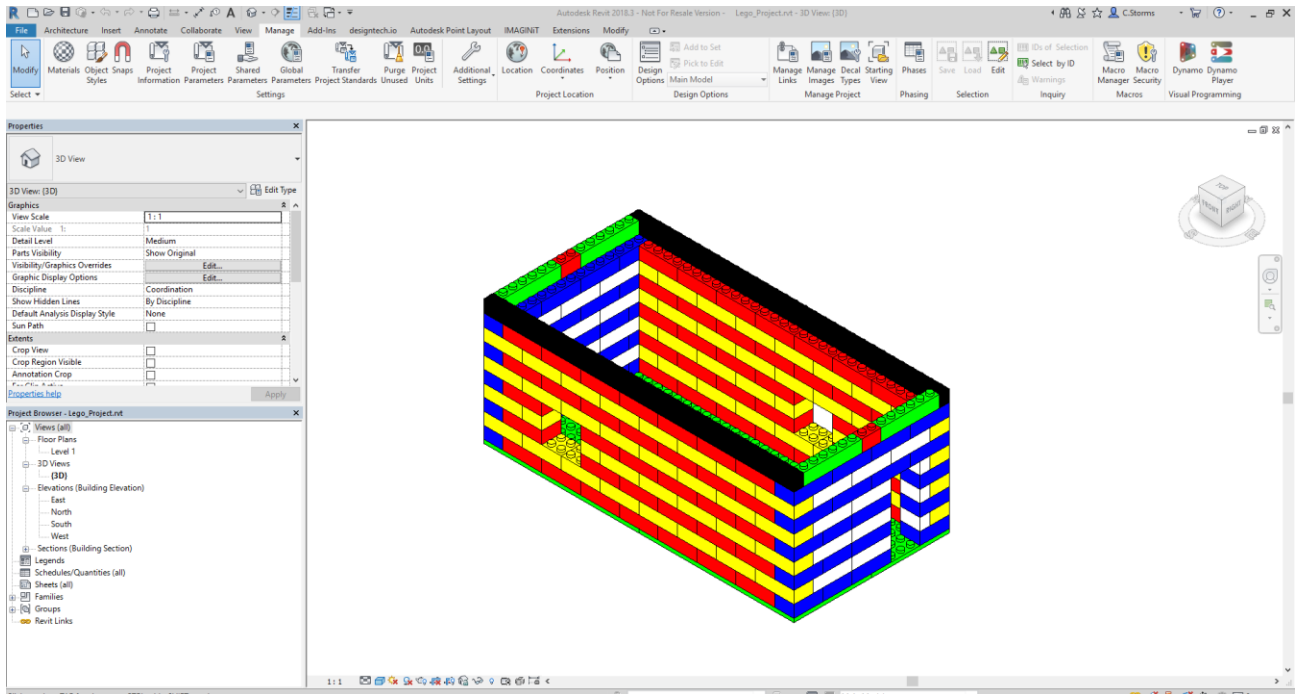
Legend off

## Ex 4c) Donut - Mandrill

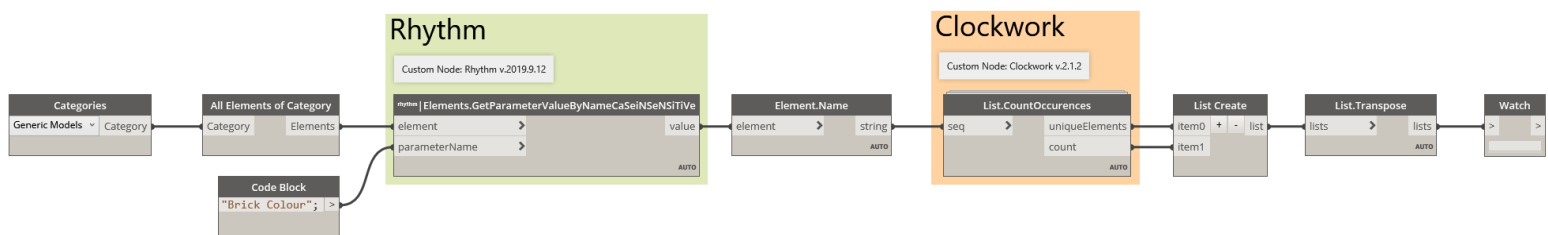
In this exercise we will extract some data from our Revit project and display it visually using the custom [Mandrill](#) package by [Konrad K Sobon](#) from the [archi+lab](#) blog.

### Task 1

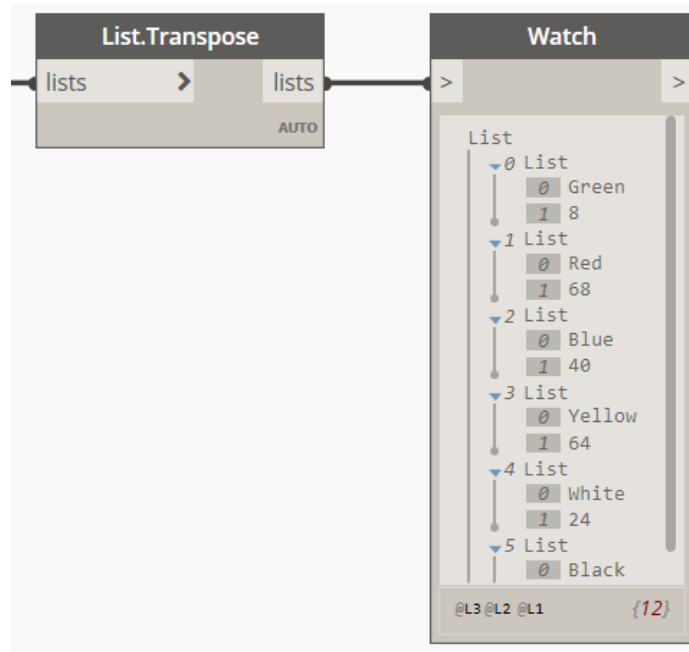
1. From the dataset folder **Ex\_4c** open the following file: **Lego\_Project.rvt** in Revit.



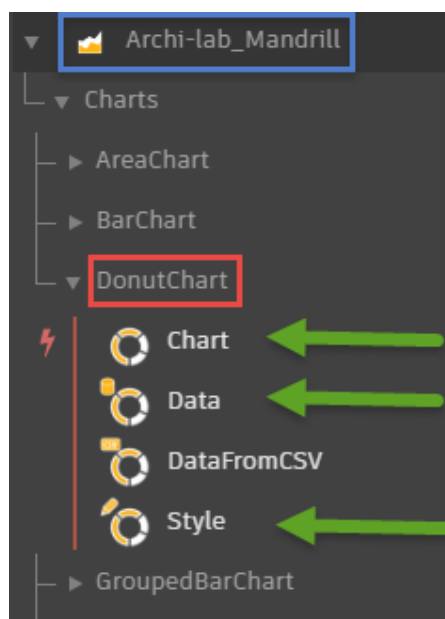
2. Start Dynamo from the Manage tab.
3. Now Open the Dynamo graph called **Ex4c\_Donut\_T1\_Begin.dyn** from the dataset folder **Ex\_4c**.



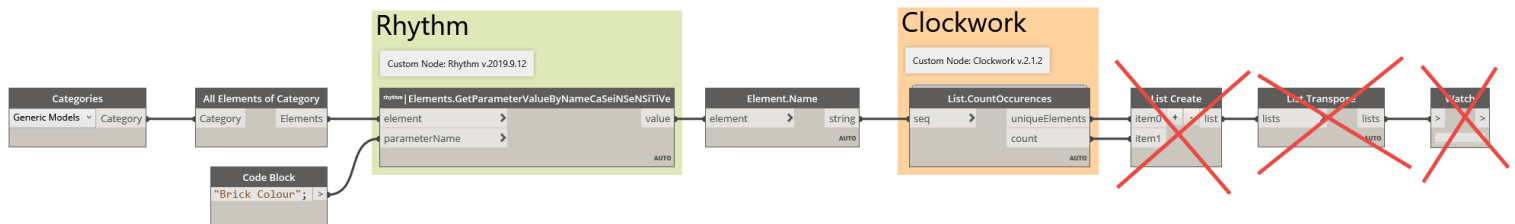
4. Hit the Run button to see the information we get, with this graph as is.
5. The final output is a list, of six lists, each with two items in it. We can now see each colour, and the number of Lego bricks that are in that colour.



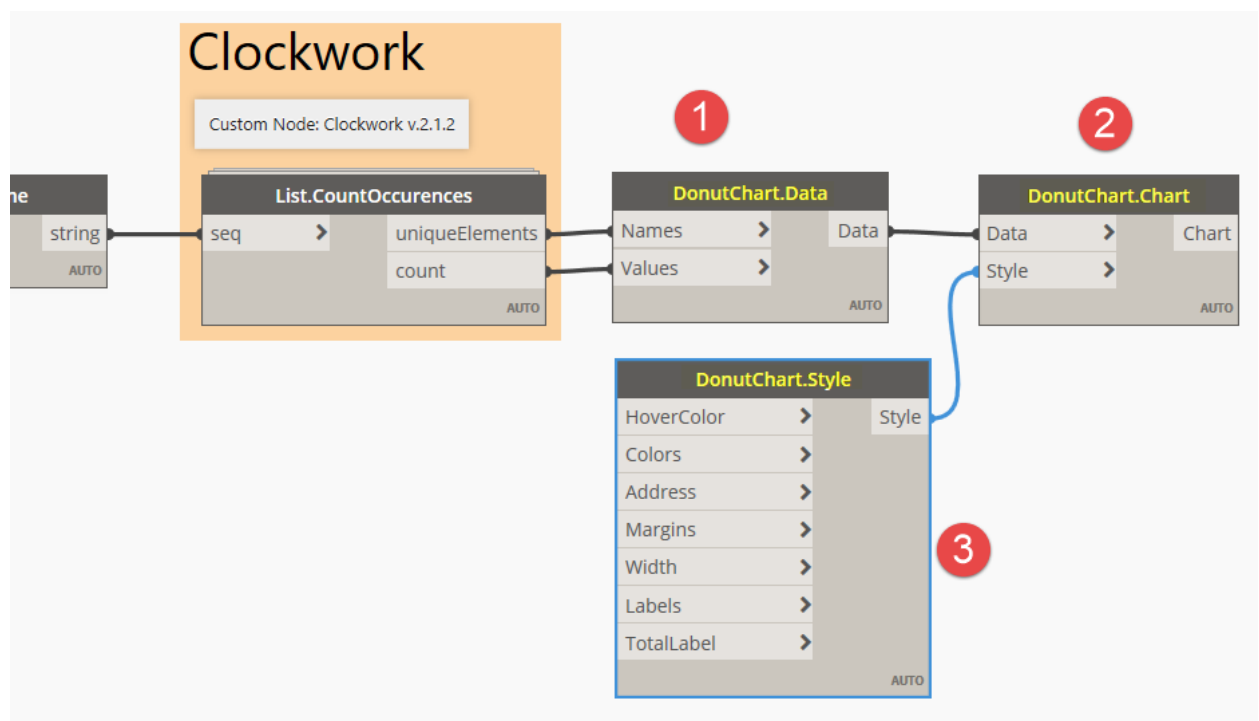
6. This gives us the information we wanted, how many bricks there are of each colour, however the **Watch** node is not the best way to visualize this information. By adding five nodes from the **Mandrill** custom package we can make this data POP!
7. We are going to make a Donut Chart using the **Archi-lab\_Mandrill** package nodes. The first three nodes we will need are: **DonutChart.Chart**, **DonutChart.Data**, and **DonutChart.Style**. We can find these nodes under Archi-lab\_Mandrill > Charts > DonutChart.



8. Before we can connect these nodes to our graph, we need to first delete three of the existing nodes from the end of the graph: **List Create**, **List Transpose** and **Watch**.

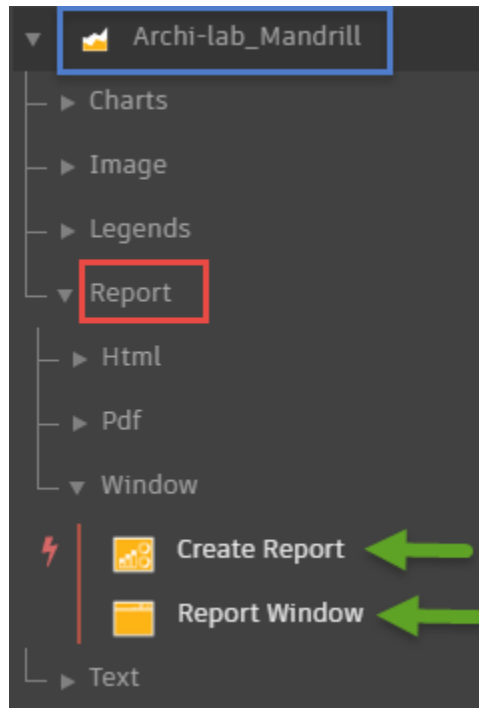


9. Now we can connect our new nodes to our graph as shown in the following image. **Note:** no inputs are needed for the **DonutChart.Style** node, as we will be using all the defaults for this task.

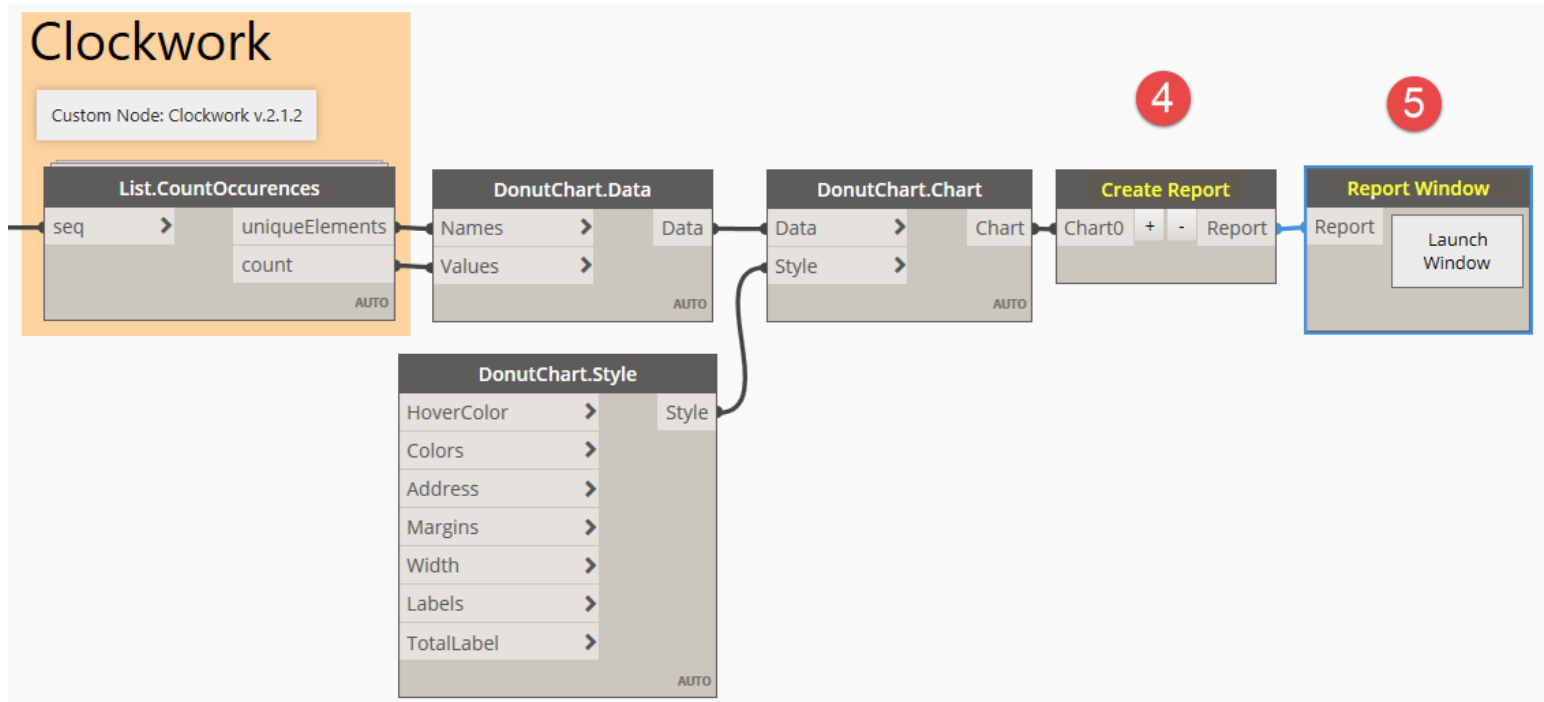


10. Next, we can add the final two **Archi-lab\_Mandrill** nodes to our canvas: **Create Report** and **Report Window**. These nodes can be found under Archi-lab\_Mandrill > Report > Window.

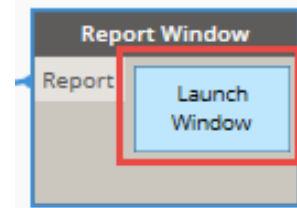




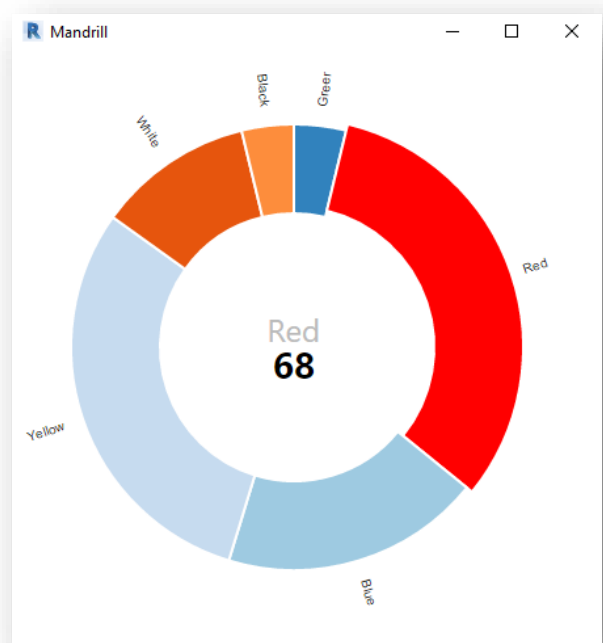
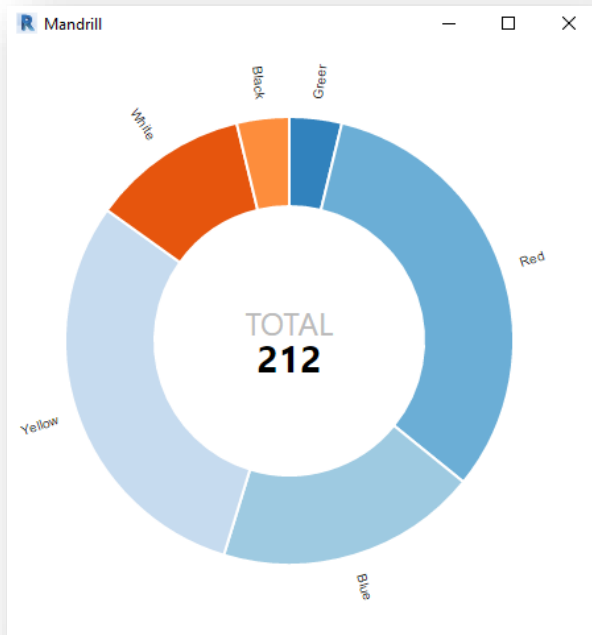
11. Add these two nodes to the end of our graph in this order: **Create Report** then **Report Window**, next hit F5 to run the graph.



12. Even though we ran the graph and it took a while for the graph to complete we don't see anything on the screen, no visualization? To see the visualization, we need to **Launch Window** by selecting those words in the **Report Window** node.



13. The result is a new window opens with a Donut chart that shows our data in it. The colours are on the outside and the total number of bricks in the center. If you hover your mouse over one of the colours it turns red and you see the total number of bricks in that colour displayed in the center of the donut. **Note:** you may need to adjust the size of the new window that opens to make sure all the chart is visible.

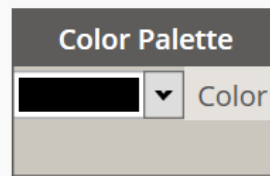


## Task 2

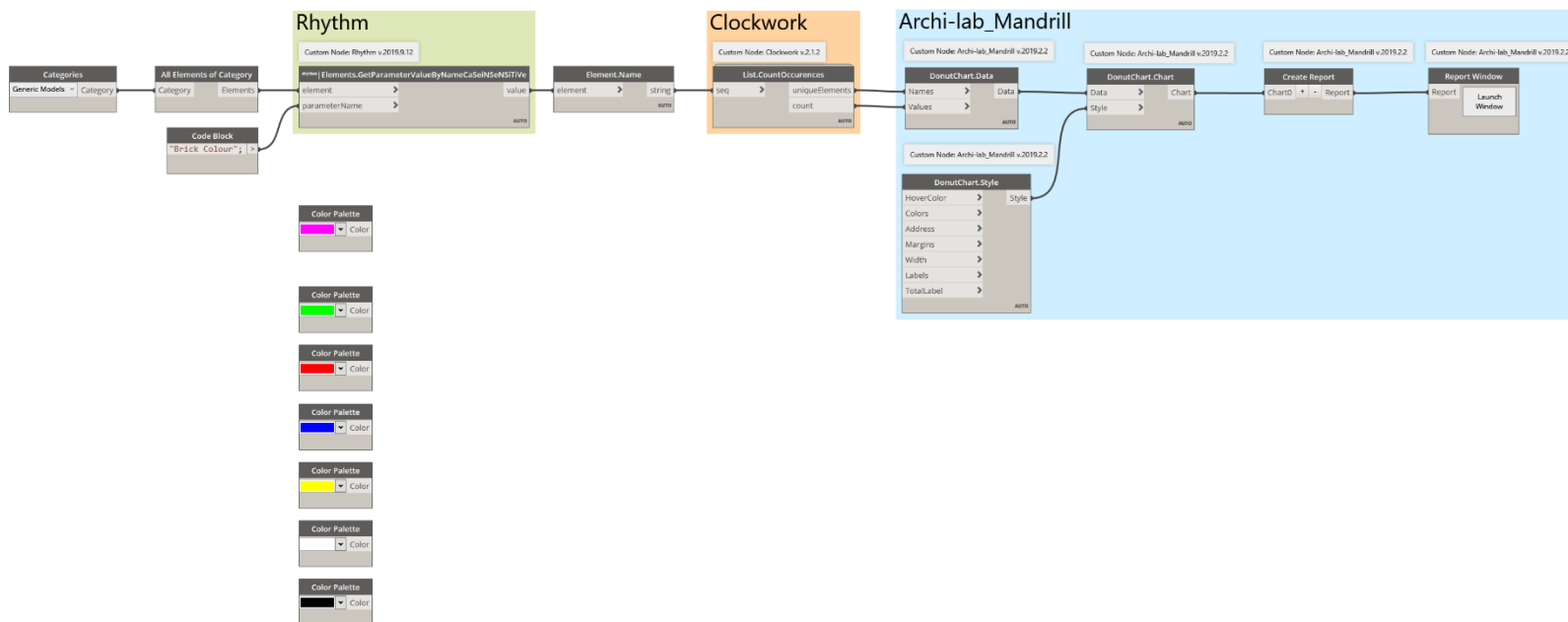
In this task we will add some style to our Donut Chart from task 1.

1. For this task, you can continue to work with the Revit file and Dynamo graph from task 1, or you can open the **Lego\_Project.rvt** Revit file, and then open the Dynamo file **Ex4c\_Donut\_T2\_Begin.dyn** file from the **Ex\_4c** folder in the lab dataset.

- We will start by adding the **Color Palette** node, we will need seven of them on the canvas. These will give us a way to pick the colours we want to be displayed for each one of our data categories (brick colours) and the *HoverColor*.



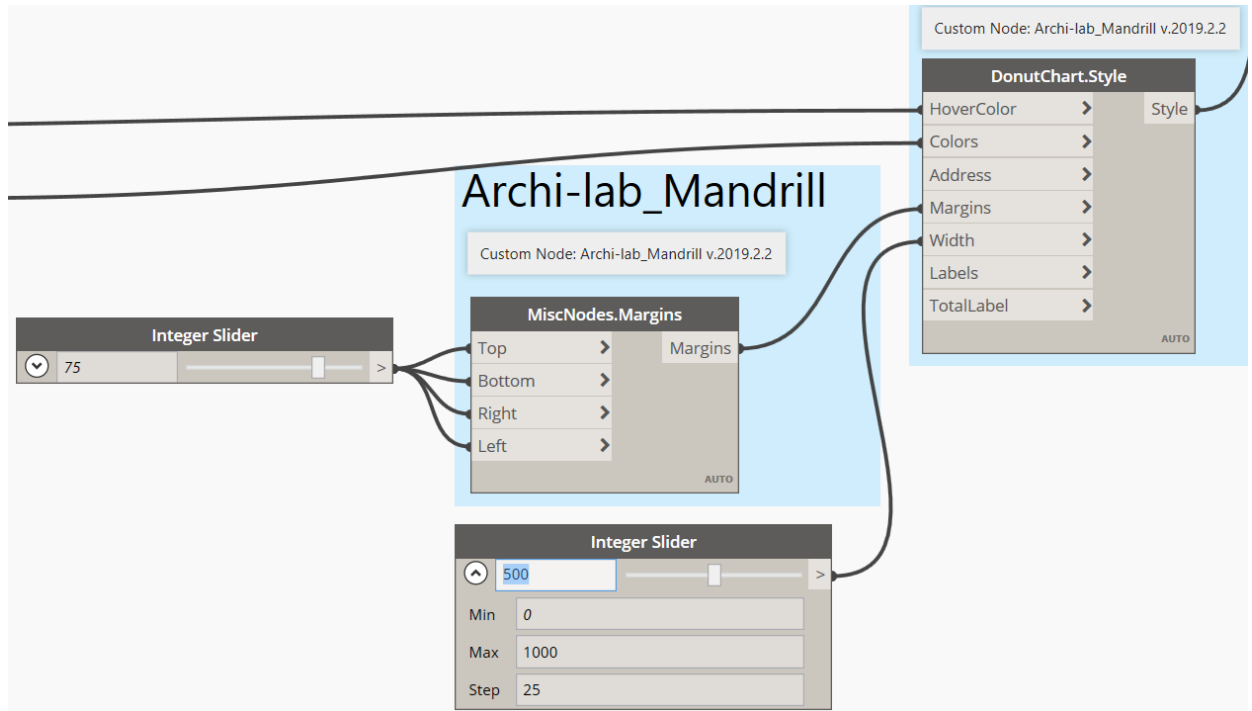
- Once you have those seven nodes on the canvas stack them in a vertical column. Then starting from the top working your way down pick the following colours: **Pink**, **Green**, **Red**, **Blue**, **Yellow**, **White**, **Black**.



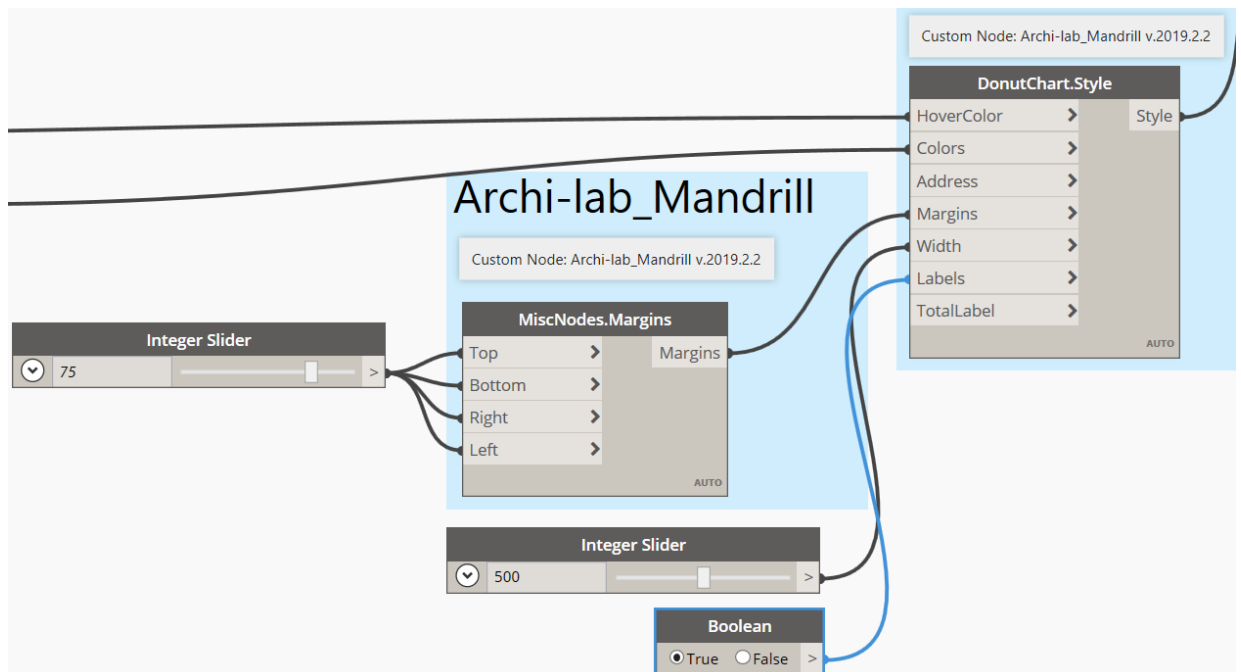
- The Pink colour will connect to the *HoverColor* input in the **DonutChart.Style** node. For the other six colours we need to create a list, we can do this by adding a **List Create** node on the canvas and hitting the “+” sign five times to make six inputs. Then we connect the **List Create** output to the *Colors* input in the **DonutChart.Style** node.

- 
- The screenshot displays the 'Archi-lab\_Mandrill' node, which is a custom node version 2019.2.2. The node is connected to an 'Integer Slider' and a 'MiscNodes.Margins' node. The 'Integer Slider' is set to 75, with a minimum of 0 and a maximum of 100. The 'MiscNodes.Margins' node is connected to the 'Archi-lab\_Mandrill' node, and its 'Margins' property is set to 'AUTO'. The 'Archi-lab\_Mandrill' node has a 'Style' property set to 'Style'.
- Archi-lab\_Mandrill**  
Custom Node: Archi-lab\_Mandrill v.2019.2.2
- Integer Slider**  
75  
Min: 0  
Max: 100  
Step: 1
- MiscNodes.Margins**  
Top  
Bottom  
Right  
Left  
Margins  
AUTO
- DonutChart.Style**  
HoverColor  
Colors  
Address  
Margins  
Width  
Labels  
TotalLabel  
Style  
AUTO

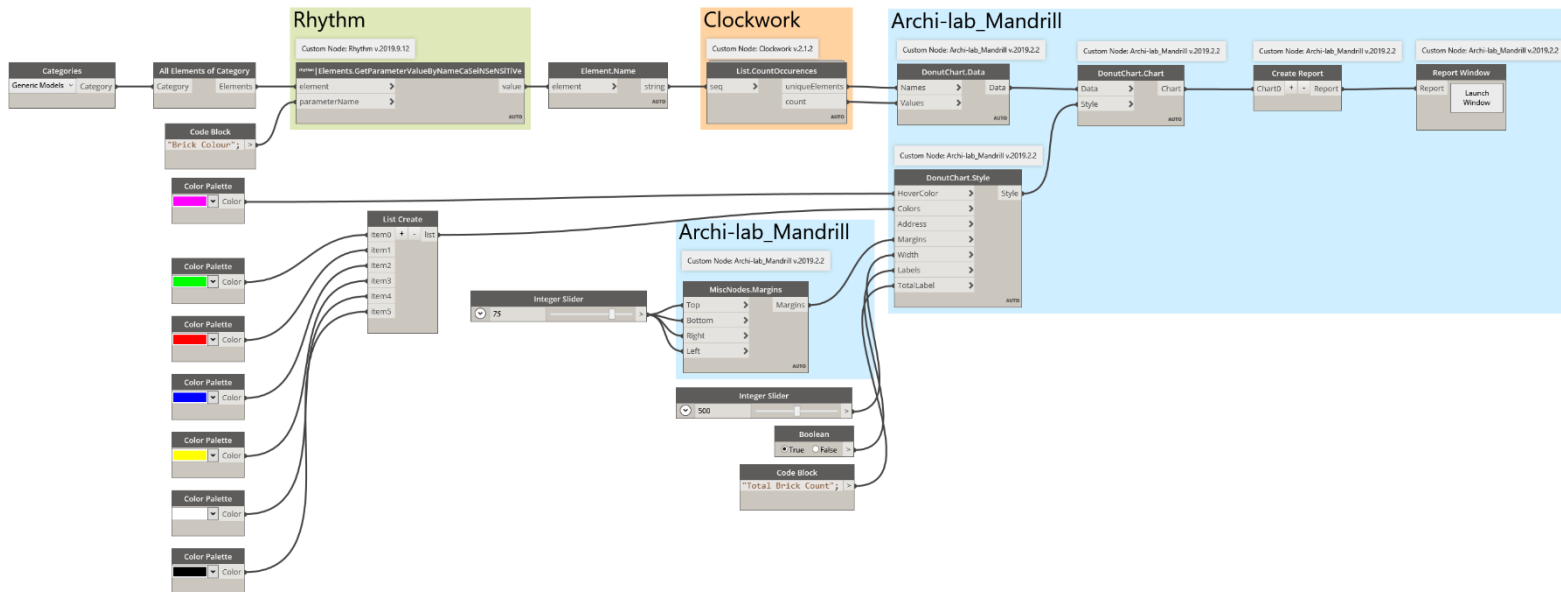
6. For the *Width* input we can copy the **Integer Slider**, and change it's setting to a Max of 1000, and a Step of 25. Then connect it to the *Width* input with a setting of 500.



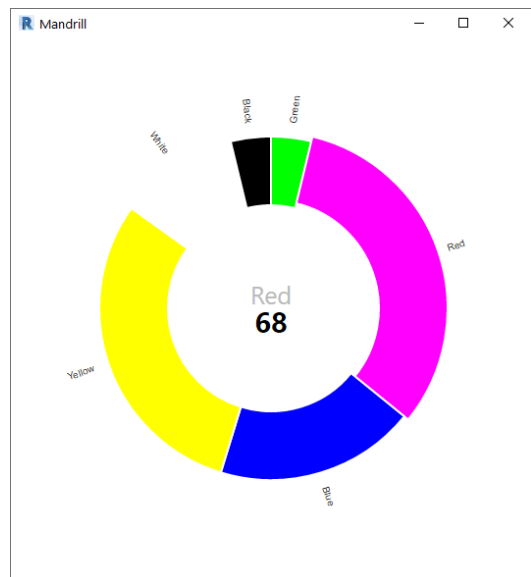
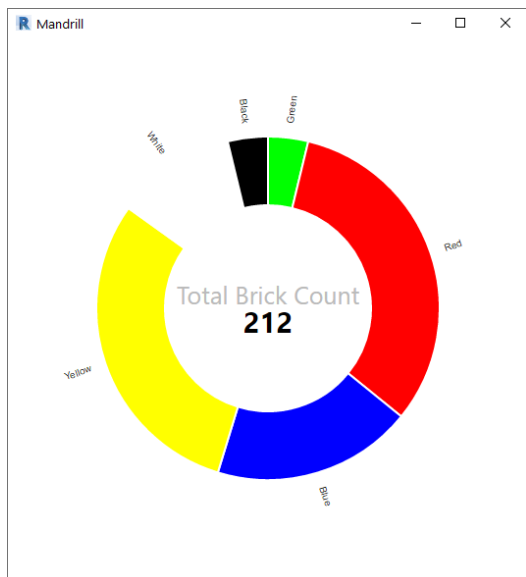
7. The next input *Labels* needs a **Boolean** node that is set to True.



8. The last input *TotalLabel* allows us to pick what the words are inside the Donut Chart. The default is Total, which is okay, but we can make it better with a **Code Block** and adding “Total Brick Count”; inside the **Code Block** before connecting it to the input.



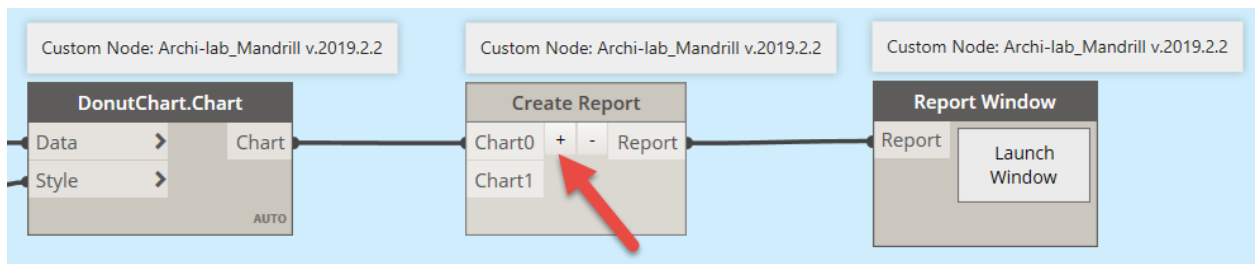
9. The last step is to run the graph, then once it is complete, we hit *Launch Window* within the **Report Window** node and see what our new and improved Donut Chart looks like?
10. The Donut Chart colours now match the colour of the bricks, and when you hover over a brick colour the HoverColor is now pink. The words have changed to Total Brick Count.



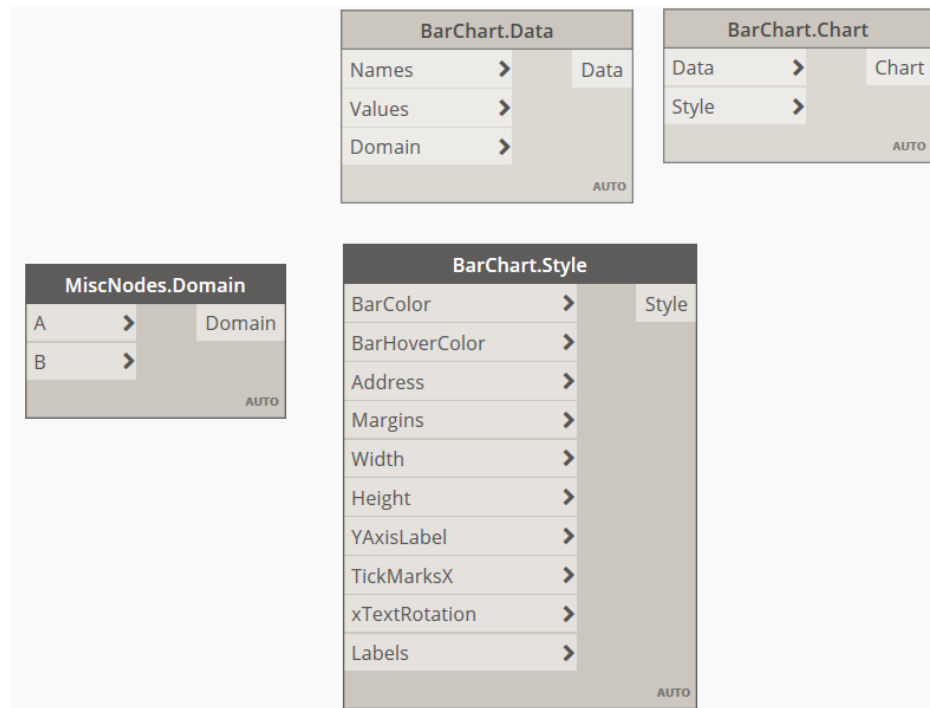
## Bonus Task

For this bonus task we will create a quick Bar Chart to go with our Donut Chart from the previous tasks.

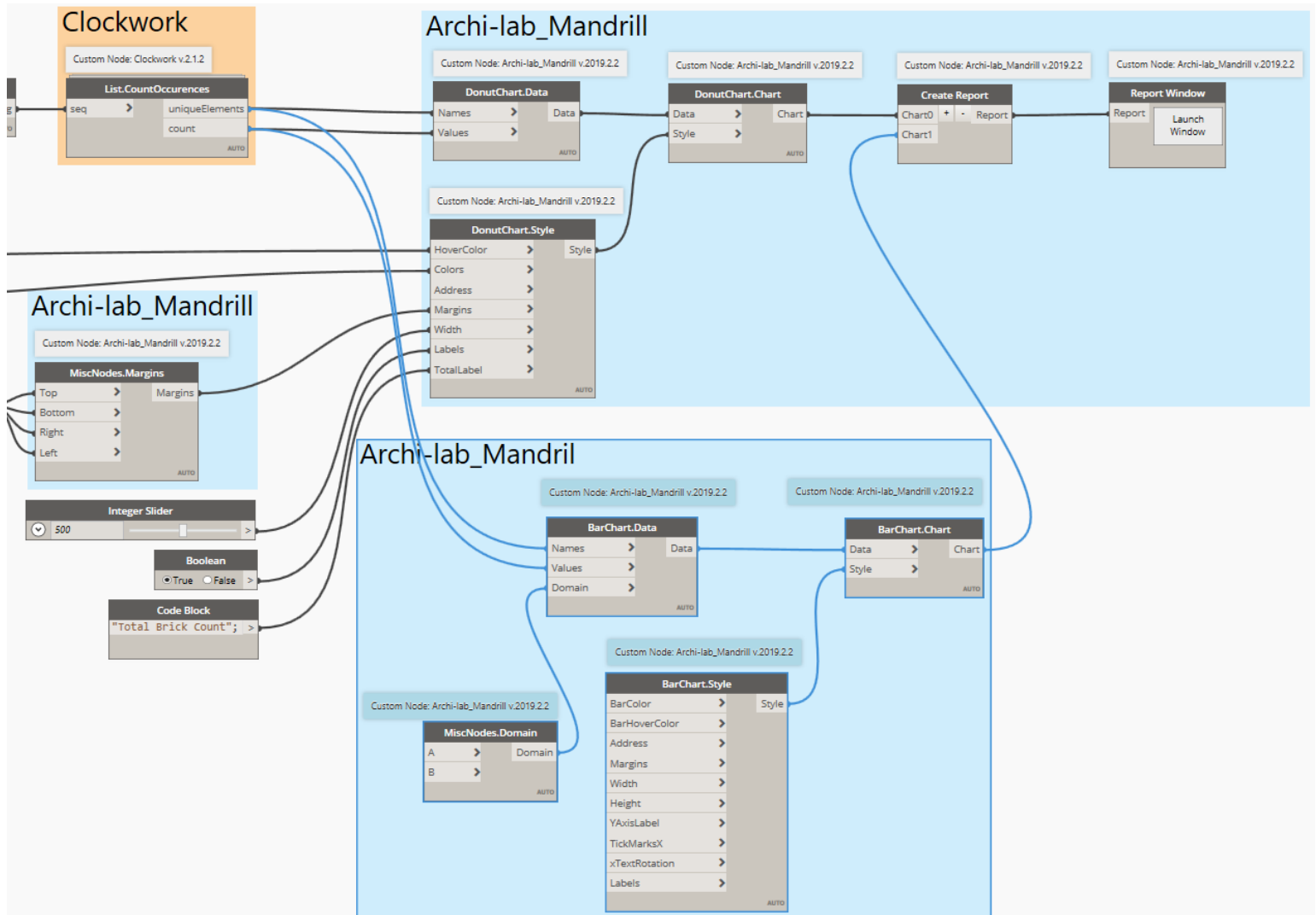
1. For this task, you can continue to work with the Revit file and Dynamo graph from task 2, or you can open the **Lego\_Project.rvt** Revit file, and then open the Dynamo file **Ex4c\_Donut\_TB1\_Begin.dyn** file from the **Ex\_4c** folder in the lab dataset.
2. With five additional nodes, four from the **Mandrill** package and one **Code Block**, plus another input added to the existing **Create Report** node, we can have a Bar Chart as well as a Donut Chart.



3. The four **Mandrill** nodes are: **BarChart.Data**, **BarChart.Chart**, **BarChart.Style** and **MiscNodes.Domain**.

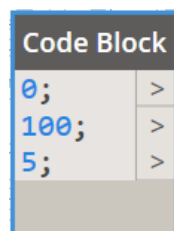


4. Connect the outs from the **List.CountOccurrences** node into the **BarChart.Data** node inputs *Names* (*uniqueElements*) and *Values* (*count*). The **Chart** output from the **BarChart.Chart** node connects to the **Chart1** input of the **Create Report** node. The rest of the connections will be as shown in the following image.



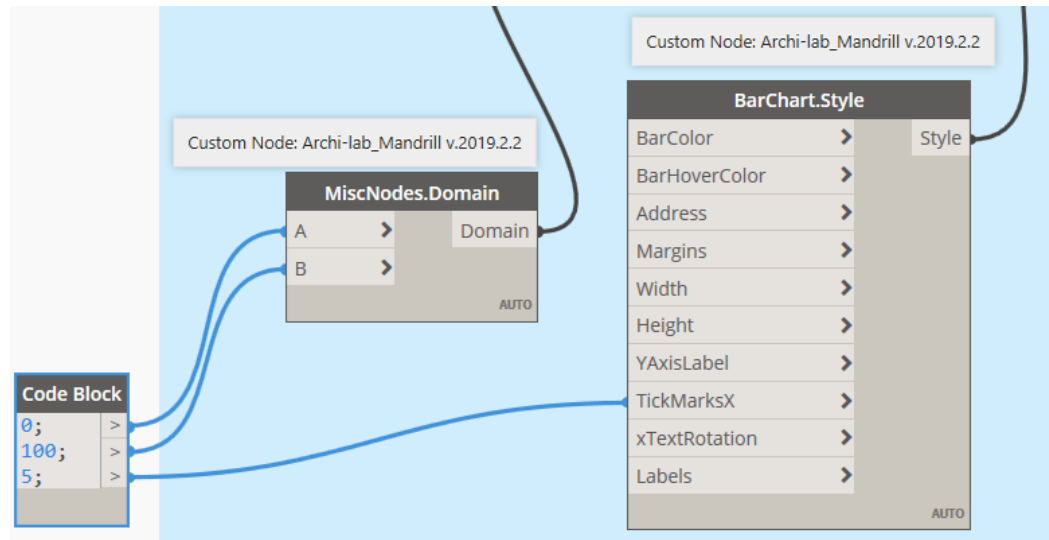
5. Last, we need to add a **Code Block** to the canvas and add the following 3 lines.

- 0;
- 100;
- 5;

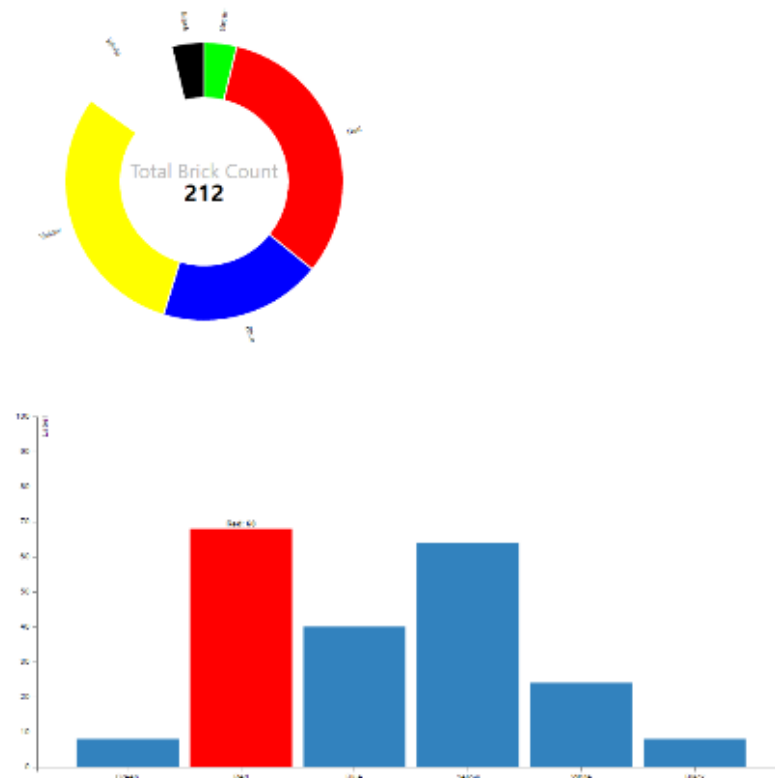




6. Connect the **Code Block** outputs as shown in the following image.

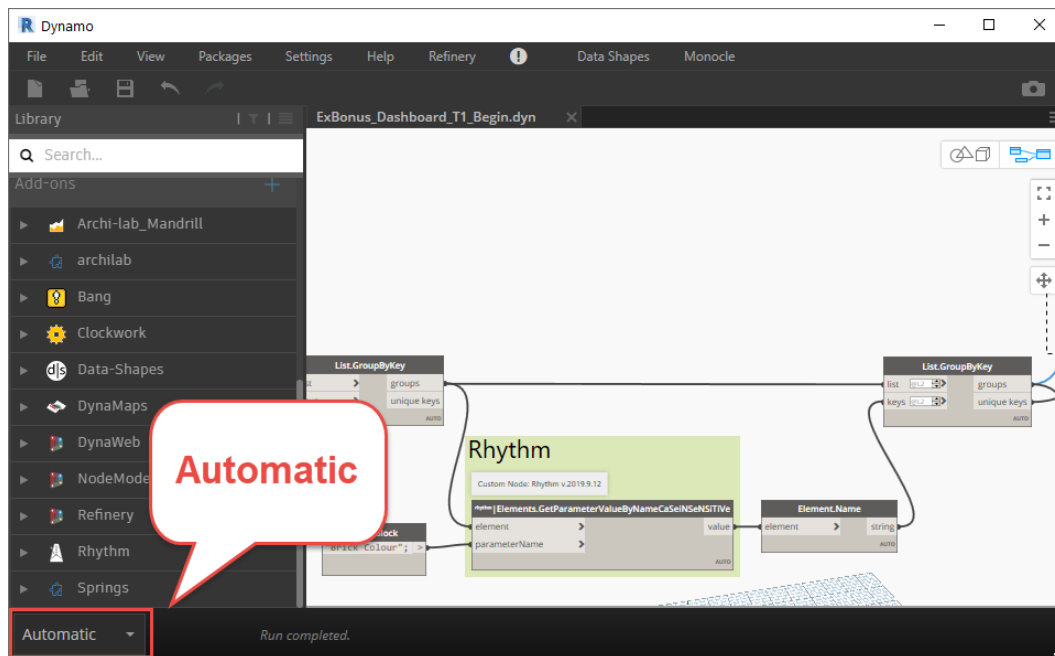


7. The last step is to run the graph, then once it is complete, we hit *Launch Window* within the **Report Window** node and see what our Bar Chart looks like with our Donut Chart.  
**Note:** That if you hover over a bar it will change colour and show its data.

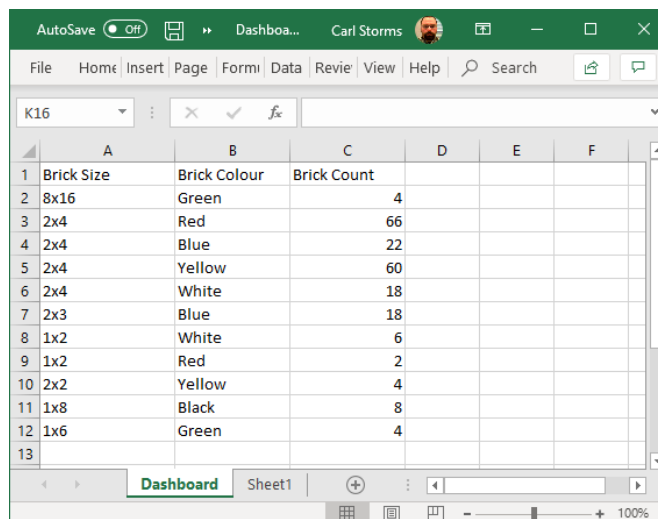




- If this graph looks familiar, it should, as it's from exercise 2. We are going to use this graph to create the Excel data file we will link into our Excel PivotTable Dashboard.  
**Note:** This time the Run feature of the graph is set to automatic, this will allow the live link of data. This also means Excel would have opened when you started the graph.



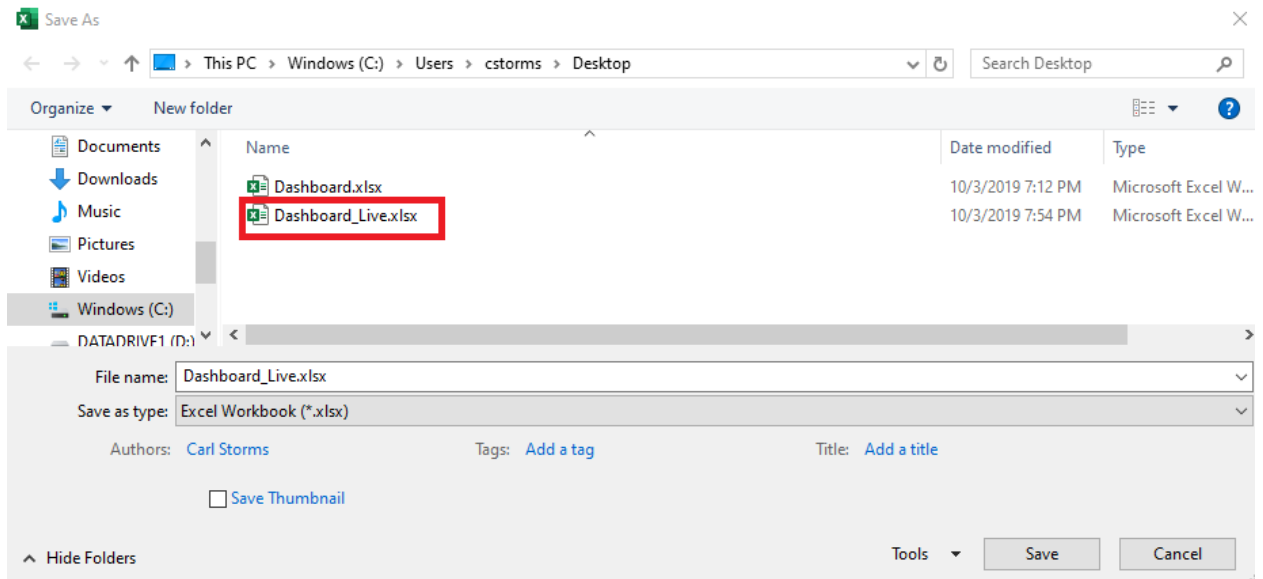
- If the Excel didn't open, you will need to make sure the **File Path** node is set correct (which is likely the case in a lab setting). Click on the **Browse...** button in the **File Path** node, and re-path to the computer desktop. When you do this, save the file using the name: **Dashboard**. Now save the Dynamo graph, then close Dynamo (keep Revit open) and reopen Dynamo and start the graph again (make sure it's the one you just saved).



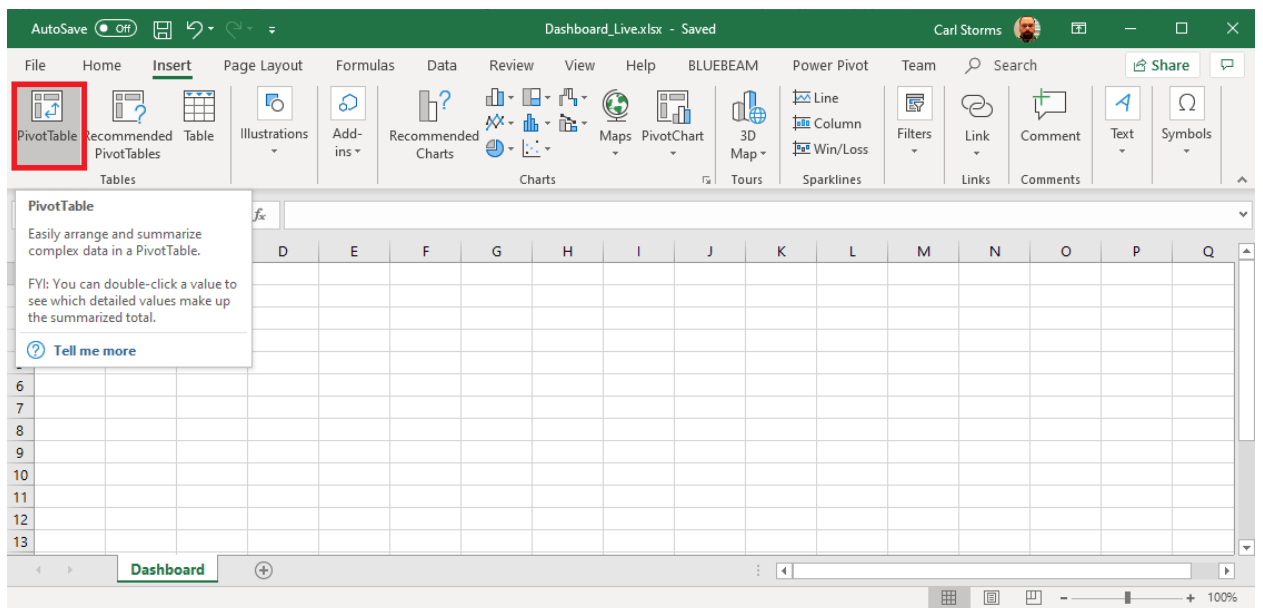
The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	Brick Size	Brick Colour	Brick Count			
2	8x16	Green	4			
3	2x4	Red	66			
4	2x4	Blue	22			
5	2x4	Yellow	60			
6	2x4	White	18			
7	2x3	Blue	18			
8	1x2	White	6			
9	1x2	Red	2			
10	2x2	Yellow	4			
11	1x8	Black	8			
12	1x6	Green	4			
13						

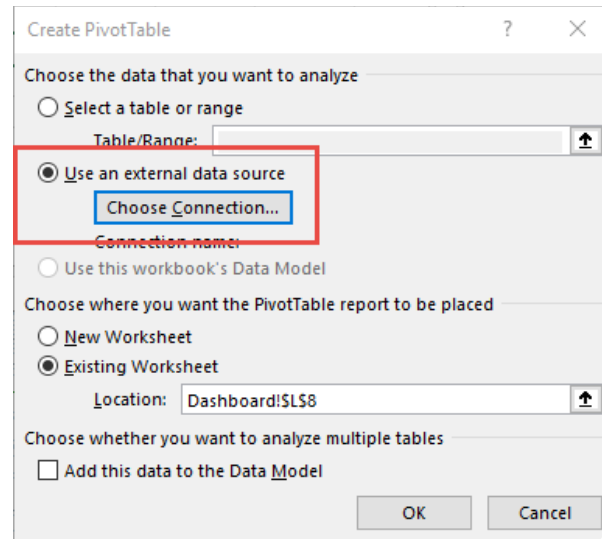
- While leaving Revit, Dynamo, and the Dashboard.xlsx files open, create a new Excel file, naming it **Dashboard\_Live.xlsx**.



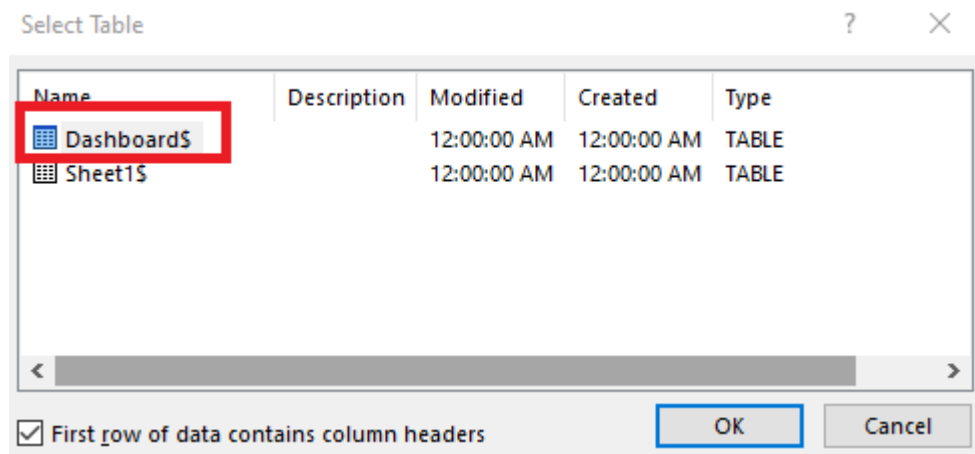
- Now we will create a Pivot table by selecting the PivotTable icon under the Insert tab.



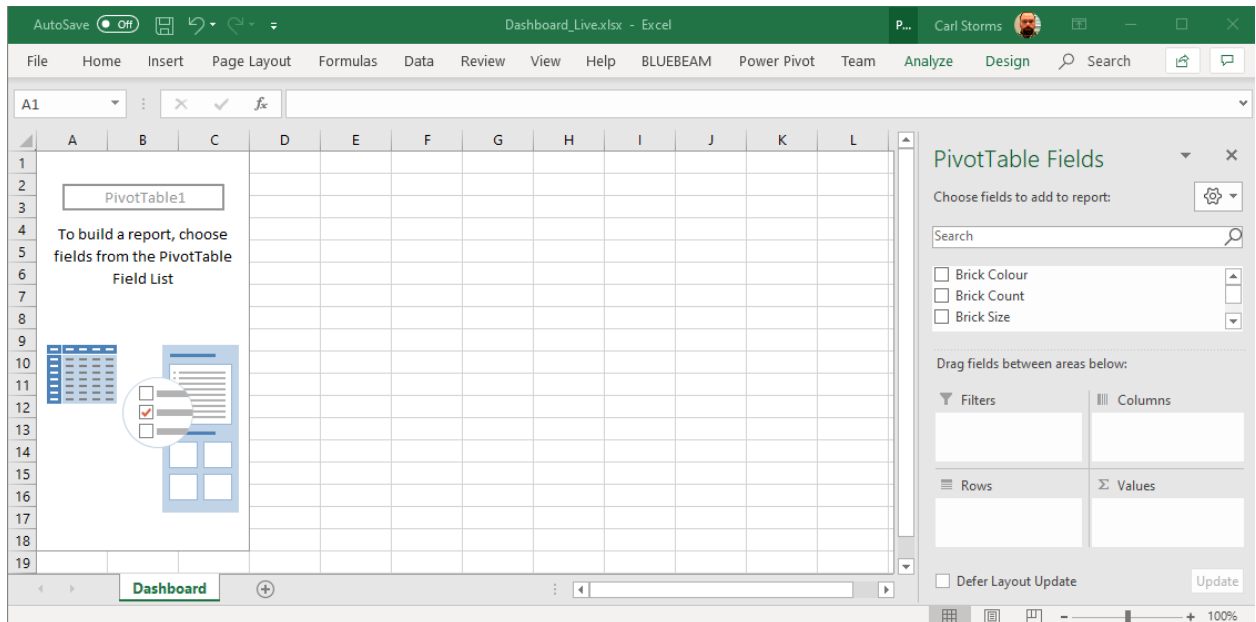
- When the *Create PivotTable* dialog box appears pick the *Use an external data source* option, and then pick **Choose Connection...** to navigate to where you saved the Excel file you just created from Dynamo.



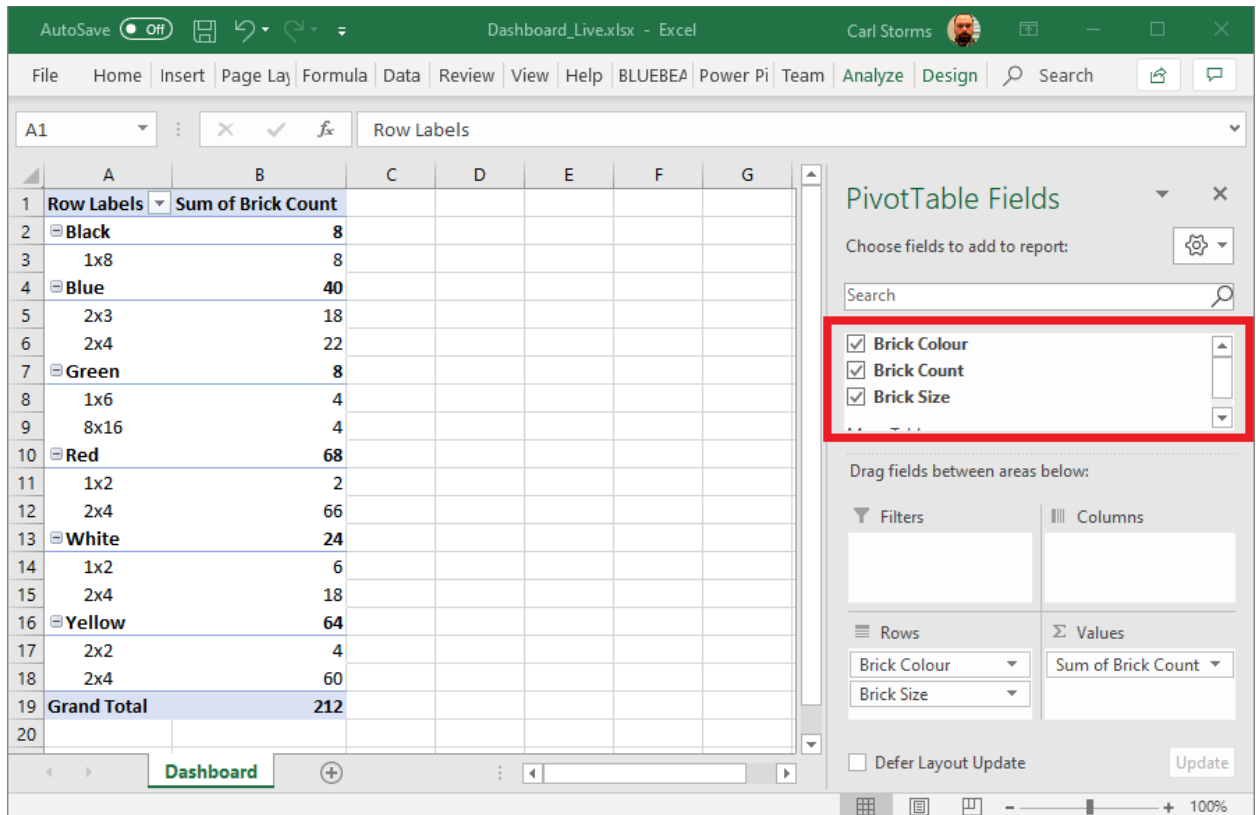
9. When the next dialog box pops up, you will want to select the Dashboard\$ option. If that option is not available hit the *Browse for More...* button on the previous dialog box and navigate to where the Dashboard.xlsx file is located. After selecting that file, you should get the pop-up shown below.



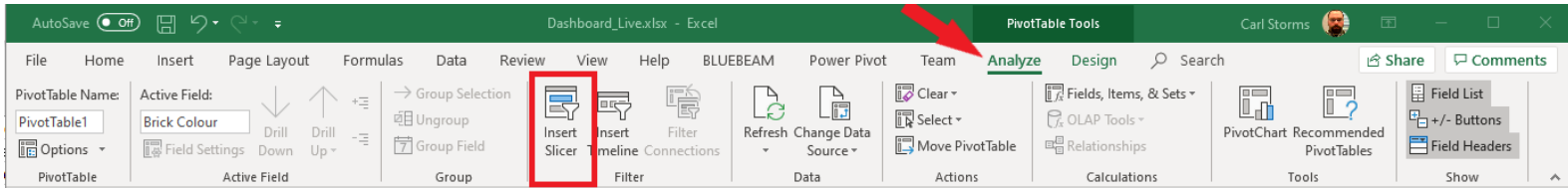
10. Then click OK, and OK again.



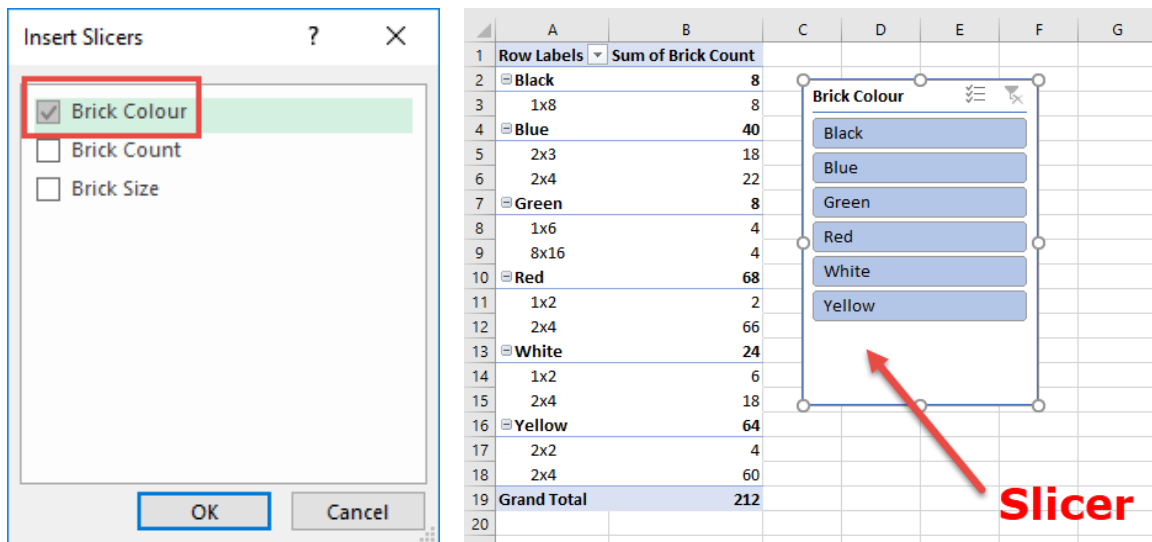
11. Now we can check the three boxes under the PivotTable panel: Brick Colour, Brick Count, and Brick Size, to see how our data looks.



12. Next, we want to add an Insert Slicer. You can find this under the *Analyze* tab in the **PivotTable Tools** Contextual tab.



13. Choose **Brick Colour** from the Insert Slicer dialog box, then hit OK.



14. Now we can add some POP to our Dashboard with PivotTable Charts. With all the cells highlighted choose PivotChart to see the options.

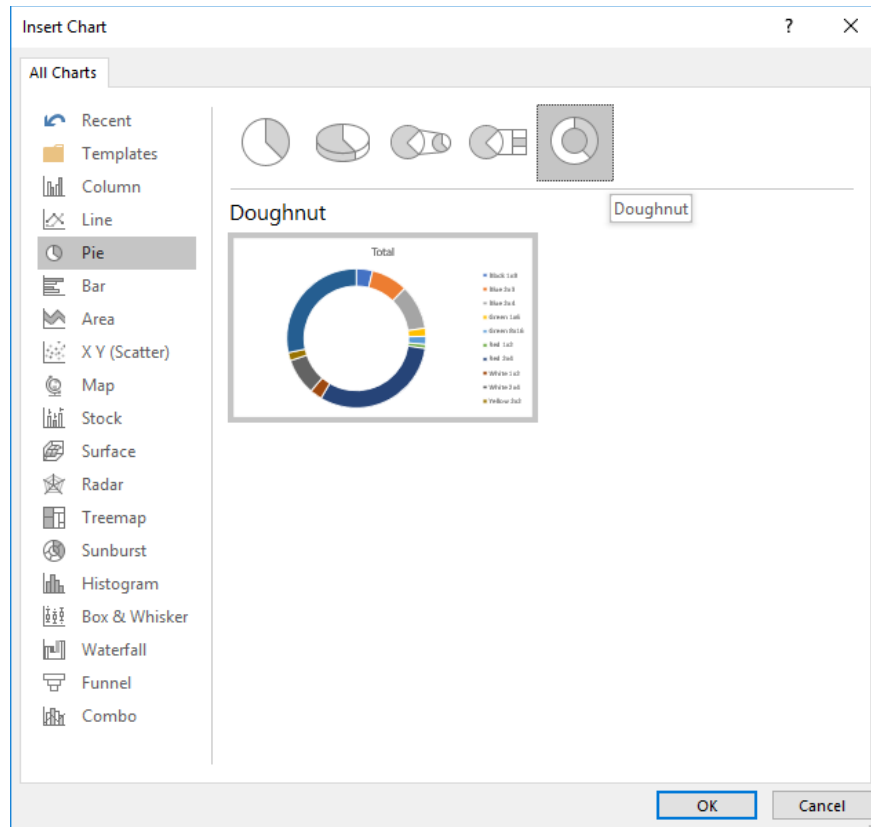
The screenshot shows the Excel interface with the 'Analyze' tab selected. The PivotTable is located in the range A2:B19. The PivotChart is a bar chart showing the count of bricks for each color. The 'PivotChart' button in the 'Tools' group is highlighted with a red box. A tooltip for the PivotChart button is visible, stating: 'Insert a PivotChart tied to the data in this PivotTable.'

Row Labels	Sum of Brick Count
Black	8
1x8	8
Blue	40
2x3	18
2x4	22
Green	8
1x6	4
8x16	4
Red	68
1x2	2
2x4	66
White	24
1x2	6
2x4	18
Yellow	64
2x2	4
2x4	60
Grand Total	212

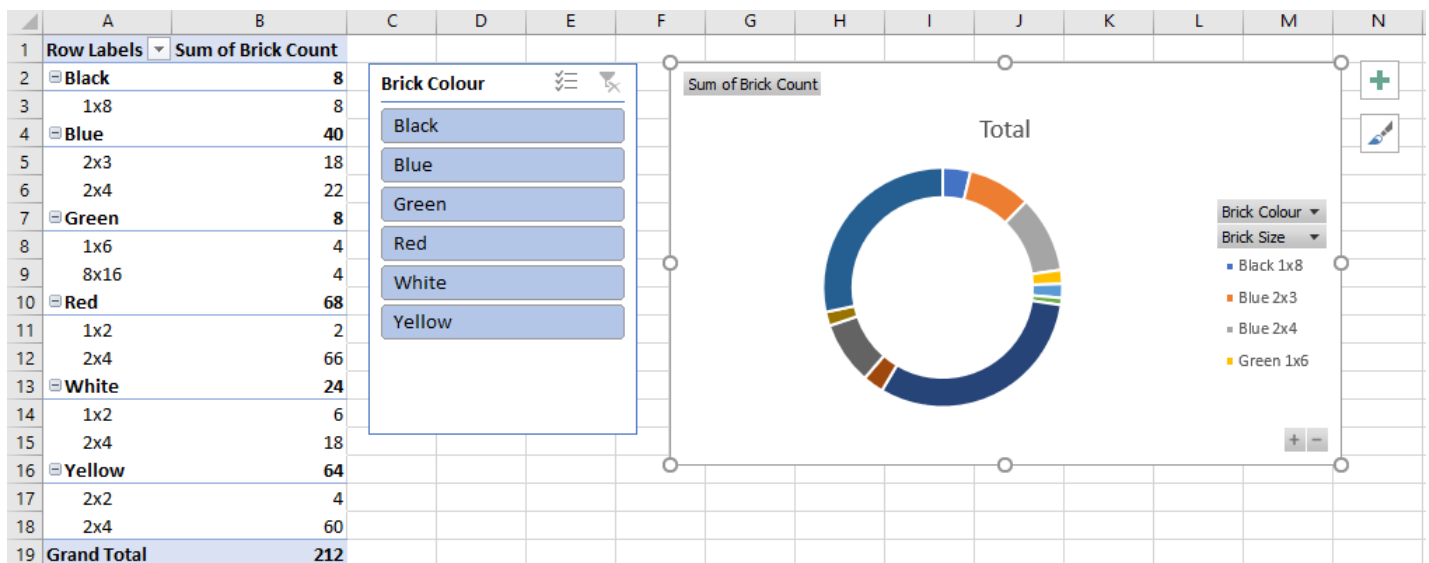
The PivotChart is a bar chart showing the count of bricks for each color. The 'Brick Colour' field is selected for the 'Columns' area, and the 'Sum of Brick Count' field is selected for the 'Values' area. The 'Brick Size' field is selected for the 'Rows' area. The 'Defer Layout Update' checkbox is checked.

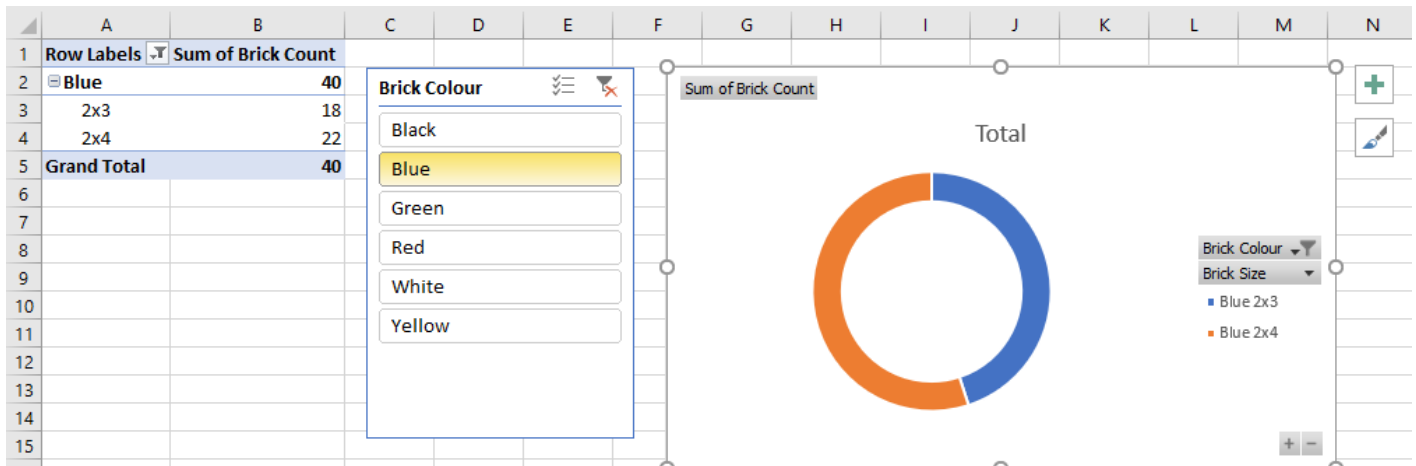
15. Let's do a Doughnut chart like we did before, you can find it under the *Pie* section of the **Insert Chart** dialog box.



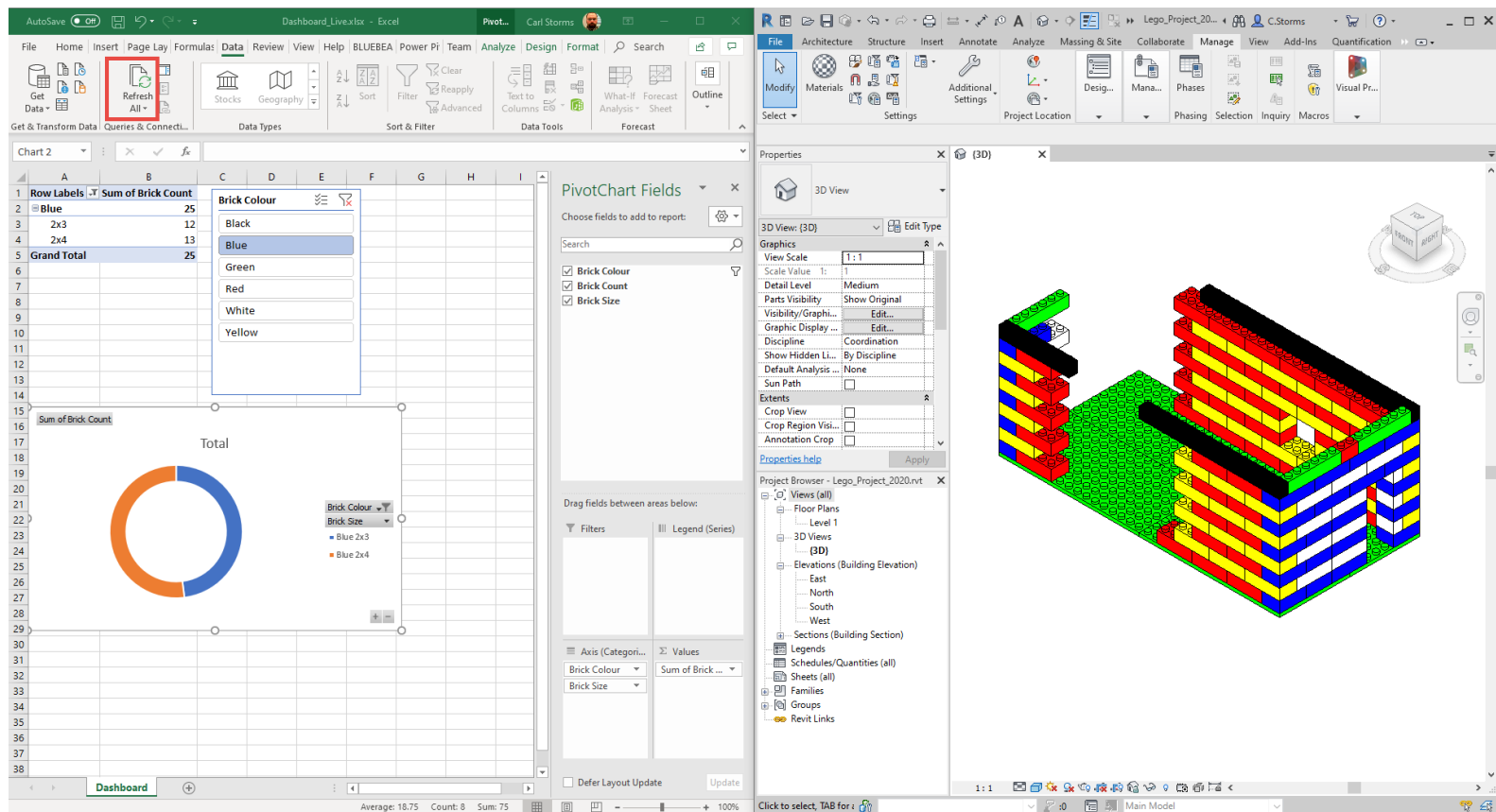


16. Hit OK and position the chart on the screen. Once positioned play with the buttons on the slicer and the see how the information changes.





17. Test the live link, in the Revit file delete some of the bricks. Then in the Dashboard hit the **Refresh All** button under **Data** tab to see the new data.



## Resources

### List@Level

[List@Level](#) blog post from the [Dynamo Blog](#)

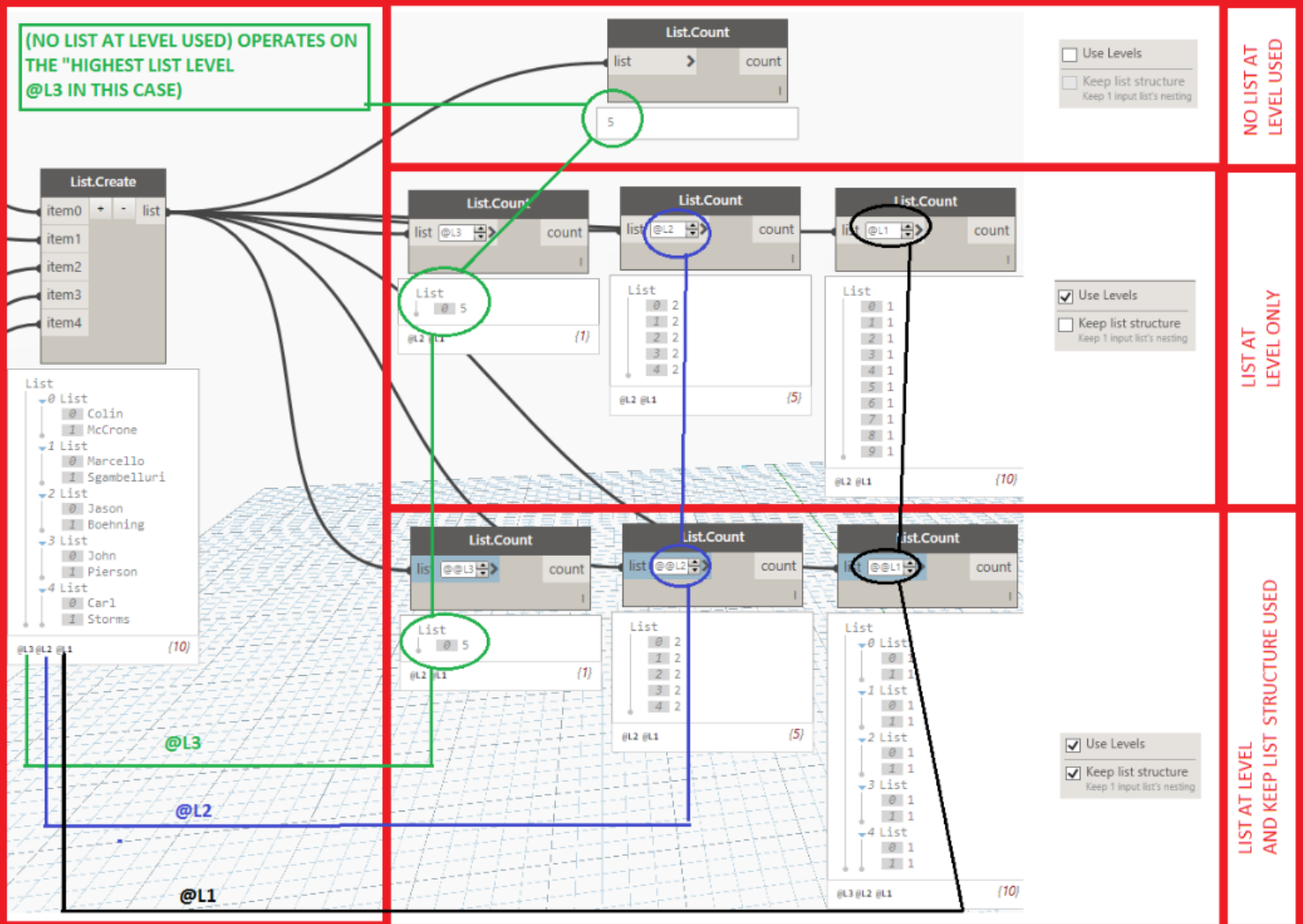
[List@Level](#) from the [Dynamo Primer](#) (starting on page 200 if looking at PDF V2.0)

[List@Level](#) explained by [Colin McCrone](#) on the [Simply Complex Podcast](#).

[List@Level](#) sample image from [Marcello Sgambelluri](#) and [The Simply Complex Blog](#)

## UNDERSTANDING DYNAMO LIST AT LEVEL (SIMPLE LIST COUNT EXAMPLE)

(NO LIST AT LEVEL USED) OPERATES ON THE "HIGHEST LIST LEVEL @L3 IN THIS CASE)



## Past Labs and Presentations

### 2018 Presentation

- Here is a link to the presentation, handout & video on Dynamo Packages from **AU 2018**.
  - [Dynamo Packages: Have You Tried These Yet?](#)

### 2018 Lab

- Here is link to the lab handout, presentation & dataset from **Data Day 2018**.
  - [Dynamo: Everyone's Doing It – Vol. 3 – Attack of the Data!](#)
- Here is link to the lab handout, presentation & dataset from **BILT ANZ 2018**.
  - [Dynamo: Everyone's Doing It – Vol. 3 – Attack of the Data!](#)

### 2017 PDUG Presentation

- Here is a link to a presentation, dataset & video on Dynamo Packages for **PDUG** (this was also done at **BILT NA 2017**).
  - [Dynamo Packages – Have You Tried These Yet?](#)
  - [Dynamo Packages – Have You Tried These Yet? \(BILT NA 2017 version\)](#)

### 2017 Lab

- Here is link to the lab handout, presentation & dataset from **BILT NA 2017**.
  - [Dynamo: Everyone's Doing It – Again!](#)

### 2016 Lab

- Here is link to the lab handout, presentation & dataset from **RTC NA 2016**.
  - [Dynamo: Everyone's Doing It!](#)
- Here is link to the lab handout, presentation & dataset from **AU 2016**.
  - [Dynamo: Everyone's Doing It!](#)

## Blogs & Websites

[AEC. you and me](#) – [Julien Benoit's](#) Blog, also the maker of the SteamNodes package

[ArchSmarter](#) – [Michael Kilkelly's](#) blog about Revit, Dynamo, and Architecture

[archi+lab](#) – [Konrad Sobon's](#) blog with lots of great Dynamo content, examples and packages

[Autodesk Revit Structure](#) – A great blog for structural Dynamo users

[Bad Monkeys](#) – A website for a great group of Dynamo cool cats

[BattleBIM](#) – blog, lots of good Dynamo and Revit content, maker of the BattleBIM package

[BIM 42](#) – [Simon Moreau's](#) blog, lots of good Dynamo and Revit content

[BIM Extension](#) – blog with Dynamo content

[Bimorph](#) – website with Dynamo content, maker of the Bimorph package

[Danimosite](#) – blog with Dynamo content

[DATAISHAPES](#) – [Mostafa El Ayoubi's](#) blog, maker of the Data-Shapes package

[DesignScript Guide](#) – PDF download of the DesignScript Guide

[Designtech](#) - makers of the Zebra package & Designtech extension for Dynamo

[Dynamaniacs of Calgary](#) – Dynamo user group

[Dynamo Blog](#) – The official DynamoBIM.org blog

[Dynamo Dictionary](#) an open source, searchable database for Dynamo functionality.

[Dynamo Learning Resources](#) – An ongoing post in the [Revit Forum](#) about Dynamo

[Dynamo Primer](#) the unofficial online user's manual

[Dynamo Developer Resources](#) – The name says it all

[DynamoNodes](#) – A blog about Dynamo nodes, great place to learn standard & custom nodes

[Enjoy Revit](#) – A blog with some good dynamo content, it's a little older but still helpful

[Github](#) – Where the open source Dynamo fun is

[Havard Vasshaug](#) – [Havard's](#) blog with good (and cool) Dynamo content

[Just Shut Up and do BIM](#) – blog with Dynamo content

[KM](#) – [Kyle Martin's](#) blog with some Dynamo content and uses

[landarchBIM](#) – [Lauren Schmidt's](#) blog, Dynamo and Revit content for landscape architecture

[Parametric Monkey](#) – A website about computational design with Dynamo (and other) tutorials

[Revit AU](#) – blog about Dynamo, C#, Revit & Revit API and lots more.

[Revit Dynamite and Ammo](#) – blog about Dynamo and Python

[Simply Complex](#) – [Marcello Sgambellur's](#) blog about Dynamo, Revit & the AECO+

[sixtysecondrevit](#) – [John Pierson's](#) blog about Dynamo and Revit, also make of Rhythm package

[Stuff and BIMs](#) – [Adam Sheather's](#) blog (creator of DynaWorks) with good Dynamo content

[Ten Commandments for Teaching Dynamo by a Former “Evangelist”](#) – By [Colin McCrone](#)

[The Livingroomcraftz](#) – Lots of interesting posts with Dynamo graphs, & maker of LRCZ package

[The Building Coder](#) – A ton of Dynamo and Coding content

[The Orchid Project](#) – blog, maker of the Orchid package

[The Revit Saver](#) – [Brian Nickel's](#) blog about Revit and Dynamo from a MEP perspective

[WC BIM](#) – blog with Dynamo content

[What Revit Wants](#) – by [Luke Johnson's](#) lots of Revit & Dynamo info + maker of Bakery package

[zhukoven](#) – blog, maker of the zhukoven package

## Courses & Training

[ArchSmarter – Learning Dynamo](#) – This is a paid course created & delivered by [Michael Kilkelly](#)

[ArchSmarter – Managing Building Data With Dynamo](#) – This is a paid course by [Michael Kilkelly](#)

[ArchSmarter – Dynamo for Busy People](#) – This is a paid course created by [Michael Kilkelly](#)

[BIMHIVE](#) – This is a site with free Dynamo tutorials

[CADLearning Dynamo 2016](#) – This is a paid course by [Jason Boehning](#) & [Marcello Sgambelluri](#)

[CADLearning Dynamo 2017](#) – This is a subscription paid course by [Jason Boehning](#)

[CADLearning Dynamo 2018](#) – This is a subscription paid course by [Jason Boehning](#)

[Diving Deeper: A beginners look at Python in Dynamo by Sol Amour](#)

[DiRoots – Dynamo Revit | Basic Free course](#)

[DiRoots – Dynamo Revit | Advanced Free course](#)

[Learning Dynamo](#) – This is a site with free Dynamo tutorials created by [Jeremy Graham](#)

[Lynda.com – Dynamo for Revit: Python Scripting](#) – This is a paid course by [Jeremy Graham](#)

[Lynda.com – Dynamo 1.x Essential Training](#) – This is a subscription paid course by [Ian Siegal](#)

[Lynda.com – Dynamo for Revit Workflow](#) – This is a subscription paid course by [Ian Siegal](#)

[Lynda.com – Dynamo for Revit Project Setup](#) – This is a subscription paid course by [Ian Siegal](#)

[Lynda.com – Dynamo 2.x Essential Training](#) – This is a subscription paid course by [Ian Siegal](#)

[Lynda.com – Paneling with Dynamo for Revit](#) – This is a paid course by [Colin McCrone](#)

[Lynda.com – Dynamo: Practical](#) – This is a subscription paid course by [Paul F Aubin](#)

[Lynda.com – Revit and Dynamo for Interior Design](#) – This is a paid course by [Bill Carney](#)

[Lynda.com – Advanced Revit and Dynamo for Interior Design](#) – Paid course by [Bill Carney](#)

[Plurasight – An Introduction to Dynamo for Daily Use Within Revit](#) – Course by [John Pierson](#)

[Plurasight – Exploring Dynamo Geometry](#) – This is a subscription paid course by [Sol Amour](#)

[ThinkParametric – Dynamo 101 Fundamentals](#) – This is a paid course by [Konrad Sobon](#)

[ThinkParametric – Create Custom View Filters using Dynamo](#) – A paid course by [Konrad Sobon](#)

[Udemy – DY01: Learn to use Dynamo BIM with Spreadsheets](#) – This is a paid course by [KIT To](#)

[Udemy – DY02: Learn to use Dynamo BIM with Spreadsheets and Lists](#) – Course by [KIT To](#)

[Udemy – Autodesk Revit 2019 Dynamo 2.0 Tests](#) – This is a paid course by [Enrique Galicia](#)

[Udemy – BIM Revit 2019 Models Tracking and Site Control](#) – Paid course by [Enrique Galicia](#)

[Udemy – BIM Revit Site Mechanics and Topography with Dynamo](#) – By [Enrique Galicia](#)

[Udemy – BIM Generative Design for Facades Revit 2018, Dynamo 2.0.2](#) – By [Enrique Galicia](#)

[Udemy – BIM Generative Design for Architectural Placement Revit 2018](#) – By [Enrique Galicia](#)

[Udemy – BIM Generative Design for Buildings Dynamo 2.0 and Refinery](#) – By [Enrique Galicia](#)

[Udemy – BIM Revit Dynamo Outputs for Documentation](#) – Paid course by [Enrique Galicia](#)

[Udemy – BIM Autodesk Revit 2018 Dynamo Player for Modeling Vol 1](#) – By [Enrique Galicia](#)

[Udemy – Gaming with BIM Models, 3d Coordination](#) – This is a paid course by [Enrique Galicia](#)

[Udemy – BIM Mastering Geometry on Revit with Dynamo Samples and Uses](#) – This is a paid course by [Enrique Galicia](#)

[Udemy – BIM Revit 2018 Dynamo Player for Modeling MEP Installations](#) – By [Enrique Galicia](#)

[Udemy – BIM Modeling Architecture LOD 300-350 Revit 2018](#) – Paid course by [Enrique Galicia](#)

[Udemy – BIM Revit 2018 Rebar Modeling with Dynamo](#) – Paid course by [Enrique Galicia](#)

[Udemy – Infracore, Civil3d, Revit and Dynamo 2018](#) – This is a course by [Enrique Galicia](#)

[Udemy – BIM Clash Detection with Revit, Navisworks, Dynamo 2018](#) – By [Enrique Galicia](#)

[Udemy – BIM Dynamo Analysis Revit 2018](#) – This is a paid course by [Enrique Galicia](#)

[Udemy – BIM Dynamo Geometry Modeling 2018](#) – This is a paid course by [Enrique Galicia](#)

[Udemy – BIM Modeling Structure LOD 300-350 Revit 2018 and Dynamo](#) – By [Enrique Galicia](#)

[Udemy – BIM 4D Simulations Revit, Dynamo, Navisworks & Max 2018](#) – By [Enrique Galicia](#)



## Free Graphs

[BIMOne – First Dynamo Script Pack by BIM One for Settings](#)

[BIMOne – 2nd pack of Dynamo scripts for model maintenance](#)

[BIMOne – Third pack of free Dynamo scripts](#)

## Podcasts & Videos

[ArchSmarter](#) Video – [How to Create a Project Dashboard using Revit, Dynamo, and Excel](#)

[ArchSmarter's YouTube Channel](#) – there are a few good getting started Dynamo videos here

[Aussie BIM Guru](#) YouTube Channel – Lots of great Dynamo videos and tutorials on this channel

[BIMHIVE](#) YouTube Channel – Dynamo Tutorials

[Danny Bentley's YouTube Channel](#) – Revit API and Dynamo Videos

[Dynamo](#) – Autodesk's Dynamo YouTube Channel

[Dynamo Chicago YouTube Channel](#)

[Dynamo-litia](#) Vimeo Page – Boston Dynamo User Group

[Dynamo Thoughts](#) – A Vodcast about teaching & learning Dynamo by [Bill Debevc](#) and [Ian Siegel](#)

[EvolveLab](#) – YouTube Channel – Dynamo Tutorials and more

[John Pierson's YouTube Channel](#) – Lots of great Dynamo videos from the maker of the Rhythm package and [sixtysecondrevit blog](#)

[Konrad Sobon's YouTube Channel](#) – Creator of [archi+lab](#) and 4 helpful dynamo packages

[Philadelphia Dynamo User Group](#) – YouTube Channel

[RoMBIS](#) YouTube Channel

[Ryan Cameron](#) YouTube Channel

[SF Computational Design Institute](#) YouTube Channel

[Sean Fruin](#) YouTube Channel

[The BIM Coordinator](#) YouTube Channel

[The Simply Complex Podcast](#) – A Dynamo podcast by [Marcello](#) with help from [Jason](#) and [John](#)

## 23 Dynamo Packages You Should Check Out

Number	Dynamo Package Name	Latest Version*
0	Ampersand	2019.5.20
1	archi-lab.net (archilab)	2020.22.7
2	Archi-lab_Mandrill	2019.2.2
3	Data-Shapes	2019.2.39
4	Bang!	2019.3.1
5	bimorphNodes	3.0.3
6	BumbleBee	2019.2.1
7	Clockwork for Dynamo 2.x	2.1.2
8	Data-Shapes	2019.2.39
9	DynaMonito	2.1.1
10	DynaMaps	2019.2.21
11	DynaShape (download <a href="#">here</a> )	0.6.0.0
12	DynaWorks19 / DynaWorks20	2.0.0 / 2.0.0
13	iris	0.1.1
14	LunchBox for Dynamo (download <a href="#">here</a> )	2018.10.30.0
15	MEPover	2019.9.3
16	Monocle	2019.6.14
17	NodeModelCharts	2.0.2
18	Orchid (download <a href="#">here</a> )	7205
19	Refinery (download <a href="#">here</a> )	0.35.0
20	Rhythm	2019.10.15
21	Simplex	2019.7.7
22	spring nodes	203.1.0

\*As of Oct. 22nd, 2019