

Geometry Nodes

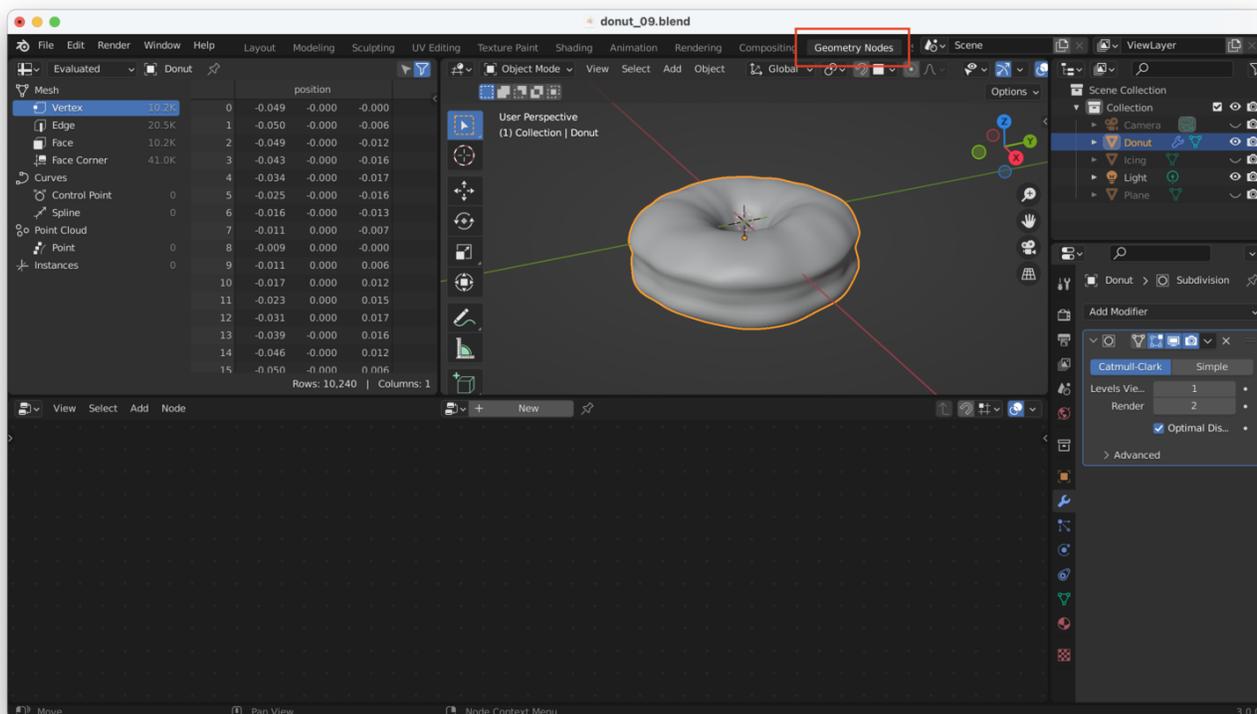
Next, we want to add some sprinkles to the donut. We'll use "geometry nodes" for this. In previous versions of the donut tutorial, "particles" were used, but "geometry nodes" are a better option in Blender 3.x.

Much like the nodes we've used so far to create texture, geometry nodes allow use to *procedurally generate* changes to the mesh (changes to our objects).

This enables all sorts of cool stuff, from the ability to dynamically generate frost or icicles on a surface, to the ability to create buildings whose height, window position, and number of floors are all procedurally controlled.

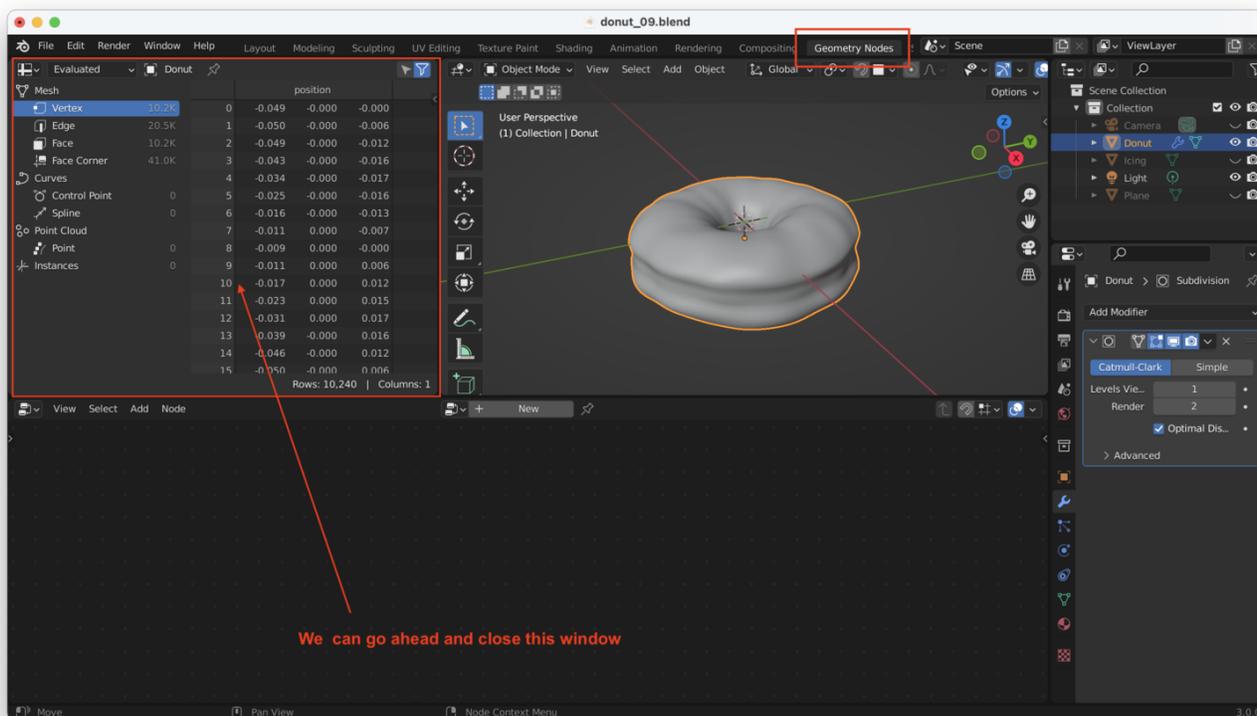
Start Using Geometry Nodes

We'll create a new *Workspace* (tab) at the top of blender, which will allow us to see our geometry node setup alongside our donut:

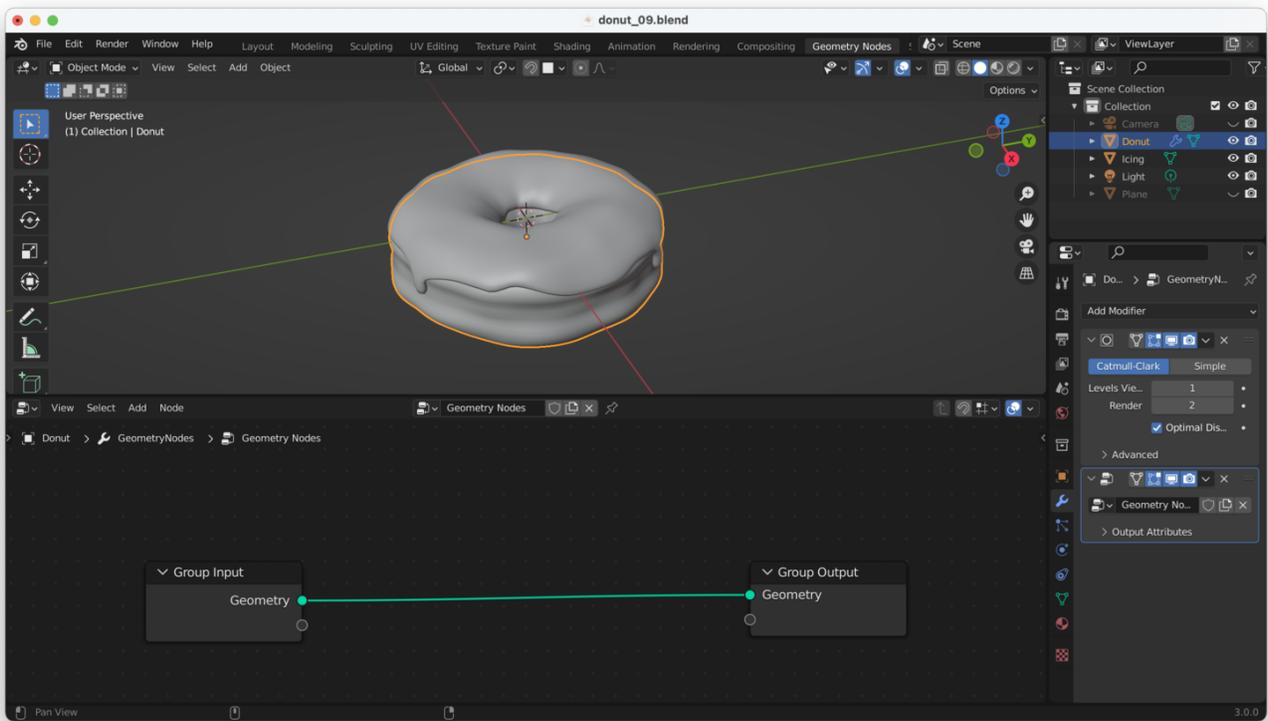
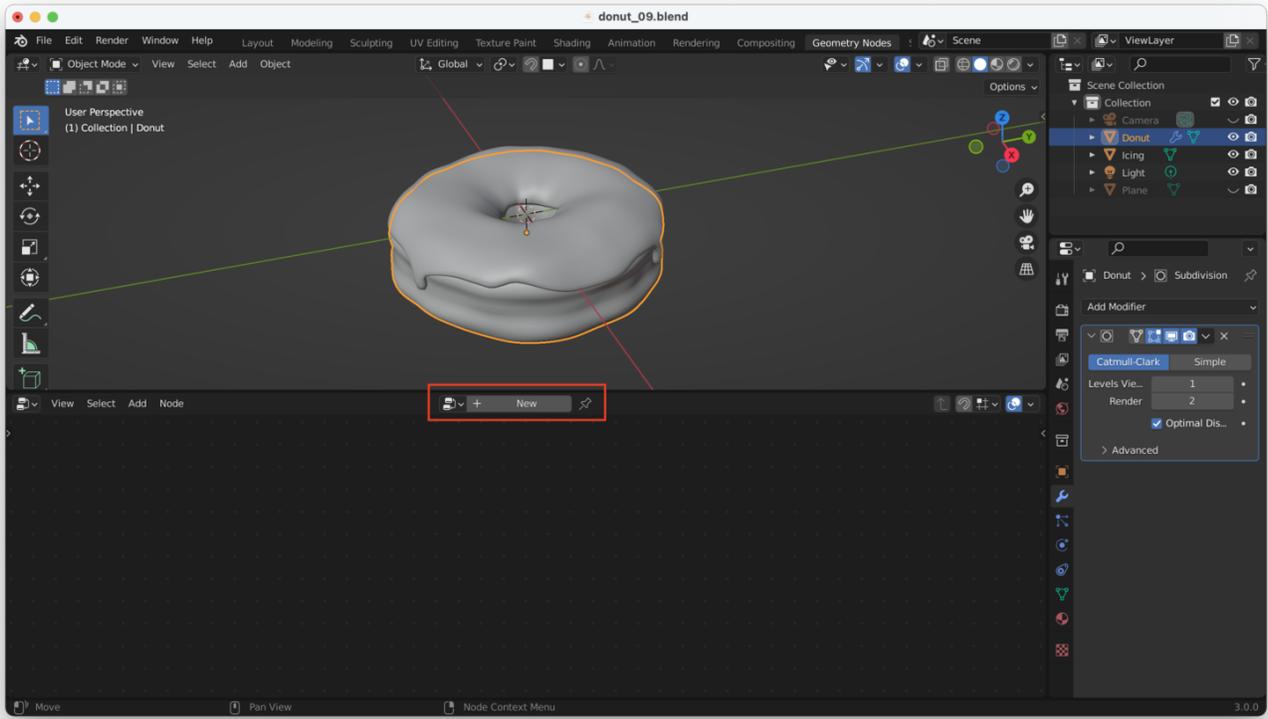


If you don't see this, look for a little down arrow near the top right corner of the top menu bar. Click on that, and choose "General → Geometry Nodes".

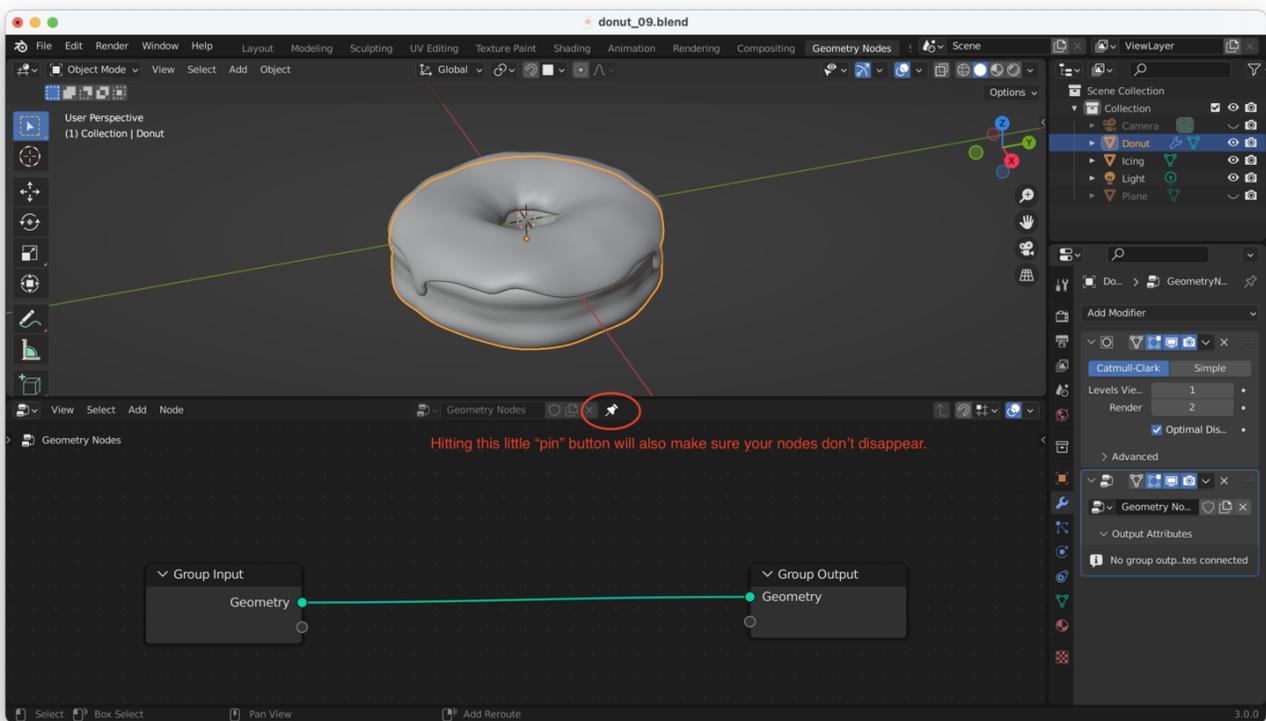
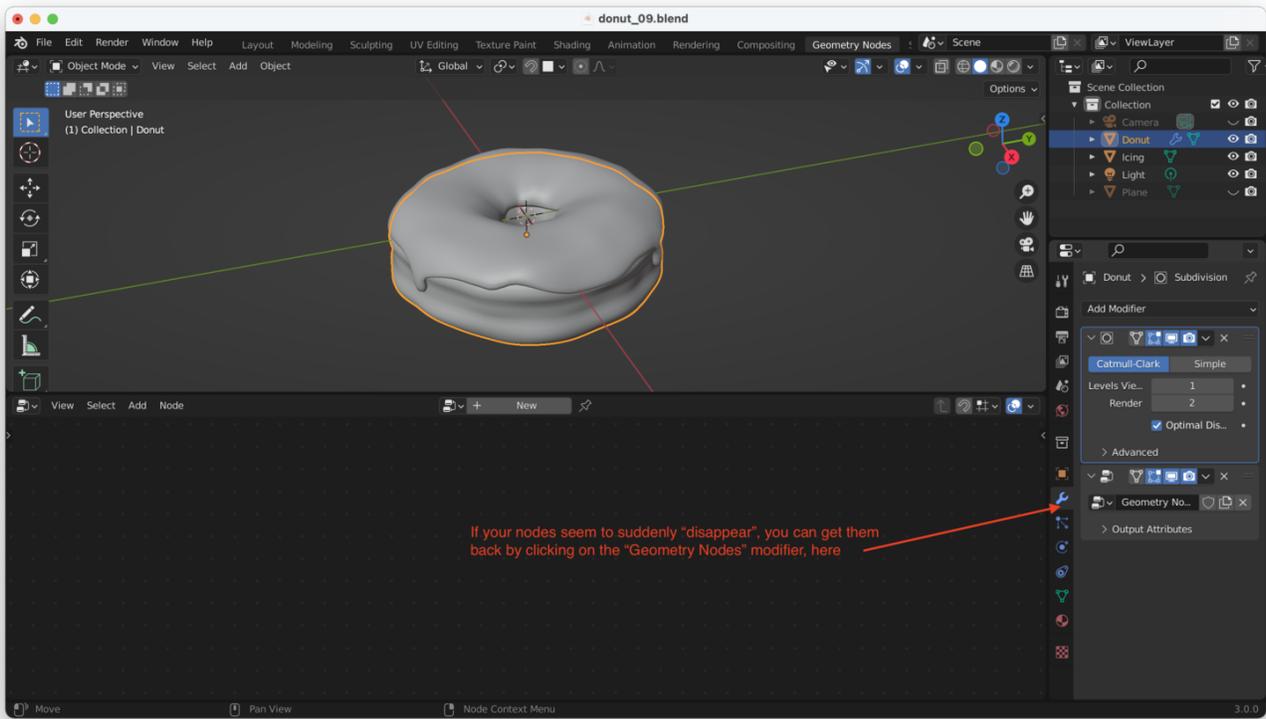
We won't be needing the window in the upper left that shows the positions of all our vertices, so we can close that:



Now, we click on "New" to create a new geometry node setup:

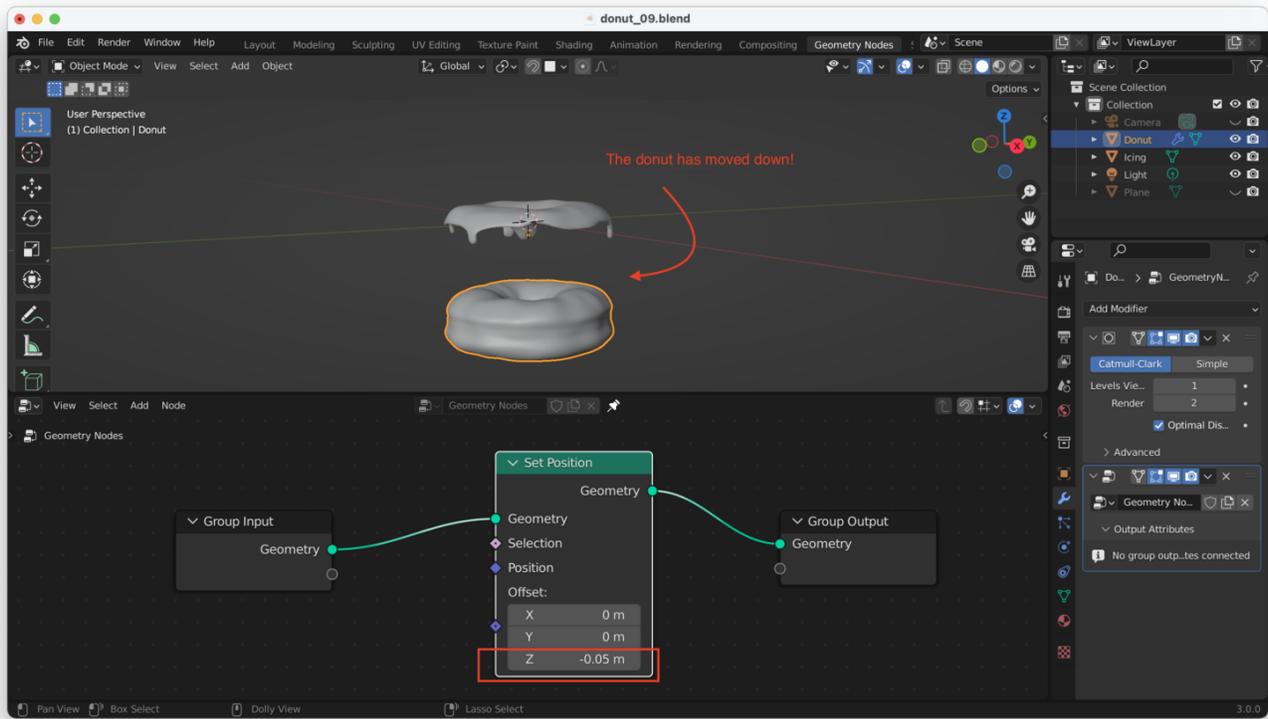


Note that if your nodes disappear, you can get them back by clicking on the "Geometry Nodes" modifier in the right-hand pane:

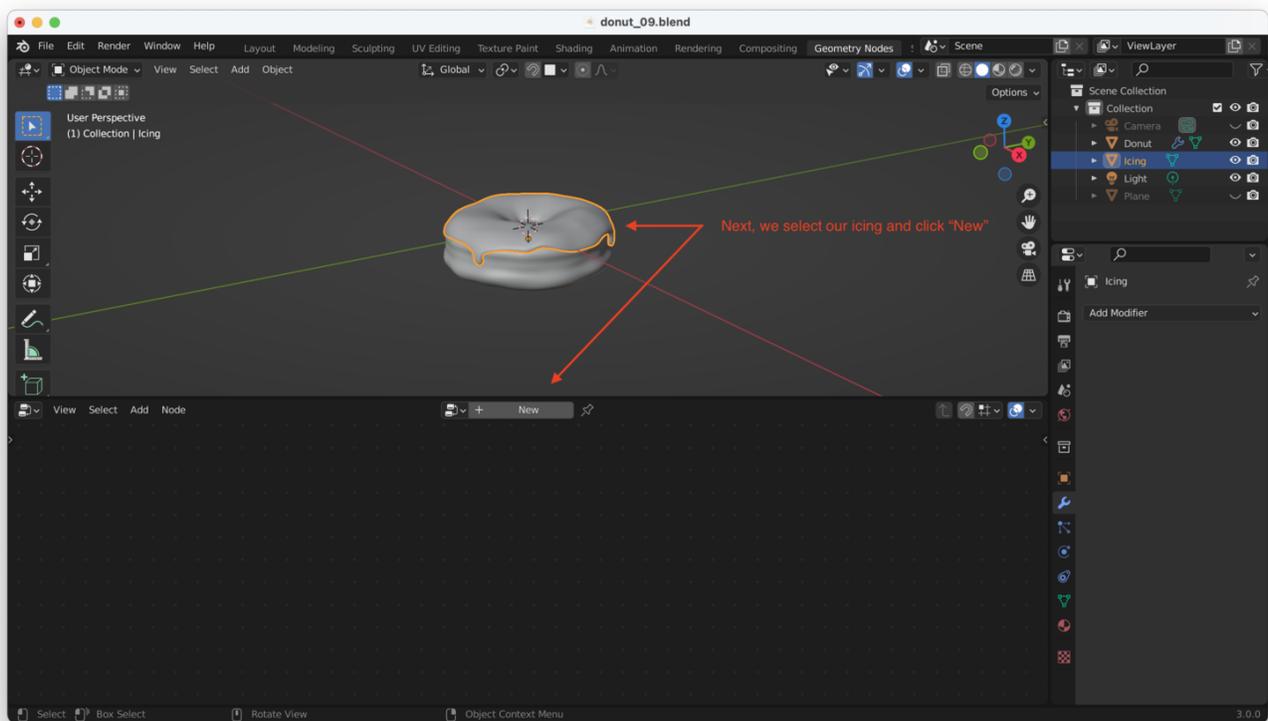
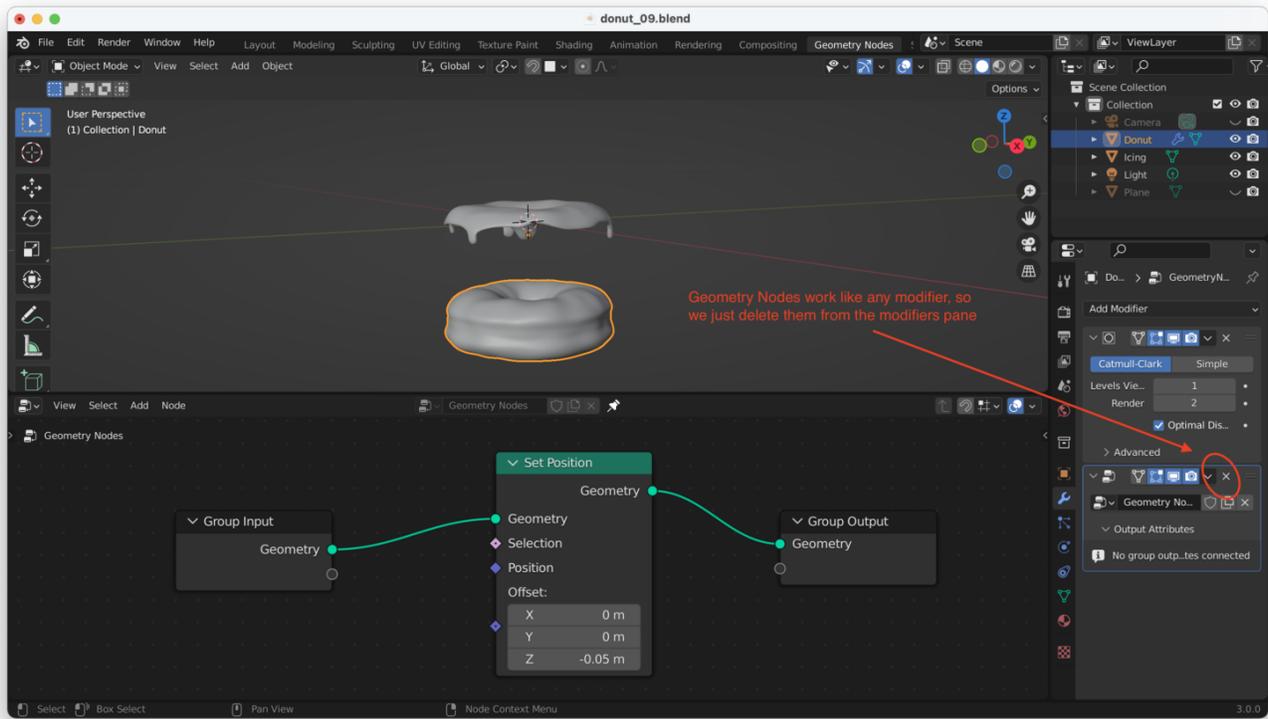


The node on the left (Group Input) reads in the existing data for our mesh (for the donut, in this case).

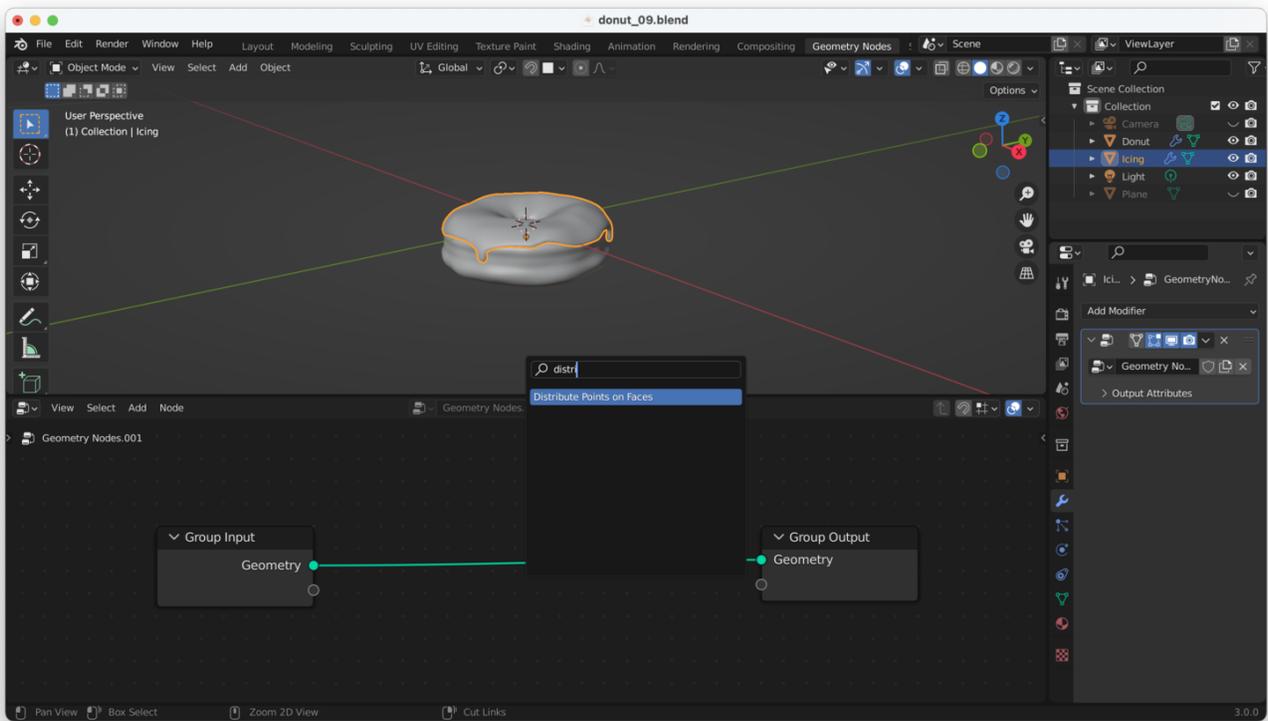
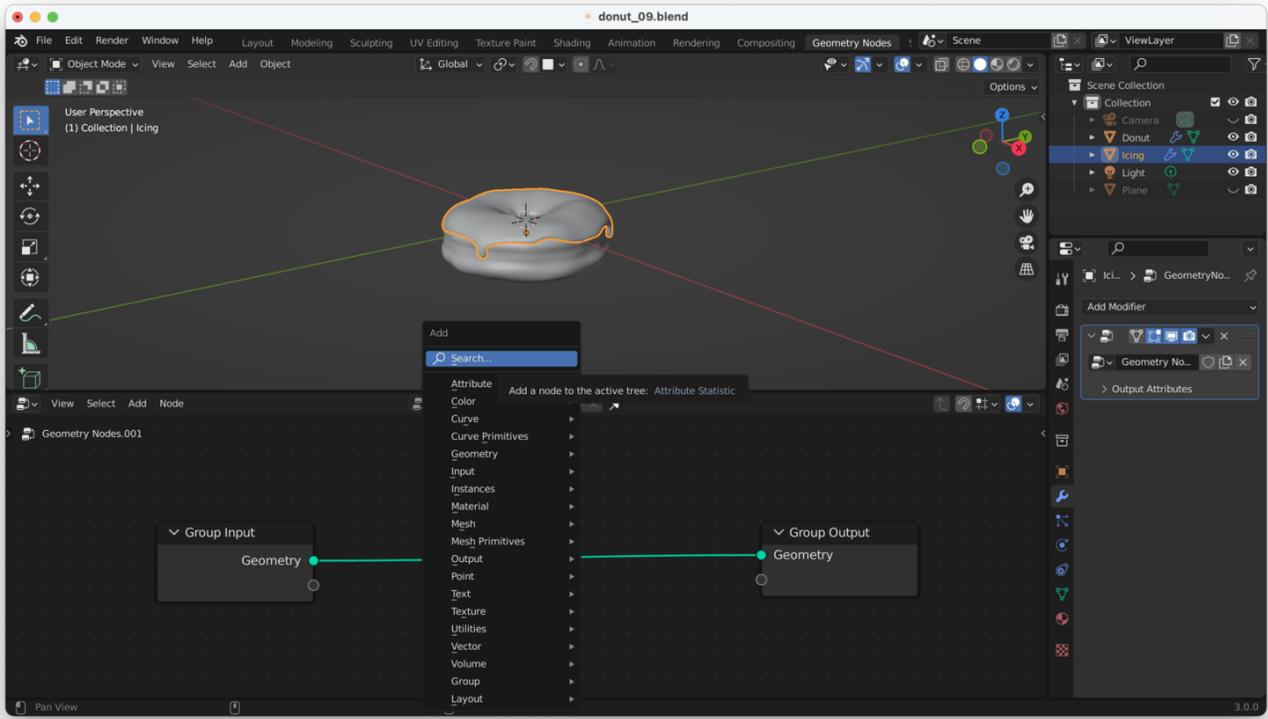
The node on the right is the final mesh, after any changes made by our geometry nodes. For instance, if we add a "set position" node and add a Z offset, the donut now moves (the output mesh position has been changed):

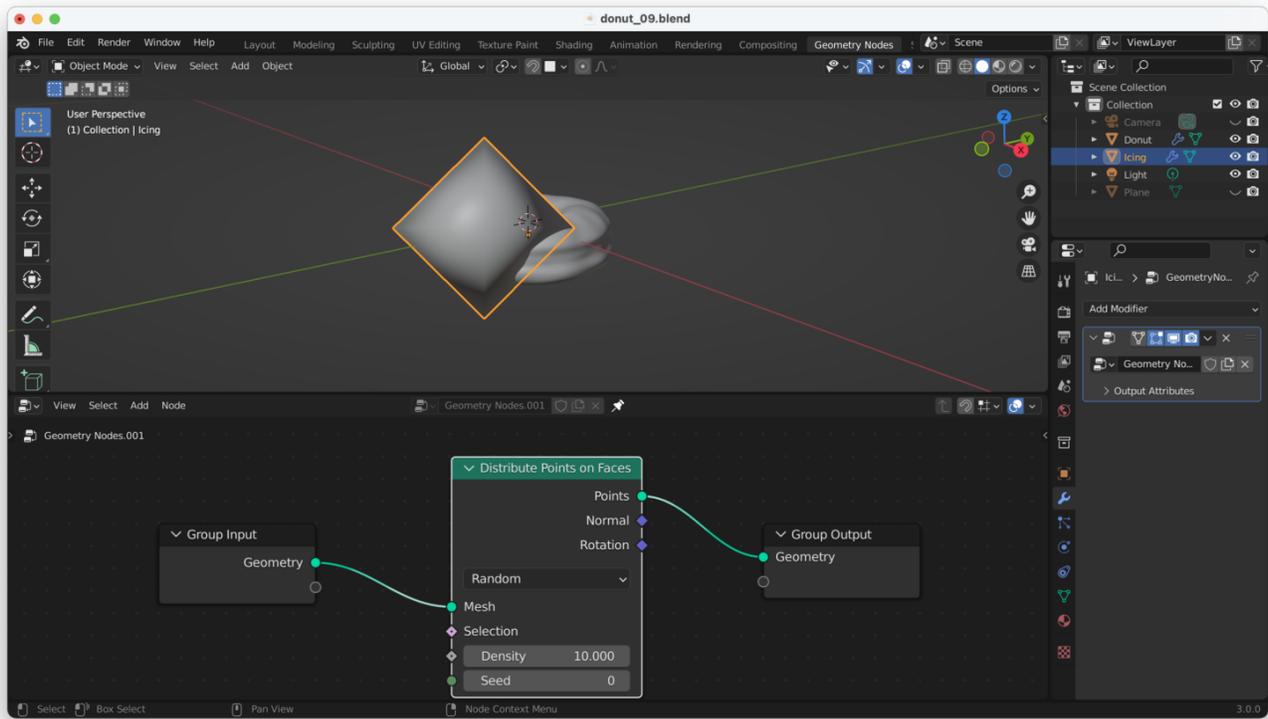


We don't want to alter the donut, but rather the icing (we are adding sprinkles). I now delete the Geometry Nodes I created for the Donut, and switch to the Icing instead:



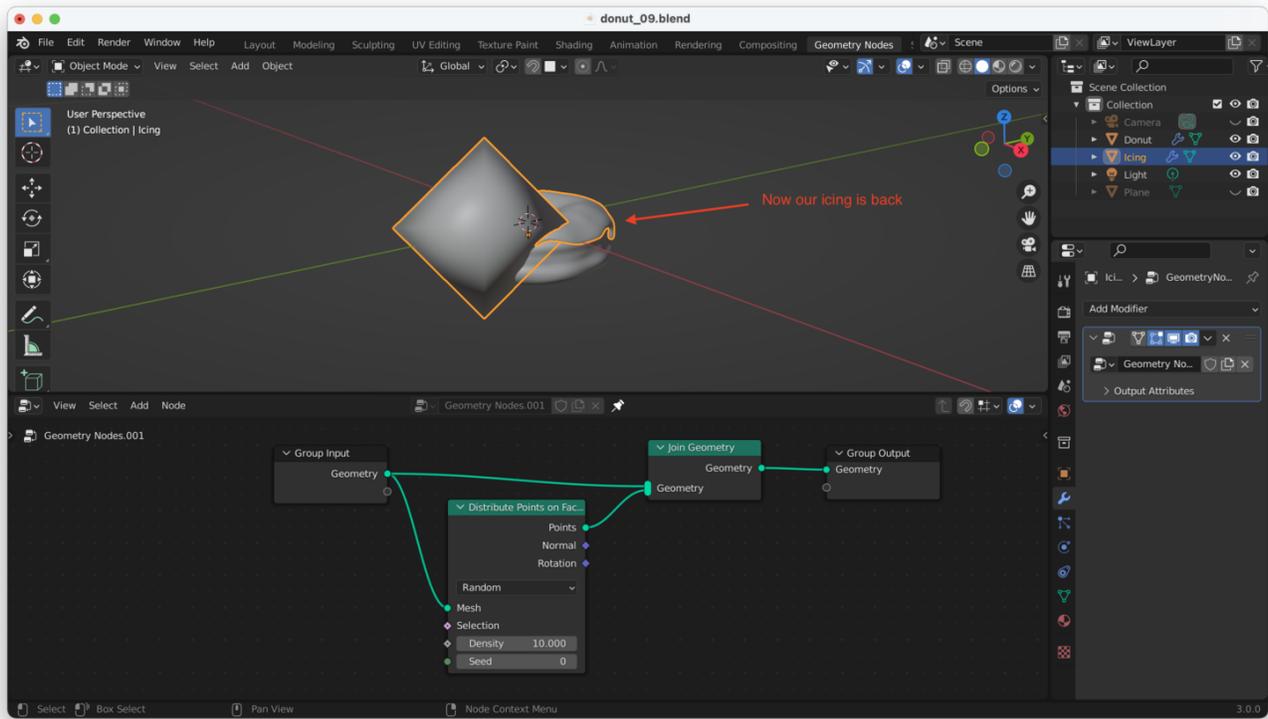
We now need to add a "Distribute Points On Faces" node. Hit "Shift + A" to add a new node, then start searching using the search bar that opens:



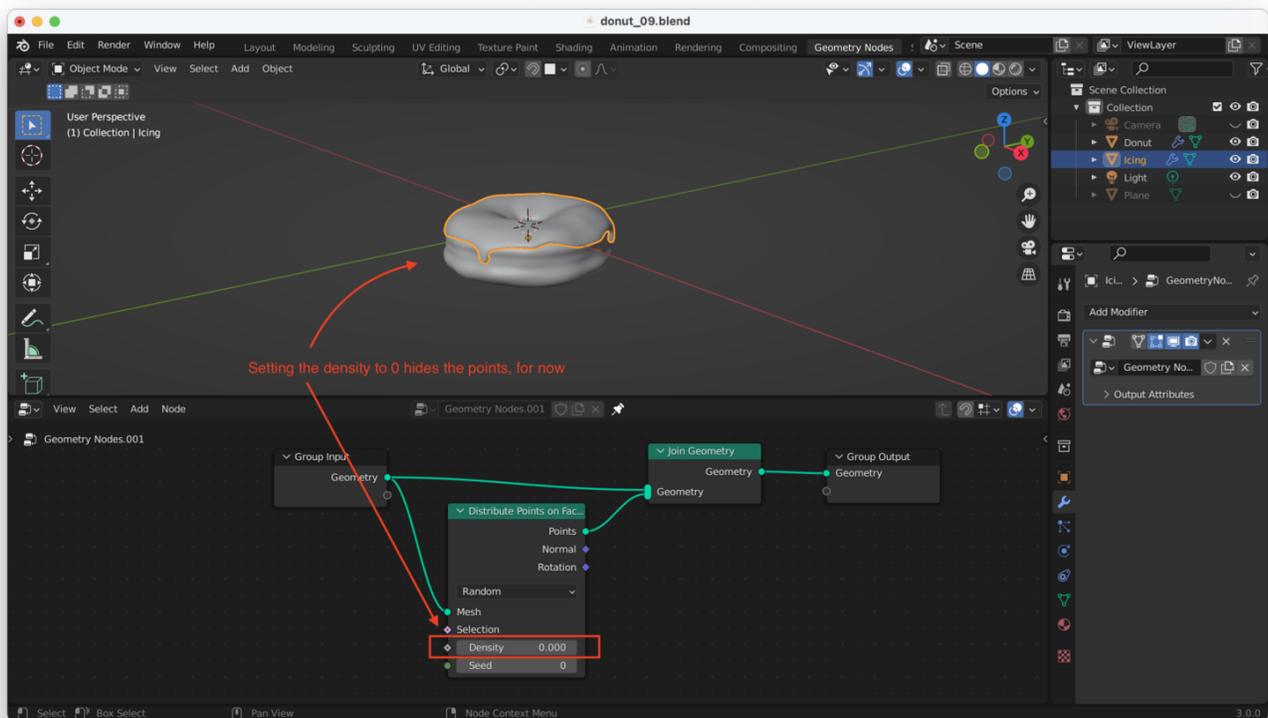


You can see there's a single enormous "point" sticking to the icing. We need to toggle the density to adjust this, which we will do later.

The icing also appears to have "disappeared". This is because we have **replaced** our icing mesh with the output from "Distribute Points on Faces". We need to add a "Join Geometry" node to combine our points with the original icing mesh, like this:



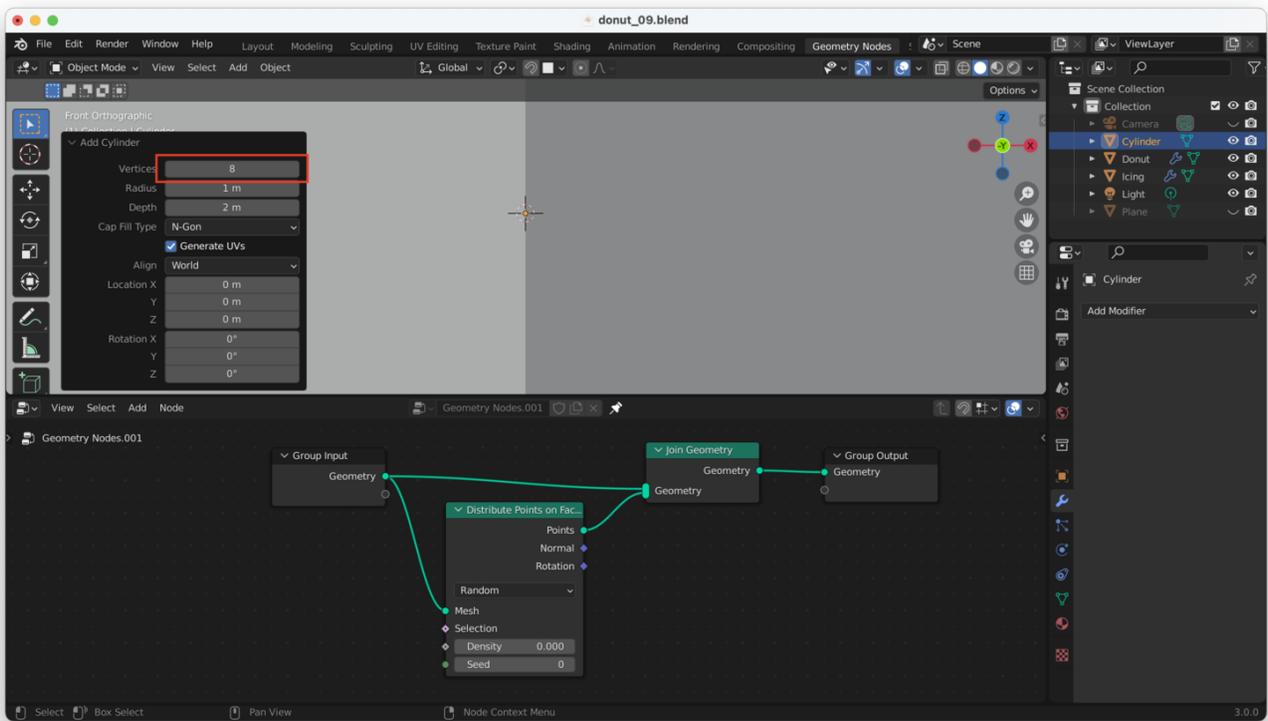
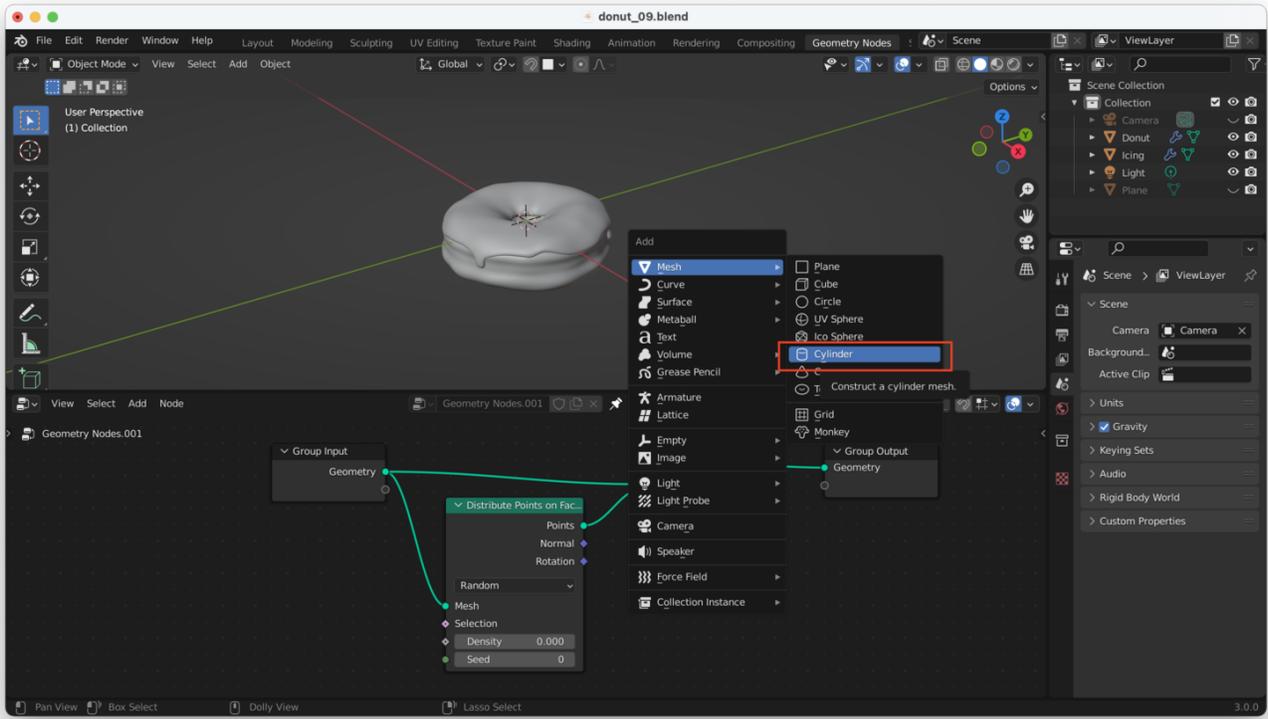
For the time being, we also set the density of our points to "0", which will hide the giant "blobs" that Blender is currently rendering in the viewport:

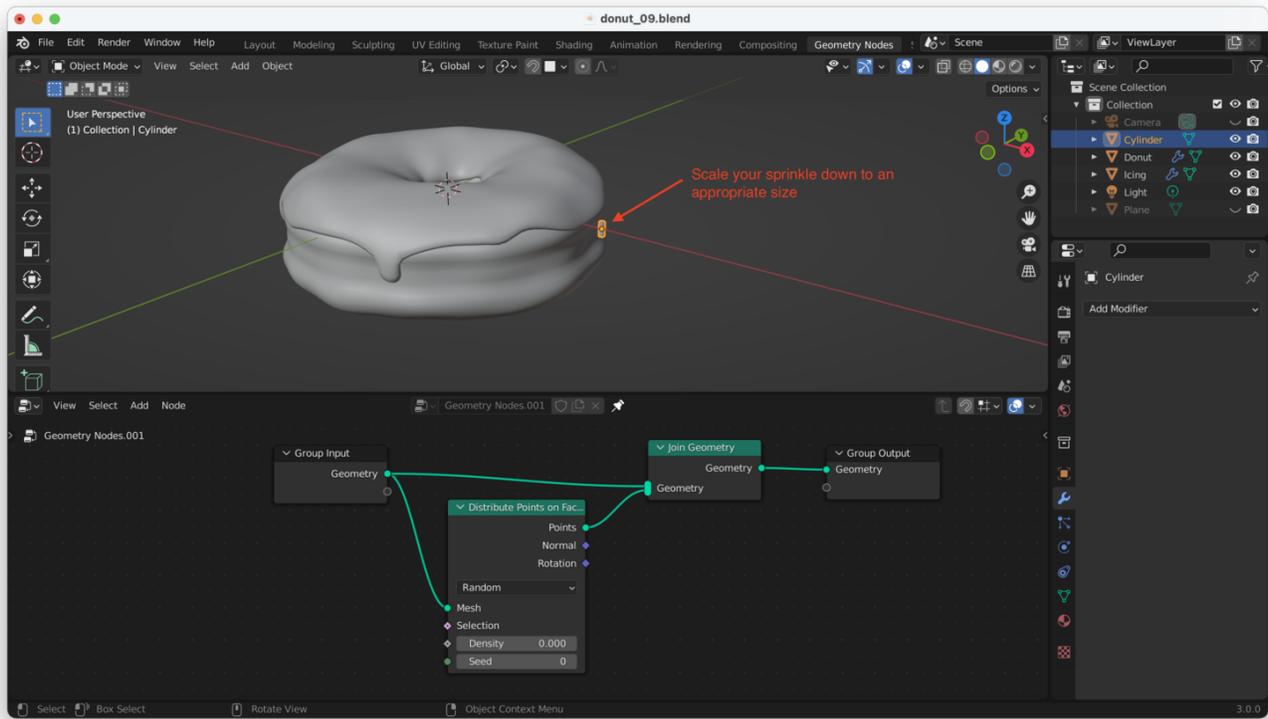


Now we take a break from Geometry Nodes to create a model for the sprinkles we want to distribute over the surface of the icing. From the viewport, we hit "Shift + A" then go to

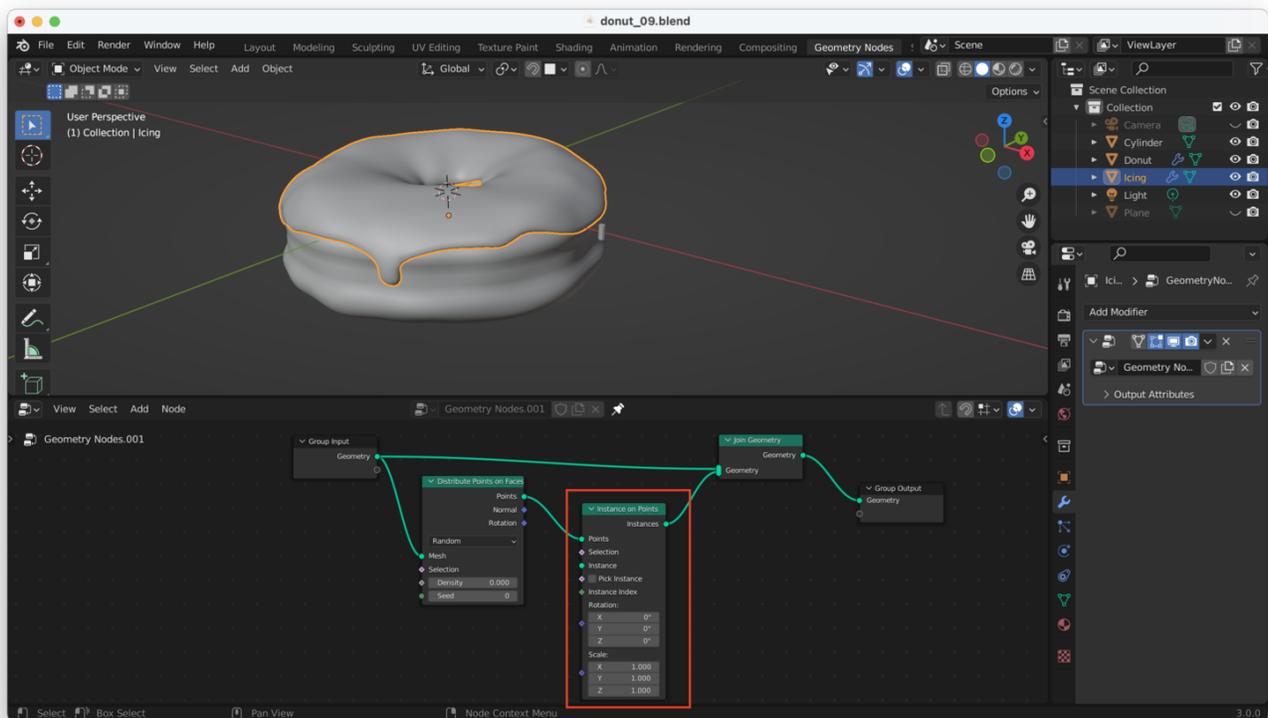
"Mesh → Cylinder" to add a new cylinder to the scene. You might have to zoom out to see it: the default size is **way too big!**

We need to scale down the sprinkle and reduce its level of detail. Because there will be dozens or hundreds of sprinkles scattered over the icing, we don't want or need excessive detail (which will slow down renders and also slow down editing in the viewport). Setting the vertex count for the sprinkle to "8" should be OK, as here:



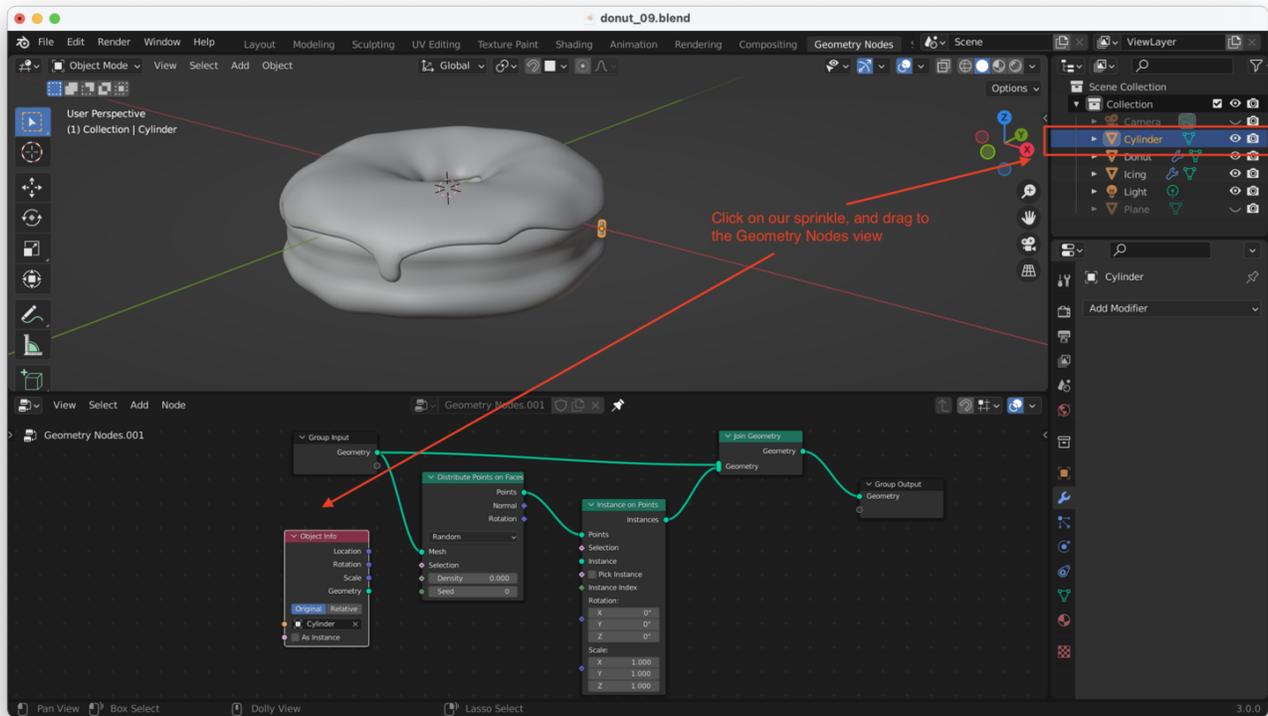


We now need to add a new geometry node, "Instance on Points" to our Icing's Geometry Node setup:

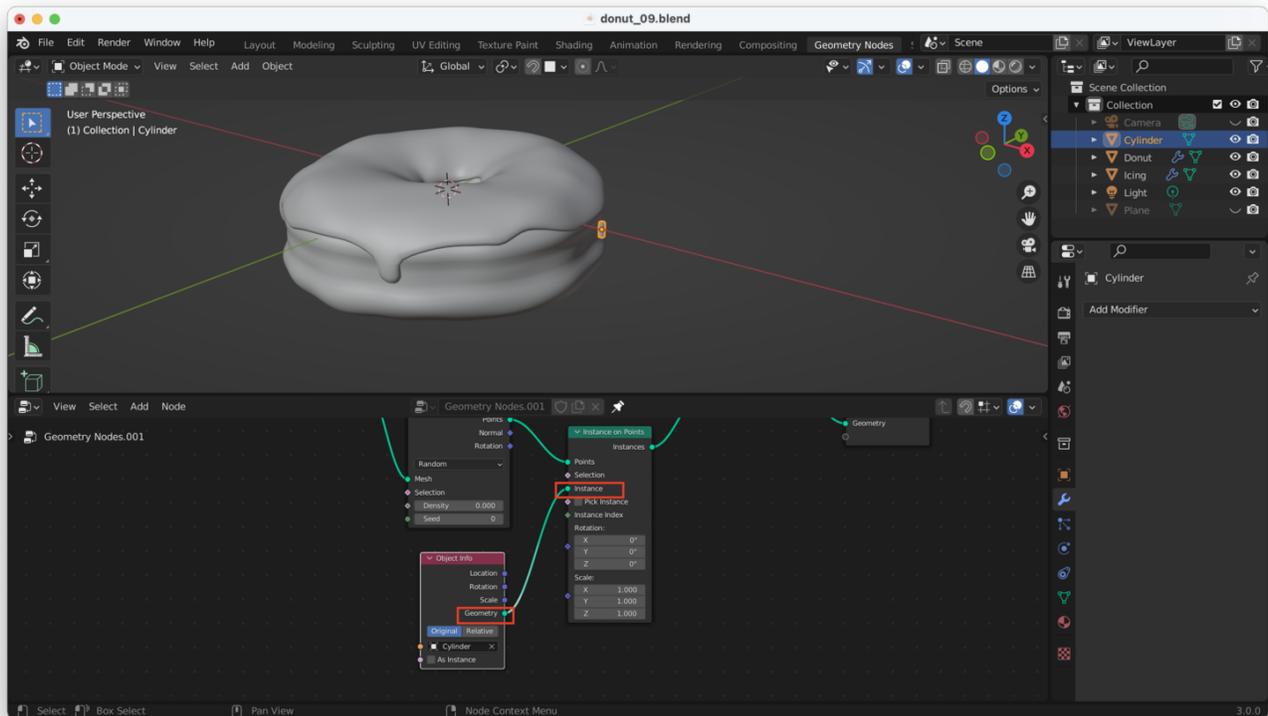


Next, we can click on our sprinkle in the Outline view at the top right, then click and drag it onto the Geometry Nodes view at the bottom. This will generate a new "Object Info"

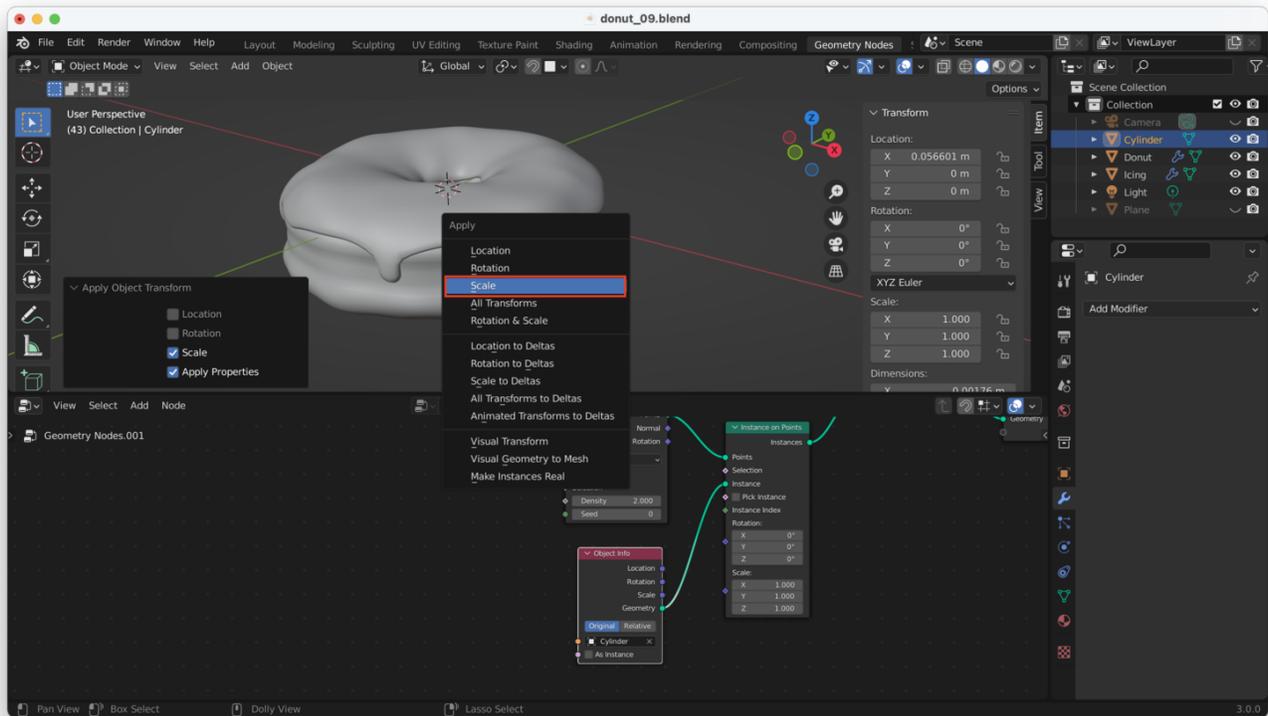
node, like this:



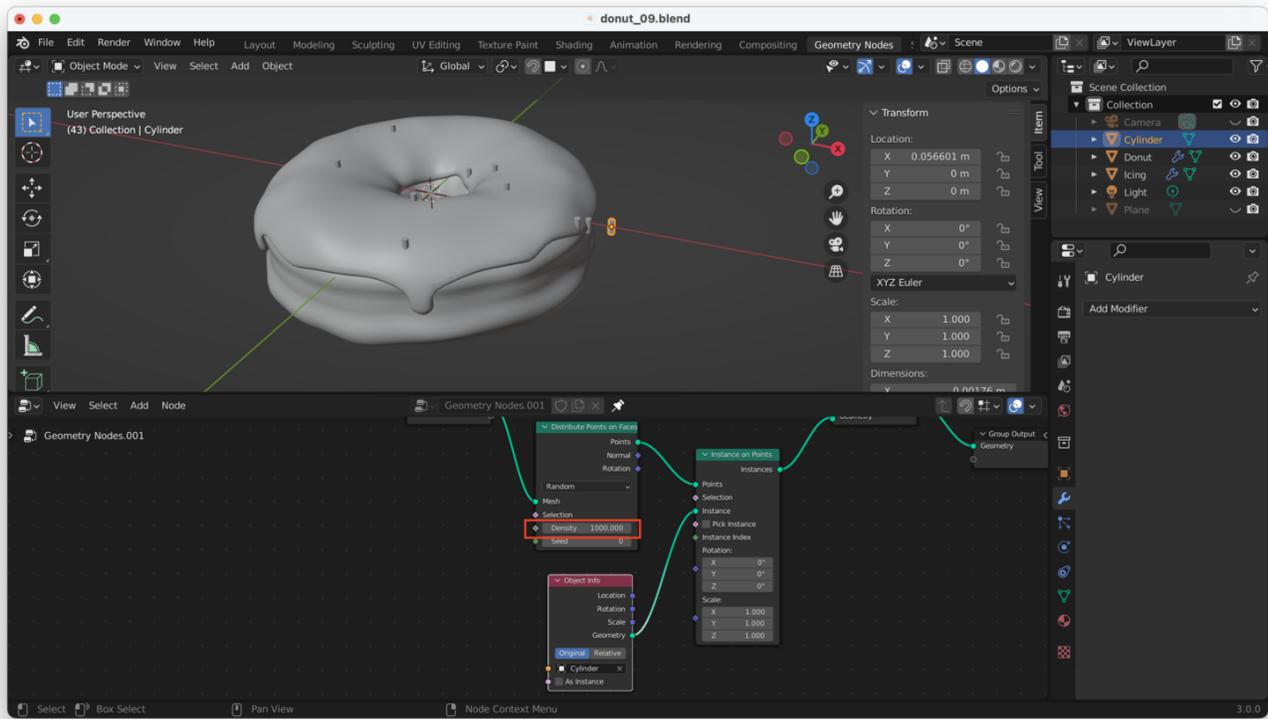
We then connect the "Geometry" output of the Object Info node to the "Instance" input on the Instance on Points node:



One last step: we scaled our sprinkle cylinder, and we now need to "apply" its new scale, before it will start showing up on the icing correctly. Click on the sprinkle, hit "Control + A", and then choose "Scale":

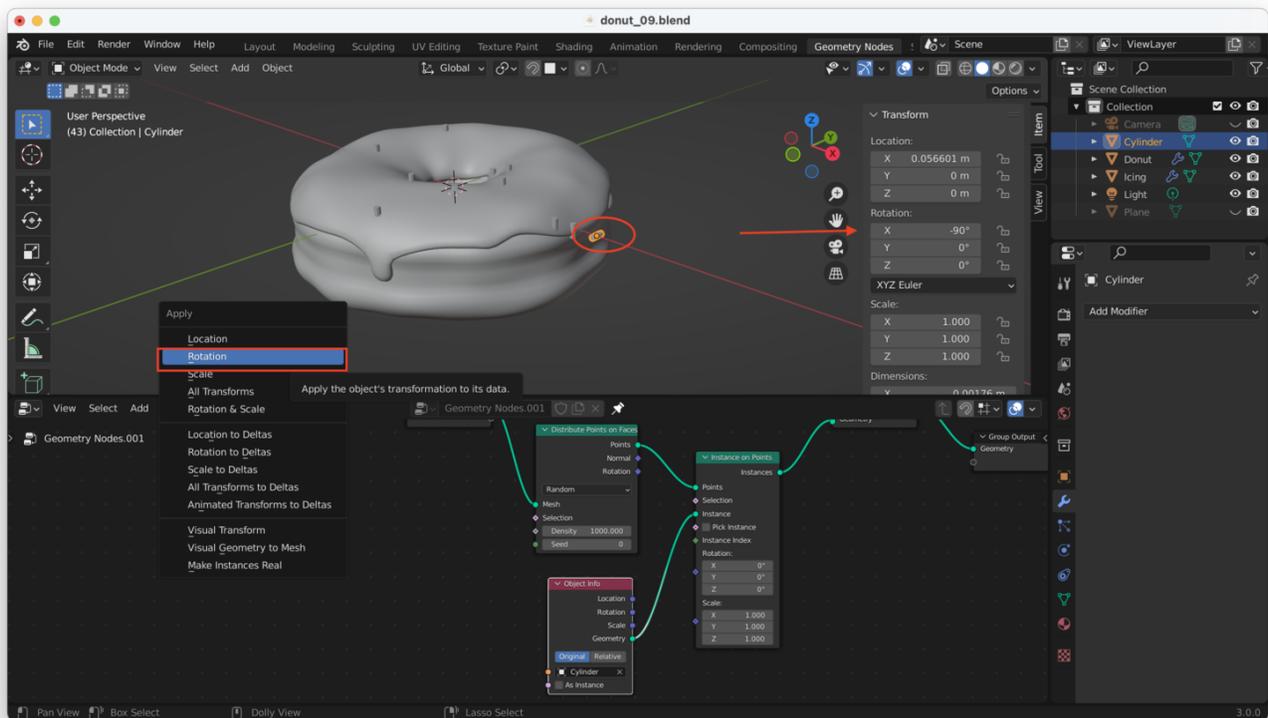


You may need to play with the density a bit before you start seeing sprinkles on your icing:

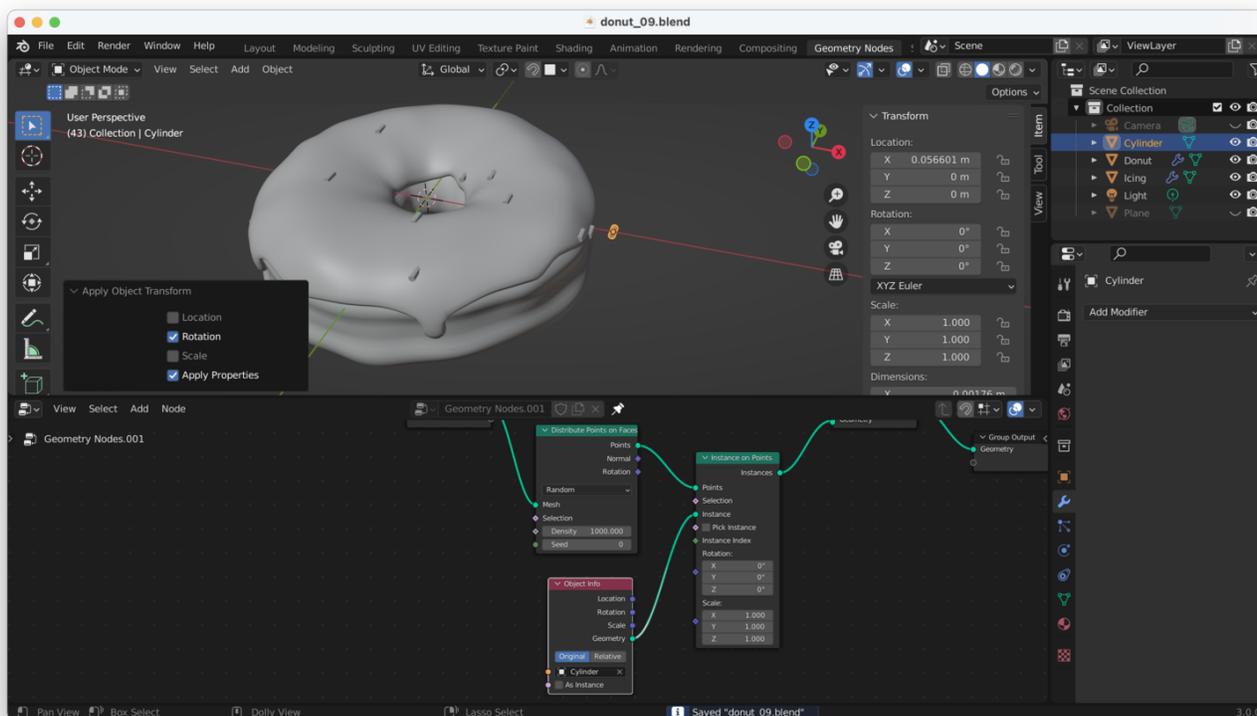


Of course, the sprinkles are now rotated the wrong way. Rotate your sprinkle so that it's sitting horizontally (-90 degrees).

Again, after you do this you have to **apply the rotation** with "Control + A", this time choosing "Rotation":

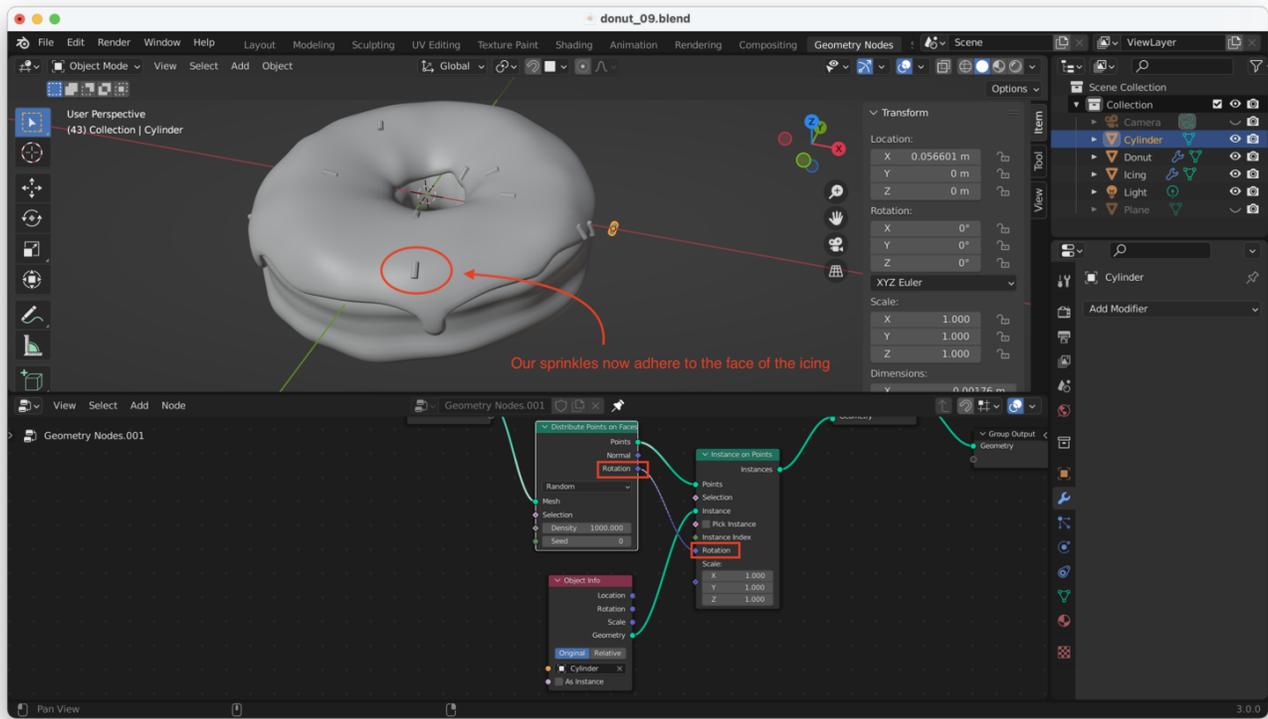


Things now look a little better, but all the sprinkles are pointing in a uniform direction and sticking out from the icing very unnaturally:



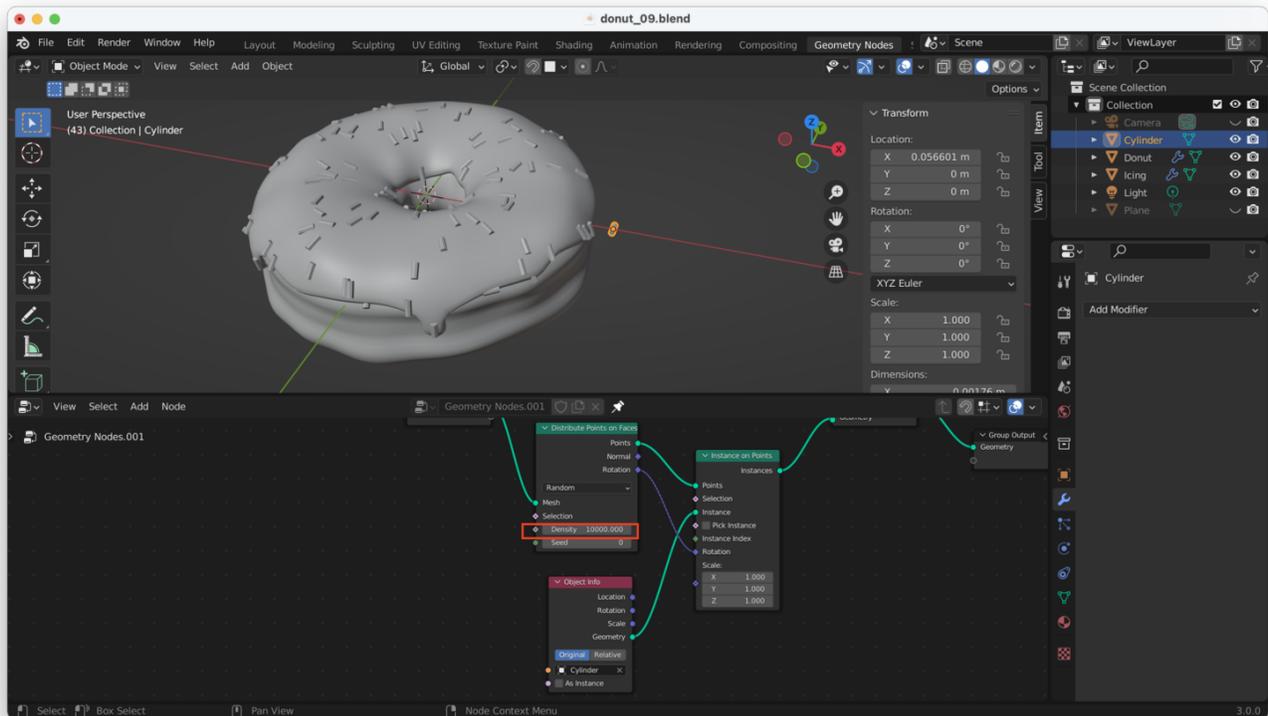
We can fix this by making the sprinkles **follow the geometry of the faces** nearest each sprinkle (i.e. the sprinkle should be aligned with the icing mesh itself).

To do this, we take the "Rotation" output from "Distribute Points on Faces", and feed it into the "Rotation" input on "Instance on Points":



We also need to increase the density. Because we are working at a very small scale, we need very high densities in order to see enough sprinkles showing up!

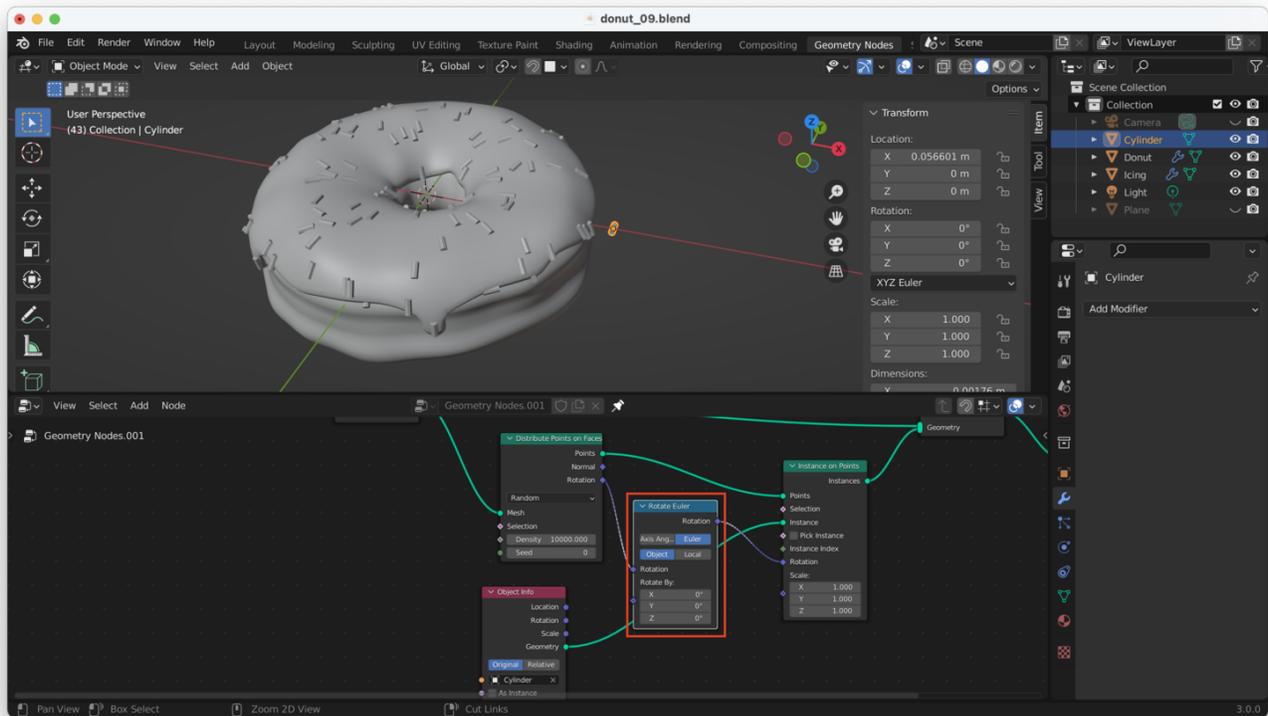
In my case, a density of about "10,000" gets me the results I'm looking for:



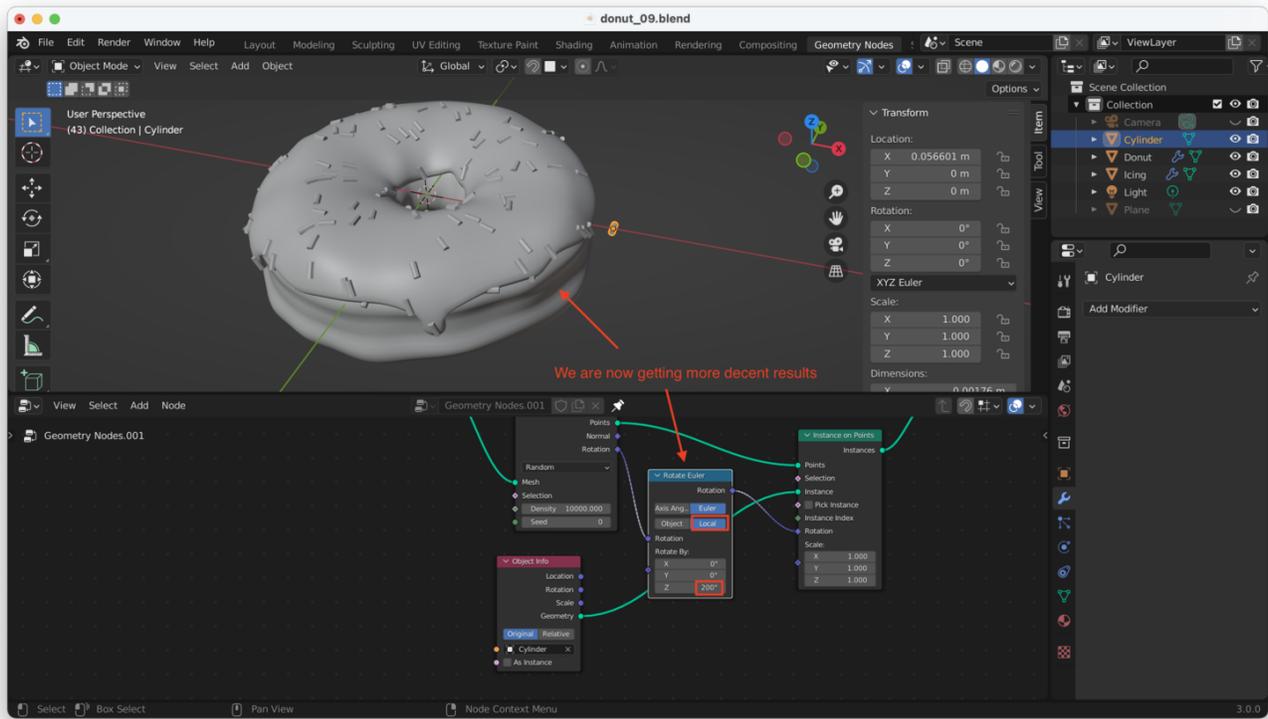
Note: The reason for this is that the units for "density" are "density per **meter**".

We **still** have a problem, though. Our sprinkles are all sort of uniformly "brushed" onto the surface of the donut. We need a way to adjust the rotation for each sprinkle to get a more organic look.

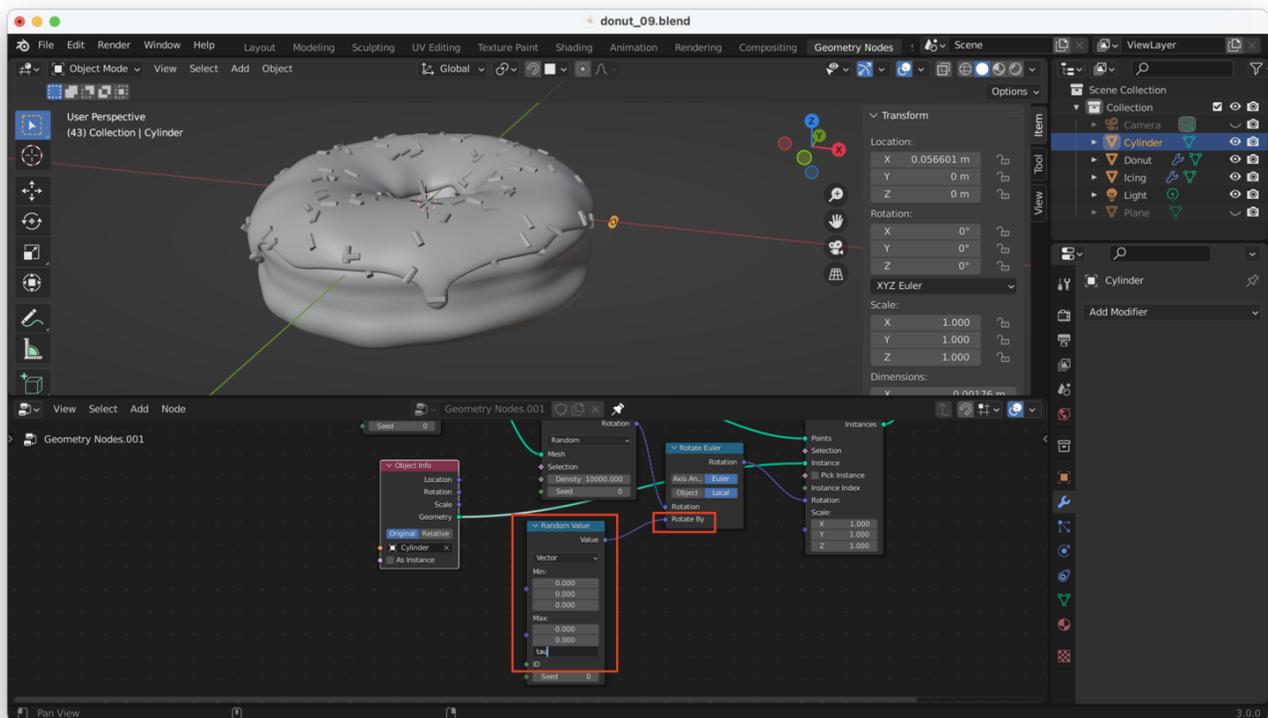
We can do this by adding a "Rotate Euler" node, like this:



We need to make some adjustments to this node to get the results we want. We need to change the rotation effect from "Object" to "Local", and we also need to tweak the "Z" value until we get a nice-looking result:



However, we *still* want to have some randomness in the rotation value for each sprinkle, so we need to add a "Random Value" node, as shown here:



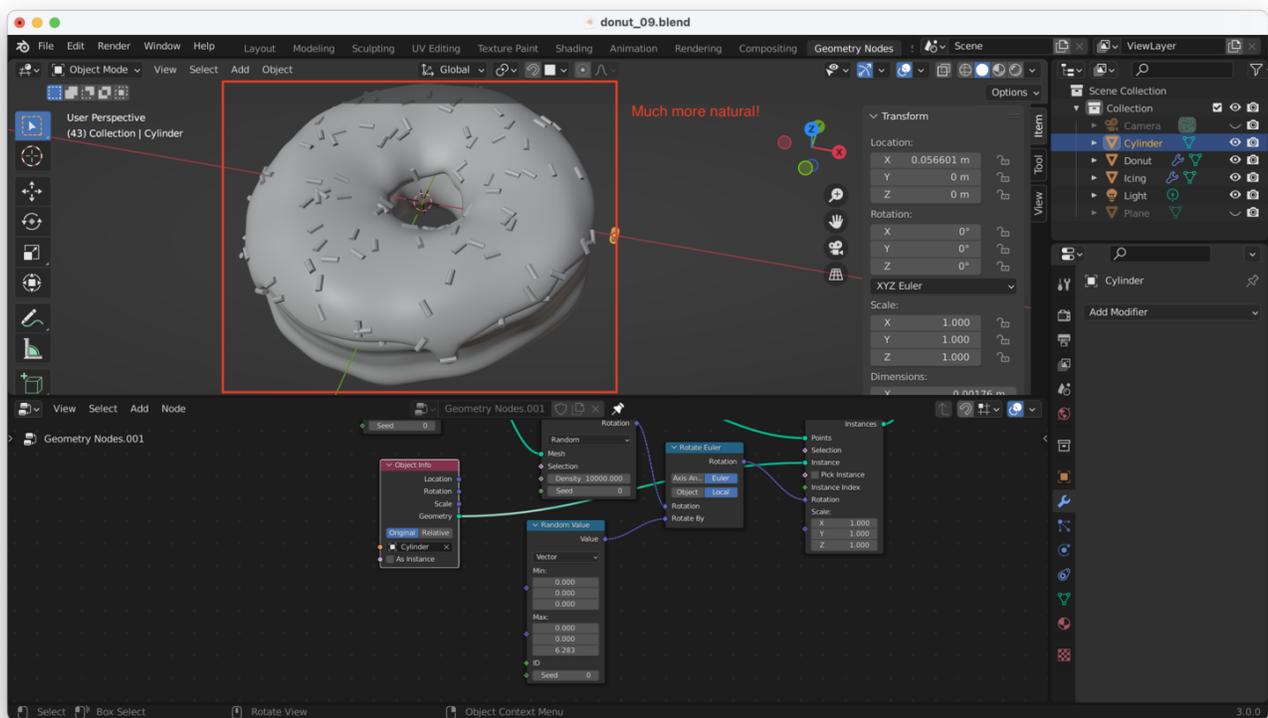
Tip: As shown here, you can type in the **name** of a mathematical constant such as "tau", and hit enter. Blender will convert that to its numerical value

for you. Neat!

There's a lot going on here, so let's explain:

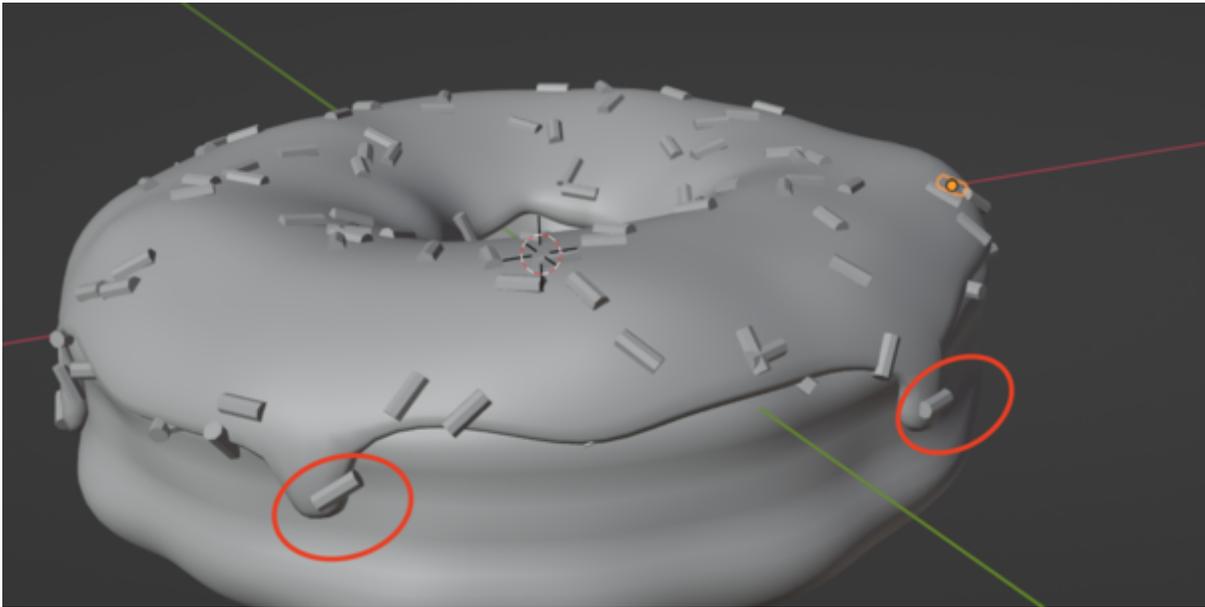
We have set the Random Value node to "Vector", meaning it will produce [X,Y,Z] vectors as output. Setting the **min** and **max** values for X and Y to zero means we won't generate any rotation in X or Y when we input our vector to the "Rotate Euler" node. We set "Z" to "tau" because "tau" is equal to 2π (360 degrees). This means our "Z" values can range from 0 degrees to 360 degrees, so our sprinkles can be rotated any direction along the Z axis.

This gives us a much nicer, much more natural-looking set of sprinkles, like this:

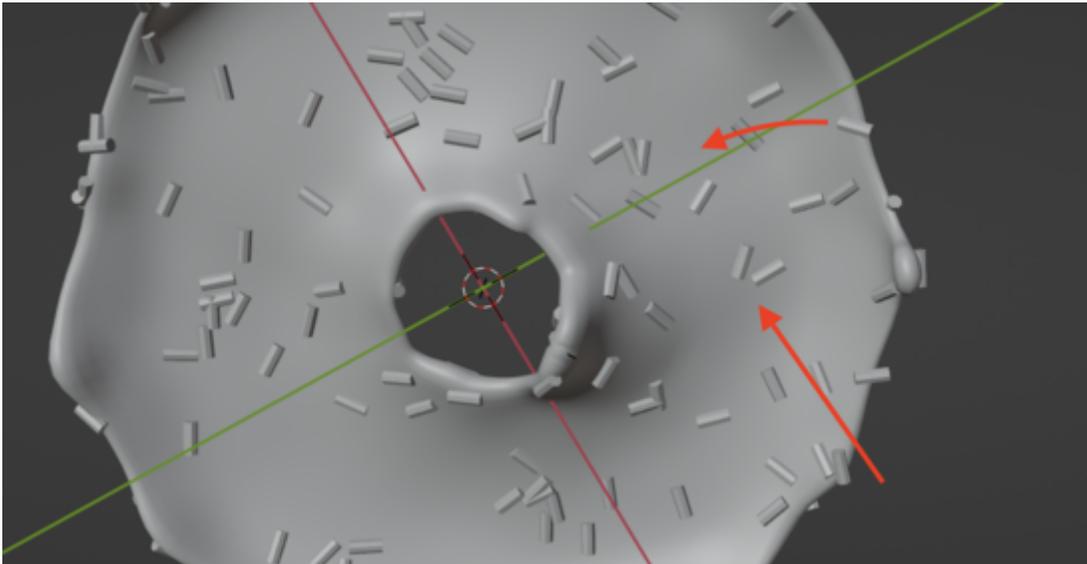


We now need to fix two more problems:

1. We don't want sprinkles showing up on the "globs" of icing at the edges:

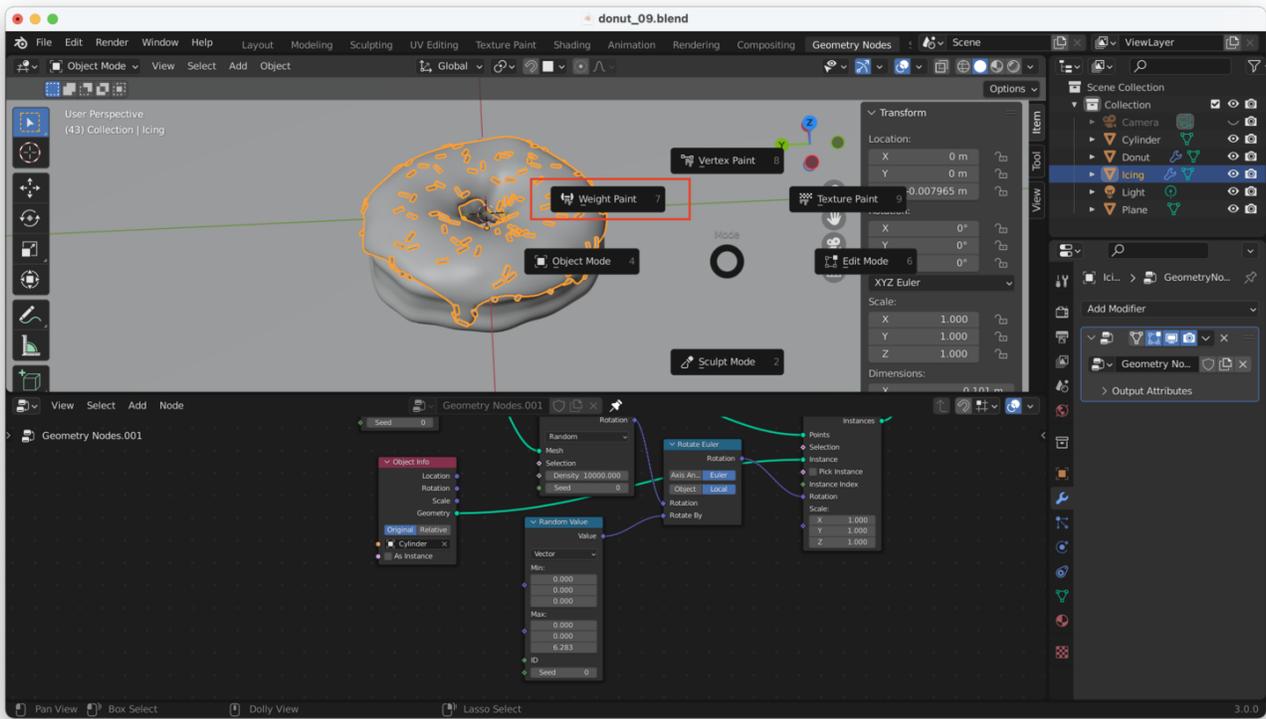


2. We also don't want sprinkles showing up "underneath" the icing:

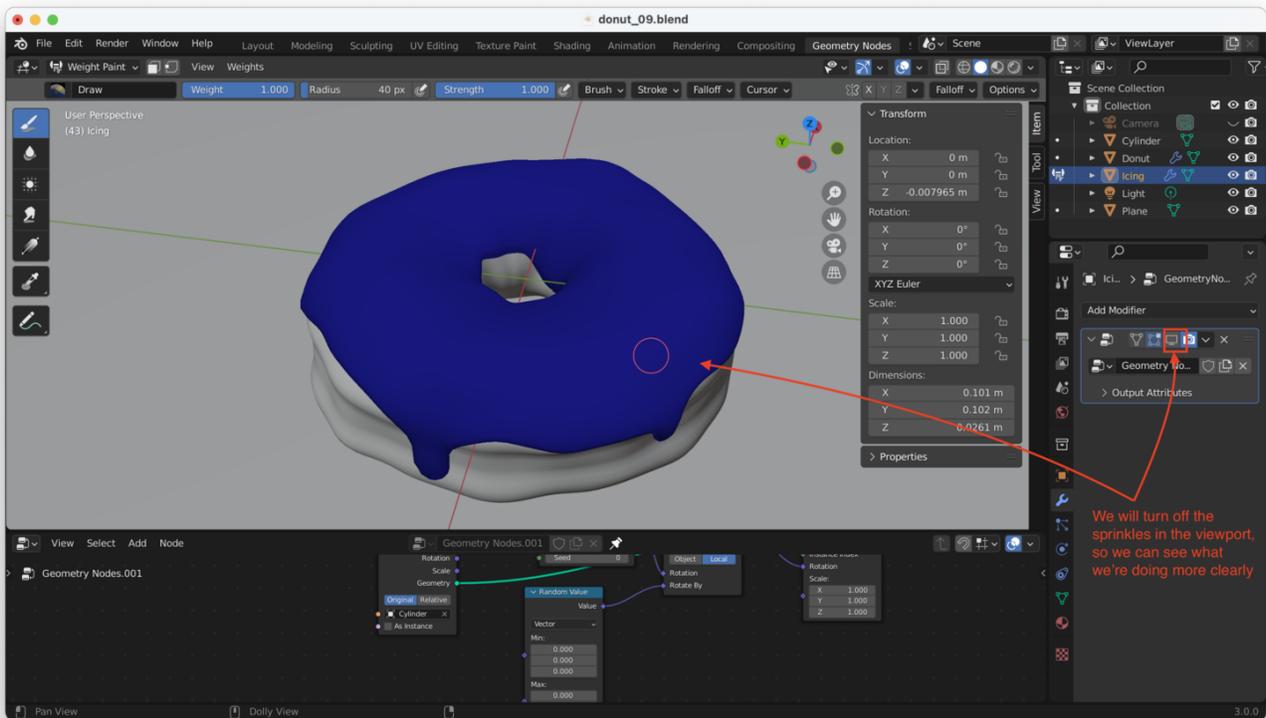


To solve these issues, we are going to use Blender's "weight paint" tool. This will let us "paint in" the areas on the icing where we *want* sprinkles to appear.

Select the icing and hit "Control + Tab" (**not** "Command + Tab", even if you're on a Mac), then choose "Weight Paint":

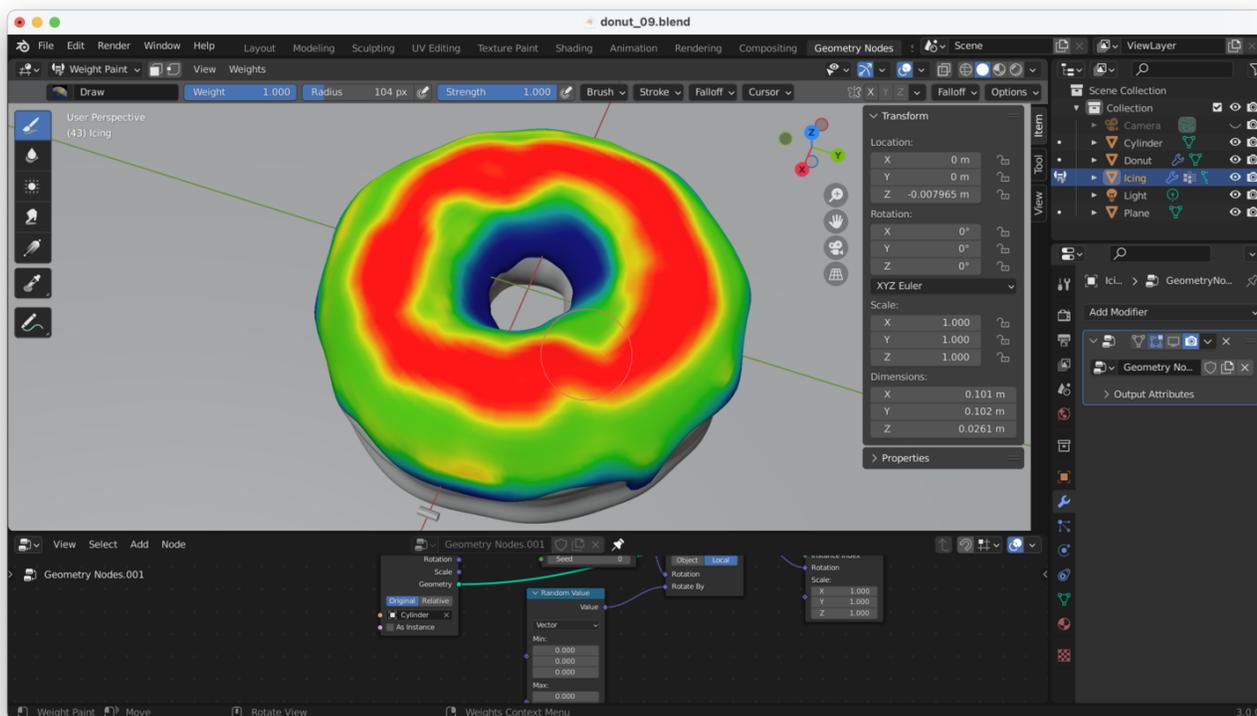


We will hide our sprinkles in the viewport, so we can more easily see what we're doing:



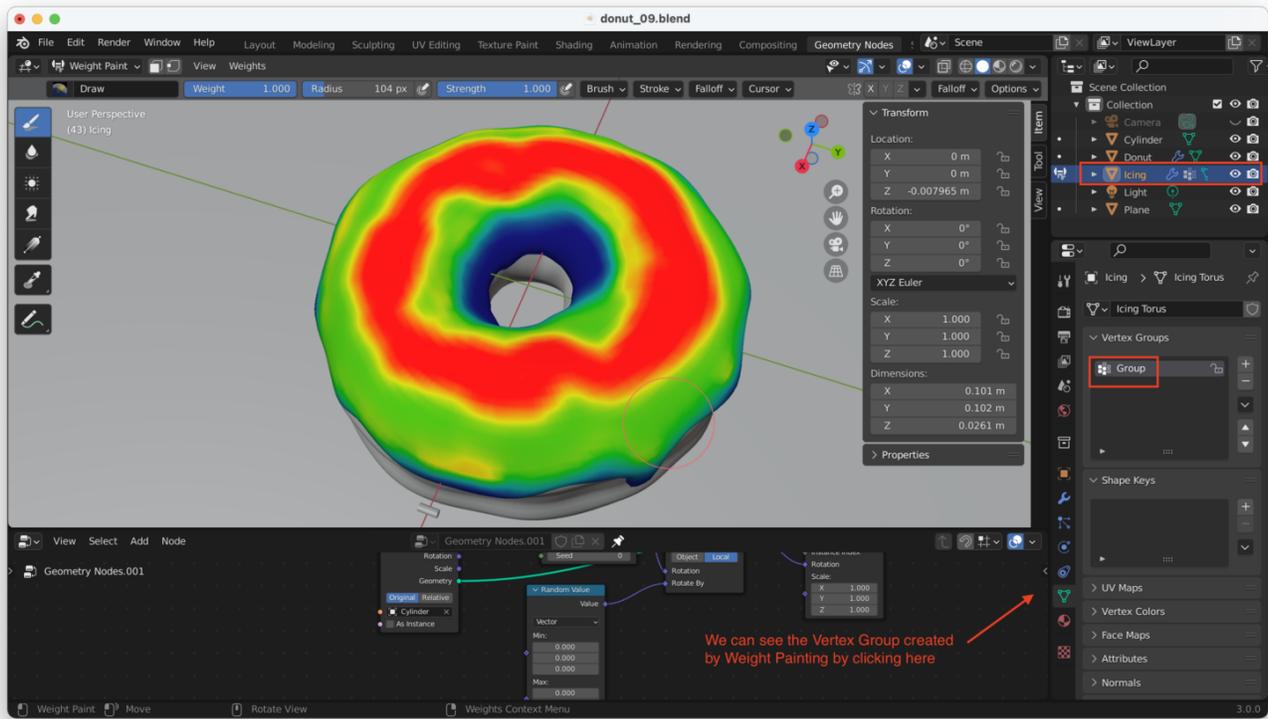
Tip: When weight painting, a blue color indicates a value of "0", while red represents a value of "1". Higher values will end up with more sprinkles.

We then paint the surface of the donut. Try to avoid painting it in a completely uniform way, as we want a more "organic" distribution for the sprinkles:

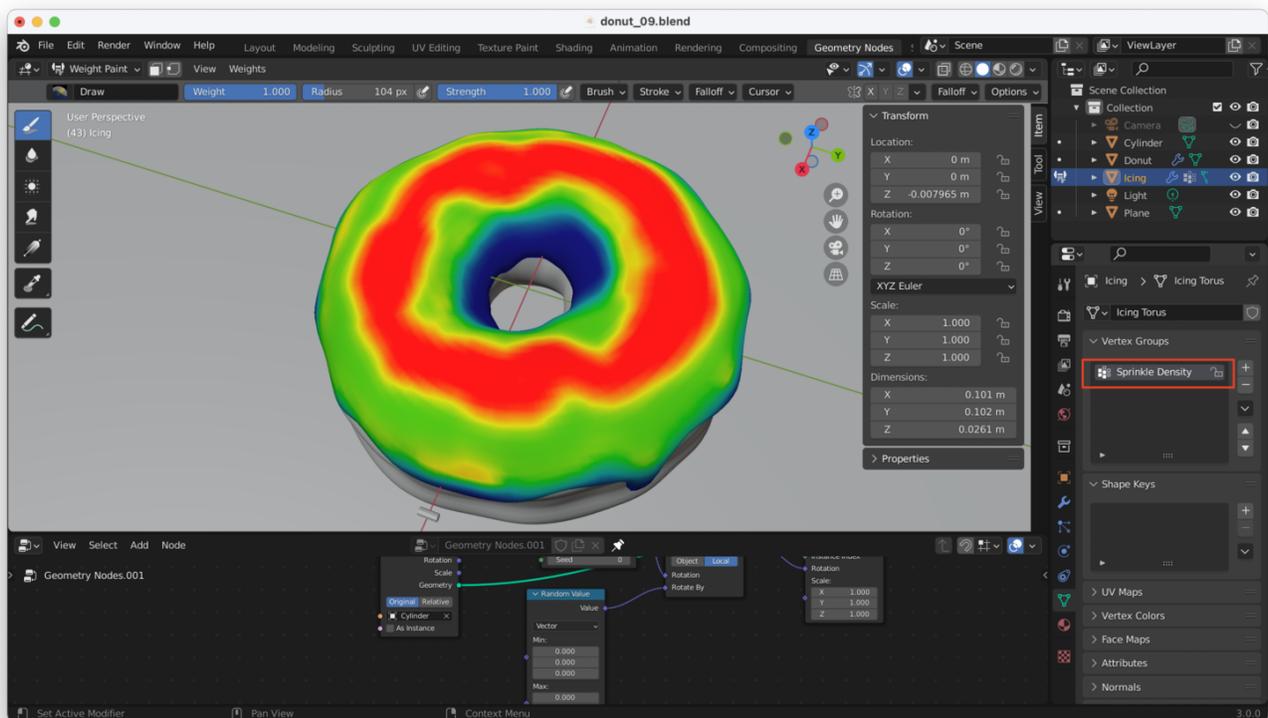


Tip: As with other painting and sculpting tools, you can change the brush size with "F" and the brush strength with "Control + F"

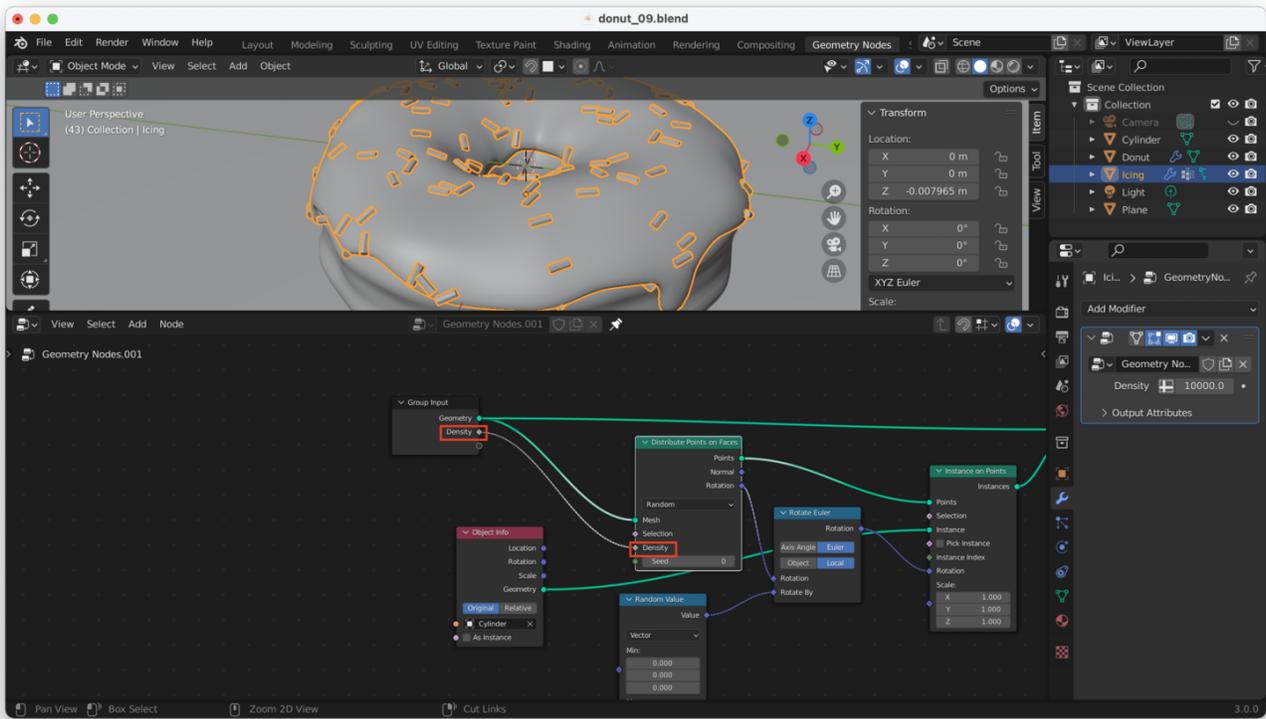
How can we get the sprinkles to use the new "Vertex Group" that was created by Weight Painting? We can see it in the "Vertex Groups" side-pane, as shown here:



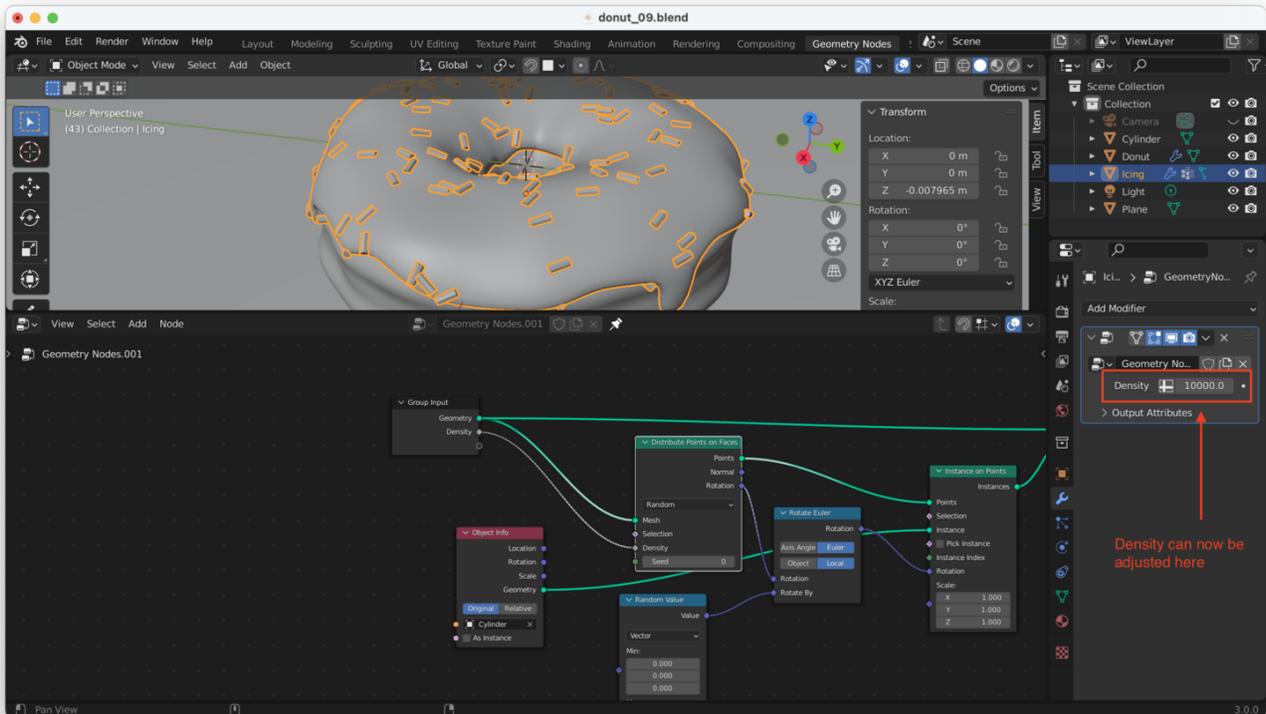
It might be a good idea to rename it, too!



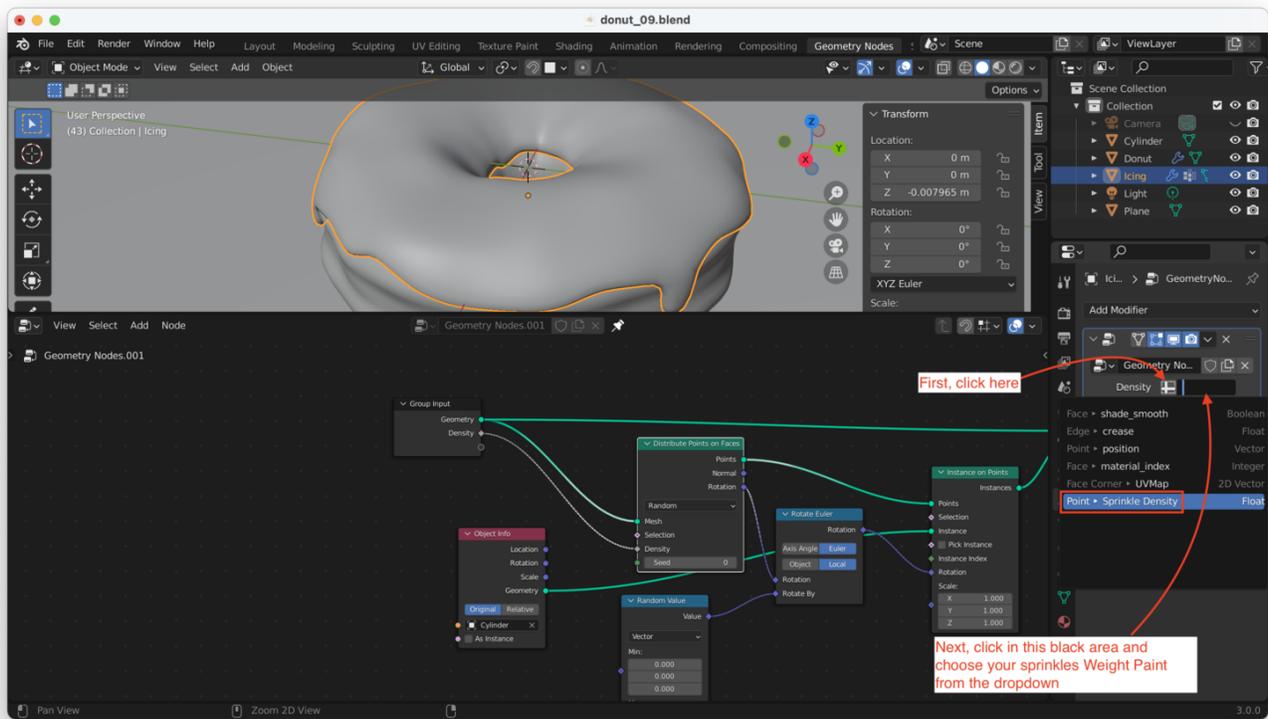
Now, we need a way to take the results of our Weight Paint and apply them to the sprinkles. We start by dragging a new wire from "Density" on the "Distribute Points on Faces" node, over to one of the blank inputs on "Group input", like this:



What this does is "expose" this density value, so it shows up in your modifier stack on the right-hand side of the window, along with all our other modifiers. This means we can adjust the "Density" setting of "Distribute Points on Faces" directly, from the "modifiers" pane:

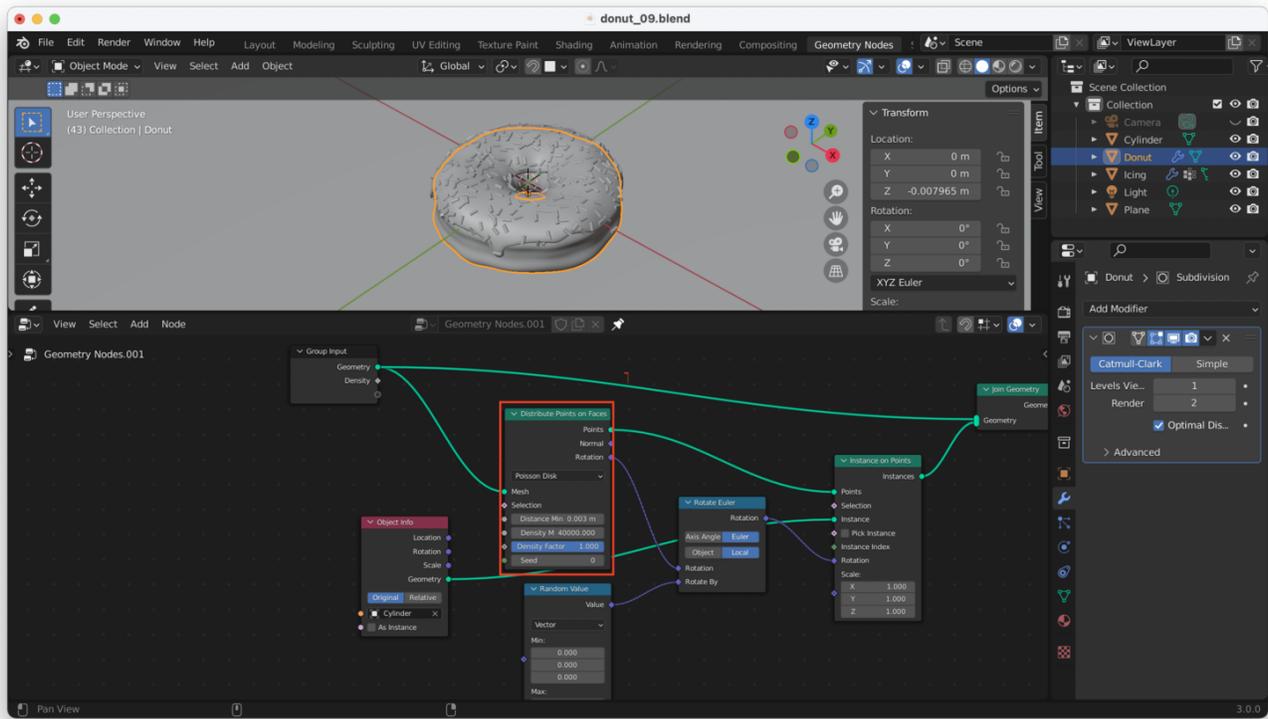


Better, using the little box next to "Density" in the "modifiers" pane, we can change the density from a constant value to the value of the Weight Paint:



After doing this, **nothing appears to happen!** Why is this? **Because the Weight Paint assigns weightings from 0 to 1, and we need values at or above 10,000.**

We can fix this by adjusting the settings of the "Distribute Points on Faces" node. We'll switch from "Random" to "Poisson Disk":



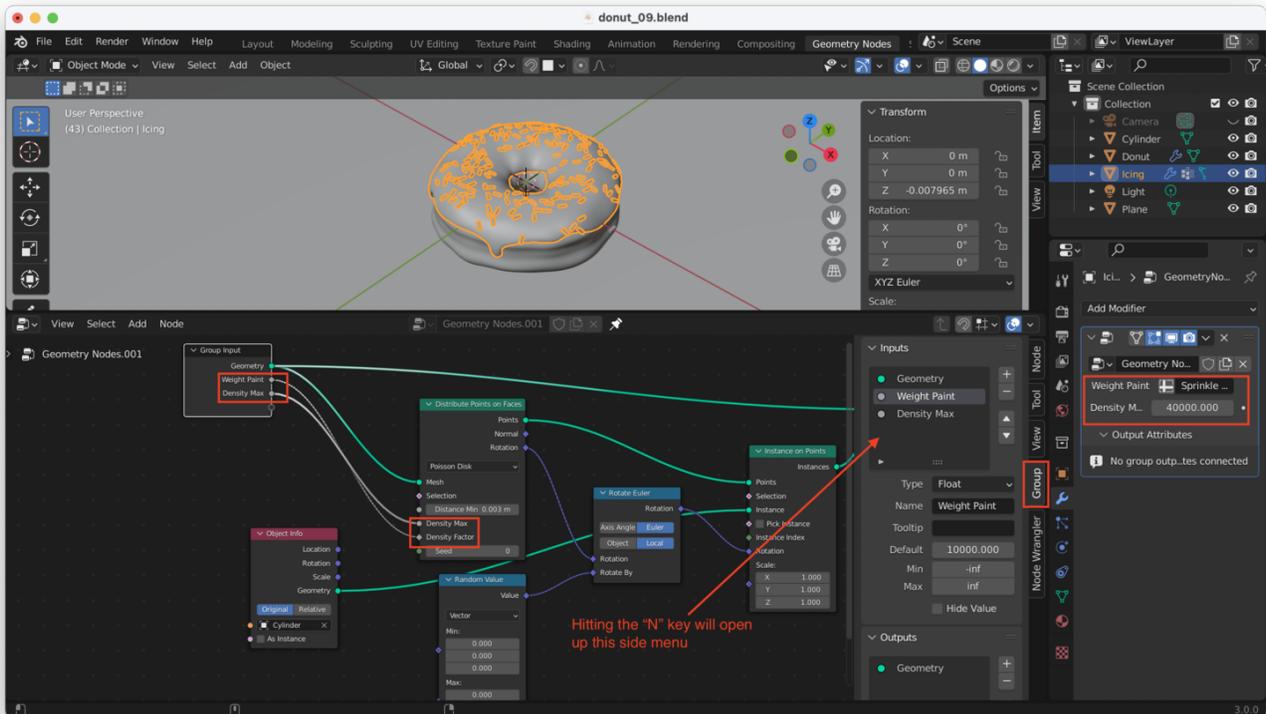
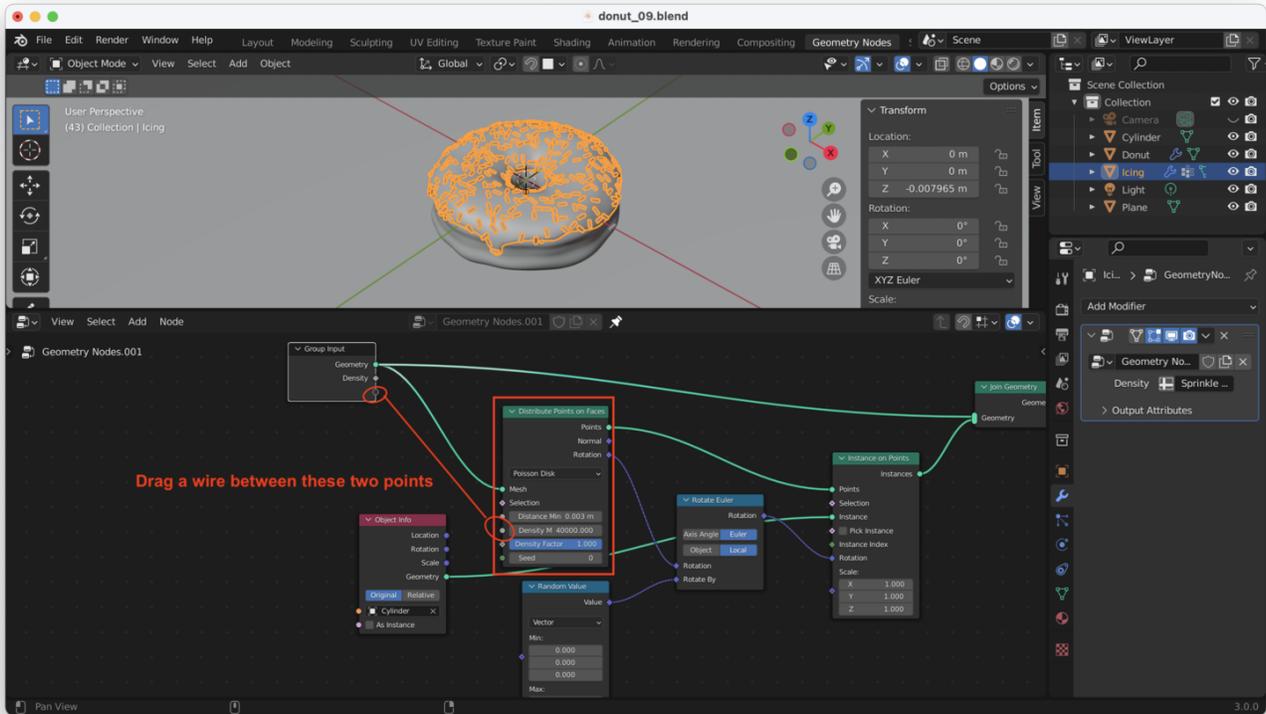
You can see this gives us additional options to play with, including:

- **Density Max**, which allows us to choose the maximum density value (we can set a high value here - like 40,000 - to generate more sprinkles)
- **Density Factor**, which allows you to multiply the "Density Max" value by some other factor: in this case, we can set the Density Factor to the output of our Weight Paint!
- **Distance Min**, which allows us to set a radius around each sprinkle, inside which other points (meaning other sprinkles) will not be generated: this helps us avoid having sprinkles "clip" through one another
- **Seed**, which chooses a random seed which affects how the sprinkles will be distributed (you can play with this value until you get a distribution of sprinkles that you like the look of!)

However, we're **still not done**. If you look closely, you'll see that the sprinkles are nicely distributed and rotated now, but we still aren't using our **Weight Paint** from earlier. We need to add a few more nodes to make that happen.

1. Drag a wire from "Density Max" on the "Distribute Points on Faces" node, to the empty input at the bottom of the "Group Input node"
2. Hit "N" from the Geometry Nodes editor window to bring up the side panel, then click on "Groups"
3. Rename the two inputs that appear there: the "Density" input from earlier can be renamed to "Weight Paint" to make its function clearer, and the new input can be called something like "Density Max"

I do my best to show all 4 steps in the two screenshots below:



As once more "nice thing" just to clean up a little more and make the "Density Max" values easier to change, we can set Density Max to a smaller value (say "2"), and add a "Math Node" between the "Density" output on our "Group Input" node and the "Density Max" input on our "Distribute Points on Faces" node, like this:

