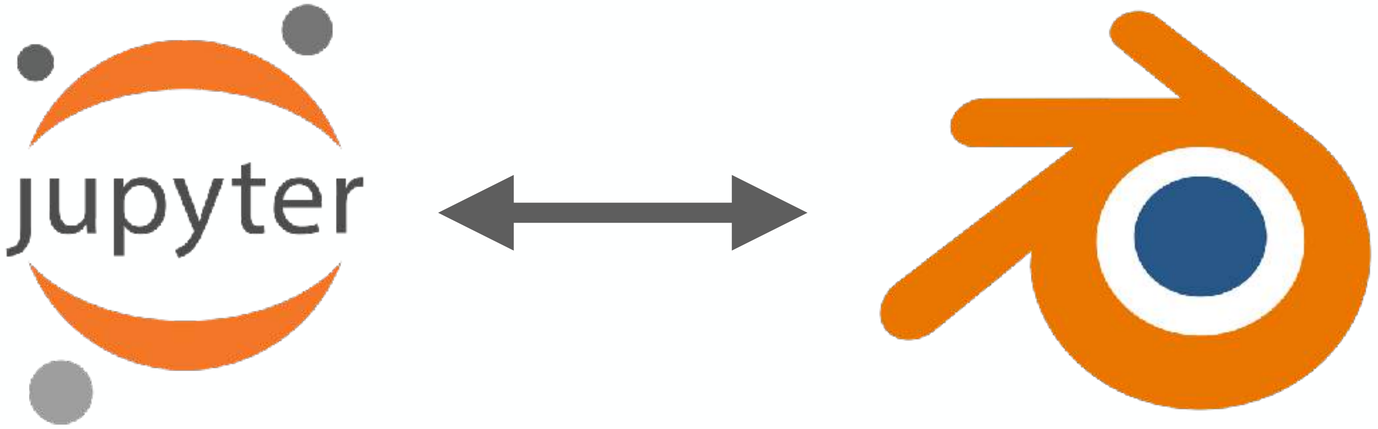
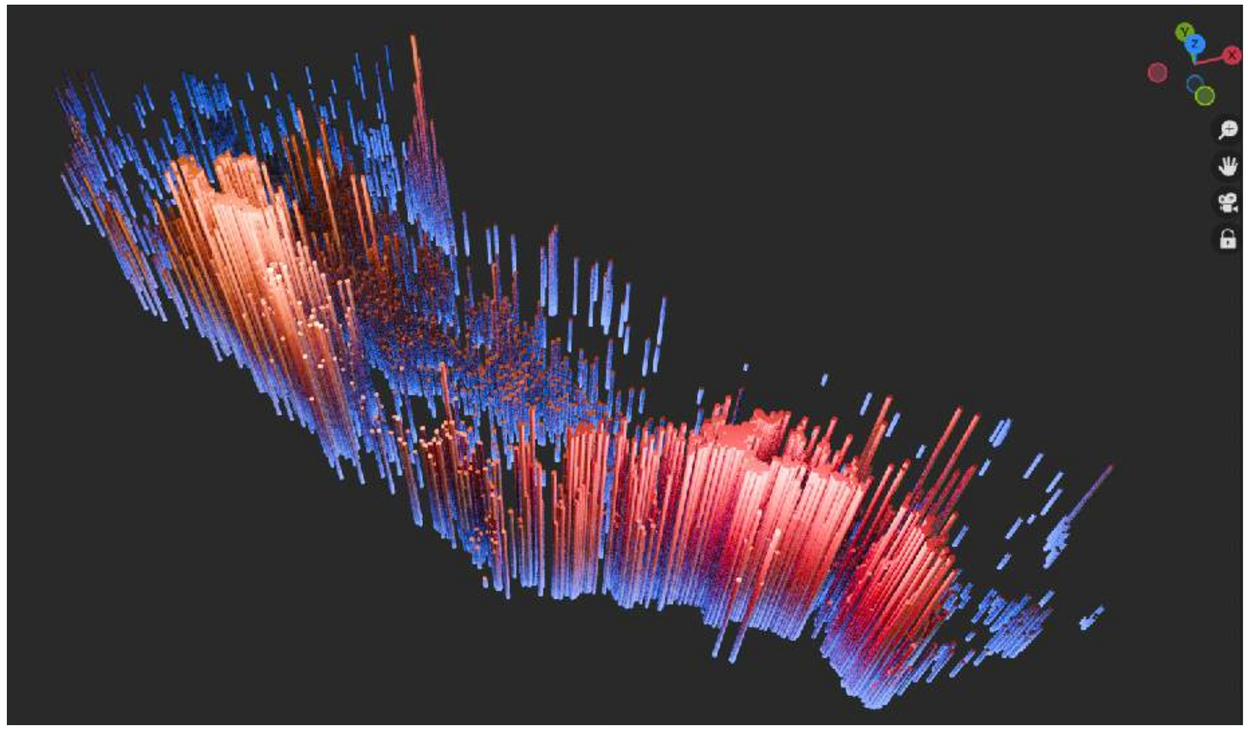


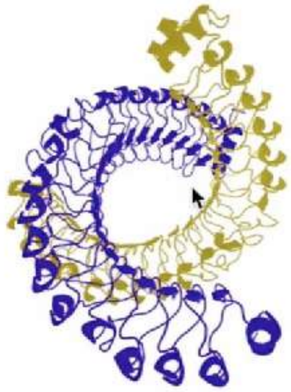
Jan-Hendrik Müller
16.01.2025,
14:00-14:45

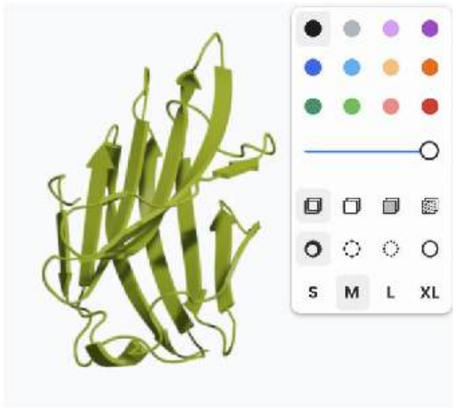
Data visualisation using Blender within Jupyter notebooks

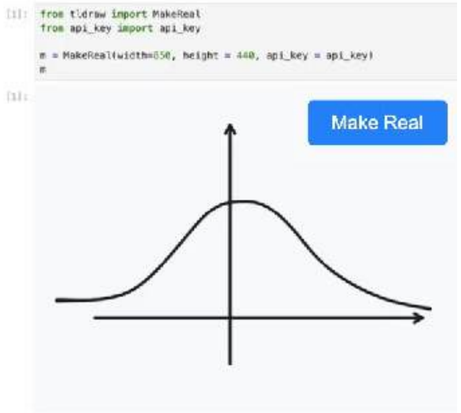


About me

- Biophysicist from Göttingen
- Open Source Contributor to Data visualization projects







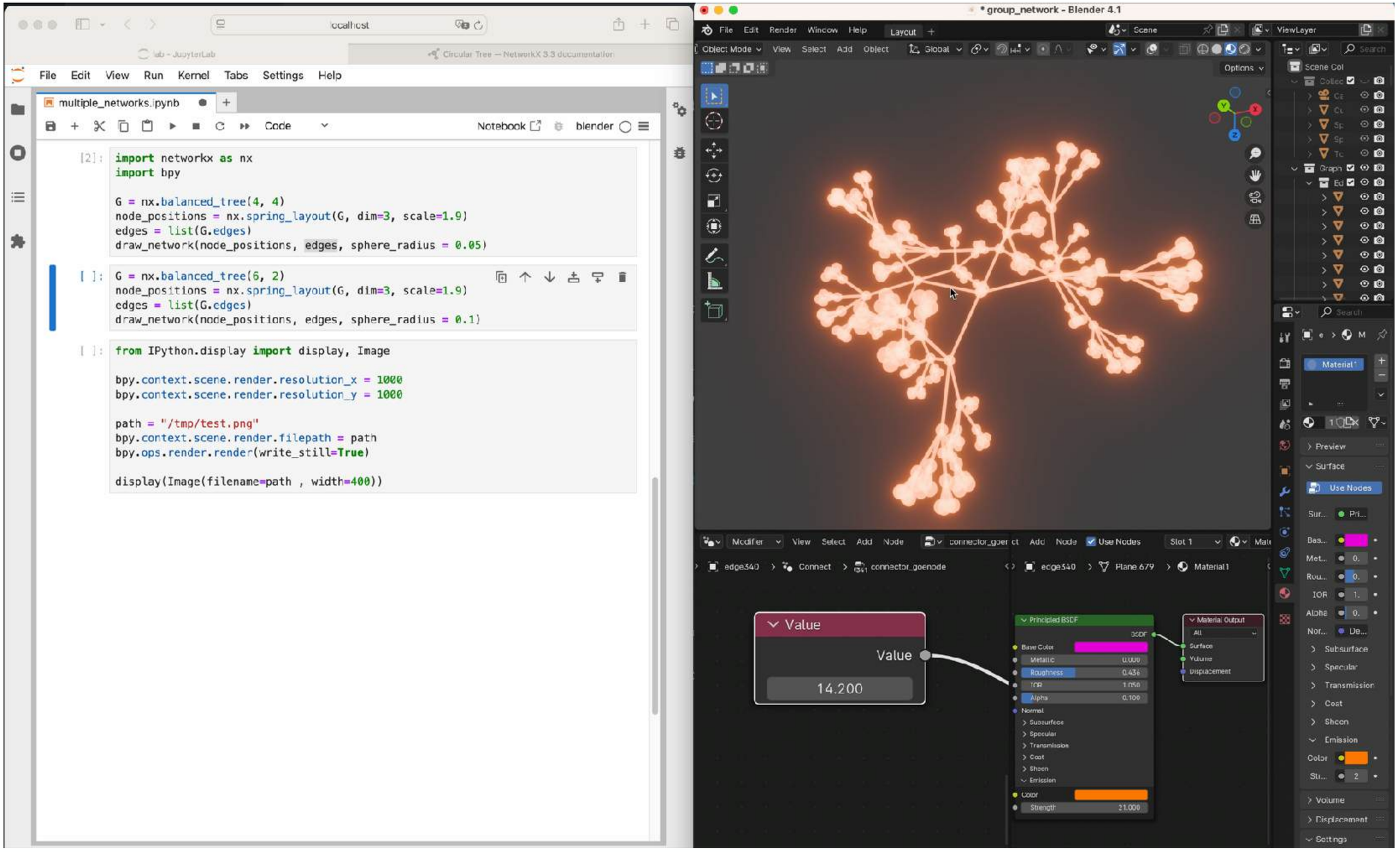
Jan-Hendrik Müller
16.01.2025,
14:00-14:45

https://www.linkedin.com/in/jan-hendrik-müller-765014209/
https://bsky.app/profile/kolibril13.bsky.social

supported in 2024 by

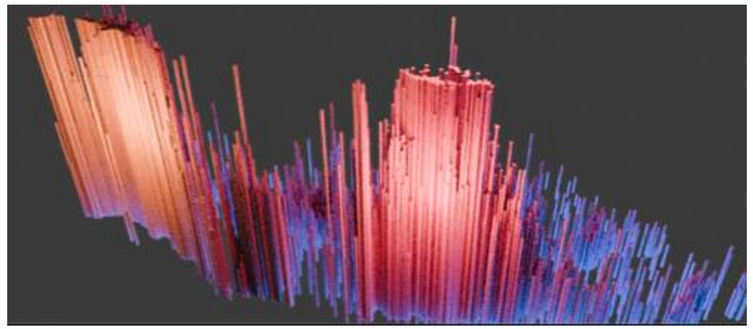
Prototype Fund

Federal Ministry of Education and Research

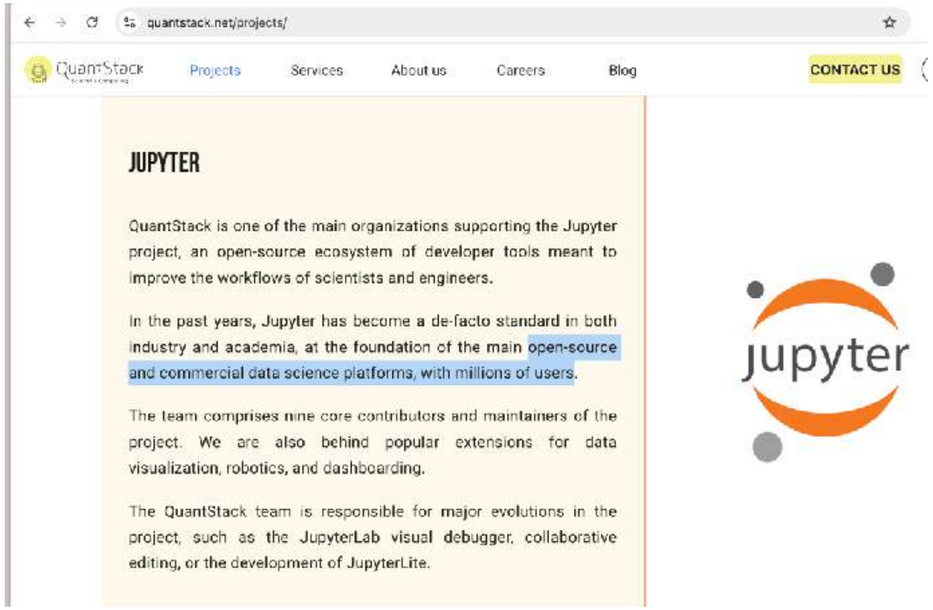

File Edit View Run Kernel Tabs Settings Help
multiple_networks.ipynb
Code
Notebook
blender
[2]:
import networkx as nx
import bpy
G = nx.balanced_tree(4, 4)
node_positions = nx.spring_layout(G, dim=3, scale=1.9)
edges = list(G.edges)
draw_network(node_positions, edges, sphere_radius = 0.05)
[ ]:
G = nx.balanced_tree(6, 2)
node_positions = nx.spring_layout(G, dim=3, scale=1.9)
edges = list(G.edges)
draw_network(node_positions, edges, sphere_radius = 0.1)
[ ]:
from IPython.display import display, Image
bpy.context.scene.render.resolution_x = 1000
bpy.context.scene.render.resolution_y = 1000
path = "/tmp/test.png"
bpy.context.scene.render.filepath = path
bpy.ops.render.render(write_still=True)
display(Image(filename=path , width=400))
* group_network - Blender 4.1
File Edit Render Window Help Layout
Object Mode View Select Add Object
Global
Options
Modifier View Select Add Node
Use Nodes
Slot 1
edge340 Connect connector_goenode
edge340 Plane.679 Material1
Value
Value
14.200
Principled BSDF
Base Color
Metallic
Roughness
IOR
Alpha
Normal
Subsurface
Specular
Transmission
Coat
Sheen
Emission
Color
Strength
Material Output
Surface
Volume
Displacement
Material
Preview
Surface
Use Nodes
Subsurface
Specular
Transmission
Coat
Sheen
Emission
Volume
Displacement
Settings

Takeaway

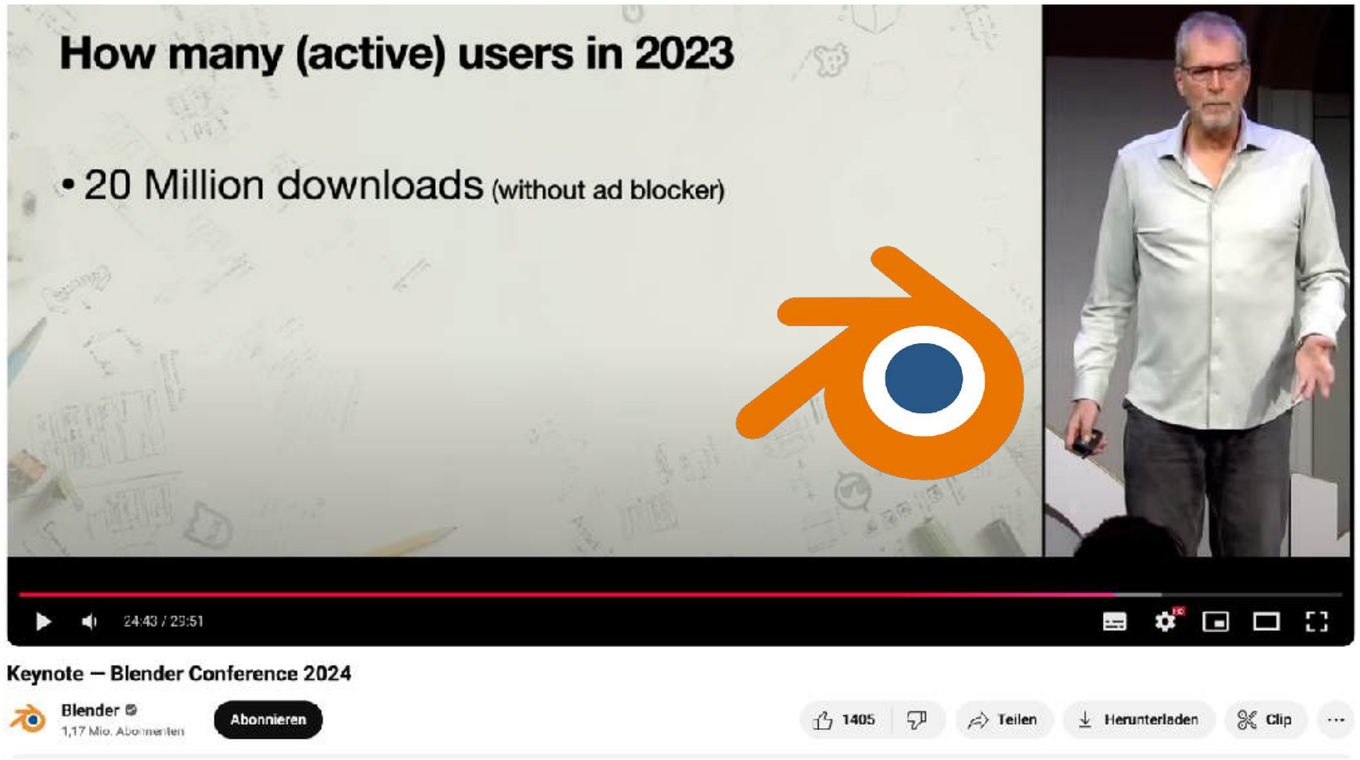
- Setup the connection
- Interact with objects
- Data processing pipeline
- Time series
- Links to resources



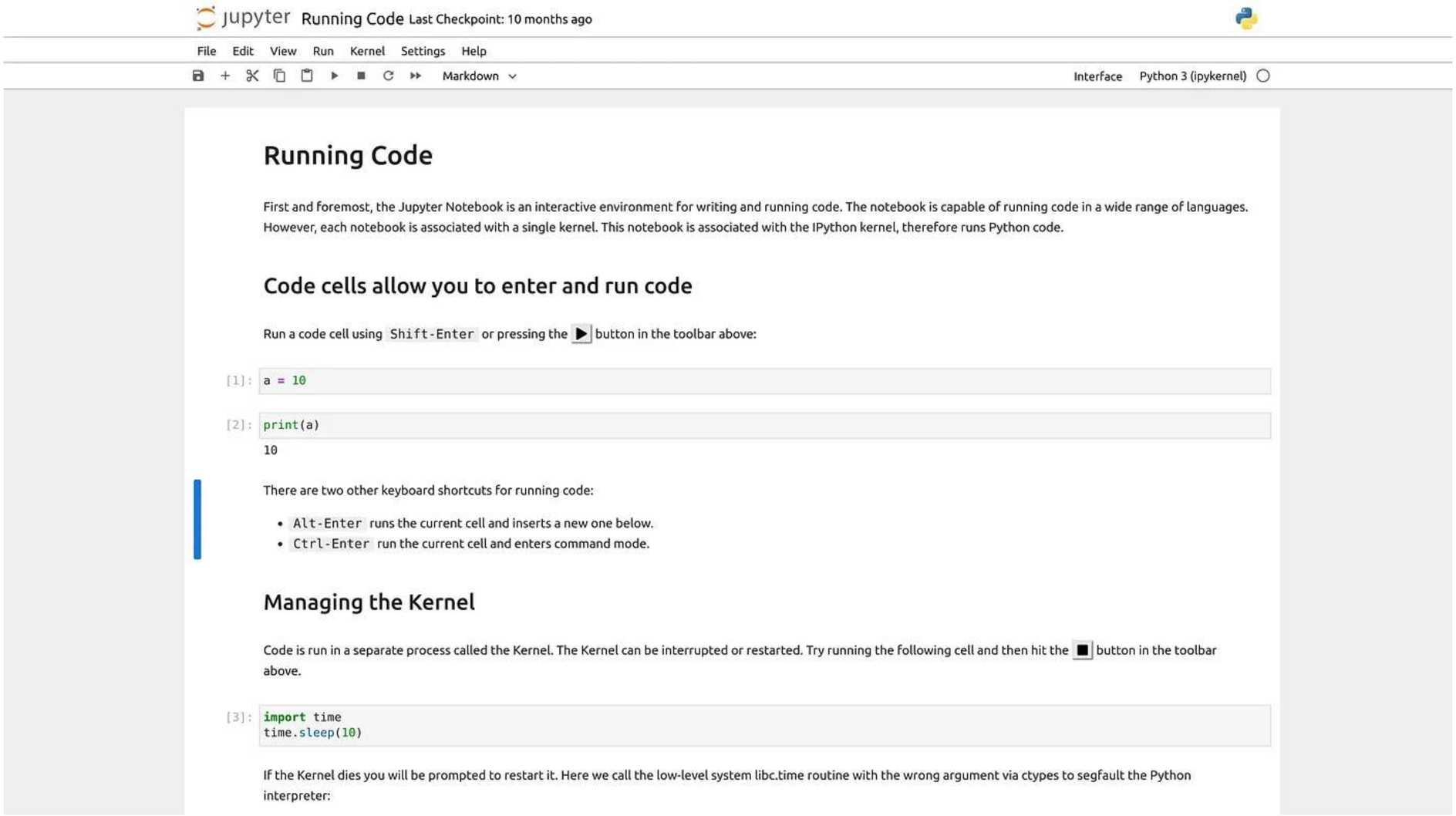
Users in Numbers



https://quantstack.net/projects/

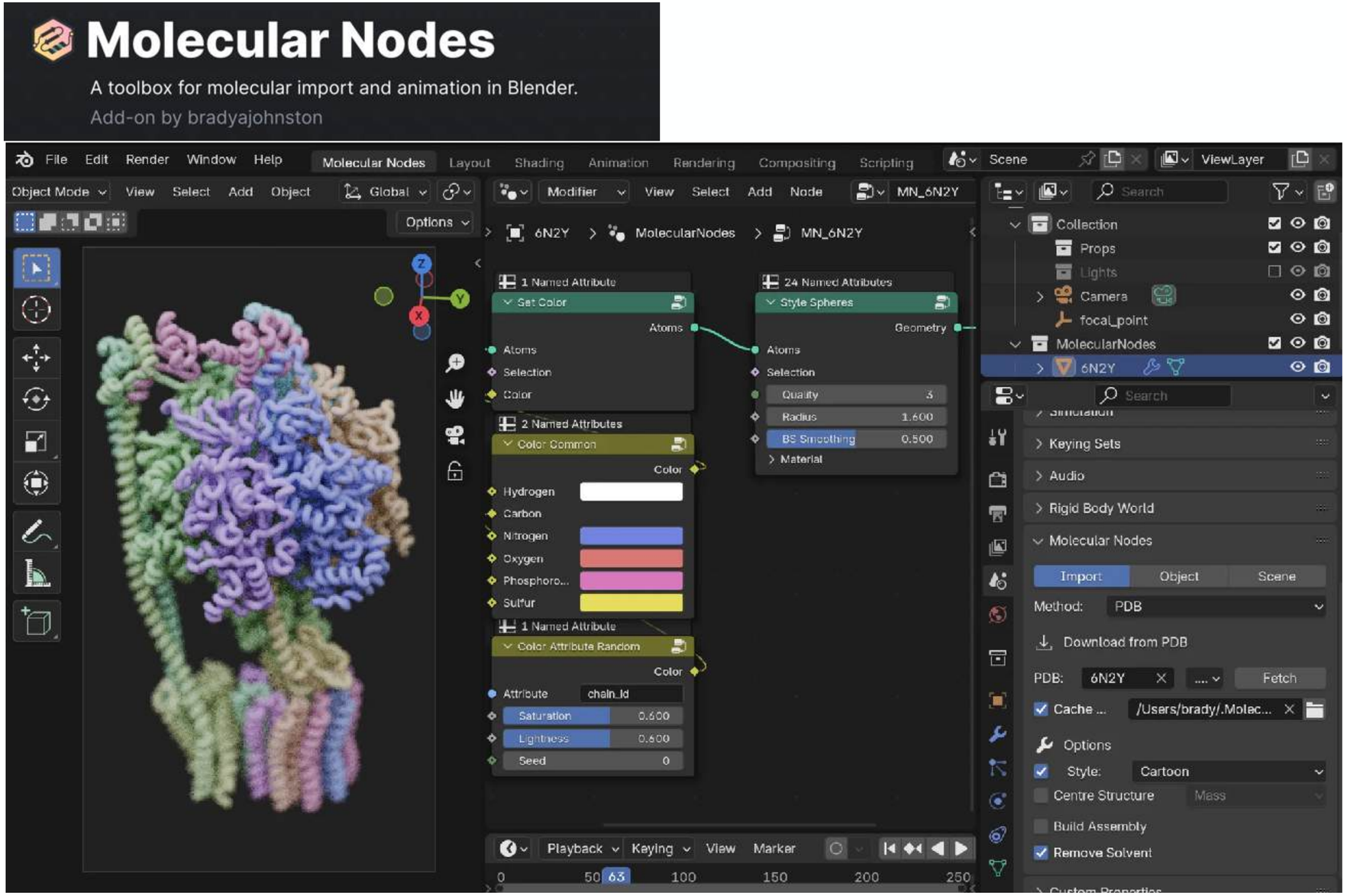


Python notebook



Blender in 100 seconds

Blender extensions



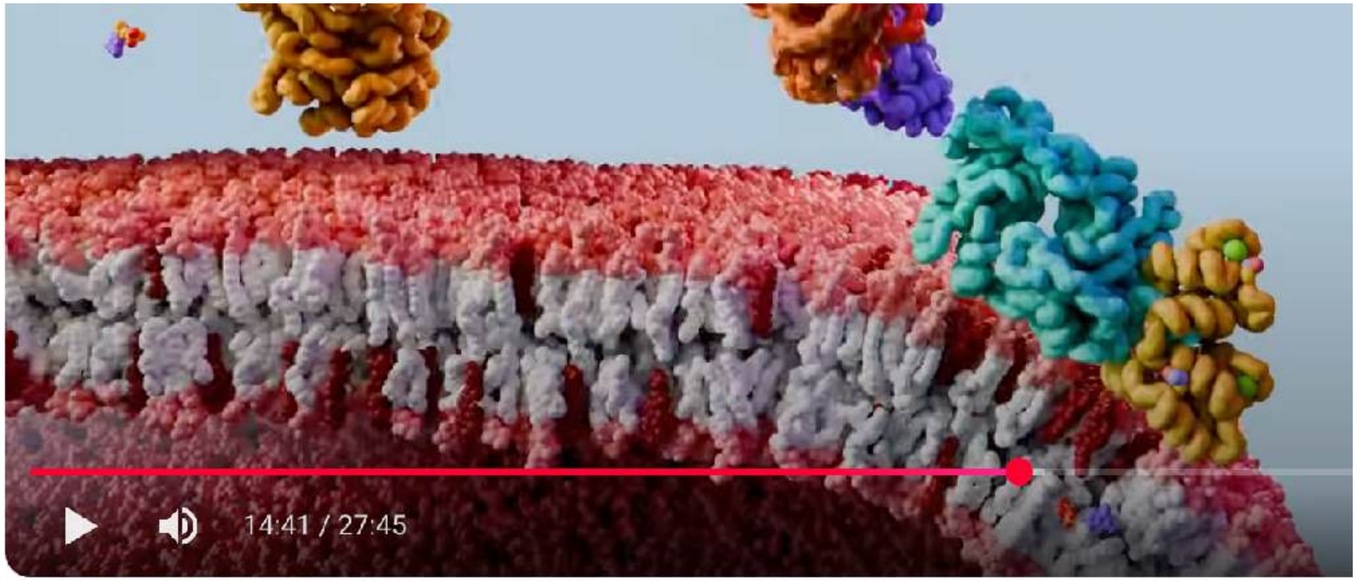
Blender extensions

Molecular Nodes

A toolbox for molecular import and animation in Blender.

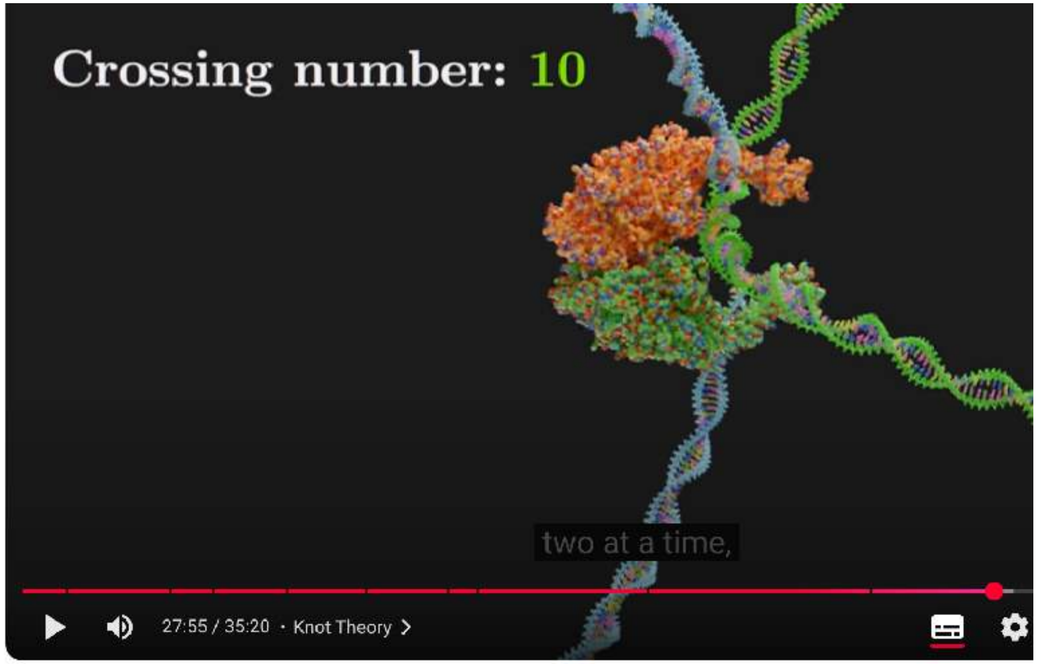
Add-on by bradyajohnston

https://bradyajohnston.github.io/MolecularNodes/examples/



Where the Light Touches Your Eyes | Phototransduction and Rhodopsi

Clockwork
44.5K subscribers
Join
Subscribe
4.6K

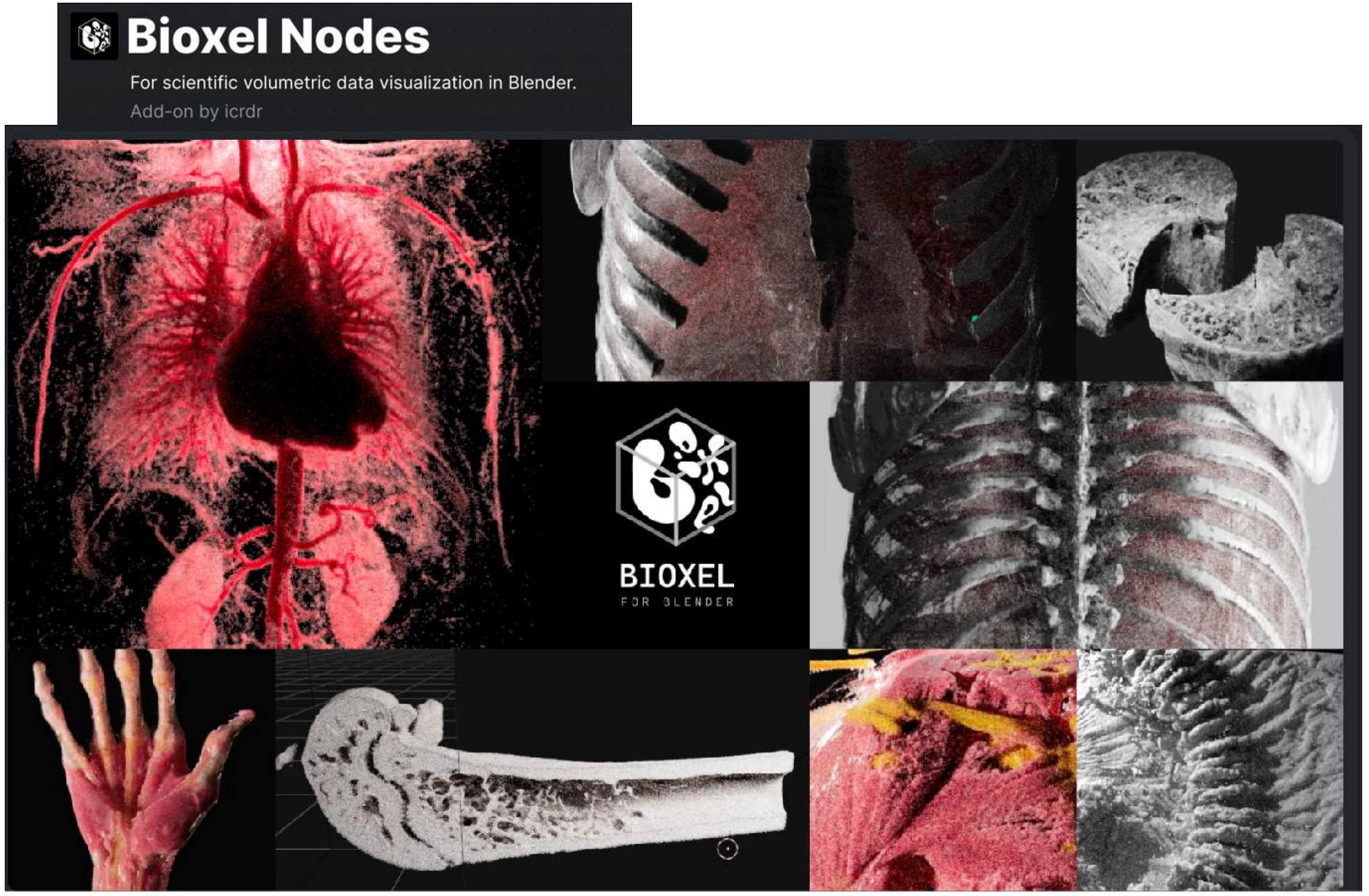


The Insane Math Of Knot Theory

Veritasium
17M subscribers
Subscribe
207K
Share

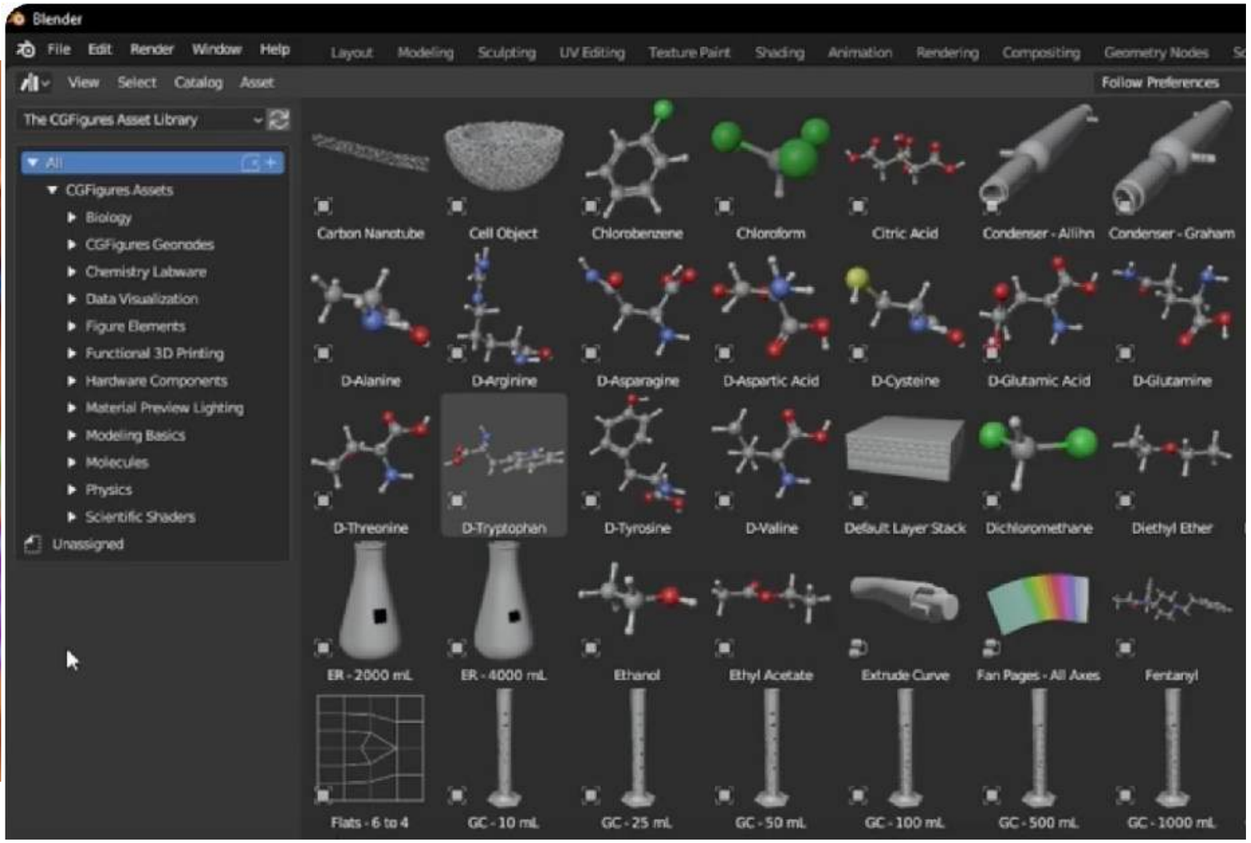
8M views 1 year ago

Blender extensions

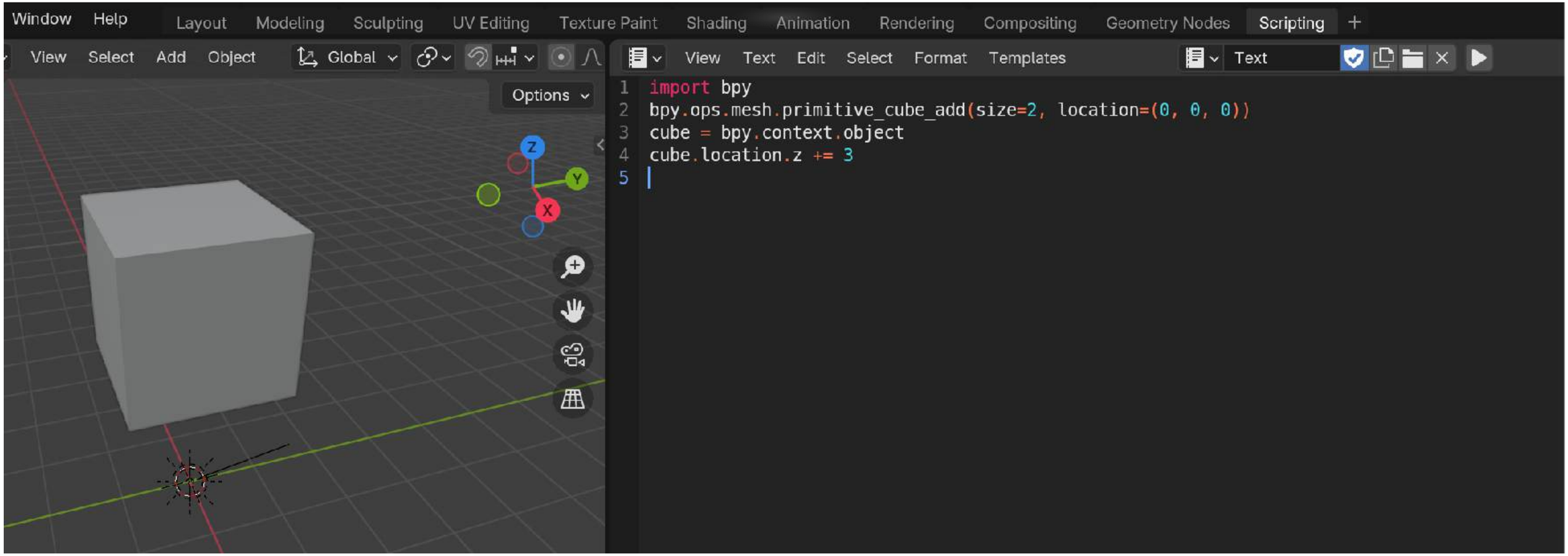


Blender extensions
free resources for scientific visualization

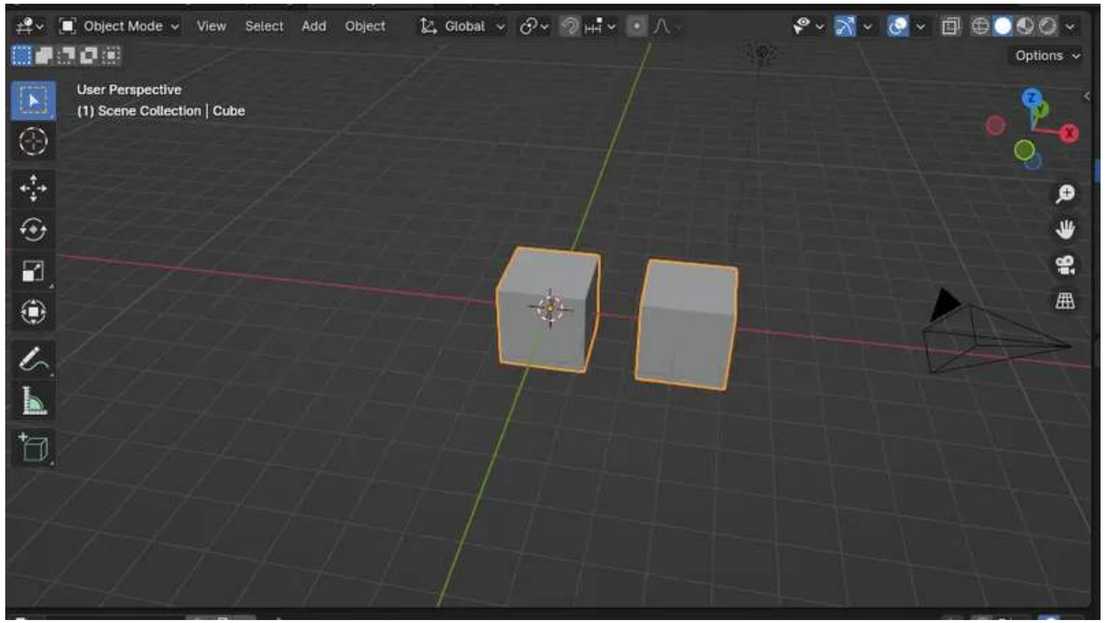
https://www.cgfigures.ca/assetlibrary

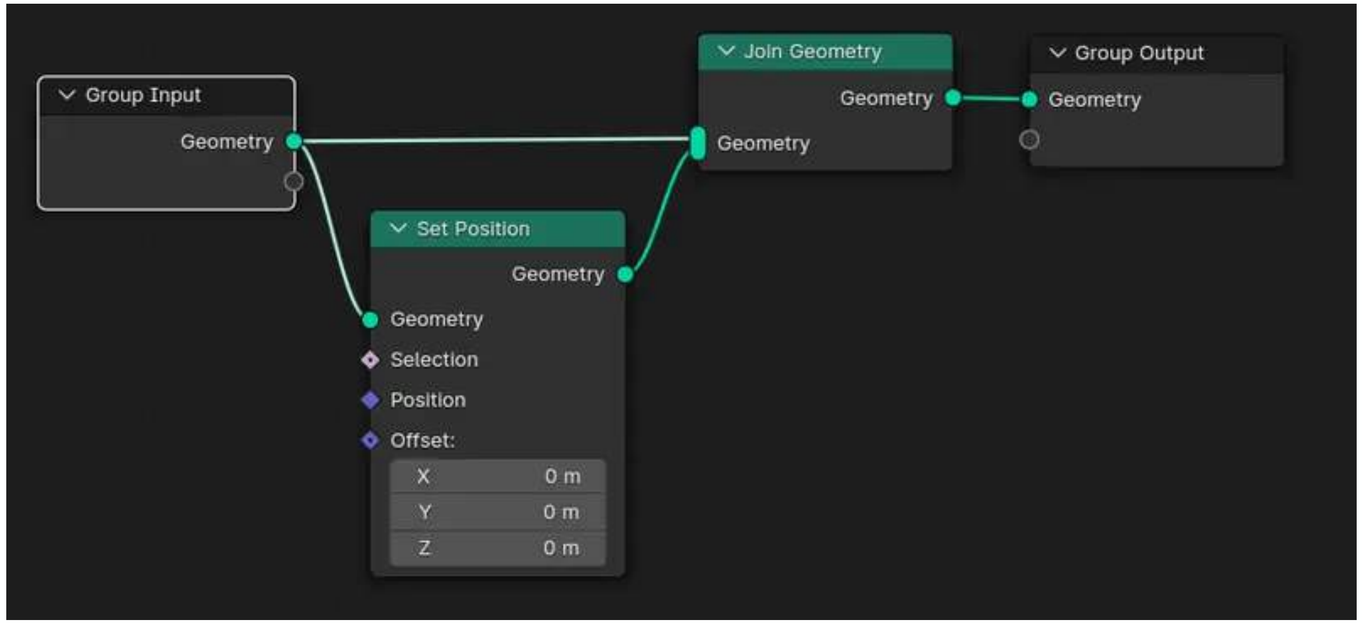


Blender Scripting

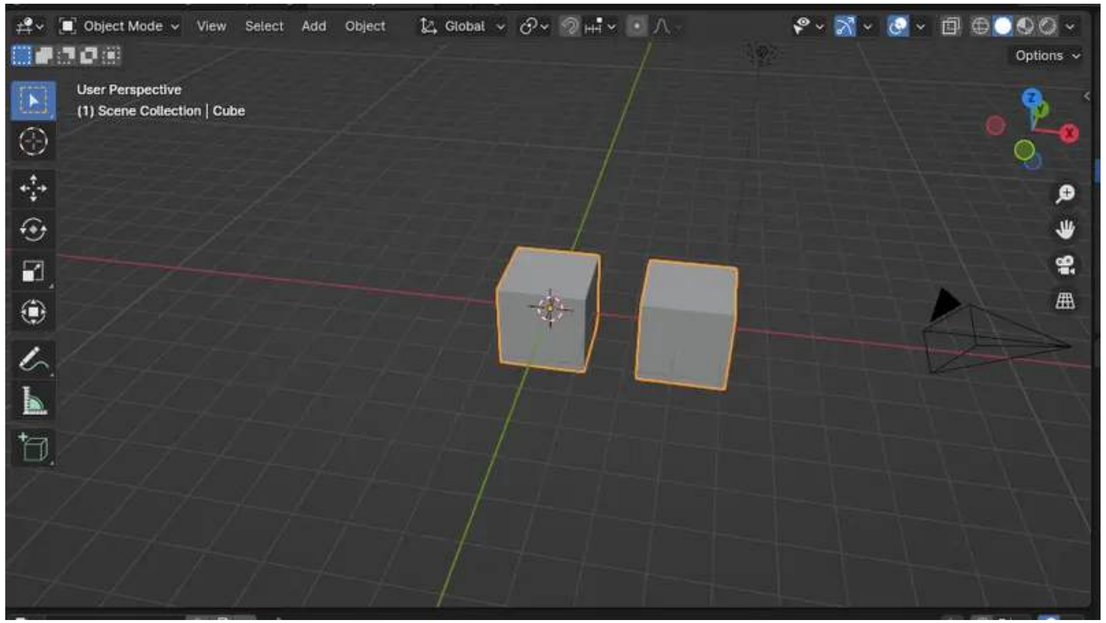


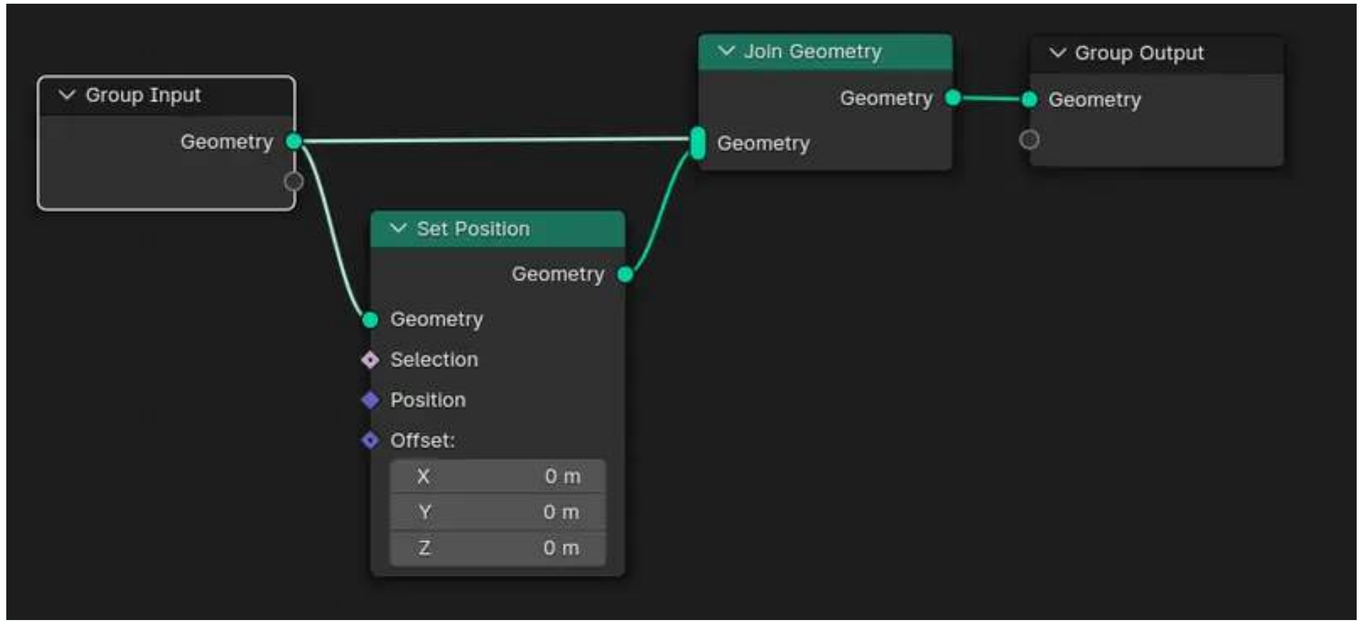
Blender Geometry Nodes



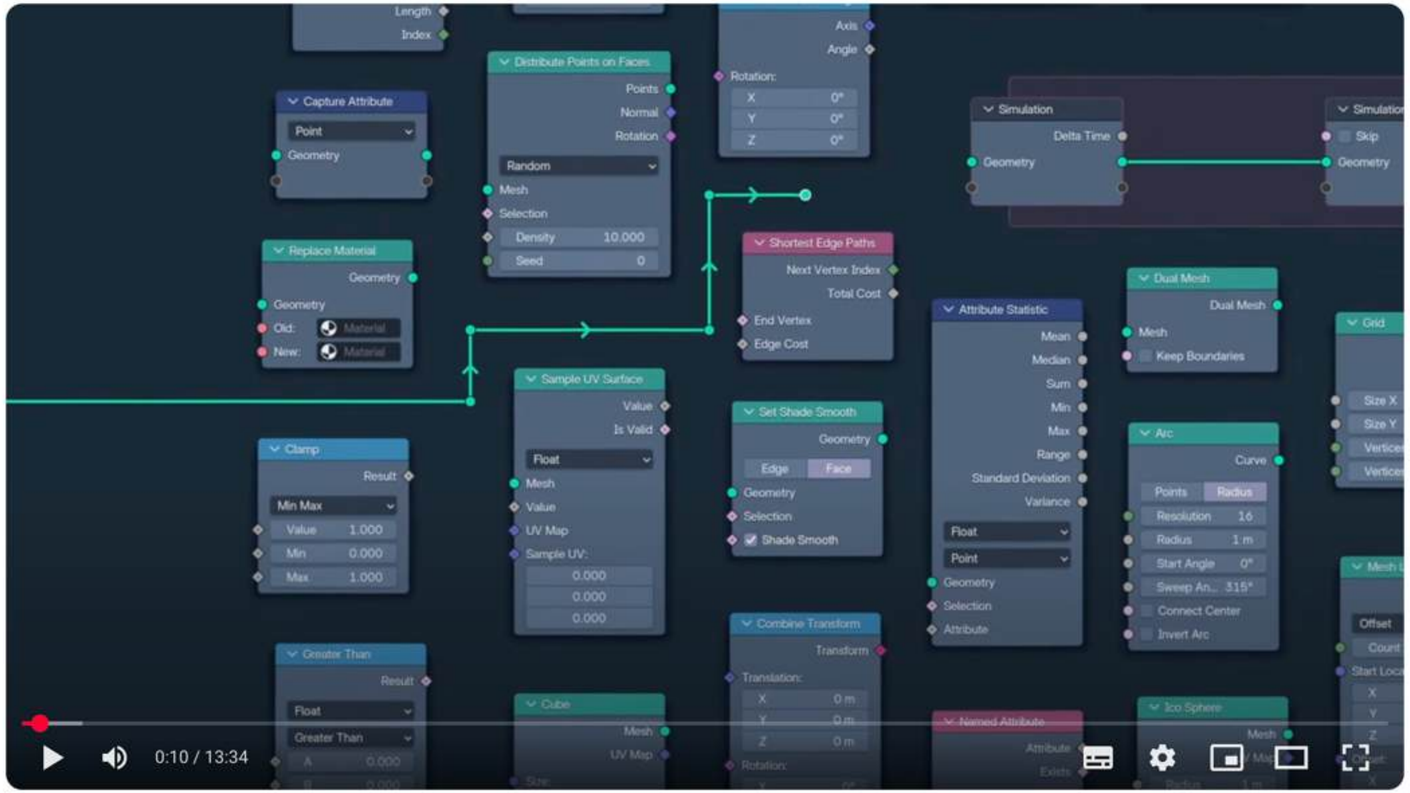


Blender Geometry Nodes





Blender Geometry Nodes Tutorial



A guide to attributes & fields - blender geometry nodes

harry blends
14.5K subscribers

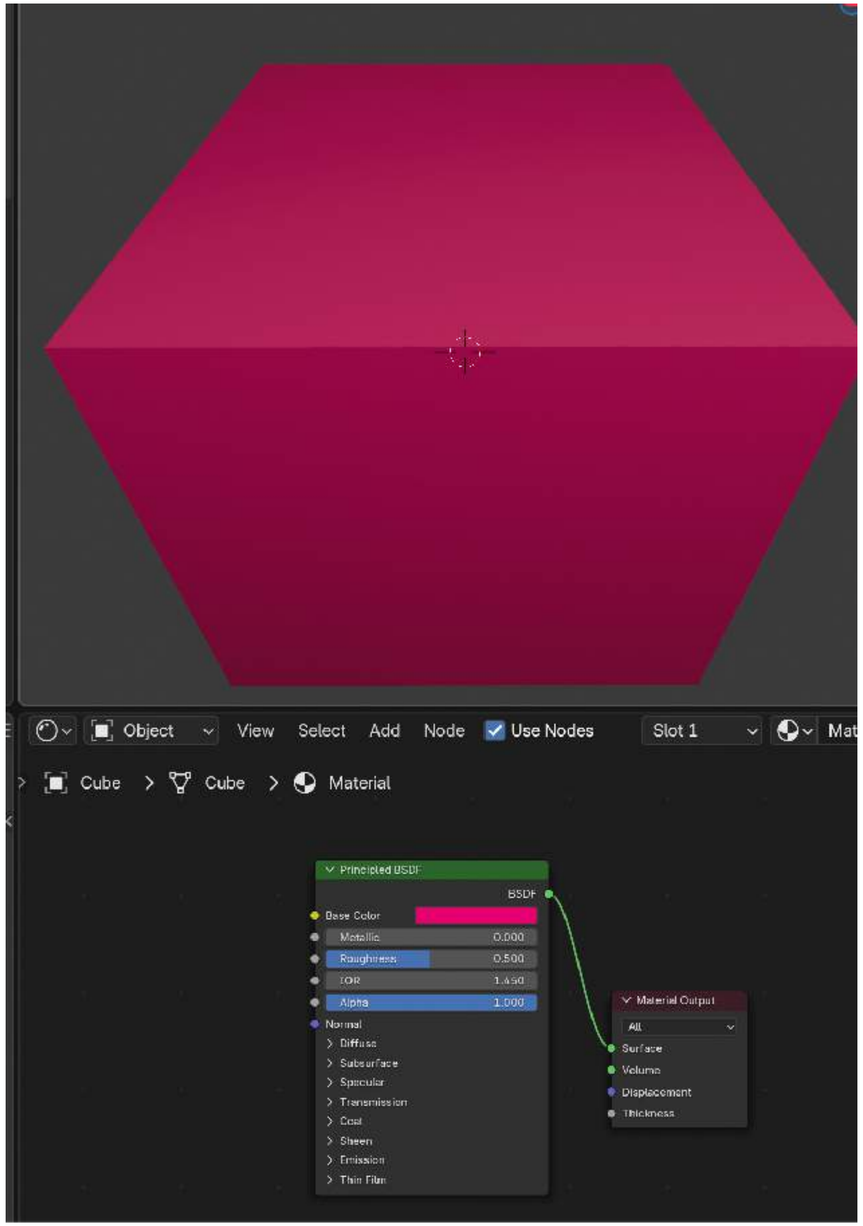
Subscribe

6.8K Share Download

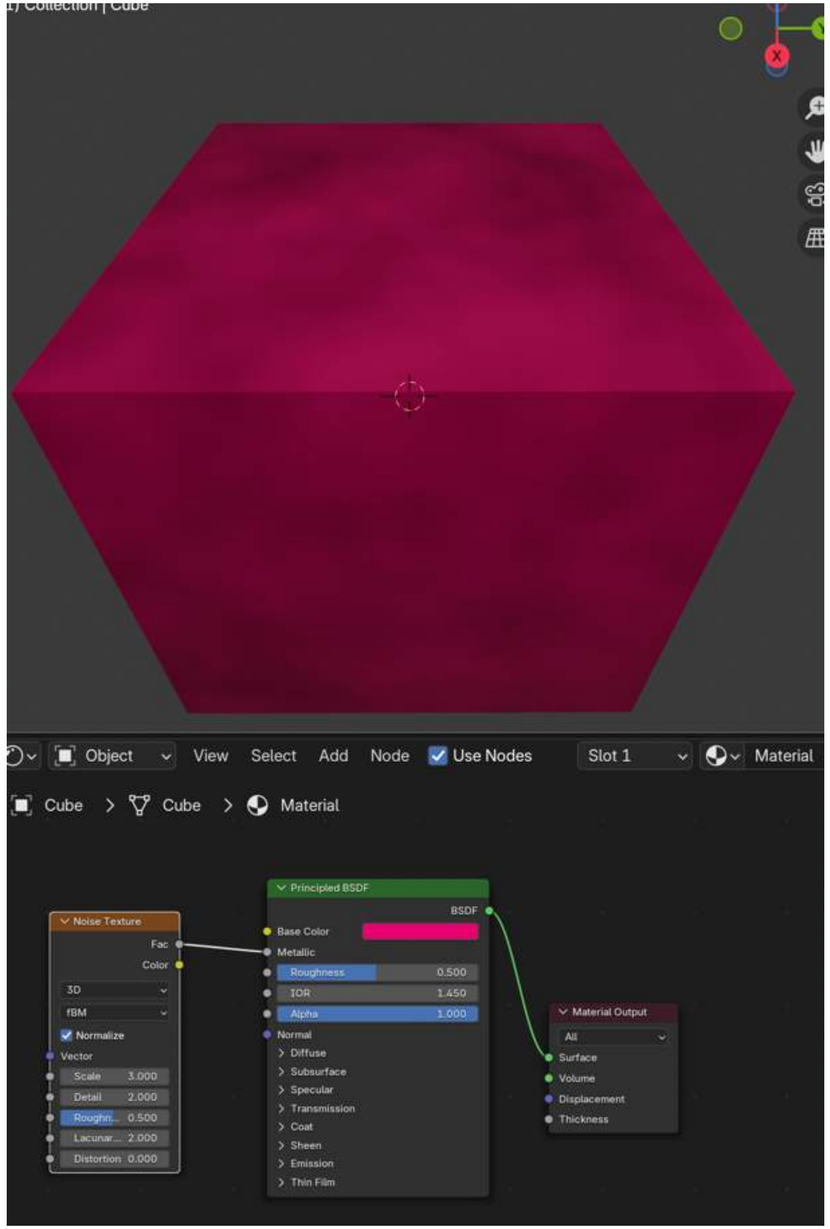
76K views 5 months ago #blenderanimation #blender3d #b3d

harry blends - https://www.youtube.com/watch?v=a-4oCHe-hDE&t=61s

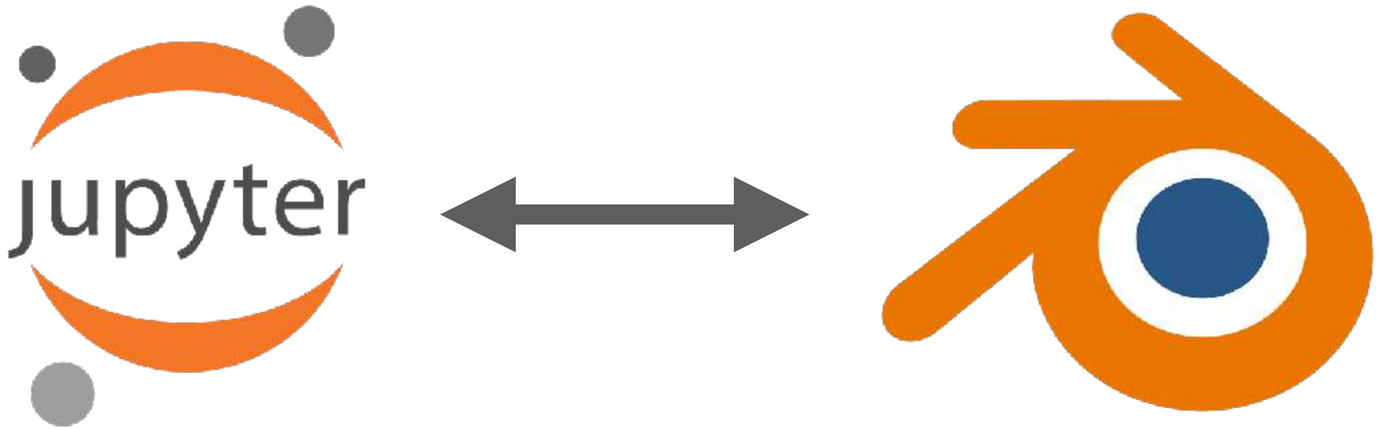
Shader Nodes



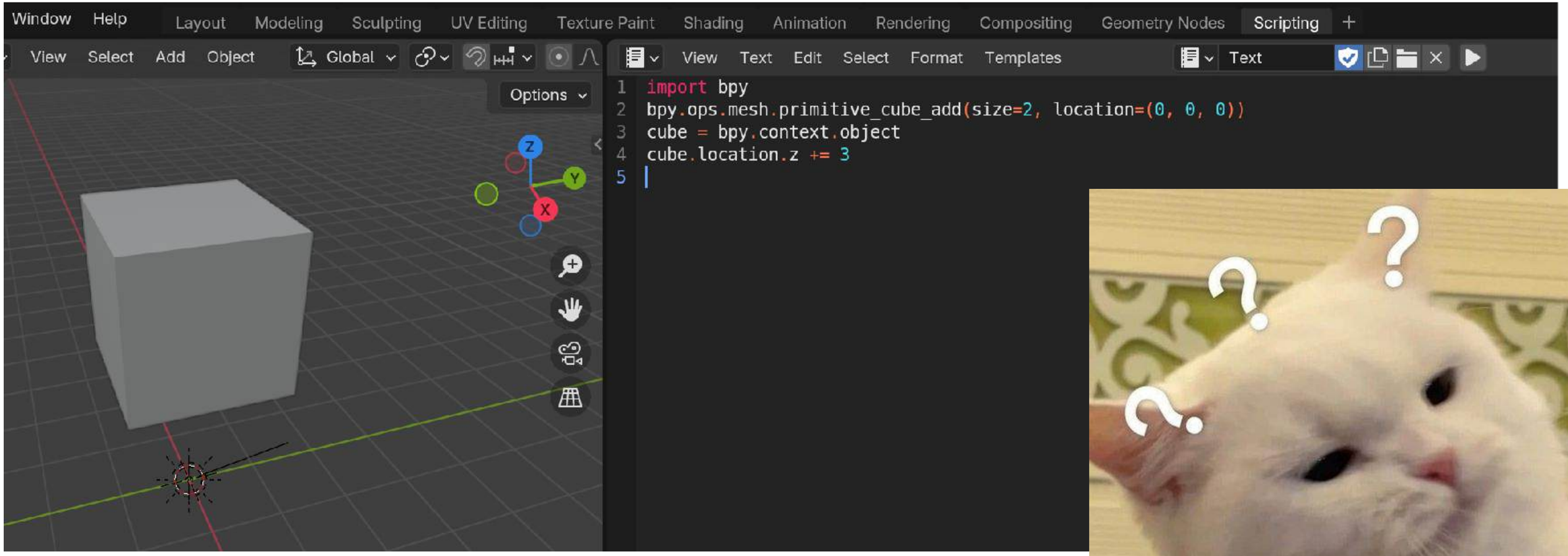
Shader Nodes



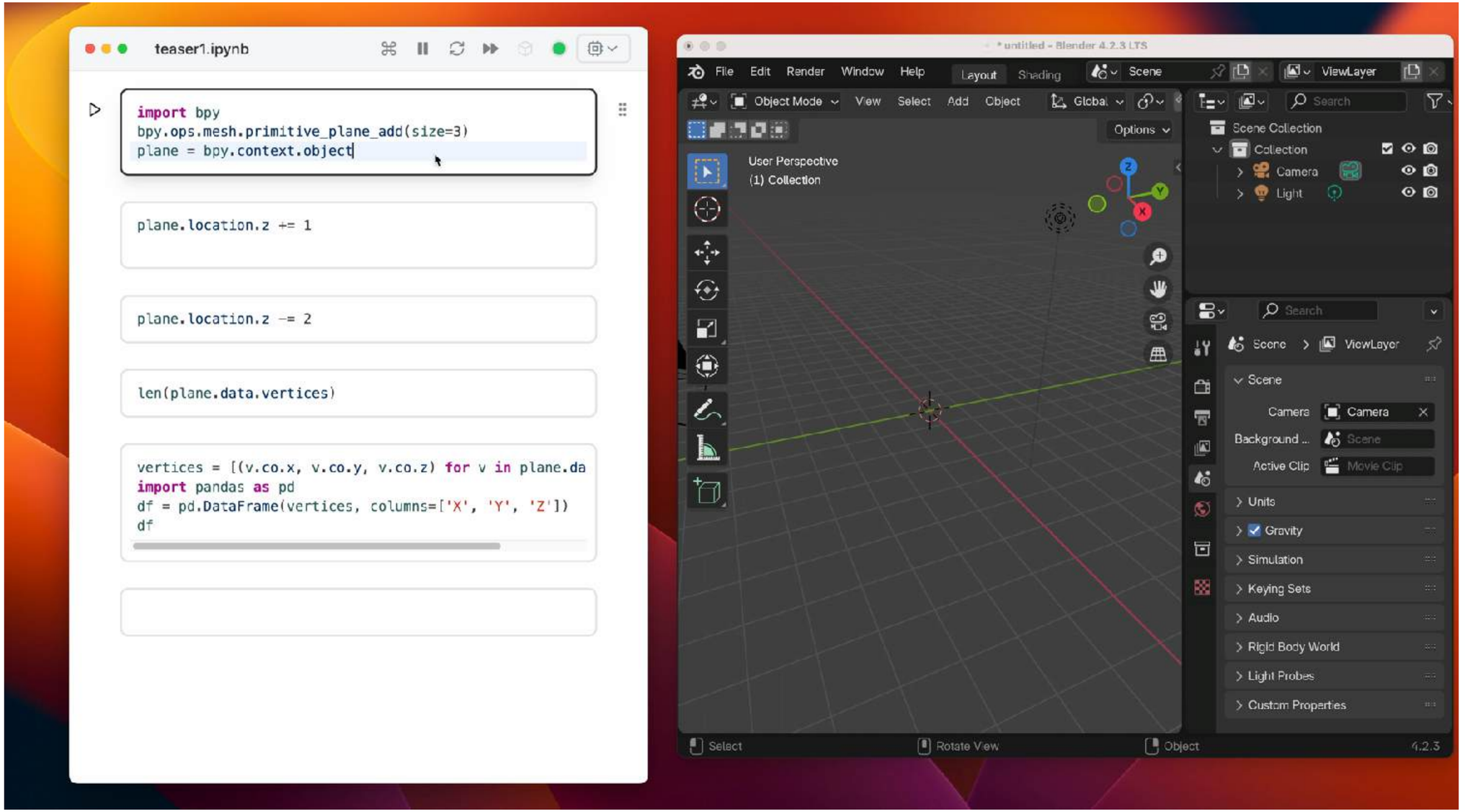
Blender and Notebooks

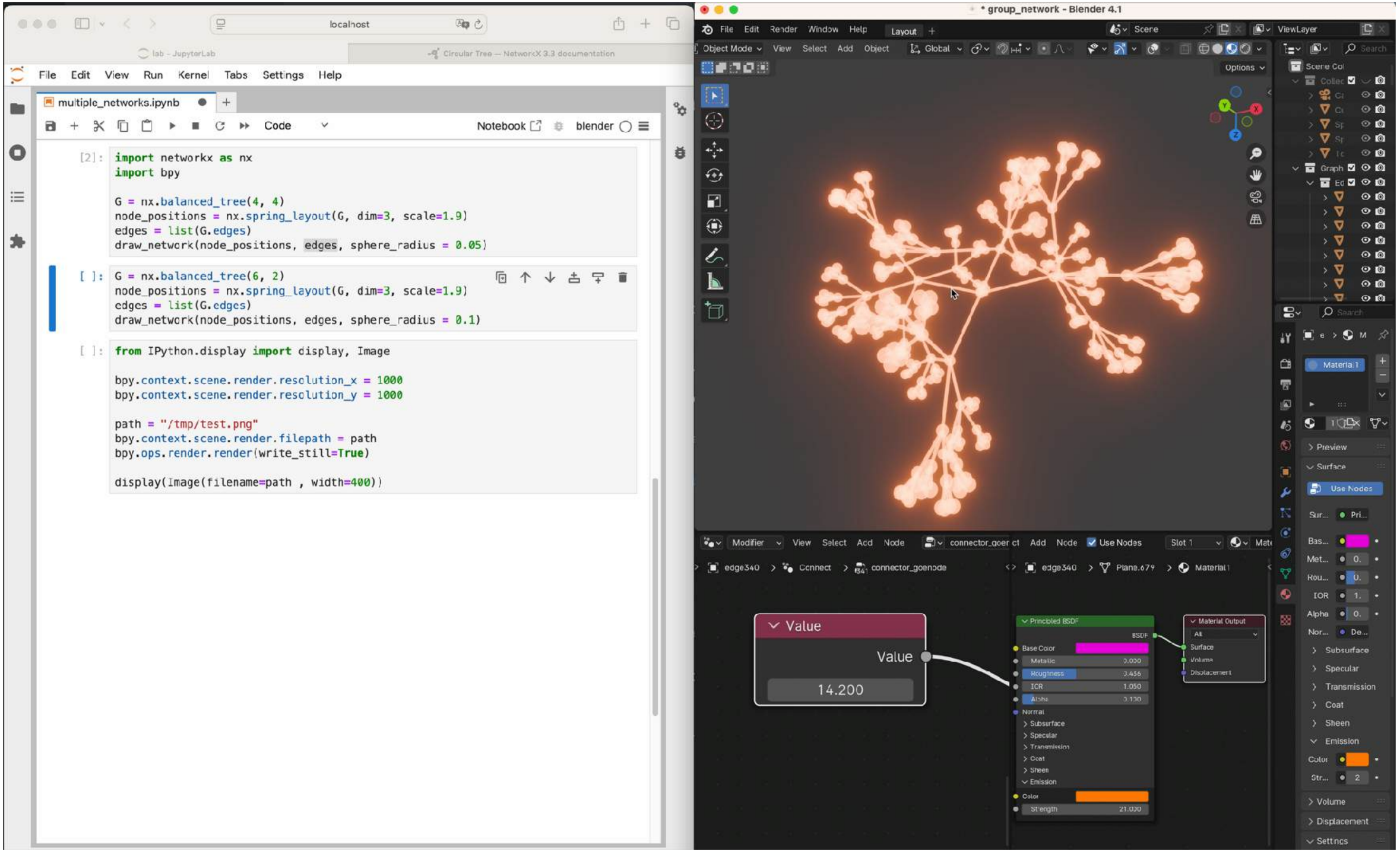


Blender Python editor



1. Iterative Workflows




localhost
lab - JupyterLab
Circular Tree — NetworkX 3.3 documentation
File Edit View Run Kernel Tabs Settings Help
multiple_networks.ipynb
Code
Notebook
blender
[2]:
import networkx as nx
import bpy
G = nx.balanced_tree(4, 4)
node_positions = nx.spring_layout(G, dim=3, scale=1.9)
edges = list(G.edges)
draw_network(node_positions, edges, sphere_radius = 0.05)
[ ]:
G = nx.balanced_tree(6, 2)
node_positions = nx.spring_layout(G, dim=3, scale=1.9)
edges = list(G.edges)
draw_network(node_positions, edges, sphere_radius = 0.1)
[ ]:
from IPython.display import display, Image
bpy.context.scene.render.resolution_x = 1000
bpy.context.scene.render.resolution_y = 1000
path = "/tmp/test.png"
bpy.context.scene.render.filepath = path
bpy.ops.render.render(write_still=True)
display(Image(filename=path , width=400))
* group_network - Blender 4.1
File Edit Render Window Help Layout
Object Mode View Select Add Object
Options
Modifier View Select Add Node
Use Nodes
edge340 Connect connector_goenode
edge340 Plane.679 Material.1
Value
Value
14.200
Principled BSDF
Base Color
Metallic 0.000
Roughness 0.456
IOR 1.050
Alpha 0.100
Normal
Subsurface
Specular
Transmission
Coat
Sheen
Emission
Color
Strength 21.000
Material Output
All
Surface
Volume
Displacement
Material.1
Preview
Surface
Use Nodes
Subsurface
Specular
Transmission
Coat
Sheen
Emission
Volume
Displacement
Settings

n3c_single_mesh_netwokx.ipynb

blender

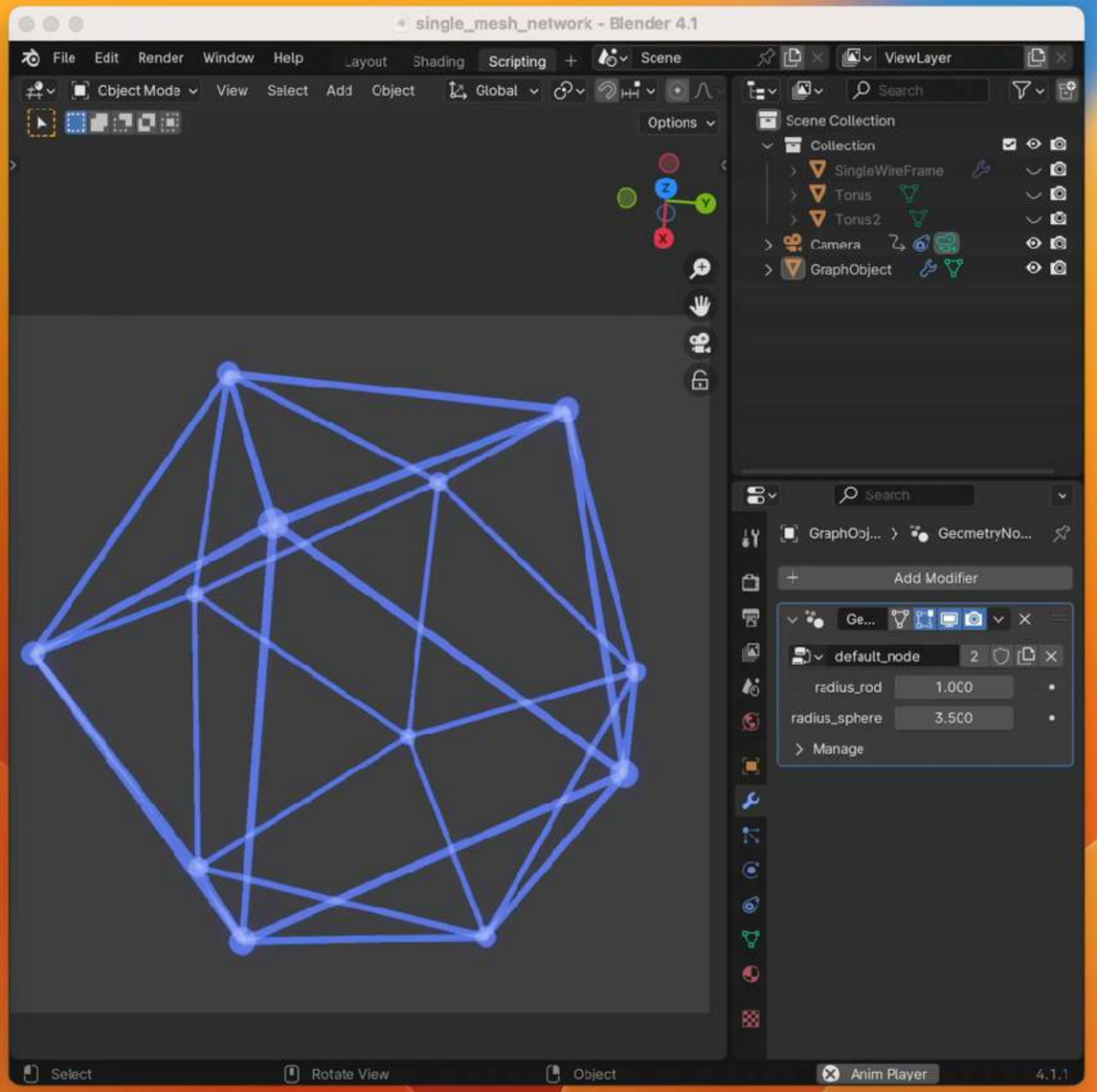
```
netw = bpy.data.objects.get('GraphObject')
netw_modif = netw.modifiers["GeometryNodes"]
netw_modif["Socket_2"]= 1.0
bpy.context.object.data.update()
```

```
netw_modif["Socket_2"]= 9.5
bpy.context.object.data.update()
```

```
material2 = bpy.data.materials.get("Material2")
for node in netw_modif.node_group.nodes:
    if node.type == 'SET_MATERIAL':
        node.inputs['Material'].default_value = material2
```

```
netw_modif["Socket_3"]= 22.0
bpy.context.object.data.update()
```

```
material1 = bpy.data.materials.get("Material1")
for node in netw_modif.node_group.nodes:
    if node.type == 'SET_MATERIAL':
        node.inputs['Material'].default_value = material1


netw_modif["Socket_2"]= 2.0
netw_modif["Socket_3"]= 3.5
bpy.context.object.data.update()
```



n3d_color_edge_to_face_brady.ipynb

blender

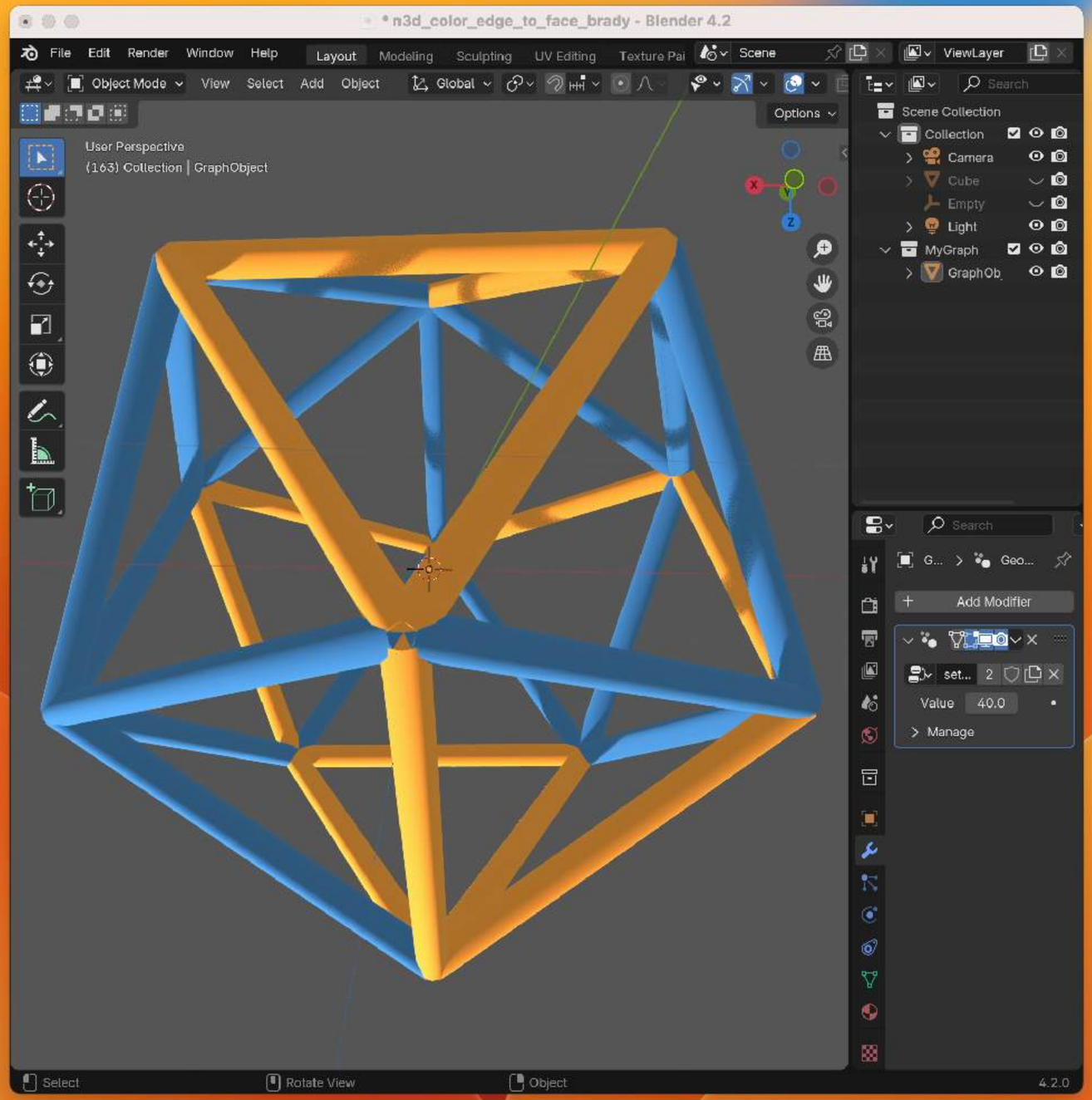
```
color = (1.0, 1.0, 1.0)  # WHITE

for edge in obj.data.edges:
    v1, v2 = edge.vertices
    set_edge_color(obj, (v1, v2), color)
```

```
color = (1.0, 0.0, 0.0)  # RED

set_edge_color(obj, (0, 1), color)
```

```
colors = [
    (1.0, 0.588, 0.149),  # ORANGE
    (0.176, 0.431, 0.686) # BLUE
]

for edge in obj.data.edges:
    v1, v2 = edge.vertices
    color = random.choice(colors)
    set_edge_color(obj, (v1, v2), color)
```

```
color = (1.0, 1.0, 1.0)  # WHITE

for edge in obj.data.edges:
    v1, v2 = edge.vertices
    set_edge_color(obj, (v1, v2), color)
```

```
my_edges = [(0, 1), (0, 5), (0, 7), (0, 8), (0, 11)]
color = (0.0, 0.0, 1.0) # BLUE

for edge in my_edges:
    set_edge_color(obj, edge, (0.0, 0.0, 1.0))
```



Code Notebook

[1]:

```
import bpy

bpy.ops.wm.open_mainfile(filepath='my_donut.blend')
```

```
Read blend: "/Users/jan-hendrik/projects/ipyblender-gui/my_donut.blend"
```

[1]:

```
{'FINISHED'}
```

[2]:

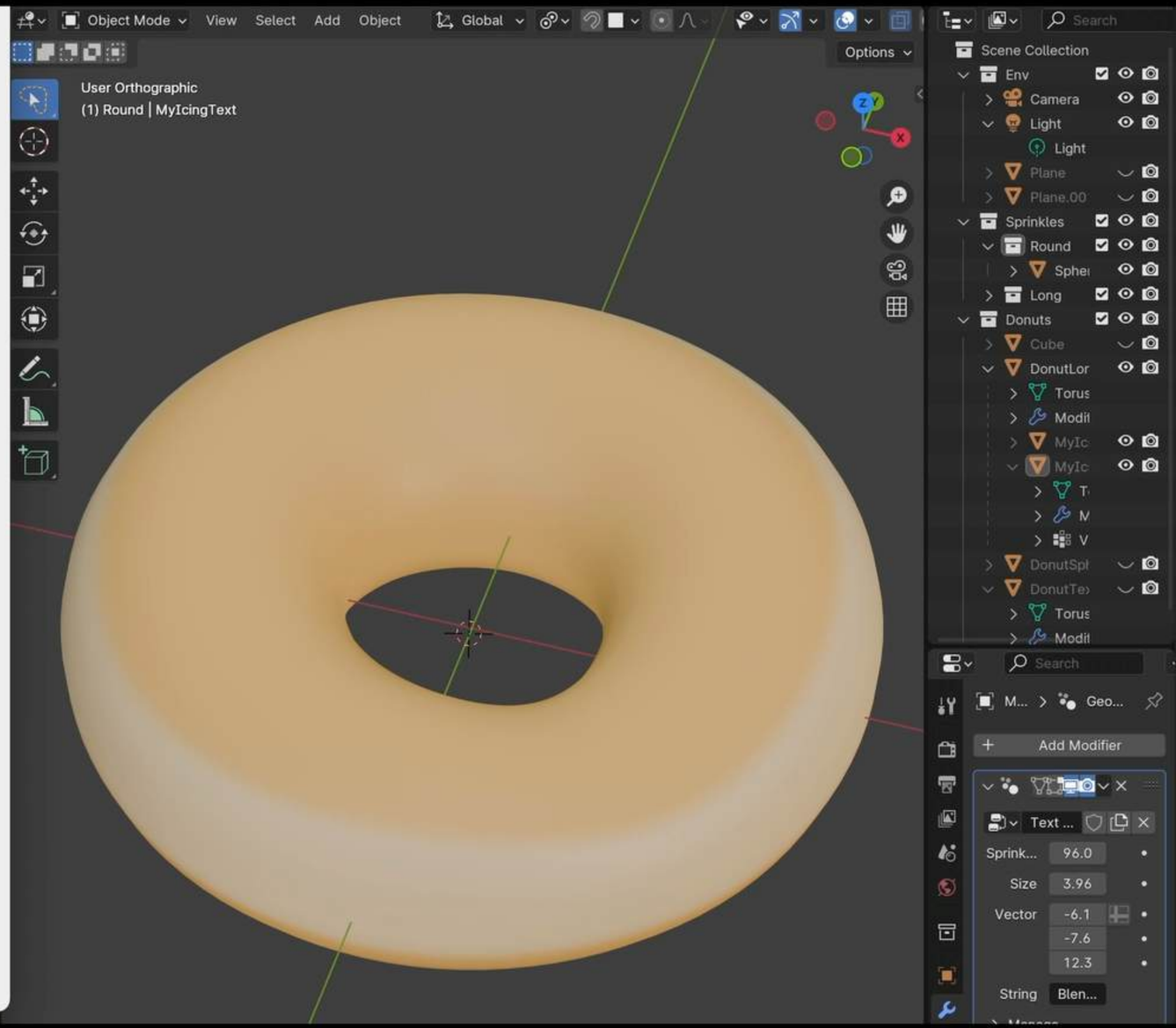
```
def show(obj):
    bpy.data.objects.get(obj).hide_viewport = False
def hide(obj):
    bpy.data.objects.get(obj).hide_viewport = True
show("DonutLong")
```

[ ]:

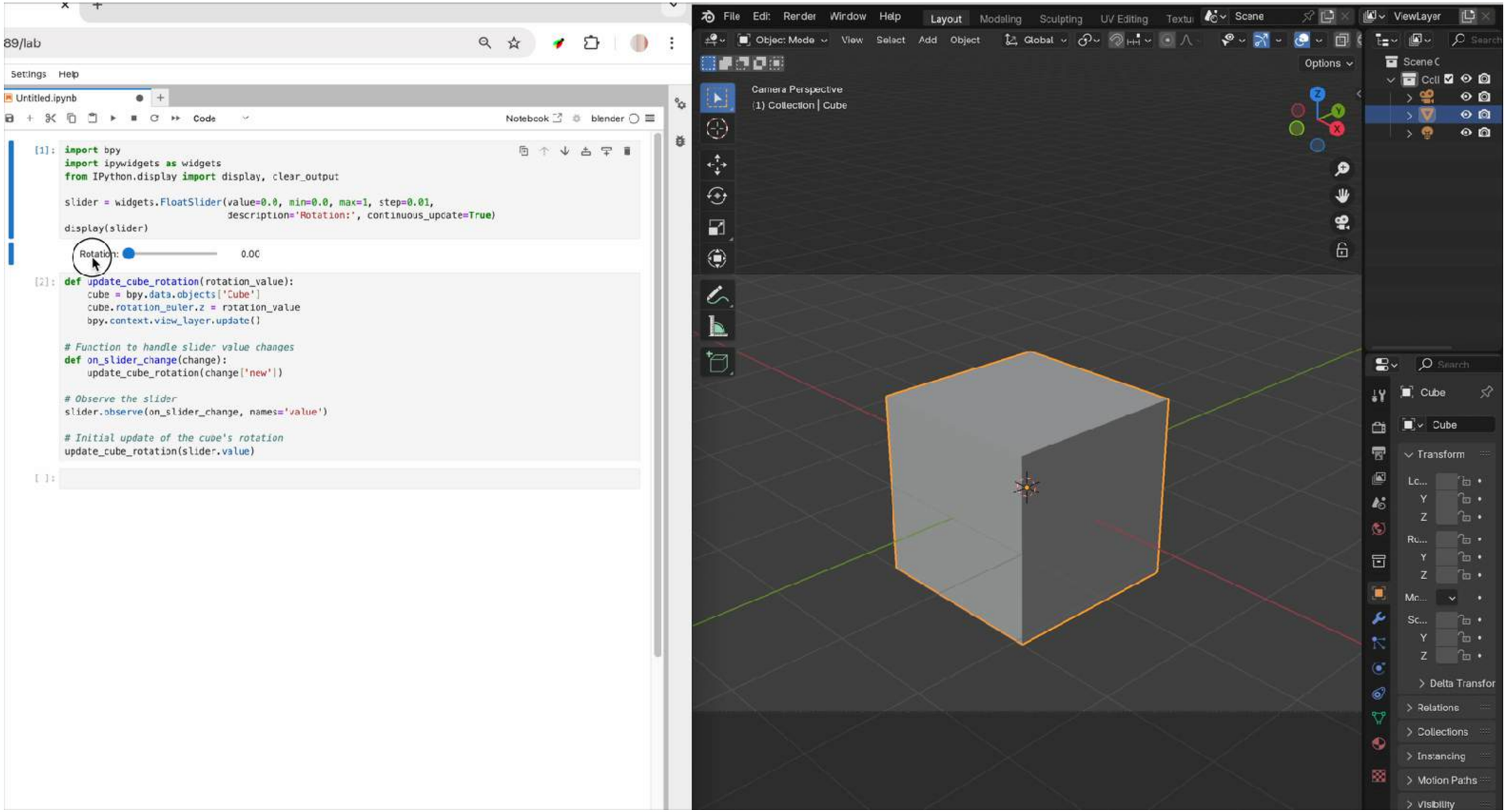
```
show("MyIcing")
```

[ ]:

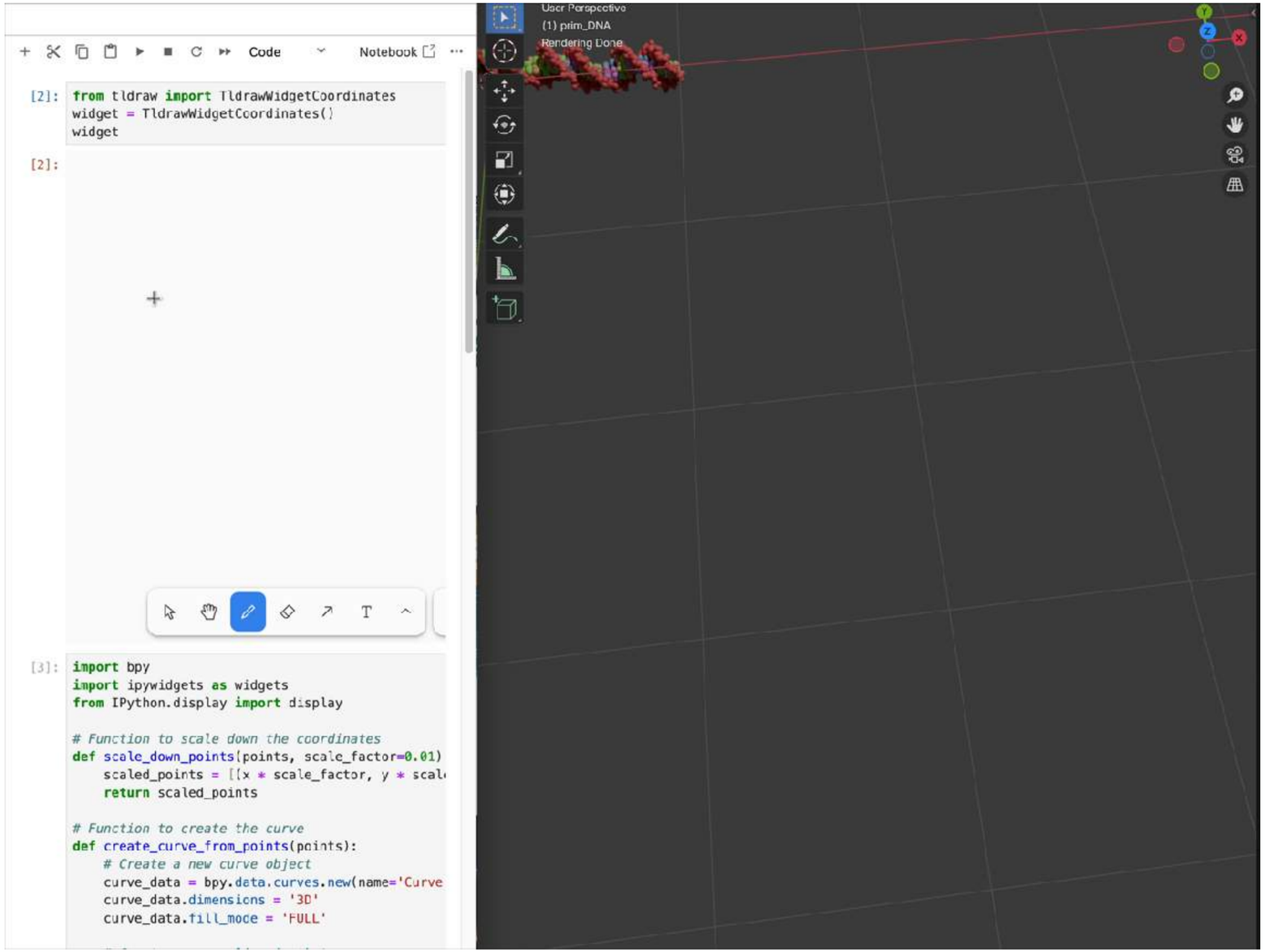
mple blender | Idle Mode: Command Ln 1, Col 16



2. Blender & JavaScript



2.1. interactive Whiteboard



https://github.com/kolibril13/jupyter-tldraw

2.2. GPT Vision API


```
from tldraw import MakeReal
from api_key import api_key

custom_prompt =  f"""
Use the bpy API to generate a blender scene.
In the providied image, you will see how the scene should look like.
Use 3D shape premitives to add the desired shapes and their colors.
Also add a camera and a light.

Reply ONLY with python code.
"""


m = MakeReal(width=650, height = 400, api_key = api_key,
             prompt = custom_prompt, run_next_cell = True)
m
```

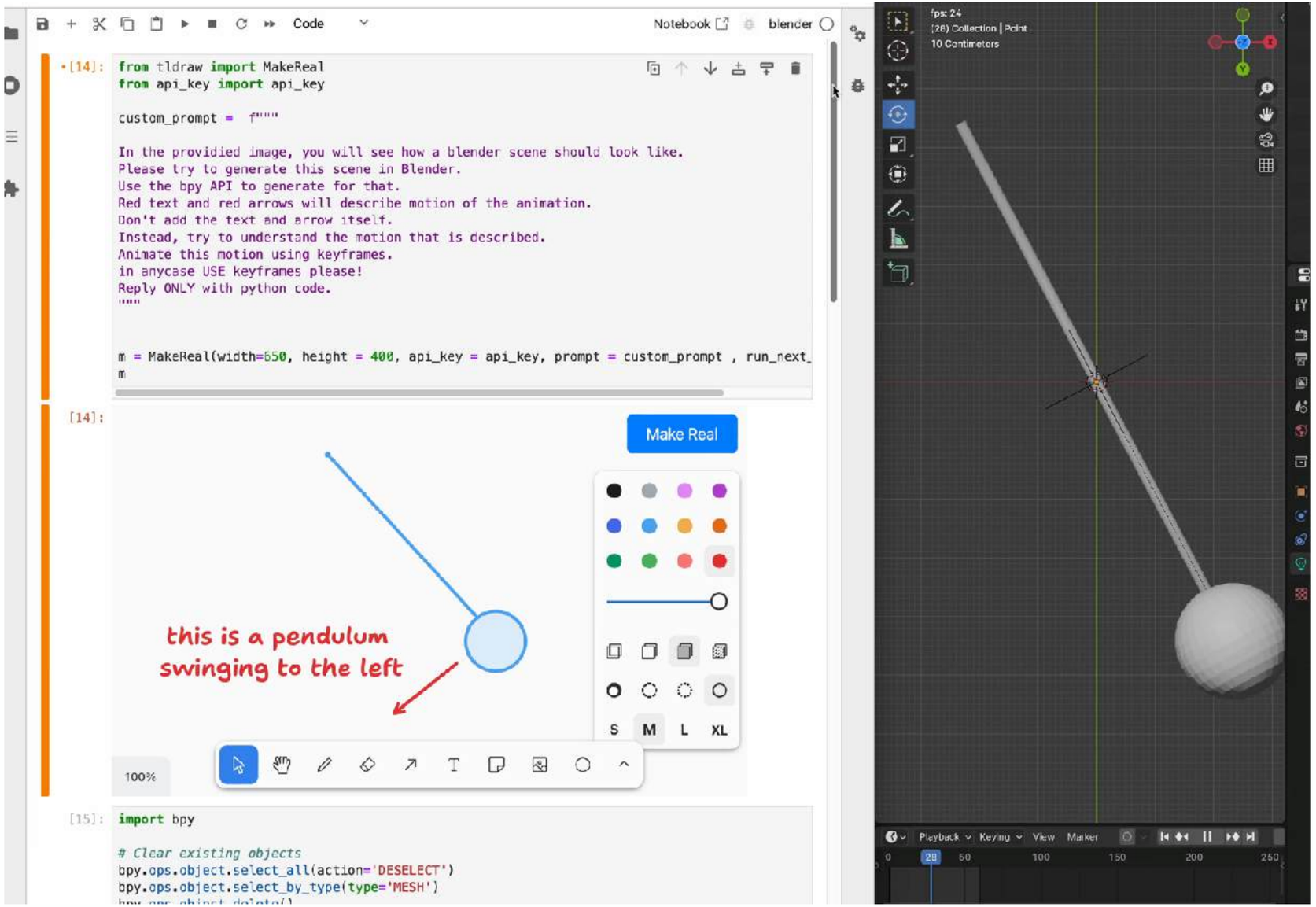
Make Real

S M L XL

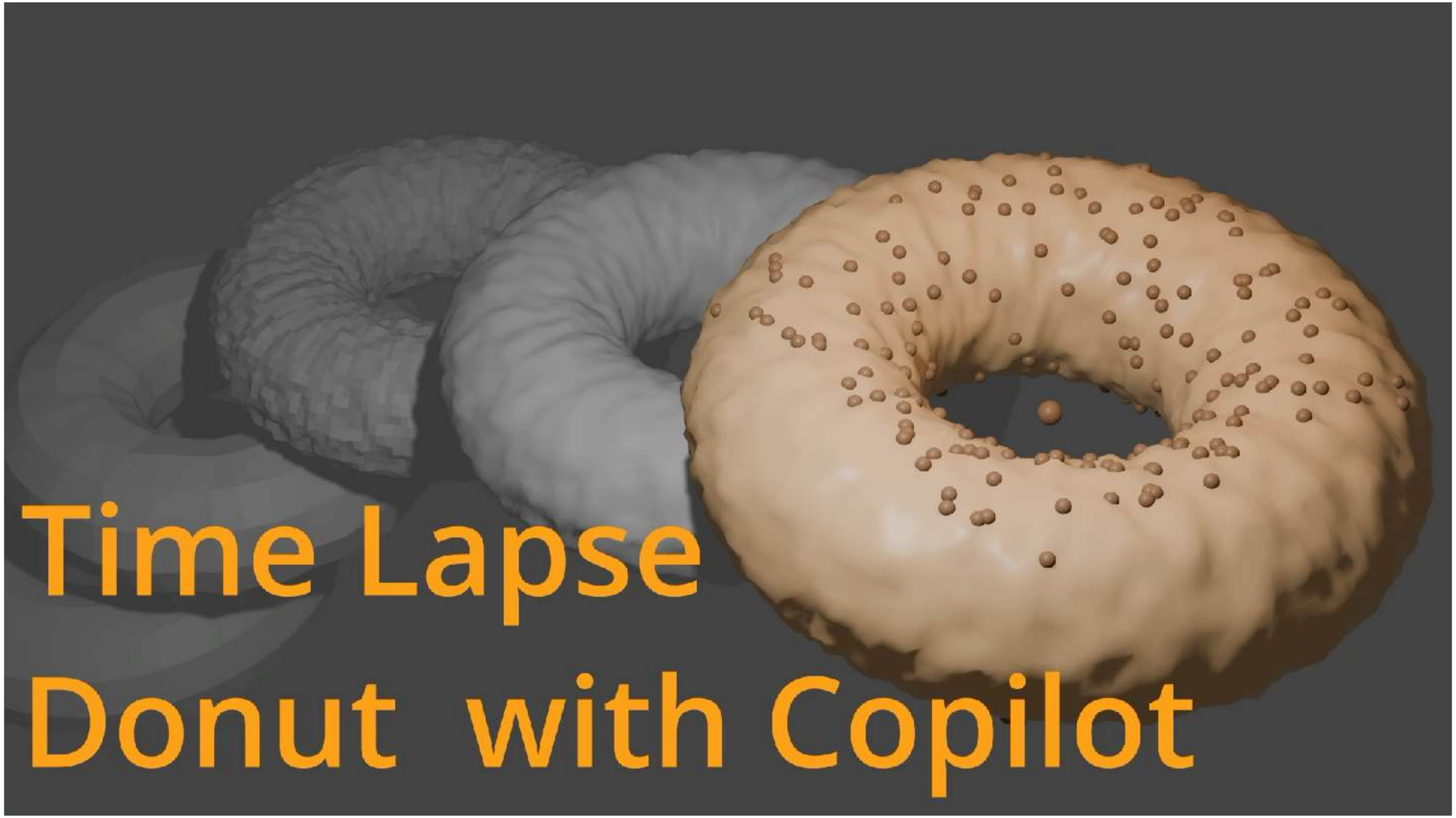
100%

User Perspective
(1) Collection | Cube

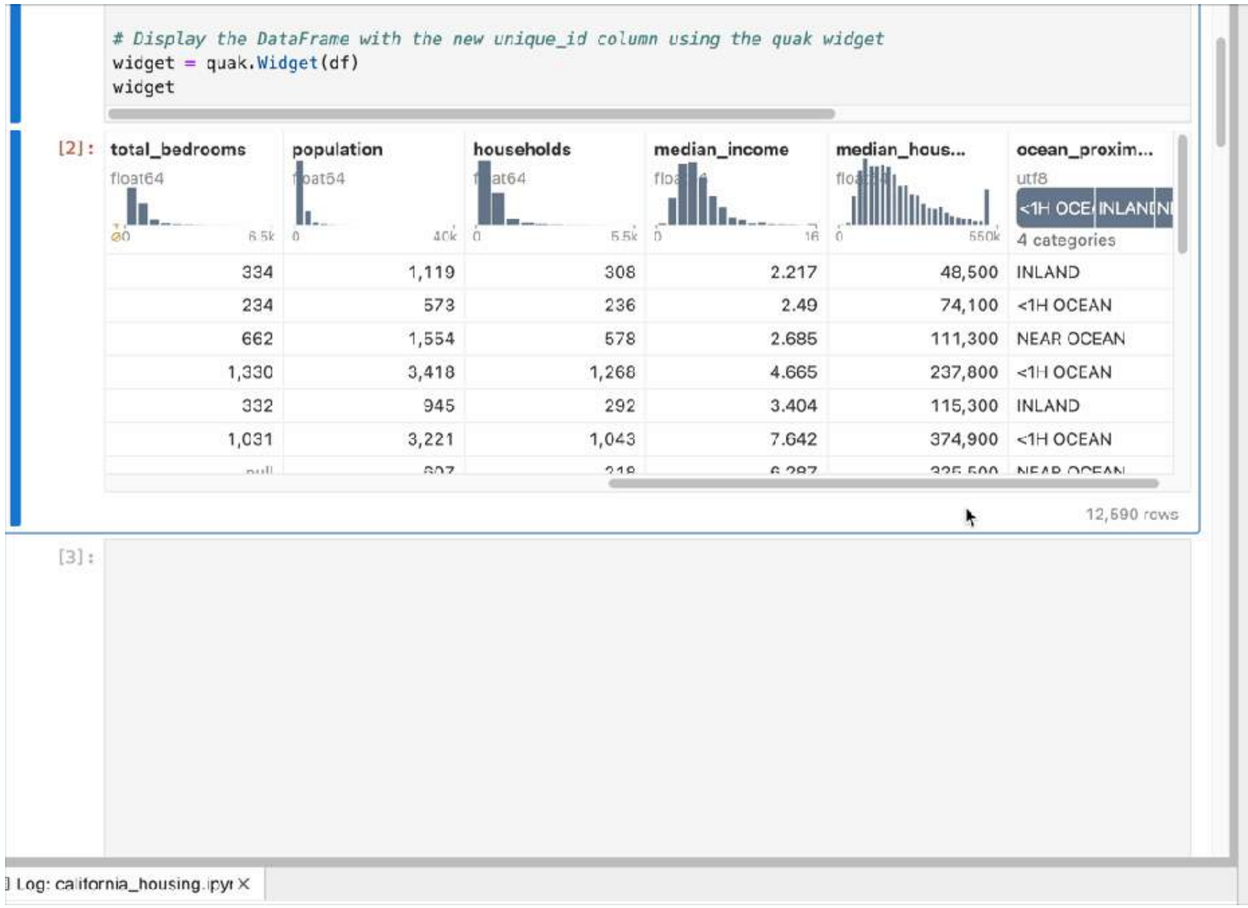
2.2. GPT Vision API



GitHub Copilot



2.3. Interactive Tables



```
# Display the DataFrame with the new unique_id column using the quak widget
widget = quak.Widget(df)
widget
```

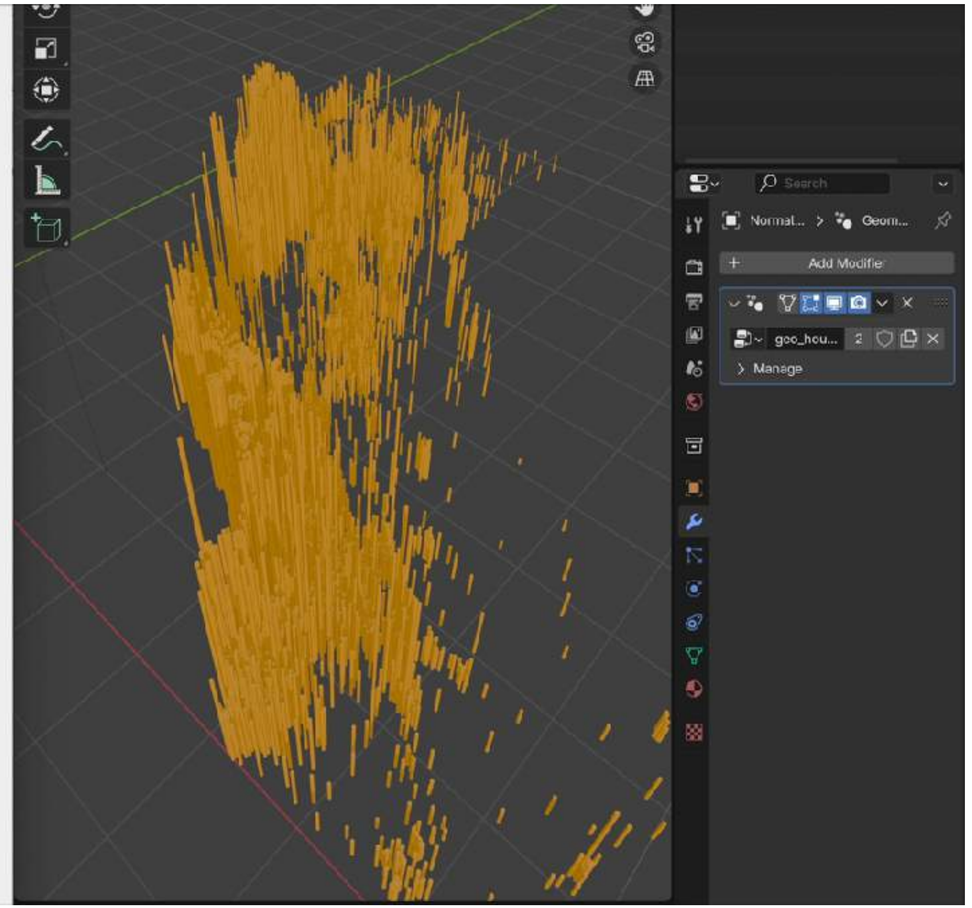
[2]:

total_bedrooms	population	households	median_income	median_hous...	ocean_proxim...
float64	float64	float64	float64	float64	utf8
					<1H OCE INLAN N
					4 categories
334	1,119	308	2.217	48,500	INLAND
234	573	236	2.49	74,100	<1H OCEAN
662	1,554	578	2.685	111,300	NEAR OCEAN
1,330	3,418	1,268	4.665	237,800	<1H OCEAN
332	945	292	3.404	115,300	INLAND
1,031	3,221	1,043	7.642	374,900	<1H OCEAN

12,590 rows

[3]:

] Log: california_housing.ipyr ×



Trevor Manz
manzt · he,him

https://github.com/manzt/quak

3. Documentation

```
In [6]: fresh_scene()
        bpy.ops.mesh.primitive_cylinder_add(radius=1, depth=2, location=(0, 0, 0))
        render_result()
```

```
In [7]: fresh_scene()
        bpy.ops.mesh.primitive_torus_add(major_radius=2, minor_radius=0.3, location=(0.0, 0))
        render_result()
```

```
In [8]: fresh_scene()
        bpy.ops.mesh.primitive_ico_sphere_add(radius=1, location=(0, 0, 0))
        render_result()
```

```
In [9]: fresh_scene()
        bpy.ops.object.text_add(location=(-4, 0, 0))
        bpy.context.object.data.body = "Hello 😎"
        bpy.context.object.scale = (2, 2, 2)
        render_result()
```

Load file

```
In [31]: fresh_scene()
         bpy.ops.wm.open_mainfile(filepath='donut.blend')
Out[31]: {'FINISHED'}
```

Choose render engine

```
In [32]: bpy.context.scene.render.engine = 'BLENDER_WORKBENCH'
         bpy.context.scene.render.resolution_x = 300
         bpy.context.scene.render.resolution_y = 200

         render_result()
```

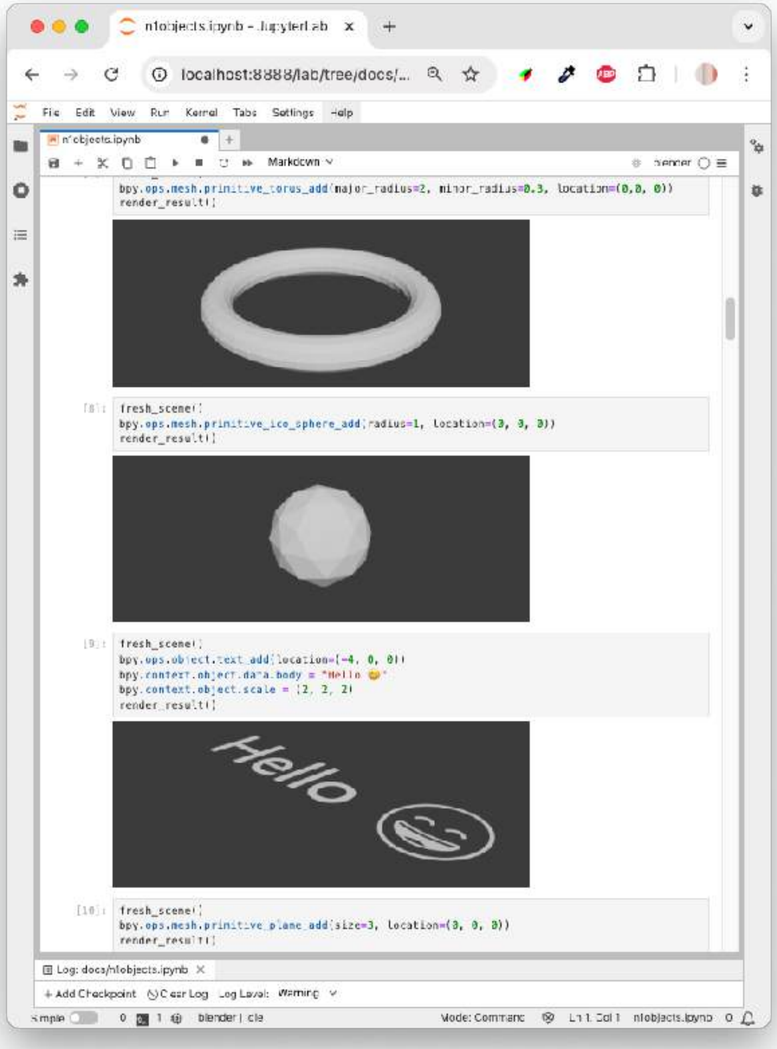
```
In [33]: bpy.context.scene.render.engine = 'BLENDER_EEVEE_NEXT'
         render_result()
```

```
In [34]: bpy.context.scene.render.engine = 'CYCLES'
         bpy.context.scene.cycles.samples = 16
         render_result()
```

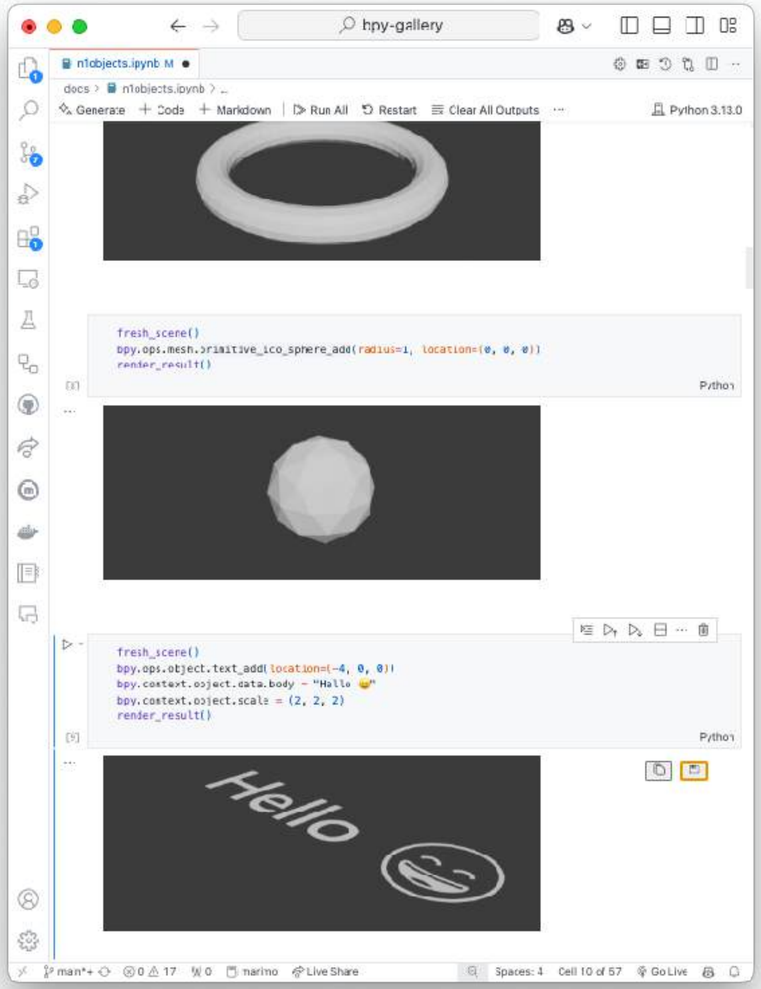
https://kolibril13.github.io/bpy-gallery/n1objects/

Notebook platforms

Jupyter Lab

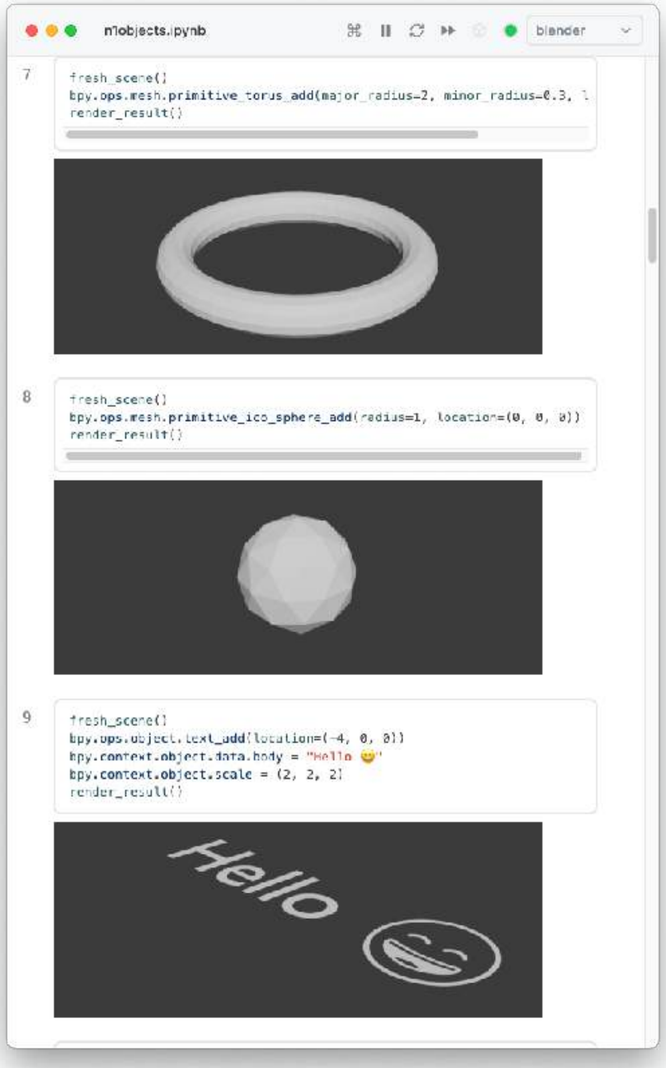


VS Code Notebook



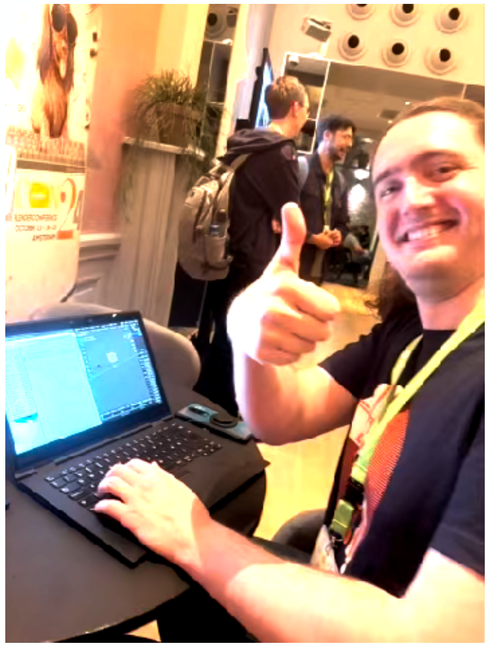
Notebook platforms

Satyrn



Marimo

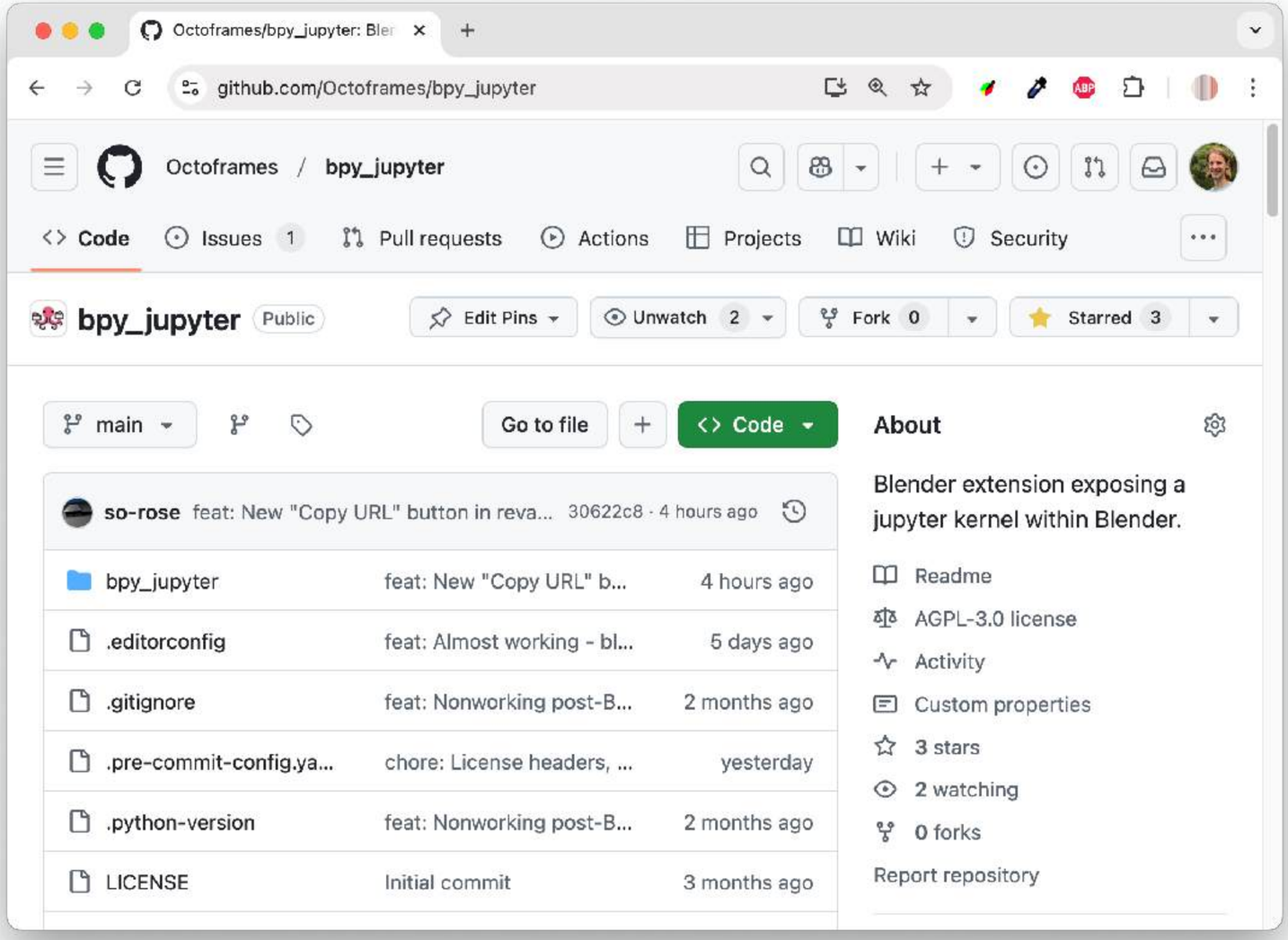
Installation: bpy_jupyter Blender Extension



Sofus Albert Høgsbro Rose
so-rose

Jan-Hendrik Müller
kolibril13

https://github.com/Octoframes/bpy_jupyter

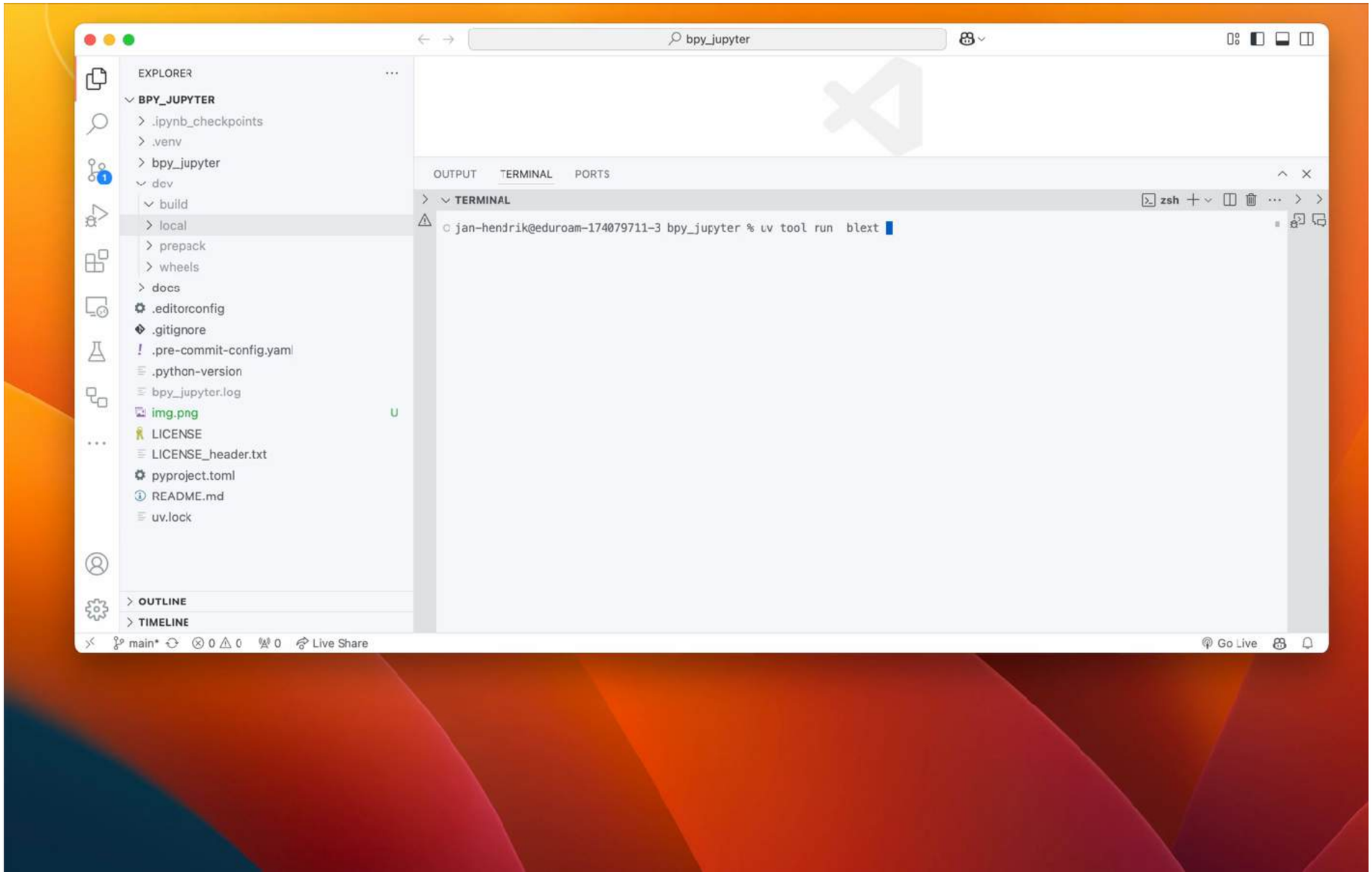


Installation

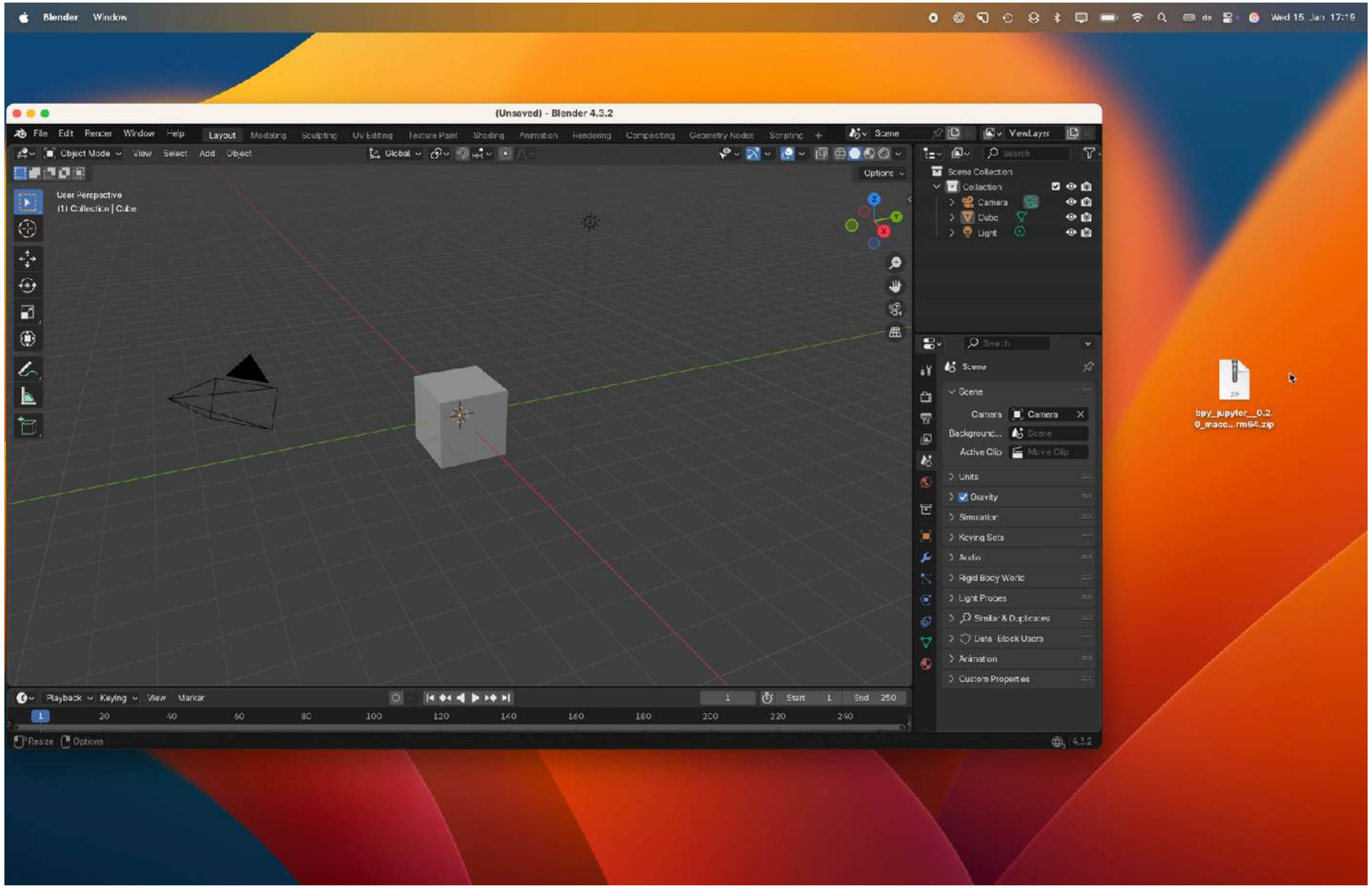
Soon on:
https://extensions.blender.org/

Currently:
1. Clone https://github.com/Octoframes/bpy_jupyter
2. Install via **uv tool run blext dev**

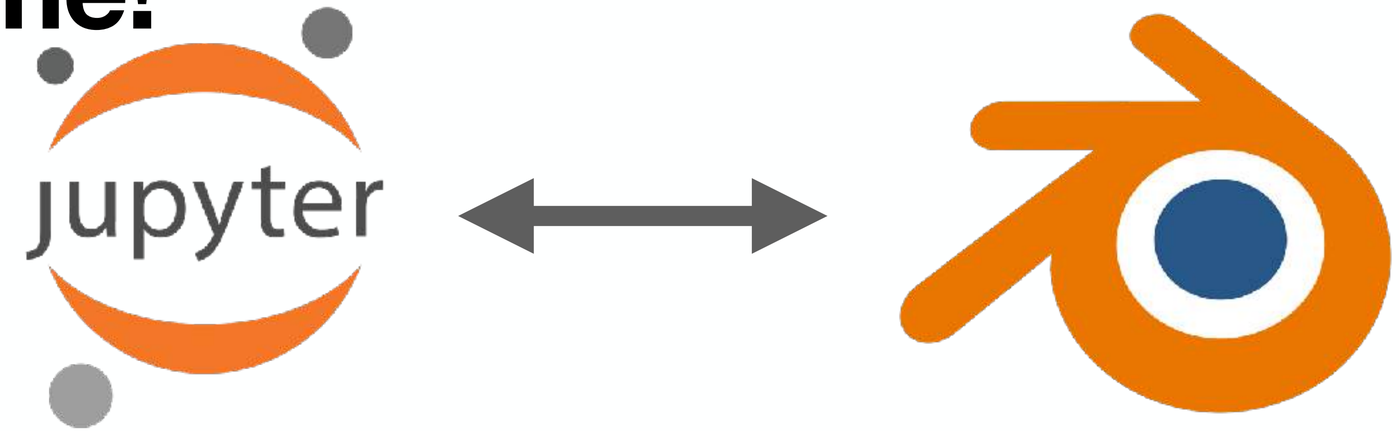
Build extension



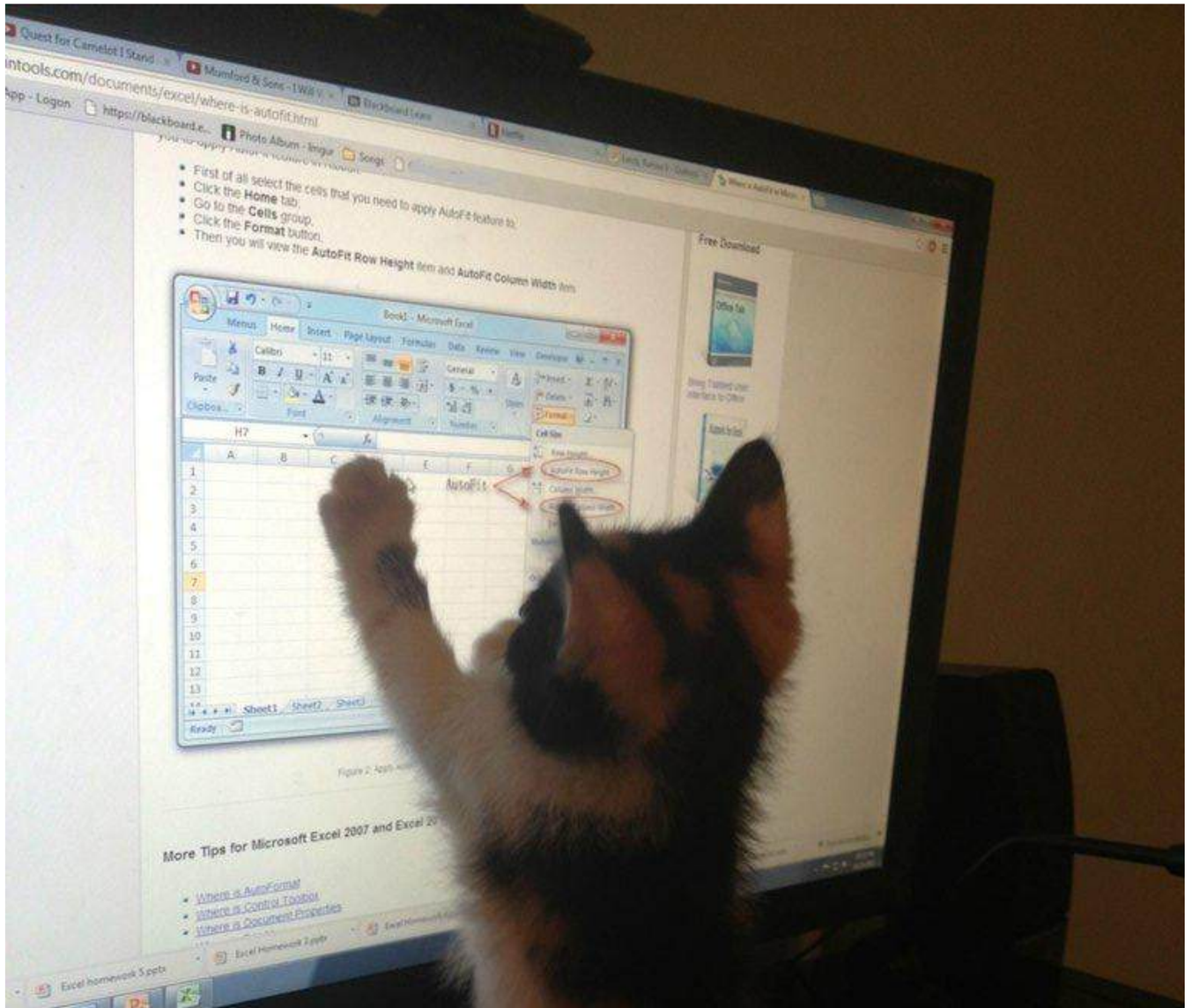
Install+Connect to JupyterLab



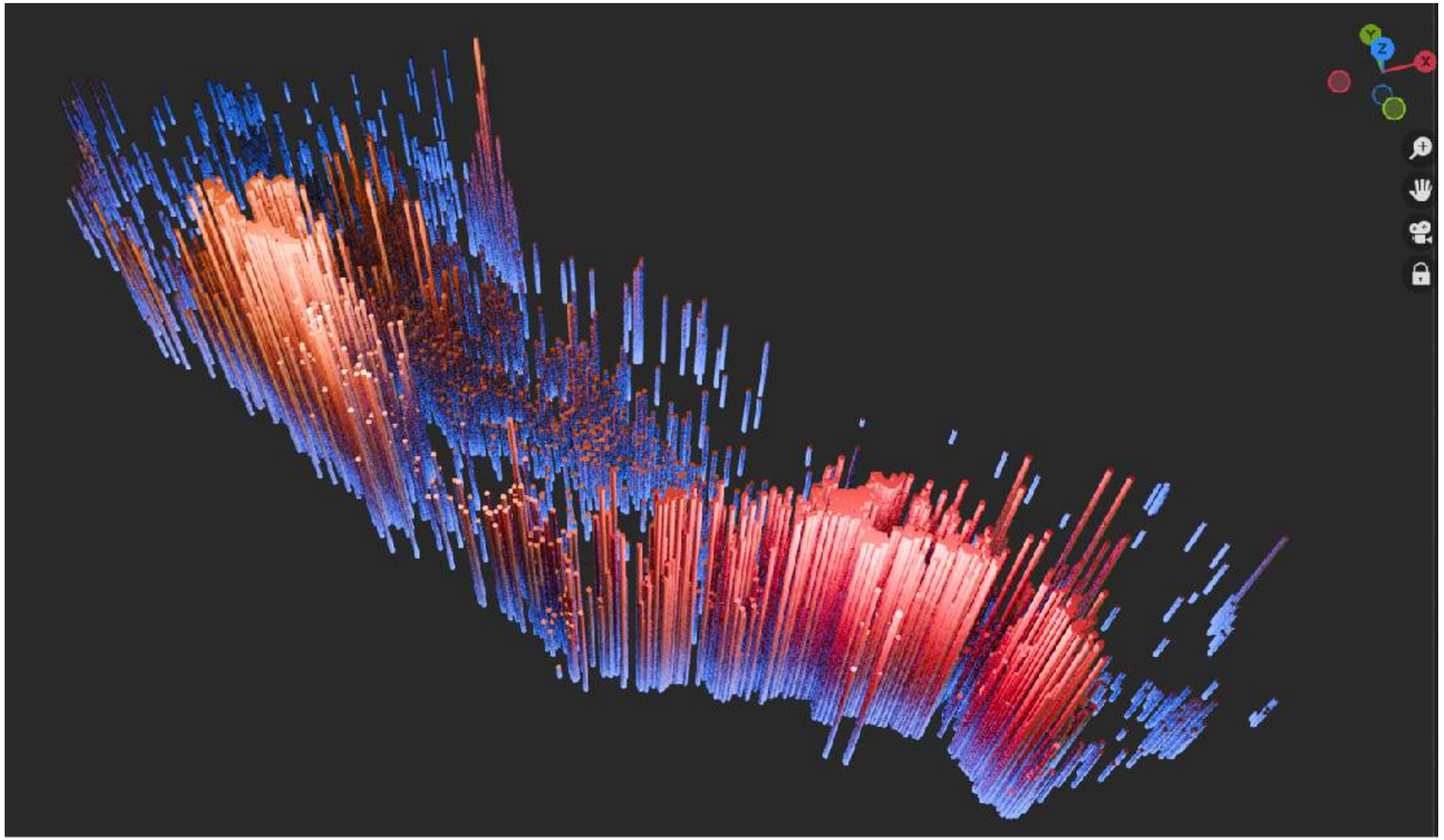
Demo time!



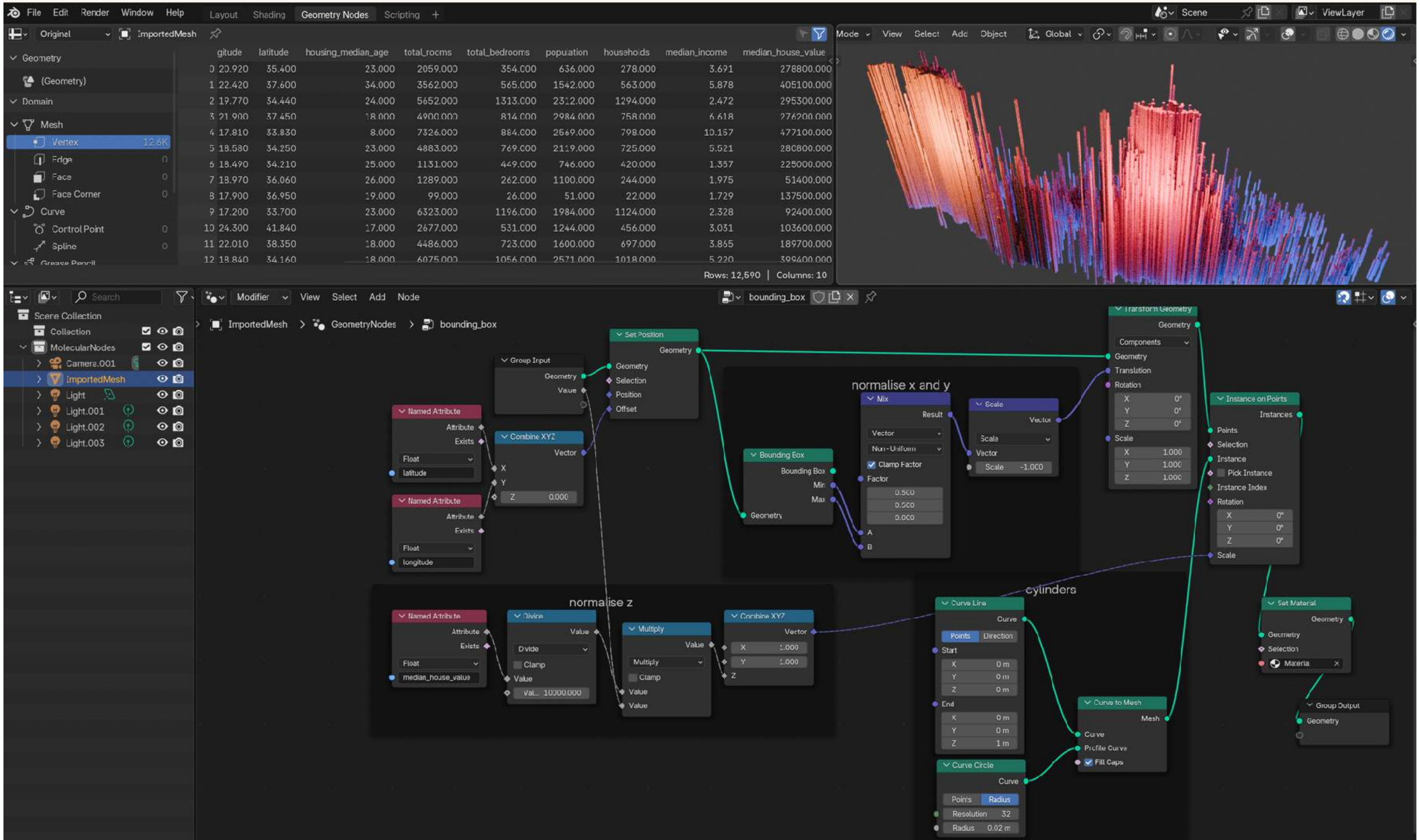
What about spreadsheets?

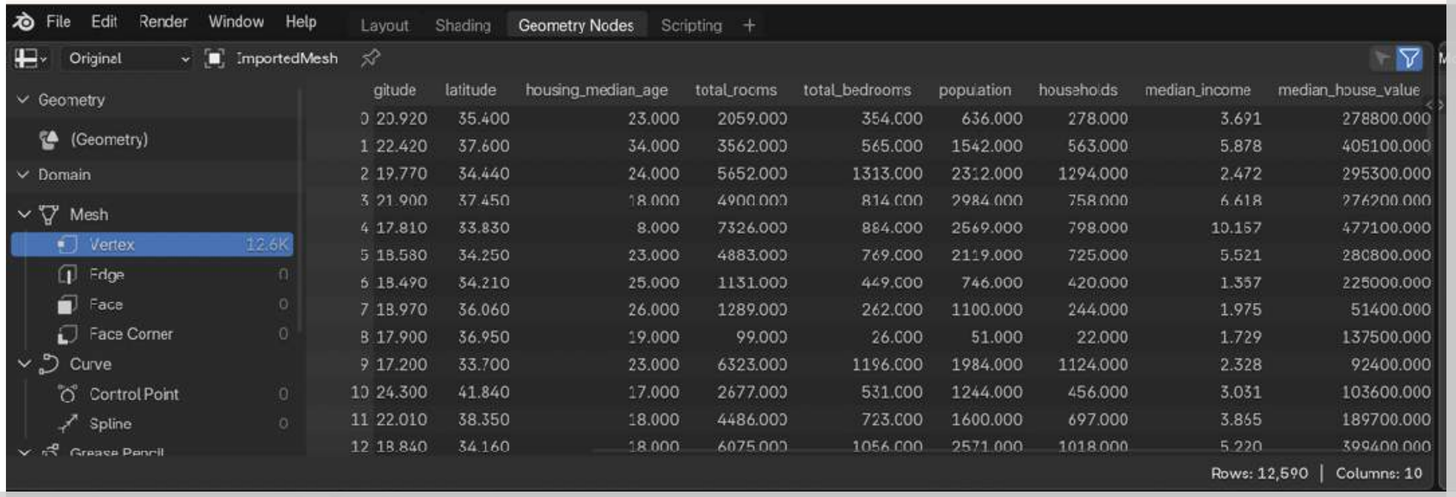


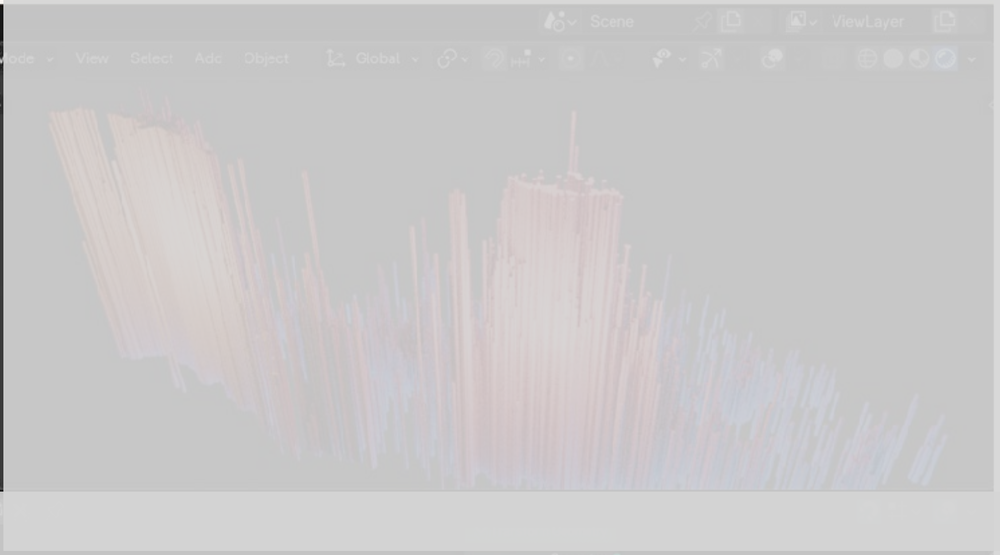
Data Processing Pipeline in Blender

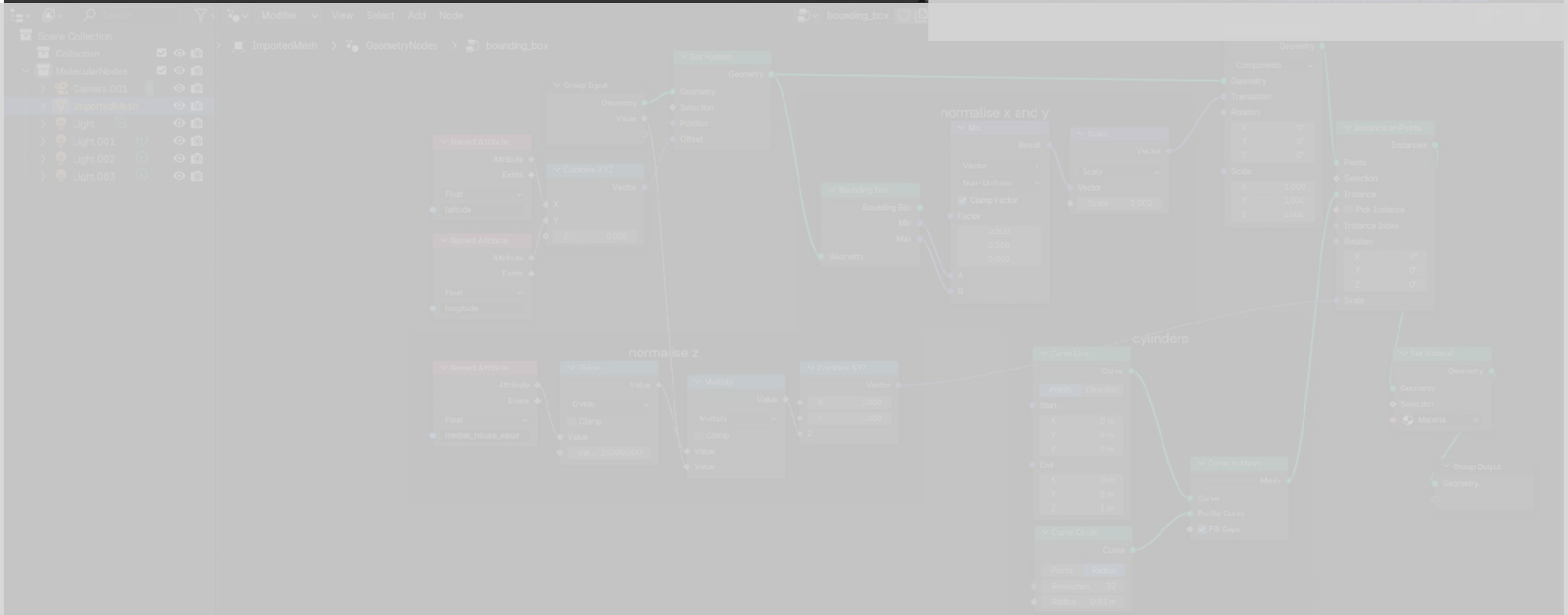


California dataset: Latitude, longitude and house prices (12k datapoints)


File Edit Render Window Help
Layout Shading Geometry Nodes Scripting
Scene ViewLayer
Original ImportedMesh
Geometry
(Geometry)
Domain
Mesh
Vertex 12.6K
Edge 0
Face 0
Face Corner 0
Curve
Control Point 0
Spline 0
gitude latitude housing_median_age total_rooms total_bedrooms population households median_income median_house_value
0 20.920 35.400 23.000 2059.000 354.000 636.000 278.000 3.691 278800.000
1 22.420 37.600 34.000 3562.000 565.000 1542.000 563.000 5.878 405100.000
2 19.770 34.440 24.000 5652.000 1313.000 2312.000 1294.000 2.472 295300.000
3 21.900 37.450 18.000 4900.000 814.000 2984.000 758.000 6.618 276200.000
4 17.810 33.830 8.000 7326.000 884.000 2569.000 798.000 10.157 477100.000
5 18.580 34.250 23.000 4883.000 769.000 2119.000 725.000 5.521 280800.000
6 18.490 34.210 25.000 1131.000 449.000 746.000 420.000 1.357 225000.000
7 18.970 36.060 26.000 1289.000 262.000 1100.000 244.000 1.975 51400.000
8 17.900 36.950 19.000 99.000 26.000 51.000 22.000 1.729 137500.000
9 17.200 33.700 23.000 6323.000 1196.000 1984.000 1124.000 2.328 92400.000
10 24.300 41.840 17.000 2677.000 531.000 1244.000 456.000 3.031 103600.000
11 22.010 38.350 18.000 4486.000 723.000 1600.000 697.000 3.855 189700.000
12 18.840 34.160 18.000 6075.000 1056.000 2571.000 1018.000 5.220 399400.000
Rows: 12,590 | Columns: 10
Mode View Select Add Object Global
Search
Scene Collection
Collection
MolecularNodes
Camera.001
ImportedMesh
Light
Light.001
Light.002
Light.003
Modifier View Select Add Node
bounding_box
ImportedMesh GeometryNodes bounding_box
Group Input
Geometry
Value
Set Position
Geometry
Selection
Position
Offset
Named Attribute
Attribute
Exists
Float
latitude
Named Attribute
Attribute
Exists
Float
longitude
Combine XYZ
Vector
X
Y
Z 0.000
normalise x and y
Bounding Box
Bounding Box
Min
Max
Geometry
Mix
Result
Vector
Non-Uniform
Clamp Factor
Factor
0.500
0.500
0.000
A
B
Scale
Vector
Scale
Vector
Scale -1.000
Transform Geometry
Geometry
Components
Geometry
Translation
Rotation
X 0°
Y 0°
Z 0°
Scale
X 1.000
Y 1.000
Z 1.000
Instance on Points
Instances
Points
Selection
Instance
Pick Instance
Instance Index
Rotation
X 0°
Y 0°
Z 0°
Scale
normalise z
Named Attribute
Attribute
Exists
Float
median_house_value
Divide
Value
Divide
Clamp
Value
Val... 10000.000
Multiply
Value
Multiply
Clamp
Value
Value
Combine XYZ
Vector
X 1.000
Y 1.000
Z
cylinders
Curve Line
Curve
Points Direction
Start
X 0 m
Y 0 m
Z 0 m
End
X 0 m
Y 0 m
Z 1 m
Curve Circle
Curve
Points Radius
Resolution 32
Radius 0.02 m
Curve to Mesh
Mesh
Curve
Profile Curve
Fill Caps
Set Material
Geometry
Geometry
Selection
Materia
Group Output
Geometry


File Edit Render Window Help
Layout Shading Geometry Nodes Scripting
Original
ImportedMesh
Geometry
(Geometry)
Domain
Mesh
Vertex 12.6K
Edge 0
Face 0
Face Corner 0
Curve
Control Point 0
Spline 0
Grease Pencil
gitude latitude housing_median_age total_rooms total_bedrooms population households median_income median_house_value
0 20.920 35.400 23.000 2059.000 354.000 636.000 278.000 3.691 278800.000
1 22.420 37.600 34.000 3562.000 565.000 1542.000 563.000 5.878 405100.000
2 19.770 34.440 24.000 5652.000 1313.000 2312.000 1294.000 2.472 295300.000
3 21.900 37.450 18.000 4900.000 814.000 2984.000 758.000 6.618 276200.000
4 17.810 33.830 8.000 7326.000 884.000 2569.000 798.000 10.157 477100.000
5 18.580 34.250 23.000 4883.000 769.000 2119.000 725.000 5.521 280800.000
6 18.490 34.210 25.000 1131.000 449.000 746.000 420.000 1.357 225000.000
7 18.970 36.060 26.000 1289.000 262.000 1100.000 244.000 1.975 51400.000
8 17.900 36.950 19.000 99.000 26.000 51.000 22.000 1.729 137500.000
9 17.200 33.700 23.000 6323.000 1196.000 1984.000 1124.000 2.328 92400.000
10 24.300 41.840 17.000 2677.000 531.000 1244.000 456.000 3.031 103600.000
11 22.010 38.350 18.000 4486.000 723.000 1600.000 697.000 3.855 189700.000
12 18.840 34.160 18.000 6075.000 1056.000 2571.000 1018.000 5.220 399400.000
Rows: 12,590 | Columns: 10




Modifier View Select Add Node
bounding_box
Scene Collection
Collection
MolecularNodes
Camera.001
ImportedMesh
Light
Light.001
Light.002
Light.003
ImportedMesh GeometryNodes bounding_box
normalise x and y
normalise z
cylinders

Data with Python

```
[1]: import numpy as np
     import databpy as db

     vertex = np.array([[0, 0, 0]])
     obj = db.create_bob(vertex, name="Mesh1")
     obj.name
```

```
[1]: 'Mesh1'
```

Brady Johnston
BradyAJohnston · he/him

https://github.com/bradyajohnston/databpy

README GPL-3.0 license

databpy

codecov 61% pypi v0.0.8 Run Tests passing ci-cd.yml passing

A set of data-oriented wrappers around the python API of Blender.

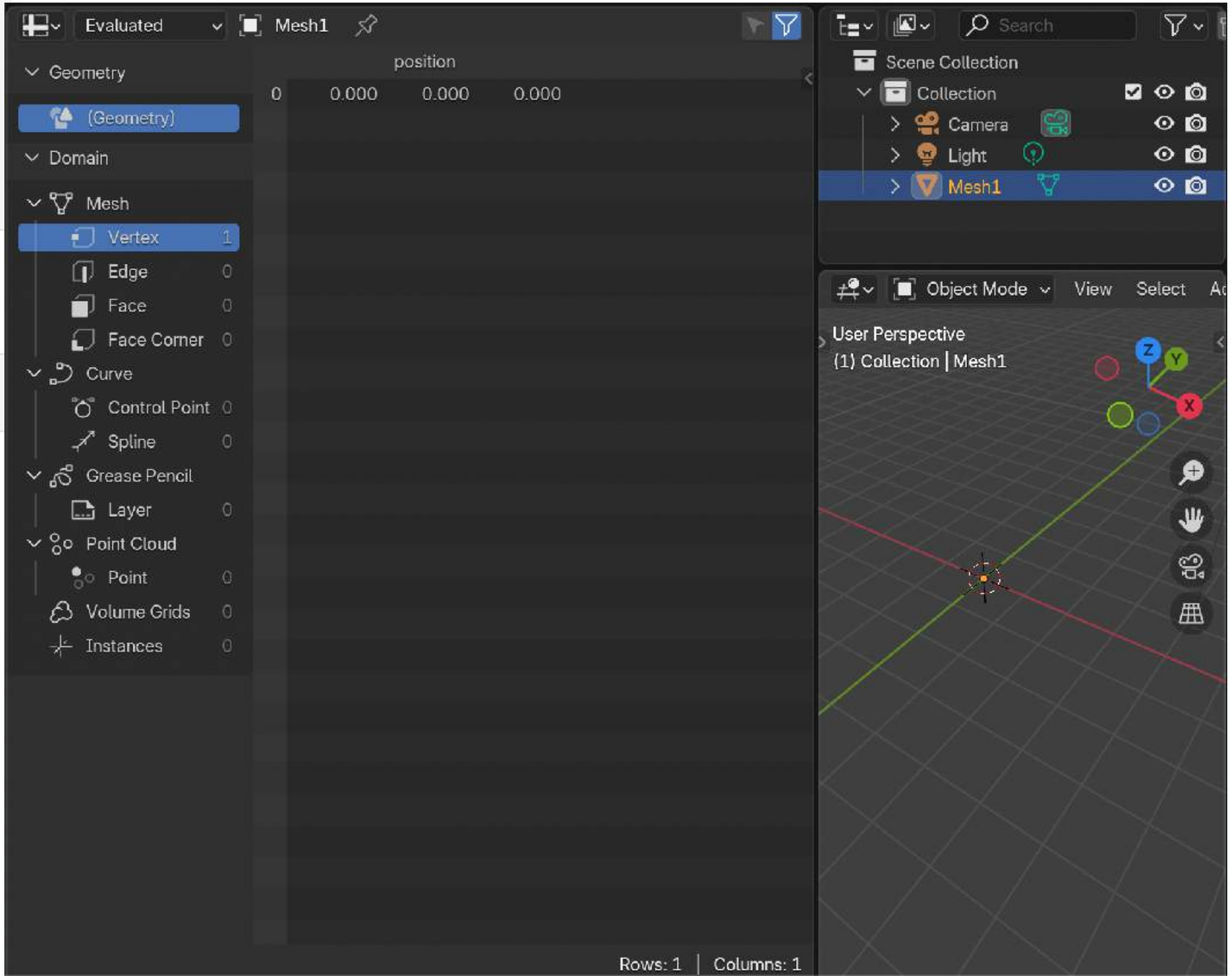
Creating Meshes

```
import numpy as np
import databpy as db

vertex = np.array([[0, 0, 0]])
obj = db.create_bob(vertex, name="Mesh1")
obj.name
```

```
'Mesh1'
```

```
obj.position
```

```
array([[0., 0., 0.]])
```



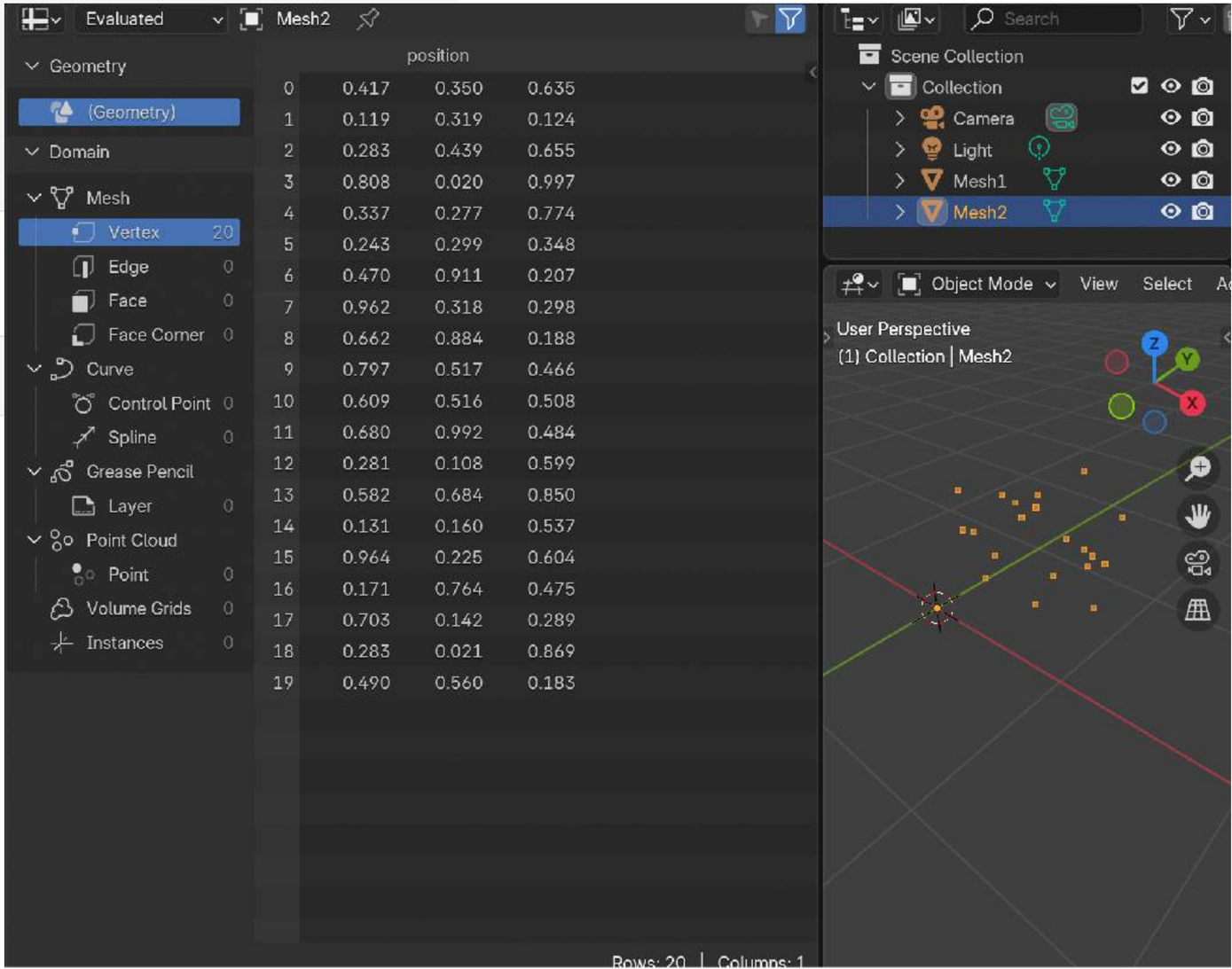
Creating Meshes

```
import numpy as np
import databpy as db

coords = np.random.rand(20, 3)
obj = db.create_bob(coords, name="Mesh2")
obj.name
```

```
'Mesh2'
```

```
obj.position
```

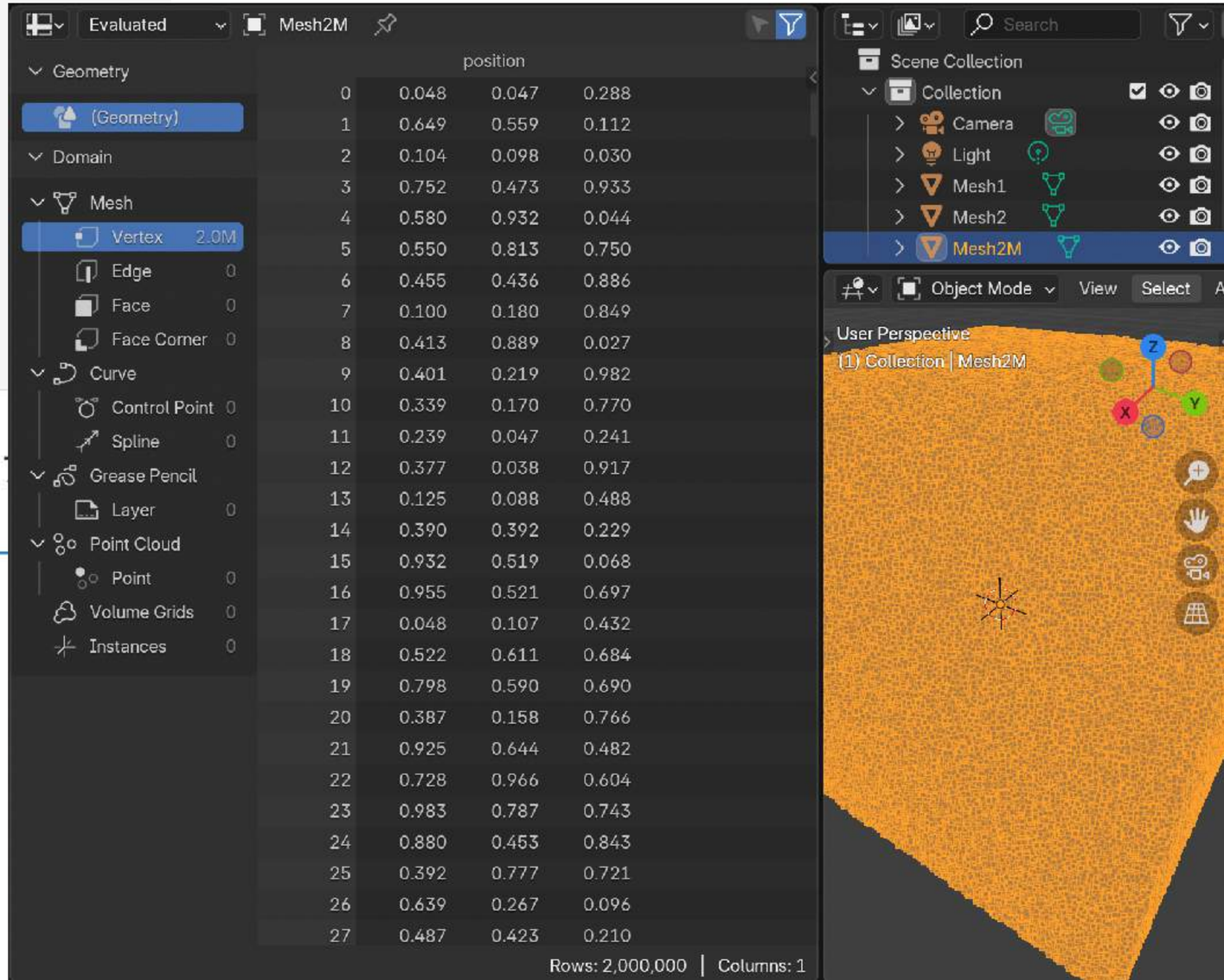
```
array([[0.41664836, 0.35038018, 0.63478756],
       [0.11875096, 0.31862018, 0.12424635],
       [0.28274721, 0.43855155, 0.6550467 ],
       [0.80807185, 0.01969962, 0.99721068],
       [0.33697239, 0.27726692, 0.7743066 ],
       [0.2434269 , 0.29881901, 0.34785211],
       [0.47011954, 0.91141248, 0.20741948],
       [0.96200317, 0.31830707, 0.29814231],
       [0.66150445, 0.88354313, 0.18802765],
       [0.79731137, 0.51676905, 0.46561515],
       [0.60903811, 0.51614606, 0.50760353],
```



Creating Meshes

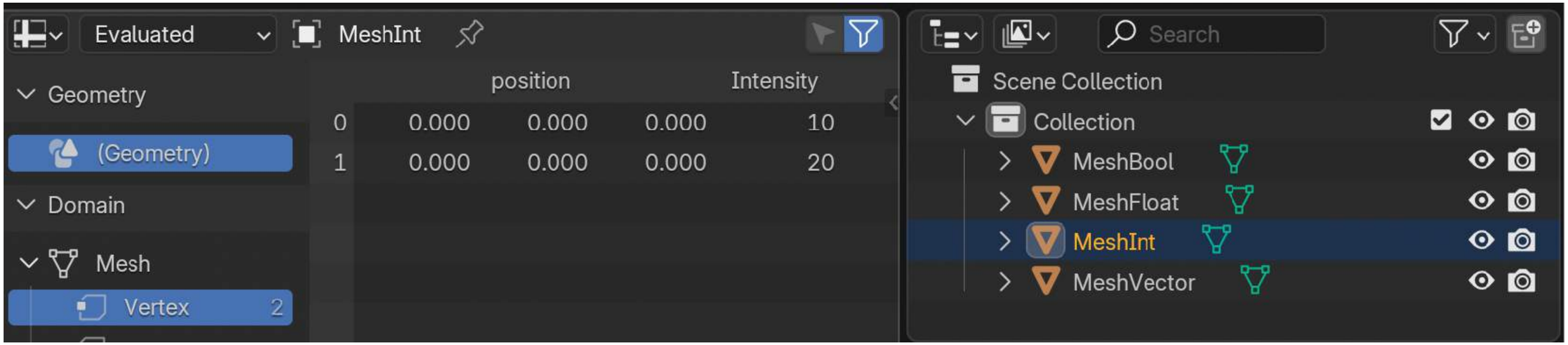
```
%%time
import numpy as np
import databpy as db

# Generate 2 million random 3D points
coords = np.random.rand(2_000_000, 3)

# Create a mesh object using databpy
obj = db.create_bob(coords, name="Mesh2M")
print(obj.name)
```

```
Mesh2M
CPU times: user 685 ms, sys: 36.3 ms, total: 
Wall time: 717 ms
```



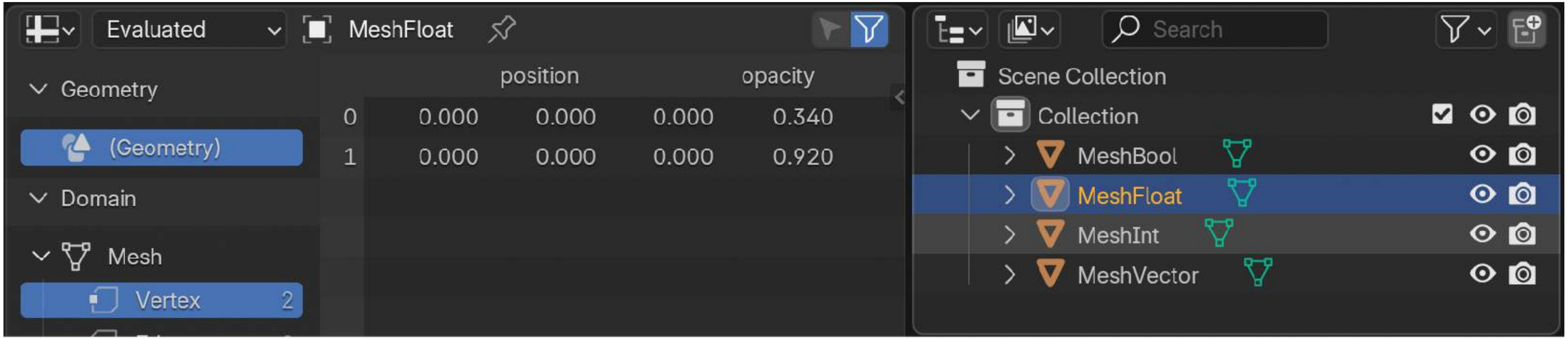
Attributes to each point

```
import numpy as np
import polars as pl
from csv_importer.parsers import polars_df_to_bob

df = pl.DataFrame(
    {
        "Intensity": [10, 20],
    }
)
bob = polars_df_to_bob(df, name="MeshInt")
```



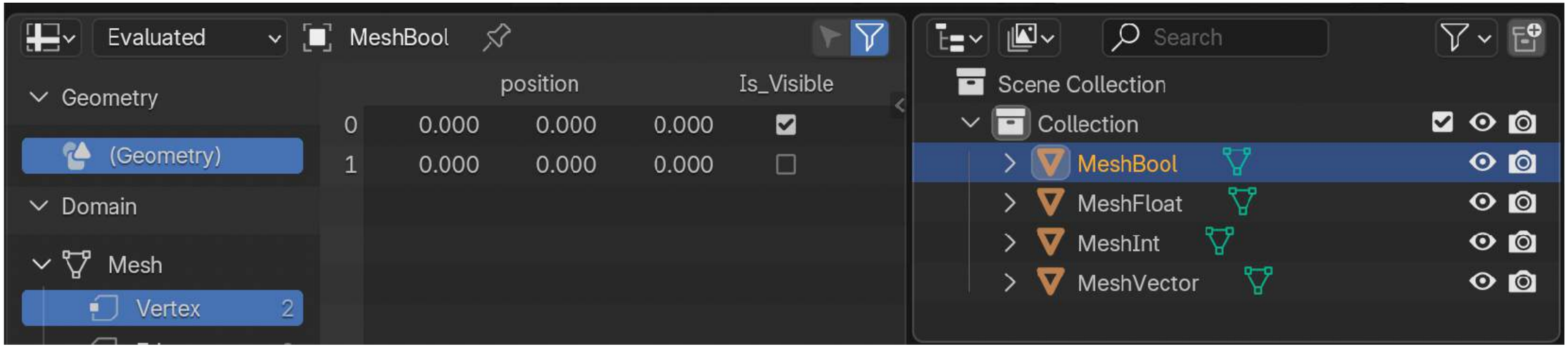
Attributes to each point

```python
import numpy as np
import polars as pl
from csv_importer.parsers import polars_df_to_bob

df = pl.DataFrame(
    {
        "opacity": [0.34, 0.92],
    }
)
bob = polars_df_to_bob(df, name="MeshFloat")
```



Attributes to each point

```python
import numpy as np
import polars as pl
from csv_importer.parsers import polars_df_to_bob

df = pl.DataFrame(
    {
        "Is_Visible": [True, False],
    }
)
bob = polars_df_to_bob(df, name="MeshBool")
```



Attributes to each point

```python
import numpy as np
import polars as pl
from csv_importer.parsers import polars_df_to_bob

df = pl.DataFrame({
    "Star": [
        [3.4, 3.5, 0.0],
        [3.1, 5.6, 0.0]
    ]
})
obj = polars_df_to_bob(df, name="MeshVector")
```

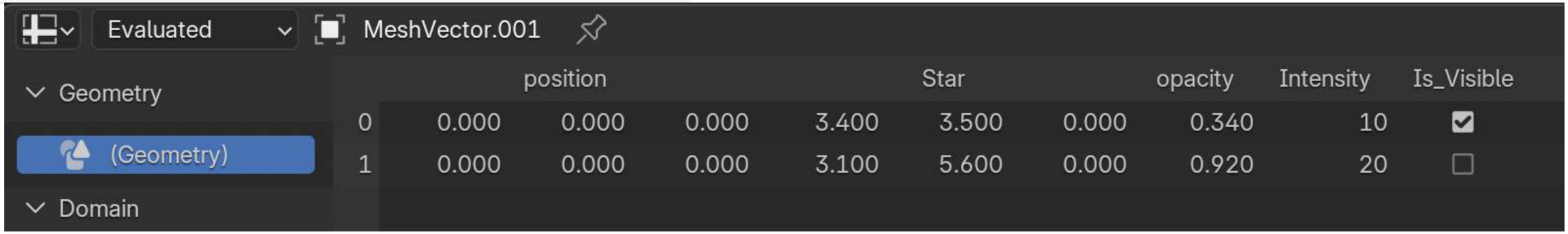
Attributes to each point

```python
import numpy as np
import polars as pl
from csv_importer.parsers import polars_df_to_bob

df = pl.DataFrame({
    "String": ["Hello", "World"],
})
# Stings are not supported
obj = polars_df_to_bob(df, name="MeshVector")
```

Automatically skipped

Attributes to each point

```python
# all together
import numpy as np
import polars as pl
from csv_importer.parsers import polars_df_to_bob

df = pl.DataFrame({
    "Intensity": [10, 20],
    "opacity": [0.34, 0.92],
    "Is_Visible": [True, False],
    "Star": [
        [3.4, 3.5, 0.0],
        [3.1, 5.6, 0.0]
    ]
})
obj = polars_df_to_bob(df, name="MeshVector")
```

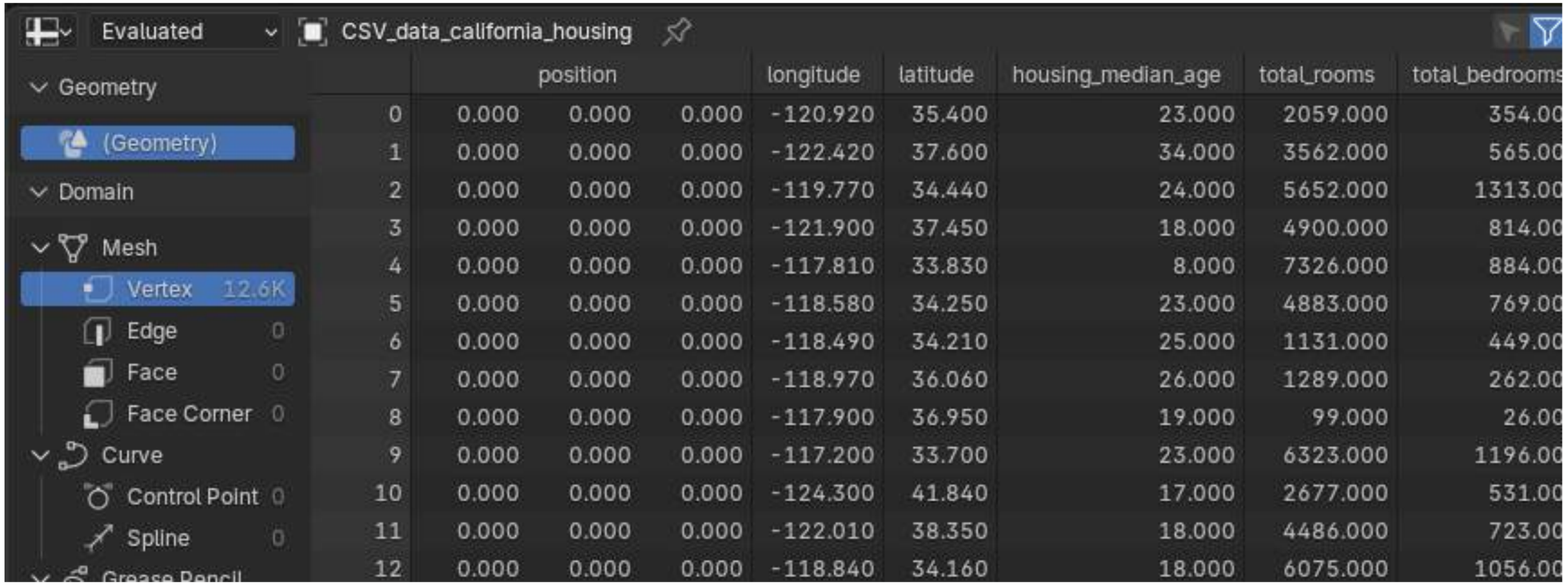


	position			Star			opacity	Intensity	Is_Visible
0	0.000	0.000	0.000	3.400	3.500	0.000	0.340	10	☑
1	0.000	0.000	0.000	3.100	5.600	0.000	0.920	20	☐

Attributes to each point

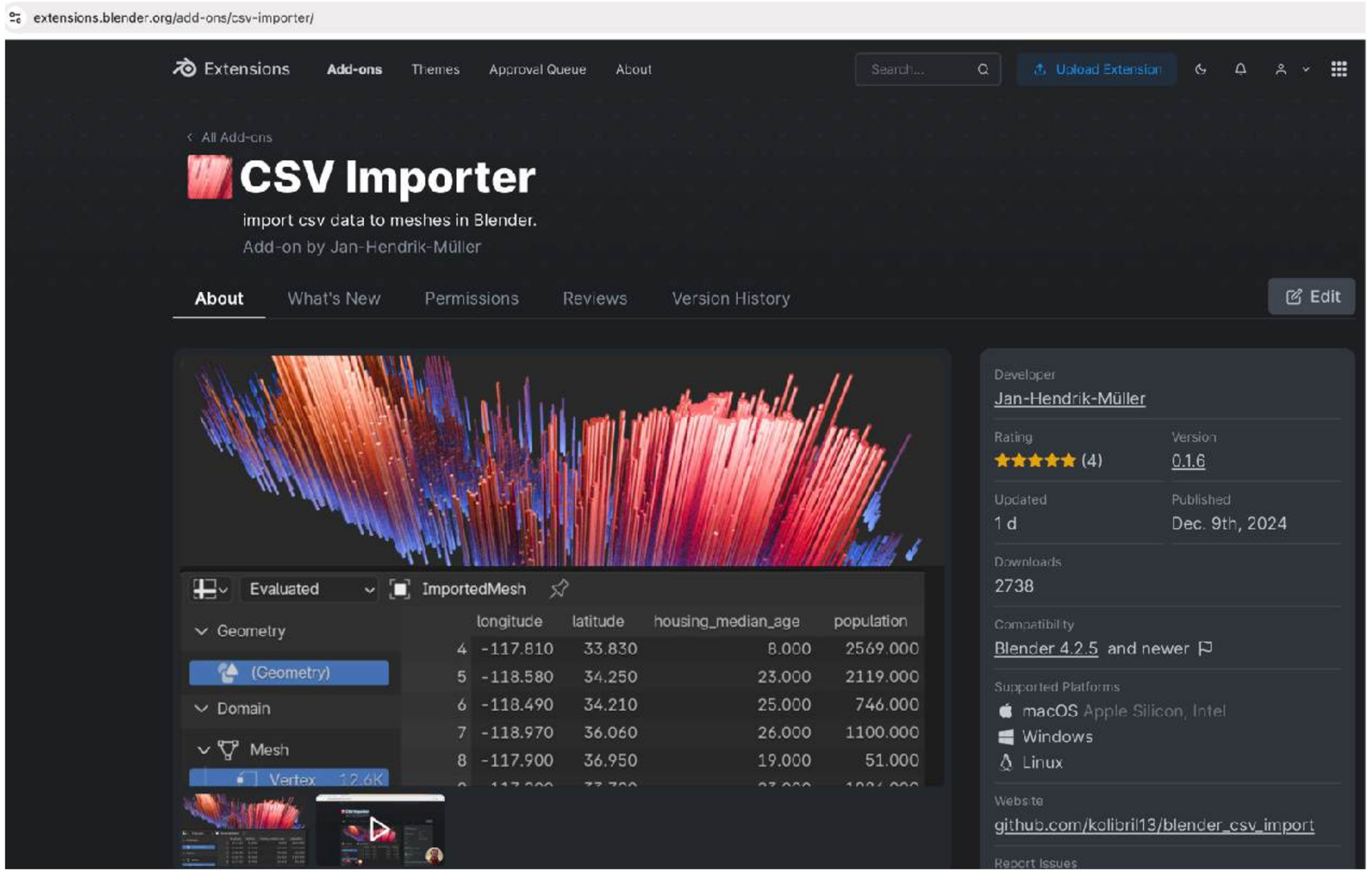
```
:
from pathlib import Path
from csv_importer.csv import load_csv

csv_path = Path.home() / "projects" / "blender_csv_import" / "docs" / "sample_datasets" / "data_california_housing.csv"

# `load_csv` is like `like polars_df_to_bob`, but with a csv path input
obj = load_csv(csv_path)

print(obj.name)
```

```
CSV_data_california_housing
```

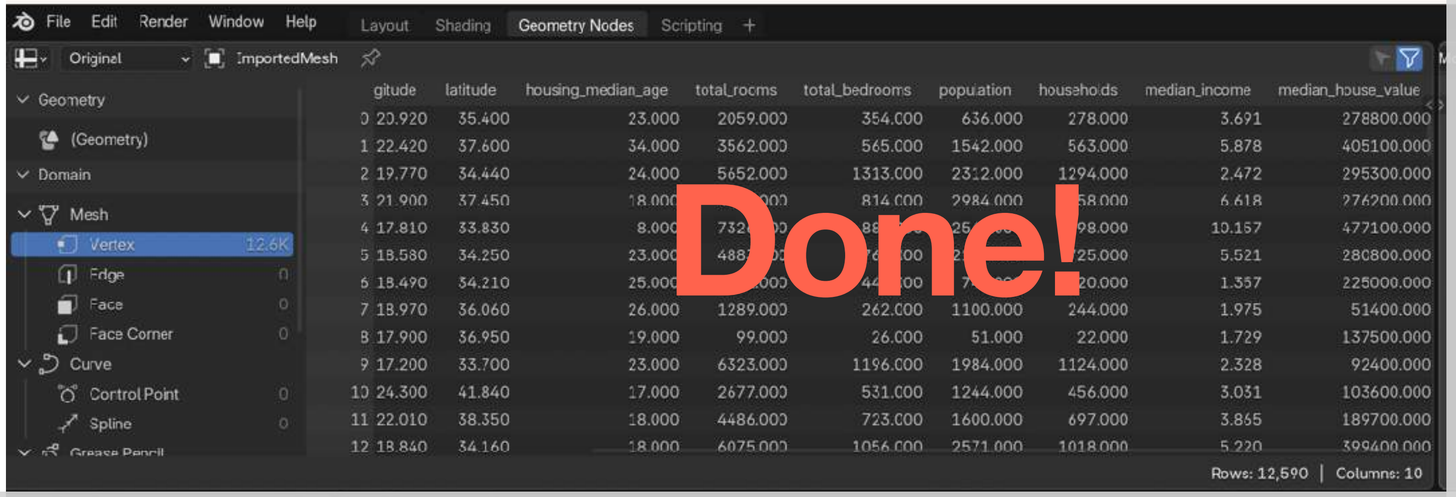


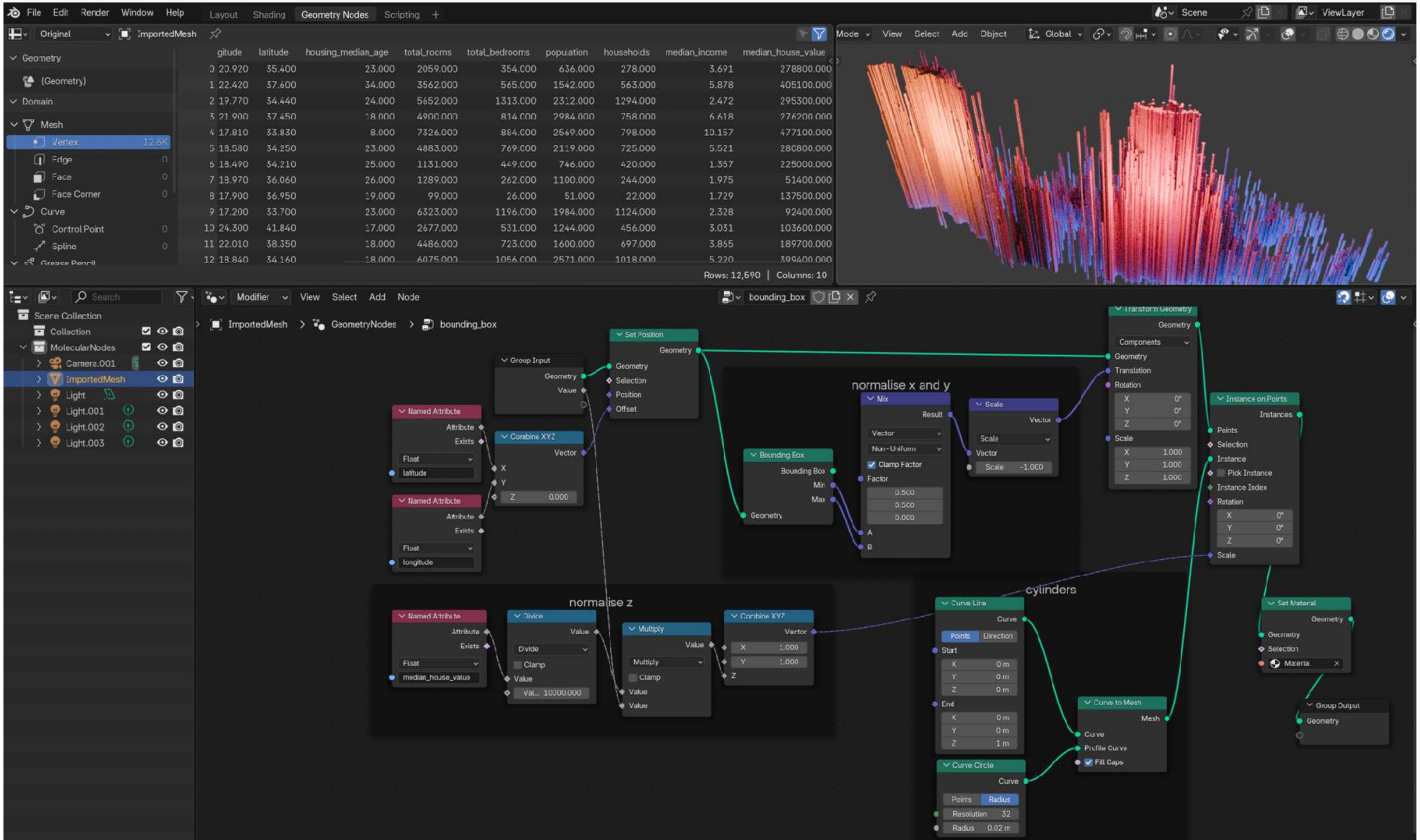
	position			longitude	latitude	housing_median_age	total_rooms	total_bedrooms
0	0.000	0.000	0.000	-120.920	35.400	23.000	2059.000	354.00
1	0.000	0.000	0.000	-122.420	37.600	34.000	3562.000	565.00
2	0.000	0.000	0.000	-119.770	34.440	24.000	5652.000	1313.00
3	0.000	0.000	0.000	-121.900	37.450	18.000	4900.000	814.00
4	0.000	0.000	0.000	-117.810	33.830	8.000	7326.000	884.00
5	0.000	0.000	0.000	-118.580	34.250	23.000	4883.000	769.00
6	0.000	0.000	0.000	-118.490	34.210	25.000	1131.000	449.00
7	0.000	0.000	0.000	-118.970	36.060	26.000	1289.000	262.00
8	0.000	0.000	0.000	-117.900	36.950	19.000	99.000	26.00
9	0.000	0.000	0.000	-117.200	33.700	23.000	6323.000	1196.00
10	0.000	0.000	0.000	-124.300	41.840	17.000	2677.000	531.00
11	0.000	0.000	0.000	-122.010	38.350	18.000	4486.000	723.00
12	0.000	0.000	0.000	-118.840	34.160	18.000	6075.000	1056.00

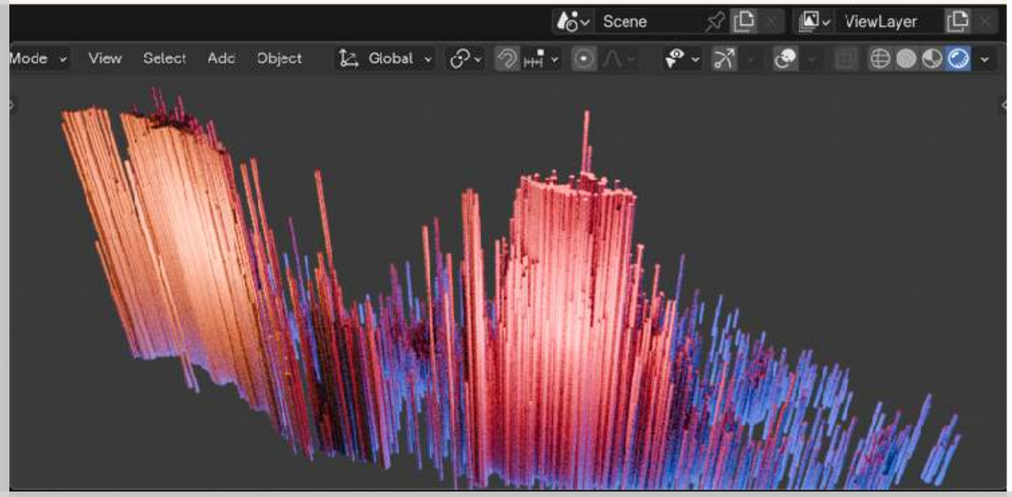
Also GUI version for CSV data

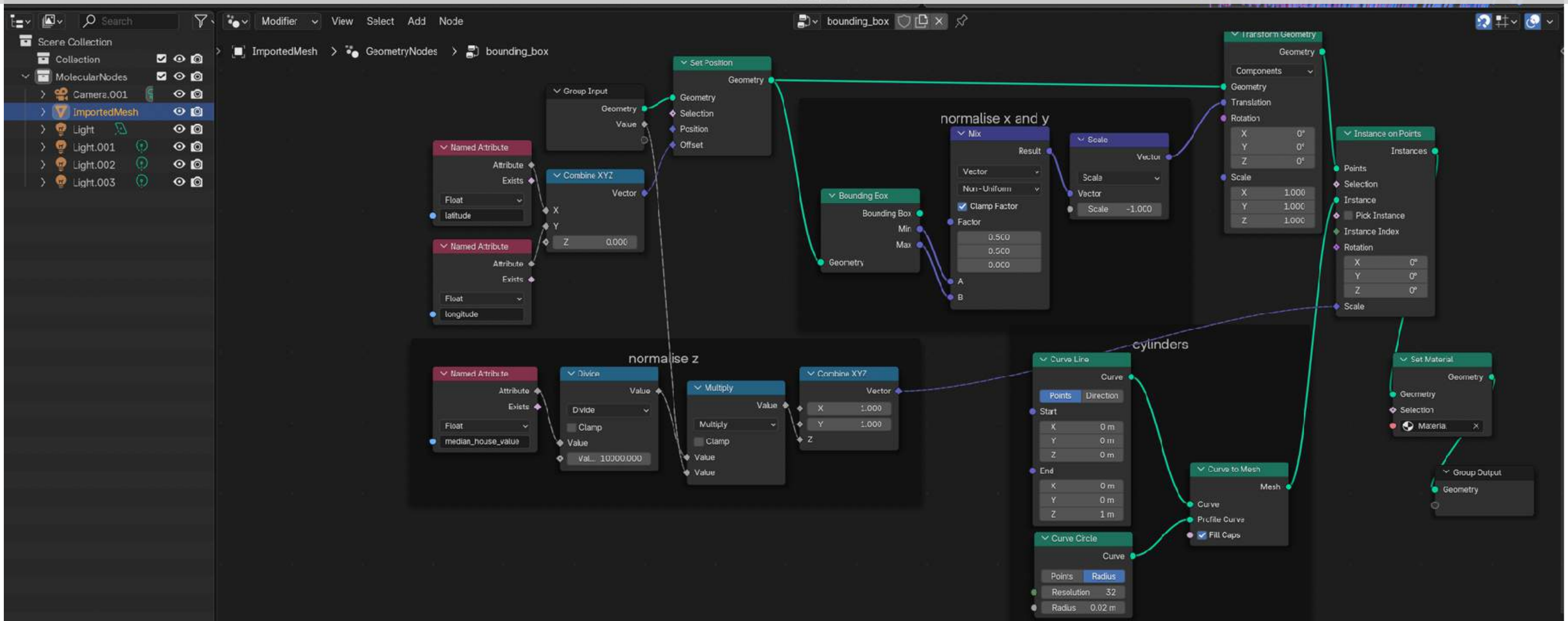


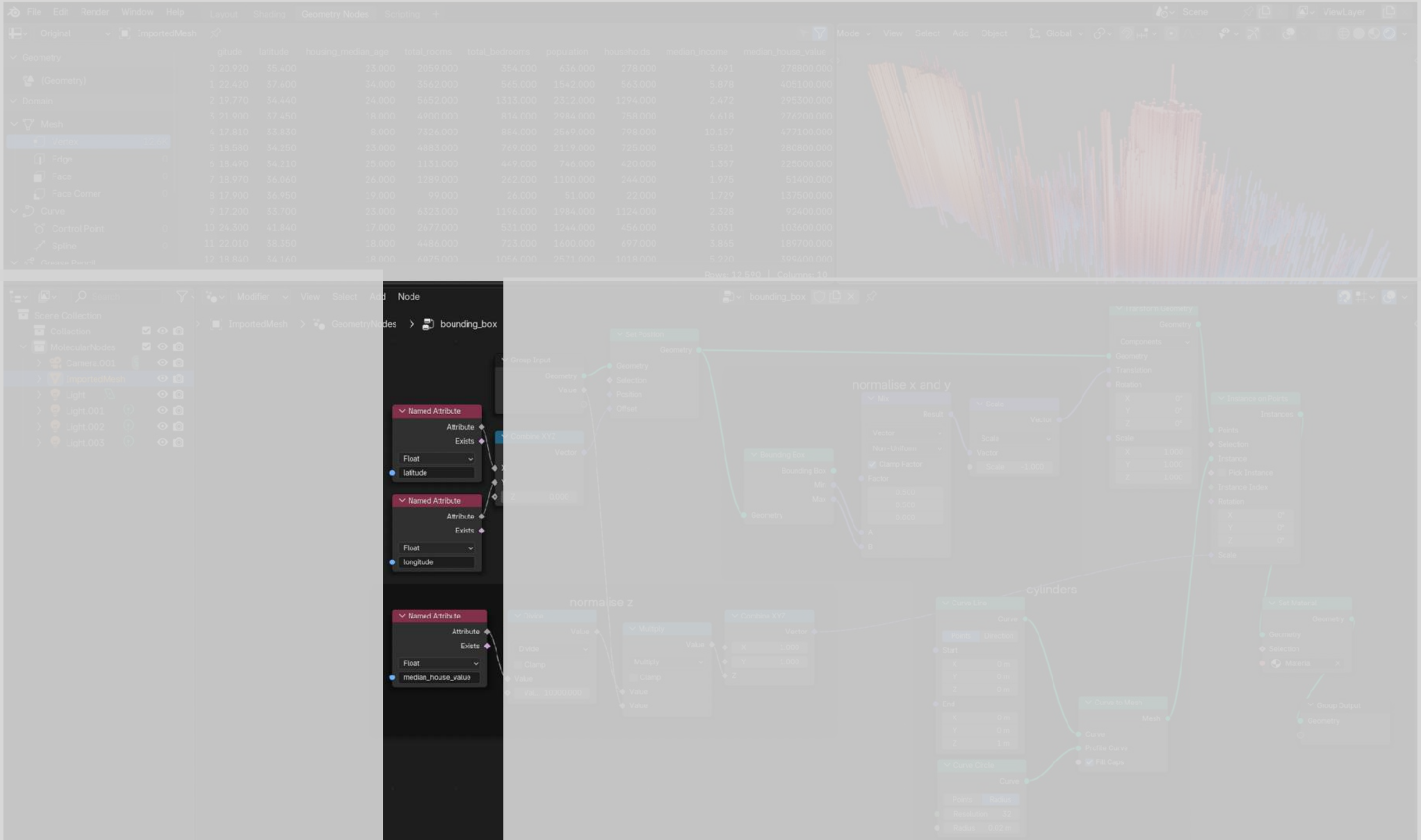
https://extensions.blender.org/add-ons/csv-importer/

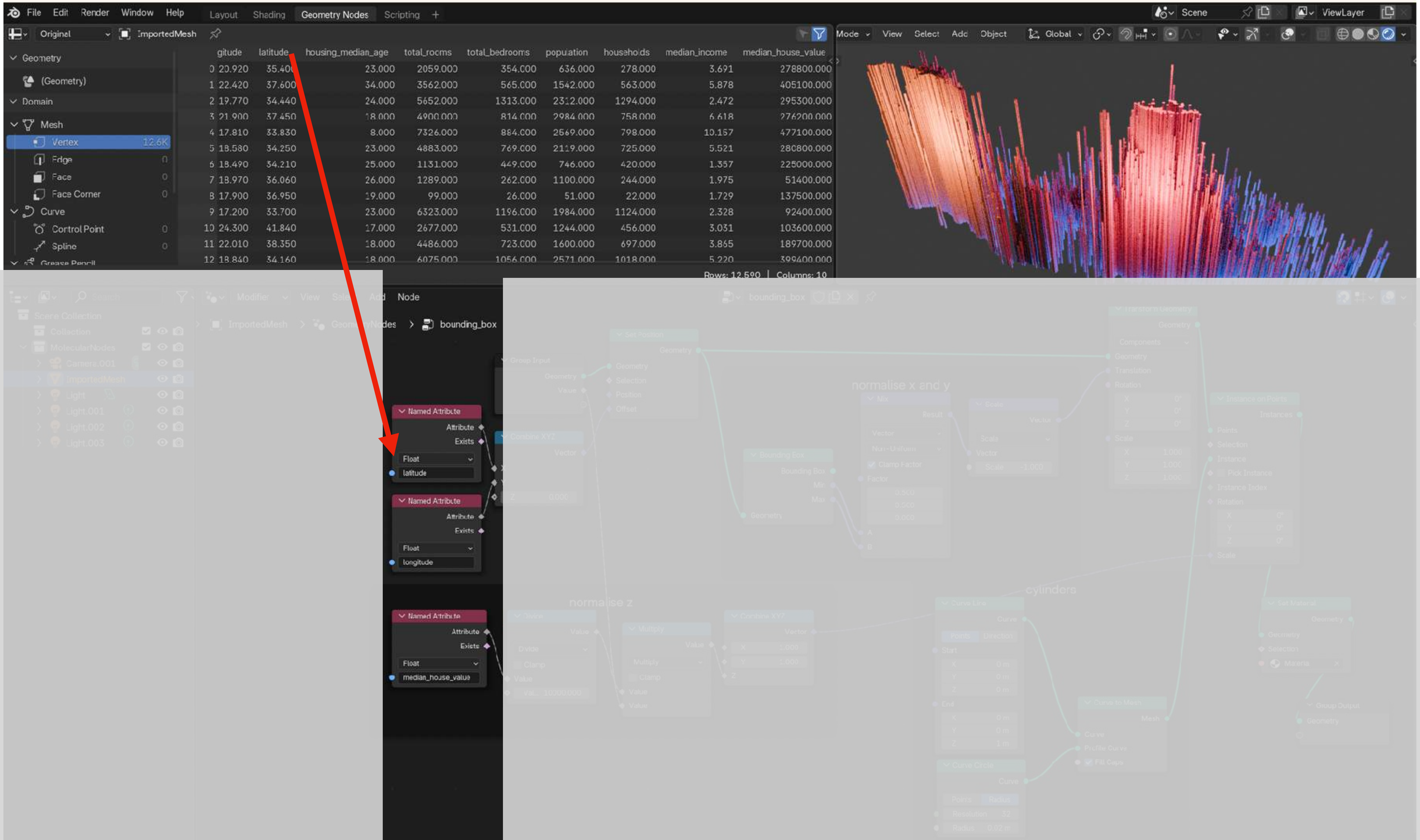

File Edit Render Window Help
Layout Shading Geometry Nodes Scripting
Original
ImportedMesh
Geometry
(Geometry)
Domain
Mesh
Vertex 12.6K
Edge 0
Face 0
Face Corner 0
Curve
Control Point 0
Spline 0
gitude latitude housing_median_age total_rooms total_bedrooms population households median_income median_house_value
0 20.920 35.400 23.000 2059.000 354.000 636.000 278.000 3.691 278800.000
1 22.420 37.600 34.000 3562.000 565.000 1542.000 563.000 5.878 405100.000
2 19.770 34.440 24.000 5652.000 1313.000 2312.000 1294.000 2.472 295300.000
3 21.900 37.450 18.000 814.000 2984.000 58.000 6.618 276200.000
4 17.810 33.830 8.000 98.000 10.157 477100.000
5 18.580 34.250 23.000 725.000 5.521 280800.000
6 18.490 34.210 25.000 20.000 1.357 225000.000
7 18.970 36.060 26.000 1289.000 262.000 1100.000 244.000 1.975 51400.000
8 17.900 36.950 19.000 99.000 26.000 51.000 22.000 1.729 137500.000
9 17.200 33.700 23.000 6323.000 1196.000 1984.000 1124.000 2.328 92400.000
10 24.300 41.840 17.000 2677.000 531.000 1244.000 456.000 3.031 103600.000
11 22.010 38.350 18.000 4486.000 723.000 1600.000 697.000 3.865 189700.000
12 18.840 34.160 18.000 6075.000 1056.000 2571.000 1018.000 5.220 399400.000
Rows: 12,590 | Columns: 10
Done!

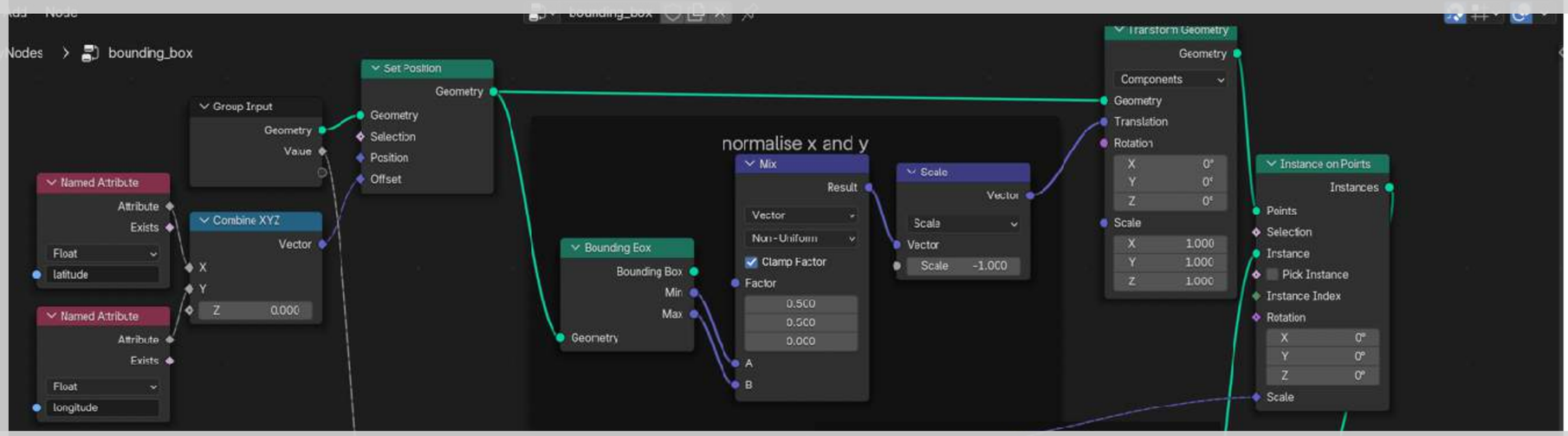

File Edit Render Window Help
Layout Shading Geometry Nodes Scripting
Scene ViewLayer
Original ImportedMesh
Geometry
(Geometry)
Domain
Mesh
Vertex 12.6K
Edge 0
Face 0
Face Corner 0
Curve
Control Point 0
Spline 0
gitude latitude housing_median_age total_rooms total_bedrooms population households median_income median_house_value
0 20.920 35.400 23.000 2059.000 354.000 636.000 278.000 3.691 278800.000
1 22.420 37.600 34.000 3562.000 565.000 1542.000 563.000 5.878 405100.000
2 19.770 34.440 24.000 5652.000 1313.000 2312.000 1294.000 2.472 295300.000
3 21.900 37.450 18.000 4900.000 814.000 2984.000 758.000 6.618 276200.000
4 17.810 33.830 8.000 7326.000 884.000 2569.000 798.000 10.157 477100.000
5 18.580 34.250 23.000 4883.000 769.000 2119.000 725.000 5.521 280800.000
6 18.490 34.210 25.000 1131.000 449.000 746.000 420.000 1.357 225000.000
7 18.970 36.060 26.000 1289.000 262.000 1100.000 244.000 1.975 51400.000
8 17.900 36.950 19.000 99.000 26.000 51.000 22.000 1.729 137500.000
9 17.200 33.700 23.000 6323.000 1196.000 1984.000 1124.000 2.328 92400.000
10 24.300 41.840 17.000 2677.000 531.000 1244.000 456.000 3.031 103600.000
11 22.010 38.350 18.000 4486.000 723.000 1600.000 697.000 3.855 189700.000
12 18.840 34.160 18.000 6075.000 1056.000 2571.000 1018.000 5.220 399400.000
Rows: 12,590 | Columns: 10
Mode View Select Add Object Global
Search
Scene Collection
Collection
MolecularNodes
Camera.001
ImportedMesh
Light
Light.001
Light.002
Light.003
Modifier View Select Add Node
bounding_box
ImportedMesh GeometryNodes bounding_box
Group Input
Geometry
Value
Set Position
Geometry
Selection
Position
Offset
Named Attribute
Attribute
Exists
Float
latitude
Named Attribute
Attribute
Exists
Float
longitude
Combine XYZ
Vector
X
Y
Z 0.000
normalise x and y
Bounding Box
Bounding Box
Min
Max
Geometry
Mix
Result
Vector
Non-Uniform
Clamp Factor
Factor
0.500
0.500
0.000
A
B
Scale
Vector
Scale
Vector
Scale -1.000
Transform Geometry
Geometry
Components
Geometry
Translation
Rotation
X 0°
Y 0°
Z 0°
Scale
X 1.000
Y 1.000
Z 1.000
Instance on Points
Instances
Points
Selection
Instance
Pick Instance
Instance Index
Rotation
X 0°
Y 0°
Z 0°
Scale
normalise z
Named Attribute
Attribute
Exists
Float
median_house_value
Divide
Value
Divide
Clamp
Value
Val... 10000.000
Multiply
Value
Multiply
Clamp
Value
Value
Combine XYZ
Vector
X 1.000
Y 1.000
Z
cylinders
Curve Line
Curve
Points Direction
Start
X 0 m
Y 0 m
Z 0 m
End
X 0 m
Y 0 m
Z 1 m
Curve Circle
Curve
Points Radius
Resolution 32
Radius 0.02 m
Curve to Mesh
Mesh
Curve
Profile Curve
Fill Caps
Set Material
Geometry
Geometry
Selection
Materia
Group Output
Geometry

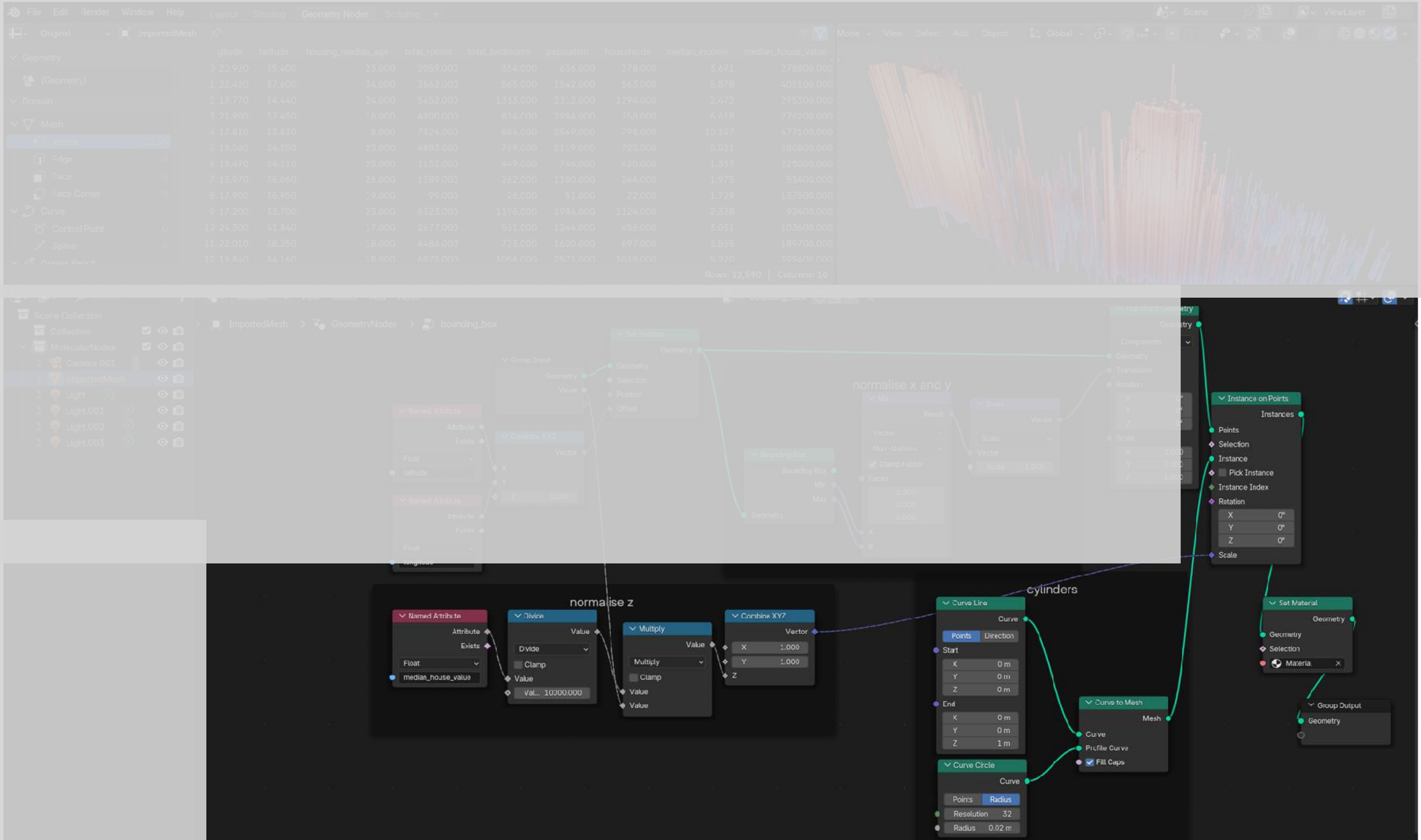

Scene
ViewLayer
Mode
View
Select
Add
Object
Global

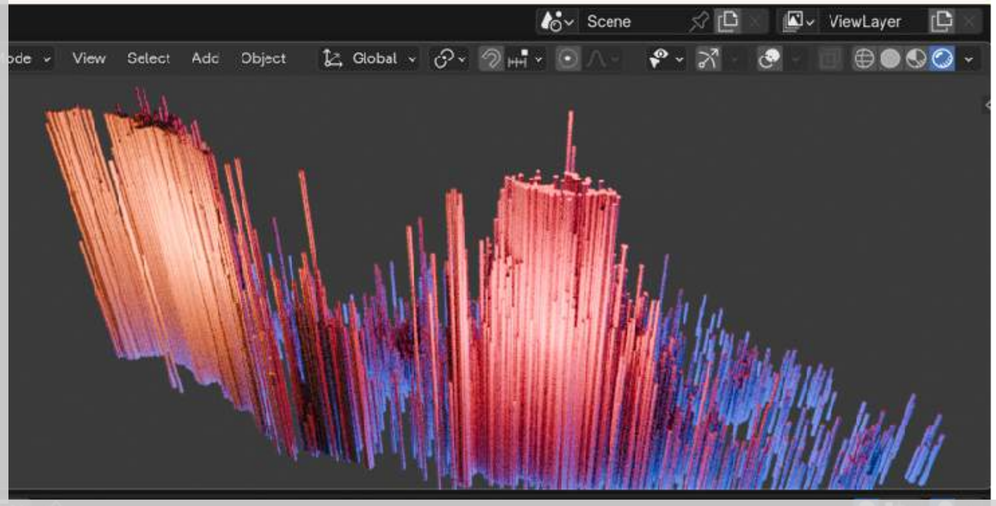

Modifier
View
Select
Add
Node
bounding_box
ImportedMesh
GeometryNodes
bounding_box
Scene Collection
Collection
MolecularNodes
Camera.001
ImportedMesh
Light
Light.001
Light.002
Light.003
Group Input
Set Position
Named Attribute
Combine XYZ
Named Attribute
normalise x and y
Bounding Box
Mix
Scale
Transform Geometry
Instance on Points
normalise z
Named Attribute
Divide
Multiply
Combine XYZ
cylinders
Curve Line
Curve Circle
Curve to Mesh
Set Material
Group Output

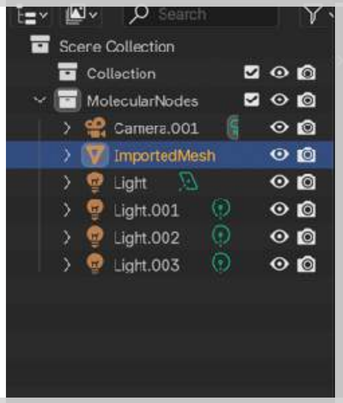

Node
bounding_box
Named Attribute
Attribute
Exists
Float
latitude
longitude
median_house_value
normalise x and y
normalise z
cylinders


File Edit Render Window Help
Layout Shading Geometry Nodes Scripting
Scene ViewLayer
Original ImportedMesh
Mode View Select Add Object Global
Geometry
(Geometry)
Domain
Mesh
Vertex 12.6K
Edge 0
Face 0
Face Corner 0
Curve
Control Point 0
Spline 0
gitude latitude housing_median_age total_rooms total_bedrooms population households median_income median_house_value
0 20.920 35.400 23.000 2059.000 354.000 636.000 278.000 3.691 278800.000
1 22.420 37.600 34.000 3562.000 565.000 1542.000 563.000 5.878 405100.000
2 19.770 34.440 24.000 5652.000 1313.000 2312.000 1294.000 2.472 295300.000
3 21.900 37.450 18.000 4900.000 814.000 2984.000 758.000 6.618 276200.000
4 17.810 33.830 8.000 7326.000 884.000 2569.000 798.000 10.157 477100.000
5 18.580 34.250 23.000 4883.000 769.000 2119.000 725.000 5.521 280800.000
6 18.490 34.210 25.000 1131.000 449.000 746.000 420.000 1.357 225000.000
7 18.970 36.060 26.000 1289.000 262.000 1100.000 244.000 1.975 51400.000
8 17.900 36.950 19.000 99.000 26.000 51.000 22.000 1.729 137500.000
9 17.200 33.700 23.000 6323.000 1196.000 1984.000 1124.000 2.328 92400.000
10 24.300 41.840 17.000 2677.000 531.000 1244.000 456.000 3.031 103600.000
11 22.010 38.350 18.000 4486.000 723.000 1600.000 697.000 3.855 189700.000
12 18.840 34.160 18.000 6075.000 1056.000 2571.000 1018.000 5.220 399400.000
Rows: 12,590 | Columns: 10
Node
bounding_box
Named Attribute
Attribute
Exists
Float
latitude
longitude
median_house_value
normalise x and y
normalise z
cylinders


bounding_box
Set Position
Geometry
Selection
Position
Offset
Group Input
Value
Named Attribute
Attribute
Exists
Float
latitude
Combine XYZ
Vector
X
Y
Z
0.000
Named Attribute
Attribute
Exists
Float
longitude
normalise x and y
Bounding Box
Bounding Box
Min
Max
Geometry
Mix
Result
Vector
Non-Uniform
Clamp Factor
Factor
0.500
0.500
0.000
A
B
Scale
Vector
Scale
Vector
Scale -1.000
Transform Geometry
Geometry
Components
Geometry
Translation
Rotation
X 0°
Y 0°
Z 0°
Scale
X 1.000
Y 1.000
Z 1.000
Instance on Points
Instances
Points
Selection
Instance
Pick Instance
Instance Index
Rotation
X 0°
Y 0°
Z 0°
Scale


normalise z
cylinders
normalise x and y
Named Attribute
Divide
Multiply
Combine XYZ
Curve Line
Curve Circle
Curve to Mesh
Instance on Points
Set Material
Group Output
median_house_value


Scene
ViewLayer
View
Select
Add
Object
Global


Search
Scene Collection
Collection
MolecularNodes
Camera.001
ImportedMesh
Light
Light.001
Light.002
Light.003

normalise x and y
normalise z
cylinders

Time Series

WUP2018-F22-Cities_Over_300K_Annual

View | Zoom 125 % | Add Category | Pivot Table | Insert | Table | Chart | Text | Shape | Media | Comment | Share | Format | Organise

Data | NOTES

Sheet

Sheet Name: Data

Background

Duplicate Sheet

Delete Sheet

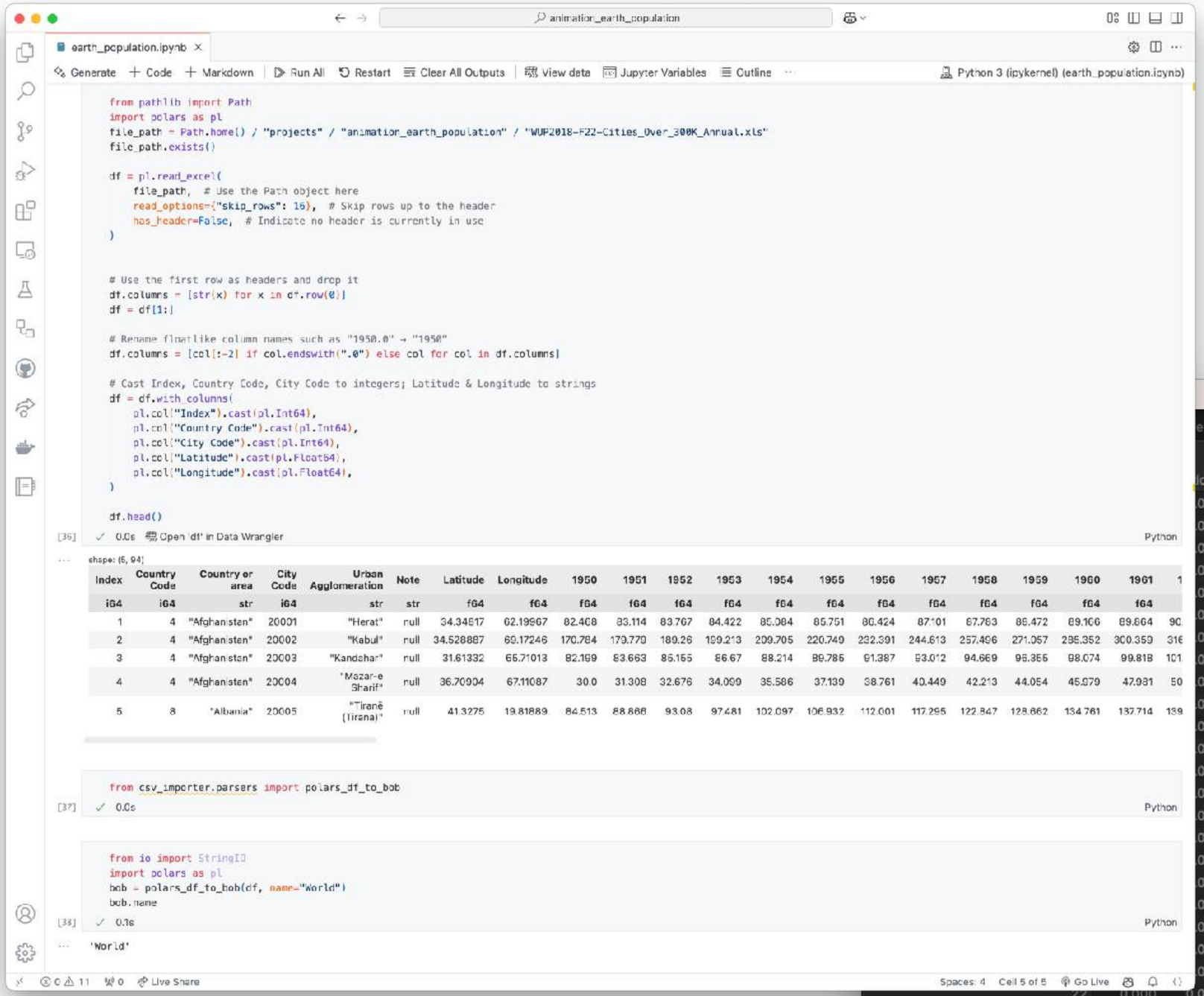
United Nations
Population Division
Department of Economic and Social Affairs

World Urbanization Prospects: The 2018 Revision
File 22: Annual Population of Urban Agglomerations with 300,000 Inhabitants or More in 2018, by Country, 1950-2035 (thousands)

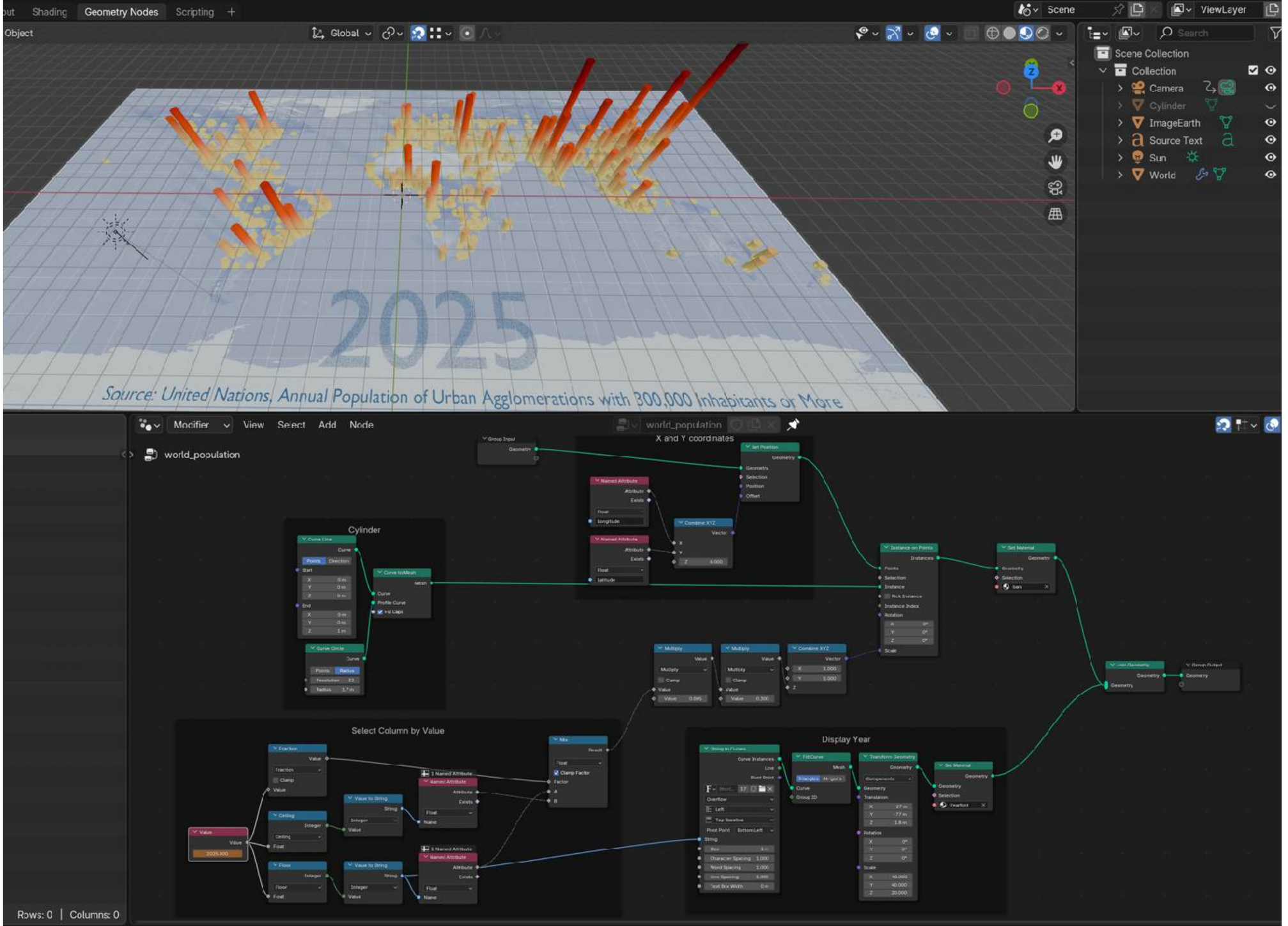
POP/DB/WUP/Rev.2018/1/F22

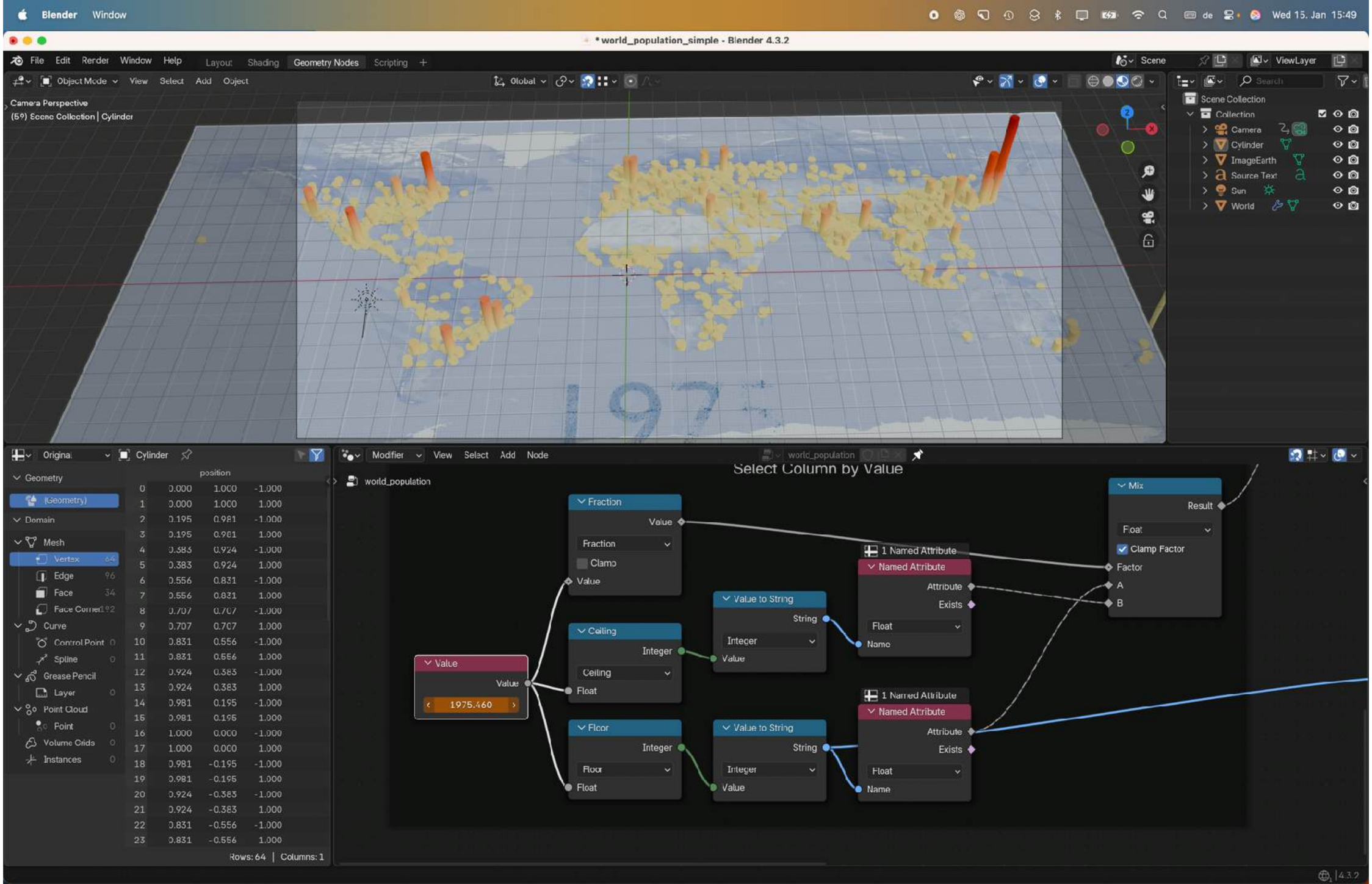
Suggested citation: United Nations, Department of Economic and Social Affairs, Population Division (2018). World Urbanization Prospects: The 2018 Revision, Online Edition.

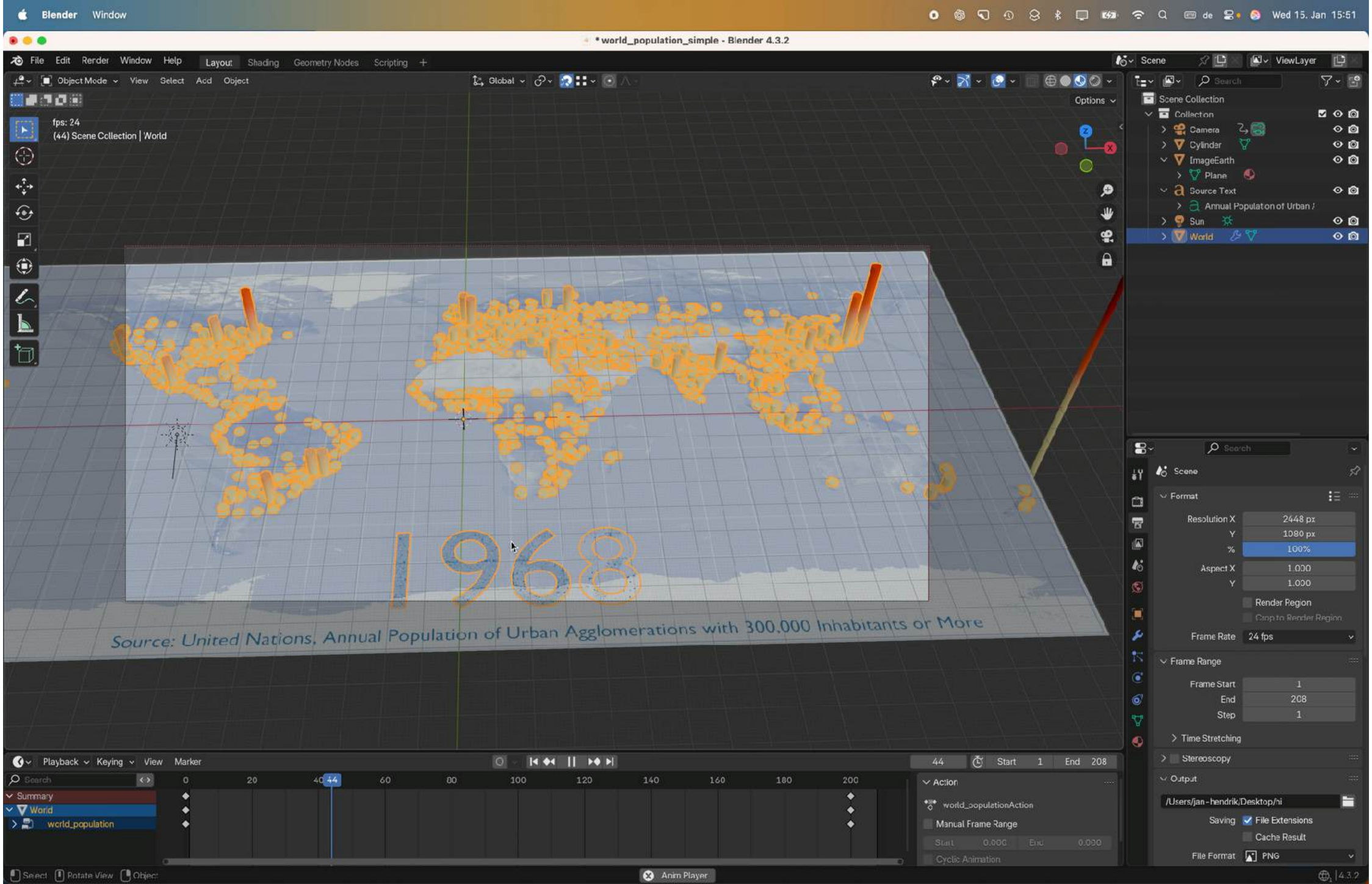
Index	Country Code	Country or area	City Code	Urban Agglomeration	Note	Latitude	Longitude	Annual Population of Urban Agglomerations with 300,000 Inhabitants or More in 2018, 1950-2035 (thousands)											
								1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1
1	4	Afghanistan	20001	Herat		34,3482	62,1997	82	83	84	84	85	86	86	87	88	88	89	
2	4	Afghanistan	20002	Kabul		34,5289	69,1725	171	180	189	199	210	221	232	245	257	271	285	
3	4	Afghanistan	20003	Kandahar		31,6133	65,7101	82	84	85	87	88	90	91	93	95	96	98	
4	4	Afghanistan	20004	Mazar-e Sharif		36,7090	67,1109	30	31	33	34	36	37	39	40	42	44	46	
5	8	Albania	20005	Tirané (Tirana)		41,3275	19,8189	65	89	93	97	102	107	112	117	123	129	135	
6	12	Algeria	20009	Annaba		36,9000	7,7667	100	103	106	109	113	116	120	124	128	132	136	
7	12	Algeria	20011	Batna		35,5560	6,1741	13	14	16	17	19	21	23	25	27	30	33	
8	12	Algeria	20015	Blida		36,4808	2,8319	34	36	38	41	43	46	49	52	55	59	62	
9	12	Algeria	20006	El Djazaïr (Algiers)	1	36,7525	3,0420	516	534	552	570	589	623	666	712	762	815	872	
10	12	Algeria	20020	El Djelfa		34,6728	3,2630	10	11	12	12	13	14	15	16	16	17	18	
11	12	Algeria	20019	Qacentina		36,3549	6,6073	88	94	100	107	114	122	130	139	148	158	169	
12	12	Algeria	20037	Sétif		36,1911	5,4137	41	43	45	48	50	53	56	58	61	64	67	
13	12	Algeria	20035	Wahran (Oran)		35,6911	-0,6417	269	272	276	279	283	286	290	294	297	301	305	
14	24	Angola	895000001	Benguela	2	-12,5760	13,4050	14	15	16	16	17	18	19	20	21	22	23	
15	24	Angola	204043	Cabinda	3	-5,5500	12,2000	5	5	5	5	5	5	5	5	5	5	5	
16	24	Angola	895000011	Cuito		-12,3600	16,9300	9	9	9	10	10	10	11	11	12	12	13	
17	24	Angola	20050	Huambo	4	-12,7761	15,7392	15	17	18	20	22	24	26	28	31	34	37	
18	24	Angola	895000000	Lobito	5	-12,3640	13,5360	23	25	27	29	31	33	36	39	42	45	48	
19	24	Angola	20049	Luanda	6	-8,8368	13,2343	138	145	152	159	166	174	182	191	200	210	219	
20	24	Angola	204044	Lubango	7	-14,9172	13,4925	12	12	12	12	13	13	13	14	14	14	15	
21	24	Angola	895000004	Malanje	8	-9,5400	16,3410	9	10	11	11	12	13	14	15	16	17	18	
22	24	Angola	895000012	Uige		-7,1069	15,4436	3	3	4	4	4	4	5	5	5	6	6	
23	32	Argentina	20057	Bahia Blanca		-38,7196	-62,2724	116	117	118	119	120	121	122	123	124	125	126	
24	32	Argentina	20058	Buenos Aires	9	-34,6051	-58,4004	5 166	5 307	5 452	5 601	5 753	5 910	6 072	6 237	6 407	6 582	6 762	
25	32	Argentina	20059	Córdoba	10	-31,4135	-64,1811	429	444	460	476	493	510	528	548	565	585	605	
26	32	Argentina	20063	Corrientes	11	-27,4606	-58,8341	64	67	70	73	76	79	82	85	89	93	97	
27	32	Argentina	20065	La Plata	12	-34,9215	-57,9545	300	308	318	327	337	347	357	368	378	390	401	
28	32	Argentina	20067	Mar Del Plata		-38,0023	-57,5575	132	139	146	152	159	166	174	182	191	200	209	
29	32	Argentina	20068	Mendoza	13	-32,8908	-68,8272	246	253	260	268	276	284	292	301	310	319	328	
30	32	Argentina	20070	Neuquén-Plottier-Cipolletti		-38,9514	-68,0648	14	15	17	19	20	23	25	27	30	33	37	
31	32	Argentina	20072	Posadas		-27,3671	-55,8961	46	48	50	53	55	58	61	63	67	70	73	
32	32	Argentina	20073	Resistencia	14	-27,4606	-58,9839	73	75	78	81	84	87	90	94	97	101	105	
33	32	Argentina	20075	Rosario	15	-32,9468	-60,6393	554	565	576	587	599	610	622	634	646	659	671	
34	32	Argentina	20076	Salta		-24,7859	-65,4117	77	81	84	87	91	95	99	103	108	112	117	
35	32	Argentina	20078	San Juan	16	-31,5375	-68,5364	108	112	115	119	123	127	131	135	140	144	149	
36	32	Argentina	20081	San Miguel de Tucumán	17	-26,8241	-65,2226	224	231	237	245	252	260	267	275	284	292	301	
37	32	Argentina	20084	San Salvador de Jujuy	18	-24,1946	-65,2971	34	35	36	37	38	39	40	41	42	43	44	
38	32	Argentina	20085	Santa Fe	19	-31,6327	-60,6983	214	219	224	229	234	239	244	249	255	260	266	
39	32	Argentina	20086	Santiago Del Estero	20	-27,7951	-64,2615	83	85	86	88	90	93	95	97	99	101	104	
40	51	Armenia	20140	Yerevan		40,1820	44,5145	341	358	375	393	412	431	452	474	497	518	538	
41	36	Australia	205171	Adelaide	21	-34,9287	138,5986	429	443	456	470	484	497	511	526	541	556	572	
42	36	Australia	205170	Brisbane	21	-27,4679	153,0231	442	456	471	486	502	518	534	550	567	585	603	
43	36	Australia	205176	Canberra	21	-35,2835	149,1281	20	22	24	26	28	31	35	38	42	46	51	
44	36	Australia	895000015	Central Coast	21	-33,4200	151,3958	72	74	76	78	80	82	84	87	89	92	94	
45	36	Australia	20093	Gold Coast-Tweed Head	21	-28,0003	153,4309	29	31	32	34	36	38	40	42	44	47	50	
46	36	Australia	205168	Melbourne	21	-37,8140	144,9633	1 332	1 377	1 424	1 473	1 524	1 574	1 626	1 680	1 735	1 792	1 851	
47	36	Australia	205177	Newcastle-Maitland	21	-32,9249	151,7409	161	166	170	174	178	182	186	191	195	199	204	
48	36	Australia	205172	Perth	21	-31,9478	115,8525	311	320	330	339	349	358	368	379	388	398	409	

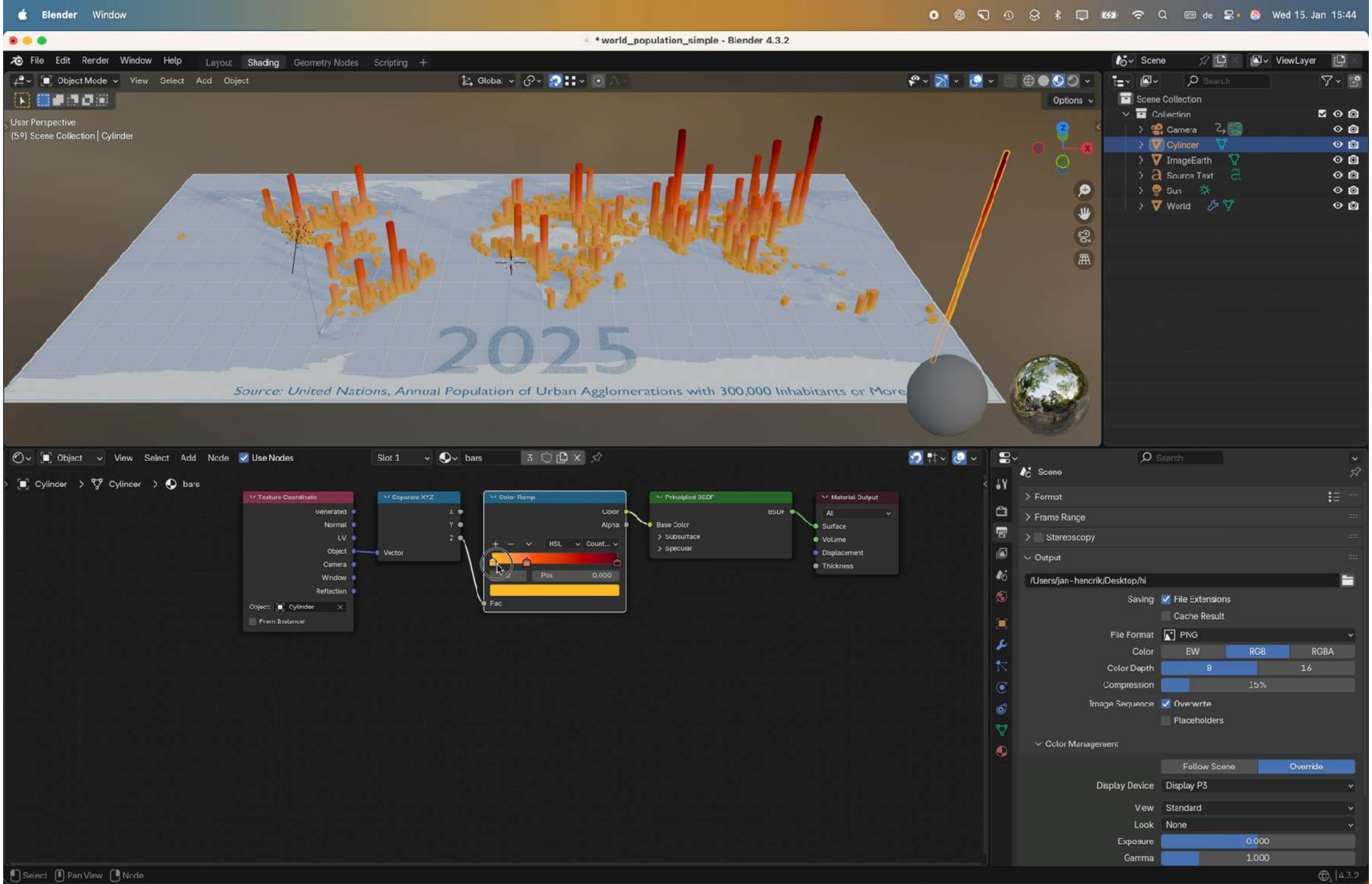

animation_earth_population
earth_population.ipynb
Generate
Code
Markdown
Run All
Restart
Clear All Outputs
View data
Jupyter Variables
Outline
Python 3 (ipykernel) (earth_population.ipynb)
from pathlib import Path
import polars as pl
file_path = Path.home() / "projects" / "animation_earth_population" / "WUP2018-F22-Cities_Over_300K_Annual.xls"
file_path.exists()
df = pl.read_excel(
file_path, # Use the Path object here
read_options={"skip_rows": 16}, # Skip rows up to the header
has_header=False, # Indicate no header is currently in use
)
# Use the first row as headers and drop it
df.columns = [str(x) for x in df.row(0)]
df = df[1:]
# Rename floatlike column names such as "1950.0" → "1950"
df.columns = [col[:-2] if col.endswith(".0") else col for col in df.columns]
# Cast Index, Country Code, City Code to integers; Latitude & Longitude to strings
df = df.with_columns(
pl.col("Index").cast(pl.Int64),
pl.col("Country Code").cast(pl.Int64),
pl.col("City Code").cast(pl.Int64),
pl.col("Latitude").cast(pl.Float64),
pl.col("Longitude").cast(pl.Float64),
)
df.head()
[36]
0.0s
Open 'df' in Data Wrangler
Python
shape: (5, 94)
Index
Country Code
Country or area
City Code
Urban Agglomeration
Note
Latitude
Longitude
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
i64
i64
str
i64
str
str
f64
f64
f64
f64
f64
f64
f64
f64
f64
f64
f64
f64
f64
f64
1
4
"Afghanistan"
20001
"Herat"
null
34.34817
62.19967
82.468
83.114
83.767
84.422
85.084
85.751
86.424
87.101
87.783
88.472
89.166
89.864
90
2
4
"Afghanistan"
20002
"Kabul"
null
34.528887
69.17246
170.784
179.779
189.26
199.213
209.705
220.749
232.391
244.613
257.496
271.057
285.352
300.359
316
3
4
"Afghanistan"
20003
"Kandahar"
null
31.61332
65.71013
82.199
83.663
85.155
86.67
88.214
89.785
91.387
93.012
94.669
96.355
98.074
99.818
101
4
4
"Afghanistan"
20004
"Mazar-e Sharif"
null
36.70904
67.11087
30.0
31.308
32.676
34.099
35.586
37.139
38.761
40.449
42.213
44.054
45.979
47.991
50
5
8
"Albania"
20005
"Tiranë (Tirana)"
null
41.3275
19.81889
84.513
88.866
93.08
97.481
102.097
106.932
112.001
117.295
122.847
128.662
134.761
137.714
139
from csv_importer.parsers import polars_df_to_bob
[37]
0.0s
Python
from io import StringIO
import polars as pl
bob = polars_df_to_bob(df, name="World")
bob.name
[38]
0.1s
Python
'World'
Live Share
Spaces: 4
Cell 5 of 5
Go Live

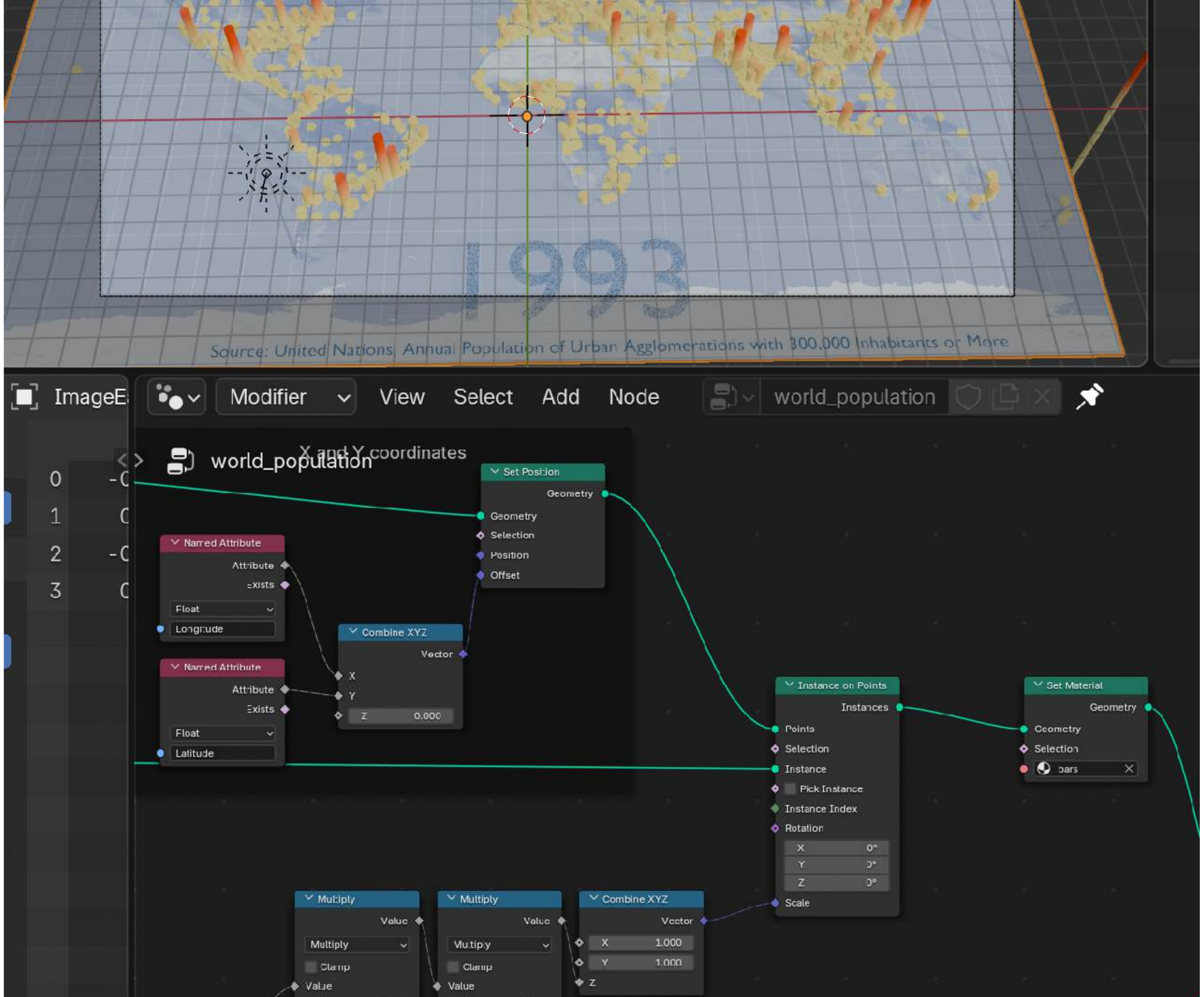
* (Unsaved) - Blender 4.3.2
exture Paint
Shading
Animation
Rendering
Compositing
Geometry Nodes
Scripting
ion
Latitude
Longitude
1950
1951
1952
1953
1954
1955
000 0.000 34.348 62.200 82.468 83.114 83.767 84.422 85.084 85.7
000 0.000 34.529 69.172 170.784 179.779 189.260 199.213 209.705 220.7
000 0.000 31.613 65.710 82.199 83.663 85.155 86.670 88.214 89.7
000 0.000 36.709 67.111 30.000 31.308 32.676 34.099 35.586 37.1
000 0.000 41.327 19.819 84.513 88.866 93.080 97.481 102.097 106.9
000 0.000 36.900 7.767 99.741 102.878 106.118 109.451 112.893 116.4
000 0.000 35.556 6.174 13.046 14.306 15.689 17.201 18.862 20.6
000 0.000 36.481 2.832 33.970 36.092 38.349 40.742 43.286 45.9
000 0.000 36.752 3.042 516.450 533.740 551.632 570.074 589.158 622.9
000 0.000 34.673 3.263 10.450 11.061 11.708 12.392 13.117 13.8
000 0.000 36.355 6.607 88.001 93.924 100.254 106.992 114.193 121.8
000 0.000 36.191 5.414 41.289 43.327 45.468 47.709 50.063 52.5
000 0.000 35.691 -0.642 268.935 272.335 275.782 279.263 282.793 286.3
000 0.000 -12.576 13.405 14.364 15.042 15.752 16.493 17.270 18.0
000 0.000 -5.550 12.200 4.722 4.710 4.698 4.687 4.675 4.6
000 0.000 -12.383 16.933 8.632 8.970 9.322 9.686 10.066 10.4
000 0.000 -12.776 15.739 15.306 16.721 18.270 19.957 21.802 23.8
000 0.000 -12.364 13.536 23.027 24.801 26.714 28.769 30.986 33.3
000 0.000 -8.837 13.234 138.413 144.939 151.782 158.928 166.422 174.2
000 0.000 -14.917 13.493 11.555 11.846 12.146 12.453 12.767 13.0
000 0.000 -9.540 16.341 9.176 9.834 10.541 11.296 12.107 12.9
000 0.000 -7.107 15.441 3.201 3.414 3.642 3.884 4.143 4.4
22 0.000 0.000 -38.720 -62.272 115.752 116.774 117.807 118.846 119.895 120.9
23 0.000 0.000 0.000 -34.605 -58.400 5166.140 5307.065 5452.036 5600.553 5753.328 5910.2
24 0.000 0.000 0.000 -31.413 -64.181 429.249 444.257 459.811 475.865 492.502 509.6


Shading
Geometry Nodes
Scripting
Object
Global
Scene
ViewLayer
Search
Scene Collection
Collection
Camera
Cylinder
ImageEarth
Source Text
Sun
World
2025
Source: United Nations, Annual Population of Urban Agglomerations with 300,000 Inhabitants or More
Modifier
View
Select
Add
Node
world_population
X and Y coordinates
Cylinder
Select Column by Value
Display Year
Rows: 0 | Columns: 0


Blender Window
Wed 15. Jan 15:49
*world_population_simple - Blender 4.3.2
File Edit Render Window Help Layout Shading Geometry Nodes Scripting
Object Mode View Select Add Object
Global
Camera Perspective
(59) Scene Collection | Cylinder
Scene
ViewLayer
Scene Collection
Collection
Camera
Cylinder
ImageEarth
Source Text
Sun
World
Original
Cylinder
Geometry
(Geometry)
Domain
Mesh
Vertex 64
Edge 96
Face 34
Face Corner 192
Curve
Control Point 0
Spline 0
Grease Pencil
Layer 0
Point Cloud
Point 0
Volume Grids 0
Instances 0
position
0 0.000 1.000 -1.000
1 0.000 1.000 1.000
2 0.195 0.981 -1.000
3 0.195 0.981 1.000
4 0.383 0.924 -1.000
5 0.383 0.924 1.000
6 0.556 0.831 -1.000
7 0.556 0.831 1.000
8 0.707 0.707 -1.000
9 0.707 0.707 1.000
10 0.831 0.556 -1.000
11 0.831 0.556 1.000
12 0.924 0.383 -1.000
13 0.924 0.383 1.000
14 0.981 0.195 -1.000
15 0.981 0.195 1.000
16 1.000 0.000 -1.000
17 1.000 0.000 1.000
18 0.981 -0.195 -1.000
19 0.981 -0.195 1.000
20 0.924 -0.383 -1.000
21 0.924 -0.383 1.000
22 0.831 -0.556 -1.000
23 0.831 -0.556 1.000
Rows: 64 | Columns: 1
Modifier View Select Add Node
world_population
Select Column by Value
Value
Value
1975.460
Fraction
Value
Fraction
Clamp
Value
Ceiling
Integer
Ceiling
Float
Floor
Integer
Floor
Float
Value to String
String
Integer
Value
Value to String
String
Integer
Value
1 Named Attribute
Named Attribute
Attribute
Exists
Float
Name
1 Named Attribute
Named Attribute
Attribute
Exists
Float
Name
Mix
Result
Float
Clamp Factor
Factor
A
B
4.3.2


Blender Window
Wed 15. Jan 15:51
*world_population_simple - Blender 4.3.2
File Edit Render Window Help Layout Shading Geometry Nodes Scripting
Object Mode View Select Add Object
Global
Options
fps: 24
(44) Scene Collection | World
Scene Collection
Collection
Camera
Cylinder
ImageEarth
Plane
Source Text
Annual Population of Urban /
Sun
World
1968
Source: United Nations, Annual Population of Urban Agglomerations with 300.000 Inhabitants or More
Scene
Format
Resolution X 2448 px
Y 1080 px
% 100%
Aspect X 1.000
Y 1.000
Render Region
Crop to Render Region
Frame Rate 24 fps
Frame Range
Frame Start 1
End 208
Step 1
Time Stretching
Stereoscopy
Output
/Users/jan-hendrik/Desktop/hi
Saving File Extensions
Cache Result
File Format PNG
Playback Keying View Marker
44 Start 1 End 208
Summary
World
world_population
Action
world_populationAction
Manual Frame Range
Cyclic Animation
Select Rotate View Object
Anim Player
4.3.2


Blender Window
Wed 15. Jan 15:44
*world_population_simple - Blender 4.3.2
File Edit Render Window Help Layout Shading Geometry Nodes Scripting
Object Mode View Select Add Object
Global
Options
User Perspective
(59) Scene Collection | Cylinder
2025
Source: United Nations, Annual Population of Urban Agglomerations with 300,000 Inhabitants or More
Scene
ViewLayer
Search
Scene Collection
Collection
Camera
Cylinder
ImageEarth
Source Text
Sun
World
Object View Select Add Node Use Nodes
Slot 1
bars
Cylinder > Cylinder > bars
Texture Coordinate
Generated
Normal
UV
Object
Camera
Window
Reflection
Object: Cylinder
From Instance
Separate XYZ
Vector
Color Ramp
Color
Alpha
HSL
Pos 0.000
Fac
Principled BSDF
BSDF
Base Color
Subsurface
Specular
Material Output
All
Surface
Volume
Displacement
Thickness
Format
Frame Range
Stereoscopy
Output
/Users/jan-hendrik/Desktop/hi
Saving
File Extensions
Cache Result
File Format
PNG
Color
BW
RGB
RGBA
Color Depth
8
16
Compression
15%
Image Sequence
Overwrite
Placeholders
Color Management
Follow Scene
Override
Display Device
Display P3
View
Standard
Look
None
Exposure
0.000
Gamma
1.000
Select
Pan View
Node
4.3.2


1993
Source: United Nations, Annual Population of Urban Agglomerations with 300,000 Inhabitants or More
ImageE
Modifier
View
Select
Add
Node
world_population
world_population
X and Y coordinates
Set Position
Geometry
Selection
Position
Offset
Named Attribute
Attribute
Exists
Float
Longitude
Named Attribute
Attribute
Exists
Float
Latitude
Combine XYZ
Vector
X
Y
Z
0.000
Instance on Points
Instances
Points
Selection
Instance
Pick Instance
Instance Index
Rotation
X
Y
Z
Scale
Set Material
Geometry
Selection
Multiply
Value
Multiply
Clamp
Value
Multiply
Value
Multiply
Clamp
Value
Combine XYZ
Vector
X
1.000
Y
1.000
Z

string_to_label.ipynb

blender

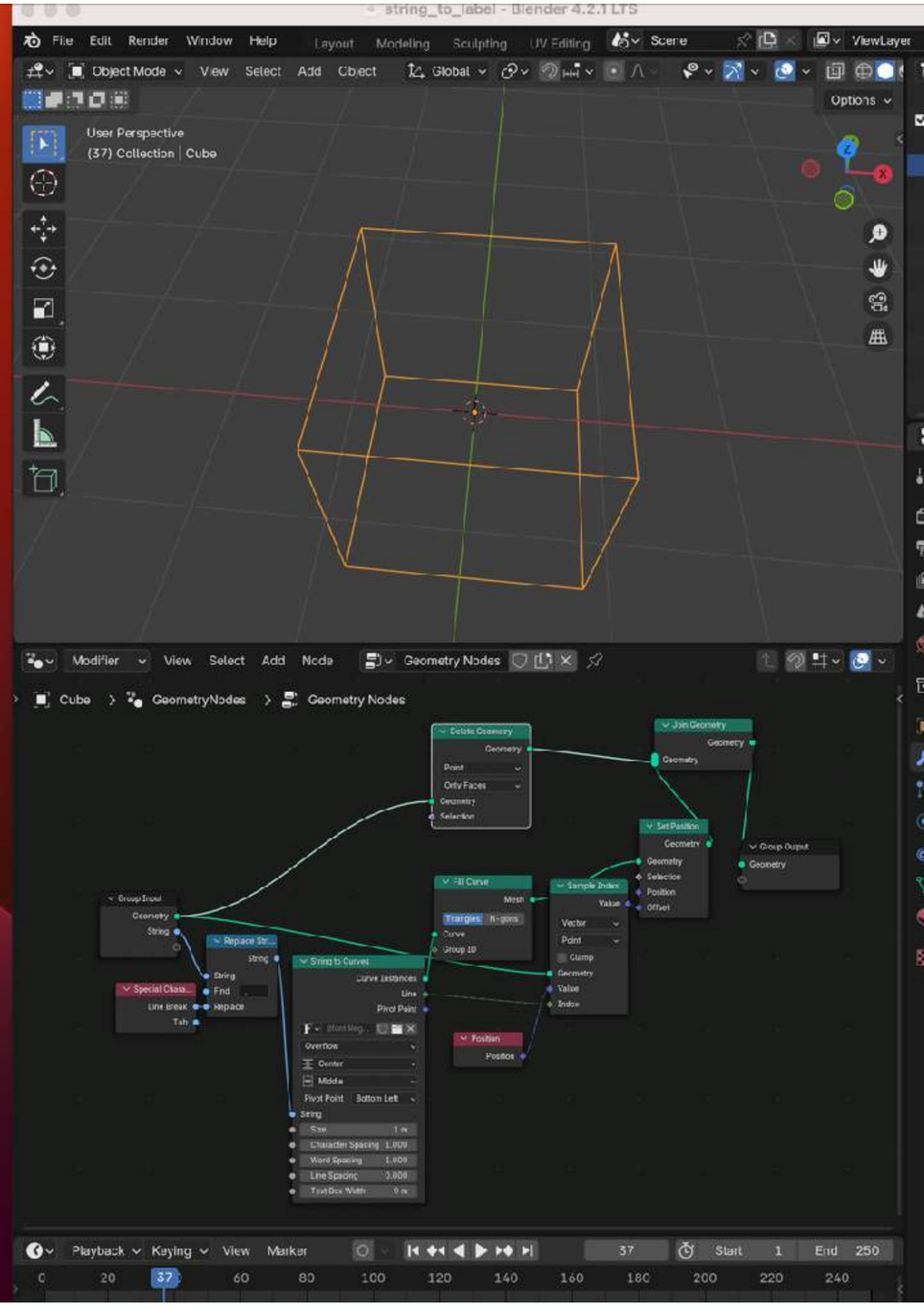
```
import bpy
blend_file_path = "string_to_label.blend"
bpy.ops.wm.open_mainfile(filepath=blend_file_path)
```

```
def set_labels(value_list):
    # Get the object and its Geometry Nodes modifier
    obj = bpy.data.objects.get("Cube")
    geo_modifier = obj.modifiers.get("GeometryNodes")
    new_lst = ','.join(value_list)
    print(new_lst)
    geo_modifier["Socket_2"] = new_lst
    obj.data.update()

# Example call with the list of values
set_labels([" "])
```

```
set_labels(["A", "B", "C"])
```

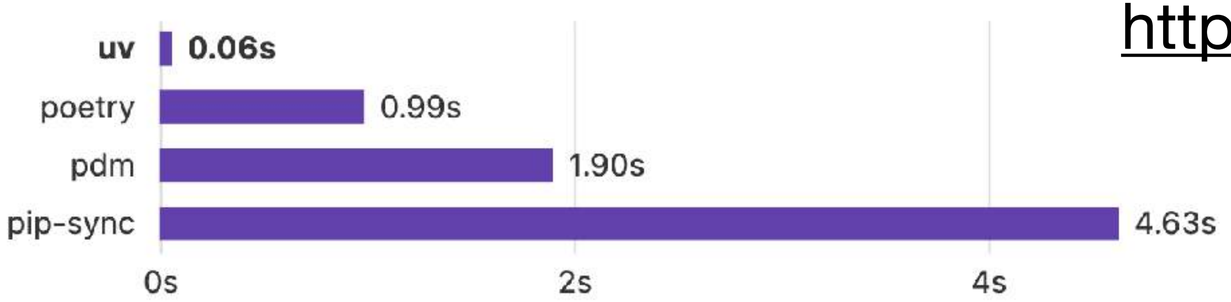
```
set_labels(["Zero","One","Two","Three","Four","Five","Six","Seven"])
```

```
set_labels(["A", "B", "C", "D", "E", "F", "G", "H"])
```



Package manager UV

An extremely fast Python package and project manager, written in Rust.



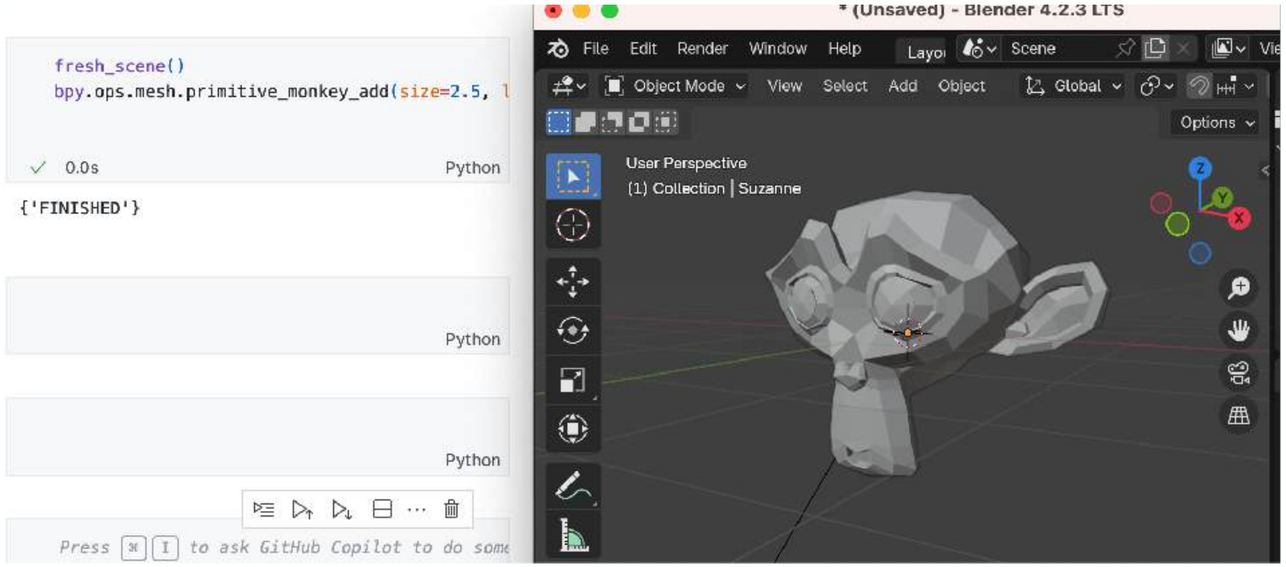
https://docs.astral.sh/uv/

Install with

```
curl -LsSf https://astral.sh/uv/install.sh | sh
```

GUI mode

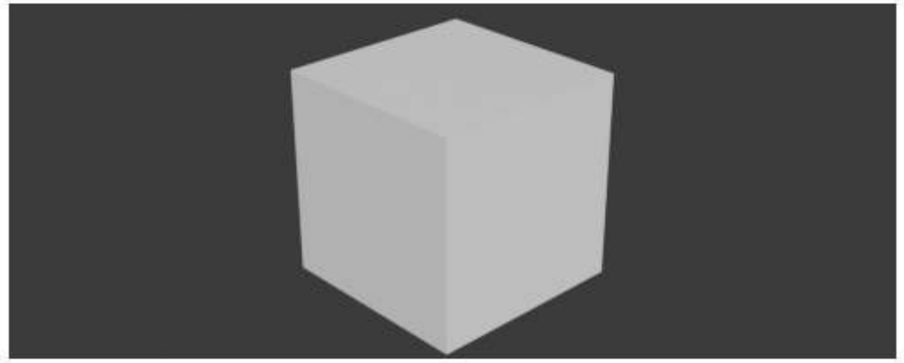
Blender GUI at the side



Headless mode

Without Blender GUI

```
fresh_scene()
bpy.ops.mesh.primitive_cube_add(size=2, location=(0, 0, 0))
render_result()
```



https://pypi.org/project/bpy/

Two options:

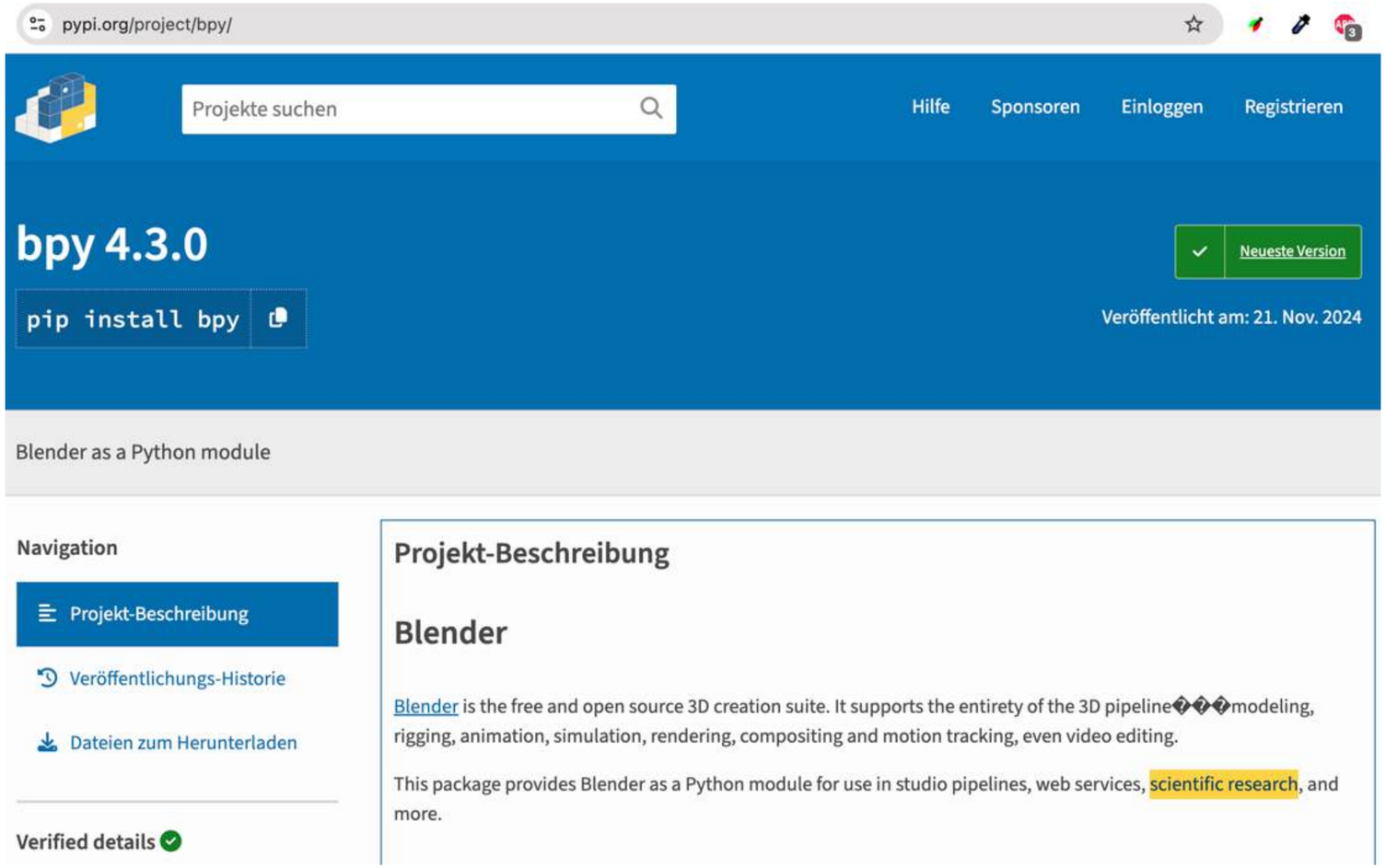
★ Blender starts Jupyter

https://github.com/Octoframes/bpy_jupyter

Jupyter starts Blender

https://github.com/Octoframes/blender_notebook

Headless mode



Headless mode

Python Enhancement Proposals | Python » PEP Index » PEP 723

Contents

- Abstract
- Motivation
- Rationale
- Specification
  - script type
  - Example
- Reference Implementation
- Backwards Compatibility
- Security Implications
- How to Teach This
- Recommendations
- Tooling buy-in
- Rejected Ideas
  - Why not use a comment block resembling requirements.txt?
  - Why not use a multi-line string?
  - Why not reuse core metadata fields?

PEP 723 – Inline script metadata

Author: Ofek Lev <ofekmeister at gmail.com>
Sponsor: Adam Turner <python at quite.org.uk>
PEP-Delegate: Brett Cannon <brett at python.org>
Discussions-To: Discourse thread
Status: Final
Type: Standards Track
Topic: Packaging
Created: 04-Aug-2023
Post-History: 04-Aug-2023, 06-Aug-2023, 23-Aug-2023, 06-Dec-2023
Replaces: 722
Resolution: 08-Jan-2024

uv run blen.py

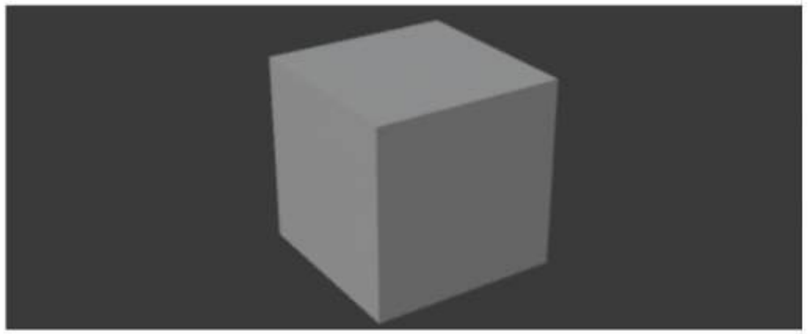
blen.py 1 | Open as a Jupyter Notebook

blen.py

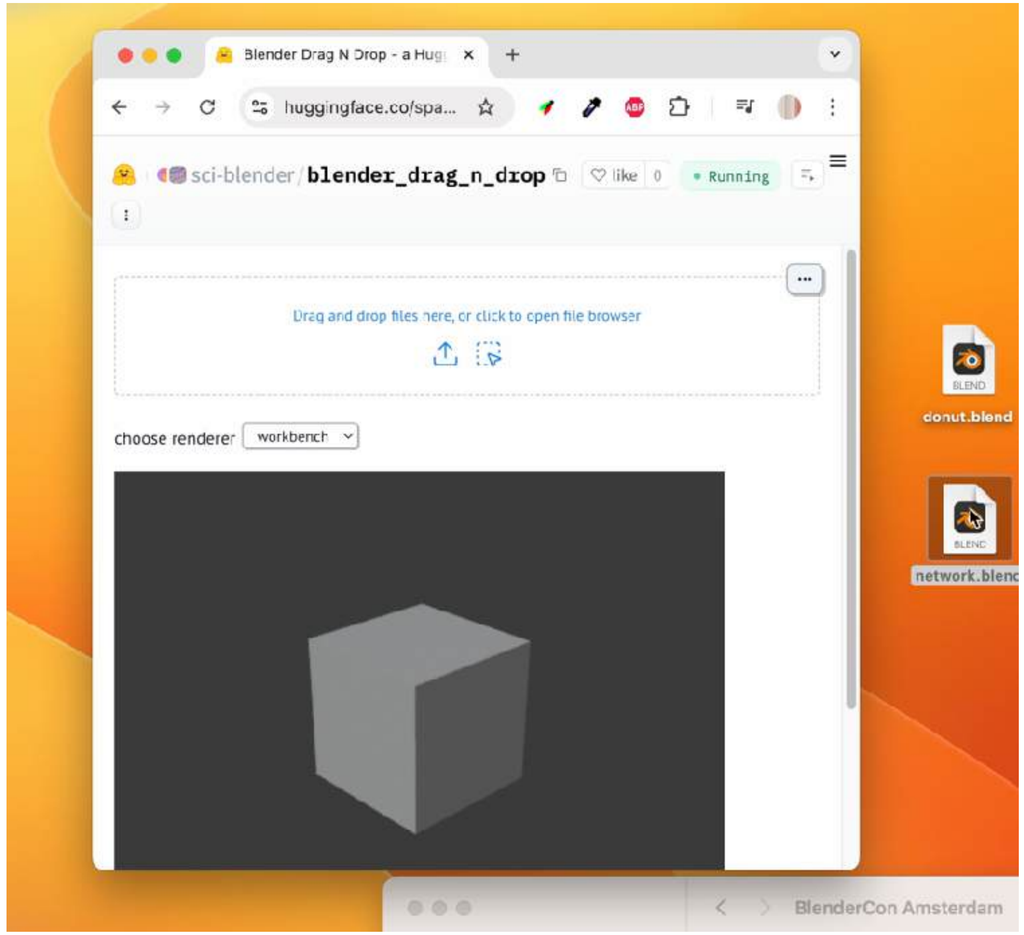
```python
# /// script
# requires-python = ">=3.11, <3.12"
# dependencies = [
#     "bpy==4.2.0",
# ]
# ///
import bpy, random, time

random.seed(time.time())

bpy.context.scene.render.engine = 'BLENDER_WORKBENCH'
bpy.context.scene.render.resolution_x = 500
bpy.context.scene.render.resolution_y = 200

bpy.data.objects['Cube'].rotation_euler[2] = random.uniform(0, 360)

bpy.ops.render.render()
bpy.data.images["Render Result"].save_render(filepath="test.png")
```

test.png

test.png



Host notebooks online (hugging face)

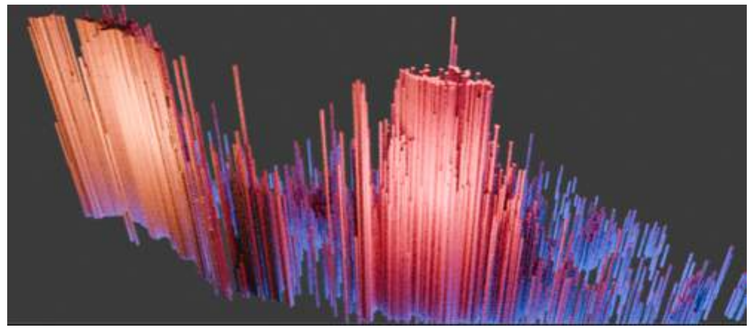


https://marimo.io/

https://huggingface.co/spaces/sci-blender/blender_drag_n_drop

Takeaway

- Setup the connection
- Interact with objects
- Data processing pipeline
- Time series
- Links to resources



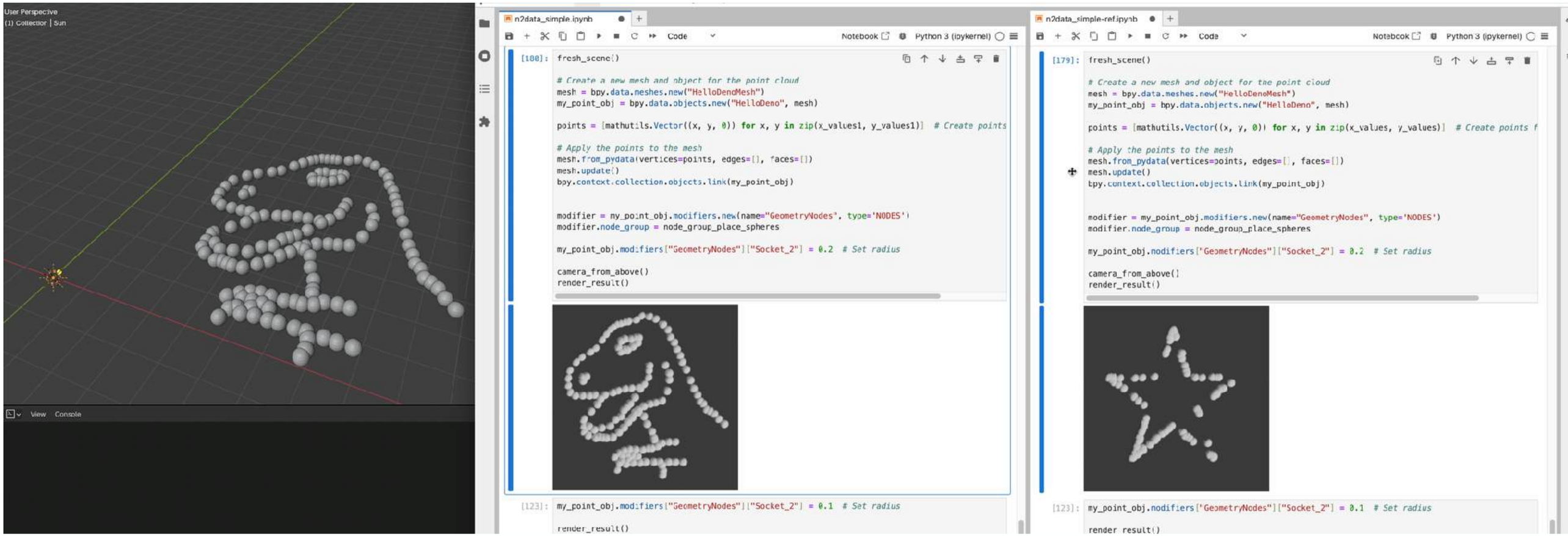
Thanks for listening!

- Come join the **Science Blender Discord**: https://discord.gg/k2Gd2cb4Kk
- Feel free to reach out after the talk
  (I’m around till Monday)
- Example Gallery (work in progress):
  https://kolibril13.github.io/bpy-gallery/n1objects/
- Interested in a **one day workshop?**

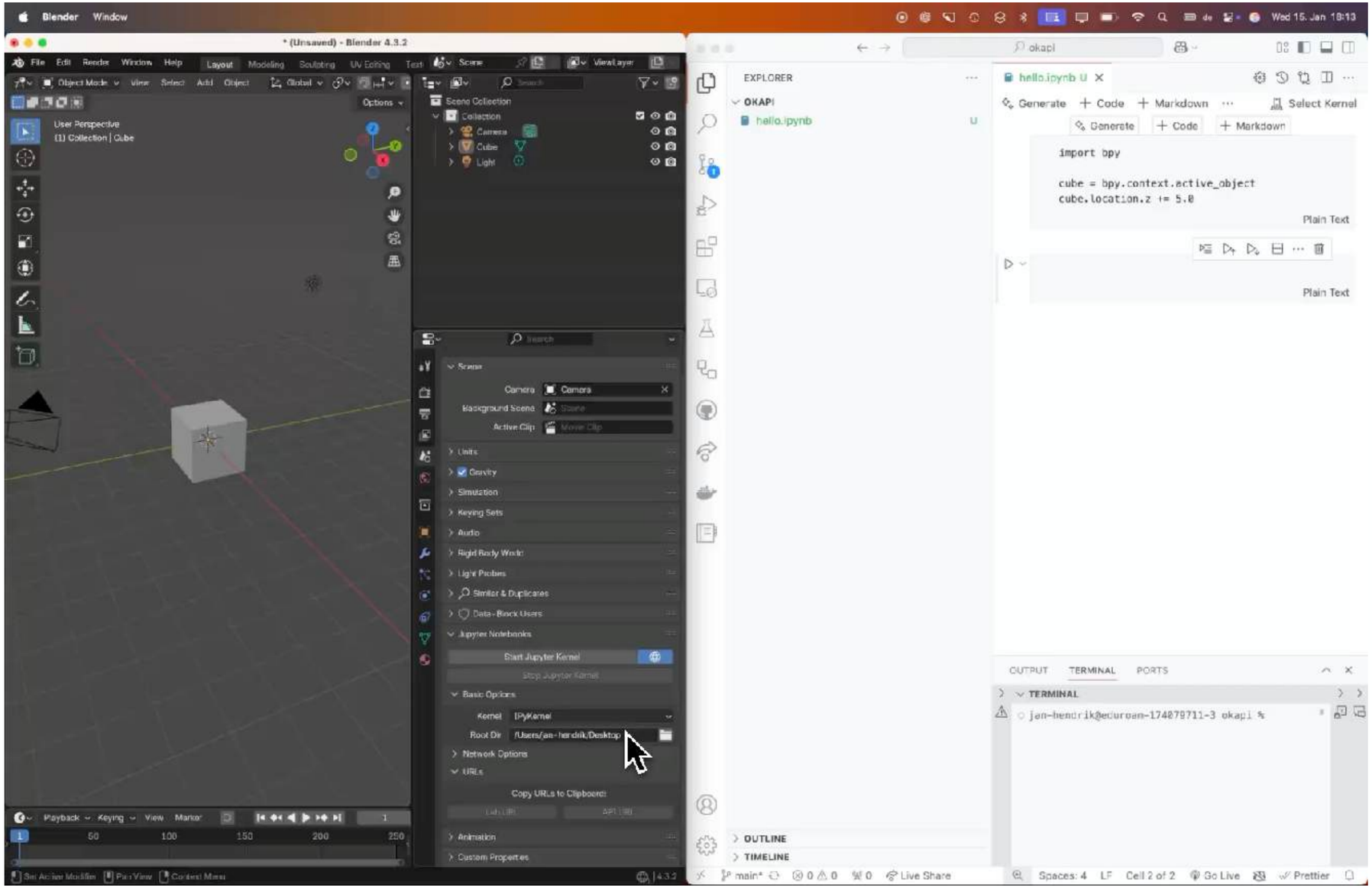
https://www.linkedin.com/in/jan-hendrik-müller-765014209/
https://bsky.app/profile/kolibril13.bsky.social

Backup slides

Two notebooks at the same time

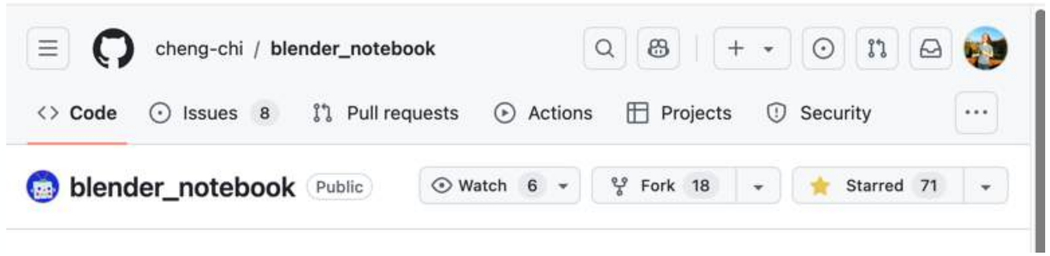


Connecting to VS Code



Install GUI-mode

```
uvx blender_notebook install --blender-exec="/Applications/Blender.app/Contents/MacOS/Blender"

uvx --python 3.11 jupyter lab

# … and select Kernel
```



Full instructions at

https://kolibril13.github.io/bpy-gallery/n0getting_started/