In [ ]:

```
import open3d as o3d
import numpy as np
import copy

""" Opening and viewing an .obj file """
obj_data = "human_nobg.obj"
mesh = o3d.io.read_triangle_mesh(obj_data)
```

In [ ]:

```
""" Visualizing point cloud """
mesh.compute_vertex_normals()
vertices = np.asarray(mesh.vertices)
o3d.visualization.draw_geometries([mesh])
```

In [ ]:

```
""" Visualizing point cloud """
mesh.compute_vertex_normals()
vertices = np.asarray(mesh.vertices)

""" connected components """
print("Generate data")
# mesh = o3dtut.get_bunny_mesh().subdivide_midpoint(number_of_iterations=2)
vert = vertices
min_vert, max_vert = vert.min(axis=0), vert.max(axis=0)
for _ in range(30):
    cube = o3d.geometry.TriangleMesh.create_box()
    cube.scale(0.005, center=cube.get_center())
    cube.translate(
        (
            np.random.uniform(min_vert[0], max_vert[0]),
            np.random.uniform(min_vert[1], max_vert[1]),
            np.random.uniform(min_vert[2], max_vert[2]),
        ),
        relative=False,
    )
    mesh += cube
mesh.compute_vertex_normals()
print("Show input mesh")
o3d.visualization.draw_geometries([mesh])
```

In [ ]:

```
""" Cluster connected triangles """
with o3d.utility.VerbosityContextManager(
        o3d.utility.VerbosityLevel.Debug) as cm:
            triangle_clusters, cluster_n_triangles, cluster_area = (
                mesh.cluster_connected_triangles())
triangle_clusters = np.asarray(triangle_clusters)
cluster_n_triangles = np.asarray(cluster_n_triangles)
```

In [ ]:

```
""" Show largest cluster """
mesh_1 = copy.deepcopy(mesh)
largest_cluster_idx = cluster_n_triangles.argmax()
triangles_to_remove = triangle_clusters != largest_cluster_idx
mesh_1.remove_triangles_by_mask(triangles_to_remove)
o3d.visualization.draw_geometries([mesh_1])
```

In [ ]:

```
o3d.io.write_triangle_mesh("human_clean.obj", mesh_1, write_triangle_uvs=True)
```

```
[Open3D WARNING] Write OBJ can not include triangle normals.
```