

```

import maya.cmds as cmds

PATH = "E:/160/Project2/V160_Proj02_DataViz_GA/"
FILE = "countypres_2000-2020.csv"
MB_PATH = "E:/160/Project2/V160_Proj02_DataViz_GA/GA_County_geom_v01.mb"
full_path = PATH + FILE

cmds.file(MB_PATH, open=True, force=True)

def create_data(full_path):
    county_values = {}
    largest_total_votes = 1000000
    smallest_total_votes = 0

    with open(full_path) as f:
        f.readline() # trash

        for current_line in f:

            if "2000,GEORGIA" in current_line:
                current_county_list = current_line.split(",") #creates a list
split every time there is a comma
                county = current_county_list[3]
                party = current_county_list[6]
                party_vote = int(current_county_list[7]) #turn a string into an
int
                total_vote = int(current_county_list[8])
                largest_total_votes = max(largest_total_votes, total_vote)
#assigns the larger of the two numbers to largest_total_votes
                smallest_total_votes = min(smallest_total_votes, total_vote)
#assigns the smaller of the two numbers to smallest_total_votes

                if county not in county_values: #checks to see if the current
county has been initialized
                    county_values[county] = {"thickness": 0, "dem_ratio": 0,
"rep_ratio": 0} #creates nested dictionaries for the current county with 3 keys

                    if party == "DEMOCRAT":

```

```

        county_values[county]["dem_ratio"] = party_vote / total_vote
#county values for x county in dem_ratio key is equal to the number of people who
voted for a party in a county divided by the total number of voters in that
county

        if party == "REPUBLICAN":
            county_values[county]["rep_ratio"] = party_vote / total_vote

        county_values[county]["thickness"] = (total_vote - 832) * 19.5 /
(263212 - 832) +0.5 #assigns the key thickness to the normalized value of the
total vote for that county compared to the largest number of votes in the year
2000 for a county (263212) and the lowest (832)

    return county_values #returns the dictionary

def perform_tasks(county_values):

    for county, values in county_values.items(): #goes through the dictionary
        county_select = "GA_" + county.title().replace(" ", "_")
        cmds.select(county_select, r=True) #selects the county
        curr = cmds.ls(selection=True)
        cmds.polyExtrudeFacet(tk=values["thickness"]) #extrudes with the counties
normalized value
        cmds.select("*Shape", deselect=True)
        obj_shd = cmds.shadingNode('blinn',n=curr[0]+"_shd", asShader=True)
        cmds.setAttr(obj_shd+".color", values["rep_ratio"],
values["dem_ratio"],0.5)
        cmds.select(curr)
        cmds.hyperShade(assign=obj_shd)

def main():
    county_values = create_data(full_path) #assigns county_values the dictionary
including relevant information
    perform_tasks(county_values) #creates nodes using county_values dictionary

main()

```