Part 2

September 17, 2021

1 Part 2

[1]:
```
# Libraries, options
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

pd.set_option('max_colwidth', 16)
pd.set_option("display.precision", 3)
```

2 County Data

[2]:
```
# Import data
data = pd.read_excel("County_Rural_Lookup_Candidate Assessment Part II.xlsx",
                     header = 3)
```

[3]:
```
# Get rid of footer, extra columns
data = data.iloc[:3142, :8]
```

[4]:
```
# Get rid of note column
data = data.drop(columns = ["Note"])
```

[5]:
```
# Rename columns
data.columns = ["geoid", "state", "county", "pop", "urban_pop", "rural_pop",␣
 ↪"pct_rural"]
```

[6]:
```
# Generate urban, rural, completely rural labels
data.loc[data.pct_rural<50, "category"] = "urban"
data.loc[(data.pct_rural>50) & (data.pct_rural<100), "category"] = "rural"
data.loc[data.pct_rural==100, "category"] = "all_rural"
```

[7]:
```
# Take a peek at the data
data.head()
```

[7]:

	geoid	state	county	pop	urban_pop	rural_pop	pct_rural	category
0	01001	AL	Autauga Coun...	54571.0	31650.0	22921.0	42.002	urban
1	01003	AL	Baldwin Coun...	182265.0	105205.0	77060.0	42.279	urban
2	01005	AL	Barbour Coun...	27457.0	8844.0	18613.0	67.790	rural
3	01007	AL	Bibb County,...	22915.0	7252.0	15663.0	68.353	rural
4	01009	AL	Blount Count...	57322.0	5760.0	51562.0	89.952	rural

3 Broadband Data

[8]:
```
# Import data
data1 = pd.read_csv("broadband_data_2020October.csv",
                    na_values = " -   ",
                    header = 18)
```

[9]:
```
# Rename columns
data1.columns = ["state", "geoid", "county", "fcc_bb", "bb"]
```

[10]:
```
# Add leading 0 to geoid's < 10k so that they match format of first dataset
data1["geoid"] = data1["geoid"].apply(lambda x: "{0:0>5}".format(x))
```

[11]:
```
# Take a peek at the data
data1.head()
```

[11]:

	state	geoid	county	fcc_bb	bb
0	AL	01001	Autauga County	0.806	0.391
1	AL	01003	Baldwin County	0.836	0.452
2	AL	01005	Barbour County	0.689	0.324
3	AL	01007	Bibb County	0.337	0.136
4	AL	01009	Blount County	0.758	0.199

4 Combined Data

[12]:
```
# Drop redundant columns from broadband data
data1 = data1.drop(columns = ["state", "county"])
```

[13]:
```
# Merge data
data = data.merge(data1, how = "inner", on = "geoid")
```

[14]:
```
# Make sure everything went okay
data.head()
```

[14]:

	geoid	state	county	pop	urban_pop	rural_pop	pct_rural	\
0	01001	AL	Autauga Coun...	54571.0	31650.0	22921.0	42.002	
1	01003	AL	Baldwin Coun...	182265.0	105205.0	77060.0	42.279	
2	01005	AL	Barbour Coun...	27457.0	8844.0	18613.0	67.790	
3	01007	AL	Bibb County,...	22915.0	7252.0	15663.0	68.353	
4	01009	AL	Blount Count...	57322.0	5760.0	51562.0	89.952	

	category	fcc_bb	bb
0	urban	0.806	0.391
1	urban	0.836	0.452
2	rural	0.689	0.324
3	rural	0.337	0.136
4	rural	0.758	0.199

5 Weighting

- For each category, I'm first going to generate a varible that equals the product of the county population and its bandwidth variables
- Then, I'm going to group the data by state/category and take the sum of each variable
- Finally, I'm going to generate a new variable equal to the quotient of the sum of the population-weighted bandwidth data and the sum of the population

[15]:
```
# Create values for pop * broadband levels
data["pop_bb"] = data["pop"] * data["bb"]
data["pop_fcc_bb"] = data["pop"] * data["fcc_bb"]
```

[16]:
```
# Group data by State/Category and take sums across categories
grouped_data = data.groupby(by = ["state", "category"]).sum()
```

[17]:
```
# Created weighted broadband variables
grouped_data["w_bb"] = grouped_data["pop_bb"] / grouped_data["pop"]
grouped_data["w_fcc_bb"] = grouped_data["pop_fcc_bb"] / grouped_data["pop"]
```

[18]:
```
# Drop extra variables, reset index
grouped_data = grouped_data[["w_bb", "w_fcc_bb"]]
grouped_data = grouped_data.reset_index()
```

[19]:
```
# Change data to wide format
x1 = grouped_data.pivot(index="state", columns="category", values="w_bb").
 ↪add_prefix("bb_")
x2 = grouped_data.pivot(index="state", columns="category", values="w_fcc_bb").
 ↪add_prefix("fcc_")
x1 = x1.merge(x2, on = "state", how = "inner").reset_index()
```

[20]:
```
# Repeat process to get overall weighted averages for each state, ignoring␣
 ↪county type
group = data.groupby("state").sum()[["pop_fcc_bb", "pop_bb", "pop"]].
 ↪reset_index()
group["bb"] = group["pop_bb"] / group["pop"]
group["fcc"] = group["pop_fcc_bb"] / group["pop"]
group = group[["state", "bb", "fcc"]]
x1 = x1.merge(group, on = "state", how = "inner").reset_index()
x1 = x1[["state", "bb", "bb_all_rural", "bb_rural", "bb_urban",
        "fcc", "fcc_all_rural", "fcc_rural", "fcc_urban"]]
```

[21]:
```
# Make sure everything went okay
x1.head()
```

[21]:

	state	bb	bb_all_rural	bb_rural	bb_urban	fcc	fcc_all_rural	fcc_rural	fcc_urban
0	AK	0.547	0.200	0.346	0.679	0.853	0.334	0.751	0.970
1	AL	0.456	0.087	0.325	0.535	0.872	0.378	0.793	0.935
2	AR	0.382	0.123	0.258	0.478	0.798	0.455	0.656	0.912
3	AZ	0.678	NaN	0.267	0.691	0.945	NaN	0.448	0.960
4	CA	0.721	0.129	0.260	0.725	0.985	0.564	0.922	0.986

6 Variable Convention

- bb refers to the microsoft measure of broadband usage
  - bb is the population-weighted average for the entire state
  - the other bb_ variables are the population-weighted average variables for the specific county types
- fcc refers to the FCC measure of broadband availability
  - the prefixes follow the same convention as the bb variables

[22]:
```
# Save to csv
x1.to_csv("part_2.csv")
```

[23]:
```
# Some summary statistics
x1.describe().transpose()[["mean", "min", "max", "std"]]
```

[23]:

	mean	min	max	std
bb	0.601	0.311	0.805	0.118
bb_all_rural	0.231	0.075	0.637	0.108

bb_rural	0.362	0.110	0.652	0.114
bb_urban	0.654	0.410	0.839	0.095
fcc	0.940	0.796	0.992	0.049
fcc_all_rural	0.684	0.198	0.960	0.193
fcc_rural	0.821	0.204	0.991	0.144
fcc_urban	0.967	0.890	0.998	0.024

7 Caveats/Concerns

- There were some missing values in the broadband data (encoded to " - ")
  - 11 missing broadband values from Virginia, Nebraska, and Texas
  - 9 missing FCC broadband values from Alaska and Oregon
- In the final dataset, there were a fair amount of missing values
  - This is likely because many states simply don't have 100% rural areas, but it would be good to double check this
- To merge the datasets, I had to fix the geoid variable
  - If I had more time, I would probably merge by using state/county combinations to make sure they produce the same results
- There may be other weighting approaches that yield better results
  - In particular, it would be possible to weight the broadband usage by percent urban/rural in each county
  - A drawback of my current approach is that it chalks up all of the usage in a 51% urban county to urban users
  - On the other hand, this would assume that urban/rural people have similar rates for broadband usage, which probably isn't true