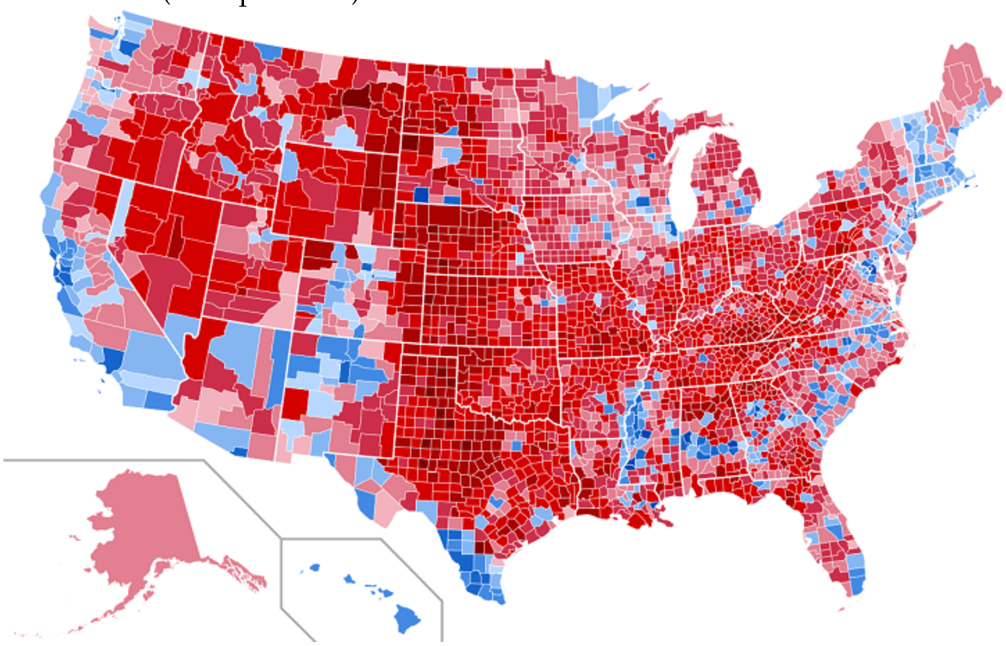
12.3 Choropleths

May 9, 2019

1 12.3 Choropleths

A **choropleth** is a map in which areas are colored according to some statistic of interest. Perhaps the most familiar example of a choropleth is the presidential election map, which shows the percentage in each county who voted for the Democratic or Republican candidate. In this graphic, the observational units are counties, and the statistic of interest is the percentage who voted for the Democratic (or Republican) candidate.



In this notebook, you will learn how to make choropleths like the one above.

1.1 Shapefiles

The shapefile format is a data format for geometric objects, such as points, lines, and polygons. A shapefile can be used to describe the boundaries of a lake, the course of a river, or the boundaries of a county.

You can find shapefiles for most geographic entities online. For example, the U.S. Census Bureau maintains shapefiles for boundaries of states, counties, and congressional districts in the United States. Shapefiles for the countries of the world can be found at this website.

I downloaded the shapefiles for U.S. counties from the Census Bureau website and uploaded them to JupyterHub. You can find them in the `/data301/data/cb_2017_us_county_5m/` directory.

In [1]: !ls /data301/data/cb_2017_us_county_5m/

```
cb_2017_us_county_5m.cpg             cb_2017_us_county_5m.shp.iso.xml
cb_2017_us_county_5m.dbf             cb_2017_us_county_5m.shp.xml
cb_2017_us_county_5m.prj             cb_2017_us_county_5m.shx
cb_2017_us_county_5m.shp             cb_2017_us_county_5m.zip
cb_2017_us_county_5m.shp.ea.iso.xml
```

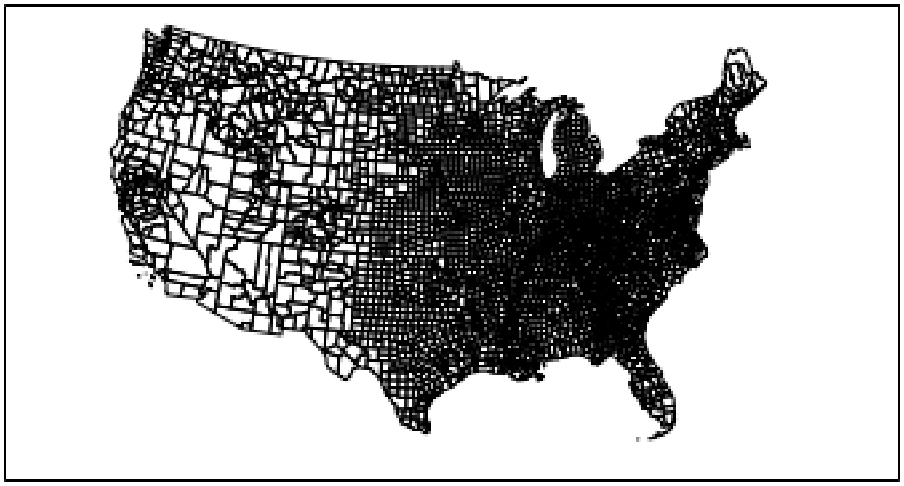
Notice that "shapefile" is somewhat of a misnomer, as the format refers not to a single file but a collection of files. The main files are:

- `.shp` - shape format, which stores the geometric objects
- `.shx` - shape index format, which indexes the objects to make them quickly searchable
- `.dbf` - attribute format, which stores additional metadata about each object
- `.prj` - projection format

To read in a shapefile using Basemap, we first set up the map, then call the `.readshapefile()` method, which takes two arguments: (1) the stem of the shapefiles (without the file extension) and (2) a name for the field that will store the attributes (you can pick any name you like, but try to be descriptive).

In [2]:
```
import matplotlib.pyplot as plt
%matplotlib inline

import cartopy.crs as ccrs
from cartopy.io.shapereader import Reader
from cartopy.feature import ShapelyFeature

ax = plt.axes(
    projection=ccrs.LambertConformal(
        central_latitude=39,
        central_longitude=-96,
        standard_parallels=(33, 45)
    )
)
ax.set_extent([-125, -66.5, 20, 50])

# Read in county-level shapefiles
fname = "/data301/data/cb_2017_us_county_5m/cb_2017_us_county_5m"
shp = Reader(fname)

# Add each county to the data set.
ax.add_geometries(shp.geometries(),
                  ccrs.PlateCarree(),
                  facecolor="None",
                  edgecolor='black')
```

Out[2]: `<cartopy.mpl.feature_artist.FeatureArtist at 0x7fb276c283c8>`



To make a choropleth, we simply need to set the `facecolor` of each geometry. First, let's read in some county-level data that we can plot.

```
In [3]: import pandas as pd
        election_df = pd.read_csv("https://raw.githubusercontent.com/dlsun/data-science-book/"
                                  "master/data/election2016.csv")
        election_df
```

```
Out[3]:     votes_dem  votes_gop  total_votes   per_dem   per_gop    diff  \
        0     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        1     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        2     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        3     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        4     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        5     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        6     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        7     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        8     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        9     93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        10    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        11    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        12    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        13    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        14    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        15    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        16    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        17    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        18    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        19    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        20    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
        21    93003.0   130413.0     246588.0  0.377159  0.528870  37,410
```

22	93003.0	130413.0	246588.0	0.377159	0.528870	37,410
23	93003.0	130413.0	246588.0	0.377159	0.528870	37,410
24	93003.0	130413.0	246588.0	0.377159	0.528870	37,410
25	93003.0	130413.0	246588.0	0.377159	0.528870	37,410
26	93003.0	130413.0	246588.0	0.377159	0.528870	37,410
27	93003.0	130413.0	246588.0	0.377159	0.528870	37,410
28	93003.0	130413.0	246588.0	0.377159	0.528870	37,410
29	5908.0	18110.0	24661.0	0.239569	0.734358	12,202
...	...	...	...	...	...	...
3111	1763.0	6971.0	9180.0	0.192048	0.759368	5,208
3112	3334.0	11077.0	15080.0	0.221088	0.734549	7,743
3113	554.0	2284.0	2955.0	0.187479	0.772927	1,730
3114	1348.0	4461.0	6191.0	0.217735	0.720562	3,113
3115	383.0	1903.0	2410.0	0.158921	0.789627	1,520
3116	8327.0	25168.0	35256.0	0.236187	0.713864	16,841
3117	1061.0	6527.0	7809.0	0.135869	0.835830	5,466
3118	6888.0	7601.0	16420.0	0.419488	0.462911	713
3119	594.0	4067.0	5079.0	0.116952	0.800748	3,473
3120	1324.0	15778.0	17935.0	0.073822	0.879732	14,454
3121	1279.0	4409.0	6200.0	0.206290	0.711129	3,130
3122	668.0	5520.0	6552.0	0.101954	0.842491	4,852
3123	271.0	3347.0	3771.0	0.071864	0.887563	3,076
3124	4200.0	11167.0	16543.0	0.253884	0.675029	6,967
3125	924.0	4418.0	5708.0	0.161878	0.774001	3,494
3126	400.0	1939.0	2535.0	0.157791	0.764892	1,539
3127	638.0	3477.0	4349.0	0.146700	0.799494	2,839
3128	11572.0	24844.0	39945.0	0.289698	0.621955	13,272
3129	1105.0	6779.0	8398.0	0.131579	0.807216	5,674
3130	6573.0	23523.0	32493.0	0.202290	0.723941	16,950
3131	115.0	1116.0	1297.0	0.088666	0.860447	1,001
3132	2535.0	11115.0	14634.0	0.173227	0.759533	8,580
3133	719.0	3437.0	4460.0	0.161211	0.770628	2,718
3134	2926.0	10266.0	14187.0	0.206245	0.723620	7,340
3135	644.0	3409.0	4297.0	0.149872	0.793344	2,765
3136	3233.0	12153.0	16661.0	0.194046	0.729428	8,920
3137	7313.0	3920.0	12176.0	0.600608	0.321945	3,393
3138	1202.0	6154.0	8053.0	0.149261	0.764187	4,952
3139	532.0	2911.0	3715.0	0.143203	0.783580	2,379
3140	294.0	2898.0	3334.0	0.088182	0.869226	2,604

	per_point_diff	state_abbr	county_name	combined_fips
0	15.17%	AK	Alaska	2013
1	15.17%	AK	Alaska	2016
2	15.17%	AK	Alaska	2020
3	15.17%	AK	Alaska	2050
4	15.17%	AK	Alaska	2060
5	15.17%	AK	Alaska	2068
6	15.17%	AK	Alaska	2070

7	15.17%	AK	Alaska	2090
8	15.17%	AK	Alaska	2100
9	15.17%	AK	Alaska	2105
10	15.17%	AK	Alaska	2110
11	15.17%	AK	Alaska	2122
12	15.17%	AK	Alaska	2130
13	15.17%	AK	Alaska	2150
14	15.17%	AK	Alaska	2164
15	15.17%	AK	Alaska	2170
16	15.17%	AK	Alaska	2180
17	15.17%	AK	Alaska	2185
18	15.17%	AK	Alaska	2188
19	15.17%	AK	Alaska	2195
20	15.17%	AK	Alaska	2198
21	15.17%	AK	Alaska	2220
22	15.17%	AK	Alaska	2230
23	15.17%	AK	Alaska	2240
24	15.17%	AK	Alaska	2261
25	15.17%	AK	Alaska	2270
26	15.17%	AK	Alaska	2275
27	15.17%	AK	Alaska	2282
28	15.17%	AK	Alaska	2290
29	49.48%	AL	Autauga County	1001
...	...	...	...	...
3111	56.73%	WV	Upshur County	54097
3112	51.35%	WV	Wayne County	54099
3113	58.54%	WV	Webster County	54101
3114	50.28%	WV	Wetzel County	54103
3115	63.07%	WV	Wirt County	54105
3116	47.77%	WV	Wood County	54107
3117	70.00%	WV	Wyoming County	54109
3118	4.34%	WY	Albany County	56001
3119	68.38%	WY	Big Horn County	56003
3120	80.59%	WY	Campbell County	56005
3121	50.48%	WY	Carbon County	56007
3122	74.05%	WY	Converse County	56009
3123	81.57%	WY	Crook County	56011
3124	42.11%	WY	Fremont County	56013
3125	61.21%	WY	Goshen County	56015
3126	60.71%	WY	Hot Springs County	56017
3127	65.28%	WY	Johnson County	56019
3128	33.23%	WY	Laramie County	56021
3129	67.56%	WY	Lincoln County	56023
3130	52.17%	WY	Natrona County	56025
3131	77.18%	WY	Niobrara County	56027
3132	58.63%	WY	Park County	56029
3133	60.94%	WY	Platte County	56031
3134	51.74%	WY	Sheridan County	56033

```
3135          64.35%         WY     Sublette County          56035
3136          53.54%         WY   Sweetwater County          56037
3137          27.87%         WY        Teton County          56039
3138          61.49%         WY        Uinta County          56041
3139          64.04%         WY     Washakie County          56043
3140          78.10%         WY       Weston County          56045

[3141 rows x 10 columns]
```

We need to merge this data with the shapefile that we just loaded. We can create a `DataFrame` out of the `records` of a shapefile.

```
In [4]: shp_df = pd.DataFrame(
            [record.attributes for record in shp.records()]
        )
        shp_df
```

```
Out[4]:            AFFGEOID        ALAND      AWATER COUNTYFP  COUNTYNS  GEOID LSAD  \
        0      0500000US04015  34475503964   387344307      015  00025445  04015   06
        1      0500000US22105   2049488093   136678798      105  00559500  22105   15
        2      0500000US16063   3111451190    11606076      063  00395624  16063   06
        3      0500000US27119   5105067510    69169913      119  00659505  27119   06
        4      0500000US38017   4571107601     7732062      017  01034226  38017   06
        5      0500000US46081   2072127374      667509      081  01266996  46081   06
        6      0500000US36095   1610503572    11676949      095  00974145  36095   06
        7      0500000US02275   6619614514  2384746838      275  02516402  02275   03
        8      0500000US13143    730804590     2616530      143  00350637  13143   06
        9      0500000US13023    559100209     8447343      023  00347451  13023   06
        10     0500000US18093   1163367128     7122898      093  00451703  18093   06
        11     0500000US18079    975364386     4546211      079  00450364  18079   06
        12     0500000US26087   1668448345    48310232      087  01622986  26087   06
        13     0500000US28017   1299609263     6380767      017  00695733  28017   06
        14     0500000US39033   1040619677     2359515      033  01074029  39033   06
        15     0500000US46099   2089691696    18198505      099  01265772  46099   06
        16     0500000US46125   1598188427     1636386      125  01265770  46125   06
        17     0500000US48471   2031033548    44838548      471  01384021  48471   06
        18     0500000US72133     88144282   111140897      133  01804547  72133   13
        19     0500000US46003   1834813747    11201379      003  01266983  46003   06
        20     0500000US48047   2443277016      741627      047  01383809  48047   06
        21     0500000US72025    151782488     1201689      025  01804492  72025   13
        22     0500000US72033     12629288     5619250      033  01804496  72033   13
        23     0500000US72101    100676181      139537      101  01804531  72101   13
        24     0500000US31029   2316533465     7978174      029  00835836  31029   06
        25     0500000US72054     39391498       34012      054  01804507  72054   13
        26     0500000US08021   3334423639     9235291      021  00198126  08021   06
        27     0500000US24043   1185609331    24803926      043  01714220  24043   06
        28     0500000US20137   2274122723     8421464      137  00485032  20137   06
        29     0500000US17053   1257719718     1640286      053  00424228  17053   06
```

...	...	...	...	...	...	...	...
3203	0500000US33009	4425224129	106489725	009	00873178	33009	06
3204	0500000US55117	1324905935	1967592839	117	01581118	55117	06
3205	0500000US28129	1647926085	2599432	129	00695785	28129	06
3206	0500000US22103	2190165998	647368450	103	01629503	22103	15
3207	0500000US24033	1250087042	41892810	033	01714670	24033	06
3208	0500000US27145	3477880457	122121594	145	00659517	27145	06
3209	0500000US25017	2117830644	75814940	017	00606935	25017	06
3210	0500000US42055	2000052118	1544300	055	01213670	42055	06
3211	0500000US17145	1144186125	13416186	145	01784940	17145	06
3212	0500000US29073	1344215195	18009291	073	00758491	29073	06
3213	0500000US18161	417568563	10238515	161	00450395	18161	06
3214	0500000US29099	1700691208	19797687	099	00758504	29099	06
3215	0500000US40131	1749945924	92823523	131	01101853	40131	06
3216	0500000US45017	987179165	29350862	017	01247986	45017	06
3217	0500000US54035	1202677932	18836099	035	01559963	54035	06
3218	0500000US69100	85098741	539090851	100	01805246	69100	12
3219	0500000US72123	179660999	115910809	123	01804542	72123	13
3220	0500000US48239	2148246715	71039573	239	01383905	48239	06
3221	0500000US53035	1023306116	442189824	035	01529223	53035	06
3222	0500000US12057	2642602901	635590801	057	00295757	12057	06
3223	0500000US17043	848222200	22887251	043	00422191	17043	06
3224	0500000US34031	481744927	32079811	031	00882232	34031	06
3225	0500000US48181	2415995947	120112446	181	01383876	48181	06
3226	0500000US13283	516529395	7805621	283	00347505	13283	06
3227	0500000US28155	1090223315	5935288	155	00695798	28155	06
3228	0500000US31101	2749531887	124672175	101	00835872	31101	06
3229	0500000US28001	1197464269	65273640	001	00695726	28001	06
3230	0500000US36069	1668114449	47820988	069	00974133	36069	06
3231	0500000US54053	1115633278	36174540	053	01560254	54053	06
3232	0500000US04025	21039796372	11501575	025	00042809	04025	06

	NAME	STATEFP
0	Mohave	04
1	Tangipahoa	22
2	Lincoln	16
3	Polk	27
4	Cass	38
5	Lawrence	46
6	Schoharie	36
7	Wrangell	02
8	Haralson	13
9	Bleckley	13
10	Lawrence	18
11	Jennings	18
12	Lapeer	26
13	Chickasaw	28
14	Crawford	39

```
15          Minnehaha    46
16             Turner    46
17             Walker    48
18       Santa Isabel    72
19             Aurora    46
20             Brooks    48
21             Caguas    72
22             Cataño    72
23            Morovis    72
24              Chase    31
25            Florida    72
26            Conejos    08
27         Washington    24
28             Norton    20
29               Ford    17
...               ...   ...
3203          Grafton    33
3204        Sheboygan    55
3205            Smith    28
3206      St. Tammany    22
3207  Prince George's    24
3208          Stearns    27
3209        Middlesex    25
3210         Franklin    42
3211            Perry    17
3212        Gasconade    29
3213            Union    18
3214        Jefferson    29
3215           Rogers    40
3216          Calhoun    45
3217          Jackson    54
3218             Rota    69
3219          Salinas    72
3220          Jackson    48
3221           Kitsap    53
3222     Hillsborough    12
3223           DuPage    17
3224          Passaic    34
3225          Grayson    48
3226         Treutlen    13
3227          Webster    28
3228            Keith    31
3229            Adams    28
3230          Ontario    36
3231            Mason    54
3232          Yavapai    04

[3233 rows x 9 columns]
```

We will need to merge `election_df` with `shp_df`. But what do we merge the `DataFrames` on? It turns out that every county in the United States is assigned a unique ID called a FIPS code. The FIPS code appears in `election_df` as `combined_fips` and in `shp_df` as `GEOID`. Let's take a look at these columns.

```
In [5]: election_df.combined_fips

Out[5]: 0        2013
        1        2016
        2        2020
        3        2050
        4        2060
        5        2068
        6        2070
        7        2090
        8        2100
        9        2105
        10       2110
        11       2122
        12       2130
        13       2150
        14       2164
        15       2170
        16       2180
        17       2185
        18       2188
        19       2195
        20       2198
        21       2220
        22       2230
        23       2240
        24       2261
        25       2270
        26       2275
        27       2282
        28       2290
        29       1001
                 ...
        3111    54097
        3112    54099
        3113    54101
        3114    54103
        3115    54105
        3116    54107
        3117    54109
        3118    56001
        3119    56003
        3120    56005
```

```
3121    56007
3122    56009
3123    56011
3124    56013
3125    56015
3126    56017
3127    56019
3128    56021
3129    56023
3130    56025
3131    56027
3132    56029
3133    56031
3134    56033
3135    56035
3136    56037
3137    56039
3138    56041
3139    56043
3140    56045
Name: combined_fips, Length: 3141, dtype: int64
```

```
In [6]: shp_df.GEOID
```

```
Out[6]: 0       04015
        1       22105
        2       16063
        3       27119
        4       38017
        5       46081
        6       36095
        7       02275
        8       13143
        9       13023
        10      18093
        11      18079
        12      26087
        13      28017
        14      39033
        15      46099
        16      46125
        17      48471
        18      72133
        19      46003
        20      48047
        21      72025
        22      72033
        23      72101
```

```
24      31029
25      72054
26      08021
27      24043
28      20137
29      17053
        ...
3203    33009
3204    55117
3205    28129
3206    22103
3207    24033
3208    27145
3209    25017
3210    42055
3211    17145
3212    29073
3213    18161
3214    29099
3215    40131
3216    45017
3217    54035
3218    69100
3219    72123
3220    48239
3221    53035
3222    12057
3223    17043
3224    34031
3225    48181
3226    13283
3227    28155
3228    31101
3229    28001
3230    36069
3231    54053
3232    04025
Name: GEOID, Length: 3233, dtype: object
```

Notice that `shp_df` treats the FIPS code as a string (so every FIPS code is exactly 5 digits, with a leading zero if necessary). On the other hand, `election_df` treats the FIPS code as an integer. If we want to join the two, we will have to cast them to the same type. It is probably easier to convert the string to an integer than vice versa.

```
In [7]: shp_df["GEOID"] = shp_df["GEOID"].astype(int)
```

Now we are ready to merge the two `DataFrames`.

```
In [8]: all_data = shp_df.merge(election_df,
                                how="left",
```

```
                              left_on="GEOID", right_on="combined_fips")
all_data
```

Out[8]:

```
          AFFGEOID        ALAND      AWATER COUNTYFP  COUNTYNS  GEOID LSAD  \
0     0500000US04015  34475503964   387344307      015  00025445   4015   06
1     0500000US22105   2049488093   136678798      105  00559500  22105   15
2     0500000US16063   3111451190    11606076      063  00395624  16063   06
3     0500000US27119   5105067510    69169913      119  00659505  27119   06
4     0500000US38017   4571107601     7732062      017  01034226  38017   06
5     0500000US46081   2072127374      667509      081  01266996  46081   06
6     0500000US36095   1610503572    11676949      095  00974145  36095   06
7     0500000US02275   6619614514  2384746838      275  02516402   2275   03
8     0500000US13143    730804590     2616530      143  00350637  13143   06
9     0500000US13023    559100209     8447343      023  00347451  13023   06
10    0500000US18093   1163367128     7122898      093  00451703  18093   06
11    0500000US18079    975364386     4546211      079  00450364  18079   06
12    0500000US26087   1668448345    48310232      087  01622986  26087   06
13    0500000US28017   1299609263     6380767      017  00695733  28017   06
14    0500000US39033   1040619677     2359515      033  01074029  39033   06
15    0500000US46099   2089691696    18198505      099  01265772  46099   06
16    0500000US46125   1598188427     1636386      125  01265770  46125   06
17    0500000US48471   2031033548    44838548      471  01384021  48471   06
18    0500000US72133     88144282   111140897      133  01804547  72133   13
19    0500000US46003   1834813747    11201379      003  01266983  46003   06
20    0500000US48047   2443277016      741627      047  01383809  48047   06
21    0500000US72025    151782488     1201689      025  01804492  72025   13
22    0500000US72033     12629288     5619250      033  01804496  72033   13
23    0500000US72101    100676181      139537      101  01804531  72101   13
24    0500000US31029   2316533465     7978174      029  00835836  31029   06
25    0500000US72054     39391498       34012      054  01804507  72054   13
26    0500000US08021   3334423639     9235291      021  00198126   8021   06
27    0500000US24043   1185609331    24803926      043  01714220  24043   06
28    0500000US20137   2274122723     8421464      137  00485032  20137   06
29    0500000US17053   1257719718     1640286      053  00424228  17053   06
...              ...          ...         ...      ...       ...    ...  ...
3203  0500000US33009   4425224129   106489725      009  00873178  33009   06
3204  0500000US55117   1324905935  1967592839      117  01581118  55117   06
3205  0500000US28129   1647926085     2599432      129  00695785  28129   06
3206  0500000US22103   2190165998   647368450      103  01629503  22103   15
3207  0500000US24033   1250087042    41892810      033  01714670  24033   06
3208  0500000US27145   3477880457   122121594      145  00659517  27145   06
3209  0500000US25017   2117830644    75814940      017  00606935  25017   06
3210  0500000US42055   2000052118     1544300      055  01213670  42055   06
3211  0500000US17145   1144186125    13416186      145  01784940  17145   06
3212  0500000US29073   1344215195    18009291      073  00758491  29073   06
3213  0500000US18161    417568563    10238515      161  00450395  18161   06
3214  0500000US29099   1700691208    19797687      099  00758504  29099   06
3215  0500000US40131   1749945924    92823523      131  01101853  40131   06
```

3216	0500000US45017	987179165	29350862	017	01247986	45017	06
3217	0500000US54035	1202677932	18836099	035	01559963	54035	06
3218	0500000US69100	85098741	539090851	100	01805246	69100	12
3219	0500000US72123	179660999	115910809	123	01804542	72123	13
3220	0500000US48239	2148246715	71039573	239	01383905	48239	06
3221	0500000US53035	1023306116	442189824	035	01529223	53035	06
3222	0500000US12057	2642602901	635590801	057	00295757	12057	06
3223	0500000US17043	848222200	22887251	043	00422191	17043	06
3224	0500000US34031	481744927	32079811	031	00882232	34031	06
3225	0500000US48181	2415995947	120112446	181	01383876	48181	06
3226	0500000US13283	516529395	7805621	283	00347505	13283	06
3227	0500000US28155	1090223315	5935288	155	00695798	28155	06
3228	0500000US31101	2749531887	124672175	101	00835872	31101	06
3229	0500000US28001	1197464269	65273640	001	00695726	28001	06
3230	0500000US36069	1668114449	47820988	069	00974133	36069	06
3231	0500000US54053	1115633278	36174540	053	01560254	54053	06
3232	0500000US04025	21039796372	11501575	025	00042809	4025	06

	NAME	STATEFP	votes_dem	votes_gop	total_votes	per_dem	\
0	Mohave	04	16485.0	54656.0	74189.0	0.222203	
1	Tangipahoa	22	16869.0	33933.0	52381.0	0.322044	
2	Lincoln	16	360.0	1184.0	1748.0	0.205950	
3	Polk	27	4712.0	8979.0	14698.0	0.320588	
4	Cass	38	31291.0	39738.0	78807.0	0.397059	
5	Lawrence	46	3356.0	7411.0	11842.0	0.283398	
6	Schoharie	36	3777.0	8173.0	12673.0	0.298035	
7	Wrangell	02	93003.0	130413.0	246588.0	0.377159	
8	Haralson	13	1474.0	9579.0	11317.0	0.130247	
9	Bleckley	13	1094.0	3717.0	4934.0	0.221727	
10	Lawrence	18	4210.0	14034.0	19076.0	0.220696	
11	Jennings	18	2364.0	8222.0	11115.0	0.212686	
12	Lapeer	26	12734.0	30037.0	45126.0	0.282188	
13	Chickasaw	28	3649.0	4127.0	7890.0	0.462484	
14	Crawford	39	4518.0	13265.0	18702.0	0.241578	
15	Minnehaha	46	30610.0	42043.0	78263.0	0.391117	
16	Turner	46	961.0	2937.0	4150.0	0.231566	
17	Walker	48	6085.0	12878.0	19683.0	0.309150	
18	Santa Isabel	72	NaN	NaN	NaN	NaN	
19	Aurora	46	340.0	974.0	1407.0	0.241649	
20	Brooks	48	1937.0	613.0	2596.0	0.746148	
21	Caguas	72	NaN	NaN	NaN	NaN	
22	Cataño	72	NaN	NaN	NaN	NaN	
23	Morovis	72	NaN	NaN	NaN	NaN	
24	Chase	31	171.0	1621.0	1851.0	0.092382	
25	Florida	72	NaN	NaN	NaN	NaN	
26	Conejos	08	1753.0	1885.0	3970.0	0.441562	
27	Washington	24	19193.0	38842.0	60731.0	0.316033	
28	Norton	20	277.0	1808.0	2190.0	0.126484	

```
29             Ford     17    1410.0    4474.0     6289.0  0.224201
...             ...    ...       ...       ...        ...       ...
3203        Grafton     33   28510.0   19010.0    50065.0  0.569460
3204      Sheboygan     55   22636.0   32368.0    58290.0  0.388334
3205          Smith     28    1598.0    5877.0     7556.0  0.211488
3206    St. Tammany     22   27716.0   90914.0   124389.0  0.222817
3207 Prince George's    24  313627.0   29290.0   351091.0  0.893293
3208        Stearns     27   25575.0   47618.0    78985.0  0.323796
3209      Middlesex     25  508919.0  216163.0   767337.0  0.663227
3210       Franklin     42   17322.0   49554.0    69345.0  0.249795
3211          Perry     17    2433.0    6816.0     9706.0  0.250670
3212      Gasconade     29    1519.0    5670.0     7444.0  0.204057
3213          Union     18     715.0    2445.0     3286.0  0.217590
3214      Jefferson     29   31546.0   68973.0   105969.0  0.297691
3215         Rogers     40    7895.0   30893.0    40834.0  0.193344
3216        Calhoun     45    3569.0    3785.0     7553.0  0.472527
3217        Jackson     54    2648.0    8959.0    12108.0  0.218698
3218           Rota     69       NaN       NaN        NaN       NaN
3219        Salinas     72       NaN       NaN        NaN       NaN
3220        Jackson     48     904.0    4266.0     5275.0  0.171374
3221         Kitsap     53   60561.0   46762.0   118625.0  0.510525
3222   Hillsborough     12  306422.0  265928.0   595072.0  0.514933
3223         DuPage     17  222499.0  164355.0   412929.0  0.538831
3224        Passaic     34  112608.0   71488.0   189099.0  0.595498
3225        Grayson     48   10276.0   35274.0    47068.0  0.218322
3226       Treutlen     13     860.0    1808.0     2700.0  0.318519
3227        Webster     28    1001.0    3949.0     5024.0  0.199244
3228          Keith     31     560.0    3203.0     3945.0  0.141952
3229          Adams     28    6921.0    5125.0    12214.0  0.566645
3230        Ontario     36   20233.0   24343.0    47664.0  0.424492
3231          Mason     54    2069.0    7597.0    10115.0  0.204548
3232        Yavapai     04   29705.0   58862.0    92780.0  0.320166

       per_gop    diff per_point_diff state_abbr          county_name  \
0     0.736713  38,171         51.45%         AZ        Mohave County
1     0.647811  17,064         32.58%         LA    Tangipahoa Parish
2     0.677346     824         47.14%         ID       Lincoln County
3     0.610899   4,267         29.03%         MN          Polk County
4     0.504245   8,447         10.72%         ND          Cass County
5     0.625823   4,055         34.24%         SD      Lawrence County
6     0.644914   4,396         34.69%         NY     Schoharie County
7     0.528870  37,410         15.17%         AK               Alaska
8     0.846426   8,105         71.62%         GA      Haralson County
9     0.753344   2,623         53.16%         GA      Bleckley County
10    0.735689   9,824         51.50%         IN      Lawrence County
11    0.739721   5,858         52.70%         IN      Jennings County
12    0.665625  17,303         38.34%         MI        Lapeer County
13    0.523067     478          6.06%         MS     Chickasaw County
```

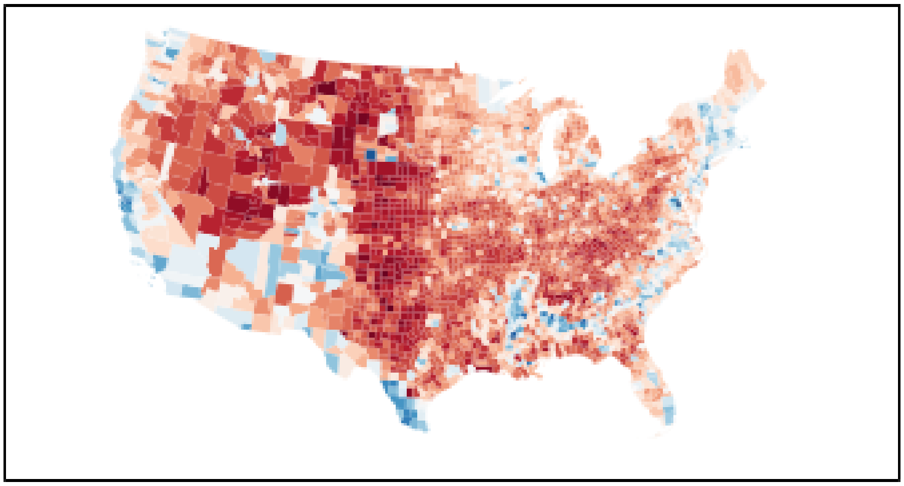
14	0.709282	8,747	46.77%	OH	Crawford County
15	0.537201	11,433	14.61%	SD	Minnehaha County
16	0.707711	1,976	47.61%	SD	Turner County
17	0.654270	6,793	34.51%	TX	Walker County
18	NaN	NaN	NaN	NaN	NaN
19	0.692253	634	45.06%	SD	Aurora County
20	0.236133	1,324	51.00%	TX	Brooks County
21	NaN	NaN	NaN	NaN	NaN
22	NaN	NaN	NaN	NaN	NaN
23	NaN	NaN	NaN	NaN	NaN
24	0.875743	1,450	78.34%	NE	Chase County
25	NaN	NaN	NaN	NaN	NaN
26	0.474811	132	3.32%	CO	Conejos County
27	0.639575	19,649	32.35%	MD	Washington County
28	0.825571	1,531	69.91%	KS	Norton County
29	0.711401	3,064	48.72%	IL	Ford County
...	...	...	...	...	...
3203	0.379706	9,500	18.98%	NH	Grafton County
3204	0.555293	9,732	16.70%	WI	Sheboygan County
3205	0.777792	4,279	56.63%	MS	Smith County
3206	0.730885	63,198	50.81%	LA	St. Tammany Parish
3207	0.083426	284,337	80.99%	MD	Prince George's County
3208	0.602874	22,043	27.91%	MN	Stearns County
3209	0.281705	292,756	38.15%	MA	Middlesex County
3210	0.714601	32,232	46.48%	PA	Franklin County
3211	0.702246	4,383	45.16%	IL	Perry County
3212	0.761687	4,151	55.76%	MO	Gasconade County
3213	0.744066	1,730	52.65%	IN	Union County
3214	0.650879	37,427	35.32%	MO	Jefferson County
3215	0.756551	22,998	56.32%	OK	Rogers County
3216	0.501125	216	2.86%	SC	Calhoun County
3217	0.739924	6,311	52.12%	WV	Jackson County
3218	NaN	NaN	NaN	NaN	NaN
3219	NaN	NaN	NaN	NaN	NaN
3220	0.808720	3,362	63.73%	TX	Jackson County
3221	0.394200	13,799	11.63%	WA	Kitsap County
3222	0.446884	40,494	6.80%	FL	Hillsborough County
3223	0.398022	58,144	14.08%	IL	DuPage County
3224	0.378045	41,120	21.75%	NJ	Passaic County
3225	0.749426	24,998	53.11%	TX	Grayson County
3226	0.669630	948	35.11%	GA	Treutlen County
3227	0.786027	2,948	58.68%	MS	Webster County
3228	0.811914	2,643	67.00%	NE	Keith County
3229	0.419600	1,796	14.70%	MS	Adams County
3230	0.510721	4,110	8.62%	NY	Ontario County
3231	0.751063	5,528	54.65%	WV	Mason County
3232	0.634426	29,157	31.43%	AZ	Yavapai County

```
      combined_fips
0            4015.0
1           22105.0
2           16063.0
3           27119.0
4           38017.0
5           46081.0
6           36095.0
7            2275.0
8           13143.0
9           13023.0
10          18093.0
11          18079.0
12          26087.0
13          28017.0
14          39033.0
15          46099.0
16          46125.0
17          48471.0
18              NaN
19          46003.0
20          48047.0
21              NaN
22              NaN
23              NaN
24          31029.0
25              NaN
26           8021.0
27          24043.0
28          20137.0
29          17053.0
...             ...
3203        33009.0
3204        55117.0
3205        28129.0
3206        22103.0
3207        24033.0
3208        27145.0
3209        25017.0
3210        42055.0
3211        17145.0
3212        29073.0
3213        18161.0
3214        29099.0
3215        40131.0
3216        45017.0
3217        54035.0
3218            NaN
```

```
3219           NaN
3220       48239.0
3221       53035.0
3222       12057.0
3223       17043.0
3224       34031.0
3225       48181.0
3226       13283.0
3227       28155.0
3228       31101.0
3229       28001.0
3230       36069.0
3231       54053.0
3232        4025.0

[3233 rows x 19 columns]
```

Now let's plot each county, with the `facecolor` representing the percentage of voters in each county that voted for the Democratic candidate (`per_dem`). To do this, we normalize all values to be between 0 and 1, and define a color map that maps numbers in $[0, 1]$ to a color.

```
In [9]: ax = plt.axes(
            projection=ccrs.LambertConformal(
                central_latitude=39,
                central_longitude=-96,
                standard_parallels=(33, 45)
            )
        )
        ax.set_extent([-125, -66.5, 20, 50])

        # Read in county-level shapefiles
        fname = "/data301/data/cb_2017_us_county_5m/cb_2017_us_county_5m"
        shp = Reader(fname)

        # define a normalizer and a color map
        import matplotlib as mpl
        norm = mpl.colors.Normalize(vmin=all_data["per_dem"].min(),
                                    vmax=all_data["per_dem"].max())
        cmap = plt.cm.RdBu

        # plot the geometries with a facecolor that depends on per_dem
        for geometry, (_, row) in zip(shp.geometries(), all_data.iterrows()):
            if ~pd.isnull(row["per_dem"]):
                ax.add_geometries([geometry],
                                  ccrs.PlateCarree(),
                                  facecolor=cmap(norm(row["per_dem"])))
```



2 Exercises

Exercise 1. Use the shapefiles for the countries of the world (`/data301/data/TM_WORLD_BORDERS_SIMPL-0.3/`) to make a choropleth showing carbon dioxide emissions per capita in 2014 (`/data301/data/co2.csv`).

(*Hint:* Some countries are missing data. One way to handle this is to: (1) fill the missing values with an arbitrary value in the range when making the initial map, and (2) go back and re-draw the polygons for those countries on top of the existing map, using a special face color to indicate that data was missing.)

```
In [10]: !ls /data301/data/TM_WORLD_BORDERS_SIMPL-0.3/
```

```
TM_WORLD_BORDERS_SIMPL-0.dbf  TM_WORLD_BORDERS_SIMPL-0.shp
TM_WORLD_BORDERS_SIMPL-0.prj  TM_WORLD_BORDERS_SIMPL-0.shx
```

```
In [11]: !ls /data301/data/co2.csv
```

```
/data301/data/co2.csv
```

```
In [13]: ax = plt.axes(
             projection=ccrs.Robinson())


         # Read in county-level shapefiles
         fname = "/data301/data/cb_2017_us_county_5m/cb_2017_us_county_5m"
         fname = "/data301/data/TM_WORLD_BORDERS_SIMPL-0.3/TM_WORLD_BORDERS_SIMPL-0"
         shp = Reader(fname)
```

```
    # Add each county to the data set.
    ax.add_geometries(shp.geometries(),
                      ccrs.PlateCarree(),
                      facecolor="None",
                      edgecolor='black')
```

Out[13]: <cartopy.mpl.feature_artist.FeatureArtist at 0x7fb26591ce48>