Instructions: 1) Save a copy of this notebook in your drive (File > Save a copy in Drive). Switch to it 2) Place "IPEDS_Dataset_Encoded" dataset in your Google drive 3) On the left of Colab click the folder icon, then Google drive icon. Allow access 4) Run this file

```
# Get institution ethnicity ratios

import pandas as pd
EFA_path = "/content/drive/MyDrive/IPEDS_Dataset_Encoded/Fall Enrollment/EFA_2015-2020_data.csv"
df_EFA = pd.read_csv(EFA_path)

# Select all students and sum together all years
df_EFA = df_EFA[df_EFA['efalevel']=='All students total']
df_EFA = df_EFA[["unitid", "year", "eftotlt", "efaiant", "efasiat", "efbkaat", "efhispt", "efnhpit", "efwhitt", "ef2mort", "ef
df_EFA = df_EFA.groupby('unitid').sum()
# df_EFA = df_EFA[df_EFA['year']=='2020']
df_EFA.reset_index(inplace=True)
df_EFA.drop('year', inplace=True, axis=1)
df_EFA.head()

from scipy.stats import entropy, chisquare
def shannon_index(row):
  # https://en.wikipedia.org/wiki/Diversity_index
  row_ps = [row.airatioU, row.asratioU, row.bkratioU, row.hiratioU, row.nhratioU, row.whratioU, row.tmratioU]
  return entropy(row_ps)

df_EFA['eftotlt'] -= (df_EFA['efunknt']+df_EFA['efnralt']) # remove from unis because it is missing in baseline data
df_EFA = df_EFA[df_EFA['eftotlt']>0]  # avoid divBy0

# Calculate ethnicity ratios and the diversity index
df_EFA['airatioU'] = df_EFA.apply(lambda row: row.efaiant / row.eftotlt, axis=1)
df_EFA['asratioU'] = df_EFA.apply(lambda row: row.efasiat / row.eftotlt, axis=1)
df_EFA['bkratioU'] = df_EFA.apply(lambda row: row.efbkaat / row.eftotlt, axis=1)
df_EFA['hiratioU'] = df_EFA.apply(lambda row: row.efhispt / row.eftotlt, axis=1)
df_EFA['nhratioU'] = df_EFA.apply(lambda row: row.efnhpit / row.eftotlt, axis=1)
df_EFA['whratioU'] = df_EFA.apply(lambda row: row.efwhitt / row.eftotlt, axis=1)
df_EFA['tmratioU'] = df_EFA.apply(lambda row: row.ef2mort / row.eftotlt, axis=1)
# df_EFA['unratioU'] = df_EFA.apply(lambda row: (row.efunknt+row.efnralt) / row.eftotlt, axis=1)
# df_EFA['nrratioU'] = df_EFA.apply(lambda row: row.efnralt / row.eftotlt, axis=1) # combine with un for consistency with cens
df_EFA['diversityU'] = df_EFA.apply(shannon_index, axis=1)
df_EFA.head()
```

	unitid	eftotlt	efaiant	efasiat	efbkaat	efhispt	efnhpit	efwhitt	ef2mor
0	100654	33802	85	98	31639	290	47	1169	47
1	100663	117964	335	7434	26633	4519	48	74531	446
2	100690	3199	11	21	1954	67	11	1120	1
3	100706	50450	567	2046	5482	2478	49	38439	138
4	100724	27390	41	125	25519	302	19	1072	31

```
# Combine with state/county information
filename_fips = "/content/drive/MyDrive/DO2022_additional_data/county_fips_master.csv"
#from https://github.com/kjhealy/fips-codes/blob/master/county_fips_master.csv
df_fips = pd.read_csv(filename_fips, encoding='latin-1')
df_fips.drop(['county','state'], inplace=True, axis=1)
df_fips.rename(columns = {"county_name":"county", "state_name":"state"}, inplace = True)
df_fips = df_fips[["fips", "county", "state"]]
df_fips["fips"] = df_fips.apply(lambda row: str(row.fips).zfill(5), axis=1)

HD_path = "/content/drive/MyDrive/IPEDS_Dataset_Encoded/Institutional Characteristics/HD_2015-2021_data.csv"
df_counties = pd.read_csv(HD_path)
df_counties = df_counties.rename(columns={'fips': 'state', 'countynm': 'county'})
df_counties = df_counties[["unitid", "year", "county", "state", "countycd", "longitud", "latitude", "instnm"]]
df_counties.drop_duplicates(subset="unitid", keep='first', inplace=True)  # drop older years
df_counties.drop('year', inplace=True, axis=1)

df_unitid_fips = pd.merge(df_fips, df_counties, on=["county", "state"])
df_unitid_fips
df_ratiosU = df_EFA.merge(df_unitid_fips, on='unitid')
df_ratiosU["county_state"] = df_ratiosU["county"] + ", " + df_ratiosU["state"]
df_ratiosU["fips_state"] = df_ratiosU.fips.str[:2]
df_ratiosU["state_abrv"] = df_ratiosU.countycd.str[-2:]
df_ratiosU.drop(['efhispt','efwhitt','efbkaat','efaiant','efasiat','efnhpit','efunknt','ef2mort','efnralt'], inplace=True, axi

filename_fin = "/content/drive/MyDrive/IPEDS_Dataset_Encoded/Institutional Finances/F_F3_1415-1920_data.csv"
```

```
df_fin = pd.read_csv(filename_fin)
df_fin.drop_duplicates(subset="unitid", keep='first', inplace=True)
df_fin = df_fin[["unitid"]]
df_fin["profit"] = 1
df_ratiosU = df_ratiosU.merge(df_fin, on='unitid', how='left')
df_ratiosU["profit"] = df_ratiosU["profit"].fillna(0)

df_fips_to_state = df_ratiosU[["fips", "state_abrv"]].copy()
df_fips_to_state.drop_duplicates(subset="fips", keep='first', inplace=True)

# df_ratiosU["private"].sum()
df_ratiosU
```

```
/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3326: DtypeWarning: Columns (13,23,48,49,50,51,5
  exec(code_obj, self.user_global_ns, self.user_ns)
/usr/local/lib/python3.8/dist-packages/IPython/core/interactiveshell.py:3326: DtypeWarning: Columns (130) have mixed typ
  exec(code_obj, self.user_global_ns, self.user_ns)
```

	unitid	eftotlt	airatioU	asratioU	bkratioU	hiratioU	nhratioU	whratioU	tmratioU	diversityU	...	county	
0	100654	33802	0.002515	0.002899	0.936010	0.008579	0.001390	0.034584	0.014023	0.320050	...	Madison County	
1	100663	117964	0.002840	0.063019	0.225772	0.038308	0.000407	0.631811	0.037842	1.069013	...	Jefferson County	
2	100690	3199	0.003439	0.006565	0.610816	0.020944	0.003439	0.350109	0.004689	0.846669	...	Montgomery County	
3	100706	50450	0.011239	0.040555	0.108662	0.049118	0.000971	0.761923	0.027532	0.882441	...	Madison County	
4	100724	27390	0.001497	0.004564	0.931690	0.011026	0.000694	0.039138	0.011391	0.332807	...	Montgomery County	
...	...	...	...	...	...	...	...	...	...	...	...	...	
7454	496265	26	0.000000	0.000000	0.000000	0.038462	0.000000	0.961538	0.000000	0.163024	...	Franklin County	Per
7455	496283	15	0.000000	0.000000	0.000000	0.200000	0.000000	0.800000	0.000000	0.500402	...	Bonneville County	
7456	496326	21	0.000000	0.047619	0.000000	0.047619	0.095238	0.714286	0.095238	0.978173	...	Ada County	
7457	496371	2	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	0.000000	0.000000	...	Lawrence County	
7458	496423	7	0.000000	0.142857	0.000000	0.285714	0.000000	0.571429	0.000000	0.955700	...	Whatcom County	W

7459 rows × 21 columns


```
# Get baseline data for ethnicity per county
RacePerCounty_path = "/content/drive/MyDrive/DO2022_additional_data/censusDataAGE15-24.csv" # 15-24 yo
df_RPC = pd.read_csv(RacePerCounty_path)
df_RPC = df_RPC.rename(columns={'countyFIPS': 'fips', 'TOT_POP':'eftotlt','Hispanic': 'efhispt', 'White':'efwhitt', 'Black or
colsC = ['eftotlt','efhispt','efwhitt','efbkaat','efaiant','efasiat','efnhpit','ef2mort']
df_RPC[colsC]=df_RPC[colsC].astype("float")
df_RPC = df_RPC[df_RPC['eftotlt']>0]  # avoid divBy0
# print(df_RPC["eftotlt"].min())
df_RPC['fips'] = df_RPC['fips'].astype("string")
df_RPC["fips"] = df_RPC.apply(lambda row: str(row.fips).zfill(5), axis=1)

# RacePerCounty_path = "/content/drive/MyDrive/DO2022_additional_data/race_per_county_clean.csv"
# df_RPC = pd.read_csv(RacePerCounty_path)
# df_RPC.columns = df_RPC.iloc[0]
# df_RPC = df_RPC.iloc[1: , :]
# df_RPC = df_RPC.drop('Geography', axis=1)
# df_RPC = df_RPC.rename(columns={'Geographic Area Name': 'County', 'Total':'eftotlt','Hispanic or Latino total': 'efhispt', '
# colsC = ['eftotlt','efhispt','efwhitt','efbkaat','efaiant','efasiat','efnhpit','ef2mort']
# df_RPC[colsC]=df_RPC[colsC].apply(lambda x: x.str.replace(',',''))
# df_RPC[colsC]=df_RPC[colsC].astype("float")
# df_RPC.head()
```

```
# df_RPC.head()

# Baseline ratios for ethnicity per county
df_ratiosC = pd.DataFrame()
df_ratiosC['fips'] = df_RPC['fips']
df_ratiosC['airatioC'] = df_RPC.apply(lambda row: row.efaiant / row.eftotlt, axis=1)
df_ratiosC['asratioC'] = df_RPC.apply(lambda row: row.efasiat / row.eftotlt, axis=1)
df_ratiosC['bkratioC'] = df_RPC.apply(lambda row: row.efbkaat / row.eftotlt, axis=1)
df_ratiosC['hiratioC'] = df_RPC.apply(lambda row: row.efhispt / row.eftotlt, axis=1)
df_ratiosC['nhratioC'] = df_RPC.apply(lambda row: row.efnhpit / row.eftotlt, axis=1)
df_ratiosC['whratioC'] = df_RPC.apply(lambda row: row.efwhitt / row.eftotlt, axis=1)
df_ratiosC['tmratioC'] = df_RPC.apply(lambda row: row.ef2mort / row.eftotlt, axis=1)
# df_ratiosC['unratioC'] = df_RPC.apply(lambda row: 1 / row.eftotlt, axis=1)
df_ratiosC.head()
```

	fips	airatioC	asratioC	bkratioC	hiratioC	nhratioC	whratioC	tmratioC
0	01001	0.003833	0.012457	0.222177	0.026831	0.000684	0.708966	0.025051
1	01003	0.007665	0.019121	0.136445	0.049570	0.000426	0.763649	0.023124
2	01005	0.003481	0.001899	0.576899	0.030380	0.000949	0.371519	0.014873
3	01007	0.003929	0.002857	0.262500	0.024286	0.000000	0.698929	0.007500
4	01009	0.006074	0.002684	0.017940	0.115836	0.000565	0.841644	0.015256

```
# # Get baseline data for ethnicity per state
# RacePerState_path = "/content/drive/MyDrive/DO2022_additional_data/race_per_state_clean.csv"
# df_RPS = pd.read_csv(RacePerState_path)
# df_RPS.set_index('Label (Grouping)', inplace=True)
# df_RPS = df_RPS.copy().T
# df_RPS = df_RPS.dropna(axis='columns', how='all')
# df_RPS = df_RPS.reset_index()
# df_RPS = df_RPS.rename_axis(None, axis=1)
# df_RPS = df_RPS.rename(columns={'index': 'State', 'Total':'eftotlt','Hispanic or Latino total': 'efhispt', 'White total':'ef
# colsS = ['eftotlt','efhispt','efwhitt','efbkaat','efaiant','efasiat','efnhpit','efunknt','ef2mort']
# df_RPS[colsS]=df_RPS[colsS].apply(lambda x: x.str.replace(',',''))
# df_RPS[colsS]=df_RPS[colsS].astype("float")
# df_RPS.head()

# df_RPS['eftotlt'] -= (df_RPS['efunknt']) # remove from states because it is missing in other baseline data
# df_RPS = df_RPS[df_RPS['eftotlt']>0]  # avoid divBy0

df_RPS = df_RPC.merge(df_fips_to_state, on='fips')
df_RPS.drop(['fips'], inplace=True, axis=1)
df_RPS = df_RPS.groupby('state_abrv', as_index=False).sum()

# Baseline ratios for ethnicity per state
df_ratiosS = pd.DataFrame()
df_ratiosS['state_abrv'] = df_RPS['state_abrv']
# df_ratiosS['state'] = df_RPS['State']
df_ratiosS['airatioS'] = df_RPS.apply(lambda row: row.efaiant / row.eftotlt, axis=1)
df_ratiosS['asratioS'] = df_RPS.apply(lambda row: row.efasiat / row.eftotlt, axis=1)
df_ratiosS['bkratioS'] = df_RPS.apply(lambda row: row.efbkaat / row.eftotlt, axis=1)
df_ratiosS['hiratioS'] = df_RPS.apply(lambda row: row.efhispt / row.eftotlt, axis=1)
df_ratiosS['nhratioS'] = df_RPS.apply(lambda row: row.efnhpit / row.eftotlt, axis=1)
df_ratiosS['whratioS'] = df_RPS.apply(lambda row: row.efwhitt / row.eftotlt, axis=1)
df_ratiosS['tmratioS'] = df_RPS.apply(lambda row: row.ef2mort / row.eftotlt, axis=1)
# df_ratiosS['unratioS'] = df_RPS.apply(lambda row: row.efunknt / row.eftotlt, axis=1)
df_ratiosS.head()

# Baseline ratios for USA as a whole
airatioUS = df_RPS['efaiant'].sum() / df_RPS['eftotlt'].sum()
asratioUS = df_RPS['efasiat'].sum() / df_RPS['eftotlt'].sum()
bkratioUS = df_RPS['efbkaat'].sum() / df_RPS['eftotlt'].sum()
hiratioUS = df_RPS['efhispt'].sum() / df_RPS['eftotlt'].sum()
nhratioUS = df_RPS['efnhpit'].sum() / df_RPS['eftotlt'].sum()
whratioUS = df_RPS['efwhitt'].sum() / df_RPS['eftotlt'].sum()
tmratioUS = df_RPS['ef2mort'].sum() / df_RPS['eftotlt'].sum()
# # unratioUS = df_RPS['efunknt'].sum() / df_RPS['eftotlt'].sum()


citutions vs state/county


.hiratioU, row.nhratioU, row.whratioU, row.tmratioU]
.hiratioS, row.nhratioS, row.whratioS, row.tmratioS]



.hiratioU, row.nhratioU, row.whratioU, row.tmratioU]
.hiratioC, row.nhratioC, row.whratioC, row.tmratioC]
```

```
.hiratioU, row.nhratioU, row.whratioU, row.tmratioU]
ıratioUS, whratioUS, tmratioUS]


.hiratioU, row.nhratioU, row.whratioU, row.tmratioU]
ıratioUS, whratioUS, tmratioUS]


r')
```

```
s=1)

L_US", "chi2_US", "diversityU", "state", "county", "fips", "fips_state", "state_abrv", "longitud", "latitude", "profit", 'efto
```

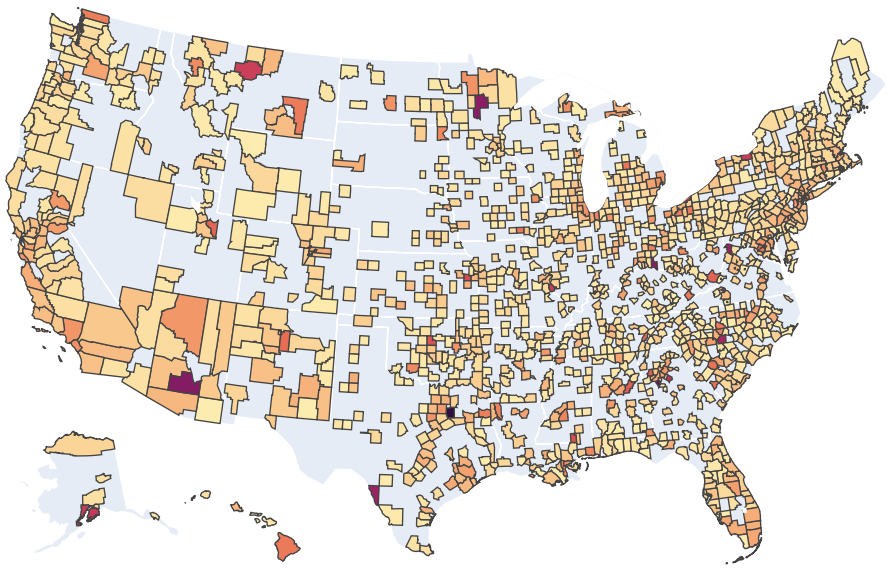
	unitid	KL_S	KL_C	KL_US	chi2_US	diversityU	state	county	fips	fips_state	state_abrv	longi
7327	238193	0.105243	0.090681	0.001408	0.002652	1.250260	Wisconsin	Milwaukee County	55079	55	WI	-87.965
4720	167534	0.040316	0.080868	0.006244	0.011708	1.193036	Massachusetts	Worcester County	25027	25	MA	-71.79
7129	226833	0.155722	0.007314	0.008092	0.014206	1.226810	Texas	Wichita County	48485	48	TX	-98.519
1648	129695	0.026082	0.005178	0.008699	0.015909	1.260637	Connecticut	Hartford County	09003	09	CT	-72.561
6761	243823	0.186651	0.366107	0.009569	0.018305	1.220216	Texas	Dallas County	48113	48	TX	-96.893
...	...	...	...	...	...	...	...	...	...	...	...	...
5446	200527	2.537098	inf	4.481102	112.540140	0.182958	North Dakota	Rolette County	38079	38	ND	-99.750
407	187596	1.898465	0.145226	4.517413	113.591561	0.186626	New Mexico	McKinley County	35031	35	NM	-108.149
384	105297	2.921498	0.169030	4.685073	118.365529	0.082513	Arizona	Apache County	04001	04	AZ	-109.21
3961	155140	4.658056	3.574919	4.806535	121.307070	0.000000	Kansas	Douglas County	20045	20	KS	-95.232
390	188216	2.096953	2.966206	4.806535	121.307070	0.000000	New Mexico	Bernalillo County	35001	35	NM	-106.664

7459 rows × 16 columns

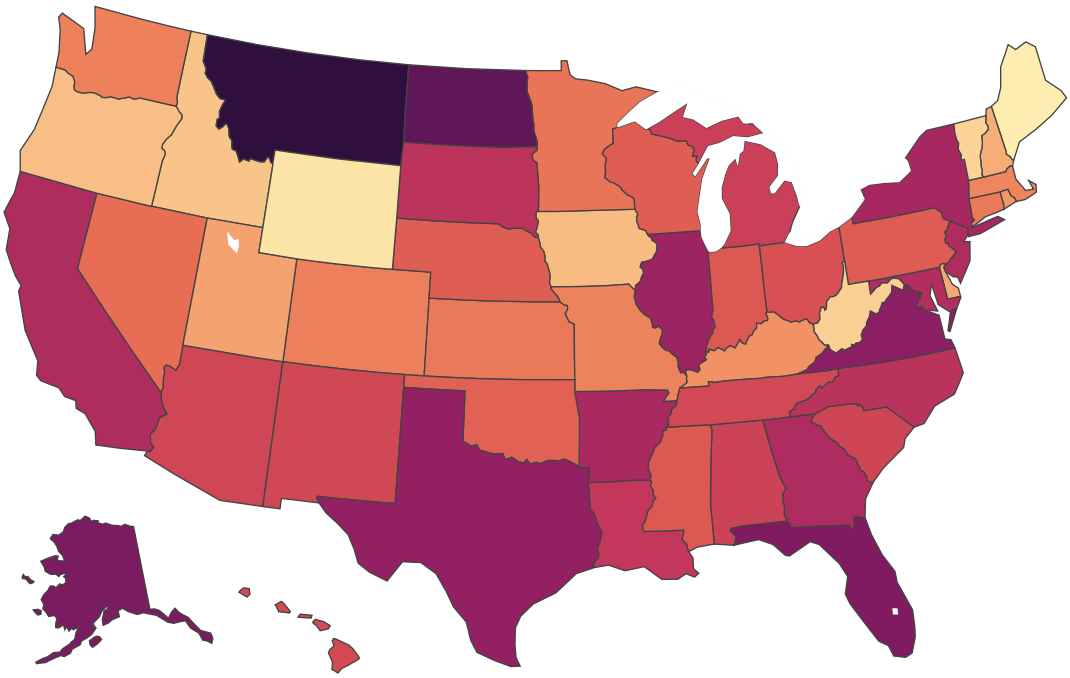

```
en


color_var, label="KL"):
ıbusercontent.com/plotly/datasets/master/geojson-counties-fips.json') as response:
ıse)
ıpby("fips")[color_var].mean()
s, geojson=counties, locations=df_counties.index, color=color_var, color_continuous_scale="matter", scope="usa", labels={color_
),"t":0,"l":0,"b":0})


lor_var, label="KL"):
```

```
by("state_abrv")[color_var].mean()
 locations=df_states.index, color=color_var,locationmode='USA-states', color_continuous_scale="matter", scope="usa", labels={c
),"t":0,"l":0,"b":0})
```

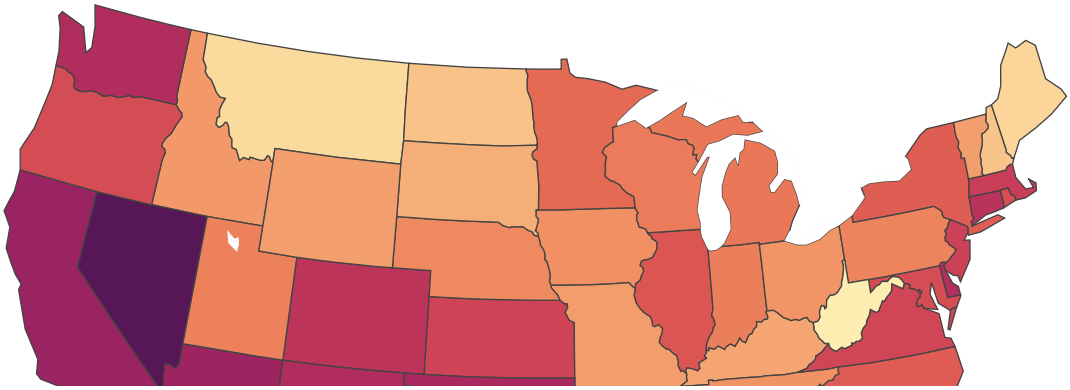
```
draw_counties(df_diversity, "KL_C")
```



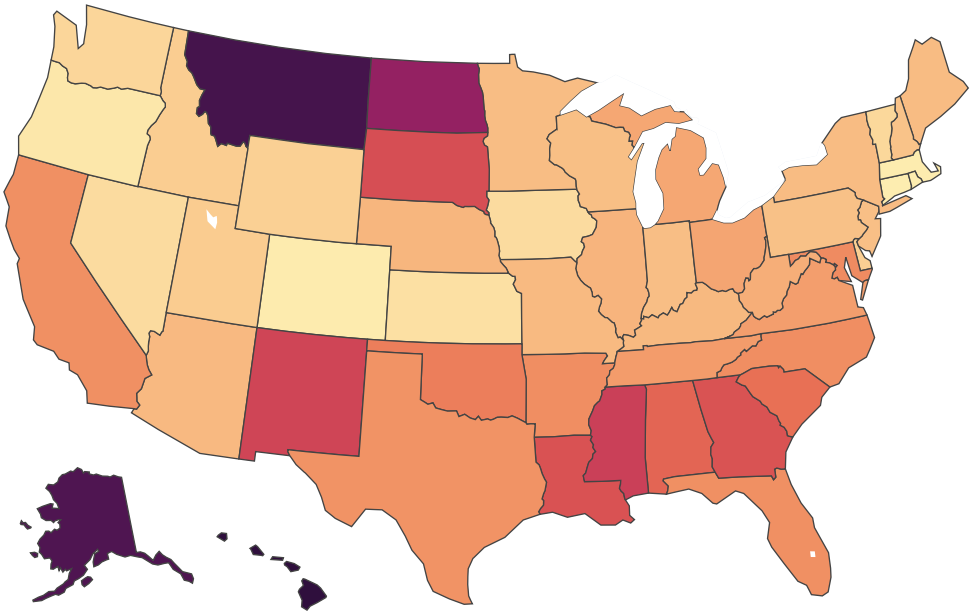
```
draw_states(df_diversity, "KL_S")
```



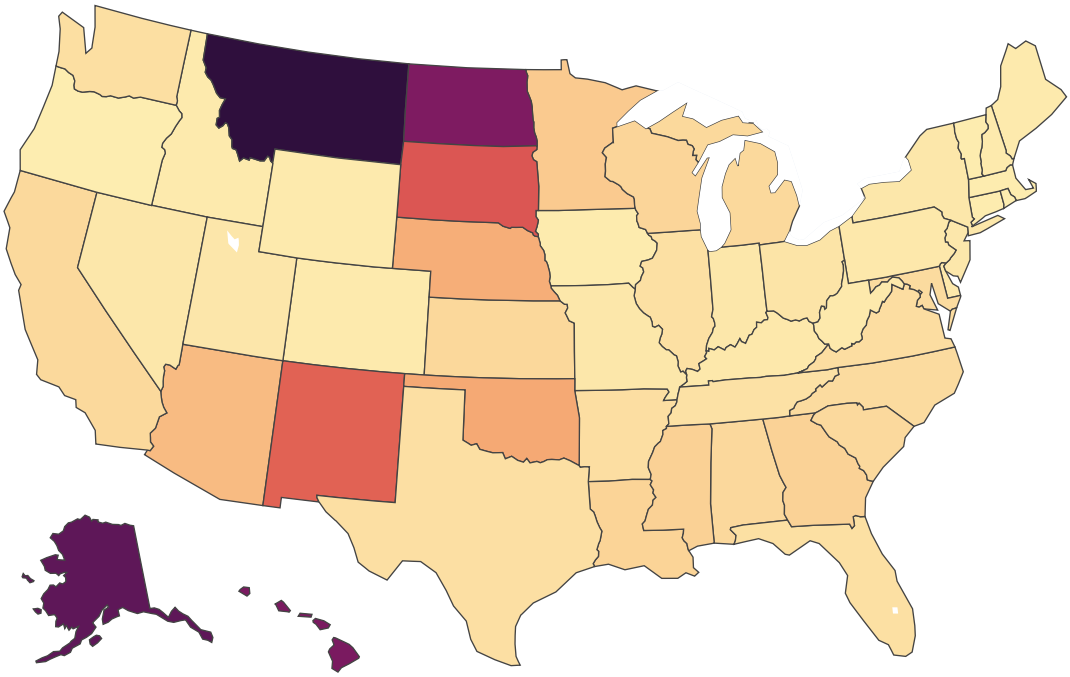
```
draw_states(df_diversity, "diversityU")
```



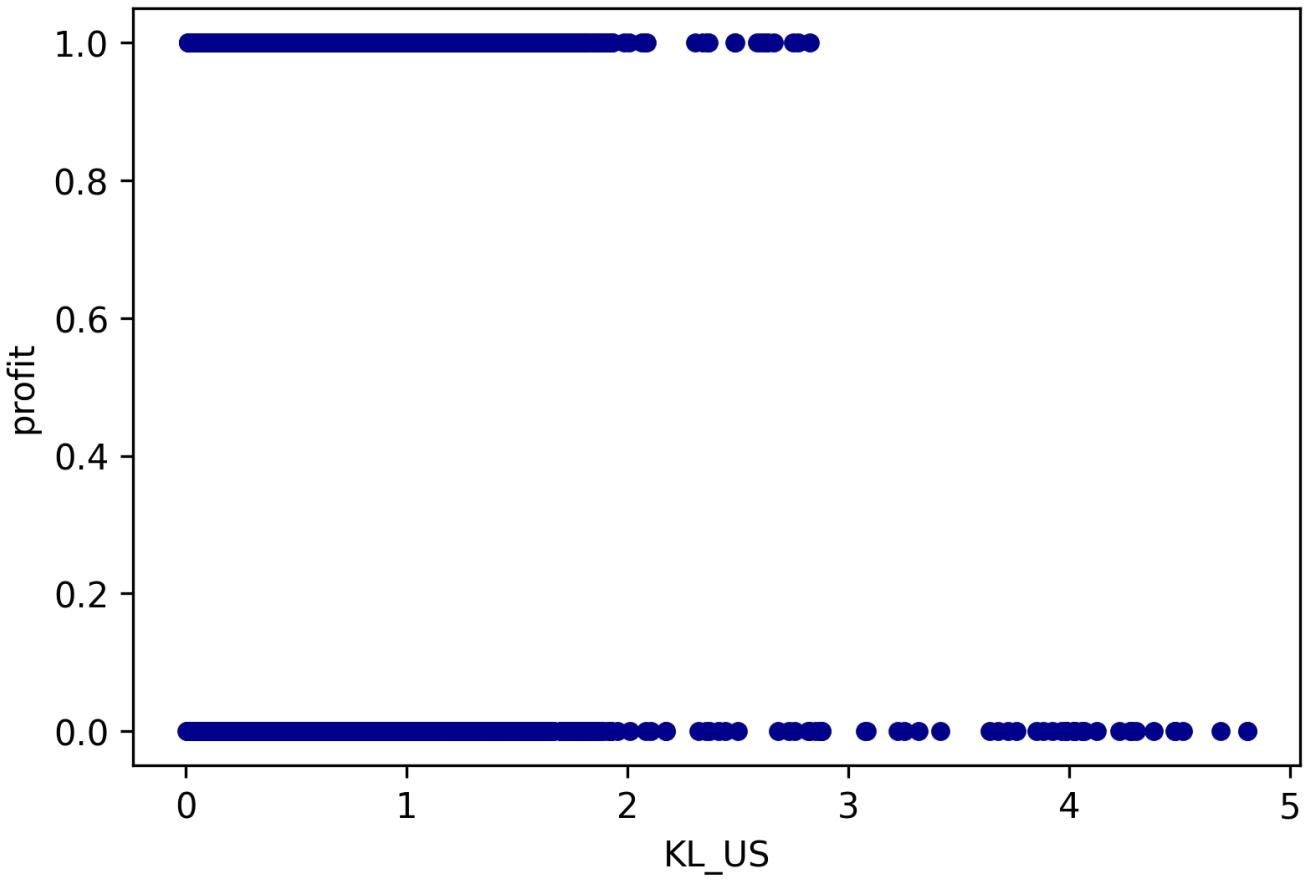
```
draw_states(df_diversity, "KL_US")
```



```
draw_states(df_diversity, "chi2_US")
```



```
ax1 = df_diversity.plot.scatter(x='KL_US',y='profit',c='DarkBlue')
```

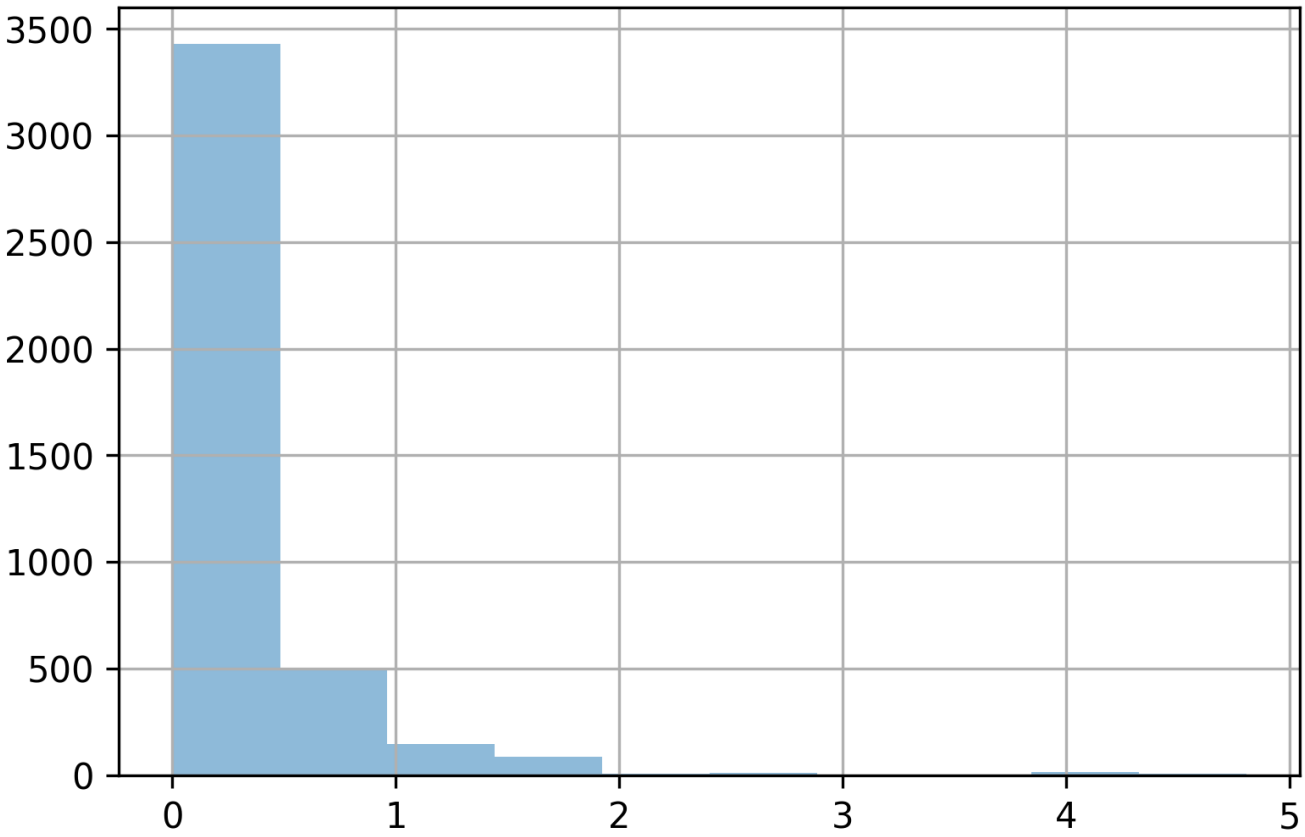



```
df_diversity[df_diversity["profit"]==1]["KL_US"].hist(bins=10,alpha=0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f893a4824c0>
```

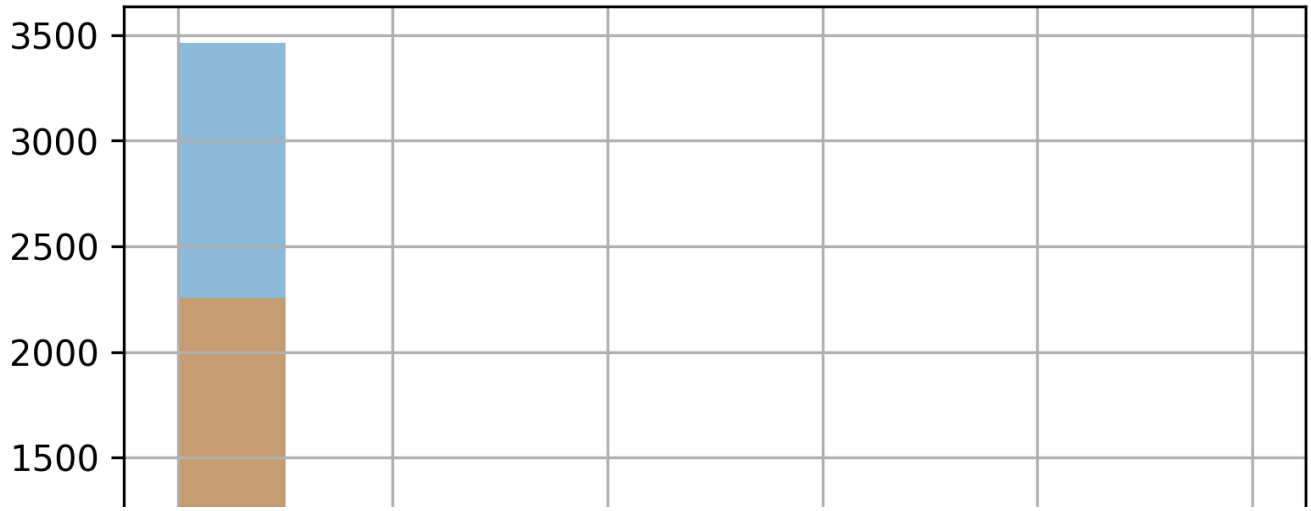

```
df_diversity[df_diversity["profit"]==0]["KL_US"].hist(bins=10,alpha=0.5)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f893918fa60>
```



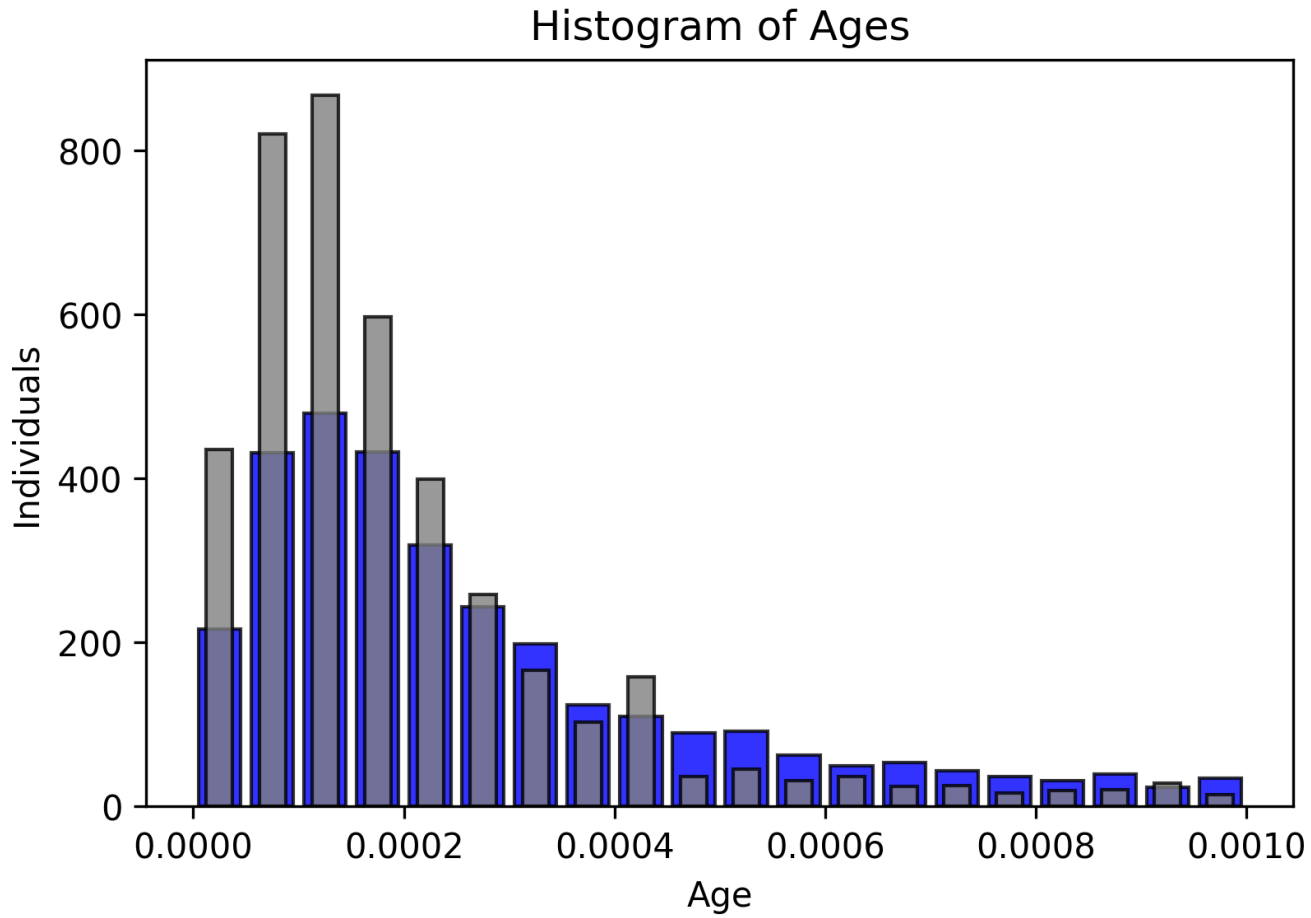

```
df_diversity[df_diversity["profit"]==0]["KL_US"].hist(bins=10,alpha=0.5,range=(0,5))
df_diversity[df_diversity["profit"]==1]["KL_US"].hist(bins=10,alpha=0.5,range=(0,5))
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f8938b32ca0>
```




```
plt.hist(df_diversity[df_diversity["profit"]==1]["KL_US"]/df_diversity[df_diversity["profit"]==1]["KL_US"].sum(), edgecolor='b
plt.hist(df_diversity[df_diversity["profit"]==0]["KL_US"]/df_diversity[df_diversity["profit"]==0]["KL_US"].sum(), edgecolor='b
plt.title("Histogram of Ages")
plt.xlabel("Deviation from representativeness (KL bin)")
plt.ylabel("Number of institutions")
```

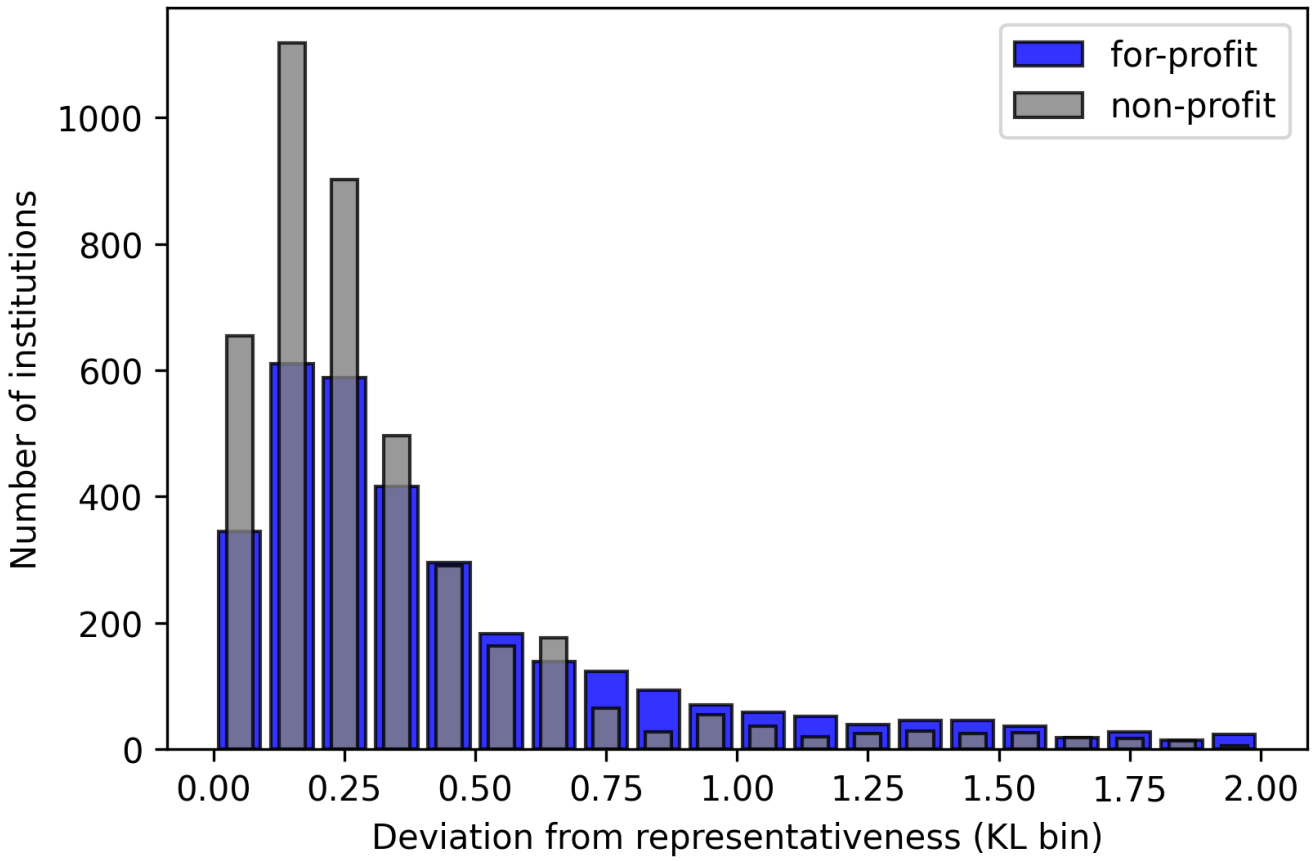
```
Text(0, 0.5, 'Individuals')
```




```
plt.hist(df_diversity[df_diversity["profit"]==1]["KL_US"], edgecolor='black',color='blue',rwidth=0.8,alpha=0.8,bins=20,range=(
plt.hist(df_diversity[df_diversity["profit"]==0]["KL_US"], edgecolor='black',color='gray',rwidth=0.5,alpha=0.8,bins=20,range=(
# plt.title("Diversity")
plt.xlabel("Deviation from representativeness (KL bin)")
```

```
plt.ylabel("Number of institutions")
plt.legend(prop={'size': 10})
```

```
<matplotlib.legend.Legend at 0x7f893b813eb0>
```




```
from scipy.stats import mannwhitneyu
mannwhitneyu(df_diversity[df_diversity["profit"]==0]["KL_US"], df_diversity[df_diversity["profit"]==1]["KL_US"], alternative="
```

```
MannwhitneyuResult(statistic=5467515.0, pvalue=4.554595377595352e-50)
```

```
df_diversity[df_diversity["eftotlt"]>1000].sort_values(by=['KL_US'])[:5]
```

	unitid	KL_S	KL_C	KL_US	chi2_US	diversityU	state	county	fips	fips_state	state_abrv	longitud
7327	238193	0.105243	0.090681	0.001408	0.002652	1.250260	Wisconsin	Milwaukee County	55079	55	WI	-87.965174
4720	167534	0.040316	0.080868	0.006244	0.011708	1.193036	Massachusetts	Worcester County	25027	25	MA	-71.79503
7129	226833	0.155722	0.007314	0.008092	0.014206	1.226810	Texas	Wichita County	48485	48	TX	-98.519386
1648	129695	0.026082	0.005178	0.008699	0.015909	1.260637	Connecticut	Hartford County	09003	09	CT	-72.561939
6761	243823	0.186651	0.366107	0.009569	0.018305	1.220216	Texas	Dallas County	48113	48	TX	-96.893564

```
df_diversity[df_diversity["eftotlt"]>1000].sort_values(by=['KL_US'])[-5:]
```

	unitid	KL_S	KL_C	KL_US	chi2_US	diversityU	state	county	fips	fips_state	state_abrv	longitud	la
5446	200527	2.537098	inf	4.481102	112.540140	0.182958	North Dakota	Rolette County	38079	38	ND	-99.750836	48
407	187596	1.898465	0.145226	4.517413	113.591561	0.186626	New Mexico	McKinley County	35031	35	NM	-108.149392	35
384	105297	2.921498	0.169030	4.685073	118.365529	0.082513	Arizona	Apache County	04001	04	AZ	-109.21684	36
3961	155140	4.658056	3.574919	4.806535	121.307070	0.000000	Kansas	Douglas County	20045	20	KS	-95.232879	38
390	188216	2.096953	2.966206	4.806535	121.307070	0.000000	New Mexico	Bernalillo County	35001	35	NM	-106.664475	35