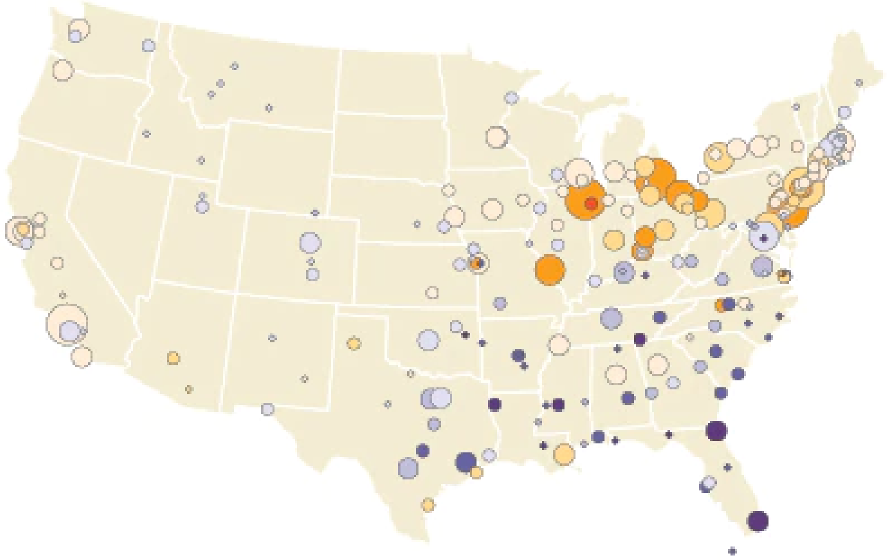
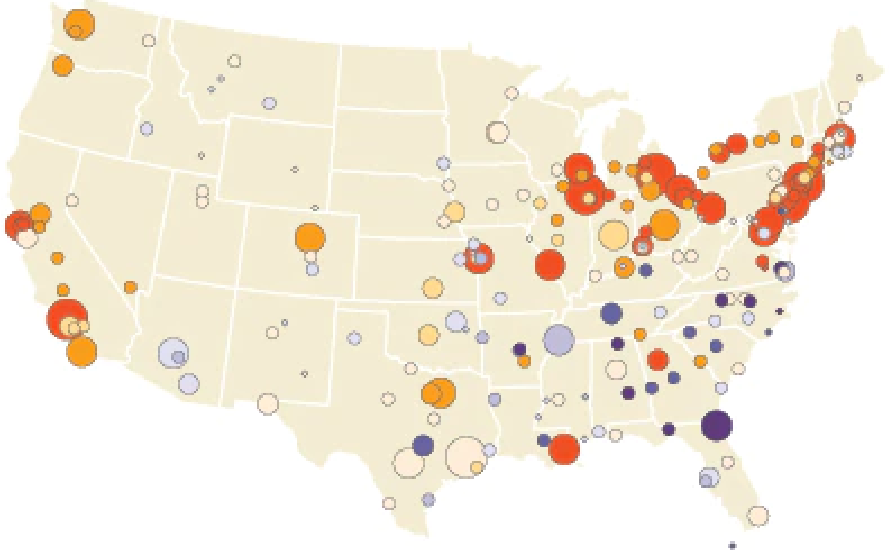
Analiza danych dotyczących migracji na terenie USA

The First Great Migration: 1910-1940



The Second Great Migration: 1940-1970



The change in share of Blacks in cities is based on the percentage point difference in the percent of population that was Black in the later time period compared to the earlier. For example, 18.3 percent of the population in Gary, IN was Black in 1940 but was just 2.3 in 1910, which represented a 16.0 percentage-point change in the share of Blacks in the city. It was the largest change in share during the First Great Migration. By the end of the Second Great Migration, Newark, NJ had realized the largest increase in Black population share, with the Black proportion of the city rising from 10.6 in 1940 to 54.2 in 1970.

Change in share of Blacks

Increasing

● 10.0 or more

● 5.0 to 9.9

● 2.5 to 4.9

○ 0.0 to 2.4

○ -2.4 to -0.1

● -5.0 to -2.5

● -10.0 to -5.1

● Less than -10.0

Decreasing

City population
(in later decade)

○ 1,000,000 or more

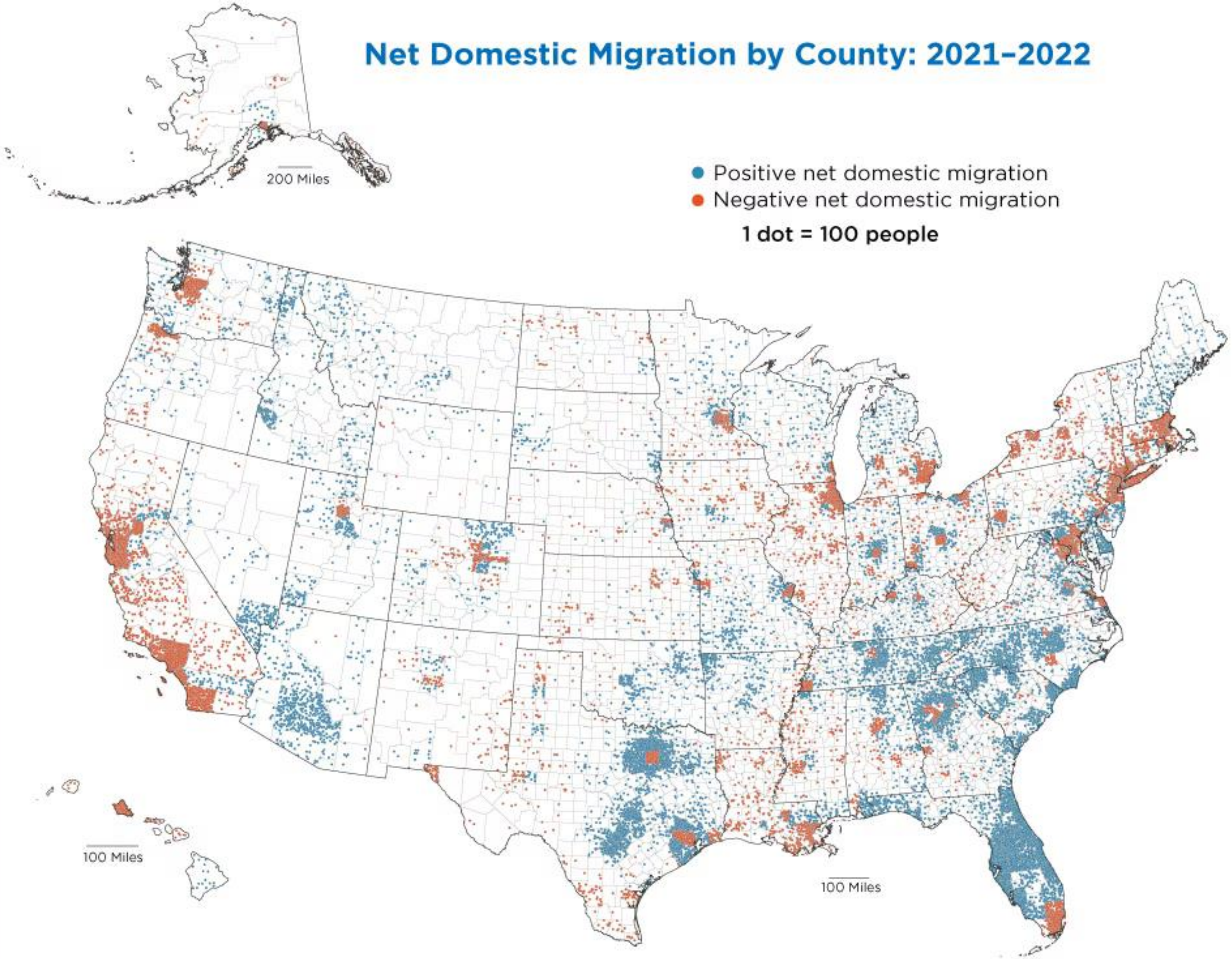
○ 500,000 to 999,999

○ 150,000 to 499,999

○ 50,000 to 149,999

○ Less than 50,000

Net Domestic Migration by County: 2021–2022



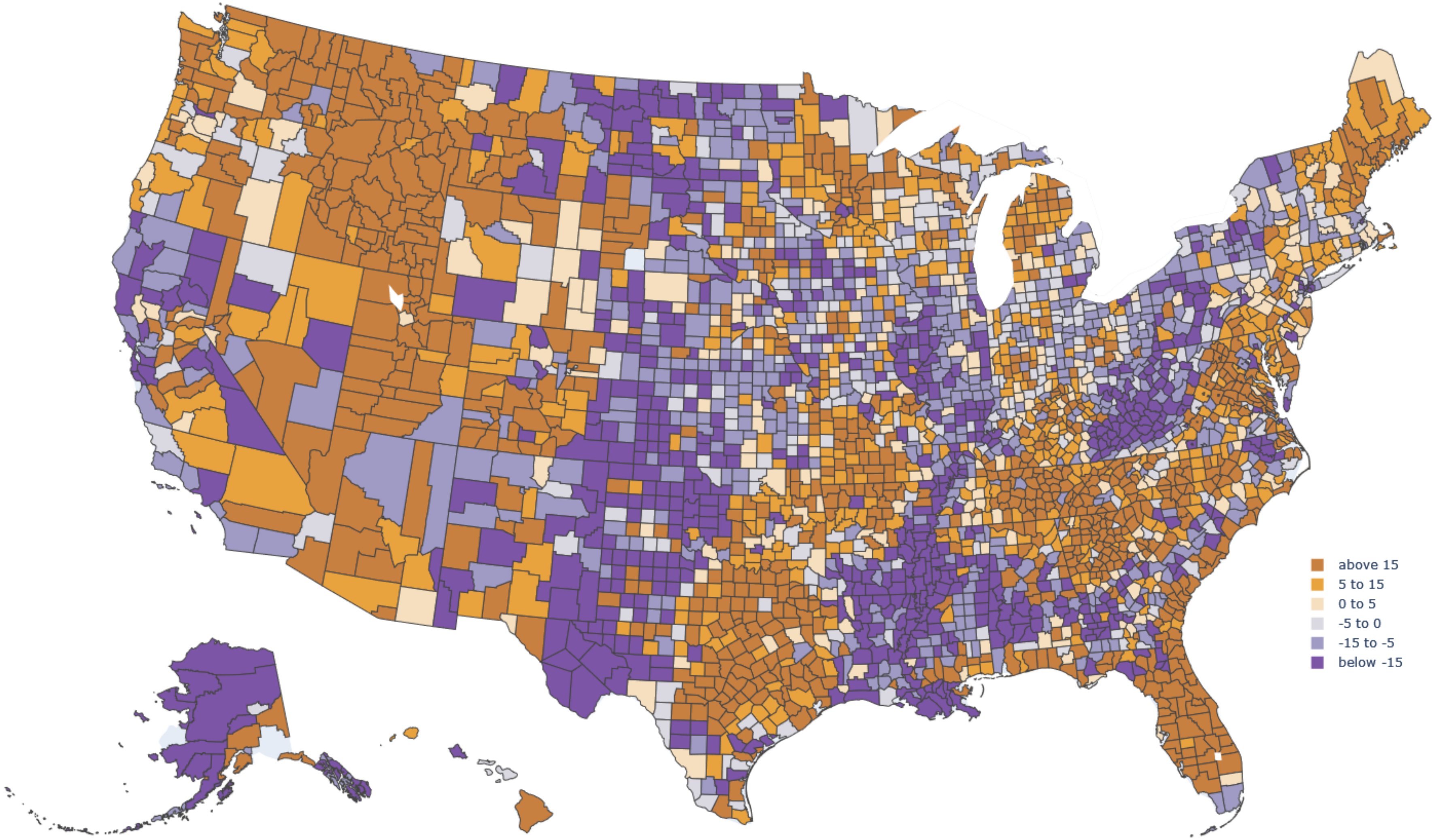
● Positive net domestic migration

● Negative net domestic migration

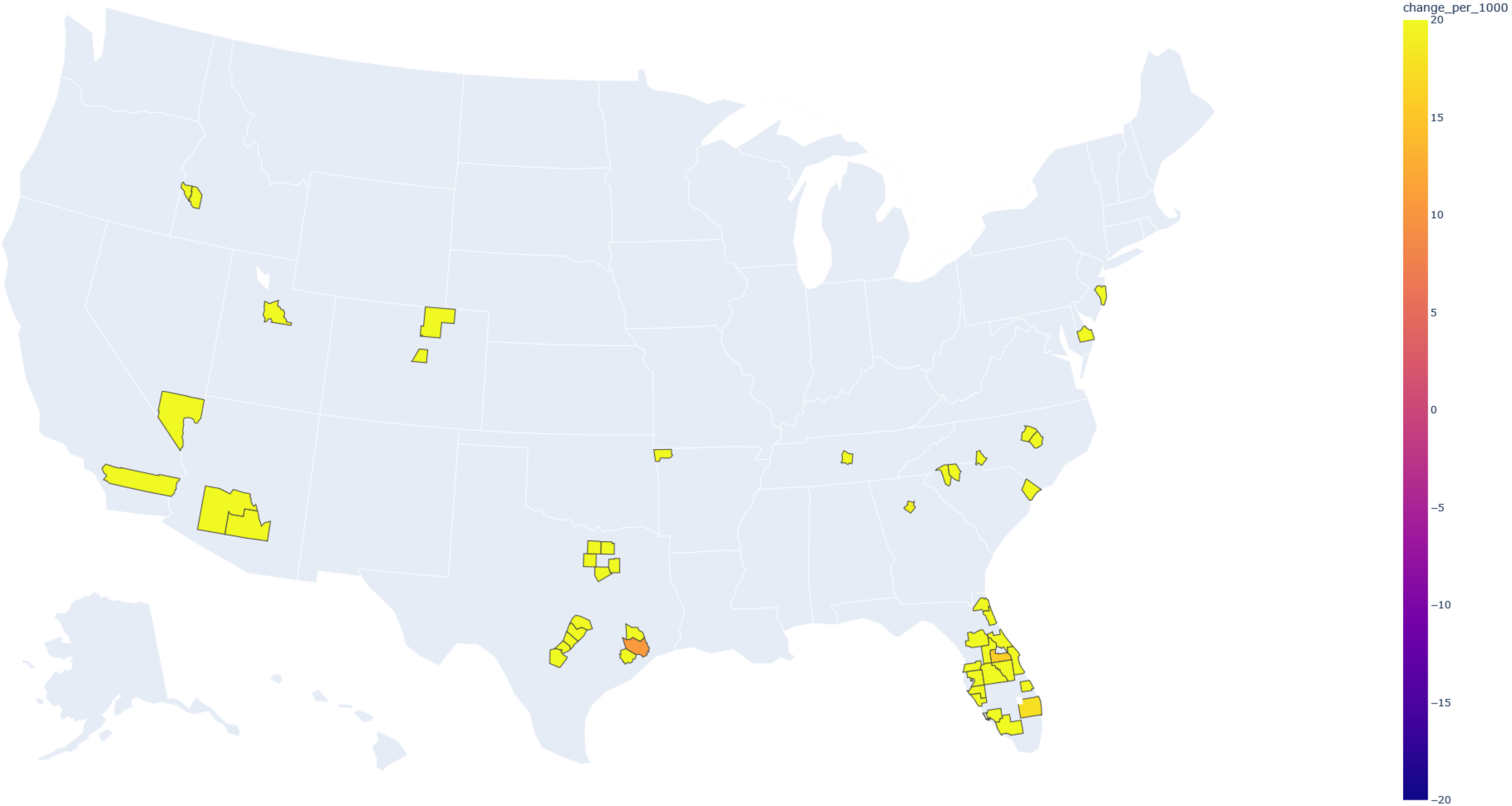
1 dot = 100 people

Source: U.S. Census Bureau, Vintage 2022 Population Estimates.

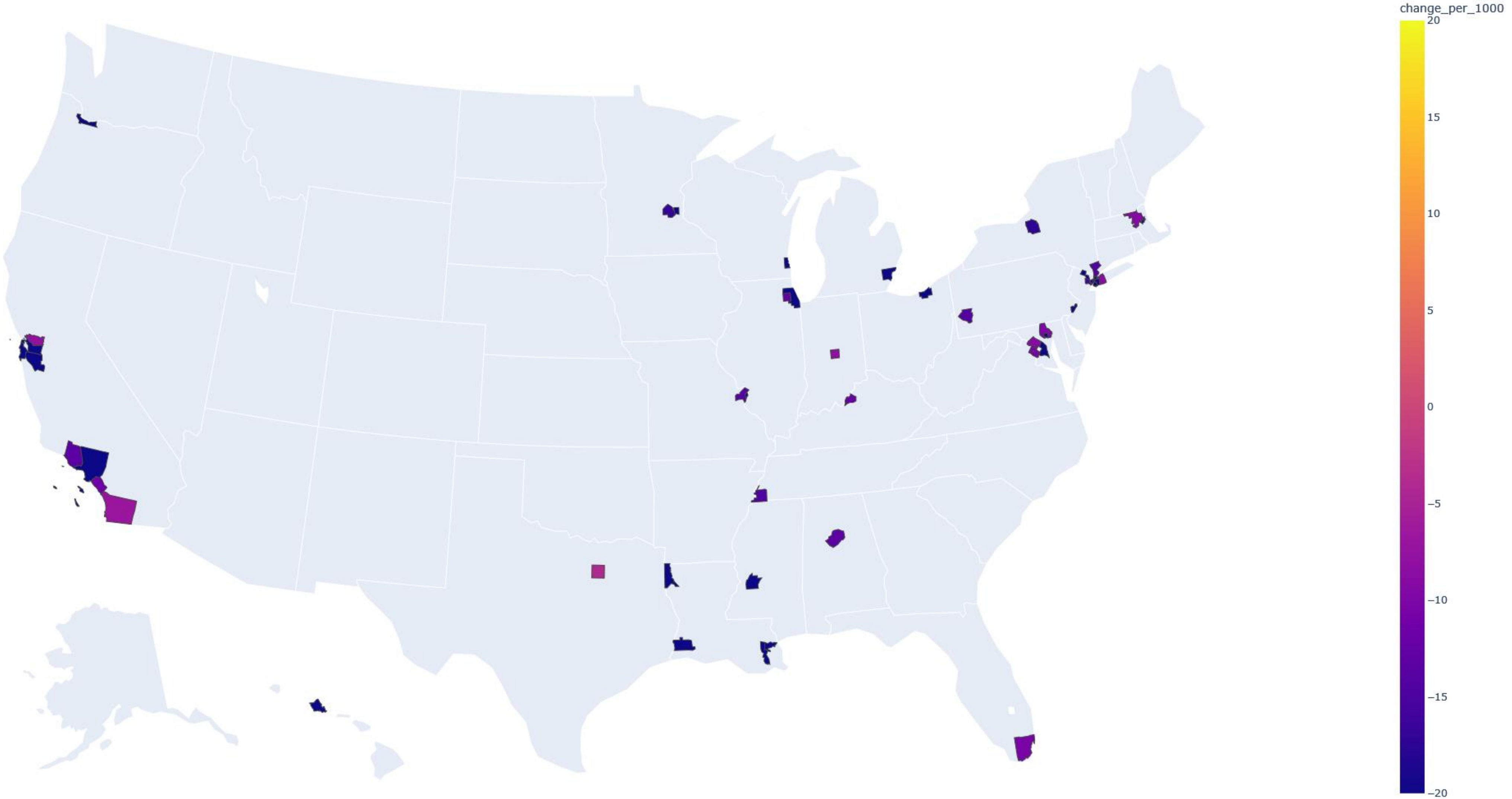
Net domestic migration per 1000 residents by county



50 hrabstw z największym przypływem ludności



50 hrabstw z największą emigracją



Dane

- CENSUS
- Bezrobocie
- PKB i zarobki
- House price index
- Dane dotyczące zdrowia
- Edukacja

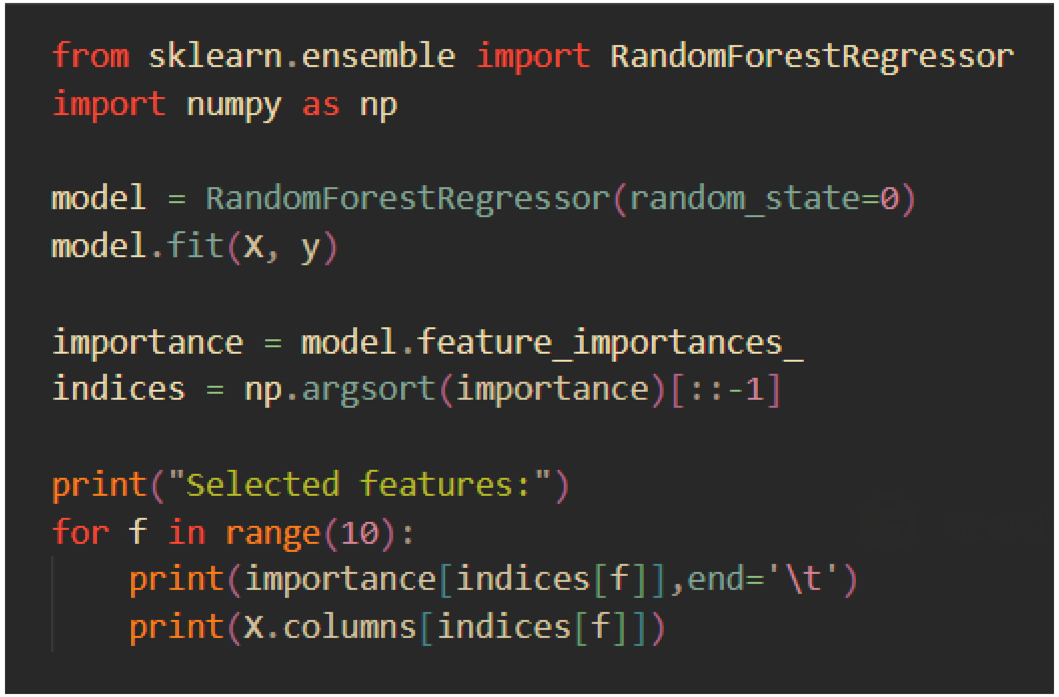
Badania korelacji

..\research\SWEETVIZ_REPORT.html

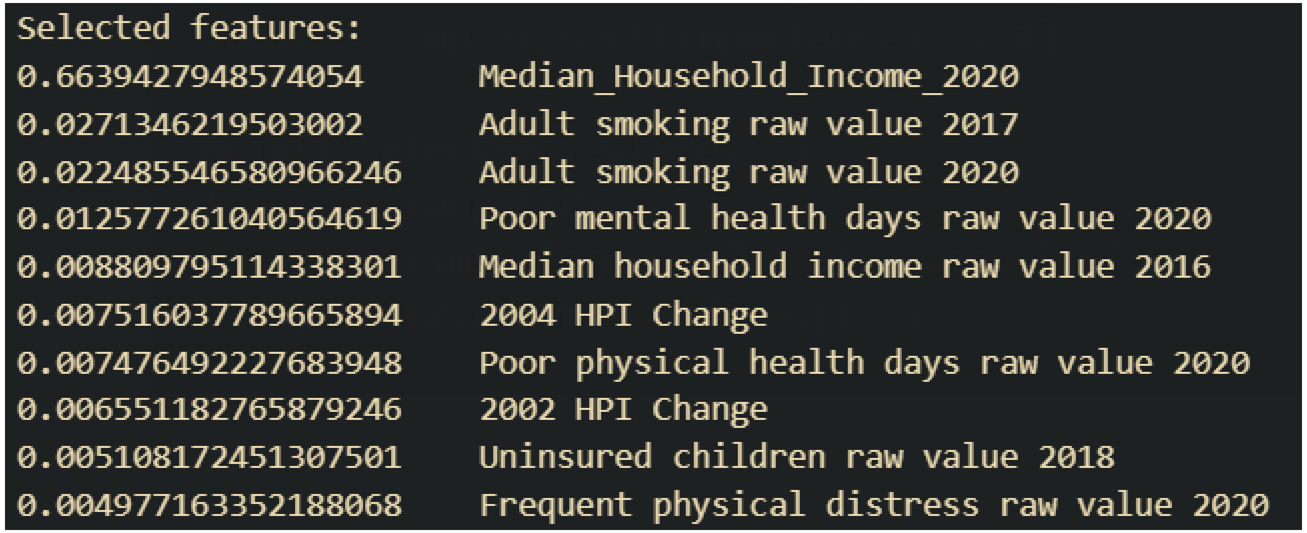
Badania wpływu zmiennych

Sklearn feature selection

RandomForrestRegressor

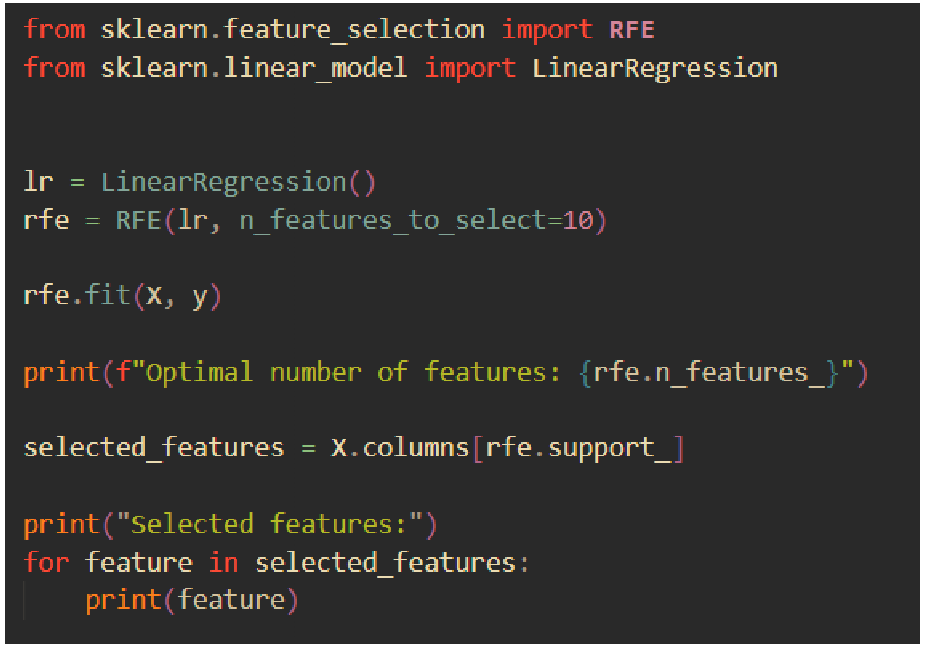


```python
from sklearn.ensemble import RandomForestRegressor
import numpy as np

model = RandomForestRegressor(random_state=0)
model.fit(X, y)

importance = model.feature_importances_
indices = np.argsort(importance)[::-1]

print("Selected features:")
for f in range(10):
    print(importance[indices[f]],end='\t')
    print(X.columns[indices[f]])
```

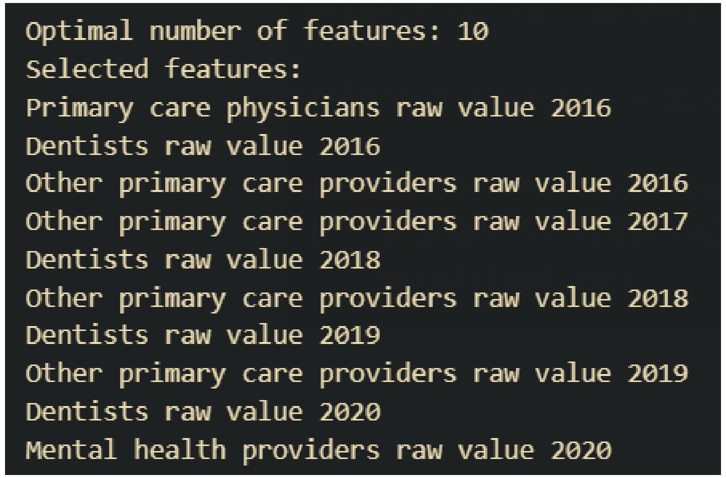


```
Selected features:
0.6639427948574054      Median_Household_Income_2020
0.0271346219503002      Adult smoking raw value 2017
0.022485546580966246    Adult smoking raw value 2020
0.012577261040564619    Poor mental health days raw value 2020
0.008809795114338301    Median household income raw value 2016
0.007516037789665894    2004 HPI Change
0.007476492227683948    Poor physical health days raw value 2020
0.006551182765879246    2002 HPI Change
0.005108172451307501    Uninsured children raw value 2018
0.004977163352188068    Frequent physical distress raw value 2020
```

RFE - recursive feature elimination

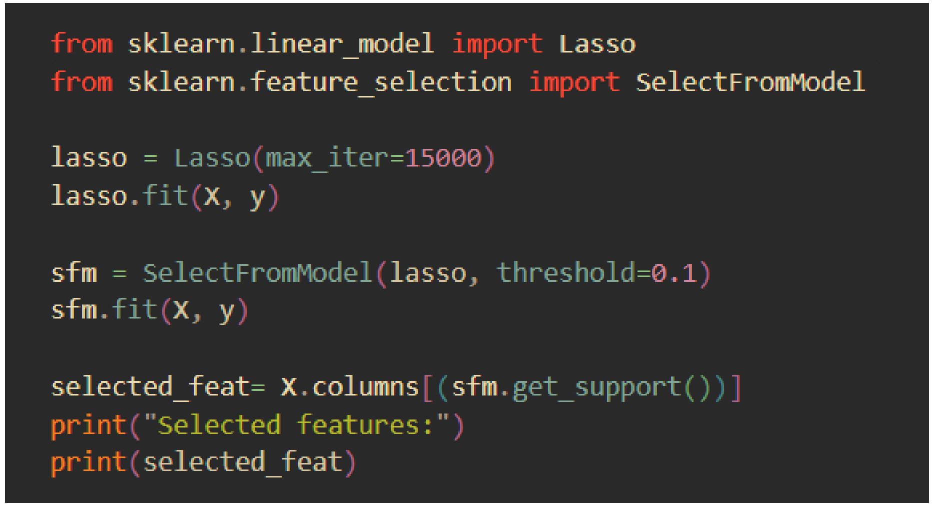


```python
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression


lr = LinearRegression()
rfe = RFE(lr, n_features_to_select=10)

rfe.fit(X, y)

print(f"Optimal number of features: {rfe.n_features_}")

selected_features = X.columns[rfe.support_]

print("Selected features:")
for feature in selected_features:
    print(feature)
```

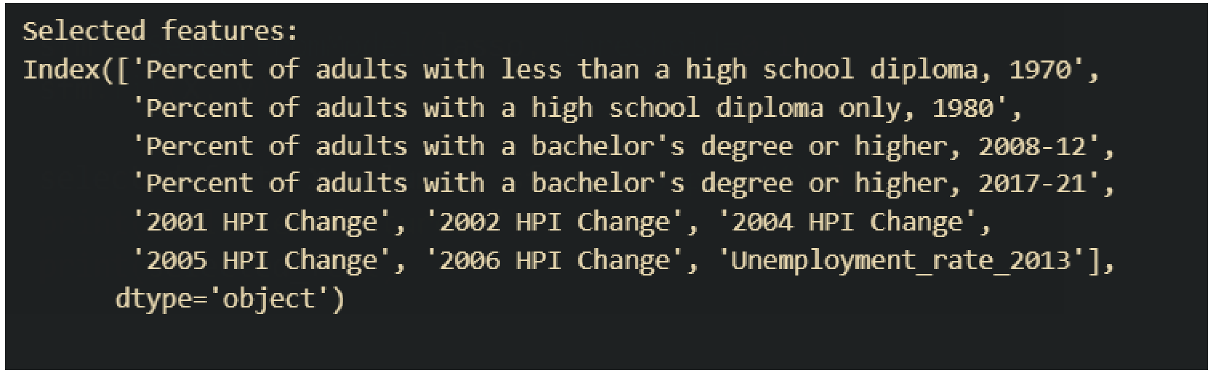


```
Optimal number of features: 10
Selected features:
Primary care physicians raw value 2016
Dentists raw value 2016
Other primary care providers raw value 2016
Other primary care providers raw value 2017
Dentists raw value 2018
Other primary care providers raw value 2018
Dentists raw value 2019
Other primary care providers raw value 2019
Dentists raw value 2020
Mental health providers raw value 2020
```

Lasso - linear model trained with L1 prior as regularizer

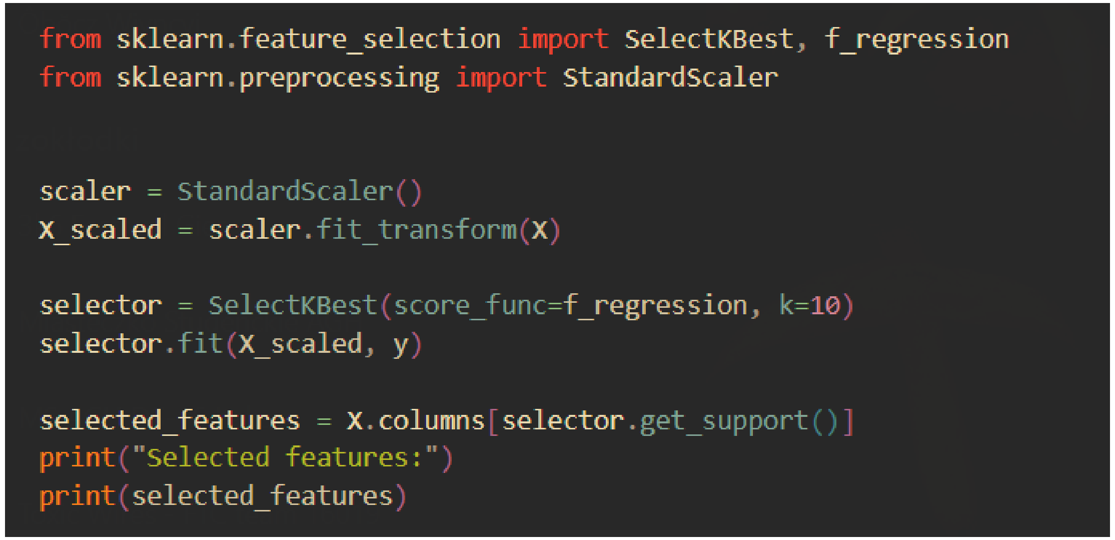


```python
from sklearn.linear_model import Lasso
from sklearn.feature_selection import SelectFromModel

lasso = Lasso(max_iter=15000)
lasso.fit(X, y)

sfm = SelectFromModel(lasso, threshold=0.1)
sfm.fit(X, y)

selected_feat= X.columns[(sfm.get_support())]
print("Selected features:")
print(selected_feat)
```

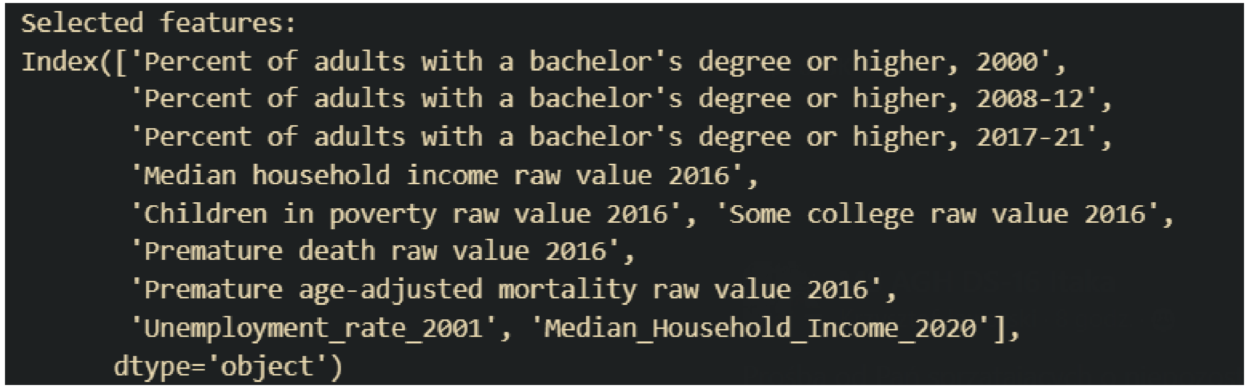


```
Selected features:
Index(['Percent of adults with less than a high school diploma, 1970',
       'Percent of adults with a high school diploma only, 1980',
       'Percent of adults with a bachelor's degree or higher, 2008-12',
       'Percent of adults with a bachelor's degree or higher, 2017-21',
       '2001 HPI Change', '2002 HPI Change', '2004 HPI Change',
       '2005 HPI Change', '2006 HPI Change', 'Unemployment_rate_2013'],
      dtype='object')
```

SelectKBest - removes all but the highest scoring features

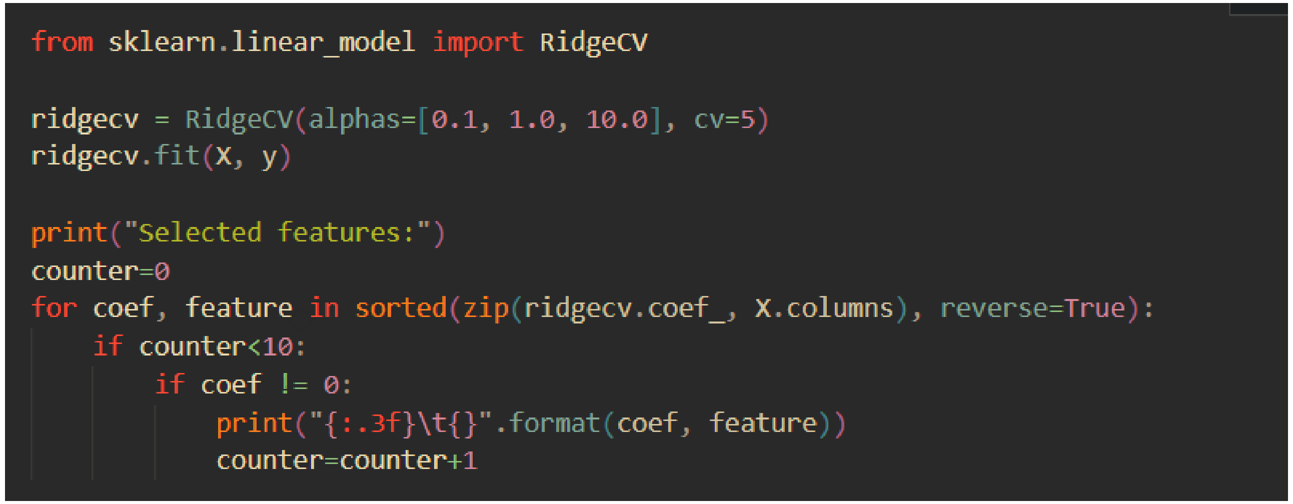


```python
from sklearn.feature_selection import SelectKBest, f_regression
from sklearn.preprocessing import StandardScaler


scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

selector = SelectKBest(score_func=f_regression, k=10)
selector.fit(X_scaled, y)

selected_features = X.columns[selector.get_support()]
print("Selected features:")
print(selected_features)
```

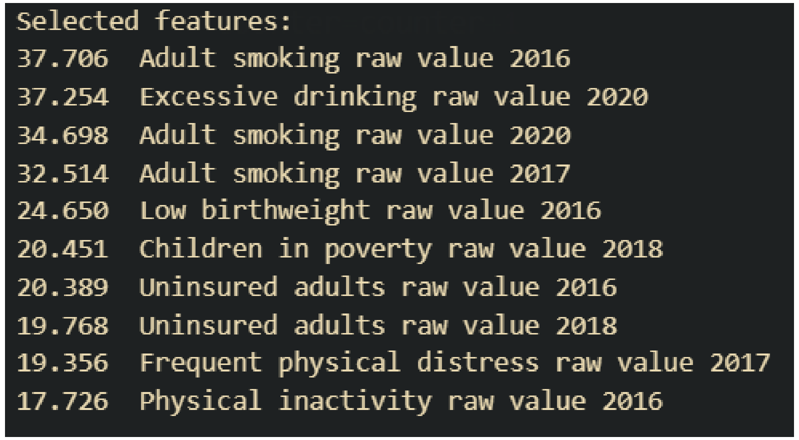


```
Selected features:
Index(['Percent of adults with a bachelor's degree or higher, 2000',
       'Percent of adults with a bachelor's degree or higher, 2008-12',
       'Percent of adults with a bachelor's degree or higher, 2017-21',
       'Median household income raw value 2016',
       'Children in poverty raw value 2016', 'Some college raw value 2016',
       'Premature death raw value 2016',
       'Premature age-adjusted mortality raw value 2016',
       'Unemployment_rate_2001', 'Median_Household_Income_2020'],
      dtype='object')
```

RidgeCV - ridge regression with built-in cross-validation



```
from sklearn.linear_model import RidgeCV

ridgecv = RidgeCV(alphas=[0.1, 1.0, 10.0], cv=5)
ridgecv.fit(X, y)

print("Selected features:")
counter=0
for coef, feature in sorted(zip(ridgecv.coef_, X.columns), reverse=True):
    if counter<10:
        if coef != 0:
            print("{:.3f}\t{}".format(coef, feature))
            counter=counter+1
```



```
Selected features:
37.706  Adult smoking raw value 2016
37.254  Excessive drinking raw value 2020
34.698  Adult smoking raw value 2020
32.514  Adult smoking raw value 2017
24.650  Low birthweight raw value 2016
20.451  Children in poverty raw value 2018
20.389  Uninsured adults raw value 2016
19.768  Uninsured adults raw value 2018
19.356  Frequent physical distress raw value 2017
17.726  Physical inactivity raw value 2016
```