Load Libraries

In [1]:
```
import plotly.express as px
import pandas as pd
import numpy as np
import sys
import plotly

import warnings
warnings.filterwarnings('ignore')
```

Define a few helper functions

In [2]:
```
def model(ckd_rate,unemp_rate,laseniors10,lalowi10,lasnap10,x1,x2,x3):


    """
    define a regression model that calculates future CKD rate
    as weighted average of past CKD rate and unemployment rate.

    More weight is given to past CKD than current unemployment

    """
    future_ckd = (x1+0.4)*laseniors10 + 0.80*ckd_rate + 0.2*unemp_rate + x2*0.2*lalowi10
    return future_ckd

def make_predictions(test_df, year, x1,x2,x3):

    """
    Make predictions on a set of counties for a given year
    in the future. Note that current available data for
    CKD prevalence is limited to 2019

    Note that predictions are more accurate when projections
    are not too far into the future
    """


    test_df['Year'] = year
    test_df = test_df.groupby(
        ['State','StateAbr','County','FIPS','FIPS3','Year']
    ).mean().reset_index()
    test_df['CkdRate'] = test_df.apply(
        lambda row: model(
            row['CkdRate'],row['unEmpRate'],row['laseniors10'],row['lalowi10'],row['lasn
        axis=1
    )

    return test_df


def combine_datasets(train, test):

    """
    combine original data with prediction into
    one dataset for convenience

    """
    dfnew = pd.concat([train,test])
```

```
    return dfnew
```

Preview CKD trends by year

According to CDC reports the prevalence of CKD among Medicare beneficiaries aged 65 or older has been increasing over the years. Data was retrieved from the Original Data source, processed using the steps highlighted below and store in github

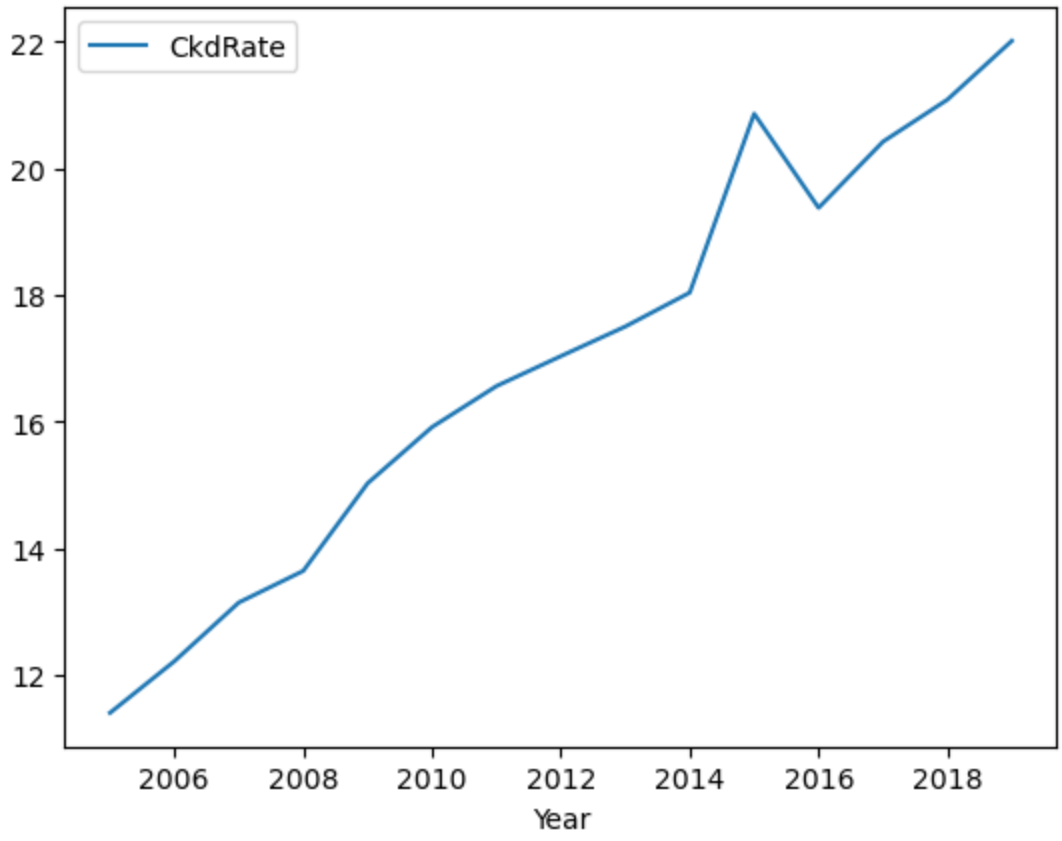
Other sources

- guide to FIPS codes
- State and County FIPS data
- State and County FIPS GIT - fips2county.tsv comes from here
- CDC Stats on CKD in adults - national prevalence about 15%
- world atlas cool project
- Choropleth guide plotly
- Color scale examples

In [3]:
```
github_loc='https://github.com/nmmarcelnv/cmsdatajam/blob/main/data/'
fname='Prevalence_of_CKD_by_US_State_and_County_by_County_2019.parquet'

ckd = pd.read_parquet(f'{github_loc}{fname}?raw=true')
ckd.columns = ['CkdRate','County','State','Year']
ckd = ckd[['County','State','Year','CkdRate']]
ckd['County'] = ckd['County'].str.upper()
ckd['State'] = ckd['State'].str.upper()

_=ckd.groupby(['Year'])[['CkdRate']].mean().plot()
```



Open source data on Geolocation and FIPS codes

We used open source FIPS data which assigns a unique ID to every country in the US. This allows us examine prevalence of CKD at a granular level by county

In [4]:
```
dtypes = {
    'StateFIPS':str, 'CountyFIPS_3':str, 'CountyName':str, 'StateName':str,
    'CountyFIPS':str,'StateAbbr':str, 'STATE_COUNTY':str
}

fips = pd.read_csv(
    'https://raw.githubusercontent.com/ChuckConnell/articles/master/fips2county.tsv',
    sep='\t',
    usecols=['CountyName','StateName','CountyFIPS','CountyFIPS_3','StateAbbr']
    ,dtype=dtypes
)
fips.columns=['FIPS3','County','State','FIPS','StateAbr']
fips = fips[['State','StateAbr','County','FIPS','FIPS3']]
fips['County'] = fips['County'].str.upper()
fips['State'] = fips['State'].str.upper()
fips.head(2)
```

Out[4]:

	State	StateAbr	County	FIPS	FIPS3
0	ALABAMA	AL	AUTAUGA	01001	001
1	ALABAMA	AL	BALDWIN	01003	003

Open source data on unemployment rate by county in the US

There are many dimensions and factors that determine the health status of individuals. We decided to incorporate porverty and employment rate in our model since there is evidence that CKD incidence is particularly high among black and other minority demographics, typically leaving under poverty level.

In [5]:
```
unemp_df = pd.read_csv(
    "https://raw.githubusercontent.com/plotly/datasets/master/fips-unemp-16.csv",dtype={
unemp_df['unemp'] = (unemp_df['unemp']/unemp_df['unemp'].max()) *(100)
unemp_df = unemp_df.rename(columns={'fips':'FIPS', 'unemp':'unEmpRate'})

unemp_df.head(2)
```

Out[5]:

	FIPS	unEmpRate
0	01001	22.553191
1	01003	22.978723

Load Data about Food Deserts and Food Insecurity

It is known that healthy diet can reduce the risk of conditions such as diabetes and hypertension, which are the primary risk factors for CKD. In particular, there is research evidence that adherence to the **DASH** (Dietary Approaches to Stop Hypertension) reduced the risk of CKD by up to 15%.

The problem however is that adherence to healthy eating habits such as the DASH diet is particularly low among black and other minority groups, which unfortunately have the highest prevalence of CKD in the US

In [6]:
```
data_atlas_link = 'https://www.ers.usda.gov/webdocs/DataFiles/80591/FoodAccessResearchAt
#data_atlas_link = '/Users/marcelvnguemaha/Downloads/FoodAccessResearchAtlasData2019.xls
```

```
usecols = [
    'State','County','PovertyRate', 'Pop2010',
    'laseniors1share','laseniors10share','laseniors20share',
    'lalowi1share','lalowi10share','lalowi20share',
    'lasnap1share','lasnap10share','lasnap20share',
    'TractWhite','TractBlack','TractAsian','TractNHOPI','TractAIAN','TractOMultir','Trac
]

dfatlas = pd.read_excel(data_atlas_link,sheet_name='Food Access Research Atlas', usecols
dfatlas.columns = [x.replace('share','') for x in dfatlas.columns]
dfatlas['State'] = dfatlas['State'].str.upper()
dfatlas['County'] = dfatlas['County'].apply(lambda x: x.upper().replace(' COUNTY','').st

dfatlas = dfatlas.groupby(['State','County']).mean().reset_index().fillna(0)

cols = ['TractWhite','TractBlack','TractAsian','TractNHOPI','TractAIAN','TractOMultir',]
for c in cols:
    dfatlas[c] = 100*dfatlas[c]/dfatlas['Pop2010']
dfatlas['TractSNAP'] = 100*dfatlas['TractSNAP']/dfatlas['OHU2010']
```

In [7]:
```
dfatlas.head()
```

Out[7]:

	State	County	Pop2010	OHU2010	PovertyRate	lalowi1	laseniors1	lasnap1	lalow
0	ALABAMA	AUTAUGA	4547.583333	1685.083333	16.130786	25.403440	8.654175	10.894014	14.522
1	ALABAMA	BALDWIN	5879.516129	2360.645161	11.845546	22.606621	13.014789	5.696397	2.895
2	ALABAMA	BARBOUR	3050.777778	1091.111111	29.299325	32.887964	9.515163	16.997808	12.765
3	ALABAMA	BIBB	5728.750000	1988.250000	12.193524	30.021718	10.926803	8.989706	0.437
4	ALABAMA	BLOUNT	6369.111111	2397.555556	14.850748	32.120502	12.496288	8.368605	0.000

5 rows × 21 columns

Combine Datasets

In [8]:
```
tmp = pd.merge(fips,ckd, on=['State','County'])
ckddf = pd.merge(tmp,unemp_df, on=['FIPS'])
df = pd.merge(ckddf, dfatlas, on=['State','County'])
#df.to_parquet('../data/DataProcessed.parquet')
```

In [9]:
```
df.head()
```

Out[9]:

	State	StateAbr	County	FIPS	FIPS3	Year	CkdRate	unEmpRate	Pop2010	OHU2010
0	ALABAMA	AL	AUTAUGA	01001	001	2005	9.95	22.553191	4547.583333	1685.083333
1	ALABAMA	AL	AUTAUGA	01001	001	2006	10.99	22.553191	4547.583333	1685.083333
2	ALABAMA	AL	AUTAUGA	01001	001	2007	14.86	22.553191	4547.583333	1685.083333
3	ALABAMA	AL	AUTAUGA	01001	001	2008	15.19	22.553191	4547.583333	1685.083333
4	ALABAMA	AL	AUTAUGA	01001	001	2009	12.20	22.553191	4547.583333	1685.083333

5 rows × 27 columns

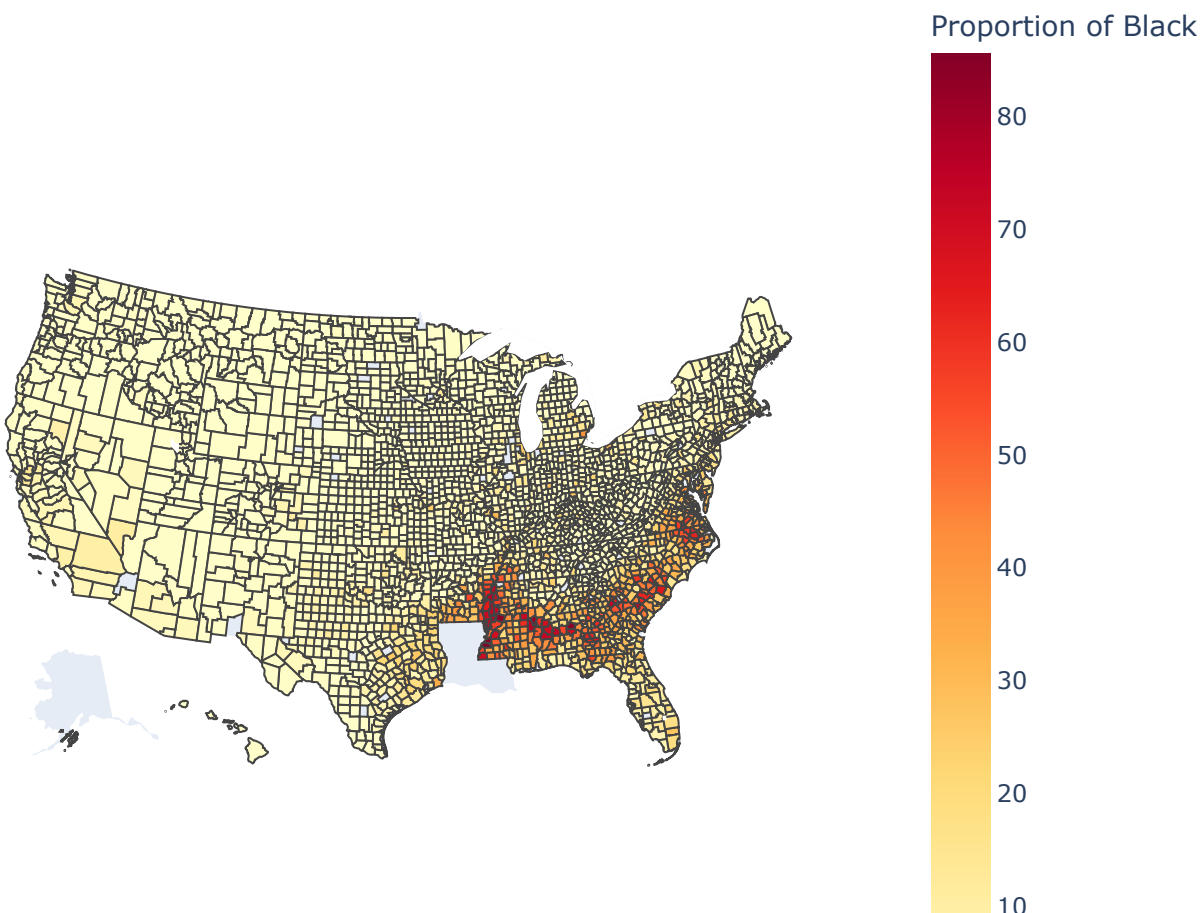
In [10]:
```
plotly.offline.init_notebook_mode(connected=True)
```

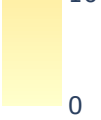
In [11]:
```
fig = px.choropleth(
    df,
    geojson="https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-f
    locations="FIPS",
    color='CkdRate',
    scope='usa',
    color_continuous_scale='YlOrRd',
    #color_continuous_scale='Viridis',
    range_color=(20, 40),
    hover_data = {'State':True,'County':True},
    labels={'CkdRate':'CKD Prevalence (%)'}
)

fig.layout.template = None
fig.show()
```

In [12]:
```
from urllib.request import urlopen
import json
with urlopen('https://raw.githubusercontent.com/plotly/datasets/master/geojson-counties-
    counties = json.load(response)


fig = px.choropleth(df,
                    geojson=counties,
                    locations='FIPS',
                    #locations=["CA", "TX", "NY"],
                    color='TractBlack',
                    color_continuous_scale='YlOrRd',
                    #locationmode='USA-states',
                    #range_color=(20, 40),
                    scope="usa",
                    hover_data = {'State':True,'County':True},
                    labels={'TractBlack':'Proportion of Black'}
                )
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```




0