process of producing aluminum from ore by cleaning and melting recycled cans. non-ferrous Singapore has an ambitious goal of becoming a zerowaste nation. The amount of waste disposed of in Singapore has increased sevenfold over the last 40 years. At this rate, Semakau Landfill, Singapore's only landfill, will run out of space by 2035. Making matters worse, Singapore has limited land for building new incineration plants or landfills. [2]: # Reading files
import pandas as pd
df1 = pd.read\_csv('wastestats.csv')
df2 = pd.read\_csv('2018\_2019\_waste.csv')
df3= pd.read\_csv('energy\_saved.csv')
Out[3]: waste\_type wa

saved from recycling

energy by

reducing or

materials from

scratch? For

aluminum can

manufacturers can skip the

waste\_type waste\_disposed\_of\_tonne total\_waste\_recycled\_tonne total\_waste\_generated\_tonne recycling\_rate year Out

**3** C&D 9700 1585700.0 1595400 0.99 2016

2 Plastics 762700 59500.0 822200 0.07 2016 0 Construction& 1440 1434 2019

 4
 Horticultural waste
 111500
 209000.0
 320500
 0.65
 2016
 2 Paper/Cardboard
 1011
 449
 2019

 3
 Plastics
 930
 37
 2019

**1** Ferrous Metal 1278 1270 2019

**4** Food 7440 136 2019

hour (kWh) and the

(barrels) by
recycling 1
metric tonne
(1000
kilogram)
per waste

amount of Unnamed: Unnamed: Unnamed: Unnamed: crude oil 1 2 3 4 5

approximately NaN NaN NaN NaN NaN NaN oil

1 NaN NaN NaN NaN NaN

2 material Plastic Glass Ferrous Non-Ferrous Metal Paper Metal

3 energy\_saved 5774 Kwh 42 Kwh 642 Kwh 14000 Kwh 4000 kWh

4 crude\_oil saved 16 barrels NaN 1.8 barrels 40 barrels 1.7 barrels

In [6]: # Lowering all letter cases
df1=df1[df1["year"]=>2015]
df1["waste\_type"] = df1["waste\_type"].str.lower()
df2["waste\_type"] = df2["waste\_type"].str.lower()
df2["waste\_type"] = df2["waste\_type"].str.lower()
df2["waste\_type"] = df2["waste\_type"].str.lower()
df2["waste\_type"] = df2["waste\_type"].str.lower()
df3 fobcuste type"].str.lower()
df3 ener save dtype=object) 'non-ferrous metal', 'used slag', 'ash & sl udge', 'glass', 'textile/leather', 'scrap tyres', 'others (stones, ceramics & rubber etc.)',

'others (stones, ceramics & rubber etc)',
'construction debris',
'non-ferrous metals', 'ash and sludge', 'pl

'others (stones, ceramic, rubber, etc.)'], dtype=object)

**2** ferrous metal 642 3 non-ferrous metal 14000 # Adding the new colum: 'total\_energy\_saved' to df1

for waste\_type in categories:
 for index, row in df1.iterrows(): if row['waste\_type'] == waste\_type:
 df1.loc[index, "total\_energy\_saved"] = row['total\_waste\_recycled\_tonne']\*df3[df3['material']==waste\_type]['energy\_saved'].values # Adding the new colum: 'total\_energy\_saved' to df2 columns= ['waste\_type','total\_waste\_generated\_tonne','total\_waste\_recycled\_tonne','year'] df2.columns = columns df2['total\_waste\_generated\_tonne'] = df2['total\_waste\_generated\_tonne']\*1000
df2['total\_waste\_recycled\_tonne'] = df2['total\_waste\_recycled\_tonne']\*1000 for waste\_type in categories: for index, row in df2.iterrows(): if row['waste\_type'] == waste\_type: df2.loc[index, "total\_energy\_saved"] = row['total\_waste\_recycled\_tonne']\*df3.loc[df3['material']==waste\_type,'energy\_saved'].values

# Creating the annual\_energy\_savings DataFrame

annual\_energy\_savings = pd.concat([annual\_energy\_savings1,annual\_energy\_savings2],ignore\_index=True).set\_index('year')
annual\_energy\_savings = annual\_energy\_savings.groupby('year').sum() annual\_energy\_savings total\_energy\_saved **2015** 3.435929e+09 **2016** 2.554433e+09 **2017** 2.470596e+09 **2018** 2.698130e+09

**2019** 2.765440e+09

# Creating the annual\_energy\_savings from df1
annual\_energy\_savings1= df1[['year','total\_energy\_saved']].sort\_values(by='year') #cheking

annual\_energy\_savings2= df2[['year','total\_energy\_saved']].sort\_values(by='year') #cheking

# Creating the annual\_energy\_savings from df2