Python in the VE

IES

VE Scripts – Benchmarking consumption data CBECS (USA)

What? In this article we look at a script that uses publicly available data (CBECS) on energy use in commercial buildings to produce a range of benchmarks by climate zone and building category.

Why? The Commercial Buildings Energy Consumption Survey (CBECS) is a national sample survey that collects information on the stock of U.S. commercial buildings, including their energy-related building characteristics and energy usage data. See https://www.eia.gov/consumption/commercial/about.php

CBECS Microdata is utilised. Download the Microdata from https://www.eia.gov/consumption/commercial/data/2018/index.php?view=microdata:

2018 CBECS microdata

File name	Format	Release date	Revised date
2018 CBECS microdata	SAS	November 2021	December 2022
2018 CBECS microdata	CSV	November 2021	December 2022
Variable and response codebook	XLS	November 2021	December 2022
User's Guide to the 2018 CBECS Public Use Microdata File	PDF	November 2021	January 2023

The Codebook and User's guide provide detail on the data utilised by the script.

Note that the records held in the CSV dataset represent CBECS data observations do not correspond to real buildings but are instead presented as an abstraction, which is to say that for every observation there is a given number of buildings that are supposed to be represented by that observation that that number is provided by the FINALWT field.

The script processes the CBECS Microdata to generate quantile data filtered by climate zone and building category.

A similar script to process RECS Microdata is also provided. Download the Microdata from https://www.eia.gov/consumption/residential/data/2020/index.php?view=microdata

```
"""
=================
CBECS data import
=================

Module description
------------------
Gets and filters CBECS CSV data to useful data fields
Converts energy use to metric
Calculates an EUI kWh/m2
Sorts by climate and building type
Determines EUI quantiles for climate and building type
Exports to xlsx files and SQL database

Notes
-----

CBECS data observations do not correspond to real buildings but are instead
presented as an abstraction, which is to say that for every observation there
is a given number of buildings that are supposed to be represented by that
observation that that number is provided by the FINALWT field. So if I sum up
all of the values of FINALWT, CBECS tells me that the number of commercial
buildings in the US is almost 6 million.

"""

import pandas as pd
import os
from pathlib import Path


# globals
CD = os.getcwd()

if __name__ == '__main__':

    # Source data file
    file_name = os.path.join(CD, 'cbecs2018_final_public.csv')

    # Add dict to dataframe
    df = pd.read_csv(file_name)
    all_columns = df.columns

    # Omit columns not in wanted list
    # Stat data?
    wanted_columns = [
                        'PUBID',
                        'REGION',
                        'CENDEV',
                        'PBA',
                        'PUBCLIM',
                        'SQFT',
                        'SQFTC',
                        'NFLOOR',
                        'YRCONC',
                        'VACANT',
                        'PBAPLUS',
                        'HDD65',
                        'CDD65',
                        'SOPANEL',
                        'SOTHERM',
                        'MFBTU',
                        'ELCNS',
                        'NGBTU',
                        'FKBTU',
                        'DHBTU',
                        'MFHTBTU',
                        'MFCLBTU',
                        'MFWTBTU',
                        'FINALWT'
                      ]
    omit_columns = [x for x in all_columns if x not in wanted_columns]

    # Remove unwanted columns
    df.drop(omit_columns, axis=1, inplace=True)

    # Rename columns
    df.rename(columns={'PUBCLIM':'Climate_zone'}, inplace=True)
    df.rename(columns={'PBA':'Building_type'}, inplace=True)
    df.rename(columns={'FINALWT':'NMBR_OF_BLDGS'}, inplace=True)

    # Fill nan with zeros
    df = df.fillna(0)

    # Conversion to metric
    # By inspection major fuels includes fossil, elec and
    # district energy i.e. MFBTU = ELCNS + NGBTU + FKBTU + DHBTU

    # MFBTU = Annual major fuels consumption (thous Btu)

    # ELCNS = Annual electricity consumption (kWh)
    # NGBTU = Annual natural gas consumption (thous Btu)
    # FKBTU = Annual fuel oil consumption (thous Btu)
    # DHBTU = Annual district heat consumption (thous Btu)

    # End use split of thermal uses

    # MFHTBTU = Major fuels heating use (thous Btu)
    # MFCLBTU = Major fuels cooling use (thous Btu)
    # MFWTBTU = Major fuels water heating use (thous Btu)

    df['AllKWH'] = df['MFBTU'] * 0.2930710694

    df['ElecKWH'] = df['ELCNS'] * 1.0
    df['GasKWH'] = df['NGBTU'] * 0.2930710694
    df['OilKWH'] = df['FKBTU'] * 0.2930710694
    df['DistrictKWH'] = df['DHBTU'] * 0.2930710694

    df['FuelsHtgKWH'] = df['MFHTBTU'] * 0.2930710694
    df['FuelsClgKWH'] = df['MFCLBTU'] * 0.2930710694
    df['FuelsDhwKWH'] = df['MFWTBTU'] * 0.2930710694

    df['M2'] = df['SQFT'] * 0.092903

    # Create an EUI in metric units that we can assess & compare
    df['EUIKWHm2'] = df['AllKWH']  / df['M2']

    # Remove any rows where the EUI was zero
    print("Before cleaning: {}".format(df.shape[0]))
    df.drop(df[df.EUIKWHm2 == 0].index, inplace=True)
    print("After cleaning: {}".format(df.shape[0]))

    # Substitute building type for labels:
    bldg_type = {
                    1:'Vacant',
                    2:'Office',
                    4:'Laboratory',
                    5:'Non-refrigerated warehouse',
                    6:'Food sales',
                    7:'Public order and safety',
                    8:'Out-patient health care',
                    11:'Refrigerated warehouse',
                    12:'Religious worship',
                    13:'Public assembly',
                    14:'Education',
                    15:'Food service',
                    16:'In-patient health care',
                    17:'Nursing',
                    18:'Lodging',
                    23:'Strip shopping center',
                    24:'Enclosed mall',
                    25:'Retail other than mall',
                    26:'Service',
                    91:'Other'
                }

    df['Building_type']= df['Building_type'].apply(lambda x: bldg_type.get(x))

    # Sort by climate zone and building type
    # Climate zones are
    # 1=Cold or very cold
    # 2=Cool
    # 3=Mixed mild
    # 4=Warm
    # 5=Hot or very hot
    # 7=Withheld for confidentiality
    df.sort_values(by=['Climate_zone','Building_type'],axis=0, inplace= True, ascending=True)

    # Determine total number of aggregated buildings by climate zone and building type
    stats1 = df.groupby(['Climate_zone','Building_type']).agg(group_number_bldgs = ('NMBR_OF_BLDGS', 'sum'))

    # Determine mean floor area by climate and building type
    stats2 = df.groupby(['Climate_zone','Building_type']).agg(group_mean_m2 = ('M2', 'mean'))

    # Determine EUI deciles by by climate PUBCLIM and building use PBA
    def qq00(x):
        return x.quantile(0.0)
    def qq25(x):
        return x.quantile(0.25)
    def qq50(x):
        return x.quantile(0.50)
    def qq75(x):
        return x.quantile(0.75)
    def qq100(x):
        return x.quantile(1.0)

    stats3 = df.groupby(['Climate_zone','Building_type']).agg(EUI_kwhm2_Q00  = ('EUIKWHm2', qq00),
                                                              EUI_kwhm2_Q25  = ('EUIKWHm2', qq25),
                                                              EUI_kwhm2_Q50  = ('EUIKWHm2', qq50),
                                                              EUI_kwhm2_Q75  = ('EUIKWHm2', qq75),
                                                              EUI_kwhm2_Q100 = ('EUIKWHm2', qq100))

    # Merge stats dataframes
    stats = pd.concat([stats1, stats2, stats3], axis=1, join='outer')

    # Export to xlsx
    df.to_excel(os.path.join(CD, 'CBECS_microdata_sorted_by_climate_and_use.xlsx'), index=False)
    stats.to_excel(os.path.join(CD, 'CBECS_microdata_stats_by_climate_and_use.xlsx'))
```

Readme comments

Metric only

Import modules

Get the script's directory

We use the __name__ conditional to ensure we only run the script when it is the main program

Create the pathname for the CSV file assuming it is placed in the same directory as the script

Read the CSV file into dataframe
Get a list of the dataframe columns

Create a list columns that are required

List comprehension – create a list of columns that will be deleted

Use df.drop to remove all unwanted columns. Inplace True changes the dataframe in place

Rename key columns that we will output with more readable labels

Handle the case of missing data – replace with zeros

Notes of data column labels and what they contain

Convert IP units to metric

Create an EUI column using the data in two other columns

Drop any rows where the EUI is zero (Index refers to rows)

Create a dict of type # : label so the output will be more readable

Use a lambda function to apply the type # : label mapping to the dataframe 'Building type' column

Sort the dataframe rows by climate zone then by Building type

Use groupby function by climate zone & Building type to group rows to determine the number of buildings for the group. This returns a new dataframe with each group as a row with mean sum NMBR ... as columns

Use groupby function by climate zone & Building type to group rows to determine the mean building floor area for the group. This returns a new dataframe with each group as a row with mean M2 as columns

Define some functions to calculate quantiles

Use groupby function by climate zone & Building type to group rows to determine the a series of quantiles for the group. This returns a new dataframe with each group as a row with the quantile as columns

The groupby dataframes have the same shape so we can combine them: axis=1 is concatenate along the column axis, outer preserves all data

Export dataframes to spreadsheets

Output ...

Climate_zone	Building_type	group_number_bldgs	group_mean_m2	EUI_kwhm2_Q00	EUI_kwhm2_Q25	EUI_kwhm2_Q50	EUI_kwhm2_Q75	EUI_kwhm2_Q100
1	Education	31128.17	14359.66949	19.49143591	137.6931702	185.5469637	233.175414	416.5747878
	Food sales	14119.7	4479.613745	433.2310301	562.9089536	688.3700756	1069.282256	1504.506217
	Food service	27949.12	476.0217001	144.9751946	312.6253343	360.1151236	771.4896716	1875.961356
	Laboratory	50.29	17279.958	685.1430248	685.1430248	685.1430248	685.1430248	685.1430248
	Lodging	7900.06	7357.9176	130.3655405	208.2009616	271.5577653	357.0630742	365.3734599
	Non-refrigerated warehouse	97700.27	9756.531686	4.073930363	45.23156217	99.88074638	175.8714523	488.2484163
	Nursing	3623.81	11625.10397	118.8207703	233.7185514	330.887168	486.7628414	733.7680057
	Office	82934.81	11683.16276	35.70019684	148.9904922	197.4652705	268.7845268	1466.70557
	Other	9342.27	10282.03953	92.69014512	157.1154146	399.0151532	975.5263389	1845.338553
	Out-patient health care	8411.5	6222.759069	91.63175204	259.2404367	294.4147717	380.856364	566.4732571
	Public assembly	23774.18	9454.384679	68.50737521	200.2801394	293.0329275	443.7411887	831.8080487
	Public order and safety	7698.38	1126.448875	133.1668534	141.1905927	168.1165255	198.0525171	215.1064326
	Refrigerated warehouse	118.55	6689.016	634.8510124	634.8510124	634.8510124	634.8510124	634.8510124
	Religious worship	30761.65	2308.63955	70.925531	105.8083988	177.8228542	226.9591901	270.4393948
	Retail other than mall	29572.63	5822.874185	50.24814551	127.0107422	188.8577	287.1378762	500.8195359
	Service	110910.81	1313.505492	4.538827106	71.0133263	128.1171859	233.8657746	1669.287896
	Strip shopping center	6999.06	7082.821494	177.637129	358.2794514	368.3715472	590.4369872	650.6889987
	Vacant	621.37	1486.448	90.35047503	90.35047503	90.35047503	90.35047503	90.35047503
2	Education	90339.77	15451.74329	38.87809429	158.291985	207.4190899	256.2748902	1016.159154
	Food sales	36268.36	4480.425835	146.5795556	463.9369715	746.6934832	864.8545347	1174.916559
	Food service	77840.9	636.4701581	52.72945752	493.4023626	972.6588992	1466.218286	3147.01023
	Laboratory	3941.81	24410.26325	133.0566502	357.1584153	637.0317911	892.7634513	1387.221956
	Lodging	36681.73	7498.626686	117.7346767	175.9535472	215.3749382	328.1752672	772.2399387
	Non-refrigerated warehouse	227984.31	13777.84006	5.91731375	69.98750941	113.2946948	166.7831867	1217.682012
	Nursing	17504.29	12117.3899	164.03439	234.3683244	396.1343534	536.5193725	804.5551829
	Office	238302.29	13692.99788	42.83254188	139.1831653	196.9079927	265.3011954	689.7150347
	Other	31698.94	6885.854231	9.614864643	93.78399503	252.3388039	407.1223777	1834.258823
	Out-patient health care	24751.63	8888.317652	90.68979084	178.785781	268.5331122	373.6772853	625.0911569
	Public assembly	157207.35	8808.742981	1.432184811	128.2635017	237.4509256	416.1280259	685.3007105
	Public order and safety	24014.04	8022.793403	134.3919324	205.1558221	259.0760153	353.3953535	458.5104239
	Refrigerated warehouse	325.41	25399.69878	223.0830705	295.1780746	408.1319619	456.8554089	463.2556251
	Religious worship	114923.56	2044.485353	11.93553063	66.62905287	121.2012988	159.1716385	411.0945245
	Retail other than mall	97334.19	8462.302013	15.86344746	113.8857506	244.9854593	371.8739672	624.5660169
	Service	329149.53	1747.679426	1.53234311	60.49904929	130.4591053	203.5733846	1024.388656
	Strip shopping center	45309.22	7211.02744	237.7045809	357.6182851	460.027856	507.3276931	677.7837062
	Vacant	57678.66	9713.589294	2.436046118	43.86815576	93.20408837	184.6512033	322.7716241
3	Education	64865.92	12932.98058	49.93493402	150.3505767	204.47831	251.449223	619.0783878