

### 3 - Cleaning land use data, aligning and merging census data and land use data

November 29, 2023

In this notebook I shall load and clean the land use data. I shall then ensure the census data and land use data is harmonized as a number of Local Authorities changed in the time between the two datasets were published. I will then merge the dataframes and use the combined data to create calculate population density per Local Authority.

```
[1]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import re
```

```
[2]: #load land use data
land_use_df = pd.read_excel('Data/Land_Use_England_2018.xlsx',
    sheet_name='P400b', skiprows=lambda x: x in [0,1])
col_names = land_use_df.columns.values.tolist()
land_use_df.head(20)
```

```
[2]:      Unnamed: 0      Unnamed: 1      Unnamed: 2      Unnamed: 3 \
0      NaN      NaN      NaN      Community Buildings
1      NaN      NaN      NaN      NaN
2      England      NaN      93491.087766
3      NaN      NaN      NaN      NaN
4      ONS Code      Region      NaN      NaN
5      E12000001      North East      NaN      5305.544389
6      E12000002      North West      NaN      12046.474266
7      E12000003      Yorkshire and The Humber      NaN      9300.505675
8      E12000004      East Midlands      NaN      8557.350873
9      E12000005      West Midlands      NaN      10434.317971
10     E12000006      East of England      NaN      12088.302047
11     E12000007      London      NaN      7573.471722
12     E12000008      South East      NaN      18202.348617
13     E12000009      South West      NaN      9982.772207
14      NaN      NaN      NaN      NaN
15      NaN      NaN      NaN      NaN
16     ONS Code      Local Authority Name      NaN      NaN
17     E07000223      Adur      NaN      136.174483
18     E07000026      Allerdale      NaN      212.544487
```

19	E07000032	Amber Valley	NaN	201.878538
Unnamed: 4 Community Service Defence Buildings \				
0	Leisure and recreational buildings	Total	Total	
1	NaN	NaN	NaN	
2	584.034064	94075.121831	124.920265	
3	NaN	NaN	NaN	
4	NaN	NaN	NaN	
5	43.087363	5348.631753	3.416098	
6	72.643596	12119.117862	4.093185	
7	62.84281	9363.348485	14.774515	
8	47.276395	8604.627269	5.641154	
9	91.525782	10525.843753	8.474661	
10	65.115192	12153.417238	13.78367	
11	62.143074	7635.614796	4.021424	
12	81.458467	18283.807084	28.329912	
13	57.941384	10040.713592	42.385646	
14	NaN	NaN	NaN	
15	NaN	NaN	NaN	
16	NaN	NaN	NaN	
17	0.273077	136.447561	0	
18	1.315606	213.860093	0.012543	
19	2.515884	204.394422	0.075455	
	Unnamed: 7	Unnamed: 8	Unnamed: 9	... Forest, open land and water \
0	Industry	Offices	Retail	...
1	NaN	NaN	NaN	...
2	18657.980659	7129.774265	11277.09771	...
3	NaN	NaN	NaN	...
4	NaN	NaN	NaN	...
5	1825.777728	374.894628	674.181709	...
6	3042.419359	1179.570717	1859.270599	...
7	3445.211559	522.541923	1066.332916	...
8	2101.052211	826.020658	1171.331674	...
9	2064.183642	727.134699	1348.286577	...
10	1712.664234	755.218586	1173.676185	...
11	704.818756	833.216726	1120.527861	...
12	2152.76703	1169.99448	1610.329879	...
13	1609.08614	741.181848	1253.16031	...
14	NaN	NaN	NaN	...
15	NaN	NaN	NaN	...
16	NaN	NaN	NaN	...
17	13.897324	3.504225	9.323311	...
18	54.399175	9.273168	24.714412	...
19	27.627493	7.233384	75.519006	...

Outdoor recreation Residential Gardens Undeveloped land Unnamed: 41 \

0	Total	Total	Total	NaN
1	NaN	NaN	NaN	NaN
2	289759.684248	633383.384814	104139.452472	NaN
3	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN
5	14086.319795	22980.450424	6934.839141	NaN
6	35691.281031	68676.203081	14071.398648	NaN
7	25838.666962	57981.233344	12256.70608	NaN
8	27311.589053	62674.031408	10083.289717	NaN
9	31466.023778	70111.705214	10855.571891	NaN
10	40721.709506	91961.521215	13418.017826	NaN
11	21625.582654	37899.116695	7589.485655	NaN
12	61653.348217	133924.012887	18465.413781	NaN
13	31365.163255	87175.110546	10464.729733	NaN
14	NaN	NaN	NaN	NaN
15	NaN	NaN	NaN	NaN
16	NaN	NaN	NaN	NaN
17	144.547655	569.287584	61.048666	NaN
18	1901.033558	1563.679759	330.814803	NaN
19	1066.301073	1848.897813	295.145568	NaN

	Non-developed Use	Unnamed: 43	Vacant	Unnamed: 45	Unnamed: 46
0	Total	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN
2	12128884.31023	NaN	22979.610234	NaN	1.325743e+07
3	NaN	NaN	NaN	NaN	NaN
4	NaN	NaN	NaN	NaN	NaN
5	805709.598981	NaN	1321.377965	NaN	8.673976e+05
6	1348907.426965	NaN	2186.787003	NaN	1.485416e+06
7	1435214.499328	NaN	1906.812577	NaN	1.555162e+06
8	1456146.450134	NaN	2854.911589	NaN	1.580851e+06
9	1181989.690094	NaN	2311.611821	NaN	1.300313e+06
10	1800025.543945	NaN	4580.942499	NaN	1.953590e+06
11	95381.66985	NaN	987.117964	NaN	1.594623e+05
12	1748817.517307	NaN	4329.855224	NaN	1.934530e+06
13	2256691.913444	NaN	2500.193592	NaN	2.420712e+06
14	NaN	NaN	NaN	NaN	NaN
15	NaN	NaN	NaN	NaN	NaN
16	NaN	NaN	NaN	NaN	NaN
17	3418.435055	NaN	4.535912	NaN	4.364506e+03
18	127158.413179	NaN	51.458655	NaN	1.320548e+05
19	23861.418671	NaN	75.472277	NaN	2.654366e+04

[20 rows x 47 columns]

```
[3]: #Reconfigure land use dataframe and remove unnecessary columns
```

```

land_use_df = land_use_df[['Unnamed: 0', 'Unnamed: 1', 'Developed Use ', '
↳Non-developed Use', 'Unnamed: 46']]
land_use_df.rename(columns={'Unnamed: 0': 'ONS Code', 'Unnamed: 1': 'Local_
↳Authority Name', 'Unnamed: 46': 'Total Land Area' }, inplace=True)
land_use_df.drop([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,16],
↳axis=0, inplace = True)
print(land_use_df.shape)
land_use_df.head()

```

(326, 5)

```

[3]:      ONS Code Local Authority Name Developed Use  Non-developed Use  \
17  E07000223                Adur      941.534725      3418.435055
18  E07000026            Allerdale    4844.924035    127158.413179
19  E07000032        Amber Valley    2606.772331     23861.418671
20  E07000224                Arun    2791.129278     19613.024334
21  E07000170            Ashfield    2028.578319     8881.797678

      Total Land Area
17      4364.505691
18    132054.795869
19     26543.663279
20     22450.444783
21     10955.544524

```

```

[4]: #Remove white space from column name
land_use_df.rename(columns={"Developed Use ": "Developed Use"}, inplace = True)

```

The two datasets I am using have some differences in the Local Authorities included. The census data is from 2021 whereas the land use data was published in 2018. The census data included a list of 7 Local Authorities which had been newly created in 2020 combining several previous Local Authorities.

In order to merge the two datasets and perform meaningful data analysis I would like to include these new Local Authorities so I am going to aggregate and sum the land use data for the Local Authorities which no longer exist to create rows for the Local Authorities created in 2020 so that the land use data and the census data match.

```

[5]: def combine_las_to_remove(remove_list, land_use_df):
      """Creates a new dataframe with data for LAs that have been combined into_
↳one new LA."""
      #df = pd.DataFrame(columns=['ONS Code', 'Local Authority Name', 'Developed_
↳Use', 'Non-developed Use', 'Total Land Area'])
      empty_list = []
      for index, row in land_use_df.iterrows():
          if row['Local Authority Name'] in remove_list:
              md = land_use_df.loc[index, :].values.flatten().tolist()
              empty_list.append(md)

```

```

    add_df = pd.DataFrame(empty_list, columns=['ONS Code', 'Local Authority_
↳Name', 'Developed Use', 'Non-developed Use', 'Total Land Area'])
    print(add_df)
    return add_df

def sum_las_to_remove(add_df, new_la_code, new_la_name):
    """Takes df and totals all numeric columns to create new row with data for_
↳new combined LA"""
    add_df.loc['Column_Total'] = add_df.sum(numeric_only=True, axis=0)
    add_df.loc['Column_Total', 'Local Authority Name'] = new_la_name
    add_df.loc['Column_Total', 'ONS Code'] = new_la_code
    return add_df

def add_new_la_delete_abolished_las(land_use_df, sum_df, remove_list):
    """Adds new combined LA row to land_use_df and removes rows which refer to_
↳those former
    LAs which no longer exist"""
    land_use_df = pd.concat([land_use_df, sum_df.tail(1)], axis=0)
    land_use_df = land_use_df[land_use_df['Local Authority Name'].
↳isin(remove_list) == False]
    return land_use_df

```

```

[6]: #Create new LA North Northamptonshire
northamptonshire_remove = ['Corby', 'East Northamptonshire', 'Kettering',
↳'Wellingborough']
nn = sum_las_to_remove(combine_las_to_remove(northamptonshire_remove,
↳land_use_df), 'E06000061', 'North Northamptonshire')

land_use_df = add_new_la_delete_abolished_las(land_use_df, nn,
↳northamptonshire_remove)

```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use	\
0	E07000150	Corby	1465.991360	6377.268316	
1	E07000152	East Northamptonshire	4791.998684	46134.578032	
2	E07000153	Kettering	2163.510188	21092.661844	
3	E07000156	Wellingborough	1831.957328	14397.351187	
	Total Land Area				
0	8027.939668				
1	50978.586697				
2	23348.916855				
3	16303.622392				

```
[7]: #Create new LA West Northamptonshire
west_northamptonshire_remove = ['Daventry', 'Northampton', 'South_
↳Northamptonshire']
wn = sum_las_to_remove(combine_las_to_remove(west_northamptonshire_remove,
↳land_use_df), 'E06000062', 'West Northamptonshire')

land_use_df = add_new_la_delete_abolished_las(land_use_df, wn,
↳west_northamptonshire_remove)
land_use_df
```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use \
0	E07000151	Daventry	3457.307159	62792.502320
1	E07000154	Northampton	3064.873298	4952.512582
2	E07000155	South Northamptonshire	3461.975891	59819.154564

	Total Land Area
0	66559.824011
1	8076.167194
2	63399.917721

```
[7]:
```

	ONS Code	Local Authority Name	Developed Use \
17	E07000223	Adur	941.534725
18	E07000026	Allerdale	4844.924035
19	E07000032	Amber Valley	2606.772331
20	E07000224	Arun	2791.129278
21	E07000170	Ashfield	2028.578319
...	...	...	...
340	E07000128	Wyre	2571.244346
341	E07000239	Wyre Forest	1937.962717
342	E06000014	York	3664.585248
Column_Total	E06000061	North Northamptonshire	10253.457561
Column_Total	E06000062	West Northamptonshire	9984.156347

	Non-developed Use	Total Land Area
17	3418.435055	4364.505691
18	127158.413179	132054.795869
19	23861.418671	26543.663279
20	19613.024334	22450.444783
21	8881.797678	10955.544524
...	...	...
340	29877.321859	32495.519662
341	17580.747614	19534.968482
342	23503.199857	27198.689875
Column_Total	88001.859379	98659.065612
Column_Total	127564.169465	138035.908925

[321 rows x 5 columns]

```
[8]: #Create new LA East Suffolk
east_suffolk_remove = ['Suffolk Coastal', 'Waveney']

es = sum_las_to_remove(combine_las_to_remove(east_suffolk_remove, land_use_df),
↳ 'E07000244', 'East Suffolk')

land_use_df = add_new_la_delete_abolished_las(land_use_df, es,
↳ east_suffolk_remove)
land_use_df
```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use \
0	E07000205	Suffolk Coastal	5221.292686	86760.638200
1	E07000206	Waveney	2903.206311	34247.957062

	Total Land Area
0	92032.120098
1	37186.018738

```
[8]:
```

	ONS Code	Local Authority Name	Developed Use \
17	E07000223	Adur	941.534725
18	E07000026	Allerdale	4844.924035
19	E07000032	Amber Valley	2606.772331
20	E07000224	Arun	2791.129278
21	E07000170	Ashfield	2028.578319
...	...	...	...
341	E07000239	Wyre Forest	1937.962717
342	E06000014	York	3664.585248
Column_Total	E06000061	North Northamptonshire	10253.457561
Column_Total	E06000062	West Northamptonshire	9984.156347
Column_Total	E07000244	East Suffolk	8124.498998

	Non-developed Use	Total Land Area
17	3418.435055	4364.505691
18	127158.413179	132054.795869
19	23861.418671	26543.663279
20	19613.024334	22450.444783
21	8881.797678	10955.544524
...	...	...
341	17580.747614	19534.968482
342	23503.199857	27198.689875
Column_Total	88001.859379	98659.065612
Column_Total	127564.169465	138035.908925
Column_Total	121008.595262	129218.138837

[320 rows x 5 columns]

```
[9]: #Create new LA West Suffolk
west_suffolk_remove = ['Forest Heath', 'St Edmundsbury']

ws = sum_las_to_remove(combine_las_to_remove(west_suffolk_remove, land_use_df),
↳ 'E07000245', 'West Suffolk')

land_use_df = add_new_la_delete_abolished_las(land_use_df, ws,
↳ west_suffolk_remove)
land_use_df
```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use \
0	E07000201	Forest Heath	3314.602865	34398.542398
1	E07000204	St Edmundsbury	4225.520842	61359.313296

	Total Land Area
0	37753.030274
1	65695.866382

```
[9]:
```

	ONS Code	Local Authority Name	Developed Use \
17	E07000223	Adur	941.534725
18	E07000026	Allerdale	4844.924035
19	E07000032	Amber Valley	2606.772331
20	E07000224	Arun	2791.129278
21	E07000170	Ashfield	2028.578319
...	...	...	...
342	E06000014	York	3664.585248
Column_Total	E06000061	North Northamptonshire	10253.457561
Column_Total	E06000062	West Northamptonshire	9984.156347
Column_Total	E07000244	East Suffolk	8124.498998
Column_Total	E07000245	West Suffolk	7540.123708

	Non-developed Use	Total Land Area
17	3418.435055	4364.505691
18	127158.413179	132054.795869
19	23861.418671	26543.663279
20	19613.024334	22450.444783
21	8881.797678	10955.544524
...	...	...
342	23503.199857	27198.689875
Column_Total	88001.859379	98659.065612
Column_Total	127564.169465	138035.908925
Column_Total	121008.595262	129218.138837
Column_Total	95757.855695	103448.896657

[319 rows x 5 columns]



```
[10]: #Create new LA Buckinghamshire
buckinghamshire_remove = ['Aylesbury Vale', 'Chiltern', 'South Bucks',
↳ 'Wycombe']

bu = sum_las_to_remove(combine_las_to_remove(buckinghamshire_remove,
↳ land_use_df), 'E06000060', 'Buckinghamshire')

land_use_df = add_new_la_delete_abolished_las(land_use_df, bu,
↳ buckinghamshire_remove)
land_use_df
```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use \
0	E07000004	Aylesbury Vale	5614.533011	84446.918097
1	E07000005	Chiltern	1762.424950	17854.434945
2	E07000006	South Bucks	1947.908308	12147.920232
3	E07000007	Wycombe	3035.642362	29382.736449

	Total Land Area
0	90271.305526
1	19634.287278
2	14126.190591
3	32456.862300

```
[10]:
```

	ONS Code	Local Authority Name	Developed Use \
17	E07000223	Adur	941.534725
18	E07000026	Allerdale	4844.924035
19	E07000032	Amber Valley	2606.772331
20	E07000224	Arun	2791.129278
21	E07000170	Ashfield	2028.578319
...	...	...	...
Column_Total	E06000061	North Northamptonshire	10253.457561
Column_Total	E06000062	West Northamptonshire	9984.156347
Column_Total	E07000244	East Suffolk	8124.498998
Column_Total	E07000245	West Suffolk	7540.123708
Column_Total	E06000060	Buckinghamshire	12360.508631

	Non-developed Use	Total Land Area
17	3418.435055	4364.505691
18	127158.413179	132054.795869
19	23861.418671	26543.663279
20	19613.024334	22450.444783
21	8881.797678	10955.544524
...	...	...
Column_Total	88001.859379	98659.065612
Column_Total	127564.169465	138035.908925
Column_Total	121008.595262	129218.138837
Column_Total	95757.855695	103448.896657

Column\_Total      143832.009723      156488.645695

[316 rows x 5 columns]

```
[11]: #Create new LA Bournemouth, Christchurch and Poole
bournemouth_remove = ['Christchurch', 'Bournemouth', 'Poole']
bo = sum_las_to_remove(combine_las_to_remove(bournemouth_remove, land_use_df),
↳ 'E06000058', 'Bournemouth, Christchurch and Poole')

land_use_df = add_new_la_delete_abolished_las(land_use_df, bo,
↳ bournemouth_remove)
land_use_df
```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use \
0	E06000028	Bournemouth	1811.915546	2918.119748
1	E07000048	Christchurch	1234.350128	3931.935215
2	E06000029	Poole	2128.148876	5325.401353

	Total Land Area
0	4734.429178
1	5169.241925
2	7475.605457

```
[11]:
```

	ONS Code	Local Authority Name	Developed Use \
17	E07000223	Adur	941.534725
18	E07000026	Allerdale	4844.924035
19	E07000032	Amber Valley	2606.772331
20	E07000224	Arun	2791.129278
21	E07000170	Ashfield	2028.578319
...	...	...	...
Column_Total	E06000062	West Northamptonshire	9984.156347
Column_Total	E07000244	East Suffolk	8124.498998
Column_Total	E07000245	West Suffolk	7540.123708
Column_Total	E06000060	Buckinghamshire	12360.508631
Column_Total	E06000058	Bournemouth, Christchurch and Poole	5174.41455

	Non-developed Use	Total Land Area
17	3418.435055	4364.505691
18	127158.413179	132054.795869
19	23861.418671	26543.663279
20	19613.024334	22450.444783
21	8881.797678	10955.544524
...	...	...
Column_Total	127564.169465	138035.908925
Column_Total	121008.595262	129218.138837
Column_Total	95757.855695	103448.896657
Column_Total	143832.009723	156488.645695

Column\_Total          12175.456316          17379.276560

[314 rows x 5 columns]

```
[12]: #Create new LA Dorset
dorset_remove = ['East Dorset', 'North Dorset', 'Purbeck', 'West Dorset',
↳ 'Weymouth and Portland']
do = sum_las_to_remove(combine_las_to_remove(dorset_remove, land_use_df),
↳ 'E06000059', 'Dorset')

land_use_df = add_new_la_delete_abolished_las(land_use_df, do, dorset_remove)
land_use_df
```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use \
0	E07000049	East Dorset	2568.760317	32834.893333
1	E07000050	North Dorset	2871.859823	58023.678570
2	E07000051	Purbeck	2505.263248	40215.201047
3	E07000052	West Dorset	4808.103449	103802.947421
4	E07000053	Weymouth and Portland	953.383341	3334.330291

	Total Land Area
0	35424.549588
1	60905.907559
2	42732.640241
3	108674.947336
4	4298.340078

```
[12]:
```

	ONS Code	Local Authority Name	Developed Use \
17	E07000223	Adur	941.534725
18	E07000026	Allerdale	4844.924035
19	E07000032	Amber Valley	2606.772331
20	E07000224	Arun	2791.129278
21	E07000170	Ashfield	2028.578319
...	...	...	...
Column_Total	E07000244	East Suffolk	8124.498998
Column_Total	E07000245	West Suffolk	7540.123708
Column_Total	E06000060	Buckinghamshire	12360.508631
Column_Total	E06000058	Bournemouth, Christchurch and Poole	5174.41455
Column_Total	E06000059	Dorset	13707.370179

	Non-developed Use	Total Land Area
17	3418.435055	4364.505691
18	127158.413179	132054.795869
19	23861.418671	26543.663279
20	19613.024334	22450.444783
21	8881.797678	10955.544524
...	...	...

Column_Total	121008.595262	129218.138837
Column_Total	95757.855695	103448.896657
Column_Total	143832.009723	156488.645695
Column_Total	12175.456316	17379.276560
Column_Total	238211.050661	252036.384802

[310 rows x 5 columns]

```
[13]: #Create new LA Somerset West and Taunton
somerst_remove = ['Taunton Deane', 'West Somerset']
so = sum_las_to_remove(combine_las_to_remove(somerst_remove, land_use_df),
↳ 'E07000246', 'Somerset West and Taunton')

land_use_df = add_new_la_delete_abolished_las(land_use_df, so, somerst_remove)
land_use_df = land_use_df.reset_index()
land_use_df = land_use_df.drop(['index'], axis=1)
land_use_df
```

	ONS Code	Local Authority Name	Developed Use	Non-developed Use \
0	E07000190	Taunton Deane	3396.448087	42788.369460
1	E07000191	West Somerset	2457.754874	72294.996021

	Total Land Area
0	46250.105992
1	74755.219251

```
[13]: ONS Code Local Authority Name Developed Use \
0 E07000223 Adur 941.534725
1 E07000026 Allerdale 4844.924035
2 E07000032 Amber Valley 2606.772331
3 E07000224 Arun 2791.129278
4 E07000170 Ashfield 2028.578319
.. ...
304 E07000245 West Suffolk 7540.123708
305 E06000060 Buckinghamshire 12360.508631
306 E06000058 Bournemouth, Christchurch and Poole 5174.41455
307 E06000059 Dorset 13707.370179
308 E07000246 Somerset West and Taunton 5854.202961
```

	Non-developed Use	Total Land Area
0	3418.435055	4364.505691
1	127158.413179	132054.795869
2	23861.418671	26543.663279
3	19613.024334	22450.444783
4	8881.797678	10955.544524
..	...	...
304	95757.855695	103448.896657
305	143832.009723	156488.645695

```

306      12175.456316      17379.276560
307      238211.050661      252036.384802
308      115083.365482      121005.325243

```

[309 rows x 5 columns]

Now I want to calculate the percentage of developed land in each Local Authority as I am intending to use this as one of the measure of how urbanised a Local Authority is.

```

[14]: #Calculate percentage of deveopled land in each Local Authority
land_use_df['% Developed Land'] = land_use_df['Developed Use']/
    ↳land_use_df['Total Land Area']*100
decimals = 2
land_use_df['% Developed Land'] = land_use_df['% Developed Land'].apply(lambda_
    ↳x: round(x, decimals))
land_use_df

```

```

[14]:      ONS Code      Local Authority Name Developed Use \
0      E07000223      Adur      941.534725
1      E07000026      Allerdale      4844.924035
2      E07000032      Amber Valley      2606.772331
3      E07000224      Arun      2791.129278
4      E07000170      Ashfield      2028.578319
..      ...      ...
304      E07000245      West Suffolk      7540.123708
305      E06000060      Buckinghamshire      12360.508631
306      E06000058      Bournemouth, Christchurch and Poole      5174.41455
307      E06000059      Dorset      13707.370179
308      E07000246      Somerset West and Taunton      5854.202961

```

```

      Non-developed Use      Total Land Area      % Developed Land
0      3418.435055      4364.505691      21.57
1      127158.413179      132054.795869      3.67
2      23861.418671      26543.663279      9.82
3      19613.024334      22450.444783      12.43
4      8881.797678      10955.544524      18.52
..      ...      ...
304      95757.855695      103448.896657      7.29
305      143832.009723      156488.645695      7.90
306      12175.456316      17379.276560      29.77
307      238211.050661      252036.384802      5.44
308      115083.365482      121005.325243      4.84

```

[309 rows x 6 columns]

Now I shall load the age\_group\_proportions\_census data I produced and saved earlier.

```
[15]: census_age_proportion_df = pd.read_csv('age_group_proportions_census_df.csv',
      ↪index_col=0)
      census_age_proportion_df
```

```
[15]:
```

	Area name	Area code	Region	Proportion pop aged 0-19 \
0	County Durham	E06000047	North East	21.796591
1	Darlington	E06000005	North East	22.541744
2	Hartlepool	E06000001	North East	23.401950
3	Middlesbrough	E06000002	North East	25.573315
4	Northumberland	E06000057	North East	19.962570
..	...	...	...	...
326	Caerphilly	W06000018	Wales	22.797044
327	Blaenau Gwent	W06000019	Wales	21.375187
328	Torfaen	W06000020	Wales	22.535211
329	Monmouthshire	W06000021	Wales	19.677419
330	Newport	W06000022	Wales	24.436090

  

	Proportion pop aged 20-64	Proportion pop aged 65+ \
0	56.904808	21.356062
1	56.957328	20.686456
2	56.663055	19.718310
3	57.748436	16.817234
4	54.585153	25.452277
..	...	...
326	57.248437	20.011370
327	58.295964	20.328849
328	56.446371	20.801733
329	54.623656	25.913978
330	58.583960	17.042607

  

	Proportion pop aged 0-34	All persons
0	39.590117	522100
1	39.888683	107800
2	41.170098	92300
3	46.977067	143900
4	34.591391	320600
..	...	...
326	40.648096	175900
327	40.209268	66900
328	40.628386	92300
329	33.763441	93000
330	44.674185	159600

[331 rows x 8 columns]

The census data includes Wales whereas the land use data does not. So to combine the two dataframes I shall drop the rows with Welsh local authorities as I have no land use data for them.

```
[16]: country = ['Wales']

#Drop any rows from dataframe which include Wales in the region column
census_age_proportion_df =
    ↪census_age_proportion_df[census_age_proportion_df['Region'].isin(country) ==
    ↪False]
census_age_proportion_df
```

```
[16]:
```

	Area name	Area code	Region \
0	County Durham	E06000047	North East
1	Darlington	E06000005	North East
2	Hartlepool	E06000001	North East
3	Middlesbrough	E06000002	North East
4	Northumberland	E06000057	North East
..	...	...	...
304	Tewkesbury	E07000083	South West
305	Mendip	E07000187	South West
306	Sedgemoor	E07000188	South West
307	Somerset West and Taunton	E07000246	South West
308	South Somerset	E07000189	South West

	Proportion pop aged 0-19	Proportion pop aged 20-64 \
0	21.796591	56.904808
1	22.541744	56.957328
2	23.401950	56.663055
3	25.573315	57.748436
4	19.962570	54.585153
..	...	...
304	22.233930	55.321391
305	21.705426	54.521964
306	21.212121	54.864434
307	20.648030	53.748412
308	20.555877	53.966416

	Proportion pop aged 65+	Proportion pop aged 0-34	All persons
0	21.356062	39.590117	522100
1	20.686456	39.888683	107800
2	19.718310	41.170098	92300
3	16.817234	46.977067	143900
4	25.452277	34.591391	320600
..	...	...	...
304	22.233930	38.566913	94900
305	23.772610	36.434109	116100
306	23.763955	37.559809	125400
307	25.603558	36.213469	157400
308	25.535611	36.595252	172700

[309 rows x 8 columns]

Now I shall use sets to check whether the Local Authorities in both my census dataframe and my land use dataframe correspond exactly with one another.

```
[17]: census_list = census_age_proportion_df['Area name'].tolist()
land_use_list = land_use_df['Local Authority Name'].tolist()

#Use sets to check two lists are identical
c = set(census_list)-set(land_use_list)
d = set(land_use_list)-set(census_list)
e = []
e.append(d)
e.append(c)
e
```

```
[17]: [{"King's Lynn and West Norfolk"}, {'King's Lynn and West Norfolk'}]
```

There is one discrepancy between the two dataframes, one Local Authority has a different type of apostrophe in the two dataframes so I shall amend this so they are both the same and then check that has been done.

```
[18]: #Amend apostrophe in census data
kl = census_age_proportion_df[census_age_proportion_df == 'King's Lynn and West_
↳Norfolk'].stack().index.tolist()

census_age_proportion_df.at[kl[0]] = "King's Lynn and West Norfolk"
```

```
[19]: #Use sets to check two lists are identical
census_list = census_age_proportion_df['Area name'].tolist()
c = set(census_list)-set(land_use_list)
d = set(land_use_list)-set(census_list)
e = []
e.append(d)
e.append(c)
e
```

```
[19]: [set(), set()]
```

That has worked. The two dataframes I wish to combine have different column names for the same information so I will amend the census dataframe so it has the same column names as the land use dataframe.

```
[20]: #Chnage column names in census data so matches land use data
census_age_proportion_df = census_age_proportion_df.rename(columns={'Area name':
↳ 'Local Authority Name', 'Area code': 'ONS Code'})
census_age_proportion_df.head()
```



```
[20]: Local Authority Name  ONS Code      Region  Proportion pop aged 0-19 \
0      County Durham    E06000047  North East      21.796591
1      Darlington      E06000005  North East      22.541744
2      Hartlepool      E06000001  North East      23.401950
3      Middlesbrough   E06000002  North East      25.573315
4      Northumberland  E06000057  North East      19.962570
```

```
      Proportion pop aged 20-64  Proportion pop aged 65+ \
0      56.904808                21.356062
1      56.957328                20.686456
2      56.663055                19.718310
3      57.748436                16.817234
4      54.585153                25.452277
```

```
      Proportion pop aged 0-34  All persons
0      39.590117              522100
1      39.888683              107800
2      41.170098              92300
3      46.977067              143900
4      34.591391              320600
```

Now I will combine the two dataframes merging them with a left outer join on the ONS code and the Local Authority Name .

```
[21]: combined_df = land_use_df.merge(census_age_proportion_df, how = 'outer', on =_
↳ ['ONS Code', 'Local Authority Name'])
combined_df.head()
```

```
[21]: ONS Code Local Authority Name Developed Use Non-developed Use \
0 E07000223      Adur      941.534725      3418.435055
1 E07000026      Allerdale  4844.924035      127158.413179
2 E07000032      Amber Valley  2606.772331      23861.418671
3 E07000224      Arun      2791.129278      19613.024334
4 E07000170      Ashfield  2028.578319      8881.797678
```

```
      Total Land Area  % Developed Land      Region  Proportion pop aged 0-19 \
0      4364.505691      21.57      South East      21.860465
1      132054.795869      3.67      North West      20.083247
2      26543.663279      9.82      East Midlands      20.443740
3      22450.444783      12.43      South East      19.114078
4      10955.544524      18.52      East Midlands      22.723674
```

```
      Proportion pop aged 20-64  Proportion pop aged 65+ \
0      54.418605                23.720930
1      55.254943                24.869927
2      56.973059                22.503962
3      52.487864                28.398058
4      57.561362                19.635788
```

	Proportion pop aged 0-34	All persons
0	36.279070	64500
1	35.379813	96100
2	36.846276	126200
3	34.344660	164800
4	40.934283	126300

I will drop any columns I do not need and rename some of the columns.

```
[22]: combined_df = combined_df.drop(['Developed Use', 'Non-developed Use', 'Region',
    ↪ 'Proportion pop aged 0-19', 'Proportion pop aged 20-64', 'Proportion pop
    ↪ aged 65+'], axis=1)
combined_df = combined_df.rename(columns={'Total Land Area': 'Total Land Area
    ↪ (hectares)', 'Area code': 'ONS Code'})
combined_df.head()
```

```
[22]:      ONS Code Local Authority Name Total Land Area (hectares) \
0  E07000223          Adur          4364.505691
1  E07000026      Allerdale      132054.795869
2  E07000032    Amber Valley      26543.663279
3  E07000224          Arun       22450.444783
4  E07000170      Ashfield      10955.544524
```

	% Developed Land	Proportion pop aged 0-34	All persons
0	21.57	36.279070	64500
1	3.67	35.379813	96100
2	9.82	36.846276	126200
3	12.43	34.344660	164800
4	18.52	40.934283	126300

I would like to calculate the population density for each Local Authority and include this in the combined dataframe.

```
[23]: #Add new column with population density
combined_df['Population Density (people per hectare)'] = combined_df['All
    ↪ persons']/ combined_df['Total Land Area (hectares)']
combined_df
```

```
[23]:      ONS Code      Local Authority Name \
0  E07000223          Adur
1  E07000026      Allerdale
2  E07000032    Amber Valley
3  E07000224          Arun
4  E07000170      Ashfield
..      ...
304 E07000245      West Suffolk
305 E06000060    Buckinghamshire
```

```

306 E06000058 Bournemouth, Christchurch and Poole
307 E06000059 Dorset
308 E07000246 Somerset West and Taunton

```

	Total Land Area (hectares)	% Developed Land	Proportion pop aged 0-34 \
0	4364.505691	21.57	36.279070
1	132054.795869	3.67	35.379813
2	26543.663279	9.82	36.846276
3	22450.444783	12.43	34.344660
4	10955.544524	18.52	40.934283
..	...	...	...
304	103448.896657	7.29	41.601780
305	156488.645695	7.90	40.462846
306	17379.276560	29.77	40.144891
307	252036.384802	5.44	32.481560
308	121005.325243	4.84	36.213469

	All persons	Population Density (people per hectare)
0	64500	14.778306
1	96100	0.727728
2	126200	4.754430
3	164800	7.340612
4	126300	11.528409
..	...	...
304	179800	1.738056
305	553100	3.534442
306	400300	23.033180
307	379600	1.506132
308	157400	1.300769

[309 rows x 7 columns]

I am happy with the combined dataframe and will save it.

```
[24]: combined_df.to_csv('combined_df.csv')
```

```
[ ]:
```