

Updated on 5th February, 2024: NUTS 1 and NUTS 2 information added

# Classfying the postal codes into urban/rural typology

```
In [1]: # importing libraries
```

```
import pandas as pd
import numpy as np
```

## Urban/rural typology of NUTS3 regions

A NUTS 3 region is classified as:

**predominantly urban (PU)**, if the share of population living in rural LAU2 is below 15 %;

**intermediate (IN)**, if the share of population living in rural LAU2 is between 15 % and 50 %;

**predominantly rural (PR)**, if the share of population living in rural LAU2 is higher than 50 %.

**Local Administrative Units (LAUs)** : low level administrative divisions of a country below that of a province.

**Rural LAU2:** LAU2s with a *population density* below 150 inhabitants per km<sup>2</sup> (The OECD methodology, [https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Urban-rural\\_typology#cite\\_note-1](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Urban-rural_typology#cite_note-1))

Source: European Commission (DG REGIO and DG AGRI)

[https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Urban-rural\\_typology](https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Archive:Urban-rural_typology)

## Data source for the postal code and NUTS id:

<https://ec.europa.eu/eurostat/de/web/nuts/correspondence-tables/>

<https://ec.europa.eu/eurostat/de/web/nuts/correspondence-tables/postcodes-and-nuts>

I joined the urban classification with NUTS 3 ID table and the postal codes for NUTS 3 ID to get the final output of urban classification by postal codes.

```
In [74]: # Loading the csv file for Urban/rural typology of NUTS3 regions
df_class = pd.read_csv('Urban_rural_typology_of_NUTS_3_regions_DE.csv', encoding='cp12
df_class.head(9)
```

Out[74]:

	NUTS_ID	NUTS_NAME	category
0	DE111	Stuttgart, Stadtkreis	PU
1	DE112	Böblingen	PU
2	DE113	Esslingen	PU
3	DE114	Göppingen	IN
4	DE115	Ludwigsburg	PU
5	DE116	Rems-Murr-Kreis	PU
6	DE117	Heilbronn, Stadtkreis	IN
7	DE118	Heilbronn, Landkreis	IN
8	DE119	Hohenlohekreis	IN

0	DE111	Stuttgart, Stadtkreis	PU
1	DE112	Böblingen	PU
2	DE113	Esslingen	PU
3	DE114	Göppingen	IN
4	DE115	Ludwigsburg	PU
5	DE116	Rems-Murr-Kreis	PU
6	DE117	Heilbronn, Stadtkreis	IN
7	DE118	Heilbronn, Landkreis	IN
8	DE119	Hohenlohekreis	IN

In [75]:

```
df_class['NUTS1_ID'] = df_class['NUTS_ID'].astype(str).str[:3]
df_class.head(9)
```

Out[75]:

	NUTS_ID	NUTS_NAME	category	NUTS1_ID
0	DE111	Stuttgart, Stadtkreis	PU	DE1
1	DE112	Böblingen	PU	DE1
2	DE113	Esslingen	PU	DE1
3	DE114	Göppingen	IN	DE1
4	DE115	Ludwigsburg	PU	DE1
5	DE116	Rems-Murr-Kreis	PU	DE1
6	DE117	Heilbronn, Stadtkreis	IN	DE1
7	DE118	Heilbronn, Landkreis	IN	DE1
8	DE119	Hohenlohekreis	IN	DE1

0	DE111	Stuttgart, Stadtkreis	PU	DE1
1	DE112	Böblingen	PU	DE1
2	DE113	Esslingen	PU	DE1
3	DE114	Göppingen	IN	DE1
4	DE115	Ludwigsburg	PU	DE1
5	DE116	Rems-Murr-Kreis	PU	DE1
6	DE117	Heilbronn, Stadtkreis	IN	DE1
7	DE118	Heilbronn, Landkreis	IN	DE1
8	DE119	Hohenlohekreis	IN	DE1

In [76]:

```
df_class['NUTS1_ID'].unique()
```

Out[76]:

```
array(['DE1', 'DE2', 'DE3', 'DE4', 'DE5', 'DE6', 'DE7', 'DE8', 'DE9',
      'DEA', 'DEB', 'DEC', 'DED', 'DEE', 'DEF', 'DEG'], dtype=object)
```

In [77]:

```
df_class['NUTS1_ID'].nunique()
```

Out[77]:

```
16
```

In [78]:

```
df_class.dtypes
```

Out[78]:

```
NUTS_ID      object
NUTS_NAME    object
category     object
NUTS1_ID     object
dtype: object
```

In [79]:

```
len(df_class)
```

Out[79]: 429

```
In [80]: # Loading the file with NUTS3 ID and postal codes

df_plz = pd.read_csv('pc2020_DE_NUTS-2021_v4.0.csv', sep= ";")

#df_plz[['NUTS3', 'postal_code']] = df_plz['NUTS3;CODE'].str.split(';', expand=True)

df_plz['NUTS3'] = df_plz['NUTS3'].str.replace('\'', '')
df_plz['CODE'] = df_plz['CODE'].str.replace('\'', '')

#df_plz = df_plz[['NUTS3', 'postal_code']]

df_plz.head(9)
```

Out[80]:

	NUTS3	CODE
0	DEA1D	41363
1	DEA1E	41366
2	DEA1E	41748
3	DEA1E	41749
4	DEA1E	41751
5	DEA1E	41747
6	DEA23	50667
7	DEA23	50668
8	DEA23	50670

```
In [81]: df_plz['NUTS1_ID'] = df_plz['NUTS3'].astype(str).str[:3]
df_plz
```

Out[81]:

	NUTS3	CODE	NUTS1_ID
0	DEA1D	41363	DEA
1	DEA1E	41366	DEA
2	DEA1E	41748	DEA
3	DEA1E	41749	DEA
4	DEA1E	41751	DEA
...	...	...	...
8315	DE722	35641	DE7
8316	DE724	35080	DE7
8317	DE725	35315	DE7
8318	DE914	38543	DE9
8319	DE712	60312	DE7

8320 rows × 3 columns

In [82]: `df_plz['NUTS1_ID'].unique()`Out[82]: `array(['DEA', 'DED', 'DE8', 'DEE', 'DEF', 'DE9', 'DE4', 'DE5', 'DE7',  
 'DEB', 'DE1', 'DE2', 'DEG', 'DEC', 'DE3', 'DE6'], dtype=object)`In [83]: `df_plz['NUTS1_ID'].nunique()`

Out[83]: 16

In [84]: `len(df_plz['CODE'])`

Out[84]: 8320

In [85]: `de4_plz = df_plz[df_plz['NUTS1_ID'] == 'DE4']  
de4_plz`

Out[85]:

	NUTS3	CODE	NUTS1_ID
<b>197</b>	DE405	16352	DE4
<b>198</b>	DE40A	16565	DE4
<b>199</b>	DE40D	16918	DE4
<b>320</b>	DE40I	03202	DE4
<b>1360</b>	DE406	15755	DE4
...	...	...	...
<b>8111</b>	DE409	15366	DE4
<b>8162</b>	DE40D	16835	DE4
<b>8174</b>	DE40F	19309	DE4
<b>8209</b>	DE407	03253	DE4
<b>8229</b>	DE40F	19357	DE4

227 rows × 3 columns

In [86]:

```
de4_class = df_class[df_class['NUTS1_ID'] == 'DE4']
de4_class
```

Out[86]:

	NUTS_ID	NUTS_NAME	category	NUTS1_ID
141	DE411	Frankfurt (Oder), Kreisfreie Stadt	IN	DE4
142	DE412	Barnim	IN	DE4
143	DE413	Märkisch-Oderland	IN	DE4
144	DE414	Oberhavel	IN	DE4
145	DE415	Oder-Spree	IN	DE4
146	DE416	Ostprignitz-Ruppin	PR	DE4
147	DE417	Prignitz	PR	DE4
148	DE418	Uckermark	PR	DE4
149	DE421	Brandenburg an der Havel, Kreisfreie Stadt	IN	DE4
150	DE422	Cottbus, Kreisfreie Stadt	IN	DE4
151	DE423	Potsdam, Kreisfreie Stadt	IN	DE4
152	DE424	Dahme-Spreewald	IN	DE4
153	DE425	Elbe-Elster	PR	DE4
154	DE426	Havelland	IN	DE4
155	DE427	Oberspreewald-Lausitz	IN	DE4
156	DE428	Potsdam-Mittelmark	IN	DE4
157	DE429	Spree-Neiße	IN	DE4
158	DE42A	Teltow-Fläming	PR	DE4

## Debugging: NUTS3 ID for DE4 in PLZ differ from the classification data

In [87]: *# NUTS3 in PLZ data: this seems correct*

```
np.sort(de4_plz['NUTS3'].unique())
```

Out[87]:

```
array(['DE401', 'DE402', 'DE403', 'DE404', 'DE405', 'DE406', 'DE407',
      'DE408', 'DE409', 'DE40A', 'DE40B', 'DE40C', 'DE40D', 'DE40E',
      'DE40F', 'DE40G', 'DE40H', 'DE40I'], dtype=object)
```

In [88]: *# NUTS3 in classification data*

```
np.sort(de4_class['NUTS_ID'].unique())
```

Out[88]:

```
array(['DE411', 'DE412', 'DE413', 'DE414', 'DE415', 'DE416', 'DE417',
      'DE418', 'DE421', 'DE422', 'DE423', 'DE424', 'DE425', 'DE426',
      'DE427', 'DE428', 'DE429', 'DE42A'], dtype=object)
```

In [89]: *# change the NUTS\_ID in the classification file according to the NUTS3 level names and*

```
de4_nuts2012 = pd.DataFrame({"Code2021": ['DE401', 'DE402', 'DE403', 'DE404', 'DE405', 'DE406', 'DE407', 'DE408', 'DE409', 'DE40A', 'DE40B', 'DE40C', 'DE40D', 'DE40E', 'DE40F', 'DE40G', 'DE40H', 'DE40I'],
                             "NUTS_level3": ['Brandenburg an der Havel, Kreisfreie Stadt', 'Frankfurt (Oder), Kreisfreie Stadt', 'Potsdam, Kreisfreie Stadt', 'Cottbus, Kreisfreie Stadt', 'Dahme-Spreewald', 'Elbe-Elster', 'Havelland', 'Märkisch-Oderland', 'Oberhavel', 'Oder-Spree', 'Ostprignitz-Ruppin', 'Prignitz', 'Uckermark', 'Teltow-Fläming', 'Spree-Neiße', 'Oberspreewald-Lausitz']})
```

```
'Dahme-Spreewald', 'Elbe-Elster', 'Havelland',
'Oberhavel', 'Oberspreewald-Lausitz', 'Oder-Spree',
'Potsdam-Mittelmark', 'Prignitz', 'Spree-Neiße']
```

```
de4_nuts2012
```

Out[89]:

	Code2021	NUTS_level3
0	DE401	Brandenburg an der Havel, Kreisfreie Stadt
1	DE402	Cottbus, Kreisfreie Stadt
2	DE403	Frankfurt (Oder), Kreisfreie Stadt
3	DE404	Potsdam, Kreisfreie Stadt
4	DE405	Barnim
5	DE406	Dahme-Spreewald
6	DE407	Elbe-Elster
7	DE408	Havelland
8	DE409	Märkisch-Oderland
9	DE40A	Oberhavel
10	DE40B	Oberspreewald-Lausitz
11	DE40C	Oder-Spree
12	DE40D	Ostprignitz-Ruppin
13	DE40E	Potsdam-Mittelmark
14	DE40F	Prignitz
15	DE40G	Spree-Neiße
16	DE40H	Teltow-Fläming
17	DE40I	Uckermark

In [90]: *## testing - change the values by mapping in the classification data*

```
de4_class['NUTS_ID'] = de4_class['NUTS_ID'].map({'DE411': 'DE403', 'DE412': 'DE405', 'DE413': 'DE409', 'DE414': 'DE40A', 'DE415': 'DE40C', 'DE416': 'DE40D', 'DE417': 'DE40E', 'DE418': 'DE40F', 'DE419': 'DE40G', 'DE420': 'DE40H', 'DE421': 'DE401', 'DE422': 'DE402', 'DE423': 'DE404', 'DE424': 'DE406', 'DE425': 'DE407', 'DE426': 'DE408', 'DE427': 'DE40I', 'DE428': 'DE40J', 'DE429': 'DE40K', 'DE430': 'DE40L'})
```

```
pd.merge(de4_class, de4_nuts2012, left_on = ['NUTS_ID'], right_on = ['Code2021'], how = 'left')
```

C:\Users\leajo\AppData\Local\Temp\ipykernel\_18940\1997341574.py:3: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
de4_class['NUTS_ID'] = de4_class['NUTS_ID'].map({'DE411': 'DE403', 'DE412': 'DE405', 'DE413': 'DE409', 'DE414': 'DE40A', 'DE415': 'DE40C', 'DE416': 'DE40D', 'DE417': 'DE40E', 'DE418': 'DE40F', 'DE419': 'DE40G', 'DE420': 'DE40H', 'DE421': 'DE401', 'DE422': 'DE402', 'DE423': 'DE404', 'DE424': 'DE406', 'DE425': 'DE407', 'DE426': 'DE408', 'DE427': 'DE40I', 'DE428': 'DE40J', 'DE429': 'DE40K', 'DE430': 'DE40L'})
```

Out[90]:

	NUTS_ID	NUTS_NAME	category	NUTS1_ID	Code2021	NUTS_level3
0	DE403	Frankfurt (Oder), Kreisfreie Stadt	IN	DE4	DE403	Frankfurt (Oder), Kreisfreie Stadt
1	DE405	Barnim	IN	DE4	DE405	Barnim
2	DE409	Märkisch-Oderland	IN	DE4	DE409	Märkisch-Oderland
3	DE40A	Oberhavel	IN	DE4	DE40A	Oberhavel
4	DE40C	Oder-Spree	IN	DE4	DE40C	Oder-Spree
5	DE40D	Ostprignitz-Ruppin	PR	DE4	DE40D	Ostprignitz-Ruppin
6	DE40F	Prignitz	PR	DE4	DE40F	Prignitz
7	DE40I	Uckermark	PR	DE4	DE40I	Uckermark
8	DE401	Brandenburg an der Havel, Kreisfreie Stadt	IN	DE4	DE401	Brandenburg an der Havel, Kreisfreie Stadt
9	DE402	Cottbus, Kreisfreie Stadt	IN	DE4	DE402	Cottbus, Kreisfreie Stadt
10	DE404	Potsdam, Kreisfreie Stadt	IN	DE4	DE404	Potsdam, Kreisfreie Stadt
11	DE406	Dahme-Spreewald	IN	DE4	DE406	Dahme-Spreewald
12	DE407	Elbe-Elster	PR	DE4	DE407	Elbe-Elster
13	DE408	Havelland	IN	DE4	DE408	Havelland
14	DE40B	Oberspreewald-Lausitz	IN	DE4	DE40B	Oberspreewald-Lausitz
15	DE40E	Potsdam-Mittelmark	IN	DE4	DE40E	Potsdam-Mittelmark
16	DE40G	Spree-Neiße	IN	DE4	DE40G	Spree-Neiße
17	DE40H	Teltow-Fläming	PR	DE4	DE40H	Teltow-Fläming

In [91]:

```
## change the values by mapping in the classification data
```

```
df_class['NUTS_ID'] = df_class['NUTS_ID'].map({'DE411': 'DE403', 'DE412': 'DE405', 'DE413': 'DE406', 'DE414': 'DE407', 'DE415': 'DE40C', 'DE416': 'DE40D', 'DE417': 'DE40E', 'DE418': 'DE40F', 'DE419': 'DE40G', 'DE420': 'DE40H', 'DE421': 'DE401', 'DE422': 'DE402', 'DE423': 'DE403', 'DE424': 'DE404', 'DE425': 'DE407', 'DE426': 'DE408', 'DE427': 'DE409', 'DE428': 'DE40A', 'DE429': 'DE40G', 'DE42A': 'DE40H'})
```

In [92]:

```
df_class.head()
```

Out[92]:

	NUTS_ID	NUTS_NAME	category	NUTS1_ID
0	NaN	Stuttgart, Stadtkreis	PU	DE1
1	NaN	Böblingen	PU	DE1
2	NaN	Esslingen	PU	DE1
3	NaN	Göppingen	IN	DE1
4	NaN	Ludwigsburg	PU	DE1



## Merge with the clean data

```
In [93]: ## Load df_class again
df_class2 = pd.read_csv('Urban_rural_typology_of_NUTS_3_regions_DE.csv', encoding='cp1
df_class2.head(9)
```

```
Out[93]:
```

	NUTS_ID	NUTS_NAME	category
0	DE111	Stuttgart, Stadtkreis	PU
1	DE112	Böblingen	PU
2	DE113	Esslingen	PU
3	DE114	Göppingen	IN
4	DE115	Ludwigsburg	PU
5	DE116	Rems-Murr-Kreis	PU
6	DE117	Heilbronn, Stadtkreis	IN
7	DE118	Heilbronn, Landkreis	IN
8	DE119	Hohenlohekreis	IN

```
In [94]: DE4_map = {'DE411': 'DE403', 'DE412': 'DE405', 'DE413': 'DE409', 'DE414': 'DE40A',
                    'DE415': 'DE40C', 'DE416': 'DE40D', 'DE417': 'DE40E', 'DE418': 'DE40F',
                    'DE421': 'DE401', 'DE422': 'DE402', 'DE423': 'DE403', 'DE424': 'DE404',
                    'DE425': 'DE407', 'DE426': 'DE408', 'DE427': 'DE409', 'DE428': 'DE40A',
                    'DE429': 'DE40G', 'DE42A': 'DE40H'}

for key, value in DE4_map.items():
    df_class2['NUTS_ID'].mask(df_class2['NUTS_ID'] == key, value, inplace=True)
```

```
In [95]: # Creating NUTS1 ID column

df_class2['NUTS1_ID'] = df_class2['NUTS_ID'].astype(str).str[:3]
df_class2.head()
```

```
Out[95]:
```

	NUTS_ID	NUTS_NAME	category	NUTS1_ID
0	DE111	Stuttgart, Stadtkreis	PU	nan
1	DE112	Böblingen	PU	nan
2	DE113	Esslingen	PU	nan
3	DE114	Göppingen	IN	nan
4	DE115	Ludwigsburg	PU	nan

```
In [96]: test3 = df_class2[df_class2['NUTS1_ID'] == 'DE4']
test3['NUTS_ID'].unique()
```

```
Out[96]: array(['DE403', 'DE405', 'DE409', 'DE40A', 'DE40C', 'DE40D', 'DE40F',
                'DE40I', 'DE401', 'DE402', 'DE404', 'DE406', 'DE407', 'DE408',
                'DE40B', 'DE40E', 'DE40G', 'DE40H'], dtype=object)
```

```
In [97]: ## Merging the PLZ and classification data after fixing the discrepancies in DE3 NUTS3

df_with_null = pd.merge(df_plz, df_class2, left_on = ['NUTS3'], right_on = ['NUTS_ID'])
df_with_null = df_with_null[['NUTS3', 'CODE', 'NUTS_NAME', 'category']]
df_with_null.head(10)
```

```
Out[97]:
```

	NUTS3	CODE	NUTS_NAME	category
0	DEA1D	41363	Rhein-Kreis Neuss	PU
1	DEA1E	41366	Viersen	PU
2	DEA1E	41748	Viersen	PU
3	DEA1E	41749	Viersen	PU
4	DEA1E	41751	Viersen	PU
5	DEA1E	41747	Viersen	PU
6	DEA23	50667	Köln, Kreisfreie Stadt	PU
7	DEA23	50668	Köln, Kreisfreie Stadt	PU
8	DEA23	50670	Köln, Kreisfreie Stadt	PU
9	DEA29	41836	Heinsberg	PU

```
In [98]: df_with_null['NUTS1_ID'] = df_with_null['NUTS3'].astype(str).str[:3]
df_with_null.head(10)
```

```
Out[98]:
```

	NUTS3	CODE	NUTS_NAME	category	NUTS1_ID
0	DEA1D	41363	Rhein-Kreis Neuss	PU	DEA
1	DEA1E	41366	Viersen	PU	DEA
2	DEA1E	41748	Viersen	PU	DEA
3	DEA1E	41749	Viersen	PU	DEA
4	DEA1E	41751	Viersen	PU	DEA
5	DEA1E	41747	Viersen	PU	DEA
6	DEA23	50667	Köln, Kreisfreie Stadt	PU	DEA
7	DEA23	50668	Köln, Kreisfreie Stadt	PU	DEA
8	DEA23	50670	Köln, Kreisfreie Stadt	PU	DEA
9	DEA29	41836	Heinsberg	PU	DEA

```
In [99]: df_with_null['NUTS1_ID'].unique()
```

```
Out[99]: array(['DEA', 'DED', 'DE8', 'DEE', 'DEF', 'DE9', 'DE4', 'DE5', 'DE7',
        'DEB', 'DE1', 'DE2', 'DEG', 'DEC', 'DE3', 'DE6'], dtype=object)
```

```
In [100]: df_with_null['NUTS1_ID'].nunique()
```

```
Out[100]: 16
```

```
In [101... df_with_null['NUTS2_ID'] = df_with_null['NUTS3'].astype(str).str[:4]
df_with_null.head(10)
```

```
Out[101]:
```

	NUTS3	CODE	NUTS_NAME	category	NUTS1_ID	NUTS2_ID
0	DEA1D	41363	Rhein-Kreis Neuss	PU	DEA	DEA1
1	DEA1E	41366	Viersen	PU	DEA	DEA1
2	DEA1E	41748	Viersen	PU	DEA	DEA1
3	DEA1E	41749	Viersen	PU	DEA	DEA1
4	DEA1E	41751	Viersen	PU	DEA	DEA1
5	DEA1E	41747	Viersen	PU	DEA	DEA1
6	DEA23	50667	Köln, Kreisfreie Stadt	PU	DEA	DEA2
7	DEA23	50668	Köln, Kreisfreie Stadt	PU	DEA	DEA2
8	DEA23	50670	Köln, Kreisfreie Stadt	PU	DEA	DEA2
9	DEA29	41836	Heinsberg	PU	DEA	DEA2

```
In [102... df_with_null['NUTS2_ID'].unique()
```

```
Out[102]: array(['DEA1', 'DEA2', 'DEA3', 'DEA4', 'DEA5', 'DED4', 'DE80', 'DEE0',
      'DEF0', 'DE93', 'DE40', 'DE91', 'DE92', 'DE94', 'DE50', 'DE71',
      'DEB1', 'DEB2', 'DEB3', 'DE11', 'DE12', 'DE13', 'DE14', 'DE21',
      'DE22', 'DE27', 'DE23', 'DE24', 'DE25', 'DED2', 'DED5', 'DEG0',
      'DE26', 'DE72', 'DE73', 'DEC0', 'DE30', 'DE60'], dtype=object)
```

```
In [103... df_with_null['NUTS2_ID'].nunique()
```

```
Out[103]: 38
```

```
In [104... df_with_null[df_with_null['NUTS1_ID'] == 'DE4']
```

Out[104]:

	NUTS3	CODE	NUTS_NAME	category	NUTS1_ID	NUTS2_ID
<b>197</b>	DE405	16352	Barnim	IN	DE4	DE40
<b>198</b>	DE40A	16565	Oberhavel	IN	DE4	DE40
<b>199</b>	DE40D	16918	Ostprignitz-Ruppin	PR	DE4	DE40
<b>320</b>	DE40I	03202	Uckermark	PR	DE4	DE40
<b>1360</b>	DE406	15755	Dahme-Spreewald	IN	DE4	DE40
...	...	...	...	...	...	...
<b>8111</b>	DE409	15366	Märkisch-Oderland	IN	DE4	DE40
<b>8162</b>	DE40D	16835	Ostprignitz-Ruppin	PR	DE4	DE40
<b>8174</b>	DE40F	19309	Prignitz	PR	DE4	DE40
<b>8209</b>	DE407	03253	Elbe-Elster	PR	DE4	DE40
<b>8229</b>	DE40F	19357	Prignitz	PR	DE4	DE40

227 rows × 6 columns

## Adding other NUTS level

used file: NUTS2021

source: <https://ec.europa.eu/eurostat/de/web/nuts/background>

In [105...

## Adding NUTS1

```
NUTS1 = pd.read_csv('NUTS1.csv', encoding='cp1252')
NUTS1
```

Out[105]:

	Code 2021	NUTS level 1
<b>0</b>	DE1	Baden-Württemberg
<b>1</b>	DE2	Bayern
<b>2</b>	DE3	Berlin
<b>3</b>	DE4	Brandenburg
<b>4</b>	DE5	Bremen
<b>5</b>	DE6	Hamburg
<b>6</b>	DE7	Hessen
<b>7</b>	DE8	Mecklenburg-Vorpommern
<b>8</b>	DE9	Niedersachsen
<b>9</b>	DEA	Nordrhein-Westfalen
<b>10</b>	DEB	Rheinland-Pfalz
<b>11</b>	DEC	Saarland
<b>12</b>	DED	Sachsen
<b>13</b>	DEE	Sachsen-Anhalt
<b>14</b>	DEF	Schleswig-Holstein
<b>15</b>	DEG	Thüringen
<b>16</b>	DEZ	Extra-Regio NUTS 1

In [106...

```
## Adding NUTS2
NUTS2 = pd.read_csv('NUTS2.csv', encoding='cp1252')
NUTS2
```

Out[106]:

	Code 2021	NUTS level 2
0	DE11	Stuttgart
1	DE12	Karlsruhe
2	DE13	Freiburg
3	DE14	Tübingen
4	DE21	Oberbayern
5	DE22	Niederbayern
6	DE23	Oberpfalz
7	DE24	Oberfranken
8	DE25	Mittelfranken
9	DE26	Unterfranken
10	DE27	Schwaben
11	DE30	Berlin
12	DE40	Brandenburg
13	DE50	Bremen
14	DE60	Hamburg
15	DE71	Darmstadt
16	DE72	Gießen
17	DE73	Kassel
18	DE80	Mecklenburg-Vorpommern
19	DE91	Braunschweig
20	DE92	Hannover
21	DE93	Lüneburg
22	DE94	Weser-Ems
23	DEA1	Düsseldorf
24	DEA2	Köln
25	DEA3	Münster
26	DEA4	Detmold
27	DEA5	Arnsberg
28	DEB1	Koblenz
29	DEB2	Trier
30	DEB3	Rheinhessen-Pfalz
31	DEC0	Saarland
32	DED2	Dresden
33	DED4	Chemnitz

Code 2021		NUTS level 2
34	DED5	Leipzig
35	DEE0	Sachsen-Anhalt
36	DEF0	Schleswig-Holstein
37	DEG0	Thüringen
38	DEZZ	Extra-Regio NUTS 2

In [107...

df\_with\_null.head()

Out[107]:

	NUTS3	CODE	NUTS_NAME	category	NUTS1_ID	NUTS2_ID
0	DEA1D	41363	Rhein-Kreis Neuss	PU	DEA	DEA1
1	DEA1E	41366	Viersen	PU	DEA	DEA1
2	DEA1E	41748	Viersen	PU	DEA	DEA1
3	DEA1E	41749	Viersen	PU	DEA	DEA1
4	DEA1E	41751	Viersen	PU	DEA	DEA1

In [108...

### try left join instead

In [109...

df\_with\_null = pd.merge(df\_with\_null, NUTS1, left\_on = ['NUTS1\_ID'], right\_on = ['Code  
df\_with\_null.head(10)

Out[109]:

	NUTS3	CODE	NUTS_NAME	category	NUTS1_ID	NUTS2_ID	Code 2021	NUTS level 1
0	DEA1D	41363	Rhein-Kreis Neuss	PU	DEA	DEA1	DEA	Nordrhein-Westfalen
1	DEA1E	41366	Viersen	PU	DEA	DEA1	DEA	Nordrhein-Westfalen
2	DEA1E	41748	Viersen	PU	DEA	DEA1	DEA	Nordrhein-Westfalen
3	DEA1E	41749	Viersen	PU	DEA	DEA1	DEA	Nordrhein-Westfalen
4	DEA1E	41751	Viersen	PU	DEA	DEA1	DEA	Nordrhein-Westfalen
5	DEA1E	41747	Viersen	PU	DEA	DEA1	DEA	Nordrhein-Westfalen
6	DEA23	50667	Köln, Kreisfreie Stadt	PU	DEA	DEA2	DEA	Nordrhein-Westfalen
7	DEA23	50668	Köln, Kreisfreie Stadt	PU	DEA	DEA2	DEA	Nordrhein-Westfalen
8	DEA23	50670	Köln, Kreisfreie Stadt	PU	DEA	DEA2	DEA	Nordrhein-Westfalen
9	DEA29	41836	Heinsberg	PU	DEA	DEA2	DEA	Nordrhein-Westfalen

In [110...

```
df_with_null = pd.merge(df_with_null, NUTS2, left_on = ['NUTS2_ID'], right_on = ['Code  
df_with_null.head(10)
```



Out[110]:

	NUTS3	CODE	NUTS_NAME	category	NUTS1_ID	NUTS2_ID	Code 2021_x	NUTS level 1	Code 2021_y	NU leve
0	DEA1D	41363	Rhein-Kreis Neuss	PU	DEA	DEA1	DEA	Nordrhein- Westfalen	DEA1	Düsseld
1	DEA1E	41366	Viersen	PU	DEA	DEA1	DEA	Nordrhein- Westfalen	DEA1	Düsseld
2	DEA1E	41748	Viersen	PU	DEA	DEA1	DEA	Nordrhein- Westfalen	DEA1	Düsseld
3	DEA1E	41749	Viersen	PU	DEA	DEA1	DEA	Nordrhein- Westfalen	DEA1	Düsseld
4	DEA1E	41751	Viersen	PU	DEA	DEA1	DEA	Nordrhein- Westfalen	DEA1	Düsseld
5	DEA1E	41747	Viersen	PU	DEA	DEA1	DEA	Nordrhein- Westfalen	DEA1	Düsseld
6	DEA23	50667	Köln, Kreisfreie Stadt	PU	DEA	DEA2	DEA	Nordrhein- Westfalen	DEA2	Kö
7	DEA23	50668	Köln, Kreisfreie Stadt	PU	DEA	DEA2	DEA	Nordrhein- Westfalen	DEA2	Kö
8	DEA23	50670	Köln, Kreisfreie Stadt	PU	DEA	DEA2	DEA	Nordrhein- Westfalen	DEA2	Kö
9	DEA29	41836	Heinsberg	PU	DEA	DEA2	DEA	Nordrhein- Westfalen	DEA2	Kö



In [111...

```
df_with_null = df_with_null[['NUTS3', 'CODE', 'NUTS_NAME', 'category', 'NUTS1_ID', 'NUTS2_ID', 'Code 2021_x', 'NUTS level 1', 'Code 2021_y', 'NUTS level 2']]
df_with_null.head(10)
```

Out[111]:

	NUTS3	CODE	NUTS_NAME	category	NUTS1_ID	NUTS2_ID	NUTS level 1	NUTS level 2
0	DEA1D	41363	Rhein-Kreis Neuss	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
1	DEA1E	41366	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
2	DEA1E	41748	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
3	DEA1E	41749	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
4	DEA1E	41751	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
5	DEA1E	41747	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
6	DEA23	50667	Köln, Kreisfreie Stadt	PU	DEA	DEA2	Nordrhein-Westfalen	Köln
7	DEA23	50668	Köln, Kreisfreie Stadt	PU	DEA	DEA2	Nordrhein-Westfalen	Köln
8	DEA23	50670	Köln, Kreisfreie Stadt	PU	DEA	DEA2	Nordrhein-Westfalen	Köln
9	DEA29	41836	Heinsberg	PU	DEA	DEA2	Nordrhein-Westfalen	Köln

In [112]:

```
df_with_null = df_with_null.rename(columns={'NUTS3': 'NUTS3_ID', 'NUTS_NAME' : 'NUTS3_NAME', 'category': 'CLASSIFICATION', 'NUTS1_ID': 'NUTS1_ID', 'NUTS2_ID': 'NUTS2_ID', 'NUTS level 1': 'NUTS1_NAME', 'NUTS level 2': 'NUTS2_NAME'})
df_with_null.head()
```

Out[112]:

	NUTS3_ID	POSTAL_CODE	NUTS3_NAME	CLASSIFICATION	NUTS1_ID	NUTS2_ID	NUTS1_NAME	NUTS2_NAME
0	DEA1D	41363	Rhein-Kreis Neuss	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
1	DEA1E	41366	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
2	DEA1E	41748	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
3	DEA1E	41749	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf
4	DEA1E	41751	Viersen	PU	DEA	DEA1	Nordrhein-Westfalen	Düsseldorf



# missing classification

In [116]:

```
df_class_null = df_with_null[df_with_null['CLASSIFICATION'].isnull()]
df_class_null.head()
```

Out[116]:

	NUTS3_ID	POSTAL_CODE	NUTS3_NAME	CLASSIFICATION	NUTS1_ID	NUTS2_ID	NUTS1_NAME
64	DEA2D	52249	NaN	NaN	DEA	DEA2	Nordrhein-Westfalen
65	DEA2D	52134	NaN	NaN	DEA	DEA2	Nordrhein-Westfalen
69	DEA2D	52499	NaN	NaN	DEA	DEA2	Nordrhein-Westfalen
70	DEA2D	52159	NaN	NaN	DEA	DEA2	Nordrhein-Westfalen
88	DEA2D	52152	NaN	NaN	DEA	DEA2	Nordrhein-Westfalen

In [117... `len(df_class_null)`

Out[117]: 654

In [118... `df_class_null['NUTS1_ID'].unique()`

Out[118]: array(['DEA', 'DED', 'DE8', 'DE9', 'DEB'], dtype=object)

In [119... `df_not_null = df_with_null[df_with_null['CLASSIFICATION'].notnull()]`  
`len(df_not_null)`

Out[119]: 7666

In [120... `df_not_null_final = df_not_null[['NUTS3_ID', 'NUTS1_NAME', 'NUTS2_NAME', 'NUTS3_NAME', 'POSTAL_CODE', 'CLASSIFICATION']]`  
`df_not_null_final.head()`

Out[120]:

	NUTS3_ID	NUTS1_NAME	NUTS2_NAME	NUTS3_NAME	POSTAL_CODE	CLASSIFICATION
0	DEA1D	Nordrhein-Westfalen	Düsseldorf	Rhein-Kreis Neuss	41363	PU
1	DEA1E	Nordrhein-Westfalen	Düsseldorf	Viersen	41366	PU
2	DEA1E	Nordrhein-Westfalen	Düsseldorf	Viersen	41748	PU
3	DEA1E	Nordrhein-Westfalen	Düsseldorf	Viersen	41749	PU
4	DEA1E	Nordrhein-Westfalen	Düsseldorf	Viersen	41751	PU

In [121... `df_not_null_final.to_csv('classification_urban_by_postal_code.csv', encoding='cp1252')`

In [125... `df_class_null_final = df_class_null[['NUTS3_ID', 'NUTS1_NAME', 'NUTS2_NAME', 'NUTS3_NAME', 'POSTAL_CODE', 'CLASSIFICATION']]`  
`df_class_null_final.head()`

Out[125]:

	NUTS3_ID	NUTS1_NAME	NUTS2_NAME	NUTS3_NAME	POSTAL_CODE	CLASSIFICATION
64	DEA2D	Nordrhein-Westfalen	Köln	NaN	52249	NaN
65	DEA2D	Nordrhein-Westfalen	Köln	NaN	52134	NaN
69	DEA2D	Nordrhein-Westfalen	Köln	NaN	52499	NaN
70	DEA2D	Nordrhein-Westfalen	Köln	NaN	52159	NaN
88	DEA2D	Nordrhein-Westfalen	Köln	NaN	52152	NaN

```
In [126... df_class_null_final.to_csv('classification_missing_by_postal_code.csv', encoding='cp12
```

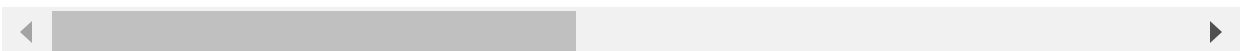
## There are unclassified postal codes that needs further investigation:

It means if the postal code of the respondent is one of these 881 postal codes, we do not know their classification.

Today there are 28,278 different postal codes assigned in Germany, of which 8,181 are for towns, 16,137 for PO boxes, 3,095 for major customers and 865 are so-called "action postal codes" (e.g. for competitions). Three buildings in Frankfurt aM also have their own postal code, namely Messeturm (60308), Opernturm (60306) and Taunusturm (60310). The fourth German building with its own delivery zip code is the Schneefernerhaus on the Zugspitze (82475). With the bridegroom oak in the Dodauer Forest in Eutin, even a single tree can be reached under a zip code (23701).

234 places in Germany have more than one postal code. Berlin is the city with the most so-called "delivery postcodes" (190), followed by Hamburg with 100 and Munich with 75. Of the federal states, North Rhine-Westphalia has the most active postal codes and Bremen has the fewest.

<https://group.dhl.com/de/presse/pressemitteilungen/2018/25-jahre-fuenfstellige-postleitzahlen-in-deutschland.html#:~:text=Heute%20sind%20in%20Deutschland%2028.278,%22Aktions%20DPLZ%22>



[https://tbed.org/eudemo/index.php?tablename=nuts3\\_vw&function=details&where\\_field=nuts\\_code&where\\_value=DEA2D](https://tbed.org/eudemo/index.php?tablename=nuts3_vw&function=details&where_field=nuts_code&where_value=DEA2D)