

DSC630_WK00-term_Kim-Schreck

June 1, 2024

1 DSC630_WK00-term_Kim-Schreck

```
[1]: # imports

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from sklearn.decomposition import PCA
import thinkstats2
import thinkplot
import datetime
import statistics
import first
from scipy.stats import poisson
from sklearn.preprocessing import PolynomialFeatures
import math
from sklearn.tree import DecisionTreeRegressor
from sklearn import tree, metrics
import pandas_datareader.data as web
import pandas_datareader as web
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import RFE
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
from sklearn.metrics import classification_report
from sklearn import model_selection
from sklearn.linear_model import LogisticRegression
```

```
from sklearn import preprocessing
from sklearn import utils
```

2 DSC630_WK01-ms01.02

```
[2]: # 01.02.00

## DSC630_term_milestone01_Schreck-Kim.pdf
```

```
[3]: # 01.02.01

## Milestone One - Project Plan
```

```
[4]: # 01.02.02

## Milestone One - Peer Review
```

3 DSC630_WK02-ms02.02

```
[5]: # 02.02.00

## DSC630_term_milestone02_Schreck-Kim.pdf
```

```
[6]: # 02.02.01
# read csv
# dataset ( )_20240320113224.csv
# dt01

dt01 = pd.read_csv(' ( )_20240320113224.csv')
```

```
[7]: # 02.02.02
# read csv
# dataset seoul - SeoulRealEstate.csv
# dt02

dt02 = pd.read_csv(' _ .csv')
```

```
[8]: # 02.02.03
# return first ten rows
# dt01

dt01.head(10)
```

```
[8]:      (1)      (2) 2023. 10 2023. 10.1 2023. 11 2023. 11.1 2023. 12 2023. 12.1
0      (1)      (2)
1                96.7      94.2      96.8      94.4      96.7      94.2
```

2	98.5	93.0	98.7	93.1	98.8	93.1
3	94.4	90.7	94.5	90.8	94.5	90.7
4	100.9	99.2	101.2	99.6	101.2	99.6
5	98.6	97.1	98.9	97.5	99.0	97.6
6	97.3	93.5	97.4	93.7	97.4	93.7
7	95.7	91.6	95.9	91.9	95.9	91.8
8	96.1	92.5	96.1	92.6	96.1	92.5
9	94.2	90.3	94.3	90.4	94.2	90.4

```
[9]: # 02.02.04
      # return last ten rows
      # dt01

      dt01.tail(10)
```

	(1)	(2)	2023. 10	2023. 10.1	2023. 11	2023. 11.1	2023. 12	2023. 12.1
17			94.1	91.4	94.3	91.6	94.2	91.5
18			95.5	93.5	95.5	93.4	95.3	93.2
19			94.2	90.1	94.3	90.2	94.2	90.0
20			95.8	94.1	96.0	94.4	96.1	94.4
21			98.3	96.4	98.4	96.4	98.2	96.1
22			95.8	92.0	95.9	92.0	95.8	91.8
23			103.3	103.8	103.3	103.8	103.1	103.6
24			101.6	101.8	101.6	101.7	101.5	101.6
25			100.5	99.7	100.7	99.9	100.7	99.8
26			97.9	95.9	98.1	96.2	98.1	96.2

```
[10]: # 02.02.05
        # return dimensions
        # dt01

        print(dt01.shape)
```

(27, 8)

```
[11]: # 02.02.06
        # return first ten rows
        # dt02

        dt02.head(10)
```

	id	lat	lng	...	min_sales	max_sales	avg_sales
0	2766	37.681604	127.056592	...	60100.0	62000.0	61000.0
1	5860	37.679290	127.057021	...	48600.0	52200.0	51000.0
2	15564	37.676882	127.058075	...	36000.0	46000.0	40500.0
3	3700	37.675277	127.060001	...	34000.0	34800.0	34500.0
4	6204	37.676381	127.058361	...	27900.0	50300.0	40000.0
5	3697	37.677354	127.057269	...	29200.0	39000.0	34500.0

6	19203	37.676590	127.057114	...	36500.0	64600.0	63000.0
7	3699	37.673836	127.060131	...	24300.0	40400.0	31000.0
8	948	37.681097	127.053757	...	17100.0	26600.0	23500.0
9	15206	37.679962	127.053507	...	15800.0	26000.0	22500.0

[10 rows x 11 columns]

```
[12]: # 02.02.07
# return last ten rows
# dt02

dt02.tail(10)
```

```
[12]:      id      lat      lng ... min_sales max_sales avg_sales
4011  16702  37.530280  126.834985 ...  24400.0   24400.0   24500.0
4012   4464  37.530704  126.833889 ...  13000.0   30600.0   21000.0
4013   7676  37.533702  126.867811 ...  59900.0   82600.0   70000.0
4014   6389  37.534864  126.869493 ...  65600.0   74900.0   68000.0
4015   5757  37.534237  126.869249 ...  66400.0   66500.0   66500.0
4016   6433  37.532183  126.866836 ...  53300.0   89000.0   63000.0
4017  19721  37.531725  126.829592 ...  20100.0   20100.0   20000.0
4018  21277  37.531586  126.828611 ...  18200.0   25800.0   24000.0
4019   6434  37.530969  126.867097 ...  53500.0   91500.0   70000.0
4020  18026  37.530168  126.829991 ...  11500.0   22200.0   19500.0
```

[10 rows x 11 columns]

```
[13]: # 02.02.08
# return dimensions
# dt02

print(dt02.shape)
```

(4021, 11)

```
[14]: # 02.02.09
# confirm column names
# dt01

dt01.columns
```

```
[14]: Index([' (1)', ' (2)', '2023. 10', '2023. 10.1', '2023. 11',
        '2023. 11.1', '2023. 12', '2023. 12.1'],
        dtype='object')
```

```
[15]: # 02.02.10
# confirm column names
# dt02
```

```
dt02.columns
```

```
[15]: Index(['id', 'lat', 'lng', 'households', 'buildDate', 'score', 'm2', 'p',  
          'min_sales', 'max_sales', 'avg_sales'],  
          dtype='object')
```

```
[16]: # 02.02.11  
      # column rename to remove spaces  
      # dt01  
  
      dt01_rn = dt01.rename(columns = {  
          ' (1)': ' _01',  
          ' (2)': ' _02',  
          '2023. 10': '2023_10',  
          '2023. 10.1': '2023_10_01',  
          '2023. 11': '2023_11',  
          '2023. 11.1': '2023_11_01',  
          '2023. 12': '2023_12'  
      })
```

```
[17]: # 02.02.12  
      # confirm column names  
      # dt01  
  
      dt01_rn.columns
```

```
[17]: Index([' _01', ' _02', '2023_10', '2023_10_01', '2023_11', '2023_11_01',  
          '2023_12', '2023. 12.1'],  
          dtype='object')
```

```
[18]: # 02.02.13  
      # column rename to remove spaces  
      # dt02  
  
      dt02_rn = dt02.rename(columns = {  
          'id': 'id',  
          'lat': 'lat',  
          'lng': 'lon',  
          'households': 'households',  
          'buildDate': 'date_build',  
          'score': 'score',  
          'm2': 'm2',  
          'p': 'pyung',  
          'min_sales': 'sales_min',  
          'max_sales': 'sales_max',  
          'avg_sales': 'sales_ave',
```

```
})
```

```
[19]: # 02.02.14
      # confirm column names
      # dt02

      dt02_rn.columns
```

```
[19]: Index(['id', 'lat', 'lon', 'households', 'date_build', 'score', 'm2', 'pyung',
          'sales_min', 'sales_max', 'sales_ave'],
          dtype='object')
```

```
[20]: # 02.02.15
      # return dimensions
      # dt01

      print(dt01_rn.shape)
```

```
(27, 8)
```

```
[21]: # 02.02.16
      # return dimensions
      # dt02

      print(dt02_rn.shape)
```

```
(4021, 11)
```

4 DSC630_WK06-ms03.02

```
[22]: # 03.02.00

      ## DSC630_term_milestone03_Schreck-Kim.pdf
```

```
[23]: # 03.02.01
      # confirm column names
      # dt01

      dt01_rn.columns
```

```
[23]: Index(['_01', '_02', '2023_10', '2023_10_01', '2023_11', '2023_11_01',
          '2023_12', '2023. 12.1'],
          dtype='object')
```

```
[24]: # 03.02.02
      # confirm column names
      # dt02
```

```
dt02_rn.columns
```

```
[24]: Index(['id', 'lat', 'lon', 'households', 'date_build', 'score', 'm2', 'pyung',  
        'sales_min', 'sales_max', 'sales_ave'],  
        dtype='object')
```

```
[25]: # 03.02.03  
      # return first ten rows  
      # dt01  
  
dt01.head(10)
```

```
[25]:      (1)      (2) 2023. 10 2023. 10.1 2023. 11 2023. 11.1 2023. 12 2023. 12.1  
0      (1)      (2)  
1              96.7      94.2      96.8      94.4      96.7      94.2  
2              98.5      93.0      98.7      93.1      98.8      93.1  
3              94.4      90.7      94.5      90.8      94.5      90.7  
4             100.9      99.2     101.2      99.6     101.2      99.6  
5              98.6      97.1      98.9      97.5      99.0      97.6  
6              97.3      93.5      97.4      93.7      97.4      93.7  
7              95.7      91.6      95.9      91.9      95.9      91.8  
8              96.1      92.5      96.1      92.6      96.1      92.5  
9              94.2      90.3      94.3      90.4      94.2      90.4
```

```
[26]: # 03.02.04  
      # return last ten rows  
      # dt01  
  
dt01.tail(10)
```

```
[26]:      (1)      (2) 2023. 10 2023. 10.1 2023. 11 2023. 11.1 2023. 12 2023. 12.1  
17              94.1      91.4      94.3      91.6      94.2      91.5  
18              95.5      93.5      95.5      93.4      95.3      93.2  
19              94.2      90.1      94.3      90.2      94.2      90.0  
20              95.8      94.1      96.0      94.4      96.1      94.4  
21              98.3      96.4      98.4      96.4      98.2      96.1  
22              95.8      92.0      95.9      92.0      95.8      91.8  
23             103.3     103.8     103.3     103.8     103.1     103.6  
24             101.6     101.8     101.6     101.7     101.5     101.6  
25             100.5      99.7     100.7      99.9     100.7      99.8  
26              97.9      95.9      98.1      96.2      98.1      96.2
```

```
[27]: # 03.02.05  
      # return first ten rows  
      # dt02  
  
dt02.head(10)
```

```
[27]:
```

	id	lat	lng	...	min_sales	max_sales	avg_sales
0	2766	37.681604	127.056592	...	60100.0	62000.0	61000.0
1	5860	37.679290	127.057021	...	48600.0	52200.0	51000.0
2	15564	37.676882	127.058075	...	36000.0	46000.0	40500.0
3	3700	37.675277	127.060001	...	34000.0	34800.0	34500.0
4	6204	37.676381	127.058361	...	27900.0	50300.0	40000.0
5	3697	37.677354	127.057269	...	29200.0	39000.0	34500.0
6	19203	37.676590	127.057114	...	36500.0	64600.0	63000.0
7	3699	37.673836	127.060131	...	24300.0	40400.0	31000.0
8	948	37.681097	127.053757	...	17100.0	26600.0	23500.0
9	15206	37.679962	127.053507	...	15800.0	26000.0	22500.0

[10 rows x 11 columns]

```
[28]: # 03.02.06
# return last ten rows
# dt02

dt02.tail(10)
```

```
[28]:
```

	id	lat	lng	...	min_sales	max_sales	avg_sales
4011	16702	37.530280	126.834985	...	24400.0	24400.0	24500.0
4012	4464	37.530704	126.833889	...	13000.0	30600.0	21000.0
4013	7676	37.533702	126.867811	...	59900.0	82600.0	70000.0
4014	6389	37.534864	126.869493	...	65600.0	74900.0	68000.0
4015	5757	37.534237	126.869249	...	66400.0	66500.0	66500.0
4016	6433	37.532183	126.866836	...	53300.0	89000.0	63000.0
4017	19721	37.531725	126.829592	...	20100.0	20100.0	20000.0
4018	21277	37.531586	126.828611	...	18200.0	25800.0	24000.0
4019	6434	37.530969	126.867097	...	53500.0	91500.0	70000.0
4020	18026	37.530168	126.829991	...	11500.0	22200.0	19500.0

[10 rows x 11 columns]

5 DSC630_WK09-ms04.02

```
[29]: # 04.02.00

## DSC630_term_milestone04_Schreck-Kim.pdf
```

```
[30]: # 04.02.01-01
# rename dt01, dt02
# read csv (27 added) dt03-dt44
# this step makes it easier to work with the data during transformations
# relabeled datasets in english

dt01_real_estate_pr_01 = dt01
```



```

dt02_real_estate_se_01 = dt02
dt03_stats_vital_kr_01 = pd.read_csv('          _20240427103929.csv')
dt04_housing_census_01 = pd.
    ↪read_csv('    __of_          _20240427102536.csv')
dt05_housing_constr_01 = pd.read_csv('          _20240427104335.csv')
dt06_household_empt_02 = pd.read_csv('          _20240427102612.csv')
dt07_housing_loss___01 = pd.read_csv('          _20240427102442.csv')
dt08_housing_use_sv_01 = pd.read_csv('          _20240427104424.csv')
dt09_jeonse_sales___01 = pd.read_csv('          _20240427102712.
    ↪csv')
dt10_jeonse_market__02 = pd.read_csv('    _of_          _20240427102647.
    ↪csv')
dt11_birthingrate_age__01 = pd.read_csv('          _20240427103626.csv')
dt12_birthingrate_age__02 = pd.read_csv('          _20240427103530.
    ↪csv')
dt13_pop_houshold___01 = pd.read_csv('          _20240427102858.csv')
dt14_pop_houshold___02 = pd.read_csv('          _20240427103017.csv')
dt15_pop_houshold___03 = pd.read_csv('          _20240427103320.csv')
dt16_pop_houshold___04 = pd.read_csv('          _20240427103114.csv')
dt17_pop_houshold___05 = pd.read_csv('          _20240427103142.csv')
dt18_pop_houshold___06 = pd.read_csv('          _20240427103208.csv')
dt19_pop_houshold___07 = pd.read_csv('          _20240427103233.csv')
dt20_pop_houshold___08 = pd.read_csv('          _20240427103256.csv')
dt21_pop_houshold___09 = pd.
    ↪read_csv('          _20240427103044.csv')
dt22_pop_estimate___01 = pd.read_csv('          _20240427103433.csv')
dt23_pop_future_pro_01 = pd.read_csv('          _20240427103555.
    ↪csv')
dt24_pop_future_pro_02 = pd.
    ↪read_csv('          _20240427103708.csv')
dt25_pop_future_pro_02 = pd.read_csv('          _20240427103505.csv')
dt26_salaries_entir_01 = pd.read_csv('          _20240427104104.csv')
dt27_income_job_typ_01 = pd.read_csv('          _20240427102358.
    ↪csv')
dt28_employment_eff_01 = pd.read_csv('          _2010__20240427103804.
    ↪csv')
dt29_income_median__01 = pd.read_csv('          _20240427104029.csv')
dt30_ROK_demo_____01 = pd.read_csv('ROK_income_welfare.csv')
dt31_ROK_demo_____02 = pd.read_csv('          _20240427102712.csv')
dt32_ROK_population_01 = pd.read_csv('          _20240427102647.csv')
dt33_ROK_population_01 = pd.read_csv('2022_          .csv')
dt34_ROK_population_01 = pd.read_csv('2010_          .csv')
dt35_ROK_population_01 = pd.read_csv('2005_          .csv')
dt36_ROK_population_01 = pd.read_csv('2000_          .csv')
dt37_ROK_population_01 = pd.read_csv('1995_          .csv')
dt38_ROK_population_01 = pd.read_csv('1990_          .csv')

```

```

dt39_ROK_population_01 = pd.read_csv('1985_ _ .csv')
dt40_ROK_population_01 = pd.read_csv('1980_ _ .csv')
dt41_ROK_population_01 = pd.read_csv('1975_ _ .csv')
dt42_ROK_population_01 = pd.read_csv('1970_ _ .csv')
dt43_Seoul_apartment_01 = pd.read_csv(' _ _20240427103530.csv')
dt44_ROK_population_01 = pd.read_csv(' _ _20240427102647.csv')
dt45_RE_price_01 = pd.read_csv(' _ .csv')

```

```

[31]: # 04.02.01-02
      # return columns
      # dt01-dt44

print(' _ _ _ _ _')
print(' _ _ _ _ _dt01 _ _ _ _ _')
print('dt01 columns:',dt01_real_estate_pr_01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt02 _ _ _ _ _')
print('dt02 columns:',dt02_real_estate_se_01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt03 _ _ _ _ _')
print('dt03 columns:',dt03_stats_vital_kr_01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt04 _ _ _ _ _')
print('dt04 columns:',dt04_housing_census_01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt05 _ _ _ _ _')
print('dt05 columns:',dt05_housing_constr_01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt06 _ _ _ _ _')
print('dt06 columns:',dt06_household_empt_02.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt07 _ _ _ _ _')
print('dt07 columns:',dt07_housing_loss__01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt08 _ _ _ _ _')
print('dt08 columns:',dt08_housing_use_sv_01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt09 _ _ _ _ _')
print('dt09 columns:',dt09_jeonse_sales__01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt10 _ _ _ _ _')
print('dt10 columns:',dt10_jeonse_market__02.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt11 _ _ _ _ _')
print('dt11 columns:',dt11_birthrate_age__01.columns)
print(' _ _ _ _ _')
print(' _ _ _ _ _dt12 _ _ _ _ _')

```

```

print('dt12 columns:',dt12_birthrate_age__02.columns)
print('-----dt13-----')
print('dt13 columns:',dt13_pop_houshold__01.columns)
print('-----dt14-----')
print('dt14 columns:',dt14_pop_houshold__02.columns)
print('-----dt15-----')
print('dt15 columns:',dt15_pop_houshold__03.columns)
print('-----dt16-----')
print('dt16 columns:',dt16_pop_houshold__04.columns)
print('-----dt17-----')
print('dt17 columns:',dt17_pop_houshold__05.columns)
print('-----dt18-----')
print('dt18 columns:',dt18_pop_houshold__06.columns)
print('-----dt19-----')
print('dt19 columns:',dt19_pop_houshold__07.columns)
print('-----dt20-----')
print('dt20 columns:',dt20_pop_houshold__08.columns)
print('-----dt21-----')
print('dt21 columns:',dt21_pop_houshold__09.columns)
print('-----dt22-----')
print('dt22 columns:',dt22_pop_estimate__01.columns)
print('-----dt23-----')
print('dt23 columns:',dt23_pop_future_pro_01.columns)
print('-----dt24-----')
print('dt24 columns:',dt24_pop_future_pro_02.columns)
print('-----dt25-----')
print('dt25 columns:',dt25_pop_future_pro_02.columns)
print('-----dt26-----')
print('dt26 columns:',dt26_salaries_entir_01.columns)
print('-----dt27-----')
print('dt27 columns:',dt27_income_job_typ_01.columns)
print('-----')

```

```

print('-----dt28-----')
print('dt28 columns:',dt28_employment_eff_01.columns)
print('-----dt29-----')
print('dt29 columns:',dt29_income_median__01.columns)
print('-----dt30-----')
print('dt30 columns:',dt30_ROK_demo_____01.columns)
print('-----dt31-----')
print('dt31 columns:',dt31_ROK_demo_____02.columns)
print('-----dt32-----')
print('dt32 columns:',dt32_ROK_population_01.columns)
print('-----dt33-----')
print('dt33 columns:',dt33_ROK_population_01.columns)
print('-----dt34-----')
print('dt34 columns:',dt34_ROK_population_01.columns)
print('-----dt35-----')
print('dt35 columns:',dt35_ROK_population_01.columns)
print('-----dt36-----')
print('dt36 columns:',dt36_ROK_population_01.columns)
print('-----dt37-----')
print('dt37 columns:',dt37_ROK_population_01.columns)
print('-----dt38-----')
print('dt38 columns:',dt38_ROK_population_01.columns)
print('-----dt39-----')
print('dt39 columns:',dt39_ROK_population_01.columns)
print('-----dt40-----')
print('dt40 columns:',dt40_ROK_population_01.columns)
print('-----dt41-----')
print('dt41 columns:',dt41_ROK_population_01.columns)
print('-----dt42-----')
print('dt42 columns:',dt42_ROK_population_01.columns)
print('-----dt43-----')
print('dt43 columns:',dt43_Seoul_apr_____01.columns)

```

```

print('
print('-----dt44-----
print('dt44 columns:',dt44_R0K_population_01.columns)
print('
print('-----dt45-----
print('dt45 columns:',dt45_RE_price_____01.columns)

```

```

-----
-----dt01-----
-----
dt01 columns: Index([' (1)', ' (2)', '2023. 10', '2023. 10.1', '2023. 11',
                    '2023. 11.1', '2023. 12', '2023. 12.1'],
                    dtype='object')

```

```

-----
-----dt02-----
-----
dt02 columns: Index(['id', 'lat', 'lng', 'households', 'buildDate', 'score',
                    'm2', 'p',
                    'min_sales', 'max_sales', 'avg_sales'],
                    dtype='object')

```

```

-----
-----dt03-----
-----
dt03 columns: Index(['By items', '2021', '2022', '2023 p'], dtype='object')

```

```

-----
-----dt04-----
-----
dt04 columns: Index(['By administrative divisions', 'By the kind of house',
                    'By the number of dwelling households', '2010', '2010.1', '2010.2',
                    '2010.3', '2010.4', '2010.5', '2010.6', '2010.7', '2010.8', '2010.9'],
                    dtype='object')

```

```

-----
-----dt05-----
-----
dt05 columns: Index(['Category(1)', 'Category(1).1', '2023.12', '2023.12.1',
                    '2023.12.2',
                    '2023.12.3', '2023.12.4', '2023.12.5', '2023.12.6', '2023.12.7',
                    '2023.12.8', '2023.12.9', '2023.12.10', '2023.12.11', '2023.12.12',
                    '2023.12.13', '2023.12.14', '2023.12.15', '2023.12.16', '2023.12.17',
                    '2024.01', '2024.01.1', '2024.01.2', '2024.01.3', '2024.01.4',
                    '2024.01.5', '2024.01.6', '2024.01.7', '2024.01.8', '2024.01.9',
                    '2024.01.10', '2024.01.11', '2024.01.12', '2024.01.13', '2024.01.14',
                    '2024.01.15', '2024.01.16', '2024.01.17', '2024.02', '2024.02.1',

```

```

        '2024.02.2', '2024.02.3', '2024.02.4', '2024.02.5', '2024.02.6',
        '2024.02.7', '2024.02.8', '2024.02.9', '2024.02.10', '2024.02.11',
        '2024.02.12', '2024.02.13', '2024.02.14', '2024.02.15', '2024.02.16',
        '2024.02.17'],
        dtype='object')

-----
-----dt06-----
-----
dt06 columns: Index(['Area(1)', 'Size(1)', '2023.12', '2024.01', '2024.02'],
dtype='object')

-----
-----dt07-----
-----
dt07 columns: Index(['Classification(1)', '2022', '2022.1', '2022.2', '2022.3',
'2022.4',
        '2022.5'],
        dtype='object')

-----
-----dt08-----
-----
dt08 columns: Index(['Category(1)', 'Category(1).1', '2023.12', '2023.12.1',
'2023.12.2',
        '2023.12.3', '2023.12.4', '2023.12.5', '2023.12.6', '2023.12.7',
        '2023.12.8', '2023.12.9', '2023.12.10', '2023.12.11', '2023.12.12',
        '2023.12.13', '2023.12.14', '2023.12.15', '2023.12.16', '2023.12.17',
        '2024.01', '2024.01.1', '2024.01.2', '2024.01.3', '2024.01.4',
        '2024.01.5', '2024.01.6', '2024.01.7', '2024.01.8', '2024.01.9',
        '2024.01.10', '2024.01.11', '2024.01.12', '2024.01.13', '2024.01.14',
        '2024.01.15', '2024.01.16', '2024.01.17', '2024.02', '2024.02.1',
        '2024.02.2', '2024.02.3', '2024.02.4', '2024.02.5', '2024.02.6',
        '2024.02.7', '2024.02.8', '2024.02.9', '2024.02.10', '2024.02.11',
        '2024.02.12', '2024.02.13', '2024.02.14', '2024.02.15', '2024.02.16',
        '2024.02.17'],
        dtype='object')

-----
-----dt09-----
-----
dt09 columns: Index(['Type(1)', 'Region(1)', 'Region(2)', 'Region(3)',
'2023.10', '2023.11',
        '2023.12', '2024.01', '2024.02', '2024.03'],
        dtype='object')

-----
-----dt10-----

```

```
-----
dt10 columns: Index(['Type(1)', 'Region(1)', 'Region(2)', 'Region(3)',
'2023.10', '2023.11',
      '2023.12', '2024.01', '2024.02', '2024.03'],
      dtype='object')
```

```
-----
-----dt11-----
-----
```

```
dt11 columns: Index(['By variant', 'By age', '2022', '2023', '2024', '2025',
'2026', '2027',
      '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035', '2036',
      '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044', '2045',
      '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053', '2054',
      '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062', '2063',
      '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071', '2072'],
      dtype='object')
```

```
-----
-----dt12-----
-----
```

```
dt12 columns: Index(['By variant', 'By the type of population', '2022', '2023',
'2024',
      '2025', '2026', '2027', '2028', '2029', '2030', '2031', '2032', '2033',
      '2034', '2035', '2036', '2037', '2038', '2039', '2040', '2041', '2042',
      '2043', '2044', '2045', '2046', '2047', '2048', '2049', '2050', '2051',
      '2052', '2053', '2054', '2055', '2056', '2057', '2058', '2059', '2060',
      '2061', '2062', '2063', '2064', '2065', '2066', '2067', '2068', '2069',
      '2070', '2071', '2072'],
      dtype='object')
```

```
-----
-----dt13-----
-----
```

```
dt13 columns: Index(['By administrative divisions(eup, myeon, dong)', '2022',
'2022.1',
      '2022.2', '2022.3', '2022.4', '2022.5', '2022.6', '2022.7', '2022.8',
      '2022.9', '2022.10', '2022.11', '2022.12', '2022.13', '2022.14',
      '2022.15', '2022.16', '2022.17', '2022.18', '2022.19'],
      dtype='object')
```

```
-----
-----dt14-----
-----
```

```
dt14 columns: Index(['By administrative divisions', '2005', '2005.1', '2005.2',
'2005.3',
      '2005.4', '2005.5', '2005.6', '2005.7', '2005.8', '2005.9', '2005.10',
      '2005.11', '2005.12', '2005.13', '2005.14', '2005.15', '2005.16',
```

```

    '2005.17', '2005.18', '2005.19'],
    dtype='object')

-----
-----
-----dt15-----
-----
dt15 columns: Index(['By administrative divisions', '1970', '1970.1', '1970.2',
'1970.3',
    '1970.4', '1970.5', '1970.6', '1970.7', '1970.8', '1970.9', '1970.10',
    '1970.11'],
    dtype='object')

-----
-----
-----dt16-----
-----
dt16 columns: Index(['By administrative divisions', '1995', '1995.1', '1995.2',
'1995.3',
    '1995.4', '1995.5', '1995.6', '1995.7', '1995.8', '1995.9', '1995.10',
    '1995.11', '1995.12', '1995.13', '1995.14', '1995.15', '1995.16'],
    dtype='object')

-----
-----
-----dt17-----
-----
dt17 columns: Index(['By administrative divisions', '1990', '1990.1', '1990.2',
'1990.3',
    '1990.4', '1990.5', '1990.6', '1990.7', '1990.8', '1990.9', '1990.10',
    '1990.11', '1990.12', '1990.13', '1990.14', '1990.15', '1990.16'],
    dtype='object')

-----
-----
-----dt18-----
-----
dt18 columns: Index(['By administrative divisions', '1985', '1985.1', '1985.2',
'1985.3',
    '1985.4', '1985.5', '1985.6', '1985.7', '1985.8', '1985.9', '1985.10',
    '1985.11', '1985.12', '1985.13', '1985.14', '1985.15', '1985.16'],
    dtype='object')

-----
-----
-----dt19-----
-----
dt19 columns: Index(['By administrative divisions', '1980', '1980.1', '1980.2',
'1980.3',
    '1980.4', '1980.5', '1980.6', '1980.7', '1980.8', '1980.9', '1980.10',
    '1980.11', '1980.12', '1980.13', '1980.14', '1980.15', '1980.16'],
    dtype='object')

```



```

-----
-----dt20-----
-----
dt20 columns: Index(['By administrative divisions', '1975', '1975.1', '1975.2',
'1975.3',
      '1975.4', '1975.5', '1975.6', '1975.7', '1975.8', '1975.9', '1975.10',
      '1975.11'],
      dtype='object')

-----
-----dt21-----
-----
dt21 columns: Index(['By administrative divisions', '2000', '2000.1', '2000.2',
'2000.3',
      '2000.4', '2000.5', '2000.6', '2000.7', '2000.8', '2000.9', '2000.10',
      '2000.11', '2000.12', '2000.13', '2000.14', '2000.15', '2000.16',
      '2000.17', '2000.18', '2000.19'],
      dtype='object')

-----
-----dt22-----
-----
dt22 columns: Index(['By variant', 'By gender', 'By age', '2022', '2023',
'2024', '2025',
      '2026', '2027', '2028', '2029', '2030', '2031', '2032', '2033', '2034',
      '2035', '2036', '2037', '2038', '2039', '2040', '2041', '2042', '2043',
      '2044', '2045', '2046', '2047', '2048', '2049', '2050', '2051', '2052',
      '2053', '2054', '2055', '2056', '2057', '2058', '2059', '2060', '2061',
      '2062', '2063', '2064', '2065', '2066', '2067', '2068', '2069', '2070',
      '2071', '2072'],
      dtype='object')

-----
-----dt23-----
-----
dt23 columns: Index(['By variant', 'By the cause of population change', '2022',
'2023',
      '2024', '2025', '2026', '2027', '2028', '2029', '2030', '2031', '2032',
      '2033', '2034', '2035', '2036', '2037', '2038', '2039', '2040', '2041',
      '2042', '2043', '2044', '2045', '2046', '2047', '2048', '2049', '2050',
      '2051', '2052', '2053', '2054', '2055', '2056', '2057', '2058', '2059',
      '2060', '2061', '2062', '2063', '2064', '2065', '2066', '2067', '2068',
      '2069', '2070', '2071', '2072'],
      dtype='object')

-----
-----dt24-----
-----

```

```
dt24 columns: Index(['By scenarios', 'By the structure of population', '2070',
                    '2071',
                    '2072'],
                    dtype='object')
```

```
-----dt25-----
```

```
dt25 columns: Index(['By variant', 'By the structure of population and
dependency ratio',
                    '2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030',
                    '2031', '2032', '2033', '2034', '2035', '2036', '2037', '2038', '2039',
                    '2040', '2041', '2042', '2043', '2044', '2045', '2046', '2047', '2048',
                    '2049', '2050', '2051', '2052', '2053', '2054', '2055', '2056', '2057',
                    '2058', '2059', '2060', '2061', '2062', '2063', '2064', '2065', '2066',
                    '2067', '2068', '2069', '2070', '2071', '2072'],
                    dtype='object')
```

```
-----dt26-----
```

```
dt26 columns: Index(['Personal Loans(1)', 'Personal Loans(2)', '2022.1/2',
                    '2022.2/2'], dtype='object')
```

```
-----dt27-----
```

```
dt27 columns: Index(['Classification of Si, Do(1)', 'Classification of
Income(1)', '2019',
                    '2019.1', '2019.2', '2019.3', '2019.4', '2019.5'],
                    dtype='object')
```

```
-----dt28-----
```

```
dt28 columns: Index(['Classification(1)', '2022', '2022.1', '2022.2', '2022.3',
                    '2022.4'], dtype='object')
```

```
-----dt29-----
```

```
dt29 columns: Index(['Item', '2018', '2019', '2020', '2021', '2022'],
                    dtype='object')
```

```
-----dt30-----
```

```
dt30 columns: Index(['id', 'year', 'wave', 'region', 'income', 'family_member',
```

```

'gender',
    'year_born', 'education_level', 'marriage', 'religion', 'occupation',
    'company_size', 'reason_none_worker'],
    dtype='object')

-----
-----dt31-----
-----
dt31 columns: Index(['By variant', 'By the structure of population and
dependency ratio',
    '2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030',
    '2031', '2032', '2033', '2034', '2035', '2036', '2037', '2038', '2039',
    '2040', '2041', '2042', '2043', '2044', '2045', '2046', '2047', '2048',
    '2049', '2050', '2051', '2052', '2053', '2054', '2055', '2056', '2057',
    '2058', '2059', '2060', '2061', '2062', '2063', '2064', '2065', '2066',
    '2067', '2068', '2069', '2070', '2071', '2072'],
    dtype='object')

-----
-----dt32-----
-----
dt32 columns: Index(['Date', 'Region', 'Birth', 'Birth_rate', 'Death',
'Death_rate',
    'Divorce', 'Divorce_rate', 'Marriage', 'Marriage_rate',
    'Natural_growth', 'Natural_growth_rate'],
    dtype='object')

-----
-----dt33-----
-----
dt33 columns: Index(['By administrative divisions(eup, myeon, dong)', '2022',
'2022.1',
    '2022.2', '2022.3', '2022.4', '2022.5', '2022.6', '2022.7', '2022.8',
    '2022.9', '2022.10', '2022.11', '2022.12', '2022.13', '2022.14',
    '2022.15', '2022.16', '2022.17', '2022.18', '2022.19'],
    dtype='object')

-----
-----dt34-----
-----
dt34 columns: Index(['By administrative divisions(eup, myeon, dong)', '2010',
'2010.1',
    '2010.2', '2010.3', '2010.4', '2010.5', '2010.6', '2010.7', '2010.8',
    '2010.9', '2010.10', '2010.11', '2010.12', '2010.13', '2010.14',
    '2010.15', '2010.16', '2010.17', '2010.18', '2010.19'],
    dtype='object')

```

```

-----dt35-----
dt35 columns: Index(['By administrative divisions', '2005', '2005.1', '2005.2',
'2005.3',
    '2005.4', '2005.5', '2005.6', '2005.7', '2005.8', '2005.9', '2005.10',
    '2005.11', '2005.12', '2005.13', '2005.14', '2005.15', '2005.16',
    '2005.17', '2005.18', '2005.19'],
    dtype='object')

-----dt36-----
dt36 columns: Index(['By administrative divisions', '2000', '2000.1', '2000.2',
'2000.3',
    '2000.4', '2000.5', '2000.6', '2000.7', '2000.8', '2000.9', '2000.10',
    '2000.11', '2000.12', '2000.13', '2000.14', '2000.15', '2000.16',
    '2000.17', '2000.18', '2000.19'],
    dtype='object')

-----dt37-----
dt37 columns: Index(['By administrative divisions', '1995', '1995.1', '1995.2',
'1995.3',
    '1995.4', '1995.5', '1995.6', '1995.7', '1995.8', '1995.9', '1995.10',
    '1995.11', '1995.12', '1995.13', '1995.14', '1995.15', '1995.16'],
    dtype='object')

-----dt38-----
dt38 columns: Index(['By administrative divisions', '1990', '1990.1', '1990.2',
'1990.3',
    '1990.4', '1990.5', '1990.6', '1990.7', '1990.8', '1990.9', '1990.10',
    '1990.11', '1990.12', '1990.13', '1990.14', '1990.15', '1990.16'],
    dtype='object')

-----dt39-----
dt39 columns: Index(['By administrative divisions', '1985', '1985.1', '1985.2',
'1985.3',
    '1985.4', '1985.5', '1985.6', '1985.7', '1985.8', '1985.9', '1985.10',
    '1985.11', '1985.12', '1985.13', '1985.14', '1985.15', '1985.16'],
    dtype='object')

-----dt40-----

```

```

-----
dt40 columns: Index(['By administrative divisions', '1980', '1980.1', '1980.2',
'1980.3',
    '1980.4', '1980.5', '1980.6', '1980.7', '1980.8', '1980.9', '1980.10',
    '1980.11', '1980.12', '1980.13', '1980.14', '1980.15', '1980.16'],
    dtype='object')
-----
-----dt41-----
-----
dt41 columns: Index(['By administrative divisions', '1975', '1975.1', '1975.2',
'1975.3',
    '1975.4', '1975.5', '1975.6', '1975.7', '1975.8', '1975.9', '1975.10',
    '1975.11'],
    dtype='object')
-----
-----dt42-----
-----
dt42 columns: Index(['By administrative divisions', '1970', '1970.1', '1970.2',
'1970.3',
    '1970.4', '1970.5', '1970.6', '1970.7', '1970.8', '1970.9', '1970.10',
    '1970.11'],
    dtype='object')
-----
-----dt43-----
-----
dt43 columns: Index(['index', 'name', 'gugun', 'dong', 'buildDate', 'min_sales',
'max_sales',
    'avg_sales', 'area', 'floor', 'pricePerArea'],
    dtype='object')
-----
-----dt44-----
-----
dt44 columns: Index(['Date', 'Region', 'Birth', 'Birth_rate', 'Death',
'Death_rate',
    'Divorce', 'Divorce_rate', 'Marriage', 'Marriage_rate',
    'Natural_growth', 'Natural_growth_rate'],
    dtype='object')
-----
-----dt45-----
-----
dt45 columns: Index(['Region', 'By Housing Type', '1986.01', '1986.02',
'1986.03', '1986.04',
    '1986.05', '1986.06', '1986.07', '1986.08',

```

```
...
'2012.03', '2012.04', '2012.05', '2012.06', '2012.07', '2012.08',
'2012.09', '2012.10', '2012.11', '2012.12'],
dtype='object', length=326)
```

```
[32]: # 04.02.01-02-01
# renaming and translating three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt01

dt01_real_estate_pr_01.rename(columns = {' (1)': 'region_01'}, inplace = True)
dt01_real_estate_pr_01.rename(columns = {' (2)': 'region_02'}, inplace = True)
dt01_real_estate_pr_01.rename(columns = {'2023. 10': '2023_10_00'}, inplace =
↳ True)
dt01_real_estate_pr_01.rename(columns = {'2023. 10.1': '2023_10_01'}, inplace =
↳ True)
dt01_real_estate_pr_01.rename(columns = {'2023. 11': '2023_11_00'}, inplace =
↳ True)
dt01_real_estate_pr_01.rename(columns = {'2023. 11.1': '2023_11_01'}, inplace =
↳ True)
dt01_real_estate_pr_01.rename(columns = {'2023. 12': '2023_12_00'}, inplace =
↳ True)
dt01_real_estate_pr_01.rename(columns = {'2023. 12.1': '2023_12_01'}, inplace =
↳ True)
print('dt01 columns:', dt01_real_estate_pr_01.columns)
```

```
dt01 columns: Index(['region_01', 'region_02', '2023_10_00', '2023_10_01',
'2023_11_00',
'2023_11_01', '2023_12_00', '2023_12_01'],
dtype='object')
```

```
[33]: # 04.02.01-02-02
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt02

dt02_real_estate_se_01.rename(columns = {'id': 'id'}, inplace = True)
dt02_real_estate_se_01.rename(columns = {'lat': 'lat'}, inplace = True)
dt02_real_estate_se_01.rename(columns = {'lng': 'lon'}, inplace = True)
dt02_real_estate_se_01.rename(columns = {'households': 'households'}, inplace =
↳ True)
dt02_real_estate_se_01.rename(columns = {'build_date': 'date_build'}, inplace =
↳ True)
dt02_real_estate_se_01.rename(columns = {'score': 'score'}, inplace = True)
dt02_real_estate_se_01.rename(columns = {'m2': 'm2'}, inplace = True)
```

```

dt02_real_estate_se_01.rename(columns = {'p':'pyung'}, inplace = True)
dt02_real_estate_se_01.rename(columns = {'min_sales':'sales_min'}, inplace = 
↳True)
dt02_real_estate_se_01.rename(columns = {'max_sales':'sales_max'}, inplace = 
↳True)
dt02_real_estate_se_01.rename(columns = {'avg_sales':'sales_ave'}, inplace = 
↳True)
print('dt02 columns:',dt02_real_estate_se_01.columns)

```

```

dt02 columns: Index(['id', 'lat', 'lon', 'households', 'buildDate', 'score',
'm2', 'pyung',
'sales_min', 'sales_max', 'sales_ave'],
dtype='object')

```

```

[34]: # 04.02.01-02-03
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt03

dt03_stats_vital_kr_01.rename(columns = {'By items':'item'}, inplace = True)
dt03_stats_vital_kr_01.rename(columns = {'2021':'2021'}, inplace = True)
dt03_stats_vital_kr_01.rename(columns = {'2022':'2022'}, inplace = True)
dt03_stats_vital_kr_01.rename(columns = {'2023 p':'2023_p'}, inplace = True)
print('dt03 columns:',dt03_stats_vital_kr_01.columns)

```

```

dt03 columns: Index(['item', '2021', '2022', '2023 p'], dtype='object')

```

```

[35]: # 04.02.01-02-04
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt04

dt04_housing_census_01.rename(columns = {'By administrative divisions':
↳'admin_division'}, inplace = True)
dt04_housing_census_01.rename(columns = {'By the kind of house':
↳'type_housing'}, inplace = True)
dt04_housing_census_01.rename(columns = {'By the number of dwelling households':
↳'household_number_dwelling'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010':'2010_00'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.1':'2010_01'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.2':'2010_02'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.3':'2010_03'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.4':'2010_04'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.5':'2010_05'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.6':'2010_06'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.7':'2010_07'}, inplace = True)

```

```
dt04_housing_census_01.rename(columns = {'2010.8':'2010_08'}, inplace = True)
dt04_housing_census_01.rename(columns = {'2010.9':'2010_09'}, inplace = True)
print('dt04 columns:',dt04_housing_census_01.columns)
```

```
dt04 columns: Index(['admin_division', 'type_housing',
                    'household_number_dwelling',
                    '2010_00', '2010_01', '2010_02', '2010_03', '2010_04', '2010_05',
                    '2010_06', '2010_07', '2010_08', '2010_09'],
                    dtype='object')
```

```
[36]: # 04.02.01-02-05
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt05

dt05_housing_constr_01.rename(columns = {'Category(1)':'category_01'}, inplace=
    ↪ True)
dt05_housing_constr_01.rename(columns = {'Category(1).1':'category_02'},
    ↪ inplace = True)
dt05_housing_constr_01.rename(columns = {'2023.12':'2023_12_00'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.1':'2023_12_01'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.2':'2023_12_02'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.3':'2023_12_03'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.4':'2023_12_04'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.5':'2023_12_05'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.6':'2023_12_06'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.7':'2023_12_07'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.8':'2023_12_08'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.9':'2023_12_09'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.10':'2023_12_10'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.11':'2023_12_11'}, inplace =
    ↪ True)
dt05_housing_constr_01.rename(columns = {'2023.12.12':'2023_12_12'}, inplace =
    ↪ True)
```



```

dt05_housing_constr_01.rename(columns = {'2023.12.13':'2023_12_13'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2023.12.14':'2023_12_14'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2023.12.15':'2023_12_15'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2023.12.16':'2023_12_16'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2023.12.17':'2023_12_17'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01':'2024_01_00'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.1':'2024_01_01'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.2':'2024_01_02'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.3':'2024_01_03'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.4':'2024_01_04'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.5':'2024_01_05'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.6':'2024_01_06'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.7':'2024_01_07'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.8':'2024_01_08'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.9':'2024_01_09'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.10':'2024_01_10'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.11':'2024_01_11'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.12':'2024_01_12'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.13':'2024_01_13'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.14':'2024_01_14'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.15':'2024_01_15'}, inplace =_
↳True)
dt05_housing_constr_01.rename(columns = {'2024.01.16':'2024_01_16'}, inplace =_
↳True)

```

```

dt05_housing_constr_01.rename(columns = {'2024.01.17':'2024_01_17'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02':'2024_02_00'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.1':'2024_02_01'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.2':'2024_02_02'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.3':'2024_02_03'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.4':'2024_02_04'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.5':'2024_02_05'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.6':'2024_02_06'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.7':'2024_02_07'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.8':'2024_02_08'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.9':'2024_02_09'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.10':'2024_02_10'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.11':'2024_02_11'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.12':'2024_02_12'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.13':'2024_02_13'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.14':'2024_02_14'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.15':'2024_02_15'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.16':'2024_02_16'}, inplace =
↳True)
dt05_housing_constr_01.rename(columns = {'2024.02.17':'2024_02_17'}, inplace =
↳True)
print('dt05 columns:',dt05_housing_constr_01.columns)

```

```

dt05 columns: Index(['category_01', 'category_02', '2023_12_00', '2023_12_01',
'2023_12_02',
    '2023_12_03', '2023_12_04', '2023_12_05', '2023_12_06', '2023_12_07',
    '2023_12_08', '2023_12_09', '2023_12_10', '2023_12_11', '2023_12_12',
    '2023_12_13', '2023_12_14', '2023_12_15', '2023_12_16', '2023_12_17',
    '2024_01_00', '2024_01_01', '2024_01_02', '2024_01_03', '2024_01_04',

```

```

'2024_01_05', '2024_01_06', '2024_01_07', '2024_01_08', '2024_01_09',
'2024_01_10', '2024_01_11', '2024_01_12', '2024_01_13', '2024_01_14',
'2024_01_15', '2024_01_16', '2024_01_17', '2024_02_00', '2024_02_01',
'2024_02_02', '2024_02_03', '2024_02_04', '2024_02_05', '2024_02_06',
'2024_02_07', '2024_02_08', '2024_02_09', '2024_02_10', '2024_02_11',
'2024_02_12', '2024_02_13', '2024_02_14', '2024_02_15', '2024_02_16',
'2024_02_17'],
dtype='object')

```

```

[37]: # 04.02.01-02-06
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt06

dt06_household_empty_02.rename(columns = {'Area(1)': 'area'}, inplace = True)
dt06_household_empty_02.rename(columns = {'Size(1)': 'size'}, inplace = True)
dt06_household_empty_02.rename(columns = {'2023.12': '2023_12'}, inplace = True)
dt06_household_empty_02.rename(columns = {'2024.01': '2024_01'}, inplace = True)
dt06_household_empty_02.rename(columns = {'2024.02': '2024_02'}, inplace = True)
print('dt06 columns:', dt06_household_empty_02.columns)

```

```

dt06 columns: Index(['area', 'size', '2023_12', '2024_01', '2024_02'],
dtype='object')

```

```

[38]: # 04.02.01-02-07
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt07

dt07_housing_loss___01.rename(columns = {'Classification(1)': 'area'}, inplace = True)
dt07_housing_loss___01.rename(columns = {'2022': '2022_00'}, inplace = True)
dt07_housing_loss___01.rename(columns = {'2022.1': '2022_01'}, inplace = True)
dt07_housing_loss___01.rename(columns = {'2022.2': '2022_02'}, inplace = True)
dt07_housing_loss___01.rename(columns = {'2022.3': '2022_03'}, inplace = True)
dt07_housing_loss___01.rename(columns = {'2022.4': '2022_04'}, inplace = True)
dt07_housing_loss___01.rename(columns = {'2022.5': '2022_05'}, inplace = True)
print('dt07 columns:', dt07_housing_loss___01.columns)

```

```

dt07 columns: Index(['area', '2022_00', '2022_01', '2022_02', '2022_03',
'2022_04',
'2022_05'],
dtype='object')

```

```

[39]: # 04.02.01-02-08
# renaming three columns for clarity

```

```

# this step makes it easier to work with the data during transformations
# return new column names
# dt08

dt08_housing_use_sv_01.rename(columns = {'Category(1)': 'category_01'}, inplace=
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'Category(1).1': 'category_02'},
    ↪ inplace = True)
dt08_housing_use_sv_01.rename(columns = {'2023.12': '2023_12_00'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.1': '2023_12_01'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.2': '2023_12_02'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.3': '2023_12_03'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.4': '2023_12_04'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.5': '2023_12_05'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.6': '2023_12_06'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.7': '2023_12_07'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.8': '2023_12_08'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.9': '2023_12_09'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.10': '2023_12_10'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.11': '2023_12_11'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.12': '2023_12_12'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.13': '2023_12_13'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.14': '2023_12_14'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.15': '2023_12_15'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.16': '2023_12_16'}, inplace =
    ↪ True)
dt08_housing_use_sv_01.rename(columns = {'2023.12.17': '2023_12_17'}, inplace =
    ↪ True)

```

```

dt08_housing_use_sv_01.rename(columns = {'2024.01':'2024_01_00'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.1':'2024_01_01'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.2':'2024_01_02'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.3':'2024_01_03'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.4':'2024_01_04'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.5':'2024_01_05'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.6':'2024_01_06'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.7':'2024_01_07'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.8':'2024_01_08'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.9':'2024_01_09'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.10':'2024_01_10'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.11':'2024_01_11'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.12':'2024_01_12'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.13':'2024_01_13'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.14':'2024_01_14'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.15':'2024_01_15'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.16':'2024_01_16'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.01.17':'2024_01_17'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02':'2024_02_00'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.1':'2024_02_01'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.2':'2024_02_02'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.3':'2024_02_03'}, inplace =
↳True)

```

```

dt08_housing_use_sv_01.rename(columns = {'2024.02.4':'2024_02_04'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.5':'2024_02_05'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.6':'2024_02_06'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.7':'2024_02_07'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.8':'2024_02_08'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.9':'2024_02_09'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.10':'2024_02_10'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.11':'2024_02_11'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.12':'2024_02_12'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.13':'2024_02_13'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.14':'2024_02_14'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.15':'2024_02_15'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.16':'2024_02_16'}, inplace =
↳True)
dt08_housing_use_sv_01.rename(columns = {'2024.02.17':'2024_02_17'}, inplace =
↳True)
print('dt08 columns:',dt08_housing_use_sv_01.columns)

```

```

dt08 columns: Index(['category_01', 'category_02', '2023_12_00', '2023_12_01',
'2023_12_02',
    '2023_12_03', '2023_12_04', '2023_12_05', '2023_12_06', '2023_12_07',
    '2023_12_08', '2023_12_09', '2023_12_10', '2023_12_11', '2023_12_12',
    '2023_12_13', '2023_12_14', '2023_12_15', '2023_12_16', '2023_12_17',
    '2024_01_00', '2024_01_01', '2024_01_02', '2024_01_03', '2024_01_04',
    '2024_01_05', '2024_01_06', '2024_01_07', '2024_01_08', '2024_01_09',
    '2024_01_10', '2024_01_11', '2024_01_12', '2024_01_13', '2024_01_14',
    '2024_01_15', '2024_01_16', '2024_01_17', '2024_02_00', '2024_02_01',
    '2024_02_02', '2024_02_03', '2024_02_04', '2024_02_05', '2024_02_06',
    '2024_02_07', '2024_02_08', '2024_02_09', '2024_02_10', '2024_02_11',
    '2024_02_12', '2024_02_13', '2024_02_14', '2024_02_15', '2024_02_16',
    '2024_02_17'],
    dtype='object')

```

```
[40]: # 04.02.01-02-09
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt09

dt09_jeonse_sales__01.rename(columns = {'Type(1)': 'type'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'Region(1)': 'region_01'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'Region(2)': 'region_02'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'Region(3)': 'region_03'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'2023.10': '2023_10'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'2023.11': '2023_11'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'2023.12': '2023_12'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'2024.01': '2024_01'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'2024.02': '2024_02'}, inplace = True)
dt09_jeonse_sales__01.rename(columns = {'2024.03': '2024_03'}, inplace = True)
print('dt09 columns:', dt09_jeonse_sales__01.columns)
```

```
dt09 columns: Index(['type', 'region_01', 'region_02', 'region_03', '2023_10',
                    '2023_11',
                    '2023_12', '2024_01', '2024_02', '2024_03'],
                    dtype='object')
```

```
[41]: # 04.02.01-02-10
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt10

dt10_jeonse_market__02.rename(columns = {'Type(1)': 'type'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'Region(1)': 'region_01'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'Region(2)': 'region_02'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'Region(3)': 'region_03'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'2023.10': '2023_10'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'2023.11': '2023_11'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'2023.12': '2023_12'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'2024.01': '2024_01'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'2024.02': '2024_02'}, inplace = True)
dt10_jeonse_market__02.rename(columns = {'2024.03': '2024_03'}, inplace = True)
print('dt10 columns:', dt10_jeonse_market__02.columns)
```

```
dt10 columns: Index(['type', 'region_01', 'region_02', 'region_03', '2023_10',
```

```
'2023_11',
      '2023_12', '2024_01', '2024_02', '2024_03'],
      dtype='object')
```

```
[42]: # 04.02.01-02-11
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt11

dt11_birthingrate_age__01.rename(columns = {'By variant':'variant'}, inplace =_
↳True)
dt11_birthingrate_age__01.rename(columns = {'By age':'age'}, inplace = True)
print('dt11 columns:',dt11_birthingrate_age__01.columns)

dt11 columns: Index(['variant', 'age', '2022', '2023', '2024', '2025', '2026',
'2027',
      '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035', '2036',
      '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044', '2045',
      '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053', '2054',
      '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062', '2063',
      '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071', '2072'],
      dtype='object')
```

```
[43]: # 04.02.01-02-12
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt12

dt12_birthingrate_age__02.rename(columns = {'By variant':'variant'}, inplace =_
↳True)
dt12_birthingrate_age__02.rename(columns = {'By the type of population':
↳'population_type'}, inplace = True)
print('dt12 columns:',dt12_birthingrate_age__02.columns)

dt12 columns: Index(['variant', 'population_type', '2022', '2023', '2024',
'2025', '2026',
      '2027', '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035',
      '2036', '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044',
      '2045', '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053',
      '2054', '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062',
      '2063', '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071',
      '2072'],
      dtype='object')
```

```
[44]: # 04.02.01-02-13
# renaming three columns for clarity
```



```

# this step makes it easier to work with the data during transformations
# return new column names
# dt13

dt13_pop_houshold___01.rename(columns = {'By administrative divisions(eup,
↳myeon, dong)': 'admin_division'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022': '2022_00'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.1': '2022_01'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.2': '2022_02'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.3': '2022_03'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.4': '2022_04'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.5': '2022_05'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.6': '2022_06'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.7': '2022_07'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.8': '2022_08'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.9': '2022_09'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.10': '2022_10'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.11': '2022_11'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.12': '2022_12'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.13': '2022_13'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.14': '2022_14'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.15': '2022_15'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.16': '2022_16'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.17': '2022_17'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.18': '2022_18'}, inplace = True)
dt13_pop_houshold___01.rename(columns = {'2022.19': '2022_19'}, inplace = True)
print('dt13 columns:', dt13_pop_houshold___01.columns)

```

```

dt13 columns: Index(['admin_division', '2022_00', '2022_01', '2022_02',
'2022_03', '2022_04',
      '2022_05', '2022_06', '2022_07', '2022_08', '2022_09', '2022_10',
      '2022_11', '2022_12', '2022_13', '2022_14', '2022_15', '2022_16',
      '2022_17', '2022_18', '2022_19'],
      dtype='object')

```

```

[45]: # 04.02.01-02-14
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt14

dt14_pop_houshold___02.rename(columns = {'By administrative divisions':
↳'admin_division'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005': '2005_00'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.1': '2005_01'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.2': '2005_02'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.3': '2005_03'}, inplace = True)

```

```

dt14_pop_houshold___02.rename(columns = {'2005.4': '2005_04'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.5': '2005_05'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.6': '2005_06'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.7': '2005_07'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.8': '2005_08'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.9': '2005_09'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.10': '2005_10'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.11': '2005_11'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.12': '2005_12'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.13': '2005_13'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.14': '2005_14'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.15': '2005_15'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.16': '2005_16'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.17': '2005_17'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.18': '2005_18'}, inplace = True)
dt14_pop_houshold___02.rename(columns = {'2005.19': '2005_19'}, inplace = True)
print('dt14 columns:', dt14_pop_houshold___02.columns)

```

```

dt14 columns: Index(['admin_division', '2005_00', '2005_01', '2005_02',
'2005_03', '2005_04',
        '2005_05', '2005_06', '2005_07', '2005_08', '2005_09', '2005_10',
        '2005_11', '2005_12', '2005_13', '2005_14', '2005_15', '2005_16',
        '2005_17', '2005_18', '2005_19'],
        dtype='object')

```

```

[46]: # 04.02.01-02-15
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt15

dt15_pop_houshold___03.rename(columns = {'By administrative divisions':
↳ 'admin_division'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970': '1970_00'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.1': '1970_01'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.2': '1970_02'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.3': '1970_03'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.4': '1970_04'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.5': '1970_05'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.6': '1970_06'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.7': '1970_07'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.8': '1970_08'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.9': '1970_09'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.10': '1970_10'}, inplace = True)
dt15_pop_houshold___03.rename(columns = {'1970.11': '1970_11'}, inplace = True)
print('dt15 columns:', dt15_pop_houshold___03.columns)

```

```

dt15 columns: Index(['admin_division', '1970_00', '1970_01', '1970_02',

```

```
'1970_03', '1970_04',
    '1970_05', '1970_06', '1970_07', '1970_08', '1970_09', '1970_10',
    '1970_11'],
    dtype='object')
```

```
[47]: # 04.02.01-02-16
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt16

dt16_pop_houshold___04.rename(columns = {'By administrative divisions':
    ↪ 'admin_division'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995': '1995_00'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.1': '1995_01'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.2': '1995_02'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.3': '1995_03'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.4': '1995_04'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.5': '1995_05'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.6': '1995_06'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.7': '1995_07'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.8': '1995_08'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.9': '1995_09'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.10': '1995_10'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.11': '1995_11'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.12': '1995_12'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.13': '1995_13'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.14': '1995_14'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.15': '1995_15'}, inplace = True)
dt16_pop_houshold___04.rename(columns = {'1995.16': '1995_16'}, inplace = True)
print('dt16 columns:', dt16_pop_houshold___04.columns)
```

```
dt16 columns: Index(['admin_division', '1995_00', '1995_01', '1995_02',
    '1995_03', '1995_04',
    '1995_05', '1995_06', '1995_07', '1995_08', '1995_09', '1995_10',
    '1995_11', '1995_12', '1995_13', '1995_14', '1995_15', '1995_16'],
    dtype='object')
```

```
[48]: # 04.02.01-02-17
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt17

dt17_pop_houshold___05.rename(columns = {'By administrative divisions':
    ↪ 'admin_division'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990': '1990_00'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.1': '1990_01'}, inplace = True)
```

```

dt17_pop_houshold___05.rename(columns = {'1990.2': '1990_02'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.3': '1990_03'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.4': '1990_04'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.5': '1990_05'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.6': '1990_06'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.7': '1990_07'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.8': '1990_08'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.9': '1990_09'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.10': '1990_10'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.11': '1990_11'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.12': '1990_12'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.13': '1990_13'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.14': '1990_14'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.15': '1990_15'}, inplace = True)
dt17_pop_houshold___05.rename(columns = {'1990.16': '1990_16'}, inplace = True)
print('dt17 columns:', dt17_pop_houshold___05.columns)

```

```

dt17 columns: Index(['admin_division', '1990_00', '1990_01', '1990_02',
'1990_03', '1990_04',
        '1990_05', '1990_06', '1990_07', '1990_08', '1990_09', '1990_10',
        '1990_11', '1990_12', '1990_13', '1990_14', '1990_15', '1990_16'],
        dtype='object')

```

```

[49]: # 04.02.01-02-18
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt18

dt18_pop_houshold___06.rename(columns = {'By administrative divisions':
↳ 'admin_division'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985': '1985_00'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.1': '1985_01'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.2': '1985_02'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.3': '1985_03'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.4': '1985_04'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.5': '1985_05'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.6': '1985_06'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.7': '1985_07'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.8': '1985_08'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.9': '1985_09'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.10': '1985_10'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.11': '1985_11'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.12': '1985_12'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.13': '1985_13'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.14': '1985_14'}, inplace = True)
dt18_pop_houshold___06.rename(columns = {'1985.15': '1985_15'}, inplace = True)

```

```
dt18_pop_houshold___06.rename(columns = {'1985.16':'1985_16'}, inplace = True)
print('dt18 columns:',dt18_pop_houshold___06.columns)
```

```
dt18 columns: Index(['admin_division', '1985_00', '1985_01', '1985_02',
                    '1985_03', '1985_04',
                    '1985_05', '1985_06', '1985_07', '1985_08', '1985_09', '1985_10',
                    '1985_11', '1985_12', '1985_13', '1985_14', '1985_15', '1985_16'],
                    dtype='object')
```

```
[50]: # 04.02.01-02-19
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt19

dt19_pop_houshold___07.rename(columns = {'By administrative divisions':
    ↪ 'admin_division'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980':'1980_00'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.1':'1980_01'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.2':'1980_02'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.3':'1980_03'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.4':'1980_04'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.5':'1980_05'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.6':'1980_06'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.7':'1980_07'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.8':'1980_08'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.9':'1980_09'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.10':'1980_10'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.11':'1980_11'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.12':'1980_12'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.13':'1980_13'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.14':'1980_14'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.15':'1980_15'}, inplace = True)
dt19_pop_houshold___07.rename(columns = {'1980.16':'1980_16'}, inplace = True)
print('dt19 columns:',dt19_pop_houshold___07.columns)
```

```
dt19 columns: Index(['admin_division', '1980_00', '1980_01', '1980_02',
                    '1980_03', '1980_04',
                    '1980_05', '1980_06', '1980_07', '1980_08', '1980_09', '1980_10',
                    '1980_11', '1980_12', '1980_13', '1980_14', '1980_15', '1980_16'],
                    dtype='object')
```

```
[51]: # 04.02.01-02-20
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt20
```

```

dt20_pop_houshold___08.rename(columns = {'By administrative divisions':
    ↪'admin_division'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975':'1975_00'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.1':'1975_01'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.2':'1975_02'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.3':'1975_03'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.4':'1975_04'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.5':'1975_05'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.6':'1975_06'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.7':'1975_07'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.8':'1975_08'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.9':'1975_09'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.10':'1975_10'}, inplace = True)
dt20_pop_houshold___08.rename(columns = {'1975.11':'1975_11'}, inplace = True)
print('dt20 columns:',dt20_pop_houshold___08.columns)

```

```

dt20 columns: Index(['admin_division', '1975_00', '1975_01', '1975_02',
'1975_03', '1975_04',
      '1975_05', '1975_06', '1975_07', '1975_08', '1975_09', '1975_10',
      '1975_11'],
      dtype='object')

```

```

[52]: # 04.02.01-02-21
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt21

dt21_pop_houshold___09.rename(columns = {'By administrative divisions':
    ↪'admin_division'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000':'2000_00'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.1':'2000_01'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.2':'2000_02'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.3':'2000_03'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.4':'2000_04'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.5':'2000_05'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.6':'2000_06'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.7':'2000_07'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.8':'2000_08'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.9':'2000_09'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.10':'2000_10'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.11':'2000_11'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.12':'2000_12'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.13':'2000_13'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.14':'2000_14'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.15':'2000_15'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.16':'2000_16'}, inplace = True)

```



```
dt21_pop_houshold___09.rename(columns = {'2000.17':'2000_17'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.18':'2000_18'}, inplace = True)
dt21_pop_houshold___09.rename(columns = {'2000.19':'2000_19'}, inplace = True)
print('dt21 columns:',dt21_pop_houshold___09.columns)
```

```
dt21 columns: Index(['admin_division', '2000_00', '2000_01', '2000_02',
                    '2000_03', '2000_04',
                    '2000_05', '2000_06', '2000_07', '2000_08', '2000_09', '2000_10',
                    '2000_11', '2000_12', '2000_13', '2000_14', '2000_15', '2000_16',
                    '2000_17', '2000_18', '2000_19'],
                    dtype='object')
```

```
[53]: # 04.02.01-02-22
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt22

dt22_pop_estimate___01.rename(columns = {'By variant':'variant'}, inplace =
↳True)
dt22_pop_estimate___01.rename(columns = {'By gender':'gender'}, inplace = True)
dt22_pop_estimate___01.rename(columns = {'By age':'age'}, inplace = True)
print('dt22 columns:',dt22_pop_estimate___01.columns)
```

```
dt22 columns: Index(['variant', 'gender', 'age', '2022', '2023', '2024', '2025',
                    '2026',
                    '2027', '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035',
                    '2036', '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044',
                    '2045', '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053',
                    '2054', '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062',
                    '2063', '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071',
                    '2072'],
                    dtype='object')
```

```
[54]: # 04.02.01-02-23
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt23

dt23_pop_future_pro_01.rename(columns = {'By variant':'variant'}, inplace =
↳True)
dt23_pop_future_pro_01.rename(columns = {'By the cause of population change':
↳'cause'}, inplace = True)
print('dt23 columns:',dt23_pop_future_pro_01.columns)
```

```
dt23 columns: Index(['variant', 'cause', '2022', '2023', '2024', '2025', '2026',
                    '2027',
                    '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035', '2036',
```

```

'2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044', '2045',
'2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053', '2054',
'2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062', '2063',
'2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071', '2072'],
dtype='object')

```

```

[55]: # 04.02.01-02-24
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt24

dt24_pop_future_pro_02.rename(columns = {'By the structure of population':
↳ 'structure'}, inplace = True)
print('dt24 columns:', dt24_pop_future_pro_02.columns)

```

```

dt24 columns: Index(['By scenarios', 'structure', '2070', '2071', '2072'],
dtype='object')

```

```

[56]: # 04.02.01-02-25
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt25

dt25_pop_future_pro_02.rename(columns = {'By variant': 'variant'}, inplace =
↳ True)
dt25_pop_future_pro_02.rename(columns = {'By the structure of population and
↳ dependency ratio': 'dependency_ratio'}, inplace = True)
print('dt25 columns:', dt25_pop_future_pro_02.columns)

```

```

dt25 columns: Index(['variant', 'dependency_ratio', '2022', '2023', '2024',
'2025', '2026',
'2027', '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035',
'2036', '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044',
'2045', '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053',
'2054', '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062',
'2063', '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071',
'2072'],
dtype='object')

```

```

[57]: # 04.02.01-02-26
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt26

```



```

dt26_salaries_entir_01.rename(columns = {'Personal Loans(1)':
    ↳'loans_personal_01'}, inplace = True)
dt26_salaries_entir_01.rename(columns = {'Personal Loans(2)':
    ↳'loans_personal_02'}, inplace = True)
dt26_salaries_entir_01.rename(columns = {'2022.1/2':'2022_01_02'}, inplace =
    ↳True)
dt26_salaries_entir_01.rename(columns = {'2022.2/2':'2022_02_02'}, inplace =
    ↳True)
print('dt26 columns:',dt26_salaries_entir_01.columns)

```

```

dt26 columns: Index(['loans_personal_01', 'loans_personal_02', '2022_01_02',
'2022_02_02'], dtype='object')

```

```

[58]: # 04.02.01-02-27
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt27

```

```

dt27_income_job_typ_01.rename(columns = {'Classification of Si, Do(1)':
    ↳'classification_si_do'}, inplace = True)
dt27_income_job_typ_01.rename(columns = {'Classification of Income(1)':
    ↳'classification_income'}, inplace = True)
dt27_income_job_typ_01.rename(columns = {'2019':'2019_00'}, inplace = True)
dt27_income_job_typ_01.rename(columns = {'2019.1':'2019_01'}, inplace = True)
dt27_income_job_typ_01.rename(columns = {'2019.2':'2019_02'}, inplace = True)
dt27_income_job_typ_01.rename(columns = {'2019.3':'2019_03'}, inplace = True)
dt27_income_job_typ_01.rename(columns = {'2019.4':'2019_04'}, inplace = True)
dt27_income_job_typ_01.rename(columns = {'2019.5':'2019_05'}, inplace = True)
print('dt27 columns:',dt27_income_job_typ_01.columns)

```

```

dt27 columns: Index(['classification_si_do', 'classification_income', '2019_00',
'2019_01',
'2019_02', '2019_03', '2019_04', '2019_05'],
dtype='object')

```

```

[59]: # 04.02.01-02-28
# renaming three columns for clarity
# this step makes it easier to work with the data during transformations
# return new column names
# dt28

```

```

dt28_employment_eff_01.rename(columns = {'Classification(1)':'classification'},
    ↳inplace = True)
dt28_employment_eff_01.rename(columns = {'2022':'2022_00'}, inplace = True)
dt28_employment_eff_01.rename(columns = {'2022.1':'2022_01'}, inplace = True)
dt28_employment_eff_01.rename(columns = {'2022.2':'2022_02'}, inplace = True)

```

```
dt28_employment_eff_01.rename(columns = {'2022.3': '2022_03'}, inplace = True)
dt28_employment_eff_01.rename(columns = {'2022.4': '2022_04'}, inplace = True)
print('dt28 columns:', dt28_employment_eff_01.columns)
```

```
dt28 columns: Index(['classification', '2022_00', '2022_01', '2022_02',
                    '2022_03',
                    '2022_04'],
                    dtype='object')
```

[]:

```
[60]: # 04.02.01-02-29
      # renaming three columns for clarity
      # this step makes it easier to work with the data during transformations
      # return new column names
      # dt29
```

```
dt29_income_median__01.rename(columns = {'Item': 'item'}, inplace = True)
print('dt29 columns:', dt29_income_median__01.columns)
```

```
dt29 columns: Index(['item', '2018', '2019', '2020', '2021', '2022'],
                    dtype='object')
```

```
[61]: # 04.02.01-02-30
      # renaming three columns for clarity
      # this step makes it easier to work with the data during transformations
      # dt30
```

```
dt30_ROK_demo____01.rename(columns = {"family_member": "fam_mem_num"},
                             inplace = True)
dt30_ROK_demo____01.rename(columns = {"year_born": "birth_year"}, inplace = True)
dt30_ROK_demo____01.rename(columns = {"reason_none_worker": "unemployment_reason"},
                             inplace = True)
print('dt30 columns:', dt30_ROK_demo____01.columns)
```

```
dt30 columns: Index(['id', 'year', 'wave', 'region', 'income', 'fam_mem_num',
                    'gender',
                    'birth_year', 'education_level', 'marriage', 'religion', 'occupation',
                    'company_size', 'unemployment_reason'],
                    dtype='object')
```

```
[62]: # 04.02.01-02-30-01
      # preparing data for modeling
      # add index column
      # index column to select specific rows
      # dt30
```

```
dt30_ROK_demo_____01.insert(0, 'index', range(0, 0 +  
↳len(dt30_ROK_demo_____01)))
```

```
[63]: # 04.02.01-02-30-02  
# preparing data for modeling  
# create dummy variables  
# due to returning boolean values, converting dummies to integers  
# dt30  
  
dt30_dv = pd.get_dummies(dt30_ROK_demo_____01, drop_first = True, dtype = int)
```

```
[64]: # 04.02.01-02-30-03  
# preparing data for modeling  
# focus on marriage rate as it correlates with housing  
# split data  
# target = marriage  
# marriage = variable 01  
# select columns  
# for dataset dt30 containing income-level data  
# dt30  
  
dt30_x01 = dt30_dv.drop(['marriage'], axis = 1)  
dt30_y01 = dt30_dv['marriage']
```

```
[65]: # 04.02.01-02-30-04  
# preparing data for modeling  
# focus on marriage rate as it correlates with housing  
# target = marriage  
# marriage = variable 01  
# split into train and test  
# for dataset dt30 containing income-level data  
# dt30  
  
dt30_x01_trn, dt30_x01_tst, dt30_y01_trn, dt30_y01_tst =  
↳train_test_split(dt30_x01, dt30_y01, test_size = 0.3, random_state = 0)
```

```
[66]: # 04.02.01-02-30-05  
# preparing data for modeling  
# focus on marriage rate as it correlates with housing  
# target = marriage  
# marriage = variable 01  
# assign regression variable  
# for dataset dt30 containing income-level data  
# dt30  
  
dt30_lr01 = LinearRegression()
```

```
[67]: # 04.02.01-02-30-06
# fit data for modeling
# focus on marriage rate as it correlates with housing
# target = marriage
# marriage = variable 01
# fit variables to model
# for dataset dt30 containing income-level data
# dt30

dt30_lr01.fit(dt30_x01_trn, dt30_y01_trn)
```

```
[67]: LinearRegression()
```

```
[68]: # 04.02.01-02-30-07
# predict data for modeling
# focus on marriage rate as it correlates with housing
# target = marriage
# marriage = variable 01
# fit variables to model
# for dataset dt30 containing income-level data
# dt30

dt30_y01_pdct = dt30_lr01.predict(dt30_x01_tst)
```

```
[69]: # 04.02.01-02-30-08
# preparing data for modeling
# focus on marriage rate as it correlates with housing
# target = marriage
# marriage = variable 01
# assign variable for rmse and r2
# for dataset dt30 containing income-level data
# dt30

dt30_rmse01 = np.sqrt(mean_squared_error(dt30_y01_tst, dt30_y01_pdct))
dt30_r201 = r2_score(dt30_y01_tst, dt30_y01_pdct)
```

```
[70]: # 04.02.01-02-30-09
# run model
# focus on marriage rate as it correlates with housing
# target = marriage
# marriage = variable 01
# return rmse and r2 dt04
# rmse: 0.8258674228682785
# r2: 0.4301953970153479
# for dataset dt30 containing income-level data
# dt30
```

```
print(f'rmse: {dt30_rmse01}')
```

```
print(f'r2: {dt30_r201}')
```

rmse: 0.8258674228682786

r2: 0.4301953970153478

```
[71]: # 04.02.01-02-30-10
# focus on marriage rate as it correlates with housing
# assign variable for pca
# marriage = variable 01
# for dataset dt30 containing income-level data
# dt30
```

```
pca = PCA(.9)
```

```
[72]: # 04.02.01-02-30-11
# focus on marriage rate as it correlates with housing
# calculate pca
# marriage = variable 01
# for dataset dt30 containing income-level data
# dt30
```

```
pca.fit(dt30_x01_trn)
dt30_x01_pca_trn = pca.transform(dt30_x01_trn)
dt30_x01_pca_tst = pca.transform(dt30_x01_tst)
```

```
[73]: # 04.02.01-02-30-12
# focus on marriage rate as it correlates with housing
# return pca calculation matrix
# marriage = variable 01
# for dataset dt30 containing income-level data
# dt30

print(f'features in pca matrix: {dt30_x01_pca_trn.shape[1]}')
```

features in pca matrix: 1

```
[74]: # 04.02.01-02-30-13
# preparing data for modeling
# focus on education level as it correlates with housing
# target = education level
# education level = variable 02
# select columns
# for dataset dt30 containing education-level data
# dt30
```

```
dt30_x02 = dt30_dv.drop(['education_level'], axis = 1)
dt30_y02 = dt30_dv['education_level']
```

```
[75]: # 04.02.01-02-30-14
# preparing data for modeling
# focus on education level as it correlates with housing
# target = education level
# education level = variable 02
# split into train and test
# for dataset dt30 containing education-level data
# dt30

dt30_x02_trn, dt30_x02_tst, dt30_y02_trn, dt30_y02_tst = 
↳ train_test_split(dt30_x02, dt30_y02, test_size = 0.3, random_state = 0)
```

```
[76]: # 04.02.01-02-30-15
# preparing data for modeling
# focus on education level as it correlates with housing
# target = education level
# education level = variable 02
# assign regression variable
# for dataset dt30 containing education-level data
# dt30

dt30_lr02 = LinearRegression()
```

```
[77]: # 04.02.01-02-30-16
# fit data for modeling
# focus on education level as it correlates with housing
# target = education level
# education level = variable 02
# fit variables to model
# for dataset dt30 containing education-level data
# dt30

dt30_lr02.fit(dt30_x02_trn, dt30_y02_trn)
```

```
[77]: LinearRegression()
```

```
[78]: # 04.02.01-02-30-17
# predict data for modeling
# focus on education level as it correlates with housing
# target = education level
# education level = variable 02
# fit variables to model
# for dataset dt30 containing education-level data
# dt30

dt30_y02_pdct = dt30_lr02.predict(dt30_x02_tst)
```

```
[79]: # 04.02.01-02-30-18
# preparing data for modeling
# focus on education level as it correlates with housing
# target = education level
# education level = variable 02
# assign variable for rmse and r2
# for dataset dt30 containing education-level data
# dt30

dt30_rmse02 = np.sqrt(mean_squared_error(dt30_y02_tst, dt30_y02_pdct))
dt30_r202 = r2_score(dt30_y02_tst, dt30_y02_pdct)
```

```
[80]: # 04.02.01-02-30-19
# run model
# focus on education level as it correlates with housing
# target = education level
# education level = variable 02
# return rmse and r2 dt04
# rmse: 1.0642771383355127
# r2: 0.5929125769657074
# rmse is above 1.0
# r2 is below 0.7
# for dataset dt30 containing education-level data
# dt30

print(f'rmse: {dt30_rmse02}')
print(f'r2: {dt30_r202}')
```

```
rmse: 1.0642771383355296
r2: 0.5929125769656947
```

```
[81]: # 04.02.01-02-30-20
# focus on education level as it correlates with housing
# calculate pca
# target = education level
# education level = variable 02
# for dataset dt30 containing education-level data
# dt30

pca.fit(dt30_x02_trn)
dt30_x02_pca_trn = pca.transform(dt30_x02_trn)
dt30_x02_pca_tst = pca.transform(dt30_x02_tst)
```

```
[82]: # 04.02.01-02-30-21
# focus on education level as it correlates with housing
# return pca calculation matrix
# target = education level
# education level = variable 02
```

```
# for dataset dt30 containing education-level data
# dt30

print(f'features in pca matrix: {dt30_x02_pca_trn.shape[1]}')
```

features in pca matrix: 1

```
[83]: # 04.02.01-01
# rename columns
# dt45

dt45_RE_price_____01.rename(columns = {'Region':'region'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'By Housing Type':'type'}, inplace =
↳ True)
dt45_RE_price_____01.rename(columns = {'1986.01':'1986_01'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.02':'1986_02'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.03':'1986_03'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.04':'1986_04'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.05':'1986_05'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.06':'1986_06'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.07':'1986_07'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.08':'1986_08'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.09':'1986_09'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.10':'1986_10'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.11':'1986_11'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1986.12':'1986_12'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.01':'1987_01'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.02':'1987_02'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.03':'1987_03'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.04':'1987_04'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.05':'1987_05'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.06':'1987_06'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.07':'1987_07'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.08':'1987_08'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.09':'1987_09'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.10':'1987_10'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.11':'1987_11'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1987.12':'1987_12'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.01':'1988_01'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.02':'1988_02'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.03':'1988_03'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.04':'1988_04'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.05':'1988_05'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.06':'1988_06'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.07':'1988_07'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.08':'1988_08'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'1988.09':'1988_09'}, inplace = True)
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

dt45_RE_price_____01.rename(columns = {'2012.04':'2012_04'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.05':'2012_05'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.06':'2012_06'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.07':'2012_07'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.08':'2012_08'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.09':'2012_09'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.10':'2012_10'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.11':'2012_11'}, inplace = True)
dt45_RE_price_____01.rename(columns = {'2012.12':'2012_12'}, inplace = True)
print('dt45 columns:',dt45_RE_price_____01.columns)

```

```

dt45 columns: Index(['region', 'type', '1986_01', '1986_02', '1986_03',
'1986_04', '1986_05',
      '1986_06', '1986_07', '1986_08',
      ...
      '2012_03', '2012_04', '2012_05', '2012_06', '2012_07', '2012_08',
      '2012_09', '2012_10', '2012_11', '2012_12'],
      dtype='object', length=326)

```

```

[84]: # 04.02.01-02
      # return columns
      # dt01-dt44

print('_____')
print('-----dt01-----')
print('dt01 columns:',dt01_real_estate_pr_01.columns)
print('_____')
print('-----dt02-----')
print('dt02 columns:',dt02_real_estate_se_01.columns)
print('_____')
print('-----dt03-----')
print('dt03 columns:',dt03_stats_vital_kr_01.columns)
print('_____')
print('-----dt04-----')
print('dt04 columns:',dt04_housing_census_01.columns)
print('_____')
print('-----dt05-----')
print('dt05 columns:',dt05_housing_constr_01.columns)
print('_____')
print('-----dt06-----')
print('dt06 columns:',dt06_household_empty_02.columns)
print('_____')
print('-----dt07-----')
print('dt07 columns:',dt07_housing_loss__01.columns)
print('_____')
print('-----dt08-----')
print('dt08 columns:',dt08_housing_use_sv_01.columns)

```

```

print(' -----dt09-----')
print('dt09 columns:',dt09_jeonse_sales__01.columns)
print(' -----dt10-----')
print('dt10 columns:',dt10_jeonse_market__02.columns)
print(' -----dt11-----')
print('dt11 columns:',dt11_birthrate_age__01.columns)
print(' -----dt12-----')
print('dt12 columns:',dt12_birthrate_age__02.columns)
print(' -----dt13-----')
print('dt13 columns:',dt13_pop_houshold__01.columns)
print(' -----dt14-----')
print('dt14 columns:',dt14_pop_houshold__02.columns)
print(' -----dt15-----')
print('dt15 columns:',dt15_pop_houshold__03.columns)
print(' -----dt16-----')
print('dt16 columns:',dt16_pop_houshold__04.columns)
print(' -----dt17-----')
print('dt17 columns:',dt17_pop_houshold__05.columns)
print(' -----dt18-----')
print('dt18 columns:',dt18_pop_houshold__06.columns)
print(' -----dt19-----')
print('dt19 columns:',dt19_pop_houshold__07.columns)
print(' -----dt20-----')
print('dt20 columns:',dt20_pop_houshold__08.columns)
print(' -----dt21-----')
print('dt21 columns:',dt21_pop_houshold__09.columns)
print(' -----dt22-----')
print('dt22 columns:',dt22_pop_estimate__01.columns)
print(' -----dt23-----')
print('dt23 columns:',dt23_pop_future_pro_01.columns)
print(' -----dt24-----')

```



```

print('dt24 columns:',dt24_pop_future_pro_02.columns)
print(' -----dt25-----')
print('dt25 columns:',dt25_pop_future_pro_02.columns)
print(' -----dt26-----')
print('dt26 columns:',dt26_salaries_entir_01.columns)
print(' -----dt27-----')
print('dt27 columns:',dt27_income_job_typ_01.columns)
print(' -----dt28-----')
print('dt28 columns:',dt28_employment_eff_01.columns)
print(' -----dt29-----')
print('dt29 columns:',dt29_income_median__01.columns)
print(' -----dt30-----')
print('dt30 columns:',dt30_ROK_demo_____01.columns)
print(' -----dt31-----')
print('dt31 columns:',dt31_ROK_demo_____02.columns)
print(' -----dt32-----')
print('dt32 columns:',dt32_ROK_population_01.columns)
print(' -----dt33-----')
print('dt33 columns:',dt33_ROK_population_01.columns)
print(' -----dt34-----')
print('dt34 columns:',dt34_ROK_population_01.columns)
print(' -----dt35-----')
print('dt35 columns:',dt35_ROK_population_01.columns)
print(' -----dt36-----')
print('dt36 columns:',dt36_ROK_population_01.columns)
print(' -----dt37-----')
print('dt37 columns:',dt37_ROK_population_01.columns)
print(' -----dt38-----')
print('dt38 columns:',dt38_ROK_population_01.columns)
print(' -----dt39-----')
print('dt39 columns:',dt39_ROK_population_01.columns)
print(' -----')

```

```

print('-----dt40-----')
print('dt40 columns:',dt40_ROK_population_01.columns)
print('-----dt41-----')
print('dt41 columns:',dt41_ROK_population_01.columns)
print('-----dt42-----')
print('dt42 columns:',dt42_ROK_population_01.columns)
print('-----dt43-----')
print('dt43 columns:',dt43_Seoul_apartment_01.columns)
print('-----dt44-----')
print('dt44 columns:',dt44_ROK_population_01.columns)
print('-----dt45-----')
print('dt45 columns:',dt45_RE_price_01.columns)

```

```

-----
-----dt01-----
-----
dt01 columns: Index(['region_01', 'region_02', '2023_10_00', '2023_10_01',
'2023_11_00',
      '2023_11_01', '2023_12_00', '2023_12_01'],
      dtype='object')

```

```

-----dt02-----
-----
dt02 columns: Index(['id', 'lat', 'lon', 'households', 'buildDate', 'score',
'm2', 'pyung',
      'sales_min', 'sales_max', 'sales_ave'],
      dtype='object')

```

```

-----dt03-----
-----
dt03 columns: Index(['item', '2021', '2022', '2023 p'], dtype='object')

```

```

-----dt04-----
-----
dt04 columns: Index(['admin_division', 'type_housing',
'household_number_dwelling',
      '2010_00', '2010_01', '2010_02', '2010_03', '2010_04', '2010_05',
      '2010_06', '2010_07', '2010_08', '2010_09'],
      dtype='object')

```

```

-----
-----dt05-----
-----
dt05 columns: Index(['category_01', 'category_02', '2023_12_00', '2023_12_01',
'2023_12_02',
    '2023_12_03', '2023_12_04', '2023_12_05', '2023_12_06', '2023_12_07',
    '2023_12_08', '2023_12_09', '2023_12_10', '2023_12_11', '2023_12_12',
    '2023_12_13', '2023_12_14', '2023_12_15', '2023_12_16', '2023_12_17',
    '2024_01_00', '2024_01_01', '2024_01_02', '2024_01_03', '2024_01_04',
    '2024_01_05', '2024_01_06', '2024_01_07', '2024_01_08', '2024_01_09',
    '2024_01_10', '2024_01_11', '2024_01_12', '2024_01_13', '2024_01_14',
    '2024_01_15', '2024_01_16', '2024_01_17', '2024_02_00', '2024_02_01',
    '2024_02_02', '2024_02_03', '2024_02_04', '2024_02_05', '2024_02_06',
    '2024_02_07', '2024_02_08', '2024_02_09', '2024_02_10', '2024_02_11',
    '2024_02_12', '2024_02_13', '2024_02_14', '2024_02_15', '2024_02_16',
    '2024_02_17'],
    dtype='object')
-----
-----dt06-----
-----
dt06 columns: Index(['area', 'size', '2023_12', '2024_01', '2024_02'],
    dtype='object')
-----
-----dt07-----
-----
dt07 columns: Index(['area', '2022_00', '2022_01', '2022_02', '2022_03',
'2022_04',
    '2022_05'],
    dtype='object')
-----
-----dt08-----
-----
dt08 columns: Index(['category_01', 'category_02', '2023_12_00', '2023_12_01',
'2023_12_02',
    '2023_12_03', '2023_12_04', '2023_12_05', '2023_12_06', '2023_12_07',
    '2023_12_08', '2023_12_09', '2023_12_10', '2023_12_11', '2023_12_12',
    '2023_12_13', '2023_12_14', '2023_12_15', '2023_12_16', '2023_12_17',
    '2024_01_00', '2024_01_01', '2024_01_02', '2024_01_03', '2024_01_04',
    '2024_01_05', '2024_01_06', '2024_01_07', '2024_01_08', '2024_01_09',
    '2024_01_10', '2024_01_11', '2024_01_12', '2024_01_13', '2024_01_14',
    '2024_01_15', '2024_01_16', '2024_01_17', '2024_02_00', '2024_02_01',
    '2024_02_02', '2024_02_03', '2024_02_04', '2024_02_05', '2024_02_06',
    '2024_02_07', '2024_02_08', '2024_02_09', '2024_02_10', '2024_02_11',
    '2024_02_12', '2024_02_13', '2024_02_14', '2024_02_15', '2024_02_16',

```

```

        '2024_02_17'],
dtype='object')

-----
-----dt09-----
-----
dt09 columns: Index(['type', 'region_01', 'region_02', 'region_03', '2023_10',
'2023_11',
        '2023_12', '2024_01', '2024_02', '2024_03'],
dtype='object')

-----
-----dt10-----
-----
dt10 columns: Index(['type', 'region_01', 'region_02', 'region_03', '2023_10',
'2023_11',
        '2023_12', '2024_01', '2024_02', '2024_03'],
dtype='object')

-----
-----dt11-----
-----
dt11 columns: Index(['variant', 'age', '2022', '2023', '2024', '2025', '2026',
'2027',
        '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035', '2036',
'2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044', '2045',
'2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053', '2054',
'2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062', '2063',
'2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071', '2072'],
dtype='object')

-----
-----dt12-----
-----
dt12 columns: Index(['variant', 'population_type', '2022', '2023', '2024',
'2025', '2026',
        '2027', '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035',
'2036', '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044',
'2045', '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053',
'2054', '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062',
'2063', '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071',
'2072'],
dtype='object')

-----
-----dt13-----
-----
dt13 columns: Index(['admin_division', '2022_00', '2022_01', '2022_02',

```

```

'2022_03', '2022_04',
    '2022_05', '2022_06', '2022_07', '2022_08', '2022_09', '2022_10',
    '2022_11', '2022_12', '2022_13', '2022_14', '2022_15', '2022_16',
    '2022_17', '2022_18', '2022_19'],
    dtype='object')

-----
-----dt14-----
-----
dt14 columns: Index(['admin_division', '2005_00', '2005_01', '2005_02',
'2005_03', '2005_04',
    '2005_05', '2005_06', '2005_07', '2005_08', '2005_09', '2005_10',
    '2005_11', '2005_12', '2005_13', '2005_14', '2005_15', '2005_16',
    '2005_17', '2005_18', '2005_19'],
    dtype='object')

-----
-----dt15-----
-----
dt15 columns: Index(['admin_division', '1970_00', '1970_01', '1970_02',
'1970_03', '1970_04',
    '1970_05', '1970_06', '1970_07', '1970_08', '1970_09', '1970_10',
    '1970_11'],
    dtype='object')

-----
-----dt16-----
-----
dt16 columns: Index(['admin_division', '1995_00', '1995_01', '1995_02',
'1995_03', '1995_04',
    '1995_05', '1995_06', '1995_07', '1995_08', '1995_09', '1995_10',
    '1995_11', '1995_12', '1995_13', '1995_14', '1995_15', '1995_16'],
    dtype='object')

-----
-----dt17-----
-----
dt17 columns: Index(['admin_division', '1990_00', '1990_01', '1990_02',
'1990_03', '1990_04',
    '1990_05', '1990_06', '1990_07', '1990_08', '1990_09', '1990_10',
    '1990_11', '1990_12', '1990_13', '1990_14', '1990_15', '1990_16'],
    dtype='object')

-----
-----dt18-----
-----
dt18 columns: Index(['admin_division', '1985_00', '1985_01', '1985_02',
'1985_03', '1985_04',

```

```

        '1985_05', '1985_06', '1985_07', '1985_08', '1985_09', '1985_10',
        '1985_11', '1985_12', '1985_13', '1985_14', '1985_15', '1985_16'],
dtype='object')

-----
-----dt19-----
-----
dt19 columns: Index(['admin_division', '1980_00', '1980_01', '1980_02',
'1980_03', '1980_04',
        '1980_05', '1980_06', '1980_07', '1980_08', '1980_09', '1980_10',
        '1980_11', '1980_12', '1980_13', '1980_14', '1980_15', '1980_16'],
dtype='object')

-----
-----dt20-----
-----
dt20 columns: Index(['admin_division', '1975_00', '1975_01', '1975_02',
'1975_03', '1975_04',
        '1975_05', '1975_06', '1975_07', '1975_08', '1975_09', '1975_10',
        '1975_11'],
dtype='object')

-----
-----dt21-----
-----
dt21 columns: Index(['admin_division', '2000_00', '2000_01', '2000_02',
'2000_03', '2000_04',
        '2000_05', '2000_06', '2000_07', '2000_08', '2000_09', '2000_10',
        '2000_11', '2000_12', '2000_13', '2000_14', '2000_15', '2000_16',
        '2000_17', '2000_18', '2000_19'],
dtype='object')

-----
-----dt22-----
-----
dt22 columns: Index(['variant', 'gender', 'age', '2022', '2023', '2024', '2025',
'2026',
        '2027', '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035',
        '2036', '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044',
        '2045', '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053',
        '2054', '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062',
        '2063', '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071',
        '2072'],
dtype='object')

-----
-----dt23-----
-----

```

```
dt23 columns: Index(['variant', 'cause', '2022', '2023', '2024', '2025', '2026',
'2027',
    '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035', '2036',
    '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044', '2045',
    '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053', '2054',
    '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062', '2063',
    '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071', '2072'],
    dtype='object')
```

```
-----dt24-----
dt24 columns: Index(['By scenarios', 'structure', '2070', '2071', '2072'],
    dtype='object')
```

```
-----dt25-----
dt25 columns: Index(['variant', 'dependency_ratio', '2022', '2023', '2024',
'2025', '2026',
    '2027', '2028', '2029', '2030', '2031', '2032', '2033', '2034', '2035',
    '2036', '2037', '2038', '2039', '2040', '2041', '2042', '2043', '2044',
    '2045', '2046', '2047', '2048', '2049', '2050', '2051', '2052', '2053',
    '2054', '2055', '2056', '2057', '2058', '2059', '2060', '2061', '2062',
    '2063', '2064', '2065', '2066', '2067', '2068', '2069', '2070', '2071',
    '2072'],
    dtype='object')
```

```
-----dt26-----
dt26 columns: Index(['loans_personal_01', 'loans_personal_02', '2022_01_02',
'2022_02_02'], dtype='object')
```

```
-----dt27-----
dt27 columns: Index(['classification_si_do', 'classification_income', '2019_00',
'2019_01',
    '2019_02', '2019_03', '2019_04', '2019_05'],
    dtype='object')
```

```
-----dt28-----
dt28 columns: Index(['classification', '2022_00', '2022_01', '2022_02',
'2022_03',
    '2022_04'],
```

```

dtype='object')

-----
-----dt29-----
-----
dt29 columns: Index(['item', '2018', '2019', '2020', '2021', '2022'],
dtype='object')

-----
-----dt30-----
-----
dt30 columns: Index(['index', 'id', 'year', 'wave', 'region', 'income',
'fam_mem_num',
'gender', 'birth_year', 'education_level', 'marriage', 'religion',
'occupation', 'company_size', 'unemployment_reason'],
dtype='object')

-----
-----dt31-----
-----
dt31 columns: Index(['By variant', 'By the structure of population and
dependency ratio',
'2022', '2023', '2024', '2025', '2026', '2027', '2028', '2029', '2030',
'2031', '2032', '2033', '2034', '2035', '2036', '2037', '2038', '2039',
'2040', '2041', '2042', '2043', '2044', '2045', '2046', '2047', '2048',
'2049', '2050', '2051', '2052', '2053', '2054', '2055', '2056', '2057',
'2058', '2059', '2060', '2061', '2062', '2063', '2064', '2065', '2066',
'2067', '2068', '2069', '2070', '2071', '2072'],
dtype='object')

-----
-----dt32-----
-----
dt32 columns: Index(['Date', 'Region', 'Birth', 'Birth_rate', 'Death',
'Death_rate',
'Divorce', 'Divorce_rate', 'Marriage', 'Marriage_rate',
'Natural_growth', 'Natural_growth_rate'],
dtype='object')

-----
-----dt33-----
-----
dt33 columns: Index(['By administrative divisions(eup, myeon, dong)', '2022',
'2022.1',
'2022.2', '2022.3', '2022.4', '2022.5', '2022.6', '2022.7', '2022.8',
'2022.9', '2022.10', '2022.11', '2022.12', '2022.13', '2022.14',
'2022.15', '2022.16', '2022.17', '2022.18', '2022.19'],
dtype='object')

```



```

-----
-----dt34-----
-----
dt34 columns: Index(['By administrative divisions(eup, myeon, dong)', '2010',
'2010.1',
      '2010.2', '2010.3', '2010.4', '2010.5', '2010.6', '2010.7', '2010.8',
      '2010.9', '2010.10', '2010.11', '2010.12', '2010.13', '2010.14',
      '2010.15', '2010.16', '2010.17', '2010.18', '2010.19'],
      dtype='object')
-----
-----dt35-----
-----
dt35 columns: Index(['By administrative divisions', '2005', '2005.1', '2005.2',
'2005.3',
      '2005.4', '2005.5', '2005.6', '2005.7', '2005.8', '2005.9', '2005.10',
      '2005.11', '2005.12', '2005.13', '2005.14', '2005.15', '2005.16',
      '2005.17', '2005.18', '2005.19'],
      dtype='object')
-----
-----dt36-----
-----
dt36 columns: Index(['By administrative divisions', '2000', '2000.1', '2000.2',
'2000.3',
      '2000.4', '2000.5', '2000.6', '2000.7', '2000.8', '2000.9', '2000.10',
      '2000.11', '2000.12', '2000.13', '2000.14', '2000.15', '2000.16',
      '2000.17', '2000.18', '2000.19'],
      dtype='object')
-----
-----dt37-----
-----
dt37 columns: Index(['By administrative divisions', '1995', '1995.1', '1995.2',
'1995.3',
      '1995.4', '1995.5', '1995.6', '1995.7', '1995.8', '1995.9', '1995.10',
      '1995.11', '1995.12', '1995.13', '1995.14', '1995.15', '1995.16'],
      dtype='object')
-----
-----dt38-----
-----
dt38 columns: Index(['By administrative divisions', '1990', '1990.1', '1990.2',
'1990.3',
      '1990.4', '1990.5', '1990.6', '1990.7', '1990.8', '1990.9', '1990.10',
      '1990.11', '1990.12', '1990.13', '1990.14', '1990.15', '1990.16'],
      dtype='object')

```

```

-----
-----dt39-----
-----
dt39 columns: Index(['By administrative divisions', '1985', '1985.1', '1985.2',
'1985.3',
      '1985.4', '1985.5', '1985.6', '1985.7', '1985.8', '1985.9', '1985.10',
      '1985.11', '1985.12', '1985.13', '1985.14', '1985.15', '1985.16'],
      dtype='object')
-----
-----dt40-----
-----
dt40 columns: Index(['By administrative divisions', '1980', '1980.1', '1980.2',
'1980.3',
      '1980.4', '1980.5', '1980.6', '1980.7', '1980.8', '1980.9', '1980.10',
      '1980.11', '1980.12', '1980.13', '1980.14', '1980.15', '1980.16'],
      dtype='object')
-----
-----dt41-----
-----
dt41 columns: Index(['By administrative divisions', '1975', '1975.1', '1975.2',
'1975.3',
      '1975.4', '1975.5', '1975.6', '1975.7', '1975.8', '1975.9', '1975.10',
      '1975.11'],
      dtype='object')
-----
-----dt42-----
-----
dt42 columns: Index(['By administrative divisions', '1970', '1970.1', '1970.2',
'1970.3',
      '1970.4', '1970.5', '1970.6', '1970.7', '1970.8', '1970.9', '1970.10',
      '1970.11'],
      dtype='object')
-----
-----dt43-----
-----
dt43 columns: Index(['index', 'name', 'gugun', 'dong', 'buildDate', 'min_sales',
'max_sales',
      'avg_sales', 'area', 'floor', 'pricePerArea'],
      dtype='object')
-----
-----dt44-----
-----

```

```
dt44 columns: Index(['Date', 'Region', 'Birth', 'Birth_rate', 'Death',
'Death_rate',
'Divorce', 'Divorce_rate', 'Marriage', 'Marriage_rate',
'Natural_growth', 'Natural_growth_rate'],
dtype='object')
```

```
-----dt45-----
```

```
dt45 columns: Index(['region', 'type', '1986_01', '1986_02', '1986_03',
'1986_04', '1986_05',
'1986_06', '1986_07', '1986_08',
...
'2012_03', '2012_04', '2012_05', '2012_06', '2012_07', '2012_08',
'2012_09', '2012_10', '2012_11', '2012_12'],
dtype='object', length=326)
```

```
[85]: # 04.02.01-03
# return info
# dt01-dt44

print('-----dt01-----')
print('dt01 types:',dt01_real_estate_pr_01.info())
print('-----dt02-----')
print('dt02 types:',dt02_real_estate_se_01.info())
print('-----dt03-----')
print('dt03 types:',dt03_stats_vital_kr_01.info())
print('-----dt04-----')
print('dt04 types:',dt04_housing_census_01.info())
print('-----dt05-----')
print('dt05 types:',dt05_housing_constr_01.info())
print('-----dt06-----')
print('dt06 types:',dt06_household_empty_02.info())
print('-----dt07-----')
print('dt07 types:',dt07_housing_loss__01.info())
print('-----dt08-----')
print('dt08 types:',dt08_housing_use_sv_01.info())
print('-----dt09-----')
```

```

print('dt09 types:',dt09_jeonse_sales__01.info())
print('-----dt10-----')
print('dt10 types:',dt10_jeonse_market__02.info())
print('-----dt11-----')
print('dt11 types:',dt11_birthrate_age__01.info())
print('-----dt12-----')
print('dt12 types:',dt12_birthrate_age__02.info())
print('-----dt13-----')
print('dt13 types:',dt13_pop_houshold__01.info())
print('-----dt14-----')
print('dt14 types:',dt14_pop_houshold__02.info())
print('-----dt15-----')
print('dt15 types:',dt15_pop_houshold__03.info())
print('-----dt16-----')
print('dt16 types:',dt16_pop_houshold__04.info())
print('-----dt17-----')
print('dt17 types:',dt17_pop_houshold__05.info())
print('-----dt18-----')
print('dt18 types:',dt18_pop_houshold__06.info())
print('-----dt19-----')
print('dt19 types:',dt19_pop_houshold__07.info())
print('-----dt20-----')
print('dt20 types:',dt20_pop_houshold__08.info())
print('-----dt21-----')
print('dt21 types:',dt21_pop_houshold__09.info())
print('-----dt22-----')
print('dt22 types:',dt22_pop_estimate__01.info())
print('-----dt23-----')
print('dt23 types:',dt23_pop_future_pro_01.info())
print('-----dt24-----')
print('dt24 types:',dt24_pop_future_pro_02.info())
print('-----')

```

```

print('-----dt25-----')
print('dt25 types:',dt25_pop_future_pro_02.info())
print('-----dt26-----')
print('dt26 types:',dt26_salaries_entir_01.info())
print('-----dt27-----')
print('dt27 types:',dt27_income_job_typ_01.info())
print('-----dt28-----')
print('dt28 types:',dt28_employment_eff_01.info())
print('-----dt29-----')
print('dt29 types:',dt29_income_median_01.info())
print('-----dt30-----')
print('dt30 types:',dt30_ROK_demo_____01.info())
print('-----dt31-----')
print('dt31 types:',dt31_ROK_demo_____02.info())
print('-----dt32-----')
print('dt32 types:',dt32_ROK_population_01.info())
print('-----dt33-----')
print('dt33 types:',dt33_ROK_population_01.info())
print('-----dt34-----')
print('dt34 types:',dt34_ROK_population_01.info())
print('-----dt35-----')
print('dt35 types:',dt35_ROK_population_01.info())
print('-----dt36-----')
print('dt36 types:',dt36_ROK_population_01.info())
print('-----dt37-----')
print('dt37 types:',dt37_ROK_population_01.info())
print('-----dt38-----')
print('dt38 types:',dt38_ROK_population_01.info())
print('-----dt39-----')
print('dt39 types:',dt39_ROK_population_01.info())
print('-----dt40-----')
print('dt40 types:',dt40_ROK_population_01.info())

```

```

print(' -----dt41-----')
print('dt41 types:',dt41_ROK_population_01.info())
print(' -----dt42-----')
print('dt42 types:',dt42_ROK_population_01.info())
print(' -----dt43-----')
print('dt43 types:',dt43_Seoul_apartment_01.info())
print(' -----dt44-----')
print('dt44 types:',dt44_ROK_population_01.info())
print(' -----dt45-----')
print('dt45 types:',dt45_RE_price_01.info())

```

```

-----
-----dt01-----
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  -
0   region_01    27 non-null    object
1   region_02    27 non-null    object
2   2023_10_00   27 non-null    object
3   2023_10_01   27 non-null    object
4   2023_11_00   27 non-null    object
5   2023_11_01   27 non-null    object
6   2023_12_00   27 non-null    object
7   2023_12_01   27 non-null    object
dtypes: object(8)
memory usage: 1.8+ KB
dt01 types: None

```

```

-----dt02-----
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4021 entries, 0 to 4020
Data columns (total 11 columns):
#   Column      Non-Null Count  Dtype
---  -
0   id           4021 non-null   int64
1   lat          4021 non-null   float64
2   lon          4021 non-null   float64

```

```

3 households 4021 non-null int64
4 buildDate 4021 non-null int64
5 score 4021 non-null float64
6 m2 4021 non-null int64
7 pyung 4021 non-null int64
8 sales_min 3931 non-null float64
9 sales_max 3931 non-null float64
10 sales_ave 3931 non-null float64

```

dtypes: float64(6), int64(5)

memory usage: 345.7 KB

dt02 types: None

```

-----
-----dt03-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 16 entries, 0 to 15

Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype
0	item	16 non-null	object
1	2021	16 non-null	float64
2	2022	16 non-null	float64
3	2023 p)	16 non-null	object

dtypes: float64(2), object(2)

memory usage: 644.0+ bytes

dt03 types: None

```

-----
-----dt04-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 43 entries, 0 to 42

Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	admin_division	43 non-null	object
1	type_housing	43 non-null	object
2	household_number_dwelling	43 non-null	object
3	2010_00	43 non-null	object
4	2010_01	43 non-null	object
5	2010_02	43 non-null	object
6	2010_03	43 non-null	object
7	2010_04	43 non-null	object
8	2010_05	43 non-null	object
9	2010_06	43 non-null	object
10	2010_07	43 non-null	object
11	2010_08	43 non-null	object

```

12 2010_09                                43 non-null    object
dtypes: object(13)
memory usage: 4.5+ KB
dt04 types: None

```

```

-----
-----dt05-----
-----

```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 21 entries, 0 to 20
```

```
Data columns (total 56 columns):
```

#	Column	Non-Null Count	Dtype
0	category_01	21 non-null	object
1	category_02	21 non-null	object
2	2023_12_00	21 non-null	object
3	2023_12_01	21 non-null	object
4	2023_12_02	21 non-null	object
5	2023_12_03	21 non-null	object
6	2023_12_04	21 non-null	object
7	2023_12_05	21 non-null	object
8	2023_12_06	21 non-null	object
9	2023_12_07	21 non-null	object
10	2023_12_08	21 non-null	object
11	2023_12_09	21 non-null	object
12	2023_12_10	21 non-null	object
13	2023_12_11	21 non-null	object
14	2023_12_12	21 non-null	object
15	2023_12_13	21 non-null	object
16	2023_12_14	21 non-null	object
17	2023_12_15	21 non-null	object
18	2023_12_16	21 non-null	object
19	2023_12_17	21 non-null	object
20	2024_01_00	21 non-null	object
21	2024_01_01	21 non-null	object
22	2024_01_02	21 non-null	object
23	2024_01_03	21 non-null	object
24	2024_01_04	21 non-null	object
25	2024_01_05	21 non-null	object
26	2024_01_06	21 non-null	object
27	2024_01_07	21 non-null	object
28	2024_01_08	21 non-null	object
29	2024_01_09	21 non-null	object
30	2024_01_10	21 non-null	object
31	2024_01_11	21 non-null	object
32	2024_01_12	21 non-null	object
33	2024_01_13	21 non-null	object
34	2024_01_14	21 non-null	object


```

35 2024_01_15 21 non-null object
36 2024_01_16 21 non-null object
37 2024_01_17 21 non-null object
38 2024_02_00 21 non-null object
39 2024_02_01 21 non-null object
40 2024_02_02 21 non-null object
41 2024_02_03 21 non-null object
42 2024_02_04 21 non-null object
43 2024_02_05 21 non-null object
44 2024_02_06 21 non-null object
45 2024_02_07 21 non-null object
46 2024_02_08 21 non-null object
47 2024_02_09 21 non-null object
48 2024_02_10 21 non-null object
49 2024_02_11 21 non-null object
50 2024_02_12 21 non-null object
51 2024_02_13 21 non-null object
52 2024_02_14 21 non-null object
53 2024_02_15 21 non-null object
54 2024_02_16 21 non-null object
55 2024_02_17 21 non-null object

```

dtypes: object(56)

memory usage: 9.3+ KB

dt05 types: None

```

-----
-----dt06-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 20 entries, 0 to 19

Data columns (total 5 columns):

#	Column	Non-Null Count	Dtype
0	area	20 non-null	object
1	size	20 non-null	object
2	2023_12	20 non-null	float64
3	2024_01	20 non-null	float64
4	2024_02	20 non-null	float64

dtypes: float64(3), object(2)

memory usage: 932.0+ bytes

dt06 types: None

```

-----
-----dt07-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 21 entries, 0 to 20

Data columns (total 7 columns):

#	Column	Non-Null Count	Dtype
0	area	21 non-null	object
1	2022_00	21 non-null	object
2	2022_01	21 non-null	object
3	2022_02	21 non-null	object
4	2022_03	21 non-null	object
5	2022_04	21 non-null	object
6	2022_05	21 non-null	object

dtypes: object(7)

memory usage: 1.3+ KB

dt07 types: None

-----dt08-----

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 21 entries, 0 to 20

Data columns (total 56 columns):

#	Column	Non-Null Count	Dtype
0	category_01	21 non-null	object
1	category_02	21 non-null	object
2	2023_12_00	21 non-null	object
3	2023_12_01	21 non-null	object
4	2023_12_02	21 non-null	object
5	2023_12_03	21 non-null	object
6	2023_12_04	21 non-null	object
7	2023_12_05	21 non-null	object
8	2023_12_06	21 non-null	object
9	2023_12_07	21 non-null	object
10	2023_12_08	21 non-null	object
11	2023_12_09	21 non-null	object
12	2023_12_10	21 non-null	object
13	2023_12_11	21 non-null	object
14	2023_12_12	21 non-null	object
15	2023_12_13	21 non-null	object
16	2023_12_14	21 non-null	object
17	2023_12_15	21 non-null	object
18	2023_12_16	21 non-null	object
19	2023_12_17	21 non-null	object
20	2024_01_00	21 non-null	object
21	2024_01_01	21 non-null	object
22	2024_01_02	21 non-null	object
23	2024_01_03	21 non-null	object
24	2024_01_04	21 non-null	object
25	2024_01_05	21 non-null	object
26	2024_01_06	21 non-null	object

```

27 2024_01_07 21 non-null object
28 2024_01_08 21 non-null object
29 2024_01_09 21 non-null object
30 2024_01_10 21 non-null object
31 2024_01_11 21 non-null object
32 2024_01_12 21 non-null object
33 2024_01_13 21 non-null object
34 2024_01_14 21 non-null object
35 2024_01_15 21 non-null object
36 2024_01_16 21 non-null object
37 2024_01_17 21 non-null object
38 2024_02_00 21 non-null object
39 2024_02_01 21 non-null object
40 2024_02_02 21 non-null object
41 2024_02_03 21 non-null object
42 2024_02_04 21 non-null object
43 2024_02_05 21 non-null object
44 2024_02_06 21 non-null object
45 2024_02_07 21 non-null object
46 2024_02_08 21 non-null object
47 2024_02_09 21 non-null object
48 2024_02_10 21 non-null object
49 2024_02_11 21 non-null object
50 2024_02_12 21 non-null object
51 2024_02_13 21 non-null object
52 2024_02_14 21 non-null object
53 2024_02_15 21 non-null object
54 2024_02_16 21 non-null object
55 2024_02_17 21 non-null object

```

dtypes: object(56)

memory usage: 9.3+ KB

dt08 types: None

```

-----
-----dt09-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 164 entries, 0 to 163

Data columns (total 10 columns):

#	Column	Non-Null Count	Dtype
0	type	164 non-null	object
1	region_01	164 non-null	object
2	region_02	164 non-null	object
3	region_03	164 non-null	object
4	2023_10	164 non-null	float64
5	2023_11	164 non-null	float64
6	2023_12	164 non-null	float64

```

7   2024_01      164 non-null    float64
8   2024_02      164 non-null    float64
9   2024_03      164 non-null    float64
dtypes: float64(6), object(4)
memory usage: 12.9+ KB
dt09 types: None

```

```

-----
-----dt10-----
-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 164 entries, 0 to 163
Data columns (total 10 columns):
#   Column      Non-Null Count  Dtype
---  -
0   type         164 non-null    object
1   region_01    164 non-null    object
2   region_02    164 non-null    object
3   region_03    164 non-null    object
4   2023_10      164 non-null    float64
5   2023_11      164 non-null    float64
6   2023_12      164 non-null    float64
7   2024_01      164 non-null    float64
8   2024_02      164 non-null    float64
9   2024_03      164 non-null    float64
dtypes: float64(6), object(4)
memory usage: 12.9+ KB
dt10 types: None

```

```

-----
-----dt11-----
-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36 entries, 0 to 35
Data columns (total 53 columns):
#   Column      Non-Null Count  Dtype
---  -
0   variant     36 non-null    object
1   age         36 non-null    object
2   2022        36 non-null    float64
3   2023        36 non-null    float64
4   2024        36 non-null    float64
5   2025        36 non-null    float64
6   2026        36 non-null    float64
7   2027        36 non-null    float64
8   2028        36 non-null    float64
9   2029        36 non-null    float64
10  2030        36 non-null    float64

```

11	2031	36 non-null	float64
12	2032	36 non-null	float64
13	2033	36 non-null	float64
14	2034	36 non-null	float64
15	2035	36 non-null	float64
16	2036	36 non-null	float64
17	2037	36 non-null	float64
18	2038	36 non-null	float64
19	2039	36 non-null	float64
20	2040	36 non-null	float64
21	2041	36 non-null	float64
22	2042	36 non-null	float64
23	2043	36 non-null	float64
24	2044	36 non-null	float64
25	2045	36 non-null	float64
26	2046	36 non-null	float64
27	2047	36 non-null	float64
28	2048	36 non-null	float64
29	2049	36 non-null	float64
30	2050	36 non-null	float64
31	2051	36 non-null	float64
32	2052	36 non-null	float64
33	2053	36 non-null	float64
34	2054	36 non-null	float64
35	2055	36 non-null	float64
36	2056	36 non-null	float64
37	2057	36 non-null	float64
38	2058	36 non-null	float64
39	2059	36 non-null	float64
40	2060	36 non-null	float64
41	2061	36 non-null	float64
42	2062	36 non-null	float64
43	2063	36 non-null	float64
44	2064	36 non-null	float64
45	2065	36 non-null	float64
46	2066	36 non-null	float64
47	2067	36 non-null	float64
48	2068	36 non-null	float64
49	2069	36 non-null	float64
50	2070	36 non-null	float64
51	2071	36 non-null	float64
52	2072	36 non-null	float64

dtypes: float64(51), object(2)

memory usage: 15.0+ KB

dt11 types: None

```
-----
-----
-----dt12-----
```

```

-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45 entries, 0 to 44
Data columns (total 53 columns):
#   Column                Non-Null Count  Dtype
---  -
0   variant                45 non-null    object
1   population_type        45 non-null    object
2   2022                   45 non-null    float64
3   2023                   45 non-null    float64
4   2024                   45 non-null    float64
5   2025                   45 non-null    float64
6   2026                   45 non-null    float64
7   2027                   45 non-null    float64
8   2028                   45 non-null    float64
9   2029                   45 non-null    float64
10  2030                   45 non-null    float64
11  2031                   45 non-null    float64
12  2032                   45 non-null    float64
13  2033                   45 non-null    float64
14  2034                   45 non-null    float64
15  2035                   45 non-null    float64
16  2036                   45 non-null    float64
17  2037                   45 non-null    float64
18  2038                   45 non-null    float64
19  2039                   45 non-null    float64
20  2040                   45 non-null    float64
21  2041                   45 non-null    float64
22  2042                   45 non-null    float64
23  2043                   45 non-null    float64
24  2044                   45 non-null    float64
25  2045                   45 non-null    float64
26  2046                   45 non-null    float64
27  2047                   45 non-null    float64
28  2048                   45 non-null    float64
29  2049                   45 non-null    float64
30  2050                   45 non-null    float64
31  2051                   45 non-null    float64
32  2052                   45 non-null    float64
33  2053                   45 non-null    float64
34  2054                   45 non-null    float64
35  2055                   45 non-null    float64
36  2056                   45 non-null    float64
37  2057                   45 non-null    float64
38  2058                   45 non-null    float64
39  2059                   45 non-null    float64
40  2060                   45 non-null    float64
41  2061                   45 non-null    float64

```

```

42  2062          45 non-null    float64
43  2063          45 non-null    float64
44  2064          45 non-null    float64
45  2065          45 non-null    float64
46  2066          45 non-null    float64
47  2067          45 non-null    float64
48  2068          45 non-null    float64
49  2069          45 non-null    float64
50  2070          45 non-null    float64
51  2071          45 non-null    float64
52  2072          45 non-null    float64

```

dtypes: float64(51), object(2)

memory usage: 18.8+ KB

dt12 types: None

```

-----
-----dt13-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 325 entries, 0 to 324

Data columns (total 21 columns):

#	Column	Non-Null Count	Dtype
0	admin_division	325 non-null	object
1	2022_00	325 non-null	object
2	2022_01	325 non-null	object
3	2022_02	325 non-null	object
4	2022_03	325 non-null	object
5	2022_04	325 non-null	object
6	2022_05	325 non-null	object
7	2022_06	325 non-null	object
8	2022_07	325 non-null	object
9	2022_08	325 non-null	object
10	2022_09	325 non-null	object
11	2022_10	325 non-null	object
12	2022_11	325 non-null	object
13	2022_12	325 non-null	object
14	2022_13	325 non-null	object
15	2022_14	325 non-null	object
16	2022_15	325 non-null	object
17	2022_16	325 non-null	object
18	2022_17	325 non-null	object
19	2022_18	325 non-null	object
20	2022_19	325 non-null	object

dtypes: object(21)

memory usage: 53.4+ KB

dt13 types: None

```

-----

```

```

-----
-----dt14-----
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 21 columns):
#   Column          Non-Null Count  Dtype
---  -
0   admin_division  21 non-null    object
1   2005_00         21 non-null    object
2   2005_01         21 non-null    object
3   2005_02         21 non-null    object
4   2005_03         21 non-null    object
5   2005_04         21 non-null    object
6   2005_05         21 non-null    object
7   2005_06         21 non-null    object
8   2005_07         21 non-null    object
9   2005_08         21 non-null    object
10  2005_09         21 non-null    object
11  2005_10         21 non-null    object
12  2005_11         21 non-null    object
13  2005_12         21 non-null    object
14  2005_13         21 non-null    object
15  2005_14         21 non-null    object
16  2005_15         21 non-null    object
17  2005_16         21 non-null    object
18  2005_17         21 non-null    object
19  2005_18         21 non-null    object
20  2005_19         21 non-null    object
dtypes: object(21)
memory usage: 3.6+ KB
dt14 types: None

```

```

-----
-----dt15-----
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16 entries, 0 to 15
Data columns (total 13 columns):
#   Column          Non-Null Count  Dtype
---  -
0   admin_division  16 non-null    object
1   1970_00         16 non-null    object
2   1970_01         16 non-null    object
3   1970_02         16 non-null    object
4   1970_03         16 non-null    object
5   1970_04         16 non-null    object
6   1970_05         16 non-null    object

```


7	1970_06	16 non-null	object
8	1970_07	16 non-null	object
9	1970_08	16 non-null	object
10	1970_09	16 non-null	object
11	1970_10	16 non-null	object
12	1970_11	16 non-null	object

dtypes: object(13)

memory usage: 1.8+ KB

dt15 types: None

 -----dt16-----

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 20 entries, 0 to 19

Data columns (total 18 columns):

#	Column	Non-Null Count	Dtype
0	admin_division	20 non-null	object
1	1995_00	20 non-null	object
2	1995_01	20 non-null	object
3	1995_02	20 non-null	object
4	1995_03	20 non-null	object
5	1995_04	20 non-null	object
6	1995_05	20 non-null	object
7	1995_06	20 non-null	object
8	1995_07	20 non-null	object
9	1995_08	20 non-null	object
10	1995_09	20 non-null	object
11	1995_10	20 non-null	object
12	1995_11	20 non-null	object
13	1995_12	20 non-null	object
14	1995_13	20 non-null	object
15	1995_14	20 non-null	object
16	1995_15	20 non-null	object
17	1995_16	20 non-null	object

dtypes: object(18)

memory usage: 2.9+ KB

dt16 types: None

 -----dt17-----

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 20 entries, 0 to 19

Data columns (total 18 columns):

#	Column	Non-Null Count	Dtype
---	--------	----------------	-------

0	admin_division	20 non-null	object
1	1990_00	20 non-null	object
2	1990_01	20 non-null	object
3	1990_02	20 non-null	object
4	1990_03	20 non-null	object
5	1990_04	20 non-null	object
6	1990_05	20 non-null	object
7	1990_06	20 non-null	object
8	1990_07	20 non-null	object
9	1990_08	20 non-null	object
10	1990_09	20 non-null	object
11	1990_10	20 non-null	object
12	1990_11	20 non-null	object
13	1990_12	20 non-null	object
14	1990_13	20 non-null	object
15	1990_14	20 non-null	object
16	1990_15	20 non-null	object
17	1990_16	20 non-null	object

dtypes: object(18)
memory usage: 2.9+ KB
dt17 types: None

-----dt18-----

```
<class 'pandas.core.frame.DataFrame'>
```

RangeIndex: 18 entries, 0 to 17

Data columns (total 18 columns):

#	Column	Non-Null Count	Dtype
0	admin_division	18 non-null	object
1	1985_00	18 non-null	object
2	1985_01	18 non-null	object
3	1985_02	18 non-null	object
4	1985_03	18 non-null	object
5	1985_04	18 non-null	object
6	1985_05	18 non-null	object
7	1985_06	18 non-null	object
8	1985_07	18 non-null	object
9	1985_08	18 non-null	object
10	1985_09	18 non-null	object
11	1985_10	18 non-null	object
12	1985_11	18 non-null	object
13	1985_12	18 non-null	object
14	1985_13	18 non-null	object
15	1985_14	18 non-null	object
16	1985_15	18 non-null	object
17	1985_16	18 non-null	object

```
dtypes: object(18)
memory usage: 2.7+ KB
dt18 types: None
```

```
-----
-----dt19-----
-----
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 16 entries, 0 to 15
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	admin_division	16 non-null	object
1	1980_00	16 non-null	object
2	1980_01	16 non-null	object
3	1980_02	16 non-null	object
4	1980_03	16 non-null	object
5	1980_04	16 non-null	object
6	1980_05	16 non-null	object
7	1980_06	16 non-null	object
8	1980_07	16 non-null	object
9	1980_08	16 non-null	object
10	1980_09	16 non-null	object
11	1980_10	16 non-null	object
12	1980_11	16 non-null	object
13	1980_12	16 non-null	object
14	1980_13	16 non-null	object
15	1980_14	16 non-null	object
16	1980_15	16 non-null	object
17	1980_16	16 non-null	object

```
dtypes: object(18)
memory usage: 2.4+ KB
dt19 types: None
```

```
-----
-----dt20-----
-----
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 16 entries, 0 to 15
```

```
Data columns (total 13 columns):
```

#	Column	Non-Null Count	Dtype
0	admin_division	16 non-null	object
1	1975_00	16 non-null	object
2	1975_01	16 non-null	object
3	1975_02	16 non-null	object
4	1975_03	16 non-null	object
5	1975_04	16 non-null	object

```

6   1975_05          16 non-null    object
7   1975_06          16 non-null    object
8   1975_07          16 non-null    object
9   1975_08          16 non-null    object
10  1975_09          16 non-null    object
11  1975_10          16 non-null    object
12  1975_11          16 non-null    object
dtypes: object(13)
memory usage: 1.8+ KB
dt20 types: None

```

```

-----
-----dt21-----
-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 21 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   admin_division  21 non-null    object
1   2000_00         21 non-null    object
2   2000_01         21 non-null    object
3   2000_02         21 non-null    object
4   2000_03         21 non-null    object
5   2000_04         21 non-null    object
6   2000_05         21 non-null    object
7   2000_06         21 non-null    object
8   2000_07         21 non-null    object
9   2000_08         21 non-null    object
10  2000_09         21 non-null    object
11  2000_10         21 non-null    object
12  2000_11         21 non-null    object
13  2000_12         21 non-null    object
14  2000_13         21 non-null    object
15  2000_14         21 non-null    object
16  2000_15         21 non-null    object
17  2000_16         21 non-null    object
18  2000_17         21 non-null    object
19  2000_18         21 non-null    object
20  2000_19         21 non-null    object

```

```

dtypes: object(21)
memory usage: 3.6+ KB
dt21 types: None

```

```

-----
-----dt22-----
-----

```

```

<class 'pandas.core.frame.DataFrame'>

```

RangeIndex: 69 entries, 0 to 68

Data columns (total 54 columns):

#	Column	Non-Null Count	Dtype
0	variant	69 non-null	object
1	gender	69 non-null	object
2	age	69 non-null	object
3	2022	69 non-null	int64
4	2023	69 non-null	int64
5	2024	69 non-null	int64
6	2025	69 non-null	int64
7	2026	69 non-null	int64
8	2027	69 non-null	int64
9	2028	69 non-null	int64
10	2029	69 non-null	int64
11	2030	69 non-null	int64
12	2031	69 non-null	int64
13	2032	69 non-null	int64
14	2033	69 non-null	int64
15	2034	69 non-null	int64
16	2035	69 non-null	int64
17	2036	69 non-null	int64
18	2037	69 non-null	int64
19	2038	69 non-null	int64
20	2039	69 non-null	int64
21	2040	69 non-null	int64
22	2041	69 non-null	int64
23	2042	69 non-null	int64
24	2043	69 non-null	int64
25	2044	69 non-null	int64
26	2045	69 non-null	int64
27	2046	69 non-null	int64
28	2047	69 non-null	int64
29	2048	69 non-null	int64
30	2049	69 non-null	int64
31	2050	69 non-null	int64
32	2051	69 non-null	int64
33	2052	69 non-null	int64
34	2053	69 non-null	int64
35	2054	69 non-null	int64
36	2055	69 non-null	int64
37	2056	69 non-null	int64
38	2057	69 non-null	int64
39	2058	69 non-null	int64
40	2059	69 non-null	int64
41	2060	69 non-null	int64
42	2061	69 non-null	int64
43	2062	69 non-null	int64

```

44 2063      69 non-null    int64
45 2064      69 non-null    int64
46 2065      69 non-null    int64
47 2066      69 non-null    int64
48 2067      69 non-null    int64
49 2068      69 non-null    int64
50 2069      69 non-null    int64
51 2070      69 non-null    int64
52 2071      69 non-null    int64
53 2072      69 non-null    int64

```

dtypes: int64(51), object(3)

memory usage: 29.2+ KB

dt22 types: None

```

-----
-----dt23-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 7 entries, 0 to 6

Data columns (total 53 columns):

#	Column	Non-Null Count	Dtype
0	variant	7 non-null	object
1	cause	7 non-null	object
2	2022	7 non-null	float64
3	2023	7 non-null	float64
4	2024	7 non-null	float64
5	2025	7 non-null	float64
6	2026	7 non-null	float64
7	2027	7 non-null	float64
8	2028	7 non-null	float64
9	2029	7 non-null	float64
10	2030	7 non-null	float64
11	2031	7 non-null	float64
12	2032	7 non-null	float64
13	2033	7 non-null	float64
14	2034	7 non-null	float64
15	2035	7 non-null	float64
16	2036	7 non-null	float64
17	2037	7 non-null	float64
18	2038	7 non-null	float64
19	2039	7 non-null	float64
20	2040	7 non-null	float64
21	2041	7 non-null	float64
22	2042	7 non-null	float64
23	2043	7 non-null	float64
24	2044	7 non-null	float64
25	2045	7 non-null	float64

```

26 2046      7 non-null    float64
27 2047      7 non-null    float64
28 2048      7 non-null    float64
29 2049      7 non-null    float64
30 2050      7 non-null    float64
31 2051      7 non-null    float64
32 2052      7 non-null    float64
33 2053      7 non-null    float64
34 2054      7 non-null    float64
35 2055      7 non-null    float64
36 2056      7 non-null    float64
37 2057      7 non-null    float64
38 2058      7 non-null    float64
39 2059      7 non-null    float64
40 2060      7 non-null    float64
41 2061      7 non-null    float64
42 2062      7 non-null    float64
43 2063      7 non-null    float64
44 2064      7 non-null    float64
45 2065      7 non-null    float64
46 2066      7 non-null    float64
47 2067      7 non-null    float64
48 2068      7 non-null    float64
49 2069      7 non-null    float64
50 2070      7 non-null    float64
51 2071      7 non-null    float64
52 2072      7 non-null    float64
dtypes: float64(51), object(2)
memory usage: 3.0+ KB
dt23 types: None

-----
-----dt24-----
-----

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 476 entries, 0 to 475
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   By scenarios 476 non-null    object
1   structure    476 non-null    object
2   2070         476 non-null    float64
3   2071         476 non-null    float64
4   2072         476 non-null    float64
dtypes: float64(3), object(2)
memory usage: 18.7+ KB
dt24 types: None
-----

```

```
-----
-----dt25-----
-----
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 21 entries, 0 to 20
```

```
Data columns (total 53 columns):
```

#	Column	Non-Null Count	Dtype
0	variant	21 non-null	object
1	dependency_ratio	21 non-null	object
2	2022	21 non-null	float64
3	2023	21 non-null	float64
4	2024	21 non-null	float64
5	2025	21 non-null	float64
6	2026	21 non-null	float64
7	2027	21 non-null	float64
8	2028	21 non-null	float64
9	2029	21 non-null	float64
10	2030	21 non-null	float64
11	2031	21 non-null	float64
12	2032	21 non-null	float64
13	2033	21 non-null	float64
14	2034	21 non-null	float64
15	2035	21 non-null	float64
16	2036	21 non-null	float64
17	2037	21 non-null	float64
18	2038	21 non-null	float64
19	2039	21 non-null	float64
20	2040	21 non-null	float64
21	2041	21 non-null	float64
22	2042	21 non-null	float64
23	2043	21 non-null	float64
24	2044	21 non-null	float64
25	2045	21 non-null	float64
26	2046	21 non-null	float64
27	2047	21 non-null	float64
28	2048	21 non-null	float64
29	2049	21 non-null	float64
30	2050	21 non-null	float64
31	2051	21 non-null	float64
32	2052	21 non-null	float64
33	2053	21 non-null	float64
34	2054	21 non-null	float64
35	2055	21 non-null	float64
36	2056	21 non-null	float64
37	2057	21 non-null	float64
38	2058	21 non-null	float64
39	2059	21 non-null	float64


```

40 2060          21 non-null    float64
41 2061          21 non-null    float64
42 2062          21 non-null    float64
43 2063          21 non-null    float64
44 2064          21 non-null    float64
45 2065          21 non-null    float64
46 2066          21 non-null    float64
47 2067          21 non-null    float64
48 2068          21 non-null    float64
49 2069          21 non-null    float64
50 2070          21 non-null    float64
51 2071          21 non-null    float64
52 2072          21 non-null    float64

```

dtypes: float64(51), object(2)

memory usage: 8.8+ KB

dt25 types: None

```

-----
-----dt26-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 14 entries, 0 to 13

Data columns (total 4 columns):

#	Column	Non-Null Count	Dtype
0	loans_personal_01	14 non-null	object
1	loans_personal_02	14 non-null	object
2	2022_01_02	14 non-null	int64
3	2022_02_02	14 non-null	int64

dtypes: int64(2), object(2)

memory usage: 580.0+ bytes

dt26 types: None

```

-----
-----dt27-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 17 entries, 0 to 16

Data columns (total 8 columns):

#	Column	Non-Null Count	Dtype
0	classification_si_do	17 non-null	object
1	classification_income	17 non-null	object
2	2019_00	17 non-null	object
3	2019_01	17 non-null	object
4	2019_02	17 non-null	object
5	2019_03	17 non-null	object
6	2019_04	17 non-null	object

```

7    2019_05                17 non-null    object
dtypes: object(8)
memory usage: 1.2+ KB
dt27 types: None

```

```

-----dt28-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23 entries, 0 to 22
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   classification  23 non-null    object
1   2022_00         23 non-null    object
2   2022_01         23 non-null    object
3   2022_02         23 non-null    object
4   2022_03         23 non-null    object
5   2022_04         23 non-null    object
dtypes: object(6)
memory usage: 1.2+ KB
dt28 types: None

```

```

-----dt29-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9 entries, 0 to 8
Data columns (total 6 columns):
#   Column  Non-Null Count  Dtype
---  -
0   item    9 non-null      object
1   2018    9 non-null      float64
2   2019    9 non-null      float64
3   2020    9 non-null      float64
4   2021    9 non-null      float64
5   2022    9 non-null      float64
dtypes: float64(5), object(1)
memory usage: 564.0+ bytes
dt29 types: None

```

```

-----dt30-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 92857 entries, 0 to 92856
Data columns (total 15 columns):
#   Column          Non-Null Count  Dtype

```

```

---  -----  -----  -----
0  index          92857 non-null  int64
1  id             92857 non-null  int64
2  year           92857 non-null  int64
3  wave           92857 non-null  int64
4  region         92857 non-null  int64
5  income         92857 non-null  float64
6  fam_mem_num    92857 non-null  int64
7  gender         92857 non-null  int64
8  birth_year     92857 non-null  int64
9  education_level 92857 non-null  int64
10 marriage       92857 non-null  int64
11 religion       92857 non-null  int64
12 occupation     92857 non-null  object
13 company_size   92857 non-null  object
14 unemployment_reason 92857 non-null  object

```

dtypes: float64(1), int64(11), object(3)

memory usage: 10.6+ MB

dt30 types: None

```

-----
-----dt31-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 21 entries, 0 to 20

Data columns (total 53 columns):

#	Column	Non-Null Count	Dtype
0	By variant	21 non-null	object
1	By the structure of population and dependency ratio	21 non-null	object
2	2022	21 non-null	
	float64		
3	2023	21 non-null	
	float64		
4	2024	21 non-null	
	float64		
5	2025	21 non-null	
	float64		
6	2026	21 non-null	
	float64		
7	2027	21 non-null	
	float64		
8	2028	21 non-null	
	float64		
9	2029	21 non-null	
	float64		
10	2030	21 non-null	
	float64		

11	2031	21 non-null
float64		
12	2032	21 non-null
float64		
13	2033	21 non-null
float64		
14	2034	21 non-null
float64		
15	2035	21 non-null
float64		
16	2036	21 non-null
float64		
17	2037	21 non-null
float64		
18	2038	21 non-null
float64		
19	2039	21 non-null
float64		
20	2040	21 non-null
float64		
21	2041	21 non-null
float64		
22	2042	21 non-null
float64		
23	2043	21 non-null
float64		
24	2044	21 non-null
float64		
25	2045	21 non-null
float64		
26	2046	21 non-null
float64		
27	2047	21 non-null
float64		
28	2048	21 non-null
float64		
29	2049	21 non-null
float64		
30	2050	21 non-null
float64		
31	2051	21 non-null
float64		
32	2052	21 non-null
float64		
33	2053	21 non-null
float64		
34	2054	21 non-null
float64		

```

    35  2055                                21 non-null
float64
    36  2056                                21 non-null
float64
    37  2057                                21 non-null
float64
    38  2058                                21 non-null
float64
    39  2059                                21 non-null
float64
    40  2060                                21 non-null
float64
    41  2061                                21 non-null
float64
    42  2062                                21 non-null
float64
    43  2063                                21 non-null
float64
    44  2064                                21 non-null
float64
    45  2065                                21 non-null
float64
    46  2066                                21 non-null
float64
    47  2067                                21 non-null
float64
    48  2068                                21 non-null
float64
    49  2069                                21 non-null
float64
    50  2070                                21 non-null
float64
    51  2071                                21 non-null
float64
    52  2072                                21 non-null
float64
dtypes: float64(51), object(2)
memory usage: 8.8+ KB
dt31 types: None

```

```

-----
-----dt32-----
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4860 entries, 0 to 4859
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -

```

```

0   Date                4860 non-null  object
1   Region              4860 non-null  object
2   Birth               4716 non-null  float64
3   Birth_rate          4709 non-null  float64
4   Death               4716 non-null  float64
5   Death_rate          4709 non-null  float64
6   Divorce             4716 non-null  float64
7   Divorce_rate        4709 non-null  float64
8   Marriage             4716 non-null  float64
9   Marriage_rate        4709 non-null  float64
10  Natural_growth      4716 non-null  float64
11  Natural_growth_rate  4709 non-null  float64

```

dtypes: float64(10), object(2)

memory usage: 455.8+ KB

dt32 types: None

```

-----
-----dt33-----
-----

```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 325 entries, 0 to 324

Data columns (total 21 columns):

#	Column	Non-Null Count	Dtype
0	By administrative divisions(eup, myeon, dong)	325 non-null	object
1	2022	325 non-null	object
2	2022.1	325 non-null	object
3	2022.2	325 non-null	object
4	2022.3	325 non-null	object
5	2022.4	325 non-null	object
6	2022.5	325 non-null	object
7	2022.6	325 non-null	object
8	2022.7	325 non-null	object
9	2022.8	325 non-null	object
10	2022.9	325 non-null	object
11	2022.10	325 non-null	object
12	2022.11	325 non-null	object
13	2022.12	325 non-null	object
14	2022.13	325 non-null	object
15	2022.14	325 non-null	object
16	2022.15	325 non-null	object
17	2022.16	325 non-null	object
18	2022.17	325 non-null	object
19	2022.18	325 non-null	object
20	2022.19	325 non-null	object

dtypes: object(21)

memory usage: 53.4+ KB

dt33 types: None

```

-----
-----dt34-----
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 21 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   By administrative divisions(eup, myeon, dong) 21 non-null    object
1   2010                                           21 non-null    object
2   2010.1                                         21 non-null    object
3   2010.2                                         21 non-null    object
4   2010.3                                         21 non-null    object
5   2010.4                                         21 non-null    object
6   2010.5                                         21 non-null    object
7   2010.6                                         21 non-null    object
8   2010.7                                         21 non-null    object
9   2010.8                                         21 non-null    object
10  2010.9                                         21 non-null    object
11  2010.10                                        21 non-null    object
12  2010.11                                        21 non-null    object
13  2010.12                                        21 non-null    object
14  2010.13                                        21 non-null    object
15  2010.14                                        21 non-null    object
16  2010.15                                        21 non-null    object
17  2010.16                                        21 non-null    object
18  2010.17                                        21 non-null    object
19  2010.18                                        21 non-null    object
20  2010.19                                        21 non-null    object
dtypes: object(21)
memory usage: 3.6+ KB
dt34 types: None

```

```

-----
-----dt35-----
-----
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21 entries, 0 to 20
Data columns (total 21 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   By administrative divisions 21 non-null    object
1   2005                         21 non-null    object
2   2005.1                       21 non-null    object
3   2005.2                       21 non-null    object
4   2005.3                       21 non-null    object
5   2005.4                       21 non-null    object

```

6	2005.5	21 non-null	object
7	2005.6	21 non-null	object
8	2005.7	21 non-null	object
9	2005.8	21 non-null	object
10	2005.9	21 non-null	object
11	2005.10	21 non-null	object
12	2005.11	21 non-null	object
13	2005.12	21 non-null	object
14	2005.13	21 non-null	object
15	2005.14	21 non-null	object
16	2005.15	21 non-null	object
17	2005.16	21 non-null	object
18	2005.17	21 non-null	object
19	2005.18	21 non-null	object
20	2005.19	21 non-null	object

dtypes: object(21)

memory usage: 3.6+ KB

dt35 types: None

```
-----
-----dt36-----
-----
```

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 21 entries, 0 to 20

Data columns (total 21 columns):

#	Column	Non-Null Count	Dtype
0	By administrative divisions	21 non-null	object
1	2000	21 non-null	object
2	2000.1	21 non-null	object
3	2000.2	21 non-null	object
4	2000.3	21 non-null	object
5	2000.4	21 non-null	object
6	2000.5	21 non-null	object
7	2000.6	21 non-null	object
8	2000.7	21 non-null	object
9	2000.8	21 non-null	object
10	2000.9	21 non-null	object
11	2000.10	21 non-null	object
12	2000.11	21 non-null	object
13	2000.12	21 non-null	object
14	2000.13	21 non-null	object
15	2000.14	21 non-null	object
16	2000.15	21 non-null	object
17	2000.16	21 non-null	object
18	2000.17	21 non-null	object
19	2000.18	21 non-null	object
20	2000.19	21 non-null	object


```
dtypes: object(21)
memory usage: 3.6+ KB
dt36 types: None
```

```
-----
-----dt37-----
-----
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 20 entries, 0 to 19
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	By administrative divisions	20 non-null	object
1	1995	20 non-null	object
2	1995.1	20 non-null	object
3	1995.2	20 non-null	object
4	1995.3	20 non-null	object
5	1995.4	20 non-null	object
6	1995.5	20 non-null	object
7	1995.6	20 non-null	object
8	1995.7	20 non-null	object
9	1995.8	20 non-null	object
10	1995.9	20 non-null	object
11	1995.10	20 non-null	object
12	1995.11	20 non-null	object
13	1995.12	20 non-null	object
14	1995.13	20 non-null	object
15	1995.14	20 non-null	object
16	1995.15	20 non-null	object
17	1995.16	20 non-null	object

```
dtypes: object(18)
```

```
memory usage: 2.9+ KB
```

```
dt37 types: None
```

```
-----
-----dt38-----
-----
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 20 entries, 0 to 19
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	By administrative divisions	20 non-null	object
1	1990	20 non-null	object
2	1990.1	20 non-null	object
3	1990.2	20 non-null	object
4	1990.3	20 non-null	object
5	1990.4	20 non-null	object

6	1990.5	20 non-null	object
7	1990.6	20 non-null	object
8	1990.7	20 non-null	object
9	1990.8	20 non-null	object
10	1990.9	20 non-null	object
11	1990.10	20 non-null	object
12	1990.11	20 non-null	object
13	1990.12	20 non-null	object
14	1990.13	20 non-null	object
15	1990.14	20 non-null	object
16	1990.15	20 non-null	object
17	1990.16	20 non-null	object

dtypes: object(18)

memory usage: 2.9+ KB

dt38 types: None

 -----dt39-----

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 18 entries, 0 to 17

Data columns (total 18 columns):

#	Column	Non-Null Count	Dtype
0	By administrative divisions	18 non-null	object
1	1985	18 non-null	object
2	1985.1	18 non-null	object
3	1985.2	18 non-null	object
4	1985.3	18 non-null	object
5	1985.4	18 non-null	object
6	1985.5	18 non-null	object
7	1985.6	18 non-null	object
8	1985.7	18 non-null	object
9	1985.8	18 non-null	object
10	1985.9	18 non-null	object
11	1985.10	18 non-null	object
12	1985.11	18 non-null	object
13	1985.12	18 non-null	object
14	1985.13	18 non-null	object
15	1985.14	18 non-null	object
16	1985.15	18 non-null	object
17	1985.16	18 non-null	object

dtypes: object(18)

memory usage: 2.7+ KB

dt39 types: None

 -----dt40-----

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 16 entries, 0 to 15

Data columns (total 18 columns):

#	Column	Non-Null Count	Dtype
0	By administrative divisions	16 non-null	object
1	1980	16 non-null	object
2	1980.1	16 non-null	object
3	1980.2	16 non-null	object
4	1980.3	16 non-null	object
5	1980.4	16 non-null	object
6	1980.5	16 non-null	object
7	1980.6	16 non-null	object
8	1980.7	16 non-null	object
9	1980.8	16 non-null	object
10	1980.9	16 non-null	object
11	1980.10	16 non-null	object
12	1980.11	16 non-null	object
13	1980.12	16 non-null	object
14	1980.13	16 non-null	object
15	1980.14	16 non-null	object
16	1980.15	16 non-null	object
17	1980.16	16 non-null	object

dtypes: object(18)

memory usage: 2.4+ KB

dt40 types: None

-----dt41-----

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 16 entries, 0 to 15

Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	By administrative divisions	16 non-null	object
1	1975	16 non-null	object
2	1975.1	16 non-null	object
3	1975.2	16 non-null	object
4	1975.3	16 non-null	object
5	1975.4	16 non-null	object
6	1975.5	16 non-null	object
7	1975.6	16 non-null	object
8	1975.7	16 non-null	object
9	1975.8	16 non-null	object
10	1975.9	16 non-null	object
11	1975.10	16 non-null	object

```

12  1975.11                                16 non-null    object
dtypes: object(13)
memory usage: 1.8+ KB
dt41 types: None

```

```

-----
-----dt42-----
-----

```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 16 entries, 0 to 15
```

```
Data columns (total 13 columns):
```

#	Column	Non-Null Count	Dtype
0	By administrative divisions	16 non-null	object
1	1970	16 non-null	object
2	1970.1	16 non-null	object
3	1970.2	16 non-null	object
4	1970.3	16 non-null	object
5	1970.4	16 non-null	object
6	1970.5	16 non-null	object
7	1970.6	16 non-null	object
8	1970.7	16 non-null	object
9	1970.8	16 non-null	object
10	1970.9	16 non-null	object
11	1970.10	16 non-null	object
12	1970.11	16 non-null	object

```
dtypes: object(13)
```

```
memory usage: 1.8+ KB
```

```
dt42 types: None
```

```

-----
-----dt43-----
-----

```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 4544 entries, 0 to 4543
```

```
Data columns (total 11 columns):
```

#	Column	Non-Null Count	Dtype
0	index	4544 non-null	int64
1	name	4544 non-null	object
2	gugun	4544 non-null	object
3	dong	4544 non-null	object
4	buildDate	4462 non-null	float64
5	min_sales	4333 non-null	float64
6	max_sales	4333 non-null	float64
7	avg_sales	4333 non-null	float64
8	area	4544 non-null	int64
9	floor	4544 non-null	int64

```

10 pricePerArea 4333 non-null float64
dtypes: float64(5), int64(3), object(3)
memory usage: 390.6+ KB
dt43 types: None

```

```

-----
-----dt44-----
-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4860 entries, 0 to 4859
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Date                  4860 non-null   object
1   Region                4860 non-null   object
2   Birth                 4716 non-null   float64
3   Birth_rate            4709 non-null   float64
4   Death                 4716 non-null   float64
5   Death_rate            4709 non-null   float64
6   Divorce               4716 non-null   float64
7   Divorce_rate          4709 non-null   float64
8   Marriage               4716 non-null   float64
9   Marriage_rate          4709 non-null   float64
10  Natural_growth         4716 non-null   float64
11  Natural_growth_rate    4709 non-null   float64
dtypes: float64(10), object(2)
memory usage: 455.8+ KB
dt44 types: None

```

```

-----
-----dt45-----
-----

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57 entries, 0 to 56
Columns: 326 entries, region to 2012_12
dtypes: object(326)
memory usage: 145.3+ KB
dt45 types: None

```

```

[86]: # 04.02.01-04
      # return dimensions
      # dt01-dt44

print('dt01 dimensions:',dt01_real_estate_pr_01.shape)
print('dt02 dimensions:',dt02_real_estate_se_01.shape)
print('dt03 dimensions:',dt03_stats_vital_kr_01.shape)
print('dt04 dimensions:',dt04_housing_census_01.shape)

```

```

print('dt05 dimensions:', dt05_housing_constr_01.shape)
print('dt06 dimensions:', dt06_household_empty_02.shape)
print('dt07 dimensions:', dt07_housing_loss___01.shape)
print('dt08 dimensions:', dt08_housing_use_sv_01.shape)
print('dt09 dimensions:', dt09_jeonse_sales___01.shape)
print('dt10 dimensions:', dt10_jeonse_market__02.shape)
print('dt11 dimensions:', dt11_birthrate_age__01.shape)
print('dt12 dimensions:', dt12_birthrate_age__02.shape)
print('dt13 dimensions:', dt13_pop_houshold___01.shape)
print('dt14 dimensions:', dt14_pop_houshold___02.shape)
print('dt15 dimensions:', dt15_pop_houshold___03.shape)
print('dt16 dimensions:', dt16_pop_houshold___04.shape)
print('dt17 dimensions:', dt17_pop_houshold___05.shape)
print('dt18 dimensions:', dt18_pop_houshold___06.shape)
print('dt19 dimensions:', dt19_pop_houshold___07.shape)
print('dt20 dimensions:', dt20_pop_houshold___08.shape)
print('dt21 dimensions:', dt21_pop_houshold___09.shape)
print('dt22 dimensions:', dt22_pop_estimate___01.shape)
print('dt23 dimensions:', dt23_pop_future_pro_01.shape)
print('dt24 dimensions:', dt24_pop_future_pro_02.shape)
print('dt25 dimensions:', dt25_pop_future_pro_02.shape)
print('dt26 dimensions:', dt26_salaries_entir_01.shape)
print('dt27 dimensions:', dt27_income_job_typ_01.shape)
print('dt28 dimensions:', dt28_employment_eff_01.shape)
print('dt29 dimensions:', dt29_income_median__01.shape)
print('dt30 dimensions:', dt30_ROK_demo_______01.shape)
print('dt31 dimensions:', dt31_ROK_demo_______02.shape)
print('dt32 dimensions:', dt32_ROK_population_01.shape)
print('dt33 dimensions:', dt33_ROK_population_01.shape)
print('dt34 dimensions:', dt34_ROK_population_01.shape)
print('dt35 dimensions:', dt35_ROK_population_01.shape)
print('dt36 dimensions:', dt36_ROK_population_01.shape)
print('dt37 dimensions:', dt37_ROK_population_01.shape)
print('dt38 dimensions:', dt38_ROK_population_01.shape)
print('dt39 dimensions:', dt39_ROK_population_01.shape)
print('dt40 dimensions:', dt40_ROK_population_01.shape)
print('dt41 dimensions:', dt41_ROK_population_01.shape)
print('dt42 dimensions:', dt42_ROK_population_01.shape)
print('dt43 dimensions:', dt43_Seoul_apr______01.shape)
print('dt44 dimensions:', dt44_ROK_population_01.shape)
print('dt45 dimensions:', dt45_RE_price_______01.shape)

```

```

dt01 dimensions: (27, 8)
dt02 dimensions: (4021, 11)
dt03 dimensions: (16, 4)
dt04 dimensions: (43, 13)
dt05 dimensions: (21, 56)
dt06 dimensions: (20, 5)

```

```

dt07 dimensions: (21, 7)
dt08 dimensions: (21, 56)
dt09 dimensions: (164, 10)
dt10 dimensions: (164, 10)
dt11 dimensions: (36, 53)
dt12 dimensions: (45, 53)
dt13 dimensions: (325, 21)
dt14 dimensions: (21, 21)
dt15 dimensions: (16, 13)
dt16 dimensions: (20, 18)
dt17 dimensions: (20, 18)
dt18 dimensions: (18, 18)
dt19 dimensions: (16, 18)
dt20 dimensions: (16, 13)
dt21 dimensions: (21, 21)
dt22 dimensions: (69, 54)
dt23 dimensions: (7, 53)
dt24 dimensions: (476, 5)
dt25 dimensions: (21, 53)
dt26 dimensions: (14, 4)
dt27 dimensions: (17, 8)
dt28 dimensions: (23, 6)
dt29 dimensions: (9, 6)
dt30 dimensions: (92857, 15)
dt31 dimensions: (21, 53)
dt32 dimensions: (4860, 12)
dt33 dimensions: (325, 21)
dt34 dimensions: (21, 21)
dt35 dimensions: (21, 21)
dt36 dimensions: (21, 21)
dt37 dimensions: (20, 18)
dt38 dimensions: (20, 18)
dt39 dimensions: (18, 18)
dt40 dimensions: (16, 18)
dt41 dimensions: (16, 13)
dt42 dimensions: (16, 13)
dt43 dimensions: (4544, 11)
dt44 dimensions: (4860, 12)
dt45 dimensions: (57, 326)

```

```

[87]: # 04.02.01-05
      # return first and last 6 rows
      # dt01-dt44

      print('-----dt01-----')
      print('dt01 first 6 rows:', dt01_real_estate_pr_01.head(6))

```

```

print('dt01 last 6 rows:',dt01_real_estate_pr_01.tail(6))
print('-----dt02-----')
print('dt02 frst 6 rows:',dt02_real_estate_se_01.head(6))
print('dt02 last 6 rows:',dt02_real_estate_se_01.tail(6))
print('-----dt03-----')
print('dt03 frst 6 rows:',dt03_stats_vital_kr_01.head(6))
print('dt03 last 6 rows:',dt03_stats_vital_kr_01.tail(6))
print('-----dt04-----')
print('dt04 frst 6 rows:',dt04_housing_census_01.head(6))
print('dt04 last 6 rows:',dt04_housing_census_01.tail(6))
print('-----dt05-----')
print('dt05 frst 6 rows:',dt05_housing_constr_01.head(6))
print('dt05 last 6 rows:',dt05_housing_constr_01.tail(6))
print('-----dt06-----')
print('dt06 frst 6 rows:',dt06_household_empty_02.head(6))
print('dt06 last 6 rows:',dt06_household_empty_02.tail(6))
print('-----dt07-----')
print('dt07 frst 6 rows:',dt07_housing_loss__01.head(6))
print('dt07 last 6 rows:',dt07_housing_loss__01.tail(6))
print('-----dt08-----')
print('dt08 frst 6 rows:',dt08_housing_use_sv_01.head(6))
print('dt08 last 6 rows:',dt08_housing_use_sv_01.tail(6))
print('-----dt09-----')
print('dt09 frst 6 rows:',dt09_jeonse_sales__01.head(6))
print('dt09 last 6 rows:',dt09_jeonse_sales__01.tail(6))
print('-----dt10-----')
print('dt10 frst 6 rows:',dt10_jeonse_market__02.head(6))
print('dt10 last 6 rows:',dt10_jeonse_market__02.tail(6))
print('-----dt11-----')
print('dt11 frst 6 rows:',dt11_birthrate_age__01.head(6))
print('dt11 last 6 rows:',dt11_birthrate_age__01.tail(6))
print('-----dt12-----')
print('dt12 frst 6 rows:',dt12_birthrate_age__02.head(6))
print('dt12 last 6 rows:',dt12_birthrate_age__02.tail(6))
print('-----dt13-----')

```



```

print('dt13 frst 6 rows:',dt13_pop_houshold__01.head(6))
print('dt13 last 6 rows:',dt13_pop_houshold__01.tail(6))
print('-----dt14-----')
print('dt14 frst 6 rows:',dt14_pop_houshold__02.head(6))
print('dt14 last 6 rows:',dt14_pop_houshold__02.tail(6))
print('-----dt15-----')
print('dt15 frst 6 rows:',dt15_pop_houshold__03.head(6))
print('dt15 last 6 rows:',dt15_pop_houshold__03.tail(6))
print('-----dt16-----')
print('dt16 frst 6 rows:',dt16_pop_houshold__04.head(6))
print('dt16 last 6 rows:',dt16_pop_houshold__04.tail(6))
print('-----dt17-----')
print('dt17 frst 6 rows:',dt17_pop_houshold__05.head(6))
print('dt17 last 6 rows:',dt17_pop_houshold__05.tail(6))
print('-----dt18-----')
print('dt18 frst 6 rows:',dt18_pop_houshold__06.head(6))
print('dt18 last 6 rows:',dt18_pop_houshold__06.tail(6))
print('-----dt19-----')
print('dt19 frst 6 rows:',dt19_pop_houshold__07.head(6))
print('dt19 last 6 rows:',dt19_pop_houshold__07.tail(6))
print('-----dt20-----')
print('dt20 frst 6 rows:',dt20_pop_houshold__08.head(6))
print('dt20 last 6 rows:',dt20_pop_houshold__08.tail(6))
print('-----dt21-----')
print('dt21 frst 6 rows:',dt21_pop_houshold__09.head(6))
print('dt21 last 6 rows:',dt21_pop_houshold__09.tail(6))
print('-----dt22-----')
print('dt22 frst 6 rows:',dt22_pop_estimate__01.head(6))
print('dt22 last 6 rows:',dt22_pop_estimate__01.tail(6))
print('-----dt23-----')
print('dt23 frst 6 rows:',dt23_pop_future_pro_01.head(6))
print('dt23 last 6 rows:',dt23_pop_future_pro_01.tail(6))
print('-----dt24-----')
print('dt24 frst 6 rows:',dt24_pop_future_pro_02.head(6))
print('dt24 last 6 rows:',dt24_pop_future_pro_02.tail(6))
print('-----')

```

```

print('-----dt25-----')
print('dt25 frst 6 rows:',dt25_pop_future_pro_02.head(6))
print('dt25 last 6 rows:',dt25_pop_future_pro_02.tail(6))
print('-----dt26-----')
print('dt26 frst 6 rows:',dt26_salaries_entir_01.head(6))
print('dt26 last 6 rows:',dt26_salaries_entir_01.tail(6))
print('-----dt27-----')
print('dt27 frst 6 rows:',dt27_income_job_typ_01.head(6))
print('dt27 last 6 rows:',dt27_income_job_typ_01.tail(6))
print('-----dt28-----')
print('dt28 frst 6 rows:',dt28_employment_eff_01.head(6))
print('dt28 last 6 rows:',dt28_employment_eff_01.tail(6))
print('-----dt29-----')
print('dt29 frst 6 rows:',dt29_income_median__01.head(6))
print('dt29 last 6 rows:',dt29_income_median__01.tail(6))
print('-----dt30-----')
print('dt30 frst 6 rows:',dt30_ROK_demo_____01.head(6))
print('dt30 last 6 rows:',dt30_ROK_demo_____01.tail(6))
print('-----dt31-----')
print('dt31 frst 6 rows:',dt31_ROK_demo_____02.head(6))
print('dt31 last 6 rows:',dt31_ROK_demo_____02.tail(6))
print('-----dt32-----')
print('dt32 frst 6 rows:',dt32_ROK_population_01.head(6))
print('dt32 last 6 rows:',dt32_ROK_population_01.tail(6))
print('-----dt33-----')
print('dt33 frst 6 rows:',dt33_ROK_population_01.head(6))
print('dt33 last 6 rows:',dt33_ROK_population_01.tail(6))
print('-----dt34-----')
print('dt34 frst 6 rows:',dt34_ROK_population_01.head(6))
print('dt34 last 6 rows:',dt34_ROK_population_01.tail(6))
print('-----dt35-----')
print('dt35 frst 6 rows:',dt35_ROK_population_01.head(6))
print('dt35 last 6 rows:',dt35_ROK_population_01.tail(6))
print('-----dt36-----')
print('dt36 frst 6 rows:',dt36_ROK_population_01.head(6))
print('dt36 last 6 rows:',dt36_ROK_population_01.tail(6))

```

```

print('
print('-----dt37-----
print('dt37 frst 6 rows:',dt37_ROK_population_01.head(6))
print('dt37 last 6 rows:',dt37_ROK_population_01.tail(6))
print('
print('-----dt38-----
print('dt38 frst 6 rows:',dt38_ROK_population_01.head(6))
print('dt38 last 6 rows:',dt38_ROK_population_01.tail(6))
print('
print('-----dt39-----
print('dt39 frst 6 rows:',dt39_ROK_population_01.head(6))
print('dt39 last 6 rows:',dt39_ROK_population_01.tail(6))
print('
print('-----dt40-----
print('dt40 frst 6 rows:',dt40_ROK_population_01.head(6))
print('dt40 last 6 rows:',dt40_ROK_population_01.tail(6))
print('
print('-----dt41-----
print('dt41 frst 6 rows:',dt41_ROK_population_01.head(6))
print('dt41 last 6 rows:',dt41_ROK_population_01.tail(6))
print('
print('-----dt42-----
print('dt42 frst 6 rows:',dt42_ROK_population_01.head(6))
print('dt42 last 6 rows:',dt42_ROK_population_01.tail(6))
print('
print('-----dt43-----
print('dt43 frst 6 rows:',dt43_Seoul_aprt____01.head(6))
print('dt43 last 6 rows:',dt43_Seoul_aprt____01.tail(6))
print('
print('-----dt44-----
print('dt44 frst 6 rows:',dt44_ROK_population_01.head(6))
print('dt44 last 6 rows:',dt44_ROK_population_01.tail(6))
print('
print('-----dt45-----
print('dt45 frst 6 rows:',dt45_RE_price____01.head(6))
print('dt45 last 6 rows:',dt45_RE_price____01.tail(6))

```

-----dt01-----

```

dt01 frst 6 rows:  region_01 region_02 2023_10_00  ...  2023_11_01 2023_12_00
2023_12_01
0      (1)      (2)      ...
1              96.7  ...      94.4      96.7      94.2
2              98.5  ...      93.1      98.8      93.1
3              94.4  ...      90.8      94.5      90.7
4             100.9  ...      99.6     101.2     99.6

```


5	Natural increase rate(per 1,000 population)	-1.1	-2.4	-2.4
dt03 last 6 rows:			item	2021
2022 2023 p)				
10	Crude marriage rate(per 1,000 population)	3.8	3.7	3.8
11	Divorces(cases)	101673.0	93232.0	92394
12	Crude divorce rate(per 1,000 population)	2.0	1.8	1.8
13	Life expectancy at birth-total(age)	83.6	82.7	-
14	Life expectancy at birth-male(age)	80.6	79.9	-
15	Life expectancy at birth-female(age)	86.6	85.6	-

-----dt04-----

dt04 frst 6 rows:	admin_division	...
2010_09		
0	By administrative divisions	... Housing units-Total number of rooms-8 and over...
1	Whole country	...
933343		
2	Whole country	...
87793		
3	Whole country	...
217854		
4	Whole country	...
217980		
5	Whole country	...
131279		

[6 rows x 13 columns]

dt04 last 6 rows:	admin_division		type_housing
... 2010_08	2010_09		
37	Whole country	Non-residential building in the housing	... 132 2495
38	Whole country	Non-residential building in the housing	... 45 1291
39	Whole country	Non-residential building in the housing	... 30 662
40	Whole country	Non-residential building in the housing	... 23 422
41	Whole country	Non-residential building in the housing	... 0 310
42	Whole country	Non-residential building in the housing	... 0 1235

[6 rows x 13 columns]

-----dt05-----

dt05 frst 6 rows:	category_01	category_02	...
2024_02_16 2024_02_17			
0	Category(1)	Category(1)	... Gyeongsangnam-do Jeju
1	Total	Total	... 169 255
2	Public sector	Sub-total	... 0 0

3	Sub-total	National public housing	...	0	0
4	Sub-total	Public rental housing	...	0	0
5	Sub-total	Public parcelling-out	...	0	0

[6 rows x 56 columns]

dt05 last 6 rows:			category_01	category_02	...	2024_02_16
2024_02_17						
15	Housing companies	National public housing	...	0	0	
16	Housing companies	Public rental housing	...	0	0	
17	Housing companies	Public parcelling-out	...	0	0	
18	Private sector	Sub-total	...	169	255	
19	Private sector	Private Rental Housing	...	0	0	
20	Private sector	Private parcelling-out	...	169	255	

[6 rows x 56 columns]

-----dt06-----

dt06 frst 6 rows:		area	size	2023_12	2024_01	2024_02
0	The Whole Country	Extra Small	107.7	107.1	107.7	
1	The Whole Country	Small	113.3	113.4	113.5	
2	The Whole Country	Mid-small	118.4	118.6	118.6	
3	The Whole Country	Mid-large	133.5	133.1	133.8	
4	The Whole Country	Large	143.5	140.9	140.7	
5	Seoul	Extra Small	121.3	121.2	123.7	

dt06 last 6 rows:		area	size	2023_12	2024_01
2024_02					
14	Seoul Metropolitan Area	Large	156.6	157.0	157.5
15	Non-Seoul Metropolitan Area	Extra Small	101.6	101.2	101.3
16	Non-Seoul Metropolitan Area	Small	102.5	102.5	102.5
17	Non-Seoul Metropolitan Area	Mid-small	105.6	105.6	105.5
18	Non-Seoul Metropolitan Area	Mid-large	116.6	116.0	115.8
19	Non-Seoul Metropolitan Area	Large	129.5	124.8	123.4

-----dt07-----

dt07 frst 6 rows:		area	2022_00	...	2022_04	2022_05
0	Classification(1)	Total	...	multiplex house	Apartment	
1	Whole country	96157	...	12130	13071	
2	Capital Area	44265	...	10209	5304	
3	Other Region	51892	...	1921	7767	
4	Seoul	17168	...	2244	1210	
5	Busan	6066	...	593	1529	

[6 rows x 7 columns]

dt07 last 6 rows:		area	2022_00	2022_01	2022_02	2022_03	2022_04
-------------------	--	------	---------	---------	---------	---------	---------

2022_05							
15	Chungcheongnam-do	3099	2798	215	0	86	0
16	Jeollabuk-do	3157	2626	127	0	34	370
17	Jeollanam-do	4153	4038	51	28	0	36
18	Gyeongsangbuk-do	5011	4472	230	82	27	200
19	Gyeongsangnam-do	6191	4964	160	27	103	937
20	Jeju	974	718	63	140	30	23

-----dt08-----

dt08 first 6 rows:	category_01	category_02	...	2024_02_16
2024_02_17				
0	Category(1)	Category(1)	...	Gyeongnam Jeju
1	Total	Total	...	1547 845
2	Public Sector	Total	...	0 0
3	Sum	National Rental	...	0 0
4	Sum	Public Rental	...	0 0
5	Sum	Public lotting-out	...	0 0

[6 rows x 56 columns]

dt08 last 6 rows:	category_01	category_02	...	2024_02_16
2024_02_17				
15	Housing Company	National Rental	...	0 0
16	Housing Company	Public Rental	...	0 0
17	Housing Company	Public lotting-out	...	0 0
18	Private Sector	Total	...	1547 845
19	Private	Private Rental	...	0 0
20	Private	Private lotting-out	...	1547 845

[6 rows x 56 columns]

-----dt09-----

dt09 first 6 rows:	type	region_01	region_02	...
2024_01	2024_02	2024_03		
0	Total	The Whole Country	Sub Summary	... 87.7 87.6
88.3				
1	Total	Seoul Metropolitan Area	Sub Summary	... 86.1 86.7
87.9				
2	Total	Non-Seoul Metropolitan Area	Sub Summary	... 89.1 88.4
88.6				
3	Total	6 Large Cities	Sub Summary	... 85.0 84.2
84.8				
4	Total	5 Large Cities	Sub Summary	... 84.9 83.5
84.1				
5	Total	9 Provinces	Sub Summary	... 89.8 89.9

90.1

[6 rows x 10 columns]

```
dt09 last 6 rows:           type region_01 region_02 ... 2024_01
2024_02 2024_03
158 Detached Houses Chungnam Sub Summary ... 99.5 98.8 99.0
159 Detached Houses Jeonbuk Sub Summary ... 92.2 93.4 92.7
160 Detached Houses Jeonnam Sub Summary ... 98.7 100.1 100.6
161 Detached Houses Gyeongbuk Sub Summary ... 100.8 100.6 100.9
162 Detached Houses Gyeongnam Sub Summary ... 97.0 96.1 97.0
163 Detached Houses Jeju Sub Summary ... 95.1 94.1 93.3
```

[6 rows x 10 columns]

```
-----
-----dt10-----
-----
dt10 frst 6 rows:           type           region_01 region_02 ...
2024_01 2024_02 2024_03
0 Total The Whole Country Sub Summary ... 91.8 92.0
92.6
1 Total Seoul Metropolitan Area Sub Summary ... 91.8 92.6
93.6
2 Total Non-Seoul Metropolitan Area Sub Summary ... 91.8 91.4
91.6
3 Total 6 Large Cities Sub Summary ... 88.6 88.3
89.6
4 Total 5 Large Cities Sub Summary ... 87.9 87.2
88.3
5 Total 9 Provinces Sub Summary ... 93.4 93.6
93.5
```

[6 rows x 10 columns]

```
dt10 last 6 rows:           type region_01 region_02 ... 2024_01
2024_02 2024_03
158 Detached Houses Chungnam Sub Summary ... 97.3 97.7 97.3
159 Detached Houses Jeonbuk Sub Summary ... 93.1 94.2 94.4
160 Detached Houses Jeonnam Sub Summary ... 97.9 97.7 97.7
161 Detached Houses Gyeongbuk Sub Summary ... 99.6 99.5 99.7
162 Detached Houses Gyeongnam Sub Summary ... 96.1 96.2 96.1
163 Detached Houses Jeju Sub Summary ... 96.2 96.4 95.4
```

[6 rows x 10 columns]

```
-----
-----dt11-----
-----
dt11 frst 6 rows: variant age 2022 2023 ... 2069 2070
```


	2071	2072							
0	Medium	TFR	0.78000	0.72000	...	1.08000	1.08000	1.08000	1.08000
1	Medium	15 years	0.00024	0.00001	...	0.00024	0.00024	0.00024	0.00024
2	Medium	16 years	0.00022	0.00002	...	0.00042	0.00042	0.00042	0.00042
3	Medium	17 years	0.00035	0.00004	...	0.00071	0.00071	0.00071	0.00071
4	Medium	18 years	0.00044	0.00010	...	0.00119	0.00119	0.00119	0.00119
5	Medium	19 years	0.00102	0.00021	...	0.00193	0.00193	0.00193	0.00193

[6 rows x 53 columns]

dt11 last 6 rows:			variant	age	2022	2023	...	2069	2070
	2071	2072							
30	Medium	44 years	0.00163	0.00138	...	0.00582	0.00582	0.00582	0.00582
31	Medium	45 years	0.00066	0.00062	...	0.00319	0.00319	0.00319	0.00319
32	Medium	46 years	0.00025	0.00025	...	0.00163	0.00163	0.00163	0.00163
33	Medium	47 years	0.00009	0.00010	...	0.00077	0.00077	0.00077	0.00077
34	Medium	48 years	0.00003	0.00003	...	0.00034	0.00034	0.00034	0.00034
35	Medium	49 years	0.00001	0.00001	...	0.00014	0.00014	0.00014	0.00014

[6 rows x 53 columns]

-----dt12-----

dt12 frst 6 rows:			variant	population_type
...	2071	2072		
0	Medium	Population in working ages(thousands): Total(1...	...	16855.0
		16575.0		
1	Medium	Population in working ages(thousands): 15-24 y...	...	2219.0
		2151.0		
2	Medium	Population in working ages(thousands): 25-49 y...	...	7661.0
		7641.0		
3	Medium	Population in working ages(thousands): 50-64 y...	...	6975.0
		6784.0		
4	Medium	-Population percentage(%): 15-24 years	...	6.0
		5.9		
5	Medium	-Population percentage(%): 25-49 years	...	20.9
		21.1		

[6 rows x 53 columns]

dt12 last 6 rows:			variant	population_type
...	2071	2072		
39	Medium	Population in youth ages(thousands): 19-34 years	...	4586.0
		4505.0		
40	Medium	-Population percentage(%): 19-34 years	...	12.5
		12.4		
41	Medium	Population in child ages(thousands): 0-14 years	...	2382.0
		2376.0		
42	Medium	-Population percentage(%): 0-14 years	...	6.5

6.6

43 Medium Population in child ages(thousands): 0-5 years ... 962.0
964.0

44 Medium -Population percentage(%): 0-5 years ... 2.6
2.7

[6 rows x 53 columns]

-----dt13-----

dt13 first 6 rows: admin_division ...
2022_19
0 By administrative divisions(eup, myeon, dong) ... Living quaters other than
housing unit (Housing)
1 Whole country ...
1162311
2 Total for eups ...
81935
3 Total for myeons ...
110037
4 Total for dongs ...
970339
5 Seoul ...
287651

[6 rows x 21 columns]

dt13 last 6 rows: admin_division 2022_00 2022_01 2022_02 ... 2022_16
2022_17 2022_18 2022_19
319 Jeju 676375 340665 335710 ... 31919 36792 6212
20148
320 Total for eups 154606 79775 74831 ... 8784 5822 1889
3592
321 Total for myeons 36397 18878 17519 ... 1399 1107 488
919
322 Total for dongs 485372 242012 243360 ... 21736 29863 3835
15637
323 Jeju-si 495281 248647 246634 ... 20525 29820 4224
15947
324 Seogwipo-si 181094 92018 89076 ... 11394 6972 1988
4201

[6 rows x 21 columns]

-----dt14-----

dt14 first 6 rows: admin_division ...

```

2005_19
0 By administrative divisions ... Living quaters other than housing unit
(Housing)
1           Whole country ...
206511
2           Dong area ...
184683
3           Eup area ...
7433
4           Myeon area ...
14395
5           Seoul ...
66992

```

```

[6 rows x 21 columns]
dt14 last 6 rows:      admin_division  2005_00  2005_01  ... 2005_17 2005_18
2005_19
15 Chungcheongnam-do  1889495   952674  ...   18223   10739   3756
16 Jeollabuk-do      1784013   877578  ...    3733    9152   1988
17 Jeollanam-do      1819819   892503  ...    5471   12417   5816
18 Gyeongsangbuk-do  2607641  1301568  ...   27898   19489   4561
19 Gyeongsangnam-do  3056356  1533029  ...   23546   17261  11751
20 Jeju-do           531887   264503  ...   18296    4209   2721

```

```

[6 rows x 21 columns]
-----
-----
-----dt15-----
-----
dt15 frst 6 rows:      admin_division  ...      1970_11
0 By administrative divisions ... Foreigner female (Person)
1           Whole country ...      14340
2           City area ...      11509
3           Eup area ...      1705
4           Myeon area ...      1126
5           Seoul ...      4867

```

```

[6 rows x 13 columns]
dt15 last 6 rows:      admin_division 1970_00 1970_01 1970_02 ... 1970_08
1970_09 1970_10 1970_11
10 Chungcheongnam-do  500119  499666   453 ... 1413318   2011   1023
988
11 Jeollabuk-do      427311  426967   344 ... 1216435   1685   873
812
12 Jeollanam-do      703851  703534   317 ... 1992042   1433   745
688
13 Gyeongsangbuk-do  850237  849538   699 ... 2272457   3226   1704
1522

```

14	Gyeongsangnam-do	574183	573888	295	...	1561776	1035	573
462								
15	Jeju-do	82955	82886	69	...	190009	293	157
136								

[6 rows x 13 columns]

```
-----
-----dt16-----
-----
```

dt16 frst 6 rows: admin_division ...

1995_16

0	By administrative divisions	...	Foreigner household(female) (Person)
1	Whole country	...	23044
2	Eup area	...	1382
3	Myeon area	...	2738
4	Dong area	...	18924
5	Seoul	...	6354

[6 rows x 18 columns]

dt16 last 6 rows: admin_division 1995_00 1995_01 ... 1995_14 1995_15

1995_16

14	Chungcheongnam-do	1766854	888460	...	1833	1033	800
15	Jeollabuk-do	1902044	942496	...	1486	583	903
16	Jeollanam-do	2066842	1023838	...	733	432	301
17	Gyeongsangbuk-do	2676312	1335643	...	3814	1884	1930
18	Gyeongsangnam-do	3845622	1934816	...	4069	2688	1381
19	Jeju-do	505438	249138	...	343	184	159

[6 rows x 18 columns]

```
-----
-----dt17-----
-----
```

dt17 frst 6 rows: admin_division ...

1990_16

0	By administrative divisions	...	Foreigner household(female) (Person)
1	Whole country	...	9290
2	City area	...	8562
3	Eup area	...	564
4	Myeon area	...	164
5	Seoul	...	4343

[6 rows x 18 columns]

dt17 last 6 rows: admin_division 1990_00 1990_01 ... 1990_14 1990_15

1990_16

14	Chungcheongnam-do	2013926	1021345	...	656	352	304
15	Jeollabuk-do	2069960	1030300	...	582	307	275

16	Jeollanam-do	2507439	1266653	...	495	307	188
17	Gyeongsangbuk-do	2860595	1430330	...	486	287	199
18	Gyeongsangnam-do	3672396	1842058	...	887	551	336
19	Jeju-do	514605	254322	...	169	90	79

[6 rows x 18 columns]

```
-----
-----dt18-----
-----
```

dt18 frst 6 rows: admin_division ...

1985_16

0	By administrative divisions	...	Foreigner household(female) (Person)
1	Whole country	...	12633
2	City area	...	11022
3	Eup area	...	1200
4	Myeon area	...	411
5	Seoul	...	5989

[6 rows x 18 columns]

dt18 last 6 rows: admin_division 1985_00 1985_01 ... 1985_14 1985_15

1985_16

12	Chungcheongnam-do	3001179	1511968	...	1342	708	634
13	Jeollabuk-do	2202078	1100948	...	813	438	375
14	Jeollanam-do	3748428	1887493	...	922	507	415
15	Gyeongsangbuk-do	3010945	1512918	...	944	556	388
16	Gyeongsangnam-do	3516660	1765080	...	2160	1521	639
17	Jeju-do	488576	239529	...	276	142	134

[6 rows x 18 columns]

```
-----
-----dt19-----
-----
```

dt19 frst 6 rows: admin_division ...

1980_16

0	By administrative divisions	...	Foreigner household(female) (Person)
1	Whole country	...	11605
2	City area	...	9484
3	Eup area	...	1323
4	Myeon area	...	798
5	Seoul	...	4947

[6 rows x 18 columns]

dt19 last 6 rows: admin_division 1980_00 1980_01 ... 1980_14 1980_15

1980_16

10	Chungcheongnam-do	2956214	1492697	...	1552	846	706
11	Jeollabuk-do	2287689	1145795	...	969	543	426

12	Jeollanam-do	3779736	1902595	...	959	516	443
13	Gyeongsangbuk-do	4954559	2483511	...	2547	1514	1033
14	Gyeongsangnam-do	3322132	1662540	...	1586	1025	561
15	Jeju-do	462941	226522	...	332	203	129

[6 rows x 18 columns]

```

-----
-----dt20-----
-----
dt20 frst 6 rows:          admin_division ...          1975_11
0 By administrative divisions ... Foreigner female (Person)
1           Whole country ...          11650
2           City area ...          9463
3           Eup area ...          1404
4           Myeon area ...          783
5           Seoul ...          3847

```

[6 rows x 13 columns]

```

dt20 last 6 rows:          admin_division 1975_00 1975_01 1975_02 ... 1975_08
1975_09 1975_10 1975_11
10 Chungcheongnam-do 535308 534947 361 ... 1448042 1530 788
742
11 Jeollabuk-do 444643 444356 287 ... 1219185 1310 733
577
12 Jeollanam-do 733138 732772 366 ... 1964343 1371 769
602
13 Gyeongsangbuk-do 953864 953131 733 ... 2409358 2699 1541
1158
14 Gyeongsangnam-do 628694 628286 408 ... 1636756 1334 859
475
15 Jeju-do 92150 92087 63 ... 212175 246 136
110

```

[6 rows x 13 columns]

```

-----
-----dt21-----
-----
dt21 frst 6 rows:          admin_division ...
2000_19
0 By administrative divisions ... Living quaters other than housing unit
(Housing)
1           Whole country ...
91814
2           Eup area ...
7254
3           Myeon area ...

```

16805

4 Dong area ...

67755

5 Seoul ...

21572

[6 rows x 21 columns]

dt21 last 6 rows: admin_division 2000_00 2000_01 ... 2000_17 2000_18
2000_19

15	Chungcheongnam-do	1845321	927824	...	8483	22993	3374
16	Jeollabuk-do	1890669	936683	...	2257	22055	2102
17	Jeollanam-do	1996456	988249	...	1832	30758	3229
18	Gyeongsangbuk-do	2724931	1361753	...	11072	38928	4566
19	Gyeongsangnam-do	2978502	1488847	...	10557	30656	6378
20	Jeju-do	513260	254449	...	9404	5929	2351

[6 rows x 21 columns]

-----dt22-----

dt22 frst 6 rows:	variant	gender		age	2022	...	2069
2070	2071	2072					
0	Medium	Total	Total	51672569	...	37670942	37181774
36222293							36698961
1	Medium	Total	0-4 years	1494041	...	796509	800555
804122							803192
2	Medium	Total	5-9 years	2142084	...	776167	777063
785080							780360
3	Medium	Total	10-14 years	2311839	...	837953	815634
787174							798743
4	Medium	Total	15-19 years	2287873	...	1028273	989573
916355							951796
5	Medium	Total	20-24 years	3105237	...	1321585	1296545
1234634							1267466

[6 rows x 54 columns]

dt22 last 6 rows: variant gender age ... 2070 2071
2072

63	Medium	Female	80 years and over	...	4520432	4485427	4479800
64	Medium	Female	80-84 years	...	1427757	1434155	1471905
65	Medium	Female	85-89 years	...	1440436	1378597	1312037
66	Medium	Female	90-94 years	...	1000044	1029912	1063593
67	Medium	Female	95-99 years	...	507533	492777	479859
68	Medium	Female	100 years and over	...	144662	149986	152406

[6 rows x 54 columns]

474	High-Low-Low	Median age	60.9	60.9	60.9
475	High-Low-Low	Mean age	56.2	56.2	56.1

-----dt25-----

dt25 frst 6 rows:	variant	dependency_ratio	...	2071
2072				
0	Medium	Total population(persons)	...	36698961.00 36222293.00
1	Medium	Male(persons)	...	18438434.00 18209029.00
2	Medium	Female(persons)	...	18260527.00 18013264.00
3	Medium	Sex ratio(males per 100 females)	...	101.00 101.10
4	Medium	Population Growth Rate	...	-1.31 -1.31
5	Medium	0-14 years(persons)	...	2382295.00 2376376.00

[6 rows x 53 columns]

dt25 last 6 rows:	variant	dependency_ratio	2022	2023	2024	...	2068
2069	2070	2071	2072				
15	Medium	Median age	44.9	45.5	46.1	...	63.0 63.1 63.2 63.3
63.4							
16	Medium	Median age-Male	43.6	44.0	44.6	...	61.7 61.9 62.0 62.1
62.2							
17	Medium	Median age-Female	46.4	47.0	47.6	...	64.4 64.4 64.4 64.5
64.6							
18	Medium	Mean age	43.9	44.4	44.9	...	58.4 58.5 58.5 58.6
58.6							
19	Medium	Mean age - male	42.7	43.2	43.7	...	57.2 57.3 57.4 57.5
57.6							
20	Medium	Mean age - female	45.0	45.6	46.1	...	59.5 59.6 59.6 59.7
59.7							

[6 rows x 53 columns]

-----dt26-----

dt26 frst 6 rows:	loans_personal_01	...	2022_02_02
0	average loan	...	5115
1	average loan	...	3269
2	average loan	...	1847
3	average loan	...	1222
4	average loan	...	1965
5	average loan	...	1718

[6 rows x 4 columns]

dt26 last 6 rows:	loans_personal_01	...	2022_02_02
8	median loan	...	5000
9	median loan	...	2142

10	median loan	...	2343
11	median loan	...	10240
12	median loan	...	3832
13	median loan	...	854

[6 rows x 4 columns]

-----dt27-----

dt27 first 6 rows:		classification_si_do	classification_income	
...	2019_04	2019_05		
0	Classification of Si, Do(1)	Classification of Income(1)	...	Free Total
1	Whole Country	Total	...	3.9 100.0
2	Whole Country	Low-Income Group	...	5.9 100.0
3	Whole Country	Middle-Income Group	...	3.2 100.0
4	Whole Country	High-Income Group	...	1.8 100.0
5	Seoul Metropolitan Area	Total	...	3.7 100.0

[6 rows x 8 columns]

dt27 last 6 rows:		classification_si_do	classification_income	...	2019_04
					2019_05
11	Metropolitan City Etc.	Middle-Income Group	...	2.9	100.0
12	Metropolitan City Etc.	High-Income Group	...	1.7	100.0
13	do Area	Total	...	4.8	100.0
14	do Area	Low-Income Group	...	6.8	100.0
15	do Area	Middle-Income Group	...	3.7	100.0
16	do Area	High-Income Group	...	2.1	100.0

[6 rows x 8 columns]

-----dt28-----

dt28 first 6 rows:		classification	...
			2022_04
0	Classification(1)	...	Ratio of Business Places below Standard Employ...
1	Total	...	18.24
2	Agriculture, Forestry and Fisheries	...	0.00
3	Mining Industry	...	0.00
4	Manufacturing	...	15.34
5	Electricity, Gas, Water Industry	...	1.64

[6 rows x 6 columns]

dt28 last 6 rows:		classification 2022_00	
...	2022_03 2022_04		
17	Education	124 ...	7
5.65			
18	Human Health and Social Work Activities	324 ...	8
2.47			
19	Arts, Sports and Recreation Related Services(9...	51 ...	7
13.73			
20	Membership organizations, Repair and Other Per...	34 ...	4
11.76			
21	Private Households with Employed Persons and U...	0 ...	0
0.00			
22	Extra-Territorial Organizations and Bodies	4 ...	0
0.00			

[6 rows x 6 columns]

 -----dt29-----

dt29 first 6 rows:		item 2018 ...	
2021	2022		
0	Average income	297.0 ... 333.0	353.0
1	Median income	220.0 ... 250.0	267.0
2	Lower than 50% of the median income	20.4 ... 20.0	19.9
3	50% to 150% of the median income	49.0 ... 50.5	51.1
4	Higher than or equal 150% of the median income	30.6 ... 29.5	29.0
5	Lower 20% threshold (P20)	108.0 ... 125.0	135.0

[6 rows x 6 columns]

dt29 last 6 rows:		item 2018 ...	
2021	2022		
3	50% to 150% of the median income	49.0 ... 50.5	51.1
4	Higher than or equal 150% of the median income	30.6 ... 29.5	29.0
5	Lower 20% threshold (P20)	108.0 ... 125.0	135.0
6	Lower 40% threshold (P40)	183.0 ... 210.0	225.0
7	Lower 60% threshold (P60)	269.0 ... 300.0	318.0
8	Lower 80% threshold (P80)	436.0 ... 478.0	503.0

[6 rows x 6 columns]

 -----dt30-----

dt30 first 6 rows:	index	id	year	...	occupation	company_size
unemployment_reason						

0	0	10101	2005	...	8
1	1	10101	2011	...	10
2	2	10101	2012	...	10
3	3	10101	2013	...	1
4	4	10101	2014	...	10
5	5	10101	2015	...	10

[6 rows x 15 columns]

dt30 last 6 rows:				index	id	year	...	occupation	company_size	unemployment_reason
92851	92851	98000701	2013	...	874			1		
92852	92852	98000701	2014	...	874			1		
92853	92853	98000701	2015	...	874			1		
92854	92854	98000701	2016	...	874			1		
92855	92855	98000701	2017	...	874			1		
92856	92856	98000701	2018	...	874			1		

[6 rows x 15 columns]

-----dt31-----

dt31 frst 6 rows:				By variant	...	2072
0	Medium	...	36222293.00			
1	Medium	...	18209029.00			
2	Medium	...	18013264.00			
3	Medium	...	101.10			
4	Medium	...	-1.31			
5	Medium	...	2376376.00			

[6 rows x 53 columns]

dt31 last 6 rows:				By variant	By the structure of population and dependency ratio	...	2071	2072
15	Medium			Median age	...	63.3		
63.4								
16	Medium			Median age-Male	...	62.1		
62.2								
17	Medium			Median age-Female	...	64.5		
64.6								
18	Medium			Mean age	...	58.6		
58.6								
19	Medium			Mean age - male	...	57.5		
57.6								
20	Medium			Mean age - female	...	59.7		
59.7								

[6 rows x 53 columns]

 -----dt32-----

dt32 first 6 rows:	Date	Region	...	Natural_growth
Natural_growth_rate				
0 1/1/2000	Busan	...	1877.0	5.8
1 1/1/2000	Chungcheongbuk-do	...	979.0	7.7
2 1/1/2000	Chungcheongnam-do	...	932.0	5.7
3 1/1/2000	Daegu	...	1940.0	9.1
4 1/1/2000	Daejeon	...	1294.0	11.2
5 1/1/2000	Gangwon-do	...	899.0	6.8

[6 rows x 12 columns]

dt32 last 6 rows:	Date	Region	...	Natural_growth
Natural_growth_rate				
4854 6/1/2022	Jeollabuk-do	...	-632.0	-4.3
4855 6/1/2022	Jeollanam-do	...	-805.0	-5.4
4856 6/1/2022	Sejong	...	141.0	4.5
4857 6/1/2022	Seoul	...	-494.0	-0.6
4858 6/1/2022	Ulsan	...	9.0	0.1
4859 6/1/2022	Whole country	...	-6019.0	-1.4

[6 rows x 12 columns]

 -----dt33-----

dt33 first 6 rows:	By administrative divisions(eup, myeon, dong)	...
2022.19		
0	By administrative divisions(eup, myeon, dong)	Living quarters other than housing unit (Housing)
1	Whole country	...
1162311		
2	Total for eups	...
81935		
3	Total for myeons	...
110037		
4	Total for dongs	...
970339		
5	Seoul	...
287651		

[6 rows x 21 columns]

dt33 last 6 rows:	By administrative divisions(eup, myeon, dong)	2022	...
2022.18 2022.19			
319	Jeju	676375	6212 20148
320	Total for eups	154606	1889 3592
321	Total for myeons	36397	488 919

322	Total for dongs	485372	...	3835	15637
323	Jeju-si	495281	...	4224	15947
324	Seogwipo-si	181094	...	1988	4201

[6 rows x 21 columns]

-----dt34-----

dt34 frst 6 rows: By administrative divisions(eup, myeon, dong) ...
2010.19
0 By administrative divisions(eup, myeon, dong) ... Living quaters other than
housing unit (Housing)
1 Whole country ...
303097
2 Total for dongs ...
270518
3 Total for eups ...
9816
4 Total for myeons ...
22763
5 Seoul ...
97389

[6 rows x 21 columns]

dt34 last 6 rows: By administrative divisions(eup, myeon, dong) 2010 ...
2010.18 2010.19
15 Chungcheongnam-do 2028002 ... 8617 7489
16 Jeollabuk-do 1777220 ... 6403 2850
17 Jeollanam-do 1741499 ... 9045 5442
18 Gyeongsangbuk-do 2600032 ... 14479 6036
19 Gyeongsangnam-do 3160154 ... 12123 16056
20 Jeju 531905 ... 2988 4238

[6 rows x 21 columns]

-----dt35-----

dt35 frst 6 rows: By administrative divisions ...
2005.19
0 By administrative divisions ... Living quaters other than housing unit
(Housing)
1 Whole country ...
206511
2 Dong area ...
184683
3 Eup area ...

7433

4 Myeon area ...

14395

5 Seoul ...

66992

[6 rows x 21 columns]

dt35 last 6 rows: By administrative divisions 2005 2005.1 ... 2005.17
2005.18 2005.19

15	Chungcheongnam-do	1889495	952674	...	18223	10739	3756
16	Jeollabuk-do	1784013	877578	...	3733	9152	1988
17	Jeollanam-do	1819819	892503	...	5471	12417	5816
18	Gyeongsangbuk-do	2607641	1301568	...	27898	19489	4561
19	Gyeongsangnam-do	3056356	1533029	...	23546	17261	11751
20	Jeju-do	531887	264503	...	18296	4209	2721

[6 rows x 21 columns]

-----dt36-----

dt36 frst 6 rows: By administrative divisions ...
2000.19

0 By administrative divisions ... Living quaters other than housing unit
(Housing)

1 Whole country ...

91814

2 Eup area ...

7254

3 Myeon area ...

16805

4 Dong area ...

67755

5 Seoul ...

21572

[6 rows x 21 columns]

dt36 last 6 rows: By administrative divisions 2000 2000.1 ... 2000.17
2000.18 2000.19

15	Chungcheongnam-do	1845321	927824	...	8483	22993	3374
16	Jeollabuk-do	1890669	936683	...	2257	22055	2102
17	Jeollanam-do	1996456	988249	...	1832	30758	3229
18	Gyeongsangbuk-do	2724931	1361753	...	11072	38928	4566
19	Gyeongsangnam-do	2978502	1488847	...	10557	30656	6378
20	Jeju-do	513260	254449	...	9404	5929	2351

[6 rows x 21 columns]

 -----dt37-----

dt37 frst 6 rows: By administrative divisions ...
 1995.16

0	By administrative divisions	...	Foreigner household(female) (Person)	
1	Whole country	...		23044
2	Eup area	...		1382
3	Myeon area	...		2738
4	Dong area	...		18924
5	Seoul	...		6354

[6 rows x 18 columns]

dt37 last 6 rows: By administrative divisions 1995 1995.1 ... 1995.14
 1995.15 1995.16

14	Chungcheongnam-do	1766854	888460	...	1833	1033	800
15	Jeollabuk-do	1902044	942496	...	1486	583	903
16	Jeollanam-do	2066842	1023838	...	733	432	301
17	Gyeongsangbuk-do	2676312	1335643	...	3814	1884	1930
18	Gyeongsangnam-do	3845622	1934816	...	4069	2688	1381
19	Jeju-do	505438	249138	...	343	184	159

[6 rows x 18 columns]

 -----dt38-----

dt38 frst 6 rows: By administrative divisions ...
 1990.16

0	By administrative divisions	...	Foreigner household(female) (Person)	
1	Whole country	...		9290
2	City area	...		8562
3	Eup area	...		564
4	Myeon area	...		164
5	Seoul	...		4343

[6 rows x 18 columns]

dt38 last 6 rows: By administrative divisions 1990 1990.1 ... 1990.14
 1990.15 1990.16

14	Chungcheongnam-do	2013926	1021345	...	656	352	304
15	Jeollabuk-do	2069960	1030300	...	582	307	275
16	Jeollanam-do	2507439	1266653	...	495	307	188
17	Gyeongsangbuk-do	2860595	1430330	...	486	287	199
18	Gyeongsangnam-do	3672396	1842058	...	887	551	336
19	Jeju-do	514605	254322	...	169	90	79

[6 rows x 18 columns]

 -----dt39-----

dt39 frst 6 rows: By administrative divisions ...
 1985.16

0	By administrative divisions	...	Foreigner household(female) (Person)	
1	Whole country	...		12633
2	City area	...		11022
3	Eup area	...		1200
4	Myeon area	...		411
5	Seoul	...		5989

[6 rows x 18 columns]

dt39 last 6 rows: By administrative divisions 1985 1985.1 ... 1985.14
 1985.15 1985.16

12	Chungcheongnam-do	3001179	1511968	...	1342	708	634
13	Jeollabuk-do	2202078	1100948	...	813	438	375
14	Jeollanam-do	3748428	1887493	...	922	507	415
15	Gyeongsangbuk-do	3010945	1512918	...	944	556	388
16	Gyeongsangnam-do	3516660	1765080	...	2160	1521	639
17	Jeju-do	488576	239529	...	276	142	134

[6 rows x 18 columns]

 -----dt40-----

dt40 frst 6 rows: By administrative divisions ...
 1980.16

0	By administrative divisions	...	Foreigner household(female) (Person)	
1	Whole country	...		11605
2	City area	...		9484
3	Eup area	...		1323
4	Myeon area	...		798
5	Seoul	...		4947

[6 rows x 18 columns]

dt40 last 6 rows: By administrative divisions 1980 1980.1 ... 1980.14
 1980.15 1980.16

10	Chungcheongnam-do	2956214	1492697	...	1552	846	706
11	Jeollabuk-do	2287689	1145795	...	969	543	426
12	Jeollanam-do	3779736	1902595	...	959	516	443
13	Gyeongsangbuk-do	4954559	2483511	...	2547	1514	1033
14	Gyeongsangnam-do	3322132	1662540	...	1586	1025	561
15	Jeju-do	462941	226522	...	332	203	129

[6 rows x 18 columns]

-----dt41-----

dt41 frst 6 rows: By administrative divisions ... 1975.11

0	By administrative divisions	...	Foreigner female (Person)
1	Whole country	...	11650
2	City area	...	9463
3	Eup area	...	1404
4	Myeon area	...	783
5	Seoul	...	3847

[6 rows x 13 columns]

dt41 last 6 rows: By administrative divisions 1975 1975.1 ... 1975.9
1975.10 1975.11

10	Chungcheongnam-do	535308	534947	...	1530	788	742
11	Jeollabuk-do	444643	444356	...	1310	733	577
12	Jeollanam-do	733138	732772	...	1371	769	602
13	Gyeongsangbuk-do	953864	953131	...	2699	1541	1158
14	Gyeongsangnam-do	628694	628286	...	1334	859	475
15	Jeju-do	92150	92087	...	246	136	110

[6 rows x 13 columns]

-----dt42-----

dt42 frst 6 rows: By administrative divisions ... 1970.11

0	By administrative divisions	...	Foreigner female (Person)
1	Whole country	...	14340
2	City area	...	11509
3	Eup area	...	1705
4	Myeon area	...	1126
5	Seoul	...	4867

[6 rows x 13 columns]

dt42 last 6 rows: By administrative divisions 1970 1970.1 ... 1970.9
1970.10 1970.11

10	Chungcheongnam-do	500119	499666	...	2011	1023	988
11	Jeollabuk-do	427311	426967	...	1685	873	812
12	Jeollanam-do	703851	703534	...	1433	745	688
13	Gyeongsangbuk-do	850237	849538	...	3226	1704	1522
14	Gyeongsangnam-do	574183	573888	...	1035	573	462
15	Jeju-do	82955	82886	...	293	157	136

[6 rows x 13 columns]

-----dt43-----

```

-----
dt43 frst 6 rows:      index      name gugun dong ... avg_sales  area  floor
pricePerArea
0      0              ...  101000.0  139    42    697.841727
1      1              ...   89000.0  105    32    790.476191
2      2      2      ...   71500.0   86    26    720.930233
3      3              ...   65500.0  102    31    637.254902
4      4              ...   70000.0   91    28    692.307692
5      5              ...   62000.0   84    25    630.952381

```

[6 rows x 11 columns]

```

dt43 last 6 rows:      index      name gugun  dong ... avg_sales  area
floor pricePerArea
4538    45              ...   88000.0  109    33    807.339450
4539    46              ...   99000.0  113    34    876.106195
4540    47      5      ...   89500.0   80    24    1000.000000
4541    48              ...   84500.0   83    25    891.566265
4542    49              ...  109000.0   93    28   1043.010753
4543    50              ...   98500.0  117    35    717.948718

```

[6 rows x 11 columns]

```

-----
-----dt44-----
-----

```

```

dt44 frst 6 rows:      Date              Region ... Natural_growth
Natural_growth_rate
0  1/1/2000      Busan ...           1877.0              5.8
1  1/1/2000  Chungcheongbuk-do ...           979.0              7.7
2  1/1/2000  Chungcheongnam-do ...           932.0              5.7
3  1/1/2000      Daegu ...          1940.0              9.1
4  1/1/2000     Daejeon ...          1294.0             11.2
5  1/1/2000  Gangwon-do ...           899.0              6.8

```

[6 rows x 12 columns]

```

dt44 last 6 rows:      Date              Region ... Natural_growth
Natural_growth_rate
4854  6/1/2022  Jeollabuk-do ...          -632.0             -4.3
4855  6/1/2022  Jeollanam-do ...          -805.0             -5.4
4856  6/1/2022      Sejong ...           141.0              4.5
4857  6/1/2022      Seoul ...          -494.0             -0.6
4858  6/1/2022      Ulsan ...            9.0              0.1
4859  6/1/2022  Whole country ...        -6019.0             -1.4

```

[6 rows x 12 columns]

```

-----
-----dt45-----
-----

```

```

-----
dt45 first 6 rows:
2012_10 2012_11 2012_12
0 Whole Country Total 22.6 22.9 ... 107.4 107.9 108.3
108.6
1 Whole Country Apartments 14.4 14.8 ... 109.3 110.0 110.6
110.9
2 Whole Country single 41.1 41.4 ... 103.5 103.7 103.9
103.9
3 Whole Country Row Houses 22.1 22.2 ... 105.3 105.6 105.7
105.8
4 Seoul Total 23.0 23.5 ... 106.0 106.5 106.8
107.0
5 Seoul Apartments 14.2 14.8 ... 107.2 107.9 108.4
108.7

```

[6 rows x 326 columns]

```

dt45 last 6 rows:
2012_11 2012_12
51 Non Metropolitan Area single - ... 104.4 104.5 104.6
52 Non Metropolitan Area Row Houses - ... 108.2 108.5 108.9
53 5 Metropolitan Cities Total - ... 109.4 109.8 110.1
54 5 Metropolitan Cities Apartments - ... 111.5 112.0 112.4
55 5 Metropolitan Cities single - ... 104.2 104.4 104.5
56 5 Metropolitan Cities Row Houses - ... 107.7 107.9 108.0

```

[6 rows x 326 columns]

```

[88]: # 04.02.01-06
# return types
# dt01-dt44

print('-----')
print('-----dt01-----')
print('dt01 types:',dt01_real_estate_pr_01.dtypes)
print('-----')
print('-----dt02-----')
print('dt02 types:',dt02_real_estate_se_01.dtypes)
print('-----')
print('-----dt03-----')
print('dt03 types:',dt03_stats_vital_kr_01.dtypes)
print('-----')
print('-----dt04-----')
print('dt04 types:',dt04_housing_census_01.dtypes)
print('-----')
print('-----dt05-----')
print('dt05 types:',dt05_housing_constr_01.dtypes)

```

```

print('
print('-----dt06-----
print('dt06 types:',dt06_household_empty_02.dtypes)
print('
print('-----dt07-----
print('dt07 types:',dt07_housing_loss__01.dtypes)
print('
print('-----dt08-----
print('dt08 types:',dt08_housing_use_sv_01.dtypes)
print('
print('-----dt09-----
print('dt09 types:',dt09_jeonse_sales__01.dtypes)
print('
print('-----dt10-----
print('dt10 types:',dt10_jeonse_market__02.dtypes)
print('
print('-----dt11-----
print('dt11 types:',dt11_birthrate_age__01.dtypes)
print('
print('-----dt12-----
print('dt12 types:',dt12_birthrate_age__02.dtypes)
print('
print('-----dt13-----
print('dt13 types:',dt13_pop_houshold__01.dtypes)
print('
print('-----dt14-----
print('dt14 types:',dt14_pop_houshold__02.dtypes)
print('
print('-----dt15-----
print('dt15 types:',dt15_pop_houshold__03.dtypes)
print('
print('-----dt16-----
print('dt16 types:',dt16_pop_houshold__04.dtypes)
print('
print('-----dt17-----
print('dt17 types:',dt17_pop_houshold__05.dtypes)
print('
print('-----dt18-----
print('dt18 types:',dt18_pop_houshold__06.dtypes)
print('
print('-----dt19-----
print('dt19 types:',dt19_pop_houshold__07.dtypes)
print('
print('-----dt20-----
print('dt20 types:',dt20_pop_houshold__08.dtypes)
print('
print('-----dt21-----

```

```

print('dt21 types:',dt21_pop_houshold__09.dtypes)
print('-----dt22-----')
print('dt22 types:',dt22_pop_estimate__01.dtypes)
print('-----dt23-----')
print('dt23 types:',dt23_pop_future_pro_01.dtypes)
print('-----dt24-----')
print('dt24 types:',dt24_pop_future_pro_02.dtypes)
print('-----dt25-----')
print('dt25 types:',dt25_pop_future_pro_02.dtypes)
print('-----dt26-----')
print('dt26 types:',dt26_salaries_entir_01.dtypes)
print('-----dt27-----')
print('dt27 types:',dt27_income_job_typ_01.dtypes)
print('-----dt28-----')
print('dt28 types:',dt28_employment_eff_01.dtypes)
print('-----dt29-----')
print('dt29 types:',dt29_income_median__01.dtypes)
print('-----dt30-----')
print('dt30 types:',dt30_ROK_demo______01.dtypes)
print('-----dt31-----')
print('dt31 types:',dt31_ROK_demo______02.dtypes)
print('-----dt32-----')
print('dt32 types:',dt32_ROK_population_01.dtypes)
print('-----dt33-----')
print('dt33 types:',dt33_ROK_population_01.dtypes)
print('-----dt34-----')
print('dt34 types:',dt34_ROK_population_01.dtypes)
print('-----dt35-----')
print('dt35 types:',dt35_ROK_population_01.dtypes)
print('-----dt36-----')
print('dt36 types:',dt36_ROK_population_01.dtypes)
print('-----')

```

```

print('-----dt37-----')
print('dt37 types:',dt37_ROK_population_01.dtypes)
print('-----dt38-----')
print('dt38 types:',dt38_ROK_population_01.dtypes)
print('-----dt39-----')
print('dt39 types:',dt39_ROK_population_01.dtypes)
print('-----dt40-----')
print('dt40 types:',dt40_ROK_population_01.dtypes)
print('-----dt41-----')
print('dt41 types:',dt41_ROK_population_01.dtypes)
print('-----dt42-----')
print('dt42 types:',dt42_ROK_population_01.dtypes)
print('-----dt43-----')
print('dt43 types:',dt43_Seoul_apartment_01.dtypes)
print('-----dt44-----')
print('dt44 types:',dt44_ROK_population_01.dtypes)
print('-----dt45-----')
print('dt45 types:',dt45_RE_price_01.dtypes)

```

```

-----
-----dt01-----
-----

```

```

dt01 types: region_01      object
region_02      object
2023_10_00      object
2023_10_01      object
2023_11_00      object
2023_11_01      object
2023_12_00      object
2023_12_01      object
dtype: object

```

```

-----
-----dt02-----
-----

```

```

dt02 types: id            int64
lat          float64
lon          float64
households   int64

```

```

buildDate      int64
score          float64
m2             int64
pyung          int64
sales_min      float64
sales_max      float64
sales_ave      float64
dtype: object

```

```

-----
-----dt03-----
-----

```

```

dt03 types: item      object
2021        float64
2022        float64
2023 p)     object
dtype: object

```

```

-----
-----dt04-----
-----

```

```

dt04 types: admin_division      object
type_housing                    object
household_number_dwelling      object
2010_00                         object
2010_01                         object
2010_02                         object
2010_03                         object
2010_04                         object
2010_05                         object
2010_06                         object
2010_07                         object
2010_08                         object
2010_09                         object
dtype: object

```

```

-----
-----dt05-----
-----

```

```

dt05 types: category_01      object
category_02                  object
2023_12_00                   object
2023_12_01                   object
2023_12_02                   object
2023_12_03                   object
2023_12_04                   object
2023_12_05                   object
2023_12_06                   object

```


2023_12_07	object
2023_12_08	object
2023_12_09	object
2023_12_10	object
2023_12_11	object
2023_12_12	object
2023_12_13	object
2023_12_14	object
2023_12_15	object
2023_12_16	object
2023_12_17	object
2024_01_00	object
2024_01_01	object
2024_01_02	object
2024_01_03	object
2024_01_04	object
2024_01_05	object
2024_01_06	object
2024_01_07	object
2024_01_08	object
2024_01_09	object
2024_01_10	object
2024_01_11	object
2024_01_12	object
2024_01_13	object
2024_01_14	object
2024_01_15	object
2024_01_16	object
2024_01_17	object
2024_02_00	object
2024_02_01	object
2024_02_02	object
2024_02_03	object
2024_02_04	object
2024_02_05	object
2024_02_06	object
2024_02_07	object
2024_02_08	object
2024_02_09	object
2024_02_10	object
2024_02_11	object
2024_02_12	object
2024_02_13	object
2024_02_14	object
2024_02_15	object
2024_02_16	object
2024_02_17	object

dtype: object

```

-----
-----dt06-----
-----
dt06 types: area          object
size          object
2023_12       float64
2024_01       float64
2024_02       float64
dtype: object
-----
-----dt07-----
-----
dt07 types: area          object
2022_00       object
2022_01       object
2022_02       object
2022_03       object
2022_04       object
2022_05       object
dtype: object
-----
-----dt08-----
-----
dt08 types: category_01   object
category_02   object
2023_12_00    object
2023_12_01    object
2023_12_02    object
2023_12_03    object
2023_12_04    object
2023_12_05    object
2023_12_06    object
2023_12_07    object
2023_12_08    object
2023_12_09    object
2023_12_10    object
2023_12_11    object
2023_12_12    object
2023_12_13    object
2023_12_14    object
2023_12_15    object
2023_12_16    object
2023_12_17    object
2024_01_00    object
2024_01_01    object

```

```

2024_01_02    object
2024_01_03    object
2024_01_04    object
2024_01_05    object
2024_01_06    object
2024_01_07    object
2024_01_08    object
2024_01_09    object
2024_01_10    object
2024_01_11    object
2024_01_12    object
2024_01_13    object
2024_01_14    object
2024_01_15    object
2024_01_16    object
2024_01_17    object
2024_02_00    object
2024_02_01    object
2024_02_02    object
2024_02_03    object
2024_02_04    object
2024_02_05    object
2024_02_06    object
2024_02_07    object
2024_02_08    object
2024_02_09    object
2024_02_10    object
2024_02_11    object
2024_02_12    object
2024_02_13    object
2024_02_14    object
2024_02_15    object
2024_02_16    object
2024_02_17    object

```

```
dtype: object
```

```
-----dt09-----
```

```

dt09 types: type          object
region_01    object
region_02    object
region_03    object
2023_10      float64
2023_11      float64
2023_12      float64
2024_01      float64
2024_02      float64

```

```
2024_03      float64
dtype: object
```

```
-----
-----
-----dt10-----
-----
```

```
dt10 types: type      object
region_01      object
region_02      object
region_03      object
2023_10        float64
2023_11        float64
2023_12        float64
2024_01        float64
2024_02        float64
2024_03        float64
dtype: object
```

```
-----
-----
-----dt11-----
-----
```

```
dt11 types: variant    object
age          object
2022         float64
2023         float64
2024         float64
2025         float64
2026         float64
2027         float64
2028         float64
2029         float64
2030         float64
2031         float64
2032         float64
2033         float64
2034         float64
2035         float64
2036         float64
2037         float64
2038         float64
2039         float64
2040         float64
2041         float64
2042         float64
2043         float64
2044         float64
2045         float64
2046         float64
```

2047	float64
2048	float64
2049	float64
2050	float64
2051	float64
2052	float64
2053	float64
2054	float64
2055	float64
2056	float64
2057	float64
2058	float64
2059	float64
2060	float64
2061	float64
2062	float64
2063	float64
2064	float64
2065	float64
2066	float64
2067	float64
2068	float64
2069	float64
2070	float64
2071	float64
2072	float64

dtype: object

```
-----
-----dt12-----
-----
```

dt12 types: variant object

population_type	object
2022	float64
2023	float64
2024	float64
2025	float64
2026	float64
2027	float64
2028	float64
2029	float64
2030	float64
2031	float64
2032	float64
2033	float64
2034	float64
2035	float64
2036	float64

2037	float64
2038	float64
2039	float64
2040	float64
2041	float64
2042	float64
2043	float64
2044	float64
2045	float64
2046	float64
2047	float64
2048	float64
2049	float64
2050	float64
2051	float64
2052	float64
2053	float64
2054	float64
2055	float64
2056	float64
2057	float64
2058	float64
2059	float64
2060	float64
2061	float64
2062	float64
2063	float64
2064	float64
2065	float64
2066	float64
2067	float64
2068	float64
2069	float64
2070	float64
2071	float64
2072	float64

dtype: object

```
-----
-----dt13-----
-----
```

dt13 types: admin_division object

2022_00	object
2022_01	object
2022_02	object
2022_03	object
2022_04	object
2022_05	object

2022_06	object
2022_07	object
2022_08	object
2022_09	object
2022_10	object
2022_11	object
2022_12	object
2022_13	object
2022_14	object
2022_15	object
2022_16	object
2022_17	object
2022_18	object
2022_19	object

dtype: object

```
-----
-----
-----dt14-----
-----
```

dt14 types: admin_division		object
2005_00	object	
2005_01	object	
2005_02	object	
2005_03	object	
2005_04	object	
2005_05	object	
2005_06	object	
2005_07	object	
2005_08	object	
2005_09	object	
2005_10	object	
2005_11	object	
2005_12	object	
2005_13	object	
2005_14	object	
2005_15	object	
2005_16	object	
2005_17	object	
2005_18	object	
2005_19	object	

dtype: object

```
-----
-----
-----dt15-----
-----
```

dt15 types: admin_division		object
1970_00	object	
1970_01	object	

1970_02	object
1970_03	object
1970_04	object
1970_05	object
1970_06	object
1970_07	object
1970_08	object
1970_09	object
1970_10	object
1970_11	object

dtype: object

```
-----
-----dt16-----
-----
```

dt16 types: admin_division object

1995_00	object
1995_01	object
1995_02	object
1995_03	object
1995_04	object
1995_05	object
1995_06	object
1995_07	object
1995_08	object
1995_09	object
1995_10	object
1995_11	object
1995_12	object
1995_13	object
1995_14	object
1995_15	object
1995_16	object

dtype: object

```
-----
-----dt17-----
-----
```

dt17 types: admin_division object

1990_00	object
1990_01	object
1990_02	object
1990_03	object
1990_04	object
1990_05	object
1990_06	object
1990_07	object
1990_08	object


```
1990_09      object
1990_10      object
1990_11      object
1990_12      object
1990_13      object
1990_14      object
1990_15      object
1990_16      object
dtype: object
```

```
-----
-----dt18-----
-----
```

```
dt18 types: admin_division      object
1985_00      object
1985_01      object
1985_02      object
1985_03      object
1985_04      object
1985_05      object
1985_06      object
1985_07      object
1985_08      object
1985_09      object
1985_10      object
1985_11      object
1985_12      object
1985_13      object
1985_14      object
1985_15      object
1985_16      object
dtype: object
```

```
-----
-----dt19-----
-----
```

```
dt19 types: admin_division      object
1980_00      object
1980_01      object
1980_02      object
1980_03      object
1980_04      object
1980_05      object
1980_06      object
1980_07      object
1980_08      object
1980_09      object
1980_10      object
```

```
1980_11      object
1980_12      object
1980_13      object
1980_14      object
1980_15      object
1980_16      object
dtype: object
```

```
-----
-----dt20-----
-----
```

```
dt20 types: admin_division      object
1975_00      object
1975_01      object
1975_02      object
1975_03      object
1975_04      object
1975_05      object
1975_06      object
1975_07      object
1975_08      object
1975_09      object
1975_10      object
1975_11      object
dtype: object
```

```
-----
-----dt21-----
-----
```

```
dt21 types: admin_division      object
2000_00      object
2000_01      object
2000_02      object
2000_03      object
2000_04      object
2000_05      object
2000_06      object
2000_07      object
2000_08      object
2000_09      object
2000_10      object
2000_11      object
2000_12      object
2000_13      object
2000_14      object
2000_15      object
2000_16      object
2000_17      object
```

```
2000_18      object
2000_19      object
dtype: object
```

```
-----
-----dt22-----
-----
```

```
dt22 types: variant    object
gender      object
age         object
2022        int64
2023        int64
2024        int64
2025        int64
2026        int64
2027        int64
2028        int64
2029        int64
2030        int64
2031        int64
2032        int64
2033        int64
2034        int64
2035        int64
2036        int64
2037        int64
2038        int64
2039        int64
2040        int64
2041        int64
2042        int64
2043        int64
2044        int64
2045        int64
2046        int64
2047        int64
2048        int64
2049        int64
2050        int64
2051        int64
2052        int64
2053        int64
2054        int64
2055        int64
2056        int64
2057        int64
2058        int64
2059        int64
```

2060	int64
2061	int64
2062	int64
2063	int64
2064	int64
2065	int64
2066	int64
2067	int64
2068	int64
2069	int64
2070	int64
2071	int64
2072	int64

dtype: object

```
-----
-----dt23-----
-----
```

dt23 types: variant object

cause	object
-------	--------

2022	float64
2023	float64
2024	float64
2025	float64
2026	float64
2027	float64
2028	float64
2029	float64
2030	float64
2031	float64
2032	float64
2033	float64
2034	float64
2035	float64
2036	float64
2037	float64
2038	float64
2039	float64
2040	float64
2041	float64
2042	float64
2043	float64
2044	float64
2045	float64
2046	float64
2047	float64
2048	float64
2049	float64

2050	float64
2051	float64
2052	float64
2053	float64
2054	float64
2055	float64
2056	float64
2057	float64
2058	float64
2059	float64
2060	float64
2061	float64
2062	float64
2063	float64
2064	float64
2065	float64
2066	float64
2067	float64
2068	float64
2069	float64
2070	float64
2071	float64
2072	float64

dtype: object

```
-----
-----dt24-----
-----
```

dt24 types:	By scenarios	object
structure		object
2070		float64
2071		float64
2072		float64

dtype: object

```
-----
-----dt25-----
-----
```

dt25 types:	variant	object
dependency_ratio		object
2022		float64
2023		float64
2024		float64
2025		float64
2026		float64
2027		float64
2028		float64
2029		float64

2030	float64
2031	float64
2032	float64
2033	float64
2034	float64
2035	float64
2036	float64
2037	float64
2038	float64
2039	float64
2040	float64
2041	float64
2042	float64
2043	float64
2044	float64
2045	float64
2046	float64
2047	float64
2048	float64
2049	float64
2050	float64
2051	float64
2052	float64
2053	float64
2054	float64
2055	float64
2056	float64
2057	float64
2058	float64
2059	float64
2060	float64
2061	float64
2062	float64
2063	float64
2064	float64
2065	float64
2066	float64
2067	float64
2068	float64
2069	float64
2070	float64
2071	float64
2072	float64

dtype: object

-----dt26-----

```

dt26 types: loans_personal_01    object
loans_personal_02    object
2022_01_02            int64
2022_02_02            int64
dtype: object

```

```

-----
-----dt27-----
-----

```

```

dt27 types: classification_si_do    object
classification_income    object
2019_00                    object
2019_01                    object
2019_02                    object
2019_03                    object
2019_04                    object
2019_05                    object
dtype: object

```

```

-----
-----dt28-----
-----

```

```

dt28 types: classification    object
2022_00                    object
2022_01                    object
2022_02                    object
2022_03                    object
2022_04                    object
dtype: object

```

```

-----
-----dt29-----
-----

```

```

dt29 types: item    object
2018    float64
2019    float64
2020    float64
2021    float64
2022    float64
dtype: object

```

```

-----
-----dt30-----
-----

```

```

dt30 types: index            int64
id            int64
year          int64
wave          int64

```

region	int64
income	float64
fam_mem_num	int64
gender	int64
birth_year	int64
education_level	int64
marriage	int64
religion	int64
occupation	object
company_size	object
unemployment_reason	object

dtype: object

-----dt31-----

dt31 types: By variant		object
By the structure of population and dependency ratio	object	
2022	float64	
2023	float64	
2024	float64	
2025	float64	
2026	float64	
2027	float64	
2028	float64	
2029	float64	
2030	float64	
2031	float64	
2032	float64	
2033	float64	
2034	float64	
2035	float64	
2036	float64	
2037	float64	
2038	float64	
2039	float64	
2040	float64	
2041	float64	
2042	float64	
2043	float64	
2044	float64	
2045	float64	
2046	float64	
2047	float64	
2048	float64	
2049	float64	
2050	float64	
2051	float64	

2052	float64
2053	float64
2054	float64
2055	float64
2056	float64
2057	float64
2058	float64
2059	float64
2060	float64
2061	float64
2062	float64
2063	float64
2064	float64
2065	float64
2066	float64
2067	float64
2068	float64
2069	float64
2070	float64
2071	float64
2072	float64

dtype: object

-----dt32-----

dt32 types: Date	object
Region	object
Birth	float64
Birth_rate	float64
Death	float64
Death_rate	float64
Divorce	float64
Divorce_rate	float64
Marriage	float64
Marriage_rate	float64
Natural_growth	float64
Natural_growth_rate	float64

dtype: object

-----dt33-----

dt33 types: By administrative divisions(eup, myeon, dong)	object
2022	object
2022.1	object
2022.2	object
2022.3	object

2022.4	object
2022.5	object
2022.6	object
2022.7	object
2022.8	object
2022.9	object
2022.10	object
2022.11	object
2022.12	object
2022.13	object
2022.14	object
2022.15	object
2022.16	object
2022.17	object
2022.18	object
2022.19	object
dtype: object	

-----dt34-----

dt34 types: By administrative divisions(eup, myeon, dong)		object
2010	object	
2010.1	object	
2010.2	object	
2010.3	object	
2010.4	object	
2010.5	object	
2010.6	object	
2010.7	object	
2010.8	object	
2010.9	object	
2010.10	object	
2010.11	object	
2010.12	object	
2010.13	object	
2010.14	object	
2010.15	object	
2010.16	object	
2010.17	object	
2010.18	object	
2010.19	object	
dtype: object		

-----dt35-----

dt35 types: By administrative divisions	object
---	--------

2005	object
2005.1	object
2005.2	object
2005.3	object
2005.4	object
2005.5	object
2005.6	object
2005.7	object
2005.8	object
2005.9	object
2005.10	object
2005.11	object
2005.12	object
2005.13	object
2005.14	object
2005.15	object
2005.16	object
2005.17	object
2005.18	object
2005.19	object
dtype: object	

```
-----
-----dt36-----
-----
```

dt36 types: By administrative divisions		object
2000	object	
2000.1	object	
2000.2	object	
2000.3	object	
2000.4	object	
2000.5	object	
2000.6	object	
2000.7	object	
2000.8	object	
2000.9	object	
2000.10	object	
2000.11	object	
2000.12	object	
2000.13	object	
2000.14	object	
2000.15	object	
2000.16	object	
2000.17	object	
2000.18	object	
2000.19	object	
dtype: object		

```

-----
-----dt37-----
-----
dt37 types: By administrative divisions      object
1995                                         object
1995.1                                     object
1995.2                                     object
1995.3                                     object
1995.4                                     object
1995.5                                     object
1995.6                                     object
1995.7                                     object
1995.8                                     object
1995.9                                     object
1995.10                                    object
1995.11                                    object
1995.12                                    object
1995.13                                    object
1995.14                                    object
1995.15                                    object
1995.16                                    object
dtype: object

-----
-----dt38-----
-----
dt38 types: By administrative divisions      object
1990                                         object
1990.1                                     object
1990.2                                     object
1990.3                                     object
1990.4                                     object
1990.5                                     object
1990.6                                     object
1990.7                                     object
1990.8                                     object
1990.9                                     object
1990.10                                    object
1990.11                                    object
1990.12                                    object
1990.13                                    object
1990.14                                    object
1990.15                                    object
1990.16                                    object
dtype: object

-----
-----dt39-----
-----

```

```

-----
dt39 types: By administrative divisions    object
1985                                     object
1985.1                                 object
1985.2                                 object
1985.3                                 object
1985.4                                 object
1985.5                                 object
1985.6                                 object
1985.7                                 object
1985.8                                 object
1985.9                                 object
1985.10                               object
1985.11                               object
1985.12                               object
1985.13                               object
1985.14                               object
1985.15                               object
1985.16                               object
dtype: object

```

```

-----
-----dt40-----
-----

```

```

dt40 types: By administrative divisions    object
1980                                     object
1980.1                                 object
1980.2                                 object
1980.3                                 object
1980.4                                 object
1980.5                                 object
1980.6                                 object
1980.7                                 object
1980.8                                 object
1980.9                                 object
1980.10                               object
1980.11                               object
1980.12                               object
1980.13                               object
1980.14                               object
1980.15                               object
1980.16                               object
dtype: object

```

```

-----
-----dt41-----
-----

```

```

dt41 types: By administrative divisions    object

```

```

1975                object
1975.1              object
1975.2              object
1975.3              object
1975.4              object
1975.5              object
1975.6              object
1975.7              object
1975.8              object
1975.9              object
1975.10             object
1975.11             object
dtype: object

```

```

-----
-----dt42-----
-----

```

```

dt42 types: By administrative divisions    object
1970                object
1970.1              object
1970.2              object
1970.3              object
1970.4              object
1970.5              object
1970.6              object
1970.7              object
1970.8              object
1970.9              object
1970.10             object
1970.11             object
dtype: object

```

```

-----
-----dt43-----
-----

```

```

dt43 types: index          int64
name                object
gugun               object
dong                object
buildDate           float64
min_sales            float64
max_sales            float64
avg_sales            float64
area                 int64
floor                int64
pricePerArea         float64
dtype: object

```

```

-----dt44-----
dt44 types: Date          object
Region          object
Birth           float64
Birth_rate      float64
Death           float64
Death_rate      float64
Divorce         float64
Divorce_rate    float64
Marriage        float64
Marriage_rate   float64
Natural_growth  float64
Natural_growth_rate float64
dtype: object

-----dt45-----
dt45 types: region      object
type        object
1986_01      object
1986_02      object
1986_03      object
...
2012_08      object
2012_09      object
2012_10      object
2012_11      object
2012_12      object
Length: 326, dtype: object

```

```

[89]: # 04.02.02-01
      # no numeric data to plot
      # dt01

      # dt01_real_estate_pr_01

```

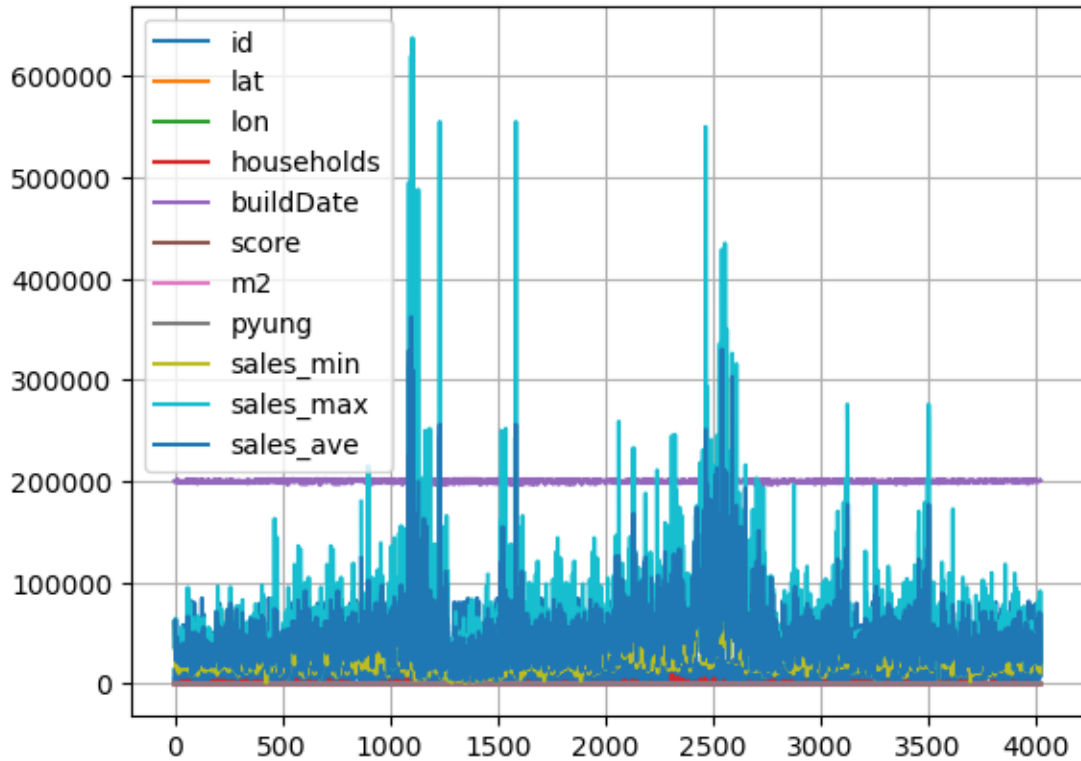
```

[90]: # 04.02.02-02
      # return basic plot
      # rendered basic plot to see vially and to determine if data is numeric.
      # dt02

      dt02_real_estate_se_01.plot()
      plt.box(True)
      plt.grid(True)

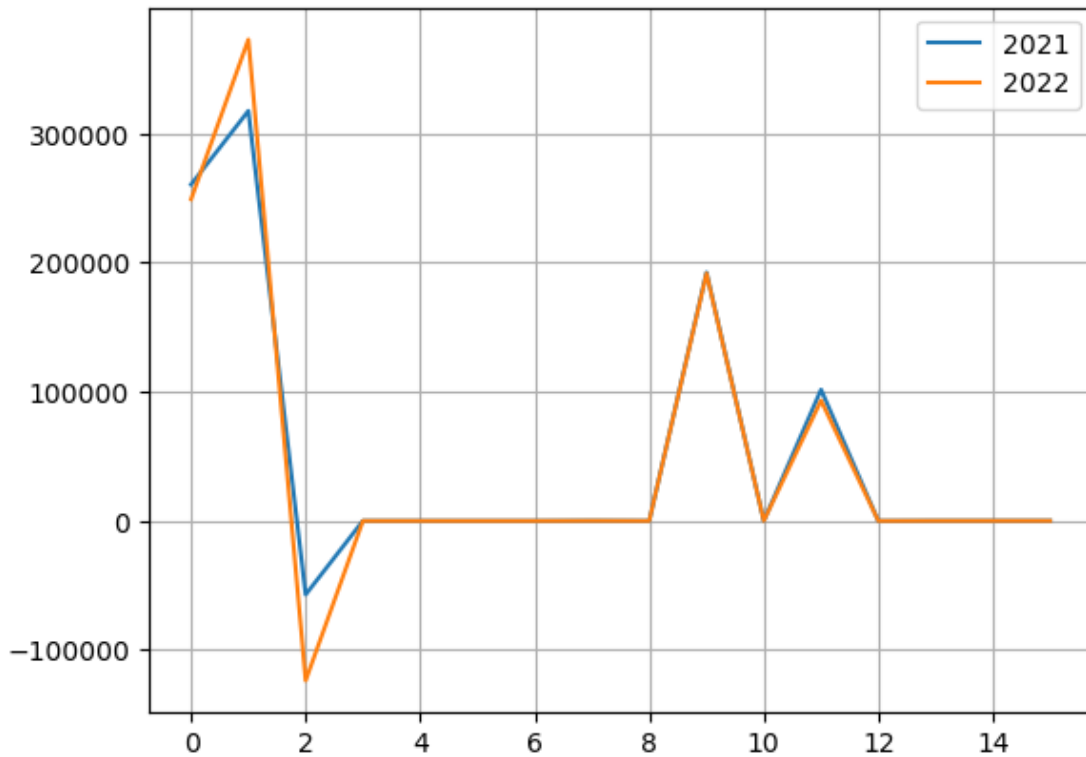
```

```
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[91]: # 04.02.02-03
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt03

dt03_stats_vital_kr_01.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```

```
[92]: # 04.02.02-04
      # no numeric data to plot
      # dt04

      # dt04_housing_census_01
```

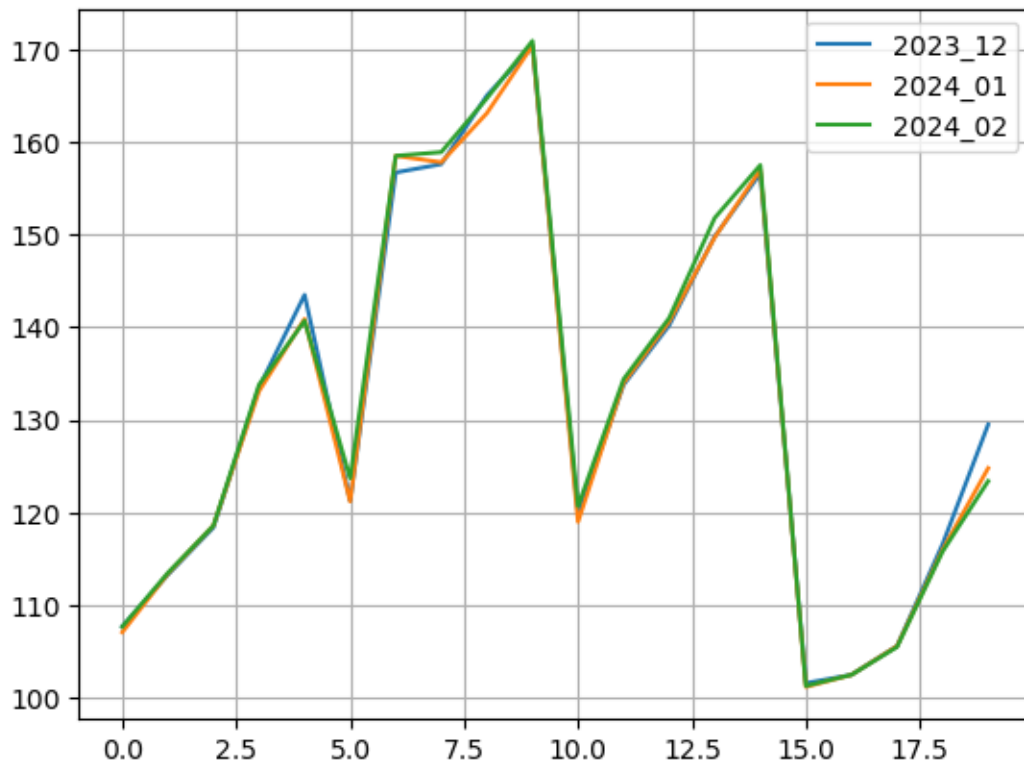
```
[93]: # 04.02.02-05
      # no numeric data to plot
      # dt05

      # dt05_housing_constr_01
```

```
[94]: # 04.02.02-06
      # return basic plot
      # rendered basic plot to see vially and to determine if data is numeric.
      # dt06

      dt06_household_empty_02.plot()
      plt.box(True)
      plt.grid(True)
      plt.title('', fontsize = 16, color = '#0047ab')
      plt.xlabel('', fontsize = 14, color = '#0047ab')
```

```
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[95]: # 04.02.02-07
# no numeric data to plot
# dt07

# dt07_housing_loss__01
```

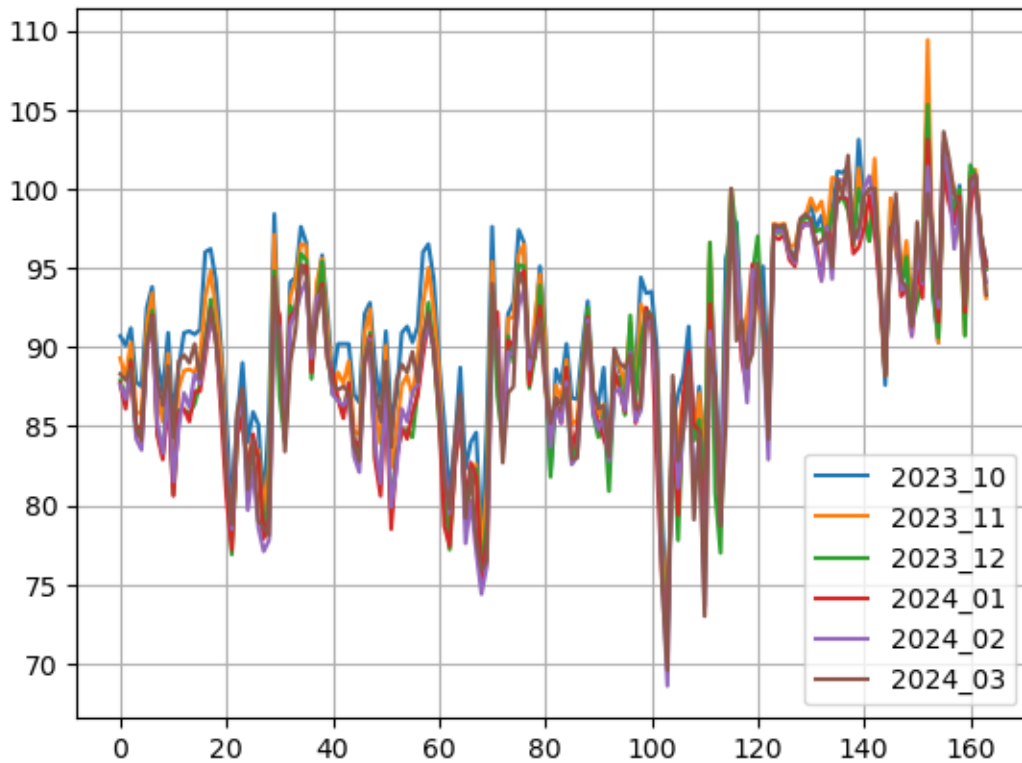
```
[96]: # 04.02.02-08
# no numeric data to plot
# dt08

# dt08_housing_use_sv_01
```

```
[97]: # 04.02.02-09
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt09

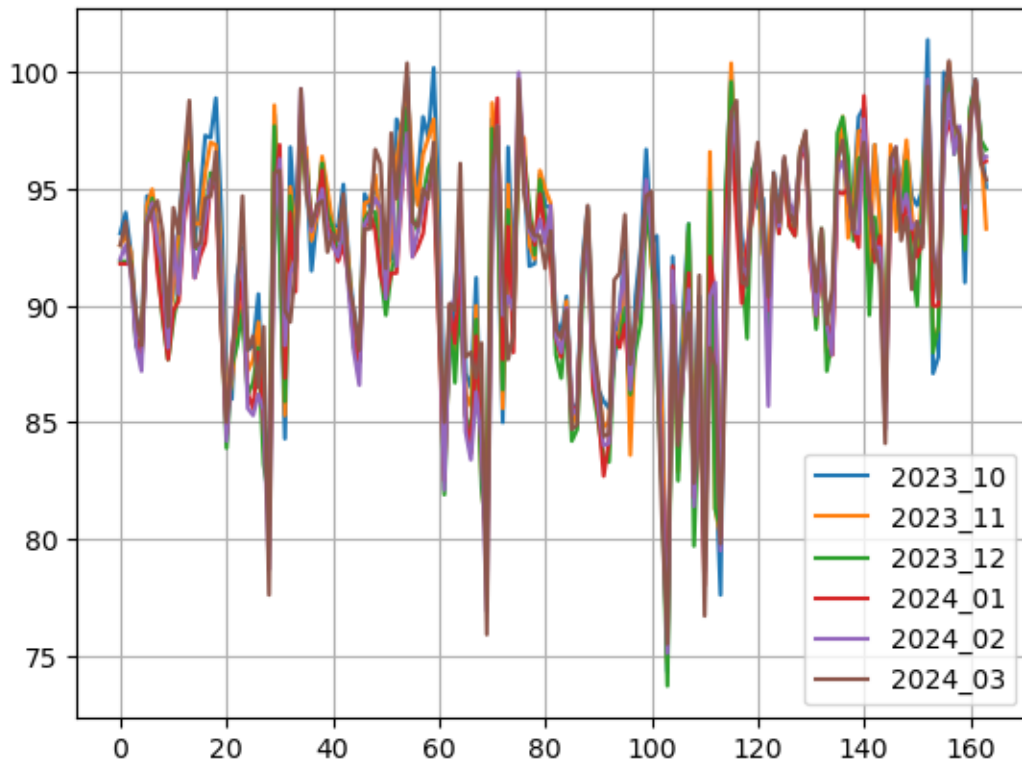
dt09_jeonse_sales__01.plot()
plt.box(True)
```

```
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



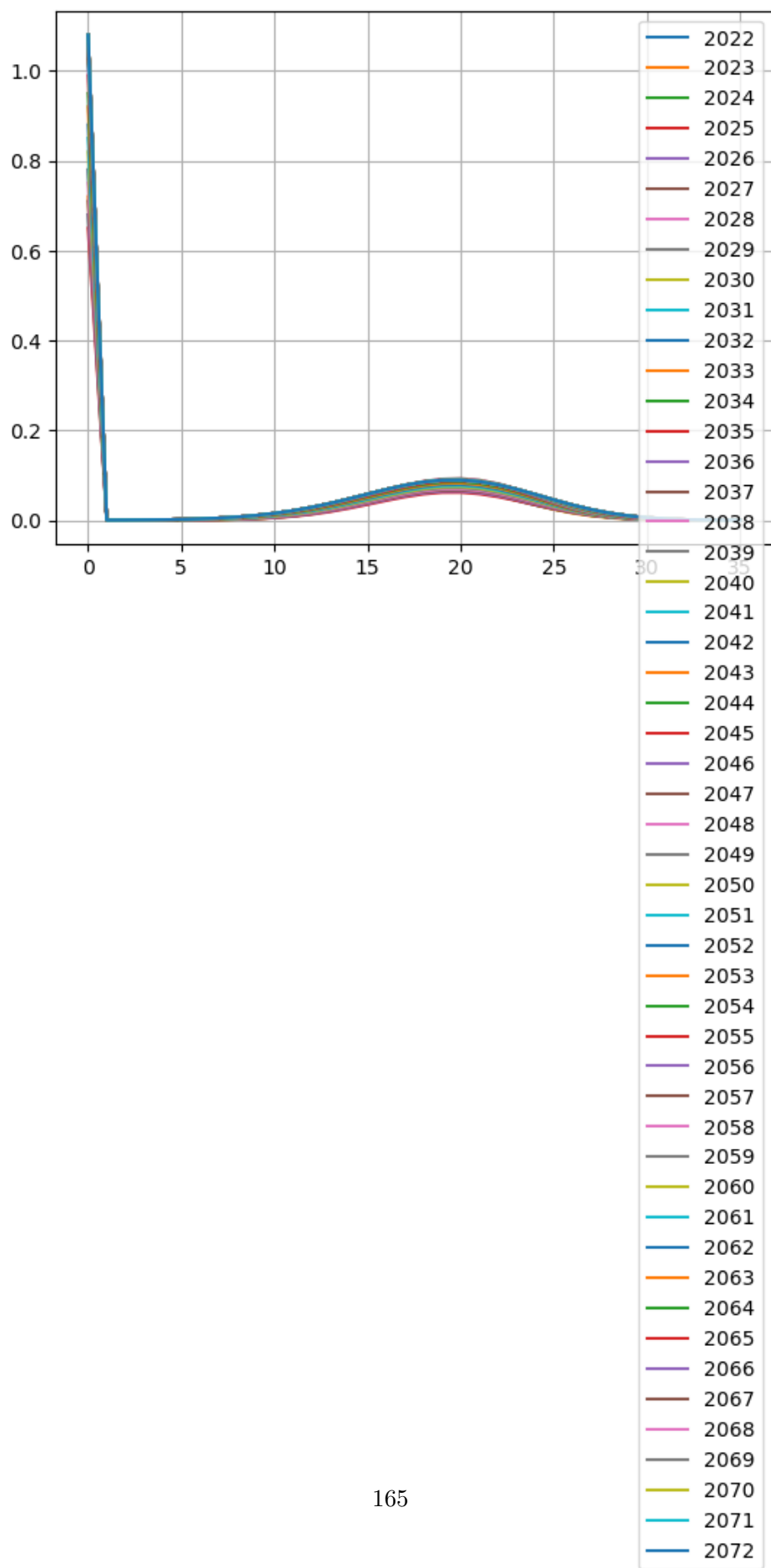
```
[98]: # 04.02.02-10
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt10

dt10_jeonse_market__02.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



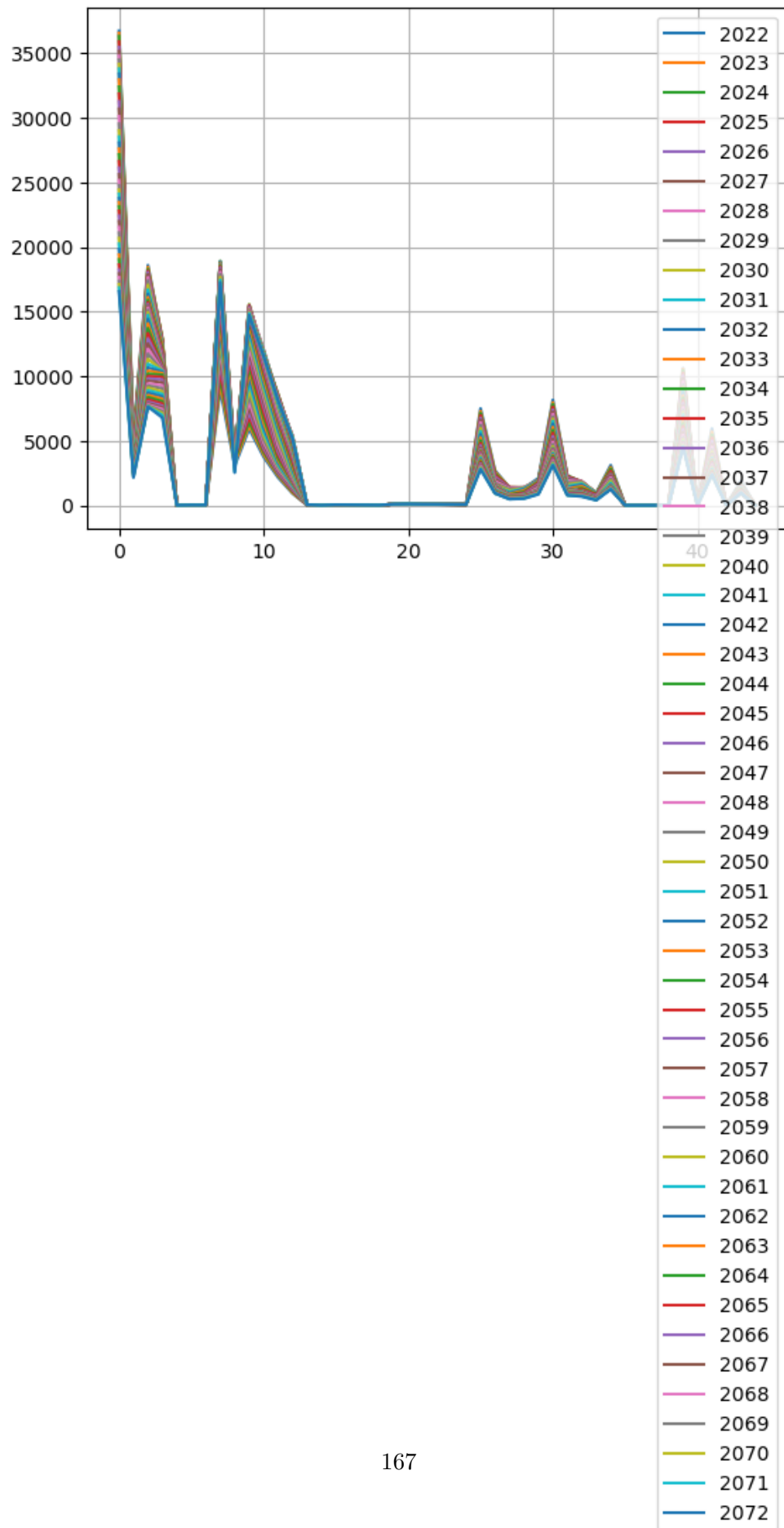
```
[99]: # 04.02.02-11
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt11

dt11_birthrate_age__01.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[100]: # 04.02.02-12
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt12

dt12_birthrate_age__02.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[101]: # 04.02.02-13
# no numeric data to plot
# dt13

# dt13_pop_houshold__01
```

```
[102]: # 04.02.02-14
# no numeric data to plot
# dt14

# dt14_pop_houshold__02
```

```
[103]: # 04.02.02-15
# no numeric data to plot
# dt15

# dt15_pop_houshold__03
```

```
[104]: # 04.02.02-16
# no numeric data to plot
# dt16

# dt16_pop_houshold__04
```

```
[105]: # 04.02.02-17
# no numeric data to plot
# dt17

# dt17_pop_houshold__05
```

```
[106]: # 04.02.02-18
# no numeric data to plot
# dt18

# dt18_pop_houshold__06
```

```
[107]: # 04.02.02-19
# no numeric data to plot
# dt19

# dt19_pop_houshold__07
```

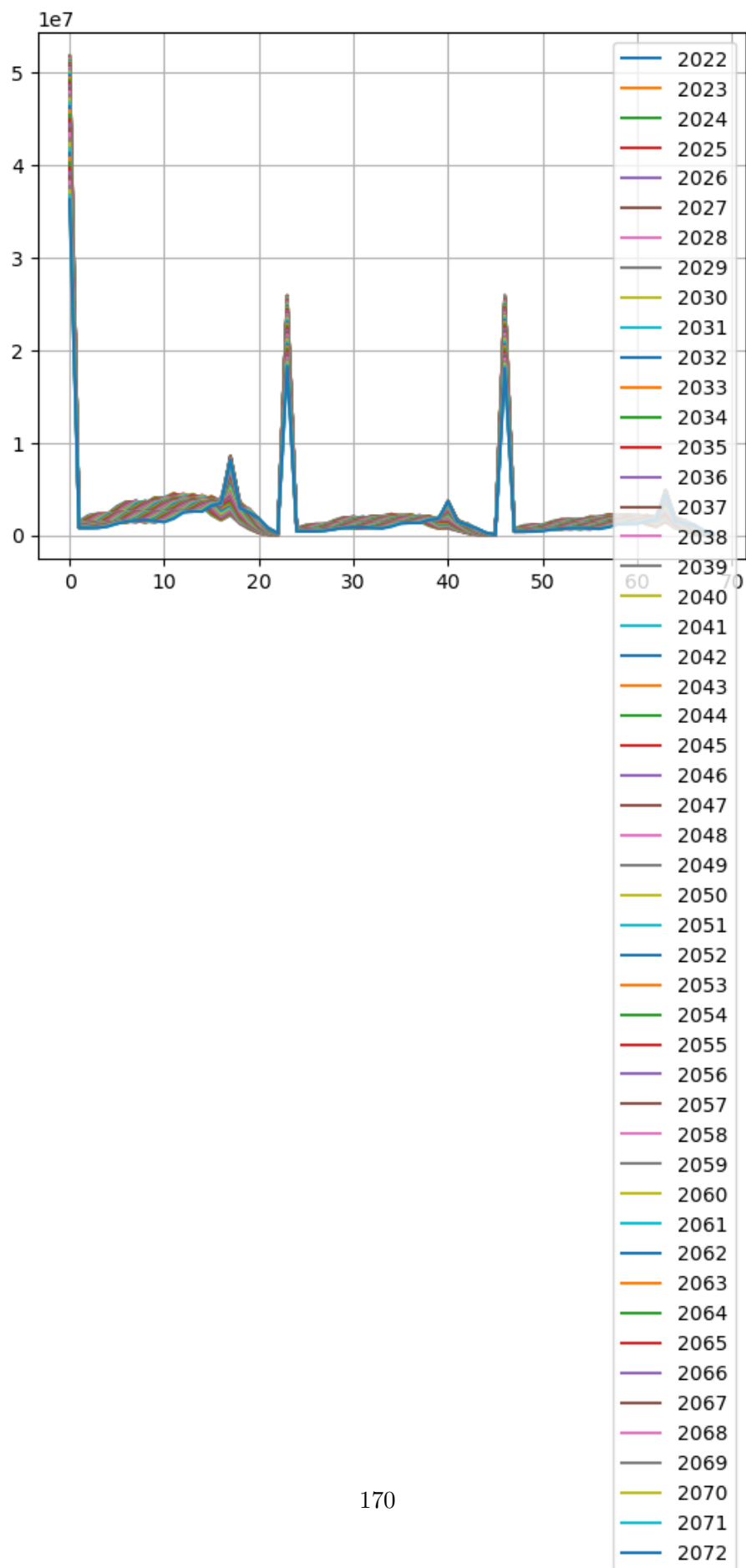
```
[108]: # 04.02.02-20
# no numeric data to plot
# dt20
```



```
# dt20_pop_household__08
```

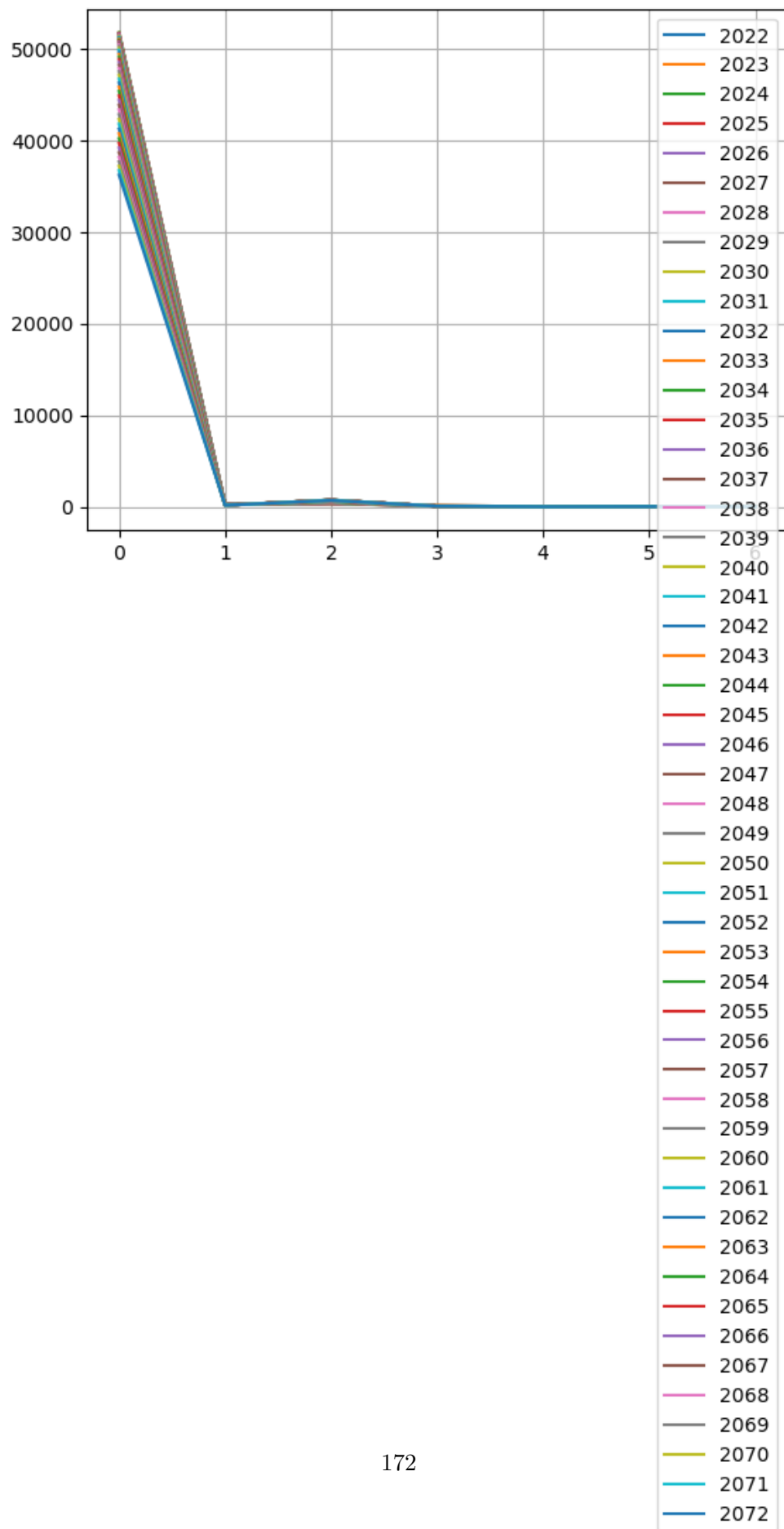
```
[109]: # 04.02.02-21  
# no numeric data to plot  
# dt21  
  
# dt21_pop_household__09
```

```
[110]: # 04.02.02-08  
# return basic plot  
# rendered basic plot to see vially and to determine if data is numeric.  
# dt22  
  
dt22_pop_estimate__01.plot()  
plt.box(True)  
plt.grid(True)  
plt.title('', fontsize = 16, color = '#0047ab')  
plt.xlabel('', fontsize = 14, color = '#0047ab')  
plt.ylabel('', fontsize = 14, color = '#0047ab')  
plt.show()
```



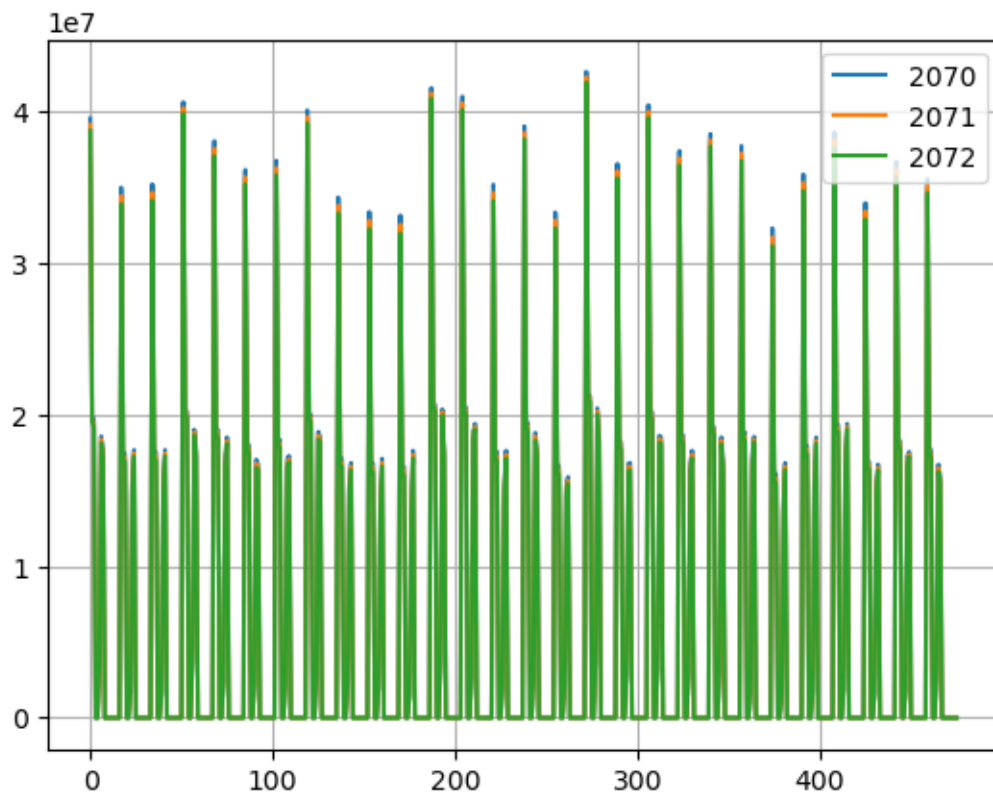
```
[111]: # 04.02.02-09
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt23

dt23_pop_future_pro_01.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



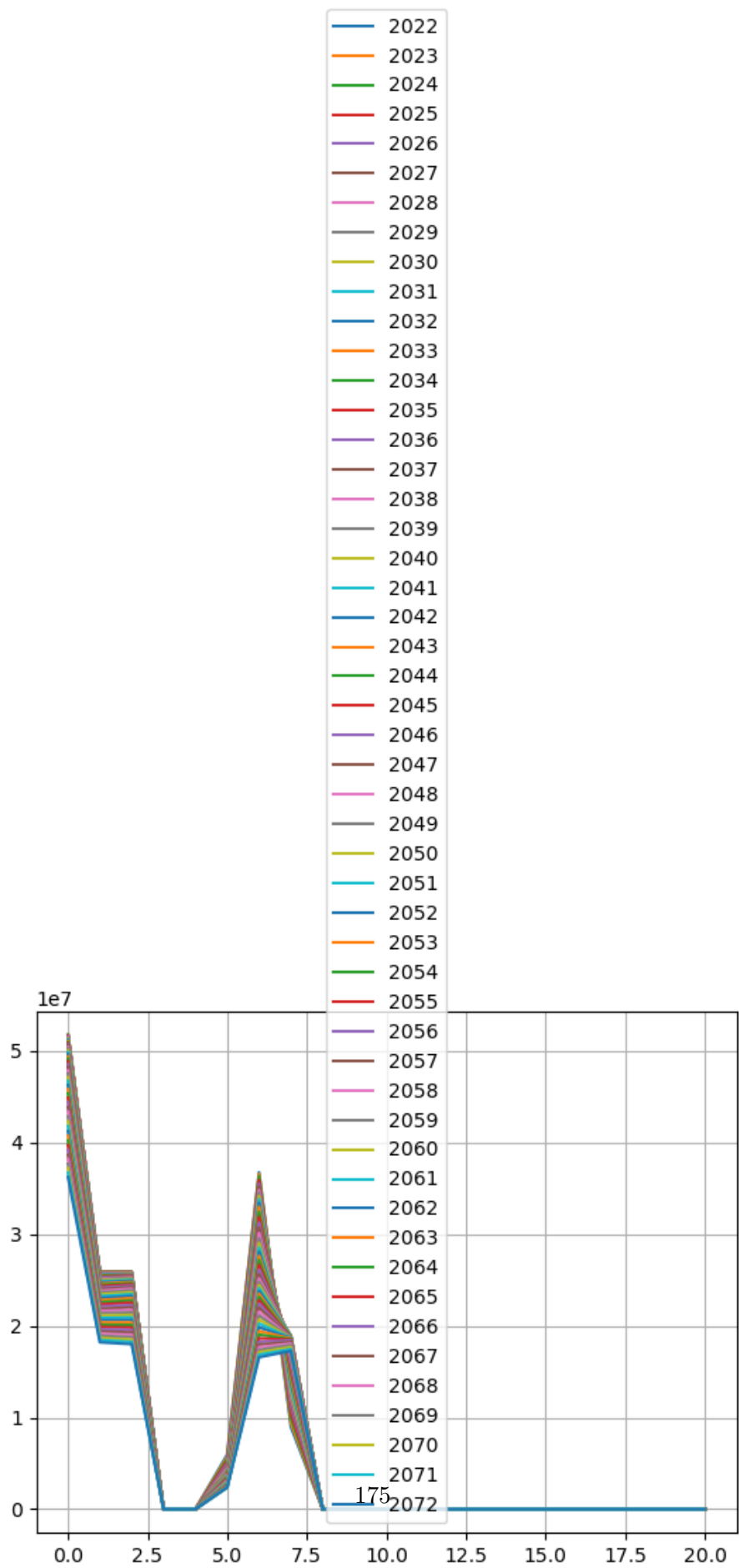
```
[112]: # 04.02.02-10
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt24

dt24_pop_future_pro_02.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



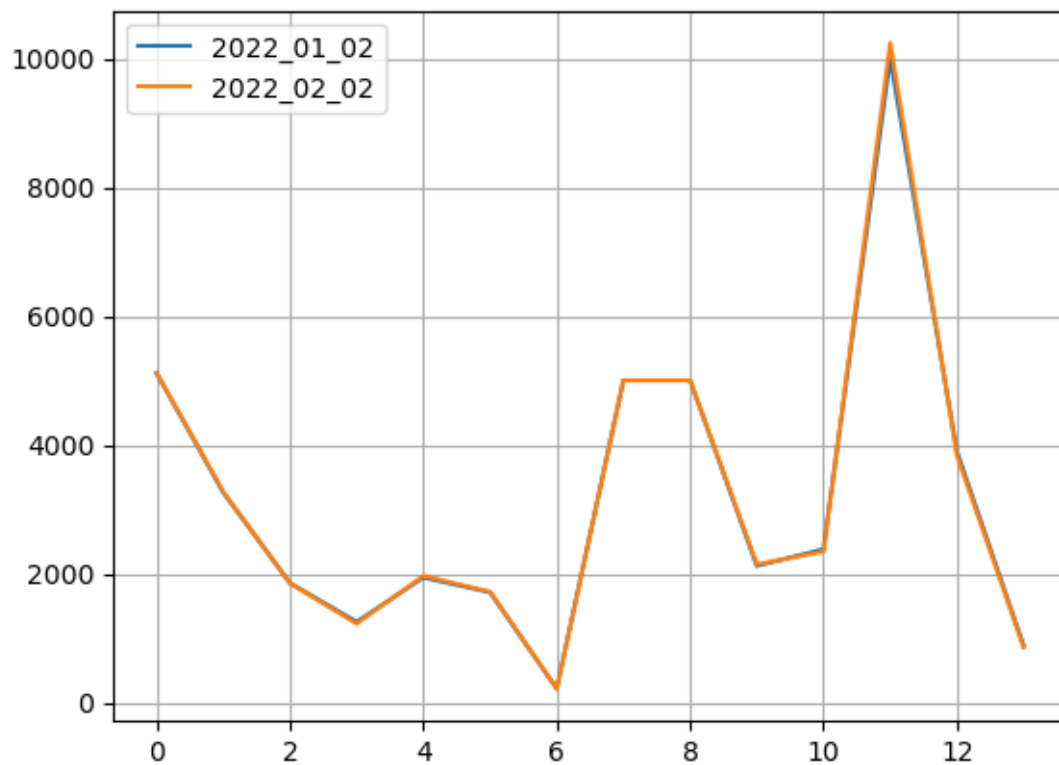
```
[113]: # 04.02.02-11
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt25
```

```
dt25_pop_future_pro_02.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[114]: # 04.02.02-12
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt26

dt26_salaries_entir_01.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[115]: # 04.02.02-27
# no numeric data to plot
# dt27

# dt27_income_job_typ_01
```

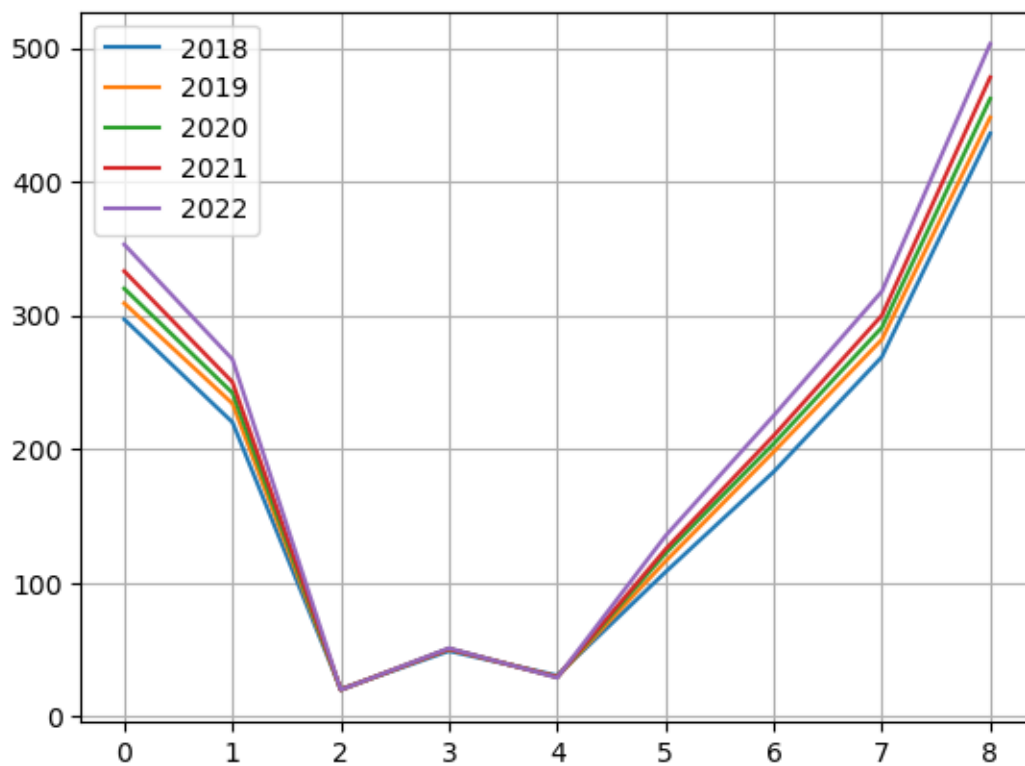


```
[116]: # 04.02.02-28
# no numeric data to plot
# dt28

# dt28_employment_eff_01
```

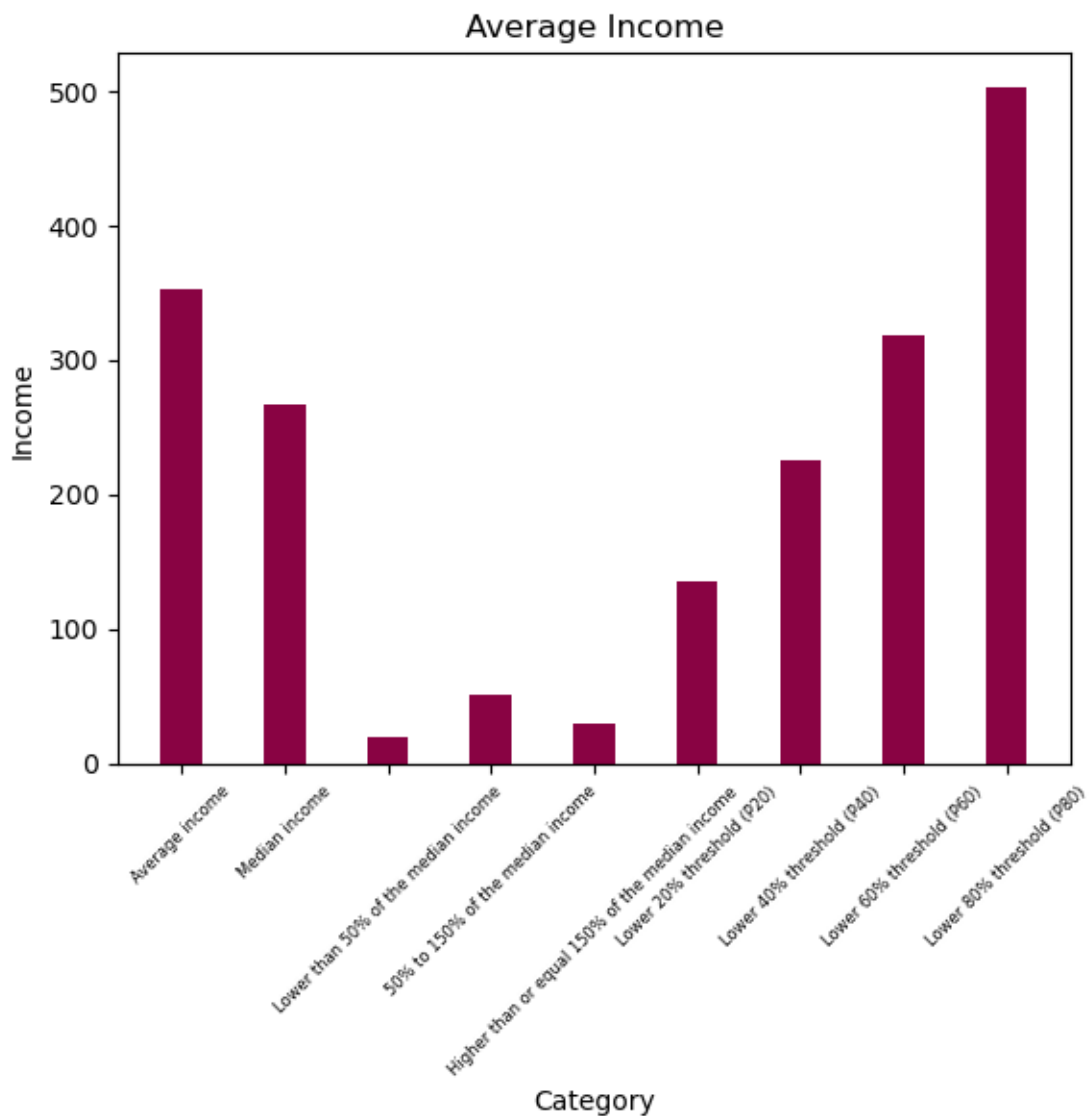
```
[117]: # 04.02.02-13
# return basic plot
# rendered basic plot to see vially and to determine if data is numeric.
# dt29

dt29_income_median__01.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[118]: # 04.02.03-01
# rendered plot for average income
# dt29
```

```
plt.bar(dt29_income_median__01['item'], dt29_income_median__01['2022'], color = '#890343',
        width = 0.4)
plt.xticks(fontsize = 6)
plt.xticks(rotation = 45)
plt.xlabel('Category')
plt.ylabel('Income')
plt.title('Average Income')
plt.show()
```

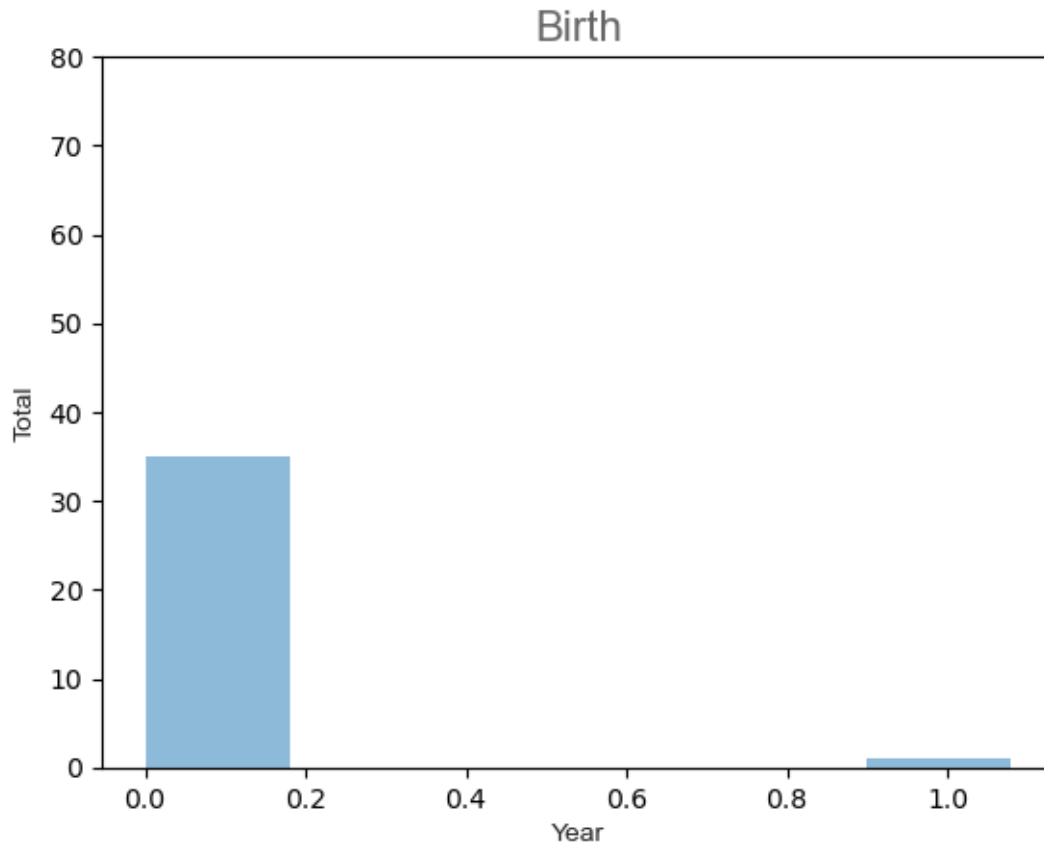


```
[119]: # 04.02.04-01
# select relevant columns
# assign variables
# dt04-dt11

dt04_housing_census_01_tot = dt04_housing_census_01.iloc[1:43]
dt05_housing_constr_01_tot = dt05_housing_constr_01.iloc[2:21]
dt06_household_empty_02_tot = dt06_household_empty_02.iloc[0:4]
dt07_housing_loss__01_tot = dt07_housing_loss__01.iloc[1]
dt08_housing_use_sv_01_tot = dt08_housing_use_sv_01.iloc[1]
dt09_jeonse_sales__01_tot = dt09_jeonse_sales__01.iloc[0]
dt11_birthrate_age__01_tot = dt11_birthrate_age__01.iloc[0:36]
```

```
[120]: # 04.02.05-06
# plot birth rate future
# dt12

plt.hist([dt11_birthrate_age__01_tot['2072']], bins = 6, alpha = 0.5)
font1 = {'family':'arial','color':'#666666','size':16}
font2 = {'family':'arial','color':'#222222','size':10}
plt.title('Birth', loc = 'center', fontdict = font1)
plt.xlabel('Year', fontdict = font2)
plt.ylabel('Total', fontdict = font2)
plt.ylim(0, 80)
plt.xlim()
plt.show()
```



```
[121]: # 04.02.06-01
# select relevant columns
# assign variables
# dt04-dt09, dt11, dt30

print('-dt04_housing_census_01_tot-----')
print(dt04_housing_census_01_tot.head(43))
print('-dt05_housing_constr_01_tot-----')
print(dt05_housing_constr_01_tot.head(21))
print('-dt06_household_empty_02_tot-----')
print(dt06_household_empty_02_tot.head(4))
print('-dt07_housing_loss__01_tot-----')
print(dt07_housing_loss__01_tot.head(7))
print('-dt08_housing_use_sv_01_tot-----')
print(dt08_housing_use_sv_01_tot.head(99))
print('-dt09_jeonse_sales__01_tot-----')
print(dt09_jeonse_sales__01_tot.head(9))
print('-dt11_birthrte_age__01_tot-----')
print(dt11_birthrte_age__01_tot.head(36))
```

```
print('-dt30_ROK_demo_____01_tot-----')
dt30_ROK_demo_____01
```

```
-dt04_housing_census_01_tot-----
admin_division      type_housing  ... 2010_08 2010_09
1  Whole country      Total  ... 249530 933343
2  Whole country      Total  ... 154235 87793
3  Whole country      Total  ... 80619 217854
4  Whole country      Total  ... 11697 217980
5  Whole country      Total  ... 2003 131279
6  Whole country      Total  ... 564 96730
7  Whole country      Total  ... 244 47780
8  Whole country      Total  ... 168 32616
9  Whole country      Total  ... 0 20290
10 Whole country      Total  ... 0 81021
11 Whole country      Detached house  ... 168233 902172
12 Whole country      Detached house  ... 77398 76841
13 Whole country      Detached house  ... 76808 209740
14 Whole country      Detached house  ... 11278 212290
15 Whole country      Detached house  ... 1871 128784
16 Whole country      Detached house  ... 519 95439
17 Whole country      Detached house  ... 214 47118
18 Whole country      Detached house  ... 145 32194
19 Whole country      Detached house  ... 0 19980
20 Whole country      Detached house  ... 0 79786
21 Whole country      Apartment  ... 66619 5141
22 Whole country      Apartment  ... 66413 5129
23 Whole country      Apartment  ... 204 12
24 Whole country      Apartment  ... 2 0
25 Whole country      Row house  ... 5017 898
26 Whole country      Row house  ... 4995 893
27 Whole country      Row house  ... 22 5
28 Whole country      Row house  ... 0 0
29 Whole country      Multiplex house  ... 1749 134
30 Whole country      Multiplex house  ... 1728 132
31 Whole country      Multiplex house  ... 21 2
32 Whole country      Multiplex house  ... 0 0
33 Whole country      Non-residential building in the housing  ... 7912 24998
34 Whole country      Non-residential building in the housing  ... 3701 4798
35 Whole country      Non-residential building in the housing  ... 3564 8095
36 Whole country      Non-residential building in the housing  ... 417 5690
37 Whole country      Non-residential building in the housing  ... 132 2495
38 Whole country      Non-residential building in the housing  ... 45 1291
39 Whole country      Non-residential building in the housing  ... 30 662
40 Whole country      Non-residential building in the housing  ... 23 422
41 Whole country      Non-residential building in the housing  ... 0 310
42 Whole country      Non-residential building in the housing  ... 0 1235
```

[42 rows x 13 columns]

```
-dt05_housing_constr_01_tot-----
      category_01  ... 2024_02_17
2      Public sector  ...      0
3      Sub-total    ...      0
4      Sub-total    ...      0
5      Sub-total    ...      0
6      Local governments  ...      0
7      Local governments  ...      0
8      Local governments  ...      0
9      Local governments  ...      0
10 Land & Housing Corporation(LH)  ...      0
11 Land & Housing Corporation(LH)  ...      0
12 Land & Housing Corporation(LH)  ...      0
13 Land & Housing Corporation(LH)  ...      0
14      Housing companies  ...      0
15      Housing companies  ...      0
16      Housing companies  ...      0
17      Housing companies  ...      0
18      Private sector    ...    255
19      Private sector    ...      0
20      Private sector    ...    255
```

[19 rows x 56 columns]

```
-dt06_household_empt_02_tot-----
      area      size  2023_12  2024_01  2024_02
0 The Whole Country  Extra Small    107.7    107.1    107.7
1 The Whole Country      Small    113.3    113.4    113.5
2 The Whole Country  Mid-small    118.4    118.6    118.6
3 The Whole Country  Mid-large    133.5    133.1    133.8
```

```
-dt07_housing_loss___01_tot-----
area      Whole country
2022_00      96157
2022_01      45609
2022_02      21033
2022_03       4314
2022_04      12130
2022_05      13071
```

Name: 1, dtype: object

```
-dt08_housing_use_sv_01_tot-----
category_01  Total
category_02  Total
2023_12_00    33440
2023_12_01     2076
2023_12_02     2644
2023_12_03    15388
2023_12_04      525
2023_12_05    1355
```

2023_12_06	16
2023_12_07	45
2023_12_08	463
2023_12_09	15
2023_12_10	608
2023_12_11	479
2023_12_12	5653
2023_12_13	74
2023_12_14	1010
2023_12_15	1983
2023_12_16	734
2023_12_17	372
2024_01_00	36762
2024_01_01	4073
2024_01_02	2714
2024_01_03	13097
2024_01_04	676
2024_01_05	3529
2024_01_06	435
2024_01_07	287
2024_01_08	943
2024_01_09	20
2024_01_10	1189
2024_01_11	220
2024_01_12	2516
2024_01_13	1397
2024_01_14	878
2024_01_15	302
2024_01_16	3913
2024_01_17	573
2024_02_00	38729
2024_02_01	3496
2024_02_02	253
2024_02_03	11513
2024_02_04	2528
2024_02_05	1994
2024_02_06	2839
2024_02_07	2370
2024_02_08	115
2024_02_09	1360
2024_02_10	1969
2024_02_11	1597
2024_02_12	2289
2024_02_13	546
2024_02_14	555
2024_02_15	2913
2024_02_16	1547
2024_02_17	845

Name: 1, dtype: object

-dt09_jeonse_sales__01_tot-----

type	Total
region_01	The Whole Country
region_02	Sub Summary
region_03	Sub Summary
2023_10	90.7
2023_11	89.3
2023_12	87.9
2024_01	87.7
2024_02	87.6

Name: 0, dtype: object

-dt11_birthrate_age__01_tot-----

	variant	age	2022	2023	...	2069	2070	2071	2072
0	Medium	TFR	0.78000	0.72000	...	1.08000	1.08000	1.08000	1.08000
1	Medium	15 years	0.00024	0.00001	...	0.00024	0.00024	0.00024	0.00024
2	Medium	16 years	0.00022	0.00002	...	0.00042	0.00042	0.00042	0.00042
3	Medium	17 years	0.00035	0.00004	...	0.00071	0.00071	0.00071	0.00071
4	Medium	18 years	0.00044	0.00010	...	0.00119	0.00119	0.00119	0.00119
5	Medium	19 years	0.00102	0.00021	...	0.00193	0.00193	0.00193	0.00193
6	Medium	20 years	0.00185	0.00046	...	0.00308	0.00308	0.00308	0.00308
7	Medium	21 years	0.00289	0.00092	...	0.00478	0.00478	0.00478	0.00478
8	Medium	22 years	0.00389	0.00178	...	0.00724	0.00724	0.00724	0.00724
9	Medium	23 years	0.00549	0.00328	...	0.01067	0.01067	0.01067	0.01067
10	Medium	24 years	0.00722	0.00572	...	0.01529	0.01529	0.01529	0.01529
11	Medium	25 years	0.01001	0.00948	...	0.02128	0.02128	0.02128	0.02128
12	Medium	26 years	0.01436	0.01490	...	0.02872	0.02872	0.02872	0.02872
13	Medium	27 years	0.02190	0.02218	...	0.03756	0.03756	0.03756	0.03756
14	Medium	28 years	0.03111	0.03124	...	0.04752	0.04752	0.04752	0.04752
15	Medium	29 years	0.04323	0.04158	...	0.05809	0.05809	0.05809	0.05809
16	Medium	30 years	0.05829	0.05223	...	0.06849	0.06849	0.06849	0.06849
17	Medium	31 years	0.07222	0.06183	...	0.07777	0.07777	0.07777	0.07777
18	Medium	32 years	0.08062	0.06891	...	0.08493	0.08493	0.08493	0.08493
19	Medium	33 years	0.07994	0.07221	...	0.08904	0.08904	0.08904	0.08904
20	Medium	34 years	0.07699	0.07104	...	0.08946	0.08946	0.08946	0.08946
21	Medium	35 years	0.06821	0.06554	...	0.08598	0.08598	0.08598	0.08598
22	Medium	36 years	0.05610	0.05661	...	0.07891	0.07891	0.07891	0.07891
23	Medium	37 years	0.04431	0.04572	...	0.06901	0.06901	0.06901	0.06901
24	Medium	38 years	0.03379	0.03448	...	0.05739	0.05739	0.05739	0.05739
25	Medium	39 years	0.02481	0.02424	...	0.04529	0.04529	0.04529	0.04529
26	Medium	40 years	0.01721	0.01586	...	0.03384	0.03384	0.03384	0.03384
27	Medium	41 years	0.01106	0.00965	...	0.02388	0.02388	0.02388	0.02388
28	Medium	42 years	0.00640	0.00545	...	0.01588	0.01588	0.01588	0.01588
29	Medium	43 years	0.00351	0.00285	...	0.00993	0.00993	0.00993	0.00993
30	Medium	44 years	0.00163	0.00138	...	0.00582	0.00582	0.00582	0.00582
31	Medium	45 years	0.00066	0.00062	...	0.00319	0.00319	0.00319	0.00319
32	Medium	46 years	0.00025	0.00025	...	0.00163	0.00163	0.00163	0.00163
33	Medium	47 years	0.00009	0.00010	...	0.00077	0.00077	0.00077	0.00077


```

34 Medium 48 years 0.00003 0.00003 ... 0.00034 0.00034 0.00034 0.00034
35 Medium 49 years 0.00001 0.00001 ... 0.00014 0.00014 0.00014 0.00014

```

[36 rows x 53 columns]

-dt30_ROK_demo_____01_tot-----

```

[121]:      index      id  year  ...  occupation  company_size  unemployment_reason
0         0      10101  2005  ...              8
1         1      10101  2011  ...             10
2         2      10101  2012  ...             10
3         3      10101  2013  ...              1
4         4      10101  2014  ...             10
...      ...      ...  ...  ...      ...
92852  92852  98000701  2014  ...          874              1
92853  92853  98000701  2015  ...          874              1
92854  92854  98000701  2016  ...          874              1
92855  92855  98000701  2017  ...          874              1
92856  92856  98000701  2018  ...          874              1

```

[92857 rows x 15 columns]

```

[122]: # 04.02.04-03
# convert to float
# assign variables
# dt04-dt05

dt04_housing_census_01_tot['2010_00'] = dt04_housing_census_01_tot['2010_00'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_01'] = dt04_housing_census_01_tot['2010_01'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_02'] = dt04_housing_census_01_tot['2010_02'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_03'] = dt04_housing_census_01_tot['2010_03'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_04'] = dt04_housing_census_01_tot['2010_04'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_05'] = dt04_housing_census_01_tot['2010_05'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_06'] = dt04_housing_census_01_tot['2010_06'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_07'] = dt04_housing_census_01_tot['2010_07'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_08'] = dt04_housing_census_01_tot['2010_08'].
    ↳astype(float)
dt04_housing_census_01_tot['2010_09'] = dt04_housing_census_01_tot['2010_09'].
    ↳astype(float)

```

```

dt05_housing_constr_01_tot['2023_12_00'] =_
    ↳dt05_housing_constr_01_tot['2023_12_00'].astype(float)
dt05_housing_constr_01_tot['2023_12_01'] =_
    ↳dt05_housing_constr_01_tot['2023_12_01'].astype(float)
dt05_housing_constr_01_tot['2023_12_02'] =_
    ↳dt05_housing_constr_01_tot['2023_12_02'].astype(float)
dt05_housing_constr_01_tot['2023_12_03'] =_
    ↳dt05_housing_constr_01_tot['2023_12_03'].astype(float)
dt05_housing_constr_01_tot['2023_12_04'] =_
    ↳dt05_housing_constr_01_tot['2023_12_04'].astype(float)
dt05_housing_constr_01_tot['2023_12_05'] =_
    ↳dt05_housing_constr_01_tot['2023_12_05'].astype(float)
dt05_housing_constr_01_tot['2023_12_06'] =_
    ↳dt05_housing_constr_01_tot['2023_12_06'].astype(float)
dt05_housing_constr_01_tot['2023_12_07'] =_
    ↳dt05_housing_constr_01_tot['2023_12_07'].astype(float)
dt05_housing_constr_01_tot['2023_12_08'] =_
    ↳dt05_housing_constr_01_tot['2023_12_08'].astype(float)
dt05_housing_constr_01_tot['2023_12_09'] =_
    ↳dt05_housing_constr_01_tot['2023_12_09'].astype(float)
dt05_housing_constr_01_tot['2023_12_10'] =_
    ↳dt05_housing_constr_01_tot['2023_12_10'].astype(float)
dt05_housing_constr_01_tot['2023_12_11'] =_
    ↳dt05_housing_constr_01_tot['2023_12_11'].astype(float)
dt05_housing_constr_01_tot['2023_12_12'] =_
    ↳dt05_housing_constr_01_tot['2023_12_12'].astype(float)
dt05_housing_constr_01_tot['2023_12_13'] =_
    ↳dt05_housing_constr_01_tot['2023_12_13'].astype(float)
dt05_housing_constr_01_tot['2023_12_14'] =_
    ↳dt05_housing_constr_01_tot['2023_12_14'].astype(float)
dt05_housing_constr_01_tot['2023_12_15'] =_
    ↳dt05_housing_constr_01_tot['2023_12_15'].astype(float)
dt05_housing_constr_01_tot['2023_12_16'] =_
    ↳dt05_housing_constr_01_tot['2023_12_16'].astype(float)
dt05_housing_constr_01_tot['2024_02_17'] =_
    ↳dt05_housing_constr_01_tot['2024_02_17'].astype(float)

```

```

[123]: # 04.02.06-02
# select relevant columns
# assign variables
# dt04

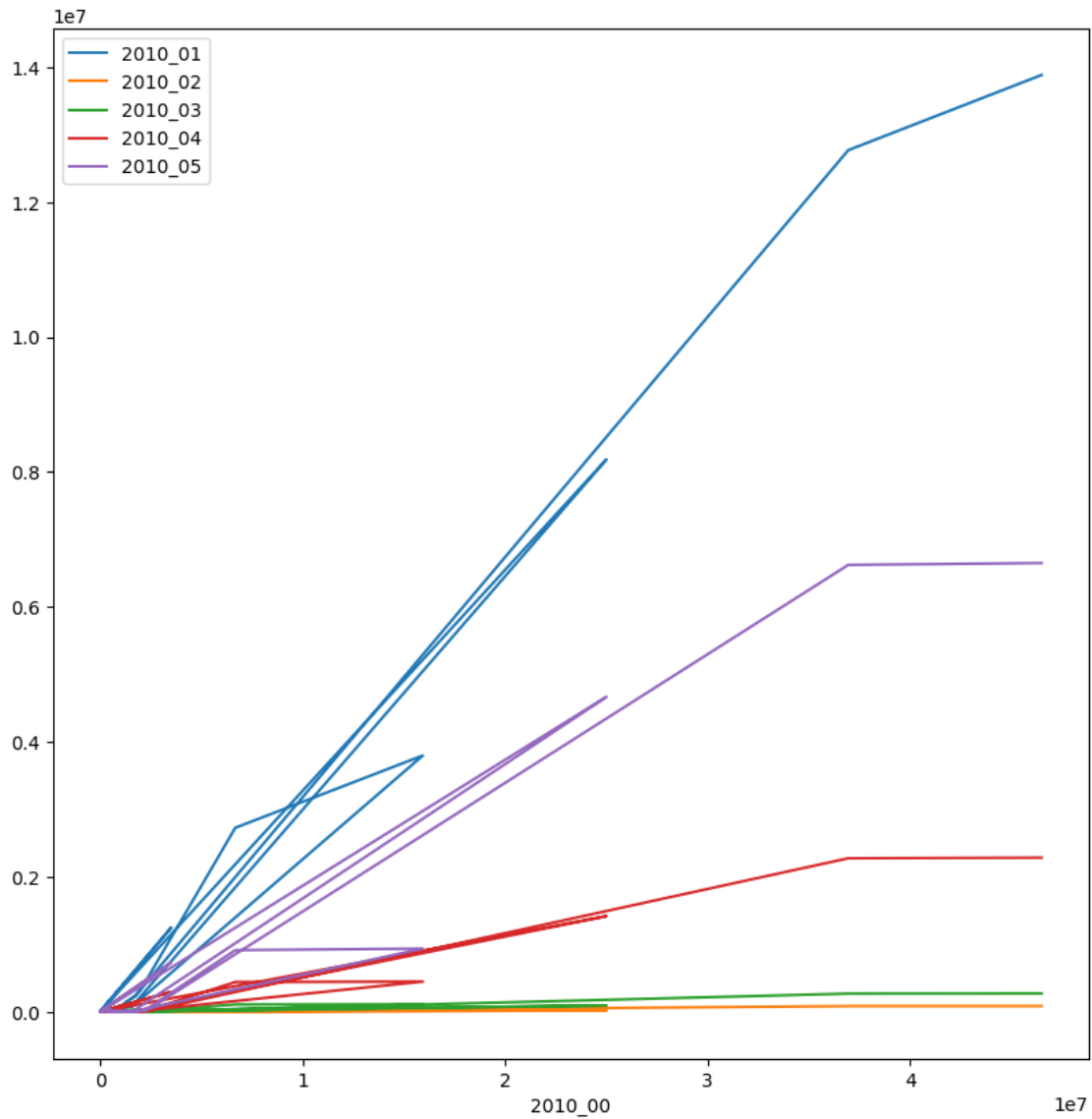
dt04_housing_census_01_tot.plot(x = "2010_00", y = ['2010_01', '2010_02',_
    ↳'2010_03', '2010_04', '2010_05'], kind = "line", figsize = (10, 10))

```

```

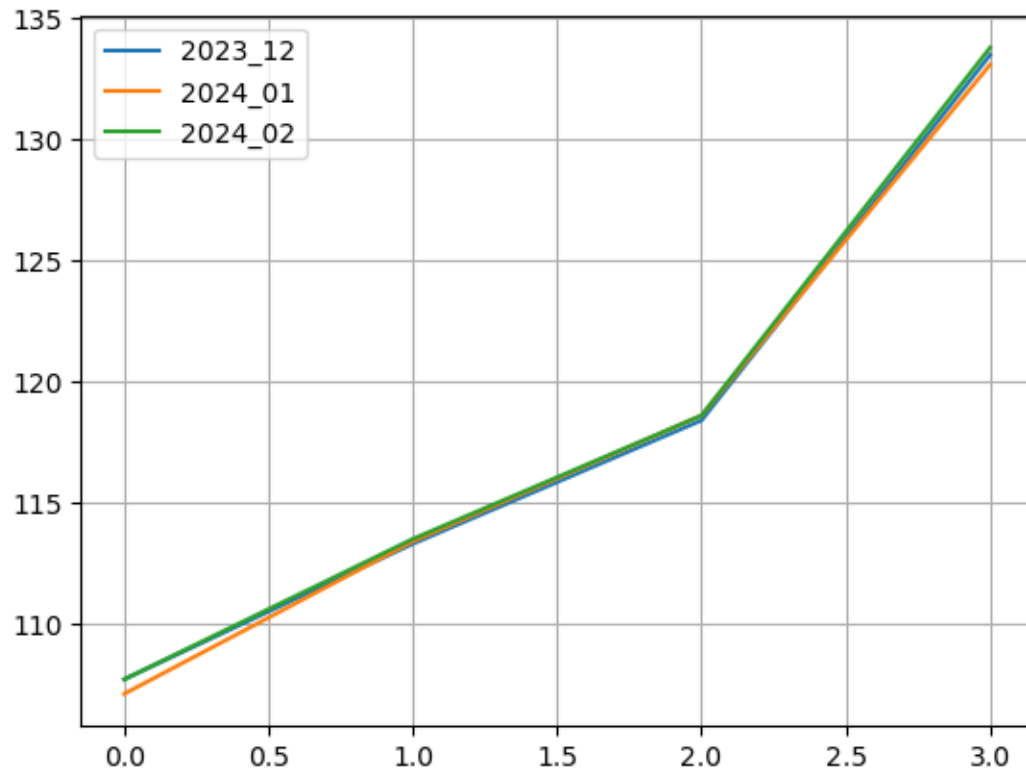
[123]: <Axes: xlabel='2010_00'>

```



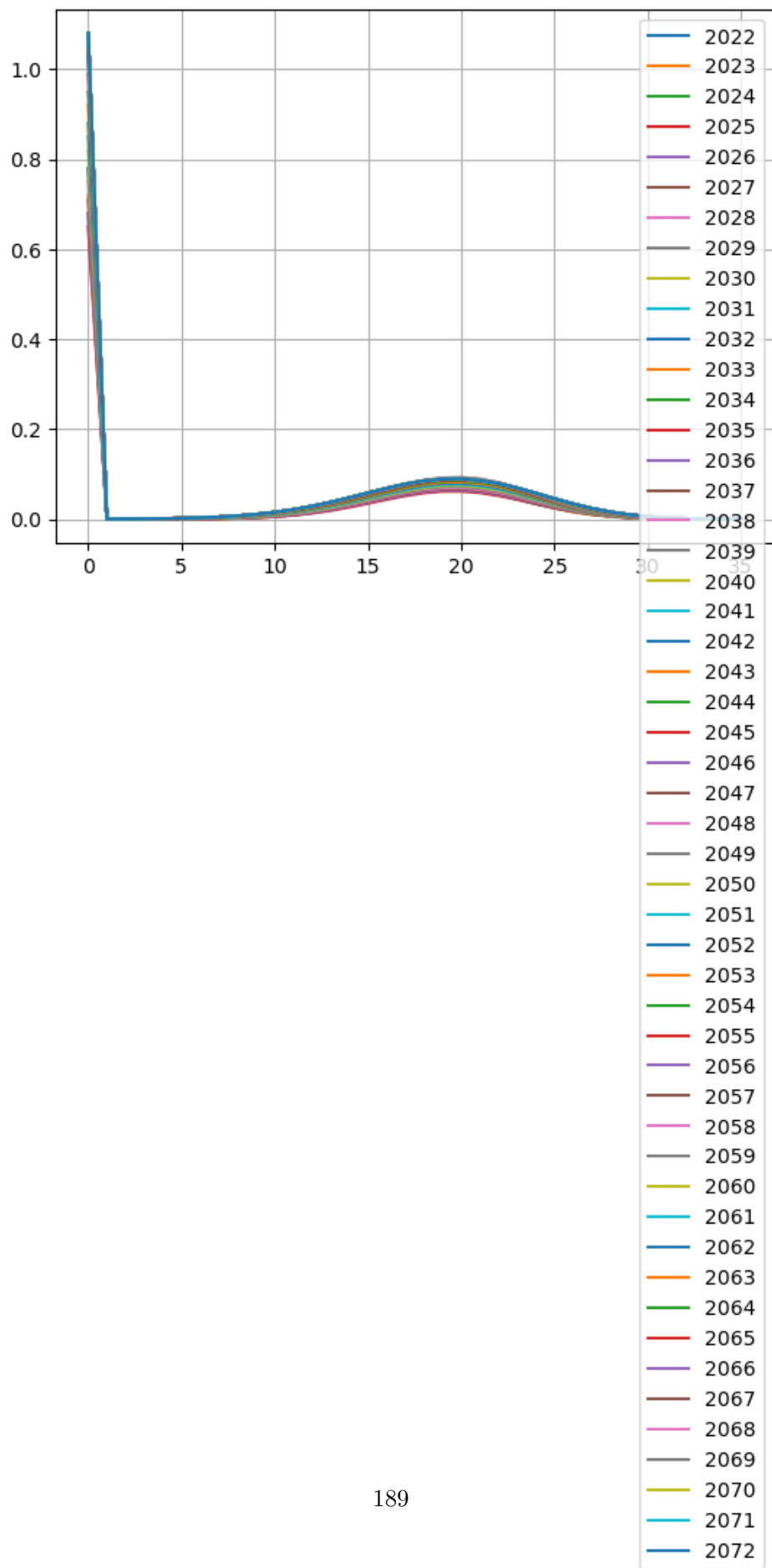
```
[124]: # 04.02.06-03
# return plot of vacant housing
# dt06

dt06_household_empt_02_tot.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



```
[125]: # 04.02.06-04
# return plot of vacant housing
# dt11

dt11_birthrate_age__01_tot.plot()
plt.box(True)
plt.grid(True)
plt.title('', fontsize = 16, color = '#0047ab')
plt.xlabel('', fontsize = 14, color = '#0047ab')
plt.ylabel('', fontsize = 14, color = '#0047ab')
plt.show()
```



6 DSC630_WK12-ms05.02

```
[126]: # 05.02.00-01

## DSC630_term_milestone05_Schreck-Kim.pdf
```

```
[127]: # 05.02.01-01
# convert integer date column to dates in US format
# EDA of marriage rate and income levels in milestone four
# will explore population trends and housing prices in milestone five
# dt32 contains data regarding population trends
# dt32

dt32_ROK_population_01['Date'] = pd.to_datetime(dt32_ROK_population_01['Date'],
↪format = '%m/%d/%Y')
dt32_ROK_population_01.head()
```

```
[127]:
```

	Date	Region	...	Natural_growth	Natural_growth_rate
0	2000-01-01	Busan	...	1877.0	5.8
1	2000-01-01	Chungcheongbuk-do	...	979.0	7.7
2	2000-01-01	Chungcheongnam-do	...	932.0	5.7
3	2000-01-01	Daegu	...	1940.0	9.1
4	2000-01-01	Daejeon	...	1294.0	11.2

[5 rows x 12 columns]

```
[128]: # 05.02.01-02
# add index column as ID
# dt32

dt32_ROK_population_01.insert(0, 'ID', range(0, 0 +
↪len(dt32_ROK_population_01)))
```

```
[129]: # 05.02.01-03
# filter Seoul as the metro area is one-third of entire population
# dt32

dt32_ROK_population_01.set_index("Region", inplace = True)
dt32_ROK_population_01_seoul = dt32_ROK_population_01.loc["Seoul"]
```

```
[130]: # 05.02.01-04
# clean dataset
# renamed original columns for clarity and use in future steps
# dt32
```

```

dt32_ROK_population_01_seoul.rename(columns = {"Region": "region"}, inplace =
↳True)
dt32_ROK_population_01_seoul.rename(columns = {"ID": "id"}, inplace = True)
dt32_ROK_population_01_seoul.rename(columns = {"Date": "date"}, inplace = True)
dt32_ROK_population_01_seoul.rename(columns = {"Birth": "birth"}, inplace =
↳True)
dt32_ROK_population_01_seoul.rename(columns = {"Birth_rate": "birth_rate"},
↳inplace = True)
dt32_ROK_population_01_seoul.rename(columns = {"Death": "death"}, inplace =
↳True)
dt32_ROK_population_01_seoul.rename(columns = {"Death_rate": "death_rate"},
↳inplace = True)
dt32_ROK_population_01_seoul.rename(columns = {"Divorce": "divorce"}, inplace =
↳True)
dt32_ROK_population_01_seoul.rename(columns = {"Divorce_rate": "divorce_rate"},
↳inplace = True)
dt32_ROK_population_01_seoul.rename(columns = {"Marriage": "marriage"}, inplace
↳= True)
dt32_ROK_population_01_seoul.rename(columns = {"Marriage_rate":
↳"marriage_rate"}, inplace = True)
dt32_ROK_population_01_seoul.rename(columns = {"Natural_growth":
↳"natural_growth"}, inplace = True)
dt32_ROK_population_01_seoul.rename(columns = {"Natural_growth_rate":
↳"natural_growth_rate"}, inplace = True)

```

```

[131]: # 05.02.01-05
# dropped columns display ratio of variable per year to population; contains
↳incomplete data
# rates contain inconsistent data
# remove irrelevant and incomplete columns
# dt32

dt32_ROK_population_01_seoul_drp = dt32_ROK_population_01_seoul.
↳drop(['birth_rate', 'death_rate', 'divorce_rate', 'marriage_rate',
↳'natural_growth_rate'], axis = 1)

```

```

[132]: # 05.02.01-06
# return first and last ten rows of dataset with removed columns
# dt32

print(dt32_ROK_population_01_seoul_drp.head(10))
print(dt32_ROK_population_01_seoul_drp.tail(10))

```

	id	date	birth	death	divorce	marriage	natural_growth
Region							
Seoul	15	2000-01-01	12866.0	3931.0	1835.0	6781.0	8935.0

Seoul	33	2000-02-01	11797.0	3218.0	1912.0	5596.0	8579.0
Seoul	51	2000-03-01	12453.0	3368.0	2333.0	6781.0	9085.0
Seoul	69	2000-04-01	10782.0	3204.0	1916.0	6667.0	7578.0
Seoul	87	2000-05-01	10817.0	3064.0	2108.0	8428.0	7753.0
Seoul	105	2000-06-01	9743.0	2959.0	2264.0	7160.0	6784.0
Seoul	123	2000-07-01	9990.0	3077.0	2165.0	5162.0	6913.0
Seoul	141	2000-08-01	10539.0	3215.0	2358.0	4660.0	7324.0
Seoul	159	2000-09-01	11020.0	3181.0	2001.0	3694.0	7839.0
Seoul	177	2000-10-01	11659.0	3472.0	2347.0	6393.0	8187.0
	id	date	birth	death	divorce	marriage	natural_growth
Region							
Seoul	4695	2021-09-01	3753.0	3928.0	1228.0	2698.0	-175.0
Seoul	4713	2021-10-01	3533.0	4171.0	1077.0	2858.0	-638.0
Seoul	4731	2021-11-01	3513.0	4490.0	1209.0	3107.0	-977.0
Seoul	4749	2021-12-01	3034.0	5365.0	1266.0	3754.0	-2331.0
Seoul	4767	2022-01-01	4285.0	4648.0	1058.0	2708.0	-363.0
Seoul	4785	2022-02-01	3452.0	4511.0	1007.0	2979.0	-1059.0
Seoul	4803	2022-03-01	4002.0	6529.0	1084.0	2774.0	-2527.0
Seoul	4821	2022-04-01	3603.0	5351.0	1056.0	2844.0	-1747.0
Seoul	4839	2022-05-01	3372.0	4172.0	1263.0	3259.0	-800.0
Seoul	4857	2022-06-01	3137.0	3631.0	1088.0	2630.0	-494.0

```
[133]: # 05.02.02-01
# renaming three columns for clarity for use in future steps
# dt30 contains data regarding demographics for family members, birth years,
# and unemployment
# dt30

dt30_ROK_demo_01.rename(columns={"family_member": "fam_mem_num"}, inplace=
# True)
dt30_ROK_demo_01.rename(columns={"year_born": "birth_year"}, inplace =
# True)
dt30_ROK_demo_01.rename(columns={"reason_none_worker":
# "unemployment_reason"}, inplace = True)
```

```
[134]: # 05.02.02-02
# return first and last ten rows
# verifying renamed columns
# dt02

print(dt30_ROK_demo_01.head(10))
print(dt30_ROK_demo_01.tail(10))
```

	index	id	year	...	occupation	company_size	unemployment_reason
0	0	10101	2005	...			8
1	1	10101	2011	...			10
2	2	10101	2012	...			10
3	3	10101	2013	...			1

4	4	10101	2014	...			10
5	5	10101	2015	...			10
6	6	10101	2016	...			10
7	7	10101	2017	...			10
8	8	20101	2005	...	421	1	
9	9	20101	2006	...			8

[10 rows x 15 columns]

	index	id	year	...	occupation	company_size	unemployment_reason
92847	92847	97990701	2016	...			10
92848	92848	97990701	2017	...			10
92849	92849	98000701	2011	...	874	1	
92850	92850	98000701	2012	...	874	1	
92851	92851	98000701	2013	...	874	1	
92852	92852	98000701	2014	...	874	1	
92853	92853	98000701	2015	...	874	1	
92854	92854	98000701	2016	...	874	1	
92855	92855	98000701	2017	...	874	1	
92856	92856	98000701	2018	...	874	1	

[10 rows x 15 columns]

```
[135]: # 05.02.03-01
# rename year columns for clarity for use in future steps
# dt33 - dt42

dt33_ROK_population_01.rename(columns = {"2022": "2022_tt1"}, inplace = True)
dt34_ROK_population_01.rename(columns = {"2010": "2010_tt1"}, inplace = True)
dt35_ROK_population_01.rename(columns = {"2005": "2005_tt1"}, inplace = True)
dt36_ROK_population_01.rename(columns = {"2000": "2000_tt1"}, inplace = True)
dt37_ROK_population_01.rename(columns = {"1995": "1995_tt1"}, inplace = True)
dt38_ROK_population_01.rename(columns = {"1990": "1990_tt1"}, inplace = True)
dt39_ROK_population_01.rename(columns = {"1985": "1985_tt1"}, inplace = True)
dt40_ROK_population_01.rename(columns = {"1980": "1980_tt1"}, inplace = True)
dt41_ROK_population_01.rename(columns = {"1975": "1975_tt1"}, inplace = True)
dt42_ROK_population_01.rename(columns = {"1970": "1970_tt1"}, inplace = True)
```

```
[136]: # 05.02.04-01
# assign variables for dt32 for four columns:
# date, birth rate, marriage rate, natural growth rate
# the selected metrics were chosen due to their consistencies with outliers and
↳ relevance of data
# dt32

dt32_ROK_population_01_seoul_drp_01 = dt32_ROK_population_01_seoul_drp['date']
dt32_ROK_population_01_seoul_drp_02 = dt32_ROK_population_01_seoul_drp['birth']
```

```
dt32_ROK_population_01_seoul_drp_03 =
    ↪dt32_ROK_population_01_seoul_drp['marriage']
dt32_ROK_population_01_seoul_drp_04 =
    ↪dt32_ROK_population_01_seoul_drp['natural_growth']
```

```
[137]: # 05.02.05-01
# assign variables for dt30 for three columns:
# year, income, education level
# the selected metrics were chosen due to their consistencies with outliers and
    ↪relevance of data
# dt30
```

```
dt30_ROK_demo_____01_01 = dt30_ROK_demo_____01['year']
dt30_ROK_demo_____01_02 = dt30_ROK_demo_____01['income']
dt30_ROK_demo_____01_03 = dt30_ROK_demo_____01['education_level']
```

```
[138]: # 05.02.06-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt33
```

```
dt33_ROK_population_01.insert(0, 'index', range(0, 0 +
    ↪len(dt33_ROK_population_01)))
dt33_ROK_population_01_id = dt33_ROK_population_01.loc[1]
```

```
[139]: # 05.02.06-02
# change column names for dt33
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 2022 population
# dt33
```

```
dt33_ROK_population_01_id.rename({"2022_ttl": "2022_00"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.1": "2022_01"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.2": "2022_02"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.3": "2022_03"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.4": "2022_04"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.5": "2022_05"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.6": "2022_06"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.7": "2022_07"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.8": "2022_08"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.9": "2022_09"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.10": "2022_10"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.11": "2022_11"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.12": "2022_12"}, inplace = True)
```

```

dt33_ROK_population_01_id.rename({"2022.13": "2022_13"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.14": "2022_14"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.15": "2022_15"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.16": "2022_16"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.17": "2022_17"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.18": "2022_18"}, inplace = True)
dt33_ROK_population_01_id.rename({"2022.19": "2022_19"}, inplace = True)

```

```

[140]: # 05.02.06-03
# assign variables for dt33
# add variable 00 for total
# new variable for each column in dataset
# 2022 population
# dt33

dt33_ROK_population_01_id_00 = dt33_ROK_population_01_id['2022_00']
dt33_ROK_population_01_id_01 = dt33_ROK_population_01_id['2022_01']
dt33_ROK_population_01_id_02 = dt33_ROK_population_01_id['2022_02']
dt33_ROK_population_01_id_03 = dt33_ROK_population_01_id['2022_03']
dt33_ROK_population_01_id_04 = dt33_ROK_population_01_id['2022_04']
dt33_ROK_population_01_id_05 = dt33_ROK_population_01_id['2022_05']
dt33_ROK_population_01_id_06 = dt33_ROK_population_01_id['2022_06']
dt33_ROK_population_01_id_07 = dt33_ROK_population_01_id['2022_07']
dt33_ROK_population_01_id_08 = dt33_ROK_population_01_id['2022_08']
dt33_ROK_population_01_id_09 = dt33_ROK_population_01_id['2022_09']
dt33_ROK_population_01_id_10 = dt33_ROK_population_01_id['2022_10']
dt33_ROK_population_01_id_11 = dt33_ROK_population_01_id['2022_11']
dt33_ROK_population_01_id_12 = dt33_ROK_population_01_id['2022_12']
dt33_ROK_population_01_id_13 = dt33_ROK_population_01_id['2022_13']
dt33_ROK_population_01_id_14 = dt33_ROK_population_01_id['2022_14']
dt33_ROK_population_01_id_15 = dt33_ROK_population_01_id['2022_15']
dt33_ROK_population_01_id_16 = dt33_ROK_population_01_id['2022_16']
dt33_ROK_population_01_id_17 = dt33_ROK_population_01_id['2022_17']
dt33_ROK_population_01_id_18 = dt33_ROK_population_01_id['2022_18']
dt33_ROK_population_01_id_19 = dt33_ROK_population_01_id['2022_19']

```

```

[141]: # 05.02.07-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt34

dt34_ROK_population_01.insert(0, 'index', range(0, 0 + len(dt34_ROK_population_01)))
dt34_ROK_population_01_id = dt34_ROK_population_01.loc[1]

```

```
[142]: # 05.02.07-02
# change column names for dt34
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 2010 population
# dt34

dt34_ROK_population_01_id.rename({"2010_tt1": "2010_00"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.1": "2010_01"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.2": "2010_02"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.3": "2010_03"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.4": "2010_04"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.5": "2010_05"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.6": "2010_06"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.7": "2010_07"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.8": "2010_08"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.9": "2010_09"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.10": "2010_10"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.11": "2010_11"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.12": "2010_12"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.13": "2010_13"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.14": "2010_14"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.15": "2010_15"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.16": "2010_16"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.17": "2010_17"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.18": "2010_18"}, inplace = True)
dt34_ROK_population_01_id.rename({"2010.19": "2010_19"}, inplace = True)
```

```
[143]: # 05.02.07-03
# assign variables for dt34
# add variable 00 for total
# new variable for each column in dataset
# 2010 population
# dt34

dt34_ROK_population_01_id_00 = dt34_ROK_population_01_id['2010_00']
dt34_ROK_population_01_id_01 = dt34_ROK_population_01_id['2010_01']
dt34_ROK_population_01_id_02 = dt34_ROK_population_01_id['2010_02']
dt34_ROK_population_01_id_03 = dt34_ROK_population_01_id['2010_03']
dt34_ROK_population_01_id_04 = dt34_ROK_population_01_id['2010_04']
dt34_ROK_population_01_id_05 = dt34_ROK_population_01_id['2010_05']
dt34_ROK_population_01_id_06 = dt34_ROK_population_01_id['2010_06']
dt34_ROK_population_01_id_07 = dt34_ROK_population_01_id['2010_07']
dt34_ROK_population_01_id_08 = dt34_ROK_population_01_id['2010_08']
dt34_ROK_population_01_id_09 = dt34_ROK_population_01_id['2010_09']
dt34_ROK_population_01_id_10 = dt34_ROK_population_01_id['2010_10']
```

```

dt34_ROK_population_01_id_11 = dt34_ROK_population_01_id['2010_11']
dt34_ROK_population_01_id_12 = dt34_ROK_population_01_id['2010_12']
dt34_ROK_population_01_id_13 = dt34_ROK_population_01_id['2010_13']
dt34_ROK_population_01_id_14 = dt34_ROK_population_01_id['2010_14']
dt34_ROK_population_01_id_15 = dt34_ROK_population_01_id['2010_15']
dt34_ROK_population_01_id_16 = dt34_ROK_population_01_id['2010_16']
dt34_ROK_population_01_id_17 = dt34_ROK_population_01_id['2010_17']
dt34_ROK_population_01_id_18 = dt34_ROK_population_01_id['2010_18']
dt34_ROK_population_01_id_19 = dt34_ROK_population_01_id['2010_19']

```

```

[144]: # 05.02.08-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt35

dt35_ROK_population_01.insert(0, 'index', range(0, 0 +
↪len(dt35_ROK_population_01)))
dt35_ROK_population_01_id = dt35_ROK_population_01.loc[1]

```

```

[145]: # 05.02.08-02
# change column names for dt35
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 2005 population
# dt35

dt35_ROK_population_01_id.rename({"2005_tt1": "2005_00"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.1": "2005_01"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.2": "2005_02"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.3": "2005_03"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.4": "2005_04"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.5": "2005_05"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.6": "2005_06"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.7": "2005_07"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.8": "2005_08"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.9": "2005_09"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.10": "2005_10"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.11": "2005_11"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.12": "2005_12"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.13": "2005_13"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.14": "2005_14"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.15": "2005_15"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.16": "2005_16"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.17": "2005_17"}, inplace = True)
dt35_ROK_population_01_id.rename({"2005.18": "2005_18"}, inplace = True)

```

```
dt35_ROK_population_01_id.rename({"2005.19": "2005_19"}, inplace = True)
```

```
[146]: # 05.02.08-03
# assign variables for dt35
# add variable 00 for total
# new variable for each column in dataset
# 2005 population
# dt35

dt35_ROK_population_01_id_00 = dt35_ROK_population_01_id['2005_00']
dt35_ROK_population_01_id_01 = dt35_ROK_population_01_id['2005_01']
dt35_ROK_population_01_id_02 = dt35_ROK_population_01_id['2005_02']
dt35_ROK_population_01_id_03 = dt35_ROK_population_01_id['2005_03']
dt35_ROK_population_01_id_04 = dt35_ROK_population_01_id['2005_04']
dt35_ROK_population_01_id_05 = dt35_ROK_population_01_id['2005_05']
dt35_ROK_population_01_id_06 = dt35_ROK_population_01_id['2005_06']
dt35_ROK_population_01_id_07 = dt35_ROK_population_01_id['2005_07']
dt35_ROK_population_01_id_08 = dt35_ROK_population_01_id['2005_08']
dt35_ROK_population_01_id_09 = dt35_ROK_population_01_id['2005_09']
dt35_ROK_population_01_id_10 = dt35_ROK_population_01_id['2005_10']
dt35_ROK_population_01_id_11 = dt35_ROK_population_01_id['2005_11']
dt35_ROK_population_01_id_12 = dt35_ROK_population_01_id['2005_12']
dt35_ROK_population_01_id_13 = dt35_ROK_population_01_id['2005_13']
dt35_ROK_population_01_id_14 = dt35_ROK_population_01_id['2005_14']
dt35_ROK_population_01_id_15 = dt35_ROK_population_01_id['2005_15']
dt35_ROK_population_01_id_16 = dt35_ROK_population_01_id['2005_16']
dt35_ROK_population_01_id_17 = dt35_ROK_population_01_id['2005_17']
dt35_ROK_population_01_id_18 = dt35_ROK_population_01_id['2005_18']
dt35_ROK_population_01_id_19 = dt35_ROK_population_01_id['2005_19']
```

```
[147]: # 05.02.09-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt36

dt36_ROK_population_01.insert(0, 'index', range(0, 0 + len(dt36_ROK_population_01)))
dt36_ROK_population_01_id = dt36_ROK_population_01.loc[1]
```

```
[148]: # 05.02.09-02
# change column names for dt36
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 2000 population
# dt36
```



```

dt36_ROK_population_01_id.rename({"2000_tt1": "2000_00"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.1": "2000_01"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.2": "2000_02"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.3": "2000_03"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.4": "2000_04"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.5": "2000_05"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.6": "2000_06"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.7": "2000_07"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.8": "2000_08"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.9": "2000_09"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.10": "2000_10"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.11": "2000_11"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.12": "2000_12"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.13": "2000_13"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.14": "2000_14"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.15": "2000_15"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.16": "2000_16"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.17": "2000_17"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.18": "2000_18"}, inplace = True)
dt36_ROK_population_01_id.rename({"2000.19": "2000_19"}, inplace = True)

```

```

[149]: # 05.02.09-03
# assign variables for dt36
# add variable 00 for total
# new variable for each column in dataset
# 2000 population
# dt36

```

```

dt36_ROK_population_01_id_00 = dt36_ROK_population_01_id['2000_00']
dt36_ROK_population_01_id_01 = dt36_ROK_population_01_id['2000_01']
dt36_ROK_population_01_id_02 = dt36_ROK_population_01_id['2000_02']
dt36_ROK_population_01_id_03 = dt36_ROK_population_01_id['2000_03']
dt36_ROK_population_01_id_04 = dt36_ROK_population_01_id['2000_04']
dt36_ROK_population_01_id_05 = dt36_ROK_population_01_id['2000_05']
dt36_ROK_population_01_id_06 = dt36_ROK_population_01_id['2000_06']
dt36_ROK_population_01_id_07 = dt36_ROK_population_01_id['2000_07']
dt36_ROK_population_01_id_08 = dt36_ROK_population_01_id['2000_08']
dt36_ROK_population_01_id_09 = dt36_ROK_population_01_id['2000_09']
dt36_ROK_population_01_id_10 = dt36_ROK_population_01_id['2000_10']
dt36_ROK_population_01_id_11 = dt36_ROK_population_01_id['2000_11']
dt36_ROK_population_01_id_12 = dt36_ROK_population_01_id['2000_12']
dt36_ROK_population_01_id_13 = dt36_ROK_population_01_id['2000_13']
dt36_ROK_population_01_id_14 = dt36_ROK_population_01_id['2000_14']
dt36_ROK_population_01_id_15 = dt36_ROK_population_01_id['2000_15']
dt36_ROK_population_01_id_16 = dt36_ROK_population_01_id['2000_16']
dt36_ROK_population_01_id_17 = dt36_ROK_population_01_id['2000_17']

```

```
dt36_ROK_population_01_id_18 = dt36_ROK_population_01_id['2000_18']
dt36_ROK_population_01_id_19 = dt36_ROK_population_01_id['2000_19']
```

```
[150]: # 05.02.10-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt37

dt37_ROK_population_01.insert(0, 'index', range(0, 0 +
↳ len(dt37_ROK_population_01)))
dt37_ROK_population_01_id = dt37_ROK_population_01.loc[1]
```

```
[151]: # 05.02.10-02
# change column names for dt37
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 1995 population
# dt37

dt37_ROK_population_01_id.rename({"1995_tt1": "1995_00"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.1": "1995_01"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.2": "1995_02"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.3": "1995_03"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.4": "1995_04"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.5": "1995_05"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.6": "1995_06"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.7": "1995_07"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.8": "1995_08"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.9": "1995_09"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.10": "1995_10"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.11": "1995_11"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.12": "1995_12"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.13": "1995_13"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.14": "1995_14"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.15": "1995_15"}, inplace = True)
dt37_ROK_population_01_id.rename({"1995.16": "1995_16"}, inplace = True)
```

```
[152]: # 05.02.10-03
# assign variables for dt37
# add variable 00 for total
# new variable for each column in dataset
# 1995 population
# dt37

dt37_ROK_population_01_id_00 = dt37_ROK_population_01_id['1995_00']
```



```

dt37_ROK_population_01_id_01 = dt37_ROK_population_01_id['1995_01']
dt37_ROK_population_01_id_02 = dt37_ROK_population_01_id['1995_02']
dt37_ROK_population_01_id_03 = dt37_ROK_population_01_id['1995_03']
dt37_ROK_population_01_id_04 = dt37_ROK_population_01_id['1995_04']
dt37_ROK_population_01_id_05 = dt37_ROK_population_01_id['1995_05']
dt37_ROK_population_01_id_06 = dt37_ROK_population_01_id['1995_06']
dt37_ROK_population_01_id_07 = dt37_ROK_population_01_id['1995_07']
dt37_ROK_population_01_id_08 = dt37_ROK_population_01_id['1995_08']
dt37_ROK_population_01_id_09 = dt37_ROK_population_01_id['1995_09']
dt37_ROK_population_01_id_10 = dt37_ROK_population_01_id['1995_10']
dt37_ROK_population_01_id_11 = dt37_ROK_population_01_id['1995_11']
dt37_ROK_population_01_id_12 = dt37_ROK_population_01_id['1995_12']
dt37_ROK_population_01_id_13 = dt37_ROK_population_01_id['1995_13']
dt37_ROK_population_01_id_14 = dt37_ROK_population_01_id['1995_14']
dt37_ROK_population_01_id_15 = dt37_ROK_population_01_id['1995_15']
dt37_ROK_population_01_id_16 = dt37_ROK_population_01_id['1995_16']

```

```

[153]: # 05.02.11-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt38

dt38_ROK_population_01.insert(0, 'index', range(0, 0 +
↪len(dt38_ROK_population_01)))
dt38_ROK_population_01_id = dt38_ROK_population_01.loc[1]

```

```

[154]: # 05.02.11-02
# change column names for dt38
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 1990 population
# dt38

dt38_ROK_population_01_id.rename({"1990_tt1": "1990_00"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.1": "1990_01"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.2": "1990_02"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.3": "1990_03"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.4": "1990_04"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.5": "1990_05"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.6": "1990_06"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.7": "1990_07"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.8": "1990_08"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.9": "1990_09"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.10": "1990_10"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.11": "1990_11"}, inplace = True)

```

```

dt38_ROK_population_01_id.rename({"1990.12": "1990_12"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.13": "1990_13"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.14": "1990_14"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.15": "1990_15"}, inplace = True)
dt38_ROK_population_01_id.rename({"1990.16": "1990_16"}, inplace = True)

```

```

[155]: # 05.02.11-03
# assign variables for dt38
# add variable 00 for total
# new variable for each column in dataset
# 1990 population
# dt38

dt38_ROK_population_01_id_00 = dt38_ROK_population_01_id['1990_00']
dt38_ROK_population_01_id_01 = dt38_ROK_population_01_id['1990_01']
dt38_ROK_population_01_id_02 = dt38_ROK_population_01_id['1990_02']
dt38_ROK_population_01_id_03 = dt38_ROK_population_01_id['1990_03']
dt38_ROK_population_01_id_04 = dt38_ROK_population_01_id['1990_04']
dt38_ROK_population_01_id_05 = dt38_ROK_population_01_id['1990_05']
dt38_ROK_population_01_id_06 = dt38_ROK_population_01_id['1990_06']
dt38_ROK_population_01_id_07 = dt38_ROK_population_01_id['1990_07']
dt38_ROK_population_01_id_08 = dt38_ROK_population_01_id['1990_08']
dt38_ROK_population_01_id_09 = dt38_ROK_population_01_id['1990_09']
dt38_ROK_population_01_id_10 = dt38_ROK_population_01_id['1990_10']
dt38_ROK_population_01_id_11 = dt38_ROK_population_01_id['1990_11']
dt38_ROK_population_01_id_12 = dt38_ROK_population_01_id['1990_12']
dt38_ROK_population_01_id_13 = dt38_ROK_population_01_id['1990_13']
dt38_ROK_population_01_id_14 = dt38_ROK_population_01_id['1990_14']
dt38_ROK_population_01_id_15 = dt38_ROK_population_01_id['1990_15']
dt38_ROK_population_01_id_16 = dt38_ROK_population_01_id['1990_16']

```

```

[156]: # 05.02.12-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt39

dt39_ROK_population_01.insert(0, 'index', range(0, 0 + len(dt39_ROK_population_01)))
dt39_ROK_population_01_id = dt39_ROK_population_01.loc[1]

```

```

[157]: # 05.02.12-02
# change column names for dt39
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 1985 population

```

```

# dt29

dt39_ROK_population_01_id.rename({"1985_ttl": "1985_00"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.1": "1985_01"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.2": "1985_02"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.3": "1985_03"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.4": "1985_04"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.5": "1985_05"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.6": "1985_06"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.7": "1985_07"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.8": "1985_08"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.9": "1985_09"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.10": "1985_10"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.11": "1985_11"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.12": "1985_12"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.13": "1985_13"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.14": "1985_14"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.15": "1985_15"}, inplace = True)
dt39_ROK_population_01_id.rename({"1985.16": "1985_16"}, inplace = True)

```

```

[158]: # 05.02.12-03
# assign variables for dt39
# add variable 00 for total
# new variable for each column in dataset
# 1985 population
# dt39

dt39_ROK_population_01_id_00 = dt39_ROK_population_01_id['1985_00']
dt39_ROK_population_01_id_01 = dt39_ROK_population_01_id['1985_01']
dt39_ROK_population_01_id_02 = dt39_ROK_population_01_id['1985_02']
dt39_ROK_population_01_id_03 = dt39_ROK_population_01_id['1985_03']
dt39_ROK_population_01_id_04 = dt39_ROK_population_01_id['1985_04']
dt39_ROK_population_01_id_05 = dt39_ROK_population_01_id['1985_05']
dt39_ROK_population_01_id_06 = dt39_ROK_population_01_id['1985_06']
dt39_ROK_population_01_id_07 = dt39_ROK_population_01_id['1985_07']
dt39_ROK_population_01_id_08 = dt39_ROK_population_01_id['1985_08']
dt39_ROK_population_01_id_09 = dt39_ROK_population_01_id['1985_09']
dt39_ROK_population_01_id_10 = dt39_ROK_population_01_id['1985_10']
dt39_ROK_population_01_id_11 = dt39_ROK_population_01_id['1985_11']
dt39_ROK_population_01_id_12 = dt39_ROK_population_01_id['1985_12']
dt39_ROK_population_01_id_13 = dt39_ROK_population_01_id['1985_13']
dt39_ROK_population_01_id_14 = dt39_ROK_population_01_id['1985_14']
dt39_ROK_population_01_id_15 = dt39_ROK_population_01_id['1985_15']
dt39_ROK_population_01_id_16 = dt39_ROK_population_01_id['1985_16']

```

```

[159]: # 05.02.13-01
# add index column

```

```

# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt40

dt40_ROK_population_01.insert(0, 'index', range(0, 0 +
↳ len(dt40_ROK_population_01)))
dt40_ROK_population_01_id = dt40_ROK_population_01.loc[1]

```

```

[160]: # 05.02.13-02
# change column names for dt40
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 1980 population
# dt40

dt40_ROK_population_01_id.rename({"1980_tt1": "1980_00"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.1": "1980_01"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.2": "1980_02"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.3": "1980_03"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.4": "1980_04"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.5": "1980_05"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.6": "1980_06"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.7": "1980_07"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.8": "1980_08"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.9": "1980_09"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.10": "1980_10"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.11": "1980_11"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.12": "1980_12"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.13": "1980_13"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.14": "1980_14"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.15": "1980_15"}, inplace = True)
dt40_ROK_population_01_id.rename({"1980.16": "1980_16"}, inplace = True)

```

```

[161]: # 05.02.13-03
# assign variables for dt40
# add variable 00 for total
# new variable for each column in dataset
# 1980 population
# dt40

dt40_ROK_population_01_id_00 = dt40_ROK_population_01_id['1980_00']
dt40_ROK_population_01_id_01 = dt40_ROK_population_01_id['1980_01']
dt40_ROK_population_01_id_02 = dt40_ROK_population_01_id['1980_02']
dt40_ROK_population_01_id_03 = dt40_ROK_population_01_id['1980_03']
dt40_ROK_population_01_id_04 = dt40_ROK_population_01_id['1980_04']
dt40_ROK_population_01_id_05 = dt40_ROK_population_01_id['1980_05']

```

```

dt40_ROK_population_01_id_06 = dt40_ROK_population_01_id['1980_06']
dt40_ROK_population_01_id_07 = dt40_ROK_population_01_id['1980_07']
dt40_ROK_population_01_id_08 = dt40_ROK_population_01_id['1980_08']
dt40_ROK_population_01_id_09 = dt40_ROK_population_01_id['1980_09']
dt40_ROK_population_01_id_10 = dt40_ROK_population_01_id['1980_10']
dt40_ROK_population_01_id_11 = dt40_ROK_population_01_id['1980_11']
dt40_ROK_population_01_id_12 = dt40_ROK_population_01_id['1980_12']
dt40_ROK_population_01_id_13 = dt40_ROK_population_01_id['1980_13']
dt40_ROK_population_01_id_14 = dt40_ROK_population_01_id['1980_14']
dt40_ROK_population_01_id_15 = dt40_ROK_population_01_id['1980_15']
dt40_ROK_population_01_id_16 = dt40_ROK_population_01_id['1980_16']

```

```

[162]: # 05.02.14-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt41

dt41_ROK_population_01.insert(0, 'index', range(0, 0 +
↪len(dt41_ROK_population_01)))
dt41_ROK_population_01_id = dt41_ROK_population_01.loc[1]

```

```

[163]: # 05.02.14-02
# change column names for dt41
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 1975 population
# dt41

dt41_ROK_population_01_id.rename({"1975_tt1": "1975_00"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.1": "1975_01"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.2": "1975_02"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.3": "1975_03"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.4": "1975_04"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.5": "1975_05"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.6": "1975_06"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.7": "1975_07"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.8": "1975_08"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.9": "1975_09"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.10": "1975_10"}, inplace = True)
dt41_ROK_population_01_id.rename({"1975.11": "1975_11"}, inplace = True)

```

```

[164]: # 05.02.14-03
# assign variables for dt41
# add variable 00 for total
# new variable for each column in dataset

```

```

# 1975 population
# dt41

dt41_ROK_population_01_id_00 = dt41_ROK_population_01_id['1975_00']
dt41_ROK_population_01_id_01 = dt41_ROK_population_01_id['1975_01']
dt41_ROK_population_01_id_02 = dt41_ROK_population_01_id['1975_02']
dt41_ROK_population_01_id_03 = dt41_ROK_population_01_id['1975_03']
dt41_ROK_population_01_id_04 = dt41_ROK_population_01_id['1975_04']
dt41_ROK_population_01_id_05 = dt41_ROK_population_01_id['1975_05']
dt41_ROK_population_01_id_06 = dt41_ROK_population_01_id['1975_06']
dt41_ROK_population_01_id_07 = dt41_ROK_population_01_id['1975_07']
dt41_ROK_population_01_id_08 = dt41_ROK_population_01_id['1975_08']
dt41_ROK_population_01_id_09 = dt41_ROK_population_01_id['1975_09']
dt41_ROK_population_01_id_10 = dt41_ROK_population_01_id['1975_10']
dt41_ROK_population_01_id_11 = dt41_ROK_population_01_id['1975_11']

```

```

[165]: # 05.02.15-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt42

dt42_ROK_population_01.insert(0, 'index', range(0, 0 +
↪len(dt42_ROK_population_01)))
dt42_ROK_population_01_id = dt42_ROK_population_01.loc[1]

```

```

[166]: # 05.02.15-02
# change column names for dt42
# this step makes data easier to navigate
# removing decimal from column names to make easier to select string
# add variable 00 for total
# 1970 population
# dt42

dt42_ROK_population_01_id.rename({"1970_tt1": "1970_00"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.1": "1970_01"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.2": "1970_02"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.3": "1970_03"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.4": "1970_04"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.5": "1970_05"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.6": "1970_06"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.7": "1970_07"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.8": "1970_08"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.9": "1970_09"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.10": "1970_10"}, inplace=True)
dt42_ROK_population_01_id.rename({"1970.11": "1970_11"}, inplace=True)

```



```
[167]: # 05.02.15-03
# assign variables for dt42
# add variable 00 for total
# new variable for each column in dataset
# 1970 population
# dt42

dt42_ROK_population_01_id_00 = dt42_ROK_population_01_id['1970_00']
dt42_ROK_population_01_id_01 = dt42_ROK_population_01_id['1970_01']
dt42_ROK_population_01_id_02 = dt42_ROK_population_01_id['1970_02']
dt42_ROK_population_01_id_03 = dt42_ROK_population_01_id['1970_03']
dt42_ROK_population_01_id_04 = dt42_ROK_population_01_id['1970_04']
dt42_ROK_population_01_id_05 = dt42_ROK_population_01_id['1970_05']
dt42_ROK_population_01_id_06 = dt42_ROK_population_01_id['1970_06']
dt42_ROK_population_01_id_07 = dt42_ROK_population_01_id['1970_07']
dt42_ROK_population_01_id_08 = dt42_ROK_population_01_id['1970_08']
dt42_ROK_population_01_id_09 = dt42_ROK_population_01_id['1970_09']
dt42_ROK_population_01_id_10 = dt42_ROK_population_01_id['1970_10']
dt42_ROK_population_01_id_11 = dt42_ROK_population_01_id['1970_11']
```

```
[168]: # 05.02.16-01
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt43

dt43_Seoul_apr_____01.insert(0, 'id', range(0, 0 + len(dt43_Seoul_apr_____01)))
dt43_Seoul_apr_____01_id = dt43_Seoul_apr_____01.loc[0]
```

```
[169]: # 05.02.16-02
# change column names for dt43
# this step makes data easier to navigate
# add variable 00 for total
# future population
# dt43

dt43_Seoul_apr_____01_id.rename({'id': "dt13_00"}, inplace=True)
```

```
[170]: # 05.02.16-03
# add index column
# index column to select specific rows
# assign variable to row 'whole country' for entire population
# dt32

"""dt32_ROK_population_01_seoul_drp.insert(0, 'index', range(0, 0 + len(dt32_ROK_population_01_seoul_drp)))"""
```

```

"""dt32_R0K_population_01_seoul_drp_id = dt32_R0K_population_01_seoul_drp.
↪ loc[0]"""
dt32_R0K_population_01_seoul_drp_id = dt32_R0K_population_01_seoul_drp

```

```

[171]: # 05.02.16-04
# change column names for dt32
# this step makes data easier to navigate
# add variable 00 for total
# future population
# dt32

dt32_R0K_population_01_seoul_drp_id.rename({'index': "dt32_00"}, inplace=True)

```

```

[172]: # 05.02.16-05
# create indices for dt32
# future population
# create list with two columns for visualizations
# dt32

dt32_R0K_population_01_seoul_drp_id_dt = [
    ['2022', 51672569.0],
    ['2023', 51712619.0],
    ['2024', 51751065.0],
    ['2025', 51684564.0],
    ['2026', 51609121.0],
    ['2027', 51534551.0],
    ['2028', 51459877.0],
    ['2029', 51384052.0],
    ['2030', 51305713.0],
    ['2031', 51223269.0],
    ['2032', 51135265.0],
    ['2033', 51040632.0],
    ['2034', 50937726.0],
    ['2035', 50824868.0],
    ['2036', 50701389.0],
    ['2037', 50567057.0],
    ['2038', 50417167.0],
    ['2039', 50247657.0],
    ['2040', 50059218.0],
    ['2041', 49851637.0],
    ['2042', 49625415.0],
    ['2043', 49380766.0],
    ['2044', 49117432.0],
    ['2045', 48835031.0],
    ['2046', 48532231.0],
    ['2047', 48208374.0],
    ['2048', 47863071.0],

```



```

        ['2049', 47495914.0],
        ['2050', 47106960.0],
        ['2051', 46697391.0],
        ['2052', 46267614.0],
        ['2053', 45818111.0],
        ['2054', 45350648.0],
        ['2055', 44867534.0],
        ['2056', 44371641.0],
        ['2057', 43864958.0],
        ['2058', 43349217.0],
        ['2059', 42827316.0],
        ['2060', 42302086.0],
        ['2061', 41775305.0],
        ['2062', 41248542.0],
        ['2063', 40723791.0],
        ['2064', 40202384.0],
        ['2065', 39685210.0],
        ['2066', 39172974.0],
        ['2067', 38666431.0],
        ['2068', 38165892.0],
        ['2069', 37670942.0],
        ['2070', 37181774.0],
        ['2071', 36698961.0],
        ['2072', 36222293.0]
    ]

dt32_R0K_population_01_seoul_drp_id_dt_00 = pd.
↳DataFrame(dt32_R0K_population_01_seoul_drp_id_dt, columns = ['Year',
↳'Population'])

```

```

[173]: # 05.02.16-06
# convert integer date column to dates in international format
# US format is incompatible
# convert strings to floats
# this step makes all data quantifiable for calculations and visualizations
# dt32

dt32_R0K_population_01_seoul_drp_01_dtc = pd.
↳to_datetime(dt32_R0K_population_01_seoul_drp_01, format = '%Y/%m/%d')
dt32_R0K_population_01_seoul_drp_02_flt = dt32_R0K_population_01_seoul_drp_02.
↳astype(float)
dt32_R0K_population_01_seoul_drp_03_flt = dt32_R0K_population_01_seoul_drp_03.
↳astype(float)
dt32_R0K_population_01_seoul_drp_04_flt = dt32_R0K_population_01_seoul_drp_04.
↳astype(float)

```

```

[174]: # 05.02.17-01
# convert strings to floats

```

```
# this step makes all data quantifiable for calculations and visualizations  
# dt30
```

```
dt30_ROK_demo_____01_01_flt = dt30_ROK_demo_____01_01.astype(float)  
dt30_ROK_demo_____01_02_flt = dt30_ROK_demo_____01_02.astype(float)  
dt30_ROK_demo_____01_03_flt = dt30_ROK_demo_____01_03.astype(float)
```

```
[175]: # 05.02.18-01  
# drop index 0  
# add variable 0 for total  
# convert strings to floats  
# add variable 0 for total  
# this step makes all data quantifiable for calculations and visualizations  
# dt33
```

```
dt33_ROK_population_01_id_00_flt = float(dt33_ROK_population_01_id_00)  
dt33_ROK_population_01_id_01_flt = float(dt33_ROK_population_01_id_01)  
dt33_ROK_population_01_id_02_flt = float(dt33_ROK_population_01_id_02)  
dt33_ROK_population_01_id_03_flt = float(dt33_ROK_population_01_id_03)  
dt33_ROK_population_01_id_04_flt = float(dt33_ROK_population_01_id_04)  
dt33_ROK_population_01_id_05_flt = float(dt33_ROK_population_01_id_05)  
dt33_ROK_population_01_id_06_flt = float(dt33_ROK_population_01_id_06)  
dt33_ROK_population_01_id_07_flt = float(dt33_ROK_population_01_id_07)  
dt33_ROK_population_01_id_08_flt = float(dt33_ROK_population_01_id_08)  
dt33_ROK_population_01_id_09_flt = float(dt33_ROK_population_01_id_09)  
dt33_ROK_population_01_id_10_flt = float(dt33_ROK_population_01_id_10)  
dt33_ROK_population_01_id_11_flt = float(dt33_ROK_population_01_id_11)  
dt33_ROK_population_01_id_12_flt = float(dt33_ROK_population_01_id_12)  
dt33_ROK_population_01_id_13_flt = float(dt33_ROK_population_01_id_13)  
dt33_ROK_population_01_id_14_flt = float(dt33_ROK_population_01_id_14)  
dt33_ROK_population_01_id_15_flt = float(dt33_ROK_population_01_id_15)  
dt33_ROK_population_01_id_16_flt = float(dt33_ROK_population_01_id_16)  
dt33_ROK_population_01_id_17_flt = float(dt33_ROK_population_01_id_17)  
dt33_ROK_population_01_id_18_flt = float(dt33_ROK_population_01_id_18)  
dt33_ROK_population_01_id_19_flt = float(dt33_ROK_population_01_id_19)
```

```
[176]: # 05.02.18-02  
# drop index 0  
# add variable 0 for total  
# convert strings to floats  
# add variable 0 for total  
# this step makes all data quantifiable for calculations and visualizations  
# dt34
```

```
dt34_ROK_population_01_id_00_flt = float(dt34_ROK_population_01_id_00)  
dt34_ROK_population_01_id_01_flt = float(dt34_ROK_population_01_id_01)  
dt34_ROK_population_01_id_02_flt = float(dt34_ROK_population_01_id_02)
```

```

dt34_ROK_population_01_id_03_flt = float(dt34_ROK_population_01_id_03)
dt34_ROK_population_01_id_04_flt = float(dt34_ROK_population_01_id_04)
dt34_ROK_population_01_id_05_flt = float(dt34_ROK_population_01_id_05)
dt34_ROK_population_01_id_06_flt = float(dt34_ROK_population_01_id_06)
dt34_ROK_population_01_id_07_flt = float(dt34_ROK_population_01_id_07)
dt34_ROK_population_01_id_08_flt = float(dt34_ROK_population_01_id_08)
dt34_ROK_population_01_id_09_flt = float(dt34_ROK_population_01_id_09)
dt34_ROK_population_01_id_10_flt = float(dt34_ROK_population_01_id_10)
dt34_ROK_population_01_id_11_flt = float(dt34_ROK_population_01_id_11)
dt34_ROK_population_01_id_12_flt = float(dt34_ROK_population_01_id_12)
dt34_ROK_population_01_id_13_flt = float(dt34_ROK_population_01_id_13)
dt34_ROK_population_01_id_14_flt = float(dt34_ROK_population_01_id_14)
dt34_ROK_population_01_id_15_flt = float(dt34_ROK_population_01_id_15)
dt34_ROK_population_01_id_16_flt = float(dt34_ROK_population_01_id_16)
dt34_ROK_population_01_id_17_flt = float(dt34_ROK_population_01_id_17)
dt34_ROK_population_01_id_18_flt = float(dt34_ROK_population_01_id_18)
dt34_ROK_population_01_id_19_flt = float(dt34_ROK_population_01_id_19)

```

```

[177]: # 05.02.18-03
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt35

```

```

dt35_ROK_population_01_id_00_flt = float(dt35_ROK_population_01_id_00)
dt35_ROK_population_01_id_01_flt = float(dt35_ROK_population_01_id_01)
dt35_ROK_population_01_id_02_flt = float(dt35_ROK_population_01_id_02)
dt35_ROK_population_01_id_03_flt = float(dt35_ROK_population_01_id_03)
dt35_ROK_population_01_id_04_flt = float(dt35_ROK_population_01_id_04)
dt35_ROK_population_01_id_05_flt = float(dt35_ROK_population_01_id_05)
dt35_ROK_population_01_id_06_flt = float(dt35_ROK_population_01_id_06)
dt35_ROK_population_01_id_07_flt = float(dt35_ROK_population_01_id_07)
dt35_ROK_population_01_id_08_flt = float(dt35_ROK_population_01_id_08)
dt35_ROK_population_01_id_09_flt = float(dt35_ROK_population_01_id_09)
dt35_ROK_population_01_id_10_flt = float(dt35_ROK_population_01_id_10)
dt35_ROK_population_01_id_11_flt = float(dt35_ROK_population_01_id_11)
dt35_ROK_population_01_id_12_flt = float(dt35_ROK_population_01_id_12)
dt35_ROK_population_01_id_13_flt = float(dt35_ROK_population_01_id_13)
dt35_ROK_population_01_id_14_flt = float(dt35_ROK_population_01_id_14)
dt35_ROK_population_01_id_15_flt = float(dt35_ROK_population_01_id_15)
dt35_ROK_population_01_id_16_flt = float(dt35_ROK_population_01_id_16)
dt35_ROK_population_01_id_17_flt = float(dt35_ROK_population_01_id_17)
dt35_ROK_population_01_id_18_flt = float(dt35_ROK_population_01_id_18)
dt35_ROK_population_01_id_19_flt = float(dt35_ROK_population_01_id_19)

```

```
[178]: # 05.02.18-04
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt36

dt36_ROK_population_01_id_00_flt = float(dt36_ROK_population_01_id_00)
dt36_ROK_population_01_id_01_flt = float(dt36_ROK_population_01_id_01)
dt36_ROK_population_01_id_02_flt = float(dt36_ROK_population_01_id_02)
dt36_ROK_population_01_id_03_flt = float(dt36_ROK_population_01_id_03)
dt36_ROK_population_01_id_04_flt = float(dt36_ROK_population_01_id_04)
dt36_ROK_population_01_id_05_flt = float(dt36_ROK_population_01_id_05)
dt36_ROK_population_01_id_06_flt = float(dt36_ROK_population_01_id_06)
dt36_ROK_population_01_id_07_flt = float(dt36_ROK_population_01_id_07)
dt36_ROK_population_01_id_08_flt = float(dt36_ROK_population_01_id_08)
dt36_ROK_population_01_id_09_flt = float(dt36_ROK_population_01_id_09)
dt36_ROK_population_01_id_10_flt = float(dt36_ROK_population_01_id_10)
dt36_ROK_population_01_id_11_flt = float(dt36_ROK_population_01_id_11)
dt36_ROK_population_01_id_12_flt = float(dt36_ROK_population_01_id_12)
dt36_ROK_population_01_id_13_flt = float(dt36_ROK_population_01_id_13)
dt36_ROK_population_01_id_14_flt = float(dt36_ROK_population_01_id_14)
dt36_ROK_population_01_id_15_flt = float(dt36_ROK_population_01_id_15)
dt36_ROK_population_01_id_16_flt = float(dt36_ROK_population_01_id_16)
dt36_ROK_population_01_id_17_flt = float(dt36_ROK_population_01_id_17)
dt36_ROK_population_01_id_18_flt = float(dt36_ROK_population_01_id_18)
dt36_ROK_population_01_id_19_flt = float(dt36_ROK_population_01_id_19)
```

```
[179]: # 05.02.18-05
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt37

dt37_ROK_population_01_id_00_flt = float(dt37_ROK_population_01_id_00)
dt37_ROK_population_01_id_01_flt = float(dt37_ROK_population_01_id_01)
dt37_ROK_population_01_id_02_flt = float(dt37_ROK_population_01_id_02)
dt37_ROK_population_01_id_03_flt = float(dt37_ROK_population_01_id_03)
dt37_ROK_population_01_id_04_flt = float(dt37_ROK_population_01_id_04)
dt37_ROK_population_01_id_05_flt = float(dt37_ROK_population_01_id_05)
dt37_ROK_population_01_id_06_flt = float(dt37_ROK_population_01_id_06)
dt37_ROK_population_01_id_07_flt = float(dt37_ROK_population_01_id_07)
dt37_ROK_population_01_id_08_flt = float(dt37_ROK_population_01_id_08)
dt37_ROK_population_01_id_09_flt = float(dt37_ROK_population_01_id_09)
```

```

dt37_ROK_population_01_id_10_flt = float(dt37_ROK_population_01_id_10)
dt37_ROK_population_01_id_11_flt = float(dt37_ROK_population_01_id_11)
dt37_ROK_population_01_id_12_flt = float(dt37_ROK_population_01_id_12)
dt37_ROK_population_01_id_13_flt = float(dt37_ROK_population_01_id_13)
dt37_ROK_population_01_id_14_flt = float(dt37_ROK_population_01_id_14)
dt37_ROK_population_01_id_15_flt = float(dt37_ROK_population_01_id_15)
dt37_ROK_population_01_id_16_flt = float(dt37_ROK_population_01_id_16)

```

```

[180]: # 05.02.18-06
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt38

```

```

dt38_ROK_population_01_id_00_flt = float(dt38_ROK_population_01_id_00)
dt38_ROK_population_01_id_01_flt = float(dt38_ROK_population_01_id_01)
dt38_ROK_population_01_id_02_flt = float(dt38_ROK_population_01_id_02)
dt38_ROK_population_01_id_03_flt = float(dt38_ROK_population_01_id_03)
dt38_ROK_population_01_id_04_flt = float(dt38_ROK_population_01_id_04)
dt38_ROK_population_01_id_05_flt = float(dt38_ROK_population_01_id_05)
dt38_ROK_population_01_id_06_flt = float(dt38_ROK_population_01_id_06)
dt38_ROK_population_01_id_07_flt = float(dt38_ROK_population_01_id_07)
dt38_ROK_population_01_id_08_flt = float(dt38_ROK_population_01_id_08)
dt38_ROK_population_01_id_09_flt = float(dt38_ROK_population_01_id_09)
dt38_ROK_population_01_id_10_flt = float(dt38_ROK_population_01_id_10)
dt38_ROK_population_01_id_11_flt = float(dt38_ROK_population_01_id_11)
dt38_ROK_population_01_id_12_flt = float(dt38_ROK_population_01_id_12)
dt38_ROK_population_01_id_13_flt = float(dt38_ROK_population_01_id_13)
dt38_ROK_population_01_id_14_flt = float(dt38_ROK_population_01_id_14)
dt38_ROK_population_01_id_15_flt = float(dt38_ROK_population_01_id_15)
dt38_ROK_population_01_id_16_flt = float(dt38_ROK_population_01_id_16)

```

```

[181]: # 05.02.18-07
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt39

```

```

dt39_ROK_population_01_id_00_flt = float(dt39_ROK_population_01_id_00)
dt39_ROK_population_01_id_01_flt = float(dt39_ROK_population_01_id_01)
dt39_ROK_population_01_id_02_flt = float(dt39_ROK_population_01_id_02)
dt39_ROK_population_01_id_03_flt = float(dt39_ROK_population_01_id_03)
dt39_ROK_population_01_id_04_flt = float(dt39_ROK_population_01_id_04)

```

```

dt39_ROK_population_01_id_05_flt = float(dt39_ROK_population_01_id_05)
dt39_ROK_population_01_id_06_flt = float(dt39_ROK_population_01_id_06)
dt39_ROK_population_01_id_07_flt = float(dt39_ROK_population_01_id_07)
dt39_ROK_population_01_id_08_flt = float(dt39_ROK_population_01_id_08)
dt39_ROK_population_01_id_09_flt = float(dt39_ROK_population_01_id_09)
dt39_ROK_population_01_id_10_flt = float(dt39_ROK_population_01_id_10)
dt39_ROK_population_01_id_11_flt = float(dt39_ROK_population_01_id_11)
dt39_ROK_population_01_id_12_flt = float(dt39_ROK_population_01_id_12)
dt39_ROK_population_01_id_13_flt = float(dt39_ROK_population_01_id_13)
dt39_ROK_population_01_id_14_flt = float(dt39_ROK_population_01_id_14)
dt39_ROK_population_01_id_15_flt = float(dt39_ROK_population_01_id_15)
dt39_ROK_population_01_id_16_flt = float(dt39_ROK_population_01_id_16)

```

```

[182]: # 05.02.18-08
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt40

dt40_ROK_population_01_id_00_flt = float(dt40_ROK_population_01_id_00)
dt40_ROK_population_01_id_01_flt = float(dt40_ROK_population_01_id_01)
dt40_ROK_population_01_id_02_flt = float(dt40_ROK_population_01_id_02)
dt40_ROK_population_01_id_03_flt = float(dt40_ROK_population_01_id_03)
dt40_ROK_population_01_id_04_flt = float(dt40_ROK_population_01_id_04)
dt40_ROK_population_01_id_05_flt = float(dt40_ROK_population_01_id_05)
dt40_ROK_population_01_id_06_flt = float(dt40_ROK_population_01_id_06)
dt40_ROK_population_01_id_07_flt = float(dt40_ROK_population_01_id_07)
dt40_ROK_population_01_id_08_flt = float(dt40_ROK_population_01_id_08)
dt40_ROK_population_01_id_09_flt = float(dt40_ROK_population_01_id_09)
dt40_ROK_population_01_id_10_flt = float(dt40_ROK_population_01_id_10)
dt40_ROK_population_01_id_11_flt = float(dt40_ROK_population_01_id_11)
dt40_ROK_population_01_id_12_flt = float(dt40_ROK_population_01_id_12)
dt40_ROK_population_01_id_13_flt = float(dt40_ROK_population_01_id_13)
dt40_ROK_population_01_id_14_flt = float(dt40_ROK_population_01_id_14)
dt40_ROK_population_01_id_15_flt = float(dt40_ROK_population_01_id_15)
dt40_ROK_population_01_id_16_flt = float(dt40_ROK_population_01_id_16)

```

```

[183]: # 05.02.18-09
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt41

```

```

dt41_ROK_population_01_id_00_flt = float(dt41_ROK_population_01_id_00)
dt41_ROK_population_01_id_01_flt = float(dt41_ROK_population_01_id_01)
dt41_ROK_population_01_id_02_flt = float(dt41_ROK_population_01_id_02)
dt41_ROK_population_01_id_03_flt = float(dt41_ROK_population_01_id_03)
dt41_ROK_population_01_id_04_flt = float(dt41_ROK_population_01_id_04)
dt41_ROK_population_01_id_05_flt = float(dt41_ROK_population_01_id_05)
dt41_ROK_population_01_id_06_flt = float(dt41_ROK_population_01_id_06)
dt41_ROK_population_01_id_07_flt = float(dt41_ROK_population_01_id_07)
dt41_ROK_population_01_id_08_flt = float(dt41_ROK_population_01_id_08)
dt41_ROK_population_01_id_09_flt = float(dt41_ROK_population_01_id_09)
dt41_ROK_population_01_id_10_flt = float(dt41_ROK_population_01_id_10)
dt41_ROK_population_01_id_11_flt = float(dt41_ROK_population_01_id_11)

```

```

[184]: # 05.02.18-10
# drop index 0
# add variable 0 for total
# convert strings to floats
# add variable 0 for total
# this step makes all data quantifiable for calculations and visualizations
# dt42

dt42_ROK_population_01_id_00_flt = float(dt42_ROK_population_01_id_00)
dt42_ROK_population_01_id_01_flt = float(dt42_ROK_population_01_id_01)
dt42_ROK_population_01_id_02_flt = float(dt42_ROK_population_01_id_02)
dt42_ROK_population_01_id_03_flt = float(dt42_ROK_population_01_id_03)
dt42_ROK_population_01_id_04_flt = float(dt42_ROK_population_01_id_04)
dt42_ROK_population_01_id_05_flt = float(dt42_ROK_population_01_id_05)
dt42_ROK_population_01_id_06_flt = float(dt42_ROK_population_01_id_06)
dt42_ROK_population_01_id_07_flt = float(dt42_ROK_population_01_id_07)
dt42_ROK_population_01_id_08_flt = float(dt42_ROK_population_01_id_08)
dt42_ROK_population_01_id_09_flt = float(dt42_ROK_population_01_id_09)
dt42_ROK_population_01_id_10_flt = float(dt42_ROK_population_01_id_10)
dt42_ROK_population_01_id_11_flt = float(dt42_ROK_population_01_id_11)

```

```

[185]: # 05.02.19-01
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt33

```



```

dt33_ROK_population_01_id_ttl = dt33_ROK_population_01_id_00_flt +
    ↪dt33_ROK_population_01_id_01_flt + dt33_ROK_population_01_id_02_flt +
    ↪dt33_ROK_population_01_id_03_flt + dt33_ROK_population_01_id_04_flt +
    ↪dt33_ROK_population_01_id_05_flt + dt33_ROK_population_01_id_06_flt +
    ↪dt33_ROK_population_01_id_07_flt + dt33_ROK_population_01_id_08_flt +
    ↪dt33_ROK_population_01_id_09_flt + dt33_ROK_population_01_id_10_flt +
    ↪dt33_ROK_population_01_id_11_flt + dt33_ROK_population_01_id_12_flt +
    ↪dt33_ROK_population_01_id_13_flt + dt33_ROK_population_01_id_14_flt +
    ↪dt33_ROK_population_01_id_15_flt + dt33_ROK_population_01_id_16_flt +
    ↪dt33_ROK_population_01_id_17_flt + dt33_ROK_population_01_id_18_flt +
    ↪dt33_ROK_population_01_id_19_flt
print(dt33_ROK_population_01_id_ttl)

```

291008943.0

```

[186]: # 05.02.19-02
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt34

dt34_ROK_population_01_id_ttl = dt34_ROK_population_01_id_00_flt +
    ↪dt34_ROK_population_01_id_01_flt + dt34_ROK_population_01_id_02_flt +
    ↪dt34_ROK_population_01_id_03_flt + dt34_ROK_population_01_id_04_flt +
    ↪dt34_ROK_population_01_id_05_flt + dt34_ROK_population_01_id_06_flt +
    ↪dt34_ROK_population_01_id_07_flt + dt34_ROK_population_01_id_08_flt +
    ↪dt34_ROK_population_01_id_09_flt + dt34_ROK_population_01_id_10_flt +
    ↪dt34_ROK_population_01_id_11_flt + dt34_ROK_population_01_id_12_flt +
    ↪dt34_ROK_population_01_id_13_flt + dt34_ROK_population_01_id_14_flt +
    ↪dt34_ROK_population_01_id_15_flt + dt34_ROK_population_01_id_16_flt +
    ↪dt34_ROK_population_01_id_17_flt + dt34_ROK_population_01_id_18_flt +
    ↪dt34_ROK_population_01_id_19_flt
print(dt34_ROK_population_01_id_ttl)

```

259127241.0

```

[187]: # 05.02.19-03
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt35

```



```

dt35_ROK_population_01_id_ttl = dt35_ROK_population_01_id_00_flt +
    ↪dt35_ROK_population_01_id_01_flt + dt35_ROK_population_01_id_02_flt +
    ↪dt35_ROK_population_01_id_03_flt + dt35_ROK_population_01_id_04_flt +
    ↪dt35_ROK_population_01_id_05_flt + dt35_ROK_population_01_id_06_flt +
    ↪dt35_ROK_population_01_id_07_flt + dt35_ROK_population_01_id_08_flt +
    ↪dt35_ROK_population_01_id_09_flt + dt35_ROK_population_01_id_10_flt +
    ↪dt35_ROK_population_01_id_11_flt + dt35_ROK_population_01_id_12_flt +
    ↪dt35_ROK_population_01_id_13_flt + dt35_ROK_population_01_id_14_flt +
    ↪dt35_ROK_population_01_id_15_flt + dt35_ROK_population_01_id_16_flt +
    ↪dt35_ROK_population_01_id_17_flt + dt35_ROK_population_01_id_18_flt +
    ↪dt35_ROK_population_01_id_19_flt
print(dt35_ROK_population_01_id_ttl)

```

247744145.0

```

[188]: # 05.02.19-04
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt36

dt36_ROK_population_01_id_ttl = dt36_ROK_population_01_id_00_flt +
    ↪dt36_ROK_population_01_id_01_flt + dt36_ROK_population_01_id_02_flt +
    ↪dt36_ROK_population_01_id_03_flt + dt36_ROK_population_01_id_04_flt +
    ↪dt36_ROK_population_01_id_05_flt + dt36_ROK_population_01_id_06_flt +
    ↪dt36_ROK_population_01_id_07_flt + dt36_ROK_population_01_id_08_flt +
    ↪dt36_ROK_population_01_id_09_flt + dt36_ROK_population_01_id_10_flt +
    ↪dt36_ROK_population_01_id_11_flt + dt36_ROK_population_01_id_12_flt +
    ↪dt36_ROK_population_01_id_13_flt + dt36_ROK_population_01_id_14_flt +
    ↪dt36_ROK_population_01_id_15_flt + dt36_ROK_population_01_id_16_flt +
    ↪dt36_ROK_population_01_id_17_flt + dt36_ROK_population_01_id_18_flt +
    ↪dt36_ROK_population_01_id_19_flt
print(dt36_ROK_population_01_id_ttl)

```

236363768.0

```

[189]: # 05.02.19-05
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt37

```

```

dt37_ROK_population_01_id_ttl = dt37_ROK_population_01_id_00_flt +
    ↪dt37_ROK_population_01_id_01_flt + dt37_ROK_population_01_id_02_flt +
    ↪dt37_ROK_population_01_id_03_flt + dt37_ROK_population_01_id_04_flt +
    ↪dt37_ROK_population_01_id_05_flt + dt37_ROK_population_01_id_06_flt +
    ↪dt37_ROK_population_01_id_07_flt + dt37_ROK_population_01_id_08_flt +
    ↪dt37_ROK_population_01_id_09_flt + dt37_ROK_population_01_id_10_flt +
    ↪dt37_ROK_population_01_id_11_flt + dt37_ROK_population_01_id_12_flt +
    ↪dt37_ROK_population_01_id_13_flt + dt37_ROK_population_01_id_14_flt +
    ↪dt37_ROK_population_01_id_15_flt + dt37_ROK_population_01_id_16_flt
print(dt37_ROK_population_01_id_ttl)

```

280517615.0

```

[190]: # 05.02.19-06
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt38

dt38_ROK_population_01_id_ttl = dt38_ROK_population_01_id_00_flt +
    ↪dt38_ROK_population_01_id_01_flt + dt38_ROK_population_01_id_02_flt +
    ↪dt38_ROK_population_01_id_03_flt + dt38_ROK_population_01_id_04_flt +
    ↪dt38_ROK_population_01_id_05_flt + dt38_ROK_population_01_id_06_flt +
    ↪dt38_ROK_population_01_id_07_flt + dt38_ROK_population_01_id_08_flt +
    ↪dt38_ROK_population_01_id_09_flt + dt38_ROK_population_01_id_10_flt +
    ↪dt38_ROK_population_01_id_11_flt + dt38_ROK_population_01_id_12_flt +
    ↪dt38_ROK_population_01_id_13_flt + dt38_ROK_population_01_id_14_flt +
    ↪dt38_ROK_population_01_id_15_flt + dt38_ROK_population_01_id_16_flt
print(dt38_ROK_population_01_id_ttl)

```

271785440.0

```

[191]: # 05.02.19-07
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt39

```

```

dt39_ROK_population_01_id_ttl = dt39_ROK_population_01_id_00_flt +
    ↪dt39_ROK_population_01_id_01_flt + dt39_ROK_population_01_id_02_flt +
    ↪dt39_ROK_population_01_id_03_flt + dt39_ROK_population_01_id_04_flt +
    ↪dt39_ROK_population_01_id_05_flt + dt39_ROK_population_01_id_06_flt +
    ↪dt39_ROK_population_01_id_07_flt + dt39_ROK_population_01_id_08_flt +
    ↪dt39_ROK_population_01_id_09_flt + dt39_ROK_population_01_id_10_flt +
    ↪dt39_ROK_population_01_id_11_flt + dt39_ROK_population_01_id_12_flt +
    ↪dt39_ROK_population_01_id_13_flt + dt39_ROK_population_01_id_14_flt +
    ↪dt39_ROK_population_01_id_15_flt + dt39_ROK_population_01_id_16_flt
print(dt39_ROK_population_01_id_ttl)

```

252215303.0

```

[192]: # 05.02.19-08
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt40

dt40_ROK_population_01_id_ttl = dt40_ROK_population_01_id_00_flt +
    ↪dt40_ROK_population_01_id_01_flt + dt40_ROK_population_01_id_02_flt +
    ↪dt40_ROK_population_01_id_03_flt + dt40_ROK_population_01_id_04_flt +
    ↪dt40_ROK_population_01_id_05_flt + dt40_ROK_population_01_id_06_flt +
    ↪dt40_ROK_population_01_id_07_flt + dt40_ROK_population_01_id_08_flt +
    ↪dt40_ROK_population_01_id_09_flt + dt40_ROK_population_01_id_10_flt +
    ↪dt40_ROK_population_01_id_11_flt + dt40_ROK_population_01_id_12_flt +
    ↪dt40_ROK_population_01_id_13_flt + dt40_ROK_population_01_id_14_flt +
    ↪dt40_ROK_population_01_id_15_flt + dt40_ROK_population_01_id_16_flt
print(dt40_ROK_population_01_id_ttl)

```

232537495.0

```

[193]: # 05.02.19-09
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt41

dt41_ROK_population_01_id_ttl = dt41_ROK_population_01_id_00_flt +
    ↪dt41_ROK_population_01_id_01_flt + dt41_ROK_population_01_id_02_flt +
    ↪dt41_ROK_population_01_id_03_flt + dt41_ROK_population_01_id_04_flt +
    ↪dt41_ROK_population_01_id_05_flt + dt41_ROK_population_01_id_06_flt +
    ↪dt41_ROK_population_01_id_07_flt + dt41_ROK_population_01_id_08_flt +
    ↪dt41_ROK_population_01_id_09_flt + dt41_ROK_population_01_id_10_flt +
    ↪dt41_ROK_population_01_id_11_flt
print(dt41_ROK_population_01_id_ttl)

```

152348958.0

```
[194]: # 05.02.19-10
# calculate total for entire country
# add variable 0 for total
# assign variable for total as _ttl
# return values
# dt42

dt42_ROK_population_01_id_ttl = dt42_ROK_population_01_id_00_flt +
    ↪dt42_ROK_population_01_id_01_flt + dt42_ROK_population_01_id_02_flt +
    ↪dt42_ROK_population_01_id_03_flt + dt42_ROK_population_01_id_04_flt +
    ↪dt42_ROK_population_01_id_05_flt + dt42_ROK_population_01_id_06_flt +
    ↪dt42_ROK_population_01_id_07_flt + dt42_ROK_population_01_id_08_flt +
    ↪dt42_ROK_population_01_id_09_flt + dt42_ROK_population_01_id_10_flt +
    ↪dt42_ROK_population_01_id_11_flt
print(dt42_ROK_population_01_id_ttl)
```

135256424.0

```
[195]: # 05.02.20-01
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt33

dt33_ROK_population_01_id_ttl_ = [
    dt33_ROK_population_01_id_00_flt,
    dt33_ROK_population_01_id_01_flt,
    dt33_ROK_population_01_id_02_flt,
    dt33_ROK_population_01_id_03_flt,
    dt33_ROK_population_01_id_04_flt,
    dt33_ROK_population_01_id_05_flt,
    dt33_ROK_population_01_id_06_flt,
    dt33_ROK_population_01_id_07_flt,
    dt33_ROK_population_01_id_08_flt,
    dt33_ROK_population_01_id_09_flt,
    dt33_ROK_population_01_id_10_flt,
    dt33_ROK_population_01_id_11_flt,
    dt33_ROK_population_01_id_12_flt,
    dt33_ROK_population_01_id_13_flt,
    dt33_ROK_population_01_id_14_flt,
    dt33_ROK_population_01_id_15_flt,
    dt33_ROK_population_01_id_16_flt,
    dt33_ROK_population_01_id_17_flt,
    dt33_ROK_population_01_id_18_flt,
    dt33_ROK_population_01_id_19_flt
```

```

    ]
dt33_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt33_ROK_population_01_id_ttl_)
print(dt33_ROK_population_01_id_ttl_x)

```

14550447.15

```

[196]: # 05.02.20-02
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt34

dt34_ROK_population_01_id_ttl_ = [
    dt34_ROK_population_01_id_00_flt,
    dt34_ROK_population_01_id_01_flt,
    dt34_ROK_population_01_id_02_flt,
    dt34_ROK_population_01_id_03_flt,
    dt34_ROK_population_01_id_04_flt,
    dt34_ROK_population_01_id_05_flt,
    dt34_ROK_population_01_id_06_flt,
    dt34_ROK_population_01_id_07_flt,
    dt34_ROK_population_01_id_08_flt,
    dt34_ROK_population_01_id_09_flt,
    dt34_ROK_population_01_id_10_flt,
    dt34_ROK_population_01_id_11_flt,
    dt34_ROK_population_01_id_12_flt,
    dt34_ROK_population_01_id_13_flt,
    dt34_ROK_population_01_id_14_flt,
    dt34_ROK_population_01_id_15_flt,
    dt34_ROK_population_01_id_16_flt,
    dt34_ROK_population_01_id_17_flt,
    dt34_ROK_population_01_id_18_flt,
    dt34_ROK_population_01_id_19_flt
]

dt34_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt34_ROK_population_01_id_ttl_)
print(dt34_ROK_population_01_id_ttl_x)

```

12956362.05

```

[197]: # 05.02.20-03
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt35

```

```

dt35_ROK_population_01_id_ttl_ = [
    dt35_ROK_population_01_id_00_flt,
    dt35_ROK_population_01_id_01_flt,
    dt35_ROK_population_01_id_02_flt,
    dt35_ROK_population_01_id_03_flt,
    dt35_ROK_population_01_id_04_flt,
    dt35_ROK_population_01_id_05_flt,
    dt35_ROK_population_01_id_06_flt,
    dt35_ROK_population_01_id_07_flt,
    dt35_ROK_population_01_id_08_flt,
    dt35_ROK_population_01_id_09_flt,
    dt35_ROK_population_01_id_10_flt,
    dt35_ROK_population_01_id_11_flt,
    dt35_ROK_population_01_id_12_flt,
    dt35_ROK_population_01_id_13_flt,
    dt35_ROK_population_01_id_14_flt,
    dt35_ROK_population_01_id_15_flt,
    dt35_ROK_population_01_id_16_flt,
    dt35_ROK_population_01_id_17_flt,
    dt35_ROK_population_01_id_18_flt,
    dt35_ROK_population_01_id_19_flt
]

dt35_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt35_ROK_population_01_id_ttl_)
print(dt35_ROK_population_01_id_ttl_x)

```

12387207.25

```

[198]: # 05.02.20-04
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt36

dt36_ROK_population_01_id_ttl_ = [
    dt36_ROK_population_01_id_00_flt,
    dt36_ROK_population_01_id_01_flt,
    dt36_ROK_population_01_id_02_flt,
    dt36_ROK_population_01_id_03_flt,
    dt36_ROK_population_01_id_04_flt,
    dt36_ROK_population_01_id_05_flt,
    dt36_ROK_population_01_id_06_flt,
    dt36_ROK_population_01_id_07_flt,
    dt36_ROK_population_01_id_08_flt,
    dt36_ROK_population_01_id_09_flt,

```

```

dt36_ROK_population_01_id_10_flt,
dt36_ROK_population_01_id_11_flt,
dt36_ROK_population_01_id_12_flt,
dt36_ROK_population_01_id_13_flt,
dt36_ROK_population_01_id_14_flt,
dt36_ROK_population_01_id_15_flt,
dt36_ROK_population_01_id_16_flt,
dt36_ROK_population_01_id_17_flt,
dt36_ROK_population_01_id_18_flt,
dt36_ROK_population_01_id_19_flt
]
dt36_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt36_ROK_population_01_id_ttl_)
print(dt36_ROK_population_01_id_ttl_x)

```

11818188.4

```

[199]: # 05.02.20-05
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt37

dt37_ROK_population_01_id_ttl_ = [
    dt37_ROK_population_01_id_00_flt,
    dt37_ROK_population_01_id_01_flt,
    dt37_ROK_population_01_id_02_flt,
    dt37_ROK_population_01_id_03_flt,
    dt37_ROK_population_01_id_04_flt,
    dt37_ROK_population_01_id_05_flt,
    dt37_ROK_population_01_id_06_flt,
    dt37_ROK_population_01_id_07_flt,
    dt37_ROK_population_01_id_08_flt,
    dt37_ROK_population_01_id_09_flt,
    dt37_ROK_population_01_id_10_flt,
    dt37_ROK_population_01_id_11_flt,
    dt37_ROK_population_01_id_12_flt,
    dt37_ROK_population_01_id_13_flt,
    dt37_ROK_population_01_id_14_flt,
    dt37_ROK_population_01_id_15_flt,
    dt37_ROK_population_01_id_16_flt
]
dt37_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt37_ROK_population_01_id_ttl_)
print(dt37_ROK_population_01_id_ttl_x)

```

16501036.176470589

```

[200]: # 05.02.20-06
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt38

dt38_ROK_population_01_id_ttl_ = [
    dt38_ROK_population_01_id_00_flt,
    dt38_ROK_population_01_id_01_flt,
    dt38_ROK_population_01_id_02_flt,
    dt38_ROK_population_01_id_03_flt,
    dt38_ROK_population_01_id_04_flt,
    dt38_ROK_population_01_id_05_flt,
    dt38_ROK_population_01_id_06_flt,
    dt38_ROK_population_01_id_07_flt,
    dt38_ROK_population_01_id_08_flt,
    dt38_ROK_population_01_id_09_flt,
    dt38_ROK_population_01_id_10_flt,
    dt38_ROK_population_01_id_11_flt,
    dt38_ROK_population_01_id_12_flt,
    dt38_ROK_population_01_id_13_flt,
    dt38_ROK_population_01_id_14_flt,
    dt38_ROK_population_01_id_15_flt,
    dt38_ROK_population_01_id_16_flt
]

dt38_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt38_ROK_population_01_id_ttl_)
print(dt38_ROK_population_01_id_ttl_x)

```

15987378.823529411

```

[201]: # 05.02.20-07
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt39

dt39_ROK_population_01_id_ttl_ = [
    dt39_ROK_population_01_id_00_flt,
    dt39_ROK_population_01_id_01_flt,
    dt39_ROK_population_01_id_02_flt,
    dt39_ROK_population_01_id_03_flt,
    dt39_ROK_population_01_id_04_flt,
    dt39_ROK_population_01_id_05_flt,
    dt39_ROK_population_01_id_06_flt,

```



```

dt39_ROK_population_01_id_07_flt,
dt39_ROK_population_01_id_08_flt,
dt39_ROK_population_01_id_09_flt,
dt39_ROK_population_01_id_10_flt,
dt39_ROK_population_01_id_11_flt,
dt39_ROK_population_01_id_12_flt,
dt39_ROK_population_01_id_13_flt,
dt39_ROK_population_01_id_14_flt,
dt39_ROK_population_01_id_15_flt,
dt39_ROK_population_01_id_16_flt
]
dt39_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt39_ROK_population_01_id_ttl_)
print(dt39_ROK_population_01_id_ttl_x)

```

14836194.294117646

```

[202]: # 05.02.20-08
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt40

dt40_ROK_population_01_id_ttl_ = [
    dt40_ROK_population_01_id_00_flt,
    dt40_ROK_population_01_id_01_flt,
    dt40_ROK_population_01_id_02_flt,
    dt40_ROK_population_01_id_03_flt,
    dt40_ROK_population_01_id_04_flt,
    dt40_ROK_population_01_id_05_flt,
    dt40_ROK_population_01_id_06_flt,
    dt40_ROK_population_01_id_07_flt,
    dt40_ROK_population_01_id_08_flt,
    dt40_ROK_population_01_id_09_flt,
    dt40_ROK_population_01_id_10_flt,
    dt40_ROK_population_01_id_11_flt,
    dt40_ROK_population_01_id_12_flt,
    dt40_ROK_population_01_id_13_flt,
    dt40_ROK_population_01_id_14_flt,
    dt40_ROK_population_01_id_15_flt,
    dt40_ROK_population_01_id_16_flt
]
dt40_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt40_ROK_population_01_id_ttl_)
print(dt40_ROK_population_01_id_ttl_x)

```

13678676.176470589

```
[203]: # 05.02.20-09
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt41

dt41_ROK_population_01_id_ttl_ = [
    dt41_ROK_population_01_id_00_flt,
    dt41_ROK_population_01_id_01_flt,
    dt41_ROK_population_01_id_02_flt,
    dt41_ROK_population_01_id_03_flt,
    dt41_ROK_population_01_id_04_flt,
    dt41_ROK_population_01_id_05_flt,
    dt41_ROK_population_01_id_06_flt,
    dt41_ROK_population_01_id_07_flt,
    dt41_ROK_population_01_id_08_flt,
    dt41_ROK_population_01_id_09_flt,
    dt41_ROK_population_01_id_10_flt,
    dt41_ROK_population_01_id_11_flt
]

dt41_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt41_ROK_population_01_id_ttl_)
print(dt41_ROK_population_01_id_ttl_x)
```

12695746.5

```
[204]: # 05.02.20-10
# calculate mean
# assign variable for mean as _x
# add variable 0 for total
# return mean
# dt42

dt42_ROK_population_01_id_ttl_ = [
    dt42_ROK_population_01_id_00_flt,
    dt42_ROK_population_01_id_01_flt,
    dt42_ROK_population_01_id_02_flt,
    dt42_ROK_population_01_id_03_flt,
    dt42_ROK_population_01_id_04_flt,
    dt42_ROK_population_01_id_05_flt,
    dt42_ROK_population_01_id_06_flt,
    dt42_ROK_population_01_id_07_flt,
    dt42_ROK_population_01_id_08_flt,
    dt42_ROK_population_01_id_09_flt,
    dt42_ROK_population_01_id_10_flt,
    dt42_ROK_population_01_id_11_flt
]
```

```

    ]
dt42_ROK_population_01_id_ttl_x = statistics.
    ↪mean(dt42_ROK_population_01_id_ttl_)
print(dt42_ROK_population_01_id_ttl_x)

```

11271368.666666666

```

[205]: # 05.02.21-01
# combine variables into new df
# compiles 19 rows: 01-19
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt33

np.random.seed(123)
dt33_ROK_population_01_id_ttl_yr_x = {
    '00':dt33_ROK_population_01_id_00_flt,
    '01':dt33_ROK_population_01_id_01_flt,
    '02':dt33_ROK_population_01_id_02_flt,
    '03':dt33_ROK_population_01_id_03_flt,
    '04':dt33_ROK_population_01_id_04_flt,
    '05':dt33_ROK_population_01_id_05_flt,
    '06':dt33_ROK_population_01_id_06_flt,
    '07':dt33_ROK_population_01_id_07_flt,
    '08':dt33_ROK_population_01_id_08_flt,
    '09':dt33_ROK_population_01_id_09_flt,
    '10':dt33_ROK_population_01_id_10_flt,
    '11':dt33_ROK_population_01_id_11_flt,
    '12':dt33_ROK_population_01_id_12_flt,
    '13':dt33_ROK_population_01_id_13_flt,
    '14':dt33_ROK_population_01_id_14_flt,
    '15':dt33_ROK_population_01_id_15_flt,
    '16':dt33_ROK_population_01_id_16_flt,
    '17':dt33_ROK_population_01_id_17_flt,
    '18':dt33_ROK_population_01_id_18_flt,
    '19':dt33_ROK_population_01_id_19_flt
}
dt33_ROK_population_01_id_ttl_yr_y = pd.
    ↪DataFrame(dt33_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt33_ROK_population_01_id_ttl_yr_y)

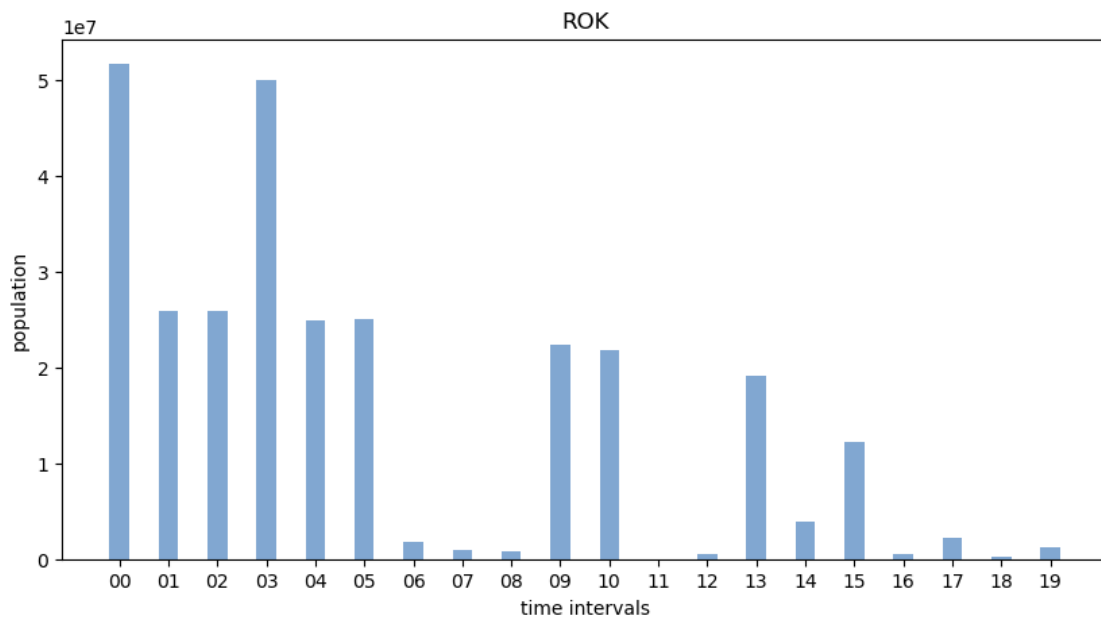
```

	00	01	02	...	17	18	19
0	51692272.0	25835298.0	25856974.0	...	2283238.0	211392.0	1162311.0

[1 rows x 20 columns]

```
[206]: # 05.02.21-02
# plot total populations by year
# combines 9 datasets: dt33-dt42
# assign variable for plot components as _yr_y
# dt33

dt33_ROK_population_01_id_ttl_yr_x_yr = list(dt33_ROK_population_01_id_ttl_yr_x.
↳keys())
dt33_ROK_population_01_id_ttl_yr_x_val =↳
↳list(dt33_ROK_population_01_id_ttl_yr_x.values())
dt33_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt33_ROK_population_01_id_ttl_yr_x_yr,↳
↳dt33_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()
```



```
[207]: # 05.02.21-03
# combine variables into new df
# compiles 19 rows: 01-19
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt34
```

```

np.random.seed(123)
dt34_ROK_population_01_id_ttl_yr_x = {
    '00':dt34_ROK_population_01_id_00_flt,
    '01':dt34_ROK_population_01_id_01_flt,
    '02':dt34_ROK_population_01_id_02_flt,
    '03':dt34_ROK_population_01_id_03_flt,
    '04':dt34_ROK_population_01_id_04_flt,
    '05':dt34_ROK_population_01_id_05_flt,
    '06':dt34_ROK_population_01_id_06_flt,
    '07':dt34_ROK_population_01_id_07_flt,
    '08':dt34_ROK_population_01_id_08_flt,
    '09':dt34_ROK_population_01_id_09_flt,
    '10':dt34_ROK_population_01_id_10_flt,
    '11':dt34_ROK_population_01_id_11_flt,
    '12':dt34_ROK_population_01_id_12_flt,
    '13':dt34_ROK_population_01_id_13_flt,
    '14':dt34_ROK_population_01_id_14_flt,
    '15':dt34_ROK_population_01_id_15_flt,
    '16':dt34_ROK_population_01_id_16_flt,
    '17':dt34_ROK_population_01_id_17_flt,
    '18':dt34_ROK_population_01_id_18_flt,
    '19':dt34_ROK_population_01_id_19_flt
}
dt34_ROK_population_01_id_ttl_yr_y = pd.
    ↳DataFrame(dt34_ROK_population_01_id_ttl_yr_x, index=[0])
print(dt34_ROK_population_01_id_ttl_yr_y)

```

```

           00           01           02  ...           17           18           19
0  48580293.0  24167098.0  24413195.0  ...  1314452.0  161393.0  303097.0

```

[1 rows x 20 columns]

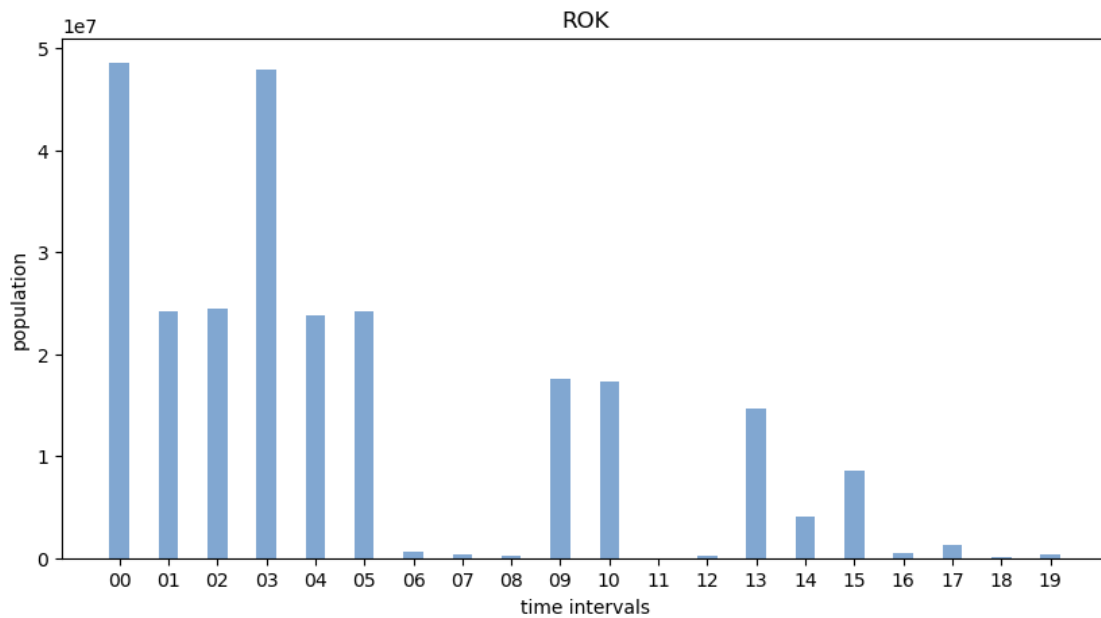
```

[208]: # 05.02.21-04
# plot total populations by year
# combines 9 datasets: dt33-dt42
# assign variable for plot components as _yr_y
# dt34

dt34_ROK_population_01_id_ttl_yr_x_yr = list(dt34_ROK_population_01_id_ttl_yr_x.
    ↳keys())
dt34_ROK_population_01_id_ttl_yr_x_val =
    ↳list(dt34_ROK_population_01_id_ttl_yr_x.values())
dt34_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt34_ROK_population_01_id_ttl_yr_x_yr,
    ↳dt34_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")

```

```
plt.title("ROK")
plt.show()
```



```
[209]: # 05.02.21-05
# combine variables into new df
# compiles 19 rows: 01-19
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt35

np.random.seed(123)
dt35_ROK_population_01_id_ttl_yr_x = {
    '00':dt35_ROK_population_01_id_00_flt,
    '01':dt35_ROK_population_01_id_01_flt,
    '02':dt35_ROK_population_01_id_02_flt,
    '03':dt35_ROK_population_01_id_03_flt,
    '04':dt35_ROK_population_01_id_04_flt,
    '05':dt35_ROK_population_01_id_05_flt,
    '06':dt35_ROK_population_01_id_06_flt,
    '07':dt35_ROK_population_01_id_07_flt,
    '08':dt35_ROK_population_01_id_08_flt,
    '09':dt35_ROK_population_01_id_09_flt,
    '10':dt35_ROK_population_01_id_10_flt,
    '11':dt35_ROK_population_01_id_11_flt,
    '12':dt35_ROK_population_01_id_12_flt,
```

```

        '13':dt35_ROK_population_01_id_13_flt,
        '14':dt35_ROK_population_01_id_14_flt,
        '15':dt35_ROK_population_01_id_15_flt,
        '16':dt35_ROK_population_01_id_16_flt,
        '17':dt35_ROK_population_01_id_17_flt,
        '18':dt35_ROK_population_01_id_18_flt,
        '19':dt35_ROK_population_01_id_19_flt
    }
dt35_ROK_population_01_id_ttl_yr_y = pd.
    ↳DataFrame(dt35_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt35_ROK_population_01_id_ttl_yr_y)

```

```

           00           01           02  ...           17           18           19
0  47278951.0  23623954.0  23654997.0  ...  1229208.0  208690.0  206511.0

```

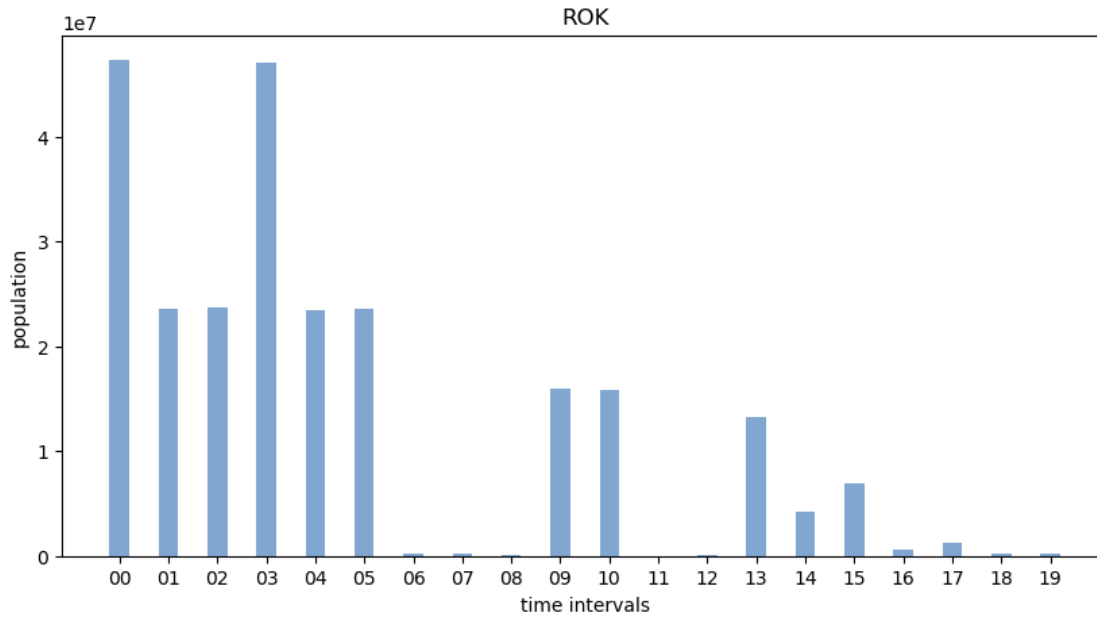
[1 rows x 20 columns]

```

[210]: # 05.02.21-06
        # plot total populations by year
        # combines 19 rows: 01-19
        # assign variable for plot components as _yr_y
        # dt35

dt35_ROK_population_01_id_ttl_yr_x_yr = list(dt35_ROK_population_01_id_ttl_yr_x.
    ↳keys())
dt35_ROK_population_01_id_ttl_yr_x_val =_
    ↳list(dt35_ROK_population_01_id_ttl_yr_x.values())
dt35_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt35_ROK_population_01_id_ttl_yr_x_yr,_
    ↳dt35_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```



```
[211]: # 05.02.21-07
# combine variables into new df
# compiles 19 rows: 01-19
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt36

np.random.seed(123)
dt36_ROK_population_01_id_ttl_yr_x = {
    '00':dt36_ROK_population_01_id_00_flt,
    '01':dt36_ROK_population_01_id_01_flt,
    '02':dt36_ROK_population_01_id_02_flt,
    '03':dt36_ROK_population_01_id_03_flt,
    '04':dt36_ROK_population_01_id_04_flt,
    '05':dt36_ROK_population_01_id_05_flt,
    '06':dt36_ROK_population_01_id_06_flt,
    '07':dt36_ROK_population_01_id_07_flt,
    '08':dt36_ROK_population_01_id_08_flt,
    '09':dt36_ROK_population_01_id_09_flt,
    '10':dt36_ROK_population_01_id_10_flt,
    '11':dt36_ROK_population_01_id_11_flt,
    '12':dt36_ROK_population_01_id_12_flt,
    '13':dt36_ROK_population_01_id_13_flt,
    '14':dt36_ROK_population_01_id_14_flt,
    '15':dt36_ROK_population_01_id_15_flt,
```



```

        '16':dt36_ROK_population_01_id_16_flt,
        '17':dt36_ROK_population_01_id_17_flt,
        '18':dt36_ROK_population_01_id_18_flt,
        '19':dt36_ROK_population_01_id_19_flt
    }
dt36_ROK_population_01_id_ttl_yr_y = pd.
    ↳DataFrame(dt36_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt36_ROK_population_01_id_ttl_yr_y)

```

```

           00           01           02  ...           17           18           19
0  46136101.0  23158582.0  22977519.0  ...  472236.0  401470.0  91814.0

```

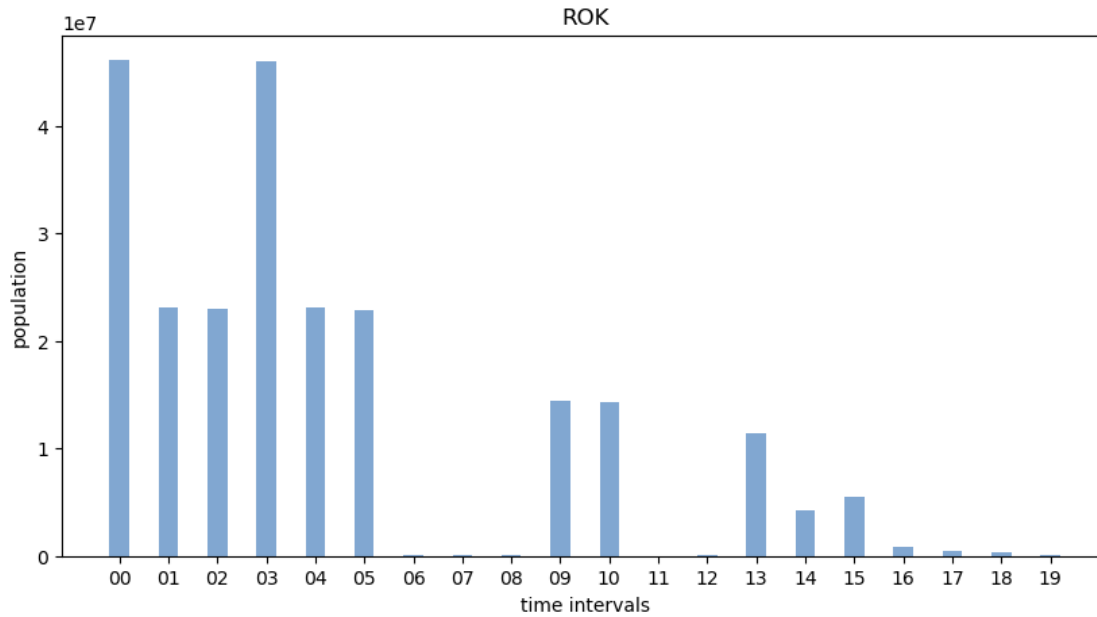
[1 rows x 20 columns]

```

[212]: # 05.02.21-08
# plot total populations by year
# combines 19 rows: 01-19
# assign variable for plot components as _yr_y
# dt36

dt36_ROK_population_01_id_ttl_yr_x_yr = list(dt36_ROK_population_01_id_ttl_yr_x.
    ↳keys())
dt36_ROK_population_01_id_ttl_yr_x_val =
    ↳list(dt36_ROK_population_01_id_ttl_yr_x.values())
dt36_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt36_ROK_population_01_id_ttl_yr_x_yr,
    ↳dt36_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```



```
[213]: # 05.02.21-09
# combine variables into new df
# compiles 16 rows: 01-16
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt37

np.random.seed(123)
dt37_ROK_population_01_id_ttl_yr_x = {
    '00':dt37_ROK_population_01_id_00_flt,
    '01':dt37_ROK_population_01_id_01_flt,
    '02':dt37_ROK_population_01_id_02_flt,
    '03':dt37_ROK_population_01_id_03_flt,
    '04':dt37_ROK_population_01_id_04_flt,
    '05':dt37_ROK_population_01_id_05_flt,
    '06':dt37_ROK_population_01_id_06_flt,
    '07':dt37_ROK_population_01_id_07_flt,
    '08':dt37_ROK_population_01_id_08_flt,
    '09':dt37_ROK_population_01_id_09_flt,
    '10':dt37_ROK_population_01_id_10_flt,
    '11':dt37_ROK_population_01_id_11_flt,
    '12':dt37_ROK_population_01_id_12_flt,
    '13':dt37_ROK_population_01_id_13_flt,
    '14':dt37_ROK_population_01_id_14_flt,
    '15':dt37_ROK_population_01_id_15_flt,
```

```

        '16':dt37_ROK_population_01_id_16_flt
    }
dt37_ROK_population_01_id_ttl_yr_y = pd.
    ↪DataFrame(dt37_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt37_ROK_population_01_id_ttl_yr_y)

```

	00	01	02	...	14	15	16
0	44608726.0	22389324.0	22219402.0	...	55016.0	31972.0	23044.0

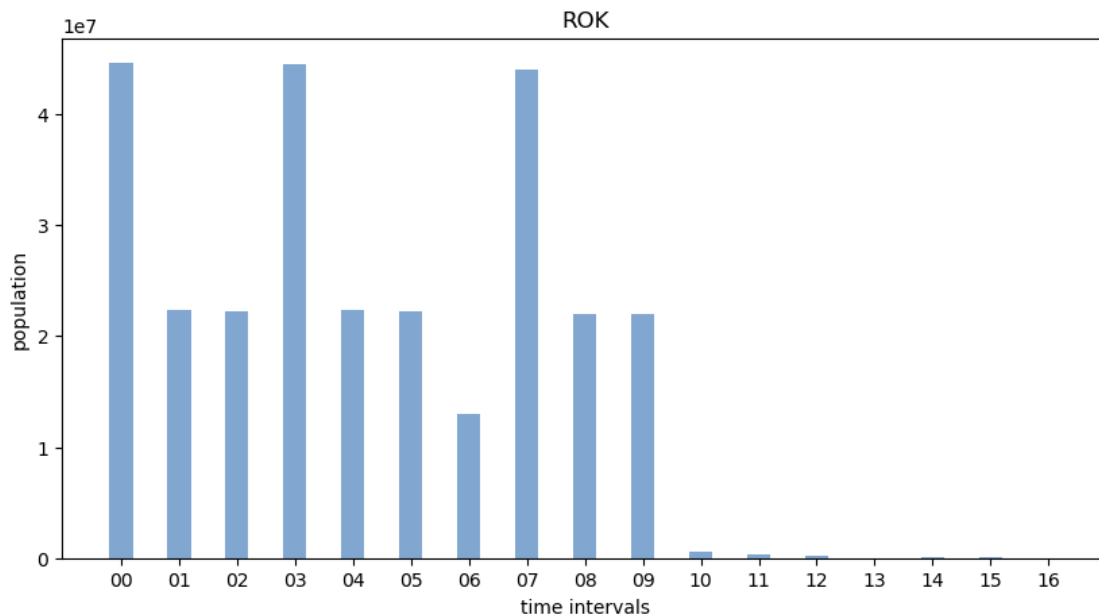
[1 rows x 17 columns]

```

[214]: # 05.02.21-10
# plot total populations by year
# combines 16 rows: 01-16
# assign variable for plot components as _yr_y
# dt37

dt37_ROK_population_01_id_ttl_yr_x_yr = list(dt37_ROK_population_01_id_ttl_yr_x.
    ↪keys())
dt37_ROK_population_01_id_ttl_yr_x_val = ↪
    ↪list(dt37_ROK_population_01_id_ttl_yr_x.values())
dt37_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt37_ROK_population_01_id_ttl_yr_x_yr,↪
    ↪dt37_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```



```
[215]: # 05.02.21-11
# combine variables into new df
# compiles 16 rows: 01-16
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt38

np.random.seed(123)
dt38_ROK_population_01_id_ttl_yr_x = {
    '00':dt38_ROK_population_01_id_00_flt,
    '01':dt38_ROK_population_01_id_01_flt,
    '02':dt38_ROK_population_01_id_02_flt,
    '03':dt38_ROK_population_01_id_03_flt,
    '04':dt38_ROK_population_01_id_04_flt,
    '05':dt38_ROK_population_01_id_05_flt,
    '06':dt38_ROK_population_01_id_06_flt,
    '07':dt38_ROK_population_01_id_07_flt,
    '08':dt38_ROK_population_01_id_08_flt,
    '09':dt38_ROK_population_01_id_09_flt,
    '10':dt38_ROK_population_01_id_10_flt,
    '11':dt38_ROK_population_01_id_11_flt,
    '12':dt38_ROK_population_01_id_12_flt,
    '13':dt38_ROK_population_01_id_13_flt,
    '14':dt38_ROK_population_01_id_14_flt,
    '15':dt38_ROK_population_01_id_15_flt,
    '16':dt38_ROK_population_01_id_16_flt
}
dt38_ROK_population_01_id_ttl_yr_y = pd.
    ↪DataFrame(dt38_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt38_ROK_population_01_id_ttl_yr_y)
```

```

      00      01      02  ...      14      15      16
0  43410899.0  21782154.0  21628745.0  ...  20525.0  11235.0  9290.0
```

```
[1 rows x 17 columns]
```

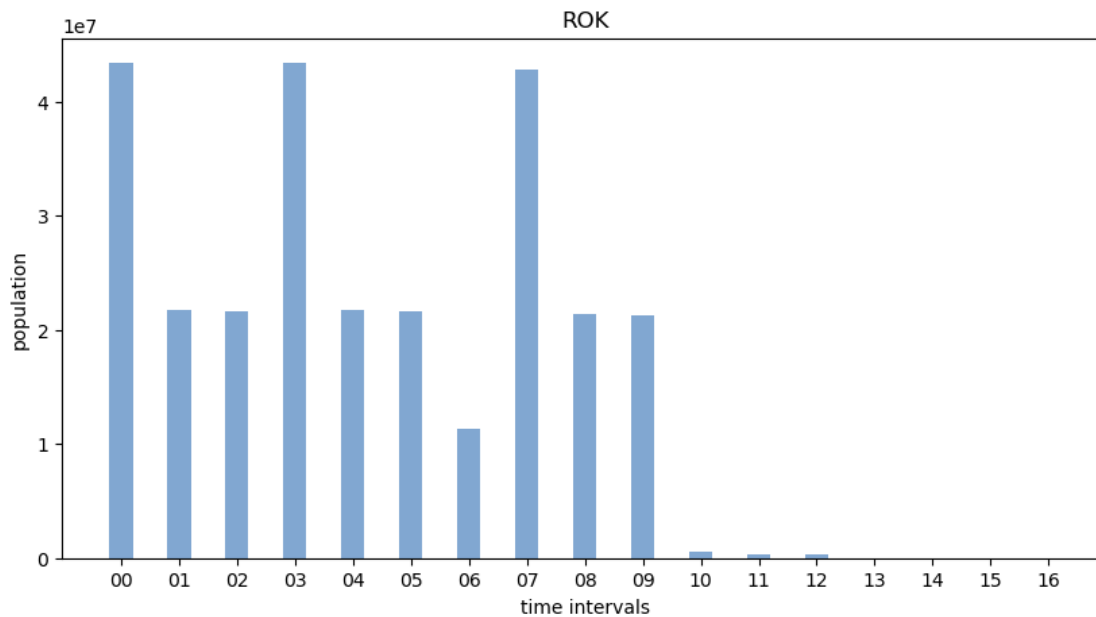
```
[216]: # 05.02.21-12
# plot total populations by year
# combines 16 rows: 01-16
# assign variable for plot components as _yr_y
# dt38

dt38_ROK_population_01_id_ttl_yr_x_yr = list(dt38_ROK_population_01_id_ttl_yr_x.
    ↪keys())
```

```

dt38_ROK_population_01_id_ttl_yr_x_val =
    ↪list(dt38_ROK_population_01_id_ttl_yr_x.values())
dt38_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt38_ROK_population_01_id_ttl_yr_x_yr,
    ↪dt38_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```



```

[217]: # 05.02.21-13
# combine variables into new df
# compiles 16 rows: 01-16
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt39

np.random.seed(123)
dt39_ROK_population_01_id_ttl_yr_x = {
    '00':dt39_ROK_population_01_id_00_flt,
    '01':dt39_ROK_population_01_id_01_flt,
    '02':dt39_ROK_population_01_id_02_flt,
    '03':dt39_ROK_population_01_id_03_flt,
    '04':dt39_ROK_population_01_id_04_flt,
    '05':dt39_ROK_population_01_id_05_flt,

```

```

        '06':dt39_ROK_population_01_id_06_flt,
        '07':dt39_ROK_population_01_id_07_flt,
        '08':dt39_ROK_population_01_id_08_flt,
        '09':dt39_ROK_population_01_id_09_flt,
        '10':dt39_ROK_population_01_id_10_flt,
        '11':dt39_ROK_population_01_id_11_flt,
        '12':dt39_ROK_population_01_id_12_flt,
        '13':dt39_ROK_population_01_id_13_flt,
        '14':dt39_ROK_population_01_id_14_flt,
        '15':dt39_ROK_population_01_id_15_flt,
        '16':dt39_ROK_population_01_id_16_flt
    }
dt39_ROK_population_01_id_ttl_yr_y = pd.
    ↳DataFrame(dt39_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt39_ROK_population_01_id_ttl_yr_y)

```

```

           00           01           02  ...           14           15           16
0  40448486.0  20243765.0  20204721.0  ...  28834.0  16201.0  12633.0

```

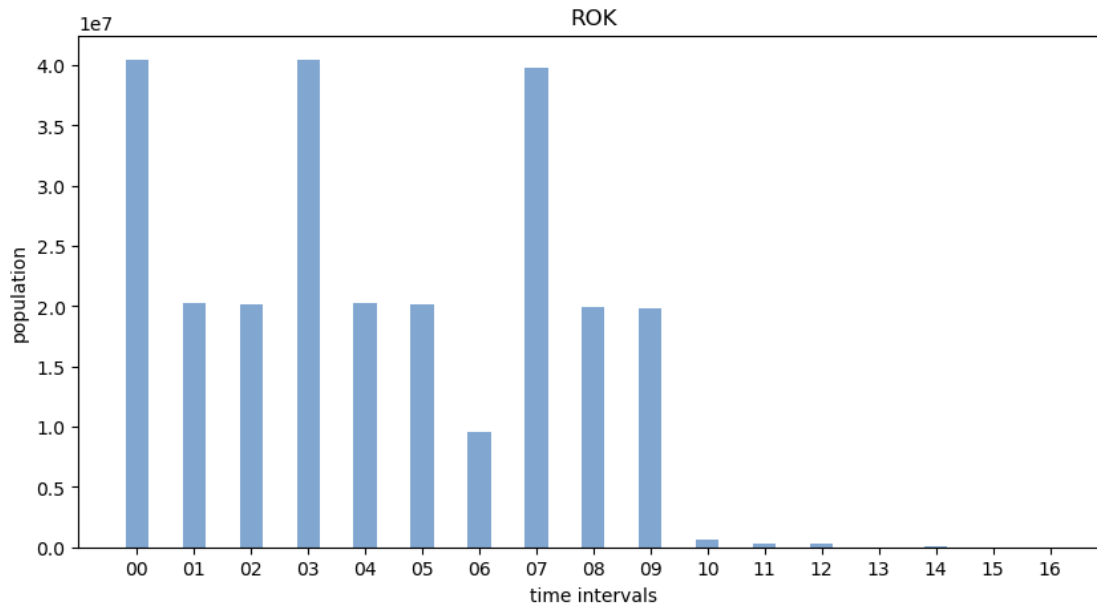
[1 rows x 17 columns]

```

[218]: # 05.02.21-14
        # plot total populations by year
        # combines 16 rows: 01-16
        # assign variable for plot components as _yr_y
        # dt39

dt39_ROK_population_01_id_ttl_yr_x_yr = list(dt39_ROK_population_01_id_ttl_yr_x.
    ↳keys())
dt39_ROK_population_01_id_ttl_yr_x_val =↳
    ↳list(dt39_ROK_population_01_id_ttl_yr_x.values())
dt39_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt39_ROK_population_01_id_ttl_yr_x_yr,↳
    ↳dt39_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```



```
[219]: # 05.02.21-15
# combine variables into new df
# compiles 16 rows: 01-16
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt40

np.random.seed(123)
dt40_ROK_population_01_id_ttl_yr_x = {
    '00':dt40_ROK_population_01_id_00_flt,
    '01':dt40_ROK_population_01_id_01_flt,
    '02':dt40_ROK_population_01_id_02_flt,
    '03':dt40_ROK_population_01_id_03_flt,
    '04':dt40_ROK_population_01_id_04_flt,
    '05':dt40_ROK_population_01_id_05_flt,
    '06':dt40_ROK_population_01_id_06_flt,
    '07':dt40_ROK_population_01_id_07_flt,
    '08':dt40_ROK_population_01_id_08_flt,
    '09':dt40_ROK_population_01_id_09_flt,
    '10':dt40_ROK_population_01_id_10_flt,
    '11':dt40_ROK_population_01_id_11_flt,
    '12':dt40_ROK_population_01_id_12_flt,
    '13':dt40_ROK_population_01_id_13_flt,
    '14':dt40_ROK_population_01_id_14_flt,
    '15':dt40_ROK_population_01_id_15_flt,
```

```

        '16':dt40_ROK_population_01_id_16_flt
    }
dt40_ROK_population_01_id_ttl_yr_y = pd.
    ↪DataFrame(dt40_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt40_ROK_population_01_id_ttl_yr_y)

```

	00	01	02	...	14	15	16
0	37436315.0	18767201.0	18669114.0	...	29500.0	17895.0	11605.0

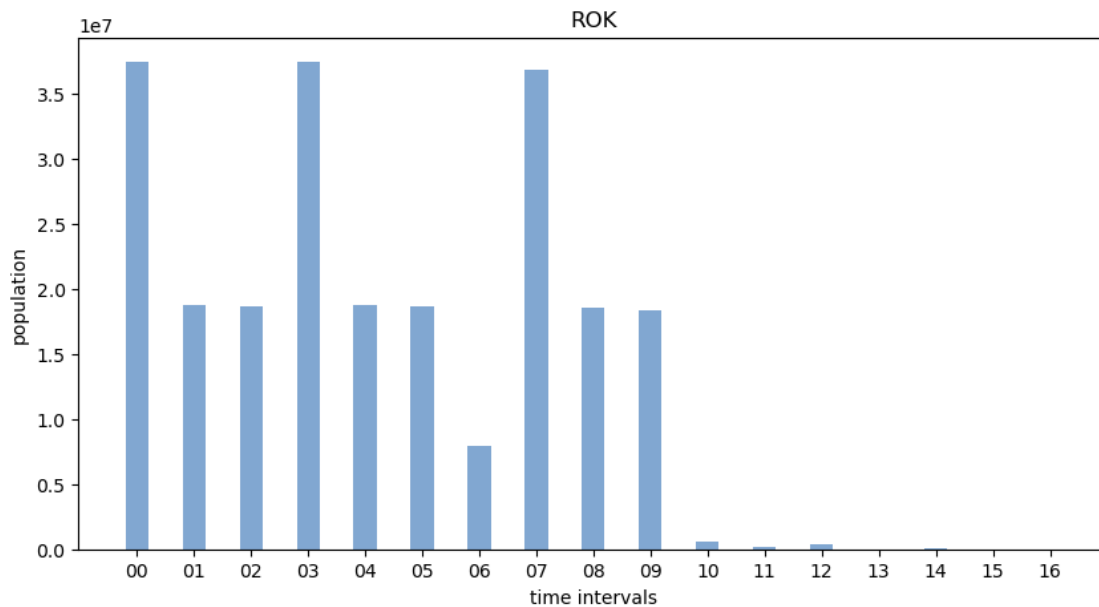
[1 rows x 17 columns]

```

[220]: # 05.02.21-16
# plot total populations by year
# combines 16 rows: 01-16
# assign variable for plot components as _yr_y
# dt40

dt40_ROK_population_01_id_ttl_yr_x_yr = list(dt40_ROK_population_01_id_ttl_yr_x.
    ↪keys())
dt40_ROK_population_01_id_ttl_yr_x_val = ↪
    ↪list(dt40_ROK_population_01_id_ttl_yr_x.values())
dt40_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt40_ROK_population_01_id_ttl_yr_x_yr,↪
    ↪dt40_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```




```
[221]: # 05.02.21-17
# combine variables into new df
# compiles 11 rows: 01-11
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt41

np.random.seed(123)
dt41_ROK_population_01_id_ttl_yr_x = {
    '00':dt41_ROK_population_01_id_00_flt,
    '01':dt41_ROK_population_01_id_01_flt,
    '02':dt41_ROK_population_01_id_02_flt,
    '03':dt41_ROK_population_01_id_03_flt,
    '04':dt41_ROK_population_01_id_04_flt,
    '05':dt41_ROK_population_01_id_05_flt,
    '06':dt41_ROK_population_01_id_06_flt,
    '07':dt41_ROK_population_01_id_07_flt,
    '08':dt41_ROK_population_01_id_08_flt,
    '09':dt41_ROK_population_01_id_09_flt,
    '10':dt41_ROK_population_01_id_10_flt,
    '11':dt41_ROK_population_01_id_11_flt
}
dt41_ROK_population_01_id_ttl_yr_y = pd.
↳DataFrame(dt41_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt41_ROK_population_01_id_ttl_yr_y)
```

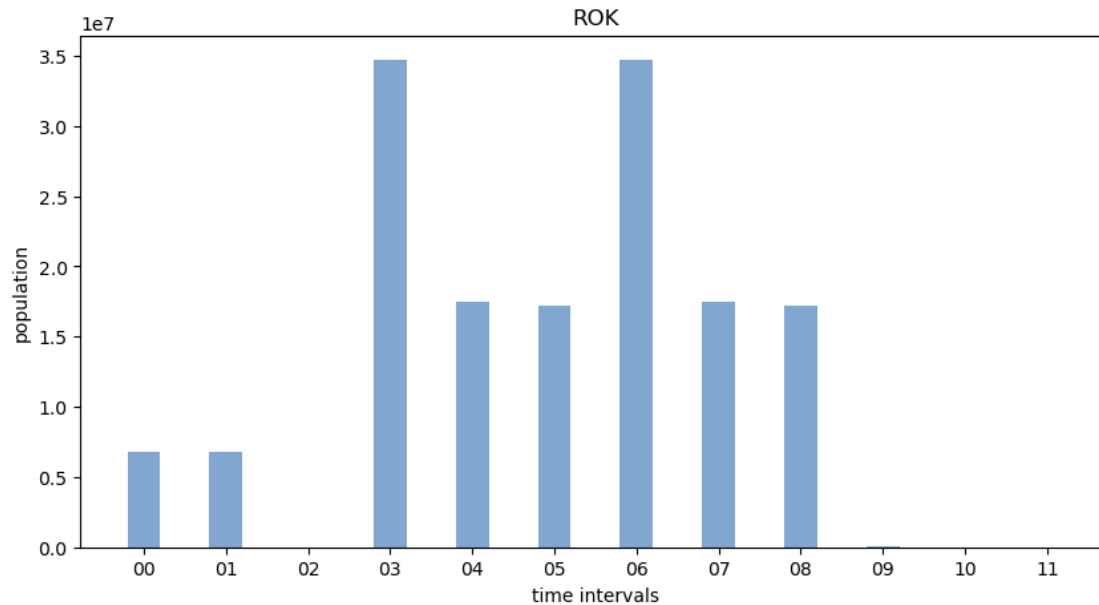
	00	01	02	...	09	10	11
0	6761239.0	6754257.0	6982.0	...	27648.0	15998.0	11650.0

[1 rows x 12 columns]

```
[222]: # 05.02.21-18
# plot total populations by year
# combines 11 rows: 01-11
# assign variable for plot components as _yr_y
# dt41

dt41_ROK_population_01_id_ttl_yr_x_yr = list(dt41_ROK_population_01_id_ttl_yr_x.
↳keys())
dt41_ROK_population_01_id_ttl_yr_x_val =↳
↳list(dt41_ROK_population_01_id_ttl_yr_x.values())
dt41_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt41_ROK_population_01_id_ttl_yr_x_yr,↳
↳dt41_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
```

```
plt.ylabel("population")
plt.title("ROK")
plt.show()
```



```
[223]: # 05.02.21-19
# combine variables into new df
# compiles 11 rows: 01-11
# assign variable for new dataset as _yr_x
# add variable 0 for total
# return df
# dt42

np.random.seed(123)
dt42_ROK_population_01_id_ttl_yr_x = {
    '00':dt42_ROK_population_01_id_00_flt,
    '01':dt42_ROK_population_01_id_01_flt,
    '02':dt42_ROK_population_01_id_02_flt,
    '03':dt42_ROK_population_01_id_03_flt,
    '04':dt42_ROK_population_01_id_04_flt,
    '05':dt42_ROK_population_01_id_05_flt,
    '06':dt42_ROK_population_01_id_06_flt,
    '07':dt42_ROK_population_01_id_07_flt,
    '08':dt42_ROK_population_01_id_08_flt,
    '09':dt42_ROK_population_01_id_09_flt,
    '10':dt42_ROK_population_01_id_10_flt,
    '11':dt42_ROK_population_01_id_11_flt
}
```

```

    }
dt42_ROK_population_01_id_ttl_yr_y = pd.
    ↪DataFrame(dt42_ROK_population_01_id_ttl_yr_x, index = [0])
print(dt42_ROK_population_01_id_ttl_yr_y)

```

```

      00      01      02  ...      09      10      11
0  5863440.0  5856901.0  6539.0  ...  30402.0  16062.0  14340.0

```

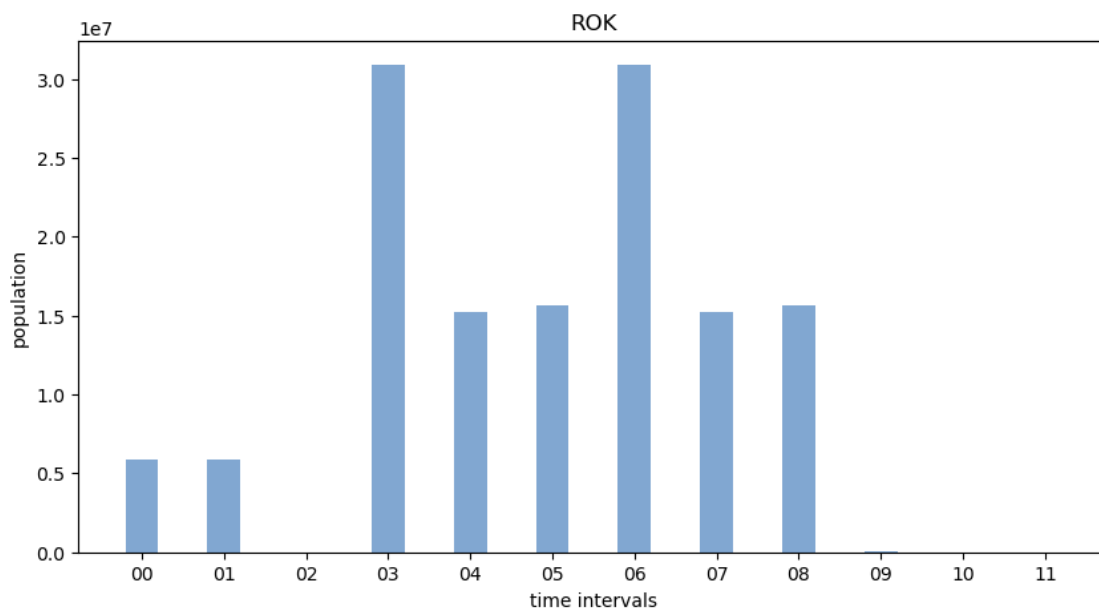
[1 rows x 12 columns]

```

[224]: # 05.02.21-20
# plot total populations by year
# combines 11 rows: 01-11
# assign variable for plot components as _yr_y
# dt42

dt42_ROK_population_01_id_ttl_yr_x_yr = list(dt42_ROK_population_01_id_ttl_yr_x.
    ↪keys())
dt42_ROK_population_01_id_ttl_yr_x_val =
    ↪list(dt42_ROK_population_01_id_ttl_yr_x.values())
dt42_ROK_population_01_id_ttl_yr_x_fig = plt.figure(figsize = (10, 5))
plt.bar(dt42_ROK_population_01_id_ttl_yr_x_yr,
    ↪dt42_ROK_population_01_id_ttl_yr_x_val, color = '#81A7D1', width = 0.4)
plt.xlabel("time intervals")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```



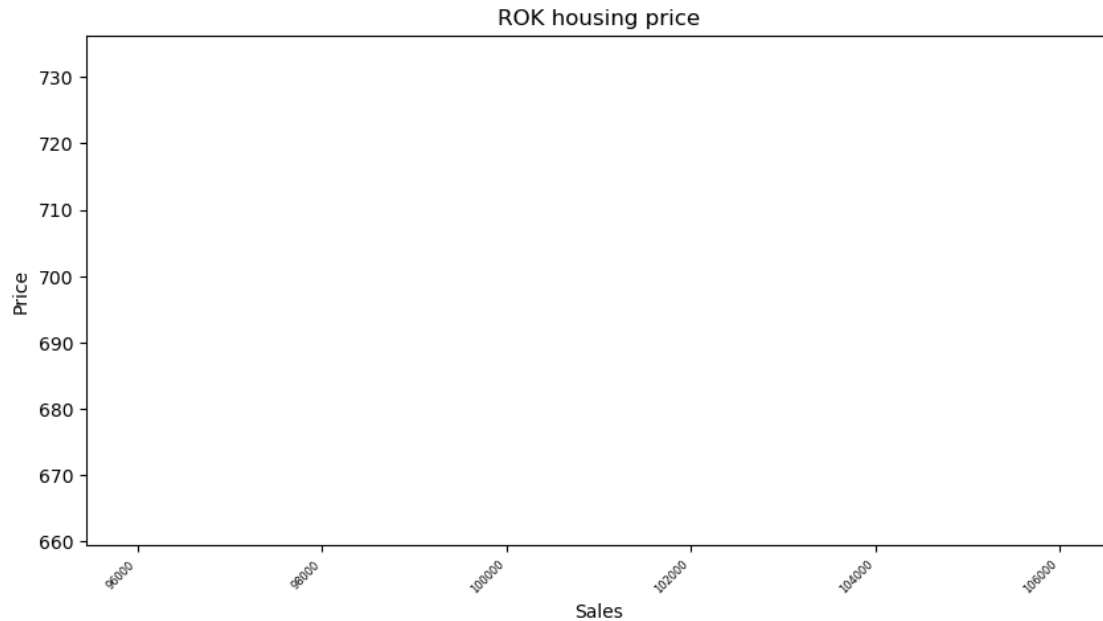
```
[225]: # 05.02.22-01
# return real estate trend to date
# assign variable for plot components as _00
# dt43
```

```
print(dt43_Seoul_apartment_01_id)
```

```
dt13_00          0
index            0
name
gugun
dong
buildDate        200006.0
min_sales        97000.0
max_sales        107000.0
avg_sales        101000.0
area             139
floor            42
pricePerArea     697.841727
Name: 0, dtype: object
```

```
[226]: # 05.02.22-02
# plot population trend to date
# assign variable for plot components as _00
# dt43
```

```
dt43_Seoul_apartment_01_id_x = dt43_Seoul_apartment_01_id['avg_sales']
dt43_Seoul_apartment_01_id_y = dt43_Seoul_apartment_01_id['pricePerArea']
dt43_Seoul_apartment_01_id_fig = plt.figure(figsize = (10, 5))
plt.plot(dt43_Seoul_apartment_01_id_x, dt43_Seoul_apartment_01_id_y, '-.')
plt.xticks(
    rotation = 45,
    horizontalalignment = 'right',
    fontweight = 'light',
    fontsize = 6
)
plt.xlabel("Sales")
plt.ylabel("Price")
plt.title("ROK housing price")
plt.show()
```



```
[227]: # 05.02.23-01
# plot total populations by year
# combine variables into new df
# compiles 9 datasets: dt42-dt33
# assign variable for new dataset as _ttl_x
# return df
# dt42-dt33

np.random.seed(123)
dt00_ROK_population_ttl_y = {
    '1970':dt42_ROK_population_01_id_ttl,
    '1975':dt41_ROK_population_01_id_ttl,
    '1980':dt40_ROK_population_01_id_ttl,
    '1985':dt39_ROK_population_01_id_ttl,
    '1990':dt38_ROK_population_01_id_ttl,
    '1995':dt37_ROK_population_01_id_ttl,
    '2000':dt36_ROK_population_01_id_ttl,
    '2005':dt35_ROK_population_01_id_ttl,
    '2010':dt34_ROK_population_01_id_ttl,
    '2022':dt33_ROK_population_01_id_ttl
}
dt00_ROK_population_ttl_x = pd.DataFrame(dt00_ROK_population_ttl_y, index = [0])
print(dt00_ROK_population_ttl_x)
```

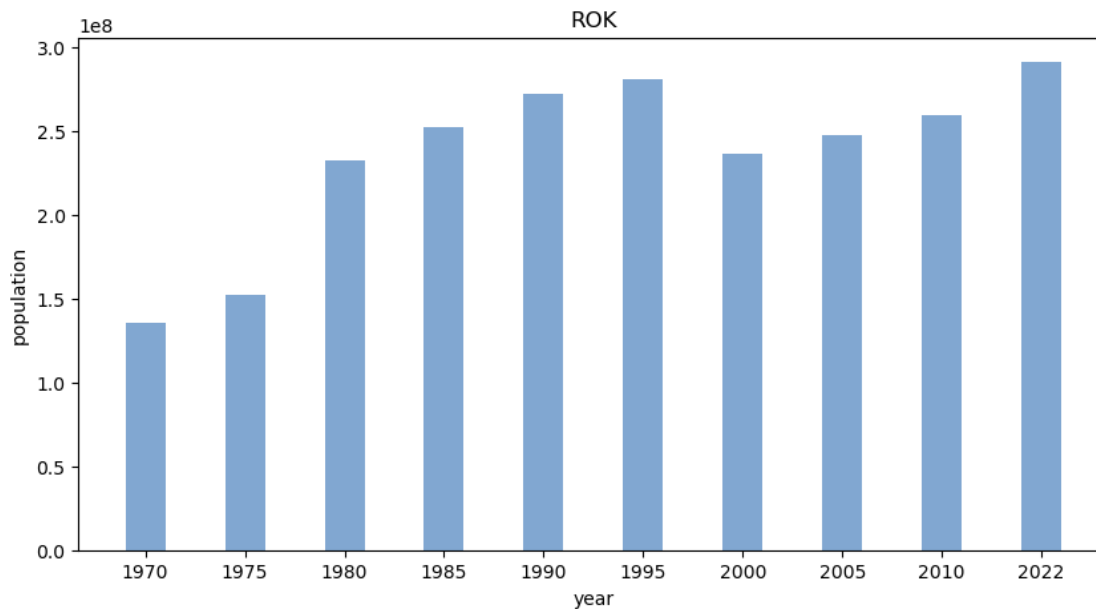
	1970	1975	1980	...	2005	2010
2022						
0	135256424.0	152348958.0	232537495.0	...	247744145.0	259127241.0

291008943.0

[1 rows x 10 columns]

```
[228]: # 05.02.23-02
# plot total populations by year
# combines 9 datasets: dt42-dt33
# assign variable for plot components as _ttl_y
# dt42-dt33

dt00_ROK_population_ttl_y_yr = list(dt00_ROK_population_ttl_y.keys())
dt00_ROK_population_ttl_y_val = list(dt00_ROK_population_ttl_y.values())
dt00_ROK_population_ttl_y_fig = plt.figure(figsize = (10, 5))
plt.bar(dt00_ROK_population_ttl_y_yr, dt00_ROK_population_ttl_y_val, color='
↳'#81A7D1', width = 0.4)
plt.xlabel("year")
plt.ylabel("population")
plt.title("ROK")
plt.show()
```



```
[229]: # 05.02.23-03
# combine variables into new df
# compiles 9 datasets: dt42-dt33
# assign variable for new dataset as _x
# return df
# dt42-dt33
```

```

np.random.seed(123)
dt00_ROK_population_ttl_mn_y = {
    '1970':dt42_ROK_population_01_id_ttl_x,
    '1975':dt41_ROK_population_01_id_ttl_x,
    '1980':dt40_ROK_population_01_id_ttl_x,
    '1985':dt39_ROK_population_01_id_ttl_x,
    '1990':dt38_ROK_population_01_id_ttl_x,
    '1995':dt37_ROK_population_01_id_ttl_x,
    '2000':dt36_ROK_population_01_id_ttl_x,
    '2005':dt35_ROK_population_01_id_ttl_x,
    '2010':dt34_ROK_population_01_id_ttl_x,
    '2022':dt33_ROK_population_01_id_ttl_x
}
dt00_ROK_population_ttl_mn_x = pd.DataFrame(dt00_ROK_population_ttl_mn_y, index_
    ↪= [0])
print(dt00_ROK_population_ttl_mn_x)

```

```

          1970          1975  ...          2010          2022
0  1.127137e+07  12695746.5  ...  12956362.05  14550447.15

```

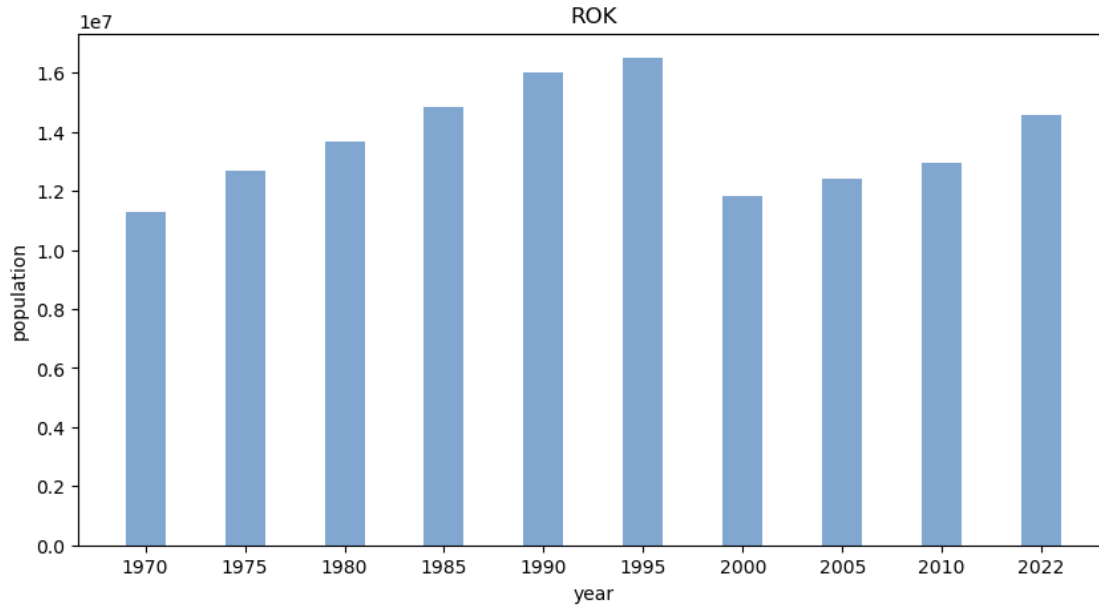
[1 rows x 10 columns]

```

[230]: # 05.02.23-04
# plot mean populations by year
# combines 9 datasets: dt42-dt33
# assign variable for plot components as _mn_y
# dt42-dt33

dt00_ROK_population_ttl_mn_y_yr = list(dt00_ROK_population_ttl_mn_y.keys())
dt00_ROK_population_ttl_mn_y_val = list(dt00_ROK_population_ttl_mn_y.values())
dt00_ROK_population_ttl_mn_y_fig = plt.figure(figsize = (10, 5))
plt.bar(dt00_ROK_population_ttl_mn_y_yr, dt00_ROK_population_ttl_mn_y_val,
    ↪color = '#81A7D1', width = 0.4)
plt.xlabel("year")
plt.ylabel("population")
plt.title("ROK")
plt.show()

```



```
[231]: # 05.02.24-01
# combine variables into new df
# compiles 9 datasets: dt00
# assign variable for new dataset as _00
# return df
# dt42-dt33

np.random.seed(123)
dt00_ROK_population_ttl_00_y = {
    '1970':dt42_ROK_population_01_id_00_flt,
    '1975':dt41_ROK_population_01_id_00_flt,
    '1980':dt40_ROK_population_01_id_00_flt,
    '1985':dt39_ROK_population_01_id_00_flt,
    '1990':dt38_ROK_population_01_id_00_flt,
    '1995':dt37_ROK_population_01_id_00_flt,
    '2000':dt36_ROK_population_01_id_00_flt,
    '2005':dt35_ROK_population_01_id_00_flt,
    '2010':dt34_ROK_population_01_id_00_flt,
    '2022':dt33_ROK_population_01_id_00_flt
}
dt00_ROK_population_ttl_00_x = pd.DataFrame(dt00_ROK_population_ttl_00_y, index_
↪=[0])
print(dt00_ROK_population_ttl_00_x)
```

	1970	1975	1980	...	2005	2010	2022
0	5863440.0	6761239.0	37436315.0	...	47278951.0	48580293.0	51692272.0

[1 rows x 10 columns]

```
[232]: # 05.02.24-02
# create vertices for total population
# assign variable for new dataset as _00
# return df
# dt42-dt33

dt00_ROK_population_ttl_00_x = [
    ['1970', dt42_ROK_population_01_id_00_flt],
    ['1975', dt41_ROK_population_01_id_00_flt],
    ['1980', dt40_ROK_population_01_id_00_flt],
    ['1985', dt39_ROK_population_01_id_00_flt],
    ['1990', dt38_ROK_population_01_id_00_flt],
    ['1995', dt37_ROK_population_01_id_00_flt],
    ['2000', dt36_ROK_population_01_id_00_flt],
    ['2005', dt35_ROK_population_01_id_00_flt],
    ['2010', dt34_ROK_population_01_id_00_flt],
    ['2022', dt33_ROK_population_01_id_00_flt]
]
dt00_ROK_population_ttl_00_x_y = pd.DataFrame(dt00_ROK_population_ttl_00_x,
    ↪columns = ['Year', 'Population'])
```

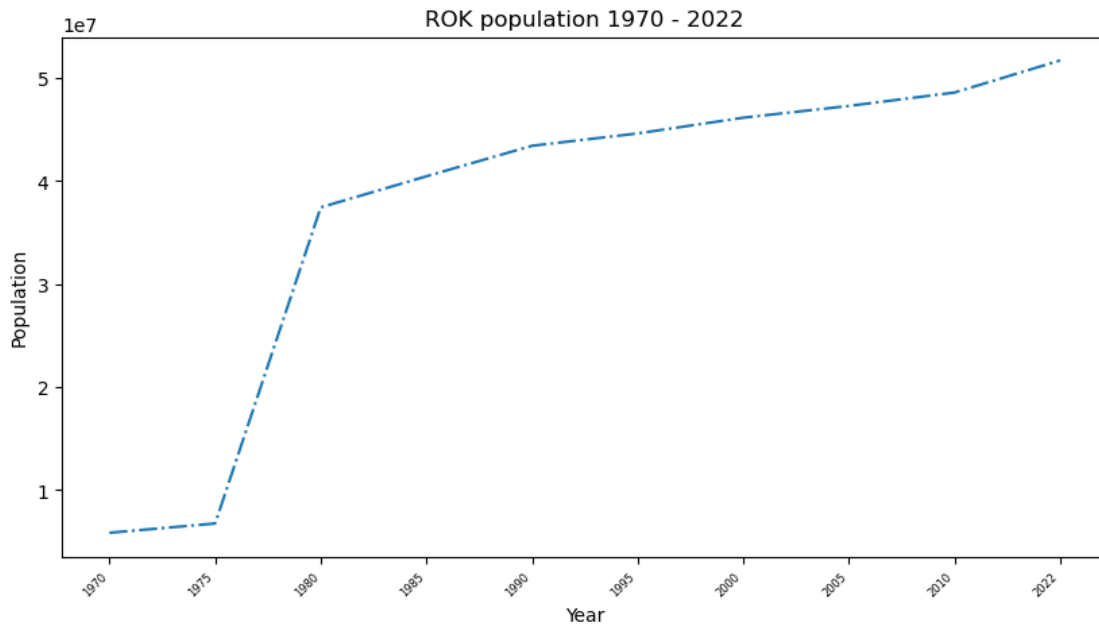
```
[233]: # 05.02.25-01
# split data into a training and test set
# target = Year_2022
# Year_2022 = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign regression variable
# for dataset dt_00 containing population data
# dt42-dt33

dt00_ROK_population_ttl_00_x_dv_lr01 = LinearRegression()
```

```
[234]: # 05.02.25-02
# plot population future projection
# assign variable for plot components as _32
# the population is expected to peak at 2024
# dt32

dt32_ROK_population_01_seoul_drp_id_dt_00_x =
    ↪dt00_ROK_population_ttl_00_x_y['Year']
dt32_ROK_population_01_seoul_drp_id_dt_00_y =
    ↪dt00_ROK_population_ttl_00_x_y['Population']
dt32_ROK_population_01_seoul_drp_id_dt_00_fig = plt.figure(figsize = (10, 5))
plt.plot(dt32_ROK_population_01_seoul_drp_id_dt_00_x,
    ↪dt32_ROK_population_01_seoul_drp_id_dt_00_y, '-.')
```

```
plt.xticks(
    rotation = 45,
    horizontalalignment = 'right',
    fontweight = 'light',
    fontsize = 6
)
plt.xlabel("Year")
plt.ylabel("Population")
plt.title("ROK population 1970 - 2022")
plt.show()
```



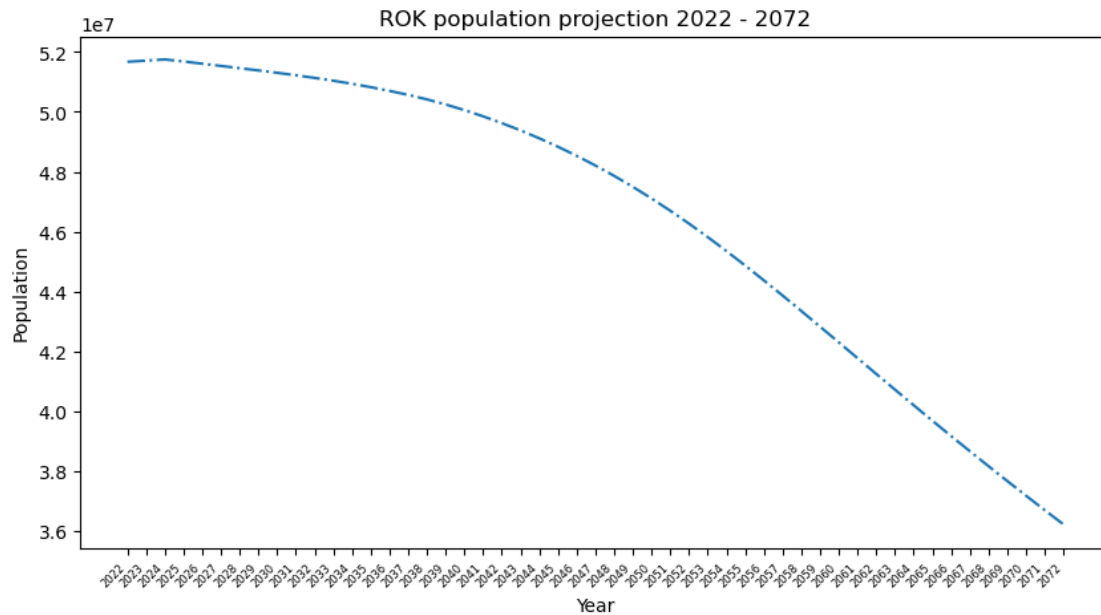
```
[235]: # 05.02.25-03
# plot total populations by year
# combines 9 datasets: dt32
# assign variable for plot components as _32
# the population is expected steadily decline from 2024
# dt32

dt00_ROK_population_ttl_00_x_y_x = ␣
    ↳ dt32_ROK_population_01_seoul_drp_id_dt_00['Year']
dt00_ROK_population_ttl_00_x_y_y = ␣
    ↳ dt32_ROK_population_01_seoul_drp_id_dt_00['Population']
dt00_ROK_population_ttl_00_x_y_fig = plt.figure(figsize = (10, 5))
plt.plot(dt00_ROK_population_ttl_00_x_y_x, dt00_ROK_population_ttl_00_x_y_y, '-.
    ↳ ')
plt.xticks(
```

```

rotation = 45,
horizontalalignment = 'right',
fontweight = 'light',
fontsize = 6
)
plt.xlabel("Year")
plt.ylabel("Population")
plt.title("ROK population projection 2022 - 2072")
plt.show()

```



```

[236]: # 05.02.25-04
# create dummy variables
# due to returning boolean values, converting dummies to integers -----
# for dataset dt31 containing population data
# dt31

dt31_ROK_demo_____02_dv = pd.get_dummies(dt31_ROK_demo_____02, drop_first =
↪ True, dtype=int)

```

```

[237]: # 05.02.25-05
# Split the data into a training and test set
# target = 2072
# 2072 = variable 01
# select columns
# for dataset dt31 containing population data
# dt31

```

```
dt31_ROK_demo_____02_dv_x01 = dt31_ROK_demo_____02_dv.drop(['2072'], axis = 1)
dt31_ROK_demo_____02_dv_y01 = dt31_ROK_demo_____02_dv['2072']
```

```
[238]: # 05.02.25-06
# Split the data into a training and test set
# target = 2072
# 2072 = variable 01
# split into train and test
# for dataset dt31 containing population data
# dt31

dt31_ROK_demo_____02_dv_x01_trn, dt31_ROK_demo_____02_dv_x01_tst,
dt31_ROK_demo_____02_dv_y01_trn, dt31_ROK_demo_____02_dv_y01_tst =
train_test_split(dt31_ROK_demo_____02_dv_x01,
dt31_ROK_demo_____02_dv_y01, test_size = 0.3, random_state = 0)
```

```
[239]: # 05.02.25-07
# Split the data into a training and test set
# target = 2072
# 2072 = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign regression variable
# for dataset dt31 containing population data
# dt31

dt31_ROK_demo_____02_dv_lr01 = LinearRegression()
```

```
[240]: # 05.02.25-08
# Split the data into a training and test set
# target = 2072
# 2072 = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# fit variables to model
# for dataset dt31 containing population data
# dt31

dt31_ROK_demo_____02_dv_lr01.fit(dt31_ROK_demo_____02_dv_x01_trn,
dt31_ROK_demo_____02_dv_y01_trn)
```

```
[240]: LinearRegression()
```

```
[241]: # 05.02.25-09
# Split the data into a training and test set
# target = 2072
# 2072 = variable 01
```

```

# Run a linear regression and report the R2-value and RMSE on the test set.
# fit variables to model
# for dataset dt31 containing population data
# dt31

dt31_ROK_demo_____02_dv_pdct = dt31_ROK_demo_____02_dv_lr01.
  ↳predict(dt31_ROK_demo_____02_dv_x01_tst)

```

```

[242]: # 05.02.25-10
# Split the data into a training and test set
# target = 2072
# 2072 = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign variable for rmse and r2
# for dataset dt31 containing population data
# dt31

dt31_ROK_demo_____02_dv_rmse01 = np.
  ↳sqrt(mean_squared_error(dt31_ROK_demo_____02_dv_y01_tst,↳
  ↳dt31_ROK_demo_____02_dv_pdct))
dt31_ROK_demo_____02_dv_r201 = r2_score(dt31_ROK_demo_____02_dv_y01_tst,↳
  ↳dt31_ROK_demo_____02_dv_pdct)

```

```

[243]: # 05.02.25-11
# Split the data into a training and test set
# target = 2072
# 2072 = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# return rmse and r2 dt04
# rmse: 10.53
# r2: 1.0
# rmse is significantly lower than other models but still high
# r2 is ideal
# this is reflective of the population trend
# for dataset dt31 containing population data
# dt31

print(f'rmse: {dt31_ROK_demo_____02_dv_rmse01}')
print(f'r2: {dt31_ROK_demo_____02_dv_r201}')

```

```

rmse: 10.533955178045936
r2: 0.9999999999972669

```

```

[244]: # 05.02.26-01
# return dt43
# dt43

```

```
dt43_Seoul_apartment_01
```

```
[244]:
```

	id	index	name	gugun	...	avg_sales	area	floor	pricePerArea
0	0	0		...	101000.0	139	42	697.841727	
1	1	1		...	89000.0	105	32	790.476191	
2	2	2	2	...	71500.0	86	26	720.930233	
3	3	3		...	65500.0	102	31	637.254902	
4	4	4		...	70000.0	91	28	692.307692	
...	
4539	4539	46		...	99000.0	113	34	876.106195	
4540	4540	47		...	89500.0	80	24	1000.000000	
4541	4541	48		...	84500.0	83	25	891.566265	
4542	4542	49		...	109000.0	93	28	1043.010753	
4543	4543	50		...	98500.0	117	35	717.948718	

[4544 rows x 12 columns]

```
[245]: # 05.02.26-02
# sort by build date
# dt43

dt43_Seoul_apartment_01_sort = dt43_Seoul_apartment_01.sort_values(by =
↳ 'buildDate')
dt43_Seoul_apartment_01_sort
```

```
[245]:
```

	id	index	name	gugun	...	avg_sales	area	floor	pricePerArea
2935	2935	62		...	57000.0	44	13	1204.545455	
4440	4440	62		...	57000.0	44	13	1204.545455	
2565	2565	62		...	57000.0	44	13	1204.545455	
530	530	5		...	23000.0	26	8	884.615385	
684	684	5		...	23000.0	26	8	884.615385	
...	
4254	4254	48	13	...	NaN	0	0	NaN	
4344	4344	138	1	...	NaN	0	0	NaN	
4452	4452	74		...	NaN	0	0	NaN	
4460	4460	82		...	NaN	0	0	NaN	
4490	4490	112		...	NaN	0	0	NaN	

[4544 rows x 12 columns]

```
[246]: # 05.02.26-03
# remove NaN values
# dt43

dt43_Seoul_apartment_01_sort.dropna(inplace = True)
dt43_Seoul_apartment_01_sort
```

```
[246]:
```

	id	index	name	gugun	...	avg_sales	area	floor	pricePerArea
	2935	2935	62		...	57000.0	44	13	1204.545455
	4440	4440	62		...	57000.0	44	13	1204.545455
	2565	2565	62		...	57000.0	44	13	1204.545455
	530	530	5		...	23000.0	26	8	884.615385
	684	684	5		...	23000.0	26	8	884.615385
...
	2344	2344	13		...	130000.0	62	19	1370.967742
	4288	4288	82		...	103000.0	90	27	1000.000000
	2413	2413	82		...	103000.0	90	27	1000.000000
	4116	4116	82		...	103000.0	90	27	1000.000000
	1748	1748	115		...	42500.0	16	5	1750.000000

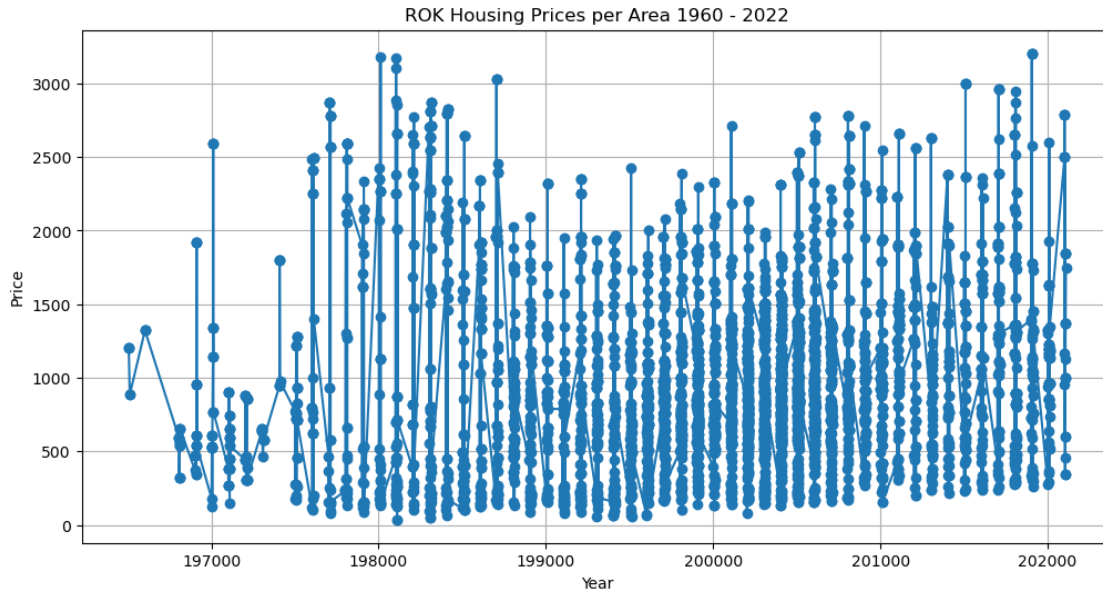
[4333 rows x 12 columns]

```
[247]: # 05.02.26-04
# change year to integer
# change price to float
# dt43

dt43_Seoul_apartment_01_sort_x = dt43_Seoul_apartment_01_sort['buildDate'].
    ↳astype(int)
dt43_Seoul_apartment_01_sort_y = dt43_Seoul_apartment_01_sort['pricePerArea'].
    ↳astype(float)
```

```
[248]: # 05.02.26-05
# plot annual housing prices
# housing prices increased in the 80s and is now increasing again in the present
# dt43

plt.figure(figsize = (12, 6))
plt.plot(dt43_Seoul_apartment_01_sort_x, dt43_Seoul_apartment_01_sort_y, marker = 'o',
    ↳linestyle = '-')
plt.title('ROK Housing Prices per Area 1960 - 2022')
plt.xlabel('Year')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```



```
[249]: # 05.02.26-06
# create dummy variables
# due to returning boolean values, converting dummies to integers -----
# dt43

dt43_Seoul_apartment_01_sort_dv = pd.get_dummies(dt43_Seoul_apartment_01_sort,
↳ drop_first = True, dtype=int)
```

```
[250]: # 05.02.26-07
# Split the data into a training and test set
# target = pricePerArea
# pricePerArea = variable 01
# select columns
# for dataset dt43 containing price data
# dt43

dt43_Seoul_apartment_01_sort_dv_x01 = dt43_Seoul_apartment_01_sort_dv.
↳ drop(['pricePerArea'], axis = 1)
dt43_Seoul_apartment_01_sort_dv_y01 =
↳ dt43_Seoul_apartment_01_sort_dv['pricePerArea']
```

```
[251]: # 05.02.26-08
# Split the data into a training and test set
# target = pricePerArea
# pricePerArea = variable 01
# split into train and test
# for dataset dt43 containing price data
```



```
# dt43

dt43_Seoul_apr_____01_sort_dv_x01_trn, dt43_Seoul_apr_____01_sort_dv_x01_tst,
↳dt43_Seoul_apr_____01_sort_dv_y01_trn,
↳dt43_Seoul_apr_____01_sort_dv_y01_tst =
↳train_test_split(dt43_Seoul_apr_____01_sort_dv_x01,
↳dt43_Seoul_apr_____01_sort_dv_y01, test_size = 0.3, random_state = 0)
```

```
[252]: # 05.02.26-09
# Split the data into a training and test set
# target = pricePerArea
# pricePerArea = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign regression variable
# for dataset dt43 containing price data
# dt43
```

```
dt43_Seoul_apr_____01_sort_dv_lr01 = LinearRegression()
```

```
[253]: # 05.02.26-10
# Split the data into a training and test set
# target = pricePerArea
# pricePerArea = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# fit variables to model
# for dataset dt43 containing price data
# dt43
```

```
dt43_Seoul_apr_____01_sort_dv_lr01.fit(dt43_Seoul_apr_____01_sort_dv_x01_trn,
↳dt43_Seoul_apr_____01_sort_dv_y01_trn)
```

```
[253]: LinearRegression()
```

```
[254]: # 05.02.26-11
# Split the data into a training and test set
# target = pricePerArea
# pricePerArea = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# fit variables to model
# for dataset dt43 containing price data
# dt43
```

```
dt43_Seoul_apr_____01_sort_dv_pdct = dt43_Seoul_apr_____01_sort_dv_lr01.
↳predict(dt43_Seoul_apr_____01_sort_dv_x01_tst)
```

```
[255]: # 05.02.26-12
# Split the data into a training and test set
```

```

# target = pricePerArea
# pricePerArea = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign variable for rmse and r2
# for dataset dt43 containing price data
# dt43

dt43_Seoul_apartment_01_sort_dv_rmse01 = np.
    ↪sqrt(mean_squared_error(dt43_Seoul_apartment_01_sort_dv_y01_tst,
    ↪dt43_Seoul_apartment_01_sort_dv_pdct))
dt43_Seoul_apartment_01_sort_dv_r201 =
    ↪r2_score(dt43_Seoul_apartment_01_sort_dv_y01_tst,
    ↪dt43_Seoul_apartment_01_sort_dv_pdct)

```

```

[256]: # 05.02.26-13
# Split the data into a training and test set
# target = pricePerArea
# pricePerArea = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# return rmse and r2 dt04
# rmse: 7476990.22
# r2: -139790867.78
# rmse is very high
# r2 is very low
# this is reflective of the nature of the housing industry
# prices fluctuate and vary
# for dataset dt43 containing price data
# dt43

print(f'rmse: {dt43_Seoul_apartment_01_sort_dv_rmse01}')
print(f'r2: {dt43_Seoul_apartment_01_sort_dv_r201}')

```

rmse: 3991658.719437283
r2: -39841159.86725998

```

[257]: # 05.02.26-14
# assign variable for pca
# pricePerArea = variable 01
# for dataset dt43 containing price data
# dt43

pca = PCA(.9)

```

```

[258]: # 05.02.26-15
# calculate pca
# pricePerArea = variable 01
# for dataset dt43 containing price data

```

```
# dt43

pca.fit(dt43_Seoul_apartment_01_sort_dv_x01_trn)
dt43_Seoul_apartment_01_sort_dv_x01_trn_pca = pca.
    ↪transform(dt43_Seoul_apartment_01_sort_dv_x01_trn)
dt43_Seoul_apartment_01_sort_dv_x01_tst_pca = pca.
    ↪transform(dt43_Seoul_apartment_01_sort_dv_x01_tst)
```

```
[259]: # 05.02.26-16
# return pca calculation matrix
# pricePerArea = variable 01
# for dataset dt43 containing price data
# dt43

print(f'features in pca matrix: {dt43_Seoul_apartment_01_sort_dv_x01_trn_pca.
    ↪shape[1]}')
```

features in pca matrix: 1

```
[260]: # 05.02.26-17
# remove null
# return dimensions
# dt43

dt43_Seoul_apartment_01_sort.isnull().sum() * 100 / dt43_Seoul_apartment_01_sort.
    ↪shape[0]
```

```
[260]: id          0.0
index         0.0
name          0.0
gugun         0.0
dong          0.0
buildDate     0.0
min_sales     0.0
max_sales     0.0
avg_sales     0.0
area          0.0
floor         0.0
pricePerArea  0.0
dtype: float64
```

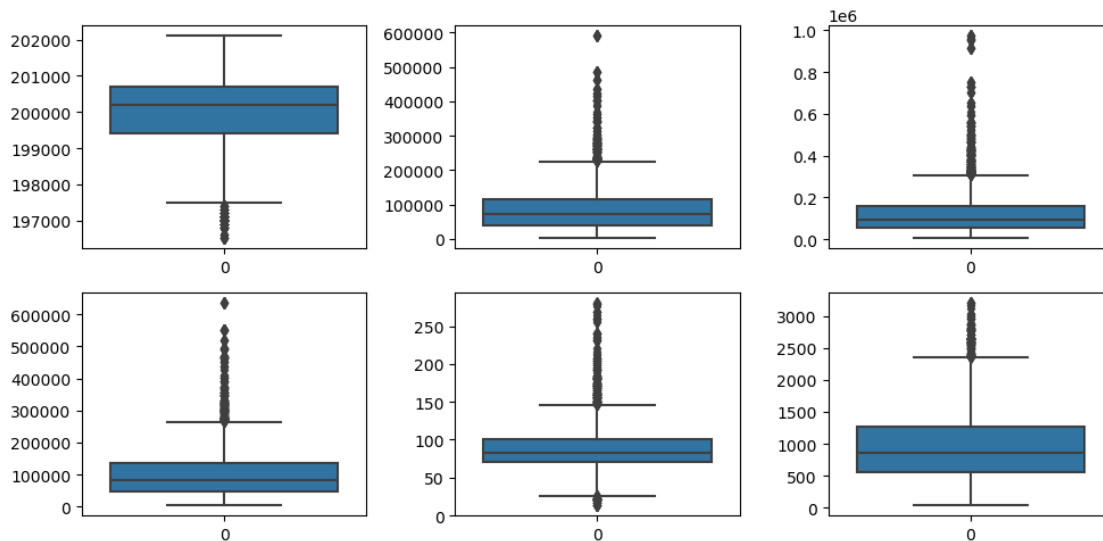
```
[261]: # 05.02.26-17
# subplot
# plot columns
# return dimensions
# dt43

fig, axs = plt.subplots(2,3, figsize = (10,5))
```

```

dt43_Seoul_apartment_01_sort_plt01 = sns.
    ↳boxplot(dt43_Seoul_apartment_01_sort['buildDate'], ax = axs[0,0])
dt43_Seoul_apartment_01_sort_plt02 = sns.
    ↳boxplot(dt43_Seoul_apartment_01_sort['min_sales'], ax = axs[0,1])
dt43_Seoul_apartment_01_sort_plt03 = sns.
    ↳boxplot(dt43_Seoul_apartment_01_sort['max_sales'], ax = axs[0,2])
dt43_Seoul_apartment_01_sort_plt04 = sns.
    ↳boxplot(dt43_Seoul_apartment_01_sort['avg_sales'], ax = axs[1,0])
dt43_Seoul_apartment_01_sort_plt05 = sns.
    ↳boxplot(dt43_Seoul_apartment_01_sort['area'], ax = axs[1,1])
dt43_Seoul_apartment_01_sort_plt06 = sns.
    ↳boxplot(dt43_Seoul_apartment_01_sort['pricePerArea'], ax = axs[1,2])
plt.tight_layout()

```

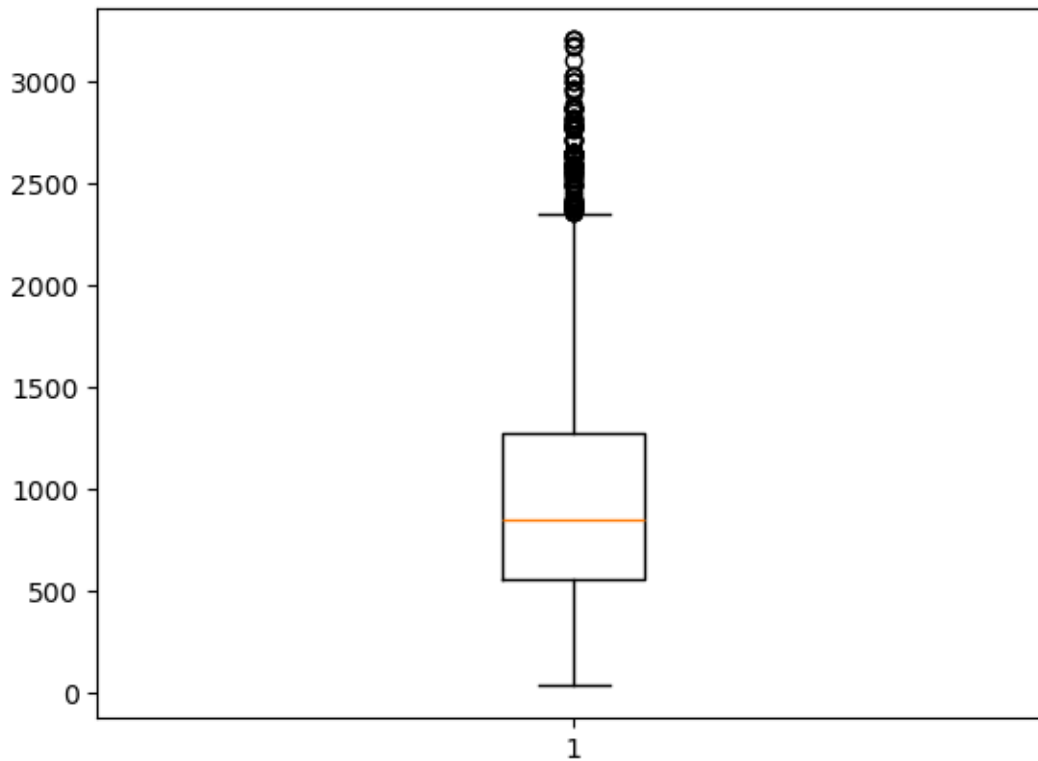


```

[262]: # 05.02.26-18
# plot price per area
# first and third quantiles
# dt43

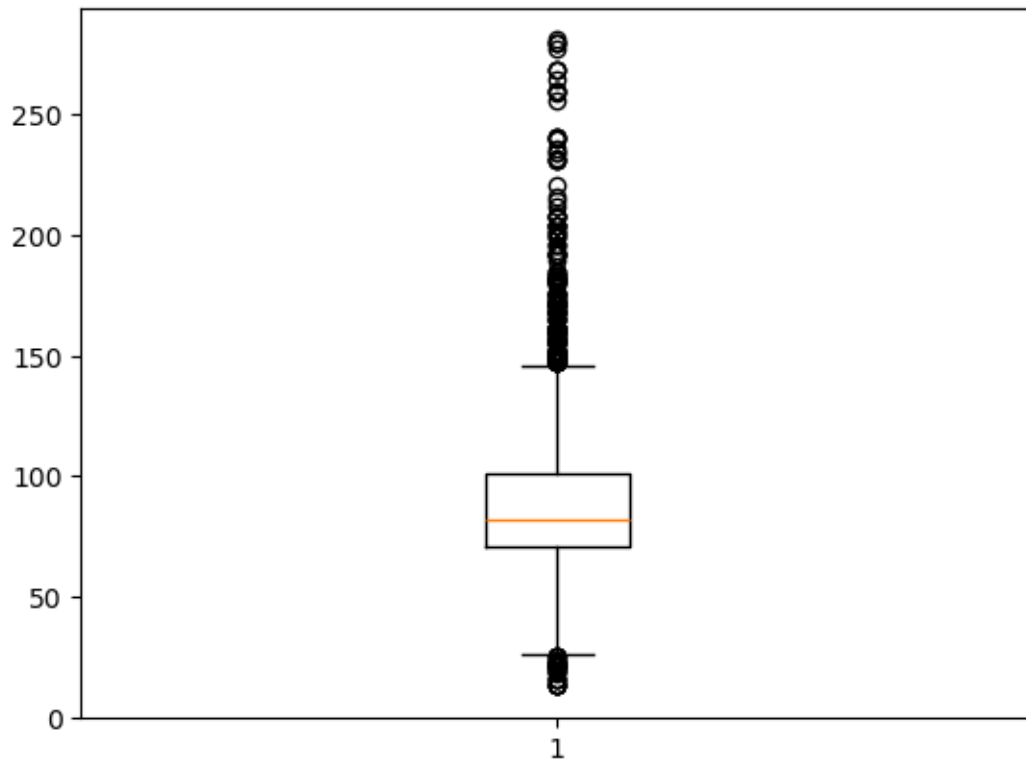
plt.boxplot(dt43_Seoul_apartment_01_sort.pricePerArea)
Q1 = dt43_Seoul_apartment_01_sort.pricePerArea.quantile(0.25)
Q3 = dt43_Seoul_apartment_01_sort.pricePerArea.quantile(0.75)
IQR = Q3 - Q1
dt43_Seoul_apartment_01_sort_ppa =
    ↳dt43_Seoul_apartment_01_sort[(dt43_Seoul_apartment_01_sort.pricePerArea >= Q1
    ↳- 1.5 * IQR) & (dt43_Seoul_apartment_01_sort.pricePerArea <= Q3 + 1.5 * IQR)]

```



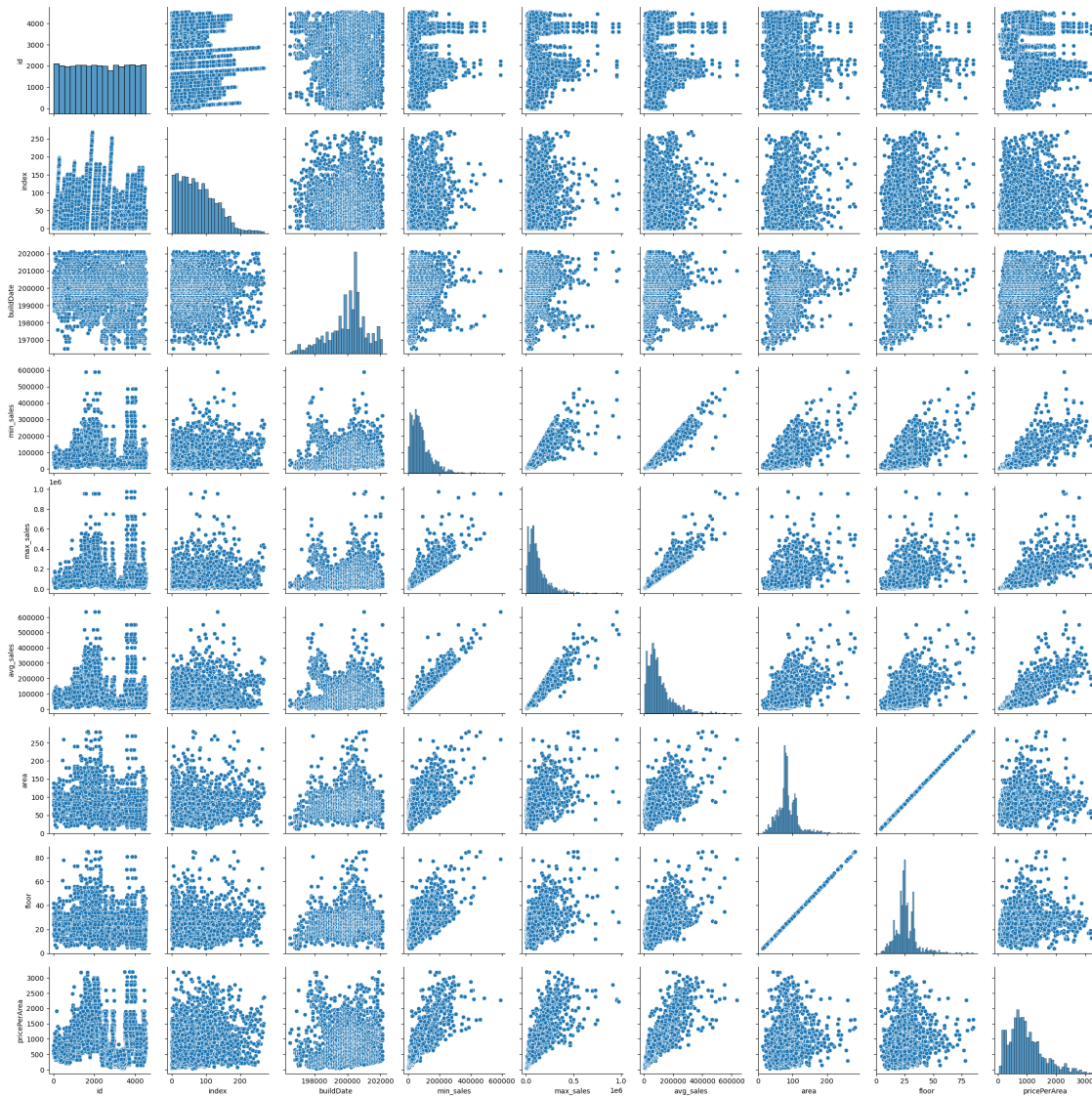
```
[263]: # 05.02.26-19
# plot price per area
# first and third quantiles
# outlier treatment for area
# dt43

plt.boxplot(dt43_Seoul_apartment_01_sort.area)
Q1 = dt43_Seoul_apartment_01_sort.area.quantile(0.25)
Q3 = dt43_Seoul_apartment_01_sort.area.quantile(0.75)
IQR = Q3 - Q1
dt43_Seoul_apartment_01_sort_area =
    dt43_Seoul_apartment_01_sort[(dt43_Seoul_apartment_01_sort.area >= Q1 - 1.5 * IQR) & (dt43_Seoul_apartment_01_sort.area <= Q3 + 1.5 * IQR)]
```



```
[264]: # 05.02.26-20
# plot pair plot
# dt43

sns.pairplot(dt43_Seoul_apr____01_sort)
plt.show()
```



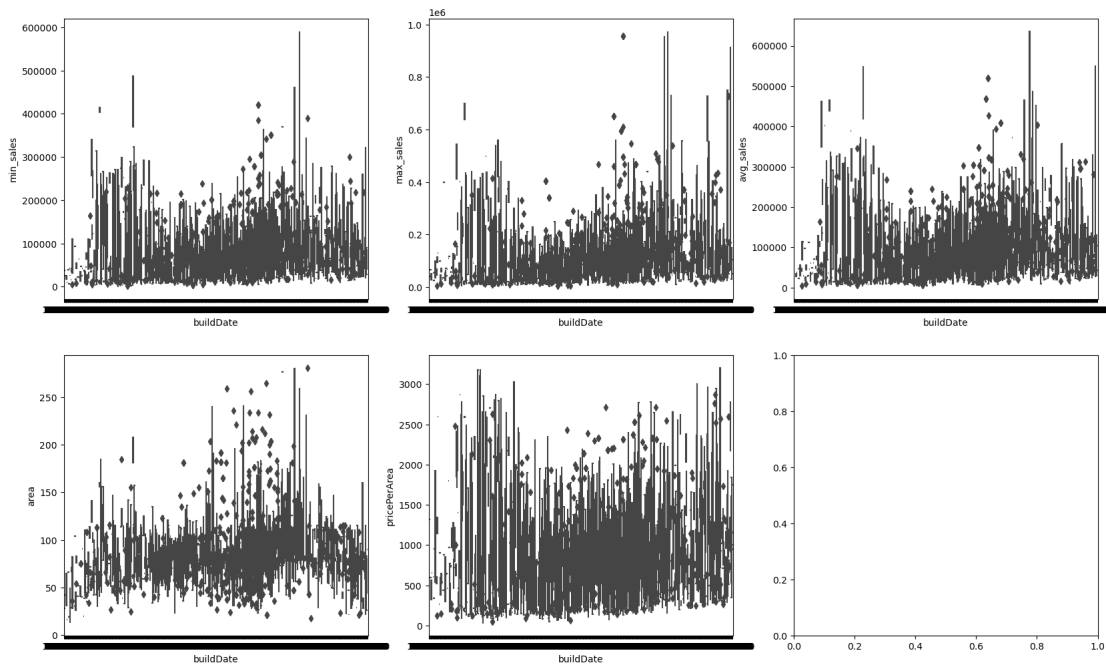
```
[265]: # 05.02.26-21
# plot five columns
# x = build date
# dt43

plt.figure(figsize = (20, 12))
plt.subplot(2,3,1)
sns.boxplot(x = 'buildDate', y = 'min_sales', data = dt43_Seoul_apartment_01_sort)
plt.subplot(2,3,2)
sns.boxplot(x = 'buildDate', y = 'max_sales', data = dt43_Seoul_apartment_01_sort)
```

```

plt.subplot(2,3,3)
sns.boxplot(x = 'buildDate', y = 'avg_sales', data = dt43_Seoul_apartment_01_sort)
plt.subplot(2,3,4)
sns.boxplot(x = 'buildDate', y = 'area', data = dt43_Seoul_apartment_01_sort)
plt.subplot(2,3,5)
sns.boxplot(x = 'buildDate', y = 'pricePerArea', data = dt43_Seoul_apartment_01_sort)
plt.subplot(2,3,6)
plt.show()

```



```

[266]: # 05.02.26-23
# get the dummy variables for feature 'pricePerArea'
# store dummies in a new variable - 'dt43_Seoul_apartment_01_sort_dv'
# dt43

dt43_Seoul_apartment_01_sort_dv = pd.
    ↳ get_dummies(dt43_Seoul_apartment_01_sort['pricePerArea'])

```

```

[267]: # 05.02.26-24
# concat dummies
# dt43

dt43_Seoul_apartment_01_sort_conc = pd.concat([dt43_Seoul_apartment_01_sort,
    ↳ dt43_Seoul_apartment_01_sort_dv], axis = 1)

```



```
[268]: # 05.02.26-25
# return first ten rows of 'dt43_Seoul_apr_____01_sort_dv'
# dt43

dt43_Seoul_apr_____01_sort_dv.head(10)
```

```
[268]:      36.363636      47.272727      ...      3177.419355      3204.545455
2935      False      False      ...      False      False
4440      False      False      ...      False      False
2565      False      False      ...      False      False
530       False      False      ...      False      False
684       False      False      ...      False      False
3638      False      False      ...      False      False
4016      False      False      ...      False      False
3863      False      False      ...      False      False
2564      False      False      ...      False      False
4439      False      False      ...      False      False
```

[10 rows x 2081 columns]

```
[269]: # 05.02.26-26
# split into train and test data
# set same rows
# split 75 / 25
# dt43

np.random.seed(0)
dt43_Seoul_apr_____01_sort_dv_trn, dt43_Seoul_apr_____01_sort_dv_tst = \
    ↪train_test_split(dt43_Seoul_apr_____01_sort, train_size = 0.75, test_size = \
    ↪0.25, random_state = 100)
```

```
[270]: # 05.02.26-27
# assign scaler
# dt43

sclr = MinMaxScaler()
```

```
[271]: # 05.02.26-28
# fit to model
# dt43

dt43_Seoul_apr_____01_sort_dv_var = ['buildDate', 'min_sales', 'max_sales', \
    ↪'avg_sales', 'area', 'pricePerArea']
dt43_Seoul_apr_____01_sort_dv_trn[dt43_Seoul_apr_____01_sort_dv_var] = sclr.
    ↪fit_transform(dt43_Seoul_apr_____01_sort_dv_trn[dt43_Seoul_apr_____01_sort_dv_var])
```

```
[272]: # 05.02.26-29
# return first ten rows
# dt43
```

```
dt43_Seoul_apartment_01_sort.head(10)
```

```
[272]:
```

	id	index	name	gugun	...	avg_sales	area	floor	pricePerArea
2935	2935	62		...		57000.0	44	13	1204.545455
4440	4440	62		...		57000.0	44	13	1204.545455
2565	2565	62		...		57000.0	44	13	1204.545455
530	530	5		...		23000.0	26	8	884.615385
684	684	5		...		23000.0	26	8	884.615385
3638	3638	135		...		37000.0	28	8	1321.428571
4016	4016	135		...		37000.0	28	8	1321.428571
3863	3863	135		...		37000.0	28	8	1321.428571
2564	2564	61		...		30500.0	42	13	595.238095
4439	4439	61		...		30500.0	42	13	595.238095

```
[10 rows x 12 columns]
```

```
[273]: # 05.02.26-30
# omit three columns
# dt43
```

```
dt43_Seoul_apartment_01_sort.drop(['name'], axis = 1, inplace = True)
dt43_Seoul_apartment_01_sort.drop(['gugun'], axis = 1, inplace = True)
dt43_Seoul_apartment_01_sort.drop(['dong'], axis = 1, inplace = True)
```

```
[274]: dt43_Seoul_apartment_01_sort.head()
```

```
[274]:
```

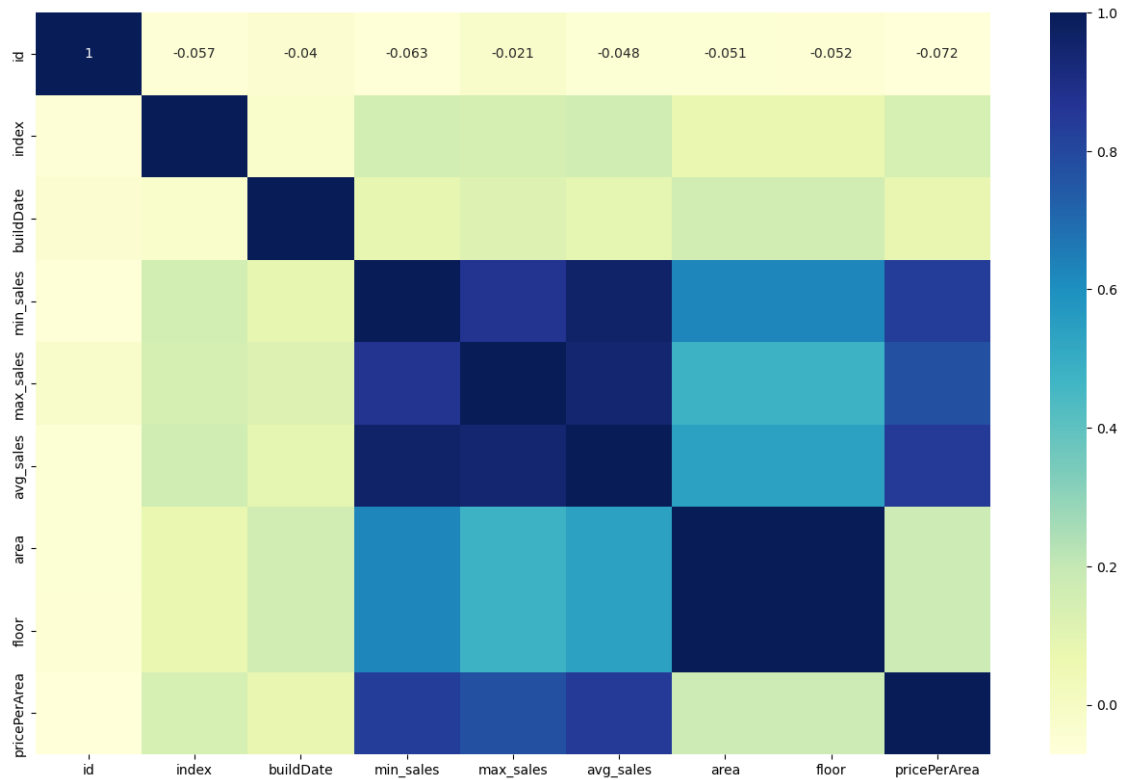
	id	index	buildDate	min_sales	...	avg_sales	area	floor	pricePerArea
2935	2935	62	196506.0	53000.0	...	57000.0	44	13	1204.545455
4440	4440	62	196506.0	53000.0	...	57000.0	44	13	1204.545455
2565	2565	62	196506.0	53000.0	...	57000.0	44	13	1204.545455
530	530	5	196512.0	23000.0	...	23000.0	26	8	884.615385
684	684	5	196512.0	23000.0	...	23000.0	26	8	884.615385

```
[5 rows x 9 columns]
```

```
[275]: # 05.02.26-31
# plot heat map
```

```
# dt43

plt.figure(figsize = (16, 10))
sns.heatmap(dt43_Seoul_apt_____01_sort.corr(), annot = True, cmap="YlGnBu")
plt.show()
```



```
[276]: # 05.02.27-01
# return first and last ten rows
# return columns
# dt09

print(dt09_jeonse_sales___01.head(10))
print(dt09_jeonse_sales___01.tail(10))
print(dt09_jeonse_sales___01.columns)
```

	type	region_01	...	2024_02	2024_03
0	Total	The Whole Country	...	87.6	88.3
1	Total	Seoul Metropolitan Area	...	86.7	87.9
2	Total	Non-Seoul Metropolitan Area	...	88.4	88.6
3	Total	6 Large Cities	...	84.2	84.8
4	Total	5 Large Cities	...	83.5	84.1
5	Total	9 Provinces	...	89.9	90.1
6	Total	8 Provinces	...	91.8	91.8

7	Total	Seoul	...	85.1	87.8
8	Total	Seoul	...	83.3	86.3
9	Total	Seoul	...	87.3	88.8

[10 rows x 10 columns]

	type	region_01	region_02	...	2024_01	2024_02	2024_03
154	Detached Houses	Ulsan	Sub Summary	...	91.6	92.5	93.1
155	Detached Houses	Sejong	Sub Summary	...	101.4	102.9	103.6
156	Detached Houses	Gangwon	Sub Summary	...	99.1	100.4	101.9
157	Detached Houses	Chungbuk	Sub Summary	...	97.8	96.2	99.6
158	Detached Houses	Chungnam	Sub Summary	...	99.5	98.8	99.0
159	Detached Houses	Jeonbuk	Sub Summary	...	92.2	93.4	92.7
160	Detached Houses	Jeonnam	Sub Summary	...	98.7	100.1	100.6
161	Detached Houses	Gyeongbuk	Sub Summary	...	100.8	100.6	100.9
162	Detached Houses	Gyeongnam	Sub Summary	...	97.0	96.1	97.0
163	Detached Houses	Jeju	Sub Summary	...	95.1	94.1	93.3

[10 rows x 10 columns]

```
Index(['type', 'region_01', 'region_02', 'region_03', '2023_10', '2023_11',
      '2023_12', '2024_01', '2024_02', '2024_03'],
      dtype='object')
```

```
[277]: # 05.02.27-02
# omit all rows except 0
# rows omitted because of irrelevance
# dt09

dt09_jeonse_sales__01_row = dt09_jeonse_sales__01.iloc[0:1]
dt09_jeonse_sales__01_row
```

```
[277]:      type      region_01  region_02 ... 2024_01 2024_02 2024_03
0  Total  The Whole Country  Sub Summary ...   87.7   87.6   88.3
```

[1 rows x 10 columns]

```
[278]: # 05.02.27-03
# omit four columns
# rows omitted because of irrelevance
# dt09

dt09_jeonse_sales__01_row.drop(['type'], axis = 1, inplace = True)
dt09_jeonse_sales__01_row.drop(['region_01'], axis = 1, inplace = True)
dt09_jeonse_sales__01_row.drop(['region_02'], axis = 1, inplace = True)
dt09_jeonse_sales__01_row.drop(['region_03'], axis = 1, inplace = True)
dt09_jeonse_sales__01_row
```

```
[278]:      2023_10 2023_11 2023_12 2024_01 2024_02 2024_03
0      90.7   89.3   87.9   87.7   87.6   88.3
```

```
[279]: # 05.02.27-04
# convert all values to floats
# dt09

dt09_jeonse_sales___01_row['2023_10'].astype(float)
dt09_jeonse_sales___01_row['2023_11'].astype(float)
dt09_jeonse_sales___01_row['2023_12'].astype(float)
dt09_jeonse_sales___01_row['2024_01'].astype(float)
dt09_jeonse_sales___01_row['2024_02'].astype(float)
dt09_jeonse_sales___01_row['2024_03'].astype(float)
```

```
[279]: 0      88.3
      Name: 2024_03, dtype: float64
```

```
[280]: # 05.02.27-04
# create new data frame
# dt09

dt09_jeonse_sales___01_row_lst = [
    ['2023.10', 90.7],
    ['2023.11', 89.3],
    ['2023.12', 87.9],
    ['2024.01', 87.7],
    ['2024.02', 87.6],
    ['2024.03', 88.3]
]

dt09_jeonse_sales___01_row_lst_df = pd.
↳DataFrame(dt09_jeonse_sales___01_row_lst, columns = ['Year', 'Price'])
dt09_jeonse_sales___01_row_lst_df
```

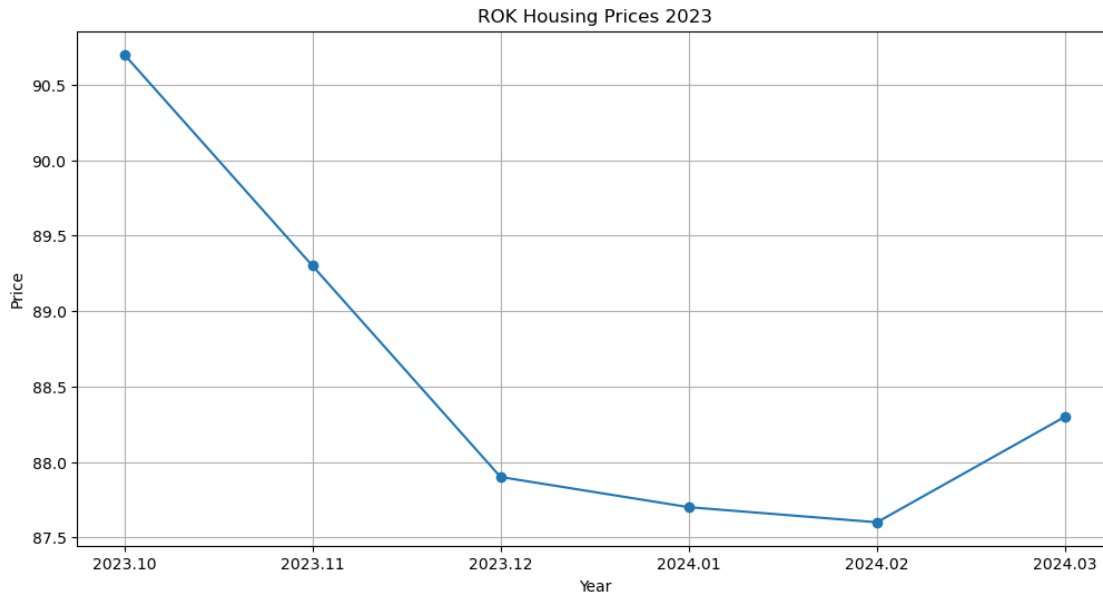
```
[280]:      Year  Price
0  2023.10   90.7
1  2023.11   89.3
2  2023.12   87.9
3  2024.01   87.7
4  2024.02   87.6
5  2024.03   88.3
```

```
[281]: # 05.02.27-05
# plot annual housing prices
# housing prices increased in the 80s and is now increasing again in the present
# dt43

plt.figure(figsize = (12, 6))
plt.plot(dt09_jeonse_sales___01_row_lst_df['Year'],
↳dt09_jeonse_sales___01_row_lst_df['Price'], marker = 'o', linestyle = '-')

```

```
plt.title('ROK Housing Prices 2023')
plt.xlabel('Year')
plt.ylabel('Price')
plt.grid(True)
plt.show()
```



```
[282]: # 05.02.28-01
# return first and last ten rows
# return columns
# dt44

print(dt44_ROK_population_01.head(10))
print(dt44_ROK_population_01.tail(10))
print(dt44_ROK_population_01.columns)
```

	Date	Region	...	Natural_growth	Natural_growth_rate
0	1/1/2000	Busan	...	1877.0	5.8
1	1/1/2000	Chungcheongbuk-do	...	979.0	7.7
2	1/1/2000	Chungcheongnam-do	...	932.0	5.7
3	1/1/2000	Daegu	...	1940.0	9.1
4	1/1/2000	Daejeon	...	1294.0	11.2
5	1/1/2000	Gangwon-do	...	899.0	6.8
6	1/1/2000	Gwangju	...	1553.0	13.5
7	1/1/2000	Gyeonggi-do	...	9757.0	12.9
8	1/1/2000	Gyeongsangbuk-do	...	1132.0	4.8
9	1/1/2000	Gyeongsangnam-do	...	1803.0	6.9

[10 rows x 12 columns]

	Date	Region	...	Natural_growth	Natural_growth_rate
4850	6/1/2022	Gyeongsangbuk-do	...	-1042.0	-4.9
4851	6/1/2022	Gyeongsangnam-do	...	-809.0	-3.0
4852	6/1/2022	Incheon	...	-195.0	-0.8
4853	6/1/2022	Jeju	...	-92.0	-1.7
4854	6/1/2022	Jeollabuk-do	...	-632.0	-4.3
4855	6/1/2022	Jeollanam-do	...	-805.0	-5.4
4856	6/1/2022	Sejong	...	141.0	4.5
4857	6/1/2022	Seoul	...	-494.0	-0.6
4858	6/1/2022	Ulsan	...	9.0	0.1
4859	6/1/2022	Whole country	...	-6019.0	-1.4

[10 rows x 12 columns]

```
Index(['Date', 'Region', 'Birth', 'Birth_rate', 'Death', 'Death_rate',
      'Divorce', 'Divorce_rate', 'Marriage', 'Marriage_rate',
      'Natural_growth', 'Natural_growth_rate'],
      dtype='object')
```

```
[283]: # 05.02.28-02
# convert integer date column to dates in international format
# dt44

dt44_R0K_population_01['Date'] = pd.to_datetime(dt44_R0K_population_01['Date'],
format='%m/%d/%Y')
dt44_R0K_population_01.head()
```

	Date	Region	...	Natural_growth	Natural_growth_rate
0	2000-01-01	Busan	...	1877.0	5.8
1	2000-01-01	Chungcheongbuk-do	...	979.0	7.7
2	2000-01-01	Chungcheongnam-do	...	932.0	5.7
3	2000-01-01	Daegu	...	1940.0	9.1
4	2000-01-01	Daejeon	...	1294.0	11.2

[5 rows x 12 columns]

```
[284]: # 05.02.28-03
# add index column
# dt44

dt44_R0K_population_01.insert(0, 'ID', range(0, 0 +
len(dt44_R0K_population_01)))
```

```
[285]: # 05.02.28-04
# filter whole country
# dt44

dt44_R0K_population_01.set_index("Region", inplace = True)
dt44_R0K_population_01_ent = dt44_R0K_population_01.loc["Whole country"]
```

```
[286]: # 05.02.28-05
# filter whole country
# dt44

dt44_ROK_population_01_ent
```

```
[286]:
```

	ID	Date	...	Natural_growth	Natural_growth_rate
Region			...		
Whole country	17	2000-01-01	...	37187.0	9.3
Whole country	35	2000-02-01	...	36417.0	9.7
Whole country	53	2000-03-01	...	38286.0	9.5
Whole country	71	2000-04-01	...	32638.0	8.4
Whole country	89	2000-05-01	...	32630.0	8.1
...
Whole country	4787	2022-02-01	...	-8535.0	-2.2
Whole country	4805	2022-03-01	...	-21562.0	-4.9
Whole country	4823	2022-04-01	...	-15573.0	-3.7
Whole country	4841	2022-05-01	...	-8852.0	-2.0
Whole country	4859	2022-06-01	...	-6019.0	-1.4

[270 rows x 12 columns]

```
[287]: # 05.02.28-06
# omit four columns
# rows omitted because of irrelevance
# dt44

dt44_ROK_population_01_ent.drop(['Birth_rate'], axis = 1, inplace = True)
dt44_ROK_population_01_ent.drop(['Death_rate'], axis = 1, inplace = True)
dt44_ROK_population_01_ent.drop(['Divorce_rate'], axis = 1, inplace = True)
dt44_ROK_population_01_ent.drop(['Marriage_rate'], axis = 1, inplace = True)
dt44_ROK_population_01_ent.drop(['Natural_growth_rate'], axis = 1, inplace =
↪ True)
dt44_ROK_population_01_ent
```

```
[287]:
```

	ID	Date	Birth	...	Divorce	Marriage	Natural_growth
Region				...			
Whole country	17	2000-01-01	61644.0	...	9045.0	30120.0	37187.0
Whole country	35	2000-02-01	56723.0	...	9392.0	26187.0	36417.0
Whole country	53	2000-03-01	59878.0	...	10878.0	30418.0	38286.0
Whole country	71	2000-04-01	53058.0	...	8640.0	28179.0	32638.0
Whole country	89	2000-05-01	52492.0	...	10210.0	35330.0	32630.0
...
Whole country	4787	2022-02-01	20654.0	...	7136.0	15308.0	-8535.0
Whole country	4805	2022-03-01	22925.0	...	7882.0	15316.0	-21562.0
Whole country	4823	2022-04-01	21124.0	...	7198.0	15795.0	-15573.0
Whole country	4841	2022-05-01	20007.0	...	8372.0	17041.0	-8852.0

Whole country 4859 2022-06-01 18830.0 ... 7586.0 14898.0 -6019.0

[270 rows x 7 columns]

```
[288]: # 05.02.28-06
# plot six variables 2000 - 2020
# dt44

plt.figure(figsize = (12, 6))
plt.title('ROK Demographic Trends 2000 - 2020')
plt.grid(True)

plt.plot(
    'Date', 'Birth', data = dt44_ROK_population_01_ent,
    marker = '',
    color = '#318ce7',
    linewidth = 1,
    linestyle = 'solid',
    label = "Birth"
)

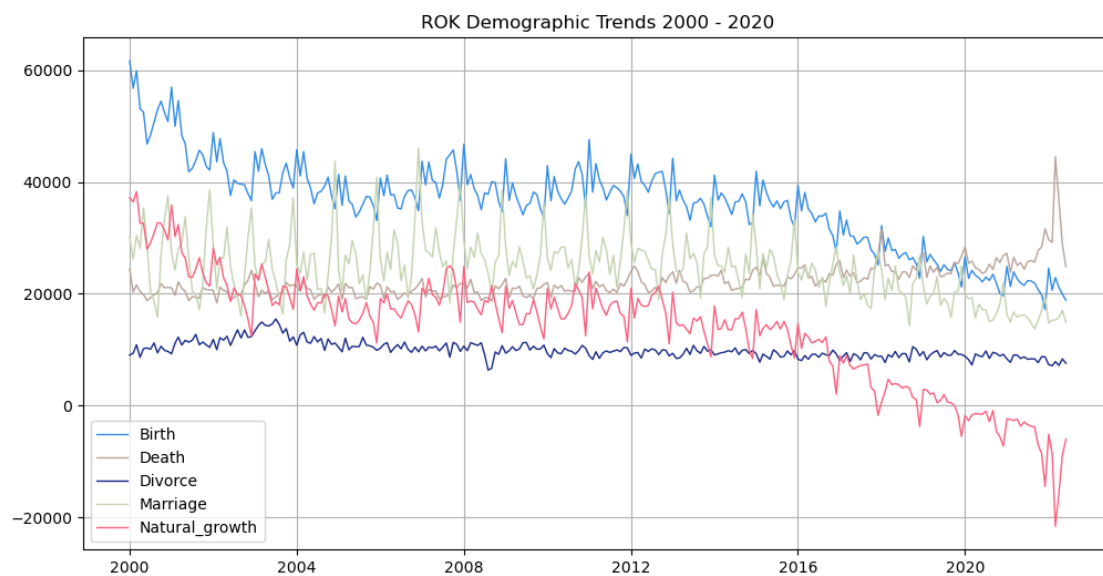
plt.plot(
    'Date', 'Death', data = dt44_ROK_population_01_ent,
    marker = '',
    color = '#bba498',
    linewidth = 1,
    linestyle = 'solid',
    label = "Death"
)

plt.plot(
    'Date', 'Divorce', data = dt44_ROK_population_01_ent,
    marker = '',
    color = '#182a86',
    linewidth = 1,
    linestyle = 'solid',
    label = "Divorce"
)

plt.plot(
    'Date', 'Marriage', data = dt44_ROK_population_01_ent,
    marker = '',
    color = '#c4d4b1',
    linewidth = 1,
    linestyle = 'solid',
    label = "Marriage"
)
```

```
plt.plot(
    'Date', 'Natural_growth', data = dt44_ROK_population_01_ent,
    marker = '',
    color = '#fb607f',
    linewidth = 1,
    linestyle = 'solid',
    label = "Natural_growth"
)

plt.legend()
plt.show()
```



```
[289]: # 05.02.29-01
# return dataframe
# dt45

print(dt45_RE_price_____01)
```

	region	type	1986_01	...	2012_10	2012_11	2012_12
0	Whole Country	Total	22.6	...	107.9	108.3	108.6
1	Whole Country	Apartments	14.4	...	110.0	110.6	110.9
2	Whole Country	single	41.1	...	103.7	103.9	103.9
3	Whole Country	Row Houses	22.1	...	105.6	105.7	105.8
4	Seoul	Total	23.0	...	106.5	106.8	107.0
5	Seoul	Apartments	14.2	...	107.9	108.4	108.7
6	Seoul	single	37.8	...	103.5	103.6	103.6
7	Seoul	Row Houses	23.5	...	105.3	105.4	105.6

8	6 Large Cities	Total	24.0	...	108.1	108.5	108.8
9	6 Large Cities	Apartments	16.7	...	110.1	110.6	111.0
10	6 Large Cities	single	42.0	...	104.1	104.2	104.3
11	6 Large Cities	Row Houses	22.3	...	104.7	104.8	104.8
12	Seoul Metropolitan Area	Total	-	...	106.1	106.5	106.6
13	Seoul Metropolitan Area	Apartments	-	...	107.6	108.0	108.2
14	Seoul Metropolitan Area	single	-	...	102.9	103.0	103.0
15	Seoul Metropolitan Area	Row Houses	-	...	104.8	104.9	105.0
16	Gyeonggi-do	Total	-	...	106.6	106.9	107.0
17	Gyeonggi-do	Apartments	-	...	108.0	108.5	108.6
18	Gyeonggi-do	single	-	...	102.2	102.2	102.3
19	Gyeonggi-do	Row Houses	-	...	106.1	106.2	106.1
20	Gangwon-do	Total	-	...	114.6	115.1	115.2
21	Gangwon-do	Apartments	-	...	117.9	118.3	118.4
22	Gangwon-do	single	-	...	107.5	107.8	108.0
23	Gangwon-do	Row Houses	-	...	111.3	111.8	111.5
24	Chungcheongbuk-do	Total	-	...	113.7	114.4	114.7
25	Chungcheongbuk-do	Apartments	-	...	119.2	120.2	120.7
26	Chungcheongbuk-do	single	-	...	104.3	104.8	105.1
27	Chungcheongbuk-do	Row Houses	-	...	107.7	108.4	108.7
28	Chungcheongnam-do	Total	-	...	115.2	116.9	117.8
29	Chungcheongnam-do	Apartments	-	...	121.9	124.6	126.1
30	Chungcheongnam-do	single	-	...	103.2	103.2	103.0
31	Chungcheongnam-do	Row Houses	-	...	108.6	109.1	109.6
32	Jeollabuk-do	Total	-	...	108.1	108.2	108.2
33	Jeollabuk-do	Apartments	-	...	109.8	110.0	110.1
34	Jeollabuk-do	single	-	...	103.6	103.5	103.5
35	Jeollabuk-do	Row Houses	-	...	111.5	111.7	111.3
36	Jeollanam-do	Total	-	...	108.6	108.8	108.7
37	Jeollanam-do	Apartments	-	...	110.6	110.8	110.7
38	Jeollanam-do	single	-	...	104.3	104.4	104.4
39	Jeollanam-do	Row Houses	-	...	102.5	102.7	102.4
40	Gyeongsangbuk-do	Total	-	...	113.2	113.9	114.8
41	Gyeongsangbuk-do	Apartments	-	...	118.5	119.4	120.6
42	Gyeongsangbuk-do	single	-	...	101.9	102.0	102.0
43	Gyeongsangbuk-do	Row Houses	-	...	109.6	110.0	111.2
44	Gyeongsangnam-do	Total	-	...	106.4	106.7	106.9
45	Gyeongsangnam-do	Apartments	-	...	107.4	107.8	107.9
46	Gyeongsangnam-do	single	-	...	107.0	107.1	107.3
47	Gyeongsangnam-do	Row Houses	-	...	108.0	107.9	107.9
48	Jeju-do	Total	-	...	-	-	-
49	Non Metropolitan Area	Total	-	...	110.6	111.2	111.5
50	Non Metropolitan Area	Apartments	-	...	113.9	114.6	115.1
51	Non Metropolitan Area	single	-	...	104.4	104.5	104.6
52	Non Metropolitan Area	Row Houses	-	...	108.2	108.5	108.9
53	5 Metropolitan Cities	Total	-	...	109.4	109.8	110.1
54	5 Metropolitan Cities	Apartments	-	...	111.5	112.0	112.4
55	5 Metropolitan Cities	single	-	...	104.2	104.4	104.5

56 5 Metropolitan Cities Row Houses - ... 107.7 107.9 108.0

[57 rows x 326 columns]

```
[290]: # 05.02.29-02
# return columns
# dt45

print(dt45_RE_price_____01.columns)
```

```
Index(['region', 'type', '1986_01', '1986_02', '1986_03', '1986_04', '1986_05',
      '1986_06', '1986_07', '1986_08',
      ...,
      '2012_03', '2012_04', '2012_05', '2012_06', '2012_07', '2012_08',
      '2012_09', '2012_10', '2012_11', '2012_12'],
      dtype='object', length=326)
```

```
[291]: # 05.02.29-03
# filter whole country
# dt45

dt45_RE_price_____01.set_index("region", inplace = True)
dt45_RE_price_____01 = dt45_RE_price_____01.loc["Whole Country"]
```

```
[292]: # 05.02.29-04
# return whole country
# dt45

dt45_RE_price_____01
```

```
[292]:
```

		type	1986_01	1986_02	...	2012_10	2012_11	2012_12
region					...			
Whole Country	Total		22.6	22.9	...	107.9	108.3	108.6
Whole Country	Apartments		14.4	14.8	...	110.0	110.6	110.9
Whole Country	single		41.1	41.4	...	103.7	103.9	103.9
Whole Country	Row Houses		22.1	22.2	...	105.6	105.7	105.8

[4 rows x 325 columns]

```
[293]: # 05.02.29-05
# convert to float
# dt45

dt45_RE_price_____01['1986_01'].astype(float)
dt45_RE_price_____01['1986_02'].astype(float)
dt45_RE_price_____01['1986_03'].astype(float)
dt45_RE_price_____01['1986_04'].astype(float)
dt45_RE_price_____01['1986_05'].astype(float)
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

dt45_RE_price_____01['2009_12'].astype(float)
dt45_RE_price_____01['2010_01'].astype(float)
dt45_RE_price_____01['2010_02'].astype(float)
dt45_RE_price_____01['2010_03'].astype(float)
dt45_RE_price_____01['2010_04'].astype(float)
dt45_RE_price_____01['2010_05'].astype(float)
dt45_RE_price_____01['2010_06'].astype(float)
dt45_RE_price_____01['2010_07'].astype(float)
dt45_RE_price_____01['2010_08'].astype(float)
dt45_RE_price_____01['2010_09'].astype(float)
dt45_RE_price_____01['2010_10'].astype(float)
dt45_RE_price_____01['2010_11'].astype(float)
dt45_RE_price_____01['2010_12'].astype(float)
dt45_RE_price_____01['2011_01'].astype(float)
dt45_RE_price_____01['2011_02'].astype(float)
dt45_RE_price_____01['2011_03'].astype(float)
dt45_RE_price_____01['2011_04'].astype(float)
dt45_RE_price_____01['2011_05'].astype(float)
dt45_RE_price_____01['2011_06'].astype(float)
dt45_RE_price_____01['2011_07'].astype(float)
dt45_RE_price_____01['2011_08'].astype(float)
dt45_RE_price_____01['2011_09'].astype(float)
dt45_RE_price_____01['2011_10'].astype(float)
dt45_RE_price_____01['2011_11'].astype(float)
dt45_RE_price_____01['2011_12'].astype(float)
dt45_RE_price_____01['2012_01'].astype(float)
dt45_RE_price_____01['2012_02'].astype(float)
dt45_RE_price_____01['2012_03'].astype(float)
dt45_RE_price_____01['2012_04'].astype(float)
dt45_RE_price_____01['2012_05'].astype(float)
dt45_RE_price_____01['2012_06'].astype(float)
dt45_RE_price_____01['2012_07'].astype(float)
dt45_RE_price_____01['2012_08'].astype(float)
dt45_RE_price_____01['2012_09'].astype(float)
dt45_RE_price_____01['2012_10'].astype(float)
dt45_RE_price_____01['2012_11'].astype(float)
dt45_RE_price_____01['2012_12'].astype(float)

```

```

[293]: region
       Whole Country    108.6
       Whole Country    110.9
       Whole Country    103.9
       Whole Country    105.8
       Name: 2012_12, dtype: float64

```

```

[294]: # 05.02.29-06
       # return dataframe

```

```
# dt45
```

```
print(dt45_RE_price_____01)
```

	type	1986_01	1986_02	...	2012_10	2012_11	2012_12
region				...			
Whole Country	Total	22.6	22.9	...	107.9	108.3	108.6
Whole Country	Apartments	14.4	14.8	...	110.0	110.6	110.9
Whole Country	single	41.1	41.4	...	103.7	103.9	103.9
Whole Country	Row Houses	22.1	22.2	...	105.6	105.7	105.8

[4 rows x 325 columns]

```
[295]: # 05.02.29-07
```

```
# switch axes
```

```
# dt45
```

```
dt45_RE_price_____01_tp = dt45_RE_price_____01.transpose()
```

```
print(dt45_RE_price_____01_tp)
```

region	Whole Country	Whole Country	Whole Country	Whole Country
type	Total	Apartments	single	Row Houses
1986_01	22.6	14.4	41.1	22.1
1986_02	22.9	14.8	41.4	22.2
1986_03	23.5	15.3	42.3	23.1
1986_04	24.0	15.6	43.0	23.5
...
2012_08	105.5	108.9	103.4	105.0
2012_09	107.4	109.3	103.5	105.3
2012_10	107.9	110.0	103.7	105.6
2012_11	108.3	110.6	103.9	105.7
2012_12	108.6	110.9	103.9	105.8

[325 rows x 4 columns]

```
[296]: # 05.02.29-08
```

```
# create dataframe
```

```
# dt45
```

```
dt45_RE_price_____01_tp_df = {
```

```
    'date': [
```

```
    '1986_01',
```

```
    '1986_02',
```

```
    '1986_03',
```

```
    '1986_04',
```

```
    '1986_05',
```

```
    '1986_06',
```

```
    '1986_07',
```

'1986_08',
'1986_09',
'1986_10',
'1986_11',
'1986_12',
'1987_01',
'1987_02',
'1987_03',
'1987_04',
'1987_05',
'1987_06',
'1987_07',
'1987_08',
'1987_09',
'1987_10',
'1987_11',
'1987_12',
'1988_01',
'1988_02',
'1988_03',
'1988_04',
'1988_05',
'1988_06',
'1988_07',
'1988_08',
'1988_09',
'1988_10',
'1988_11',
'1988_12',
'1989_01',
'1989_02',
'1989_03',
'1989_04',
'1989_05',
'1989_06',
'1989_07',
'1989_08',
'1989_09',
'1989_10',
'1989_11',
'1989_12',
'1990_01',
'1990_02',
'1990_03',
'1990_04',
'1990_05',
'1990_06',

'1990_07',
'1990_08',
'1990_09',
'1990_10',
'1990_11',
'1990_12',
'1991_01',
'1991_02',
'1991_03',
'1991_04',
'1991_05',
'1991_06',
'1991_07',
'1991_08',
'1991_09',
'1991_10',
'1991_11',
'1991_12',
'1992_01',
'1992_02',
'1992_03',
'1992_04',
'1992_05',
'1992_06',
'1992_07',
'1992_08',
'1992_09',
'1992_10',
'1992_11',
'1992_12',
'1993_01',
'1993_02',
'1993_03',
'1993_04',
'1993_05',
'1993_06',
'1993_07',
'1993_08',
'1993_09',
'1993_10',
'1993_11',
'1993_12',
'1994_01',
'1994_02',
'1994_03',
'1994_04',
'1994_05',

'1994_06',
'1994_07',
'1994_08',
'1994_09',
'1994_10',
'1994_11',
'1994_12',
'1995_01',
'1995_02',
'1995_03',
'1995_04',
'1995_05',
'1995_06',
'1995_07',
'1995_08',
'1995_09',
'1995_10',
'1995_11',
'1995_12',
'1996_01',
'1996_02',
'1996_03',
'1996_04',
'1996_05',
'1996_06',
'1996_07',
'1996_08',
'1996_09',
'1996_10',
'1996_11',
'1996_12',
'1997_01',
'1997_02',
'1997_03',
'1997_04',
'1997_05',
'1997_06',
'1997_07',
'1997_08',
'1997_09',
'1997_10',
'1997_11',
'1997_12',
'1998_01',
'1998_02',
'1998_03',
'1998_04',

'1998_05',
'1998_06',
'1998_07',
'1998_08',
'1998_09',
'1998_10',
'1998_11',
'1998_12',
'1999_01',
'1999_02',
'1999_03',
'1999_04',
'1999_05',
'1999_06',
'1999_07',
'1999_08',
'1999_09',
'1999_10',
'1999_11',
'1999_12',
'2000_01',
'2000_02',
'2000_03',
'2000_04',
'2000_05',
'2000_06',
'2000_07',
'2000_08',
'2000_09',
'2000_10',
'2000_11',
'2000_12',
'2001_01',
'2001_02',
'2001_03',
'2001_04',
'2001_05',
'2001_06',
'2001_07',
'2001_08',
'2001_09',
'2001_10',
'2001_11',
'2001_12',
'2002_01',
'2002_02',
'2002_03',

'2002_04',
'2002_05',
'2002_06',
'2002_07',
'2002_08',
'2002_09',
'2002_10',
'2002_11',
'2002_12',
'2003_01',
'2003_02',
'2003_03',
'2003_04',
'2003_05',
'2003_06',
'2003_07',
'2003_08',
'2003_09',
'2003_10',
'2003_11',
'2003_12',
'2004_01',
'2004_02',
'2004_03',
'2004_04',
'2004_05',
'2004_06',
'2004_07',
'2004_08',
'2004_09',
'2004_10',
'2004_11',
'2004_12',
'2005_01',
'2005_02',
'2005_03',
'2005_04',
'2005_05',
'2005_06',
'2005_07',
'2005_08',
'2005_09',
'2005_10',
'2005_11',
'2005_12',
'2006_01',
'2006_02',

'2006_03',
'2006_04',
'2006_05',
'2006_06',
'2006_07',
'2006_08',
'2006_09',
'2006_10',
'2006_11',
'2006_12',
'2007_01',
'2007_02',
'2007_03',
'2007_04',
'2007_05',
'2007_06',
'2007_07',
'2007_08',
'2007_09',
'2007_10',
'2007_11',
'2007_12',
'2008_01',
'2008_02',
'2008_03',
'2008_04',
'2008_05',
'2008_06',
'2008_07',
'2008_08',
'2008_09',
'2008_10',
'2008_11',
'2008_12',
'2009_01',
'2009_02',
'2009_03',
'2009_04',
'2009_05',
'2009_06',
'2009_07',
'2009_08',
'2009_09',
'2009_10',
'2009_11',
'2009_12',
'2010_01',

```
'2010_02',  
'2010_03',  
'2010_04',  
'2010_05',  
'2010_06',  
'2010_07',  
'2010_08',  
'2010_09',  
'2010_10',  
'2010_11',  
'2010_12',  
'2011_01',  
'2011_02',  
'2011_03',  
'2011_04',  
'2011_05',  
'2011_06',  
'2011_07',  
'2011_08',  
'2011_09',  
'2011_10',  
'2011_11',  
'2011_12',  
'2012_01',  
'2012_02',  
'2012_03',  
'2012_04',  
'2012_05',  
'2012_06',  
'2012_07',  
'2012_08',  
'2012_09',  
'2012_10',  
'2012_11',  
'2012_12'  
],  
'total': [  
22.6,  
22.9,  
23.5,  
24.0,  
24.0,  
24.0,  
23.9,  
24.0,  
24.3,  
24.5,
```

24.1,
23.9,
23.8,
24.2,
24.9,
25.6,
25.7,
26.0,
26.0,
26.1,
27.4,
28.0,
28.5,
28.5,
28.9,
29.8,
31.2,
32.2,
32.4,
32.6,
32.7,
33.3,
33.4,
33.2,
32.7,
32.3,
32.6,
33.7,
34.9,
36.3,
36.6,
36.5,
36.4,
36.5,
37.3,
38.0,
38.0,
38.0,
39.0,
43.6,
44.6,
45.7,
45.2,
44.7,
44.4,
44.6,
45.2,

45.4,
45.4,
44.4,
44.4,
45.2,
46.5,
47.5,
47.1,
46.4,
45.9,
45.8,
46.1,
46.4,
45.9,
45.2,
45.3,
45.8,
47.4,
48.2,
48.6,
48.4,
48.2,
48.6,
49.3,
49.4,
49.1,
48.6,
48.6,
49.3,
50.1,
50.3,
50.1,
49.7,
49.5,
49.5,
49.7,
49.9,
49.9,
49.8,
49.9,
50.1,
50.7,
50.9,
51.0,
51.1,
51.1,
51.3,

51.9,
52.1,
52.1,
52.1,
52.1,
52.6,
53.3,
53.6,
53.6,
53.6,
53.5,
53.6,
53.9,
54.1,
54.1,
54.0,
54.2,
54.8,
55.7,
56.0,
56.1,
56.1,
56.1,
56.5,
57.4,
57.6,
57.6,
57.5,
57.9,
58.4,
58.9,
59.0,
58.8,
58.7,
58.6,
58.6,
58.8,
58.8,
58.4,
58.0,
57.4,
56.0,
53.7,
50.7,
48.7,
47.7,
47.3,

47.7,
47.8,
47.4,
47.1,
47.3,
48.2,
49.6,
50.6,
51.3,
51.8,
52.1,
52.6,
53.4,
54.7,
55.3,
55.3,
55.2,
55.9,
57.5,
58.8,
59.6,
59.6,
59.5,
59.6,
60.5,
61.6,
62.3,
62.1,
61.4,
61.4,
62.9,
64.4,
65.2,
65.8,
66.4,
67.5,
69.1,
70.7,
71.2,
71.2,
71.5,
73.0,
75.0,
76.7,
77.7,
77.9,
77.9,

78.2,
79.0,
80.0,
79.9,
79.1,
78.7,
78.6,
79.2,
79.8,
79.9,
79.7,
79.2,
78.8,
78.7,
79.0,
79.0,
78.5,
77.6,
77.2,
77.4,
77.6,
77.7,
77.5,
76.9,
76.3,
75.5,
75.2,
74.8,
74.3,
73.7,
73.3,
73.5,
73.8,
74.1,
74.2,
74.3,
74.4,
74.5,
75.1,
75.6,
75.8,
75.9,
76.2,
76.5,
77.1,
77.5,
77.8,

77.9,
78.0,
78.1,
78.8,
79.6,
80.4,
80.9,
81.2,
81.4,
81.8,
82.1,
82.1,
82.1,
82.2,
82.3,
82.5,
82.7,
82.9,
82.9,
83.0,
83.3,
83.8,
84.3,
84.5,
84.8,
85.0,
85.1,
85.4,
85.6,
85.3,
84.3,
83.5,
83.3,
83.4,
83.5,
83.7,
83.9,
84.2,
84.6,
85.8,
86.5,
86.9,
87.2,
87.4,
88.0,
88.6,
89.1,

```

89.5,
89.8,
90.1,
90.4,
91.0,
91.7,
92.7,
93.4,
94.2,
95.7,
97.4,
98.5,
99.4,
100.0,
100.8,
101.9,
103.3,
104.2,
104.8,
104.9,
105.1,
105.5,
106.0,
106.5,
106.6,
106.8,
106.9,
105.5,
107.4,
107.9,
108.3,
108.6,
]}
dt45_RE_price_____01_tp_df01 = pd.DataFrame(dt45_RE_price_____01_tp_df)

dt45_RE_price_____01_tp_df01

```

```

[296]:
      date  total
0   1986_01   22.6
1   1986_02   22.9
2   1986_03   23.5
3   1986_04   24.0
4   1986_05   24.0
..    ...    ...
319 2012_08  105.5
320 2012_09  107.4
321 2012_10  107.9

```

```
322 2012_11 108.3
323 2012_12 108.6
```

```
[324 rows x 2 columns]
```

```
[297]: # 05.02.29-09
# create dummy variables
# due to returning boolean values, converting dummies to integers -----
# dt45
```

```
dt45_RE_price_____01_tp_df01_dv = pd.
↳get_dummies(dt45_RE_price_____01_tp_df01, drop_first = True, dtype=int)
```

```
[298]: # 05.02.29-10
# split the data into a training and test set
# target = total
# total = variable 01
# select columns
# for dataset dt45 containing price data
# dt45
```

```
dt45_RE_price_____01_tp_df01_dv_x01 = dt45_RE_price_____01_tp_df01_dv.
↳drop(['total'], axis = 1)
dt45_RE_price_____01_tp_df01_dv_y01 =
↳dt45_RE_price_____01_tp_df01_dv['total']
```

```
[299]: # 05.02.29-11
# Split the data into a training and test set
# target = total
# total = variable 01
# split into train and test
# for dataset dt45 containing price data
# dt45
```

```
dt45_RE_price_____01_tp_df01_dv_x01_trn,
↳dt45_RE_price_____01_tp_df01_dv_x01_tst,
↳dt45_RE_price_____01_tp_df01_dv_y01_trn,
↳dt45_RE_price_____01_tp_df01_dv_y01_tst =
↳train_test_split(dt45_RE_price_____01_tp_df01_dv_x01,
↳dt45_RE_price_____01_tp_df01_dv_y01, test_size = 0.3, random_state = 0)
```

```
[300]: # 05.02.29-12
# Split the data into a training and test set
# target = total
# total = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign regression variable
```

```
# for dataset dt45 containing price data
# dt45

dt45_RE_price_____01_tp_df01_dv_lr01 = LinearRegression()
```

```
[301]: # 05.02.29-13
# Split the data into a training and test set
# target = total
# total = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign regression variable
# for dataset dt45 containing price data
# dt45

dt45_RE_price_____01_tp_df01_dv_lr01.
  ↳fit(dt45_RE_price_____01_tp_df01_dv_x01_trn,␣
  ↳dt45_RE_price_____01_tp_df01_dv_y01_trn)
```

```
[301]: LinearRegression()
```

```
[302]: # 05.02.29-14
# Split the data into a training and test set
# target = total
# total = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# fit variables to model
# for dataset dt45 containing price data
# dt45

dt45_RE_price_____01_tp_df01_dv_pdct = dt45_RE_price_____01_tp_df01_dv_lr01.
  ↳predict(dt45_RE_price_____01_tp_df01_dv_x01_tst)
```

```
[303]: # 05.02.29-15
# Split the data into a training and test set
# target = tota
# tota = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# assign variable for rmse and r2
# for dataset dt45 containing price data
# dt45

dt45_RE_price_____01_tp_df01_dv_rmse01 = np.
  ↳sqrt(mean_squared_error(dt45_RE_price_____01_tp_df01_dv_y01_tst,␣
  ↳dt45_RE_price_____01_tp_df01_dv_pdct))
dt45_RE_price_____01_tp_df01_dv_r201 =␣
  ↳r2_score(dt45_RE_price_____01_tp_df01_dv_y01_tst,␣
  ↳dt45_RE_price_____01_tp_df01_dv_pdct)
```

```
[304]: # 05.02.29-16
# Split the data into a training and test set
# target = total
# total = variable 01
# Run a linear regression and report the R2-value and RMSE on the test set.
# return rmse and r2 dt25
# rmse: 42.10
# r2: -3.13
# rmse is very high
# r2 is very low
# this is reflective of the nature of the housing industry
# prices fluctuate and vary
# for dataset dt45 containing price data
# dt45
```

```
print(f'rmse: {dt45_RE_price_____01_tp_df01_dv_rmse01}')
print(f'r2: {dt45_RE_price_____01_tp_df01_dv_r201}')
```

rmse: 42.101305181183925

r2: -3.1274475249950084

```
[305]: # 05.02.29-17
# assign variables for logistic regression
# dt45

dt45_RE_price_____01_tp_df01_x2 = dt45_RE_price_____01_tp_df01[['date']]
dt45_RE_price_____01_tp_df01_y2 = dt45_RE_price_____01_tp_df01[['total']]
```

```
[306]: # 05.02.29-18
# split data for logistic regression model
# dt45

dt45_RE_price_____01_tp_df01_x2_trn,dt45_RE_price_____01_tp_df01_x2_tst,dt45_RE_price_____01_tp_df01_y2_trn,dt45_RE_price_____01_tp_df01_y2_tst = train_test_split(dt45_RE_price_____01_tp_df01_x2,dt45_RE_price_____01_tp_df01_y2,random_state = 100)
```

```
[307]: # 05.02.29-10
# make data categorical for y
# dt45

lab_y01 = preprocessing.LabelEncoder()
dt45_RE_price_____01_tp_df01_y2_trn = lab_y01.fit_transform(dt45_RE_price_____01_tp_df01_y2_trn)
```

```
[308]: # 05.02.29-20
# assign variable for logistic regression
# dt45

dt45_RE_price_____01_tp_df01_logreg = LogisticRegression()
```

```
dt45_RE_price_____01_tp_df01_logreg.  
↳fit(dt45_RE_price_____01_tp_df01_x2_trn,dt45_RE_price_____01_tp_df01_y2_trn)
```

[308]: LogisticRegression()

```
[309]: # 05.02.29-21  
# predict based on model  
# dt45  
  
dt45_RE_price_____01_tp_df01_y2_prd = dt45_RE_price_____01_tp_df01_logreg.  
↳predict(dt45_RE_price_____01_tp_df01_x2_tst)  
print (dt45_RE_price_____01_tp_df01_x2_tst)  
print (dt45_RE_price_____01_tp_df01_y2_prd)
```

```
date  
223 2004_08  
60 1991_01  
232 2005_05  
175 2000_08  
142 1997_11  
..  
3 1986_04  
266 2008_03  
111 1995_04  
263 2007_12  
315 2012_04
```

[130 rows x 1 columns]

```
[59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59  
59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59  
59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59  
59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59  
59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59 59  
59 59 59 59 59 59 59 59 59 59]
```

```
[310]: # 05.02.29-22  
# unable to return results on cotinuous and binary data  
# dt45  
  
'''print('Accuracy: ', metrics.  
↳accuracy_score(dt45_RE_price_____01_tp_df01_y2_tst,▯  
↳dt45_RE_price_____01_tp_df01_y2_prd))  
print('Recall: ', metrics.recall_score(dt45_RE_price_____01_tp_df01_y2_tst,▯  
↳dt45_RE_price_____01_tp_df01_y2_prd, zero_division=1))  
print('Precision: ', metrics.  
↳precision_score(dt45_RE_price_____01_tp_df01_y2_tst,▯  
↳dt45_RE_price_____01_tp_df01_y2_prd, zero_division=1))
```

```
print('CL Report: ', metrics.
      ↪classification_report(dt45_RE_price____01_tp_df01_y2_tst,
      ↪dt45_RE_price____01_tp_df01_y2_prd, zero_division=1))'''
```

```
[310]: "print('Accuracy: ',
metrics.accuracy_score(dt45_RE_price____01_tp_df01_y2_tst,
dt45_RE_price____01_tp_df01_y2_prd))\nprint('Recall: ',
metrics.recall_score(dt45_RE_price____01_tp_df01_y2_tst,
dt45_RE_price____01_tp_df01_y2_prd, zero_division=1))\nprint('Precision: ',
metrics.precision_score(dt45_RE_price____01_tp_df01_y2_tst,
dt45_RE_price____01_tp_df01_y2_prd, zero_division=1))\nprint('CL Report: ',
metrics.classification_report(dt45_RE_price____01_tp_df01_y2_tst,
dt45_RE_price____01_tp_df01_y2_prd, zero_division=1))"
```

```
[311]: # 05.02.29-23
# assign variable for predict probability
# dt45

dt45_RE_price____01_tp_df01_y2_prd_prb =
      ↪dt45_RE_price____01_tp_df01_logreg.
      ↪predict_proba(dt45_RE_price____01_tp_df01_x2_tst)[:,1]
```

```
[312]: # 05.02.29-24
# make data categorical for y
# dt45

lab_y02 = preprocessing.LabelEncoder()
dt45_RE_price____01_tp_df01_y2_tst = lab_y02.
      ↪fit_transform(dt45_RE_price____01_tp_df01_y2_tst)
```

```
[313]: # 05.02.29-25
# assign variables for false and true positive rates
# multiclass format not supported
# dt45

"""false_positive_rate, true_positive_rate, _ = metrics.
      ↪roc_curve(dt45_RE_price____01_tp_df01_y2_tst,
      ↪dt45_RE_price____01_tp_df01_y2_prd_prb)"""
```

```
[313]: 'false_positive_rate, true_positive_rate, _ =
metrics.roc_curve(dt45_RE_price____01_tp_df01_y2_tst,
dt45_RE_price____01_tp_df01_y2_prd_prb)'
```

```
[314]: # 05.02.29-26
# assign variables for auc score
# multi_class must be in ('ovo', 'ovr')
# dt45
```

```
"""auc = metrics.roc_auc_score(dt45_RE_price____01_tp_df01_y2_tst,
↪dt45_RE_price____01_tp_df01_y2_prd_prb)"""
```

```
[314]: 'auc = metrics.roc_auc_score(dt45_RE_price____01_tp_df01_y2_tst,
dt45_RE_price____01_tp_df01_y2_prd_prb)'
```

```
[315]: # 05.02.29-27
# plot ROC curve
# dt45

"""plt.plot(false_positive_rate, true_positive_rate, label="AUC="+str(auc))
plt.title('ROC Curve')
plt.ylabel('True Positive Rate')
plt.xlabel('false Positive Rate')
plt.legend(loc=4)"""
```

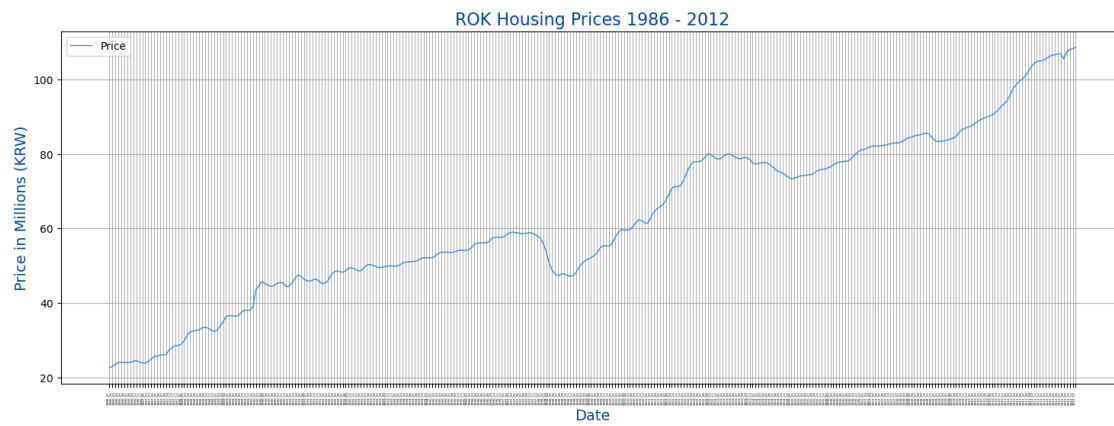
```
[315]: 'plt.plot(false_positive_rate,
true_positive_rate, label="AUC="+str(auc))\nplt.title(\ 'ROC
Curve'\)\nplt.ylabel(\ 'True Positive Rate'\)\nplt.xlabel(\ 'false Positive
Rate'\)\nplt.legend(loc=4)'
```

```
[316]: # 05.02.29-28
# plot prices
# dt44

plt.figure(figsize = (18, 6))
plt.title('ROK Housing Prices 1986 - 2012', fontsize = 16, color = '#0047ab')
plt.grid(True)
plt.box(True)
plt.xlabel('Date', fontsize = 14, color = '#0047ab')
plt.ylabel('Price in Millions (KRW)', fontsize = 14, color = '#0047ab')
plt.xticks(
    rotation = 90,
    horizontalalignment = 'right',
    fontweight = 'light',
    fontsize = 3
)
plt.plot(
    'date', 'total', data = dt45_RE_price____01_tp_df01,
    marker = '',
    color = '#318ce7',
    linewidth = 1,
    linestyle = 'solid',
    label = "Price"
)
```



```
plt.legend()  
plt.show()
```



```
[ ]:
```