

```
In [1]: # import modułów
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
import matplotlib.pyplot as plt
```

```
In [2]: # import danych pobranych z Banku Danych Lokalnych GUS
part1 = pd.read_excel('F:\DataScience\Data\Inflacja\data1.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part2 = pd.read_excel('F:\DataScience\Data\Inflacja\data2.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part3 = pd.read_excel('F:\DataScience\Data\Inflacja\data3.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part4 = pd.read_excel('F:\DataScience\Data\Inflacja\data4.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part5 = pd.read_excel('F:\DataScience\Data\Inflacja\data5.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part6 = pd.read_excel('F:\DataScience\Data\Inflacja\data6.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part7 = pd.read_excel('F:\DataScience\Data\Inflacja\data7.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part8 = pd.read_excel('F:\DataScience\Data\Inflacja\data8.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part9 = pd.read_excel('F:\DataScience\Data\Inflacja\data9.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part10 = pd.read_excel('F:\DataScience\Data\Inflacja\data10.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part11 = pd.read_excel('F:\DataScience\Data\Inflacja\data11.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part12 = pd.read_excel('F:\DataScience\Data\Inflacja\data12.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part13 = pd.read_excel('F:\DataScience\Data\Inflacja\data13.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
part14 = pd.read_excel('F:\DataScience\Data\Inflacja\data14.xlsx', sheet_name = 'DANE', usecols = ['Rodzaje produktów', 'Rok', 'Wartosc'])
```

```
In [3]: # połączenie danych, ustawienie indeksów oraz unstack kolumny "Rok" z wierszy do kolumn dla lepszej prezentacji danych
part1.set_index(['Rodzaje produktów', 'Rok'], inplace = True)
data = part1.unstack(level = 'Rok')

parts = [part2, part3, part4, part5, part6, part7, part8, part9, part10, part11, part12, part13, part14]
for part in parts:
    part.set_index(['Rodzaje produktów', 'Rok'], inplace = True)
    part = part.unstack(level = 'Rok')
    data = data.append(part)
data.head()
```

Out[3]:

| | Wartosc | | | | | | | | | | |
|--|---------|------|------|------|------|------|------|------|------|------|------|
| Rok | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 |
| Rodzaje produktów | | | | | | | | | | | |
| chleb mieszany zwykły (pszenno-żytni) - za 1kg | 3.24 | 3.58 | 3.75 | 3.89 | 3.92 | 3.98 | 3.96 | 4.01 | 4.13 | 4.41 | 4.68 |
| kasza gryczana prażona cała - za 1kg | 4.52 | 6.68 | 4.53 | 4.85 | 5.0 | 5.04 | 5.36 | 5.0 | 4.33 | 4.09 | 4.69 |
| kasza jęczmienna "Mazurska" - za 1kg | 1.32 | 1.76 | 1.88 | 2.06 | 2.13 | 2.1 | 2.26 | 2.08 | 1.83 | 1.89 | 1.92 |
| mąka pszenna "Poznańska", workowana - za 1kg | 1.01 | 1.36 | 1.32 | 1.3 | 1.1 | 1.07 | 0.97 | 1.04 | 1.11 | 1.18 | 1.17 |
| cegła budowlana pełna palona kl.15 - za 1szt | 1.23 | 1.21 | 1.2 | 1.18 | 1.17 | 1.15 | 1.14 | 1.14 | 1.19 | 1.29 | 1.37 |

```
In [4]: # Usunięcie multiindeksu "Wartosc"
data = data.transpose()
data.reset_index(inplace = True)
data = data.drop(columns = ['level_0'])
data.set_index('Rok',inplace = True)
data = data.transpose()
data.head()
```

Out[4]:

| Rok | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 |
|--|------|------|------|------|------|------|------|------|------|------|------|
| Rodzaje produktów | | | | | | | | | | | |
| chleb mieszany zwykły (pszenno-żytni) - za 1kg | 3.24 | 3.58 | 3.75 | 3.89 | 3.92 | 3.98 | 3.96 | 4.01 | 4.13 | 4.41 | 4.68 |
| kasza gryczana prażona cała - za 1kg | 4.52 | 6.68 | 4.53 | 4.85 | 5.0 | 5.04 | 5.36 | 5.0 | 4.33 | 4.09 | 4.69 |
| kasza jęczmienna "Mazurska" - za 1kg | 1.32 | 1.76 | 1.88 | 2.06 | 2.13 | 2.1 | 2.26 | 2.08 | 1.83 | 1.89 | 1.92 |
| mąka pszenna "Poznańska", workowana - za 1kg | 1.01 | 1.36 | 1.32 | 1.3 | 1.1 | 1.07 | 0.97 | 1.04 | 1.11 | 1.18 | 1.17 |
| cegła budowlana pełna palona kl.15 - za 1szt | 1.23 | 1.21 | 1.2 | 1.18 | 1.17 | 1.15 | 1.14 | 1.14 | 1.19 | 1.29 | 1.37 |

```
In [5]: # Zmiana nazwy kolumn
data.reset_index(inplace = True)
data.rename(columns = {'Rodzaje produktów': 'ProduktLubUsługa'}, inplace = True)
data.set_index('ProduktLubUsługa', inplace = True)
data.head()
```

Out[5]:

| Rok | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 |
|--|------|------|------|------|------|------|------|------|------|------|------|
| ProduktLubUsługa | | | | | | | | | | | |
| chleb mieszany zwykły (pszenno-żytni) - za 1kg | 3.24 | 3.58 | 3.75 | 3.89 | 3.92 | 3.98 | 3.96 | 4.01 | 4.13 | 4.41 | 4.68 |
| kasza gryczana prażona cała - za 1kg | 4.52 | 6.68 | 4.53 | 4.85 | 5.0 | 5.04 | 5.36 | 5.0 | 4.33 | 4.09 | 4.69 |
| kasza jęczmienna "Mazurska" - za 1kg | 1.32 | 1.76 | 1.88 | 2.06 | 2.13 | 2.1 | 2.26 | 2.08 | 1.83 | 1.89 | 1.92 |
| mąka pszenna "Poznańska", workowana - za 1kg | 1.01 | 1.36 | 1.32 | 1.3 | 1.1 | 1.07 | 0.97 | 1.04 | 1.11 | 1.18 | 1.17 |
| cegła budowlana pełna palona kl.15 - za 1szt | 1.23 | 1.21 | 1.2 | 1.18 | 1.17 | 1.15 | 1.14 | 1.14 | 1.19 | 1.29 | 1.37 |

```
In [6]: # Usunięcie duplikatów i sprawdzenie rozmiaru tablicy
data.reset_index(inplace = True)
data.drop_duplicates(subset = ['ProduktLubUsługa'], inplace = True)
data.set_index('ProduktLubUsługa', inplace = True)
data.shape
```

Out[6]: (193, 11)

```
In [7]: # Usunięcie produktów i usług z brakującymi wartościami w wierszach
data = data.replace('-', np.NaN)
data.dropna(how = 'any', inplace = True)
data.shape
```

Out[7]: (117, 11)

```
In [8]: # Eksport danych do MS Excel w celu korekty nazw produktów i usług oraz usunięcie pozycji nietypowych
data.reset_index(inplace = True)
excelWriter = pd.ExcelWriter('F:\DataScience\Data\Inflacja\data_export.xlsx')
data.to_excel(excelWriter, index = False)
excelWriter.save()
```

```
In [9]: # Ponowny import poprawionych danych wraz z kolumną zawierającą szacowane miesięczne spożycie(koszyk inflacyjny)
data = pd.read_excel('F:\DataScience\Data\Inflacja\data_import.xlsx')
data.set_index('ProduktLubUsługa', inplace = True)
data.head(10)
```

Out[9]:

| | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | 2020 | miesięcznie | rocznie |
|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-------------|---------|
| ProduktLubUsługa | | | | | | | | | | | | | |
| baleron gotowany - za 1kg | 17.58 | 17.89 | 19.28 | 19.80 | 20.19 | 19.55 | 19.61 | 21.08 | 21.88 | 23.72 | 26.40 | 0.0 | 0.5 |
| bateria zlewozmywakowa | 134.07 | 138.05 | 144.79 | 151.25 | 151.01 | 150.52 | 152.73 | 155.21 | 159.54 | 164.25 | 165.55 | 0.0 | 0.0 |
| benzyna silnikowa bezołowiowa 95 - za 1L | 4.59 | 5.14 | 5.73 | 5.51 | 5.30 | 4.65 | 4.37 | 4.63 | 4.95 | 5.02 | 4.48 | 0.0 | 0.0 |
| bilet do kina | 15.07 | 15.62 | 16.36 | 17.06 | 17.65 | 18.23 | 18.73 | 19.48 | 19.80 | 20.02 | 19.34 | 1.0 | 12.0 |
| bilet do teatru | 34.53 | 37.10 | 40.21 | 41.61 | 43.63 | 45.14 | 46.62 | 48.21 | 49.41 | 51.01 | 53.97 | 0.0 | 0.0 |
| bilet normalny na przejazd autobusem miejskim | 2.23 | 2.34 | 2.58 | 2.72 | 2.73 | 2.71 | 2.71 | 2.69 | 2.74 | 2.77 | 2.87 | 10.0 | 120.0 |
| bilet normalny na przejazd tramwajem | 2.53 | 2.63 | 2.95 | 3.28 | 3.30 | 3.32 | 3.32 | 3.33 | 3.36 | 3.40 | 3.56 | 10.0 | 120.0 |
| boczek surowy - za 1kg | 12.06 | 12.54 | 14.40 | 14.93 | 14.51 | 13.69 | 14.30 | 16.47 | 16.30 | 17.44 | 18.99 | 0.5 | 6.0 |
| boczek wędzony - za 1kg | 16.58 | 17.13 | 19.25 | 20.09 | 20.15 | 19.62 | 19.91 | 22.28 | 23.31 | 24.79 | 27.91 | 0.5 | 6.0 |
| botki męskie skórzane na podeszwie nieskórzanej - za 1parę | 188.46 | 195.95 | 211.60 | 223.58 | 227.91 | 227.47 | 225.02 | 218.93 | 223.76 | 228.65 | 232.23 | 2.0 | 24.0 |

```
In [10]: # wygenerowanie kolumn z kosztami danych produktów w danym roku

for number in range(0,11):
    data['kwota_'+str(2010+number)] = data[2010+number]*data['rocznie']
data.head(10)
```

Out[10]:

| | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | ... | kwota_2011 | kwota_2012 | kwota_2013 | kwota_2014 | kwota_2015 | kwota_2016 | kwota_2017 | kwota_2018 | kwota_2019 | kwota |
|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----|------------|------------|------------|------------|------------|------------|------------|------------|------------|-------|
| ProduktLubUsługa | | | | | | | | | | | | | | | | | | | | | |
| baleron gotowany - za 1kg | 17.58 | 17.89 | 19.28 | 19.80 | 20.19 | 19.55 | 19.61 | 21.08 | 21.88 | 23.72 | ... | 8.945 | 9.64 | 9.90 | 10.095 | 9.775 | 9.805 | 10.54 | 10.94 | 11.86 | |
| bateria zlewozmywakowa | 134.07 | 138.05 | 144.79 | 151.25 | 151.01 | 150.52 | 152.73 | 155.21 | 159.54 | 164.25 | ... | 0.000 | 0.00 | 0.00 | 0.000 | 0.000 | 0.000 | 0.00 | 0.00 | 0.00 | |
| benzyna silnikowa bezołowiowa 95 - za 1L | 4.59 | 5.14 | 5.73 | 5.51 | 5.30 | 4.65 | 4.37 | 4.63 | 4.95 | 5.02 | ... | 0.000 | 0.00 | 0.00 | 0.000 | 0.000 | 0.000 | 0.00 | 0.00 | 0.00 | |
| bilet do kina | 15.07 | 15.62 | 16.36 | 17.06 | 17.65 | 18.23 | 18.73 | 19.48 | 19.80 | 20.02 | ... | 187.440 | 196.32 | 204.72 | 211.800 | 218.760 | 224.760 | 233.76 | 237.60 | 240.24 | ; |
| bilet do teatru | 34.53 | 37.10 | 40.21 | 41.61 | 43.63 | 45.14 | 46.62 | 48.21 | 49.41 | 51.01 | ... | 0.000 | 0.00 | 0.00 | 0.000 | 0.000 | 0.000 | 0.00 | 0.00 | 0.00 | |
| bilet normalny na przejazd autobusem miejskim | 2.23 | 2.34 | 2.58 | 2.72 | 2.73 | 2.71 | 2.71 | 2.69 | 2.74 | 2.77 | ... | 280.800 | 309.60 | 326.40 | 327.600 | 325.200 | 325.200 | 322.80 | 328.80 | 332.40 | ; |
| bilet normalny na przejazd tramwajem | 2.53 | 2.63 | 2.95 | 3.28 | 3.30 | 3.32 | 3.32 | 3.33 | 3.36 | 3.40 | ... | 315.600 | 354.00 | 393.60 | 396.000 | 398.400 | 398.400 | 399.60 | 403.20 | 408.00 | , |
| boczek surowy - za 1kg | 12.06 | 12.54 | 14.40 | 14.93 | 14.51 | 13.69 | 14.30 | 16.47 | 16.30 | 17.44 | ... | 75.240 | 86.40 | 89.58 | 87.060 | 82.140 | 85.800 | 98.82 | 97.80 | 104.64 | |
| boczek wędzony - za 1kg | 16.58 | 17.13 | 19.25 | 20.09 | 20.15 | 19.62 | 19.91 | 22.28 | 23.31 | 24.79 | ... | 102.780 | 115.50 | 120.54 | 120.900 | 117.720 | 119.460 | 133.68 | 139.86 | 148.74 | . |
| botki męskie skórzane na podeszwie nieskórzanej - za 1parę | 188.46 | 195.95 | 211.60 | 223.58 | 227.91 | 227.47 | 225.02 | 218.93 | 223.76 | 228.65 | ... | 4702.800 | 5078.40 | 5365.92 | 5469.840 | 5459.280 | 5400.480 | 5254.32 | 5370.24 | 5487.60 | 5! |

10 rows × 24 columns



```
In [11]: # Usunięcie wierszy z produktami/usługami nieobcnymi w koszyku inflacyjnym
data = data.where(data['kwota_2020'] != 0).dropna()
data.head(10)
```

Out[11]:

| | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | 2017 | 2018 | 2019 | ... | kwota_2011 | kwota_2012 | kwota_2013 | kwota_2014 | kwota_2015 | kwota_2016 | kwota_2017 | kwota_2018 | kwota_2019 | kwota_2020 |
|--|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|-----|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| ProduktLubUsługa | | | | | | | | | | | | | | | | | | | | | |
| baleron gotowany - za 1kg | 17.58 | 17.89 | 19.28 | 19.80 | 20.19 | 19.55 | 19.61 | 21.08 | 21.88 | 23.72 | ... | 8.945 | 9.64 | 9.90 | 10.095 | 9.775 | 9.805 | 10.54 | 10.94 | 11.86 | 12.72 |
| bilet do kina | 15.07 | 15.62 | 16.36 | 17.06 | 17.65 | 18.23 | 18.73 | 19.48 | 19.80 | 20.02 | ... | 187.440 | 196.32 | 204.72 | 211.800 | 218.760 | 224.760 | 233.76 | 237.60 | 240.24 | 248.40 |
| bilet normalny na przejazd autobusem miejskim | 2.23 | 2.34 | 2.58 | 2.72 | 2.73 | 2.71 | 2.71 | 2.69 | 2.74 | 2.77 | ... | 280.800 | 309.60 | 326.40 | 327.600 | 325.200 | 325.200 | 322.80 | 328.80 | 332.40 | 336.00 |
| bilet normalny na przejazd tramwajem | 2.53 | 2.63 | 2.95 | 3.28 | 3.30 | 3.32 | 3.32 | 3.33 | 3.36 | 3.40 | ... | 315.600 | 354.00 | 393.60 | 396.000 | 398.400 | 398.400 | 399.60 | 403.20 | 408.00 | 416.40 |
| boczek surowy - za 1kg | 12.06 | 12.54 | 14.40 | 14.93 | 14.51 | 13.69 | 14.30 | 16.47 | 16.30 | 17.44 | ... | 75.240 | 86.40 | 89.58 | 87.060 | 82.140 | 85.800 | 98.82 | 97.80 | 104.64 | 112.80 |
| boczek wędzony - za 1kg | 16.58 | 17.13 | 19.25 | 20.09 | 20.15 | 19.62 | 19.91 | 22.28 | 23.31 | 24.79 | ... | 102.780 | 115.50 | 120.54 | 120.900 | 117.720 | 119.460 | 133.68 | 139.86 | 148.74 | 157.62 |
| botki męskie skórzane na podeszwie nieskórzanej - za 1parę | 188.46 | 195.95 | 211.60 | 223.58 | 227.91 | 227.47 | 225.02 | 218.93 | 223.76 | 228.65 | ... | 4702.800 | 5078.40 | 5365.92 | 5469.840 | 5459.280 | 5400.480 | 5254.32 | 5370.24 | 5487.60 | 5598.00 |
| bułka pszenna - za 50g | 0.40 | 0.44 | 0.44 | 0.43 | 0.42 | 0.41 | 0.41 | 0.42 | 0.44 | 0.48 | ... | 158.400 | 158.40 | 154.80 | 151.200 | 147.600 | 147.600 | 151.20 | 158.40 | 172.80 | 187.20 |
| centralne ogrzewanie lokali mieszkalnych - za 1m2 | 3.52 | 3.70 | 3.81 | 3.87 | 3.97 | 3.98 | 3.94 | 3.85 | 3.82 | 3.88 | ... | 1776.000 | 1828.80 | 1857.60 | 1905.600 | 1910.400 | 1891.200 | 1848.00 | 1833.60 | 1862.40 | 1896.00 |
| chleb pszenno-żytni - za 1kg | 3.24 | 3.58 | 3.75 | 3.89 | 3.92 | 3.98 | 3.96 | 4.01 | 4.13 | 4.41 | ... | 343.680 | 360.00 | 373.44 | 376.320 | 382.080 | 380.160 | 384.96 | 396.48 | 423.36 | 451.20 |

10 rows × 24 columns



```
In [12]: # Wycięcie niezbędnych kolumn do dalszych obliczeń i transpoza

lista = []
for number in range(0,11):
    position = 'kwota_'+ str(2010+number)
    lista.append(position)
lista
koszyk = data[lista]
koszyk = koszyk.transpose()
koszyk
```

Out[12]:

| ProduktLubUsługa | baleron gotowany - za 1kg | bilet do kina | bilet normalny na przejazd autobusem miejskim | bilet normalny na przejazd tramwajem | boczek surowy - za 1kg | boczek wędzony - za 1kg | botki męskie skórzane na podeszwie nieskórzanej - za 1parę | bułka pszenna - za 50g | centralne ogrzewanie lokali mieszkalnych - za 1m2 | chleb pszenno-żytni - za 1kg | ... | sól warzona biała workowana - za 1kg | strzyżenie włosów męskich | szynka wieprzowa gotowana - za 1kg | śledź solony, niepatroszony - za 1kg | śmietana o zawartości tłuszczu 18% - za 1l | truskawki mrożone - za 0,5kg | wino białe gronowe, wytrawne - za 0,75l | wi |
|------------------|---------------------------|---------------|---|--------------------------------------|------------------------|-------------------------|--|------------------------|---|------------------------------|-----|--------------------------------------|---------------------------|------------------------------------|--------------------------------------|--|------------------------------|---|----|
| kwota_2010 | 8.790 | 180.84 | 267.6 | 303.6 | 72.36 | 99.48 | 4523.04 | 144.0 | 1689.6 | 311.04 | ... | 0.372 | 174.84 | 393.66 | 23.184 | 32.82 | 91.44 | 434.88 | 8 |
| kwota_2011 | 8.945 | 187.44 | 280.8 | 315.6 | 75.24 | 102.78 | 4702.80 | 158.4 | 1776.0 | 343.68 | ... | 0.456 | 184.08 | 399.24 | 27.216 | 33.36 | 97.32 | 437.76 | 8 |
| kwota_2012 | 9.640 | 196.32 | 309.6 | 354.0 | 86.40 | 115.50 | 5078.40 | 158.4 | 1828.8 | 360.00 | ... | 0.480 | 192.12 | 421.74 | 33.120 | 33.54 | 99.60 | 442.56 | 9 |
| kwota_2013 | 9.900 | 204.72 | 326.4 | 393.6 | 89.58 | 120.54 | 5365.92 | 154.8 | 1857.6 | 373.44 | ... | 0.492 | 196.20 | 433.62 | 34.812 | 34.20 | 100.80 | 454.08 | 9 |
| kwota_2014 | 10.095 | 211.80 | 327.6 | 396.0 | 87.06 | 120.90 | 5469.84 | 151.2 | 1905.6 | 376.32 | ... | 0.480 | 201.96 | 439.02 | 35.028 | 34.50 | 99.60 | 460.80 | 10 |
| kwota_2015 | 9.775 | 218.76 | 325.2 | 398.4 | 82.14 | 117.72 | 5459.28 | 147.6 | 1910.4 | 382.08 | ... | 0.480 | 205.92 | 436.50 | 36.036 | 33.42 | 96.00 | 465.60 | 10 |
| kwota_2016 | 9.805 | 224.76 | 325.2 | 398.4 | 85.80 | 119.46 | 5400.48 | 147.6 | 1891.2 | 380.16 | ... | 0.468 | 210.60 | 441.36 | 38.088 | 33.12 | 93.96 | 468.96 | 11 |
| kwota_2017 | 10.540 | 233.76 | 322.8 | 399.6 | 98.82 | 133.68 | 5254.32 | 151.2 | 1848.0 | 384.96 | ... | 0.480 | 220.08 | 460.08 | 40.392 | 36.54 | 93.84 | 1151.52 | 11 |
| kwota_2018 | 10.940 | 237.60 | 328.8 | 403.2 | 97.80 | 139.86 | 5370.24 | 158.4 | 1833.6 | 396.48 | ... | 0.492 | 232.08 | 480.06 | 39.096 | 39.84 | 102.00 | 1183.68 | 12 |
| kwota_2019 | 11.860 | 240.24 | 332.4 | 408.0 | 104.64 | 148.74 | 5487.60 | 172.8 | 1862.4 | 423.36 | ... | 0.588 | 252.00 | 511.92 | 37.404 | 41.16 | 111.72 | 1205.28 | 13 |
| kwota_2020 | 13.200 | 232.08 | 344.4 | 427.2 | 113.94 | 167.46 | 5573.52 | 190.8 | 1876.8 | 449.28 | ... | 0.612 | 302.64 | 571.32 | 38.988 | 42.24 | 116.64 | 1176.00 | 14 |

11 rows × 54 columns



```
In [13]: # Dodanie kolumny z sumą za każdy rok

lista = []
for number in range(0,11):
    suma = koszyk.loc['kwota_' + str(2010+number)].sum()
    lista.append(round(suma,2))

suma_koszyk = pd.Series(lista)
koszyk.reset_index(inplace = True)
koszyk['suma_koszyk'] = suma_koszyk
koszyk.rename(columns = {'index':'Rok'}, inplace = True)
koszyk.set_index('Rok', inplace = True)
koszyk
```

Out[13]:

| ProduktLubUsługa | baleron gotowany - za 1kg | bilet do kina | bilet normalny na przejazd autobusem miejskim | bilet normalny na przejazd tramwajem | boczek surowy - za 1kg | boczek wędzony - za 1kg | botki męskie skórzane na podeszwie nieskórzanej - za 1parę | bułka pszenna - za 50g | centralne ogrzewanie lokali mieszkalnych - za 1m2 | chleb pszenno-żytni - za 1kg | ... | strzyżenie włosów męskich | szynka wieprzowa gotowana - za 1kg | śledź solony, niepatroszony - za 1kg | śmietana o zawartości tłuszczu 18% - za 1l | truskawki mrożone - za 0,5kg | wino białe gronowe, wytrawne - za 0,75l | wizyta u lekarza specjalisty | wódka czysta 40% - za 0,5 |
|------------------|---------------------------|---------------|---|--------------------------------------|------------------------|-------------------------|--|------------------------|---|------------------------------|-----|---------------------------|------------------------------------|--------------------------------------|--|------------------------------|---|------------------------------|---------------------------|
| Rok | | | | | | | | | | | | | | | | | | | |
| kwota_2010 | 8.790 | 180.84 | 267.6 | 303.6 | 72.36 | 99.48 | 4523.04 | 144.0 | 1689.6 | 311.04 | ... | 174.84 | 393.66 | 23.184 | 32.82 | 91.44 | 434.88 | 817.44 | 470.11 |
| kwota_2011 | 8.945 | 187.44 | 280.8 | 315.6 | 75.24 | 102.78 | 4702.80 | 158.4 | 1776.0 | 343.68 | ... | 184.08 | 399.24 | 27.216 | 33.36 | 97.32 | 437.76 | 873.72 | 477.61 |
| kwota_2012 | 9.640 | 196.32 | 309.6 | 354.0 | 86.40 | 115.50 | 5078.40 | 158.4 | 1828.8 | 360.00 | ... | 192.12 | 421.74 | 33.120 | 33.54 | 99.60 | 442.56 | 938.28 | 486.41 |
| kwota_2013 | 9.900 | 204.72 | 326.4 | 393.6 | 89.58 | 120.54 | 5365.92 | 154.8 | 1857.6 | 373.44 | ... | 196.20 | 433.62 | 34.812 | 34.20 | 100.80 | 454.08 | 986.88 | 488.81 |
| kwota_2014 | 10.095 | 211.80 | 327.6 | 396.0 | 87.06 | 120.90 | 5469.84 | 151.2 | 1905.6 | 376.32 | ... | 201.96 | 439.02 | 35.028 | 34.50 | 99.60 | 460.80 | 1038.12 | 520.01 |
| kwota_2015 | 9.775 | 218.76 | 325.2 | 398.4 | 82.14 | 117.72 | 5459.28 | 147.6 | 1910.4 | 382.08 | ... | 205.92 | 436.50 | 36.036 | 33.42 | 96.00 | 465.60 | 1070.28 | 601.41 |
| kwota_2016 | 9.805 | 224.76 | 325.2 | 398.4 | 85.80 | 119.46 | 5400.48 | 147.6 | 1891.2 | 380.16 | ... | 210.60 | 441.36 | 38.088 | 33.12 | 93.96 | 468.96 | 1113.24 | 589.21 |
| kwota_2017 | 10.540 | 233.76 | 322.8 | 399.6 | 98.82 | 133.68 | 5254.32 | 151.2 | 1848.0 | 384.96 | ... | 220.08 | 460.08 | 40.392 | 36.54 | 93.84 | 1151.52 | 1179.36 | 584.81 |
| kwota_2018 | 10.940 | 237.60 | 328.8 | 403.2 | 97.80 | 139.86 | 5370.24 | 158.4 | 1833.6 | 396.48 | ... | 232.08 | 480.06 | 39.096 | 39.84 | 102.00 | 1183.68 | 1247.64 | 585.31 |
| kwota_2019 | 11.860 | 240.24 | 332.4 | 408.0 | 104.64 | 148.74 | 5487.60 | 172.8 | 1862.4 | 423.36 | ... | 252.00 | 511.92 | 37.404 | 41.16 | 111.72 | 1205.28 | 1349.40 | 583.91 |
| kwota_2020 | 13.200 | 232.08 | 344.4 | 427.2 | 113.94 | 167.46 | 5573.52 | 190.8 | 1876.8 | 449.28 | ... | 302.64 | 571.32 | 38.988 | 42.24 | 116.64 | 1176.00 | 1516.44 | 615.81 |

11 rows × 55 columns

In [14]: *# wyodrębnienie kolumny z sumą i obliczenie rok-rocznej inflacji oraz sumarycznej od roku 2010*

```
koszyk.reset_index(inplace = True)
inflacja = koszyk[['Rok', 'suma_koszyk']]
inflacja.set_index('Rok', inplace = True)
lista = inflacja['suma_koszyk'].tolist()

#obliczenie iflacji rok-rocznej
infl_r_r = [0]
for number in range(0, len(lista)-1):
    value = round(((lista[number+1]/lista[number])*100)-100,2)
    infl_r_r.append(value)

# obliczenie inflacji sumarycznej
infl_sum = [0]
for number in range(0,len(lista)-1):
    value = round(((lista[number+1]/lista[0])*100)-100,2)
    infl_sum.append(value)

# utworzenie nowych kolumn
inflacja.reset_index(inplace = True)
inflacja['inflacja_r_r'] = infl_r_r
inflacja['inflacja_od_2010'] = infl_sum
inflacja.set_index('Rok', inplace = True)
```

In [15]: *# zmiana nazw w indeksie 'Rok': np. kwota_2014 ---> 2014*

```
inflacja.reset_index(inplace = True)
inflacja['Rok'] = inflacja['Rok'].str.replace('kwota_', '')
inflacja.set_index('Rok', inplace = True)
inflacja
```

Out[15]:

| ProduktLubUsługa | suma_koszyk | inflacja_r_r | inflacja_od_2010 |
|------------------|-------------|--------------|------------------|
| Rok | | | |
| 2010 | 16318.71 | 0.00 | 0.00 |
| 2011 | 17047.39 | 4.47 | 4.47 |
| 2012 | 18111.97 | 6.24 | 10.99 |
| 2013 | 18697.57 | 3.23 | 14.58 |
| 2014 | 18939.38 | 1.29 | 16.06 |
| 2015 | 19003.33 | 0.34 | 16.45 |
| 2016 | 19013.70 | 0.05 | 16.51 |
| 2017 | 19840.77 | 4.35 | 21.58 |
| 2018 | 20240.86 | 2.02 | 24.03 |
| 2019 | 20886.57 | 3.19 | 27.99 |
| 2020 | 21741.04 | 4.09 | 33.23 |


```
In [17]: # Prezentacja danych na wykresie
plt.figure( figsize = (15,9))
plt.title("Inflacja w Polsce dla wybranego koszyka dóbr i usług")
plt.ylabel("Inflacja %")
plt.xlabel("Rok")
plt.plot(inflacja['inflacja_r_r'], marker = 'o', linestyle = ':', color = 'r', label = 'Inflacja r/r')
plt.plot(inflacja['inflacja_od_2010'],linestyle = 'dashed', color = 'g', marker = 'o', label = 'Inflacja sumaryczna')
plt.grid(linestyle = '--')
plt.legend(loc = 'upper left', fontsize = 15)
plt.xticks(rotation = 45)
plt.show()
```

