1-employment-rate-2011-2020

May 5, 2024

0.0.1 EMPLOYMENT RATE ANALYSIS IN EU FROM 2011 TO 2020

ABOUT THE DATASET   Source organisation URL https://ec.europa.eu/eurostat Source activity EU Labour Force Survey (EU-LFS) Source reference EU Labour Force Survey (EU LFS) data; downloaded from Eurostat's online database as table t2020_10 Data type Survey data Harmonised Harmonised at EU level Time format Years

Sex Women Age From 20 to 64 years Unit Percentage of total population

```python
[1]: # Libraries import
     import pandas as pd
     import numpy as np
```

```python
[2]: #loading the dataset for Employment Rate in the EU
     employment_rate = pd.read_excel("Employment Rate in the EU.xlsx")
```

```python
[3]: employment_rate
```

```
[3]:           Geographic region\Time  2003  2004  2005  2006  2007  \
     0                         Austria   NaN   NaN   NaN   NaN   NaN
     1                         Belgium   NaN   NaN   NaN   NaN   NaN
     2                        Bulgaria   NaN   NaN   NaN   NaN   NaN
     3                     Switzerland   NaN   NaN   NaN   NaN   NaN
     4                          Cyprus   NaN   NaN   NaN   NaN   NaN
     5                         Czechia   NaN   NaN   NaN   NaN   NaN
     6                         Germany   NaN   NaN   NaN   NaN   NaN
     7                         Denmark   NaN   NaN   NaN   NaN   NaN
     8           Euro area (19 countries)   NaN   NaN   NaN   NaN   NaN
     9                         Estonia   NaN   NaN   NaN   NaN   NaN
     10                         Greece   NaN   NaN   NaN   NaN   NaN
     11                          Spain   NaN   NaN   NaN   NaN   NaN
     12  European Union - 27 countries (from 2020)   NaN   NaN   NaN   NaN   NaN
     13                        Finland   NaN   NaN   NaN   NaN   NaN
     14                         France  64.6  64.6  64.8  65.0  65.9
     15                        Croatia   NaN   NaN   NaN   NaN   NaN
     16                        Hungary   NaN   NaN   NaN   NaN   NaN
     17                        Ireland   NaN   NaN   NaN   NaN   NaN
     18                        Iceland   NaN   NaN   NaN   NaN   NaN
     19                          Italy   NaN   NaN   NaN   NaN   NaN
```

20	Lithuania	NaN	NaN	NaN	NaN	NaN
21	Luxembourg	NaN	NaN	NaN	NaN	NaN
22	Latvia	NaN	NaN	NaN	NaN	NaN
23	Montenegro	NaN	NaN	NaN	NaN	NaN
24	North Macedonia	NaN	NaN	NaN	NaN	NaN
25	Malta	NaN	NaN	NaN	NaN	NaN
26	Netherlands	NaN	NaN	NaN	NaN	NaN
27	Norway	NaN	NaN	NaN	NaN	NaN
28	Poland	NaN	NaN	NaN	NaN	NaN
29	Portugal	NaN	NaN	NaN	NaN	NaN
30	Romania	NaN	NaN	NaN	NaN	NaN
31	Serbia	NaN	NaN	NaN	NaN	NaN
32	Sweden	NaN	NaN	NaN	NaN	NaN
33	Slovenia	NaN	NaN	NaN	NaN	NaN
34	Slovakia	NaN	NaN	NaN	NaN	NaN
35	Türkiye	NaN	NaN	NaN	NaN	NaN

	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019 \
0	NaN	68.2	68.8	69.2	69.6	70.0	70.1	70.2	70.9	71.4	71.7	72.4
1	NaN	61.0	61.6	61.5	61.7	62.1	62.9	63.0	63.0	63.6	65.5	66.5
2	NaN	63.6	60.3	59.4	59.8	60.2	61.5	63.3	63.6	66.8	67.8	70.2
3	NaN	NaN	73.2	74.6	75.3	75.3	76.2	76.8	77.5	77.4	77.9	78.6
4	NaN	68.3	68.8	67.7	64.8	62.2	63.9	64.0	64.1	66.2	68.9	70.1
5	NaN	61.4	60.9	61.7	62.5	63.8	64.7	66.4	68.6	70.5	72.2	72.7
6	NaN	67.8	68.8	70.3	70.7	71.5	72.1	72.6	73.5	74.2	74.8	75.5
7	NaN	73.1	72.0	71.4	71.1	71.2	71.0	71.5	72.5	73.2	73.9	74.7
8	NaN	61.7	61.6	62.0	61.9	62.0	62.7	63.4	64.4	65.4	66.4	67.2
9	NaN	69.9	67.4	68.6	71.1	71.6	71.9	73.7	73.7	76.8	76.8	77.5
10	NaN	52.4	51.2	48.1	44.6	42.9	43.8	45.6	46.2	47.4	48.3	50.5
11	NaN	56.8	56.3	56.1	54.6	53.8	54.8	56.4	58.1	59.6	61.0	62.1
12	NaN	60.6	60.7	60.9	61.0	61.1	62.0	63.0	64.1	65.2	66.3	67.1
13	NaN	71.5	70.3	70.7	71.3	71.0	71.2	70.7	70.8	71.3	73.4	74.5
14	66.5	65.9	65.8	65.6	66.0	66.4	67.0	67.4	67.7	68.1	68.9	69.4
15	NaN	58.0	56.4	53.6	52.6	52.8	54.2	55.9	56.6	58.3	60.1	61.5
16	NaN	58.6	59.0	59.2	60.4	61.2	64.1	66.2	68.9	69.9	71.4	72.1
17	NaN	62.8	61.0	60.1	60.2	61.2	62.3	63.7	65.3	66.9	67.9	68.8
18	NaN	77.8	77.6	77.9	79.1	79.5	81.7	83.3	84.4	84.5	83.2	83.0
19	NaN	49.5	49.2	49.6	50.1	49.6	50.1	50.5	51.6	52.5	53.2	53.9
20	NaN	67.2	65.0	66.6	67.9	68.6	70.6	72.2	74.3	75.5	76.7	77.4
21	NaN	61.5	62.0	61.9	64.1	63.9	65.5	65.0	65.1	67.5	68.0	68.1
22	NaN	66.2	64.2	65.1	66.2	67.5	68.4	70.5	71.7	72.6	74.7	75.5
23	NaN	NaN	NaN	44.4	46.2	47.5	49.7	51.6	51.3	51.4	52.9	54.2
24	NaN	37.1	37.5	38.8	38.7	40.7	40.8	42.1	42.5	43.7	45.2	48.4
25	NaN	40.0	41.6	43.8	48.0	51.7	54.3	55.3	58.0	60.6	64.1	65.8
26	NaN	70.9	70.9	71.5	72.0	71.7	70.8	71.9	72.7	73.9	75.2	76.5
27	NaN	77.9	76.9	77.1	77.3	77.1	77.1	76.7	76.7	76.2	76.5	76.8
28	NaN	55.6	55.6	55.8	56.2	56.3	58.2	59.9	61.5	63.0	64.5	64.9

```
29    NaN  62.3  62.1  62.4  60.8  60.5  62.9  64.8  66.4  69.0  71.5  72.2
30    NaN  45.2  47.6  47.7  48.2  48.2  49.2  50.3  51.4  54.0  54.5  55.4
31    NaN   NaN  44.5  42.6  42.3  44.3  45.7  46.8  50.1  52.6  53.9  56.3
32    NaN  74.9  74.2  75.7  76.1  76.4  76.9  77.6  78.4  79.0  79.4  78.9
33    NaN  67.6  66.0  64.4  64.2  62.6  63.2  64.2  66.2  69.3  71.2  72.5
34    NaN  62.4  61.6  61.6  61.5  62.0  62.9  64.6  67.2  69.4  70.3  71.7
35    NaN  25.8  28.0  29.7  30.9  31.8  31.6  32.5  33.2  34.5  35.2  34.4

      2020
0    70.6
1    65.6
2    68.5
3    78.3
4    69.1
5    71.9
6    74.5
7    74.3
8    66.1
9    76.9
10   48.7
11   60.0
12   66.2
13   73.8
14   69.3
15   61.3
16   71.9
17   65.8
18   79.2
19   52.1
20   75.8
21   68.5
22   75.0
23   48.8
24   49.0
25   67.8
26   76.6
27   76.5
28   65.2
29   71.4
30   55.4
31   57.0
32   77.4
33   71.7
34   70.9
35   32.0
```

DATA CLEANING AND EXPLORATION The data for our employment rate in EU comes with a lot of NAN values which will hinder our analysis. The next step for our analysis is to drop all NAN values in our Dataset.

```
[4]: #dropping of NAN values in our dataset
     employment_rate = employment_rate.
      ↪drop(columns=['2003',          '2004',          '2005',          '2006',          '2007',          '2(
```

```
[5]: employment_rate
```

```
[5]:                     Geographic region\Time  2009   2010   2011   2012   2013  \
     0                                  Austria  68.2   68.8   69.2   69.6   70.0
     1                                  Belgium  61.0   61.6   61.5   61.7   62.1
     2                                 Bulgaria  63.6   60.3   59.4   59.8   60.2
     3                              Switzerland   NaN   73.2   74.6   75.3   75.3
     4                                   Cyprus  68.3   68.8   67.7   64.8   62.2
     5                                  Czechia  61.4   60.9   61.7   62.5   63.8
     6                                  Germany  67.8   68.8   70.3   70.7   71.5
     7                                  Denmark  73.1   72.0   71.4   71.1   71.2
     8                   Euro area (19 countries)  61.7   61.6   62.0   61.9   62.0
     9                                  Estonia  69.9   67.4   68.6   71.1   71.6
     10                                  Greece  52.4   51.2   48.1   44.6   42.9
     11                                   Spain  56.8   56.3   56.1   54.6   53.8
     12   European Union - 27 countries (from 2020)  60.6   60.7   60.9   61.0   61.1
     13                                 Finland  71.5   70.3   70.7   71.3   71.0
     14                                  France  65.9   65.8   65.6   66.0   66.4
     15                                 Croatia  58.0   56.4   53.6   52.6   52.8
     16                                 Hungary  58.6   59.0   59.2   60.4   61.2
     17                                 Ireland  62.8   61.0   60.1   60.2   61.2
     18                                 Iceland  77.8   77.6   77.9   79.1   79.5
     19                                   Italy  49.5   49.2   49.6   50.1   49.6
     20                               Lithuania  67.2   65.0   66.6   67.9   68.6
     21                              Luxembourg  61.5   62.0   61.9   64.1   63.9
     22                                  Latvia  66.2   64.2   65.1   66.2   67.5
     23                              Montenegro   NaN    NaN   44.4   46.2   47.5
     24                          North Macedonia  37.1   37.5   38.8   38.7   40.7
     25                                   Malta  40.0   41.6   43.8   48.0   51.7
     26                             Netherlands  70.9   70.9   71.5   72.0   71.7
     27                                  Norway  77.9   76.9   77.1   77.3   77.1
     28                                  Poland  55.6   55.6   55.8   56.2   56.3
     29                                Portugal  62.3   62.1   62.4   60.8   60.5
     30                                 Romania  45.2   47.6   47.7   48.2   48.2
     31                                  Serbia   NaN   44.5   42.6   42.3   44.3
     32                                  Sweden  74.9   74.2   75.7   76.1   76.4
     33                                Slovenia  67.6   66.0   64.4   64.2   62.6
     34                                Slovakia  62.4   61.6   61.6   61.5   62.0
     35                                 Türkiye  25.8   28.0   29.7   30.9   31.8
```

```
      2014  2015  2016  2017  2018  2019  2020
0     70.1  70.2  70.9  71.4  71.7  72.4  70.6
1     62.9  63.0  63.0  63.6  65.5  66.5  65.6
2     61.5  63.3  63.6  66.8  67.8  70.2  68.5
3     76.2  76.8  77.5  77.4  77.9  78.6  78.3
4     63.9  64.0  64.1  66.2  68.9  70.1  69.1
5     64.7  66.4  68.6  70.5  72.2  72.7  71.9
6     72.1  72.6  73.5  74.2  74.8  75.5  74.5
7     71.0  71.5  72.5  73.2  73.9  74.7  74.3
8     62.7  63.4  64.4  65.4  66.4  67.2  66.1
9     71.9  73.7  73.7  76.8  76.8  77.5  76.9
10    43.8  45.6  46.2  47.4  48.3  50.5  48.7
11    54.8  56.4  58.1  59.6  61.0  62.1  60.0
12    62.0  63.0  64.1  65.2  66.3  67.1  66.2
13    71.2  70.7  70.8  71.3  73.4  74.5  73.8
14    67.0  67.4  67.7  68.1  68.9  69.4  69.3
15    54.2  55.9  56.6  58.3  60.1  61.5  61.3
16    64.1  66.2  68.9  69.9  71.4  72.1  71.9
17    62.3  63.7  65.3  66.9  67.9  68.8  65.8
18    81.7  83.3  84.4  84.5  83.2  83.0  79.2
19    50.1  50.5  51.6  52.5  53.2  53.9  52.1
20    70.6  72.2  74.3  75.5  76.7  77.4  75.8
21    65.5  65.0  65.1  67.5  68.0  68.1  68.5
22    68.4  70.5  71.7  72.6  74.7  75.5  75.0
23    49.7  51.6  51.3  51.4  52.9  54.2  48.8
24    40.8  42.1  42.5  43.7  45.2  48.4  49.0
25    54.3  55.3  58.0  60.6  64.1  65.8  67.8
26    70.8  71.9  72.7  73.9  75.2  76.5  76.6
27    77.1  76.7  76.7  76.2  76.5  76.8  76.5
28    58.2  59.9  61.5  63.0  64.5  64.9  65.2
29    62.9  64.8  66.4  69.0  71.5  72.2  71.4
30    49.2  50.3  51.4  54.0  54.5  55.4  55.4
31    45.7  46.8  50.1  52.6  53.9  56.3  57.0
32    76.9  77.6  78.4  79.0  79.4  78.9  77.4
33    63.2  64.2  66.2  69.3  71.2  72.5  71.7
34    62.9  64.6  67.2  69.4  70.3  71.7  70.9
35    31.6  32.5  33.2  34.5  35.2  34.4  32.0
```

Next is to drop the row number 8 and 12 because they contain data which doesn't correspond with our dataset

```
[6]: employment_rate.drop([8,12], axis=0, inplace=True)
     employment_rate
```

```
[6]:   Geographic region\Time  2009  2010  2011  2012  2013  2014  2015  2016  \
     0               Austria   68.2  68.8  69.2  69.6  70.0  70.1  70.2  70.9
     1               Belgium   61.0  61.6  61.5  61.7  62.1  62.9  63.0  63.0
```

2	Bulgaria	63.6	60.3	59.4	59.8	60.2	61.5	63.3	63.6
3	Switzerland	NaN	73.2	74.6	75.3	75.3	76.2	76.8	77.5
4	Cyprus	68.3	68.8	67.7	64.8	62.2	63.9	64.0	64.1
5	Czechia	61.4	60.9	61.7	62.5	63.8	64.7	66.4	68.6
6	Germany	67.8	68.8	70.3	70.7	71.5	72.1	72.6	73.5
7	Denmark	73.1	72.0	71.4	71.1	71.2	71.0	71.5	72.5
9	Estonia	69.9	67.4	68.6	71.1	71.6	71.9	73.7	73.7
10	Greece	52.4	51.2	48.1	44.6	42.9	43.8	45.6	46.2
11	Spain	56.8	56.3	56.1	54.6	53.8	54.8	56.4	58.1
13	Finland	71.5	70.3	70.7	71.3	71.0	71.2	70.7	70.8
14	France	65.9	65.8	65.6	66.0	66.4	67.0	67.4	67.7
15	Croatia	58.0	56.4	53.6	52.6	52.8	54.2	55.9	56.6
16	Hungary	58.6	59.0	59.2	60.4	61.2	64.1	66.2	68.9
17	Ireland	62.8	61.0	60.1	60.2	61.2	62.3	63.7	65.3
18	Iceland	77.8	77.6	77.9	79.1	79.5	81.7	83.3	84.4
19	Italy	49.5	49.2	49.6	50.1	49.6	50.1	50.5	51.6
20	Lithuania	67.2	65.0	66.6	67.9	68.6	70.6	72.2	74.3
21	Luxembourg	61.5	62.0	61.9	64.1	63.9	65.5	65.0	65.1
22	Latvia	66.2	64.2	65.1	66.2	67.5	68.4	70.5	71.7
23	Montenegro	NaN	NaN	44.4	46.2	47.5	49.7	51.6	51.3
24	North Macedonia	37.1	37.5	38.8	38.7	40.7	40.8	42.1	42.5
25	Malta	40.0	41.6	43.8	48.0	51.7	54.3	55.3	58.0
26	Netherlands	70.9	70.9	71.5	72.0	71.7	70.8	71.9	72.7
27	Norway	77.9	76.9	77.1	77.3	77.1	77.1	76.7	76.7
28	Poland	55.6	55.6	55.8	56.2	56.3	58.2	59.9	61.5
29	Portugal	62.3	62.1	62.4	60.8	60.5	62.9	64.8	66.4
30	Romania	45.2	47.6	47.7	48.2	48.2	49.2	50.3	51.4
31	Serbia	NaN	44.5	42.6	42.3	44.3	45.7	46.8	50.1
32	Sweden	74.9	74.2	75.7	76.1	76.4	76.9	77.6	78.4
33	Slovenia	67.6	66.0	64.4	64.2	62.6	63.2	64.2	66.2
34	Slovakia	62.4	61.6	61.6	61.5	62.0	62.9	64.6	67.2
35	Türkiye	25.8	28.0	29.7	30.9	31.8	31.6	32.5	33.2

	2017	2018	2019	2020
0	71.4	71.7	72.4	70.6
1	63.6	65.5	66.5	65.6
2	66.8	67.8	70.2	68.5
3	77.4	77.9	78.6	78.3
4	66.2	68.9	70.1	69.1
5	70.5	72.2	72.7	71.9
6	74.2	74.8	75.5	74.5
7	73.2	73.9	74.7	74.3
9	76.8	76.8	77.5	76.9
10	47.4	48.3	50.5	48.7
11	59.6	61.0	62.1	60.0
13	71.3	73.4	74.5	73.8
14	68.1	68.9	69.4	69.3

```
15   58.3   60.1   61.5   61.3
16   69.9   71.4   72.1   71.9
17   66.9   67.9   68.8   65.8
18   84.5   83.2   83.0   79.2
19   52.5   53.2   53.9   52.1
20   75.5   76.7   77.4   75.8
21   67.5   68.0   68.1   68.5
22   72.6   74.7   75.5   75.0
23   51.4   52.9   54.2   48.8
24   43.7   45.2   48.4   49.0
25   60.6   64.1   65.8   67.8
26   73.9   75.2   76.5   76.6
27   76.2   76.5   76.8   76.5
28   63.0   64.5   64.9   65.2
29   69.0   71.5   72.2   71.4
30   54.0   54.5   55.4   55.4
31   52.6   53.9   56.3   57.0
32   79.0   79.4   78.9   77.4
33   69.3   71.2   72.5   71.7
34   69.4   70.3   71.7   70.9
35   34.5   35.2   34.4   32.0
```

EXPLORE OUR DATASET

[7]: ```python
#checking the shape of the dataframe
employment_rate.shape
```

[7]: (34, 13)

[8]: ```python
#Basic info about the DataFramE
employment_rate.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34 entries, 0 to 35
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Geographic region\Time  34 non-null     object
 1   2009                  31 non-null     float64
 2   2010                  33 non-null     float64
 3   2011                  34 non-null     float64
 4   2012                  34 non-null     float64
 5   2013                  34 non-null     float64
 6   2014                  34 non-null     float64
 7   2015                  34 non-null     float64
 8   2016                  34 non-null     float64
 9   2017                  34 non-null     float64
 10  2018                  34 non-null     float64
```

```
11  2019                         34 non-null    float64
12  2020                         34 non-null    float64
dtypes: float64(12), object(1)
memory usage: 3.7+ KB
```

[9]:
```
#Checking for duplicates
employment_rate.duplicated().sum()
```

[9]: 0

[10]:
```
#Checking for null values
employment_rate.isnull().any()
```

[10]:
```
Geographic region\Time    False
2009                       True
2010                       True
2011                      False
2012                      False
2013                      False
2014                      False
2015                      False
2016                      False
2017                      False
2018                      False
2019                      False
2020                      False
dtype: bool
```

Notice there is null values in our dataset which can be seen in our 2009 and 2010 columns. Next we have to drop the null values in these columns

[11]:
```
employment_rate = employment_rate.dropna()
```

[12]:
```
#Rechecking for null values again after dropping the null values in those␣
 ↪various columns
employment_rate.isnull().sum()
```

[12]:
```
Geographic region\Time    0
2009                      0
2010                      0
2011                      0
2012                      0
2013                      0
2014                      0
2015                      0
2016                      0
2017                      0
2018                      0
```

```
2019                        0
2020                        0
dtype: int64
```

[13]: ```python
#rechecking for null values
employment_rate.isnull().any()
```

[13]: ```
Geographic region\Time    False
2009                      False
2010                      False
2011                      False
2012                      False
2013                      False
2014                      False
2015                      False
2016                      False
2017                      False
2018                      False
2019                      False
2020                      False
dtype: bool
```

Now we have thoroughly checked our dataset and removed all outliers than can hinder our analysis. Our dtaset is finally clean

TASK ANALYSIS

QUESTION 1:

What is the average employment rate of women in the EU from 2011-2020?

[14]: ```python
employment_rate.head()
```

[14]: 
| | Geographic region\Time | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 | 2016 | \ |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Austria | 68.2 | 68.8 | 69.2 | 69.6 | 70.0 | 70.1 | 70.2 | 70.9 | |
| 1 | Belgium | 61.0 | 61.6 | 61.5 | 61.7 | 62.1 | 62.9 | 63.0 | 63.0 | |
| 2 | Bulgaria | 63.6 | 60.3 | 59.4 | 59.8 | 60.2 | 61.5 | 63.3 | 63.6 | |
| 4 | Cyprus | 68.3 | 68.8 | 67.7 | 64.8 | 62.2 | 63.9 | 64.0 | 64.1 | |
| 5 | Czechia | 61.4 | 60.9 | 61.7 | 62.5 | 63.8 | 64.7 | 66.4 | 68.6 | |

| | 2017 | 2018 | 2019 | 2020 |
|---|---|---|---|---|
| 0 | 71.4 | 71.7 | 72.4 | 70.6 |
| 1 | 63.6 | 65.5 | 66.5 | 65.6 |
| 2 | 66.8 | 67.8 | 70.2 | 68.5 |
| 4 | 66.2 | 68.9 | 70.1 | 69.1 |
| 5 | 70.5 | 72.2 | 72.7 | 71.9 |

```
[15]: employment_rate_average = employment_rate[["2011", "2012", "2013", "2014",␣
      ↪"2015", "2016", "2017", "2018", "2019", "2020"]].mean()
```

```
[16]: employment_rate_average.round(2)
```

```
[16]: 2011    61.06
      2012    61.36
      2013    61.61
      2014    62.57
      2015    63.61
      2016    64.67
      2017    66.11
      2018    67.29
      2019    68.19
      2020    67.31
      dtype: float64
```

```
[17]: #Converting the employment_rate_average for each year into pandas dataframe for␣
      ↪easy save into excel file
      year_2011 = [61.06]
      year_2012 = [61.36]
      year_2013 = [61.61]
      year_2014 = [62.57]
      year_2015 = [63.61]
      year_2016 = [64.67]
      year_2017 = [66.11]
      year_2018 = [67.29]
      year_2019 = [68.19]
      year_2020 = [67.31]
      df = pd.DataFrame([year_2011, year_2012, year_2013, year_2014, year_2015,␣
       ↪year_2016, year_2017, year_2018, year_2019, year_2020])
      df = df.transpose()
      df.columns = ["year_2011", "year_2012", "year_2013", "year_2014", "year_2015",␣
       ↪"year_2016", "year_2017", "year_2018", "year_2019", "year2020"]
      df.head()
```

```
[17]:    year_2011  year_2012  year_2013  year_2014  year_2015  year_2016  \
      0      61.06      61.36      61.61      62.57      63.61      64.67

         year_2017  year_2018  year_2019  year2020
      0      66.11      67.29      68.19     67.31
```

```
[18]: employment_rate_average = df
      employment_rate_average.columns = ['2011', '2012', '2013', '2014', '2015',␣
       ↪'2016', '2017', '2018', '2019', '2020']
      employment_rate_average
```

```
[18]:      2011    2012    2013    2014    2015    2016    2017    2018    2019    2020
      0  61.06   61.36   61.61   62.57   63.61   64.67   66.11   67.29   68.19   67.31
```

```
[19]:  # Save dataframe to Excel Workbook
       employment_rate_average.to_excel("employment_rate_average_2011_to_2020.xlsx",↳
       ↳index = False)
```

```
[ ]:
```

QUESTION 2:

How is the rate of change for women employment in the EU from 2011-2020 To find the rate of change for each year from 2011 to 2020, we have to find the total percentage sum for each year.

```
[20]:  #To find the total percent sum for each year, we have to drop the string column↳
       ↳in order for our syntax to read our data
       employment_rate_drop = employment_rate.drop(columns=['Geographic↳
       ↳region\Time','2009','2010'])
       employment_rate_drop.head()
```

```
[20]:      2011   2012   2013   2014   2015   2016   2017   2018   2019   2020
       0  69.2   69.6   70.0   70.1   70.2   70.9   71.4   71.7   72.4   70.6
       1  61.5   61.7   62.1   62.9   63.0   63.0   63.6   65.5   66.5   65.6
       2  59.4   59.8   60.2   61.5   63.3   63.6   66.8   67.8   70.2   68.5
       4  67.7   64.8   62.2   63.9   64.0   64.1   66.2   68.9   70.1   69.1
       5  61.7   62.5   63.8   64.7   66.4   68.6   70.5   72.2   72.7   71.9
```

```
[21]:  #Total percent sum for each year
       total_percent_sum = employment_rate_drop.sum(axis=0)
       total_percent_sum
```

```
[21]:  2011      1892.8
       2012      1902.3
       2013      1910.0
       2014      1939.7
       2015      1972.0
       2016      2004.8
       2017      2049.4
       2018      2086.0
       2019      2113.9
       2020      2086.7
       dtype: float64
```

```
[22]:  #Converting total percent sum for each year into pandas dataframe for easy save↳
       ↳into excel file
       year_2011 = [1892.8]
```

```
year_2012 = [1902.3]
year_2013 = [1910.0]
year_2014 = [1939.7]
year_2015 = [1972.0]
year_2016 = [2004.8]
year_2017 = [2049.4]
year_2018 = [2086.0]
year_2019 = [2113.9]
year_2020 = [2086.7]
data = pd.DataFrame([year_2011, year_2012, year_2013, year_2014, year_2015,␣
 ↪year_2016, year_2017, year_2018, year_2019, year_2020])
data = data.transpose()
data.columns = ["year_2011", "year_2012", "year_2013", "year_2014",␣
 ↪"year_2015", "year_2016", "year_2017", "year_2018", "year_2019", "year2020"]
data.head()
```

[22]:   year_2011 year_2012 year_2013 year_2014 year_2015 year_2016 \
      0   1892.8   1902.3   1910.0   1939.7   1972.0   2004.8

          year_2017 year_2018 year_2019 year2020
      0   2049.4   2086.0   2113.9   2086.7

[23]:
```
percent_sum = data
percent_sum.columns = ['2011', '2012', '2013', '2014', '2015', '2016', '2017',␣
 ↪'2018', '2019', '2020']
percent_sum
```

[23]:   2011   2012   2013   2014   2015   2016   2017   2018   2019 \
      0   1892.8 1902.3 1910.0 1939.7 1972.0 2004.8 2049.4 2086.0 2113.9

            2020
      0   2086.7

[24]:
```
# Save dataframe to Excel Workbook
percent_sum.to_excel("total_percent_sum_2011_to_2020.xlsx", index = False)
```

[ ]:

Next, is to find the rate of change for women employment in the EU from 2011-2020

[25]:
```
#The change in employment rate from 2011 to 2020
total_percent_sum.pct_change().round(3)
```

[25]: 2011   NaN
      2012   0.005
      2013   0.004
      2014   0.016

```
2015    0.017
2016    0.017
2017    0.022
2018    0.018
2019    0.013
2020   -0.013
dtype: float64
```

[26]:
```python
#Converting the change in years into pandas dataframe for easy save into excel␣
 ↪file
year_2012 = [0.005]
year_2013 = [0.004]
year_2014 = [0.016]
year_2015 = [0.017]
year_2016 = [0.017]
year_2017 = [0.022]
year_2018 = [0.018]
year_2019 = [0.013]
year_2020 = [-0.013]
data1 = pd.DataFrame([year_2012, year_2013, year_2014, year_2015, year_2016,␣
 ↪year_2017, year_2018, year_2019, year_2020])
data1 = data1.transpose()
data1.columns = ["year_2012", "year_2013", "year_2014", "year_2015",␣
 ↪"year_2016", "year_2017", "year_2018", "year_2019", "year2020"]
data1.head()
```

[26]:
```
   year_2012  year_2013  year_2014  year_2015  year_2016  year_2017  \
0      0.005      0.004      0.016      0.017      0.017      0.022

   year_2018  year_2019  year2020
0      0.018      0.013    -0.013
```

[27]:
```python
change = data1
change.columns = ['2012', '2013', '2014', '2015', '2016', '2017', '2018',␣
 ↪'2019', '2020']
```

[28]:
```python
change
```

[28]:
```
    2012   2013   2014   2015   2016   2017   2018   2019   2020
0  0.005  0.004  0.016  0.017  0.017  0.022  0.018  0.013 -0.013
```

[ ]:
```python
# Save dataframe to Excel Workbook
change.to_excel("change_in_employment_rate_from_2011_to_2020.xlsx", index =␣
 ↪False)
```

MY VISUALIZATION WILL BE ON TABLEAU

[ ]: