FABIZ, IMBA

RESEARCH METHODS PROJECT

UNEMPLOYMENT ANALYSIS IN ROMANIA

TEAM MEMBERS

01 AMELIA DARIA NISTOR

02 SUNITA MARIA GEORGESCU

03 DARIA ELISA VUC

INTRODUCTION

The tendency of the unemployment rate to be unstable is because many job seekers are not satisfied with the salary and benefits that the employers provide, or they do not have the required skills by employers due to lack of experience.

It is very important for a country to find solutions for the problem of unemployment, because it determines the level of satisfaction among people and influences the growth of the economy.

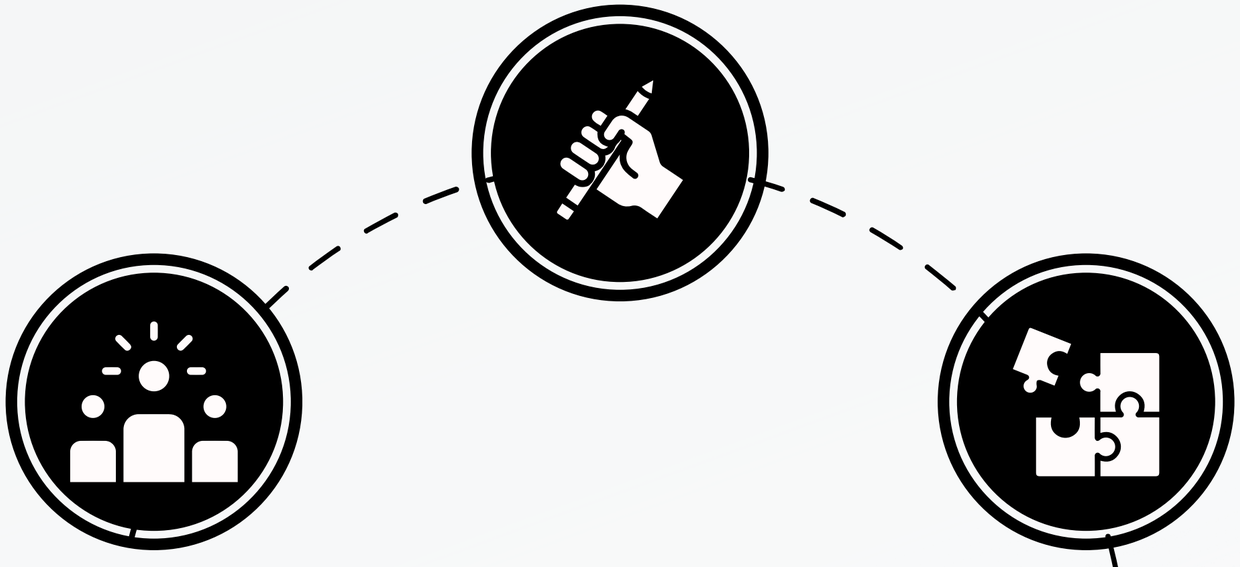
DATA&RESEARCH QUESTION

- Data taken from the World Bank website
- Data from 1992 to 2022 (30 years) for a relevant overview

Unemployment rate is the **dependent variable** Y

The **independent variables** (X) that we test are:

- interest rate
- government expenditure
- population growth
- foreign direct investment



What are the most relevant variables that influence the rate of unemployment in Romania among government expenditure, foreign direct investment, inflation rate, population growth and lending interest rate (loans), from a statistical point of view?

IMPORT DATA

- Import Libraries

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

- Import Dataset

data_df=pd.read_csv('/content/drive/MyDrive/UnemploymentandVariablesDatasetFinal.csv')
data_df.head()

	Foreign direct investment	Government Expenditure	Interest rate	Population Growth	Unemployment rate
0	0.306508	3587916667	211.205588	-0.903463	8.260
1	0.356585	3253996448	255.166862	-0.136109	8.373
2	1.133915	4139835296	136.759393	-0.145379	8.170
3	1.119418	4584710419	32.242485	-0.202319	8.010
4	0.712022	4311709374	38.829301	-0.288129	6.740

CORRELATION INDEX

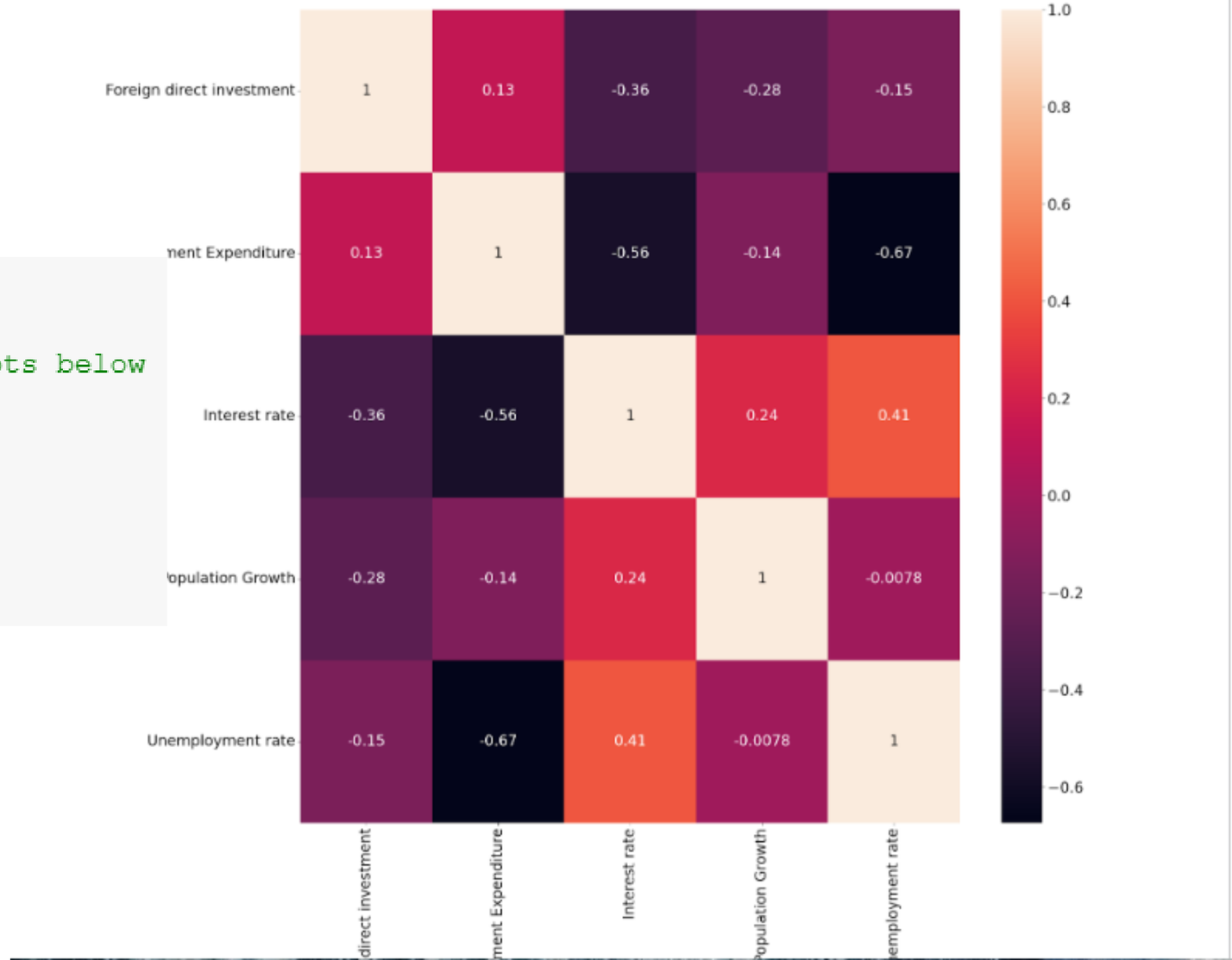
```
corrolation_list = ['Foreign direct investment', 'Government Expenditure','Interest rate','Population Growth','Unemployment rate']
up_data = data_df[corrolation_list].corr()
display(up_data)
```

	Foreign direct investment	Government Expenditure	Interest rate	Population Growth	Unemployment rate
Foreign direct investment	1.000000	0.128906	-0.363938	-0.279338	-0.145553
Government Expenditure	0.128906	1.000000	-0.564051	-0.136761	-0.673612
Interest rate	-0.363938	-0.564051	1.000000	0.235475	0.408903
Population Growth	-0.279338	-0.136761	0.235475	1.000000	-0.007826
Unemployment rate	-0.145553	-0.673612	0.408903	-0.007826	1.000000

CORRELATION HEATMAP

```
import seaborn as sns
import matplotlib.pyplot as plt
# to help you visualize these correlations, see the plots below
# the lighter the colour the higher the correlation

plt.rcParams.update({'font.size': 20})
plt.figure(figsize=(20,20))
sns.heatmap(up_data, annot=True);
```



MULTIPLE LINEAR REGRESSION

```
x=data_df.drop(['Unemployment rate'], axis=1).values
y=data_df ['Unemployment rate'].values
```

```
print(x)
```

```
[[ 3.06508326e-01  3.58791667e+09  2.11205588e+02 -9.03462928e-01]
 [ 3.56585212e-01  3.25399645e+09  2.55166862e+02 -1.36109142e-01]
 [ 1.13391484e+00  4.13983530e+09  1.36759393e+02 -1.45379065e-01]
 [ 1.11941808e+00  4.58471042e+09  3.22424847e+01 -2.02318796e-01]
 [ 7.12021743e-01  4.31170937e+09  3.88293013e+01 -2.88129486e-01]
 [ 3.41529919e+00  4.51510476e+09  1.54763480e+02 -2.87897952e-01]
 [ 4.87096009e+00  4.95877452e+09  5.90965828e+01 -2.06980242e-01]
 [ 2.89543421e+00  6.01082383e+09  4.58037811e+01 -1.56978616e-01]
 [ 2.78361317e+00  6.28060982e+09  4.56665940e+01 -1.29440040e-01]
 [ 2.86420760e+00  6.38110664e+09  3.44770124e+01 -1.39542999e+00]
 [ 2.48342020e+00  6.82889316e+09  2.25398863e+01 -1.83065499e+00]
 [ 3.18995908e+00  9.27492020e+09  1.52734890e+01 -7.21262297e-01]
 [ 8.59368513e+00  1.10418160e+10  1.18743636e+01 -5.69786273e-01]
 [ 6.60067176e+00  1.59889664e+10  9.01491283e+00 -6.17530957e-01]
 [ 9.02006052e+00  1.94301639e+10  6.55851412e+00 -5.92402560e-01]
 [ 5.78679016e+00  2.68701733e+10  4.83732928e+00 -1.47722298e+00]
 [ 6.37738086e+00  3.40359401e+10  7.85080256e+00 -1.66638264e+00]
 [ 2.66377051e+00  2.80384019e+10  5.58742004e+00 -8.33088755e-01]
 [ 1.89010749e+00  2.55199345e+10  6.09141669e+00 -5.93959184e-01]
 [ 1.23049331e+00  2.59357385e+10  5.78925329e+00 -4.91866213e-01]
 [ 1.70128984e+00  2.46786229e+10  3.33492268e+00 -4.45177936e-01]
 [ 2.03110515e+00  2.70617927e+10  3.98471236e+00 -3.71323063e-01]
 [ 1.93737111e+00  2.88090052e+10  1.06830988e+00 -3.74575498e-01]
 [ 2.42727525e+00  2.47530561e+10 -5.94156437e-01 -4.70052228e-01]
 [ 3.37423267e+00  2.82190037e+10 -1.54479667e+00 -5.73660845e-01]
 [ 2.83273374e+00  3.30964009e+10  1.33902121e+00 -5.78007015e-01]
 [ 3.01811604e+00  4.05378808e+10  4.62548443e+00 -5.87493307e-01]
 [ 2.93423090e+00  4.40300619e+10  3.82785433e+00 -5.26814822e-01]
 [ 1.43315648e+00  4.69217425e+10  2.63107311e+00 -5.50759893e-01]
 [ 4.10699674e+00  5.11053339e+10  5.05232876e+00 -7.46036479e-01]
 [ 3.95173511e+00  5.04005460e+10  1.37954887e+01 -3.93250874e-01]]
```

```
print(y)
```

```
[8.26  8.373 8.17  8.01  6.74  5.52  5.63  6.24  6.97  6.56  8.11  6.95
 7.72  7.17  7.27  6.41  5.79  6.86  6.96  7.18  6.79  7.1   6.8   6.81
 5.9   4.93  4.19  3.91  5.03  5.59  5.436]
```

- **Split the dataset in training set and test set**

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.3,random_state=0)
```

- **Train the model on the training set**

```
from sklearn.linear_model import LinearRegression
ml=LinearRegression()
ml.fit(x_train,y_train)
```

LinearRegression

- **Predict the test set results**

Here, these are all predicted results, which are 10.

```
y_pred=ml.predict(x_test)
print(y_pred)
[7.28107814 5.27836863 6.55628869 7.26529439 5.55475737 5.99108793
 6.15337027 6.95343103 6.24221816 6.31009248]
```

In order to see more clearly, we are using the values of our x variable from the first data set and value it to see how close is it from our actual outcome which is y:

0	0.306508	3587916667	211.205588	-0.903463

```
ml.predict([[ 0.306508 ,3587916667  ,211.205588, -0.903463]])
array([7.55093803])
```

- **Evaluate the model**

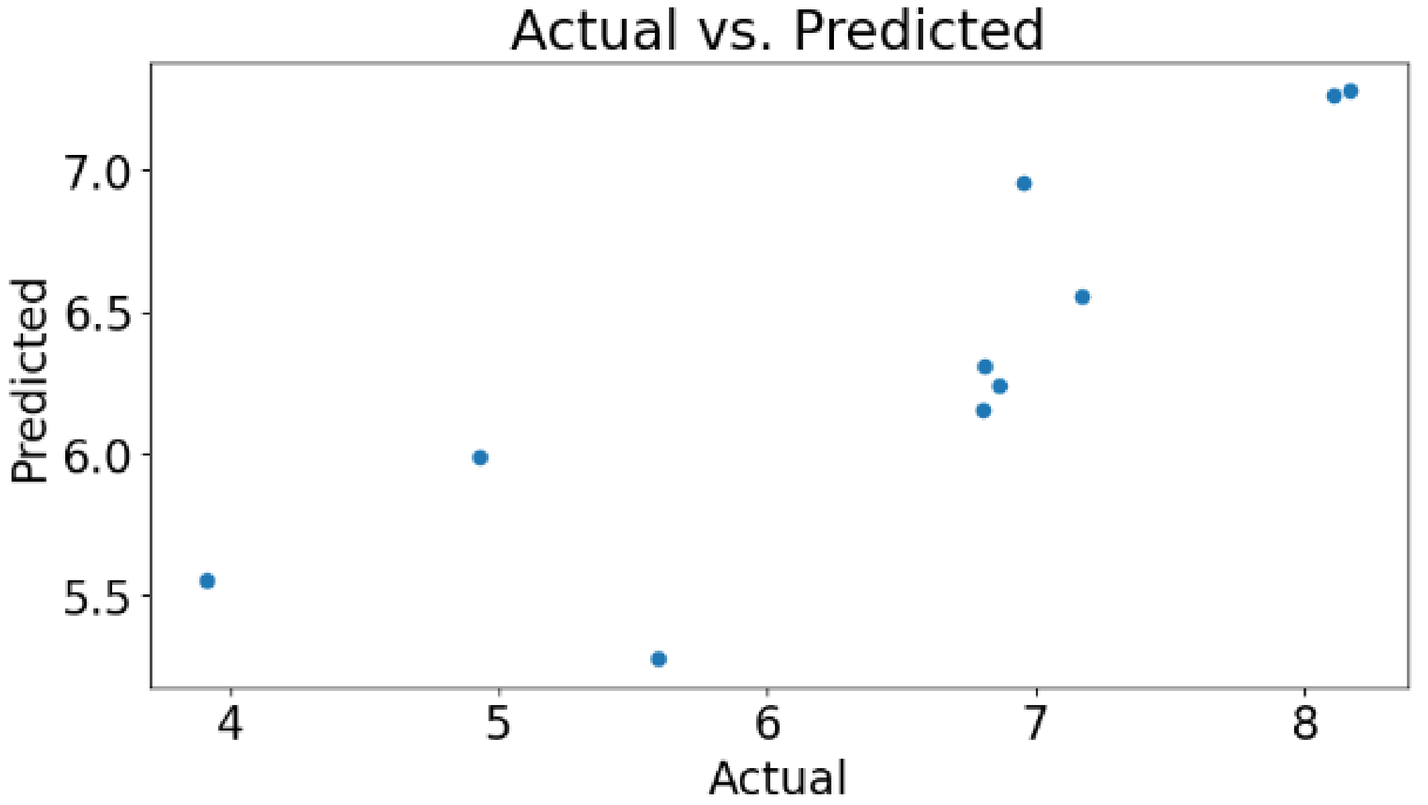
```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

```
0.580280325753812
```

- **Plot the results**

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,5))
plt.scatter(y_test,y_pred)
plt.xlabel('Actual')
plt.ylabel('Predicted')
plt.title('Actual vs. Predicted')
```

```
Text(0.5, 1.0, 'Actual vs. Predicted')
```



- **Predicted values**

```
pred_y_df=pd.DataFrame({'Actual                    Value':y_test,'Predicted
value':y_pred,'Difference': y_test-y_pred})
pred_y_df[0:30]
```

Actual Value	Predicted value	Difference	
0	8.17	7.281078	0.888922
1	5.59	5.278369	0.311631
2	7.17	6.556289	0.613711
3	8.11	7.265294	0.844706
4	3.91	5.554757	-1.644757
5	4.93	5.991088	-1.061088
6	6.80	6.153370	0.646630
7	6.95	6.953431	-0.003431
8	6.86	6.242218	0.617782
9	6.81	6.310092	0.499908

CONCLUSION

THANK YOU!