STAT40800 Data Programming with Python (online)

Data Analysis Project

Autumn 2021/22

1 ABSTRACT

This data analysis project aims at studying y the 2016 Irish Census data. The census is a detailed account of every person living in Ireland. A census is carried out every 5 years by the Central Statistics Office (CSO).The aim of this project is to analyze the themes in this data and try to identify trends or patterns within the data and do some inference

For this Project, Three themes were considered. They were **Theme 1: Sex, Age and Marital Status ,Theme 2: Principal Status ,Theme 3: Education** at the level of county. An extensive literature review was done to identify and emphasize on variables that could act as a good predictor for the Unemployment factor present in each county.The most important variables to include in a simple linear regression model were identified and were then used to build the regression model. Based on the predictions done by the model, The best feature was chosen after extensive validation.

2 INTRODUCTION

The Central Statistics Office (CSO) is in charge of collecting, collating, analyzing, and sharing statistical data on Ireland's population and economy. The Republic of Ireland's Census 2016 was held on Sunday, April 24, 2016, to administer a national census. The Central Statistics Office (CSO) conducted the census, which revealed a total population of 4,761,865, up 3.8 percent from the previous census in 2011.This was the lowest recorded population growth rate since the 1991 census, with lower birth rates and reduced net migration contributing to the drop in population growth rates.Between April and December 2017, the census results were gradually issued in a series of reports organized as summaries or in-depth results of certain themes such as age, race, or religion.

The census of population is a detailed count of every person living in Ireland on a particular date.The population of the state is determined by the census. It also assists the government in planning how it will provide public services to the state's citizens.Age, marital status, sex, place of birth, occupation, and religion are among the inquiries on the form. The data is then utilized to plan infrastructure for public transportation, healthcare, housing, and education. Census data is used by local governments to forecast demand for their services and facilities.

The census data is broken down into 15 themes namely Theme 1: Sex, age and marital status,Theme 2: Migration, ethnicity, religion and foreign languages,Theme 3: Irish language,Theme 4: Families,Theme 5: Private households,Theme 6: Housing,Theme 7: Communal establishments,Theme 8: Principal status,Theme 9: Social class and socio-economic group,Theme 10: Education,Theme 11: Commuting,Theme 12: Disability, carers and general health,Theme 13: Occupations,Theme 14: Industries,Theme 15: Motor car availability, PC ownership and internet access. These themes are further classified into many tables which explain the characteristics of the county based on the data. The themes that are chosen for this project are **Theme 1: Sex, Age and Marital Status ,Theme 2: Principal Status ,Theme 3: Education** .These themes consisted tables with information regarding age and gender-wise distribution, principal status and education status of each individual in county.

For theme 1, there were 105 columns describing how total number of people in a county is distributed across all age groups and gender. This was very intuitive as it helped to understand the composition of male and female in a county. This provided us information like what proportion of people belonged to a certain age group. Using this, in the project , there was some analysis done the male female composition, categorisation of people based on age like children, youth,teenager,young adults,middle age, old age. Using this, the proportion of people who are at the right age to be employed was determined which was opted as a feature indicator to predict the unemployment in each county.

For theme 2, There were 27 columns which described the status of the people like whether are unemployed, or whether they are looking for their first job, where they are students, whether they are disabled, whether they are looking after their family and so on. These were again categorised into male and female to make it the gender distribution among statuses clear. But for this project, the main concern was to predict the unemployment count per county. For this, only two data was considered from the theme, they were '*Looking for first job*' and '*Unemployed having lost or given up previous job*' . The rest deemed unnecessary as they clearly didn't portray the meaning of unemployment.

For theme 3, there were three tables describing the Education status of the individuals. They described whether the individual has completed his/her education, whether it is in progress and the field of study of the individual. This was described using 110 feature vectors. But for this project, Total no of individuals who have completed studying or still studying was demmed sufficient to act as predictor variables for unemployment count per county.

This project will begin by doing an Explorative Data Analysis on the given dataset and try to find and filter the neccessary feature and group them into meaningful tables. Then there will analysis on the distribution of these selected variable.**This report will aim to find the best possible predictor variable which can be used to predict the unemployment count in the county.** The main aim is to try build a linear relationship between the total unemployement and any of the feature like education or age . This is verified using various methodologies in this report. The finding and conclusions are presented below in the report.

3 RESULTS AND DISCUSSION

3.1 Install Packages

Import the necessary packages needed for this project. I have imported 'warning' and set filterwarnings to ignore. This is to ignore the warning raised by seaborn package for distplot. This is done just to make the report more attractive and in no way affects the execution of the code.

In [838…

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import pearsonr
import statsmodels.api as sm
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import scipy.stats as stats
from sklearn.cluster import KMeans
import warnings
warnings.filterwarnings('ignore')
```

3.2 Import Census Data

I imported the *Census_by_county.csv* and created a dataframe by name census_data which contains all features. This will act as the base dataframe for our further investigation. I have defined a lambda function which does a simple preprocessing step of replacing the comma in the dataframe. I have done a small check for missing values for confirmation before proceeding with our further analysis.

NB: Census_by_county.csv should be in the same directory

In [839…

```
#import the census_by_county dataset provided for the year 2016
census_data= pd.read_csv('Census_by_county.csv')

#defining a preprocessing step using lambda function to remove comma and use this to
prep = lambda x: str(x.replace(',',''))

#check for missing values in the census_data
census_data.isnull().sum()
```

Out[839…

```
GUID          0
GEOGID        0
GEOGDESC      0
T1_1AGE0M     0
T1_1AGE1M     0
             ..
T15_3_B       0
T15_3_OTH     0
T15_3_N       0
T15_3_NS      0
T15_3_T       0
Length: 802, dtype: int64
```

3.3 Gender Distribution In County

Using the census_data, i created a dataframe for gender wise distribution in county . This data frame also included the total population and male female ratio in county which was computed from the given data.

In [840…

```
#Now create a table gender_county which will have male female composition inside eac
```

```
gender_county = census_data[['GEOGDESC','T1_1AGETM','T1_1AGETF','T1_1AGETT']]

#using prep function for preprocessing
gender_county=gender_county.applymap(prep)

#converting string to float for population
gender_county[['T1_1AGETM','T1_1AGETF','T1_1AGETT']]=gender_county[['T1_1AGETM','T1_

#rename the columns in gender_county
gender_county = gender_county.rename(columns = {"GEOGDESC":"County_Name","T1_1AGETM"

#Calculating the Male female Ratio
gender_county['Male_Female_Ratio']= (gender_county['Male_Population'] / gender_count
gender_county
```

Out[840…

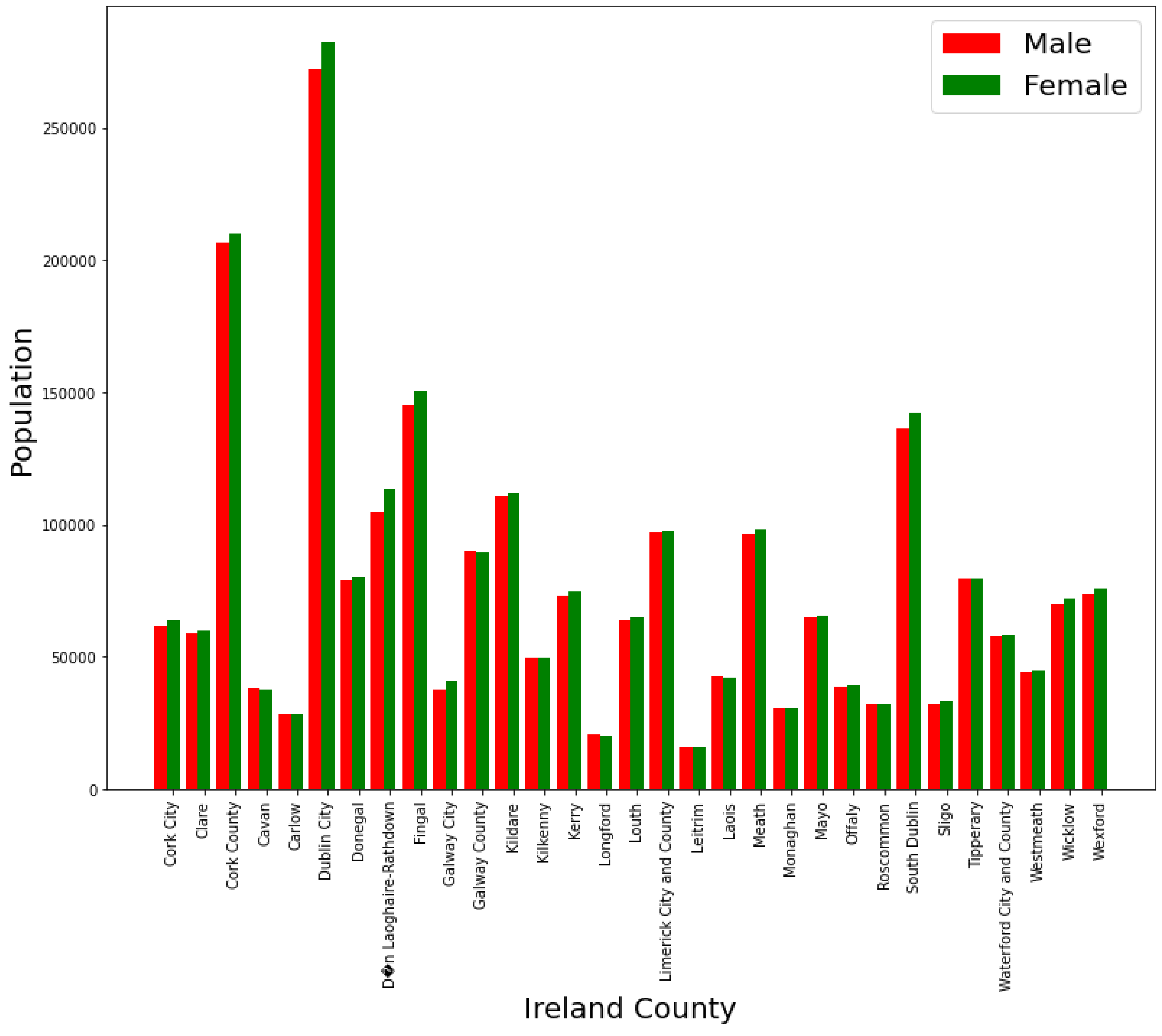
	County_Name	Male_Population	Female_Population	Total_Population	Male_Female_Ratio
0	Cork City	61722	63935	125657	0.965387
1	Clare	58785	60032	118817	0.979228
2	Cork County	206953	210258	417211	0.984281
3	Cavan	38330	37846	76176	1.012789
4	Carlow	28465	28467	56932	0.999930
5	Dublin City	272270	282284	554554	0.964525
6	Donegal	79022	80170	159192	0.985680
7	D�n Laoghaire-Rathdown	104584	113434	218018	0.921981
8	Fingal	145240	150780	296020	0.963258
9	Galway City	37800	40868	78668	0.924929
10	Galway County	89863	89527	179390	1.003753
11	Kildare	110546	111958	222504	0.987388
12	Kilkenny	49533	49699	99232	0.996660
13	Kerry	73055	74652	147707	0.978607
14	Longford	20587	20286	40873	1.014838
15	Louth	63633	65251	128884	0.975203
16	Limerick City and County	97340	97559	194899	0.997755
17	Leitrim	16064	15980	32044	1.005257
18	Laois	42811	41886	84697	1.022084
19	Meath	96776	98268	195044	0.984817
20	Monaghan	30866	30520	61386	1.011337
21	Mayo	65047	65460	130507	0.993691
22	Offaly	38838	39123	77961	0.992715
23	Roscommon	32377	32167	64544	1.006528
24	South Dublin	136277	142490	278767	0.956397
25	Sligo	32365	33170	65535	0.975731

	County_Name	Male_Population	Female_Population	Total_Population	Male_Female_Ratio
26	Tipperary	79668	79885	159553	0.997284
27	Waterford City and County	57651	58525	116176	0.985066
28	Westmeath	44082	44688	88770	0.986439
29	Wicklow	70156	72269	142425	0.970762
30	Wexford	73722	76000	149722	0.970026

I developed a plot to get a overview of male vs female population in each county. We can see that the female population is consistenly greater than or equal to male population in the every county. Though, the difference is in few thousands, it is clearly visible in the given plot. Female population dominates male in each of the county.

In [841…

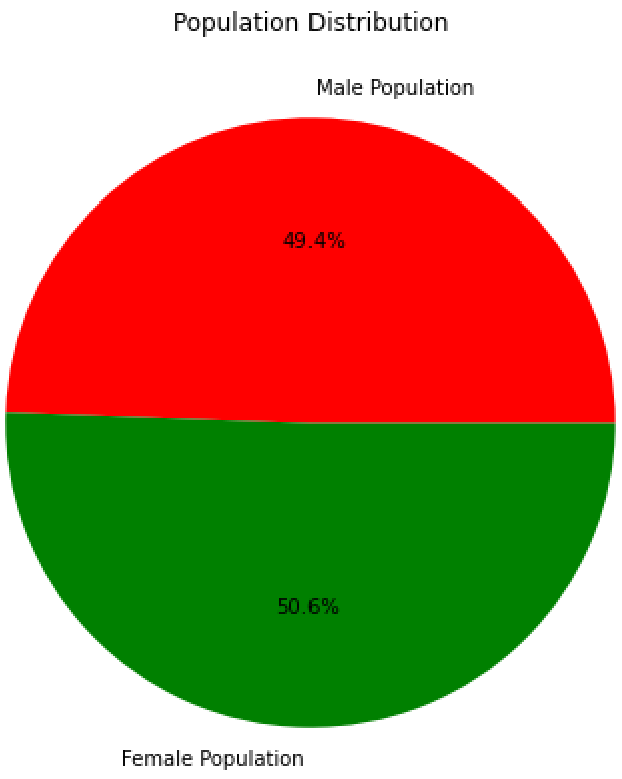
```
############# Plotting the Gender Wise distribution per county #####################
x=np.arange(31)
plt.figure(figsize=(13,10))
plt.bar(x-0.4,gender_county.Male_Population,width=0.4,color='r')
plt.bar(x,gender_county.Female_Population,width=0.4,color='g')
plt.xticks(x,gender_county['County_Name'].values,rotation=90)
plt.xlabel('Ireland County',fontsize=20)
plt.ylabel('Population',fontsize=20)
plt.legend(['Male','Female'],fontsize=20)
plt.show()
```



Using the above gender distribution table, I have calculated the total male and female population of the county and also calculated the average male and female population per county. This is done just to check the relationship among these two features. I have showed the relationship among these in terms of total population by plotting a pie chart.

In [842…

```
#Calculating Total Male Population and Avg male Population per county
Total_Male_Ireland =  gender_county['Male_Population'].sum()
Avg_Male_County = int(gender_county['Male_Population'].mean())

#Calculating Total Female Population and Avg Female Population per county
Total_Female_Ireland =  gender_county['Female_Population'].sum()
Avg_Female_County = int(gender_county['Female_Population'].mean())

#Calculating Total Population of Ireland
Total_Pop_Ireland = Total_Male_Ireland + Total_Female_Ireland


#Plotting the above data

population = [Total_Male_Ireland,Total_Female_Ireland]
my_Labels = ['Male Population','Female Population']
my_colors = ['red','green']

plt.figure(figsize=(7,7))
plt.pie(population,labels=my_Labels,autopct='%1.1f%%',colors=my_colors)
plt.title('Population Distribution')
plt.axis('off')
plt.show()
```



3.4 Age Wise Distribution In County

I have created two base dataframes for Male and Female which consists of age wise distribution of the entire population in the county. I have preprocessed these data frames using the lambda function which was defined in the earlier part of this report. These data frame consists of all the age groups starting from 0 and they are arranged according to county.

In [843…

```
#Getting the agewise data from census_data and storing in Male and Female table acco
Age_Male_Distribution = census_data[['GEOGDESC','T1_1AGE0M','T1_1AGE1M','T1_1AGE2M',
]]

Age_Male_Distribution=Age_Male_Distribution.applymap(prep)

Age_Female_Distribution = census_data[['GEOGDESC','T1_1AGE0F','T1_1AGE1F','T1_1AGE2F
]]

Age_Female_Distribution=Age_Female_Distribution.applymap(prep)
```

Using the above two dataframes, I have created a dataframe *Age_Gender_Distribution* which has the population categories into different age groups. Children is considered as aged between 0 and 16. Teenage is considered as aged between 16 and 19. Youth is considered as aged between

19 and 25. YoungAdult is considered as aged between 25-40. Middle_age is considered as aged between 40 and 70. Old_age is considered as aged between 70 and above. This distribution is divided across male and female.

In [844…

```
#Creating the bins for male and female
Age_Gender_Distribution = pd.DataFrame(Age_Male_Distribution['GEOGDESC'])
Age_Gender_Distribution = Age_Gender_Distribution.rename(columns={'GEOGDESC':'County


#creating children aged between 0-16 for male and females
Age_Gender_Distribution['Children_Male'] = Age_Male_Distribution[['T1_1AGE0M','T1_1A
Age_Gender_Distribution['Children_Female'] = Age_Female_Distribution[['T1_1AGE0F','T

#creating Teenage aged between 16-19 for male and females
Age_Gender_Distribution['Teenage_Male'] = Age_Male_Distribution[['T1_1AGE16M','T1_1A
Age_Gender_Distribution['Teenage_Female'] = Age_Female_Distribution[['T1_1AGE16F','T

#creating Youth aged between 19-25 for male and females
Age_Gender_Distribution['Youth_Male'] = Age_Male_Distribution[['T1_1AGE20_24M']].ast
Age_Gender_Distribution['Youth_Female'] = Age_Female_Distribution[['T1_1AGE20_24F']]

#creating young_adults aged between 25-40 for male and females
Age_Gender_Distribution['YoungAdults_Male'] = Age_Male_Distribution[['T1_1AGE25_29M'
Age_Gender_Distribution['YoungAdults_Female'] = Age_Female_Distribution[['T1_1AGE25_

#creating middle_age  between 40-70 for male and females
Age_Gender_Distribution['middle_age_Male'] = Age_Male_Distribution[['T1_1AGE40_44M',
Age_Gender_Distribution['middle_age_Female'] = Age_Female_Distribution[['T1_1AGE40_4

#creating old_age  between 70&above for male and females
Age_Gender_Distribution['old_age_Male'] = Age_Male_Distribution[['T1_1AGE70_74M','T1
Age_Gender_Distribution['old_age_Female'] = Age_Female_Distribution[['T1_1AGE70_74F'

Age_Gender_Distribution
```

Out[844…

	County_Name	Children_Male	Children_Female	Teenage_Male	Teenage_Female	Youth_Male	You
0	Cork City	9719	9410	3288	3449	6152	
1	Clare	13754	13474	3153	3101	2985	
2	Cork County	51789	49499	10916	10307	10072	
3	Cavan	9552	9181	2084	2020	1837	
4	Carlow	6795	6573	1554	1375	1747	
5	Dublin City	44954	43197	12033	11810	21673	
6	Donegal	19257	18100	4420	4122	3762	
7	D�n Laoghaire-Rathdown	21736	20827	5857	5809	7759	
8	Fingal	39075	37373	7133	6782	8012	
9	Galway City	7076	6924	2033	2209	4046	
10	Galway County	22135	21004	4732	4369	4227	

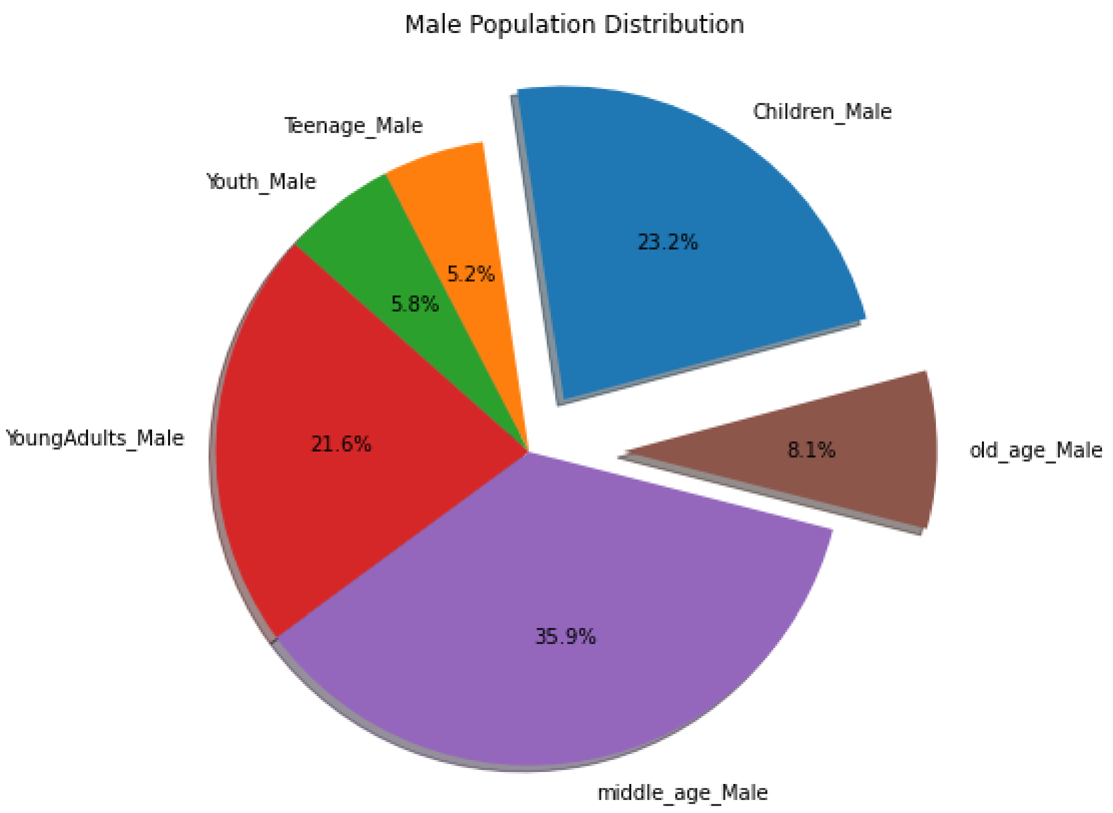
	County_Name	Children_Male	Children_Female	Teenage_Male	Teenage_Female	Youth_Male	You
11	Kildare	29252	27567	6276	5944	6249	
12	Kilkenny	12014	11200	2618	2482	2373	
13	Kerry	15530	15109	3696	3562	3345	
14	Longford	5213	4891	1049	958	978	
15	Louth	16039	15477	3381	3316	3628	
16	Limerick City and County	21522	20495	5566	5235	6367	
17	Leitrim	3737	3575	769	723	662	
18	Laois	11287	10743	2085	2025	2155	
19	Meath	26361	25394	5267	4903	4803	
20	Monaghan	7640	7163	1700	1472	1551	
21	Mayo	14464	13898	3357	3180	2991	
22	Offaly	9663	9181	2226	2067	1983	
23	Roscommon	7416	7148	1586	1520	1436	
24	South Dublin	34669	33150	7393	6903	8210	
25	Sligo	7316	6864	1846	1707	1796	
26	Tipperary	18597	17734	4255	4003	3992	
27	Waterford City and County	13358	12771	3182	2919	3097	
28	Westmeath	10680	10405	2390	2278	2578	
29	Wicklow	17440	16835	3649	3391	3629	
30	Wexford	18022	16971	3931	3807	3489	

Using the data frame *Age_Gender_Distribution* ,I calculated the total sum of population in each category specifically for male and female. Then i showed the distribution using a pie chart. I have added the explode feature in the pie chart which separates the category of people whom i assumed to be not part of the working class category.. This has been done seperately for both gender using seperate pie charts. Both are having similar distribution in which majority of it is under the middle age category and nearly 24 % are children i.e aged between 0-16. There is quite a significant amount of population who are young adults aged between 25 and 40.

In [845…

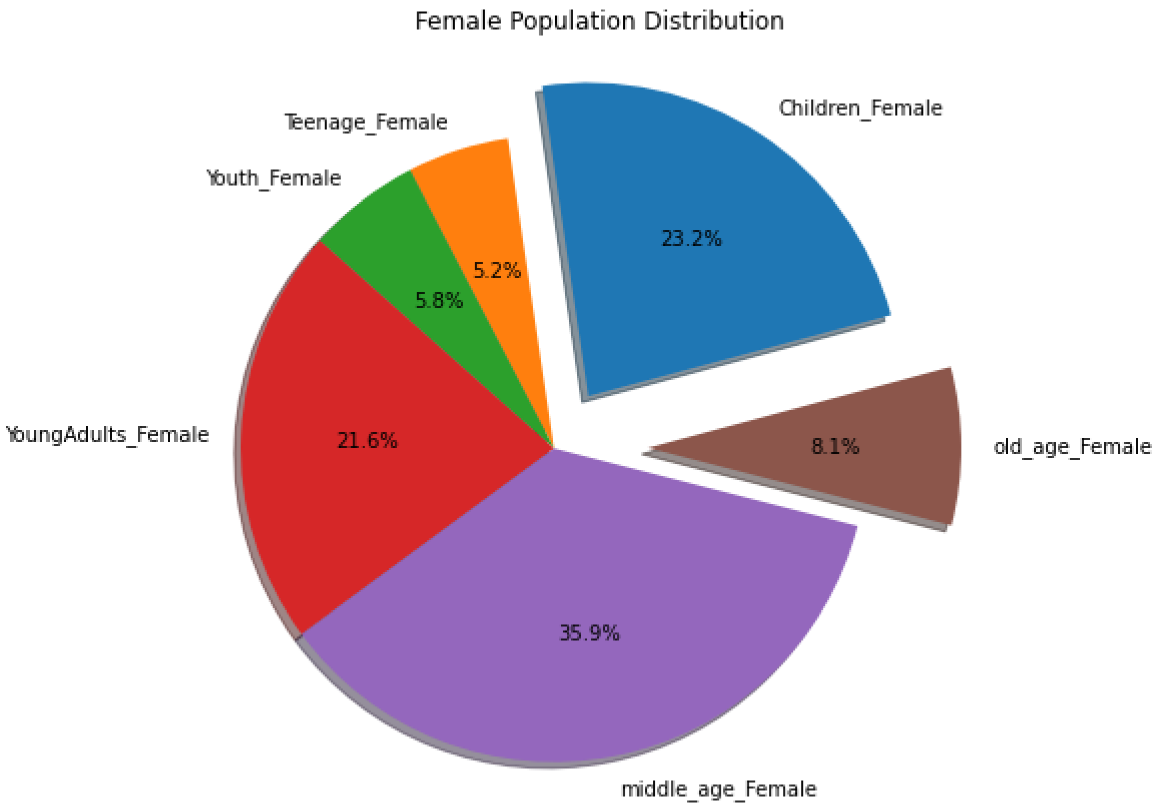
```
# Getting the total no of males for each age category from Age_Gender_Distribution T
children_male_total = Age_Gender_Distribution['Children_Male'].sum(axis=0)
teenage_male_total = Age_Gender_Distribution['Teenage_Male'].sum(axis=0)
youth_male_total = Age_Gender_Distribution['Youth_Male'].sum(axis=0)
young_adults_male_total = Age_Gender_Distribution['YoungAdults_Male'].sum(axis=0)
```

```
middle_age_male_total = Age_Gender_Distribution['middle_age_Male'].sum(axis=0)
old_age_male_total = Age_Gender_Distribution['old_age_Male'].sum(axis=0)

#Creating a list of the above data to provide it for pie chard and specifying the La
male_population_distribution = [children_male_total,teenage_male_total,youth_male_to
my_Labels = ['Children_Male','Teenage_Male','Youth_Male','YoungAdults_Male','middle_
my_explode = (0.2,0,0,0,0,0.3)

#Plotting the Male Population Distribution
plt.figure(figsize=(7,7))
plt.pie(male_population_distribution,labels=my_Labels,autopct='%1.1f%%',startangle=1
plt.title('Male Population Distribution')
plt.axis('equal')
plt.show()
```



In [846…

```
# Getting the total no of females for each age category from Age_Gender_Distribution
children_female_total = Age_Gender_Distribution['Children_Female'].sum(axis=0)
teenage_female_total = Age_Gender_Distribution['Teenage_Female'].sum(axis=0)
youth_female_total = Age_Gender_Distribution['Youth_Female'].sum(axis=0)
young_adults_female_total = Age_Gender_Distribution['YoungAdults_Female'].sum(axis=0
middle_age_female_total = Age_Gender_Distribution['middle_age_Female'].sum(axis=0)
old_age_female_total = Age_Gender_Distribution['old_age_Female'].sum(axis=0)

#Creating a list of the above data to provide it for pie chard and specifying the La
female_population_distribution = [children_female_total,teenage_female_total,youth_f
my_Labels = ['Children_Female','Teenage_Female','Youth_Female','YoungAdults_Female',
my_explode = (0.2,0,0,0,0,0.3)

#Plotting the Male Population Distribution
plt.figure(figsize=(7,7))
plt.pie(male_population_distribution,labels=my_Labels,autopct='%1.1f%%',startangle=1
plt.title('Female Population Distribution')
plt.axis('equal')
plt.show()
```

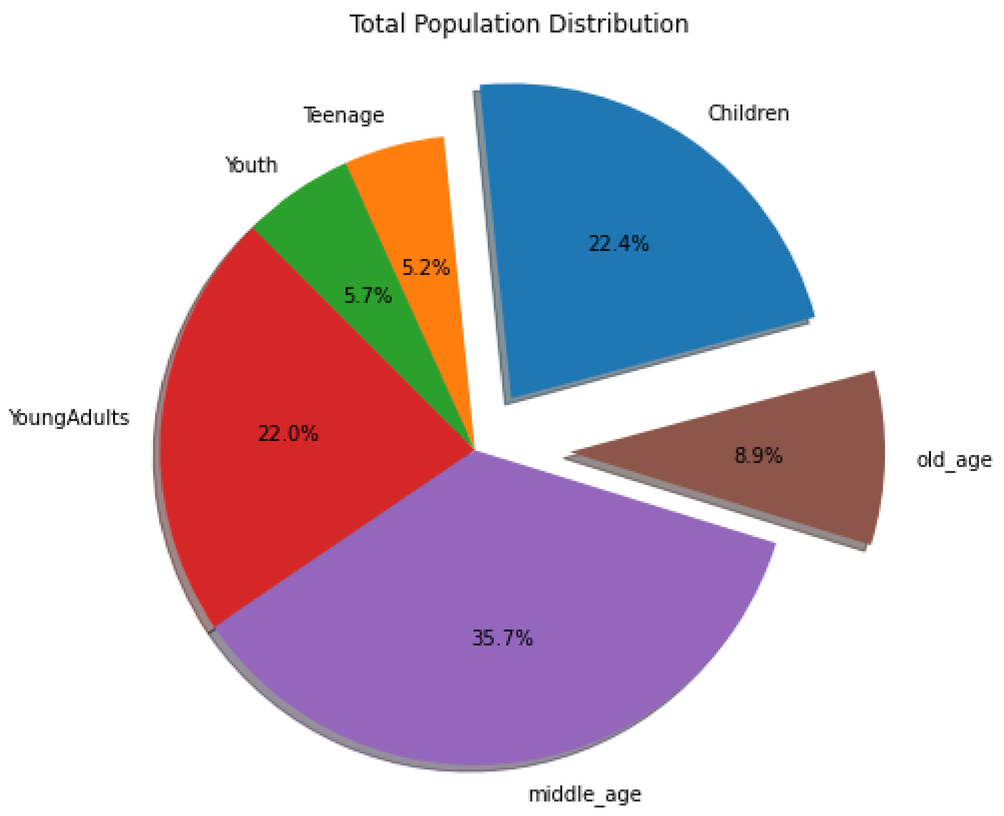


Plotting using Pie Chart

Using the above two data, I developed a pie chart discarding the gender distribution. The pie chart was very similar to above ones with very minute differences in the bins.

In [847…

```
#Calculating the proportion for all types of age group for total population in irela

tot_Children = (Age_Gender_Distribution['Children_Male'].sum() + Age_Gender_Distribu
tot_Teenage = (Age_Gender_Distribution['Teenage_Male'].sum() + Age_Gender_Distributi
tot_Youth = (Age_Gender_Distribution['Youth_Male'].sum() + Age_Gender_Distribution['
tot_YoundAdults = (Age_Gender_Distribution['YoungAdults_Male'].sum() + Age_Gender_Di
tot_MiddleAge = (Age_Gender_Distribution['middle_age_Male'].sum() + Age_Gender_Distr
tot_OldAge = (Age_Gender_Distribution['old_age_Male'].sum() + Age_Gender_Distributio


#Creating a list of the above data to provide it for pie chard and specifying the la
total_population_distribution = [tot_Children,tot_Teenage,tot_Youth,tot_YoundAdults,
my_Labels = ['Children','Teenage','Youth','YoungAdults','middle_age','old_age']
my_explode = (0.2,0,0,0,0,0.3)

#Plotting the Male Population Distribution
plt.figure(figsize=(7,7))
plt.pie(total_population_distribution,labels=my_Labels,autopct='%1.1f%%',startangle=
plt.title('Total Population Distribution')
plt.axis('equal')
plt.show()
```



3.4 Unemployment Distribution In County

I have created a dataframe *unemployment_county* which stores the details of the total unemployment categorised by gender in a county. For this, only two data was considered from the *census data*, they were 'Looking for first job' and 'Unemployed having lost or given up previous job'. According to CSO website, unemployment is considered for these two variable and hence i have used the same. I have stored the male female unemployment ratio as well to get a better idea about the distribution.

In [848…

```
############### get unemployment per county ######################################
#Temporary dataframe to store the required columns from census data
temp_df=census_data[['GEOGDESC','T8_1_LFFJM','T8_1_ULGUPJM','T8_1_LFFJF','T8_1_ULGUP
temp_df[['T8_1_LFFJM','T8_1_ULGUPJM','T8_1_LFFJF','T8_1_ULGUPJF','T8_1_LFFJT','T8_1_

## Unemployment data per county for male and Female
unemployment_county=pd.DataFrame(temp_df['GEOGDESC'])
unemployment_county=unemployment_county.rename(columns={'GEOGDESC':'County_Name'})
unemployment_county['Male_Unemployed'] = temp_df['T8_1_LFFJM'] + temp_df['T8_1_ULGUP
unemployment_county['Female_Unemployed'] = temp_df['T8_1_LFFJF'] + temp_df['T8_1_ULG
unemployment_county['Total_Unemployed'] = temp_df['T8_1_LFFJT'] + temp_df['T8_1_ULGU
unemployment_county['maleFemale_UnemployedRatio'] = unemployment_county['Male_Unempl

unemployment_county
```

Out[848…

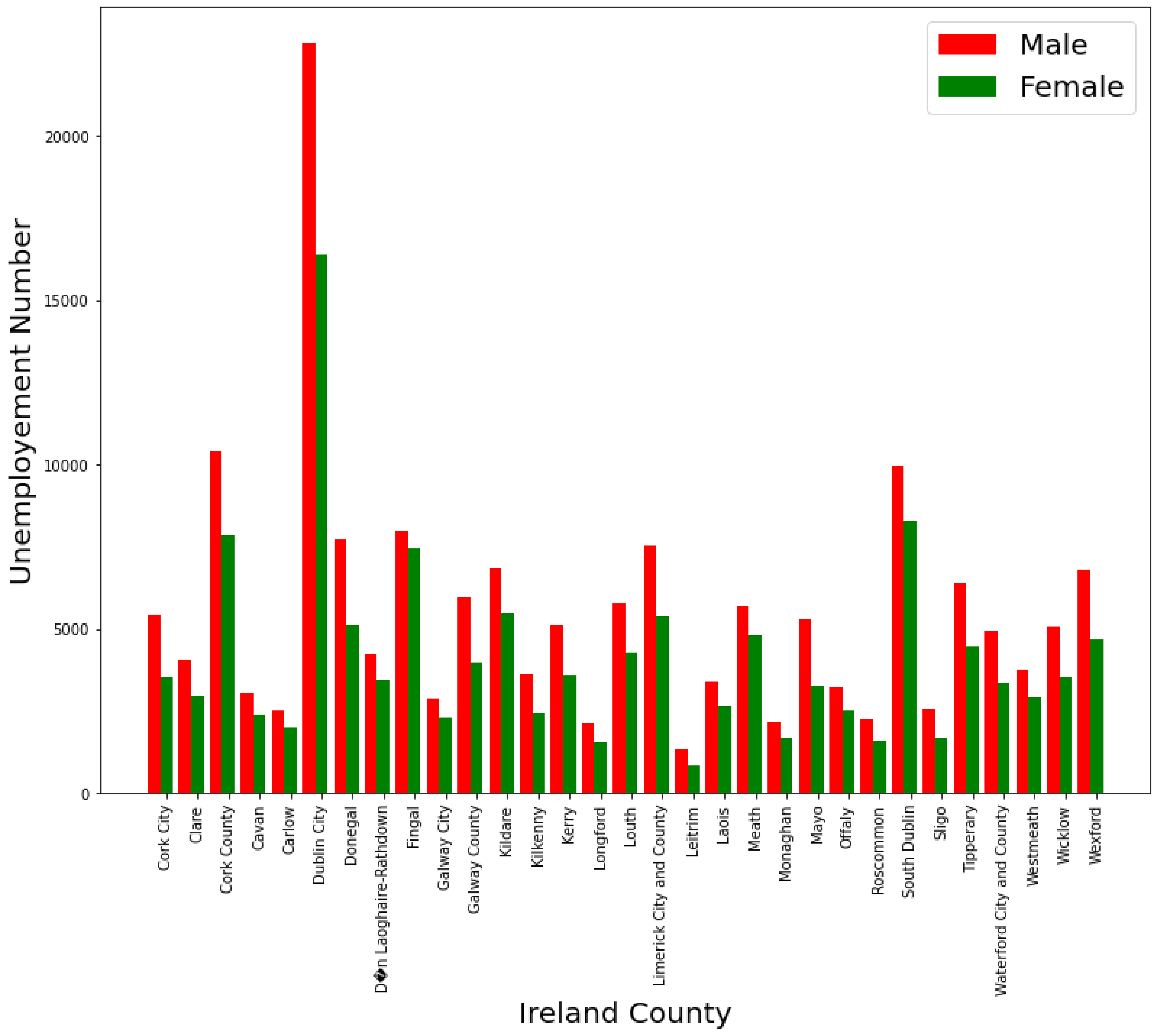
	County_Name	Male_Unemployed	Female_Unemployed	Total_Unemployed	maleFemale_Unemplo
0	Cork City	5428	3515	8943	
1	Clare	4077	2941	7018	

	County_Name	Male_Unemployed	Female_Unemployed	Total_Unemployed	maleFemale_Unemploy
2	Cork County	10418	7869	18287	
3	Cavan	3042	2371	5413	
4	Carlow	2538	1969	4507	
5	Dublin City	22813	16387	39200	
6	Donegal	7699	5130	12829	
7	D�n Laoghaire-Rathdown	4256	3460	7716	
8	Fingal	7974	7441	15415	
9	Galway City	2854	2321	5175	
10	Galway County	5961	3977	9938	
11	Kildare	6828	5469	12297	
12	Kilkenny	3600	2444	6044	
13	Kerry	5109	3592	8701	
14	Longford	2129	1572	3701	
15	Louth	5774	4284	10058	
16	Limerick City and County	7556	5379	12935	
17	Leitrim	1337	826	2163	
18	Laois	3398	2670	6068	
19	Meath	5714	4809	10523	
20	Monaghan	2168	1663	3831	
21	Mayo	5309	3282	8591	
22	Offaly	3213	2531	5744	
23	Roscommon	2260	1587	3847	
24	South Dublin	9987	8278	18265	
25	Sligo	2556	1694	4250	
26	Tipperary	6389	4478	10867	
27	Waterford City and County	4949	3374	8323	
28	Westmeath	3756	2910	6666	
29	Wicklow	5060	3543	8603	
30	Wexford	6778	4700	11478	

The below plot is to to analyse if there is a significant difference between male and female unemployment so that we can decide if we can use total unemployment as the dependent variable for our analysis. By looking at the plot, we can see that there is difference between the unemployment among male and female but it is not of significant amount. Hence we can consider total unemployment per county as the dependent variable for our regression analysis.

In [849…

```
#plotting unemployment distribution in county
x=np.arange(31)
plt.figure(figsize=(13,10))
plt.bar(x-0.4,unemployment_county.Male_Unemployed,width=0.4,color='r')
plt.bar(x,unemployment_county.Female_Unemployed,width=0.4,color='g')
plt.xticks(x,unemployment_county['County_Name'].values,rotation=90)
plt.xlabel('Ireland County',fontsize=20)
plt.ylabel('Unemployement Number',fontsize=20)
plt.legend(['Male','Female'],fontsize=20)
plt.show()
```



Plotting using Bar Plot

The above graph is the pictorial representation of the relationship between the unemployment distribution among male and females. From the below table we can see that the mean value between male and female unemployed is almost similar and they vary over a standard deviation of 3937 and 2931 respectively. Looking at this data, if we combine the two to calculate the total unemployed, it will be easier for prediction

In [850…

```
####### Analysing the mean and Variance of male vs female   ########################
male_female_Unemployment = unemployment_county[['Male_Unemployed','Female_Unemployed
male_female_Unemployment.describe()
```

Out[850…

	Male_Unemployed	Female_Unemployed
count	31.000000	31.000000
mean	5513.870968	4079.548387
std	3937.427703	2931.320372
min	1337.000000	826.000000
25%	3127.500000	2407.500000
50%	5060.000000	3460.000000
75%	6583.500000	4754.500000
max	22813.000000	16387.000000

Plotting using Box Plot to interpret Unemployment Data

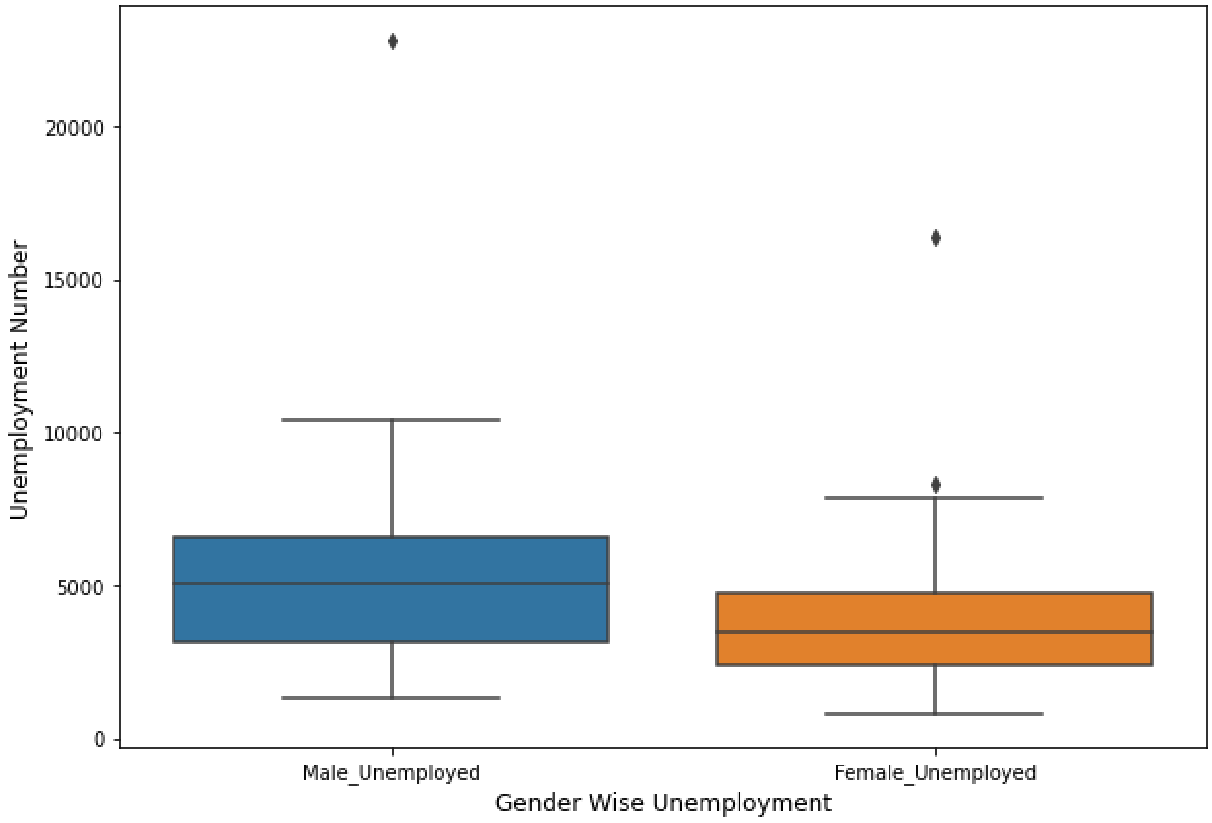
I have done a box plot for male and female unemployed in each county. Looking at the box plot we can see that there are some outliers for each case. Those are generated because of the unemployment count in *Dublin City* . But we cannot discard this value because it is a significant value and if discarded it will affect the total unemployment count in the country. So in this case, even if its an outlier, we keep it in the dataset.

In [851…

```
#box plot
plt.figure(figsize=(10,7))
sns.boxplot(data=male_female_Unemployment)
plt.ylabel('Unemployment Number',fontsize=12)
plt.xlabel('Gender Wise Unemployment',fontsize=12)
```

Out[851…

```
Text(0.5, 0, 'Gender Wise Unemployment')
```

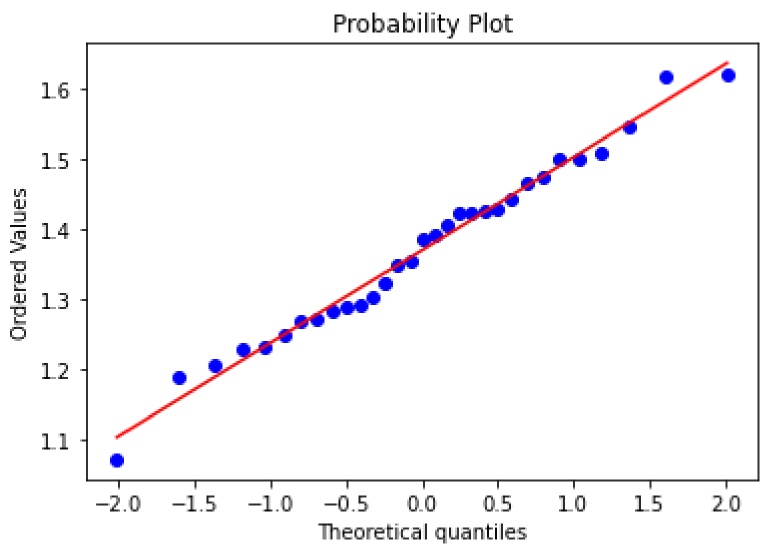


Q-Q Plot to identify Normal Distribution

This section is just an exploration of verifying Male female unemployment ratio being a normal distribution. I have made use of quantile-quantile (Q-Q) Plot to determine this. A Q-Q plot compares the observed distribution to the expected normal distribution.For normally distributed data, observations should lie approximately on a straight line. If the data is non-normal, the points form a curve that deviates markedly from a straight line. Possible outliers are points at the ends of the line, distanced from the bulk of the observations. By looking at the plot we can see that all observations lie on the straight line saying it fits the normal distribution.

In [852…

```
# QQ Plot male:female unemployement ratio
plt.figure()
stats.probplot(unemployment_county.maleFemale_UnemployedRatio, dist='norm',plot=plt)
```



Useful Analytics

There are some specific information about the specific of the unemployment in Ireland. We can see some insights like Total no of unemployed males in Ireland which is 170930 but for females it is 126466. The count does not provide any significant information as we cannot compare to anything. The Percentage of unemployed males in Ireland is 7.25% which is nearly 2% higher than female unemployment percent which 5.25%.The total unemployment percentage of people in ireland is 6.24% .But males provide the main contribution for this percentage.

In [853…

```
##Calcuating total unemployemt of male and females in Irelnad
Total_Unemployed_Males_Ireland = unemployment_county['Male_Unemployed'].sum()
Total_Unemployed_Females_Ireland = unemployment_county['Female_Unemployed'].sum()
Total_Unemployed_Ireland = unemployment_county['Total_Unemployed'].sum()

#Printing the data
print("Total no of unemployed males in Ireland is",Total_Unemployed_Males_Ireland)
print("Total no of unemployed females in Ireland is",Total_Unemployed_Females_Irelan
print("Total no of unemployed people in Ireland is",Total_Unemployed_Ireland)

#Calculating percentage of unemployment
Percent_Unemployed_Male = Total_Unemployed_Males_Ireland / Total_Male_Ireland *100
Percent_Unemployed_Female = Total_Unemployed_Females_Ireland / Total_Female_Ireland
Percent_Unemployed = Total_Unemployed_Ireland/Total_Pop_Ireland *100

print("Percentage of unemployed males in Ireland is",Percent_Unemployed_Male)
print("Percentage  of unemployed females in Ireland is",Percent_Unemployed_Female)
print("Percentage  of unemployed people in Ireland is",Percent_Unemployed)
```

```
Total no of unemployed males in Ireland is 170930
Total no of unemployed females in Ireland is 126466
Total no of unemployed people in Ireland is 297396
Percentage of unemployed males in Ireland is 7.259937445528171
Percentage  of unemployed females in Ireland is 5.2531385037282385
Percentage  of unemployed people in Ireland is 6.245368148824042
```

3.5 Education Status among the population In County

I have created a DataFrame *county_Education* which stored the education status of the population in the county. It consists of Two main information. One is total number of people who have completed the education and the other is the total number of people whose education is still in progress.This information is categorized among male and female. The total of these data is also stored as a seperate column per county in this dataframe.

In [854…

```
##################### table per county for Education ######################
county_Education = pd.DataFrame(gender_county['County_Name'])

county_Education['Education_Completed_Males'] = census_data['T10_4_TM']
county_Education['Education_Completed_Females'] = census_data['T10_4_TF']
county_Education['Education_Completed_Both'] = census_data['T10_4_TT']
county_Education['Education_InProgress_Males'] = census_data['T10_2_SASM']
county_Education['Education_InProgress_Females'] = census_data['T10_2_SASF']
county_Education['Education_InProgress_Total'] = census_data['T10_2_SAST']


county_Education=county_Education.applymap(prep)

county_Education['Education_Completed_Males'] = county_Education['Education_Complete
county_Education['Education_Completed_Females'] = county_Education['Education_Comple
county_Education['Education_Completed_Both'] = county_Education['Education_Completed
county_Education['Education_InProgress_Males'] = county_Education['Education_InProgr
county_Education['Education_InProgress_Females'] = county_Education['Education_InPro
county_Education['Education_InProgress_Total'] = county_Education['Education_InProgr


county_Education
```

Out[854…

	County_Name	Education_Completed_Males	Education_Completed_Females	Education_Completed_
0	Cork City	40673	42923	8
1	Clare	38209	39553	7
2	Cork County	131985	137463	26
3	Cavan	24779	24536	4
4	Carlow	18134	18420	3
5	Dublin City	185165	195589	38
6	Donegal	51460	53248	10
7	D�n Laoghaire-Rathdown	67287	76348	14
8	Fingal	86572	93578	18
9	Galway City	22909	25608	4
10	Galway County	58012	58812	11
11	Kildare	67486	70188	13
12	Kilkenny	32316	33138	6
13	Kerry	49594	51219	10
14	Longford	13092	13066	2
15	Louth	39882	41863	8
16	Limerick City and County	62167	63692	12

	County_Name	Education_Completed_Males	Education_Completed_Females	Education_Completed_I
17	Leitrim	10784	10800	2
18	Laois	26905	26378	5
19	Meath	59539	61840	12
20	Monaghan	19882	20046	3
21	Mayo	43980	44676	8
22	Offaly	24911	25472	5
23	Roscommon	21673	21655	4
24	South Dublin	83369	90921	17
25	Sligo	21073	22257	4
26	Tipperary	52498	53416	10
27	Waterford City and County	37464	38850	7
28	Westmeath	27913	28771	5
29	Wicklow	44976	47431	9
30	Wexford	49095	51511	10

3.6 Independent Variables chosen to represent the Unemployment Count

I have chosen four variables to act as independent variable to predict Unemployment.

First is **Education Completed per County** : Total Number of People who have completed their education in each county. This was chosen because education plays a vital factor in determining if a person can get employed or not.So keeping that in mind, I chose Education Completed as one of the Independent Variable for our model.

Second is **Education InProgress per County** : Total no of people who are still pursuing their education in each county.

Third is **Male Female Ratio Per county** : This takes the male female ratio in each county. So it can be related to unemployment as total population is an important factor.

Fourth one is **People in Working Age** : Total number of people whose age lies between 16 - 70. Assuming that people below this age will not be the right age to be employed and people above this age would be retired. This was based on my assumption and can be modified to requirement.

3.6.1 Checking Normality of the variable Education_Completed_Both

In [855…

```
#Describing the feature Education_Completed_Both
print(county_Education.Education_Completed_Both.describe())
```

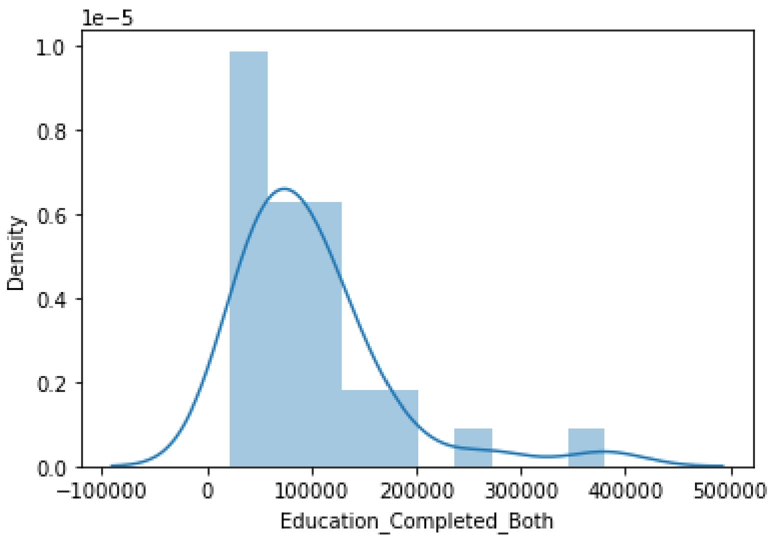
```
count        31.000000
mean      99904.903226
std       74095.697057
min       21584.000000
25%       49849.000000
50%       83596.000000
75%      119101.500000
max      380754.000000
Name: Education_Completed_Both, dtype: float64
```

In [856…

```
#plotting the distribution using distplot
plt.figure()
sns.distplot(county_Education.Education_Completed_Both,hist=True,kde=True,bins=10,no
```

Out[856…

```
<AxesSubplot:xlabel='Education_Completed_Both', ylabel='Density'>
```



By looking at the Numerical distribution values, we see that Education completed count has a median value of 83596. But the mean is slightly higher which is having the value 99904. Since the mean value is higher than the median , it is positively skewed.

This can be clearly viewed by the plot that is created using seaborn. We can see that the blue line is not present in the center as its mean is greater than the median. Hence it is Positively skewed. We can see that the most of the observation lie within 2,00,000 people.

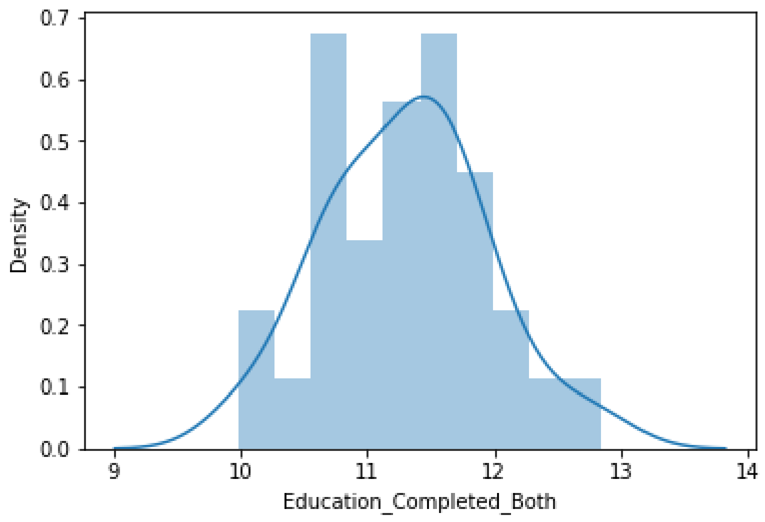
Since the distribution was positively skewed, i did log convert of this distribution in order to obtain a more normal distribution.

In [857…

```
#using Logarithmic method to normalise the Distribution
log_Education_Completed_Both= np.log(county_Education.Education_Completed_Both)
sns.distplot(log_Education_Completed_Both,hist=True,kde=True,bins=10,norm_hist=True)
```

Out[857…

```
<AxesSubplot:xlabel='Education_Completed_Both', ylabel='Density'>
```



After doing the log transformation on the variable, it is more symmetric and is concentrated aroung its median. This distribution is now suitable for our further analysis to determing unemployment count

3.6.2 Checking Normality of the variable Education_InProgress_Total

In [858…

```
#Describing the feature Education_InProgress_Total
print(county_Education.Education_InProgress_Total.describe())
```

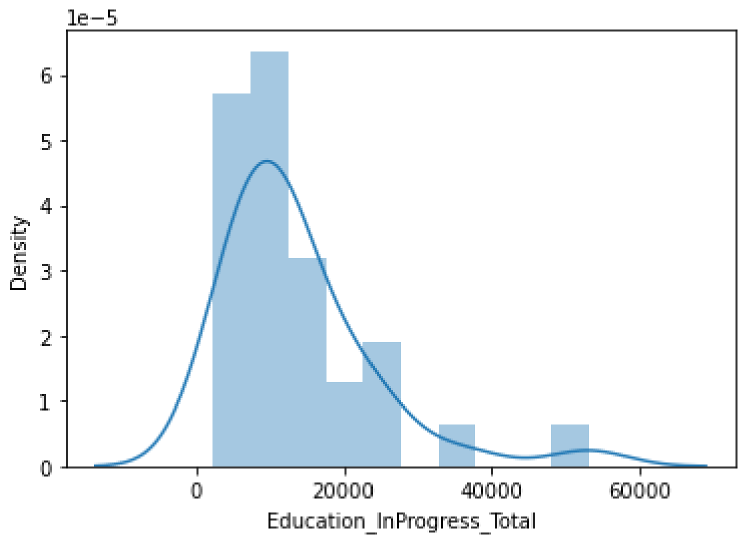
```
count       31.000000
mean     13778.322581
std      10512.017334
min       2343.000000
25%       6470.500000
50%      11145.000000
75%      16219.500000
max      53067.000000
Name: Education_InProgress_Total, dtype: float64
```

In [859…

```
#plotting the distribution using distplot
plt.figure()
sns.distplot(county_Education.Education_InProgress_Total,hist=True,kde=True,bins=10,
```

Out[859…

```
<AxesSubplot:xlabel='Education_InProgress_Total', ylabel='Density'>
```



By looking at the Numerical distribution values, we see that Education in progress count has a median value of 11145. But the mean is slightly higher which is having the value 13778. Since the mean value is higher than the median , it is positively skewed.

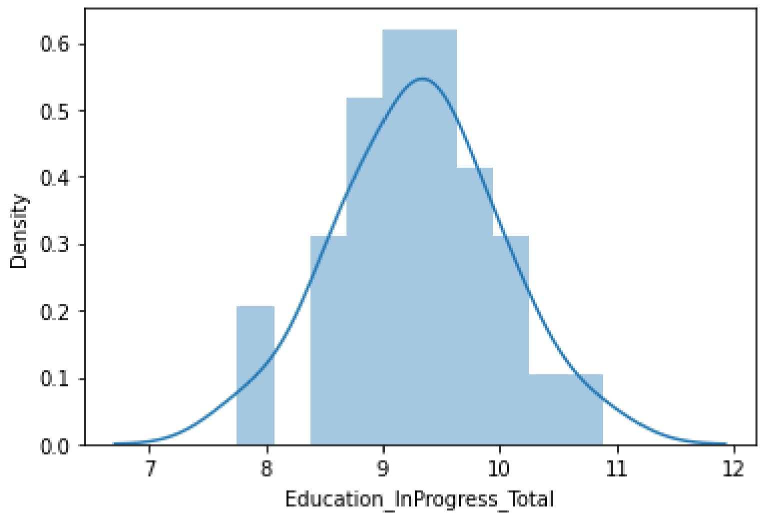
This can be clearly viewed by the plot that is created using seaborn. We can see that the blue line is not present in the center as its mean is greater than the median. Hence it is Positively skewed. We can see that the most of the observation lie within 2,00,000 people.

In [860…

```
#using logarithmic method to normalise the data
log_Education_InProgress_Total= np.log(county_Education.Education_InProgress_Total)
sns.distplot(log_Education_InProgress_Total,hist=True,kde=True,bins=10,norm_hist=Tru
```

Out[860…

```
<AxesSubplot:xlabel='Education_InProgress_Total', ylabel='Density'>
```



After doing the log transformation on the variable, it is more symmetric and is concentrated aroung its median. This distribution is now suitable for our further analysis to determing unemployment count

3.6.2 Checking Normality of the variable Male Female Ratio

In [861…

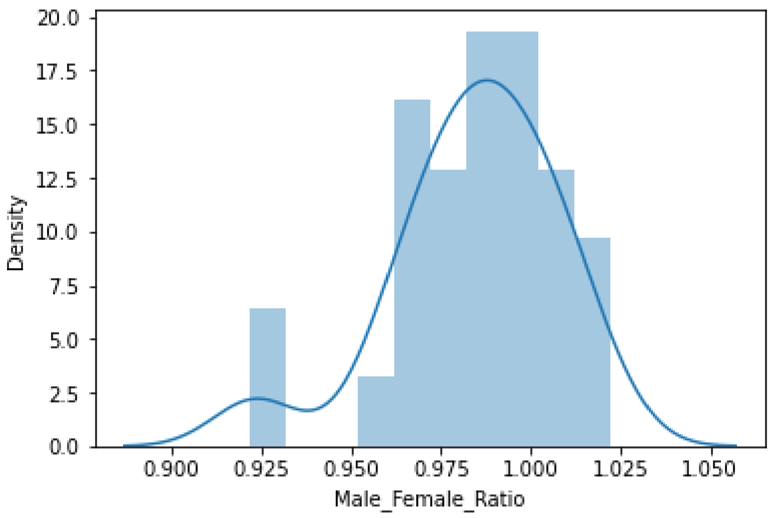
```
## Checking the normality of Male Female Ratio from gender_County
```

```
print(gender_county.Male_Female_Ratio.describe())
sns.distplot( gender_county.Male_Female_Ratio,hist=True,kde=True,bins=10,norm_hist=T
```

```
count    31.000000
mean      0.984333
std       0.023178
min       0.921981
25%       0.972983
50%       0.985680
75%       0.998842
max       1.022084
Name: Male_Female_Ratio, dtype: float64
```

Out[861…

```
<AxesSubplot:xlabel='Male_Female_Ratio', ylabel='Density'>
```



By looking at the Numerical distribution values, we see that Male female ratio median value is almost same as Mean. Hence we can see it is normally distributed since it is a ratio. Looking at the plot as well, we can see that the plot looks to follow normal distribution

3.6.2 Checking Normality of the variable People_In_Working_Age

In [862…

```
# Describing the feature People_In_Working_Age
People_In_Working_Age= Age_Gender_Distribution['Teenage_Male'] + Age_Gender_Distribu
print(People_In_Working_Age.describe())
```

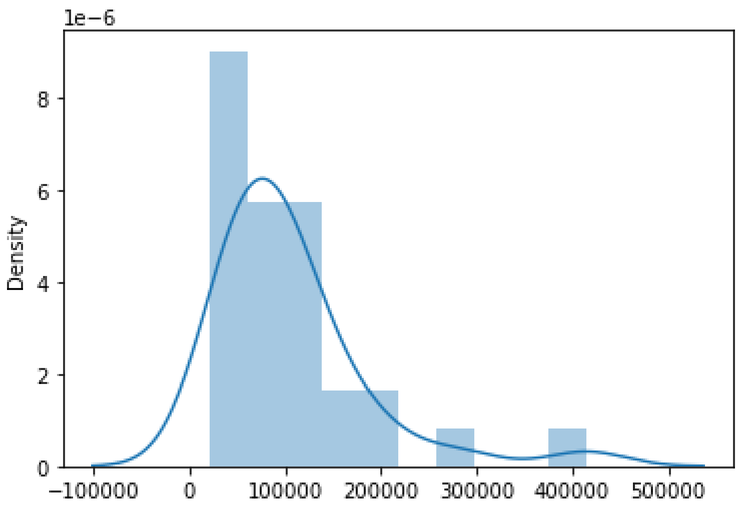
```
count        31.000000
mean     105398.032258
std       80657.855580
min       21070.000000
25%       54187.500000
50%       86868.000000
75%      124310.000000
max      415032.000000
dtype: float64
```

In [863…

```
#plotting the distribution using distplot
plt.figure()
sns.distplot( People_In_Working_Age,hist=True,kde=True,bins=10,norm_hist=True)
```

Out[863…

```
<AxesSubplot:ylabel='Density'>
```

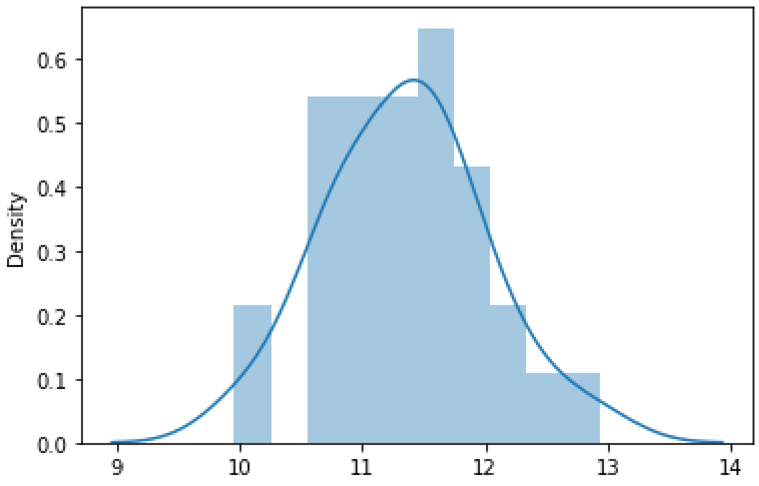


By looking at the Numerical distribution values, we see that People in Working Age has a median value of 86868. But the mean is slightly higher which is having the value 10539. Since the mean value is higher than the median , it is positively skewed.

This can be clearly viewed by the plot that is created using seaborn. We can see that the blue line is not present in the center as its mean is greater than the median. Hence it is Positively skewed. We can see that the most of the observation lie within 2,00,000 people.

```
In [864…
#Making total people in working age as normal distribution
log_People_In_Working_Age= np.log(People_In_Working_Age)
sns.distplot(log_People_In_Working_Age,hist=True,kde=True,bins=10,norm_hist=True)
```

```
Out[864… <AxesSubplot:ylabel='Density'>
```



After doing the log transformation on the variable, it is more symmetric and is concentrated aroung its median. This distribution is now suitable for our further analysis to determing unemployment count

3.6 Building the Linear Regression model

3.6.1 Developing the training set for Linear Regression Model

The *Training_Data* will consist of four feature columns namely *Education_Completed_Total, Education_InProgress_Total, Male_Female_Ratio_County, People_In_Working_Age,Total_Unemployed*. We will use different evaluation methods to determine the best feature to predict unemployment

In [865…

```
########### creating the training set ###########
Training_Data = pd.DataFrame(gender_county['County_Name'])
Training_Data['Education_Completed_Total'] = log_Education_Completed_Both
Training_Data['Education_InProgress_Total'] = log_Education_InProgress_Total
Training_Data['Male_Female_Ratio_County'] = gender_county.Male_Female_Ratio
Training_Data['People_In_Working_Age'] = log_People_In_Working_Age
Training_Data['Total_Unemployed']= unemployment_county['Total_Unemployed']
Training_Data
```

Out[865…

	County_Name	Education_Completed_Total	Education_InProgress_Total	Male_Female_Ratio_County
0	Cork City	11.333751	9.703450	0.965387
1	Clare	11.261408	9.251770	0.979228
2	Cork County	12.504131	10.489411	0.984281
3	Cavan	10.805984	8.701845	1.012789
4	Carlow	10.506546	8.532476	0.999930
5	Dublin City	12.849909	10.879311	0.964525
6	Donegal	11.558931	9.497697	0.985680
7	D�n Laoghaire-Rathdown	11.875031	10.152065	0.921981
8	Fingal	12.101545	10.097120	0.963258
9	Galway City	10.789670	9.322508	0.924929
10	Galway County	11.668424	9.607572	1.003753
11	Kildare	11.832644	9.931054	0.987388
12	Kilkenny	11.089103	9.007490	0.996660
13	Kerry	11.521023	9.379999	0.978607
14	Longford	10.171910	8.026824	1.014838
15	Louth	11.311360	9.318746	0.975203
16	Limerick City and County	11.742918	9.911505	0.997755
17	Leitrim	9.979708	7.759187	1.005257
18	Laois	10.883373	8.778326	1.022084
19	Meath	11.706673	9.684398	0.984817

	County_Name	Education_Completed_Total	Education_InProgress_Total	Male_Female_Ratio_County
20	Monaghan	10.594833	8.505525	1.011337
21	Mayo	11.392519	9.258464	0.993691
22	Offaly	10.827409	8.771680	0.992715
23	Roscommon	10.676554	8.520787	1.006528
24	South Dublin	12.068476	10.078826	0.956397
25	Sligo	10.676601	8.753529	0.975731
26	Tipperary	11.570383	9.467460	0.997284
27	Waterford City and County	11.242612	9.240967	0.985066
28	Westmeath	10.945247	8.968778	0.986439
29	Wicklow	11.433958	9.382612	0.970762
30	Wexford	11.518967	9.309733	0.970026

We take the training set and split it using *train_test_split* with 20% as test set. Then we train the regression model with the training set and determing the Intercepts and Coefficient. Using this Model, then we predict for the unknown test set and compare the prediction with the test data. I have created a dataframe *df_pred* that has the predicted values and the actual values

In [866…

```
######################## Building the Model for prediction ########################

# creating the training set
X= Training_Data[['Education_Completed_Total','Education_InProgress_Total','Male_Fem
y= Training_Data['Total_Unemployed']

# Train test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat

# MultiplE LINEAR Regression

regressor = LinearRegression()
regressor.fit(X_train, y_train)

#Displaying the coefficients of the Model
print('Intercept:', regressor.intercept_)
print('Coefficients:', regressor.coef_)

#predicting for the disjoin test set
y_pred = regressor.predict(X_test)

#Creating a dataframe displaying the actual and predicted value for better understan
df_pred= pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df_pred
```

```
Intercept: -142260.26420505345
Coefficients: [-19717.66392867 -12448.5177756   17205.87513803  41761.55083194]
```

Out[866…

	Actual	Predicted
2	18287	21373.912060
29	8603	11278.738701

	Actual	Predicted
13	8701	11697.713944
10	9938	13300.254993
27	8323	8600.335011
25	4250	1825.672201
22	5744	5673.842080

Linear regression of some dependent variable y on the set of independent variables $\mathbf{x} = (x_1, ..., x_r)$, where r is the number of predictors, you assume a linear relationship between y and x: $\boldsymbol{y = \beta_0 + \beta_1 x_1 + \cdots + \beta_r x_r + \varepsilon}$. This equation is the regression equation. $\beta_0, \beta_1, ..., \beta_r$ are the **regression coefficients**. For our training set we have considered 4 variables. Hence we are getting the 4 coefficients value as [-19717.66392867 -12448.5177756 17205.87513803 41761.55083194] and the intercept value as -142260.2

When we plot the model, we can see that Its not providing an accurate prediction. We can see the model under prediction some of the values and there might be many reasons for this type of behavior. One of the main reason is the high correlation among the features in the training set.

The isolation of the link between each independent variable and the dependent variable is a primary goal of regression analysis. When all of the other independent variables are held constant, a regression coefficient represents the mean change in the dependent variable for each 1 unit change in an independent variable.

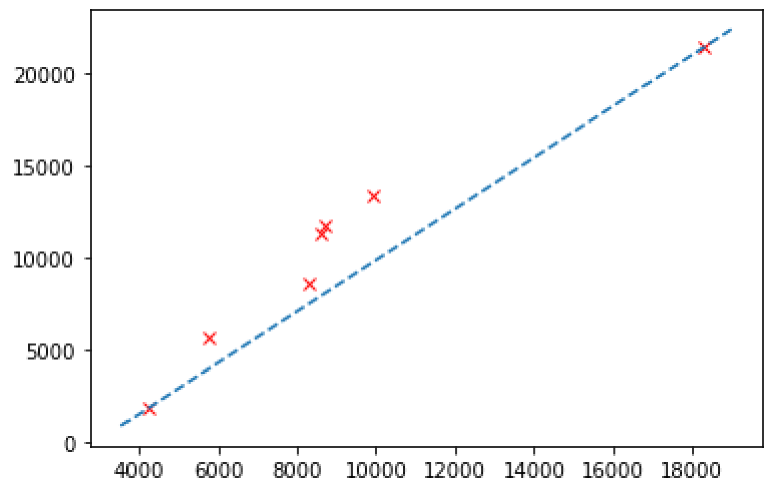
So we need to determine which features are highly correlated and exclude them from the training set to get a better prediction .

In [867…

```
########### Plotting the regression prediction ############################
fig = plt.figure()
plt.plot(y_test,y_pred,'kx',color='r')
plt.plot(plt.xlim(), plt.ylim(), ls="--")
```

Out[867…

```
[<matplotlib.lines.Line2D at 0x218f73d5df0>]
```



We can get many parameter of the model when we fit using stat_model.api. We can create the model using *sm.OLD* and then fit. When we print the summary, we get all kinds of statistical information about the model.

In [868…
```
####### Getting the model description using statsmodel api ################

X_train.insert(0,'intercept',1)     #To display the intercept
mod = sm.OLS(y_train,X_train)
res = mod.fit()
print(res.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:      Total_Unemployed   R-squared:                       0.760
Model:                           OLS   Adj. R-squared:                  0.710
Method:                Least Squares   F-statistic:                     15.06
Date:               Mon, 20 Dec 2021   Prob (F-statistic):           1.06e-05
Time:                       09:38:18   Log-Likelihood:                -230.47
No. Observations:                 24   AIC:                             470.9
Df Residuals:                     19   BIC:                             476.8
Df Model:                          4                                         
Covariance Type:           nonrobust                                         
========================================================================================
==========
                                  coef    std err          t      P>|t|      [0.025      
0.975]
----------------------------------------------------------------------------------------
----------
intercept                   -1.423e+05   5.59e+04     -2.546      0.020   -2.59e+05   
-2.53e+04
Education_Completed_Total   -1.972e+04   2.35e+04     -0.839      0.412   -6.89e+04   
2.94e+04
Education_InProgress_Total  -1.245e+04   9518.552     -1.308      0.207   -3.24e+04   
7474.040
Male_Female_Ratio_County     1.721e+04   5.41e+04      0.318      0.754   -9.61e+04   
1.3e+05
People_In_Working_Age        4.176e+04   2.88e+04      1.452      0.163   -1.84e+04   
1.02e+05
==============================================================================
Omnibus:                      26.104   Durbin-Watson:                   2.219
Prob(Omnibus):                 0.000   Jarque-Bera (JB):               49.443
Skew:                          2.036   Prob(JB):                     1.84e-11
Kurtosis:                      8.733   Cond. No.                     1.73e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
[2] The condition number is large, 1.73e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Identifying the correlation between the features

Function *remove_collinear_features* removes collinear features in a dataframe with a correlation coefficient greater than the threshold. Removing collinear features can help a model to generalize and improves the interpretability of the model. This takes the training set and particular threshold value for the correlation. It then removes the features having correlation greater than the threshold and returns a dataset having independent variable in comparison to threshold. It will display the features that are removed and which have higher correlation value. We took the threshold value as 0.6 for our data

In [869…
```
#Function which takes the trainingset and removes the highly correlated features
def remove_collinear_features(x, threshold):
    # Calculate the correlation matrix
    corr_matrix = x.corr()
    iters = range(len(corr_matrix.columns) - 1)
    drop_cols = []
    # Iterate through the matrix and compare correlations
    for i in iters:
        for j in range(i+1):
            item = corr_matrix.iloc[j:(j+1), (i+1):(i+2)]
            col = item.columns
            row = item.index
            val = abs(item.values)

            # If correlation exceeds the threshold
            if val >= threshold:
                # Print the correlated features and the correlation value
                print(col.values[0], "|", row.values[0], "|", round(val[0][0], 2))
                drop_cols.append(col.values[0])

    # Drop one of each pair of correlated columns
    drops = set(drop_cols)
    x = x.drop(columns=drops)

    return x
```

In [870…

```
#Creating a new training set without highly correlated features
New_Training_Data=remove_collinear_features(X,0.6)
```

```
Education_InProgress_Total | Education_Completed_Total | 0.97
People_In_Working_Age | Education_Completed_Total | 1.0
People_In_Working_Age | Education_InProgress_Total | 0.98
```

Observations

We find that Education_InProgress_Total and Education_Completed_Total are having a high correlation score of 0.97. Similarly People_In_Working_Age and Education_Completed_Total fully correlated. People_In_Working_Age and Education_InProgress_Total have a high correlation score of 0.98. These features do not show the independence property. So the new training set is void of these features. It consists of only Male_Female_Ratio_County and Education_Completed_Total. We now should rebuild the model using these new features and check for accuracy.

In [871…

```
#Rebuilding the model with new updated training set

X_train, X_test, y_train, y_test = train_test_split(New_Training_Data, y, test_size=

##### MultiplE LINEAR rEGRESSION #########
regressor = LinearRegression()
regressor.fit(X_train, y_train)
print('Intercept:', regressor.intercept_)
print('Coefficients:', regressor.coef_)

#predicting for the disjoin test set
y_pred = regressor.predict(X_test)
```

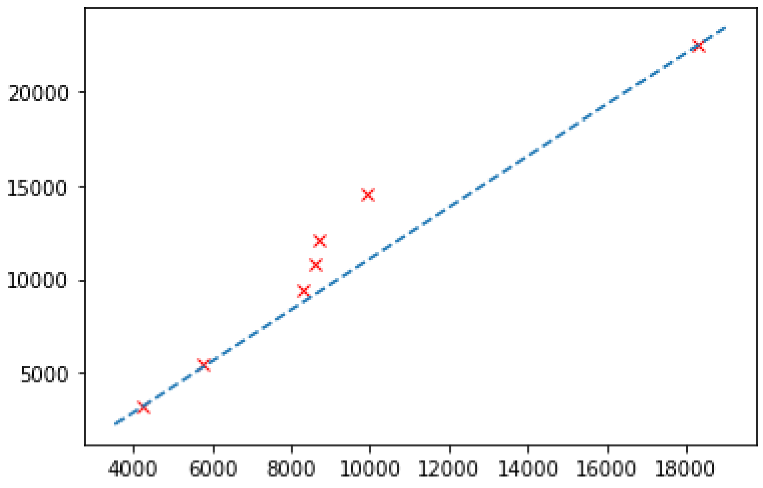
```
Intercept: -146370.49698109238
Coefficients: [10332.93639891 40243.37546324]
```

In [872…

```
#Plotting the regression model using the two features
fig = plt.figure()
plt.plot(y_test,y_pred,'kx',color='r')
plt.plot(plt.xlim(), plt.ylim(), ls="--")
```

Out[872…

```
[<matplotlib.lines.Line2D at 0x218f743d2e0>]
```



Measure the better model,based on AIC score

we can see that the model fits better than the previous case. We can say that the model is performing better by having a look at the AIC score of the model. The later model AIC score is lower than that of the earlier model.

In [873…

```python
#######Model Validation using sm ################

X_train.insert(0,'intercept',1)     #To display the intercept
mod = sm.OLS(y_train,X_train)
res = mod.fit()
print(res.summary())
```

```
                            OLS Regression Results                            
==============================================================================
Dep. Variable:       Total_Unemployed   R-squared:                       0.731
Model:                            OLS   Adj. R-squared:                  0.706
Method:                 Least Squares   F-statistic:                     28.60
Date:                Mon, 20 Dec 2021   Prob (F-statistic):           1.01e-06
Time:                        09:38:21   Log-Likelihood:                -231.83
No. Observations:                  24   AIC:                             469.7
Df Residuals:                      21   BIC:                             473.2
Df Model:                           2                                         
Covariance Type:            nonrobust                                         
======================================================================================
=========
                                coef    std err          t      P>|t|      [0.025      
0.975]
--------------------------------------------------------------------------------------
---------
intercept                 -1.464e+05   4.86e+04     -3.010      0.007   -2.47e+05   
-4.53e+04
Education_Completed_Total  1.033e+04   1489.147      6.939      0.000    7236.085   
1.34e+04
Male_Female_Ratio_County   4.024e+04   3.83e+04      1.051      0.305   -3.94e+04   
1.2e+05
==============================================================================
Omnibus:                       24.821   Durbin-Watson:                   2.358
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               42.998
Skew:                           1.988   Prob(JB):                     4.60e-10
Kurtosis:                       8.214   Cond. No.                         842.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

Verifying The best feature based on P-value

If the p-value for a variable is less than your significance level, your sample data provide enough evidence to reject the null hypothesis for the entire population. Your data favor the hypothesis that there is a non-zero correlation. Changes in the independent variable are associated with changes in the dependent variable at the population level. This variable is statistically significant and probably a worthwhile addition to your regression model.On the other hand, a p-value that is greater than the significance level indicates that there is insufficient evidence in your sample to conclude that a non-zero correlation exists.

Looking at the above summary we can see that p value for Male_Female_Ratio_County is greater than 5% . Hence it does not add any value for the prediction made by the model. Hence we will consider only Education_Completed_Total

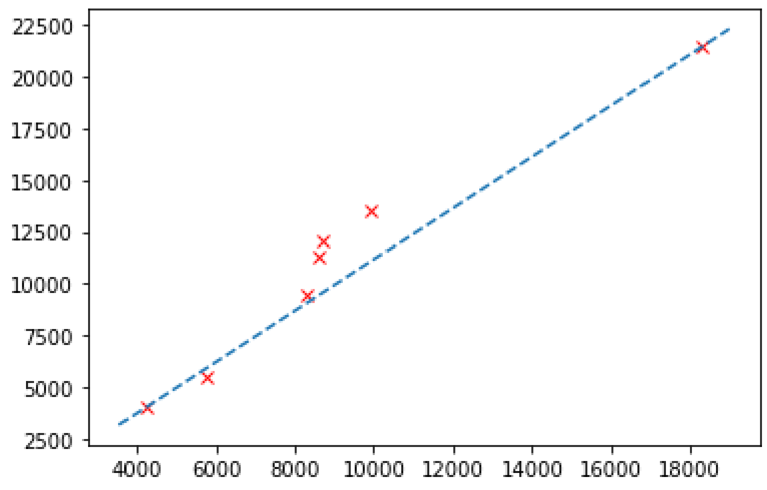
Building the model using the best feature

In [874…

```
Final_Training_Set = Training_Data[['Education_Completed_Total']]
X_train, X_test, y_train, y_test = train_test_split(Final_Training_Set, y, test_size

##### LINEAR rEGRESSION #########
regressor = LinearRegression()
regressor.fit(X_train, y_train)

#predicting for the disjoin test set
y_pred = regressor.predict(X_test)

#plotting the model
fig = plt.figure()
plt.plot(y_test,y_pred,'kx',color='r')
plt.plot(plt.xlim(), plt.ylim(), ls="--")
```

Out[874…

```
[<matplotlib.lines.Line2D at 0x218f76bbc40>]
```



In [875…

```
#######Model Validation using sm ###############

X_train.insert(0,'intercept',1)     #To display the intercept
mod = sm.OLS(y_train,X_train)
res = mod.fit()
print(res.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         Total_Unemployed   R-squared:                       0.717
```

```
Model:                            OLS   Adj. R-squared:                  0.704
Method:                 Least Squares   F-statistic:                     55.83
Date:                Mon, 20 Dec 2021   Prob (F-statistic):           1.80e-07
Time:                        09:38:23   Log-Likelihood:                -232.44
No. Observations:                  24   AIC:                             468.9
Df Residuals:                      22   BIC:                             471.2
Df Model:                           1                                         
Covariance Type:            nonrobust                                         
=====================================================================================
=========
                              coef    std err          t      P>|t|      [0.025      
0.975]
-------------------------------------------------------------------------------------
---------
intercept               -9.756e+04   1.44e+04     -6.783      0.000   -1.27e+05   
-6.77e+04
Education_Completed_Total  9517.1426   1273.705      7.472      0.000    6875.641    
1.22e+04
==============================================================================
Omnibus:                       24.863   Durbin-Watson:                   2.087
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               48.363
Skew:                           1.880   Prob(JB):                     3.15e-11
Kurtosis:                       8.850   Cond. No.                         197.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly spe
cified.
```

Interpreting the Model

By looking at the model we can see that the model is fairly better than the earlier as it has a lower AIC score. Similarly the F-statistic score of 55 % and p value of 0% indicate that there is sufficient evidence to reject the null hypothesis. The adjusted Rsquared value for this model indicates that approximately 70% of the variance of Total_Unemployment was explained by the predictor variables included in the model.

4 CONCLUSIONS

In this project we were trying to determine the best possible predictor variable to predict the unemployement per county. We initially considered four set of variables *Education_Completed, Education_In_Progress, Male_Female_Ratio and Total_People_In_Working_Age* . By just having a glance at the variable, we feel that this might account for linear relationship with the total unemployment count. But When we try top infer this using various methods, we come to know the true picture.

For Linear Regression model , we assume that the predictor variables are normally distributed and independent. If these assumption hold true, then only we can get a meaningfull model. So to verify this, we checked if all the features are normally distributed. But they were not, So we used logarithmic method to make them normally distributed. Them using these New data we tried to find which is the best feature.

In this report, I have mainly used two validation methods to determine the feature importance. One is *Correlation score* and other is *P-value score*. First I filtered out the features having very high correlation using a user defined function remove_collinear_features for which we defined the threshold as 0.6. All the features which had higher correlation score was removed. High correlation means they are dependent, Hence it violates the Regression Analysis assumption. So we removed *Education_InProgress_Total and People_In_Working_Age* .

Now with the remaining two features, I build the linear regression model and tried to verify the P-score. The P-Value provides probability of the hypothesis test, So in a regression model the P-Value for each independent variable tests the Null Hypothesis that there is “No Correlation” between the independent and the dependent variable,this also helps to determine the relationship observed in the sample also exists in the larger data.So if the P-Value is less than the significance level (usually 0.05) then your model fits the data well. In our case, p-value is less than 0.05 only for *Education_Completed* which means that the model fits for this feature better than the rest of them.

So we can conclude that, The best predictor for Total Unemployment in each county is Education_Completed based on all the factors and we were able to find a linear relationship with the variable.

However we can see that, the model accuracy is not perfect. There is some error. This section needs further investigation to determine if the linear regression is the best model fot this dataset. There can be complex models which can provide better results for this dataset. This can be part of the furher research which can be done on the Census Data.