

IT EMPLOYMENT STATUS-EDA

It is an everliving jobs in world So that we analyze IT EMPLOYMENT STATUS in 2020 year. The data set is extracted from [kaggle.com](https://www.kaggle.com) The tools which I used is jupyter notebook and some libraries such as pandas,numpy and matplotlib,seaborn .The course [Data Analysis with Python: Zero to Pandas](#), I learned pandas and plotting libraries etc.

Downloading the Dataset

*I download the dataset from [kaggle.com](https://www.kaggle.com) * link for dataset is given below

<https://www.kaggle.com/datasets/parulpandey/2020-it-salary-survey-for-eu-region>

```
!pip install jovian opendatasets --upgrade --quiet
```

Let's begin by downloading the data, and listing the files within the dataset.

The dataset has been downloaded and extracted.

Let us save and upload our work to Jovian before continuing.

```
project_name = "EDA-IT employ in Europe"
```

```
import opendatasets as od dataset_url = 'https://www.kaggle.com/datasets/parulpandey/2020-it-salary-survey-for-eu-region' od.download('https://www.kaggle.com/datasets/parulpandey/2020-it-salary-survey-for-eu-region')
```

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

```
data_df=pd.read_csv('IT Salary Survey EU 2020(1).csv')
data_df
```

	Timestamp	Age	Gender	City	Position	Total years of experience	Years of experience in Germany	Seniority level	Your main technology / programming language	technology language
0	11/24/2020 11:14	26.0	Male	Munich	Software Engineer	5	3	Senior	TypeScript	
1	11/24/2020 11:14	26.0	Male	Berlin	Backend Developer	7	4	Senior	Ruby	
2	11/24/2020 11:14	29.0	Male	Berlin	Software Engineer	12	6	Lead	Javascript / Typescript	Java

	Timestamp	Age	Gender	City	Position	Total years of experience	Years of experience in Germany	Seniority level	Your main technology / programming language	technologies / languages
3	11/24/2020 11:15	28.0	Male	Berlin	Frontend Developer	4	1	Junior	Javascript	
4	11/24/2020 11:15	37.0	Male	Berlin	Backend Developer	17	6	Senior	C# .NET	.NET
...	
1248	1/18/2021 15:05	31.0	Male	Berlin	Backend Developer	9	5	Senior	Java	Python, Typescript
1249	1/18/2021 17:46	33.0	Male	Berlin	Researcher/ Consumer Insights Analyst	10	1.5	Senior	consumer analysis	
1250	1/18/2021 23:20	39.0	Male	Munich	IT Operations Manager	15	2	Lead	PHP	Python, / T
1251	1/19/2021 10:17	26.0	Male	Saarbrücken	Frontend Developer	7	7	Middle	JavaScript	Java Docker
1252	1/19/2021 12:01	26.0	Male	Berlin	DevOps	2	2	Middle	yaml	Python Docker

1253 rows × 21 columns

```
!pip install jovian --upgrade -q
```

```
import jovian
```

```
jovian.commit(project="EDA-IT employ in Europe")
```

```
[jovian] Creating a new project "redraven46/EDA-IT employ in Europe"
```

```
[jovian] Committed successfully! https://jovian.ai/redraven46/eda-it-employ-in-europe
```

```
'https://jovian.ai/redraven46/eda-it-employ-in-europe'
```

Data Preparation and Cleaning

In this section we go through data cleaning and preparation for other process. We use pandas and numpy to cleaning and preparation. Lets start this.

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

```
data_df
```

	Timestamp	Age	Gender	City	Position	Total years of experience	Years of experience in Germany	Seniority level	Your main technology / programming language	technologies / languages
0	11/24/2020 11:14	26.0	Male	Munich	Software Engineer	5	3	Senior	TypeScript	Python, JavaScript, TypeScript
1	11/24/2020 11:14	26.0	Male	Berlin	Backend Developer	7	4	Senior	Ruby	Python, JavaScript, Ruby
2	11/24/2020 11:14	29.0	Male	Berlin	Software Engineer	12	6	Lead	Javascript / Typescript	Java, JavaScript, TypeScript
3	11/24/2020 11:15	28.0	Male	Berlin	Frontend Developer	4	1	Junior	Javascript	JavaScript
4	11/24/2020 11:15	37.0	Male	Berlin	Backend Developer	17	6	Senior	C# .NET	.NET
...
1248	1/18/2021 15:05	31.0	Male	Berlin	Backend Developer	9	5	Senior	Java	Python, JavaScript, Java, TypeScript
1249	1/18/2021 17:46	33.0	Male	Berlin	Researcher/ Consumer Insights Analyst	10	1.5	Senior	consumer analysis	Python, JavaScript, Java, TypeScript
1250	1/18/2021 23:20	39.0	Male	Munich	IT Operations Manager	15	2	Lead	PHP	Python, JavaScript, PHP, TypeScript
1251	1/19/2021 10:17	26.0	Male	Saarbrücken	Frontend Developer	7	7	Middle	JavaScript	Java, JavaScript, Docker
1252	1/19/2021 12:01	26.0	Male	Berlin	DevOps	2	2	Middle	yaml	Python, Docker

1253 rows × 21 columns

```
data_df.shape
```

```
(1253, 21)
```

In this step we are going to drop the null values.

```
new_data_df=data_df.dropna()  
new_data_df
```

	Timestamp	Age	Gender	City	Position	Total years of experience	Years of experience in Germany	Seniority level	Your main technology / programming language	technologi languag
0	11/24/2020 11:14	26.0	Male	Munich	Software Engineer	5	3	Senior	TypeScript	Ko
2	11/24/2020 11:14	29.0	Male	Berlin	Software Engineer	12	6	Lead	Javascript / Typescript	Javasc
5	11/24/2020 11:15	32.0	Male	Berlin	DevOps	5	1	Senior	AWS, GCP, Python,K8s	Pyth Cloud, Ku
19	11/24/2020 11:20	34.0	Male	Berlin	Mobile Developer	11	5	Lead	kotlin	Ko Typescr
25	11/24/2020 11:25	38.0	Male	Berlin	Team Lead	18	4	Lead	iOS	
...	
1237	1/12/2021 12:48	37.0	Male	Berlin	Mobile Developer	15	5	Senior	Android	
1244	1/16/2021 22:23	32.0	Male	Munich	Software Engineer	10	5	Head	Scala	Javasc Java /
1248	1/18/2021 15:05	31.0	Male	Berlin	Backend Developer	9	5	Senior	Java	Pyth Typescrip
1251	1/19/2021 10:17	26.0	Male	Saarbrücken	Frontend Developer	7	7	Middle	JavaScript	Javasc Docker, I
1252	1/19/2021 12:01	26.0	Male	Berlin	DevOps	2	2	Middle	yaml	Python, A Docker

490 rows × 21 columns

The duplicate values is given

```
new_data_df.Timestamp.drop_duplicates()
```

```
1217    1/1/2021 13:40
1236    1/11/2021 12:12
1237    1/12/2021 12:48
1244    1/16/2021 22:23
1248    1/18/2021 15:05
...
1121    12/8/2020 1:09
1130    12/8/2020 20:04
1136    12/8/2020 23:46
1123    12/8/2020 8:17
```

1137 12/9/2020 4:46

Name: Timestamp, Length: 438, dtype: object

```
new_data_df.sort_values("Timestamp", inplace=True)
new_data_df.drop_duplicates(subset="Timestamp", keep=False, inplace=True)
new_data_df
```

/opt/conda/lib/python3.9/site-packages/pandas/util/_decorators.py:311:

SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
return func(*args, **kwargs)
```

	Timestamp	Age	Gender	City	Position	Experience_years	Years of experience in Germany	Seniority level	Technology	technolog language
1217	1/1/2021 13:40	34	Male	Prague	Backend Developer	14	0	Lead	C#	Java .NET,
1236	1/11/2021 12:12	39	Male	Berlin	QA Engineer	7	5	Senior	Java	Ko
1237	1/12/2021 12:48	37	Male	Berlin	Mobile Developer	15	5	Senior	Android	
1244	1/16/2021 22:23	32	Male	Munich	Software Engineer	10	5	Head	Scala	Java Java
1248	1/18/2021 15:05	31	Male	Berlin	Backend Developer	9	5	Senior	Java	Py Typescrip
...
1121	12/8/2020 1:09	35	Male	Frankfurt	Software Engineer	10	1	Senior	Java	Python, J SQL
1130	12/8/2020 20:04	29	Male	Berlin	DevOps	7	3	Senior	Kubernetes, Openstack	Python [
1136	12/8/2020 23:46	34	Male	Cologne	CTO	14	2	Lead	C#	Java .NET,
1123	12/8/2020 8:17	27	Male	Munich	Software Engineer	2	5	Junior	Java	Java
1137	12/9/2020 4:46	42	Male	Hamburg	Data Scientist	2	2	Senior	Python	Python, S

438 rows x 21 columns

```
new_data_df.shape
```

(440, 21)

```
data_df.describe()
```

	Age	Yearly brutto salary (without bonus and stocks) in EUR	Annual brutto salary (without bonus and stocks) one year ago. Only answer if staying in the same country
count	1226.000000	1.253000e+03	8.850000e+02
mean	32.509788	8.027904e+07	6.322459e+05
std	5.663804	2.825061e+09	1.680508e+07
min	20.000000	1.000100e+04	1.100000e+04
25%	29.000000	5.880000e+04	5.500000e+04
50%	32.000000	7.000000e+04	6.500000e+04
75%	35.000000	8.000000e+04	7.500000e+04
max	69.000000	1.000000e+11	5.000000e+08

```
new_data_df.info()
```

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

```
Int64Index: 440 entries, 1217 to 1137
```

```
Data columns (total 21 columns):
```

#	Column	Non-Null
Count	Dtype	
---	-----	-----
0	Timestamp	438 non-
null	object	
1	Age	438 non-
null	float64	
2	Gender	438 non-
null	object	
3	City	438 non-
null	object	
4	Position	438 non-
null	object	
5	Experience_years	438 non-
null	object	
6	Years of experience in Germany	438 non-
null	object	
7	Seniority level	438 non-
null	object	
8	Technology	438 non-
null	object	
9	Other technologies/programming languages you use often	438 non-
null	object	
10	Yearly brutto salary (without bonus and stocks) in EUR	438 non-
null	float64	
11	Yearly bonus + stocks in EUR	438 non-

```

null    object
   12 Annual_Salary                                438 non-
null    float64
   13 Annual bonus+stocks one year ago. Only answer if staying in same country 438 non-
null    object
   14 Number of vacation days                      438 non-
null    object
   15 Employment status                            438 non-
null    object
   16 contract duration                            438 non-
null    object
   17 Main language at work                        438 non-
null    object
   18 Company size                                438 non-
null    object
   19 Company_Type                                438 non-
null    object
   20 Have you lost your job due to the coronavirus outbreak? 438 non-
null    object
dtypes: float64(3), object(18)
memory usage: 75.6+ KB

```

We are going to rename and convert the columns names and properties for our use lets start

```
new_data_df.rename(columns={'Your main technology / programming language':'Technology'})
```

```
new_data_df.rename(columns={'Company type':'Company_Type'},inplace=True)
```

```
new_data_df.rename(columns={'Annual brutto salary (without bonus and stocks) one year a
```

```
new_data_df.rename(columns={'Total years of experience':'Experience_years'},inplace=Tru
```

```
new_data_df.where(~(new_data_df.Gender.str.contains('; ', na=False)), np.nan, inplace=Tr
```

```
new_data_df.where(~(new_data_df.Technology.str.contains('; ', na=False)), np.nan, inplac
```

```
new_data_df.info()
```

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

```
Int64Index: 440 entries, 1217 to 1137
```

```
Data columns (total 21 columns):
```

#	Column	Non-Null
Count	Dtype	
---	-----	-----
0	Timestamp	440 non-
null	object	
1	Age	440 non-
null	float64	
2	Gender	440 non-
null	object	
3	City	440 non-
null	object	
4	Position	440 non-
null	object	
5	Experience_years	440 non-
null	object	
6	Years of experience in Germany	440 non-
null	object	
7	Seniority level	440 non-
null	object	
8	Technology	440 non-
null	object	
9	Other technologies/programming languages you use often	440 non-
null	object	
10	Yearly brutto salary (without bonus and stocks) in EUR	440 non-
null	float64	
11	Yearly bonus + stocks in EUR	440 non-
null	object	
12	Annual_Salary	440 non-
null	float64	
13	Annual bonus+stocks one year ago. Only answer if staying in same country	440 non-
null	object	
14	Number of vacation days	440 non-
null	object	
15	Employment status	440 non-
null	object	
16	contract duration	440 non-
null	object	
17	Main language at work	440 non-
null	object	
18	Company size	440 non-
null	object	
19	Company_Type	440 non-


```
null    object
```

```
20 Have you lost your job due to the coronavirus outbreak?
```

```
440 non-
```

```
null    object
```

```
dtypes: float64(3), object(18)
```

```
memory usage: 75.6+ KB
```

```
new_data_df.where(~(new_data_df.City.str.contains('; ', na=False)), np.nan, inplace=True)
```

```
new_data_df.where(~(new_data_df.Experience_years.str.contains(' ', na=False)), np.nan,
```

```
new_data_df['Age'] = new_data_df['Age'].astype(np.int32)
```

```
/tmp/ipykernel_37/3838825615.py:1: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
new_data_df['Age'] = new_data_df['Age'].astype(np.int32)
```

```
new_data_df.describe()
```

	Age	Yearly brutto salary (without bonus and stocks) in EUR	Annual_Salary
count	438.000000	438.000000	438.000000
mean	32.858447	75189.680365	67073.646119
std	5.280079	44038.375530	21617.634422
min	22.000000	12000.000000	11000.000000
25%	29.000000	60000.000000	55000.000000
50%	32.500000	71405.000000	66000.000000
75%	36.000000	83000.000000	75750.000000
max	56.000000	850000.000000	200000.000000

```
new_data_df
```

	Timestamp	Age	Gender	City	Position	Experience_years	Years of experience in Germany	Seniority level	Technology	technology language
1217	1/1/2021 13:40	34	Male	Prague	Backend Developer	14	0	Lead	C#	Java, .NET, Kotlin
1236	1/11/2021 12:12	39	Male	Berlin	QA Engineer	7	5	Senior	Java	Kotlin

	Timestamp	Age	Gender	City	Position	Experience_years	Years of experience in Germany	Seniority level	Technology	technolog langua
1237	1/12/2021 12:48	37	Male	Berlin	Mobile Developer	15	5	Senior	Android	
1244	1/16/2021 22:23	32	Male	Munich	Software Engineer	10	5	Head	Scala	Java
1248	1/18/2021 15:05	31	Male	Berlin	Backend Developer	9	5	Senior	Java	Py Typescrip
...
1121	12/8/2020 1:09	35	Male	Frankfurt	Software Engineer	10	1	Senior	Java	Python, .NET, SQL
1130	12/8/2020 20:04	29	Male	Berlin	DevOps	7	3	Senior	Kubernetes, Openstack	Python, .NET, SQL
1136	12/8/2020 23:46	34	Male	Cologne	CTO	14	2	Lead	C#	Java, .NET, SQL
1123	12/8/2020 8:17	27	Male	Munich	Software Engineer	2	5	Junior	Java	Java
1137	12/9/2020 4:46	42	Male	Hamburg	Data Scientist	2	2	Senior	Python	Python, SQL

438 rows × 21 columns

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "redraven46/eda-it-employ-in-europe" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/redraven46/eda-it-employ-in-europe
'https://jovian.ai/redraven46/eda-it-employ-in-europe'
```

Exploratory Analysis and Visualization

We are going to visualize our data corresponding to our [use.In](#) this we are going to explore 4 visualize 1-Barplot 2-piechart 3-histogram 4-graph

Let's begin by importing matplotlib.pyplot and seaborn .

```
import seaborn as sns
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline

sns.set_style('darkgrid')
matplotlib.rcParams['font.size'] = 14
matplotlib.rcParams['figure.figsize'] = (9, 5)
matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

The First Plot

we 1st plot a barplot between top technologu and Annual Salary.

```
new_data_df.Technology.nunique()
```

122

```
top_technology=new_data_df.Technology.value_counts().head(15)
top_technology
```

Java	78
Python	66
C++	18
C#	14
PHP	13
python	13
Python	12
Swift	12
JavaScript	11
Go	11
Kotlin	9
.NET	7
Javascript	7
Scala	7
TypeScript	6

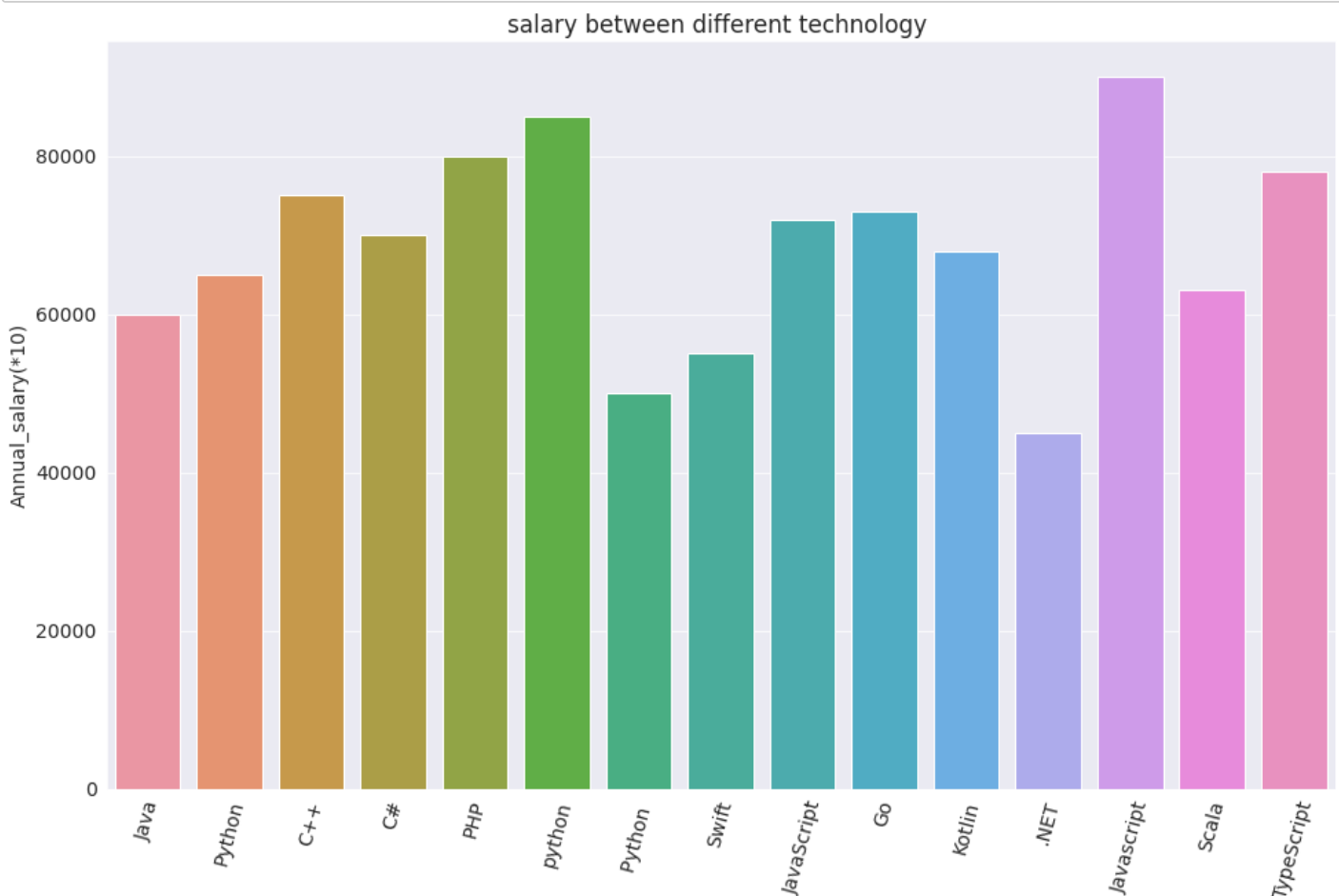
Name: Technology, dtype: int64

```
top_salary=new_data_df.Annual_Salary.value_counts().head(15)
top_salary
```

60000.0	36
65000.0	33
75000.0	27
70000.0	26
80000.0	19
85000.0	14
50000.0	13
55000.0	12
72000.0	11
73000.0	10
68000.0	10
45000.0	9
90000.0	8
63000.0	8
78000.0	7

Name: Annual_Salary, dtype: int64

```
plt. figure(figsize=(16,10))
plt.xticks(rotation=75)
plt.ylabel("Annual_salary(*10)")
plt.title("salary between different technology ")
sns.barplot(x=top_technology.index, y=top_salary.index);
```



we first get top salary and top technology . after We use seaborn library to plot a Barplot.From this plot we can conclude that javascript and python languages are getting high salary

The Second plot

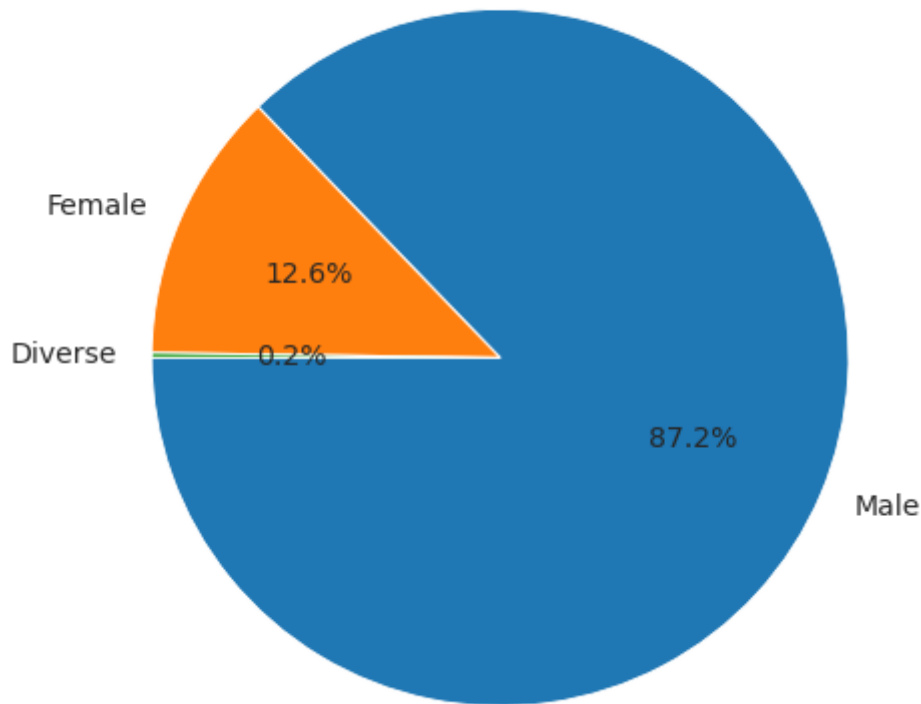
we do a pie chart for comparing men and women labours

```
gender_counts = new_data_df.Gender.value_counts()
gender_counts
```

```
Male      382
Female    55
Diverse    1
Name: Gender, dtype: int64
```

```
plt.figure(figsize=(14,8))
plt.title("difference between Gender ")
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.1f%%', startangle=180);
```

difference between Gender



Fisrt we get the male and women counts next we use the plt function to plot the piechart. from the we conclude that male labours mostly higher than female labours. And a little amount of Diverse

The third plot

we plot a histogram between company type and their salary

```
company_type=new_data_df.Company_Type.nunique()  
company_type
```

32

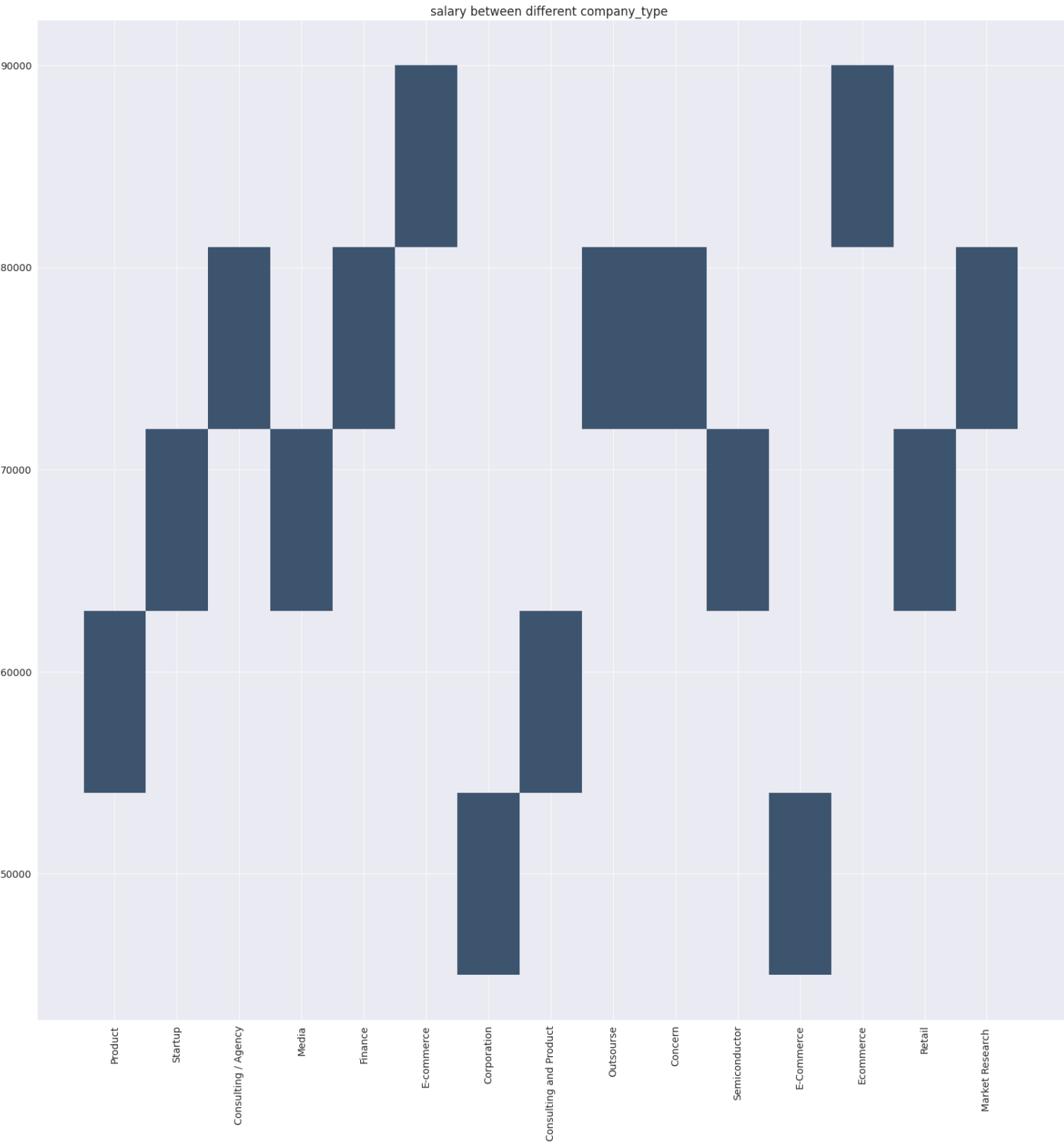
TODO - Explore one or more columns by plotting a graph below, and add some explanation about it

```
company_type=new_data_df.Company_Type.value_counts().head(15)  
company_type
```

Product	284
Startup	75
Consulting / Agency	48
Media	2
Finance	2
E-commerce	2
Corporation	2
Consulting and Product	1
Outsource	1
Concern	1

Semiconductor 1
E-Commerce 1
Ecommerce 1
Retail 1
Market Research 1
Name: Company_Type, dtype: int64

```
plt. figure(figsize=(26,26))  
plt.xticks(rotation=90)  
plt.title("salary between different company_type ")  
sns.histplot(x=company_type.index, y=top_salary.index);
```

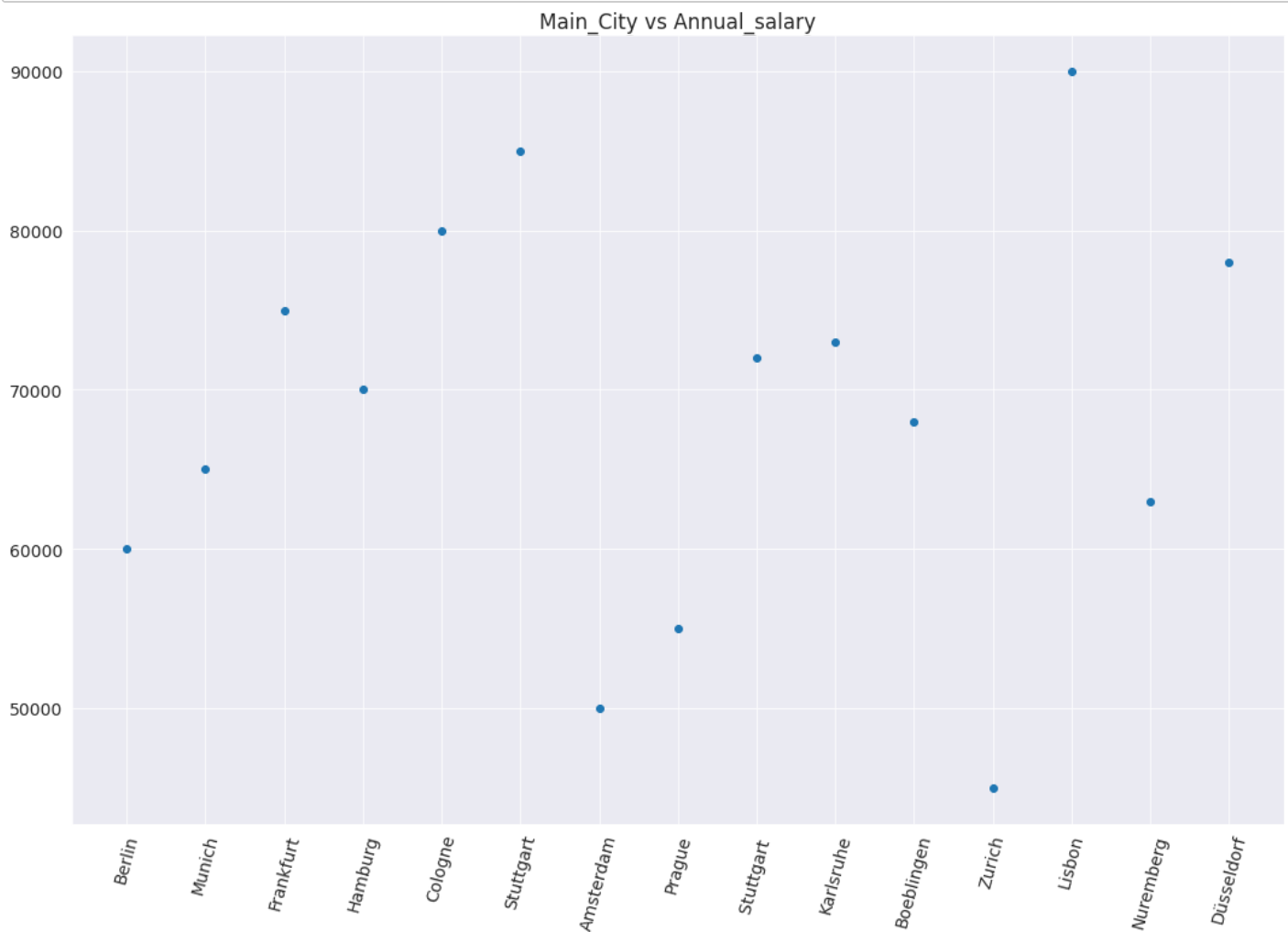


Getting company type we plot a histogram from the plot come to know that E commerce company and market reasearch company has high salary workers.

The fourth Plot

we plot a graph between
salary and main city.

```
plt.figure(figsize=(18,12))
plt.xticks(rotation=75)
plt.scatter(Main_City.index,top_salary.index);
plt.title("Main_City vs Annual_salary");
```



Getting top main city From the plot we can conclude that lisbon has high salary

Let us save and upload our work to Jovian before continuing

```
import jovian
```

```
jovian.commit()
```

[jovian] Updating notebook "redraven46/eda-it-employ-in-europe" on <https://jovian.ai>

[jovian] Committed successfully! <https://jovian.ai/redraven46/eda-it-employ-in-europe>
'<https://jovian.ai/redraven46/eda-it-employ-in-europe>'

ANSWERING SOME QUESTIONS

1. which city has highest number of companies ?

```
Main_City=new_data_df.City.value_counts().head(15)
Main_City
```

```
Berlin      225
Munich      92
Frankfurt   19
Hamburg     14
Cologne     6
Stuttgart   6
Amsterdam   4
Prague      4
Stuttgart   3
Karlsruhe   3
Boeblingen  2
Zurich       2
Lisbon       2
Nuremberg   2
Düsseldorf  2
Name: City, dtype: int64
```

2. what is the Mean values for men and women ?

```
new_data_df.groupby(['Gender']).mean()
```

	Age	Yearly brutto salary (without bonus and stocks) in EUR	Annual_Salary
Gender			
Diverse	22.000000	159000.000000	98000.000000
Female	30.163636	59270.818182	53419.000000
Male	33.274869	77262.264398	68958.670157

3. what is the average salary of IT employ

```
average_salary=new_data_df.Annual_Salary.sum()/len(new_data_df.Annual_Salary)
average_salary
print(average_salary, "euro")
```

67073.64611872146 euro

4.show the graphical view of Age vs Salary.

```
new_data_df.Age.nunique()
```

30

```
age=new_data_df.Age.value_counts().head(10)  
age
```

```
30    35  
33    34  
34    33  
35    33  
32    32  
31    31  
28    31  
36    25  
29    24  
26    22
```

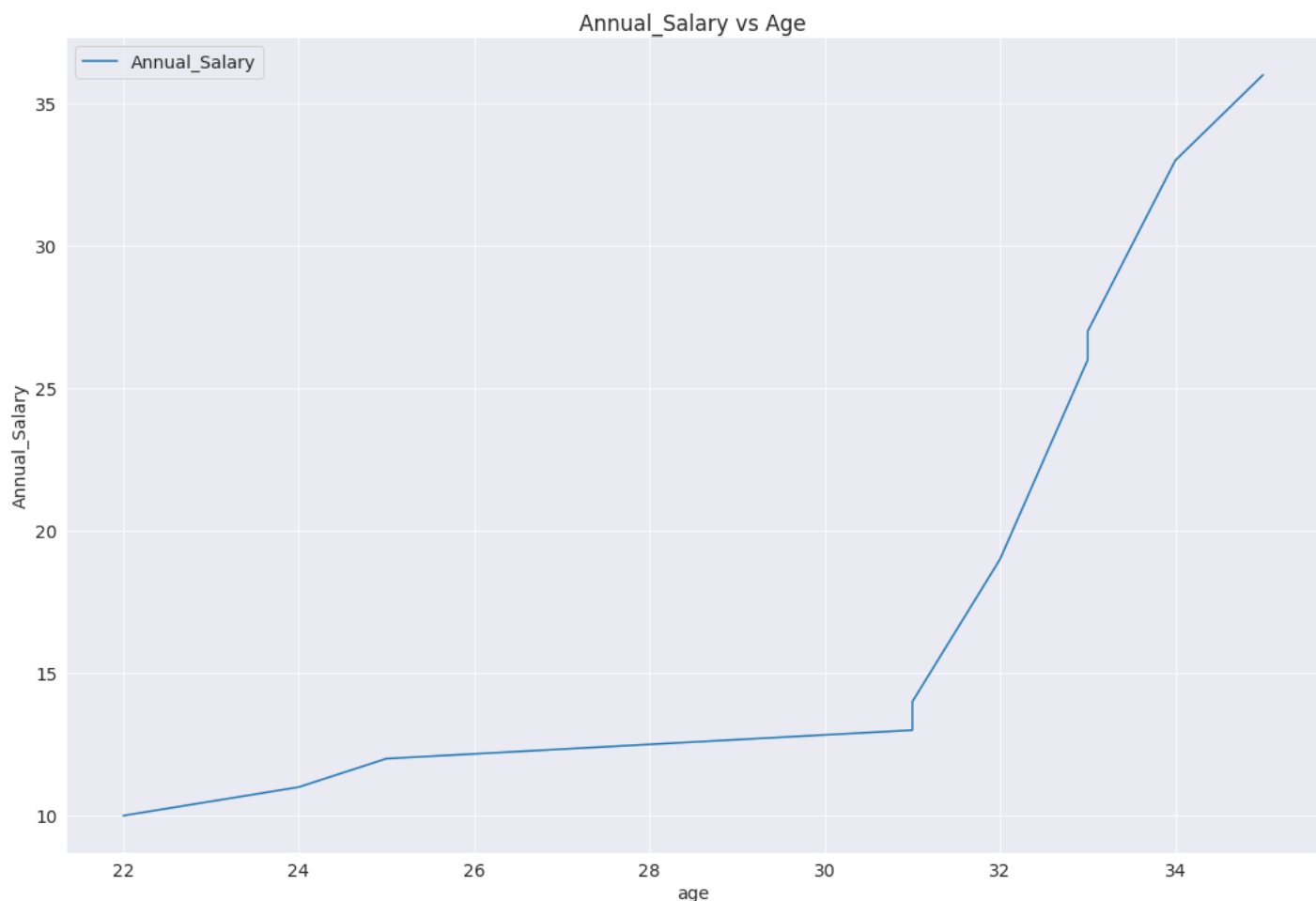
Name: Age, dtype: int64

```
salary=new_data_df.Annual_Salary.value_counts().head(10)  
salary
```

```
60000.0    36  
65000.0    33  
75000.0    27  
70000.0    26  
80000.0    19  
85000.0    14  
50000.0    13  
55000.0    12  
72000.0    11  
73000.0    10
```

Name: Annual_Salary, dtype: int64

```
plt.figure(figsize=(18,12))  
plt.plot(age,salary)  
  
plt.xlabel('age')  
plt.ylabel('Annual_Salary')  
  
plt.title("Annual_Salary vs Age")  
plt.legend(['Annual_Salary']);
```



5. What the mean values of annual_salary corresponding to experience?

```
new_data_df.groupby(['Experience_years']).mean().head(20)
```

	Age	Yearly brutto salary (without bonus and stocks) in EUR	Annual_Salary
Experience_years			
0	28.000000	31000.000000	22500.000000
0.8	24.000000	48000.000000	48000.000000
1	27.250000	58750.000000	48000.000000
1.5	25.000000	49850.000000	36000.000000
10	34.207547	80409.622642	73171.698113
11	33.875000	83031.250000	76000.000000
12	35.476190	80871.428571	76142.857143
13	34.600000	83421.000000	74999.000000
14	34.937500	88000.000000	75987.500000
15	36.320000	81140.000000	78080.000000
16	36.250000	80112.500000	77437.500000
17	37.666667	80000.000000	75083.333333
18	40.000000	197285.714286	83428.571429

	Age	Yearly brutto salary (without bonus and stocks) in EUR	Annual_Salary
Experience_years			
19	40.666667	76000.000000	73333.333333
2	29.315789	54447.368421	44552.631579
2,5	26.000000	40000.000000	33000.000000
2.5	32.000000	46666.666667	38166.666667
20	40.000000	95986.666667	85426.666667
21	42.000000	76000.000000	70500.000000
22	40.000000	72000.000000	45000.000000

Here the Zero in first represents Freshers

Let us save and upload our work to Jovian before continuing.

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "redraven46/eda-it-employ-in-europe" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/redraven46/eda-it-employ-in-europe
'https://jovian.ai/redraven46/eda-it-employ-in-europe'
```

Inferences and Conclusion

From The Analyze we can conclude that python and javascript programmers can get high salary in future. Also In europe Lisbon City have high annual salary. Berlin City have high number of companies. men have more work than women If you have high experience you will have high salary

```
import jovian
```

```
jovian.commit()
```

```
[jovian] Updating notebook "redraven46/eda-it-employ-in-europe" on https://jovian.ai
[jovian] Committed successfully! https://jovian.ai/redraven46/eda-it-employ-in-europe
'https://jovian.ai/redraven46/eda-it-employ-in-europe'
```

References and Future Work

Heart Attack Analysis From this analysis we can find how many peoples suffer from heart attack. So that we can find how to increase in production of pacemakers. So that in future who will have a pacemaker and other heart related startups have high reputation.

```
import jovian
```

```
jovian.commit(project="EDA-IT employ in Europe")
```