In this section of jupyter notebook, I will work on ds_salaries_df. Let's start the data wrangling with this data frame.

In [3]:
```
import pandas as pd
```

In [5]:
```
ds_salaries_df = pd.read_csv("/Users/ertuboston/Documents/Data_Science_Merrimack/DSE5002/PROJ
ds_salaries_df
```

Out[5]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	salar
0	0	2020	MI	FT	Data Scientist	70000	EUR	
1	1	2020	SE	FT	Machine Learning Scientist	260000	USD	
2	2	2020	SE	FT	Big Data Engineer	85000	GBP	
3	3	2020	MI	FT	Product Data Analyst	20000	USD	
4	4	2020	SE	FT	Machine Learning Engineer	150000	USD	
...	...	...	...	...	...	...	...	
602	602	2022	SE	FT	Data Engineer	154000	USD	
603	603	2022	SE	FT	Data Engineer	126000	USD	
604	604	2022	SE	FT	Data Analyst	129000	USD	
605	605	2022	SE	FT	Data Analyst	150000	USD	
606	606	2022	MI	FT	AI Scientist	200000	USD	

607 rows × 12 columns

In [7]:
```
### First, I would like to create a dataset that has only data scientist as job_title.

ds_salaries_df = ds_salaries_df[ds_salaries_df['job_title'].str.contains('Data Scientist', ca
ds_salaries_df

### total row went down to 159 rows.
```

Out[7]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	sal
0	0	2020	MI	FT	Data Scientist	70000	EUR	
6	6	2020	SE	FT	Lead Data Scientist	190000	USD	
7	7	2020	MI	FT	Data Scientist	11000000	HUF	
10	10	2020	EN	FT	Data Scientist	45000	EUR	
11	11	2020	MI	FT	Data Scientist	3000000	INR	
...	...	...	...	...	...	...	...	
592	592	2022	SE	FT	Data Scientist	230000	USD	
593	593	2022	SE	FT	Data Scientist	150000	USD	
596	596	2022	SE	FT	Data Scientist	210000	USD	
598	598	2022	MI	FT	Data Scientist	160000	USD	
599	599	2022	MI	FT	Data Scientist	130000	USD	

159 rows × 12 columns

In [9]:
```
### There are some columns that we won't need for the purpose of our research
### We can drop them to have a better look on dataabs

print(ds_salaries_df.columns)
```

```
Index(['Unnamed: 0', 'work_year', 'experience_level', 'employment_type',
       'job_title', 'salary', 'salary_currency', 'salary_in_usd',
       'employee_residence', 'remote_ratio', 'company_location',
       'company_size'],
      dtype='object')
```

In [11]:
```
ds_salaries_df.drop(columns = ['Unnamed: 0','remote_ratio','company_size','salary', 'salary_c
ds_salaries_df
```

```
/var/folders/yn/lfh7s3f52q18zdwkxgbxg58r0000gn/T/ipykernel_7295/2517256342.py:1: SettingWithCo
pyWarning:
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
  ds_salaries_df.drop(columns = ['Unnamed: 0','remote_ratio','company_size','salary', 'salary_
currency'], inplace=True)
```

Out[11]:

	work_year	experience_level	employment_type	job_title	salary_in_usd	employee_residence	compa
0	2020	MI	FT	Data Scientist	79833	DE	
6	2020	SE	FT	Lead Data Scientist	190000	US	
7	2020	MI	FT	Data Scientist	35735	HU	
10	2020	EN	FT	Data Scientist	51321	FR	
11	2020	MI	FT	Data Scientist	40481	IN	
...	...	...	...	...	...	...	
592	2022	SE	FT	Data Scientist	230000	US	
593	2022	SE	FT	Data Scientist	150000	US	
596	2022	SE	FT	Data Scientist	210000	US	
598	2022	MI	FT	Data Scientist	160000	US	
599	2022	MI	FT	Data Scientist	130000	US	

159 rows × 7 columns

In [13]:
```
### Now I will save this cleaned data as csv file.

ds_salaries_df.to_csv('clean_ds_salaries_df.csv', index = False)
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: