1 Projet : Enquête sur le salaire des Data Scientists en 2023

1.1 But du projet

La science des données est un domaine en plein essor, et les scientifiques des données jouent un rôle crucial dans l'analyse et l'interprétation de grands volumes de données. Cette profession étant de plus en plus demandée, il est important de comprendre les facteurs susceptibles d'influencer les salaires des Data Scientists. Cette analyse se concentre sur l'étude de ces facteurs et de leur impact sur les salaires.

L'objectif de cette étude est d'examiner les facteurs qui influencent les salaires des Data Scientists. Source : Kaggle

1.2 Résultats

Après l'analyse des données, On en déduit que le salaire d'un data scientist depend de 3 facteurs qui sont :

- La taille de l'entreprise : Plus l'entreprise est grande plus, le salaire est élevé.
- La localisation de l'entreprise : Le salaire le plus élevé dans ce dataset est de 30'400'000 CLP. Convertis en USD, 40'038 USD qui est très petit comparé au salaire le plus élevé obtenu aux États Unis : 412'000 USD.
- Le niveau d'expérience

[56]:
```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
```

1.3 Chargement du Dataset

[57]:
```
data = pd.read_csv('ds_salaries.csv')
```

1.4 Étape 1 : Pré-traitement des données

[58]:
```
data.head(6)
```

[58]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	100
1	2023	MI	CT	ML Engineer					
2	2023	MI	CT	ML Engineer					
3	2023	SE	FT	Data Scientist					
4	2023	SE	FT	Data Scientist					
5	2023	SE	FT	Applied Scientist					

```
1   30000             USD          30000                 US           100
2   25500             USD          25500                 US           100
3  175000             USD         175000                 CA           100
4  120000             USD         120000                 CA           100
5  222200             USD         222200                 US             0

  company_location company_size
0               ES            L
1               US            S
2               US            S
3               CA            M
4               CA            M
5               US            L
```

[59]:
```
data.columns
```

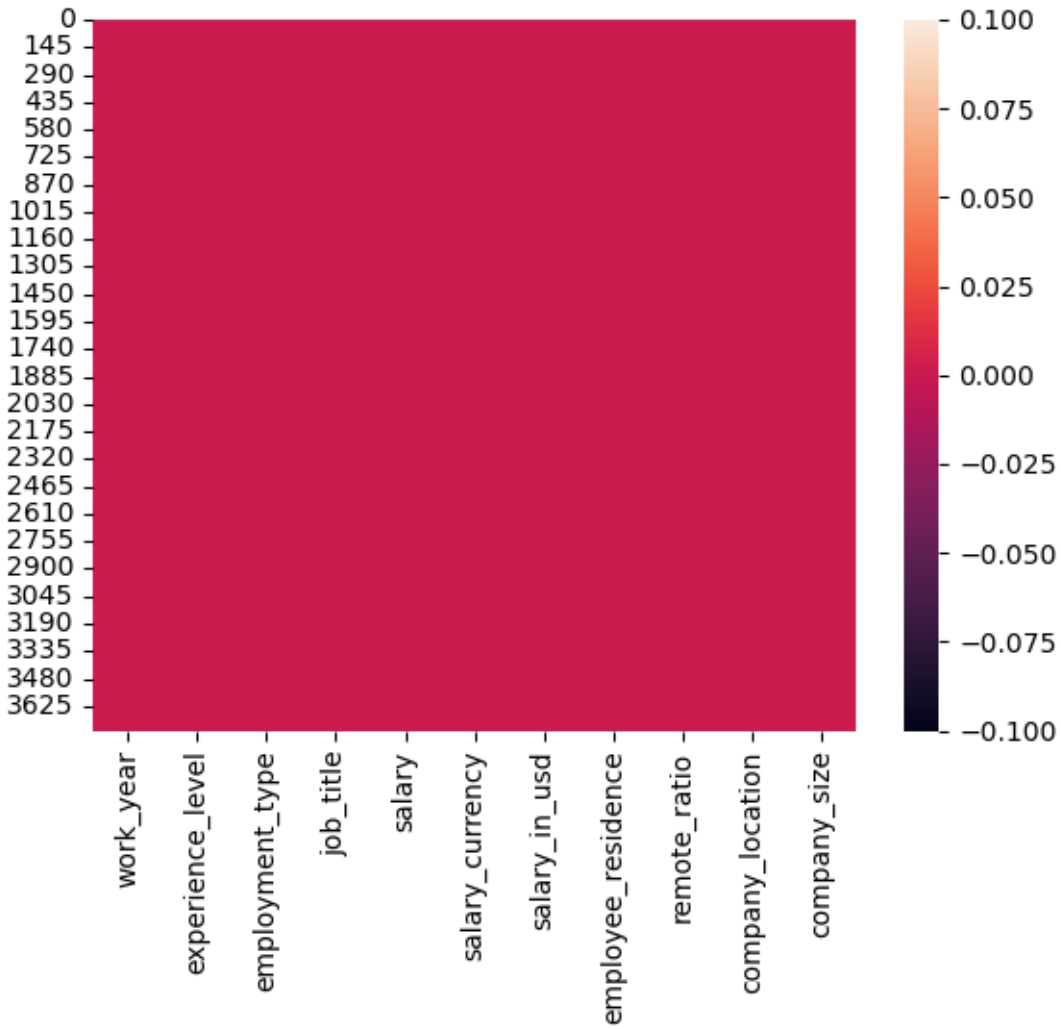
[59]:
```
Index(['work_year', 'experience_level', 'employment_type', 'job_title',
       'salary', 'salary_currency', 'salary_in_usd', 'employee_residence',
       'remote_ratio', 'company_location', 'company_size'],
      dtype='object')
```

[60]:
```
dimension = data.shape
print(dimension)
```

```
(3755, 11)
```

[61]:
```
sns.heatmap(data.isna())
```

[61]:
```
<Axes: >
```



[62]:
```
correlation = data.corr()
print(correlation)
```

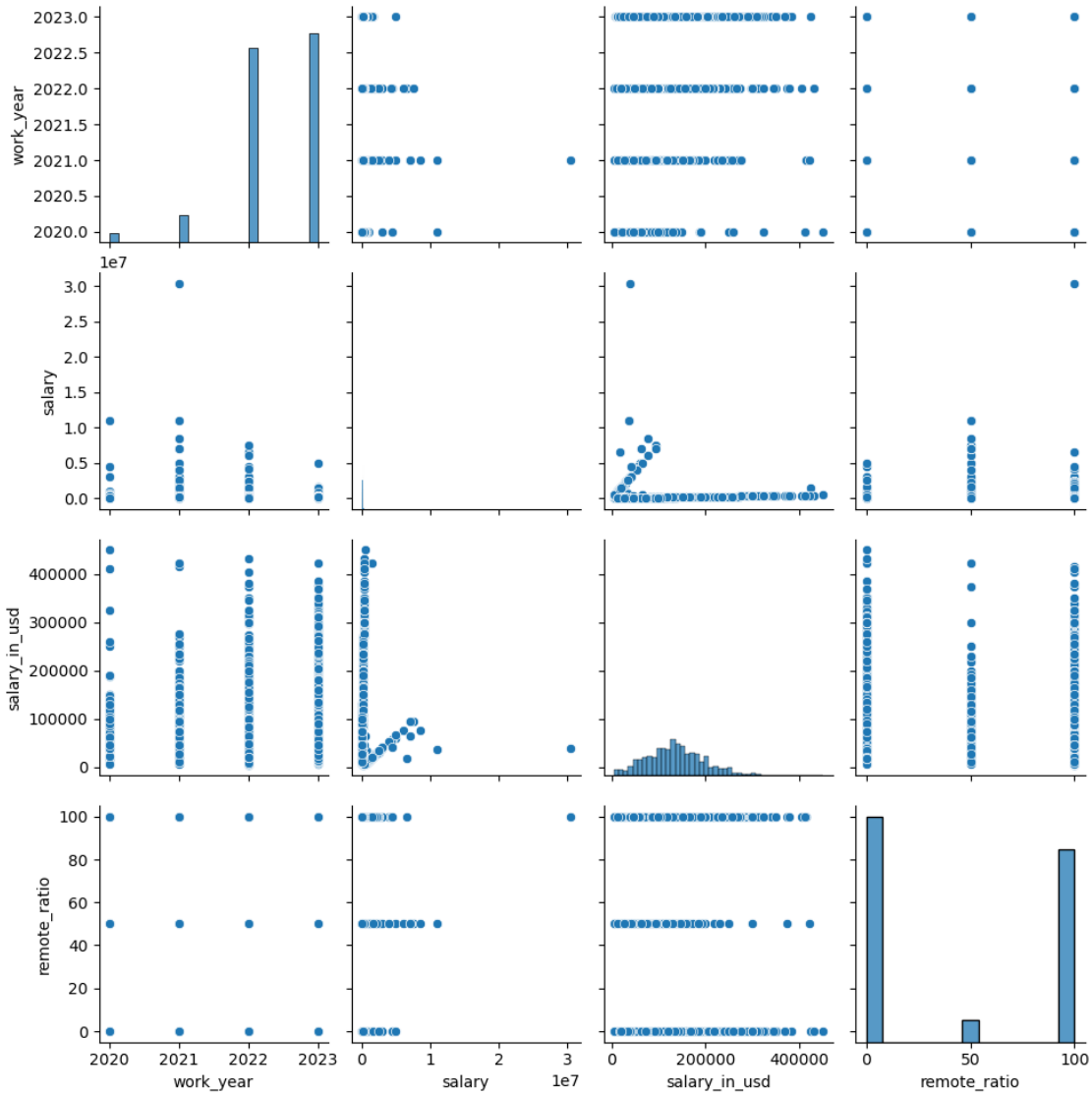
```
                work_year    salary  salary_in_usd  remote_ratio
work_year        1.000000 -0.094724       0.228290     -0.236430
salary          -0.094724  1.000000      -0.023676      0.028731
salary_in_usd    0.228290 -0.023676       1.000000     -0.064171
remote_ratio    -0.236430  0.028731      -0.064171      1.000000
```

```
C:\Users\konai\AppData\Local\Temp\ipykernel_12940\3497694653.py:1:
FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  correlation = data.corr()
```

[63]:
```
sns.pairplot(data)
```

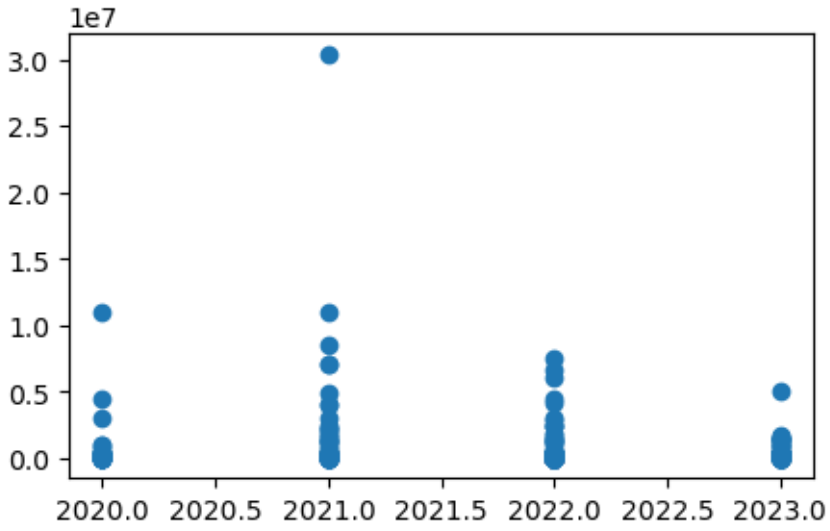
```
C:\Users\konai\anaconda3\lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

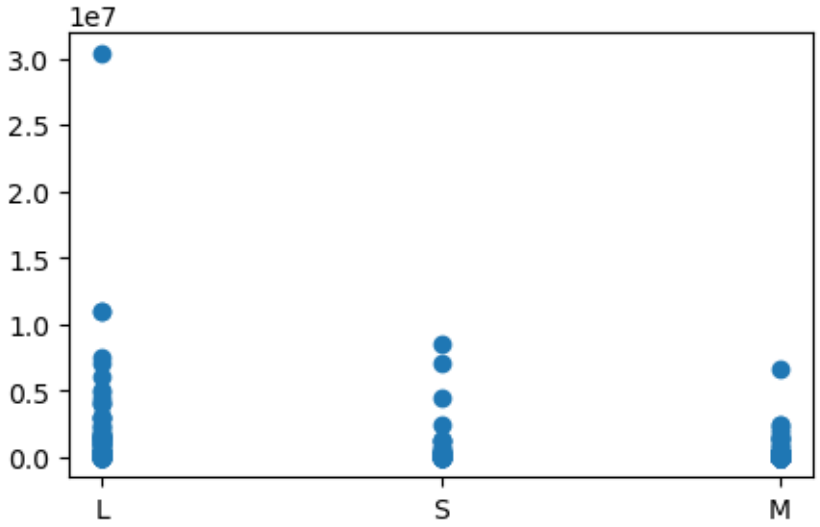
[63]: `<seaborn.axisgrid.PairGrid at 0x1a27c681cd0>`



[64]:
```
plt.figure(figsize=(5,3))
plt.subplot()
X = data['work_year']
Y = data['salary']
plt.scatter(X, Y)
plt.show()
plt.figure(figsize=(5,3))
```

```
plt.subplot()
X1 = data['company_size']
Y1 = data['salary']
plt.scatter(X1, Y1)
plt.show()
```





[65]:
```
df = data.copy()
```

[66]:
```
salary = df['salary']
salary.value_counts()
```

[66]:
```
100000    112
150000    100
120000     99
160000     85
130000     85
          ...
241871      1
93919       1
385000      1
225900      1
412000      1
Name: salary, Length: 815, dtype: int64
```

[67]:
```
job_title = df['job_title']
job_title.value_counts()
```

[67]:
```
Data Engineer                          1040
Data Scientist                          840
Data Analyst                            612
Machine Learning Engineer               289
Analytics Engineer                      103
                                       ...
Principal Machine Learning Engineer       1
Azure Data Engineer                       1
Manager Data Management                   1
Marketing Data Engineer                   1
Finance Data Analyst                      1
Name: job_title, Length: 93, dtype: int64
```

1.5 Étape 2 : Récupération des données des Data Scientists

[68]:
```
data_1 = df.where(job_title == 'Data Scientist')
data_1.head()
```

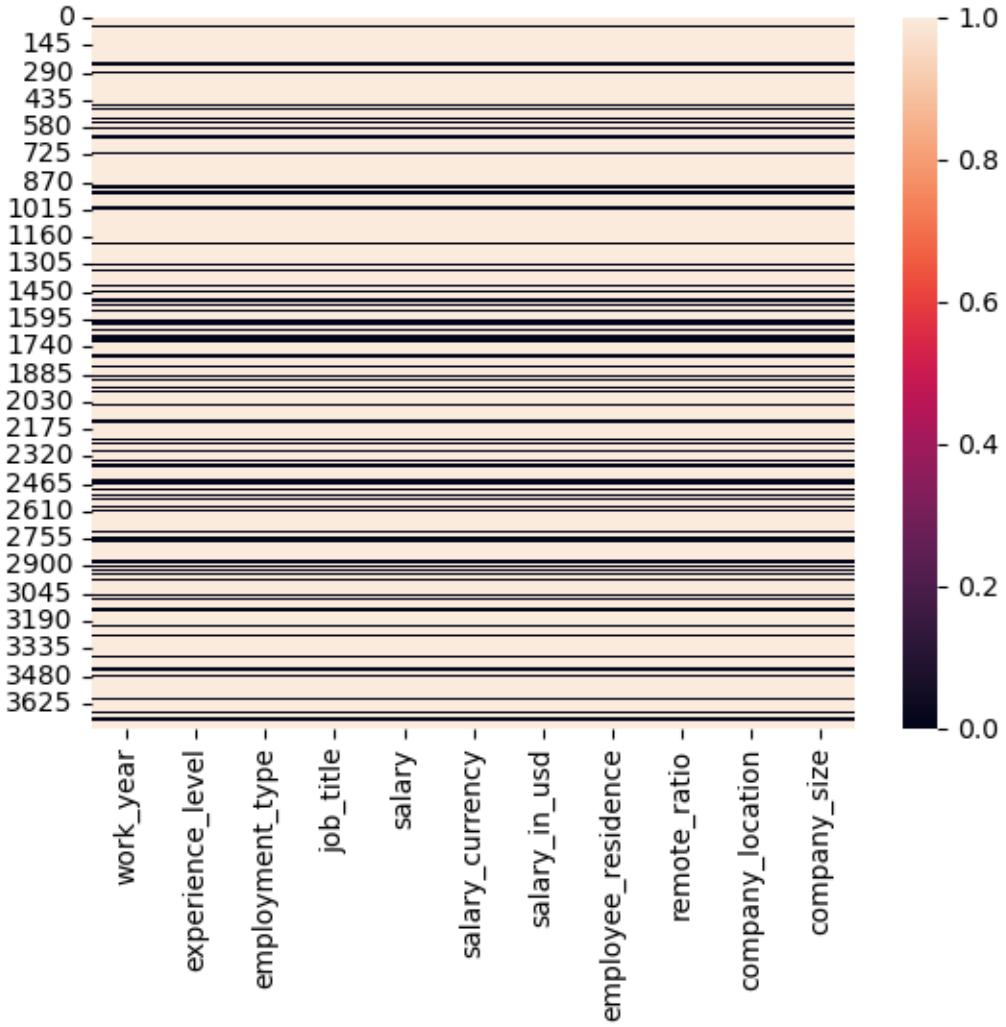
[68]:
```
   work_year experience_level employment_type       job_title    salary  \
0        NaN              NaN             NaN             NaN       NaN
1        NaN              NaN             NaN             NaN       NaN
2        NaN              NaN             NaN             NaN       NaN
3     2023.0               SE              FT  Data Scientist  175000.0
4     2023.0               SE              FT  Data Scientist  120000.0

  salary_currency  salary_in_usd employee_residence  remote_ratio  \
0             NaN            NaN                NaN           NaN
1             NaN            NaN                NaN           NaN
2             NaN            NaN                NaN           NaN
3             USD       175000.0                 CA         100.0
4             USD       120000.0                 CA         100.0
```

	company_location	company_size
0	NaN	NaN
1	NaN	NaN
2	NaN	NaN
3	CA	M
4	CA	M

1.5.1 Suppression des valeurs manquantes

[69]:
```
sns.heatmap(data_1.isna(), cbar = 'False')
```

[69]: <Axes: >



[70]:
```
data_1.dropna(axis=0, inplace = True)
data_1.reset_index(drop = True, inplace = True)
data_1.head()
```

[70]:

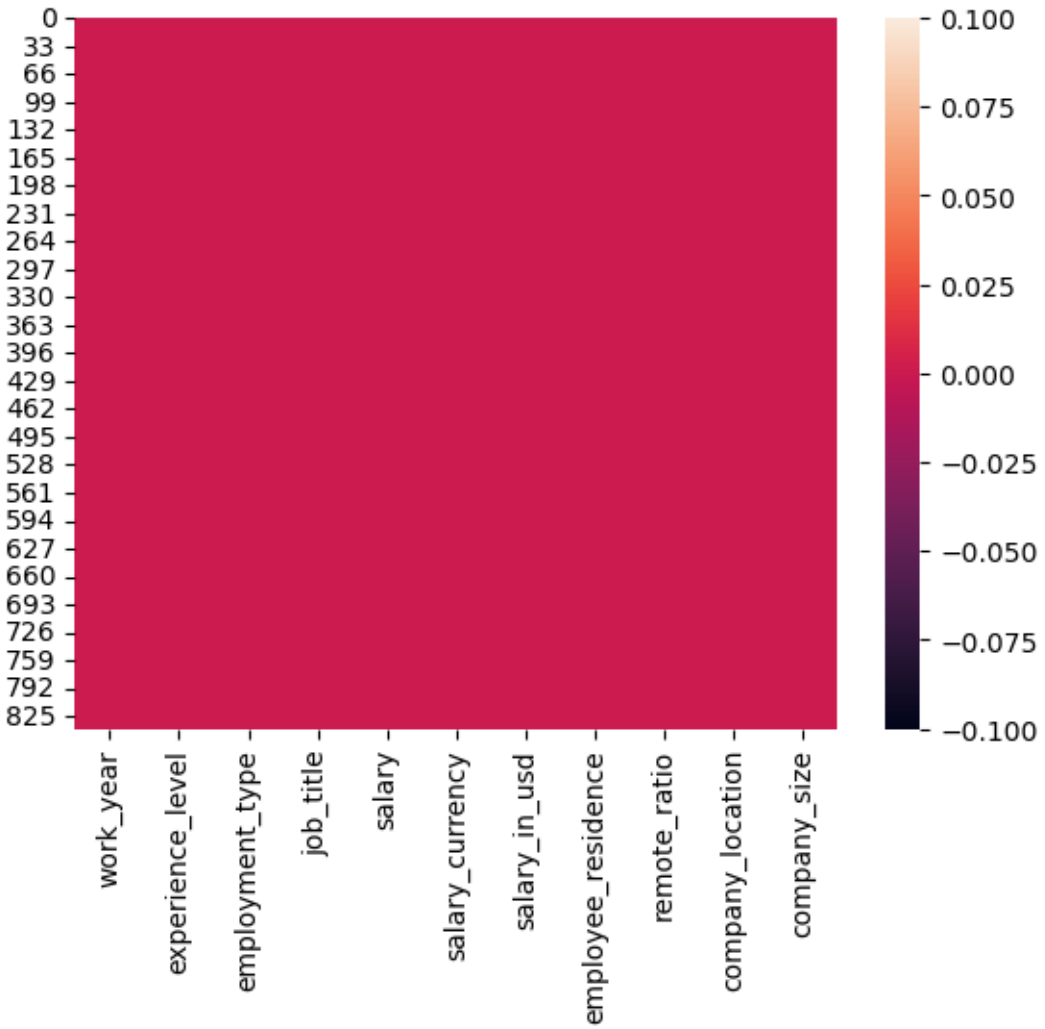
	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2023.0	SE	FT	Data Scientist	175000.0	USD	175000.0	CA	100.0	CA	M
1	2023.0	SE	FT	Data Scientist	120000.0	USD	120000.0	CA	100.0	CA	M
2	2023.0	SE	FT	Data Scientist	219000.0	USD	219000.0	CA	0.0	CA	M
3	2023.0	SE	FT	Data Scientist	141000.0	USD	141000.0	CA	0.0	CA	M
4	2023.0	SE	FT	Data Scientist	147100.0	USD	147100.0	US	0.0	US	M

[71]:
```
data_1.shape
```

[71]:
```
(840, 11)
```

[72]:
```
sns.heatmap(data_1.isna(), cbar = 'False')
```

[72]:
```
<Axes: >
```



[73]:
```
salary_dataScientist = data_1['salary']
salary_dataScientist_in_usd = data_1['salary_in_usd']
salary_dataScientist.value_counts()
```

[73]:
```
120000.0    25
140000.0    25
141525.0    22
191475.0    22
129300.0    18
            ..
249500.0     1
149850.0     1
182750.0     1
161500.0     1
412000.0     1
```

```
Name: salary, Length: 302, dtype: int64
```

[74]:
```
salary_dataScientist.max()
#salary_dataScientist_in_usd.max()
```

[74]:
```
30400000.0
```

[75]:
```
salary_dataScientist.min()
```

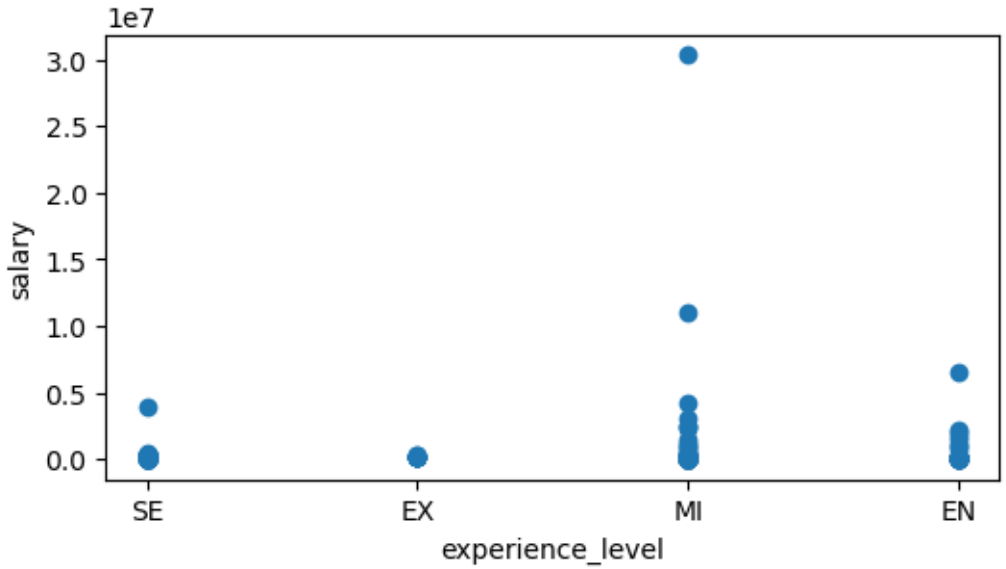
[75]:
```
10000.0
```

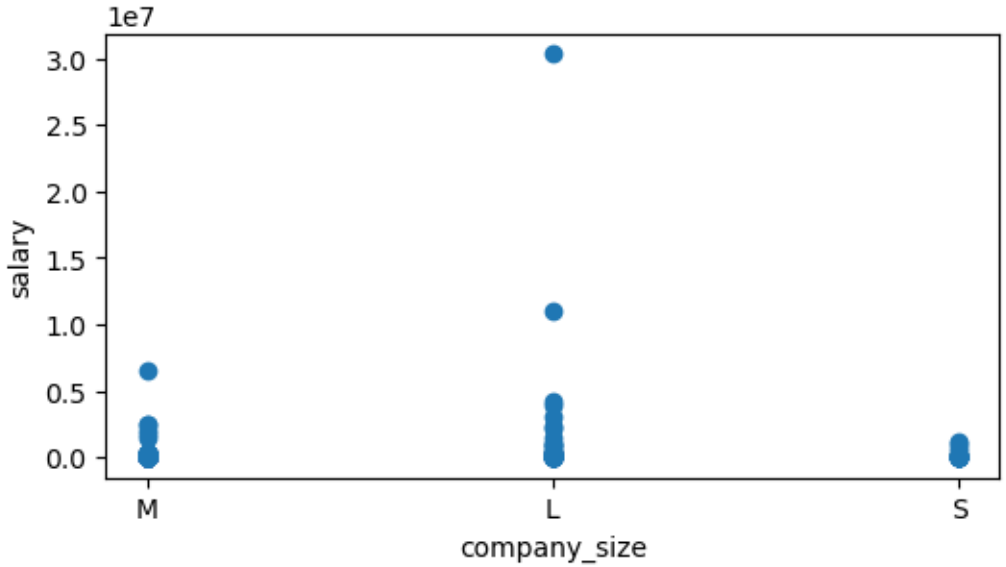
[76]:
```
salary_dataScientist.mean()
```

[76]:
```
239073.47619047618
```

[77]:
```
plt.figure()
plt.subplots(figsize = (6,3))
plt.xlabel('experience_level')
plt.ylabel('salary')
X = data_1['experience_level']
Y = data_1['salary']
plt.scatter(X, Y)
plt.show()
plt.subplots(figsize = (6,3))
plt.xlabel('company_size')
plt.ylabel('salary')
X1 = data_1['company_size']
Y1 = data_1['salary']
plt.scatter(X1, Y1)
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```





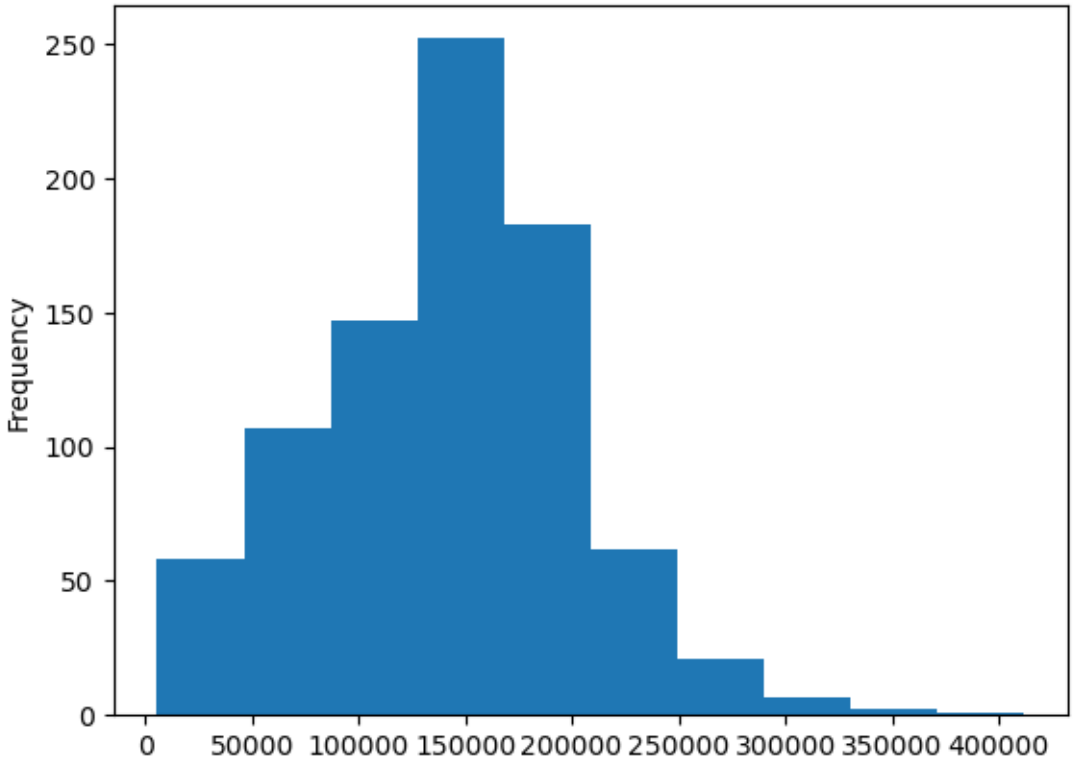
[78]:
```
sal_cur = data_1['salary_currency']
sal_cur.value_counts()
```

[78]:
```
USD    701
EUR     72
GBP     28
INR     18
CAD      7
```

```
CHF      2
BRL      2
HUF      2
AUD      2
HKD      1
THB      1
PLN      1
TRY      1
CLP      1
SGD      1
Name: salary_currency, dtype: int64
```

[79]:
```
salary_dataScientist_in_usd.plot.hist()
```

[79]:
```
<Axes: ylabel='Frequency'>
```



[80]:
```
salary_mean = data_1.groupby(['company_location']).mean()['salary']
salary_in_usd_mean = data_1.groupby(['company_location']).mean()['salary_in_usd']
```

```
C:\Users\konai\AppData\Local\Temp\ipykernel_12940\807864439.py:1: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
```

```
  salary_mean = data_1.groupby(['company_location']).mean()['salary']
C:\Users\konai\AppData\Local\Temp\ipykernel_12940\807864439.py:2: FutureWarning:
The default value of numeric_only in DataFrameGroupBy.mean is deprecated. In a
future version, numeric_only will default to False. Either specify numeric_only
or select only columns which should be valid for the function.
  salary_in_usd_mean =
data_1.groupby(['company_location']).mean()['salary_in_usd']
```

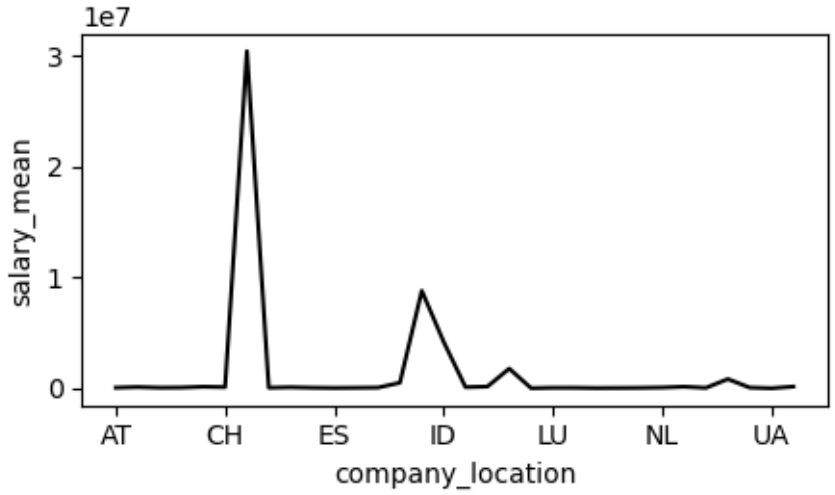
[81]:
```
print(salary_mean)
```

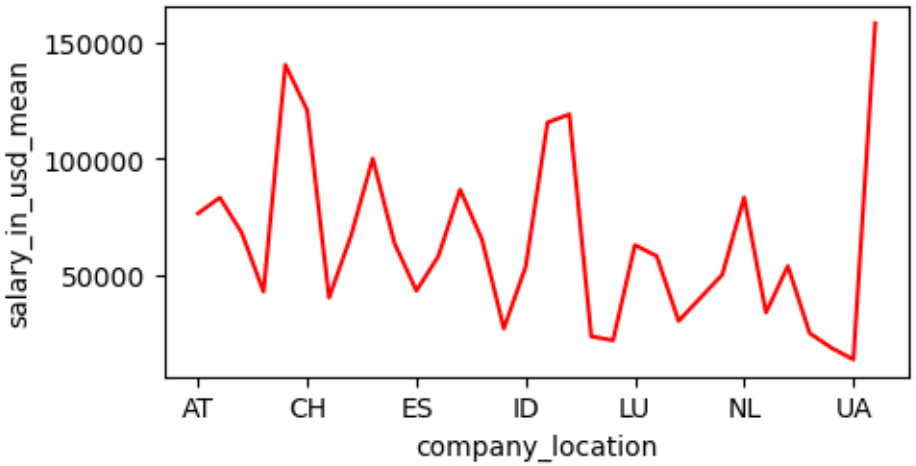
```
company_location
AT    6.600000e+04
AU    1.200000e+05
BE    6.475000e+04
BR    7.725000e+04
CA    1.472667e+05
CH    1.135000e+05
CL    3.040000e+07
DE    6.023714e+04
DZ    1.000000e+05
EE    5.900000e+04
ES    4.023571e+04
FR    5.324800e+04
GB    7.223065e+04
HK    5.100000e+05
HU    8.800000e+06
ID    4.200000e+06
IE    1.125000e+05
IL    1.600000e+05
IN    1.784000e+06
IT    1.900000e+04
LU    5.500000e+04
LV    5.400000e+04
MX    3.000000e+04
MY    4.000000e+04
NG    5.000000e+04
NL    7.925000e+04
PL    1.500000e+05
RO    5.000000e+04
TH    8.400000e+05
TR    7.166667e+04
UA    1.340000e+04
US    1.625481e+05
Name: salary, dtype: float64
```

[82]:
```
salary_mean.plot(figsize = (5,2.5), color = 'k')
plt.ylabel('salary_mean')
plt.figure()
```

```
salary_in_usd_mean.plot(figsize = (5,2.5), color = 'r')
plt.ylabel('salary_in_usd_mean')
```

[82]: Text(0, 0.5, 'salary_in_usd_mean')





[83]:
```
print(salary_in_usd_mean)
```

```
company_location
AT      76352.000000
AU      83171.000000
BE      68030.500000
BR      42605.750000
CA     140403.619048
```

```
CH    120747.500000
CL     40038.000000
DE     66623.857143
DZ    100000.000000
EE     63312.000000
ES     43058.821429
FR     57838.333333
GB     86613.290323
HK     65062.000000
HU     26709.500000
ID     53416.000000
IE    115514.750000
IL    119059.000000
IN     23367.733333
IT     21669.000000
LU     62726.000000
LV     57946.500000
MX     30000.000000
MY     40000.000000
NG     50000.000000
NL     83264.750000
PL     33609.000000
RO     53654.000000
TH     24740.000000
TR     18390.333333
UA     13400.000000
US    158283.875371
Name: salary_in_usd, dtype: float64
```

[84]:
```
salary_in_usd_mean.mean()
```

[84]:
```
63737.722291017126
```

[85]:
```
location = data_1['company_location']
dt = data_1.where(location == 'CL')
dt.dropna(axis = 0, inplace = True)
dt.reset_index(drop = True, inplace = True)
```

[86]:
```
dt
```

[86]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	company_location	company_size
0	2021.0	MI	FT	Data Scientist	30400000.0	CLP	40038.0	CL	100.0	CL	L