In [29]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing,svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [30]:

```
ds=pd.read_csv(r"C:\Users\pucha\Downloads\ds_salaries.csv")
ds
```

Out[30]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	s
0	2023	SE	FT	Principal Data Scientist	80000	EUR	
1	2023	MI	CT	ML Engineer	30000	USD	
2	2023	MI	CT	ML Engineer	25500	USD	
3	2023	SE	FT	Data Scientist	175000	USD	
4	2023	SE	FT	Data Scientist	120000	USD	
...	...	...	...	...	...	...	
3750	2020	SE	FT	Data Scientist	412000	USD	
3751	2021	MI	FT	Principal Data Scientist	151000	USD	
3752	2020	EN	FT	Data Scientist	105000	USD	
3753	2020	EN	CT	Business Data Analyst	100000	USD	
3754	2021	SE	FT	Data Science Manager	7000000	INR	

3755 rows × 11 columns

In [31]:

```
ds=ds[['salary','work_year']]
ds.columns=['sl','wy']
ds.head(18)
```

Out[31]:

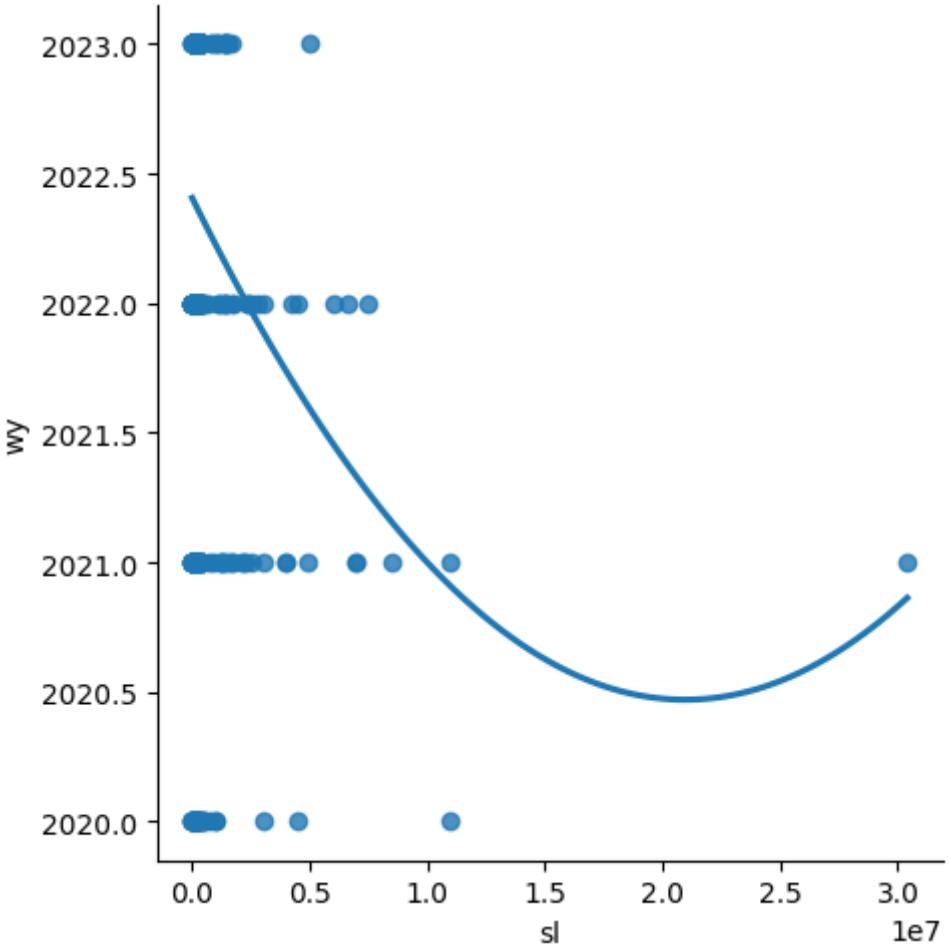
	sl	wy
0	80000	2023
1	30000	2023
2	25500	2023
3	175000	2023
4	120000	2023
5	222200	2023
6	136000	2023
7	219000	2023
8	141000	2023
9	147100	2023
10	90700	2023
11	130000	2023
12	100000	2023
13	213660	2023
14	130760	2023
15	147100	2023
16	90700	2023
17	170000	2023

In [32]:

```
sns.lmplot(x="sl",y="wy",data=ds,order=2,ci=None)
```

Out[32]:

```
<seaborn.axisgrid.FacetGrid at 0x2b588805210>
```



In [33]:

```
ds.describe()
```

Out[33]:

	sl	wy
count	3.755000e+03	3755.000000
mean	1.906956e+05	2022.373635
std	6.716765e+05	0.691448
min	6.000000e+03	2020.000000
25%	1.000000e+05	2022.000000
50%	1.380000e+05	2022.000000
75%	1.800000e+05	2023.000000
max	3.040000e+07	2023.000000

In [34]:

```
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3755 entries, 0 to 3754
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   sl      3755 non-null   int64
 1   wy      3755 non-null   int64
dtypes: int64(2)
memory usage: 58.8 KB
```

In [35]:

```
ds.fillna(method="ffill",inplace=True)
```

```
C:\Users\pucha\AppData\Local\Temp\ipykernel_2108\2683886818.py:1: Setti
ngWithCopyWarning:
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 (ht
tps://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy)
  ds.fillna(method="ffill",inplace=True)
```

In [36]:

```
x=np.array(ds['sl']).reshape(-1,1)
y=np.array(ds['wy']).reshape(-1,1)
ds.dropna(inplace=True)
```

```
C:\Users\pucha\AppData\Local\Temp\ipykernel_2108\3958565703.py:3: Setti
ngWithCopyWarning:
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 (ht
tps://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy)
  ds.dropna(inplace=True)
```

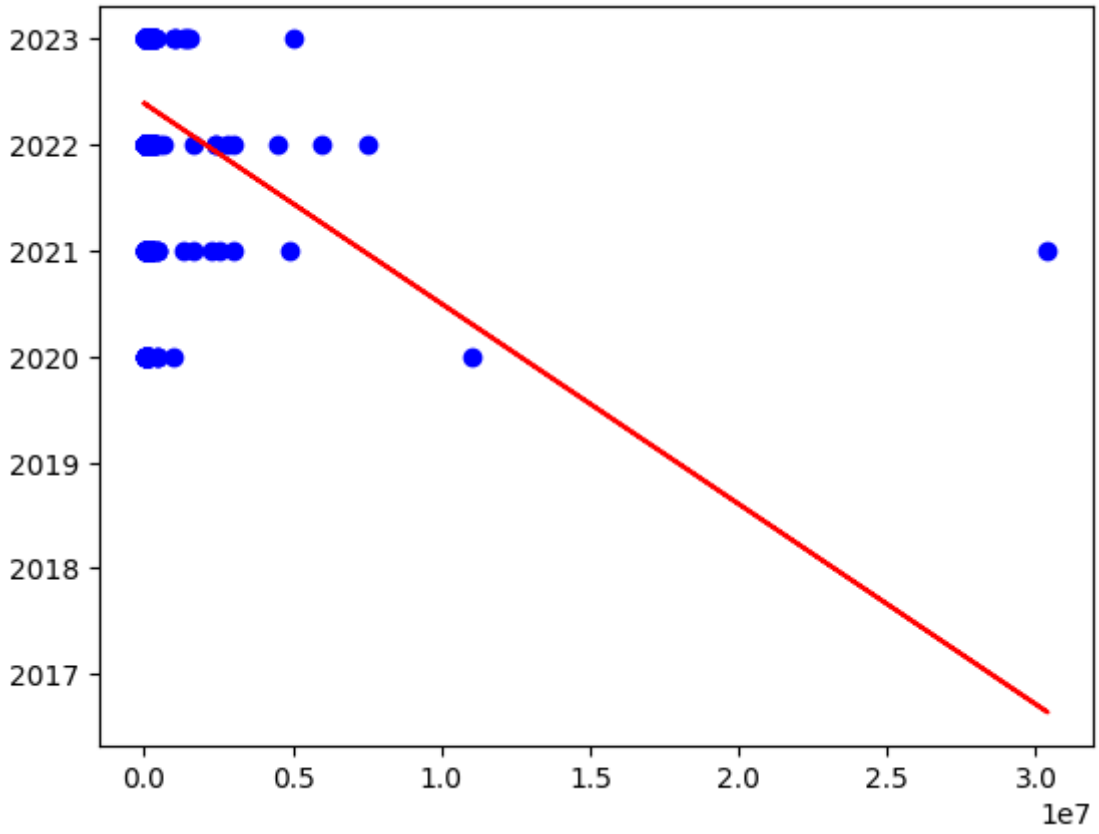
In [37]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.5)
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

```
-0.019027341762368977
```

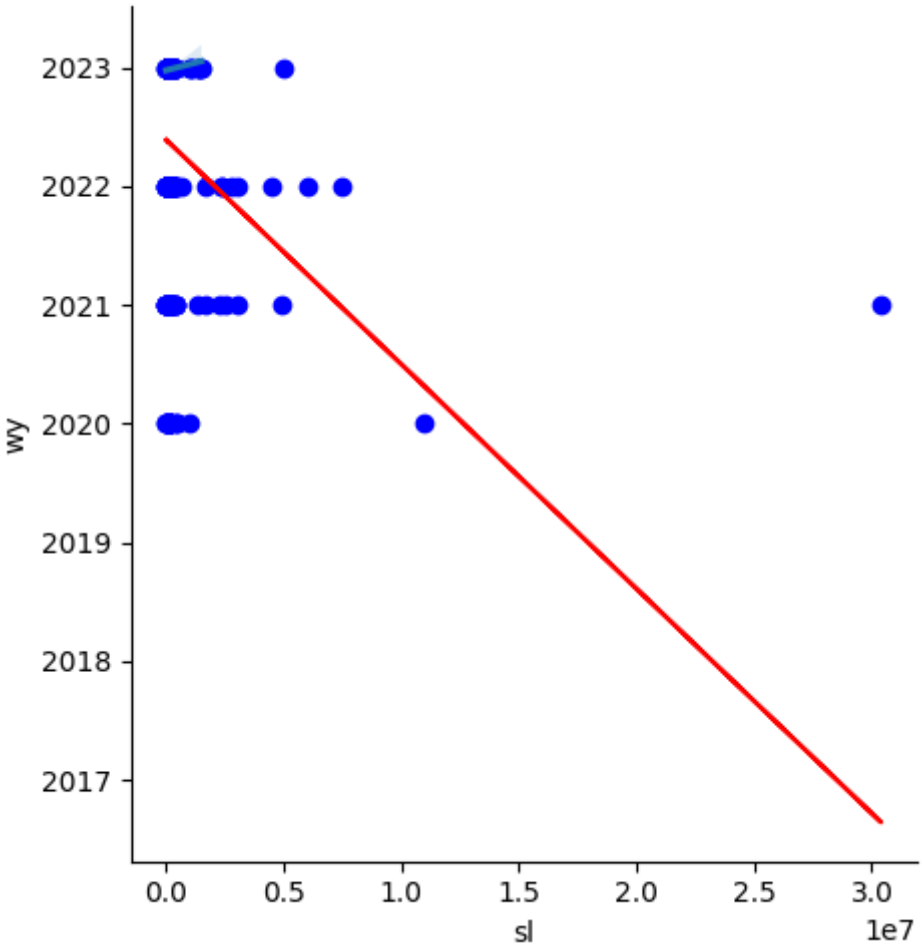
In [38]:

```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='r')
plt.show()
```



In [45]:

```
ds1999=ds[555:999][:]
sns.lmplot(x="sl",y="wy",data=ds1999,order=1)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='r')
plt.show()
```



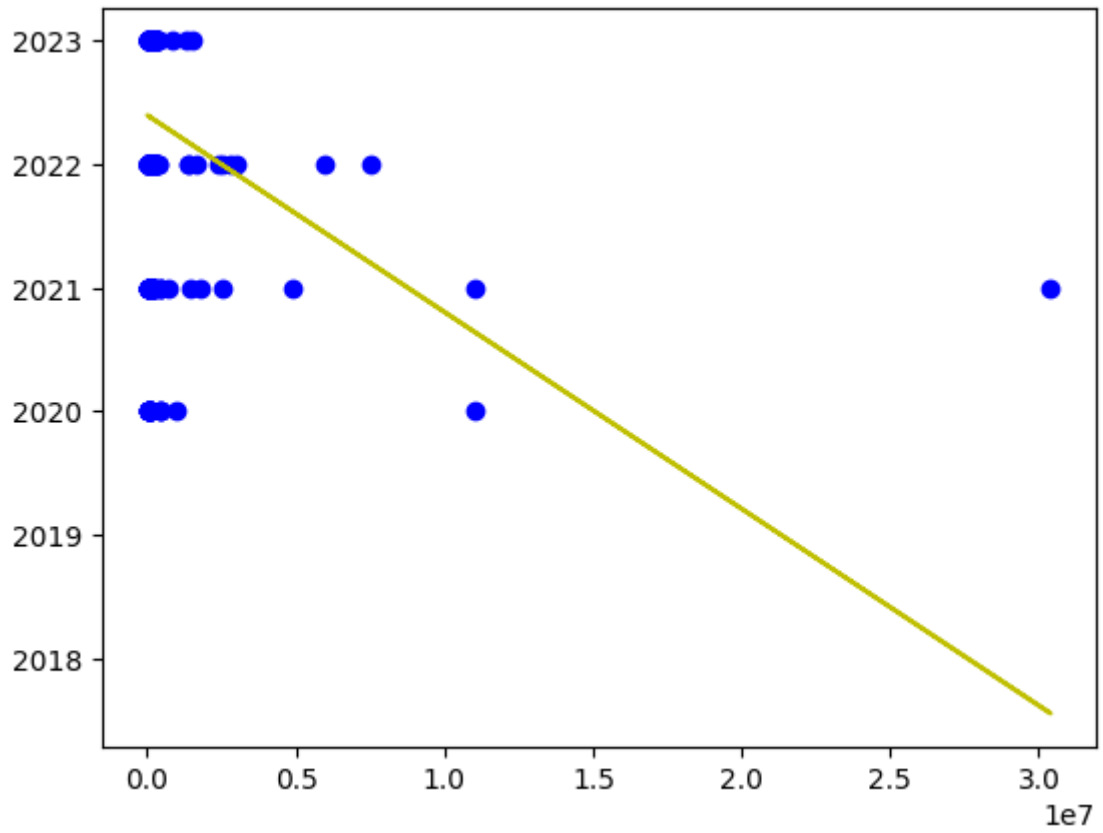
In [46]:

```
ds1999.fillna(method='ffill',inplace=True)
x=np.array(ds['sl']).reshape(-1,1)
y=np.array(ds['wy']).reshape(-1,1)
ds.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression()
regr.fit(x_train,y_train)
print("Regression:",regr.score(x_test,y_test))
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='y')
plt.show()
```

```
C:\Users\pucha\AppData\Local\Temp\ipykernel_2108\1259858354.py:4: Setti
ngWithCopyWarning:
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 (ht
tps://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#ret
urning-a-view-versus-a-copy)
  ds.dropna(inplace=True)

Regression: -0.0010257775883855125
```



In [47]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2_score:",r2)
```

```
R2_score: -0.0010257775883855125
```

In [ ]:

```

```