ex4(ids)))

February 19, 2024

[ ]:
```
EX NO: 4 - Exploratory Data Analysis
Date:12-02-2024
HARIHARAN K - URK22AI1048
```

[ ]:
```
Aim

To demonstrate the exploratory data analysis using python for data science
applications

Description

Exploratory Data Analysis is a crucial step before you jump to machine learning␣
 ↪or
modeling of dataOnce Exploratory Data Analysis is complete and insights are␣
 ↪drawn, its
feature can be used for supervised and unsupervised machine learning modeling.

Description of data
#Load the required libraries
import pandas as pd
import numpy as np
import matplotlib. pyplot as plt
#Load the Data
df = pd.read_csv(&#39;filename.csv&#39;)
#View the data
df.head()
#Basic information
df.info()
#Describe the data
df.describe()
#Find the duplicates
df.duplicated().sum()
# Unique values
df [&#39;column name&#39;]. unique()

Handling missing data
```

```
There are several options for handling missing values.
#Find null values
df.isnull().sum()
#Replace null values
df.replace(np.nan,&#39;0&#39;,inplace = True)
#Check the changes now

df.isnull().sum()

Handling outliers

An outlier is something which is separate or different from the crowd. Outliers␣
 ↪can be
a result of a mistake during data collection or it can be just an indication of␣
 ↪variance in your
data. Some of the methods for detecting and handling outliers are Box plot,␣
 ↪Scatter Plot, Z-
Score and IQR (Inter-Quartile Range)

Box Plot

The whiskers extend from the edges of the box to show the range of the data.␣
 ↪Outlier points
are those past the end of the whiskers. Boxplots show robust measures of␣
 ↪location and spread
as well as providing information about symmetry and outliers.
df [[&#39;Column name&#39;]]. boxplot ()

Scatterplot

The data are displayed as a collection of points, each having the value of one␣
 ↪variable
determining the position on the horizontal axis and the value of the other␣
 ↪variable determining
the position on the vertical axis.

# Scatter plot between two columns
plt.scatter(df[&#39;column 1&#39;], df[&#39;column 2&#39;])
plt.show()
Z-score:
The Z-score is the signed number of standard deviations by which the value of an
observation or data point is above the mean value of what is being observed or␣
 ↪measured.
df[&#39;Income zscore&#39;] = stats.zscore(df[&#39;Income&#39;])
print(df.head())
```

```
IQR:
The interquartile range (IQR) is a measure of statistical dispersion, being␣
 ↪equal to the
difference between 75th and 25th percentiles, or between upper and lower␣
 ↪quartiles.IQR = Q3
− Q1.
#calculate interquartile range of values in single column
q75, q25 = np.percentile(df[&#39;column name'], [75 ,25])
iqr = q75 - q25
#display interquartile range

iqr
#define function to calculate interquartile range
def find_iqr(x):
return np.subtract(*np.percentile(x, [75, 25]))
#Calculate IQR for more than one column

df[[&#39;column 1&#39;, &#39;column 2&#39;]].apply(find_iqr)
#Calculate IQR for all columns
df.apply(find_iqr)
# Removing Outlier from Data Frame
z-score and IQR are used to remove the outliers form the dataset
df = df[~((df &lt; (Q1−1.5 * IQR)) |(df &gt; (Q3 + 1.5 * IQR))).any(axis=1)]
df=df[(z&lt;3).all(axis=1)]

Understanding relationships and new insights through plots
The common data visualizations techniques used for understanding the data set␣
 ↪relationships
are histogram and Heat Maps

Histogram

A histogram is a great tool for quickly assessing a probability distribution␣
 ↪that is easy for
interpretation by almost any audience.
df.hist()

Heat Map

The Heat Map procedure shows the distribution of a quantitative variable over␣
 ↪all
combinations of 2 categorical factors. The correlation between two random␣
 ↪variables is a
number that runs from -1 through 0 to +1 and indicates a strong inverse␣
 ↪relationship, no
relationship, and a strong direct relationship, respectively.
```

```
# generating pairwise correlation
corr = df.corr()
# Displaying dataframe as an heatmap with diverging colourmap as coolwarm
corr.style.background_gradient(cmap =&#39;coolwarm&#39;)
```

[5]:
```
import numpy as np
from scipy.stats import zscore as stats
import pandas as pd
import matplotlib.pyplot as plt
df=pd.read_csv("Salary.csv")
df
```

[5]:

	job_title	experience_level	employment_type	work_models	work_year	employee_residence	salary	salary_currency	salary_in_usd	company_location	company_size
0	Data Engineer	Mid-level	Full-time	Remote	2024	United States	148100	USD	148100	United States	Medium
1	Data Engineer	Mid-level	Full-time	Remote	2024	United States	98700	USD	98700	United States	Medium
2	Data Scientist	Senior-level	Full-time	Remote	2024	United States	140032	USD	140032	United States	Medium
3	Data Scientist	Senior-level	Full-time	Remote	2024	United States	100022	USD	100022	United States	Medium
4	BI Developer	Mid-level	Full-time	On-site	2024	United States	120000	USD	120000	United States	Medium
..	…	…	…	…	…	…	…	…	…	…	…
993	Data Engineer	Mid-level	Full-time	On-site	2023	United States	133000	USD	133000	United States	Medium
994	Data Engineer	Mid-level	Full-time	On-site	2023	United States	58400	USD	58400	United States	Medium
995	Data Manager	Senior-level	Full-time	Remote	2023	United States	144300	USD	144300		
996	Data Manager	Senior-level	Full-time	Remote	2023	United States	104800	USD	104800		
997	Data Analytics Lead	Senior-level	Full-time	On-site	2023	United States	115920	USD	115920		

```
995    United States       Medium
996    United States       Medium
997    United States       Medium

[998 rows x 11 columns]
```

[2]:
```
#1.Remove the columns that has null value form data_science_salaries.
#URK22AI1048
df.dropna(axis=1,inplace=False)
```

[2]:
```
                    job_title experience_level employment_type work_models  \
0                Data Engineer        Mid-level       Full-time      Remote
1                Data Engineer        Mid-level       Full-time      Remote
2               Data Scientist     Senior-level       Full-time      Remote
3               Data Scientist     Senior-level       Full-time      Remote
4                 BI Developer        Mid-level       Full-time     On-site
…                           …                 …               …           …
11082       Staff Data Analyst      Entry-level        Contract      Hybrid
11083       Staff Data Analyst  Executive-level       Full-time     On-site
11084  Machine Learning Manager     Senior-level       Full-time      Hybrid
11085            Data Engineer        Mid-level       Full-time      Hybrid
11086           Data Scientist     Senior-level       Full-time     On-site

      company_size
0           Medium
1           Medium
2           Medium
3           Medium
4           Medium
…            …
11082        Large
11083       Medium
11084        Large
11085        Large
11086        Small

[11087 rows x 5 columns]
```

[10]:
```
#2.Remove the rows between 5000 to 8000 when they have any null value.
#URK22AI1048
s=df.iloc[5000:6001,:]
s1=s.dropna()
s1
```

[10]:
```
                    job_title experience_level employment_type work_models  \
5000         Applied Scientist     Senior-level       Full-time     On-site
5001         Applied Scientist     Senior-level       Full-time     On-site
```

```
5002              Data Scientist     Senior-level       Full-time      Remote
5003              Data Scientist     Senior-level       Full-time      Remote
5004               Data Engineer     Senior-level       Full-time      Remote
…                            …              …               …          …
5996  Machine Learning Engineer     Senior-level       Full-time     On-site
5997  Machine Learning Engineer     Senior-level       Full-time     On-site
5998              Data Scientist     Senior-level       Full-time     On-site
5999              Data Scientist     Senior-level       Full-time     On-site
6000               Data Engineer  Executive-level       Full-time     On-site

      work_year employee_residence    salary salary_currency  salary_in_usd  \
5000     2023.0      United States  222200.0             USD       222200.0
5001     2023.0      United States  136000.0             USD       136000.0
5002     2023.0      United States  161000.0             USD       161000.0
5003     2023.0      United States  151000.0             USD       151000.0
5004     2023.0      United States  136994.0             USD       136994.0
…         …                  …       …               …             …
5996     2023.0      United States  240000.0             USD       240000.0
5997     2023.0      United States  180000.0             USD       180000.0
5998     2023.0      United States  270000.0             USD       270000.0
5999     2023.0      United States  220000.0             USD       220000.0
6000     2023.0      United States  194500.0             USD       194500.0

     company_location company_size
5000    United States        Large
5001    United States        Large
5002    United States       Medium
5003    United States       Medium
5004    United States       Medium
…                 …            …
5996    United States       Medium
5997    United States       Medium
5998    United States       Medium
5999    United States       Medium
6000    United States       Medium

[1001 rows x 11 columns]
```

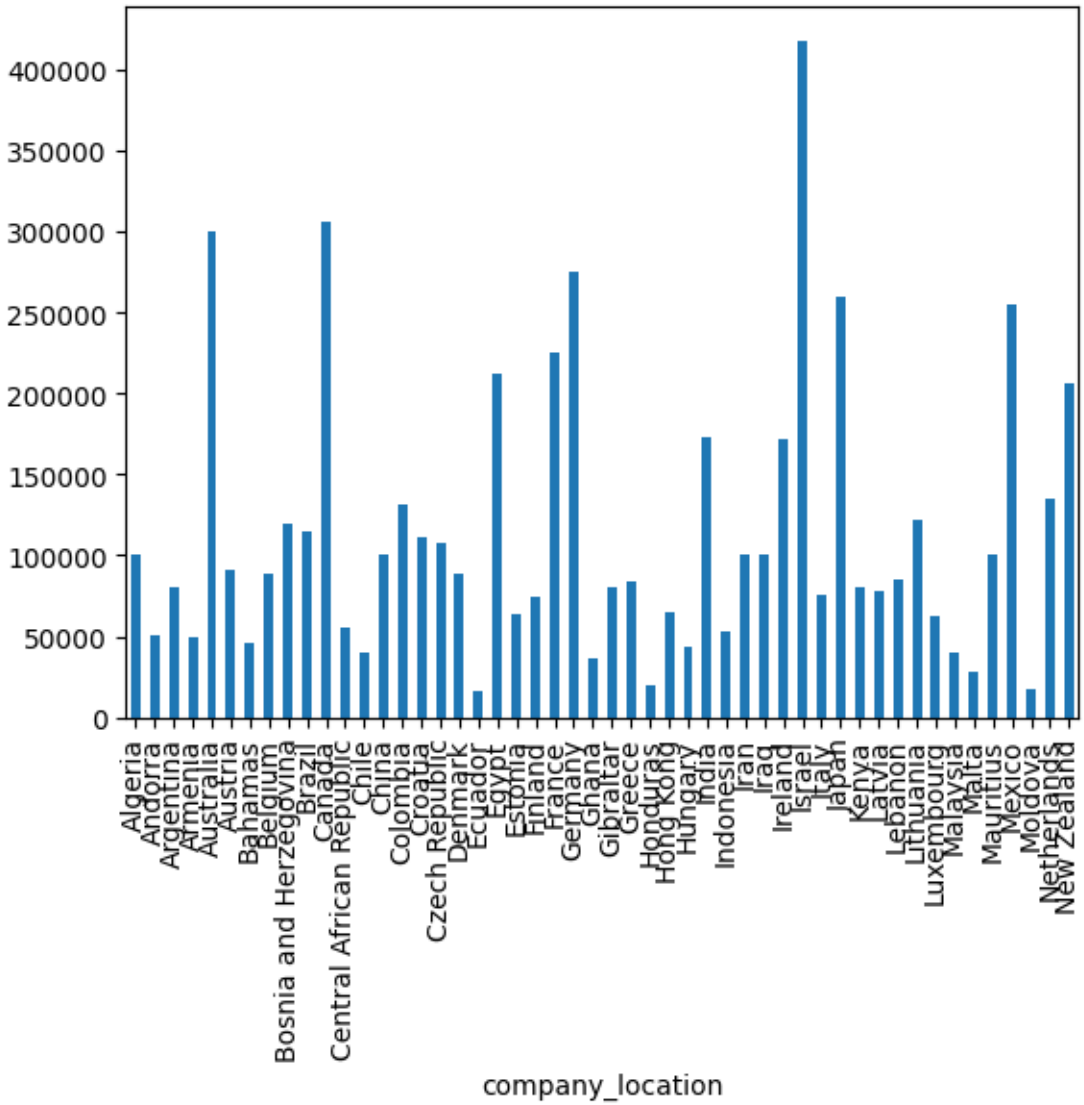
[15]:
```
#3. Find and remove the duplicate rows.
#URK22AI1048

#S = df[df.duplicated()]----It display duplicated rows
S1 = df.drop_duplicates()
S1
```

[15]:
```
                    job_title experience_level employment_type work_models  \
0              Data Engineer        Mid-level       Full-time      Remote
```

```
1                 Data Engineer        Mid-level       Full-time      Remote
2               Data Scientist     Senior-level       Full-time      Remote
3               Data Scientist     Senior-level       Full-time      Remote
4                 BI Developer        Mid-level       Full-time     On-site
…                           …              …              …         …
11082       Staff Data Analyst      Entry-level        Contract      Hybrid
11083       Staff Data Analyst  Executive-level       Full-time     On-site
11084  Machine Learning Manager     Senior-level       Full-time      Hybrid
11085             Data Engineer        Mid-level       Full-time      Hybrid
11086           Data Scientist     Senior-level       Full-time     On-site

       work_year employee_residence    salary salary_currency  salary_in_usd  \
0         2024.0      United States  148100.0             USD       148100.0
1         2024.0      United States   98700.0             USD        98700.0
2         2024.0      United States  140032.0             USD       140032.0
3         2024.0      United States  100022.0             USD       100022.0
4         2024.0      United States  120000.0             USD       120000.0
…          …                 …       …               …            …
11082     2020.0             Canada   60000.0             CAD        44753.0
11083     2020.0            Nigeria   15000.0             USD        15000.0
11084     2020.0             Canada  157000.0             CAD       117104.0
11085     2020.0            Austria   65000.0             EUR        74130.0
11086     2020.0            Austria   80000.0             EUR        91237.0

      company_location company_size
0        United States       Medium
1        United States       Medium
2        United States       Medium
3        United States       Medium
4        United States       Medium
…                  …           …
11082           Canada        Large
11083           Canada       Medium
11084           Canada        Large
11085          Austria        Large
11086          Austria        Small

[6601 rows x 11 columns]
```

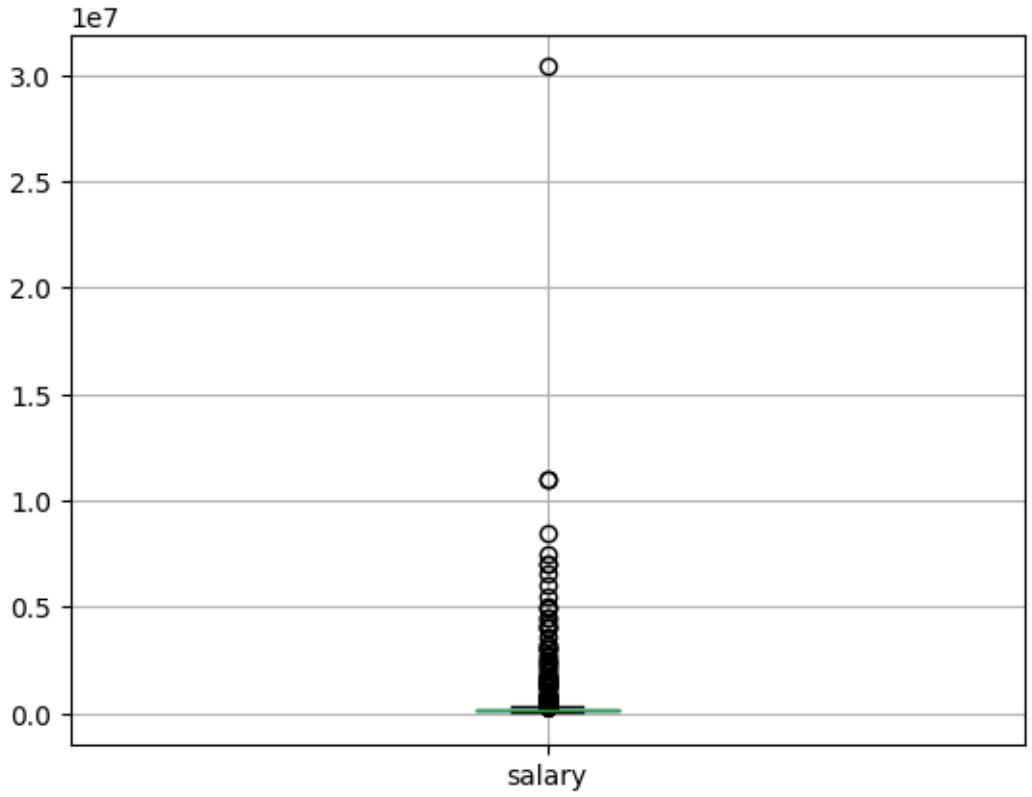
[25]:
```
#4.Draw the bar chart for the max 'salary_in_usd' of each country_location to␣
 ↪detect the top paid country.
#URK22AI1048

s = df.groupby('company_location')['salary_in_usd'].max().head(50)
s.plot.bar()
```

[25]:
```
<Axes: xlabel='company_location'>
```



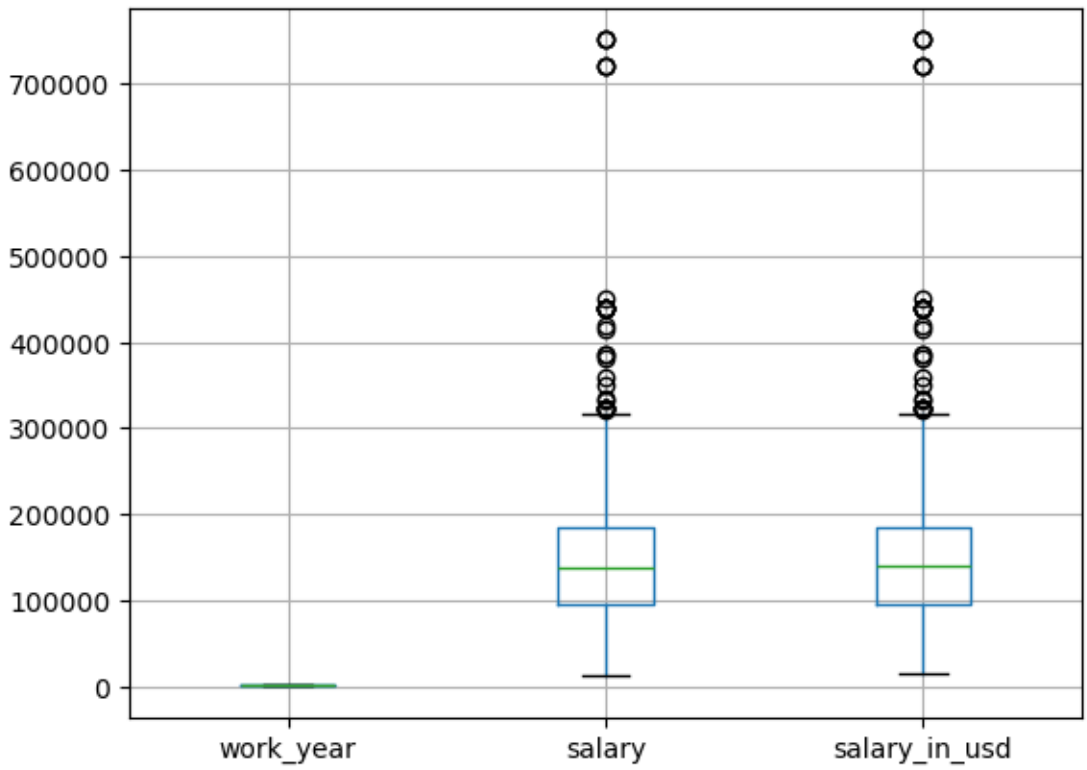
[29]:
```
#5. Find the outliers in 'salary' column of the data_science_salaries.
#URK22AI1048

df[['salary']].boxplot()
```

[29]: <Axes: >



[3]:
```
#6.Calculate the IQR using quantile and remove the outliers in work_year column␣
 ↪using IQR
#URK22AI1048


q75, q25 = np.percentile(df['work_year'], [75, 25])
iqr = q75 - q25

Q1 = df['work_year'].quantile(0.25)
Q3 = df['work_year'].quantile(0.75)
IQR = Q3 - Q1
l = Q1 - 1.5 * IQR
b = Q3 + 1.5 * IQR

s1 = df.loc[(df['work_year'] >= l) & (df['work_year'] <= b)]
s1.boxplot()
```

[3]: <Axes: >



[9]:
```
#7.Calculate the z-score of the work_year column and remove the outlier. Verify using box plot
#URK22AI1048



df['work_yeard'] = stats(df['work_year'])
df[['work_yeard']].boxplot()
df
```

[9]:

	job_title	experience_level	employment_type	work_models	\
0	Data Engineer	Mid-level	Full-time	Remote	
1	Data Engineer	Mid-level	Full-time	Remote	
2	Data Scientist	Senior-level	Full-time	Remote	
3	Data Scientist	Senior-level	Full-time	Remote	
4	BI Developer	Mid-level	Full-time	On-site	
..	…	…	…	…	
993	Data Engineer	Mid-level	Full-time	On-site	
994	Data Engineer	Mid-level	Full-time	On-site	
995	Data Manager	Senior-level	Full-time	Remote	
996	Data Manager	Senior-level	Full-time	Remote	
997	Data Analytics Lead	Senior-level	Full-time	On-site	

```
     work_year employee_residence  salary salary_currency  salary_in_usd  \
0     0.762638      United States  148100             USD         148100
1     0.762638      United States   98700             USD          98700
2     0.762638      United States  140032             USD         140032
3     0.762638      United States  100022             USD         100022
4     0.762638      United States  120000             USD         120000
..         …                 …     …               …            …
993  -1.311238      United States  133000             USD         133000
994  -1.311238      United States   58400             USD          58400
995  -1.311238      United States  144300             USD         144300
996  -1.311238      United States  104800             USD         104800
997  -1.311238      United States  115920             USD         115920

    company_location company_size  work_yeard
0      United States       Medium    0.762638
1      United States       Medium    0.762638
2      United States       Medium    0.762638
3      United States       Medium    0.762638
4      United States       Medium    0.762638
..               …           …         …
993    United States       Medium   -1.311238
994    United States       Medium   -1.311238
995    United States       Medium   -1.311238
996    United States       Medium   -1.311238
997    United States       Medium   -1.311238

[998 rows x 12 columns]
```

[80]:
```
#8.Insert a new_salary column by convertin usd to inr with 40% decrease.
#URK22AI1048

df['new_salary'] = df['salary_in_usd'] * 74.50 * (1 - 0.40)
df.new_salary
```

[80]:
```
0        6620070.0
1        4411890.0
2        6259430.4
3        4470983.4
4        5364000.0
           …
11082    2000459.1
11083     670500.0
11084    5234548.8
11085    3313611.0
11086    4078293.9
Name: new_salary, Length: 11087, dtype: float64
```

[86]:
```
#9.Rename the column 'work_model' into 'Work_mode'.
#URK22AI1048

df.rename(columns={'work_models': 'Work_mode'}, inplace=False)
```

[86]:

```
                    job_title experience_level employment_type Work_mode  \
0                Data Engineer        Mid-level       Full-time    Remote
1                Data Engineer        Mid-level       Full-time    Remote
2               Data Scientist     Senior-level       Full-time    Remote
3               Data Scientist     Senior-level       Full-time    Remote
4                 BI Developer        Mid-level       Full-time   On-site
…                           …                 …               …       …
11082       Staff Data Analyst      Entry-level        Contract    Hybrid
11083       Staff Data Analyst  Executive-level       Full-time   On-site
11084  Machine Learning Manager     Senior-level       Full-time    Hybrid
11085            Data Engineer        Mid-level       Full-time    Hybrid
11086           Data Scientist     Senior-level       Full-time   On-site

       work_year employee_residence    salary salary_currency  salary_in_usd  \
0         2024.0      United States  148100.0             USD       148100.0
1         2024.0      United States   98700.0             USD        98700.0
2         2024.0      United States  140032.0             USD       140032.0
3         2024.0      United States  100022.0             USD       100022.0
4         2024.0      United States  120000.0             USD       120000.0
…          …                  …       …               …            …
11082     2020.0             Canada   60000.0             CAD        44753.0
11083     2020.0            Nigeria   15000.0             USD        15000.0
11084     2020.0             Canada  157000.0             CAD       117104.0
11085     2020.0            Austria   65000.0             EUR        74130.0
11086     2020.0            Austria   80000.0             EUR        91237.0

      company_location company_size  work_yeard   s  new_salary
0        United States       Medium         NaN NaN   6620070.0
1        United States       Medium         NaN NaN   4411890.0
2        United States       Medium         NaN NaN   6259430.4
3        United States       Medium         NaN NaN   4470983.4
4        United States       Medium         NaN NaN   5364000.0
…                  …            …          …  ..          …
11082           Canada        Large         NaN NaN   2000459.1
11083           Canada       Medium         NaN NaN    670500.0
11084           Canada        Large         NaN NaN   5234548.8
11085          Austria        Large         NaN NaN   3313611.0
11086          Austria        Small         NaN NaN   4078293.9

[11087 rows x 14 columns]
```

[90]:
```
#10. Remove the column with index 0,1,6,7,8,9.
#URK22AI1048
p=[0, 1, 6, 7, 8, 9]
d = df.drop(p, inplace=False)
d
```

[90]:

```
                          job_title experience_level employment_type  \
2                     Data Scientist     Senior-level       Full-time
3                     Data Scientist     Senior-level       Full-time
4                       BI Developer        Mid-level       Full-time
5                       BI Developer        Mid-level       Full-time
10     Business Intelligence Developer        Mid-level       Full-time
…                                  …                …               …
11082              Staff Data Analyst      Entry-level        Contract
11083              Staff Data Analyst  Executive-level       Full-time
11084         Machine Learning Manager     Senior-level       Full-time
11085                   Data Engineer        Mid-level       Full-time
11086                   Data Scientist     Senior-level       Full-time

      work_models  work_year employee_residence    salary salary_currency  \
2          Remote     2024.0      United States  140032.0             USD
3          Remote     2024.0      United States  100022.0             USD
4         On-site     2024.0      United States  120000.0             USD
5         On-site     2024.0      United States   62100.0             USD
10        On-site     2024.0      United States   87800.0             USD
…             …          …                  …         …               …
11082      Hybrid     2020.0             Canada   60000.0             CAD
11083     On-site     2020.0            Nigeria   15000.0             USD
11084      Hybrid     2020.0             Canada  157000.0             CAD
11085      Hybrid     2020.0            Austria   65000.0             EUR
11086     On-site     2020.0            Austria   80000.0             EUR

       salary_in_usd company_location company_size  work_yeard   s  new_salary
2           140032.0    United States       Medium         NaN NaN   6259430.4
3           100022.0    United States       Medium         NaN NaN   4470983.4
4           120000.0    United States       Medium         NaN NaN   5364000.0
5            62100.0    United States       Medium         NaN NaN   2775870.0
10           87800.0    United States       Medium         NaN NaN   3924660.0
…                …                …            …          …  ..         …
11082        44753.0           Canada        Large         NaN NaN   2000459.1
11083        15000.0           Canada       Medium         NaN NaN    670500.0
11084       117104.0           Canada        Large         NaN NaN   5234548.8
11085        74130.0          Austria        Large         NaN NaN   3313611.0
11086        91237.0          Austria        Small         NaN NaN   4078293.9

[11081 rows x 14 columns]
```

[14]:
```
#11. Display 20 rows with missing values in the company_location column and␣
 ↪drop the missing values.
#URK22AI1048

p=df[df['company_location'].isnull()].head(20)

pd1 = p.dropna(subset=['company_location'], inplace=False)
print(pd1)
p
```

```
Empty DataFrame
Columns: [job_title, experience_level, employment_type, work_models, work_year,
employee_residence, salary, salary_currency, salary_in_usd, company_location,
company_size, work_yeard]
Index: []
```

[14]:
```
Empty DataFrame
Columns: [job_title, experience_level, employment_type, work_models, work_year,
employee_residence, salary, salary_currency, salary_in_usd, company_location,
company_size, work_yeard]
Index: []
```

[ ]:
```
#12. Identify the missing values in the all columns and perform the following␣
 ↪operations.
#URK22AI1048

#a) Fill the missing values with '0'
df.fillna(0)

#b) Fill the missing values with mean value
p=df.mean()
df.fillna(p)

#c) Fill the missing values with median value
p1=df.median()
df.fillna(p1)

#d) Fill the missing values with previous value
df.fillna(method='ffill')

#e) Fill the missing values with next value
df.fillna(method='bfill')

#f) Fill the missing values with linear interpolation
df.interpolate(method='linear')
```

[ ]:
```
#13. Plot the heatmap using the correlation for employee table and titanic␣
 ↪dataset.
#URK22AI1048

import seaborn as sns
corr1=df1.corr()
sns.heatmap(corr1,annot=True)
plt.show()
corr = df1.corr()
corr.style.background_gradient(cmap ='Blues'
```

[108]:
```
#14. Calculate the mean, median, std deviation, variance of given dataset's␣
 ↪quantitative data.
#URK22AI1048

print(df.mean())
print(df.median())
print(df.std())
print(df.var())
```

```
work_year        2.022850e+03
salary           1.696318e+05
salary_in_usd    1.496212e+05
work_yeard                NaN
s                         NaN
new_salary       6.688069e+06
dtype: float64
work_year          2023.0
salary           142352.0
salary_in_usd    142000.0
work_yeard            NaN
s                     NaN
new_salary       6347400.0
dtype: float64
work_year        5.646007e-01
salary           4.081698e+05
salary_in_usd    6.670842e+04
work_yeard                NaN
s                         NaN
new_salary       2.981866e+06
dtype: float64
work_year        3.187739e-01
salary           1.666026e+11
salary_in_usd    4.450014e+09
work_yeard                NaN
s                         NaN
new_salary       8.891528e+12
dtype: float64
```

```
<ipython-input-108-0fb359c62523>:4: FutureWarning: The default value of
numeric_only in DataFrame.mean is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(df.mean())
<ipython-input-108-0fb359c62523>:5: FutureWarning: The default value of
numeric_only in DataFrame.median is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(df.median())
<ipython-input-108-0fb359c62523>:6: FutureWarning: The default value of
numeric_only in DataFrame.std is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(df.std())
<ipython-input-108-0fb359c62523>:7: FutureWarning: The default value of
numeric_only in DataFrame.var is deprecated. In a future version, it will
default to False. In addition, specifying 'numeric_only=None' is deprecated.
Select only valid columns or specify the value of numeric_only to silence this
warning.
  print(df.var())
```

[ ]:
```
Result:

The Above Program were Created and Executed Successfully
```