DSC530-T301 Week 12 by Stephanie Benavidez

12.2 Final Term Project

For this final term project, I thought that since I am working on my Master's in data science I would look for a data science dataset to see what the future holds. Throughout the classes that I have taken so far, I have learned that data scientists have a variety of fields and responsibilities.

A few of the responsibilities that I have learned so far are as follows: • Data acquisition • Data preparation • Data cleaning • Data transformation • Handling Outliers • Data integration • Data Reduction • Data Mining • Model Building

The datasets that I am using are coming from:

1. Data Science Salaries 2023. (n.d.). Retrieved from www.kaggle.com (http://www.kaggle.com): https://www.kaggle.com/datasets/arnabchaki/data-science-salaries-2023?select=ds_salaries.csv (https://www.kaggle.com/datasets/arnabchaki/data-science-salaries-2023?select=ds_salaries.csv) Duggal, N. (2023, February 21).
2. Vaddoriya, M. (n.d.). Data Science job salary Dataset. Retrieved from www.kaggle.com/datasets (http://www.kaggle.com/datasets): https://www.kaggle.com/datasets/milanvaddoriya/data-science-job-salary (https://www.kaggle.com/datasets/milanvaddoriya/data-science-job-salary)

With these datasets they will need to clean up, and the column names with need to be linked to create one data frame. I will be conducting several tasks throughout this project and will try to summarize the results after each task has been completed to analyze the information that is shown.

Join me through this journey.

Load Datasets, load libraries, and clean up datasets

In [1]:
```
# Load all of the necessary libraries
import pandas as pd
import numpy as np
import openpyxl
import os
from matplotlib import pyplot as plt
from scipy.stats import norm
import scipy.stats as stats
import statsmodels.api as sm
```

In [2]:
```
# set/check working directory
os.getcwd()
```

Out[2]:
```
'C:\\Users\\SBenavidez\\PycharmProjects\\pythonProject\\venv\\Scripts'
```

In [3]:
```
# change working directory
os.chdir("U:\School Homework\Spring 2023\Data Exp. & Analysis\Data Exp Project")
```

In [4]:
```
# retrieve all excel files in new working directory
path = "U:/School Homework/Spring 2023/Data Exp. & Analysis/Data Exp Project"

# check file names and assign to variable files
os.listdir(path)
files = os.listdir(path)
print(files)
```

```
['Clean_Salaries_Data.csv', 'datascience_salaries.csv', 'ds_salaries.csv']
```

In [5]:
```
# read each csv file and rename columns
# read datascience_salaries  and show column names
data1 = pd.read_csv('datascience_salaries.csv')
data1.columns
```

Out[5]:
```
Index(['Unnamed: 0', 'job_title', 'job_type', 'experience_level', 'location',
       'salary_currency', 'salary'],
      dtype='object')
```

In [6]:
```
# rename columns in data1
data1 = data1.rename(columns = {'job_title':'Job_Title','job_type':'Job_Type', 'experience_level':'Experience_
                                'salary_currency':'Salary_Currency', 'salary':'Salary'}, inplace = False)
# drop unnamed column and show first 5 rows of data1 with renamed columns
data1 = data1.drop('Unnamed: 0', axis =1)
data1.head()
```

Out[6]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary
0	Data scientist	Full Time	Senior	New York City	USD	149000
1	Data scientist	Full Time	Senior	Boston	USD	120000
2	Data scientist	Full Time	Senior	London	USD	68000
3	Data scientist	Full Time	Senior	Boston	USD	120000
4	Data scientist	Full Time	Senior	New York City	USD	149000

In [7]:
```
# read ds_salaries and show column names
data2 = pd.read_csv('ds_salaries.csv')
data2.columns
```

Out[7]:
```
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')
```

In [8]:
```
# rename columns in data2
data2 = data2.rename(columns = {'experience_level':'Experience_level', 'employment_type':'Job_Type',
                                'job_title':'Job_Title','salary':'Salary','salary_currency':'Salary_Currency',
                                'remote_ratio':'Remote_Ratio','company_location':'Location'}, inplace = False)
# show first 5 rows of data2 with renamed columns
data2.head()
```

Out[8]:

	work_year	Experience_level	Job_Type	Job_Title	Salary	Salary_Currency	salary_in_usd	employee_residence	Remote_Ratio	Locat
0	2023	SE	FT	Principal Data Scientist	80000	EUR	85847	ES	100	
1	2023	MI	CT	ML Engineer	30000	USD	30000	US	100	
2	2023	MI	CT	ML Engineer	25500	USD	25500	US	100	
3	2023	SE	FT	Data Scientist	175000	USD	175000	CA	100	
4	2023	SE	FT	Data Scientist	120000	USD	120000	CA	100	

In [9]:
```
# create an empty dataframe
dproject = pd.DataFrame()

# concatenate data1, data2, data3 into dataframe
dproject = pd.concat([data1, data2])
# display concatenated dataframe
dproject
```

Out[9]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	work_year	salary_in_usd	employee_residence	Remot
0	Data scientist	Full Time	Senior	New York City	USD	149000	NaN	NaN	NaN	
1	Data scientist	Full Time	Senior	Boston	USD	120000	NaN	NaN	NaN	
2	Data scientist	Full Time	Senior	London	USD	68000	NaN	NaN	NaN	
3	Data scientist	Full Time	Senior	Boston	USD	120000	NaN	NaN	NaN	
4	Data scientist	Full Time	Senior	New York City	USD	149000	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	...	...	
3750	Data Scientist	FT	SE	US	USD	412000	2020.0	412000.0	US	
3751	Principal Data Scientist	FT	MI	US	USD	151000	2021.0	151000.0	US	
3752	Data Scientist	FT	EN	US	USD	105000	2020.0	105000.0	US	
3753	Business Data Analyst	CT	EN	US	USD	100000	2020.0	100000.0	US	
3754	Data Science Manager	FT	SE	IN	INR	7000000	2021.0	94665.0	IN	

4926 rows × 11 columns

In [10]:
```
# removing duplicates from rows and printing out the results
duplicate = dproject[dproject.duplicated()]
print("\n\nDuplicate Rows : \n {}".format(duplicate))
drop_duplicate = dproject.drop_duplicates(keep=False)
print("\n\nResult of CSV after duplicates are removed :\n",drop_duplicate.head())
```

```
Duplicate Rows :
           Job_Title   Job_Type Experience_level       Location  \
3     Data scientist  Full Time           Senior         Boston
4     Data scientist  Full Time           Senior  New York City
5     Data scientist  Full Time           Senior         London
18    Data scientist  Full Time           Senior         London
25    Data scientist  Full Time           Senior       San Jose
...              ...        ...              ...            ...
3439  Data Scientist         FT               MI             US
3440   Data Engineer         FT               SE             US
3441   Data Engineer         FT               SE             US
3586   Data Engineer         FT               MI             US
3709  Data Scientist         FT               MI             DE

     Salary_Currency  Salary  work_year  salary_in_usd employee_residence  \
3                USD  120000        NaN            NaN                NaN
4                USD  149000        NaN            NaN                NaN
5                USD   68000        NaN            NaN                NaN
18               USD   68000        NaN            NaN                NaN
25               USD   68000        NaN            NaN                NaN
...              ...     ...        ...            ...                ...
3439             USD   78000     2022.0        78000.0                 US
3440             USD  135000     2022.0       135000.0                 US
3441             USD  115000     2022.0       115000.0                 US
3586             USD  200000     2021.0       200000.0                 US
3709             EUR   76760     2021.0        90734.0                 DE

      Remote_Ratio company_size
3              NaN          NaN
4              NaN          NaN
5              NaN          NaN
18             NaN          NaN
25             NaN          NaN
...            ...          ...
3439         100.0            M
3440         100.0            M
3441         100.0            M
3586         100.0            L
3709          50.0            L

[1516 rows x 11 columns]
```

```
Result of CSV after duplicates are removed :
          Job_Title   Job_Type Experience_level                Location  \
6   Data scientist  Full Time           Senior  Research Triangle Park
7   Data scientist  Full Time           Senior                  Sydney
8   Data scientist  Full Time           Senior           San Francisco
10  Data scientist  Full Time            Entry               BangPa-in
12  Data scientist  Full Time           Senior                   NAMER

   Salary_Currency  Salary  work_year  salary_in_usd employee_residence  \
6              USD   69000        NaN            NaN                NaN
7              USD   68000        NaN            NaN                NaN
8              USD  140000        NaN            NaN                NaN
10             USD   35000        NaN            NaN                NaN
12             USD   68000        NaN            NaN                NaN

    Remote_Ratio company_size
6            NaN          NaN
7            NaN          NaN
8            NaN          NaN
10           NaN          NaN
12           NaN          NaN
```

In [11]:
```
# check to see if there is any  missing data in the csv file
pd.isnull(dproject).head()
```

Out[11]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	work_year	salary_in_usd	employee_residence	Remote_Ra
0	False	False	False	False	False	False	True	True	True	Tr
1	False	False	False	False	False	False	True	True	True	Tr
2	False	False	False	False	False	False	True	True	True	Tr
3	False	False	False	False	False	False	True	True	True	Tr
4	False	False	False	False	False	False	True	True	True	Tr

In [12]:
```
# fill NaN values with a o
dproject = dproject.fillna(0)
dproject.head()
```

Out[12]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	work_year	salary_in_usd	employee_residence	Remote_Ra
0	Data scientist	Full Time	Senior	New York City	USD	149000	0.0	0.0	0	
1	Data scientist	Full Time	Senior	Boston	USD	120000	0.0	0.0	0	
2	Data scientist	Full Time	Senior	London	USD	68000	0.0	0.0	0	
3	Data scientist	Full Time	Senior	Boston	USD	120000	0.0	0.0	0	
4	Data scientist	Full Time	Senior	New York City	USD	149000	0.0	0.0	0	

In [13]:
```
# use replace to get rid of NaN values
dproject.replace('',np.nan, inplace = True)
dproject
```

Out[13]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	work_year	salary_in_usd	employee_residence	Remot
0	Data scientist	Full Time	Senior	New York City	USD	149000	0.0	0.0	0	
1	Data scientist	Full Time	Senior	Boston	USD	120000	0.0	0.0	0	
2	Data scientist	Full Time	Senior	London	USD	68000	0.0	0.0	0	
3	Data scientist	Full Time	Senior	Boston	USD	120000	0.0	0.0	0	
4	Data scientist	Full Time	Senior	New York City	USD	149000	0.0	0.0	0	
...	...	...	...	...	...	...	...	...	...	
3750	Data Scientist	FT	SE	US	USD	412000	2020.0	412000.0	US	
3751	Principal Data Scientist	FT	MI	US	USD	151000	2021.0	151000.0	US	
3752	Data Scientist	FT	EN	US	USD	105000	2020.0	105000.0	US	
3753	Business Data Analyst	CT	EN	US	USD	100000	2020.0	100000.0	US	
3754	Data Science Manager	FT	SE	IN	INR	7000000	2021.0	94665.0	IN	

4926 rows × 11 columns

In [14]:
```
# use describe function to see information
dproject.describe(include = 'all')
```

Out[14]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	work_year	salary_in_usd	employee_residenc
count	4926	4926	4926	4926	4926	4.926000e+03	4926.000000	4926.000000	492
unique	98	6	8	392	20	NaN	NaN	NaN	7
top	Data Engineer	FT	SE	US	USD	NaN	NaN	NaN	U
freq	1040	3718	2516	3040	4381	NaN	NaN	NaN	300
mean	NaN	NaN	NaN	NaN	NaN	1.607765e+05	1541.618555	104867.400325	Na
std	NaN	NaN	NaN	NaN	NaN	5.890704e+05	860.983044	80379.246693	Na
min	NaN	NaN	NaN	NaN	NaN	6.000000e+03	0.000000	0.000000	Na
25%	NaN	NaN	NaN	NaN	NaN	6.800000e+04	2020.000000	17707.750000	Na
50%	NaN	NaN	NaN	NaN	NaN	1.200000e+05	2022.000000	110000.000000	Na
75%	NaN	NaN	NaN	NaN	NaN	1.650000e+05	2023.000000	160000.000000	Na
max	NaN	NaN	NaN	NaN	NaN	3.040000e+07	2023.000000	450000.000000	Na

In [15]:
```
# create new dataframe to only include certain columns needed
dproject2 = dproject[['Job_Title', 'Job_Type', 'Experience_level', 'Location', 'Salary_Currency', 'Salary', 'F
dproject2
```

Out[15]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	Remote_Ratio
0	Data scientist	Full Time	Senior	New York City	USD	149000	0.0
1	Data scientist	Full Time	Senior	Boston	USD	120000	0.0
2	Data scientist	Full Time	Senior	London	USD	68000	0.0
3	Data scientist	Full Time	Senior	Boston	USD	120000	0.0
4	Data scientist	Full Time	Senior	New York City	USD	149000	0.0
...	...	...	...	...	...	...	...
3750	Data Scientist	FT	SE	US	USD	412000	100.0
3751	Principal Data Scientist	FT	MI	US	USD	151000	100.0
3752	Data Scientist	FT	EN	US	USD	105000	100.0
3753	Business Data Analyst	CT	EN	US	USD	100000	100.0
3754	Data Science Manager	FT	SE	IN	INR	7000000	50.0

4926 rows × 7 columns

In [16]:
```
# check shape of dataframe
dproject2.shape
```

Out[16]:
```
(4926, 7)
```

In [17]:
```
# getting length and info from dataframe
print(len(dproject2))
print(dproject2.info)
```

```
4926
<bound method DataFrame.info of                       Job_Title   Job_Type Experience_level       Location  \
0               Data scientist  Full Time           Senior  New York City
1               Data scientist  Full Time           Senior         Boston
2               Data scientist  Full Time           Senior         London
3               Data scientist  Full Time           Senior         Boston
4               Data scientist  Full Time           Senior  New York City
...                        ...        ...              ...            ...
3750            Data Scientist         FT               SE             US
3751  Principal Data Scientist         FT               MI             US
3752            Data Scientist         FT               EN             US
3753     Business Data Analyst         CT               EN             US
3754      Data Science Manager         FT               SE             IN

     Salary_Currency   Salary  Remote_Ratio
0                USD   149000           0.0
1                USD   120000           0.0
2                USD    68000           0.0
3                USD   120000           0.0
4                USD   149000           0.0
...              ...      ...           ...
3750             USD   412000         100.0
3751             USD   151000         100.0
3752             USD   105000         100.0
3753             USD   100000         100.0
3754             INR  7000000          50.0

[4926 rows x 7 columns]>
```

In [18]:
```
# need to replace abbreviation values in Job_Type column
dict = {'FT':'Full Time', 'CT':'Contract', 'FL':'Full Time', 'PT':'Part Time'}
dproject2['Job_Type'] = dproject2['Job_Type'].replace(dict)
dproject2
```

```
C:\Users\SBenavidez\AppData\Local\Temp\ipykernel_30368\3917256907.py:3: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  dproject2['Job_Type'] = dproject2['Job_Type'].replace(dict)
```

Out[18]:

	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	Remote_Ratio
0	Data scientist	Full Time	Senior	New York City	USD	149000	0.0
1	Data scientist	Full Time	Senior	Boston	USD	120000	0.0
2	Data scientist	Full Time	Senior	London	USD	68000	0.0
3	Data scientist	Full Time	Senior	Boston	USD	120000	0.0
4	Data scientist	Full Time	Senior	New York City	USD	149000	0.0
...	...	...	...	...	...	...	...
3750	Data Scientist	Full Time	SE	US	USD	412000	100.0
3751	Principal Data Scientist	Full Time	MI	US	USD	151000	100.0
3752	Data Scientist	Full Time	EN	US	USD	105000	100.0
3753	Business Data Analyst	Contract	EN	US	USD	100000	100.0
3754	Data Science Manager	Full Time	SE	IN	INR	7000000	50.0

4926 rows × 7 columns

In [19]:
```
# need to replace abbreviation values in Job_Type column
dict2 = {'SE':'Senior', 'MI':'Mid', 'EN':'Entry', 'EX':'Executive'}
dproject2['Experience_level'] = dproject2['Experience_level'].replace(dict2)
dproject2
```

```
C:\Users\SBenavidez\AppData\Local\Temp\ipykernel_30368\3204903893.py:3: 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 (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  dproject2['Experience_level'] = dproject2['Experience_level'].replace(dict2)
```

Out[19]:

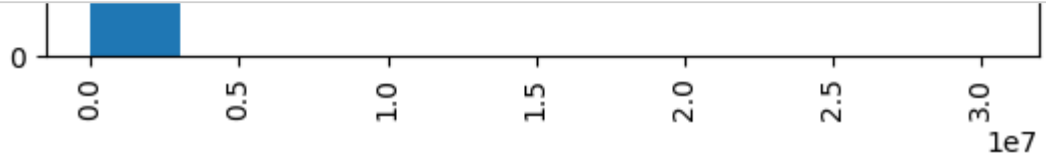
	Job_Title	Job_Type	Experience_level	Location	Salary_Currency	Salary	Remote_Ratio
0	Data scientist	Full Time	Senior	New York City	USD	149000	0.0
1	Data scientist	Full Time	Senior	Boston	USD	120000	0.0
2	Data scientist	Full Time	Senior	London	USD	68000	0.0
3	Data scientist	Full Time	Senior	Boston	USD	120000	0.0
4	Data scientist	Full Time	Senior	New York City	USD	149000	0.0
...	...	...	...	...	...	...	...
3750	Data Scientist	Full Time	Senior	US	USD	412000	100.0
3751	Principal Data Scientist	Full Time	Mid	US	USD	151000	100.0
3752	Data Scientist	Full Time	Entry	US	USD	105000	100.0
3753	Business Data Analyst	Contract	Entry	US	USD	100000	100.0
3754	Data Science Manager	Full Time	Senior	IN	INR	7000000	50.0

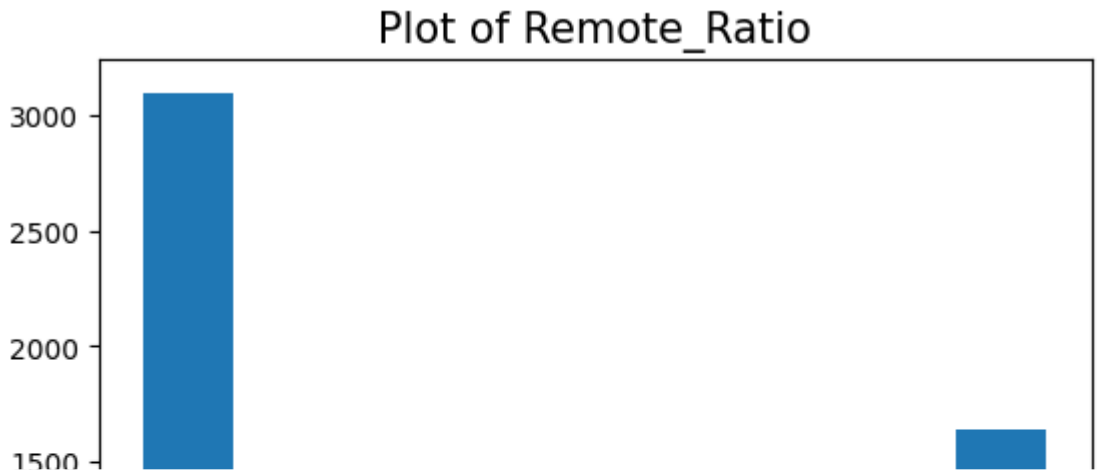
4926 rows × 7 columns

In [20]:
```
# write dproject2 to csv
dproject2.to_csv('Clean_Salaries_Data.csv', index=False)
```

Plotting of histograms for the variables chosen

In [21]:
```
# plotting each column in a histagram
for c in dproject2.columns:
    plt.title("Plot of " + c, fontsize = 15)
    plt.hist(dproject2[c], bins = 10, label=[c])
    plt.xticks(rotation = 90)
    spacing = 0.1000
    plt.subplots_adjust(bottom=spacing)
    plt.show()
```





For each Column Histogram, I will try to analyze the information as best as possible.

For the plot of the Job Title, it is easy to see that several Job Titles are over 2500, however, it is hard to tell what job title is hitting that mark.

For the plot of the Job Type, it clearly stats that all of the job's in this dataset are full time. From their it looks like the order would be internship, part time, and contract being the lowest.

For the plot of Experience Level, a Senior level is the highest in the dataset. Second would be a Mid level employee, third would be an entry level employee, and lastly would be an executive level employee.

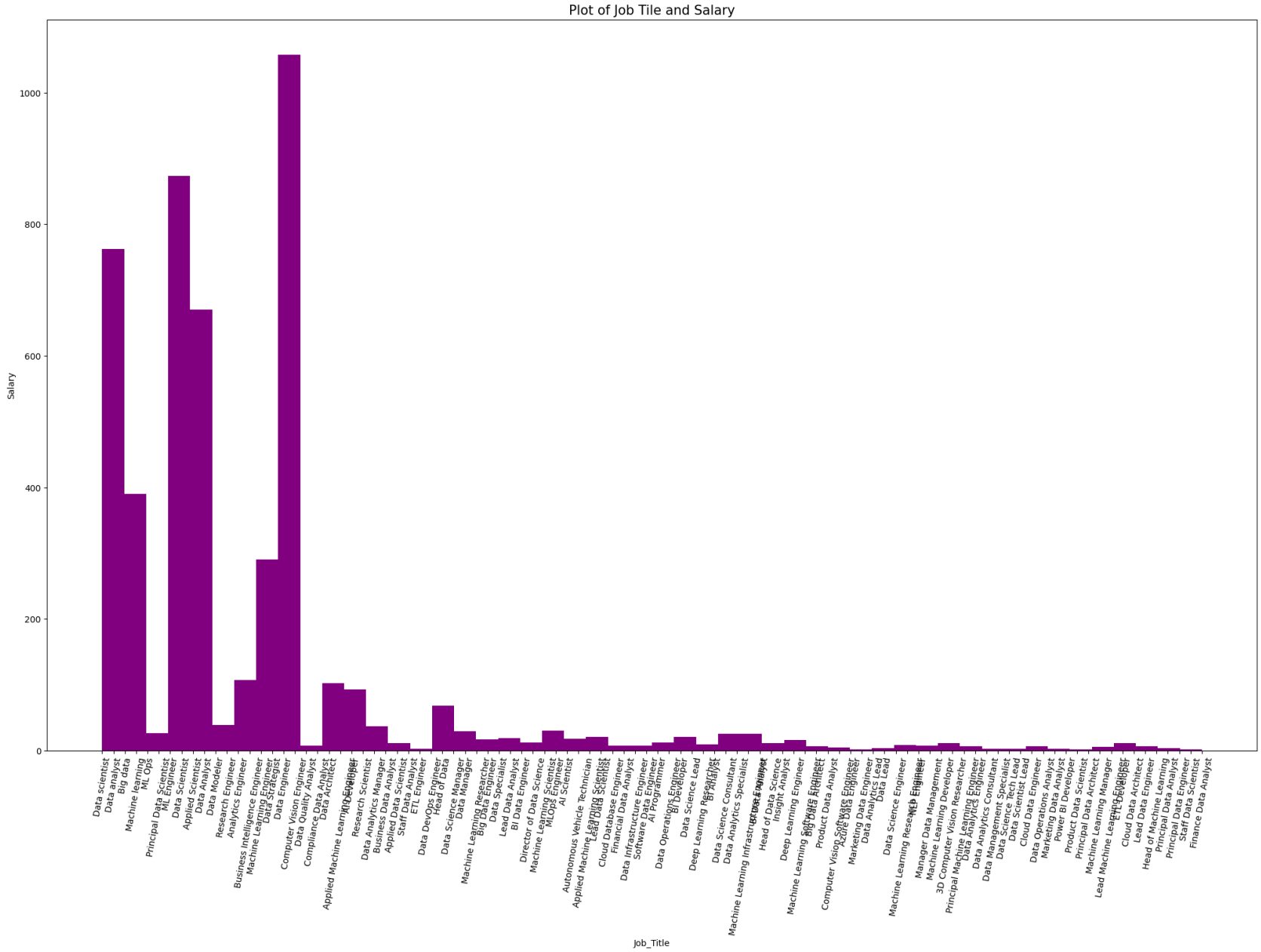
For the plot of the location in the dataset, it is hard to read which location has the largest location. However, you can see that it is over the 3500 marker.

For the plot of Salary Currency, it looks there are three currency's that are over the 4000 marker. Those three are USD, EUR, and GBP.

For the plot of Salary, from the column itself it doesn't tell you too much detail other than it is almost at the 5000 marker.

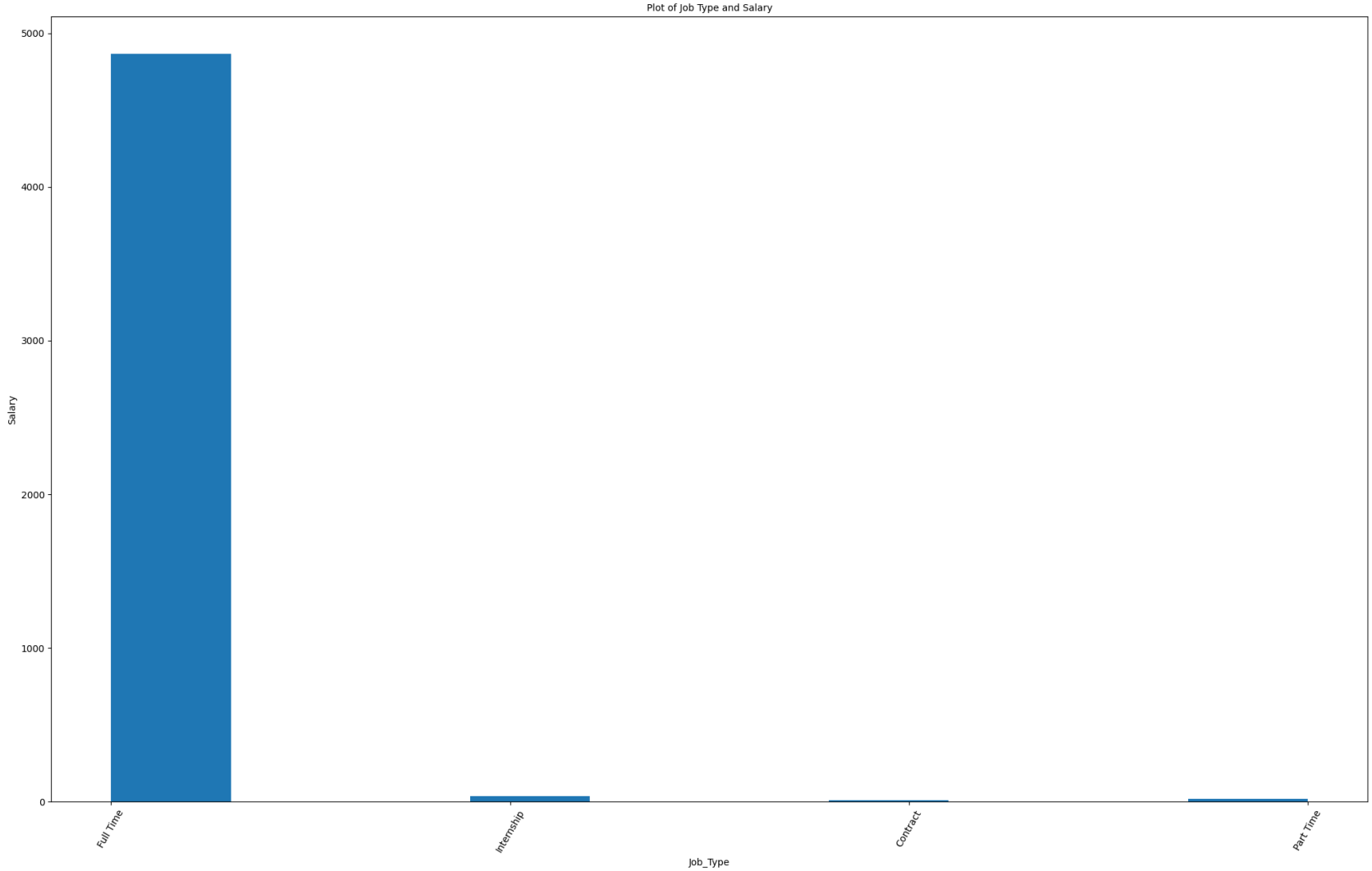
For the plot of Remote Ratio, you can see that majority of the jobs don't have the capability of working remotely. There are a few 60%

In [22]:
```
# plotting histogram for Job_Title & Salary
plt.figure(figsize =(25,15))
plt.title('Plot of Job Tile and Salary',fontsize=15)
plt.hist(dproject2['Job_Title'],bins=50,color='purple')
plt.xlabel('Job_Title')
plt.ylabel('Salary')
plt.xticks(rotation = 80)
spacing = 0.1000
plt.subplots_adjust(bottom=spacing)
plt.show()
```



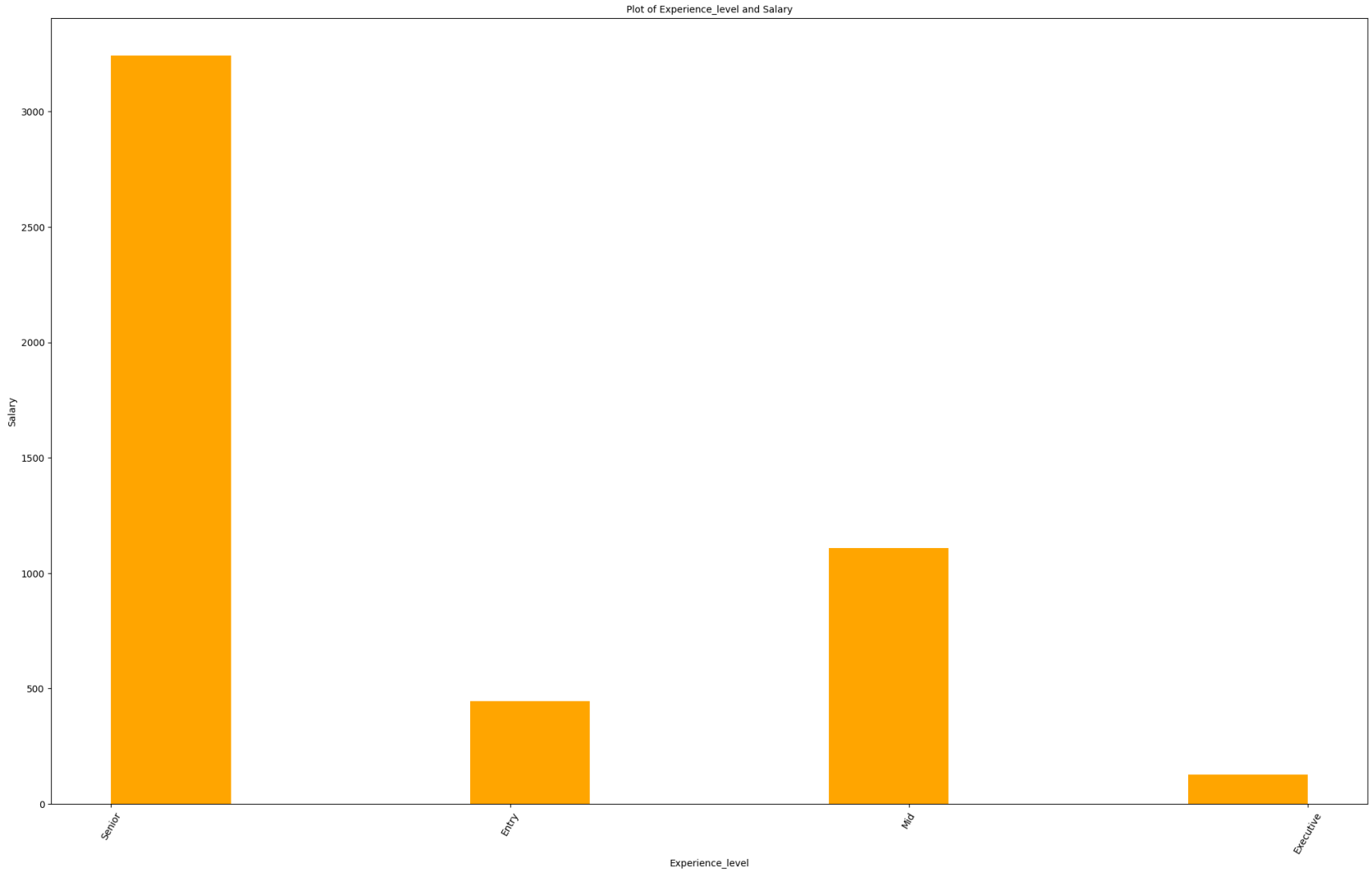
For the comparison histogram of the Job Title and Salary, you can see that the top three job title are Data Engineer, ML Engineer, and Data Scientist. Though the titles are still hard to read, it seems like a Finance Data Analyst or a form of Data Analyst have the lowest Salary.

In [23]:
```
# plotting histogram for Job_Type & Salary
plt.figure(figsize =(25,15))
plt.title('Plot of Job Type and Salary',fontsize=10)
plt.hist(dproject2['Job_Type'],bins=10)
plt.xlabel('Job_Type')
plt.ylabel('Salary')
plt.xticks(rotation = 60)
plt.show()
```



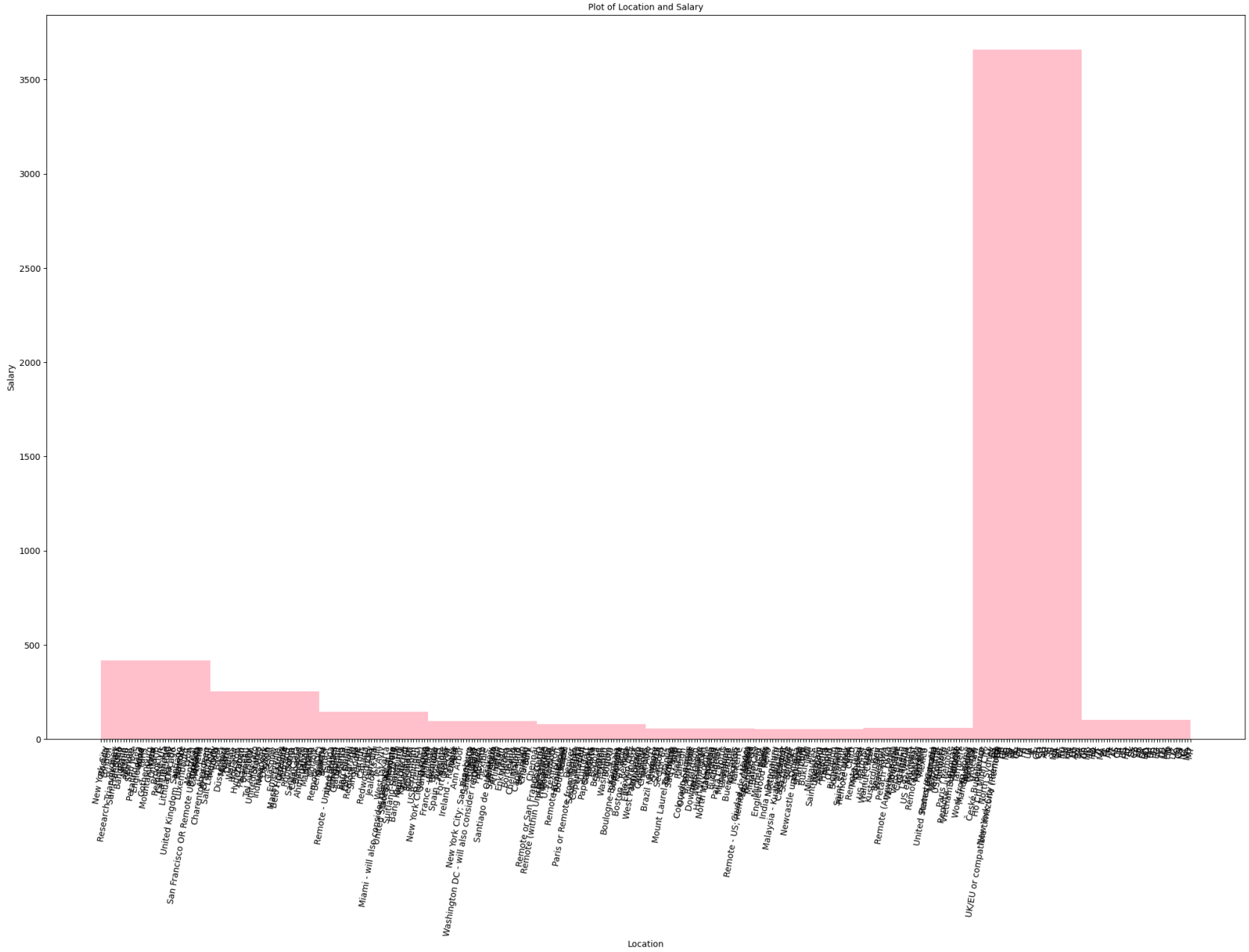
For the Job Type vs Salary histogram plot, it shows the same information as the column histogram showed, showing that the highest salary would be a full time employee with the contract employee having the lowest salary.

In [24]:
```
# plotting histogram for Exeperience_Level & Salary
plt.figure(figsize =(25,15))
plt.title('Plot of Experience_level and Salary',fontsize=10)
plt.hist(dproject2['Experience_level'],bins=10, color = 'orange')
plt.xlabel('Experience_level')
plt.ylabel('Salary')
plt.xticks(rotation = 60)
plt.show()
```



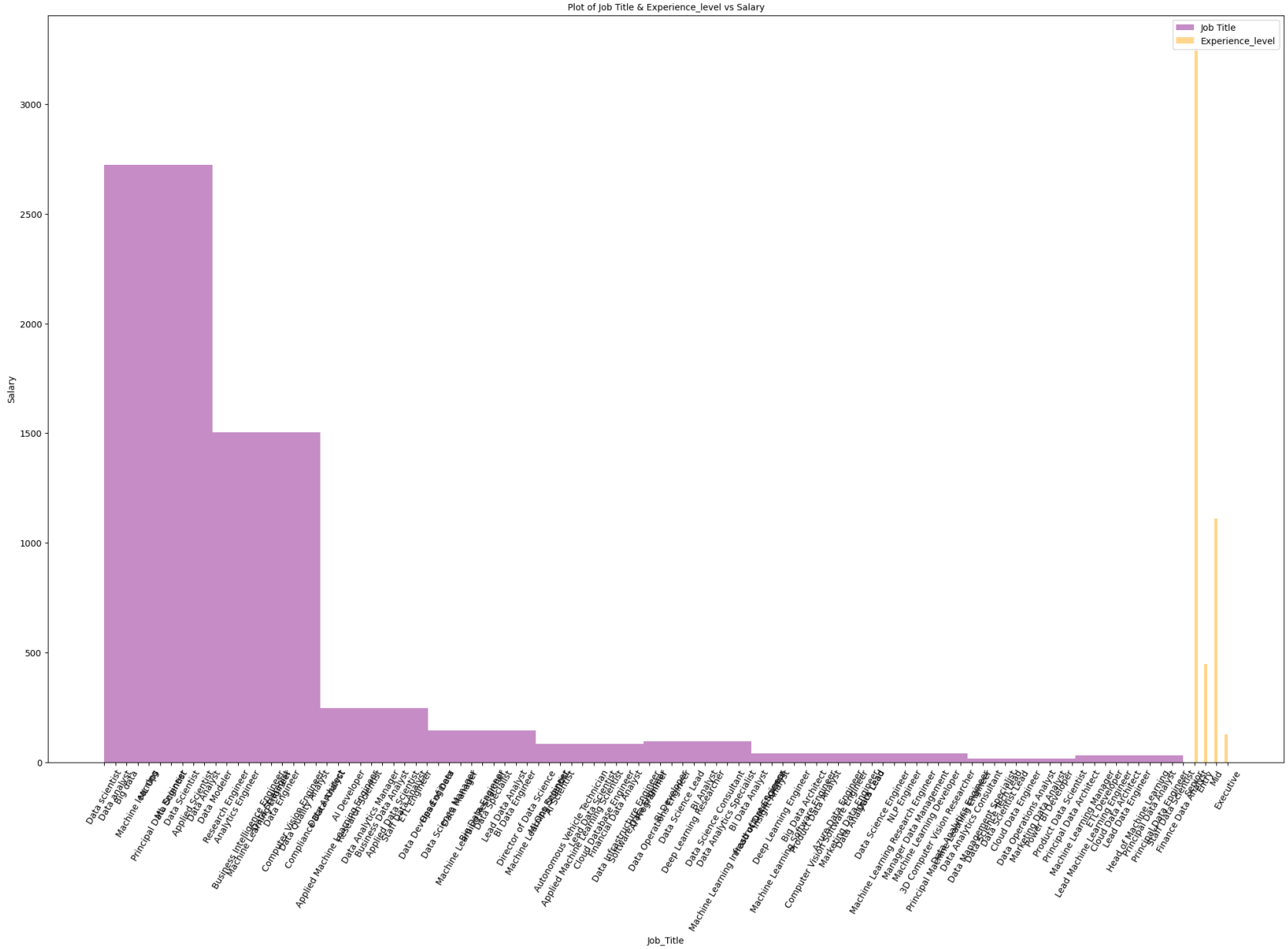
For the Experience Level vs Salary histogram plot, it shows the same information as the column histogram showed, showing that the highest salary would be a Senior level employee with the Executive level employee having the lowest salary.

In [25]:
```
# plotting histogram for Exeperience_Level & Salary
plt.figure(figsize =(25,15))
plt.title('Plot of Location and Salary',fontsize=10)
plt.hist(dproject2['Location'],bins=10, color = 'pink')
plt.xlabel('Location')
plt.ylabel('Salary')
plt.xticks(rotation = 80)
spacing = 0.1000
plt.subplots_adjust(bottom=spacing)
plt.show()
```



From the Location histogram, it is still hard to read which locations have the largest salary in this dataset.

In [26]:
```
# plotting histogram for Job Title Job Type & Salary
plt.figure(figsize =(25,15))
plt.title('Plot of Job Title & Experience_level vs Salary',fontsize=10)
plt.hist(dproject2['Job_Title'],bins=10,alpha= 0.45, color='purple')
plt.hist(dproject2['Experience_level'],bins=10, alpha= 0.45, color= 'orange')
plt.xlabel('Job_Title')
plt.ylabel('Salary')
plt.legend(['Job Title', 'Experience_level'])
plt.xticks(rotation = 60)
spacing = 0.1000
plt.subplots_adjust(bottom=spacing)
plt.show()
```



For this plot, I wanted to compare the Experience level with the Job Title against the Salary. For the Senior Level, it looks like the Data Scientist salary has the largest salary, then a data analyst, and big data job titles.

Descriptive Charateristics about the variables

In [27]:
```
# mean of dproject2
dproject2.mean(numeric_only=True)
```

Out[27]:
```
Salary          160776.466098
Remote_Ratio        35.272026
dtype: float64
```

In [28]:
```
# median of dproject2
dproject2.median(numeric_only=True)
```

Out[28]:
```
Salary          120000.0
Remote_Ratio         0.0
dtype: float64
```

In [29]:
```
# mode of dproject2
dproject2.mode(numeric_only=True)
```

Out[29]:

	Salary	Remote_Ratio
0	68000	0.0

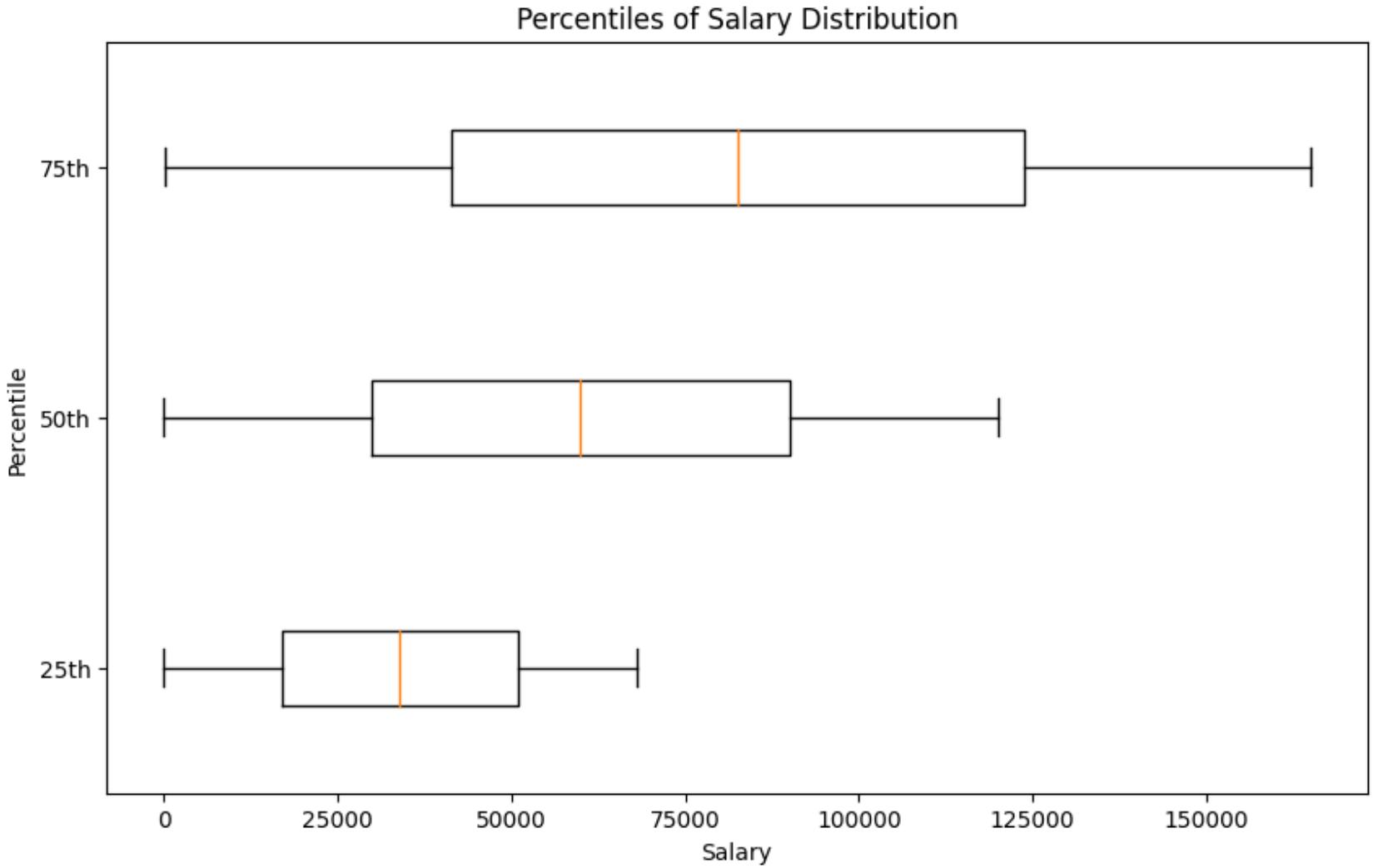
In [30]:
```
# std of dproject2
dproject2.std(numeric_only=True)
```

Out[30]:
```
Salary          589070.359482
Remote_Ratio        46.771907
dtype: float64
```

The analysis of the descriptive chartesitsics on the numeric values of the variables are as follows:

1. mean of dproject 2 are Salary is 160776.466098, and the Remote_Ratio is 35.272026
2. median of dproject 2 are Salary is 120000.0, and the Remote_Ratio is 0
3. mode of dproject 2 are Sarlary is 68000, and the Remote_Ratio is 0
4. std of dproject 2 are Salary is 589070.359482, and the Remote_Ratio is 46.771907

In [31]:
```
# percentiles of dproject2
d25th_percentile = dproject2.quantile(0.25, numeric_only=True)
d50th_percentile = dproject2.quantile(0.50, numeric_only=True)
d75th_percentile = dproject2.quantile(0.75, numeric_only=True)

# Create a list of percentiles
percentiles = [d25th_percentile, d50th_percentile, d75th_percentile]

# Plot the percentiles using a box plot
plt.figure(figsize=(10, 6))
plt.boxplot(percentiles, vert=False)
plt.xlabel('Salary')
plt.ylabel('Percentile')
plt.title('Percentiles of Salary Distribution')
plt.yticks([1, 2, 3], ['25th', '50th', '75th'])
plt.show()
```



For the Percentiles of the Salary Distribution, you can see that the Salary for the 75th percentile range is around 80,000 to over 125,000. The range for the 50th percentile is 50,000 to 75,000. The 25th percentile is 25,00 to 50,000.

In [32]:
```
# skewness of dproject2
dproject2.skew(numeric_only=True)
```

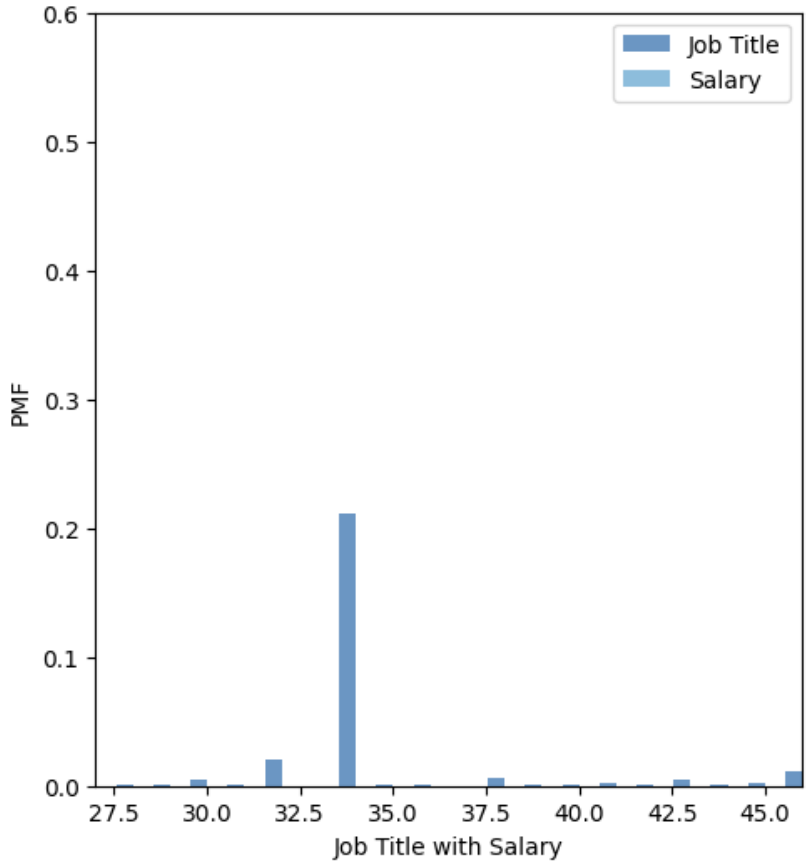
Out[32]:
```
Salary          32.844717
Remote_Ratio     0.616219
dtype: float64
```

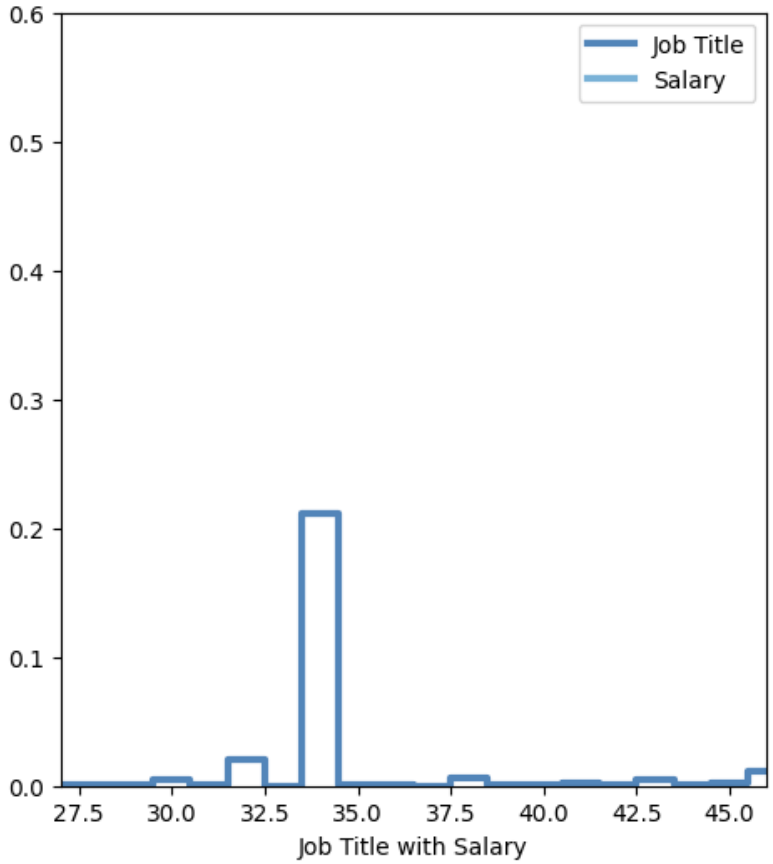
For the skewness of dproject 2, the Salary is 32.844717, and the Remote_Ratio is 0.616219.

Compare two scenarios using PMF

In [33]:
```
# load new libraries
import nsfg
import first
from thinkstats2 import Pmf
import thinkplot

# create variable to show meaning and how to plot pmf
jobtitle = dproject2['Job_Title']
salary = dproject2['Salary']
jobtitle_pmf = Pmf(dproject['Job_Title'].astype('category').cat.codes, label = "Job Title")
salary_pmf = Pmf(dproject['Salary'], label = "Salary")

# Plot Pmf
width = 0.45
axis = [27, 46, 0, 0.6]
thinkplot.PrePlot(2, cols=2)
thinkplot.Hist(jobtitle_pmf, align="right", width=width)
thinkplot.Hist(salary_pmf, align="left", width=width)
thinkplot.Config(xlabel="Job Title with Salary", ylabel="PMF", axis=axis)

thinkplot.PrePlot(2)
thinkplot.SubPlot(2)
thinkplot.Pmfs([jobtitle_pmf, salary_pmf])
thinkplot.Config(xlabel="Job Title with Salary", axis=axis)

# Customize the axis labels and title
thinkplot.SubPlot(1)
thinkplot.Config(axis=axis)
```

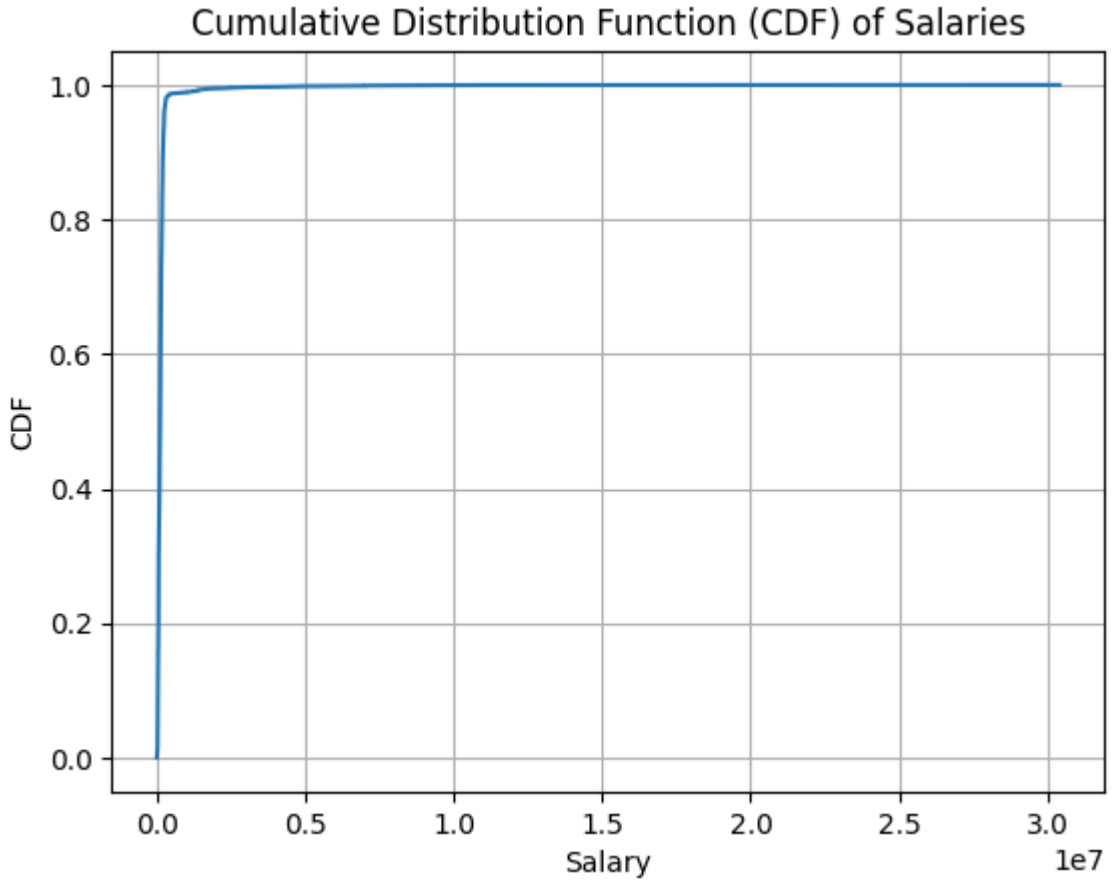
```
C:\Users\SBenavidez\PycharmProjects\pythonProject\venv\Scripts\thinkstats2.py:162: FutureWarning: iteritems
is deprecated and will be removed in a future version. Use .items instead.
  self.d.update(obj.value_counts().iteritems())
C:\Users\SBenavidez\PycharmProjects\pythonProject\venv\Scripts\thinkstats2.py:162: FutureWarning: iteritems
is deprecated and will be removed in a future version. Use .items instead.
  self.d.update(obj.value_counts().iteritems())
```





Create a CDF with one of your variables

In [34]:
```
# using 'Salary' from dproject 2 create CDF
salary_data = np.array(dproject2['Salary'])

salaries = np.sort(dproject2['Salary'])
cdf = np.arange(1, len(salaries) + 1) / len(salaries)

# plot CDF
plt.plot(salaries, cdf)
plt.xlabel('Salary')
plt.ylabel('CDF')
plt.title('Cumulative Distribution Function (CDF) of Salaries')
plt.grid(True)
plt.show()
```



From the CDF, you can see that it start a 0 goes up to 0.999, and tails to the right through the end.

In [35]:
```
# check probability of cdf
target_salary = 115000
index = np.searchsorted(salaries, target_salary)
if index >= len(salaries):
    probability = 1.0
else:
    probability = cdf[index]

print(f"The probability of having a salary <= {target_salary} is {probability:.2f}")
```

```
The probability of having a salary <= 115000 is 0.47
```

The probability of having a salary less than or equal to the target salary of 115,000 is 47%

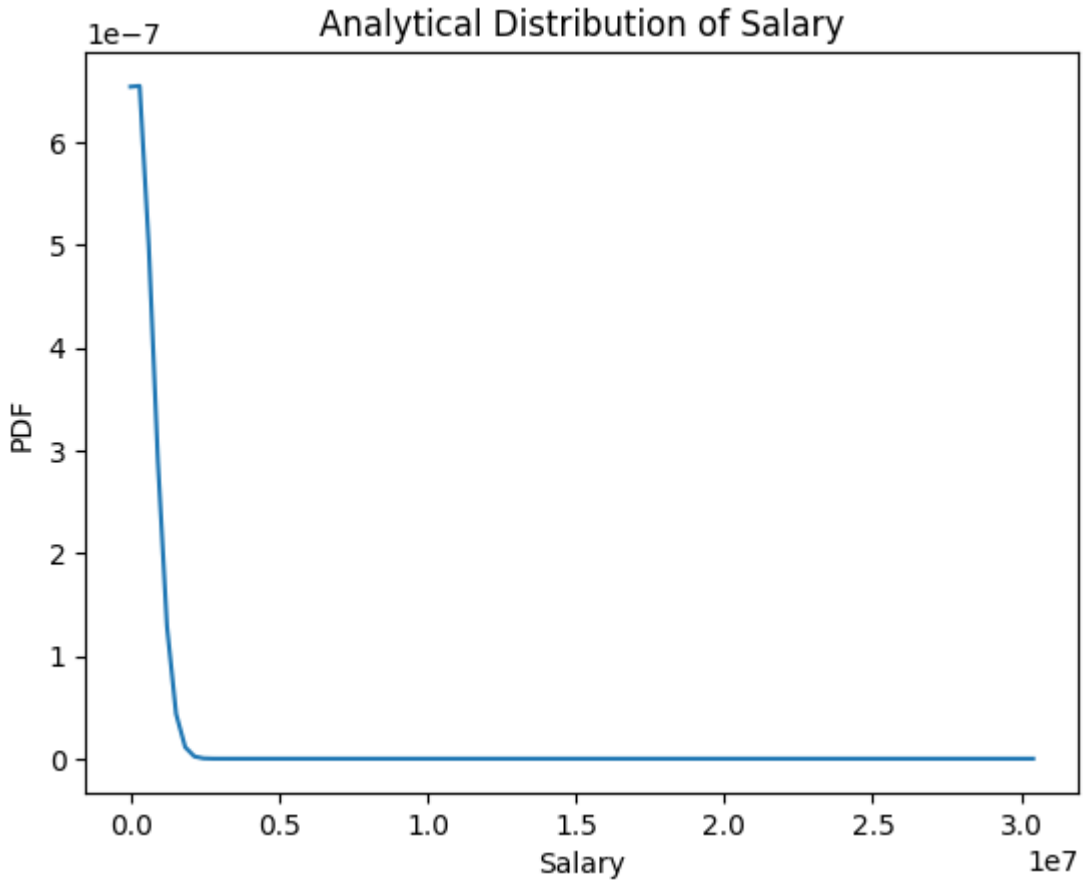
In [36]:
```
# calculate percentiles of 25 & 75 of target salary
s25th_percentile = np.percentile(dproject2['Salary'], 25)
s75th_percentile = np.percentile(dproject2['Salary'], 75)

print(f"The 25th percentile of the Salary is {s25th_percentile:.2f}")
print(f"The 75th percentile of the Salary is {s75th_percentile:.2f}")
```

```
The 25th percentile of the Salary is 68000.00
The 75th percentile of the Salary is 165000.00
```

The Salary in the 25th percentile is $68,000, and the 75th percentile is$ 165,000.

Plot 1 analytical distribution

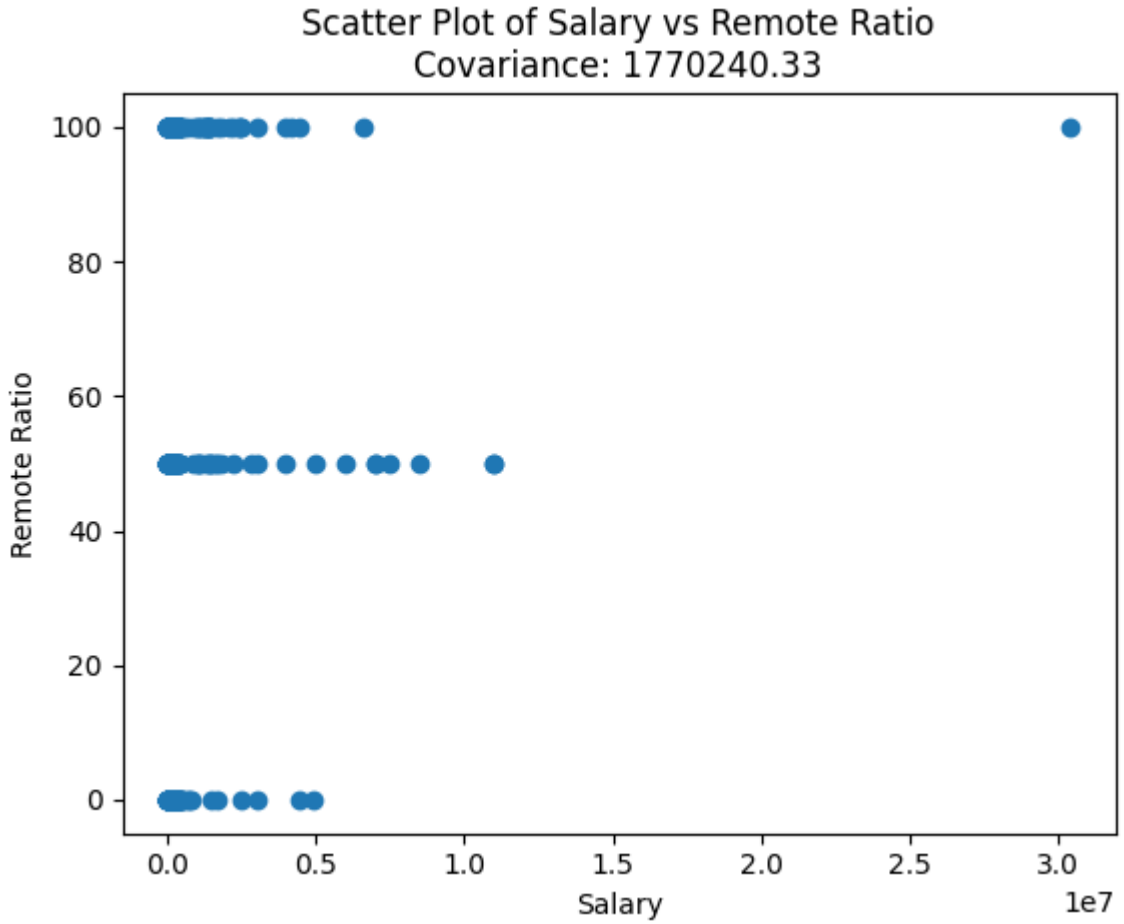
In [37]:
```
# calculate mean and std of salary_data
mean = salary_data.mean()
std = salary_data.std()

# generate x values
x = np.linspace(salary_data.min(), salary_data.max(), 100)

# calculate y values using PDF
y = norm.pdf(x, loc=mean, scale=std)

# plot the PDF
plt.plot(x, y)
plt.xlabel('Salary')
plt.ylabel('PDF')
plt.title('Analytical Distribution of Salary')

plt.show()
```



For the Analytical Distribution PDF of the Salary, you can see that it starts off at a PDF of 6, slopes down to 0.25 and tails to the right to 3.0.

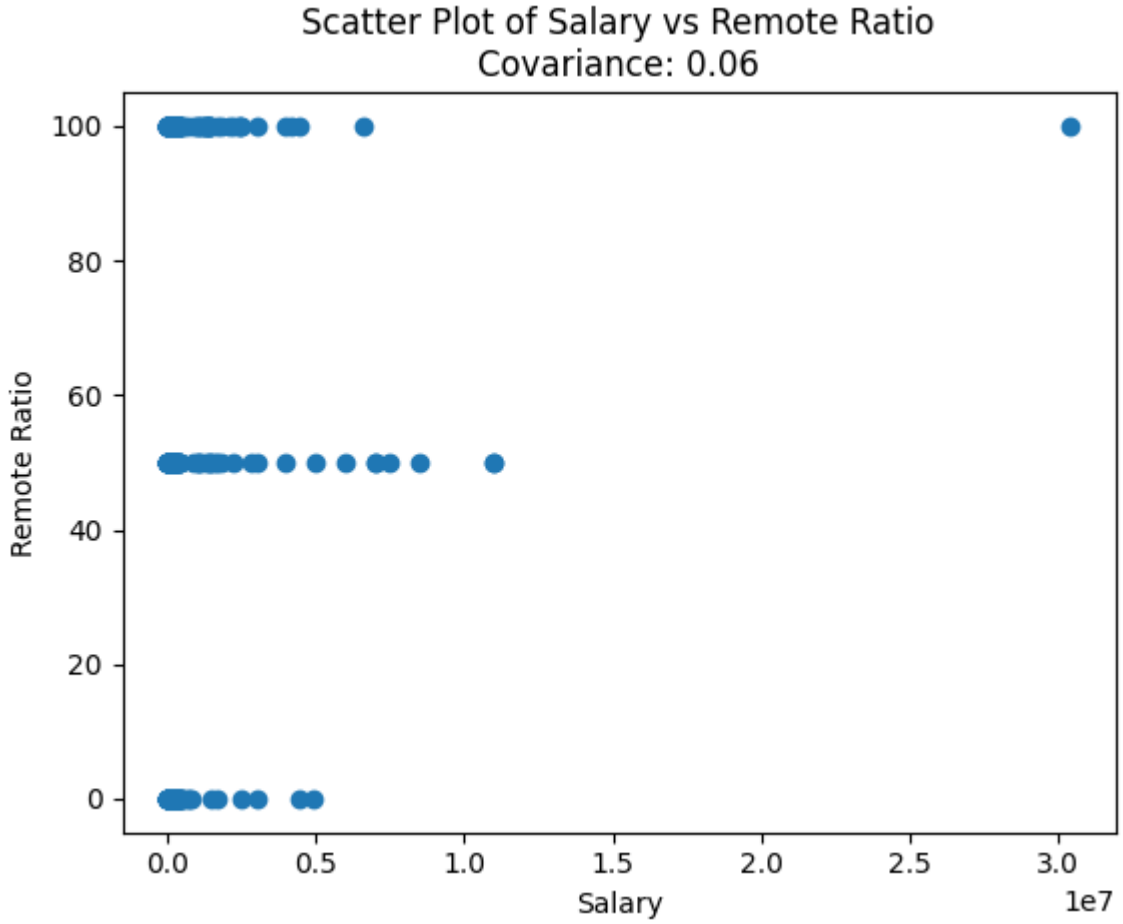
Create 2 scatter plots using covariance, Pearson's correlation, and Non-Linear Relationships

In [38]:
```
# use salary_data and create new variable called remote_data
remote_data = np.array(dproject2['Remote_Ratio'])

# calculate covariance of salary_data and remote_data
covariance = np.cov(salary_data, remote_data)[0, 1]

# create a scatter plot of covariance
plt.scatter(salary_data, remote_data)
plt.xlabel('Salary')
plt.ylabel('Remote Ratio')
plt.title('Scatter Plot of Salary vs Remote Ratio\nCovariance: {:.2f}'.format(covariance))

plt.show()
```



For the scatter plot on the covariance, you can see that the covariance of the Salary vs Remote Ratio is `1770240.33` . The range for the Remote Ratio is plotted at 0, 50, and 100.

In [39]:
```
# using salary_data and remote_data calculate Pearson's correlation
corr = np.corrcoef(salary_data, remote_data)[0, 1]

# create a scatter plot of Pearson's Correlation
plt.scatter(salary_data, remote_data)
plt.xlabel('Salary')
plt.ylabel('Remote Ratio')
plt.title('Scatter Plot of Salary vs Remote Ratio\nCovariance: {:.2f}'.format(corr))

plt.show()
```



For the scatter plot on the Pearson's correclation, you can see that the covariance of the Salary vs Remote Ratio is `0.06` . The range

In [40]:
```
# using salary_data and remote_data create a Non-Linear Relationship
log_data =  np.log(salary_data)

# create a scatter plot
plt.scatter(log_data, remote_data)
plt.xlabel('Log Salary')
plt.ylabel('Remote Ratio')
plt.title('Scatter Plot of Log Salary vs Remote Ratio')

plt.show()
```

For the scatter plot on the Non_linear Relationship, you can see that the log Salary vs Remote Ratio is a different range. The log range of the salary goes from 10-16, where the remote ratio is still at 0, 50, and 100.

Hypothesis Test

In [41]:
```
# define check_normality against salary_data and remote_data
def check_normality(salary_data, remote_data):
    test_stat_var, p_value_var= stats.levene(salary_data, remote_data)
    print("p value:%.4f" % p_value_var)
    if p_value_var <0.05:
        print("Reject null hypothesis >> The variances of the samples are different.")
    else:
        print("Fail to reject null hypothesis >> The variances of the samples are same.")

check_normality(salary_data,remote_data)
```

```
p value:0.0000
Reject null hypothesis >> The variances of the samples are different.
```

For the check_normality test, the pvalue is 0, and the hypothesis was rejected because the samples are different.

In [42]:
```
# checking for proper test to select on hypothesis
ttest,p_value = stats.ttest_ind(salary_data, remote_data)
print("p value:%.8f" % p_value)
print("since the hypothesis is one sided >> use p_value/2 >> p_value_one_sided:%.4f" %(p_value/2))
if p_value/2 <0.05:
    print("Reject null hypothesis")
else:
    print("Fail to reject null hypothesis")
```

```
p value:0.00000000
since the hypothesis is one sided >> use p_value/2 >> p_value_one_sided:0.0000
Reject null hypothesis
```

For checking for proper test on hypothesis, the pvalue is 0 again, and the hypothesis is rejected again.However, it is stating that since the hypothesis is one sided >> use p_value/2 >> p_value_one_sided:0.0000.

In [43]:
```
# create another distribution to check hypothesis
# define the null and alternative hypothesis
specific_distribution = 'norm'
H0 = f"The distribution of salaries is {specific_distribution}"
H1 = f"The distribution of salaries is different from {specific_distribution}"

# perform the Kolmogorov-Smirnov test
test_stat, p_value = stats.kstest(dproject2['Salary'], specific_distribution)

# set the significance level
alpha = 0.05

# interpret the results
if p_value < alpha:
    print(f"Reject H0: {H1}")
else:
    print(f"Fail to reject H0: {H0}")
```

```
Reject H0: The distribution of salaries is different from norm
```

For the last distribution to check the hypothesis, it rejected the istribution of salaries because they are different from the `norm` .

Conduct a Regression Analysis on one dependent and one explanatory variable

In [44]:
```
# define dependent and explanatory variable
Y = dproject2['Salary']
X = dproject2['Remote_Ratio']

# add constant term to X
X = sm.add_constant(X)

# fit linear regression model
model = sm.OLS(Y, X)
results = model.fit()

# print regression results
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.004
Model:                            OLS   Adj. R-squared:                  0.004
Method:                 Least Squares   F-statistic:                     20.41
Date:                Fri, 02 Jun 2023   Prob (F-statistic):           6.39e-06
Time:                        22:37:44   Log-Likelihood:                -72427.
No. Observations:                4926   AIC:                         1.449e+05
Df Residuals:                    4924   BIC:                         1.449e+05
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const         1.322e+05   1.05e+04     12.603      0.000    1.12e+05    1.53e+05
Remote_Ratio   809.2116    179.112      4.518      0.000     458.072    1160.351
==============================================================================
Omnibus:                    13333.376   Durbin-Watson:                   1.891
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        454127929.349
Skew:                          32.858   Prob(JB):                         0.00
Kurtosis:                    1489.015   Cond. No.                         73.4
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

For the Regression Analysis result summary, you can see that there are a total of 4926 number of observations, the date the observations are ran are 6/2/2023. The number of residuals are 4924, and there is only 1 model. The F-statistic is 20.41. The skew of the dataset is 32.858, and if remember earlier we did a skewness and that value was 32.844717. Which is a difference of 0.013283. The Omnibus test the skew and kurtosis, and based on the results the omnibus is high. The Durbin-Watson test for homoscdeasticity and is between 1 and 2 so is normal. The Jarque-Bera is like the Omnibus and tests off the skew and kurtosis, and should be a test confirmation of Omnibus, which in these results we don't because it is higher.

Summarize your Statistical/Hypothetical Questions from your Term Project in 250-500 Words

After completing all of the tasks for this course, I feel that the outcome of this EDA could have been better. I originally wanted to use a dataset from my company on different analyses of oil, gas, and water based on the location in the landing zone where the production volumes were produced. However, after going through the proper channels at work, they didn’t want the dataset information to be used due to confidentiality agreements within the company.

Since that process took longer than expected, I found a dataset that was relative to what I am going to school for. Relatively, we all want to do well in our future endeavors, so why not get a glance at what we would be looking forward to through a salary and job title perspective?

Throughout each analysis, I tried to plot, test, and distinguish the dataset as best as the dataset allowed. I placed each analysis under each task that was given. After going back over each task analysis, I feel that maybe the dataset should have had more numerical data to be able to give a better hypothesis or better statistics than given. This is all based on choosing bad datasets, which is going to be something learned throughout our journey in the data science world.

Though the dataset variables were not perfect, I do feel that having more variables that had more numerical data versus text data would have been more beneficial. What variables would have helped? With this dataset, it is hard to tell exactly which variables would have made the results better.

With this term project, I wanted to try to code the visualizations as I am learning them. Some of them didn’t turn out the way that I would have liked them to. The visualizations that I had pictured, I could have done possibly better in a PowerBI visualization, but I wanted to try and code through Python instead. Yes, there were some assumptions for the Location or Job Title that I may not have correctly analyzed. These assumptions were based on what the visualization represented best.

The biggest challenge was having the correct dataset, especially when you have a vision of one dataset, and then have to change it closer to the end of the course. Some of the other challenges faced with this project is not having enough data in the dataset to be able to properly get the results that you originally wanted.

In conclusion, I thought this was still neat to be able to perform all of these tasks with this finalized dataset even if the data didn’t fully represent the visualizations as I had hoped. Though I am looking forward to the continuous learning throughout each of these courses I am taking to help strengthen these visualizations so that they may represent the data as originally planned.