Data Science Job Salaries Dataset

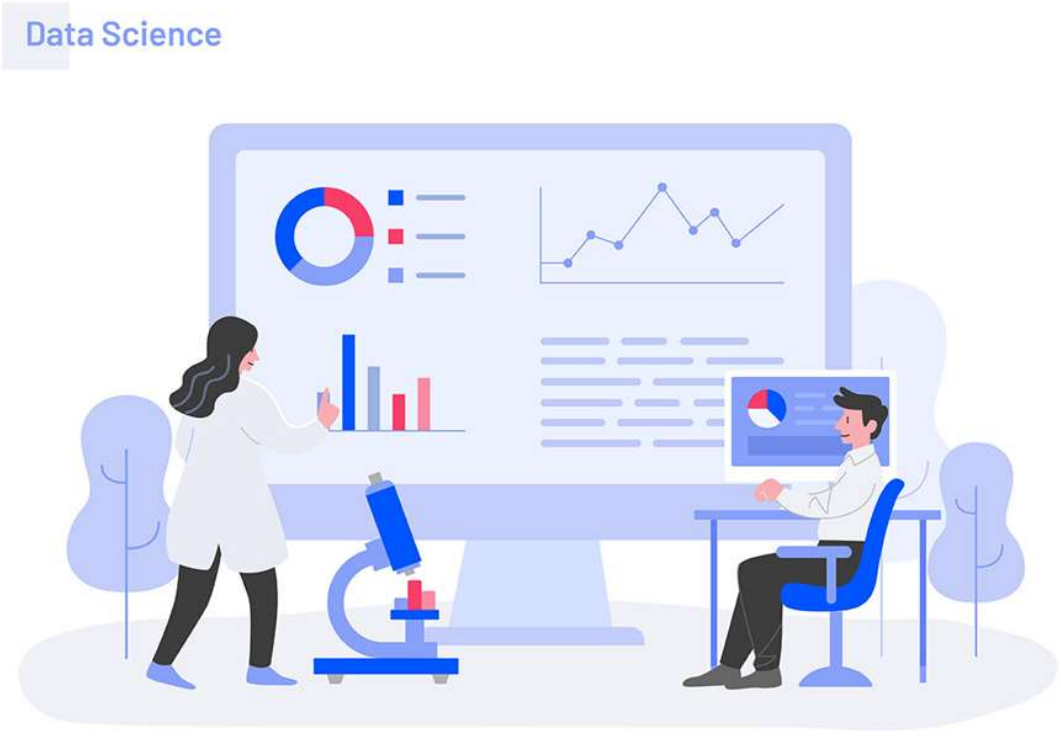
Data science jobs are some of the fastest-growing, most in-demand in technology. Since 2012, Data Scientist roles have increased by 650 percent, and this rise shows no sign of stopping. The U.S. Bureau of Labor Statistics predicts that the demand for data science skills will increase another 27.9 percent by 2026. And, according to a report from McKinsey, that spells a shortage of between 140,000 and 190,000 people with analytical skills in the U.S. alone—not to mention another 1.5 million managers and analysts who will be required to understand how data analysis drives decision-making.

Data Scientist salaries have also risen with demand; Data Scientists can typically expect to make six figures. Demand also translates into an ability to relocate far more easily—from city to city, and even internationally.

Problem Statement

Explore every features in the dataset to come up with actionable observations and overall recommendations which will benefit anyone/company going into the field of Data Science and Data Analysis.

- Work Year Analysis(with Salary, Remote Ratio)
- Experience Level Analysis (with Employment Type, Top 3 Job Title, Company Size)
- Company Location Analysis (with Experience Level)
- Salary Analysis (with Work Year, Experience Level, Company Size, Job Title, Remote Ratio)



In [79]:
```
#import pandas and numpy for data manipulation
import pandas as pd
import numpy as np

#import pyplot and seaborn for visualization
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

#import datetime for datetime manipulation
from datetime import datetime as dt
```

In [80]:
```
ds = pd.read_csv(r'C:\Users\user\OneDrive\Desktop\Data Science and ML\10Alytics\Interr
```

In [81]:
```
ds.head()
```

Out[81]:

	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	

In [4]:
```
ds.tail()
```

Out[4]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	sa
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	

Data Inspection and Manipulation

In [5]:
```
# the no, of rows and columns
ds.shape
```

Out[5]:
```
(607, 12)
```

In [6]:
```
# identify the columns in the dataset
ds.columns
```

Out[6]:
```
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 [7]:
```
# checking data types
ds.dtypes
```

Out[7]:
```
Unnamed: 0             int64
work_year              int64
experience_level      object
employment_type       object
job_title             object
salary                 int64
salary_currency       object
salary_in_usd          int64
employee_residence    object
remote_ratio           int64
company_location      object
company_size          object
dtype: object
```

In [8]:
```
# the info about the data
ds.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 607 entries, 0 to 606
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Unnamed: 0          607 non-null    int64 
 1   work_year           607 non-null    int64 
 2   experience_level    607 non-null    object
 3   employment_type     607 non-null    object
 4   job_title           607 non-null    object
 5   salary              607 non-null    int64 
 6   salary_currency     607 non-null    object
 7   salary_in_usd       607 non-null    int64 
 8   employee_residence  607 non-null    object
 9   remote_ratio        607 non-null    int64 
 10  company_location    607 non-null    object
 11  company_size        607 non-null    object
dtypes: int64(5), object(7)
memory usage: 57.0+ KB
```

In [9]:
```
# check the missing values
ds.isna().sum()
```

Out[9]:
```
Unnamed: 0            0
work_year             0
experience_level      0
employment_type       0
job_title             0
salary                0
salary_currency       0
salary_in_usd         0
employee_residence    0
remote_ratio          0
company_location      0
company_size          0
dtype: int64
```

In [10]:
```
ds.isnull()
```

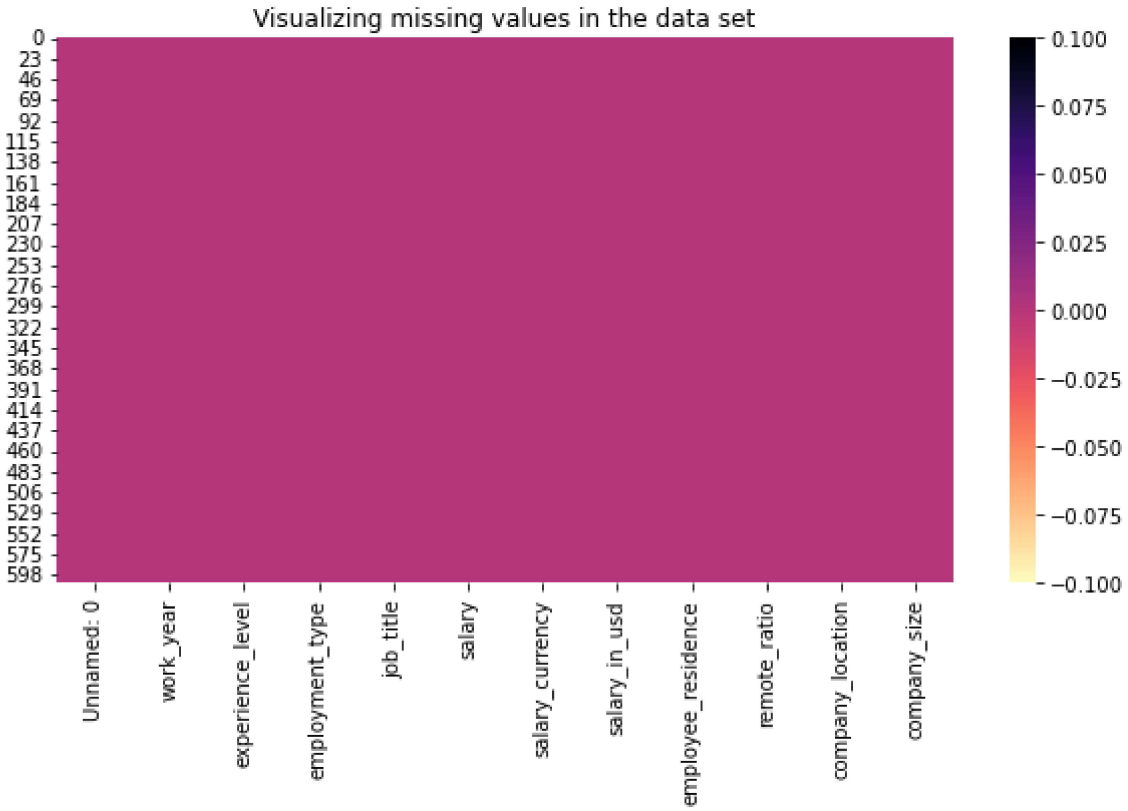
Out[10]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_currency	sal
0	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	
602	False	False	False	False	False	False	False	
603	False	False	False	False	False	False	False	
604	False	False	False	False	False	False	False	
605	False	False	False	False	False	False	False	
606	False	False	False	False	False	False	False	

607 rows × 12 columns

In [11]:
```
# Visualizing missing values
plt.figure(figsize = (10, 5))
plt.title('Visualizing missing values in the data set')
sns.heatmap(ds.isnull(),  cbar = True, cmap = 'magma_r')
```

Out[11]:
```
<AxesSubplot:title={'center':'Visualizing missing values in the data set'}>
```



In [12]:
```
# the stastical descriptive analysis of the numerical data
ds.describe().astype('int')
```

Out[12]:

	Unnamed: 0	work_year	salary	salary_in_usd	remote_ratio
count	607	607	607	607	607
mean	303	2021	324000	112297	70
std	175	0	1544357	70957	40
min	0	2020	4000	2859	0
25%	151	2021	70000	62726	50
50%	303	2022	115000	101570	100
75%	454	2022	165000	150000	100
max	606	2022	30400000	600000	100

In [13]:
```
ds.shape
```

Out[13]:
```
(607, 12)
```

Features in the data set and their meaning

work_year: the year the salary was paid

experience_level: EN-Entry level, MI-Junior Mid-level, SE-Intermediate Senior Level, EX-Expert Executive level, Director.

employment_type: PT-Part time, FT-Full time, CT-Contract, FL-Freelance.

job_title: role worked during the year.

salary: the total gross salary amount paid.

salary_currency: the currency of the salary paid as an ISO 4217 currency code.

salary_in_usd: the salary in USD, fx rate divided by average.

employee_residence: employee's primary country of residence code in during the year of work as an ISO 3166 country code.

remote_ratio:0-no remote work, 50-partially remote work, 100-fully remote work

company_location: employer's main office

company_size: S-less than 50 employees(small), M- < 250 and >50 employees(medium), and L->250 employees(large)

Exploratory Data Analysis: Relationship, insights and Visualizations

- Univariate Analysis
- Bivariate Analysis
- Multivariate Analysis

In [14]:
```
ds.head()
```

Out[14]:

	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	

In [15]:
```
# drop the 'Unnamed: 0' column
df = ds.drop(columns = 'Unnamed: 0', axis = 1)
```

```
df
```

Out[15]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd
0	2020	MI	FT	Data Scientist	70000	EUR	79833
1	2020	SE	FT	Machine Learning Scientist	260000	USD	260000
2	2020	SE	FT	Big Data Engineer	85000	GBP	109024
3	2020	MI	FT	Product Data Analyst	20000	USD	20000
4	2020	SE	FT	Machine Learning Engineer	150000	USD	150000
...	...	...	...	...	...	...	...
602	2022	SE	FT	Data Engineer	154000	USD	154000
603	2022	SE	FT	Data Engineer	126000	USD	126000
604	2022	SE	FT	Data Analyst	129000	USD	129000
605	2022	SE	FT	Data Analyst	150000	USD	150000
606	2022	MI	FT	AI Scientist	200000	USD	200000

607 rows × 11 columns

In [16]:
```
df.head()
```

Out[16]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	e
0	2020	MI	FT	Data Scientist	70000	EUR	79833	
1	2020	SE	FT	Machine Learning Scientist	260000	USD	260000	
2	2020	SE	FT	Big Data Engineer	85000	GBP	109024	
3	2020	MI	FT	Product Data Analyst	20000	USD	20000	
4	2020	SE	FT	Machine Learning Engineer	150000	USD	150000	

In [17]:
```
df.shape
```

Out[17]:
```
(607, 11)
```

In [18]:
```
# Experience level

def experience_bracket(experience_level):
    if experience_level == 'EN':
        return 'Entry level'
    elif experience_level == 'MI':
        return 'Junior Mid-level'
    elif experience_level == 'SE':
        return 'Senior Intermediate-level'
    elif experience_level == 'EX':
        return 'Executive Expert level'
    else:
        return 'Director'
```

In [19]:
```
# Create a new column for experience category
df['experience_category'] = df['experience_level'].apply(experience_bracket)
```

In [20]:
```
# Employment category

def employment_bracket(employment_type):
    if employment_type == 'PT':
        return 'Part time'
    elif employment_type == 'FT':
        return 'Full time'
    elif employment_type == 'CT':
        return 'Contract'
    else:
        return 'Freelance'
```

In [21]:
```
df['employment_category'] = df['employment_type'].apply(employment_bracket)
```

In [22]:
```
# remote ratio category

def remote_bracket(remote_ratio):
    if remote_ratio == 0:
        return 'No remote'
    elif remote_ratio == 50:
        return 'partially remote'
    else:
        return 'Fully remote'
```

In [23]:
```
df['remote-category'] = df['remote_ratio'].apply(remote_bracket)
```

In [57]:
```
# company size category

def company_size_bracket(company_size):
    if company_size == 'S':
        return 'Small'
    elif company_size == 'M':
        return 'Medium'
    else:
        return 'Large'
```

In [58]:
```
df['company_size_category'] = df['company_size'].apply(company_size_bracket)
```

In [59]:
```
df.head()
```

Out[59]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	e
0	2020	MI	FT	Data Scientist	70000	EUR	79833	
1	2020	SE	FT	Machine Learning Scientist	260000	USD	260000	
2	2020	SE	FT	Big Data Engineer	85000	GBP	109024	
3	2020	MI	FT	Product Data Analyst	20000	USD	20000	
4	2020	SE	FT	Machine Learning Engineer	150000	USD	150000	

In [60]:
```
df.tail()
```

Out[60]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd
602	2022	SE	FT	Data Engineer	154000	USD	154000
603	2022	SE	FT	Data Engineer	126000	USD	126000
604	2022	SE	FT	Data Analyst	129000	USD	129000
605	2022	SE	FT	Data Analyst	150000	USD	150000
606	2022	MI	FT	AI Scientist	200000	USD	200000

In [61]:
```
df.drop(df.columns[[1, 2, 8, 10]], axis = 1)
```

Out[61]:

	work_year	job_title	salary	salary_currency	salary_in_usd	employee_residence	company_locati
0	2020	Data Scientist	70000	EUR	79833	DE	
1	2020	Machine Learning Scientist	260000	USD	260000	JP	
2	2020	Big Data Engineer	85000	GBP	109024	GB	
3	2020	Product Data Analyst	20000	USD	20000	HN	I
4	2020	Machine Learning Engineer	150000	USD	150000	US	
...	...	...	...	...	...	...	
602	2022	Data Engineer	154000	USD	154000	US	
603	2022	Data Engineer	126000	USD	126000	US	
604	2022	Data Analyst	129000	USD	129000	US	
605	2022	Data Analyst	150000	USD	150000	US	
606	2022	AI Scientist	200000	USD	200000	IN	

607 rows × 11 columns

In [62]:
```
df.shape
```

Out[62]:
```
(607, 15)
```

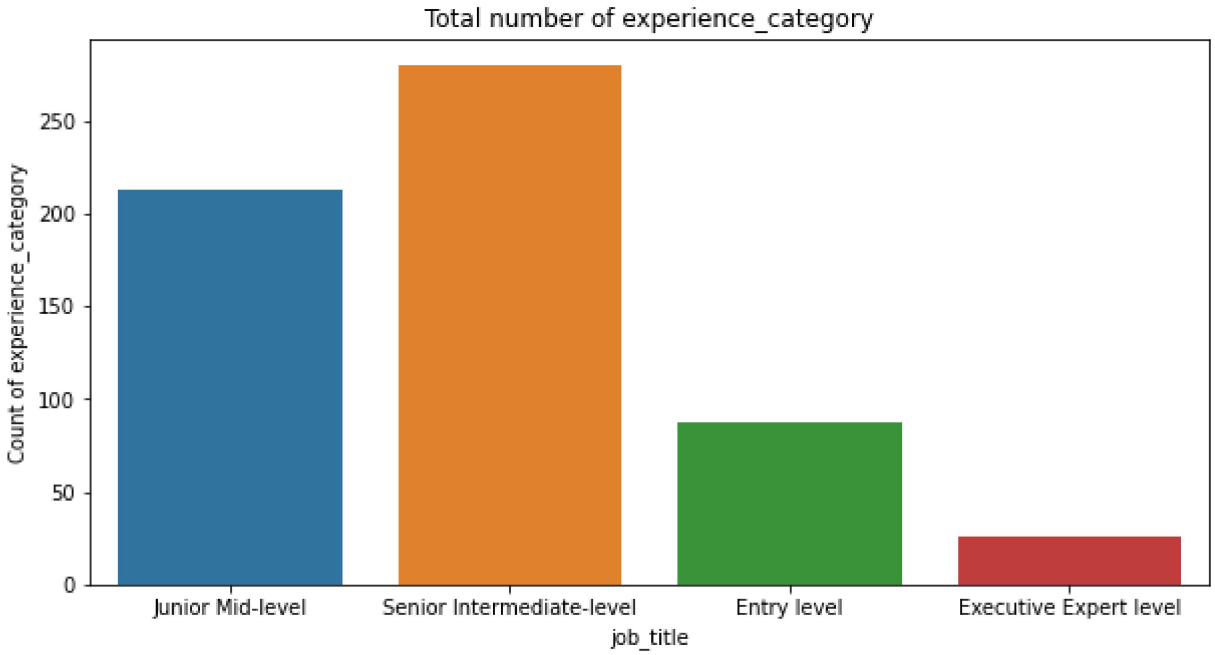
Univariate Analysis

In [98]:
```
df['work_year'].groupby(df.experience_category).count()
```

Out[98]:
```
experience_category
Entry level                  88
Executive Expert level       26
Junior Mid-level            213
Senior Intermediate-level   280
Name: work_year, dtype: int64
```

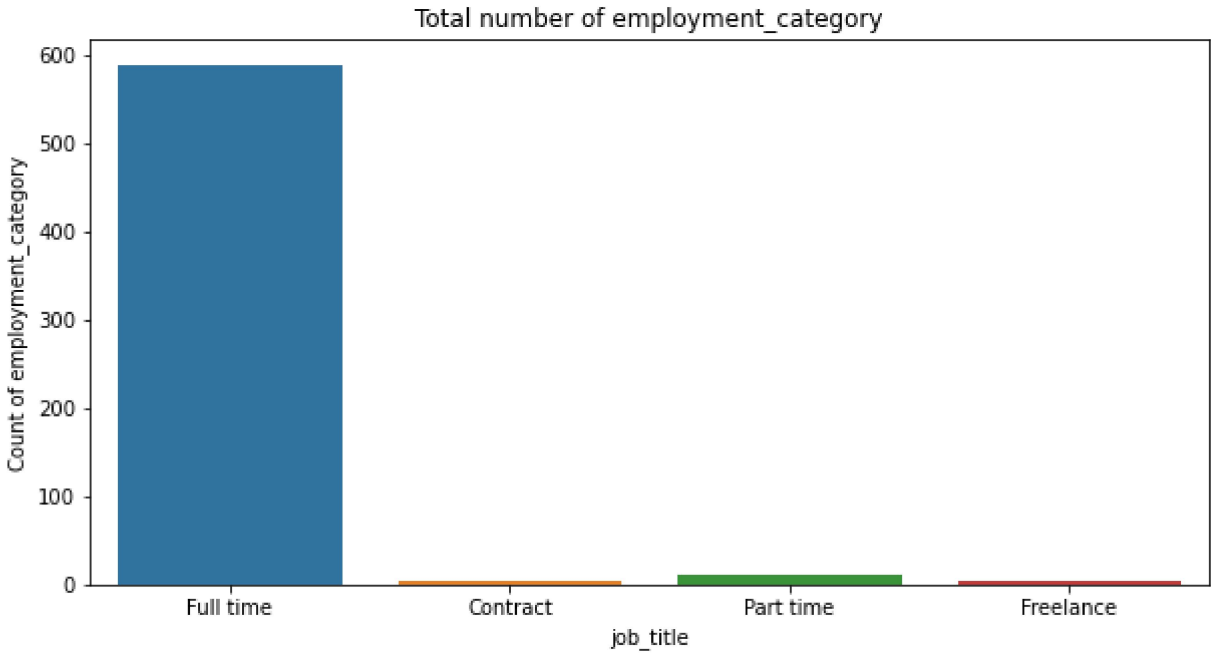
In [63]:
```
# Visualize experience_category
plt.figure(figsize=(10,5))
sns.countplot(x="experience_category", data=df)
plt.title("Total number of experience_category")
plt.xlabel("job_title")
plt.ylabel("Count of experience_category")
```

Out[63]:
```
Text(0, 0.5, 'Count of experience_category')
```



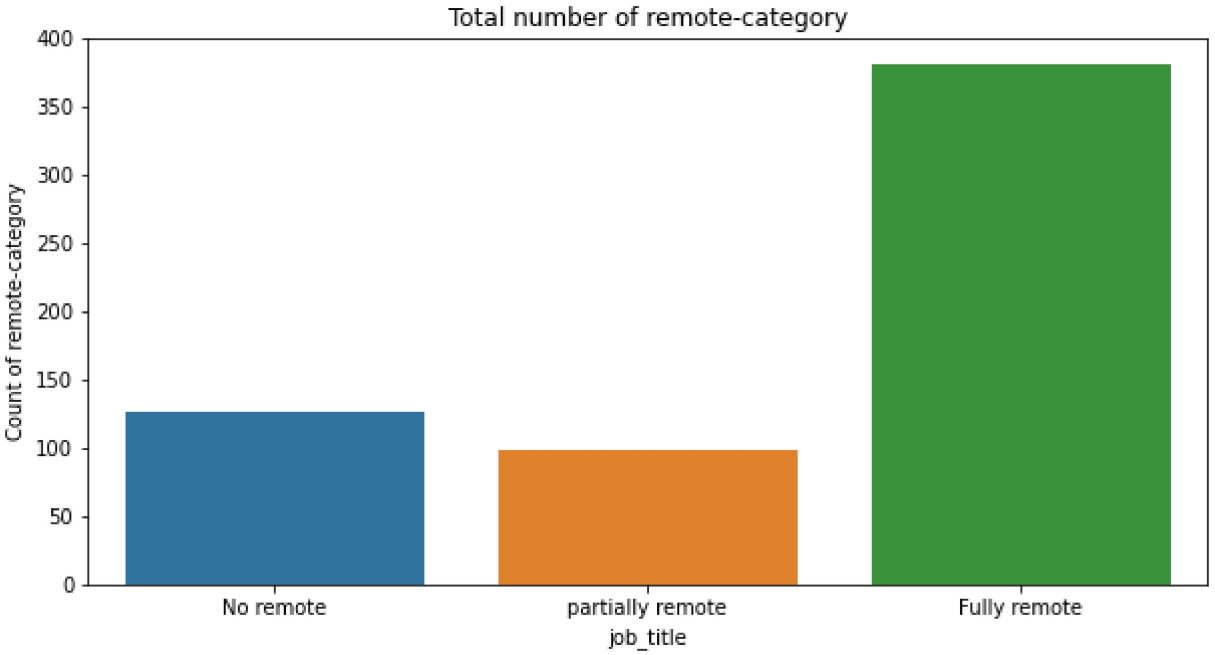
In [64]:
```
# Visualize employment_category
plt.figure(figsize=(10,5))
sns.countplot(x="employment_category", data=df)
plt.title("Total number of employment_category")
plt.xlabel("job_title")
plt.ylabel("Count of employment_category")
```

Out[64]:
```
Text(0, 0.5, 'Count of employment_category')
```



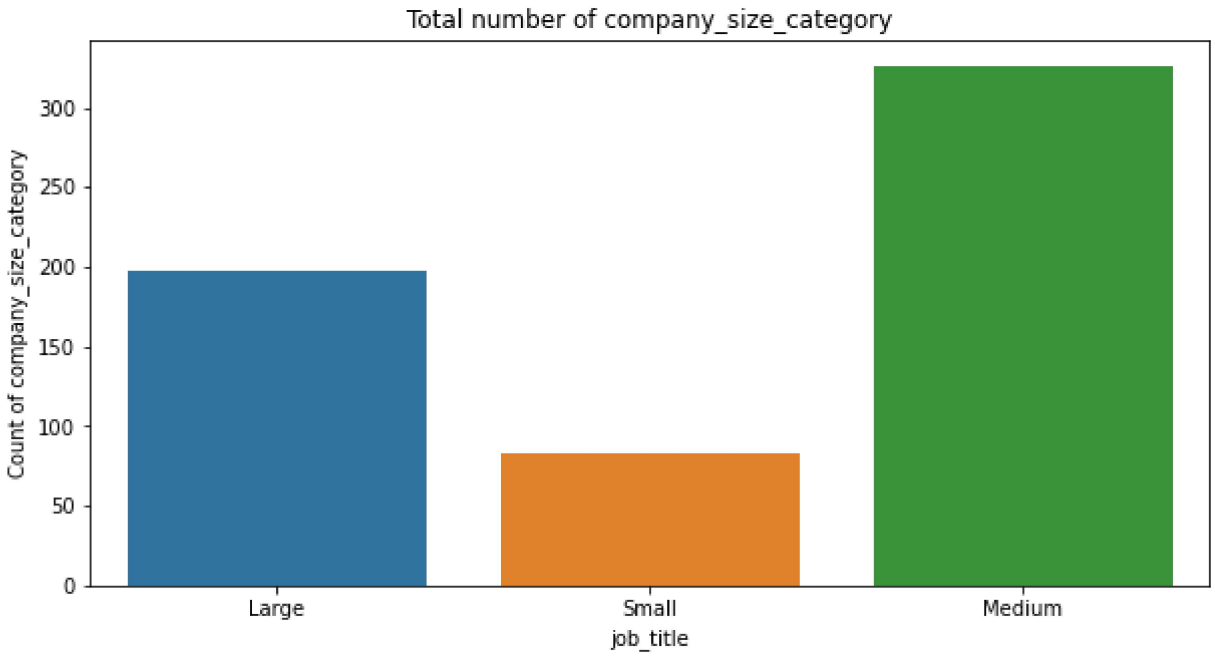
In [65]:
```
# Visualize remote-category
plt.figure(figsize=(10,5))
sns.countplot(x="remote-category", data=df)
plt.title("Total number of remote-category")
plt.xlabel("job_title")
plt.ylabel("Count of remote-category")
```

Out[65]:
```
Text(0, 0.5, 'Count of remote-category')
```



In [66]:
```
# Visualize company_size_category
plt.figure(figsize=(10,5))
sns.countplot(x="company_size_category", data=df)
plt.title("Total number of company_size_category")
plt.xlabel("job_title")
plt.ylabel("Count of company_size_category")
```

Out[66]:
```
Text(0, 0.5, 'Count of company_size_category')
```



In [67]:
```
# Total Salary for the year
df.salary.sum()
```

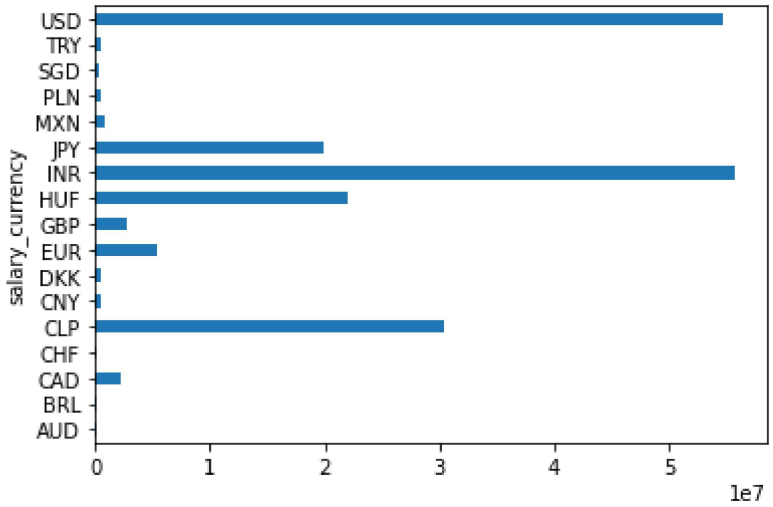
Out[67]:
```
196668038
```

In [75]:
```
# Total salary for the year by salary currency
df_salary_currency = df['salary'].groupby(df.salary_currency).sum()
df_salary_currency
```

Out[75]:
```
salary_currency
AUD      241000
BRL      171600
CAD     2214500
CHF      115000
CLP    30400000
CNY      539000
DKK      480000
EUR     5441699
GBP     2712856
HUF    22000000
INR    55734997
JPY    19950000
MXN      778000
PLN      440000
SGD      280000
TRY      538000
USD    54631386
Name: salary, dtype: int64
```

In [76]:
```
df_salary_currency.plot.barh()
```

Out[76]:
```
<AxesSubplot:ylabel='salary_currency'>
```



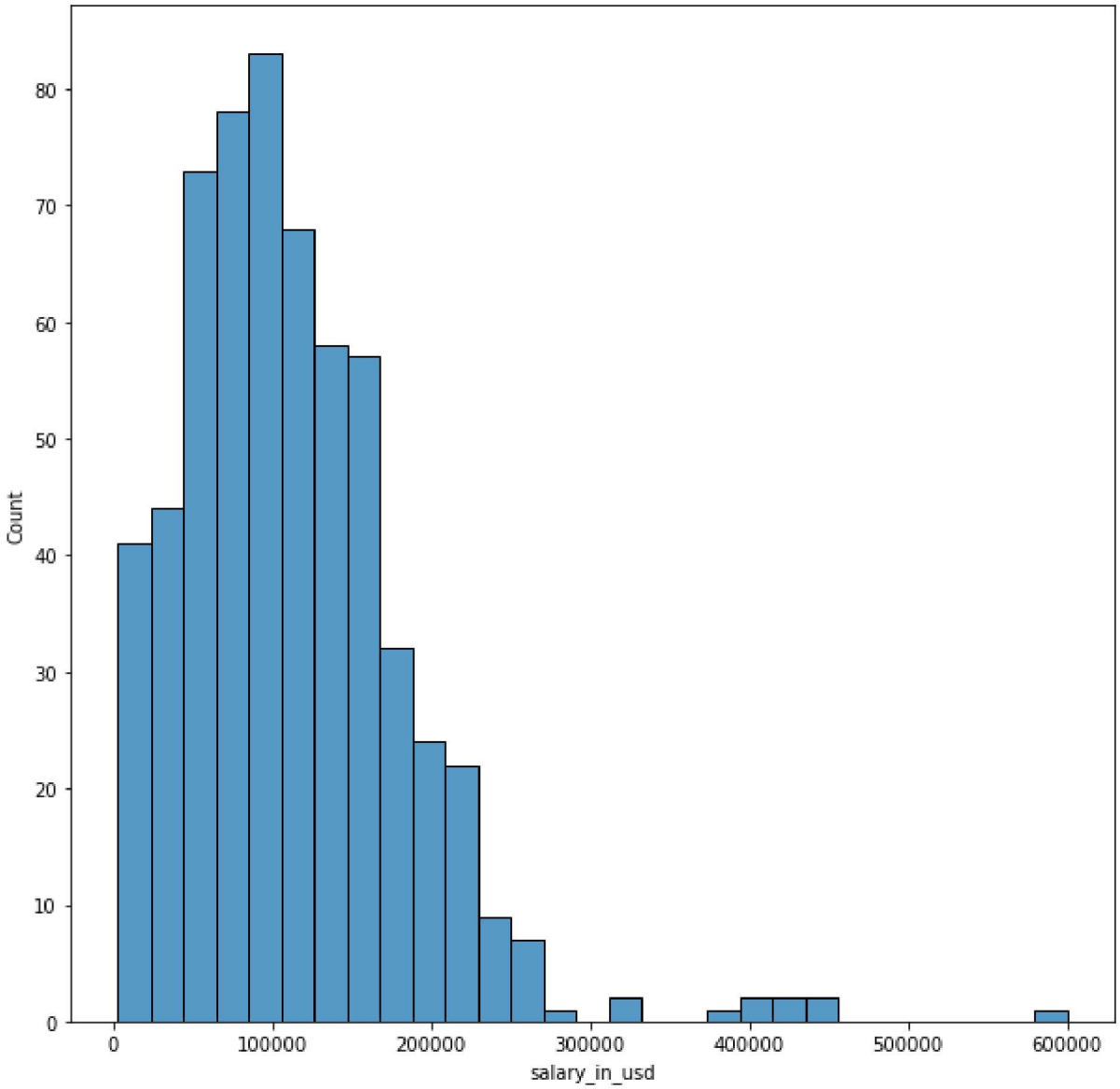
Observation

- it is observed that the employees were paid most with INR followed by USD compare to other salary currency paid for the year.

In [248…]
```
# Numerical Data - Histogram

fig, axes = plt.subplots(figsize = (10, 10))
sns.histplot(x='salary_in_usd', data=df)
```

Out[248]:
```
<AxesSubplot:xlabel='salary_in_usd', ylabel='Count'>
```



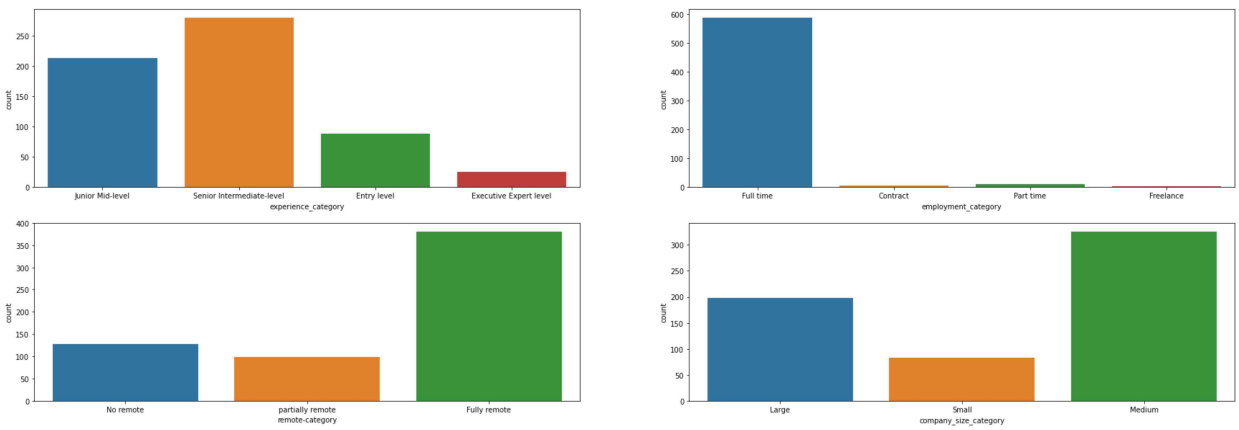
In [250…

```
# Analysis, we will create a subplot

fig, axes = plt.subplots(2, 2, figsize = (30, 10))
sns.countplot(x='experience_category', data=df, ax=axes[0,0])
sns.countplot(x='employment_category', data=df, ax=axes[0,1])
sns.countplot(x='remote-category', data=df, ax=axes[1,0])
sns.countplot(x='company_size_category', data=df, ax=axes[1,1])
```

Out[250]:

```
<AxesSubplot:xlabel='company_size_category', ylabel='count'>
```



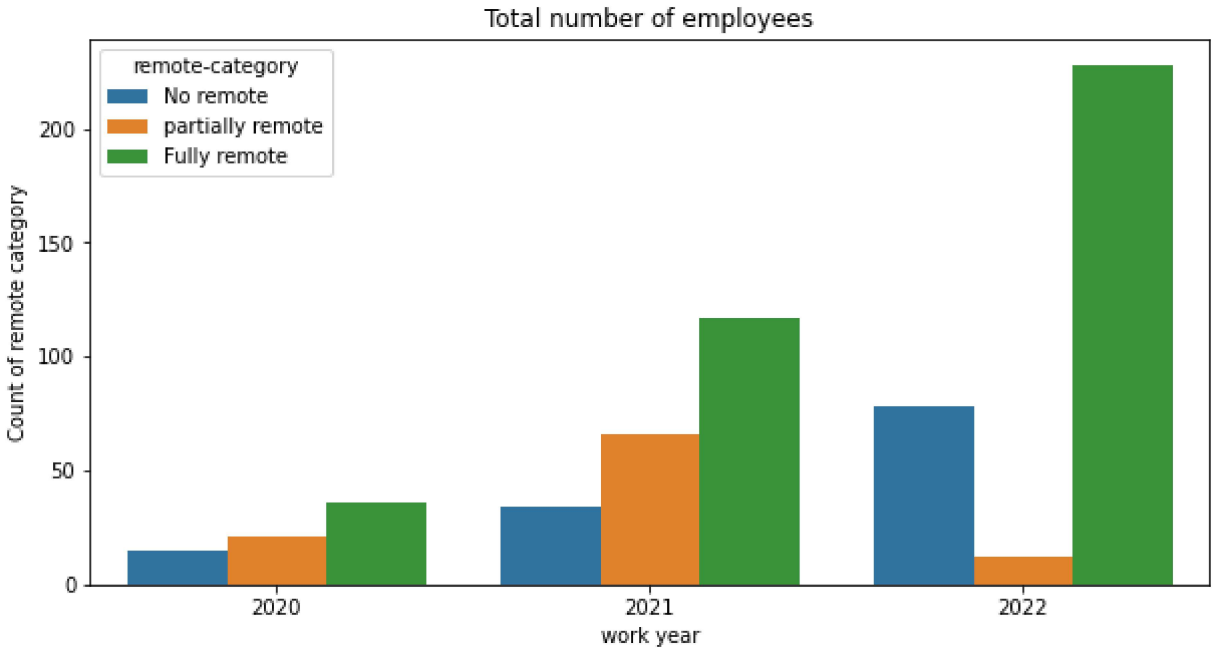
Bivariate Analysis

- Experience level
- Work Year Analysis
- Salary Analysis

In [100…
```
# Visualize remote ratio by work year

plt.figure(figsize=(10,5))
sns.countplot(x="work_year", data=df, hue="remote-category")
plt.title("Remote ratio by work year")
plt.xlabel("work year")
plt.ylabel("Count of remote category")
```

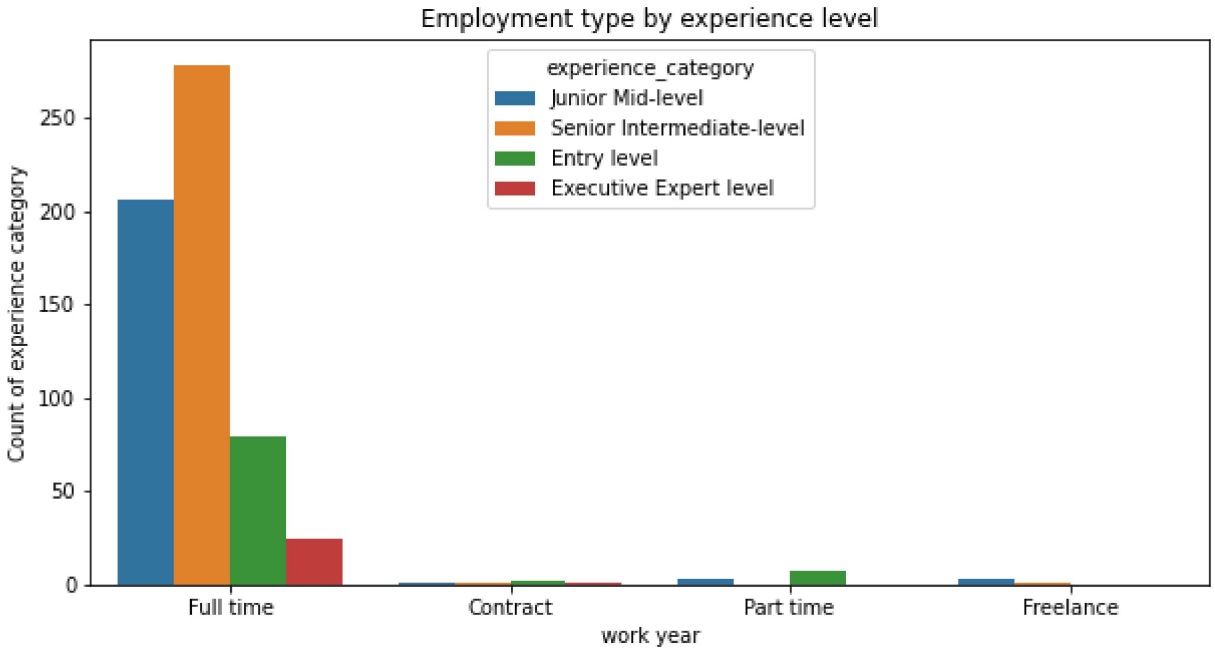
Out[100]:
```
Text(0, 0.5, 'Count of remote category')
```



In [102…
```
# Visualize employment type by experience level

plt.figure(figsize=(10,5))
sns.countplot(x="employment_category", data=df, hue="experience_category")
plt.title("Employment type by experience level")
plt.xlabel("work year")
plt.ylabel("Count of experience category")
```

Out[102]:
```
Text(0, 0.5, 'Count of experience category')
```



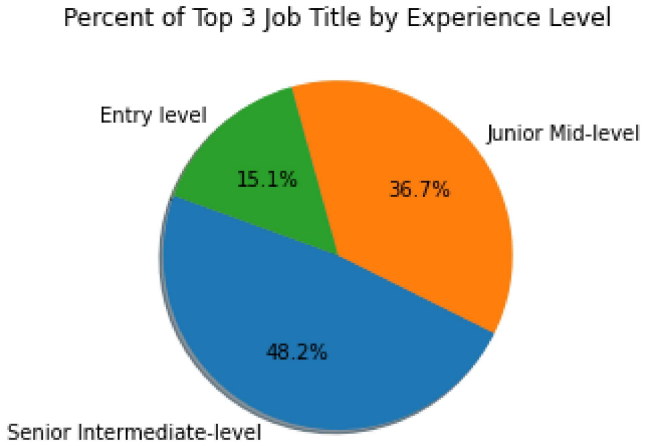
In [114…
```
# Top 3 Job Title by Experience Level
top_3 = (df.groupby('experience_category').job_title.count().sort_values(ascending=Fal
top_3
```

Out[114]:
```
experience_category
Senior Intermediate-level    280
Junior Mid-level             213
Entry level                   88
Name: job_title, dtype: int64
```

In [115…
```
# Top 3 Job Title by Experience Level
colors = ['#1f77b4', '#ff7f0e','#2ca02c', '#d62728', '#8c564b']
explode = (0.1, 0, 0, 0, 0)
plt.pie(top_3, labels = top_3.index  , colors = colors,
        autopct = '%1.1f%%', shadow = True, startangle = 160)

plt.title('Percent of Top 3 Job Title by Experience Level')

plt.show()
```



In [120…
```
# Company Size by Experience Level
cse = df['experience_category'].groupby(df.company_size_category).count()
```

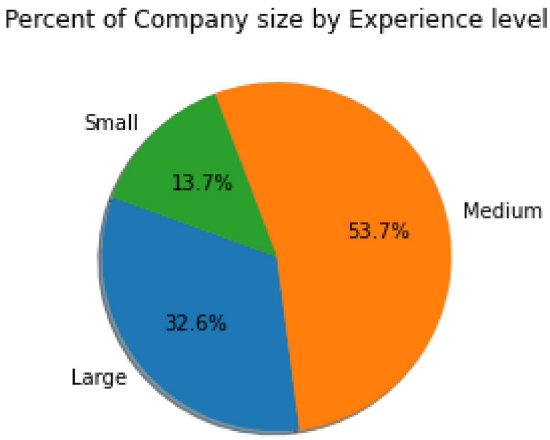
```
cse
```

Out[120]:
```
company_size_category
Large     198
Medium    326
Small      83
Name: experience_category, dtype: int64
```

In [125…
```
# View percentage distribution of company Size by Experience Level
colors = ['#1f77b4', '#ff7f0e','#2ca02c', '#d62728', '#8c564b']
explode = (0.1, 0, 0, 0, 0)
plt.pie(cse, labels = cse.index  , colors = colors,
        autopct = '%1.1f%%', shadow = True, startangle = 160)

plt.title('Percent of Company size by Experience level')

plt.show()
```

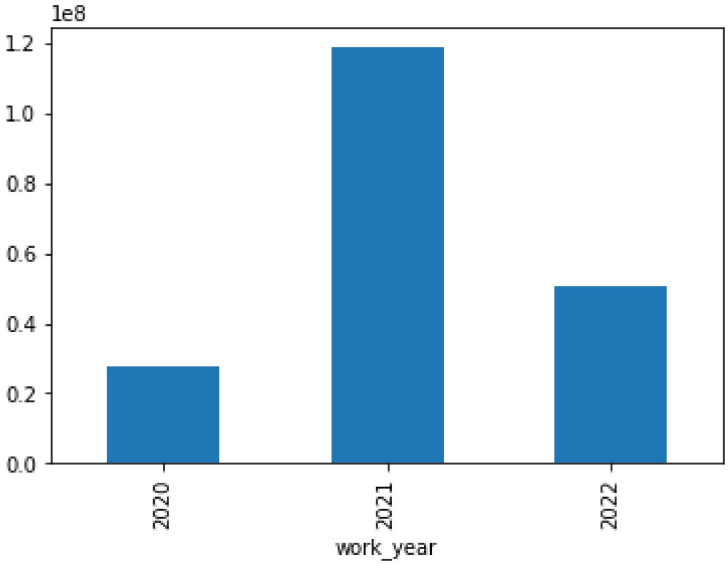


In [173…
```
# sum of salary by work year
sw = df['salary'].groupby(df.work_year).sum().astype(int)
sw
```

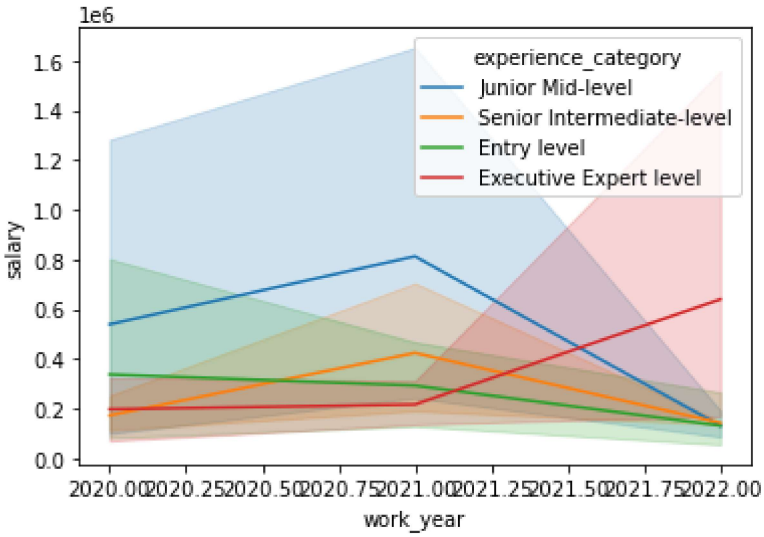
Out[173]:
```
work_year
2020     27531809
2021    118628993
2022     50507236
Name: salary, dtype: int32
```

In [174…
```
sw.plot.bar()
```

Out[174]:
```
<AxesSubplot:xlabel='work_year'>
```



In [164…
```
# salary by experience level
sns.lineplot(data = df, x = 'work_year', y = 'salary', hue = 'experience_category')
plt.show()
```

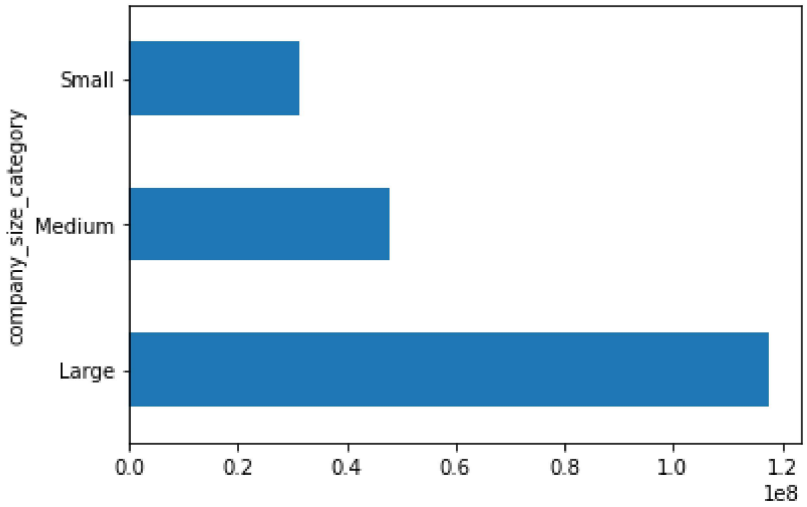


In [168…
```
# salary by company size
sc = df['salary'].groupby(df.company_size_category).sum()
sc
```

Out[168]:
```
company_size_category
Large     117551772
Medium     47766335
Small      31349931
Name: salary, dtype: int64
```

In [171…
```
sc.plot.barh()
```

Out[171]:
```
<AxesSubplot:ylabel='company_size_category'>
```



In [195…

```
# Highest salaries by job title
hs = df['salary'].groupby(df.job_title).max().sort_values(ascending = False)
hs
```

Out[195]:

```
job_title
Data Scientist                                30400000
BI Data Analyst                               11000000
ML Engineer                                    8500000
Data Science Manager                           7000000
Head of Machine Learning                       6000000
Machine Learning Engineer                      4900000
Data Engineer                                  4450000
Lead Data Scientist                            3000000
Big Data Engineer                              1672000
Lead Data Analyst                              1450000
Business Data Analyst                          1400000
AI Scientist                                   1335000
Principal Data Engineer                         600000
Product Data Analyst                            450000
Financial Data Analyst                          450000
Data Analyst                                    450000
Research Scientist                              450000
Applied Machine Learning Scientist              423000
Data Science Consultant                         423000
Principal Data Scientist                        416000
Data Analytics Lead                             405000
3D Computer Vision Researcher                   400000
Applied Data Scientist                          380000
Director of Data Science                        325000
Lead Data Engineer                              276000
Data Architect                                  266400
Machine Learning Scientist                      260000
NLP Engineer                                    240000
Head of Data                                    235000
Head of Data Science                            224000
Analytics Engineer                              205300
Director of Data Engineering                    200000
Machine Learning Infrastructure Engineer        195000
Computer Vision Engineer                        180000
Data Engineering Manager                        174000
Principal Data Analyst                          170000
Data Specialist                                 165000
Cloud Data Engineer                             160000
Data Science Engineer                           159500
Machine Learning Manager                        157000
Data Analytics Manager                          150260
Computer Vision Software Engineer               150000
Big Data Architect                              125000
Data Analytics Engineer                         110000
Staff Data Scientist                            105000
Machine Learning Developer                      100000
Lead Machine Learning Engineer                   80000
Marketing Data Analyst                           75000
ETL Developer                                    50000
Finance Data Analyst                             45000
Name: salary, dtype: int64
```

Observation

- Data Scientist is the highest paid in terms of salary followed by BI Data Analyst compare to others.

In [204…]:
```
# Average Salary by Company Location
df['salary'].groupby(df.company_location).mean().astype(int).sort_values(ascending = F
```

Out[204]:
```
company_location
CL    30400000
HU    11000000
JP     3408666
IN     2065208
AS     1335000
MX      279333
CH      275000
CN      199500
US      187715
DK      185000
TR      179333
IL      160000
RU      157500
PL      142500
AU      130333
NZ      125000
SG      120000
CA      115306
IQ      100000
AE      100000
DZ      100000
BE       72500
DE       71289
BR       65200
IE       65000
AT       64000
GB       62294
RO       60000
FR       56881
SI       54000
CZ       49499
NL       49100
ES       47382
GR       47363
PT       42100
MY       40000
HR       40000
LU       38333
IT       31100
NG       30000
EE       30000
MT       24000
CO       21844
HN       20000
MD       18000
UA       13400
PK       13333
KE        9272
IR        4000
VN        4000
Name: salary, dtype: int32
```

Observation

- Company location with the highest average paid salary to employees is CL followed by HU.

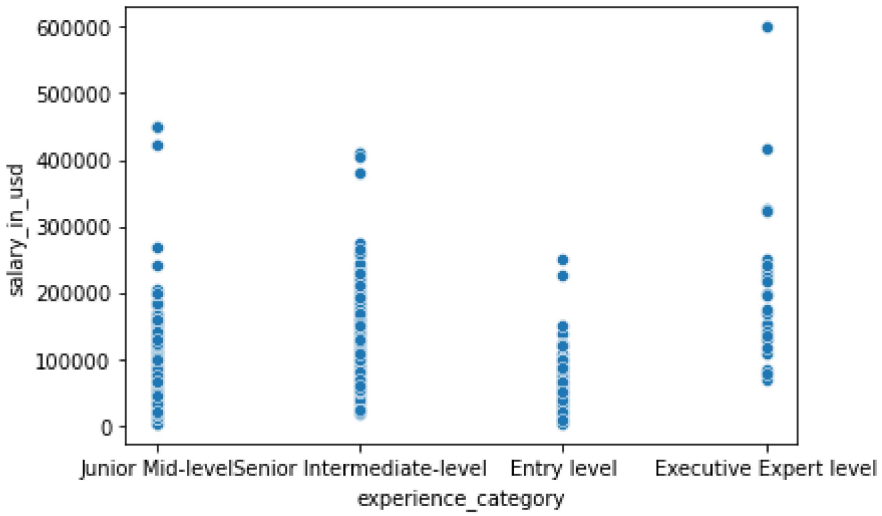
In [205…
```
df.columns
```

Out[205]:
```
Index(['work_year', 'experience_level', 'employment_type', 'job_title',
       'salary', 'salary_currency', 'salary_in_usd', 'employee_residence',
       'remote_ratio', 'company_location', 'company_size',
       'experience_category', 'employment_category', 'remote-category',
       'company_size_category'],
      dtype='object')
```

In [253…
```
# Numerical vs Numerical

sns.scatterplot(x = df['experience_category'], y = df['salary_in_usd'])
```

Out[253]:
```
<AxesSubplot:xlabel='experience_category', ylabel='salary_in_usd'>
```

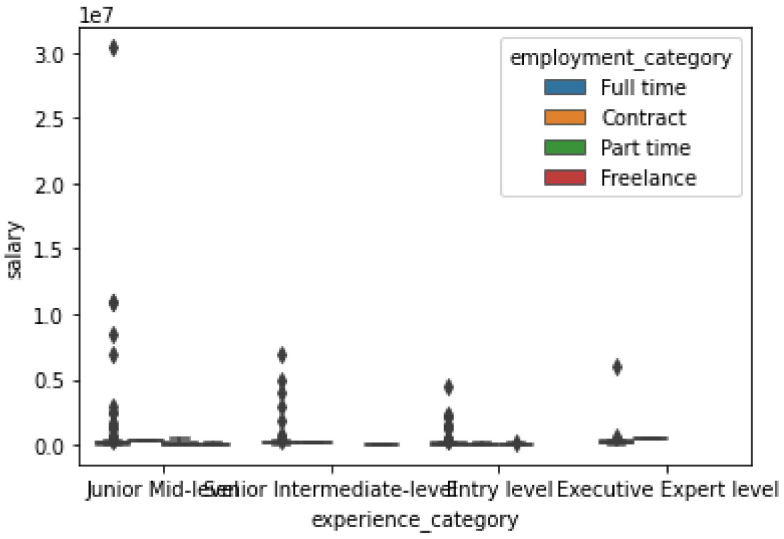


Multivariate Analysis

- comparing more than one features

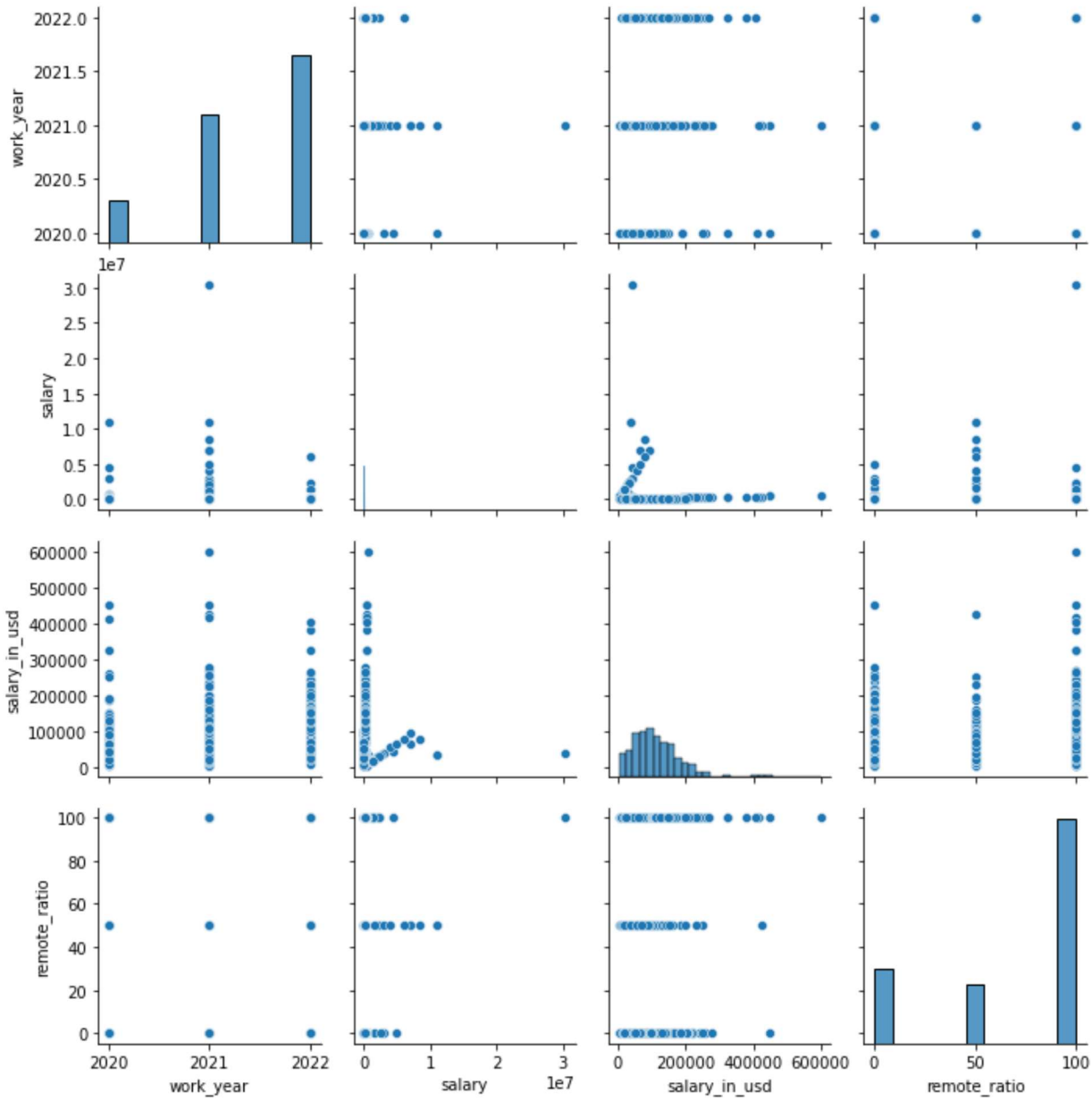
In [261…
```
sns.boxplot(x = 'experience_category', y = 'salary', data = df, hue = 'employment_cate
```

Out[261]:
```
<AxesSubplot:xlabel='experience_category', ylabel='salary'>
```



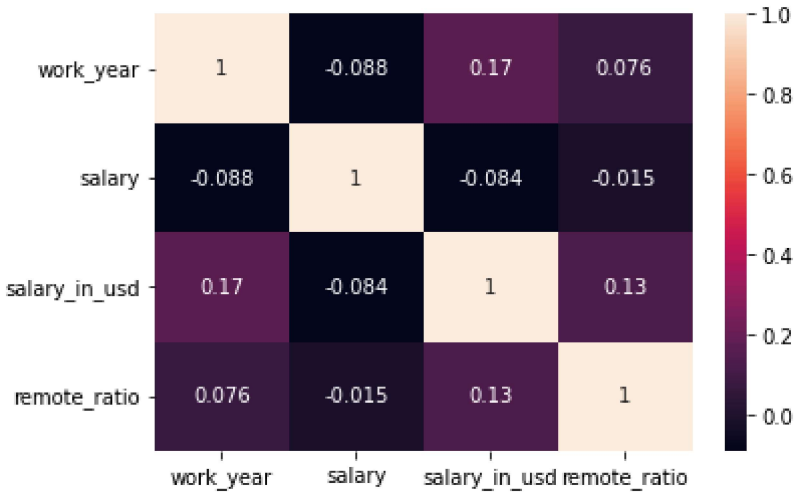
In [268…]
```
sns.pairplot(data = df)
```

Out[268]:
```
<seaborn.axisgrid.PairGrid at 0x12692810430>
```



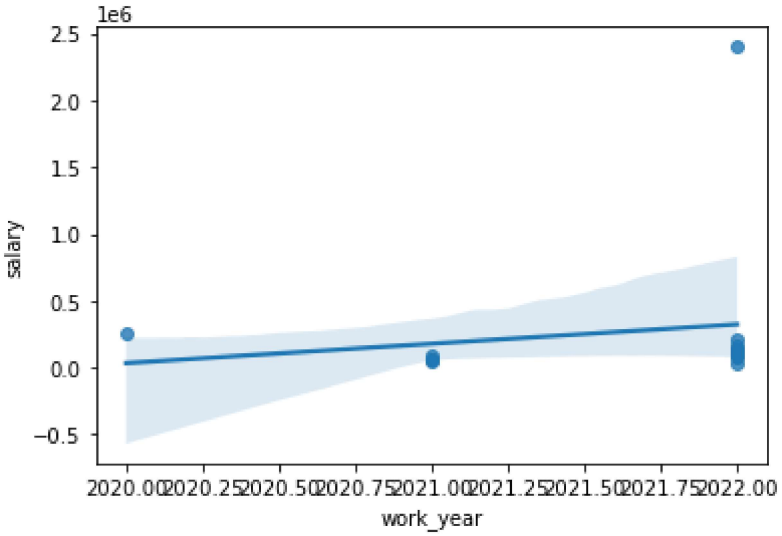
In [269…]
```
corel = df.corr()
sns.heatmap(corel, annot = True)
```

Out[269]:
```
<AxesSubplot:>
```



In [271…
```
# simple regression of work year and salary
simple_lin_reg = df[['work_year', 'salary']].sample(15, random_state = 2)
# regression plot
sns.regplot(x = 'work_year', y = 'salary', data = simple_lin_reg)
```

Out[271]:
```
<AxesSubplot:xlabel='work_year', ylabel='salary'>
```



In [275…
```
from sklearn.metrics import mean_squared_error
y_true = df.work_year
y_pred = df.salary

mean_squared_error(y_true, y_pred)
```

Out[275]:
```
2484781263443.247
```

Observations

- it is observed that there is no strong correlation between work year salary, which tends toward 0 with the slope(line of best fit) being negative. That is, the line of regression is negative with mean square errors being high; which shows that there is a large vertical distance between work_year and salary data.

- salary in USD and work year shows that there is positive correlation between the two, and that means that based on the work year between 2020 and 2022, employees are handsomely pay well in USD as salary across all company locations.
- work year and remote ratio isn't bad, as this also show a slight positive correlation and that means that companies at different location are beginning to adopt the remote kind of work, which spiked in 2022 as Most companies are fully going remote.

Overall Recommendations

- Companies across the locations can explore taking employees as freelance while reducing their overhead and running cost of the business
- Based on the first, they can as well work fully remotely while delivering their jobs which is basically the trend in reference to the historical data given and the analysis is done, and this will help company of small size to scale to larger company while generating more revenues.
- Companies with small size should invest in their current employees in terms of technical knowledge in the field of Data science and BI Data Analysis, and embrace more of entry level Data Scientist and Data Analyst so that they can grow along with the company while learning under the guidianship of the invested and trained employees.

In [ ]: