WOMEN TECHSTERS FELLOWSHIP (CLASS OF 2024)

MONTH 3 PROJECT - DATA ANALYSIS PROJECT

DATA SCIENCE AND ARTIFICIAL INTELLIGENCE GROUP C

STUDY GROUP 2 (TEAM 8)

TEAM 8 MEMBERS

1. OPEYEMI AWE
2. OLAWUMI SALAAM
3. JOY ALABI
4. MARIAN KUTSOATI

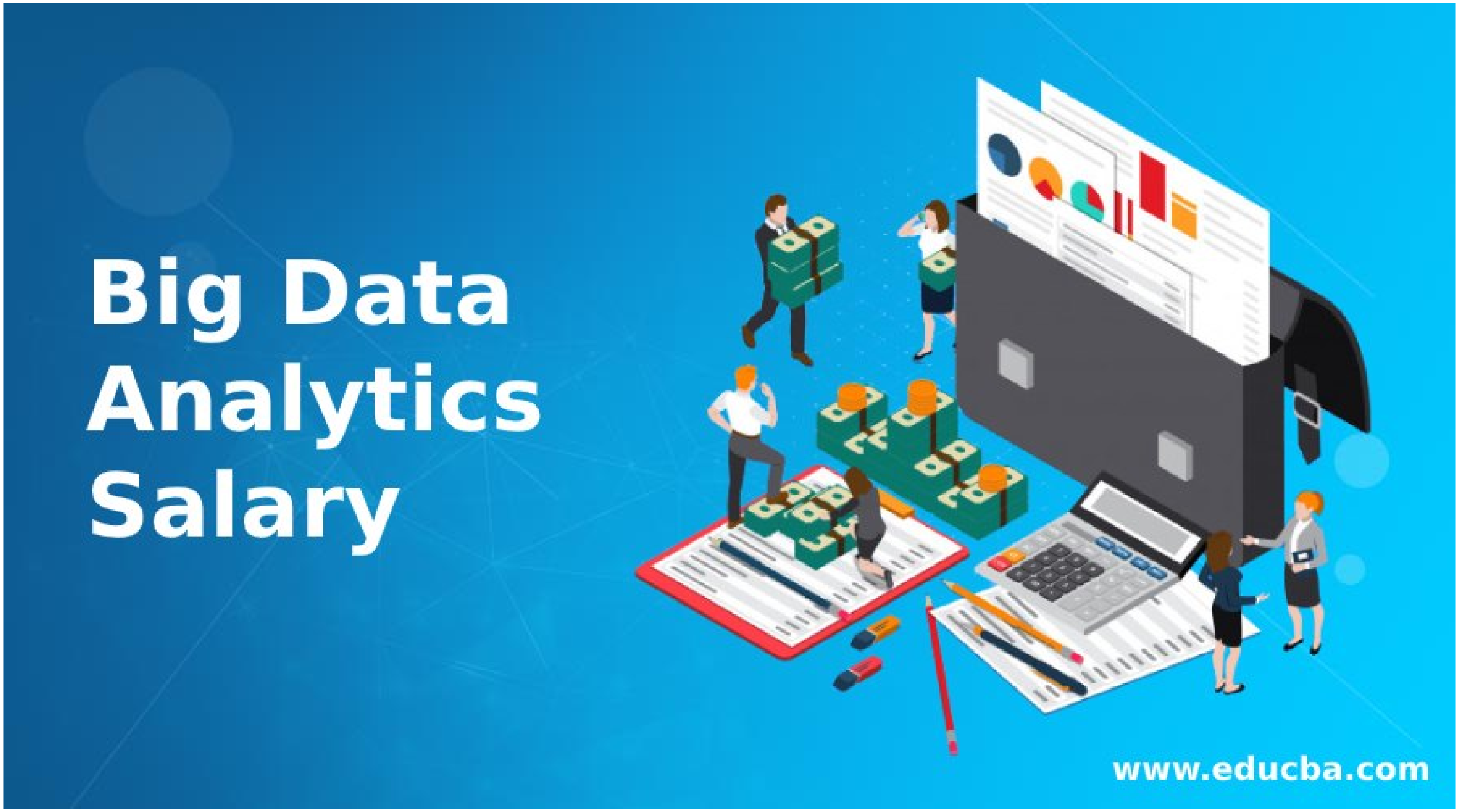
As a Data Scientist, we were task to explore a chosen dataset and we are to select an appropriate dataset, identifying problem statement, and transversing through the phases of the project goals. Being a Data scientist, we decided to explore a dataset focusing on salaries in the fields of Artificial Intelligence (AI), Machine Learning (ML), and Big Data.We are to analyze and give insights into salary dynamics, contribute to industry growth, and advocate for fair compensation practices.The following steps was followed using Python and its packages to analyze the dataset, and we gave a valuable insights and a data-driven recommendation:

Data collection

Data Pre-Processing

Exploratory Data Analysis

Conclusion and Recommendation



Introduction:

This project utilizes a dataset sourced from Kaggle, focusing on salary analysis within the AI/ML and Big Data industry. It contains salary information and various factors influencing work dynamics in the AI/ML and Big Data industry. The aim of this project is to empower stakeholders, including newbies, experienced professionals, hiring managers, recruiters, and startup founders, within the AI/ML and Big Data industry, by providing comprehensive insights into salary dynamics.

Link to the dataset: https://www.kaggle.com/datasets/cedricaubin/ai-ml-salaries/data

Problem Statement:

The primary objective of this project is to analyze the AI/ML Salaries dataset and create visualizations that effectively communicate the insights. Through analyzing the salary data and leveraging various visualizations such as line graphs, histograms, bar charts, scatter plots, box plots e.t.c., the project seeks to identify trends, explore demographic variations, examine the impact of remote work, and identify key drivers of salary fluctuations. Additionally, it aims to provide comparative analysis across different factors, offer career planning insights, support recruitment strategies, facilitate startup decisions, enable informed decision-making, and promote industry transparency. By achieving these objectives and leveraging visualizations effectively, stakeholders will gain valuable insights to navigate salary dynamics, optimize recruitment and retention efforts, foster industry growth, and promote fair compensation practices

Key Objectives

Salary Distribution: Visualize the distribution of salaries across different experience levels, employment types, and company sizes within the AI/ML and Big Data industry. Identify any significant trends or outliers.

Geographic Salary Disparities: Map the average salaries by country to highlight regional disparities in compensation. Explore how salary levels vary based on employee residence and company location.

Remote Work Trends: Analyze the prevalence of remote work in the industry and its impact on salary levels. Visualize the relationship between remote work ratios and salary, considering factors such as experience level and employment type.

Company Size and Salary: Investigate how company size influences salary levels and work dynamics. Create visualizations to compare average salaries and remote work ratios across small, medium, and large companies.

Temporal Analysis: Track salary trends over time (years) to identify any significant fluctuations or patterns. Visualize changes in salary levels, remote work ratios, and other relevant factors over different work years.

Comparative Analysis: Conduct a comparative analysis of salary levels and work dynamics across different job titles. Visualize how salaries vary between entry-level, mid-level, senior-level, and executive-level positions.

About the dataset

The salaries are from ai-jobs. Ai-jobs collects salary information anonymously from professionals all over the world in the AI/ML and Big Data space. The Dataset contains 7366 rows and 11 columns. The dataset contains one table structured and explanation of each column names are as follow:

work_year: The year the salary was paid.

experience_level: The experience level in the job during the year with the following possible values: EN: Entry-level / Junior MI: Mid-level / Intermediate SE: Senior-level / Expert EX: Executive-level / Director employment_type: The type of employement for the role:

PT: Part-time

FT: Full-time

CT: Contract

FL: Freelance

job_title: The role worked in 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 avg. USD rate for the respective year via fxdata.foorilla.com).

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

remote_ratio: The overall amount of work done remotely, possible values are as follows:

0: No remote work (less than 20%)

50: Partially remote

100: Fully remote (more than 80%)

company_location: The country of the employer's main office or contracting branch as an ISO 3166 country code.

company_size: The average number of people that worked for the company during the year:

S: less than 50 employees (small)

M: 50 to 250 employees (medium)

L: more than 250 employees (large)

Link to the project

Kaggle: https://www.kaggle.com/code/opeyemiawe/salary-analysis-ai-ml-big-data-industry

Google Colab: https://colab.research.google.com/drive/1a9fmV-nwTYPedKhnJblogk3cra7OXf1o?usp=sharing

In [1]:

```
!pip install country_converter
```

```
Requirement already satisfied: country_converter in c:\users\hp\anaconda3\lib\site-packages (1.2)
Requirement already satisfied: pandas>=1.0 in c:\users\hp\anaconda3\lib\site-packages (from country_converter) (2.0.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=1.0->cou
ntry_converter) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=1.0->country_conve
rter) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=1.0->country_con
verter) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\hp\anaconda3\lib\site-packages (from pandas>=1.0->country_conv
erter) (1.24.3)
Requirement already satisfied: six>=1.5 in c:\users\hp\anaconda3\lib\site-packages (from python-dateutil>=2.8.2->pandas
>=1.0->country_converter) (1.16.0)
```

Data Collection:

To collect and analyze the data in google colab using Python, The required libraries were imported.

Importing the necessary libraries

In [2]:

```
# for data wrangling and preprocessing
import numpy as np
import pandas as pd

# for visualisations
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.figure_factory as ff
import plotly.graph_objects as go

# for converting country names
import country_converter as coco
```

Reading the dataset

In [3]:

```
# read the data into a pandas dataframe
df= pd.read_csv(r"C:\Users\HP\Desktop\Tech4dev\Monthly_project\Month_3\salaries (1).csv")
# check the first 5 rows of the data
df.head()
```

Out[3]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	compa
0	2023	EN	FT	Data Scientist	100000	USD	100000	US	100	
1	2023	MI	FT	Machine Learning Engineer	45000	EUR	48585	IT	100	
2	2023	MI	FT	Data Analyst	142000	USD	142000	US	0	
3	2023	MI	FT	Data Analyst	128000	USD	128000	US	0	
4	2023	SE	FT	ETL Developer	99500	USD	99500	US	0	

In [4]:

```
#check the last 5 rows
df.tail()
```

Out[4]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	co
7361	2020	SE	FT	Data Scientist	412000	USD	412000	US	100	
7362	2021	MI	FT	Principal Data Scientist	151000	USD	151000	US	100	
7363	2020	EN	FT	Data Scientist	105000	USD	105000	US	100	
7364	2020	EN	CT	Business Data Analyst	100000	USD	100000	US	100	
7365	2021	SE	FT	Data Science Manager	7000000	INR	94665	IN	50	

Data Preprocessing

To clean the dataset, we carefully assess the dataset to identify some issues to handle

Checking the Data

In [5]:

```
# check the size (the number of rows and columns)
df.shape
```

Out[5]:

```
(7366, 11)
```

In [6]:

```
# check the column names, count and datatype
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7366 entries, 0 to 7365
Data columns (total 11 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   work_year           7366 non-null   int64 
 1   experience_level    7366 non-null   object
 2   employment_type     7366 non-null   object
 3   job_title           7366 non-null   object
 4   salary              7366 non-null   int64 
 5   salary_currency     7366 non-null   object
 6   salary_in_usd       7366 non-null   int64 
 7   employee_residence  7366 non-null   object
 8   remote_ratio        7366 non-null   int64 
 9   company_location    7366 non-null   object
 10  company_size        7366 non-null   object
dtypes: int64(4), object(7)
memory usage: 633.1+ KB
```

In [7]:

```
# check for missing values
df.isnull().sum()
```

Out[7]:

```
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
```

Great! Because all the results are zero, it means there is no missing values in any of the columns in our dataset

In [8]:

```
# check for duplicates in the data
df.duplicated().sum()
```

Out[8]:

```
2952
```

Handling duplicates is crucial to ensure the accuracy and reliability of our analysis results, our dataset contains 2,952 duplicate entries.

In [9]:

```
# drop the rows with duplicates
df.drop_duplicates(inplace=True)

# check again for duplicates
df.duplicated().sum()
```

Out[9]:

```
0
```

There are no longer any duplicate values present.

In [10]:

```
#Checking the size again (number of rows and columns)
df.shape
```

Out[10]:

```
(4414, 11)
```

Finally, we got 4414 rows with 11 columns

Checking the uniqueness of each column

In [11]:

```
# View the number of unique values in each column
df.nunique()
```

Out[11]:

```
work_year                4
experience_level         4
employment_type          4
job_title              118
salary                1273
salary_currency         22
salary_in_usd         1540
employee_residence      85
remote_ratio             3
company_location        73
company_size             3
dtype: int64
```

In [12]:

```
#check the uniqueness for remote_ratio
df['remote_ratio'].unique()
```

Out[12]:

```
array([100,   0,  50], dtype=int64)
```

There are three categories representing the extent of remote work based on percentage distribution. We will rename these categories to provide more descriptive names.

In [13]:

```python
# for better presentation, let us convert it to categorical value by replacing
#Replacing "0" with "No Remote", "50" with "Partial/Hybrid", "100" with "Fully Remote"
df['remote_ratio'] = df['remote_ratio'].replace({
    0: "No Remote",
    50: "Partial/ Hybrid",
    100: "Fully Remote"
})
```

In [14]:

```python
# check for the unique categories of the experience level in the data
df['experience_level'].unique()
```

Out[14]:

```
array(['EN', 'MI', 'SE', 'EX'], dtype=object)
```

The categories are currently represented with aliases, and we will assign more descriptive names to each category.

In [15]:

```python
# rename the experience level categories
#replace "EN" with "Entrylevel", "MI" with "Mid-level","SE" with "Senior-level" and "EX" with "Executive-level"
df['experience_level'] = df['experience_level'].replace({
    "EN" : "Entry-level",
    "MI" : "Mid-level",
    "SE" : "Senior-level",
    "EX" : "Executive-level"
})
```

In [16]:

```python
#check for the unique categories of employment type in the data
df['employment_type'].unique()
```

Out[16]:

```
array(['FT', 'CT', 'PT', 'FL'], dtype=object)
```

There are four categories in the type of employment and are currently represented with aliases, we will assign more descriptive names to each category.

In [17]:

```python
# rename the employment type categories
#replace "PT" with "Part-timel", "FT" with "Full-time","CT" with "Contract" and "FL" with "Freelance"
df['employment_type'] = df['employment_type'].replace(
    {
        "PT": "Part-time",
        "FT": "Full-time",
        "CT": "Contract",
        "FL": "Freelance"
    }
)
```

In [18]:

```
# check for the unique values in company size
df['company_size'].unique()
```

Out[18]:

```
array(['M', 'S', 'L'], dtype=object)
```

There are three categories in the type of company size and are currently represented with aliases, we will assign more descriptive names to each category.

In [19]:

```
# rename the company size categories
#replace "M" with "Medium", "S" with "Small", and "L" with "Large"
df["company_size"] = df["company_size"].replace(
    {
        "M": "Medium",
        "S": "Small",
        "L":"Large"
    }
)
```

In [20]:

```
#Checking to confirm the dataset is clean
df
```

Out[20]:

	work_year	experience_level	employment_type	job_title	salary	salary_currency	salary_in_usd	employee_residence	remote_ratio	c
0	2023	Entry-level	Full-time	Data Scientist	100000	USD	100000	US	Fully Remote	
1	2023	Mid-level	Full-time	Machine Learning Engineer	45000	EUR	48585	IT	Fully Remote	
2	2023	Mid-level	Full-time	Data Analyst	142000	USD	142000	US	No Remote	
3	2023	Mid-level	Full-time	Data Analyst	128000	USD	128000	US	No Remote	
4	2023	Senior-level	Full-time	ETL Developer	99500	USD	99500	US	No Remote	
...	...	...	...	...	...	...	...	...	...	
7361	2020	Senior-level	Full-time	Data Scientist	412000	USD	412000	US	Fully Remote	
7362	2021	Mid-level	Full-time	Principal Data Scientist	151000	USD	151000	US	Fully Remote	
7363	2020	Entry-level	Full-time	Data Scientist	105000	USD	105000	US	Fully Remote	
7364	2020	Entry-level	Contract	Business Data Analyst	100000	USD	100000	US	Fully Remote	
7365	2021	Senior-level	Full-time	Data Science Manager	7000000	INR	94665	IN	Partial/ Hybrid	

4414 rows × 11 columns

Exciting! Now we have a cleaned and prepared dataset, we're ready to dive into its insights

Exploratory Data Analysis

Descriptive Statistics

In [21]:

```
#Check the statistics of the columns
#Note, only the columns with numerical datatype are listed
# We will use the describe method to print the summary statistics on the numeric features
df.describe()
```

Out[21]:

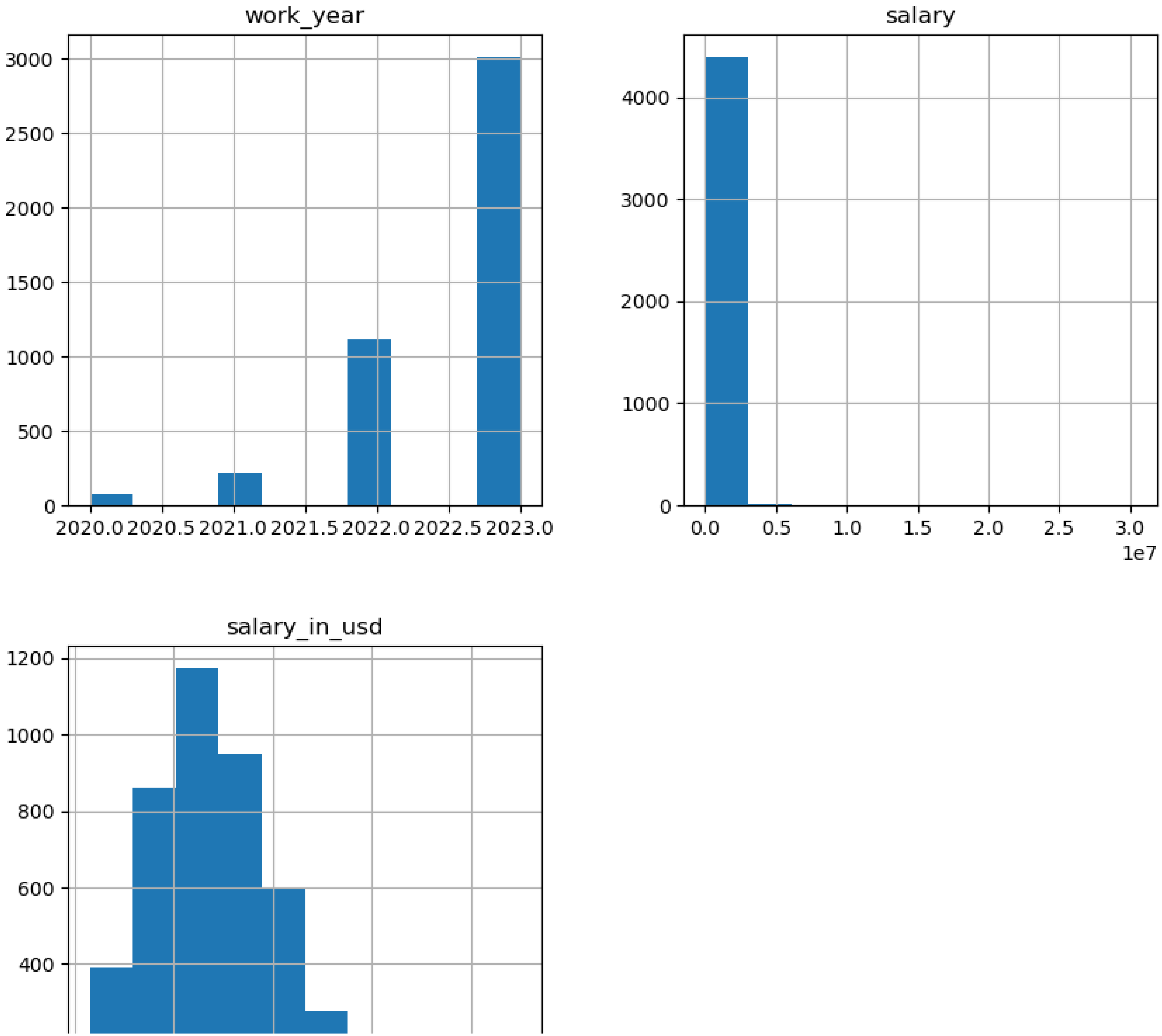
	work_year	salary	salary_in_usd
count	4414.000000	4.414000e+03	4414.000000
mean	2022.599456	1.933878e+05	143743.741278
std	0.663345	6.366969e+05	67131.533475
min	2020.000000	1.400000e+04	15000.000000
25%	2022.000000	9.938250e+04	95000.000000
50%	2023.000000	1.401750e+05	138900.000000
75%	2023.000000	1.896125e+05	185000.000000
max	2023.000000	3.040000e+07	450000.000000

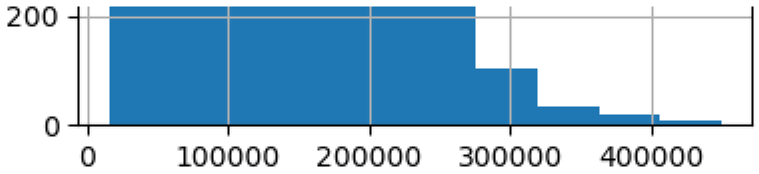
In [22]:

```
# Check the distribution of data in the columns with numerical data type
df.hist(figsize=(10,10))
```

Out[22]:

```
array([[<Axes: title={'center': 'work_year'}>,
        <Axes: title={'center': 'salary'}>],
       [<Axes: title={'center': 'salary_in_usd'}>, <Axes: >]],
      dtype=object)
```


work_year
3000
2500
2000
1500
1000
500
0
2020.0
2020.5
2021.0
2021.5
2022.0
2022.5
2023.0
salary
4000
3000
2000
1000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e7
salary_in_usd
1200
1000
800
600
400



Data Visualization

Univariate analysis

We want to explore each columns in the dataset to see the distributions of features, and to get some useful informations. There are mainly 2 parts in this section: Analysis on Numerical columns and Analysis on categorical columns.

Numerical Columns:

work_year : The year the salary was paid

salary_in_usd : The salary in USD

remote_ratio : The overall amount of work done remotely,

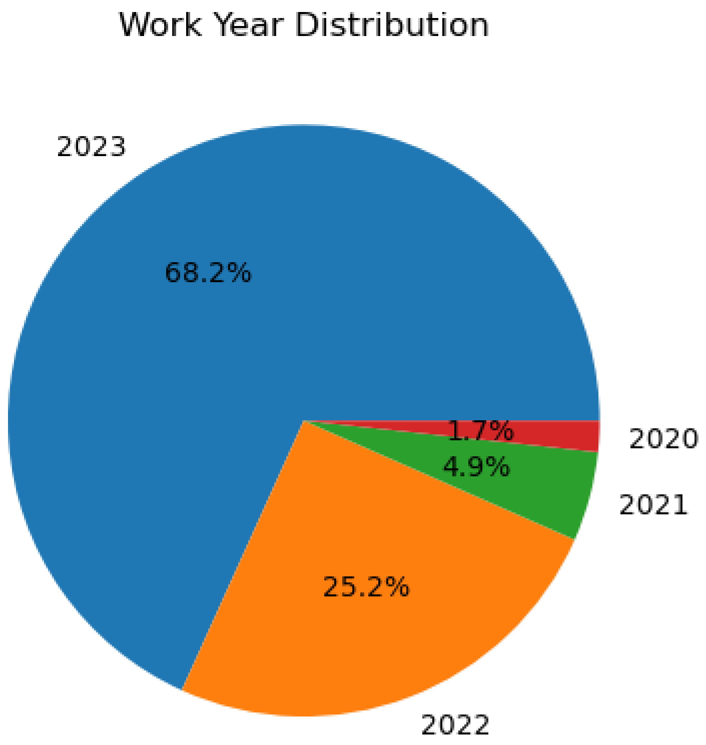
In [23]:
```
df.columns
```

Out[23]:
```
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')
```

Work Year Distribution

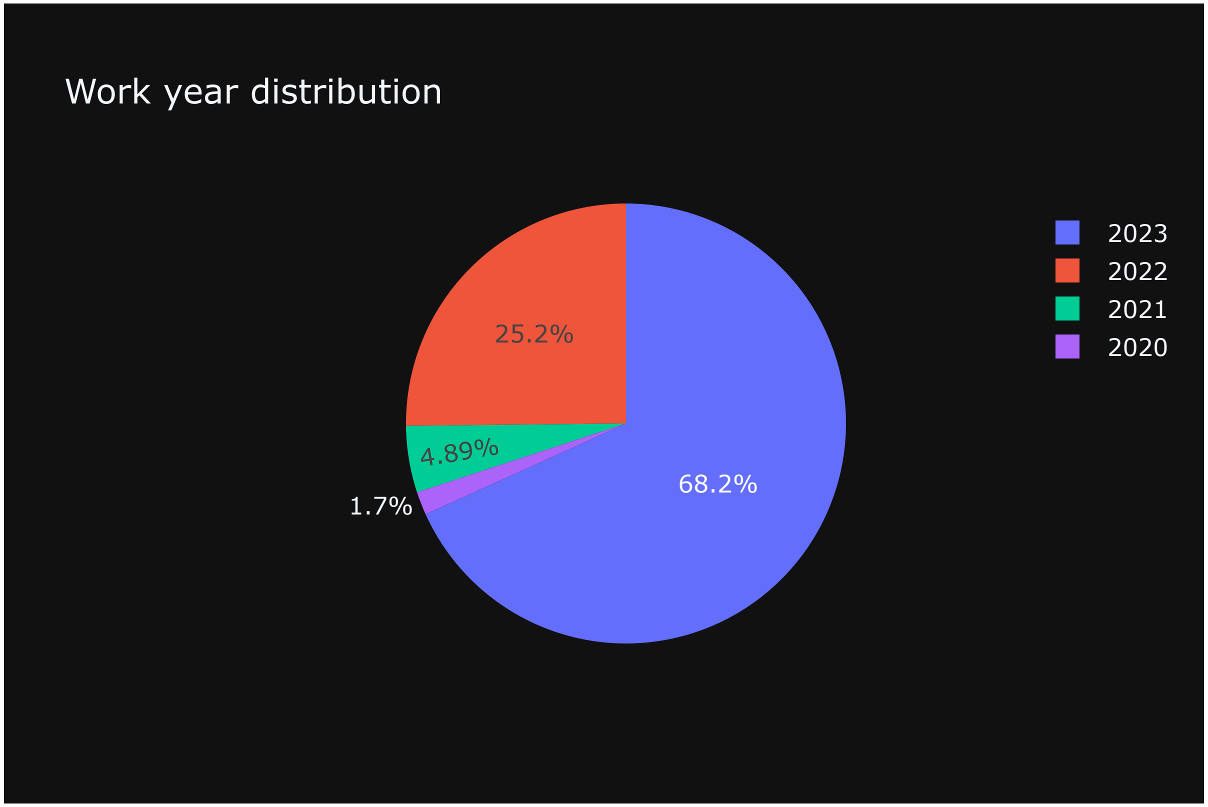
Which year is most represented in the Dataset?

In [24]:
```
#Using Piechart
year_count= df['work_year'].value_counts()
plt.pie(year_count.values, labels=year_count.index, autopct='%1.1f%%')
plt.title("Work Year Distribution");
```



In [25]:

```
#using piechart on plotly
fig = px.pie(names=year_count.index,
             values=year_count.values,
             title='Work year distribution',
             template='plotly_dark',
             width=600,  # Set the width of the plot
             height=400)  # Set the height of the plot
fig.show()
```

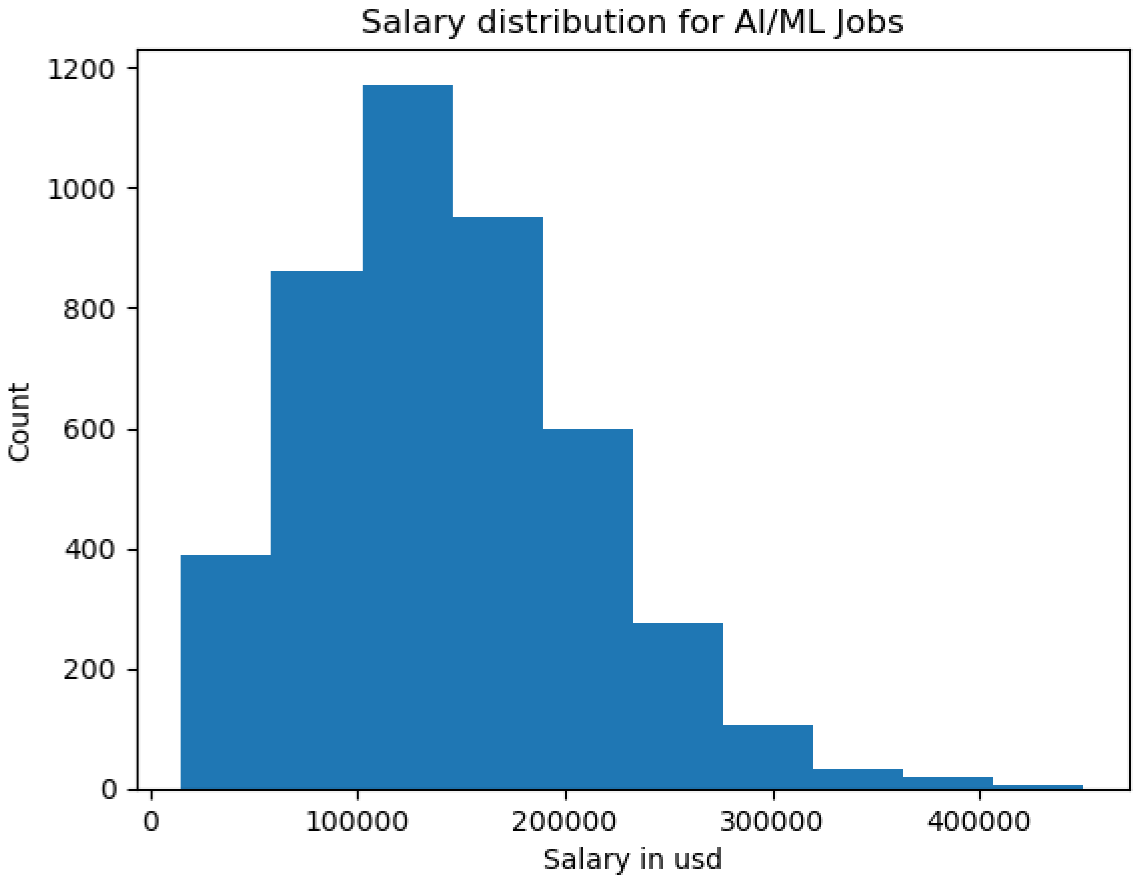


INSIGHT: The dataset demonstrates a notable growth trend from 2020 to 2023, with 2023 accounting for 68.2% of the data and 2022 following at 25.2%. This suggests the continuous evolution of the Artificial Intelligence and Big Data field.

Salary Distribution:

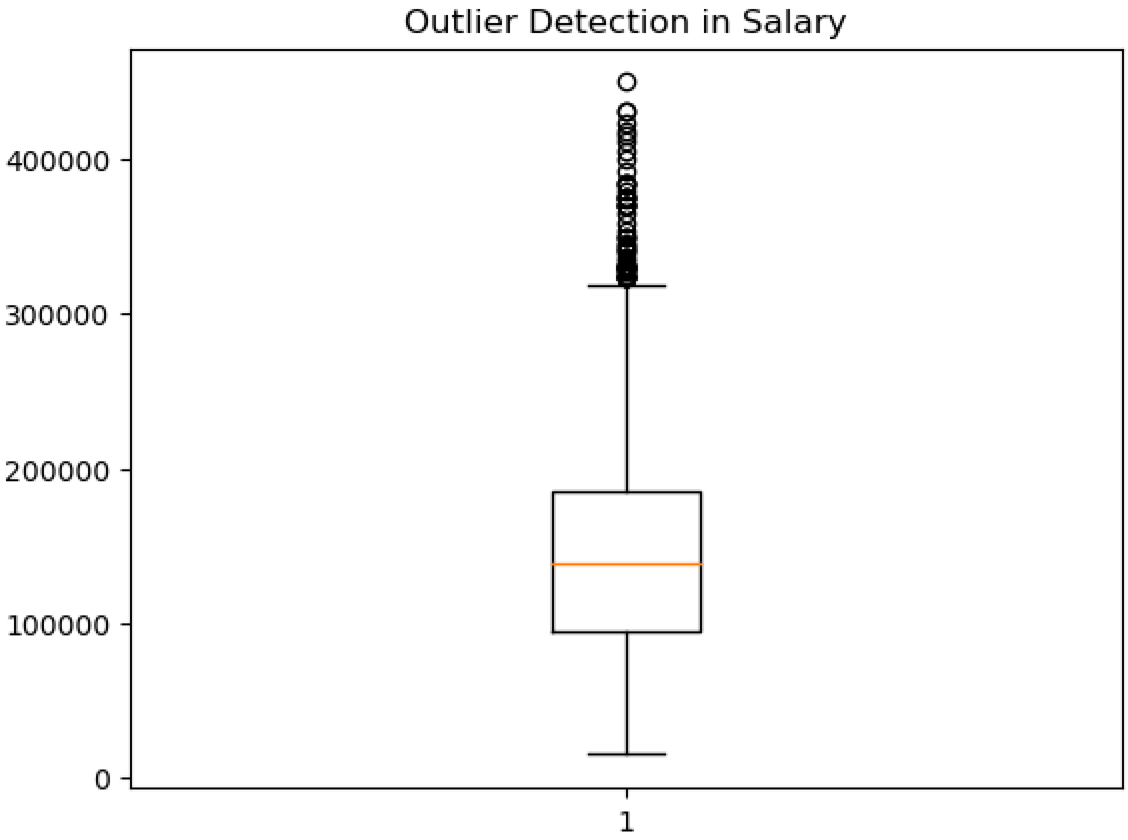
How does the distribution of salaries across the dataset vary, and what insights can be gained regarding central tendency, spread, and the presence of outliers through histograms or box plots?

In [26]:
```
# generate histogram on the salary. note : we will be using salary_in_usd to have it on the same scale
plt.hist(df['salary_in_usd'])
plt.xlabel("Salary in usd")
plt.ylabel("Count")
plt.title("Salary distribution for AI/ML Jobs");
```



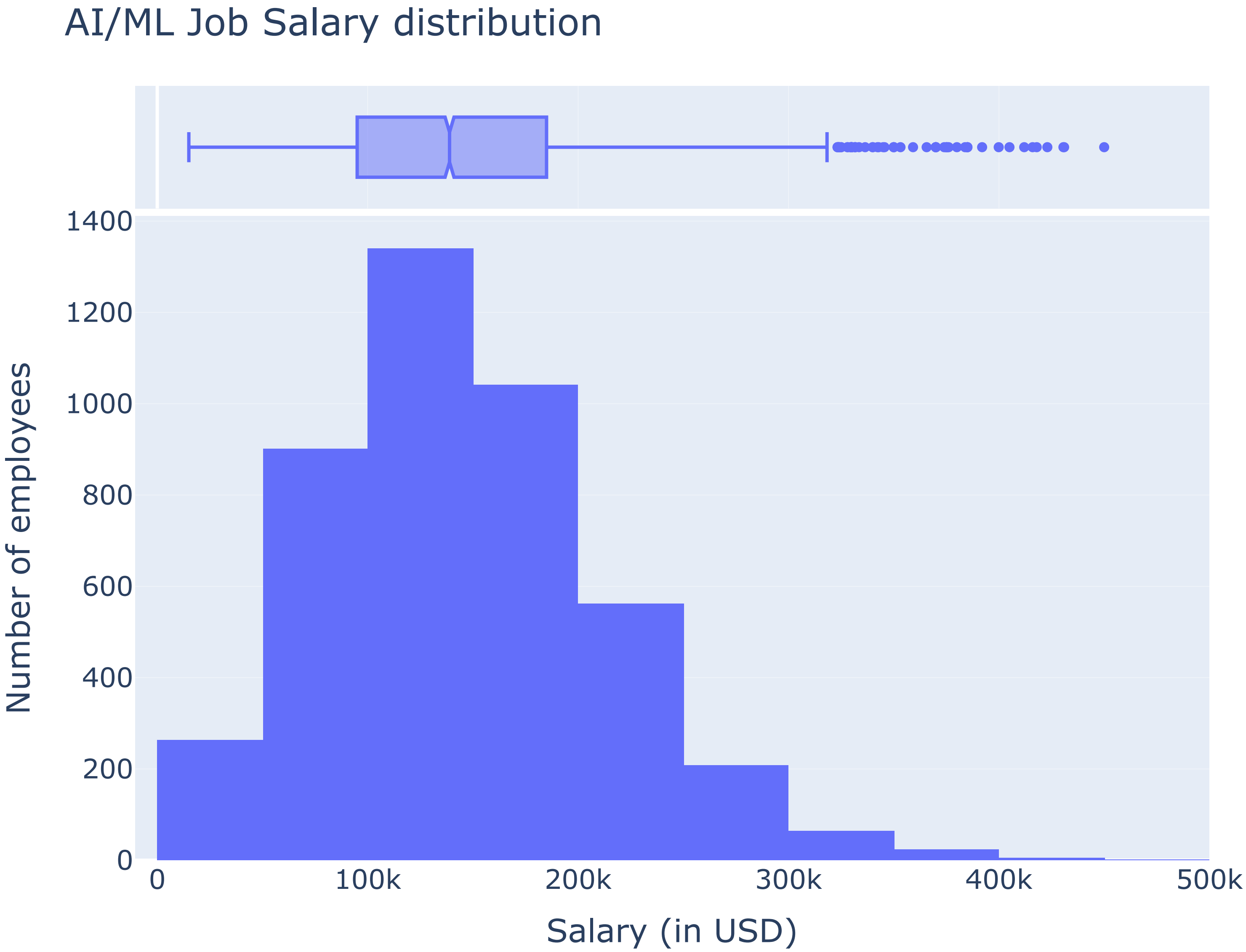
In [27]:

```
#using boxplot
plt.boxplot(df['salary_in_usd'])
plt.title("Outlier Detection in Salary");
```



In [28]:

```
#using histogram and boxplot
fig = px.histogram(data_frame=df, x="salary_in_usd", nbins=10, marginal='box')

fig.update_layout(title='AI/ML Job Salary distribution',
                  yaxis_title='Number of employees',
                  xaxis_title='Salary (in USD)',
                  font_size=16,
                  width=800,  # Set the width of the plot
                  height=600)  # Set the height of the plot

fig.show()
```



INSIGHT: The salary distribution exhibits a slight right skewness attributed to outliers, which is common given the multifaceted nature of salary determinants. Further analysis of our data will help uncover the underlying factors contributing to employee salaries.

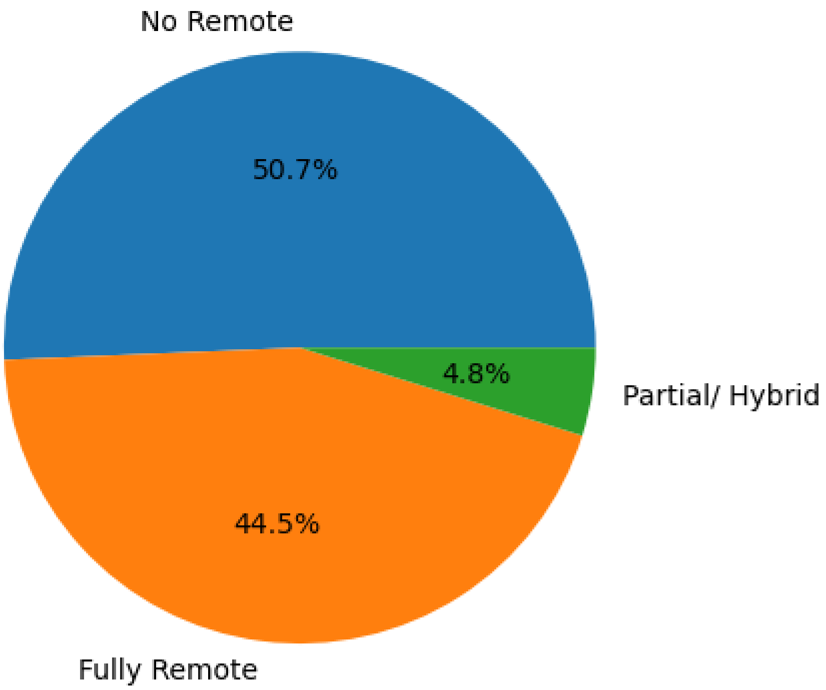
Remote_Ratio Distribution

Remote_Ratio: What is the distribution of remote work ratios among the dataset entries, and how does it reflect the prevalence of remote work arrangements within the industry?

```
In [29]: df['remote_ratio'].unique()
```

```
Out[29]: array(['Fully Remote', 'No Remote', 'Partial/ Hybrid'], dtype=object)
```

```
In [30]: # using matplotlib, plot pie chart to visualise the remote work distribution
         remote_count= df['remote_ratio'].value_counts()
         plt.pie(remote_count.values, labels= remote_count.index,autopct='%1.1f%%');
```



INSIGHT: The distribution of the remote_ratio in the dataset indicates that approximately half of the jobs are onsite, with 44% being fully remote and only around 5% being hybrid.

Categorical Features

experience_level

employment_type

job_title

employee_residence

company_location

company_size

Experience_Level:

What is the distribution of experience levels among the dataset entries, and how does it reflect the composition of the workforce in terms of experience?

In [31]:

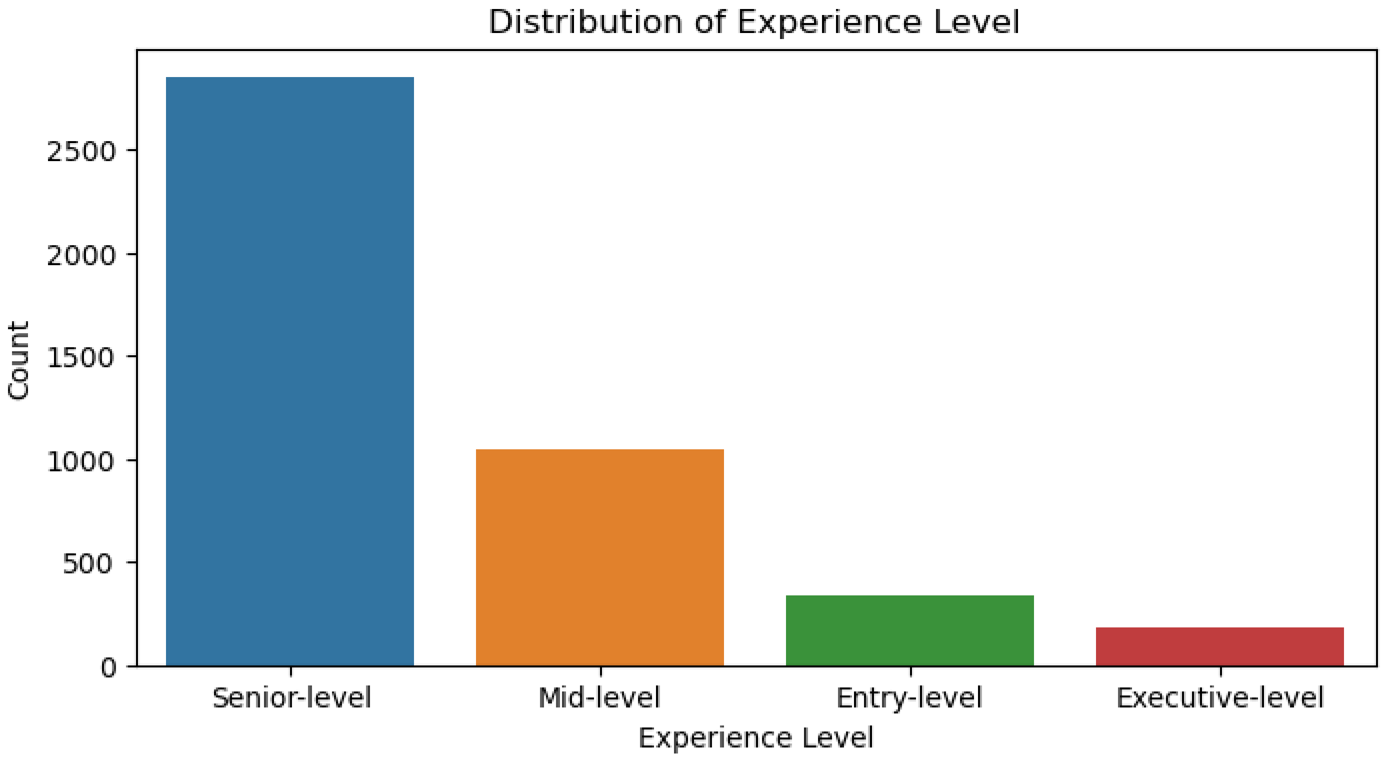
```
#Checking
df["experience_level"].unique()
```

Out[31]:

```
array(['Entry-level', 'Mid-level', 'Senior-level', 'Executive-level'],
      dtype=object)
```

In [32]:

```
# using a countplot to visualise the experience level distribution
plt.figure(figsize=(8, 4))
sns.countplot(df, x='experience_level',  order=df['experience_level'].value_counts().index);
plt.title('Distribution of Experience Level')
plt.xlabel("Experience Level")
plt.ylabel("Count");
```



INSIGHT: The analysis unveiled a predominant presence of employees with senior-level experience , with executive-level employees being the least represented.

Employment type:

What is the distribution of employment types (Part-time, Full-time, Contract, Freelance) within the industry, and how does it reflect the prevailing employment arrangements?

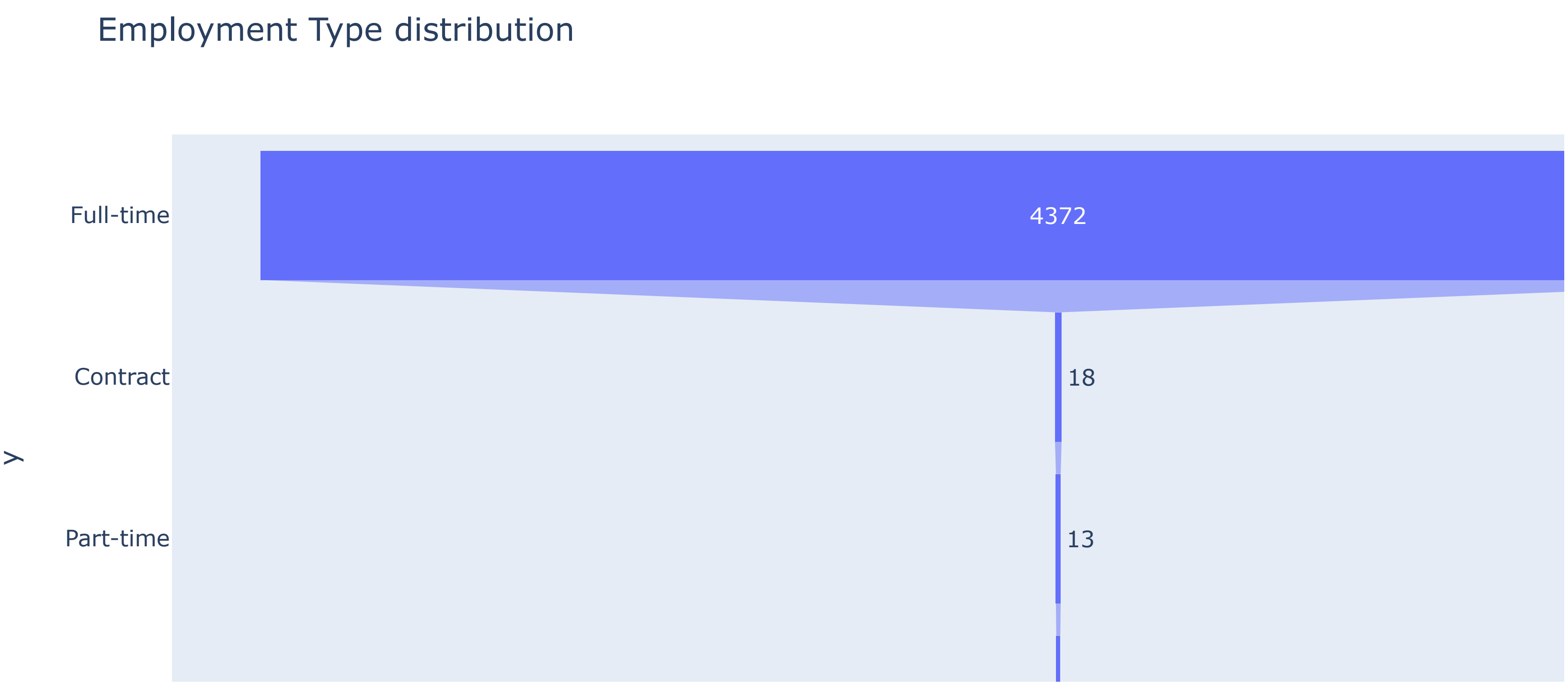
In [33]:
```
#checking
df["employment_type"].unique()
```

Out[33]:
```
array(['Full-time', 'Contract', 'Part-time', 'Freelance'], dtype=object)
```

In [34]:
```
# check counts in the employment type
df['employment_type'].value_counts()
```

Out[34]:
```
employment_type
Full-time    4372
Contract       18
Part-time      13
Freelance      11
Name: count, dtype: int64
```

In [35]:
```
#using funnel
fig = px.funnel(df, x=df['employment_type'].value_counts().values, y=df['employment_type'].value_counts().index, title=
fig.show()
```



INSIGHT: The predominant type of employment observed in the dataset is full-time, suggesting that a majority of employees are engaged on a full-time basis. Conversely, only a small fraction of employees are on contract, part-time, or freelance terms.

Job_title Distribution:

What are the most frequent job titles within the dataset, and how does this reflect the diversity of roles in the AI/ML and Big Data industry?

In [36]:

```
# check for the no of unique job titles represented in the data
df['job_title'].nunique()
```

Out[36]:

```
118
```

There are 118 distinct job titles identified within the AIML/Big data industries in our dataset, showcasing the industry's diverse range of job roles. Despite this diversity, we will focus our visualization on the top 10 job titles.

In [37]:

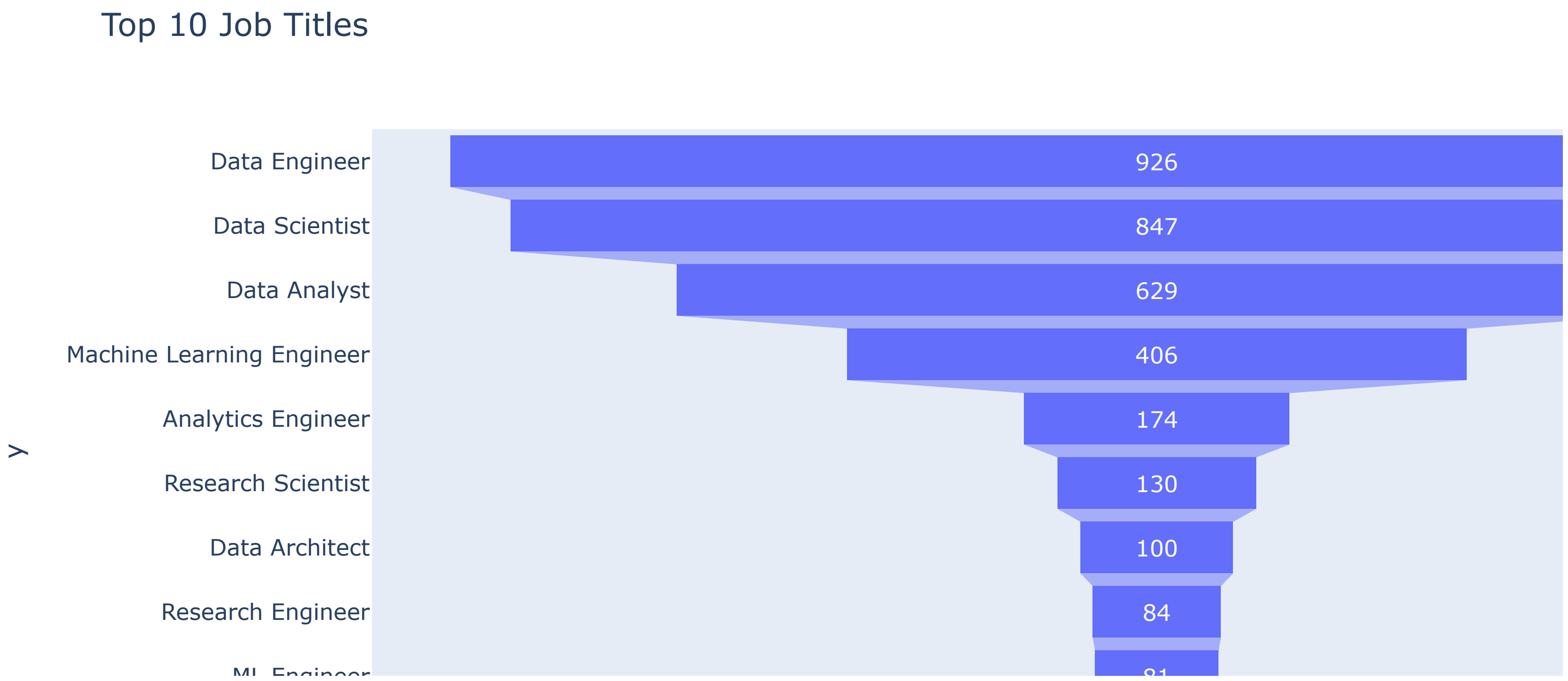
```
# filter the top 10 most frequent jobs in the data
JobTitle_count= df['job_title'].value_counts()[:10]
JobTitle_count
```

Out[37]:

```
job_title
Data Engineer                926
Data Scientist               847
Data Analyst                 629
Machine Learning Engineer    406
Analytics Engineer           174
Research Scientist           130
Data Architect               100
Research Engineer             84
ML Engineer                   81
Applied Scientist             68
Name: count, dtype: int64
```

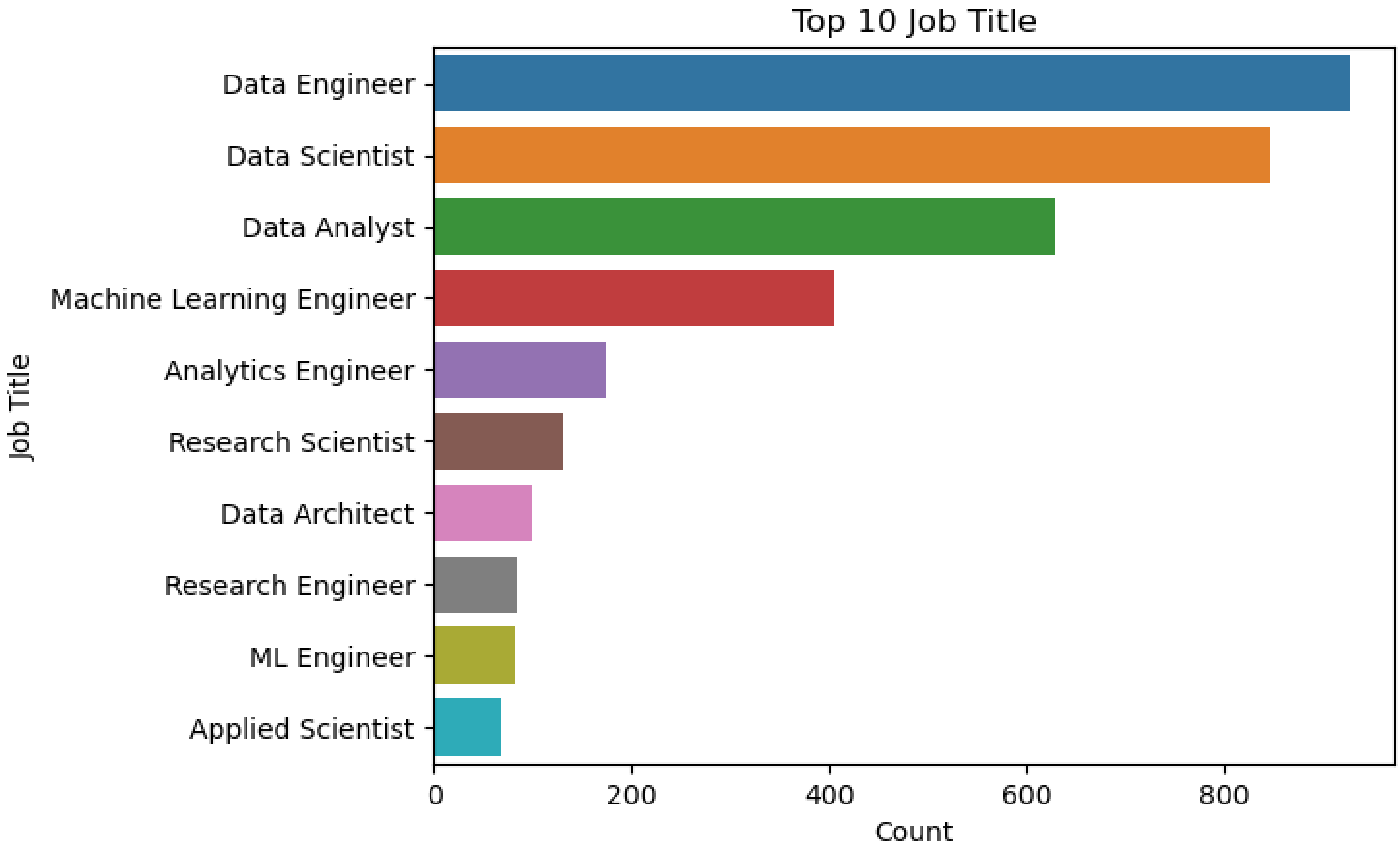
In [38]:

```
# using the funnel plot, et's visualize the top 10 job_titles
fig = px.funnel(df, y=JobTitle_count.index, x=JobTitle_count.values, title='Top 10 Job Titles')
fig.show()
```



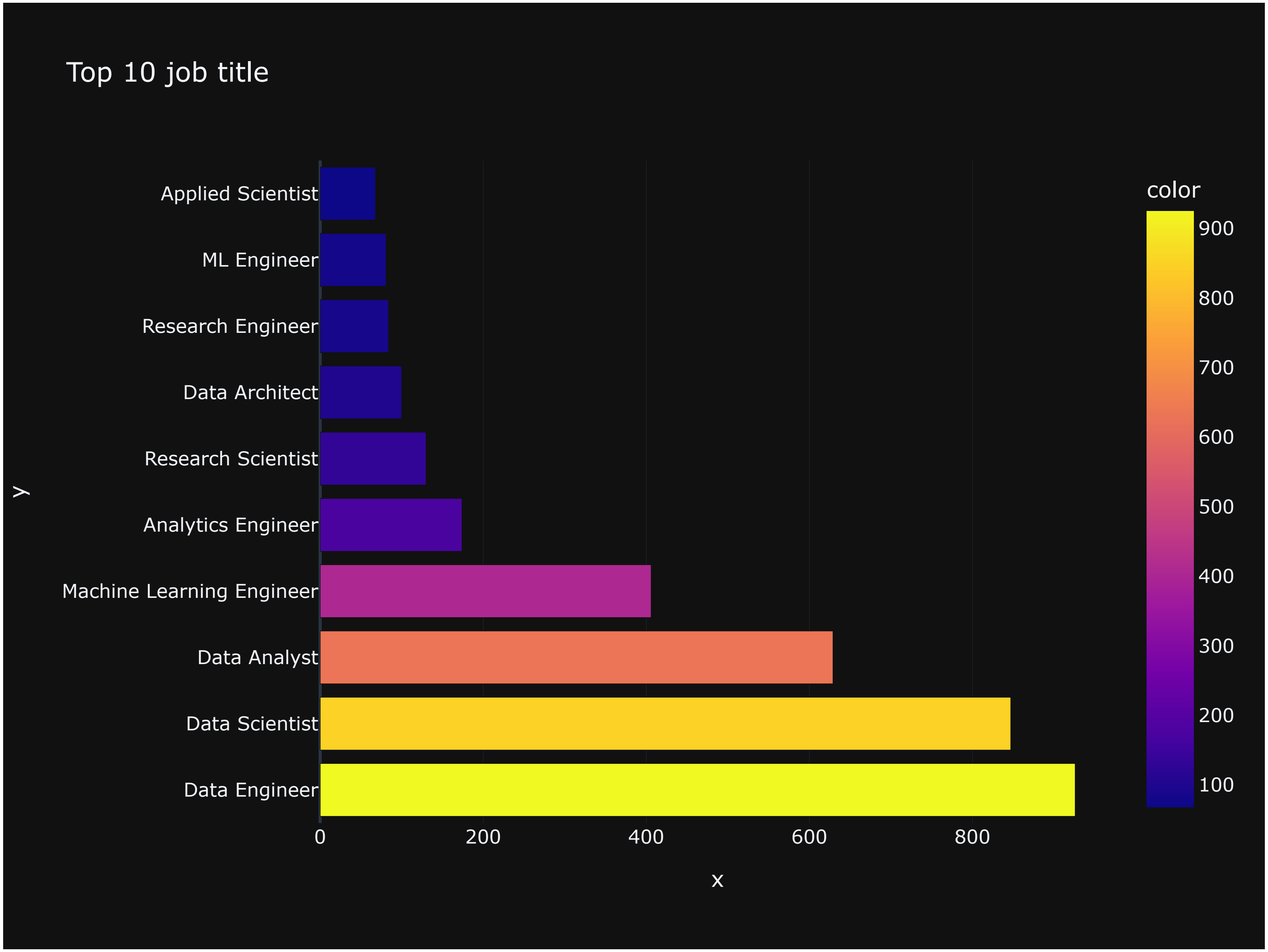
In [39]:

```
# using barplot
sns.barplot(y=JobTitle_count.index, x=JobTitle_count.values)
plt.ylabel("Job Title")
plt.xlabel("Count")
plt.title("Top 10 Job Title");
```



In [40]:

```
#using plotly, lets visualize the top 10 job_titles
fig = px.bar(
    x=JobTitle_count,
    y=JobTitle_count.index,
    color=JobTitle_count,
    template='plotly_dark',
    title='Top 10 job title',
    width=800,  # Set the width of the plot
    height=600  # Set the height of the plot
)

fig.show()
```



INSIGHT: Upon filtering the data for the top 10 job roles, it was found that Data Engineer, Data Scientist and Data Analyst ranked the top 3 frequent job titles.

Employee_residence:

What is the geographical distribution of employee residences and company locations across different countries, and how does it reflect the global footprint of the AI/ML and Big Data industry?

In [41]:

```
#converts country names in a DataFrame column to their ISO3 country codes using the country_converter
countryNames = coco.convert(names=df['employee_residence'], to="ISO3")
df['employee_residence'] = countryNames
```

In [42]:

```
#checking the uniqueness
df['employee_residence'].nunique()
```

Out[42]:

```
85
```

The data includes employees from 85 countries. To streamline our analysis and visualizations, we will focus on the top 10 countries where the employees reside.

In [43]:

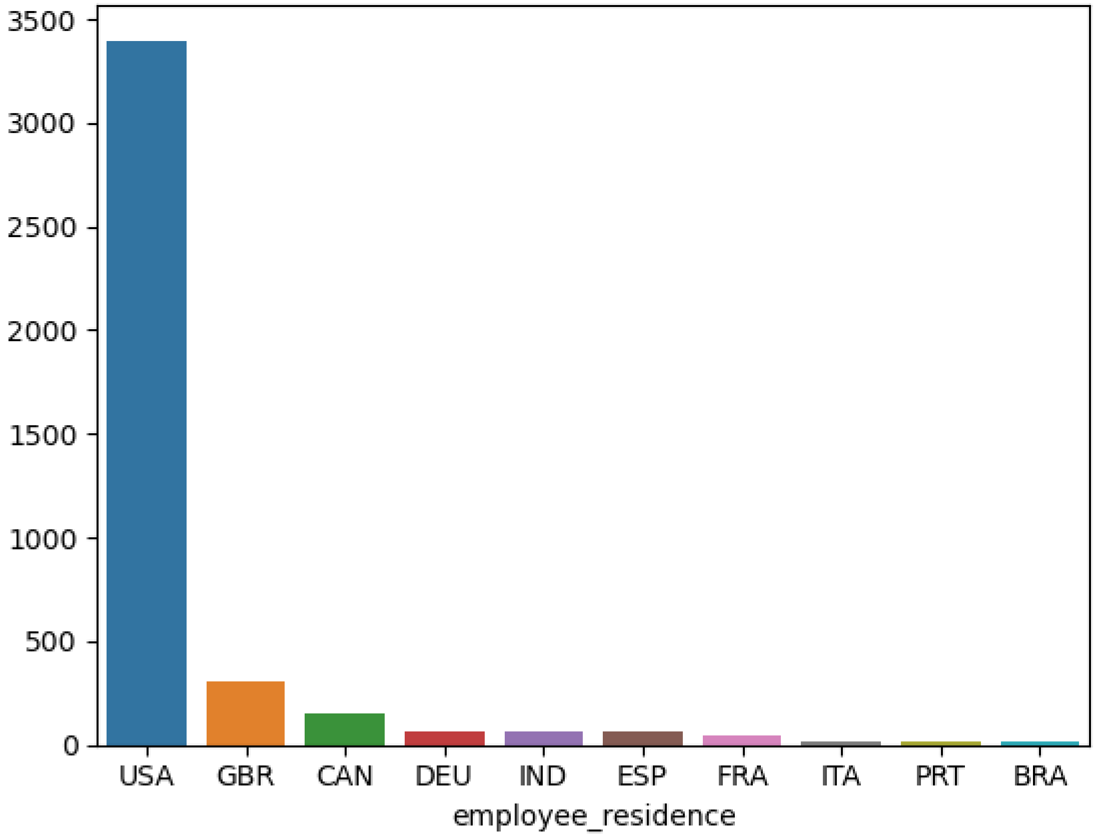
```
# filter top 10 employees residence
residence_count= df['employee_residence'].value_counts()[:10]
residence_count
```

Out[43]:

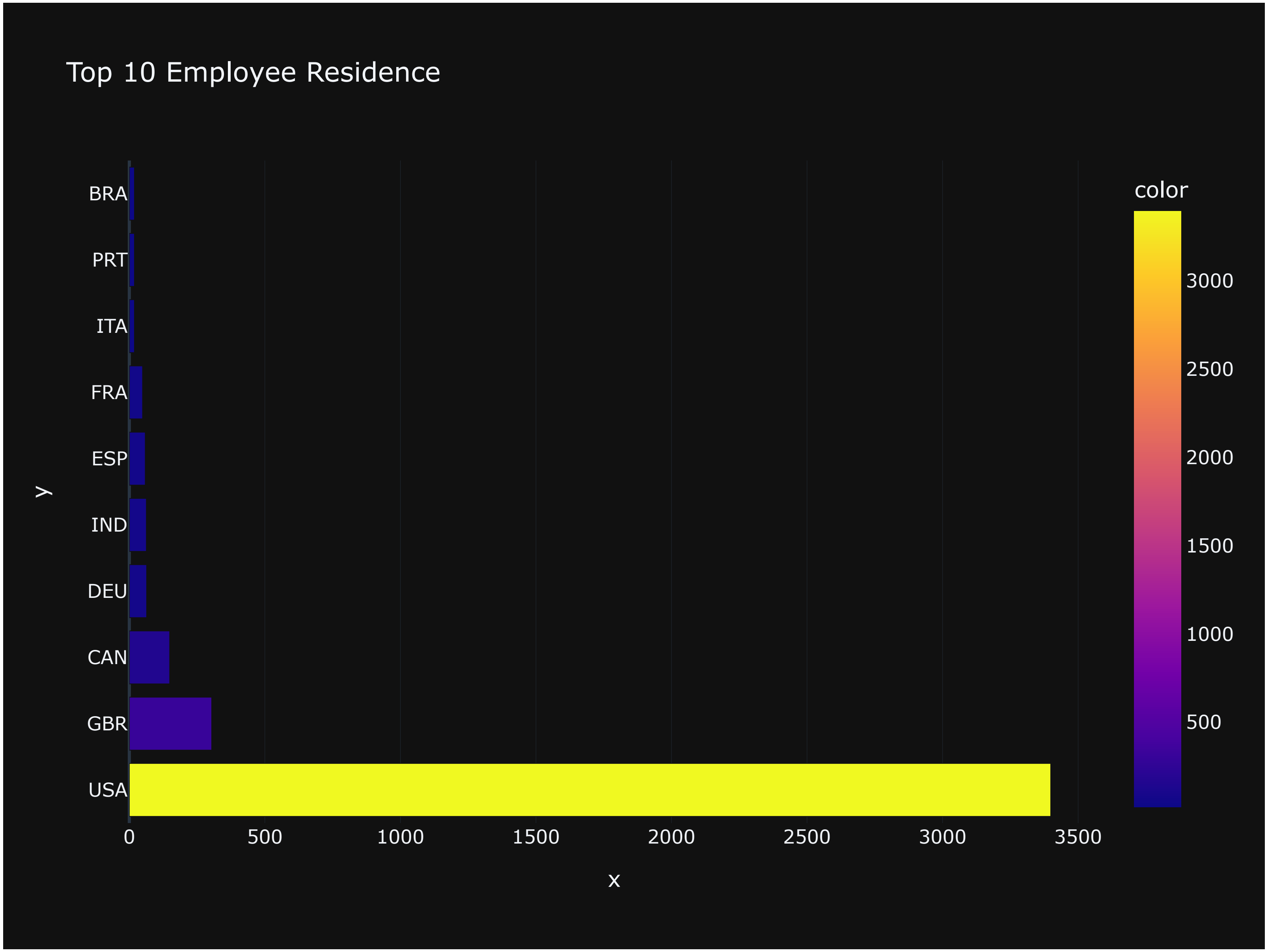
```
employee_residence
USA    3399
GBR     304
CAN     149
DEU      64
IND      63
ESP      59
FRA      49
ITA      19
PRT      19
BRA      19
Name: count, dtype: int64
```

In [44]:

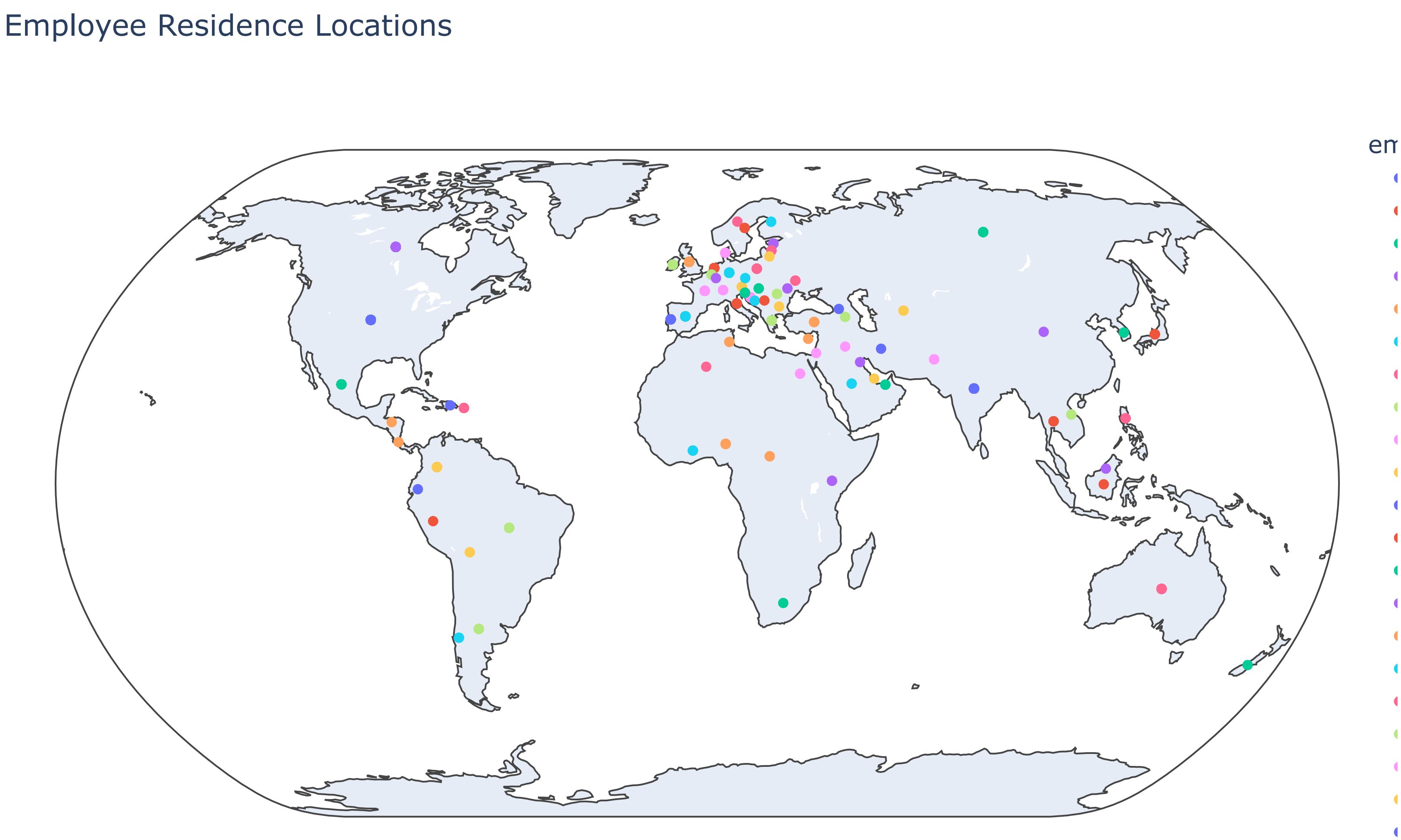
```
#Using seaborn, lets visualize by creating a barplot
sns.barplot(x=residence_count.index, y=residence_count.values);
```



```
In [45]: #using plotly
         px.bar(
             x=residence_count,
             y=residence_count.index,
             color=residence_count,
             template='plotly_dark',
             title='Top 10 Employee Residence',
             width=800,  # Set the width of the plot
             height=600  # Set the height of the plot

         )
```

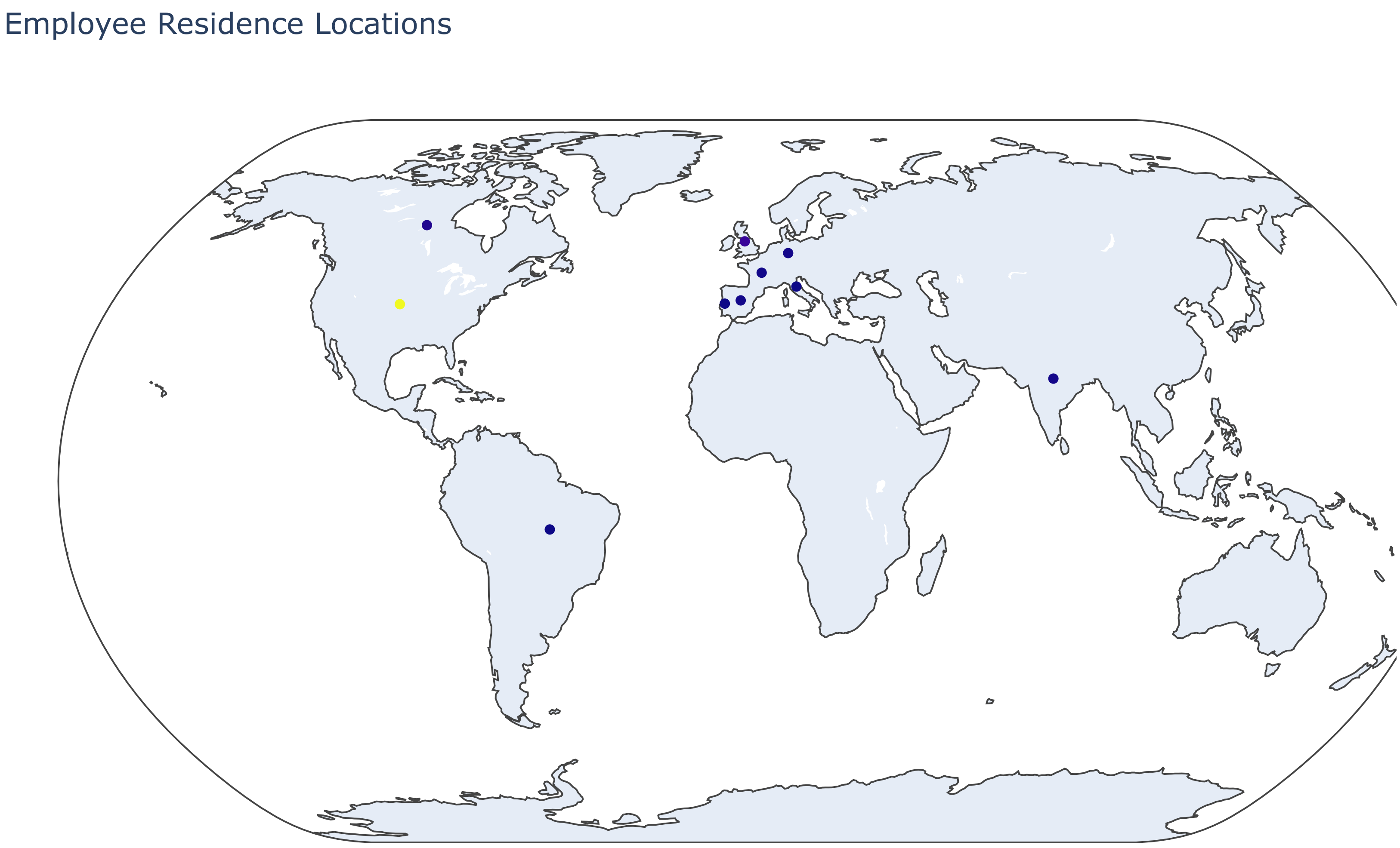


In [46]:
```
# Map visualization for all employee's residence
fig_residence = px.scatter_geo(df, locations='employee_residence', locationmode='ISO-3', color='employee_residence',
                               projection='natural earth', title='Employee Residence Locations',
                               width=1000,  # Set the width of the plot
    height=600  # Set the height of the plot
)
fig_residence.show()
```



```
In [47]: # Map visualization for all employee's residence
fig_residence = px.scatter_geo(df, locations=residence_count.index, locationmode='ISO-3', color=residence_count.values,
                               projection='natural earth', title='Employee Residence Locations',
                               width=1000,  # Set the width of the plot
    height=600  # Set the height of the plot
```

```
                                )
fig_residence.show()
```



INSIGHT: The analysis indicates that the majority of employees are residing in the USA.

Company Location:

In [48]:

```
# convert the company location in to country names based on ISO3 standard
companyLocation = coco.convert(names=df['company_location'], to="ISO3")
df['company_location'] = companyLocation
```

In [49]:

```
# check for the number of countries where the companies captured in the dataset are located
df['company_location'].nunique()
```

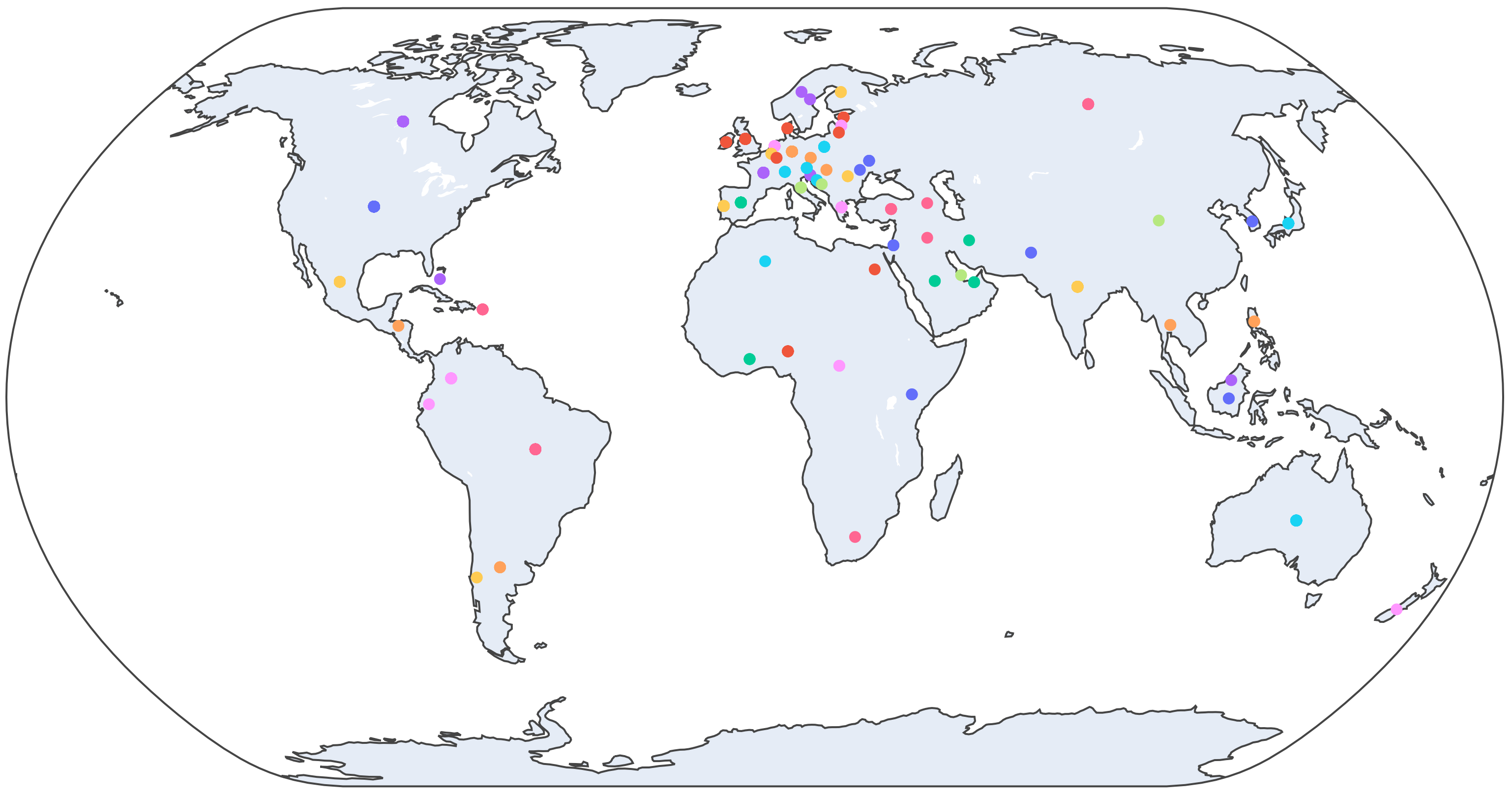
Out[49]:

```
73
```

In [50]:

```
# Map visualization for company location
fig_residence = px.scatter_geo(df, locations='company_location', locationmode='ISO-3', color='company_location',
                               projection='natural earth', title='Company Locations',
                               width=1000,  # Set the width of the plot
    height=600  # Set the height of the plot
                               )
fig_residence.show()
```

Company Locations



The data includes companies located in 73 countries. To facilitate our analysis and visualizations, we will focus on the top 10 countries where these companies are based.

In [51]:

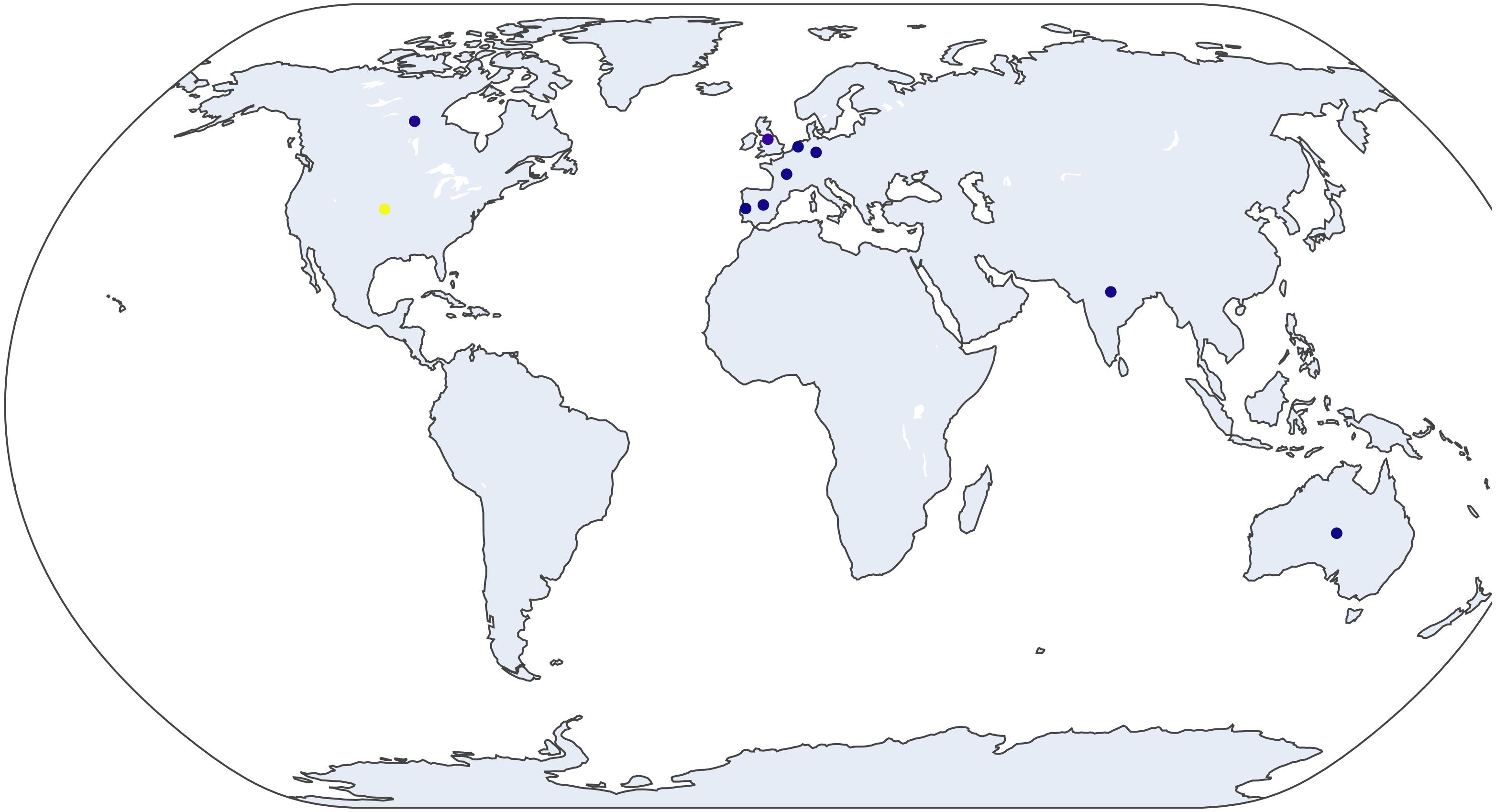
```
# filter top 10 company location
Top10_companyLocation= df['company_location'].value_counts()[:10]
Top10_companyLocation
```

Out[51]:

```
company_location
USA    3446
GBR     311
CAN     150
DEU      71
ESP      56
IND      49
FRA      45
AUS      21
PRT      17
NLD      17
Name: count, dtype: int64
```

In [52]:

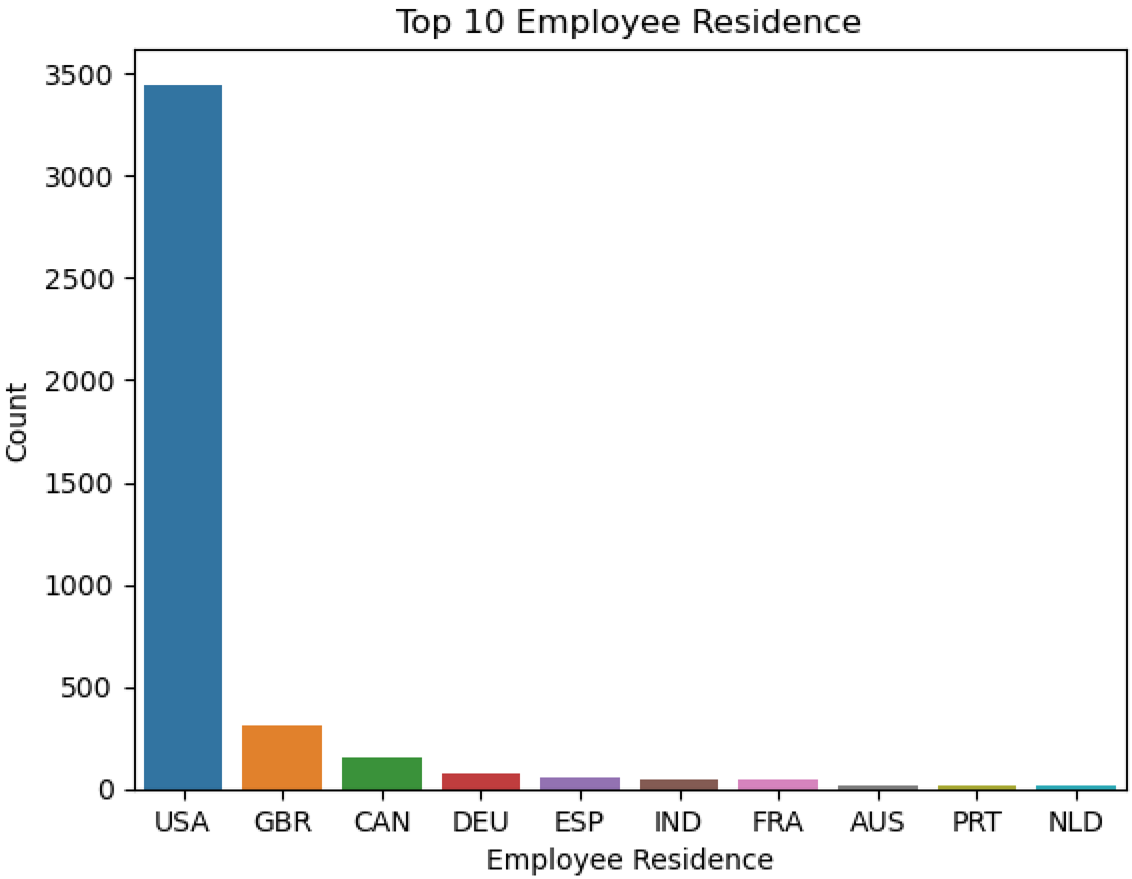
```
# Map visualization for top 10 company location
fig_residence = px.scatter_geo(df, locations=Top10_companyLocation.index, locationmode='ISO-3', color=Top10_companyLoca
                               projection='natural earth', title='Top 10 Company Locations',
                               width=1000,  # Set the width of the plot
    height=600  # Set the height of the plot
                               )
fig_residence.show()
```



Top 10 Company Locations

In [53]:

```
sns.barplot(x=Top10_companyLocation.index, y=Top10_companyLocation.values)
plt.title("Top 10 Employee Residence")
plt.xlabel("Employee Residence")
plt.ylabel("Count");
```



INSIGHT: A significant proportion of the companies in the dataset are located in the USA, suggesting that the USA serves as a prominent hub for AI and Big Data industries.

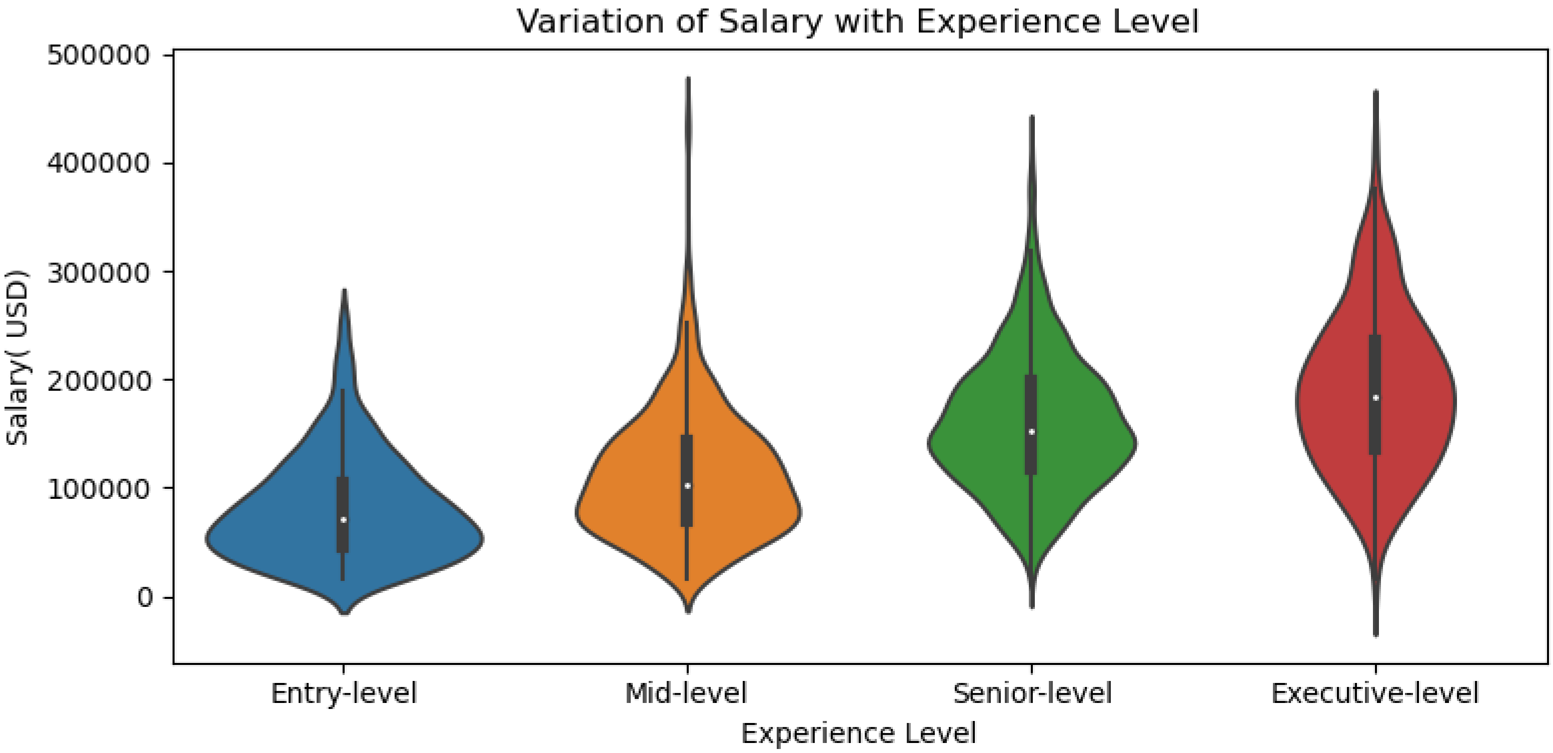
Bivariate Analysis

The Bivariate analyses will help us explore the relationships and interactions between different variables in the dataset, providing deeper insights into factors influencing salaries, remote work preferences, and other aspects of employment in the AI/ML and Big Data space.

Finding: How does salary vary between different experience levels within the AI/ML and Big Data space?

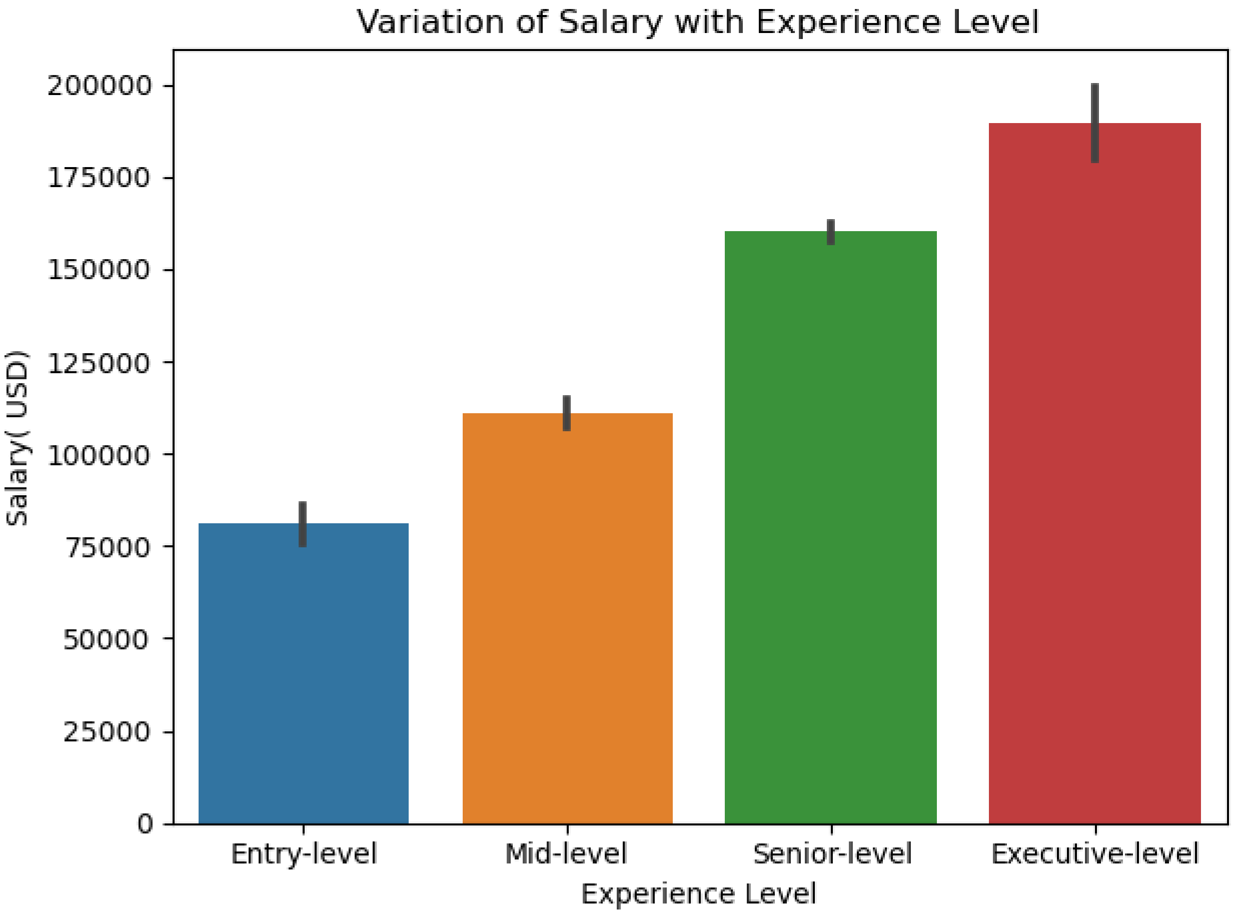
In [54]:

```
#using violinplot
plt.figure(figsize=(8, 4))
sns.violinplot(df, x= 'experience_level', y='salary_in_usd')
plt.title("Variation of Salary with Experience Level")
plt.ylabel("Salary( USD)")
plt.xlabel("Experience Level")
plt.tight_layout()
```



In [55]:

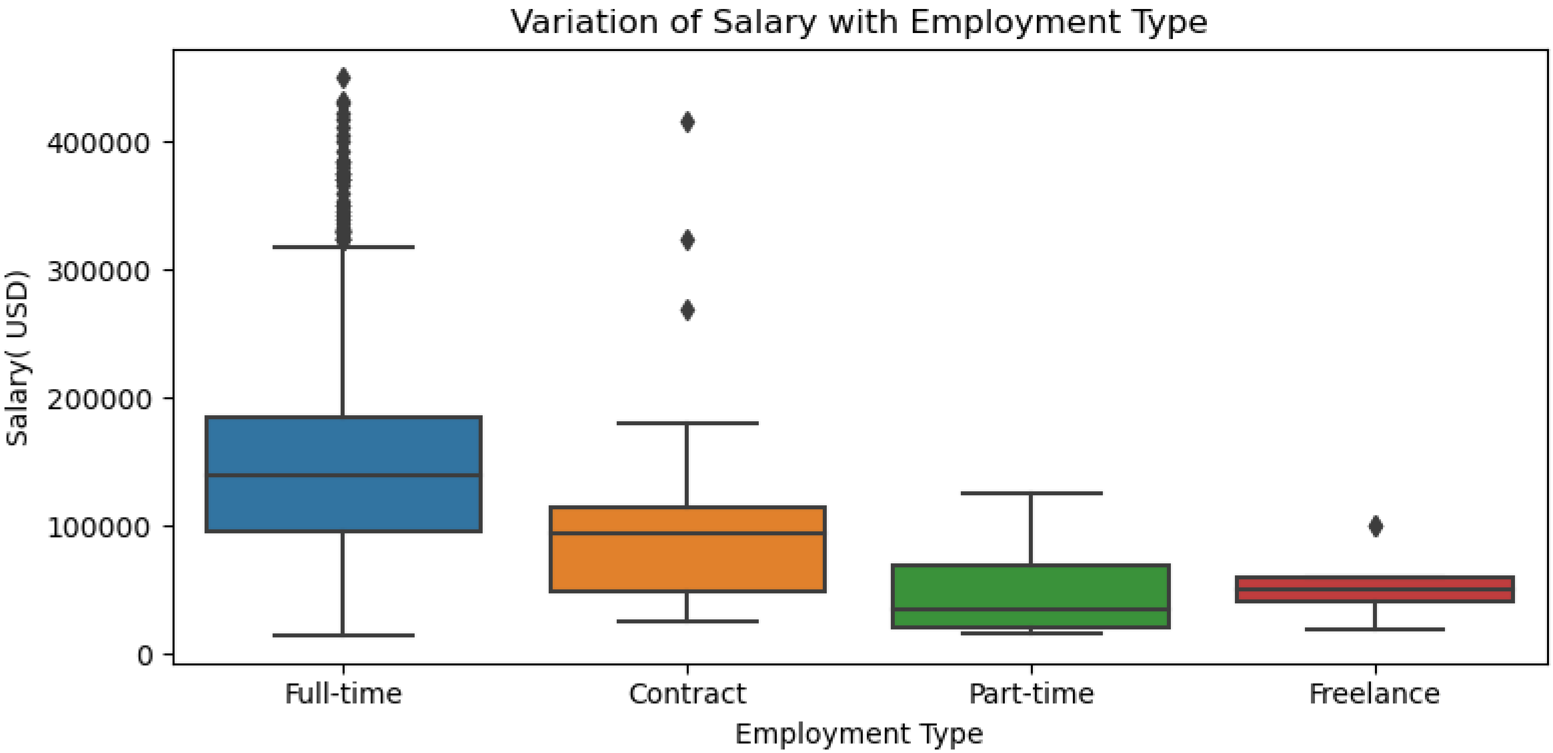
```
#using barplot
sns.barplot(df, x="experience_level", y="salary_in_usd")
plt.title("Variation of Salary with Experience Level")
plt.ylabel("Salary( USD)")
plt.xlabel("Experience Level")
plt.tight_layout()
```



INSIGHT: Variations in employee salaries are evident based on the experience level, with executive-level employees earning the most and entry-level employees earning the least. This trend aligns with expectations, as experience level is a significant determinant of salaries.

Finding: We want to find out how salary vary between different employment type within the AI/ML and Big Data space?

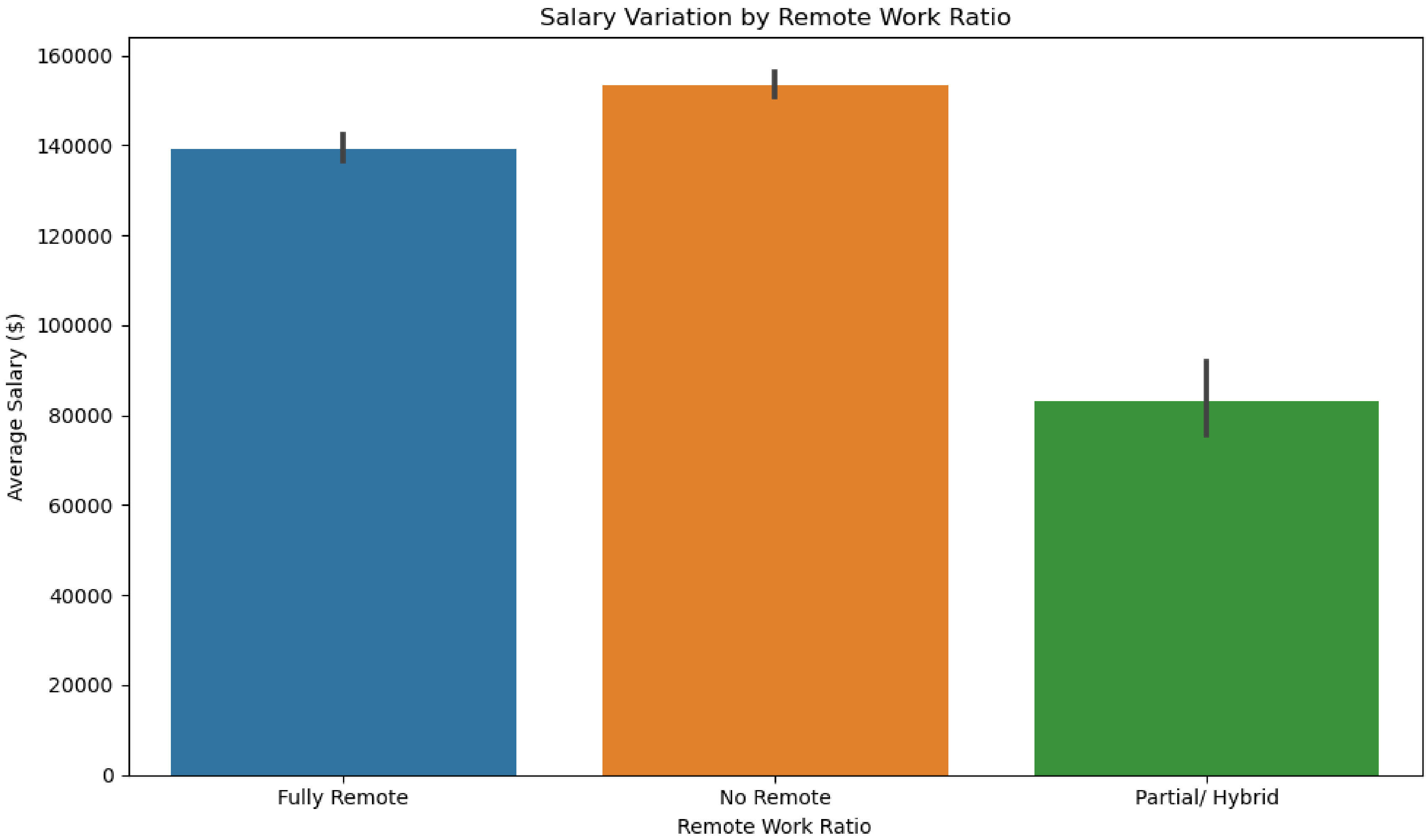
In [56]:
```
#using boxplot
plt.figure(figsize=(8, 4))
sns.boxplot(df, x="employment_type" , y="salary_in_usd")
plt.title("Variation of Salary with Employment Type")
plt.ylabel("Salary( USD)")
plt.xlabel("Employment Type")
plt.tight_layout()
```



INSIGHT: Employees on full-time jobs earn the highest salaries, whereas freelancers earn the least. This observation underscores the significant impact of employment type on employee salaries.

Finding: We want to know how the salary vary with remote option

In [57]:
```
#Using matplotlib, create a barplot
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='remote_ratio', y='salary_in_usd')
plt.title('Salary Variation by Remote Work Ratio')
plt.xlabel('Remote Work Ratio')
plt.ylabel('Average Salary ($)')
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```



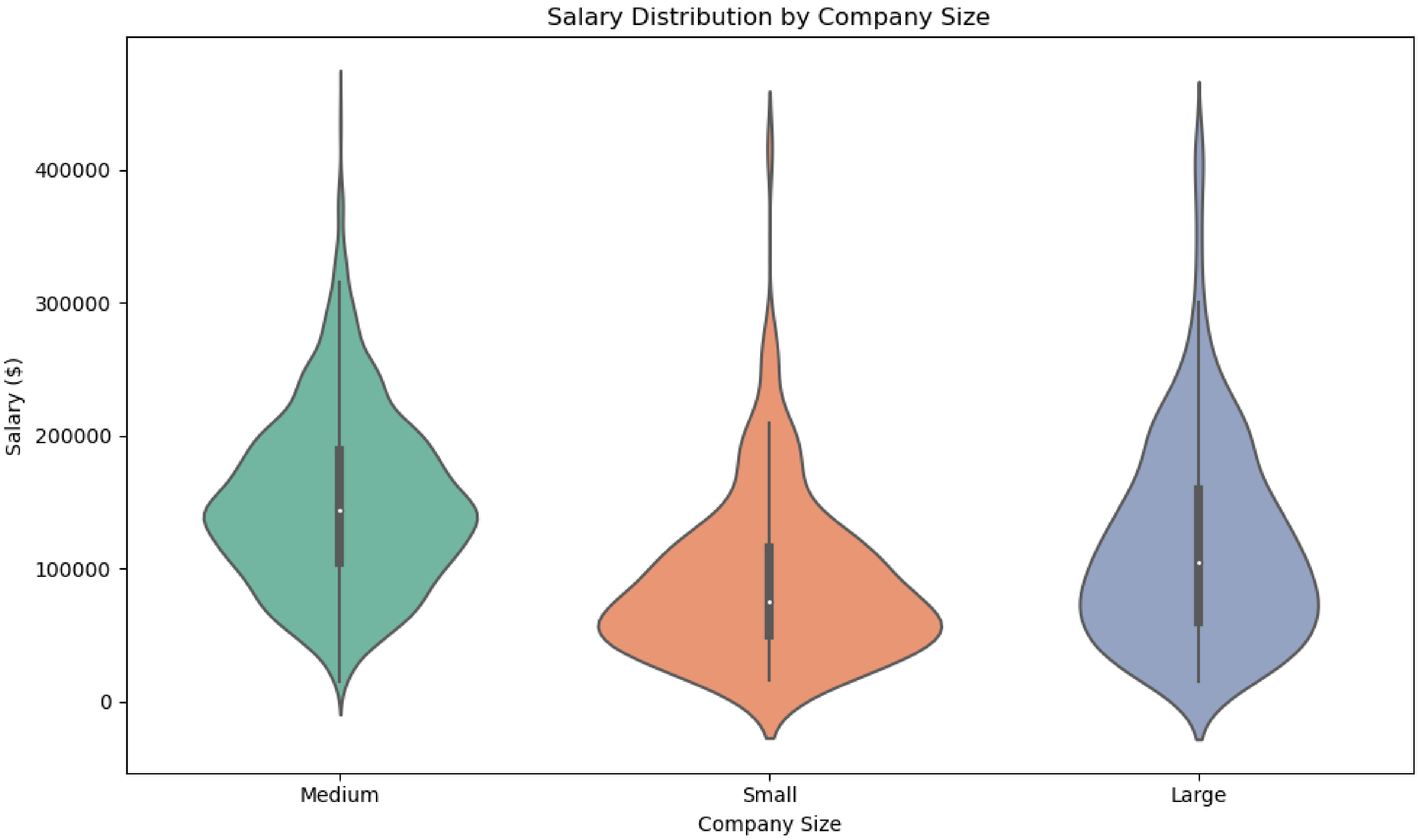
INSIGHT: Employees working onsite typically earn more than those on hybrid arrangements, highlighting the influence of remote options on salaries.

Finding: What is the correlation between salary and the company_size?

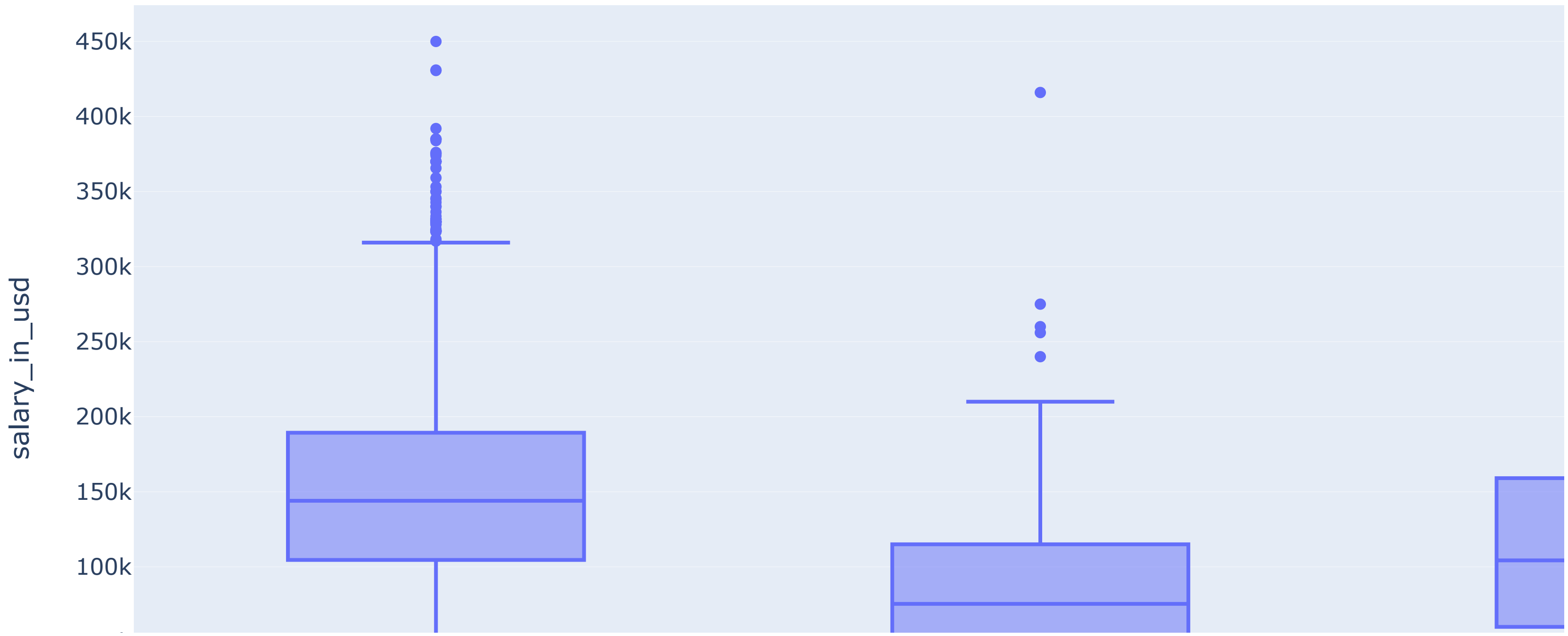
In [58]:
```
# check for the unique values
df['company_size'].unique()
```

Out[58]:
```
array(['Medium', 'Small', 'Large'], dtype=object)
```

In [59]:
```
# using seaborn, create a box plot
plt.figure(figsize=(10, 6))
sns.violinplot(data=df, x='company_size', y='salary_in_usd', palette='Set2')
plt.title('Salary Distribution by Company Size')
plt.xlabel('Company Size')
plt.ylabel('Salary ($)')
plt.tight_layout()
plt.show()
```



In [60]:
```
#Using plotly
fig = px.box(df, x='company_size', y='salary_in_usd')
fig.show()
```



Insight: Companies with medium sizes have a higher salary structure.

Multivariate Analysis:

This will help us to understand the relationships and interactions between multiple variables simultaneously in our dataset.This approach will enables us to gain comprehensive insights into the underlying relationships among multiple variables, leading to more informed decision-making and a better understanding of the factors driving outcomes of interest.

In [61]:
```
#checking the columns
df.columns
```

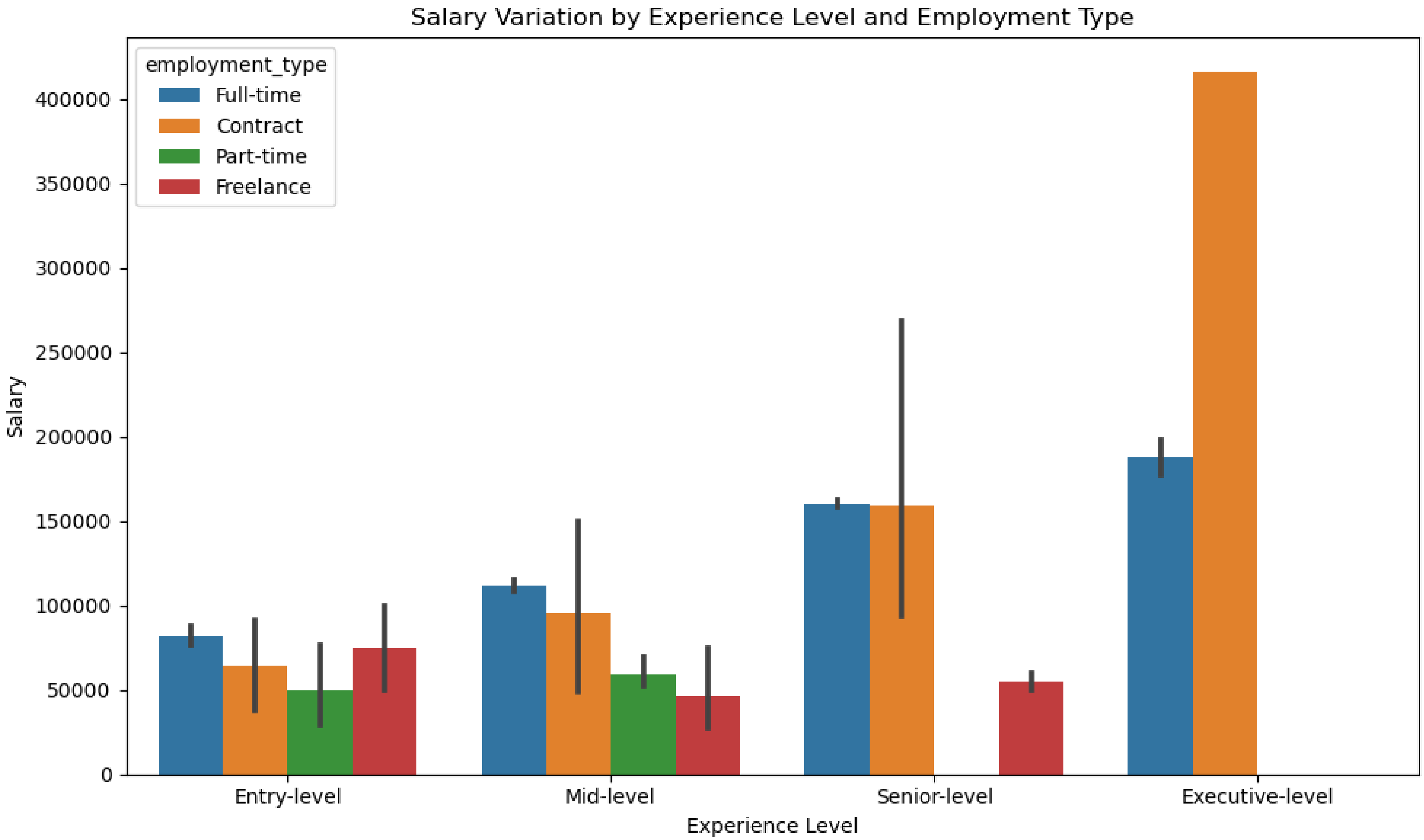
Out[61]:
```
Index(['work_year', 'experience_level', 'employment_type', 'job_title',
       'salary', 'salary_currency', 'salary_in_usd', 'employee_residence',
       'remote_ratio', 'company_location', 'company_size'],
      dtype='object')
```

In [62]:
```
#print columns and datatype
print(df.columns)
print(df.dtypes)
```

```
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')
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          object
company_location      object
company_size          object
dtype: object
```

Finding: We want to know if there is a significant difference in salaries between full-time and part-time positions, considering different experience levels

In [63]:
```
#df_melted = df.melt(id_vars='experience_level', var_name='employment_type', value_name='salary_in_usd')
# Plot the grouped bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='experience_level', y='salary_in_usd', hue='employment_type')
plt.title('Salary Variation by Experience Level and Employment Type')
plt.xlabel('Experience Level')
plt.ylabel('Salary')
plt.tight_layout()
```



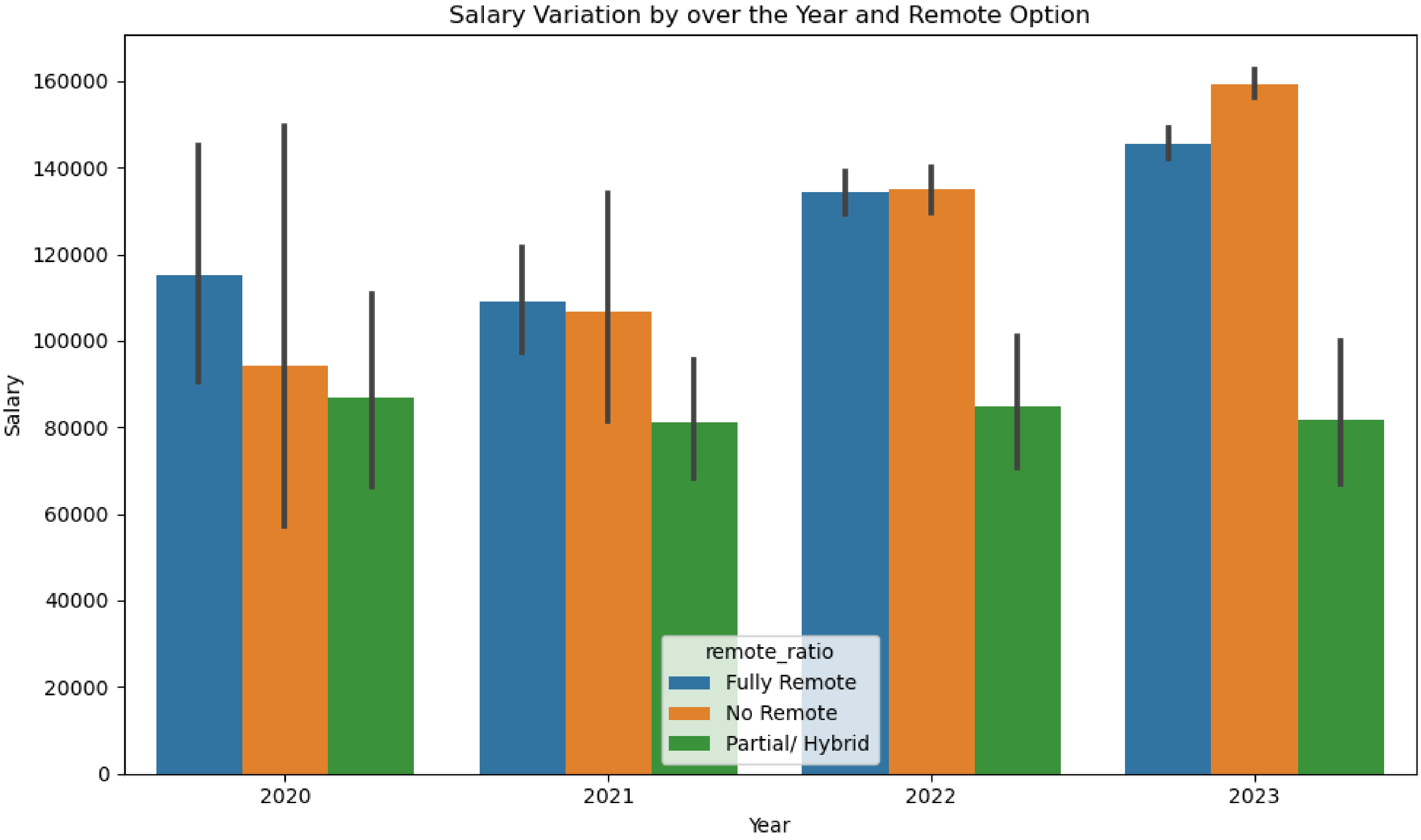
INSIGHT: The observed trend reveals that both experience level and employment type significantly impact employee salaries. Full-time employees generally earn the most across experience levels, except for Executive level employees on Contract, who earn approximately twice as much as other employees in the dataset

Finding: Are there any notable trends or patterns in salary fluctuations over the years, especially concerning changes in remote work policies or economic factors?

In [64]:

```
# Plot the grouped bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='work_year', y='salary_in_usd', hue='remote_ratio')
```

```
plt.title('Salary Variation by over the Year and Remote Option')
plt.xlabel('Year')
plt.ylabel('Salary')
#plt.xticks(rotation=45)
plt.tight_layout()
```

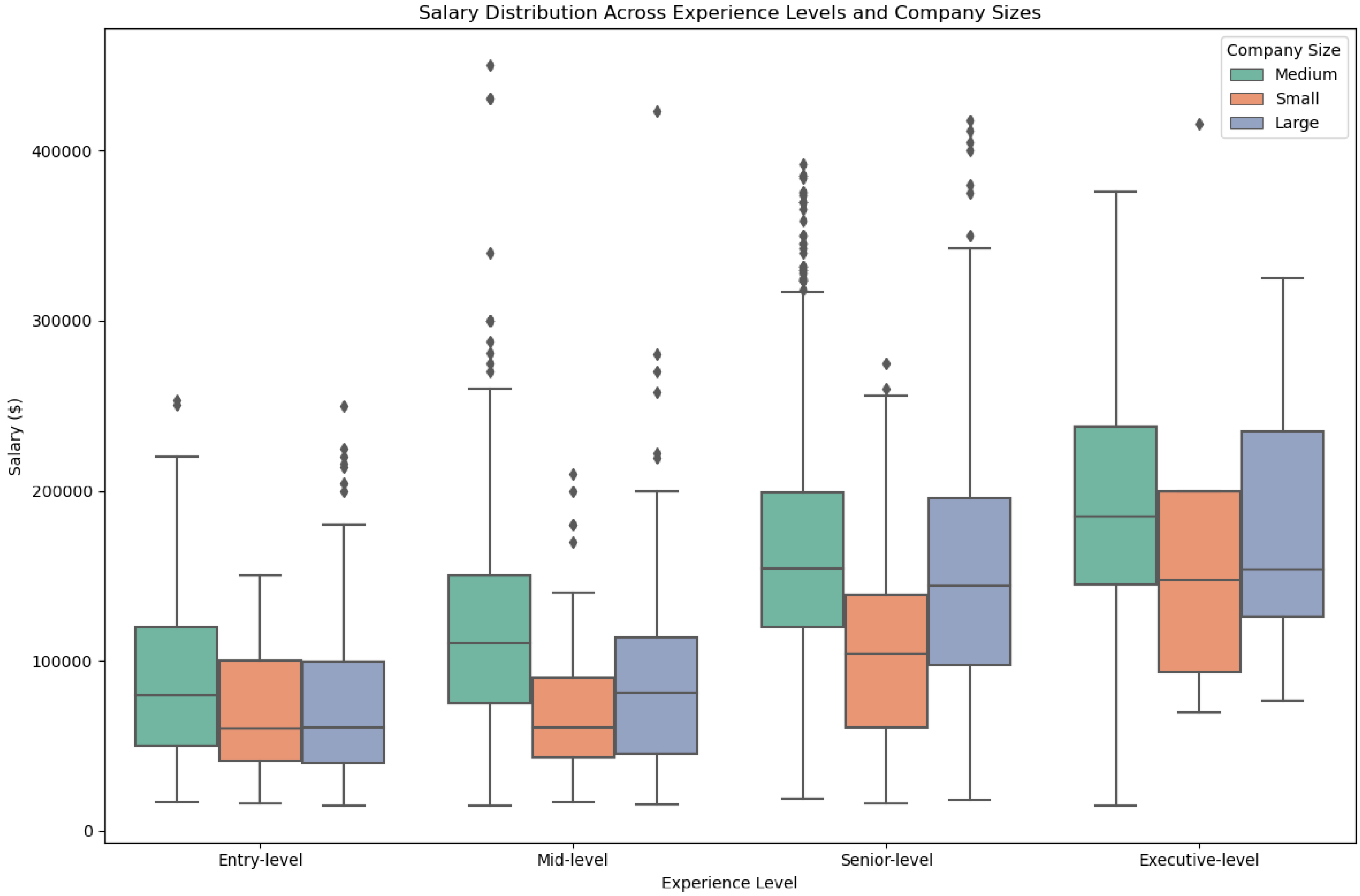


INSIGHT: The report highlights notable trends in employee salaries based on work modes over the years. In 2020, remote workers earned the most, likely due to widespread adoption of fully remote work during the Covid-19 lockdown. Subsequently, in 2021, remote workers' salaries decreased, possibly due to economic challenges. By 2022, salaries for both fully remote and onsite workers were comparable, with onsite workers surpassing remote workers' salaries in 2023. Overall, there's a significant growth in employee salaries within the AI/ML and Big Data industries.

Findings: We want to know if salaries differ significantly based on company size, and how does this relationship vary across different experience levels?

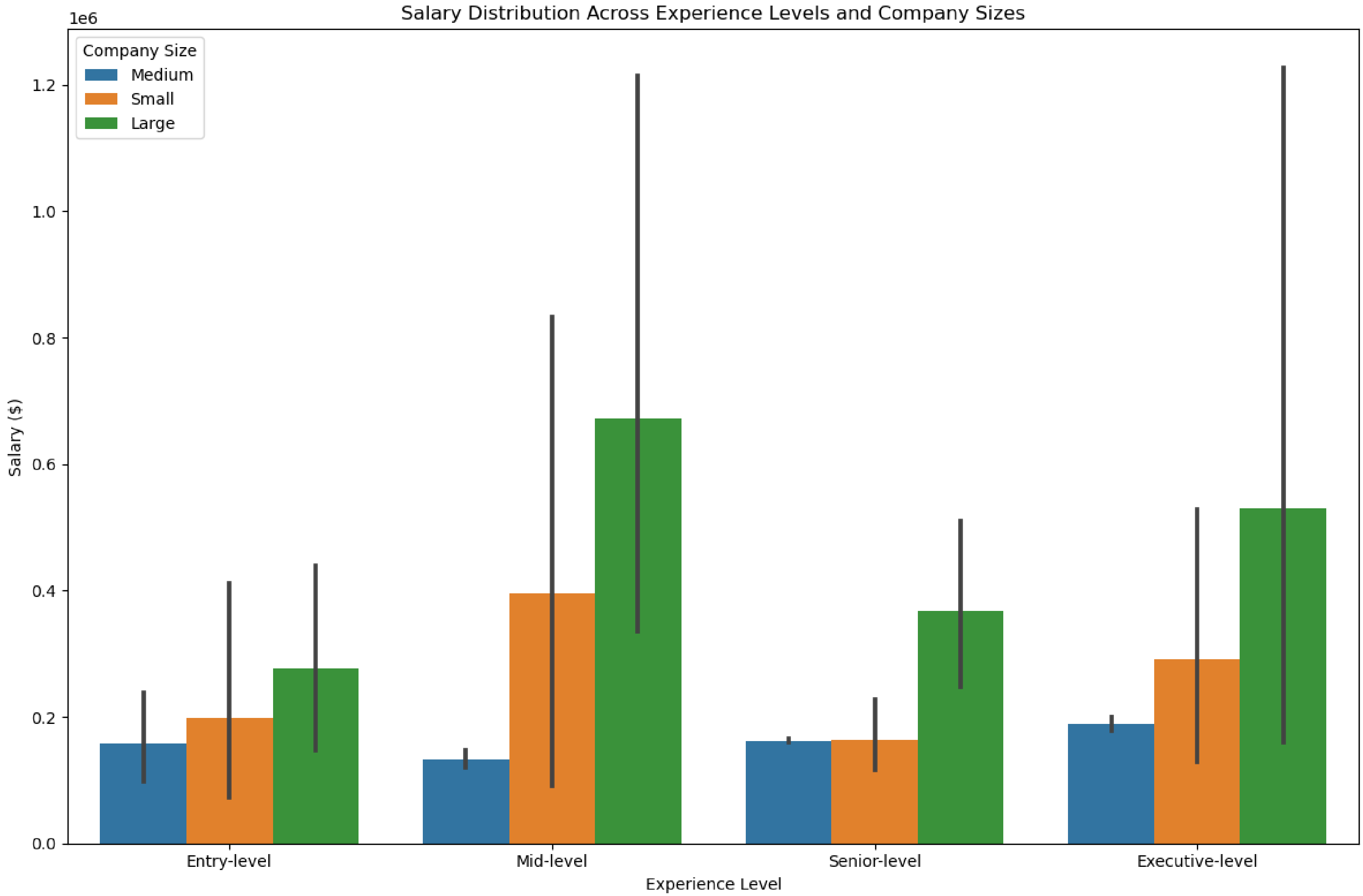
In [65]:

```python
#Using boxplot
plt.figure(figsize=(12, 8))
#sns.set_style('whitegrid')
sns.boxplot(data=df, x='experience_level', y='salary_in_usd', hue='company_size', palette='Set2')
plt.title('Salary Distribution Across Experience Levels and Company Sizes')
plt.xlabel('Experience Level')
plt.ylabel('Salary ($)')
plt.legend(title='Company Size')
plt.tight_layout()
```



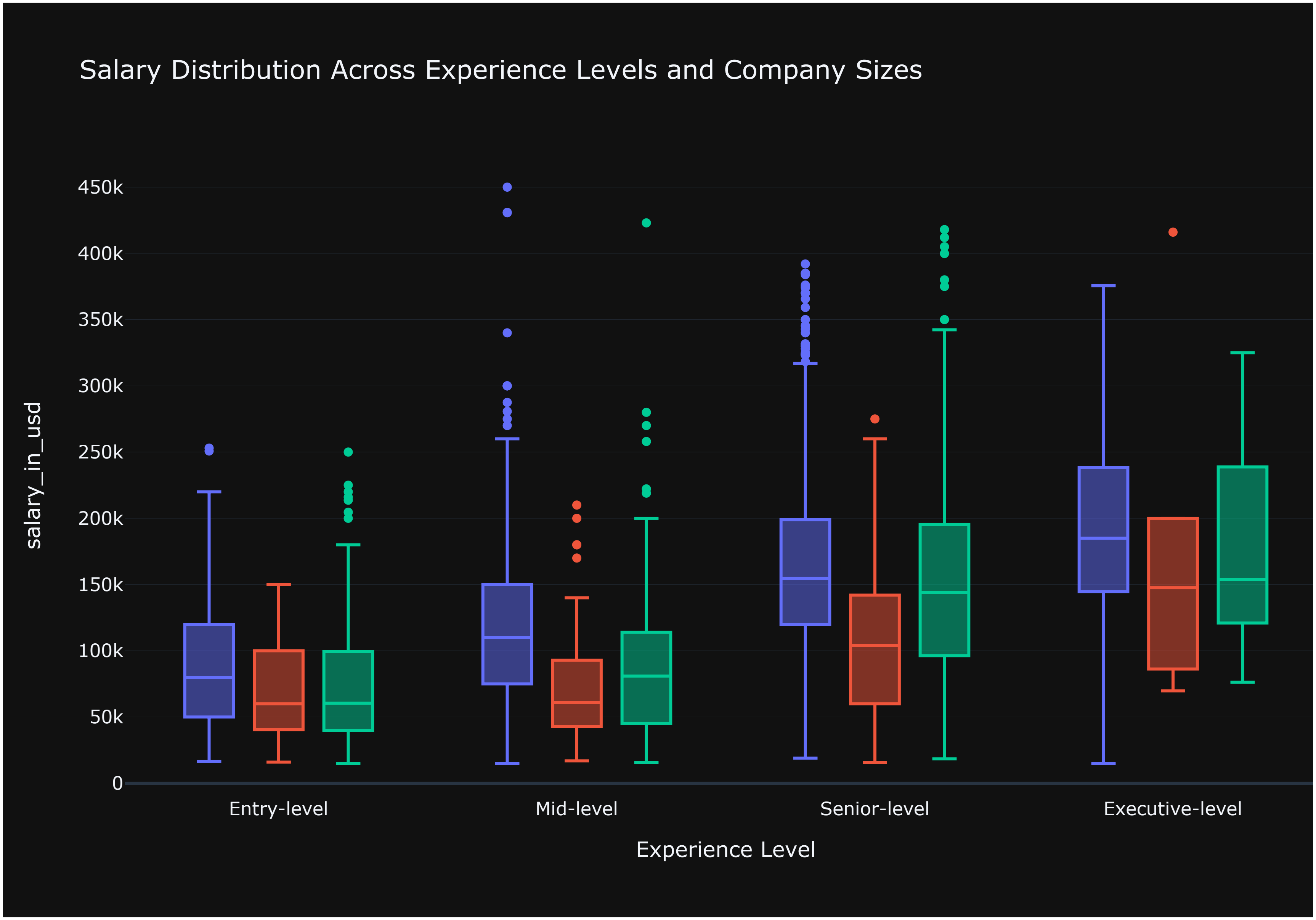
In [66]:

```
#Using barplot
plt.figure(figsize=(12, 8))
#sns.set_style('whitegrid')
sns.barplot(data=df, x='experience_level', y='salary', hue='company_size')
plt.title('Salary Distribution Across Experience Levels and Company Sizes')
plt.xlabel('Experience Level')
plt.ylabel('Salary ($)')
plt.legend(title='Company Size')
plt.tight_layout()
plt.show()
```



In [67]:

```
# Plot the box plot using Plotly Express
fig = px.box(df, x='experience_level', y='salary_in_usd', color='company_size',
             template='plotly_dark',
             title='Salary Distribution Across Experience Levels and Company Sizes',
             labels={'experience_level': 'Experience Level', 'salary': 'Salary ($)', 'company_size': 'Company Size'})
fig.update_layout(xaxis={'title': 'Experience Level'},
                  width=1000,  # Set the width of the plot
    height=600  # Set the height of the plot
                   )
fig.show()
```

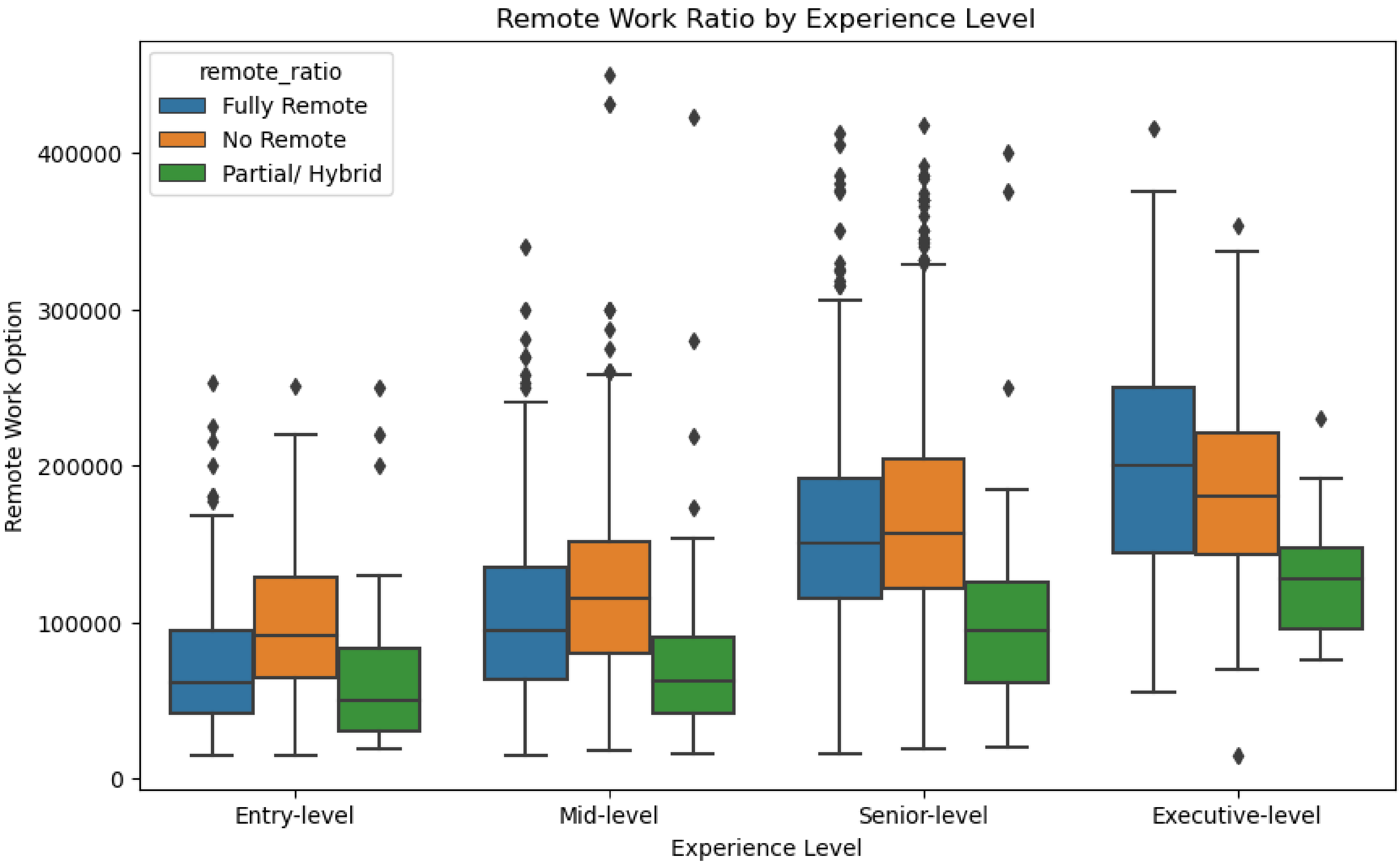


INSIGHT: It is observed that Medium sized company pays the highest salary accross the experience levels

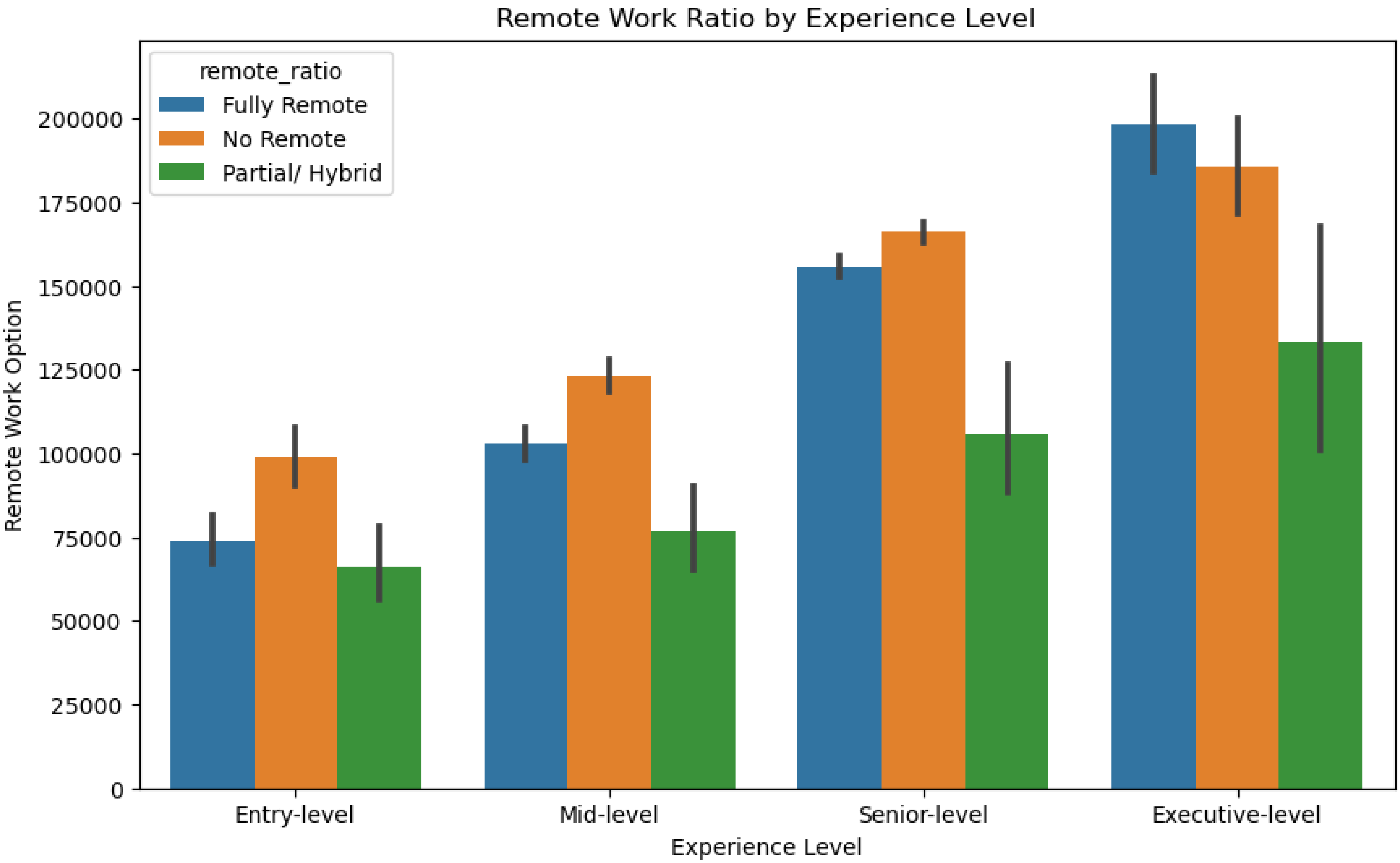
Finding: Do employees with higher experience levels tend to work more remotely, and how does this affect their salaries?

In [68]:

```
# Plot the box plot
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='experience_level', y='salary_in_usd',hue= "remote_ratio")
plt.title('Remote Work Ratio by Experience Level')
plt.xlabel('Experience Level')
plt.ylabel('Remote Work Option');
```



In [69]:
```
# Using seaborn, Plot the bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='experience_level', y='salary_in_usd',hue= "remote_ratio")
plt.title('Remote Work Ratio by Experience Level')
plt.xlabel('Experience Level')
plt.ylabel('Remote Work Option');
```

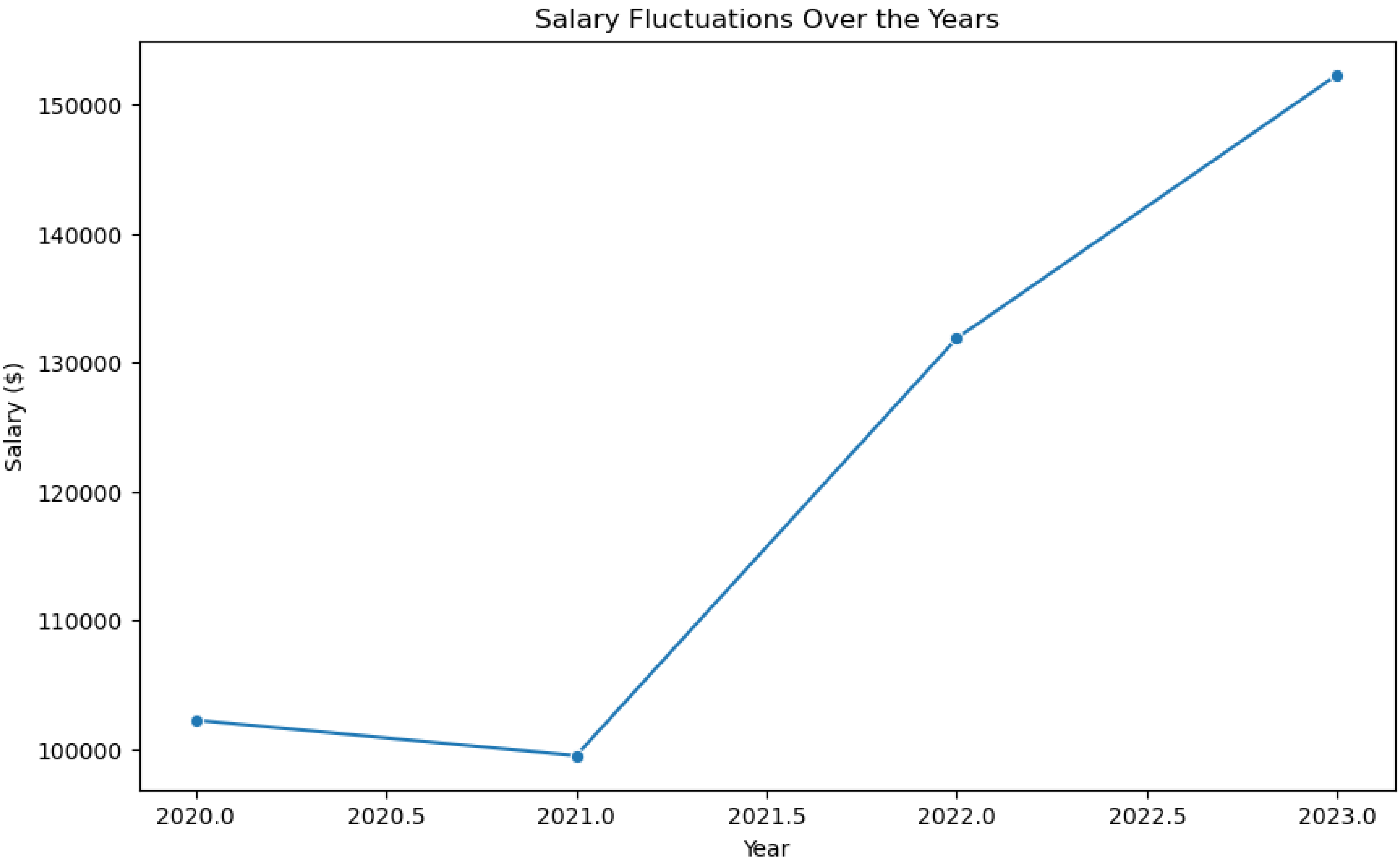


INSIGHT: Yes, The analysis indicates that Executive-level employees prefer remote work arrangements and typically earn the highest salaries.

Finding: Are there any notable trends or patterns in salary fluctuations over the years, especially concerning changes in remote work?

In [70]:

```
# Using the seaborn, Plot the line plot
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x='work_year', y='salary_in_usd', marker='o', errorbar=None)
plt.title('Salary Fluctuations Over the Years')
plt.xlabel('Year')
plt.ylabel('Salary ($)');
```

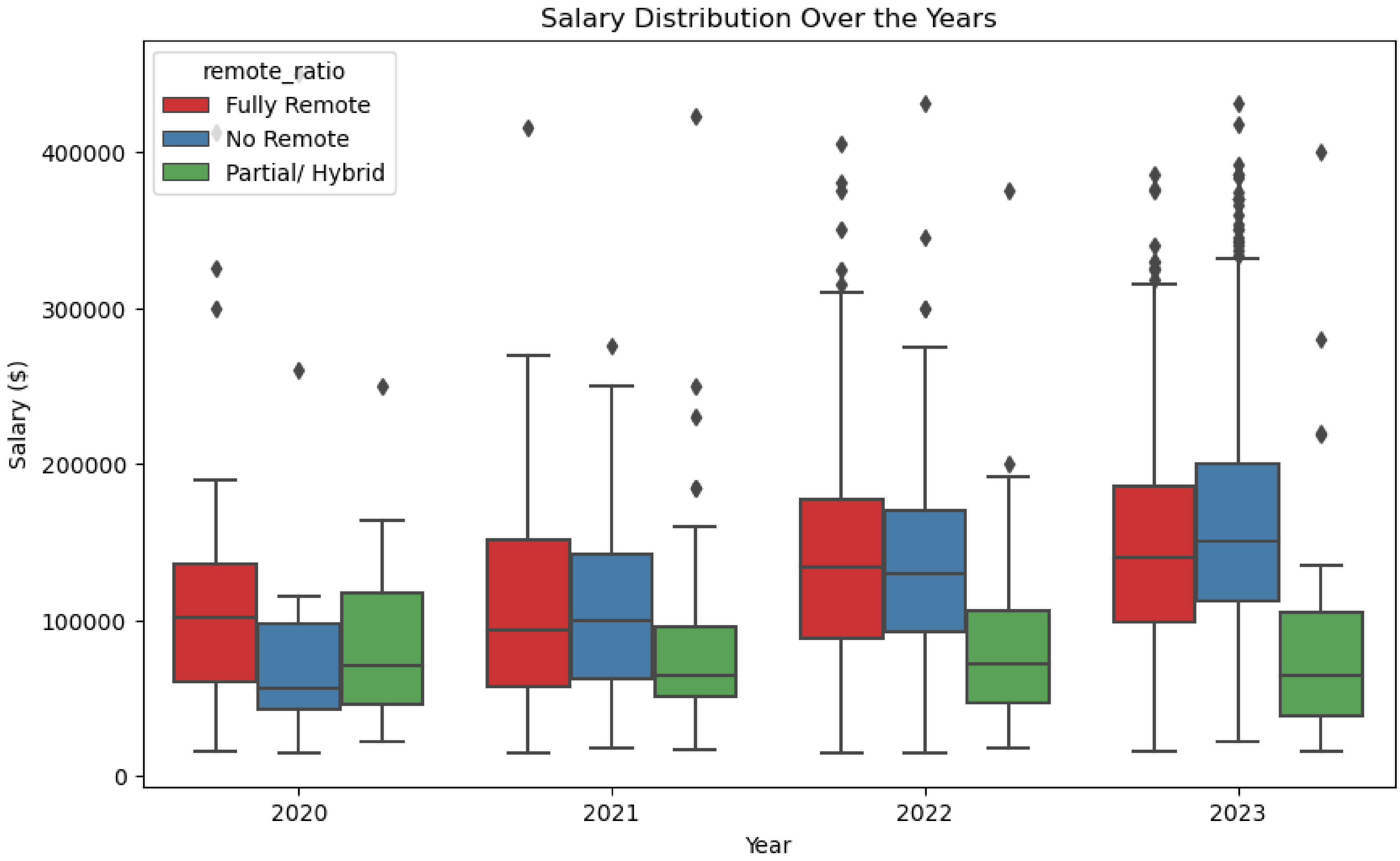


In [71]:

```
# Plot the box plot
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='work_year', y='salary_in_usd', hue= "remote_ratio",palette='Set1')
plt.title('Salary Distribution Over the Years')
plt.xlabel('Year')
plt.ylabel('Salary ($)')
```

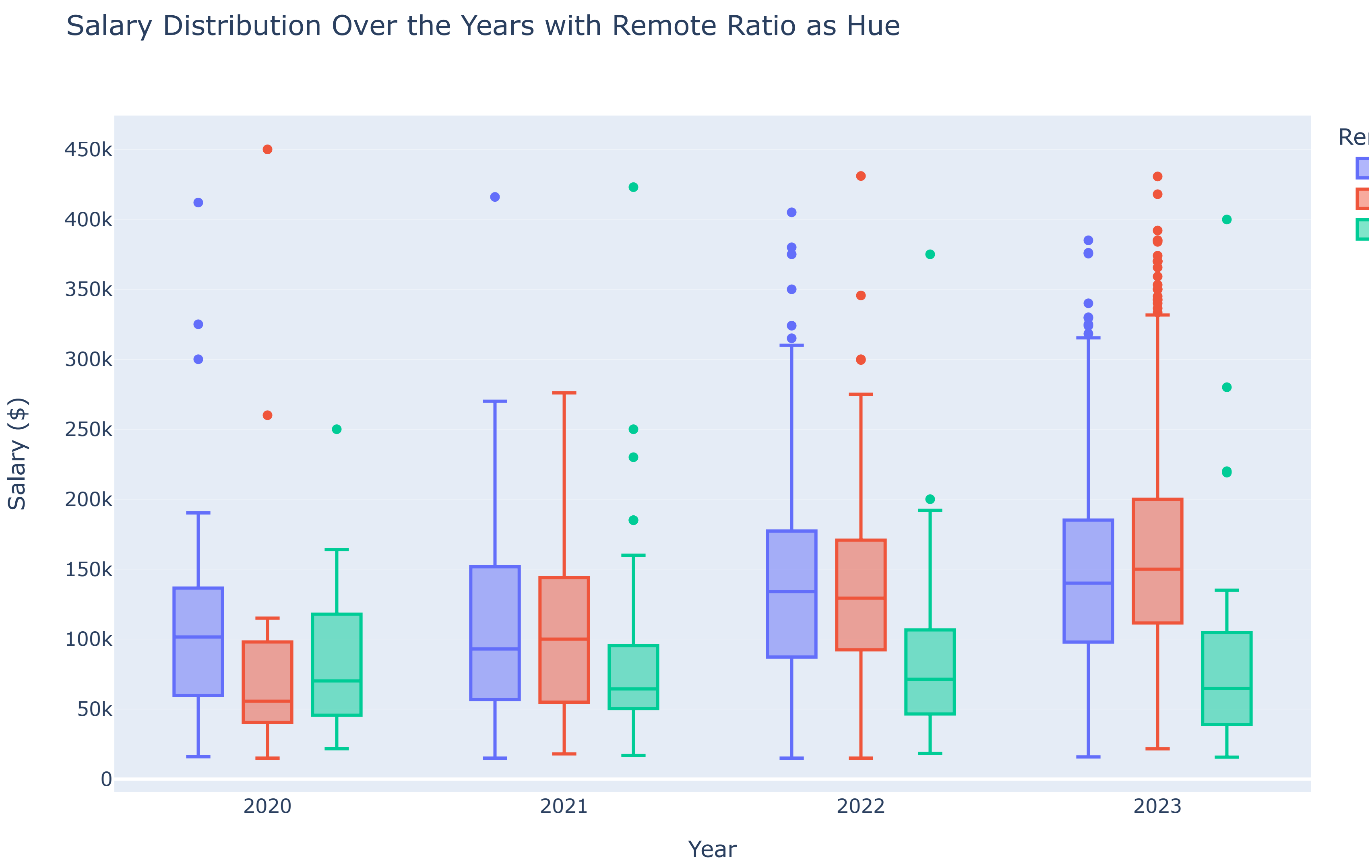
Out[71]:

```
Text(0, 0.5, 'Salary ($)')
```



In [72]:

```
# Plot the box plot using Plotly Express
fig = px.box(df, x='work_year', y='salary_in_usd', color='remote_ratio',
             title='Salary Distribution Over the Years with Remote Ratio as Hue',
             labels={'work_year': 'Year', 'salary_in_usd': 'Salary ($)', 'remote_ratio': 'Remote Work Option'})
fig.update_layout(xaxis={'title': 'Year'},
                 width=1000,  # Set the width of the plot
    height=600  # Set the height of the plot
                  )
fig.show()
```

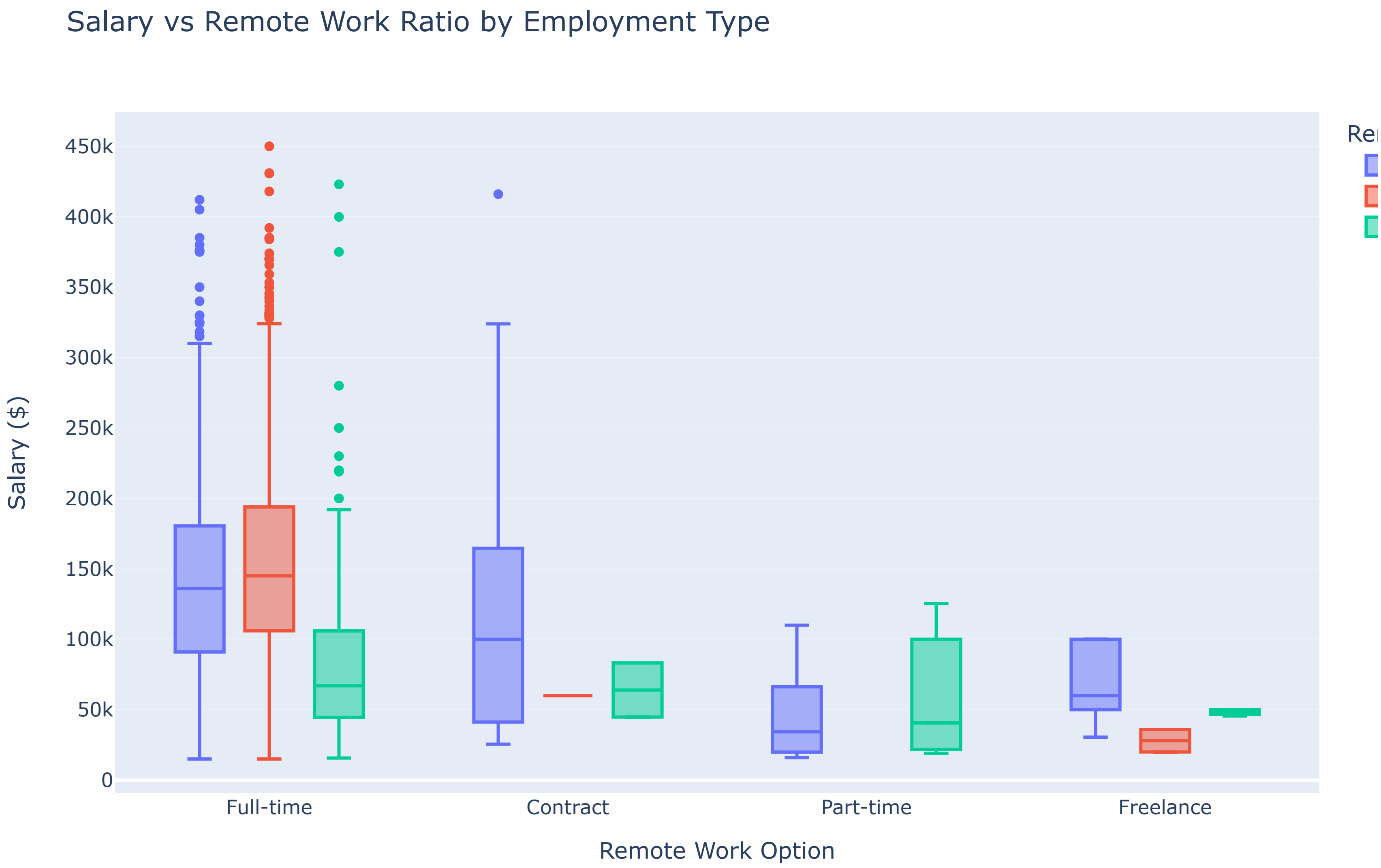

Salary Distribution Over the Years with Remote Ratio as Hue

INSIGHT: A downward trend in salaries between 2020 and 2021, possibly due to the Covid-19 lockdown, is observed. However, from 2021 to 2023, there is a significant rise in salaries on average. Additionally, remote workers enjoyed the highest pay between 2020 and 2022, but in 2023, onsite workers earned more than fully-remote and partially remote workers..

What is the correlation between salary and the remote work ratio, when considering different types of employment (full-time, part-time, contract, freelance)?

In [73]:

```
# Plot the violin plot using Plotly Express
fig = px.box(df, y='salary_in_usd', x='employment_type', color='remote_ratio',
                title='Salary vs Remote Work Ratio by Employment Type',
                labels={'salary_in_usd': 'Salary ($)', 'remote_ratio': 'Remote Work Option', 'experience_level': 'Exper
             width=1000,  # Set the width of the plot
    height=600  # Set the height of the plot
             )
fig.update_layout(xaxis={'title': 'Remote Work Option'})
fig.show()
```

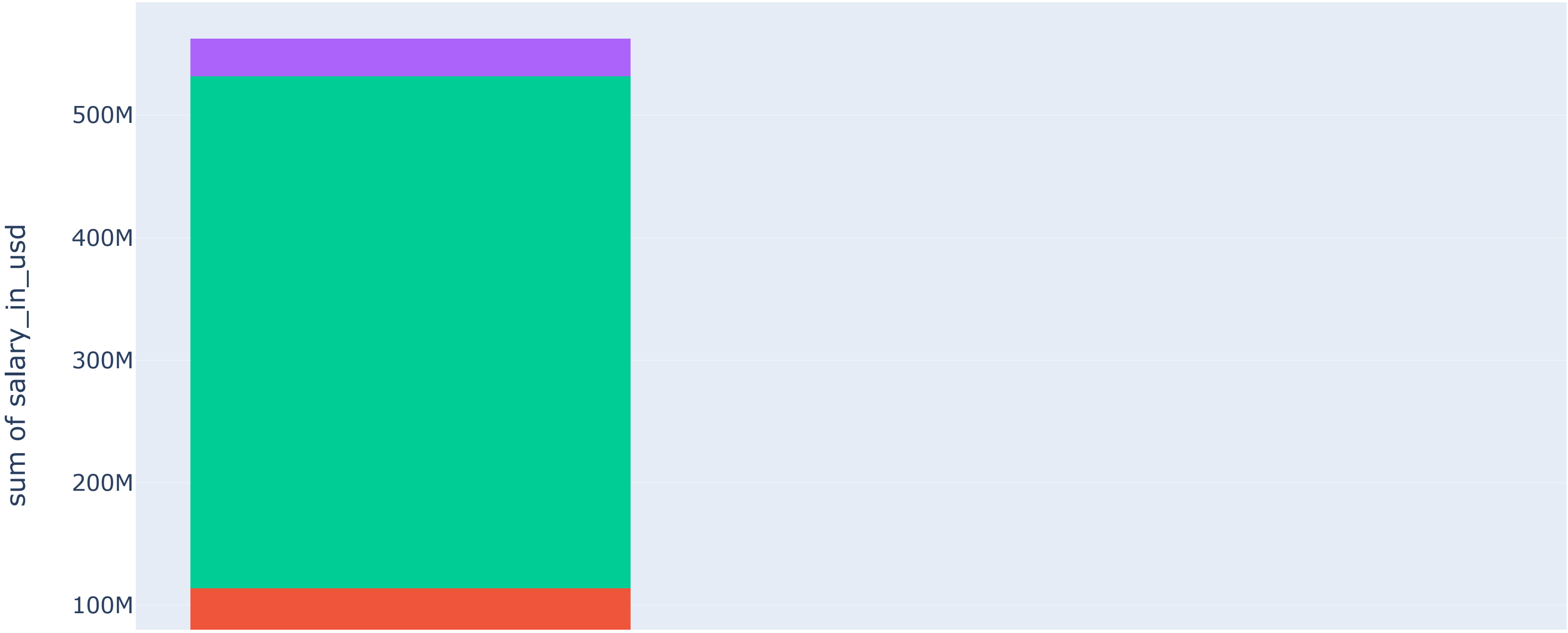

Salary vs Remote Work Ratio by Employment Type
Re
450k
400k
350k
300k
250k
200k
150k
100k
50k
0
Salary ($)
Full-time
Contract
Part-time
Freelance
Remote Work Option

INSIGHT: Within the full-time employment type category, onsite employees (no remote) earn the highest salary. However, in the contract category, fully remote employees earn the highest salary. In the part-time category, hybrid workers earn more than fully remote workers. In the freelance category, fully remote workers lead in terms of salary earned. Overall, there is a correlation between salary, employment type, and remote options.

How do salaries vary between different company sizes, and does this relationship hold consistent across various experience levels and job roles?

In [74]:

```
#Using plotly express
fig = px.histogram(df, y="salary_in_usd", x="company_size", color="experience_level")
fig.show()
```

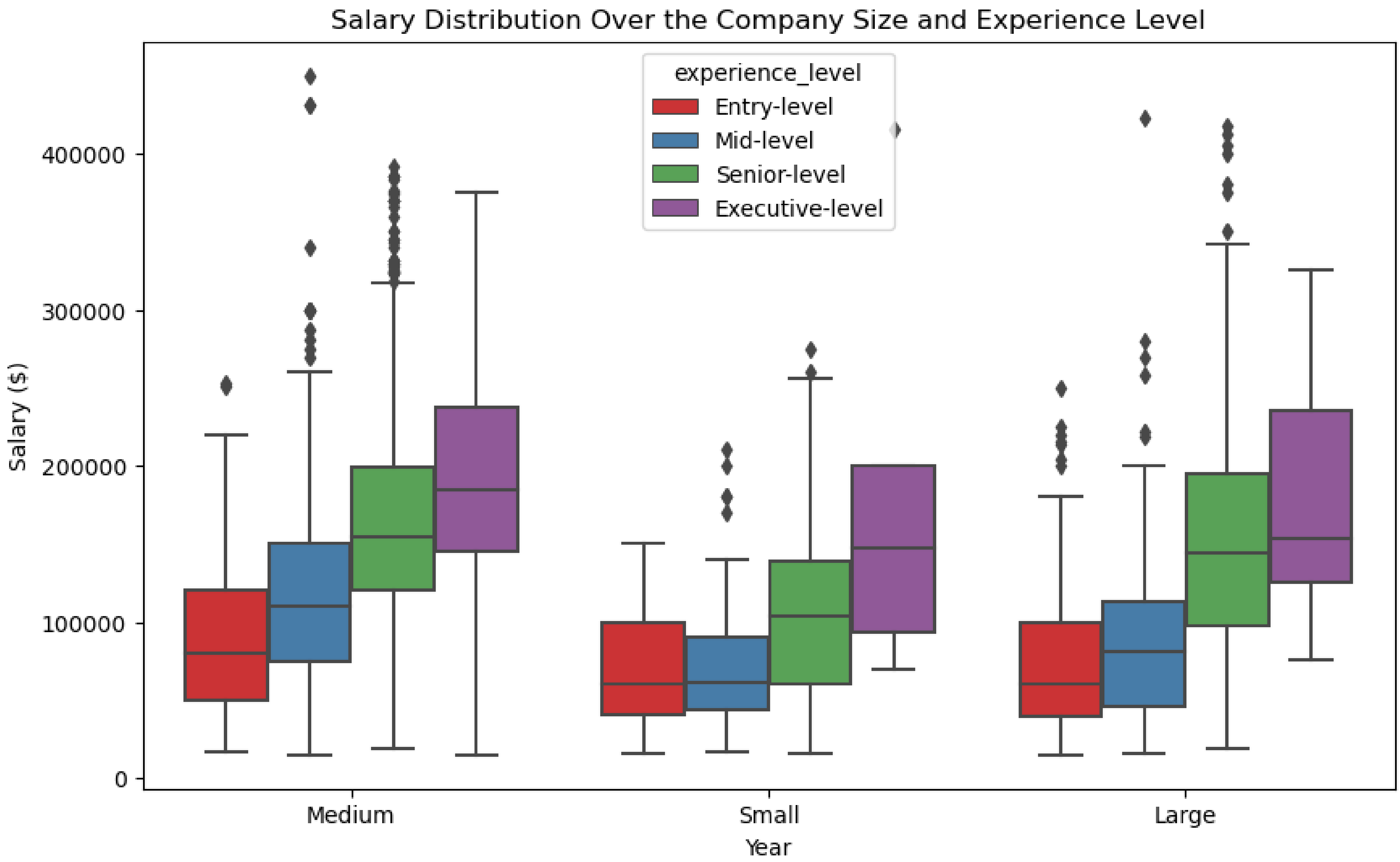


In [75]:

```
# Plot the box plot
plt.figure(figsize=(10, 6))
sns.boxplot(data=df, x='company_size', y='salary_in_usd', hue= "experience_level",palette='Set1')
plt.title('Salary Distribution Over the Company Size and Experience Level')
plt.xlabel('Year')
plt.ylabel('Salary ($)')
```

Out[75]:

```
Text(0, 0.5, 'Salary ($)')
```



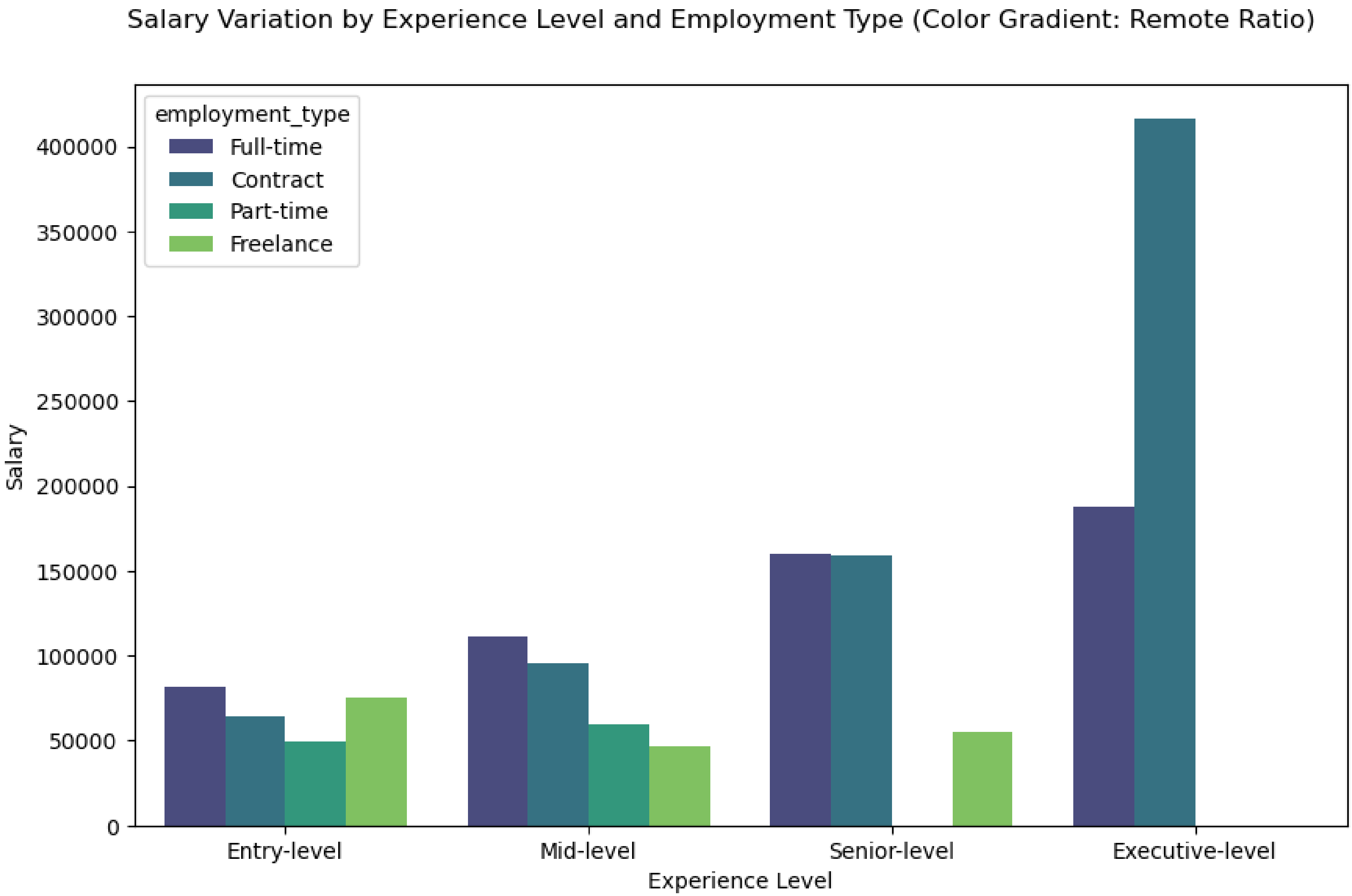
INSIGHTS: There is variation in salary in relation to the Company Size. Also employees salaries increases with their level of experience. This relationship hold consistent across the various experience levels.

Finding: How do salaries vary among different experience levels and types of employment within the AI/ML and Big Data space, while considering the remote work ratio?

```
In [76]: plt.figure(figsize=(10, 6))
         sns.barplot(data=df, x='experience_level', y='salary_in_usd', hue='employment_type', palette='viridis', ci=None)
         plt.suptitle('Salary Variation by Experience Level and Employment Type (Color Gradient: Remote Ratio)')
```

```python
plt.subplots_adjust(top=0.9)  # Adjust the subplot layout to accommodate the title
plt.xlabel('Experience Level')
plt.ylabel('Salary');
```

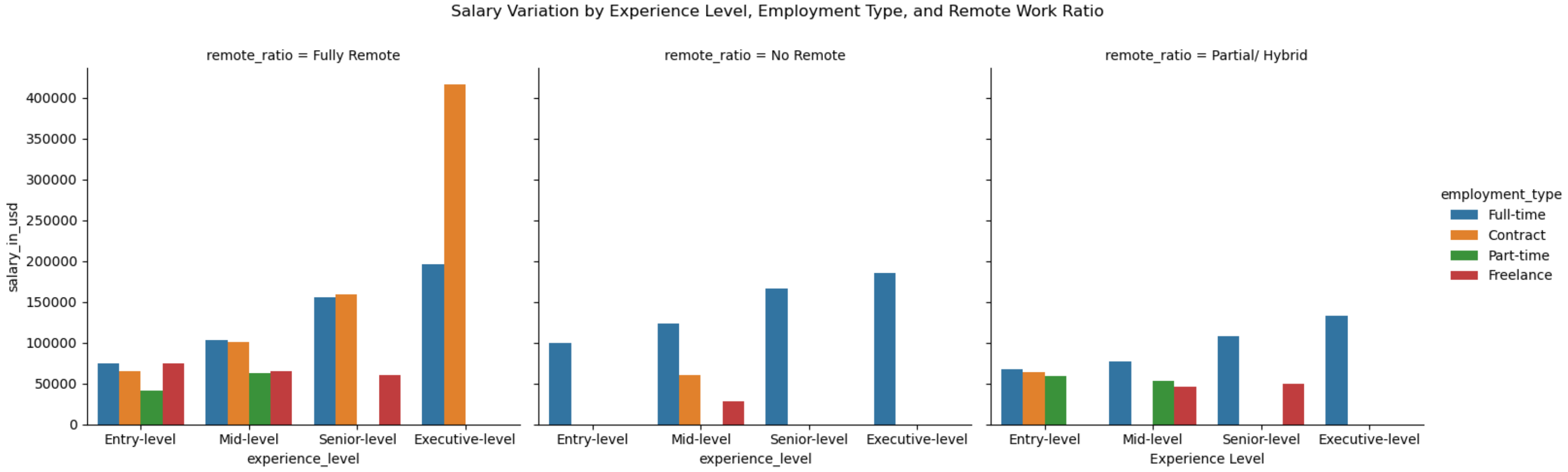
```
C:\Users\HP\AppData\Local\Temp\ipykernel_40964\409282422.py:2: FutureWarning:



The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.
```



In [77]:
```
# Faceted bar plot with color encoding for remote_ratio
plt.figure(figsize=(12, 6))
sns.catplot(data=df, x='experience_level', y='salary_in_usd', hue='employment_type', col='remote_ratio', kind='bar',err
plt.suptitle('Salary Variation by Experience Level, Employment Type, and Remote Work Ratio')
plt.subplots_adjust(top=0.85)  # Adjust the subplot layout to accommodate the title
plt.xlabel('Experience Level')
plt.ylabel('Salary');
```

```
C:\Users\HP\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```

```
<Figure size 1200x600 with 0 Axes>
```



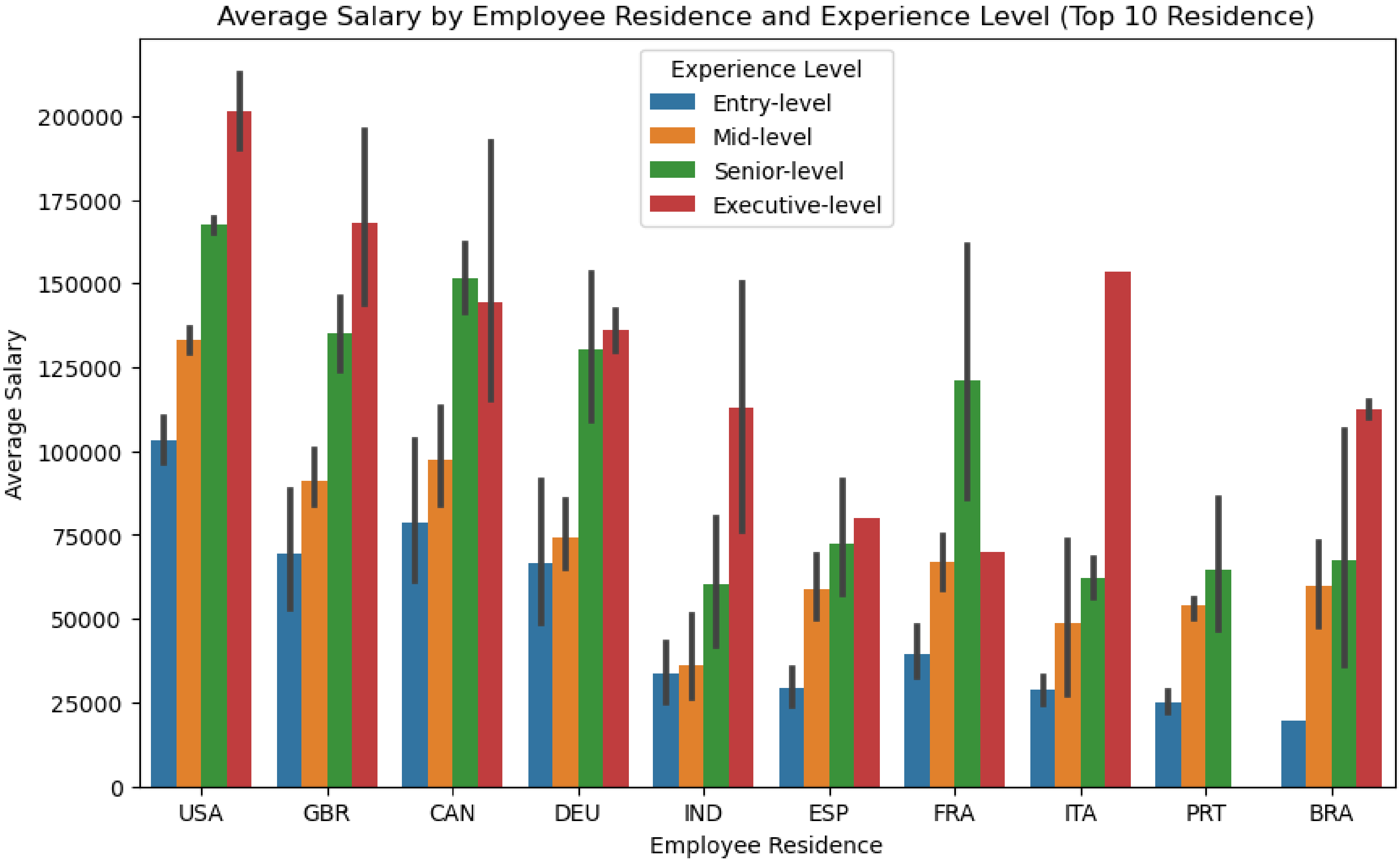
INSIGHT: Based on the remote ratio options. There is a clear indication that On-site(No Remote) workers earn the highest salaries across all experience levels. Also, employement type influences the employees salary. Employees who were employed on full time basis earns the highest salaries

Finding: Is there a significant difference in salaries based on the employee's primary country of residence, company's main office location, and their respective experience levels?

In [78]:
```
df['experience_level'].unique()
```

Out[78]:
```
array(['Entry-level', 'Mid-level', 'Senior-level', 'Executive-level'],
      dtype=object)
```

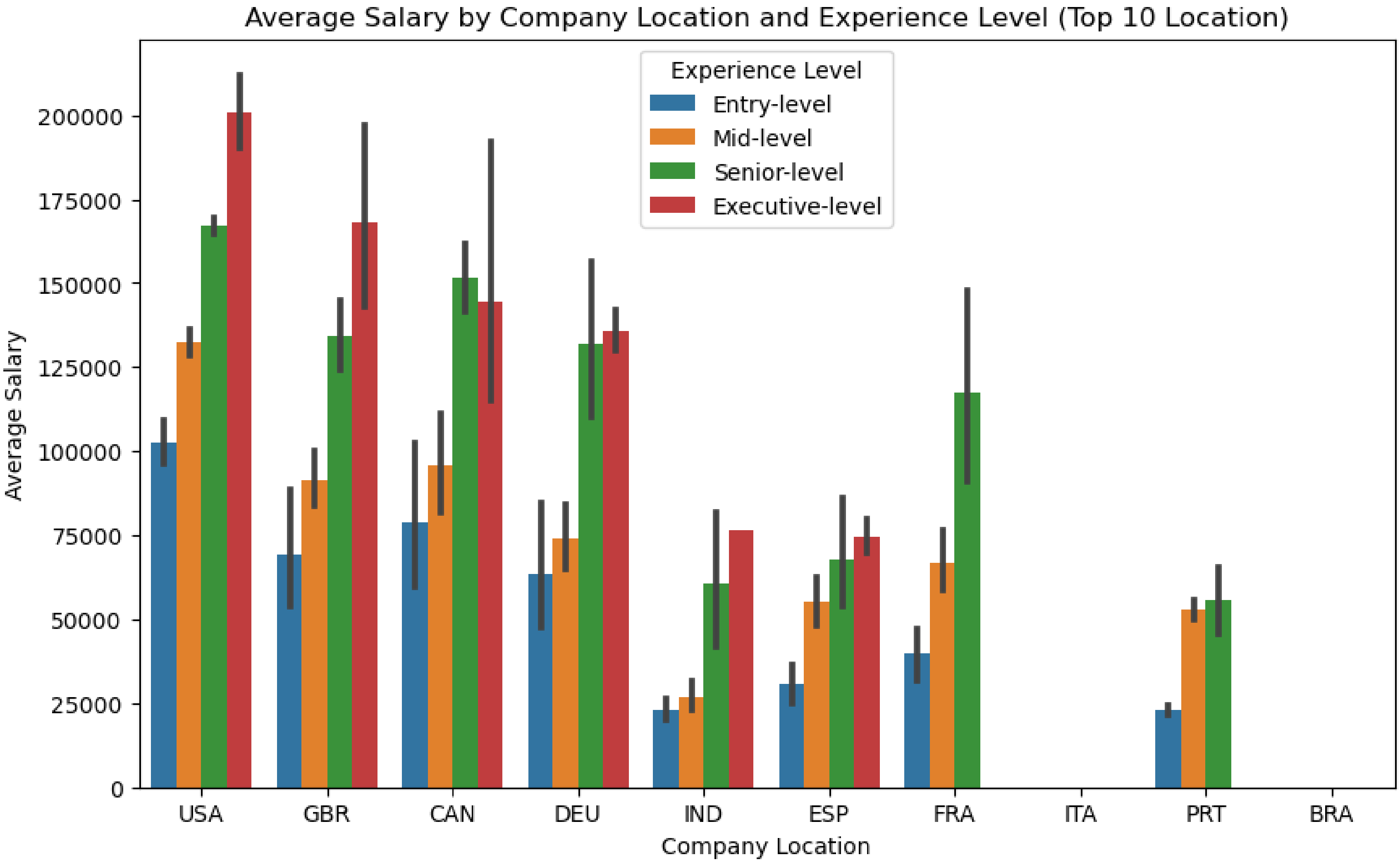
In [79]:
```
# Filter top 10 employee residences
top_employee_residence = df['employee_residence'].value_counts().nlargest(10).index
df_top = df[df['employee_residence'].isin(top_employee_residence)]


# Create the bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=df_top, x='employee_residence', y='salary_in_usd', hue='experience_level',
            order=top_employee_residence,
            hue_order=['Entry-level', 'Mid-level', 'Senior-level', 'Executive-level'])
plt.title('Average Salary by Employee Residence and Experience Level (Top 10 Residence)')
plt.xlabel('Employee Residence')
plt.ylabel('Average Salary')
plt.legend(title='Experience Level');
```



INSIGHT: Yes, there is a significant difference in salaries based on the employee's primary country of residence, and their respective experience levels. The analysis of the data revealed that there is difference in employees salary based on the employees residence and experience level. Employees who are resident in USA are the highest paid among the top 10 countries.On average, the executive level experience type benefits from high salaries

In [80]:

```
# Filter top 10 employee residences
top_companyLocation = df['company_location'].value_counts().nlargest(10).index
df_top = df[df['employee_residence'].isin(top_companyLocation)]


# Create the bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=df_top, x='company_location', y='salary_in_usd', hue='experience_level',
            order=top_employee_residence,
            hue_order=['Entry-level', 'Mid-level', 'Senior-level', 'Executive-level'])
plt.title('Average Salary by Company Location and Experience Level (Top 10 Location)')
plt.xlabel('Company Location')
plt.ylabel('Average Salary')
plt.legend(title='Experience Level');
```



INSIGHT: There is a significant difference in salaries based on the company's main office location, and their respective experience levels.The analysis of the data revealed that there is difference in employees salary based on the company's location and experience level. On average, the executive level experience type benefits from high salaries

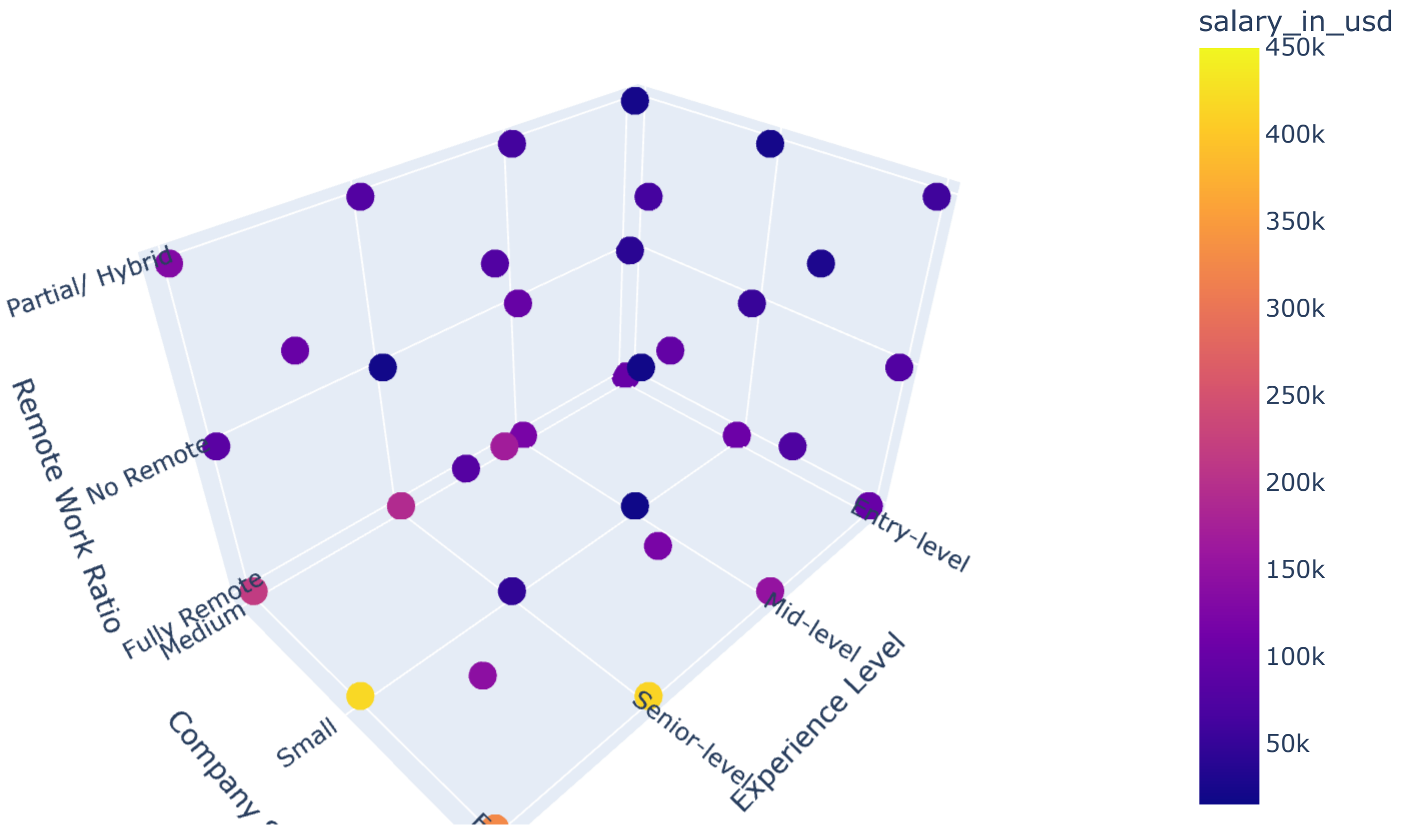
In conclusion, there is a significant difference in salaries based on the employee's primary country of residence, company's main office location, and their respective experience levels. Based on this analysis, USA tops the salary ranking in terms of employees residence and company location.

Finding: What are the salary trends when analyzing the interaction between employee's experience levels, company sizes, and remote work ratios?

In [81]:

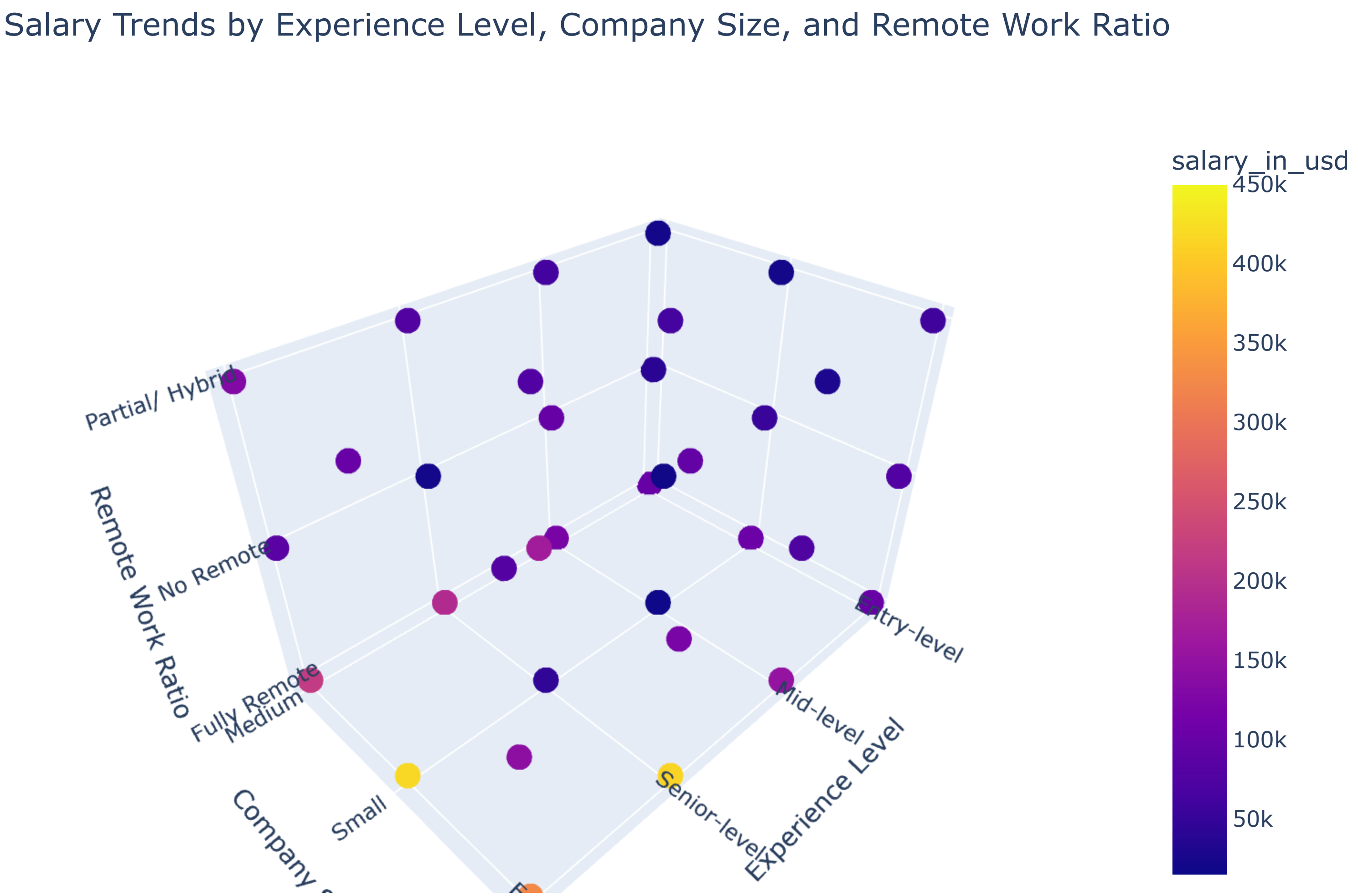
```
# Create scatter plot with color encoding for salary
fig = px.scatter_3d(df, x='experience_level', y='company_size', z='remote_ratio', color='salary_in_usd',
                    labels={'experience_level': 'Experience Level', 'company_size': 'Company Size', 'remote_ratio': 'Re
                    title='Salary Trends by Experience Level, Company Size, and Remote Work Ratio',
                    width=800,  # Set the width of the plot
    height=600  # Set the height of the plot
                    )
fig.show()
```

Salary Trends by Experience Level, Company Size, and Remote Work Ratio



What are the salary trends when analyzing the interaction between employee's experience levels, company sizes, and remote work ratios?

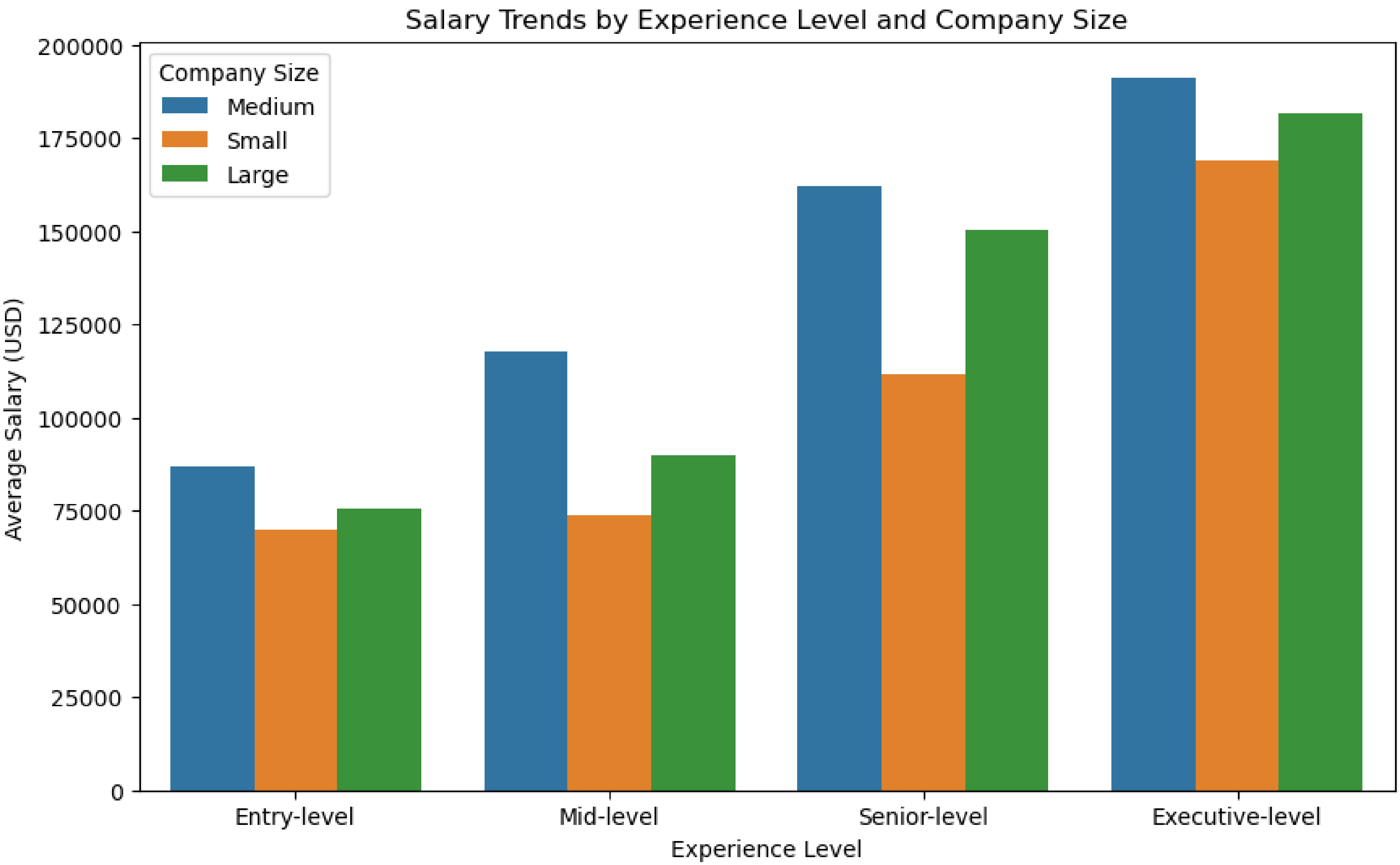
In [82]:
```
# Create scatter plot with color encoding for salary
fig = px.scatter_3d(df, x='experience_level', y='company_size', z='remote_ratio', color='salary_in_usd',
                    labels={'experience_level': 'Experience Level', 'company_size': 'Company Size', 'remote_ratio': 'Re
                    title='Salary Trends by Experience Level, Company Size, and Remote Work Ratio',
                   width=800,  # Set the width of the plot
    height=600  # Set the height of the plot
                    )
fig.show()
```



In [83]:

```
# Grouped bar plot
plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='experience_level', y='salary_in_usd', hue='company_size', ci=None)
plt.title('Salary Trends by Experience Level and Company Size')
plt.xlabel('Experience Level')
plt.ylabel('Average Salary (USD)')
plt.legend(title='Company Size')
plt.show()
```

```
C:\Users\HP\AppData\Local\Temp\ipykernel_40964\2957686620.py:3: FutureWarning:



The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.
```



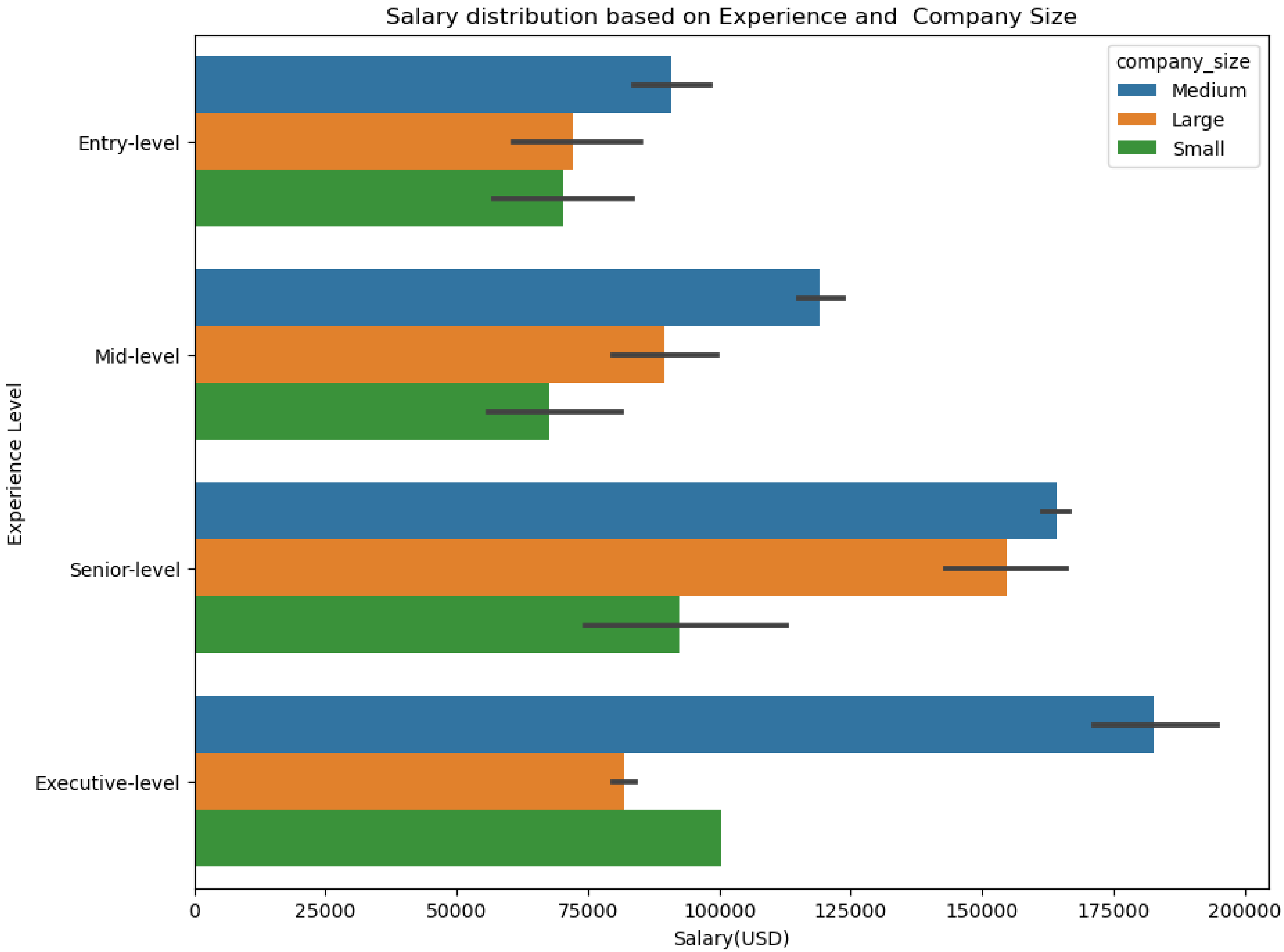
Finding: Are there any noticeable patterns in salary distributions when examining the relationship between employee experience levels, company sizes, remote work ratios, and the specific job titles or roles held by the employees?

The tasks in two steps:

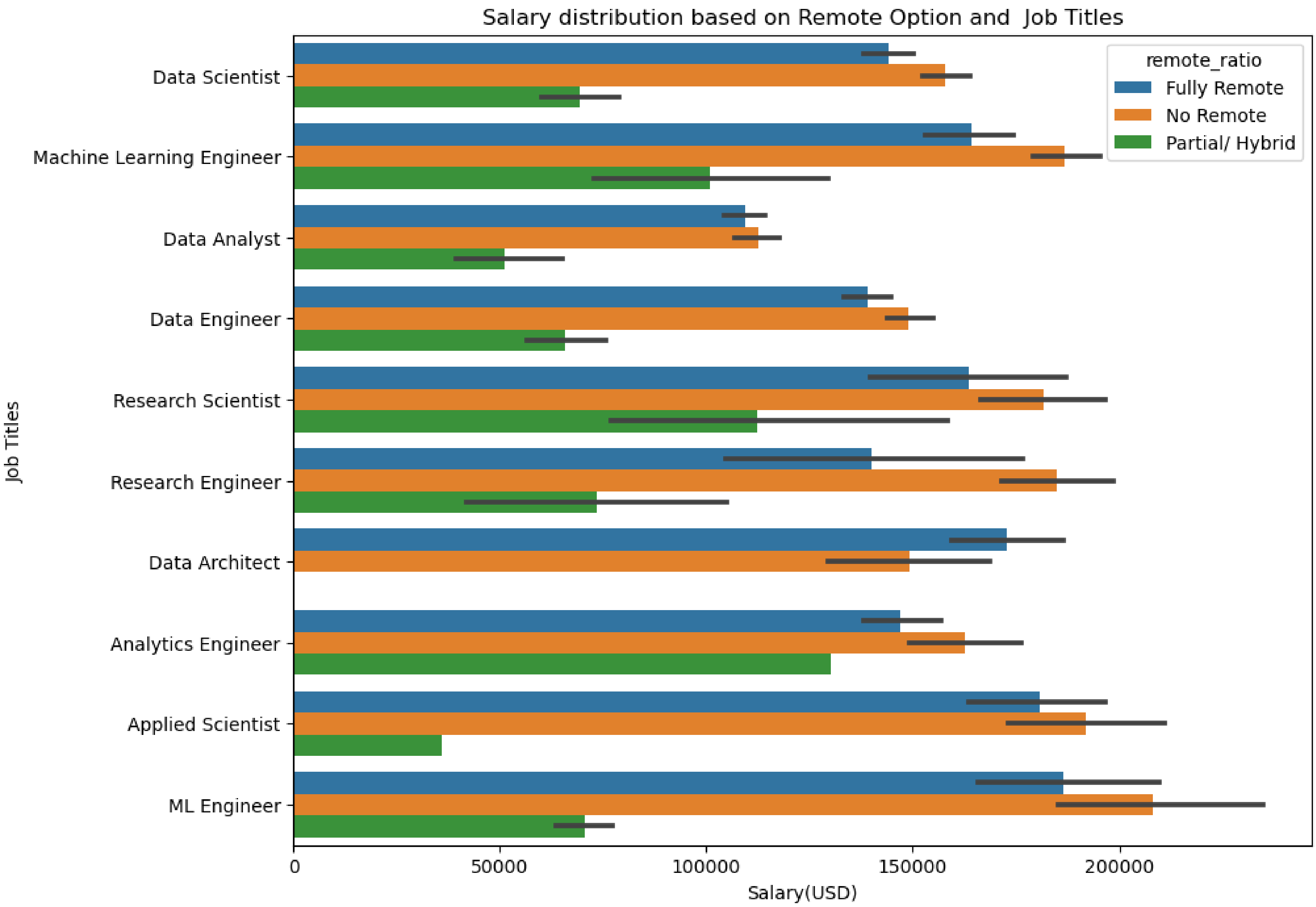
(1)salary distribution by employee levels and company size (2)salary distribution by remote ratios and top10 job titles

In [84]:
```
top_10_job_titles= df['job_title'].value_counts()[:10].index

# Filter the dataset for relevant job titles and experience levels
```

```python
relevant_job_titles = top_10_job_titles
relevant_exp_levels = ['Entry-level', 'Mid-level', 'Senior-level', 'Executive-level']
df_filtered = df[(df['job_title'].isin(relevant_job_titles)) & (df['experience_level'].isin(relevant_exp_levels))]

plt.figure(figsize=(10, 8))
sns.barplot(data=df_filtered,  x='salary_in_usd', y='experience_level', hue="company_size")
plt.title("Salary distribution based on Experience and  Company Size")
plt.xlabel("Salary(USD)")
plt.ylabel("Experience Level");
```



```
In [85]: plt.figure(figsize=(10, 8))
         sns.barplot(data=df_filtered,  x='salary_in_usd', y='job_title', hue="remote_ratio")
         plt.title("Salary distribution based on Remote Option and  Job Titles")
         plt.xlabel("Salary(USD)")
         plt.ylabel("Job Titles");
```

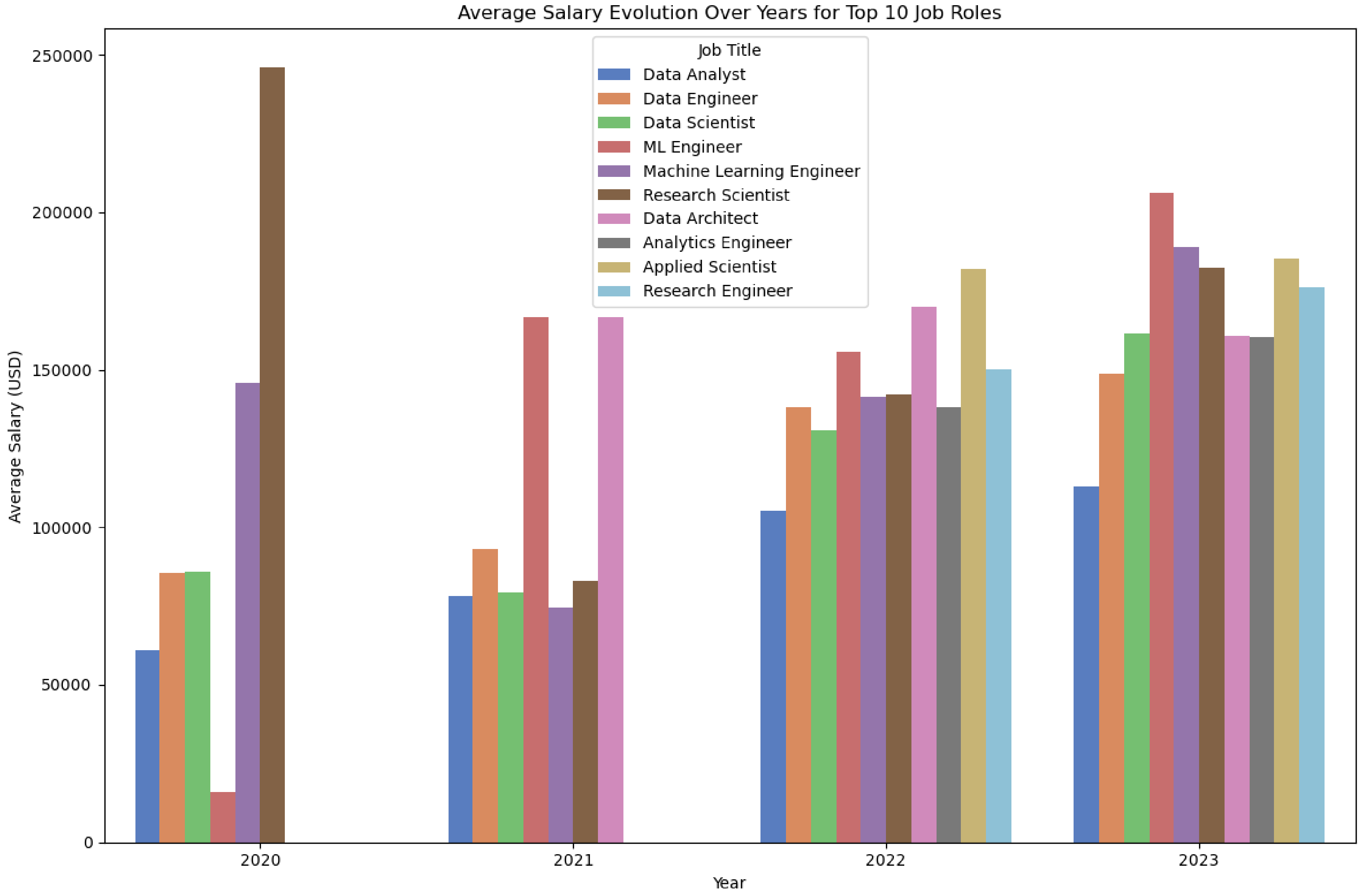


Insight: The analysis shows that among the top 10 job titles, employees working in medium-sized companies received the highest salaries, and those who worked on-site (without remote work) earned the highest salaries overall.

Finding: How has the salary evolved over the years among the top 10 Job roles in the AI/ML and Big Data Industries?

In [86]:

```
# Filter data for top 10 job roles
top_10_jobs = df['job_title'].value_counts().head(10).index
df_top_10 = df[df['job_title'].isin(top_10_jobs)]

# Group by year and job title to calculate average salary
df_avg_salary = df_top_10.groupby(['work_year', 'job_title'])['salary_in_usd'].mean().reset_index()

# Plot evolution over years using a grouped bar plot
plt.figure(figsize=(12, 8))
sns.barplot(data=df_avg_salary, x='work_year', y='salary_in_usd', hue='job_title', palette='muted')
plt.title('Average Salary Evolution Over Years for Top 10 Job Roles')
plt.xlabel('Year')
plt.ylabel('Average Salary (USD)')
plt.legend(title='Job Title')
plt.tight_layout()
```

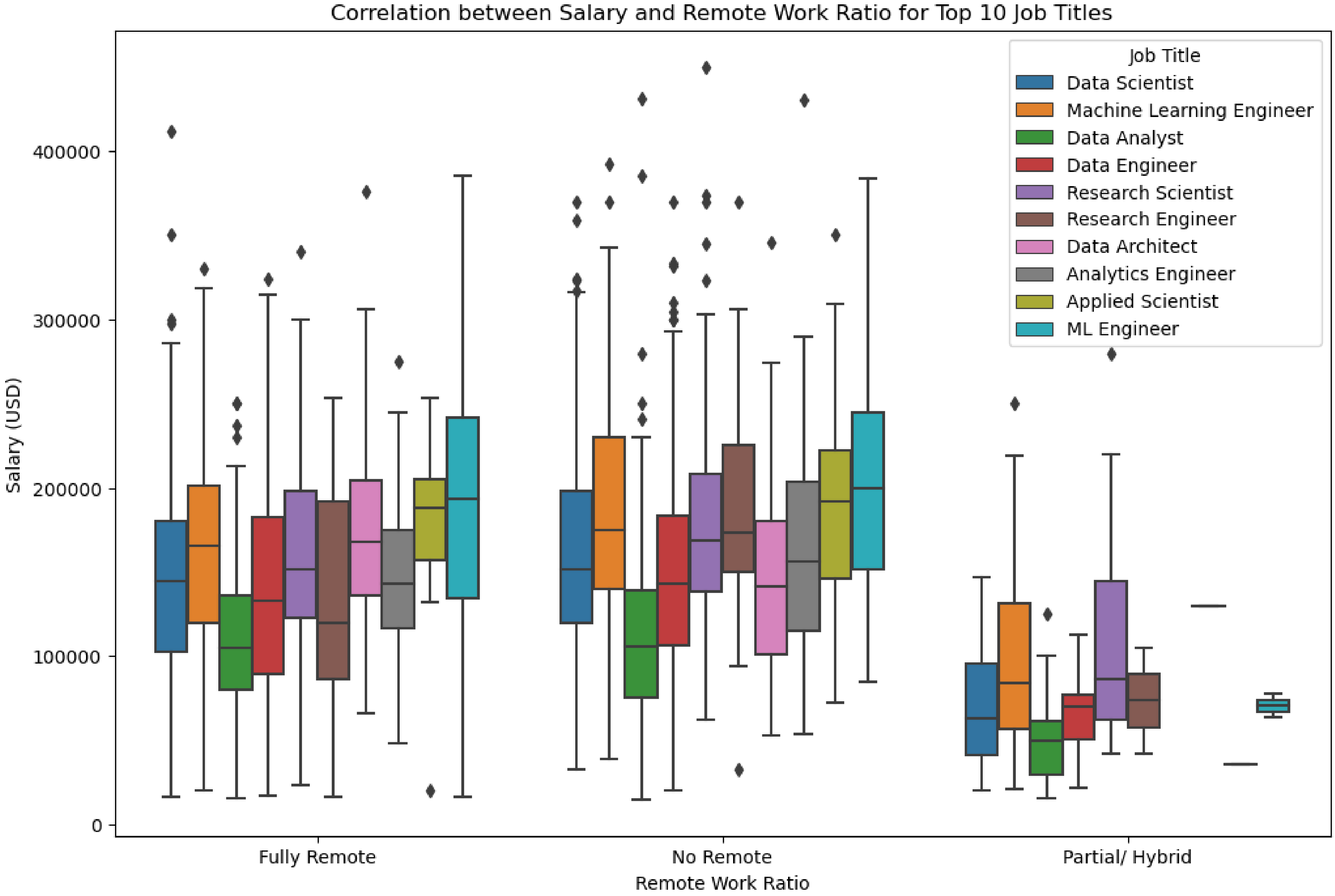


One notable trend observed is the evolution of salaries over the years among the top 10 job roles. For instance, the role of Machine Learning Engineer experienced a significant shift from being the lowest-paying role in 2020 to the highest-paying role in 2023. Conversely, the role of Applied Scientist witnessed a decline in salary ranking over the same period. This highlights dynamic changes in salary structures within the AI/ML and Big Data industries, reflecting shifts in demand, skill requirements, and market conditions over time.

Finding: What is the correlation between salary and the percentage of remote work allowed for different job titles or roles?

In [87]:

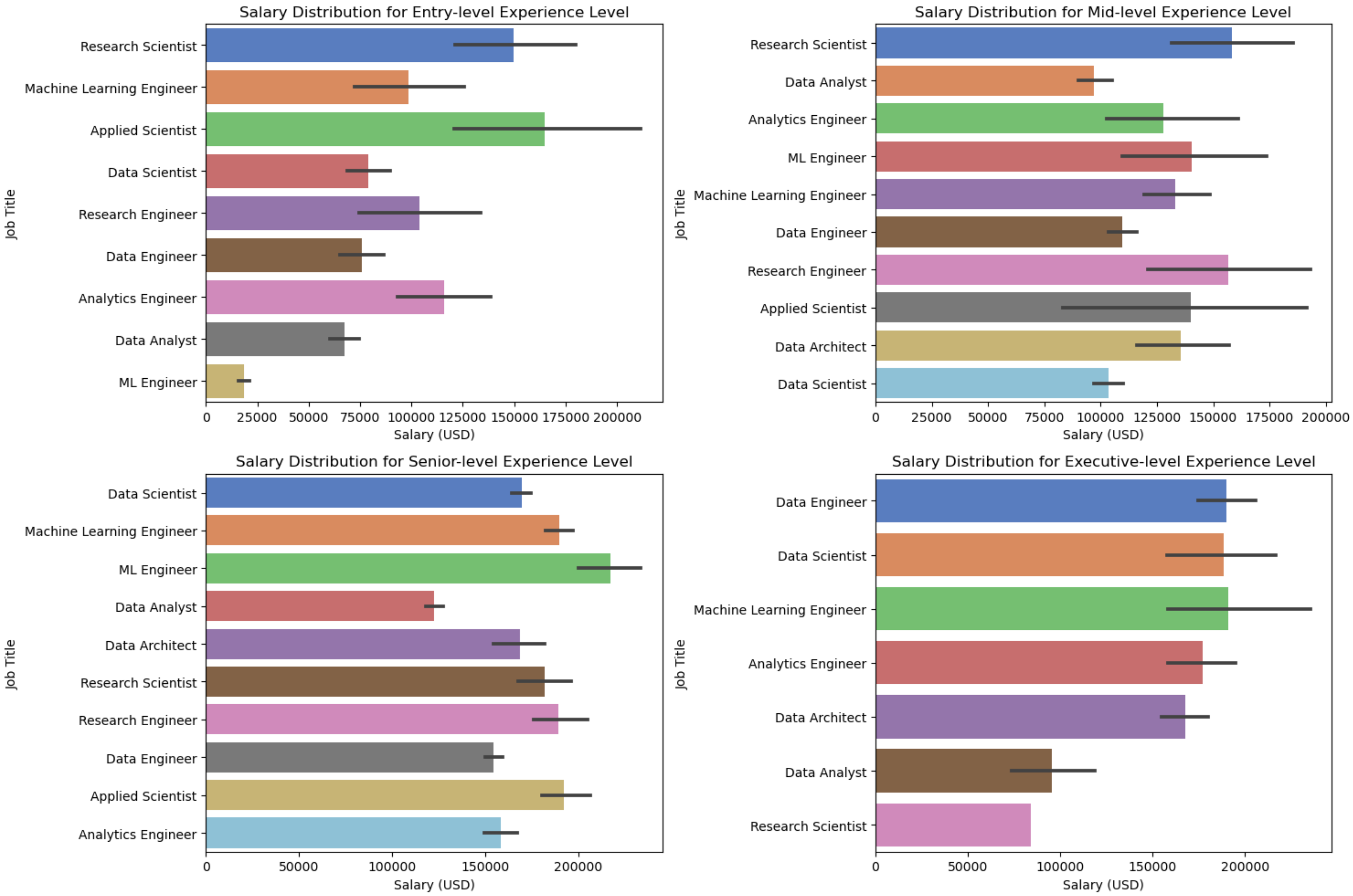
```
# Filter DataFrame to include only the top 10 job titles
df_filtered = df[df['job_title'].isin(top_10_job_titles)]

# Visualize using box plot
plt.figure(figsize=(12, 8))
sns.boxplot(data=df_filtered, x='remote_ratio', y='salary_in_usd', hue='job_title')
plt.title('Correlation between Salary and Remote Work Ratio for Top 10 Job Titles')
plt.xlabel('Remote Work Ratio')
plt.ylabel('Salary (USD)')
plt.legend(title='Job Title');
```



What is the trend in salaries among the top 10 Job roles acrosss the different experience levels

In [88]:
```
# Get unique experience levels
experience_levels = df_filtered['experience_level'].unique()

# Set up subplots
num_plots = len(experience_levels)
num_cols = 2  # Number of columns for subplots
```

```
num_rows = (num_plots + num_cols - 1) // num_cols  # Calculate number of rows

# Create subplots
fig, axes = plt.subplots(num_rows, num_cols, figsize=(15, 5*num_rows))

# Flatten the axes array for easier indexing
axes = axes.flatten()

# Loop through each unique experience level
for i, exp_level in enumerate(experience_levels):
    # Filter DataFrame for the current experience level
    df_exp_level = df_filtered[df_filtered['experience_level'] == exp_level]

    # Sort the DataFrame by salary_in_usd within each experience level
    df_exp_level_sorted = df_exp_level.sort_values(by='salary_in_usd', ascending=False)

    # Plot the group bar chart on the current subplot
    sns.barplot(data=df_exp_level_sorted, y='job_title', x='salary_in_usd', palette='muted', ax=axes[i])

    axes[i].set_title(f'Salary Distribution for {exp_level} Experience Level')
    axes[i].set_ylabel('Job Title')
    axes[i].set_xlabel('Salary (USD)')


# Adjust layout
plt.tight_layout()
plt.show()
```



SUMMARY

In summary, it can be deduced from the dataset that there is a notable growth trend in employee salaries within the AI/ML and Big Data industries over the years, indicating the increasing demand for skilled professionals and the evolving nature of the field.

KEY INSIGHTS:

1. The salary distribution of the employees in the AI/ML and Big Data industry as captored in the data is a normal/ right skewed distribution. The minimum salary is 15k and the maximum is 318k. On average, employees earn around 139k. There is presence of outliers with some employees earning as high as 450k. There are factors found to have influenced how much employees earn such as experience level, employment types, company sizes and remote work options.

2. Geographic Salary Disparities. The employees captured in the data are resident in 85 countries and companies located in 112 countries. the analysis revealed that both emplyees's residence and company location inluences their average salary. For better insight, top 10 countries were analysed.Employees who resides in USA and employees who work with companies that are located in the USA earn the biggest salary

3. Remote Work Trends: The distribution of the remote_ratio in the data revealed that about half of the jobs are not remote(i.e onsite), while 44% are fully remote and just about 5% is hybrid.There is an obverved trend that salaries varies depending on the remote work option. On average, the employees that work onsite(i.e No Remote) earns the most while the employees working on Hybrid option(i.e partial remote) earns the least salary. The fully remote workers are in between.

4. Company Size and Salary: The company sizes are in three categories namely Small, Medium and Large. Upon analysis, it was revealed that company size did affected the average salary but not in the conventional order. Medium sized company paid the highest salary followed by large sized company. Employees that work on site (no remote) earns more accross the three company sizes

5. Temporal Analysis: There is a clear indication that the salary evolved in an a significant upward trend over the years. This can indicates that the Artificial Intelligence and Big data is an evolving field. There was a slight decline in average salaries in 2021. This may attributed to the effect of Covid-19 lockdown.The subsequent years witnessed salary growth. Notably, remote workers enjoyed higher salaries between 2020 and 2022, highlighting the impact of remote work. However, in 2023, on-site (No Remote) workers surpassed both fully-remote and partially-remote workers in earning higher salaries. Over the years, salaries have evolved accross different experience levels and employment type.

6. Comparative Analysis: There is a clear indication that salaries differs accross the different job titles according to the employees experience level. As an entry level employee, the highest paid role is Applied Scientist while Machine Learning Engineer is the least paid role. In the Mid-level experience category,Reseach Scientist is the highest paid role while data analyst is the least paid role. In the Senior-level category, the highest paying role is Machine Learning engineer while Data Analyst is the least paid role and,

finally, the analytics Engineer, Data engineer, Machine Learning Engineer and Data scientist are in the same range and highest paid role while Research scientist becomes the least paid role.

RECOMMENDATION

Based on the key insights derived from the analysis, the following recommendations can be made:

1. Address Salary Disparities: Companies should review their salary structures to ensure fairness and equity across different factors such as experience level, employment types, and remote work options. Addressing salary disparities will help improve employee satisfaction and retention while fostering a more inclusive work environment.

2. Strategic Geographic Expansion: Given the observed geographic salary disparities, companies may consider expanding their operations or hiring talent in regions where salaries align with budgetary constraints and talent expectations. Additionally, offering competitive compensation packages in high-demand locations such as the USA can help attract and retain top talent.

3. Flexible Work Policies: Implementing flexible work policies that accommodate different remote work options can help enhance employee satisfaction and productivity. Companies should provide resources and support for effective remote collaboration while ensuring fair compensation for onsite, hybrid, and fully remote workers.

4. Company Size Considerations: Despite conventional wisdom, medium-sized companies were found to pay the highest salaries, followed by large-sized companies. This highlights the need for companies of all sizes to review and adjust their salary structures to remain competitive in the talent market.

5. Continuous Salary Reviews: In light of the upward salary trend over the years, companies should conduct regular salary reviews and adjustments to remain competitive and attract top talent. Monitoring industry salary benchmarks and economic indicators will help ensure that salary levels remain aligned with market trends.

6. Investment in Employee Development: To address salary differences across job titles and experience levels, companies should invest in employee development programs. Providing opportunities for continuous learning, upskilling, and career advancement will not only enhance employee skills but also contribute to improved job satisfaction and retention.

7. Promote Diversity and Inclusion: Recognizing the influence of demographic factors on salary disparities, companies should prioritize diversity and inclusion initiatives. By fostering diverse talent pipelines and addressing unconscious biases, organizations

can create opportunities for underrepresented groups and promote a more equitable salary landscape.

In Conclusion, by implementing these recommendations, companies can optimize their salary structures, foster a more inclusive work environment, and attract and retain top talent in the competitive AI/ML and Big Data industry.

Data source https://www.kaggle.com/datasets/cedricaubin/ai-ml-salaries/data