In [24]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [25]:

```
df = pd.read_pickle('df_cleaned')
```

In [26]:

```
df
```

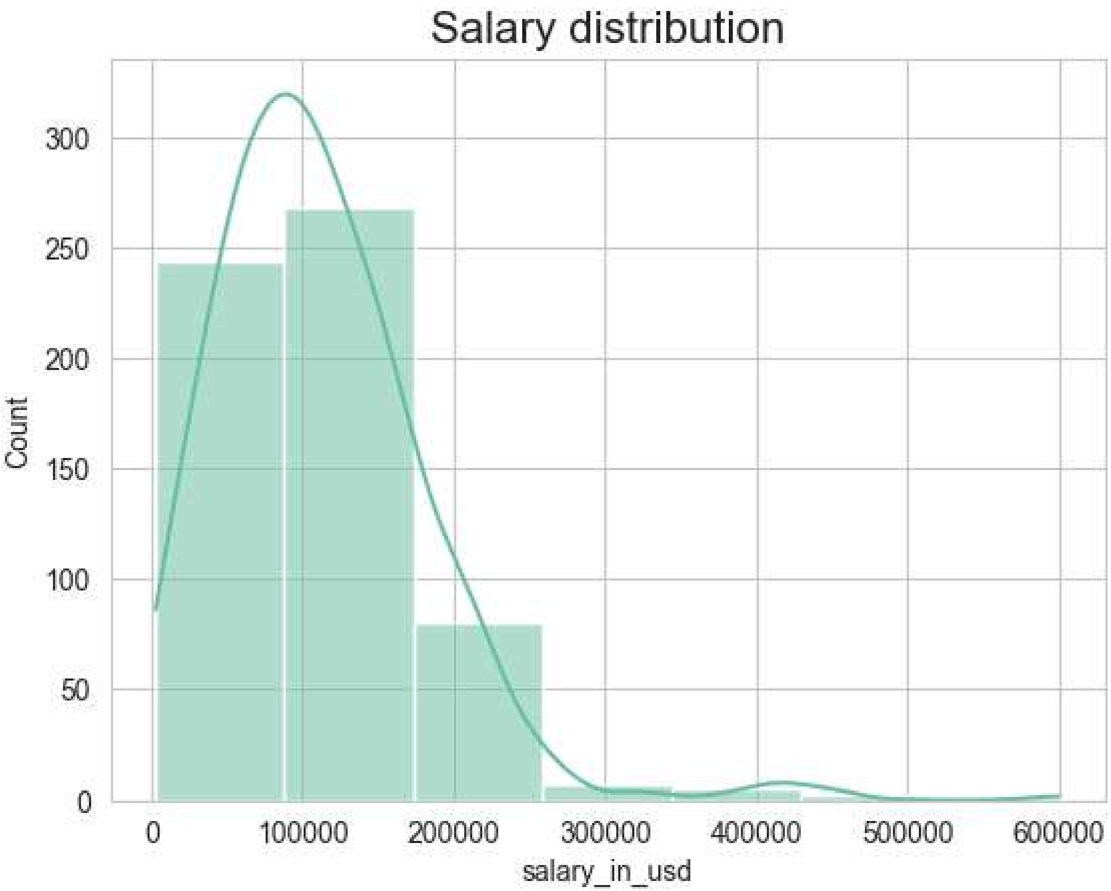
Out[26]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary_in_usd	en
0	0	2020	Mid	Full-time	data scientist	79833	
1	1	2020	Senior	Full-time	machine learning scientist	260000	
2	2	2020	Senior	Full-time	big data engineer	109024	
3	3	2020	Mid	Full-time	product data analyst	20000	
4	4	2020	Senior	Full-time	machine learning engineer	150000	
...	...	...	...	...	...	...	
602	602	2022	Senior	Full-time	data engineer	154000	
603	603	2022	Senior	Full-time	data engineer	126000	
604	604	2022	Senior	Full-time	data analyst	129000	
605	605	2022	Senior	Full-time	data analyst	150000	
606	606	2022	Mid	Full-time	ai scientist	200000	

607 rows × 10 columns

In [27]:

```
sns.histplot(df['salary_in_usd'],kde=True, bins=7)
plt.title('Salary distribution', fontdict={'fontsize': 16})
plt.show()
```



Mean salary by experience_level

In [28]:

```
mean_salary_by_experince = df.groupby('experience_level')['salary_in_usd'].mean().s
```

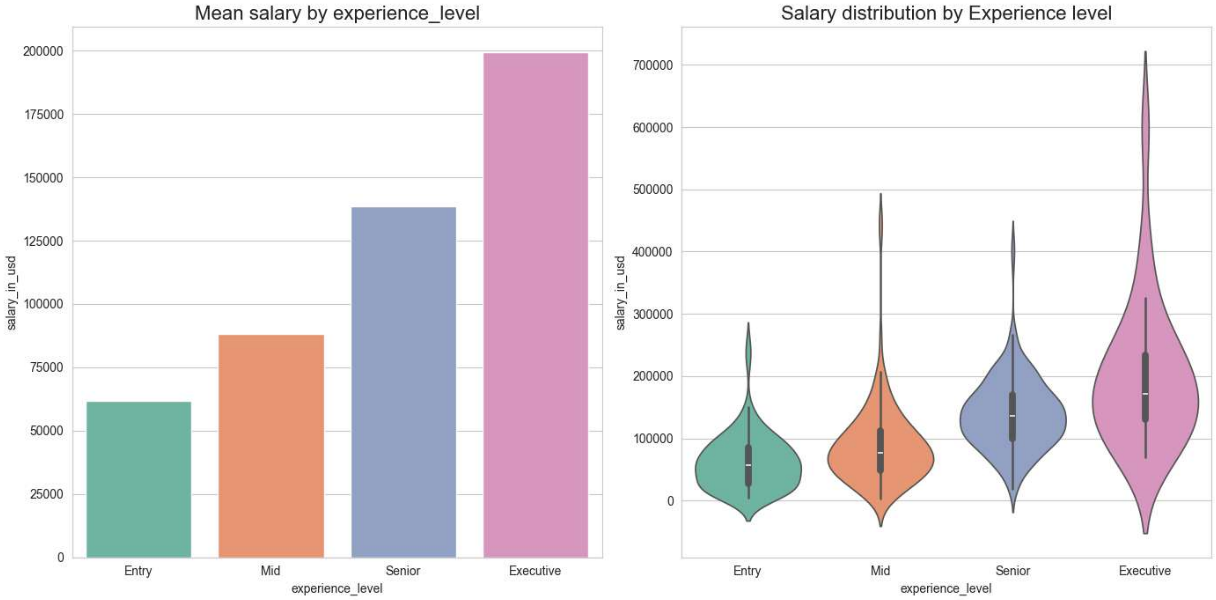
In [29]:

```
order_list = mean_salary_by_experince['experience_level'].tolist()
```

In [30]:

```
sns.set_style('whitegrid')
plt.figure(figsize=(14,7))
plt.subplot(1,2,1)
ax = sns.barplot(data=mean_salary_by_experince, x='experience_level', y = 'salary_i
ax.set_title('Mean salary by experience_level', fontdict={'fontsize':16})
plt.subplot(1,2,2)
ax1 = sns.violinplot(data=df, x='experience_level', y='salary_in_usd', palette='Set
ax1.set_title('Salary distribution by Experience level', fontdict={'fontsize':16})
plt.tight_layout()
plt.show()
```

```
C:\Users\pc\AppData\Local\Temp\ipykernel_20688\276185928.py:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.barplot(data=mean_salary_by_experince, x='experience_level', y = 'salary_
in_usd', palette='Set2')
C:\Users\pc\AppData\Local\Temp\ipykernel_20688\276185928.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax1 = sns.violinplot(data=df, x='experience_level', y='salary_in_usd', palette='Se
t2', order=order_list)
```



In [31]:
```
df
```

Out[31]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary_in_usd	en
0	0	2020	Mid	Full-time	data scientist	79833	
1	1	2020	Senior	Full-time	machine learning scientist	260000	
2	2	2020	Senior	Full-time	big data engineer	109024	
3	3	2020	Mid	Full-time	product data analyst	20000	
4	4	2020	Senior	Full-time	machine learning engineer	150000	
...	...	...	...	...	...	...	
602	602	2022	Senior	Full-time	data engineer	154000	
603	603	2022	Senior	Full-time	data engineer	126000	
604	604	2022	Senior	Full-time	data analyst	129000	
605	605	2022	Senior	Full-time	data analyst	150000	
606	606	2022	Mid	Full-time	ai scientist	200000	

607 rows × 10 columns

Mean salary by employment type

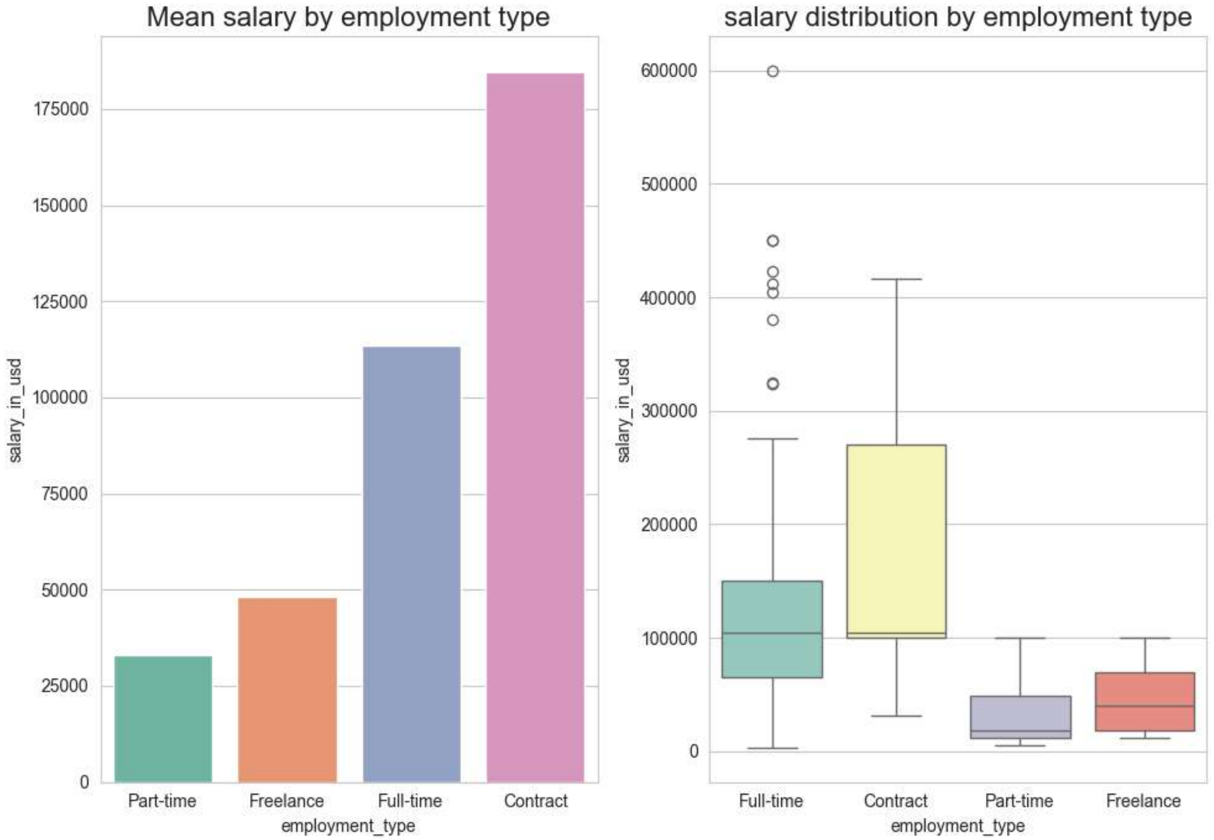
In [32]:

```
mean_salary_by_emp_type = df.groupby('employment_type')['salary_in_usd'].mean().sor
```

In [33]:

```
plt.figure(figsize=(10,7))
plt.subplot(1,2,1)
ax = sns.barplot(data=mean_salary_by_emp_type, x='employment_type', y='salary_in_us
ax.set_title("Mean salary by employment type", fontdict={'fontsize':16})

plt.subplot(1,2,2)
ax1 = sns.boxplot(data=df, x='employment_type', y='salary_in_usd', hue='employment_
ax1.set_title("salary distribution by employment type", fontdict={'fontsize':16})
plt.tight_layout()
```

```
C:\Users\pc\AppData\Local\Temp\ipykernel_20688\3558404321.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.barplot(data=mean_salary_by_emp_type, x='employment_type', y='salary_in_u
sd', palette='Set2')
```

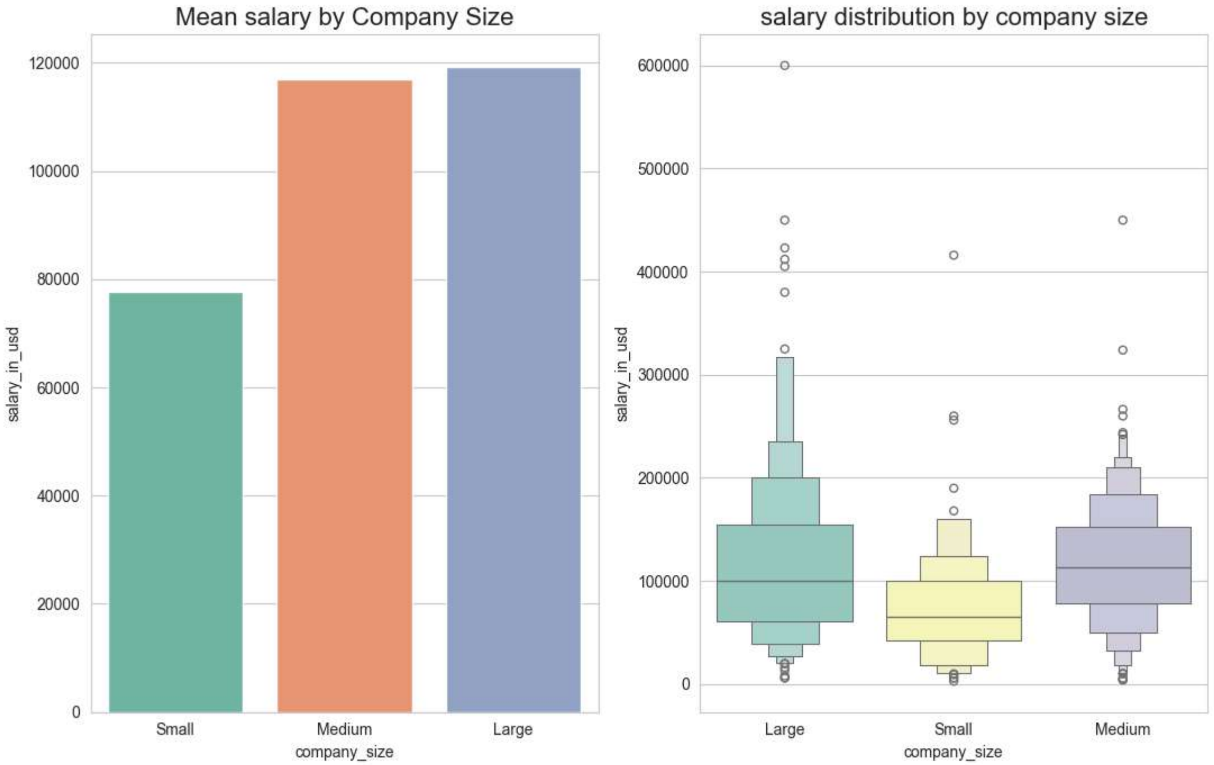


salary distribution by company size

In [34]:
```
mean_salary_by_company_size = df.groupby('company_size')['salary_in_usd'].mean().so
```

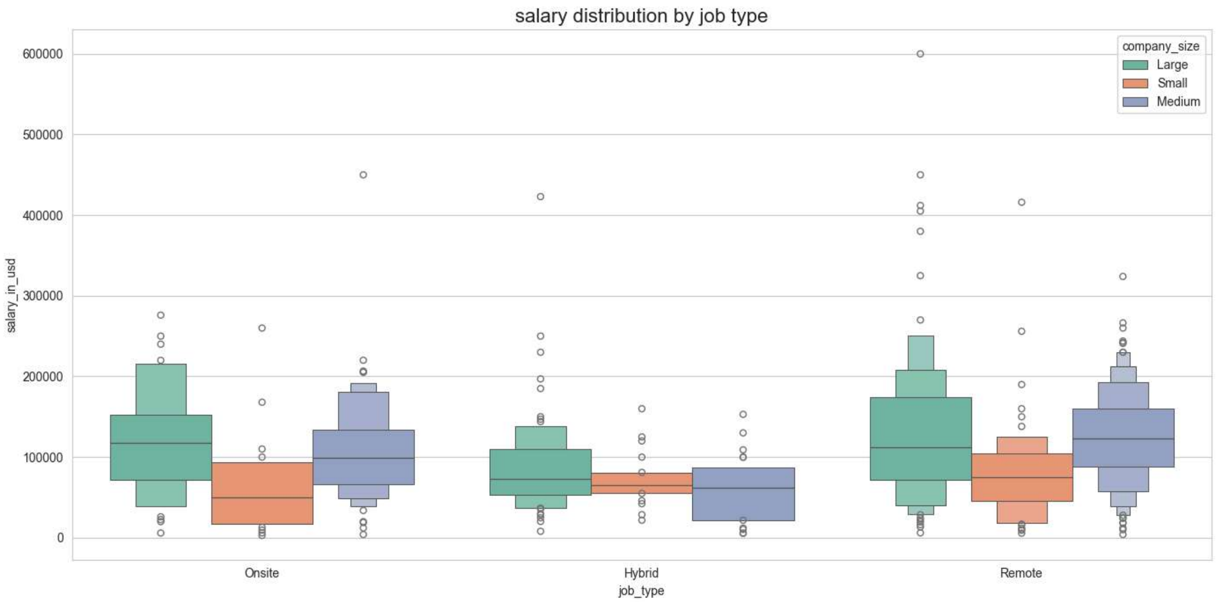
In [35]:
```
plt.figure(figsize=(11,7))
plt.subplot(1,2,1)
ax = sns.barplot(data=mean_salary_by_company_size, x='company_size', y='salary_in_u
ax.set_title("Mean salary by Company Size", fontdict={'fontsize':16})

plt.subplot(1,2,2)
ax1 = sns.boxenplot(data=df, x='company_size', y='salary_in_usd', hue='company_size
ax1.set_title("salary distribution by company size", fontdict={'fontsize':16})
plt.tight_layout()
```

```
C:\Users\pc\AppData\Local\Temp\ipykernel_20688\1565677435.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.1
4.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.barplot(data=mean_salary_by_company_size, x='company_size', y='salary_in_
usd', palette='Set2')
```



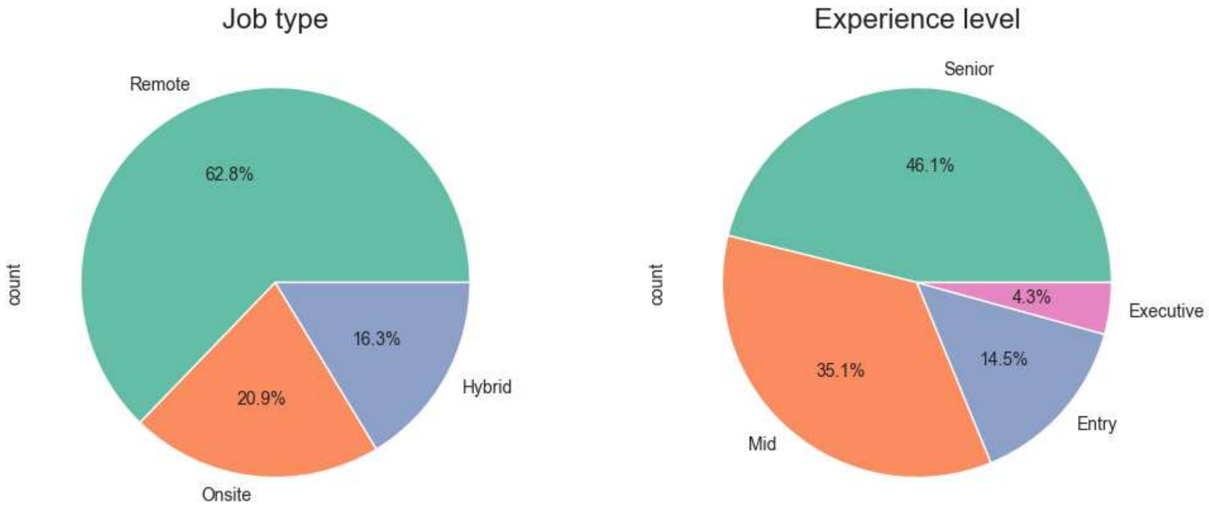
salary distribution by job type

In [36]:
```
plt.figure(figsize=(14,7))

ax1 = sns.boxenplot(data=df, x='job_type', y='salary_in_usd', hue='company_size', p
ax1.set_title("salary distribution by job type", fontdict={'fontsize':16})
plt.tight_layout()
```



Job type count

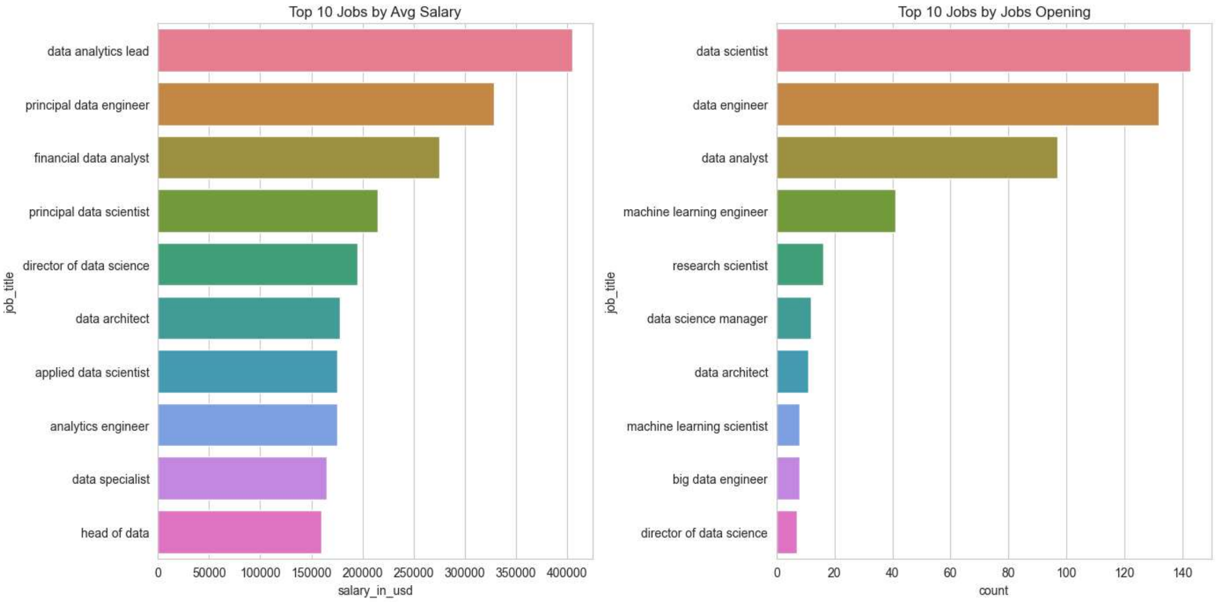
```
In [37]: plt.figure(figsize=(12,5))
         sns.set_palette('Set2')
         plt.subplot(1,2,1)
         ax = df['job_type'].value_counts().plot(kind='pie', autopct='%1.1f%%')
         ax.set_title('Job type', fontdict={'fontsize':16} )

         plt.subplot(1,2,2)
         ax1 = df['experience_level'].value_counts().plot(kind='pie', autopct='%1.1f%%')
         ax1.set_title('Experience level', fontdict={'fontsize':16} )
         plt.show()
```



Top 10 Jobs by Avg Salary

In [38]:
```
top_10_jobs_salary = df.groupby('job_title')['salary_in_usd'].mean().sort_values(as
top_10_jobs_openings = df['job_title'].value_counts().sort_values(ascending=False).
```

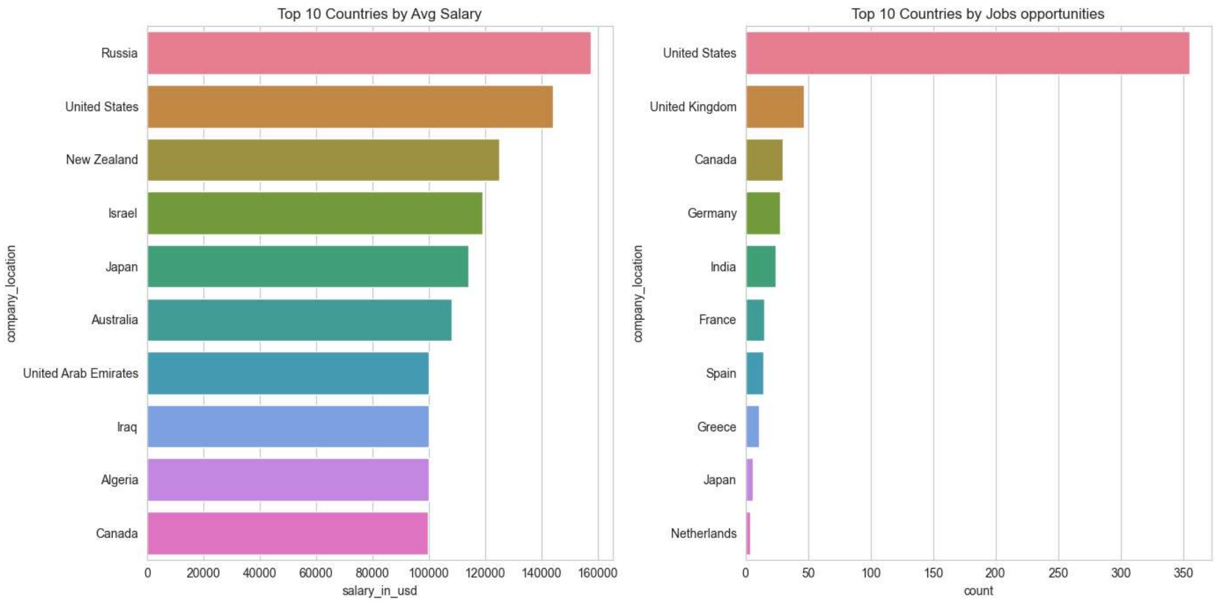
In [39]:
```
plt.figure(figsize=(14,7))
sns.set_palette('Set2')
plt.subplot(1,2,1)

ax = sns.barplot(data=top_10_jobs_salary, x='salary_in_usd', y='job_title', hue='jo
ax.set_title('Top 10 Jobs by Avg Salary')

plt.subplot(1,2,2)
ax1 = sns.barplot(data=top_10_jobs_openings, x='count', y='job_title', hue='job_tit
ax1.set_title('Top 10 Jobs by Jobs Opening')

plt.tight_layout()
```



Top 10 Countries by Avg Salary

In [40]:
```
top_10_countries_by_salary = df.groupby('company_location')['salary_in_usd'].mean()
top_10_countries_by_openings = df['company_location'].value_counts().sort_values(as
```

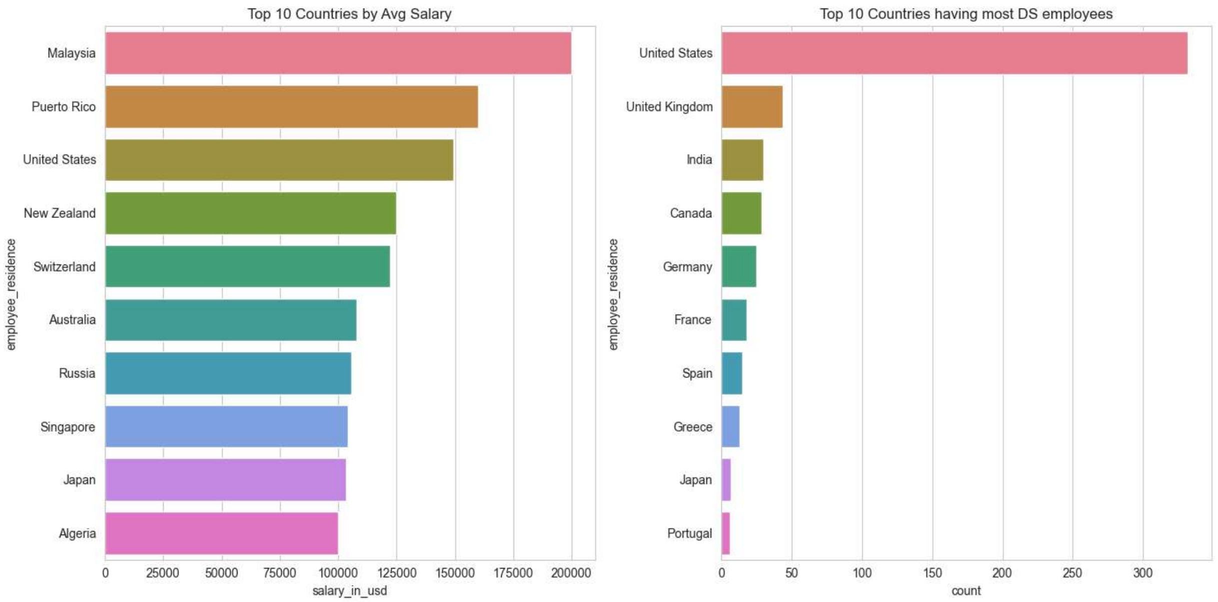
In [41]:
```
plt.figure(figsize=(14,7))
sns.set_palette('Set2')
plt.subplot(1,2,1)

ax = sns.barplot(data=top_10_countries_by_salary, x='salary_in_usd', y='company_loc
ax.set_title('Top 10 Countries by Avg Salary')

plt.subplot(1,2,2)
ax1 = sns.barplot(data=top_10_countries_by_openings, x='count', y='company_location
ax1.set_title('Top 10 Countries by Jobs opportunities')

plt.tight_layout()
```



Top 10 residence Countries by Avg Salary

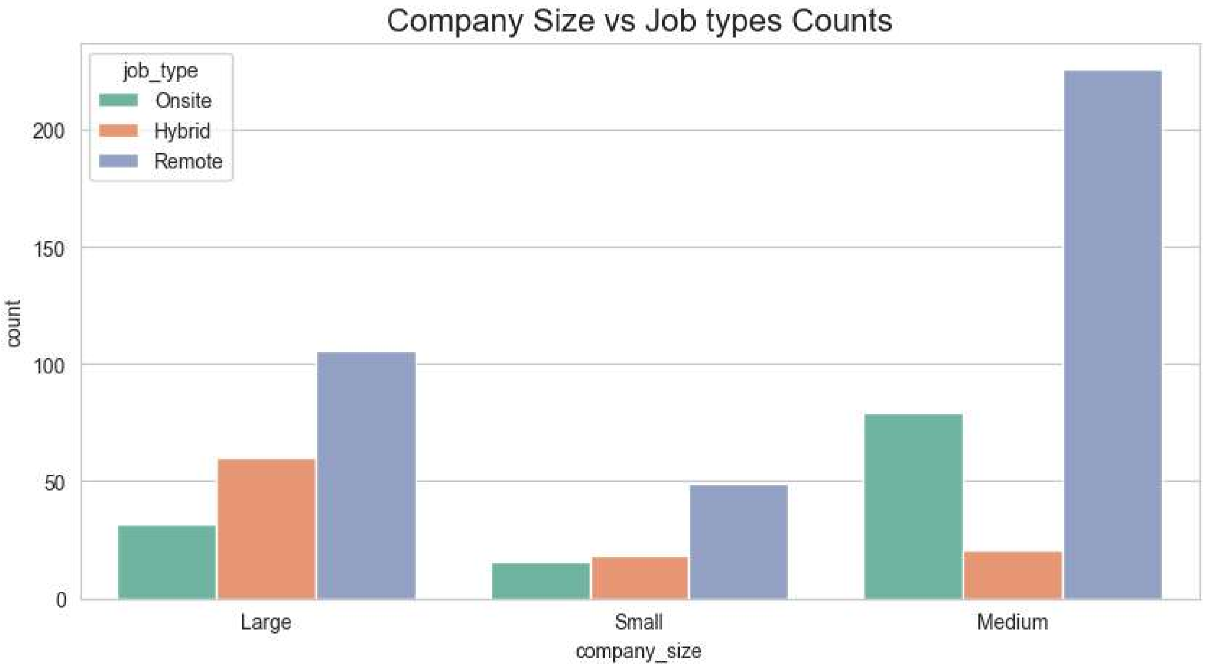
In [42]:
```
top_10_res_countries_by_salary = df.groupby('employee_residence')['salary_in_usd'].
top_10_res_countries_by_openings = df['employee_residence'].value_counts().sort_val
```

In [43]:
```
plt.figure(figsize=(14,7))
sns.set_palette('Set2')
plt.subplot(1,2,1)

ax = sns.barplot(data=top_10_res_countries_by_salary, x='salary_in_usd', y='employe
ax.set_title('Top 10 Countries by Avg Salary')

plt.subplot(1,2,2)
ax1 = sns.barplot(data=top_10_res_countries_by_openings, x='count', y='employee_res
ax1.set_title('Top 10 Countries having most DS employees')

plt.tight_layout()
```



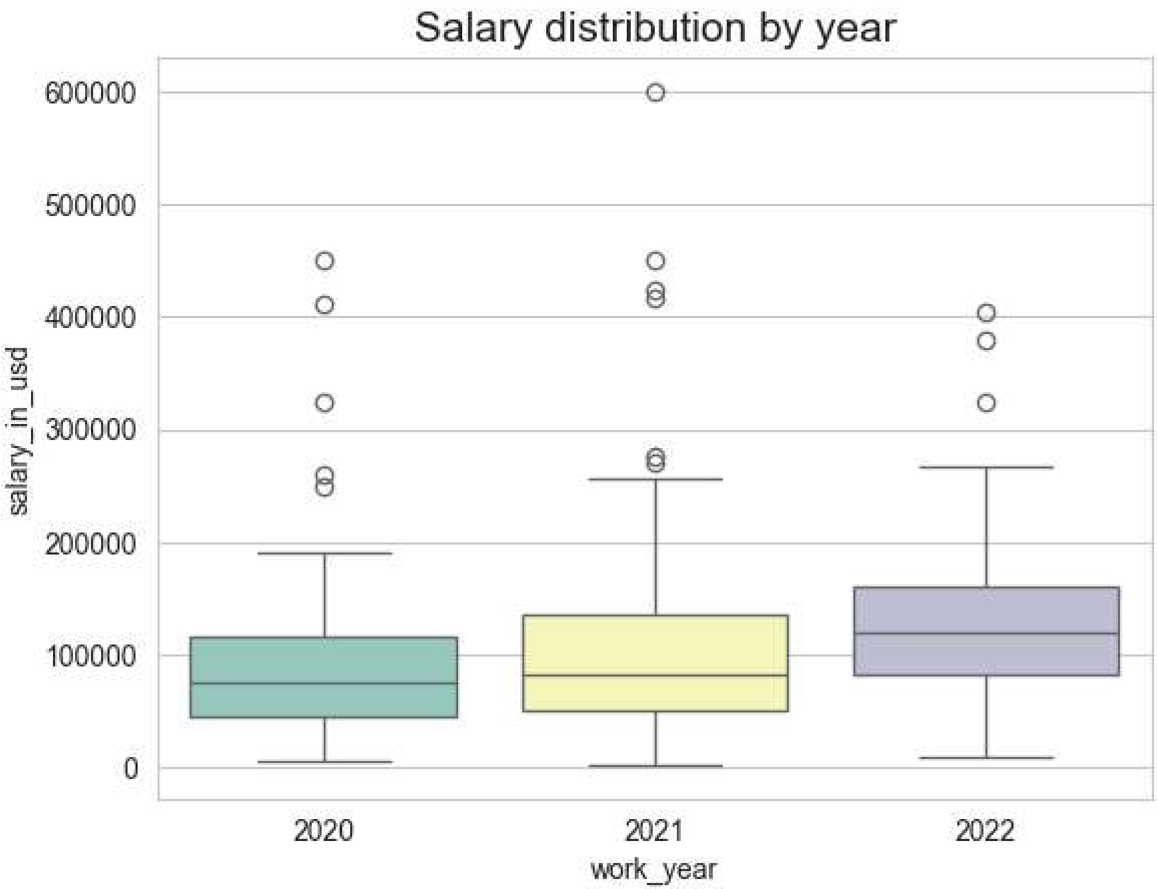
Company Size vs Job types Counts

In [44]:
```
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='company_size', hue='job_type')
plt.title('Company Size vs Job types Counts', fontdict={'fontsize':16})
plt.show()
```



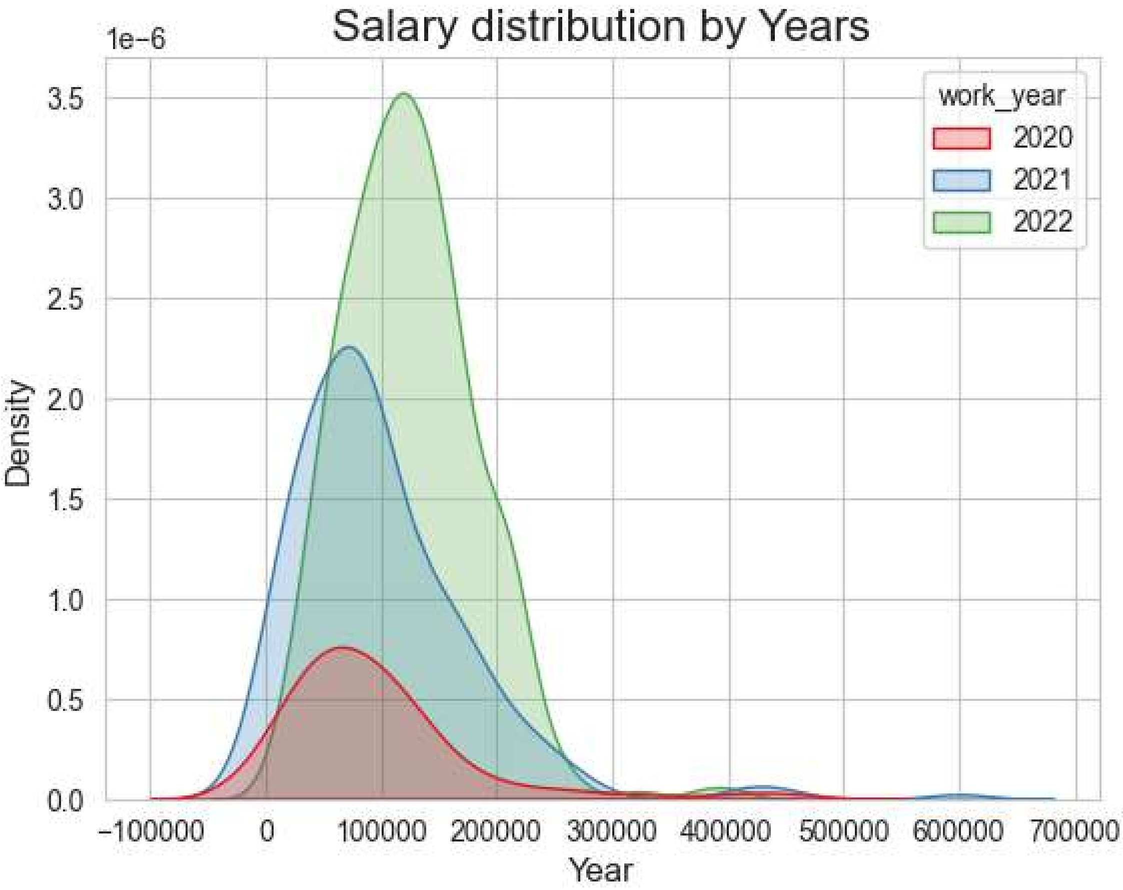
Salary ditribution by year

In [45]:
```
sns.boxplot(data=df, x='work_year', y='salary_in_usd',hue='work_year', palette='Set
plt.title('Salary distribution by year', fontdict={"fontsize":15})
plt.show()
```



```
In [46]: sns.set_style('whitegrid')
         sns.kdeplot(data=df, x='salary_in_usd', hue='work_year', fill=True, palette='Set1')
         plt.title('Salary distribution by Years', fontsize=16)
         plt.xlabel('Year',fontsize=12)
         plt.ylabel('Density',fontsize=12)
```

```
Out[46]: Text(0, 0.5, 'Density')
```


Salary distribution by Years
1e−6
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Density
−100000
0
100000
200000
300000
400000
500000
600000
700000
Year
work_year
🟥 2020
🟦 2021
🟩 2022

In [ ]: