In [ ]:
```
#Daisy Rivera, 04/20/2025

Project 1
```

Your CEO has decided that the company needs a full-time data scientist, and possibly a team of them in the future. She thinks she needs someone who can help drive data science within then entire organization and could potentially lead a team in the future. She understands that data scientist salaries vary widely across the world and is unsure what to pay them. To complicate matters, salaries are going up due to the great recession and the market is highly competitive. Your CEO has asked you to prepare an analysis on data science salaries and provide them with a range to be competitive and get top talent. The position can work offshore, but the CEO would like to know what the difference is for a person working in the United States. Your company is currently a small company but is expanding rapidly.

Prepare your analysis in an R file. Your final product should be a power point presentation giving your recommendation to the CEO. CEOs do not care about your code and don't want to see it. They want to see visuals and a well thought out analysis. You will need to turn in the power point and the code as a flat R file.

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import os
```

In [11]:
```
infile = "data/daisy_rivera.module05RProject.csv"
ds_salaries_df=pd.read_csv(infile)

ds_salaries_df
```

Out[11]:

	Unnamed: 0	work_year	experience_level	employment_type	job_title	salary	salary_cu
0	0	2020	MI	FT	Data Scientist	70000	
1	1	2020	SE	FT	Machine Learning Scientist	260000	
2	2	2020	SE	FT	Big Data Engineer	85000	
3	3	2020	MI	FT	Product Data Analyst	20000	
4	4	2020	SE	FT	Machine Learning Engineer	150000	
...	...	...	...	...	...	...	
602	602	2022	SE	FT	Data Engineer	154000	
603	603	2022	SE	FT	Data Engineer	126000	
604	604	2022	SE	FT	Data Analyst	129000	
605	605	2022	SE	FT	Data Analyst	150000	
606	606	2022	MI	FT	AI Scientist	200000	

607 rows × 12 columns

In [12]:
```
list(ds_salaries_df.columns)
```

Out[12]:
```
['Unnamed: 0',
 'work_year',
 'experience_level',
 'employment_type',
 'job_title',
 'salary',
 'salary_currency',
 'salary_in_usd',
 'employee_residence',
 'remote_ratio',
 'company_location',
 'company_size']
```

In [14]:
```
data_scientists = ds_salaries_df[ds_salaries_df['job_title'] == 'Data Scientist']

avg_salary_by_location = data_scientists.groupby('company_location')['salary_in_usd

avg_salary_by_location.sort_values(by='salary_in_usd', ascending=False).head(10)
```

Out[14]:

	company_location	salary_in_usd
22	US	143115.68
4	CH	122346.00
12	IL	119059.00
7	DZ	100000.00
10	GB	88177.36
1	AU	86703.00
3	CA	77787.00
0	AT	76352.00
6	DE	69640.14
15	LU	62726.00

In [15]:
```
data_scientists = ds_salaries_df[ds_salaries_df['job_title'] == 'Data Scientist']

global_avg_salary = data_scientists['salary_in_usd'].mean()
global_min_salary = data_scientists['salary_in_usd'].min()
global_max_salary = data_scientists['salary_in_usd'].max()

print(f"Global Average Salary: ${round(global_avg_salary):,}")
print(f"Global Minimum Salary: ${round(global_min_salary):,}")
print(f"Global Maximum Salary: ${round(global_max_salary):,}")
```

```
Global Average Salary: $108,188
Global Minimum Salary: $2,859
Global Maximum Salary: $412,000
```

In [7]:
```
us_data_scientists = data_scientists[data_scientists['employee_residence'] == 'US']

us_avg_salary = us_data_scientists['salary_in_usd'].mean()
us_min_salary = us_data_scientists['salary_in_usd'].min()
us_max_salary = us_data_scientists['salary_in_usd'].max()

print(f"U.S. Average Salary: ${round(us_avg_salary):,}")
print(f"U.S. Minimum Salary: ${round(us_min_salary):,}")
print(f"U.S. Maximum Salary: ${round(us_max_salary):,}")
```

```
U.S. Average Salary: $149,408
U.S. Minimum Salary: $58,000
U.S. Maximum Salary: $412,000
```

In [61]:
```
percent_difference = ((us_avg_salary - global_avg_salary) / global_avg_salary) * 10

print(percent_difference)
```

```
38.10086618757243
```

In [19]:
```
data_scientists = ds_salaries_df[ds_salaries_df['job_title'] == 'Data Scientist']

data_scientists.loc[:, 'experience_level'] = data_scientists['experience_level'].re
    'EN': 'Entry-level',
    'MI': 'Mid-level',
    'SE': 'Senior',
    'EX': 'Executive'
})
```

In [20]:
```
exp_level = data_scientists.groupby('experience_level')['salary_in_usd'].agg(
    avg_salary='mean',
    min_salary='min',
    max_salary='max',
    median_salary='median'
).round(0).astype(int).reset_index()

print(exp_level)
```

```
  experience_level  avg_salary  min_salary  max_salary  median_salary
0      Entry-level       55331        4000      105000          50484
1        Mid-level       82039        2859      200000          77479
2           Senior      152971       20171      412000         140400
```

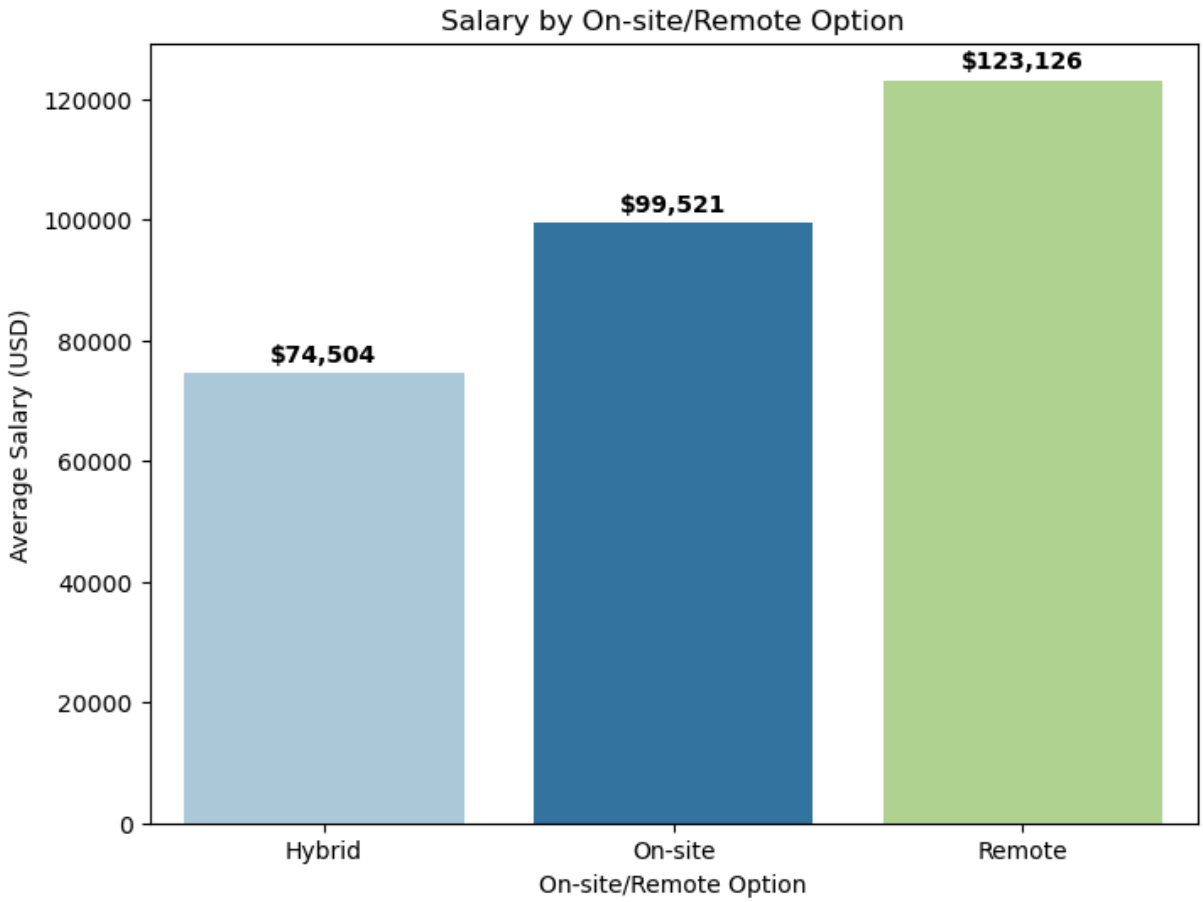
In [57]:
```
data_scientists = ds_salaries_df[ds_salaries_df['job_title'] == 'Data Scientist'].c

data_scientists['remote_status'] = data_scientists['remote_ratio'].map({
    0: 'On-site',
    50: 'Hybrid',
    100: 'Remote'
})

remote_salary = data_scientists.groupby('remote_status')['salary_in_usd'].mean().ro
print(remote_salary)
```

```
  remote_status  salary_in_usd
0        Hybrid          74504
1       On-site          99521
2        Remote         123126
```

In [40]:
```
remote_avg = data_scientists.groupby('remote_status')['salary_in_usd'].mean().round

plt.figure(figsize=(8, 6))
sns.barplot(data=remote_avg, x='remote_status', y='salary_in_usd', palette='Paired'

for index, row in remote_avg.iterrows():
    plt.text(index, row['salary_in_usd'] + 2000, f"${row['salary_in_usd']:,}",
             ha='center', fontweight='bold')

plt.title("Salary by On-site/Remote Option")
plt.ylabel("Average Salary (USD)")
```

```
plt.xlabel("On-site/Remote Option")
plt.show()
```

```
C:\Users\daisy\AppData\Local\Temp\ipykernel_14092\2394195481.py:6: 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.

  sns.barplot(data=remote_avg, x='remote_status', y='salary_in_usd', palette='Paire
d')
```



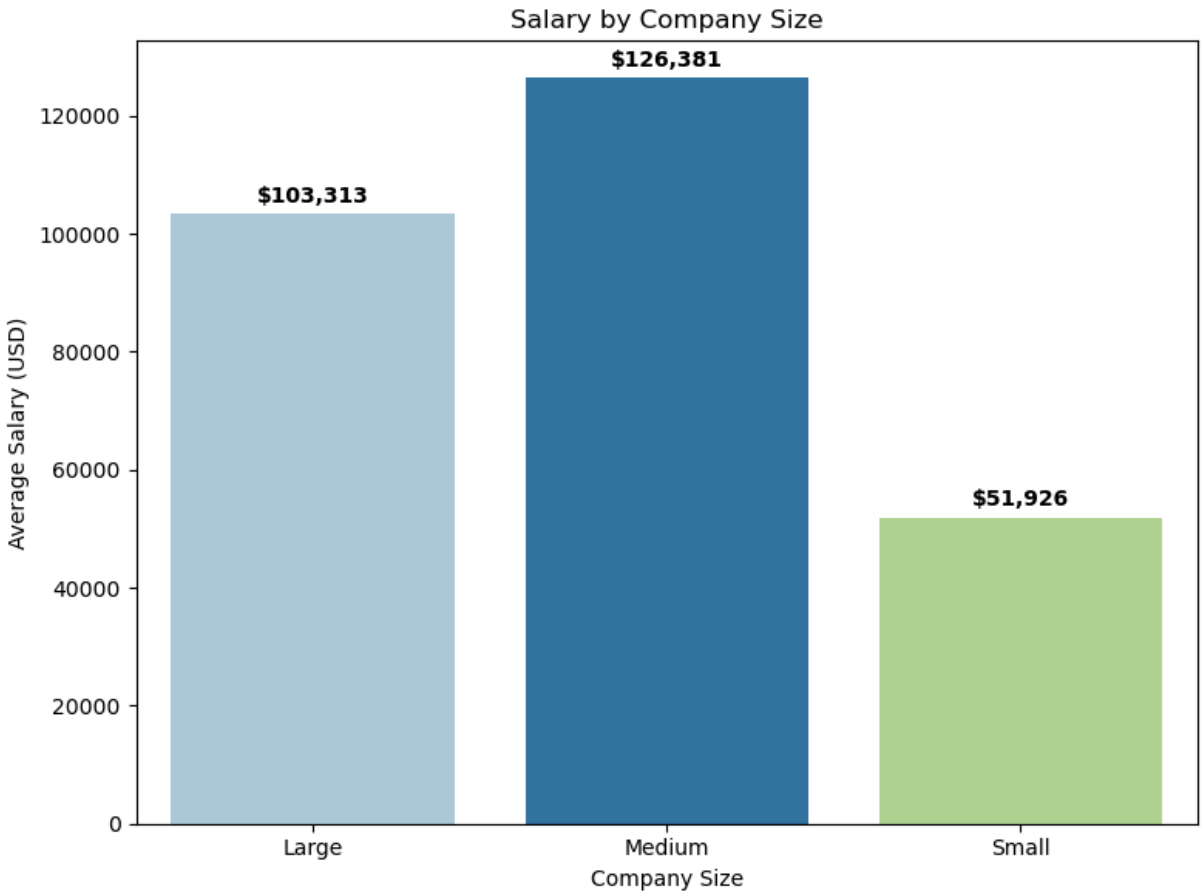
In [42]:
```
data_scientists.loc[:, 'company_size'] = data_scientists['company_size'].replace({
    'S': 'Small',
    'M': 'Medium',
    'L': 'Large'
})

company_size_salary = data_scientists.groupby('company_size')['salary_in_usd'].mean

print(company_size_salary)
```

```
  company_size  salary_in_usd
0        Large         103313
1       Medium         126381
2        Small          51926
```

In [55]:
```
plt.figure(figsize=(8, 6))
sns.barplot(
    data=company_size_salary,
```

```
    x='company_size',
    y='salary_in_usd',
    palette='Paired'
)


for index, row in company_size_salary.iterrows():
    plt.text(index, row['salary_in_usd'] + 2000, f"${row['salary_in_usd']:,}",
             ha='center', fontweight='bold')

plt.title("Salary by Company Size")
plt.xlabel("Company Size")
plt.ylabel("Average Salary (USD)")
plt.tight_layout()
plt.savefig("salary_by_company_size_paired_palette.png")
plt.show()
```

```
C:\Users\daisy\AppData\Local\Temp\ipykernel_14092\1240318588.py:2: 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.

  sns.barplot(
```



In [ ]: