

```
In [6]: import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
In [7]: df = pd.read_csv('Aidata.csv', sep=';')
```

```
In [8]: df.head()
```

Out[8]:

|   | Company_Name | Website                   | Location            | Minimum Project Size | Average Hourly Rate | Number of Employees | Percent AI Service Focus | Unnamed: 7 |
|---|--------------|---------------------------|---------------------|----------------------|---------------------|---------------------|--------------------------|------------|
| 0 | Smith.ai     | https://smith.ai/         | Los Altos, CA       | Undisclosed          | Undisclosed         | 250 - 999           | %١٠                      | NaN        |
| 1 | Moovelt      | https://moove-it.com/     | Austin, TX          | \$100,000+           | 50—99 / hr          | 50 - 249            | %١٥                      | NaN        |
| 2 | TechSpeedInc | http://www.techspeed.com/ | Portland, OR        | \$1,000+             | < \$25 / hr         | 50 - 249            | %٤٠                      | NaN        |
| 3 | WillDom      | https://willdom.com/      | Fort Lauderdale, FL | \$10,000+            | 50—99 / hr          | 250 - 999           | %١٠                      | NaN        |
| 4 | Icreon       | https://www.icreon.com/en | New York, NY        | \$50,000+            | 100—149 / hr        | 250 - 999           | %١٠                      | NaN        |

```
In [4]: df.dtypes
```

Out[4]: Company\_Name str  
Website str  
Location str  
Minimum Project Size str  
Average Hourly Rate str  
Number of Employees str  
Percent AI Service Focus str  
Unnamed: 7 str  
dtype: object

```
In [5]: df.columns
```

Out[5]: Index(['Company\_Name', 'Website', 'Location', 'Minimum Project Size',  
'Average Hourly Rate', 'Number of Employees',  
'Percent AI Service Focus', 'Unnamed: 7'],  
dtype='str')

```
In [9]: df.isnull().sum()
```

Out[9]: Company\_Name 0  
Website 1  
Location 13  
Minimum Project Size 0  
Average Hourly Rate 0  
Number of Employees 0  
Percent AI Service Focus 0  
Unnamed: 7 3099  
dtype: int64

```
In [10]: missing_percentage = df.isnull().mean() * 100
threshold = 30
columns_to_drop = missing_percentage[missing_percentage > threshold].index
df = df.drop(columns=columns_to_drop)
print(f"Columns dropped: {list(columns_to_drop)}")
print(f"Remaining columns: {df.columns}")
```

Columns dropped: ['Unnamed: 7']  
Remaining columns: Index(['Company\_Name', 'Website', 'Location', 'Minimum Project Size',  
'Average Hourly Rate', 'Number of Employees',  
'Percent AI Service Focus'],  
dtype='str')

```
In [11]: import pandas as pd
import numpy as np

df = pd.read_csv('Aidata.csv', sep=';')
```

```
def convert_to_numeric(val):
    if pd.isna(val):
        return np.nan
    arabic_map = str.maketrans('٠١٢٣٤٥٦٧٨٩', '0123456789')
    clean_val = str(val).translate(arabic_map).replace('%', '').strip()
    try:
        return float(clean_val)
    except:
        return np.nan

df['AI_Focus_Numeric'] = df['Percent AI Service Focus'].apply(convert_to_numeric)

Q1 = df['AI_Focus_Numeric'].quantile(0.25)
Q3 = df['AI_Focus_Numeric'].quantile(0.75)
IQR = Q3 - Q1

lower_limit = Q1 - 1.5 * IQR
upper_limit = Q3 + 1.5 * IQR

outliers = df[(df['AI_Focus_Numeric'] < lower_limit) | (df['AI_Focus_Numeric'] > upper_limit)]

print(f"--- Detection Results ---")
print(f"Lower Limit: {lower_limit}")
print(f"Upper Limit: {upper_limit}")
print(f"Total Outliers Found: {len(outliers)}")
print("\nOutlier Data Rows:")
print(outliers[['Company_Name', 'Percent AI Service Focus']])
```

--- Detection Results ---

Lower Limit: -50.0

Upper Limit: 110.0

Total Outliers Found: 4

Outlier Data Rows:

|    | Company_Name | Percent AI Service Focus |
|----|--------------|--------------------------|
| 7  | RevolveAI    | %٤٠٠                     |
| 10 | Sciant       | %٩٠٠                     |
| 13 | WorkingMouse | %٩٠٠                     |
| 15 | AESstudio    | %٥٠٠                     |

```
In [12]: df['AI_Focus_Cleaned'] = df['AI_Focus_Numeric'].clip(lower=lower_limit, upper=upper_limit)

new_outliers = df[(df['AI_Focus_Cleaned'] < lower_limit) | (df['AI_Focus_Cleaned'] > upper_limit)]

print(f"Number of outliers after handling: {len(new_outliers)}")

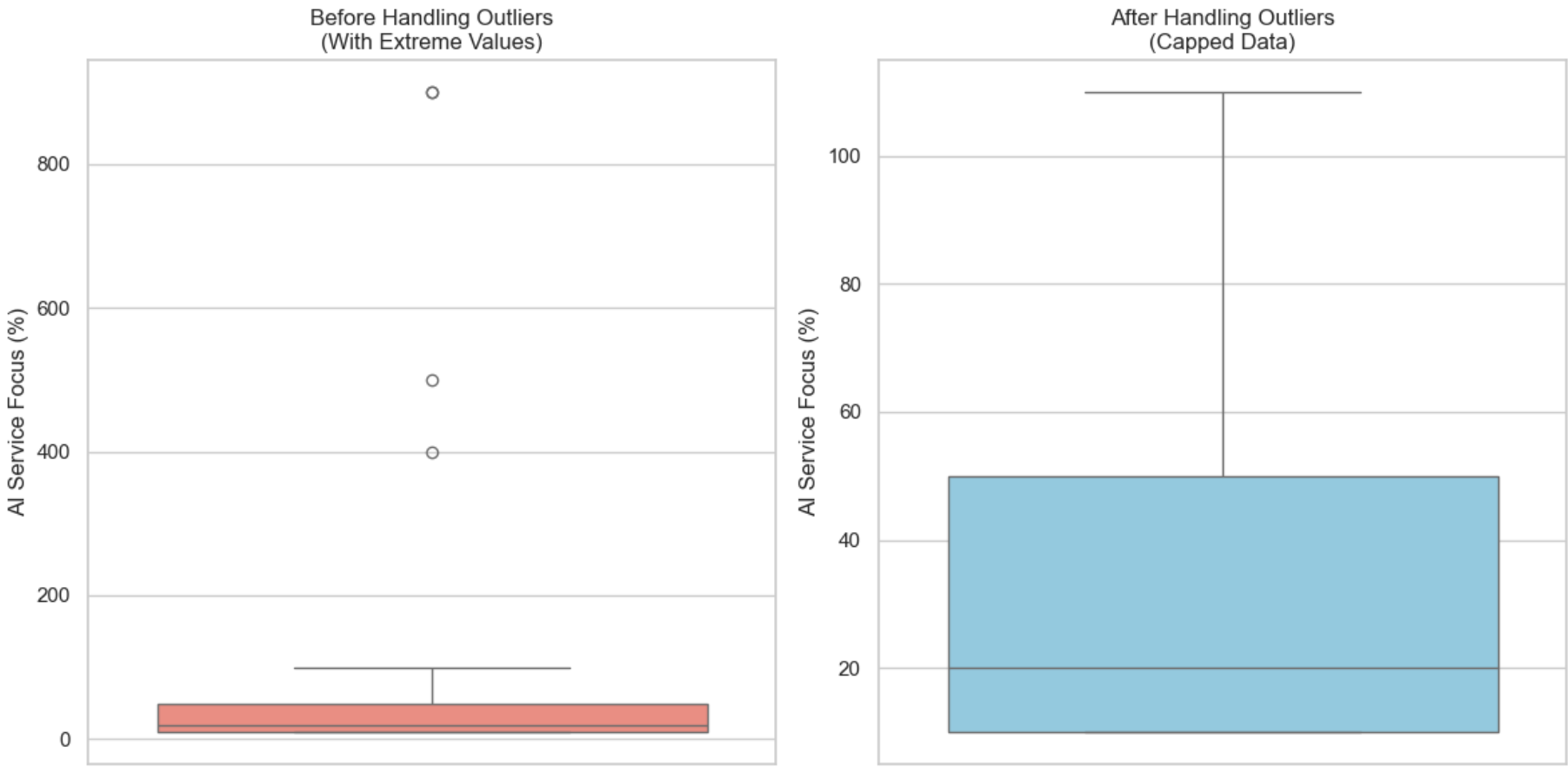
print("\nSample of treated data (Original vs Cleaned):")
print(df[['Percent AI Service Focus', 'AI_Focus_Numeric', 'AI_Focus_Cleaned']].iloc[outliers.index])
```

Number of outliers after handling: 0

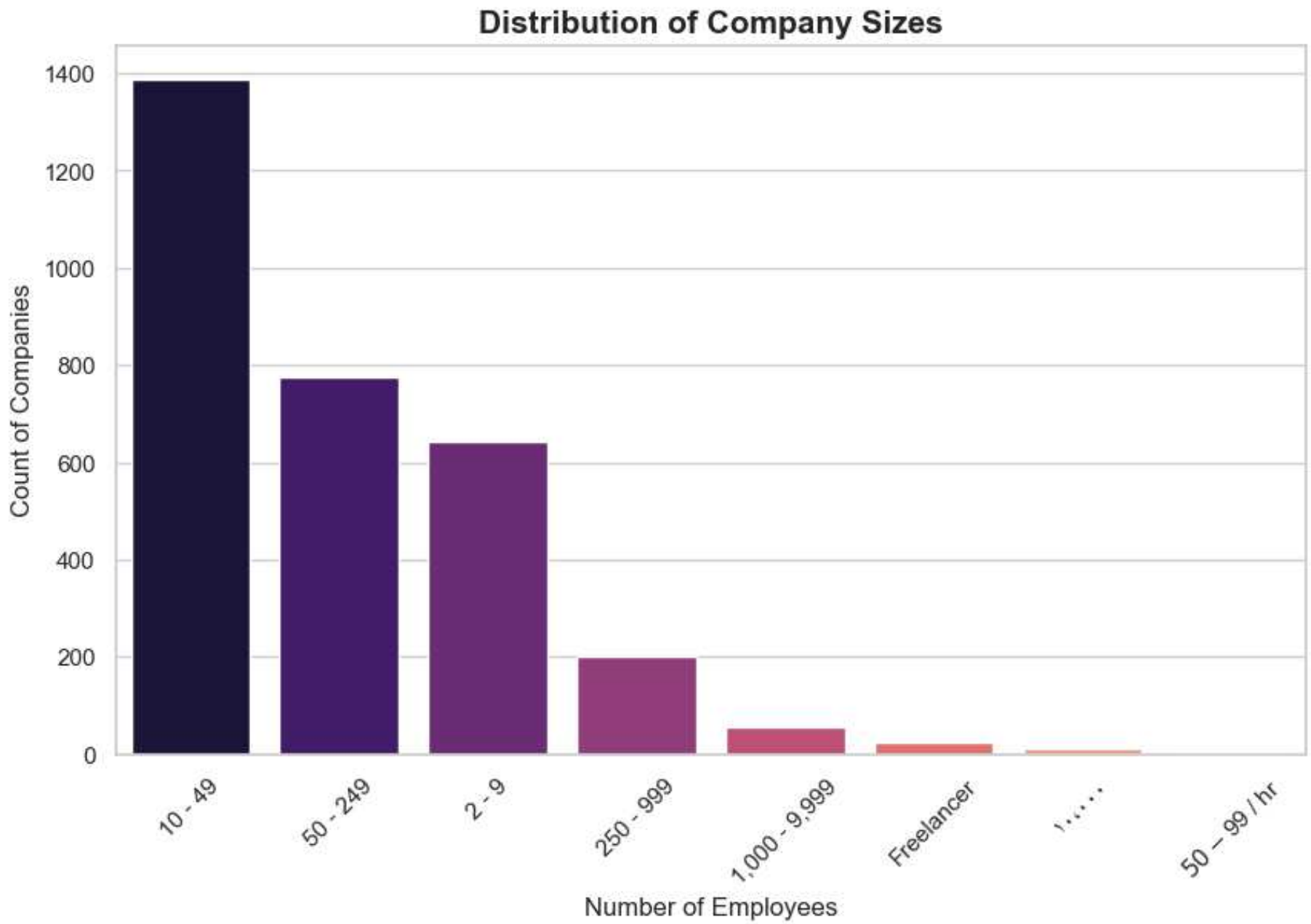
Sample of treated data (Original vs Cleaned):

|    | Percent AI Service Focus | AI_Focus_Numeric | AI_Focus_Cleaned |
|----|--------------------------|------------------|------------------|
| 7  | 110.0                    | 400.0            | %٤٠٠             |
| 10 | 110.0                    | 900.0            | %٩٠٠             |
| 13 | 110.0                    | 900.0            | %٩٠٠             |
| 15 | 110.0                    | 500.0            | %٥٠٠             |

```
In [14]: import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.boxplot(y=df['AI_Focus_Numeric'], color="salmon")
plt.title('Before Handling Outliers\n(With Extreme Values)')
plt.ylabel('AI Service Focus (%)')
plt.subplot(1, 2, 2)
sns.boxplot(y=df['AI_Focus_Cleaned'], color="skyblue")
plt.title('After Handling Outliers\n(Capped Data)')
plt.ylabel('AI Service Focus (%)')
plt.tight_layout()
plt.show()
```



```
In [15]: import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10, 6))
sns.countplot(data=df, x='Number of Employees', palette="magma",
order=df['Number of Employees'].value_counts().index)
plt.title('Distribution of Company Sizes', fontsize=15, fontweight='bold')
plt.xlabel('Number of Employees', fontsize=12)
plt.ylabel('Count of Companies', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```

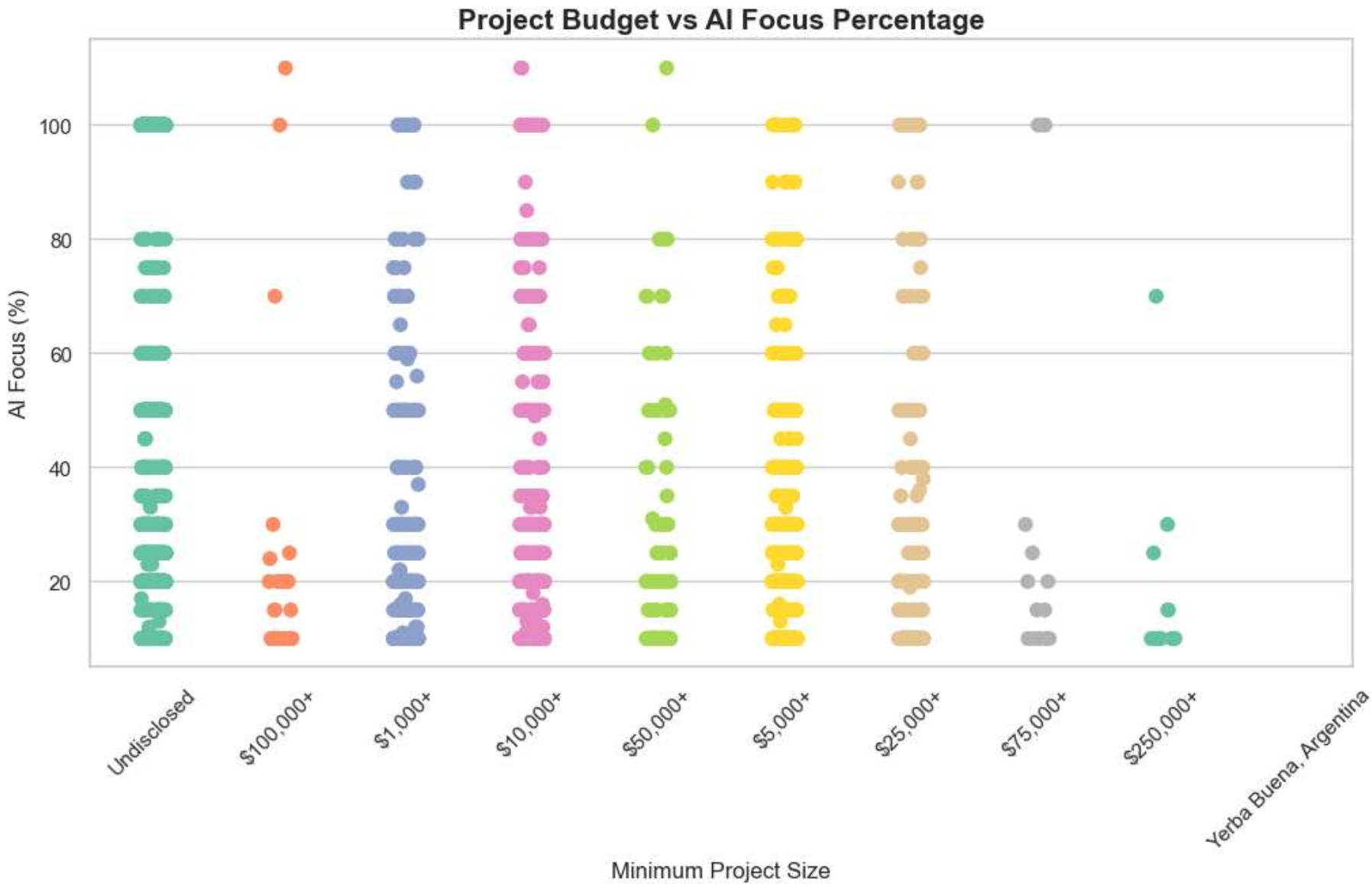


```
In [17]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(12, 6))
sns.stripplot(data=df, x='Minimum Project Size', y='AI_Focus_Cleaned',
palette="Set2", size=8, jitter=True)

plt.title('Project Budget vs AI Focus Percentage', fontsize=15, fontweight='bold')
plt.xlabel('Minimum Project Size', fontsize=12)
```

```
plt.ylabel('AI Focus (%)', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```

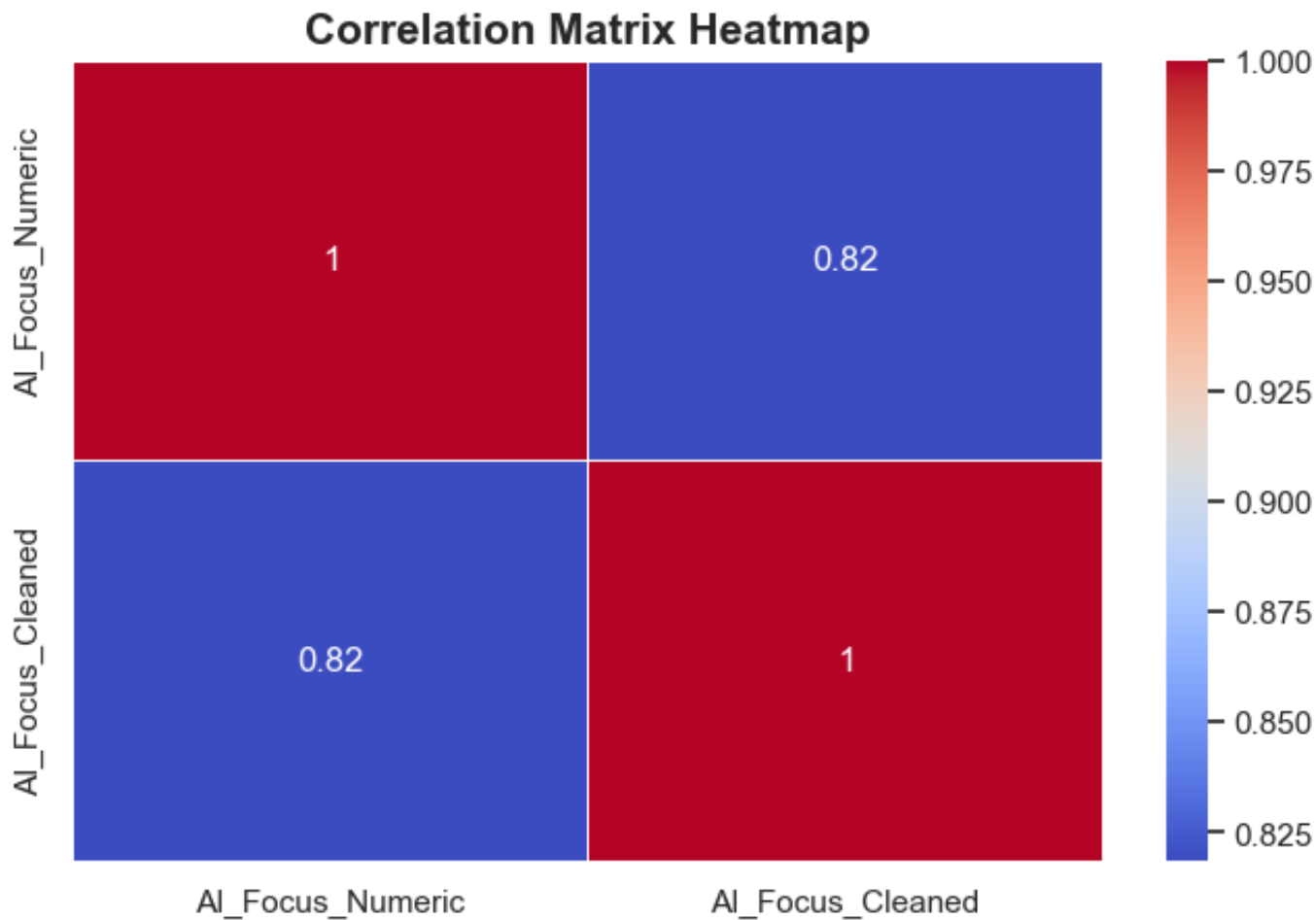


```
In [18]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(8, 5))
correlation = df[['AI_Focus_Numeric', 'AI_Focus_Cleaned']].corr()

sns.heatmap(correlation, annot=True, cmap='coolwarm', linewidths=0.5)

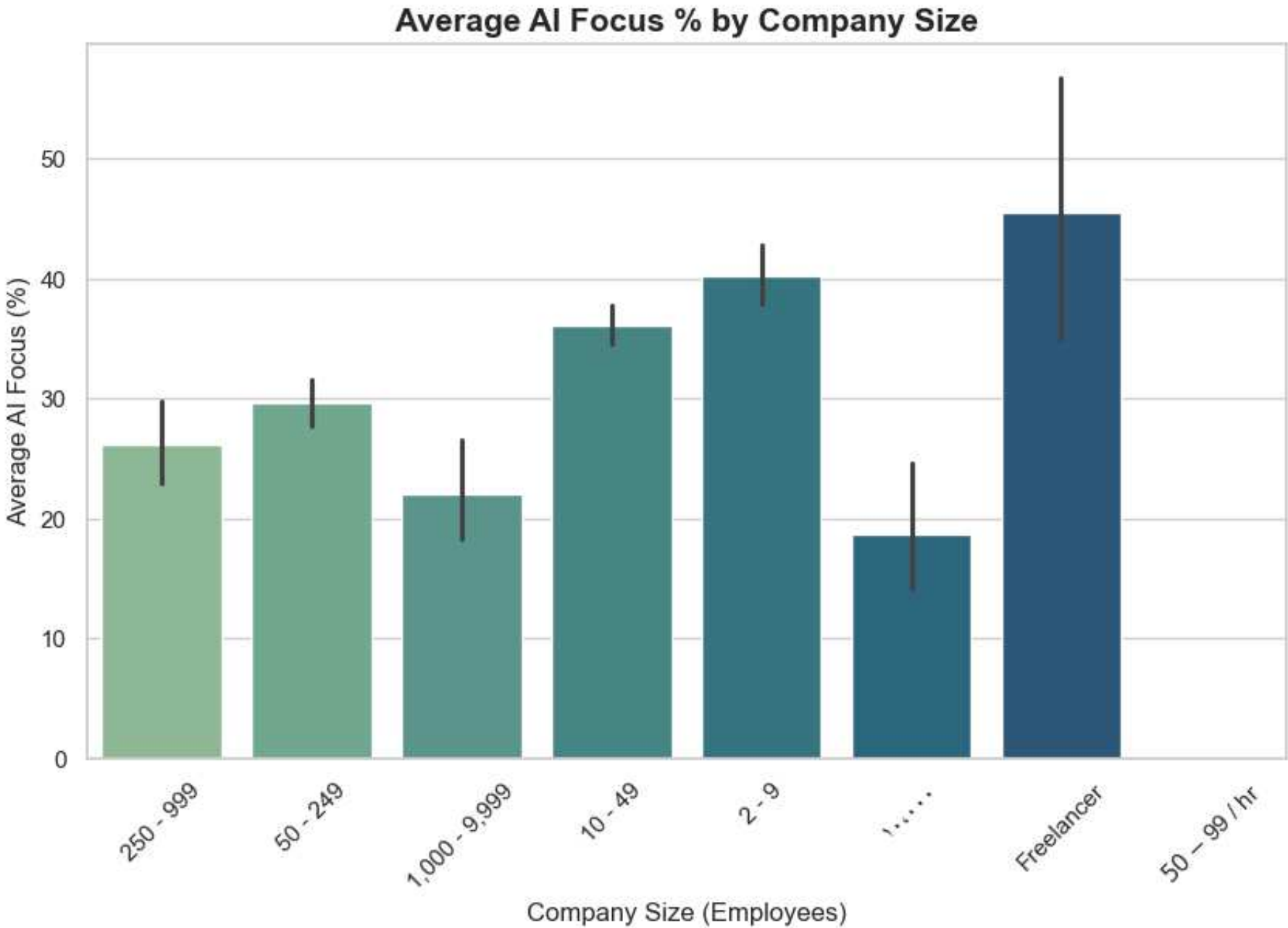
plt.title('Correlation Matrix Heatmap', fontsize=15, fontweight='bold')
plt.show()
```



```
In [19]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.barplot(data=df, x='Number of Employees', y='AI_Focus_Cleaned', palette="crest")
```

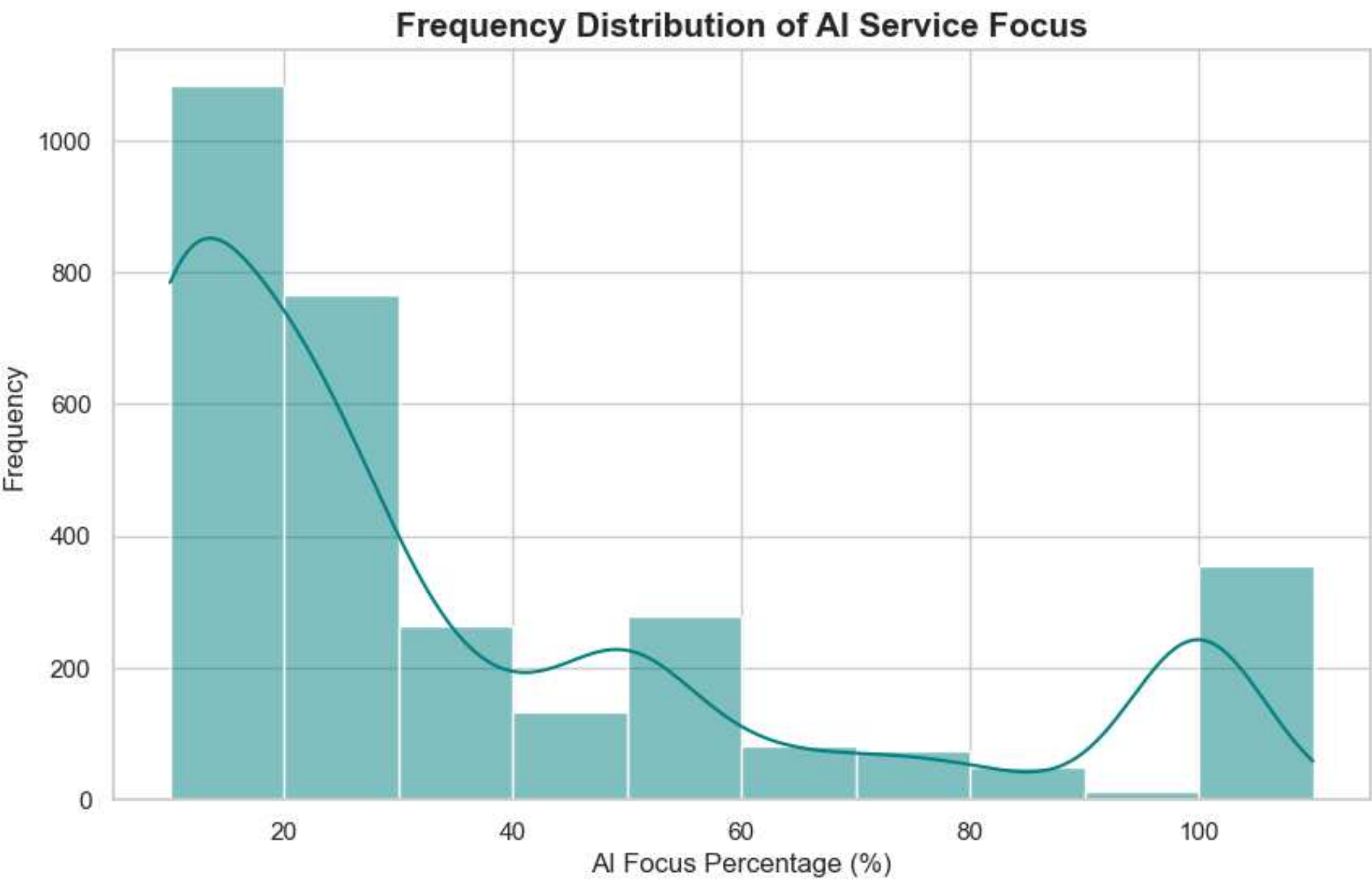
```
plt.title('Average AI Focus % by Company Size', fontsize=15, fontweight='bold')
plt.xlabel('Company Size (Employees)', fontsize=12)
plt.ylabel('Average AI Focus (%)', fontsize=12)
plt.xticks(rotation=45)
plt.show()
```



```
In [21]: import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10, 6))
sns.histplot(df['AI_Focus_Cleaned'], kde=True, color="teal", bins=10)

plt.title('Frequency Distribution of AI Service Focus', fontsize=15, fontweight='bold')
plt.xlabel('AI Focus Percentage (%)', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.show()
```



In [ ]: