In [ ]:

```
# REMOTE WORK & MENTAL HEALTH
```

Import all required libraries

In [3]:

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

Load the Dataset

In [4]:

```
df = pd.read_csv('C:/Users/Dell/Desktop/DATA ANALYSIS/archive/Impact_of_Remote_Work_on_Mental_Health.csv')
df
```

Out[4]:

	Employee_ID	Age	Gender	Job_Role	Industry	Years_of_Experience	Work_Location	Hours_Worked_Per_Week	Number_of_Virtual_M
0	EMP0001	32	Non-binary	HR	Healthcare	13	Hybrid	47	
1	EMP0002	40	Female	Data Scientist	IT	3	Remote	52	
2	EMP0003	59	Non-binary	Software Engineer	Education	22	Hybrid	46	
3	EMP0004	27	Male	Software Engineer	Finance	20	Onsite	32	
4	EMP0005	49	Male	Sales	Consulting	32	Onsite	35	
...	...	...	...	...	...	...	...	...	
4995	EMP4996	32	Male	Sales	Consulting	4	Onsite	24	
4996	EMP4997	39	Female	Sales	Healthcare	27	Onsite	48	
4997	EMP4998	42	Female	Sales	Healthcare	21	Hybrid	34	
4998	EMP4999	27	Female	Sales	Healthcare	26	Remote	58	
4999	EMP5000	29	Male	HR	IT	30	Onsite	20	

5000 rows × 20 columns

Display the first 5 rows of the dataset

In [5]:

```
df.head()
```

Out[5]:

	Employee_ID	Age	Gender	Job_Role	Industry	Years_of_Experience	Work_Location	Hours_Worked_Per_Week	Number_of_Virtual_Meeti
0	EMP0001	32	Non-binary	HR	Healthcare	13	Hybrid	47	
1	EMP0002	40	Female	Data Scientist	IT	3	Remote	52	
2	EMP0003	59	Non-binary	Software Engineer	Education	22	Hybrid	46	
3	EMP0004	27	Male	Software Engineer	Finance	20	Onsite	32	
4	EMP0005	49	Male	Sales	Consulting	32	Onsite	35	

Prints information about the DataFrame.

In [8]:

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 20 columns):
 #   Column                             Non-Null Count  Dtype 
---  ------                             --------------  ----- 
 0   Employee_ID                        5000 non-null   object
 1   Age                                5000 non-null   int64 
 2   Gender                             5000 non-null   object
 3   Job_Role                           5000 non-null   object
 4   Industry                           5000 non-null   object
 5   Years_of_Experience                5000 non-null   int64 
 6   Work_Location                      5000 non-null   object
 7   Hours_Worked_Per_Week              5000 non-null   int64 
 8   Number_of_Virtual_Meetings         5000 non-null   int64 
 9   Work_Life_Balance_Rating           5000 non-null   int64 
 10  Stress_Level                       5000 non-null   object
 11  Mental_Health_Condition            3804 non-null   object
 12  Access_to_Mental_Health_Resources  5000 non-null   object
 13  Productivity_Change                5000 non-null   object
 14  Social_Isolation_Rating            5000 non-null   int64 
 15  Satisfaction_with_Remote_Work      5000 non-null   object
 16  Company_Support_for_Remote_Work    5000 non-null   int64 
 17  Physical_Activity                  3371 non-null   object
 18  Sleep_Quality                      5000 non-null   object
 19  Region                             5000 non-null   object
dtypes: int64(7), object(13)
memory usage: 781.4+ KB
None
```

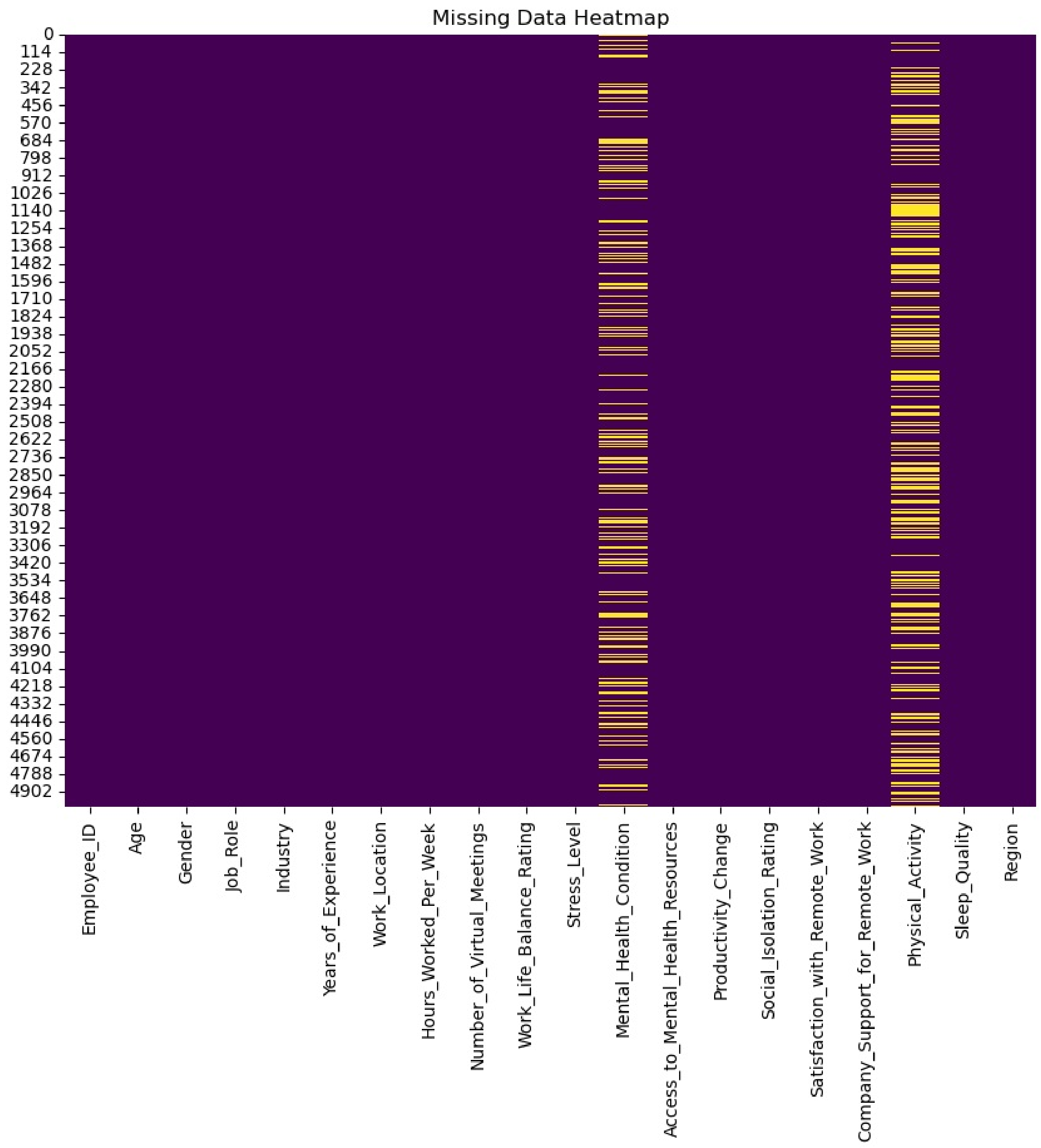
Count of null values in a column

In [52]:
```
print(df.isnull().sum())
```

```
Employee_ID                             0
Age                                     0
Gender                                  0
Job_Role                                0
Industry                                0
Years_of_Experience                     0
Work_Location                           0
Hours_Worked_Per_Week                   0
Number_of_Virtual_Meetings              0
Work_Life_Balance_Rating                0
Stress_Level                            0
Mental_Health_Condition              1196
Access_to_Mental_Health_Resources       0
Productivity_Change                     0
Social_Isolation_Rating                 0
Satisfaction_with_Remote_Work           0
Company_Support_for_Remote_Work         0
Physical_Activity                    1629
Sleep_Quality                           0
Region                                  0
dtype: int64
```

Show missing data using Heatmap

In [10]:
```
plt.figure(figsize=(10, 8))
sns.heatmap(df.isnull(), cbar=False, cmap='viridis')
plt.title('Missing Data Heatmap')
plt.show()
```



Replace Missing values

In [7]:
```
df['Mental_Health_Condition'].fillna('None', inplace=True)
df['Physical_Activity'].fillna(df['Physical_Activity'].mode()[0], inplace=True)
```

Again Check null values in a column

In [8]:
```
print(df.isnull().sum())
```

```
Employee_ID                          0
Age                                  0
Gender                               0
Job_Role                             0
Industry                             0
Years_of_Experience                  0
Work_Location                        0
Hours_Worked_Per_Week                0
Number_of_Virtual_Meetings           0
Work_Life_Balance_Rating             0
Stress_Level                         0
Mental_Health_Condition              0
Access_to_Mental_Health_Resources    0
Productivity_Change                  0
Social_Isolation_Rating              0
Satisfaction_with_Remote_Work        0
Company_Support_for_Remote_Work      0
Physical_Activity                    0
Sleep_Quality                        0
Region                               0
dtype: int64
```

In [9]:
```
df
```

Out[9]:

	Employee_ID	Age	Gender	Job_Role	Industry	Years_of_Experience	Work_Location	Hours_Worked_Per_Week	Number_of_Virtual_M
0	EMP0001	32	Non-binary	HR	Healthcare	13	Hybrid	47	
1	EMP0002	40	Female	Data Scientist	IT	3	Remote	52	
2	EMP0003	59	Non-binary	Software Engineer	Education	22	Hybrid	46	
3	EMP0004	27	Male	Software Engineer	Finance	20	Onsite	32	
4	EMP0005	49	Male	Sales	Consulting	32	Onsite	35	
...	...	...	...	...	...	...	...	...	
4995	EMP4996	32	Male	Sales	Consulting	4	Onsite	24	
4996	EMP4997	39	Female	Sales	Healthcare	27	Onsite	48	
4997	EMP4998	42	Female	Sales	Healthcare	21	Hybrid	34	
4998	EMP4999	27	Female	Sales	Healthcare	26	Remote	58	
4999	EMP5000	29	Male	HR	IT	30	Onsite	20	

5000 rows × 20 columns

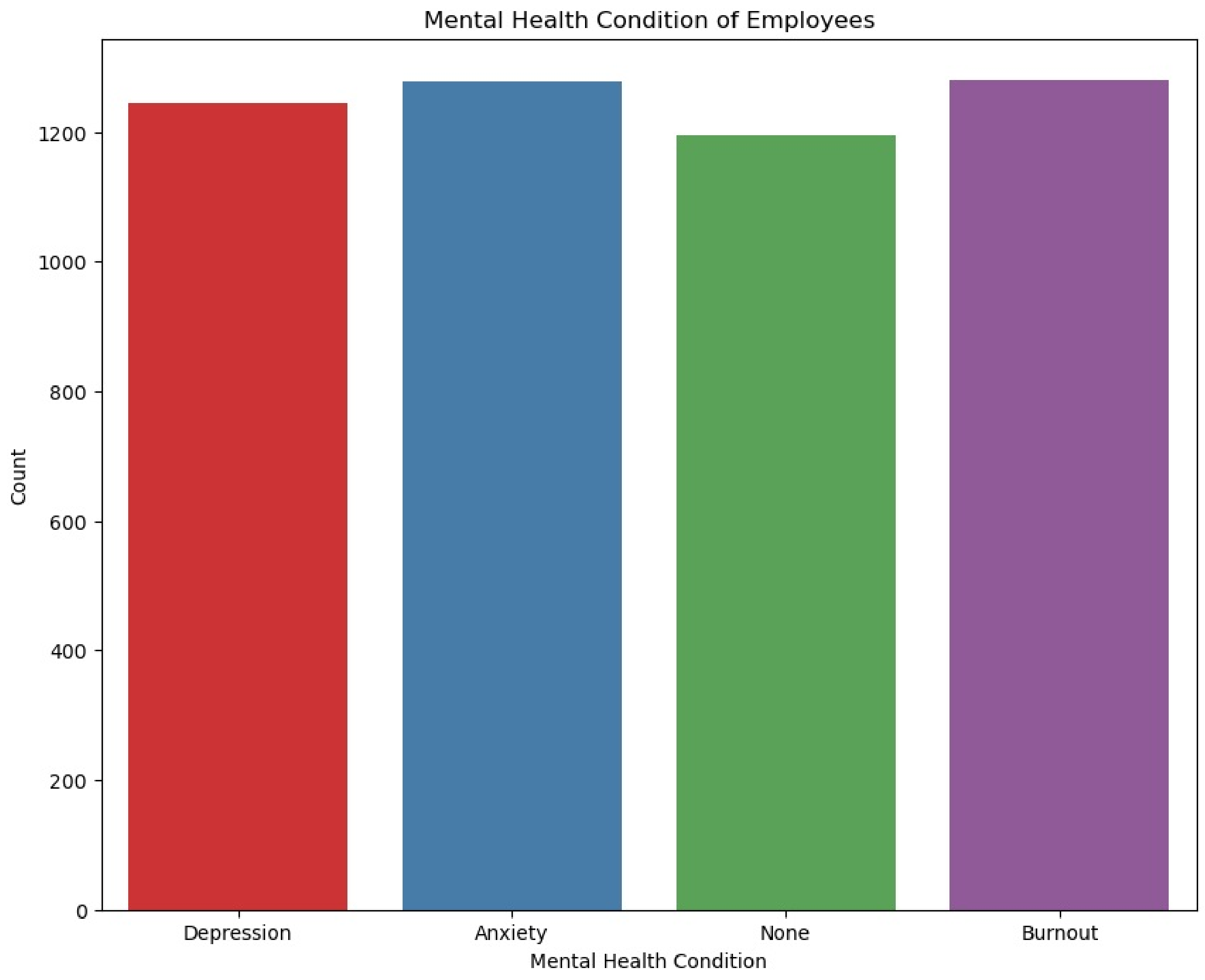
Export Cleaned dataset

In [10]:
```
df.to_csv("C:\\Users\\Dell\\Desktop\\DATA ANALYSIS\\archive\\modified_data.csv")
```

In [ ]:
```
                              # Data analysis & Visualization
```

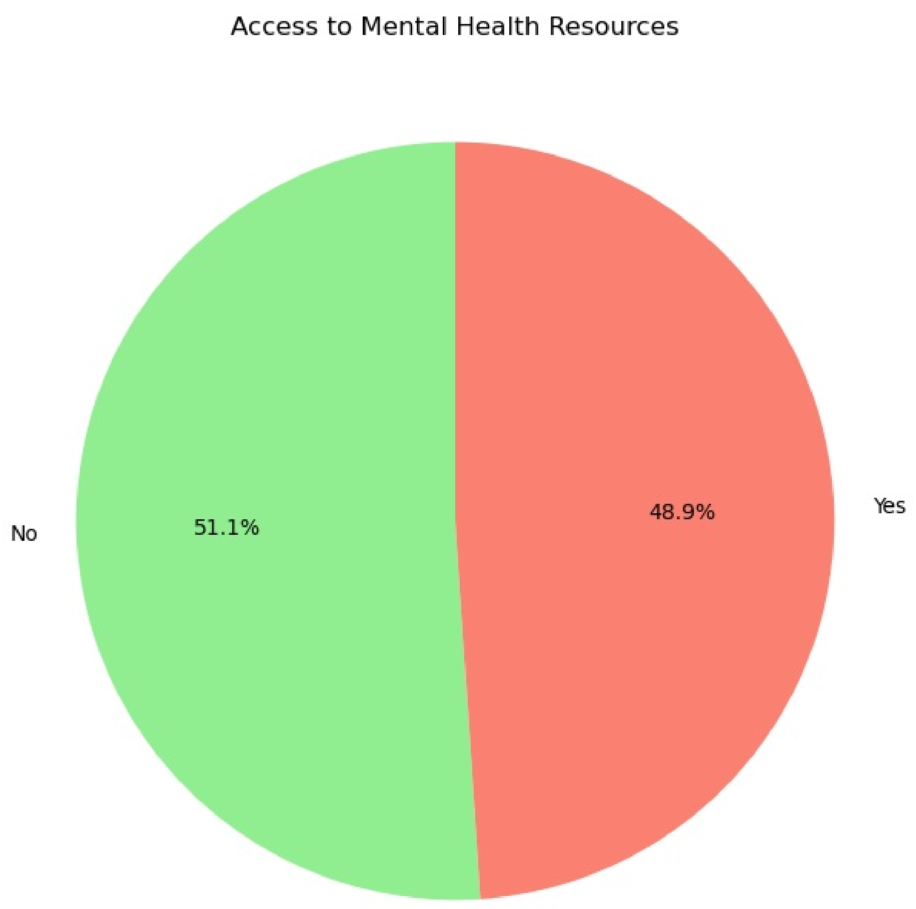
Mental Health Condition of Employees

In [18]:
```
plt.figure(figsize=(10,8))
sns.countplot(data=df, x='Mental_Health_Condition', palette='Set1')
plt.title('Mental Health Condition of Employees')
plt.xlabel('Mental Health Condition')
plt.ylabel('Count')
plt.show()
```



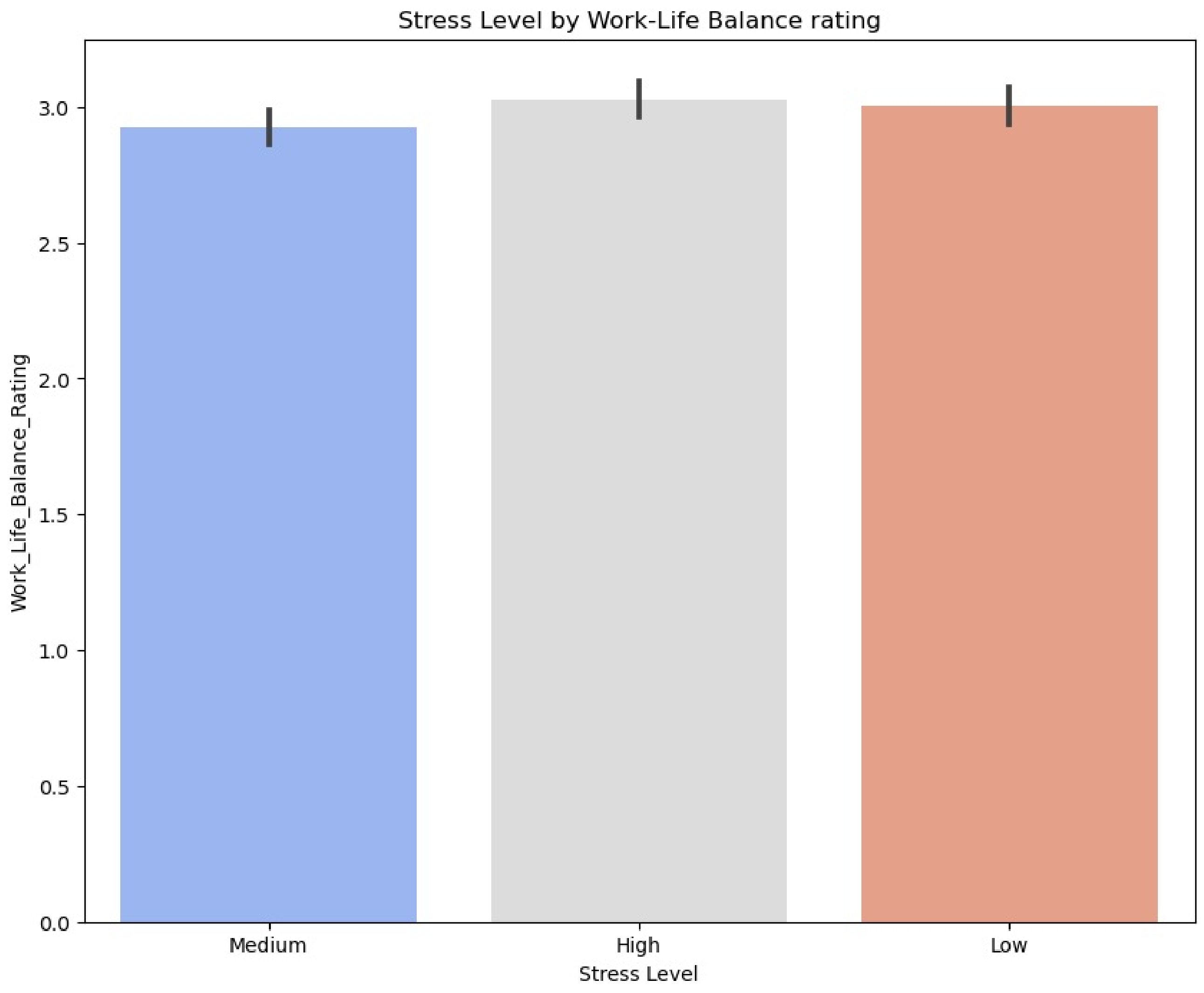
Access to Mental Health Resources

In [21]:
```
plt.figure(figsize=(10,8))
df['Access_to_Mental_Health_Resources'].value_counts().plot.pie(autopct='%1.1f%%',startangle=90,colors=['lightg
plt.title('Access to Mental Health Resources')
plt.ylabel('')
plt.show()
```



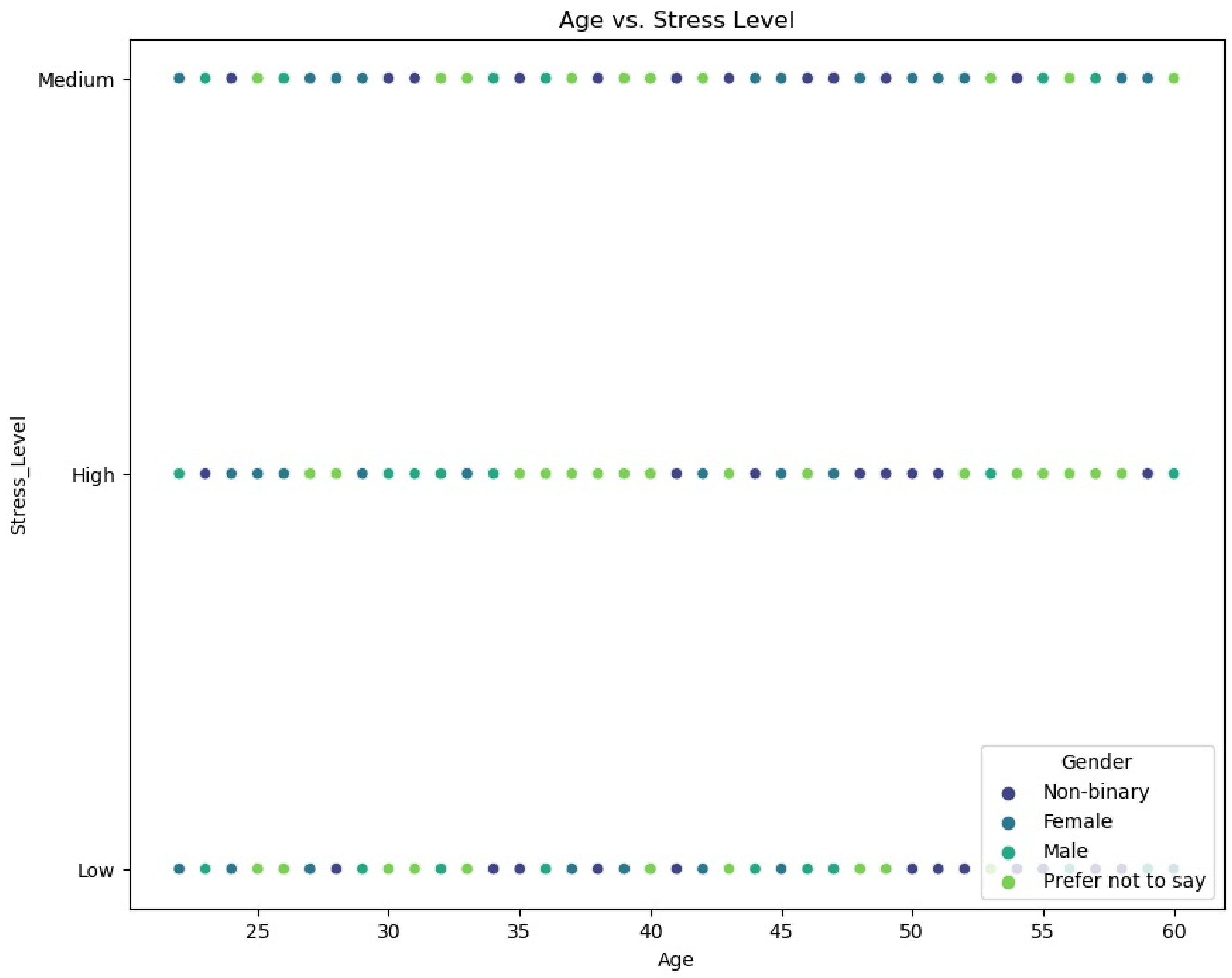
Stress Level by Work-Life Balance rating

In [26]:
```
plt.figure(figsize=(10,8))
sns.barplot(y='Work_Life_Balance_Rating', x='Stress_Level', data=df,palette="coolwarm")
plt.title('Stress Level by Work-Life Balance rating')
plt.xlabel('Stress Level')
plt.ylabel('Work_Life_Balance_Rating')
plt.show()
```



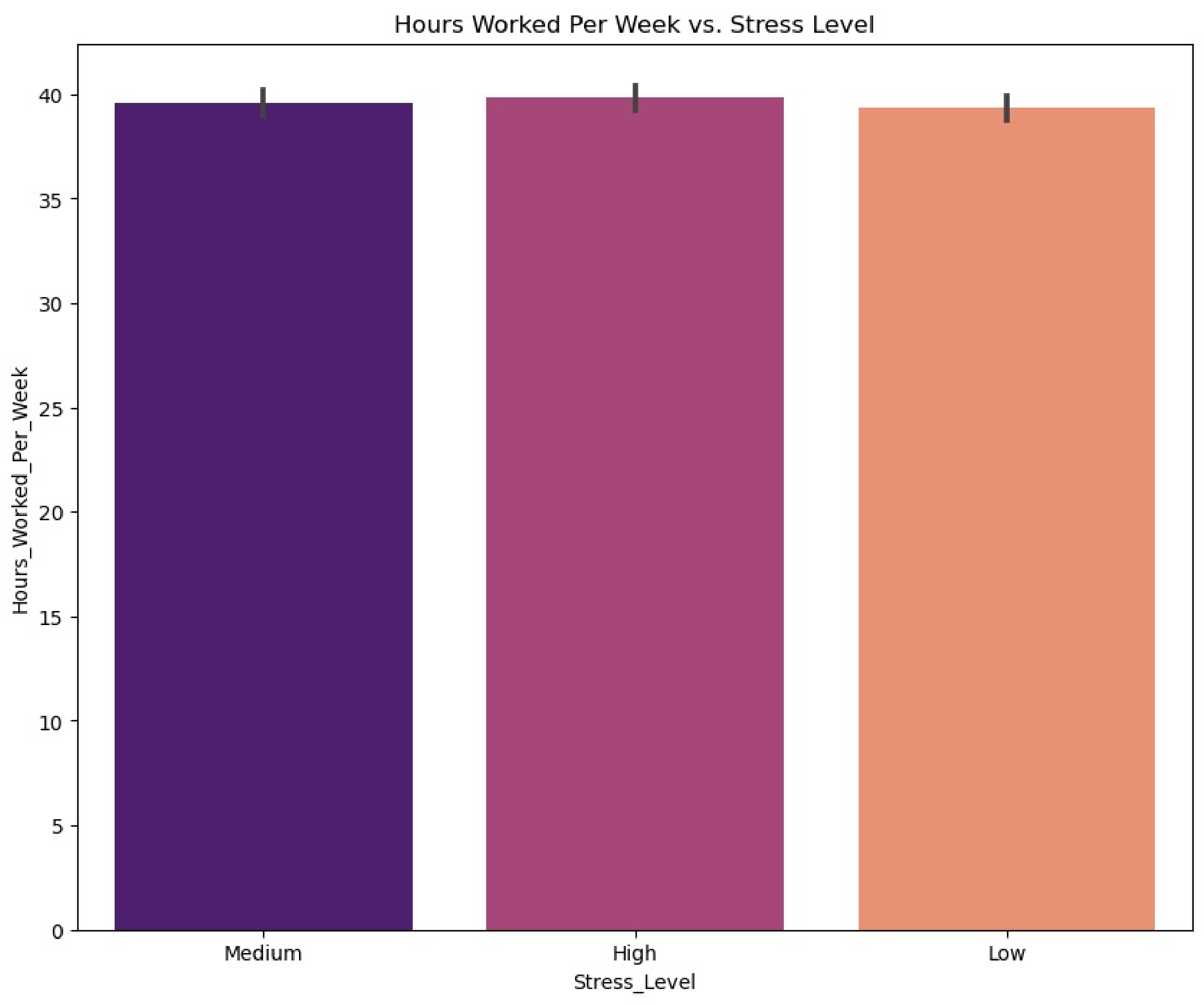
Stress Level vs Age

```
In [23]: plt.figure(figsize=(10,8))
         sns.scatterplot(data=df, x='Age', y='Stress_Level', hue='Gender', palette='viridis')
         plt.title('Age vs. Stress Level')
         plt.show()
```



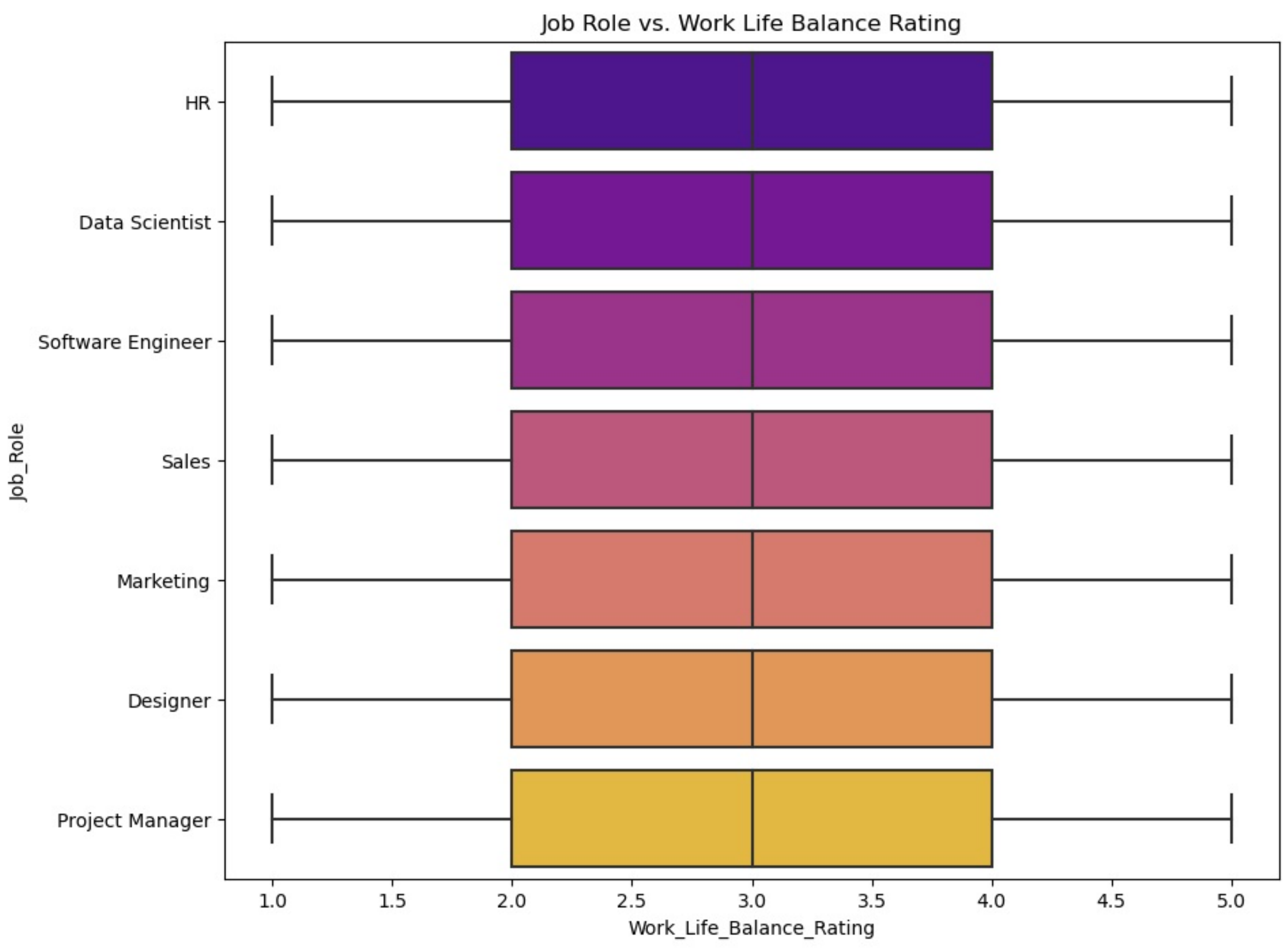
Hours Worked Per Week vs Stress Level

In [11]:
```
plt.figure(figsize=(10,8))
sns.barplot(data=df, x='Stress_Level', y='Hours_Worked_Per_Week', palette='magma')
plt.title('Hours Worked Per Week vs. Stress Level')
plt.show()
```



Job Role vs Work Life Balance Rating

```
In [29]: plt.figure(figsize=(10,8))
         sns.boxplot(data=df, x='Work_Life_Balance_Rating', y='Job_Role', palette='plasma')
         plt.title('Job Role vs. Work Life Balance Rating')
         plt.show()
```



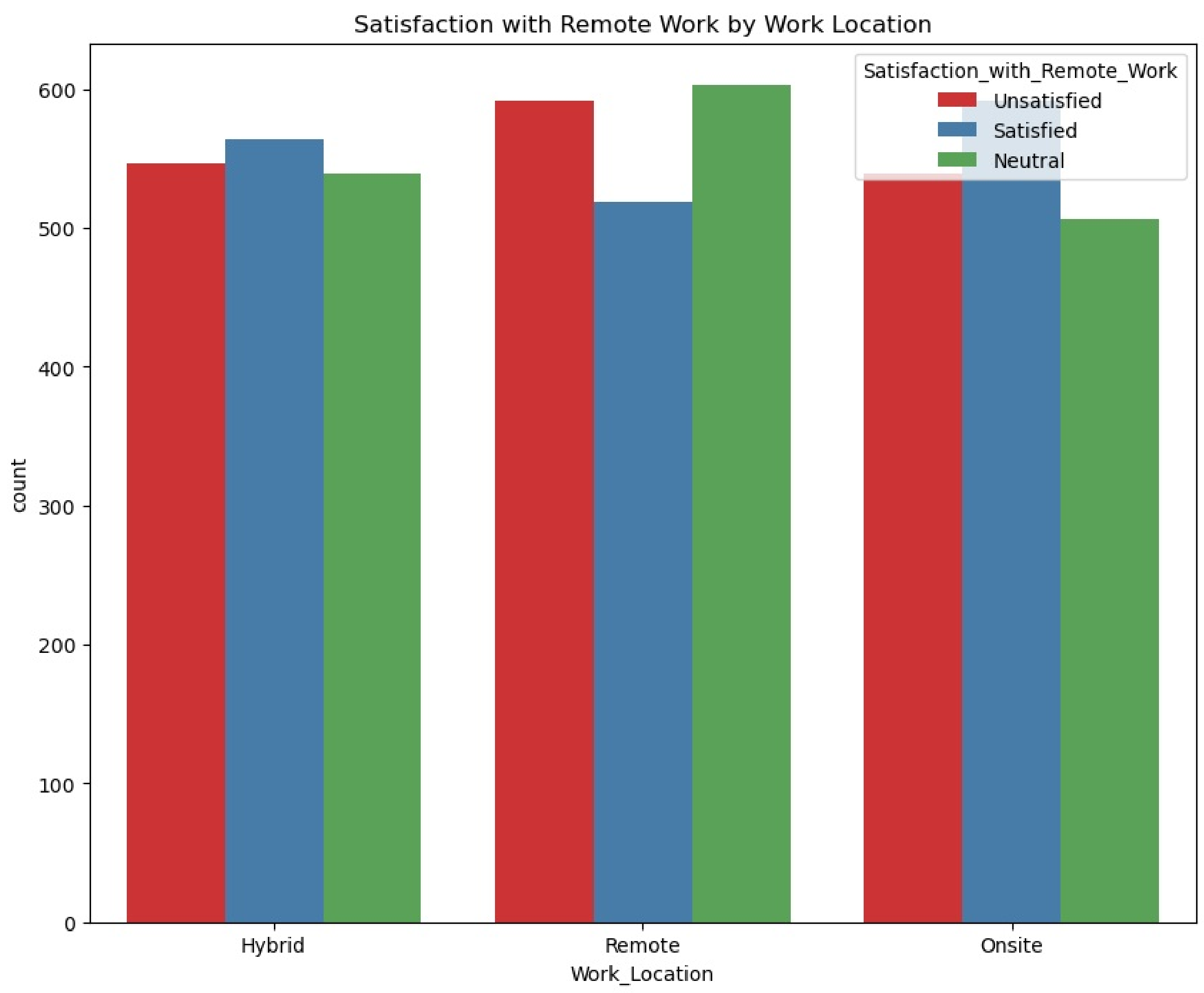
In [26]:
```
if df['Stress_Level'].dtype == 'object':
    df['Stress_Level'] = df['Stress_Level'].astype('category').cat.codes

print(df.dtypes)
```

```
Employee_ID                          object
Age                                   int64
Gender                               object
Job_Role                             object
Industry                             object
Years_of_Experience                   int64
Work_Location                        object
Hours_Worked_Per_Week                 int64
Number_of_Virtual_Meetings            int64
Work_Life_Balance_Rating              int64
Stress_Level                           int8
Mental_Health_Condition              object
Access_to_Mental_Health_Resources    object
Productivity_Change                  object
Social_Isolation_Rating               int64
Satisfaction_with_Remote_Work        object
Company_Support_for_Remote_Work       int64
Physical_Activity                    object
Sleep_Quality                        object
Region                               object
dtype: object
```

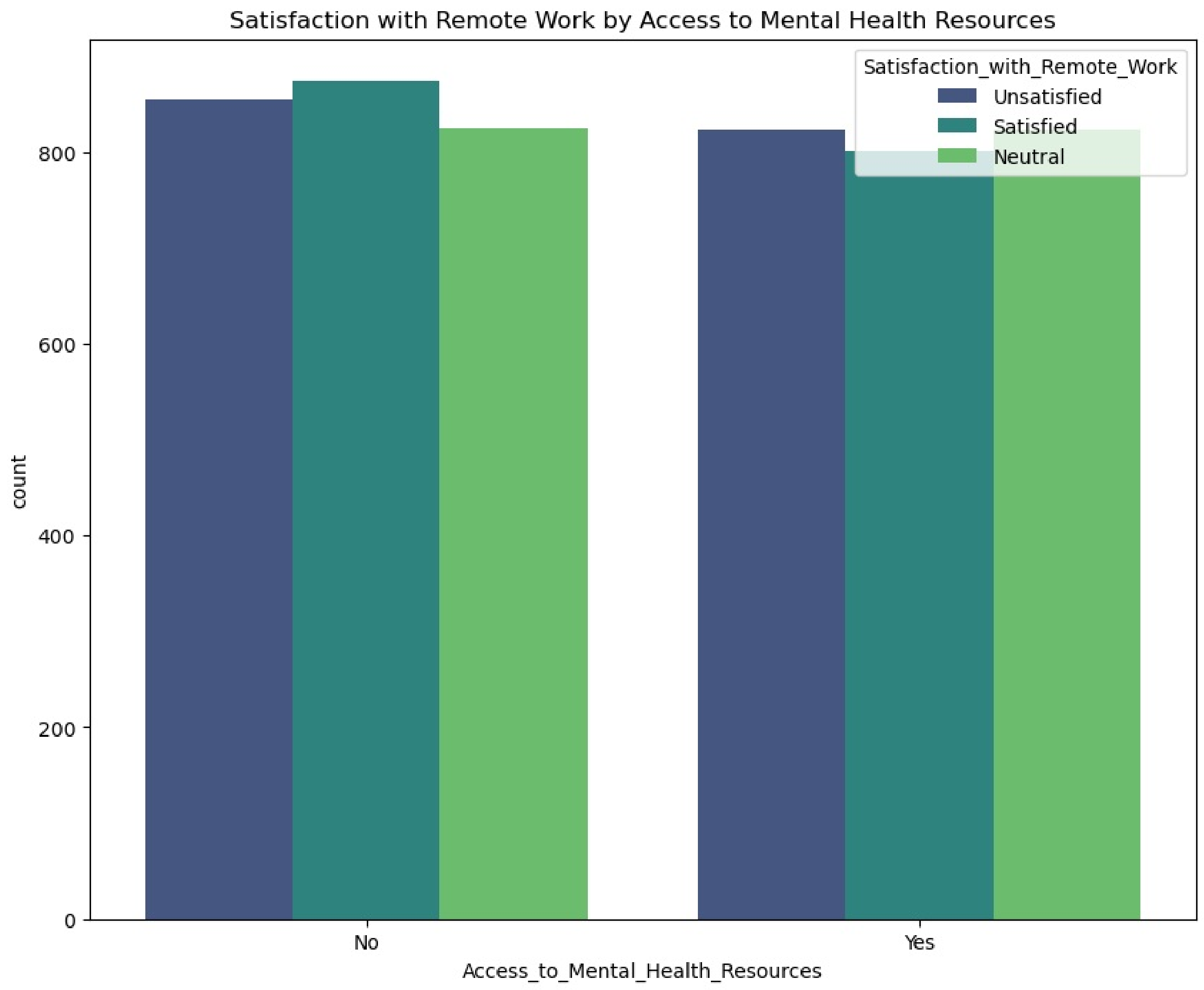
Satisfaction with Remote Work by Work Location

In [25]:
```
plt.figure(figsize=(10,8))
sns.countplot(data=df, x='Work_Location', hue='Satisfaction_with_Remote_Work', palette='Set1')
plt.title('Satisfaction with Remote Work by Work Location')
plt.show()
```



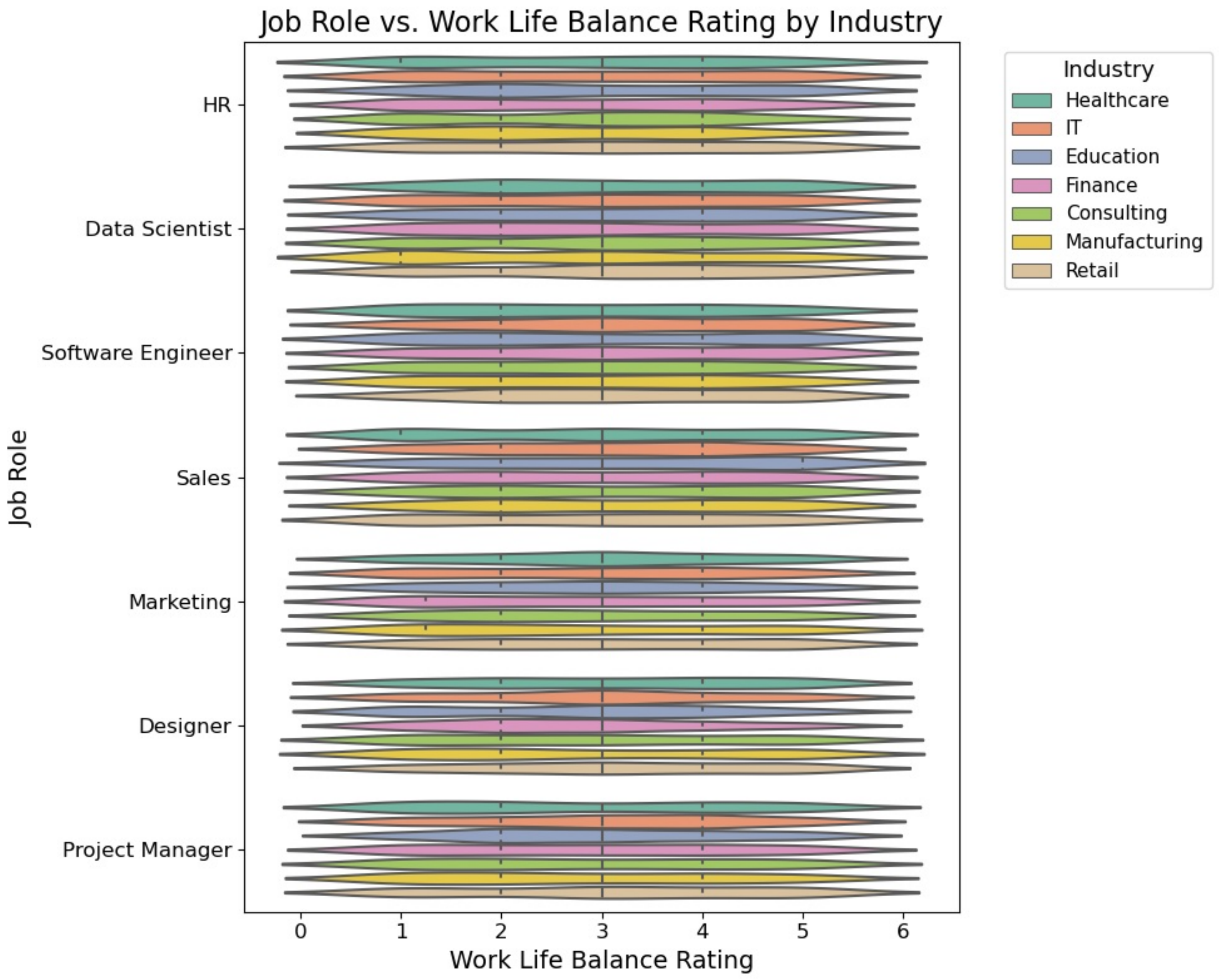
Satisfaction with Remote Work by Access to Mental Health Resources

In [31]:
```
plt.figure(figsize=(10,8))
sns.countplot(data=df, x='Access_to_Mental_Health_Resources', hue='Satisfaction_with_Remote_Work', palette='vir
plt.title('Satisfaction with Remote Work by Access to Mental Health Resources')
plt.show()
```



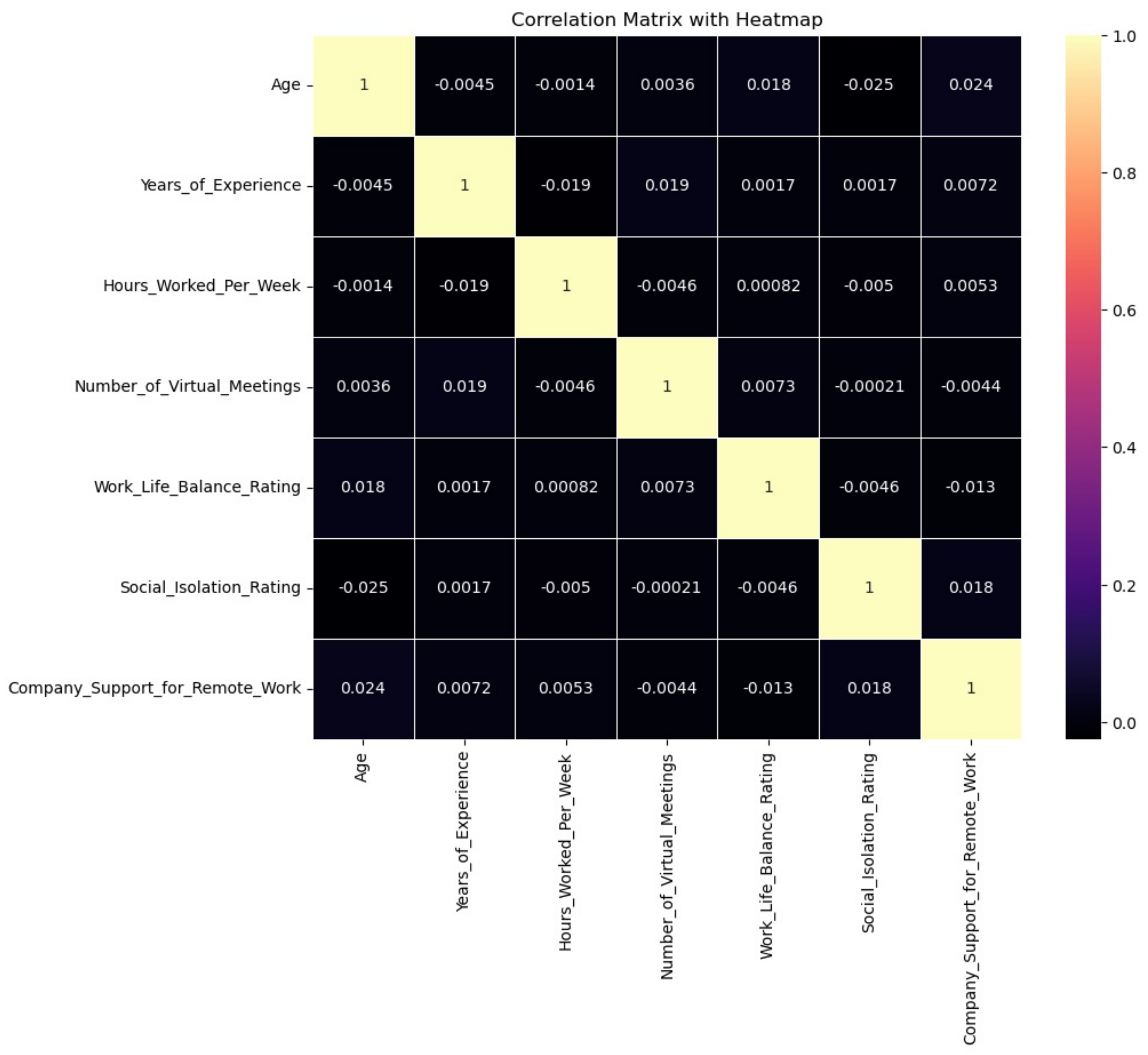
Job Role vs. Work Life Balance Rating by Industry

In [39]:
```
plt.figure(figsize=(10,8))
sns.violinplot(data=df, x='Work_Life_Balance_Rating', y='Job_Role', hue='Industry', palette='Set2', inner="quar
plt.title('Job Role vs. Work Life Balance Rating by Industry', fontsize=16)
plt.xlabel('Work Life Balance Rating', fontsize=14)
plt.ylabel('Job Role', fontsize=14)
plt.legend(title='Industry', title_fontsize='13', fontsize='11', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.tight_layout()
plt.show()
```



Correlation Matrix with Heatmap

In [17]:
```
numeric_columns = df.select_dtypes(include=['number'])
correlation_matrix = numeric_columns.corr()
plt.figure(figsize=(10,8))
sns.heatmap(correlation_matrix, annot=True, cmap='magma', linewidths=0.5)
plt.title('Correlation Matrix with Heatmap')
plt.show()
```



	Age	Years_of_Experience	Hours_Worked_Per_Week	Number_of_Virtual_Meetings	Work_Life_Balance_Rating	Social_Isolation_Rating	Company_Support_for_Remote_Work
Age	1	-0.0045	-0.0014	0.0036	0.018	-0.025	0.024
Years_of_Experience	-0.0045	1	-0.019	0.019	0.0017	0.0017	0.0072
Hours_Worked_Per_Week	-0.0014	-0.019	1	-0.0046	0.00082	-0.005	0.0053
Number_of_Virtual_Meetings	0.0036	0.019	-0.0046	1	0.0073	-0.00021	-0.0044
Work_Life_Balance_Rating	0.018	0.0017	0.00082	0.0073	1	-0.0046	-0.013
Social_Isolation_Rating	-0.025	0.0017	-0.005	-0.00021	-0.0046	1	0.018
Company_Support_for_Remote_Work	0.024	0.0072	0.0053	-0.0044	-0.013	0.018	1

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js