In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:
```
productivity_df = pd.read_csv('remote_work_productivity.csv', index_col=0).dropn
```

In [4]:
```
productivity_df.head()
```

Out[4]:

Employee_ID	Employment_Type	Hours_Worked_Per_Week	Productivity_Score	Well_Bein
1	Remote	29	75	
2	In-Office	45	49	
3	Remote	34	74	
4	Remote	25	81	
5	Remote	50	70	

In [5]:
```
productivity_df.info()
```

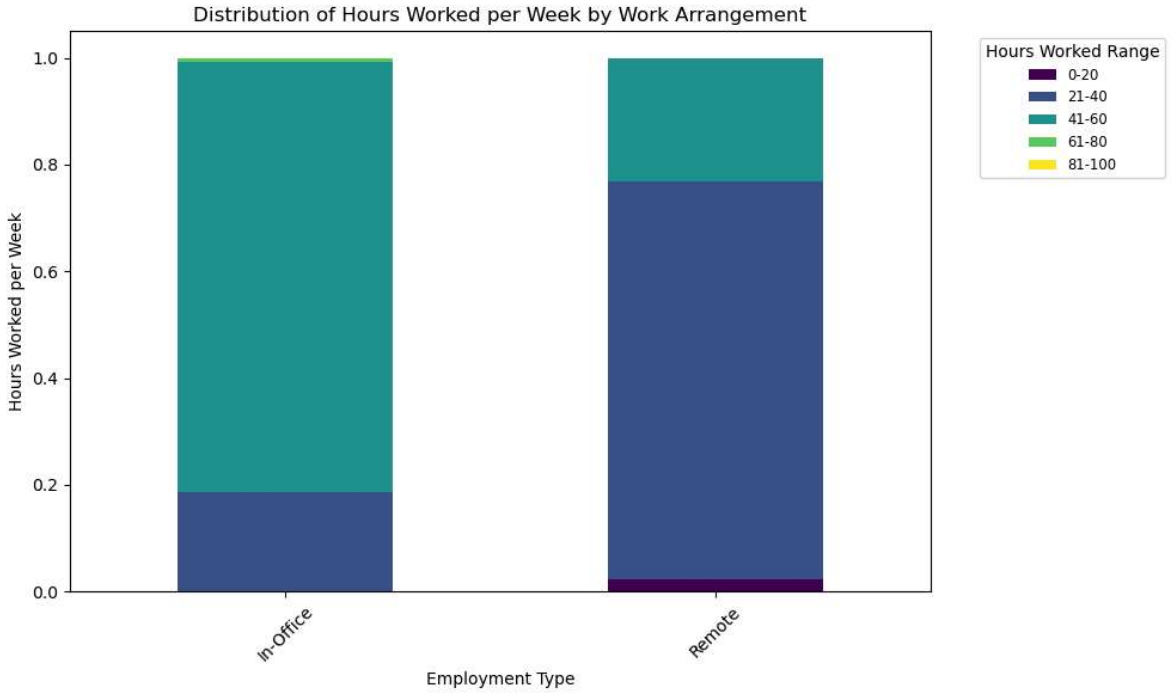
```
<class 'pandas.core.frame.DataFrame'>
Index: 1000 entries, 1 to 1000
Data columns (total 4 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Employment_Type       1000 non-null   object
 1   Hours_Worked_Per_Week 1000 non-null   int64
 2   Productivity_Score    1000 non-null   int64
 3   Well_Being_Score      1000 non-null   int64
dtypes: int64(3), object(1)
memory usage: 39.1+ KB
```

In [12]:
```
productivity_df.groupby('Employment_Type')['Hours_Worked_Per_Week'].value_counts
```

Out[12]:
```
Employment_Type  Hours_Worked_Per_Week
In-Office        45                      0.088235
                 41                      0.078431
                 44                      0.074510
                 40                      0.066667
                 47                      0.064706
                                            ...
Remote           16                      0.004082
                 51                      0.002041
                 52                      0.002041
                 55                      0.002041
                 56                      0.002041
Name: proportion, Length: 72, dtype: float64
```

In [6]:
```
bins = [0, 20, 40, 60, 80, 100]
labels = ["0-20", "21-40", "41-60", "61-80", "81-100"]

productivity_df['Hours_Worked_Range'] = pd.cut(productivity_df['Hours_Worked_Per
```

```
grouped_data = productivity_df.groupby('Employment_Type')['Hours_Worked_Range'].

ax = grouped_data.plot(kind='bar', stacked=True, colormap='viridis', figsize=(10

plt.xlabel('Employment Type')
plt.ylabel('Hours Worked per Week')
plt.title('Distribution of Hours Worked per Week by Work Arrangement')

plt.legend(title='Hours Worked Range', bbox_to_anchor=(1.05, 1), loc='upper left
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

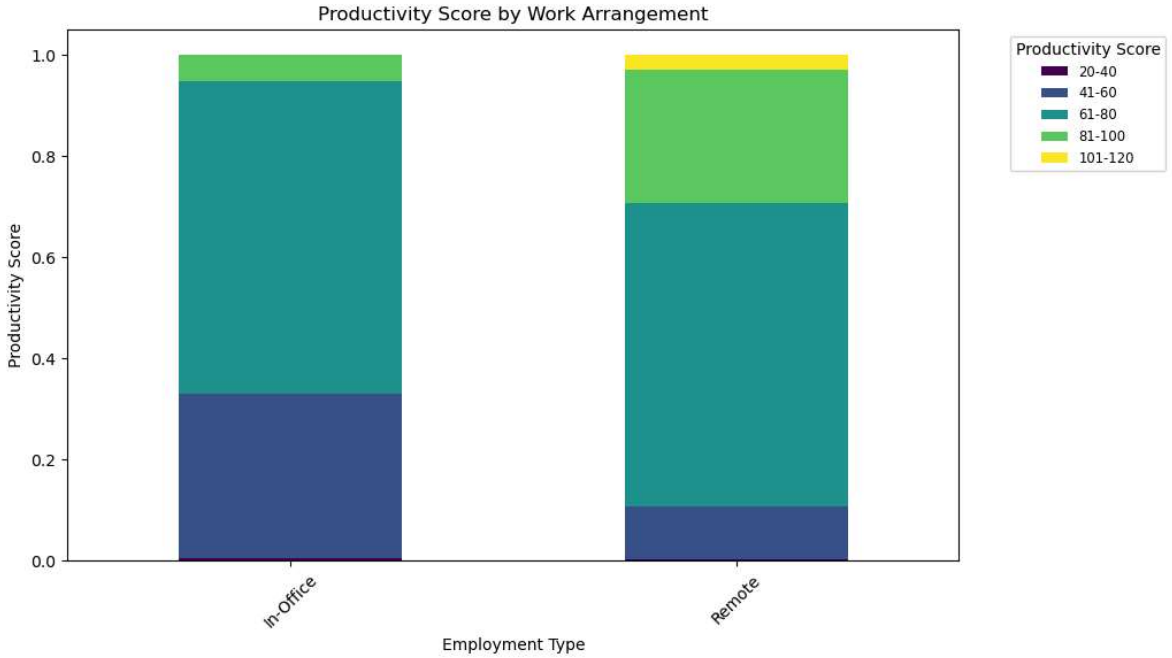


In [7]:
```
productivity_df.groupby('Employment_Type')['Productivity_Score'].value_counts(no
```

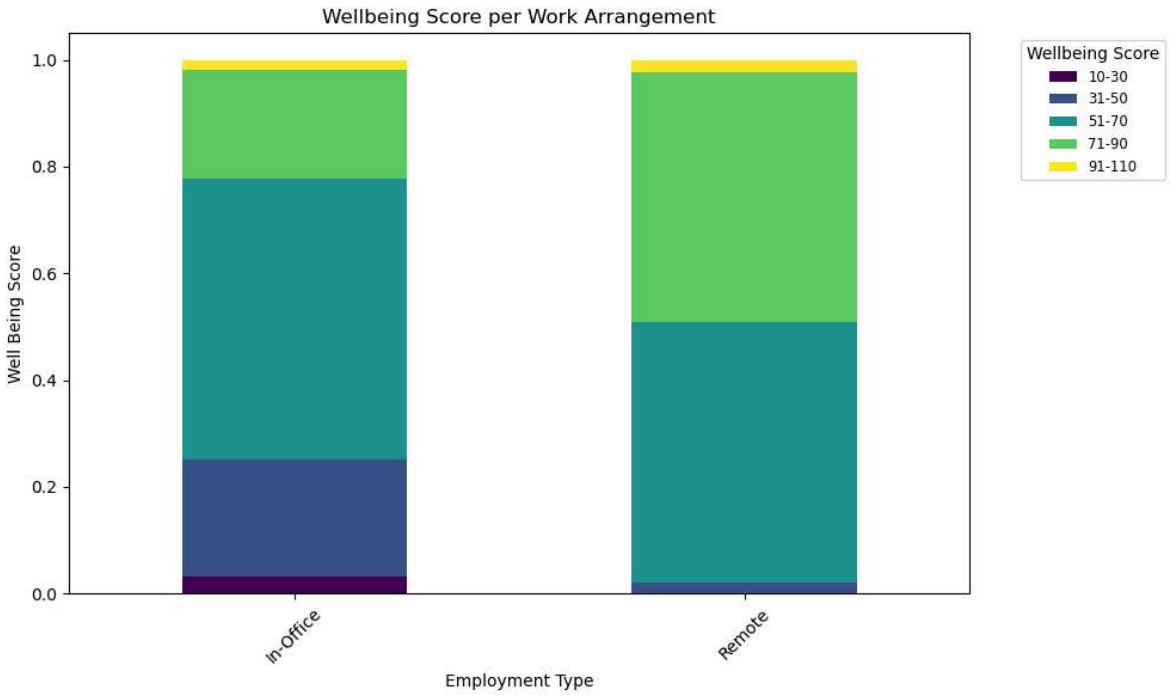
Out[7]:
```
Employment_Type  Productivity_Score
In-Office        66                   0.047059
                 63                   0.047059
                 60                   0.043137
                 74                   0.039216
                 62                   0.039216
                                        ...
Remote           52                   0.002041
                 51                   0.002041
                 47                   0.002041
                 44                   0.002041
                 36                   0.002041
Name: proportion, Length: 117, dtype: float64
```

In [8]:
```
bins = [20, 40, 60, 80, 100, 120]
labels = ["20-40", "41-60", "61-80", "81-100", "101-120"]

productivity_df['Productivity_Score'] = pd.cut(productivity_df['Productivity_Sco

grouped_data = productivity_df.groupby('Employment_Type')['Productivity_Score'].

ax = grouped_data.plot(kind='bar', stacked=True, colormap='viridis', figsize=(10
```

```
plt.xlabel('Employment Type')
plt.ylabel('Productivity Score')
plt.title('Productivity Score by Work Arrangement')

plt.legend(title='Productivity Score', bbox_to_anchor=(1.05, 1), loc='upper left
plt.xticks(rotation=45)
plt.show()
```



In [9]:
```
bins = [10, 30, 50, 70, 90, 110]
labels = ["10-30", "31-50", "51-70", "71-90", "91-110"]

productivity_df['Well_Being_Score'] = pd.cut(productivity_df['Well_Being_Score']

grouped_data = productivity_df.groupby('Employment_Type')['Well_Being_Score'].va

ax = grouped_data.plot(kind='bar', stacked=True, colormap='viridis', figsize=(10

plt.xlabel('Employment Type')
plt.ylabel('Well Being Score')
plt.title('Wellbeing Score per Work Arrangement')

plt.legend(title='Wellbeing Score', bbox_to_anchor=(1.05, 1), loc='upper left',
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [31]:
```
mental_health_df = pd.read_csv('Impact_of_Remote_Work_on_Mental_Health.csv', ind
```

In [33]:
```
mental_health_df.head()
```

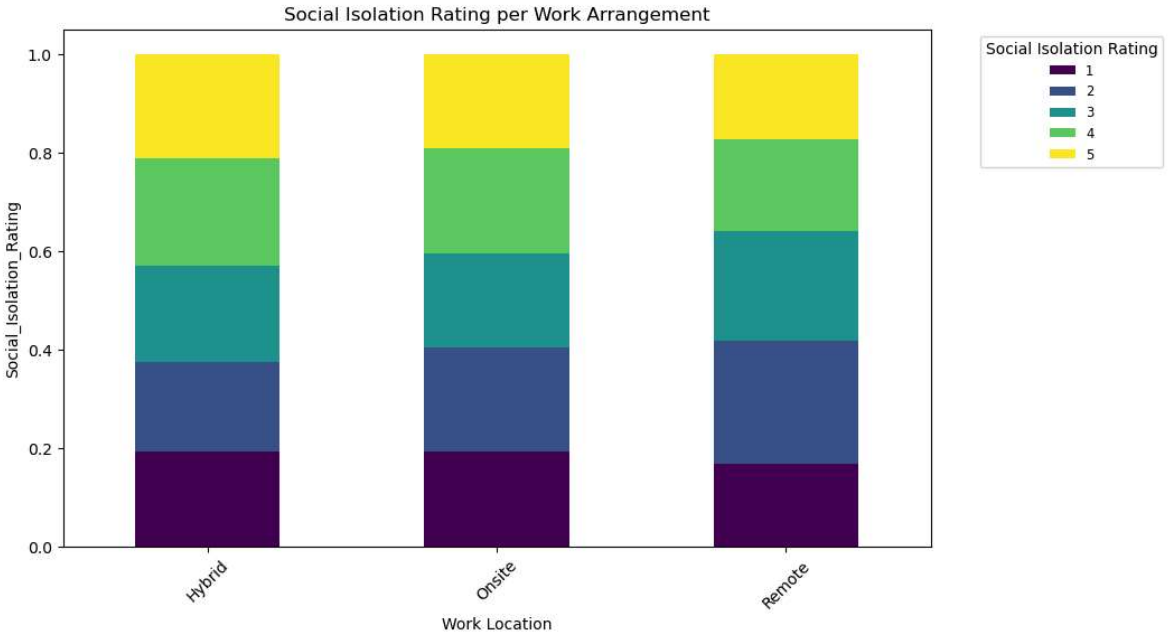
Out[33]:

Employee_ID	Age	Gender	Job_Role	Industry	Years_of_Experience	Work_Location
EMP0001	32	Non-binary	HR	Healthcare	13	Hybrid
EMP0002	40	Female	Data Scientist	IT	3	Remote
EMP0007	31	Prefer not to say	Sales	IT	24	Remote
EMP0010	30	Female	HR	IT	28	Hybrid
EMP0013	40	Female	Marketing	Consulting	1	Remote

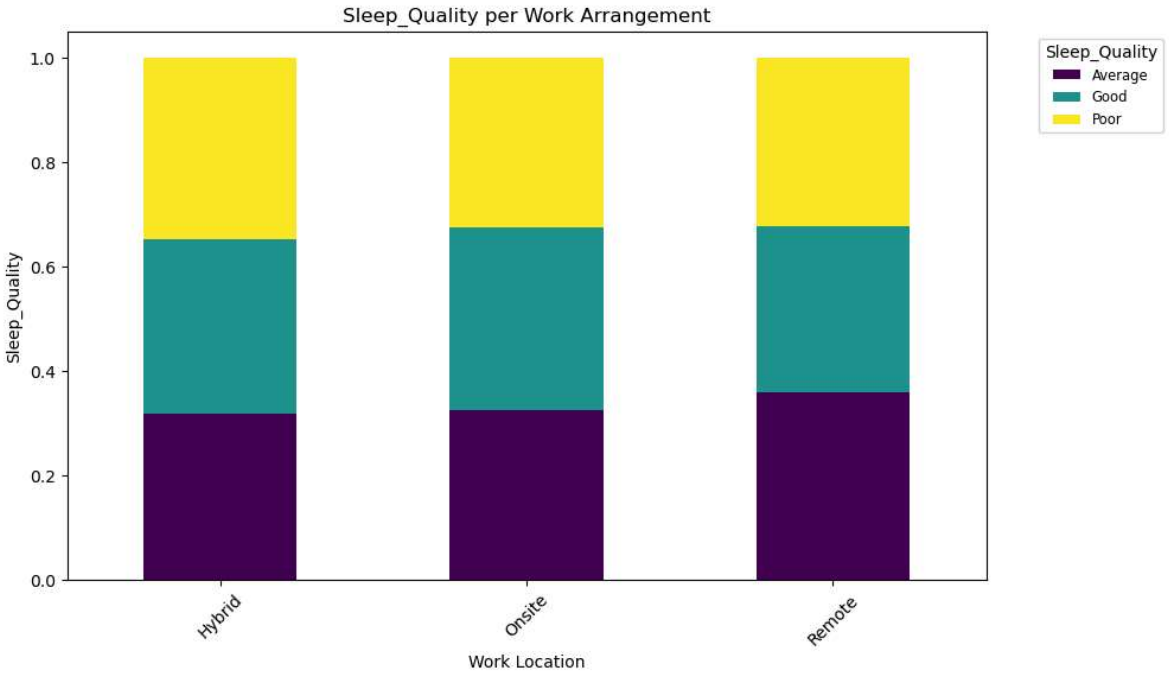
In [34]:
```
mental_health_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2577 entries, EMP0001 to EMP4998
Data columns (total 19 columns):
 #   Column                            Non-Null Count  Dtype
---  ------                            --------------  -----
 0   Age                               2577 non-null   int64
 1   Gender                            2577 non-null   object
 2   Job_Role                          2577 non-null   object
 3   Industry                          2577 non-null   object
 4   Years_of_Experience               2577 non-null   int64
 5   Work_Location                     2577 non-null   object
 6   Hours_Worked_Per_Week             2577 non-null   int64
 7   Number_of_Virtual_Meetings        2577 non-null   int64
 8   Work_Life_Balance_Rating          2577 non-null   int64
 9   Stress_Level                      2577 non-null   object
 10  Mental_Health_Condition           2577 non-null   object
 11  Access_to_Mental_Health_Resources 2577 non-null   object
 12  Productivity_Change               2577 non-null   object
 13  Social_Isolation_Rating           2577 non-null   int64
 14  Satisfaction_with_Remote_Work     2577 non-null   object
 15  Company_Support_for_Remote_Work   2577 non-null   int64
 16  Physical_Activity                 2577 non-null   object
 17  Sleep_Quality                     2577 non-null   object
 18  Region                            2577 non-null   object
dtypes: int64(7), object(12)
memory usage: 402.7+ KB
```

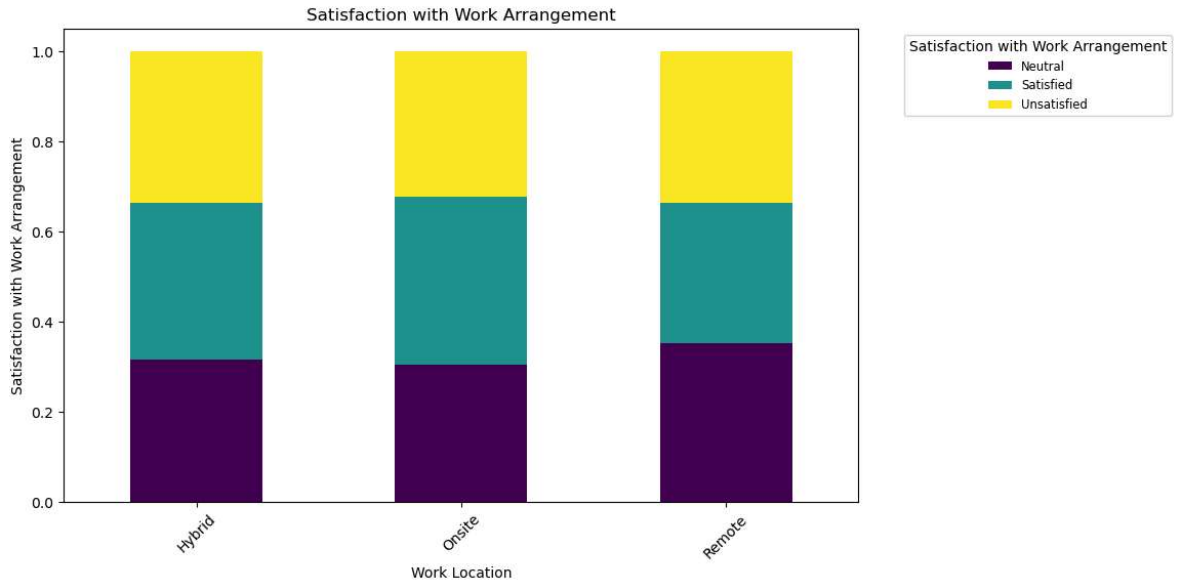
In [44]:
```
mental_health_df.groupby('Work_Location')['Social_Isolation_Rating'].value_count

grouped_data = mental_health_df.groupby('Work_Location')['Social_Isolation_Ratin

ax = grouped_data.plot(kind='bar', stacked=True, colormap='viridis', figsize=(10

plt.xlabel('Work Location')
plt.ylabel('Social_Isolation_Rating')
plt.title('Social Isolation Rating per Work Arrangement')

plt.legend(title='Social Isolation Rating', bbox_to_anchor=(1.05, 1), loc='upper
plt.xticks(rotation=45)
plt.show()
```



In [45]:
```
mental_health_df.groupby('Work_Location')['Sleep_Quality'].value_counts(normaliz

grouped_data = mental_health_df.groupby('Work_Location')['Sleep_Quality'].value_

ax = grouped_data.plot(kind='bar', stacked=True, colormap='viridis', figsize=(10

plt.xlabel('Work Location')
plt.ylabel('Sleep_Quality')
plt.title('Sleep_Quality per Work Arrangement')

plt.legend(title='Sleep_Quality', bbox_to_anchor=(1.05, 1), loc='upper left', fo
plt.xticks(rotation=45)
plt.show()
```



In [48]:
```
mental_health_df.groupby('Work_Location')['Satisfaction_with_Remote_Work'].value

grouped_data = mental_health_df.groupby('Work_Location')['Satisfaction_with_Remo

ax = grouped_data.plot(kind='bar', stacked=True, colormap='viridis', figsize=(10

plt.xlabel('Work Location')
plt.ylabel('Satisfaction with Work Arrangement')
plt.title('Satisfaction with Work Arrangement')

plt.legend(title='Satisfaction with Work Arrangement', bbox_to_anchor=(1.05, 1),
plt.xticks(rotation=45)
plt.show()
```


Satisfaction with Work Arrangement
Satisfaction with Work Arrangement
Neutral
Satisfied
Unsatisfied
1.0
0.8
0.6
0.4
0.2
0.0
Satisfaction with Work Arrangement
Hybrid
Onsite
Remote
Work Location

In [ ]: