7.1 Project 2. Draft. Milestone 2. Applied Data. Jennifer Barrera Conde

October 9, 2024

1 7.1 Project 2. Draft

2 Milestone 2

3 Applied Data

4 Jennifer Barrera Conde

4.1 Load Data and describe it

[1]:
```
import pandas as pd

# Load dataset
data = pd.read_csv('remote-work-and-mental-health.csv')

# Data info
data.info()

# Display first few rows
data.head()
```

```
<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
```

[1]:

	Employee_ID	Age	Gender	Job_Role	Industry	Years_of_Experience	Work_Location	Hours_Worked_Per_Week	Number_of_Virtual_Meetings	Work_Life_Balance_Rating	Stress_Level	Mental_Health_Condition	Access_to_Mental_Health_Resources	Productivity_Change	Social_Isolation_Rating	Satisfaction_with_Remote_Work
0	EMP0001	32	Non-binary	HR	Healthcare	13	Hybrid	47	7	2	Medium	Depression	No	Decrease	1	Unsatisfied
1	EMP0002	40	Female	Data Scientist	IT	3	Remote	52	4	1	Medium	Anxiety	No	Increase	3	Satisfied
2	EMP0003	59	Non-binary	Software Engineer	Education	22	Hybrid	46	11	5	Medium	Anxiety	No	No Change	4	Unsatisfied
3	EMP0004	27	Male	Software Engineer	Finance	20	Onsite	32	8	4	High	Depression	Yes	Increase	3	Unsatisfied
4	EMP0005	49	Male	Sales	Consulting	32	Onsite	35	12	2	High	NaN	Yes	Decrease	3	Unsatisfied

```
   Company_Support_for_Remote_Work Physical_Activity Sleep_Quality  \
0                                1            Weekly          Good
1                                2            Weekly          Good
2                                5               NaN          Poor
3                                3               NaN          Poor
4                                3            Weekly       Average

          Region
0         Europe
1           Asia
2  North America
3         Europe
4  North America
```

[2]:
```
import numpy as np

# Summary statistics
print(data.describe())

# Check for missing values
print(data.isnull().sum())
```

```
                Age  Years_of_Experience  Hours_Worked_Per_Week  \
count  5000.000000          5000.000000            5000.000000
mean     40.995000            17.810200              39.614600
std      11.296021            10.020412              11.860194
min      22.000000             1.000000              20.000000
25%      31.000000             9.000000              29.000000
50%      41.000000            18.000000              40.000000
75%      51.000000            26.000000              50.000000
max      60.000000            35.000000              60.000000

       Number_of_Virtual_Meetings  Work_Life_Balance_Rating  \
count                 5000.000000               5000.000000
mean                     7.559000                  2.984200
std                      4.636121                  1.410513
min                      0.000000                  1.000000
25%                      4.000000                  2.000000
50%                      8.000000                  3.000000
75%                     12.000000                  4.000000
max                     15.000000                  5.000000

       Social_Isolation_Rating  Company_Support_for_Remote_Work
count              5000.000000                      5000.000000
mean                  2.993800                         3.007800
std                   1.394615                         1.399046
min                   1.000000                         1.000000
25%                   2.000000                         2.000000
```

```
50%                                 3.000000                              3.000000
75%                                 4.000000                              4.000000
max                                 5.000000                              5.000000
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
```

[6]:
```python
# Handle missing data
# Drop rows with missing values
data_clean = data.dropna()
```

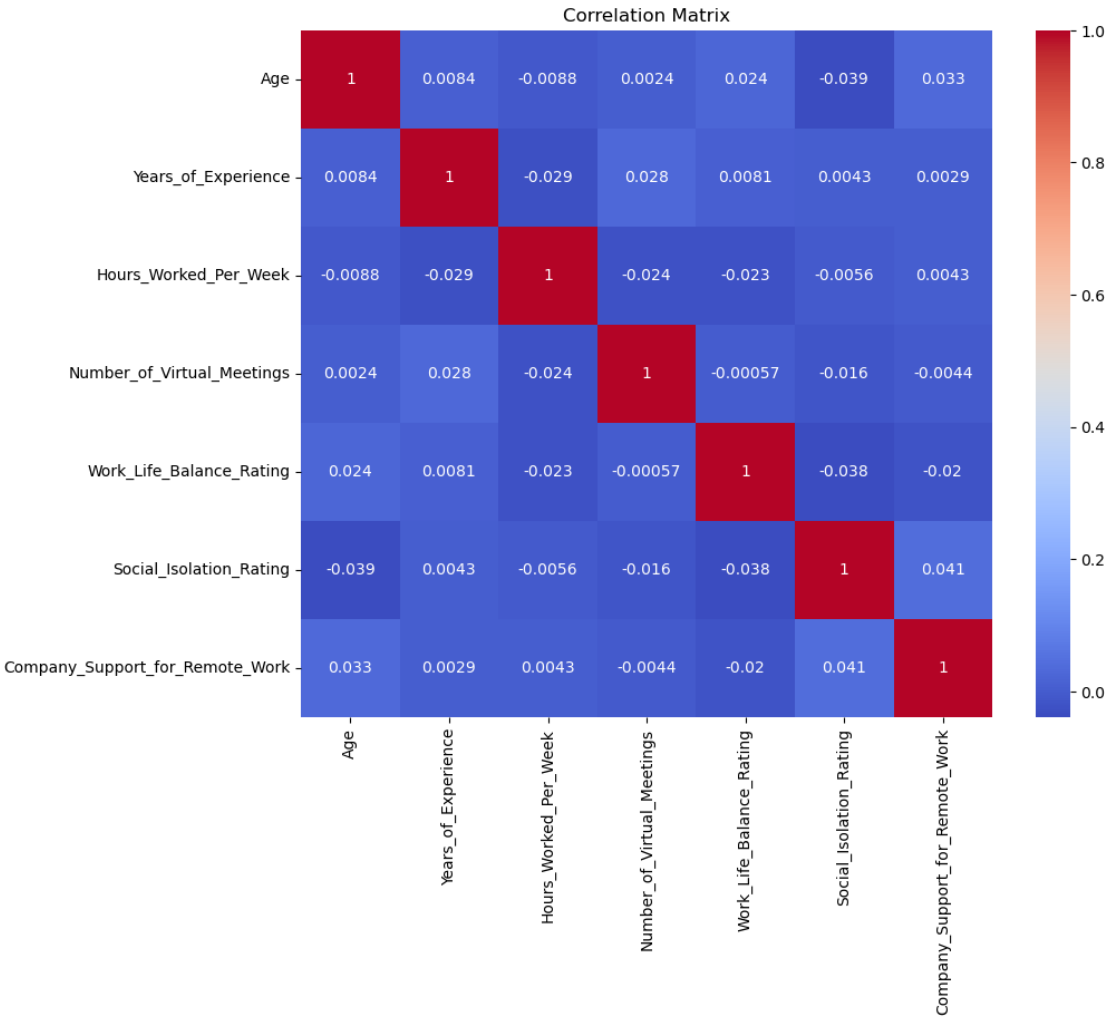
[ ]:
```python

```

4.2 Correlation Matrix:

Analyze correlations between numeric variables to explore relationships, especially focusing on mental health indicators.

[7]:
```python
import seaborn as sns
import matplotlib.pyplot as plt

# Select only numeric columns from the dataset for correlation analysis
numeric_data = data_clean.select_dtypes(include=['float64', 'int64'])

# Calculate correlation matrix for numeric data
corr_matrix = numeric_data.corr()

# Plot heatmap
plt.figure(figsize=(10,8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
```

```
plt.title('Correlation Matrix')
plt.show()
```



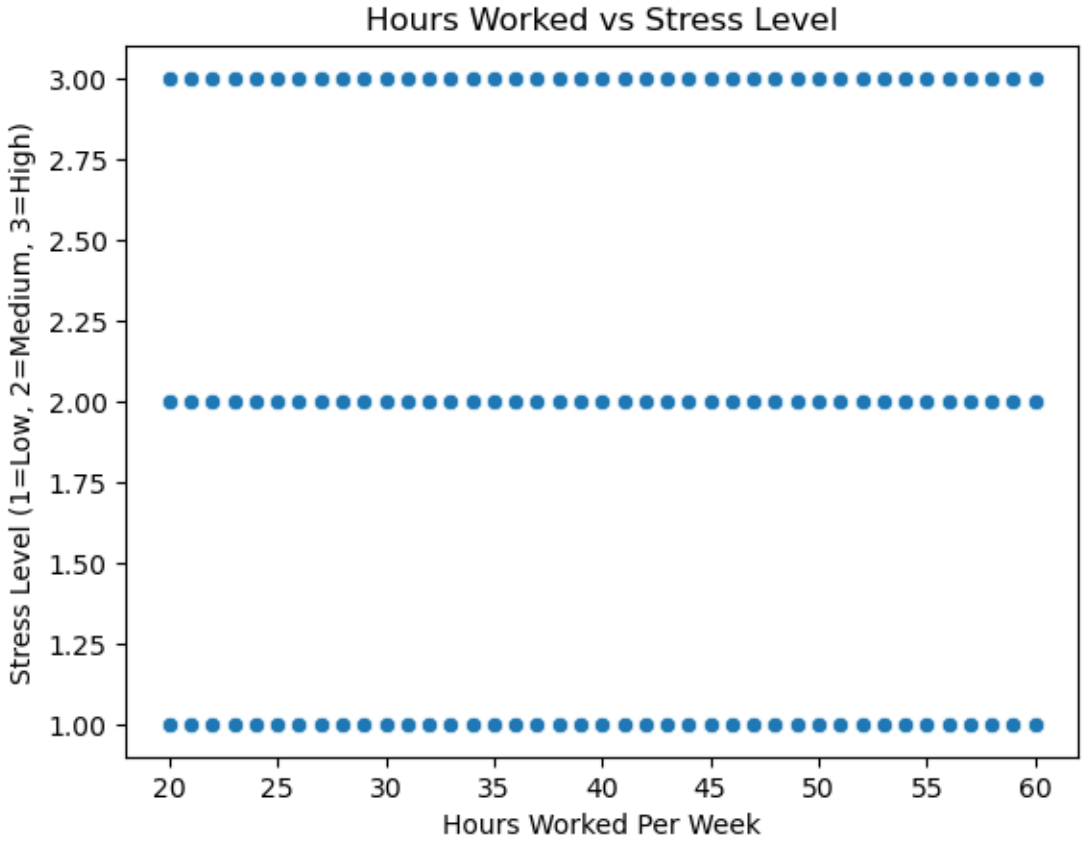
[ ]:

4.3 Data Visualization:

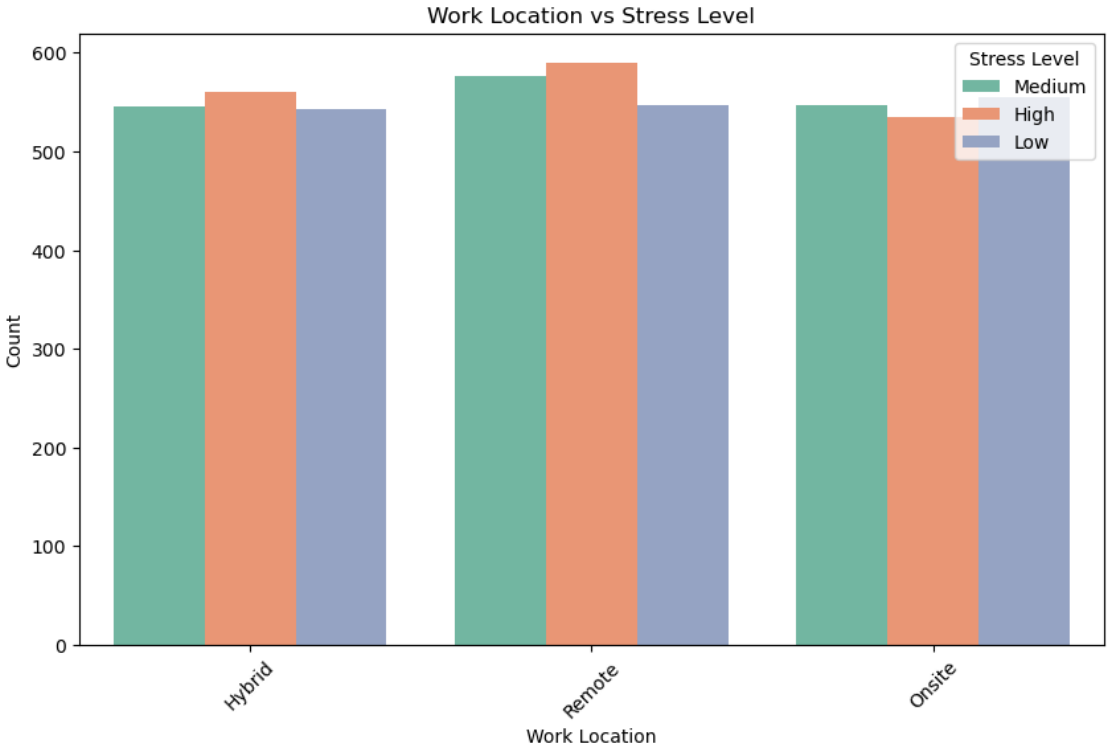
Usage of visualizations to detect patterns, such as the relationship between working hours and stress levels or job satisfaction.

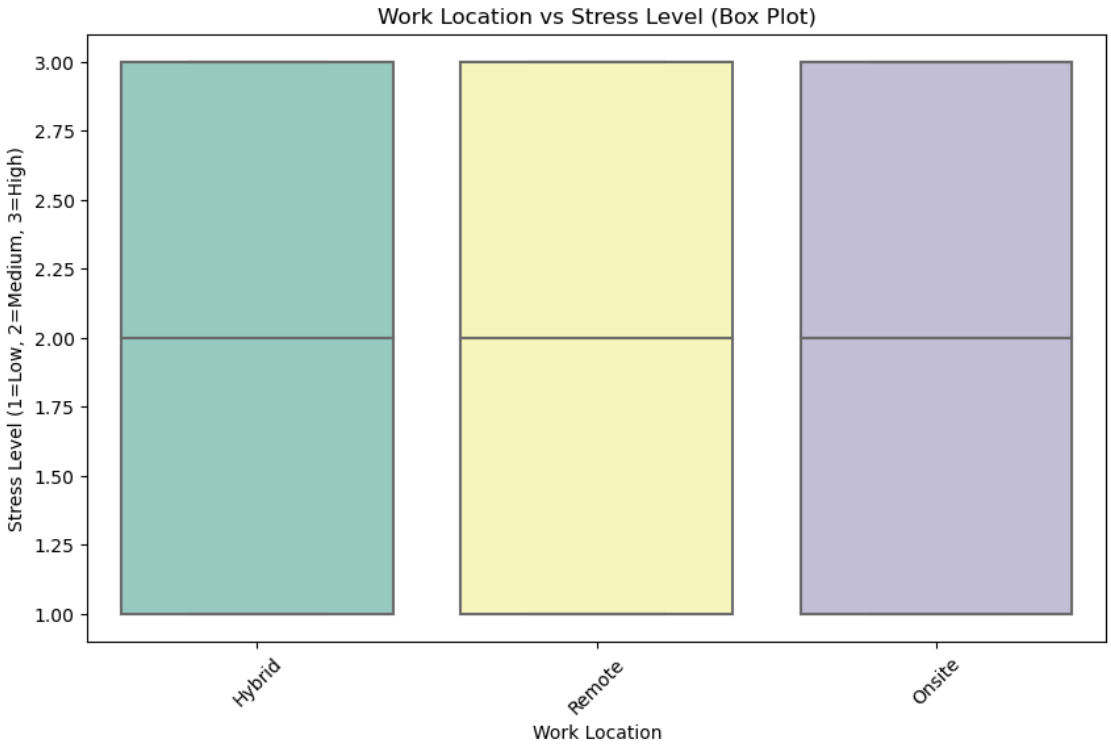
[15]:
```
# First, let's clean up the column names to remove any extra spaces and make␣
↪them uniform
data.columns = data.columns.str.strip()

# Encode 'Stress_Level' as a numeric variable (Low=1, Medium=2, High=3, etc.)
```

```
stress_mapping = {'Low': 1, 'Medium': 2, 'High': 3}
data['Stress_Level_Encoded'] = data['Stress_Level'].map(stress_mapping)

# Check if there are any missing values in Hours_Worked_Per_Week or␣
 ↪Stress_Level_Encoded
data[['Hours_Worked_Per_Week', 'Stress_Level_Encoded']].isnull().sum()

# Now let's generate the scatter plot
import seaborn as sns
import matplotlib.pyplot as plt

# Plot: Hours Worked vs Stress Level
sns.scatterplot(data=data, x='Hours_Worked_Per_Week', y='Stress_Level_Encoded')
plt.title('Hours Worked vs Stress Level')
plt.xlabel('Hours Worked Per Week')
plt.ylabel('Stress Level (1=Low, 2=Medium, 3=High)')
plt.show()
```



[18]:
```
# First, let's create a grouped bar plot to compare Work_Location with␣
 ↪Stress_Level
import seaborn as sns
import matplotlib.pyplot as plt

# Grouped bar plot: Work_Location vs Stress_Level
plt.figure(figsize=(10,6))
sns.countplot(data=data, x='Work_Location', hue='Stress_Level', palette='Set2')
plt.title('Work Location vs Stress Level')
plt.xlabel('Work Location')
plt.ylabel('Count')
plt.xticks(rotation=45)
plt.legend(title='Stress Level')
plt.show()

# Box plot to visualize distribution of Stress_Level (encoded) for each␣
 ↪Work_Location
plt.figure(figsize=(10,6))
sns.boxplot(data=data, x='Work_Location', y='Stress_Level_Encoded',␣
 ↪palette='Set3')
plt.title('Work Location vs Stress Level (Box Plot)')
plt.xlabel('Work Location')
plt.ylabel('Stress Level (1=Low, 2=Medium, 3=High)')
plt.xticks(rotation=45)
plt.show()
```




Work Location vs Stress Level (Box Plot)
Stress Level (1=Low, 2=Medium, 3=High)
3.00
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
Hybrid
Remote
Onsite
Work Location

[ ]: